Supporting Information

Data-Driven Parametrization of Molecular Mechanics Force Fields for Expansive Chemical Space Coverage

Tianze Zheng,*,† Ailun Wang,† Xu Han,† Yu Xia,† Xingyuan Xu,† Jiawei Zhan,†,‡ Yu Liu,† Yang Chen,† Zhi Wang,† Xiaojie Wu,† Sheng Gong,† and Wen Yan*,†

†*ByteDance Research*

‡*work done as intern at ByteDance Research*

E-mail: zhengtianze@bytedance.com; wen.yan@bytedance.com

Contents

Supporting Figures

1 Dataset construction

1.1 Selection of raw molecules

To provide a comprehensive coverage of the chemical space, we chose ChEMBL,[1] a large-scale bioactivity database for drug discovery, as the primary source of raw molecules. To enhance data quality, we applied filters to retain molecules with fewer than 10 aromatic rings, a polar surface area (PSA) below 200 $Å^2$, and a quantitative estimate of drug-likeness (QED) greater than 0.2. Additionally, expert-curated molecules from ZINC20[2] were incorporated to further broaden the chemical space coverage. We also filtered the raw molecules by element and hybridization type, retaining only those with elements C, H, O, N, P, S, F, Cl, Br, I, and hybridization types s, sp, sp2, and sp3. In total, 2.46 million molecules represented by SMILES were prepared for fragmentation.

1.2 Fragmentation algorithm

Starting from the selected raw molecules, we employed an in-house graph expansion-based fragmentation algorithm to generate molecular fragments (See Fig. S1). As discussed in the main article, the fragments should be sufficiently large to capture the local environment of bonded structures, while being small enough to eliminate long range interactions and optimize computational efficiency. For each raw molecule, we traversed every bond, angle and non-ring torsion, retraining their neighboring conjugated atoms while severing distant atoms. The severed bonds were capped by hydrogen atoms (aliphatic carbons), or methyl groups (other heavy atoms). The resulting fragments were then duplicated by their SMILES patterns and filtered by the number of atoms, excluding those with more than 70 atoms.

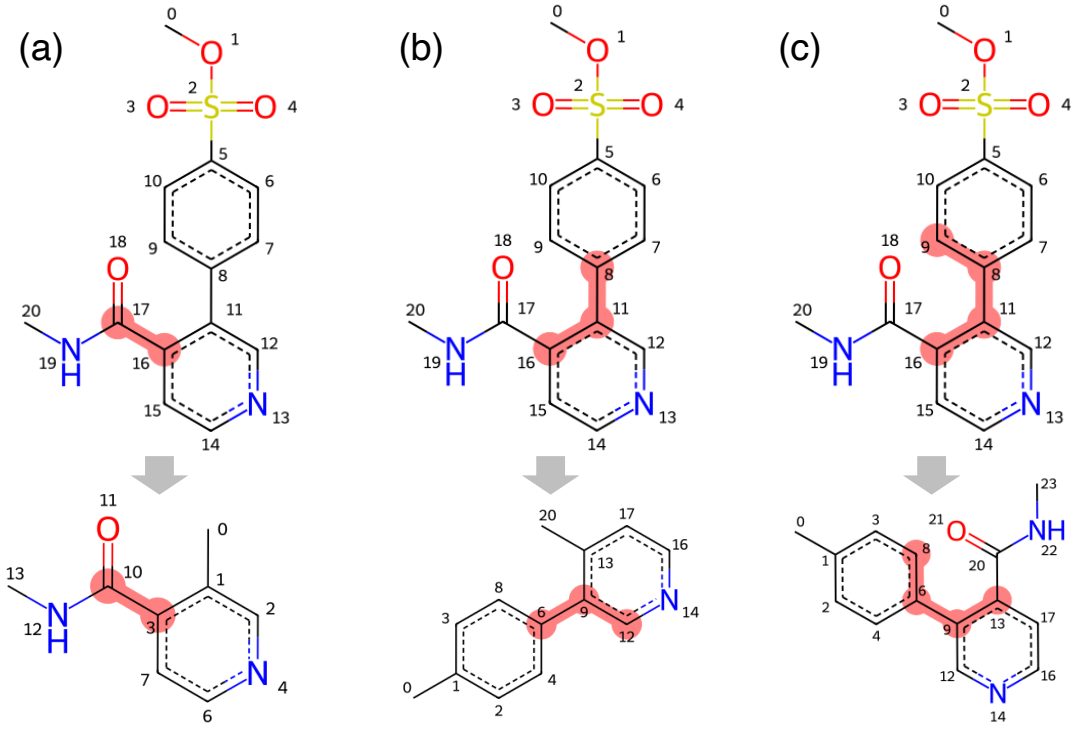

Figure S1: **Fragmentation examples.** Fragments (bottom) of the same raw molecule (top) from the highlighted bond (a), angle (b) and torsion (c).

1.3 Enumeration of protonation states and pKa filtering

Afterward, the fragments were expanded by enumerating possible protonation states using a straightforward reaction template method.[3] The simplicity of these templates, however, resulted in some improbable protonation states. To address this, we filtered the states using a pKa range from 0 to 14. Classic Epik (version 6.5)[4] was utilized to evaluate the micro-pKa of titratable sites. Fragments with a predicted micro-pKa of their protonated titratable sites smaller than 0 or deprotonated titratable sites greater than 14 were excluded. This relatively loose filtering condition was employed to encompass most protonation states likely encountered in simulations. Ultimately, 2.4 million fragments with unique SMILES were obtained.

1.4 QM methods and workflow

From the 2.4 million fragments, two QM datasets, namely *optimization dataset* and *torsion dataset*, were constructed at the B3LYP-D3(BJ)/DZVP[5] level of theory, using Q-Chem 6.1[6] and GPU4PySCF 1.0[7] as QM engines, respectively. When generating the *optimization dataset*, initial 3D conformation of each fragment were generated from its SMILES pattern using the RDKit software,[8] followed by up to 200 rounds of geometric optimization using the geomeTRIC[9] optimizer. Single point energy and Hessian were then calculated using Q-Chem 6.1[6] with unpruned SG-2 (75/302) and unpruned SG-3 (99/590) grids, respectively. For the *torsion dataset*, two subsets were curated separately: non-ring torsions and in-ring torsions. Starting from the conformations in the *optimization dataset*, each non-ring torsion was initially scanned by rotating the torsion angle in 15° increments, generating 24 initial frames that were then optimized using the geomeTRIC optimizer with the torsion angle constrained. Additionally, 1 million in-ring torsions were sampled from the *optimization dataset* and scanned using a frame-by-frame approach with a 20 kcal/mol energy threshold, as they might not encompass the full 360° space.

1.5 QM engine

In the curation of *torsion dataset*, we used GPU4PySCF 1.0[7] instead of Q-Chem 6.1 as the QM engine, which accelerated nearly 50 million conformation optimizations, involving over 2 billion single-point energy and gradient calculations. To ensure the consistency, we retained the B3LYP-D3(BJ) functional and DZVP basis set as used in Q-Chem calculations. Benchmarking on 10 typical molecules at both B3LYP/def2-SVP and B3LYP/def2-TZVPP levels, GPU4PySCF demonstrated remarkable accuracy, with a maximum energy difference of 6×10^{-5} Hartree and a gradient discrepancy within 2×10^{-4} Hartree/Bohr compared to Q-Chem 6.1. These results aligned with our own measurements using B3LYP-D3(BJ)/DZVP. Additionally, the grid density in GPU4PySCF was set to 75/302 to match the computational parameters in Q-Chem.

