SCIENCE ADVANCES | RESEARCH ARTICLE

CHEMICAL PHYSICS

Accurate global machine learning force fields for molecules with hundreds of atoms

Stefan Chmiela[1,2]*, Valentin Vassilev-Galindo[3], Oliver T. Unke[4,1], Adil Kabylda[3], Huziel E. Sauceda[5,1,6,2], Alexandre Tkatchenko[3]*, Klaus-Robert Müller[1,2,4,7,8]*

Global machine learning force fields, with the capacity to capture collective interactions in molecular systems, now scale up to a few dozen atoms due to considerable growth of model complexity with system size. For larger molecules, locality assumptions are introduced, with the consequence that nonlocal interactions are not described. Here, we develop an exact iterative approach to train global symmetric gradient domain machine learning (sGDML) force fields (FFs) for several hundred atoms, without resorting to any potentially uncontrolled approximations. All atomic degrees of freedom remain correlated in the global sGDML FF, allowing the accurate description of complex molecules and materials that present phenomena with far-reaching characteristic correlation lengths. We assess the accuracy and efficiency of sGDML on a newly developed MD22 benchmark dataset containing molecules from 42 to 370 atoms. The robustness of our approach is demonstrated in nanosecond path-integral molecular dynamics simulations for supramolecular complexes in the MD22 dataset.

INTRODUCTION

Modern machine learning force fields (MLFFs) bridge the accuracy gap between highly efficient but exceedingly approximate classical force fields (FFs) and prohibitively expensive high-level ab initio methods (*1*–*3*). This optimism is based on the universal nature of machine learning (ML) models, which gives them virtually unrestricted descriptive power compared to the statically parameterized interactions in classical mechanistic FFs. Traditional ML approaches strive toward general assumptions about the problem at hand, such as continuity and differentiability, when constructing models. In principle, any physical property of interest captured in a dataset can be parameterized this way, including collective interactions that are too intricate to extract from the many-body wave function. Hence, MLFFs can give unprecedented insights into quantum many-body mechanisms (*1*). Albeit, the exceptional expressive power of global MLFFs goes along with a stark increase in parametric complexity (*4*–*6*) over classical FFs.

As a trade-off, many ML models reintroduce some of the classical mechanistic restrictions on the allowed interactions between atoms. It is unclear to which extent this departure from unbiased ML models compromises their advantages over classical FFs. In particular, localization assumptions are often made to allow a reduction of degrees of freedom across large structures. For example, message-passing neural networks only allow mean-field exchanges between local atomic neighborhoods, which leads to information loss over long distances (*7*). This causes a truncation of long-range interactions, which, in local models, are assumed to have a rather small contribution to the overall dynamics of the system. Nonetheless, it has been shown that long-range effects can play an important role (*8*–*12*), limiting the predictive power of local models in nanoscale and mesoscale systems (*13*, *14*). Several recent MLFF models (*8*, *9*, *15*–*20*) introduce empirical correction terms for specific long-range effects (e.g., electrostatics), yet long-range electron correlation effects remain poorly characterized. The number of available MLFF approaches augmented with physical interaction models indicates that we are observing an emerging field that has not yet settled on a universal solution.

In contrast, global models (*21*–*26*) are able to include all interaction scales, but they face the challenge of having to couple at least a quadratic set of atom-atom interactions (see Fig. 1). Such scaling behavior provides a hard computational constraint and has therefore slowed the development of global models in recent years. Current global models are thus restricted to system sizes of only a few dozen atoms, although accurate ab initio reference data are available for much bigger systems. Here, we develop a combined closed-form and iterative approach to train global MLFF kernel models for large molecules. Our spectral analysis of these models (see Fig. 2) demonstrates that the number of effective degrees of freedom in large molecules is substantially reduced compared to N^2 and can be captured using a low-dimensional representation (*27*, *28*). Using this insight, our large-scale framework lowers the memory and computational time requirements of the model simultaneously. In a two-step procedure, the effective degrees of freedom are solved in closed-form, before iteratively converging the remaining fluctuations to the exact solution for the full problem.

Our focus is on ML models based on Gaussian processes (GPs), because they have several unique properties such as linearity and loss function convexity that can be exploited in pursuit of our goal. We demonstrate the effectiveness of our solution on the symmetric gradient domain ML (sGDML) FF (*22*, *23*). It allows us to reliably reconstruct sGDML FFs for significantly larger molecules and materials than previously possible (*29*). Our training scheme can handle systems that contain several hundreds of atoms, all of

---

[1]Machine Learning Group, Technische Universität Berlin, 10587 Berlin, Germany. [2]Berlin Institute for the Foundations of Learning and Data – BIFOLD, Germany. [3]Department of Physics and Materials Science, University of Luxembourg, L-1511 Luxembourg City, Luxembourg. [4]Google Research, Brain Team, Berlin, Germany. [5]Departamento de Materia Condensada, Instituto de Física, Universidad Nacional Autónoma de México, Cd. de México C.P. 04510, Mexico. [6]BASLEARN - TU Berlin/BASF Joint Lab for Machine Learning, Technische Universität Berlin, 10587 Berlin, Germany. [7]Max Planck Institute for Informatics, Stuhlsatzenhausweg, 66123 Saarbrücken, Germany. [8]Department of Artificial Intelligence, Korea University, Anam-dong, Seongbuk-gu, Seoul 02841, Korea.

*Corresponding author. Email: stefan@chmiela.com (S.C.); alexandre.tkatchenko@uni.lu (A.T.); klaus-robert.mueller@tu-berlin.de (K.-R.M.)

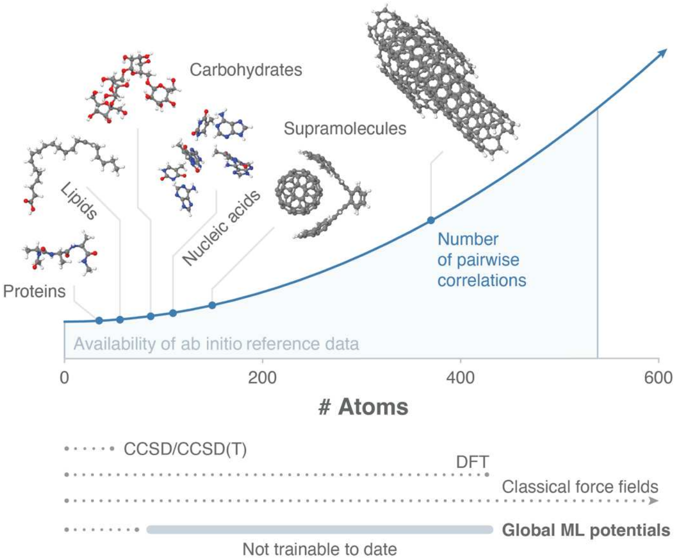

Fig. 1. Current global MLFFs only scale to system sizes of a few dozen atoms, restricted by the computational challenge of having to couple a quadratic amount of atom-atom interactions. However, accurate ab initio reference data are available for much bigger systems (light blue area). This work scales global models with ab initio accuracy to hundreds of atoms, as is demonstrated on examples from four major classes of biomolecules and supramolecules.

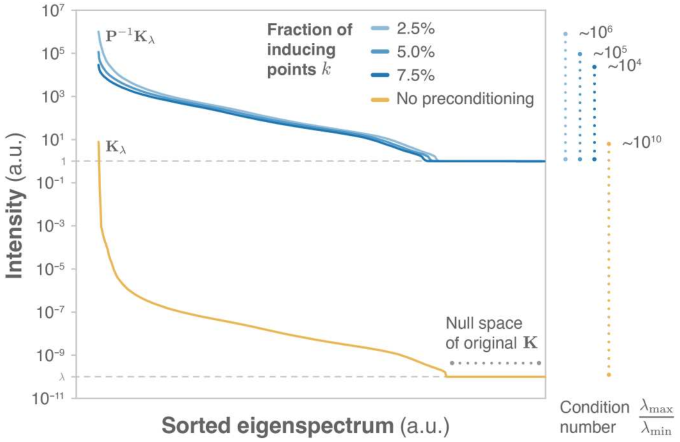

Fig. 2. Eigenvalue spectrum of the regularized sGDML kernel matrix $\mathbf{K}_\lambda$ (malonaldehyde, σ=12, $\lambda=10^{-10}$) before and after preconditioning $\mathbf{P}^{-1}\mathbf{K}_\lambda$ with an increasing number of inducing points *k* (2.5, 5, and 7.5% of the total number of training points *m*). Note that regularization lifts the null-space of the original kernel matrix to λ. Because the same regularization is applied to the preconditioner, the spectrum of $\mathbf{P}^{-1}\mathbf{K}_\lambda$ is thus bounded from below by 1. The dominant part of the spectrum is increasingly attenuated by the preconditioner, reducing the condition number of the matrix and thus accelerating convergence of the CG algorithm. a.u., arbitrary units.

which are fully coupled within the model. We demonstrate that this parametric flexibility is leveraged to let all atoms participate in generating the energy and force predictions. Our development allows us to study supramolecular complexes, nanostructures, and four major classes of biomolecular systems in stable nanosecond-long molecular dynamics (MD) simulations. All of these systems present phenomena with far-reaching characteristic correlation lengths. We offer these datasets as a benchmark (called MD22) that presents new challenges with respect to system size (42 to 370 atoms), flexibility, and degree of nonlocality. Hence, MD22 can be regarded as the next generation of the now well-established MD17 dataset (*22*).

