STRUCTURE-CONSTRAINED KOOPMAN GENERATORS FOR FLEXIBLE TIME-STRIDE MOLECULAR DYNAMICS SIMULATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Trajectory-based models can accelerate molecular dynamics (MD) simulations by predicting configurations over long temporal strides, but most are tied to a single prescribed temporal resolution during training, limiting their ability to leverage heterogeneous trajectory datasets and to infer dynamics at unseen time strides. To this end, we introduce FLEXMD, a continuous-time Koopman framework that enables flexible time-stride prediction via structure-constrained equivariant latent dynamics. It encodes molecular configurations into an E(3)-invariant latent space and evolves them with a constrained linear generator. The generator preserves a learned quadratic invariant that regularizes latent evolution and prevents exponential amplification of error with positive definite matrix. The closed-form propagation of latent dynamics via matrix exponential enables a single model to learn across multiple temporal strides and perform inference at flexible temporal resolutions without iterative latent-space integration. We evaluate our method across various benchmarking datasets, including MD17, AdK, ATLAS, MD-CATH, and PET-MAD. Crucially, our model achieves accurate and stable long-horizon simulation, supports flexible inference across substantial training-test stride gaps, and benefits from multi-stride training that robustly improves generalization across temporal resolutions. These results demonstrate that the structure-constrained Koopman generator offers a principled and scalable path toward reliable long-timescale molecular simulation.

1 INTRODUCTION

Molecular dynamics (MD) simulation is a cornerstone of drug discovery, materials science, and protein folding studies (Shaw et al., 2010; Noé et al., 2020). Traditional approaches rely on either high-fidelity *ab initio* methods or empirical force fields to evolve molecular systems over time (Car & Parrinello, 1985; Behler & Parrinello, 2007). Recently, deep learning has emerged as a compelling alternative, particularly via machine-learning force fields (MLFFs) that approximate potential energy surfaces from data (Schütt et al., 2017; Unke et al., 2021; Batatia et al., 2022; Liao & Smidt, 2023). Despite their accuracy, MLFFs still heavily rely on classical numerical integration with very small time-strides to ensure stability and limit energy drift, making long-horizon simulations computationally expensive (Batatia et al., 2022; Liao & Smidt, 2023). To improve efficiency, trajectory-based models have been proposed to learn coarse-grained dynamics by directly predicting molecular states over large temporal strides (Xu et al., 2024; Wu et al., 2024; Han et al., 2022).

Recent advances extend the trajectory-based paradigm by framing molecular dynamics as a trajectory-generative problem and using deep generative models to model time-evolved configurations over long horizons (Jing et al., 2024b; Bigi et al., 2025; Shen et al., 2025; Shoghi et al., 2026). Although these methods enable inference with large strides, most existing methods use fixed-stride formulations in which the simulation time-stride is specified during training. This creates two related challenges. First, a model trained with a large stride alone generally cannot provide accurate, finer-grained predictions. For example, MarS-FM must be combined with MDGen (Jing et al., 2024b) for local finer-grained trajectory generation (Kapusniak et al., 2026). Second, trajectory datasets are typically curated at dataset-specific sampling intervals. Models trained on one temporal stride cannot readily exploit datasets sampled at different lag time, and the heterogeneity of saved-frame

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

time lags complicates training a trajectory-generative model across different time strides. These limitations motivate a framework that learns from trajectories with heterogeneous temporal strides, supports flexible-stride inference, and enables efficient, stable long-horizon MD simulation.

We address this problem with FLEXMD, a quadratic-invariant Koopman framework for continuous-time latent molecular dynamics. Koopman theory provides the underlying modeling principle: nonlinear configuration dynamics can be represented approximately as linear evolution in a learned observable space (Lusch et al., 2018). Existing Koopman approaches, however, primarily target general dynamical systems (Grozavescu et al., 2026) or structured dynamical systems (Goyal et al., 2025); they do not jointly address geometric symmetry, continuous-time propagation, and flexible-stride prediction for molecular trajectories.

Our idea first uses an E(3)-equivariant encoder to map each molecular configuration to a representation invariant under global rigid transformations. A constrained linear generator then continuously evolves the latent representation over time. The generator exactly preserves a learned quadratic form in latent space. Under a positive-definiteness condition, this constraint prevents exponential error amplification in the induced norm and yields the linear error accumulation. The proposed formulation combines the temporal flexibility of continuous-time modeling with the efficiency of direct trajectory prediction. A single model can therefore be trained with observations from multiple strides and evaluated at different temporal resolutions without iterative integration in latent space. The quadratic constraint further reduces error amplification, supporting efficient long-stride inference, yielding stable long-horizon simulations.

Our contributions are threefold. First, we design an E(3)-equivariant latent propagation framework for molecular trajectory generation across temporal resolutions. The framework supports training on multiple time-strides and flexible time-stride inference through a closed-form latent propagator. Second, we formulate a continuous-time Koopman generator with a structured parameterization that exactly preserves a learned quadratic latent invariant. Under a positive-definiteness condition, we further characterize the resulting latent error growth with a controlled bound theoretically and empirically. Third, we evaluate FLEXMD across five benchmarks spanning small molecules, protein dynamics, protein thermodynamics, and materials simulations. FLEXMD achieves lower long-horizon prediction errors and longer stable rollouts than the evaluated baselines, while maintaining strong structural, distributional, and thermodynamic performance. In experiments with flexible time-stride training, the model incorporates transition pairs sampled from Markov state-model-guided trajectories and improves structural fidelity. Moreover, joint training across multiple time-strides outperforms separately trained baselines, paving the way and suggesting a promising direction for scalable training of trajectory models that can operate across heterogeneous temporal resolutions.

2 PRELIMINARIES

Molecular Dynamics. Consider a molecular system comprising N atoms, its configurational state is represented as a graph $\widetilde{G}_t = (V, E_t, X, R_t, P_t)$ at time t, with nodes $V = \{1, \dots, N\}$, edges E_t defined by interactions (e.g., proximity or bonds), node features $X \in \mathbb{R}^{N \times d_x}$, coordinates $R_t \in \mathbb{R}^{N \times 3}$, and atomic momenta $P_t \in \mathbb{R}^{N \times 3}$. An MD simulator numerically evolves the phase-space state using an integration step Δt, producing a trajectory $\{\widetilde{G}_t\}_{t=0}^{T}$. From this trajectory, one may construct time-lagged pairs $(\widetilde{G}_t, \widetilde{G}_{t+\tau})$ at lag $\tau = \ell \Delta t$. The resulting dynamics define a generally nonlinear flow map $\widetilde{G}_{t+\tau} = \Phi^\tau(\widetilde{G}_t), \Phi^\tau : \widetilde{\mathcal{G}} \to \widetilde{\mathcal{G}}$, on the molecular phase space $\widetilde{\mathcal{G}}$. In practice, however, momenta are unavailable in most trajectory datasets, which typically retain only atomic types and coordinates. We therefore take the configurational graph $G_t = (V, E_t, X, R_t)$ as the default and learn the position-space evolution $G_{t+\tau} = \Phi^\tau(G_t), \Phi^\tau : \mathcal{G} \to \mathcal{G}$. When momentum annotations are available, our framework extends naturally to $\widetilde{G}_t$ by treating P_t as an additional equivariant vector feature and jointly predicting positions and momenta,

Molecular dynamics consider trajectories $\{G_t\}_{t=0}^{T}$ generated by an MD simulator with step size Δt. One can construct time-lagged pairs $(G_t, G_{t+\tau})$ with $\tau = \ell \Delta t$, sampled from a trajectory. The dynamics can be regarded as a nonlinear system on $\mathcal{X}$, governed by the flow map $G_{t+\tau} = \Phi^\tau(G_t)$, where $\Phi^\tau : \mathcal{X} \to \mathcal{X}$ describes the evolution over lag time τ. We provide a notation table in Appendix A.

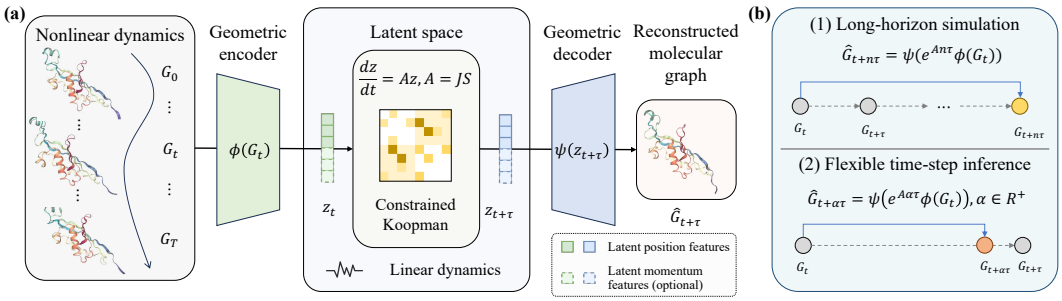


Figure 1: (a): Molecular trajectory data in nonlinear configuration space are mapped to a latent space via an equivariant encoder, where the latent dynamics evolve linearly under a quadratic-form-preserving Koopman generator $\dot{z} = Az$ with $A = JS$. A decoder reconstructs molecular configurations. (b): The closed-form solution $z(t) = e^{At}z(0)$ supports long-horizon rollout and flexible time-stride prediction.

Koopman Theory. Koopman theory provides a linear representation of nonlinear dynamics by lifting the evolution to the observable space. For any observable $f : \mathcal{X} \to \mathbb{R}$, the Koopman operator $\mathcal{K}^\tau$ is defined as $(\mathcal{K}^\tau f)(G) = f(\Phi^\tau(G))$. Although Φ^τ is nonlinear, $\mathcal{K}^\tau$ is a linear operator acting on a function space, generally infinite-dimensional. Its spectral structure characterizes coherent dynamical modes, where eigenfunctions evolve linearly over time. For continuous-time dynamics, the infinitesimal Koopman generator $\mathcal{A}_{op}$ is defined as $\mathcal{A}_{op}f = \lim_{t\to 0} \frac{\mathcal{K}^t f - f}{t}$, which induces the evolution equation $\frac{d}{dt}f = \mathcal{A}_{op}f$. This yields a linear dynamical system in function space.

Trajectory-Based ML Methods for Molecular Dynamics. Recent advancements in learning-based MD methods aim to bypass traditional integration costs through learned surrogates. These methods can be broadly categorized into: (i) Equivariant Integrators and Simulators, such as EGNN (Satorras et al., 2021) and EGHN (Han et al., 2022), which parameterize the molecular graph and learn direct state-to-state regressions (Klein et al., 2023; Bigi et al., 2025); (ii) Equivariant Neural Operators, including ITO (Schreiner et al., 2023) and EGNO (Xu et al., 2024), which model the evolution of the system via neural operators; and (iii) Equivariant Generative Models, such as MDGen (Jing et al., 2024b) and STAR-MD (Shoghi et al., 2026), which utilize diffusion or autoregressive processes to sample conformational ensembles. MarS-FM (Kapusniak et al., 2026) offers a complementary perspective by learning to generate transitions between discrete metastable states defined by an underlying Markov state model. Despite their success, existing simulators and operators often rely on regressing complex nonlinearities, leading to cumulative integration errors. We provide a brief review of related works in Appendix B.1.

