JCTC Journal of Chemical Theory and Computation

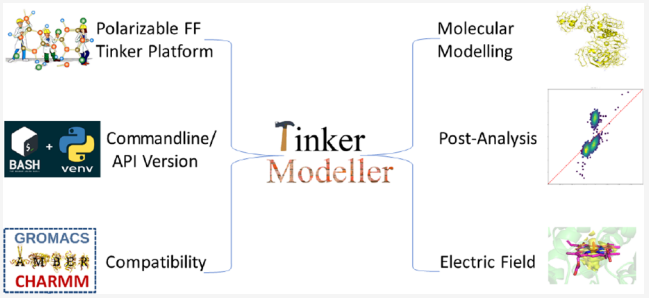
pubs.acs.org/JCTC Article

TinkerModeller: An Efficient Tool for Building Biological Systems in Tinker Simulations

Xujian Wang, Haodong Liu, Yu Li, Jiahuang Li,* and Wan-Lu Li*

Cite This: J. Chem. Theory Comput. 2025, 21, 2712−2722 Read Online

ABSTRACT: Polarizable force fields advance our understanding of electrostatic interactions in molecular systems; however, their widespread application is limited by the complexity of required molecular modeling. We here present TinkerModeller (TKM), a versatile software package designed to streamline the construction of biological systems in the Tinker molecular simulation software. The core functionality of TKM lies in its capacity to generate input files for complex molecular systems and facilitate the conversion from classical to polarizable force fields. With a user-friendly, standalone script, TKM provides an intuitive interface that supports users from molecular modeling through to postanalysis, creating a comprehensive platform for molecular dynamics simulations within Tinker. Furthermore, TKM includes an electric field (EF) postanalysis module, introducing a novel approach that employs charge methods and point charge approximations for efficient internal EF estimation. This module offers a computationally low-demand solution for high-throughput EF estimation. Our work paves the way for broader, more accessible use of polarizable force fields within Tinker and introduces a new method for EF estimation, advancing our capacity to explore electrostatic effects in biological and materials science applications.



1. INTRODUCTION

Molecular dynamics (MD) simulations are essential for understanding the behavior of materials and biomolecules at the atomic scale. MD simulations can utilize a range of force fields, each designed to capture different aspects of the molecular interactions. Among the most commonly used are conventional all-atom force fields (cMD), such as AMBER force field,[1,2] which offer good accuracy and computational efficiency. These force fields are widely applied in studies of protein−ligand interactions,[3,4] enzymatic catalysis,[5−8] and other biomolecular processes. However, cMD approaches have limitations, particularly their inability to account for intramolecular polarization and multipole effects. This can lead to inaccuracies in modeling specific nonbonded interactions that require a more nuanced representation of electrostatics.

To address the limitations of cMD force fields, polarizable force fields have been developed, including the CHARMM Drude force field[9−11] and the AMBER pGM force field,[12−14] both of which account for polarization effects in different ways. Among these, AMOEBA[15−17] (atomic multipole optimized energetics for biomolecular applications) has achieved remarkable accuracy due to its explicit incorporation of dipole and quadrupole moments, allowing for a more refined description of electrostatic interactions in molecular simulations. This makes AMOEBA particularly useful for simulating systems, where detailed electrostatic interactions, such as ion−water interaction,[18] $\pi-\pi$ stacking,[19] and hydrogen bonding,[20] are critical. However, the increased complexity of AMOEBA comes at a cost. Calculating dynamic polarization and multipole interactions requires significantly more computational resources than conventional force fields. The need for iterative calculations to account for induced dipoles and higher-order electrostatic effects results in slower simulations, limiting the widespread adoption of AMOEBA despite its superior accuracy.[21,22]

A further challenge with the broader use of polarizable force fields like AMOEBA lies in the compatibility between different force fields and simulation platforms. Conventional MD engines, such as those used in AMBER,[23] GROMACS,[24] and CHARMM,[25] are optimized for cMD simulations and may not be equipped to handle the more complex integration methods required by polarizable force fields. On the other hand, platforms such as Tinker[26] and Tinker-HP[27] are designed specifically for AMOEBA but are less commonly used for cMD. This limitation can pose challenges for researchers seeking to leverage both the efficiency of cMD and the accuracy of polarizable force fields.[28−31]

Received: October 30, 2024
Revised: February 19, 2025
Accepted: February 20, 2025
Published: February 25, 2025

Another limitation stems from the fundamental differences in topological logic between classical force fields and AMOEBA. Classical force fields like AMBER,[1,2] CHARMM,[32,33] and OPLS[34] rely on generalized atom types and interaction terms that are transferable across a wide range of biomolecules. This allows for easier application and parametrization across various systems without significant reconfiguration. In contrast, AMOEBA uses highly specific atom types, assigning unique types to nearly every distinct chemical environment in a molecule. For instance, while classical force fields may treat an α-C uniformly across different residues, AMOEBA assigns distinct atom types depending on local bonding environments.[16] This increased precision in topology ensures greater accuracy in simulations but complicates the process of building molecular systems.

To address these challenges, we developed TinkerModeller (TKM), a versatile and open-source tool designed to streamline the construction of molecular systems for AMOEBA force field simulations. TKM dramatically simplifies the process of building complex biological systems by offering a fast, intuitive interface that bridges the gap between cMD and polarizable MD approaches. One of the key features is its ability to seamlessly convert classical force fields, enabling transitions from widely used force fields, AMBER into the AMOEBA force field in Tinker format. This functionality streamlines the process, eliminating the need for time-consuming and error-prone manual parametrization. In addition to its force field conversion functionality, TKM is equipped with robust postanalysis tools, including a force-field-independent electric field (EF) analysis module. This feature allows researchers to conduct rapid, accurate EF calculations across different molecular systems, providing critical insights into electrostatic interactions—a key advantage when studying complex molecular behaviors in both enzyme and material systems.[35−41] By integration of these capabilities, TKM offers a solution that combines the high precision of the AMOEBA force field with the flexibility and efficiency of conventional MD platforms. It is designed to significantly reduce the barriers to using polarizable force fields, making advanced molecular simulations more accessible and practical for a broad range of applications in biomolecular and materials science.

2. METHODS

2.1. Electric Field Analysis. With the advancement of vibrational Stark effect spectroscopy,[42] researchers are now able to probe the EF within the active sites of enzymes with greater precision. Complementing these experimental techniques, Bhowmick et al.[43] established a robust theoretical framework for computationally determining the EF at atomic sites. The EF at atom i in the x direction can be defined as follows:

$$E_x^i = \sum_j E_x^{j\to i} = \sum_j \left(E_{x,\mathrm{perm}}^{j\to i} + E_{x,\mathrm{ind}}^{j\to i}\right) \tag{1}$$

This equation consists of two primary components: the permanent and induced EFs. The detailed equation is given by

$$E_x^{j\to i} = -T_x q^j + \sum_{m=y,z} T_{xm}\mu_m^j - \frac{1}{3}\sum_{m=y,z}\sum_{n=y,z} T_{xmn} Q_{mn}^j \tag{2}$$

where

$$T_{xy\ldots} = \frac{1}{4\pi\epsilon_0}\nabla_x\nabla_y\ldots\frac{1}{r_{ij}} \tag{3}$$

The EF is a result of charges (q) and multipoles such as dipoles (μ) and quadrupoles (Q). These components contribute to the overall EF based on their magnitude and spatial distribution, represented by the spatial tensor T.

The EF along a bond between atoms i and j is defined as follows:

$$E_{\mathrm{proj}}^{ij} = \left(\frac{\vec{E}^i + \vec{E}^j}{2}\right)\cdot\vec{u}^{ij} \tag{4}$$

where $\vec{u}^{ij}$ is the unitary vector along the bond direction. However, most EF calculations rely on polarizable force fields like AMOEBA,[22,35,36,43] which require significant computational resources and involve complex postanalysis procedures. To accelerate EF calculations and provide an approximate estimation, we assumed that in enzymatic reactions, the contribution from multipoles is relatively small and that charges predominate in determining the EF. Under this assumption, the EF simplifies to

$$E_x^i \approx \sum_j E_{x,\mathrm{charge}}^{j\to i} = \sum_j -T_x q^j \tag{5}$$

This simplification allows for a more computationally efficient estimation of EFs, sacrificing some precision for speed in scenarios, in which rapid evaluations are necessary.

2.2. Subgraph Isomorphism Algorithm. In the fields of pattern recognition and computer vision, determining whether two graphs (composed of nodes and edges) are (sub)-isomorphic is a fundamental task.[44] While numerous algorithms address this problem, the VF2 algorithm[45,46] stands out for its speed, accuracy, and robustness. Given two graphs, G_1 (N_1, E_1) and G_2 (N_2, E_2), with nodes (N) and edges (E), isomorphism checking involves generating a set of $N_1 \times N_2$ node mappings. However, only a small subset of these mappings represents valid isomorphisms, which are the focus of the analysis.

