MedeA MLPG: Generating Machine Learned-Potentials from First Principles Data

Contents

1 Introduction

The *Machine-Learned Potential Generator* employs energies, forces, and stresses computed from first principles to create machine-learned potentials (MLPs), which in the same manner as classical forcefields are created for use within the *MedeA LAMMPS* environment to calculate properties of interest. A well trained MLP can be applied to arbitrary conformations, as long as these are covered by the training set, with comparable accuracy as the first principles calculations used to create the potential, but at a much lower computational cost. Nevertheless, the predictive power of MLPs may degrade when it comes to extrapolations outside the range covered by the training set.

The *MedeA Machine-Learned Potential Generator* provides access to different approaches for the generation of machine-learned potentials. To be specific, Spectral Neighbor Analysis Potentials (SNAP) and Neural Network Potentials (NNP) are supported and can be used with *MedeA LAMMPS*. SNAP potentials are created by employing the FitSNAP code [1], [3], [5] in combination with Powell's Derivative Free Optimizer to optimize the so-called hyper-parameters as, e.g., the cutoff radii and the relative radii and weights of the various atom types [7]. In contrast, NNPs are generated using the n2p2 code by Singraber *et al.* [9] with the weights and biases of the neural network composed of two hidden layers, each containing a user-definable number of neurons, optimized with the Kalman filter fading memory approach. To use the thus generated NNP in *MedeA* the n2p2 LAMMPS interface by Singraber *et al.* [11] is used.

The theory behind these two MLP approaches is discussed in detail in the section on the Theory of Machine-Learned Potentials.

[1] A. P. Thompson, L. P. Swiler, C. R. Trott, S. M. Foiles, and G. J. Tucker, *Spectral neighbor analysis method for automated generation of quantum-accurate interatomic potentials*, J. Comp. Phys. **285**, 316 (2015) (DOI)

[3] M. A. Wood and A. P. Thompson, *Extending the accuracy of the SNAP interatomic potential form*, J. Chem. Phys. **148**, 241721 (2018) (DOI)

[5] M. A. Cusentino, M. A. Wood, and A. P. Thompson, *Explicit Multielement Extension of the Spectral Neighbor Analysis Potential for Chemically Complex Systems*, J. Phys. Chem. A **124**, 5456 (2020) (DOI)

[7] M. J. D. Powell, *The NEWUOA software for unconstrained optimization without derivatives*, In: G. Di Pillo and M. Roma (eds) *Large-Scale Nonlinear Optimization. Nonconvex Optimization and Its Applications* **83** (Springer, Boston 2006) (DOI)

[9] A. Singraber, T. Morawietz, J. Behler, and C. Dellago, *Parallel Multistream Training of High-Dimensional Neural Network Potentials*, J. Chem. Theory Comput. **15**, 3075-3092 (2019) (DOI)

[11] A. Singraber, J. Behler and C. Dellago, *Library-Based LAMMPS Implementation of High-Dimensional Neural Network Potentials*, J. Chem. Theory Comput. **15**, 1827-1840 (2019) (DOI)

2 Working with the Machine-Learned Potential Generator

The generation and optimization of MLPs typically proceeds in three steps. First, a training set is created from first principles calculations for a number of different material systems varying in structure and possibly also in chemical composition. Second, an MLP is derived by minimizing the deviations of the energies, forces, and possibly stresses arising from the first principles and MLP descriptions. Finally, the MLP is validated by predicting selected properties for material systems not included in the training set and comparing the results to those obtained from first principles.

3 Training Set Creation

The first task, before generating an MLP, is to define and create a suitable initial training data set. This data set determines the material system and the physical/chemical properties that can be described by the MLP. Examples of material properties that can be targeted include lattice parameters, elastic coefficients, free energies of bulk structures, as well as phase stabilities, surface energies, stacking fault energies, defect (e.g., vacancy and interstitial) formation energies, phonon dispersions, diffusion coefficients, and thermal expansion coefficients. This basic information guides the creation of structural models to be included in the training set.

It is important to note that the training set can be successively extended in case interest in materials properties other than those initially aimed at arises. In this case, new structures can be simply added to the original training set and the generation of the MLP repeated with the extended training set. Since the overall effort to generate an MLP is determined by the computational effort of the *ab initio* calculations for the training set structures rather than the fit procedure, the above is an efficient approach.