3 METHOD

Overview. We study the problem of learning molecular dynamics from trajectory data defined over molecular configurations $G_t \in \mathcal{X}$, where $\mathcal{X}$ denotes the space of molecular graphs with 3D geometry. The underlying evolution is modeled by a nonlinear flow map $\Phi^\tau : \mathcal{X} \to \mathcal{X}$ such that $G_{t+\tau} = \Phi^\tau(G_t)$ by most existing works (Xu et al., 2024; Jing et al., 2024b). Directly approximating Φ^τ using neural networks often leads to instability and degraded accuracy over long time horizons. The proposed FLEXMD learns a structured latent representation in which nonlinear dynamics are mapped to a linear continuous-time system with quadratic invariant structure. The model consists of three components: $G \xrightarrow{\phi} z \xrightarrow{e^{At}} z_t \xrightarrow{\psi} \hat{G}_t$, where $\phi : \mathcal{X} \to \mathbb{R}^{2d}$ is an encoder, $A \in \mathbb{R}^{2d\times 2d}$ is a structured Koopman generator, and $\psi : \mathbb{R}^{2d} \to \mathcal{X}$ is a decoder. This formulation enables continuous-time latent evolution with an explicit algebraic constraint on the generator, while avoiding direct approximation of the nonlinear flow in configuration space.

3.1 Equivariant Latent Representation

Molecular systems exhibit invariance to global translations and equivariance to rotations in $\mathbb{R}^3$. A molecular configuration $G = (V, E, X, R)$ consists of atomic features $X \in \mathbb{R}^{N \times d_x}$ and Cartesian coordinates $R \in \mathbb{R}^{N \times 3}$. These correspond to different geometric types: X are scalar features (invariant under rotations), while R are vector features (equivariant under rotations). An equivariant encoder must therefore treat these components in a manner consistent with E(3) symmetry.

Equivariant Encoding and Decoding. We implement ϕ with EGNN-style message passing (Satorras et al., 2021): edge messages depend on scalar node features and squared interatomic distances, while coordinate updates are scalar-weighted sums of relative displacement vectors. Thus, the scalar features are invariant and the coordinate stream is equivariant to global rigid transformations. Pooling the final scalar features yields a permutation- and E(3)-invariant latent state; the layer updates and decoder are specified in Appendix C.1.

Invariant Readout. The encoder produces node-wise embeddings consisting of scalar (and optionally vector) channels. To obtain a global latent representation, we apply an invariant pooling operator (e.g., sum or mean over nodes) to the scalar channels: $z = \mathrm{Pool}\big(\{h_i^{(L)}\}_{i=1}^N\big)$, which ensures that z is invariant to permutations, translations, and rotations. Vector features are either pooled via invariant norms or projected to scalars before aggregation.

Latent Representation. The latent state is given by $z = \phi(G), z \in \mathbb{R}^{2d}$, where z is a global equivariant representation of the molecular configuration. The dimensionality $2d$ is chosen to match the structure required by the quadratic constrained generator in the subsequent section. Although the latent dimension is even because of the generator parameterization, the two blocks of z are not assigned physical position or momentum semantics. Velocities and momenta are unavailable in most trajectory datasets, so the full latent vector is treated as a learned data representation constrained to evolve under a quadratic-form-preserving linear operator. z as a data-driven representation that admits a quadratic invariant under a structured linear evolution. This design offers flexibility while retaining the structure constraints of the Koopman framework.

This construction ensures that (i) scalar atomic features and vector geometric information are integrated in a symmetry-consistent manner, and (ii) the resulting latent representation is invariant to global rigid transformations. Consequently, the learned dynamics operate on intrinsic molecular structure rather than coordinate-dependent artifacts, forming a principled foundation for learning the structured latent evolution.

3.2 Structured Koopman Latent Dynamics

While the encoder constructs an equivariant latent representation, the key challenge lies in modeling its temporal evolution. A naive linear parameterization of latent dynamics, as commonly used in Koopman-based approaches, may lead to unstable or poorly conditioned generators when applied to molecular systems. To this end, we impose an algebraic constraint on the generator that preserves a learned quadratic form. This constraint restricts the hypothesis class of the latent dynamics and regularizes the learned operator.

Structured Latent Dynamics. We model the latent evolution as a linear dynamical system $\frac{dz}{dt} = Az$, where the generator is parameterized in quadratic invariant form as

$$A = JS, \quad J = \begin{pmatrix} 0 & I_d \\ -I_d & 0 \end{pmatrix}, \tag{1}$$

with $S \in \mathbb{R}^{2d \times 2d}$ parameterized as $S = BB^\top + \epsilon_S I$, where B is learned and the fixed $\epsilon_S > 0$ ensures $S \succ 0$. This structure induces a quadratic invariant $H(z) = \frac{1}{2} z^\top S z$, such that the dynamics can be written as $\frac{dz}{dt} = J \nabla H(z)$. Here, the quantity $H(z)$ is a learned latent invariant. From a Koopman perspective, this corresponds to learning a finite-dimensional invariant subspace in which the dynamics evolve under a structured linear operator. In contrast to unconstrained Koopman models, the proposed parameterization restricts the generator to a quadratic-form-preserving family, which can improve conditioning and robustness in long-horizon molecular trajectory prediction.

Error Propagation. The imposed structure also influences how approximation errors accumulate over time. For a rollout initialized at G_0, define the encoded true latent state after k prediction strides as $z_k^* := \phi(\Phi^{k\tau}(G_0))$, and the corresponding predicted latent state as $\hat{z}_k := e^{Ak\tau}\phi(G_0)$. Thus, $z_0^* = \hat{z}_0 = \phi(G_0)$. The latent prediction error is $e_k := \hat{z}_k - z_k^*$, and the one-stride discrepancy is $\delta_k := z_{k+1}^* - e^{A\tau} z_k^*$. It follows that

$$e_{k+1} = e^{A\tau} e_k - \delta_k. \tag{2}$$

Since the designed flow preserves the quadratic form induced by S, error propagation can be analyzed in the corresponding S-induced norm.

Based on this recursion, we obtain the following characterization of error growth.

Proposition 3.1 (Quadratic-Invariant–Controlled Error Growth)**.** *Let $\dot{z} = Az$ with $A = JS$ and $S \succ 0$. Define $\|z\|_S := \sqrt{z^\top S z}$. For the error and one-stride discrepancy defined above, $e_0 = 0$ and $e_{k+1} = e^{A\tau} e_k - \delta_k$. For the latent prediction error and one-stride discrepancy defined above, $e_0 = 0$ and $e_{k+1} = e^{A\tau} e_k - \delta_k$. Since $e^{A\tau}$ preserves the S-norm, the error after $n \geq 1$ prediction strides satisfies*

$$\|e_n\|_S \leq \sum_{i=0}^{n-1} \|\delta_i\|_S. \tag{3}$$

This result highlights that, under the structured dynamics, error growth is controlled in the S-norm and avoids exponential amplification from the homogeneous latent propagator under the stated assumptions. A Euclidean-norm consequence requires the conditioning factor relating the S-norm to the Euclidean norm. A fully rigorous formulation and proof are provided in Appendix D.

3.3 Training and Inference with Flexible Time-stride

Training Objective. We learn the encoder ϕ, decoder ψ, and generator A jointly from trajectory data. The training objective combines reconstruction and latent dynamics consistency:

$$\mathcal{L} = \lambda_{\text{rec}} \|G_t - \psi(\phi(G_t))\|^2 + \lambda_{\text{dyn}} \|\phi(G_{t+\tau}) - e^{A\tau}\phi(G_t)\|^2. \tag{4}$$

Optionally, multi-stride consistency can be enforced by $\|\phi(G_{t+k\tau}) - e^{Ak\tau}\phi(G_t)\|^2$, which further stabilizes long-horizon predictions. This training formulation encourages the learned representation to align with a linear evolution whose generator preserves a learned quadratic form while preserving reconstructive fidelity.

Inference Tasks. The model supports long-horizon rollout by $\hat{G}_{n\tau} = \psi\big(e^{An\tau}\phi(G_0)\big)$ and flexible time-stride simulation by $\hat{G}_{\alpha\tau} = \psi\big(e^{A\alpha\tau}\phi(G_0)\big), \alpha \in \mathbb{R}$. Specifically, the latent dynamics admit a closed-form solution for continuous-time evolution with $z(t) = e^{At}z(0)$. Given an initial configuration G_0, predictions at arbitrary time t are obtained via $\hat{G}_t = \psi\big(e^{At}\phi(G_0)\big)$.

Training with Momentum Features. When atomic momenta are available, we augment the graph with $P_t \in \mathbb{R}^{N\times 3}$, yielding $\widetilde{G}_t = (V, E_t, X, R_t, P_t)$. Momenta are treated as rotation-equivariant, translation-invariant vector features. The encoder maps $\widetilde{G}_t$ to $\widetilde{z}_t = [q_t; p_t] \in \mathbb{R}^{2d}$, where q_t and p_t are learned blocks without prescribed physical meaning; the same generator $A = JS$ propagates this state, and the decoder predicts coordinates and momenta. We train this variant with $\mathcal{L}_{\text{mom}} = \lambda_R\|R_t - \widehat{R}_t\|^2 + \lambda_P\|\bar{P}_t - \widehat{\bar{P}}_t\|^2 + \lambda_{\text{dyn}} \left\|\widetilde{z}_{t+\tau} - e^{A\tau}\widetilde{z}_t\right\|^2$, where $\bar{P}$ denotes momenta after mass scaling and dataset-level standardization. We invert these transformations for physical-unit reporting; implementation details appear in Appendix C.2.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate FlexMD on five benchmarks spanning molecular, protein, and materials dynamics: MD17, adenylate kinase (AdK), ATLAS, MD-CATH, and PET-MAD. Dataset splits,

preprocessing, time strides, and sampling protocols follow the corresponding prior work (Xu et al., 2024; Jing et al., 2024b; Kapusniak et al., 2026; Bigi et al., 2025); the detailed experimental setup will be given in Appendix E.1.

Baselines. We compare FLEXMD with representative methods from several modeling paradigms. RF is a random-feature regression baseline; TFN, EGNN, and SE(3)-Transformer are equivariant graph predictors; EGHN is an equivariant hierarchy-based graph network; EGNO is an equivariant graph neural operator with temporal convolutions; and ESTAG is a spatiotemporal graph-attention model. Generative baselines include MDGen, FlashMD, ConfRover, and AlphaFlow, representing conditional diffusion, direct long-stride prediction, autoregressive conformational modeling, and flow matching, respectively. For MD-CATH, we additionally include MARS-FM, its hierarchical/direct-sampling variant, MARS-FM followed by MDGen for short-timescale refinement, and the pretrained BioEmu model. The detailed baseline configurations will be given in Appendix E.2.

Tasks. a) *Long-horizon trajectory generation*: We evaluate FLEXMD using long-horizon rollouts on MD17, AdK, and ATLAS. We distinguish between the physical lag time τ and the discrete time-stride index ℓ, related by $\tau = \ell\Delta t$, where Δt denotes the fundamental MD integration step. Unless otherwise specified, reported time-stride values refer to ℓ. b) *Variable time-stride inference*: To evaluate flexible temporal-resolution modeling, we consider settings in which the test stride ℓ_{test} differs from the training stride. Specifically, ℓ_{test} is sampled from $\{1, 5, 10, 25, 100, 1{,}000, 3{,}000\}$, with corresponding lag time $\tau_{\text{test}} = \ell_{\text{test}}\Delta t$. This setting measures generalization across temporal resolutions, whereas most discrete-time baselines are tied to a fixed training stride. c) *Variable time-stride training*: To assess robustness to temporal resolution during training, we train FLEXMD on trajectories sampled at multiple time strides and evaluate it alongside MARS-FM (Kapusniak et al., 2026), which models transitions between long-timescale metastable states.

