Generating Minimal Training Sets for Machine Learned Potentials

Jan Finkbeiner*
Peter Grünberg Institute Forschungszentrum Jülich GmbH Wilhelm-Johnen-Straße, 52428 Jülich, Germany

Samuel Tovey* and Christian Holm†
Institute for Computational Physics University of Stuttgart Allmandring 3, 70569 Stuttgart, Germany

(Received 21 September 2022; revised 11 September 2023; accepted 19 March 2024; published 15 April 2024)

This Letter presents a novel approach for identifying uncorrelated atomic configurations from extensive datasets with a nonstandard neural network workflow known as random network distillation (RND) for training machine-learned interatomic potentials (MLPs). This method is coupled with a DFT workflow wherein initial data are generated with cheaper classical methods before only the minimal subset is passed to a more computationally expensive *ab initio* calculation. This benefits training not only by reducing the number of expensive DFT calculations required but also by providing a pathway to the use of more accurate quantum mechanical calculations. The method's efficacy is demonstrated by constructing machine-learned interatomic potentials for the molten salts KCl and NaCl. Our RND method allows accurate models to be fit on minimal datasets, as small as 32 configurations, reducing the required structures by at least 1 order of magnitude compared to alternative methods. This reduction in dataset sizes not only substantially reduces computational overhead for training data generation but also provides a more comprehensive starting point for active-learning procedures.

DOI: 10.1103/PhysRevLett.132.167301

Data-driven approaches for reconstructing potential energy surfaces have provided scientists with a unique environment for combining two thriving research areas: machine learning and molecular dynamics. These machine learning approaches aim to use data from expensive *ab initio* calculations such as density functional theory (DFT) to fit a model, which may then be used to perform molecular dynamics (MD) simulations at roughly the speed and on scales of a classical approach while retaining the accuracy of the *ab initio* computations. The last decade has seen significant advances in the use of machine learning algorithms for the development of these potentials (MLPs) [1,2], be it Gaussian process regression [3], neural networks [4–8], or other kernel methods [9–12]. A fundamental component to fitting these potentials that has recently become an active area of research is how to select data from these *ab initio* computations so that one minimizes the size of training datasets while maximally representing the underlying potential energy surface (PES). Typically, this data selection is made uniformly in time, energy, or local energies if a classical potential is used at the initial data selection stages [13–16]. In more recent studies, active learning approaches have been implemented to iteratively correct a potential as it ventures into poorly defined areas of configurations space [11,12,17]. In some cases, configurations are deliberately constructed, such as in the case of randomized atomic-system generator (RAG) sampling [18] or kernel functions applied to identify unique structures in descriptor space [19]. While we oftentimes focus predominantly on identifying relevant configurations through physical properties such as energy or forces, the problem of data selection for data-driven model training is present in all fields of machine learning and therefore, it can be instructive to look into methods adopted by the broader community. One such approach developed in reinforcement learning is random network distillation (RND) [20]. This approach has been used previously to identify unseen regions of target space for a reinforcement learner and ignore those regions the machine learning algorithm is believed to have explored [20]. However, the design of the problem closely mirrors that of selecting data for the development of machine-learned inter-atomic potentials and, therefore, is of interest to the community. RND is a method that utilizes the intrinsic bias of a neural network architecture to identify regions of the underlying data manifold that will result in a better model after training [21]. When used for data selection, the goal of RND is to take a large set of data and reduce it to a much smaller but still representative subset on which a model can be trained. The method is built upon two neural networks, the target network: $f\colon \mathcal{R}^M \to \mathcal{R}^N$ which acts as an embedding operation for data of dimension $\mathcal{M}$ to a space of dimension N, and the predictor network: $g\colon \mathcal{R}^M \to \mathcal{R}^N$ which is trained to predict the output of the target network iteratively. Before the data selection occurs, the RND mechanism must be seeded. To do so, all points in the large dataset are passed through each neural network, and a distance metric is used to compute the distance between the representations generated by f and g for each point. The

