Additive Cook's Distance Guided Training Set Reduction for Generalizable Foundation Models of Interatomic Potentials

Ilgar Baghishov[*]
Department of Chemistry
University of Texas at Austin

Ravan Khidirov[*]
Department of Information Technology
Baku Higher Oil School

Ayan Ismayilova
Department of Computer Science
French-Azerbaijani University

Jan Janssen
Department of Computational Materials Design
Max Planck Institute for Sustainable Materials

Danny Perez[†]
Theoretical Division T-1
Los Alamos National Laboratory
danny_perez@lanl.gov

Graeme Henkelman[†]
Department of Chemistry
University of Texas at Austin
henkelman@utexas.edu

Abstract

Foundation models for machine learned interatomic potentials (MLIPs) build upon large training sets that are computationally expensive and often contain redundant information that impairs generalization. To address this, we derive Additive Cook's Distance (ACD), a novel influence measure quantifying the impact of data point addition. We use this in our stepwise ACD algorithm, an iterative method that starts with a small data subset and greedily adds the most influential configurations from the remaining pool. We validate our approach on two distinct MLIP benchmarks. For a linear qSNAP potential on a beryllium dataset with high configurational diversity, stepwise ACD achieves full-dataset accuracy using only half the data. We then apply our method to the non-linear MACE model by first linearizing it to select a representative subset from the chemically diverse Materials Project (MPTrj) dataset. A final MACE model trained only on this curated subset shows superior generalization to unseen structures, outperforming a model trained on the full dataset. This work demonstrates that stepwise ACD is a powerful strategy to reduce computational cost while enhancing the generalizability of MLIP foundation models.

1 Introduction

The performance of machine-learned interatomic potentials (MLIPs) depends on the quality and diversity of the training data [1]. This data is usually generated from expensive density functional theory (DFT) calculations on atomic configurations obtained via methods such as random sampling [2], geometry optimization [3, 4], molecular dynamics [5], design of experiments [6, 7], active learning [8–10], or their combinations [11–17]. As DFT datasets grow, researchers often use them entirely, which, perhaps counterintuitively, can be detrimental. Training on large datasets is computationally

[*]Equal contribution.
[†]Corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI4Mat.

expensive and they may contain redundant information. Such redundancy can bias a model, impairing its generalizability and accuracy, which supports findings that more data can sometimes reduce performance [18]. Therefore, intelligent data subsampling is important to reduce training costs and improve model performance by creating more effective training sets.

Early subsampling efforts for MLIPs often relied on random sampling, which, despite its simplicity, proved surprisingly effective for systems such as liquid water [19]. More structured approaches, including cluster [20] and stratified [21] sampling, partition configurations into groups before uniform selection. A significant advancement came from uncertainty-driven methods, which prioritize configurations with high prediction variance [22]. These techniques, integrated within active learning frameworks, are used not only to generate new training configurations but also to select representative subsets from existing datasets [10, 23]. Recently, random network distillation has leveraged the disagreement between fixed and trainable neural networks to identify under-sampled regions, achieving 10-fold dataset reductions for reactive systems such as molten salts [24].

Another class of methods employs advanced distance and matrix-based techniques. Farthest point sampling (FPS) maximizes coverage in the feature space by iteratively selecting configurations most distant from those already chosen. Similarly, the local-environment-guided selection of atomic structures maintains a bank of distinct local atomic environments, adding new structures only if their environments are sufficiently dissimilar from those already included in training set [25]. Matrix factorization methods such as CUR decomposition—which approximates a data matrix as the product of a small subset of its columns (C), rows (R), and a linking matrix (U)—identify structurally important configurations [26], and the associated leverage scores can be used independently as a sampling technique that we benchmark in this work. While extensions like block CUR decomposition have been proposed [27], they have not yet been applied to MLIP training set subsampling. Other approaches, such as principal covariates regression (PCovR), consider variance in both atomic environments and target properties [28].

In this work, we employ Cook's distance [29] a metric of a data point's influence measuring how model predictions change upon its *removal*. Here, we introduce a novel variant, which we term Additive Cook's Distance (ACD), to measure the influence of a data point upon its *addition* to the training set. To differentiate between metrics, the standard version will be called Subtractive Cook's Distance (SCD). ACD enables the stepwise construction of the training set from a small initial subset, analogous to active learning. We adopt this additive procedure based on its computational efficiency in contrast to a subtractive approach. Additive building of the data set avoids the large matrix inversion that would be required at the first step of stepwise SCD method. We demonstrate that this influence-guided approach can enhance MLIP foundation models. Specifically, we demonstrate that a MACE foundation model trained on a subset of the Materials Project dataset (MPTrj)—selected by our stepwise ACD method and containing only half the data—achieves superior generalizability compared to a model trained on the full dataset.

2 Methods

2.1 Subtractive Cook's Distance

Leverage identifies data points with unique features. In a linear model with a descriptor matrix X, the leverage of a point i is the diagonal element h_{ii} of the hat matrix $H = X(X^TX)^{-1}X^T$. For numerical stability, we compute H using a singular value decomposition (SVD) of $X = U\Sigma V^T$, which simplifies the hat matrix to $H = UU^T$.

While leverage identifies potentially importance, SCD quantifies configuration's actual influence on the fitted model by measuring the change in predictions upon its removal.

