Fine-tuning Foundation Models for Molecular Dynamics: A Data-Efficient Approach with Random Features

Pietro Novelli
Istituto Italiano di Tecnologia

Luigi Bonati
Istituto Italiano di Tecnologia

Pedro J. Buigues
Istituto Italiano di Tecnologia

Giacomo Meanti
University of Grenoble Alpes

Lorenzo Rosasco
MaLGa-DIBRIS, University of Genoa
CBMM - MIT
Istituto Italiano di Tecnologia

Michele Parrinello
Istituto Italiano di Tecnologia

Massimiliano Pontil
Istituto Italiano di Tecnologia

Abstract

Accurate modeling of atomistic interactions using machine learning potentials has become an essential tool for molecular dynamics simulations. However, training these models typically requires large amounts of expensive *ab initio* data, such as those generated by density functional theory. Recently, foundation models trained on large and diverse datasets have shown promise because of their good performance, even on out-of-distribution systems. Despite this progress, they are still far from optimal and often require further fine-tuning. Doing so, especially in a data-efficient and computationally feasible way, remains a key challenge. In response, we present *franken*, which combines a representation extracted from graph neural networks with random features models. Through experiments on systems from the TM23 transition metals dataset, we show that *franken* provides accurate and robust molecular dynamics simulations with minimal sample complexity, providing an efficient path to high-quality results.

1 Introduction

Machine-learning (ML) interatomic potentials[1] have become crucial tools for molecular dynamics[2] (MD) simulations in a wide range of systems, managing to provide similar accuracy to *ab initio* calculations at a tiny fraction of the cost. They work by fitting the potential energy surface (PES) as a function of atomic coordinates to a set of reference quantum-mechanical calculations, typically performed within the framework of density functional theory (DFT). Many ML architectures have been designed around the properties of the PES, such as the invariance under roto-translation and permutation of atoms of the same chemical species. To meet these requirements, one can either use physically-motivated invariant descriptors together with feed-forward neural networks [3]/kernel methods [4] or graph neural networks (GNNs) such as [5, 6]. The latter encodes symmetries directly within the architecture [7], implicitly learning an effective representation of the system.

One limitation of ML potentials is that they are typically system-specific, requiring large, high-quality datasets containing all relevant configurations, which can be prohibitively expensive to generate. The recent emergence of atomistic foundation models [8, 9], trained on large and diverse datasets such as OpenCatalyst [8] and MaterialsProject [10], introduced a paradigm shift. These models exhibit a

Machine Learning and the Physical Sciences Workshop, NeurIPS 2024.

remarkable degree of generalization, showing robustness even when applied to systems that differ from their training data. Yet, despite their broad applicability, foundation models often fail to deliver the accuracy required on specific systems. Hence, an additional round of fine-tuning [11] is made necessary.

Building upon [12, 13], we propose *franken*, a fine-tuning method that addresses these challenges by combining the generalization power of foundation models with the computational and data efficiency of random features (RF) models. Leveraging the well-understood theoretical properties of RF models [14, 15], the global optimum solution for *franken* can be computed in closed form, resulting in an extremely fast optimization. Our method provides a lightweight and theoretically grounded solution for enhancing foundation models, with promising outcomes in challenging MD scenarios.

2 The Proposed Method: Franken

Our method aims to obtain a model for the potential energy of atomistic systems by fine-tuning a foundation model. Its mechanism can be decomposed into three steps:

$$\boldsymbol{R} \xmapsto{\text{(1) GNN}} h(\boldsymbol{R}) \xmapsto{\text{(2) RF}} \phi(h(\boldsymbol{R})) \xmapsto[\text{readout}]{\text{(3) energy}} \langle \boldsymbol{w}, \phi(h(\boldsymbol{R})) \rangle \,.$$

(1) Extracting feature maps from GNN. The first step of *franken*'s pipeline is representing the chemical environment of each atom in a configuration $\boldsymbol{R} \in \mathbb{R}^{3N}$ with a vector $h(\boldsymbol{R}) \in \mathbb{R}^d$ of SO(3)-invariant features extracted from the inner layers of a pre-trained atomistic foundation model [9, 16]. Although we focus on invariant descriptors, in cases where the GNN backbone employs equivariant message-passing schemes, the invariant features of the inner layers are determined by the underlying equivariant information from previous ones. This enables *franken* to leverage equivariant properties indirectly while maintaining the computational simplicity of invariant features.

