

# **Towards Exact Molecular Dynamics Simulations with Invariant Machine-Learned Models**

vorgelegt von

M. Sc.

Stefan Chmiela

von der Fakultät IV - Elektrotechnik und Informatik  
der Technischen Universität Berlin  
zur Erlangung des akademischen Grades

Doktor der Naturwissenschaften  
– Dr. rer. nat. –

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Manfred Opper  
Gutachter: Prof. Dr. Klaus-Robert Müller  
Gutachter: Prof. Dr. Alexandre Tkatchenko  
Gutachter: Prof. Dr. Frank Noé

Tag der wissenschaftlichen Aussprache: 28. Juni 2019

Berlin 2019



## Acknowledgements

Many people have contributed to this thesis in one way or another - whether through fruitful collaborations, by teaching me something new, as a source of motivation or by simply providing pleasant company. These people have made my time at the lab not only enriching, but also an enjoyable experience.

I want to express my deepest gratitude to Prof. Dr. Klaus-Robert Müller for his invaluable guidance, encouragement and exceptional support as my advisor over the years. In countless discussions he provided critical advice and valuable insight, while allowing much freedom in my research. Due to the interdisciplinary nature of my work, I was also fortunate to work closely with Prof. Dr. Alexandre Tkatchenko as my second advisor. With his unbounded curiosity and infectious ambition he broadened my perspective on what it means to be a scientist. Klaus and Alex always did their best to put me on the right track. Both have invested time and energy beyond what I could have reasonably expected from a professional relationship.

This work would have not been possible without the collaboration of my dear colleagues. I took particular pleasure in working closely with Huziel E. Sauceda, who I became good friends with far beyond our collaboration. He always patiently shared his knowledge and his constructive comments were instrumental for my thesis. A special thanks also goes to my other co-authors Kristof Schütt and Igor Poltavsky. I was fortunate to interact with many more wonderful people over the years, in particular my office mates: Binh Thanh Bui, Mihail Bogojeski, Felix Brockherde, Duncan Blythe and Alexander Binder. Furthermore, I would like to thank Grégoire Montavon, Wiktor Pronobis, Danny Panknin, Michael Gastegger, Alexander Bauer, Marina Vidovic, Nico Görnitz, Sergej Dogadov, Stephanie Brandl, Maximilian Alber, Miriam Hägele and Andreas Ziehe for creating a pleasurable atmosphere in the lab. I would also like to thank Andrea Gerdes, Imke Weitkamp, and Dominik Kühne for allowing me to focus on research, undisturbed from administrative chores.

Finally, I am deeply grateful to my parents and my siblings for their ongoing encouragement to chase my dreams.



## Abstract

Molecular dynamics (MD) simulations constitute the cornerstone of contemporary atomistic modeling in chemistry, biology, and materials science. However, one of the widely recognized and increasingly pressing issues in MD simulations is the lack of accuracy of underlying classical interatomic potentials, which hinders truly predictive modeling of dynamics and function of (bio)molecular systems. Classical potentials often fail to faithfully capture key quantum effects in molecules and materials. In this thesis, we develop a combined machine learning (ML) and quantum mechanics approach that enables the direct reconstruction of flexible molecular force fields from high-level *ab initio* calculations.

We approach this challenge by incorporating fundamental physical symmetries and conservation laws into ML techniques. Using conservation of energy – a fundamental property of closed classical and quantum mechanical systems – we derive an efficient gradient-domain machine learning (GDML) model. The challenge of constructing conservative force fields is accomplished by learning in a Hilbert space of vector-valued functions that obey the law of energy conservation.

We proceed with the development of a multi-partite matching algorithm that enables a fully automated recovery of physically relevant point-group and fluxional symmetries from the training dataset into a symmetric variant of our model. The developed symmetric GDML (sGDML) approach faithfully reproduces global force fields at quantum-chemical CCSD(T) level of accuracy and allows converged MD simulations with fully quantized electrons and nuclei.

We present MD simulations, for flexible molecules with up to a few dozen atoms and provide insights into the dynamical behavior of these molecules. Our approach provides the key missing ingredient for achieving spectroscopic accuracy in molecular simulations.



## Zusammenfassung

Molekulardynamik (MD) -Simulationen bilden den Eckpfeiler der heutigen atomistischen Modellierung in Chemie, Biologie und den Materialwissenschaften. Ein allgemein anerkanntes und immer dringlicheres Problem ist jedoch die mangelnde Genauigkeit der zugrunde liegenden klassischen interatomaren Potentiale. Diese verhindern eine wirklich prädiktive Modellierung der Dynamik und Funktion von (bio-)molekularen Systemen. Klassische Potentiale erfassen wichtige Quanteneffekte in Molekülen und Materialien oft nicht genau genug. In dieser Arbeit entwickeln wir einen kombinierten Ansatz aus maschinellem Lernen (ML) und Quantenmechanik, der die direkte Rekonstruktion flexibler molekularer Kraftfelder aus hochgenauen *Ab-initio*-Berechnungen ermöglicht.

Wir begegnen dieser Herausforderung, indem wir grundlegende physikalische Symmetrien und Erhaltungssätze in ML-Techniken integrieren. Unter Verwendung von Energieerhaltung - einer grundlegenden Eigenschaft geschlossener klassischer und quantenmechanischer Systeme - leiten wir ein effizientes Gradient-Domain-Machine-Learning-Modell (GDML) ab. Die Herausforderung, konservative Kraftfelder zu konstruieren, wird durch das Lernen in einem Hilbert-Raum vektorwertiger Funktionen gelöst, die dem Gesetz der Energieerhaltung folgen.

Wir fahren mit der Entwicklung eines multi-partiten Matching-Algorithmus fort, der eine vollautomatische Erkennung physikalisch relevanter Punktgruppen- und dynamischen Symmetrien aus dem Trainingsdatensatz erkennt und deren Integration in eine symmetrische Variante unseres Modells ermöglicht. Der entwickelte symmetrische GDML-Ansatz (sGDML) bildet globale Kraftfelder auf dem Niveau quantenchemischer CCSD(T)-Berechnungen genau ab und ermöglicht konvergierte MD-Simulationen mit vollständig quantisierten Elektronen und Atomkernen.

Wir präsentieren MD-Simulationen für flexible Moleküle mit bis zu ein paar Dutzend Atomen und geben Einblicke in das dynamische Verhalten dieser Moleküle. Unser Ansatz liefert den fehlenden Schlüsselbestandteil für die Erzielung spektroskopischer Genauigkeit in molekularen Simulationen.



# Contents

<b>List of Figures</b>	<b>xiii</b>
<b>List of Tables</b>	<b>xxi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Theoretical background . . . . .	3
1.1.1 Ab initio quantum chemistry . . . . .	3
1.1.2 Electron correlation . . . . .	5
1.1.3 Density Functional Theory . . . . .	7
1.1.4 Molecular dynamics . . . . .	9
1.1.5 Conservation laws . . . . .	11
1.2 Description of chapters . . . . .	11
1.3 Previously published work . . . . .	13
<b>2 Hilbert space learning</b>	<b>15</b>
2.1 Hilbert spaces . . . . .	16
2.1.1 Reproducing kernels . . . . .	16
2.1.2 Representer theorem . . . . .	16
2.2 Gaussian process models . . . . .	17
2.2.1 Gaussian process regression . . . . .	19
2.3 Encoding prior information . . . . .	19
2.3.1 Observations . . . . .	20
2.3.2 Covariance function . . . . .	21
2.3.3 Mean function . . . . .	24
2.4 Summary . . . . .	24
<b>3 Energy-conserving molecular force fields</b>	<b>27</b>
3.1 Local linearizations of the PES . . . . .	28
3.1.1 Hellman-Feynman theorem . . . . .	28

3.1.2	Noise amplification by differentiation . . . . .	29
3.2	Gradient domain machine learning (GDML) . . . . .	32
3.2.1	Multiple output GPs . . . . .	32
3.2.2	Conservative vector-valued GPs . . . . .	34
3.2.3	Force field covariance function . . . . .	38
3.3	Numerical experiments . . . . .	41
3.3.1	Datasets . . . . .	42
3.3.2	Baseline tests . . . . .	42
3.3.3	Driving MD simulations with GDML . . . . .	45
3.4	Practical considerations . . . . .	46
3.4.1	Explicit treatment of N-body correlations . . . . .	46
3.4.2	Numerical stability . . . . .	47
3.5	Software implementation . . . . .	49
3.5.1	Program overview . . . . .	49
3.6	Summary . . . . .	50
<b>4</b>	<b>Point groups and fluxional symmetries</b>	<b>53</b>
4.1	Positive-semidefinite assignment . . . . .	55
4.1.1	Solving the multi-way matching problem . . . . .	55
4.1.2	Symmetric kernels . . . . .	57
4.2	Symmetric gradient domain learning (sGDML) . . . . .	58
4.2.1	Training . . . . .	59
4.2.2	Descriptors . . . . .	60
4.3	Numerical experiments . . . . .	60
4.3.1	Datasets . . . . .	61
4.3.2	Forces and energies from GDML to sGDML@DFT to sGDML@CCSD(T) . . . . .	62
4.3.3	Molecular dynamics with <i>ab initio</i> accuracy . . . . .	64
4.3.4	CCSD(T)-level vibrational spectra . . . . .	66
4.3.5	Probability distributions CCSD(T) vs. DFT . . . . .	67
4.3.6	Symmetry compression . . . . .	68
4.4	Discussion . . . . .	72
4.5	Practical considerations . . . . .	75
4.5.1	Hybrid loss functions . . . . .	75
4.5.2	Imposing permutational symmetry . . . . .	77
4.5.3	Degenerate eigenvalues and the bi-partite matching algorithm . . . . .	79
4.6	Software implementation . . . . .	80

4.7	Summary . . . . .	80
<b>5</b>	<b>Conclusion</b>	<b>83</b>
5.1	Outlook . . . . .	84
	<b>Bibliography</b>	<b>87</b>
	<b>Appendix A Derivations</b>	<b>99</b>
A.1	Derivative observations . . . . .	99
A.1.1	Matérn covariance derivatives . . . . .	99
A.2	GDML model derivation . . . . .	101
A.2.1	Integration constant . . . . .	103
A.2.2	Bi-partite matching cost matrix . . . . .	103
A.2.3	Permutation matrices notation . . . . .	103
	<b>Appendix B Numerical results</b>	<b>105</b>
B.0.1	Energy-trained baseline model . . . . .	105
B.0.2	Non-conservative baseline model . . . . .	106
B.0.3	Probability distributions of the dihedral angles in ethanol (sGDML@CCSD(T) versus sGDML@DFT) . . . . .	107
	<b>Appendix C Software implementation</b>	<b>109</b>
C.0.1	User Input . . . . .	109
C.1	Usage . . . . .	111
C.1.1	Training . . . . .	112
C.1.2	Inference . . . . .	113
C.2	Example Application: Paracetamol . . . . .	114



# List of Figures

1.1	The path-integral molecular dynamics method approximates nuclear quantum effects by exploiting an isomorphism between a $P$ particle classical polymer and a quantum system. The equilibrium averages of this polymer approximate the properties of the quantum particle. This method is exact in the limit of the number of copies $P \rightarrow \infty$ . . . . .	10
2.1	Example functions drawn from GP priors based on different types of covariance functions. The squared exponential kernel defines a smooth, infinitely differentiable space of solutions (left), whereas the exponential kernel gives rise to non-differentiable functions (right). A well-defined hypothesis space can drastically simplify the learning problem. . . . .	17
3.1	A noisy approximation of a sine wave (blue). Although all instantaneous values are represented well, the derivative of the approximation is a poor estimator for the true derivative. This is because differentiation amplifies the high frequency noise component within the approximation (middle). Integration on the other hand acts as a low-pass filter (right) that attenuates noise. It is therefore easier to approximate a function with accurate first derivatives from derivative examples instead of function values. Note that integrals are only defined up to an additive constant, which needs to be recovered separately. . . . .	30

3.2	The construction of ML models: First, reference data from an MD trajectory are sampled. (a) The geometry of each molecule is encoded in a descriptor. This representation introduces elementary transformational invariances of energy and constitutes the first part of the prior. A kernel function then relates all descriptors to form the kernel matrix – the second part of the prior. The kernel function encodes similarity between data points. Our particular choice makes only weak assumptions: It limits the frequency spectrum of the resulting model and adds the energy conservation constraint. Hess, Hessian. (c) These general priors are sufficient to reproduce good estimates from a restricted number of force samples. (b) A comparable energy model is not able to reproduce the PES to the same level of detail. . . . .	31
3.3	Differentiation of a PES estimator (blue) versus direct force field reconstruction (red). The law of energy conservation is trivially obeyed in the first case, but requires explicit <i>a priori</i> constraints in the latter scenario. Both approaches yield estimates for energy and forces, but a direct reconstruction of the force fields avoids the amplification of estimation errors due to the derivative operator. The challenge in estimating force fields directly lies in the complexity arising from their high $3N$ -dimensionality. . . . .	33
3.4	Modeling gradient fields (leftmost subfigure) based on a small number of examples. With GDML, a conservative vector field estimate $\hat{\mathbf{f}}$ is obtained directly (purple). In contrast, a naïve estimator $\hat{\mathbf{f}}^-$ with no information about the correlation structure of its outputs is not capable to uphold the energy conservation constraint (blue). We perform a Helmholtz decomposition of the naïve non-conservative vector field estimate to show the error component due violation of the law of energy conservation (red). This significant contribution to the overall prediction error is completely avoided with the GDML approach. . . . .	34
3.5	The blue line highlights the subset of parameter space for $\alpha_2$ and $\beta_1$ that yields conservative vector field estimates from the model in Eq. 3.12. The curl of the predicted vector field vanishes, i.e. $\nabla \times \hat{\mathbf{f}}_F = \mathbf{0}$ only when $\beta_1 = \alpha_2$ . This is not the case for any of the off-diagonal parameter configurations. In the shown example, the the configuration $(1, -1)$ has a constant curl of $(0, 0, 2)^\top$ in the direction orthogonal to the $\alpha_2$ - $\beta_1$ -plane. . . . .	36

---

3.6 Predicted energies (a) and forces (b) for 500 consecutive time steps along a MD trajectory of uracil at 500 K. The highly accurate GDML predictions (gray) follow the reference trajectory (black, dashed) closely. To highlight small deviations, the area between both curves is marked red. . . . .	41
3.7 Efficiency of GDML predictor versus a model that has been trained on energies. (a) Required number of samples for a force prediction performance of MAE ( $1 \text{ kcal mol}^{-1} \text{ \AA}^{-1}$ ) with the energy-based model (gray) and GDML (blue). The energy-based model was not able to achieve the targeted performance with the maximum number of 63,000 samples for aspirin. (b) Force prediction errors for the converged models (same number of partial derivative samples and energy samples). (c) Energy prediction errors for the converged models. All reported prediction errors have been estimated via cross-validation. . . . .	43
3.8 Results of classical and PIMD simulations. The recently developed estimators based on perturbation theory were used to evaluate structural and electronic observables [1]. (a) Comparison of the interatomic distance distributions, $h(r) = \left\langle \frac{2}{N(N-1)} \sum_{i < j}^N \delta(r - \  \mathbf{r}_i - \mathbf{r}_j \ ) \right\rangle_{P,t}$ , obtained from GDML (blue line) and DFT (dashed red line) with classical MD (main frame), and PIMD (inset). a.u., arbitrary units. (b) Probability distribution of the dihedral angles (corresponding to carboxylic acid and ester functional groups) using a 20 ps time interval from a total PIMD trajectory of 200 ps. . . . .	45
3.9 The Pearson correlation coefficient $\rho_{k_0, k_d} = \text{cov}(k_0, k_d) / (\sigma_{k_0} \sigma_{k_d})$ of a pair of covariance functions in dependence of their spatial separation $d$ . Here, $\sigma_{k_0}$ and $\sigma_{k_d}$ are the standard deviations of both covariance functions. We compare the Matérn covariances $k_0 = C_{\nu=n+\frac{1}{2}}(\ x\ )$ and $k_d = C_{\nu=n+\frac{1}{2}}(\ x - d\ )$ for $n = 2$ (red) and its second derivatives, as used in the GDML approach (blue). The correlation for small distances drops off quickly using the gradient domain covariance function, which improves the numerical stability of the GP. . . . .	48



- 4.5 Molecular dynamics simulations for ethanol. (A) Potential energy profile of the dihedral angle describing the rotation of the hydroxyl group for CCSD(T) (red) vs. DFT (blue). The energetic barriers predicted by sGDML@CCSD(T) are:  $M_t \rightarrow M_g$ : 1.18 kcal mol<sup>-1</sup>,  $M_{g-} \rightarrow M_{g+}$ : 1.19 kcal mol<sup>-1</sup>, and  $M_g \rightarrow M_t$ : 1.07 kcal mol<sup>-1</sup>. The dashed lines show the probability distributions obtained from PIMD at 300K. (B) Joint probability distribution function for the two dihedral angles of the methyl and hydroxyl functional groups. Each minimum is annotated with the occupation probability obtained from classical and path-integral MD in comparison with experimental values. (C) Analysis of vibrational spectra (velocity–velocity autocorrelation function). (top) Comparison between the vibrational spectrum obtained from PIMD simulations at 300K for sGDML@CCSD(T) and its sGDML@DFT counterpart; (middle) comparison between the sGDML@CCSD(T) PIMD spectrum and the harmonic approximation based on CCSD(T) frequencies; (bottom) comparison of sGDML@CCSD(T) PIMD spectra at 300K and 100K. The right-most panel shows several characteristic normal modes of ethanol, where atomic displacements are illustrated by green arrows. . . . .

4.6 Analysis of MD simulations with sGDML for malonaldehyde and aspirin. The MD simulations at 300 K were carried out for 500 ps. (A) Joint probability distributions of the dihedral angles in malonaldehyde, describing the rotation of both aldehyde groups based on classical MD simulations for sGDML@CCSD(T) and sGDML@DFT. The configurations (1) and (2) are representative structures of the most sampled regions of the PES. (B) Joint probability distributions of the dihedral angles in aspirin, describing the rotation of the ester and carboxylic acid groups based on PIMD simulations for sGDML@CCSD and sGDML@DFT using 16 beads at 300 K. The potential energy profile for the ester angle in kcal mol<sup>-1</sup> is shown for sGDML@CCSD (red), sGDML@DFT (blue) and compared with the CCSD reference (black, dashed). Contour lines show the differences of both distributions on a log scale. Both panels also show a comparison of the vibrational spectra generated via the velocity-velocity autocorrelation function obtained with sGDML@CCSD(T)/CCSD (red) and sGDML@DFT (blue). . . . .

4.7 Accuracy of the sGDML model in comparison to a traditional force field. We contrast the dihedral angle probability distributions of ethanol, malonaldehyde and aspirin obtained from classical MD simulations at 300 K with sGDML (left column) versus the AMBER [3, 4] (right column) force field. The ethanol simulations were carried out at constant energy (NVE), whereas a constant temperature (NVT) was used for malonaldehyde and aspirin. (A) Ethanol: the coupling between the hydroxyl and methyl rotor is absent in AMBER. Moreover, the probability distribution shows an unphysical harmonic sampling at room temperature, revealing the oversimplified harmonic description of bonded interactions in that force field. (B, C) Malonaldehyde and aspirin: the formulation of the AMBER force field is dominated by Coulomb interactions, which can lead an incomplete description of the PES and even spurious global minima in the case of aspirin. The length of the simulations was 0.5 ns. . . . .	73
4.8 Eigenspectra of the adjacency matrices of two highly symmetric molecules: benzene with symmetries in two dimensions (left) and the $C_{20}$ fullerene with symmetries in three dimensions (right). Benzene has 12 point group symmetries and eight out of its 12 eigenvalues are degenerate (shown in red). $C_{20}$ has 120 symmetries, with 15 degenerate eigenvalues out of 20. An unambiguous assignment of eigenvectors between several near-isomorphic instances of these structures (close to equilibrium) is therefore impossible. Our proposed multi-partite matching algorithm resolves the inconsistencies across multiple bi-partite assignments in the training set that arise from this ambiguity and other factors. . . . .	79
B.1 Comparison of probability distributions of the dihedral angles (methyl rotor vs. hydroxyl rotor) of ethanol obtained from classical and path-integral MD simulations at 300 K. We contrast the results from a sGDML model trained on CCSD(T) versus DFT reference calculations. The inclusion of nuclear quantum effects improves the sampling of the PES for both levels of theory. The sampling was performed during 0.5 ns of simulation, using 16 beads for PIMD. . . . .	107

C.1 Top: From a provided dataset of molecular geometries with corresponding energy and force labels, our sGMDL implementation creates a fully cross-validated FF model. Bottom: This lightweight model can then be used to speed up various PES sampling intensive applications, like molecular dynamics or the computation of transition paths. Interfacing ASE allows for easy computation of normal modes, vibrational spectra or nudged elastic band optimizations (middle row). Our interface to i-PI enables path integral molecular dynamics simulations (PIMD), which we use to compute the free energies and interatomic distance distributions $h(\mathbf{r})$ with classical MD and PIMD (bottom row). . . . .	118
---	-----



# List of Tables

3.1	GDML prediction accuracy for interatomic forces and total energies for all datasets. Energy errors are in $\text{kcal mol}^{-1}$ , force errors in $\text{kcal mol}^{-1} \text{\AA}^{-1}$ . Each model is trained on 1000 geometries with corresponding force labels. . .	44
4.1	Recovering the permutation-inversion (PI) group of symmetry operations of fluxional molecules from short MD trajectories. We used our multi-partite matching algorithm to recover the symmetries of the molecules used in Longuet-Higgins [5]. Our algorithm identifies PI group symmetries (a superset that also includes the PG), as well as additional symmetries that are an artifact of the metric used to compare molecular graphs in our matching algorithm. Each dataset consists of a MD trajectory of 5000 time steps. . . .	57
4.2	Relative increase in accuracy of the sGDML@DFT vs. the non-symmetric GDML model: the benefit of a symmetric model is directly linked to the number of permutational symmetries in the system. All symmetry counts include the identity transformation. . . . .	59
4.3	Prediction accuracy for interatomic forces and total energies of the sGDML@DFT on all datasets. Energy errors are in $\text{kcal mol}^{-1}$ , force errors in $\text{kcal mol}^{-1} \text{\AA}^{-1}$ . Each model is trained on 1000 geometries with corresponding force labels. . .	62
4.4	Prediction accuracy for interatomic forces and total energies of the sGDML@CCSD(T) model on all datasets. Energy errors are in $\text{kcal mol}^{-1}$ , force errors in $\text{kcal mol}^{-1} \text{\AA}^{-1}$ . . . . .	64
4.5	Prediction accuracy for interatomic forces and total energies using the original sGDML model and a compressed variant $\text{sGDML}_{\downarrow N}$ that only considers the non-symmetric atomic degrees of freedom $\downarrow N$ . Both model types have been trained on 1000 data points. The best result for each dataset is highlighted by bold face. . . . .	70

4.6 Prediction accuracy for interatomic forces and total energies using the original sGDML model with a training set size of $M = 1000$ and the compressed variant $\text{sGDML}_{\downarrow N}$ with increased training set size $\tilde{M}$ to match the complexity of the optimization problem during training. The best result for each dataset is highlighted by bold face.	72
4.7 Prediction accuracy for interatomic forces and total energies using the original sGDML model and a variant sGDML+E that has been extended with additional energy constraints in the loss function. Both model types have been trained on 1000 data points. The sGDML+E model consistently overfits the energy constraints at the cost of force prediction accuracy. The best result for each dataset is highlighted by bold face.	75
B.1 Accuracy of the converged energy-based predictor. All training set sizes $M$ are chosen to match the complexity of the optimization problem in the corresponding force model (number of samples times number of partial derivatives). Energy errors are in $\text{kcal mol}^{-1}$ , force errors in $\text{kcal mol}^{-1} \text{\AA}^{-1}$ .	105
B.2 Accuracy of the naïve force predictor based on a training set size of $M = 1000$ . This model learns all output components independently, without constraining the predicted forces to be energy conserving. It is identical to the GDML model in all other aspects. Energy prediction errors are not available, because the resulting force fields are not integrable.	106
B.3 Properties of MD datasets that were used for numerical testing.	108
C.1 Training times for various sGDML models based on 1000 reference data using an analytic solver on a Intel Xeon E5-2640 CPU at 2.40GHz. For the same models we also list the force and energy prediction performances for sequential evaluations of individual geometries and batch evaluations of 1000 geometries on a 2.8 GHz Intel Core i7 notebook.	110

# Chapter 1

## Introduction

Molecular dynamics (MD) simulations are the cornerstone of contemporary atomistic modeling in chemistry, biology, and materials science. They reveal the equilibrium thermodynamic and dynamical properties of a system at finite temperature, while simultaneously providing insight into its motion at atomic scale [6]. The predictive power of these simulations is however only as good as the underlying description of the interatomic forces. Most commonly, the forces are obtained from classical potentials, which provide a mechanistic description in terms of fixed interaction patterns between bonds and bond angles within a molecule. What makes these so-called classical force fields (FF) appealing is that they can be fitted empirically to experimental or *ab initio* data and evaluated very efficiently, due to their low number of parameters. However, these advantages come at the severe cost of accuracy: their rigid functional form prohibits capture of a wide range of important effects such as the anharmonic nature of atomic bonds, charge transfer and many-body effects. It is thus widely recognized that classical potentials hinder truly predictive modeling of the dynamics and function of (bio)molecular systems. While FFs come in many levels of sophistication, they can never be exact, because there is no known analytic parametrization of the true quantum mechanical atomic interactions as described by the Schrödinger equation (SE).

One possible solution to the accuracy problem is provided by direct *ab initio* molecular dynamics (AIMD) simulations, where the quantum-mechanical forces are computed on the fly for atomic configurations at every time step. The majority of AIMD simulations employ the current workhorse method of electronic-structure theory, namely density-functional approximations (DFA) to the exact solution of the SE for a system of nuclei and electrons. Unfortunately, different DFAs could yield contrasting results for the structure, dynamics, and properties of molecular systems. Furthermore, DFA calculations are not systematically improvable. Another option is the use of explicitly correlated *post*

*Hartree–Fock* methods in AIMD simulations, alas this leads to a steep increase in the computational resources required.

A series of methodological advances in the field of machine learning (ML) have opened up another avenue, by providing easy-to-parameterize universal approximators with more flexibility in the reconstruction of potential-energy surface (PES) and corresponding FFs. Recently, a wide range of sophisticated ML models for small molecules and elemental materials [7–60] have been proposed for constructing PES from DFA calculations. While these results are encouraging, direct ML fitting of molecular PESs relies on the availability of large reference datasets to obtain an accurate model. Frequently, those ML models require training on thousands or even millions of atomic configurations, preventing the construction of ML models using high-level *ab initio* methods, for which energies and forces only for 100s of conformations can be practically computed.

This predicament suggest that a tight integration between ML and physics is necessary to close the gap between efficient FFs and accurate high-level *ab initio* methods. The key idea explored in this thesis, is to take advantage of conserved quantities in dynamical processes in addition to other fundamental physical laws to inform a universal approximator without compromising its generality. Statistical inference is thus focused on the challenging aspects of the problem, while *a priori* knowledge about the atomic interactions is represented exactly and artifact-free. As a result, we expect significant improvements in data efficiency in the reconstruction process.

Turning this concept into a scientific contribution is a challenge that requires expertise across both disciplines. It is important to recognize parallels and identify how related problems have been approached in the past to avoid duplication of efforts or the pursuit of fruitless endeavors. Ideally, this cross-pollination between fields will provide fresh perspectives on long standing challenges, as the goals of ML and natural sciences are somewhat complementary: Whereas natural laws represent concise descriptions of the underlying mechanisms of a process, ML models can uncover regularities within observations without relying on such high-level concepts and thus help to discover causal structures.

With those considerations in mind, we set out to construct FFs with the accuracy of high-level *ab initio* calculations. We approach this challenge using principles of probabilistic inference, which define a set of hypotheses and conditions them on the made observations. The resulting predictions are particularly robust to overfitting, because all viable hypotheses are always taken into account. Hilbert space learning algorithms enable us to rigorously incorporate fundamental temporal and spatial symmetries of atomic systems to create robust models for which parametrization from highly accurate,

but costly coupled cluster reference data becomes viable. This development leads us through the fields of computational physics, as well as operator, group, and optimization theory. We use our models to carry out MD simulations at the coupled cluster level of electronic-structure theory and provide insights into the dynamical behavior of molecules with up to a few dozen atoms. To the best of our knowledge, we are first to allow converged MD simulations with fully quantized electrons and nuclei at this scale.

## 1.1 Theoretical background

First, we will briefly introduce the relevant fundamentals of quantum mechanics that crucially informed and motivated the development of our approaches. We begin with the concept of the PES, which arises from the Born-Oppenheimer approximation to the SE, as the solution for its electronic degrees of freedom. We will outline the hierarchy of electronic structure methods most commonly used to solve the electronic SE and review the numerical methods underpinning them. To complete the full approximation of the SE, we turn our attention to dynamics of the nuclei, which are typically treated classically in a process called MD simulation. Within that framework, a technique known as path-integral MD (PIMD) provides a way to account for nuclear quantum effects [61]. In preparation for the development of our PES models, we finally review Noether's theorem that formulates a connection between conservation laws and symmetries of physical systems, providing valuable constraints for their behavior. Overall, this chapter attempts to outline the fundamental reasons for the high computational complexity of *ab initio* methods, which ultimately spurred our efforts documented in this thesis.

### 1.1.1 Ab initio quantum chemistry

Each measurable quantity in a physical system has an associated quantum mechanical operator that describes it. The total energy of a system of electrons with coordinates  $\mathbf{r}$  and mass  $m_e$ , and nuclei with coordinates  $\mathbf{R}$  and mass  $M_i$ , atomic number  $Z_i$ , is described by the Hamiltonian operator, as the sum of their kinetic and potential energies  $\hat{H} = \hat{T}_n + \hat{T}_e + \hat{V}_{nn} + \hat{V}_{ee} + \hat{V}_{en}$  with

$$\begin{aligned}\hat{T}_n &= -\sum_i \frac{1}{2M_i} \nabla_{\mathbf{R}_i}^2 & \hat{V}_{nn}(\mathbf{R}) &= \sum_i \sum_{j>i} \frac{Z_i Z_j}{\|\mathbf{R}_i - \mathbf{R}_j\|} \\ \hat{T}_e &= -\sum_i \frac{1}{2m_e} \nabla_{\mathbf{r}_i}^2 & \hat{V}_{ee}(\mathbf{r}) &= \sum_i \sum_{j>i} \frac{1}{\|\mathbf{r}_i - \mathbf{r}_j\|} \\ && \hat{V}_{en}(\mathbf{r}, \mathbf{R}) &= -\sum_i \sum_j \frac{Z_i}{\|\mathbf{R}_i - \mathbf{r}_j\|},\end{aligned}\tag{1.1}$$

in atomic units. The operators  $\hat{T}_n$ ,  $\hat{T}_e$ ,  $\hat{V}_{nn}(\mathbf{R})$ ,  $\hat{V}_{ee}(\mathbf{r})$  and  $\hat{V}_{en}(\mathbf{r}, \mathbf{R})$  represent the nuclear and electron kinetic energy, as well as the nuclear-nuclear, electron-electron and nuclear-electron interaction potentials, respectively. We can therefore obtain all possible outcomes of a total energy measurement from the spectrum of the Hamiltonian. In other words, solving a quantum mechanical problem entails the diagonalization of  $\hat{H}$ . This gives rise to the (non-relativistic and time-independent) SE

$$\hat{H}\Psi_i(\mathbf{r}, \mathbf{R}) = E_i\Psi_i(\mathbf{r}, \mathbf{R}),\tag{1.2}$$

where  $\Psi_i$  are the eigenfunctions of the Hilbert space defined by the Hamiltonian, i.e. the systems stationary states [62, 63]. Each eigenfunction describes the energy state for a given energy  $E_i$ . The eigenfunction with the lowest energy  $\Psi_0$  is referred to as the ground-state wavefunction whereas all the other ones are excited state wavefunctions.

### Born-Oppenheimer approximation

An almost universally used simplification of the SE, is the so-called Born-Oppenheimer (BO) approximation, which separates the wavefunction  $\Psi(\mathbf{r}, \mathbf{R}) = \psi_e(\mathbf{r}; \mathbf{R})\psi_n(\mathbf{R})$  into a product of nuclear and electron terms. The underlying idea is that the much lighter electrons can be assumed to adjust instantly to nuclear motion. On the electronic timescale, nuclei are effectively stationary and almost act like an external potential on the electrons [63]. With a Hamiltonian

$$\hat{H}_{el} = \hat{T}_e + \hat{V}_{nn} + \hat{V}_{ee} + \hat{V}_{en}\tag{1.3}$$

that only depends parametrically on the position of the nuclei, while neglecting their motion, the SE can be solved for the electronic degree of freedom. The resulting electronic energy is a function of nuclear coordinates, giving rise to the concept of a PES [62]. Many properties of atomic structures can be explored in terms of the topography of that surface.

The corresponding nuclear SE for  $\Psi_n(\mathbf{R})$  describes how the nuclei move on that PES and it is solved independently to complete the approximation of the full wave function.

Typically, nuclear motion is not treated in terms of quantum mechanics, but rather classically via integration of Newton's equations of motion. This process is referred to as a molecular dynamics (MD) simulation. With increasingly accurate descriptions of the quantum mechanical behavior of electrons, the lack of nuclear quantum effects is however starting to become a significant source of error [64]. Methods such as path integral MD (PIMD) [65] can incorporate quantum mechanics into MD simulations, albeit at very high additional computational cost.

Even within the BO approximation, the SE can not be solved analytically. Neither is it possible to solve it using general grid-based techniques for boundary value partial differential equations, due to the large number of degrees of freedom. Even at meager resolution and without accounting for boundary conditions, a discretization grid would be impossible to keep in memory. In practice, further approximations are necessary to make interesting problems computationally tractable.

### Variational optimization

Frequently, the so-called variational principle is used to estimate the lowest energy eigenstate of the SE, the ground state  $E_0$ . In this approach, the wavefunction is expanded in an incomplete basis  $\Psi(\mathbf{p})$  with a number of adjustable parameters  $p_i \in \mathbf{p}$ . The energy functional

$$E[\mathbf{p}] = \frac{\langle \Psi | \hat{H} | \Psi \rangle}{\langle \Psi | \Psi \rangle} \geq E_0 \quad (1.4)$$

is subsequently minimized via gradient descent on the derivative  $\partial E / \partial \mathbf{p}$ . Due to this discrete finite-dimensional parametrization, the SE reduces to an eigenvalue problem (or generalized eigenvalue problem in the case of a non-orthogonal basis). Expanding  $\Psi = \sum_i^N p_i \phi_i$  in a linear basis is especially convenient, as it allows solving the SE in matrix form. Because these *trial wavefunctions* usually do not span the full Hilbert space, the true energy eigenstates  $E_i$  are approached from above with increasing quality of the basis, allowing a systematic improvement of the solution [66].

#### 1.1.2 Electron correlation

The simplest way to represent a wavefunction is to assume that it is composed of single-particle basis functions, such that

$$\Psi^H(\mathbf{r}_1, \dots, \mathbf{r}_N) = \phi_1(\mathbf{r}_1)\phi_2(\mathbf{r}_2)\cdots\phi_N(\mathbf{r}_N). \quad (1.5)$$

Under this model assumption, the motion of the electrons is uncorrelated: each moves in a mean field Coulomb potential induced by the other electrons. Unsurprisingly, this severe simplification fails to capture some of the essential interactions due to the lack of pairwise and higher-order terms: first and foremost, instantaneous electron repulsion. One important consequence of electron repulsion is described by the *Pauli exclusion principle*, which says that two identical fermions (such as electrons) can never occupy the same quantum state. This outcome can be enforced via anti-symmetry of the wavefunction, such that

$$\Psi^{\text{HF}}(\mathbf{r}_1, \dots, \mathbf{r}_i, \dots, \mathbf{r}_j, \dots, \mathbf{r}_N) = -\Psi^{\text{HF}}(\mathbf{r}_1, \dots, \mathbf{r}_j, \dots, \mathbf{r}_i, \dots, \mathbf{r}_N). \quad (1.6)$$

Anti-symmetrization requires a summation over all  $N!$  possible electronic state configurations in Eq. 1.5, which can however be expressed efficiently by means of the so-called *Slater determinant*. Computing a determinant only takes  $\mathcal{O}(N^3)$  steps, foregoing the combinatorial cost of an explicit expansion. This set of approximations is known as the *Hartree-Fock* (HF) scheme. Even in the infinite basis set limit, the HF method is not accurate. It converges to the *Hartree-Fock limit*, overestimating the energy according to the variational theorem of quantum mechanics [62, 66].

