Predicting the self-energy of Anderson impurity models with neural networks

A thesis presented for the degree of
Bachelor of Science in Physics

by

Benjamin Czasch

Supervised by

Dr.techn. Markus Wallerberger
and
Univ.Prof. Dr.rer.nat. Karsten Held

Institute of Applied Physics
vienna university of technology
Vienna, Austria
January 9th, 2023

Abstract

Dense feed-forward neural networks are employed to predict the self-energy of a single-impurity Anderson model. The nets vary in size from one to three hidden layers, containing 20 to 180 neurons per layer. Sparse modelling of the self-energy is applied to reduce the computational costs. Various features-sets containing different types of approximations such as second order perturbation theory and approaches based on model simplifications are compared for training the neural networks. The dependence of the generalization error on feature-set and network size is analysed. Results show that approximations based perturbation theory work best as an aid to predicting the self-energy achieving relative errors below the 5% mark, although this difference in performance decreases as the model size increases. Further, the implications of choosing different performance measures to train and evaluate learning algorithms are briefly discussed.

Acknowledgements

First and foremost, I would like to thank *Dr. Markus Wallerberger (TU Wien)* for introducing me to the topic of machine learning via his thought-provoking lecture *Machine Learning in Physics*, where he showed lots of passion and knowledge. I would also like to thank him for being a very helpful and patient supervisor.

Next, I would like to thank *Prof. Dr. Karsten Held (TU Wien)* and the whole *theoretical solid-state physics group* for their comprehensive lectures on quantum mechanics and many-body quantum field theory, which made these topics accessible and laid a strong foundation for constitutive lectures. I would especially like to thank *Prof. Dr. Karsten Held* for his help with some technical aspects of this thesis.

Finally, I would like to mention the Youtube channel of *Prof. Dr. Andrew Mitchell (University College Dublin)*, which further helped me to understand the very basics of quantum many-body theory. Further, the Youtube channel and blog of *Andrej Karpathy* were very helpful in delving deeper into the inner workings of neural networks.

Contents

1 Introduction **1**

2 System and Methods **2**
2.1 Anderson impurity model 2
2.1.1 Symmetries and conservations 3
2.1.2 Analytical solution 3
2.2 Self-energy of the impurity-site 6
2.2.1 Approximations 7
2.2.2 Sparse modeling and numerical calculations 8
2.3 Machine learning 10
2.3.1 Designing a "fair" experiment 10
2.3.2 Data sets and feature sets 10
2.3.3 Neural networks 12
2.3.4 Training and Evaluation 13

3 Results and Discussion **16**
3.1 Final losses 16
3.2 Taking a look at predictions 20

4 Conclusions and Outlook **22**

A Final Loss Tables **23**

List of Figures

2.1 Anderson model with a single impurity- and many bath-sites. 2
2.2 Anderson model with a single impurity- and a single bath-site. 4
2.3 Example calculations of approximations of the self-energy for different sets of model parameters. 9
2.4 Histograms of energy levels ε_i and hopping-amplitudes V_i in the data set. 11
2.5 Histograms of thermodynamic β and temperature in the data set. . . 11
2.6 Data scales of the self-energy dataset. 14

3.1 Final losses after evaluation of 5-fold cross-validation. 17
3.2 Final losses after evaluation of 5-fold cross-validation, considering only Re[$\mathbf{\Sigma_f}$]. 18
3.3 Final losses after evaluation of 5-fold cross-validation, considering Im[$\mathbf{\Sigma_f}$]. 19
3.4 Example predictions for different sets of model parameters, feature sets and neural networks. 21

List of Tables

2.1 Parameters of the Anderson model and their distributions in the data set. 10
2.2 Number of features in each feature set used for training. 12
2.3 Number of parameters of a neural network, including the output layer, but excluding the first hidden layer. 13
2.4 Number of parameters in the first hidden layer. 13

A.1 Mean MSE for each model and feature set after 5-fold cross validation. 23
A.2 Mean MAPE for each model and feature set after 5-fold cross validation. 23
A.3 MSE-based ranks of feature sets. 24
A.4 MAPE-based ranks of feature sets. 24

1. Introduction

The Anderson impurity model has been playing an important role in understanding the physics of quantum many-body impurity problems. Initially answering the question of why some types atoms embedded in a host-crystal exhibit magnetic moments while others do not [Anderson, 1961] [Varma and Yafet, 1976], it contributed significantly to understanding the Kondo-effect [Schrieffer and Wolff, 1966] [Hewson, 1997]. More recent applications include analysing the properties of C60 molecules on metal surfaces [Sau et al., 2008], tuning properties of carbon nanotubes [Latil et al., 2004] [Eichler et al., 2009] and understanding the properties of electrons confined in quantum dots [Meir et al., 1993] [Wingreen and Meir, 1994].

It also serves as an auxiliary-model for dynamical mean field theory ($DMFT$), which is an important method for studying the properties of materials that are significantly influenced by the dynamics of strongly correlated systems of electrons [Georges and Kotliar, 1992] [Georges et al., 1996].

Many of these applications require the solution of a single-impurity Anderson model (e.g. in the form of a Green's function or self-energy). Solving this fermionic problem becomes increasingly difficult as the total number of particles increases [Troyer and Wiese, 2005]. As a result, approximate methods are often used to arrive at a solution.

There has been a growing interest in using machine learning techniques to solve quantum many-body problems, which potentially offer a large reduction in the computational resources required to infer solutions compared to other methods. Possibilities include finding the ground state wave function [Carleo and Troyer, 2017], constructing effective models [Rigo and Mitchell, 2020], and predicting spectral functions [Sturm et al., 2021] and Green's functions [Arsenault et al., 2014].

In this thesis, we investigate the use of dense feed-forward neural networks to predict the self-energy of a single-impurity Anderson model on the imaginary frequency axis. The model parameters as they appear in the Hamiltonian and different types of approximations (f.e. approximations based on the diagrammatic expansion of the self-energy) serve as a basis for training and inference. We try to determine which approximation works best in aiding the neural networks to predict the self-energy.

A possible application of this would be to quickly explore a wide range of potential model parameters for a given task, since once the networks are trained, inference happens much faster than calculating the solution with other tools, but the solution also is less reliable.

2. System and Methods

This section provides details on the Anderson impurity model and its self-energy, as well as the machine learning methods used to predict the self-energy.

2.1 Anderson impurity model

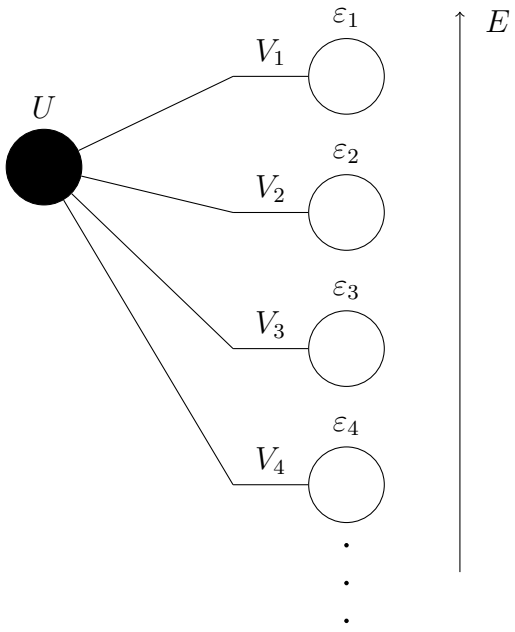

Figure 2.1: Anderson model with a single impurity- and many bath-sites. The left circle represents the impurity-site, with an energy level of $\varepsilon_0 = 0$ and a Coulomb interaction strength of $U \neq 0$. The right circles represent the bath-sites without Coulomb interaction. The energy levels ε_i of these sites can be positive or negative. This is indicated by the fact that they are drawn at different heights with respect to the impurity-site. The possibility of electron hopping is indicated by the lines connecting the bath-sites to the impurity-site, with V_i as the hopping-amplitude [Hewson, 1997].

Throughout this thesis, an Anderson impurity model with one impurity-site will be considered [Anderson, 1961] [Bruus and Flensberg, 2004, p. 142]. Its Hamiltonian is given by

$$\mathcal{H} = U d_\uparrow^\dagger d_\uparrow d_\downarrow^\dagger d_\downarrow + \sum_{k=1}^{N} \sum_{\sigma}^{\{\uparrow,\downarrow\}} \left(\varepsilon_k n_{k\sigma} + V_k (c_{k\sigma}^\dagger d_\sigma + d_\sigma^\dagger c_{k\sigma}) \right), \tag{2.1}$$

where N is the number of bath-sites, d marks the impurity-site, $k \in \{1, ..., N\}$ enumerate the bath-sites and $\sigma \in \{\uparrow, \downarrow\}$ is the spin-index (creation- and annihilation-operators fulfil the canonical anti-commutation relations [Bruus and Flensberg, 2004, p. 13]).

U, $\{V_k\}_k$ and $\{\varepsilon_k\}_k$ are real scalars. U is the strength of the Coulomb interaction, $\{V_k\}_k$ are the hopping-amplitudes and $\{\epsilon_k\}_k$ are the energy levels of the bath-sites. Note that throughout this work, the chemical potential μ and the energy level of the impurity-site ε_{imp} have been set to zero, and the Coulomb interaction strength U has been set to one.

Two variants of the model are used within the scope of this thesis. The first variant is one with a single impurity-site and three bath-sites. The second one will be used as an approximation of the first one with only a single bath-site.

To further analyse the Hamiltonian $\mathcal{H}$ and its properties, we choose the standard basis of the Fock space $\mathcal{F}$, namely the occupation number basis states. We adopt the convention that a basis state consisting of $n_{tot} = n$ particles is denoted as

$$|n_{0\uparrow}, n_{0\downarrow}, n_{1\uparrow}, n_{1\downarrow}, ..., n_{N\uparrow}, n_{N\downarrow}\rangle = c^\dagger_{i\uparrow} c^\dagger_{i\downarrow} c^\dagger_{j\uparrow} c^\dagger_{j\downarrow} c^\dagger_{k\uparrow} c^\dagger_{k\downarrow} ... \, |vac\rangle \,, \quad i < j < k, \tag{2.2}$$

where $\uparrow$ before $\downarrow$ and $i < j < k$ define the order of creation operators (the impurity creation-operator always comes first if the impurity site is populated, as indicated by n_0). For an introduction to many body physics we refer the reader to [Zagoskin, 2014, Bruus and Flensberg, 2004].