In the experiments reported below, we employed the MACE-MP0 foundation model, based on the MACE architecture [9] and optimized on the Materials Project [10] database. Inspired by the readout function of the MACE architecture, the descriptors h are obtained by concatenating the invariant node features $\boldsymbol{v}^{(l)}$ at different interaction steps l up to L into a single vector: $h(\boldsymbol{R}) = (\boldsymbol{v}^{(0)}, \boldsymbol{v}^{(1)}, ..., \boldsymbol{v}^{(L)})$.

(2) Random features models. The GNN descriptors $h(\boldsymbol{R}) \in \mathbb{R}^d$ are then transformed with an additional non-linearity via Random Features (RF) maps [17, 18, 19]. These are non-linear functions $\phi : \mathbb{R}^d \to \mathbb{R}^D$ that, as the output dimension D grows, asymptotically approximate a given kernel function [20]. Therefore, RFs can be seen as large-scale alternatives to exact kernel methods attaining similar learning guarantees [14, 15]. In particular, we can use RFs to approximate the popular Gaussian kernel using the following map [17, 19]

$$\phi(h(\boldsymbol{R})) := \sin\left(\boldsymbol{W} \cdot h(\boldsymbol{R}) + \boldsymbol{b}\right),$$

where $\boldsymbol{W} \in \mathbb{R}^{D \times d}$ has rows sampled independently from a standard multivariate normal, and $\boldsymbol{b} \in \mathbb{R}^D$ with entries sampled i.i.d. uniformly in the range $[0, 2\pi)$. Notice how the parameters $\boldsymbol{W}$ and $\boldsymbol{b}$ need not be learned, just sampled once. Similar strategies can also be adapted for polynomial kernels [18] or softmax kernels [21].

(3) Readout: energy and forces predictions. In the last step, we model the atomic energy for the n-th atom as the scalar product between the *fixed* GNN+RF descriptors $\phi_n(\boldsymbol{R}) := \phi(h_n(\boldsymbol{R}))$ and a *learnable* vector of coefficients $\boldsymbol{w} \in \mathbb{R}^D$

$$\epsilon_n(\boldsymbol{R}; \boldsymbol{w}) := \frac{1}{N} \langle \phi_n(\boldsymbol{R}), \boldsymbol{w} \rangle \,.$$

The total energy is then obtained via a pooling operation $E(\boldsymbol{R}; \boldsymbol{w}) = N \sum_{n=1}^{N} \epsilon_n(\boldsymbol{R}; \boldsymbol{w})$, while the forces are calculated as the gradient of the total energy

$$\boldsymbol{F}(\boldsymbol{R}; \boldsymbol{w}) = -\nabla_{\boldsymbol{R}} E(\boldsymbol{R}; \boldsymbol{w}) = -\boldsymbol{w}^\top \sum_{n=1}^{N} \nabla_{\boldsymbol{R}} \phi_n(\boldsymbol{R}),$$

where $\nabla_{\boldsymbol{R}} \phi_n(\boldsymbol{R}) \in \mathbb{R}^{D \times 3N}$ is the Jacobian of ϕ with respect to $\boldsymbol{R}$ which can be efficiently computed using automatic differentiation.

Franken optimization. To train *franken* over a dataset $\mathcal{D} := (\boldsymbol{R}_t; E_t, \boldsymbol{F}_t)_{t=1}^T$, we only need to optimize the vector of coefficients $\boldsymbol{w}$. In practice, we minimize a convex combination of least squares loss functions for energy and forces

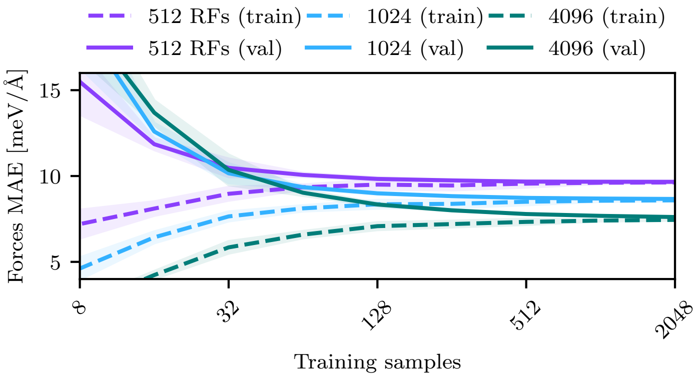
$$\ell_\alpha(\boldsymbol{w}) := (1-\alpha)\ell_{\mathrm{E}}(\boldsymbol{w}) + \alpha\, \ell_{\mathrm{F}}(\boldsymbol{w}), \tag{1}$$