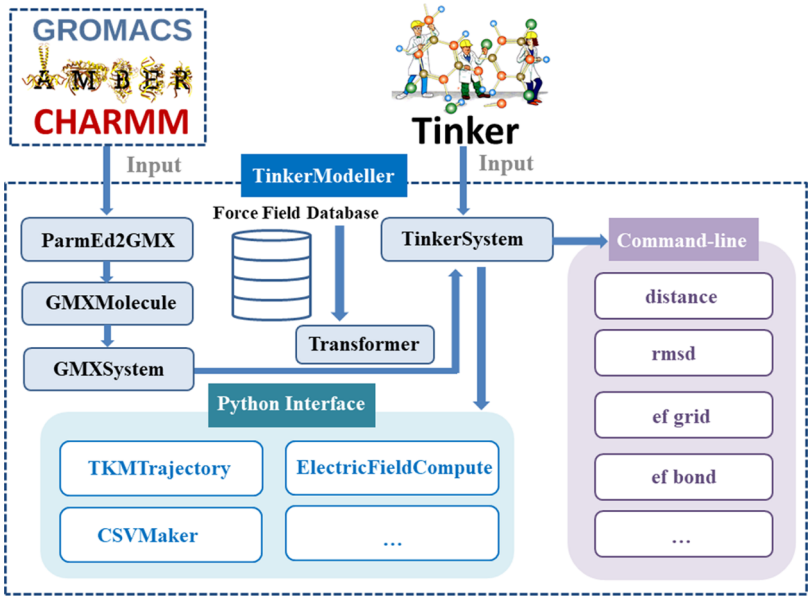
To minimize the computational complexity of exhaustive search, VF2 introduces a state space representation.[45,46] For each state s, the set $M(s)$ represents a partial solution to the matching problem. The algorithm attempts to match subgraphs $G_1(s)$ and $G_2(s)$, recursively updating the state from s to s' as it explores potential matches. Operating as a depth-first search algorithm, VF2 applies a set of predefined feasibility rules when updating states, ensuring the validity of the matches.

$$F(s, n_1, n_2) = F_{\mathrm{syn}}(s, n_1, n_2) \wedge F_{\mathrm{sem}}(s, n_1, n_2) \tag{6}$$

where $n_1 \in N_1$ and $n_2 \in N_2$. Feasibility rules consist of two components: syntactic feasibility, which depends solely on the structure of the graphs, and semantic feasibility, which depends on the attributes of the nodes and edges (the details of the two rules are illustrated in refs 45 and 46). If the selected nodes n_1 and n_2 are feasible according to these rules, they form a pair $p = (n_1, n_2)$, leading to the successor state $s' = s \cup p$. This method is particularly useful in biological systems, where the complexity and variability of residue structures require precise, topology-based identification across large data sets. We implemented the VF2 algorithm in TKM (discussed later) because it efficiently handles the graph isomorphism problem,

https://doi.org/10.1021/acs.jctc.4c01463
J. Chem. Theory Comput. 2025, 21, 2712−2722

Scheme 1. Schematic Workflow of TinkerModeller



allowing us to accurately match atom-level connectivity patterns within residues.

3. SOFTWARE FUNCTIONALITIES AND VALIDATIONS

In this section, we provide an overview of TKM, as well as its functionalities and implementation of its key components. TKM is primarily written in Python,[47] a widely used language in scientific research for its versatility and ease of use. For performance-critical tasks, such as root mean square deviation (RMSD) calculations, some postanalysis modules are implemented in C++ to ensure higher computational efficiency. We offer both a command-line interface with various modeling and analysis modules, as well as a Python package for seamless integration into existing workflows. Despite the use of multiple programming languages, the entire application programming interface (API) is fully integrated into Python, allowing users to easily extend and customize the software with their own Python code.

3.1. General Workflow. The core components of TKM include both GROMACS and Tinker format readers. Due to the inconsistent atom type and residue naming conventions in many Protein Data Bank (PDB) files, we use GROMACS formats (.gro and .top) in combination with the AMBER force fields to standardize and clean data for molecular modeling. To further enhance usability, we integrated ParmEd[48] into our toolset (ParmEd2GMX), as shown in Scheme 1, enabling the conversion of AMBER (.prm and .crd) and CHARMM (.psf and .crd) formats into GROMACS format. This approach allows for a unified interface to handle inputs from multiple software platforms.

All input data from GROMACS are stored in the GMXSystem module. The Transformer module then converts this data into the TinkerSystem using our predefined force field database. During this process, the Transformer assigns atom types by considering atom connectivity, AMBER force field atom types, and residue names to ensure precise mapping to the AMOEBA force field. Our current implementation includes several up-to-date AMOEBA force fields, such as AMOEBABio09,[16,49−53] AMOEBAPro13,[16] AMOEBA-Nuc17,[17] and AMOEBABio18.[15−17] Additionally, new AMOEBA force fields can be easily incorporated into the database, and users can add custom force fields by following the provided database format. Once all data are converted from the GMXSystem, the system can perform various tasks, the primary one being exporting data as a Tinker XYZ (TXYZ) file to transform a conventional MD system into a polarizable MD system. We also offer tools to modify the system, such as merging multiple systems or altering topologies before exporting.

Users can also use the TinkerSystem module to read TXYZ files generated by either Tinker or TKM, as well as trajectory files in .ARC and .DCD formats (the latter powered by MDAnalysis[54,55]) from simulations. With access to trajectories or coordinates along with topologies, users can perform common MD analyses, such as RMSD, for evaluating atomic fluctuations and EF analysis. Additionally, the Python interface, as shown in Scheme 1, allows users to define custom algorithms for specific calculations, showcasing the flexibility of TKM. More than just a modeling tool for Tinker, TKM integrates a comprehensive range of functionalities. Its Python package offers a programmable and flexible system for both building and analysis, while the command-line version provides an efficient and user-friendly approach to achieving these tasks.

3.2. Molecular Modeling with AMOEBA Force Field. TKM is designed to be compatible with input formats from AMBER, CHARMM, and GROMACS. At the core of its functionality is the Transformer module, which allows users to leverage advanced molecular modeling tools, such as CHARMM-GUI[56] and AmberTools,[23] for system construction using classical force fields, and then convert them into the AMOEBA force fields with TXYZ format for further use in Tinker (Scheme 2). These capabilities significantly streamline tasks like solvation, charge equilibration, and periodic box setup, making the overall modeling process more efficient.

3.2.1. Robustness Testing for Molecular Modeling. To rigorously evaluate the robustness of TKM, we created a comprehensive test set for molecular modeling, selecting proteins from Enzyme Commission (EC) categories EC1 to EC6 to ensure a wide range of structural diversity. This selection process yielded a total of 100 protein structures.

https://doi.org/10.1021/acs.jctc.4c01463
J. Chem. Theory Comput. 2025, 21, 2712−2722

Scheme 2. Example of Using the Transformer Module in TKM[a]

```
# Python Interface
from tinkermodellor import TinkerModellor
tkm = TinkerModellor( )
  # Force field option: 1-AMOEBABio18, 2-AMOEBABio09, 3-AMOEBAPro13, 4-AMOEBANuc17
tkm.transform(top='input.top' , crd='input.gro' , xyz='output.xyz ', ff=1)

# Command Line Version
  # This is the usage for the GROMACS input file.
  # Amber format (.prm/.crd) and CHARMM format (.psf/.crd) are also supported.
tkm transform --top input.top --crd input.gro --xyz output.xyz --ff 1
```

[a]For more details, please refer to https://tinkermodellor-tutorial.readthedocs.io/en/latest/Python_Interface.html#transform-module.

Given the inherent variability in PDB file formats, each file was first standardized using pdbfixer, addressing common issues such as missing atoms, incorrect formatting, and chain gaps. Following this initial cleanup, we employed *pdb4amber* in AmberTools23 to filter the files, isolating only the protein components and removing any ligands, solvent molecules, or nonstandard residues. After this filtration step, 32 protein structures were excluded due to the presence of nonstandard residues or covalent ligands, resulting in a final set of 68 protein samples for further testing.

The remaining 68 samples were then parametrized using *pdb2gmx* in GROMACS with the AMBER14 force field.[1] During this process, two samples (PDB IDs: 6SWC and 3RL6) failed due to a lack of appropriate parameters for certain residues, leaving 66 valid samples in GROMACS format for TKM testing. TKM was able to successfully convert all 66 GROMACS-formatted cases into the AMOEBABio18 force fields, generating the corresponding TXYZ files for each protein. For further validation, we also utilized the *pdbxyz* module in Tinker to parametrize the same 68 PDB files generated by *pdb4amber*, all of which were successfully converted using the AMOEBABio18 force field, highlighting the robustness of both tools in processing diverse protein structures.