2.1.1 Symmetries and conservations

- The total number of particles is conserved by $\mathcal{H}$, formally speaking

$$n_{tot} = \sum_{k,\sigma} n_{k\sigma} : \quad [\mathcal{H}, n_{tot}] = 0. \tag{2.3}$$

- $\mathcal{H}$ does not have any spin-flip terms, which means that

$$S^z_{tot} = \sum_{k\sigma} S^z_{k\sigma} : \quad [\mathcal{H}, S^z_{tot}] = 0. \tag{2.4}$$

- Since U, $\{V_k\}_k$ and $\{\varepsilon_k\}_k$ do not depend on spin, the Eigenenergies of the Eigenstates do not change if the spin of every particle is flipped:

$$\sigma \neq \sigma' : \; \{c_{k\sigma}, c^\dagger_{k\sigma}\} \to \{c_{k\sigma'}, c^\dagger_{k\sigma'}\} \; \Rightarrow \; E_i \to E'_i = E_i. \tag{2.5}$$

2.1.2 Analytical solution

The eigendecomposition of the Hamiltonian is required to be able to calculate the Green's function via the Lehmann-representation (see equation (2.25)). It is therefore compelling to seek for a compact analytically expression for the eigendecomposition of the Hamiltonian.

We consider the Anderson model with a single impurity- and a single bath-site. In the occupation number basis (see equation (2.2)), the Hamiltonian can be expressed as a 16x16-matrix. Considering the symmetries of the Hamiltonian, it

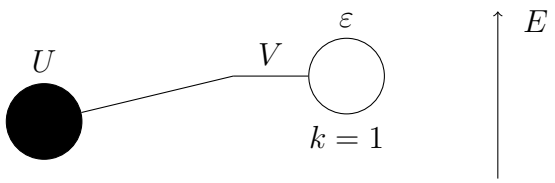

Figure 2.2: Anderson model with a single impurity- and a single bath-site. The left circle embodies the impurity-site, with an energy level of $\varepsilon_0 = 0$ and a Coulomb interaction strength of $U \neq 0$. The right circles represents the bath-site without Coulomb interaction. The energy level ε of this bath-site can be positive or negative. This is indicated by the fact that it is drawn at different height with respect to the impurity-site. The possibility of electron hopping is indicated by the line connecting the bath-site to the impurity-site, with V as the hopping-amplitude.

follows that the Hamiltonian matrix is block-diagonal in the tuple of quantum numbers (n_{tot}, S^z_{tot}). Also, the eigenstates of (n_{tot}, S^z_{tot}) and $(n_{tot}, -S^z_{tot})$ are degenerate. Therefore it is sufficient to only diagonalise the blocks $(n_{tot}, |S^z_{tot}|)$.

Note that the elements of the Hamiltonian matrix are defined as

$$H_{nm} = \langle n| \mathcal{H} |m\rangle , \tag{2.6}$$

where $\{|n\rangle\}$ is the occupation number basis as defined in equation (2.2). The evaluation of scalar products of this form is achieved by inserting $|n\rangle$, $|m\rangle$ and $\mathcal{H}$ in notation of 2^{nd} quantization. The resulting expression can be evaluated by using the canonical commutation relations.

$\underline{(n_{tot} = 1, S^z_{tot} = \pm 1/2)}$:

As discussed earlier, from the symmetries of $\mathcal{H}$ follows that if we compute the eigenstates consisting of $\{|\uparrow\rangle_0 , |\uparrow\rangle_1\}$ and their corresponding eigenvalues, we immediately know the eigenstates and eigenvalues consisting of $\{|\downarrow\rangle_0 , |\downarrow\rangle_1\}$. The eigenvalues are the same, and the states are obtained by the transformation

$$|\uparrow\rangle_0 \to |\downarrow\rangle_0 , \quad |\uparrow\rangle_1 \to |\downarrow\rangle_1 . \tag{2.7}$$

Thus, we need to evaluate only one Hamiltonian matrix block, namely $\tilde{H}_{1,\pm 1/2}$. By choosing the order $(|\uparrow\rangle_0 , |\uparrow\rangle_1)$ we get

$$\tilde{H}_{1,\pm 1/2} = \begin{pmatrix} 0 & V \\ V & \varepsilon \end{pmatrix} . \tag{2.8}$$

Solving the eigenvalue equation we find the eigenenergies and corresponding eigenstates to be

$$E_{1,2} = \frac{1}{2}\varepsilon \mp \sqrt{\frac{1}{4}\varepsilon^2 + V^2}, \quad |\psi_{1,2}\rangle = \frac{1}{\sqrt{V^2 + E^2_{1,2}}}(V |\uparrow\rangle_0 + E_{1,2} |\uparrow\rangle_1). \tag{2.9}$$

.

$\underline{(n_{tot} = 2, S^z_{tot} = \pm 1)}$:

There are only two possible states with these quantum numbers: $|\uparrow\rangle_0 |\uparrow\rangle_1$ and $|\downarrow\rangle_0 |\downarrow\rangle_1$. Thus we find that the Hamiltonian block $\tilde{H}_{2,\pm 1}$ effectively is a scalar:

$$\tilde{H}_{2,\pm 1} = \varepsilon. \tag{2.10}$$

It immediately follows that

$$E_3 = \varepsilon, \quad |\psi_{3,1}\rangle = |\uparrow\rangle_0 |\uparrow\rangle_1 .^{1}$$

$\underline{(n_{tot} = 2, S^z_{tot} = 0)}$:

If we choose the order of states to be $(|\uparrow\rangle_0 |\downarrow\rangle_1 , |\downarrow\rangle_0 |\uparrow\rangle_1 , |\uparrow\downarrow\rangle_0 , |\uparrow\downarrow\rangle_1)$, the Hamiltonian block $\tilde{H}_{2,0}$ reads

$$\tilde{H}_{2,0} = \begin{pmatrix} \varepsilon & 0 & V & V \\ 0 & \varepsilon & -V & -V \\ V & -V & U & 0 \\ V & -V & 0 & 2\varepsilon \end{pmatrix}. \tag{2.12}$$

Attempts to diagonalise this matrix lead to a *casus irreducibilis* for some of the eigenvalues. A compact solution has only been found for the special case of $U = 2\varepsilon$.

For the general case, only

$$E_4 = \varepsilon, \quad |\psi_4\rangle = \frac{1}{\sqrt{2}}(|\uparrow\rangle_0 |\downarrow\rangle_1 + |\downarrow\rangle_0 |\uparrow\rangle_1) \tag{2.13}$$

has been found. Proceeding with the aforementioned special case, we find

$$E_5 = 2\varepsilon, \quad |\psi_5\rangle = \frac{1}{\sqrt{2}}(|\uparrow\downarrow\rangle_0 - |\uparrow\downarrow\rangle_1) \tag{2.14}$$

and

$$E_{6,7} = \frac{3}{2}\varepsilon \mp \sqrt{\frac{1}{4}\varepsilon^2 + 4V^2}$$
$$|\psi_{6,7}\rangle = \frac{1}{\sqrt{8V^2 + 2(\varepsilon - E_{6,7})^2}}(2t(|\downarrow\rangle_0 |\uparrow\rangle_1 - |\uparrow\rangle_0 |\downarrow\rangle_1) + (\varepsilon - E_{6,7})(|\uparrow\downarrow\rangle_0 + |\uparrow\downarrow\rangle_1)). \tag{2.15}$$

$\underline{(n_{tot} = 3, S^z_{tot} = \pm 1/2)}$:

We take the order of states to be $(|\uparrow\downarrow\rangle_0 |\uparrow\rangle_1 , |\uparrow\rangle_0 |\uparrow\downarrow\rangle_1)$. This leads to

$$\tilde{H}_{3,\pm 1/2} = \begin{pmatrix} U + \varepsilon & -V \\ -V & 2\varepsilon \end{pmatrix}, \tag{2.16}$$

[1] We can find an interesting interpretation for this: Recall that for a time-independent problem, the time-evolution operator $U(t)$ is given by

$$U(t) = e^{-i\mathcal{H}t} = \sum_{k=0}^{\infty} \frac{(-i\mathcal{H}t)^k}{k!}. \tag{2.11}$$

When acting with $\mathcal{H}$ on either of the two states (to compute their time evolution), there are not enough degrees of freedom in the system that would allow for these states to change: The electrons cannot hop because of the Pauli-principle, and the spins can not flip because $\mathcal{H}$ does not have any spin-flip terms, it conserves S^z_{tot}. Therefore it follows that $|\uparrow\rangle_0 |\uparrow\rangle_1$ and $|\downarrow\rangle_0 |\downarrow\rangle_1$ must eigenstates of the Hamiltonian. The corresponding eigenenergy must be ε, since the energy level of the bath-site is ε and the energy level of the impurity-site is equal to zero.

with the following eigenvalues and eigenstates:

$$E_{8,9} = \frac{3\varepsilon + U}{2} \mp \frac{1}{2}\sqrt{(\varepsilon - U)^2 + 4t^2}$$
$$|\psi_{8,9}\rangle = \frac{1}{\sqrt{V^2 + (E_{9,8} - 2\varepsilon)^2}}(V\,|\uparrow\downarrow\rangle_0\,|\uparrow\rangle_1 + (E_{9,8} - 2\varepsilon)\,|\uparrow\rangle_0\,|\uparrow\downarrow\rangle_1). \tag{2.17}$$

$\underline{(n_{tot} = 4, S^z_{tot} = 0)}$:

There is only one state in this case, namely $|\uparrow\downarrow\rangle_0\,|\uparrow\downarrow\rangle_1$. Therefore we arrive at

$$E_{10} = U + 2\varepsilon, \quad |\psi_{10}\rangle = |\uparrow\downarrow\rangle_0\,|\uparrow\downarrow\rangle_1\,. \tag{2.18}$$

2.2 Self-energy of the impurity-site

Self-energy

The central object of interest within this thesis is the Fourier transform of the impurity-site's imaginary time self-energy, denoted as $\Sigma_\sigma(i\omega_n)$. It is given by the Dyson equation as [Bruus and Flensberg, 2004, p. 236]

$$\Sigma_\sigma(i\omega_n) = G_{0,\sigma}(i\omega_n)^{-1} - G_\sigma(i\omega_n)^{-1}. \tag{2.19}$$

Because of the symmetries of the Hamiltonian we can omit the spin-variable σ in our notation and write

$$\Sigma(i\omega_n) = G_0(i\omega_n)^{-1} - G(i\omega_n)^{-1}. \tag{2.20}$$

To distinguish this self-energy from its approximations (which are described in section (2.2.1)), this self-energy is also referred to as the self-energy of the *full* model, Σ_f.