$$\ell_{\mathrm{E}}(\boldsymbol{w}) := \sum_{t=1}^{T} \left(E(\boldsymbol{R}_t;\boldsymbol{w}) - E_t\right)^2 \qquad \ell_{\mathrm{F}}(\boldsymbol{w}) := \sum_{t=1}^{T} \left\|\boldsymbol{F}(\boldsymbol{R}_t;\boldsymbol{w}) - \boldsymbol{F}_t\right\|^2 .$$

Working out the gradient of the loss functions, as shown in Appendix A, allows us to compute the global minimum of Eq. (1) analytically. This is possible due to the convexity properties of RF models. By solving the closed-form expression for the optimal parameters, *franken* achieves efficient fine-tuning without the need for iterative gradient descent, further reducing computational costs.

3 Experiments

Since our main focus is performing MD simulations, we need ML potentials that, in addition to accurately predicting energy and forces, generate reliable trajectories that adhere to the underlying physics over time [22]. To this aim, we fine-tuned the MACE-MP0 foundation model [9] on copper (Cu) data from the TM23 dataset [23]. This dataset contains configurations extracted from DFT-based MD simulations at three different temperatures: *Cold* = $0.25\ T_{\mathrm{m}}$, *Warm* = $0.75\ T_{\mathrm{m}}$, and *Melt* = $1.25\ T_{\mathrm{m}}$, where $T_{\mathrm{m}} = 1358$ K is the melting point. To assess the data-efficiency, we trained *franken* models on 5 random subsamples of the training dataset (2700 configurations), with sizes ranging from 8 up to 2048 samples. For each model, the accuracy of force predictions is evaluated on the validation dataset (300 configurations). Furthermore, to assess the quality of the MD generated by *franken*, we performed a 50 ps-long simulation for each of the three temperatures considered above. Specifically, we compared the resulting radial distribution function [2] (RDF) with the reference ones provided in [23].



Train time on `4xA100 GPUs`

# Samples	Train time [s]
8	5.8 ± 0.4
32	8.4 ± 0.5
128	16.6 ± 0.9
512	46.4 ± 0.9
2048	163.8 ± 0.6

Figure 1: **Sample complexity, forces prediction.** Training (dashed lines) and validation (solid lines) mean absolute errors corresponding to different numbers of RFs as a function of the training samples.

Table 1: Forces accuracy

Model	Forces MAE
MACE-MP0 [9] (zero-shot)	93.15 meV/Å
FLARE [24] (from scratch)	8.82 meV/Å [23]
franken (fine-tuning)	**7.61 meV/Å**

Accuracy: The forces' mean absolute error (MAE) reaches values ≤ 10 meV/Å already with few tens of training points, see Fig. 1. As the model complexity (that is, the number D of RFs) increases, the model performance improves accordingly. To put these results in perspective, in Table 1, we compare *franken* against MACE-MP0 zero-shot, as well as the kernel-based FLARE method [24] reported in [23].

Data efficiency: In Fig. 1, we plot training and validation errors as a function of the number of training configurations.

The difference between training and validation errors is a proxy of the generalization capability of the model, with large values typically associated to overfitting. For *franken* the gap between train and validation errors is rapidly depleted as the number of training samples is increased. For example, with 1024 random features, just 128 samples suffice to have a validation error of $9\ \mathrm{meV/Å}$, only $0.64\ \mathrm{meV/Å}$ away from the training error.

Even more strikingly, in Fig. 2 we show the Radial Distribution Function (RDF) resulting from MD simulations with *franken* potentials. Already at 32 training samples, *franken* potentials can generate dynamics whose resulting RDF is virtually identical to the DFT reference (dotted gray) for every temperature considered. This number can be further lowered to 8 samples for the *cold* and *warm* regimes, which are closer to the pre-training distribution of the backbone foundation model [9].

