BIOMD: ALL-ATOM GENERATIVE MODEL FOR BIOMOLECULAR DYNAMICS SIMULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Molecular dynamics (MD) simulations are essential tools in computational chemistry and drug discovery, offering crucial insights into dynamic molecular behavior. However, their utility is significantly limited by substantial computational costs, which severely restrict accessible timescales for many biologically relevant processes. Despite the encouraging performance of existing machine learning (ML) methods, they struggle to generate extended biomolecular system trajectories, primarily due to the lack of MD datasets and the large computational demands of modeling long historical trajectories. Here, we introduce BioMD, the first all-atom generative model to simulate long-timescale protein-ligand dynamics using a hierarchical framework of forecasting and interpolation. We demonstrate the effectiveness and versatility of BioMD on the DD-13M (ligand unbinding) and MISATO datasets. For both datasets, BioMD generates highly realistic conformations, showing high physical plausibility and low reconstruction errors. Besides, BioMD successfully generates ligand unbinding paths for 97.1% of the protein-ligand systems within ten attempts, demonstrating its ability to explore critical unbinding pathways. Collectively, these results establish BioMD as a tool for simulating complex biomolecular processes, offering broad applicability for computational chemistry and drug discovery.

1 INTRODUCTION

Molecular dynamics (MD) simulations have emerged as an indispensable tool in computational chemistry and drug discovery, offering insights into the dynamic behavior of biomolecular systems. Through numerical integration of Newton's equations of motion, MD simulations directly produce atomic trajectories that reveal the time evolution of molecular structures (Hollingsworth & Dror, 2018). These trajectories enable the exploration of conformational ensembles, optimization of small molecule structures, and identification of potential binding sites, significantly accelerating the design and development of novel therapeutics (Karplus & McCammon, 2002).

Despite their utility, traditional MD simulations face substantial computational limitations. The core bottleneck lies in the intensive calculation of non-bonded forces, particularly van der Waals and electrostatic interactions, which scale quadratically with the number of atoms (Dror et al., 2012; Adcock & McCammon, 2006). Furthermore, accurately resolving high-frequency atomic vibrations necessitates extremely small time steps (on the order of femtoseconds), severely limiting the accessible simulation timescales (Shaw et al., 2010; 2009). Exploring biologically relevant processes, which often span microseconds to milliseconds, remains computationally intensive, restricting the practical application of atomistic MD to obtain trajectories.

Recently, machine learning (ML) methods have emerged as computational alternatives to molecular dynamics (MD) simulations. Key advances include models for generating protein conformation ensembles (Lewis et al., 2025) and neural network potentials trained on quantum mechanical data (Wang et al., 2024a). For biomolecular systems, AlphaFold 3 (Abramson et al., 2024) has demonstrated promising accuracy in predicting protein–ligand interactions. Despite these achievements, generating full MD trajectories for complex protein–ligand systems using ML remains a major challenge. Existing approaches tend to fall into two categories: (i) methods that can generate protein conformation ensembles but cannot produce time-resolved trajectories Jing et al. (2024a); Wang et al. (2024b), or (ii) methods that attempt trajectory modeling but struggle to capture protein–ligand interactions. For

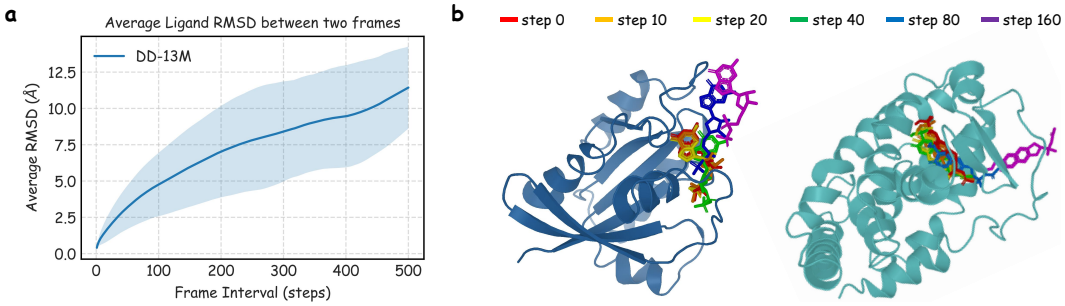

Figure 1. Average Ligand RMSD between two frames. (a) Line plot showing that the average ligand RMSD between two frames in the same trajectory increases with the frame interval. (b) Examples of ligand unbinding trajectories at time steps 0, 10, 20, 40, 80, and 160.

example, NeuralMD (Liu et al., 2024) treats protein atoms as static and only models ligand dynamics, while MDGen (Jing et al., 2024b) is specifically designed for peptides and proteins and does not handle small-molecule ligands. This limitation arises from both the complexity of protein-ligand energy landscapes and the scarcity of high-quality trajectory data for training generative models.

To address these limitations, we propose BioMD, a hierarchical framework for generating all-atom biomolecular trajectories. Building upon the insight that short-timescale conformational changes exhibit little conformational change (**Figure 1**), BioMD decomposes long trajectory generation into two synergistic stages: forecasting of large-step conformations, followed by interpolation to refine intermediate steps. This strategy reduces sequence length by decoupling long-term evolution from local dynamics and helps manage the error accumulation problem for generating long trajectories. Crucially, BioMD unifies forecasting and interpolation within a conditional flow matching model, where we use the "noising-as-masking" methods following Diffusion Forcing (Chen et al., 2024) to our time-scale transformer. We apply independent noise to each frame, which enables flexible conditioning on partial trajectory segments, and we implement different tasks simply by using different masking schedules. Inspired by the success of AlphaFold 3, BioMD generates all-atom trajectories using a velocity network that adapts its core transformer architecture, while employing an SE(3)-equivariant graph transformer to encode the initial conformation as conditional embeddings.

To evaluate the effectiveness of BioMD, we conducted experiments on two datasets: MISATO (Siebenmorgen et al., 2024) and DD-13M (Li et al., 2025). Our results show that BioMD generates highly realistic conformations with promising physical stability, evidenced by low energy and reconstruction errors across both benchmarks. On the MISATO dataset, which focuses on ligand dynamics within the binding pocket, our model accurately captures the system's conformational flexibility, outperforming existing methods. For the more challenging task of ligand unbinding on the DD-13M dataset, BioMD successfully generates complete unbinding paths for up to 97.1% of the protein-ligand systems, demonstrating a robust ability to explore critical and long-timescale biomolecular pathways. Collectively, these results establish BioMD as a powerful and efficient tool for simulating complex biomolecular processes, offering broad applicability for computational chemistry and drug discovery.

2 Related Works

Conformational Ensemble Generation. One major line of research uses ML to generate a biomolecule's conformational ensemble by modeling the equilibrium distribution of its dynamic structures. Early efforts like AlphaFold2 Jumper et al. (2021) produce a set of diverse conformations primarily through multiple sequence alignment (MSA) subsampling and masking techniques Stein & Mchaourab (2022); del Alamo et al. (2022); Wayment-Steele et al. (2024). More advanced approaches now directly learn the conformational distribution from large-scale MD datasets using flow-based Noé et al. (2019); Jing et al. (2024a) or diffusion-based Wang et al. (2024b); Jing et al. (2023); Lu et al. (2024); Zheng et al. (2024); Lu et al. (2025) generative models. Models such as BioEmu Lewis et al. (2025) can effectively generate diverse and physically plausible conformations, providing a powerful alternative to extensive MD sampling to understand a conformational space. However, these

methods are fundamentally time-agnostic; they can sample what conformations are possible but lack the temporal information to show the kinetic pathways between them.

Trajectory Learning for MD Simulation. To capture these kinetic pathways, a complementary research direction aims to generate full, time-ordered trajectories. Approaches like EquiJump Costa et al. (2024) learn to sample future states based solely on the current conformation. To capture higher-order dependencies between the frames, MDGen Jing et al. (2024b) models the joint distribution of entire trajectories via masked frame modeling. CONFROVER Shen et al. (2025) models these dependencies auto-regressively by conditioning each frame on its entire history through a causal transformer. While powerful, these methods are often specialized for protein-only dynamics. Conversely, methods that model protein-ligand interactions often introduce other simplifications. For instance, NeuralMD Liu et al. (2024) treats the protein receptor as static, which limits the scope of accessible dynamics.