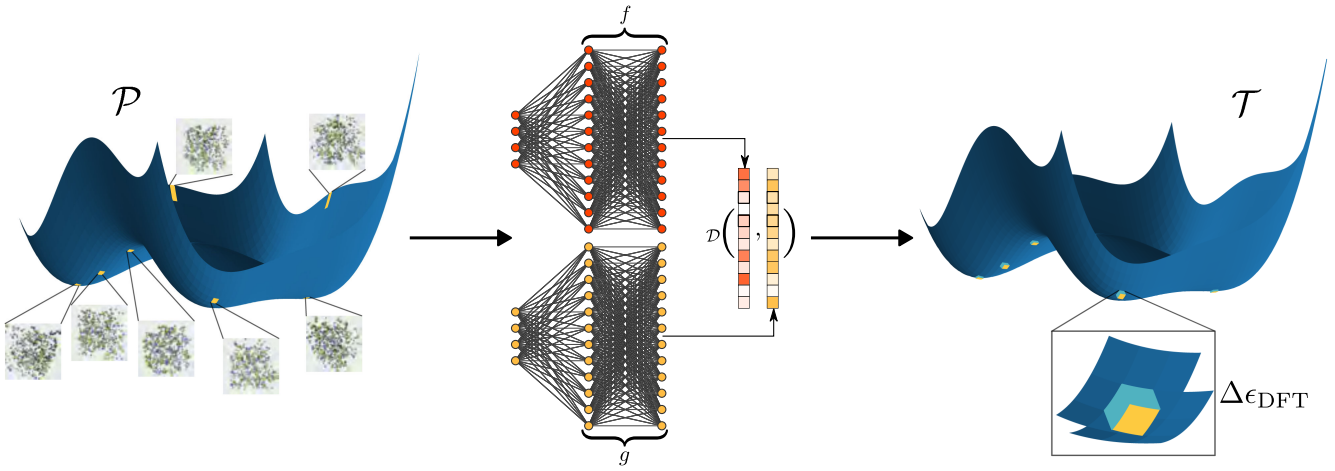

FIG. 1. Use of random network distillation to fill a training set $\mathcal{T}$. In the initial stage, classical MD simulations are used to sample configuration space and build the data pool $\mathcal{P}$ before the RND architecture is used to select unique configurations and add these to the training data. This method involves passing a representation of the configuration into the two neural networks, f, and g. These representations are then compared using a selected distance metric, d, and based on their distance, are either added to the training data, or thrown away as they have already been seen. These training data are then passed through a DFT calculation to label the configurations with a corrected energy, $\Delta\epsilon_{\mathrm{DFT}}$, and forces before training a machine-learned potential.

point with the greatest distance, p_i, is selected and added to the training set, $\mathcal{T}$. The predictor network, g, is then trained on the representation generated by $f(p_i)$. This process is continued until a dataset of a desired size has been selected.

Our Letter applies RND to selecting a representative subset of atomistic configurations on which a machine-learned potential will be trained. In building the initial data pool from which the subset is selected, large amounts of configuration space must be covered so that the chosen training set is informative. One approach is to use classical MD simulations to quickly span the configuration space at a lower accuracy. In this Letter, MD simulations are performed in systems made up of 100 atoms in a Nosè-Hoover chain [22,23] enforced NPT ensemble using the LAMMPS simulation software [24]. Interactions between the constituent atoms are defined using the Born-Meyer-Huggins-Tosi-Fumi potential [25–29] parametrized based on literature values [30] and accompanied by P^3M electrostatic corrections [31]. The simulations are run under a temperature ramp from 1100 and 1700 K to cover the liquid phase of the salts. From this data pool, RND selects representative subsets of varying sizes. For the application of RND, atomic configurations are mapped into a descriptor space using untrained SchNet graph-based representations [5,32]. These representations are then passed through the target and predictor network to perform the dataset selection. Once a subset is selected, single-point density functional theory (DFT) calculations are performed on the smaller datasets. These DFT simulations are performed with the CP2K simulation software [33], using the PBE-GGA [34] functionals, double-zeta MOLOPT basis sets optimized for dense liquids [35], GTH pseudopotentials [36], and RVV10 nonlocal integral corrections [37]. The workflow from classical MD to DFT single-point calculations is outlined in Fig. 1. While this classical to *ab initio* transfer method appears to work in the case of simple liquids, it relies on the similarity of the configuration spaces across these levels of accuracy. Therefore, it is not *a priori* valid for more complex systems, and further investigation should be performed in this direction. A benefit of RND as a data-selection method is that it scales only with N data points desired in the final dataset, as the use of two neural networks introduces, through the training procedure, memory of what has been seen before, thus avoiding the expensive nature of other descriptor-based selection methods, for example, farthest-distance approaches, which, in their vanilla implementation, scale like $\mathcal{O}(N^2)$ [19,38]. Furthermore, it separates itself from other descriptor-based methods in that it requires no training in the SchNet representation beforehand nor the existence of a well-defined distance metric on the descriptor space. Therefore, it is agnostic to the descriptor and imposes little to no bias on the problem.