1.6 Filtering QM data

After optimization, we applied filters to the structural changes, specifically focusing on covalent bond breakages and new bond formations. Covalent bond breakages were identified if the bond length exceeded 120% of the single bond lengths of corresponding elements from OpenFF. MDAnalysis[10] software was employed to check if the distances between non-covalently connected atom pairs exceeded their respective vdW radii, indicating potential new bond formation. After excluding conformers with bond breakages or new bond formations during optimization, the remaining conformers constituted the *torsion dataset*.

For Hessian matrices, we applied sanity check using their eigenvalues. Since the Hessian should be positive definite for an equilibrium state, we discarded data with the smallest eigenvalue lower than -2 kcal/mol/Å^2. Additionally, when the smallest absolute eigenvalue was much greater than zero (we used a threshold of 10 kcal/mol/Å^2), indicating the violation of the translational invariance, the data was also discarded.

1.7 Diversity of chemical space

The diversity of chemical space covered by our curated datasets is measured using the Morgan fingerprint, with the size of the fingerprint set to 2048 and a radius of 2, focusing on the two central atoms in the selected torsion. This torsional fingerprints were also used to evaluate the diversities of different datasets for comparison. For each dataset, we traversed every torsion in each molecule, recording the torsional fingerprint only if its circular standard deviation (std.) is greater than 0.3. This threshold corresponds to sampling within a range of 60 degrees approximately, ensuring sufficient exploration of the torsional angles space. Torsional fingerprints were calculated for SPICE, GEOM and our curated datasets as 2048-dimensional data for each torsion, which were reduced to 50 dimensions and further to 2 dimensions using t-SNE for visualization.

1.8 Diverse test dataset

From our curated *torsion dataset*, a diverse torsion dataset was extracted using the torsional fingerprints, to serve as a robust and challenging benchmark for assessing the performance of force fields in predicting torsional energy profiles. To construct such diverse test set, the *torsion dataset* was filtered by a stricter pKa range from 5.4 to 9.4 (see Section 1.3). The qualified fragments were split into two subsets: in-ring and non-ring, according to the nature of the rotated bond. In each subset, torsional fingerprints were calculated for each molecule on its rotated torsion, with similarities measured by the Tanimoto similarity coefficient. A diverse set of 1000 molecules was then selected from the in-ring and non-ring subsets respectively using the min-max algorithm, resulting the ByteDance diverse torsion dataset(BDTorsion). The remaining *torsion dataset* was used as the training dataset for ByteFF.

2 Model architecture

2.1 Input features

For comprehensive description of the molecular graph, both atomic and bond features were extracted from the molecular graph. Atom features include the element type, ring connectivity, minimum ring size, and formal charge of each atom. While bond features include bond order, and whether the bond is in ring. After continuous vectorial embedding for both atom and bond features, they were further concatenated into node and edge feature embeddings, respectively. Embeddings for each node or bond were then independently processed through a few MLP layers to produce the output embeddings, before being fed into the node and edge channels of the modified Edge-augmented Graph Transformer (EGT) architecture.

2.2 Neural network architecture

ByteFF employed a modified EGT architecture[11] to learn molecular mechanics force field parameters from the molecular graph. A main characteristic of the EGT architecture is the residual edge channel, which incorporate the edge features into the original node-based attention mechanism. In this architecture, node embeddings get updated through a self-attention mechanism in node channels, while edge channels, which keep track of the bond features, directly contribute to this process, allowing the graph structure to evolve dynamically. The interaction between node and edge channels ensure that the structural information evolves from layer to layer, enabling the model to learn the structural features of the molecular graph. Instead of using the vanilla form of the EGT architecture, we replaced the global self-attention mechanism in EGT with the local self-attention mechanism such that the attention mechanism is constrained to the local neighborhood of each node. Additionally, the hidden states of the bidirectional edges were averaged, so that $h_{e_{ij}} = h_{e_{ji}}$.

2.3 Symmetry preserving mechanisms

When describing a molecular graph, it is important to recognize the chemical symmetry of the molecule. Accurate identifying chemically equivalent atoms and bonds not only helps to provide reasonable node/edge embeddings in the featurization module, it also ensures permutation invariance when deriving force field parameters in output module.

For each molecule, atom and bond features were extracted independently, and canonical atom rankings were calculated using the RDKit package to identify chemically equivalent atoms and bonds. In the featurization module, embeddings of chemically equivalent atoms and bonds were averaged to preserve chemical symmetry of the molecule. After the GNN module, hidden states were processed through the edge-augmented symmetry-preserving pooling to obtain symmetric input for the MLP layers in the output module. With respect to the specific permutation invariance of bond, angle, proper, and improper torsion, the edge-augmented symmetry-preserving pooling is processed in the following manner:

$$
\begin{aligned}
h_{r_{ij}} =& NN_r([h_{x_i} : h_{e_{ij}} : h_{x_j}]) + NN_r([h_{x_j} : h_{e_{ji}} : h_{x_i}]) \\
h_{\theta_{ij}} =& NN_\theta([h_{x_i} : h_{e_{ij}} : h_{x_j} : h_{e_{jk}} : h_{x_k}]) \\
&+ NN_\theta([h_{x_k} : h_{e_{kj}} : h_{x_j} : h_{e_{ji}} : h_{x_i}]) \\
h_{\phi_{ijkl}} =& NN_\phi([h_{x_i} : h_{e_{ij}} : h_{x_j} : h_{e_{jk}} : h_{x_k} : h_{e_{kl}} : h_{x_l}]) \\
&+ NN_\phi([h_{x_l} : h_{e_{lk}} : h_{x_k} : h_{e_{kj}} : h_{x_j} : h_{e_{ji}} : h_{x_i}]) \\
h_{\psi_{ijkl}} =& NN_\psi([h_{x_i} : h_{e_{ij}} : h_{x_j}]) \\
&+ NN_\psi([h_{x_i} : h_{e_{ik}} : h_{x_k}]) \\
&+ NN_\psi([h_{x_i} : h_{e_{il}} : h_{x_l}])
\end{aligned}
\tag{1}
$$

Where h_{x_i} denotes node hidden states of atom i, $h_{e_{ij}}$ denotes edge hidden states between atom i and j, NN_r, NN_θ, NN_ϕ and NN_ψ denote the MLP layers for bond, angle, proper, and improper torsions, respectively. The colon sign (:) denotes concatenation. Following this process, the permutation invariance is evident as in $h_{r_{ij}} = h_{r_{ji}}$, $h_{\theta_{ijk}} = h_{\theta_{kji}}$, $h_{\phi_{ijkl}} = h_{\phi_{lkji}}$.

For improper dihedral, with atom i being the center atom, the value of $h_{\psi_{ijkl}}$ remains the same for all permutations of j, k, l in this notation.

