Grappa - A Machine Learned Molecular Mechanics Force Field

Leif Seute [1] **Eric Hartmann** [1 2] **Jan Stühmer** [1 3] **Frauke Gräter** [1 2]

Abstract

Simulating large molecular systems over long timescales requires force fields that are both accurate and efficient. In recent years, E(3) equivariant neural networks have lifted the tension between computational efficiency and accuracy of force fields, but they are still several orders of magnitude more expensive than established molecular mechanics (MM) force fields. Here, we propose Grappa, a machine learning framework to predict MM parameters from the molecular graph, employing a graph attentional neural network and a transformer with symmetry-preserving positional encoding. The resulting Grappa force field outperformstabulated and machine-learned MM force fields in terms of accuracy at the same computational efficiency and can be used in existing Molecular Dynamics (MD) engines like GROMACS and OpenMM. It predicts energies and forces of small molecules, peptides, RNA and — showcasing its extensibility to uncharted regions of chemical space — radicals at state-of-the-art MM accuracy. We demonstrate Grappa's transferability to macromolecules in MD simulations from a small fast folding protein up to a whole virus particle. Our force field sets the stage for biomolecular simulations closer to chemical accuracy, but with the same computational cost as established protein force fields.

1. Introduction

In recent years, advances in geometric deep learning have led to the development of highly accurate machine learned force fields, reshaping the field of computational chemistry and Molecular Dynamics (MD) simulations. E(3) equivariant neural networks (Musaelian et al., 2022; Batatia et al., 2023; Unke et al., 2022; Schütt et al., 2021) are capable of predicting energies and forces of small molecules to great accuracy with lower computational cost than quantum mechanical (QM) methods. However, these models are several orders of magnitude more expensive than Molecular Mechanics (MM) force fields, which employ a simple physics-inspired functional form to parametrize the potential energy surface of a molecular system, hence trading off accuracy in favor of efficiency. For MD simulations of large systems such as proteins and polynucleotides, MM force fields are well established and widely used. Established MM force fields rely on lookup tables with a finite set of atom types characterized by chemical properties of the atom and its bonded neighbors for parameterization. Recently, the Espaloma approach (Wang et al., 2022) has demonstrated that machine learning can be used to increase the accuracy of MM force fields by learning to assign parameters based on a graph representation of the molecule with chemical properties that rely on expert knowledge, such as orbital hybridization states or formal charge, as input features.

In this work, we propose a novel machine learning framework, Grappa (Graph Attentional Protein Parametrization), to learn MM parameters directly from the molecular graph, improving accuracy on a broad range of chemical space and eliminating the need for hand-crafted features. Grappa employs a graph attentional neural network to construct atom embeddings capable of representing chemical environments based on the 2D molecular graph, followed by a transformer (Vaswani et al., 2023) with symmetry-preserving positional encoding. Since MM parameters only have to be predicted once per molecule, Grappa can be incorporated into highly optimized MM engines such as GROMACS (Abraham et al., 2015) and OpenMM (Eastman et al., 2013). This allows energy and force evaluations with the same computational cost as traditional force fields, at state-of-the-art MM accuracy.

We show that Grappa outperforms traditional MM force fields and the machine-learned MM force field Espaloma on the Espaloma dataset (Takaba et al., 2023), which contains over 14,000 molecules and more than one million states, covering small molecules, peptides and RNA. Since Grappa uses no hand-crafted chemical features, it can be

[1]Heidelberg Institute for Theoretical Studies, Schloss-Wolfsbrunnenweg 35, 69118 Heidelberg, Germany [2]Interdisciplinary Center for Scientific Computing, Heidelberg University, INF 205, 69120 Heidelberg, Germany [3]Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Kaiserstr. 12, 76131 Karlsruhe, Germany. Correspondence to: Leif Seute <leif.seute@h-its.org>.

Proceedings of the 41st International Conference on Machine Learning, AI4Science Workshop, Copyright 2024 by the author(s).

extended to uncharted regions of chemical space, which we demonstrate on the example of peptide radicals. Grappa is transferable to individual macromolecules and assemblies such as proteins and viruses, which exhibit similar dynamics as established force fields. Starting from an unfolded initial state, MD simulations of small proteins parametrized by Grappa recover experimentally determined folding structures of small proteins, suggesting that Grappa captures the physics underlying protein folding. We demonstrate the efficiency of Grappa, which is inhereted from MM, by simulating a system of one million atoms with the proposed force field on a single GPU, with a similar number of timesteps per second as a highly performant E(3) equivariant neural network (Musaelian et al., 2022) on over 4,000 GPUs.

1.1. Molecular Mechanics

In MM, the potential energy of a system with a given molecular graph is expressed as a sum of contributions from different interactions. Bonded interactions are described by functions of E(3)-invariant internal coordinates such as the lengths r_{ij} of bonds between two atoms, angles θ_{ijk} between three consecutive atoms and dihedrals ϕ_{ijkl} of two planes spanned by four atoms. For the dihedrals, one considers interactions between four atoms that are either consecutively bonded (torsions) or where three atoms are bonded to a central atom (impropers), which do not reflect an independent degree of freedom but are used to maintain planarity of certain chemical groups. One commonly uses harmonic potentials for bond stretching and angle bending and a periodic function for the dihedral potential. The potential energy of all interactions along bonds then is given by

$$\begin{aligned} E_{\text{bonded}}(\mathbf{x}) = &\sum_{(ij)\in\text{bonds}} k_{ij}(r_{ij} - r_{ij}^{(0)})^2 \\ &+ \sum_{(ijk)\in\text{angles}} k_{ijk}(\theta_{ijk} - \theta_{ijk}^{(0)})^2 \\ &+ \sum_{(ijkl)\in\text{dihedrals}} \sum_{n=1}^{n_{\text{periodicity}}} k_{ijkl}\cos(n\phi_{ijkl}), \end{aligned} \quad (1)$$

with the equilibrium values (of bonds and angles) $r_{ij}^{(0)}$ and $\theta_{ijk}^{(0)}$, and the force constants k_{ij}, k_{ijk} and k_{ijkl}, as the set of MM parameters, which we denote as

$$\boldsymbol{\xi} \equiv \{\xi_{ij\ldots}^{(l)} \,|\, l \in \{\text{bonds, angles, torsions, impropers}\}\}. \quad (2)$$

For the periodic dihedral potential, a common choice is a Fourier series with the constraint that the dihedral potential is extremal at and symmetric around zero, which eliminates the need for sine terms. Additionally, atom pairs that are not included in such N-body bonded interaction terms contribute to the potential energy through pairwise nonbonded interaction, typically described by Lennard-Jones and Coulomb potentials.

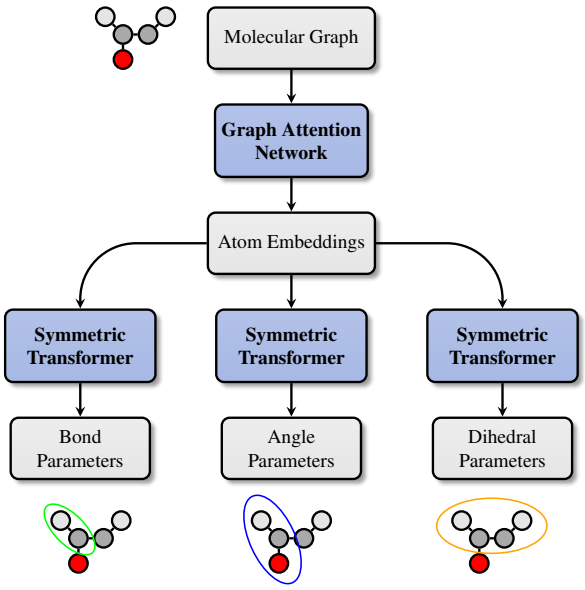


Figure 1: Grappa predicts MM parameters in two steps. First, atom embeddings are predicted from the molecular graph with a graph neural network. Then, transformers with symmetric positional encoding followed by permutation invariant pooling maps the embeddings to MM parameters with desired permutation symmetries. Once the MM parameters are predicted, the potential energy surface can be evaluated with MM-efficiency for different spatial conformations.

Traditional MM force fields define a finite set of atom types determined by hand-crafted rules, which are used to assign the free parameters $\{k_{ij}, r_{ij}^{(0)}, \ldots\}$ based on lookup tables for possible combinations of atom types. In Grappa, we replace this scheme by learning the parameters from the molecular graph directly, which allows for a more flexible and transferable description of the potential energy surface.

A fundamental limitation of standard MM is the assumption of a constant molecular graph topology, which is enforced by the use of harmonic bond potentials. While this restricts accuracy and prohibits the description of bond-changing chemical reactions, the physical interpretability of the potential energy function ensures that simulated systems remain stable, even in states that are poorly described by the force field.

2. Grappa

Inspired by the atom typing with hand-crafted rules in traditional MM force fields and in analogy to Wang et al. 2022, Grappa first predicts d-dimensional atom embeddings,

$$\boldsymbol{\nu} = \{\nu_i \in \mathbb{R}^d | i \in \mathcal{V}\}, \quad (3)$$

which can represent local chemical environments that are encoded in the structure of the molecular graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the set of nodes $\mathcal{V}$ represents the atoms and the set of edges $\mathcal{E}$ represents the bonds. In a second step (Figure 3), for each interaction type l MM parameters $\xi^{(l)}$ are predicted from the embeddings of the atoms involved in the respective energy contribution,

$$\xi^{(l)}_{ij\ldots} = \psi^{(l)}\left(\nu_i, \nu_j, \ldots\right), \tag{4}$$

using a transformer $\psi^{(l)}$ that is invariant under certain permutations. With the energy function of MM, the predicted parameter set $\boldsymbol{\xi}$ defines a potential energy surface, which can finally be evaluated for different spatial conformations $\mathbf{x}$ of the molecule,

$$E(\mathbf{x}) = E_{\text{MM}}\left(\mathbf{x}, \boldsymbol{\xi}\right). \tag{5}$$

Since the mapping from molecular graph to energy is differentiable with respect to the model parameters and spatial positions, it can be optimized on predicting QM energies and forces end-to-end. Notably, the machine learning model does not depend on the spatial conformation of the molecule, thus it has to be evaluated only once per molecule and the computational cost of each subsequent energy evaluation is given by the MM energy function.

Grappa currently only predicts bonded MM parameters since we expect that nonbonded interactions are not covered sufficiently by the monomeric datasets used for training, rendering the nonbonded parameters underdetermined. The nonbonded parameters are taken from established MM force fields that can reproduce solute interactions and melting points, which we expect to be strongly dependent on nonbonded interactions.

2.1. Permutation symmetries in MM

For the mapping from atom embeddings $\boldsymbol{\nu}$ to MM parameters, we postulate certain permutation symmetries that the model should respect. To derive these symmetries, we consider the energy function of MM as a decomposition into contributions from subgraphs of the featurized molecular graph that correspond to bonds, angles, torsions and improper dihedrals. We demand invariance of the energy contribution under node permutations that induce isomorphisms of the respective subgraph. For bonds, angles and torsions, these permutations leave the respective spatial coordinate invariant, thus we can simply achieve invariance of the energy contribution by demanding invariance of the MM parameters,

$$\xi^{(\text{bond})}_{ij} = \xi^{(\text{bond})}_{ji}, \tag{6}$$

$$\xi^{(\text{angle})}_{ijk} = \xi^{(\text{angle})}_{kji}, \tag{7}$$

$$\xi^{(\text{torsion})}_{ijkl} = \xi^{(\text{torsion})}_{lkji}. \tag{8}$$

For improper dihedrals, however, not all subgraph isomorphisms leave the dihedral angle invariant and demanding parameter invariance under those permutations would lead to an energy contribution that is not invariant. In Grappa, we solve this problem by decomposing the improper torsion contributions into three terms, as described in A.4.

