Graph Fourier Neural ODEs: Modeling Spatial-temporal Multi-scales in Molecular Dynamics

Anonymous authors
Paper under double-blind review

Abstract

Accurately predicting long-horizon molecular dynamics (MD) trajectories remains a significant challenge, as existing deep learning methods often struggle to retain fidelity over extended simulations. We hypothesize that one key factor limiting accuracy is the difficulty of capturing interactions that span distinct spatial and temporal scales—ranging from high-frequency local vibrations to low-frequency global conformational changes. To address these limitations, we propose **Graph Fourier Neural ODEs (GF-NODE)**, integrating a graph Fourier transform for spatial frequency decomposition with a Neural ODE framework for continuous-time evolution. Specifically, GF-NODE first decomposes molecular configurations into multiple spatial frequency modes using the graph Laplacian, then evolves the frequency components in time via a learnable Neural ODE module that captures both local and global dynamics, and finally reconstructs the updated molecular geometry through an inverse graph Fourier transform. By explicitly modeling high- and low-frequency phenomena in this unified pipeline, GF-NODE more effectively captures long-range correlations and local fluctuations alike. Experimental results on challenging MD benchmarks, including MD17 and alanine dipeptide, demonstrate that GF-NODE achieves state-of-the-art accuracy while preserving essential geometrical features over extended simulations. These findings highlight the promise of bridging spectral decomposition with continuous-time modeling to improve the robustness and predictive power of MD simulations. Our implementation is publicly available at `https://anonymous.4open.science/r/GF-NODE-code-B289/`

1 Introduction

Molecular dynamics (MD) simulations are indispensable tools for investigating the behavior of molecular systems at the atomic level, offering profound insights into physical (Bear & Blaisten-Barojas, 1998), chemical (Wang et al., 2011), and biological (Salo-Ahen et al., 2020) processes. These simulations must capture interactions occurring across a wide range of spatial and temporal scales—from localized bond vibrations to long-range non-bonded interactions—posing significant computational challenges. Accurately modeling these multiscale interactions is crucial for uncovering the mechanisms underlying complex molecular phenomena but remains prohibitively expensive for large systems and long trajectories (Vakis et al., 2018).

In recent years, Graph Neural ODEs (Zang & Wang, 2020; Huang et al., 2023a) have gained traction for modeling continuous-time dynamics in multi-agent systems, including MD. By learning an Ordinary Differential Equation (ODE) function via Graph Neural Networks and solving it numerically, these methods allow flexible sampling at arbitrary time points. Such a continuous-time formulation is well-suited for capturing multiple temporal scales inherent in molecular simulations. However, significant challenges persist in accounting for the rich spatial multiscale effects that span localized bond vibrations to extended nonbonded interactions. On another front, Fourier Neural Operators (FNOs) (Li et al., 2020) have demonstrated success in learning operators by decomposing signals into different frequency modes, thereby capturing various spatial scales effectively. Yet, they are not tailored for graph-structured molecular data or continuous-time temporal evolution, limiting their direct applicability to general MD simulations.

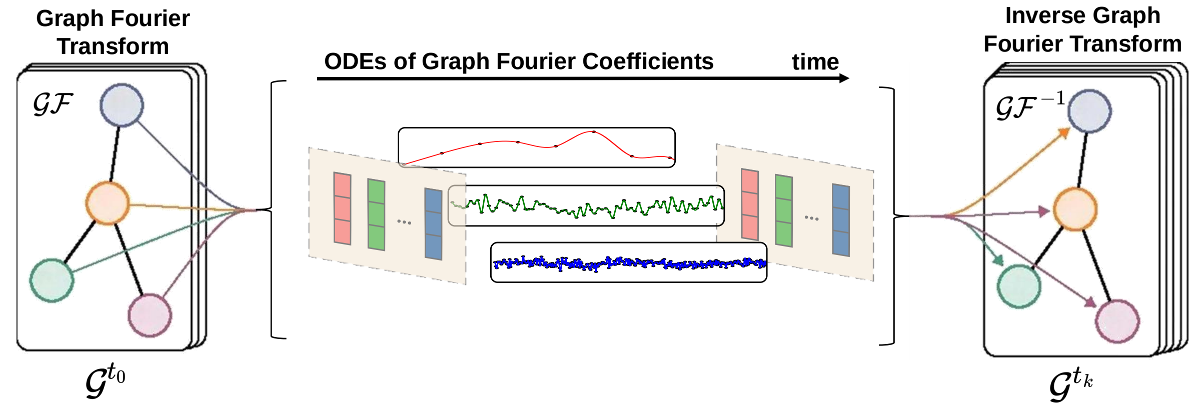
Graph Fourier Transform
$\mathcal{GF}$
ODEs of Graph Fourier Coefficients
time
Inverse Graph Fourier Transform
$\mathcal{GF}^{-1}$
$\mathcal{G}^{t_0}$
$\mathcal{G}^{t_k}$

Figure 1: Overview of the Graph Fourier Neural ODE framework. The molecular graph $\mathcal{G}^{t_0}$ is first transformed into the spectral domain using a Graph Fourier Transform ($\mathcal{GF}$), decomposing the spatial structure into frequency components. Neural ODEs are then applied to evolve the Fourier coefficients over time. The evolved coefficients are finally transformed back into the physical domain using an inverse Graph Fourier Transform ($\mathcal{GF}^{-1}$), reconstructing the molecular graph at future time t_k.

Despite the promise of Graph Neural ODEs for handling multiple temporal scales via continuous time modeling, they alone are inadequate for fully capturing the complex spatial frequency components of molecular systems. Conversely, while approaches based on Fourier transforms can model multiple spatial scales, they do not naturally handle the intricacies of molecular graphs or continuous-time dynamics. To address these limitations, we introduce *Graph Fourier Neural ODEs (GF-NODE)*. As demonstrated in Figure 1, our framework explicitly integrates a graph Fourier transform—for decomposing and encoding spatial multiscale interactions—with a Neural ODE framework for continuous-time modeling of each spatial frequency. By leveraging an inverse graph Fourier transform at the end of the pipeline, GF-NODE reconstructs the molecular state in physical space, thereby enabling a unified approach to spatial and temporal multiscale simulation.

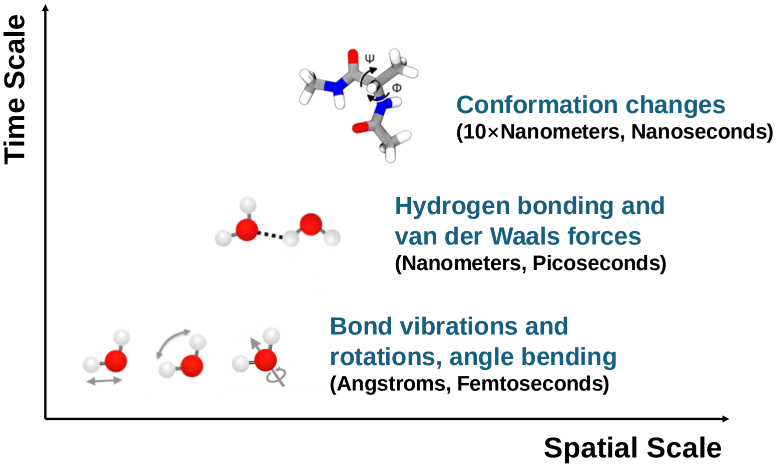
Time Scale
Spatial Scale
Ψ
Φ
Conformation changes
(10×Nanometers, Nanoseconds)
Hydrogen bonding and
van der Waals forces
(Nanometers, Picoseconds)
Bond vibrations and
rotations, angle bending
(Angstroms, Femtoseconds)

Figure 2: Illustration of spatial and temporal multiscale interactions in molecular dynamics. The x-axis indicates spatial scales ranging from local bond vibrations to global conformational changes. The y-axis represents timescales ranging from femtoseconds (bond oscillations) to nanoseconds (conformational rearrangements). Local interactions, such as bond vibrations and angle bending, occur over short spatial scales (angstroms) and fast timescales (femtoseconds). Non-local interactions, including hydrogen bonding and van der Waals forces, span larger spatial scales (nanometers) and intermediate timescales (picoseconds). Conformational changes involve the longest spatial and temporal scales, potentially up to nanoseconds or longer.

We conduct extensive experiments on benchmark molecular dynamics datasets, including MD17 and alanine dipeptide. Empirical results show that GF-NODE achieves state-of-the-art accuracy in predicting molecular trajectories over long-horizon simulations, preserves essential geometric properties such as bond lengths and angles, and demonstrates stable performance over temporal super-resolution tasks. These findings underscore the importance of explicitly decomposing molecular configurations into spatial frequency modes and evolving them continuously in time. Our analysis suggests that this multiscale perspective is instrumental for capturing both rapid local fluctuations and slow global conformational changes.

Our contribution can be summarized as follows.

(a) New perespective. We provide a new perspective on spatial-temporal multiscale modeling for molecular dynamics by jointly capturing spatial and temporal interactions within a single framework.

(b) Novel architecture. Building on this perspective, we introduce the *Graph Fourier Neural ODEs (GF-NODE)* architecture, which combines a graph Fourier transform for decomposing molecular interactions into distinct spatial frequencies with a Neural ODE formulation to evolve these frequencies continuously in time.

(c) Good performance. Through extensive experiments on benchmark molecular dynamics datasets, we demonstrate that GF-NODE not only outperforms existing approaches in predictive accuracy but also effectively preserves critical molecular geometrical features, thus offering a promising solution for multiscale MD simulations.

2 Related Work

We review relevant works on multi-scale modeling in molecular dynamics, focusing on neural operator models and graph neural ODEs.

2.1 Classical Molecular Simulation Methods

Traditional molecular simulation methods, including force-field based MD (e.g., AMBER (Cornell et al., 1995; Brooks et al., 2009), GAMD (Li et al., 2022b)) and *ab initio* techniques such as Car–Parrinello MD (Hutter, 2012), have been foundational in exploring molecular behavior. However, these methods face significant limitations: force-field simulations require extremely small time steps to accurately resolve high-frequency bond vibrations, which hampers long-term stability and computational efficiency; *ab initio* MD, though offering first-principles accuracy, is computationally prohibitive for large systems and long trajectories; and while coarse-grained models (e.g., MARTINI (Souza et al., 2021)) enable more efficient multiscale simulations, they often compromise on molecular detail and accuracy, particularly in reproducing local interactions and maintaining seamless force consistency at multiscale interfaces.

2.2 Neural Operator Models

Neural operator models (Kovachki et al., 2023) have emerged as powerful tools for learning mappings between infinite-dimensional function spaces, demonstrating success in modeling complex dynamical systems. Among these, *Fourier Neural Operators (FNOs)* (Li et al., 2020; 2022a; Liu & Jafarzadeh, 2023; Kovachki et al., 2021; Koshizuka et al.) are particularly notable for handling spatial multi-scale interactions in partial differential equation (PDE) data by learning representations in the Fourier domain. However, while FNOs efficiently capture spatial hierarchies, they do not inherently model temporal dynamics, making them suboptimal for time-evolving molecular systems.

In contrast, recent operator-based methods such as the *Implicit Transfer Operator (ITO)* (Schreiner et al., 2024), *Timewarp* (Klein et al., 2024), and *Equivariant Graph Neural Operator (EGNO)* (Xu et al., 2024) focus on temporal multi-scale modeling in molecular dynamics. ITO and Timewarp introduce coarse-graining and adaptive time-stepping mechanisms to accelerate long-horizon simulations. EGNO employs neural operators with SE(3) equivariance to capture rotational and translational symmetries, yet it primarily addresses temporal evolution without explicitly handling spatial multi-scale effects. While these models successfully extend the applicability of neural operators to molecular simulations, none jointly addresses both spatial and temporal multi-scales in molecular dynamics.