To further assess the functionality of TKM, we performed energy minimization on the 66 TXYZ files generated by TKM and the 68 TXYZ files generated by Tinker *pdbxyz* using the AMOEBABio18 force fields. Every file produced by TKM was processed successfully; however, one file generated by Tinker *pdbxyz* (PDB ID: 6SWC) failed to minimize the structure due to missing force field parameters. This failure was consistent with the initial parametrization issues observed during the GROMACS preparation, confirming that the error was not caused by TKM but inherent to the lack of force field coverage for this specific protein structure.

3.2.2. Protonation State and Terminal Atom Determination. The setting of the protonation state is very important because different protonation states may change the function of the enzyme and lead to huge changes in the simulation results.[57,58] In the AMOEBA force field, nine amino acids adopt multiple protonation states, with glutamic acid, aspartic acid, and lysine each having two states and histidine having three. After modeling with Tinker and TKM, we quantified the occurrences of these protonation states in each protein. As illustrated in Figure 1A, the two tools yielded identical protonation state assignments across all test cases, with detailed results uploaded to the public database. This consistency demonstrates the reliability of TKM in accurately determining the protonation states. To further explore the

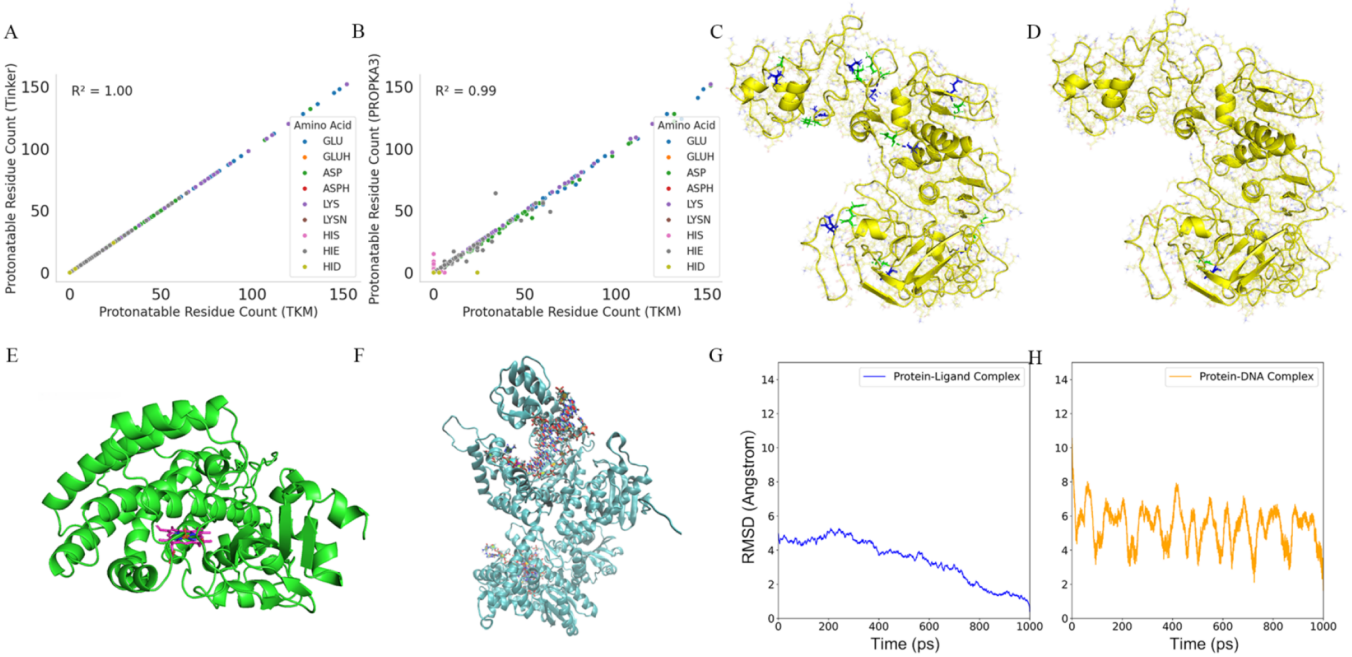


Figure 1. Molecular modeling and validation using the AMOEBA force field with TKM. (A) and (B) Comparison of protonation state assignments between Tinker and TKM, and between PROPKA3 and TKM, respectively, by assessing the number of occurrence of the nine protonation states: GLU/GLUH and ASP/ASPH for glutamate and aspartate in their deprotonated (negative) and protonated (neutral) forms, LYS/LYSN for lysine in its protonated (positive) and neutral forms, and HIS/HIE/HID for histidine. In histidine, HID (protonated at the first nitrogen in the imidazole ring) and HIE (protonated at the second nitrogen) are neutral, while HIS is protonated at both sites. (C) and (D) Terminal atom identification by TKM and Tinker, respectively. (E) and (F) Protein−ligand complex (PDB ID: 6GII) and a protein−DNA complex (PDB ID: 4DS4) constructed by TKM. (G) and (H) RMSD curves of both complex systems after simulation.

https://doi.org/10.1021/acs.jctc.4c01463
J. Chem. Theory Comput. 2025, 21, 2712−2722

flexibility of protonation state assignment, we employed PROPKA3[59] to predict protonation states at a physiological pH of 7. Interestingly, this analysis revealed discrepancies between the results predicted by PROPKA3 and those obtained from TKM (Figure 1B), highlighting the differences in the algorithms and approaches used for protonation prediction in these tools. Notably, AMBER allows precise modification of amino acid protonation states based on PROPKA3 outputs. Since TKM is compatible with the AMBER format, users can leverage this compatibility to accurately edit protonation states. Additionally, CHARMM-GUI and GROMACS also provide various tools for adjusting protonation states. TKM can integrate results from these different modeling tools, facilitating seamless format and force field conversions. This significantly enhances the convenience of protonation state control within the AMOEBA force field.

In addition to protonation states, the correct assignment of terminal atoms, such as nitrogen and hydrogen at the N-terminus and oxygen and carbon at the C-terminus, is critical for biomolecular simulations. These atoms are specifically parametrized in the AMOEBA force field, and accurate identification is necessary to ensure proper force field application. To address this, we utilized a nonamer structure (PDB ID: 2C8G) as a test case for terminal atom assignment. TKM employs a depth-first search algorithm that uses atomic connectivity to identify terminal atoms in each protein chain and assigns the corresponding terminal atom types, ensuring accurate representation (Figure 1C). This approach is particularly useful when working with complex systems involving multiple chains, as terminal atoms at each chain need to be treated independently. In comparison, the default functionality of Tinker recognizes only the first and last residues of the PDB file as terminals, which can lead to errors in multimers or more complex assemblies (Figure 1D). In such cases, TKM ensures that each chain, even in multimeric proteins like dimers or octamers, has its terminal atoms properly assigned, making it more robust and versatile for handling diverse biomolecular systems.

3.2.3. Extending AMOEBA for Organic Molecules and Complex System Integration. The AMOEBA force field supports a limited range of biomolecules, primarily focusing on nucleic acids and proteins. Although this coverage includes many essential biomolecular systems, organic molecules play a critical role in numerous biological processes, such as cellular signaling[21] and altering biochemical function.[60] In these areas, organic molecules require special attention, and the need for accurate modeling is paramount. However, the complexity of the AMOEBA force field distinguishes it from general force fields like GAFF[61] or CGenFF,[62] commonly used for organic molecules in classical MD simulations. Intricate parametrization in AMOEBA requires that each atom be meticulously characterized to ensure precise modeling, making it challenging to balance transferability and accuracy when working with a broader range of molecular systems.

Given this complexity, users typically need to fit parameters manually for organic ligands manually. Fortunately, two reliable methods exist for generating force field parameters: using Psi4[63] combined with the poledit module in Tinker or Poltype2[64] software. Both approaches produce a force field parameter file and a TXYZ file for Tinker simulations. However, Tinker simulations rely on a single coordinate file (TXYZ) and a single force field file (.prm), requiring careful manual editing to ensure compatibility between organic molecules and the broader biomolecular system. This manual editing presents significant challenges, especially when complex biological systems are composed of multiple organic and biomolecular components.

By integration of TKM, users can easily merge these distinct systems without the concern of atom-type mismatches between different molecular entities. The functionality of TKM extends beyond simple file editing; it includes features that evaluate and adjust the coordination distances between atom pairs, ensuring accurate spatial arrangement of atoms. Additionally, when a merged molecule overlaps with a trivial molecule, such as a solvent molecule, TKM will automatically remove the unnecessary entity, optimizing the integrity of the system. By taking these additional steps, TKM effectively prevents MD simulation errors, such as simulation crashes, which can occur due to poorly merged or misconfigured systems.