The Cook's distance D_i for a data point i is defined as:

$$D_i = \frac{\sum_{j=1}^{N}(\hat{y}_{j(n+i)}) - \hat{y}_{j(n)})^2}{ps^2} \tag{1}$$

where p is the number of model parameters, $\hat{y}_{j(n+i)}$ is the prediction for data point j from the full model, $\hat{y}_{j(n)}$ is the prediction when point i is excluded from the training set, and $s^2 = \frac{\mathbf{e}^{\mathrm{T}}\mathbf{e}}{n-p}$ is the mean squared error. As shown in Eq. 2, it can be expressed more efficiently using the residual e_i and

the leverage h_{ii}.

$$D_i = \frac{e_i^2}{ps^2}\left[\frac{h_{ii}}{(1-h_{ii})^2}\right] = \frac{e_i^2}{ps^2}\left[\frac{x_i(X_n^T X_n)^{-1}x_i^T}{(1-x_i(X_n^T X_n)^{-1}x_i^T)^2}\right] \tag{2}$$

We propose using Cook's distance to select the most influential configurations for training. However, the standard formulation measures the influence of a single data row (e.g., energy). Since a configuration's influence depends on its collective energy, forces, and stress data, we derived a generalized block Cook's distance. The full derivation is provided in Appendix A and the final formula is given in Eq. 3

$$D_m = \frac{1}{ps^2} e_m^T (I - H_{mm})^{-1} H_{mm} (I - H_{mm})^{-1} e_m \tag{3}$$

Where m are data rows of a configuration, $e_m = y_m - X_m \beta_{(n+m)}$ and $H_{mm} = U_m U_m^T$.

2.2 Additive Cook's Distance

As an alternative to subsampling from a large dataset, we developed an additive approach that iteratively builds a training set from a small initial subset. For this, we derived a score analogous to Cook's distance, which we term additive Cook's distance (ACD), to measure the influence of adding a candidate data point to an existing model. The stepwise process begins with a small subset of data. In each step, we calculate ACD for all remaining candidate points to identify which would be most influential. The highest-scoring configuration is then added to the training set. To make this process computationally feasible, we avoid model refitting by using a low-rank update to efficiently update the $(X^T X)^{-1}$ matrix after each addition. ACD for adding a single data point i to a model trained on n existing points X_n is given by Eq. 4:

$$D_i = \frac{e_i^2}{ps^2}\left[\frac{x_i(X_n^T X_n)^{-1}x_i^T}{1 + x_i(X_n^T X_n)^{-1}x_i^T}\right] \tag{4}$$

As with SCD, a block version of this metric can be derived to handle full atomic configuration data defined by rows m. Its derivation can be found in Appendix B with the result here,

$$D_m = \frac{1}{ps^2} e_m^T (I + X_m (X_n^T X_n)^{-1} X_m^T)^{-1} X_m (X_n^T X_n)^{-1} X_m^T e_m. \tag{5}$$

The computational cost of performing the matrix inversion update was similar to that of evaluating the ACD score for the MPTrj dataset, both requiring roughly 20–30 ms per iteration

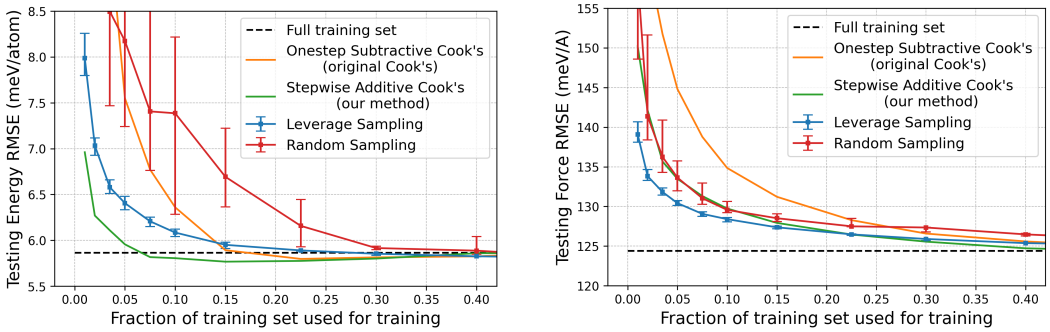
3 Results

3.1 Benchmark 1: qSNAP Fusion Applications

To validate the stepwise ACD for training set reduction, we benchmarked it on a linear model where the theory of leverage and influence scores is exact. We used the quadratic Spectral Neighbor Analysis Potential (qSNAP) [30] potential with a high DFT convergence beryllium dataset for fusion applications, consisting of 20,000 highly diverse non-equilibrium atomic configurations generated via entropy maximization [7, 31, 32]. We extracted qSNAP descriptors using FitSNAP [33] and then fit a linear regression model.

We evaluated four subsampling strategies by comparing their resulting test RMSE for energy and forces as a function of training set fraction (Figure 1). The beryllium dataset was split into a testing set of 10,000 configurations and a training set of 10,000 configuration. The training set was sub-sampled while the test set remained fixed. The benchmarked methods were: (1) random sampling, (2) sampling with probabilities proportional to leverage scores, (3) selecting points with the highest SCD, and (4) our proposed stepwise ACD method.

The results clearly demonstrate the advantage of influence-guided data reduction. Random sampling performs poorly, while leverage and standard Cook's distance offer some improvements. Our stepwise additive Cook's distance method consistently achieves the lowest energy RMSE across all dataset sizes. Notably, this method reaches the same testing accuracy as the full dataset using only half of the training set. In this calculation, all selection criteria were based on energy influence alone. We



(a) Testing Energy RMSE.

(b) Testing Force RMSE.

