

Using neural networks to extrapolate the results of *ab initio* No-Core Shell Model computations

Bachelor thesis by Marc Leon Agel
Date of submission: October 25, 2021

1. Review: Prof. Dr. Robert Roth
2. Review: M.Sc. Marco Knöll
Darmstadt



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Physics Department
Institut für Kernphysik
Theoretical Nuclear Physics
++

Erklärung zur Abschlussarbeit gemäß §22 Abs. 7 und §23 Abs. 7 APB der TU Darmstadt

Hiermit versichere ich, Marc Leon Agel, die vorliegende Bachelorarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mir ist bekannt, dass im Fall eines Plagiats (§38 Abs. 2 APB) ein Täuschungsversuch vorliegt, der dazu führt, dass die Arbeit mit 5,0 bewertet und damit ein Prüfungsversuch verbraucht wird. Abschlussarbeiten dürfen nur einmal wiederholt werden.

Bei der abgegebenen Thesis stimmen die schriftliche und die zur Archivierung eingereichte elektronische Fassung gemäß §23 Abs. 7 APB überein.

Bei einer Thesis des Fachbereichs Architektur entspricht die eingereichte elektronische Fassung dem vorgestellten Modell und den vorgelegten Plänen.

Darmstadt, 25. Oktober 2021

Marc Leon Agel

Marc Leon Agel



Danksagung

Vielen Dank an Prof. Robert Roth, Marco Knöll und Tobias Wolfgruber für die Unterstützung und Betreuung meiner Bachelorarbeit.

Desweiteren möchte ich mich herzlich bei Bastian Arnold für das Korrekturlesen und für viele hilfreiche Vorschläge zur Verbesserung der Texte bedanken.

Contents

1	Introduction	6
2	Theoretical Foundations	7
2.1	The <i>ab initio</i> No-Core Shell Model formalism	7
2.1.1	Similarity Renormalization Group	8
2.2	Machine Learning	9
2.2.1	Introduction to Artificial Neural Networks	9
2.2.2	Structure and Mathematical Framework	10
2.2.3	Learning Process of a network	12
3	Framework	15
3.1	Training Process	15
3.1.1	Network structure	15
3.1.2	Other parameters	16
3.1.3	Preparation of input data for the training process	17
3.2	Evaluation Process	19
3.3	Classical Extrapolation methods	20
4	Extrapolations for the nuclei ^2H, ^3H and ^4He	22
4.1	Results	22
4.2	Comparison with Classical Extrapolations	24
5	Influence of training set modifications	26
5.1	Limitation on the model-space dimension	26
5.2	Restriction to SRG evolved interactions	27
5.3	Results and Comparison	27
5.4	Conclusion	30
6	Difference based extrapolation framework	31
6.1	New framework	31
6.2	Results and Comparison	32
6.3	Conclusion	35
7	Extrapolation of ^6Li	36
7.1	Results	36
7.2	Conclusion	38
8	Outlook	40
8.1	Physical Frequency Selection	40



8.2 Manual Frequency Selection	42
9 Conclusion	44

1 Introduction

An atomic nucleus consists of subnuclear particles called nucleons, which in turn consist of smaller elementary particles called quarks. As a result of the strong nuclear force, binding potentials of a nucleus are of very complex nature which cannot be described by simple radially symmetric potentials. Since the interactions between nucleons are not yet fully understood, the formulated quantummechanical many-body problem of a nucleus is an active research area in physics. Modern approaches to solving this problem include the so-called *ab initio* No-Core Shell Model (NCSM), which provides quantifiable approximations by using controlled truncations of the full nuclear Hilbert space. From the systematic increasement of these truncations and, thus, the corresponding model-spaces, converging series emerge whose limits are the solutions in the full Hilbert space.

As NCSM calculations get more computationally expensive for larger model-space sizes and heavier nuclei, the actual ground-state energy cannot be directly constructed from the resulting energy sequences. To estimate the ground-state energy out of the approximate energies coming from NCSM calculations, extrapolation methods have to be used. Classical extrapolation strategies include exponential function fits on the converging sequences. In this thesis, we investigate another approach to this extrapolation by using artificial neural networks (ANN). ANN's are general data processing constructs that can model complex connections in data. For that, the networks have to be trained on a dataset with known predictions.

There have been various applications in extrapolating NCSM sequences. This includes the work of Negoita et. al [1] and of Jiang et. al [2]. In this thesis, we aim to reconstruct and modify the extrapolation framework which was developed by Cedric Wenz [3] and directly extrapolates an absolute ground-state energy by taking untransformed NCSM sequences for multiple oscillator frequencies into account. Based on this absolute-based extrapolation framework, we further develop a second framework which is based on extrapolating sequences of differences between two results of consecutive model-space sizes. The networks get trained by NCSM datasets for the nuclei ^2H , ^3H and ^4He .

For both frameworks, the extrapolation quality gets analyzed in dependence on the model-space size of the evaluation data, as well as the preconditioning of the used Hamiltonian using the Similarity Renormalization Group (SRG) transformation, by evaluating the nuclei used in training with different models for the nuclear interaction. Afterwards, we are going to discuss the influence of two different modifications in the training data set on the extrapolations.

In a final application, we will discuss the extrapolation of both frameworks by evaluating the nucleus ^6Li , which was previously unseen by the networks. We will further be drawing a conclusion about the two extrapolation frameworks as well as the training set modifications and state a final extrapolation result for this nucleus.

Lastly, we want to briefly outlay two general ideas for the extrapolation of NCSM calculations by preselecting the input data.

2 Theoretical Foundations

2.1 The *ab initio* No-Core Shell Model formalism

A nucleus is bound together by the complex interactions between the nucleons. Nucleonic interactions are typically very short-ranged. This can be seen by looking at the mean binding energy per nucleon. In nuclei of all masses, this binding energy per nucleon is around 8 MeV. Since this does not scale with the amount of interaction pairs of nuclei, which grows proportional to the square of the mass number A , it follows that nucleons only interact with a constant amount of next neighbors. We can thus approximate a nuclear Hamiltonian by limiting ourselves to nucleonic interactions between two or three nucleons. Interactions between three nucleons cannot be described by ones between two nucleons and are thus irreducible, so that they have to be modeled separately. The Hamiltonian is given by

$$\hat{H} = T + V = \frac{1}{A} \sum_{i < j}^A \frac{(\vec{p}_i - \vec{p}_j)^2}{2m} + \sum_{i < j}^A V_{\text{NN},ij} + \sum_{i < j < k}^A V_{\text{NNN},ijk} + \dots \quad (2.1)$$

The interaction terms V_{NN} and V_{NNN} are highly complex and do not only depend on the distance between the nucleons but also on the spin-orbit coupling, the relative momentum and tensor terms.

We use an *ab initio* method to numerically compute the solution of the stationary Schrödinger equation for a nucleus. *Ab initio* is a way to solve nuclear many-body problems based on realistic interactions using controlled truncations with quantified theoretical uncertainties. For that, we need a realistic model of the nuclear hamiltonian using interactions which are precise enough. All *ab initio* methods use some sort of quantifiable truncation of the full Hilbert space to a smaller model-space.

In the *No-Core Shell Model* (NCSM), we want to solve the stationary Schrödinger equation

$$\hat{H} |E_n\rangle = E_n |E_n\rangle \quad (2.2)$$

using many-body Slater determinants $\{|\phi_i\rangle\}$ constructed from a harmonic oscillator single-particle basis. Since the single-particle harmonic oscillator basis depends on a previously chosen oscillator frequency $\hbar\Omega$, we have to specify this free parameter, which in turn determines the oscillator length $(\hbar/m\Omega)^{1/2}$. To numerically compute the solution, we expand the above equation into a matrix eigenvalue problem

$$\sum_j \langle \phi_i | \hat{H} | \phi_j \rangle \langle \phi_j | E_n \rangle = E_n \langle \phi_i | E_n \rangle. \quad (2.3)$$

The matrix consists of the matrix elements of the Hamiltonian $\{\langle \phi_i | \hat{H} | \phi_j \rangle\}_{ij}$ and the solution is the eigenvalue E_n as well as the eigenvector $\{\langle \phi_i | E_n \rangle\}_i$. The wanted eigenstate of \hat{H} can then be found by

$$|E_n\rangle = \sum_i \langle \phi_i | E_n \rangle |\phi_i\rangle. \quad (2.4)$$

This method is still not computationally feasible, since the many-body basis is infinite dimensional. We thus have to truncate the Hilbert space \mathcal{H}_A to a smaller model-space \mathcal{M} of dimension D . In the NCSM, this truncation is chosen such that Slater determinants with at most N_{\max} excitation quanta are used in the model-space. The ^{16}O configuration shown in Figure 2.1 has $N_{\max} = 6$ excitation quanta and the respective Slater determinant is present in all model-spaces with $N_{\max} \geq 6$. To solve the eigenvalue problem in the finite model-space, we have to do a variational calculation with the trial state

$$|\psi_T\rangle = \sum_i^D C_i |\psi_i\rangle, \quad (2.5)$$

which formally leads to the matrix eigenvalue problem, truncated to the model-space \mathcal{M} . Obviously, the reconstructed many-body state $\psi_T(D)$ and in particular the calculated energy $E_n(D)$ is dependent on the dimension D of the model-space. From the properties of variational principles, it follows that the ground state energy E_0 is always less than the expectation value of the trial state. In fact, the much stronger *Hylleraas-Undheim Theorem* holds. It states that all states have a monotonously decreasing energy

$$E_n(D) \geq E_n(D+1) \quad (2.6)$$

with increasing model-space dimension D , if the model-space of dimension $D+1$ is constructed by appending one basis vector to the model-space of dimension D . Furthermore, the energies cannot fall below the exact energy E_n of the full Hilbert space. Both the monotonicity and the boundedness from below imply the convergence of $E_n(D)$ to the exact energy.

To calculate the eigenvalues of truncated matrices, we can use iterative methods such as the *Lanczos algorithm* and obtain the lowest energy values. Since the matrix dimensions scale with the total amount of nucleons A , it is not feasible to calculate energies of heavier nuclei up to a satisfactory convergence. This opens doors to various extrapolation methods and is the direct connection to the interest of this thesis in extrapolating ground-state energy values using neural networks [4, 5].

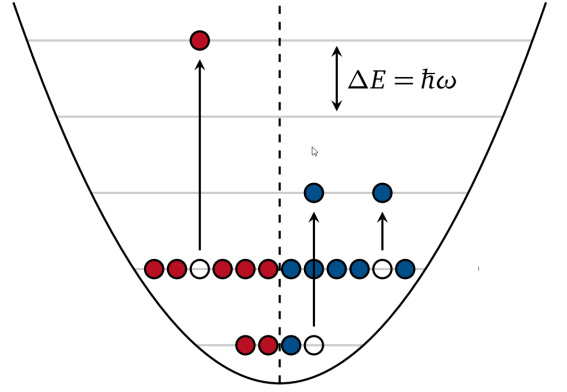


Figure 2.1: An example configuration of ^{16}O with $N_{\max} = 6$ excitation quanta [4]. The horizontal lines represent different energy levels of the harmonic oscillator. Multiple nucleons (blue and red balls) are excited from their ground state, leaving holes (uncolored balls).

2.1.1 Similarity Renormalization Group

To improve the results of NCSM calculations and improve the convergence of the energy sequences, we can pre-diagonalize the Hamiltonian using the *similarity renormalization group* (SRG). The idea is to use a unitary transformation of the Hamiltonian with a unitary operator \hat{U}_α to define an evolved Hamiltonian \hat{H}_α with

$$\hat{H}_\alpha = \hat{U}_\alpha^\dagger \hat{H} \hat{U}_\alpha =: T_{\text{rel}} + V_\alpha. \quad (2.7)$$

The evolution of \hat{H}_α is given by the *flow equation*

$$\frac{d\hat{H}_\alpha}{d\alpha} = [\hat{\eta}_\alpha, \hat{H}_\alpha] \quad (2.8)$$

with the *evolution generator*

$$\hat{\eta}_\alpha = \frac{d\hat{H}_\alpha}{d\alpha} \hat{U}_\alpha^\dagger = -\hat{\eta}_\alpha^\dagger. \quad (2.9)$$

For a choice of an evolution generator and a flow parameter α , we get an evolved Hamiltonian \hat{H}_α with matrix entries more focused to the diagonal. This pre-diagonalization conserves the spectrum of \hat{H} and can thus be used in No-Core Shell Model calculations to gain a faster convergence [4].

2.2 Machine Learning

Since the results of No-Core Shell Model calculations can show a variety of different convergence behaviors for different observables depending on the nucleus or the chosen models of the nuclear force, it is of large interest to develop extrapolation methods which capture the complexity of the resulting sequences and predict accurate values for various input sequences.

In this section, we provide the theoretical foundation of artificial neural networks, which have already been used in various other physical applications, for example in modeling β -decay half lifes [6] or nuclear charge radii [7].

2.2.1 Introduction to Artificial Neural Networks

An artificial neural network is a pattern recognition algorithm in the field of machine learning. The term *neural network* refers to the original idea to mimic the learning mechanism in biological organisms.

Inside the human brain, countless cells called neurons are connected to each other in a highly complex system. The strengths of these synaptic connections often change in response to external stimuli. This change is how the process of learning takes place in living organisms [8].

This learning mechanism is translated to a machine learning algorithm by using basic data processing units that model the behavior of neurons. Because of that, the basic units of an artificial neural network are called neurons as well. These neurons are linked via directed connections. If we wanted to model the structure of the brain more precisely, there would be no restrictions on which neuron could link to which, since this is the case in the brain. The machine learning counterpart to this is called a *recurrent network*. The mathematical model of recurrent networks can get arbitrarily complex, such that we only look at neural networks where the neurons are grouped in different layers and only connections from one layer to the next are permitted. This is called a *feedforward network*. Throughout this thesis, we only look at feedforward networks.

In a feedforward network, we hope that the layers correspond to rising levels of abstraction, such that the neurons in earlier layers may capture every detail of the input data and neurons in later layers may capture subpatterns in the input data.

The process of using neural networks to make predictions about data sets consists of multiple phases. In general, we want to *train* a neural network by using data sets with known predictions and use them to adjust the model parameters of the network to maximize the precision of the predictions. For that, we need a large data set of NCSM calculations with known predictions. We therefore use sequences that have converged enough so that the limit can be used as a prediction target. This set is called the *training set*. After training a network, the nets will be *validated* using samples from a *validation set*, which also has known prediction

targets. By comparing the predictions of the network with the targets, we can quantify the quality of the network. In Figure 2.2, we can see examples of networks, that underfit or overfit the training data, as well as a network that fits the training data well.

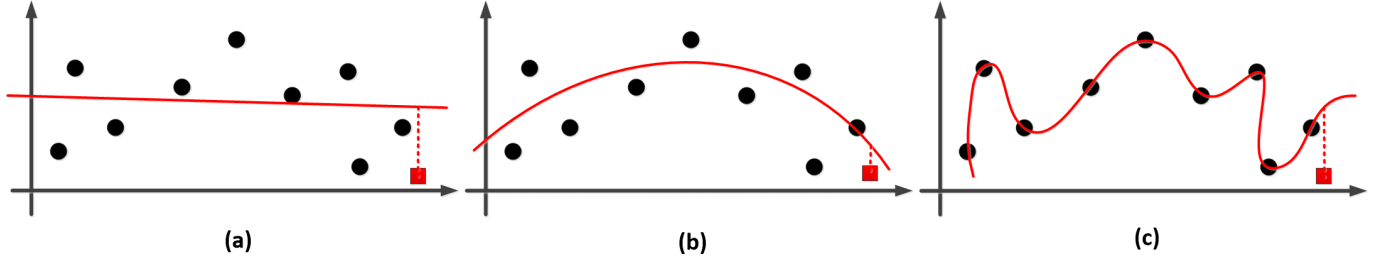


Figure 2.2: Visualization of different qualities of network fits [9]. The red line represents the data fit that was generated during training. The black dots symbolize the training samples and the red square denotes a validation sample. In the first example (a), the network doesn't fit the training data well, such that the network prediction of the validation sample indicated by the meeting point of the dashed and solid red line is far away from the prediction target. Such a network would not qualify for evaluation. In the second example (b), the network fits the data well, such that the prediction is close to the target. This network would meet the quality requirement. In the last example (c), the network fits the training data too well, such that the prediction for the validation target is too unrealistic. This network would not meet the quality requirements.