2.3 Graph Neural ODE Models

Graph Neural ODEs combine Graph Neural Networks with Neural ODE frameworks (Chen et al., 2018; Kidger, 2022; Goyal & Benner, 2023; Holt et al., 2022; Luo et al., 2023; Luo et al.) to model continuous-time dynamics on graph-structured data (Zang & Wang, 2020; Kim et al., 2021; Huang et al., 2020; 2021; 2023a;b). These methods excel in capturing temporal multi-scale behavior by allowing flexible time integration, making them well-suited for systems with varying temporal resolutions. However, they primarily focus on modeling temporal dependencies, with little emphasis on explicitly handling spatial multi-scales in molecular dynamics.

A key limitation of existing Graph Neural ODE models is their reliance on local message passing, which inherently constrains their ability to capture long-range spatial dependencies within molecular systems. As a result, they may fail to adequately represent the interplay between localized, high-frequency interactions (e.g., bond vibrations) and global, low-frequency effects (e.g., large-scale conformational changes). Unlike spectral-based approaches that can decompose spatial hierarchies, standard Graph Neural ODEs lack a mechanism to explicitly encode spatial multi-scale structures, limiting their effectiveness in modeling complex molecular dynamics.

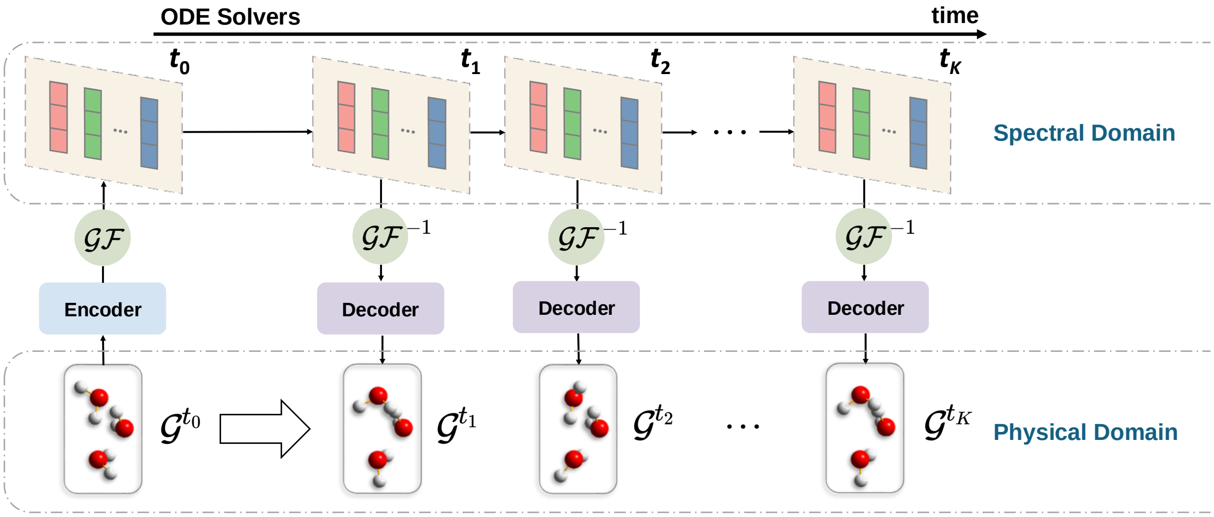

Figure 3: An overview of the proposed GF-NODE architecture. The model first encodes the initial molecular graph $\mathcal{G}^{t_0}$ from the physical domain into the spectral domain via a graph Fourier transform ($\mathcal{GF}$). Neural ODE solvers then propagate the dynamics of the Fourier coefficients continuously across time points $t_0, t_1, \ldots, t_K$. The transformed coefficients are subsequently decoded back to the physical domain using an inverse graph Fourier transform ($\mathcal{GF}^{-1}$), reconstructing the molecular graphs $\mathcal{G}^{t_1}, \mathcal{G}^{t_2}, \ldots, \mathcal{G}^{t_K}$ at future time points. This design enables efficient modeling of spatial and temporal multiscale dynamics in molecular systems.

3 The Proposed Approach

We propose **GF-NODE**, a framework specifically designed to address the limitations of existing methods in capturing both spatial and temporal multiscale dynamics in molecular systems. As discussed in Sections 1 and 2, current approaches either handle spatial scales using Fourier-based methods or focus on temporal scales using Graph Neural ODEs, but they do not jointly model these scales in a unified framework. GF-NODE directly addresses this gap by integrating the *Graph Fourier Transform (GFT)* and *Neural Ordinary Differential Equations (Neural ODEs)* to simultaneously decompose spatial interactions and model their continuous-time evolution.

Specifically, as illustrated in Figure 3, GF-NODE first applies a Graph Fourier Transform to decompose the molecular graph into different frequency components, effectively separating localized, high-frequency interactions from global, low-frequency patterns. This spectral representation allows the model to process molecular structures in a frequency-adaptive manner, capturing both fine-grained local interactions and large-scale conformational changes. The decomposed spectral coefficients are then evolved continuously over time using Neural ODEs, ensuring flexible and adaptive modeling of multiscale temporal dynamics. Finally, the inverse Graph Fourier Transform reconstructs the molecular graph in the physical domain, preserving both local and global structural information over long-horizon molecular simulations.

This design enables GF-NODE to overcome key challenges in molecular dynamics modeling: (1) explicitly encoding spatial multi-scale interactions via spectral decomposition, (2) leveraging continuous-time evolution to capture complex temporal dependencies, and (3) integrating these spatial and temporal scales into a single end-to-end framework. Below, we detail the core components and operations of GF-NODE.

3.1 Notation and problem setup

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ represent a molecular graph, where N is the number of nodes $i \in \{1, \dots, N\}$ (atoms), and $\mathcal{E}$ is the set of edges representing chemical bonds or interactions. Edges are identified by checking whether the distance between atoms falls below a threshold. Each node i has:

- Invariant (scalar) features $\mathbf{h}_i \in \mathbb{R}^F$, which include the velocity magnitude $\|\mathbf{v}_i\|$ and a normalized version of the atomic number Z_i;
- Vector features $\mathbf{z}_i \in \mathbb{R}^{m \times 3}$, which contain position $\mathbf{x}_i$ and velocity $\mathbf{v}_i$ (i.e. $\mathbf{z}_i = (\mathbf{x}_i, \mathbf{v}_i)$ when $m = 2$).

At time t_0, the molecular system's state is given as $\{\mathbf{h}, \mathbf{z}\}$, and we aim to predict the future configuration $\mathcal{G}^{(t_k)}$ for any $t_k > t_0$.

3.2 Graph Neural Network encoder

We use a Graph Neural Network (GNN) to encode scalar features $\mathbf{h}_i \in \mathbb{R}^F$ and vector features $\mathbf{z}_i \in \mathbb{R}^{m \times 3}$ for each node i. The initial scalar feature $\mathbf{h}_i^{(0)}$ for each atom i is formed by concatenating $\|\mathbf{v}_i\|$ and $\frac{Z_i}{Z_{\max}}$, where Z_i is the atomic number of atom i, and $Z_{\max}$ is a reference maximum. This concatenated vector is then mapped by a linear embedding layer, producing the hidden dimension used by the GNN. The vector feature $\mathbf{z}_i^{(0)}$ contains $\mathbf{x}_i$ and $\mathbf{v}_i$. Each **GNN** layer performs message passing, where node i's features are updated based on its neighbors $\mathcal{N}(i)$:

$$\mathbf{m}_{ij}^{(l)} = \phi_e\left(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}, \mathbf{r}_{ij}^{(l)}\right), \tag{1}$$

$$\mathbf{h}_i^{(l+1)} = \phi_h\left(\mathbf{h}_i^{(l)}, \sum_{j \in \mathcal{N}(i)} \mathbf{m}_{ij}^{(l)}\right), \tag{2}$$

$$\mathbf{x}_i^{(l+1)} = \mathbf{x}_i^{(l)} + \frac{1}{|\mathcal{N}(i)|} \sum_{j \in \mathcal{N}(i)} \psi\left(\mathbf{m}_{ij}^{(l)}\right), \tag{3}$$

where $\mathbf{r}_{ij}^{(l)} = \mathbf{x}_i^{(l)} - \mathbf{x}_j^{(l)}$, and ϕ_e, ϕ_h, and ψ are neural networks. Similarly, we also update the velocity $\mathbf{v}_i^{(l)}$ if present. After L layers, the GNN produces the encoded features $\mathbf{h}_i^{(L)}$ and $\mathbf{z}_i^{(L)}$, capturing both local and global molecular information. These features are then passed to the Graph Fourier Transform (GFT) for spectral decomposition.

3.3 Graph Fourier Transform

Molecular dynamics exhibit behavior on multiple spatial scales: large-scale "global" deformations can be viewed as low-frequency modes, whereas fast "local" vibrations correspond to high frequency modes. In classical signal processing, Fourier analysis decomposes signals into sinusoids of different frequencies. Similarly, on graphs, we can decompose node-based signals into eigenmodes of a suitable operator (often the graph Laplacian). By retaining or emphasizing certain frequency bands, one can explicitly model global vs. local patterns.

Graph Signal as Combination of Laplacian Eigenvectors. A scalar function $f : \mathcal{V} \to \mathbb{R}$ on the nodes can be seen as a vector $\mathbf{f} \in \mathbb{R}^N$. The graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ (or a symmetrized variant) admits an eigen-decomposition:

$$\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top, \tag{4}$$

where $\mathbf{U} = [\mathbf{u}_0, \mathbf{u}_1, \ldots, \mathbf{u}_{N-1}]$ is an orthonormal matrix of eigenvectors, i.e. $\mathbf{u}_k \in \mathbb{R}^N$, and $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_0, \lambda_1, \ldots, \lambda_{N-1})$ contains ascending eigenvalues $0 \le \lambda_0 \le \lambda_1 \le \cdots \le \lambda_{N-1}$.

In graph signal processing, the eigenvector $\mathbf{u}_k$ is viewed as the "k-th frequency basis." Smaller eigenvalues ($\lambda_0, \lambda_1, \ldots$) correspond to low-frequency (more global, smooth) variations on the graph, while larger eigenvalues correspond to high-frequency (local, rapidly changing) modes. Hence, any signal $\mathbf{f} \in \mathbb{R}^N$ can be written as a linear combination:

$$\mathbf{f} = \sum_{k=0}^{N-1} \alpha_k \mathbf{u}_k, \quad \text{where } \alpha_k = \mathbf{u}_k^\top \mathbf{f}. \tag{5}$$

This collection of coefficients $\{\alpha_k\}$ is the Graph Fourier Transform (GFT) of $\mathbf{f}$. We **now denote** α_k as $\tilde{f}_k$.

Truncation at M Bases. For large N, we often use only the first M eigenvectors $\{\mathbf{u}_0, \ldots, \mathbf{u}_{M-1}\}$ to approximate $\mathbf{f}$. This yields a band-limited or multiscale representation:

$$\mathbf{f} \approx \sum_{k=0}^{M-1} \alpha_k \mathbf{u}_k, \tag{6}$$

where $M \le N$. Sorting λ_k in ascending order ensures the lowest-frequency modes (i.e., global scales) appear first, and higher-frequency (local scales) modes appear last. By choosing M appropriately, we focus on the most critical modes for modeling global vs. local behavior.