Figure 1: Comparison of subsampling strategies for a qSNAP Beryllium potential. Testing RMSE for (a) energy and (b) force as a function of the training set fraction used for training.

hypothesize that the "block" variant of our method (Eq. 5), which considers the combined influence of a configuration's energy and forces, would further improve force accuracy. The derivation for this block Cook's distance score is detailed in Appendix A and B for the standard data point removal and the data point addition respectively.

3.2 Benchmark 2: MACE Foundation Model for Diverse Chemistry

While Cook's distance is directly applicable to linear models like qSNAP, it is not clear that it can be applied to non-linear message-passing models. To demonstrate this, we benchmark stepwise ACD method to reduce the chemically diverse Materials Project Trajectory (MPTrj) dataset with over 1.5 million structures spanning the majority of the periodic table. The target is reducing this dataset for training the MACE foundation model while maintaining equivalent or superior generalizability compared to one trained on the full training set.

For this test, we partitioned the MPTrj data set by structure ID (MP-ID) into 70% training, 10% validation and 20% testing sets. Each set contains all snapshots from the geometry optimization of its MP-IDs, ensuring no overlap of crystal structures between sets. This ensures that the test set contains crystal structures unseen during training, providing a measure of generalization. Then, we generated subsets of this training data to be used for non-linear MACE training: 50% subset of the training set sampled randomly, 50% subset of the training set sampled according to leverage score derived probabilities, 50% subset of the training set selected via stepwise ACD method, and the full training set as a reference. We also include the 'data leakage' model which was trained with overlapping validation and test data to illustrate the effect of overfitting and artificially low test errors. In this case, we combined the validation and test sets, selected the model based on this combined set, and then evaluated its errors. We compare subsampling strategies based on testing errors of these trained MACE models in Table 1. To use the ACD formula on the MPTrj dataset, we linearized the MACE by extracting the descriptors from the pre-trained MACE-MP-0 foundation model and bypassed the final non-linear readout layers. These descriptors then served as features in a linear model for the ACD formula. As we did not consider descriptor derivatives, this linear fit is based on energies only.

Table 1 compares MACE models trained on various dataset subsets to assess how data reduction affects generalization. The two "no subsampling" cases establish baselines: the unbiased model was trained and validated in the standard way—using a validation set distinct from the test set—so the reported errors reflect genuine out-of-sample performance, while the "data-leakage" version intentionally combines the validation and test sets during training. This setup allows the model to indirectly see the test data and thus produce artificially low errors that serve as a lower-bound reference. Against the unbiased model, the stepwise additive Cook's-distance (ACD) method achieves lower RMSE (L_2) and (L_4) norms for both energy and forces and, remarkably, even surpasses the data-leakage model in (L_4) errors. Because higher-order norms amplify large deviations, these reductions demonstrate that ACD pruning effectively suppresses outlier predictions and enhances tail stability, indicating improved extrapolative behavior. The leverage-sampled subset performs comparably well, achieving the lowest force RMSE. As in benchmark 1, this can be attributed to

Table 1: MACE model testing errors obtained using different training subsets. Each model was trained on the full dataset, a 50% random subset, a 50% leverage-sampled subset, and a 50% subset selected via the stepwise additive Cook's-distance method. $\|\Delta E\|_p$ and $\|\Delta E\|_p$ denote the L_p norms of energy and force errors, respectively (e.g., $\|\Delta E\|_1$ = Energy MAE, $\|\Delta F\|_2$ = Force RMSE). The configuration labeled "data leakage" includes overlapping validation and test data and is shown for comparison. Bold values mark the two best results per column. MACE model is trained with Huber loss.

Subsampling Strategy	**Energy error (meV/atom)**			**Force error (meV/Å)**		
	$\|\Delta E\|_1$	$\|\Delta E\|_2$	$\|\Delta E\|_4$	$\|\Delta F\|_1$	$\|\Delta F\|_2$	$\|\Delta F\|_4$
Stepwise ACD	**33.46**	**68.37**	**252.18**	88.41	**188.03**	**2063.18**
Random	37.33	81.16	362.52	**67.33**	227.36	4517.35
Leverage	36.63	**75.86**	**297.61**	69.68	**170.34**	**1785.22**
No subsampling	**36.46**	100.26	1270.46	**68.52**	455.96	13065.16
No subsampling (data leakage)	26.41	65.07	360.63	59.37	170.55	2144.17

the fact that the ACD method considers leverage and energy residuals only; it does not explicitly account for force errors. Incorporating force residuals into the Cook's-distance metric, as proposed in the block variant (Eq. 5), could therefore further reduce force-related errors. Since the stepwise ACD algorithm is implemented efficiently using low-rank matrix updates: after adding each new data point, only a $(3N + 1) \times (3N + 1)$ matrix—where N is the number of atoms in the candidate configuration—needs to be inverted. Because this matrix is small, the update completes within milliseconds on a GPU. Overall, reducing the MPTrj training set by half required 9 hours on a single GPU.

Overall, both influence- and leverage-based pruning outperform random selection and can rival or even exceed full-data performance—sometimes surpassing the data-leakage lower bound—while improving computational efficiency, and generalizability to unseen configurations. By removing redundant data, the model implicitly assigns lower weight to similar configurations and greater importance to rare or outlier samples. This rebalancing enhances the model's ability to generalize, since emphasizing non-redundant and diverse regions of configuration space improves extrapolation. Consequently, we observe lower test errors—especially notable given that our test set was intentionally constructed to exclude atomic configurations seen during training and to differ by crystal structure—thus providing a meaningful measure of performance on unseen phases and structures.