Metrics. We assess performance from three perspectives: prediction accuracy, rollout stability, and distributional fidelity. For trajectory prediction, we report conformation mean squared error (MSE) over rollout horizons (Xu et al., 2024; Wu et al., 2024). For stability, we measure bond-connectivity preservation during generated simulations (Fu et al., 2023). For protein dynamics, following prior work (Jing et al., 2024b), we report pairwise RMSD, global RMSF, and per-target RMSF to quantify structural evolution and residue-level flexibility. For distributional evaluation, we report the Wasserstein-2 distance (W_2) in coordinate and PCA-reduced spaces, as well as the KL and Jensen–Shannon divergences of protein observables and the mean absolute error (MAE) of folding free energies on MD-CATH. Unless otherwise stated, arrows in table captions indicate whether higher or lower values are preferred.

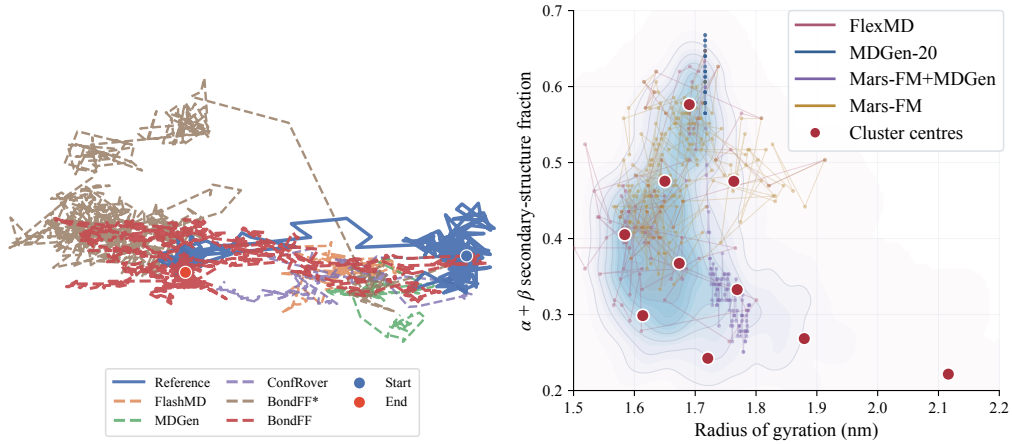
4.2 RESULTS AND ANALYSIS

4.2.1 LONG-HORIZON TRAJECTORY GENERATION

Results on MD17. On MD17, FLEXMD achieves the lowest final-step conformation MSE for all eight molecules (Table 5). It also yields the longest structurally connected rollouts for seven of the eight molecules and remains competitive on naphthalene (Table 6). These results support the effectiveness of the quadratic-form-preserving Koopman generator for accurate, stable long-horizon prediction. We provide full results and detailed analysis in Appendix F.1.

Results on ATLAS. As shown in Table 1, FLEXMD achieves the best performance across the structural and distributional metrics on ATLAS, demonstrating its ability to model high-dimensional protein dynamics. It obtains the highest correlations for pairwise RMSD, global RMSF, and per-target RMSF, indicating that the generated trajectories capture both global conformational evolution and local residue flexibility. FLEXMD also achieves the lowest Wasserstein-2 distances in coordinate and PCA spaces, showing close alignment with the reference distribution.

Compared with generative baselines such as MDGen, FlashMD, and AlphaFlow, FLEXMD provides stronger structural consistency while maintaining competitive distributional fidelity. FLEXMD *—trained on coarser time-stride snapshots—also remains competitive, supporting the robustness of the continuous-time formulation to changes in temporal resolution. In addition, FLEXMD has the lowest



(a) PCA visualization of long-horizon molecular trajectories on 6pxz_B_R1 (ATLAS). FLEXMD preserves global structure and dynamical consistency.

(b) MD-CATH trajectories in the space of radius of gyration and secondary-structure fraction. Shading shows the reference distribution; red markers indicate cluster centers.

Figure 2: Qualitative comparison of generated molecular dynamics on ATLAS and MD-CATH.

Table 1: Protein dynamics modeling on ATLAS: structural correlation (↑), distributional distance (↓), and runtime. FLEXMD and the baselines are trained and evaluated with time-stride 40 ($\ell = \ell_{\text{test}} = 40$); FLEXMD * is trained with time-stride 120 and evaluated with time-stride 40.

	MDGen	FlashMD	AlphaFlow	ConfRover	FLEXMD	FLEXMD*
Pairwise RMSD r ↑	0.48± 0.03	0.39± 0.01	0.53± 0.10	0.46± 0.02	**0.72**± 0.01	0.63± 0.02
Global RMSF r ↑	0.50± 0.02	0.60± 0.06	0.48± 0.04	0.59± 0.01	**0.71**± 0.03	0.62± 0.05
Per-target RMSF r ↑	0.71± 0.03	0.75± 0.03	0.82± 0.05	0.83± 0.01	**0.86**± 0.05	0.84± 0.04
Root mean $\mathcal{W}_2$ dist. ↓	2.69± 0.06	2.71± 0.04	2.67± 0.002	2.63± 0.02	**2.51**± 0.01	2.66± 0.05
MD PCA $\mathcal{W}_2$ dist. ↓	1.89± 0.01	1.95± 0.03	1.49± 0.03	1.83± 0.02	**1.42**± 0.02	1.65± 0.04
JSD ↓	0.56± 0.01	0.72± 0.01	0.89± 0.02	0.52± 0.03	**0.48**± 0.03	0.51± 0.01
TICA ↑	0.12± 0.02	0.11± 0.02	0.09± 0.01	0.15± 0.02	**0.17**± 0.004	0.15± 0.02
Runtime (s)*	1.92	5.28	134.75	2.29	0.97	0.97

* FLEXMD * is trained with time-stride-120 interval snapshots ($\ell = 120$) and evaluated with time-stride 40.

reported runtime, benefiting from linear latent evolution and closed-form propagation. It therefore offers a favorable combination of accuracy, stability, and computational efficiency for large-scale protein simulation.

Figure 2(a) shows two-dimensional PCA projections of reference and generated rollouts for the ATLAS protein `6pxz_B_R1`. The reference trajectory forms a low-dimensional manifold with progression from the start to the end state. Most compared baselines do not reach the projected reference endpoint, whereas FLEXMD more closely follows the reference path and approaches its endpoint. This qualitative comparison is consistent with the ATLAS results in Table 1. Overall, the trajectory projection visualization highlights that FLEXMD achieves better long-horizon fidelity compared to existing approaches.

4.2.2 FLEXIBLE TIME-STRIDE TRAINING

Results on MD-CATH. Following MARS-FM (Kapusniak et al., 2026), we assess whether generated conformations reproduce the structural and thermodynamic distributions of MD-CATH test domains at 450 K. The train/test split limits sequence identity to 20%; we generate 500 conformations per domain in each of ten independent inference runs.

We train the flexible-time-stride model in two stages: first on all chronological frame pairs at $\ell_{\text{train}} = 1$ (1 ns), then on MSM-guided source–target pairs spanning multiple temporal separations. Retaining both frame indices lets us condition on each pair's physical separation, combining dense short-time supervision with state-balanced coverage of longer-time transitions. At inference,

Table 2: Protein distribution modeling on MD-CATH. Pearson correlation (↑) is reported for structural correlations and folding free energies; forward KL divergence, JSD, and folding-free-energy MAE (↓) are reported for distributional metrics. Results are shown as mean ± standard deviation over ten inference runs using 500 generated conformations per domain.

Method	P. RMSD r ↑	G. RMSF r ↑	P.-t. RMSF r ↑	G. KL ↓	G. JSD ↓	S. KL ↓	S. JSD ↓	MSM JSD ↓	ΔG_{fold} MAE ↓
MD (Oracle)	0.82± 0.01	0.82± 0.01	0.90± 0.01	0.58± 0.02	0.07± 0.00	0.61± 0.05	0.07± 0.00	0.19± 0.01	0.90± 0.05
MDGen-20	0.40± 0.02	0.38± 0.01	0.38± 0.01	0.89± 0.01	0.20± 0.00	1.85± 0.02	0.30± 0.00	0.43± 0.01	1.44± 0.03
MDGen-100	0.42± 0.05	0.49± 0.02	0.64± 0.01	1.13± 0.03	0.15± 0.00	1.17± 0.05	0.18± 0.01	0.29± 0.01	1.21± 0.01
BioEmu	0.25± 0.00	0.41± 0.00	0.66± 0.00	3.83± 0.01	0.40± 0.00	4.17± 0.02	0.41± 0.00	0.51± 0.00	4.67± 0.00
MARS-FM	0.65± 0.00	0.71± 0.00	0.89± 0.00	0.55± 0.00	**0.10**± 0.00	0.93± 0.01	0.13± 0.00	0.19± 0.00	1.05± 0.00
+ MDGen-20	0.63± 0.00	0.69± 0.00	0.83± 0.00	0.98± 0.02	0.13± 0.00	0.73± 0.01	0.11± 0.00	0.24± 0.00	1.02± 0.00
FLEXMD	**0.69**± 0.03	**0.74**± 0.01	**0.89**± 0.02	**0.53**± 0.04	**0.10**± 0.05	**0.69**± 0.00	**0.11**± 0.02	**0.17**± 0.02	**1.02**± 0.02

P. RMSD: pairwise RMSD; G. RMSF: global RMSF; P.-t. RMSF: per-target RMSF; G. KL: gyration-radius KL divergence; G. JSD: gyration-radius JSD; S. KL: secondary-structure KL divergence; S. JSD: secondary-structure JSD; MSM JSD: Markov-state-model JSD; ΔG_{fold} MAE: folding-free-energy mean absolute error.

we alternate between $\ell = 1$ and $\ell = 125$ to generate 500 conformations; the latter approximates the mean separation of the MSM-guided pairs. Adaptive time-stride selection remains future work. Further details appear in Appendix E.3.

The reported baselines are MDGen-20, MDGen-100, BioEmu, direct MARS-FM sampling, and MARS-FM followed by MDGen-20 refinement. We report Pearson correlations for RMSD and RMSF, forward KL and Jensen–Shannon divergences (JSD) for structural observables, and folding-free-energy MAE. Figure 2(b) illustrates trajectories from FLEXMD and selected baselines for one MD-CATH domain. In this example, FLEXMD samples the reference distribution's densely populated region and reaches several cluster neighbourhoods, consistent with the distributional results in Table 2.

Results of PET-MAD. Following the FlashMD NVT protocol (Bigi et al., 2025), we report the absolute difference between effective and target temperatures, $|\Delta T|$, under Langevin and stochastic velocity rescaling (SVR) thermostats; lower is better. We compare with FlashMD because both models use momentum inputs and the same inference-time energy-conservation filter, and report results with and without that filter. Further protocol details appear in Appendix F.2.

Without the filter, FLEXMD (1 fs) reduces the Langevin and SVR errors at the 1 fs stride to 23.80 and 5.30 K, respectively, from FlashMD's 35.50 and 7.30 K. The corresponding single-stride FLEXMD models also have lower mean errors than FlashMD at 4 and 16 fs under both thermostats. Jointly trained FLEXMD * has higher unfiltered errors than FLEXMD (1 fs) at 1 fs, but lower errors than the corresponding single-stride FLEXMD models at 4 and 16 fs; at 16 fs, its errors are 15.42 K under Langevin and 3.90 K under SVR. This suggests a benefit from joint time-stride training at the larger evaluated strides. With the filter, all reported mean errors are at most 0.60 K, and the methods perform similarly. As $|\Delta T|$ is an absolute measure, positive and negative deviations in FlashMD's original convention are treated alike.