As demonstrated in Figure 1E, we modeled the P450 protein (PDB ID: 6GII) with its heme group submerged in water. The protein and solvents were parametrized using the AMOEBA-Bio18 force field, and the multipole and polarizability parameters of the ligands were derived using Psi4 in combination with the poledit module. The remaining parameters were based on previous studies.[65] These two systems were then seamlessly merged by using TKM (Scheme 3). For more complex systems, such as a protein−DNA complex (Figure 1F), the structure was obtained from the Protein Data Bank (PDB ID: 4DS4). In this case, the protein was parametrized by the AMOEBABio18 force field, while the DNA used the AMOEBANuc17 force field. Both systems were successfully constructed and merged using TKM. To further demonstrate TKM's capability in modeling complex biological systems, we selected other two systems: an RNA−protein complex (Figure S1A) and a protein−DNA−RNA terpolymer (Figure S1B). Unlike protein modeling, the modeling of DNA−RNA systems requires additional considerations, including terminal atoms and specific protonation states. This successful molecular modeling result confirms that TKM possesses a broad range of functionalities for modeling biological macromolecules. Notably, we also observed that AMOEBA carbohydrate and lipid force fields have been developed, further expanding its applicability to a broader range of biomolecules.[66,67] Although our current modeling does not include carbohydrates and lipids, we are developing more advanced topology-processing modules to enhance the modeling of these complex biomolecular systems and streamline the overall workflow.

Scheme 3. Example of Using the Merge Module in TKM[a]

```
# Python Interface
  # Merge input2.xyz into input1.xyz, and merge ff2.prm into ff1.prm
  # Force field (.prm) is optional
tkm.merge (tk1='input1.xyz', tk2='input2.xyz', ffl='ff1.prm', ff2='ff2.prm', \
          tinker_xyz='output.xyz', ffout='ffout.prm',)

# Command Line Version
tkm merge --tkl input1.xyz --tk2 input2.xyz --xyz output.xyz --ff1 input1.prm \
          --input2.prm --ffout output.prm
```

[a]For more details, please refer to https://tinkermodellor-tutorial.readthedocs.io/en/latest/Python_Interface.htmlxxxhashxxxmerge-module.

https://doi.org/10.1021/acs.jctc.4c01463
J. Chem. Theory Comput. 2025, 21, 2712−2722

To further evaluate the efficiency and robustness of TKM, we performed energy minimization and MD simulations for the first two systems we built (see the Supporting Information for details). As the simulation progressed, the RMSD values of the protein−ligand and protein−DNA systems remained stable with minimal fluctuations, indicating consistent behavior without anomalies (Figure 1G,H). These results confirm that the system constructed by TKM performs as expected and is fully compatible with Tinker, demonstrating a successful setup using the AMOEBA force field.

3.3. Postanalysis. *3.3.1. Postanalysis Tools in Tinker-Modeller.* Postanalysis is a critical step following the completion of a simulation, as it allows researchers to quantitatively evaluate key metrics. TKM provides several fundamental tools for postsimulation analysis, including RMSD calculations to assess atomic fluctuations, as well as distance and angle measurements. Although the current postanalysis capabilities are somewhat limited, the Python API of TKM enhances flexibility by offering the TKMTrajectory module, which stores trajectory coordinates. This enables users to define custom postanalysis functions, giving them the ability to calculate specific metrics relevant to their research objectives (Scheme 4).

Scheme 4. Example of Using Self-Defined Postanalysis Function in TinkerModeller

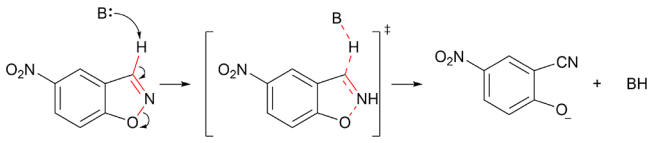
```
# Python Interface
def self_defined_analysis(crd, box):
    # crd: 3D coordinates of atoms, shape=(nframes, natoms,3)
    # box: box information,shape=(nframes,3,3)
    pass

from tinkermodellor import TKMTrajectory
traj _object = TKMTrajectory( )
  # Determine the topology from TXYZ file, then load the trajectory (.arc)
traj_object.read_from_tinker ('input.xyz')
traj_object.read_from_traj ('input.arc')
crd = traj_object.AtomCrds
box = traj_object.Box

self_defined_analysis (crd, box)
```

3.3.2. Force-Field-Independent Electric Field Calculation. In the context of postanalysis, understanding the EF distribution within a molecular system is essential for interpreting the mechanistic details of enzymatic reactions. EFs have been shown to play a crucial role in modulating reaction rates by influencing the rate-limiting steps of various biochemical processes.[35−38] For example, in Kemp eliminases, the alignment of the EF with the direction of electron flow can lower the activation energy, thereby accelerating the reaction.[22,43] Here, we propose a promising and fast method for EF calculations, which has the potential for future high-throughput EF screening. To demonstrate its effectiveness, we use a well-studied case, Kemp elimination, as a model system.

In our approach, we acknowledge that while the AMOEBA force field provides high accuracy for EF calculations (eqs 1 and 2), it also comes with significant computational cost. As an alternative, we suggest that EFs can be approximated using only atomic charges without explicitly including multipoles (eq 5). This eliminates the need for time-consuming dynamic updates of multipole moments, significantly accelerating EF calculations. Theoretically, since the contribution of multipole moments to the EF diminishes much more rapidly with distance compared to charge contributions (eqs 2 and 3), the resulting simplification introduces minimal loss of accuracy in EF calculations. Additionally, our analysis reveals that the charge distribution is broader and exhibits larger absolute values compared with multipole contributions (Figure S2), leading to a greater impact on EF calculations (eq 2). The combination of the charge distribution's greater magnitude and the rapid decay of multipole moments may support the dominant role of charge in EF calculations.

To validate our assumptions, we conducted a 100 ps simulation using the AMOEBABio18 force field, generating 1,000 frames. Using these frames, we calculated the EF projected onto three reactive bonds (C−H, C−N, and N−O) directly involved in the chemical reaction (Scheme 5). The

Scheme 5. Kemp Elimination Reaction Mechanism Catalyzed by Kemp Eliminases



projection of the EF follows the definition in eq 4. First, we used the ELECTRIC[68] to perform EF analysis without any simplifications, using its results as the baseline. We then used TKM to calculate the EF by using only atomic charges within the AMOEBA force field. Our results showed that the point charge approximation achieved a correlation coefficient of 0.77 with the baseline (Figure S3), indicating that this assumption is reasonable.

After demonstrating the feasibility of our method, we further explored ways to improve the accuracy and speed of this method. We introduced three mainstream atomic partial charge determination methods implemented in Open Babel:[69] the Electronegativity Equalization Method (EEM),[70] Extended Charge Equilibration Method (EQEq),[71] and Charge Transfer with Polarization Current Equalization (QTPIE).[72] Unlike AMOEBA's charge, which is highly decoupled from the multipole moments,[15−17] these charge methods inherently account for charge transfer effects, mutual polarization effects, and other electrostatic effects into consideration. We believe that this implicit consideration can speed up the calculation and also improve the calculation accuracy. We then conducted a series of tests using the three charge methods. Figure 2A−C shows that all methods follow a similar trend, where the projected EF onto a specific chemical bond fluctuates significantly in the first 10 frames before stabilizing as the system approaches equilibrium. However, EEM and EQeq exhibit noticeable deviations from AMOEBABio18, with mean absolute errors (MAEs) of 21.93 and 26.17 MV/cm, respectively. In contrast, QTPIE demonstrates better agreement with AMOEBA, showing smaller discrepancies (Figure 2F) and achieving a higher correlation coefficient (0.89), compared to the EF calculated using AMOEBA's charges (0.77). These discrepancies likely arise because the AMOEBA force field dynamically updates charges and includes multipole effects, whereas these methods estimate partial charges based on a single frame without fully capturing charge redistribution due to protein conformational changes. To improve accuracy, we enhanced the TKM by incorporating an on-the-fly charge-based EF calculation feature, which

https://doi.org/10.1021/acs.jctc.4c01463
J. Chem. Theory Comput. 2025, 21, 2712−2722