4 Conclusion

In this work, we introduced stepwise additive Cook's distance method for training set data reduction for MLIPs. We highlighted its computational efficiency for two benchmarks: a linear qSNAP model for single element (beryllium) dataset of diverse atomic configurations and a non-linear MACE foundation model trained on the chemically diverse MPTrj dataset. Our results show that our method outperforms random sampling and even surpasses the performance of models trained on the full training set. Specifically, by reducing the MPTrj training set by 50%, and retraining MACE model we achieved lower energy and force RMSE on a held-out test set, indicating improved generalizability.

The success of the additive Cook's distance stems from its ability to identify and retain configurations that are not only outliers in feature space (high leverage) but also have a large impact on model predictions (high residual). This process effectively removes redundant data while preserving the most influential and diverse atomic configurations, leading to more generalizable ML foundation models of interatomic potentials.

Acknowledgments

References

[1] Bowen Deng, Yunyeong Choi, Peichen Zhong, Janosh Riebesell, Shashwat Anand, Zhuohan Li, KyuJung Jun, Kristin A. Persson, and Gerbrand Ceder. Systematic softening in universal machine learning interatomic potentials. *npj Computational Materials*, 11(1), January 2025. ISSN 2057-3960. doi: 10.1038/s41524-024-01500-6. URL http://dx.doi.org/10.1038/s41524-024-01500-6.

[2] J. S. Smith, O. Isayev, and A. E. Roitberg. Ani-1: an extensible neural network potential with dft accuracy at force field computational cost. *Chemical Science*, 8(4):3192–3203, 2017. ISSN 2041-6539. doi: 10.1039/c6sc05720a. URL http://dx.doi.org/10.1039/C6SC05720A.

[3] Bowen Deng, Peichen Zhong, KyuJung Jun, Janosh Riebesell, Kevin Han, Christopher J. Bartel, and Gerbrand Ceder. Chgnet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nature Machine Intelligence*, 5(9):1031–1041, September 2023. ISSN 2522-5839. doi: 10.1038/s42256-023-00716-3. URL http://dx.doi.org/10.1038/s42256-023-00716-3.

[4] Jonathan Schmidt, Noah Hoffmann, Hai-Chen Wang, Pedro Borlido, Pedro J. M. A. Carriço, Tiago F. T. Cerqueira, Silvana Botti, and Miguel A. L. Marques. Machine-learning-assisted determination of the global zero-temperature phase diagram of materials. *Advanced Materials*, 35(22), April 2023. ISSN 1521-4095. doi: 10.1002/adma.202210788. URL http://dx.doi.org/10.1002/adma.202210788.

[5] Shuhao Zhang, Małgorzata Z. Makoś, Ryan B. Jadrich, Elfi Kraka, Kipton Barros, Benjamin T. Nebgen, Sergei Tretiak, Olexandr Isayev, Nicholas Lubbers, Richard A. Messerly, and Justin S. Smith. Exploring the frontiers of condensed-phase chemistry with a general reactive machine learning potential. *Nature Chemistry*, 16(5):727–734, March 2024. ISSN 1755-4349. doi: 10.1038/s41557-023-01427-3. URL http://dx.doi.org/10.1038/s41557-023-01427-3.

[6] Mariia Karabin and Danny Perez. An entropy-maximization approach to automated training set generation for interatomic potentials. *The Journal of Chemical Physics*, 153(9):094110, 09 2020. ISSN 0021-9606. doi: 10.1063/5.0013059. URL https://doi.org/10.1063/5.0013059.

[7] David Montes de Oca Zapiain, Mitchell A Wood, Nicholas Lubbers, Carlos Z Pereyra, Aidan P Thompson, and Danny Perez. Training data selection for accuracy and transferability of interatomic potentials. *npj Computational Materials*, 8(1):189, 2022.

[8] Evgeny V. Podryabinkin and Alexander V. Shapeev. Active learning of linearly parametrized interatomic potentials. *Computational Materials Science*, 140:171–180, December 2017. ISSN 0927-0256. doi: 10.1016/j.commatsci.2017.08.031. URL http://dx.doi.org/10.1016/j.commatsci.2017.08.031.

[9] Kevin Tran and Zachary W. Ulissi. Active learning across intermetallics to guide discovery of electrocatalysts for co2 reduction and h2 evolution. *Nature Catalysis*, 1(9):696–703, September 2018. ISSN 2520-1158. doi: 10.1038/s41929-018-0142-1. URL http://dx.doi.org/10.1038/s41929-018-0142-1.

[10] Justin S. Smith, Ben Nebgen, Nicholas Lubbers, Olexandr Isayev, and Adrian E. Roitberg. Less is more: Sampling chemical space with active learning. *The Journal of Chemical Physics*, 148(24), May 2018. ISSN 1089-7690. doi: 10.1063/1.5023802. URL http://dx.doi.org/10.1063/1.5023802.

[11] Lowik Chanussot, Abhishek Das, Siddharth Goyal, Thibaut Lavril, Muhammed Shuaibi, Morgane Riviere, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, Aini Palizhati, Anuroop Sriram, Brandon Wood, Junwoong Yoon, Devi Parikh, C. Lawrence Zitnick, and Zachary Ulissi. Open catalyst 2020 (oc20) dataset and community challenges. *ACS Catalysis*, 11(10):6059–6072, May 2021. ISSN 2155-5435. doi: 10.1021/acscatal.0c04525. URL http://dx.doi.org/10.1021/acscatal.0c04525.