2.2. The Grappa architecture

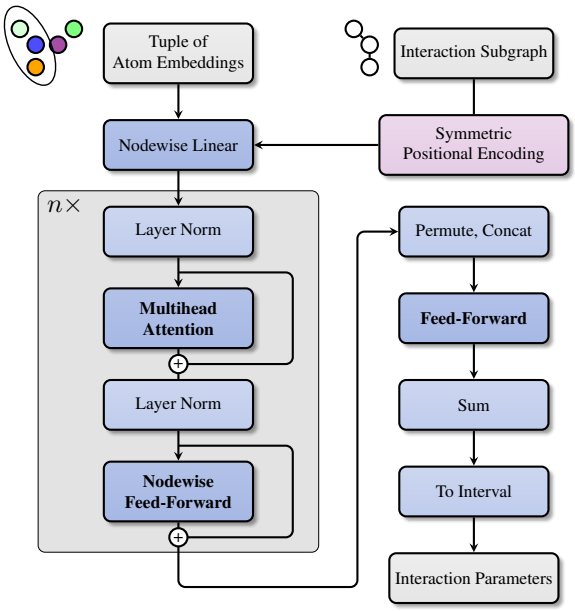


Figure 2: Architecture of the symmetric transformer: Atom embeddings are equipped with a permutation invariant positional encoding determined by the subgraph they represent. They are then passed through $n = 3$ permutation equivariant transformer layers, symmetry-pooled and mapped to the possible range of the respective parameter.

To predict atom embeddings from the molecular graph, Grappa employs a graph attentional neural network inspired by the transformer architecture (Vaswani et al., 2023). Multihead dot-product attention on graph edges (Veličković et al., 2018) is followed by nodewise feed-forward layers. We use residual connections (He et al., 2015) and layer normalization (Ba et al., 2016), which has been demonstrated to enhance the expressivity of the attention layer (Brody et al., 2023).

For the map from these embeddings to MM parameters, it is desirable to use an architecture that respects the permutation symmetries (Eqs. 6 - 8) by design, constraining the space of possible models to those that are physically sensible. In the spirit of equivariant machine learning, we use permutation equivariant layers followed by final symmetric pooling. However, since we do not require invariance under all permutations but only under permutations as defined in Eqs. 6 - 8, we can increase expressivity by allowing the model to

break symmetries that are not required.

Following these considerations, we use a transformer architecture based on multi-head dot-attention with a positional encoding that is invariant under the required symmetries but can break others. For example for angles, this positional encoding is given by

$$(\nu_i, \nu_j, \nu_k) \mapsto (\nu_i \oplus 0,\ \nu_j \oplus 1,\ \nu_k \oplus 0)\,, \tag{9}$$

where the $\oplus$ operation appends the respective value to the node feature vector, making it invariant under $ijk \to kji$ but not under e.g. $ijk \to jik$. After these equivariant layers, we apply a multilayer perceptron (MLP) on the concatenated permuted node embeddings and sum over the desired set of permutations $\mathcal{P}$,

$$z_{ij\ldots} = \sum_{\sigma \in \mathcal{P}} \mathrm{MLP}\left(\left[\nu_{\sigma(i)}, \nu_{\sigma(j)}, \ldots\right]\right), \tag{10}$$

defining a symmetry pooling operation with $\mathcal{P}$-invariant output. We call this combination of permutation invariant positional encoding with permutation equivariant layers and symmetric pooling the *symmetric transformer*, which is illustrated in Figure 2. The symmetric transformer can be generalized to permutation symmetries of arbitrary subgraphs by using the eigenvectors of the graph Laplacian (Yun et al., 2020) as positional encoding.

Finally, we map to the range of physically sensible parameters, e.g. $(0, \infty)$ for bond and angle force constants or $(0, \pi)$ for equilibrium angles $\theta^{(0)}$. To this end, we use scaled and shifted versions of ELU and the sigmoid function as described in A.2.3. While one could also use the exponential for mapping to $(0, \infty)$, ELU's linear behaviour towards large inputs is favorable for producing stable gradients during optimization. With the scaling we ensure that a normally distributed output of the neural network is mapped to a distribution with mean and standard deviation that is suitable for the respective MM parameter, which can be seen as a normalization technique (Ba et al., 2016). For predicting dihedral parameters ξ_{ijkl}, we use a sigmoid gate, which allows the model to supress dihedral modes that are not needed.

As input feature, we use one-hot encodings of the atomic number, the number of neighbors of the respective node and membership in loops of length 3 to 8, which can be directly calculated from the molecular graph. The assignment of nonbonded MM parameters is done using a traditional force field of choice. Since the bonded parameters predicted by Grappa may depend on the scheme by which nonbonded parameters are assigned, we pass the partial charge of each atom as input feature to Grappa, which also allows to encode the total charge of a molecule without breaking graph symmetries as one would potentially do by using the formal charge instead.

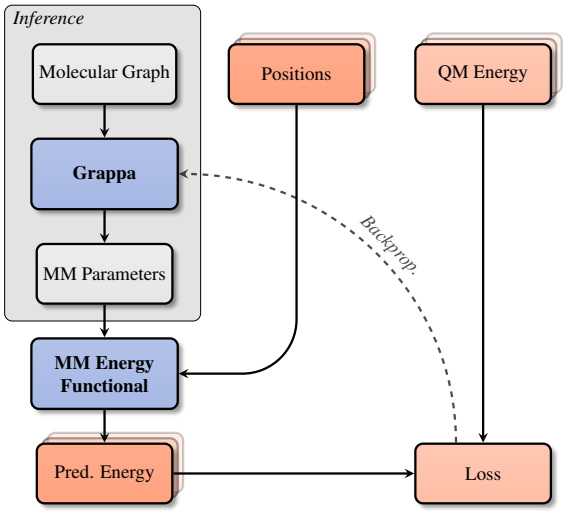


Figure 3: Grappa predicts one set of parameters per molecule. With the MM energy functional (Eq. **??**), the parameters can be mapped to energies and forces of given states, whose deviation from the ground truth is minimized during training.

2.3. Training

We train Grappa to minimize the mean squared error (MSE) between QM energies E_{QM} and forces $-\nabla_{\mathbf{x}} E_{\mathrm{QM}}$, whose contribution we weight by the hyperparameter λ_F, and its prediction, E and $\nabla_{\mathbf{x}} E$ respectively, which includes the non-learnable nonbonded contribution. At the start of training, we also include the deviation of predicted bond, angle and torsion MM parameters to those of a given traditional force field, weighted by λ_{trad} and, for regularization, the $L2$ norm of dihedral MM parameters $\xi^{(\mathrm{dih})}_{ijkl}$, weighted by λ_{dih} as in (Takaba et al., 2023). That is, we use the loss function

$$\begin{aligned}\mathcal{L} = &\ \mathrm{MSE}\left(E, E_{\mathrm{QM}}\right) + \lambda_F \mathrm{MSE}\left(\nabla_{\mathbf{x}} E, \nabla_{\mathbf{x}} E_{\mathrm{QM}}\right) \\ &+ \lambda_{\mathrm{MM}} \mathrm{MSE}\left(\xi, \xi_{\mathrm{trad}}\right) + \lambda_{\mathrm{dih}} \left\|\xi^{(\mathrm{dih})}\right\|_2^2 .\end{aligned} \tag{11}$$

Since MM can only predict energy differences of states, not formation energies, we subtract the mean of target and predicted energies for each molecule. As shown previously for other machine learned force fields (Schütt et al., 2017; 2021), we find training on forces $-\nabla_r E$ in addition to energies important also in our setting, for two reasons. First, the energy is a global, pooled quantity and thus less expressive than the forces, which are local and by a factor of $3N$ more numerous. Second, as shown in (Czarnecki et al., 2017), learning derivatives of the target with respect to the input can lead to improved generalization and data efficiency, effectively smoothing the learned potential energy surface.

The motivation for the different terms in the loss function, details on the training procedure and hyperparameters for all models mentioned in Sections 3 and 4 can be found in A.3.

3. Results

3.1. Grappa is state-of-the-art

To demonstrate that Grappa is the current state-of-the-art MM force field in terms of accuracy, we train and evaluate a Grappa model on the dataset reported in the follow-up paper (Takaba et al., 2023) to Espaloma, which contains 17,427 unique molecules and over one million conformations. Our training and test partition is identical with the one from Espaloma, where the molecules were divided into 80% training, 10% validation and 10% test set, based on isomeric SMILES strings. The dataset covers small molecules, peptides and RNA with states sampled from the Boltzmann distribution at 300 K and 500 K, from optimization trajectories and from torsion scans. The nonbonded contribution is calculated using the OpenFF-2.0.0 force field (Boothroyd et al., 2022) and partial charges from the AM1-BCC method, as in Espaloma. We train the model for 1,000 epochs on an A100 GPU, which takes about one day.

For all types of molecules, Grappa outperforms established MM force fields and Espaloma in terms of energy and force accuracy on Boltzmann-sampled states as shown in Table 1 and Figure 4a. While the Boltzmann samples are the more relevant benchmarking data for using Grappa in MD simulations, we also compare performance on torsion scans and optimization trajectories (Table A6). There, we find Grappa and Espaloma to be competitive, while Grappa is more accurate for forces and less accurate for energies. Grappa also outperforms the baselines if only trained on a fraction of Espaloma's training set (Figure A10), indicating high data efficiency. Chemical properties based on expert knowledge, e.g. hybridization and aromaticity, have long been used to assign MM parameters (He et al., 2020; Boothroyd et al., 2022). We show that accurate MM parameters can be predicted directly from the molecular graph, without relying on hand-crafted input features.

3.2. Grappa is extensible across chemical space

If a certain kind of chemistry is not part of the training set of a machine learned force field, predictions can go awry. While the current datasets already cover a significant part of chemical space, extensions should be straightforward and acessible. To demonstrate Grappa's extensibility, we train it on the Grappa-1.3 dataset, which includes dipeptides sampled at 300 and 1000 K, N- and C-terminal amino acids, dipeptides containing the non-standard residues hydroxyproline and DOPA (dihydroxyphenylalanine), and, demonstrating Grappa's capability to parametrize rather uncommon molecules, peptide radicals, which play a role in enzyme catalysis (Lebrette et al., 2023) and as mechanoradicals (Zapp et al., 2020).

To create these additional datasets, we calculate DFT energies and gradients of states that were sampled from MD simulations performed with traditional force fields at 300 K and 1000 K, where we increase the temperature to 1000 K in between the samplings of 300 K-states to enrich conformational diversity, as described in A.5.

In Table 2, we report Grappa's accuracy on the Grappa-1.3 dataset. Again, Grappa outperforms the baselines Amber99SB-ILDN and Gaff-2.11 on all peptide datasets, also for states sampled at 1000 K, indicating that Grappa is robust under conformational perturbations that lead to out-of-equilibrium states. For the radical peptides, there is no baseline since, to the best of our knowledge, Grappa is the first MM force field that covers this part of chemical space. On the other datasets from Espaloma, Grappa remains competitive, as can be seen in Table A7.