After checking the quality of a network, we can then feed *evaluation data* with unknown prediction targets into the network and gain a network prediction.

2.2.2 Structure and Mathematical Framework

To understand how an artificial neural network processes large amounts of data, we must first discuss the structure of such networks.

In an ANN, the basic data processing unit is a neuron. Its only job is to take input signals from directed connections to other neurons and process these inputs to generate an output signal, which can itself be connected to the input of other neurons. In Figure 2.3, a sample neuron is depicted.

The activation a of a neuron is computed by adding the input signals x_j multiplied with connection weights w_j , representing the strength of the connection. This weighted sum is then added to a neuron bias b and normalized using an activation function σ , yielding

$$a = \sigma \left(\sum_j w_j x_j + b \right). \quad (2.10)$$

In most use-cases, the activation function σ is chosen such that the neuron outputs are restricted to subsets of \mathbb{R} . To force neuron activations to lie in $(0, 1)$, one can use the sigmoid function $\sigma(z) = [1 + \exp(-z)]^{-1}$.

In feedforward neural networks, neurons are organized in layers 1 to N . In each layer n , there are $L^{(n)}$ neurons and the neurons have an outbound connection to every neuron in layer $n + 1$. The $L^{(1)}$ neurons of

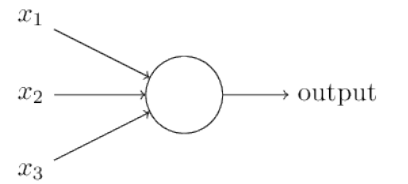


Figure 2.3: An example neuron with three input connections x_1 , x_2 and x_3 and an output connection. [10]

layer 1 have no input connections and can be used to directly feed $L^{(1)}$ -dimensional input data to the network. The first layer of a neural network is thus called the *input layer*. When we feed $L^{(1)}$ -dimensional data into the network, it propagates forward, layer-by-layer until the last layer, where the activations can be viewed as the processed output of the network. The last layer is thus called the *output layer*. The layers between the input and the output layer are called *hidden layers*, since the activation of those neurons are neither direct inputs nor outputs of the data processing. In Figure 2.4, we can see a sample feed forward neural network.

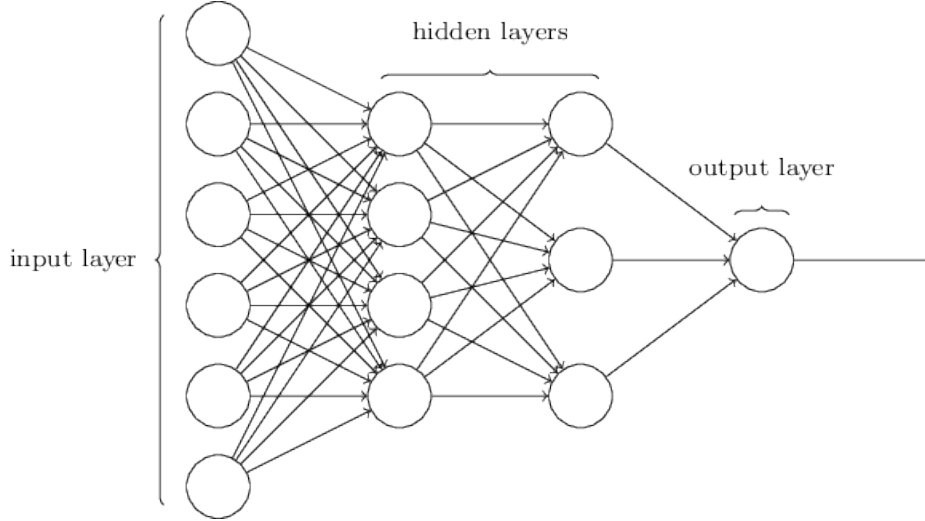


Figure 2.4: A neural network with four layers [10]. Here, there are 6 neuron in the input layer ($L^{(1)} = 6$), 3 neurons in each hidden layer ($L^{(2)} = L^{(3)} = 3$) and one output neuron in the output layer ($L^{(4)} = 1$). The connection between the neurons are limited to adjacent layers and information is only flowing from left to right. The depicted network is thus a feedforward network.

Now, let $\vec{a}^{(n)} = (a_1, a_2, \dots, a_{L^{(n)}}) \in \mathbb{R}^{L^{(n)}}$ be the activations and $\vec{b}^{(n)} = (b_1, b_2, \dots, b_{L^{(n)}}) \in \mathbb{R}^{L^{(n)}}$ be the biases of the neurons in Layer n . For $1 < n < N$, the activation of the neurons in layer $n + 1$ can be functionally described by

$$A^{(n)} : \mathbb{R}^{L^{(n)}} \mapsto \mathbb{R}^{L^{(n+1)}} \quad (2.11)$$

$$\vec{a}^{(n)} \mapsto \vec{a}^{(n+1)} = \sigma \left(W^{(n)} \vec{a}^{(n)} + b^{(n+1)} \right). \quad (2.12)$$

Here, $W^{(n)} = (w_{ij}^{(n)})$ is the $(L^{(n+1)} \times L^{(n)})$ -dimensional Matrix of the weights $w_{ij}^{(n)}$ from neuron j in layer n to neuron i in layer $n + 1$. Furthermore, the activation function σ is applied elementwise. In that sense, one can view a neural network as a function

$$f : \mathbb{R}^{L^{(1)}} \mapsto \mathbb{R}^{L^{(N)}} \quad (2.13)$$

$$x \mapsto \left(A^{(N-1)} \circ \dots \circ A^{(2)} \right) (W^{(1)}x + b^{(2)}). \quad (2.14)$$

Before one constructs a feedforward artificial neural network, one has to think about the structure (i.e the count of layers and the count of neurons in each layer) and the used activation function. Those values have to be chosen such that the network can fulfill the data requirements and output the right data.

For example, if one uses an ANN to classify greyscale images of pixel-size 24×24 into two classes A and B (this is called a *binary classification problem*), one could choose $L^{(1)} = 24 \cdot 24 = 576$ and $L^{(N)} = 2$, such that

each greyscale value can be fed into one input neuron and each output neuron corresponds to one of the two classes A and B . If one restricts the output neurons to the range $(0, 1)$, one could extract the prediction of the network by choosing the neuron with the highest activation.

The function (2.13) is dependent on the weights and biases of the connections and neurons. They can be viewed as internal parameters of f . When the network is initialized, they are often set to random values and the network will output random data. The challenge of machine learning is to apply algorithms on the outputs of the network to carefully adjust the weights and biases which results in the optimization of the network model [10].

2.2.3 Learning Process of a network

There are many ways to tackle the optimization process of neural networks. Training methods all rely on evaluating the outputs of some given inputs and comparing them to the values that *should* be output by the network.

To quantify the performance of a given network f , we define a *loss function* which depends on given inputs $x_1, x_2, \dots, x_n \in \mathbb{R}^{L^{(1)}}$, their desired outputs $o_1, o_2, \dots, o_n \in \mathbb{R}^{L^{(N)}}$ and, most importantly, the weights w and the biases b of the network, by

$$C(w, b) = \frac{1}{n} \sum_{i=1}^n \|f(x_i) - o_i\|^2. \quad (2.15)$$

Note that the loss function is always positive and only zero, when the network predicts the correct output for every input. There are many ways to choose a loss function that has these properties. The function (2.15) is called the *mean squared error* (MSE) and is used throughout this thesis as the loss function.

Using this terminology, we can translate the optimization process of a network to a well-posed problem of finding the minimum of a function. Usually, the number of dimensions is too high to compute the minimum of the loss function analytically. One has to rely on numerical methods to find a local minimum of (2.15). A basic numerical algorithm that is used in machine learning is called *gradient descent*. Even though this algorithm is very simplistic, it provides the basis for more sophisticated optimization algorithms such as the *AdamW algorithm*, which is used in this thesis.

The gradient descend method is an iterative method that can be used to find a local minimum of an arbitrary differentiable function $f : \mathbb{R}^n \mapsto \mathbb{R}, x \mapsto f(x)$. Starting from a point $x^{(0)} \in \mathbb{R}^n$, we iteratively adjust the value by a finite amount Δx . The function will change by

$$\Delta f \approx \frac{\partial f}{\partial x_1} \Delta x_1 + \dots + \frac{\partial f}{\partial x_2} \Delta x_2 = (\nabla f)(x) \cdot \Delta x. \quad (2.16)$$

If we choose $\Delta x = -\eta(\nabla f)(x)$, with a small positive parameter η called the *learning rate*, we can force the right hand side of (2.16) to be negative. In that way, if we adjust x by Δx , we can ensure that the new function value at $x^{(1)} = x^{(0)} + \Delta x$ will be smaller, if the approximation in (2.16) is good enough. For that, it is important to carefully choose the value of η small enough, but not too small. If the step size is too small, the new value will not have changed much and the total amount of steps must be increased to obtain a reasonable optimization.

Applied to the loss function (2.15), we get the iteration rule

$$w_i \mapsto w'_i = w_i - \eta \frac{\partial C}{\partial w_i}, \quad (2.17)$$

$$b_i \mapsto b'_i = b_i - \eta \frac{\partial C}{\partial b_i}. \quad (2.18)$$

Since the cost function depends on every input x_i , it is computationally infeasible to calculate the gradient of the loss function directly. Therefore, we employ the *stochastic gradient descent* algorithm. The idea is to approximate the gradient of the loss function by dividing the sample set into batches and calculating the gradient only using a single batch of samples. The resulting gradient will be used to adjust the weights and biases. This allows for more steps to be taken with a single iteration over the training data set, but the steps themselves are not as accurate as with gradient descent. An iteration over the whole training data set is commonly called an *epoch*. The training of a single network can be done by specifying a total amount of training *epochs*, or by specifying a *threshold* for the loss function for when the training should end.

To compute the gradient of the loss function for a single training sample x , a classical backpropagation algorithm is used. The derivative of the loss function

$$C_x(w, b) = \|a_i^{(N)} - o_i\|^2 \quad (2.19)$$

with respect to a weight $w_{ij}^{(N-1)}$ from the j -th neuron in the second last layer to the i -th neuron in the last layer is calculated using the chain rule to

$$\frac{\partial C_x}{\partial w_{ij}^{(N-1)}} = \frac{\partial C}{\partial a_i^{(N)}} \frac{\partial a_i^{(N)}}{\partial z_i^{(N)}} \frac{\partial z_i^{(N)}}{\partial w_{ij}^{(N-1)}}, \quad (2.20)$$

where $z_i^{(n)} = \sum_{j=1} w_{ij}^{(n-1)} a_j^{(n-1)} + b_i^{(n)}$ is the activation of neuron i in layer n before it is fed into the activation function. Using the Loss function from (2.19), we get

$$\frac{\partial C_x}{\partial w_{ij}^{(N-1)}} = 2(a_i^{(N)} - o_i^{(N)}) \cdot \sigma'(z_i^{(N)}) \cdot a_j^{(N-1)}. \quad (2.21)$$

Analogously, the derivative of C_x with respect to the bias $b_i^{(N)}$ is given by

$$\frac{\partial C_x}{\partial b_i^{(N)}} = 2(a_i^{(N)} - o_i^{(N)}) \cdot \sigma'(z_i^{(N)}). \quad (2.22)$$

These equations can be used to adjust the weights of the second last layer, as well as the biases of the last. Since we also need the derivative with respect to weights and biases in the earlier layers, we can use the clever trick of applying this step iteratively to one layer backwards at a time. This is why this algorithm is called backpropagation. To compute those derivatives, we need to know how the loss function changes if activations in the second last layer change. This is given by

$$\frac{\partial C_x}{\partial a_j^{(N-1)}} = \sum_{i=1}^{L^N} 2(a_i^{(N)} - o_i^{(N)}) \cdot \sigma'(z_i^{(N)}) \cdot w_{ij}^{(N-1)}. \quad (2.23)$$

We can use this derivative to calculate derivatives of the loss function with respect to earlier layers and iteratively determine the gradient of C_x . The gradient of the whole loss function C is then given by the average

$$\nabla C = \sum_x \nabla C_x \quad (2.24)$$

over all training samples in the batch [10].

3 Framework

The ground-state energy sequences of NCSM calculations can show different convergence rates depending on the nucleus. For lighter nuclei, it is computationally feasible to calculate these energy sequences to a high value of N_{\max} , such that the clear convergence to the actual ground-state energy can easily be seen. For heavier nuclei, the extension of the model-space for NCSM calculations to a higher N_{\max} results in the need of much more computational power, since there are more possibilities to distribute the N_{\max} excitation quanta to the nucleons. Therefore, NCSM calculations of heavier nuclei are restricted to model-spaces of lower N_{\max} , such that the resulting energy sequences won't show a satisfactory convergence to extract the exact ground-state energy of the nucleus, given the used model of the interactions. For those cases, we need a ready-to-use extrapolation method, which should provide an estimate and its uncertainty of the ground-state energy, given those unconverged sequences.

The aim of this thesis is to provide an extrapolation framework for ground-state energy sequences resulting from NCSM calculations using neural networks. In order to use neural networks to extrapolate energies from unconverged sequences, we must first train the network on selected training data. In this chapter, we describe the whole workflow of training and evaluating a network, as well as a state-of-the-art extrapolation method using exponential function fits which will later be used to roughly assess the quality of the new extrapolation methods.

3.1 Training Process

Before we introduce the complete workflow to train a network on NCSM ground-state energy sequences, we first have to decide on the network structure to use. This includes the structure of the input layer, the structure of the output layer, as well as the structure of the hidden layers.

3.1.1 Network structure

NCSM calculations are usually done with multiple different oscillator frequencies $\hbar\Omega$. Different oscillator frequencies result in different convergence rates of the energy sequences, but the limit stays the same. Thus, we want to incorporate more than one frequency into the extrapolation in order to give the maximum amount of information about the interaction and the nucleus into the network. For that, we do not only try to extrapolate a single energy sequence, but input multiple sequences for different oscillator frequencies into the network. We decide on 3 energy sequences with given frequencies to input into the network.

In order to determine the structure of the input layer, we must further choose a fixed length of the energy sequences which are put into the network. For NCSM calculations for heavier nuclei, there are not many N_{\max} values available, which means that we have to intentionally keep the sequence length small enough. We

decide on energy sequences of length 4. In summary, the input layer thus has to consist of $L^{(1)} = 3 \times 4 = 12$ input neurons.

Since the output of the neural network should be a single real number indicating the extrapolated ground-state energy, the output layer consists of $L^{(N)} = 1$ neuron.

For the hidden layers, there is no generally accepted method of defining the most efficient structure. For the most part, the count of the hidden layers, as well as the count of the neurons in those layers have to be chosen empirically, such that the network neither underfits nor overfits the data. For our framework, we choose $N = 4$ total layers, such that there are two hidden layers of $L^{(2)} = 24$ and $L^{(3)} = 12$ neurons.

Each layer besides the output layer will be given a *rectifier* activation function (*ReLU*) given by

$$\sigma(z) = \max(0, z), \quad (3.1)$$

which restricts the activations of the neurons to be positive. This function is chosen because it empirically yields the best results for the extrapolation. Furthermore, it simplifies the calculations on the neural network and with it the computation time. Notice that the activations of the last layer will not get rectified, as the extrapolations, and therefore the network output, should not be limited to positive ground-state energies. As a consequence, the connections of the second-last layer to the last layer will also be responsible for the sign of the extrapolated energy. In Figure 3.1, the data flow of our extrapolation framework is shown.