After selecting the subsets, machine learning models are trained on the *ab initio* data. This Letter uses the machine learning framework SchNet [5,32]. SchNet is a graph neural network (GNN) based architecture that builds representations from atomic coordinates while respecting the symmetries inherent to the system. Models are trained on subsets of varying sizes and compared with more commonly used training data selection methods. Figure 2 outlines the results of the investigation. The figure displays both the RMSE and L4 error calculations for the force predictions of the machine learning models on previously unseen validation data as a function of dataset size for the KCl model (see Supplemental Material [39] for NaCl plots). In each plot, the color and shape of the lines

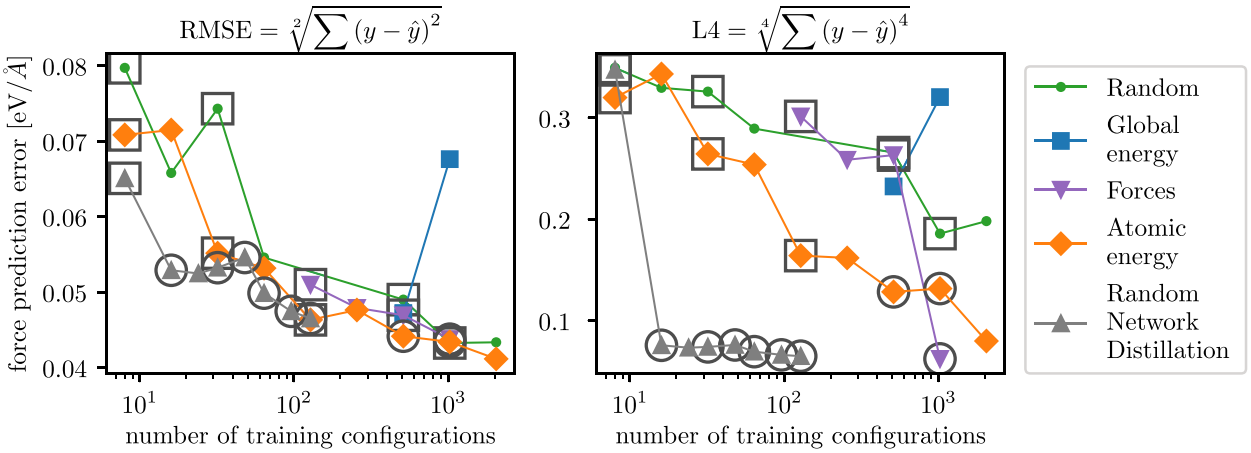

FIG. 2. The RMSE and L4 loss compared to the number of training configurations for different data selection algorithms show the convergence of the model loss. Circles correspond to those models that could be used to run a stable MD simulation, whereas a square indicates that the potential leads to a breakdown of the MD simulation after some time. This labeling measures how well the training data represent the configuration space.

correspond to a data-selection method, black circles surrounding a point symbolize that a successful MD simulation was performed using this model, and a black square shows that the simulation failed before 100 ps. Simulation failure is decided by either drift in energy and temperature, artifacts in the radial distribution function computations (e.g., peaks below atomic radii signifying atoms collapsing on top of one another, see Supplemental Material Fig. 5 [39]), or large forces experienced during the run. In the RMSE plots, while it is clear that RND generates models with lower loss values, the differences are not large compared with the other techniques. What is clear is that far more of the RND-trained models can perform MD simulations, as seen in the number of circles along the line. This trend is elucidated in the L4 error plot, where we can see that the RND-trained datasets converge much faster than all other methods to a minimum value. L4 error values have the impact of penalising outliers to a greater extent than their RMSE counterparts. The reduction in L4 error suggests that RND can identify maximally separated points, thus reducing the number of outliers in the validation data. This trend persists even when compared with other data selection techniques, which explicitly consider local atomic effects, e.g., force selection and atomic energy selection. Interestingly, the L4 error coincides with the successful running of a simulation. This relationship suggests that using loss functions that penalize outliers significantly is a good indicator of whether a potential will succeed.