Usually, in addition to ground state structures, distorted structures must also be considered. Distorted structures may deviate from the ground state structures by isotropic or uniaxial compressions or expansions, or by shear strains. Additional structures that may be required include structures with vacancies and/or interstitial atoms. Finally, structures selected from *ab initio* molecular dynamics (MD) simulations should also be included. Ideally, the MD-derived structures should be generated at temperatures covering the range to be described by the MLP.

Once the training set structures have been created with the tools provided by the *MedeA* software, *ab initio* calculations are performed. In this step, it is important to use the same computational parameters for all calculations to guarantee consistency of the training set data, i.e., the energies, forces, and stresses. Consistency of the computational parameters clearly reduces systematic errors intrinsic to the chosen DFT functional. The results of all these calculations are stored in the form of a *MedeA* fitting training set which is created using the *MedeA* Fitting Data Manager (cf. with Fitting Data Manager).

Note: All structures in the training set are required to have P1 symmetry to be considered by the *Machine-Learned Potential Generator*.

Note: Always ensure that the computational parameters of all DFT calculations used for the training set are compatible with particular attention on the PAW potentials, the planewave cutoff energy, and the k-point spacing.

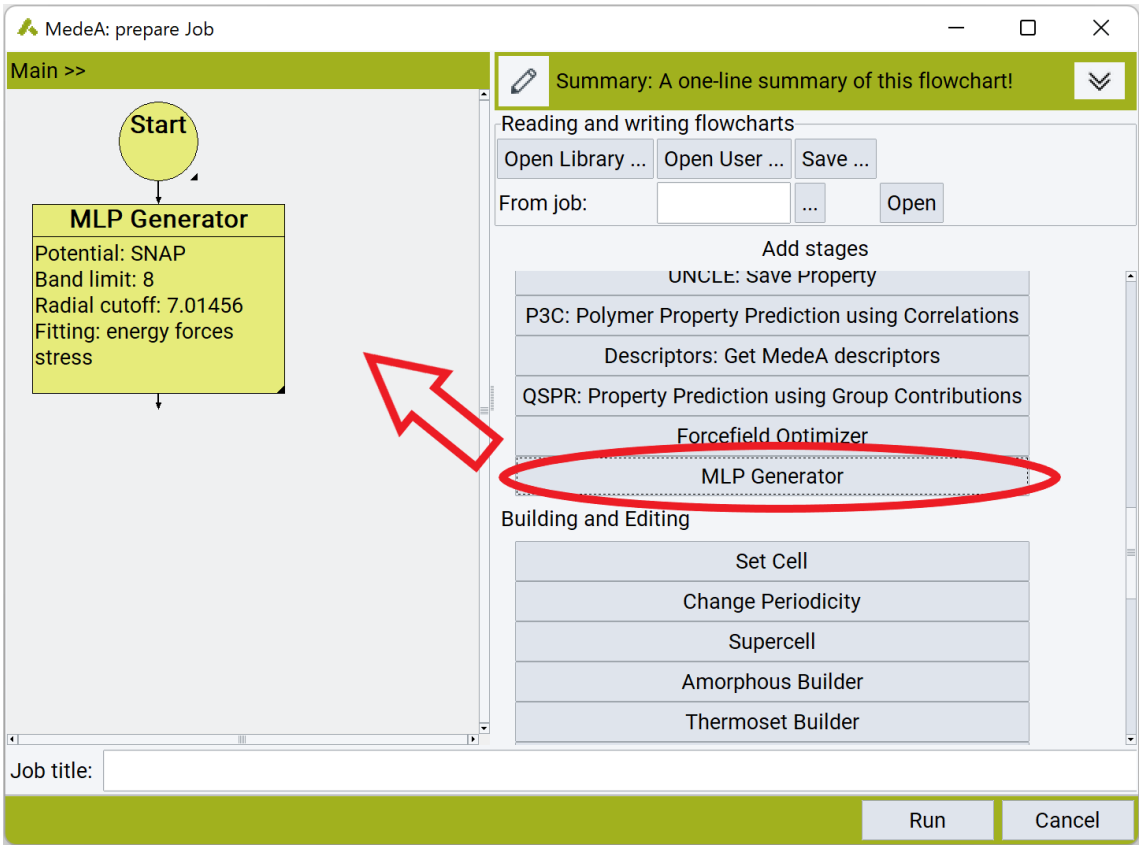
The optimum planewave cutoff energy value depends on the PAW potentials used for the elements of interest and should be set to the highest value of all potentials used for the training set calculations.