2.4 Total charge preserving

In the output module, the total charge of the molecule was preserved by using the bond charge correction (BCC) algorithm. Instead of directly predicting the partial charge on each atom, we predicted the charge transferred across each bond and summed them up to obtain the charge on each atom. In the BCC algorithm, two atoms involved in a bond were perturbed by equal amount but opposite charges, ensuring no net gain or loss of charge on the molecule level, thus preserving the total charge of the molecule.

To achieve this, the transferred charge across bond between atom i and j, q_{ij}, is predicted by an anti-symmetric function under exchange of atom i and j:

$$q_{ij} = f\left(NN_q([h_{x_i} : h_{e_{ij}} : h_{x_j}]) - NN_q([h_{x_j} : h_{e_{ji}} : h_{x_i}])\right), \tag{2}$$

where function f is an odd function, modeled by an MLP with no bias and tanh activation function. Then, q_{ij}s are summed up to obtain the charge on each atom:

$$q_i = q_i^0 + \sum_{j \in \mathcal{N}(i)} q_{ij}, \tag{3}$$

where $\mathcal{N}(i)$ denotes the neighbors of atom i, and q_i^0 is the formal charge of atom i averaged according to the chemical symmetry.

2.5 Hyperparameter configurations

Following the model architecture defined above, following hyperparameter configurations were used in the ByteFF model: In the featurization processes, both node and edges embeddings were generated by 2 MLP layers with 64 hidden units and GELU activation. In

the GNN layers, 4 layers of modified EGT were used with 256 hidden units and GELU activation. In the output module, the node and edge hidden states were first post-processed by 4 MLP layers with 512 hidden units and GELU activation, followed by edge-augmented symmetry-preserving pooling and total charge preserving treatment utilizing the bond charge correction algorithm. The force field parameters were predicted by 4 MLP layers with 256 hidden units and GELU activation. Five models, each initialized with different random seeds, were trained. The final force field parameters were obtained by averaging the predictions from these five models.

3 Training details

3.1 Training datasets

As documented in Table 1, three datasets were involved in the training procedure: *optimization dataset*, *torsion dataset* and *off-equilibrium dataset*. The *off-equilibrium dataset* contains three sub-dataset used by Espaloma-0.3.0,[12] namely SPICE-Pubchem, SPICE-Dipeptide and RNA-Diverse. The *optimization dataset* and *torsion dataset* were split into training and validation sets with a ratio of 9:1, while the *off-equilibrium dataset* was split into training/validation and test sets following the same manner in Espaloma-0.3.0.

3.2 Training strategies

The training procedure of ByteFF models comprises three stages: pre-training, training, and the optional fine-tuning stage.

In the pre-training stage, the goal was to optimize the model to learn reasonable force field parameters using the *optimization dataset*. This involved fitting various parameters through multiple well-designed loss functions: (1) Non-bonded parameters and force constants of proper torsions were fitted to those used by GAFF-2.2 with mean squared error (MSE) losses. (2) Equilibrium values of bonded parameters were optimized with energy-based loss functions w.r.t. corresponding terms on conformations in the *optimization dataset*. (3) Force constants were fitted by minimizing the discrepancies of Hessian matrices between QM and the force field, using the mean absolute percentage error (MAPE) losses. Here, only the Hessian blocks corresponding to bond and angle terms were fitted, leading to the term "partial Hessian loss".

For the first loss, GAFF-2.2 parameters with AM1BCC charges were obtained for all molecules in the *optimization dataset* using Antechamber.[13,14] From the resulting itp files, the σ, ϵ and q values were extracted and served as labels for training non-bonded parameters. Meanwhile, the values of proper torsion force constants were used as labels to train proper

torsions. After pre-training, the model was able to accurately reproduce the GAFF-2.2 non-bonded parameters and proper torsion force constants. The root mean squared error (RMSE) for σ, ϵ, q and proper torsion k on the validation set was 1.1×10^{-3} Å, 4.2×10^{-4} kcal/mol, 1.3×10^{-2} elementary charges and 7.0×10^{-2} kcal/mol, respectively.

In the training stage, the curated *torsion dataset* was incorporated to train the force constants of proper torsion using the Boltzmann MSE losses,[15] while other force field parameters were trained using the same loss functions and labels as in the pre-training stage. Due to the limitation of functional forms in MMFF, it is usually considered unattainable to perfectly fit QM PES using an MMFF. With the focus on torsional energy profiles, we chose to fit the projection of PES on the torsional degree of freedom, using an iterative optimization-and-training approach. Such approach includes (1) optimizing the QM conformations from *torsion datasets* using the force field, with the scanned torsion constrained and atoms restrained, (2) training the force field parameters on the optimized conformations, and (3) attenuate the strength of the positional restraints as the force field's accuracy improved. In the training processes, L1-norm regularization was applied to the force constants of proper torsions to restrain redundant degrees of freedom.

After the training stage, an optional fine-tuning stage was performed to further improved the model's performance by incorporating the *off-equilibrium dataset* and refine the force field parameters with QM energy and forces.

3.3 Loss functions

In the multi-stage training procedures, various forms of loss functions were employed to optimize the performance of ByteFF models in an efficient and robust manner. The detailed loss functions are described in the follows.

3.3.1 MSE loss

In the pre-training, training, and fine-tuning stages, non-bonded parameters, including σ, ϵ and q in Eq. 2, were trained using the MSE loss functions.

$$\begin{aligned}
\mathcal{L}^{\sigma}_{\mathrm{MSE}} =& \frac{1}{N_{\mathrm{atom}}} \sum_{N_{\mathrm{atom}}} (\sigma_i - \hat{\sigma}_i)^2 \\
\mathcal{L}^{\epsilon}_{\mathrm{MSE}} =& \frac{1}{N_{\mathrm{atom}}} \sum_{N_{\mathrm{atom}}} (\epsilon_i - \hat{\epsilon}_i)^2 \\
\mathcal{L}^{q}_{\mathrm{MSE}} =& \frac{1}{N_{\mathrm{atom}}} \sum_{N_{\mathrm{atom}}} (q_i - \hat{q}_i)^2
\end{aligned} \tag{4}$$

in which N_{atom} is the number of atoms, $\hat{\sigma}$, $\hat{\epsilon}$ and $\hat{q}$ denote model predictions.

In the pre-training stage, MSE loss was also used to fit the force constants of proper torsions $k_\phi^{n_\phi}$.

$$\mathcal{L}^{k_\phi}_{\mathrm{MSE}} = \frac{1}{4N_{\mathrm{proper}}} \sum_{\mathrm{propers}} \sum_{n_\phi} (k_\phi^{n_\phi} - \hat{k}_\phi^{n_\phi})^2 \tag{5}$$

in which n_ϕ refers to the periodicity of proper torsions and $\hat{k}_\phi$ denotes model prediction. Using a periodicity of 4 instead of 1 or 2 as typically used in GAFF, we labeled $k_\phi^{n_\phi}$ terms missing in GAFF as zeros.