3 PRELIMINARIES

Notations. A complex $\mathcal{C}$ is composed of a protein $\mathcal{P}$ and a ligand ℓ. The trajectory of a complex contains $T+1$ frames of coordinates, denoted as $\mathbf{X}_T = \{\mathbf{x}_0, \mathbf{x}_1, \cdots \mathbf{x}_T\} \in \mathbb{R}^{(T+1)\times N\times 3}$, where $\mathbf{x}_t = [\mathbf{x}_t^{\mathcal{P}}, \mathbf{x}_t^{\ell}] \in \mathbb{R}^{N\times 3}$ represents the concatenation of protein coordinates $\mathbf{x}_t^{\mathcal{P}}$ and ligand coordinates $\mathbf{x}_t^{\ell}$ at time-step t, and N is the number of atoms in the complex. The complex trajectory prediction task is defined as generating subsequent conformations (coordinates) of a complex trajectory given its initial conformation (i.e., the first frame).

Molecular dynamics. Molecular dynamics (MD) simulates the time evolution of a particle system under classical mechanics. It leverages numerical schemes such as Verlet integration (Verlet, 1967) or Langevin dynamics to generate trajectories approximating the Boltzmann distribution. In the simplest deterministic case with no friction or noise, each particle i evolves according to $\mathrm{d}x_i = \frac{p_i}{m_i}\,\mathrm{d}t, \mathrm{d}p_i = -\nabla_{x_i} E(x)\,\mathrm{d}t$, where p_i and m_i are the momentum and mass, and $E(x)$ is the potential energy function. Metadynamics (Laio & Parrinello, 2002; Barducci et al., 2011; Li et al., 2025) extends MD by introducing a history-dependent bias potential $V(s,t)$, constructed over collective variables $s(x)$ as $V(s,t) = \sum_{t'<t} w \exp\Big(-\frac{\|s(x(t))-s(x(t'))\|^2}{2\sigma^2}\Big)$, where Gaussians of height w and width σ are periodically added to discourage revisiting explored states. This bias fills free-energy wells and enhances sampling of rare events and transition pathways beyond the reach of standard MD.

Flow matching based models. Flow matching (FM) (Lipman et al., 2023) is an efficient and simulation-free method for training continuous normalizing flows (CNFs), a class of generative models based on ordinary differential equations (ODEs). In Euclidean space, CNFs define a transformation $\phi_\tau(\cdot) : \mathbb{R}^{N\times 3} \to \mathbb{R}^{N\times 3}$ via an ODE governed by a time-dependent vector field (or velocity) v_τ:

$$\frac{d}{d\tau}\phi_\tau(\mathbf{x}^0) = v_\tau(\phi_\tau(\mathbf{x}^0)), \quad \phi_0(\mathbf{x}^0) = \mathbf{x}^0, \quad \tau \in [0,1], \tag{1}$$

where $\mathbf{x}^0$ is sampled from a simple distribution p_0, and ϕ_τ evolves it over time $\tau \in [0,1]$ to match the target distribution p_1 at $\tau = 1$. Since v_τ is unknown, FM learns v_τ by regressing the conditional flow $u(\phi_\tau(\mathbf{x}^0|\mathbf{x}^1)) = \frac{d}{d\tau}\phi_\tau(\mathbf{x}^0|\mathbf{x}^1)$, where $\phi_\tau(\mathbf{x}^0|\mathbf{x}^1)$ interpolates between $\mathbf{x}^0 \sim p_0$ and $\mathbf{x}^1 \sim p_1$. In our setting, each conformation $\mathbf{x}_t \in \mathbb{R}^{N\times 3}$ represents a frame in a complex trajectory, and FM is used to generate future frames from an initial structure.

4 BIOMD METHOD

4.1 A UNIFIED GENERATIVE FRAMEWORK VIA FLOW MATCHING

Our model capitalizes on a fundamental insight into molecular dynamics: conformational changes are typically subtle over short timescales but can involve significant global movements over longer timescales (**Figure 1**). This principle underpins our hierarchical prediction framework, which decomposes the generation of long trajectories into two principal stages: coarse-grained forecasting and fine-grained interpolation (**Figure 2**).

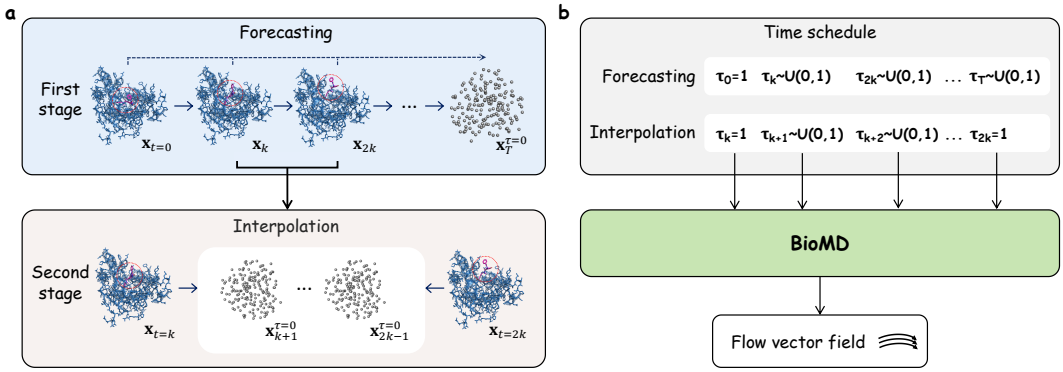

Figure 2. Model framework. (a) The hierarchical framework, showing the two-stage process of coarse-grained forecasting followed by fine-grained interpolation. (b) The time scheduling mechanism for forecasting and interpolation tasks, where known frames are noise-free ($\tau = 1$) and generated frames are noised ($\tau \in [0, 1]$).

Notably, this entire framework is implemented within a single model architecture that processes the sequence of the whole trajectory at once. We adopt a "noise as mask" strategy, where the distinction between the two stages is made simply by varying the input masking patterns (Figure 2b). In this unified framework, each frame in an input sequence is independently perturbed by noise according to a time variable τ. Known or conditioning frames are kept clean (equivalent to setting their corresponding $\tau = 1$, i.e., "unmasked"), while frames to be generated are initialized from pure noise (equivalent to $\tau = 0$, i.e., "masked") and then iteratively denoised.

Let a trajectory sequence be denoted by $\mathbf{X} = \{\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \ldots, \mathbf{x}_{t_L}\}$. During training, we sample a vector of independent time steps $\mathbf{T} = \{\tau_{t_1}, \tau_{t_2}, \ldots, \tau_{t_L}\}$, where each $\tau_{t_i} \sim U(0, 1)$. The sequence is then noised to $\mathbf{X^T} = \{\mathbf{x}^\tau_{t_1}, \ldots, \mathbf{x}^\tau_{t_L}\}$, where each frame is an interpolation between the real coordinates and Gaussian noise $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$: $\mathbf{x}^\tau_{t_i} = \tau_{t_i}\mathbf{x}_{t_i} + (1 - \tau_{t_i})\boldsymbol{\epsilon}_i$. The corresponding ground-truth velocity field for the sequence is $\mathbf{U^T} = \{\mathbf{u}^\tau_{t_1}, \ldots, \mathbf{u}^\tau_{t_L}\}$, with $\mathbf{u}^\tau_{t_i} = (\mathbf{x}_{t_i} - \mathbf{x}^\tau_{t_i})/(1 - \tau_{t_i})$.

Our velocity model u_θ takes the entire noisy sequence and conditioning information to predict the velocities for all frames simultaneously. The training objective is a Mean Squared Error loss over the entire sequence:

$$\mathcal{L}_{\text{flow}} = \text{MSE}(u_\theta(\mathbf{X^T}, \mathbf{Z}, \mathbf{T}), \mathbf{U^T}). \tag{2}$$

where $\mathbf{Z}$ contains static information including the first frame coordinate $\mathbf{x}_0$, amino acid sequence $\mathbf{s}$, and ligand atom types $\mathbf{a}$. We explore two modeling approaches: BioMD-rel, which predicts coordinate changes relative to an anchor frame, and BioMD-abs, which predicts absolute atomic coordinates. For clarity, we focus on the absolute coordinate prediction task below.

4.2 Hierarchical Generation with Forecasting and Interpolation

The two stages of our hierarchical framework are realized simply by applying different masking schedules to our unified model during training and inference.

4.2.1 Coarse-grained Forecasting

The first stage generates a coarse-grained trajectory, constructed by sampling every $k = 10$ steps from the full trajectory, resulting in a sequence $\mathbf{X}_C = \{\mathbf{x}_0, \mathbf{x}_k, \mathbf{x}_{2k}, \ldots\}$. This task is framed as a forecasting problem where, given the initial frame $\mathbf{x}_0$, the model must generate all subsequent frames.

This is achieved by applying a specific masking schedule to our unified framework. During training, the time step for the initial frame is always fixed at $\tau_0 = 1$ (making it a known, "unmasked" condition), while the time steps for all other frames $\{\tau_k, \tau_{2k}, \ldots\}$ are sampled independently from $U(0, 1)$. The model u_θ is trained to predict the velocities for all frames in the sequence, conditioned on the clean initial frame.

