Nuclear discontinuities through the prism of Machine Learning

R.-D. Lasseri[1], D. Regnier[2], S. Hilaire[2],

[1]Centre Borelli, ENS-Paris Saclay, France
[2]CEA DAM/DIF, France

Table of contents

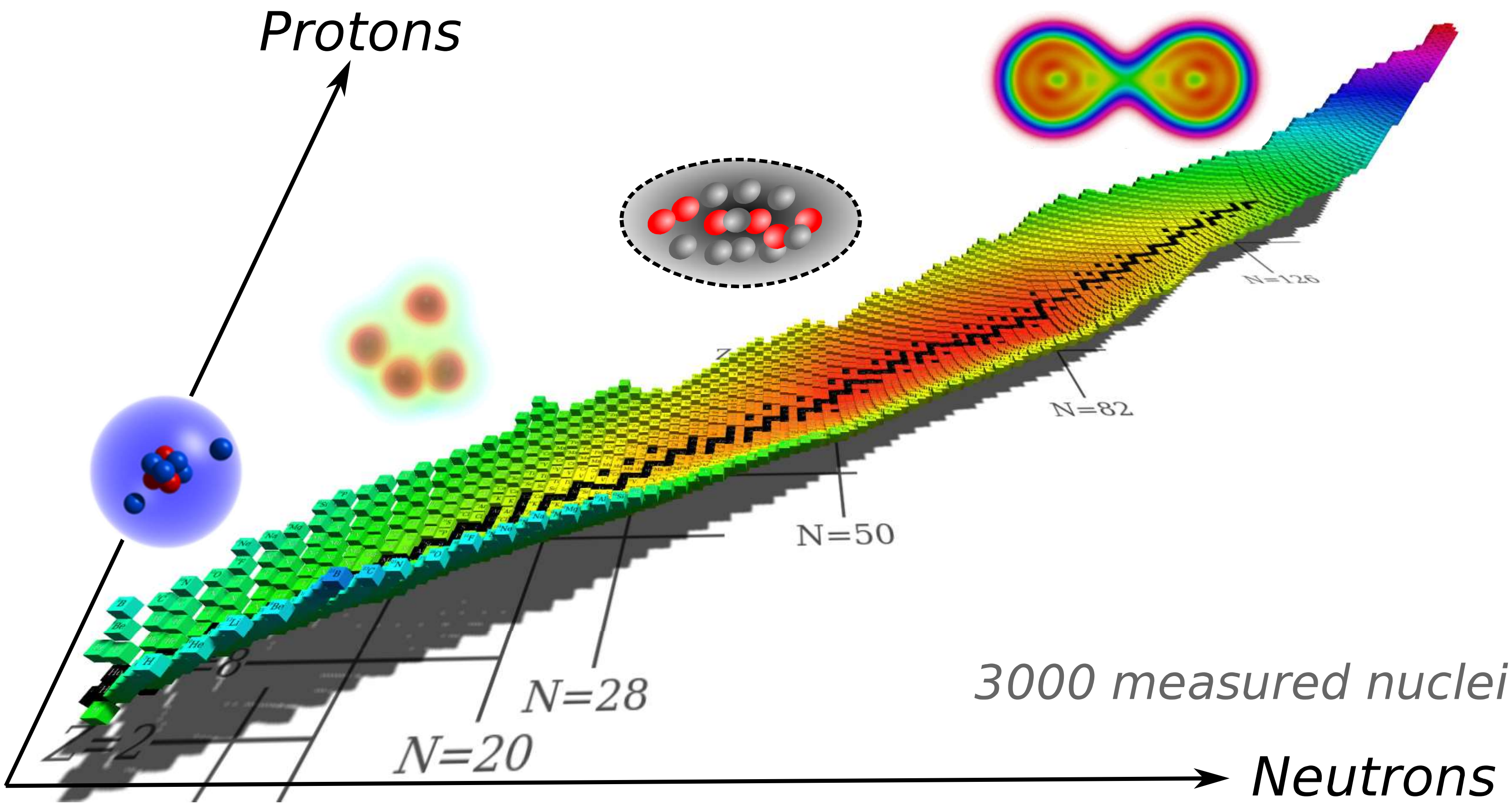
Predicting nuclear structure on the whole chart...

Why so complex ?

- Three fundamental interactions
- Non elementary fermions
- Mesoscopic many-body problem

Some open questions

- Properties of exotic matter ?
- Mechanisms of nucleosynthesis ?
- Super-heavy island of stability ?



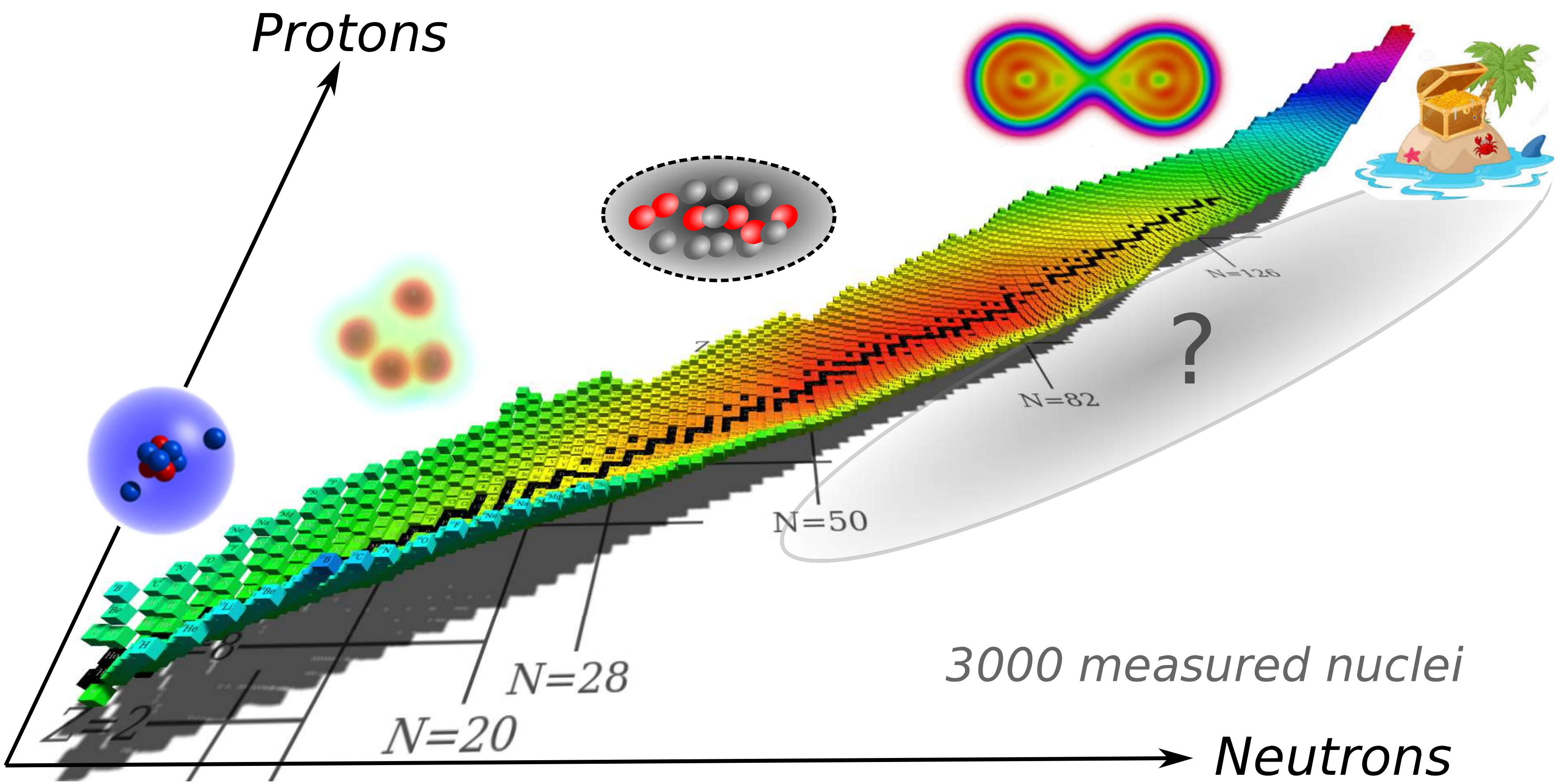
Predicting nuclear structure on the whole chart...

Why so complex ?

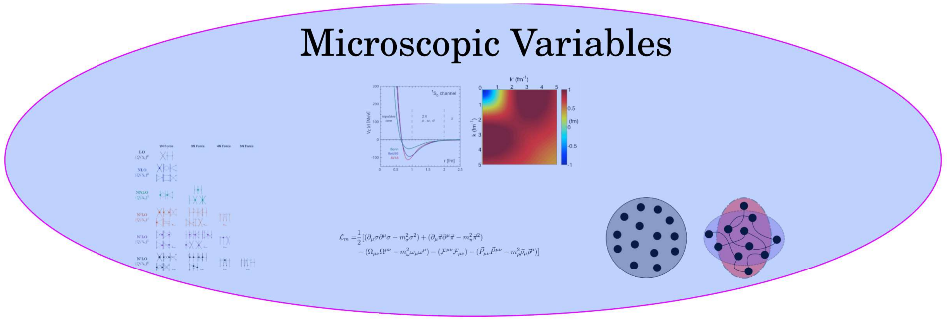
- Three fundamental interactions
- Non elementary fermions
- Mesoscopic many-body problem

Some open questions

- Properties of exotic matter ?
- Mechanisms of nucleosynthesis ?
- Super-heavy island of stability ?

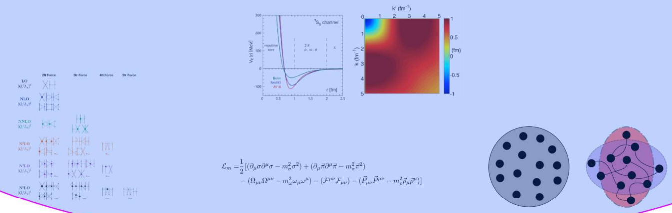


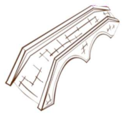
Building bridges – Global Strategy



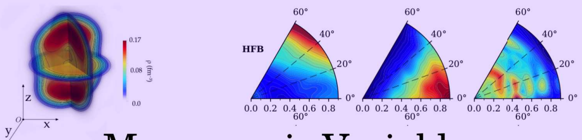
Building bridges – Global Strategy

Microscopic Variables



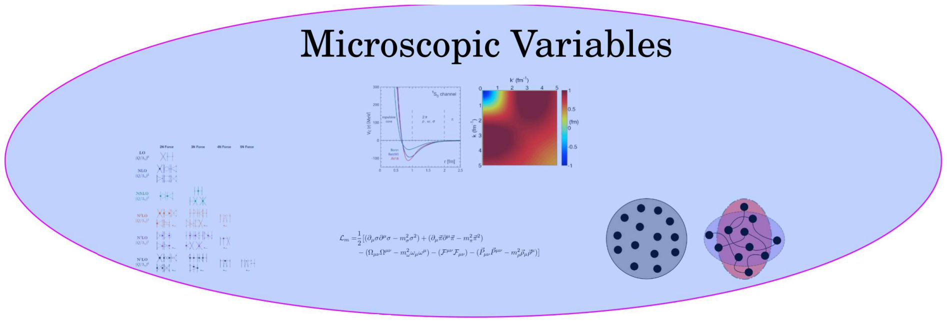


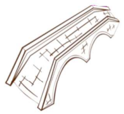
Modelisation {
- HFB
- Ab-Initio
- Machine Learning



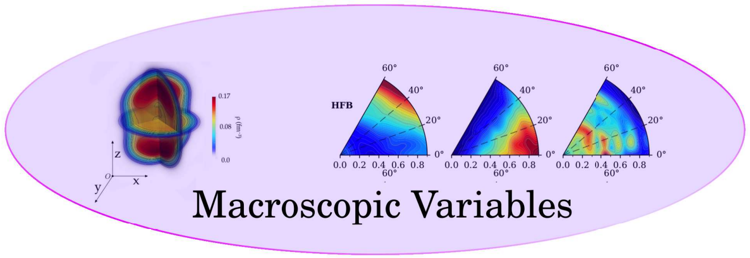
Macroscopic Variables

Building bridges – Global Strategy

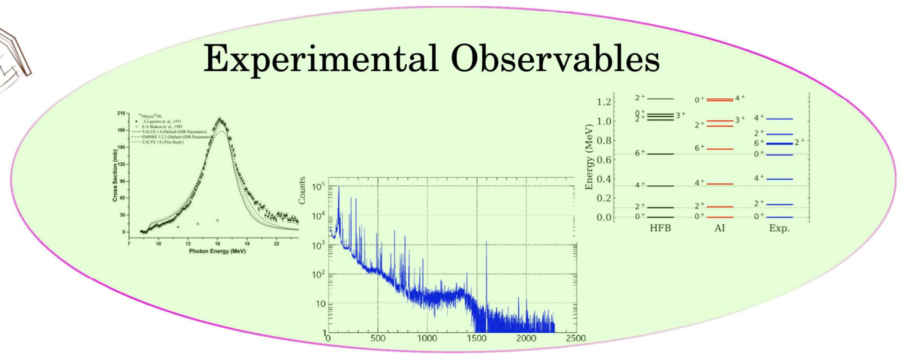




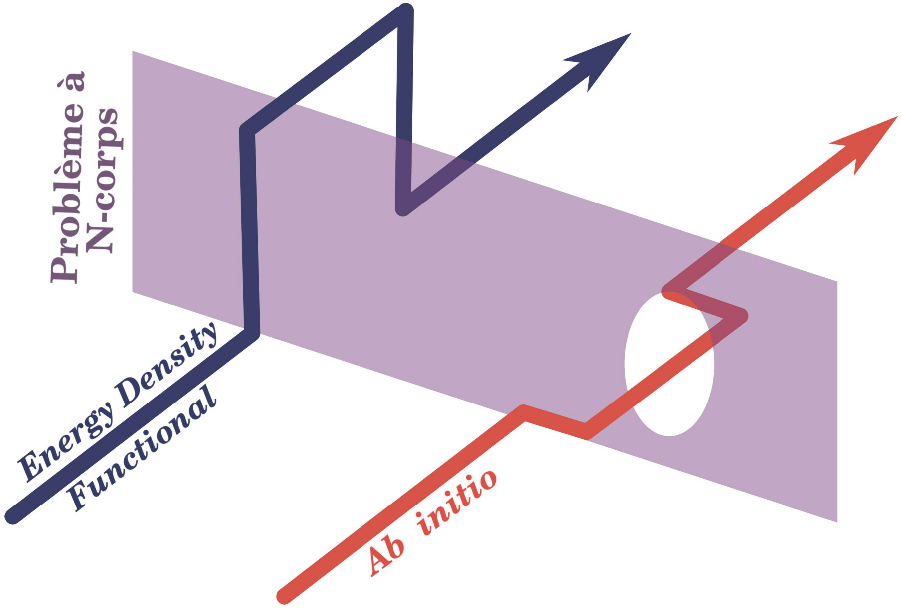
Modelisation {
- HFB
- Ab-Initio
- Machine Learning



Evaluation {
- QRPA/5DCH
- Machine Learning



Limits of our current paradigm



Limitations

- No link with QCD
- Difficult to link with experiments
- Spuriosities
- Numerical Cost

Toward a new paradigm ?

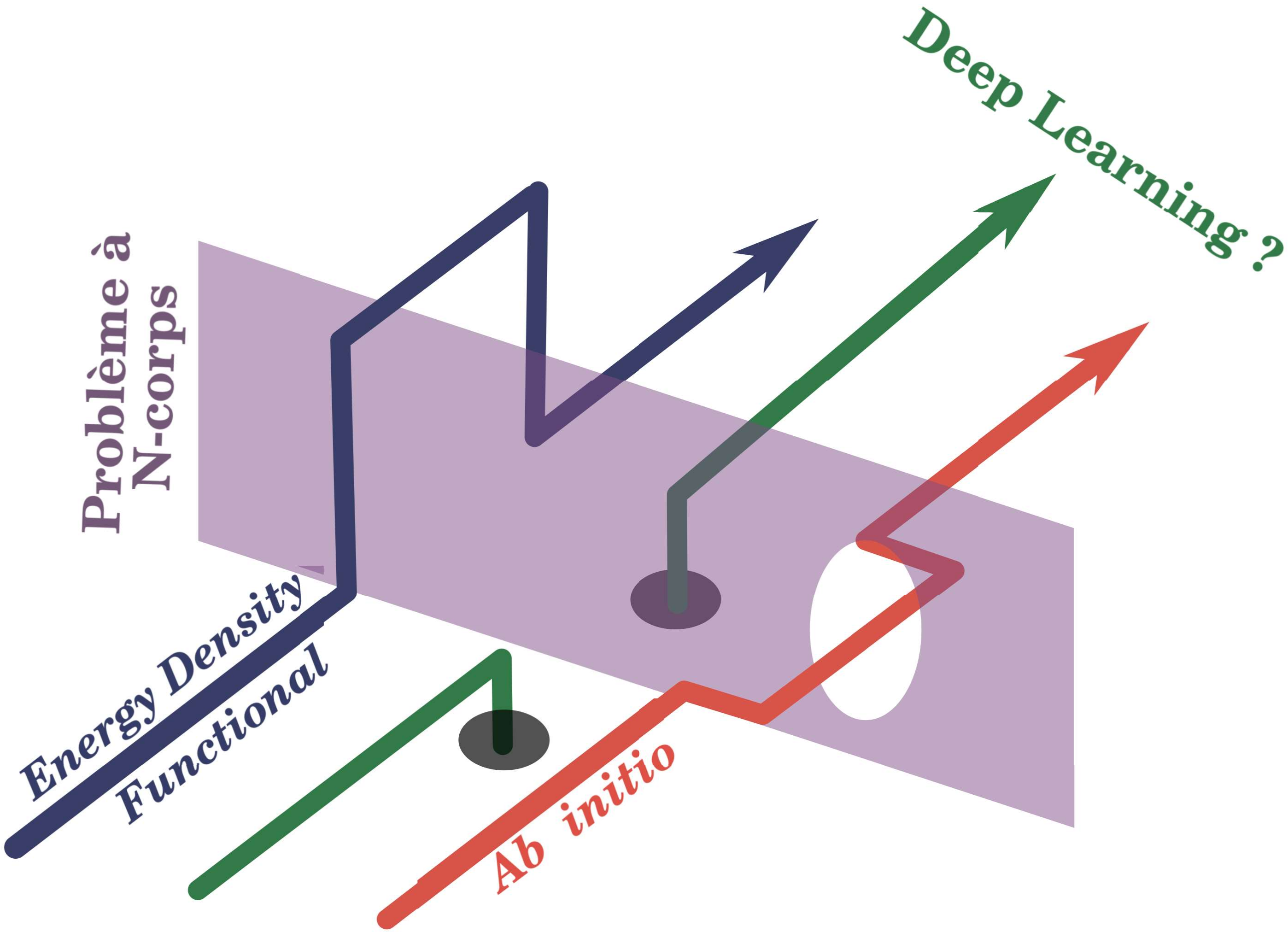


Table of contents

Statistical Learning – A fundamental postulate

- Let X be a vector space of all possible "inputs"
- Let Y be a vector space of all the possible "outputs"
- Let Z be the product space $Z = X \times Y$

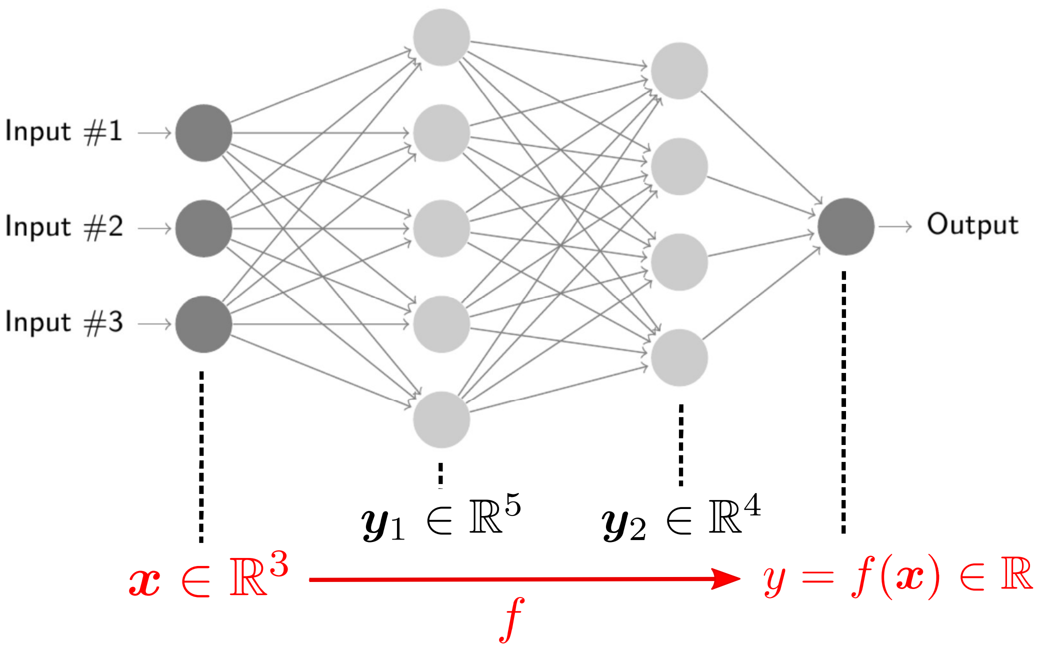
Postulate

$$\exists p \in \mathcal{Z} | p(z) = p(x, y)$$

Where p is an unknown probability distribution mapping the "inputs" to the "outputs"

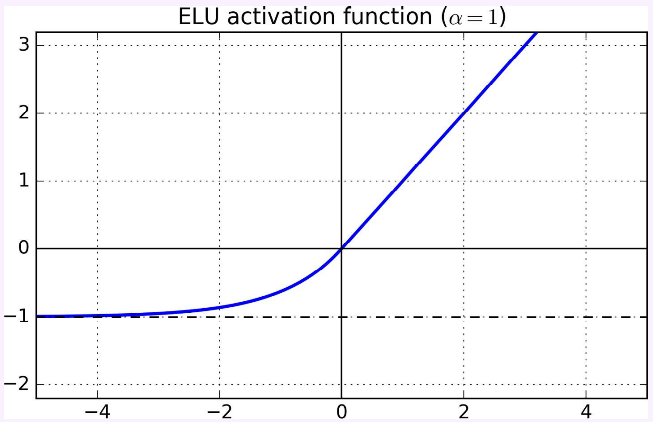
In Physics, because of the causal assumption $\rightarrow$ **There is** something to learn.