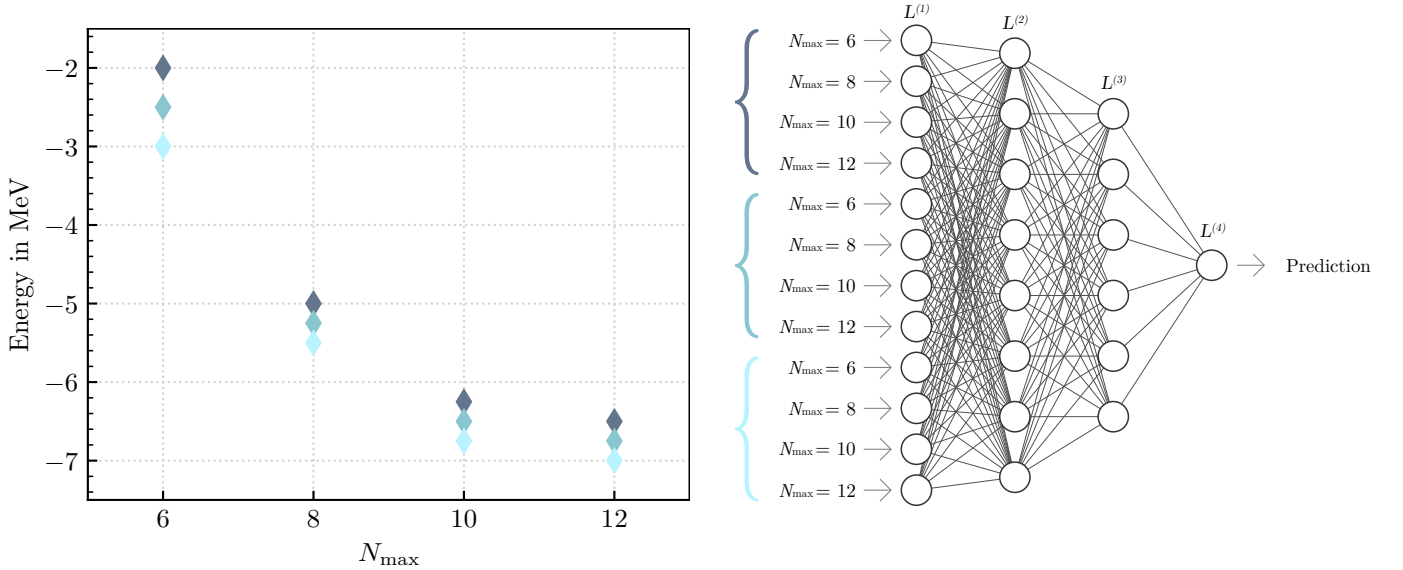


Figure 3.1: Our extrapolation framework visualized. On the left, an exemplary formatted input is shown, consisting of three sequences of four n_{\max} values. The data is input into the network, which then outputs an estimation on the common sequence limit. Note that not all neuron in layers $L^{(2)}$ and $L^{(3)}$ are shown for the sake of visibility. In our framework, $L^{(2)} = 24$ and $L^{(3)} = 12$.

3.1.2 Other parameters

The loss function of our network is a standard *Mean Squared Error* (MSE) loss given for a single network input x by

$$C_x = \|f(x) - o\|^2, \quad (3.2)$$

where $f(x)$ is the vector of activations of the neurons in the output layer given the input of x , and o is the vector of desired activations. The unit of the loss function is therefore MeV^2 . For n input samples x_i with desired outputs o_i , the loss is averaged over all samples yielding

$$C = \frac{1}{n} \sum_{i=1}^n C_{x_i} = \frac{1}{n} \sum_{i=1}^n \|f(x_i) - o_i\|^2. \quad (3.3)$$

For the optimization process, we opt for the *AdamW* algorithm. This is a variant of the *Adam* algorithm [11, 12], which is based on the simple stochastic gradient descent algorithm introduced in subsection 2.2.3. As such, the general optimization process is the same as with stochastic gradient descent.

For gradient-based optimization algorithms, we have to choose a learning rate η , which determines the ratio of the step size to the value of the gradient. Values of η that are too large result in an instability of the optimization algorithm, while values, which are too small, result in smaller step sizes. The learning rate is empirically chosen to be $\eta = 0.01$, such that both extreme cases will be avoided.

Overall, we iterate 20 times (*epochs*) over the training data set, using batch sizes of 512. Since the learning rate can independently be adjusted for each epoch, it is common practice to reduce the learning rate once the optimization has stagnated, such that further optimization may be achieved. In our network training, we decrease the learning rate by a factor of 2, if the loss function as measured by a small set of input data called the *development set* does not decrease by 10% over 2 epochs (This mechanism is called *learning rate scheduling*).

After the optimization is done, the network is validated using a *validation set*. Here, the loss is computed using all samples in the validation set and compared to a threshold of 0.3 MeV^2 . If the loss is smaller than the threshold, the network is suited for our extrapolation. Otherwise, the network will be discarded.

To implement the neural network, we use *pytorch*, which is a machine learning library for Python.

3.1.3 Preparation of input data for the training process

The networks have to be trained using energy sequences with known limits. Since it is impossible to calculate the limit using the full Hilbert space, we have to base the training on NCSM sequences which can be calculated to a sufficiently large value of N_{max} , such that the lowest value for the highest N_{max} gives a good enough approximate for the limit. For that, we need to assume that the sequences have more or less converged to the limit. For the training, we thus use NCSM calculations from the nuclei ^2H , ^3H and ^4He . The interactions used are chiral two-body and three-body interactions.

From the NCSM energy sequences, we first generate a large pool of input data which is formatted such that it can be input into the networks. This is necessary since NCSM energy sequences are generally calculated from more than three oscillator frequencies and more values of N_{max} .

To generate formatted input sequences, we first purge a few exotic cases from the raw input data. Since oscillator frequencies which are too high or too low result in slow converging sequences, we restrict the data of the training process to oscillator frequencies between $\hbar\Omega = 12 \text{ MeV}$ and $\hbar\Omega = 32 \text{ MeV}$. Furthermore, we discard every NCSM calculation done for $N_{\text{max}} = 0$, since they can result in unphysical energy values. For each nucleus and interaction, we also discard sequences for which the difference between the energy value for the highest N_{max} and the limit is higher than 0.05 MeV to rule out further special sequences that do not converge enough to the desired limit. In Figure 3.2, an example *raw* NCSM calculation for ^2H is shown, using

an SRG evolved Hamiltonian based on a chiral $N^3\text{LO}$ interaction with two-body interactions and a cutoff at 400 MeV by Entem, Machleidt and Nosyk [13]. The flow parameter is given by 0.04 fm^4 . This is an example of a calculation, which is suitable for training, since the calculations were done for sufficiently large N_{max} and the limit could be extracted. We can see the convergence for higher N_{max} to a value independent on the oscillator frequency.

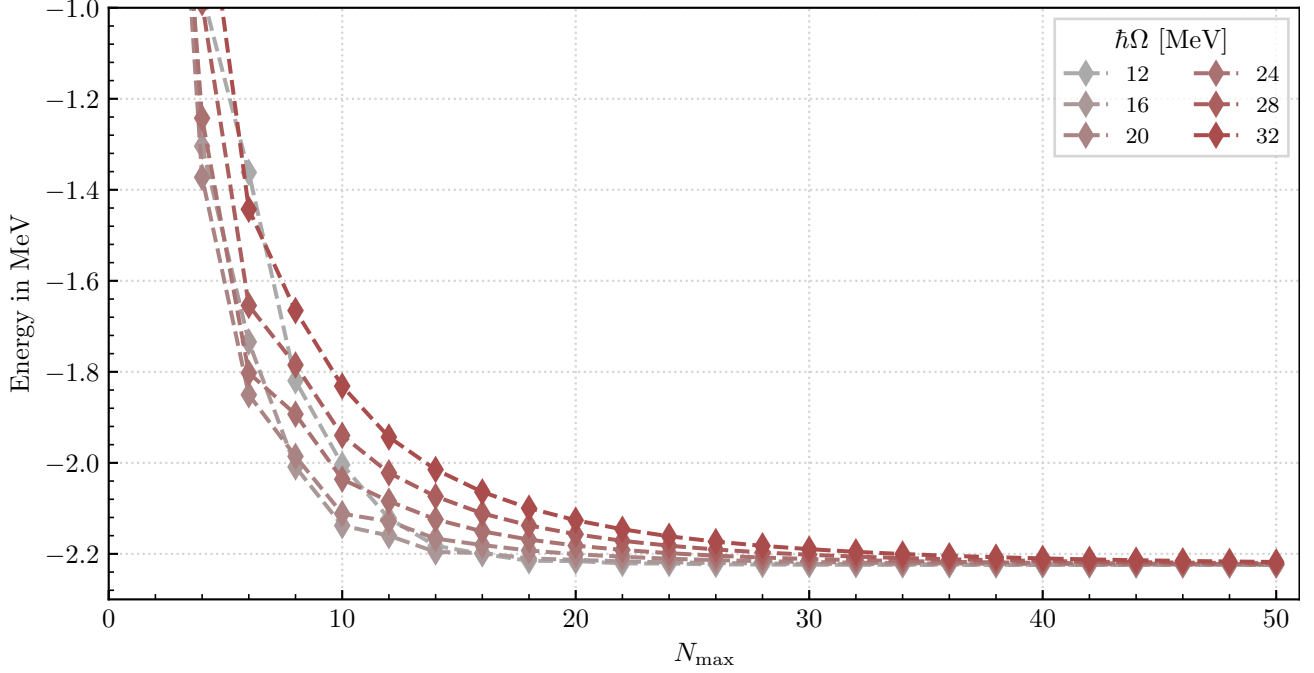


Figure 3.2: NCSM results for the ^2H nucleus. Here, the interaction used for the Hamiltonian is a chiral $N^3\text{LO}$ interaction with two-body interactions and a cutoff at 400 MeV by Entem, Machleidt and Nosyk [13]. Furthermore, the Hamiltonian is SRG evolved using a flow parameter of 0.04 fm^4 . Each point is the result of a NCSM calculation with a given N_{max} and oscillator frequency $\hbar\Omega$.

Afterwards, we consider every permutation of three frequencies belonging to the six calculated frequencies. Here, we deliberately want the order of the frequencies to not matter, since the network should be trained independent of the order of the inputted data. For each permutation of frequencies, we can choose any four consecutive N_{max} values to generate a formatted input sequence.

For any NCSM calculation with $k \geq 3$ Frequencies and $n \geq 4$ values of N_{max} , we can thus get $\frac{k!}{(k-3)!}(n-3)$ formatted input sequences into the pool. In the example from above, there are $k = 6$ frequencies and $n = 25$, yielding 2640 different formatted sequences. We call this preformatting the *inflation* process of the input data. With each inflated sequence coming from the same NCSM calculation, we associate the same limit extracted as the minimum value of the maximum N_{max} , which will be used as the target output of the network in the training process.

This large pool of inflated sequences is the basis for the complete training of a network, i.e. for the training itself, but also for the validation and the development set. As the training set should be very large to incorporate as many different sequences into the network as possible, we draw 200,000 random samples from the pool. For the validation of the network, we draw 10,000 random samples and for the development set, we draw 5,000 samples.

Even though the training set will be of a large size, all the samples are derived from a small list of nuclei, for which NCSM calculations up to a high enough N_{max} are possible to extract a meaningful limit. As a consequence, all the inflated sequences only cover a small range of absolute energy values as well as a small range of convergence rates. In order to make the network prediction independent of the absolute energy values and the convergence rates, every sample is also multiplied by a randomly chosen constant, to vary the convergence rate, and shifted by a randomly chosen constant, to vary the absolute energies in the sequence. For our work, the multiplier is chosen uniformly from the interval $[0.5, 2]$, and the shift is chosen uniformly from $[-10 \text{ MeV}, 10 \text{ MeV}]$.

3.2 Evaluation Process

To evaluate the network, we use a fixed evaluation set consisting of interactions for the three nuclei ^2H , ^3H and ^4He , which were also used in the training as well as the nucleus ^6Li , to test the extrapolation on a nucleus which was previously unseen by the network. For evaluation, we use semilocal momentum space regulated chiral interactions. Furthermore, each interaction will get evaluated using an SRG evolved hamiltonian with flow parameters of $\alpha = 0.04 \text{ fm}^4$ and $\alpha = 0.08 \text{ fm}^4$, to analyze the extrapolations in dependence on the flow parameter. These interactions belong to a different family than the interactions used in training. This means that the network still has to perform an extrapolation, even for the reused nuclei ^2H , ^3H and ^4He .

In the previous section we described the training process of a single network. Since one network can only output one prediction for a given input sequence, we cannot generate a big enough evaluation statistic if we only evaluate the data with one network. To fix that, we train multiple networks which will be evaluated. For our work, 100 networks will be trained to generate a broad statistic.

Since the networks should be applied to heavier nuclei, for which NCSM calculations cannot be done for sufficiently large N_{max} , we want to analyze how the evaluation changes if we restrict the evaluation samples to certain N_{max} values. To demonstrate this evaluation process, we will be using the example of a semilocal momentum space regulated chiral interaction of order $N^2\text{LO}$ with two-body interactions and a cutoff at 450 MeV for ^2H with a SRG flow parameter of 0.04 fm^4 [14], for which NCSM calculations have been done with 7 frequencies from $\hbar\Omega = 16 \text{ MeV}$ to $\hbar\Omega = 40 \text{ MeV}$. To simulate a real use-case of extrapolation, the evaluation process is first restricted to sequences with N_{max} values of 6, 8, 10, 12. Since there are more

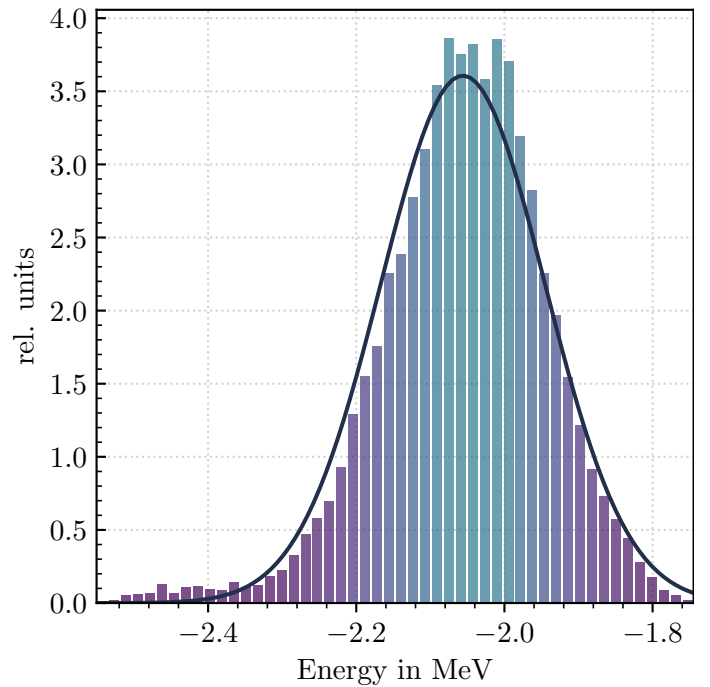


Figure 3.3: An example histogram of evaluations for a semilocal momentum space regulated chiral interaction of order $N^2\text{LO}$ with two-body interactions and a cutoff at 450 MeV for ^2H using a SRG flow parameter of 0.04 fm^4 [14]. The bar heights have been normed such that the total area of them is 1. The networks are evaluated using sequences from $N_{\text{max}} = 6$ to $N_{\text{max}} = 12$

than 3 frequencies available, a similar inflation process, as described in the previous section, will format the sequences according to the network structure. Now, every single sample can be evaluated by every of the 100 trained networks to generate a broad histogram of predictions. A sample histogram can be seen in Figure 3.3.

To quantify the evaluation of the networks, we can calculate the mean evaluation as the extrapolated ground state energy as well as the standard deviation to express the error of the extrapolation. To see how the evaluation changes if we consider sequences of higher N_{\max} , we also do this evaluation process for sequences with maximum N_{\max} of 16, 20 and 24. The data for all those evaluations will be condensed in a single plot for a given nucleus and interaction. In Figure 3.4, we see the evaluation of the example interaction from above.