During inference, this setup supports multiple generation strategies:

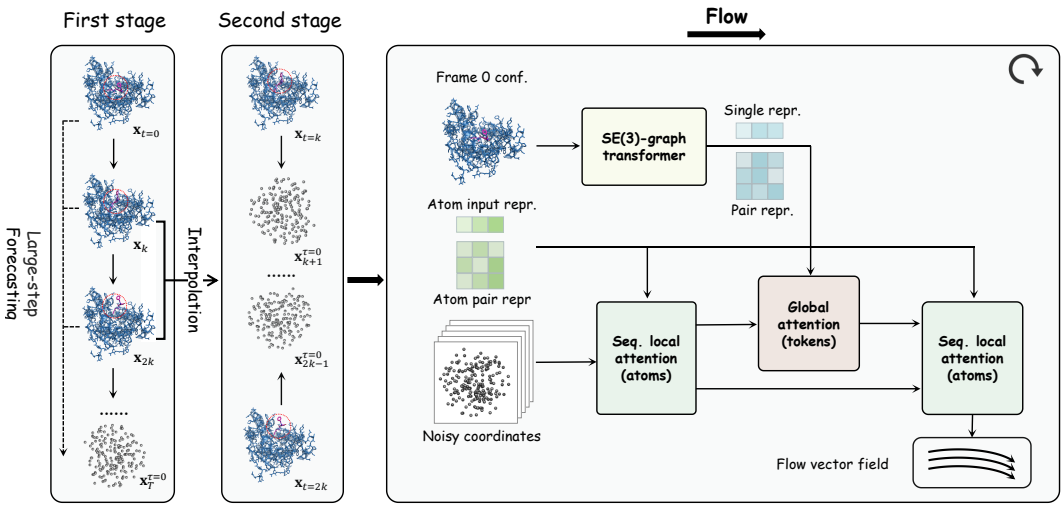

Figure 3. Detailed architecture of BioMD. The model operates in two modes, **Forecasting** and **Interpolation**, set up by the hierarchical framework (left). The core velocity network (right) processes noisy coordinates, conditioned on features from an SE(3)-Graph Transformer. A local-global-local attention pathway generates the final flow vector field used for trajectory generation.

- **All-at-once:** All future frames $\{\mathbf{x}_k, \mathbf{x}_{2k}, \dots\}$ are generated concurrently. We set $\tau_0 = 1$, initialize all other frames from noise (i.e., their τ values start at 0), and use an ODE solver like the Euler method to integrate all frames simultaneously to $\tau = 1$.
- **Auto-regressive (AR):** Frames are generated in sequential blocks of size j. To generate one such block, the model conditions on the previously generated history. This is controlled by the time variable τ: the τ values for all frames in the history are held constant at 1, making them clean, "nmasked" inputs. The τ values for all j frames within the current target block are then jointly evolved from 0 to 1 by the ODE solver. This process simultaneously denoises all frames in the block, using the generated history as context. Once generated, this block is added to the history, and the process is repeated for the next block until the full trajectory is complete.

4.2.2 Fine-grained Interpolation

After obtaining the coarse-grained trajectory $\{\mathbf{x}_0, \mathbf{x}_k, \mathbf{x}_{2k}, \dots\}$, the second stage replenishes the intermediate frames. This is an interpolation task, where for each coarse interval, we generate the frames $\{\mathbf{x}_{ik+1}, \dots, \mathbf{x}_{(i+1)k-1}\}$ conditioned on the two "anchor" frames, $\mathbf{x}_{ik}$ and $\mathbf{x}_{(i+1)k}$.

This task uses the exact same velocity model u_θ and training framework, differing only in the data and masking schedule. The input sequence is now a fine-grained segment $\mathbf{X}_I = \{\mathbf{x}_{ik}, \mathbf{x}_{ik+1}, \dots, \mathbf{x}_{(i+1)k}\}$. During training, the anchor frames are designated as known by fixing their time steps $\tau_{ik} = 1$ and $\tau_{(i+1)k} = 1$. The time steps for all intermediate frames are sampled independently from $U(0, 1)$. The model learns to generate the intermediate trajectory conditioned on the start and end conformations.

During inference, this task is always performed in an "all-at-once" manner. The anchor frames $\mathbf{x}_{ik}$ and $\mathbf{x}_{(i+1)k}$ are provided as clean inputs (their $\tau = 1$), while all intermediate frames are initialized from noise (their $\tau = 0$). The model then simultaneously generates all $k - 1$ intermediate frames by integrating them to $\tau = 1$. This process is described by:

$$\hat{\mathbf{Y}}_{ik}^{\tau+\Delta\tau} = \hat{\mathbf{Y}}_{ik}^{\tau} + u_\theta(\hat{\mathbf{X}}_I^{\mathbf{T}}, \mathbf{Z}_{\text{seq}}, \mathbf{T}) \cdot \Delta\tau, \tag{3}$$

where $\mathbf{Y}_{ik}$ represents the block of intermediate frames, and the velocity predictions are extracted for only those frames. This hierarchical approach allows BioMD to efficiently generate long, physically plausible trajectories.

4.3 VELOCITY MODEL ARCHITECTURE

BioMD is a generative model that operates directly on all-atom Cartesian coordinates. In contrast to approaches that rely on internal coordinates such as coarse-grained backbones and torsion angles, our method directly models all atoms, enabling it to capture subtle structural variations that are critical for realistic biomolecular dynamics. The effectiveness of this all-atom modeling strategy has been demonstrated by state-of-the-art biomolecular structure models like AlphaFold3 (Abramson et al., 2024). Notably, our unified model architecture is capable of performing both the forecasting and interpolation tasks (subsec. 4.2.1 and 4.2.2) within the same framework.

Our velocity model architecture is specifically tailored for generating trajectories from a single initial structure (**Figure 3**). The model first employs an SE(3) Graph Transformer to encode the initial conformation, creating rich single and pair representations. Subsequently, our core generative module, the `FlowTrajectoryTransformer` (**Algorithm 6**), operates on the entire trajectory sequence. To effectively capture complex biomolecular dynamics, each block of this transformer incorporates two primary attention mechanisms: `AttentionPairBias` is responsible for modeling intra-frame spatial interactions, while `TemporalAttention` specifically addresses inter-frame temporal dependencies by focusing on the same atom or token across different time steps. By stacking these two attention mechanisms, the model can simultaneously process spatial and temporal information, which is crucial for accurate trajectory prediction.

4.4 AUXILIARY LOSSES

In addition to the primary flow-matching objective, we incorporate several auxiliary losses to improve the physical plausibility of the generated structures. These losses are applied to the final predicted coordinates, which are obtained using the model's output velocity field.

- **Ligand Bond Loss:** To preserve the ligand's local structure, we introduce a bond loss following AlphaFold 3 Abramson et al. (2024). For each bonded atom pair in the ligand, we compute the mean squared error between the predicted inter-atomic distance and its ground-truth value, ensuring that the generated ligand structure maintains correct bond lengths throughout the generated trajectory.
- **Collision Loss:** To ensure physical plausibility and prevent steric clashes, we implement a collision loss that applies a squared penalty to non-bonded atom pairs that are unrealistically close. This loss operates on both protein-ligand and intra-ligand interactions, and penalizes inter-atomic distance that falls below a predefined threshold.
- **Ligand Geometric Center Loss:** To penalize unrealistic rigid-body movements of ligands, we define a geometric center loss. This loss calculates the mean squared error between the geometric center of the predicted ligand atoms and that of the ground-truth ligand atoms, penalizing large and unrealistic movements of the entire molecule.