4.2.3 FLEXIBLE TIME-STRIDE INFERENCE

Results on MD17 with flexible time-stride inference. We evaluate flexible time-stride generalization on MD17 (aspirin), where FLEXMD is trained across a range of time strides ($\ell \in [5, 3{,}000]$) and evaluated over the same range ($\ell_{\text{test}} \in [5, 3{,}000]$). Table 3 reports the baseline results with their strongest available configurations, while FLEXMD is evaluated across various training and test strides. FLEXMD achieves consistently strong performance across the evaluated strides while supporting continuous-time inference within a single model. In contrast, EGNO degrades when evaluated at smaller strides, whereas MDGen and ConfRover rely on interpolation and access to future states rather than causal rollout. FLEXMD matches or outperforms the strongest baseline in most regimes and is particularly effective at large test strides, where it achieves the lowest errors at $\ell_{\text{test}} = 100$, 1,000, and 3,000. These results indicate that the continuous-time, constrained Koopman formulation supports robust extrapolation across temporal resolutions without retraining or task-specific adaptation.

Table 3: Flexible time-stride evaluation on MD17 (aspirin): Next Time-stride MSE. All models are trained on strides from 5 to 3,000 and evaluated across the same range (lower is better). For baselines, we report their strongest available configurations: EGNO cannot extrapolate to smaller strides, while MDGen and ConfRover operate via interpolation and require access to future states. FLEXMD supports continuous-time inference and is evaluated uniformly across all test strides.*

Test time-step ℓ_{test}	EGNO	MDGen	ConfRover	FLEXMD (trained at ℓ)					
				$\ell = 5$	$\ell = 10$	$\ell = 25$	$\ell = 100$	$\ell = 1,000$	$\ell = 3,000$
5	4.95e-6± 0.5	3.50e-6± 0.5	2.50e-6± 0.6	**1.88e-6**± 0.8	1.96e-6± 0.5	9.96e-6± 0.7	8.24e-6± 0.7	7.21e-6± 1.1	3.64e-6± 0.6
10	4.43e-5± 0.2	6.40e-5± 0.7	9.84e-5± 0.4	7.83e-6± 0.4	8.16e-6± 0.5	8.29e-6± 0.3	**7.09e-6**± 0.5	1.88e-5± 0.1	1.02e-5± 0.7
25	4.07e-4± 0.2	3.40e-5± 0.5	6.80e-5± 0.4	5.79e-5± 0.3	6.65e-5± 1.1	3.09e-4± 0.1	5.09e-4± 0.2	**3.17e-5**± 0.8	4.32e-5± 0.7
100	5.51e-3± 0.4	3.42e-4± 0.4	8.62e-4± 0.5	1.63e-3± 0.5	2.19e-3± 0.1	5.19e-3± 0.6	4.97e-3± 0.9	4.45e-3± 0.2	**1.52e-4**± 0.1
1,000	5.97e-2± 0.6	2.40e-2± 0.4	9.20e-2± 0.6	4.29e-2± 0.8	3.21e-2± 0.9	2.27e-2± 0.6	3.49e-2± 0.5	1.85e-2± 0.8	**1.42e-2**± 0.6
3,000	1.79e-1± 0.2	9.26e-2± 0.5	1.16e-1± 0.6	3.16e-2± 0.1	2.38e-2± 0.1	**1.29e-2**± 0.7	1.75e-2± 0.6	1.96e-2± 0.3	2.23e-2± 0.4

* Most baselines achieve their best applicable performance when the training and test strides are matched.

Table 4: Flexible time-stride MD simulation on AdK: Next Time-stride MSE. Models are trained with stride $\ell = 5$ and evaluated at strides from 5 to 100 (lower is better). Entries marked "–" correspond to extremely collapsed inferred structures.*

ℓ_{test}	EGNN	EGHN	EGNO	ESTAG	MDGen	FlashMD	AlphaFlow	ConfRover	FLEXMD
5	2.19± 0.34	2.01± 1.08	2.29± 1.47	1.82± 1.64	1.20± 0.47	1.35± 0.40	1.14± 0.96	1.19± 0.51	**0.50**± 0.51
10	4.39± 2.40	3.49± 1.78	3.94± 1.29	3.24± 2.64	2.59± 0.55	2.10± 1.28	2.09± 1.66	1.49± 1.66	**0.61**± 0.10
25	–	–	–	–	5.59± 1.11	5.39± 2.48	3.24± 2.53	–	**0.78**± 0.82
100	–	–	–	–	–	–	–	–	**0.91**± 0.69

* At the omitted settings (–), the corresponding models produce extremely collapsed inferred structures, leading to excessively large errors.

Results on AdK with flexible time-stride inference. As shown in Table 4, FLEXMD achieves lower prediction error than the competing methods across the evaluated time strides, with the largest gains at longer horizons. While the baselines show increasing error as the evaluation stride grows, FLEXMD degrades more gradually, suggesting stronger generalization across temporal resolutions. This empirical trend is consistent with the norm-preserving latent propagator, although the theoretical bound applies to latent error and does not by itself guarantee physical trajectory stability.

Parameter sensitivity analysis and ablation study. We analyze latent dimensionality, training stride, and the structured Koopman generator on MD17 (aspirin). Larger latent dimensions improve convergence and final accuracy, with diminishing gains beyond $2d = 128$, while larger training strides make optimization more difficult (Appendix Fig. 4). In the ablation, constrained and unconstrained generators have comparable short-stride errors, but the unconstrained variant becomes unstable at test strides of 1,000 and 3,000, where the constrained variant remains stable (Appendix Table 8). These results support the quadratic-form constraint in addressing the error accumulation for long-stride inference.

5 CONCLUSION

We introduced FLEXMD, a structure-preserving framework for learning molecular dynamics that integrates Koopman linearization, a quadratic-form-preserving generator constraint, and geometric equivariance. By modeling latent dynamics with a structured linear generator that preserves a learned quadratic form, FLEXMD enables stable, continuous-time prediction with analytical propagation, avoiding the limitations of discrete-time and purely data-driven approaches. Across multiple benchmarks, FLEXMD demonstrates consistent gains in long-horizon accuracy, stability, and temporal generalization, including strong performance under arbitrary time-step evaluation. These results suggest that combining algebraic structure with operator learning provides an effective path toward scalable and reliable molecular simulation. Future work may extend this framework to incorporate stochastic dynamics, more expressive Hamiltonians, and applications to larger biomolecular systems and drug discovery tasks.

AI USE STATEMENT

In this work, we used generative AI tools to assist with coding, understand existing literature, provide feedback on research methodology and experiments, improve writing and presentation, and write scripts. We did not use these tools to develop theoretical models or conceptual frameworks; formulate mathematical claims; provide key steps in mathematical proofs; design research methodology or experiments; draft proofs; propose or refine hypotheses; implement methods; translate text; clean or reformat datasets; support qualitative or thematic analysis; or interpret results. Generating synthetic datasets was not applicable to this work.

We also used generative AI tools for brainstorming, searching for information, and suggesting keywords for the paper. We reviewed all AI-assisted work and manually surveyed the literature to check AI-generated research ideas for overlap with prior work. We take responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our study uses molecular, protein, and materials simulation benchmarks and does not involve human participants or personal data. Although improved molecular simulation can support scientific discovery, it could also aid harmful chemical design; applications in sensitive settings therefore require appropriate oversight and independent validation. We discuss the method's limitations and broader impacts in Appendix G and Appendix H.

REPRODUCIBILITY STATEMENT

The model is described in the main text, and the *Experiments* section reports the evaluation metrics and results. The appendix gives implementation details in Appendix C.1, assumptions and a complete proof in Appendix D, dataset splits and preprocessing in Appendix E.1, baseline configurations in Appendix E.2, and the MSM-guided sampling procedure in Appendix E.3. These sections also report training settings, the stated random seed, and evaluation protocols to support reproduction of the results. We also uploaded anonymous code as supplementary materials.

REFERENCES

Ilyes Batatia, David P Kovacs, Gregor Simm, Christoph Ortner, and Gábor Csányi. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in Neural Information Processing Systems*, 35:11423–11436, 2022.

Jörg Behler and Michele Parrinello. Generalized Neural-Network Representation of High-Dimensional Potential-Energy Surfaces. *Physical review letters*, 98(14):146401, 2007.

Filippo Bigi, Sanggyu Chong, Agustinus Kristiadi, and Michele Ceriotti. FlashMD: Long-Stride, Universal Prediction of Molecular Dynamics. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Richard Car and Mark Parrinello. Unified Approach for Molecular Dynamics and Density-Functional Theory. *Physical review letters*, 55(22):2471, 1985.

Kaiwen Cheng, Yutian Liu, Zhiwei Nie, Mujie Lin, Yanzhen Hou, Yiheng Tao, Chang Liu, Jie Chen, Youdong Mao, and Yonghong Tian. ProAR: Probabilistic Autoregressive Modeling for Molecular Dynamics. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(1):147–155, Mar. 2026. doi: 10.1609/aaai.v40i1.36974.

Xiang Fu, Zhenghao Wu, Wujie Wang, Tian Xie, Sinan Keten, Rafael Gomez-Bombarelli, and Tommi Jaakkola. Forces are not Enough: Benchmark and Critical Evaluation for Machine Learning Force Fields with Molecular Simulations. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. Survey Certification.

Pawan Goyal, Süleyman Yıldız, and Peter Benner. Deep Learning for Structure-Preserving Universal Stable Koopman-Inspired Embeddings for Nonlinear Canonical Hamiltonian Dynamics. *Machine Learning: Science and Technology*, 6(1):015063, 2025.

Rares Grozavescu, Pengyu Zhang, Etienne Meunier, and Mark Girolami. Koopman Autoencoders with Continuous-Time Latent Dynamics for Fluid Dynamics Forecasting. *arXiv preprint arXiv:2602.02832*, 2026.

Lingbing Guo, Weiqing Wang, Zhuo Chen, Ningyu Zhang, Zequn Sun, Yixuan Lai, Qiang Zhang, and Huajun Chen. Newton–Cotes Graph Neural Networks: On the Time Evolution of Dynamic Systems. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 21541–21557. Curran Associates, Inc., 2023.

Jiaqi Han, Wenbing Huang, Tingyang Xu, and Yu Rong. Equivariant graph hierarchy-based neural networks. *Advances in Neural Information Processing Systems*, 35:9176–9187, 2022.

Jiaqi Han, Minkai Xu, Aaron Lou, Haotian Ye, and Stefano Ermon. Geometric Trajectory Diffusion Models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 25628–25662. Curran Associates, Inc., 2024. doi: 10.52202/079017-0806.

Tim Hsu, Babak Sadigh, Vasily Bulatov, and Fei Zhou. Score dynamics: Scaling molecular dynamics with picoseconds time steps via conditional diffusion model. *Journal of Chemical Theory and Computation*, 20(6):2335–2348, 2024.

Wenbing Huang, Jiaqi Han, Yu Rong, Tingyang Xu, Fuchun Sun, and Junzhou Huang. Equivariant Graph Mechanics Networks with Constraints. In *International Conference on Learning Representations*, 2022.

Bowen Jing, Bonnie Berger, and Tommi Jaakkola. AlphaFold Meets Flow Matching for Generating Protein Ensembles. In *Forty-first International Conference on Machine Learning*, 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.

Kacper Kapusniak, Cristian Gabellini, Michael Bronstein, Prudencio Tossou, and Francesco Di Giovanni. Mars-fm: Generative modeling of molecular dynamics via markov state models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 19073–19098, 2026.

Leon Klein, Andrew Foong, Tor Fjelde, Bruno Mlodozeniec, Marc Brockschmidt, Sebastian Nowozin, Frank Noe, and Ryota Tomioka. Timewarp: Transferable Acceleration of Molecular Dynamics by Learning Time-Coarsened Dynamics. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 52863–52883. Curran Associates, Inc., 2023.

Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant Graph Attention Transformer for 3D Atomistic Graphs. In *The Eleventh International Conference on Learning Representations*, 2023.

Yang Liu, Jiashun Cheng, Haihong Zhao, Tingyang Xu, Peilin Zhao, Fugee Tsung, Jia Li, and Yu Rong. SEGNO: Generalizing Equivariant Graph Neural Networks with Physical Inductive Biases. In *The Twelfth International Conference on Learning Representations*, 2024.

Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Deep Learning for Universal Linear Embeddings of Nonlinear Dynamics. *Nature Communications*, 9(1):4950, 2018.

Frank Noé, Alexandre Tkatchenko, Klaus-Robert Müller, and Cecilia Clementi. Machine Learning for Molecular Simulation. *Annual review of physical chemistry*, 71(1):361–390, 2020.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International Conference on Machine Learning*, pp. 9323–9332. PMLR, 2021.

Mathias Schreiner, Ole Winther, and Simon Olsson. Implicit Transfer Operator Learning: Multiple Time-Resolution Models for Molecular Dynamics. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 36449–36462. Curran Associates, Inc., 2023.

Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

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. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Liang Shi, Jiarui Lu, Junqi Liu, Chence Shi, Zhi Yang, and Jian Tang. Atomic Trajectory Modeling with State Space Models for Biomolecular Dynamics. *arXiv preprint arXiv:2603.17633*, 2026.

Nima Shoghi, Yuxuan Liu, Yuning Shen, Rob Brekelmans, Pan Li, and Quanquan Gu. Scalable Spatio-Temporal SE (3) Diffusion for Long-Horizon Protein Dynamics. *arXiv preprint arXiv:2602.02128*, 2026.

Oliver T Unke, Stefan Chmiela, Michael Gastegger, Kristof T Schütt, Huziel E Sauceda, and Klaus-Robert Müller. SpookyNet: Learning Force Fields with Electronic Degrees of Freedom and Nonlocal Effects. *Nature communications*, 12(1):7273, 2021.

Fang Wu and Stan Z. Li. DiffMD: A Geometric Diffusion Model for Molecular Dynamics Simulations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(4):5321–5329, Jun. 2023. doi: 10.1609/aaai.v37i4.25663.

Liming Wu, Zhichao Hou, Jirui Yuan, Yu Rong, and Wenbing Huang. Equivariant spatio-temporal attentive graph networks to simulate physical dynamics. *Advances in Neural Information Processing Systems*, 36, 2024.

Minkai Xu, Jiaqi Han, Aaron Lou, Jean Kossaifi, Arvind Ramanathan, Kamyar Azizzadenesheli, Jure Leskovec, Stefano Ermon, and Anima Anandkumar. Equivariant Graph Neural Operator for Modeling 3D Dynamics. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

APPENDIX CONTENTS

A Notation List **15**

B Related Works **15**

B.1 Trajectory-based Learning for Molecular Dynamics 15

C Implementaion Details **16**

C.1 Equivariant encoder and decoder 16

C.2 Optional Momentum-Augmented Training 16

D Error Bound for the Quadratic-Form-Preserving Generator **17**

E Experimental Details **19**

E.1 Datasets 19

E.2 Baselines 22

E.3 Curated MSM-guided flexible-stride dataset 22

F Supplemented Experiments **23**

F.1 Results on MD17 23

F.2 Results on PET-MAD 24

F.3 Parameter sensitivity analysis. 24

F.4 Ablation study. 25

G Limitations **25**

H Broader Impact. **25**

A NOTATION LIST

We provide a complete notation list involved in this paper for reader's reference:

Symbol	Description
$2d$	Total dimension of the latent phase space (q, p)
$z \in \mathbb{R}^{2d}$	The learned latent state vector
$\mathcal{X}$	Molecular configuration space $\mathbb{R}^{3N} \times \mathcal{A}^N$
N	Total number of atoms in a molecular system
$\mathcal{A}$	Finite set of atomic types (atomic numbers)
$G_t = (V, E_t, X, R_t)$	Molecular graph at time t
$V = \{1, \dots, N\}$	Set of nodes representing atoms
$E_t \subseteq V \times V$	Set of edges (interactions) defined by distance or bonding
d_x	Dimension of atomic feature embeddings
$X \in \mathbb{R}^{N \times d_x}$	Matrix of node-wise atomic features (e.g., embeddings of $\mathcal{A}$)
$R_t \in \mathbb{R}^{N \times 3}$	Matrix of Cartesian coordinates for all N atoms
$\Phi^\tau : \mathcal{X} \to \mathcal{X}$	Nonlinear flow map for a lag time τ
Δt	Fundamental integration step of the MD engine
ℓ	Training time-stride (number of integration steps between molecular pairs)
ℓ_{test}	Testing time-stride
$\tau = \ell \Delta t$	Training lag time
$\tau_{\text{test}} = \ell_{\text{test}} \Delta t$	Testing lag time
$\phi : \mathcal{X} \to \mathbb{R}^{2d}$	E(3)-equivariant encoder mapping graph to latent space
$2d$	Latent dimension
h_i, h_j	Scalar node features at nodes i and j
m_{ij}	Message computed along edge (i, j)
$r_{ij} = r_i - r_j$	Relative displacement vector between atoms i and j
$\mathcal{N}(i)$	Neighborhood of node i in the graph
L	Number of layers in the equivariant encoder
$z \in \mathbb{R}^{2d}$	Latent state vector in $\mathbb{R}^{2d}$
$\psi : \mathbb{R}^{2d} \to \mathcal{X}$	Neural decoder reconstructing the molecular graph $\hat{G}$
$\mathcal{K}^\tau$	Infinite-dimensional Koopman operator for lag time τ
$\mathcal{A}_{op}$	Infinitesimal Koopman generator (operator level)
$A \in \mathbb{R}^{2d \times 2d}$	Finite-dimensional linear Koopman generator matrix
$J \in \mathbb{R}^{2d \times 2d}$	Fixed skew-symmetric block matrix $\begin{pmatrix} 0 & I_d \\ -I_d & 0 \end{pmatrix}$
$S \in \mathbb{R}^{2d \times 2d}$	Symmetric matrix defining the preserved quadratic form
$H(z) = \frac{1}{2} z^\top S z$	Learned quadratic latent invariant
e^{At}	Matrix exponential for continuous-time latent propagation
$\alpha \in \mathbb{R}$	Temporal scaling factor for flexible time-stride simulation
L_ψ	Lipschitz constant of the decoder ψ
ε	Upper bound on the single-stride latent approximation error δ_k

B RELATED WORKS

B.1 TRAJECTORY-BASED LEARNING FOR MOLECULAR DYNAMICS

The simulation of molecular trajectories via deep learning has evolved through three primary paradigms, distinguished by their mathematical representation of the dynamical flow.

Equivariant Integrators and Simulators This category focuses on modeling the transition of molecular coordinates in the physical space. Methods such as EGNN (Satorras et al., 2021) and GMN (Huang et al., 2022) provide the geometric foundations for these models. Neural integrators like SEGNO (Liu et al., 2024) and NCGNN (Guo et al., 2023) parameterize the infinitesimal change (vector field), while "jump" simulators like TimeWarp (Klein et al., 2023) and FlashMD (Bigi et al., 2025) utilize GNN-based regressors to map configurations across large temporal strides. While efficient, these methods typically model nonlinear trajectories directly, which can lead to numerical instability and "unphysical" drift when applied recursively over long horizons.

Equivariant Neural Operators Rather than predicting individual coordinate changes, neural operators seek to learn the underlying transformation of the system's functions. Implicit Transfer Operator (ITO) learning (Schreiner et al., 2023) and Equivariant Graph Neural Operators (EGNO) (Xu et al., 2024) represent this shift toward learning the global flow or Koopman-style transitions. More recently, state-space models (SSMs) have been adapted for atomic trajectories (Shi et al., 2026). Our **FLEXMD** framework advances this line of work by introducing a structure-preserving constraint: we require the learned Koopman generator to follow a quadratic-constrained form ($A = JS$). Under the stated symmetry and definiteness conditions, this parameterization preserves a quadratic form in latent space and provides a continuous-time representation with controlled latent amplification.

Equivariant Generative Trajectory Models A third paradigm treats MD as a distribution modeling task, aiming to sample the equilibrium ensemble or transition paths. Diffusion-based frameworks such as DiffMD (Wu & Li, 2023), GeoTDM (Han et al., 2024), and Score Dynamics (Hsu et al., 2024) learn to reverse a noise process to generate physical configurations. Simultaneously, autoregressive models like ProAR (Cheng et al., 2026) and ConfRover (Shen et al., 2025) scale these concepts to biomolecular dynamics. While these models excel at capturing the stochasticity of thermodynamic ensembles, they are often computationally intensive and lack the deterministic latent structure required for precise physical interpolation.

C IMPLEMENTAION DETAILS

C.1 EQUIVARIANT ENCODER AND DECODER

We use EGNN-style message passing (Satorras et al., 2021) for the encoder and decoder. For an input graph $G = (V, E, X, R)$, let x_i denote the atom identity or features at node i. The encoder initializes the scalar node state as $h_i^{(0)} = \mathrm{Emb}(x_i)$ and the coordinate state as $r_i^{(0)} = r_i$. At layer l, the edge message depends on the squared interatomic distance:

$$d_{ij}^{(l)} = \|r_i^{(l)} - r_j^{(l)}\|_2^2, \qquad m_{ij}^{(l)} = \varphi_e^{(l)}\big(h_i^{(l)}, h_j^{(l)}, d_{ij}^{(l)}\big), \tag{5}$$

where $\varphi_e^{(l)}$ is an MLP producing invariant edge features. The updates are

$$h_i^{(l+1)} = h_i^{(l)} + \varphi_h^{(l)}\left(h_i^{(l)}, \sum_{j \in \mathcal{N}(i)} m_{ij}^{(l)}\right), \tag{6}$$

$$r_i^{(l+1)} = r_i^{(l)} + \sum_{j \in \mathcal{N}(i)} (r_i^{(l)} - r_j^{(l)})\varphi_x^{(l)}\big(m_{ij}^{(l)}\big). \tag{7}$$

The scalar update uses invariant inputs, while the coordinate increment is a scalar-weighted sum of relative vectors. Hence scalar features remain invariant, and coordinates remain equivariant under rigid rotations and translations. With a distance-based, permutation-consistent edge construction, these properties hold across layers. We use residual updates and keep the constructed cutoff graph fixed across layers. A permutation-invariant sum of the final scalar features, followed by an MLP, gives the latent state:

$$z = \phi(G) = \mathrm{MLP}_\phi\left(\sum_{i=1}^{N} h_i^{(L)}\right) \in \mathbb{R}^{2d}. \tag{8}$$

The coordinate stream influences the readout through the distances in the messages but is not concatenated to it. Therefore, z is invariant to atom permutations and global rigid transformations.

The decoder receives the latent features for prediction. For a rollout initialized at G_0, the decoder uses the initial edge index and edge features, including distances computed from R_0, at each prediction time. No ground-truth geometry at a future time is supplied during prediction.

C.2 OPTIONAL MOMENTUM-AUGMENTED TRAINING

When the dataset provides momentum features, we extend our implementation with the supervision on momentum. The key difference is that the $2d$ latent features will be explicitly partitioned into

two d-dimensional blocks,

$$z = \begin{bmatrix} q \\ p \end{bmatrix}, \qquad q, p \in \mathbb{R}^d, \tag{9}$$

where q is the latent position-like component and p is the latent momentum-like component. This is a phase-space parameterization of the learned representation: q and p are not required to equal Cartesian positions and physical momenta. In particular, when only configurations are observed, both blocks are inferred jointly by the encoder from (X, R); no unobserved physical momentum is imputed.