The extension to the Grappa-1.3 dataset indicates that Grappa can learn new chemistries accurately without negatively affecting the performance on the original datasets. Due to Grappa's limited complexity of input features – only connectivity, atomic numbers and partial charges – the chemical space accessible for parameterization is vast. As long as a molecule has a fixed connectivity and the MM energy functional is a reasonable approximation, Grappa can be trained to predict parameters for it.

3.3. Grappa is compatible with multiple nonbonded parameter schemes

Grappa-1.3 is trained to predict bonded parameters that are compatible with partial charges from AM1-BCC, ff99SB and CHARMM36. Table 3 shows that, in combination with all of these charge models, Grappa-1.3 can predict energies and forces of dipeptides to higher accuracy than established force fields (Table 2). Also for partial charges from ff99SB with additional Gaussian noise of scale 0.1 e, Grappa-1.3 is able to predict energies and forces accurately, indicating that the model is robust under small changes in the partial charges and that Grappa can thus be used in combination with charge models that are not present in the training data. This can be partly attributed to the fact that, on the datasets considered, nonbonded interactions contribute only a small fraction to the total interaction. For learning or evaluating the quality of nonbonded parameters by accuracy of energies and forces, we expect datasets that include intermolecular interactions to be more suitable.

3.4. Grappa can be combined with established force fields

Grappa's modular formulation allows for learning only a certain type of interaction while keeping the others fixed to those of an established force field. To quantify the improvement of Grappa over the widely used protein force field Amber ff99SB-ILDN (Lindorff-Larsen et al., 2010), we

Dataset	Test Mols	Confs		Grappa	Espaloma	Gaff-2.11	ff14SB, RNA.OL3	Mean Predictor
SPICE-Pubchem	1411	60853	*Energy*	**2.3**	**2.3**	4.6		18.4
			Force	**6.1**	6.8	14.6		23.4
SPICE-DES-Monomers	39	2032	*Energy*	**1.3**	1.4	2.5		8.2
			Force	**5.2**	5.9	11.1		21.3
SPICE-Dipeptide	67	2592	*Energy*	**2.3**	3.1	4.5	4.6	18.7
			Force	**5.4**	7.8	12.9	12.1	21.6
RNA-Diverse	6	357	*Energy*	**3.3**	4.2	6.5	6.0	5.4
			Force	**3.7**	4.4	16.7	19.4	17.1
RNA-Trinucleotide	64	35811	*Energy*	**3.5**	3.8	5.9	6.1	5.3
			Force	**3.6**	4.3	17.1	19.7	17.7

Table 1: Accuracy of Grappa, Espaloma 0.3 (Takaba et al., 2023) and established MM force fields on test molecules of the Espaloma dataset. We report the RMSE of centered energies in kcal/mol and the componentwise RMSE of forces in kcal/mol/Å, uncertainties can be found in Table A6. Gaff-2.11 (He et al., 2020) is a general-purpose force field, ff14SB (Maier et al., 2015) is an established protein force field and RNA.OL3 (Zgarbová et al., 2011) is specialized to RNA. The SPICE-Pubchem and SPICE-DES-Monomers datasets, containing small molecules, along with the dipeptide dataset, are subsets of the SPICE dataset (Eastman et al., 2023). The RNA datasets feature trinucleotide states calculated with the B3LYP-D3BJ functional.

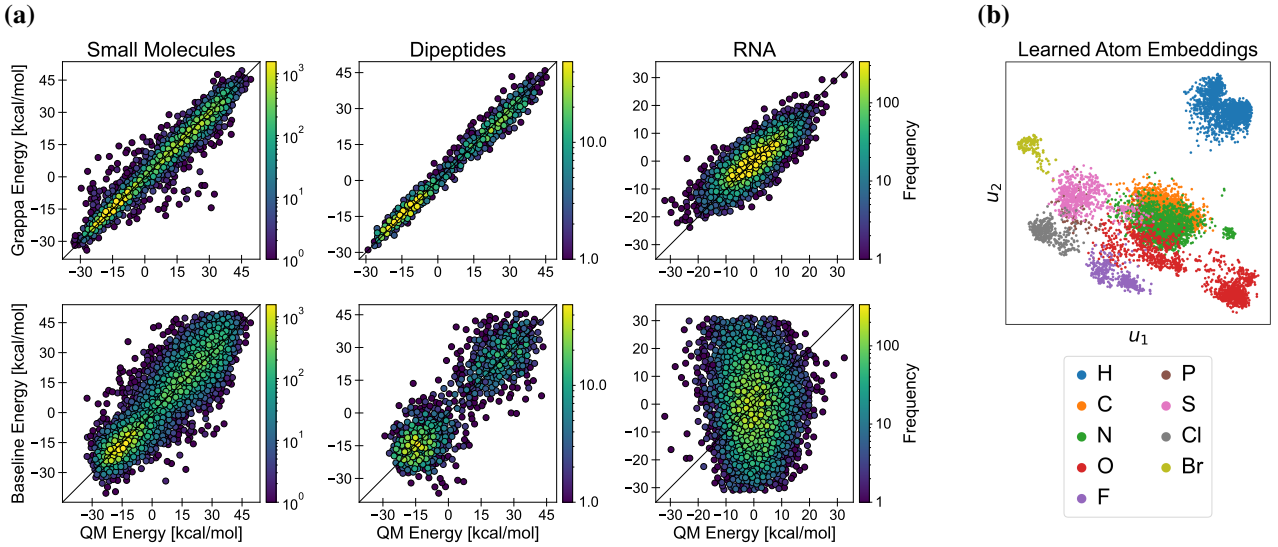


Figure 4: **(a)** Comparison of energy predictions of Grappa-1.3 and the established force fields Gaff-2.11, ff99SB-ILDN (Lindorff-Larsen et al., 2010) and RNA.OL3 for test molecules from Espaloma's SPICE-Pubchem, SPICE-Dipeptide and RNA-Trinucleotide datasets; force predictions are depicted at A12. **(b)** The two first principal components u_1 and u_2 of predicted atom embeddings from the Espaloma test dataset can be related to a combination of the main group and period in the periodic table of elements. Lines of constant main group or period are represented by approximate diagonals in latent space.

train Grappa models that predict only certain types of MM parameters and evaluate their accuracy in Table 4. While for each contribution, replacing the ff99DB-ILDN parameters with those predicted by Grappa improves the accuracy, the bond and angle contributions have the most potential for improvement.

3.5. Grappa is interpretable

MM parameters In the classical mechanics potential functions used in MM, a physical meaning can be attributed to parameters. Hence, badly assigned parameters could be noticeable, e.g. if a bond is much shorter or stiffer than bonds between similar atoms or the amplitude of a proper

Dataset	Test Mols	Confs		Grappa	Espaloma	Gaff-2.11	ff99SB-ILDN	Mean Predictor
SPICE-Dipeptide	67	2592	*Energy*	**2.4**	3.1	4.5		19.1
			Force	**5.4**	7.8	12.9		38.4
Dipeptides-300K	72	3600	*Energy*	**2.6**		4.5	4.1	7.8
			Force	**5.9**		10.9	11.8	43.7
Dipeptides-1000K	72	2160	*Energy*	**5.4**		8.6	8.5	18.9
			Force	**11.6**		16.1	17.8	74.3
Non-Capped-Peptides	10	500	*Energy*	**2.2**		4.2	4.0	7.2
			Force	**6.1**		12.1	12.6	45.8
Radical-Dipetides	28	272	*Energy*	**3.3**				8.7
			Force	**6.8**				41.3

Table 2: Accuracy of Grappa-1.3 with nonbonded parameters from ff99SB, Espaloma 0.3 (Takaba et al., 2023) and established MM force fields on a subset of test molecules from the Grappa-1.3 dataset. The performance on small molecules and RNA is similar to the one reported in Table 1 and can be found in Table A7. As in Table 1, we report the RMSE of centered energies in kcal/mol and the componentwise RMSE of forces in kcal/mol/Å.

Charge Model	Energy RMSE	Force RMSE
ff99SB	2.55 ± 0.05	5.85 ± 0.06
CHARMM36	2.58 ± 0.07	6.01 ± 0.07
AM1-BCC	2.79 ± 0.07	6.13 ± 0.07

Table 3: Test accuracy of Grappa-1.3 on the Dipeptides-300K dataset in kcal/mol(/Å) with sets of nonbonded parameters obtained from ff99SB, CHARMM36 and AM1-BCC charges with Lennard-Jones parameters from the OpenFF-2.2.0 force field.

Learnable contribution			RMSE [kcal/mol(/Å)]	
Bond	Angle	Dihedral	Energy	Force
✓	✓	✓	2.5	5.9
✓	✓		3.0	6.1
✓			3.9	7.4
	✓		3.8	9.2
		✓	3.5	11.0
			4.1	11.8

Table 4: Test accuracy of Grappa models that predict MM parameters for different interaction types. For the non-learnable types, MM parameters are taken from ff99SB-ILDN. The models are trained on the Dipeptides-300K and -1000K datasets and evaluated on Dipeptides-300K with ff99SB partial charges.

dihedral is untypically high. Apart from exposing the parameters in the simulation files, Grappa also provides figures of the parameter distributions and, if available, comparisons to parameters from an established force field upon parametrization. For the case of the protein ubiquitin considered in section A.6.1, parameter distributions and a comparison to ff99-SBILDN are depicted in Figure A11. While the established protein force field and Grappa-1.3 predict similar bond distances, we observe deviations for other types of parameters.

Latent space To generalize across the broad range of chemical space covered by the training set, Grappa has to learn highly informative atom embeddings that capture the local environment in the molecular graph. For interpreting this latent representation, we apply a two-dimensional principal component analysis on a set of predicted atom embeddings (3) of test molecules (Figure 4b). It turns out that the two principal components can be related to main group and period of the periodic table of elements, whose structure is learned implicitly by Grappa.

4. Force field validation

Especially for large biomolecules, force fields are not only required to predict energies and forces as accurately as possible, but their predictions should behave in such a way that certain macroscopic properties such as stability and consistency with experimentally determined folding states are fulfilled in MD simulations. Grappa is well suited to overcome the gap between empirically validated, established protein force fields and machine learned force fields.

In this section, we demonstrate that Grappa is on par with established force fields when it comes to MD simulations of proteins over timescales of hundreds of nanoseconds and that it captures the physics that cause stability of protein folds. All simulations mentioned in this section were per-

formed with GROMACS, for which the Grappa package[1] provides a command line interface, briefly described in A.1. Using this interface (or a similar one for OpenMM), large biomolecular systems with over 100,000 atoms are parameterized within minutes on a CPU (Figure A6).

4.1. Grappa keeps large proteins stable during MD

A key quality check for a protein force field is if a native protein structure remains stable over time during an MD simulation. QM methods and E(3) equivariant neural networks suffer from the problem that while their end-to-end predictions are accurate, they might fail in yielding stable simulations because of diverging gradients of the potential energy surface when extrapolating beyond regions covered by the training dataset. Particularly over long timescales, the system might drift out of the range of validity of the model and small errors might accumulate, leading to instabilities of the system and a potential crash of the simulation. MM with its interpretable, physics-inspired energy function, with parameters which are conformation-independent, ensures a higher degree of stability, and we indeed did not observe simulation crashes.