With successful model fits, the trained potentials can be utilized in scaled-up MD simulations to measure all relevant properties accessible in MD. As an example, one such thermophysical observable of interest to the community is the density of a liquid at different temperatures. To reproduce the density is typically a challenging task for machine-learned potentials as it requires a good representation of configuration space in the training data, typically achieved through active learning and accurate *ab initio* data [40] with correct dispersion interactions. NPT simulations are performed using a custom-written SchNet plugin for LAMMPS [24] on scaled-up system sizes of 400 atoms. Densities are computed from 1 ns simulations at several temperatures and plotted against DFT and experimental density values in Fig. 3. The DFT values are taken from 10 ps DFT-MD simulations in an NPT ensemble with 400 atoms using the same DFT parameters as in the single-point calculations. We can see that the MLPs accurately reproduce the underlying DFT data with temperature, suggesting that the RND-selected dataset of only 32 configurations adequately mapped the configuration space of the salts.

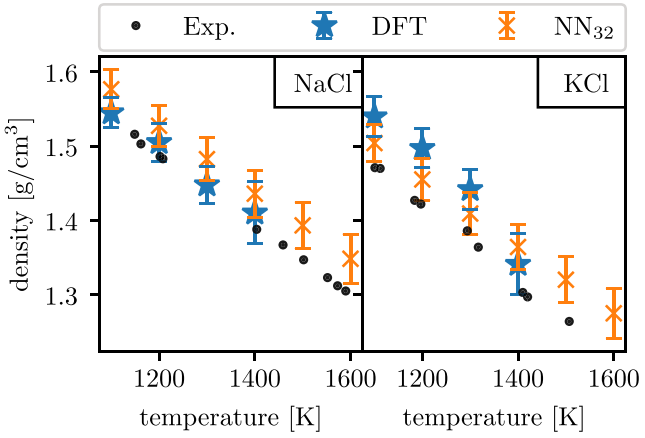

FIG. 3. Density of each salt at different temperatures computed with the machine learned potentials trained on 32 configurations (orange crosses), using pure DFT-MD (blue stars) and experimental data (black dots) taken from Ref. [41]. The experimental data are provided with 4 significant digits, making the error smaller than the symbol size in this figure.

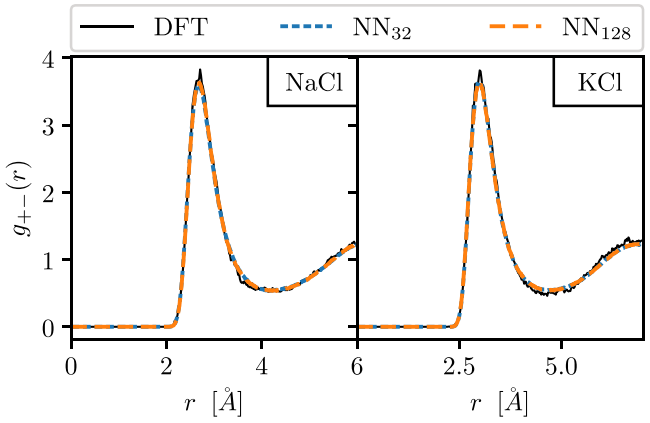

FIG. 4. Comparison of radial distribution functions generated from MD simulations performed in an NVT ensemble using the machine learned potentials trained on 32 (short dashed blue) and 128 (long dashed orange) configurations, respectively, and the underlying density functional theory data (solid black).

Another important observable in MD simulations is the radial distribution function (RDF), which can be directly related to the DFT data on which the ML model was trained. To generate data for the RDF calculations, NVT simulations are performed at densities fixed to those of the compared experimental values. The MDSuite post-processing software [42] is then used to compute the RDFs. The MD simulations are run for 1 ns using a Nosè-Hoover chain [22,23] with a coupling constant of 100 fs. To create reference data, 10 ps DFT-MD runs in an NVT ensemble are also performed using the parameters described for the single-point calculations. Figure 4 compares the anion-cation RDF curves for the machine-learned potentials against the reference DFT data. RDFs are shown for two different models trained on different amounts of data. In all cases, the ML potentials accurately reproduced the underlying DFT data.

Finally, the dynamic properties of the salts are assessed in the form of self-diffusion coefficients and ionic conductivity. The trajectories from 1 ns MD studies are used along with the MDSuite software [42] in the computation of the properties. Tables I and II compare the results computed from the MD simulations with those of the experiment.