If the system is a metal then a k-mesh spacing below **0.3** 1/Ang should be used.

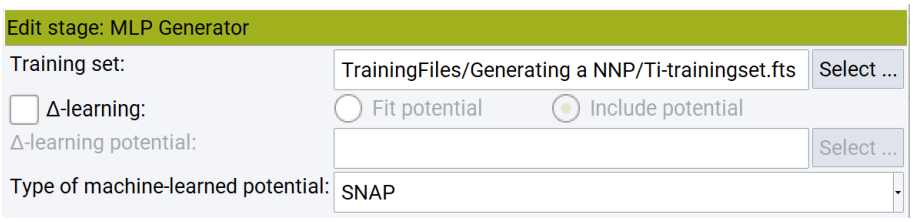
---

4 Setting up and Generating a Machine-Learned Potential

The *Machine-Learned Potential Generator* is accessed within the *MedeA* flowchart paradigm. This can be done by opening the flowchart interface with Jobs >> New Job... and adding an **MLP Generator** stage to the empty flowchart.



Configure the *Machine-Learned Potential Generator* by double-clicking on the **MLP Generator** stage. A variety of options are offered.



The first step consists of selecting the fitting training set defining the Training set to be used to generate the MLP. Use the Select button to open a file selection dialog.

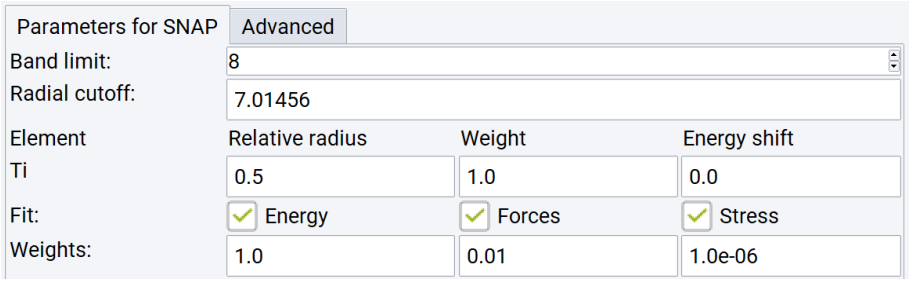
The checkbox Δ -learning can be used to request the fitting or the use of a Δ-potential. If checked, the radio buttons Fit potential and Include potential become active. The Fit potential button can be chosen if the training set selected above contains data for the fitting of a Δ-potential. In this case a Δ-potential will be created. An existing Δ-potential can be used in a fit by choosing the Include potential button. The entry field Δ -learning potential will become active and requires the selection of a forcefield file with a Δ-potential.

Next, the user can select the type of machine-learned potential with the combo box Type of machine-learned potential. Currently both **SNAP** and **NNP** are supported. Different parameters are shown for each of these two options.

4.1 SNAP

When the **SNAP** machine-learned potential is selected, two configuration tabs: **Parameters for SNAP**, and **Advanced**, become available.

Parameters for SNAP



The SNAP fit is controlled by the following parameters in this tab:

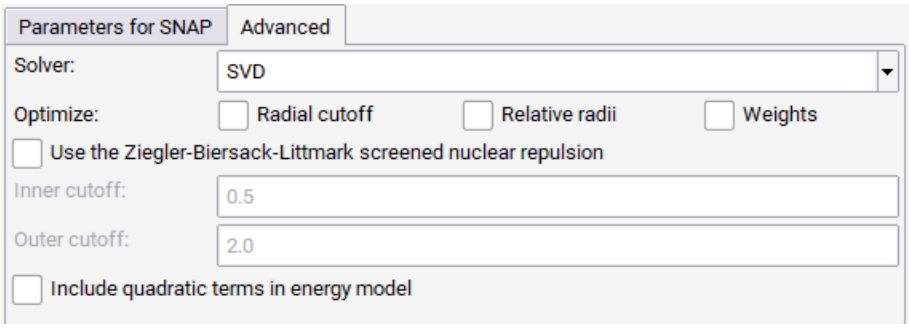
Band limit relates to the maximum angular momentum used in the spherical harmonics expansion of the local atomic descriptions defining the angular accuracy of potential. Only even values are allowed.