However, since it is not ensured a priori that macrostates such as protein folds remain stable during MD, we next assess Grappa's capability of keeping protein folds close to their experimentally determined structure. As example system, we use ubiquitin (PDB ID: 1UBQ), which is a protein with 76 amino acids whose fold contains a beta-sheet and an alpha-helix. We perform a 200 nanoseconds MD simulation of ubiquitin in aqueous solution with Grappa and Amber ff99SB-ILDN (Lindorff-Larsen et al., 2010), for which details are given in A.6.1. We find that the C-alpha RMSD from the initial state is bounded by about 4 Å in 200 ns simulation with both Grappa and Amber ff99SB-ILDN (Figure 5c), indicating that the folded state of the protein is stable. Structural fluctuations on timescales of up to 2 ns are of similar magnitude (Figure 5b).

4.2. Grappa is orders of magnitude more resource-efficient than E3 equivariant models

Efficiency of force fields is crucial; otherwise the computational cost for simulating large systems on long timescales can prohibit their application in simulations of many systems of interest. To demonstrate Grappa's efficiency, which it inherits from molecular mechanics, and to showcase its capability of jointly parametrizing RNA and proteins, we simulate the virus STMV (The Amber Project, 2023) in solution – a system with approximately one million atoms, visualized in Fig. A9 (a).

A recently proposed machine learned force field relying on an E(3) equivariant neural network with state-of-the-art accuracy and efficiency, Allegro (Musaelian et al., 2022), has been shown to be capable of performing near-quantum-accuracy simulations of systems of unprecedented size. This is achieved by using a strictly local architecture, which allows parallelization across thousands of GPUs. Since they are not constrained by the energy functional of MM, E3 equivariant models like Allegro, NequIP (Batzner et al., 2022) or MACE (Batatia et al., 2023) have greatly improved accuracy but are orders of magnitude more expensive than MM force fields like Grappa.

For the system at hand, Allegro achieves a performance of 106 timesteps per second on 4,000 A100 GPUs (Kozinsky et al., 2023). For Grappa, at reduced accuracy and incapable of describing topological changes directly, we measure a performance of 101 timesteps per second on a single A100 GPU in GROMACS (Abraham et al., 2015) without system-specific optimizations.

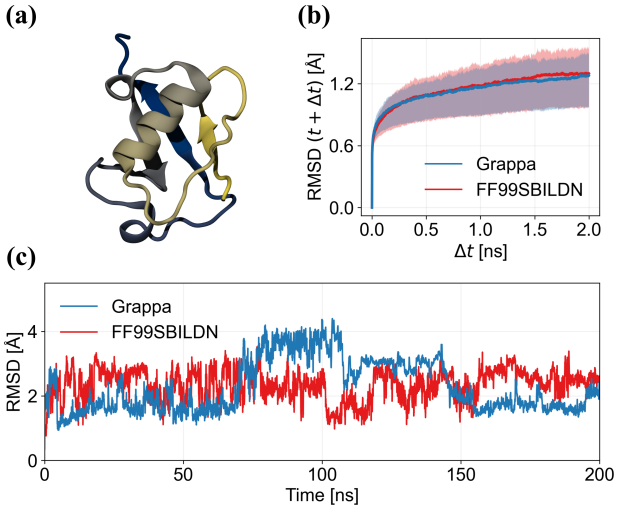


Figure 5: **(a)** The protein ubiquitin with color-coded sequence position. **(c)** The mean C-alpha root mean square deviation (RMSD) and its 25th and 75th percentile of 1000 random pairs of frames that are separated by the time difference Δt. **(b)** C-alpha RMSD from the initial state during MD simulation of ubiquitin in water with Grappa.

5. Conclusions

With Grappa, we propose a machine learning framework for molecular mechanics force fields with state-of-the-art accuracy that can be used seamlessly in established MD engines. Our experiments show that Grappa is transferable to large biomolecules like proteins and viruses, which sets the stage for large scale simulations at improved accuracy, with the same computational efficiency as established MM force fields.

[1] https://github.com/graeter-group/grappa

Unlike most traditional MM force fields, Grappa is not specialized to a certain chemical species, but offers consistent parameterization of molecules with different chemistries, such as a wide range of small molecules, peptides and nucleotides, with a single model at a higher level of accuracy than previous MM force fields. Grappa not only enhances existing parameter sets, but also reaches previously inaccesible regions of chemical space, such as non proteinogenic amino acids or protein radicals.

Since current state-of-the-art E(3) equivariant neural network potentials like Allegro are several orders of magnitude more expensive than molecular mechanics, we regard machine learned molecular mechanics approaches like Grappa as suitable method for simulating large systems, especially if computational ressources are limited or when it is not necessary to simulate all parts of a system at quantum accuracy.

6. Outlook

While Grappa achieves high computational efficiency through molecular mechanics, it also inherits its fundamental limitations, including restricted accuracy and the inability to directly describe chemical reactions. However, Grappa is well suited for efficient reparametrization of large molecules that undergo local topological changes induced by chemical reactions, e.g. in kinetic Monte Carlo simulations (Rennekamp et al., 2020). This is due to both, Grappa's finite field-of-view, which ensures locality on the molecular graph, as well as the lack of hand-crafted chemical features, which allows to apply the parametrization to cut-out regions of molecules.

Grappa is a general and versatile framework to obtain MM parameters for a broad range of molecules and can be seamlessly extended to new chemistries. To this end, the Grappa package implements workflows for retraining on the provided and custom new datasets.

While we consider the consistency with established nonbonded parameters an advantage of Grappa, an extension of the framework to nonbonded parameters is a straightforward next step. It would render Grappa fully independent from traditional MM force fields and could potentially improve its accuracy further.

7. Reproducibility statement

Grappa is released as open source software under the GNU General Public License v3.0 [2] along with the Grappa dataset and SMILES strings used for the train-val-test partition and configuration files to reproduce Table 1 and Table 2. Repositories for reproducing the MD simulations [3] and creating the Grappa-1.3 datasets [4] are publicly available.

Author Contributions

Leif Seute: Methodology, Conceptualization, Software, Writing - Original Draft, Investigation, Visualization, Data Curation
Eric Hartmann: Conceptualization, Software, Verification, Visualization, Writing - Review and editing
Jan Stühmer: Formal analysis, Writing - Review and editing
Frauke Gräter: Supervision, Conceptualization, Writing - Review and editing

Acknowledgements

This research was supported by the Klaus Tschira Foundation. The project has also received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 101002812) [F.G.]. We thank Kai Riedmiller and Jannik Buhr for application and software related discussions, Fabian Grünewald, Camilo Aponte-Santamaria and Vsevolod Viliuga for discussions on future applications and Daniel Sucerquia for conversations on the role of internal coordinates in MM.

References

Abraham, M. J., Murtola, T., Schulz, R., Páll, S., Smith, J. C., Hess, B., and Lindahl, E. Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, 1-2: 19–25, 2015. ISSN 2352-7110.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization, 2016.

Batatia, I., Kovács, D. P., Simm, G. N. C., Ortner, C., and Csányi, G. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields, 2023.

Batzner, S., Musaelian, A., Sun, L., Geiger, M., Mailoa, J. P., Kornbluth, M., Molinari, N., Smidt, T. E., and Kozinsky, B. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*, 13(1), may 2022. doi: 10.1038/

[2] https://github.com/graeter-group/grappa.

[3] https://github.com/LeifSeute/validate-grappa

[4] https://github.com/LeifSeute/grappa-data-creation

s41467-022-29939-5. URL https://doi.org/10.1038%2Fs41467-022-29939-5.

Blondel, A. and Karplus, M. New formulation for derivatives of torsion angles and improper torsion angles in molecular mechanics: Elimination of singularities. *Journal of Computational Chemistry*, 17(9):1132–1141, 1996.

Boothroyd, S., Behara, P., Madin, O., Hahn, D., Jang, H., Gapsys, V., et al. Development and benchmarking of open force field 2.0.0 — the sage small molecule force field. *ChemRxiv*, 2022. doi: 10.1021/acs.jctc.3c00039. URL https://doi.org/10.1021/acs.jctc.3c00039. This content is a preprint and has not been peer-reviewed.

Brody, S., Alon, U., and Yahav, E. On the expressivity role of layernorm in transformers' attention. *arXiv preprint arXiv:2305.02582*, 2023.

Bussi, G., Donadio, D., and Parrinello, M. Canonical sampling through velocity rescaling. 126(1):014101. ISSN 0021-9606, 1089-7690. doi: 10.1063/1.2408420.

Clevert, D.-A., Unterthiner, T., and Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus), 2016.

Czarnecki, W. M., Osindero, S., Jaderberg, M., Świrszcz, G., and Pascanu, R. Sobolev training for neural networks, 2017.

Eastman, P., Friedrichs, M. S., Chodera, J. D., Radmer, R. J., Bruns, C. M., Ku, J. P., Beauchamp, K. A., Lane, T. J., Wang, L.-P., Shukla, D., Tye, T., Houston, M., Stich, T., Klein, C., Shirts, M. R., and Pande, V. S. Openmm 4: A reusable, extensible, hardware independent library for high performance molecular simulation. *Journal of Chemical Theory and Computation*, 9(1):461–469, 2013. doi: 10.1021/ct300857j. URL https://doi.org/10.1021/ct300857j. PMID: 23316124.

Eastman, P., Behara, P., Dotson, D., and et al. Spice, a dataset of drug-like molecules and peptides for training machine learning potentials. *Scientific Data*, 10 (11), 2023. URL https://doi.org/10.1038/s41597-022-01882-6.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015.

He, X., Man, V., Yang, W., Lee, T.-S., and Wang, J. A fast and high-quality charge model for the next generation general amber force field. *The Journal of chemical physics*, 153:114502, 09 2020. doi: 10.1063/5.0019056. URL https://doi.org/10.1063/5.0019056.

Hess, B. P-LINCS: A parallel linear constraint solver for molecular simulation. 4(1):116–122. ISSN 1549-9618, 1549-9626. doi: 10.1021/ct700200b. URL https://pubs.acs.org/doi/10.1021/ct700200b.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization, 2017.

Kozinsky, B., Musaelian, A., Johansson, A., and Batzner, S. Scaling the leading accuracy of deep equivariant models to biomolecular simulations of realistic size. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701092. doi: 10.1145/3581784.3627041. URL https://doi.org/10.1145/3581784.3627041.

Lebrette, H., Srinivas, V., John, J., Aurelius, O., Kumar, R., Lundin, D., Brewster, A. S., Bhowmick, A., Sirohiwal, A., Kim, I.-S., Gul, S., Pham, C., Sutherlin, K. D., Simon, P., Butryn, A., Aller, P., Orville, A. M., Fuller, F. D., Alonso-Mori, R., Batyuk, A., Sauter, N. K., Yachandra, V. K., Yano, J., Kaila, V. R. I., Sjöberg, B.-M., Kern, J., Roos, K., and Högbom, M. Structure of a ribonucleotide reductase r2 protein radical. *Science*, 382(6666):109–113, 2023. doi: 10.1126/science.adh8160. URL https://www.science.org/doi/abs/10.1126/science.adh8160.

Lindorff-Larsen, K., Piana, S., Palmo, K., Maragakis, P., Klepeis, J., Dror, R., and Shaw, D. Improved side-chain torsion potentials for the amber ff99sb protein force field. *Proteins*, 78(8):1950–1958, Jun 2010. doi: 10.1002/prot.22711. URL https://doi.org/10.1002/prot.22711.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts, 2017.