Speed: The embarrassingly parallel training algorithm of *franken* allowed us to fine-tune the foundation model [9] in mere seconds, with a training time growing linearly with the number of samples (see the Table in Fig. 1). The inference performance, measured in terms of frames per second (FPS) on a single A100 GPU, is 62 FPS, which is about 15% faster than the original foundation model (54 FPS). This is explained by the fact that *franken* stops the forward pass through the GNN as soon as it collects the needed descriptors, ignoring, e.g., readout layers. Both training and inference times have a negligible dependence on the number of random features throughout the range 128-4096 that we have tested. This makes the case for *franken* being a powerful yet lightweight fine-tuning scheme.

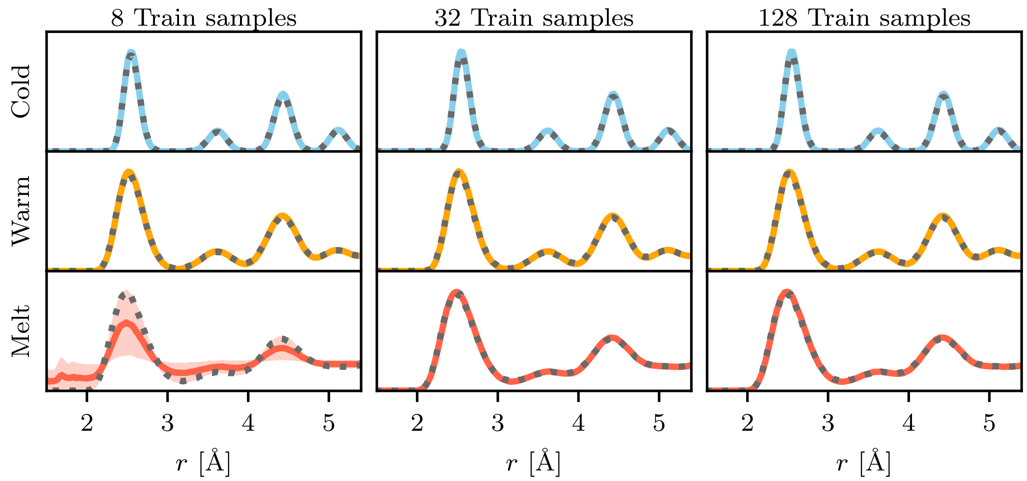


Figure 2: **Sample complexity, MD simulations**. Radial distribution functions generated from MD simulations with *franken* potentials at different temperatures (rows) and with 4096 RFs. Different columns correspond to different numbers of training samples. Each panel shows the mean (solid line) and standard deviation (shaded area) over 5 models trained on independent sub-splits, together with the reference calculated from the TM23 dataset (dotted line).

4 Conclusion

We presented a preview of *franken*, a method that combines the robustness of atomistic foundation models with the efficiency of RF models to train accurate ML potentials in a matter of seconds on a single GPU. Preliminary experiments on copper systems show *franken*'s accuracy, data efficiency, and speed. Crucially, RF models can be easily implemented in deep learning libraries as neural network modules, enabling seamless integration with existing foundation models. Further ongoing studies are comparing the performance of *franken* with other baselines, as well as the behavior when changing the GNN backbone and/or RF model, evaluated on a range of different metrics, from mean absolute errors to dynamic stability [22].

References

[1] Oliver T Unke, Stefan Chmiela, Huziel E Sauceda, Michael Gastegger, Igor Poltavsky, Kristof T Schuutt, Alexandre Tkatchenko, and Klaus-Robert Muller. Machine learning force fields. *Chemical Reviews*, 121(16):10142–10186, 2021.

[2] Daan Frenkel and Berend Smit. *Understanding molecular simulation: from algorithms to applications*. Elsevier, 2023.

[3] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98(14):146401, 2007.

[4] Albert P Bartók, Mike C Payne, Risi Kondor, and Gábor Csányi. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Physical review letters*, 104(13):136403, 2010.

[5] Ilyes Batatia, David P Kovacs, Gregor Simm, Christoph Ortner, and Gábor Csányi. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in Neural Information Processing Systems*, 35:11423–11436, 2022.

[6] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):2453, 2022.