In the fine-tuning stage, MSE losses were used to fit both force and energy.

$$\mathcal{L}^{\mathrm{Force}}_{\mathrm{MSE}} = \frac{1}{3N_{\mathrm{atom}}} \sum_{N_{\mathrm{atom}}} \left| \mathbf{F}_i^{\mathrm{QM}} - \mathbf{F}_i^{\mathrm{MM}} \right|^2 \tag{6}$$

$$\mathcal{L}^{\mathrm{Energy}}_{\mathrm{MSE}} = \frac{1}{N_{\mathrm{conf}}} \sum_{N_{\mathrm{conf}}} \left[\mathbf{E}_i^{\mathrm{QM}} - \mathbf{E}_i^{\mathrm{MM}} - \frac{1}{N_{\mathrm{conf}}} \sum_{N_{\mathrm{conf}}} (\mathbf{E}_i^{\mathrm{QM}} - \mathbf{E}_i^{\mathrm{MM}}) \right]^2 \tag{7}$$

The force field predicted energy ($\mathbf{E_i^{MM}}$) and labeled energy ($\mathbf{E_i^{QM}}$) of each conformation were aligned by the average of all conformations. The gradient of $\mathcal{L}^{\mathrm{Force}}_{\mathrm{MSE}}$ and $\mathcal{L}^{\mathrm{Energy}}_{\mathrm{MSE}}$ w.r.t. non-bonded parameters were truncated.

3.3.2 Energy-based loss

Energy-based loss functions were used to train equilibrium values of bond length (r_0), angle (θ_0), and improper torsion force constants (k_ψ) because corresponding energies should be minimized at equilibrium conformations, and their harmonic formulations are well-suited for training. During the training process, the gradients of other force constants were clamped to ensure only these terms are trainable.

$$\begin{aligned} \mathcal{L}^{r_0}_{E_{\text{bond}}} &= \frac{1}{N_{\text{bond}}} E^{\text{MM}}_{\text{bond}} \\ \mathcal{L}^{\theta_0}_{E_{\text{angle}}} &= \frac{1}{N_{\text{angle}}} E^{\text{MM}}_{\text{angle}} \\ \mathcal{L}^{k_\psi}_{E_{\text{improper}}} &= \frac{1}{N_{\text{improper}}} E^{\text{MM}}_{\text{improper}} \end{aligned} \tag{8}$$

Since the equilibrium value of improper was fixed to 180° to maintain the planarity of corresponding structures, $\mathcal{L}^{k_\psi}$ were used to prevent large k_ψ when equilibrium structures significantly deviate from planarity. The gradients w.r.t. other force field parameters were truncated.

3.3.3 Partial Hessian loss

When training force constants of bond (k_r), angle (k_θ), and improper (k_ψ), we used the MAPE of partial Hessian blocks as the loss function. This loss was designed utilizing the energy decomposition nature in MMFF, which avoided the $O(n^2)$ complexity of calculating the full Hessian matrix. For example, the partial Hessian block relating to atom i and j contributed by bond $(i-j)$ can be represented as:

$$\mathbf{H}^{\text{bond}_{ij}}_{ij} = \begin{pmatrix} \frac{\partial^2 E^{\text{MM}}_{\text{bond}_{ij}}}{\partial x_i x_j} & \frac{\partial^2 E^{\text{MM}}_{\text{bond}_{ij}}}{\partial x_i \partial y_j} & \frac{\partial^2 E^{\text{MM}}_{\text{bond}_{ij}}}{\partial x_i \partial z_j} \\ \frac{\partial^2 E^{\text{MM}}_{\text{bond}_{ij}}}{\partial y_i \partial x_j} & \frac{\partial^2 E^{\text{MM}}_{\text{bond}_{ij}}}{\partial y_i y_j} & \frac{\partial^2 E^{\text{MM}}_{\text{bond}_{ij}}}{\partial y_i \partial z_j} \\ \frac{\partial^2 E^{\text{MM}}_{\text{bond}_{ij}}}{\partial z_i \partial x_j} & \frac{\partial^2 E^{\text{MM}}_{\text{bond}_{ij}}}{\partial z_i \partial y_j} & \frac{\partial^2 E^{\text{MM}}_{\text{bond}_{ij}}}{\partial z_i z_j} \end{pmatrix}, \tag{9}$$

In addition to H_{ij}, the same bond $(i-j)$ also contributes to another partial Hessian block H_{ji}. Similarly, one angle contributes to six partial Hessian blocks, and one proper or improper torsion contributes to twelve blocks. Summing up all the contributions, the partial Hessian block predicted by the force field was obtained:

$$\mathbf{H}_{ij}^{\mathrm{MM}} = \mathbf{H}_{ij}^{\mathrm{bond}} + \mathbf{H}_{ij}^{\mathrm{angle}} + \mathbf{H}_{ij}^{\mathrm{proper}} + \mathbf{H}_{ij}^{\mathrm{improper}}. \tag{10}$$

The MAPE loss was then used to evaluate discrepancies of partial Hessian blocks between QM and force field predictions:

$$\mathcal{L}_{\mathrm{MAPE}}^{k_r,k_\theta,k_\psi} = \frac{1}{2 \times (N_{\mathrm{bond}} + N_{\mathrm{angle}})} \sum_{(ij)\in \mathrm{bonds,angles}} \frac{\sum |\mathbf{H}_{ij}^{\mathrm{QM}} - \mathbf{H}_{ij}^{\mathrm{MM}}|}{\max(|\mathrm{tr}(\mathbf{H}_{ij}^{\mathrm{QM}})|, LB)}. \tag{11}$$

The discrepancy between QM and MM was normalized by the trace of QM partial Hessian blocks, as it was a rotation invariant of the partial Hessian. The trace was clamped to a lower bound of LB, which was set to 10.0 in the pre-training stage. The gradient w.r.t. force field parameters except for k_r, k_θ and k_ψ were truncated.

3.3.4 Boltzmann MSE loss

When trained with the *torsion dataset*, force constants of proper torsions (k_ϕ) were fitted using the Boltzmann MSE loss, similar to that used by OPLS-AA.[16] In the *torsion dataset*, each molecule contains multiple (N_{conf}) conformations. Before calculating the loss function, the force field predicted energy (E^{MM}) and QM energy (E^{QM}) of each conformation were aligned w.r.t the corresponding minimum energy of all configurations:

$$\begin{aligned} \hat{E}_i^{\mathrm{QM}} &= E_i^{\mathrm{QM}} - \min_i(E_i^{\mathrm{QM}}), \\ \hat{E}_i^{\mathrm{MM}} &= E_i^{\mathrm{MM}} - \min_i(E_i^{\mathrm{MM}}). \end{aligned} \tag{12}$$

The Boltzmann MSE loss was then calculated by reweighing the MSE error using the Boltzmann formulation weight.

$$\mathcal{L}^{k_\phi}_{\text{Bolt\_MSE}} = \frac{1}{N_{\text{conf}}} \sum_{i=1}^{N_{\text{conf}}} w_i \cdot \left(\hat{E}_i^{\text{QM}} - \hat{E}_i^{\text{MM}} - \frac{\sum_{i=1}^{N_{\text{conf}}} (w_i (\hat{E}_i^{\text{QM}} - \hat{E}_i^{\text{MM}}))}{\sum_{i=1}^{N_{\text{conf}}} w_i} \right)^2, \tag{13}$$

where $w_i = \min\left\{1.0, \exp\left(\frac{\alpha - \min(\hat{E}_i^{\text{QM}}, \hat{E}_i^{\text{MM}})}{\beta}\right)\right\}$, while energy clamp α and decay weight β are hyperparameters of the Boltzmann weighting function. Gradients with respect to all force field parameters, except for k_ϕ, were truncated. Additionally, gradients with respect to k_ϕ were truncated for torsion angles with a circular standard deviation below 0.3, as insufficient sampling leads to large uncertainty which is inappropriate for reliable training.