Maier, J. A., Martinez, C., Kasavajhala, K., Wickstrom, L., Hauser, K. E., and Simmerling, C. ff14sb: Improving the accuracy of protein side chain and backbone parameters from ff99sb. *Journal of Chemical Theory and Computation*, 11(8):3696–3713, 2015. ISSN 1549-9618. doi: 10.1021/acs.jctc.5b00255. URL https://doi.org/10.1021/acs.jctc.5b00255.

Musaelian, A., Batzner, S., Johansson, A., Sun, L., Owen, C. J., Kornbluth, M., and Kozinsky, B. Learning local equivariant representations for large-scale atomistic dynamics, 2022. URL https://arxiv.org/abs/2204.05249.

Parrinello, M. and Rahman, A. Polymorphic transitions in single crystals: A new molecular dynamics method. 52(12):7182–7190. ISSN 0021-8979. doi: 10.1063/

1.328693. URL https://doi.org/10.1063/1.328693.

Paszke, A., Gross, S., Massa, F., Lerer, A., and et al. Pytorch: An imperative style, high-performance deep learning library, 2019.

Rennekamp, B., Kutzki, F., Obarska-Kosinska, A., Zapp, C., and Gräter, F. Hybrid kinetic monte carlo/molecular dynamics simulations of bond scissions in proteins. *Journal of Chemical Theory and Computation*, 16(1):553–563, 2020. doi: 10.1021/acs.jctc.9b00786. URL https://doi.org/10.1021/acs.jctc.9b00786. PMID: 31738552.

Schütt, K. T., Kindermans, P.-J., Sauceda, H. E., Chmiela, S., Tkatchenko, A., and Müller, K.-R. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions, 2017. URL https://arxiv.org/abs/1706.08566.

Schütt, K. T., Unke, O. T., and Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra, 2021. URL https://arxiv.org/abs/2102.03150.

Smith, D. G. A., Burns, L. A., and Simmonett, e. a. Psi4 1.4: Open-source software for high-throughput quantum chemistry. *J. Chem. Phys.*, 2020. doi: 10.1063/5.0006002.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 06 2014.

Takaba, K., Pulido, I., Henry, M., MacDermott-Opeskin, H., Chodera, J. D., and Wang, Y. Espaloma-0.3.0: Machine-learned molecular mechanics force field for the simulation of protein-ligand systems and beyond, 2023.

The Amber Project. Amber23 benchmark. The Amber Project, 2023. URL https://ambermd.org/GPUPerformance.php. Access date: 2024-04-22. URL subject to change.

Unke, O. T., Stöhr, M., Ganscha, S., Unterthiner, T., Maennel, H., Kashubin, S., Ahlin, D., Gastegger, M., Sandonas, L. M., Tkatchenko, A., and Müller, K.-R. Accurate machine learned quantum-mechanical force fields for biomolecular simulations, 2022.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks, 2018. URL https://arxiv.org/abs/1710.10903.

Wang, M., Zheng, D., Ye, Z., Gan, Q., Li, M., Song, X., Zhou, J., Ma, C., Yu, L., Gai, Y., Xiao, T., He, T., Karypis, G., Li, J., and Zhang, Z. Deep graph library: A graph-centric, highly-performant package for graph neural networks, 2020.

Wang, Y., Fass, J., Kaminow, B., Herr, J. E., Rufa, D., Zhang, I., Pulido, I., Henry, M., Macdonald, H. E. B., Takaba, K., and Chodera, J. D. End-to-end differentiable construction of molecular mechanics force fields. *Chemical Science*, 13(41):12016–12033, 2022. doi: 10.1039/d2sc02739a. URL https://doi.org/10.1039%2Fd2sc02739a.

Yun, S., Jeong, M., Kim, R., Kang, J., and Kim, H. J. Graph transformer networks, 2020.

Zapp, C., Obarska-Kosinska, A., Rennekamp, B., Kurth, M., Hudson, D. M., Mercadante, D., Barayeu, U., Dick, T. P., Denysenkov, V., Prisner, T., Bennati, M., Daday, C., Kappl, R., and Gräter, F. Mechanoradicals in tensed tendon collagen as a source of oxidative stress. *Nature Communications*, 11(1):2315, May 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-15567-4. URL https://doi.org/10.1038/s41467-020-15567-4.

Zgarbová, M., Otyepka, M., Šponer, J., Mládek, A., Banáš, P., III, T. E. C., and Jurečka, P. Refinement of the cornell et al. nucleic acids force field based on reference quantum chemical calculations of glycosidic torsion profiles. *Journal of Chemical Theory and Computation*, 7(9):2886–2902, 2011. ISSN 1549-9618. doi: 10.1021/ct200162x. URL https://doi.org/10.1021/ct200162x.

A. Appendix

A.1. The Grappa package

Grappa utilizes PyTorch (Paszke et al., 2019) and DGL (Wang et al., 2020) in its implementation.

The package is released as open source software under the GNU General Public License v3.0 and is available at `https://github.com/graeter-group/grappa` and on pypi,

```
pip install grappa-ff
```

It brings a command line interface to reparametrize a GROMACS topology file obtained from a traditional force field:

```
grappa_gmx -f topol.top -o new_topol.top
```

For OpenMM, we provide a class that can be used to reparametrize a system with Grappa:

```
from grappa import OpenmmGrappa

# use a traditional OpenMM forcefield
# to obtain a system from your topology
top, system = ...

# download a pretrained grappa model
ff = OpenmmGrappa.from_tag('grappa-1.3')

# re-parametrize the system using grappa
system = ff.parametrize_system(system, top)

# continue with usual workflow
```

Pretrained released models can be accessed by using a tag; the model weights are downloaded automatically from the respective release on GitHub. Released models also contain a dictionary with results on test datasets, a list of identifiers for molecules used for training and validation and a configuration file to reproduce training on the same dataset. Further details on the package can be found on GitHub.

A.2. Model architecture

In this section, we provide a detailed description of the architecture of Grappa, including the graph attentional neural network and the symmetric transformer.

A.2.1. Graph attentional neural network

Grappa's graph attentional neural network (Figure A7) is a modification of the transformer architecture (Vaswani et al., 2023) for graphs, similar to the Graph Transformer Network (Yun et al., 2020), but e.g. omits the locality-breaking graph Laplacian eigenvectors as positional encoding. Instead, we encode the graph structure by constraining the attention mechanism to edges of the graph as in GAT (Veličković et al., 2018), which enforces locality: In each attention update only neighboring nodes can influence each other. We use a multi-head dot-product attention mechanism and a 2-layer MLP as feed-forward network, where the hidden feature dimension is four times the node feature dimension.

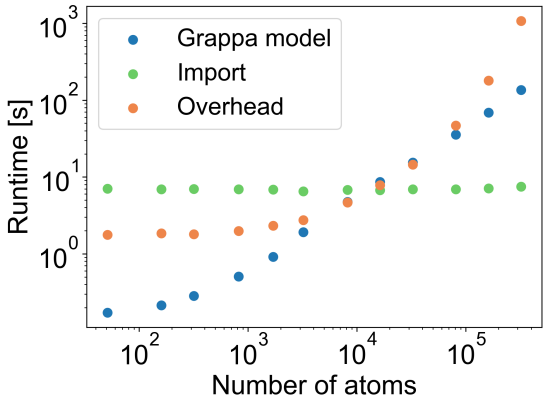


Figure A6: For molecules with up to 300,000 atoms, the runtime of a parameterization with the Grappa package in CPU mode is largely due to overhead. Due to its finite field-of-view, parametrizations can also be parallelised across several nodes by splitting the molecular graph into subgraphs if necessary.

After initializing the node features ν_i as described below, we apply a single nodewise linear layer followed by an exponential linear unit (ELU) (Clevert et al., 2016),

$$\nu_i \leftarrow \mathrm{ELU}\left(W\nu_i + b\right),\ W \in \mathbb{R}^{d\times d_{\text{in}}},\ b \in \mathbb{R}^d, \tag{12}$$

Then, we apply L_{GNN} graph attentional layers, each of which is given by

$$\nu_i \leftarrow \mathrm{LayerNorm}(\nu_i), \tag{13}$$

$$e_{ii'}^{(k)} = \left(W^{(k)}\nu_i\right)\cdot\left(W^{(k)}\nu_{i'}\right), \tag{14}$$

$$\mathrm{att}_{ii'}^{(k)} = \frac{\exp\left(e_{ii'}^{(k)}\right)}{\sum_{n\in\mathcal{N}(i)}\exp\left(e_{in}^{(k)}\right)}, \tag{15}$$

$$V_i^{(k)} = \sum_{i'\in\mathcal{N}(i)} \mathrm{att}_{ii'}^{(k)} W^{(k)}\nu_{i'}, \tag{16}$$

$$V_i \leftarrow \left(W_O \mathrm{Concat}\left[V_i^{(1)},\dots,V_i^{(N_{\text{heads}})}\right]\right), \tag{17}$$

$$\nu_i \leftarrow \mathrm{LayerNorm}(V_i) + \nu_i, \tag{18}$$

$$\nu_i \leftarrow \mathrm{ELU}\left(W_2\,\mathrm{ELU}\left(W_1\nu_i + b_1\right) + b_2\right) + \nu_i, \tag{19}$$

where $\mathcal{N}(i)$ is the set of neighbors of node i including the node itself, (k) denotes the attention head and where the

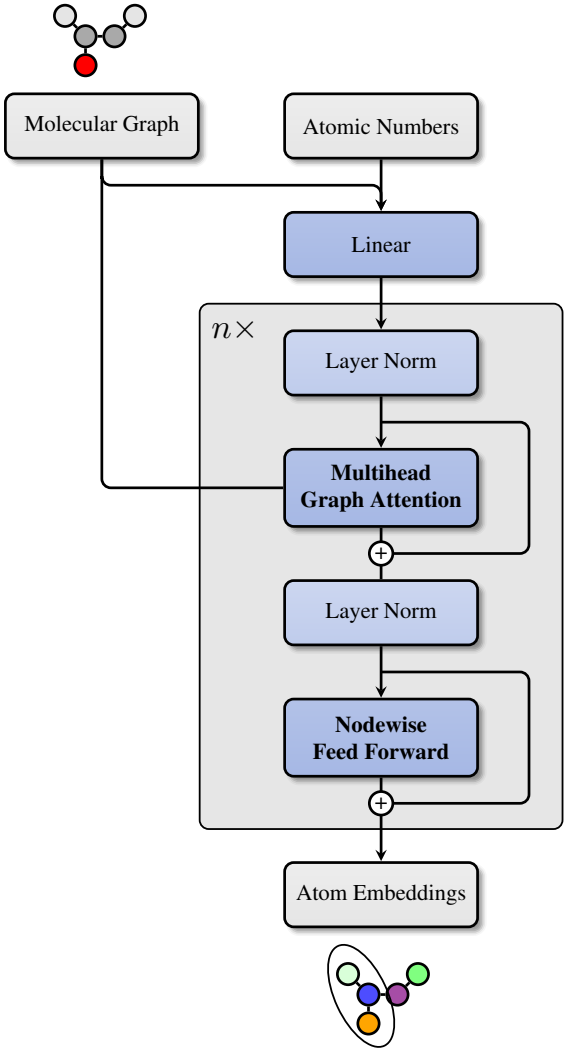


Figure A7: Grappa's graph attentional neural network as described in Section A.2.1.

weights

$$W^{(k)} \in \mathbb{R}^{d\times d/N_{\text{heads}}}, \tag{20}$$

$$W_O \in \mathbb{R}^{d\times d}, \tag{21}$$

$$W_1 \in \mathbb{R}^{4d\times d}, \tag{22}$$

$$W_2 \in \mathbb{R}^{d\times 4d}. \tag{23}$$

and biases b_1, b_2 are learnable and independent for each layer. We use dropout (Srivastava et al., 2014) on the node feature update in (18) and (19). Finally, we project onto the output dimension d_{emb} by a linear layer,

$$\nu_i \leftarrow W\nu_i, W \in \mathbb{R}^{d_{\text{emb}}\times d}. \tag{24}$$

As initial node features we choose a one-hot encoding of the atomic number, a one-hot encoding of whether the node is in a loop, a one-hot encoding of the potential loop size and the degree of the node. Since Grappa only predicts bonded parameters, we have to ensure consistency with the nonbonded parameters from the traditional force field, which is why we one-hot encode the traditional force field used for the nonbonded contribution of the respective state if we train on a dataset that contains data from different nonbonded methods. We also include the partial charges (the scalar value concatenated with a 16-dimensional binning between -2e and 2e) as node input features, which allows us to describe differently charged conformations of the same molecule without resorting to global or graph-symmetry breaking features like the formal charge.

