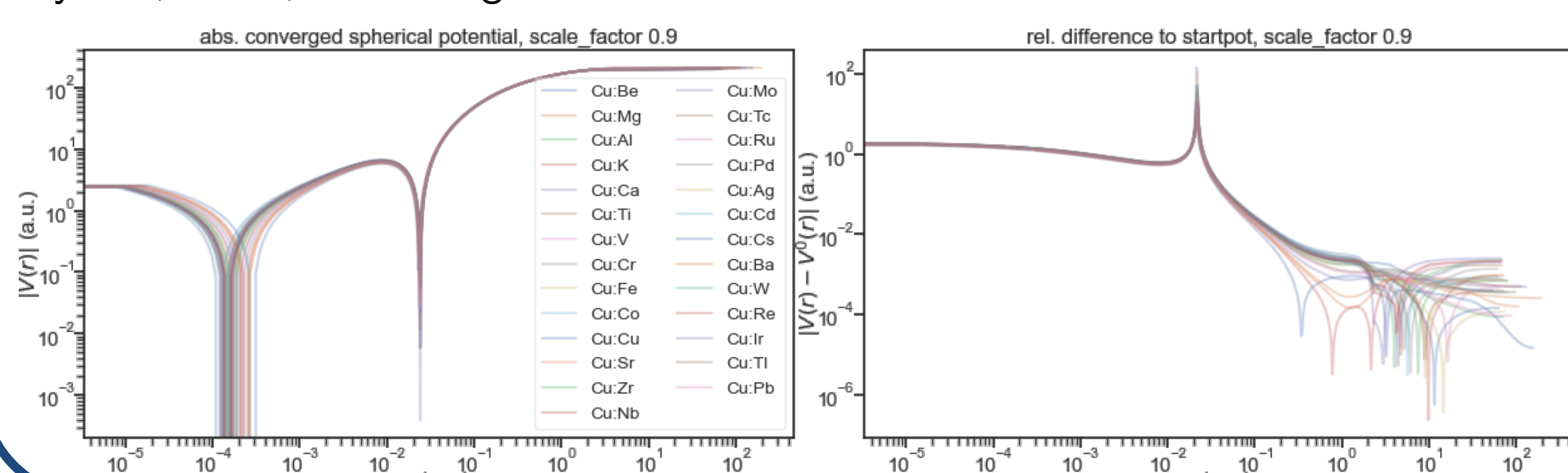