Machine Learning – Neural network



$$\boldsymbol{y}_1 = f_1(\boldsymbol{x}) = A_1(W_1 \cdot \boldsymbol{x} + \boldsymbol{b}_1)$$
$$\boldsymbol{y}_2 = f_2(\boldsymbol{y}_1) = A_2(W_2 \cdot \boldsymbol{y}_1 + \boldsymbol{b}_1)$$
$$y = f_3(\boldsymbol{y}_2) = A_3(W_3 \cdot \boldsymbol{y}_2 + \boldsymbol{b}_2)$$
$$y = f(\boldsymbol{x}) = f_3 \circ f_2 \circ f_1(\boldsymbol{x})$$

A_1, A_2, A_3 = non-linear functions.



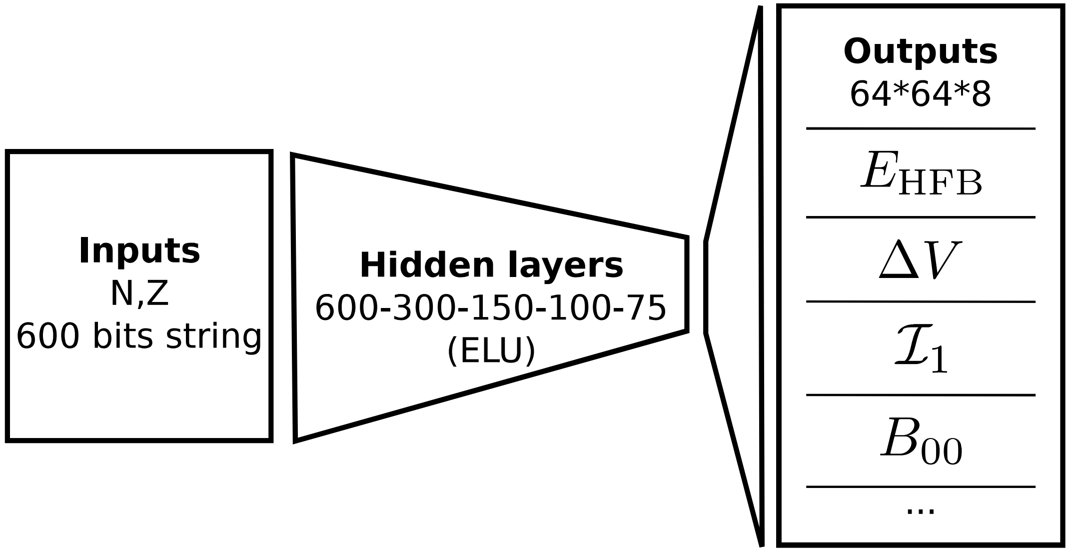
W_1, W_2, W_3 = matrices,
$\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3$ = vectors.

We fit these parameters so to reproduce some training data $(\boldsymbol{x}^i, y^i), i \in [0, N]$.

Table of contents

Building the neural network

Architecture:

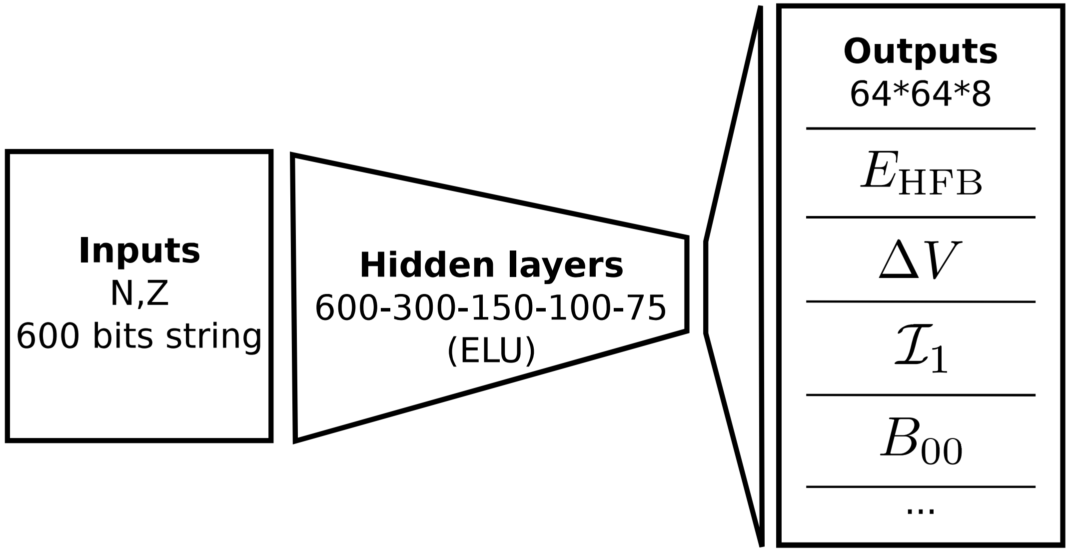


Implementation:

- Keras/TensorFlow
- Fast GPU execution

Building the neural network

Architecture:



Implementation:

- Keras/TensorFlow
- Fast GPU execution

TensorFlow NVIDIA

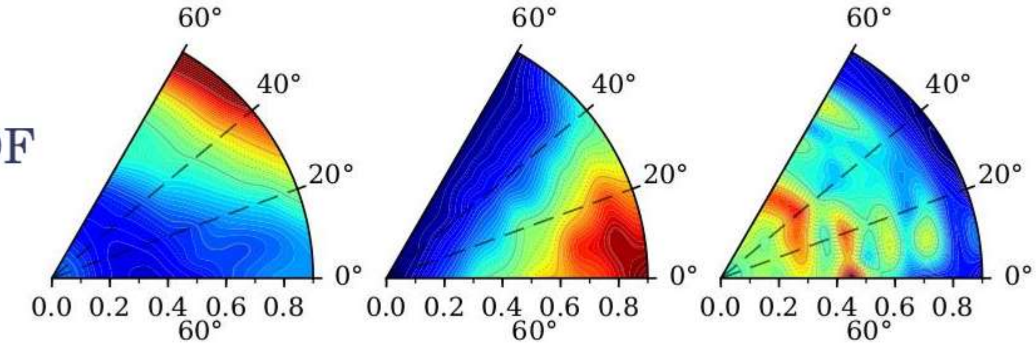
Training:

- Training set: sample from 2100 even-even nuclei, Gogny D1S functional
- Loss function based on a weighted sum of:

$$\mathcal{L}_t(N,Z) = \frac{6}{\pi B^2} \int_{\beta,\gamma} \left| t_{\text{AI}}(\beta,\gamma) - t_{\text{HFB}}(\beta,\gamma) \right|^2 \mathrm{d}\beta \beta \mathrm{d}\gamma, \quad (1)$$

with $t = E_{HFB}, \Delta V, \mathcal{I}_1, \dots$.

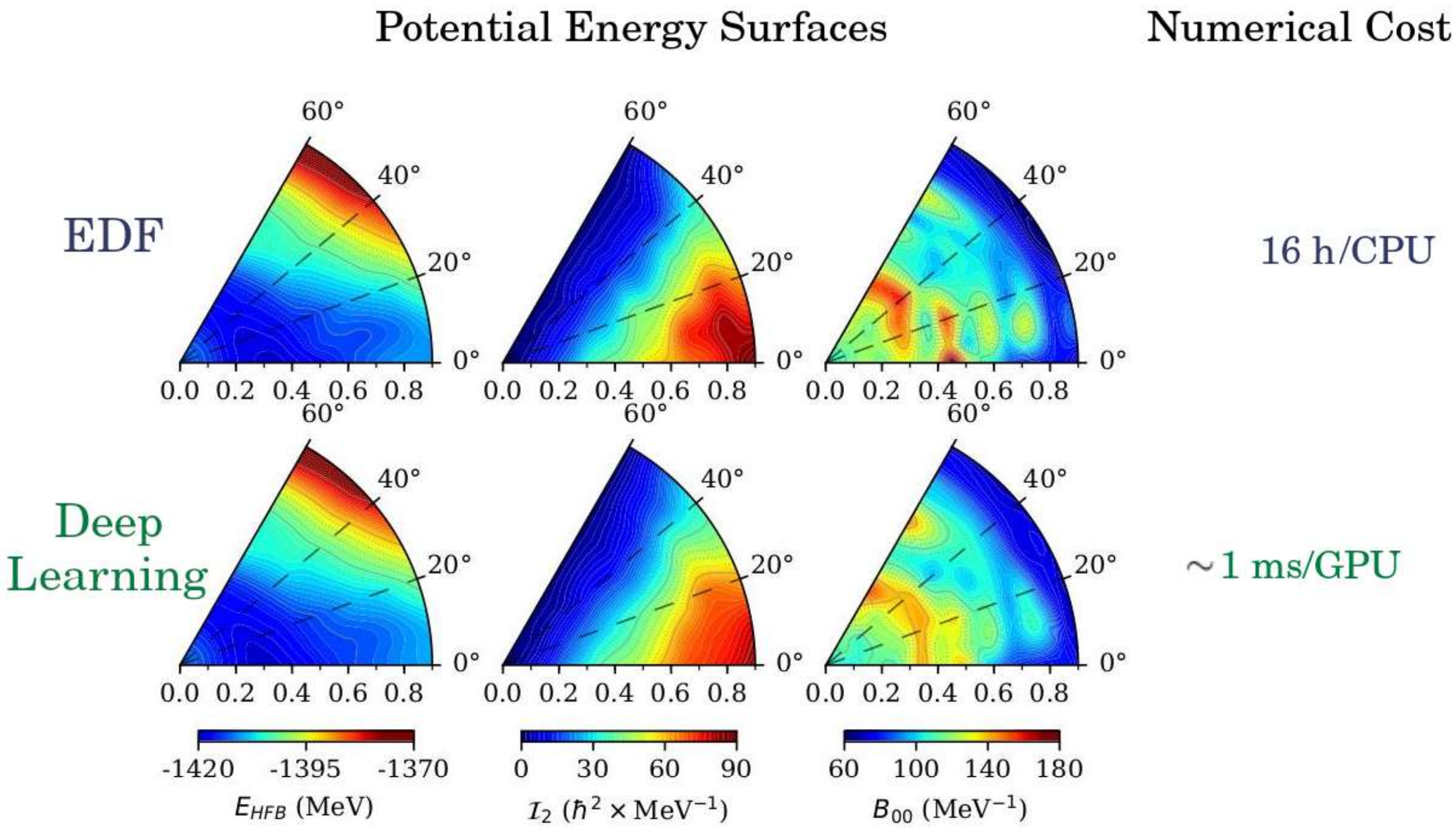
PES Prediction using Deep Learning – Published in Phys Rev Lett (2020)

	Potential Energy Surfaces	Numerical Cost
EDF		16 h/CPU

Applications

- Accurate and fast observable predictions (experimental and astrophysical applications)
- New EDF families (Ongoing work @ULB)

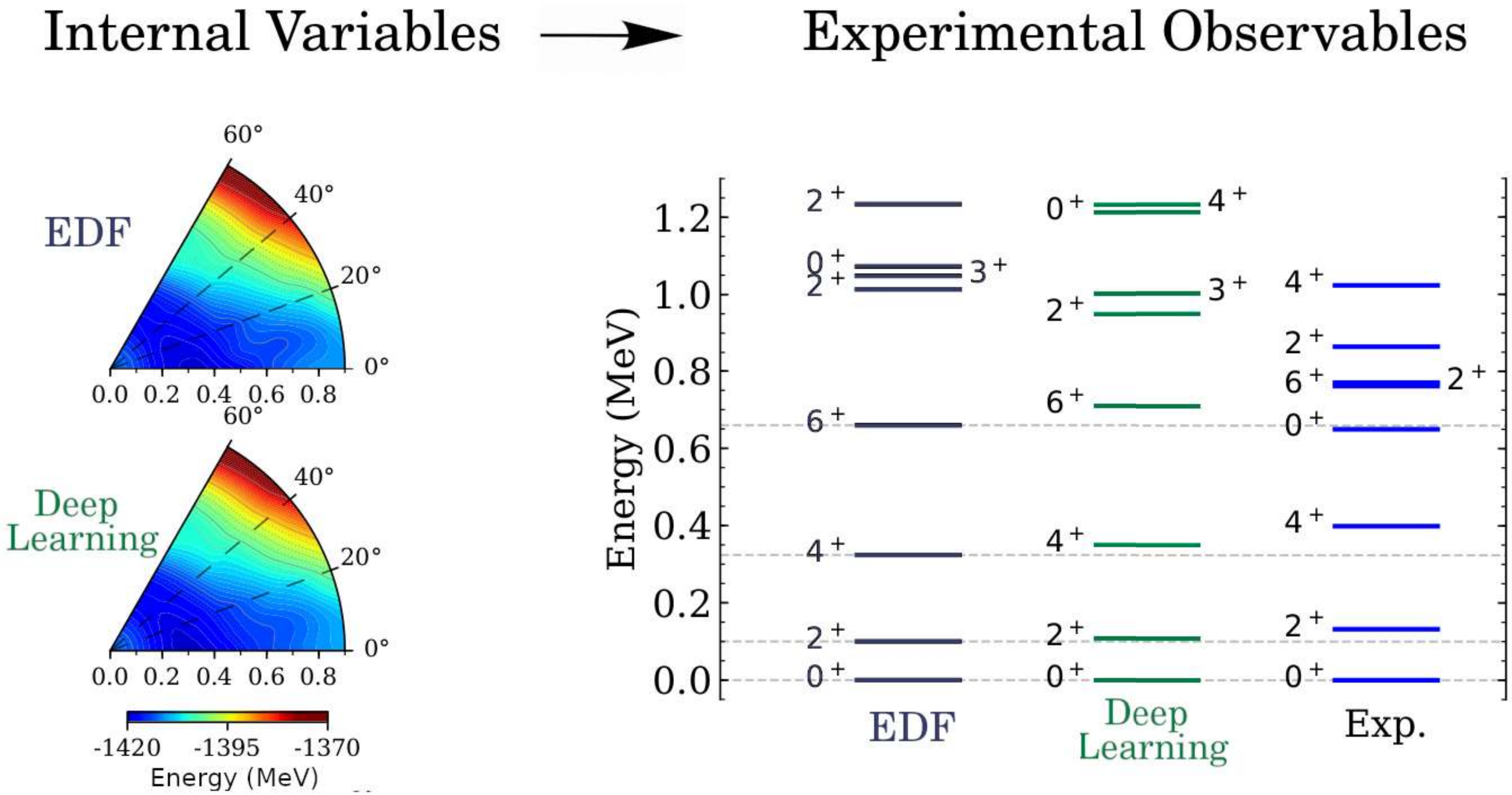
PES Prediction using Deep Learning – Published in Phys Rev Lett (2020)



Applications

- Accurate and fast observable predictions (experimental and astrophysical applications)
- New EDF families (Ongoing work @ULB)

PES Prediction using Deep Learning – Published in Phys Rev Lett (2020)

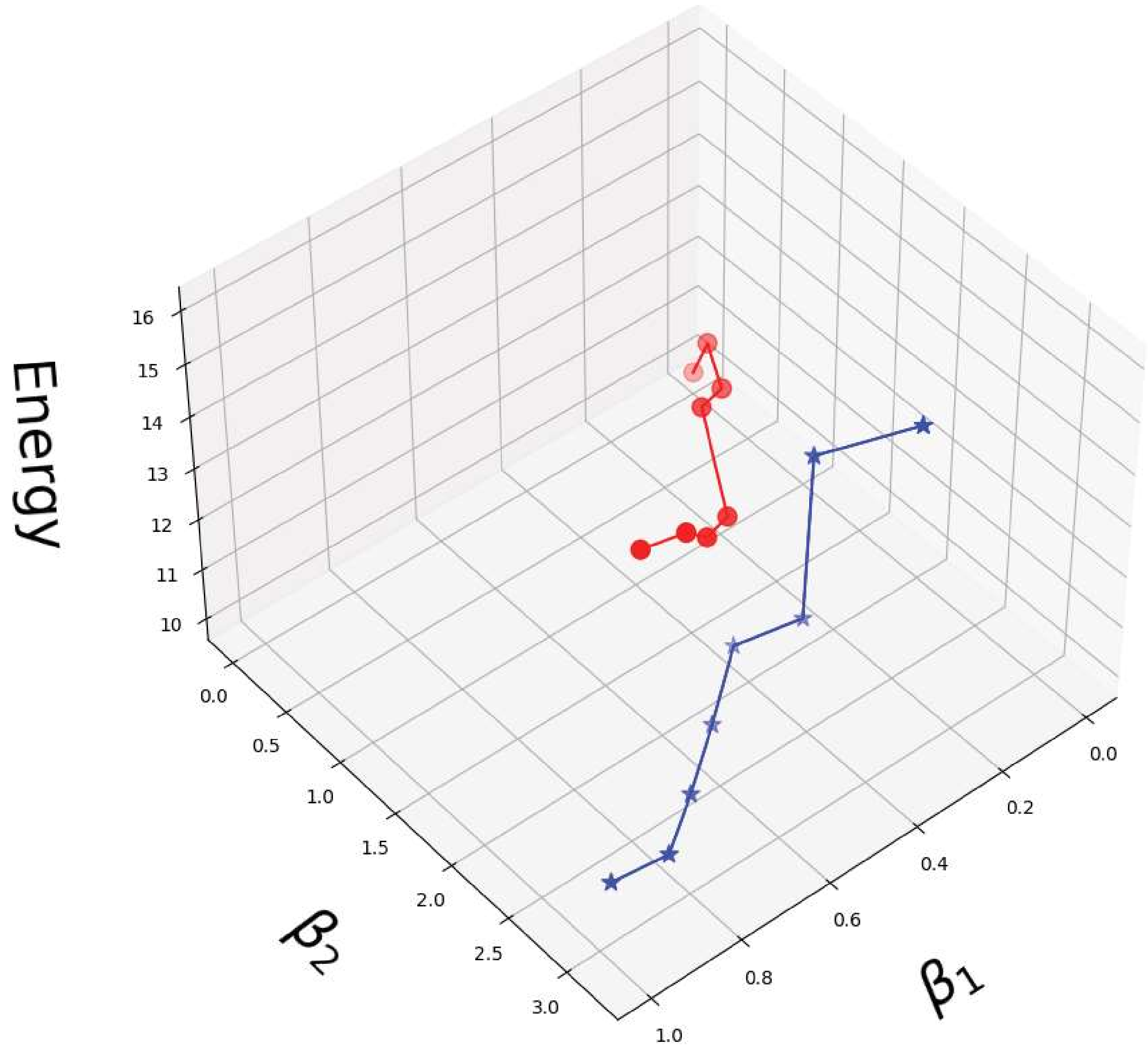


Applications

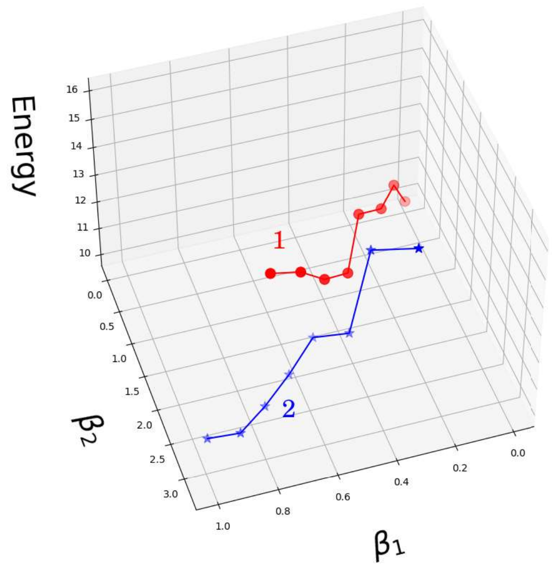
- Accurate and fast observable predictions (experimental and astrophysical applications)
- New EDF families (Ongoing work @ULB)