Radial cutoff is a factor combined with the entries for Relative radius and multiplying the sum of two atom-type specific relative radii to determine the cutoff radius used for a particular atom in the local atomic cluster of another atom. Hence, the cutoff radius applied for a particular local atomic cluster depends on the types of both the central atom and any other atom in that cluster.

For each element in the training set the Relative radius, Weight and Energy shift can be specified. While Weight allows to distinguish different atom types in the expansion of the density of neighbor atoms in the local atomic environments, Relative radius specifies an atom's contribution to the atom-type dependent cutoff radius, which is obtained from adding the relative radii of the central atom of a local atomic environment and a particular atom of the environment and then multiplying with Radial cutoff. Finally, Energy shift specifies an additive energy shift applied to a particular atom-type prior to the actual fitting procedure.

The user can toggle if Energy, Forces, and Stress are considered in the fit. If toggled their weights can be defined. Although setting a particular weight to zero would have the same effect as unchecking the respective box, using the check box is more efficient and robust as it reduces the size of the underlying numerical problem.

Advanced



The following parameters, requiring more detailed knowledge on SNAP fitting, can be configured in the **Advanced** tab:

The Solver combo box allows the user to select the algorithm for solving the linear equations. Possible options are SVD, LASSO, RIDGE, and ELASTIC.

The three checkboxes labeled Radial cutoff, Relative radii and Weights can be set to perform an optimization of the corresponding parameters instead of defining them by hand. Since the optimization of these parameters requires a larger number of fits (on the order of 100) the run time will be significantly prolonged when set. Nevertheless, optimization of these so-called hyperparameters is strongly recommended.

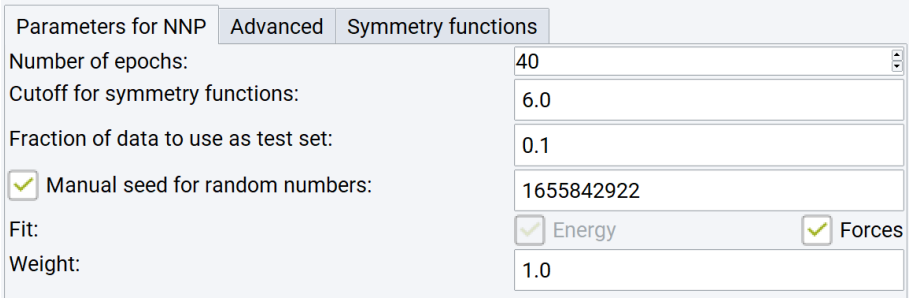
With Use the Ziegler-Biersack-Littmark screened nuclear repulsion the Ziegler-Biersack-Littmark potential can be added to the machine-learned potential. The inner and outer cutoffs of the Ziegler-Biersack-Littmark potential can be controlled with the variables Inner cutoff and Outer cutoff.

The checkbox Include quadratic terms in energy model can be used to extend SNAP with quadratic terms. Note that including these terms considerably increases the computational effort and memory usage. For this reason, it is recommended to reduce the band limit once the Include quadratic terms in energy model box is checked.

4.2 NNP

When the **NNP** machine-learned potential is selected, three configuration tabs: **Parameters for NNP**, **Advanced**, and **Symmetry functions**, become available.

Parameters for NNP



Basic NNP fit options are controlled in this tab. The following variables can be set:

Number of epochs determines the total number of epochs.

Cutoff for symmetry functions describes the cutoff radius of the default symmetry functions set. This variable is discarded when a custom set of symmetry functions in the **Symmetry functions** tab is defined.

Fraction of data to use as test set sets the ratio of structures to be randomly transferred from the training set to the validation set.

When Manual seed for random numbers is set the initial seed for the random number generator is user-definable, otherwise a random value is used for the initial seed.