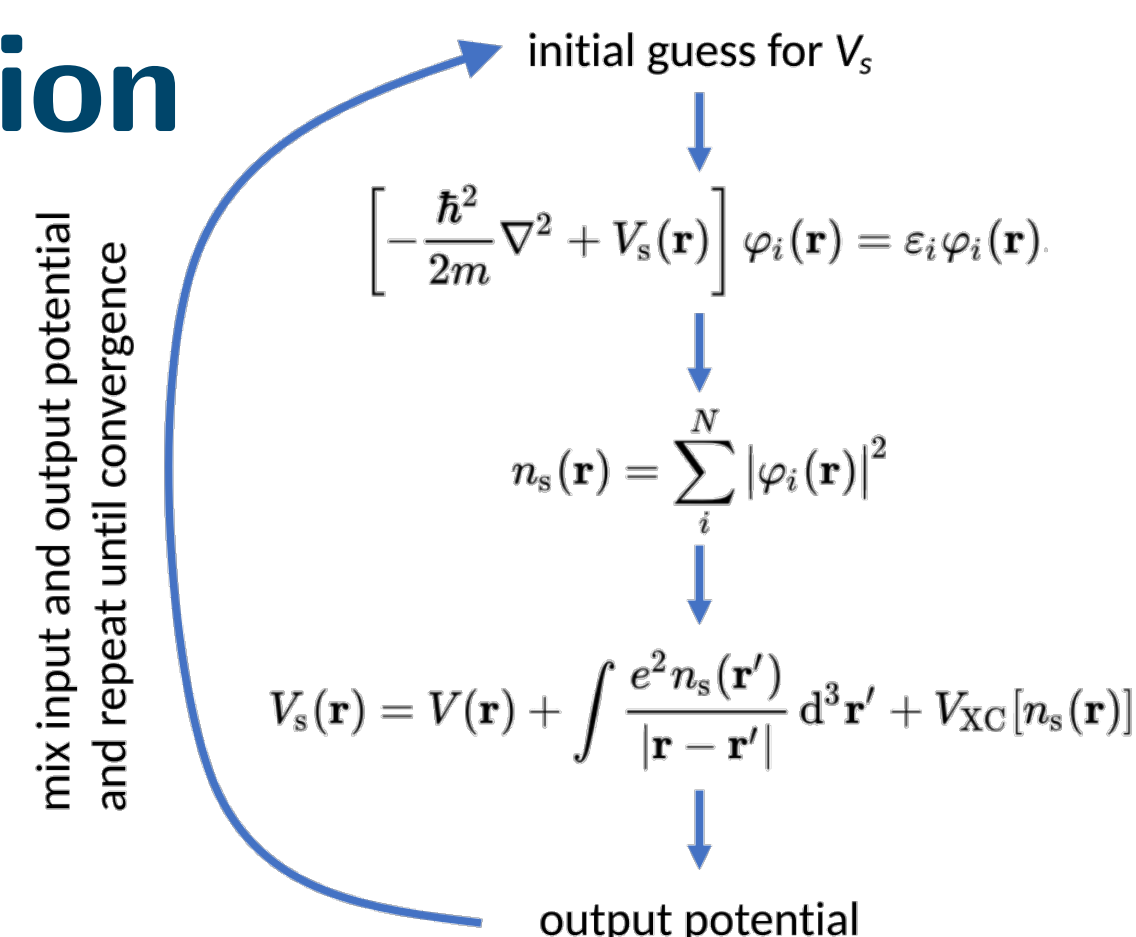