Table of contents

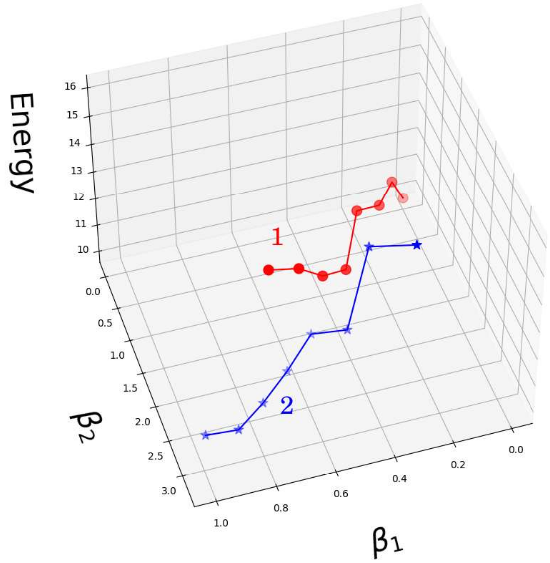
Discontinuities – A painfull story

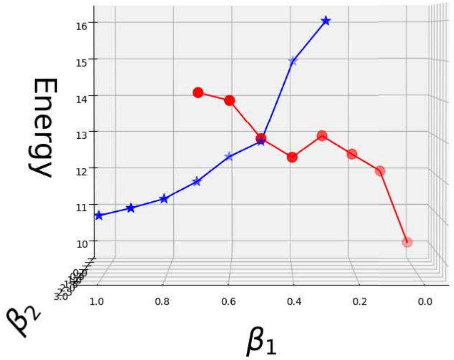


Discontinuities – A painfull story

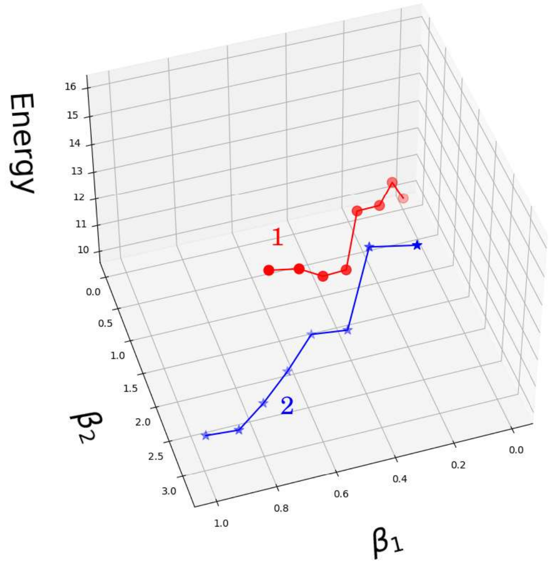


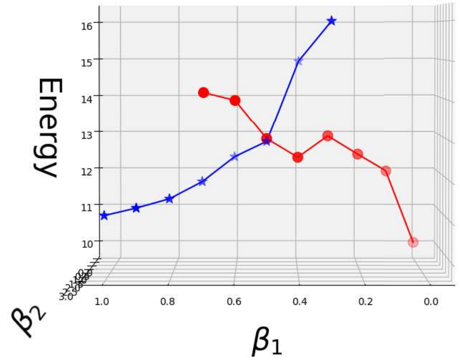
Discontinuities – A painfull story

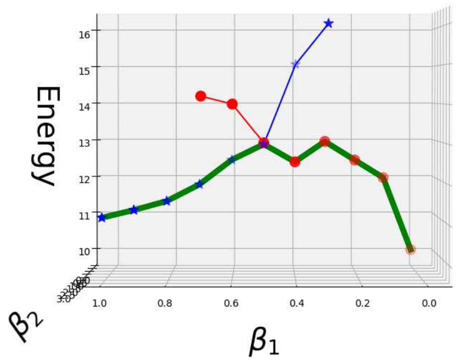




Discontinuities – A painfull story

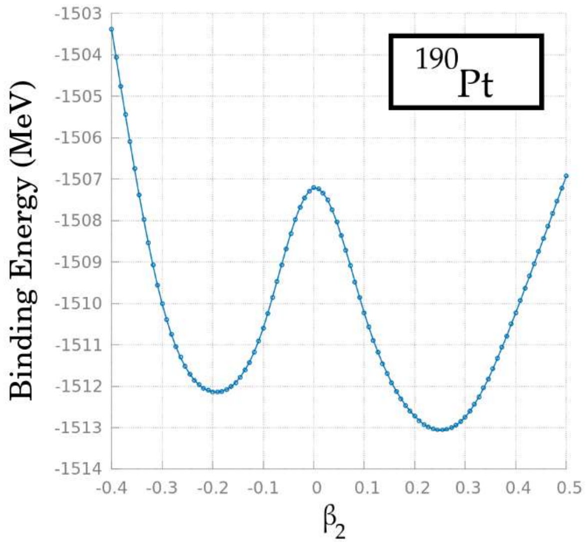






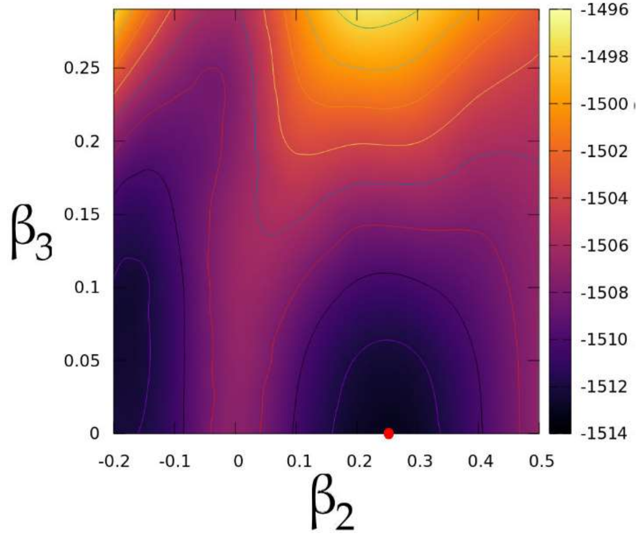
Discontinuities – How to get rid of them ?

- Bruteforce
- "Smart" smoothing: DPM Method
- A curious alternative: Generative Machine Learning



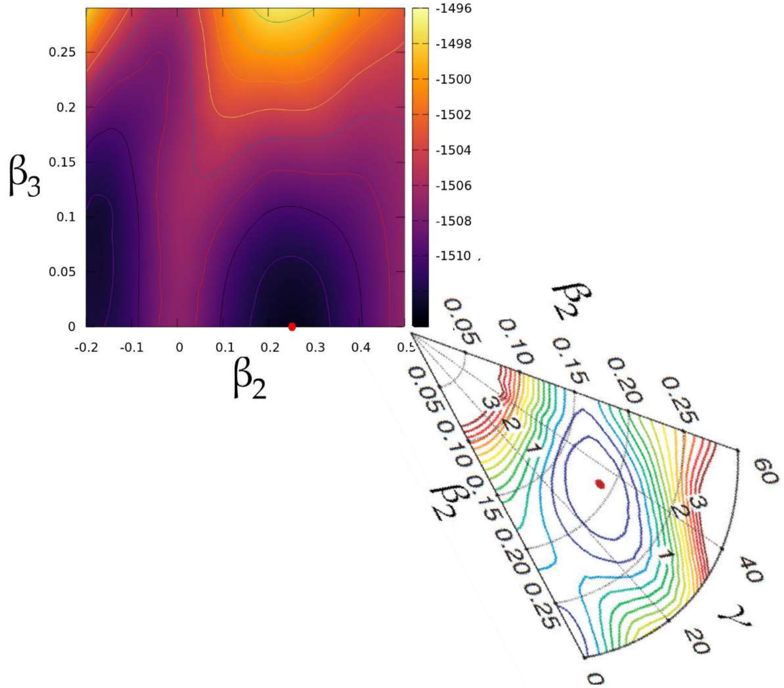
Discontinuities – How to get rid of them ?

- Bruteforce
- "Smart" smoothing: DPM Method
- A curious alternative: Generative Machine Learning



Discontinuities – How to get rid of them ?

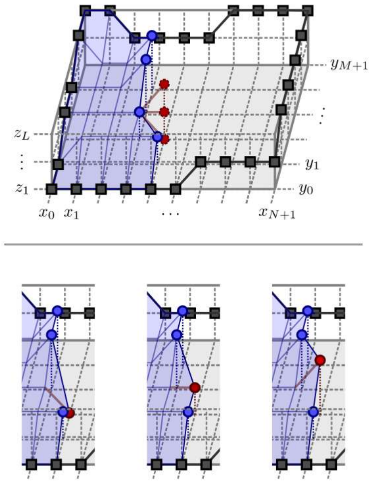
- Bruteforce
- "Smart" smoothing: DPM Method
- A curious alternative: Generative Machine Learning



Exponential increases of computation time

Discontinuities – How to get rid of them ?

- Bruteforce
- "Smart" smoothing: DPM Method
- A curious alternative: Generative Machine Learning

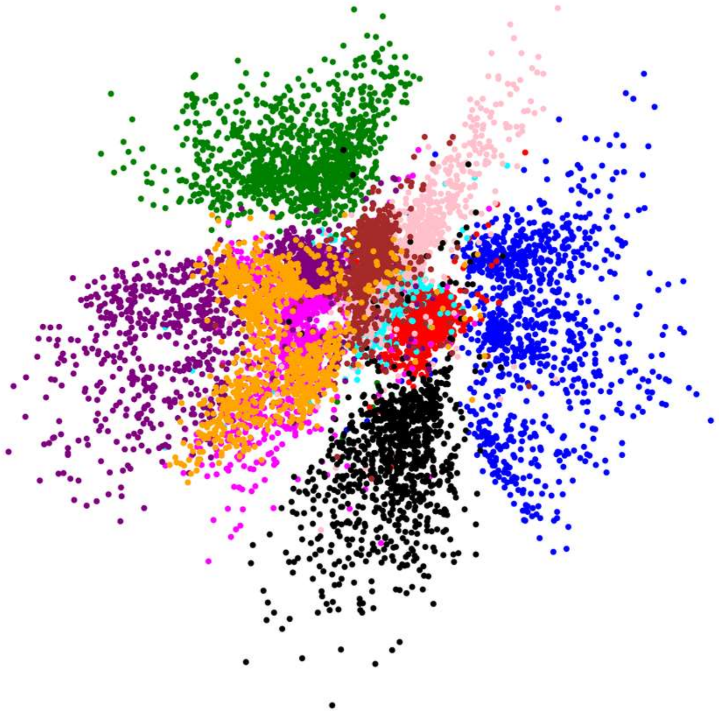


Promising, yet expensive and relies on physicist insights (Hi Rémi ! :)) [1]

[1] Lau, Bernard, Simenel

Discontinuities – How to get rid of them ?

- Bruteforce
- "Smart" smoothing: DPM Method
- A curious alternative: Generative Machine Learning



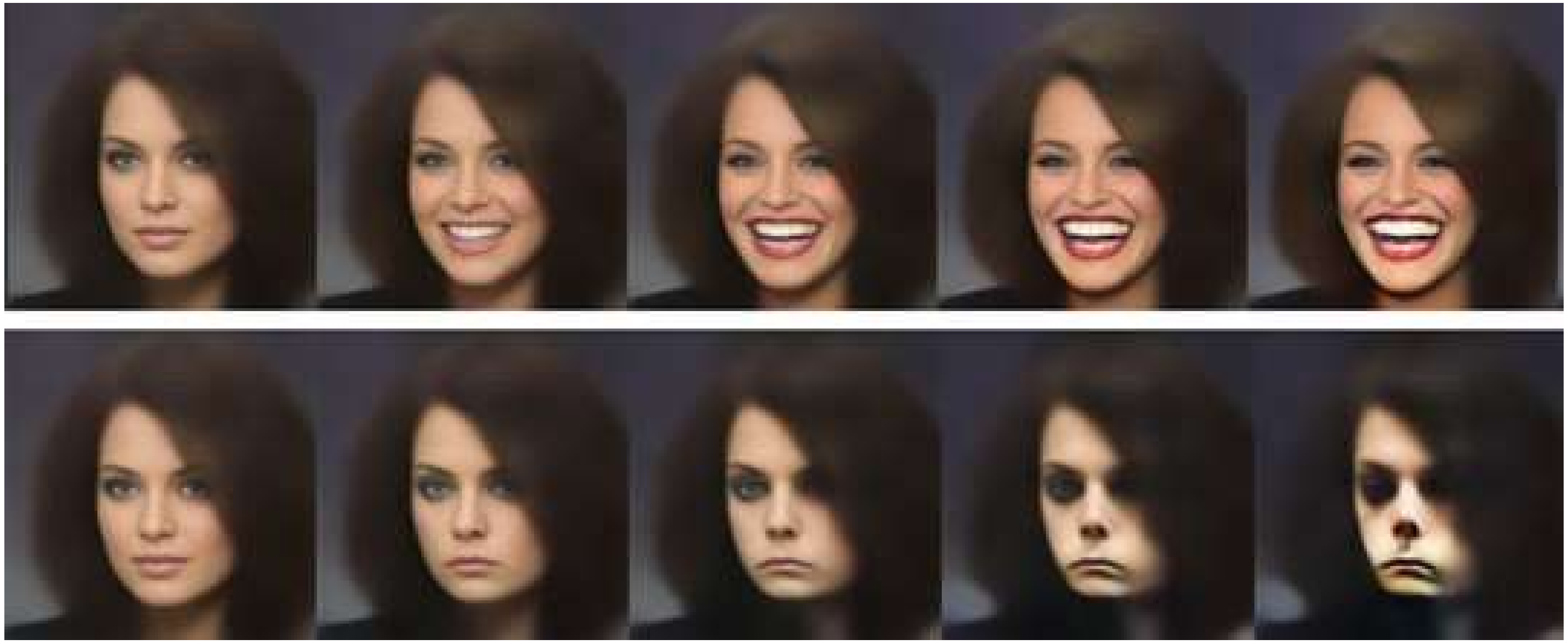
Physically *cumbersome* but very promising

General idea: building manifolds of many-body states

Generative Adversarial Networks, Auto Encoders: capacity to

1. Reduce information to a small optimal latent space (neck)
2. Generate a continuous outputs from the latent space

Example: the smile vector (T. White, Victoria Univ. of Wellington)

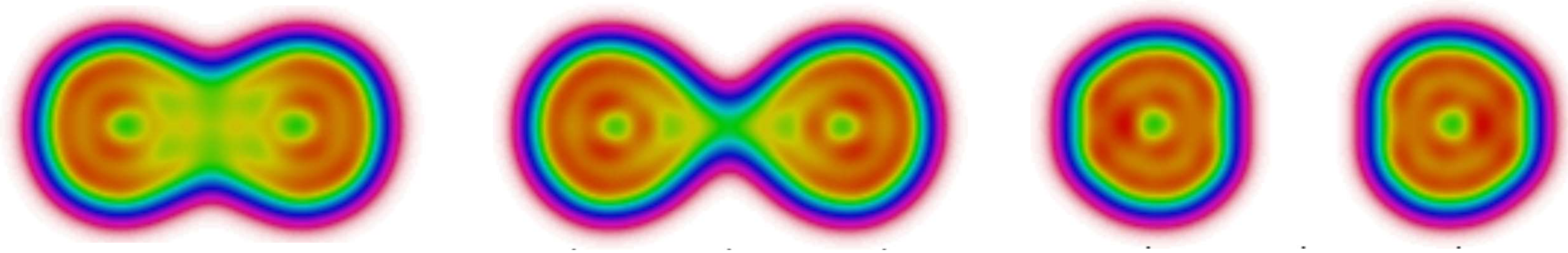


General idea: building manifolds of many-body states

Generative Adversarial Networks, Auto Encoders: capacity to

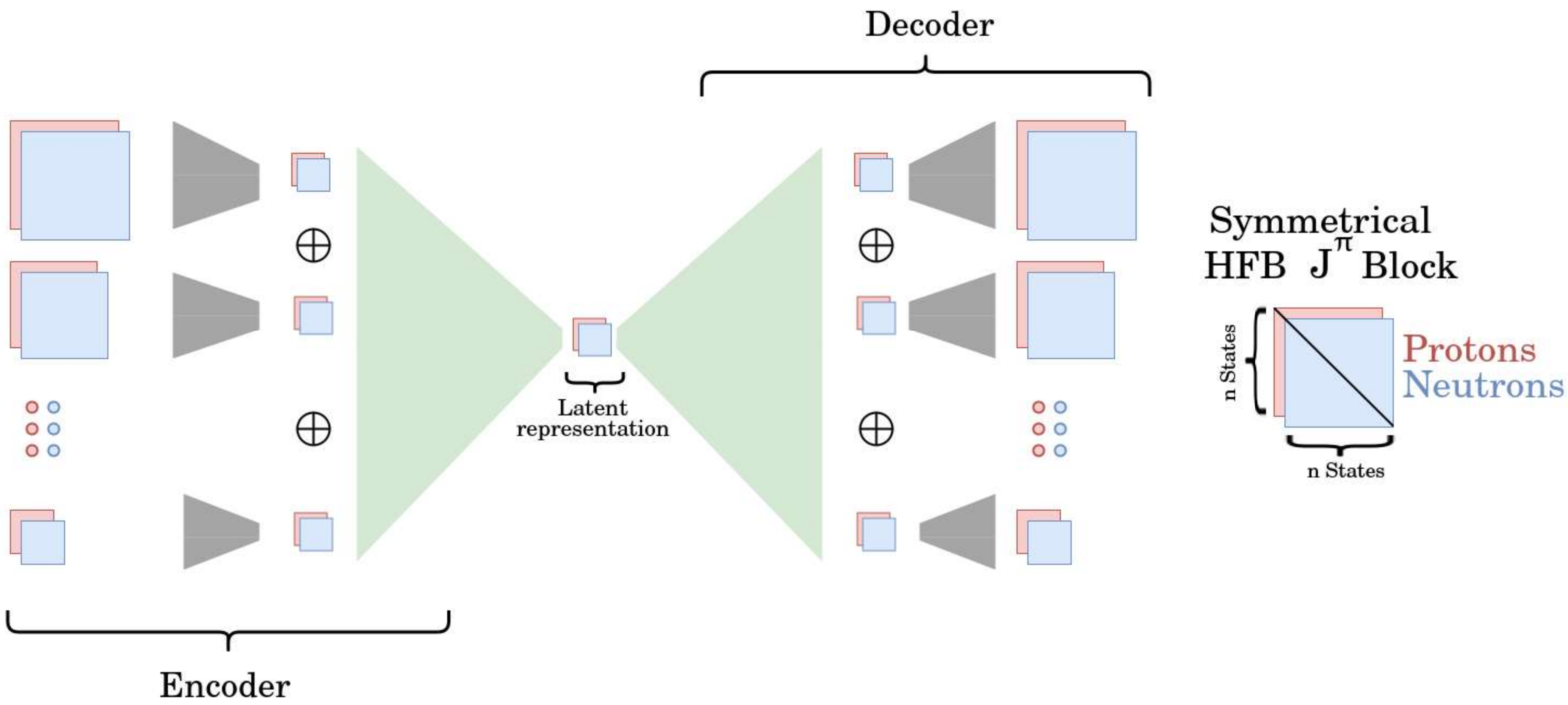
1. Reduce information to a small optimal latent space (neck)
2. Generate a continuous outputs from the latent space

Project: continuous manifolds of Hartree-Fock-Bogoliubov states



A new way to include the diabatic effects in our description of fission ?

Cascade Auto Encoder: Reconstruction the full HFB matrix – Training



Losses:

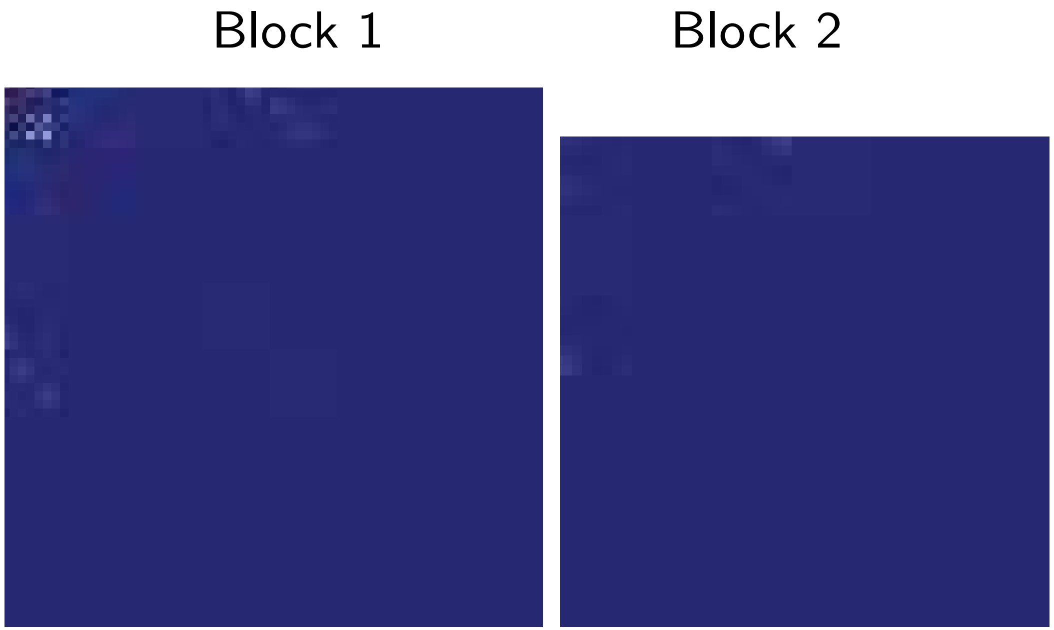
- Mean Square Error / Mean Absolute Error
- Mean Square Error + Trace conservation + Idempotence conservation + ...