Applying the GFT to Scalar vs. Vector Features. For *scalar* features $\mathbf{H} \in \mathbb{R}^{N \times F}$, where F is the latent feature size, we substitute $\tilde{f}_k$ for $\tilde{\mathbf{H}}_k$ in the equation $\tilde{f}_k = \mathbf{u}_k^\top f$ (from Equation 5) to get:

$$\tilde{\mathbf{H}} = \left[\tilde{\mathbf{H}}_k\right]_{k=0}^{M-1} = \mathbf{U}_{(:,0:M)}^\top \mathbf{H}, \tag{7}$$

where $\mathbf{U}_{(:,0:M)}$ denotes the first M eigenvectors. For *vector* features $\mathbf{Z} \in \mathbb{R}^{N \times m \times 3}$, where m is the feature size of the vector feature in the 3D space, we first remove the mean to ensure translational invariance (since the 0-th eigenvector $\mathbf{u}_0$ corresponds to the constant mode):

$$\mathbf{Z}_c = \mathbf{Z} - \overline{\mathbf{Z}}. \tag{8}$$

where $\overline{\mathbf{Z}}$ is the global mean over all nodes. We then apply the same truncated basis $\mathbf{U}_{(:,0:M)}^\top$ to each coordinate dimension, substituting $\tilde{f}_k$ for $\tilde{\mathbf{Z}}_k$ in the equation $\tilde{f}_k = \mathbf{u}_k^\top f$ (from Equation 5) to get:

$$\tilde{\mathbf{Z}} = \mathbf{U}_{(:,0:M)}^\top \mathbf{Z}_c \in \mathbb{R}^{M \times m \times 3}. \tag{9}$$

Indices closer to $k = 0$ indicate more global motions, while larger k (up to $M - 1$) captures more local, high-frequency fluctuations. In practice, M can be a hyperparameter that determines how many eigenmodes we keep, balancing efficiency (fewer modes to evolve) and accuracy (how many scales are captured).

Theoretical Analysis. The Graph Fourier Transform (GFT) decomposes molecular dynamics into modes corresponding to global and local spatial patterns, as determined by the eigenvalues and eigenvectors of the graph Laplacian. As demonstrated in Proposition 3.1, low-frequency modes capture smooth, global deformations, while high-frequency modes represent localized structural variations. This decomposition enables efficient modeling of multiscale spatial dynamics.

Proposition 3.1 (Global vs. Local Spatial Scales)**.** *Let $\mathbf{u}_k$ be the k-th eigenvector of $\mathbf{L}$ with eigenvalue λ_k. Suppose $\mathbf{x}$ encodes atomic coordinates or their latent features. Then:*

1. ***If λ_k is small**, the corresponding mode $\mathbf{u}_k$ represents slowly varying (global) deformations across the molecule.*

2. ***If*** λ_k ***is large****, the corresponding mode* $\mathbf{u}_k$ *represents rapidly changing (local) structural variations.*

Regarding the number of modes M we need to use, Theorem 3.2 guarantees that we can use the first few modes to represent the entire system accurately:

Theorem 3.2 (Spectral Truncation Error)**.** *Truncating the spectral representation to the first* M *modes,* $\mathbf{x}_{(M)} = \mathbf{U}_{(:,0:M)}\mathbf{U}_{(:,0:M)}^{\top}\mathbf{x}$*, yields an* ℓ^2*-norm error:*

$$\left\|\mathbf{x} - \mathbf{x}_{(M)}\right\|_2^2 = \sum_{k=M}^{N-1} \left|\mathbf{u}_k^{\top}\mathbf{x}\right|^2 . \tag{10}$$

For α*-bandlimited signals, choosing* M *such that* $\lambda_{M-1} \leq \alpha$ *guarantees exact recovery.*

We provide detailed proofs and derivations in Appendix C, where we analyze the spectral decomposition's efficacy and demonstrate its suitability for multiscale molecular modeling.

3.4 Neural ODEs in the Spectral Domain

We propose a novel approach that models the temporal evolution of the **Fourier coefficients** using Neural ODEs. By representing molecular interactions in the frequency domain, this approach enables the decomposition of dynamics across different spatial scales and provides a more compact representation for modeling temporal evolution. Neural ODEs then learn the dynamics of these Fourier coefficients over time, offering a continuous-time framework that captures multiscale spatial and temporal interactions simultaneously.

NeuralODE Preliminaries. Neural Ordinary Differential Equations (Neural ODEs) (Chen et al., 2018) provide a framework for modeling continuous-time dynamics by learning the evolution of a system as a set of differential equations parameterized by a neural network. Specifically, for a system's state $\mathbf{h}(t)$, its temporal evolution is defined as:

$$\frac{d\mathbf{h}(t)}{dt} = f_\theta(\mathbf{h}(t), t), \tag{11}$$

where f_θ is a neural network parameterized by θ that learns the dynamics of $\mathbf{h}(t)$ over time. Given an initial state $\mathbf{h}(t_0)$, the state at any future time t is computed by solving the ODE:

$$\mathbf{h}(t) = \mathbf{h}(t_0) + \int_{t_0}^{t} f_\theta(\mathbf{h}(\tau), \tau)d\tau. \tag{12}$$

This integral can be evaluated numerically using adaptive ODE solvers, such as Runge-Kutta methods, allowing Neural ODEs to handle irregularly sampled data and continuously model temporal dynamics.

Joint Scalar-Vector Block-Diagonal Formulation. We update the spectral coefficients $[\tilde{\mathbf{H}}, \tilde{\mathbf{Z}}]$ jointly. We can write the combined spectral features as

$$\begin{pmatrix}\tilde{\mathbf{H}}\\ \tilde{\mathbf{Z}}\end{pmatrix} \in \mathbb{R}^{N\times F} \oplus \mathbb{R}^{N\times m\times 3}, \tag{13}$$

and define Neural ODE dynamics over continuous time t :

$$\frac{d}{dt}\begin{pmatrix}\tilde{\mathbf{H}}(t)\\ \tilde{\mathbf{Z}}(t)\end{pmatrix} = \begin{pmatrix}f_\theta(\tilde{\mathbf{H}}(t), t)\\ g_\theta(\tilde{\mathbf{Z}}(t), t)\end{pmatrix}. \tag{14}$$

Mathematically, in the Fourier space, this equates to a block-diagonal operator $\mathbf{M}_\theta$. Let $\tilde{\mathbf{H}}_\omega$ and $\tilde{\mathbf{Z}}_\omega$ denote the coefficients for each frequency mode ω. A single ODE step can be represented as:

$$\begin{pmatrix}\tilde{\mathbf{H}}_\omega\\ \tilde{\mathbf{Z}}_\omega\end{pmatrix} \mapsto \begin{bmatrix}\mathbf{M}_\theta^{(\mathbf{h})} & \mathbf{0}\\ \mathbf{0} & \mathbf{M}_\theta^{(\mathbf{z})}\end{bmatrix} \cdot \begin{pmatrix}\tilde{\mathbf{H}}_\omega\\ \tilde{\mathbf{Z}}_\omega\end{pmatrix}. \tag{15}$$

Here, $\mathbf{M}_\theta^{(\mathbf{h})}$ operates on the scalar channels and $\mathbf{M}_\theta^{(\mathbf{z})}$ on the vector channels, allowing them to evolve in a coordinated but distinct manner. This is beneficial to preserving the 3D interactions while keeping track of the nuanced representations in the latent space – thus preserving the 3D interactions while capturing the nuanced latent representations in a block-diagonal design.

Implementation wise, each ODE function f_θ or g_θ takes the current spectral modes ($\tilde{\mathbf{H}}_\omega$ or $\tilde{\mathbf{Z}}_\omega$) and the time t, and produces their rate of change for the integrator. Specifically, we use multi-head self-attention across these modes so that each frequency mode can attend to others. Formally, if $\mathbf{h}_\omega(t)$ denotes the coefficients corresponding to frequency mode ω at time t, the self-attention step can be written as

$$\mathbf{h}'_\omega(t) = \mathrm{MHAttn}\big(\mathbf{h}_\omega(t)\big), \tag{16}$$

where MHAttn is the multi-head attention layer across different frequency modes. Next, the time t is directly concatenated along the feature dimension via a learned embedding $\gamma(t) \in \mathbb{R}^d$, yielding

$$\mathbf{h}''_\omega(t) = \big[\mathbf{h}'_\omega(t) \,\|\, \gamma(t)\big]. \tag{17}$$

Finally, each mode $\mathbf{h}''_\omega(t)$ is transformed by a mode-wise linear weight $\mathbf{W}_\omega$, resulting in the derivative:

$$f_\theta\big(\mathbf{h}_\omega(t), t\big) \;=\; \mathbf{W}_\omega\, \mathbf{h}''_\omega(t). \tag{18}$$

A similar procedure applies for g_θ, handling any additional vector channels by suitably reshaping and performing attention across modes.

To predict at time t_k, we numerically integrate $\frac{d}{dt}[\tilde{\mathbf{H}}, \tilde{\mathbf{Z}}]$ from t_0 to t_k (e.g., via dopri5). Low-frequency components capture long-range, slower dynamics, while high-frequency components capture faster local fluctuations.

3.5 From Spectral Domain to Physical Domain

Inverse GFT. After evolving the spectral coefficients $\tilde{\mathbf{H}}\left(t_k\right)$ and $\tilde{\mathbf{Z}}\left(t_k\right)$, we recover the node-level signals via the inverse GFT:

$$\mathbf{H}\left(t_k\right) = \mathbf{U}_{(:,0:M)}\tilde{\mathbf{H}}\left(t_k\right), \quad \mathbf{Z}_c\left(t_k\right) = \mathbf{U}_{(:,0:M)}\tilde{\mathbf{Z}}\left(t_k\right), \tag{19}$$

where $\mathbf{U}_{(:,0:M)}$ (the matrix of the first M eigenvectors) is the same truncated basis used during the forward GFT. Finally, we restore the global translation by adding back the mean $\overline{\mathbf{Z}}$ that was subtracted earlier:

$$\mathbf{Z}\left(t_k\right) = \mathbf{Z}_c\left(t_k\right) + \overline{\mathbf{Z}}. \tag{20}$$

Graph Neural Network Decoder. We refine local interactions at each predicted time t_k with a GNN decoder that again operates on $\big[\mathbf{H}(t_k), \mathbf{Z}(t_k)\big]$. This step can help capture short-range correlations that may not be fully resolved in the spectral update. The decoder GNN has a structure similar to the encoder:

$$\mathbf{m}_{ij}^{(\text{dec})} = \phi'_e\Big(\mathbf{h}_i(t_k),\, \mathbf{h}_j(t_k),\, \mathbf{r}_{ij}^{(\text{dec})}(t_k)\Big), \tag{21}$$

$$\mathbf{h}_i^{(\text{dec})}(t_k) = \phi'_h\Big(\mathbf{h}_i(t_k),\, \sum_{j\in\mathcal{N}(i)} \mathbf{m}_{ij}^{(\text{dec})}\Big), \tag{22}$$

$$\mathbf{x}_i^{(\text{dec})}(t_k) = \mathbf{x}_i(t_k) \;+\; \frac{1}{|\mathcal{N}(i)|} \sum_{j\in\mathcal{N}(i)} \psi'\big(\mathbf{m}_{ij}^{(\text{dec})}\big), \tag{23}$$

and similarly for $\mathbf{v}_i$. Here, $\mathbf{r}_{ij}^{(\text{dec})}(t_k) = \mathbf{x}_i(t_k) - \mathbf{x}_j(t_k)$, and ϕ'_e, ϕ'_h, ψ' are learnable functions.

4 Experiments

In this section, we provide a comprehensive evaluation of our proposed model on molecular dynamics datasets, comparing its performance against several baselines and conducting detailed ablation studies to assess the contributions of the different components. Our experimental evaluation is designed to address the following key **research questions**:

- **RQ1: Prediction Accuracy.** Does the proposed GF-NODE framework deliver improved molecular dynamics prediction accuracy compared to state-of-the-art methods?
- **RQ2: Continuous Time Modeling.** How effectively does the continuous-time evolution component capture multiscale temporal dynamics—including long-horizon forecasting and super-resolution—compared to variants without continuous-time propagation?
- **RQ3: Spatial Multiscale Modeling.** How crucial is the explicit spectral decomposition for capturing spatial multiscale interactions, and how do different Fourier mode interaction schemes and the number of retained modes affect overall performance?