3.3.5 L1-norm regularization

In both training and fine-tuning stages, L1-norm regularization was used to restraint the redundant degrees of freedom of $k_\phi^{n_p}$ predicted by the model.

$$\mathcal{L}^{k_\phi}_{\text{L1-norm}} = \frac{1}{4N_{\text{proper}}} \sum_{\text{propers}} \sum_{n_\phi} \left| \hat{k}_\phi^{n_\phi} \right| \tag{14}$$

3.4 Loss functions used in different stages

As demonstrated in the main article, different datasets and loss functions were combined in different training stages. In the pre-training stage, the *optimization dataset* was used, and the overall loss function was given by:

$$\begin{aligned}
\mathcal{L}_{\text{pre-training}} =& \mathcal{L}_{\text{optimization}} \\
=& \lambda^{\sigma}_{\text{MSE}} \mathcal{L}^{\sigma}_{\text{MSE}} + \lambda^{\epsilon}_{\text{MSE}} \mathcal{L}^{\epsilon}_{\text{MSE}} + \lambda^{q}_{\text{MSE}} \mathcal{L}^{q}_{\text{MSE}} + \lambda^{k_\phi}_{\text{MSE}} \mathcal{L}^{k_\phi}_{\text{MSE}} \\
&+ \lambda^{r_0}_{E_{\text{bond}}} \mathcal{L}^{r_0}_{E_{\text{bond}}} + \lambda^{\theta_0}_{E_{\text{angle}}} \mathcal{L}^{\theta_0}_{E_{\text{angle}}} + \lambda^{k_\psi}_{E_{\text{improper}}} \mathcal{L}^{k_\psi}_{E_{\text{improper}}} \\
&+ \lambda^{k_r, k_\theta, k_\psi}_{\text{MAPE}} \mathcal{L}^{k_r, k_\theta, k_\psi}_{\text{MAPE}}.
\end{aligned} \tag{15}$$

Since individual terms in the loss function are able to decrease simultaneously, the value of λs were simply set to 1.0 with satisfying performance and not further optimized.

In the training stage, both *optimization* and *torsion datasets* were used, with the overall loss function given by:

$$\begin{aligned}
\mathcal{L}_{\text{training}} =& \mathcal{L}_{\text{optimization}} + \mathcal{L}_{\text{torsion}} \\
\mathcal{L}_{\text{optimization}} =& \lambda^{\sigma}_{\text{MSE}} \mathcal{L}^{\sigma}_{\text{MSE}} + \lambda^{\epsilon}_{\text{MSE}} \mathcal{L}^{\epsilon}_{\text{MSE}} + \lambda^{q}_{\text{MSE}} \mathcal{L}^{q}_{\text{MSE}} \\
& + \lambda^{r_0}_{E_{\text{bond}}} \mathcal{L}^{r_0}_{E_{\text{bond}}} + \lambda^{\theta_0}_{E_{\text{angle}}} \mathcal{L}^{\theta_0}_{E_{\text{angle}}} + \lambda^{k_\psi}_{E_{\text{improper}}} \mathcal{L}^{k_\psi}_{E_{\text{improper}}} \\
& + \lambda^{k_r, k_\theta, k_\psi}_{\text{MAPE}} \mathcal{L}^{k_r, k_\theta, k_\psi}_{\text{MAPE}} \\
\mathcal{L}_{\text{torsion}} =& \lambda^{k_\phi}_{\text{Bolt_MSE}} \mathcal{L}^{k_\phi}_{\text{Bolt_MSE}} + \lambda^{k_\phi}_{\text{L1-norm}} \mathcal{L}^{k_\phi}_{\text{L1-norm}},
\end{aligned} \tag{16}$$

in which all λs were set to 1.0, the clamp and decay weights in the Boltzmann MSE loss were set to 10.0 and 2.0 respectively. Several alternative values of $\lambda^{k_\phi}_{\text{L1-norm}}$ were tested and the best were chosen according to the resulting validation loss.

In the fine-tuning stage, in addition to the *optimization* and *torsion datasets*, the *off-equilibrium dataset* was also incorporated, with the overall loss function given by:

$$\begin{aligned}
\mathcal{L}_{\text{training}} =& \mathcal{L}_{\text{optimization}} + \mathcal{L}_{\text{torsion}} + \mathcal{L}_{\text{off-equilibrium}} \\
\mathcal{L}_{\text{off-equilibrium}} =& \lambda^{\text{Energy}}_{\text{MSE}} \mathcal{L}^{\text{Energy}}_{\text{MSE}} + \lambda^{\text{Force}}_{\text{MSE}} \mathcal{L}^{\text{Force}}_{\text{MSE}},
\end{aligned} \tag{17}$$

where $\mathcal{L}_{\text{optimization}}$ and $\mathcal{L}_{\text{torsion}}$ were the same as those in the training stage. All λs were set to 1.0, except for $\lambda^{\text{Force}}_{\text{MSE}}$, which was set to 0.1. A few alternative ratios of $\lambda^{\text{Energy}}_{\text{MSE}}$ to $\lambda^{\text{Force}}_{\text{MSE}}$ were tested and the best were chosen according to the resulting validation loss.

All loss functions are differentiable with respect to the force field parameters, allowing gradients to be backpropagated to the model parameters. We implemented gradient back-propagation and batch training using PyTorch,[17] with gradient clipping applied based on

the maximum values of each force field parameter.

$$\frac{\partial \mathcal{L}}{\partial \theta_i} = \frac{\partial \mathcal{L}}{\partial \theta_i} \cdot \frac{\kappa_{\theta_i}}{\max_i(\frac{\partial \mathcal{L}}{\partial \theta_i})} \tag{18}$$

3.5 Force field geometric optimization

In the force field geometric optimization, torsion constraint was applied to the scanned torsion in the *torsion dataset*, such that the scanned torsion angle was fixed during the optimization. We implemented a batched L-BFGS[18] with SHAKE algorithm[19] using PyTorch to automate and accelerate the optimization process.

4 Metrics for force field evaluation

To evaluate the performance of ByteFF models and directly compare various force fields, several evaluation datasets were used, including OpenFFBenchmark dataset and a series of torsion scan datasets (TorsionNet500 and BDTorsion). These evaluation datasets were entirely separate from the training data, ensuring no risk of data leakage.

4.1 Metrics on OpenFFBenchmark dataset

On the OpenFFBenchmark public dataset, three metrics were calculated for evaluating the performance of force fields: root-mean-square deviation (RMSD) of atomic positions, torsion fingerprint deviation (TFD),[20] and relative energy differences ($\Delta\Delta E$) using the compare-force-fields scheme defined by OpenFF team.[21,22] With these metrics, we compared the performance of ByteFF models with GAFF-2.2, OpenFF-2.0.0 and Espaloma-0.3.0. The results of OpenFF-2.0.0 were similar to those in 22, with slight difference mainly arose from the download failure of several conformers (a total of 71,456 conformers were downloaded). Additionally, 142 conformers were removed due to mismatch with their mapped SMILES. Furthermore, OpenFF-2.0.0 failed to label 8 molecules, and Espaloma-0.3.0 failed to label 5 molecules, which were excluded from the evaluation.