What do we want to learn ?

Currently the ρ matrix at Relativistic Mean Field Level (without κ)

The problem is:

- In an H.O basis
- Axially symmetric $\rightarrow \Omega^{\pi}$ block diagonal.
- Hermiticity $\rightarrow$ Symmetric



Over 1000 β_2 configurations.

Tips, tricks and limitations

A few limitations:

1. GPU VRAM (From 4Gb to 40Gb)
2. Vanishing Gradients
3. Optimal metrics/cost function

Duplication Matrix :
$D_n \mathrm{vech}(A) = \mathrm{vec}(A)$

Elimination matrix:
$L_n \mathrm{vec}(A) = \mathrm{vech}(A)$

vech(A) being the half-vectorization of A Efficient (vetorizable) way to go from

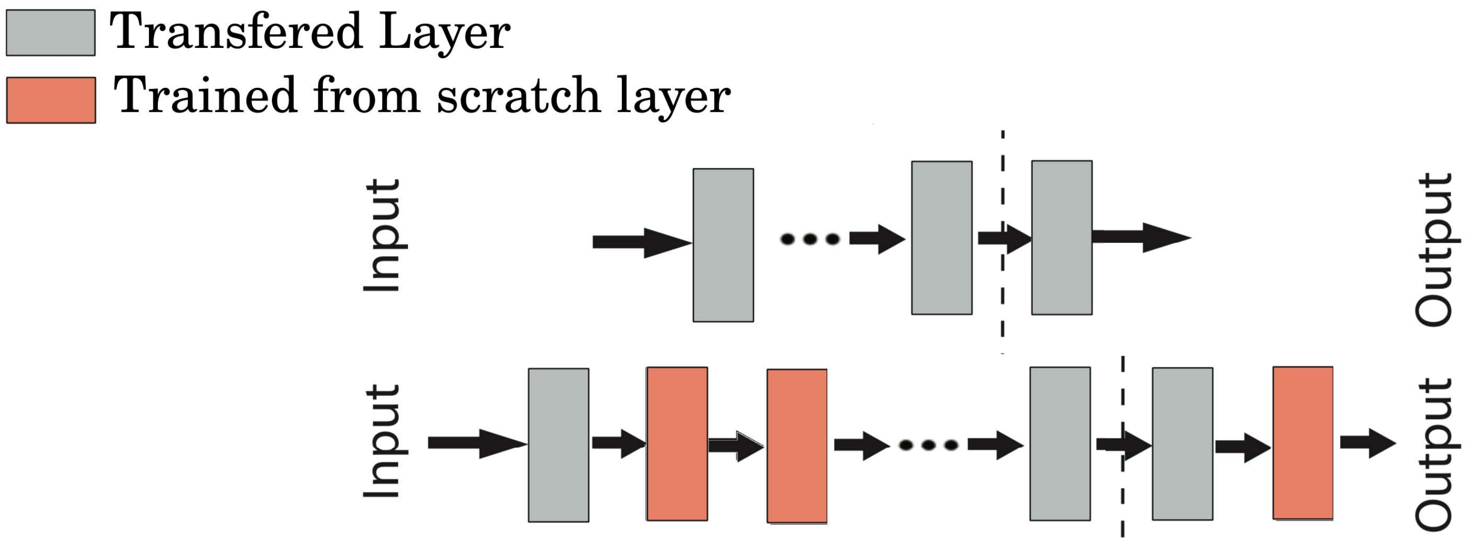
$$n^2 \rightarrow \frac{n(n+1)}{2}$$

Tips, tricks and limitations

A few limitations:

1. GPU VRAM (From 4Gb to 40Gb)
2. Vanishing Gradients
3. Optimal metrics/cost function

Transfer learning strategy ⇒ One order of magnitude decrease of the loss

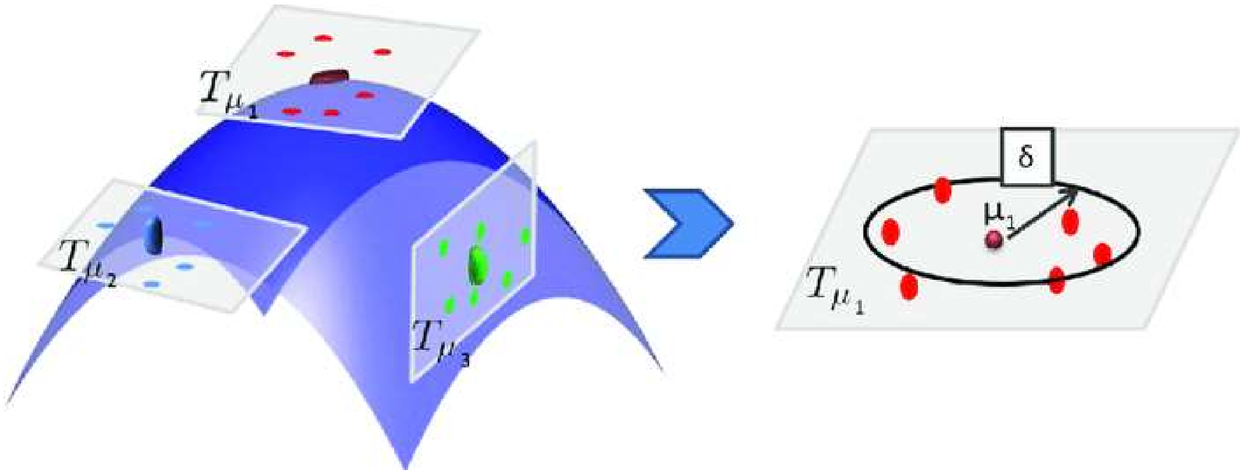


Tips, tricks and limitations

A few limitations:

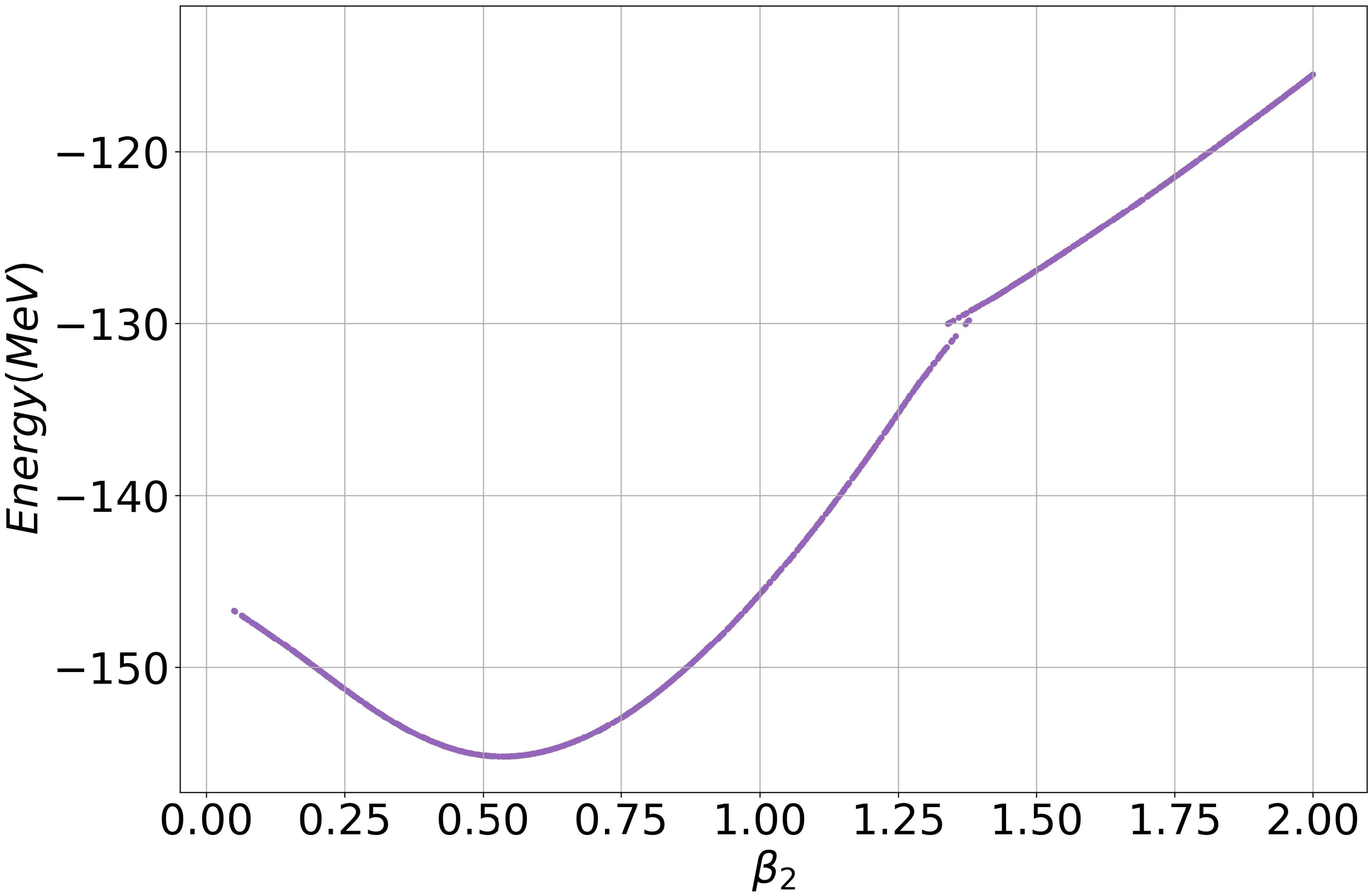
1. GPU VRAM (From 4Gb to 40Gb)
2. Vanishing Gradients
3. Optimal metrics/cost function

What is the accurate distance in between two Slater/ρ matrices ?