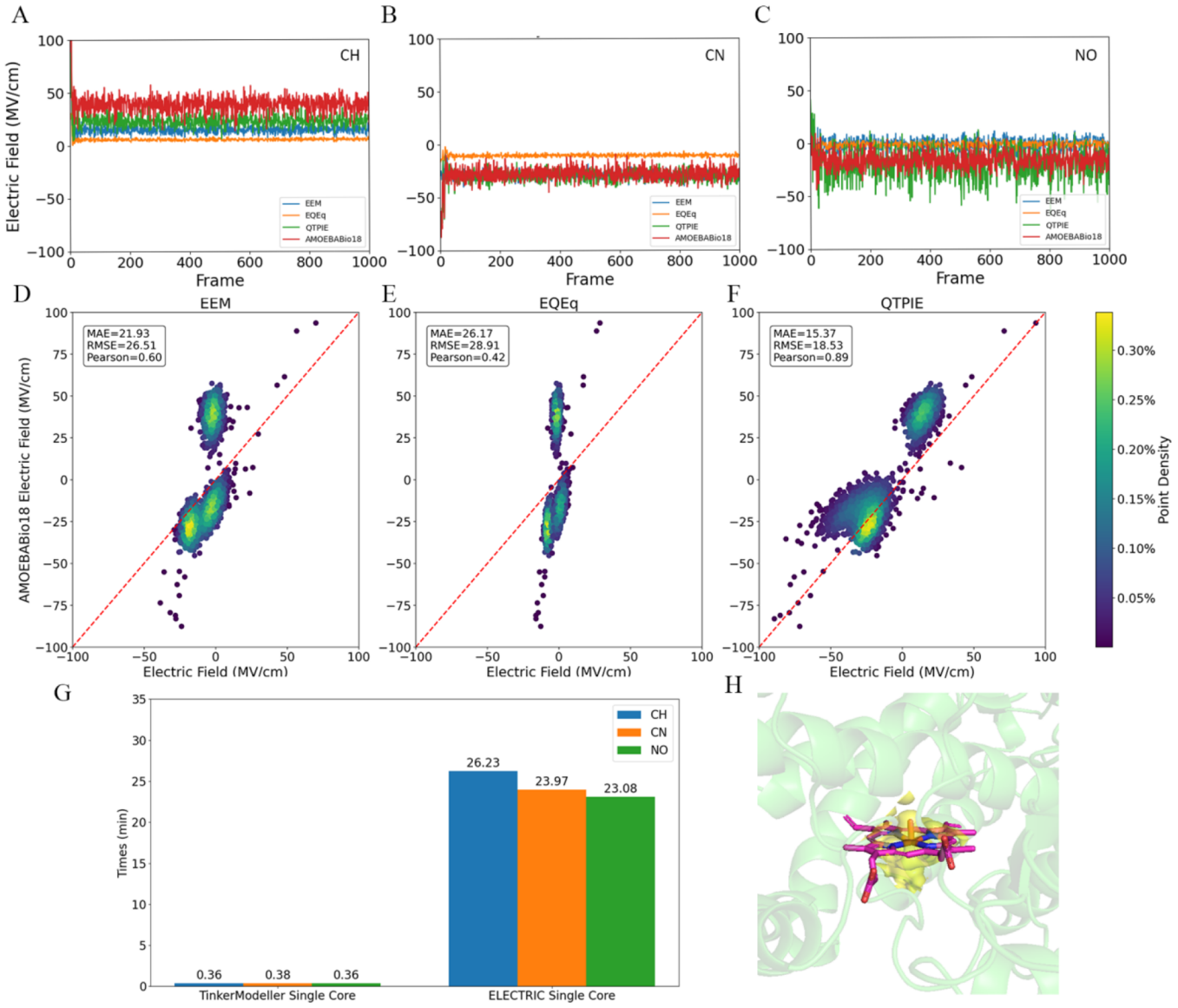


Figure 2. Electric field calculation and validation. (A−C) Illustration of the electric field calculated by ELECTRIC and TKM, projected onto three C−H, C−N, and N−O reactive bonds, respectively. (D−F) Electric field distribution calculated using three different charge methods, with the AMOEBA force field serving as the reference. Dots in different colors representing the density distribution. (G) Comparison of the computational performance of TKM against Tinker with the ELECTRIC module for electric field calculation. (H) Electric field magnitude plot visualized using the grid method.

dynamically updates atomic charges at each frame. This refinement led to notable improvements: the correlation coefficient for EEM increased from 0.60 to 0.84, for EQeq from 0.42 to 0.87, and for QTPIE from 0.89 to 0.94 (see Figure S4). After applying the on-the-fly method, all methods achieved correlation coefficients above 0.80, surpassing the 0.77 correlation observed for AMOEBA's charge model. These results indicate that while these methods rely on point charge models, incorporating dynamic charge updates significantly improves computational efficiency while maintaining high accuracy.

To assess the computational performance of TKM in EF calculations, we compared it with ELECTRIC by using the AMOEBABio18 force field. On a single core, TKM completed the calculations in under 1 min, whereas ELECTRIC took approximately 25 min (Figure 2G). This significant performance improvement—by up to 2 orders of magnitude—demonstrates that TKM achieves a considerable increase in computational efficiency. This acceleration is accomplished with only a minor reduction in accuracy, making it a promising approach for high-throughput EF calculations in large-scale simulations.

We also investigated an alternative approach by focusing on the characterization of EFs in specific chemical regions, rather than limiting the analysis to projections along individual chemical bonds, as is traditionally done.[38−41] This method accounts for the fact that electrostatic effects extend beyond bond-specific interactions, influencing the broader environment including the active center of enzymatic systems. Relying solely on bond projections can overlook critical charge redistributions occurring at nearby sites within the active region. To demonstrate this, we used the CYP450 enzyme (PDB ID: 6GII) as a model system and calculated the EFs surrounding the heme group. The QTPIE method was

https://doi.org/10.1021/acs.jctc.4c01463
J. Chem. Theory Comput. 2025, 21, 2712−2722

employed to assign charges, and the resulting EF was visualized by generating an isosurface around the central iron atom (Figure 2H). This approach provides a more comprehensive representation of EF distribution within the active site, providing new insight into how the strength and configuration of the environmental EF influence catalytic processes, shifting the focus away from immediate chemical bonds to broader field effects.

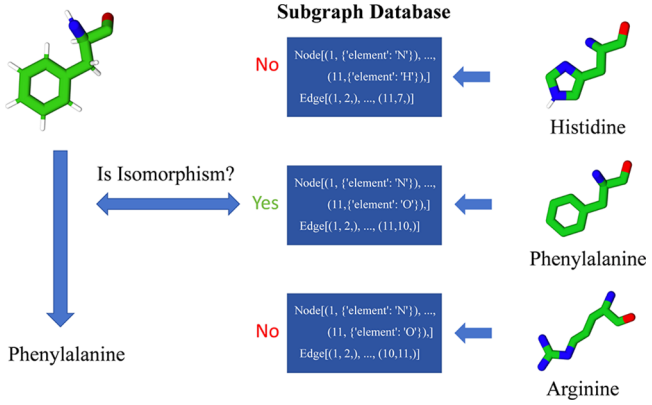
Through the solid verification of computational accuracy and efficiency, we believe that this approach holds promise as an efficient method for EF estimation in future applications requiring rapid and computationally low-cost assessments.

3.4. Structural Visualization and Residue Information Restoration. Visualization of simulation outputs is a critical aspect of postanalysis, particularly for examining protein conformations and structural dynamics. While the TXYZ format retains essential information such as atomic coordinates and connectivity, it lacks the detailed residue and chain information needed for visualization in tools like PyMOL.[73] In contrast, the PDB format is widely supported and offers comprehensive structural details. However, the standard conversion of TXYZ files to PDB format using the *xyzpdb* module of Tinker often leads to incomplete files that omit crucial residue and chain information.

TKM addresses this shortcoming by enhancing the converted PDB files with enriched residue names and chain indices, which are indispensable for advanced analyses, such as residue fluctuation and residue pairwise distance correlation. This improvement is achieved through the integration of the VF2 algorithm. Using a depth-first search strategy, TKM segments the system based on atomic connectivity, identifies protein chains, and further subdivides each chain into residue sets. A subgraph matching technique is then employed to restore accurate residue and chain information by comparing the identified segments with a predefined database (Schemes 6 and 7). The resulting enriched PDB files can subsequently be used for further postanalysis and visualization tasks.

To validate the accuracy of the transformation process, we benchmarked the results against reference files. Both TKM and Tinker were used to generate the TXYZ system, which was then converted into a PDB format using TKM. The conversion process was successful for all generated files, with no errors observed in residue or chain assignments. Notably, TKM accurately handled residues with varying protonation states, as well as cysteine types. Finally, the converted PDB file was visualized using 3D Protein Imaging[74] and Pymol,[73] confirming the precision of the restored structures (Figure 3).

Scheme 6. Schematic Workflow for Restoring Residue Information Using a Subgraph Isomorphism Algorithm[a]



[a]Residue determination is achieved by dividing the residue into nodes and edge information, which act as constraint conditions. These constraints are then used in the subgraph isomorphism algorithm (SIA) to match the subgraph within a predefined database, ultimately determining the residue type.