[12] Richard Tran, Janice Lan, Muhammed Shuaibi, Brandon M. Wood, Siddharth Goyal, Abhishek Das, Javier Heras-Domingo, Adeesh Kolluru, Ammar Rizvi, Nima Shoghi, Anuroop Sriram, Félix Therrien, Jehad Abed, Oleksandr Voznyy, Edward H. Sargent, Zachary Ulissi, and C. Lawrence Zitnick. The open catalyst 2022 (oc22) dataset and challenges for oxide electrocatalysts. *ACS Catalysis*, 13(5):3066–3084, February 2023. ISSN 2155-5435. doi: 10.1021/acscatal.2c05426. URL http://dx.doi.org/10.1021/acscatal.2c05426.

[13] Luis Barroso-Luque, Muhammed Shuaibi, Xiang Fu, Brandon M. Wood, Misko Dzamba, Meng Gao, Ammar Rizvi, C. Lawrence Zitnick, and Zachary W. Ulissi. Open materials 2024 (omat24) inorganic materials dataset and models, 2024. URL https://arxiv.org/abs/2410.12771.

[14] Daniel S. Levine, Muhammed Shuaibi, Evan Walter Clark Spotte-Smith, Michael G. Taylor, Muhammad R. Hasyim, Kyle Michel, Ilyes Batatia, Gábor Csányi, Misko Dzamba, Peter Eastman, Nathan C. Frey, Xiang Fu, Vahe Gharakhanyan, Aditi S. Krishnapriyan, Joshua A. Rackers, Sanjeev Raja, Ammar Rizvi, Andrew S. Rosen, Zachary Ulissi, Santiago Vargas, C. Lawrence Zitnick, Samuel M. Blau, and Brandon M. Wood. The open molecules 2025 (omol25) dataset, evaluations, and models, 2025. URL https://arxiv.org/abs/2505.08762.

[15] Vahe Gharakhanyan, Luis Barroso-Luque, Yi Yang, Muhammed Shuaibi, Kyle Michel, Daniel S. Levine, Misko Dzamba, Xiang Fu, Meng Gao, Xingyu Liu, Haoran Ni, Keian Noori, Brandon M. Wood, Matt Uyttendaele, Arman Boromand, C. Lawrence Zitnick, Noa Marom, Zachary W. Ulissi, and Anuroop Sriram. Open molecular crystals 2025 (omc25) dataset and models, 2025. URL https://arxiv.org/abs/2508.02651.

[16] Anuroop Sriram, Logan M. Brabson, Xiaohan Yu, Sihoon Choi, Kareem Abdelmaqsoud, Elias Moubarak, Pim de Haan, Sindy Löwe, Johann Brehmer, John R. Kitchin, Max Welling, C. Lawrence Zitnick, Zachary Ulissi, Andrew J. Medford, and David S. Sholl. The open dac 2025 dataset for sorbent discovery in direct air capture, 2025. URL https://arxiv.org/abs/2508.03162.

[17] Maksim Kulichenko, Benjamin Nebgen, Nicholas Lubbers, Justin S. Smith, Kipton Barros, Alice E. A. Allen, Adela Habib, Emily Shinkle, Nikita Fedik, Ying Wai Li, Richard A. Messerly, and Sergei Tretiak. Data generation for machine learning interatomic potentials and beyond. *Chemical Reviews*, 124(24): 13681–13714, November 2024. ISSN 1520-6890. doi: 10.1021/acs.chemrev.4c00572. URL http://dx.doi.org/10.1021/acs.chemrev.4c00572.

[18] Jason B Gibson, Tesia D Janicki, Ajinkya C Hire, Chris Bishop, J Matthew D Lane, and Richard G Hennig. When more data hurts: Optimizing data coverage while mitigating diversity-induced underfitting in an ultrafast machine-learned potential. *Physical Review Materials*, 9(4):043802, 2025.

[19] Nore Stolte, János Daru, Harald Forbert, Dominik Marx, and Jörg Behler. Random sampling versus active learning algorithms for machine learning potentials of quantum liquid water. *Journal of Chemical Theory and Computation*, 21(2):886–899, January 2025. ISSN 1549-9626. doi: 10.1021/acs.jctc.4c01382. URL http://dx.doi.org/10.1021/acs.jctc.4c01382.

[20] Tran Doan Huan, Rohit Batra, James Chapman, Sridevi Krishnan, Lihua Chen, and Rampi Ramprasad. A universal strategy for the creation of machine learning-based atomistic force fields. *npj Computational Materials*, 3(1), September 2017. ISSN 2057-3960. doi: 10.1038/s41524-017-0042-y. URL http://dx.doi.org/10.1038/s41524-017-0042-y.

[21] Ji Qi, Tsz Wai Ko, Brandon C. Wood, Tuan Anh Pham, and Shyue Ping Ong. Robust training of machine learning interatomic potentials with dimensionality reduction and stratified sampling. *npj Computational Materials*, 10(1), February 2024. ISSN 2057-3960. doi: 10.1038/s41524-024-01227-4. URL http://dx.doi.org/10.1038/s41524-024-01227-4.

[22] Maksim Kulichenko, Kipton Barros, Nicholas Lubbers, Ying Wai Li, Richard Messerly, Sergei Tretiak, Justin S. Smith, and Benjamin Nebgen. Uncertainty-driven dynamics for active learning of interatomic potentials. *Nature Computational Science*, 3(3):230–239, March 2023. ISSN 2662-8457. doi: 10.1038/s43588-023-00406-5. URL http://dx.doi.org/10.1038/s43588-023-00406-5.