Comparison of structural representations for machine learning-accelerated ab initio calculations

Johannes Wasmer, Philipp Rüßmann and Stefan Blügel
Peter Grünberg Institute (PGI-1) and Institute for Advanced Simulation (IAS-1),
Forschungszentrum Jülich

Introduction

• Here we investigate the feasibility of accelerating the density functional theory code juKKR [1] with machine learning potentials (MLPs) by developing a surrogate model. Its plugin aiida-kkr [2] for the computational infrastructure platform AiiDA [3] allows to build complex workflows with ease. With it, we have created a database of 7100 embeddings of single-atom impurities into elemental host crystals, so far, as training data.



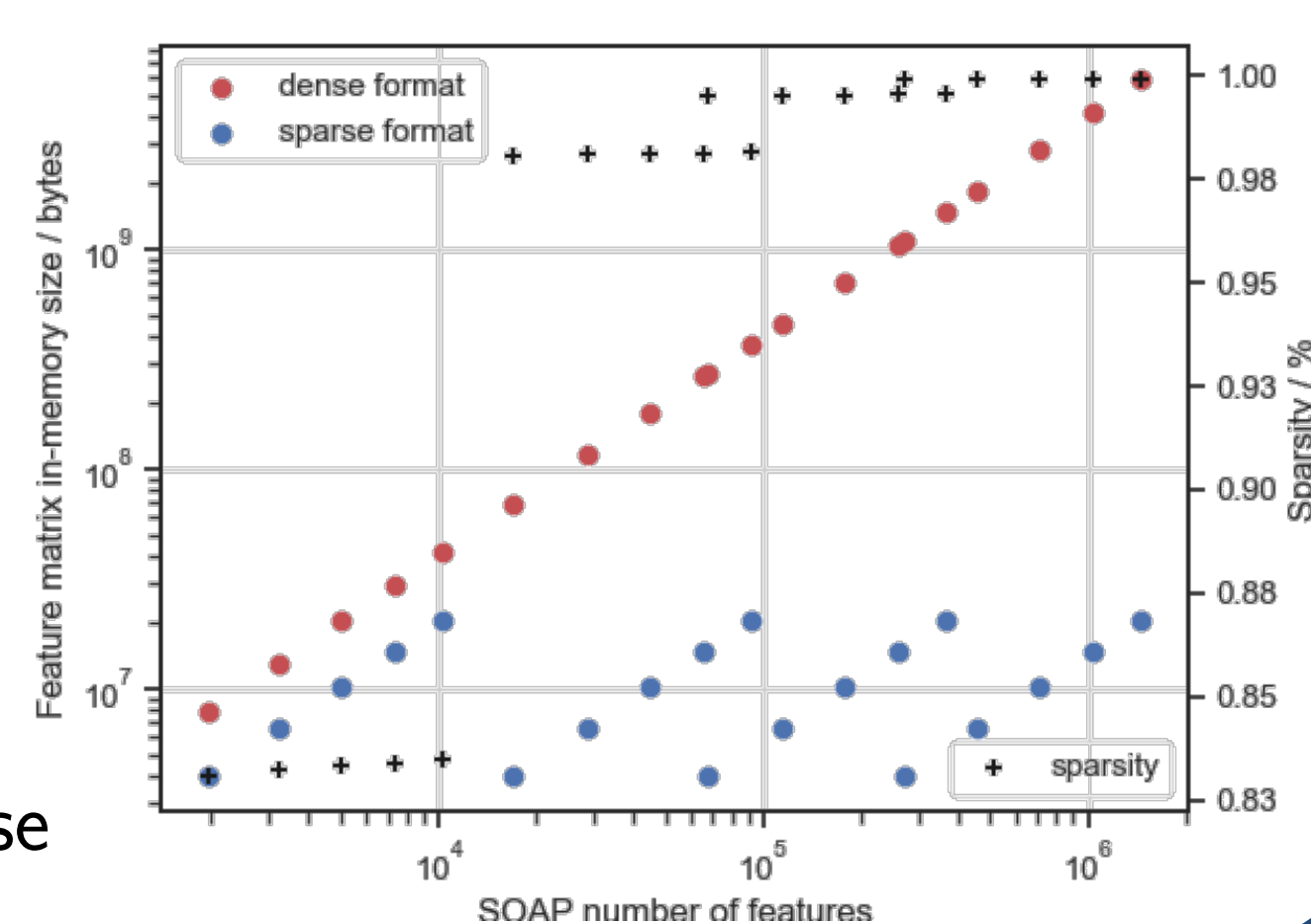
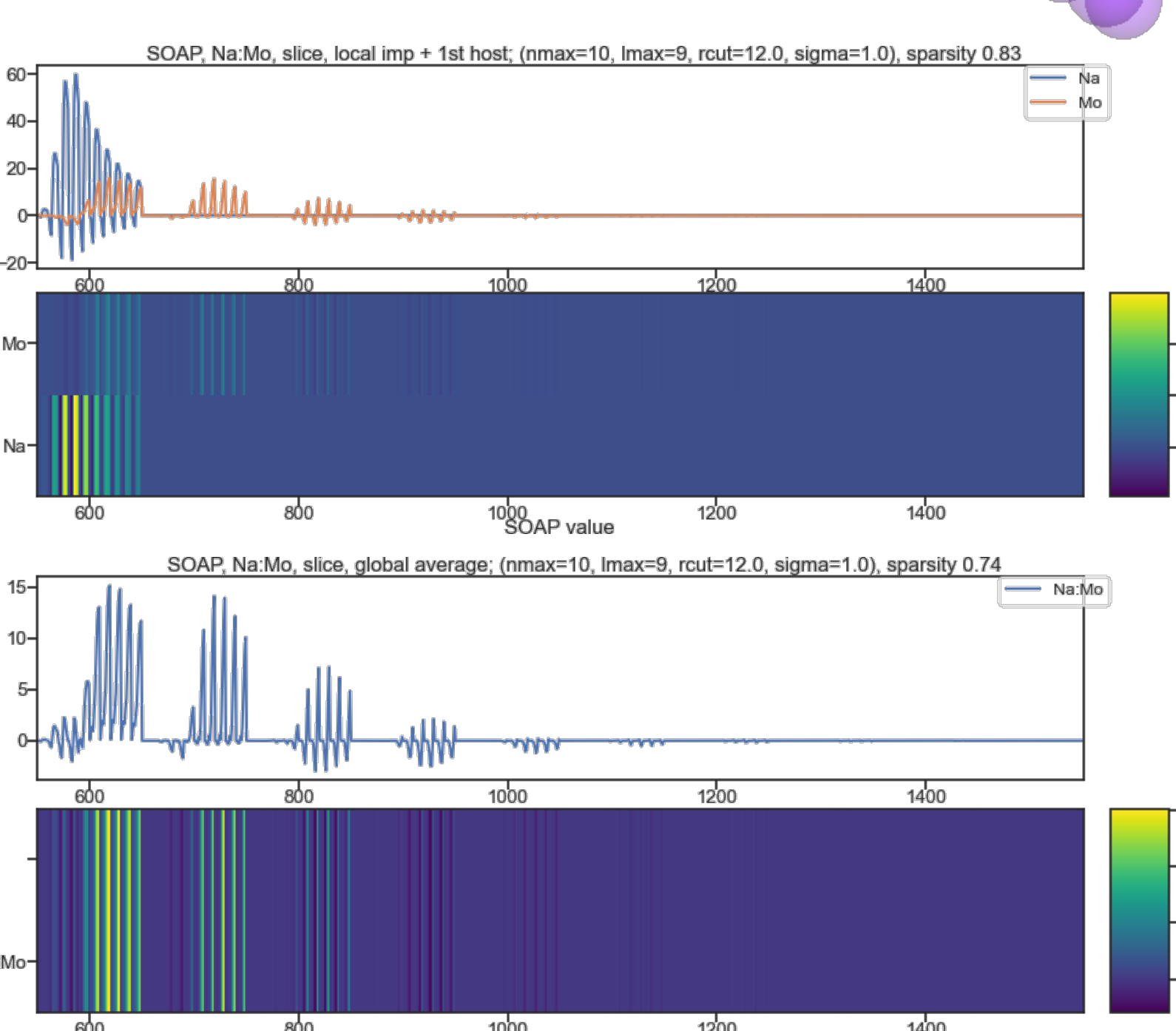
Representing structures