RESULTS

Large-scale sGDML algorithm

A data-efficient reconstruction of FFs with strong generalization properties requires models that implement the appropriate prior knowledge to compensate for finite reference dataset sizes. Quantum chemical interactions are highly complex and can thus not be fully specified even by massive datasets. We therefore exploit that many complex interactions can be summarized in terms of simple constraints derived from conservation laws that are far more effective. GPs provide a particularly1 elegant way of incorporating such constraints, because they are closed with respect to linear transformations, such as integral operators or partial differential equations. A combination of linear constraints can be written as a new instance of a GP. This principle is used by the sGDML approach to construct models that implement all important invariance properties of MLFFs (*22*). One of its key characteristics is the use of a kernel $\mathbf{k}(\mathbf{x},\mathbf{x}') = \nabla_\mathbf{x} k_E(\mathbf{x},\mathbf{x}')\nabla_{\mathbf{x}'}^\top$ that models the FF $\mathbf{f}_\mathbf{F}$ as a transformation of an unknown potential energy surface f_E such that

$$\mathbf{f}_\mathbf{F} = -\nabla f_E \sim \mathcal{GP}[-\nabla \mu_E(\mathbf{X}), \nabla_\mathbf{x} k_E(\mathbf{x},\mathbf{x}')\nabla_{\mathbf{x}'}^\top] \tag{1}$$

Here, μ_E: $\mathbb{R}^d \to \mathbb{R}$ and $k_E : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ are the respective prior mean and prior covariance functions that define the latent energy–based GP. The training on force examples is motivated by the fact that they are available analytically from electronic structure calculations via the Hellmann-Feynman theorem, with only moderate computational overhead atop energy measurements. Force samples constitute a more efficient way to generate reference data, which sGDML is able to take advantage of (*30*). Despite their nonparametric nature, sGDML FFs generally use around one order of magnitude fewer parameters than deep neural network architectures with comparable accuracy, making them computationally less expensive to evaluate (see Appendix C).

GPs can be solved in closed form, because their convex loss function yields a linear system when its derivative is set to zero (*31*). The resulting system $\alpha = (\mathbf{K} + \lambda\mathbb{I})^{-1}\mathbf{y}$ is symmetric positive definite by definition of the kernel function, which allows a solution via Cholesky decomposition (*32*, *33*) (see the "Cholesky decomposition" section). While this approach is efficient and numerically stable, it does not scale to large matrix sizes due to its quadratic memory cost. In search for a solution, a variety of approximation techniques have emerged (*34*–*41*), all drawing on the same insight that kernel matrices often have a comparatively small numerical rank (i.e., a rapidly decaying eigenvalue spectrum; see Fig. 2) that can be exploited to formulate a lower-dimensional problem. However, this is only the case with noisy training data, which causes many degrees of freedom of the kernel matrix to become dispensable. Ab initio reference calculations are essentially noise-free, giving rise to kernel matrices that cannot be compressed without compromising the accuracy of the predictor (*42*). Therefore, continued efforts are being made toward enabling exact large-scale GP inference via numerical gradient-based optimization schemes as opposed to analytical matrix decomposition approaches (*43*–*47*). Those techniques exploit that the kernel matrix $\hat{f}(\mathbf{x})$ only appears as a matrix-vector product in the gradient $\nabla_\alpha \mathcal{L}[\hat{f}(\mathbf{x}), \mathbf{y}]$ of the loss function

$$\begin{aligned}\mathcal{L}[\hat{f}(\mathbf{x}), \mathbf{y}] &= [\hat{f}(\mathbf{x}) - \mathbf{y}]^2 + \lambda \parallel \hat{f} \parallel^2 \\ &= \mathbf{y}^\top \mathbf{y} - (2\mathbf{y}^\top + \alpha^\top \mathbf{K} + \lambda \alpha^\top)\mathbf{K}\alpha\end{aligned} \tag{2}$$

which can be evaluated on the fly without ever storing **K**. Using a gradient descent scheme, the parameters are then updated until convergence to the exact solution (or as close as necessary) using iterates of the form

$$\alpha_t = \alpha_{t-1} - \gamma \underbrace{[(\mathbf{K} + \lambda \mathbb{I})\alpha_{t-1} - \mathbf{y}]}_{\nabla_\alpha \mathcal{L}[\hat{f}(\mathbf{x}), \mathbf{y}]} \tag{3}$$

Here, the hyperparameter γ determines the step size (learning rate) with which the minimum of the loss function is approached. In practice, the convergence of this algorithm is severely impeded in regions with large differences in curvature, causing the iterates to jump back and forth across narrow valleys, which slows down progress in some directions on the loss surface (see Fig. 3).

The conjugate gradient (CG) descent algorithm (*48*–*50*) avoids this behavior by exploiting the special topography of GP loss surfaces to make optimal choices of step sizes γ_t and descent directions $\mathbf{p}_t$. It expresses the solution $\alpha = \sum_{t=1}^{m} \gamma_t \mathbf{p}_t$ in a basis of *m* mutually conjugate vectors with respect to $\mathbf{K}_\lambda = \mathbf{K} + \lambda \mathbb{I}$, such that, for any pair, $\mathbf{p}_i^\top \mathbf{K}_\lambda \mathbf{p}_j = 0$. Instead of simply following the negative gradient $\mathbf{r}_t = \nabla_\alpha \mathcal{L}[\hat{f}(\mathbf{x}), \mathbf{y}]$ as in Eq. 3, each gradient step is projected away from all previous minimization directions

$$\mathbf{p}_t = \mathbf{r}_t - \sum_{i<t} \frac{\mathbf{p}_i^\top \mathbf{K}_\lambda \mathbf{r}_t}{\mathbf{p}_i^\top \mathbf{K}_\lambda \mathbf{p}_i} \mathbf{p}_i \tag{4}$$

Furthermore, rather than fixing the learning rate heuristically, the convexity of the learning problem allows an optimal choice at each iteration step

$$\gamma_t = \frac{\langle \mathbf{p}_t, \mathbf{r}_t \rangle}{\langle \mathbf{p}_t, \mathbf{p}_t \rangle_{\mathbf{K}_\lambda}} \tag{5}$$

Together, the iterates $\alpha_{t+1} = \alpha_t - \gamma_t \mathbf{p}_t$ look very similar to Eq. 3. In contrast to basic gradient descent schemes, this algorithm is parameter-free, foregoing a laborious parameter search. The CG algorithm can be reformulated in a memory efficient way that avoids storing all previous search directions and residual vectors that appear in Eq. 4, using the mutual orthogonality of all $\mathbf{r}_i$ and $\mathbf{p}_i$. For $\mathbf{K}_\lambda$ with fast-decaying eigenvalue spectra [as is almost always the case, see Marchenko-Pastur distribution (*51*)], the CG expansion can typically be truncated early (i.e., at $t < m$) to reach superlinear convergence in practice (*50*).

One important caveat of the CG algorithm is, however, its numerical robustness. The number of required iteration steps until convergence $m \sim \mathcal{O}(\kappa)$ is proportional to the condition number $\kappa = \lambda_{\max}/\lambda_{\min}$ (the ratio between largest and smallest eigenvalue) of $\mathbf{K}_\lambda$, which is notoriously bad for correlation matrices, with a characteristic steep spectral drop-off. A direct application of this iterative scheme will therefore lead to impracticably slow convergence or even divergence due to numerical errors (see Figs. 2 and 4) (*52*, *53*). The standard way of dealing with ill-conditioned problems is to reformulate the original linear system to an equivalent one, for which the iterative solver converges faster. Using a nonsingular preconditioning matrix **P**, we can write

$$\alpha = (\mathbf{P}^{-1}\mathbf{K}_\lambda)^{-1}\mathbf{P}^{-1}\mathbf{y} \tag{6}$$

to make the convergence depend on the condition number of $\mathbf{P}^{-1}\mathbf{K}_\lambda$ rather than $\mathbf{K}_\lambda$. To continue to satisfy the symmetric and positive definiteness assumptions of the CG algorithm, we require that $\mathbf{P}^{-1} = \mathbf{Q}\mathbf{Q}^\top$ exits, although the algorithm can be formulated in a way that does not require explicit factorization.