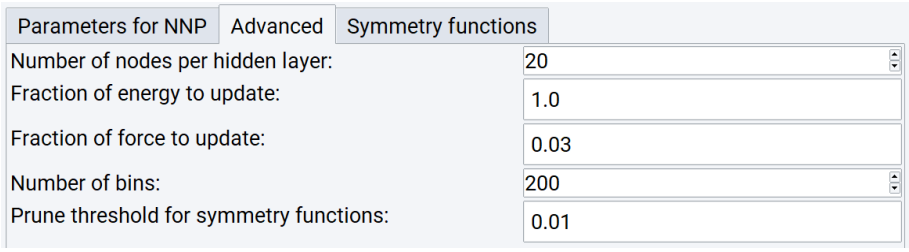
With the check box Fit Forces the user can toggle whether forces are considered in addition to the energies in the NNP fit. If set the weight of these forces compared to the energies can be set in the related Weight entry field.

Advanced

The following parameters, requiring more detailed knowledge on NNP fitting, can be configured in the **Advanced** tab:

Number of nodes per hidden layer defines the number of nodes in both the hidden layers of the neural network.

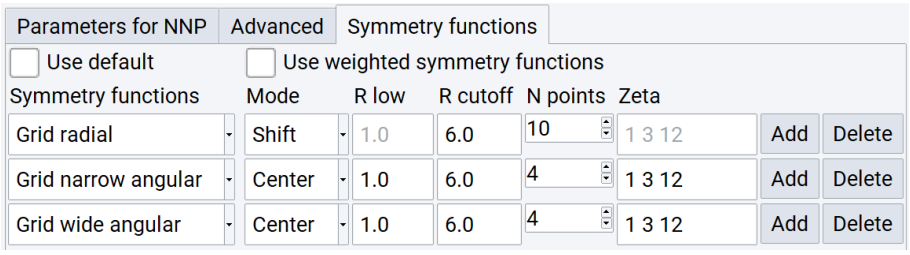
The fields Fraction of energy to update and Fraction of force to update define the fraction of energy and force updates per epoch.



Number of bins defines the number of bins for the symmetry function histograms.

Prune threshold for symmetry functions sets the pruning range threshold. All symmetry functions with an activation range below this value will be removed.

Symmetry functions



If desired the symmetry functions can be customized in this tab. Any custom defined symmetry functions will override the cutoff radius setting in the **Parameters for NNP** tab. Otherwise the current default set of descriptors is used; 10 shifted radial symmetry functions on a grid as defined by the method proposed by Imbalzano *et al.* [13] , i.e., $n_r = 10$ (see Theory of Machine-Learned Potentials); centered narrow and wide angular functions generated with $\lambda = \{-1, +1\}$, $\zeta = \{1, 3, 12\}$ on a grid based on Gastegger *et al.* [14] , $n_a = 4$, and $r_{\text{low}} = 1$.

For most users, it should be sufficient to stay with the default set of symmetry functions with the check box Use default . If this check box is unchecked the set of symmetry functions becomes user-definable.

Toggle with Use weighted symmetry functions between weighted and unweighted symmetry functions [14] .

Symmetry functions sets can be added and removed with the Add and Delete buttons. The first column of the table **Symmetry functions** allows the definition of the symmetry functions to be set to Grid radial to define radial functions on a grid [13] (Theory of Machine-Learned Potentials), or to Grid narrow angular for narrow angular functions on a grid, or to Grid wide angular for wide angular symmetry functions on a grid. Both angular grid definitions are based on the work by Gastegger *et al.* [14]. With the second Mode column it is possible to select whether the symmetry function should be **Centered** or **Shifted**. In the column R low the lower bound, used to define the angular symmetry functions on the grid, is set, and in the column R cutoff the cutoff radius for all symmetry functions is specified (this overwrites the cutoff specified in the **Parameters for NNP** tab). The column N points configures the number of grid points on which the given set of symmetry functions is defined. The last column Zeta defines the exponents of the angular symmetry functions and can be specified as a space-separated list. Its lowest value should be equal or greater than one and this column is only active for angular symmetry functions.