4.1 Dataset

We evaluate our model using the MD17 dataset (Chmiela et al., 2017), which contains molecular dynamics trajectories for eight small molecules: Aspirin, Benzene, Ethanol, Malonaldehyde, Naphthalene, Salicylic Acid, Toluene, and Uracil. The dataset provides atomic trajectories simulated under quantum mechanical forces, capturing realistic molecular motions.

We also evaluated our model on the alanine dipeptide dataset (Schreiner et al., 2024), a standard benchmark for studying conformational dynamics in proteins. Our task is to predict the future positions of atoms given the initial state of the molecular system.

4.2 Experimental Setup

For each molecule, we partition the trajectory data into training, validation, and test sets, using 500 samples for training, 2000 for validation, and 2000 for testing. The time scope ΔT for each piece of data is set to 3000 simulation steps, providing a challenging prediction horizon.

A key aspect of our experimental setup is the use of **irregular timestep sampling**, in contrast to the equi-timestep sampling used in some baseline models like EGNO, to better mimic the variable time intervals in real-world physical systems. This setting tests the models' ability to handle irregular temporal data. Although each trajectory spans 3000 simulation steps, we randomly sample only 8 data points per instance. Because these samples are drawn from different points in the 3000-step window across data instances, this strategy enables the model to learn the dynamics over the entire time span without requiring training on every timestep, thereby significantly enhancing **efficiency**. Nevertheless, we also provide evaluations based on equi-timestep sampling in Appendix B for completeness.

4.3 Baseline Models

We compare our model against several state-of-the-art approaches:

- **NDCN** (Zang & Wang, 2020): A Graph Neural ODE model that integrates graph neural networks into the ODE framework to learn continuous-time dynamics of networked systems.
- **LG-ODE** (Huang et al., 2020): A latent graph-based ODE model that integrates latent representations into continuous-time evolution.
- **EGNN** (Satorras et al., 2021): An Equivariant Graph Neural Network that models molecular systems using 3D equivariant message passing but without explicit time propagation.
- **EGNO** (Xu et al., 2024): An Equivariant Graph Neural Operator that captures temporal dynamics using neural operators with regular timesteps.
- **ITO** (Schreiner et al., 2024): An Implicit Time-stepping Operator that integrates differential equations into the learning process for temporal evolution.

These baselines represent a range of approaches for modeling molecular dynamics, including methods that focus primarily on spatial modeling (EGNN) or temporal modeling (NDCN, LG-ODE, EGNO, ITO).

4.4 Results and Analysis (RQ1)

Units and Calculations: Note that the alanine dipeptide dataset operates in nanometers (nm), whereas the MD17 dataset uses angstroms (Å). When calculating bond lengths and bond angles, we consider all heavy atoms (excluding hydrogen) to focus on the core structure.

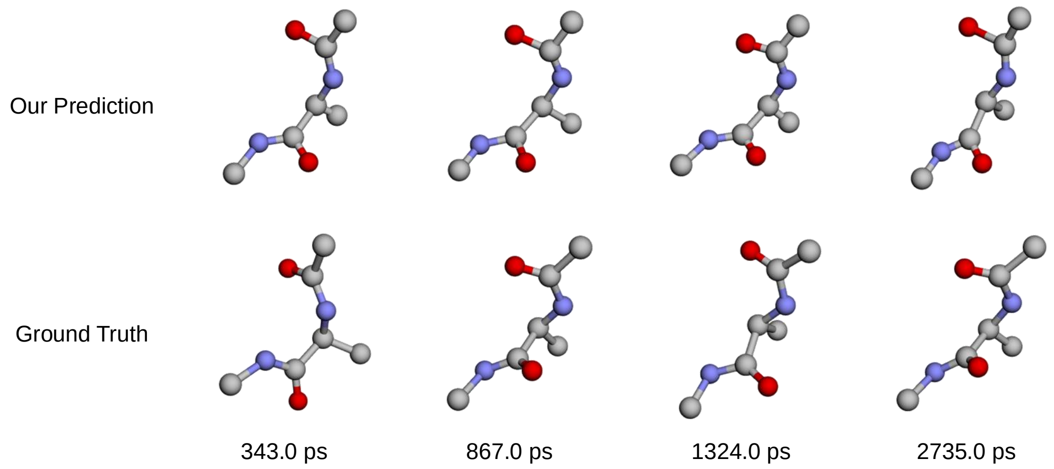

Figure 4: Representative snapshots of alanine dipeptide, comparing our model's predicted conformations (top row) against the ground-truth simulation (bottom row) at four different timestamps. The close agreement illustrates the model's ability to preserve key structural features over an extended trajectory.

To address **RQ1**, we evaluate the performance of our model and the baselines on the MD17 dataset and alanine dipeptide dataset, and Figure 4 provides visualization of our predictions. The test Mean Squared Error (MSE) on the eight molecules in MD17 and alanine dipeptide for our model and the baseline methods are summarized in Tables 1 and 3, under **irregular** timestep sampling. For completeness, we also provide results in the equi-timestep setting in Table 7 in Appendix B.

Table 1: MSE ($\times 10^{-2}$ Å^2) on the MD17 dataset with **irregular** timestep sampling. Best results are in **bold**, and second-best are underlined.

	Aspirin	Benzene	Ethanol	Malonaldehyde	Naphthalene	Salicylic	Toluene	Uracil
NDCN	29.75±0.02	70.13±0.98	10.05±0.02	42.28±0.07	2.30±0.00	3.43±0.05	12.33±0.00	2.39±0.00
LG-ODE	51.65±0.01	68.29±0.21	12.32±0.05	43.95±0.07	2.38±0.02	2.85±0.08	18.11±0.09	2.38±0.07
EGNN	9.09±0.10	49.15±1.68	4.46±0.01	12.52±0.05	0.40±0.02	0.89±0.01	8.98±0.09	0.64±0.00
EGNO	10.60±0.01	52.53±2.40	4.52±0.06	12.89±0.06	0.46±0.01	1.07±0.00	9.31±0.10	0.67±0.01
ITO	12.74±0.10	57.84±0.86	7.23±0.00	19.53±0.01	1.77±0.01	2.53±0.03	9.96±0.04	1.71±0.15
Ours	**6.46**±0.03	**1.52**±0.08	**2.74**±0.05	**10.54**±0.01	**0.23**±0.02	**0.63**±0.01	**1.80**±0.05	**0.41**±0.01

Table 2: Mean Absolute Errors (MAEs) and relative errors for bond lengths and bond angles on alanine dipeptide. Best results are in **bold**.

Model	**Bond Length MAE (nm)**	**Rel. Err. (%)**	**Bond Angle MAE (°)**	**Rel. Err. (%)**
EGNN	0.0209 ± 0.0006	15.32 ± 0.49	12.44 ± 0.91	10.48 ± 0.76
EGNO	0.0229 ± 0.0018	16.75 ± 1.23	10.54 ± 0.11	8.89 ± 0.11
Ours	$\mathbf{0.0188} \pm 0.0022$	$\mathbf{13.74} \pm 1.66$	$\mathbf{10.47} \pm 1.03$	$\mathbf{8.80} \pm 0.89$

From Table 1, we observe that our model consistently outperforms the baseline methods across all eight molecules under irregular timestep sampling. This demonstrates the effectiveness of our approach in jointly modeling spatial and temporal multiscale interactions. The performance gains are particularly pronounced for molecules such as Benzene and Aspirin, where our model significantly reduces the MSE compared to the baselines.

Benzene Drifting. Interestingly, we find that Benzene exhibits substantial drift during simulation. In our data, the maximum

Table 3: Mean Squared Error (MSE) ($\times 10^{-3}$ nm^2) on the alanine dipeptide dataset. Best results are in **bold**.

Model	**MSE** ($\times 10^{-3}$ nm^2)
NDCN	12.27 ± 0.19
ITO	26.95 ± 0.19
EGNO	6.92 ± 0.26
EGNN	5.67 ± 0.08
Ours	$\mathbf{4.48} \pm 0.07$

x-coordinate is 197.981 Å, while the minimum x-coordinate is -178.112 Å, indicating large translations and rotations of the entire molecule. In contrast, other molecules exhibit minimal net translation, typically remaining within ±3Å from the origin. Methods that enforce strict invariance to translations and rotations (e.g., EGNN) may underperform on such drifting systems, since the global drift is part of the actual dynamics. Indeed, we discovered that *replacing EGNN-based layers with standard message passing layers* (e.g., SAGEConv) *can further boost performance on drifting molecules* like Benzene. We provide details of the dataset statistics in Table 6 of Appendix A, and experiment results using different GNN architectures in Table 9 of Appendix B.

Experimental Results on Alanine Dipeptide. In addition, from Table 3, our model achieves the lowest MSE among all compared methods. This indicates a significant improvement on more complex molecular dynamics data and demonstrates the robustness of our approach.

Analysis of Molecular Structure Recovery. To further evaluate predictions at a structural level, we analyzed the bond lengths and bond angles for alanine dipeptide. Table 2 shows the Mean Absolute Errors (MAEs) and relative errors for bond lengths and angles. Our method achieves the lowest errors, indicating superior recovery of internal molecular structures compared to baselines.

TICA Analysis. Time-lagged Independent Component Analysis (TICA)(Molgedey & Schuster, 1994) is used to extract slow collective motions from the trajectories. Figure 5 shows that, across various numbers of TICA components, our model consistently achieves lower TICA scores (i.e., better alignment with the underlying slow modes) compared to EGNN and EGNO, indicating more effective capture of the underlying multiscale dynamics.

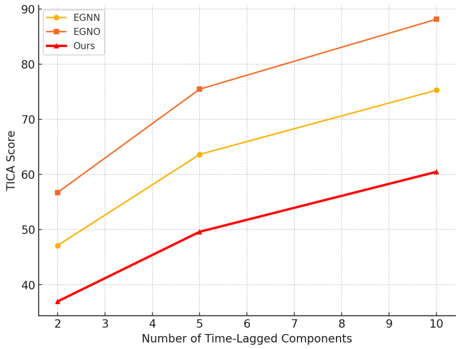

Figure 5: TICA scores for varying numbers of components. Lower scores indicate better alignment with slow modes.

4.5 Continuous-Time Dynamics and Temporal Super-Resolution (RQ2)

Ablation on continuous time components. To assess our model's ability to capture temporal multiscale dynamics, we perform an ablation study on the continuous-time components. Specifically, we investigate:

1. **No ODEs Evolution:** Removing the continuous-time evolution altogether.
2. **No ODEs on Scalar Channels:** Freeze the ODEs for among scalar features $\mathbf{h}$ during time propagation.
3. **No ODEs on Vector Channels:** Freeze the ODEs among vector features $\mathbf{x}$ during time propagation.

Table 4 reports the test errors (MSE) for these variants across several molecules. The inferior performance of the ablated models confirms the importance of the continuous-time dynamics and the effectiveness of the block-diagonal architecture that jointly propagates scalar and vector features.

Table 4: Ablation study on continuous-time evolution components (MSE $\times 10^{-2}$ Å^2 values). Lower values indicate better performance. All results are **inferior** to our standard model (w/ ode).

	Aspirin	Benzene	Ethanol	Malonaldehyde	Naphthalene	Salicylic	Toluene	Uracil
no_ode	6.56±0.03	1.87±0.08	3.09±0.05	11.58±0.02	0.42±0.02	0.86±0.01	3.06±0.05	0.61±0.01
no_ode_h	6.77±0.05	1.89±0.06	3.11±0.03	10.70±0.02	0.42±0.01	0.88±0.02	3.48±0.04	0.59±0.02
no_ode_x	6.95±0.03	1.79±0.07	3.74±0.05	10.61±0.03	0.42±0.01	0.87±0.01	2.89±0.04	0.59±0.01
w/ ode	**6.46**±0.03	**1.52**±0.08	**2.74**±0.05	**10.54**±0.01	**0.23**±0.02	**0.63**±0.01	**1.80**±0.05	**0.41**±0.01

Super-Resolution Task. To further validate temporal generalization, we perform a super-resolution experiment on the alanine dipeptide dataset by predicting trajectories at a 10× finer temporal resolution than the training samples, under the equi-timestep setting. Figure 6 compares the MSE for the original versus the super-resolved predictions. Our model maintains low error under super-resolution, demonstrating its ability to interpolate continuous dynamics effectively.