Because omitting electron correlation leads to serious deviations from experimental results, several so-called *post Hartree-Fock* approaches reintroduce electron correlation in controlled ways to improve the approximation under computational complexity considerations. For brevity, we will not discuss them broadly and only give a short review of some of the fundamental principles that most of them employ. We should remark at this point, that none of these methods are computationally cheap, because all are fundamentally restricted by the combinatorics of an interacting many-body system. The fact that the interactions grow factorially with the number of electrons is inevitable. While this challenge can be approached from many different angles, the cost-accuracy trade-off is principally determined by the maximum interaction level that is being considered in the respective method.

## Configuration interaction

The configuration interaction (CI) method increases the flexibility of the wavefunction by mixing elements of higher atomic orbitals from excited states into the ground state Slater determinant. Again, a combinatorial expansion of the wavefunction

$$\Psi^{\text{CI}} = (1 + \hat{C}_1 + \dots + \hat{C}_N)\Psi^{\text{HF}} \quad (1.7)$$

is required, this time involving Slater determinants for the configurations in which electrons are promoted from the occupied to the unoccupied orbitals. Here,  $\hat{C}_i$  denotes the excitation operator with  $i$  excited electrons, whose coefficients are once again determined using variational optimization. With this extension of the HF method, changes to the electronic distribution due to electron correlation can be captured, but the computational complexity effectively restricts this approach to single (CIS) and double excitations (CISD). While the doubly excited configurations are the most important, the effect of higher order excitations is by no means negligible. Only in the limit of full configuration interaction (full CI), the result is of the many-body SE will be exact [62, 66].

### Coupled cluster

The coupled-cluster (CC) scheme is another numerical technique to treat correlation based on HF or other trial wavefunctions. Instead of a linear excitation operator, CC uses an exponential operator, which results in a product of configurations

$$\Psi^{\text{CC}} = \exp(\hat{T})\Psi^{\text{HF}} = \exp(\hat{T}_1 + \dots + \hat{T}_N)\Psi^{\text{HF}} \quad (1.8)$$

for  $N$  electrons. A Taylor series expansion of the exponential  $\exp(\hat{T}) = \sum_i \frac{1}{i!} \hat{T}_i$  reveals that the powers of  $\hat{T}$  generate additional excited determinants, a superset that also includes the CI form of the wavefunction. Once again, this expansion is only computationally tractable if truncated early, typically after singly excited and doubly excited configurations (CCSD). A common variant additionally accounts for the effect of triple excitations using perturbation theory (CCSD(T)), which can have a substantial contribution [66]<sup>1</sup>. While both variants are highly accurate, their scaling is poor, with  $\mathcal{O}(N^6)$  and  $\mathcal{O}(N^7)$ , respectively. This complexity severely limits the applicability of the CC method to small systems and even then prevents sampling intensive tasks like long-timescale MD. Being the costliest algorithm that sees regular use, the CC level of theory is widely considered to be the 'gold standard' of quantum chemistry. Remarkably, it sometimes even exceeds the best experimental results [67].

#### 1.1.3 Density Functional Theory

An alternative to HF calculations is Density Functional Theory (DFT), which reduces the many-body to a single-body problem involving the three-dimensional electronic density  $\rho(\mathbf{r})$ , rather than a  $3N$ -dimensional many-electron wavefunction. This method makes

---

<sup>1</sup>A perturbative triples correction can of course also be added in the CI approach.

solving the SE for correlated systems computationally practicable, while still accounting for interaction effects, albeit approximately.

Hohenberg and Kohn [68] showed, that the energy of a system

$$E[\rho] = V_{\text{en}}[\rho] + T[\rho] + V_{\text{ee}}[\rho] = \int \rho(\mathbf{r}) v(\mathbf{r}) d^3\mathbf{r} + T[\rho] + V_{\text{ee}}[\rho] \quad (1.9)$$

is fully determined by the electron density in its ground state. The electronic density is therefore sufficient for the formulation of the Hamiltonian from which all observables of the system can be derived. Here,  $T[\rho]$ ,  $V_{\text{ee}}[\rho]$ , and  $V_{\text{en}}[\rho]$  are functionals describing the kinetic energy associated with the given electron density, as well as the electron-electron and nuclear-electron (due to the BO approximation) interaction energies, respectively. Since the nuclear-electron energy is simply expressed as the result of an interaction with some field  $v(\mathbf{r})$ , external interactions can be included in a straightforward way. Unfortunately, the exact forms of  $T[\rho]$  and  $V_{\text{ee}}[\rho]$  are unknown, with no available fundamental theory from which they can be derived. In practice, these functionals need to be approximated, often based on experimental results, which is why DFT is considered to not be an *ab initio* method [66].

To resolve that issue, Kohn and Sham [69] proposed to express the ground state density in terms of a system of non-interacting pseudo-particles in independent orbitals  $\phi_i$ , very much like in the HF method. For this system, the electron density and kinetic energy,

$$\rho(\mathbf{r}) = \sum_i \phi_i^2(\mathbf{r}) \quad \text{and} \quad T_s[\rho] = -\frac{1}{2} \sum_i \langle \phi_i | \nabla^2 | \phi_i \rangle \quad (1.10)$$

are exact, just like the coulombic part of the electron repulsion energy  $J[\rho]$ . With that, the total energy functional takes the form

$$E[\rho] = V_{\text{en}}[\rho] + T_s[\rho] + J[\rho] + E_{\text{xc}}[\rho], \quad (1.11)$$

$$\text{with } E_{\text{xc}}[\rho] = T[\rho] - T_s[\rho] + V_{\text{ee}}[\rho] - J[\rho], \quad (1.12)$$

where  $E_{\text{xc}}[\rho]$  is the exchange-correlation functional. Kohn-Sham DFT gives a set of eigenvalue problems governed by an effective Hamiltonian

$$\hat{H}^{\text{KS}}(\mathbf{r}) = -\frac{1}{2} \nabla^2 + v_{\text{KS}}(\mathbf{r}), \quad \hat{H}^{\text{KS}} \phi_i(\mathbf{r}) = \epsilon_i \phi_i(\mathbf{r}), \quad (1.13)$$

which are solved variationally. Here,  $\epsilon_i$  are the energies of the corresponding orbitals. [66]

The computational cost of DFT only scales with  $\mathcal{O}(N^3)$ , allowing the calculation of reference datasets in the order of hundreds of thousands for small molecules. We can

therefore use DFT to verify the correct behavior of our models in realistic long-timescale simulations for the system sizes considered here. In contrast, only a few hundreds calculations can be performed using CCSD(T) in the same timeframe.

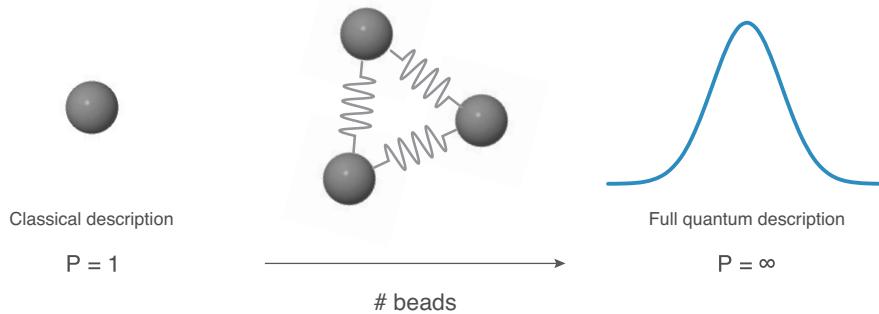
#### 1.1.4 Molecular dynamics

So far, we have addressed the solution of the electronic structure problem, which is only the first step in studying the dynamics of a molecular system according to the BO approximation. To complete the picture, a description of the nuclear interaction is still missing.

Most commonly, the principles of classical mechanics are employed to model the dynamics of nuclei, notwithstanding their quantum mechanical nature. Newton's equations of motion are integrated to propagate the evolution of nuclei on the PES in time. While the potential energy for each configuration of nuclei can in principle be obtained using *ab initio* methods, their computational cost bars access to sufficiently long simulation trajectories necessary to ensure a converged sampling of the PES. One alternative is the use of classical FFs, but their inaccuracy continues to take away from the predictive accuracy of MD simulations. Thankfully, recent ML-based developments are slowly starting to bridge the gap between accurate *ab initio* methods and efficient FFs.

MD simulations provide a picture of the dynamical behavior of an atomic system to pursue the understanding of properties such as absorption spectra, rate constants, and transport properties. The same methodology can be applied to study its macroscopic properties, by simulating the probability distribution over its microscopic state. This so-called *statistical mechanical ensemble* gives insight about its average thermodynamic quantities (such as pressure, temperature or volume), structure, and free energies along reaction paths [63, 70].

With increasing accuracy of the description of the PES, the significance of nuclear quantum effects (NQEs) is gaining importance. Previously, the residual error caused by omitting NQEs was considered negligible, in contrast to the inaccuracy of the PES model [64]. Nowadays, the inclusion of NQE becomes mandatory, as a lack thereof becomes the primary cause of deviation from experiment. Even basic properties like densities, heats of vaporization, solvation energies and dielectric constants are misrepresented without proper account of NQE [71].



**Figure 1.1** The path-integral molecular dynamics method approximates nuclear quantum effects by exploiting an isomorphism between a  $P$  particle classical polymer and a quantum system. The equilibrium averages of this polymer approximate the properties of the quantum particle. This method is exact in the limit of the number of copies  $P \rightarrow \infty$ .

### Nuclear quantum effects

A classical treatment of the nuclear dynamics is sufficiently accurate in many cases, but sometimes nuclear quantum effects need to be accounted for due to significant nuclear delocalization. For instance, this is the case in systems with light atoms, at low temperatures or for shallow potential energy landscapes. NQEs such as zero-point energy (ZPE) and tunnelling can induce significant deviations from the classical behavior. For water, a prototypical case study within the context of NQEs, the importance of NQEs has been demonstrated for ample of aspects [72]. Recent findings even demonstrate that NQEs should not only be considered in strongly interacting systems, as previously thought, but also in weak interaction as present in noble gases [73] and alkanes [74, 75].

One way of incorporating nuclear quantum effects (NQEs) is the path-integral molecular dynamics (PIMD) method based on Feynman path integrals, which establishes a one-to-one correspondence between the properties of a quantum object and a purely classical system. A 'quantum' atom becomes a ring polymer of  $P$  coupled replicas of a classical atom (so called beads), connected by harmonic oscillators. The equilibrium averages of the polymer approximate the properties of the quantum particle. In the limit of the number of copies  $P \rightarrow \infty$ , convergence to full quantum statistics is guaranteed (see Figure 1.1). This makes it possible to study quantum dynamic and thermodynamic, as well as spectroscopic properties of a system using classical approaches, without needing to solve the nuclear SE [76, 65, 1].

In practice, large  $P$  are required to obtain a good approximation of the correct quantum result, multiplying the number of interatomic potential evaluations in each MD step. While these evaluations are independent and can be parallelized, the computational burden

increases by factor  $P$  just from the copies. Moreover, longer MD trajectories are needed with growing number of beads, causing an overall non-linear scaling behavior. An accurate treatment of NEQs is therefore computationally prohibitive in AIMD simulations as it further compounds their already high computational cost.

### 1.1.5 Conservation laws

Conservation laws describe invariant properties of closed physical systems over time. They are fundamental principles of nature that characterize symmetries that must not be violated. Their big appeal is that they enable the description of macroscopic systems, without the need to consider its microscopic details. As such, conservation laws provide strong constraints on any description of a physical system.

Noether's theorem [77] states that each conserved quantity is associated with a differentiable symmetry of the action of a physical system. The action is represented as the integral of the Lagrangian  $L$  over time

$$\mathcal{S} = \int_{t_1}^{t_2} L[q(t), \dot{q}(t), t] dt, \quad (1.14)$$

with  $\dot{q} = \partial q / \partial t$ . The behavior of any dynamical system is described by the trajectories through phase space for which the action is stationary. A symmetry of a system is defined as any coordinate  $q_k$  that does not appear on the Lagrangian, with the results that  $\partial L / \partial q_k = 0$ . Then, we have for the Euler-Lagrange equation of motion

$$\frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{q}_k} \right) = \frac{\partial L}{\partial q_k} = 0 \quad \rightarrow \quad \frac{\partial L}{\partial \dot{q}_k} = C, \quad (1.15)$$

where  $C$  is a constant and  $\partial L / \partial \dot{q}_k$  is a conserved quantity, i.e. independent of time. For example, in Cartesian coordinates, the conserved quantity  $\partial L / \partial \dot{x} = m\dot{x}$  is the linear momentum. Conserved quantities in Lagrangian systems include the total energy (following from temporal invariance), as well as angular and linear momentum (roto-translational invariance).

## 1.2 Description of chapters

This thesis is structured into three main parts. We have already given a brief introduction of the basics of quantum mechanics and will now continue with the relevant ML concepts. In the following, we will present the first main contribution of this thesis, which is the

development of a gradient domain machine learning (GDML) approach for reconstructing energy-conserving molecular force fields. We proceed with the development of a novel multi-partite matching algorithm that is able to automatically identify and recover the relevant static and dynamic molecular graph symmetries as represented in the data. It allows us to construct data-efficient symmetric variants of our model (sGDML), tailored to the specific spatial symmetries of the targeted molecule. Throughout, we carefully verify the predictive capabilities of our models by applying them in MD simulations and comparing the outcome with experimental results. Finally, we provide a user-friendly software implementation of (s)GDML to make our results widely accessible.

**Chapter 2: Hilbert space learning** We introduce the theoretical foundations of Hilbert space learning algorithms and cover the mathematical tools that will be used in the following chapters. Particular emphasis is placed on how to incorporate prior information into GPs to construct especially data efficient and robust predictors. We adopt an operator perspective on the regression problem that allows a more intuitive incorporation of operator constraints and conservation laws.

**Chapter 3: Energy-conserving molecular force fields** The definition of GPs is generalized to vector-valued outputs, enabling joint inference of multiple related properties with set correlation structure. We use this formalism to define a predictor that explicitly maps to energy conserving vector fields through the use of a specialized covariance function. This allows the efficient reconstruction of molecular FFs in the gradient domain, solely based on interatomic forces as reference. We demonstrate that our approach enables MD simulations for molecules at DFT level of accuracy at a fraction of cost of explicit AIMD calculations.

**Chapter 4: Point groups and fluxional symmetries** Building on the developments from the previous chapter, we proceed with the incorporation of molecular point group and fluxional symmetries into our model. This requires the development of a multi-partite graph matching algorithm that enables the automated recovery of physically relevant symmetries from the training set. These additional constraints help to reduce the data requirements of our model even further and finally allow the construction of FFs from even higher-level *ab initio* calculations. Extensive numerical experiments demonstrate that our model enables path-integral MD simulations at quantum-chemical CCSD(T) level of accuracy for flexible molecules with up to a few dozen atoms.

### 1.3 Previously published work

Many results in this thesis have previously been published in journals. We focus on the work documented in the following articles:

- Chmiela, S., Tkatchenko, A., Sauceda, H. E., Poltavsky, I., Schütt, K. T., Müller, K.-R. (2017) "Machine Learning of Accurate Energy-conserving Molecular Force Fields". In: *Science Advances*, 3(5), e1603015.
- Chmiela, S., Sauceda, H. E., Müller, K.-R., Tkatchenko, A. (2018) "Towards Exact Molecular Dynamics Simulations with Machine-Learned Force Fields". In: *Nature Communications*, 9(1), 3887.
- Chmiela, S., Sauceda, Poltavsky, I., H. E., Müller, K.-R., Tkatchenko, A. (2019) "sGDML: Constructing Accurate and Data Efficient Molecular Force Fields Using Machine Learning". In: *Computer Physics Communications*, 10.1016/j.cpc.2019.02.007

Additional co-authored publications are listed in the following.

- Schütt, K. T., Arbabzadah, F., Chmiela, S., Müller, K. R., Tkatchenko, A. (2017) "Quantum-chemical insights from deep tensor neural networks". In: *Nature Communications*, 8, 13890.
- Schütt K. T., Kindermans, P.-J. , Sauceda, H. E. , Chmiela, S. , Tkatchenko, A. , Müller, K.-R. (2017) "SchNet: A continuous-filter convolutional neural network for modeling quantum interactions.". In: *Advances in Neural Information Processing Systems*, 31, pages 991–1001.
- Sauceda, H. E., Chmiela, S., Poltavsky, I., Müller, K.-R., Tkatchenko, A. (2019) "Molecular Force Fields with Gradient-Domain Machine Learning: Construction and Application to Dynamics of Small Molecules with Coupled Cluster Forces". In: *The Journal of Chemical Physics*, 150, 2019, 114102.



# Chapter 2

## Hilbert space learning

Supervised ML infers a relationship between pairs of inputs  $\mathbf{x} \in \mathcal{X}$  and associated outputs  $y \in \mathcal{Y}$  from a finite *training set* of  $M$  examples. The objective is to formulate a hypothesis that generalizes beyond these known data points, which is estimated by measuring the prediction error of the model on an independent *test set*. For obvious reasons, it is desirable for ML models to be data efficient, in the sense that their generalization error falls quickly with growing training set size. Efficiency is particularly important when the dataset is compiled from computationally expensive high-level *ab initio* calculations. After all, proxy models are only practical if the full procedure of data generation, training and inference outpaces the method they imitate. This is a challenge, as many ML algorithms have been developed under the assumption that data is abundant. For example, recent deep learning architectures require hundreds of thousands or even millions of data points until they are able to give useful predictions. Because their non-convex cost function mandates numerical solvers, the reconstruction of a potential for a single small molecule can therefore take several days, even on modern GPU hardware.

A more efficient alternative is provided by Hilbert space learning algorithms, as they operate in spaces of functions that match prior beliefs about the observed process. While a small number of training points is not enough to reconstruct general functions, it might be sufficient to constrain a well-behaved function space. This is alluring, because even complex physical processes involve quantities with well understood properties that can be exploited to define the structure of those Hilbert spaces. One additional benefit is that Hilbert spaces arise naturally from various problem formulations in physics and thus mediate between the two disciplines.

## 2.1 Hilbert spaces

A *Hilbert space*  $\mathcal{H}$  is a vector space over  $\mathbb{R}$  with an inner product that yields a complete metric space. The inner product gives rise to a norm  $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$ , which induces a distance metric  $d(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|$ , for  $\mathbf{x}, \mathbf{x}' \in \mathcal{H}$ . Although any  $N$ -dimensional Euclidean space  $\mathbb{R}^N$  is technically a Hilbert space, this formalism becomes particularly interesting in infinite dimension, where  $\mathcal{H}$  is a space of functions, while retaining almost all of linear algebra from vector spaces.

### 2.1.1 Reproducing kernels

Many ML algorithms make use of infinite dimensional Hilbert spaces indirectly via the *kernel-trick*, which allows to express inner products of mappings  $\Phi : \mathcal{X} \rightarrow \mathcal{H}$  in terms of inputs  $\mathbf{x} \in \mathcal{X}$  via a *kernel function*  $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ :

$$k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle_{\mathcal{H}}. \quad (2.1)$$

Eq. 2.1 holds true for any symmetric and positive semi-definite kernel, i.e. it is required that  $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x})$  and any linear combination  $f = \sum_i \alpha_i \Phi(\mathbf{x}_i)$  with  $\alpha_i \in \mathbb{R}$  must satisfy

$$\langle f, f \rangle_{\mathcal{H}} = \sum_{ij} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \geq 0. \quad (2.2)$$

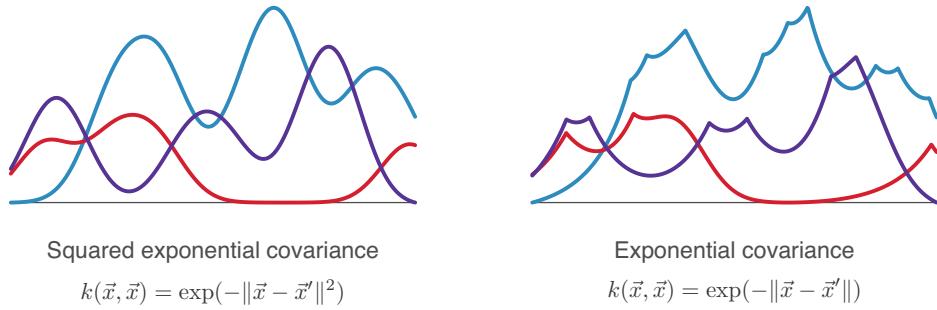
These two properties guarantee the *reproducing property* of  $\mathcal{H}$

$$f(\mathbf{x}) = \langle k(\cdot, \mathbf{x}), f \rangle_{\mathcal{H}}, \quad (2.3)$$

due to which any evaluation of  $f$  corresponds to an inner product evaluation in  $\mathcal{H}$  between the representer  $k(\cdot, \mathbf{x}) = \Phi(\mathbf{x})$  of  $\mathbf{x}$  and the function itself. We say that  $k$  is reproducing for a subset of  $\mathcal{H}$ , the *reproducing kernel Hilbert space* (RKHS). Intuitively, this means that the feature maps  $\Phi(\mathbf{x}_i)$  for all training points  $i \in [1, \dots, M]$  provide an over-complete basis for the RKHS [78].

### 2.1.2 Representer theorem

The computational tractability of Hilbert space learning algorithms is afforded by the *representer theorem* which states that in an RKHS  $\mathcal{H}$ , the minimizer  $\hat{f} \in \mathcal{H}$  of a loss function



**Figure 2.1** Example functions drawn from GP priors based on different types of covariance functions. The squared exponential kernel defines a smooth, infinitely differentiable space of solutions (left), whereas the exponential kernel gives rise to non-differentiable functions (right). A well-defined hypothesis space can drastically simplify the learning problem.

$\mathcal{L}: \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$  in a regularized risk functional with  $\lambda > 0$ ,

$$\hat{f} = \arg \min_{f \in \mathcal{F}} \left[ \frac{1}{M} \sum_i^M \mathcal{L}(f(\mathbf{x}_i), y_i) + \lambda \|f\|^2 \right], \quad (2.4)$$

admits a representation of the form

$$f(\cdot) = \sum_i^M \alpha_i k(\cdot, \mathbf{x}_i) \quad (2.5)$$

for any  $\alpha_i$ . It therefore reduces the infinite-dimensional minimization problem in a function space to finding the optimal values for a  $M$ -dimensional vector of coefficients  $\alpha$  [79–81]. Because we are not fitting a model with a fixed number of predetermined parameters, Hilbert space algorithms are generally regarded as non-parametric methods.

## 2.2 Gaussian process models

When formulated in terms of the squared loss  $\mathcal{L}(\hat{f}(\mathbf{x}), y) = (\hat{f}(\mathbf{x}) - y)^2$ , the regularized risk functional in Eq. 2.4 can be interpreted as the maximum *a posteriori* estimate of a Gaussian process (GP) [81]. One common perspective on GPs is that they specify a prior distribution over a function space. GPs are defined as a collection of random variables that jointly represent the distribution of the function  $f(\mathbf{x})$  at each location  $\mathbf{x}$  and thus as a generalization of the Gaussian probability distribution from vectors to functions. This conceptual extension makes it possible to model complex beliefs.

At least in part, the success of GPs – in contrast other stochastic processes – can be attributed to the fact that they are completely defined by only the first- and second-

order moments, the mean  $\mathbb{E}[f(\mathbf{x})] = \mu(\mathbf{x})$  and covariance  $k(\mathbf{x}, \mathbf{x}')$  for all pairs of random variables [82]:

$$f(\mathbf{x}) \sim \mathcal{GP} [\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')]. \quad (2.6)$$

Any symmetric and positive definite function is a valid covariance that specifies the prior distribution over functions we expect to observe and want to capture by a GP. Altering this function can change the realizations of the GP drastically: e.g. the squared exponential kernel  $k(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2 \sigma^{-1})$  (with a freely selectable length-scale parameters  $\sigma$ ) defines a smooth, infinitely differentiable function space, whereas the exponential kernel  $k(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\| \sigma^{-1})$  produces a non-differentiable realizations (see Figure 2.1). The ability to define a prior explicitly, gives us the opportunity to express a wide range of hypotheses like boundary conditions, coupling between variables or different symmetries like periodicity or group invariants. Most critically, the prior characterizes the generalization behavior of the GP, defining how it extrapolates to previously unseen data. Furthermore, the closure properties of covariance functions allow many compositions, providing additional flexibility to encode complex domain knowledge from existing simple priors [83].

The challenge in applying GP models lies in finding a kernel that represents the structure in the data that is being modeled. Many kernels are able to approximate universal continuous functions on a compact subset arbitrarily well, but a strong prior restricts the hypothesis space and drastically improves the convergence of a GP while preventing overfitting [84]. Each training point conditions the GP, which allows increasingly accurate predictions from the posterior distribution over functions with growing training set size.

A number of attractive properties beyond their expressivity make GPs particularly useful in the physical sciences:

- There is a unique and exact closed form solution for the predictive posterior, which allows GPs to be trained analytically. Not only is this faster and more accurate than numerical solvers, but also more robust. E.g. choosing the hyper-parameters of the numerical solver for NNs often involves intuition and time-consuming trial and error.
- Because a trained model is the average of *all* hypotheses that agree with the data, GPs are less prone to overfit, which minimizes the chance of artifacts in the reconstruction [85]. Other types of methods that start from a more general hypothesis space require more complex regularization schemes.

- Lastly, their simple linear form makes GPs easier to interpret, which simplifies analysis of the modeled phenomena and supports theory building.

### 2.2.1 Gaussian process regression

It is straightforward to use GPs for regression: Given a sample  $(\mathbf{X}, \mathbf{y}) = \{(\mathbf{x}_i, y_i)\}_i^M$ , we compute the sample covariance matrix  $(\mathbf{K})_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$  and use the posterior mean

$$\mu(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})] = k_{\mathbf{X}}(\mathbf{x})^\top (\mathbf{K} + \lambda \mathbb{I})^{-1} \mathbf{y} \quad (2.7)$$

to make predictions about new points  $\mathbf{x}$ . Here,  $k_{\mathbf{X}}(\mathbf{x}) = [k(\mathbf{x}, \mathbf{x}_1), \dots, k(\mathbf{x}, \mathbf{x}_M)]^\top$  is the vector of covariances between the new point  $\mathbf{x}$  and all training points. In the frequentist interpretation, this algorithm is also referred to as *kernel ridge regression*.

We can also calculate the variability of the hypotheses at every point via the posterior variance

$$\sigma^2(\mathbf{x}) = \mathbb{E}[(f(\mathbf{x}) - \mu(\mathbf{x}))^2] = k(\mathbf{x}, \mathbf{x}) - k_{\mathbf{X}}(\mathbf{x})^\top (\mathbf{K} + \lambda \mathbb{I})^{-1} k_{\mathbf{X}}(\mathbf{x}), \quad (2.8)$$

which gives us an idea about the uncertainty of the prediction. We remark here, that the posterior variance is generally not a measure for the accuracy of the prediction. It rather describes how well the hypothesized space of solutions is conditioned by the observations and whether the made assumptions are correct.

## 2.3 Encoding prior information

Prior knowledge about the problem at hand is an essential ingredient to the learning task, as it can drastically increase the efficiency of the training process and robustness of the reconstruction. A ML model that starts from a general set of assumptions will require more training data to achieve the same performance, compared to one that is restricted to solutions with certain known properties. A unique feature of GPs is that they provide a direct way to incorporate such constraints on the hypothesis space [78].

In the context of this thesis, we are particularly interested in regularities that arise from invariances and symmetries of physical systems. The idea to reduce equations in a way that leaves them invariant is not new in physics. In fact, Jacobi already developed a procedure to simplify Hamilton's dynamical equations of mechanics based on the conserved quantities of dynamical systems [86] in the middle of the eighteenth century. Heisenberg was the first to apply group theory to quantum mechanics, where he exploited the permuta-

tional symmetry of indistinguishable quantum particles in 1926. Even in modern physics, new symmetries are still routinely discovered [87].

Often, these can be expressed in simple terms, although they originate from complex interactions. It is a fascinating prospect that those regularities can be exploited without an understanding of the underlying principles that cause them. In that sense, statistical models allow us to describe a system long before we fully understand them, which can provide insights that would not be possible otherwise. In contrast, traditional FFs are restricted to prescribed interaction patterns and are not able to recover new structure from data.

In this chapter we will review the three most important ways to include prior knowledge in GPs: indirectly via composition of the training dataset and directly by construction of suitable mean and covariance functions. The choice of covariance function is especially important, which is why we will describe several distinct ways to construct them.

### 2.3.1 Observations

It is easy to see that the composition of the training dataset plays a crucial role in how well a GP predictor will perform. Not only can the distribution of the training dataset reflect prior knowledge, likewise the kind of observable and its representation set the focus on what is believed to be important for the inference task at hand.

#### Sampling process

Every datapoint in the training set reflects a small amount of knowledge about some unknown process that we aim to recover. Most individual training points are rather insignificant on their own, because they only provide extremely localized information, unless they represent boundary conditions or important topographical features, like extrema. Seen as a sampling distribution however, they collectively carry additional high-level information. For example, trajectories of dynamic processes are often multi-modally distributed, revealing the preferred configurational states of a system. The frequency of occurrence of each configurational state is proportional to the probability that it is visited. It might therefore be desirable to bias the sampling towards the frequently visited states to promote a more accurate reconstruction in those areas. Other applications might call for a uniform sampling scheme, that assigns equal importance to all states. Clearly, the sampling process influences how well the trained model will perform in various applications, giving rise to a variety of different stratification and active learning [11, 43, 55, 56, 60] techniques.

What observable is measured during the sampling process is equally important. Are we making experimental measurements with a large noise floor or collecting essentially exact *ab initio* reference calculations? Can we gather more information per sample location than just the function value, e.g. by taking derivative observations [88, 89], compressed measurements [90] or group-level statistics [91]?

Developing a good sampling scheme is non-trivial in many cases and often requires domain expertise. It contributes to a successful reconstruction to a significant degree.

### Representation

Once the data is captured, it needs to be represented in terms of features that are considered to be particularly informative, i.e. well-correlated with the prediction target. For example, parametrizing a molecular graph in terms of dihedral angles instead of pairwise distances might be advantageous when modeling complex transition paths.

The representation of the data also provides the first opportunity to incorporate known invariances of the task at hand. Especially in physical systems, certain transformations conserve its properties, which introduces redundancies that can be exploited with a representation that shares those invariances. E.g. physical systems can generally be translated and rotated in space without affecting their attributes. Often, the invariances extend to more interesting group of transformations like rotations, reflections or permutations, providing further opportunities to reformulate the learning problem into a simpler, but equivalent one. Conveniently, any non-linear map  $\mathbf{D} : \mathcal{X} \rightarrow \mathcal{D}$  of the input to a covariance function yields another valid covariance function, providing a direct way to incorporate desired invariances into existing kernels [92].

#### 2.3.2 Covariance function

Symmetries in the input data naturally translate to symmetries in the output. If a molecular graph is mirror symmetric, so will be its potential energy surface. However, sometimes there is structure in the output that is not tied to the input at all. This is the case when the predicted property is subject to a conservation law, e.g. the energy of a system is conserved as its geometry transforms through time. There is no representation of individual data points that would be able to capture this kind of symmetry.

Instead, conservation laws have to be incorporated as constraints into the predictor, to restrict the space of feasible solutions. This is achieved elegantly in GPs, via modification of the covariance function in a way that gives rise to a prior that obeys the desired symmetry. Any function drawn from that prior will then inherit the same invariances [78, 82]. Before

developing a covariance function that fits our problem, we will briefly highlight different ways to construct them. After all, arbitrary functions of two inputs  $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$  are not necessarily valid covariance functions. For that purpose we will switch away from the probabilistic view that we held so far and provide a perspective that is more intuitive in the physics context.

### Integral transforms

We can think of the covariance function as a kernel of an linear integral transform that defines an operator

$$\hat{T}_k f(\mathbf{x}) = \int_{\mathcal{X}} k(\mathbf{x}, \mathbf{x}') f(\mathbf{x}') d\mathbf{x}', \quad (2.9)$$

which maps a function  $f(\mathbf{x})$  from one domain to another [82, 93]. In this view,  $\hat{T}_k f(\mathbf{x}) = \hat{f}(\mathbf{x})$  corresponds to the posterior mean of our GP. Note, that  $\hat{T}_k f(\mathbf{x})$  remains a continuous function, even if we discretize the integration domain. This is the case in the regression setting, when we are only able to observe our target function partially. With that in mind, an integral operator can be regarded as a continuous generalization of the matrix-vector product using a square matrix with entries  $(\mathbf{K})_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$  and a vector  $\alpha$ . Then,

$$(\mathbf{K}\alpha)_i = \sum_j^M k(\mathbf{x}_i, \mathbf{x}_j) \alpha_j \quad (2.10)$$

is the discrete analogon to  $\hat{T}_k f(\mathbf{x})$  [94]. Note, that this expression is closed under linear transformation: any linear constraint  $\hat{G}[\hat{T}_k]$  propagates into the integral and gives rise to a new covariance function.

However, there are several alternative construction options, one of them through explicit definition of the frequency spectrum of  $\hat{T}_k$ . Due to the translational symmetry of physical systems, we are particularly interested in stationary covariance functions that only depend on pairwise distances  $\delta = \mathbf{x} - \mathbf{x}'$  between points. In that setting, *Bochner's theorem* says that symmetric, positive definite kernels can be constructed via the inverse Fourier transform of a probability density function  $p(\delta)$  in frequency space [92, 93, 82, 95]:

$$k(\delta) = \mathcal{F}(p(\delta)) = \int p(\omega) e^{-i\omega^\top \delta} d\omega. \quad (2.11)$$

The following perspective might however be more intuitive when approaching this problem form a physics background: Since  $\hat{T}_k f(\mathbf{x})$  is the reconstruction from point-wise observations  $y_i = f(\mathbf{x}_i)$ , we are ideally looking for an operator that leaves our unknown

target function invariant, such that  $\hat{T}_k f(\mathbf{x}) = f(\mathbf{x})$ . This is another way of saying that our estimate  $\hat{f}(\mathbf{x})$  lives in the space spanned by the eigenfunctions  $\varphi_i \in \mathcal{F}$  of the operator defined by the kernel function (with coefficients  $c_i \in \mathbb{R}$ ), giving

$$\hat{f}(\mathbf{x}) = \sum_i c_i \varphi_i(\mathbf{x}) \quad \text{with} \quad \hat{T}_k \varphi_i = \lambda_i \varphi_i. \quad (2.12)$$

It is impossible to overlook that there is a strong analogy between the covariance function in a GP process and the Hamiltonian in the SE. Both operators formulate constraints that give rise to Hilbert space of possible states of the modeled object, whether it is the wavefunction or the hypothesis space of the GP. Although this is where the similarities end, this connection certainly illustrates that GPs are particularly suitable to reconstruct physical processes in a principled way.

**Example**

Consider the squared exponential (SE) kernel

$$k_{\text{SE}}(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{-\|\mathbf{x} - \mathbf{x}'\|^2}{\sigma^2}\right). \quad (2.13)$$

What prior information does it encode? It turns out, that it encodes the most basic and in fact, a necessary condition for any reconstruction problem: the smoothness assumption. Reconstruction is only possible, if there is some underlying regularity, i.e. if similar inputs produce similar outputs. Only then can we extrapolate from a limited number of observations.

It is more intuitive to think about smoothness in terms of the power spectrum in the frequency domain [93]. A smooth function is called band-limited, because it only carries negligible energy after a certain cut-off frequency. This is the case for the SE kernel, as the power spectrum representation of its Fourier transform reveals (which is another SE function). To see how a kernel affects the prediction, we rewrite Eq. 2.9 in the frequency domain, with Fourier transform  $\mathcal{F}$ ,

$$\mathcal{F}(T_k f(\mathbf{x})) = \mathcal{F}(k(\mathbf{x}, \mathbf{x}')) \mathcal{F}(f(\mathbf{x}')) \quad (2.14)$$

and observe that the prediction decomposes into the product of the spectrum of the kernel and the spectrum of the observed function. In the case of the SE,  $\mathcal{F}(k(\mathbf{x}, \mathbf{x}'))$  attenuates the energy in the high frequencies as it slowly approaches zero. The SE kernel thus acts like a low-pass filter that lets smooth functions pass unaffected. If the function is too complex, it recovers the low frequency portion of signals.

### 2.3.3 Mean function

In most applications, the GP prior mean function  $\mu(\mathbf{x}) = 0$  is set to zero, which leads to predictions  $\hat{f}(\mathbf{x}) \approx 0$  as  $\|\mathbf{x} - \mathbf{x}'\| \rightarrow 0$  for stationary kernels. Convergence to a constant outside of the training regime is desirable for data-driven models, because it means that the prediction degrades gracefully in the limit, instead of producing unforeseeable results. However, if a certain *asymptotic* behavior of the modeled function is known, the prior mean function offers the possibility to prescribe it. For example, we could introduce a log barrier function

$$\mu(\mathbf{x}) = -\log(\mathbf{b} - \mathbf{x}) \quad (2.15)$$

that ramps up the predicted quantity towards infinity for  $\mathbf{x} \geq \mathbf{0}$ . In a molecular PES model, such a barrier would represent an atom dissociation limit, which could be useful to ensure that a dynamical process stays confined to the data regime.