A.2.2. Symmetric Transformer

The graph attentional neural network is followed by four (parallel) symmetric transformers (Figure 2), one for each type of interaction, that is one for bond, one for angle, one for torsions and one for improper dihedral parameters. For each interaction (e.g. for each angle in the molecular graph), the node embeddings of the atoms involved is mapped to a set of respective MM parameters (e.g. equilibrium angle and force constant).

As described in Section 2.2, we add a symmetric positional encoding to the node features, as in (9), that is invariant under the desired set of permutations but can break symmetries that are not necessary to make the model more expressive while keeping equivariance under the permutations that we use for invariant pooling (10) later on. From the symmetries Eqs. 6-8, and from the considerations for improper dihedrals in Section A.4, we can derive the following positional encodings

$$\text{PE}_{\text{angle}} = (0, 1, 0), \tag{25}$$

$$\text{PE}_{\text{torsion}} = (0, 1, 1, 0), \tag{26}$$

$$\text{PE}_{\text{improper}} = (0, 1, 1, 0). \tag{27}$$

The bond positional encoding is not needed because the permutations of the bond parameters are the full group S_2 and there is no symmetry that needs to be broken. For torsions and impropers, the positional encoding does not break all unnecessary symmetries, for example the symmetry $ijkl \rightarrow ljki$ is still present with positional encoding, and will be broken later.

We then apply a transformer acting one nodes that represent the atoms involved in the interaction, that is, after a projection on features of dimension d_T,

$$\nu_i \leftarrow \text{ELU}\left(W\nu_i + b\right),\ W \in \mathbb{R}^{d_T\times d_{\text{emb}}},\ b \in \mathbb{R}^{d_T}, \tag{28}$$

we apply L_{T} transformer layers, each of which is given by scaled dot-product attention with $N_{\text{T-heads}}$ heads and a 2-layer MLP with hidden dimension $4d_T$, skip connections,

dropout and layer normalization as in the original transformer architecture (Vaswani et al., 2023).

We then apply a symmetry pooling operation by passing permuted versions of concatenated node embeddings to a L_{pool}-layer MLP with hidden dimension d_{pool} and summing over all permutations σ in the respective symmetry group $\mathcal{P}$,

$$z_{ij\ldots} = \sum_{\sigma \in \mathcal{P}} \text{MLP}\left(\left[\nu_{\sigma(i)}, \nu_{\sigma(j)}, \ldots\right]\right), \tag{29}$$

The two dimensional scores z_{ij} for bonds and z_{ijk} for angles are finally mapped to the range of the respective set of MM parameters by scaled and shifted versions of ELU and the sigmoid function,

$$k_{ij} = \text{ToPos}\left(z_{ij,\,0}\right), \tag{30}$$

$$r_{ij}^{(0)} = \text{ToPos}\left(z_{ij,\,1}\right), \tag{31}$$

$$k_{ijk} = \text{ToPos}\left(z_{ijk,\,0}\right), \tag{32}$$

$$\theta_{ijk}^{(0)} = \text{ToRange}_{(0,\pi)}\left(z_{ijk,\,0}\right), \tag{33}$$

where ToPos and $\text{ToRange}_{(0,\pi)}$ are defined in Section A.2.3 and $z_{\ldots,\,n}$ denotes the n-th entry of the score vector $z_{\ldots}$. For dihedral force constants, the $2n_{\text{periodicity}}$ dimensional scores z_{ijkl} are fed through a sigmoid gate to make the model more expressive for small force constants,

$$k_{ijkl,\,m} = z_{ijkl,\,2m+1} \times \text{sigmoid}\left(z_{ijkl,\,2m}\right). \tag{34}$$

A.2.3. Scaling of neural network outputs

To map the scores predicted from the model to the range of the respective MM parameter, we use scaled and shifted versions of the sigmoid function for mapping to a specified interval and of ELU for mapping to positive values. We choose these functions because they are differentiable and, in contrast to the exponential, only grow linearly for large inputs, which can help to stabilize training. To allow the model to predict scores that are of order unity, or approximately normally distributed, we choose the scaling and shifting parameters in such a way that a normally distributed input would lead to an output distribution whose mean and standard deviation are close to a given target mean μ and standard deviation σ. We choose this target standard deviation by calculating mean and standard deviation of traditional MM force field parameters on a given subset of the training data, except for equilibrium angles, where we restrict μ to $\pi/2$.

For mapping to positive values, we use

$$\text{ToPos}[\mu, \sigma]\left(z\right) = \sigma\left(\text{ELU}\left(\frac{\mu}{\sigma} + x - 1\right) + 1\right), \tag{35}$$

and for mapping to a specified interval $(0, \gamma)$, we restrict ourselves to the case $\mu = \gamma/2$ use

$$\text{ToRange}[\gamma/2, \sigma]_{(0,\gamma)}\left(z\right) = \gamma\ \text{sigmoid}\left(\frac{4\sigma}{\gamma} x\right). \tag{36}$$

We can see that the mean and standard deviation of the output are indeed close to the target mean and standard deviation if we consider the asymptotic behavior of the sigmoid function and ELU as their input approaches zero,

$$\text{ELU}(x) \sim x + \mathcal{O}\left(x^2\right) \quad \text{as} \quad x \to 0, \tag{37}$$

$$\text{sigmoid}(x) \sim \frac{1}{2} + \frac{x}{4} + \mathcal{O}\left(x^2\right) \quad \text{as} \quad x \to 0, \tag{38}$$

which follows from the Taylor expansion of the ELU and sigmoid functions around zero. Thus, using that z has vanishing mean and unit standard deviation, we have indeed, to first order in z,

$$< \text{ToRange}\left(z\right) > \approx \gamma\left(1/2 + 4\sigma/\gamma < z >\right) = \gamma/2$$

and

$$\text{Var}\left[\text{ToRange}\left(z\right)\right] \approx < \gamma^2\left(1/2 + \frac{\sigma}{\gamma} z\right)^2 > -\gamma^2/4$$
$$= \gamma^2\left(1/4 + \frac{\sigma}{\gamma} < z > + \sigma^2/\gamma^2 < z^2 > -1/4\right) = \sigma^2.$$

For mapping to positive values, we expand ELU in $\mu/\sigma - 1 + z$ around zero, which is reasonable if $\mu/\sigma \approx 1$, and find in analogy that target mean and standard deviation are recovered to first order in $\mu/\sigma - 1 + z$.

A.3. Hyperparameters and training details

For all models discussed in this work, we use the hyperparameters listed in Table A5 and a dihedral periodicity of $n_{\text{periodicity}} = 3$. All relative weights between energies, forces and MM parameters rely on units formed by kcal/mol, Å and radian. We train the models using the Adam (Kingma & Ba, 2017) optimizer with $\epsilon = 10^{-8}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$.

First, we train for 2 epochs on traditional MM parameters only, which can be interpreted as a form of pretraining that is efficient in bringing the model in a state where the gradients of the QM-part of the loss functions are informative. We start the training on energies and forces with $\lambda_{\text{trad}} = 10^{-3}$, which we set to zero after 100 epochs, preventing the model from converging towards unphysical local minima of the QM-part of the loss function. When changing weights of terms of the loss function as described above, we re-initialize the optimizer with random moments and apply a warm restart (Loshchilov & Hutter, 2017), during which we linearly increase the learning rate in 500 training steps to decorrelate the optimizer state from the gradient updates of the previous loss function. Besides pretraining on traditional MM parameters directly, we have found penalizing

Hyperparameter	Variable	Value
Graph Neural Network		
Layers	L_{GNN}	7
Hidden dimension	d	512
Attention heads	N_{heads}	16
Embedding dimension	d_{emb}	256
Dropout probability		0.3
Symmetric Transformer		
Transformer layers	L_T	3
Transformer hidden dim	d_T	512
Attention heads	$N_{\text{T-heads}}$	8
Pooling layers	L_{pool}	3
Pooling hidden dim	d_{pool}	256
Dropout probability		0.5
Training setup		
Learning rate		1.5×10^{-5}
Molecules per batch		32
States per molecule		32
Force weight	λ_F	0.8
Dihedral L_2 weight	λ_{dih}	10^{-3}
Traditional MM weight	λ_{trad}	0

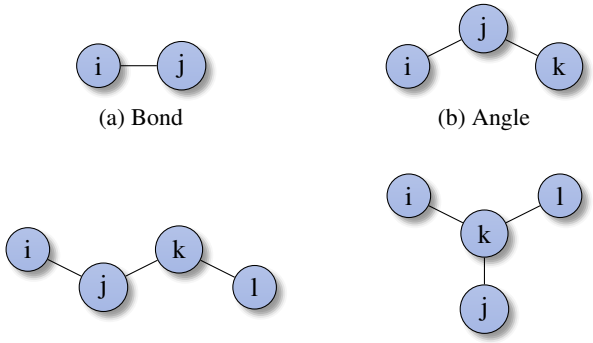
Table A5: The hyperparameters used for training the Grappa models in this work.

large torsion and improper force constants $\xi^{(\text{dih})}$ to benefit generalizability across conformational space.

As validation loss, we use a linear combination of the energy and force root mean squared error (RMSE) on the sub-datasets (the rows in Table 1), averaged over all sub-datasets, with a weight of 1 for the energy RMSE and 3 for the force RMSE to prevent overfitting on a molecule type. We use this validation loss for early stopping and for learning rate scheduling. For the model trained on the Espaloma dataset (Table 1), the validation loss did not improve after 1000 epochs, which corresponds to about 25 hours of training on an A100 GPU.

A.4. Improper dihedrals

As discussed in Section 2.2, we postulate that the energy contributions from the different interactions are symmetric under certain permutations of the embeddings and spatial positions of the atoms involved. These permutations are given by the isomorphisms of the respective subgraph that describes the interaction. In the case of angles, bonds and torsions, these symmetries are given by Eqs. 6-8, for improper dihedrals, the symmetries are given by all six permutations that leave the central atom invariant, as can be seen from the structure of the subgraphs in Figure A8. For bonds, angles and torsions, the interaction coordinates distance, angle and the cosine of the dihedral angle are invariant under the respective permutations, which is why we can construct an invariant energy contribution simply by enforcing invariance of the MM parameters under the respective permutations.