Scheme 7. Example of Using the tk2pdb Module in TinkerModeller[a]

```
# Python Interface
  # depth: the maxium depth-first search depth for the atom connectivity
  # style:   the input style of the input file, 1-TXYZ generated by TKM, \
  #          2-TXYZ generated by Tinker
tkm.tk2pdb(tk='input.xyz' ,pdb='output.pdb' ,depth=1000,style=1)

# Command Line Version
tkm tk2pdb --tk input.xyz --pdb output.pdb --depth 1000 --style 1
```

[a]For more details, please refer to https://tinkermodellor-tutorial.readthedocs.io/en/latest/Python_Interface.html#tk2pdb-module.

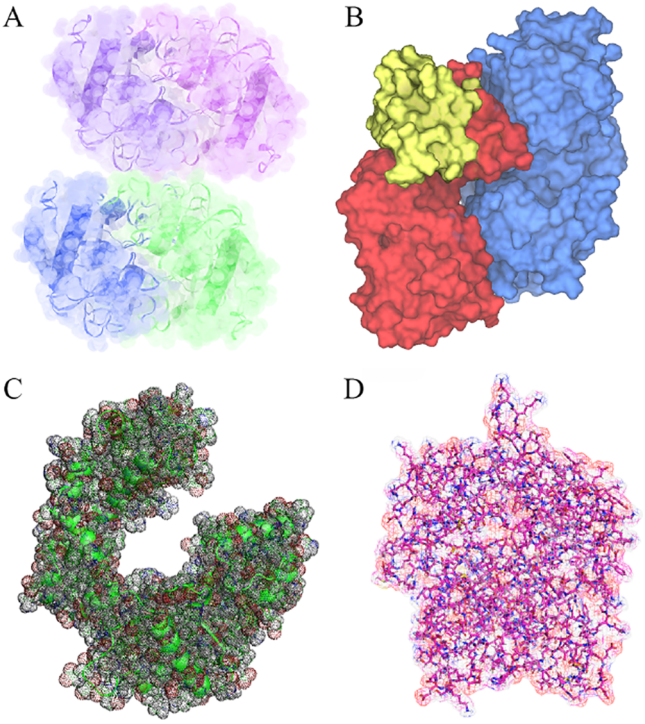


Figure 3. Structure visualization using PDB files generated by TinkerModeller: (A) tetrameric protein (PDB ID: 6K8P); (B) trimeric protein (PDB ID: 5FO1); and (C, D) proteins (PDB IDs: 1ZQL and 6I5M, respectively) visualized using PyMOL.

4. CONCLUSIONS

The AMOEBA force field, as a key component of the next generation of force fields, holds great promise for advancing molecular simulations due to its ability to incorporate polarization effects. However, its adoption has been hindered by the complexities associated with molecular modeling and postanalysis. To address these challenges, we developed TKM, a comprehensive molecular modeling software designed to integrate seamlessly with the Tinker simulation program. TKM not only facilitates the construction of AMOEBA systems but

https://doi.org/10.1021/acs.jctc.4c01463
J. Chem. Theory Comput. 2025, 21, 2712−2722

also provides conversion tools to transform classical force field systems into AMOEBA-compatible formats, bridging the gap between traditional and polarizable force fields.

Through rigorous diversity tests, TKM has demonstrated robustness and efficiency in both molecular modeling and postsimulation analyses. A key innovation introduced by this tool is a fast, simplified method for estimating EFs that perform well compared to the more computationally intensive AMOEBA calculations. This advancement holds significant potential for accelerating high-throughput EF calculations. Additionally, TKM introduces a novel approach to EF analysis by calculating the EF across chemical regions using grid-based visualization, offering deeper insights into the influence of EFs in reactive environments. Overall, TKM is a versatile, user-friendly tool that streamlines the modeling of AMOEBA force field systems within the Tinker software suite. Its modular design and ease of use are expected to promote broader adoption of AMOEBA in future studies, enabling researchers to better explore complex molecular systems, where polarization and multipole interactions play critical roles.

Although TKM has demonstrated strong performance in modeling and postprocessing, there is still room for development. The AMOEBA force field has been extended to include carbohydrates and lipids, broadening its applicability to more biomolecular systems. To fully leverage these advancements, we need to develop more sophisticated topology-processing modules for carbohydrates and lipids to facilitate the modeling of complex biomolecular structures and streamline the overall workflow. Future updates will focus on integrating the corresponding force fields and supporting modeling tools. Additionally, some advanced postanalysis methods, such as Velocity Autocorrelation Function calculations and Protein Conformation Cluster Analysis, have yet to be implemented. Expanding TKM to incorporate these capabilities is a key objective. Overall, TKM will remain an active, long-term development and maintenance project with a continued commitment to enhancing its functionality to support a broader range of biomolecular simulations.

■ ASSOCIATED CONTENT

Data Availability Statement

The source code of TinkerModeller can be accessed through https://github.com/WanluLigroupUCSD/TinkerModellor. The tutorial for TinkerModeller can be found in https://tinkermodellor-tutorial.readthedocs.io/en/latest/index.html. All the testing data are available through https://zenodo.org/records/13885736.

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jctc.4c01463.

Calculation workflow and computational details; electric field analysis specifics with electric field grid visualization; and justification of charge-based electric field analysis and the test data set with a link to the uploaded test file (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Jiahuang Li − *School of Biopharmacy, China Pharmaceutical University, Nanjing 211198, China; Changzhou High-Tech Research Institute, Nanjing University, Changzhou 213164, China;* orcid.org/0000-0003-3513-4159; Email: lijiah@cpu.edu.cn

Wan-Lu Li − *Aiiso Yufeng Li Family Department of Chemical and Nano Engineering and Program of Materials Science and Engineering, University of California San Diego, La Jolla, California 92093, United States;* orcid.org/0000-0003-0098-0670; Email: wal019@ucsd.edu

Authors

Xujian Wang − *Aiiso Yufeng Li Family Department of Chemical and Nano Engineering, University of California San Diego, La Jolla, California 92093, United States; School of Biopharmacy, China Pharmaceutical University, Nanjing 211198, China;* orcid.org/0009-0004-0146-9991

Haodong Liu − *School of Biopharmacy, China Pharmaceutical University, Nanjing 211198, China;* orcid.org/0009-0007-6832-5388

Yu Li − *Aiiso Yufeng Li Family Department of Chemical and Nano Engineering, University of California San Diego, La Jolla, California 92093, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jctc.4c01463

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work used the computational resources from the Expanse supercomputer at the San Diego Supercomputer Center (SDSC) through allocations of CHM230035 and CHE230113. X.W. and J.L. are also grateful for the financial support of China National Innovation and Entrepreneurship Training Program for Undergraduate (No. 202410316011Z) and Changzhou Science and Technology Bureau (No. 2022169).

■ REFERENCES

(1) Maier, J. A.; Martinez, C.; Kasavajhala, K.; Wickstrom, L.; Hauser, K. E.; Simmerling, C. ff14SB: Improving the Accuracy of Protein Side Chain and Backbone Parameters from ff99SB. *J. Chem. Theory Comput.* **2015**, *11*, 3696−3713.

(2) Tian, C.; Kasavajhala, K.; Belfon, K. A. A.; Raguette, L.; Huang, H.; Migues, A. N.; Bickel, J.; Wang, Y.; Pincay, J.; Wu, Q.; Simmerling, C. ff19SB: Amino-Acid-Specific Protein Backbone Parameters Trained against Quantum Mechanics Energy Surfaces in Solution. *J. Chem. Theory Comput.* **2020**, *16*, 528−552.

(3) He, X.; Liu, S.; Lee, T.-S.; Ji, B.; Man, V. H.; York, D. M.; Wang, J. Fast, Accurate, and Reliable Protocols for Routine Calculations of Protein−Ligand Binding Affinities in Drug Design Projects Using AMBER GPU-TI with ff14SB/GAFF. *ACS Omega* **2020**, *5*, 4611−4619.

(4) Shim, J.; Alexander, D.; MacKerell, J. Computational Ligand-Based Rational Design: Role of Conformational Sampling and Force Fields in Model Development. *MedChemComm* **2011**, *2*, 356−370.

(5) Ramos, M. J.; Fernandes, P. A. Computational Enzymatic Catalysis. *Acc. Chem. Res.* **2008**, *41*, 689−698.

(6) Moin, S. T.; Hofer, T. S.; Sattar, R.; Randolf, B. R.; Rode, B. M. Molecular Dynamics Simulation of Mammalian 15S-Lipoxygenase with AMBER Force Field. *Eur. Biophys. J.* **2011**, *40*, 715−726.

(7) Wang, J.; Xu, Y.; Wang, X.; Li, J.; Hua, Z. Mechanism of Mutation-Induced Effects on the Catalytic Function of TEV Protease: A Molecular Dynamics Study. *Molecules* **2024**, *29*, 1071.