In the spirit of how the Slater determinant accounts for the average effect of electron repulsion without explicit correlation, the mean of a GP is used to prescribe a sensible predictor response outside of the data regime.

## 2.4 Summary

In this chapter we have reviewed the general concept of Hilbert space learning and discussed how it relates to GPs. This powerful formalism provides various ways in which the natural invariances of the data can be taken into account, to construct highly data efficient predictors without loss of generality.

We have attempted to highlight conceptual similarities between the Hamiltonian and GP covariance functions, by introducing the operator interpretation of the regression problem. In this view, the reconstructed function lies in the eigenspace of the integral operator defined by the covariance function. Via its closure properties, this space of solutions can be shaped, e.g. by imposing linear operator constraints like conservation laws and symmetries. Stationary covariance functions can be alternatively constructed via direct specification of the frequency spectrum and subsequent inverse Fourier transform. The asymptotic behavior of the GP is controlled via the mean function. We have proposed the introduction of a log barrier function to approximate a realistic predictor response as the molecule approaches the dissociation limit.

In the following two chapters we will get more specific and use these techniques to construct an efficient and accurate ML model that encodes all spatial and temporal symmetries of PESs for small molecules.



# Chapter 3

## Energy-conserving molecular force fields

**Partial results of the presented work have been published in:**

- Chmiela, S., Tkatchenko, A., Sauceda, H. E., Poltavsky, I., Schütt, K. T., Müller, K.-R. (2017) "Machine Learning of Accurate Energy-conserving Molecular Force Fields". In: *Science Advances*, 3(5), e1603015.

A fundamental property that any force field  $\mathbf{F}(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)$  must satisfy is the conservation of total energy, which implies that  $\mathbf{F}(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N) = -\nabla E(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N)$ . Any classical mechanistic expressions for the potential energy (also denoted as classical FF) or analytically derivable ML approaches trained on energies satisfy energy conservation by construction. However, even if conservation of energy is satisfied implicitly within an approximation, this does not imply that the model will be able to accurately follow the trajectory of the true *ab initio* potential, which was used to fit the force field. In particular, small energy/force inconsistencies between the force field model and *ab initio* calculations can lead to unforeseen artifacts in the PES topography, such as spurious critical points that can give rise to incorrect molecular dynamics (MD) trajectories. Another fundamental problem is that classical and ML force fields focusing on energy as the main observable have to assume atomic energy additivity – an approximation that is hard to justify from quantum mechanics.

In this chapter, we present a robust solution to these challenges by constructing an explicitly conservative ML force field, which uses exclusively atomic gradient information in lieu of atomic (or total) energies. In this manner, with any number of data samples, the proposed model fulfills energy conservation by construction. Obviously, the developed ML force field can be coupled to a heat bath, making the full system (molecule and bath) non-energy-conserving.

We remark that atomic forces are true quantum-mechanical observables within the BO approximation by virtue of the Hellmann-Feynman theorem. The energy of a molecular system is recovered by analytic integration of the developed gradient-domain machine learning (GDML) model. We demonstrate that our approach is able to accurately reproduce global PESs of intermediate-sized molecules within  $0.3 \text{ kcal mol}^{-1}$  for energies and  $1 \text{ kcal mol}^{-1} \text{ \AA}^{-1}$  for atomic forces relative to the reference data. This accuracy is achieved when using less than 1000 training geometries to construct the GDML model and using energy conservation to avoid overfitting and artifacts.

### 3.1 Local linearizations of the PES

We have established in the previous chapter that regularity in the target function is a necessary condition for a successful reconstruction from a limited number of observations. This requirement is satisfied by the PES, which changes smoothly as the geometry of the physical system evolves. Observing the energy of one configuration gives us a good idea about the energy of other geometries in its immediate neighborhood. Sudden, non-continuous energy jumps are physically impossible, because they would require an infinite force acting on at least one of the atoms.

Due to that smoothness, we can locally linearize a PES without uncontrollably increasing the approximation error. A linearization can be parametrized from only a few perturbations  $f(x_i + \epsilon) \approx f(x_i) + \nabla_{x_i} f(x_i) \epsilon$  on the PES, while potentially replacing a much larger amount of expensive evaluations that would have been necessary otherwise. This raises the question, whether it is possible to condition a GP using linearizations directly, as a replacement for multiple neighboring training points. A significant improvement of the convergence rate of the learning algorithm with respect to training set size would be the outcome.

#### 3.1.1 Hellman-Feynman theorem

Clearly, the ability to learn local linearizations is particularly appealing when the target function is a known process with available derivatives, like in the case of the PES. When obtained directly, without resorting to numerical approximation, linearizations can not only increase the efficiency of the learning algorithm, but also decrease signal acquisition cost. The *Hellmann-Feynman theorem* indeed provides a way to obtain analytical derivatives. It relates changes in the total energy  $\partial E$  with respect to any variation  $\partial \lambda$  of the

Hamiltonian through the expectation value

$$\frac{\partial E}{\partial \lambda} = \left\langle \Psi_\lambda \left| \frac{\partial \hat{H}_\lambda}{\partial \lambda} \right| \Psi_\lambda \right\rangle, \quad (3.1)$$

which allows the direct computation of forces  $\mathbf{F} = -\partial E / \partial \mathbf{R}$  as derivatives with respect to nuclei positions  $\mathbf{R}$ . In the case of DFT, forces can just as well be expressed in terms of electron density  $\rho(\mathbf{r}) = \Psi^2(\mathbf{r})$ ,

$$\mathbf{F} = -\frac{\partial E}{\partial \mathbf{R}} = -\int \frac{\partial V_{\text{ext}}(\mathbf{r}, \mathbf{R})}{\partial \mathbf{R}} \rho(\mathbf{r}) d\mathbf{r}, \quad (3.2)$$

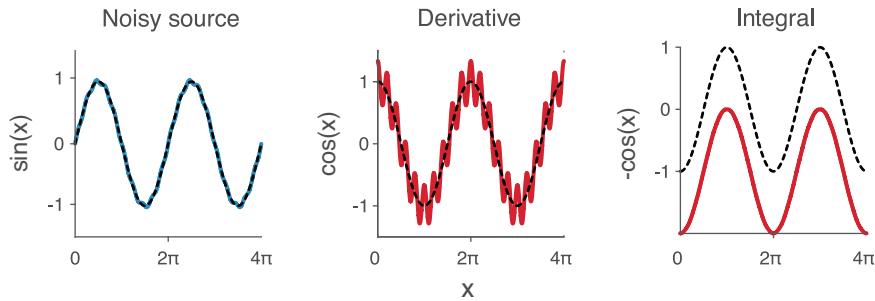
when no wave-function is explicitly available. They only depend on the potential energy due to the external field  $V_{\text{ext}}(\mathbf{r}, \mathbf{R})$ , as it is the only functional involving nuclei positions [96, 97].

Once the SE is solved for a particular atomic configuration to compute the energy, this theorem makes the additional computation of forces relatively cheap, by reusing some of the results. The fascinating part is that force observations are considerably more informative, as they represent a linearization of the PES in all directions of the  $3N$ -dimensional configuration space. Gathering a similar amount of insight about the PES numerically, would require solutions of the SE for at least  $3N + 1$  perturbations  $E(r_1, \dots, r_i + \epsilon, \dots, r_{3N})$  of the original geometry at each point. Even then, the obtained force would be subject to approximation error and oftentimes inconsistent with the energy measurement. In contrast, computing analytical forces using Hellman-Feynman theorem only requires 1–7 times the computational effort of a single energy calculation. Effectively, this theorem thus offers a more efficient way to sample PESs.

In the next section, we will develop a GP with an associated Hilbert space of energy conserving vector-valued functions, which will enable us to formulate the PES reconstruction problem in the gradient domain and thus allow us to make efficient use of those analytic forces.

### 3.1.2 Noise amplification by differentiation

A reconstruction of high-dimensional signals from derivative observations increases data-efficiency, but more crucially, also leads to a better representation derivatives in the model. While most empirical models based on point evaluations of the target function have an analytical form that allows *a posteriori* differentiation (see Figure 3.3), the resulting derivative estimates are not regulated within the loss function of the model and a faithful reconstruction is hence not guaranteed. Inevitably, this can lead to artifacts.



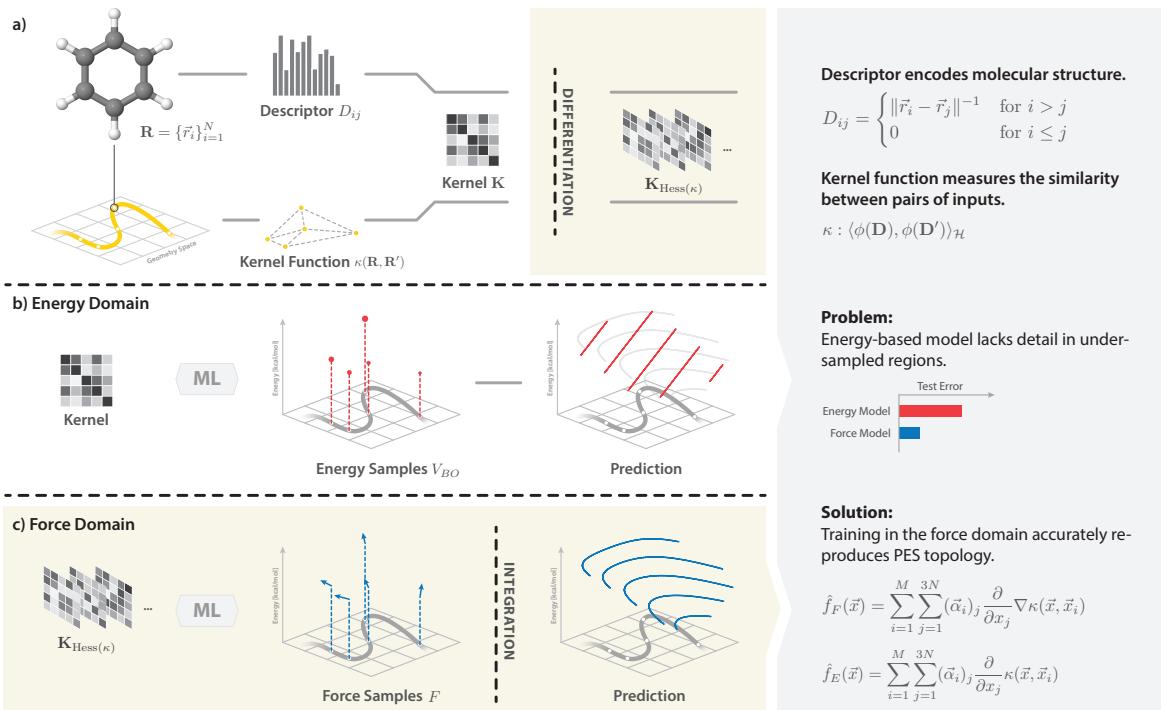
**Figure 3.1** A noisy approximation of a sine wave (blue). Although all instantaneous values are represented well, the derivative of the approximation is a poor estimator for the true derivative. This is because differentiation amplifies the high frequency noise component within the approximation (middle). Integration on the other hand acts as a low-pass filter (right) that attenuates noise. It is therefore easier to approximate a function with accurate first derivatives from derivative examples instead of function values. Note that integrals are only defined up to an additive constant, which needs to be recovered separately.

It is widely accepted, that reconstructions of functions based on a limited number of observations will generally not be error-free, either due to aliasing effects, non-ideal choice of hypothesis space or noisy training data [80]. Furthermore, the use regularization terms in the loss function of ML models will promote these errors into the high-frequency band of the residual error function. Unfortunately, the application of the derivative operator amplifies high frequencies  $\omega$  with increasing gain [98], drastically magnifying these errors. The derivative of a model  $\hat{f}'$  in the frequency domain is

$$\mathcal{F}[\hat{f}'] = i\omega \mathcal{F}[\hat{f}], \quad (3.3)$$

were  $\mathcal{F}[\hat{f}]$  is its Fourier transform (see Figure 3.1). A low test error does therefore not necessarily imply that an energy-trained PES model also reconstructs the forces of the target function reliably.

Several de-noising schemes have been developed as a post-processing step, e.g. via low-rank projection by means of principal component analysis (PCA) [99–102]. We note however, that these approaches only treat symptoms without addressing their cause. In the next section, we will develop an approach to construct FFs that are energy-conserving *a priori*, thus avoiding the application of the noise-amplifying derivative operator to a parameterized PES model.



**Figure 3.2** The construction of ML models: First, reference data from an MD trajectory are sampled. (a) The geometry of each molecule is encoded in a descriptor. This representation introduces elementary transformational invariances of energy and constitutes the first part of the prior. A kernel function then relates all descriptors to form the kernel matrix – the second part of the prior. The kernel function encodes similarity between data points. Our particular choice makes only weak assumptions: It limits the frequency spectrum of the resulting model and adds the energy conservation constraint. Hess, Hessian. (c) These general priors are sufficient to reproduce good estimates from a restricted number of force samples. (b) A comparable energy model is not able to reproduce the PES to the same level of detail.

## 3.2 Gradient domain machine learning (GDML)

The GDML approach explicitly constructs an energy-conserving force field, avoiding the application of the noise-amplifying derivative operator to a parameterized potential energy model. This can be achieved by directly learning the functional relationship

$$\hat{\mathbf{f}}_{\mathbf{F}} : \mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N\} \xrightarrow{\text{ML}} \mathbf{F} \quad (3.4)$$

between sets of atomic coordinates and interatomic forces, instead of computing the gradient of the PES model (see Figure 3.2, c and b). This requires constraining the solution space of all arbitrary vector fields to the subset of energy-conserving gradient fields. The PES can be obtained through direct integration of  $\hat{\mathbf{f}}_{\mathbf{F}}$  up to an additive constant. To construct  $\hat{\mathbf{f}}_{\mathbf{F}}$ , we use a generalization of common scalar-valued GPs for structured vector fields [103–105].

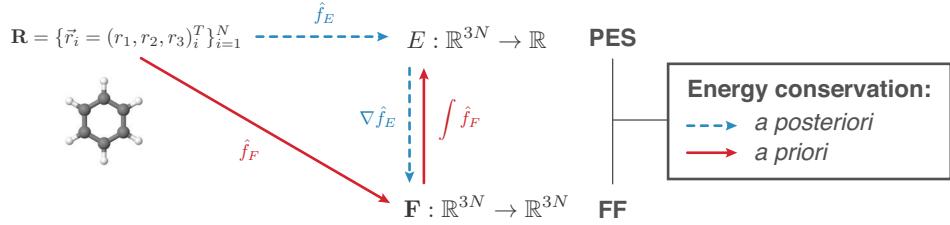
### 3.2.1 Multiple output GPs

In the simplest, and by far most prevalent regression setting, a *single* output variable  $y$  is predicted from an input vector  $\mathbf{x}$ . While this seems like the natural way to cast the PES reconstruction problem at first glance, a direct energy prediction approach carries significant disadvantages in practice, as discussed previously. Instead of reconstructing the PES directly, we will thus pursue the reconstruction of the associated force field, i.e. the negative gradient of the PES (see schematic in Figure 3.3). This somewhat unconventional approach constitutes a considerably more complex *multiple output* regression problem with vector-valued labels  $\mathbf{y}$ . It appears at first, that the higher dimensionality of the prediction target would nullify the advantages afforded by derivative measurements, but we will demonstrate later how a physically motivated formulation of the learning problem can prevent that.

Naively, and without any knowledge about the properties of the predicted vector field, we would model each output variable separately and treat them as independent, implicitly assuming that the individual outputs do not affect each other. Such a vector-valued estimator would take the form

$$\hat{\mathbf{f}}(\mathbf{x}) = [\hat{f}_1(\mathbf{x}), \dots, \hat{f}_N(\mathbf{x})]^\top, \quad (3.5)$$

where each component is a separate scalar-valued GP [106]. However, an independence assumption is hard to justify in many practical multi-output scenarios. For the force field reconstruction task, we can say with certainty that a coupling between output dimensions



**Figure 3.3** Differentiation of a PES estimator (blue) versus direct force field reconstruction (red). The law of energy conservation is trivially obeyed in the first case, but requires explicit *a priori* constraints in the latter scenario. Both approaches yield estimates for energy and forces, but a direct reconstruction of the force fields avoids the amplification of estimation errors due to the derivative operator. The challenge in estimating force fields directly lies in the complexity arising from their high  $3N$ -dimensionality.

is present due to the global nature of atomic interactions. But even if the output channels were independent *a priori*, correlations between the individual noise processes associated with each component could introduce dependencies in the posterior process [82]. An artificial decoupling would therefore ignore valuable information and yield suboptimal estimates.

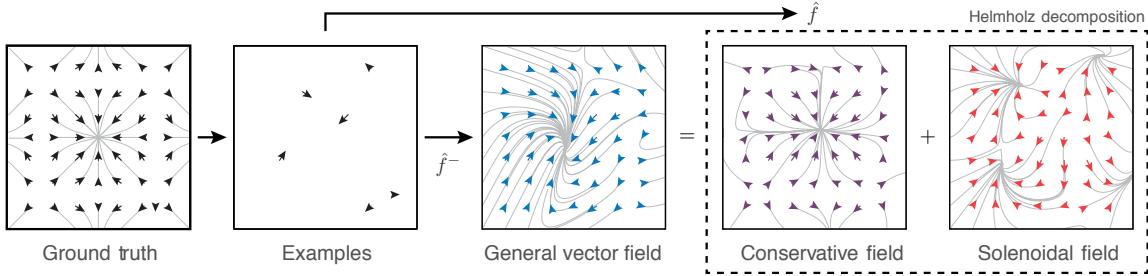
This would be an unfortunate conclusion, because multivariate output dependencies can be naturally captured by GPs through the correlation structure of the prior. Instead of mapping to scalar outputs, we can model the covariance function as a matrix  $\mathbf{k} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^{N \times N}$  that expresses the interaction among multiple output components. Together with a vector-valued mean function  $\mu : \mathcal{X} \rightarrow \mathbb{R}^N$ , we can then sample realizations of vector-valued functions from the GP

$$\mathbf{f}(\mathbf{x}) \sim \mathcal{GP} [\mu(\mathbf{x}), \mathbf{k}(\mathbf{x}, \mathbf{x}')]. \quad (3.6)$$

In this setting, the corresponding RKHS is vector-valued and it has been shown that the representer theorem continues to hold [107]. Each component of the kernel function  $(\mathbf{k})_{ij}$  specifies a covariance between a pair of outputs  $f_i(\mathbf{x})$  and  $f_j(\mathbf{x})$ , which makes it straightforward to impose linear constraints  $\mathbf{g}(\mathbf{x}) = \hat{G}[\mathbf{f}(\mathbf{x})]$  on the GP prior

$$\mathbf{g}(\mathbf{x}) \sim \mathcal{GP} \left[ \hat{G}\mu(\mathbf{x}), \hat{G}\mathbf{k}(\mathbf{x}, \mathbf{x}') \hat{G}'^\top \right]. \quad (3.7)$$

and hence also the posterior [108, 103, 109, 110]. Here,  $\hat{G}$  and  $\hat{G}'$  act on the first and second argument of the kernel function, respectively. Linear constraints include simple conservation laws, but also operations like differential equations, allowing the construction of models that are consistent with the laws that underpin many physical processes [111–115].



**Figure 3.4** Modeling gradient fields (leftmost subfigure) based on a small number of examples. With GDML, a conservative vector field estimate  $\hat{\mathbf{f}}$  is obtained directly (purple). In contrast, a naïve estimator  $\hat{\mathbf{f}}^-$  with no information about the correlation structure of its outputs is not capable to uphold the energy conservation constraint (blue). We perform a Helmholtz decomposition of the naïve non-conservative vector field estimate to show the error component due violation of the law of energy conservation (red). This significant contribution to the overall prediction error is completely avoided with the GDML approach.

Single output GPs are included as a special case in this multiple output generalization: setting the matrix-valued kernel function  $\mathbf{k}(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}')\mathbb{1}_N$  to a diagonal matrix treats all outputs as independent and hence recovers the decoupled-output GP.

### 3.2.2 Conservative vector-valued GPs

The correlation between dimensions  $F_i(\mathbf{R})$  in a FF follows directly from  $\mathbf{F}(\mathbf{R}) = -\nabla E(\mathbf{R})$ . Unless the energy  $E$  decomposes without loss into contributions of its individual free parameters  $r_i \in \mathbf{R}$ , so that

$$E(\mathbf{R}) = E(r_1) + \dots + E(r_N) \quad \text{and thus} \quad F_i(\mathbf{R}) = -\frac{\partial E(r_i)}{\partial r_i}, \quad (3.8)$$

its partial derivatives are correlated and can not be modeled independently. We therefore aim to construct a GP that inherits the correct structure of a *conservative* force field in order to increase the accuracy of the predictor and to ensure integrability, so that the corresponding energy potential can be recovered from the same model.

**Example**

We illustrate the benefit of energy conservation with the help of a two-dimensional toy problem: a synthetic potential created by two harmonic oscillators along the coordinate axes (see Figure 3.4). We wish to reconstruct this potential from a sparse set of gradient measurements  $\nabla f(\mathbf{x}_i)$  at locations  $\mathbf{x}_i = [x_1, x_2]^\top_i$ . Although this potential can be decomposed according to Eq. 3.8, we will not leverage that prior knowledge for the purpose of our example.

Instead, we train a naïve estimator  $\hat{\mathbf{f}}^-(\mathbf{x}) = [\hat{f}_1^-(\mathbf{x}), \hat{f}_2^-(\mathbf{x})]^\top$  that disregards any relationship between the two partial derivatives. It consists of two independent zero-mean GP models  $\hat{f}_i^- : \mathbb{R}^2 \rightarrow \mathbb{R}$  that both use the squared exponential kernel as covariance function. Note, that each  $\hat{f}_i^-$  depends on both inputs in  $\mathbf{x}$ . In general, the predictions made by this naïve estimator are non-conservative, as can be demonstrated via a potentially non-vanishing curl

$$\nabla \times \hat{\mathbf{f}}^- = \left( \frac{\partial \hat{f}_2^-}{\partial x_1} - \frac{\partial \hat{f}_1^-}{\partial x_2} \right) \mathbf{e}_3 \neq 0, \quad (3.9)$$

where  $\mathbf{e}_3 = [0, 0, 1]^\top$  is the standard basis vector for the third coordinate axis. It is easy to tell from that definition, that a coupling between both outputs is indispensable to impose the zero curl constraint.

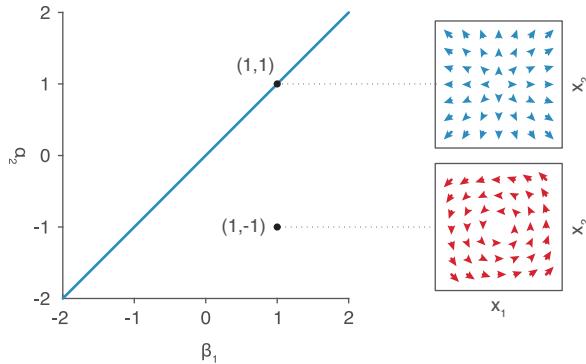
We will now use the Helmholtz theorem [116] to uniquely decompose one instance of this naïve estimator into a sum of a curl-free (conservative)  $\nabla E$  and a divergence-free (solenoidal)  $\nabla \times \mathbf{A}$  vector fields:

$$\hat{\mathbf{f}}^- = -\nabla E + \nabla \times \mathbf{A}. \quad (3.10)$$

This allows a qualitative assessment of the prediction error introduced as a direct result of violating the law of energy conservation. We perform the decomposition numerically, by sampling the gradient estimate given by  $\hat{\mathbf{f}}^-$  on a regular grid to project it onto the closest conservative vector field  $\nabla E$  by solving the Poisson equation with Neumann boundary conditions

$$\Delta E = \operatorname{div} \hat{\mathbf{f}}^- \quad \text{with} \quad \nabla E|_{\partial\Omega} = \hat{\mathbf{f}}^-|_{\partial\Omega}, \quad (3.11)$$

where  $\partial\Omega$  is the domain boundary and  $\Delta = \nabla^2$  denotes the Laplace operator. Figure 3.4 shows the residual curl component  $\nabla \times \mathbf{A}$ , which exemplifies the systematic error made by a non-conservative estimator.



**Figure 3.5** The blue line highlights the subset of parameter space for  $\alpha_2$  and  $\beta_1$  that yields conservative vector field estimates from the model in Eq. 3.12. The curl of the predicted vector field vanishes, i.e.  $\nabla \times \hat{\mathbf{f}}_F = \mathbf{0}$  only when  $\beta_1 = \alpha_2$ . This is not the case for any of the off-diagonal parameter configurations. In the shown example, the configuration  $(1, -1)$  has a constant curl of  $(0, 0, 2)^\top$  in the direction orthogonal to the  $\alpha_2$ - $\beta_1$ -plane.

### Example

To illustrate another advantage of imposing the aforementioned integrability constraints, consider the following toy problem in just two dimensions: Given a set of input data  $\mathbf{x}_i = [x_1, x_2]^\top \in \mathbb{R}^2$  we train a predictor  $\hat{\mathbf{f}}_F$  that maps each input to a corresponding gradient vector  $\mathbf{y}_i \in \mathbb{R}^2$  of some unknown function  $f_E$ .

Instead of using a conservative model, we use a naïve approach where each component of the gradient is learned independently. For the purpose of this example we will limit ourselves to linear models and construct a predictor

$$\hat{\mathbf{f}}_F(\mathbf{x}) = \begin{bmatrix} \alpha_1 x_1 + \alpha_2 x_2 \\ \beta_1 x_1 + \beta_2 x_2 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}. \quad (3.12)$$

The weights  $\alpha_i$  and  $\beta_i$  for  $i \in \{1, 2\}$  are chosen independent of each other.

Since only integrable vector fields are sensible estimates for our problem, we investigate which parameter combinations of this unconstrained model represent valid gradient fields. Integration of the first element with respect to the first free variable  $x_1$  yields

$$\hat{f}_E = \frac{\alpha_1}{2} x_1^2 + \alpha_2 x_1 x_2 + c(x_2) \quad (3.13)$$

where  $c(x_2)$  is the integration constant that depends on the remaining free variable.

**Example (cont.)**

Differentiation with respect to  $x_2$  yields

$$(\hat{\mathbf{f}}_{\mathbf{F}}(\mathbf{x}))_1 = \alpha_2 x_1 + \frac{dc}{dx_2} \stackrel{!}{=} \beta_1 x_1 + \beta_2 x_2 = (\hat{\mathbf{f}}_{\mathbf{F}}(\mathbf{x}))_2, \quad (3.14)$$

hence  $\alpha_2 \stackrel{!}{=} \beta_1$  in order for the model to be integrable. While the optimal parameters for this model are determined in  $\mathbb{R}^4$ , only a much smaller subspace is spanned by energy conserving solutions (see Figure 3.5).

We start by considering, that the force field estimator  $\hat{\mathbf{f}}_{\mathbf{F}}(\mathbf{x})$  and the PES estimator  $\hat{f}_E(\mathbf{x})$  are related via some operator  $\hat{G}$ . To impose energy conservation, we require that the curl vanishes for every input to the transformed energy model<sup>1</sup>:

$$\nabla \times \hat{G}[\hat{f}_E] = \mathbf{0}. \quad (3.15)$$

As expected, this is satisfied by the derivative operator  $\hat{G} = \nabla$  or, in the case of energies and forces, the negative gradient operator

$$\hat{\mathbf{f}}_{\mathbf{F}}(\mathbf{x}) = \hat{G}[\hat{f}_E](\mathbf{x}) = -\nabla \hat{f}_E(\mathbf{x}). \quad (3.16)$$

As outlined previously, we can directly apply this transformation to a standard scalar-valued 'energy' GP with realizations  $f_E : \mathcal{X}^{3N} \rightarrow \mathbb{R}$ . Since differentiation is a linear operator, the result is another GP with realizations  $\mathbf{f}_{\mathbf{F}} : \mathcal{X}^{3N} \rightarrow \mathbb{R}^{3N}$ :

$$\hat{f}_F \sim \mathcal{GP}\left[-\nabla \mu(\mathbf{x}), \nabla_{\mathbf{x}} k(\mathbf{x}, \mathbf{x}') \nabla_{\mathbf{x}'}^\top\right]. \quad (3.17)$$

Note, that this gives the second derivative of the original kernel (with respect to each of the two inputs) as the (co-)variance structure, with entries

$$k_{ij} = \frac{\partial^2 k}{\partial \mathbf{x}_i \partial \mathbf{x}'_j}. \quad (3.18)$$

It is equivalent (up to sign) to the Hessian  $\nabla k \nabla^\top = \text{Hess}_{\mathbf{x}}(k)$  (i.e. second derivative with respect to one of the inputs), provided that the original covariance function  $k$  is stationary (see Appendix A.1). A GP using this covariance enables inference based on the distribution

---

<sup>1</sup>For illustrative purposes, we use the definition of curl in three dimensions here, but the theory directly generalizes to arbitrary dimension. One way to prove this is via path-independence of conservative vector fields: the circulation of a gradient along any closed curve is zero and the curl is the limit of such circulations.

of partial derivative observations, instead of function values [88, 89]. Effectively, this allows training GP models in the gradient domain.

This Hessian kernel gives rise to the following force model as the posterior mean of the corresponding GP:

$$\hat{\mathbf{f}}_F(\mathbf{x}) = \sum_i^M \sum_j^{3N} (\alpha_{ij})_j \frac{\partial}{\partial x_j} \nabla k(\mathbf{x}, \mathbf{x}_i) \quad (3.19)$$

Because the trained model is a (fixed) linear combination of kernel functions, integration only affects the kernel function itself. The corresponding expression for the energy predictor

$$\hat{f}_E(\mathbf{x}) = \sum_i^M \sum_j^{3N} (\alpha_{ij})_j \frac{\partial}{\partial x_j} k(\mathbf{x}, \mathbf{x}_i) + c \quad (3.20)$$

is therefore neither problem-specific, nor does it require retraining. It is however only defined up to an integration constant

$$c = \frac{1}{M} \sum_i^M E_i + \hat{f}_E(\mathbf{x}_i), \quad (3.21)$$

that we recover separately in the least-squares sense (see Appendix A.2.1). Here,  $E_i$  are the energy labels for each training example.

Thanks to the fundamental physical connection between  $\mathbf{k}_F$  and  $k_E$ , we can resort to the extensive body of existing research on suitable kernels for the energy prediction task [7, 9, 35] as starting point for the construction of a well-performing force field kernel.

### 3.2.3 Force field covariance function

We have discussed the theory behind conservative GPs and will now put it to practice. Instead of employing the previously mentioned squared exponential kernel as the basis for our force field kernel, we consider the more general Matérn family of covariance functions [117–120]

$$\begin{aligned} k : C_{v=n+\frac{1}{2}}(d) &= B(d)P_n(d), \\ B(d) &= \exp\left(-\frac{\sqrt{2v}d}{\sigma}\right), \\ P_n(d) &= \sum_{k=0}^n \frac{(n+k)!}{(2n)!} \binom{n}{k} \left(\frac{2\sqrt{2v}d}{\sigma}\right)^{n-k}, \end{aligned} \quad (3.22)$$

where  $d = \|\mathbf{x} - \mathbf{x}'\|$  is the Euclidean distance between two inputs and  $\sigma$  is the length scale of the kernel. In particular, we are interested in the subfamily where the parameter  $\nu$  is half-integer:  $\nu = n + 1/2$ , because it allows a decomposition of this expression into a product of an exponential term  $B(d)$  and a polynomial  $P_n(d)$  of order  $n$ .

These kernels are exactly  $n$ -times differentiable. The indefinitely differentiable squared exponential kernel is recovered as a special case for  $n \rightarrow \infty$  and the non-differentiable exponential kernel with  $n = 0$ . Empirical evidence indicates that kernels with limited smoothness yield better predictors [9], even if the prediction target is infinitely differentiable. It is generally assumed, that overly smooth priors are detrimental to data efficiency, as the associated hypothesis space is harder to constrain with a finite number of (potentially noisy) training examples [82]. The differentiability of functions is directly linked to the rate of decay of their spectral density at high frequencies, which has been shown to play a critical role in spatial interpolation [118].

For  $n \geq 3$  it is hard to distinguish this class of Matérn kernels from the squared exponential kernel [82] and hence its associated hypothesis space. Since the FF kernel can only be constructed from base kernels that are at least twice differentiable, we use  $n = 2$  and obtain

$$\begin{aligned}\mathbf{k}_F(\mathbf{x}, \mathbf{x}') &= \nabla k(\mathbf{x}, \mathbf{x}') \nabla^\top = \left[ \frac{\partial}{\partial \mathbf{x}'_1} \nabla k(\mathbf{x}, \mathbf{x}'), \dots, \frac{\partial}{\partial \mathbf{x}'_{3N}} \nabla k(\mathbf{x}, \mathbf{x}') \right]^\top \\ &= \left( 5(\mathbf{x} - \mathbf{x}')(\mathbf{x} - \mathbf{x}')^\top - \mathbb{1}\sigma(\sigma + \sqrt{5}d) \right) \frac{5}{3\sigma^4} \exp\left(-\frac{\sqrt{5}d}{\sigma}\right)\end{aligned}\quad (3.23)$$

for the Matérn  $\nu = 5/2$  covariance. The integral of this force field kernel (i.e. the gradient of the original kernel) is:

$$k_E(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}, \mathbf{x}') \nabla^\top = (\mathbf{x} - \mathbf{x}')(\sigma + d) \frac{5}{3\sigma^3} \exp\left(-\frac{\sqrt{5}d}{\sigma}\right). \quad (3.24)$$

A derivation for general integers  $n$  can be found in Appendix A.1.1.

### Roto-translational invariance

Covariance functions remain valid under any transformation of their domain  $\mathbf{D}: \mathcal{X} \rightarrow \mathcal{D}$ , i.e.  $k(\mathbf{D}(\mathbf{x}), \mathbf{D}(\mathbf{x}')) = k_{\mathbf{D}}(\mathbf{x}, \mathbf{x}')$  is again a kernel function. A rather trivial implication is that all invariances of that input transformation are inherited, providing yet another opportunity to characterize the properties of the predictor [92].

For example, the FF kernel in Eq. 3.23 is not invariant to relative roto-translations of its inputs, which is however a basic symmetry of physical systems. We can easily add this missing invariance by representing the molecular geometries in terms of relative distances

between atoms, instead of Cartesian coordinates. Pairwise distance matrices with entries  $(\mathbf{A})_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|$  only remove the six superfluous degrees of freedom that describe the global location and orientation of the molecular graph. Since global interactions between multiple particles can be fully described in pairwise terms (as done in the Hamiltonian), this representation does not take away from the expressiveness of the model.

The so-called Coulomb matrix representation [7] goes one step further and represents each pair of nuclei in terms their Coulomb interaction instead of a simple distance. The Coulomb energy is the only nuclei-nuclei interaction term in the Hamiltonian and empirically a good starting point for inference about molecular properties [9]. We use a slight variation of this descriptor for our purpose, whereby atoms of different type interact on a normalized scale,

$$D_{ij} = \begin{cases} \|\mathbf{r}_i - \mathbf{r}_j\|^{-1} & \text{for } i > j \\ 0 & \text{for } i \leq j \end{cases}, \quad (3.25)$$

foregoing the relative weighting with atomic numbers from the original formulation.