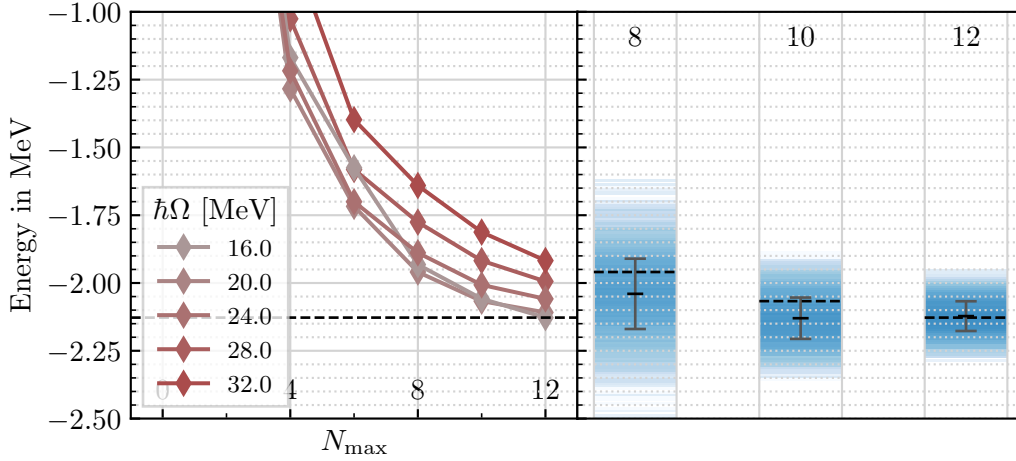


Figure 3.4: An example histogram of evaluations for a semilocal momentum space regulated chiral interaction of order N^2LO with two-body interactions and a cutoff at 450 MeV [14] for 2H using a SRG flow parameter of 0.04 fm^4 . On the left side the raw NCSM calculations are shown to compare them to the extrapolation. On the right side, for each maximum N_{\max} , the histogram, as well as its mean and standard deviation is shown, which represent the extrapolated energy and its uncertainty.

3.3 Classical Extrapolation methods

To assess our neural network extrapolation method, we want to compare it to classical, state-of-the-art model-space extrapolations. Classical extrapolations are usually done using exponential fits. Even though the convergence of ground state energy sequences does not resemble an exponential convergence, it provides a good estimate on the ground-state energy. To ensure comparability, the classical extrapolation method is restricted to the same N_{\max} values as in the neural network extrapolation, such that the input sequences are formatted exactly as described earlier, using all subsets of 3 out of all available oscillator frequencies.

In a classical extrapolation, an exponential function

$$f(x) = a \exp(-bx) + c \quad (3.4)$$

is fitted on a sequence of three N_{\max} values with a specific oscillator frequency. The resulting limit value c can then be extracted as the extrapolated value of the ground-state energy. Out of all available frequencies, we choose the frequency which has the lowest energy value at the highest N_{\max} . This is motivated by the fact that these sequences are usually more converged, which ensures more precision in the function fit.

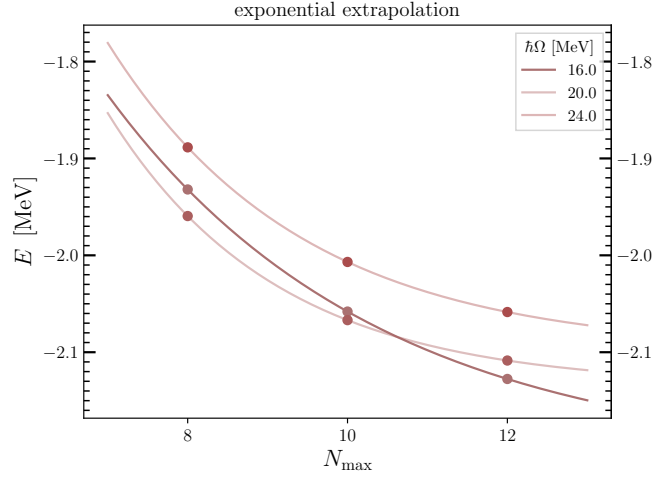


Figure 3.5: For the three different frequencies of $\hbar\Omega = 16, 20, 24$, a function fit of Equation 3.4 is shown. For the extrapolated ground-state energy, the function fit of the sequence with the lowest value for the highest N_{\max} is chosen. In this example, it is the frequency $\hbar\Omega = 16$

For example, if we wanted to extrapolate the sample in Figure 3.4 with frequencies $\hbar\Omega = 16, 20, 24$ for the case of maximum N_{\max} of 12, we choose the values $N_{\max} = 8, 10, 12$ of the $\hbar\Omega = 16$ sequence for extrapolation, since the absolute value of its sequence is the lowest among the three frequencies at $N_{\max} = 12$. The resulting function fit is shown in Figure 3.5

4 Extrapolations for the nuclei ${}^2\text{H}$, ${}^3\text{H}$ and ${}^4\text{He}$

In a first application of our extrapolation framework, we want to extrapolate the ground-state energies of the nuclei ${}^2\text{H}$, ${}^3\text{H}$ and ${}^4\text{He}$, which were already used in the training of the networks and analyze the extrapolation values in dependency on the maximum N_{max} of the input sequences. The networks were trained with NCSM calculations for chiral interactions of various leading orders and cutoffs for two-body interactions [13] as well as two- and three-body interactions [15]. Furthermore, the Hamiltonians were SRG evolved for flow parameters between 0.04 fm^4 and 0.08 fm^4 .

Instead of the many used interactions in the training, we decide on one interaction for the evaluation which will be used as a benchmark to compare this basic extrapolation framework to our more sophisticated methods which will be introduced in the later chapters. For the evaluation, we use a semi-local momentum space regulated N^2LO interaction with two-body interactions and a cutoff at 450 MeV by Maris et. al [14]. By using a different family of interactions for the evaluation than in the training, we hope to see whether the networks have captured the characteristics of the nuclei and reproduce those by extrapolating the results of previously unseen interactions. To investigate the effects of an SRG evolution of the Hamiltonian on the extrapolation results, we will further consider the above interaction in the case of an SRG evolution with a flow parameter α of 0.04 fm^4 and 0.08 fm^4 .

4.1 Results

In Figure 4.1, the evaluation of the interaction introduced above is shown for the three training nuclei.

An important metric in assessing our extrapolation comes from the variational principles of NCSM calculations for ground-state energies. Since the energies decrease monotonously with increasing model-space dimensions, it is guaranteed that the exact ground-state energy is lower than all computed NCSM values, regardless of model-space dimension. As a consequence, we can require from a reasonable extrapolation, that the extrapolated energy is lower than the lowest value at the maximum N_{max} . This minimum value is called the *variational boundary*. For all maximum N_{max} in Figure 4.1, those variational boundaries are shown as dashed lines. We can see that this condition is fulfilled for all nuclei in the case of a flow parameter of 0.04 fm^4 . This condition only fails for the extrapolations of ${}^2\text{H}$ with a flow parameter of 0.08 fm^4 . Additionally, we can also compare the extrapolation result to the variational boundary of the highest N_{max} available, as this provides a further upper bound for the exact ground-state energy. This upper bound is also interesting because it is not directly nor indirectly input into the networks, in contrast to the variational boundaries at the maximum N_{max} , and thus represents a stronger condition for the extrapolation quality of the networks. For the nuclei ${}^3\text{H}$ and ${}^4\text{He}$, this condition is also fulfilled. The extrapolations of the energies for the ${}^2\text{H}$ nucleus consistently fail to meet this condition.

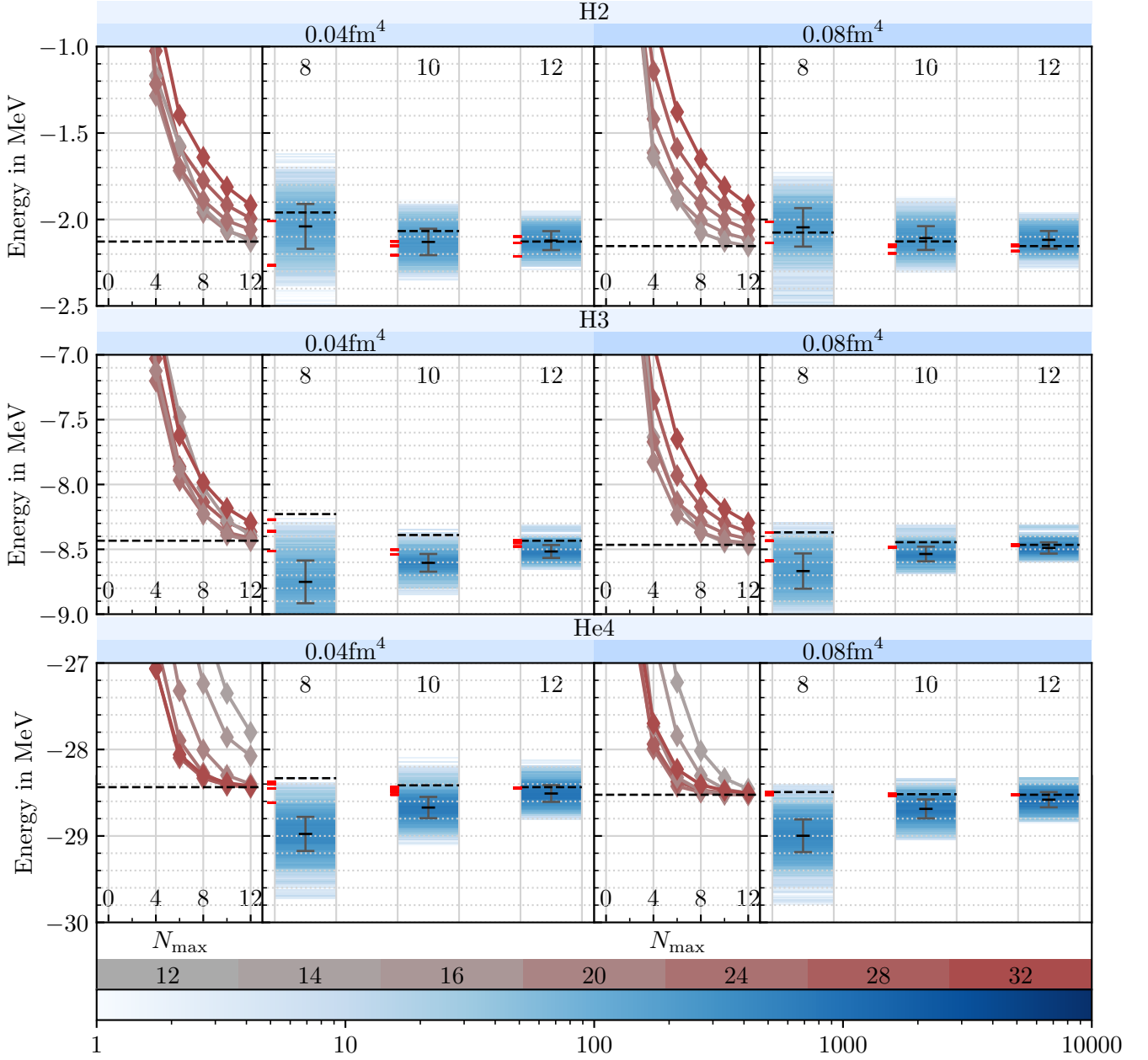


Figure 4.1: Evaluation of our extrapolation method on the nuclei ^2H , ^3H and ^4He , using a semi-local momentum space regulated $N^2\text{LO}$ interaction with two-body interactions and a cutoff at 450 MeV [14]. For each nucleus and each flow parameter, the NCSM sequences are shown on the left (the different frequencies are colored respectively to the legend, which shows the frequencies $\hbar\Omega$ in MeV) and the extrapolations for a given maximum N_{max} on the right. For each maximum N_{max} , the variational boundary is shown as a dashed line, and the classical extrapolations are shown as red ticks.

On first sight, the network predictions get more precise when we restrict the evaluation samples to higher values of N_{max} . This indicates that the networks associate steeper slopes of energy sequences with sequences that are not converged already as well as flat sequences with sequences that are more converged. Based on

this, the network tries to extrapolate the steeper sequences with lower predictions and flatter sequences with higher predictions, that lie more closely to the variational boundary. In fact, for the lower values of N_{\max} , the networks generally predict a value for the ground-state energy which is too low in comparison to the variational boundary at the given N_{\max} or even the variational boundary for the highest available N_{\max} , which is drawn as a dashed line on the left side of the plots.

Additionally, since the sequences for the different oscillator frequencies converge at different rates, they are further apart for lower N_{\max} and thus the network extrapolates them on a broader range of energies, resulting in a bigger uncertainties for smaller N_{\max} .

An exception to this general extrapolation behavior of extrapolating unreasonably low values for smaller N_{\max} and correcting the predictions towards the variational boundary for higher N_{\max} is the nucleus ${}^2\text{H}$. Here, our network shows a systematically different extrapolation behavior. First, the networks extrapolate the energies not far enough for all N_{\max} . Secondly, the predictions generally get lower if the maximum N_{\max} of the evaluation samples is increased. This may originate from the fact that the NCSM calculations for ${}^2\text{H}$ are the slowest to converge among the three nuclei. The slow convergence results in some sequences having a much flatter slope while still being too high in value with respect to the variational boundary, which is determined by the fastest converging sequence. Another possible origin of this systematic difference is that the training set of nuclei is not balanced evenly between the nuclei and ${}^2\text{H}$ may be underrepresented, such that the slow convergence behavior of ${}^2\text{H}$ is foreign to the networks.

If we look at the differences between the two SRG flow parameters of 0.04 fm^4 and 0.08 fm^4 , we cannot find a systematic differences but a general shift of the extrapolations to higher values with respect to their variational boundaries. This is not surprising, since a higher SRG flow parameter leads to faster convergence of the NCSM sequences. It results in lower variational boundaries as well as in a higher extrapolation, which is an effect we already observed in the ${}^2\text{H}$ extrapolations.

4.2 Comparison with Classical Extrapolations

In this section, we want to compare our extrapolation framework to a classical approach of extrapolating NCSM sequences. For that, exponential function fits as described in section 3.3 are used to extrapolate the limit value of the sequences. The resulting extrapolation values are shown as red tickmarks in Figure 4.1 for each maximum N_{\max} .

We first look at the exponential extrapolations of ${}^3\text{H}$ and ${}^4\text{He}$. For all N_{\max} , the exponential fit results in a much smaller uncertainty of the fitted limits. Moreover, the exponential extrapolations do not violate the variational boundary condition for those N_{\max} . Generally, the exponential extrapolations lie way closer to the variational boundary than the extrapolations from our framework. Since the evaluated sequences converge fast (the convergence can already be seen for the higher oscillator frequencies at the higher N_{\max} values), the variational bound gives a good estimate on the actual ground-state energy and thus, the exponential fits seem to be very accurate. This is reinforced by the fact that the exponential fits get more precise for the higher SRG flow parameter. This can especially be seen in the $N_{\max} = 8$ point for ${}^4\text{He}$. For the nuclei ${}^3\text{H}$ and ${}^4\text{He}$, the exponential fits clearly outperform our extrapolation framework for all N_{\max} .

Considering the fact that the NCSM calculations, from which those extrapolations base off of, already converge fast, those nuclei provide ideal conditions for the exponential extrapolations. As such, those nuclei do not represent an actual usecase for extrapolations, since they are not needed for converged NCSM results. To better compare the two extrapolation methods, we will look at the extrapolations of ${}^2\text{H}$, which provides a

more realistic use-case, as the NCSM results do not converge as fast. Here, we can see that both extrapolation methods are very comparable, both in precision and in accuracy. Interestingly, even the classical extrapolations violate the variational boundary condition for all N_{\max} except $N_{\max} = 10$ for 0.04 fm^4 . This seems unlogical at first, since exponential fits should naturally enforce a monotonicity constraint on the extrapolated limit. Actually, the exponential fits that result in an overestimation are coming from the sequences, for which the sequences have not converged as much. Since the variational boundary is determined by the frequencies for which the sequence has converged the most, it is possible for some sequences to result in a prediction which violates the variational boundary condition. This indicates that the violation of the network extrapolations does not come from an unoptimized training with unbalanced input sequences, but from the fact that the sequences are converging slowly. Also, NCSM calculations for the ground-state energy generally do not show an exponential convergence, which will lead to an increased uncertainty for lower N_{\max} . Since the exponential fits force a lower limit than the value at the highest N_{\max} , it is guaranteed that at least one prediction is lower than the variational boundary.

An important difference between the exponential fits and our extrapolation framework is a preconditioning of some sort for the input sequences. For the exponential function fits, this preconditioning consists of only using the best sequence, i.e. the sequence that has the lowest energy value at the highest N_{\max} , for the extrapolation. In our extrapolation framework, we intentionally do not preselect the "best" sequences in order to provide a more general extrapolation tool that does not depend on any sort of preselecting, which requires a metric on what sequences to pick for extrapolating the ground-state energy. In that regard, our extrapolation framework produces promising results for unconverged sequences, as they are already comparable in quality to the classical extrapolations.