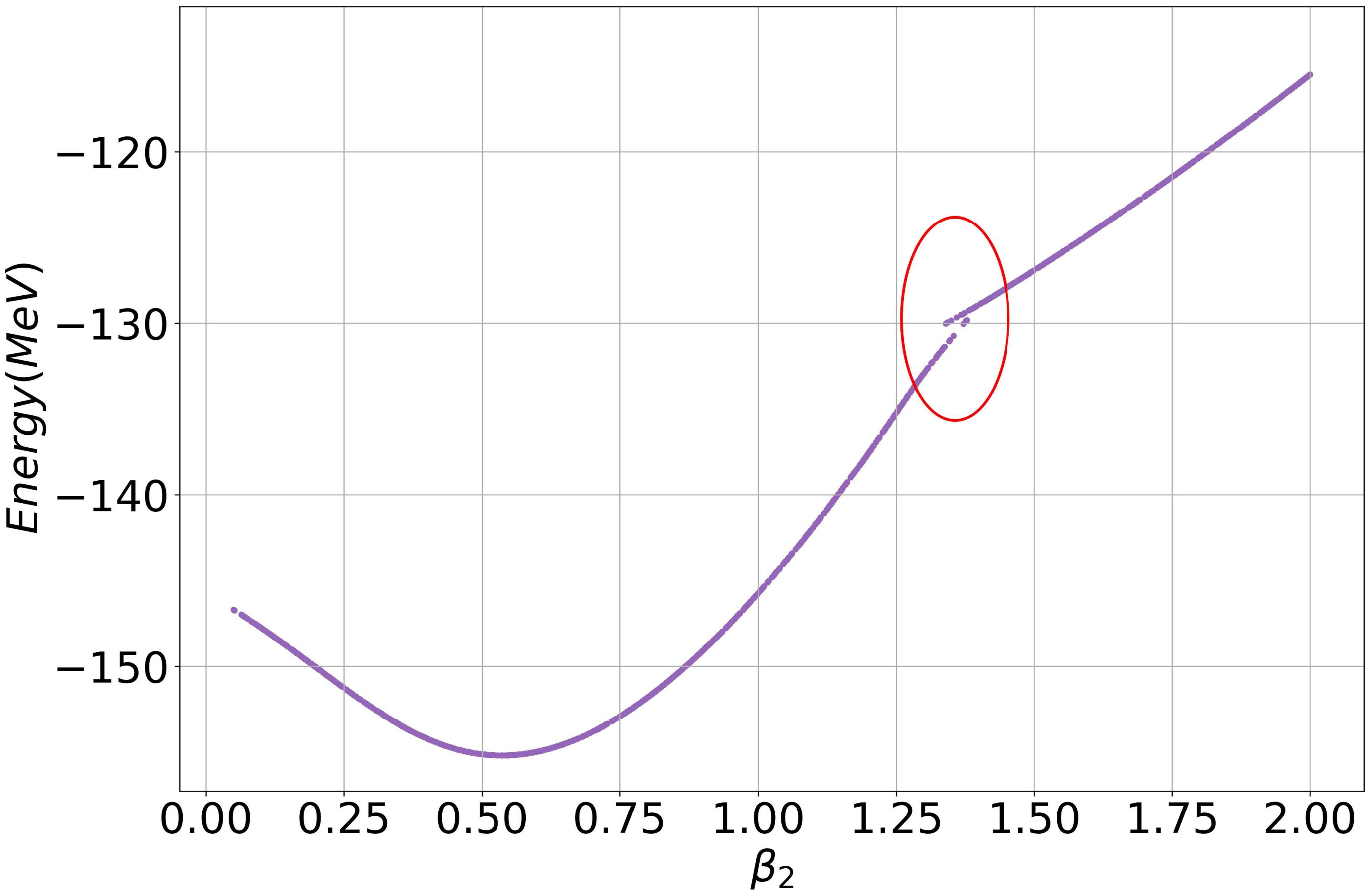
^{20}Ne: Some preliminary results

- 8 Shell
- DDME-2 Functional



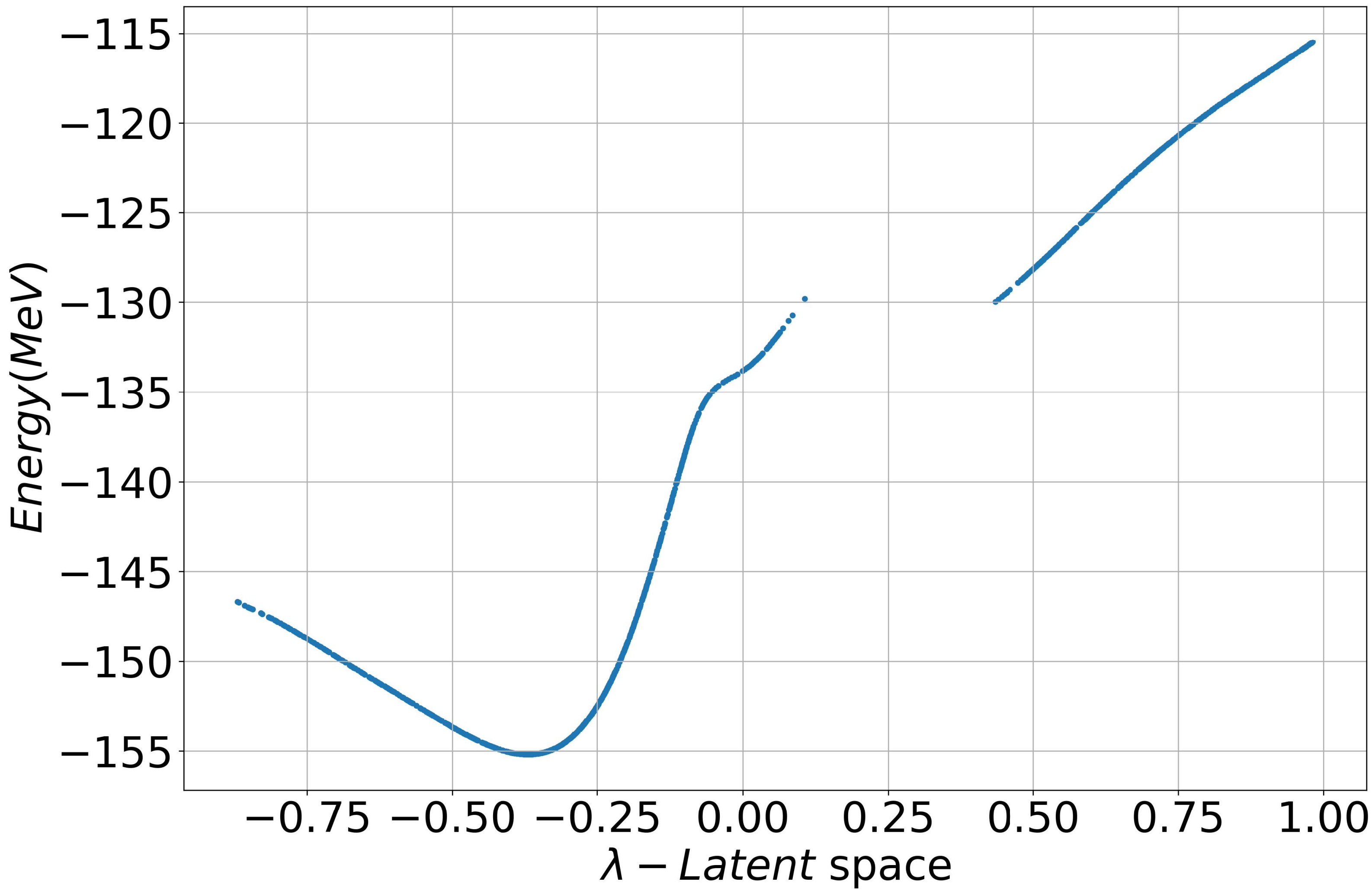
^{20}Ne: Some preliminary results

Discontinuity in the PES

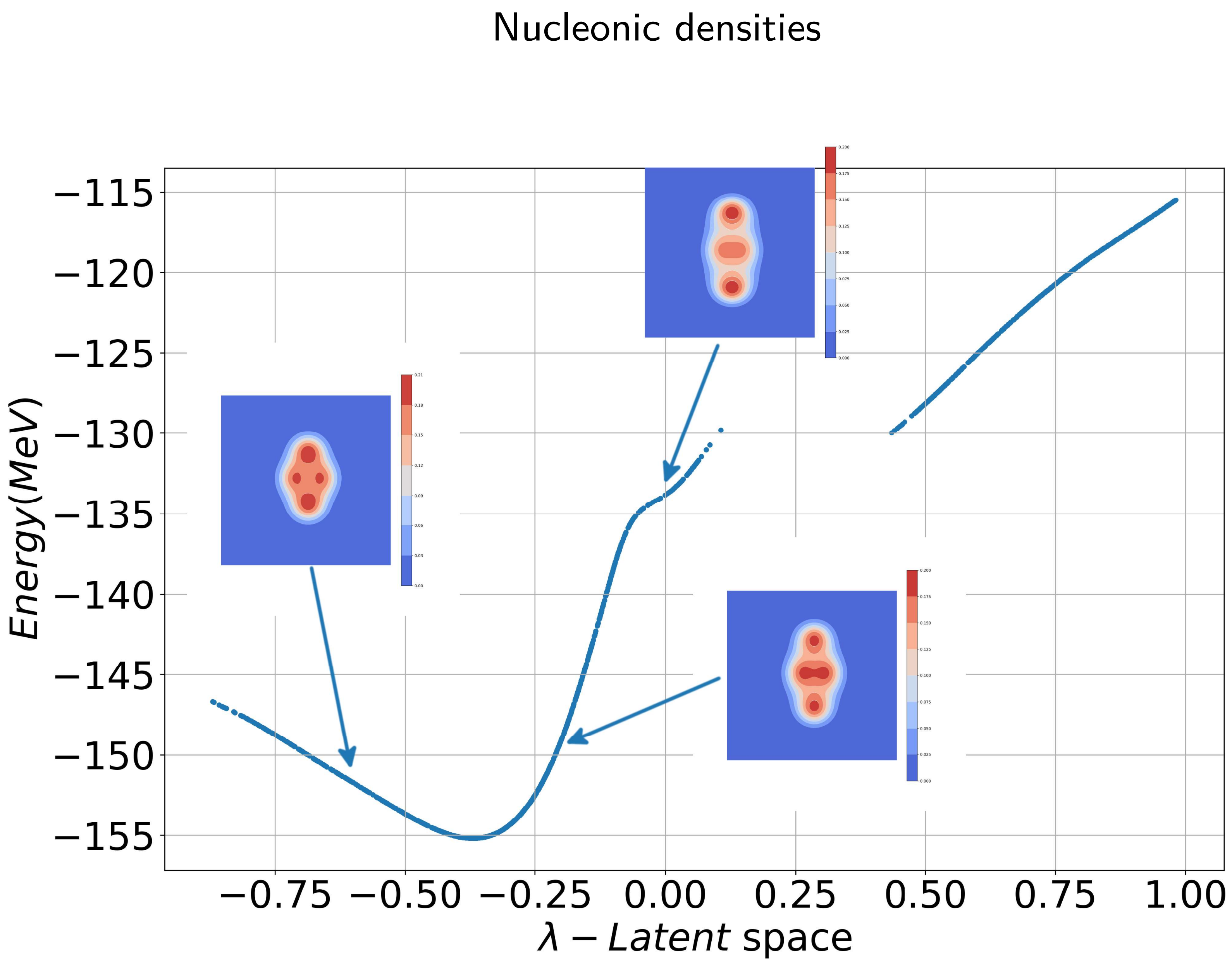


^{20}Ne: Some preliminary results

Clear clustering in the latent space

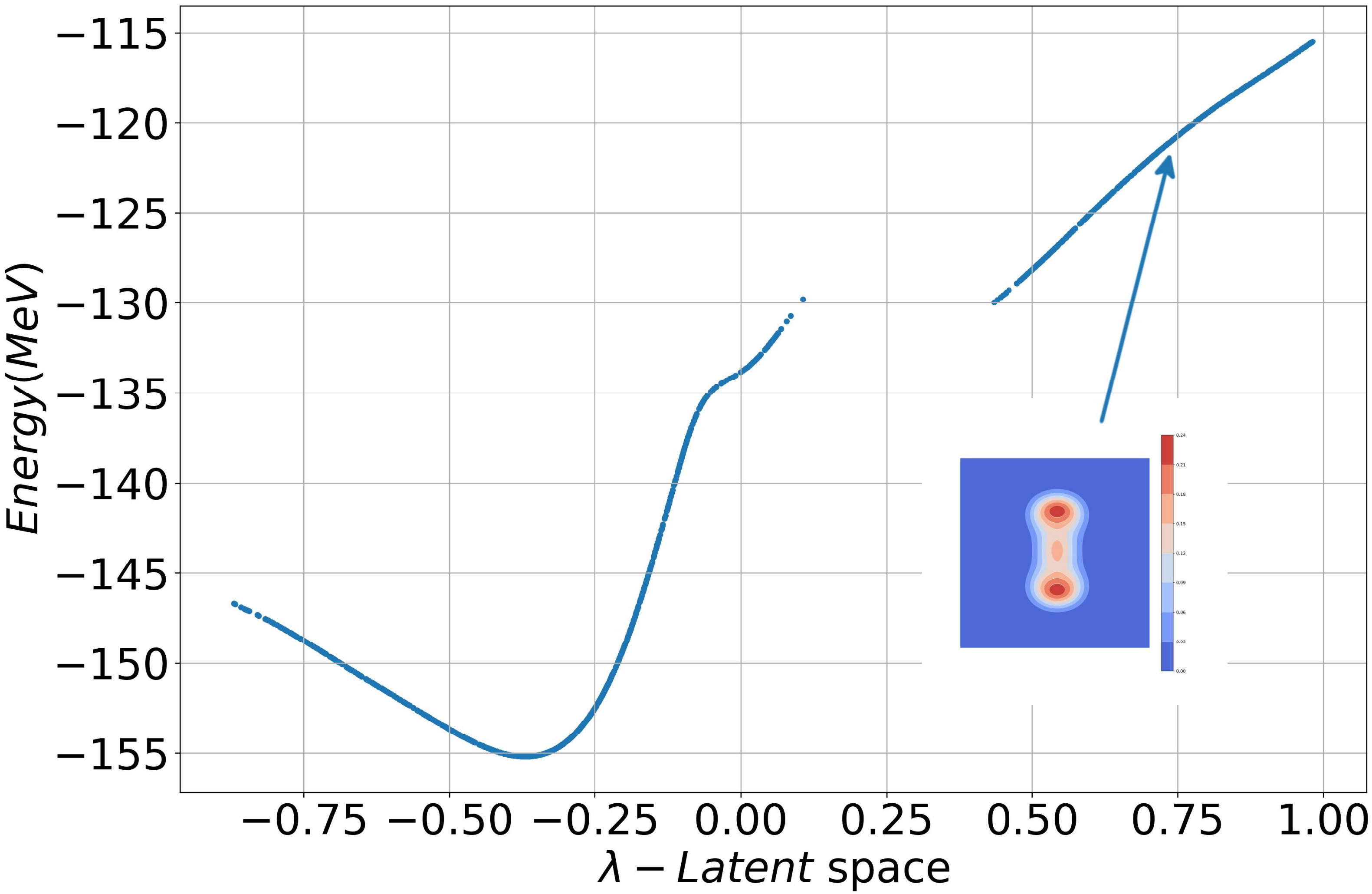


^{20}Ne: Some preliminary results



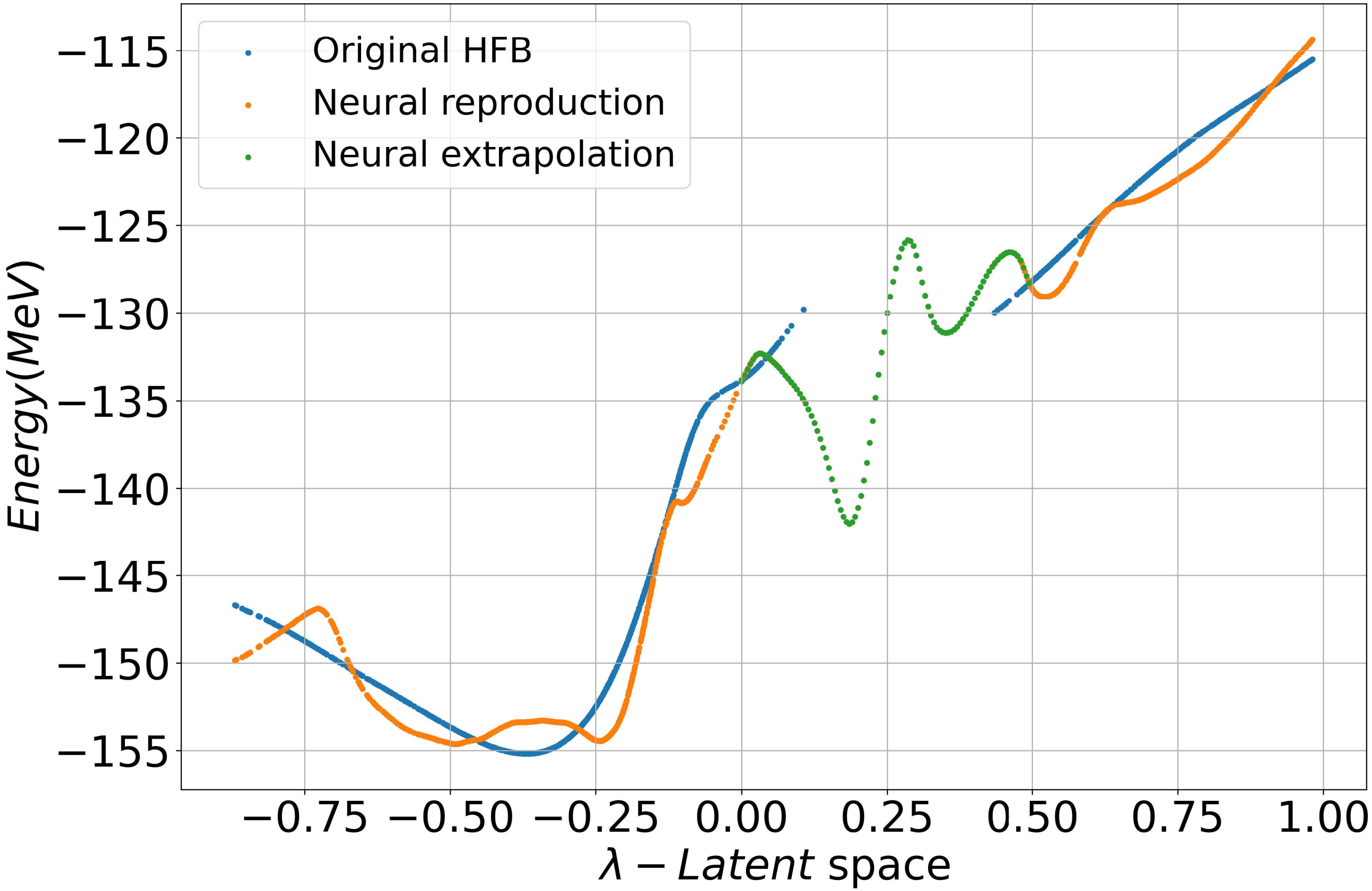
^{20}Ne: Some preliminary results

Nucleonic densities



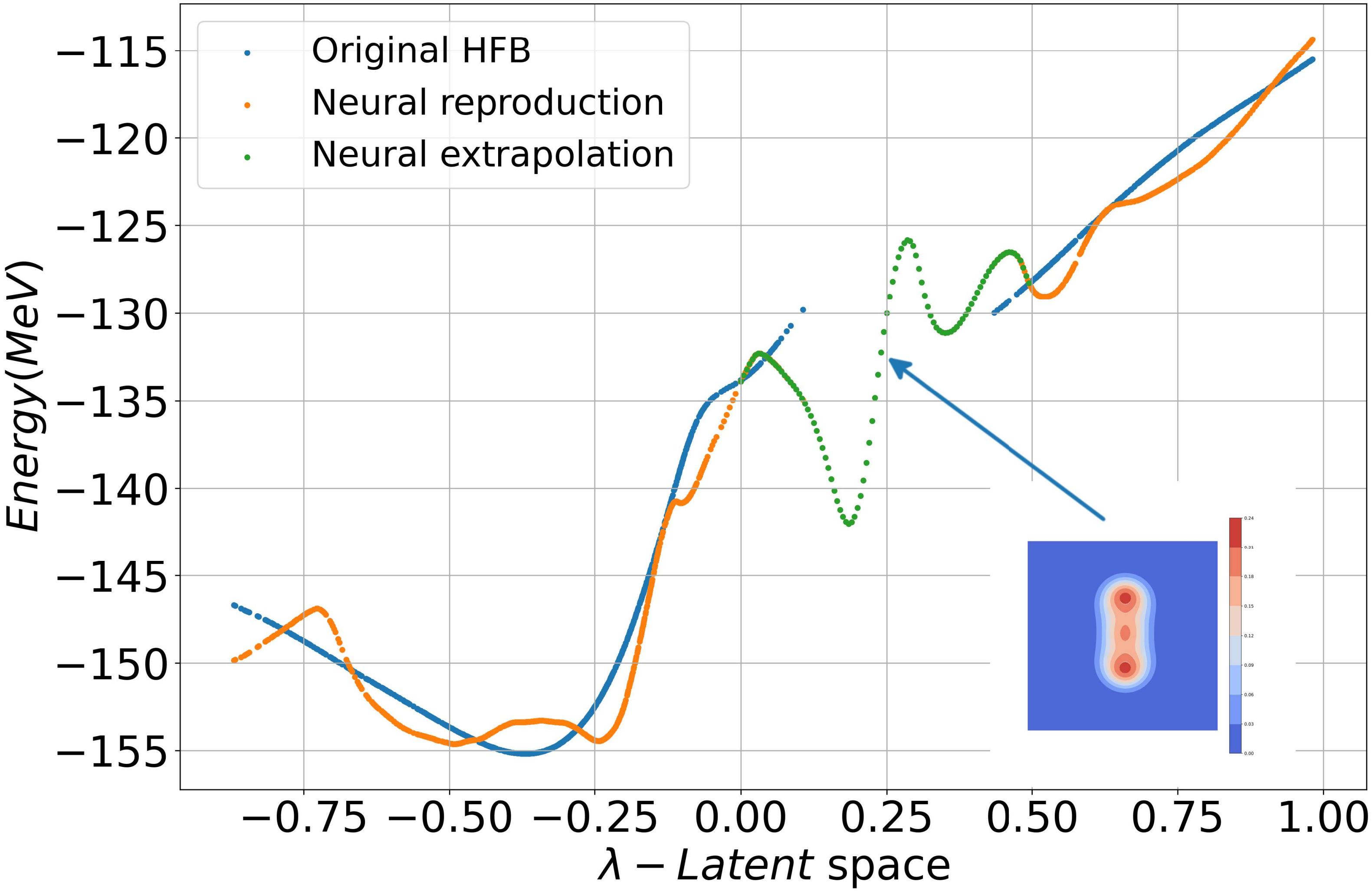
^{20}Ne: Some preliminary results

Extrapolation along the latent space



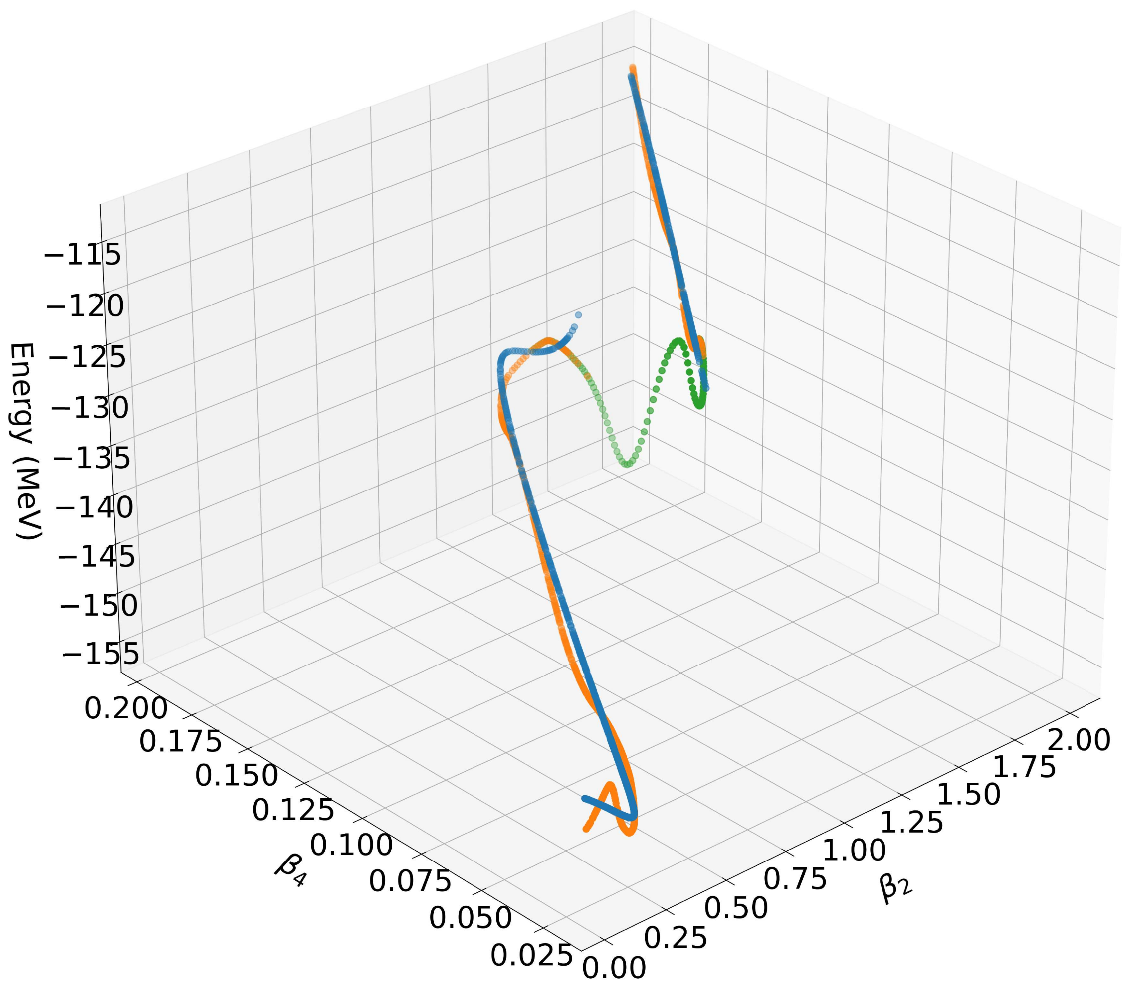
^{20}Ne: Some preliminary results

Extraplation along the latent space + Nucleonic density



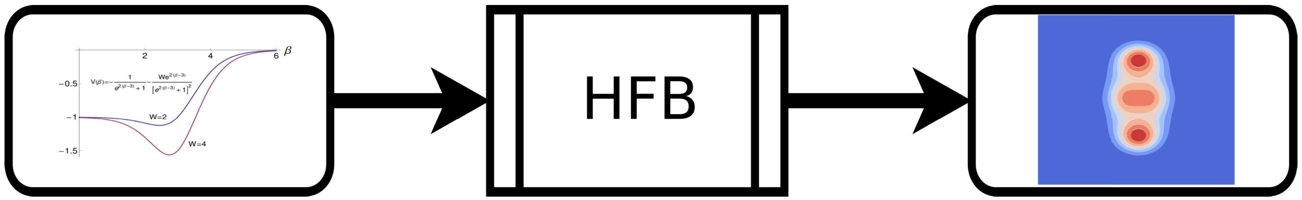
^{20}Ne: Some preliminary results

Hexadecapolar discontinuity



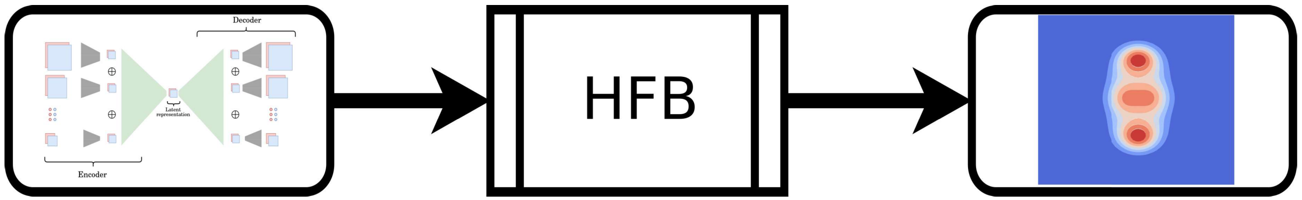
Possibles uses and roadmap

1. A good *prior*/starting point for HFB solvers
2. A direct generative approach for HFB states
3. A possible way to overcome discontinuities ?



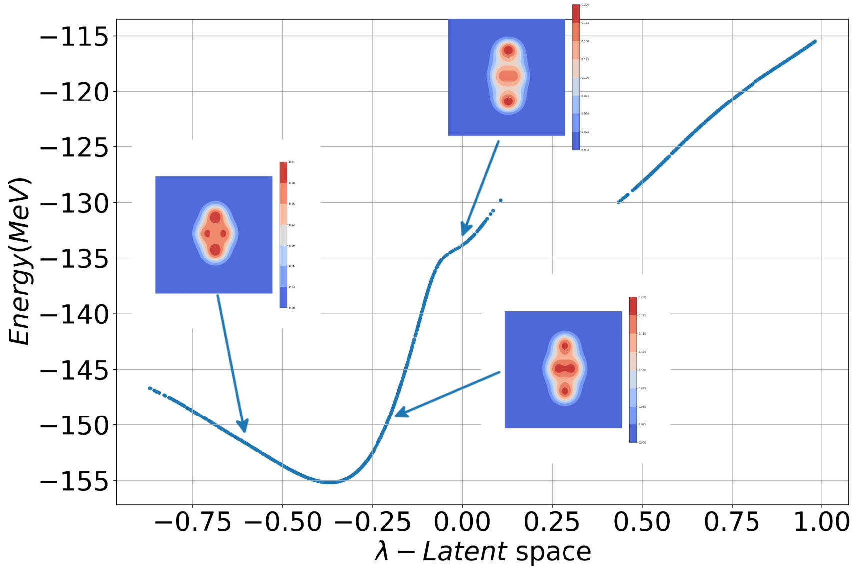
Possibles uses and roadmap

1. A good *prior*/starting point for HFB solvers
2. A direct generative approach for HFB states
3. A possible way to overcome discontinuities ?