We parameterize the canonical matrix using this ordering,

$$J = \begin{bmatrix} 0 & I_d \\ -I_d & 0 \end{bmatrix}, \qquad S = \begin{bmatrix} S_{qq} & S_{qp} \\ S_{qp}^\top & S_{pp} \end{bmatrix}, \qquad S = S^\top, \tag{10}$$

so that the latent dynamics are

$$\dot{q} = S_{qp}^\top q + S_{pp} p, \tag{11}$$

$$\dot{p} = -S_{qq} q - S_{qp} p. \tag{12}$$

The symmetric matrix S is learned directly through an unconstrained matrix B and the positive-definite parameterization $S = BB^\top + \epsilon_S I$, with a small fixed $\epsilon_S > 0$. During training, a pair $(G_t, G_{t+\tau})$ is encoded as z_t and $z_{t+\tau}$, and the predicted target is $\widehat{z}_{t+\tau} = \exp(A\tau) z_t$, where $A = JS$. The loss is

$$\mathcal{L} = \lambda_{\text{rec}} \mathcal{L}_{\text{rec}} + \lambda_{\text{dyn}} \left\| z_{t+\tau} - \exp(A\tau) z_t \right\|_2^2, \tag{13}$$

with optional multi-stride terms formed by replacing τ with $k\tau$. Because the dynamics loss acts on the concatenated vector, it supervises both the position and momentum blocks; it may equivalently be written as the sum of the two block losses $\|q_{t+\tau} - \widehat{q}_{t+\tau}\|^2$ and $\|p_{t+\tau} - \widehat{p}_{t+\tau}\|^2$.

For datasets that provide physical atomic momenta P_t, we append them as rotation-equivariant, translation-invariant vector node features and let the decoder reconstruct (R_t, P_t). We first mass-scale each momentum as $p_i/\sqrt{m_i}$ and then apply dataset-level standardization; $\bar{P}$ denotes the resulting normalized momenta. Both transformations are inverted for reporting. The corresponding reconstruction term is

$$\mathcal{L}_{\text{rec}} = \lambda_R \|R_t - \widehat{R}_t\|_2^2 + \lambda_P \|\bar{P}_t - \widehat{\bar{P}}_t\|_2^2. \tag{14}$$

Energy conservation filter. For completeness, when physical momenta are available we optionally apply the energy-conservation filter described in FlashMD (Bigi et al., 2025). This filter is an inherited post-processing procedure. Let (R_t, P_t) be the input state and $(R'_{t+\tau}, P'_{t+\tau})$ the raw prediction after one learned step. An energy model evaluates

$$E_t = K(P_t) + V(R_t), \qquad E'_{t+\tau} = K(P'_{t+\tau}) + V(R'_{t+\tau}), \qquad K(P) = \sum_i \frac{\|p_i\|_2^2}{2m_i}. \tag{15}$$

We rescale all predicted momenta by a common factor,

$$P_{t+\tau} \leftarrow \alpha P'_{t+\tau}, \qquad \alpha = \sqrt{1 - \frac{E'_{t+\tau} - E_t}{K(P'_{t+\tau})}}. \tag{16}$$

Because kinetic energy is quadratic in momentum, this correction makes the total energy equal to E_t. The filter is applied only when the radicand is nonnegative and the predicted kinetic energy is nonzero; otherwise, the raw prediction is retained. It requires one or two additional energy evaluations per step and therefore introduces computational overhead. The filter does not modify the latent quadratic form parameterization or the training objective.

D Error Bound for the Quadratic-Form-Preserving Generator

Proposition D.1 (Quadratic-Invariant–Induced Error Control)**.** *Let $\Phi^t : \mathcal{X} \to \mathcal{X}$ denote the observed molecular dynamics flow, and let $\phi : \mathcal{X} \to \mathbb{R}^{2d}$ and $\psi : \mathbb{R}^{2d} \to \mathcal{X}$ be the encoder and decoder, respectively. Consider latent dynamics governed by*

$$\frac{dz}{dt} = Az, \quad A = JS,$$

where J is the fixed skew-symmetric block matrix and $S \in \mathbb{R}^{2d\times 2d}$ is symmetric positive definite. Define the latent quadratic form

$$V(z) := \tfrac{1}{2} z^\top S z.$$

For each nonnegative integer k, let $z_k^* := \phi(\Phi^{k\tau}(G_0))$ denote the encoded true latent state after k prediction strides, and let $\hat{z}_k := e^{Ak\tau}\phi(G_0)$ denote the predicted latent state. Thus, $z_0^* = \hat{z}_0 = \phi(G_0)$ and $e_0 = 0$. Define the latent error $e_k := \hat{z}_k - z_k^*$ and

$$\delta_k := z_{k+1}^* - e^{A\tau} z_k^*.$$

Assume that:

1. *the decoder ψ is Lipschitz continuous with constant L_ψ;*
2. *there exists $\varepsilon > 0$ such that $\|\delta_k\|_S \le \varepsilon$ for all k, where $\|u\|_S := \sqrt{u^\top S u}$;*
3. *$S \succ 0$ with smallest eigenvalue $\mu > 0$.*

Then, for any integer $n \ge 1$,

$$\|e_n\|_S \le \sum_{i=0}^{n-1} \|\delta_i\|_S \le n\varepsilon,$$

and consequently,

$$\|e_n\|_2 \le \frac{1}{\sqrt{\mu}} \sum_{i=0}^{n-1} \|\delta_i\|_S, \qquad \text{and, if } \|\delta_i\|_2 \le \varepsilon_2, \quad \|e_n\|_2 \le Cn\varepsilon_2, \quad C := \sqrt{\frac{\lambda_{\max}(S)}{\mu}},$$

$$\|\psi(\hat{z}_n) - \Phi^{n\tau}(G_0)\| \le \frac{L_\psi}{\sqrt{\mu}} \sum_{i=0}^{n-1} \|\delta_i\|_S + r_n,$$

where $r_n := \|\psi(z_n^*) - \Phi^{n\tau}(G_0)\|$ is the decoded reconstruction error.

Moreover, if $\mathbb{E}[\delta_k] = 0$, $\mathbb{E}[\delta_i \delta_j^\top] = 0$ for $i \ne j$, and $\mathbb{E}\|\delta_k\|_S^2 \le \sigma^2$, then

$$\mathbb{E}\|e_n\|_S^2 \le n\sigma^2.$$

Consequently, since $\mu = \lambda_{\min}(S) > 0$,

$$\mathbb{E}\|e_n\|_2^2 \le \frac{n\sigma^2}{\mu}, \qquad \mathbb{E}\|e_n\|_2 \le \sigma\sqrt{\frac{n}{\mu}}.$$

Proof. We first show that the quadratic form $V(z) = \frac{1}{2} z^\top S z$ is invariant under the flow generated by $A = JS$. For $\dot{z} = Az = JSz$, we have

$$\frac{d}{dt} V(z) = z^\top S \dot{z} = z^\top S(JSz) = z^\top (SJS) z.$$

Since $S^\top = S$ and $J^\top = -J$, it follows that $(SJS)^\top = -SJS$, so SJS is skew-symmetric. Hence $z^\top (SJS) z = 0$, and therefore $V(z)$ is conserved along trajectories:

$$V(e^{At} z) = V(z), \quad \forall t \ge 0.$$

Because $S \succ 0$, its eigenvalues are positive and satisfy

$$\mu \|z\|_2^2 \le z^\top S z \le \lambda_{\max}(S) \|z\|_2^2.$$

Thus,

$$\|e^{At} z\|_S = \|z\|_S.$$

Moreover, $\sqrt{\mu}\|z\|_2 \le \|z\|_S \le \sqrt{\lambda_{\max}(S)}\|z\|_2$, so

$$\|e^{At}\|_2 \le C := \sqrt{\frac{\lambda_{\max}(S)}{\mu}}.$$

Next, define the latent error $e_k = \hat{z}_k - z_k^*$. By construction,

$$z_{k+1}^* = e^{A\tau} z_k^* + \delta_k, \quad \hat{z}_{k+1} = e^{A\tau} \hat{z}_k,$$

which yields the recursion

$$e_{k+1} = e^{A\tau} e_k - \delta_k.$$

Unrolling this recursion gives

$$e_n = -\sum_{i=0}^{n-1} e^{A(n-1-i)\tau} \delta_i.$$

Taking the S-norm and using the isometry property gives

$$\|e_n\|_S \le \sum_{i=0}^{n-1} \|e^{A(n-1-i)\tau} \delta_i\|_S = \sum_{i=0}^{n-1} \|\delta_i\|_S.$$

The Euclidean bound follows from $\|u\|_2 \le \|u\|_S / \sqrt{\mu}$.

By Lipschitz continuity of ψ,

$$\begin{aligned} \|\psi(\hat{z}_n) - \Phi^{n\tau}(G_0)\| &\le \|\psi(\hat{z}_n) - \psi(z_n^*)\| + r_n \\ &\le L_\psi \|e_n\|_2 + r_n \\ &\le \frac{L_\psi}{\sqrt{\mu}} \sum_{i=0}^{n-1} \|\delta_i\|_S + r_n. \end{aligned} \tag{17}$$

Using $\|\delta_i\|_S \le \varepsilon$ gives the deterministic bound.

□

E Experimental Details

E.1 Datasets

We provide the complete dataset and evaluation protocol used in the main experiments. Test trajectories are excluded from training, normalization fitting, model selection, and MSM construction.

MD17. MD17 contains ab initio molecular-dynamics trajectories generated with density-functional theory for eight molecules: aspirin, benzene, ethanol, malonaldehyde, naphthalene, salicylic acid, toluene, and uracil. We use the following protocol:

- **Splits:** For each molecule, state–future-state pairs are randomly divided into 500 training, 2,000 validation, and 2,000 test examples, following EGNO (Xu et al., 2024).
- **Time stride and trajectory length:** The default prediction window is $\Delta T = 3{,}000$ integration steps. The flexible-stride experiment uses $\ell_{\text{test}} \in \{1, 5, 10, 25, 100, 1{,}000, 3{,}000\}$, and the stability experiment rolls out for 100 ps.
- **Independent runs and seeds:** We trained both the baseline models and our model independently ten times on the MD17 dataset and evaluated each resulting model separately. Unless otherwise specified, we used 42 as the random seed for all experiments on the datasets described below.
- **Preprocessing and representation:** Following prior training protocols (Xu et al., 2024), all methods receive atomic species and Cartesian coordinates and does not use forces as an auxiliary target. The comparison follows the reported heavy-atom convention.
- **Baselines and adaptations:** RF, TFN, SE(3)-Transformer, EGNN, EGNO, ESTAG, MDGen, FlashMD, and ConfRover are evaluated using their native discrete- or generative-prediction interfaces. FlashMD is evaluated with its momentum-reduced configuration for fair comparison.

- **Training, hyperparameters, and resources:** The recorded FLEXMD configuration uses latent dimension $2d = 64$, batch size 128, and a nominal budget of 2,000 epochs. Training takes approximately 0.5 hours per molecule on eight NVIDIA GeForce RTX 3090 GPUs. The learning rate is 0.00005. The weight decay is set 0.0001, and the dropout rate is set to 0.5.