4.2 Metrics on torsion scan datasets

On torsion scan datasets, we calculated the root mean squared error (RMSE) and Boltzmann-weighted RMSE of the torsion energy profile on both TorsionNet500[23] and BDTorsion datasets. The MM energy profiles were calculated by minimizing the energy of QM-minimized conformers under MMFFs, with the scanned torsion constrained and other torsion angles restrained by 0.1 kcal/mol/rad, preventing significant conformational changes. In the RMSE calculation, the QM and MM energies were adjusted so that their average values were equal. The Boltzmann-weighted RMSE was calculated as the square root of the Boltzmann MSE

described in Section 3.3.4, with both clamp and decay parameters set to 2 kcal/mol. In the tested torsion scan datasets, conformers with different chirality were split into different molecules. Three molecules in TorsionNet500 were removed due to the mismatch between conformers and mapped SMILES. Additionally, OpenFF-2.0.0 failed to label 32 molecules in BDTorsion in-ring dataset and 7 molecules in BDTorsion non-ring dataset, while Espaloma-0.3.0 failed to label 30 molecules in BDTorsion in-ring dataset and 7 molecules in BDTorsion non-ring dataset, which were excluded from the evaluation.

4.3 Metadynamics Details

Ten drug-like fragments were selected from ligands in the FEP+ benchmark dataset[24] by experts for MetaD simulations. One torsion angle in each fragment were chosen as the collective variable (CV) to impose bias potential. All metadynamics simulations[25,26] were performed at 300 K for 1 ns using the Langevin integrator with a timestep of 1 fs. The biasing potential was added every 40 MD timesteps, with wells of width $\sigma = 0.1^\circ$ and heights $w = 0.2$ kJ/mol. The well-tempered algorithm[27] was used for smooth convergence of the bias potential, with a bias factor $\gamma = 5$. Such parameters were chosen to ensure the biasing potentials converged within the 1 ns simulations. The final biasing potentials $V(\mathbf{s})$ were converted to the free energy surfaces of the biased torsions $F(\mathbf{s})$ via the following expression:

$$F(\mathbf{s}) = -(\frac{\gamma}{\gamma - 1})V(\mathbf{s}). \tag{19}$$

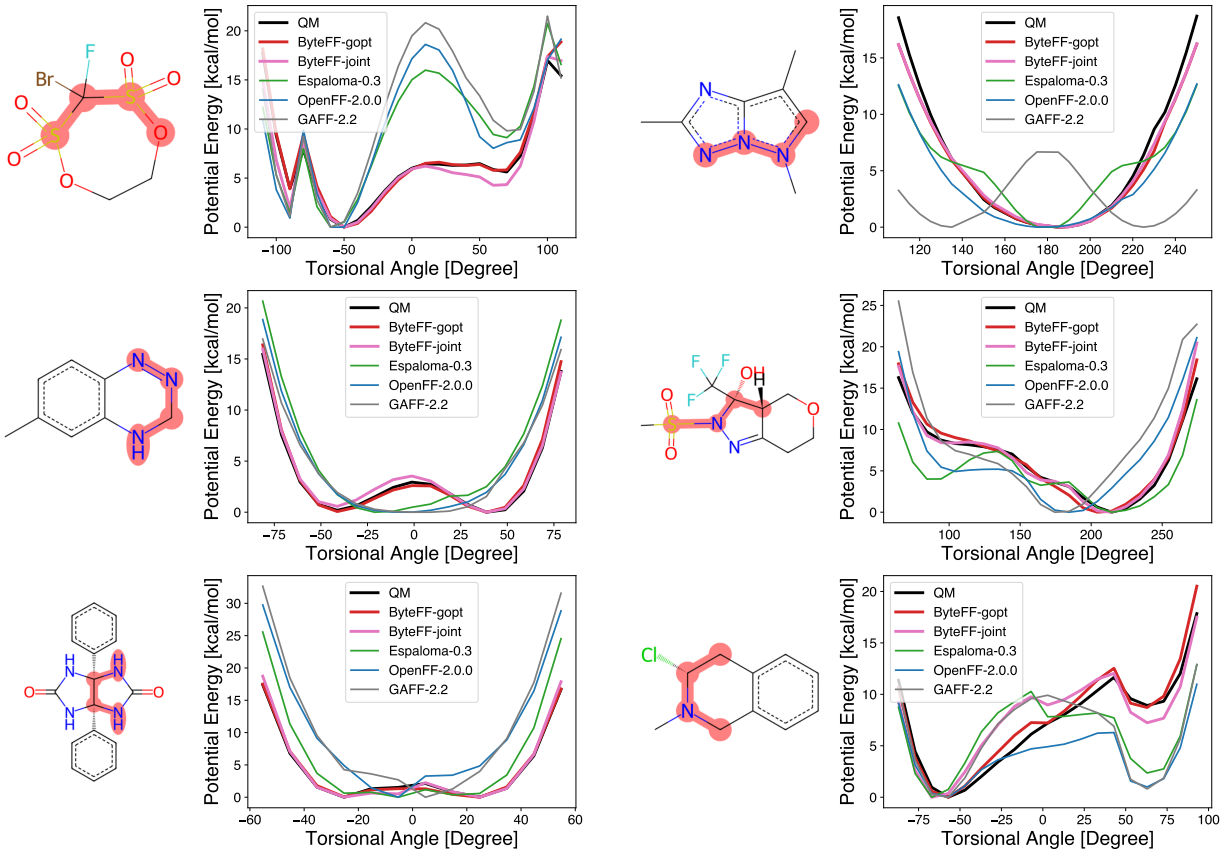

Figure S2: **Additional examples of the in-ring torsion prediction accuracy of various force fields.** As supplementary examples to Figure 4 (a-b), more example molecules with in-ring torsion predictions are shown here, covering various types and in-ring torsions and different elements. The torsional energy profile predicted by various force fields are shown for each example molecule, in comparison with reference from the QM calculation.