5 Influence of training set modifications

In the previous chapters, we have outlaid a basic extrapolation framework for ground-state energy sequences coming from NCSM calculations. The network extrapolations have provided suboptimal results for fast converging sequences from ^3H and ^4He , as the network consistently predicted a ground-state energy that is too low, especially for the smaller N_{max} values, which are important for later use-cases for nuclei whose NCSM calculations cannot be done to a sufficiently high N_{max} . For the ^2H nucleus, the networks have predicted values which were too high, but comparable to classical extrapolations coming from exponential function fits.

In this section, we will modify the training data of our extrapolation framework using two different physically motivated training modes in order to analyze how the training set affects the evaluation of the three nuclei ^2H , ^3H and ^4He . Those modifications only take place in the training process of our networks, meaning the evaluation process is unaffected by those modifications such that they can be assessed in the same way as our basic training mode. This also means that our extrapolation framework is still free of any sort of preselection of the evaluation sequences to provide a more general extrapolation, which takes every available oscillator frequency into account.

5.1 Limitation on the model-space dimension

For the training process, we deliberately use NCSM calculations which are calculated to a sufficiently high N_{max} , to extract the limit as a prediction target for the networks. In our basic training mode, the sequences further get divided into subsequences of four consecutive N_{max} values. The network then gets trained to predict the same limit value for all of those subsequences. Since there are very high N_{max} values available, the training set of input sequences also contains sequences with almost constant values, as the sequences have already converged enough to the limit value at that model-space dimension. The result in the network also getting trained on converged sequences, for which there is no extrapolation needed. In Figure 5.1, we can see an example sequence which will be used in training. It is clear that sequences above $N_{\text{max}} \approx 26$ are almost converged and thus provide no new insights for the network about how to extrapolate unconverged sequences. Furthermore, those sequences might even prevent the networks from extrapolating unconverged sequences, as the networks could learn to just reproduce the lowest value of the sequence, as is the case for the converged sequences.

In order to conform to the later use-cases of the extrapolation of unconverged sequences, we should also focus our training on the extrapolation of those sequences. To achieve this, we want to restrict the input sequences to lower values of N_{max} . As we can already see in Figure 5.1, a restriction of subsequences to a maximum N_{max} of 24 will lead to the exclusion of the extreme cases of converged sequences and an inclusion of high enough N_{max} values to allow for different convergence rates in the training set.

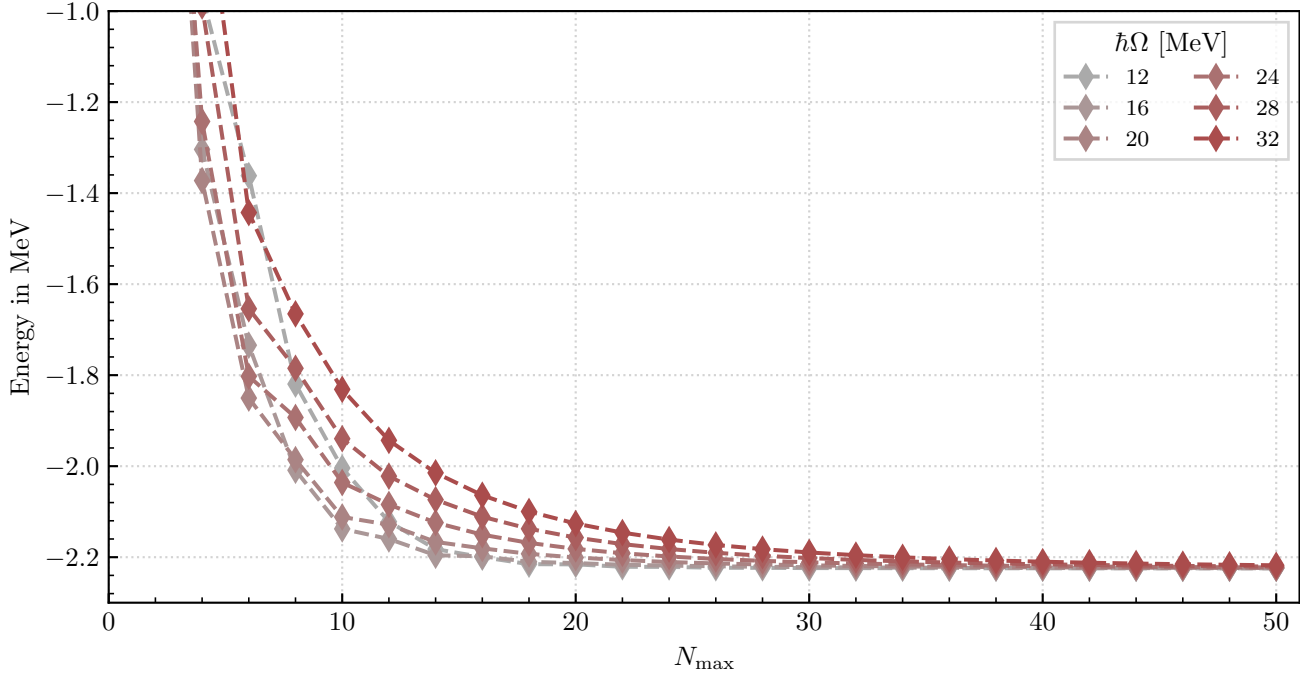


Figure 5.1: NCSM results for the ${}^2\text{H}$ nucleus. Here, the interaction used for the Hamiltonian is a chiral $N^3\text{LO}$ interaction with two-body interactions and a cutoff at 400 MeV by Entem, Machleidt and Nosyk [13]. Furthermore, the Hamiltonian is SRG evolved using a flow parameter of 0.04 fm^4 . Each point is the result of a NCSM calculation with a given N_{max} and oscillator frequency $\hbar\Omega$.

5.2 Restriction to SRG evolved interactions

In our second training mode, we want to take another approach to our extrapolation framework as the N_{max} -limitation training mode. As we have already seen, a SRG evolution of the Hamiltonian results in a faster convergence of the NCSM sequences. Taking this into account, we can also try to train the networks only using NCSM calculations with SRG evolved Hamiltonians using a minimum flow parameter of 0.04 fm^4 . In doing this, we hope to achieve a more uniform and stronger convergence in the different sequences in the training, such that the networks do not have to extrapolate as much as with unconverged sequences. This will also be useful for the later uses of the extrapolation on heavier nuclei, as an SRG evolution can and will always be done.

5.3 Results and Comparison

In order to evaluate the training-modified versions of our extrapolation framework, we have to make sure that the evaluation data conforms to the restrictions that we apply to our training set to ensure consistency. In our first training mode, the used N_{max} are restricted to be lower than 24. In the second training mode, only interactions which are SRG evolved using a flow parameter of at least 0.04 fm^4 are used. Our evaluation set, consisting of N_{max} values up to 12 and of interactions evolved using a flow parameter of 0.04 fm^4 and 0.08 fm^4 , already satisfies these requirement.

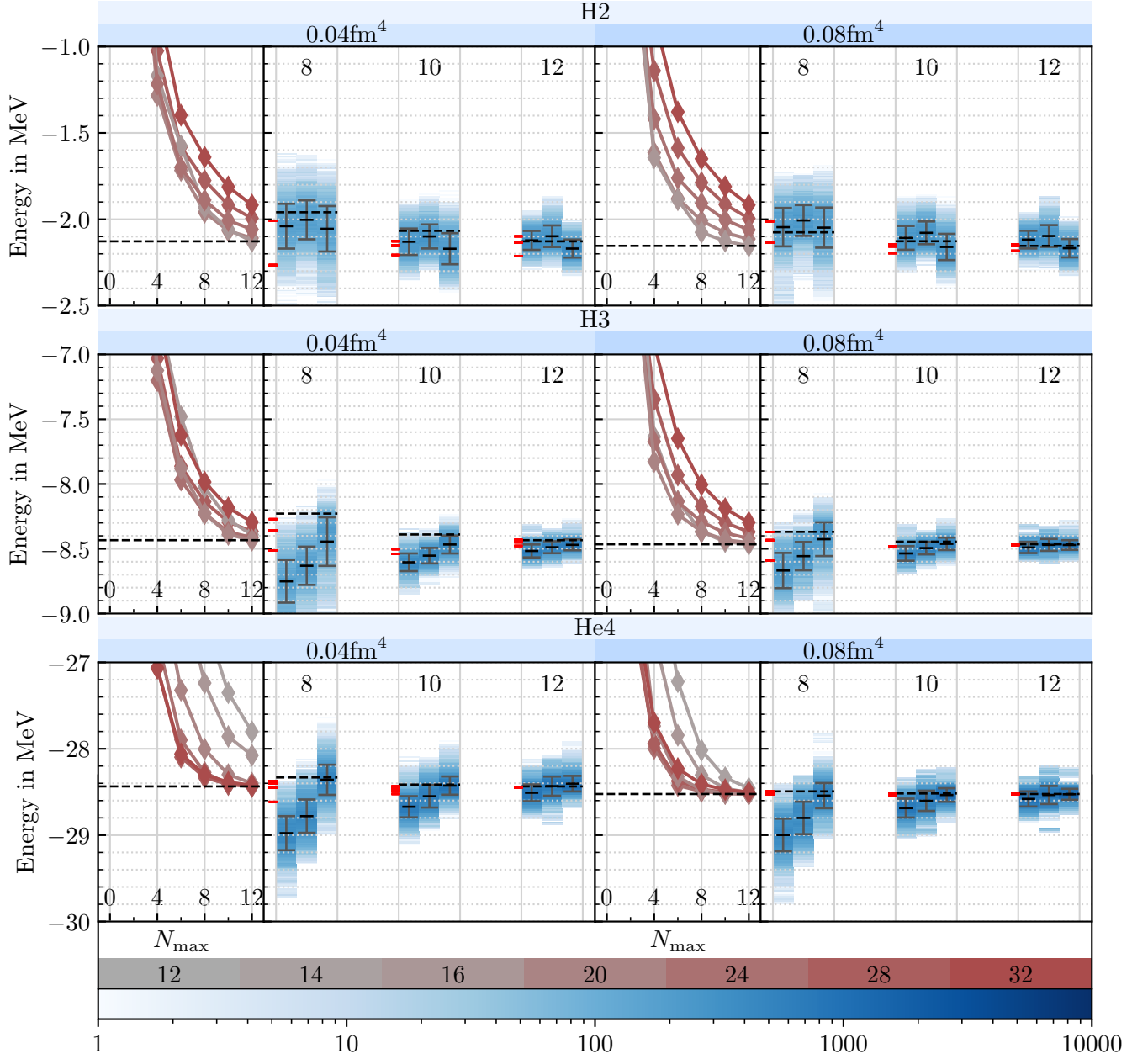


Figure 5.2: Evaluation of our training modes on the nuclei ^2H , ^3H and ^4He , using a semi-local momentum space regulated $N^2\text{LO}$ interaction with two-body interactions and a cutoff at 450 MeV [14]. The shown training modes are, in order from left to right, the basic training mode for comparison, the N_{max} -limitation training mode and the SRG-filter training mode. For each nucleus and each flow parameter, the NCSM sequences are shown on the left (the different frequencies are colored respectively to the legend, which shows the frequencies $\hbar\Omega$ in MeV) and the extrapolations for a given maximum N_{max} on the right. For each maximum N_{max} , the variational boundary is shown as a dashed line, and the classical extrapolations are shown as red ticks.

Figure 5.2 displays the evaluation of the three nuclei. Each histogram at every N_{max} is now divided into three separate histograms, showcasing the evaluation result for the different training modes. In each stack of

histograms, the leftmost is the basic training mode already introduced in chapter 3 and assessed in chapter 4. The histogram in the center corresponds to our first training mode, which restricts the N_{\max} in the training process to at most 24. The rightmost histogram is our second introduced training mode, which restricts the type of training interactions to SRG evolved interactions with a flow parameter of at least 0.04 fm^4 .

We again begin by looking at the nuclei ^3H and ^4He . For both training modes, we see that the predictions get closer to the variational boundary, which means that the training modes yield overall better predictions, since the sequences are already converged enough. Interestingly, the two training modes seem to only impact the accuracy of the prediction. The uncertainty of both training modes are comparable to the uncertainty of the prediction by the basic extrapolation framework.

If we compare the two training modes for those nuclei, we see that the N_{\max} -limitation mode is less impactful than the SRG-filter mode. This can be explained by the fact that some of the sequences of the ^3H and the ^4He nucleus are already converged enough. Since we explicitly remove bare Hamiltonians in the SRG-filter mode, the training set gets limited to a smaller size of fast converging sequences, thus training the net better to those cases. This can especially be seen in the ^4He nucleus, where there are sequences which have a very strong convergence rate. Here, the SRG-filter mode results lie very close to the variational boundary. The limitation on the N_{\max} values also have a positive impact on the evaluation. This may indicate that the networks have learned from the constant sequences of high N_{\max} values to linearly extrapolate the results up to a certain point, which results in the predictions being unreasonably low for low N_{\max} sequences, as is the case for our basic extrapolation framework.

For the nucleus ^2H , our basic framework showed a totally different systematic behavior than for the nuclei ^3H and ^4He . This is also the case for both new training modes. Especially the N_{\max} training mode seems to extrapolate not far enough, such that the predictions do not conform to the variational boundary at $N_{\max} = 12$. Even though the N_{\max} -limitation mode is thought to handle unconverged sequences better than the unmodified training mode, it predicts higher ground-state energies for both SRG flow parameters and for all N_{\max} values. This can be explained by the fact that the ^2H sequences are not only converged until $N_{\max} = 12$, but also converge quite slowly. This leads to a very broad range of energy values at $N_{\max} = 12$ for all the frequencies, but the sequences themselves are relatively flat, leading to a higher prediction of the N_{\max} -limitation training mode. The SRG-filter training mode seems to perform better than the basic training mode. Even though this training mode still produces unreasonably high predictions for $N_{\max} = 8$, the predictions for all $N_{\max} \geq 10$ are lower than the variational boundary at $N_{\max} = 12$. This is a great improvement over the other two training modes.

The comparison between the two different SRG flow parameters yield the same results as in the previous chapter, where we have considered our training without further modifications. We can see that the network predictions do not change much when evaluating the higher srg flow parameter. This is especially interesting in the case of ^4He , where the difference between the NCSM sequences is the biggest. At $\alpha = 0.04 \text{ fm}^4$, the lower sequences of $\hbar\Omega = 12 \text{ MeV}$ and 14 MeV are converging fast, but they have not reached a sufficient degree of convergence, such that the values at $N_{\max} = 12$ are still spread far apart. For $\alpha = 0.08 \text{ fm}^4$, those sequences have converged sufficiently until $N_{\max} = 12$. We can see that the networks react very consistently between the different sequences.

5.4 Conclusion

In this chapter, we have extended our basic extrapolation framework by modifying the training process of the neural networks in two distinct ways to see how they affect the extrapolation of ground-state energies. In a first modification, we have decided on only training the networks by using sequences up to $N_{\max} = 24$. In a second modification, the networks were only trained by NCSM sequences coming from SRG evolved Hamiltonians with a flow parameter α of 0.04 fm^4 or higher. We have discussed these *training modes* by evaluating the nuclei ^2H , ^3H and ^4He and comparing the results to our basic training mode from chapter 4.

To classify our training modes, we first provide two important aspects that those training modes have to show. Firstly, they should result in extrapolations which are "better" in some way when compared to the unmodified training process. For example, they should either result in a smaller uncertainty or in a different prediction which is more reasonable given the NCSM sequences. Secondly, a modification of the training set should result in consistent extrapolations across different scenarios, such as unconverged NCSM sequences and NCSM sequences with a low convergence rate.

Keeping these conditions in mind, we conclude that the SRG-filter training mode so far provided the most reasonable results across all nuclei. The unmodified training mode, as well as the N_{\max} -limitation training mode, predicted unreasonably low ground-state energies for the nuclei ^3H and ^4He yet unreasonably high energies for ^2H . For all of these nuclei, the SRG-filter training mode provided more reasonable predictions. Also, the SRG-filter training mode provided the most consistent results, since it can handle the more converged and more unconverged sequences of ^3H and ^4He respectively, as well as the slow converging sequences of ^2H .