[23] Justin S. Smith, Roman Zubatyuk, Benjamin Nebgen, Nicholas Lubbers, Kipton Barros, Adrian E. Roitberg, Olexandr Isayev, and Sergei Tretiak. The ani-1ccx and ani-1x data sets, coupled-cluster and density functional theory properties for molecules. *Scientific Data*, 7(1), May 2020. ISSN 2052-4463. doi: 10.1038/s41597-020-0473-z. URL http://dx.doi.org/10.1038/s41597-020-0473-z.

[24] Jan Finkbeiner, Samuel Tovey, and Christian Holm. Generating minimal training sets for machine learned potentials. *Physical Review Letters*, 132(16), April 2024. ISSN 1079-7114. doi: 10.1103/physrevlett.132.167301. URL http://dx.doi.org/10.1103/PhysRevLett.132.167301.

[25] L. Li, C. Zhou, A. Singh, Y. Pei, G. Henkelman, and L. Li. Local-environment-guided selection of atomic structures for the development of machine-learning potentials. *J. Chem. Phys.*, 292:074109, 2024.

[26] Giulio Imbalzano, Andrea Anelli, Daniele Giofré, Sinja Klees, Jörg Behler, and Michele Ceriotti. Automatic selection of atomic fingerprints and reference configurations for machine-learning potentials. *The Journal of Chemical Physics*, 148(24), April 2018. ISSN 1089-7690. doi: 10.1063/1.5024611. URL http://dx.doi.org/10.1063/1.5024611.

[27] Urvashi Oswal, Swayambhoo Jain, Kevin S. Xu, and Brian Eriksson. Block cur: Decomposing matrices using groups of columns, 2017. URL https://arxiv.org/abs/1703.06065.

[28] Rose K Cersonsky, Benjamin A Helfrecht, Edgar A Engel, Sergei Kliavinek, and Michele Ceriotti. Improving sample and feature selection with principal covariates regression. *Machine Learning: Science and Technology*, 2(3):035038, July 2021. ISSN 2632-2153. doi: 10.1088/2632-2153/abfe7c. URL http://dx.doi.org/10.1088/2632-2153/abfe7c.

[29] R. Dennis Cook. Influential observations in linear regression. *Journal of the American Statistical Association*, 74(365):169–174, March 1979. ISSN 1537-274X. doi: 10.1080/01621459.1979.10481634. URL http://dx.doi.org/10.1080/01621459.1979.10481634.

[30] Mitchell A. Wood and Aidan P. Thompson. Extending the accuracy of the snap interatomic potential form. *The Journal of Chemical Physics*, 148(24), March 2018. ISSN 1089-7690. doi: 10.1063/1.5017641. URL http://dx.doi.org/10.1063/1.5017641.

[31] Aparna PA Subramanyam and Danny Perez. Information-entropy-driven generation of material-agnostic datasets for machine-learning interatomic potentials. *npj Computational Materials*, 11(1):218, 2025.

[32] Ilgar Baghishov, Jan Janssen, Graeme Henkelman, and Danny Perez. Application-specific machine-learned interatomic potentials: Exploring the trade-off between precision and computational cost, 2025. URL https://arxiv.org/abs/2506.05646.

[33] A. Rohskopf, C. Sievers, N. Lubbers, M. A. Cusentino, J. Goff, J. Janssen, M. McCarthy, D. Montes de Oca Zapiain, S. Nikolov, K. Sargsyan, D. Sema, E. Sikorski, L. Williams, A. P. Thompson, and M. A. Wood. Fitsnap: Atomistic machine learning with lammps. *Journal of Open Source Software*, 8(84):5118, April 2023. ISSN 2475-9066. doi: 10.21105/joss.05118. URL http://dx.doi.org/10.21105/joss.05118.

A Derivation of Block Version for Original (Subtractive) Cook's Distance

Given that DFT data includes both energy and forces, we extended Cook's distance to consider entire configurations rather than individual data point. This new Block version of Cook's distance assesses the overall importance of a configuration, including its energy, forces, and stresses. The new definition is in 6:

$$D_m = \frac{\sum_{j=1}^{N}(\hat{y}_{j(n+m)} - \hat{y}_{j(n)})^2}{ps^2} \tag{6}$$

where m is the indices of all the rows of X corresponding to a certain configuration which can include forces and stresses in addition to energies and n is the indices of the rest of the rows. Therefore $\hat{y}_{j(n+m)}$ is the prediction on data point j of the linear regression trained on all data points and $\hat{y}_{j(n)}$ is the prediction on data point j of the linear regression trained on all data points except data points m that correspond to a certain configuration.