AdK (Adenylate Kinase). AdK is the apo adenylate-kinase equilibrium trajectory used by EGNO. It was simulated with the CHARMM27 force field in explicit water and ions under NPT conditions at 300 K and 1 bar. We use the following protocol:

- **Splits:** The processed data contain 2,481 training, 827 validation, and 878 testing sub-trajectories. The dataset does not contain enough samples to evaluate time strides greater than $1{,}000$.
- **Time step and trajectory length:** Metadata frames are separated by 240 ps and cover approximately 1.004 μs.
  - The training is using 2,481 sub-trajectories with 5 time-strides, while the last 5 frames are discard.
  - The evaluation on time-stride $5, 10, 25$, and 100 uses the first 878 sub-trajectories, while the last $n, n = \{5, 10, 25, 100\}$ frames are discard.
- **Independent runs:** We trained both the baseline models and our model independently ten times on the AdK dataset and evaluated each resulting model separately.
- **Preprocessing and representation:** We use the processed full atomic representation and construct the interaction graph with a 8 Å cutoff following ENGO (Xu et al., 2024). Coordinates are centered consistently before equivariant encoding; no test frames are used to fit preprocessing statistics.
- **Baselines and adaptations:** All baselines are trained with their default and recommended configurations. FlashMD is evaluated with its momentum-removed configuration for fair comparison.
- **Training, hyperparameters, and resources:** The recorded FLEXMD configuration uses latent dimension $2d = 256$, batch size 1, and a nominal budget of 1000 epochs. Training takes approximately 25 hours on eight RTX 3090 GPUs. The learning rate is 0.00005. The weight decay is set 0.0001, and the dropout rate is set to 0.5.

ATLAS. ATLAS contains three independent 100-ns explicit-solvent, all-atom trajectories for each of 1,390 structurally diverse proteins. We use the following MDGen-aligned protocol:

- **Splits:** Proteins are split by PDB release date into 1,265 training, 39 validation, and 82 test proteins (Jing et al., 2024b;a). All three replicas of a protein remain in the same split.
- **Time stride and trajectory length:** Each reference trajectory is 100 ns. Training examples subsample every 40 frames into 250-frame targets; the MDGen-style saved-frame interval is 10 ps. FLEXMD is trained at $\ell = 40$ for the common-stride experiment and at $\ell = 120$ for the flexbile inference experiment, then evaluated at $\ell = 40$.
- **Independent runs:** Due to the expensive training on ATLAS, we trained the model only once and performed 10 independent inference runs.
- **Preprocessing and representation:** A protein and one replica are sampled during training; random crops contain at most 256 residues. At inference, full proteins are evaluated. Generated 250-frame trajectories are compared against pooled reference frames from all three replicas.
- **Baselines and adaptations:** MDGen uses its 250-frame protein protocol. AlphaFlow and MSA subsampling follow the published ATLAS comparison.
- **Training, hyperparameters, and resources:** The recorded FLEXMD configuration uses latent dimension $d = 16$, batch size 1, and a nominal budget of 200 epochs. Training takes approximately 84 hours on eight RTX 3090 GPUs. The learning rate is 0.00005. The weight decay is set 0.0001, and the dropout rate is set to 0.5.

MD-CATH. MD-CATH contains 5,398 protein domains with frames saved every 1 ns. Domains range from 50 to 500 residues, with mean length 137 residues, and each domain has five temperatures and five replicas per temperature. We use the following protocol:

- **Splits:** For the train/validation/test split, we follow the previous works and use MMSEQS2 to exclude test domains sharing more than 20% sequence identity with any training or validation domain (Kapusniak et al., 2026). The final split is 4,304 training, 538 validation, and 495 test domains.
- **Time step and trajectory length:** Frames are sampled every 1 ns. The average replica length is 464 ns and most replicas are 500 ns long. We primarily evaluate the 450 K subset, where partial unfolding provides greater conformational diversity.
- **Independent runs:** Each domain has five replicas at each of five temperatures, giving 25 trajectories per domain. Due to the expensive training on MD-CATH, we trained the model only once and performed 10 independent inference runs.
- **Preprocessing and MSM construction:** For MARS-FM, per-domain MSMs are built by k-means clustering standardized radius of gyration and secondary-structure fractions; transition matrices are symmetrized post hoc. Generated and reference ensembles use the same alignment, observable definitions, histogram bins, and free-energy conventions.
- **Baselines and adaptations:** As we precisely following the training protocol presented in (Kapusniak et al., 2026), the results of baselines are reported from the source paper.
- **Training, hyperparameters, sampling, and resources:** The training of FLEXMD takes approximately 3–4 days on eight RTX 3090 GPUs for 500 epochs, and sampling 500 conformations for all test domains takes approximately 5 hours. The recorded FLEXMD configuration uses latent dimension $d = 256$, batch size 1. The learning rate is 0.00005. The weight decay is set 0.0001, and the dropout rate is set to 0.5.

PET-MAD. PET-MAD is the universal reference potential for the materials and molecular long-stride benchmark, rather than a learned trajectory-generation baseline. Its underlying MAD corpus contains 95,595 structures spanning 85 elements and eight chemically diverse subsets. We use the following protocol:

- **Splits:** PET-MAD itself was trained after separately shuffling each MAD subset and selecting 80%/10%/10% training/validation/test structures. For the FlashMD universal trajectory construction, 10,000 MAD structures are sampled at random; the exact held-out subset used by the present experiment should be recorded with the released data manifest.
- **Time step and trajectory length:** Each structure is equilibrated for 10 ps in NVT with a 0.5 fs step, followed by 2.5 ps of NVE production with a 0.25 fs reference step. Following the FlashMD (Bigi et al., 2025), the target learned strides are 1, 4, and 16 fs.
- **Independent runs and temperatures:** Due to the large scale of PET-MAD (approximately 28.2 GB), we trained the model only once and instead performed multiple independent inference runs (Bigi et al., 2025). Each initial structure is simulated ten times at temperatures sampled uniformly from 0–1,500 K. Positions and momenta are recorded at every reference step; examples are subsampled every 500 fs and augmented with time reversal, giving approximately one million training structures in the FlashMD construction.
- **Preprocessing and reference generation:** Structures are BFGS-relaxed until the maximum force component is below 0.01 eV/Å. PET-MAD provides the reference energy and force field. The benchmark covers inorganic materials, surfaces, molecular crystals, and molecules.
- **Baselines and adaptations:** FlashMD and FLEXMD uses its momentum-augmented formulation. Flash MD (n fs)and FLEXMD (n fs) ($n = 1, 4, 16$) are separate models trained on separate time-strides. While the FLEXMD * is one unified model trained across different time-strides. And following the FlahsMD, the energy model is implemented for energy conservation enhancement. And we use the classical force field to isolate the effect of the learned trajectory model from the underlying potential.

- **Parameters, hyperparameters, and resources:** The training of FLEXMD takes approximately 14 days on eight RTX 3090 GPUs for 10 epochs. The recorded FLEXMD configuration uses latent dimension $d = 256$, 9 layers of encoder, and batch size 1. The learning rate is 0.00005. The weight decay is set 0.0001, and the dropout rate is set to 0.5.

E.2 BASELINES

We use published implementations and evaluation conventions whenever available. Models are retrained on the corresponding training split unless a method is explicitly inference-only.

Equivariant graph predictors. RF is a random-feature regression baseline. TFN uses tensor-field convolutions to construct rotation-equivariant features, EGNN uses E(3)-equivariant message passing for discrete-time prediction, and the SE(3)-Transformer uses SE(3)-equivariant attention over graph neighborhoods. EGHN augments equivariant message passing with a hierarchy formulation and is used primarily on AdK.

Neural operators and spatiotemporal models. EGNO extends EGNN with temporal convolutions and Fourier-mode mixing, allowing a sequence of future states to be predicted from a graph representation. ESTAG models spatiotemporal dependencies with graph-attention blocks. For EGNO, we use the published MD17 and AdK settings, including the reported temporal-point counts and early stopping; other baselines follow their released configurations.

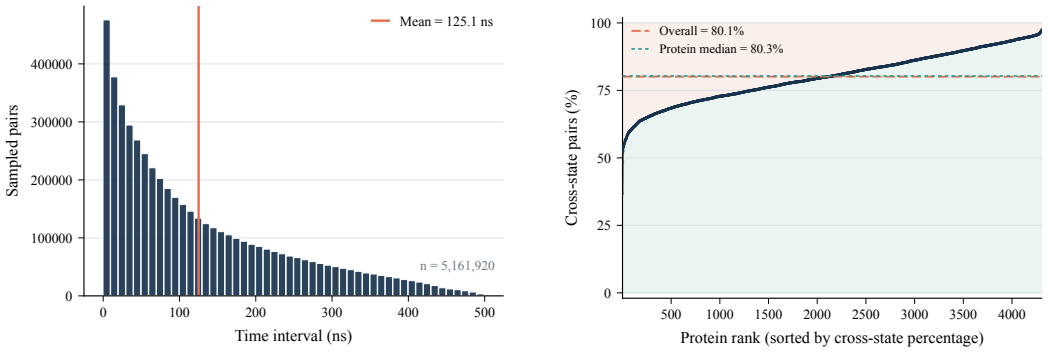
Generative trajectory models. MDGen is a conditional diffusion model that generates a trajectory window from a conditioning conformation and can be chained autoregressively. FlashMD directly predicts positions and mass-scaled momenta at a fixed long stride; on PET-MAD, its reference trajectories are generated with the PET-MAD potential. ConfRover is a conditional autoregressive model for protein conformational trajectories. AlphaFlow uses flow matching to generate protein ensembles and trajectories.

MSM and pretrained protein baselines. MARS-FM learns transitions between metastable states of per-domain Markov state models and generates long-horizon samples by hierarchical state-space sampling. We report direct MARS-FM sampling and the hybrid MARS-FM⇓MDGen configuration, in which MARS-FM proposes long-timescale states and MDGen refines transitions at a shorter timescale. BioEmu is an inference-only pretrained protein ensemble model. It uses a backbone-only representation and was trained with additional, partly proprietary data; therefore, it is included as a reference comparison rather than a controlled train-from-scratch baseline.

E.3 CURATED MSM-GUIDED FLEXIBLE-STRIDE DATASET

To complement the dense chronological supervision described in the main text, we curate a second MD-CATH training corpus using the MSM-guided sampling principle of MARS-FM (Kapusniak et al., 2026). We use the 450 K trajectories, for which MD-CATH provides one saved frame per nanosecond. For each protein, a reversible MSM is estimated from the discretized trajectories using ten states and a 50-frame (50 ns) MSM lag. Source states are selected in a state-balanced fashion, while destination states are sampled from the corresponding row of the MSM transition matrix. We then select a source and a destination frame from the same replica. Unlike the original MARS-FM pair representation, we retain both frame indices and define the physical separation explicitly: we retain pairs with $i_{\text{target}} > i_{\text{source}}$ and set $\tau = (i_{\text{target}} - i_{\text{source}})\,\text{ns}$. Consequently, the MSM lag controls state-transition estimation, whereas the recorded Δt describes the actual temporal separation of each selected pair.

The resulting curated set contains 5,161,920 source–target pairs. The mean physical separation is 125.082 ns, substantially larger than the 1 ns chronological training stride. Of all sampled pairs, 80.091% connect distinct MSM states, while the remaining pairs remain within the same state. This composition provides frequent long-range conformational supervision without discarding the within-state examples needed to learn local structural variation. Because source and target frames are sampled from the same replica, the recorded separation is well-defined; however, these pairs should be interpreted as MSM-guided temporal separations rather than necessarily as observed one-step transitions along the original trajectory.



(a) Physical time intervals between source and target frames. The mean separation is 125.082 ns over 5,161,920 pairs.