Since we are dealing with electrons, which are fermions, these many-body objects are non-zero only for fermionic Matsubara frequencies [Bruus and Flensberg, 2004, p. 189]

$$\omega_n = \frac{(2n+1)\pi}{\beta}, \quad n \in \mathbb{N}_0, \quad \beta = \frac{1}{T}. \tag{2.21}$$

Green's function

$G(i\omega_n)$ is the Fourier transform of the imaginary-time Green's function defined as [Bruus and Flensberg, 2004, p. 187]

$$G(\tau, \tau') = -\left\langle T_\tau \left(c(\tau)^\dagger(\tau')\right)\right\rangle, \quad -\beta < \tau < \beta. \tag{2.22}$$

If the physical system under consideration is time-translationally invariant, which is the case in this thesis, this simplifies to [Bruus and Flensberg, 2004, p. 89]

$$G(\tau) = -\left\langle T_\tau \left(c(\tau)c^\dagger(0)\right)\right\rangle. \tag{2.23}$$

The corresponding Fourier transform [Bruus and Flensberg, 2004, p. 189]

$$G(i\omega_n) = \int_0^\beta d\tau e^{i\omega_n\tau} G(\tau), \tag{2.24}$$

can also be expressed in the Källén-Lehmann spectral representation as [Bruus and Flensberg, 2004, p. 190]

$$G(i\omega_n) = \frac{1}{Z}\sum_{n,n'} \frac{\langle n|\, c\, |n'\rangle \langle n'|\, c^\dagger\, |n\rangle}{i\omega_n + E_n - E_{n'}}(e^{-\beta E_n} + e^{-\beta E_{n'}}). \tag{2.25}$$

Non-interacting Green's function

To compute the non-interacting Green's function G_0, we split the Hamiltonian into a non-interacting part $\mathcal{H}_0$, and an interacting part $\mathcal{H}_V$:

$$\mathcal{H} = \mathcal{H}_0 + \mathcal{H}_V. \tag{2.26}$$

The non-interacting part is characterised by being quadratic [Bruus and Flensberg, 2004, p. 129], i.e. bilinear in annihilation or creation operators [Bruus and Flensberg, 2004, p. 141]).

G_0 is then defined as the Green's function of the non-interacting part $\mathcal{H}_0$ [Bruus and Flensberg, 2004, p. 141].

2.2.1 Approximations

The neural networks are trained with approximations of the full model's self-energy, which (within the scope of this thesis) has three bath-sites and one impurity-site. Four different approximations are used. They are described in the following (the abbreviations in parentheses will be used later to refer to the different approximations).

"Single pole" self-energy, Σ_s (Ss)

The *single pole* self-energy Σ_s is the self-energy of a model with only a single bath-site, as described in chapter (2.1.2). The corresponding parameters $(\varepsilon_1, V_1)_s$ are determined by taking the parameters of the i^{th} bath-site the full model, $(\varepsilon_i, V_i)_f$, where the energy level is closest to that of the impurity-site. Since the energy level of the impurity-site is set to zero, this condition reduces to

$$i = \min_{i\in\{1,2,3\}} |\varepsilon_i|\,. \tag{2.27}$$

"Resonant" self-energy, Σ_r (Sr)

The *resonant* self-energy Σ_r is also based on a model with only one bath-site as described above, however the parameters $(\varepsilon_1, V_1)_r$ are chosen differently. ε_1 is simply set to zero, and V_1 is calculated as the mean of the hybridisation function $\Delta(\tau)$ of the full model,

$$V_1 = \frac{1}{\beta}\int_0^\beta \Delta(\tau)d\tau, \tag{2.28}$$

where $\Delta(\tau)$ is defined as [Radmanovac, 2024]

$$\Delta(\tau) = \sum_i V_i^2 \frac{e^{-\tau\varepsilon_i}}{1+e^{-\beta\varepsilon_i}}. \tag{2.29}$$

To distinguish this V_1 from that of the single pole model, we refer to V_1 of this model as V_r (Vr).

The next two approximations are based on the diagrammatic expansion of the self-energy [Bruus and Flensberg, 2004, p. 233].

1st order approximation (HF)

This approximation comes from evaluating the first two diagrams of the expansion of the self-energy, the so-called *Hartree-* and *Fock self-energy diagrams* [Bruus and Flensberg, 2004, p. 236, 237]. The first one is a single real-valued constant [Bruus and Flensberg, 2004, p. 237], and the second diagram evaluates to zero, since the Hamiltonian under consideration does not contain any spin-flip terms.

For more details on how it was calculated, see [Radmanovac, 2024].

2nd order approximation (MP2)

From the various second-order diagrams (characterised by the presence of two interaction lines), it can be shown that only the pair-bubble diagram (see [Bruus and Flensberg, 2004, p. 238]) survives for the model considered in this work. For further details, see [Radmanovac, 2024].

2.2.2 Sparse modeling and numerical calculations

Sparse modelling is an efficient way to represent the information contained in imaginary-time many-body objects [Shinaoka et al., 2017] [Li et al., 2020]. See also [Shinaoka et al., 2022] for an introduction. In contrast to choosing a finite cut-off frequency to construct the various data sets for machine learning, this method introduces a natural and systematic way to constrain the set of Matsubara frequencies $\mathcal{W}$, at which the self-energy must be evaluated on. Furthermore, meaningful dimensionality reduction is one of the most important preprocessing techniques in machine learning to reduce computational demands and the chance of overfitting [Karpathy, 2019, smaller input dimensionality].

For the purposes of this thesis this means that the self-energy only needs to be evaluated at a certain set of frequencies

$$\begin{aligned}\mathcal{W} = \{&i\omega_1, i\omega_3, i\omega_5, i\omega_7, i\omega_9, i\omega_{11}, i\omega_{13}, i\omega_{15}, i\omega_{17}, i\omega_{19}, i\omega_{21}, i\omega_{23}, i\omega_{25}, i\omega_{29}, i\omega_{33} \\ &i\omega_{37}, i\omega_{43}, i\omega_{49}, i\omega_{57}, i\omega_{69}, i\omega_{83}, i\omega_{101}, i\omega_{127}, i\omega_{165}, i\omega_{237}, i\omega_{399}, i\omega_{1207}\},\end{aligned} \tag{2.30}$$

which were determined using the library *SparseIR*[2] [Wallerberger et al., 2023].

The numerical calculations of the self-energies were carried out using the programming language *Julia*[3] [Bezanson et al., 2017] and the group internal package *Fermions.jl*.

The following figure shows numerical calculations of the discussed approximations for three different sets of model parameters.

[2] See `https://spm-lab.github.io/sparse-ir-tutorial/` for a comprehensive tutorial.

[3] see also `https://julialang.org/`

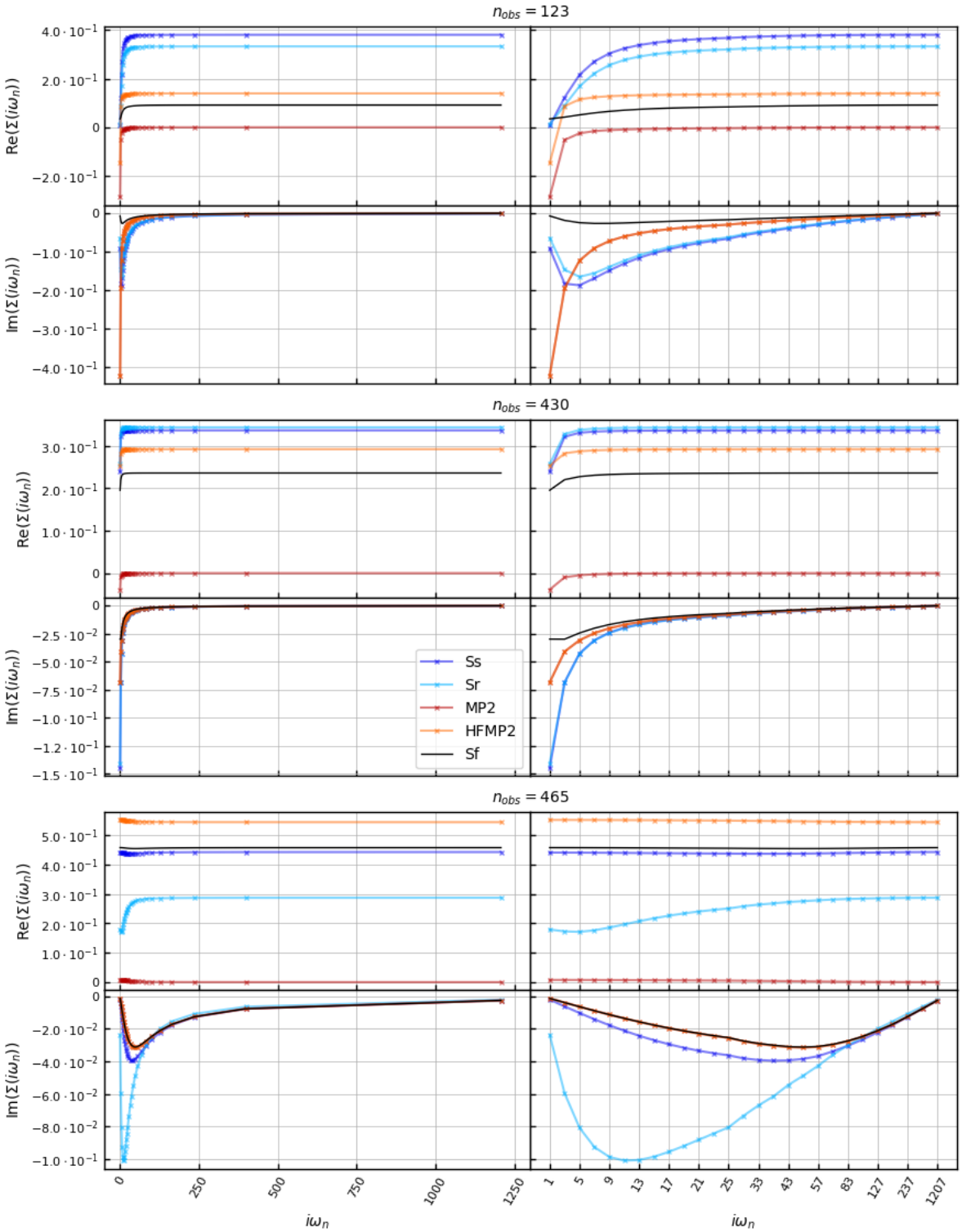

Figure 2.3: Example calculations of approximations of the self-energy for different sets of model parameters. Each of the three 4x4 plots shows the four approximations (*Ss, Sr, MP2, HFMP2*) of the self-energy for one set of model parameters, as indicated by n_{obs}. *HF* is not plotted separately, because it can be inferred from $HF = HFMP2 - MP2$. For the imaginary part, *MP2* and *HFMP2* match, because *HF* is a real-valued constant. Within a 4x4 plot, the left column shows the self energy plotted over a linear spread of frequencies, which is how one would typically plot the self-energy. To be able to better see the differences in the approximations, in the right column the frequencies $i\omega_n \in \mathcal{W}$ are spread equidistant.

2.3 Machine learning

Neural networks were used to predict the self-energy of an Anderson model with three bath-sites and one impurity-site. The following sections describe the data sets, neural network architectures, training-loop and evaluation methods to rank the feature sets.