Using the definition of $\hat{y}$:

$$\begin{aligned} D_m &= \frac{\sum_{j=1}^{N}(X_j(\beta_{(n+m)} - \beta_{(n)}))^2}{ps^2} \\ &= \frac{1}{ps^2}(\beta_{(n+m)} - \beta_{(n)})^T X_{n+m}^T X_{n+m}(\beta_{(n+m)} - \beta_{(n)}) \end{aligned} \tag{7}$$

Instead of determining $\beta_{(n)}$ through retraining of ML every time we remove each configuration m to determine the importance of that configuration, we would like to train once on all the data and obtain $\beta_{(n+m)}$ and then have a quick way to get $\beta_{(n)}$. To start let's write what is the equation for $\beta_{(n)}$:

$$\begin{aligned} \beta_{(n)} &= (X_n^T X_n)^{-1} X_n^T y_n \\ &= (X_{n+m}^T X_{n+m} - X_m^T X_m)^{-1}(X_{n+m}^T y_{n+m} - X_m^T y_m) \end{aligned} \tag{8}$$

The Woodbury matrix identity:

$$(A - UCV)^{-1} = A^{-1} + A^{-1}U(C^{-1} - VA^{-1}U)^{-1}VA^{-1} \tag{9}$$

Applying Eq. 9:

$$\begin{aligned} (X_{n+m}^T X_{n+m} - X_m^T X_m)^{-1} &= (X_{n+m}^T X_{n+m})^{-1} \\ &+ (X_{n+m}^T X_{n+m})^{-1} X_m^T (I - X_m (X_{n+m}^T X_{n+m})^{-1} X_m^T)^{-1} X_m (X_{n+m}^T X_{n+m})^{-1} \end{aligned} \tag{10}$$

Plugging Eq. 10 into Eq. 8:

$$\begin{aligned} \beta_{(n)} &= (X_{n+m}^T X_{n+m})^{-1}(X_{n+m}^T y_{n+m} - X_m^T y_m) \\ &+ (X_{n+m}^T X_{n+m})^{-1} X_m^T (I - H_{mm})^{-1} X_m (X_{n+m}^T X_{n+m})^{-1}(X_{n+m}^T y_{n+m} - X_m^T y_m) \end{aligned} \tag{11}$$

Where $H_{mm} = X_m(X_{n+m}^T X_{n+m})^{-1} X_m^T$ which is a block matrix of the whole hat matrix $H = X_{n+m}(X_{n+m}^T X_{n+m})^{-1} X_{n+m}^T$ indexed by rows m and columns m

$$\begin{aligned}\beta_{(n)} = \beta_{(n+m)} - (X_{n+m}^T X_{n+m})^{-1} X_m^T y_m \\ + (X_{n+m}^T X_{n+m})^{-1} X_m^T (I - H_{mm})^{-1} (X_m \beta_{(n+m)} - H_{mm} y_m)\end{aligned} \tag{12}$$

Combining like terms and rearranging:

$$\begin{aligned}\beta_{(n+m)} - \beta_{(n)} &= (X_{n+m}^T X_{n+m})^{-1} X_m^T (I - H_{mm})^{-1} ((I - H_{mm}) y_m - X_m \beta_{(n+m)} + H_{mm} y_m) \\ &= (X_{n+m}^T X_{n+m})^{-1} X_m^T (I - H_{mm})^{-1} (y_m - X_m \beta_{(n+m)}) \\ &= (X_{n+m}^T X_{n+m})^{-1} X_m^T (I - H_{mm})^{-1} e_m\end{aligned} \tag{13}$$

Where $e_m = (y_m - X_m \beta_{(n+m)})$.

Eq. 13 gives us a formula of difference between coefficients of linear regression models of the case when we use all data points and when we remove indices indicated by m.

$$X_{n+m}(\beta_{n+m} - \beta_n) = X_{n+m}(X_{n+m}^T X_{n+m})^{-1} X_m^T (I - H_{mm})^{-1} e_m \tag{14}$$

Multiplying by it's transpose gives:

$$\begin{aligned}&(\beta_{n+m} - \beta_n)^T X_{n+m}^T X_{n+m} (\beta_{n+m} - \beta_n) = \\ &= e_m^T ((I - H_{mm})^{-1})^T X_m ((X_{n+m}^T X_{n+m})^{-1})^T X_{n+m}^T X_{n+m} (X_{n+m}^T X_{n+m})^{-1} X_m^T (I - H_{mm})^{-1} e_m \\ &= e_m^T ((I - H_{mm})^{-1})^T X_m ((X_{n+m}^T X_{n+m})^{-1})^T X_m^T (I - H_{mm})^{-1} e_m \\ &= e_m^T ((I - H_{mm})^{-1})^T (X_m (X_{n+m}^T X_{n+m})^{-1} X_m^T)^T (I - H_{mm})^{-1} e_m \\ &= e_m^T ((I - H_{mm})^{-1})^T H_{mm} (I - H_{mm})^{-1} e_m \\ &= e_m^T (I - H_{mm})^{-1} H_{mm} (I - H_{mm})^{-1} e_m\end{aligned} \tag{15}$$

Substituting (Eq. 15) into (Eq. 7) gives the final equation for generalized Cook's distance:

$$D_m = \frac{1}{ps^2} e_m^T (I - H_{mm})^{-1} H_{mm} (I - H_{mm})^{-1} e_m \tag{16}$$

Where $e_m = y_m - X_m \beta_{(n+m)}$ and H_{mm} can be computed as $H_{mm} = U_m U_m^T$.

B Derivation of Additive Cook's Distance for both Block and Regular versions

We derived a formula similar to Cook's distance, which we call additive Cook's distance in this paper, to evaluate the impact of adding a new data point to the dataset instead of removing it. Starting with a small, leverage-sampled or even random subset, we used this additive Cook's distance to iteratively select and add the most influential data points. To avoid expensive refitting upon the addition of a new configuration to our training set, we applied a low-rank update to the matrix $(X^T X)^{-1}$ after each addition. This allowed us to quickly recompute the influence of the rest of data points and continue the selection process.