In combination with a descriptor, the FF kernel from Eq. 3.23 transforms to

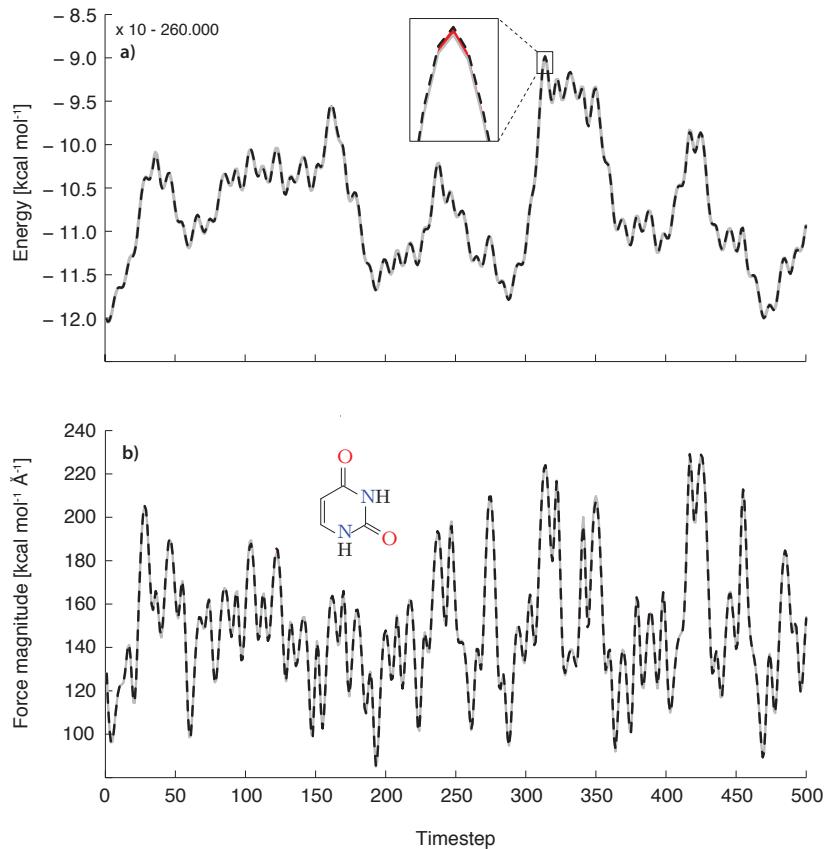
$$\mathbf{k}_F = \mathbf{J}_D (\nabla k_D \nabla^T) \mathbf{J}_D^T, \quad (3.26)$$

according to the derivative chain rule. We are therefore also interested in the Jacobian of this descriptor  $\mathbf{J}_D = [\text{vec}(\nabla_{\mathbf{r}_1} \mathbf{D}), \dots, \text{vec}(\nabla_{\mathbf{r}_N} \mathbf{D})]^T$ , which is composed of the vectorized derivatives with respect to each Cartesian coordinate in  $\mathbf{r}$ :

$$\mathbf{J}_D = (\nabla_{\mathbf{r}_i} \mathbf{D})_{ij/ji} = \begin{cases} (\mathbf{r}_i - \mathbf{r}_j) \|\mathbf{r}_i - \mathbf{r}_j\|^{-3} & \text{for } i > j \\ 0 & \text{for } i \leq j \end{cases}. \quad (3.27)$$

**Periodic boundary conditions** The Coulomb matrix can be easily extended to represent unit cell boundary conditions, which allows a description of macro-scale systems like bulk gases, liquids or crystal structures in addition to molecular structures. To achieve this, we modify the underlying Euclidean distance metric such that it adheres to the so-called *minimum-image convention* whereby each atom in the unit-cell only interacts with the closest copy of each other atom. Effectively, the region of the unit cell is (topologically) mapped onto a four-dimensional torus, making the boundaries disappear [121]. For an orthogonal unit cell, the true distance between two particles  $\mathbf{r}_i$  and  $\mathbf{r}_i$  is then

$$\tilde{\delta}_e = \begin{cases} \delta_e - \mathbf{w}_e, & \text{if } \delta_e > \frac{1}{2}\mathbf{w}_e \\ \delta_e + \mathbf{w}_e, & \text{otherwise} \end{cases} \quad (3.28)$$



**Figure 3.6** Predicted energies (a) and forces (b) for 500 consecutive time steps along a MD trajectory of uracil at 500 K. The highly accurate GDML predictions (gray) follow the reference trajectory (black, dashed) closely. To highlight small deviations, the area between both curves is marked red.

where  $\delta = \mathbf{r}_i - \mathbf{r}_j$  and  $\mathbf{w}$  is the width of the cell along all coordinate axes  $e \in \{x, y, z\}$ . This periodic extension retains all of the properties of the CM, as it essentially only translates the lattice vectors, such that none of the pairwise distances cross the unit cell walls.

With this new distance metric, the periodic system is fully described. We remark, that there is no need to account for the interactions with virtual copies of the system within the ML model, as their influence is already factored into the reference data.

### 3.3 Numerical experiments

We now proceed to evaluate the performance of the GDML approach by learning and then predicting AIMD trajectories for molecules (see Figure 3.6), including benzene, uracil, naphthalene, aspirin, salicylic acid, malonaldehyde, ethanol, and toluene (see Table B.3 for details on these molecular datasets). The GDML model for each dataset was trained

on 1000 geometries with corresponding force labels, sampled uniformly according to the MD@DFT trajectory energy distribution (see Table 3.1 and Figure 4.3).

In each numerical experiment, we measure the performance of the model using the well-established mean absolute error (MAE) and root-mean-square error (RMSE) for both, energy and force predictions (see Tables B.1 and 3.1 and Figure 4.3). Since forces are multivariate, we analyze them under two additional aspects that permit a better assessment of their topographical accuracy: The magnitude error

$$\epsilon_{\text{mag}} = \|(\hat{\mathbf{f}}_{\mathbf{F}})_i\| - \|\mathbf{F}_i\| \quad (3.29)$$

describes the average extend to which the slope of the predicted PES differs from the reference calculation, whereas the angular distance

$$\epsilon_{\text{ang}} = \frac{1}{\pi} \cos^{-1} \left( \frac{(\hat{\mathbf{f}}_{\mathbf{F}})_i \cdot \mathbf{F}_i}{\|(\hat{\mathbf{f}}_{\mathbf{F}})_i\| \|\mathbf{F}_i\|} \right) \in [0, 1] \quad (3.30)$$

measures how accurate the direction of the predicted forces is. An angular distance of zero indicates perfect alignment, while an error of one shows that the predicted force is inverted. We compute the MAE and the RMSE using both measures.

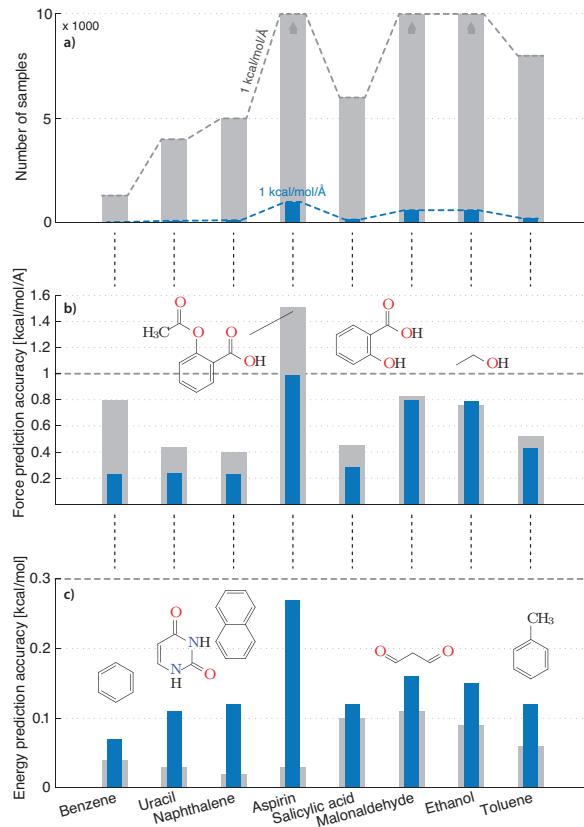
### 3.3.1 Datasets

The datasets range in size from 150 k to nearly 1 M conformational geometries, sampled from MD trajectories with a resolution of 0.5 fs, although only a very small subset is necessary to train our model. We include molecules of different sizes with corresponding PESs that exhibit different levels of complexity. The energy range across all data points within a dataset spans from 20 to 48 kcal mol<sup>-1</sup> and force components range from 266 to 570 kcal mol<sup>-1</sup> Å<sup>-1</sup> (see Table B.3). The total energy and force labels for each dataset were computed using the PBE+vdW-TS electronic structure method [122, 123].

### 3.3.2 Baseline tests

#### Training exclusively on energies

To establish a baseline, we first contrast the GDML prediction results with the output of a model that has been exclusively trained on energies (see Table B.1). For a fair comparison, we construct the energy model using the same base kernel and descriptor, but perform the hyperparameter search individually to ensure optimal model selection. Furthermore we allow the energy model to use more data points than the GDML model: specifically we



**Figure 3.7** Efficiency of GDML predictor versus a model that has been trained on energies. (a) Required number of samples for a force prediction performance of  $\text{MAE} (1 \text{ kcal mol}^{-1} \text{ \AA}^{-1})$  with the energy-based model (gray) and GDML (blue). The energy-based model was not able to achieve the targeted performance with the maximum number of 63,000 samples for aspirin. (b) Force prediction errors for the converged models (same number of partial derivative samples and energy samples). (c) Energy prediction errors for the converged models. All reported prediction errors have been estimated via cross-validation.

multiplied the training set size  $M$  by the number of atoms in one molecule times its three spatial degrees of freedom  $3N$ . This configuration yields equal kernel matrix sizes for both models and therefore equal levels of complexity in terms of the optimization problem. We compare both models on the basis of the required number of samples (Figure 3.7a) to achieve a force prediction accuracy of  $1 \text{ kcal mol}^{-1} \text{ \AA}^{-1}$ . Furthermore, the prediction accuracy of the force and energy estimates for fully converged models (w.r.t. number of samples) (Figure 3.7, b and c) are judged on the basis of the mean absolute error (MAE) and root mean square error performance measures.

It can be seen in Figure 3.7a that the GDML model achieves a force accuracy of  $1 \text{ kcal mol}^{-1} \text{ \AA}^{-1}$  using only 1000 samples for each PES reconstruction. Conversely, a pure energy-

**Table 3.1** GDML prediction accuracy for interatomic forces and total energies for all datasets. Energy errors are in  $\text{kcal mol}^{-1}$ , force errors in  $\text{kcal mol}^{-1} \text{\AA}^{-1}$ . Each model is trained on 1000 geometries with corresponding force labels.

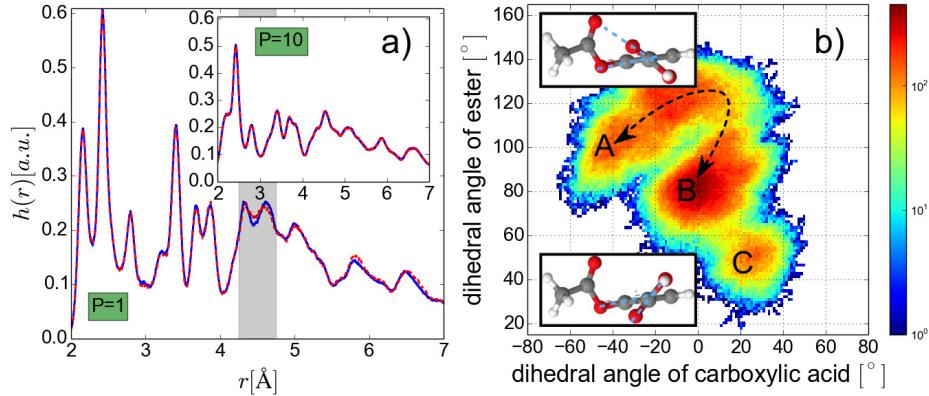
<b>Dataset</b>	<b>Energy error</b>		<b>Force error</b>		<b>Magnitude</b>		<b>Angle</b>	
	<b>MAE</b>	<b>RMSE</b>	<b>MAE</b>	<b>RMSE</b>	<b>MAE</b>	<b>RMSE</b>	<b>MAE</b>	<b>RMSE</b>
Benzene	0.07	0.09	0.23	0.34	0.21	0.30	0.0041	0.0079
Uracil	0.11	0.14	0.24	0.38	0.24	0.33	0.0040	0.0066
Naphthalene	0.12	0.15	0.23	0.34	0.21	0.28	0.0033	0.0115
Aspirin	0.27	0.36	0.99	1.41	0.91	1.19	0.0169	0.0244
Salicylic acid	0.12	0.15	0.28	0.43	0.32	0.43	0.0038	0.0065
Malonaldehyde	0.16	0.25	0.80	1.15	0.71	0.97	0.0109	0.0184
Ethanol	0.15	0.20	0.79	1.12	0.99	1.33	0.0130	0.0237
Toluene	0.12	0.16	0.43	0.62	0.35	0.45	0.0055	0.0088

based model would require up to two orders of magnitude more samples to achieve a similar accuracy.

### Training a non-conservative force model

The superior performance of the GDML model cannot be simply attributed to the greater information content of force samples. To further demonstrate that it is indeed the construction of the GDML model that leads to this positive result and not the force labels alone, we perform another experiment using a naïve force model along the lines of the toy example shown in Figure 3.4 (see Table 3.1 and Appendix B.0.2 for details on the prediction accuracy of both models). The naïve force model is nonconservative but identical to the GDML model in all other aspects. Note that its performance deteriorates significantly on all data sets compared to the full GDML model.

It is noticeable that the GDML model at convergence (w.r.t. number of samples) yields higher accuracy for forces than an equivalent energy-based model (see Figure 3.7b and Appendix B.0.1). Here, we should remark that the energy-based model trained on a very large data set can reduce the energy error to below  $0.1 \text{ kcal mol}^{-1}$ , whereas the GDML energy error remains at  $0.2 \text{ kcal mol}^{-1}$  for 1000 training samples (see Figure 3.7c). However, these errors are already significantly below thermal fluctuations ( $k_B T$ ) at room temperature ( $\sim 0.6 \text{ kcal mol}^{-1}$ ), indicating that the GDML model provides an excellent description of both energies and forces, fully preserves their consistency, and reduces the



**Figure 3.8** Results of classical and PIMD simulations. The recently developed estimators based on perturbation theory were used to evaluate structural and electronic observables [1]. (a) Comparison of the interatomic distance distributions,  $h(r) = \left\langle \frac{2}{N(N-1)} \sum_{i < j}^N \delta(r - \| \mathbf{r}_i - \mathbf{r}_j \|) \right\rangle_{P_t}$ , obtained from GDML (blue line) and DFT (dashed red line) with classical MD (main frame), and PIMD (inset). a.u., arbitrary units. (b) Probability distribution of the dihedral angles (corresponding to carboxylic acid and ester functional groups) using a 20 ps time interval from a total PIMD trajectory of 200 ps.

complexity of the learning task. These are all desirable features of models that combine rigorous physical laws with the power of data-driven machines.

### 3.3.3 Driving MD simulations with GDML

The ultimate test of any force field model is to establish its aptitude to predict statistical averages and fluctuations using MD simulations. The quantitative performance of the GDML model is demonstrated in Figure 3.8 for classical and quantum MD simulations of aspirin at  $T = 300$  K. Figure 3.8 a shows a comparison of interatomic distance distributions,  $h(r)$ , from MD@DFT and MD@GDML. Overall, we observe a quantitative agreement in  $h(r)$  between DFT and GDML simulations. The small differences in the distance range between 4.3 and 4.7 Å result from slightly higher energy barriers of the GDML model in the pathway from A to B corresponding to the collective motions of the carboxylic acid and ester groups in aspirin. These differences vanish once the quantum nature of the nuclei is introduced in the PIMD simulations [124]. In addition, long-time scale simulations are required to completely understand the dynamics of molecular systems. Figure 3.8B shows the probability distribution of the fluctuations of dihedral angles of carboxylic acid and ester groups in aspirin. This plot shows the existence of two main metastable configurations A and B and a short-lived configuration C, illustrating the nontrivial dynamics captured by the GDML model. Finally, we remark that a similarly good performance as for

aspirin is also observed for the other seven molecules shown in Figure 3.7. The efficiency of the GDML model (which is three orders of magnitude faster than DFT) should enable long-time scale PIMD simulations to obtain converged thermodynamic properties of intermediate-sized molecules with the accuracy and transferability of high-level *ab initio* methods.

### PIMD simulation details

Path-integral molecular dynamics (PIMD) is a method that incorporates quantum mechanical effects into MD simulations using Feynman’s path integral formalism (see Section 1.1.4). Here, PIMD simulations were performed using  $P = 10$  beads at ambient temperature using the GDML model interface [125] to the i-PI code [124]. We used recently developed estimators based on perturbation theory to evaluate structural and electronic observables [1]. The total time of simulation was 200 ps for aspirin and 100 ps for the rest of the molecules. We used the NVT ensemble with a time step of 0.5 fs throughout.

## 3.4 Practical considerations

### 3.4.1 Explicit treatment of N-body correlations

Long-range many-body interactions are a key ingredient in the accurate description of physical systems, crucially determining their structure, stability, and response properties [126]. As already discussed in Section 1.1.2, the accuracy of electronic structure methods is largely determined by the interaction order that is being considered. Even in the atomistic approximation, omitting interactions can lead to serious deviations from the true quantum-mechanical behavior. In fact, the Hellmann-Feynman theorem (see Eq. 3.1) relates atomic forces to the expectation value of the (many-body) Hamiltonian derivatives, showing that they do indeed interact globally.

Unsurprisingly, a genuine reconstruction of many-body phenomena thus requires a global model that correlates all atoms. The GDML model satisfies this requirement, because it couples all atoms through a matrix-valued force field covariance function (Eq. 3.24). Each of its entries  $(\mathbf{k}_F)_{ij} = \partial^2 k / \partial x_i \partial x_j$  defines a non-linear similarity between two atoms in the molecule, which leads to global interactions in the corresponding GP regression model. The posterior mean in Eq. 3.19 describes the force acting on atom  $i$  due

to atom  $j$  under fixed configuration of all other atoms as

$$\hat{\mathbf{f}}_{F_{ij}}(\mathbf{x}) = \sum_i^M (\alpha_i)_j \frac{\partial^2}{\partial x_i \partial x_j} k(\mathbf{x}, \mathbf{x}_i). \quad (3.31)$$

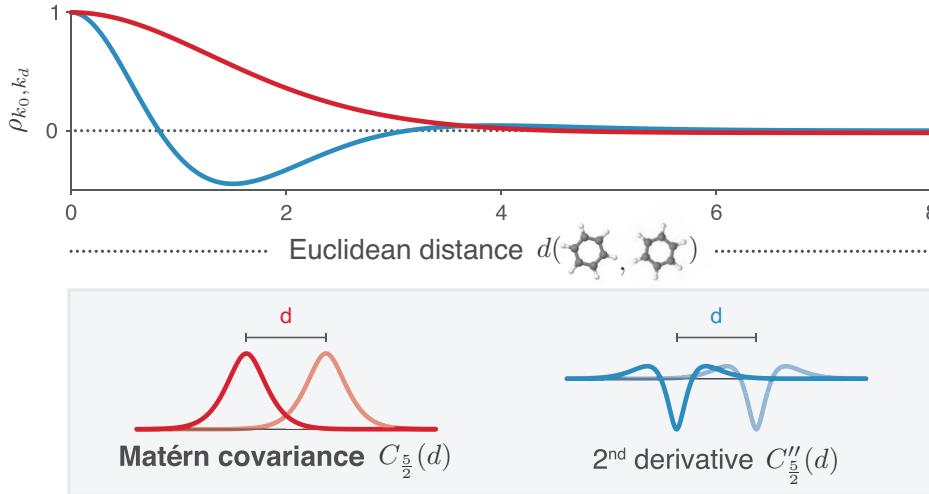
This formulation allows the GDML model to capture chemical, as well as long-range interactions, as long as they are present in the reference data and fall within the error of the model. While ML potentials are ubiquitous, a global treatment of interactions is unusual. Many existing models [14–17, 19, 20, 22–29, 31, 32, 34, 35, 37–42, 45–47, 54, 59] impose an explicit localization of individual atom contributions to the total energy, neglecting the true many-body nature of quantum-mechanical systems in favor of computational efficiency. Not least, because an explicit many-body treatment is expensive and only feasible with highly data-efficient models. The total energy is expressed as a linear combination of local environments characterized by a descriptor that acts as a non-unique partitioning function to the total energy. Unfortunately, this approach runs the risk of miss-representing the dynamical behavior of the molecule in simulation [127].

While limiting the scope of atomic interactions eventually becomes inevitable with growing system size, it is crucial to introduce this approximation in a controlled way. Only then will statistical models be truly transferable and behave predictably across a wide range of systems. Unbiased models such as GDML are required as an underpinning, in that scenario.

### 3.4.2 Numerical stability

A frequent problem with GPs are numerical instabilities due to ill-conditioned covariance matrices caused by training points that are too close together. Since we draw observations from inherently redundant MD trajectories, this is indeed a justified concern in our application.

However, our empirical observations suggest that covariance matrices based on derivative covariance functions (such as  $\mathbf{k}_F$ ) are generally better conditioned than those constructed from the unmodified covariance function  $k$ . This phenomenon is well-known in literature [128] and attributed to the fact, that derivatives of (covariance) functions are more complex and thus only weakly correlated for similar inputs, whereas there is a stronger correlation before the derivative operator is applied (see Figure 3.9). In fact, the improved conditioning of derivative covariances has previously been exploited to create numerically robust GPs via substitution of nearby points with approximate linearizations [129].



**Figure 3.9** The Pearson correlation coefficient  $\rho_{k_0, k_d} = \text{cov}(k_0, k_d) / (\sigma_{k_0} \sigma_{k_d})$  of a pair of covariance functions in dependence of their spatial separation  $d$ . Here,  $\sigma_{k_0}$  and  $\sigma_{k_d}$  are the standard deviations of both covariance functions. We compare the Matérn covariances  $k_0 = C_{v=n+\frac{1}{2}}(\|x\|)$  and  $k_d = C_{v=n+\frac{1}{2}}(\|x - d\|)$  for  $n = 2$  (red) and its second derivatives, as used in the GDML approach (blue). The correlation for small distances drops off quickly using the gradient domain covariance function, which improves the numerical stability of the GP.

However, we also remark that particular caution is required, when using a descriptor  $\mathbf{D} : \mathcal{X} \rightarrow \mathcal{D}$  in combination with derivative covariance functions. In that setting we have

$$\mathbf{k}_F = \mathbf{J}_{\mathbf{D}} \text{Hess}(k_{\mathbf{D}}) \mathbf{J}_{\mathbf{D}'}^T \quad (3.32)$$

for the covariance function after application of the derivative chain rule. Here,  $k_{\mathbf{D}}(\mathbf{x}, \mathbf{x}') = k(\mathbf{D}(\mathbf{x}), \mathbf{D}(\mathbf{x}'))$  and  $\mathbf{J}_{\mathbf{D}}$  is the Jacobian of the descriptor (see Eq. 3.27). Note, that  $\text{Hess}(k_{\mathbf{D}})$  is a Gram matrix and  $\mathbf{J}_{\mathbf{D}}$  is a projection  $\mathbf{J}_{\mathbf{D}} : \mathcal{D} \rightarrow \mathcal{X}$  from descriptor to input space. Even if  $\text{Hess}(k_{\mathbf{D}})$  is well-conditioned, the projection can be rank reducing, e.g. if  $\dim(\mathcal{D}) < \dim(\mathcal{X})$  or when  $\mathbf{J}_{\mathbf{D}}$  is rank-deficient in the first place.

Here, we use a descriptor based on pairwise distance matrices that maps from  $\dim(\mathcal{D}) = N(N-1)/2$  to input space of  $\dim(\mathcal{X}) = 3N$  (see Section 3.2.3). For  $N < 7$  atoms,  $\mathbf{J}_{\mathbf{D}}$  elevates the dimensionality of the covariance matrix and thus inevitably reduces the rank. Even for  $N \geq 7$ , the projection  $\mathbf{J}_{\mathbf{D}}$  yields a rank-deficient covariance matrix, since  $D$  removes the 6 roto-translational degrees of freedom, which leads to ambiguities in the derivative with respect to Cartesian coordinates, hence

$$\text{rank}[\mathbf{J}_{\mathbf{D}}] = \text{rank}[\mathbf{J}_{\mathbf{D}} \text{Hess}(k_{\mathbf{D}}) \mathbf{J}_{\mathbf{D}'}^T] = 3N - 6 < \dim(\mathcal{X}). \quad (3.33)$$

As a result, the FF kernel function in Eq. 3.26 requires regularization to yield a well-posed optimization problem.

## 3.5 Software implementation

We provide a Python software implementation of all models developed in this thesis, including GDML. User-friendly routines enable the reconstruction and evaluation of force fields from custom reference geometries with corresponding forces and energies. Forces and energies for new geometries can then be queried in fractions of a millisecond on a regular laptop computer (see Table C.1).

### 3.5.1 Program overview

Our main goal with this reference implementation is to provide a compact working example of the model in an accessible programming language. We offer one variant of our program with sophisticated parallel processing support for ubiquitous multi-core CPUs and another one for state-of-the-art multi-GPU computing environments. While adhering to best-practices for writing readable code, our main focus is on performance. Hence, we make full use of programming language specific optimizations, e.g. vectorized operations as a replacement for slow nested loops. These allow us to achieve performance comparable to natively compiled code.

The tasks of FF reconstruction and evaluation are separated into independent modules for training (`train`) and prediction (`predict`). All necessary routines for reference data sampling, symmetry recovery, model parametrization are packaged in the training module. It generates lightweight model files that contain the preprocessed essentials for FF evaluation, which are then independently instantiated and queried using the second module. This separation makes it possible to centralize training on a high performance computer while the completed model can be efficiently used anywhere. For that purpose, we designed the prediction module to be minimal and self-contained in the sense that it only contains logic that is absolutely essential for generating energy and forces for a given input geometry. This structure greatly simplifies the integration of GDML into any application that requires a FF. On top of that, we include a command-line interface (CLI) `sgdml` that exposes the functionality of both modules to the shell. It provides an easy introduction to GDML model reconstruction, guiding the user through the complete process.

Appendix C describes how to use our software and how to interface with the FF simulation engines ASE [130] and i-PI [131] to run various atomistic simulations, such as classical and path integral molecular dynamics, vibrational analysis, structure optimization and the computation of transition paths.

### Source code

Software, documentation, datasets and pre-trained models are available at [www.sgdml.org](http://www.sgdml.org). The GDML model developed in this chapter is accessible via the flag `use_sym=False` using the Python API or `--gdm1` through the CLI.

## 3.6 Summary

In this chapter we have developed the GDML approach, which enables accurate reconstructions of complex multidimensional PES using explicitly energy-conserving GPs. Our model allows AIMD simulations to be carried out at greatly accelerated speed, with the accuracy of high-level quantum chemistry calculations.

Achieving this goal required generalizing the GP formalism to support simultaneous mappings to multiple outputs with predefined correlation structure. It enabled us to define a covariance function that gives rise to a Hilbert space of vector-valued functions that obey the law of energy conservation. Any vector field prediction made by the GP is therefore guaranteed to be a valid force field with an associated PES. Not only does this approach simplify the learning problem, it also reduces the reference data acquisition cost via analytical gradient sampling by virtue of the Hellman-Feynman theorem.

Empirical analyses revealed the advantages of learning in the gradient domain, as opposed to PES reconstructions from energy labels alone. We have discussed why GDML meets the demands placed on force fields in practical MD simulations particularly well. Furthermore, we investigated the numerical stability of our method. The performance of our model was demonstrated for AIMD trajectories of intermediate-sized molecules, including naphthalene, benzene, toluene, naphthalene, ethanol, uracil, and aspirin. GDML is able to reproduce global PESs for these molecules with an accuracy of  $0.3 \text{ kcal mol}^{-1}$  for energies and  $1 \text{ kcal mol}^{-1}\text{\AA}^{-1}$  for atomic forces using only 1000 conformational geometries for training.

In the next chapter we will develop our model further and incorporate additional physical priors. Energy conservation is a symmetry that is implied by homogeneity of

---

time. Additionally, molecules possess well-defined rigid space group symmetries, as well as dynamic nonrigid symmetries that can be exploited to construct even more efficient models.



# Chapter 4

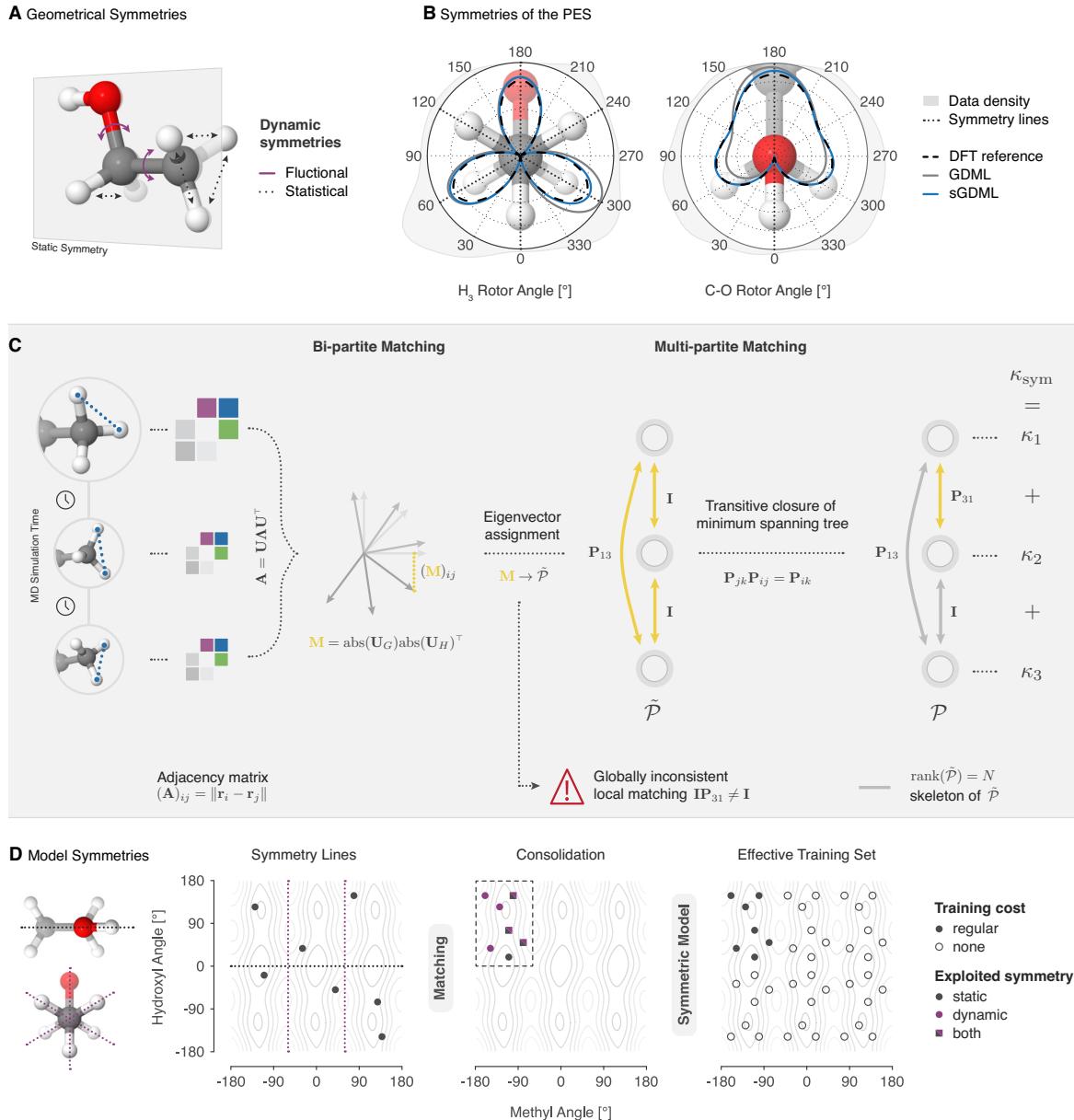
## Point groups and fluxional symmetries

**Partial results of the presented work have been published in:**

- Chmiela, S., Sauceda, H. E., Müller, K.-R., Tkatchenko, A. (2018) "Towards Exact Molecular Dynamics Simulations with Machine-Learned Force Fields". In: *Nature Communications*, 9(1), 3887.

One can classify physical symmetries of molecular systems into symmetries of space and time and specific static and dynamic symmetries of a given molecule (see Figure 4.1). Global spatial symmetries include rotational and translational invariance of the energy, while homogeneity of time implies energy conservation. These global symmetries were already successfully incorporated into the GDML model introduced in the previous chapter.

Additionally, molecules possess well-defined rigid space group symmetries (i.e. reflection operation), as well as dynamic nonrigid symmetries (i.e., methyl group rotations). For example, the benzene molecule with only six carbon and six hydrogen atoms can already be indexed in  $6!6! = 518400$  different, but physically equivalent ways. However, not all of these symmetric variants are accessible without crossing impassable energy barriers. Only the 24 symmetry elements in the  $D_{6h}$  point group of this molecule are relevant. While methods for identifying molecular point groups for polyatomic rigid molecules are readily available [132], Longuet-Higgins [5] has pointed out that non-rigid molecules have extra symmetries. These dynamical symmetries arise upon functional group rotations or torsional displacements and they are usually not incorporated in traditional force fields and electronic structure calculations. Typically, extracting nonrigid symmetries requires chemical and physical intuition about the system at hand. In this chapter we develop a physically motivated algorithm for data driven discovery of all relevant molecular symmetries from MD trajectories. This will allow us to impose the same symmetries onto the FF



**Figure 4.1** Fully data-driven symmetry discovery. (A, B) Our multipartite matching algorithm recovers a globally consistent atom-atom assignment across the whole training set of molecular conformations, which directly enables the identification and reconstructive exploitation of relevant spatial and temporal physical symmetries of the molecular dynamics. (C) The global solution is obtained via synchronization of approximate pairwise matchings based on the assignment of adjacency matrix eigenvectors, which correspond in near isomorphic molecular graphs. We take advantage of the fact that the minimal spanning set of best bipartite assignments fully describes the multipartite matching, which is recovered via its transitive closure. Symmetries that are not relevant within the scope of the training dataset are successfully ignored. (D) This enables the efficient construction of individual kernel functions for each training molecule, reflecting the joined similarity of all its symmetric variants with another molecule. The kernel exercises the symmetries by consolidating all training examples in an arbitrary reference configuration from which they are distributed across all symmetric subdomains. This approach effectively trains the fully symmetrized dataset without incurring the additional computational cost.

covariance function to constrain the hypothesis space of the GP and finally improve the data-efficiency of the model to allow training from coupled cluster reference data.

## 4.1 Positive-semidefinite assignment

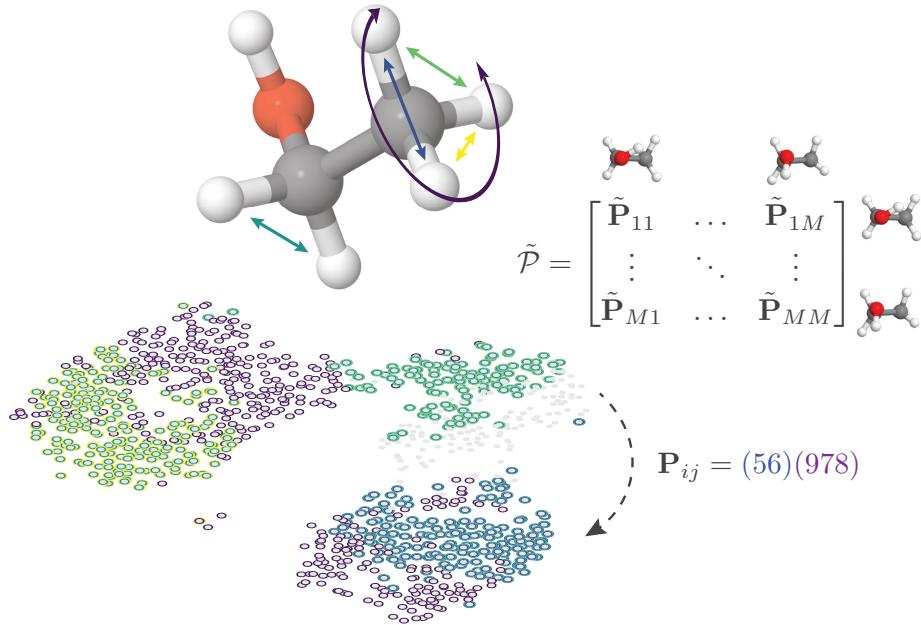
MD trajectories consist of smooth consecutive changes in nearly isomorphic molecular graphs. When sampling from these trajectories the combinatorial challenge is to correctly identify the same atoms across the examples such that the learning method can use consistent information for comparing two molecular conformations in its kernel function. While so-called bi-partite matching allows to locally assign atoms  $\mathbf{R} = \{\mathbf{r}_1, \dots, \mathbf{r}_N\}$  for each pair of molecules in the training set, this strategy alone is not sufficient as the assignment needs to be made globally consistent by multipartite matching in a second step [133–135]. The reason is that optimal bi-partite assignment yields indefinite functions in general, which are problematic in combination with kernel methods [136]. They give rise to indefinite kernel functions, which do not define a Hilbert space. Practically, there will not exist a metric space embedding of the complete set of approximate pairwise similarities defined in the kernel matrix and the learning problem becomes ill-posed. A multipartite correction is therefore necessary to recover a non-contradictory notion of similarity across the whole training set. A side benefit of such a global matching approach is that it can robustly establish correspondence between distant transformations of a geometry using intermediate pairwise matchings, even if the direct bi-partite assignment is not unambiguously possible.