(8) Wang, X.; Liu, H.; Wang, J.; Chang, L.; Cai, J.; Wei, Z.; Pan, J.; Gu, X.; Li, W.-L.; Li, J. Enzyme Tunnel Dynamics and Catalytic

Mechanism of Norcoclaurine Synthase: Insights from a Combined LiGaMD and DFT Study. *J. Phys. Chem. B* **2024**, *128*, 9385−9395.
(9) Lemkul, J. A.; Huang, J.; Roux, B.; Alexander, D.; MacKerell, J. An Empirical Polarizable Force Field Based on the Classical Drude Oscillator Model: Development History and Recent Applications. *Chem. Rev.* **2016**, *116*, 4983−5013.
(10) Lopes, P. E. M.; Huang, J.; Shim, J.; Luo, Y.; Li, H.; Roux, B.; Alexander, D.; MacKerell, J. Force Field for Peptides and Proteins Based on the Classical Drude Oscillator. *J. Chem. Theory Comput.* **2013**, *9*, 5430−5449.
(11) Aytenfisu, A. H.; Yang, M.; Alexander, D.; MacKerell, J. CHARMM Drude Polarizable Force Field for Glycosidic Linkages Involving Pyranoses and Furanoses. *J. Chem. Theory Comput.* **2018**, *14*, 3132−3143.
(12) Wei, H.; Qi, R.; Wang, J.; Cieplak, P.; Duan, Y.; Luo, R. Efficient Formulation of Polarizable Gaussian Multipole Electrostatics for Biomolecular Simulations. *J. Chem. Phys.* **2020**, *153*, 114116.
(13) Wei, H.; Cieplak, P.; Duan, Y.; Luo, R. Stress Tensor and Constant Pressure Simulation for Polarizable Gaussian Multipole Model. *J. Chem. Phys.* **2022**, *156*, 114114.
(14) Duan, Y.; Niu, T.; Wang, J.; Cieplak, P.; Luo, R. PCMRESP: A Method for Polarizable Force Field Parameter Development and Transferability of the Polarizable Gaussian Multipole Models Across Multiple Solvents. *J. Chem. Theory Comput.* **2024**, *20*, 2820−2829.
(15) Corrigan, R. A.; Qi, G.; Thiel, A. C.; Lynn, J. R.; Walker, B. D.; Casavant, T. L.; Lagardere, L.; Piquemal, J.-P.; Ponder, J. W.; Ren, P.; Schnieders, M. J. Implicit Solvents for the Polarizable Atomic Multipole AMOEBA Force Field. *J. Chem. Theory Comput.* **2021**, *17*, 2323−2341.
(16) Shi, Y.; Xia, Z.; Zhang, J.; Best, R.; Ponder, J. W.; Ren, P. Polarizable Atomic Multipole-Based AMOEBA Force Field for Proteins. *J. Chem. Theory Comput.* **2013**, *9*, 4046−4063.
(17) Zhang, C.; Lu, C.; Jing, Z.; Wu, C.; Piquemal, J.-P.; Ponder, J. W.; Ren, P. AMOEBA Polarizable Atomic Multipole Force Field for Nucleic Acids. *J. Chem. Theory Comput.* **2018**, *14*, 2084−2108.
(18) Mao, Y.; Demerdash, O.; Head-Gordon, M.; Head-Gordon, T. Assessing Ion−Water Interactions in the AMOEBA Force Field Using Energy Decomposition Analysis of Electronic Structure Calculations. *J. Chem. Theory Comput.* **2016**, *12*, 5422−5437.
(19) Zhang, C.; Bell, D.; Harger, M.; Ren, P. Polarizable Multipole-Based Force Field for Aromatic Molecules and Nucleobases. *J. Chem. Theory Comput.* **2017**, *13*, 666−678.
(20) Lin, Y.-C.; Ren, P.; Webb, L. J. AMOEBA Force Field Predicts Accurate Hydrogen Bond Counts of Nitriles in SNase by Revealing Water−Protein Interaction in Vibrational Absorption Frequencies. *J. Phys. Chem. B* **2023**, *127*, 6100−6111.
(21) Blazhynska, M.; Lagardère, L.; Liu, C.; Adjoua, O.; Ren, P.; Piquemal, J.-P. Water−Glycan Interactions Drive the SARS-CoV-2 Spike Dynamics: Insights into Glycan-Gate Control and Camouflage Mechanisms. *Chem. Sci.* **2024**, *15*, 14177−14187.
(22) Zheng, Y.; Welborn, V. V. Tuning the Catalytic Activity of Synthetic Enzyme KE15 with DNA. *J. Phys. Chem. B* **2022**, *126*, 1234−1245.
(23) Case, D. et al. *Amber 2024*; University of California: San Francisco, 2024.
(24) Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High Performance Molecular Simulations through Multi-Level Parallelism from Laptops to Supercomputers. *SoftwareX* **2015**, *1−2*, 19−25.
(25) Brooks, B. R.; et al. CHARMM: The Biomolecular Simulation Program. *J. Comput. Chem.* **2009**, *30*, 1545−1615.
(26) Rackers, J. A.; Wang, Z.; Lu, C.; Laury, M. L.; Lagardère, L.; Schnieders, M. J.; Piquemal, J.-P.; Ren, P.; Ponder, J. W. Tinker 8: Software Tools for Molecular Design. *J. Chem. Theory Comput.* **2018**, *14*, 5273−5289.
(27) Adjoua, O.; Lagardère, L.; Jolly, L.-H.; Durocher, A.; Very, T.; Dupays, I.; Wang, Z.; Inizan, T. J.; Célerse, F.; Ren, P.; Ponder, J. W.; Piquemal, J.-P. Tinker-HP: Accelerating Molecular Dynamics Simulations of Large Complex Systems with Advanced Point Dipole Polarizable Force Fields Using GPUs and Multi-GPU Systems. *J. Chem. Theory Comput.* **2021**, *17*, 2034−2053.
(28) Kamenik, A. S.; Handle, P. H.; Hofer, F.; Kahler, U.; Kraml, J.; Liedl, K. R. Polarizable and non-polarizable force fields: Protein folding, unfolding, and misfolding. *J. Chem. Phys.* **2020**, *153*, 185102.
(29) Plé, T.; Mauger, N.; Adjoua, O.; Inizan, T. J.; Lagardère, L.; Huppert, S.; Piquemal, J.-P. Routine Molecular Dynamics Simulations Including Nuclear Quantum Effects: From Force Fields to Machine Learning Potentials. *J. Chem. Theory Comput.* **2023**, *19*, 1432−1445.
(30) Lagardère, L.; Maurin, L.; Adjoua, O.; Hage, K. E.; Monmarché, P.; Piquemal, J.-P.; Hénin, J. Lambda-ABF: Simplified, Portable, Accurate, and Cost-Effective Alchemical Free-Energy Computation. *J. Chem. Theory Comput.* **2024**, *20*, 4481−4498.
(31) Wang, Y.; Inizan, T. J.; Liu, C.; Piquemal, J.-P.; Ren, P. Incorporating Neural Networks into the AMOEBA Polarizable Force Field. *J. Phys. Chem. B* **2024**, *128*, 2381−2388.
(32) MacKerell, A. D., Jr; Banavali, N.; Foloppe, N. Development and Current Status of the CHARMM Force Field for Nucleic Acids. *Biopolymers* **2000**, *56*, 257−265.
(33) Huang, J.; Rauscher, S.; Nawrocki, G.; Ran, T.; Feig, M.; de Groot, B. L.; Grubmüller, H.; MacKerell, A. D., Jr. CHARMM36m: An Improved Force Field for Folded and Intrinsically Disordered Proteins. *Nat. Methods* **2017**, *14*, 71−73.
(34) Robertson, M. J.; Tirado-Rives, J.; Jorgensen, W. L. Improved Peptide and Protein Torsional Energetics with the OPLS-AA Force Field. *J. Chem. Theory Comput.* **2015**, *11*, 3499−3509.
(35) Li, W.-L.; Head-Gordon, T. Catalytic Principles from Natural Enzymes and Translational Design Strategies for Synthetic Catalysts. *ACS Cent. Sci.* **2021**, *7*, 72−80.
(36) Welborn, V. V.; Archer, W. R.; Schulz, M. D. Characterizing Ion-Polymer Interactions in Aqueous Environment with Electric Fields. *J. Chem. Inf. Model.* **2022**, *63*, 1587−1596.
(37) Chaturvedi, S. S.; Bím, D.; Christov, C. Z.; Alexandrova, A. N. From Random to Rational: Improving Enzyme Design through Electric Fields, Second Coordination Sphere Interactions, and Conformational Dynamics. *Chem. Sci.* **2023**, *14*, 10997−11011.
(38) Hennefarth, M. R.; Alexandrova, A. N. Direct Look at the Electric Field in Ketosteroid Isomerase and Its Variants. *ACS Catal.* **2020**, *10*, 9917−9926.
(39) Li, W.-L.; Hao, H.; Head-Gordon, T. Optimizing the Solvent Reorganization Free Energy by Metal Substitution for Nanocage Catalysis. *ACS Catal.* **2022**, *12*, 3782−3788.
(40) Sebastiani, F.; Bender, T. A.; Pezzotti, S.; Li, W.-L.; Schwaab, G.; Bergman, R. G.; Raymond, K. N.; Toste, F. D.; Head-Gordon, T.; Havenith, M. An Isolated Water Droplet in the Aqueous Solution of a Supramolecular Tetrahedral Cage. *Proc. Natl. Acad. Sci. U. S. A.* **2020**, *117*, 32954−32961.
(41) Welborn, V. V.; Li, W.-L.; Head-Gordon, T. Interplay of Water and a Supramolecular Capsule for Catalysis of Reductive Elimination Reaction from Gold. *Nat. Commun.* **2020**, *11*, 415.
(42) Fried, S. D.; Boxer, S. G. Measuring Electric Fields and Noncovalent Interactions Using the Vibrational Stark Effect. *Acc. Chem. Res.* **2015**, *48*, 998−1006.
(43) Bhowmick, A.; Sharma, S. C.; Head-Gordon, T. The Importance of the Scaffold for de Novo Enzymes: A Case Study with Kemp Eliminase. *J. Am. Chem. Soc.* **2017**, *139*, 5793−5800.
(44) Conte, D.; Foggia, P.; Sansone, C.; Vento, M. *Applied Graph Theory in Computer Vision and Pattern Recognition*; Kandel, A.; Bunke, H.; Last, M., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2007; pp 85−135.
(45) Foggia, P.; Sansone, C.; Vento, M. *An Improved Algorithm for Matching Large Graphs*; Proc. 3rd IAPR-TC-15 Int. Workshop Graph-based Representation, 2001.
(46) Cordella, L. P.; Foggia, P.; Sansone, C.; Vento, M. A (Sub)Graph Isomorphism Algorithm for Matching Large Graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1367−1372.
(47) van Rossum, G. *Python Programming Language*; USENIX Annu. Tech. Conf.: Santa Clara, CA, 2007; p 36.