The machine learning framework *Tensorlfow 2.14.0*[4] [Abadi et al., 2016] is used to train the neural networks.

2.3.1 Designing a "fair" experiment

The hyperparameters model size and data preprocessing methods (feature- and label-scaling) were varied across models, with the aim of achieving a reasonably fair comparison between the various feature sets. While the usual goal of hyperparameter exploration is to find an optimal model for a given data set, the goal here is to compare the performance of different feature sets across various models.

One has to find a balance between exploring a certain range of hyperparameters and limliting the complexity and time required to train and evaluate all these different models.[5] For simplicity, a fixed set of hyperparameters has been chosen for this comparison (*grid-search*). The best feature set is determined as the one resulting in best average performance across all different models.

2.3.2 Data sets and feature sets

Anderson model parameters

An instance of an Anderson model with three bath-sites as used in this thesis is defined by a set of nine parameters. A total amount of 10.000 parameter sets, which form the basis of the data set, were drawn from probabilistic distributions. These parameters and their range of values, together with a brief description of how they were determined are listed in table (2.1).

Table 2.1: Parameters of the Anderson model and their distributions in the data set.

	description	range	distribution
β	themodynamic beta $= 1/T$	[2, 50]	T drawn from exponential distribution with $\tau = 0.5$, truncated at $T_U - T_L$ and added T_L, where $T_U = 0.5$, $T_L = 0.02$.
U	Coulomb interaction strength	1	———
ε_{imp}	impurity-site energy level	0	———
ε_i	bath-site energy level	[-5, 5]	Double exponential distribution with $\tau = 2.0$, truncated at ± 5.
V_i	hopping-amplitude	[-5, 5]	Double exponential distribution with $\tau = 0.7$, truncated at ± 5.

For each set of parameters, three energy levels ε_i and three hopping-amplitudes V_i were drawn. Further, the energies were ordered according to $\min_{i \in \{1,2,3\}} |\varepsilon_i|$. The energy level and hopping-amplitude of the simplified model are then (ε_1, V_1), as explained in section (2.2.1).

[4] see also `https://www.tensorflow.org/api_docs/python/tf/keras`

[5] Different strategies of hyperparameter optimisation have been proposed to better address this issue, such as performing random searches in the hyperparameter space [Bergstra and Bengio, 2012].

Histograms of all the drawn energy levels, hopping-amplitudes and temperatures are shown in figures (2.4) and (2.5).

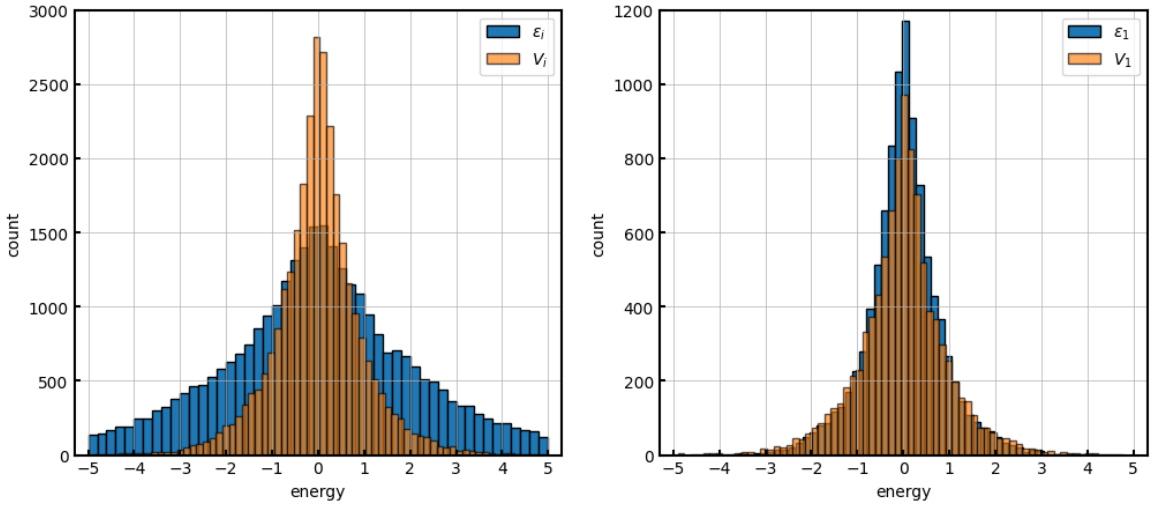

Figure 2.4: Histograms of energy-levels ϵ_i and hopping-amplitudes V_i in the data set. ε_i were drawn from a double exponential distribution with $\tau = 2.0$, truncated at ± 5. V_i were drawn from a double exponential distribution with $\tau = 0.7$, truncated at ± 5.

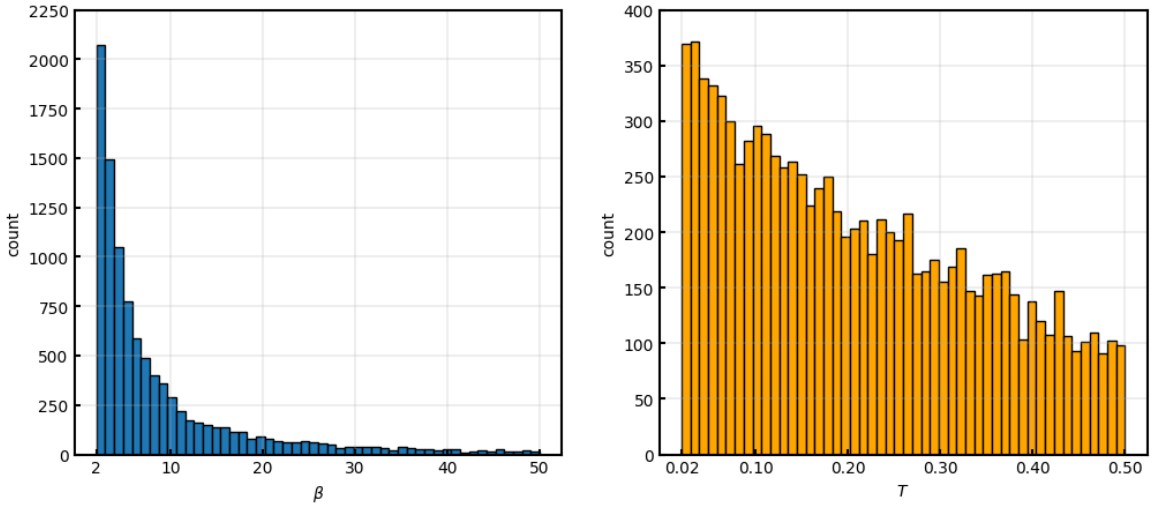

Figure 2.5: Histograms of thermodynamic β and temperature in the data set. T was drawn from an exponential distribution with $\tau = 0.5$, truncated at $T_U - T_L$ and added T_L, where $T_U = 0.5$, $T_L = 0.02$.

Feature sets

In order to easily distinguish the various data sets that were used for training, they were named in a systematic way. Each name is split up into two parts, separated by an underscore: $\langle features \rangle\_\langle labels \rangle$. The abbreviations as defined in section (2.2.1) can now be used to indicate whether the corresponding data is part of the features, or the labels. Since within the scope of this thesis, the labels are the self-energy of the full model, $\langle features \rangle$ will always reduce to *Sf*.

An example name is *par-beta-Vr-Sr_Sf*: The underscore separates features on the left from labels on the right. *par* stands for the model parameters as defined in

table (2.1), but without β (because it is indicated separately), U and ε_{imp} (because they are constant, see section (2.1)). Thus, *par* only contains ε_i and V_i, which are six parameters in total. For the meaning of the other abbreviations, see section (2.2.1). In total, eight different feature sets were used for training, summarised in the table below.

Table 2.2: Number of features in each feature set used for training. *par* contains six features. *beta*, *Vr* and *HF* are a single feature each. *Ss*, *Sr* and *MP2* contain $|\mathcal{W}| \cdot 2 = 54$ features each. Since *HFMP2* is the pointwise addition of *HF* to *MP2*, it also contains 54 features.

name	short	no. features
par-beta_Sf	par-beta	7
par-beta-Ss_Sf	-Ss	61
par-beta-Vr-Sr_Sf	-Vr-Sr	62
par-beta-Ss-Vr-Sr_Sf	-Ss-Vr-Sr	116
par-beta-HF_Sf	-HF	8
par-beta-HF-MP2_Sf	-HF-MP2	62
par-beta-HFMP2_Sf	-HFMP2	61
par-beta-Ss-Vr-Sr-HFMP2_Sf	-Ss-Vr-Sr-HFMP2	170

2.3.3 Neural networks

Fully connected feedforward neural networks are used. The number of hidden layers and the number of neurons in these hidden layers are varied across different neural networks (all hidden layers contain the same number of neurons). The idea is to base the comparison between different feature sets on a greater variety of models. Choosing only one model for comparison could result in it being too small or too large for certain feature sets.

Each network has a single input layer[6], with a width equal to the number of features. One to three dense hidden layers follow, each with weights and biases[7] and rectified linear units ($ReLU$) as activation [Nair and Hinton, 2010] [He et al., 2015]. The width of these hidden layers is either 20, 60, or 180 (all having the same width). The final output layer is also a dense layer with linear activation[8]. Since we want to predict the self-energy for a fixed number of arguments, it has two output neurons (real- and imaginary part) for each Matsubara frequency $i\omega_n \in \mathcal{W}$. Therefore, the number of neurons in the output layer is 54 for every network. Note that the input layer as well as $ReLU$- and linear activation functions do not contain any trainable parameters.

Models having the identifier *FSLS* in their name were trained with feature- and label-scaling, meaning that in the training set, each feature and label was scaled to standard-score. The same scaling parameters were then applied to the validation-/test-sets. Data normalisation speeds up training by smoothing the optimisation landscape [Santurkar et al., 2018]. To make the losses of these networks comparable for the final loss analysis, the predictions were rescaled.

[6] see `https://www.tensorflow.org/api_docs/python/tf/keras/layers/InputLayer`
[7] see `https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dense`
[8] see `https://www.tensorflow.org/api_docs/python/tf/keras/activations/linear`

Number of parameters

To calculate the total number of parameters in the i^{th} layer, one needs to consider a bias parameter for each of the n_i neurons in the layer, and $n_i \cdot n_{i-1}$ weights for the connections of the n_{i-1} neurons of the previous layer to this layer.

The number of parameters in the first hidden layers are listed in table (2.4), since they depend on the number of features. Table (2.3) shows the number of all other layers combined, which can be calculated independently of the number of features. These two tables can be used to calculate the number of trainable parameters of a given network.