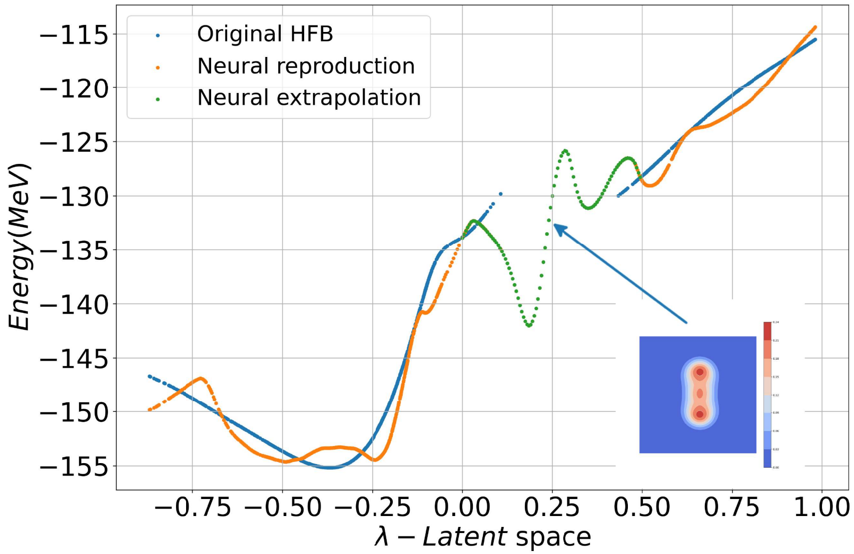
Possibles uses and roadmap

1. A good *prior*/starting point for HFB solvers
2. A direct generative approach for HFB states
3. A possible way to overcome discontinuities ?



Possibles uses and roadmap

1. A good *prior*/starting point for HFB solvers
2. A direct generative approach for HFB states
3. A possible way to overcome discontinuities ?



Possibles uses and roadmap

1. A good *prior*/starting point for HFB solvers
2. A direct generative approach for HFB states
3. A possible way to overcome discontinuities ?

Currently struggling to reproduce approximately-enough"correct" HF states

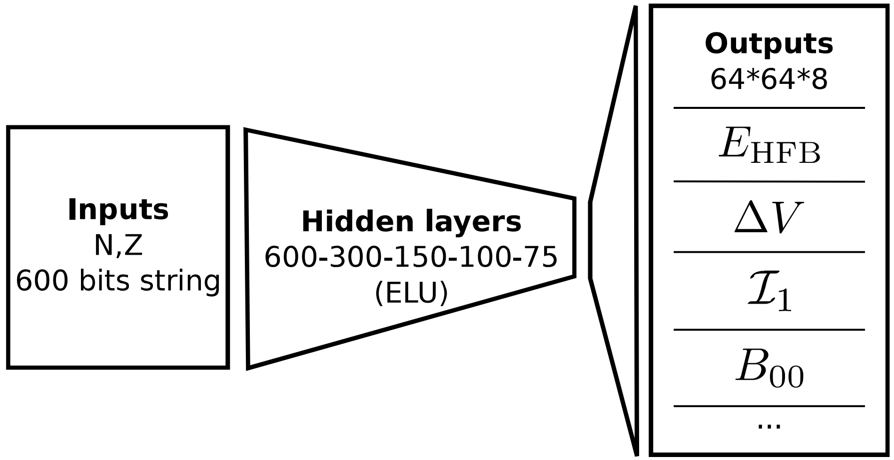
$\Downarrow$

Several open questions yet

Conclusions and outlooks

Machine Learning for:

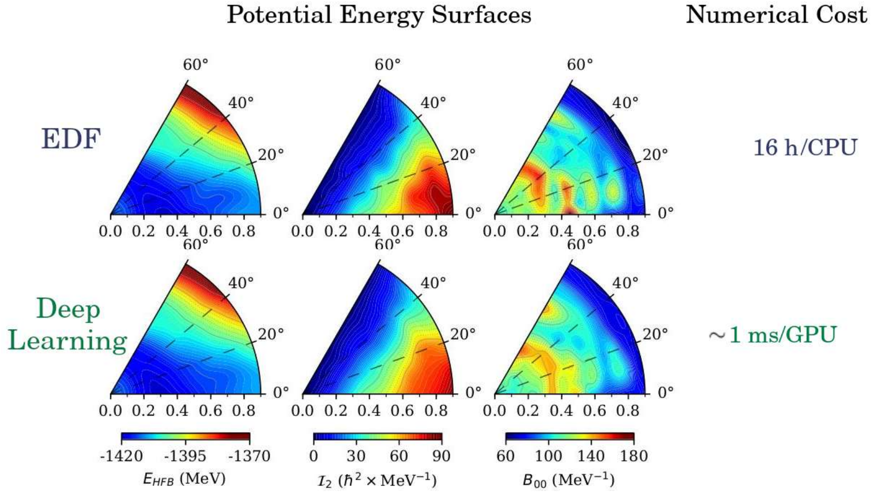
- Modelisation of collective variables
- Generation of a manifold of HFB states



Conclusions and outlooks

Machine Learning for:

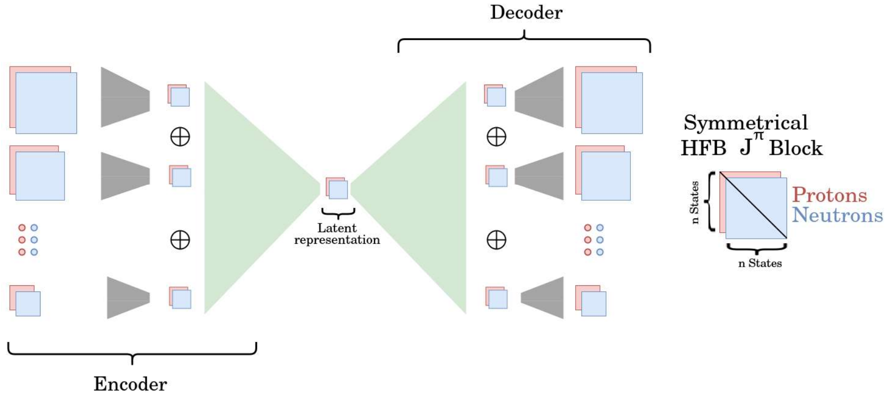
- Modelisation of collective variables
- Generation of a manifold of HFB states



Conclusions and outlooks

Machine Learning for:

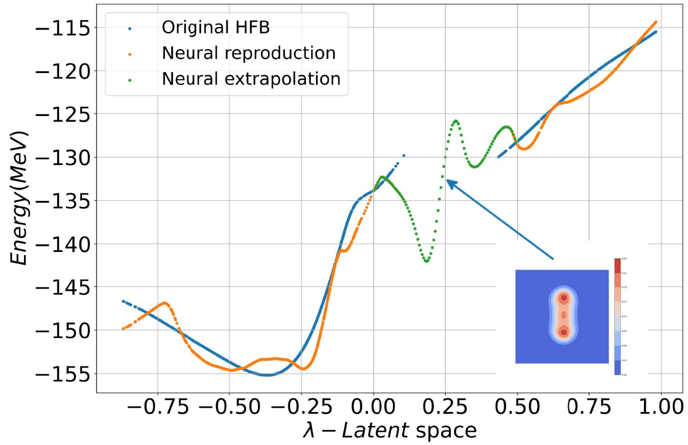
- Modelisation of collective variables
- Generation of a manifold of HFB states



Conclusions and outlooks

Machine Learning for:

- Modelisation of collective variables
- Generation of a manifold of HFB states



The **NucleAI** project

Collaborators:

- G. Hupin, D. Lacroix, IJCLab
- D. Regnier, J-P. Ebran, S. Hilaire, CEA, DAM
- S. Goriely, ULB
- J. Margueron, IPNL
- A. Penon, J. Ripoche, Magic Lemp

Support:

- NVIDIA GPU Grant Program: 2× Titan V GPU

Thank you for your attention !

The **NucleAI** project

Collaborators:

- G. Hupin, D. Lacroix, IJCLab
- D. Regnier, J-P. Ebran, S. Hilaire, CEA, DAM
- S. Goriely, ULB
- J. Margueron, IPNL
- A. Penon, J. Ripoche, Magic Lemp

Support:

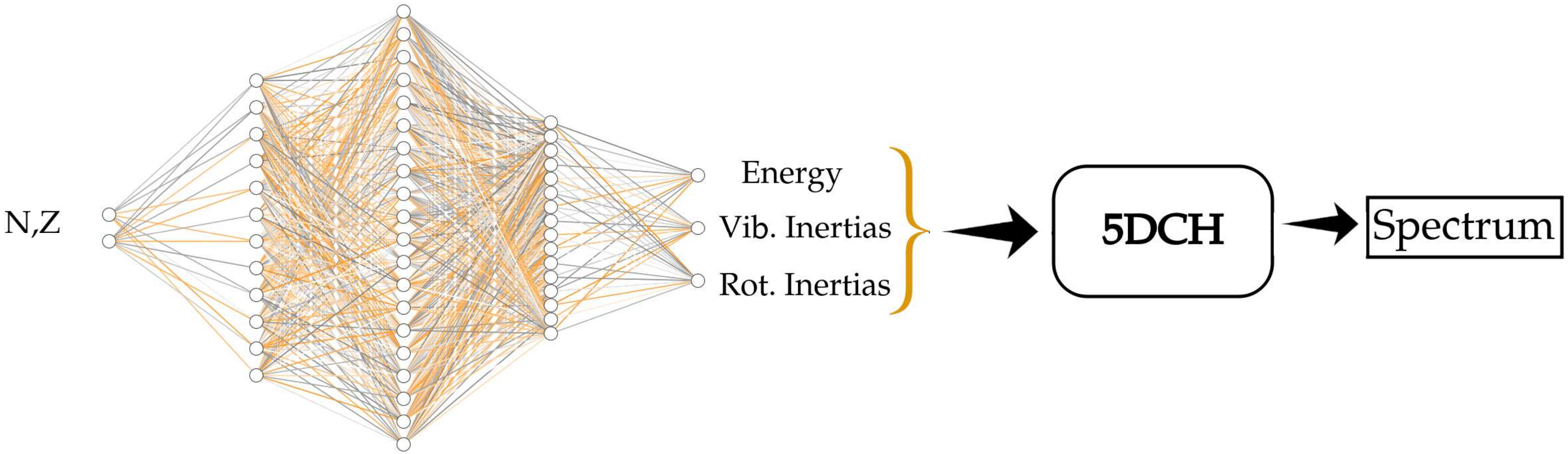
- NVIDIA GPU Grant Program: 2× Titan V GPU

Thank you for your attention !

Replacing the time consuming part by a neural network

[noframenumbering]

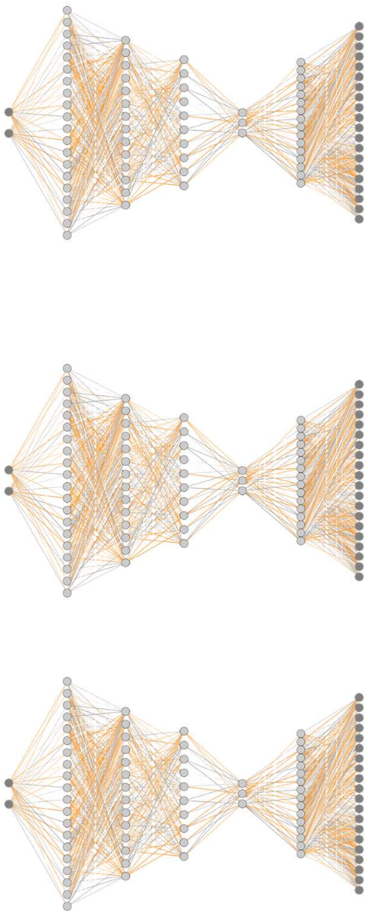
Multi task Learning



- Learning the correlations between the 8 outputs

Using a committee of neural networks

Committee "Query" Candidate

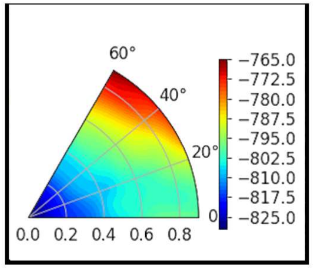


825

821

828

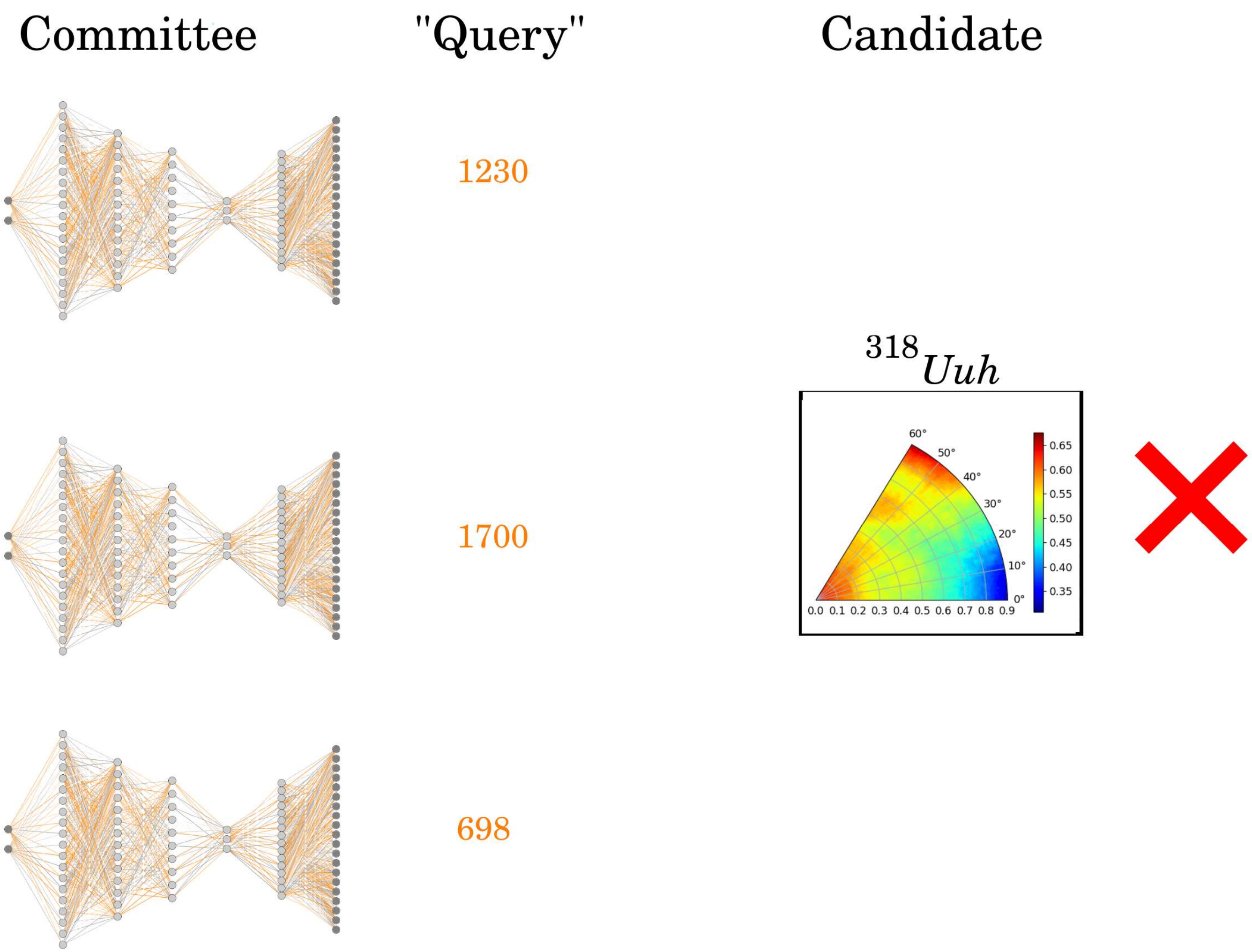
^{100}Sn



Benefits of a committee

- Less sensitive to the random initialization
- Estimation of the associated standard deviation

Using a committee of neural networks

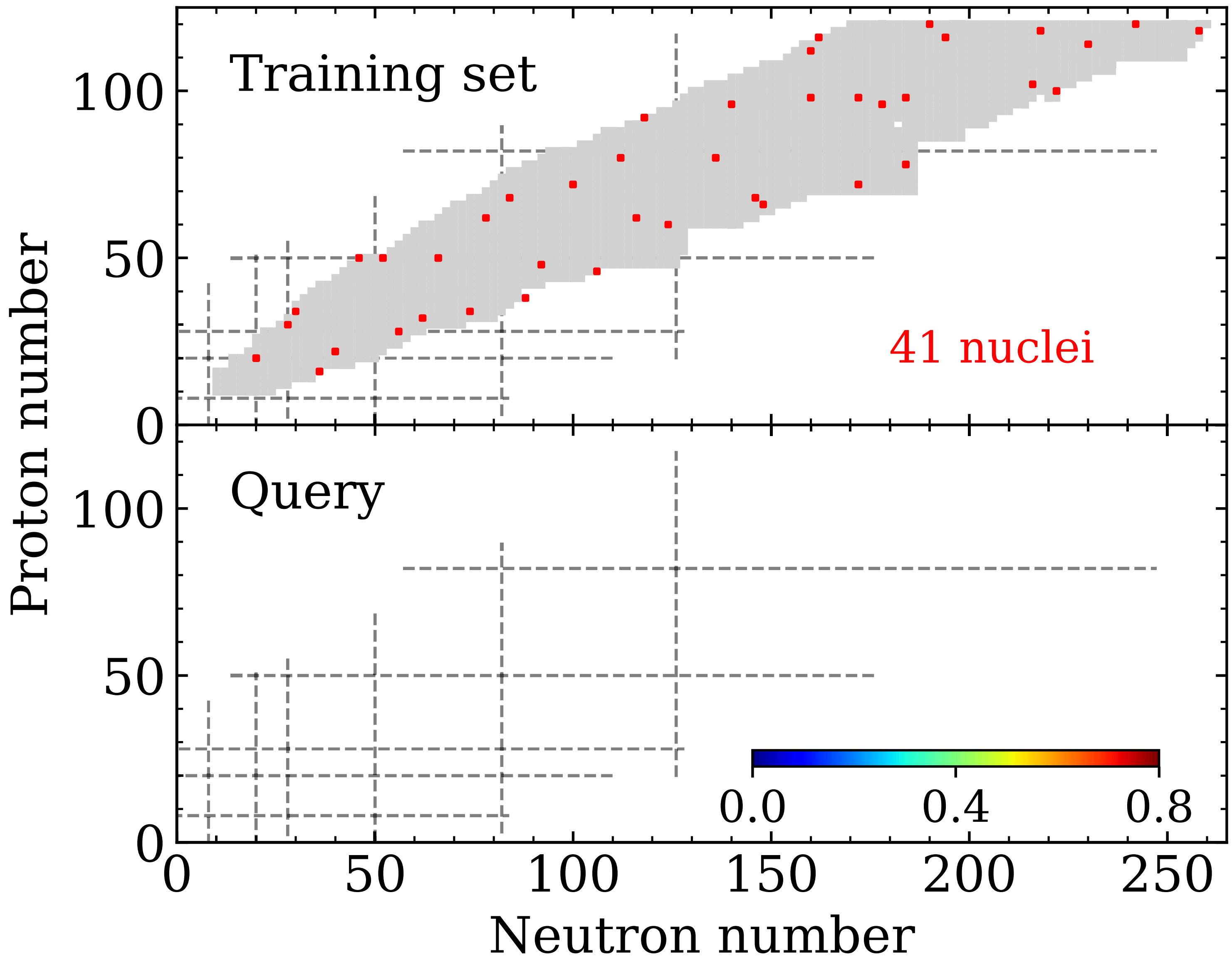


Benefits of a committee

- Less sensitive to the random initialization
- Estimation of the associated standard deviation