### 4.1.1 Solving the multi-way matching problem

We start by defining the bi-partite matching problem in terms of adjacency matrices as representation for the molecular graph. To solve the pairwise matching problem we therefore seek to find the assignment  $\tau$  which minimizes the squared Euclidean distance between the adjacency matrices  $\mathbf{A}$  of two isomorphic graphs  $G$  and  $H$  with entries  $(\mathbf{A})_{ij} = \|\mathbf{r}_i - \mathbf{r}_j\|$ , where  $\mathbf{P}(\tau)$  is the permutation matrix that realizes the assignment:

$$\arg \min_{\tau} \mathcal{L}(\tau) = \|\mathbf{P}(\tau)\mathbf{A}_G\mathbf{P}(\tau)^{\top} - \mathbf{A}_H\|^2. \quad (4.1)$$

Notably, most existing ML potentials use representations based on adjacency matrices as input [7–10, 12–54, 57–59]. An optimal assignment in terms of Eq. 4.1 therefore transfers to almost any other model and the GDML model in particular.



**Figure 4.2** T-SNE [2] embedding of all molecular geometries in an ethanol training set. Each data point is color coded to show the permutation transformations that align it with the arbitrarily chosen canonical reference state (gray points). These permutations are recovered by restricting the rank of the pairwise assignment matrix  $\tilde{\mathcal{P}}$  to obtain a consistent multi-partite matching  $\mathcal{P}$ .

Adjacency matrices of isomorphic graphs have identical eigenvalues and eigenvectors, only their assignment differs. Following the approach of Umeyama [137], we identify the correspondence of eigenvectors  $\mathbf{U}$  by projecting both sets  $\mathbf{U}_G$  and  $\mathbf{U}_H$  onto each other to find the best overlap. We use the overlap matrix,

$$\mathbf{M} = \text{abs}(\mathbf{U}_G)\text{abs}(\mathbf{U}_H)^\top \quad (4.2)$$

after sorting eigenvalues and overcoming sign ambiguity. Then  $-\mathbf{M}$  is provided as the cost matrix for the Hungarian algorithm [138], maximizing the overall overlap which finally returns the approximate assignment  $\tilde{\tau}$  that minimizes Eq. 4.1 and thus provides the results of step one of the procedure (see Appendix A.2.2). As indicated, global inconsistencies may arise, observable as violations of the transitivity property  $\tau_{jk} \circ \tau_{ij} = \tau_{ik}$  of the assignments [133]. Therefore a second step is necessary which is based on the composite matrix  $\tilde{\mathcal{P}}$  of all pairwise assignment matrices  $\tilde{\mathbf{P}}_{ij} \equiv \mathbf{P}(\tilde{\tau}_{ij})$  within the training set.

We propose to reconstruct a rank-limited  $\mathcal{P}$  via the transitive closure of the minimum spanning tree (MST) that minimizes the bi-partite matching cost (see Eq. 4.1, Figure 4.1) over the training set. The MST is constructed from the most confident bi-partite assignments and represents the rank  $N$  skeleton of  $\tilde{\mathcal{P}}$ , defining also  $\mathcal{P}$  (see Figure 4.2). Finally,

**Table 4.1** Recovering the permutation-inversion (PI) group of symmetry operations of fluxional molecules from short MD trajectories. We used our multi-partite matching algorithm to recover the symmetries of the molecules used in Longuet-Higgins [5]. Our algorithm identifies PI group symmetries (a superset that also includes the PG), as well as additional symmetries that are an artifact of the metric used to compare molecular graphs in our matching algorithm. Each dataset consists of a MD trajectory of 5000 time steps.

Molecule	PG order	PI group order	Recovered
	2	8	8
	6	6	6
	2	12	12
	6	12	12
	2	324	339
	6	36	39
	6	16	36

the resulting *multi-partite matching*  $\mathcal{P}$  is a consistent set of atom assignments across the whole training set.

As a first test, we apply our algorithm to a diverse set of non-rigid molecules that have been selected by Longuet-Higgins [5] to illustrate the concept of dynamic symmetries. Each of the chosen examples changes easily from one conformation to another due to internal rotations that can not be described by point groups. Those molecules require the more complete *permutation-inversion group* of symmetry operations that include energetically feasible permutations of identical nuclei. Our multi-partite matching algorithm successfully recovers those symmetries from short MD trajectories (set Table 4.1), giving us the confidence to proceed.

### 4.1.2 Symmetric kernels

The resulting consistent multi-partite matching  $\mathcal{P}$  enables us to construct symmetric kernel-based ML models of the form

$$\hat{f}(\mathbf{x}) = \sum_{ij}^M \alpha_{ij} k(\mathbf{x}, \mathbf{P}_{ij} \mathbf{x}_i), \quad (4.3)$$

by augmenting the training set with the symmetric variations of each molecule [139]. A particular advantage of our solution is that it can fully populate all recovered permuta-

tional configurations even if they do not form a symmetric group, severely reducing the computational effort in evaluating the model. Even if we limit the range of  $j$  to include all  $S$  unique assignments only, the major downside of this approach is that a multiplication of the training set size leads to a drastic increase in the complexity of the cubically scaling GP regression algorithm. We overcome this drawback by exploiting the fact that the set of coefficients  $\alpha$  for the symmetrized training set exhibits the same symmetries as the data, hence the linear system can be contracted to its original size, while still defining the full set of coefficients exactly.

Without affecting the pairwise similarities expressed by the kernel, we transform all training geometries into a canonical permutation  $\mathbf{x}_i \equiv \mathbf{P}_{i1}\mathbf{x}_i$ , enabling the use of uniform symmetry transformations  $\mathbf{P}_j \equiv \mathbf{P}_{1j}$ . Simplifying Eq. 4.3 accordingly, gives rise to the symmetric kernel that we originally set off to construct

$$\begin{aligned}\hat{f}(\mathbf{x}) &= \sum_i^M \alpha_i \sum_q^S k(\mathbf{x}, \mathbf{P}_q \mathbf{x}_i) \\ &= \sum_i^M \alpha_i k_{\text{sym}}(\mathbf{x}, \mathbf{x}_i),\end{aligned}\tag{4.4}$$

and yields a model with the exact same number of parameters as the original, non-symmetric one. This ansatz is known as *invariant integration* and frequently applied in ML potentials [140, 29, 22]. However, our solution, motivated by the concept of permutation-inversion groups [5], is able to truncate the sum over potentially hundreds of thousands permutations in the full symmetric group of the molecule to a few physically reasonable ones. We remark that this step is essential in making invariant integration practical beyond systems with five or six identical atoms (with  $5! = 120$  and  $6! = 720$  permutations, respectively). The largest permutation sets recovered from the datasets considered here have cardinality 12, whereas the associated symmetric groups have orders  $6!6!$ ,  $7!8!$  and  $12!10!2!$ , for benzene, toluene and azobenzene respectively (see Table 4.2). Our multi-partite matching algorithm is therefore able to shorten the sum over  $S$  in Eq. 4.4 by up to 15 orders of magnitude, without significant loss of accuracy.

## 4.2 Symmetric gradient domain learning (sGDML)

Our symmetric kernel is an extension to regular kernels and can be applied universally, in particular to kernel based force fields. Here, we construct a symmetric variant of the gradient domain learning (GDML) model, sGDML. This symmetrized GDML force field

**Table 4.2** Relative increase in accuracy of the sGDML@DFT vs. the non-symmetric GDML model: the benefit of a symmetric model is directly linked to the number of permutational symmetries in the system. All symmetry counts include the identity transformation.

<b>Molecule</b>	# Sym. in $k_{\text{sym}}$	$\Delta \text{MAE} [\%]$	
		Energy	Forces
Benzene	12	-1.6	-62.3
Uracil	1	0.0	0.0
Naphthalene	4	0.0	-52.2
Aspirin	6	-29.6	-31.3
Salicylic acid	1	0.0	0.0
Malonaldehyde	4	-37.5	-48.8
Ethanol	6	-53.4	-58.2
Toluene	12	-16.7	-67.4
Paracetamol	12	-40.7	-52.9
Azobenzene	8	-74.3	-47.4

kernel takes the form:

$$\text{Hess}(k_{\text{sym}})(\mathbf{x}, \mathbf{x}') = \sum_q^S \text{Hess}(k)(\mathbf{x}, \mathbf{P}_q \mathbf{x}') \mathbf{P}_q. \quad (4.5)$$

Accordingly, the trained force field estimator collects the contributions of the  $3N$  partial derivatives of all training points  $M$  and number of symmetry transformations  $S$  to compile the prediction. It takes the form

$$\hat{\mathbf{f}}_F(\mathbf{x}) = \sum_i^M \sum_l^{3N} \sum_q^S (\mathbf{P}_q \alpha_i)_l \frac{\partial}{\partial x_l} \nabla k(\mathbf{x}, \mathbf{P}_q \mathbf{x}_i) \quad (4.6)$$

and a corresponding energy predictor is obtained by integrating  $\hat{\mathbf{f}}_F$  with respect to the Cartesian geometry as in Eq. 3.20. Due to linearity of integration, the expression for the energy predictor is again identical up to second derivative operator on the kernel function.

### 4.2.1 Training

To construct the covariance matrix for training the sGDML model, the following formulation is used:

$$\text{Hess}(k_{\text{sym}})(\mathbf{x}, \mathbf{x}') = \frac{1}{S} \sum_{pq}^S \mathbf{P}_p^\top \text{Hess}(k)(\mathbf{P}_p \mathbf{x}, \mathbf{P}_q \mathbf{x}') \mathbf{P}_q. \quad (4.7)$$

Unlike in Eq. 4.5, both inputs to the kernel function are symmetrized here (using  $\mathbf{P}_p^\top$  and  $\mathbf{P}_q$ ) to obtain a matrix  $\mathbf{K} = K^\top$ . A permutation of the first argument of the kernel function furthermore requires an additional normalization factor.

#### 4.2.2 Descriptors

For notational convenience, we have described the formulation of the sGDML model for generic inputs  $\mathbf{x}$  up until now. When the input to the force field kernel function is a descriptor, the symmetric (training) kernel matrix evaluates to

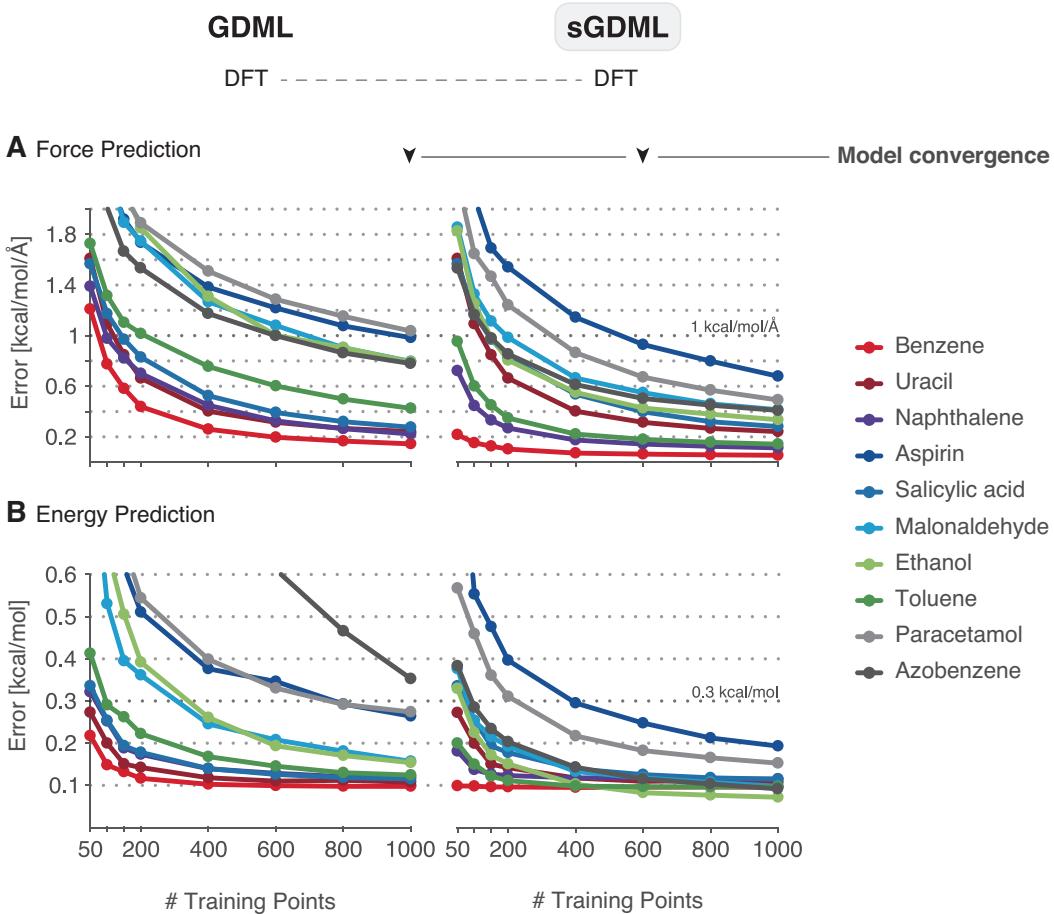
$$\begin{aligned} \text{Hess}(k_{\text{sym}})(\mathbf{D}(\mathbf{x}), \mathbf{D}(\mathbf{x}')) = \\ \frac{1}{S} \sum_{pq}^S (\mathbf{J}_{\mathbf{D}}(\mathbf{P}_p \mathbf{x}) \mathbf{P}_p)^\top \text{Hess}(k)(\mathbf{D}(\mathbf{P}_p \mathbf{x}), \mathbf{D}(\mathbf{P}_q \mathbf{x}')) \mathbf{J}_{\mathbf{D}}(\mathbf{P}_q \mathbf{x}) \mathbf{P}_q \end{aligned} \quad (4.8)$$

after application of the chain rule, where  $\mathbf{J}_{\mathbf{D}}$  is the Jacobian of the descriptor (see Section 3.2.3).

### 4.3 Numerical experiments

Every (s)GDML model is trained on a set of reference examples that reflects the population of energy states a particular molecule visits during an MD simulation at a certain temperature. For our purposes, the corresponding set of geometries is subsampled from a 200 picosecond DFT MD trajectory at 500 K following the Boltzmann distribution. Subsequently, a globally consistent permutation graph is constructed that jointly assigns all geometries in the training set, providing a small selection of physically feasible transformations that define the training set specific symmetric kernel function. In the interest of computational tractability, we shortcut this sampling process to construct sGDML@CCSD(T) and only recompute energy and force labels at this higher level of theory (see Figure 4.4).

The sGDML model can be trained in closed form, which is both quicker and more accurate than numerical solutions. Model selection is performed through a grid search on a suitable subset of the hyper-parameter space. Throughout, cross-validation with dedicated datasets for training, testing and validation are used to estimate the generalization performance of the model.



**Figure 4.3** Data efficiency gains using sGDML versus GDML. Energy and force prediction accuracy (in terms of the mean absolute error (MAE)) as a function of training set size of both models trained on DFT forces: the gain in efficiency and accuracy is directly linked to the number of symmetries in the system.

#### 4.3.1 Datasets

The data used for training the DFT models were created running *ab initio* MD in the NVT ensemble using the Nosé-Hoover thermostat at 500 K. The simulation duration was 200 ps, sampled at a resolution of 0.5 fs. We computed forces and energies using all-electrons at the generalized gradient approximation (GGA) level of theory with the Perdew-Burke-Ernzerhof (PBE) [122] exchange-correlation functional, treating van der Waals interactions with the Tkatchenko-Scheffler (TS) method [123]. All calculations were performed with FHI-aims [141]. The final training data was generated by subsampling the full trajectory under preservation of the Maxwell-Boltzmann distribution for the energies.

To create the coupled cluster datasets, we reused the same geometries as for the DFT models and recomputed energies and forces using all-electron coupled cluster with

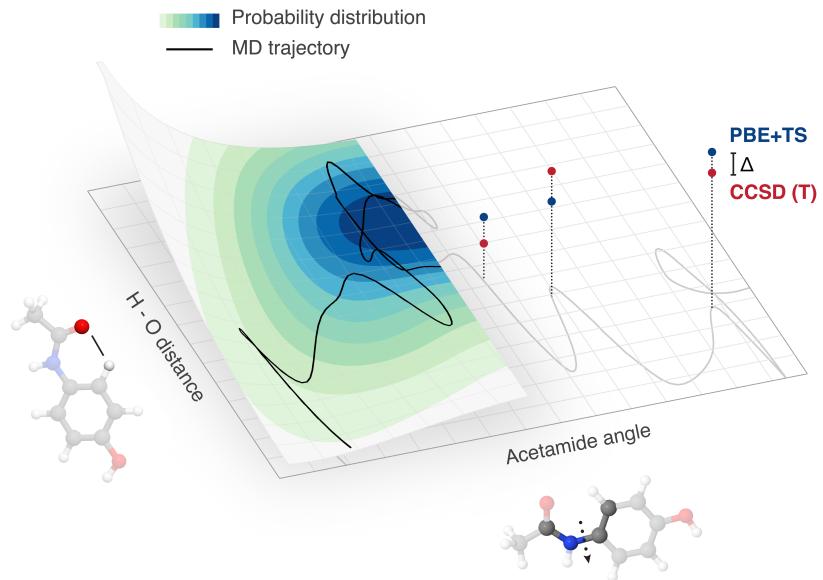
single, double, and perturbative triple excitations (CCSD(T)). The Dunning's correlation-consistent basis set cc-pVTZ was used for ethanol, cc-pVDZ for toluene and malonaldehyde and CCSD/cc-pVDZ for aspirin. All calculations were performed with the Psi4 [142, 143] software suite.

### 4.3.2 Forces and energies from GDML to sGDML@DFT to sGDML@CCSD(T)

**Table 4.3** Prediction accuracy for interatomic forces and total energies of the sGDML@DFT on all datasets. Energy errors are in  $\text{kcal mol}^{-1}$ , force errors in  $\text{kcal mol}^{-1}\text{\AA}^{-1}$ . Each model is trained on 1000 geometries with corresponding force labels.

Dataset	Energy error		Force error		Magnitude		Angle	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
	0.10	0.12	0.06	0.09	0.06	0.09	0.0009	0.0017
Benzene	0.11	0.14	0.24	0.37	0.22	0.31	0.0039	0.0064
Uracil	0.12	0.15	0.11	0.17	0.11	0.15	0.0016	0.0026
Naphthalene	0.19	0.25	0.68	0.96	0.52	0.68	0.0094	0.0139
Aspirin	0.12	0.15	0.28	0.44	0.32	0.45	0.0038	0.0064
Salicylic acid	0.10	0.13	0.41	0.62	0.39	0.56	0.0055	0.0087
Malonaldehyde	0.07	0.09	0.33	0.49	0.46	0.63	0.0051	0.0083
Ethanol	0.10	0.12	0.14	0.21	0.14	0.19	0.0020	0.0031
Toluene	0.15	0.20	0.49	0.70	0.60	0.84	0.0073	0.0118
Paracetamol	0.09	0.13	0.41	0.61	0.49	0.71	0.0059	0.0105
Azobenzene								

Our goal is to demonstrate that it is possible to construct compact sGDML models that faithfully recover CCSD(T) force fields for flexible molecules with up to 20 atoms, by using only a small set of few hundred molecular conformations. As a first step, we investigate the gain in efficiency and accuracy of sGDML model vs. GDML model employing MD trajectories of ten molecules from benzene to azobenzene computed with DFT (see Figure 4.3 and Table 4.3). Unsurprisingly, the benefit of a symmetric model is directly linked to the number of symmetries in the system. For toluene, naphthalene, aspirin, malonaldehyde, ethanol, paracetamol and azobenzene, sGDML improves the force prediction by 31.3% to 67.4% using the same training set in all cases (see Table 4.2). As expected, uracil and salicylic acid have no exploitable symmetries, hence the performance of sGDML is unchanged with respect to GDML. The inclusion of symmetries leads to a



**Figure 4.4** Reference data generation: Geometries are sampled from a sufficiently long, but cheap DFT-PBE+TS MD trajectory to ensure optimal coverage of the configuration space. Energy and force labels for this small subset of the trajectory are then recomputed at the higher CCSD(T) level of theory and used for training the sGDML model. The full PES will be reconstructed at the accuracy of the CCSD(T) reference data.

stronger improvement in force prediction performance compared to energy predictions. This is most clearly visible for the naphthalene dataset, where the force predictions even improve unilaterally. We attribute this to the difference in complexity of both quantities and the fact that an energy penalty is intentionally omitted in the cost function to avoid a tradeoff.

A minimal force accuracy required for reliable MD simulations is  $\text{MAE} = 1 \text{ kcal mol}^{-1}\text{\AA}^{-1}$ . While the GDML model can achieve this accuracy at around 800 training examples for all molecules except aspirin, sGDML only needs 200 training examples to reach the same quality. Note that energy-based ML approaches typically require two to three orders of magnitude more data [144].

Given that the novel sGDML model is data efficient and highly accurate, we are now in position to tackle CCSD(T) level of accuracy with modest computational resources. We have trained sGDML models on CCSD(T) forces for benzene, toluene, ethanol, and malonaldehyde. For the larger aspirin molecule, we used CCSD forces (see Table 4.4). The sGDML@CCSD(T) model achieves a high accuracy for energies, reducing the prediction error of sGDML@DFT by a factor of 1.4 (for ethanol) to 3.4 (for toluene). This finding leads to an interesting hypothesis that sophisticated quantum-mechanical force fields are smoother and, as a convenient side effect, easier to learn. Note that the accuracy of

**Table 4.4** Prediction accuracy for interatomic forces and total energies of the sGDML@CCSD(T) model on all datasets. Energy errors are in kcal mol<sup>-1</sup>, force errors in kcal mol<sup>-1</sup>Å<sup>-1</sup>.

Dataset	Energy error		Force error		Magnitude		Angle	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
					MAE	RMSE	MAE	RMSE
Benzene	0.004	0.005	0.04	0.06	0.04	0.06	0.0008	0.0013
Aspirin*	0.16	0.21	0.76	1.07	0.56	0.74	0.0091	0.0123
Malonaldehyde	0.06	0.08	0.37	0.56	0.34	0.46	0.0052	0.0082
Ethanol	0.05	0.07	0.35	0.51	0.47	0.65	0.0056	0.0104
Toluene	0.03	0.04	0.21	0.30	0.19	0.24	0.0028	0.0042

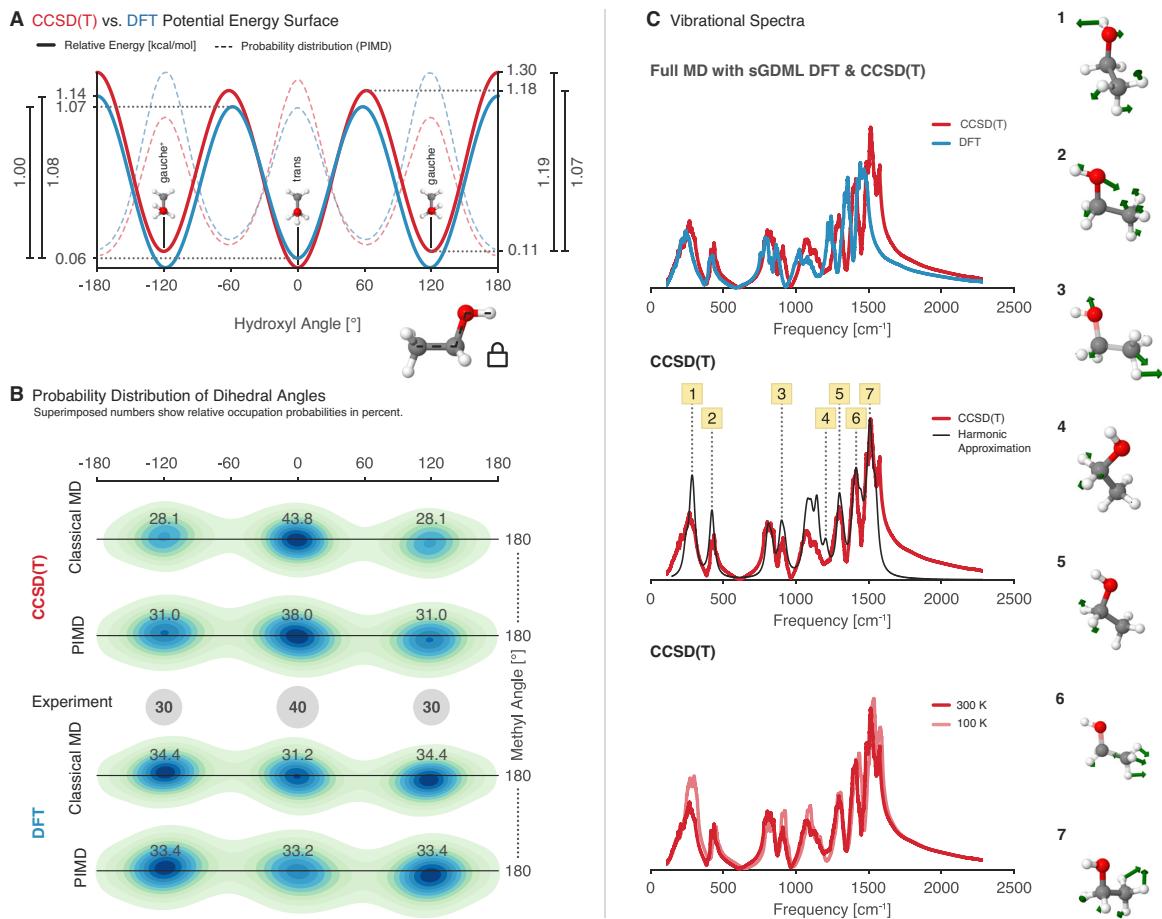
\* CCSD

the force prediction in both sGDML@CCSD(T) and sGDML@DFT is comparable, with the benzene molecule as the only exception. We attribute this aspect to slight shifts in the locations of the minima on the PES between DFT and CCSD(T), which means that the data sampling process for CCSD(T) can be further improved.

### 4.3.3 Molecular dynamics with *ab initio* accuracy

The predictive power of a force field can only be truly assessed by computing dynamical and thermodynamical observables, which require sufficient sampling of the configuration space, for example by employing molecular dynamics or Monte Carlo simulations. We remark that global error measures, such as mean average error (MAE) and root mean squared error (RMSE) are typically prone to overestimate the reconstruction quality of the force field, as they average out local topographical properties. However, these local properties can become highly relevant when the model is used for an actual analysis of MD trajectories. As a demonstration, we will use the ethanol molecule; this molecule has three minima, *gauche*± ( $M_{g\pm}$ ) and *trans* ( $M_t$ ) shown in Figure 4.5-A, where experimentally it has been confirmed that  $M_t$  is the ground state and  $M_g$  is a local minimum [145]. The energy difference between these two minima is only 0.12 kcal mol<sup>-1</sup> and they are separated by an energy barrier of 1.15 kcal mol<sup>-1</sup>. Obviously, the widely discussed ML target accuracy of 1 kcal mol<sup>-1</sup> is not sufficient to describe the dynamics of ethanol and other molecules.

This brings us to another crucial issue for predictive models: the reference data accuracy. Computing the energy difference between  $M_t$  and  $M_g$  using DFT(PBE-TS) we observe that  $M_g$  is 0.08 kcal mol<sup>-1</sup> more stable than  $M_t$ , contradicting the experimental measurements. Repeating the same calculation using CCSD(T)/cc-pVTZ we find that  $M_t$



**Figure 4.5** Molecular dynamics simulations for ethanol. (A) Potential energy profile of the dihedral angle describing the rotation of the hydroxyl group for CCSD(T) (red) vs. DFT (blue). The energetic barriers predicted by sGDML@CCSD(T) are:  $M_t \rightarrow M_g$ : 1.18 kcal mol<sup>-1</sup>,  $M_g \rightarrow M_{g\cdot}$ : 1.19 kcal mol<sup>-1</sup>, and  $M_g \rightarrow M_t$ : 1.07 kcal mol<sup>-1</sup>. The dashed lines show the probability distributions obtained from PIMD at 300K. (B) Joint probability distribution function for the two dihedral angles of the methyl and hydroxyl functional groups. Each minimum is annotated with the occupation probability obtained from classical and path-integral MD in comparison with experimental values. (C) Analysis of vibrational spectra (velocity–velocity autocorrelation function). (top) Comparison between the vibrational spectrum obtained from PIMD simulations at 300K for sGDML@CCSD(T) and its sGDML@DFT counterpart; (middle) comparison between the sGDML@CCSD(T) PIMD spectrum and the harmonic approximation based on CCSD(T) frequencies; (bottom) comparison of sGDML@CCSD(T) PIMD spectra at 300K and 100K. The rightmost panel shows several characteristic normal modes of ethanol, where atomic displacements are illustrated by green arrows.

is more stable than  $M_g$  by 0.08 kcal mol<sup>-1</sup>, in excellent agreement with experiment. From this analysis and subsequent MD simulations we conclude that CCSD(T) or sometimes even higher accuracy is necessary for truly predictive insights.

Additionally to requiring highly accurate quantum chemical approximations, the ethanol molecule also belongs to a category of fluxional molecules sensitive to nuclear quantum effects (NQE). This is because internal rotational barriers of the ethanol molecule ( $M_g \leftrightarrow M_t$ ) are on the order of  $\sim 1.2$  kcal mol $^{-1}$  (see Figure 4.5), which is neither low enough to generate frequent transitions nor high enough to avoid them. In a classical MD at room temperature the thermal fluctuations lead to inadequate sampling of the PES. By correctly including NQE via path-integral molecular dynamics (PIMD), the ethanol molecule is able to transition between  $M_g$  and  $M_t$  configurations, radically increasing the transition frequency (see Figure B.1) and generating statistical weights in excellent agreement with experiment. Figure 4.5-B shows the statistical occupations of the different minima for ethanol using classical MD and PIMD for the sGDML@CCSD(T) and sGDML@DFT models in comparison with the experimental results. Overall, our MD results for ethanol highlight the necessity of using a highly accurate force field with an equally accurate treatment of NQE for achieving reliable and quantitative understanding of molecular systems.

#### 4.3.4 CCSD(T)-level vibrational spectra

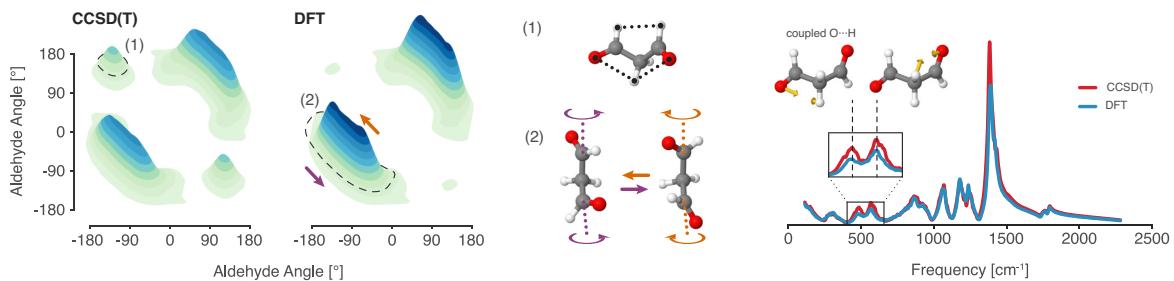
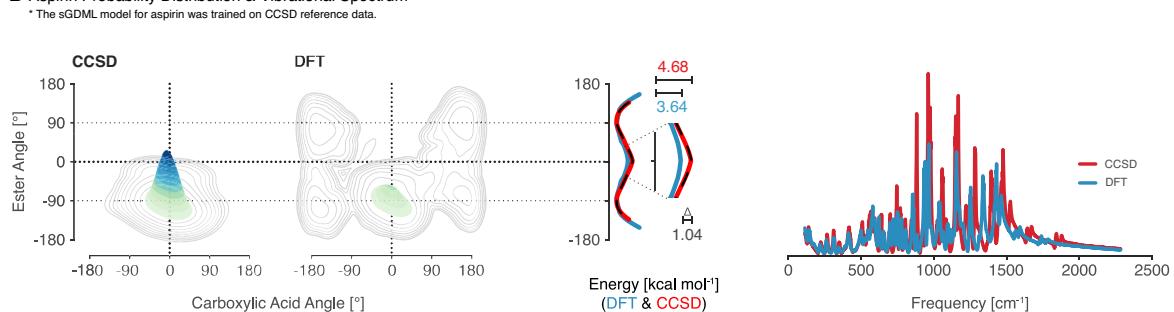
Having established the accuracy of statistical occupations of different states of ethanol, we are now in position to discuss for the first time the CCSD(T) vibrational spectrum of ethanol computed using the velocity–velocity autocorrelation function based on centroid PIMD (see Figure 4.5-C). As a reference, in Figure 4.5-C-top we compare the vibrational spectra from DFT and CCSD(T) sGDML models in the fingerprint zone, and as expected the sGDML@CCSD(T) model generates higher frequencies but both share similar shapes but slightly different peak intensities. Molecular vibrational spectra at finite temperature include anharmonic effects, hence anharmonicities can be studied by comparing the sGDML@CCSD(T) spectrum with the harmonic approximation. Figure 4.5-C-middle shows such comparison and demonstrates that low-frequency and non-symmetric vibrations are most affected by finite-temperature contributions. The thermal frequency shift can be better seen in Figure 4.5-C-bottom, where the sGDML@CCSD(T) spectrum is compared at two different temperatures. We observe that each normal mode is shifted in a specific manner and not by a simple scaling factor, as typically assumed. The most striking finding from our simulations is the resolution of the apparent mismatch between theory and experiment explaining the origin of the torsional frequency for the hydroxyl group. Experimentally, the low frequency region of ethanol, around  $\sim 210$  cm $^{-1}$ , is not fully understood, but there are frequency measurements for the hydroxyl rotor ranging in between  $\sim 202$  [146, 147] and  $\sim 207$  [148] cm $^{-1}$  for gas-phase ethanol, while theoretically we found 243.7 cm $^{-1}$  at the sGDML@CCSD(T) level of theory in the harmonic approxima-

tion. From the middle and bottom panels in Figure 4.5-C, we observe that by increasing the temperature the lowest peak shifts to substantially lower frequencies compared to the rest of the spectrum. The origin of such phenomena is the strong anharmonic behavior of the lowest normal mode  $a$ , shown in Figure 4.5-C-middle, which mainly corresponds to hydroxyl group rotations. At room temperature the frequency of this mode drops to  $\sim 215\text{ cm}^{-1}$ , corresponding to a red-shift of 12% and getting closer to the experimental results demonstrating the importance of dynamical anharmonicities.

#### 4.3.5 Probability distributions CCSD(T) vs. DFT

Finally, we illustrate the wider applicability of the sGDML model to more complex molecules than ethanol by performing a detailed analysis of MD simulations for malonaldehyde and aspirin. In Figure 4.6-A, we show the joint probability distributions of the dihedral angles (PDDA) for the malonaldehyde molecule. This molecule has a peculiar PES with two local minima with a O $\cdots$ H $\cdots$ O symmetric interaction (structure (1)), and a shallow region where the molecule fluctuates between two symmetric global minima (structure (2)). The dynamical behavior represented in structure (2) is due to the interplay of two molecular states dominated by an intramolecular O $\cdots$ H interaction and a low crossing barrier of  $\sim 0.2\text{ kcal mol}^{-1}$ . An interesting result is the nearly unvisited structure (1) by sGDML@DFT in comparison to sGDML@CCSD(T) model regardless of the great similarities of their PES, which gives an idea of the observable consequences of subtle energy differences in the PES of molecules with several degrees of freedom. In terms of spectroscopic differences, the two approximations generate spectra with very few differences (Figure 4.6-A-right), but being the most prominent the one between the two peaks around  $500\text{ cm}^{-1}$ . Such difference can be traced back to the enhanced sampling of the structure (1), and additionally it could be associated to the different nature between the methods in describing the intramolecular O $\cdots$ H coupling.