Table 2.3: Number of parameters of a neural network, including the output layer, but excluding the first hidden layer. The last layer is the output layer with $|\mathcal{W}| \cdot 2 = 54$ neurons. Each layer has $n_i \cdot n_{i-1}$ weights and n_i biases (n_i is the number of neurons in the i^{th} layer).

neurons \ hidden layers	1	2	3
20	1134	1534	1934
60	3294	6894	10494
180	9774	42174	74574

Table 2.4: Number of parameters in the first hidden layer. The number of parameters in the first hidden layer depends on the number of features n_f. It contains $n_f \cdot n$ weights and n biases, where n is the number of neurons in the first hidden layer.

neurons \ feature set	par-beta	-Ss	-Vr-Sr	-Ss-Vr-Sr	-HF	-HF-MP2	-HFMP2	-Ss-Vr-Sr-HFMP2
20	160	1240	1260	2340	180	1260	1240	3420
60	480	3720	3780	7020	540	3780	3720	10260
180	1440	11160	11340	21060	1620	11340	11160	30780

2.3.4 Training and Evaluation

Trainings-loop

The networks were trained using the statistical optimiser *Adam* [Abadi et al., 2016], because it is considered to be robust against the choice of hyperparameters [Goodfellow et al., 2016]. They were trained for a total of 1.000 epochs, using a batch-size of 32 and *mean squared error (MSE)* as the loss-function. After training for 1.000 epochs, the epoch with the lowest validation loss was restored to compare the final losses.

5-fold cross-validation

To estimate the performance of a model trained with a given feature set, each version of a model was trained five times using 5-fold cross-validation [Japkowicz

and Shah, 2011, p. 172]. From the total of 10.000 data-points, 1.000 were set aside to determine the final loss (outer split / test split). The remaining 9.000 were used for the different training-/validation splits for 5-fold cross-validation (inner splits), where the validation split is used to checkpoint the best performing model.

This ensures that the test set has not been used in any way before the final loss is determined, which reduces the bias of the performance estimate [Japkowicz and Shah, 2011, p. 177].

Loss functions

Both *mean squared error*

$$MSE(y, \hat{y}) = \frac{1}{N} \sum_{n_{obs}}^{N} \frac{1}{|\mathcal{W}|} \sum_{i\omega_n \in \mathcal{W}} [\hat{y}_{n_{obs}}(i\omega_n) - y_{n_{obs}}(i\omega_n)]^2 \tag{2.31}$$

and *mean absolute percentage error*

$$MAPE(y, \hat{y}) = \frac{1}{N} \sum_{n_{obs}}^{N} \frac{1}{|\mathcal{W}|} \sum_{i\omega_n \in \mathcal{W}} 100 \cdot \left| \frac{\hat{y}_{n_{obs}}(i\omega_n) - y_{n_{obs}}(i\omega_n)}{y_{n_{obs}}(i\omega_n)} \right| \tag{2.32}$$

are used for comparison, where $\hat{y}$ and y are the predicted and true self-energy, and $|\mathcal{W}|$ is the cardinality of the set of sparse frequencies as defined in equation (2.30).

The reason for choosing *MAPE* is explained in the following (for further discussions of accuracy measures see [Hyndman and Koehler, 2006] and [Botchkarev, 2019]). Figure (2.6) shows means and standard-deviations of the self-energy over the entire data set.

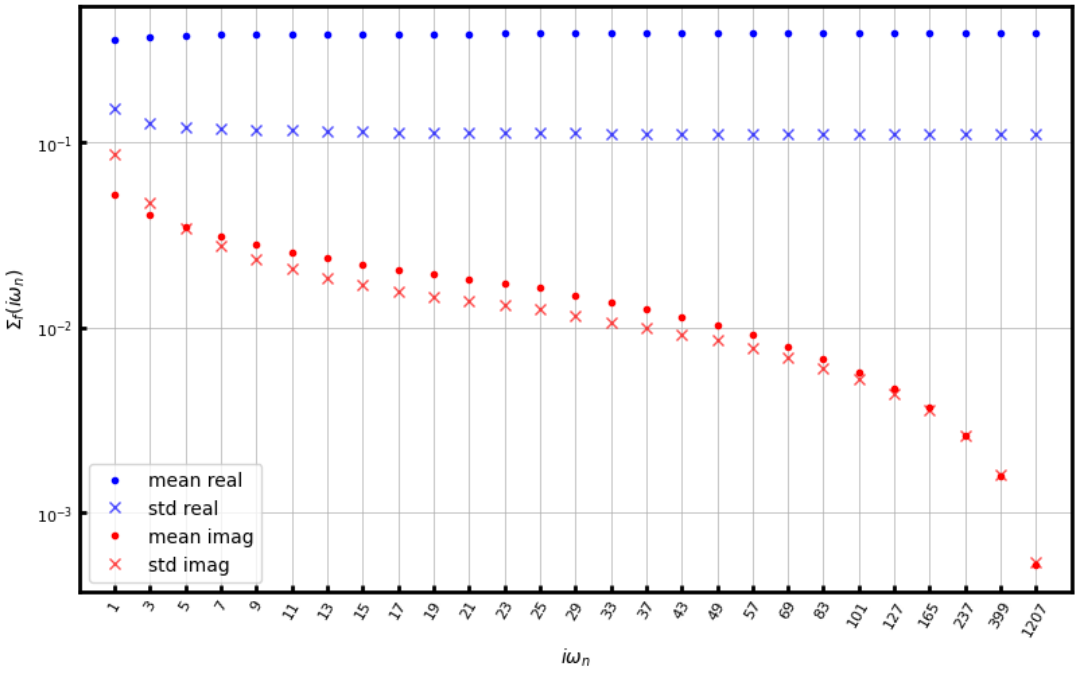

Figure 2.6: Data scales of the self-energy dataset. Mean value and standard-deviation of the absolute value of the self-energy across all 10.000 impurity models. The real part is two orders of magnitude larger than the imaginary part, which should be taken into account when choosing a loss function for evaluation.

The imaginary part is one to two orders of magnitude smaller than the real part. Hence, when using *MSE*, the error from predicting the real part is likely to dominate the total *MSE*, as it is a scale dependent error. *MAPE*, on the other hand does not suffer from this problem, since the difference between true and predicted value is divided by the true value. *MAPE* only assings importance to relative errors, irrespectively of the magnitude of the absolute errors.

From a physical point of view, it may nevertheless be desirable to place more emphasis on accurately predicting values of higher magnitude than others. For example, in a given perturbation series, higher order terms may contribute less (i.e. they are of smaller magnitude), which means that it may be more important to accurately predict lower order terms. The choice of an appropriate accuracy measure therefore also depends on the physical problem at hand.

One has to be careful when using *MAPE* when there is a possibility that the true value is zero or very close to zero. *Tensorflow* deals with this problem by adding an epsilon of 10^{-7} to the denominator.[9] Since the minimum absolute value of the self-energies in the dataset is about $2.3 \cdot 10^{-6}$, this small epsilon will skew the *MAPE* by a maximum factor of about $5 \cdot 10^{-2}$, which does not affect on the main result of the comparison between feature sets.

[9] see `https://www.tensorflow.org/api_docs/python/tf/keras/losses/MeanAbsolutePercentageError` and `https://github.com/keras-team/keras/blob/v2.15.0/keras/backend_config.py#L34`

3. Results and Discussion

3.1 Final losses

Figures (3.1) - (3.3) show the final losses after evaluating 5-fold cross-validation, tables (A.1) and (A.2) list the mean losses for comparison. As some feature sets perform very similarly (see figure (3.1) first row, second plot), tables (A.3) and (A.4) list their ranks across all different models.

Main results

Feature sets containing approximations based on the diagrammatic expansion of Σ_f clearly outperform feature sets only containing approximations based on the simplified model. Networks trained with approximations based on the diagrammatic expansion show the following characteristics compared to others:

- The final losses are smaller.
- The variation across different networks of the same type that have been trained on different splits in 5-fold cross-validation is smaller (as assessed by the error-bars in figure (3.1)).
- The variation between models of different sizes that were trained with the same feature set is much smaller (but these differences decrease as model size increases).

In particular, the feature set *-HF-MP2* results in the best average performance, see table (A.3).

Furthermore, figures (3.2) and (3.3) show that the networks perform better on the real part, than on the imaginary part.

-HF-MP2

Table (A.3) shows that in terms of mean rank based on *MSE*, the feature set *-HF-MP2* outperforms all other feature sets. It also outperforms *-Ss-Vr-Sr-HFMP2*, which contains almost three times as many features (see table (2.2)).

It is particularly interesting that *-HF-MP2* outperforms *-HFMP2*, although the only difference is that the former contains *HF* and *MP2* as separate features, while the latter contains the pointwise addition of *HF* and *MP2*. A possible explanation for this is that *-HF-MP2* contains more information than *-HFMP2*, since in *-HF-MP2*, *HF* and *MP2* are provided as separate features. As a result, this feature set also implicitly contains the difference between *HF* and *MP2*.

Another possible explanation is that training with *-HF-MP2* instead of *-HFMP2* results in the neural network having more parameters to train, since the number of parameters also depends on the number of features (see table (2.4)).

-HF

The feature set *-HF* also deserves special attention, since it outperforms all other feature sets *not* containing approximations based on the diagrammatic expansion of Σ_f, even though it contains only one more feature compared to the set *par-beta*, and more than ten times less features than the set *-Ss-Vr-Sr* (see table (2.2)).