For any preconditioner to be practical, the cost of solving $\mathbf{P}^{-1}\mathbf{K}_\lambda$ must be low (*46*). This requirement rules out optimal rank-*k*

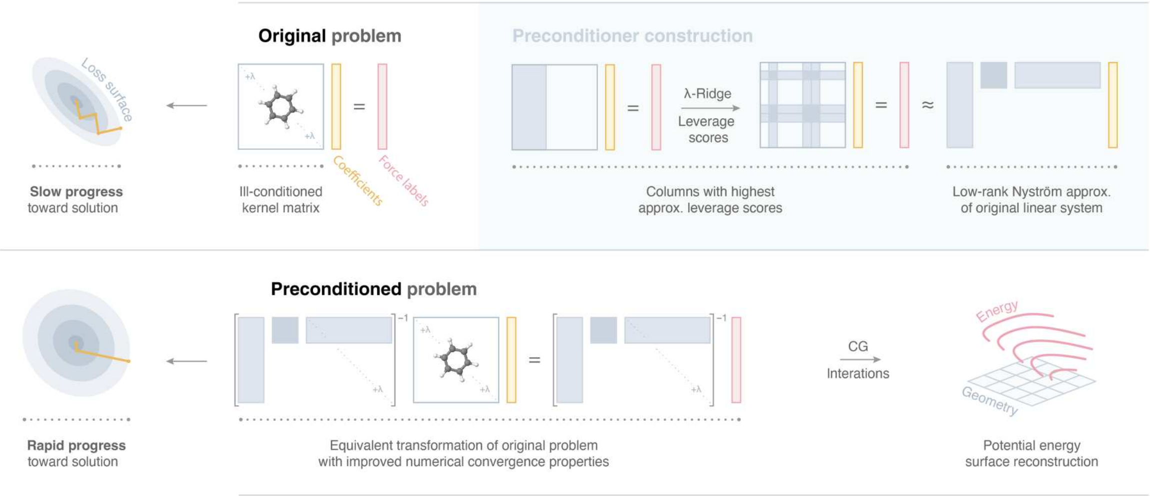

Fig. 3. The convergence rate of iterative GP training algorithms is inversely proportional to the condition number of the kernel matrix $\mathbf{K}_\lambda$, which is generally large for strongly correlated many-body systems. Using a preconditioner **P**, the original linear system is reformulated into an equivalent one $\mathbf{P}^{-1}\mathbf{K}_\lambda\alpha = \mathbf{P}^{-1}\mathbf{y}$ that is numerically easier to solve. Approximate statistical leverage scores in combination with the Nyström approximation are used to construct a memory efficient and easily invertible representation of **P**.

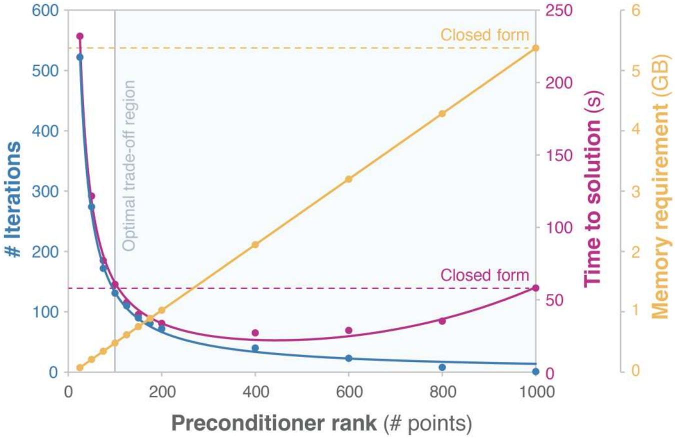

Fig. 4. Number of iterations required to train a sGDML model for malonaldehyde (sampled at 500 K, 1000 training points, σ=12, target residual error below 10^{-4}) as a function of preconditioner strength (number of inducing points). The iterative solver allows a trade-off between computational time and memory complexity. While the time to construct the preconditioner (Eq. 8) scales cubically, the increase in memory demand is linear in k. Using all training points for preconditioning ($\mathbf{P}^{-1} = \mathbf{K}_{\lambda}^{-1}$) gives the analytic solution (Cholesky decomposition) in one step. A quickly converging iterative solver can, however, reduce training time and memory demand at the same time (light blue region).

approximations obtained from the dominant k-dimensional eigenspace of $\mathbf{K}$, as such constructions incur cubic computational cost $\mathcal{O}(m^3)$ (*54*). Even algorithms that only compute the leading eigenvalues and eigenvectors are not significantly more efficient, except when $k \ll m$. Here, we use the Nyström method (*34*) as an alternative, to compute a low-cost approximation to that relevant eigenspace. Accordingly, the original kernel matrix is approximated at a cost of only $\mathcal{O}(k^2 m)$ from a subset of k of columns as

$$\mathbf{K} \approx \hat{\mathbf{K}} = \mathbf{K}_{mk}\mathbf{K}_{kk}^{-1}\mathbf{K}_{mk}^{\top} \tag{7}$$

The Woodbury matrix identity (*54*) is then used to efficiently invert $\mathbf{P} = \hat{\mathbf{K}} + \lambda\mathbb{I}$ in parts

$$\mathbf{P}^{-1} = \lambda^{-1}\left[\mathbb{I} - \mathbf{K}_{mk}\left(\lambda\mathbf{K}_{kk} + \mathbf{K}_{mk}^{\top}\mathbf{K}_{mk}\right)^{-1}\mathbf{K}_{mk}^{\top}\right] \tag{8}$$

There are many strategies to select the column subspace, the most straightforward of which is simple random sampling (*34*, *55*). Here, we chose a more effective yet still computationally economical approach based on approximate statistical leverage scores (*56*–*60*) (see the "Approximate λ-ridge leverage scores" section).

Overall, the accuracy of the approximation $\hat{\mathbf{K}}$ is primarily determined by its dimension k and so is the cost of inverting and applying $\mathbf{P}$, with runtime and memory complexities of $\mathcal{O}(mk^2)$ and $\mathcal{O}(mk)$, respectively. A stronger preconditioner will accelerate convergence, albeit at increased cost per iteration (see Fig. 4). Note that the performance and effectiveness of the outlined method crucially depend on a careful numerical implementation, as explained in the "Numerical implementation" section.

Assessment of large-scale molecular FFs

Convergence properties

To begin assessing the effectiveness of our approach, we investigate the role of the preconditioner configuration on the convergence behavior of the CG solver. In particular, we are interested in assessing the impact of the preconditioner strength on the overall training cost. A larger number of inducing points not only increase the time and memory requirements to construct $\mathbf{P}^{-1}$ but also facilitate convergence in fewer steps. The purpose of this test is to examine how effectively the critical memory restrictions of closed-form solvers can be lifted in practice. The type of chemical system represented by the training dataset is largely irrelevant for this experiment, as the convergence behavior is mainly determined by the quickly decaying spectrum of the kernel matrix (*27*), which is characteristic for strongly interacting many-body systems (see Fig. 2).

We have representatively sampled 1000 points from the malonaldehyde trajectory in the MD17 dataset (*22*), which were trained with increasing preconditioning strength ranging from 2.5 to 100%. For each configuration, the runtime and memory usage of the solver are measured (including the construction of the preconditioner). Our analysis (see Fig. 4) shows that the CG solver converges with consistent reliability when more than around 2.5% of the columns are used to construct the preconditioner. In this configuration, the memory footprint is very low, but the runtime exceeds that of the closed-form solver. However, if more than 10% are used, the memory and runtime complexity are both lower with the iterative training approach. Because of the cubic scaling in the construction of $\mathbf{P}^{-1}$, the marginal benefit of increasing the preconditioner strength eventually diminishes, and the runtime starts increasing again, along with the memory complexity. This means that smaller preconditioners not only are cheaper to construct and store but also actually perform better overall (see Fig. 4).

Scalability

Although our solver enables significantly larger training datasets than before (see Appendix B), we focus on the more practical example of scaling up system size. After all, the cost of generating massive reference datasets overbears any speed up that an ML model could provide. At large scales, atomic interactions become potentially more complex, as they involve a broader spectrum of length scales. It is this scenario, in which the combined scalabilty and data efficiency an ML model is really needed.