The N_{\max} -limitation filter provided more reasonable results for ^3H and ^4He , but too high values for ^2H , even higher than the unmodified training mode, making it the most inconsistent and unreasonable among the two training modes discussed.

Considering all of the above, it seems that the SRG-filter training mode provides the most promising results. In order to reach a final verdict about the two training modes, we will test how they behave in a scenario which is closer to a real use-case of an extrapolation in chapter 7, where we evaluate the NCSM calculations for the nucleus ^6Li .

6 Difference based extrapolation framework

In our basic extrapolation framework, we trained networks on predicting the exact ground-state energy, given a few sequences of absolute energy values. This method is inherently dependent on the absolute values of the energy, since the networks fundamentally consist of linear maps from one layer to the next. Nuclear ground-state energies have a very broad range, which we already saw in our inspected nuclei. For example, ^2H has a ground-state energy of approximately -2.1 MeV and ^4He has a ground-state energy of approximately -28.4 MeV, based off the NCSM calculations shown in Figure 5.1 and Figure 5.2. As a result, we cannot confidently exclude the possibility of the absolute ground-state energies influencing the network predictions.

In this chapter, we will analyze the influence of the absolute energies on the network prediction by implementing a different extrapolation method based on our previous framework, which will work by extrapolating the differences of the energy sequences.

6.1 New framework

In order to implement our difference-based extrapolation framework, we have to think about how we have to change the generation of the input data, the network structure as well as the extrapolation process of our basic extrapolation framework.

We want to base our network training on the same data set as the basic framework. The networks now should get the differences of consecutive ground-state energies in the sequences as an input. To achieve this, we generate the same formatted input sequences out of the raw NCSM calculations, using the inflation algorithm described in subsection 3.1.3. The formatted input data thus consists of four consecutive energy values for three oscillator frequencies, as well as a limit which will be used to compute the prediction target of our networks. To turn those sequences into our input difference sequences, we calculate the three differences between the four energy values. When the networks get input data in the form of energy differences, they cannot possibly predict an absolute ground-state energy from them. This means that the prediction target will also have to be modified to a difference. It is natural to take the difference between the last value of the energy sequence and the limit as a target. However, this leads to a problem, as the limit is the same for the three oscillator frequencies for a given nucleus and interaction, but the last values of the energy sequences can vary depending on the frequency. Because the three difference sequences are all put into a network simultaneously, we have to find a single common extrapolation target. For this, we decide to take the difference between the mean of the last values for all three sequences and the limit.

Note that, since we now only have three direct inputs available, which consist of the differences of the four absolute values, the network structure has to be adjusted to be compatible with this data. For the input layer, we have $L^{(1)} = 3 \times 3 = 9$ input neurons. Furthermore, we reduce some of the hidden neurons in the hidden layers to accommodate the reduced input neurons. We empirically decide on $L^{(2)} = 18$, $L^{(3)} = 12$. The single

output neuron stays the same. We could have chosen to keep the network structure the same, but this would require to input four differences of five energies, which would mean that those networks got more information about the sequences. To ensure comparability, we chose to keep the information the same but adjust the network structure.

Further modifications include the adjustment of our sequence shifting. In our basic framework, we decided to shift the formatted sequences by a random amount of $[-10 \text{ MeV}, 10 \text{ MeV}]$ to counter the dependency on absolute values. They are no longer needed, as they would cancel out when calculating the differences. Also, the threshold for deeming a network valid is reduced to 0.1 MeV , as the energy differences which are input are generally much smaller than the absolute values.

To compensate for the shift in the training set, we also have to modify the evaluation of a network. The evaluation will work the same way as in our basic framework, but now, the final prediction is calculated by adding the mean of the last energy values back onto the network prediction.

6.2 Results and Comparison

Using our newly constructed difference-based extrapolation framework, we can now do the same analysis on our three nuclei ^2H , ^3H and ^4He using the same interaction as used earlier. Even though the difference-based framework computes a difference prediction internally, the final evaluation will still base off the same absolute energy differences and result in an absolute energy prediction. This means that we can apply our different evaluation modifications from chapter 5 and discuss the influences of them on the difference-based evaluation. Furthermore, using the same nuclei allows us to compare the difference-based extrapolation and the absolute value extrapolation.

The results of our difference-based extrapolation is shown in Figure 6.3. The final extrapolated values and their uncertainty are shown in Table 6.1 as well for each N_{max} and each SRG flow parameter.

Looking at all three nuclei, we can see that the uncertainty of all training modes is drastically reduced. Especially for higher N_{max} , the predictions get very precise. Furthermore, the absolute value of the predictions are generally way closer to a reasonable limit. This can be seen for the nuclei ^4He and ^3H , where the networks trained with the unmodified or the N_{max} -limitation training mode in the absolute-based extrapolation framework predicted unreasonably low values for all N_{max} . This could be caused by the fact that the absolute values are purged from the resulting differences, such that the absolute value of the sequences do not matter in the extrapolation process. This leads to a focus on the convergence of the sequences. When the sequence converges, the differences tend towards zero, such that evaluation yields a prediction for the ground-state energy close to the last values in the inputted sequences. There is a disadvantage of the difference-based extrapolation in the nucleus ^4He for a flow parameter of $\alpha = 0.04 \text{ fm}^4$. Here, the sequences converge fast, but for the small values of N_{max} given, the sequences are cut off for the lower oscillator frequencies of $\hbar\Omega = 12 \text{ MeV}$ and 14 MeV . This leads to the network receiving a broad range of differences, resulting in a greater uncertainty for the ground-state energy prediction. We can also see that this effect gets better for $\alpha = 0.08 \text{ fm}^4$, where even the sequences for the lower oscillator frequencies are now visibly more converged to the limit.

Table 6.1: Extrapolation results in MeV of the difference-based framework for the nuclei ^2H (a), ^3H (b) and ^4He (c). For each interaction characterized by the flow parameter $\alpha = 0.04 \text{ fm}^4, 0.08 \text{ fm}^4$, the final extrapolation results for the given N_{max} value is shown. Here, (1) is our basic extrapolation without further modifications of the training process, (2) is the N_{max} -limitation training mode, (3) is the SRG-filter training mode.

(a)						
α	0.04 fm^4			0.08 fm^4		
N_{max}	8	10	12	8	10	12
(1)	-2.066(65)	-2.144(25)	-2.129(22)	-2.054(69)	-2.113(28)	-2.126(23)
(2)	-2.080(55)	-2.146(23)	-2.127(20)	-2.064(56)	-2.110(26)	-2.124(22)
(3)	-2.068(55)	-2.153(38)	-2.149(34)	-2.062(41)	-2.124(32)	-2.158(36)

(b)						
α	0.04 fm^4			0.08 fm^4		
N_{max}	8	10	12	8	10	12
(1)	-8.453(99)	-8.528(44)	-8.470(26)	-8.410(90)	-8.465(35)	-8.460(25)
(2)	-8.461(82)	-8.526(36)	-8.468(25)	-8.422(74)	-8.464(31)	-8.457(23)
(3)	-8.390(64)	-8.493(28)	-8.466(26)	-8.408(45)	-8.460(26)	-8.466(25)

(c)						
α	0.04 fm^4			0.08 fm^4		
N_{max}	8	10	12	8	10	12
(1)	-28.502(397)	-28.497(212)	-28.405(95)	-28.520(175)	-28.535(64)	-28.529(25)
(2)	-28.399(319)	-28.454(177)	-28.391(80)	-28.522(143)	-28.537(52)	-28.530(21)
(3)	-28.377(294)	-28.408(139)	-28.353(54)	-28.507(65)	-28.521(26)	-28.527(12)

Another effect of the difference-based extrapolation is that the different training modes now generally predict a ground-state energy in the same range. However, now the impact of the training modes has shifted from the absolute value of the prediction into the uncertainty of the prediction. This can be seen especially for the SRG-filter training mode, where the uncertainty generally gets smaller for the nuclei ^3H and ^4He , but larger for ^2H , which can be attributed to the slow convergence of ^2H sequences. For the first and second training mode of the ^2H nucleus, we can also observe that the difference-based framework produces a smaller uncertainty than the range of the classical extrapolations.

For ^2H and ^3H , we can also see that the predictions for the ground-state energy are a bit higher for the SRG flow parameter of 0.08 fm^4 than 0.04 fm^4 .

Note that the difference-based extrapolation framework is not completely independent of the absolute values of the sequences, since the prediction target for the network training is calculated using the average of the absolute values for the highest N_{max} . This can also result in a dependency on the spread of the different sequences for the evaluated maximum N_{max} .

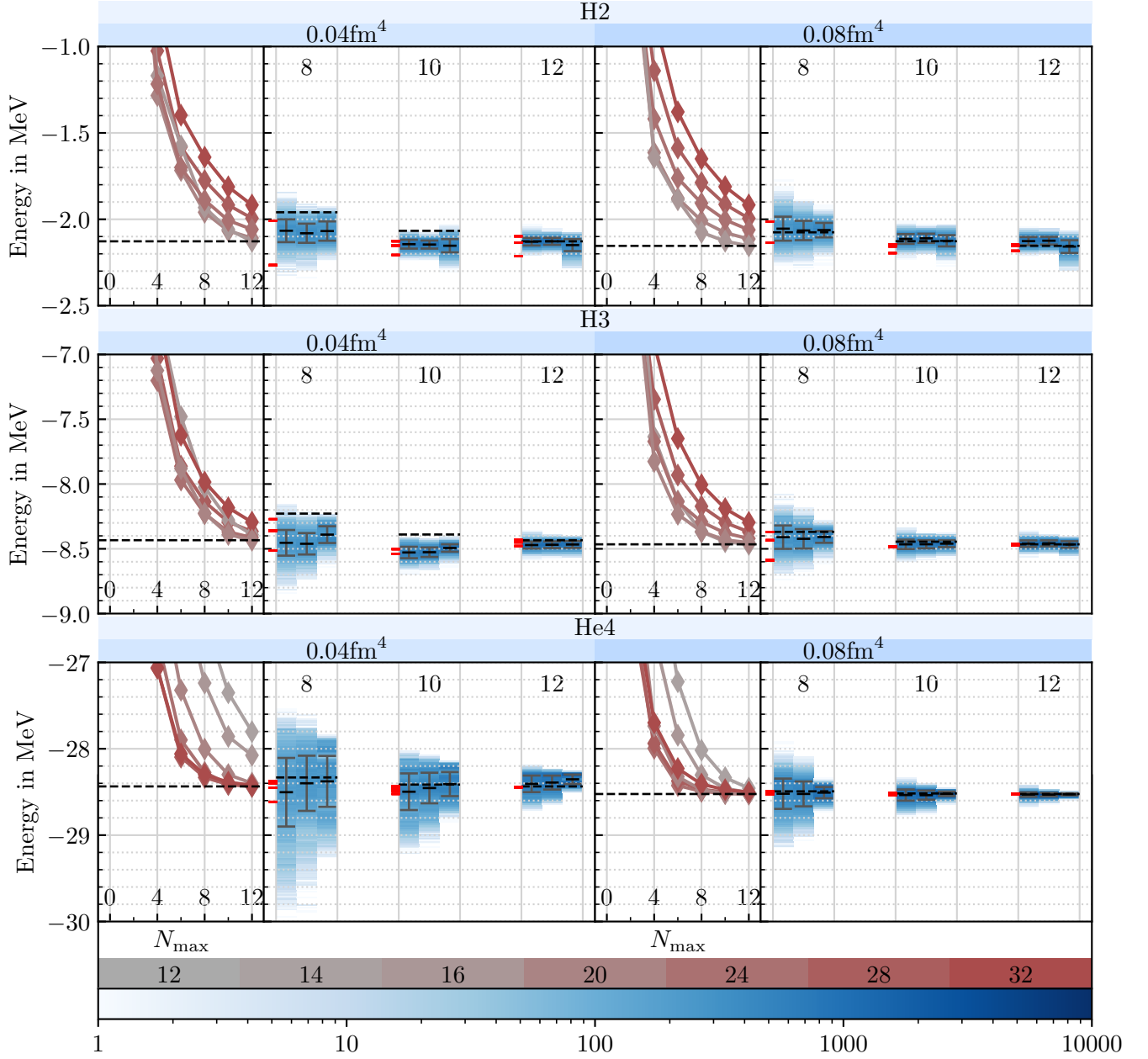


Figure 6.3: Evaluation of our training modes on the nuclei ^2H , ^3H and ^4He in the difference-based extrapolation framework. The shown training modes are, in order from left to right, the basic training mode for comparison, the N_{max} -limitation training mode and the SRG-filter training mode. For each nucleus and each flow parameter, the NCSM sequences are shown on the left (the different frequencies are colored respectively to the legend, which shows the frequencies $\hbar\Omega$ in MeV) and the extrapolations for a given maximum N_{max} on the right. For each maximum N_{max} , the variational boundary is shown as a dashed line, and the classical extrapolations are shown as red ticks.

6.3 Conclusion

In this chapter, we have introduced a second framework for extrapolating ground-state energies using neural networks. Here, we did not directly input the absolute energies into the network in return of an absolute energy prediction, but calculate the differences of consecutive energies in our sequences to input into the network. The output of the network was then trained on a prediction target calculated using the average distance of the last values in the formatted sequences to the actual limit. As a consequence, the evaluation process had to reconstruct the absolute energy prediction by adding the average energy for the highest N_{\max} back onto the networks prediction of the difference.

In order to further examine the training modifications done in the previous chapter, we have evaluated our three training nuclei ^2H , ^3H and ^4He using the additional training modifications.

We have found that the difference-based extrapolation framework generally produced predictions with a smaller uncertainty. Further, the extrapolations generated were more reasonable than those generated by our previous absolute-based framework, as they were generally too low for small N_{\max} values.

As for the training modes, we have found that they do not impact the absolute value of the prediction as much as in the absolute-based framework. Here, they have impacted the uncertainties of the predictions, where the SRG-filter training mode generally impacted the improvement of the uncertainties the most. Before we reach a final decision on the two different frameworks and the three training modes, we will test how they affect the extrapolation of the nucleus ^6Li in the next chapter, which will be closer to a real use-case of extrapolation methods.

7 Extrapolation of ${}^6\text{Li}$

In a final application of both the absolute- and the difference-based extrapolation frameworks, we want to evaluate the nucleus ${}^6\text{Li}$ by using the same interaction as in the previous chapters.

This application will be special because ${}^6\text{Li}$ is not present in the training set. Furthermore, ${}^6\text{Li}$ has a higher mass than all of the nuclei considered before. As a result, we can regard ${}^6\text{Li}$ as a nucleus for which NCSM calculations can not be done until a sufficiently high N_{max} , such that extrapolation of the ground-state energy is needed. The evaluation of the ${}^6\text{Li}$ extrapolation can thus be considered as a first test of our extrapolation frameworks in a real use-case scenario.

7.1 Results

To see how our previous extrapolation methods impact the evaluation of ${}^6\text{Li}$, we will evaluate the nucleus using both the absolute and the difference-based extrapolation framework and all the different training modes introduced in chapter 5. For the evaluation, NCSM sequences of ${}^6\text{Li}$ are only available up to $N_{\text{max}} = 10$. The evaluation will thus be done for a maximum N_{max} of 8 and 10. The resulting evaluation can be seen in Figure 7.1.