[7] Ilyes Batatia, Simon Batzner, Dávid Péter Kovács, Albert Musaelian, Gregor NC Simm, Ralf Drautz, Christoph Ortner, Boris Kozinsky, and Gábor Csányi. The design space of e (3)-equivariant atom-centered interatomic potentials. *arXiv preprint arXiv:2205.06643*, 2022.

[8] Lowik Chanussot, Abhishek Das, Siddharth Goyal, Thibaut Lavril, Muhammed Shuaibi, Morgane Riviere, Kevin Tran, Javier Heras-Domingo, Caleb Ho, Weihua Hu, et al. Open catalyst 2020 (oc20) dataset and community challenges. *Acs Catalysis*, 11(10):6059–6072, 2021.

[9] Ilyes Batatia, Philipp Benner, Yuan Chiang, Alin M Elena, Dávid P Kovács, Janosh Riebesell, Xavier R Advincula, Mark Asta, William J Baldwin, Noam Bernstein, et al. A foundation model for atomistic materials chemistry. *arXiv preprint arXiv:2401.00096*, 2023.

[10] Anubhav Jain, Shyue Ping Ong, Geoffroy Hautier, Wei Chen, William Davidson Richards, Stephen Dacek, Shreyas Cholia, Dan Gunter, David Skinner, Gerbrand Ceder, and Kristin A. Persson. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1), July 2013.

[11] Harveen Kaur, Flaviano Della Pia, Ilyes Batatia, Xavier R Advincula, Benjamin X Shi, Jinggang Lan, Gábor Csányi, Angelos Michaelides, and Venkat Kapil. Data-efficient fine-tuning of foundational models for first-principles quality sublimation enthalpies. *arXiv preprint arXiv:2405.20217*, 2024.

[12] Gurjot Dhaliwal, Prasanth B. Nair, and Chandra Veer Singh. Machine learned interatomic potentials using random features. *npj Computational Materials*, 8(1), January 2022.

[13] John Falk, Luigi Bonati, Pietro Novelli, Michele Parrinello, and Massimiliano Pontil. Transfer learning for atomistic simulations using gnns and kernel mean embeddings. *Advances in Neural Information Processing Systems*, 36, 2024.

[14] Alessandro Rudi and Lorenzo Rosasco. Generalization properties of learning with random features. *Advances in neural information processing systems*, 30, 2017.

[15] Song Mei and Andrea Montanari. The generalization error of random features regression: Precise asymptotics and the double descent curve. *Communications on Pure and Applied Mathematics*, 75(4):667–766, 2022.

[16] Yutack Park, Jaesun Kim, Seungwoo Hwang, and Seungwu Han. Scalable parallel algorithm for graph neural network interatomic potentials in molecular dynamics simulations. *Journal of Chemical Theory and Computation*, 20(11):4857–4868, 2024.

[17] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

[18] Ninh Pham and Rasmus Pagh. Fast and scalable polynomial kernels via explicit feature maps. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 239–247, 2013.

[19] Felix Xinnan X Yu, Ananda Theertha Suresh, Krzysztof M Choromanski, Daniel N Holtmann-Rice, and Sanjiv Kumar. Orthogonal random features. *Advances in neural information processing systems*, 29, 2016.

[20] I. Steinwart and A. Christmann. *Support Vector Machines*. Information Science and Statistics. Springer New York, 2008.

[21] Krzysztof Choromanski, Haoxian Chen, Han Lin, Yuanzhe Ma, Arijit Sehanobish, Deepali Jain, Michael S Ryoo, Jake Varley, Andy Zeng, Valerii Likhosherstov, et al. Hybrid random features. *arXiv preprint arXiv:2110.04367*, 2021.

[22] Xiang Fu, Zhenghao Wu, Wujie Wang, Tian Xie, Sinan Keten, Rafael Gomez-Bombarelli, and Tommi S. Jaakkola. Forces are not enough: Benchmark and critical evaluation for machine learning force fields with molecular simulations. *Transactions on Machine Learning Research*, 2023.

[23] Cameron J Owen, Steven B Torrisi, Yu Xie, Simon Batzner, Kyle Bystrom, Jennifer Coulter, Albert Musaelian, Lixin Sun, and Boris Kozinsky. Complexity of many-body interactions in transition metals via machine-learned force fields from the tm23 data set. *npj Computational Materials*, 10(1):92, 2024.