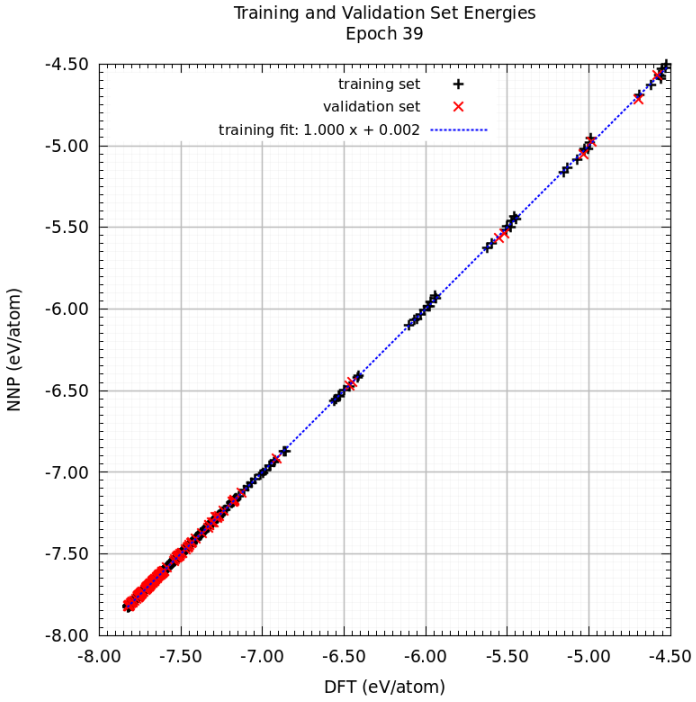
[13] G. Imbalzano, A. Anelli, D. Giofré, S. Klees, J. Behler, and M. Ceriotti, J. Chem. Phys. **148**, 241730 (2018)
[14] M. Gastegger, L. Schwiedrzik, M. Bittermann, F. Berzsenyi, and P. Marquetand, J. Chem. Phys. **148**, 241709 (2018)

4.3 Assessing the Fit Obtained by the *Machine-Learned Potential Generator*

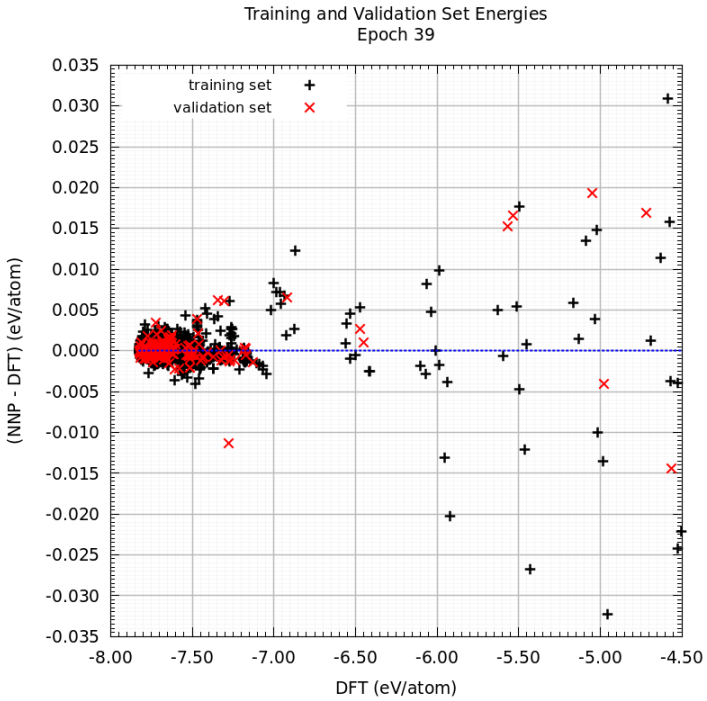
The Job.out file of the *Machine-Learned Potential Generator* provides information about the quality of the fit. A report of the deviations of the properties obtained from the MLP from those obtained from the previous first principles calculations, with rows for the energies, forces, and stresses are presented. The details of the presentation in the Job.out file depend on the use of the potential (**SNAP** or **NNP**) with the *Machine-Learned Potential Generator*.