5 EXPERIMENTS

We evaluate BioMD on two MD trajectory datasets: the MISATO Dataset (Siebenmorgen et al., 2024), which comprises protein-ligand interaction trajectories focusing on ligand movement within the protein binding pocket; and the DD-13M Dataset (Li et al., 2025), which contains trajectories of ligand unbinding from protein binding pockets and ultimately reaching the protein surface. Examples of predicted trajectories can be obtained from Zenodo. [1]

To comprehensively evaluate our model's performance in generating all-atom biomolecular trajectories, we first evaluate the physical stability of the generated structures for both datasets. For the MISATO dataset, given that this dataset provides conformational ensembles, we further evaluate our model's ability to predict the conformational flexibility of both proteins and ligands. For the DD-13M ligand unbinding dataset, we also evaluate the ligand unbinding success rate and introduce a ligand centroid trajectory similarity metric to assess the accuracy of the predicted unbinding pathways. In this paper, we compare BioMD with several established ML methods, including DenoisingLD (Fu et al., 2022; Wu & Li, 2023; Arts et al., 2023), GNNMD (Fu et al., 2022), VerletMD (Liu et al.,

[1] https://doi.org/10.5281/zenodo.16979768

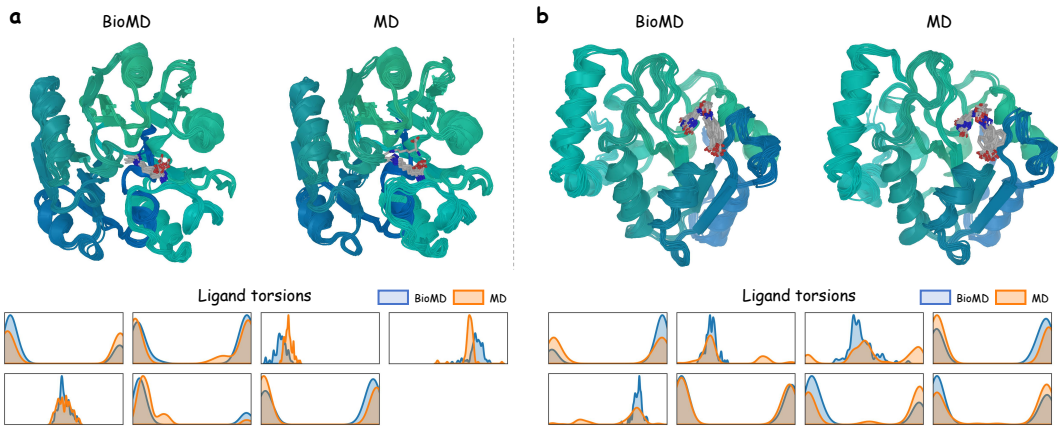

Figure 4. Conformation ensemble on the MISATO dataset. A comparison of the distributions of conformations and ligand torsion angles generated by BioMD and MD simulation for 6DGE (**a**) and 3FCF (**b**).

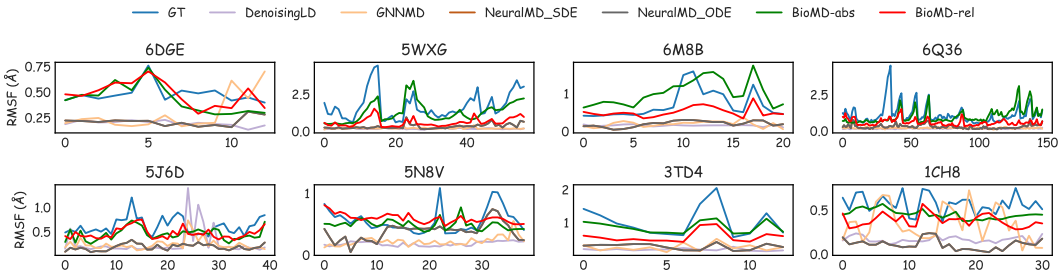

Figure 5. Ligand RMSF on the MISATO dataset. Line plot showing Ligand RMSF for eight different protein-ligand systems from the MISATO test set.

2024), and NeuralMD (Liu et al., 2024). We also include a Static model as a baseline, where the initial conformation of the system is held constant throughout the entire trajectory.

5.1 Results on MISATO

To evaluate BioMD's ability to generate realistic protein-ligand interaction trajectories, we first conduct experiments on the MISATO dataset, which focuses on ligand dynamics within the protein binding pocket. MISATO comprises nearly 20,000 protein-ligand interaction trajectories, each containing 100 frames sampled from an 8 ns MD simulation. We compare all methods on 1,031 targets with protein sequence length no longer than 800 on the MISATO test set. As shown in **Table 1**, both variants of our model, BioMD-rel and BioMD-abs, produce trajectories with promising physical stability. The bond and angle geometry errors closely approach the values of the static input structure, and the steric clash scores are orders of magnitude lower than all competing models, confirming the effectiveness of BioMD to generate physically plausible structures.

In terms of conformational flexibility, BioMD demonstrates a superior ability to capture the system's dynamic behavior. We measure Pearson's correlation between the Root Mean Square Fluctuation (RMSF) of our generated trajectories and the reference MD trajectories. BioMD achieves the highest correlation score for ligand atoms, outperforming NeuralMD by 42.8%. Besides, BioMD achieves the correlation score of 0.685 for protein atoms, while other comparing methods fail to simulate protein conformation changes. Visual analysis in **Figure 4 and Figure 5** further corroborates these findings, showing that BioMD's predicted atomic fluctuations closely trace the ground truth profiles and that the generated conformational ensemble is qualitatively similar to that of a traditional MD simulation. Collectively, these results indicate that BioMD can accurately simulate the flexibility of the entire protein-ligand complex.

Table 1. Results on the MISATO dataset. Comparison of all methods on physical stability (first six metrics) and conformational flexibility (last four metrics). Mean values on the test samples are reported.

Method	Bond Geometry[a]		Angle Geometry[a]		Steric Clashes		RMSF Correlation[b]		RMSF Value[a,c]	
	MAE	MSE	MAE	MSE	Intra-Lig	Prot-Lig	Ligand	Protein	Ligand (1.211)	Protein (1.002)
Static	.0377	.0023	.0575	.0053	0	0	-	-	-	-
DenoisingLD	$> 10^{10}$	$> 10^{27}$	.1018	.0431	.0160	.0295	-0.0290	-	$> 10^{12}$	-
GNNMD	.2123	.1032	.2115	.1072	.3626	.0028	-0.0103	-	.2165	-
NeuralMD-ODE	.0483	.0076	.0605	.0086	.0114	.0578	.3405	-	.3220	-
NeuralMD-SDE	.0483	.0076	**.0604**	.0086	.0114	.0578	.3405	-	.3220	-
VerletMD	19.73	1050	.5847	.5482	.1983	3.111	.3356	-	.3226	-
BioMD-rel	**.0395**	**.0026**	.0655	**.0075**	**.0003**	**.0006**	**.4861**	.5945	.5369	.5177
BioMD-abs	.0495	.0155	.0709	.0097	.0019	.0023	.4789	**.6854**	**.7023**	**.6242**

[a] Bond geometry (bond length) and RMSF values are in angstroms (Å). Angle geometry (bond angle) is in radians.
[b] RMSF Correlation is reported using the Pearson correlation coefficient.
[c] RMSF values for reference trajectories are given in parentheses. Values closer to those of the reference indicate better results.

Table 2. Results on the DD-13M dataset. Comparison of methods on physical stability (first six metrics), ligand unbinding path reconstruction metric (Unbinding Path RMSD), and ligand unbinding success rates. Mean values on the test samples are reported.

Method	Bond Geometry[a]		Angle Geometry[a]		Steric Clashes		Unbinding Path[a]	Unbinding Success		
	MAE	MSE	MAE	MSE	Intra-Lig	Prot-Lig	RMSD	@1	@5	@10
Static	.0254	.0013	.0461	.0037	.2778	0	.6504	0	0	0
Metadynamics[b]	.0246	.0012	.0452	.0030	.2777	0	.4217	-	-	-
BioMD-rel	**.0308**	**.0018**	.0606	.0077	.2943	.0004	.6845	.0029	.0147	.0294
BioMD-abs	.0369	.0026	**.0545**	**.0061**	**.2941**	**.0003**	.6802	.0176	.0440	.0588
BioMD-rel (AR-5)	.0580	.0100	.0918	.0184	.4021	.6375	.7055	**.7088**	**.9295**	**.9706**
BioMD-abs (AR-5)	.0728	.0111	.0802	.0132	.2943	.0009	**.5645**	.5676	.7419	.7941

[a] Bond geometry (bond length) and unbinding path RMSD values are in angstroms (Å), and angle geometry (bond angle) is in radians.
[b] The metadynamics trajectory serves as the lower-bound. The metrics are calculated among trajectories of multiple repeating simulations.

5.2 Results on DD-13M

We further evaluated BioMD on the more challenging task of ligand unbinding using the DD-13M dataset, which comprises 26,612 dissociation trajectories across 565 complexes, each with an average of 480 frames. 36 complexes were held out as a test set for evaluation, while the remaining were used for training. A key advantage of our architecture is its flexibility in supporting multiple generation strategies. A concurrent denoising of all future frames, as used on MISATO, results in minimal ligand movement because the model lacks historical guidance and averages over many potential paths. To overcome this, we generate the trajectory auto-regressively, which breaks the long-range prediction into steps and uses previously generated frames to help predict subsequent ones.