[24] Jonathan Vandermause, Steven B. Torrisi, Simon Batzner, Yu Xie, Lixin Sun, Alexie M. Kolpak, and Boris Kozinsky. On-the-fly active learning of interpretable bayesian force fields for atomistic rare events. *npj Computational Materials*, 6(1), March 2020.

Supplementary Material

A Training algorithm

Training *franken* requires to minimize the loss function Eq. (1). The convexity properties of RF models ensure that there exists a vector of coefficients $\boldsymbol{w}^*$ which *globally* minimizes Eq. (1). Without loss of generality, we now show how the global minimizer can be computed in the case $\alpha = 0$, that is

$$\ell_0(\boldsymbol{w}) = \ell_{\mathrm{E}}(\boldsymbol{w}) = \sum_{t=1}^{T} \left(E(\boldsymbol{R}_t; \boldsymbol{w}) - E_t\right)^2 + \lambda_{\mathrm{E}} \|\boldsymbol{w}\|^2, \tag{2}$$

where we added a standard L^2 regularization term $\lambda_{\mathrm{E}}\|\boldsymbol{w}\|^2$. When $\alpha \neq 0$, the derivation is exactly the same. To minimize Eq. (2) over a training dataset $\mathcal{D} := (\boldsymbol{R}_t; E_t, \boldsymbol{F}_t)_{t=1}^{T}$ let's first recall that

$$E(\boldsymbol{R}_t; \boldsymbol{w}) = N_t \sum_{n=1}^{N} \epsilon_n(\boldsymbol{R}; \boldsymbol{w}) = \sum_{n=1}^{N} \langle \phi_n(\boldsymbol{R}_t), \boldsymbol{w} \rangle = \left\langle \sum_{n=1}^{N} \phi_n(\boldsymbol{R}_t), \boldsymbol{w} \right\rangle =: \langle \bar{\phi}(\boldsymbol{R}_t), \boldsymbol{w} \rangle$$

where N_t is the number of atoms in the t-th configuration and we have defined $\bar{\phi}(\boldsymbol{R}_t) := \sum_{n=1}^{N} \phi_n(\boldsymbol{R}_t)$. Plugging the definition of the energy back into the loss function and taking the gradient of $\ell_{\mathrm{E}}(\boldsymbol{w})$ one gets:

$$\nabla_{\boldsymbol{w}} \ell_{\mathrm{E}}(\boldsymbol{w}) = 2 \left(\lambda \mathbb{1}_D + \sum_{t=1}^{T} N_t \bar{\phi}(\boldsymbol{R}_t) \otimes \bar{\phi}(\boldsymbol{R}_t) \right) \boldsymbol{w} - 2 \sum_{t=1}^{T} E_t \bar{\phi}(\boldsymbol{R}_t). \tag{3}$$

Let us now define the covariance matrix C and coefficient vector b as

$$\mathsf{C} := \sum_{t=1}^{T} N_t \bar{\phi}(\boldsymbol{R}_t) \otimes \bar{\phi}(\boldsymbol{R}_t) \in \mathbb{R}^{D \times D} \qquad \mathsf{b} := \sum_{t=1}^{T} E_t \bar{\phi}(\boldsymbol{R}_t) \in \mathbb{R}^{D}. \tag{4}$$

Since the problem Eq. (2) is strongly convex, it has a unique global minimizer corresponding to the solution of $\nabla_{\boldsymbol{w}} \ell_{\mathrm{E}}(\boldsymbol{w}) = \boldsymbol{0}$. Using the definitions in (4) one has:

$$\boldsymbol{w}^* = (\lambda \mathbb{1}_D + \mathsf{C})^{-1} \mathsf{b}. \tag{5}$$

Computational cost & parallelization

The bulk of the computation in *franken*'s algorithm is concentrated in computing Eqs. (5) and (4). Given that the number of random features is usually on the order of a few thousand, the solution of the linear system in Eq. (5) is extremely fast. On the other hand, computing the covariance in (4) requires a full pass over the backbone GNN (and in the case of forces, it also requires a backward pass). This has been consistently the most time consuming step in our experiments. Yet, since the covariance is just a sum of independent terms, its computation can be easily parallelized across different GPUs, reducing the training time by a factor of the number of GPUs.