https://doi.org/10.1021/acs.jctc.4c01463
J. Chem. Theory Comput. 2025, 21, 2712−2722

(48) Shirts, M. R.; Klein, C.; Swails, J. M.; Yin, J.; Gilson, M. K.; Mobley, D. L.; Case, D. A.; Zhong, E. D. Lessons Learned from Comparing Molecular Dynamics Engines on the SAMPL5 Dataset. *J. Comput. Aided Mol. Des.* **2017**, *31*, 147−161.
(49) Grossfield, A.; Ren, P.; Ponder, J. W. Ion Solvation Thermodynamics from Simulation with a Polarizable Force Field. *J. Am. Chem. Soc.* **2003**, *125*, 15671−15682.
(50) Ren, P.; Ponder, J. W. Polarizable Atomic Multipole Water Model for Molecular Mechanics Simulation. *J. Phys. Chem. B* **2003**, *107*, 5933−5947.
(51) Wu, J. C.; Piquemal, J.-P.; Chaudret, R.; Reinhardt, P.; Ren, P. Polarizable Molecular Dynamics Simulation of Zn(II) in Water Using the AMOEBA Force Field. *J. Chem. Theory Comput.* **2010**, *6*, 2059−2070.
(52) Wang, Z. *Monovalent Ion Parameters*. Ph.D. thesis, Department of Chemistry, Washington University in St. Louis, 2018.
(53) Ren, P.; Wu, C.; Ponder, J. W. Polarizable Atomic Multipole-Based Molecular Mechanics for Organic Molecules. *J. Chem. Theory Comput.* **2011**, *7*, 3143−3161.
(54) Michaud-Agrawal, N.; Denning, E. J.; Woolf, T. B.; Beckstein, O. MDAnalysis: A toolkit for the analysis of molecular dynamics simulations. *J. Comput. Chem.* **2011**, *32*, 2319−2327.
(55) Gowers, R. J.; Linke, M.; Barnoud, J.; Reddy, T. J. E.; Melo, M. N.; Seyler, S. L.; Dotson, D. L.; Domański, J.; Buchoux, S.; Kenney, I. M.; Beckstein, O. *MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations*; Los Alamos National Laboratory, 2016; pp 102−109.
(56) Jo, S.; Kim, T.; Iyer, V. G.; Im, W. CHARMM-GUI: A Web-Based Graphical User Interface for CHARMM. *J. Comput. Chem.* **2008**, *29*, 1859−1865.
(57) Ben-Shimon, A.; Shalev, D. E.; Niv, M. Y. Protonation States in Molecular Dynamics Simulations of Peptide Folding and Binding. *Curr. Pharm. Des.* **2013**, *19*, 4173−4181.
(58) Varma, S.; Chiu, S.-W.; Jakobsson, E. The Influence of Amino Acid Protonation States on Molecular Dynamics Simulations of the Bacterial Porin OmpF. *Biophys. J.* **2006**, *90*, 112−123.
(59) Olsson, M. H. M.; Søndergaard, C. R.; Rostkowski, M.; Jensen, J. H. PROPKA3: Consistent Treatment of Internal and Surface Residues in Empirical pKa Predictions. *J. Chem. Theory Comput.* **2011**, *7*, 525−537.
(60) Welborn, V. V. Beyond Structural Analysis of Molecular Enzyme-Inhibitor Interactions. *Electron. Struct.* **2022**, *4*, No. 014006.
(61) Wang, J.; Wolf, R. M.; Caldwell, J. W.; Kollman, P. A.; Case, D. A. Development and Testing of a General AMBER Force Field. *J. Comput. Chem.* **2004**, *25*, 1157−1174.
(62) Vanommeslaeghe, K.; Hatcher, E.; Acharya, C.; Kundu, S.; Zhong, S.; Shim, J.; Darian, E.; Guvench, O.; Lopes, P.; Vorobyov, I., Jr.; MacKerell, A. D. CHARMM General Force Field: A Force Field for Drug-like Molecules Compatible with the CHARMM All-Atom Additive Biological Force Fields. *J. Comput. Chem.* **2010**, *31*, 671−690.
(63) Smith, D. G. A.; et al. Psi4 1.4: Open-Source Software for High-Throughput Quantum Chemistry. *J. Chem. Phys.* **2020**, *152*, 184108.
(64) Walker, B.; Liu, C.; Wait, E.; Ren, P. Automation of AMOEBA Polarizable Force Field for Small Molecules: Poltype 2. *J. Comput. Chem.* **2022**, *43*, 1530−1542.
(65) Shahrokh, K.; Orendt, A.; Yost, G. S.; Cheatham, T. E., III Quantum Mechanically Derived AMBER-Compatible Heme Parameters for Various States of the Cytochrome P450 Catalytic Cycle. *J. Comput. Chem.* **2012**, *33*, 119−133.
(66) Newman, L. A.; Patton, M. G.; Rodriguez, B. A.; Sumner, E. W.; Welborn, V. V. Polarizable AMOEBA force field predicts thin and dense hydration layer around monosaccharides. *Chem. Commun.* **2024**, *60*, 14431−14434.
(67) Jing, Z.; Rackers, J. A.; Pratt, L. R.; Liu, C.; Rempe, S. B.; Ren, P. Thermodynamics of ion binding and occupancy in potassium channels. *Chem. Sci.* **2021**, *12*, 8290−8930.
(68) Nash, J.; Barnes, T.; Welborn, V. V. ELECTRIC: Electric Fields Leveraged from Multipole Expansion Calculations in Tinker Rapid Interface Code. *J. Open Source Softw.* **2020**, *5*, 2576.
(69) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An Open Chemical Toolbox. *J. Cheminf.* **2011**, *3*, 33.
(70) Vařeková, R. S.; Jiroušková, Z.; Vaněk, J.; Suchomel, Š.; Koča, J. Electronegativity Equalization Method: Parameterization and Validation for Large Sets of Organic, Organohalogene and Organometal Molecule. *Int. J. Mol. Sci.* **2007**, *8*, 572−582.
(71) Wilmer, C. E.; Kim, K. C.; Snurr, R. Q. An Extended Charge Equilibration Method. *J. Phys. Chem. Lett.* **2012**, *3*, 2506−2511.
(72) Chen, J.; Martínez, T. J. QTPIE: Charge transfer with polarization current equalization. A fluctuating charge model with correct asymptotics. *Chem. Phys. Lett.* **2007**, *438*, 315−320.
(73) Schrödinger, L.; DeLano, W. *PyMOL*. http://www.pymol.org/pymol.
(74) Tomasello, G.; Armenia, I.; Molla, G. The Protein Imager: A Full-Featured Online Molecular Viewer Interface with Server-Side HQ-Rendering Capabilities. *Bioinformatics* **2020**, *36*, 2909−2911.

https://doi.org/10.1021/acs.jctc.4c01463
J. Chem. Theory Comput. 2025, 21, 2712−2722