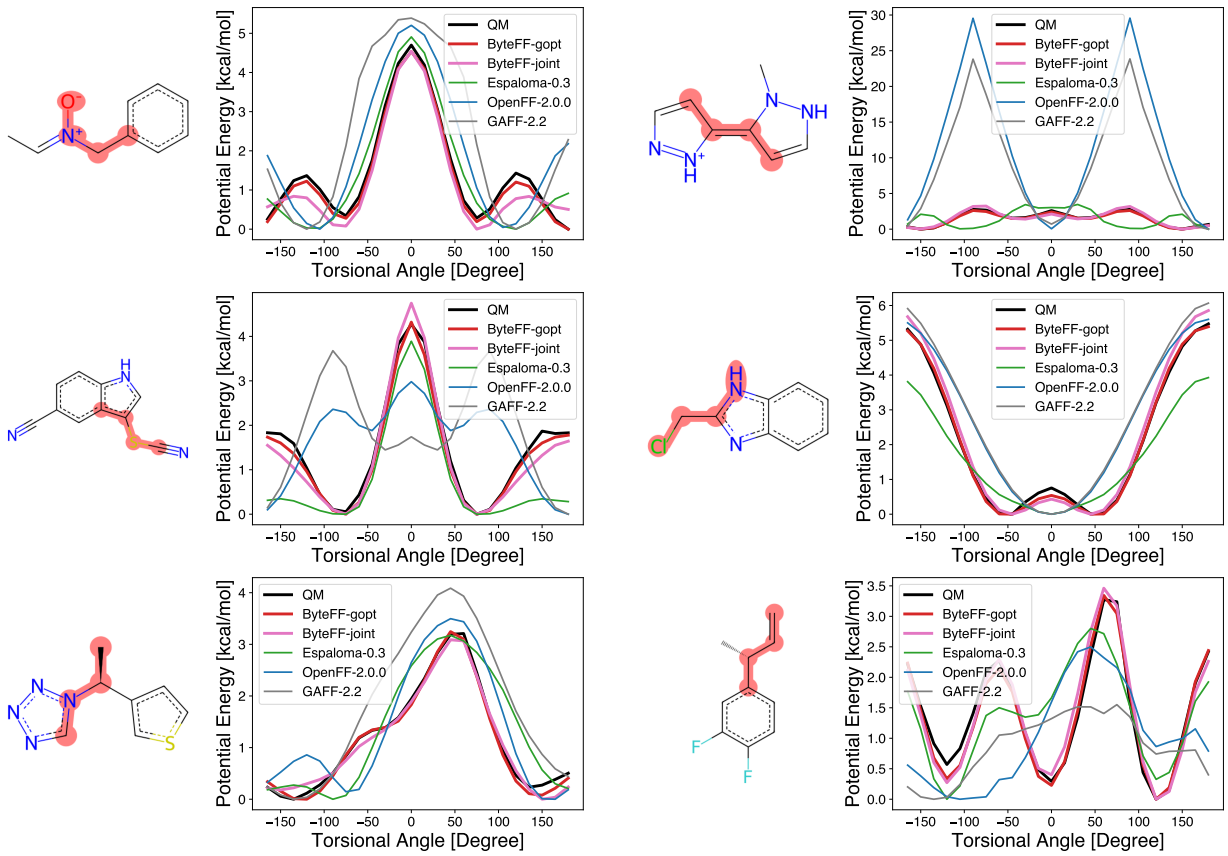

Figure S3: **Additional examples of the non-ring torsion prediction accuracy of various force fields.** As supplementary examples to Figure 4 (c-d), more molecules with non-ring torsion predictions are shown here. For each molecule, the torsional energy profile predicted by various force fields are also shown with reference from the QM calculation.

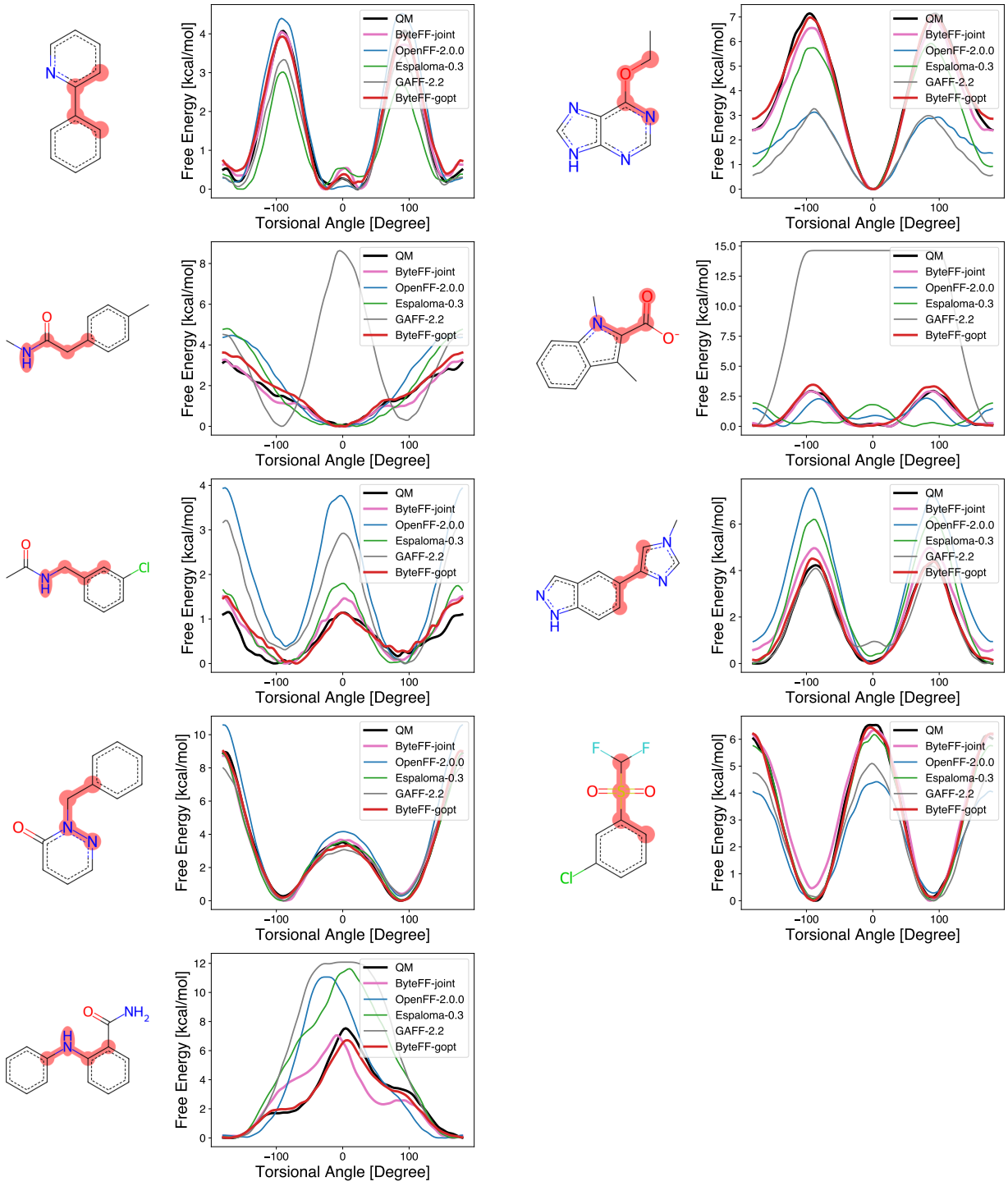

Figure S4: **Additional examples of the torsional free energy surface (FES) prediction accuracy of various force fields.** As supplementary examples to Figure 7, nine additional molecules with torsion FES predictions are shown here. For each molecule, the torsional FES predicted by various force fields are also shown with reference from the QM calculation.

References

(1) Zdrazil, B. et al. The ChEMBL Database in 2023: A Drug Discovery Platform Spanning Multiple Bioactivity Data Types and Time Periods. *Nucleic Acids Research* **2024**, *52*, D1180–D1192, DOI: 10.1093/nar/gkad1004.