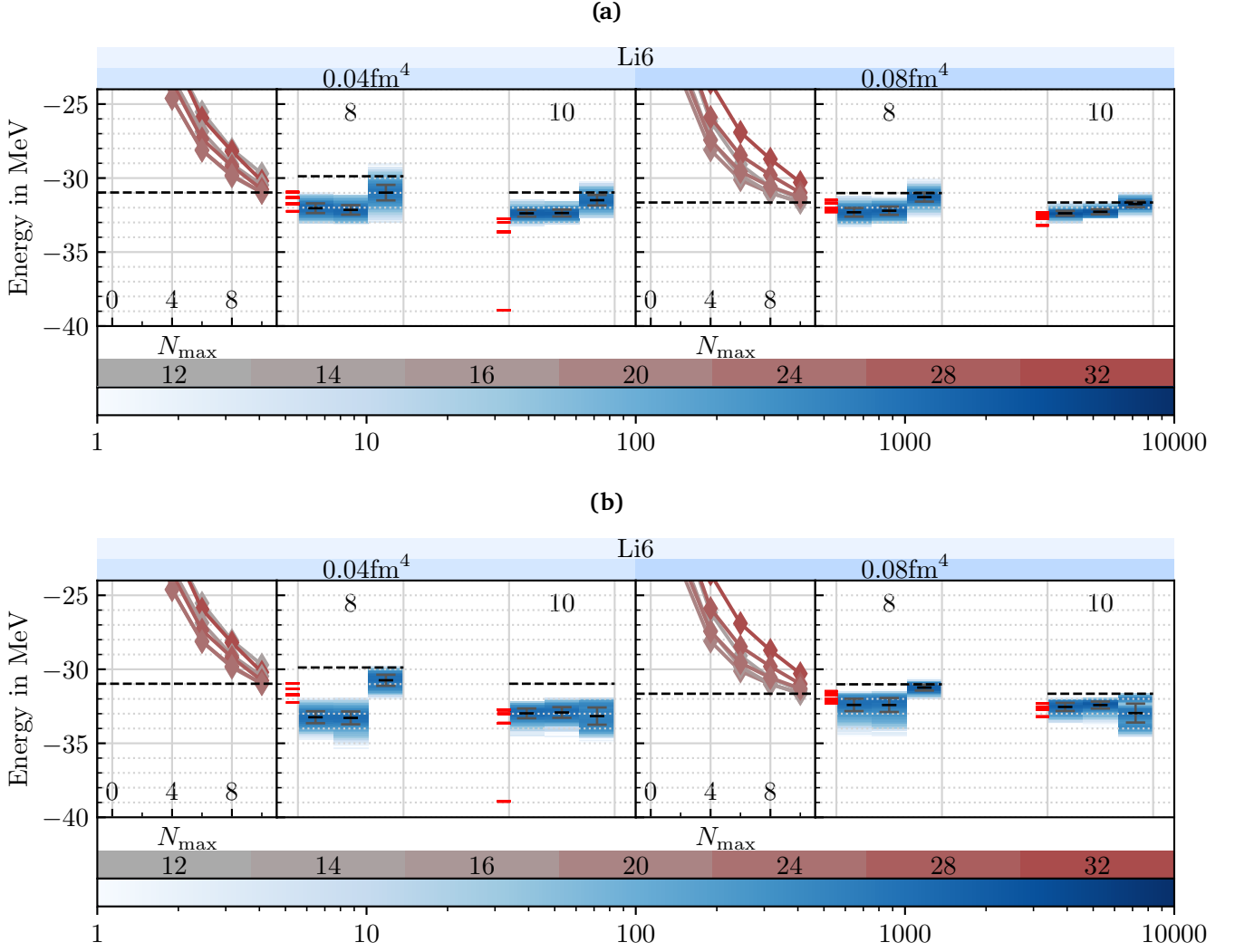
First, we take a look at the absolute-based extrapolation. Here, the unmodified training mode results in a plausible extrapolation of the given NCSM sequences, satisfying the variational boundary condition even for the highest N_{max} of 10. Also, the resulting predictions for the ground-state energy are very consistent across the different N_{max} values, in contrast to the classical extrapolations, which provide a prediction of bigger value than our framework for $N_{\text{max}} = 8$ but of lower value than our framework for $N_{\text{max}} = 10$.

In contrast to the unmodified training mode, the N_{max} -limitation training mode provides no significantly different result. The results of this training mode are very inconsistent, as they predict a lower value for $N_{\text{max}} = 8$ and $\alpha = 0.04 \text{ fm}^4$ than the unmodified training mode, but a higher value for all other N_{max} . This is the behavior we already encountered in chapter 5.

Surprisingly, the SRG-filter training mode produces a very unreasonably high extrapolation for all N_{max} values. The extrapolations produced by this training mode are very close to the variational boundaries of $N_{\text{max}} = 10$, but since the NCSM sequences of ${}^6\text{Li}$ are not sufficiently converged until this N_{max} value, the variational boundary does not provide a good comparison for the extrapolated ground-state energy.

The extrapolations for the flow parameter 0.08 fm^4 have a smaller uncertainty than those for the flow parameter 0.04 fm^4 . This is not an effect that we have seen that strong in the evaluation of the three nuclei used in training in chapter 5. This may originate from the more converged sequences that are present in the $\alpha = 0.08 \text{ fm}^4$ data set, which also explains the more consistent classical extrapolations across the N_{max} values, which are usually very dependant on the degree of convergence of the sequences.

Figure 7.1: Evaluation results for the ${}^6\text{Li}$ nucleus using the absolute (a) and the difference-based extrapolation (b). The shown training modes are, in order from left to right, the unmodified training mode for comparison, the N_{max} -limitation training mode and the SRG-filter training mode. For each nucleus and each flow parameter, the NCSM sequences are shown on the left (the different frequencies are colored respectively to the legend, which shows the frequencies $\hbar\Omega$ in MeV) and the extrapolations for a given maximum N_{max} on the right. For each maximum N_{max} , the variational boundary is shown as a dashed line, and the classical extrapolations are shown as red ticks.



For the difference-based extrapolations, the same observations about the unmodified and the N_{max} -limitation training modes can be made. However, we cannot observe the general improvement of the uncertainties in contrast to the absolute-based framework. In fact, the uncertainties are bigger in the evaluation using difference values. In chapter 6, we have found that the difference-based extrapolation produces a larger uncertainty for sequences, which converge uniformly, but are cut off at the higher N_{max} values. This effect seems to outweigh the advantages that the difference-based framework offers, resulting in an increased uncertainty for ${}^6\text{Li}$. Also, the extrapolation for the SRG-filter training mode is very inconsistent with the observations we made for the absolute-based framework. Here, the training mode produces unreasonably high values for $N_{\text{max}} = 8$ but unreasonably low values for $N_{\text{max}} = 10$.

7.2 Conclusion

In this chapter, we have applied both of the previously presented absolute and difference-based extrapolation frameworks to the nucleus ${}^6\text{Li}$. For that, we have simulated a real use-case scenario by using NCSM calculations up to $N_{\text{max}} = 10$.

Using the observations obtained from the ${}^6\text{Li}$, we can decide on a final choice for the training mode and framework, which will be used to produce a final prediction for the ${}^6\text{Li}$ nucleus.

Now that we have applied the different training modes and the two framework to a real use-case, we are ready to formulate our final verdict about the frameworks as well as the training modes. For the training modes, we have already formulated two conditions that are necessary. They have to provide more reasonable predictions than the unmodified training mode and be consistent. In the previous chapters, we have seen that the N_{max} -limitation training mode yielded inconsistent predictions for three training nuclei. This conforms to our observations in the evaluation of ${}^6\text{Li}$, where the training mode produced reasonable but inconsistent results. The SRG-filter training mode was deemed the most promisable one in the evaluation of the three training nuclei. Unfortunately, this training mode resulted in the predictions of very unreasonably high values for the ${}^6\text{Li}$ nucleus.

Each of the training modes provided good results in some cases, but worse results than the unmodified training in other cases. Based on this inconsistency, we decide that the unmodified training mode is the most stable across all different scenarios, which is why we will be using this mode in our final result for ${}^6\text{Li}$.

Now we come to a verdict about the two different extrapolation frameworks. The first framework consisted of the extrapolation of sequences of absolute energy values, while the second framework consisted of extrapolating difference sequences. While we observed that the difference-based framework produced a little smaller uncertainty for ${}^6\text{Li}$, we saw in the previous chapters 5 and 6, that the difference-based framework produced more reasonable predictions and generally smaller uncertainties as the absolute-based framework. For that reason, we decide on the difference-based framework as the generally better method of extrapolating ground-state energies.

Using the difference-based framework as well as the unmodified training process, we can state a final extrapolation result for the ${}^6\text{Li}$ nucleus. Since we have two maximum N_{max} values available, we have to decide on one for the final extrapolation. In a real use-case, one would decide on the maximum N_{max} available in order to input the maximum amount of information about the sequence limit into the network, as the higher N_{max} values are more converged. Using this, we decide on the extrapolation result of $N_{\text{max}} = 10$ as the final prediction of the ${}^6\text{Li}$ ground-state energy value, which is given by $-32.976(329)$ MeV for the interaction evolved with a flow parameter of 0.04 fm^4 and $-32.542(272)$ MeV for the flow parameter of 0.08 fm^4 . The histograms of those extrapolations can be found in Figure 7.3.

For the computation of the final prediction, as well as the error estimate, we assumed that the network prediction are normally distributed. The distribution of the predictions for the flow paramter α of 0.04 fm^4 show a good compliance to the shown normal distribution. However, we also see that the distribution is slightly skewed towards lower energy values. This effect is amplified in the distribution of predictions for $\alpha = 0.08 \text{ fm}^4$, where there is a sharp decline in predictions for high energies bigger than -32.1 MeV, but a higher amount of predictions for lower energies. As a result, the assumption of a normal distribution leads to a lower prediction than the most guessed value. To generate a more meaningful prediction and uncertainty, we could assume a probability density that takes this skewness into account.

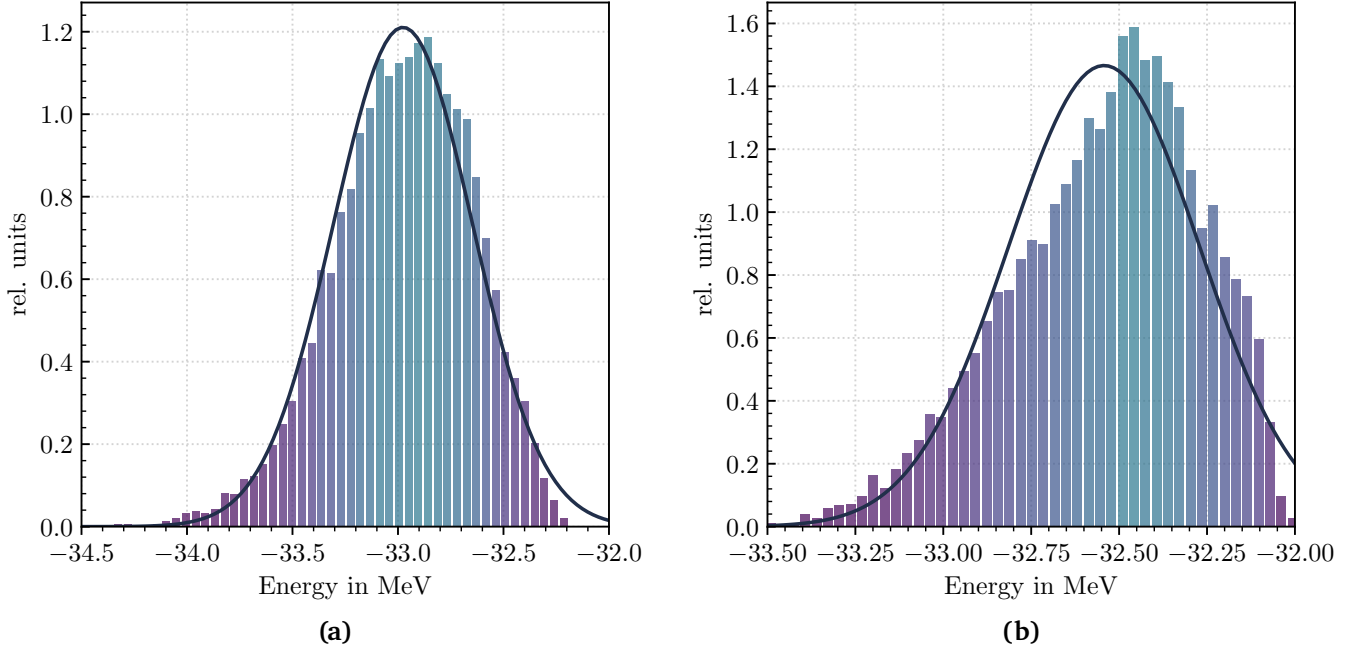


Figure 7.3: Histograms for the final extrapolation of ${}^6\text{Li}$ for the different SRG flow parameters of 0.04 fm^4 **(a)** and 0.08 fm^4 **(b)**. The bar heights have been normed such that the total area of them is 1. The final extrapolation used the difference-based framework together with an unmodified training process.

8 Outlook

In this final chapter, we want to look at a few ideas that are based on our extrapolation framework that could further improve the extrapolation of ground-state energies using neural networks.

In a first idea, we want to look at another training modification which will work by dynamically filtering frequencies.

In a second idea, we demonstrate the effect of preselecting frequencies on the extrapolation.

8.1 Physical Frequency Selection

The single-particle harmonic oscillator basis is not asymptotically good to approximate the states of nuclei, as harmonic oscillator wave functions fall off like a gaussian function, but bound states in a potential well typically fall off like an exponential function [4]. Furthermore, the composition into the harmonic oscillator Slater determinants is dependant on a free parameter for the oscillator frequency. This oscillator frequency also defines the oscillator length and thus the spatial resolution for the basis states.

As we have already seen, the dependence on the oscillator frequency results in different convergence rates for our NCSM sequences. In fact, NCSM calculations using harmonic oscillator Slater determinants typically show a parabolic convergence behaviour with respect to the used oscillator frequency. This is shown in Figure 8.1 for the nucleus ^{16}O . As a consequence, there is a special frequency, for which an optimal convergence rate is to be expected. In Figure 8.1, this minimum frequency is given by $\hbar\Omega = 20$ MeV.

In order to capture those different convergence rates, we want to restrict the sequences used to either side of the minimum, resulting in two training modes. In the first training mode, only sequences smaller than the frequency minimum are used, such that the network will get more information of the long range

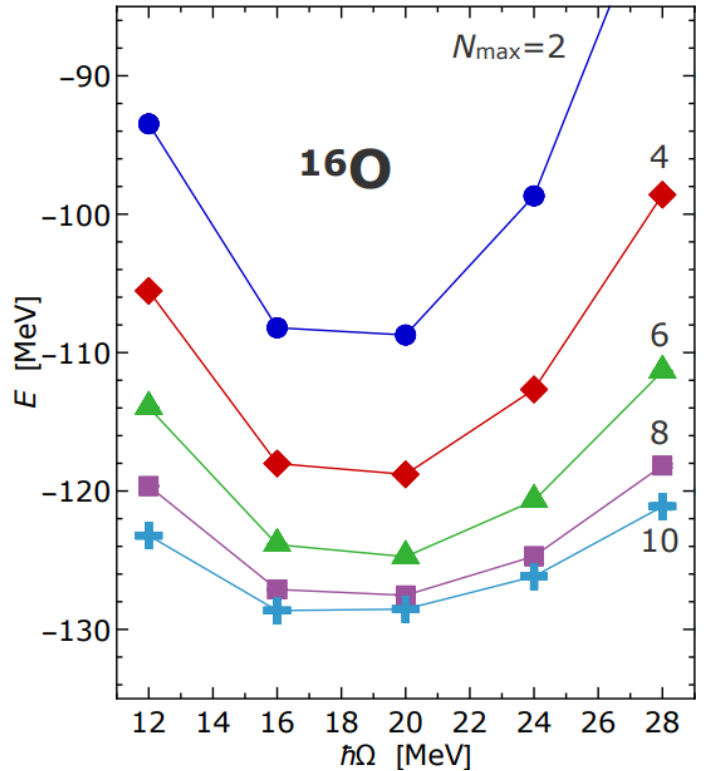


Figure 8.1: NCSM calculations for ^{16}O in dependency on the used oscillator frequency [4].

behaviour of the nucleonic interaction. As a result, the networks have to model the short range behaviour. Conversely, in the second training mode, the networks will be trained on the short range behaviour by restricting the frequencies to the ones higher than the minimum, thus modeling long range behaviour.

To implement this, we dynamically compute the frequency, for which the lowest value is achieved at $N_{\max} = 8$ for each NCSM calculation. This frequency will be used to filter the frequencies for the training. In both cases, we include the frequency minimum, as that is the frequency with the best convergence behaviour.