Long-Term Prediction Stability. We evaluate the stability of long-term predictions. We test models at 1000, 2000, 3000, 4000, and 5000 simulation steps on two representative molecules (Benzene and Malonaldehyde). Figure 7 illustrates how errors evolve over these extended horizons. Our model maintains superior performance with slower error growth, indicating better capture of global low-frequency dynamics essential for accurate long-term predictions. In contrast, baselines lacking explicit multiscale modeling accumulate errors more rapidly.

Table 4 in Appendix B provides additional ablations on the different types of temporal embeddings used in the ODE functions. A simple concatenation of the timestamp would work well enough.

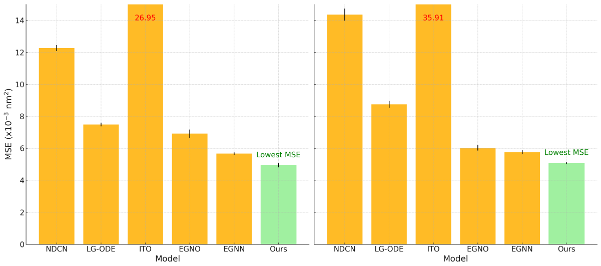

Figure 6: Comparison of MSE on the alanine dipeptide dataset: Original vs. 10× Super-resolution.

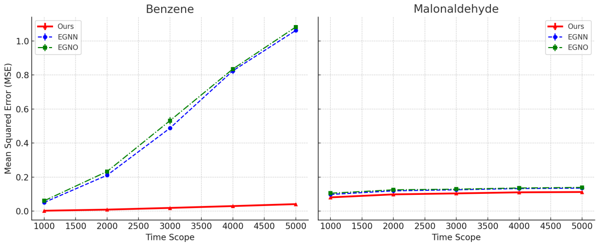

Figure 7: Error growth over long-term forecasts (up to 5000 steps) for Benzene and Malonaldehyde. Models with too high an error not presented here.

4.6 Spatial Frequency Decomposition and Multiscale Analysis (RQ3)

To examine the role of spatial multiscale modeling, we perform ablations on the spectral decomposition and mode interaction components. We consider two sets of modifications:

1. **Ablation of the Spatial Decomposition:**
    - *No Fourier-based Decomposition:* The model is run without any frequency-based transformation. The spectral decomposition is replaced by a multi-layer perception.
    - *Replacing GFT with FFT:* The Graph Fourier Transform is substituted with a standard Fast Fourier Transform, disregarding the graph structure.
2. **Fourier Mode Interaction Schemes:** We compare different strategies for inter-mode communication:
    - *Attention-based Interaction:* Each Fourier mode interacts with others via a multi-head self-attention mechanism (Used in our standard model).
    - *Concatenation-based Interaction:* Modes are concatenated before being processed.
    - *No Interaction:* Each mode is propagated independently.

Table 5 compiles the results for these spatial ablations. We observe that using the GFT for spectral decomposition—rather than an MLP or FFT—is most effective, underscoring the importance of capturing the inherent graph structure. Moreover, interaction schemes (whether via attention or concatenation) improve performance over treating modes independently.

Table 5: MSE ($\times 10^{-2}$ Å^2) on the MD17 dataset with **irregular** timestep sampling for variants in spatial decomposition and Fourier modes interactions. Best results are in **bold**, the standard model (GFT & Attn.).

	Aspirin	Benzene	Ethanol	Malonaldehyde	Naphthalene	Salicylic	Toluene	Uracil
FFT	6.53±0.03	1.94±0.08	3.36±0.05	10.83±0.01	0.43±0.02	0.88±0.01	3.36±0.05	0.60±0.01
No Fourier	6.63±0.04	1.99±0.10	3.28±0.06	10.68±0.07	0.43±0.02	0.88±0.01	3.73±0.09	0.60±0.05
No Interaction	6.51±0.03	1.78±0.08	3.28±0.05	10.64±0.01	0.41±0.02	0.87±0.02	3.08±0.05	0.59±0.01
Concat.	6.55±0.03	1.76±0.07	3.13±0.05	10.58±0.01	0.43±0.02	0.88±0.01	2.80±0.05	0.58±0.01
GFT & Attn.	**6.46**±0.03	**1.52**±0.08	**2.74**±0.05	**10.54**±0.01	**0.23**±0.02	**0.63**±0.01	**1.80**±0.05	**0.41**±0.01

Impact of the Number of Fourier Modes. Finally, we investigate how the number of retained Fourier modes affects performance. Figure 8 plots the MSE against different numbers of modes used in the spectral decomposition. We observe that the performance improves as more modes are included up to a threshold, beyond which additional modes yield diminishing returns. This behavior is consistent with the theoretical analysis presented earlier in Theorem 3.2. The number of modes used to get the optimal results for each type of molecule can be found in Table 6 of Appendix A.

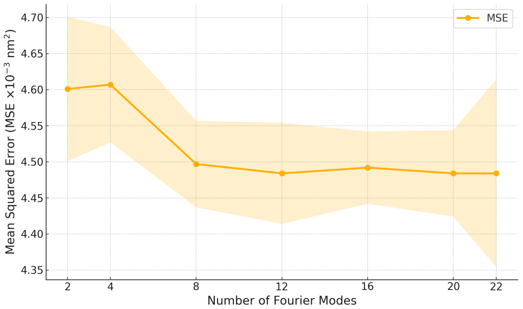

Figure 8: Effect of the number of Fourier modes on prediction performance. Performance plateaus beyond $n = 12$.

5 Conclusion

In this work, we presented Graph Fourier Neural ODEs (GF-NODE), a novel framework that unifies spatial spectral decomposition with continuous-time evolution to effectively model the multiscale dynamics inherent in molecular systems. By decomposing molecular graphs via the Graph Fourier Transform, our approach explicitly disentangles global conformational changes from local vibrational modes, and the subsequent Neural ODE-based propagation enables flexible, continuous-time forecasting of these dynamics.

Our extensive evaluations on benchmark datasets, including MD17 and alanine dipeptide, demonstrate that GF-NODE achieves state-of-the-art performance in long-horizon trajectory prediction while accurately preserving essential molecular geometries. The ablation studies further highlight the critical role of both the spectral decomposition and the continuous-time dynamics in capturing complex spatial-temporal interactions.

Overall, our results suggest that integrating spectral methods with neural ODEs holds significant promise for advancing MD simulation. We anticipate that this approach will not only improve the predictive accuracy and robustness of molecular simulations but also inspire future research on applying similar multiscale modeling techniques to other dynamic, graph-structured systems.

6 Broader Impact Statement

Our work contributes to advancing molecular dynamics simulations, which have broad implications in scientific discovery, particularly in drug design, materials science, and biomolecular modeling. By improving the accuracy and efficiency of multiscale molecular predictions, our approach could accelerate the discovery of novel therapeutics and facilitate the design of functional materials with tailored properties.

While our method relies on data-driven modeling, it does not replace physics-based simulations but rather augments them, reducing computational costs while maintaining interpretability. As with any machine learning-driven approach in scientific domains, care must be taken to ensure model reliability, particularly in high-stakes applications such as drug development. Further validation and collaboration with domain experts will be essential to maximize the positive societal impact of our work while mitigating risks related to model uncertainty.

References

Michael Bear and Estela Blaisten-Barojas. Atomic and ion collisions in solids and at surfaces: Theory, simulation and applications. *American Scientist*, 86(2):188, 1998.

Bernard R Brooks, Charles L Brooks III, Alexander D Mackerell Jr, Lennart Nilsson, Robert J Petrella, Benoît Roux, Youngdo Won, Georgios Archontis, Christian Bartels, Stefan Boresch, et al. Charmm: the biomolecular simulation program. *Journal of computational chemistry*, 30(10):1545–1614, 2009.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Stefan Chmiela, Alexandre Tkatchenko, Huziel E Sauceda, Igor Poltavsky, Kristof T Schütt, and Klaus-Robert Müller. Machine learning of accurate energy-conserving molecular force fields. *Science advances*, 3(5): e1603015, 2017.

Fan R. K. Chung. *Spectral Graph Theory.* American Mathematical Society, Providence, R.I., 1997.

Wendy D Cornell, Piotr Cieplak, Christopher I Bayly, Ian R Gould, Kenneth M Merz, David M Ferguson, David C Spellmeyer, Thomas Fox, James W Caldwell, and Peter A Kollman. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. *Journal of the American Chemical Society*, 117(19):5179–5197, 1995.

Pawan Goyal and Peter Benner. Neural ordinary differential equations with irregular and noisy data. *Royal Society Open Science*, 10(7):221475, 2023.

Samuel I Holt, Zhaozhi Qian, and Mihaela van der Schaar. Neural laplace: Learning diverse classes of differential equations in the laplace domain. In *International Conference on Machine Learning*, pp. 8811–8832. PMLR, 2022.

Zijie Huang, Yizhou Sun, and Wei Wang. Learning continuous system dynamics from irregularly-sampled partial observations. *Advances in Neural Information Processing Systems*, 33:16177–16187, 2020.

Zijie Huang, Yizhou Sun, and Wei Wang. Coupled graph ode for learning interacting system dynamics. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pp. 705–715, 2021.

Zijie Huang, Yizhou Sun, and Wei Wang. Generalizing graph ode for learning complex system dynamics across environments. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 798–809, 2023a.

Zijie Huang, Wanjia Zhao, Jingdong Gao, Ziniu Hu, Xiao Luo, Yadi Cao, Yuanzhou Chen, Yizhou Sun, and Wei Wang. Tango: Time-reversal latent graphode for multi-agent dynamical systems. *arXiv preprint arXiv:2310.06427*, 2023b.

Jürg Hutter. Car–parrinello molecular dynamics. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 2(4):604–612, 2012.

Patrick Kidger. On neural differential equations. *arXiv preprint arXiv:2202.02435*, 2022.

Suyong Kim, Weiqi Ji, Sili Deng, Yingbo Ma, and Christopher Rackauckas. Stiff neural ordinary differential equations. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(9), 2021.

Leon Klein, Andrew Foong, Tor Fjelde, Bruno Mlodozeniec, Marc Brockschmidt, Sebastian Nowozin, Frank Noé, and Ryota Tomioka. Timewarp: Transferable acceleration of molecular dynamics by learning time-coarsened dynamics. *Advances in Neural Information Processing Systems*, 36, 2024.

Takeshi Koshizuka, Masahiro Fujisawa, Yusuke Tanaka, and Issei Sato. Understanding the expressivity and trainability of fourier neural operator: A mean-field perspective. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems.*

Nikola Kovachki, Samuel Lanthaler, and Siddhartha Mishra. On universal approximation and error bounds for fourier neural operators. *Journal of Machine Learning Research*, 22(290):1–76, 2021.

Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

Z Li, DZ Huang, B Liu, and A Anandkumar. Fourier neural operator with learned deformations for pdes on general geometries. arxiv, 2022a.

Zijie Li, Kazem Meidani, Prakarsh Yadav, and Amir Barati Farimani. Graph neural networks accelerated molecular dynamics. *The Journal of Chemical Physics*, 156(14), 2022b.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

Ning Liu and Siavash Jafarzadeh. Yue yu. domain agnostic fourier neural operators. *arXiv preprint arXiv:2305.00478*, 2023.

Xiao Luo, Yiyang Gu, Huiyu Jiang, Hang Zhou, Jinsheng Huang, Wei Ju, Zhiping Xiao, Ming Zhang, and Yizhou Sun. Pgode: Towards high-quality system dynamics modeling. In *Forty-first International Conference on Machine Learning*.