To put the iterative sGDML solver to the ultimate test, we have generated a new set of MD trajectories (MD22) that cover systems of up to several hundred atoms. MD22 includes examples of four major classes of biomolecules and supramolecules, ranging from a small peptide with 42 atoms, all the way up to a double-walled nanotube with 370 atoms (see tables SI and SII). The trajectories were sampled at temperatures between 400 and 500 K at a resolution of 1 fs, with corresponding potential energy and atomic forces calculated at the PBE+MBD (*61*, *62*) level of theory. Compared to the well-established MD17 benchmark (*22*), the SDs of the potential energies are significantly larger, varying between ~8 and 77 kcal mol $^{-1}$ (MD17, ~2 to 6 kcal mol^{-1}). The SDs of the forces are, however, close between ~21 and 28 kcal mol^{-1} Å^{-1} (MD17, ~20 to 30 kcal mol^{-1}). We set the training dataset size for each of the systems, such that the root mean square test error for predicting atomic forces is around 1 kcal mol^{-1} A^{-1}. For some systems like the buckyball catcher (148 atoms) or the double-walled nanotube (370 atoms), this error is already achieved with small training set sizes of only a few hundred points. Other systems, e.g., DHA (docosahexaenoic acid) (56 atoms), stachyose (87 atoms), or the Ac-Ala3-NHMe peptide (42 atoms), require several thousands of training points for the same force prediction accuracy. The corresponding energy mean absolute errors (MAEs) range between 0.39 kcal

mol^{-1} (Ac-Ala3-NHMe) and 4.01 kcal mol^{-1} (double-walled nanotube), which is in line with our previous results on MD17 (*22*) when normalized per atom. The (independent) random errors in each atomic contribution to the overall energy prediction approximately propagate as the square root of sum of squares, which causes the energy error to scale with system size (*63*). This scaling behavior is confirmed when comparing the energy MAE per atom, which is consistently around 0.01 kcal mol^{-1} for most datasets in our study. We observe that the complexity of the learning task is correlated neither with the number of atoms nor with the simulation temperature of the reference trajectory. Rather, the difficulty to reconstruct an FF is determined by the complexity of the interactions within the system (see Table 1).

Representation of nonlocal interactions

To investigate whether the trained sGDML models provide chemically meaningful predictions, we apply sGDML to a donor-bridge-acceptor–type molecule consisting of two phenyl rings connected by an E-ethylene moiety forming a conjugated π-system (bridge) (see the "Donor-bridge-acceptor dataset" section for details on this dataset). The phenyl rings are substituted in para-position with an electron-donating dimethylamine group (donor) and an electron-withdrawing nitro group (acceptor), respectively. When the phenyl rings are coplanar, electrons are delocalized over the whole molecule and can freely "flow" from donor to acceptor. However, when the two phenyl rings are rotated against each other, the conjugation of the π-orbitals is broken and the favorable interaction between donor and acceptor is lost, increasing the potential energy of the molecule. A chemically meaningful model should predict that this energy change is delocalized over the whole π-system (as opposed to explaining it by local changes in the vicinity of the center of the rotation). To get a qualitative understanding of how these interactions are handled within the sGDML model, we investigate how individual atoms contribute toward the prediction. Being a linear combination of pairwise correlations between atoms, a partial evaluation of the model reveals the atomic contributions to each prediction (see Fig. 5). We observe that all atoms in the system participate in generating the prediction with sGDML, which would not be possible with a model that partitions the energy into localized atomic contributions. Figure 5 demonstrates that an sGDML model learns to delocalize changes in energy upon ring rotation across the whole molecule, which is in accordance with chemical intuition. Note that, starting from the global minimum structure of this system, rotating by π does not return to the starting position (despite the apparent symmetry of the molecule), because of a slight asymmetry about the central C═C bond (see overlay of structures in Fig. 5). Thus, a full rotation is necessary to return to the starting point, explaining the somewhat counterintuitive rotational energy profile.

Molecular dynamics

One of the biggest advantages of using MLFFs is that they can enable accurate large-scale simulations. Here, we test our sGDML FFs by running nanosecond-long classical MD and path-integral MD (PIMD) simulations for the double-walled nanotube saturated with hydrogen atoms at its edges. All simulations were run at a constant temperature of 300 K with a Langevin thermostat and a time step of 0.2 fs. The number of beads of the PIMD simulations was set to 16.

To confirm the reliability of any MLFF, it is first and foremost essential to assess its capability to yield stable (PI)MD simulations (*64*). In this regard, fig. S2 shows the cumulative potential energy ($V_{\text{step}} = \frac{1}{N_{\text{step}}}\sum_{k=1}^{N_{\text{step}}} V_k$, where N_{step} is the number of completed time steps, and V_k is the potential energy at the k-th step) along both MD and PIMD simulations. After thermalization (roughly 500 ps for MDs and 100 ps for PIMDs), the simulations reach equilibrium. Next, we compare the difference in geometry fluctuations between MD and PIMD simulations in Fig. 6, measured by root mean square deviations (RMSDs) from the initial geometry. The RMSDs of the PIMD present a series of peaks that are higher than those observed in the classical MD simulation. The first of

Table 1. sGDML prediction performance on large-scale datasets. All (test) errors are in kcal mol^{-1} (Å^{-1}) per atom (energy) or component (forces). The training set sizes were chosen such that the root mean square error (RMSE) of the force prediction is around 1 kcal mol^{-1} Å^{-1}. We also provide the corresponding mean absolute errors (MAEs). $\|G\|$ denotes the cardinality of the leveraged permutation group for each respective dataset [see (*23*, *30*, *88*) for details].

	Type	# Atoms	‖*G*‖	# Train.	% Data	MAE		RMSE	
						Energy	Forces	Energy	Forces
Proteins									
Ac-Ala3-NHMe	Tetrapeptide	42	18	6000	7%	0.0093	0.79	0.012	1.21
Lipids									
DHA (docosahexaenoic acid)	Fatty acid	56	6	8000	12%	0.023	0.75	0.03	1.17
Carbohydrates									
Stachyose	Tetrasaccharide	87	1	8000	29%	0.046	0.68	0.052	1.07
Nucleic acids									
AT-AT	DNA base pairs	60	36	3000	15%	0.012	0.69	0.015	1.12
AT-AT-CG-CG	DNA base pairs	118	96	2000	20%	0.012	0.7	0.015	1.22
Supramolecules									
Buckyball catcher		148	48	600	10%	0.0079	0.68	0.0099	1.02
Double-walled nanotube		370	28	800	16%	0.0108	0.52	0.0135	0.97

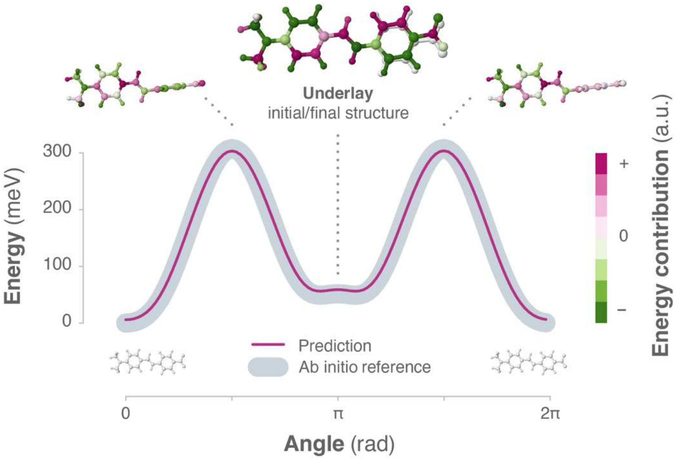

Fig. 5. Energy contributions as predicted by the sGDML FF for a donor-bridge-acceptor–type molecule (4-dimethylamino-4′-nitrostilbene). The energy profile for a full rotation around the single bond between the acceptor and the ethylene moiety is shown. When the conjugation of the π-system is broken upon rotating the phenyl rings by 90° against each other, the sGDML model predicts that the energy change is delocalized across the whole molecule.