• Structural representations or 'descriptors' let a learner infer a target property solely from the system's geometry and chemical species. They produce a feature vector or 'fingerprint' of the system.

• The Dscribe library [4] offers a variety of descriptors. As concept illustration, we focus here on its implementation of the Smooth Overlap of Atomic Orbitals descriptor (SOAP) based on expansions of atomic density fields in spherical harmonics

and radial basis functions around atom centers controlled by l_{\max} and n_{\max} . This makes it permutation-, translation- and rotation-invariant as well as composable to a global descriptor by averaging over multiple centers.

• The feature vector length depends on the resolution of this expansion and number of species, as the plot shows (number of species=[5,15,30,60], resolution $n_{\max}=l_{\max}$ =[5,6,7,8,9], $N=1000$ structures). Our dataset has 60 species. The feature matrix is 99.9% sparse, and claims around 50 GB in memory at high resolution. This constrains estimator availability to sparse or out-of-memory learning.



Acknowledgments

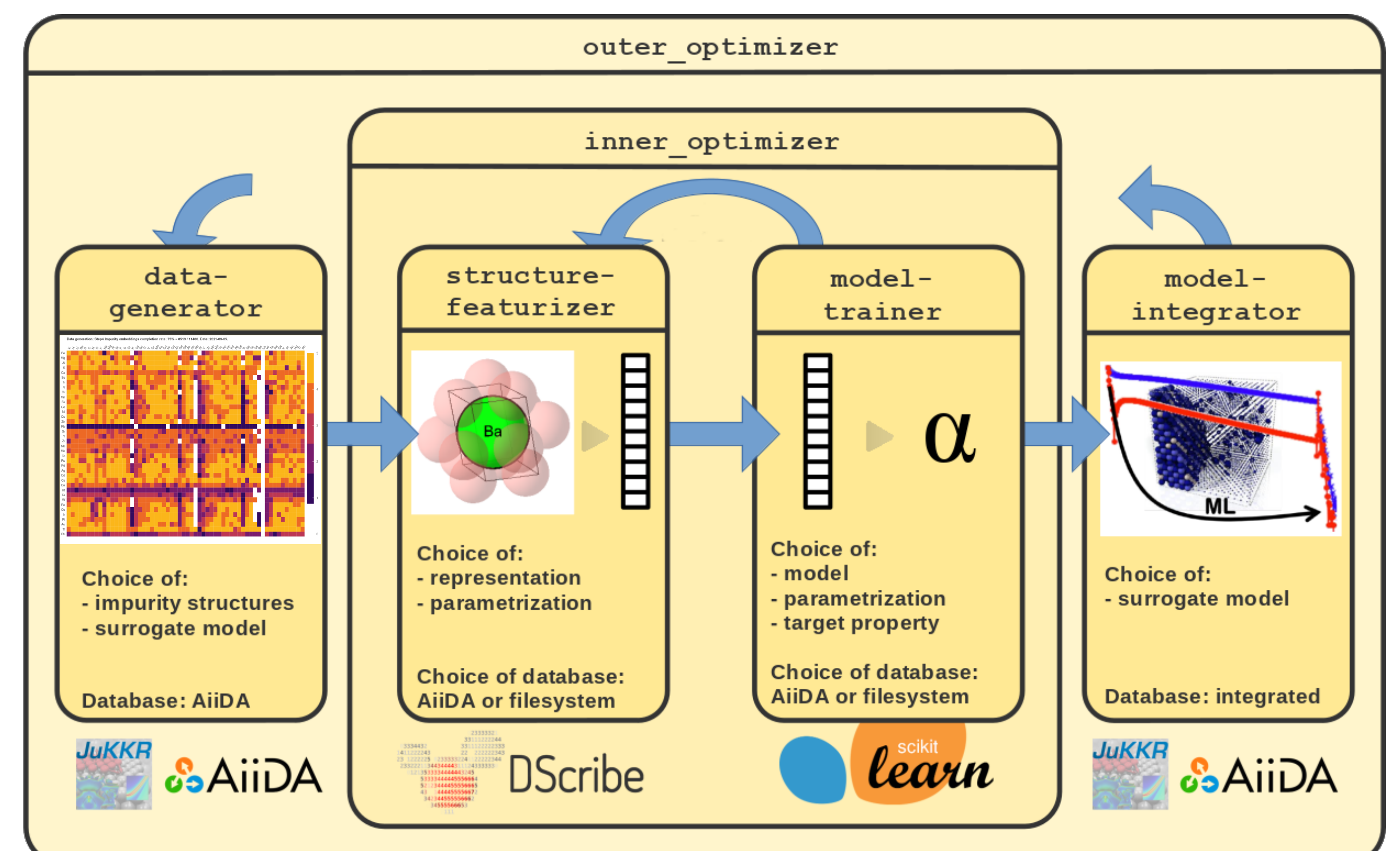
• We acknowledge support by the Joint Lab Virtual Materials Design (JL-VMD) and thank for computing time granted by the JARA Vergabegremium and provided on the JARA Partition part of the supercomputer CLAIX at RWTH Aachen University.

• This work was funded by AIDAS2 – AI, Data Analytics and Scalable Simulation – a virtual lab between CEA, France and FZJ, Germany, and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – Cluster of Excellence Matter and Light for Quantum Computing (ML4Q) EXC 2004/1 – 390534769.

References

- [1] The juKKR code package, juKKR.fz-juelich.de
- [2] P. Rüßmann, F. Bertoldo, and S. Blügel, npj Computational Materials, 7, 13 (2021)
- [3] S. P. Huber et al., Sci Data, 7, 1 (2020)
- [4] L. Himanen et al., Computer Physics Communications, 247, 106949 (2020)
- [5] F. Pedregosa et al., JMLR 12, pp. 2825-2830 (2011)