(a) Bond (b) Angle (c) Torsion (d) Improper

Figure A8: The subgraphs of the molecular graph that correspond to bonded MM interactions

For improper dihedrals, this is not the case, as the dihedral angle is not invariant under all six permutations mentioned above and a model with improper force constants that are invariant under these permutations would not have invariant improper energy contributions. In Grappa, we solve this problem by introducing more terms for improper dihedrals. For example

$$E_{\text{improper}} = k_{ijkl}\cos(\phi_{ijkl}) + k_{jikl}\cos(\phi_{jikl}), \tag{39}$$

is invariant under the permutation $ijkl \to jikl$, which can be generalized to all six improper permutations. It turns out that we can reduce the number of additional terms to two by using another symmetry of the dihedral angle,

$$\cos(\phi_{ijkl}) = \cos(\phi_{ljki}), \tag{40}$$

which can be seen from the formula for the dihedral angle in (Blondel & Karplus, 1996). This symmetry allows us to identify pairs of terms that are transformed into each other under the permutation $ijkl \to ljki$ and use a force constant that is invariant under this permutation,

$$k_{ijkl}^{(\text{improper})} = k_{ljki}^{(\text{improper})}. \tag{41}$$

A.5. Datasets

A.5.1. Espaloma dataset

We reproduce the dataset used to train and evaluate Espaloma 0.3 (Takaba et al., 2023) by downloading the pub-

lished, preprocessed data[5]. The preprocessing entails filtering out high-energy conformations as detailed in (Takaba et al., 2023). As in Espaloma, our train-val-test partitioning procedure relies on the unique isomeric SMILES strings of the molecules, which are included in the dataset. We reproduce[6] the partitioning from Espaloma 0.3[7] with the same random seed.

For the calculation of the RMSE in Table 1 and Table 2, we subtract the mean predicted energy for each molecule and for each force field since we are only interested in relative energy differences (and the mean is the unique global shift that minimizes the RMSE). For forces, we report the componentwise RMSE

$$\text{RMSE} \equiv \sqrt{\frac{1}{3N}\sum_{i=1}^{N}\sum_{j=1}^{3}\left(f_{ij} - f_{ij}^{(\text{ref})}\right)^2}$$

as it is done in Espaloma 0.3.

A.5.2. Dataset creation

Scripts for the creation of the datasets are publicly available. [8] We use the Amber ff99SB-ILDN (Lindorff-Larsen et al., 2010) force field for sampling the states and Psi4 (Smith et al., 2020) for the single point QM calculations. For sampling radical peptide states, we use a preliminary Grappa model trained on optimization trajectories and torsion scans of radical peptides. The radical peptides are what we call hydrogen-atom-transfer (HAT) type radicals, that is, they are formed by removing a single hydrogen from a peptide. For the nonbonded contribution, we use the same Lennard-Jones parameters as for the original peptide and add the partial charge of the hydrogen in the original peptide to the heavy atom it was attached to.

A.6. MD simulations

All molecular dynamics simulations are performed with GROMACS version 2023 (Abraham et al., 2015) using the Amber ff99SB-ILDN force field (Lindorff-Larsen et al., 2010) or Grappa-1.3 with Amber ff99SB-ILDN nonbonded parameters and the TIP3P water model. 2 fs time steps with LINCS constraints (Hess) on H-bonds were employed. Simulations were run at 300K and 1 bar, maintained by the v-rescale thermostat (Bussi et al.) and Parrinello-Rahman pressure coupling (Parrinello & Rahman). We apply a Coulomb and Lennard-Jones cutoff of 1.0ṅm. After system preparation, energy minimization, NVT and NPT equilibration simulations are conducted. The code for reproducing the simulations is publicly available. [9]

A.6.1. Ubiquitin simulations

The simulations of ubiquitin (PDB ID: 1UBQ) are evaluated by calculating a moving average over 0.05 ns of the C-alpha RMSD to the initial structure (Figure 5b). We also calculate the statistics of the C-alpha RMSD as function of time difference between two states. To this end, we sample 1,000 states from the trajectory for each time difference and calculate the mean C-alpha RMSD between the states. For the simulation of STMV, we use the same structure files as (Musaelian et al., 2022). We use partial charges and Lennard-Jones parameters of Amber ff99SB-ILDN (Lindorff-Larsen et al., 2010) for the proteins and of RNA.OL3 (Zgarbová et al., 2011) for RNA.

A.6.2. STMV virus

For the simulation of STMV referred to in section 4.2, we use nonbonded parameters from Amber ff99SB and RNA.OL3 repsectively.

Fig. A9 (b) shows the RMSD of protein C-alpha and RNA carbon atoms during 30 ns of MD simulation with Grappa parameters for protein and RNA, which indicates that Grappa keeps the virus stable. For large scale conformational changes or artificial viral capsid disruption, higher RMSD values would be expected.

A.7. Learning curve

We provide a learning curve for Grappa, trained on subsets of the Espaloma dataset, in Fig. A10.

A.8. Grappa's MM parameters

In Fig. A11, the MM parameters predicted by Grappa and ff99-SBILDN for the protein ubiquitin are visualized.

A.9. Force accuracy

In analogy to Fig. 4, Grappa's accuracy for predicted force components is visualized in Fig. A12.

[5] https://github.com/graeter-group/grappa/blob/master/dataset_creation/get_espaloma_split/load_esp_ds.py

[6] https://github.com/graeter-group/grappa/blob/master/dataset_creation/get_espaloma_split/load_esp_ds.py

[7] https://github.com/choderalab/refit-espaloma/blob/main/openff-default/02-train/joint-improper-charge/charge-weight-1.0/train.py, commit 3ccc44d

[8] https://github.com/LeifSeute/grappa-data-creation

[9] https://github.com/LeifSeute/validate-grappa.

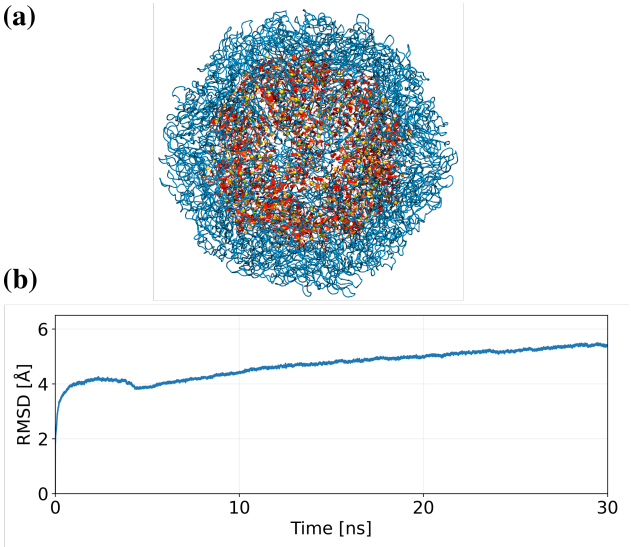


Figure A9: **(a)** The virus STMV, containing proteins (blue) and RNA (red). **(b)** RMSD of protein C-alpha and RNA carbon atoms of the virus in solution during a 30 ns MD simulation with Grappa.

A.10. Extensive tables

We provide tables for with errors for the full Grappa-1.3 and Espaloma datasets in Tab. A6 and A7.

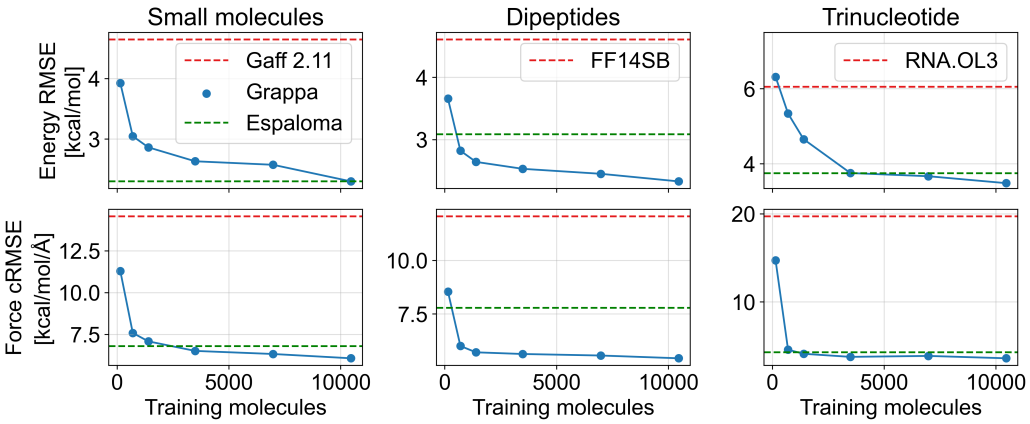


Figure A10: Test RMSE of Grappa (blue), trained on a fraction of the Espaloma-0.3 train dataset, compared with the RMSE of Espaloma-0.3 (green) trained on the full dataset and the RMSE of established force fields.

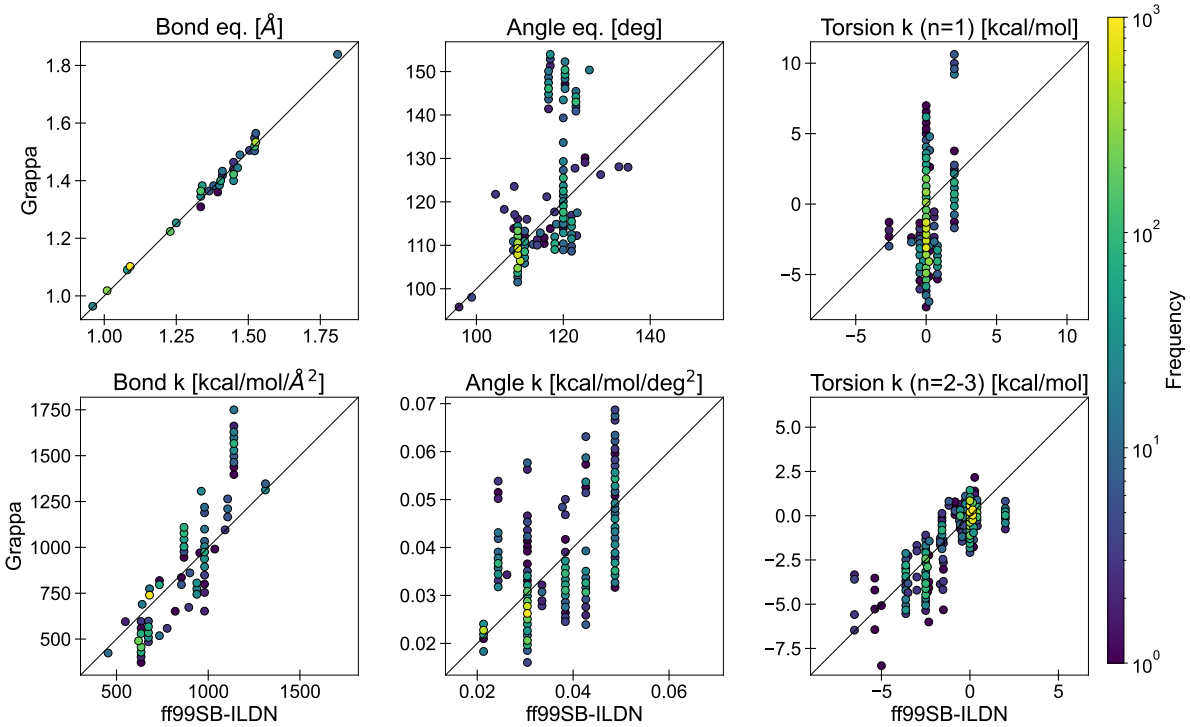


Figure A11: MM parameters for the protein ubiquitin, predicted by Grappa-1.3 and ff99SB-ILDN. The comparison suggests that Grappa predicts more continuous parameter sets than the tabulated force field ff99SB-ILDN.