For aspirin, the consequences of proper inclusion of the electron correlation are even more significant. Figure 4.6-B shows the PIMD generated PDDA for DFT and CCSD based models. By comparing the two distributions we find that sGDML@CCSD generates localized dynamics in the global energy minimum, whereas the DFT model yields a rather delocalized sampling of the PES. These two contrasting results are explained by the difference in the energetic barriers along the ester dihedral angle. The incorporation of electron correlation in CCSD increases the internal barriers by  $\sim 1\text{ kcal mol}^{-1}$ . This prediction was corroborated with explicit CCSD(T) calculations along the dihedral-angle coordinate (black dashed line in Figure 4.6-B-PES). Furthermore, the difference in the sampling is also due to the fact that the DFT model generates consistently softer interatomic interactions

**A** Malonaldehyde Probability Distribution & Vibrational Spectrum**B** Aspirin Probability Distribution & Vibrational Spectrum

**Figure 4.6** Analysis of MD simulations with sGDML for malonaldehyde and aspirin. The MD simulations at 300 K were carried out for 500 ps. (A) Joint probability distributions of the dihedral angles in malonaldehyde, describing the rotation of both aldehyde groups based on classical MD simulations for sGDML@CCSD(T) and sGDML@DFT. The configurations (1) and (2) are representative structures of the most sampled regions of the PES. (B) Joint probability distributions of the dihedral angles in aspirin, describing the rotation of the ester and carboxylic acid groups based on PIMD simulations for sGDML@CCSD and sGDML@DFT using 16 beads at 300 K. The potential energy profile for the ester angle in  $\text{kcal mol}^{-1}$  is shown for sGDML@CCSD (red), sGDML@DFT (blue) and compared with the CCSD reference (black, dashed). Contour lines show the differences of both distributions on a log scale. Both panels also show a comparison of the vibrational spectra generated via the velocity-velocity autocorrelation function obtained with sGDML@CCSD(T)/CCSD (red) and sGDML@DFT (blue).

compared to CCSD, which leads to large and visible differences in the vibrational spectra between DFT and CCSD (Figure 4.6-B-right).

### 4.3.6 Symmetry compression

By construction, the symmetric model in Eq. 4.6 is invariant to all permutational transformations of the molecular geometry that are represented in the training set. Swapping two symmetric atoms in the input yields the exact same atomic force predictions as before. This rises the question, whether we can remove the redundant degrees of freedom from the model in order to simplify it? Formally, the idea is to replace the symmetric predictor

$\hat{\mathbf{f}}: \mathcal{X}^N \rightarrow \mathbb{R}^N$  with a smaller model  $\hat{\mathbf{f}}_{|N}: \mathcal{X}^O \rightarrow \mathbb{R}^O$ , where  $O < N$ . The output for any symmetric degree of freedom can then be obtained by repeatedly evaluating the *compressed* model for the omitted dimensions. Let  $\hat{\mathbf{f}}$  be symmetric in its first two arguments such that

$$\hat{\mathbf{f}}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) = \hat{\mathbf{f}}(\mathbf{x}_2, \mathbf{x}_1, \dots, \mathbf{x}_N), \quad (4.9)$$

then the reduced model with  $O = N - 1$  interacting atoms takes the form

$$\hat{\mathbf{f}}_{|N}(\mathbf{x}_1, \mathbf{x}_3, \dots, \mathbf{x}_N) = \hat{\mathbf{f}}_{|N}(\mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_N), \quad (4.10)$$

from which we recover the full prediction as

$$\hat{\mathbf{f}}(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N) = [(\hat{\mathbf{f}}_{|N}(\mathbf{x}_1, \mathbf{x}_3, \dots, \mathbf{x}_N))_1, \hat{\mathbf{f}}_{|N}(\mathbf{x}_2, \mathbf{x}_3, \dots, \mathbf{x}_N)]^\top. \quad (4.11)$$

Of course, this principle trivially generalizes to permutation groups of arbitrary order. In the sGML model, a dimensionality reduction in input space directly translates to a reduction of the kernel matrix size from  $3NM$  to  $3OM$ , making this idea especially compelling as it drastically reduces training time. Redundant arguments are identified by examining the set of associated index assignments  $\{\tau(i)_s\}_{i \in N, s \in S}$ . Here,  $\tau(i)$  is the permutation in tuple notation that returns the new index for atom  $i$ , and  $S$  denotes the recovered permutation set. Arguments with identical index assignments are interchangeable.

To construct the compressed sGML model  $\hat{\mathbf{f}}_{|N}$ , we simply remove the rows and column in the force field kernel function that correspond with the those input dimensions. This can be accomplished elegantly, by only removing the respective rows in the descriptor Jacobian  $\mathbf{J}_D \in \mathbb{R}^{3N \times \dim(\mathcal{I})}$  for the *training* data. By leaving the descriptor Jacobian of the *input* untouched during inference, the model will still return forces and energy for the full degrees of freedom of the molecule.

## Numerical results

The usefulness of the symmetry compression approach hinges on how adversely it affects prediction accuracy. Given a fixed set of training points, we do expect a degradation due to the reduced complexity of the model, which we will investigate in the first part of this analysis. The more interesting question is, whether we can gain accuracy by using symmetry compression to keep the complexity of the training task (e.g. the size of the kernel matrix) constant, while increasing the number of training points. One important thing to keep in mind during this analysis is that the compression ratio (i.e. the number of removed atoms) is different for each dataset. Highly symmetric molecules can be

**Table 4.5** Prediction accuracy for interatomic forces and total energies using the original sGDML model and a compressed variant  $s\text{GDML}_{\downarrow N}$  that only considers the non-symmetric atomic degrees of freedom  $\downarrow N$ . Both model types have been trained on 1000 data points. The best result for each dataset is highlighted by bold face.

Dataset	$N \downarrow N$	Energy error [ $\text{kcal mol}^{-1}$ ]				Force error [ $\text{kcal mol}^{-1}\text{\AA}^{-1}$ ]				
		sGDML		$s\text{GDML}_{\downarrow N}$		sGDML		$s\text{GDML}_{\downarrow N}$		
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	
Benzene	12	2	<b>0.10</b>	<b>0.12</b>	<b>0.10</b>	<b>0.12</b>	<b>0.06</b>	<b>0.09</b>	0.11	0.17
Uracil	12	—	0.11	0.14	—	—	0.24	0.37	—	—
Naphthalene	18	5	<b>0.12</b>	<b>0.15</b>	<b>0.12</b>	<b>0.15</b>	<b>0.11</b>	<b>0.17</b>	0.21	0.31
Aspirin	21	19	<b>0.19</b>	<b>0.25</b>	0.21	0.27	<b>0.68</b>	<b>0.96</b>	0.70	1.00
Salicylic acid	16	—	0.12	0.15	—	—	0.28	0.44	—	—
Malonaldehy.	9	5	<b>0.10</b>	<b>0.13</b>	0.12	0.16	<b>0.41</b>	<b>0.62</b>	0.50	0.74
Ethanol	9	6	<b>0.07</b>	<b>0.09</b>	0.08	0.10	<b>0.33</b>	<b>0.49</b>	0.37	0.54
Toluene	15	9	<b>0.10</b>	<b>0.12</b>	<b>0.10</b>	<b>0.12</b>	<b>0.14</b>	<b>0.21</b>	0.17	0.25
Paracetamol	20	14	<b>0.15</b>	<b>0.20</b>	0.17	0.22	<b>0.49</b>	<b>0.70</b>	0.59	0.83
Azobenzene	24	8	<b>0.09</b>	<b>0.13</b>	0.21	0.27	<b>0.41</b>	<b>0.61</b>	0.69	0.98

summarized with only a few degrees of freedom, whereas non-symmetric ones like uracil and salicylic acid are not compressible at all.

First, we investigate the degradation of accuracy as a result of symmetry compression. For this test, we keep the training set size fixed at 1000 examples like in our previous experiments and compare force and energy prediction performance of the compressed model  $s\text{GDML}_{\downarrow N}$  to the unmodified sGDML model. Table 4.5 shows the results for each of the ten DFT datasets. We observe, that the energy prediction performance is largely unaffected by symmetry compression (maximum MAE degradation:  $0.02 \text{ kcal mol}^{-1}$ ), except for azobenzene (MAE degradation:  $0.12 \text{ kcal mol}^{-1}$ ). The three best converged models according to the learning curves in Figure 4.3, (namely benzene, naphthalene and toluene) even show unchanged energy prediction accuracy, despite a drastic reduction in kernel matrix size. The biggest compression ratio is possible for the highly symmetric benzene molecule, where the effective degrees of freedom reduce to one sixth, from 12 atoms to only 2. Remarkably, the energy prediction accuracy of  $s\text{GDML}_{\downarrow N}$  does not degrade at all. The second best compression ratio (18 : 5) is possible for naphthalene, also without negatively affecting energy prediction accuracy. The effect of symmetry compression is more pronounced in the force predictions. Here we observe a significant difference between absolute and relative degradation of the prediction accuracy between

the datasets. Unsurprisingly, the level of absolute degradation mostly aligns with how well each model is converged. Once again, benzene, naphthalene and toluene show the mildest absolute degradation of the force MAE ( $0.06, 0.1, 0.08 \text{ kcal mol}^{-1}\text{\AA}^{-1}$ , respectively), whereas azobenzene degrades the most, by  $0.29 \text{ kcal mol}^{-1}\text{\AA}^{-1}$ . An analysis of the relative degradation however shows that benzene and naphthalene experienced the strongest increase in force prediction error, by 183% and 191%, respectively. We observe the smallest relative degradation for aspirin, ethanol and paracetamol, however these molecules are also the ones with the lowest compression (ratios 21 : 19, 9 : 6 and 20 : 14, respectively). Overall, benzene, naphthalene and azobenzene show the best ratio of compression to error increase. Here, the accuracy penalty due to symmetry compression is particularly low, compared to the reduction in kernel matrix size.

The fact that energy prediction performance is essentially unaffected by a reduction in model complexity, further reinforces our assumption that achieving a good prediction performance for the energy is significantly easier than predicting accurate forces. These results are consistent with our findings in the previous chapter, where we discovered that very accurate energy predictions are obtainable even with a basic model (see Appendix B.1). At the same time, a low energy error does not necessarily indicate a faithful reconstruction of the PES: despite similar energy prediction accuracies, the compressed model makes worse force predictions in comparison to the original sGDML model.

We will now increase the number of training points for each  $\text{sGDML}_{\downarrow N}$  model until it reaches the same kernel size as the uncompressed model. For highly symmetric molecules like benzene, naphthalene and azobenzene this results in a drastic increase in training set size, from the initial 1000 points to 6000, 3600 and 3000, respectively. We observe an improvement of the energy and force prediction performance for all datasets, except azobenzene, which only improves in force accuracy (see Table 4.6). Once again, there is barely any change in the energy prediction performances, as those are already converged for the smaller training set sizes. The improvements in force prediction accuracy are moderate, but indicative of a positive trend. Malonaldehyde and ethanol benefit the most with absolute force improvements by  $0.04$  and  $0.03 \text{ kcal mol}^{-1}\text{\AA}^{-1}$ , respectively. In general, the least converged models with highest force prediction error in the original model benefit the most from symmetry compression and training data backfill.

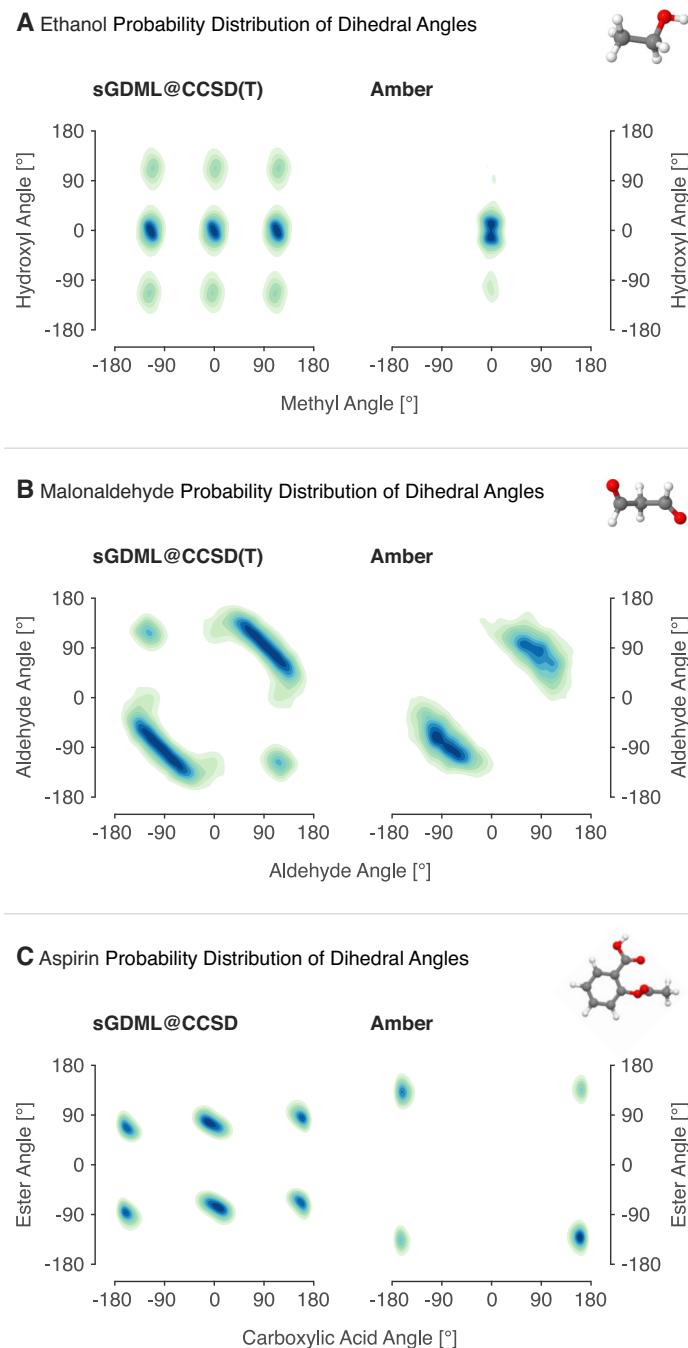
In summary, the explicit knowledge of symmetries can not only be exploited for data efficiency, but also to reduce the number of parameters and thus the complexity of the model. Our analysis shows that reducing the complexity of the sGDML learning problem via symmetry compression yields better performing models compared to simply reducing the number of training points to achieve the same effect.

**Table 4.6** Prediction accuracy for interatomic forces and total energies using the original sGDML model with a training set size of  $M = 1000$  and the compressed variant  $\text{sGDML}_{\downarrow N}$  with increased training set size  $\tilde{M}$  to match the complexity of the optimization problem during training. The best result for each dataset is highlighted by bold face.

Dataset	$\tilde{M}$	Energy error [kcal mol <sup>-1</sup> ]				Force error [kcal mol <sup>-1</sup> Å <sup>-1</sup> ]			
		sGDML		$\text{sGDML}_{\downarrow N}$		sGDML		$\text{sGDML}_{\downarrow N}$	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Benzene	6000	<b>0.10</b>	<b>0.12</b>	<b>0.10</b>	<b>0.12</b>	<b>0.06</b>	<b>0.09</b>	<b>0.06</b>	<b>0.09</b>
Uracil	–	0.11	0.14	–	–	0.24	0.37	–	–
Naphthalene	3600	<b>0.12</b>	<b>0.15</b>	<b>0.12</b>	<b>0.15</b>	<b>0.11</b>	<b>0.17</b>	<b>0.11</b>	<b>0.17</b>
Aspirin	1105	<b>0.19</b>	<b>0.25</b>	<b>0.19</b>	<b>0.25</b>	0.68	0.96	<b>0.66</b>	<b>0.93</b>
Salicylic acid	–	0.12	0.15	–	–	0.28	0.44	–	–
Malonaldehyde	1800	<b>0.10</b>	<b>0.13</b>	<b>0.10</b>	<b>0.13</b>	0.41	0.62	<b>0.37</b>	<b>0.56</b>
Ethanol	1500	<b>0.07</b>	<b>0.09</b>	<b>0.07</b>	<b>0.09</b>	0.33	0.49	<b>0.30</b>	<b>0.44</b>
Toluene	1667	<b>0.10</b>	<b>0.12</b>	<b>0.10</b>	<b>0.12</b>	<b>0.14</b>	0.21	<b>0.14</b>	<b>0.20</b>
Paracetamol	1429	<b>0.15</b>	<b>0.20</b>	<b>0.15</b>	<b>0.20</b>	0.49	0.70	<b>0.47</b>	<b>0.66</b>
Azobenzene	3000	<b>0.09</b>	<b>0.13</b>	0.18	0.23	<b>0.41</b>	<b>0.61</b>	<b>0.41</b>	<b>0.61</b>

## 4.4 Discussion

The present work enables molecular dynamics simulations of flexible molecules with up to a few dozen atoms with the accuracy of high-level *ab initio* quantum mechanics. Such simulations pave the way to computations of dynamical and thermodynamical properties of molecules with an essentially exact description of the underlying potential-energy surface. On the one hand, this is a required step towards molecular simulations with spectroscopic accuracy. On the other, our accurate and efficient sGDML model leads to unprecedented insights when interpreting the experimental vibrational spectra and dynamical behavior of molecules. The contrasting demands of accuracy and efficiency are satisfied by the sGDML model through a rigorous incorporation of physical symmetries (spatial, temporal, and local symmetries) into a gradient-domain machine learning approach. This is a significant improvement over symmetry adaption in traditional force fields and electronic-structure calculations, where usually only (global) point groups are considered. Global symmetries are increasingly less likely to occur with growing molecule size, providing diminishing returns. Local symmetries on the other hand are system size independent and preserved even when the molecule is fragmented for large-scale modeling.



**Figure 4.7** Accuracy of the sGDML model in comparison to a traditional force field. We contrast the dihedral angle probability distributions of ethanol, malonaldehyde and aspirin obtained from classical MD simulations at 300 K with sGDML (left column) versus the AMBER [3, 4] (right column) force field. The ethanol simulations were carried out at constant energy (NVE), whereas a constant temperature (NVT) was used for malonaldehyde and aspirin. (A) Ethanol: the coupling between the hydroxyl and methyl rotor is absent in AMBER. Moreover, the probability distribution shows an unphysical harmonic sampling at room temperature, revealing the oversimplified harmonic description of bonded interactions in that force field. (B, C) Malonaldehyde and aspirin: the formulation of the AMBER force field is dominated by Coulomb interactions, which can lead an incomplete description of the PES and even spurious global minima in the case of aspirin. The length of the simulations was 0.5 ns.

In many of the applications of machine learned force fields the target error is the chemical accuracy or thermochemical accuracy ( $1 \text{ kcal mol}^{-1}$ ), but this value was conceived in the sense of thermochemical experimental measurements, such as heats of formation or ionization potentials. Consequently, the accuracy in ML models for predicting the molecular PES should not be tied to this value. Here, we propose a framework for the accuracy in force fields which satisfy the stringent demands of molecular spectroscopists, being typically in the range of wavenumbers ( $\approx 0.03 \text{ kcal mol}^{-1}$ ). Reaching this accuracy will be one of the greatest challenges of ML-based force fields. We remark that energy differences between molecular conformers are often on the order of  $0.1\text{--}0.2 \text{ kcal mol}^{-1}$ , hence reaching spectroscopic accuracy in molecular simulations is needed to generate predictive results.

A comparable accuracy is not obtainable with traditional force fields (see Figure 4.7). In general, they miss most of the crucial quantum effects due to their rigid, handcrafted analytical form. For example, the absence of a term for electron lone pairs in AMBER leads to uncoupled rotors in ethanol. Furthermore the oversimplified harmonic description of bonded interactions generates an unphysical harmonic sampling at room temperature (see Figure 4.7-A). In the case of malonaldehyde (Figure 4.7-B), both distributions misleadingly resemble each other, however they emerge from different types of interactions. For AMBER, the dynamics are purely driven by Coulomb interactions, while the sampling with sGDML@CCSD(T) (structure (2) in Figure 4.6-A) is mostly guided by electron correlation effects. Lastly, a complete mismatch between the regular force field and sGDML is evident for aspirin (see Figure 4.7-C), where the interactions dominated by Coulomb forces generate a completely different PES with spurious global and local minima. It is worth mentioning, that the observed shortcomings of the AMBER force field can be addressed for a particular molecule, however only at the cost of losing generality and computational efficiency.

In the context of machine learning, our work connects to recent studies on the usage of invariance constraints for learning and representations in vision. In the human visual system and also in computer vision algorithms the incorporation of invariances such as translation, scaling and rotation of objects can in principle permit higher performance at more data efficiency [149]; learning theoretical bounds can furthermore show that the amount of data required is reduced by a factor: the number of parameters of the invariance transformation [150]. Interestingly, our study goes empirically beyond this factor, i.e. our gain in data efficiency is often more than two orders of magnitude when combining the invariances (physical symmetries). We speculate that our finding may

indicate that the learning problem itself may become less complex, i.e. that the underlying problem structure becomes significantly easier to represent.

## 4.5 Practical considerations

### 4.5.1 Hybrid loss functions

**Table 4.7** Prediction accuracy for interatomic forces and total energies using the original sGDML model and a variant sGDML+E that has been extended with additional energy constraints in the loss function. Both model types have been trained on 1000 data points. The sGDML+E model consistently overfits the energy constraints at the cost of force prediction accuracy. The best result for each dataset is highlighted by bold face.

Dataset	Energy error [kcal mol <sup>-1</sup> ]				Force error [kcal mol <sup>-1</sup> Å <sup>-1</sup> ]			
	sGDML		sGDML+E		sGDML		sGDML+E	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Benzene	0.10	0.12	<b>0.01</b>	<b>0.02</b>	<b>0.06</b>	<b>0.09</b>	0.29	0.39
Uracil	0.11	0.14	<b>0.04</b>	<b>0.06</b>	<b>0.24</b>	<b>0.37</b>	0.43	0.61
Naphthalene	0.12	0.15	<b>0.03</b>	<b>0.04</b>	<b>0.11</b>	<b>0.17</b>	0.30	0.40
Aspirin	0.19	<b>0.25</b>	<b>0.18</b>	<b>0.25</b>	<b>0.68</b>	<b>0.96</b>	0.86	1.19
Salicylic acid	0.12	0.15	<b>0.06</b>	<b>0.08</b>	<b>0.28</b>	<b>0.44</b>	0.44	0.63
Malonaldehyde	0.10	0.13	<b>0.07</b>	<b>0.11</b>	<b>0.41</b>	<b>0.62</b>	0.61	0.84
Ethanol	0.07	<b>0.09</b>	<b>0.06</b>	<b>0.09</b>	<b>0.33</b>	<b>0.49</b>	0.44	0.62
Toluene	0.10	0.12	<b>0.03</b>	<b>0.04</b>	<b>0.14</b>	<b>0.21</b>	0.32	0.41
Paracetamol	0.15	0.20	<b>0.13</b>	<b>0.18</b>	<b>0.49</b>	<b>0.70</b>	0.67	0.93
Azobenzene	<b>0.09</b>	<b>0.13</b>	0.11	0.15	<b>0.41</b>	<b>0.61</b>	0.58	0.81

The development of GDML has been guided by the objective to reproduce the dynamical behavior of molecules in MD simulations as well as possible. In MD, the PES is explored via integration of Newton's second law of motion, which exclusively involves atomic forces. This dependency is reproduced in the loss function of GDML, which only penalizes force prediction errors without imposing any explicit energy constraints on the integral of the model. Due to that, force prediction performance takes priority over energy predictions during training, giving rise to the name gradient domain machine learning.

However, since energy labels are usually available as a byproduct of force calculations, it can be tempting to include both label types in the loss function of the ML model, in the hope that they will help improve the overall prediction performance for both quantities.

A hybrid-loss function that penalizes force and energy prediction error simultaneously takes the form:

$$\begin{aligned}\mathcal{L}_{+E}(\Omega) &= \sum_i^M \begin{bmatrix} \eta \\ 1-\eta \end{bmatrix} \odot \left( \begin{bmatrix} \mathbf{J}_{\Phi_i} \\ \Phi_i^\top \end{bmatrix} \omega - \begin{bmatrix} \mathbf{F}_i \\ -E \end{bmatrix} \right)^2 = \sum_i^M \left[ \frac{\hat{\mathbf{f}}_F - \mathbf{F}_i}{\sqrt{\tilde{\eta}}(\hat{f}_E - E_i)} \right]^2 \\ &= \sum_i^M (\hat{\mathbf{f}}_F - \mathbf{F}_i)^2 + \tilde{\eta}(\hat{f}_E - E_i)^2,\end{aligned}\quad (4.12)$$

where  $\odot$  is an element-wise multiplication operator. Sometimes, a linear trade-off hyper-parameter  $\tilde{\eta} = (1-\eta)/\eta$  and  $\eta \in [0, 1]$  is introduced to account for the relative differences in units, information content and noise level of both label types [29, 39, 40]. However, a bilateral reduction of both loss terms is only possible, if both objectives are non-competing [151]. The implication is that the optimal parameter set would be effective across both tasks, which nullifies the benefits of a combined loss in the first place.

A linear combination of energy and force loss assumes that there exists a conversion factor, i.e. that both error types are proportional to each other. However, rearranging Eq. 4.12 yields the following contradiction for  $\mathcal{L}_{+E}(\Omega) \neq \mathbf{0}$ :

$$\begin{aligned}\sum_i^M (\hat{\mathbf{f}}_F - \mathbf{F}_i)^2 &= \sum_i^M \tilde{\eta}(\hat{f}_E - E_i)^2 \\ \rightarrow \sum_i^M (-\nabla(\hat{f}_E - E_i))^2 &= \sum_i^M (\sqrt{\tilde{\eta}}(\hat{f}_E - E_i))^2 \quad \not\propto\end{aligned}\quad (4.13)$$

While the derivative is a linear operator, it obviously does not map to a multiple of the original function in general. This is only the case for its eigenfunction<sup>1</sup>, the exponential. Clearly, it is therefore not justified to join both quantities in one loss function linearly, because they will cause the predictor to either overfit on the energies or the forces. Recent literature [39, 40] gives empirical evidence of such behavior. To investigate the effect of a combined loss function on the sGDML model, we have extended our original formulation with energy constraints by constructing the following modified kernel:

$$\mathbf{k}_{\text{sGDML+E}} = \begin{bmatrix} \text{Hess}(k) & \nabla k \\ (\nabla k)^\top & k \end{bmatrix}. \quad (4.14)$$

We remark, that this formulation follows directly from Eq. 4.12. Here,  $k$  is the energy kernel, which is coupled the original force field kernel by its first derivative  $\nabla k$ . Table 4.7

---

<sup>1</sup>The exponential function  $f(x) = c \exp(\lambda x)$  for arbitrary constants  $c$ , is the solution to the differential equation  $\partial f(x)/\partial x = \lambda f(x)$  with eigenvalue  $\lambda$ .

shows the prediction accuracy for both quantities using the original sGDML formulation and the extended sGDML+E variant.

We observe, that the sGDML+E model overfits its energy constraints on all datasets, at the cost of a significant degradation in force prediction accuracy. The force prediction error for benzene grew the most (by a factor of 4.8), whereas aspirin shows the mildest decline (factor 1.3). We remark, that the degradation strongly correlates with the performance of the unmodified sGDML model: the smaller the original prediction error on a dataset, the bigger the degradation in accuracy after the inclusion of energy constraints. Overall, these empirical results support our initial theoretical considerations.

For MD simulations, a model with optimal force prediction performance is desirable, in order to represent the dynamical behavior of the molecule correctly. An improved energy prediction accuracy is meaningless, if the associated MD trajectory is inaccurate due to unreliable force predictions. It may be enticing to use two separate models for predicting energies and forces, each optimized for its respective task [39, 40]. However, this introduces inconstancies between energy and force prediction along an MD trajectory, which would lead to a miss-representation of the thermodynamical properties of the system. We are thus convinced that gradient domain learning approaches this problem from the right direction.

#### 4.5.2 Imposing permutational symmetry

Same-species atoms within a molecule are indistinguishable from one another and can be exchanged while leaving its energy and forces invariant. ML models that share the same symmetry can be more data efficient, which motivated many developments in that direction early on [152, 153, 27, 154].

##### Invariant integration

Typically, permutational invariance is implemented via invariant integration, either over the permutational symmetry group of the molecule [29, 17, 22], or over the 3D rotation group of a continuous basis expansion of the nuclei positions [29, 31, 14]. Both approaches require integration over a large domain during inference, which incurs considerable computational cost. While certain basis expansions allow analytical integration, the choice is limited [29]. Alternatively, a smaller isotropic basis can be used to alleviate some of the computational burden [38–40]. Another popular approach is to fragment the molecular structure into smaller parts to reduce the cardinality of the symmetric group of

the system. However, such a localized model will not be able to faithfully reconstruct the global nature of quantum mechanics.

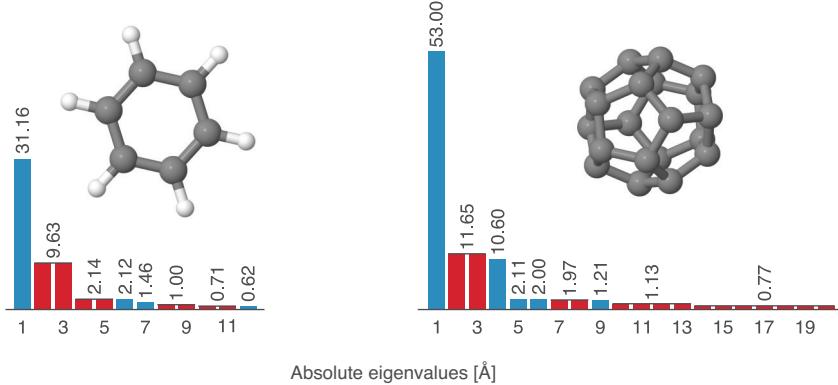
In sGML, we pursue a different approach in which we limit the invariant integration to the physical point group and fluxional symmetries that are actually relevant [5]. Relevant symmetries are those that are accessible without crossing impassable energy barriers. They can be automatically recovered from the training set and integrated into the force field kernel [155], which yields an invariant model with the exact same number of parameters as the original, non-symmetric one. Invariant integration over the set of meaningful symmetries is inexpensive: with 12 physically relevant symmetries, benzene, toluene and azobenzene are the most symmetric molecules considered here, whereas their full symmetric groups have orders  $6!6!$ ,  $7!8!$  and  $12!10!2!$ , respectively. Despite this reduction in complexity, the sGML model is indistinguishable from a fully symmetrized model, in terms of its prediction performance.

### Optimal assignment

An alternative to invariant integration is the optimal assignment approach [156], where each model input undergoes a transformation to a canonical permutational configuration before inference. Of course, the prediction then needs to be transformed back accordingly to produce the expected output. Such a model effectively performs a local reconstruction of the symmetric part of the target function which is then effectively 'tiled' across the entire input domain. This involves compressing the data  $\mathbf{x}_i$  to one of its symmetric subdomains via transformation to a fixed reference configuration  $\mathbf{P}_{i1}\mathbf{x}_i$  prior to training. Such an approach bears two major disadvantages over our proposition:

- Every query molecule must be first matched to the training set, making evaluations of the model computationally costly.
- The "tiling" process causes discontinuous seams to form along borders of neighboring symmetric subdomains, where different copies of the local model meet. These seams correspond with the symmetry lines of the molecule, which are frequently crossed during MD simulations. Moreover, they reside in the extrapolation regime of the local model, where the prediction performance is notoriously bad.

Our approach resolves these issues by effectively reconstructing all symmetric subdomains simultaneously. It retains all advantages of the assignment kernel.



**Figure 4.8** Eigenspectra of the adjacency matrices of two highly symmetric molecules: benzene with symmetries in two dimensions (left) and the C<sub>20</sub> fullerene with symmetries in three dimensions (right). Benzene has 12 point group symmetries and eight out of its 12 eigenvalues are degenerate (shown in red). C<sub>20</sub> has 120 symmetries, with 15 degenerate eigenvalues out of 20. An unambiguous assignment of eigenvectors between several near-isomorphic instances of these structures (close to equilibrium) is therefore impossible. Our proposed multi-partite matching algorithm resolves the inconsistencies across multiple bi-partite assignments in the training set that arise from this ambiguity and other factors.

#### 4.5.3 Degenerate eigenvalues and the bi-partite matching algorithm

The approximate bi-partite matching algorithm underlying our multi-partite extension assumes that the adjacency matrices of molecular graphs have non-degenerate eigenvalues (i.e. with multiplicity one). Under the additional premise that all molecular graphs in the training set are near-isomorphic, it then establishes a one-to-one correspondence between the eigenvectors to solve the matching problem [137]. However, certain graph topologies have degenerate eigenvalues and thus rotational freedom in the eigenvector basis. In those cases, the bi-matching problem is ill-posed, since an unambiguous assignment of eigenvectors is no longer possible. A high likelihood of inconsistent bi-partite matchings across the training set is the result. The purpose of our proposed subsequent multi-partite matching step is to resolve those inconsistencies.

Degenerate eigenvalues are especially prevalent in highly symmetric graphs. While molecular graphs generated in MD simulations are rarely in perfect equilibrium, their eigenvectors remain similar and therefore hard to distinguish from each other. We illustrate that by means of two molecules with full rotational symmetry: benzene and the C<sub>20</sub> fullerene. The eigenspectra of both structures in equilibrium are highly degenerate, yet our algorithm recovers the full point groups D<sub>6h</sub> (12 symmetries) and I<sub>h</sub> (120 symmetries), respectively (see Figure 4.8). This illustrates that the proposed multi-partite matching behaves robustly, even when the set of initial pairwise assignments is of poor quality.

## 4.6 Software implementation

The sGDML model developed in this chapter is also available in our software package that was introduced in section 3.5.

### Source code

Software, documentation, datasets and pre-trained models are available at:  
[www.sgdml.org](http://www.sgdml.org)

## 4.7 Summary

We have extended the GDML model developed in the previous chapter to additionally incorporate all relevant rigid space group symmetries as well as dynamic non-rigid symmetries. Typically, the identification of symmetries requires chemical and physical intuition about the system at hand, which is impractical in a ML setting. Through a data-driven multi-partite matching approach, we automate the discovery of permutation matrices of molecular graph pairs in different permutational configurations and thus between symmetric transformations undergone within the scope of a dataset. This allows us to define a compact symmetric model that can be parametrized from very small training datasets, enabling the direct construction of flexible molecular force fields from expensive high-level *ab initio* calculations.

The developed sGDML model calculates energies and forces at speeds around four and eight orders of magnitude faster than DFT and CCSD(T), respectively. Compared to conventional FFs, sGDML is however only around one to three orders of magnitude slower. This brings it closer to polarizable force fields [157] than classical force fields like AMBER [3, 4], CHARMM [158, 159], or GROMACS [160] in terms of speed.

This reconciliation of accuracy and speed allows our approach to faithfully reproduce global force fields at quantum-chemical CCSD(T) level of accuracy, while enabling converged molecular dynamics simulations with fully quantized electrons and nuclei. Such simulations are key for the accurate prediction of molecular behavior at realistic conditions, but unfeasible within brute-force *ab initio* approaches since they would require millions of CPU years.

In various numerical experiments, we have demonstrated that our sGDML model genuinely captures essential features of molecular PESs, like local energy minima and energy barriers. In fact, it is accurate enough to allow a detailed study of highly resolved

topographical differences of PESs at different levels of electronic structure theory. Furthermore, we have presented MD simulations for flexible molecules that provide insights into their the dynamical behavior. For small molecules like benzene and toluene, our model can even reach spectroscopic accuracy in the energy, with an accuracy of a few wavenumbers for the position of the spectral peaks. These results show that we achieved our goal of constructing an efficient empirical model that is able to yield highly predictive results.



# Chapter 5

## Conclusion

In this thesis, we have addressed the accuracy and computational efficiency dilemma that arises in the description of PESs. The computational cost of accurate *ab initio* calculations prohibits large numbers of (energy and force) evaluations, whereas efficient mechanistic approximations are unable to integrate important insights from quantum mechanics. Meaningful conclusions about the dynamical and thermodynamical properties of a system are however only possible with a sufficient sampling of the configuration space, which frequently entails millions of PES evaluations. In practice, this rules out the use of *ab initio* methods, to the detriment of the predictive power of these simulations. This problem is only aggravated by systems sensitive to NQEs, which require an even more expensive PIMD sampling.

As an alternative, we have proposed a combined quantum mechanics and ML approach that is able to reconcile both contradicting aspects of accuracy and computational efficiency. We have approached this challenge with techniques from probabilistic inference, using universal approximators that have the flexibility to model any atomic interaction. Typically, the parametrization of such general models relies on the availability of large reference datasets to obtain an accurate results, which would prevent the construction of ML models using high-level *ab initio* methods. We have overcome this restrictive requirement by informing the model with fundamental physical invariances and conservation laws. Not only does this approach make the models more data-efficient, it also guarantees that the incorporated physics are represented without artifacts.

We have developed models that include the full set of temporal and spatial symmetries of molecules. Homogeneity of time implies energy conservation and global spatial symmetries include rotational and translational invariance of the energy. Using a generalization of GPs to vector-valued Hilbert spaces, we have defined a predictor that explicitly maps to energy conserving solutions and thus allows the simultaneous prediction of accurate

inter-atomic forces and corresponding potential energy of molecules. Our approach emerges from the insight that FFs should be reconstructed in the gradient domain and we have shown theoretically and empirically that this approach indeed leads to an optimal prediction of forces.