In addition to the statistical information contained in Job.out, graphical output is also provided by the files:

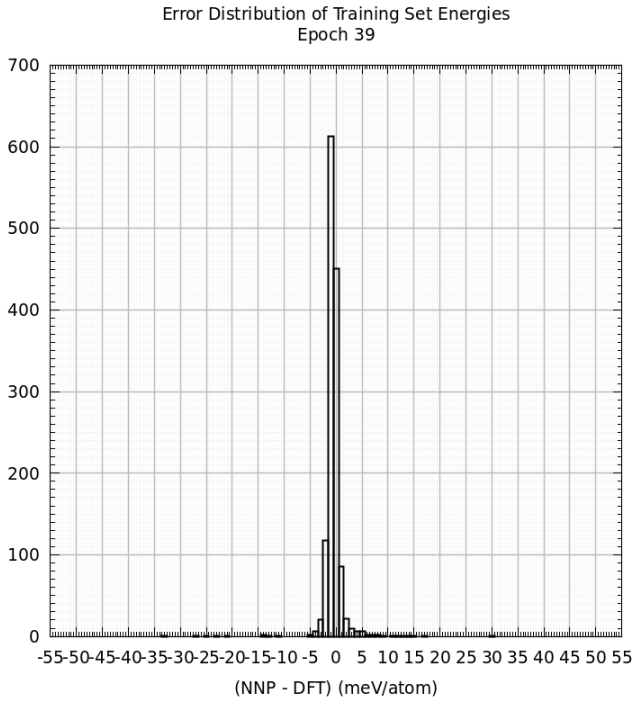
- **Energies_DFT_vs_MLP.png** shows the correlation between DFT calculated energies and MLP predicted ones for both the training (black crosses) and the validation (red crosses) set. Note that validation data is only shown when generating a **NNP**:



- **Energies_Difference_DFT_vs_MLP.png** shows the deviation of MLP predicted energies from the respective DFT values:

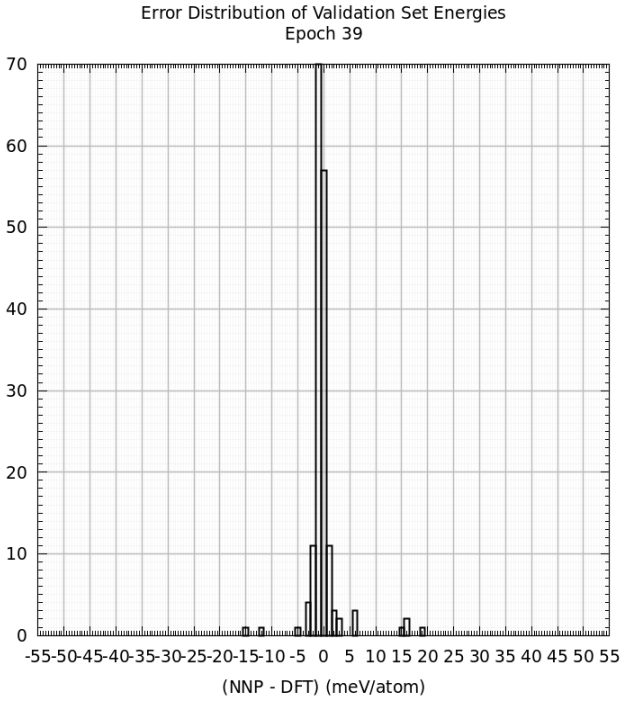


- **Energies_Error_Distribution_TrainingSet.png** contains a histogram of the errors of the MLP predicted energies for the training set:

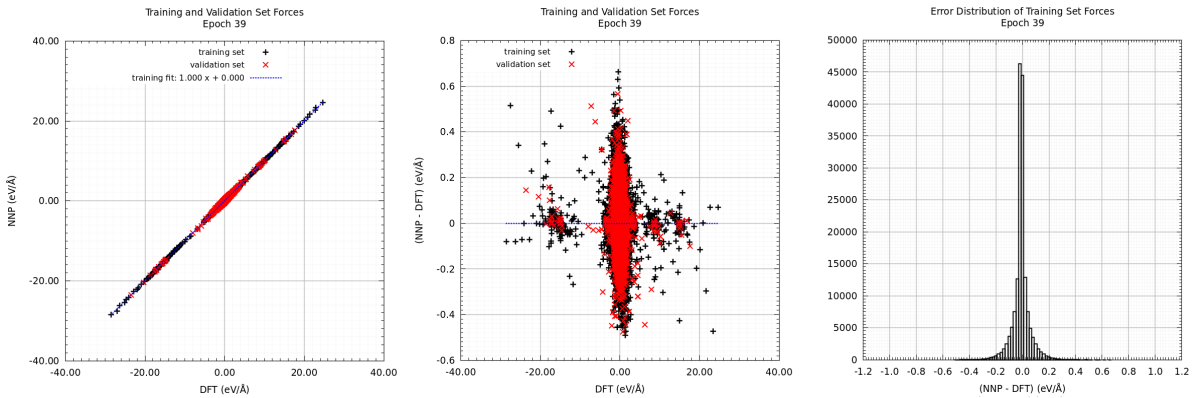


- **Energies_Error_Distribution_ValidationSet.png** contains a histogram of the errors of the MLP pre-

dicted energies for the validation set:

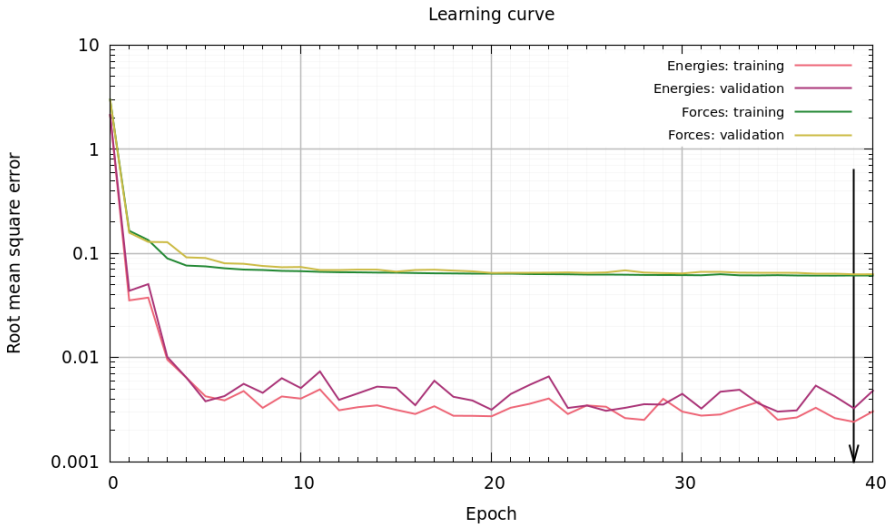


- Corresponding plots are available for the forces:



Depending on the configuration of the **MLP Generator** stage some of the plots listed above might not be generated. In the DFT vs. MLP plots, the property as calculated with the MLP is shown as a function of the DFT property. Ideally, the plot should show a straight line with slope one. The correlation between DFT and MLP property is also shown in the plot as an equation. A perfect fit would correspond in this equation to a slope of one and a slope-intercept of zero. The difference DFT vs. MLP plots show the deviation of the property calculated with the MLP from the corresponding DFT values as a function of the DFT property. These deviations would be zero in the ideal case. The error distribution plots contain histograms of the errors of the property. These distributions should be as narrow as possible and centered at zero. Ideally, only a single bar at zero should exist.

In case of NNPs, there is also a plot *learning-curve.png*, which shows the root mean squared error as a function of the epoch for energies and forces in training and test sets. For orientation, the optimum epoch is indicated in this plot by a black arrow.



4.4 Using the Potential Created by the *Machine-Learned Potential Generator*

If the *Machine-Learned Potential Generator* achieved satisfactory agreement with the first principles training set the MLP can be used in production calculations with *MedeA* LAMMPS. Each run of the *Machine-Learned Potential Generator* creates an *frc* file with the parameters of the MLP. This file can be found in the folder of the corresponding flowchart stage. Its exact location is given in the Job.out file. Save this file on your machine in *MD/data/Forcefields/custom* and read it in with Forcefields >> Read ... . When running calculations on an external JobServer this forcefield file is automatically transferred with all other data of the job.

Note: In any publication of results generated with the *MedeA Machine-Learned Potential Generator* Refs. [1], [3], [5] and Refs. [9] and [11], respectively, should be cited if **SNAP** and **NNP** have been used.

DOCUMENTATION