TABLE I. Self-diffusion coefficients computed from the ML potential simulations compared with experimental fits from Ref. [43].

	Species	D_{sim}	D_{exp}
NaCl	Na	1.118 ± 0.006	1.052 ± 0.210
	Cl	0.903 ± 0.005	0.842 ± 0.168
KCl	K	1.052 ± 0.005	1.005 ± 0.201
	Cl	1.069 ± 0.006	0.905 ± 0.181

We see that for both salts, the self-diffusion coefficients match well with experimental values, suggesting an accurate MLP trained on good *ab initio* data. Ionic conductivity measurements are also in good agreement with experimental values.

We have demonstrated that random network distillation can be used to identify relevant atomic configurations to train data-driven interatomic potentials. We did so by fitting machine-learned potentials on systems of NaCl and KCl using the SchNet framework. Furthermore, our data selection method outperformed several other approaches, including global energy selection, local energy selection, and force-based selection in model convergence. We have performed molecular dynamics simulations on scaled systems of up to 500 ion pairs and for more than 1 ns to validate the ML potentials on more significant length and timescales. The structural and dynamic properties computed from these simulations were shown to reproduce pure *ab initio* investigations and experimental data adequately. Finally, we showed that RND is capable, without additional active learning, of performing stable NPT simulations and converging to the system density expected from DFT. These results support several conclusions. Random network distillation is an efficient method for identifying unique configurations for training MLPs. Single-point DFT calculations on classically generated configurations are sufficient for producing accurate training data for machine learning models. At least for chemically simple systems, the number of configurations required for an NPT-capable model yielding accurate structures, dynamics, and densities is significantly smaller than previously reported in the literature, resulting in improved training time and reduced computational demand. This minimal training set also provides an avenue for extending the potentials to higher level *ab initio* calculations such as coupled cluster [45] or configuration interaction [46] and thereby producing MLPs beyond the accuracy of DFT. Summarizing, our method shows the possibility of highly accurate simulations at a drastically reduced computational budget. This substantially expands the possibilities of simulation methods by enabling the study of systems and structures previously prohibitively expensive to compute. Future work should investigate the application

TABLE II. Ionic conductivity data from the ML potential simulations compared with experimental values taken from Ref. [44].

	σ_{Sim}	σ_{Exp}
NaCl	3.885 ± 0.118	3.954 ± 0.032
KCl	2.779 ± 0.057	2.517 ± 0.044

of RND to more complex systems and better understand its limitations.

All data is made available through either direct request to the authors or through the DarUS dataset available at [47].

The authors acknowledge financial support from the German Funding Agency (Deutsche Forschungsgemeinschaft DFG) under Germany's Excellence Strategy EXC 2075-390740016. This work was supported by SPP 2363—"Utilization and Development of Machine Learning for Molecular Applications—Molecular Machine Learning." Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), Project-No. 497249646.