(2) Irwin, J. J.; Tang, K. G.; Young, J.; Dandarchuluun, C.; Wong, B. R.; Khurelbaatar, M.; Moroz, Y. S.; Mayfield, J.; Sayle, R. A. ZINC20—a free ultralarge-scale chemical database for ligand discovery. *Journal of chemical information and modeling* **2020**, *60*, 6065–6073.

(3) Ropp, P. J.; Kaminsky, J. C.; Yablonski, S.; Durrant, J. D. Dimorphite-DL: an open-source program for enumerating the ionization states of drug-like small molecules. *Journal of Cheminformatics* **2019**, *11*, 1–8.

(4) Shelley, J. C.; Cholleti, A.; Frye, L. L.; Greenwood, J. R.; Timlin, M. R.; Uchimaya, M. Epik: a software program for pK a prediction and protonation state generation for drug-like molecules. *Journal of computer-aided molecular design* **2007**, *21*, 681–691.

(5) Godbout, N.; Salahub, D. R.; Andzelm, J.; Wimmer, E. Optimization of Gaussian-type basis sets for local spin density functional calculations. Part I. Boron through neon, optimization technique and validation. *Canadian Journal of Chemistry* **1992**, *70*, 560–571, DOI: 10.1139/v92-079.

(6) Epifanovsky, E.; Gilbert, A. T.; Feng, X.; Lee, J.; Mao, Y.; Mardirossian, N.; Pokhilko, P.; White, A. F.; Coons, M. P.; Dempwolff, A. L.; others Software for the frontiers of quantum chemistry: An overview of developments in the Q-Chem 5 package. *The Journal of Chemical Physics* **2021**, *155*, 084801, DOI: 10.1063/5.0055522.

(7) Wu, X.; Sun, Q.; Pu, Z.; Zheng, T.; Ma, W.; Yan, W.; Yu, X.; Wu, Z.; Huo, M.; Li, X.; Ren, W.; Gong, S.; Zhang, Y.; Gao, W. Enhancing GPU-acceleration in the

Python-based Simulations of Chemistry Framework. 2024; https://arxiv.org/abs/2404.09452.

(8) RDKit: Open-source Cheminformatics. DOI: 10.5281/zenodo.10633624.

(9) Wang, L.-P.; Song, C. Geometry optimization made simple with translation and rotation coordinates. *The Journal of Chemical Physics* **2016**, *144*, 214108, DOI: 10.1063/1.4952956.

(10) Michaud-Agrawal, N.; Denning, E. J.; Woolf, T. B.; Beckstein, O. MDAnalysis: A toolkit for the analysis of molecular dynamics simulations. *Journal of Computational Chemistry* **2011**, *32*, 2319–2327, DOI: 10.1002/jcc.21787.

(11) Hussain, M. S.; Zaki, M. J.; Subramanian, D. Global Self-Attention as a Replacement for Graph Convolution. Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. New York, NY, USA, 2022; pp 655–665, DOI: 10.1145/3534678.3539296.

(12) Takaba, K.; Friedman, A.; Cavender, C.; Behara, P.; Pulido, I.; Henry, M.; MacDermott-Opeskin, H.; Iacovella, C.; Nagle, A.; Payne, A.; Shirts, M.; Mobley, D. L.; Chodera, J. D.; Wang, Y. Machine-Learned Molecular Mechanics Force Fields from Large-Scale Quantum Chemical Data. *Chemical Science* **2024**, 12861–12878, DOI: 10.1039/D4SC00690A.

(13) Wang, J.; Wang, W.; Kollman, P. A.; Case, D. A. Automatic Atom Type and Bond Type Perception in Molecular Mechanical Calculations. *Journal of Molecular Graphics and Modelling* **2006**, *25*, 247–260, DOI: 10.1016/j.jmgm.2005.12.005.

(14) Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. Development and Testing of a General Amber Force Field. *Journal of Computational Chemistry* **2004**, *25*, 1157–1174, DOI: 10.1002/jcc.20035.

(15) Dahlgren, M. K.; Schyman, P.; Tirado-Rives, J.; Jorgensen, W. L. Characterization of Biaryl Torsional Energetics and Its Treatment in OPLS All-Atom Force Fields. *Journal of Chemical Information and Modeling* **2013**, *53*, 1191–1199, DOI: 10.1021/ci4001597.

(16) Jorgensen, W. L.; Maxwell, D. S.; Tirado-Rives, J. Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. *Journal of the american chemical society* **1996**, *118*, 11225–11236.

(17) Paszke, A. et al. *Advances in Neural Information Processing Systems 32*; Curran Associates, Inc., 2019; pp 8024–8035.

(18) Liu, D. C.; Nocedal, J. On the limited memory BFGS method for large scale optimization. *Mathematical programming* **1989**, *45*, 503–528.

(19) Ryckaert, J.-P.; Ciccotti, G.; Berendsen, H. J. Numerical integration of the cartesian equations of motion of a system with constraints: molecular dynamics of n-alkanes. *Journal of computational physics* **1977**, *23*, 327–341.

(20) Schulz-Gasch, T.; Schärfer, C.; Guba, W.; Rarey, M. TFD: Torsion Fingerprints As a New Measure To Compare Small Molecule Conformations. *Journal of Chemical Information and Modeling* **2012**, *52*, 1499–1512, DOI: 10.1021/ci2002318.

(21) Lim, VT.; Hahn, DF.; Tresadern, G.; Bayly, CI.; Mobley, DL. Benchmark Assessment of Molecular Geometries and Energies from Small Molecule Force Fields [Version 1; Peer Review: 2 Approved]. *F1000Research* **2020**, *9*, 1–22, DOI: 10.12688/f1000research.27141.1.

(22) D'Amore, L. et al. Collaborative Assessment of Molecular Geometries and Energies from the Open Force Field. *Journal of Chemical Information and Modeling* **2022**, *62*, 6094–6104, DOI: 10.1021/acs.jcim.2c01185.

(23) Rai, B. K.; Sresht, V.; Yang, Q.; Unwalla, R.; Tu, M.; Mathiowetz, A. M.; Bakken, G. A. TorsionNet: A Deep Neural Network to Rapidly Predict Small-Molecule Torsional Energy Profiles with the Accuracy of Quantum Mechanics. *Journal of Chemical Information and Modeling* **2022**, *62*, 785–800, DOI: 10.1021/acs.jcim.1c01346.

(24) Ross, G. A.; Lu, C.; Scarabelli, G.; Albanese, S. K.; Houang, E.; Abel, R.; Harder, E. D.; Wang, L. The maximal and current accuracy of rigorous protein-ligand binding free energy calculations. *Communications Chemistry* **2023**, *6*, 222.

(25) Laio, A.; Parrinello, M. Escaping free-energy minima. *Proceedings of the national academy of sciences* **2002**, *99*, 12562–12566.

(26) Bussi, G.; Branduardi, D. Free-energy calculations with metadynamics: Theory and practice. *Reviews in Computational Chemistry Volume 28* **2015**, 1–49.

(27) Barducci, A.; Bussi, G.; Parrinello, M. Well-tempered metadynamics: a smoothly converging and tunable free-energy method. *Physical review letters* **2008**, *100*, 020603.