The results, summarized in **Table 2**, highlight the effectiveness of this auto-regressive strategy. While maintaining high physical stability, the BioMD-abs (AR-5) model significantly improved path accuracy, reducing the Unbinding Path RMSD to 0.5645. Most importantly, the AR strategy enabled the successful generation of complete unbinding events. The BioMD-rel (AR-5) model achieved a remarkable unbinding success rate, identifying a valid path in 70.9% of cases with a single attempt (@1), increasing to 92.9% with five attempts (@5) and 97.1% with ten attempts (@10). This demonstrates BioMD's reliability in exploring critical biomolecular pathways.

On the qualitative analysis for the 6EY8 system (**Figure 6**), our model not only reproduced the two distinct unbinding pathways found by metadynamics simulations with high fidelity but also discovered a novel third pathway, highlighting the exploratory power of our generative approach. Furthermore, BioMD achieves this with remarkable computational efficiency. While metadynamics required 2654 steps (approx. 1 hour on a single GPU) to find the first path, our model generated a complete path in under 10 seconds using just 50 coarse-grained steps.

5.3 Analysis

The success of the auto-regressive (AR) strategy in modeling long-range dynamics simultaneously exposes a fundamental challenge in generative trajectory modeling: the **error accumulation problem**.

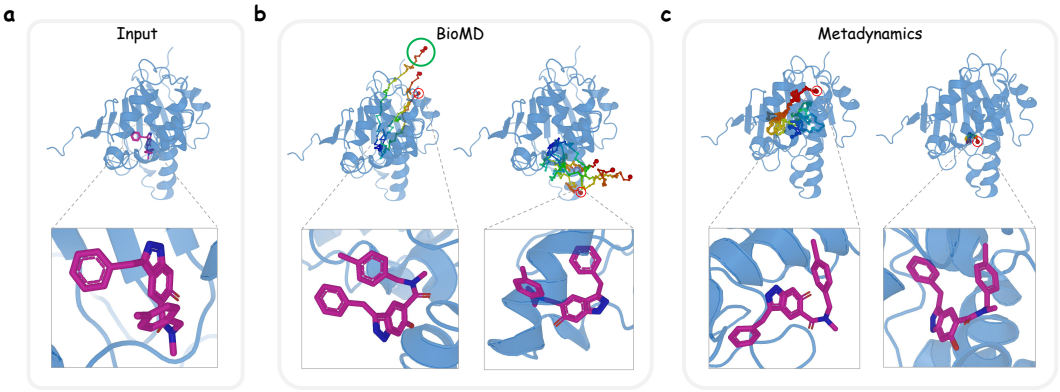

Figure 6. Ligand unbinding path on 6EY8. (a) The input conformation. (b) The unbinding pathways generated by BioMD (under 10 seconds), the novel pathway discovered by BioMD is highlighted in a green circle. (c) The reference unbinding pathways obtained using metadynamics simulations (1 hour for the left pathway).

As shown in **Table 2**, while the non-AR models produce local geometries with errors comparable to the metadynamics reference, the AR models exhibit a notable increase in error. However, thanks to our hierarchical framework, these errors remain manageable. The bond and angle MAEs for our AR models remain below 0.1 Å and 0.1 radians, respectively—a threshold well within the range of thermal fluctuations for molecular systems. These geometrical errors can be readily corrected via a simple local refinement step with minor structural deviations (< 0.1 Å), similar to the relaxation procedure used in AlphaFold. In contrast, non-hierarchical approaches are trapped between two failure modes: large AR steps yield nearly static trajectories, while small AR steps cause significant error accumulation that results in physically unrealistic structures.

Our results also reveal a distinct trade-off between predicting relative coordinate changes (BioMD-rel) and absolute coordinates (BioMD-abs). The absolute coordinate prediction method (BioMD-abs) demonstrates a superior grasp of the global conformational landscape, evidenced by its higher protein RMSF correlation on MISATO and a more accurate centroid path RMSD on DD-13M, making it the preferred choice for tasks requiring the precise reproduction of specific dynamic pathways. In contrast, the relative coordinate prediction method (BioMD-rel) excels at encouraging more exploratory behavior while preserving local chemical fidelity. Its strength is highlighted by the significantly higher unbinding success rate on DD-13M, which makes it more effective for applications focused on sampling large-scale conformational changes and discovering novel dynamic events. This functional duality means BioMD can be flexibly adapted to the specific goals of a simulation, whether the priority is accuracy in reproducing known dynamics or exploration to discover new ones.

6 Conclusion

In this work, we introduce BioMD, a novel all-atom generative model that overcomes the computational limitations of traditional molecular dynamics to simulate long-timescale biomolecular events. Our hierarchical framework, which synergistically combines coarse-grained forecasting with fine-grained interpolation, effectively mitigates error accumulation and enables the generation of physically realistic trajectories. We demonstrated BioMD's capabilities on two challenging datasets, showing it can produce stable conformations that accurately capture protein-ligand flexibility on the MISATO dataset and successfully generate complete ligand unbinding pathways for up to 97.1% of systems on the DD-13M dataset. Notably, BioMD achieves this with remarkable computational efficiency, identifying unbinding paths in seconds compared to the hours required by traditional methods like metadynamics. By offering distinct modes optimized for either accurate pathway reproduction or broad exploratory sampling, BioMD provides a powerful, flexible, and efficient tool poised to accelerate research in computational chemistry and drug discovery. [2]

[2]This paper is written with assistance from large language models (LLM) for proofreading and polishing.

REFERENCES

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J Ballard, Joshua Bambrick, et al. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493–500, 2024.

Stewart A Adcock and J Andrew McCammon. Molecular dynamics: survey of methods for simulating the activity of proteins. *Chemical reviews*, 106(5):1589–1615, 2006.

Marloes Arts, Victor Garcia Satorras, Chin-Wei Huang, Daniel Zugner, Marco Federici, Cecilia Clementi, Frank Noe, Robert Pinsler, and Rianne van den Berg. Two for one: Diffusion models and force fields for coarse-grained molecular dynamics. *Journal of Chemical Theory and Computation*, 2023.

Alessandro Barducci, Massimiliano Bonomi, and Michele Parrinello. Metadynamics. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(5):826–843, 2011.

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Allan dos Santos Costa, Ilan Mitnikov, Franco Pellegrini, Ameya Daigavane, Mario Geiger, Zhonglin Cao, Karsten Kreis, Tess Smidt, Emine Kucukbenli, and Joseph Jacobson. Equijump: Protein dynamics simulation via so (3)-equivariant stochastic interpolants. *arXiv preprint arXiv:2410.09667*, 2024.

Diego del Alamo, Davide Sala, Hassane S Mchaourab, and Jens Meiler. Sampling alternative conformational states of transporters and receptors with alphafold2. *eLife*, 11:e75751, mar 2022. doi: 10.7554/eLife.75751. URL https://doi.org/10.7554/eLife.75751.

Ron O Dror, Robert M Dirks, JP Grossman, Huafeng Xu, and David E Shaw. Biomolecular simulation: a computational microscope for molecular biology. *Annual review of biophysics*, 41:429–452, 2012.

Xiang Fu, Tian Xie, Nathan J. Rebello, Bradley Olsen, and Tommi S. Jaakkola. Simulate time-integrated coarse-grained molecular dynamics with geometric machine learning. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*, 2022. URL https://openreview.net/forum?id=rrexanVuPW5.

Scott A Hollingsworth and Ron O Dror. Molecular dynamics simulation for all. *Neuron*, 99(6): 1129–1143, 2018.

Bowen Jing, Ezra Erives, Peter Pao-Huang, Gabriele Corso, Bonnie Berger, and Tommi Jaakkola. Eigenfold: Generative protein structure prediction with diffusion models. *arXiv preprint arXiv:2304.02198*, 2023.

Bowen Jing, Bonnie Berger, and Tommi Jaakkola. Alphafold meets flow matching for generating protein ensembles. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 22277–22303, 2024a.

Bowen Jing, Hannes Stärk, Tommi Jaakkola, and Bonnie Berger. Generative modeling of molecular dynamics trajectories. *Advances in Neural Information Processing Systems*, 37:40534–40564, 2024b.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

Martin Karplus and J Andrew McCammon. Molecular dynamics simulations of biomolecules. *Nature structural biology*, 9(9):646–652, 2002.

Alessandro Laio and Michele Parrinello. Escaping free-energy minima. *Proceedings of the national academy of sciences*, 99(20):12562–12566, 2002.

Sarah Lewis, Tim Hempel, José Jiménez-Luna, Michael Gastegger, Yu Xie, Andrew YK Foong, Victor García Satorras, Osama Abdin, Bastiaan S Veeling, Iryna Zaporozhets, et al. Scalable emulation of protein equilibrium ensembles with generative deep learning. *Science*, pp. eadv9817, 2025.