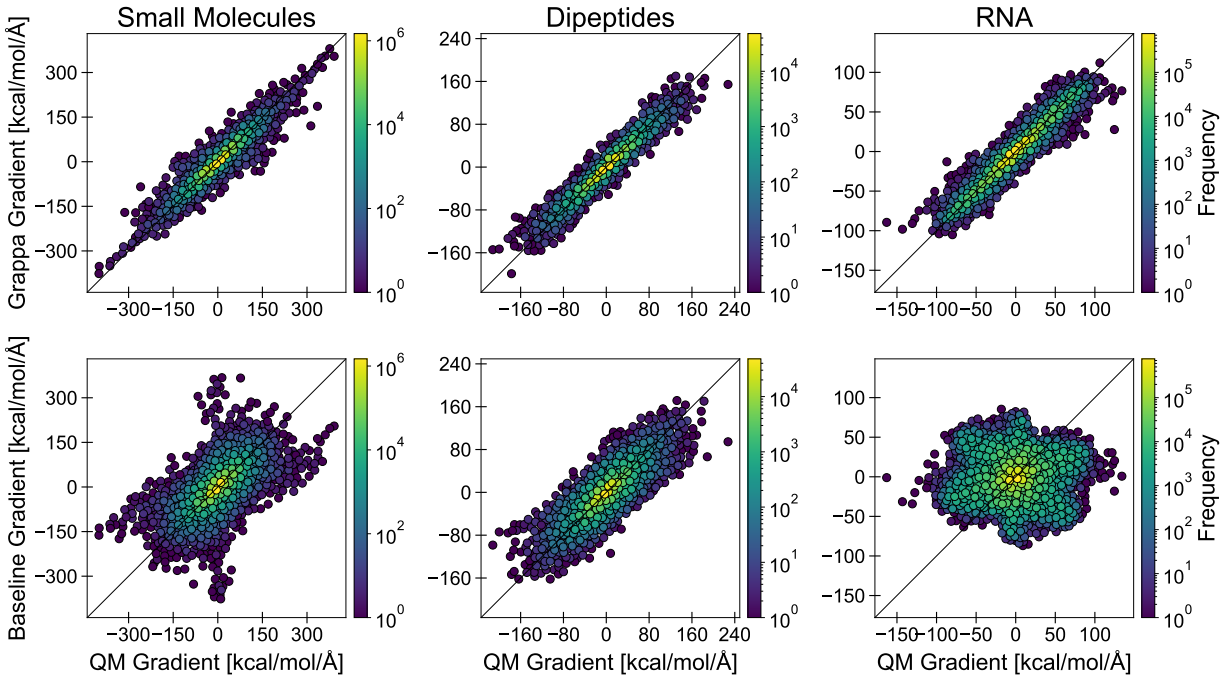


Figure A12: Comparison of gradient predictions of Grappa-1.3 and the established force fields Gaff-2.11, ff99SB-ILDN and RNA.OL3 for test molecules from Espaloma's SPICE-Pubchem, SPICE-Dipeptide and RNA-Trinucleotide datasets. For SPICE-PubChem, five molecules were filtered out because they had unphysically high gradient magnitudes of more than 400 kcal/mol/Å.

A.11. Detailed report on accuracies

Dataset	Test Mols	Confs		Grappa	Espaloma	Gaff-2.11	ff14SB, RNA.OL3	Mean Predictor
BOLTZMANN SAMPLED								
SPICE-Pubchem	1411	60853	*Energy*	**2.3** $\pm$ 0.1	**2.3** $\pm$ 0.1	4.6		18.4
			Force	**6.1** $\pm$ 0.3	6.8 $\pm$ 0.1	14.6		23.4
SPICE-DES-Monomers	39	2032	*Energy*	**1.3** $\pm$ 0.1	1.4 $\pm$ 0.3	2.5		8.2
			Force	**5.2** $\pm$ 0.2	5.9 $\pm$ 0.5	11.1		21.3
SPICE-Dipeptide	67	2592	*Energy*	**2.3** $\pm$ 0.1	3.1 $\pm$ 0.1	4.5	4.6	18.7
			Force	**5.4** $\pm$ 0.1	7.8 $\pm$ 0.2	12.9	12.1	21.6
RNA-Diverse	6	357	*Energy*	**3.3** $\pm$ 0.2	4.2 $\pm$ 0.3	6.5	6.0	5.4
			Force	**3.7** $\pm$ 0.04	4.4 $\pm$ 0.1	16.7	19.4	17.1
RNA-Trinucleotide	64	35811	*Energy*	**3.5** $\pm$ 0.1	3.8 $\pm$ 0.2	5.9	6.1	5.3
			Force	**3.6** $\pm$ 0.01	4.3 $\pm$ 0.1	17.1	19.7	17.7
TORSION SCAN								
Gen2-Torsion	131	21890	*Energy*	1.7 $\pm$ 0.2	**1.6** $\pm$ 0.3	2.7		4.7
			Force	**4.0** $\pm$ 0.4	4.7 $\pm$ 0.6	9.4		5.5
Protein-Torsion	9	6624	*Energy*	2.2 $\pm$ 0.4	**1.9** $\pm$ 0.2	3.0		3.5
			Force	3.8 $\pm$ 0.5	**3.5** $\pm$ 0.3	9.7		5.1
OPTIMIZATION								
Gen2-Opt	154	40055	*Energy*	1.8 $\pm$ 0.2	**1.7** $\pm$ 0.5	3.0		3.9
			Force	**3.8** $\pm$ 0.2	4.5 $\pm$ 0.8	9.7		5.1
Pepconf-Opt	55	14884	*Energy*	3.2 $\pm$ 0.3	**2.8** $\pm$ 0.3	5.1	4.1	6.3
			Force	**3.6** $\pm$ 0.2	4.0 $\pm$ 0.4	10.2	10.2	5.3

Table A6: Accuracy of Grappa, Espaloma 0.3 (Takaba et al., 2023) and established MM force fields on test molecules from the Espaloma dataset. We report the RMSE of molwise-centered energies in kcal/mol and the componentwise RMSE of forces in kcal/mol/Å. Gaff-2.11 (He et al., 2020) is a general-purpose force field, ff14SB (Maier et al., 2015) is an established protein force field and RNA.OL3 (Zgarbová et al., 2011) is specialized to RNA. As in Espaloma, we bootstrap the set of test molecules 1000 times and report mean and standard deviation of the RMSEs.

Dataset	Test Mols	Confs		Grappa	Espaloma	Gaff-2.11	ff99SB-ILDN	Mean Predictor
SPICE-PubChem	1411	60853	*Energy*	**2.3** $\pm$ 0.1	**2.3** $\pm$ 0.1	4.6 $\pm$ 0.1		18.8
			Force	**6.1** $\pm$ 0.3	6.8 $\pm$ 0.1	14.6 $\pm$ 0.3		41.3
SPICE-DES-Monomers	39	2032	*Energy*	**1.3** $\pm$ 0.1	1.4 $\pm$ 0.3	2.5 $\pm$ 0.2		8.6
			Force	**5.3** $\pm$ 0.2	5.9 $\pm$ 0.5	11.1 $\pm$ 0.6		37.4
SPICE-Dipeptide	67	2592	*Energy*	**2.4** $\pm$ 0.1	3.1 $\pm$ 0.1	4.5 $\pm$ 0.1		19.1
			Force	**5.4** $\pm$ 0.1	7.8 $\pm$ 0.2	12.9 $\pm$ 0.3		38.4
RNA-Diverse	6	357	*Energy*	**3.2** $\pm$ 0.2	4.2 $\pm$ 0.3	6.5 $\pm$ 0.1		6.1
			Force	**3.7** $\pm$ 0.0	4.4 $\pm$ 0.1	16.8 $\pm$ 0.1		30.2
RNA-Trinucleotide	64	23811	*Energy*	**3.5** $\pm$ 0.0	3.8 $\pm$ 0.2	6.0 $\pm$ 0.1		6.1
			Force	**3.6** $\pm$ 0.0	4.3 $\pm$ 0.1	17.0 $\pm$ 0.0		31.2
Dipeptides-300K	72	3600	*Energy*	**2.6** $\pm$ 0.1		4.5 $\pm$ 0.1	4.1 $\pm$ 0.1	7.8
			Force	**5.9** $\pm$ 0.1		10.9 $\pm$ 0.3	11.8 $\pm$ 0.1	43.7
Dipeptides-1000K	72	2160	*Energy*	**5.4** $\pm$ 0.1		8.6 $\pm$ 0.2	8.5 $\pm$ 0.2	18.9
			Force	**11.6** $\pm$ 0.1		16.1 $\pm$ 0.2	17.8 $\pm$ 0.1	74.3
Non-Capped-Peptides	10	500	*Energy*	**2.2** $\pm$ 0.2		4.2 $\pm$ 0.4	4.0 $\pm$ 0.3	7.2
			Force	**6.1** $\pm$ 0.2		12.1 $\pm$ 0.7	12.6 $\pm$ 0.5	45.8
Radical-Dipetides	28	272	*Energy*	**3.3** $\pm$ 0.3				8.7
			Force	**6.8** $\pm$ 0.2				41.3
OPTIMIZATION								
Gen2-Opt	154	29055	*Energy*	**1.7** $\pm$ 0.1	**1.7** $\pm$ 0.5	2.8 $\pm$ 0.2		4.2
			Force	**4.0** $\pm$ 0.2	4.5 $\pm$ 0.8	9.8 $\pm$ 0.3		8.7
Pepconf-Opt	55	9084	*Energy*	**2.8** $\pm$ 0.2	**2.8** $\pm$ 0.3	4.7 $\pm$ 0.3		6.5
			Force	**3.7** $\pm$ 0.2	4.0 $\pm$ 0.4	10.4 $\pm$ 0.3		9.4
TORSION SCAN								
Gen2-Torsion	131	19290	*Energy*	1.7 $\pm$ 0.1	**1.6** $\pm$ 0.3	2.6 $\pm$ 0.1		4.7
			Force	**4.2** $\pm$ 0.2	4.7 $\pm$ 0.6	9.5 $\pm$ 0.4		8.9
Protein-Torsion	9	6024	*Energy*	**1.9** $\pm$ 0.2	**1.9** $\pm$ 0.2	2.9 $\pm$ 0.2		3.5
			Force	4.2 $\pm$ 0.3	**3.5** $\pm$ 0.3	9.8 $\pm$ 0.4		9.0

Table A7: Accuracy of Grappa-1.3 with nonbonded parameters from the AM1-BCC scheme, Grappa with nonbonded parameters from Amber ff99SB-ILDN, Espaloma 0.3 (Takaba et al., 2023) and established MM force fields on test molecules of the Grappa-1.3 dataset. We report the RMSE of molwise-centered energies in kcal/mol and the componentwise RMSE of forces in kcal/mol/Å.