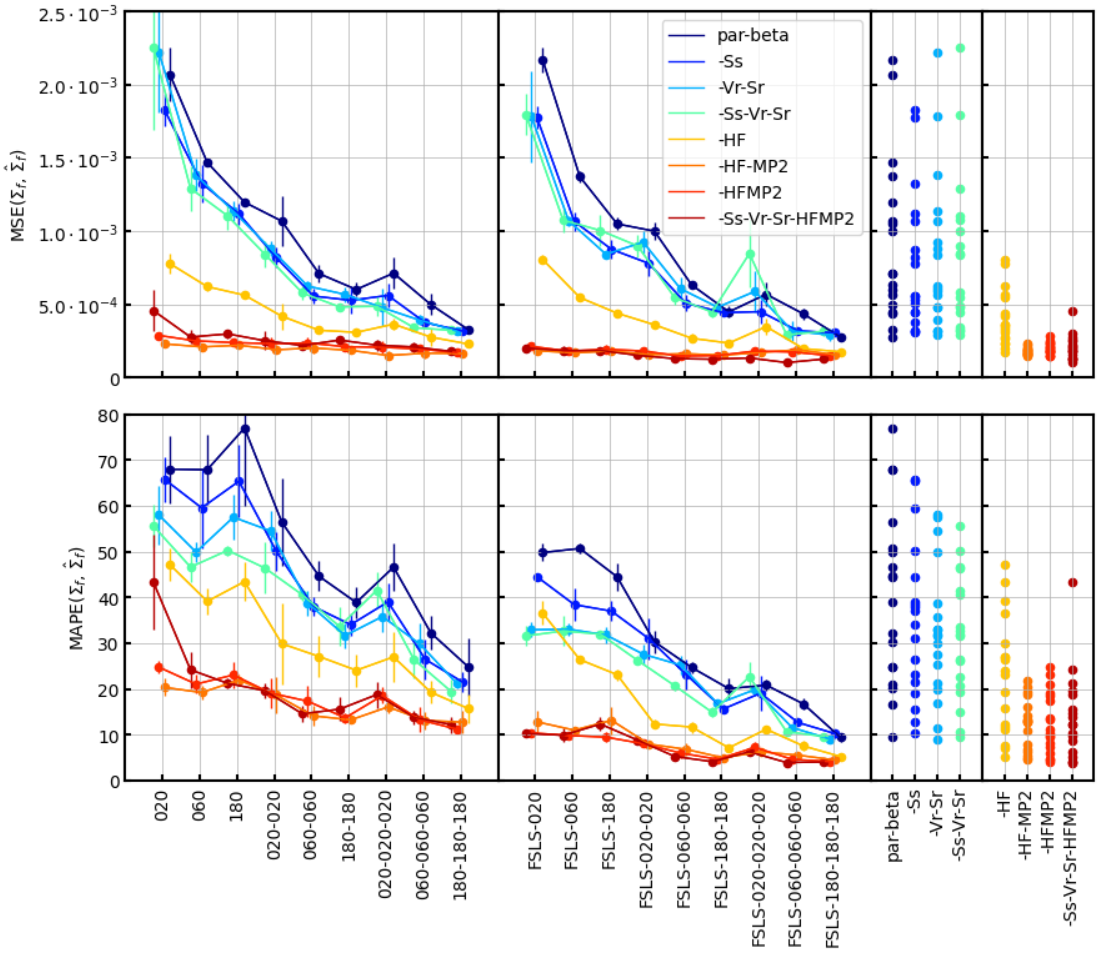

Figure 3.1: Final losses after evaluation of 5-fold cross-validation. The first two plots in each row show the means (points) and standard-deviations (error bars) of the mean squared error (*MSE*, top row of plots) and mean absolute percentage error (*MAPE*, bottom row of plots) for different models across different feature sets. *MSE* was used for training, *MAPE* only for evaluation. The models differ in size and in whether feature- and label-scaling was used for training (as indicated by the prefix *FSLS*). The size is indicated by a series of 3-digit numbers: Each number stands for the number of neurons in that layer, and the total count of numbers separated by a comma stand for the number of hidden layers (f.e. *020-020* means two hidden layers with 20 neurons each). The feature sets differ in the type of approximation they contain. Every feature set contains the energy-parameters of the full model and the *thermodynamic beta* $\beta = 1/T$. The last two plots in each row project the means from the first two plots onto a single line for comparison across feature sets. Feature sets containing diagrammatic approximations of Σ_f clearly outperform those that do not (compare the second last to the last plot in each row). The *MAPE* shows that models that use feature- and label-scaling outperform those that do not.

Machine learning related

Figure (3.1) shows that models that use feature- and label-scaling (*FSLS*) perform better than those the others. It also shows that the *MSE* is unable to pick this difference in performance. Figures (3.2) and (3.3) show that the reason for this improvement is that the models using *FSLS* better predict the imaginary part of Σ_f, while the prediction-accuracy of the real part is similar.

This means the the *MAPE* is a more discriminative measure of performance than the *MSE* for the given machine learning task, despite the fact that the models were trained on *MSE*. As explained in section (2.3.4), the reason for this is that *MAPE* is a scaled error and that the scales of real- and imaginary-parts of Σ_f are different.

Furthermore, the scales of the imaginary part itself vary by two orders of magnitude (depending on $i\omega_n$), as shown in figure (2.6). Consequently, the *MSE* also suffers from a scaling problem when the imaginary part is considered alone: Figure (3.3) shows that the *MSE* only captures relative performance differences of about 20% between models that use *FSLS* and those that do not, while the *MAPE* captures relative performance differences of more than 50%.

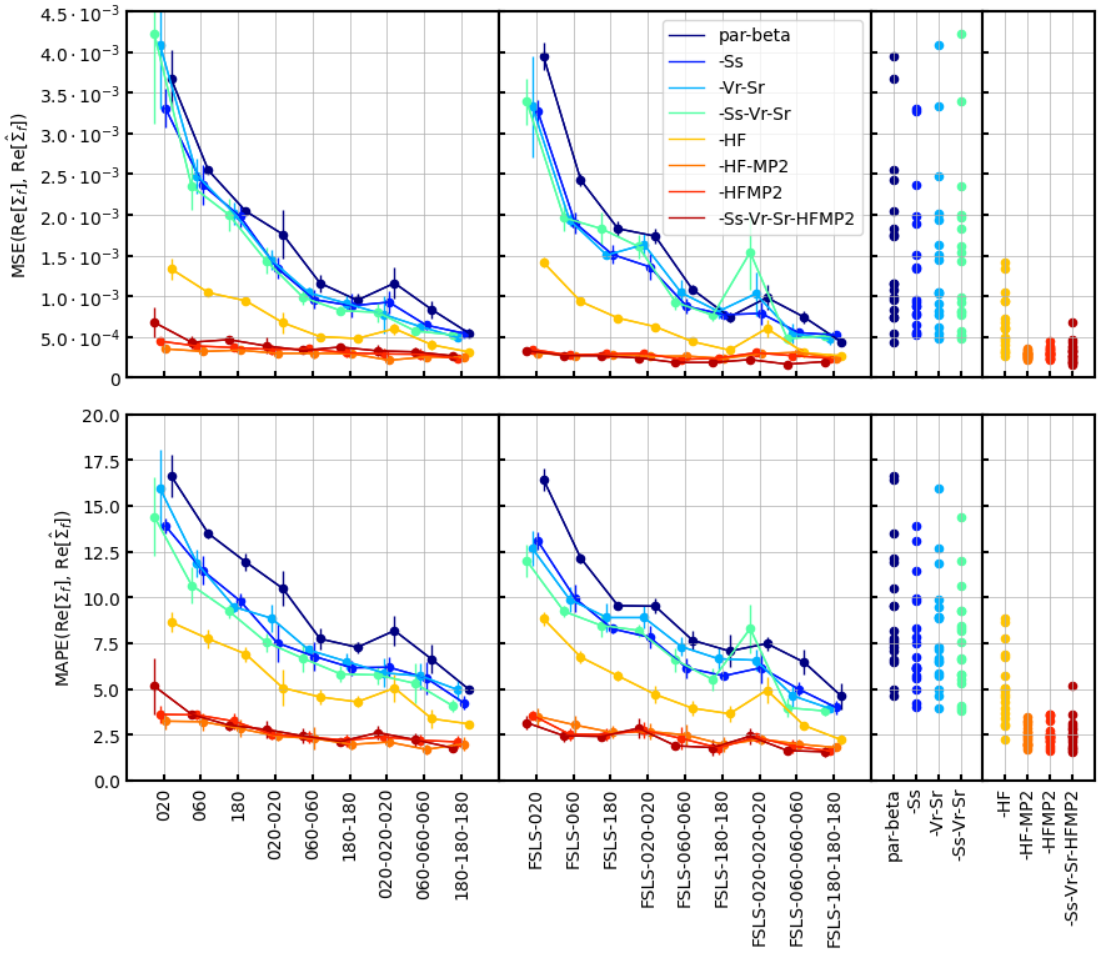

Figure 3.2: Final losses after evaluation of 5-fold cross-validation, considering only $\mathbf{Re[\Sigma_f]}$. Similar to figure (3.1), but losses are calculated based on $\mathrm{Re}[\Sigma_f]$ only. Neural networks that use feature- and label-scaling (as indicated by the prefix *FSLS*) show similar losses to those that do not.

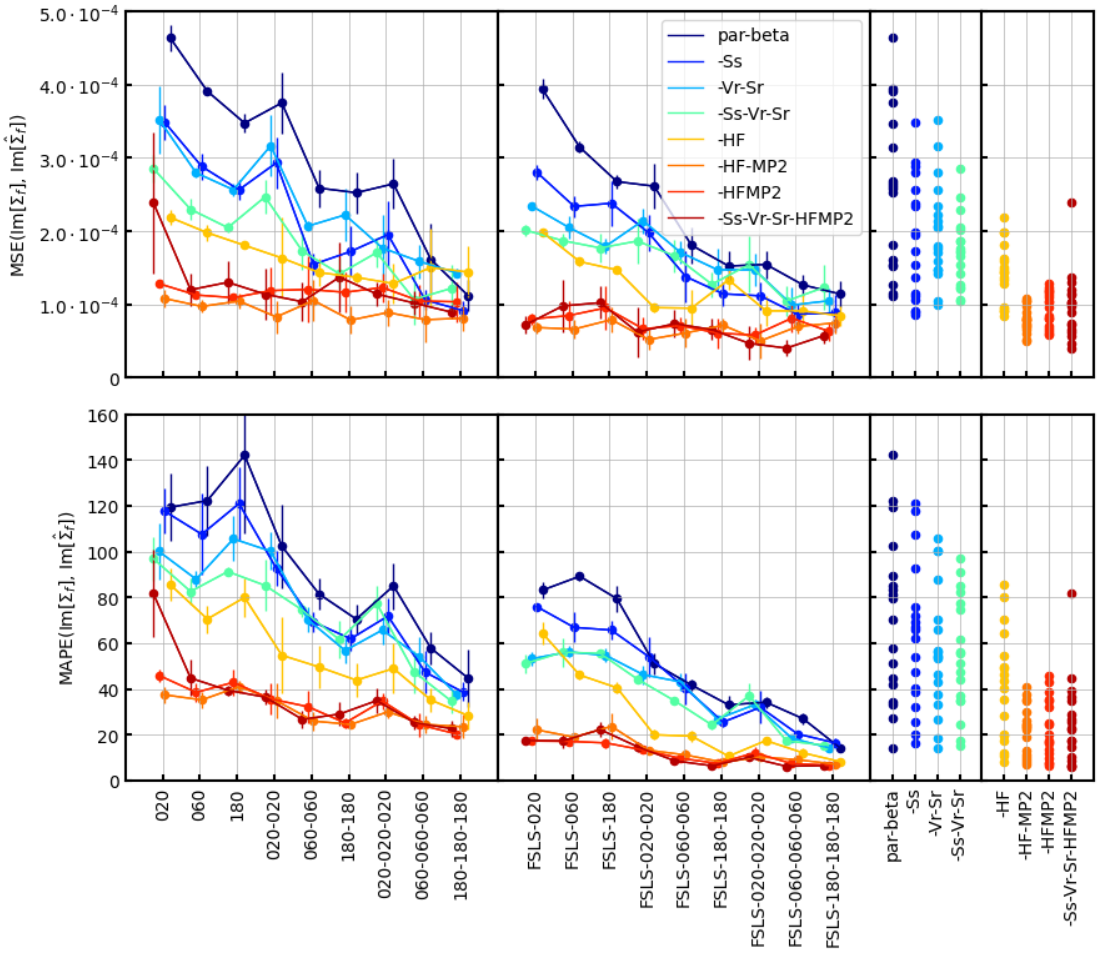

Figure 3.3: Final losses after evaluation of 5-fold cross-validation, considering only $\mathbf{Im[\Sigma_f]}$. Similar to figure (3.1), but losses are calculated based on $\mathrm{Im}[\Sigma_f]$ only. Neural networks that use feature- and label-scaling (as indicated by the prefix *FSLS*) show up to 60% smaller *MAPEs* than those that do not. This difference is less obvious when looking at the *MSE*.