Maodong Li, Jiying Zhang, Bin Feng, Wenqi Zeng, Dechin Chen, Zhijun Pan, Yu Li, Zijing Liu, and Yi Isaac Yang. Enhanced sampling, public dataset and generative model for drug-protein dissociation dynamics. *arXiv preprint arXiv:2504.18367*, 2025.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *ICLR*, 2023.

Shengchao Liu, Weitao Du, Yanjing Li, Zhuoxinran Li, Vignesh Bhethanabotla, Nakul Rampal, Omar Yaghi, Christian Borgs, Anima Anandkumar, Hongyu Guo, et al. A multi-grained symmetric differential equation model for learning protein-ligand binding dynamics. *arXiv preprint arXiv:2401.15122*, 2024.

Jiarui Lu, Bozitao Zhong, Zuobai Zhang, and Jian Tang. Str2str: A score-based framework for zero-shot protein conformation sampling. In *The Twelfth International Conference on Learning Representations*, 2024.

Jiarui Lu, Xiaoyin Chen, Stephen Zhewen Lu, Aurelie Lozano, Vijil Chenthamarakshan, Payel Das, and Jian Tang. Aligning protein conformation ensemble generation with physical feedback. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Asr955jcuZ.

Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

David E Shaw, Ron O Dror, John K Salmon, JP Grossman, Kenneth M Mackenzie, Joseph A Bank, Cliff Young, Martin M Deneroff, Brannon Batson, Kevin J Bowers, et al. Millisecond-scale molecular dynamics simulations on anton. In *Proceedings of the conference on high performance computing networking, storage and analysis*, pp. 1–11, 2009.

David E Shaw, Paul Maragakis, Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, Michael P Eastwood, Joseph A Bank, John M Jumper, John K Salmon, Yibing Shan, et al. Atomic-level characterization of the structural dynamics of proteins. *Science*, 330(6002):341–346, 2010.

Yuning Shen, Lihao Wang, Huizhuo Yuan, Yan Wang, Bangji Yang, and Quanquan Gu. Simultaneous modeling of protein conformation and dynamics via autoregression. *arXiv preprint arXiv:2505.17478*, 2025.

Till Siebenmorgen, Filipe Menezes, Sabrina Benassou, Erinc Merdivan, Kieran Didi, André Santos Dias Mourão, Radosław Kitel, Pietro Liò, Stefan Kesselheim, Marie Piraud, et al. Misato: machine learning dataset of protein–ligand complexes for structure-based drug discovery. *Nature Computational Science*, pp. 1–12, 2024.

Richard A Stein and Hassane S Mchaourab. Speach_af: Sampling protein ensembles and conformational heterogeneity with alphafold2. *PLoS computational biology*, 18(8):e1010483, 2022.

Loup Verlet. Computer "experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Physical Review*, 159(1):98–103, 1967.

Tong Wang, Xinheng He, Mingyu Li, Yatao Li, Ran Bi, Yusong Wang, Chaoran Cheng, Xiangzhen Shen, Jiawei Meng, He Zhang, et al. Ab initio characterization of protein molecular dynamics with ai2bmd. *Nature*, 635(8040):1019–1027, 2024a.

Yan Wang, Lihao Wang, Yuning Shen, Yiqun Wang, Huizhuo Yuan, Yue Wu, and Quanquan Gu. Protein conformation generation via force-guided se (3) diffusion models. In *Forty-first International Conference on Machine Learning*, 2024b.

Hannah K Wayment-Steele, Adedolapo Ojoawo, Renee Otten, Julia M Apitz, Warintra Pitsawong, Marc Hömberger, Sergey Ovchinnikov, Lucy Colwell, and Dorothee Kern. Predicting multiple conformations via sequence clustering and alphafold2. *Nature*, 625(7996):832–839, 2024.

Fang Wu and Stan Z Li. Diffmd: A geometric diffusion model for molecular dynamics simulations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 5321–5329, 2023.

Shuxin Zheng, Jiyan He, Chang Liu, Yu Shi, Ziheng Lu, Weitao Feng, Fusong Ju, Jiaxi Wang, Jianwei Zhu, Yaosen Min, et al. Predicting equilibrium distributions for molecular systems with deep learning. *Nature Machine Intelligence*, 6(5):558–567, 2024.

A TECHNICAL APPENDICES AND SUPPLEMENTARY MATERIAL

A.1 DETAILED MODEL ARCHITECTURE

Hierarchical Generation Framework. As illustrated in **Figure 3**, BioMD employs a hierarchical framework to perform both coarse-grained forecasting and fine-grained interpolation within a unified model. The specific task is controlled by applying noise selectively. For **Forecasting**, the initial frame $\mathbf{x}_0$ is provided without noise, while all subsequent frames are initialized from a standard Gaussian distribution. For **Interpolation**, two anchor frames (e.g., $\mathbf{x}_k$ and $\mathbf{x}_{2k}$) are kept clean, while the intermediate frames are initialized from noise. The model's objective is to denoise the masked frames conditioned on the known ones.

Input Representation and Conditioning. The core of the model is the `FlowModule` (**Algorithm 4**), which processes three primary inputs. The main dynamic input is the set of Noisy Coordinates ($\{\vec{\mathbf{x}}_l^{\text{noisy}}\}$), representing the current state of the trajectory. To provide structural context, the initial conformation (Frame 0 conf.) is processed by an SE(3)-Graph Transformer, as detailed in the main inference loop (**Algorithm 1**). This produces static Single ($\{\mathbf{s}_i^{\text{trunk}}\}$) and Pair ($\{\mathbf{z}_{ij}^{\text{trunk}}\}$) representations. These representations, along with other atom features, are processed by the `FlowConditioning` module (**Algorithm 5**) to generate the final conditioning signals.

Spatial-Temporal Attention Pathway. The `FlowModule` uses a local-global-local attention pathway to predict the velocity field. First, the noisy coordinates and conditioning features are passed to an `AtomAttentionHistoryEncoder`, which models local atomic environments. The resulting representations are aggregated into tokens and fed into the central `FlowTrajectoryTransformer` (**Algorithm 6**). This module integrates spatial and temporal information using two key mechanisms: `AttentionPairBias` resolves intra-frame spatial relationships, while `TemporalAttention` captures inter-frame dynamics. The globally-aware token representations are then broadcast back to the atomic level, where an `AtomAttentionDecoder` computes the final per-atom updates.

Velocity Field Prediction and Trajectory Generation. The output of the `FlowModule` is the Flow vector field ($\{\vec{\mathbf{u}}_l\}$), which represents the predicted velocity for each atom. During training (**Algorithm 2**), the model is optimized via a mean squared error loss between the predicted velocity and the true velocity. During inference (**Algorithm 3**), this vector field is used in an Euler integration step, $\vec{\mathbf{x}}_l^{\tau+1} \leftarrow \vec{\mathbf{x}}_l^{\tau} + dt \cdot \vec{\mathbf{u}}_l^{\tau}$, to iteratively update the coordinates from a noisy state to a final, structured trajectory.

A.2 AUXILIARY LOSSES

After we get the estimated vector field $\mathbf{u}_\theta$, we can get the predicted structure coordinates via

$$\hat{\mathbf{x}}_t^1 = \hat{\mathbf{x}}_t^\tau + \mathbf{u}_\theta(1-\tau), \tag{4}$$

and then we get the predicted protein and ligand structure $[\hat{\mathbf{x}}_t^{\mathcal{P}}, \hat{\mathbf{x}}_t^{\ell}] = \hat{\mathbf{x}}_t^1$.

Ligand geometric center loss. To stabilize the global placement of the ligand and prevent spurious rigid translations, we align the predicted and reference geometric centers of ligand atoms. Let $\mathbf{x}_t^\ell = \{\mathbf{x}_t^{\ell,i}\}_{i=1}^{N_\ell}$ and $\hat{\mathbf{x}}_t^\ell = \{\hat{\mathbf{x}}_t^{\ell,i}\}_{i=1}^{N_\ell}$ denote ground-truth and predicted ligand coordinates at step t. The geometric center is

$$C(\mathbf{x}_t^\ell) = \frac{1}{N_\ell}\sum_{i=1}^{N_\ell}\mathbf{x}_t^{\ell,i}, \qquad C(\hat{\mathbf{x}}_t^{\ell,i}) = \frac{1}{N_\ell}\sum_{i=1}^{N_\ell}\hat{\mathbf{x}}_t^{\ell,i},$$

and the loss is the mean-squared discrepancy

$$\mathcal{L}_{\text{center}} = \left\| C(\hat{\mathbf{x}}_t^\ell) - C(\mathbf{x}_t^\ell) \right\|_2^2.$$

This term softly anchors the ligand's global position while remaining agnostic to its internal geometry.