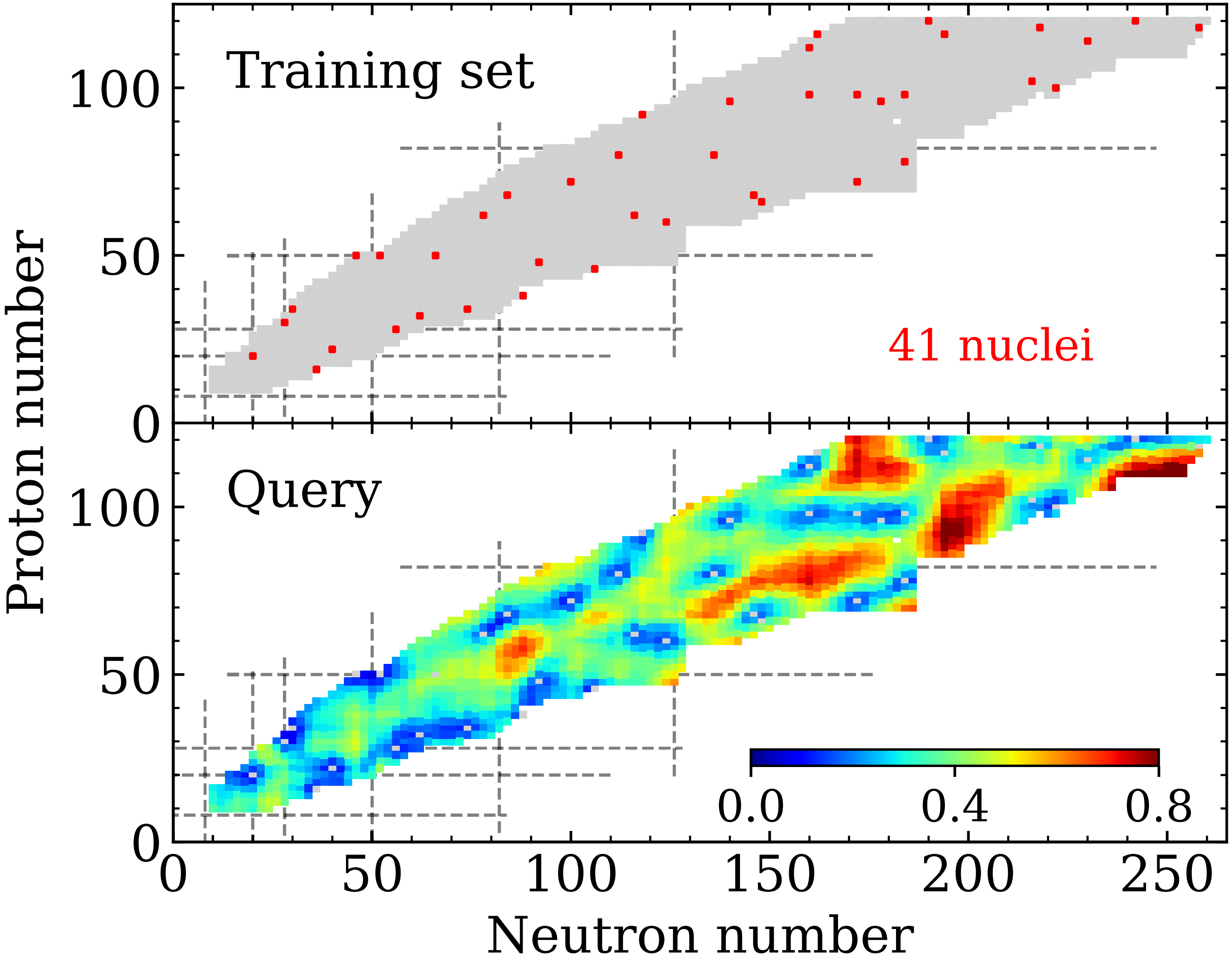
Active learning

- An incremental and automatic choice of training nuclei (5 nuclei/step)
- Query $\simeq$ standard deviation between the committee members



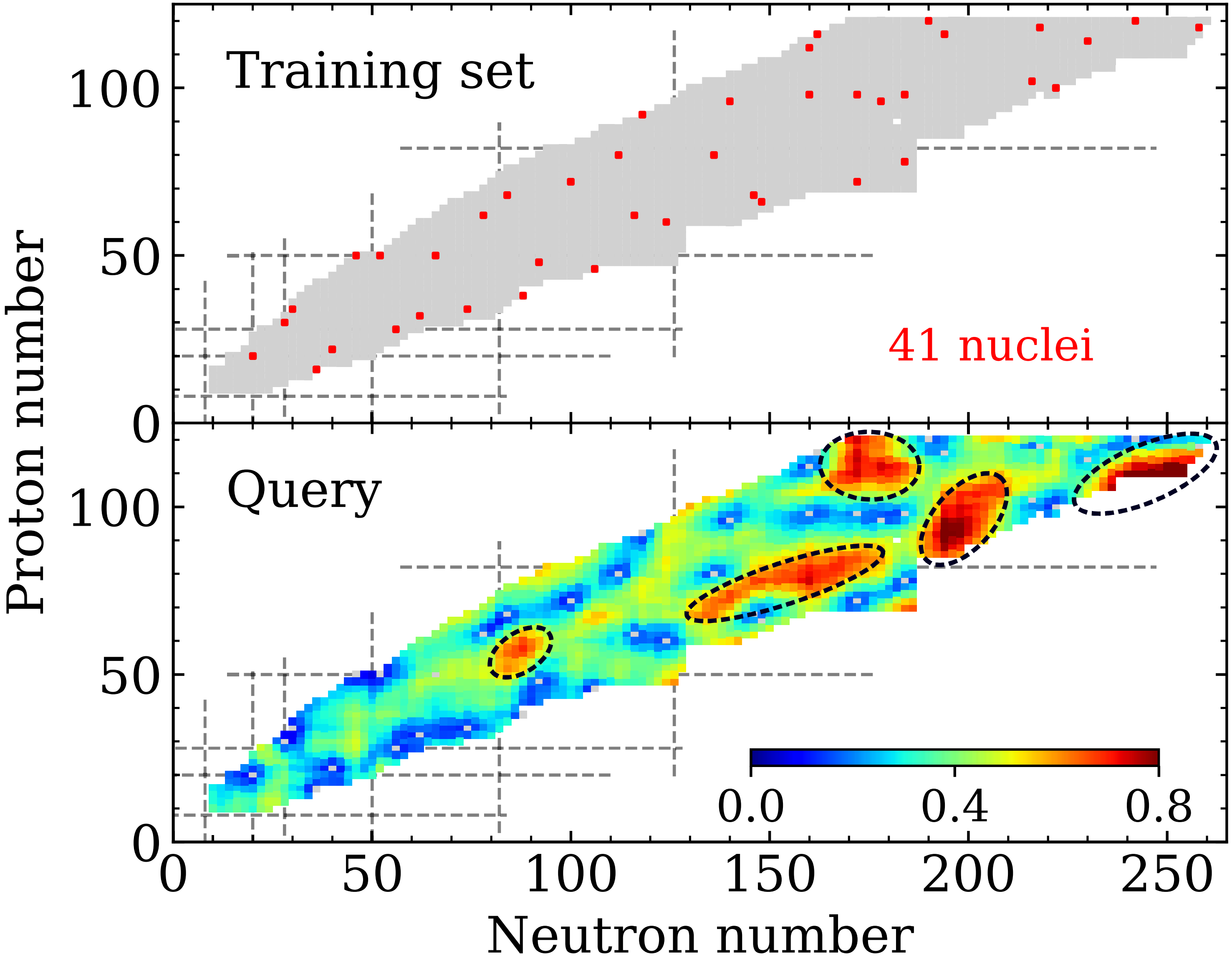
Active learning

- An incremental and automatic choice of training nuclei (5 nuclei/step)
- Query $\simeq$ standard deviation between the committee members



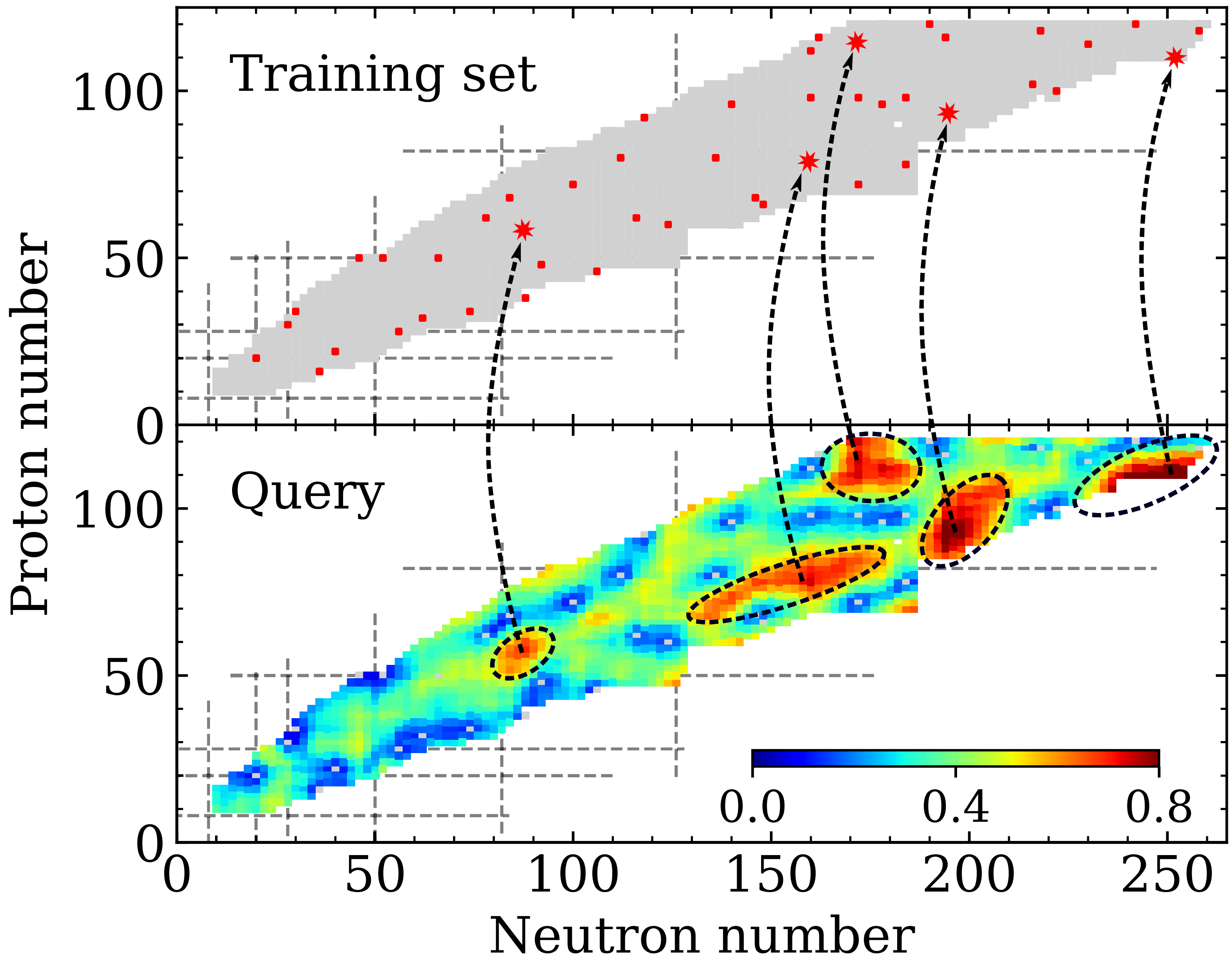
Active learning

- An incremental and automatic choice of training nuclei (5 nuclei/step)
- Query $\simeq$ standard deviation between the committee members



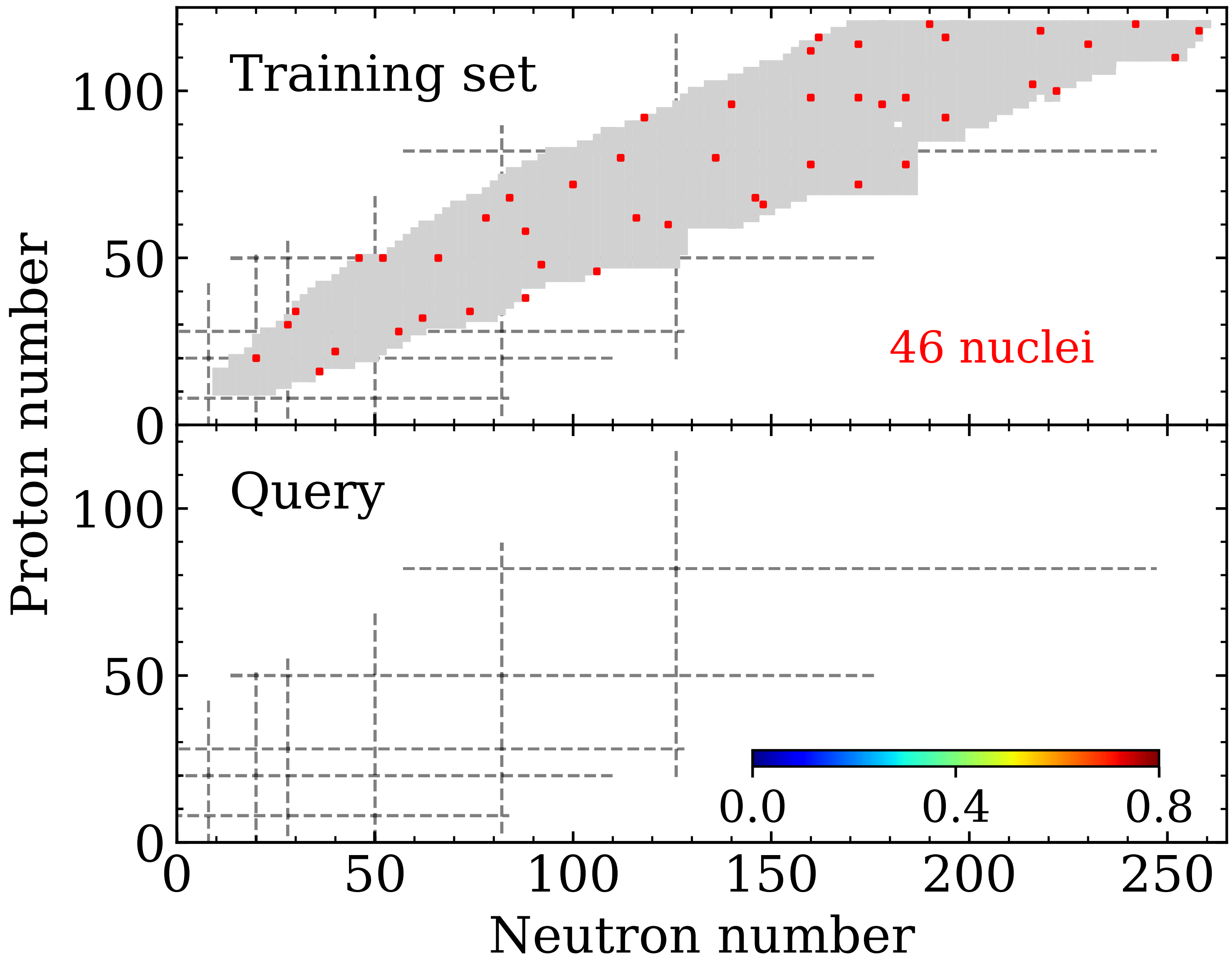
Active learning

- An incremental and automatic choice of training nuclei (5 nuclei/step)
- Query $\simeq$ standard deviation between the committee members



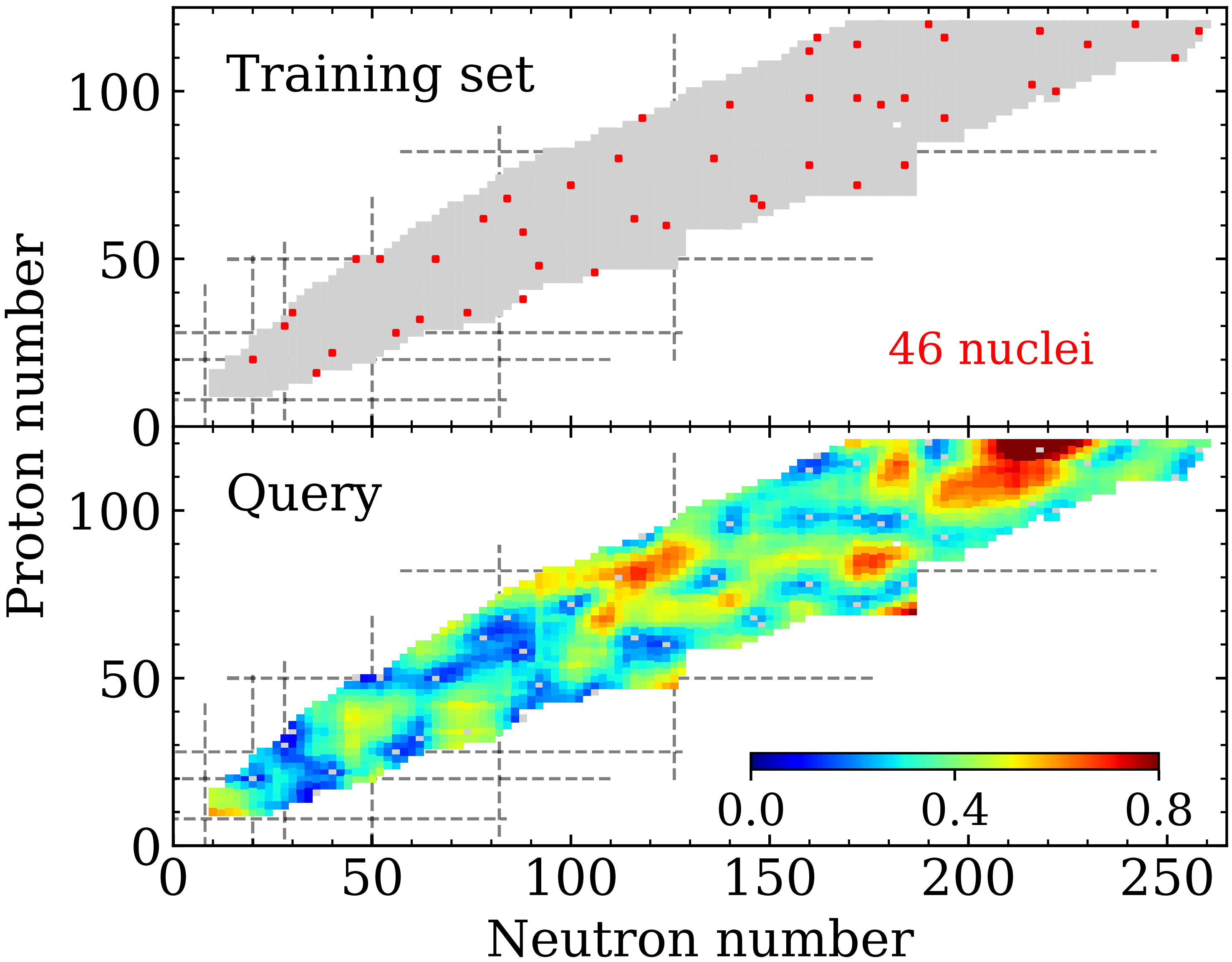
Active learning

- An incremental and automatic choice of training nuclei (5 nuclei/step)
- Query $\simeq$ standard deviation between the committee members



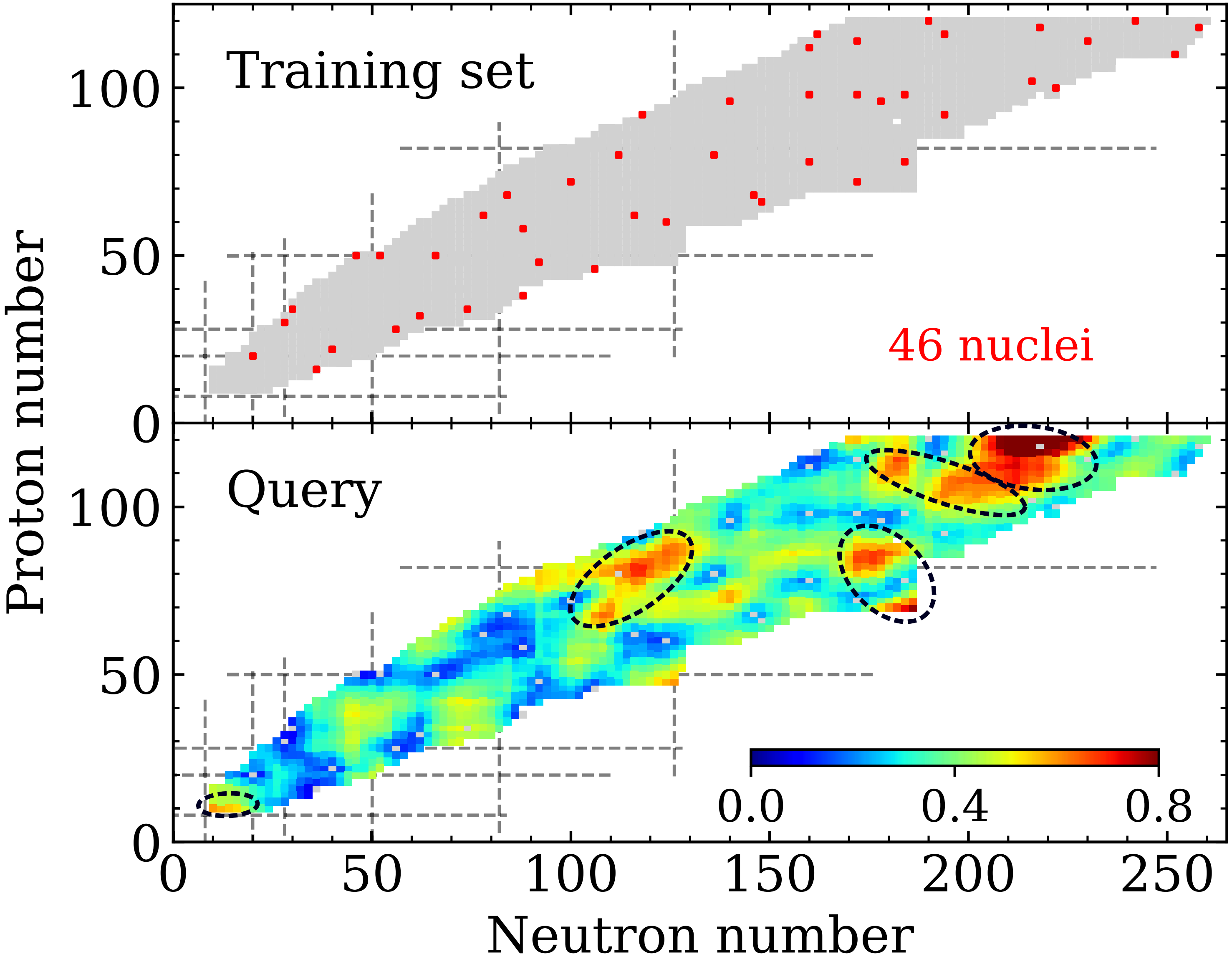
Active learning

- An incremental and automatic choice of training nuclei (5 nuclei/step)
- Query $\simeq$ standard deviation between the committee members