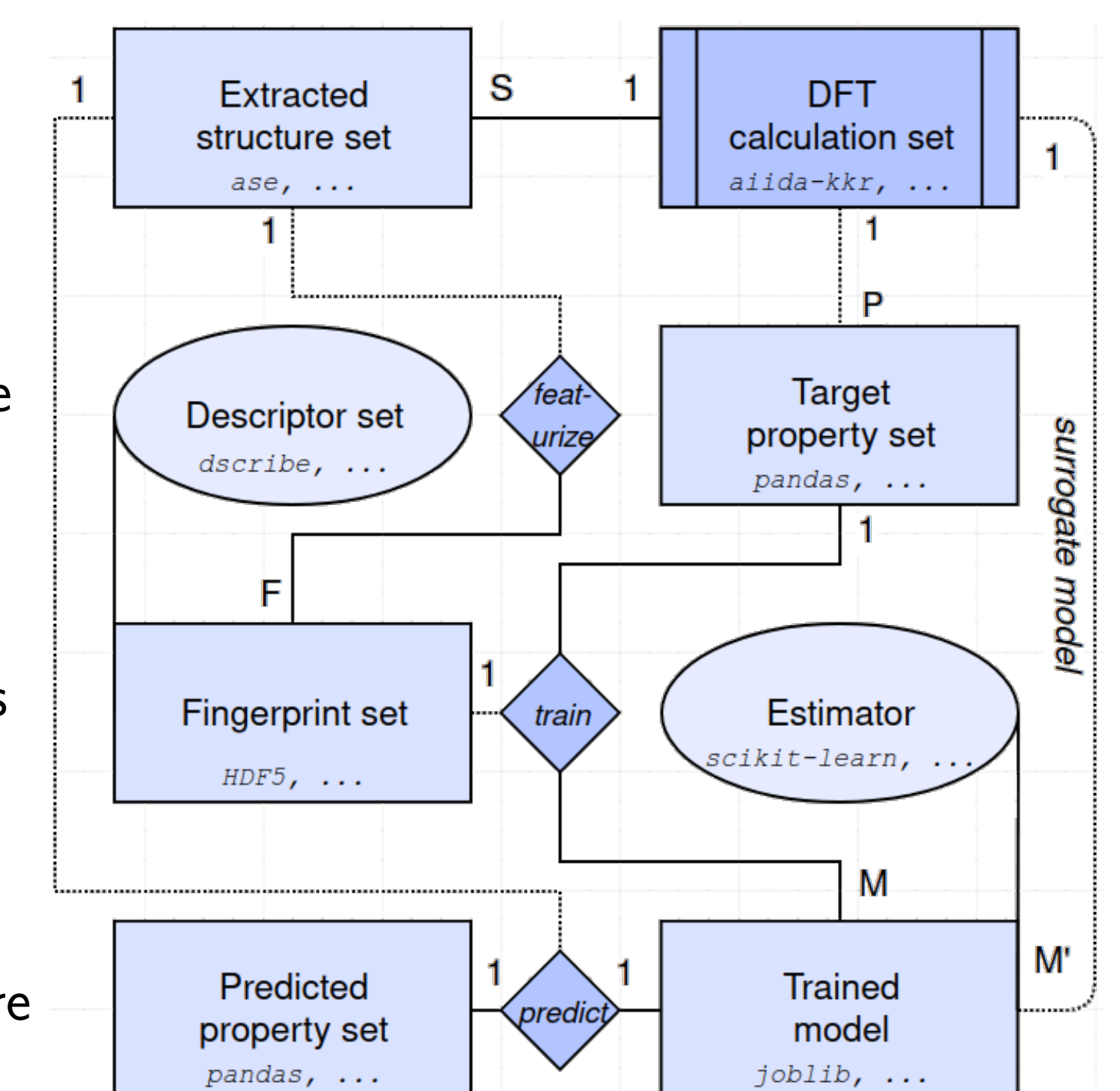
Machine learning with provenance



• We are building an atomistic machine learning workbench which will lend itself to any structural representation-based learning task. The aim is to improve reproducibility by tracking the provenance along the whole pipeline. Separate packages encapsulate each step to enable interoperability beyond aiida-kkr.

• The inner packages (featurization & model training) implement the entity relationships shown here on the filesystem as database. Features are stored in HDF5 to allow in- and out-of-memory training for scalability. The interface is decoupled from the underlying libraries for maintainability.

• We intend to leverage AiiDA as alternative database choice. This would enable fully traceable and shareable ML workflows as well as automatic compute resources integration. The main challenge here is to provide wrappers for the underlying libraries, currently Dscribe [4] and scikit-learn [5].



A proof of concept

• To demonstrate the feasibility of the initial motivation, we used the workbench for a simple classification task: the prediction of the crystal structure of the training data impurity clusters (bcc, fcc, hcp).

• As training set, we use 1752 structures. A global averaged SOAP was chosen as descriptor with a fairly low resolution $n_{\max}=l_{\max}=5$. The resulting feature vectors had a length of $2.5e5$. We used 0.75/0.25 as train/test split.

• As model we used a support vector machine from scikit-learn with linear kernel, standard scaling and default regularization. This is a quadratic optimization problem to find the separating hyperplane between the data points with the largest distance to them.

• The score (mean accuracy) was 0.93. This indicates that the SOAP parametrization indeed has encoded the crystal structure information in its fingerprints. An even better result could probably be achieved if the SOAP kernel [4] were used instead:

$$K^{\text{SOAP}}(\mathbf{p}, \mathbf{p}') = \left(\frac{\mathbf{p} \cdot \mathbf{p}'}{\sqrt{\mathbf{p} \cdot \mathbf{p} \mathbf{p}' \cdot \mathbf{p}'}} \right)^\xi$$