*These authors contributed equally to this work.
†holm@icp.uni-stuttgart.de

[1] E. Kocer, T. W. Ko, and J. Behler, Annu. Rev. Phys. Chem. **73**, 163 (2022).
[2] J. Behler and G. Csányi, Eur. Phys. J. B **94**, 142 (2021).
[3] A. P. Bartók, M. C. Payne, R. Kondor, and G. Csányi, Phys. Rev. Lett. **104**, 136403 (2010).
[4] J. Behler and M. Parrinello, Phys. Rev. Lett. **98**, 146401 (2007).
[5] K. T. Schütt, F. Arbabzadah, S. Chmiela, K. R. Müller, and A. Tkatchenko, Nat. Commun. **8**, 13890 (2017).
[6] H. Wang, L. Zhang, J. Han, and W. E, Comput. Phys. Commun. **228**, 178 (2018).
[7] V. Zaverkin and J. Kästner, J. Chem. Theory Comput. **16**, 5410 (2020).
[8] J. Zeng *et al.*, J. Chem. Phys. **159**, 054801 (2023).
[9] R. M. Balabin and E. I. Lomakina, Phys. Chem. Chem. Phys. **13**, 11710 (2011).
[10] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, Phys. Rev. Lett. **108**, 058301 (2012).
[11] J. Vandermause, S. B. Torrisi, S. Batzner, Y. Xie, L. Sun, A. M. Kolpak, and B. Kozinsky, npj Comput. Mater. **6**, 20 (2020).
[12] J. Vandermause, Y. Xie, J. S. Lim, C. J. Owen, and B. Kozinsky, Nat. Commun. **13**, 5183 (2022).
[13] D. J. Cole, L. Mones, and G. Csányi, Faraday Discuss. **224**, 247 (2020).
[14] Y. Shao, M. Hellström, A. Yllö, J. Mindemark, K. Hermansson, J. Behler, and C. Zhang, Phys. Chem. Chem. Phys. **22**, 10426 (2020).
[15] S. Tovey, A. Narayanan Krishnamoorthy, G. Sivaraman, J. Guo, C. Benmore, A. Heuer, and C. Holm, J. Phys. Chem. C **124**, 25760 (2020).
[16] J. Finkbeiner, S. Tovey, and C. Holm, arXiv:2108.01582.
[17] G. Sivaraman, A. N. Krishnamoorthy, M. Baur, C. Holm, M. Stan, G. Csányi, C. Benmore, and Á. Vázquez-Mayagoitia, npj Comput. Mater. **6**, 104 (2020).
[18] Y.-J. Choi and S.-H. Jhi, J. Phys. Chem. B **124**, 8704 (2020).
[19] S. De, A. P. Bartók, G. Csányi, and M. Ceriotti, Phys. Chem. Chem. Phys. **18**, 13754 (2016).
[20] Y. Burda, H. Edwards, A. Storkey, and O. Klimov, arXiv: 1810.12894.
[21] S. Tovey, S. Krippendorf, K. Nikolaou, and C. Holm, Mach. Learn. **4**, 035040 (2023).
[22] W. G. Hoover, Phys. Rev. A **31**, 1695 (1985).
[23] S. Nosé, J. Chem. Phys. **81**, 511 (1984).
[24] S. Plimpton, J. Comput. Phys. **117**, 1 (1995).
[25] M. Tosi and F. Fumi, J. Phys. Chem. Solids **25**, 45 (1964).
[26] F. Fumi and M. Tosi, J. Phys. Chem. Solids **25**, 31 (1964).
[27] J. E. Mayer, J. Chem. Phys. **1**, 270 (1933).
[28] M. Born and J. E. Mayer, Z. Phys. **75**, 1 (1932).
[29] M. L. Huggins and J. E. Mayer, J. Chem. Phys. **1**, 643 (1933).
[30] G. C. Pan, J. Ding, W. Wang, J. Lu, J. Li, and X. Wei, Int. J. Heat Mass Transfer **103**, 417 (2016).
[31] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (Hilger, Bristol, 1988).
[32] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, J. Chem. Phys. **148**, 241722 (2018).
[33] T. D. Kühne *et al.*, J. Chem. Phys. **152**, 194103 (2020).
[34] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).
[35] J. VandeVondele and J. Hutter, J. Chem. Phys. **127**, 114105 (2007).
[36] S. Goedecker, M. Teter, and J. Hutter, Phys. Rev. B **54**, 1703 (1996).
[37] R. Sabatini, T. Gorni, and S. de Gironcoli, Phys. Rev. B **87**, 041108(R) (2013).
[38] R. K. Cersonsky, B. A. Helfrecht, E. A. Engel, S. Kliavinek, and M. Ceriotti, Mach. Learn. **2**, 035038 (2021).
[39] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.132.167301, which includes Ref. [16], for additional information and verification of results.
[40] J. Wang, G. Román-Pérez, J. M. Soler, E. Artacho, and M.-V. Fernández-Serra, J. Chem. Phys. **134**, 024516 (2011).
[41] A. Kirshenbaum, J. Cahill, P. McGonigal, and A. Grosse, J. Inorg. Nucl. Chem. **24**, 1287 (1962).
[42] S. Tovey, F. Zills, F. Torres-Herrador, C. Lohrmann, M. Brückner, and C. Holm, J. Cheminf. **15**, 19 (2023).
[43] G. J. Janz and N. P. Bansal, J. Phys. Chem. Ref. Data **11**, 505 (1982).
[44] G. J. Janz, in *Molten Salts Handbook*, edited by G. J. Janz (Academic Press, New York, 1967), pp. 39–51.
[45] F. Coester and H. Kümmel, Nucl. Phys. **17**, 477 (1960).
[46] C. David Sherrill and H. F. Schaefer, *The Configuration Interaction Method: Advances in Highly Correlated Approaches* (Academic Press, New York, 1999), pp. 143–269.
[47] 10.18419/darus-4099.