Collision loss. To penalize steric clashes, we define a collision loss between protein–ligand atoms and within ligand atoms. Let $\mathbf{x}_t^\ell$ and $\mathbf{x}_t^\mathcal{P}$ denote ligand and protein atom coordinates at step t, and $\hat{\mathbf{x}}_t^\ell$, $\hat{\mathbf{x}}_t^\mathcal{P}$ their predictions. We compute predicted distances

$$d_{ij}^{PL} = \|\hat{\mathbf{x}}_t^{\mathcal{P},i} - \hat{\mathbf{x}}_t^{\ell,j}\|_2, \quad d_{ij}^{L} = \|\hat{\mathbf{x}}_t^{\ell,i} - \hat{\mathbf{x}}_t^{\ell,j}\|_2,$$

and corresponding ground-truth minimal distances

$$d_{ij}^{PL,gt} = \min_t \|\mathbf{x}_t^{\mathcal{P},i} - \mathbf{x}_t^{\ell,j}\|_2, \quad d_{ij}^{LL,gt} = \min_t \|\mathbf{x}_t^{\ell,i} - \mathbf{x}_t^{\ell,j}\|_2.$$

Protein–ligand and ligand–ligand thresholds are set as

$$\zeta_{ij}^{PL} = \min\Big(0.9\, d_{ij}^{PL,gt},\ \zeta_{pl}\Big) \qquad \zeta_{ij}^{LL} = \min\Big(0.9\, d_{ij}^{LL,gt},\ \zeta_{ll}\Big),$$

where $\zeta_{pl} = 3.0\,\text{Å}$ and $\zeta_{ll} = 2.0\,\text{Å}$.

The collision loss is then defined as

$$\mathcal{L}_{\text{collision}} = \sum_{i,j} \mathbf{1}\big(d_{ij}^{PL} < \zeta_{ij}^{PL}\big)\ (\zeta_{ij}^{PL} - d_{ij}^{PL})^2 + \sum_{i \neq j} \mathbf{1}\big(d_{ij}^{LL} < \zeta_{ij}^{LL}\big)\ (1 - b_{ij})\,(\zeta_{ij}^{LL} - d_{ij}^{LL})^2,$$

where $\mathbf{1}(\cdot)$ represents the indicator function and b_{ij} is the ligand bond mask to exclude bonded pairs.

Ligand bond loss. To preserve ligand bond lengths, we penalize deviations between predicted and ground-truth bonded atom distances. Let $\mathcal{B}$ denote the set of bonded atom pairs according to the ligand bond mask. For each bond $(i, j) \in \mathcal{B}$, we compute the predicted and ground-truth distances

$$d_{ij}^{\ell} = \|\hat{\mathbf{x}}_t^{\ell,i} - \hat{\mathbf{x}}_t^{\ell,j}\|_2, \quad d_{ij}^{\ell,gt} = \|\mathbf{x}_t^{\ell,i} - \mathbf{x}_t^{\ell,j}\|_2.$$

The bond loss is then defined as the mean squared deviation:

$$\mathcal{L}_{\text{bond}} = \frac{1}{|\mathcal{B}|} \sum_{(i,j)\in\mathcal{B}} \Big(d_{ij}^{\ell} - d_{ij}^{\ell,gt}\Big)^2.$$

Geometric constraint loss. We combine the above terms into a single geometric regularizer

$$\mathcal{L}_{\text{geom}} = \lambda_{\text{col}}\, \mathcal{L}_{\text{collision}} + \lambda_{\text{bond}}\, \mathcal{L}_{\text{bond}} + \lambda_{\text{ctr}}\, \mathcal{L}_{\text{center}},$$

where $\lambda_{\text{col}}, \lambda_{\text{bond}}, \lambda_{\text{ctr}} > 0$ balance steric clash avoidance, bond-length preservation, and global ligand anchoring, respectively.

A.3 Evaluation Metrics

A.3.1 Physical Stability

This metric assesses whether the generated trajectories preserve physically stable conformations, which is essential to ensure chemical validity and avoid unrealistic molecular structures. We evaluate stability from two complementary perspectives:

1. **Local Structure Stability.** To assess whether the generated trajectories maintain chemically reasonable local geometries, we calculate the deviations of bond lengths and bond angles with respect to the initial frame of the reference trajectories. Both the Mean Absolute Error (MAE) and Mean Squared Error (MSE) are reported. Lower values indicate that the generated conformations remain close to the idealized covalent structure and are thus more chemically stable.
2. **Steric Clashes.** We further quantify the presence of steric conflicts, which occur when non-bonded atoms are unrealistically close to each other. Specifically, a clash is counted if the interatomic distance (excluding bonded pairs and angle-related atoms) is less than a threshold of 1.5 Å. We compute clash scores for both intra-ligand and protein–ligand interactions, where the score corresponds to the average number of clashes per generated conformation. Lower clash scores indicate physically more plausible conformations.

A.3.2 CONFORMATIONAL FLEXIBILITY

In addition to stability, it is important that generated trajectories capture the dynamic flexibility of molecular systems. For the MISATO protein–ligand interaction dataset, we adopt the Root Mean Square Fluctuation (RMSF) to quantify the extent of atomic motion over time after trajectory alignment:

$$\text{RMSF}_i = \sqrt{\frac{1}{T}\sum_{t=1}^{T} \|\mathbf{r}_i(t) - \bar{\mathbf{r}}_i\|^2},$$

where $\mathbf{r}_i(t)$ is the position of atom i at time t, and $\bar{\mathbf{r}}_i$ is its time-averaged position.

We evaluate flexibility from two perspectives: 1. **Global Consistency.** We compute the Pearson correlation coefficient between the RMSFs of generated and reference trajectories, where higher correlation indicates better agreement in the fluctuation profiles. 2. **Magnitude Accuracy.** We also report the average RMSF of the generated trajectories. Values closer to the reference average RMSF imply that the model produces realistic levels of conformational motion rather than being overly rigid or excessively flexible.

A.3.3 UNBINDING PATH DISTANCE

For the DD-13M ligand unbinding dataset, we evaluate whether generated unbinding trajectories follow realistic spatial pathways compared to reference simulations. We compute the Root Mean Square Deviation (RMSD) between generated and reference ligand centroid trajectories with the following procedure:

1. **Trajectory Standardization.** All trajectories are resampled to a uniform length using linear interpolation, ensuring comparability between different sequences.
2. **Best-Match Search.** For each generated trajectory, we identify the reference trajectory that yields the minimum RMSD. This accounts for the possibility of multiple plausible unbinding pathways.
3. **Final Score.** The reported metric is the average of these best-match RMSDs across all generated trajectories. Lower RMSD values indicate that the model generates ligand motions more consistent with physically realistic unbinding paths.

A.3.4 UNBINDING SUCCESS

This metric evaluates whether the generated ligand trajectories successfully capture the unbinding event. Specifically, we construct the convex hull of the protein heavy atoms in the initial bound state. If at least one predicted ligand centroid position lies outside this convex hull, the trajectory is considered as a successful unbinding case.

We report the **Success@k**, which measures the probability that at least one out of k independently generated trajectories for the same protein–ligand complex achieves successful unbinding. A higher success rate indicates a better capability of the model to reproduce realistic ligand unbinding processes. Formally, for each complex with k attempts, Success@k is defined as

$$\text{Success@}k = \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}\left[\max_{1\le j\le k} s_n^{(j)} = 1\right],$$

where $s_n^{(j)}$ is the binary success indicator (1 if the j-th trajectory of complex n achieves unbinding, 0 otherwise), and N is the total number of complexes. We report Success@1, Success@5, and Success@10, which reflect performance under single, moderate, and multiple generation attempts, respectively.