For small molecules with a few dozen atoms, the initially developed energy-conserving model can be parametrized from a few thousand data points. In an effort to further increase the training data efficiency of the model, we have proceeded to incorporate spatial geometries, creating sGDML. While point group symmetries are routinely exploited in computational chemistry, we have developed a fully automated algorithm which additionally extracts all fluxional symmetries that are present in the training dataset. This required us to solve the multi-partite assignment problem using permutation synchronization. The resulting sGDML model is data efficient and accurate enough to allow the use of coupled cluster calculations as a reference.

In a series of numerical experiments, we have finally highlighted the necessity of using such accurate descriptions of forces with an equally accurate treatment of NQEs for achieving reliable and quantitative understanding of molecular systems. For the first time, we were able to compute the CCSD(T) vibrational spectrum of ethanol using the velocity-velocity autocorrelation function based on centroid PIMD. We have concluded by demonstrating the wider applicability of the sGDML by performing a detailed analysis of MD simulations of more complex molecules like malonaldehyde and aspirin. Again, we found significant consequences of a proper inclusion of the electron correlation effects, enabled by our model.

## 5.1 Outlook

There is a number of challenges that remain to be solved to extend the sGDML model in terms of its applicability and scaling to larger molecular systems. Given an extensive set of individually trained sGDML models, an unseen molecule can be represented as a non-linear combination of those models. This would allow scaling up and transferable prediction for molecules that are similar in size. For example, the well-separated inter- and intramolecular correlation scales within molecular solids suggest that a hierarchical decomposition is possible with limited degradation of prediction accuracy.

The high efficiency of GPs is due to the fact that they operate in a predefined high-dimensional feature space in which the learning task is less complex. This space is implicitly characterized by the covariance function, which was chosen based on physical intuition and previous empirical results in this work. A systematic construction of this

space would however offer many advantages, including the opportunity to transfer learned concepts in-between systems in a principled way. The theoretical foundations that would enable this effort already exists in the form of so-called random features [95].

Advanced sampling strategies could be employed to combine forces from different levels of theory to minimize the need for computationally-intensive *ab initio* calculations. Our focus in this work was on intramolecular forces in small- and medium-sized molecules. Looking ahead, it is sensible to integrate the sGDML model with an accurate intermolecular force field to enable predictive simulations of condensed molecular systems (Ref. [51] presents an intermolecular model which would be particularly suited for coupling with sGDML). Many other avenues for further development exist [161], including incorporating additional physical priors, reducing dimensionality of complex PESs, computing reaction pathways, and modeling infrared, Raman, and other spectroscopic measurements.



# Bibliography

- [1] Igor Poltavsky and Alexandre Tkatchenko. Modeling quantum nuclei with perturbed path integral molecular dynamics. *Chem. Sci.*, 7(2):1368–1372, 2016.
- [2] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *J. Mach. Learn. Res.*, 9(2579-2605):85, 2008.
- [3] Paul K. Weiner and Peter A. Kollman. AMBER: Assisted model building with energy refinement. A general program for modeling molecules and their interactions. *J. Comput. Chem.*, 2(3):287–303, 1981.
- [4] D.A. Case, I.Y. Ben-Shalom, S.R. Brozell, D.S. Cerutti, T.E. Cheatham, III, V.W.D. Cruzeiro, T.A. Darden, R.E. Duke, D. Ghoreishi, M.K. Gilson, H. Gohlke, A.W. Goetz, D. Greene, R Harris, N. Homeyer, S. Izadi, A. Kovalenko, T. Kurtzman, T.S. Lee, S. LeGrand, P. Li, C. Lin, J. Liu, T. Luchko, R. Luo, D.J. Mermelstein, K.M. Merz, Y. Miao, G. Monard, C. Nguyen, H. Nguyen, I. Omelyan, A. Onufriev, F. Pan, R. Qi, D.R. Roe, A. Roitberg, C. Sagui, S. Schott-Verdugo, J. Shen, C.L. Simmerling, J. Smith, R. Salomon-Ferrer, J. Swails, R.C. Walker, J. Wang, H. Wei, R.M. Wolf, X. Wu, L. Xiao, D.M. York D.M., and P.A. Kollman. Amber 2018. <http://ambermd.org>, 2018.
- [5] Hugh C. Longuet-Higgins. The symmetry groups of non-rigid molecules. *Mol. Phys.*, 6(5):445–460, 1963.
- [6] Mark E. Tuckerman. Ab initio molecular dynamics: Basic concepts, current trends and novel applications. *J. Phys. Condens. Matter*, 14(50):R1297, 2002.
- [7] Matthias Rupp, Alexandre Tkatchenko, Klaus-Robert Müller, and O. Anatole Von Lilienfeld. Fast and accurate modeling of molecular atomization energies with machine learning. *Phys. Rev. Lett.*, 108(5):58301, 2012.
- [8] Katja Hansen, Grégoire Montavon, Franziska Biegler, Siamac Fazli, Matthias Rupp, Matthias Scheffler, O. Anatole von Lilienfeld, Alexandre Tkatchenko, and Klaus-Robert Müller. Assessment and validation of machine learning methods for predicting molecular atomization energies. *J. Chem. Theory Comput.*, 9(8):3404–3419, 2013.
- [9] Katja Hansen, Franziska Biegler, Raghunathan Ramakrishnan, Wiktor Pronobis, O Anatole von Lilienfeld, Klaus-Robert Müller, and Alexandre Tkatchenko. Machine learning predictions of molecular properties: Accurate many-body potentials and nonlocality in chemical space. *J. Phys. Chem. Lett.*, 6(12):2326–2331, 2015.

- [10] Matthias Rupp, Raghunathan Ramakrishnan, and O. Anatole von Lilienfeld. Machine learning for quantum mechanical properties of atoms in molecules. *J. Phys. Chem. Lett.*, 6(16):3309–3313, 2015.
- [11] Venkatesh Botu and Rampi Ramprasad. Adaptive machine learning framework to accelerate ab initio molecular dynamics. *Int. J. Quantum Chem.*, 115(16):1074–1083, 2015.
- [12] Matthew Hirn, Nicolas Poirvert, and Stéphane Mallat. Quantum energy regression using scattering transforms. *CoRR*, abs/1502.02077, 2015.
- [13] Raghunathan Ramakrishnan, Pavlo O. Dral, Matthias Rupp, and O. Anatole von Lilienfeld. Big data meets quantum chemistry approximations: The  $\delta$ -machine learning approach. *J. Chem. Theory Comput.*, 11(5):2087–2096, 2015.
- [14] Sandip De, Albert P Bartók, Gábor Csányi, and Michele Ceriotti. Comparing molecules and solids across structural and alchemical space. *Phys. Chem. Chem. Phys.*, 18(20):13754–13769, 2016.
- [15] Nongnuch Artrith, Alexander Urban, and Gerbrand Ceder. Efficient and accurate machine-learning interpolation of atomic energies in compositions with many species. *Phys. Rev. B*, 96(1):14112, 2017.
- [16] Albert P. Bartók, Sandip De, Carl Poelking, Noam Bernstein, James R. Kermode, Gábor Csányi, and Michele Ceriotti. Machine learning unifies the modeling of materials and molecules. *Sci. Adv.*, 3(12):e1701816, 2017.
- [17] Aldo Glielmo, Peter Sollich, and Alessandro De Vita. Accurate interatomic force fields via machine learning with covariant kernels. *Phys. Rev. B*, 95:214302, 2017.
- [18] Kun Yao, John E. Herr, and John Parkhill. The many-body expansion combined with neural networks. *J. Chem. Phys.*, 146(1):14106, 2017.
- [19] ST John and Gábor Csányi. Many-body coarse-grained interactions using Gaussian approximation potentials. *J. Phys. Chem. B*, 121(48):10934–10949, 2017.
- [20] Felix A. Faber, Luke Hutchison, Bing Huang, Justin Gilmer, Samuel S. Schoenholz, George E. Dahl, Oriol Vinyals, Steven Kearnes, Patrick F. Riley, and O. Anatole von Lilienfeld. Prediction errors of molecular machine learning models lower than hybrid DFT error. *J. Chem. Theory Comput.*, 13(11):5255–5264, 2017.
- [21] Michael Eickenberg, Georgios Exarchakis, Matthew Hirn, Stéphane Mallat, and Louis Thiry. Solid harmonic wavelet scattering for predictions of molecule properties. *J. Chem. Phys.*, 148(24):241732, 2018.
- [22] Aldo Glielmo, Claudio Zeni, and Alessandro De Vita. Efficient nonparametric n-body force fields from machine learning. *Phys. Rev. B*, 97(18):184307, 2018.
- [23] Yu-Hang Tang, Dongkun Zhang, and George Em Karniadakis. An atomistic fingerprint algorithm for learning ab initio molecular force fields. *J. Chem. Phys.*, 148(3):34101, 2018.

- [24] Andrea Grisafi, David M. Wilkins, Gábor Csányi, and Michele Ceriotti. Symmetry-adapted machine learning for tensorial properties of atomistic systems. *Phys. Rev. Lett.*, 120:36002, 2018.
- [25] Wiktor Pronobis, Alexandre Tkatchenko, and Klaus-Robert Müller. Many-body descriptors for predicting molecular properties with machine learning: Analysis of pairwise and three-body interactions in molecules. *J. Chem. Theory Comput.*, 14(6):2991–3003, 2018.
- [26] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.*, 98(14):146401, 2007.
- [27] Albert P. Bartók, Mike C. Payne, Risi Kondor, and Gábor Csányi. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Phys. Rev. Lett.*, 104(13):136403, 2010.
- [28] K. V. Jovan Jose, Nongnuch Artrith, and Jörg Behler. Construction of high-dimensional neural network potentials using environment-dependent atom pairs. *J. Chem. Phys.*, 136(19):194111, 2012.
- [29] Albert P. Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Phys. Rev. B*, 87(18):184115, 2013.
- [30] Grégoire Montavon, Matthias Rupp, Vivekanand Gobre, Alvaro Vazquez-Mayagoitia, Katja Hansen, Alexandre Tkatchenko, Klaus-Robert Müller, and O. Anatole von Lilienfeld. Machine learning of molecular electronic properties in chemical compound space. *New J. Phys.*, 15(9):95003, 2013.
- [31] Albert P. Bartók and Gábor Csányi. Gaussian approximation potentials: A brief tutorial introduction. *Int. J. Quantum Chem.*, 115(16):1051–1057, 2015.
- [32] V. Botu and R. Ramprasad. Learning scheme to predict atomic forces and accelerate materials simulations. *Phys. Rev. B*, 92:94306, 2015.
- [33] Tristan Bereau, Denis Andrienko, and O. Anatole von Lilienfeld. Transferable atomic multipole machine learning models for small organic molecules. *J. Chem. Theory Comput.*, 11(7):3225–3233, 2015.
- [34] Zhenwei Li, James R. Kermode, and Alessandro De Vita. Molecular dynamics with on-the-fly machine learning of quantum-mechanical forces. *Phys. Rev. Lett.*, 114:96405, 2015.
- [35] Jörg Behler. Perspective: Machine learning potentials for atomistic simulations. *J. Chem. Phys.*, 145(17):170901, 2016.
- [36] Felix Brockherde, Leslie Vogt, Li Li, Mark E. Tuckerman, Kieron Burke, and Klaus-Robert Müller. Bypassing the Kohn-Sham equations with machine learning. *Nat. Commun.*, 8:872, 2017.
- [37] Michael Gastegger, Jorg Behler, and Philipp Marquetand. Machine learning molecular dynamics for the simulation of infrared spectra. *Chem. Sci.*, 8:6924–6935, 2017.

- [38] Kristof T. Schütt, Farhad Arbabzadah, Stefan Chmiela, Klaus-Robert Müller, and Alexandre Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nat. Commun.*, 8:13890, 2017.
- [39] Kristof Schütt, Pieter-Jan Kindermans, Huziel E. Sauceda, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. In *Adv. Neural. Inf. Process. Syst. 31*, pages 991–1001, 2017.
- [40] Kristof T. Schütt, H. E. Sauceda, Pieter-Jan Kindermans, Alexandre Tkatchenko, and Klaus-Robert Müller. SchNet – A deep learning architecture for molecules and materials. *J. Chem. Phys.*, 148(24):241722, 2018.
- [41] Bing Huang and O. Anatole von Lilienfeld. The "DNA" of chemistry: Scalable quantum machine learning with "amons". *arXiv preprint arXiv:1707.04146*, 2017.
- [42] 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 Comput. Mater.*, 3(1):37, 2017.
- [43] Evgeny V. Podryabinkin and Alexander V. Shapeev. Active learning of linearly parametrized interatomic potentials. *Comput. Mater. Sci.*, 140:171–180, 2017.
- [44] Pavlo O. Dral, Alec Owens, Sergei N. Yurchenko, and Walter Thiel. Structure-based sampling and self-correcting machine learning for accurate calculations of potential energy surfaces and vibrational levels. *J. Chem. Phys.*, 146(24):244108, 2017.
- [45] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and E. Weinan. Deep potential molecular dynamics: A scalable model with the accuracy of quantum mechanics. *Phys. Rev. Lett.*, 120(14):143001, 2018.
- [46] Nicholas Lubbers, Justin S. Smith, and Kipton Barros. Hierarchical modeling of molecular energies using a deep neural network. *J. Chem. Phys.*, 148(24):241715, 2018.
- [47] Kevin Ryczko, Kyle Mills, Iryna Luchak, Christa Homenick, and Isaac Tamblyn. Convolutional neural networks for atomistic systems. *Comput. Mater. Sci.*, 149:134–142, 2018.
- [48] Kenta Kanamori, Kazuaki Toyoura, Junya Honda, Kazuki Hattori, Atsuto Seko, Masayuki Karasuyama, Kazuki Shitara, Motoki Shiga, Akihide Kuwabara, and Ichiro Takeuchi. Exploring a potential energy surface by machine learning for characterizing atomic transport. *Phys. Rev. B*, 97(12):125124, 2018.
- [49] Truong Son Hy, Shubhendu Trivedi, Horace Pan, Brandon M. Anderson, and Risi Kondor. Predicting molecular properties with covariant compositional networks. *J. Chem. Phys.*, 148(24):241745, 2018.
- [50] Jiang Wang, Christoph Wehmeyer, Frank Noé, and Cecilia Clementi. Machine learning of coarse-grained molecular dynamics force fields, 2018.

- [51] Tristan Bereau, Robert A. DiStasio Jr, Alexandre Tkatchenko, and O Anatole Von Lilienfeld. Non-covalent interactions across organic and biological subsets of chemical space: Physics-based potentials parametrized from machine learning. *J. Chem. Phys.*, 148(24):241706, 2018.
- [52] Andreas Mardt, Luca Pasquali, Hao Wu, and Frank Noé. VAMPnets for deep learning of molecular kinetics. *Nat. Commun.*, 9(1):5, 2018.
- [53] Frank Noé and Hao Wu. Boltzmann generators – Sampling equilibrium states of many-body systems with deep learning. *arXiv preprint arXiv:1812.01729*, 2018.
- [54] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3D point clouds. *arXiv preprint arXiv:1802.08219*, 2018.
- [55] Justin S. Smith, Ben Nebgen, Nicholas Lubbers, Olexandr Isayev, and Adrian Roitberg. Less is more: Sampling chemical space with active learning. *J. Chem. Phys.*, 148(24):241733, 2018.
- [56] Konstantin Gubaev, Evgeny V. Podryabinkin, and Alexander V. Shapeev. Machine learning of molecular properties: Locality and active learning. *J. Chem. Phys.*, 148(24):241727, 2018.
- [57] Felix A. Faber, Anders S. Christensen, Bing Huang, and O. Anatole von Lilienfeld. Alchemical and structural distribution based representation for universal quantum machine learning. *J. Chem. Phys.*, 148(24):241717, 2018.
- [58] Anders S. Christensen, Felix A. Faber, and O. Anatole von Lilienfeld. Operators in quantum machine learning: Response properties in chemical space. *J. Phys. Chem.*, 150(6):64105, 2019.
- [59] Robin Winter, Floriane Montanari, Frank Noé, and Djork-Arné Clevert. Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chem. Sci.*, 2019.
- [60] Konstantin Gubaev, Evgeny V. Podryabinkin, Gus L. W. Hart, and Alexander V. Shapeev. Accelerating high-throughput searches for new alloys with active learning of interatomic potentials. *Comput. Mater. Sci.*, 156:148–156, 2019.
- [61] Dominik Marx and Michele Parrinello. Ab initio path integral molecular dynamics: Basic ideas. *J. Chem. Phys.*, 104(11):4077–4082, 1996.
- [62] Attila Szabo and Neil S. Ostlund. *Modern quantum chemistry: Introduction to advanced electronic structure theory*. Courier Corporation, 2012.
- [63] Dominik Marx and Jürg Hutter. *Ab initio molecular dynamics: Basic theory and advanced methods*. Cambridge University Press, 2009.
- [64] Thomas E. Markland and Michele Ceriotti. Nuclear quantum effects enter the mainstream. *Nat. Rev. Chem.*, 2:109, 2018.

- [65] Mark Tuckerman. *Statistical mechanics: Theory and molecular simulation*. Oxford University Press, 2010.
- [66] Anthony Stone. *The theory of intermolecular forces*. Oxford University Press, 2013.
- [67] Konrad Patkowski, Garold Murdachaew, Cheng-Ming Fou, and Krzysztof Szalewicz. Accurate ab initio potential for argon dimer including highly repulsive region. *Molecular Physics*, 103(15-16):2031–2045, 2005.
- [68] Pierre Hohenberg and Walter Kohn. Inhomogeneous electron gas. *Phys. Rev.*, 136(3b):B864, 1964.
- [69] Walter Kohn and Lu Jeu Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.*, 140(4a):A1133, 1965.
- [70] Mark E. Tuckerman and Glenn J. Martyna. Understanding modern molecular dynamics: Techniques and applications, 2000.
- [71] Leonid Pereyaslavets, Igor Kurnikov, Ganesh Kamath, Oleg Butin, Alexey Illarionov, Igor Leontyev, Michael Olevanov, Michael Levitt, Roger D. Kornberg, and Boris Fain. On the importance of accounting for nuclear quantum effects in ab initio calibrated force fields in biological simulations. *Proc. Natl. Acad. Sci. U.S.A.*, 115(36):8878–8882, 2018.
- [72] Aran Lamaire, Jelle Wieme, Sven M. J. Rogge, Michel Waroquier, and Veronique Van Speybroeck. On the importance of anharmonicities and nuclear quantum effects in modelling the structural properties and thermal expansion of MOF-5. *J. Chem. Phys.*, 150(9):94503, 2019.
- [73] Mark E. Tuckerman, Bruce J. Berne, Glenn J. Martyna, and Michael L. Klein. Efficient molecular dynamics and hybrid monte carlo algorithms for path integrals. *J. Chem. Phys.*, 99(4):2796–2808, 1993.
- [74] E. Balog, A. L. Hughes, and Glenn J. Martyna. Constant pressure path integral molecular dynamics studies of quantum effects in the liquid state properties of n-alkanes. *J. Chem. Phys.*, 112(2):870–880, 2000.
- [75] Glenn J. Martyna, Adam Hughes, and Mark E. Tuckerman. Molecular dynamics algorithms for path integrals at constant pressure. *J. Chem. Phys.*, 110(7):3275–3290, 1999.
- [76] David Chandler and Peter G. Wolynes. Exploiting the isomorphism between quantum theory and classical statistical mechanics of polyatomic fluids. *J. Chem. Phys.*, 74(7):4078–4095, 1981.
- [77] Emmy Noether. Invarianten beliebiger Differentialausdrücke. *Gött. Nachr., mathematisch-physikalische Klasse*, 1918:37–44, 1918.
- [78] Bernhard Schölkopf and Alexander J. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press, 2001.
- [79] Grace Wahba. *Spline models for observational data*, volume 59. SIAM, 1990.

- [80] Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. A generalized representer theorem. In *International conference on computational learning theory*, pages 416–426. Springer, 2001.
- [81] Klaus-Robert Müller, Sebastian Mika, Gunnar Rätsch, Koji Tsuda, and Bernhard Schölkopf. An introduction to kernel-based learning algorithms. *IEEE Trans. Neural Netw. Learn. Syst.*, 12(2):181–201, 2001.
- [82] Carl E. Rasmussen. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer, 2004.
- [83] David Duvenaud. *Automatic model construction with Gaussian processes*. PhD thesis, University of Cambridge, 2014.
- [84] Charles A. Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *J. Mach. Learn. Res.*, 7(Dec):2651–2667, 2006.
- [85] Andreas Damianou and Neil Lawrence. Deep Gaussian processes. In *Artificial Intelligence and Statistics*, pages 207–215, 2013.
- [86] Cornelius Lanczos. *The variational principles of mechanics*. University of Toronto Press, 1949.
- [87] Katherine Brading and Elena Castellani. *Symmetries in physics: Philosophical reflections*. Cambridge University Press, 2003.
- [88] Francis J. Narcowich and Joseph D Ward. Generalized Hermite interpolation via matrix-valued conditionally positive definite functions. *Math. Comput.*, 63(208):661–687, 1994.
- [89] Ercan Solak, Roderick Murray-Smith, William E Leithead, Douglas J. Leith, and Carl E. Rasmussen. Derivative observations in Gaussian process models of dynamic systems. In *Adv. Neural. Inf. Process. Syst.*, pages 1057–1064, 2003.
- [90] David L Donoho. Compressed sensing. *IEEE Trans. Inf. Theory*, 52(4):1289–1306, 2006.
- [91] Felix X Yu, Krzysztof Choromanski, Sanjiv Kumar, Tony Jebara, and Shih-Fu Chang. On learning from label proportions. *arXiv preprint arXiv:1402.5902*, 2014.
- [92] David J. C. MacKay. *Introduction to Gaussian processes*, volume 168 of *NATO ASI Series F: Computer and Systems Sciences*. Springer, 1998.
- [93] Alex J Smola, Bernhard Schölkopf, and Klaus-Robert Müller. The connection between regularization operators and support vector kernels. *Neural networks*, 11(4):637–649, 1998.
- [94] Christopher Heil. *Metrics, Norms, Inner Products, and Operator Theory*. Birkhäuser Basel, 2018.
- [95] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Adv. Neural. Inf. Process. Syst.*, pages 1177–1184, 2008.

- [96] Peter Politzer and Jane S. Murray. The Hellmann-Feynman theorem: A perspective. *J. Mol. Model.*, 24(9):266, 2018.
- [97] Richard Phillips Feynman. Forces in molecules. *Phys. Rev.*, 56(4):340, 1939.
- [98] C.E. Shannon. Communication in the Presence of Noise. *Proceedings of the IEEE*, 86(2):447–457, 1998.
- [99] John C Snyder, Matthias Rupp, Katja Hansen, Klaus-Robert Müller, and Kieron Burke. Finding density functionals with machine learning. *Phys. Rev. Lett.*, 108(25):253002, 2012.
- [100] John C Snyder, Matthias Rupp, Klaus-Robert Müller, and Kieron Burke. Nonlinear gradient denoising: Finding accurate extrema from inaccurate functional derivatives. *Int. J. Quantum Chem.*, 115(16):1102–1114, 2015.
- [101] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, 1998.
- [102] Bernhard Schölkopf, Sebastian Mika, Chris J. C. Burges, Philipp Knirsch, Klaus-Robert Müller, Gunnar Rätsch, and Alexander J. Smola. Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5):1000–1017, 1999.
- [103] Charles A. Micchelli and Massimiliano Pontil. On learning vector-valued functions. *Neural Comput.*, 17(1):177–204, 2005.
- [104] Andrea Caponnetto, Charles A. Micchelli, Massimiliano Pontil, and Yiming Ying. Universal multi-task kernels. *J. Mach. Learn. Res.*, 9:1615–1646, 2008.
- [105] Vikas Sindhwani, Hà Quang Minh, and Aurélie C. Lozano. Scalable matrix-valued kernel learning for high-dimensional nonlinear multivariate regression and granger causality. In *Proc. 29th Conference on Uncertainty in Artificial Intelligence*, Uai’13, pages 586–595, Arlington, Virginia, United States, 2013. AUAI Press.
- [106] Trevor Hastie, Robert Tibshirani, and Jerome H. Friedman. *The elements of statistical learning: Data mining, inference, and prediction*. Springer series in statistics. Springer, 2009.
- [107] Mauricio A. Alvarez, Lorenzo Rosasco, Neil D. Lawrence, et al. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012.
- [108] Phillip Boyle and Marcus Frean. Dependent Gaussian processes. In *Adv. Neural. Inf. Process. Syst.*, pages 217–224, 2005.
- [109] Charles A. Micchelli and Massimiliano Pontil. Kernels for multi-task learning. In *Adv. Neural. Inf. Process. Syst.*, pages 921–928, 2005.
- [110] Luca Baldassarre, Lorenzo Rosasco, Annalisa Barla, and Alessandro Verri. Multi-output learning via spectral filtering. *Machine learning*, 87(3):259–301, 2012.

- [111] Thore Graepel. Solving noisy linear operator equations by Gaussian processes: Application to ordinary and partial differential equations. In *International Conference on Machine Learning*, pages 234–241, 2003.
- [112] Simo Särkkä. Linear operators and stochastic partial differential equations in Gaussian process regression. In *International Conference on Artificial Neural Networks*, pages 151–158. Springer, 2011.
- [113] Emil M. Constantinescu and Mihai Anitescu. Physics-based covariance models for Gaussian processes with multiple outputs. *International Journal for Uncertainty Quantification*, 3(1), 2013.
- [114] Ngoc Cuong Nguyen and Jaime Peraire. Gaussian functional regression for linear partial differential equations. *Comput. Methods Appl. Mech. Eng.*, 287:69–89, 2015.
- [115] Carl Jidling, Niklas Wahlström, Adrian Wills, and Thomas B Schön. Linearly constrained Gaussian processes. In *Adv. Neural. Inf. Process. Syst.*, pages 1215–1224, 2017.
- [116] Hermann Helmholtz. Über Integrale der hydrodynamischen Gleichungen, welche den Wirbelbewegungen entsprechen. *Journal für die reine und angewandte Mathematik*, 55:25–55, 1858.
- [117] Bertil Matérn. *Spatial Variation*. Lecture notes in statistics. Springer, 1986.
- [118] Michael L. Stein. *Interpolation of Spatial Data - Some Theory for Kriging*. Springer New York, 1999.
- [119] I. S. Gradshteyn and I. M. Ryzhik. *Table of Integrals, Series, and Products, (Equation 8.468)*. 7 edition, 2007.
- [120] Tilmann Gneiting, William Kleiber, and Martin Schlather. Matérn cross-covariance functions for multivariate random fields. *J. Am. Stat. Assoc.*, 105(491):1167–1177, 2010.
- [121] Dennis C. Rapaport and Dennis C. Rapaport Rapaport. *The art of molecular dynamics simulation*. Cambridge University Press, 2004.
- [122] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. Generalized gradient approximation made simple. *Phys. Rev. Lett.*, 77:3865–3868, 1996.
- [123] Alexandre Tkatchenko and Matthias Scheffler. Accurate molecular van der waals interactions from ground-state electron density and free-atom reference data. *Phys. Rev. Lett.*, 102(7):73005, 2009.
- [124] Michele Ceriotti, Joshua More, and David E. Manolopoulos. i-PI: A python interface for ab initio path integral molecular dynamics simulations. *Comput. Phys. Commun.*, 185(3):1019–1026, 2014.
- [125] Stefan Chmiela, Huziel E. Sauceda, Igor Poltavsky, Klaus-Robert Müller, and Alexandre Tkatchenko. sGMDL: Constructing accurate and data efficient molecular force fields using machine learning. *Comput. Phys. Commun.*, 2019.

- [126] Anthony M. Reilly and Alexandre Tkatchenko. van der Waals dispersion interactions in molecular materials: Beyond pairwise additivity. *Chem. Sci.*, 6(6):3289–3301, 2015.
- [127] Huziel E. Sauceda, Stefan Chmiela, Igor Poltavsky, Klaus-Robert Müller, and Alexandre 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(11):114102, 2019.
- [128] Michael A Osborne. *Bayesian Gaussian processes for sequential prediction, optimisation and quadrature*. PhD thesis, University of Oxford, 2010.
- [129] Michael A. Osborne, Roman Garnett, and Stephen J. Roberts. Gaussian processes for global optimization. In *3rd international conference on learning and intelligent optimization*, 2009.
- [130] Ask Hjorth Larsen, Jens Jørgen Mortensen, Jakob Blomqvist, Ivano E Castelli, Rune Christensen, Marcin Dułak, Jesper Friis, Michael N Groves, Bjørk Hammer, Cory Hargus, et al. The atomic simulation environment – A python library for working with atoms. *J. Phys. Condens. Matter*, 29(27):273002, 2017.
- [131] Venkat Kapil, Mariana Rossi, Ondrej Marsalek, Riccardo Petraglia, Yair Litman, Thomas Spura, Bingqing Cheng, Alice Cuzzocrea, Robert H Meißner, David M Wilkins, et al. i-PI 2.0: A universal force engine for advanced molecular simulations. *Comput. Phys. Commun.*, 236:214–223, 2019.
- [132] Edgar B. Wilson. *Molecular Vibrations: The Theory of Infrared and Raman Vibrational Spectra*. McGraw-Hill Interamericana, 1955.
- [133] Deepti Pachauri, Risi Kondor, and Vikas Singh. Solving the multi-way matching problem by permutation synchronization. In *Adv. Neural. Inf. Process. Syst.*, pages 1860–1868, 2013.
- [134] Michele Schiavinato, Andrea Gasparetto, and Andrea Torsello. *Transitive Assignment Kernels for Structural Classification*, pages 146–159. Springer International Publishing, Cham, 2015.
- [135] Nils M. Kriege, Pierre-Louis Giscard, and Richard C. Wilson. On valid optimal assignment kernels and applications to graph classification. In *Adv. Neural. Inf. Process. Syst. 30*, pages 1623–1631, 2016.
- [136] Jean-Philippe Vert. The optimal assignment kernel is not positive definite. *CoRR*, abs/0801.4061, 2008.
- [137] Shinji Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(5):695–703, 1988.
- [138] Harold William Kuhn. The Hungarian method for the assignment problem. *Nav. Res. Logist.*, 2(1-2):83–97, 1955.
- [139] Toni Karvonen and Simo Särkkä. Fully symmetric kernel quadrature. *SIAM Journal on Scientific Computing*, 40(2):A697–a720, 2018.

- [140] Bernard Haasdonk and Hans Burkhardt. Invariant kernel functions for pattern analysis and machine learning. *Machine learning*, 68(1):35–61, 2007.
- [141] Volker Blum, Ralf Gehrke, Felix Hanke, Paula Havu, Ville Havu, Xinguo Ren, Karsten Reuter, and Matthias Scheffler. Ab initio molecular simulations with numeric atom-centered orbitals. *Comput. Phys. Commun.*, 180(11):2175–2196, 2009.
- [142] Justin M. Turney, Andrew C. Simmonett, Robert M. Parrish, Edward G. Hohenstein, Francesco A. Evangelista, Justin T. Fermann, Benjamin J. Mintz, Lori A. Burns, Jeremiah J. Wilke, Micah L. Abrams, Nicholas J. Russ, Matthew L. Leininger, Curtis L. Janssen, Edward T. Seidl, Wesley D. Allen, Henry F. Schaefer, Rollin A. King, Edward F. Valeev, C. David Sherrill, and T. Daniel Crawford. Psi4: An open-source ab initio electronic structure program. *WIREs Comput. Mol. Sci.*, 2(4):556–565, 2012.
- [143] Robert M. Parrish, Lori A. Burns, Daniel G. A. Smith, Andrew C. Simmonett, A. Eugene DePrince III, Edward G. Hohenstein, Ugur Bozkaya, Alexander Yu Sokolov, Roberto Di Remigio, Ryan M. Richard, et al. Psi4 1.1: An open-source electronic structure program emphasizing automation, advanced libraries, and interoperability. *J. Chem. Theory Comput.*, 13(7):3185–3197, 2017.
- [144] Stefan Chmiela, Alexandre Tkatchenko, Huziel E. Sauceda, Igor Poltavsky, Kristof T. Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Sci. Adv.*, 3(5):e1603015, 2017.
- [145] Leticia González, Otilia Mó, and Manuel Yáñez. Density functional theory study on ethanol dimers and cyclic ethanol trimers. *J. Chem. Phys.*, 111(9):3855–3861, 1999.
- [146] J. R. Durig and R. A. Larsen. Torsional vibrations and barriers to internal rotation for ethanol and 2, 2, 2-trifluoroethanol. *J. Mol. Struct.*, 238:195–222, 1990.
- [147] Tobias N. Wassermann and Martin A. Suhm. Ethanol monomers and dimers revisited: A raman study of conformational preferences and argon nanocoating effects. *J. Phys. Chem. A*, 114(32):8223–8233, 2010.
- [148] J. R. Durig, W. E. Bucy, C. J. Wurrey, and L. A. Carreira. Raman spectra of gases. XVI. torsional transitions in ethanol and ethanethiol. *J. Phys. Chem. A*, 79(10):988–993, 1975.
- [149] Tomaso Poggio and Fabio Anselmi. *Visual cortex and deep networks: learning invariant representations*. MIT Press, 2016.
- [150] Fabio Anselmi, Lorenzo Rosasco, and Tomaso Poggio. On invariance and selectivity in representation learning. *Inf. Inference*, 5(2):134–158, 2016.
- [151] Ozan Sener and Vladlen Koltun. Multi-task learning as multi-objective optimization. In *Adv. Neural. Inf. Process. Syst. 31*, pages 525–536, 2018.
- [152] Jörg Behler, Sönke Lorenz, and Karsten Reuter. Representing molecule-surface interactions with symmetry-adapted neural networks. *J. Chem. Phys.*, 127(1):14705, 2007.

- [153] Jörg Behler. Atom-centered symmetry functions for constructing high-dimensional neural network potentials. *J. Chem. Phys.*, 134(7):74106, 2011.
- [154] Grégoire Montavon, Katja Hansen, Siamac Fazli, Matthias Rupp, Franziska Biegler, Andreas Ziehe, Alexandre Tkatchenko, Anatole V Lilienfeld, and Klaus-Robert Müller. Learning invariant representations of molecules for atomization energy prediction. In *Adv. Neural. Inf. Process. Syst.* 25, pages 440–448, 2012.
- [155] Stefan Chmiela, Huziel E. Sauceda, Klaus-Robert Müller, and Alexandre Tkatchenko. Towards exact molecular dynamics simulations with machine-learned force fields. *Nat. Commun.*, 9(1):3887, 2018.
- [156] Holger Fröhlich, Jörg K Wegner, Florian Sieker, and Andreas Zell. Optimal assignment kernels for attributed molecular graphs. In *Proceedings of the 22nd international conference on Machine learning*, pages 225–232. ACM, 2005.
- [157] Wei Jiang, David J. Hardy, James C. Phillips, Alexander D. MacKerell, Klaus Schulten, and Benoît Roux. High-performance scalable molecular dynamics simulations of a polarizable force field based on classical drude oscillators in namd. *J. Phys. Chem. Lett.*, 2(2):87–92, 2011.
- [158] Bernard R. Brooks, Robert E. Bruccoleri, Barry D. Olafson, David J. States, S. Swaminathan, and Martin Karplus. CHARMM: A program for macromolecular energy, minimization, and dynamics calculations. *J. Comput. Chem.*, 4(2):187–217, 1983.
- [159] Bernard R Brooks, Charles L Brooks III, Alexander D MacKerell Jr, Lennart Nilsson, Robert J Petrella, Benoît Roux, Youngdo Won, Georgios Archontis, Christian Bartels, Stefan Boresch, et al. CHARMM: The biomolecular simulation program. *J. Comput. Chem.*, 30(10):1545–1614, 2009.
- [160] David Van Der Spoel, Erik Lindahl, Berk Hess, Gerrit Groenhof, Alan E. Mark, and Herman JC Berendsen. GROMACS: Fast, flexible, and free. *J. Comput. Chem.*, 26(16):1701–1718, 2005.
- [161] Phil De Luna, Jennifer Wei, Yoshua Bengio, Alán Aspuru-Guzik, and Edward Sargent. Use machine learning to find energy materials. *Nature*, 552(7683):23, 2017.

# Appendix A

## Derivations

### A.1 Derivative observations

Because differentiation is a linear operation, the derivative of a GP yields another GP, which allows inference based on derivative observations [89, 82]. The associated covariance function between partial derivatives is then

$$\text{cov}\left(\frac{\partial f}{\partial \mathbf{x}}, \frac{\partial f'}{\partial \mathbf{x}'}\right) = \frac{\partial^2 \text{cov}(f, f')}{\partial \mathbf{x} \partial \mathbf{x}'} = \frac{\partial^2 k(\mathbf{x}, \mathbf{x}')}{\partial \mathbf{x} \partial \mathbf{x}'}. \quad (\text{A.1})$$

For data points in  $3N$  dimensions, this function is matrix-valued, describing the covariances between all pairs of  $3N$  partial derivatives.