(b) Cross-state pair fraction by protein rank. Protein identifiers are omitted; horizontal lines show the pair-weighted overall and median protein-level fractions.

Figure 3: Statistics of the MSM-guided flexible-stride training pairs.

Table 5: Long-horizon prediction error on MD17: Final Time-stride conformation MSE ($\times 10^{-2}$)

Model	Aspirin	Benzene	Ethanol	Malonaldehyde	Naphthalene	Salicylic	Toluene	Uracil
RF	10.94± 0.01	103.72± 1.29	4.64± 0.01	13.93± 0.03	0.50± 0.01	1.23± 0.01	10.93± 0.04	0.64± 0.01
TFN	12.37± 0.18	58.48± 1.98	4.81± 0.04	13.62± 0.08	0.49± 0.01	1.03± 0.02	10.89± 0.01	0.84± 0.02
SE(3)-Tr.	11.12± 0.06	68.11± 0.67	4.74± 0.13	13.89± 0.02	0.52± 0.01	1.13± 0.02	10.88± 0.06	0.79± 0.02
EGNN	14.41± 0.15	62.40± 0.53	4.64± 0.01	13.64± 0.01	0.47± 0.02	1.02± 0.02	11.78± 0.07	0.64± 0.01
EGNO	9.18± 0.06	48.85± 0.55	4.62± 0.01	12.80± 0.02	0.37± 0.02	0.86± 0.02	10.21± 0.05	0.52± 0.02
ESTAG	8.25± 0.04	42.40± 0.62	5.29± 0.04	11.74± 0.03	0.28± 0.09	0.96± 0.05	8.35± 0.01	0.74± 0.04
MDGen	7.29± 0.09	49.78± 0.78	4.39± 0.08	13.58± 0.10	0.63± 0.08	0.83± 0.03	7.37± 0.02	0.82± 0.08
FlashMD	5.92± 0.09	40.85± 0.96	3.75± 0.06	12.85± 0.04	0.27± 0.06	0.85± 0.07	8.32± 0.03	0.62± 0.02
ConfRover	6.58± 0.12	39.25± 0.96	4.75± 0.01	12.28± 0.09	0.38± 0.10	0.75± 0.05	7.856± 0.06	0.52± 0.01
FLEXMD	**4.98**± 0.06	**39.24**± 0.67	**3.24**± 0.02	**10.24**± 0.03	**0.24**± 0.01	**0.36**± 0.03	**6.29**± 0.05	**0.23**± 0.03

Figure 3(a) shows the distribution of physical time intervals. The distribution is broad, reflecting the state-balanced sampling procedure rather than a fixed training lag. Figure 3(b) shows the protein-level cross-state fractions, with proteins sorted only for visualization and their identifiers omitted. The light shading is defined by the protein-level curve and separates the regions below and above each protein's cross-state fraction.

F Supplemented Experiments

F.1 Results on MD17

On MD17, FLEXMD consistently achieves the lowest final-step MSE across all molecules, demonstrating strong accuracy in long-horizon prediction as shown in Table 5. The improvement is particularly pronounced on flexible and complex molecules (e.g., aspirin, malonaldehyde, salicylic), indicating that the learned Koopman latent space effectively linearizes nonlinear dynamics while preserving geometric structure. Compared to equivariant predictors and operator-based methods,

Table 6: Trajectory stability on MD17: Maximum stable simulation time (ps) within 100 ps rollout.

Model	Aspirin	Benzene	Ethanol	Malonaldehyde	Naphthalene	Salicylic	Toluene	Uracil
EGNO	10.25± 6.21	12.58± 10.78	21.25± 4.69	19.25± 7.10	29.13± 4.79	16.82± 3.99	23.83± 2.92	12.58± 2.94
MDGen	25.30± 5.95	28.39± 9.09	22.40± 4.46	41.30± 3.86	39.46± 2.78	26.95± 3.52	31.94± 4.08	26.92± 2.43
FlashMD	44.45± 2.71	50.72± 3.60	43.57± 4.87	44.92± 4.33	35.41± 2.66	48.99± 3.81	44.60± 2.13	59.58± 4.98
ConfRover	26.83± 1.62	50.17± 11.35	5.39± 2.93	13.40± 4.74	34.84± 4.77	37.90± 5.65	20.04± 3.95	26.12± 4.11
FLEXMD	**63.30**± 1.64	**60.93**± 6.33	**50.95**± 2.58	**58.30**± 3.84	**53.19**± 1.30	**68.84**± 3.16	**52.85**± 2.91	**79.95**± 2.60

Table 7: Absolute temperature error $|\Delta T|$ (K) in PET-MAD NVT simulations (lower is better). Results are reported as mean $\pm$ standard deviation.

Time-stride	Model	W/O. energy conservation		W. energy conservation	
		Langevin	SVR	Langevin	SVR
1 fs	FlashMD (1 fs)	35.50± 1.90	7.30± 0.91	0.20± 0.71	0.60± 0.51
	FLEXMD (1 fs)	**23.80**± 0.76	**5.30**± 0.37	**0.20**± 0.20	**0.40**± 1.09
	FLEXMD *	27.23± 0.34	6.40± 0.92	**0.20**± 0.46	0.50± 0.82
4 fs	FlashMD (4 fs)	18.54± 1.42	10.20± 1.20	0.60± 2.02	0.50± 0.69
	FLEXMD (4 fs)	15.20± 0.67	7.30± 0.74	0.53± 0.90	**0.40**± 0.68
	FLEXMD *	**14.50**± 0.55	**6.50**± 0.84	**0.40**± 0.54	**0.40**± 1.10
16 fs	FlashMD (16 fs)	19.40± 1.88	5.40± 1.75	0.40± 1.06	**0.20**± 1.08
	FLEXMD (16 fs)	18.60± 1.23	4.90± 1.62	0.40± 0.97	**0.20**± 0.93
	FLEXMD *	**15.42**± 0.90	**3.90**± 0.93	**0.30**± 0.70	**0.20**± 0.98

* FlashMD/FLEXMD (n fs) is trained solely on the dataset sampled at n fs; accordingly, the three rows use $n \in \{1, 4, 16\}$ fs. FLEXMD * is trained jointly on all available time-steps. Both FLEXMD variants receive momenta as inputs and use the energy filter described in the FlashMD protocol.

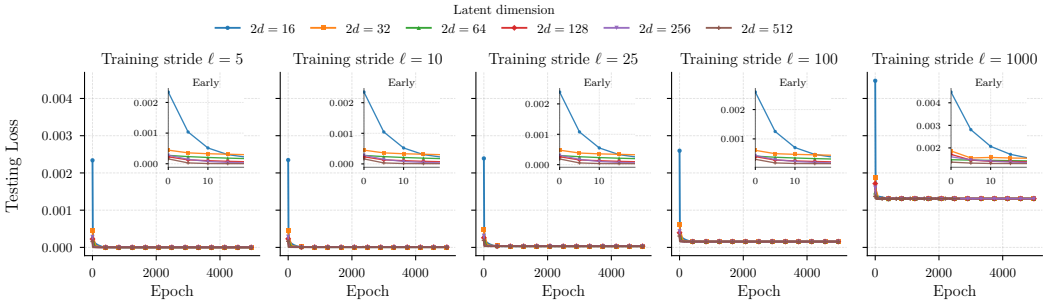


Figure 4: Testing loss vs. training epochs on MD17 (aspirin) under varying latent dimensions ($2d$) and training time-strides (τ).

FLEXMD reduces error accumulation, while also outperforming generative models that prioritize distributional modeling over precise trajectory tracking.

In terms of stability, FLEXMD significantly extends the duration of simulations that remain structurally connected, outperforming all baselines by a large margin across every molecule (Table 6). This highlights the benefit of enforcing quadratic structure in the latent dynamics, thereby mitigating drift and structural collapse over time. Combined with its continuous-time formulation, FLEXMD enables both accurate and stable long-horizon rollouts, supporting the usefulness of the proposed generator constraint for long-horizon rollout.

F.2 RESULTS ON PET-MAD

F.3 PARAMETER SENSITIVITY ANALYSIS.

Figure 4 presents the testing loss as a function of training epochs for different latent dimensions ($2d$) and training time strides (τ) on MD17 (aspirin). Increasing the latent dimension generally improves performance: models with $2d \geq 128$ achieve lower asymptotic error and faster convergence, although the gains diminish at larger dimensions. Larger time strides also make optimization more difficult and increase the final error. These results illustrate the trade-off between temporal resolution and model capacity while showing that FLEXMD remains robust across a broad range of configurations.

F.4 ABLATION STUDY.

We conduct an ablation study to isolate the impact of the quadratic-form-preserving generator constraint in the Koopman generator; results are reported in Table 8. At small test strides (5 and 10), the unconstrained and constrained Koopman variants achieve comparable errors, indicating that structural constraints are less critical for short-horizon prediction. As the test stride increases, however, the difference becomes pronounced: the unconstrained Koopman model degrades rapidly and becomes numerically unstable (denoted by "–") at test strides of 1,000 and 3,000, reflecting poor extrapolation and insufficient dynamical regularity.

In contrast, the constrained Koopman model maintains finite errors across all test strides and substantially outperforms the unconstrained variant in the long-horizon regime. Although it incurs slightly higher error at some moderate shifts, its stability at 1,000 and 3,000 steps demonstrates the value of the generator constraint for regularizing the learned operator and enabling reliable continuous-time extrapolation beyond the training regime.

Table 8: Effect of the quadratic-form-preserving generator constraint on time-stride generalization: MSE under varying training and test strides (lower is better). "–" denotes numerically unstable predictions.

Test Δt	Unconstrained Koopman		Constrained Koopman	
	Train Δt=5	Train Δt=10	Train Δt=5	Train Δt=10
5	1.88e-06± 0.09	1.92e-06± 0.46	**1.88e-06**± 0.81	1.96e-06± 0.47
10	7.18e-06± 0.21	**7.11e-06**± 0.55	7.83e-06± 0.41	8.16e-06± 0.50
25	6.24e-05± 0.47	5.98e-05± 0.44	**5.79e-05**± 0.32	6.65e-05± 1.06
100	5.77e-03± 0.14	**2.28e-04**± 0.34	1.63e-03± 0.46	2.19e-03± 0.10
1,000	–	–	4.29e-02± 0.83	**3.21e-02**± 0.88
3,000	–	–	3.16e-02± 0.14	**2.38e-02**± 0.08

G LIMITATIONS

While the proposed FLEXMD provides a principled and structure-preserving framework for molecular dynamics modeling, several limitations remain. The approach assumes the existence of a finite-dimensional Koopman subspace in which inherently nonlinear and potentially chaotic molecular dynamics can be well-approximated by linear evolution; however, such a globally valid linear representation is unlikely to exist in practice, and thus this assumption may only hold approximately. Moreover, restricting the latent generator to a quadratic-form-preserving family may reduce expressivity for dissipative, stochastic, or externally forced processes. Because the preserved quantity is defined in latent space, the method does not guarantee conservation of physical energy or other molecular observables. Finally, the method relies on high-quality trajectory data and does not explicitly address noise, partial observability, or cross-ensemble generalization, which remain common challenges for trajectory-based learning approaches and warrant further investigation.

H BROADER IMPACT.

This work presents a foundational approach for molecular dynamics and is not directly tied to immediate real-world deployment; however, several broader impacts should be considered. On the positive side, improving the accuracy and efficiency of molecular simulation may accelerate scientific discovery in areas such as drug design and materials science. At the same time, such advancements could potentially be misused, for instance, to facilitate the design of harmful chemical compounds if applied without appropriate oversight.