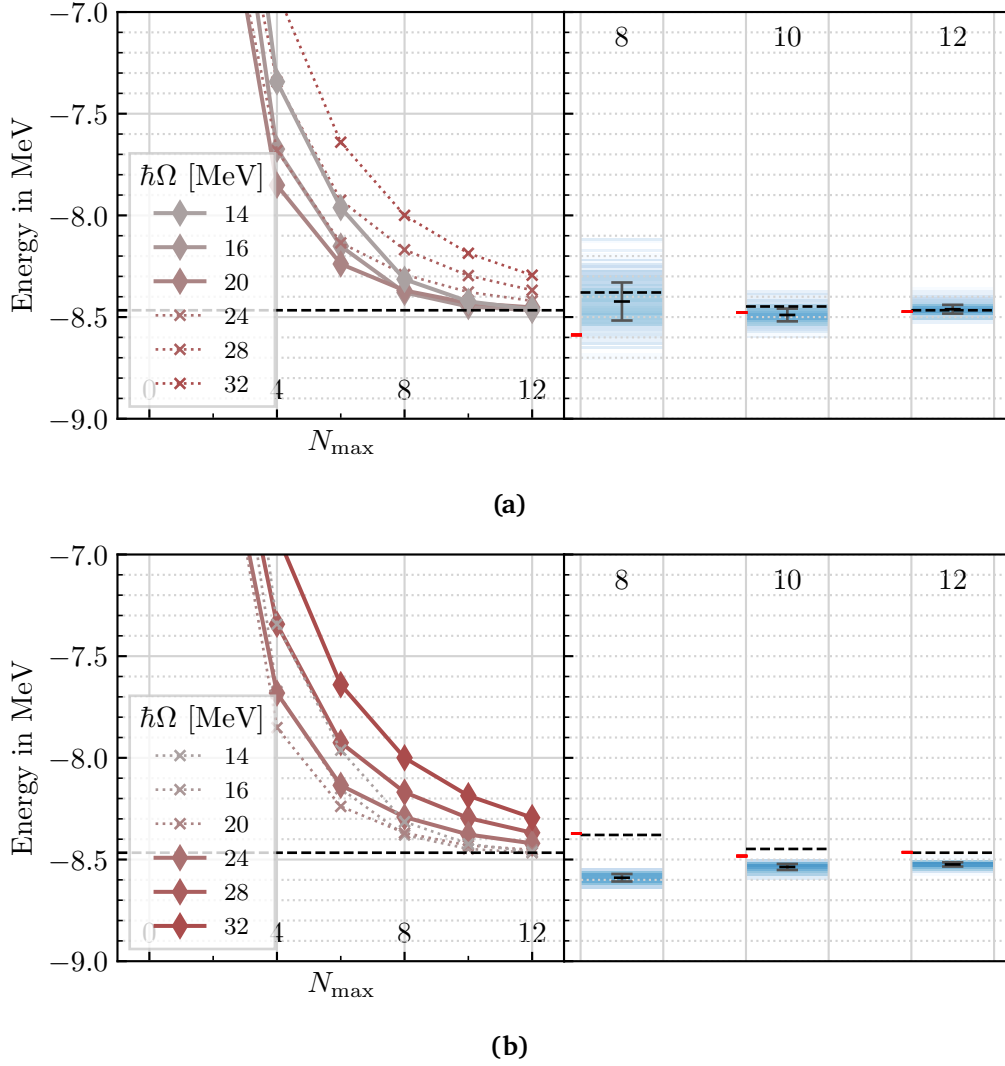


Figure 8.2: Evaluation of the frequency filter training mode on ^3H with the interaction specified above. The frequencies used in the evaluation are shown with solid lines. To achieve a better comparability with the previous extrapolations, the frequencies not taken into account are shown as dotted lines. In (a), the networks were trained on the low frequencies, thus modeling the short range behaviour. In (b), the networks were trained on the high frequencies, thus modeling the long range behaviour.

We look at a single example of the ^3H nucleus, using a semi-local momentum space regulated N^4LO interaction with two-body interactions and a cutoff at 450 MeV alongside a SRG evolution with a flow parameter of 0.08 fm^4 . Since there are six frequencies available, the evaluation for the long range training mode will include

the three lowest and the evaluation for the short range mode will include the three highest frequencies. In Figure 8.2, the evaluation results are seen for the difference-based extrapolation framework.

In the case where the networks have been trained on the low frequencies, the prediction for $N_{\max} = 8$ is too high. Also, the uncertainty is still very large, which stabilizes along with the value of the predictions for $N_{\max} \geq 10$, where the range of the predictions get much smaller. If we look at the evaluation data at $N_{\max} = 8$, we see that the sequence for $\hbar\Omega = 14$ MeV is still very steep compared to the other two sequences. As we have already seen, this broad range between different slopes result in a larger range of predictions. At $N_{\max} = 10$ the sequences converge much more uniformly, leading to the more precise predictions.

In contrast to the training with lower frequencies, the behaviour predictions of the training with higher frequencies look more consistent across the different N_{\max} . For every N_{\max} , the networks predict ground-state energies which are too low, but correct those to higher values with rising N_{\max} , similar to the observations made in the absolute-based framework. Here, the sequences of the higher frequencies converge much more uniformly, which is the reason we cannot confidentially attribute the precision of the networks prediction to the fact of them being only trained on the high frequencies. To further analyze the behaviour of the networks when trained only on short or long ranges, more investigations need to be done on nuclei which show different ranges of slopes for the high and for the low frequencies.

8.2 Manual Frequency Selection

In both of our extrapolation frameworks, we deliberately considered every available frequency to predict the ground-state energy, since we wanted to obtain a more general extrapolation. This "black-box" behavior could lead to worse predictions, but does not require a hand-selection of a subset of the sequences for extrapolation. Also, this would require a metric on what sequences to choose. For ground-state energies, the convergence is monotonic and decreasing, such that we *could* choose such a metric, where we choose the three sequences, for which the value at the highest N_{\max} are the lowest among all sequences available. We want to demonstrate this frequency preselection idea by evaluating the ${}^2\text{H}$ nucleus using the same interaction as in the previous chapters with a flow parameter of 0.08 fm^4 .

In Figure 8.3, the results are shown. If we compare these results to the results of the unmodified training in Figure 6.3 for the 0.08 fm^4 interaction of ${}^2\text{H}$, we can see that the frequency preselection significantly improves the absolute value, as well as the uncertainty of the prediction for the N_{\max} values of 8 and 10.

Of course, manually selecting the sequences which will be used in evaluation with a predefined metric limits the extrapolation framework to the types of observables for which this metric can apply. For example, after we have decided to only use the sequences with the three lowest values at the highest N_{\max} , the framework gets limited to ground-state energies. The extrapolation of other observables, that do not show the monotone decreasing convergence behavior of the ground-state energy will therefore not be optimized.

In the end, the decision to use a manual preselection of frequency is a tradeoff between a framework with more precise and more accurate predictions and a framework which is more general.

If one decides to manually preselect frequencies for evaluation, it may be of interest to also develop and analyze different metrics for choosing such sequences. In the case of our ground-state energy extrapolation framework, we chose the lowest sequences as the best sequences to choose. Another possibility would be to choose the three sequences which show the most uniform convergence behavior, as we have observed that this significantly lowers the uncertainty in the predictions.

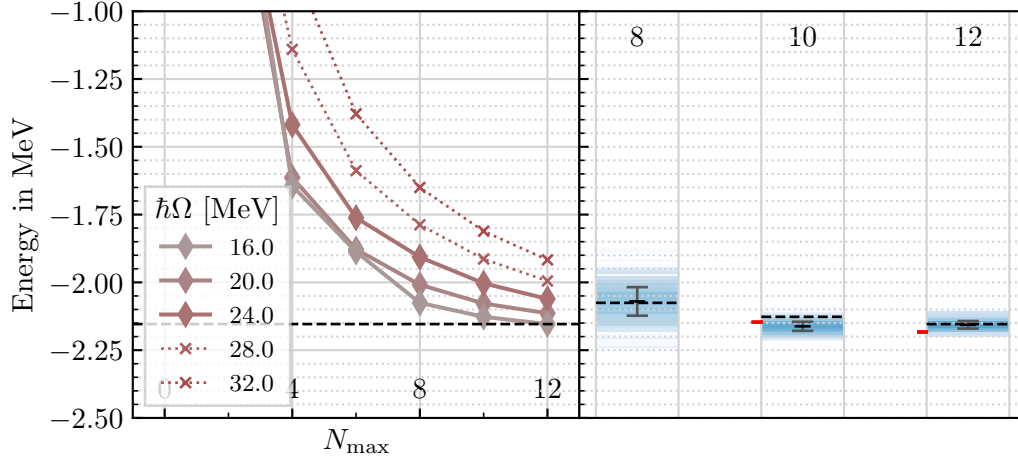


Figure 8.3: Evaluation of the ${}^2\text{H}$ nucleus in the difference-based framework using only the three lowest frequencies. In this case, the frequencies are $\hbar\Omega = 16, 20, 24$ MeV. The frequencies used in the evaluation are shown with solid lines. To achieve a better comparability to the previous extrapolations, the frequencies not taken into account are shown as dotted lines.

9 Conclusion

In this thesis, we have formulated and analyzed two different extrapolation frameworks to predict the ground-state energy out of NCSM calculations using artificial neural networks.

In our first extrapolation framework (absolute-based), the input data consists of three different sequences of four NCSM calculations for consecutive N_{\max} values. The output of the networks is then directly identified as the network prediction for the ground-state energy. In our second extrapolation framework (difference-based), we took the same sequences as described above and computed the difference of consecutive N_{\max} values, resulting in three sequences of three values. In this case, the output is identified as the difference of the prediction and the mean absolute value at the largest N_{\max} . In both frameworks, the networks are trained according to that identification by the nuclei ${}^2\text{H}$, ${}^3\text{H}$ and ${}^4\text{He}$ using chiral interactions of various leading orders and cutoffs for two-body interactions [13] as well as two- and three-body interactions [15]. Furthermore, the Hamiltonians were SRG evolved for flow parameters between 0.04 fm^4 and 0.08 fm^4 . To analyze the frameworks, we used the same nuclei with a semi-local momentum-space regulated N^2LO interaction with two-body interactions and a cutoff at 450 MeV by Maris et. al [14] which was SRG evolved using a flow parameter of 0.04 fm^4 and 0.08 fm^4 .

First, we have analyzed the extrapolation behavior of the absolute-based framework in dependency on the maximum N_{\max} value of the used evaluation sequences and the SRG flow parameter. We have found two different systematic behaviors of the extrapolations for the nuclei ${}^3\text{H}$, ${}^4\text{He}$ and the nucleus ${}^2\text{H}$. The extrapolations for the nuclei ${}^3\text{H}$ and ${}^4\text{He}$ were generally too low and the networks corrected them to higher values with higher N_{\max} values. For ${}^2\text{H}$, the extrapolations were too high at low N_{\max} values and got lower for higher N_{\max} . We have defined the variational boundary condition, which was satisfied for the extrapolations on ${}^3\text{H}$ and ${}^4\text{He}$ but not on ${}^2\text{H}$. The classical extrapolations consisting of exponential function fits generally outperformed our framework on the more converged sequences of ${}^3\text{H}$ and ${}^4\text{He}$, but were comparable for ${}^2\text{H}$.

Before analyzing the difference-based framework, we discussed two different modifications on the training set for the absolute-based framework. The first modification consisted of restricting the training samples to a maximum N_{\max} of 24, while the second modification restricted the used interactions to SRG evolved interactions with a minimum flow parameter of 0.04 fm^4 . We have introduced the conditions of prediction consistency and reasonability to assess the two training modes. We have found that the N_{\max} -limitation training mode did not conform to those two conditions, while the SRG-filter training mode did. In general, the SRG-filter training mode produced the most promising extrapolations for our training nuclei.

After that, we have analyzed the previously introduced training modes in our difference-based framework. We have found that all of the training modes, as well as the unmodified training produced much more reasonable results for the nuclei. Here, the training modes did not significantly improve the value of the prediction but rather the uncertainty. As in the absolute-based framework, the SRG-filter training mode showed the most reasonable and consistent results.

In a real use-case application of our extrapolation framework using the nucleus ${}^6\text{Li}$, which was previously unseen by the networks, we have seen that the SRG-filter training mode produced unreasonably high predictions while the N_{max} -limitation training mode did not provide sufficiently different results to the unmodified training mode. The predictions of the unmodified training produced more consistent results than the classical extrapolations accross the N_{max} values. In a final decision using the insights of the extrapolation for all four nuclei, we have found the difference-based extrapolation framework together with the unmodified training to produce the overall most reasonable results.

Finally, we have looked into two different ideas that could further develop the difference-based framework to yield better results. The first idea included a new training modification, where frequencies were only included in the training if they are bigger or smaller than the frequency which results in the best convergence. The networks were thus trained to model the long or the short range of the nuclear interaction, respectively. We have found the high frequency filter training mode to produce more consistent and more precise results, although this idea has to be further examined for more nuclei. Secondly, we have persued the idea of manually selecting the frequencies for the evaluation data by choosing a selection rule. This results in better extrapolations, but limits the generality of our extrapolation framework.

Bibliography

- [1] Gianina Alina Negoita et al. “Deep learning: Extrapolation tool for ab initio nuclear theory”. In: *Physical Review C* 99.5 (May 2019). ISSN: 2469-9993. DOI: 10.1103/PhysRevC.99.054308. URL: <http://dx.doi.org/10.1103/PhysRevC.99.054308>.
- [2] W. G. Jiang, G. Hagen, and T. Papenbrock. “Extrapolation of nuclear structure observables with artificial neural networks”. In: *Physical Review C* 100.5 (Nov. 2019). ISSN: 2469-9993. DOI: 10.1103/PhysRevC.100.054326. URL: <http://dx.doi.org/10.1103/PhysRevC.100.054326>.
- [3] Cedric T. Wenz. *Applications of Machine Learning in NCSM Calculations*. 2020.
- [4] Robert Roth. “Ab initio Calculations of Nuclear Structure”. Frontiers in nuclear and hadronic physics, Galileo Galilei Institute for Theoretical Physics. 2021. URL: https://www.ggi.infn.it/FNHP/FNHP_2021/abstract_roth.pdf (visited on 10/18/2021).
- [5] Peter Schuck Peter Ring. *The Nuclear Many-Body Problem*. Springer-Verlag Berlin Heidelberg, 1980. ISBN: 978-3-540-21206-5.
- [6] N. Costiris et al. “A Global Model of β^- -Decay Half-Lives Using Neural Networks”. In: *HNPS Proceedings* 15 (Jan. 2007). DOI: 10.12681/hnps.2640.
- [7] S Akkoyun et al. “An artificial neural network application on nuclear charge radii”. In: *Journal of Physics G: Nuclear and Particle Physics* 40.5 (Mar. 2013), p. 055106. DOI: 10.1088/0954-3899/40/5/055106. URL: <https://doi.org/10.1088/0954-3899/40/5/055106>.
- [8] Charu C. Aggarwal. “An Introduction to Neural Networks”. In: *Neural Networks and Deep Learning: A Textbook*. Cham: Springer International Publishing, 2018, pp. 1–52. ISBN: 978-3-319-94463-0. DOI: 10.1007/978-3-319-94463-0_1. URL: https://doi.org/10.1007/978-3-319-94463-0_1.
- [9] Benyamin Ghojogh and Mark Crowley. “The Theory Behind Overfitting, Cross Validation, Regularization, Bagging, and Boosting: Tutorial”. In: (May 2019).
- [10] Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.
- [11] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG].
- [12] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].
- [13] D. R. Entem, R. Machleidt, and Y. Nosyk. “High-quality two-nucleon potentials up to fifth order of the chiral expansion”. In: *Phys. Rev. C* 96 (2 Aug. 2017), p. 024004. DOI: 10.1103/PhysRevC.96.024004. URL: <https://link.aps.org/doi/10.1103/PhysRevC.96.024004>.
- [14] P. Maris et al. “Light nuclei with semilocal momentum-space regularized chiral interactions up to third order”. In: *Phys. Rev. C* 103 (5 May 2021), p. 054001. DOI: 10.1103/PhysRevC.103.054001. URL: <https://link.aps.org/doi/10.1103/PhysRevC.103.054001>.

-
- [15] Thomas Hüther et al. “Family of chiral two- plus three-nucleon interactions for accurate nuclear structure studies”. In: *Physics Letters B* 808 (2020), p. 135651. ISSN: 0370-2693. DOI: <https://doi.org/10.1016/j.physletb.2020.135651>. URL: <https://www.sciencedirect.com/science/article/pii/S0370269320304548>.