It is equivalent (up to sign) to the Hessian of the original scalar-valued kernel function with respect to either one of both inputs, if  $k$  is stationary, i.e.  $k(\mathbf{x}, \mathbf{x}') = \tilde{k}(\mathbf{x} - \mathbf{x}')$ . With  $\delta = \mathbf{x} - \mathbf{x}'$ , we can thus alternatively write

$$\frac{\partial^2 \tilde{k}}{\partial \mathbf{x} \partial \mathbf{x}'} = \frac{\partial^2 \tilde{k}}{\partial \delta^2} \frac{\partial \delta}{\partial \mathbf{x}} \frac{\partial \delta}{\partial \mathbf{x}'} = -\frac{\partial^2 \tilde{k}}{\partial \delta^2} \left(\frac{\partial \delta}{\partial \mathbf{x}}\right)^2 = \frac{\partial^2 \tilde{k}}{\partial^2 \mathbf{x}}. \quad (\text{A.2})$$

#### A.1.1 Matérn covariance derivatives

We use the Hessian of the (isotropic) kernel function from the parametric Matérn family,

$$\begin{aligned} k : C_{v=n+\frac{1}{2}}(d) &= B(d)P_n(d), \\ B(d) &= \exp\left(-\frac{\sqrt{2v}d}{\sigma}\right), \\ P_n(d) &= \sum_{k=0}^n \frac{(n+k)!}{(2n)!} \binom{n}{k} \left(\frac{2\sqrt{2v}d}{\sigma}\right)^{n-k} \end{aligned} \quad (\text{A.3})$$

to obtain the force field kernel as it is used in GDML and sGDML. In this formulation,  $d = \|\mathbf{x} - \mathbf{x}'\|$  is the Euclidean distance between two inputs and  $\sigma$  is the length scale. It can be regarded as a generalization of the universal Gaussian kernel with variable differentiability  $n$ . In our application, we use  $n = 2$  which yields a kernel function that is similar to the Laplacian kernel, but twice differentiable. Nevertheless, we derive the Hessian in full generality here. For notational convenience we write this kernel function as a product of an exponential term  $B(d)$  and a polynomial  $P_n(d)$  of order  $n$ . Then the partial derivatives in the gradient take the form

$$\frac{\partial \kappa}{\partial x_i} = B \frac{\partial P_n}{\partial x_i} + \frac{\partial B}{\partial x_i} P_n. \quad (\text{A.4})$$

They are composed of the first derivatives of the polynomial

$$\frac{\partial P_n}{\partial x_i} = \sum_{k=0}^n \frac{(n+k)!}{(2n)!} \binom{n}{k} \frac{(n-k)(x_i - x'_i)}{d^2} \left( \frac{2\sqrt{2}\sqrt{\nu}d}{\sigma} \right)^{n-k} \quad (\text{A.5})$$

and the first derivative of the exponential function

$$\frac{\partial B}{\partial x_i} = -\frac{\sqrt{2\nu}(x_i - x'_i)}{\sigma d} \exp\left(-\frac{\sqrt{2\nu}d}{\sigma}\right). \quad (\text{A.6})$$

Analogously, the entries in the corresponding Hessian evaluate to

$$\frac{\partial^2 \kappa}{\partial x_i \partial x_j} = B \frac{\partial^2 P_n}{\partial x_i \partial x_j} + \frac{\partial B}{\partial x_i} \frac{\partial P_n}{\partial x_j} + \frac{\partial B}{\partial x_j} \frac{\partial P_n}{\partial x_i} + \frac{\partial^2 B}{\partial x_i \partial x_j} P_n \quad (\text{A.7})$$

using the second derivative of the polynomial

$$\left[ \frac{\partial^2 P_n}{\partial x_i \partial x_j} \right]_{i \neq j} = \sum_{k=0}^n \frac{(n+k)!}{(2n)!} \binom{n}{k} \frac{(n-k-2)(n-k)(x_i - x'_i)(x_j - x'_j)}{d^4} \left( \frac{2\sqrt{2}\sqrt{\nu}d}{\sigma} \right)^{n-k} \quad (\text{A.8})$$

$$\left[ \frac{\partial^2 P_n}{\partial x_i \partial x_j} \right]_{i=j} = \left[ \frac{\partial^2 P_n}{\partial x_i \partial x_j} \right]_{i \neq j} + \sum_{k=0}^n \frac{(n+k)!}{(2n)!} \binom{n}{k} \frac{(n-k)}{d^2} \left( \frac{2\sqrt{2\nu}d}{\sigma} \right)^{n-k}$$

and the second derivative of the exponential

$$\begin{aligned} \left[ \frac{\partial^2 B}{\partial x_i \partial x_j} \right]_{i \neq j} &= \frac{\sqrt{2\nu}(x_i - x'_i)(x_j - x'_j)(\sqrt{2\nu}d + \sigma)}{\sigma^2 d^3} \exp\left(-\frac{\sqrt{2\nu}d}{\sigma}\right) \\ \left[ \frac{\partial^2 B}{\partial x_i \partial x_j} \right]_{i=j} &= \left[ \frac{\partial^2 B}{\partial x_i \partial x_j} \right]_{i \neq j} + \frac{\sqrt{2\nu}}{\sigma d} \exp\left(-\frac{\sqrt{2\nu}d}{\sigma}\right). \end{aligned} \quad (\text{A.9})$$

The matrix-valued force field kernel function  $\text{Hess}(\kappa)$  is then assembled according to Eq. A.7.

## A.2 GDML model derivation

When training a GDML model [144], the following quadratic objective function over  $M$  training points is minimized:

$$\mathcal{L}(\Omega) = \sum_i^M (\mathbf{J}_{\Phi_i} \omega_i - \mathbf{F}_i)^2 + \lambda \|\Omega\|^2 \quad (\text{A.10})$$

Here,  $\mathbf{J}_{\Phi_i} = \mathbf{J}_{\Phi(\mathbf{x}_i)}$  are the  $3N \times F$  Jacobi matrices of a non-linear feature transform of the training geometries  $\mathbf{x}_i$  into  $F$ -dimensional space, weighted by parameter vectors  $\omega_i$ .  $\mathbf{F}_i$  contains the atomic forces (e.g. negative energy gradients) corresponding to each geometry, stacked into a vector. For the sake of simplicity we will assume that the geometry encoded in  $\mathbf{x}_i$  is simply represented in Cartesian coordinates, but we will introduce a descriptor in the final formulation of the model. In addition, the norm of the coefficients  $\Omega = [\omega_1^\top, \dots, \omega_M^\top]^\top$  is penalized as way to regularize the complexity of the solution. The regularization strength is tuned via a hyper-parameter  $\lambda$ .

To find the minimum, we set the derivative of this cost function to zero:

$$\frac{\partial \mathcal{L}}{\partial \Omega} = 2 \sum_i^M \mathbf{J}_{\Phi_i}^\top (\mathbf{J}_{\Phi_i} \omega_i - \mathbf{F}_i) + 2\lambda \Omega = \mathbf{0} \quad (\text{A.11})$$

giving

$$\begin{aligned} \sum_i^M \mathbf{J}_{\Phi_i}^\top \mathbf{J}_{\Phi_i} \omega_i - \mathbf{J}_{\Phi_i}^\top \mathbf{F}_i &= \lambda \Omega \\ \rightarrow \Omega &= \left( \lambda \mathbb{I}_F + \sum_i^M \mathbf{J}_{\Phi_i}^\top \mathbf{J}_{\Phi_i} \right)^{-1} \sum_j^M \mathbf{J}_{\Phi_j}^\top \mathbf{F}_j \end{aligned} \quad (\text{A.12})$$

We will now aggregate the Jacobi matrices for all training points into a large matrix  $\mathbf{J}_\Phi = [\mathbf{J}_{\Phi_1}, \dots, \mathbf{J}_{\Phi_M}]$  of dimension  $3NM \times F$  and use pure matrix notation. We continue by applying the Woodbury matrix identity:

$$\begin{aligned}\Omega &= (\mathbf{J}_\Phi^\top \mathbf{J}_\Phi + \lambda \mathbb{1}_F) \mathbf{J}_\Phi^\top \mathbf{F} \\ &= \mathbf{J}_\Phi^\top \underbrace{(\mathbf{J}_\Phi \mathbf{J}_\Phi^\top + \lambda \mathbb{1}_{3NM})^{-1} \mathbf{F}}_{\mathbf{A}}\end{aligned}\tag{A.13}$$

This way we can solve the linear system above in  $3NM \ll F$ . Forces for new inputs are then computed by evaluating  $\mathbf{F}_{\text{new}} = \mathbf{J}_{\Phi_{\text{new}}} \Omega$ , which can also be written as

$$\mathbf{F}_{\text{new}} = \mathbf{J}_{\Phi_{\text{new}}} \mathbf{J}_\Phi^\top \mathbf{A}.\tag{A.14}$$

This is helpful because  $\mathbf{J}_\Phi \mathbf{J}_\Phi^\top$  and  $\mathbf{J}_{\Phi_{\text{new}}} \mathbf{J}_\Phi^\top$  are (co-)variances between derivative observation in feature space and we can apply the 'kernel trick' to express them via a kernel function that foregoes an explicit mapping [81]. We write the Jacobian  $\mathbf{J}_\Phi = \nabla \Phi^\top$  as the outer product of feature transform and derivative operator and then

$$\begin{aligned}\mathbf{J}_\Phi \mathbf{J}_\Phi^\top &= \nabla \Phi^\top (\nabla \Phi^\top)^\top \\ &= \nabla \underbrace{\Phi^\top \Phi}_{\kappa} \nabla^\top\end{aligned}\tag{A.15}$$

to substitute the inner product of feature transformations with a scalar-valued kernel function. The force field kernel in GDML is thus a matrix with entries  $k_{ij} = \partial^2 k / \partial \mathbf{x}_i \partial \mathbf{x}_j$  (see Eq. A.1). Finally, we rewrite Eq. A.14 in a more verbose way and obtain with  $\mathbf{A} = [\alpha_1^\top, \dots, \alpha_M^\top]^\top$

$$\hat{\mathbf{f}}_{\mathbf{F}}(\mathbf{x}) = \sum_i^M \sum_j^{3N} (\alpha_i)_j \frac{\partial}{\partial x_j} \nabla_{\mathbf{x}} k(\mathbf{x}, \mathbf{x}_i)\tag{A.16}$$

for the force field model, where  $\partial / \partial x_j$  is the partial derivative with respect to the  $j$ -th component of the input vector. The corresponding reconstruction of the potential energy surface is recovered up to a constant via integration:

$$\hat{f}_E(\mathbf{x}) = \sum_i^M \sum_j^{3N} (\alpha_i)_j \frac{\partial}{\partial x_j} k(\mathbf{x}, \mathbf{x}_i) + c.\tag{A.17}$$

Due to linearity of integration, the expression for the energy predictor  $\hat{f}_E(\mathbf{x})$  is identical up to the second derivative operator acting on the kernel function. The inverted sign of the energy is accounted for by use of the Hessian in Eq. A.16.

### A.2.1 Integration constant

The sum of squared deviations between predicted and reference energy at every training point is minimized to estimate the integration constant. We minimize the loss function

$$\begin{aligned}\mathcal{L}(c) &= \sum_i^M (\int \hat{\mathbf{f}}_F(\mathbf{x}_i) dx - E_i)^2 \\ &= \sum_i^M (-\hat{f}_E(\mathbf{x}_i) + c - E_i)^2,\end{aligned}\tag{A.18}$$

which unsurprisingly gives the mean of energy deviations at every training point

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial c} &= 2 \sum_i^M c - (E_i + \hat{f}_E(\mathbf{x}_i)) = 0 \\ &= 2Mc - 2 \sum_i^M E_i + \hat{f}_E(\mathbf{x}_i) \\ &\rightarrow c = \frac{1}{M} \sum_i^M E_i + \hat{f}_E(\mathbf{x}_i)\end{aligned}\tag{A.19}$$

as the best estimate for the integration constant.

### A.2.2 Bi-partite matching cost matrix

To match a pair of molecular graphs, we solve the optimal assignment problem for the eigenvectors of both adjacency matrices using the Hungarian algorithm [138]. As input, this algorithm expects a matrix with all pairwise assignment costs  $\mathbf{C}_M = -\mathbf{M}$ , which is constructed as the negative overlap matrix from Eq. 4.2. A penalty matrix with entries  $(\mathbf{C}_Z)_{ij} = \text{abs}(z_i - z_j)\epsilon$  is added to prevent the assignment of non-identical nuclei for sufficiently large  $\epsilon > 0$ . Here,  $\mathbf{Z} = [z_1, \dots, z_N]^\top$  are the charges for each nuclei in the molecule. The final const matrix is then

$$\mathbf{C} = \mathbf{C}_M + \mathbf{C}_Z.\tag{A.20}$$

### A.2.3 Permutation matrices notation

Throughout this thesis, we use permutation matrices  $\mathbf{P}(\tau) \equiv \mathbf{P}$  in column representation, obtained by permuting the columns of the identity matrix of dimension  $N \times N$ , such that  $(\mathbf{P})_{ij} = 1$  if  $j = \tau(i)$  and 0 otherwise. The multiplication  $\mathbf{Px}$  will hence permute the rows of the column vector  $\mathbf{x}$ . We do not distinguish between permutation matrices acting on different representations of the same data. While  $\mathbf{PR}^\top$  permutes the atoms of

a molecule represented by a  $3 \times N$  matrix  $\mathbf{R} = [\mathbf{r}_1, \dots, \mathbf{r}_N]$  of Cartesian coordinates,  $\mathbf{P}_{\mathcal{X}}$  represents the same permutation, but acting on a linearized input space  $\mathcal{X}$  representation  $\mathbf{R}_{\mathcal{X}} = [\mathbf{r}_1^\top, \dots, \mathbf{r}_N^\top]^\top$  of dimension  $3N \times 1$ .

# Appendix B

## Numerical results

### B.0.1 Energy-trained baseline model

**Table B.1** Accuracy of the converged energy-based predictor. All training set sizes  $M$  are chosen to match the complexity of the optimization problem in the corresponding force model (number of samples times number of partial derivatives). Energy errors are in kcal mol $^{-1}$ , force errors in kcal mol $^{-1}$  Å $^{-1}$ .

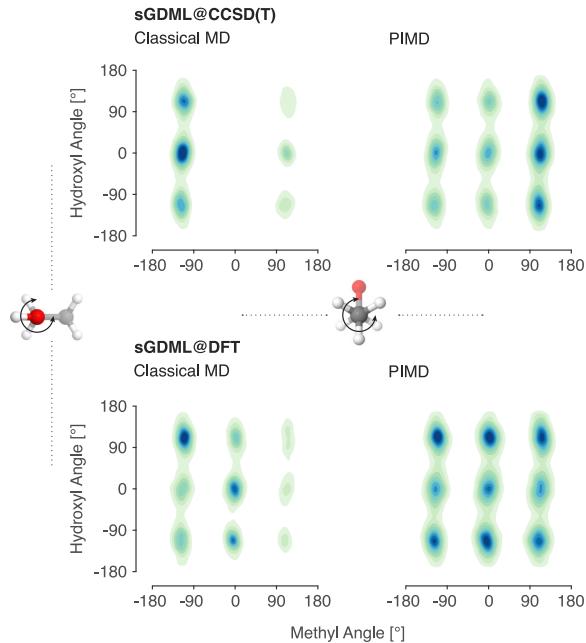
Dataset	$M$	Energy error		Force error		Magnitude		Angle	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Benzene	36K	0.04	0.06	0.80	1.16	1.00	1.38	0.0196	0.0350
Uracil	36K	0.03	0.03	0.44	0.62	0.45	0.54	0.0092	0.0148
Naphthalene	54K	0.02	0.03	0.40	0.55	0.43	0.52	0.0079	0.0129
Aspirin	63K	0.03	0.04	1.51	2.12	0.98	1.28	0.0220	0.0311
Salicylic acid	48K	0.10	0.13	0.45	0.63	0.39	0.51	0.0052	0.0090
Malonaldehy.	27K	0.11	0.16	0.83	1.16	0.80	1.05	0.0128	0.0230
Ethanol	27K	0.09	0.14	0.76	1.07	0.92	1.22	0.0116	0.0246
Toluene	45K	0.06	0.08	0.52	0.71	0.50	0.61	0.0087	0.0146

### B.0.2 Non-conservative baseline model

**Table B.2** Accuracy of the naïve force predictor based on a training set size of  $M = 1000$ . This model learns all output components independently, without constraining the predicted forces to be energy conserving. It is identical to the GDML model in all other aspects. Energy prediction errors are not available, because the resulting force fields are not integrable.

Dataset	Force error [kcal mol <sup>-1</sup> Å <sup>-1</sup> ]					
			Magnitude		Angle	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
Benzene	14.67	20.01	19.38	22.39	0.4496	0.5048
Uracil	5.91	11.29	1.90	2.84	0.1341	0.1859
Naphthalene	6.50	11.16	2.17	3.13	0.1255	0.1748
Aspirin	8.80	12.95	6.64	9.29	0.1481	0.1948
Salicylic acid	6.13	11.28	2.36	3.35	0.1183	0.1662
Malonaldehyde	19.98	27.35	17.99	22.79	0.4157	0.4664
Ethanol	18.15	24.78	24.12	30.89	0.3938	0.4506
Toluene	15.66	23.29	11.85	16.09	0.3583	0.4109

### B.0.3 Probability distributions of the dihedral angles in ethanol (sGDML@CCSD(T) versus sGDML@DFT)



**Figure B.1** Comparison of probability distributions of the dihedral angles (methyl rotor vs. hydroxyl rotor) of ethanol obtained from classical and path-integral MD simulations at 300 K. We contrast the results from a sGDML model trained on CCSD(T) versus DFT reference calculations. The inclusion of nuclear quantum effects improves the sampling of the PES for both levels of theory. The sampling was performed during 0.5 ns of simulation, using 16 beads for PIMD.

**Table B.3** Properties of MD datasets that were used for numerical testing.

Dataset	Formula	Size	Energies [kcal mol <sup>-1</sup> ]			Forces [kcal mol <sup>-1</sup> Å <sup>-1</sup> ]		
			Range	Min. ( $\times 10^4$ )	Max. ( $\times 10^4$ )	Range	Min.	Max.
Benzene	C <sub>6</sub> H <sub>6</sub>	627000	20.2	-14.653	-14.651	266.3	-126.677	139.626
Uracil	C <sub>4</sub> H <sub>4</sub> N <sub>2</sub> O <sub>2</sub>	133000	39.9	-26.012	-26.008	476.6	-237.381	239.249
Naphthalene	C <sub>10</sub> H <sub>8</sub>	326000	48.4	-24.192	-24.187	452.9	-217.207	235.688
Aspirin	C <sub>9</sub> H <sub>8</sub> O <sub>4</sub>	211000	47.0	-40.676	-40.671	404.1	-195.664	208.454
Salicylic acid	C <sub>7</sub> H <sub>6</sub> O <sub>3</sub>	320000	47.5	-31.105	-31.100	453.8	-236.086	217.687
Malonaldehyde	C <sub>3</sub> H <sub>4</sub> O <sub>2</sub>	993000	43.8	-16.751	-16.747	570.7	-286.050	284.602
Ethanol	C <sub>2</sub> H <sub>6</sub> O	555000	35.5	-9.721	-9.717	432.0	-211.104	220.900
Toluene	C <sub>7</sub> H <sub>8</sub>	442000	46.9	-17.024	-17.019	425.6	-212.984	212.617
Paracetamol	C <sub>8</sub> H <sub>9</sub> NO <sub>2</sub>	106000	57.6	-32.230	-32.294	425.9	-217.473	208.424
Azobenzene	C <sub>12</sub> H <sub>10</sub> N <sub>2</sub>	211000	51.1	-35.884	-35.879	401.1	-208.971	192.119

# Appendix C

## Software implementation

**Partial results of the presented work have been published in:**

- Chmiela, S., Sauceda, Poltavsky, I., H. E., Müller, K.-R., Tkatchenko, A. (2019) "sGML: Constructing Accurate and Data Efficient Molecular Force Fields Using Machine Learning". In: *Computer Physics Communications*, 10.1016/j.cpc.2019.02.007

### C.0.1 User Input

The essential ingredient for training and validating an sGML model is a user-provided reference dataset, specifically a set of Cartesian geometries with corresponding total energy and atomic-force labels. Those labels can be generated from any level of theory, e.g. *ab initio* calculations, any method derived from DFT (e.g. Kohn-Sham or other orbital-free variants) or even regular FFs, since the sGML model is not biased towards a specific kind of reference data. Force labels are needed, because our approach implements energy conservation as an explicit linear operator constraint, by modeling the FF reconstruction  $\hat{\mathbf{f}}_F = -\nabla \hat{f}_E$  as the transformation of an underlying energy model [144]. Force learning affords data-efficiency advantages, as they are more informative per example, while being generally cheaper to compute analytically than collecting the same derivative information via numerical approximation from energy examples. Since forces are true quantum-mechanical observables, they preserve all information regarding the quantum nature of the system and therefore pass it on to the model.

A key consideration when composing a reference dataset, is the choice of sample region on the PES. Generally, we want to keep the covered area tight, avoiding the inclusion

**Table C.1** Training times for various sGDML models based on 1000 reference data using an analytic solver on a Intel Xeon E5-2640 CPU at 2.40GHz. For the same models we also list the force and energy prediction performances for sequential evaluations of individual geometries and batch evaluations of 1000 geometries on a 2.8 GHz Intel Core i7 notebook.

<b>Dataset</b>	<b>Training</b> [min]	<b>Prediction</b> [geom./sec]	
		Sequential	Batch
Benzene	1.9	434.7	676.3
Uracil	2.0	1103.9	5326.5
Naphthalene	5.8	446.9	693.1
Aspirin	9.5	295.0	430.3
Salicylic acid	4.7	894.2	3652.1
Malonaldehyde	2.5	1001.0	3071.1
Ethanol	2.4	826.2	2557.4
Toluene	3.6	326.3	430.4
Paracetamol	7.9	208.5	247.1
Azobenzene	17.8	182.6	214.0

of configuration space that will not be explored in the specific application of the trained model. With that being said, we also aim to limit the need for extrapolation, which usually carries a performance penalty. All isomeric conformers of interest, including the transition pathways, need to be well represented in the dataset.

The sGDML model is unit-agnostic, meaning that the energy and force predictions will simply inherit the units of the training labels. Particular attention should be paid to ensuring that the unit of force (e.g. kcal mol<sup>-1</sup>Å<sup>-1</sup>) is consistent with the unit of energy (e.g. kcal mol<sup>-1</sup>) and the unit of length (e.g. Å) used in the provided energy labels and geometries, respectively. While the model will quietly convert different length units between input and output, it is not able to adapt the energy unit. As a good practice, we strongly advise against mixing units in the same dataset, since an implicit unit conversion within the trained model is not a behavior that the user expects.

All geometries within a dataset must use a consistent atom indexing and every derived model should be queried using the same order. This is because the invariance of sGDML models is restricted to permutational symmetries that are physically feasible and statistically relevant, which does not include the full symmetry group of the molecule in general. Arbitrarily indexed query geometries may not fall within the set of interchangeable representations and hence yield undefined outputs. While it would be technically

straightforward to extend the sGDML prediction routine to support randomly index inputs, we deliberately omitted that functionality in favor of evaluation speed.

We use NumPy binary files as the native file format for our application, but include converters from and to various popular plaintext formats. Support for additional file types can be easily extended, by using one of the included conversion scripts as a template. One of the main reasons for using a custom file format is the inclusion of metadata that makes the origin of each model traceable and data integrity verifiable.

## C.1 Usage

Our program includes a set of convenience routines that assist the user in reconstructing a sGDML model from beginning to end. It will walk the user through the complete process of data sampling, symmetry recovery, training with hyper-parameter optimization and validation to generate a ready-to-use model. Greater control over this procedure may be taken by running the involved subroutines individually, either via the CLI or using the Python interface of the `train` and `predict` modules. From the CLI, the assisted training process is initiated by simply calling

```
$ sgdml all <dataset_file> <n_train> <n_validate> [<n_test>] \
    [--sig <list_or_range>]
```

with a path to the reference dataset as the argument. The parameter `n_train` specifies how many data points are used for training: larger training sets yield more accurate models, but at increased computational cost. During model selection, the performance of a model candidate is assessed based on the comparison of `n_validate` predicted forces and energies with the true labels. Optionally, the number of validation points `n_test` can be specified, otherwise this parameter will be set to the maximum value for the best possible final estimate of the generalization error. Large validation and test datasets are desirable as they only increase computational cost marginally, while yielding better error estimates. Additionally, the search grid for the hyper-parameter  $\sigma$  can be specified as a space-delimited list (`-sig <s1> <s2> ... <sN>`), or a range of evenly spaced values within a given interval (`-sig <start>:<step>:<stop>`), or a combination of both.

Training, validation and test dataset are sampled from the provided bulk dataset without overlap, unless individual datasets (`-v <validation_dataset>` and/or `-t <test_dataset>`) are specified. For optimal prediction performance, it is crucial for the training set to represent the distribution the model will encounter. Likewise, we can only reliably assess its expected generalization error if we validate and test on representative datasets. With the assumption that the bulk dataset adequately describes

the molecular configuration space that will be visited in the application of trained model, our sampling method automatically extracts stratified subsets that properly follow the estimated probability energy density function of the full dataset.

### C.1.1 Training

Every sGML model emerges from a *training task*, which is a file that packages the configuration for a particular training run, including the indices of the training and validation data points, the symmetries of the molecule, as well as a particular hyper-parameters choice. A batch of training tasks for a range of hyper-parameters is generated with the `create`-command

```
$ sgdml create <dataset_file> <n_train> <n_valid> [-sig <list_or_range>]
```

which sets up a directory containing the corresponding task files. All parameters used here, have been introduced previously. This routine will sample training and validation datasets form the provided bulk dataset, recover the symmetries in the geometry and package everything into individual tasks for each  $\sigma$  in the provided range.

Using the `train`-command and the task directory created in the previous step, the training process is invoked with

```
$ sgdml train <task_dir_or_file>
```

For each training task, this resource intensive process creates a model candidate in the same directory. Alternatively, a path to a single file can be passed to execute an individual task, which is useful when submitting batch jobs to distributed computing environments. Parallelization is easy, because the full training dataset is stored in each task file, so that each training job can be performed in isolation, without referencing the potentially large common bulk dataset. All model candidates are stored in the task directory. In the next step, we will evaluate the performance of each model on the validation set and select the leading hyper-parameter choice.

The validation process is invoked via

```
$ sgdml validate <model_dir_or_file> <dataset_file>
```

for the whole directory or individual models. As the validation dataset has been predetermined during training task creation and stored in the model, we must pass the originally referenced dataset, otherwise the program can not continue.

Finally, we keep the best performing model from the full set of candidates based on the lowest root-mean-square error (RMSE), which is the metric used in the objective function for the parameterization of the model.

---

```
$ sgdml select <model_dir>
```

Because the validation dataset was used to determine the optimal hyper-parameters, it participated in the training process, very much like the actual training data. To estimate the generalization behavior of the final model in an unbiased way, we will hence use a third independent test dataset and measure its performance once again by calling

```
$ sgdml test <model_file_or_dir> <dataset_file> [<n_test>]
```

The reliability of this estimate can be improved by using as many data points as available. Omitting the last parameter selects all points for the dataset that were not involved in the training process of the model.

### Memory requirements

In this implementation, we train the sGDML analytically, i.e. by solving a linear system in closed form. While this approach is faster and more accurate than numerical methods (i.e. gradient descent), it is also highly memory demanding. Analytic solvers require the complete kernel matrix to be kept in memory at once. With  $(M \times 3N)^2$  double precision (8 byte) entries, it dominates the memory footprint of the training process. Those numbers can be used as a rough guideline for choosing a suitable hardware platform.

### C.1.2 Inference

The sGDML force estimator trained on  $M$  reference geometries, each with  $3N$  partial derivatives and  $S$  symmetry transformations, takes the form

$$\hat{\mathbf{f}}_{\mathbf{F}}(\mathbf{x}) = \sum_i^M \sum_l^{3N} \sum_q^S (\mathbf{P}_q \alpha_i)_l \frac{\partial}{\partial x_l} \nabla k(\mathbf{x}, \mathbf{P}_q \mathbf{x}_i). \quad (\text{C.1})$$

Due to linearity of integration, the corresponding energy predictor is identical up to the second derivative operator on the kernel function, which allows the simultaneous computation of both quantities without computational overhead. It is easy to see that this expression offers a lot of potential for parallelization, which we fully exploit in our code. The amount of concurrent work performed by our implementation is governed by two optional parameters that depend on the host hardware: the number of parallel processes `num_processes` and the chunk size `chunk_size` in which data items are processed at once. A chunk refers to a vectorized operation that is passed as one big task to Python's underlying high-performance libraries. Both parameters can be automatically tuned for optimal performance by simply calling

---

```
gdml_predict.set_opt_parallelism(n_reps=100)
```

after instantiation of the prediction class. This routine runs a small benchmark that tests feasible configurations by repeatedly calling the `predict`-function while measuring execution time. The number of repetitions `n_reps` can be increased to improve the reliability of the benchmark, but also increase its duration. Because this routine takes a few seconds to complete, its runtime is only amortized when followed by a large amount of FF evaluations.

Once a sGDML model is trained, it can be integrated into external programs via the `gdml_predict` module. A new model instance is created using

```
gdml_predict = GDMLPredict(model, [chunk_size], [num_processes])
```

Force and energy predictions for a geometry are then simply generated using

```
r,_ = io.read_xyz(geometry_path)
e,f = gdml_predict.predict(r)
```

This function also accepts a batch of geometries at once, which is useful in applications where multiple independent geometries need to be computed at the same time, e.g. path integral molecular dynamics with a variety of thermostats and statistical ensembles, or in transition path search.

## C.2 Example Application: Paracetamol

To outline the process of FF construction from beginning to end, we consider the paracetamol molecule as an example. Our aim is to create a model for use in long time-scale MD simulations at room temperature (300 K) and an accuracy level of PBE0+MBD. This application is interesting, because a direct sampling at this level of theory would be prohibitively expensive and require hundreds of millions of CPU hours.

First, we will generate a minimal training set that captures all relevant geometrical configurations. Unreliable predictions are prevented by ensuring that the planned simulations never wander off the regime of configuration space that is covered by training data. In the same vein, we want to exclude sections of the PES that will never be queried in the actual application of the trained model as this would unnecessarily complicate the reconstruction task. Here, we use a sufficiently long MD trajectory at a higher temperature of 500 K to provide the appropriate coverage. The actual training set is then constructed as a small subset of the original trajectory whose energies follow the Maxwell-Boltzmann distribution (see Figure C.1). Foregoing a prohibitively expensive long timescale MD simulation at the theory level DFT-PBE0+MBD with a large basis set, we use a cheap

DFT-PBE+TS trajectory as the geometry sampling method and only recompute the corresponding energy and force labels for the small subset of selected training points at the higher level of theory.

We remark that this sampling scheme is based on the assumption that the PBE+TS energy surface is a good proxy for the topographical structure of the PBE0+MBD surface, as overly strong approximations may yield a sampling profile that misses important features. It is furthermore important to choose a fine-enough time step for the MD simulation, so that the relevant areas of configuration space are sampled with correct probability. As a rule of thumb we use 1/10 of the period of the highest frequency oscillator in the system (i.e. hydrogen stretching frequencies). For example, if the highest vibration frequency in paracetamol is 3600 wavenumbers (i.e. period of 9.3 fs), then our time step works out to  $\sim 1$  fs. We have obtained the simulated trajectory as a dataset file in *extended XYZ* format, which contains our collected geometries with corresponding forces in additional columns and the energy labels in the comment line. The next step is to convert it to the native sGML binary format, which is the basis for all forthcoming steps:

```
$ sgml_dataset_from_xyz.py paracetamol.xyz
```

With the resulting dataset file `d_paracetamol.npz`, we will now run the fully automated sGML training assistant which will walk us through all steps necessary to obtain a fully trained and tested model:

```
$ sgml all d_paracetamol.npz 1000 500
```

We have chosen to reconstruct the PES using 1000 training points, sampled from the provided dataset file, and to use 500 separate geometries to validate the performance of our candidates during model selection. We omit the argument for the number of test data points, as we want the program to test the resulting model on all remaining data points from the set. The assistant will now automatically split the dataset, train a model for a series of hyper-parameter candidates, validate all models, select the most accurate one, finally test it and output a model file `m_paracetamol.npz`. Using only this file, we can easily use the newly reconstructed paracetamol force field in existing applications:

```
import numpy as np
from sgml.predict import GDMLPredict
model = np.load('m_paracetamol.npz')
gdml = GDMLPredict(model)
make predictions using
e,f = gdml.predict(r)
```

and

Interfaces to two popular FF simulation engines are already included with our software package: a Calculator for ASE [130] and a i-PI [131] ForceField-object. ASE enables various standard simulation tasks including structure optimization, vibrational analysis, molecular dynamics simulations and nudged elastic band calculations, whereas i-PI implements path integral MD to study molecular phenomena that are driven by nuclear quantum effects and a wide variety of sophisticated methods to compute quantum observables [131]. In the following, we present in step-by-step fashion how to integrate sGDML with ASE and i-PI and demonstrate practical applications for which it is useful.

### ASE: Normal mode analysis

We will now proceed with a normal mode analysis of paracetamol using ASE. After attaching the SGDMLCalculator to the Atoms-object, we relax an initial geometry `paracetamol.xyz` with the BFGS optimizer. Then we simply calculate the vibrational modes in the harmonic approximation using `Vibrations`:

```
from sgdml.intf.ase import SGDMLCalculator

from ase.io.xyz import read_xyz
from ase.optimize import BFGS
from ase.vibrations import Vibrations

mol = read_xyz('paracetamol.xyz').next()

sgdml = SGDMLCalculator('m_paracetamol.npz')
mol.set_calculator(sgdml)

vib = Vibrations(mol)
vib.run()
vib.summary()
vib.write_jmol()
vib.clean()
```

This process will output a table with all vibrational frequencies, but also write a file `vib.xyz` that can be imported into Jmol to visualize the vibrational modes. To validate the accuracy of our normal mode frequencies, we compare directly with the spectrum from DFT-PBE0+MBD using FHI-aims. Figure C.1 outlines the difference between the two sets of normal mode frequencies showing a maximum deviation of only  $\sim 4 \text{ cm}^{-1}$ . This

result evinces the robustness of our model given that no explicit information was provided regarding the normal modes.

### i-PI: Molecular dynamics

In physics and chemistry many of the molecular phenomena are driven by nuclear quantum effects (NQE), in particular for protons, this nuclear delocalization gives rise to numerous quantum phenomena e.g. zero-point energy and tunneling. Different methods have been developed to incorporate such effects in the BO approximation, path integral molecular dynamics (PIMD) being one of the most widely used. The i-PI software offers an efficient PIMD implementation including state-of-the-art integrators and thermostats [131]. The sGMDL model can be easily incorporated in i-PI as a force and energy provider class `FFsGMDL()`. Once the sGMDL force field is available in i-PI, running a MD simulation is straightforward. A minimal set up requires the initial coordinates `paracetamol.xyz`, the sGMDL model file `m_paracetamol.npz` and the input file `input.xml` which specifies the parameters of the simulation e.g. force field, ensemble, temperature, thermostat, integration step, etc. Then running the MD simulations requires just one simple command:

```
python i-pi input.xml
```

From these MD simulations, we can compute a wide variety of properties such as finite temperature vibrational spectra, free energy surfaces, radial distribution functions, energies, heat capacities, etc. As an example, we analyze the effect of the temperature on the vibrational spectrum. Figure C.1 shows the comparison of the normal modes and vibrational spectra at different temperatures (50K and 450K) using classical MD simulations. From this comparison, the effect of the anharmonicities at high temperatures is evident, given the noticeable red-shift in the frequency peaks. Beyond classical MD, we can explore the NQE by running PIMD in i-PI. An important measure of the NQE is the interatomic distance distributions,  $h(r)$ , shown in Figure C.1. The deviation between the two curves for classical MD and PIMD gives the magnitude of the delocalization of mean pair distances. This analysis provides an idea of the delocalization of the atomic nuclei in the molecule due to NQE.



**Figure C.1** Top: From a provided dataset of molecular geometries with corresponding energy and force labels, our sGMDL implementation creates a fully cross-validated FF model. Bottom: This lightweight model can then be used to speed up various PES sampling intensive applications, like molecular dynamics or the computation of transition paths. Interfacing ASE allows for easy computation of normal modes, vibrational spectra or nudged elastic band optimizations (middle row). Our interface to i-PI enables path integral molecular dynamics simulations (PIMD), which we use to compute the free energies and interatomic distance distributions  $h(\mathbf{r})$  with classical MD and PIMD (bottom row).