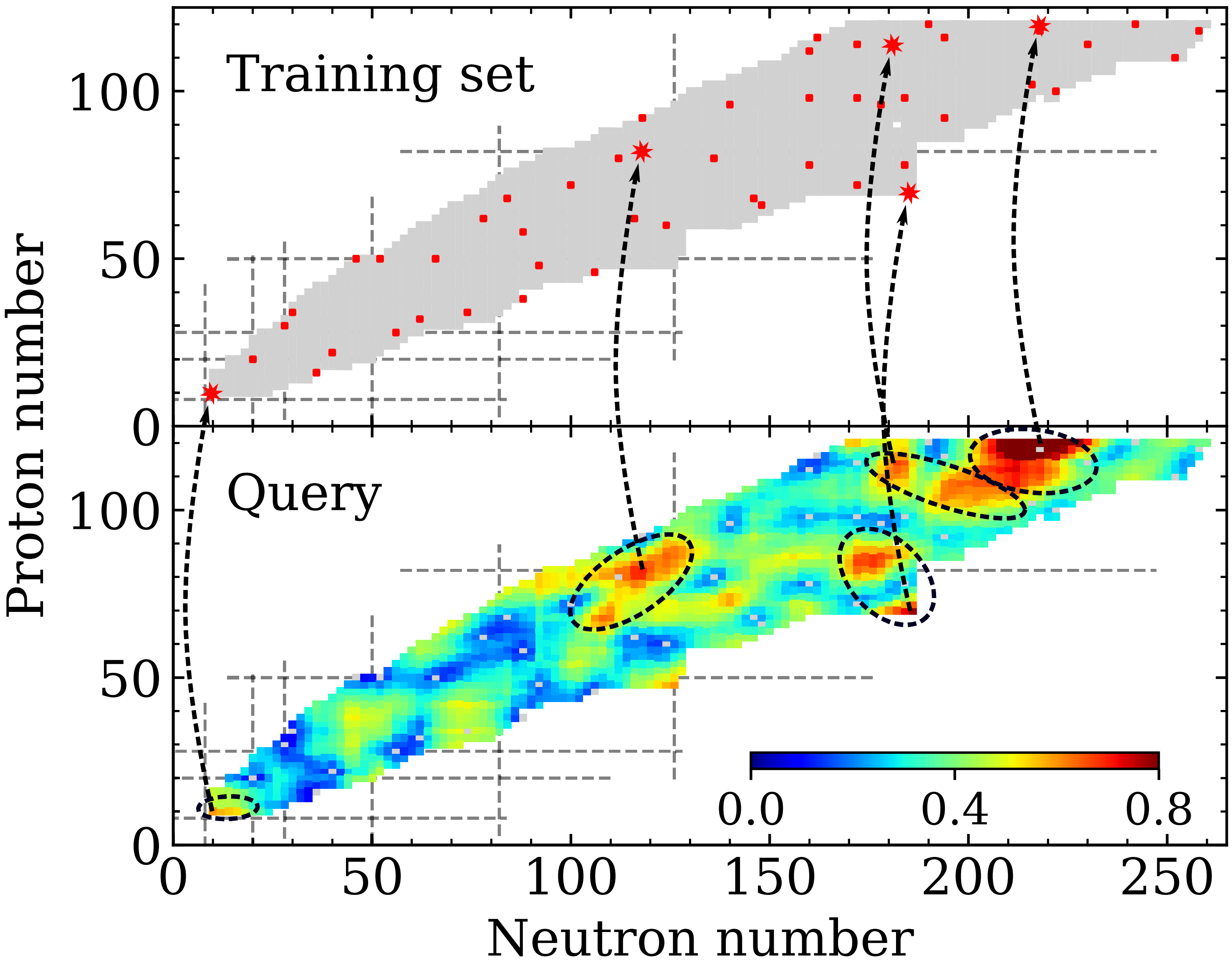
Active learning

- An incremental and automatic choice of training nuclei (5 nuclei/step)
- Query $\simeq$ standard deviation between the committee members



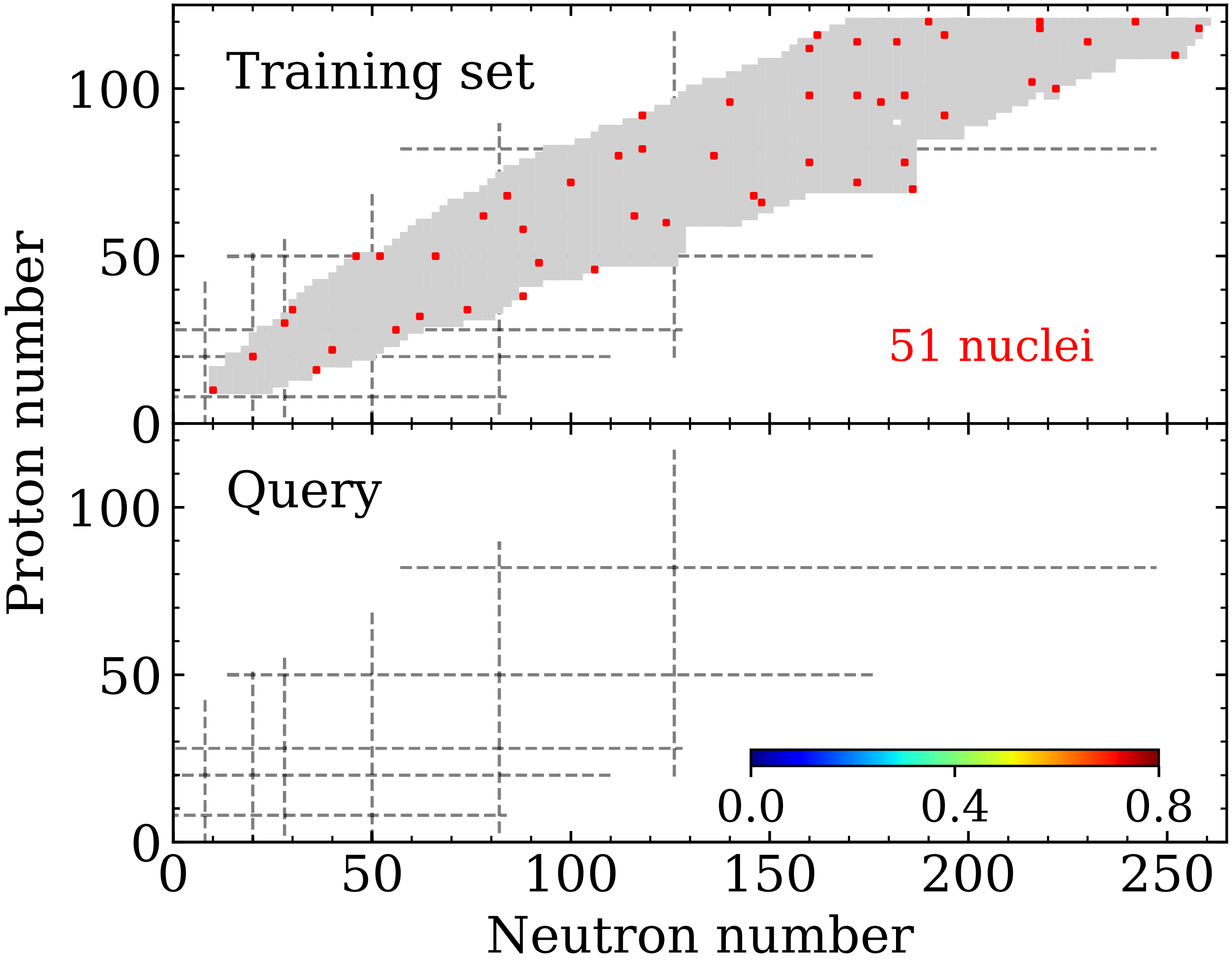
Active learning

- An incremental and automatic choice of training nuclei (5 nuclei/step)
- Query $\simeq$ standard deviation between the committee members



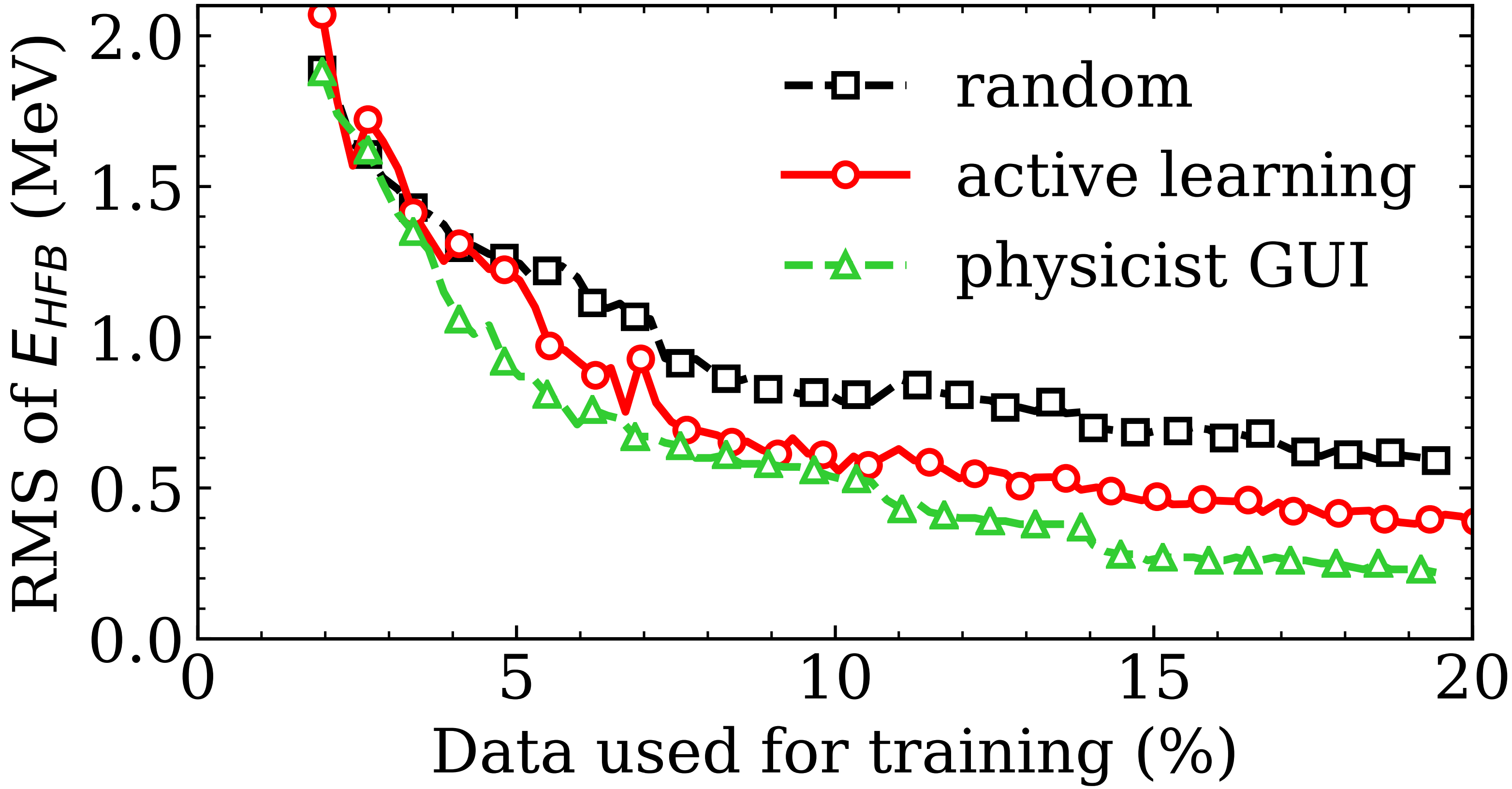
Active learning

- An incremental and automatic choice of training nuclei (5 nuclei/step)
- Query $\simeq$ standard deviation between the committee members



Root Mean Square error (RMS) of the potential energy surface

Test RMS $= \sqrt{(AI - HFB)^2}$ on the nuclei not in the training set



Root Mean Square error (RMS) of all outputs

AI versus HFB:

Train %	E_{HFB} (keV)	ΔV	$\mathcal{I}_1$ ($\hbar^2 \times \mathrm{MeV}^{-1}$)	$\mathcal{I}_2$	$\mathcal{I}_3$	$\boldsymbol{B}_{00}$ (MeV^{-1})	$\boldsymbol{B}_{01}$	$\boldsymbol{B}_{11}$	E_{GS} (keV)
5	1190	417	1.84	2.80	0.97	13.8	12.0	28.2	1325
10	**557**	**312**	**1.40**	**2.25**	**0.76**	**11.7**	**10.2**	**23.9**	**716**
15	471	247	1.25	2.02	0.69	10.6	9.4	21.9	655
20	388	202	1.22	1.96	0.68	10.2	9.1	21.2	518

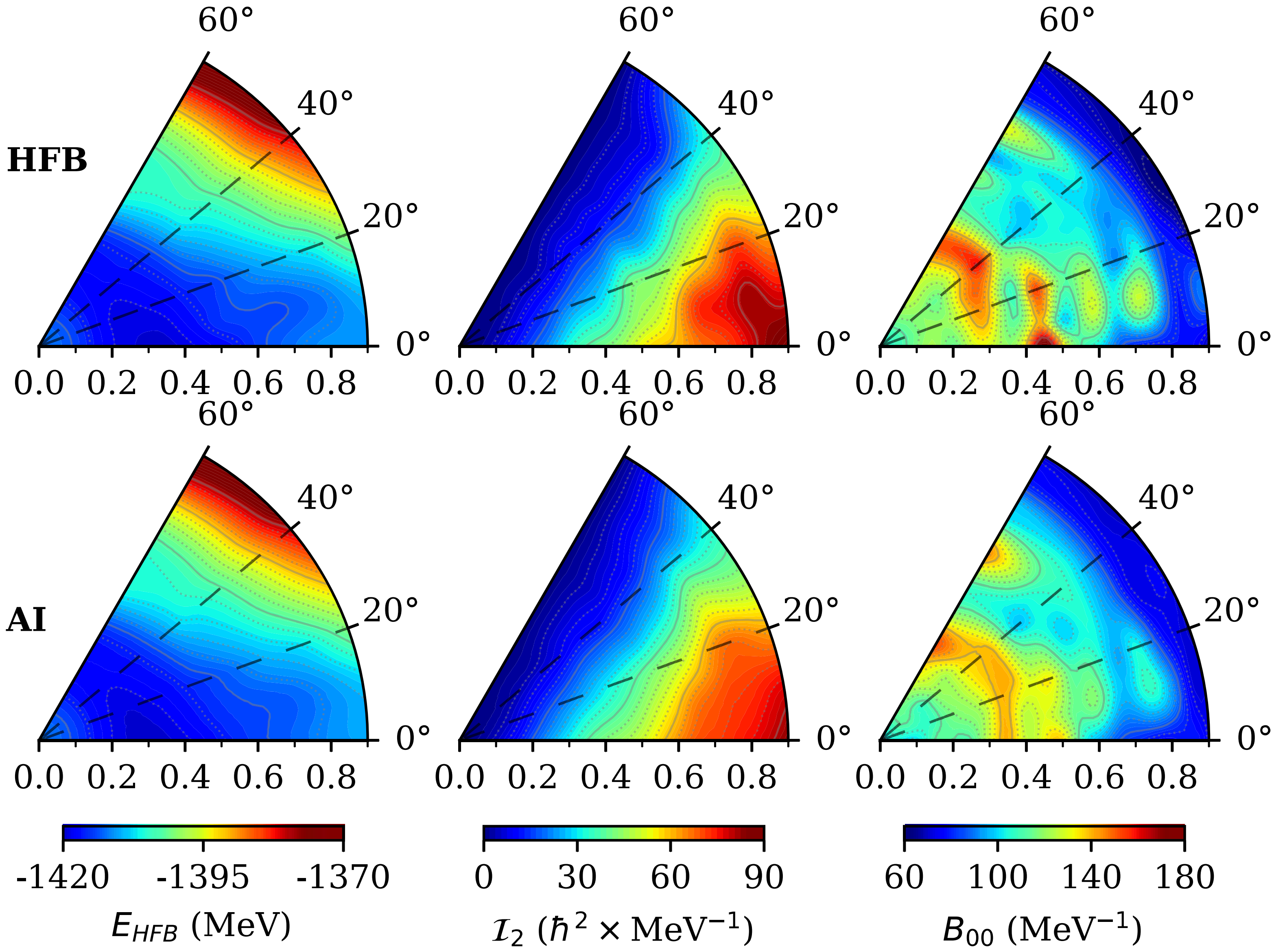
The first column contains the size of the training set in % of the AMEDEE database while the others highlight the RMS of the outputs of the AI. The last column contains the RMS associated to the correlated ground state energy E_{GS}.

Keep in mind:

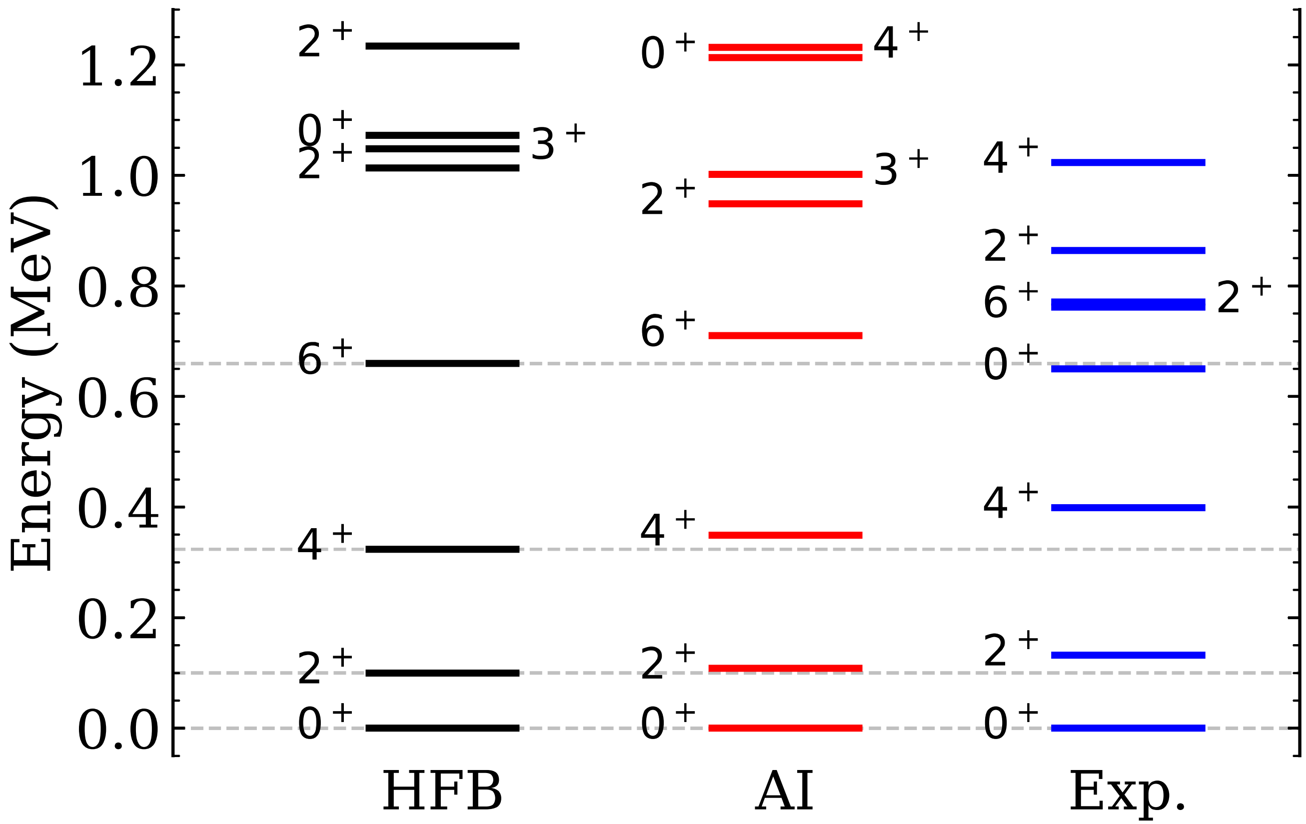
- RMS= 950 keV, AI vs Exp: Athanassopoulos *et. al* (2004), fitted on 1800 nuclei
- RMS= 790 keV: 5DCH Gogny D1M vs Exp.

Example of ^{178}Os

- RMS(E_{HFB}) $\simeq$ median RMS on the 1800 test nuclei
- Closest trained nucleus: +4 neutrons, -2 protons



Excitation spectrum of ^{178}Os



- Correlated ground state: $|E_{GS}^{AI} - E_{GS}^{HFB}| = 150$ keV
- Rotational states reproduced within 8%
- First vibrational state within 13%