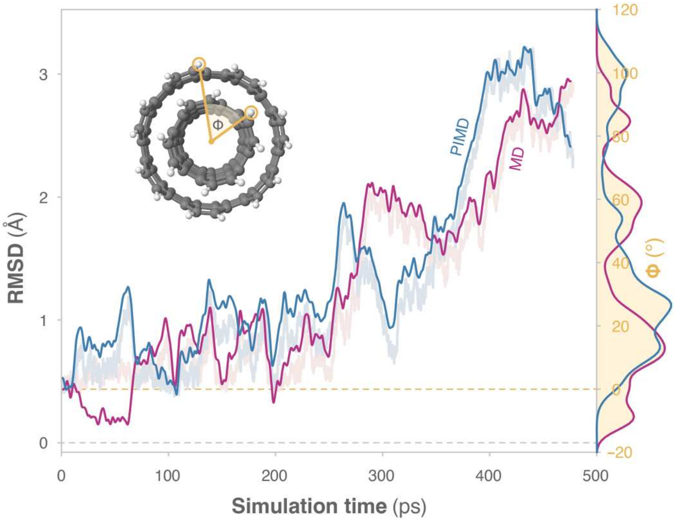

Fig. 6. Instantaneous root-mean squared deviation (RMSD in Ångström; transparent lines in the background) and angle of the relative rotation (Φ in degrees) between both nanotubes as a function of simulation time (in picoseconds). The RMSDs were computed with respect to the initial configurations used for the simulations.

such peaks appears at ~60 ps, and then there is one every 100 ps (the highest one corresponding to a RMSD greater than 3.0 Å). The origin of these RMSD fluctuations is the relative angle of rotation (Φ) of the inner nanotube with respect to the outer one (see Fig. 6). The outer and inner nanotubes have a seven- and fourfold axis of rotation (the axis parallel to the nanotubes), respectively, meaning that we have the same configuration every ~13°. However, RMSDs are dependent on atom indices, and uncoupled rotations of the nanotubes lead to a different arrangement of the atoms with respect to each other. This causes the increments observed in RMSDs along the simulations because no other degrees of freedom fluctuate as much as the angle Φ. We observe a strong correlation between the RMSD variations and the evolution of the angle Φ in a scale of 360° (a full rotation). The differences in values of Φ between the MD and PIMD simulations suggest that a coupling of nuclear quantum effects (NQEs) and long-range interactions [resembling existing studies on the stability of different aspirin crystal polymorphs (*65*)] eases the rotation of one of the nanotubes with respect to the other. The distribution of values of Φ further confirms that NQEs smoothen the rotational profile of the nanotubes. While in the PIMD values of Φ from 40° to 80° are equally sampled, in the MD, one can observe two pronounced peaks at around 60° and 80°. Note that the rather large time scale (100 ps) between each of these rotations indicates that this motion corresponds to low-frequency vibrational modes.

As a final demonstration of the capability of sGDML MLFF models for providing insights into large systems, we compute the molecular vibrational spectra of the buckyball catcher and the double-walled nanotube from MD and PIMD simulations (see fig. S3). These spectra correspond to velocity autocorrelation functions. The spectra feature peaks corresponding to ═C─H stretching modes (at around 3000 cm^{-1}) and bending modes (close to 1000 cm^{-1}), as well as peaks that correlate to C═C vibrations at around 500 and 1500 cm^{-1}. These account for expansions and contractions of the buckyball, the "hands" of the catcher and the nanotubes [for instance, see (*66*, *67*) for a discussion of the vibrational spectra of the buckyball].

Although MD and PIMD simulations provide similar spectra, the inclusion of NQEs yields more accurate frequencies. Namely, nuclear quantum delocalization leads to a shift of the ═C─H stretching mode, which puts the peak closer to 3000 cm^{-1}. This value is in agreement with other aromatic and $\pi-\pi$ interacting systems, e.g., with experimental and theoretical values of the fundamental C─H stretching modes of benzene and the benzene dimer [mainly the $\nu_{13}(\mathbf{B}_{1u})$ mode] (*68*). Hence, PIMD simulations capture some of the anharmonic behavior of the systems and correct the overestimated value of ≈3100 cm^{-1} in the classical MD. In contrast to the ═C─H stretching mode, the parts of the spectra corresponding to "long-range" vibrations (i.e., low-frequency modes) are, in general, consistent among MD and PIMD. This agreement aligns with the fact that both the buckyball catcher and the double-walled nanotube are relatively symmetric systems and that the differences between configuration spaces sampled by MD and PIMD simulations are mostly of local nature. Therefore, although NQEs promote a low-frequency mode such as the relative rotation between the nanotubes, these modes are smoothed out in the low-frequency part of the vibrational spectrum.

DISCUSSION

In conclusion, kernel-based FFs are known to be sample efficient but believed to be limited in their ability to scale well with training set or system size. A key reason is the common use of direct solvers, which factorize the kernel matrix to solve the associated optimization problem to train the model. While this approach is numerically stable, it quickly incurs a prohibitive memory and runtime complexity. This dilemma can be evaded using iterative solvers, which essentially allow kernel-based models to be trained similarly to neural networks. However, the straightforward application of iterative solvers is difficult when large kernel matrices are involved, due to their notoriously poor numerical conditioning.

In this work, we propose an iterative scheme that enables the robust application of sGDML to significantly larger systems (in terms of both training set and system size) without introducing any approximations to the original model. This is achieved with a numerical preconditioning scheme that markedly reduces the conditioning number of the learning problem, enabling rapid convergence of a CG iteration. With this advance, we are now able to apply our kernel-based model to large-scale learning tasks that have previously only been accessible to neural networks while carrying over the sample efficiency and accuracy of sGDML. We attribute the latter to the model's unique ability to represent global interactions on equal footing with local interactions, as a series of numerical experiments demonstrate. Now, molecular systems that exhibit phenomena with far-reaching characteristic correlation lengths can be studied in long–time scale MD simulations.

Breakthroughs in MLFF development are often driven by the creation of benchmark datasets that offer ever evolving challenges. Early quantum chemistry datasets, such as QM7 (*21*), MD17 (*22*), the valence electron densities for small organic molecules in (*69*), ISO17 (*70*), SN2 (*18*), or SchNOrb (*71*, *72*), focused on defining useful inference problems and opportunities in quantum chemistry, whereas later benchmarks [e.g., QM7-X (*73*)] steered the field toward developing more robust transferable models. The MD22 benchmark dataset developed in this work now offers new challenges for atomistic models with regard to molecular size and flexibility that could further advance research on novel MLFF architectures similarly to previous datasets of quantum mechanical calculations.

Our technical development allows future cross-fertilization between both MLFF development approaches: Now, kernel-based MLFFs can capitalize on the massive parallelism available on GPUs and the software infrastructure that enabled scalability of deep neural networks. On the other hand, modeling principles from kernel methods inspire the development of new architectures such as transformers using self-attention (*74*, *75*) and pave the way out of overly restrictive localization assumptions. The exclusive use of on-the-fly model evaluations by iterative solvers also represents a paradigm shift in the way kernel-based MLFFs are typically trained, which opens up new avenues for further development. We have recently shown, how this makes them amenable to powerful differential equation constraints via algorithmic differentiation techniques to simplify descriptor development and further improve data efficiency (*76*). With the ability to reconstruct MLFFs for larger systems, the need for better management of the growing set of molecular features arises. To this end, we have recently proposed a novel descriptor pruning scheme to contract trained models and make them easier to evaluate (*77*). One could also envision a systematic construction of local and nonlocal fragments [by generalizing from noninteracting to interacting amons (*78*)] that would enhance the scalability and transferability of global MLFFs. Future research will furthermore explore our significantly better scaling behavior across a broad range of application fields in the physical sciences.

MATERIALS AND METHODS

MD22 datasets

The MD22 benchmark dataset consists of MD trajectories covering seven systems from four major classes of biomolecules and supramolecules, ranging from a small peptide with 42 atoms to a double-walled nanotube with 370 atoms. The trajectories were sampled at temperatures between 400 and 500 K at a resolution of 1 fs, with corresponding potential energy and atomic forces calculated at the PBE+MBD (*61*, *62*) level of theory. More computational details are provided in tables SI and SII.

Donor-bridge-acceptor dataset

The reference data for the donor-bridge-acceptor example were generated by normal mode sampling (*79*) at 300 K with random rotations applied to individual bonds. Energies and forces were calculated using the semiempirical GFN2-xTB method (*80*). The energy profile in Fig. 5 shows a rotation around a single bond, with all other bond distances and angles fixed at their equilibrium values.

Cholesky decomposition

Because of the symmetric positive-definite properties of the kernel function, the system $\alpha = (\mathbf{K} + \lambda\mathbb{I})^{-1}\mathbf{y}$ can be solved using a Cholesky decomposition into a product of lower triangular factors $\mathbf{L} \in \mathbb{R}^{m \times m}$ (*32*, *33*)

$$\alpha = (\mathbf{K} + \lambda\mathbb{I})^{-1}\mathbf{y} = (\mathbf{L}\mathbf{L}^\top)^{-1}\mathbf{y} \tag{9}$$

and subsequent forward and backward substitution.