3.2 Taking a look at predictions

To get a better sense of for how well these models predict the self-energy, we take a look at the predictions produced by the neural networks.

Models that were trained on the feature sets *-HF-MP2* and *-Vr-Sr* were chosen for comparison as they contain the same number of features (see table 2.2)) and can thus be considered representative for the two groups of feature sets: the ones that contain approximations based on the diagrammatic expansion of Σ_f and the ones that do not contain them.

Figure (3.4) compares the predictions of three different sets of model parameters across two different feature sets and four different networks.

- To see larger differences in the model predictions, the largest and smallest networks were selected for each feature set (the other two networks have the same size but use feature- and label-scaling).
- Figure (3.1) suggests a significant difference between models that use feature- and label scaling and those that do not. To assess this qualitatively, the predictions of these models are shown side by side.

As expected from the analysis in section (3.1), *-HF-MP2* outperforms *-Vr-Sr* for most sets of Anderson model parameters.

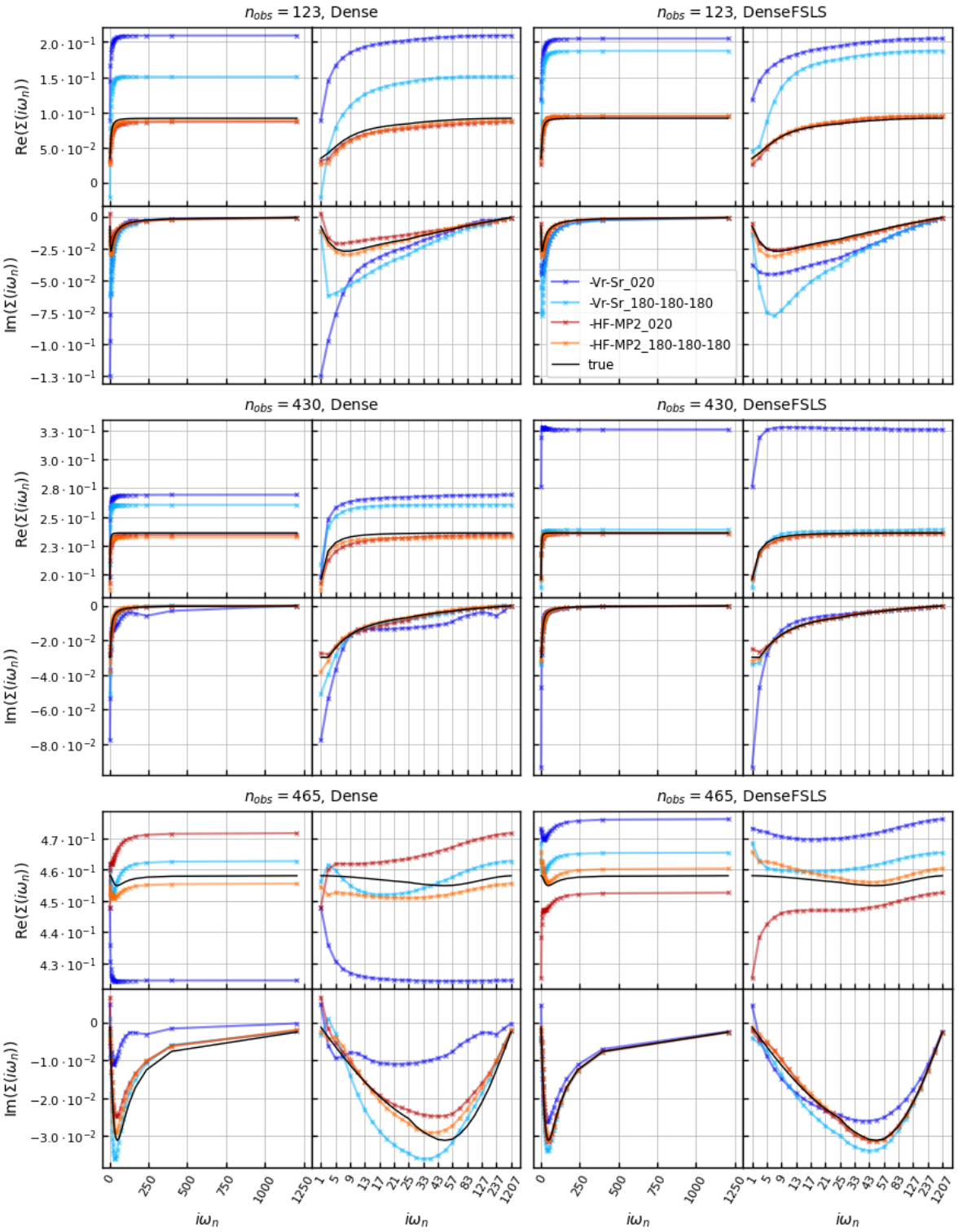

Figure 3.4: Example predictions for different sets of model parameters, feature sets and neural networks. Each of the six 4x4 plots shows the predictions of four different neural networks as well as the true self-energy for one set of model parameters, as indicated by n_{obs}. Within a 4x4 plot, the left column shows the self energy plotted over a linear spread of frequencies, which is how one would typically plot the self-energy. To be able to better see the differences in the predictions of the various neural networks, in the right column the frequencies $i\omega_n \in \mathcal{W}$ are spread equidistant. To compare models with and without feature- and label-scaling, two 4x4 plots are plotted side by side where the only difference is whether feature- and label-scaling was used for training, as indicated by *Dense* and *DenseFSLS*.

4. Conclusions and Outlook

We compared different auxiliary approximations to help neural networks predict the self-energy of an Anderson impurity model. We found that approximations based on the diagrammatic expansion of the self-energy outperformed other approximations, especially when using smaller neural networks. As the model size increases, this difference in final loss decreases. We have demonstrated the use of sparse modelling as a dimensionality reduction method for solving quantum many-body problems. We have also shown that the choice of the performance measure has a large impact on the interpretation of the learning result. It is therefore important to spend sufficient time analysing the problem in advance to be able to make an informed decision about which loss function to use.

To extend the study one could vary the amount of training data, or the number of bath sites in the impurity model. As more bath sites are added, we expect the approximations based on the simplified model with only one bath site to perform even worse.

The difference in performance of predicting real- and imaginary-parts suggest that these might require training with different hyperparameters to bring their losses to the same level.

Moving forward, two important questions need to be faced, when considering possible applications of predicting the self-energy using machine learning methods:

1. What is the computational cost of generating the data-sets and training the neural networks compared to calculating the self-energy using other tools?

2. How stable are the predictions? Is it possible to specify generalisation bounds?

To reduce the amount of computing power required to train the neural networks, one could try to use active learning techniques [Settles, 2009] to provide the training loop with more examples from specific regions of the parameter space, where the network is struggling to make accurate predictions. This could reduce the amount of data that is required for training to end up at a certain loss.

To stduy the stability of the neural networks, one could analyse how much the output varies as a function of changing the input parameters [Bousquet and Elisseeff, 2002].

A. Final Loss Tables

Table A.1: Mean MSE for each model and feature set after 5-fold cross-validation. Values are given in units of 10^{-3}.

$\times 10^{-3}$ model	par-beta	-Ss	-Vr-Sr	-Ss-Vr-Sr	-HF	-HF-MP2	-HFMP2	-Ss-Vr-Sr-HFMP2
020	2.068	1.828	2.221	2.253	0.777	0.231	0.288	0.458
060	1.468	1.327	1.380	1.291	0.621	0.210	0.251	0.277
180	1.197	1.118	1.136	1.103	0.563	0.221	0.239	0.298
020-020	1.068	0.819	0.879	0.842	0.419	0.191	0.232	0.252
060-060	0.710	0.557	0.624	0.579	0.324	0.202	0.237	0.219
180-180	0.601	0.527	0.568	0.483	0.309	0.186	0.211	0.256
020-020-020	0.713	0.561	0.477	0.486	0.365	0.151	0.207	0.222
060-060-060	0.499	0.375	0.392	0.341	0.277	0.167	0.196	0.209
180-180-180	0.329	0.319	0.317	0.323	0.230	0.165	0.170	0.179
FSLS-020	2.170	1.774	1.781	1.796	0.807	0.184	0.213	0.202
FSLS-060	1.371	1.066	1.070	1.075	0.546	0.169	0.180	0.185
FSLS-180	1.049	0.875	0.842	1.004	0.440	0.182	0.193	0.186
FSLS-020-020	1.000	0.779	0.925	0.899	0.361	0.157	0.182	0.152
FSLS-060-060	0.633	0.507	0.607	0.546	0.269	0.163	0.147	0.130
FSLS-180-180	0.449	0.444	0.481	0.447	0.236	0.153	0.151	0.126
FSLS-020-020-020	0.569	0.450	0.589	0.847	0.348	0.172	0.181	0.134
FSLS-060-060-060	0.435	0.320	0.320	0.291	0.198	0.190	0.175	0.103
FSLS-180-180-180	0.276	0.306	0.291	0.317	0.176	0.157	0.151	0.127

Table A.2: Mean MAPE for each model and feature set after 5-fold cross-validation. Values are given in units of 10^{1}.