Xiao Luo, Jingyang Yuan, Zijie Huang, Huiyu Jiang, Yifang Qin, Wei Ju, Ming Zhang, and Yizhou Sun. Hope: High-order graph ode for modeling interacting dynamics. In *International Conference on Machine Learning*, pp. 23124–23139. PMLR, 2023.

Lutz Molgedey and Heinz Georg Schuster. Separation of a mixture of independent signals using time delayed correlations. *Physical review letters*, 72(23):3634, 1994.

Outi MH Salo-Ahen, Ida Alanko, Rajendra Bhadane, Alexandre MJJ Bonvin, Rodrigo Vargas Honorato, Shakhawath Hossain, André H Juffer, Aleksei Kabedev, Maija Lahtela-Kakkonen, Anders Støttrup Larsen, et al. Molecular dynamics simulations in drug discovery and pharmaceutical development. *Processes*, 9(1): 71, 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. *Advances in Neural Information Processing Systems*, 36, 2024.

Paulo CT Souza, Riccardo Alessandri, Jonathan Barnoud, Sebastian Thallmair, Ignacio Faustino, Fabian Grünewald, Ilias Patmanidis, Haleh Abdizadeh, Bart MH Bruininks, Tsjerk A Wassenaar, et al. Martini 3: a general purpose force field for coarse-grained molecular dynamics. *Nature methods*, 18(4):382–388, 2021.

Antonis I Vakis, Vladislav A Yastrebov, Julien Scheibert, Lucia Nicola, Daniele Dini, Clotilde Minfray, Andreas Almqvist, Marco Paggi, Seunghwan Lee, Georges Limbert, et al. Modeling and simulation in tribology across scales: An overview. *Tribology International*, 125:169–199, 2018.

Quan-De Wang, Jing-Bo Wang, Juan-Qin Li, Ning-Xin Tan, and Xiang-Yuan Li. Reactive molecular dynamics simulation and chemical kinetic modeling of pyrolysis and combustion of n-dodecane. *Combustion and Flame*, 158(2):217–226, 2011.

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. *arXiv preprint arXiv:2401.11037*, 2024.

Chengxi Zang and Fei Wang. Neural dynamics on complex networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 892–902, 2020.

A Experiment Setup

In this section, we describe key details about the datasets used (MD17 and alanine dipeptide), including the simulation step sizes, and then outline the hyperparameter choices and training procedures.

A.1 Datasets

We use the MD17 dataset (Chmiela et al., 2017) for small-molecule dynamics, and the alanine dipeptide dataset (Schreiner et al., 2024) for conformational analysis in proteins. Statistics for MD17 are provided in Table 6. From the table, we observe Benzene has a very marked drifting that is much larger that the scale of the molecule itself. The MD17 dataset was generated using ab initio molecular dynamics (MD) simulations with a time step of 0.5 femtoseconds (fs). The simulations were conducted in the NVT ensemble at 500 K for a total duration of 2000 picoseconds (ps). The final dataset was created by subsampling the full trajectory, preserving the Maxwell-Boltzmann distribution for the energies. For the alanine dipeptide dataset, we use the protein fragment's trajectory recorded at a step size of 1.0 ps.

Table 6: Summary statistics for molecular structures, including the number of atoms, position extrema ($\mathrm{X}_{\mathrm{min}}$, $\mathrm{X}_{\mathrm{max}}$, $\mathrm{X}_{\mathrm{mean}}$), and velocity extrema ($\mathrm{V}_{\mathrm{min}}$, $\mathrm{V}_{\mathrm{max}}$, $\mathrm{V}_{\mathrm{mean}}$). The numbers of modes used to get the optimal results are also listed.

Molecule	**#Atoms**	**#Modes used**	$\mathbf{X}_{\mathbf{min}}$	$\mathbf{X}_{\mathbf{max}}$	$\mathbf{X}_{\mathbf{mean}}$	$\mathbf{V}_{\mathbf{min}}$	$\mathbf{V}_{\mathbf{max}}$	$\mathbf{V}_{\mathbf{mean}}$
benzene	6	4	−178.112	197.981	−27.737	−0.004	0.003	−0.000
aspirin	13	6	−3.720	3.105	0.026	−0.011	0.012	0.000
ethanol	3	3	−1.398	1.417	−0.004	−0.011	0.010	−0.000
malonaldehyde	5	5	−2.397	2.370	0.000	−0.010	0.009	0.000
naphthalene	10	4	−2.597	2.593	−0.000	−0.012	0.011	0.000
salicylic	10	8	−2.734	2.581	−0.051	−0.013	0.012	−0.000
toluene	7	7	−1.990	2.630	−0.015	−0.010	0.012	0.000
uracil	8	6	−2.338	2.558	0.012	−0.012	0.011	0.000

A.2 Training Setup

Hyperparameters. We train all models using the Adam optimizer at a learning rate of 1×10^{-4} and apply a weight decay of 1×10^{-15} for regularization. Each experiment runs for 5000 epochs, processing batches of size 50 at each training step. For molecular trajectory prediction, we set the sequence length (the number of timesteps) to 8, meaning each training sample contains 8 frames from the overall simulation. In our main experiments, we fix the hidden feature dimensionality to 64 and use a maximum future horizon of 3000 simulation steps when constructing training samples. We also specify the dopri5 ODE solver with relative and absolute tolerances of 1×10^{-3} and 1×10^{-4}, respectively, to integrate the continuous-time model components. Although many of these choices (e.g., total layers, solver tolerances) can be altered, we find these particular settings maintain a good balance of accuracy and computational efficiency.

Training Procedure. During training, each mini-batch is formed by sampling short segments of length 8 from the molecule's dynamics trajectory. The model then predicts future positions of atoms after continuous-time evolution, and the Mean Squared Error (MSE) between predicted coordinates and ground-truth coordinates is minimized. We checkpoint models whenever validation performance improves, and at the end of training, we report results using the best-performing checkpoint according to the validation set. In addition, regular evaluations on the test set help track the model's generalization to unseen trajectories. The overall process can be summarized as: (1) load training samples, (2) form mini-batches of molecular frames, (3) perform forward pass through the model to generate predictions, (4) compute the MSE loss, (5) update model parameters via backpropagation, repeating for each epoch until convergence.

By following these procedures with our chosen hyperparameters, we have observed stable convergence across both MD17 and alanine dipeptide datasets, as well as strong generalization to different segments of the trajectory during test runs.

A.3 Loss Function

To train the model, we use the Mean Squared Error (MSE) loss, which measures the difference between the predicted atomic positions and the ground truth positions at each predicted time point. Given that the goal is to predict the molecular conformations at time points $t_1, t_2, \ldots, t_K$, the MSE is calculated as follows:

Let $\mathbf{x}_i^{t_j} \in \mathbb{R}^3$ be the ground truth 3D coordinates of atom i at time t_j, and $\tilde{\mathbf{x}}_i^{t_j} \in \mathbb{R}^3$ be the predicted coordinates for the same atom at time t_j. The MSE loss is defined as:

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{NK} \sum_{j=1}^{K} \sum_{i=1}^{N} \left\| \mathbf{x}_i^{t_j} - \tilde{\mathbf{x}}_i^{t_j} \right\|_2^2, \tag{24}$$

where N is the number of atoms and K is the number of time points.

This loss encourages the model to minimize the Euclidean distance between the predicted and actual atomic positions across all time steps, ensuring accurate trajectory prediction for the molecular system.

The MSE loss is applied at each time point, thus aligning the predicted future states with the true molecular dynamics trajectory. The model is trained by minimizing $\mathcal{L}_{\mathrm{MSE}}$ over all predicted time points.

B Additional Experiment Results

Below, we provide four tables corresponding to different experimental comparisons. The first table reports performance on the MD17 dataset with regular (equi-) timesteps. The remaining three tables focus on ablation experiments conducted under irregular sampling conditions: (i) ablations on the alanine dipeptide dataset, (ii) ablations comparing different GNN architectures, and (iii) ablations on different temporal embedding approaches.

Table 7: MD17 with Regular (Equi-) Timestep Sampling. MSE ($\times 10^{-2}$ Å^2) on the MD17 dataset using regular timesteps. Best results are in **bold**, and second-best are underlined.

	Aspirin	Benzene	Ethanol	Malonaldehyde	Naphthalene	Salicylic	Toluene	Uracil
NDCN	31.73±0.40	56.21±0.30	10.74±0.02	46.55±0.28	2.25±0.01	3.58±0.11	13.92±0.02	2.38±0.00
LG-ODE	19.36±0.12	53.92±1.32	7.08±0.01	24.41±0.03	1.73±0.02	3.82±0.04	11.18±0.01	2.11±0.02
EGNN	9.24±0.07	57.85±2.70	4.63±0.00	12.81±0.01	0.38±0.01	0.85±0.00	10.41±0.04	0.56±0.02
EGNO	9.41±0.09	55.13±3.21	4.63±0.00	12.81±0.01	0.40±0.01	0.93±0.01	10.43±0.10	0.59±0.01
ITO	20.56±0.03	57.25±0.58	8.60±0.27	28.44±0.73	1.82±0.17	2.48±0.34	12.47±0.30	1.33±0.12
Ours	**6.07**±0.09	**1.51**±0.07	**2.74**±0.01	**9.43**±0.02	**0.24**±0.02	**0.63**±0.05	**1.80**±0.03	**0.41**±0.02

Explanation. Table 7 shows results for MD17 under a regular (evenly spaced) sampling scheme. Although the dataset inherently has fine-grained timesteps, we constrain both training and evaluation to equidistant frames to compare methods fairly. Our approach demonstrates consistent improvements over baselines on nearly all molecules.

Explanation. Table 8 organizes the ablation settings into two rows, each containing five columns. The first row compares our "standard" model to variants that remove specific ODE blocks or modify scalar/vector-only ODE updates ("no_ode","no_ode_h", "no_ode_x"), and the second row compares different interaction modes, time embeddings, and Fourier settings. The "standard" configuration achieves the best overall MSE.

Explanation. Table 9 shows how our model performs with different GNN backbones on MD17 under irregular sampling. Overall, SAGEConv yields robust performance for most molecules, whereas GCNConv

Table 8: **Ablation Results on Alanine Dipeptide (Irregular Sampling).** MSE ($\times 10^{-3}$ nm^2). Best results in **bold**.

	standard	no_ode	no_ode_h	no_ode_x	no_interaction
MSE	**4.48**±0.07	4.72±0.05	4.60±0.05	4.64±0.05	4.80±0.04
	interaction_concat	time_posenc	time_mlp	FFT	no_fourier
MSE	4.98±0.06	4.62±0.05	4.60±0.05	4.57±0.05	4.51±0.04

Table 9: **Ablation on GNN Architectures (Irregular Sampling).** MSE ($\times 10^{-2}$ Å^2) on MD17 comparing different GNN layers (SAGEConv, GCNConv, EGNNConv). Best results in **bold**.

	Aspirin	Benzene	Ethanol	Malonaldehyde	Naphthalene	Salicylic	Toluene	Uracil
SAGEConv	**6.46**±0.03	1.79±0.08	**2.74**±0.05	**10.54**±0.01	**0.23**±0.02	**0.63**±0.01	3.08±0.05	**0.41**±0.01
GCNConv	6.91±0.02	**1.52**±0.08	3.09±0.06	10.85±0.03	0.42±0.01	0.88±0.00	**1.80**±0.05	0.61±0.02
EGNNConv	8.85±0.02	40.86±0.98	4.41±0.06	12.49±0.00	0.40±0.01	0.87±0.01	8.63±0.04	0.62±0.02

provides better results specifically on Benzene and Toluene. EGNNConv performs well on some local metrics but struggles on large translations (i.e., Benzene).

Table 10: **Ablation on Time Embedding Approaches (Irregular Sampling).** MSE ($\times 10^{-2}$ Å^2) on MD17 across different time encoding schemes.