Approximate λ-ridge leverage scores

We use the Nyström method to project the (symmetric and positive semidefinite) kernel matrix $\mathbf{K}$ onto a subset of its columns to obtain a memory efficient and easily invertible approximation of $\mathbf{K}_\lambda$ as preconditioner for the learning problem. The choice of inducing columns determines the quality of the approximation, i.e., how well the eigenvalue spectrum of the original kernel matrix is preserved. Here, we use an extended notion of leverage score that is tailored to the context of GPs to determine a good set of inducing points. These so called statistical leverage scores depend on a regularization parameter λ that diminishes the importance of small eigendirections (*81*, *82*)

$$\tau_i(\lambda) = [\mathbf{K}(\mathbf{K} + \lambda\mathbb{I})^{-1}]_{ii} \tag{10}$$

Note that $\tau_i(\lambda)$ is the *i*th diagonal entry of the matrix product. The exact computation of $\tau_i(\lambda)$ is, however, not practical as it requires inverting $\mathbf{K}$, which is as expensive as solving the original learning problem (*83*). Instead, approximate λ-ridge leverage scores are obtained on the basis of another Nyström approximation using simple uniform column sampling. With $\mathbf{B}\mathbf{B}^\top = \mathbf{K}_{mk}\mathbf{K}_{kk}^{-1}\mathbf{K}_{mk}^\top$, we have

$$\tilde{\tau}_i(\lambda) = \mathbf{b}_i^\top(\mathbf{B}^\top\mathbf{B} + \lambda\mathbb{I})^{-1}\mathbf{b}_i \tag{11}$$

as approximate measure for the importance of each column *i*. Here, $\mathbf{b}_i$ is the *i*-th row of the Cholesky factor $\mathbf{B}$. The inducing columns for our preconditioner are then sampled according to this leverage score distribution. It has been shown that this construction approximates $\mathbf{K}_\lambda$ with constant probability within a small relative error in comparison to the best rank-*k* approximation via eigenvalue decomposition (*84*, *85*) but at a significantly lower computational cost.

Numerical implementation

The performance and effectiveness of our solver crucially depend on a careful implementation, as it involves operations that are highly susceptible to numerical error. Here, we give an overview of the most important practical considerations in our reference implementation.

Memory requirement

At every iteration, the preconditioned CG algorithm computes two matrix products: $\mathbf{K}_\lambda \boldsymbol{\alpha}_t$ (see Eq. 3) and another one involving $\mathbf{P}^{-1}$. To avoid storing the $m \times m$ kernel matrix $\mathbf{K}_\lambda$, we use the NumPy LinearOperator interface to perform matrix-vector multiplications on the fly. This reduces the memory complexity of $\mathbf{K}\boldsymbol{\alpha}_t$ to $\mathcal{O}(m)$ as only the resulting vector needs to be stored. $\mathbf{P}^{-1}$ is relatively expensive to construct and therefore retained in memory, albeit in factorized form at a smaller memory cost of only $\mathcal{O}(mk)$.

Preconditioner construction

A straightforward evaluation of the Woodbury matrix identity in Eq. 8 is known to be numerically unstable (*59*). To construct $\mathbf{P}^{-1}$, we have to take a computational detour. First, the Cholesky decomposition of the symmetric positive definite submatrix $\mathbf{K}_{kk} = \mathbf{L}_{kk}\mathbf{L}_{kk}^\top$ (Eq. 7) is generated to define $\overline{\mathbf{K}}_{mk} = \mathbf{K}_{mk}\mathbf{L}_{kk}^{-\top}$ (*57*). Using another Cholesky step, $\lambda\mathbb{I} + \overline{\mathbf{K}}_{mk}^\top \overline{\mathbf{K}}_{mk} = \mathbf{D}_{kk}\mathbf{D}_{kk}^\top$, we then construct

$$\mathbf{P}^{-1} = \lambda^{-1}[\mathbb{I} - \overline{\mathbf{K}}_{mk}\mathbf{D}_{kk}^{-\top}\mathbf{D}_{kk}^{-1}\overline{\mathbf{K}}_{mk}^\top] \tag{12}$$

However, evaluating the product $\overline{\mathbf{K}}_{mk}^\top \overline{\mathbf{K}}_{mk}$ is not advisable, because it squares the condition number of the term, which can introduce rounding errors that diminish the effectiveness of the preconditioner. We therefore perform a numerically more robust indirect Cholesky decomposition of the inverse in Eq. 8 via thin QR factorization (*56*–*60*)

$$\begin{bmatrix} \overline{\mathbf{K}}_{mk} \\ \sqrt{\lambda}\mathbb{I} \end{bmatrix} = \mathbf{Q}\begin{bmatrix} \mathbf{D}_{kk}^\top \\ \mathbf{0} \end{bmatrix} \tag{13}$$

This decomposition can be computed efficiently without storing $\mathbf{Q}$, albeit at a slightly higher cost compared to a direct Cholesky decomposition. Throughout these calculations, the memory complexity remains constant at $\mathcal{O}(mk)$. $\mathbf{P}^{-1}$ is then applied to vectors via the NumPy LinearOperator interface, without expanding the product.

Initial guess $\boldsymbol{\alpha}_0$

We use $\alpha_0 = \mathbf{0}$ as initial guess, because a nonzero initialization can adversely affect the convergence of Krylov subspace methods [see chapter 5.8.15 of Ref (*86*)].

CG solver restarts

With perfect arithmetic, the CG algorithm guarantees monotonically improving approximations of α_t. However, the series of optimization steps may eventually become nonorthogonal after many iterations, due to the accumulation of rounding errors. This can slow down or even completely stall convergence in practice (*87*). We monitor the average progress of the solver toward minimizing the residual within a rolling time window and break the cycle by restarting the CG algorithm using the latest α_t as initial guess, if necessary. Slow progress toward the solution can also be a consequence of insufficient preconditioner performance. Our implementation dynamically increases the number of inducing columns to adjust the strength of the preconditioner before every restart.

Supplementary Materials

This PDF file includes:

Appendices A to D
Tables S1 to S3
Figs. S1 to S3
References

View/request a protocol for this paper from *Bio-protocol*.

REFERENCES AND NOTES

1. O. T. Unke, S. Chmiela, H. E. Sauceda, M. Gastegger, I. Poltavsky, K. T. Schütt, A. Tkatchenko, K.-R. Müller, Machine learning force fields. *Chem. Rev.* **121**, 10142–10186 (2021).
2. J. A. Keith, V. Vassilev-Galindo, B. Cheng, S. Chmiela, M. Gastegger, K.-R. Müller, A. Tkatchenko, Combining machine learning and computational chemistry for predictive insights into chemical systems. *Chem. Rev.* **121**, 9816–9872 (2021).
3. O. A. von Lilienfeld, K.-R. Müller, A. Tkatchenko, Exploring chemical compound space with quantum-based machine learning. *Nat. Rev. Chem.* **4**, 347–358 (2020).
4. C. Zhang, S. Bengio, M. Hardt, B. Recht, O. Vinyals, Understanding deep learning (still) requires rethinking generalization. *Commun. ACM* **64**, 107–115 (2021).
5. Z. Allen-Zhu, Y. Li, Y. Liang, Learning and generalization in overparameterized neural networks, going beyond two layers, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., 2019), pp. 6155–6166.
6. A. Bietti, J. Mairal, On the inductive bias of neural tangent kernels, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., 2019), vol. 32.
7. B. Chamberlain, J. Rowbottom, M. I. Gorinova, M. Bronstein, S. Webb, E. Rossi, GRAND: Graph neural diffusion, in *International Conference on Machine Learning* (PMLR, 2021), pp. 1407–1418.
8. T. W. Ko, J. A. Finkler, S. Goedecker, J. Behler, A fourth-generation high-dimensional neural network potential with accurate electrostatics including non-local charge transfer. *Nat. Commun.* **12**, 398 (2021).
9. O. T. Unke, S. Chmiela, M. Gastegger, K. T. Schütt, H. E. Sauceda, K.-R. Müller, SpookyNet: Learning force fields with electronic degrees of freedom and nonlocal effects. *Nat. Commun.* **12**, 7273 (2021).
10. A. Ambrosetti, N. Ferri, R. A. DiStasio Jr., A. Tkatchenko, Wavelike charge density fluctuations and van der Waals interactions at the nanoscale. *Science* **351**, 1171–1176 (2016).
11. M. Stöhr, A. Tkatchenko, Quantum mechanics of proteins in explicit water: The role of plasmon-like solute-solvent interactions. *Sci. Adv.* **5**, eaax0024 (2019).
12. O. T. Unke, M. Stöhr, S. Ganscha, T. Unterthiner, H. Maennel, S. Kashubin, D. Ahlin, M. Gastegger, L. M. Sandonas, A. Tkatchenko, K.-R. Müller, Accurate machine learned quantum-mechanical force fields for biomolecular simulations. arXiv:2205.08306 [physics.chem-ph] (17 May 2022).
13. P. Hauseux, T.-T. Nguyen, A. Ambrosetti, K. S. Ruiz, S. P. A. Bordas, A. Tkatchenko, From quantum to continuum mechanics in the delamination of atomically-thin layers from substrates. *Nat. Commun.* **11**, 1651 (2020).
14. P. Hauseux, A. Ambrosetti, S. P. Bordas, A. Tkatchenko, Colossal enhancement of atomic force response in van der Waals materials arising from many-body electronic correlations. *Phys. Rev. Lett.* **128**, 106101 (2022).
15. T. Bereau, R. A. DiStasio Jr., A. Tkatchenko, O. A. Von Lilienfeld, Non-covalent interactions across organic and biological subsets of chemical space: Physics-based potentials parametrized from machine learning. *J. Chem. Phys.* **148**, 241706 (2018).
16. K. Yao, J. E. Herr, D. W. Toth, R. Mckintyre, J. Parkhill, The TensorMol-0.1 model chemistry: A neural network augmented with long-range physics. *Chem. Sci.* **9**, 2261–2269 (2018).
17. A. Grisafi, M. Ceriotti, Incorporating long-range physics in atomic-scale machine learning. *J. Chem. Phys.* **151**, 204105 (2019).
18. O. T. Unke, M. Meuwly, PhysNet: A neural network for predicting energies, forces, dipole moments and partial charges. *J. Chem. Theory Comput.* **15**, 3678–3693 (2019).
19. S. P. Niblett, M. Galib, D. T. Limmer, Learning intermolecular forces at liquid–vapor interfaces. *J. Chem. Phys.* **155**, 164101 (2021).
20. A. Gao, R. C. Remsing, Self-consistent determination of long-range electrostatics in neural network potentials. *Nat. Commun.* **13**, 1572 (2022).
21. M. Rupp, A. Tkatchenko, K.-R. Müller, O. A. von Lilienfeld, Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.* **108**, 058301 (2012).
22. S. Chmiela, A. Tkatchenko, H. E. Sauceda, I. Poltavsky, K. T. Schütt, K.-R. Müller, Machine learning of accurate energy-conserving molecular force fields. *Sci. Adv.* **3**, e1603015 (2017).
23. S. Chmiela, H. E. Sauceda, K.-R. Müller, A. Tkatchenko, Towards exact molecular dynamics simulations with machine-learned force fields. *Nat. Commun.* **9**, 3887 (2018).