Algorithm 1: Main Inference Loop

Input: $\{\mathbf{f}^*\}$, $\{\vec{\mathbf{x}}_{0,l}\}$, $N_{\text{cycle}} = 4$, $c_s = 384$, $c_z = 128$

$\{\mathbf{s}_i^{\text{inputs}}\} \leftarrow \text{InputFeatureEmbedder}(\{\mathbf{f}^*\})$;
$\mathbf{s}_i^{\text{init}} \leftarrow \text{LinearNoBias}(\mathbf{s}_i^{\text{inputs}})$;
$\mathbf{z}_{ij}^{\text{init}} \leftarrow \text{LinearNoBias}(\mathbf{s}_i^{\text{inputs}}) + \text{LinearNoBias}(\mathbf{s}_j^{\text{inputs}})$;
$\{\mathbf{z}_{ij}\}, \{\mathbf{s}_i\} \leftarrow 0, 0$;
foreach $c \in \{1, \ldots, N_{cycle}\}$ **do**
  $\mathbf{z}_{ij} \leftarrow \mathbf{z}_{ij}^{\text{init}} + \text{LinearNoBias}(\text{LayerNorm}(\mathbf{z}_{ij}))$;
  $\{\mathbf{z}_{ij}\}, \{\mathbf{s}_i\} \leftarrow \text{GraphTransformer}(\{\vec{\mathbf{x}}_{0,l}\}, \{\mathbf{s}_i\}, \{\mathbf{z}_{ij}\}, \{\mathbf{s}_i^{\text{inputs}}\})$;
  $\mathbf{s}_i \leftarrow \mathbf{s}_i^{\text{init}} + \text{LinearNoBias}(\text{LayerNorm}(\mathbf{s}_i))$;
traj_list=[$\{\vec{\mathbf{x}}_{0,l}\}$] ;
foreach $t \in \{1, \ldots, T\}$ **do**
  $\{\vec{\mathbf{x}}_{t,l}^{\text{pred}}\} \leftarrow \text{SampleFlow}(\{\vec{\mathbf{x}}_{his}\}, \{\mathbf{f}^*\}, \{\mathbf{s}_i^{\text{inputs}}\}, \{\mathbf{s}_i\}, \{\mathbf{z}_{ij}\})$;
  traj_list.add($\vec{\mathbf{x}}_t^{\text{pred}}$);
  $\{\vec{\mathbf{x}}_{his}\}$ = traj_list;
return *traj_list*

Algorithm 2: TrainFlow

Input: $\{\vec{\mathbf{x}}_l\}$,$\{\vec{\mathbf{x}}_{his}\}$, $\{\mathbf{f}^*\}$, $\{\mathbf{s}_i^{\text{inputs}}\}$, $\{\mathbf{s}_i^{\text{trunk}}\}$, $\{\mathbf{z}_{ij}^{\text{trunk}}\}$

Indepentent noise levels ;
$\tau \sim (\mathcal{U}(0,1), \mathcal{U}(0,1), \cdots, \mathcal{U}(0,1))$;
$\{\vec{\mathbf{x}}_l^0\} \sim \mathcal{N}(\vec{0}, \mathbf{I}_3)$;
$\{\vec{\mathbf{x}}_l\} \leftarrow \text{CentreRandomAugmentation}(\{\vec{\mathbf{x}}_l\})$;
$\{\vec{\mathbf{x}}_l^\tau\} = \tau\{\vec{\mathbf{x}}_l\} + (1-\tau)\{\vec{\mathbf{x}}_l^0\}$;
$\{\vec{\mathbf{u}}_l^\tau\} \leftarrow \text{FlowModule}(\{\vec{\mathbf{x}}_l^\tau\}, \{\vec{\mathbf{x}}_{his}\}, \tau, \{\mathbf{f}^*\}, \{\mathbf{s}_i^{\text{inputs}}\}, \{\mathbf{s}_i^{\text{trunk}}\}, \{\mathbf{z}_{ij}^{\text{trunk}}\})$;
$\mathcal{L}_{flow} = \text{MSE}(\{\vec{\mathbf{u}}_l^\tau\}, \{\frac{\vec{\mathbf{x}}_l - \vec{\mathbf{x}}_l^\tau}{1-\tau}\})$;
return $\mathcal{L}_{flow}$

Algorithm 3: SampleFlow

Input: $\{\vec{\mathbf{x}}_{his}\}$, $\{\mathbf{f}^*\}$, $\{\mathbf{s}_i^{\text{inputs}}\}$, $\{\mathbf{s}_i^{\text{trunk}}\}$, $\{\mathbf{z}_{ij}^{\text{trunk}}\}$

$\vec{\mathbf{x}}_l^0 \sim \mathcal{N}(\vec{0}, \mathbf{I}_3)$;
foreach τ *in* $\{0, 0.1, 0.2, ..., 0.9\}$ **do**
  $\{\vec{\mathbf{u}}_l^\tau\} \leftarrow \text{FlowModule}(\{\vec{\mathbf{x}}_l^\tau\}, \{\vec{\mathbf{x}}_{his}\}, \tau, \{\mathbf{f}^*\}, \{\mathbf{s}_i^{\text{inputs}}\}, \{\mathbf{s}_i^{\text{trunk}}\}, \{\mathbf{z}_{ij}^{\text{trunk}}\})$;
  $\vec{\mathbf{x}}_l^{\tau+1} \leftarrow \vec{\mathbf{x}}_l^\tau + dt \cdot \vec{u}_l^\tau$;
return $\{\vec{\mathbf{x}}_l^1\}$

Algorithm 4: FlowModule

Input: $\{\vec{\mathbf{x}}_l^{\text{noisy}}\}$,$\{\vec{\mathbf{x}}_{his}\}$, t, $\{\mathbf{f}^*\}$, $\{\mathbf{s}_i^{\text{inputs}}\}$, $\{\mathbf{s}_i^{\text{trunk}}\}$, $\{\mathbf{z}_{ij}^{\text{trunk}}\}$
$\sigma_{\text{data}} = 16$, $c_{\text{atom}} = 128$, $c_{\text{atompair}} = 16$, $c_{\text{token}} = 768$

$\{\mathbf{s}_i\}, \{\mathbf{z}_{ij}\} \leftarrow$ FlowConditioning$(t, \{\mathbf{f}^*\}, \{\mathbf{s}_i^{\text{inputs}}\}, \{\mathbf{s}_i^{\text{trunk}}\}, \{\mathbf{z}_{ij}^{\text{trunk}}\}, \sigma_{\text{data}})$;
Sequence-local Atom Attention with history info and aggregation to coarse-grained tokens ;
$\{a_i\}, \{q_k^{\text{skip}}\}, \{p_k^{\text{skip}}\}, \{t_k^{\text{skip}}\} \leftarrow$ AtomAttentionHistoryEncoder$(\{\vec{\mathbf{x}}_{his}\}, \{\mathbf{f}^*\}, \{\vec{\mathbf{x}}_l^{\text{noisy}}\}, \{\mathbf{s}_i\}, \{\mathbf{z}_{ij}\}, c_{\text{atom}}, c_{\text{atompair}}, c_{\text{token}})$;
Full self-attention on token level.;
$a_i \leftarrow$ LinearNoBias(LayerNorm(a_i));
$\{a_k\} \leftarrow$ FlowTrajectoryTransformer$(\{a_i\}, \{\mathbf{s}_i\}, \{\mathbf{z}_{ij}\}, \beta_{ij} = 0, N_{\text{block}} = 24, N_{\text{head}} = 16)$;
$a_i \leftarrow$ LayerNorm(a_i);
Broadcast token activations to atoms and run Atom Attention.;
$\{\vec{\mathbf{u}}_l\} \leftarrow$ AtomAttentionDecoder$(\{a_i\}, \{q_k^{\text{skip}}, p_k^{\text{skip}}, t_k^{\text{skip}}\})$;
return $\{\vec{\mathbf{u}}_l\}$

Algorithm 5: FlowConditioning

Input: $\hat{t}$, $\{\mathbf{f}^*\}$, $\{s_i^{\text{inputs}}\}$, $\{s_i^{\text{trunk}}\}$, $\{z_{ij}^{\text{trunk}}\}$, σ_{data}, $c_z = 128$, $c_s = 384$

Pair conditioning;
$z_{ij} \leftarrow$ LinearNoBias(LayerNorm(z_{ij}));
foreach $b \in \{1, 2\}$ **do**
  z_{ij} += Transition$(z_{ij},\ n = 2)$;
Single conditioning;
$s_i \leftarrow$ concat$([\ s_i^{\text{trunk}},\ s_i^{\text{inputs}}\ ])$;
$s_i \leftarrow$ LinearNoBias(LayerNorm(s_i));
foreach $b \in \{1, 2\}$ **do**
  s_i += Transition$(s_i,\ n = 2)$;
return $\{s_i\}, \{z_{ij}\}$

Algorithm 6: FlowTrajectoryTransformer

Input: $\{a_i\}$, $\{s_i\}$, $\{z_{ij}\}$, $\{\beta_{ij}\}$, N_{block}, N_{head}

for $n \in [1, \ldots, N_{\text{block}}]$ **do**
  $\{b_i\} \leftarrow$ AttentionPairBias$(\{a_i\}, \{s_i\}, \{z_{ij}\}, \{\beta_{ij}\}, N_{\text{head}})$;
  $\{b_i\} \leftarrow$ TemporalAttention$(\{a_i + b_i\})$;
  $a_i \leftarrow b_i +$ ConditionedTransitionBlock(a_i, s_i);
return $\{a_i\}$