	Aspirin	Benzene	Ethanol	Malonaldehyde	Naphthalene	Salicylic	Toluene	Uracil
posenc	6.91±0.08	1.81±0.02	3.11±0.05	10.69±0.03	4.19±0.04	0.87±0.01	3.56±0.07	0.59±0.02
mlp	6.61±0.06	1.61±0.03	3.08±0.02	10.62±0.04	0.41±0.01	0.87±0.02	3.25±0.06	0.56±0.01
concat	**6.46**±0.03	**1.52**±0.08	**2.74**±0.05	**10.54**±0.01	**0.23**±0.02	**0.63**±0.01	**1.80**±0.05	**0.41**±0.01

Explanation. Table 10 compares three different time-embedding methods under irregular timestep sampling: positional encoding (posenc), a small MLP (mlp), and a direct concatenation of time tokens (concat). Concatenation achieves the lowest MSE, suggesting that a straightforward inclusion of time in the feature vector can be beneficial, though the MLP variant also achieves competitive performance on several molecules.

C Theoretical Guarantees on Spectral Decompositions

Below, we present a concise mathematical exposition on the theoretical underpinnings of the Graph Fourier Transform (GFT) decomposition used in our framework. We explain how the eigenvalue–eigenvector structure of the graph Laplacian $\mathbf{L}$ induces a decomposition of graph signals into low-frequency (global) and high-frequency (local) modes, and we justify truncating to the first M modes. We follow standard nomenclature in spectral graph theory (Chung, 1997).

C.1 Preliminaries and Definitions

Definition C.1 (Graph Laplacian)**.** Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph with $N = |\mathcal{V}|$ vertices. Let $\mathbf{A} \in \mathbb{R}^{N \times N}$ be its adjacency matrix, and let $\mathbf{D}$ be the diagonal degree matrix, where

$$\mathbf{D}(i,i) \;=\; \sum_{j=1}^{N} \mathbf{A}(i,j). \tag{25}$$

The *graph Laplacian* is defined as

$$\mathbf{L} \;=\; \mathbf{D} \;-\; \mathbf{A}. \tag{26}$$

It is well known that $\mathbf{L}$ is real symmetric and positive semidefinite. In fact, the eigenvalues of the Laplacian matrix are real and non-negative.

Definition C.2 (Graph Fourier Transform (GFT))**.** Given the eigen-decomposition

$$\mathbf{L} \;=\; \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^\top, \tag{27}$$

where

$$\boldsymbol{\Lambda} \;=\; \operatorname{diag}(\lambda_0, \lambda_1, \ldots, \lambda_{N-1}), \quad 0 = \lambda_0 \le \lambda_1 \le \cdots \le \lambda_{N-1}, \tag{28}$$

and $\mathbf{U} = \big[\mathbf{u}_0 \mid \mathbf{u}_1 \mid \cdots \mid \mathbf{u}_{N-1}\big]$ stores the corresponding orthonormal eigenvectors in columns. For a graph signal

$$\mathbf{x} \;=\; \big(x_1, x_2, \ldots, x_N\big)^\top \in \mathbb{R}^N, \tag{29}$$

the *Graph Fourier Transform (GFT)* of $\mathbf{x}$ is given by

$$\widehat{\mathbf{x}} \;=\; \mathbf{U}^\top \mathbf{x}, \tag{30}$$

and the *inverse GFT* is

$$\mathbf{x} \;=\; \mathbf{U}\widehat{\mathbf{x}}. \tag{31}$$

C.2 Truncation and Mode Selection

Lemma C.3 (Approximation Error for Spectral Truncation)**.** *Let* $\mathbf{x} \in \mathbb{R}^N$ *be any graph signal, and let* $\mathbf{x}_{(M)}$ *be its spectral approximation obtained by keeping the first* M *modes. Then*

$$\|\mathbf{x} - \mathbf{x}_{(M)}\|_2^2 \;=\; \sum_{k=M}^{N-1} \big|\mathbf{u}_k^\top \mathbf{x}\big|^2. \tag{32}$$

Moreover, if $\mathbf{x}$ *is* α*-bandlimited in the sense that*

$$\mathbf{u}_k^\top \mathbf{x} \;=\; 0 \quad \text{for all } \lambda_k > \alpha, \tag{33}$$

then choosing M *such that* $\lambda_{M-1} \le \alpha$ *yields an exact recovery* $\mathbf{x} = \mathbf{x}_{(M)}$.

Proof. See the main text for details. We expand the signal in the Laplacian eigenbasis $\{\mathbf{u}_k\}$, and observe that discarding all modes with $k \ge M$ removes the corresponding frequency components. ☐

C.3 Low-Frequency vs. High-Frequency Modes

Because $\mathbf{L}$ is positive semidefinite and the eigenvalues $\{\lambda_i\}$ increase with i, smaller eigenvalues correspond to slow, global variations, while larger eigenvalues capture more oscillatory, local phenomena.

Proposition C.4 (Global vs. Local Spatial Scales)**.** *Let* $\mathbf{u}_k$ *be the* k*-th eigenvector of* $\mathbf{L}$ *with eigenvalue* λ_k*. Suppose* $\mathbf{x}$ *encodes atomic coordinates or their latent features. Then:*

1. *If* λ_k *is small, the corresponding mode* $\mathbf{u}_k$ *represents slowly varying (global) deformations across the molecule.*
2. *If* λ_k *is large, the corresponding mode* $\mathbf{u}_k$ *represents rapidly changing (local) structural variations.*

Proof. From standard results in spectral graph theory (Chung, 1997). The low-frequency (small λ) modes vary smoothly across edges, whereas high-frequency (large λ) modes exhibit large differences across edges. ☐

C.4 Practical Mode Truncation Criteria

Definition C.5 (Mode Retention Threshold)**.** For a desired tolerance $\epsilon > 0$, select M such that

$$\sum_{k=M}^{N-1} \left|\mathbf{u}_k^\top \mathbf{x}\right|^2 \;\le\; \epsilon \, \|\mathbf{x}\|_2^2. \tag{34}$$

In practice, one may also pick M based on $\lambda_{M-1} \le \alpha$, ignoring modes where $\lambda_k > \alpha$.

Corollary C.6 (Error Control via Low-Pass Approximation)**.** *Under the same notation as above, if*

$$\sum_{k=M}^{N-1} \left|\mathbf{u}_k^\top \mathbf{x}\right|^2 \;\le\; \epsilon \, \|\mathbf{x}\|_2^2, \tag{35}$$

then

$$\|\mathbf{x} - \mathbf{x}_{(M)}\|_2 \;\le\; \epsilon \, \|\mathbf{x}\|_2. \tag{36}$$

Hence, discarding high-frequency modes exceeding this threshold leads to a bounded approximation error.

The following sections are added during rebuttal

D Formal Proof of SO(3)-Equivariance for GF-NODE Pipeline

Below is a **formal proof** of SO(3) (rotational) equivariance for our **GF-NODE** pipeline, closely following the style of EGNO's Appendix proofs. We focus on the **3D rotational** part of SE(3); translations can be handled by the separate mean-centering step (see remarks below). Our proof is broken down into:

1. **Defining the R-action**,
2. **Showing that each module** (Fourier transforms, block-diagonal ODE, EGNN layers) **is** SO(3)**-equivariant**, and
3. **Composing these results** to conclude overall equivariance.

D.1 Formal Statement of SO(3)-Equivariance

Let

$$f = \left[f_{\mathbf{h}}, f_{\mathbf{Z}}\right]^{\top} \tag{37}$$

be a function describing the node features of a 3D molecular system over some (possibly temporal) domain D. Concretely,

- $f_{\mathbf{h}} : D \to \mathbb{R}^{N\times k}$ collects **invariant (scalar) node features**,
- $f_{\mathbf{Z}} : D \to \mathbb{R}^{N\times(m\times 3)}$ collects **equivariant (3D) features** (positions, velocities, etc.).

Denote by $\mathbf{R} \in \mathrm{SO}(3)$ a 3D rotation matrix. The **action** of $\mathbf{R}$ on f is defined by

$$(\mathbf{R}\cdot f)(t) = \left[f_{\mathbf{h}}(t), \mathbf{R} f_{\mathbf{Z}}(t)\right]^{\top}, \tag{38}$$

which rotates only the $\mathbf{Z}$-component in $\mathbb{R}^3$ and leaves the scalar $\mathbf{h}$-component invariant.

We claim that our overall **GF-NODE operator** $\mathcal{T}_\theta$ satisfies

$$\mathbf{R}\cdot\mathcal{T}_\theta(f) = \mathcal{T}_\theta\big(\mathbf{R}\cdot f\big), \tag{39}$$

i.e., $\mathcal{T}_\theta$ is SO(3)-equivariant. Formally:

Theorem D.1 (SO(3) Equivariance)**.** *Let $\mathcal{T}_\theta$ be the GF-NODE architecture composed of:*

1. *An* ***EGNN encoder*** *(mapping $[f_{\mathbf{h}}, f_{\mathbf{Z}}] \to$ encoded features),*
2. ***Mean-centering*** *and* ***Graph Fourier Transform*** *($\mathcal{F}$),*
3. *A* ***block-diagonal Neural ODE*** *in the spectral domain,*
4. ***Inverse GFT*** *($\mathcal{F}^{-1}$) plus adding back the mean, and*
5. *An* ***EGNN decoder****.*

Then for any $\mathbf{R} \in \mathrm{SO}(3)$, the pipeline satisfies

$$\mathcal{T}_\theta\big(\mathbf{R}\cdot f\big) = \mathbf{R}\cdot\mathcal{T}_\theta(f). \tag{40}$$

In other words, rotating the input 3D features by $\mathbf{R}$ is equivalent to applying $\mathcal{T}_\theta$ first and then rotating the result.

We prove this via the following steps:

1. **Lemma 1**: EGNN layers are SO(3)-equivariant.
2. **Lemma 2**: GFT and its inverse are SO(3)-equivariant (dimension-wise linearity).
3. **Lemma 3**: The block-diagonal Neural ODE in spectral space preserves **R**-equivariance on the vector channels.
4. **Conclusion**: Composing these yields the full pipeline's equivariance.

Below, we provide the details of each lemma and then the final proof of the Theorem.

D.2 EGNN Equivariance

Lemma D.2 (EGNN layers are SO(3)-equivariant)**.** *Consider a generic EGNN layer* Φ*, which updates*

$$(\mathbf{h}_i, \mathbf{x}_i) \mapsto (\mathbf{h}'_i, \mathbf{x}'_i), \tag{41}$$

using message passing:

$$\mathbf{m}_{ij} = \phi_e\big(\mathbf{h}_i, \mathbf{h}_j, \mathbf{x}_i - \mathbf{x}_j\big), \tag{42}$$

$$\mathbf{h}'_i = \phi_h\Big(\mathbf{h}_i, \sum_j \mathbf{m}_{ij}\Big), \tag{43}$$

$$\mathbf{x}'_i = \mathbf{x}_i + \ldots (\mathbf{x}_i - \mathbf{x}_j). \tag{44}$$

Then for any rotation $\mathbf{R} \in \mathrm{SO}(3)$*,*

$$\Phi\big(\mathbf{R}\,\mathbf{x}_i, \mathbf{h}_i\big) = \big(\mathbf{h}'_i, \mathbf{R}\,\mathbf{x}'_i\big). \tag{45}$$

Hence Φ *is* SO(3)*-equivariant on its 3D inputs.*

A standard proof () shows that each update depends on $\mathbf{x}_i - \mathbf{x}_j$, which under a global rotation $\mathbf{R}(\mathbf{x}_i - \mathbf{x}_j)$ transforms consistently to yield $\mathbf{R}\,\mathbf{x}'_i$. The same argument applies to 3D velocities (or any additional 3D vectors).

D.3 GFT Equivariance

We next show that the (inverse) Graph Fourier Transform is SO(3)-equivariant with respect to dimension-wise rotations of the 3D features.