24. H. E. Sauceda, S. Chmiela, I. Poltavsky, K.-R. Müller, A. Tkatchenko, Molecular force fields with gradient-domain machine learning: Construction and application to dynamics of small molecules with coupled cluster forces. *J. Chem. Phys.* **150**, 114102 (2019).
25. H. E. Sauceda, M. Gastegger, S. Chmiela, K.-R. Müller, A. Tkatchenko, Molecular force fields with gradient-domain machine learning (GDML): Comparison and synergies with classical force fields. *J. Chem. Phys.* **153**, 124109 (2020).
26. A. S. Christensen, L. A. Bratholm, F. A. Faber, O. Anatole von Lilienfeld, FCHL revisited: Faster and more accurate quantum machine learning. *J. Chem. Phys.* **152**, 044107 (2020).
27. B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* **10**, 1299–1319 (1998).
28. M. L. Braun, J. M. Buhmann, K.-R. Müller, On relevant dimensions in kernel feature spaces. *J. Mach. Learn. Res.* **9**, 1875–1908 (2008).
29. H. E. Sauceda, L. E. Gálvez-González, S. Chmiela, L. O. Paz-Borbón, K.-R. Müller, A. Tkatchenko, BIGDML–towards accurate quantum machine learning force fields for materials. *Nat. Commun.* **13**, 3733 (2022).
30. S. Chmiela, H. E. Sauceda, A. Tkatchenko, K.-R. Müller, Accurate molecular dynamics enabled by efficient physically constrained machine learning approaches, in *Machine Learning Meets Quantum Physics* (Springer, 2020), pp. 129–154.
31. C. K. Williams, C. E. Rasmussen, *Gaussian Processes for Machine Learning* (MIT Press, 2006).
32. S. Fine, K. Scheinberg, Efficient SVM training using low-rank kernel representations. *J. Mach. Learn. Res.* **2**, 243–264 (2001).
33. F. R. Bach, M. I. Jordan, Kernel independent component analysis. *J. Mach. Learn. Res.* **3**, 1–48 (2002).
34. C. K. Williams, M. Seeger, Using the Nyström method to speed up kernel machines, in *Advances in Neural Information Processing Systems* (MIT Press, 2001), pp. 682–688.
35. J. Quiñonero-Candela, C. E. Rasmussen, A unifying view of sparse approximate Gaussian process regression. *J. Mach. Learn. Res.* **6**, 1939–1959 (2005).
36. C. Yang, R. Duraiswami, L. S. Davis, Efficient kernel machines using the improved fast Gauss transform, in *Advances in Neural Information Processing Systems* (MIT Press, 2005), pp. 1561–1568.
37. Y. Shen, M. Seeger, A. Y. Ng, Fast Gaussian process regression using KD-trees, in *Advances in Neural Information Processing Systems* (MIT Press, 2006), pp. 1225–1232.
38. E. Snelson, Z. Ghahramani, Sparse Gaussian processes using pseudo-inputs, in *Advances in Neural Information Processing Systems* (MIT Press, 2006), pp. 1257–1264.
39. A. Rahimi, B. Recht, Random features for large-scale kernel machines, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., 2008), pp. 1177–1184.
40. A. Wilson, H. Nickisch, Kernel interpolation for scalable structured Gaussian processes (KISS-GP), in *International Conference on Machine Learning* (jmlr.org, 2015), pp. 1775–1784.
41. A. Rudi, L. Carratino, L. Rosasco, Falkon: An optimal large scale kernel method, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., 2017), pp. 3888–3898.
42. F. R. Bach, M. I. Jordan, Predictive low-rank decomposition for kernel methods, in *International Conference on Machine Learning* (Association for Computing Machinery, 2005), pp. 33–40.
43. I. Murray, Gaussian processes and fast matrix-vector multiplies (Numerical Mathematics in Machine Learning Workshop-International Conference, 2009).
44. A. G. Wilson, Z. Hu, R. Salakhutdinov, E. P. Xing, Deep kernel learning, in *International Conference on Artificial Intelligence and Statistics* (PMLR, 2016), pp. 370–378.
45. J. Gardner, G. Pleiss, R. Wu, K. Weinberger, A. Wilson, Product kernel interpolation for scalable Gaussian processes, in *International Conference on Artificial Intelligence and Statistics* (PMLR, 2018), pp. 1407–1416.
46. J. Gardner, G. Pleiss, K. Q. Weinberger, D. Bindel, A. G. Wilson, GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., 2018), pp. 7576–7586.
47. K. Wang, G. Pleiss, J. Gardner, S. Tyree, K. Q. Weinberger, A. G. Wilson, Exact Gaussian processes on a million data points, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., 2019), pp. 14622–14632.
48. M. R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems. *J. Res. Natl. Bur. Stand.* **49**, 409–436 (1952).
49. Y. Saad, *Iterative Methods for Sparse Linear Systems* (SIAM, 2003), vol. 82.
50. J. R. Shewchuk, An introduction to the conjugate gradient method without the agonizing pain (Carnegie Mellon University, 1994).
51. V. A. M. L. A. Pastur, Distribution of eigenvalues for some sets of random matrices. *Math. USSR Sb.* **1**, 457 (1967).
52. H. Wendland, *Scattered Data Approximation* (Cambridge Monographs on Applied and Computational Mathematics, Cambridge Univ. Press, 2010).
53. A. Wu, M. C. Aoi, J. W. Pillow, Exploiting gradients and Hessians in Bayesian optimization and Bayesian quadrature. arXiv:1704.00060 [stat.ML] (31 March 2017).
54. K. Cutajar, M. Osborne, J. Cunningham, M. Filippone, Preconditioning kernel matrices, in *International Conference on Machine Learning* (jmlr.org, 2016), pp. 2529–2538.
55. S. Kumar, M. Mohri, A. Talwalkar, Sampling methods for the Nyström method. *J. Mach. Learn. Res.* **13**, 981–1006 (2012).
56. G. H. Golub, C. F. Van Loan, *Matrix Computations* (JHU Press, ed. 4, 2013).
57. G. Wahba, *Spline Models for Observational Data* (SIAM, 1990).
58. M. W. Seeger, C. K. Williams, N. D. Lawrence, Fast forward selection to speed up sparse Gaussian process regression, in *International Workshop on Artificial Intelligence and Statistics* (PMLR, 2003), pp. 254–261.
59. L. Foster, A. Waagen, N. Aijaz, M. Hurley, A. Luis, J. Rinsky, C. Satyavolu, M. J. Way, P. Gazis, A. Srivastava, Stable and efficient Gaussian process calculations. *J. Mach. Learn. Res.* **10**, 857–882 (2009).
60. D. Eriksson, K. Dong, E. Lee, D. Bindel, A. G. Wilson, Scaling Gaussian process regression with derivatives, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., 2018), vol. 31.
61. J. P. Perdew, K. Burke, M. Ernzerhof, Generalized gradient approximation made simple. *Phys. Rev. Lett.* **77**, 3865–3868 (1996).
62. A. Tkatchenko, R. A. DiStasio Jr., R. Car, M. Scheffler, Accurate and efficient method for many-body van der Waals interactions. *Phys. Rev. Lett.* **108**, 236402 (2012).
63. J. Taylor, *Introduction to Error Analysis, The Study of Uncertainties in Physical Measurements* (University Science Books, NY, 1997).
64. X. Fu, Z. Wu, W. Wang, T. Xie, S. Keten, R. Gomez-Bombarelli, and T. Jaakkola, Forces are not enough: Benchmark and critical evaluation for machine learning force fields with molecular simulations. arXiv:2210.07237 [physics.comp-ph] (13 October 2022).
65. A. M. Reilly, A. Tkatchenko, Role of dispersion interactions in the polymorphism and entropic stabilization of the aspirin crystal. *Phys. Rev. Lett.* **113**, 055701 (2014).
66. J. P. Hare, T. J. Dennis, H. W. Kroto, R. Taylor, A. W. Allaf, S. Balm, D. R. Walton, The IR spectra of fullerene-60 and -70. *J. Chem. Soc. Chem. Commun.* , 412–413 (1991).
67. C. H. Choi, M. Kertesz, L. Mihaly, Vibrational assignment of all 46 fundamentals of C_{60} and $C_{60}{}^{6-}$: Scaled quantum mechanical results performed in redundant internal coordinates and compared to experiments. *J. Phys. Chem. A* **104**, 102–112 (2000).
68. U. Erlekam, M. Frankowski, G. Meijer, G. von Helden, An experimental value for the B1u C–H stretch mode in benzene. *J. Chem. Phys.* **124**, 171101 (2006).
69. F. Brockherde, L. Vogt, L. Li, M. E. Tuckerman, K. Burke, K.-R. Müller, Bypassing the Kohn-Sham equations with machine learning. *Nat. Commun.* **8**, 872 (2017).
70. K. T. Schütt, F. Arbabzadah, S. Chmiela, K.-R. Müller, A. Tkatchenko, Quantum-chemical insights from deep tensor neural networks. *Nat. Commun.* **8**, 13890 (2017).
71. K. T. Schütt, M. Gastegger, A. Tkatchenko, K.-R. Müller, R. J. Maurer, Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions. *Nat. Commun.* **10**, 5024 (2019).
72. M. Gastegger, A. McSloy, M. Luya, K. T. Schütt, R. J. Maurer, A deep neural network for molecular wave functions in quasi-atomic minimal basis representation. *J. Chem. Phys.* **153**, 044123 (2020).
73. J. Hoja, L. Medrano Sandonas, B. G. Ernst, A. Vazquez-Mayagoitia, R. A. DiStasio Jr., A. Tkatchenko, QM7-X, a comprehensive dataset of quantum-mechanical properties spanning the chemical space of small organic molecules. *Sci. Dat.* **8**, 43 (2021).
74. J. T. Frank, S. Chmiela, Detect the interactions that matter in matter: Geometric attention for many-body systems. arXiv:2106.02549 [cs.LG] (6 September 2021).
75. J. T. Frank, O. T. Unke, K.-R. Müller, So3krates –self-attention for higher-order geometric interactions on arbitrary length-scales. arXiv:2205.14276 [cs.LG] (28 May 2022).
76. N. F. Schmitz, K.-R. Müller, S. Chmiela, Algorithmic differentiation for automated modeling of machine learned force fields. *J. Phys. Chem. Lett.* **13**, 10183–10189 (2022).
77. A. Kabylda, V. Vassilev-Galindo, S. Chmiela, I. Poltavsky, A. Tkatchenko, Towards linearly scaling and chemically accurate global machine learning force fields for large molecules. arXiv:2209.03985 [physics.chem-ph] (8 September 2022).
78. B. Huang, O. A. von Lilienfeld, Quantum machine learning using atom-in-molecule-based fragments selected on the fly. *Nat. Chem.* **12**, 945–951 (2020).
79. J. S. Smith, O. Isayev, A. E. Roitberg, ANI-1: An extensible neural network potential with DFT accuracy at force field computational cost. *Chem. Sci.* **8**, 3192–3203 (2017).
80. C. Bannwarth, S. Ehlert, S. Grimme, GFN2-xTB – An accurate and broadly parametrized self-consistent tight-binding quantum chemical method with multipole electrostatics and density-dependent dispersion contributions. *J. Chem. Theory Comput.* **15**, 1652–1671 (2019).
81. A. Alaoui, M. W. Mahoney, Fast randomized kernel ridge regression with statistical guarantees, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., 2015), pp. 775–783.
82. A. Rudi, D. Calandriello, L. Carratino, L. Rosasco, On fast leverage score sampling and optimal learning, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., 2018), Vol. 31.