$\times 10^{1}$ model	par-beta	-Ss	-Vr-Sr	-Ss-Vr-Sr	-HF	-HF-MP2	-HFMP2	-Ss-Vr-Sr-HFMP2
020	6.801	6.582	5.802	5.567	4.714	2.037	2.478	4.343
060	6.790	5.956	4.989	4.664	3.916	1.924	2.108	2.421
180	7.700	6.538	5.752	5.028	4.346	2.195	2.308	2.124
020-020	5.645	5.013	5.453	4.641	2.991	1.869	1.912	1.965
060-060	4.463	3.802	3.874	4.059	2.711	1.415	1.734	1.460
180-180	3.890	3.406	3.167	3.367	2.409	1.326	1.371	1.560
020-020-020	4.663	3.903	3.583	4.156	2.702	1.609	1.858	1.879
060-060-060	3.226	2.649	2.987	2.644	1.931	1.313	1.329	1.382
180-180-180	2.488	2.154	2.123	1.944	1.576	1.279	1.124	1.225
FSLS-020	4.981	4.452	3.294	3.160	3.661	1.284	1.044	1.033
FSLS-060	5.070	3.844	3.307	3.268	2.643	1.102	0.985	0.991
FSLS-180	4.450	3.701	3.181	3.201	2.314	1.318	0.950	1.236
FSLS-020-020	3.033	3.101	2.762	2.627	1.237	0.799	0.806	0.885
FSLS-060-060	2.471	2.320	2.524	2.071	1.171	0.684	0.604	0.528
FSLS-180-180	2.008	1.565	1.679	1.492	0.716	0.499	0.466	0.416
FSLS-020-020-020	2.087	1.910	1.988	2.268	1.123	0.631	0.728	0.629
FSLS-060-060-060	1.675	1.271	1.158	1.069	0.757	0.557	0.464	0.395
FSLS-180-180-180	0.941	1.026	0.912	0.965	0.526	0.452	0.418	0.409

Table A.3: MSE-based ranks of feature sets across different models.

model	par-beta	-Ss	-Vr-Sr	-Ss-Vr-Sr	-HF	-HF-MP2	-HFMP2	-Ss-Vr-Sr-HFMP2
020	6	5	7	8	4	1	2	3
060	8	6	7	5	4	1	2	3
180	8	6	7	5	4	1	2	3
020-020	8	5	7	6	4	1	2	3
060-060	8	5	7	6	4	1	3	2
180-180	8	6	7	5	4	1	2	3
020-020-020	8	7	5	6	4	1	2	3
060-060-060	8	6	7	5	4	1	2	3
180-180-180	8	6	5	7	4	1	2	3
FSLS-020	8	5	6	7	4	1	3	2
FSLS-060	8	5	6	7	4	1	2	3
FSLS-180	8	6	5	7	4	1	3	2
FSLS-020-020	8	5	7	6	4	2	3	1
FSLS-060-060	8	5	7	6	4	3	2	1
FSLS-180-180	7	5	8	6	4	3	2	1
FSLS-020-020-020	6	5	7	8	4	2	3	1
FSLS-060-060-060	8	7	6	5	4	3	2	1
FSLS-180-180-180	5	7	6	8	4	3	2	1
mean	7.6	5.7	6.5	6.3	4.0	1.6	2.3	2.2

Table A.4: MAPE-based ranks of feature sets across different models.

model	par-beta	-Ss	-Vr-Sr	-Ss-Vr-Sr	-HF	-HF-MP2	-HFMP2	-Ss-Vr-Sr-HFMP2
020	8	7	6	5	4	1	2	3
060	8	7	6	5	4	1	2	3
180	8	7	6	5	4	2	3	1
020-020	8	6	7	5	4	1	2	3
060-060	8	5	6	7	4	1	3	2
180-180	8	7	5	6	4	1	2	3
020-020-020	8	6	5	7	4	1	2	3
060-060-060	8	6	7	5	4	1	2	3
180-180-180	8	7	6	5	4	3	1	2
FSLS-020	8	7	5	4	6	3	2	1
FSLS-060	8	7	6	5	4	3	1	2
FSLS-180	8	7	5	6	4	3	1	2
FSLS-020-020	7	8	6	5	4	1	2	3
FSLS-060-060	7	6	8	5	4	3	2	1
FSLS-180-180	8	6	7	5	4	3	2	1
FSLS-020-020-020	7	5	6	8	4	2	3	1
FSLS-060-060-060	8	7	6	5	4	3	2	1
FSLS-180-180-180	6	8	5	7	4	3	2	1
mean	7.7	6.6	6.0	5.6	4.1	2.0	2.0	2.0

Bibliography

[Abadi et al., 2016] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2016). TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems.

[Anderson, 1961] Anderson, P. W. (1961). Localized Magnetic States in Metals. *Physical Review*, 124(1):41–53.

[Arsenault et al., 2014] Arsenault, L.-F., Lopez-Bezanilla, A., von Lilienfeld, O. A., and Millis, A. J. (2014). Machine learning for many-body physics: The case of the Anderson impurity model. *Physical Review B*, 90(15):155136.

[Bergstra and Bengio, 2012] Bergstra, J. and Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *The Journal of Machine Learning Research*, 13:281–305.

[Bezanson et al., 2017] Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. (2017). Julia: A Fresh Approach to Numerical Computing. *SIAM Review*, 59(1):65–98.

[Botchkarev, 2019] Botchkarev, A. (2019). A New Typology Design of Performance Metrics to Measure Errors in Machine Learning Regression Algorithms. *Interdisciplinary Journal of Information, Knowledge, and Management*, 14:045–076.

[Bousquet and Elisseeff, 2002] Bousquet, O. and Elisseeff, A. (2002). Stability and Generalization. *The Journal of Machine Learning Research*, 2:499–526.

[Bruus and Flensberg, 2004] Bruus, H. and Flensberg, K. (2004). *Many-Body Quantum Theory in Condensed Matter Physics: An Introduction*. Oxford Graduate Texts. Oxford University Press, Oxford ; New York.

[Carleo and Troyer, 2017] Carleo, G. and Troyer, M. (2017). Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606.

[Eichler et al., 2009] Eichler, A., Deblock, R., Weiss, M., Karrasch, C., Meden, V., Schönenberger, C., and Bouchiat, H. (2009). Tuning the Josephson current in carbon nanotubes with the Kondo effect. *Physical Review B*, 79(16):161407.

[Georges and Kotliar, 1992] Georges, A. and Kotliar, G. (1992). Hubbard model in infinite dimensions. *Physical Review B*, 45(12):6479–6483.

[Georges et al., 1996] Georges, A., Kotliar, G., Krauth, W., and Rozenberg, M. J. (1996). Dynamical mean-field theory of strongly correlated fermion systems and the limit of infinite dimensions. *Reviews of Modern Physics*, 68(1):13–125.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification.

[Hewson, 1997] Hewson, A. C. (1997). *The Kondo Problem to Heavy Fermions*. Number 2 in Cambridge Studies in Magnetism. Cambridge University Press, Cambridge ; New York, 1st pbk. ed. with corrections edition.

[Hyndman and Koehler, 2006] Hyndman, R. J. and Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688.

[Japkowicz and Shah, 2011] Japkowicz, N. and Shah, M. (2011). *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, Cambridge.

[Karpathy, 2019] Karpathy, A. (2019). A Recipe for Training Neural Networks. http://karpathy.github.io/2019/04/25/recipe/.

[Latil et al., 2004] Latil, S., Roche, S., Mayou, D., and Charlier, J.-C. (2004). Mesoscopic Transport in Chemically Doped Carbon Nanotubes. *Physical Review Letters*, 92(25):256805.

[Li et al., 2020] Li, J., Wallerberger, M., Chikano, N., Yeh, C.-N., Gull, E., and Shinaoka, H. (2020). Sparse sampling approach to efficient ab initio calculations at finite temperature. *Physical Review B*, 101(3):035144.

[Meir et al., 1993] Meir, Y., Wingreen, N. S., and Lee, P. A. (1993). Low-temperature transport through a quantum dot: The Anderson model out of equilibrium. *Physical Review Letters*, 70(17):2601–2604.

[Nair and Hinton, 2010] Nair, V. and Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. *In Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

[Radmanovac, 2024] Radmanovac, D. (2024). *Predicting the Self-Energy of Anderson Impurity Models Using Kernel Ridge Regression*. Project Work, TU Wien, Vienna.

[Rigo and Mitchell, 2020] Rigo, J. B. and Mitchell, A. K. (2020). Machine learning effective models for quantum systems. *Physical Review B*, 101(24):241105.

[Santurkar et al., 2018] Santurkar, S., Tsipras, D., Ilyas, A., and Madry, A. (2018). How Does Batch Normalization Help Optimization? In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

[Sau et al., 2008] Sau, J. D., Neaton, J. B., Choi, H. J., Louie, S. G., and Cohen, M. L. (2008). Electronic Energy Levels of Weakly Coupled Nanostructures: C60-Metal Interfaces. *Physical Review Letters*, 101(2):026804.

[Schrieffer and Wolff, 1966] Schrieffer, J. R. and Wolff, P. A. (1966). Relation between the Anderson and Kondo Hamiltonians. *Physical Review*, 149(2):491–492.

[Settles, 2009] Settles, B. (2009). Active Learning Literature Survey. Technical Report, University of Wisconsin-Madison Department of Computer Sciences.

[Shinaoka et al., 2022] Shinaoka, H., Chikano, N., Gull, E., Li, J., Nomoto, T., Otsuki, J., Wallerberger, M., Wang, T., and Yoshimi, K. (2022). Efficient ab initio many-body calculations based on sparse modeling of Matsubara Green's function. *SciPost Physics Lecture Notes*, page 63.

[Shinaoka et al., 2017] Shinaoka, H., Otsuki, J., Ohzeki, M., and Yoshimi, K. (2017). Compressing Green's function using intermediate representation between imaginary-time and real-frequency domains. *Physical Review B*, 96(3):035147.

[Sturm et al., 2021] Sturm, E. J., Carbone, M. R., Lu, D., Weichselbaum, A., and Konik, R. M. (2021). Predicting impurity spectral functions using machine learning. *Physical Review B*, 103(24):245118.

[Troyer and Wiese, 2005] Troyer, M. and Wiese, U.-J. (2005). Computational Complexity and Fundamental Limitations to Fermionic Quantum Monte Carlo Simulations. *Physical Review Letters*, 94(17):170201.

[Varma and Yafet, 1976] Varma, C. M. and Yafet, Y. (1976). Magnetic susceptibility of mixed-valence rare-earth compounds. *Physical Review B*, 13(7):2950–2954.

[Wallerberger et al., 2023] Wallerberger, M., Badr, S., Hoshino, S., Huber, S., Kakizawa, F., Koretsune, T., Nagai, Y., Nogaki, K., Nomoto, T., Mori, H., Otsuki, J., Ozaki, S., Plaikner, T., Sakurai, R., Vogel, C., Witt, N., Yoshimi, K., and Shinaoka, H. (2023). Sparse-ir: Optimal compression and sparse sampling of many-body propagators. *SoftwareX*, 21:101266.

[Wingreen and Meir, 1994] Wingreen, N. S. and Meir, Y. (1994). Anderson model out of equilibrium: Noncrossing-approximation approach to transport through a quantum dot. *Physical Review B*, 49(16):11040–11052.

[Zagoskin, 2014] Zagoskin, A. (2014). *Quantum Theory of Many-Body Systems: Techniques and Applications*. Graduate Texts in Physics. Springer International Publishing, Cham.