The definition for the additive Cook's is the same as in Eq. 1. The difference is in the base model which in the case of normal Cook's distance is the $\hat{y}_{j(n+m)}$ model with data points m and in the additive Cook's is the $\hat{y}_{j(n)}$ model without data points m. To derive the expression let's first substitute the definition of $\hat{y}$ into Eq. 1:

$$\begin{aligned}D_m &= \frac{\sum_{j=1}^N (X_j(\beta_{(n+m)} - \beta_{(n)}))^2}{ps^2} \\ &= \frac{1}{ps^2} (\beta_{(n+m)} - \beta_{(n)})^T X_{n+m}^T X_{n+m} (\beta_{(n+m)} - \beta_{(n)})\end{aligned} \tag{17}$$

$$\beta_{(n+m)} = (X_{n+m}^T X_{n+m})^{-1} X_{n+m}^T y_{n+m} = (X_n^T X_n + X_m^T X_m)^{-1} (X_n^T y_n + X_m^T y_m) \tag{18}$$

Applying Woodbury matrix identity:

$$\begin{aligned}(X_n^T X_n + X_m^T X_m)^{-1} = (X_n^T X_n)^{-1} \\ - (X_n^T X_n)^{-1} X_m^T (I + X_m (X_n^T X_n)^{-1} X_m^T)^{-1} X_m (X_n^T X_n)^{-1}\end{aligned} \tag{19}$$

Plugging Eq. 19 back into Eq. 18:

$$\begin{aligned}\beta_{(n+m)} &= (X_n^T X_n)^{-1}(X_n^T y_n + X_m^T y_m) \\ &- (X_n^T X_n)^{-1} X_m^T (I + X_m (X_n^T X_n)^{-1} X_m^T)^{-1} X_m (X_n^T X_n)^{-1} (X_n^T y_n + X_m^T y_m) \\ &= \beta_{(n)} + (X_n^T X_n)^{-1} X_m^T y_m \\ &- (X_n^T X_n)^{-1} X_m^T (I + X_m (X_n^T X_n)^{-1} X_m^T)^{-1} (X_m \beta_{(n)} + X_m (X_n^T X_n)^{-1} X_m^T y_m)\end{aligned} \tag{20}$$

After rearranging we get:

$$\begin{aligned}\beta_{(n+m)} - \beta_{(n)} &= (X_n^T X_n)^{-1} X_m^T (I + X_m (X_n^T X_n)^{-1} X_m^T)^{-1} \\ &\quad ((I + X_m (X_n^T X_n)^{-1} X_m^T) y_m - X_m \beta_{(n)} - X_m (X_n^T X_n)^{-1} X_m^T y_m) \\ &= (X_n^T X_n)^{-1} X_m^T (I + X_m (X_n^T X_n)^{-1} X_m^T)^{-1} (y_m - X_m \beta_{(n)}) \\ &= (X_n^T X_n)^{-1} X_m^T (I + X_m (X_n^T X_n)^{-1} X_m^T)^{-1} e_m\end{aligned} \tag{21}$$

Where $e_m = (y_m - X_m \beta_{(n)})$.

$$X_{n+m}(\beta_{n+m} - \beta_n) = X_{n+m}(X_n^T X_n)^{-1} X_m^T (I + X_m (X_n^T X_n)^{-1} X_m^T)^{-1} e_m \tag{22}$$

$$\begin{aligned}&(\beta_{n+m} - \beta_n)^T X_{n+m}^T X_{n+m} (\beta_{n+m} - \beta_n) = \\ &= e_m^T ((I + H_{mm})^{-1})^T X_m ((X_n^T X_n)^{-1})^T X_{n+m}^T X_{n+m} (X_n^T X_n)^{-1} X_m^T (I + H_{mm})^{-1} e_m \\ &= e_m^T (I + H_{mm})^{-1} X_m (X_n^T X_n)^{-1} X_n^T X_n (X_n^T X_n)^{-1} X_m^T (I + H_{mm})^{-1} e_m \\ &+ e_m^T (I + H_{mm})^{-1} X_m (X_n^T X_n)^{-1} X_m^T X_m (X_n^T X_n)^{-1} X_m^T (I + H_{mm})^{-1} e_m \\ &= e_m^T (I + H_{mm})^{-1} X_m (X_n^T X_n)^{-1} X_m^T (I + H_{mm})^{-1} e_m \\ &+ e_m^T (I + H_{mm})^{-1} X_m (X_n^T X_n)^{-1} X_m^T X_m (X_n^T X_n)^{-1} X_m^T (I + H_{mm})^{-1} e_m \\ &= e_m^T (I + H_{mm})^{-1} H_{mm} (I + H_{mm}) (I + H_{mm})^{-1} e_m \\ &= e_m^T (I + H_{mm})^{-1} H_{mm} e_m\end{aligned} \tag{23}$$

Where $H_{mm} = X_m ((X_n^T X_n)^{-1})^T X_m^T$.

Substituting (Eq. 23) into (Eq. 17) we get the final Additive Block Cook's distance formula:

$$D_m = \frac{1}{ps^2} e_m^T (I + X_m (X_n^T X_n)^{-1} X_m^T)^{-1} X_m (X_n^T X_n)^{-1} X_m^T e_m \tag{24}$$

Additive Cook's distance for adding only one data point i:

$$D_i = \frac{e_i^2}{ps^2} \left[\frac{x_i (X_n^T X_n)^{-1} x_i^T}{1 + x_i (X_n^T X_n)^{-1} x_i^T} \right] \tag{25}$$