83. M. B. Cohen, Y. T. Lee, C. Musco, C. Musco, R. Peng, A. Sidford, Uniform sampling for matrix approximation, in *Conference on Innovations in Theoretical Computer Science (ITIC)* (Association for Computing Machinery, 2015), pp. 181–190.
84. P. Drineas, M. W. Mahoney, S. Muthukrishnan, Relative-error cur matrix decompositions. *SIAM J. Matrix Anal. Appl.* **30**, 844–881 (2008).
85. S. Kumar, M. Mohri, A. Talwalkar, Sampling techniques for the Nyström method, in *International Conference on Artificial Intelligence and Statistics* (PMLR, 2009) pp. 304–311.
86. J. Liesen, Z. Strakoš, *Krylov Subspace Methods: Principles and Analysis* (Oxford Univ. Press, 2013).
87. M. J. D. Powell, Restart procedures for the conjugate gradient method. *Math. Program.* **12**, 241–254 (1977).
88. S. Chmiela, "Towards exact molecular dynamics simulations with invariant machine-learned models," thesis, Technische Universität Berlin (2019).
89. V. Blum, R. Gehrke, F. Hanke, P. Havu, V. Havu, X. Ren, K. Reuter, M. Scheffler, Ab initio molecular simulations with numeric atom-centered orbitals. *Comput. Phys. Commun.* **180**, 2175–2196 (2009).
90. V. Kapil, M. Rossi, O. Marsalek, R. Petraglia, Y. Litman, T. Spura, B. Cheng, A. Cuzzocrea, R. H. Meißner, D. M. Wilkins, B. A. Helfrecht, P. Juda, S. P. Bienvenue, W. Fang, J. Kessler, I. Poltavsky, S. Vandenbrande, J. Wieme, C. Corminboeuf, T. D. Kühne, D. E. Manolopoulos, T. E. Markland, J. O. Richardson, A. Tkatchenko, G. A. Tribello, V. van Speybroeck, M. Ceriotti, i-PI 2.0: A universal force engine for advanced molecular simulations. *Comput. Phys. Commun.* **236**, 214–223 (2019).
91. S. Batzner, A. Musaelian, L. Sun, M. Geiger, J. P. Mailoa, M. Kornbluth, N. Molinari, T. E. Smidt, B. Kozinsky, E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nat. Commun.* **13**, 2453 (2022).
92. K. Schütt, O. Unke, M. Gastegger, Equivariant message passing for the prediction of tensorial properties and molecular spectra, in *International Conference on Machine Learning* (PMLR, 2021), pp. 9377–9388.

Acknowledgments

Funding: S.C. and K.-R.M. acknowledge support by the German Federal Ministry of Education and Research (BMBF) for BIFOLD (01IS18037A). K.-R.M. was partly supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grants funded by the Korea government(MSIT) (no. 2019-0-00079, Artificial Intelligence Graduate School Program, Korea University and no. 2022-0-00984, Development of Artificial Intelligence Technology for Personalized Plug-and-Play Explanation and Verification of Explanation) and by the German Federal Ministry for Education and Research (BMBF) under grants 01IS14013B-E and 01GQ1115. A.T., V.V.-G., and A.K. acknowledge support from the European Research Council (ERC) Consolidator Grant BeStMo and the Luxembourg National Research Fund (FNR) (grant C19/MS/13718694/QML-FLEX and AFR PhD grant 15720828). H.E.S. acknowledges support from DGTIC-UNAM under Project LANCAD-UNAM-DGTIC-419. Furthermore, we thank IPAM for hospitality and inspiration while finishing this manuscript. **Author contributions:** S.C. conceived the presented method and developed the computational framework. A.K., V.V.-G., and O.T.U. performed the reference data calculations. V.V.-G., O.T.U., A.K., H.E.S., and S.C. carried out and analyzed the MD simulations. S.C. and V.V.-G. created the figures, with help from other authors. K.-R.M. and A.T. were involved in conceiving, planning, and supervising the work. All authors wrote the paper, discussed the results, and commented on the manuscript. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper and/or the Supplementary Materials. Datasets, pretrained FF models, and software are available at www.sgdml.org.

Submitted 29 September 2022
Accepted 28 November 2022
Published 11 January 2023
10.1126/sciadv.adf0873