Lemma D.3 (GFT and $\mathcal{F}^{-1}$ are SO(3)-equivariant))**.** *Let* $\mathcal{F}$ *be the dimension-wise GFT mapping a function*

$$f_{\mathbf{Z}} : D \to \mathbb{R}^{N\times(m\times 3)} \tag{46}$$

to its frequency coefficients $\mathcal{F}(f_{\mathbf{Z}}) \in \mathbb{C}^{(modes)\times m\times 3}$*. Under* $\mathbf{R} \in \mathrm{SO}(3)$*, define*

$$\mathbf{R}\,\cdot\,\big(\mathcal{F} f_{\mathbf{Z}}\big) = \mathcal{F} f_{\mathbf{Z}} \textit{ but with each 3-D channel rotated by } \mathbf{R}. \tag{47}$$

Then

$$\mathbf{R}\,\cdot\,\mathcal{F}(f_{\mathbf{Z}}) = \mathcal{F}\Big(\mathbf{R}\,\cdot\,f_{\mathbf{Z}}\Big). \tag{48}$$

Similarly, $\mathcal{F}^{-1}$ *is* SO(3)*-equivariant in the sense that*

$$\mathcal{F}^{-1}\big(\mathbf{R}\,\cdot\,F\big) = \mathbf{R}\,\cdot\,\mathcal{F}^{-1}(F). \tag{49}$$

Proof Sketch. The GFT (and its inverse) act linearly along each 3D axis. If $\mathbf{R}$ rotates the 3D channels, we can commute $\mathbf{R}$ with the linear transform $\mathcal{F}$. Precisely as in the EGNO proof, the multilinear expansions show that $\mathbf{R}\cdot\mathcal{F}(f_{\mathbf{Z}}) = \mathcal{F}\big(\mathbf{R}\cdot f_{\mathbf{Z}}\big)$. The same argument applies to $\mathcal{F}^{-1}$ because it is also linear and dimension-wise.

D.4 Block-Diagonal Neural ODE Equivariance

In the GF-NODE pipeline, once we have GFT coefficients $\tilde{\mathbf{Z}}$, the **Neural ODE** acts as a block-diagonal operator:

$$\begin{pmatrix}\tilde{\mathbf{H}}\\ \tilde{\mathbf{Z}}\end{pmatrix} \mapsto \begin{pmatrix} f_\theta\big(\tilde{\mathbf{H}}\big)\\ g_\theta\big(\tilde{\mathbf{Z}}\big)\end{pmatrix}, \tag{50}$$

where $\tilde{\mathbf{Z}} \in \mathbb{C}^{(\mathrm{modes})\times m\times 3}$. Rotating $\mathbf{R}$ on these 3D channels amounts to mixing the coordinate axes linearly. Because the ODE is chosen to be channelwise or "blockwise" linear or MLP-based, it commutes with $\mathbf{R}$. Hence:

Lemma D.4 (Block-Diagonal ODE is SO(3)-equivariant)**.** *For each frequency mode, the update on* $\tilde{\mathbf{Z}}$ *is dimension-wise (like a separate channel). A global rotation* $\mathbf{R}$ *that mixes* $\tilde{\mathbf{Z}}^1, \tilde{\mathbf{Z}}^2, \tilde{\mathbf{Z}}^3$ *can be factored out of the ODE solution—so*

$$\mathbf{R}\,\cdot\,g_\theta\big(\tilde{\mathbf{Z}}\big) = g_\theta\big(\mathbf{R}\,\cdot\,\tilde{\mathbf{Z}}\big). \tag{51}$$

Integrating over t *preserves this property.*

D.5 Proof of the Main Theorem (SO(3) Equivariance)

Recall our overall operator $\mathcal{T}_\theta$ has the form:

1. **EGNN Encode**: $\big(\mathbf{h}, \mathbf{Z}\big) \mapsto \big(\mathbf{h}^{(L)}, \mathbf{Z}^{(L)}\big)$.
2. **Mean-Center + GFT**: $\mathbf{Z}^{(L)} \mapsto \mathbf{Z}_c^{(L)} \mapsto \tilde{\mathbf{Z}}^{(L)} = \mathcal{F}\big(\mathbf{Z}_c^{(L)}\big)$.
3. **Block-Diagonal ODE**: $\tilde{\mathbf{Z}}^{(L)} \mapsto \tilde{\mathbf{Z}}(t)$ for any t.
4. **Inverse GFT + Add Mean**: $\tilde{\mathbf{Z}}(t) \mapsto \mathbf{Z}_c(t) = \mathcal{F}^{-1}\big(\tilde{\mathbf{Z}}(t)\big) \mapsto \mathbf{Z}(t)$.
5. **EGNN Decode**: $\big(\mathbf{h}, \mathbf{Z}(t)\big) \mapsto \big(\mathbf{h}'(t), \mathbf{Z}'(t)\big)$.

To show $\mathbf{R} \cdot \mathcal{T}_\theta(f) = \mathcal{T}_\theta(\mathbf{R} \cdot f)$, we proceed step-by-step:

1. **EGNN Encode**: By Lemma A.1, if the input positions are replaced with $\mathbf{R}\mathbf{x}_i$, the output is $\mathbf{R}\mathbf{x}_i^{(L)}$.
2. **Mean-Center**: Under a global rotation, the centered coordinates also rotate, i.e., $\mathbf{x}_i^\circ \mapsto \mathbf{R}\,\mathbf{x}_i^\circ$.
3. **GFT**: By Lemma A.2, dimension-wise GFT on $\mathbf{R}\,\mathbf{x}_i^\circ$ yields the rotated spectral coefficients.
4. **Block-Diagonal ODE**: Lemma A.3 says the ODE in spectral space is equivariant w.r.t. 3D axis mixing, so $\mathbf{R}$ commutes with the ODE solution.
5. **Inverse GFT**: Again by Lemma A.2, inverse transforms are linear in each dimension, preserving $\mathbf{R}$ on the output.
6. **Add Mean**: The final global shift (if any) is consistent with $\mathbf{R}$.
7. **EGNN Decode**: By Lemma A.1 again, if the input to the decoder is rotated, the output is the rotated version of the unrotated output.

Hence each sub-module respects the action of $\mathbf{R}$. Composing them in order yields the final statement

$$\mathcal{T}_\theta(\mathbf{R} \cdot f) = \mathbf{R} \cdot \mathcal{T}_\theta(f). \tag{52}$$

This completes the proof of SO(3)-equivariance.

D.6 Remarks on Translations

In practice, **SE(3)** includes translations as well. Our pipeline **removes** the translational degree of freedom by **mean-centering** the positions (the DC mode). A global translation $\mathbf{x}_i \mapsto \mathbf{x}_i + \boldsymbol{\mu}$ simply shifts the mean $\overline{\mathbf{x}}$, so the centered coordinates $\mathbf{x}_i^\circ$ remain unchanged. This effectively "factors out" translation before the GFT steps. When we **re-add** the mean at the end, it ensures the final positions transform by $\mathbf{x}_i \mapsto \mathbf{x}_i + \boldsymbol{\mu}$. Thus the entire pipeline remains **invariant** to translations (i.e., translates its outputs accordingly). For brevity, the above proof focuses on **rotations** $\mathbf{R} \in \mathrm{SO}(3)$; translation invariance follows from the mean-subtraction procedure plus the decoder's reliance on relative positions.

E Rebuttal responses

Response to Reviewer MJAX

We thank the reviewer for the thorough reading of our manuscript and for the constructive feedback. Below we address every point raised. For brevity we quote the reviewer in
textititalic blue and provide our reply in plain text. All table/figure numbers refer to those in the *revised* manuscript.

R1. *"Empirical performance and significance: gains appear marginal compared with EGNN; long–term prediction not convincingly supported."*

Response. We respectfully disagree that the improvements are marginal. On MD17 with irregular time sampling (Table 1) GF–NODE reduces the test MSE by an *average of 53%* relative to the strongest baseline EGNN and by 66% relative to EGNO. On alanine–dipeptide (Table 3) the reduction is 21%. In the revised manuscript we have extended the evaluation to considerably larger systems and to a 5 × longer temporal horizon (up to $\Delta t = 1.5 \times 10^4$ MD steps,[1] This is already within the time range where we can observe the folding behavior of a protein. See Tables 13 and 14). Across *all* nine MD17 molecules and five larger systems (up to 326 heavy atoms, Table 15) GF–NODE remains the best performer, often by a factor ×2. Figure 7 (MD17) and Figure 10 (large molecules) further demonstrate the slower error growth of our model in the long–horizon regime.

R2. *"Evaluation limited to small molecules; need more complex systems."*

Response. We have added five substantially larger and topologically distinct systems: Ala_2, Ac–Ala_3–NHMe, AT–AT–CG–CG, Bucky–Catcher and a double–walled carbon nanotube (DW NT). The results are summarised in Tables 13 (Δt=3000) and 14 (Δt=10000). GF–NODE yields the lowest error on every system and time horizon. These molecules exhibit clear multi–scale behaviour (e.g., nanotube radial breathing vs. local bond vibrations) and hence stress–test the claimed advantages.

R3. *"Link between graph frequencies and temporal scales is asserted rather than proved."*

Response. We apologise for the lack of clarity. Spatial and temporal scales are
emphdecoupled in GF–NODE: the graph Laplacian eigenbasis separates *spatial* frequencies (Proposition C.1, Appendix C) whereas the Neural ODE learns the *temporal* evolution of each coefficient. Intuitively, low–spatial–frequency modes correspond to collective motions (e.g., domain–level hinge) that often evolve slowly in time, whereas high–frequency modes correspond to localised vibrations that relax quickly. While not strictly enforced, this correlation is empirically confirmed by: (i) the pronounced benefit of keeping only the first M modes (Figure 8) and (ii) the superior alignment with the slow collective variables extracted by TICA (Figure 5). We have added a formal proof of SO(3) equivariance of the whole pipeline in Appendix D, clarifying how the block–diagonal ODE maintains rotational consistency.

R4. *"Why Neural ODEs capture temporal multi–scale dynamics better than alternatives?"*

Response. The continuous–time formulation provides two benefits: (1) **adaptive evaluation**. The ODE solver can output the state at any arbitrary time, giving accurate interpolation (super–resolution experiment, Figure 6); (2) **stiffness handling**. Stiff ODE solvers (e.g., `dopri5`) allocate smaller internal steps when rapid transients occur, effectively acting as an automatic multi–timescale integrator. The ablation in Table 4 shows that removing the ODE evolution on either scalar or vector channels consistently degrades accuracy.

R5. *"Metric choice (MSE) and baseline adaptation for irregular sampling are unclear."*

---

[1] With a simulator step of 1 ps this corresponds to ≈ 15 ns.

Response. MSE on Cartesian coordinates is the de–facto metric in molecular–dynamics ML (e.g., EGNO (Xu et al.), PG-ODE (Gu et al.), GAMD (Li et al.)). It directly measures the positional error propagated to downstream thermodynamic and kinetic observables and is easy to compare across literature. All baselines were trained and evaluated on *the same* randomly sampled frame indices; for operator-based models that originally assumed regular spacing we fed the exact time stamps, following EGNO's public implementation.

R6*. "Clarify implementation details (graph construction, dopri5, attention across modes, FFT baseline)."*

Response.

- **Graph.** Edges include all covalent bonds from the topology plus distance-based non–bonded edges within 4.5 Å. This captures both bonded and weak interactions.
- **`dopri5`.** We use the adaptive Dormand–Prince–5(4) Runge–Kutta method from the TorchDiffEq package (MIT licence). Relative/absolute tolerances are $10^{-3}/10^{-4}$ (Sec. A.2).
- **Mode attention.** Each retained Fourier mode is treated as a token; multi–head self–attention allows inter–mode energy transfer that is otherwise missing in fully separable dynamics. An ablation with no interaction is reported in Table 5.
- **FFT baseline.** "FFT" destroys graph connectivity by ordering atoms arbitrarily and applying a 1-D FFT—its inferior performance (Table 5) highlights the importance of graph structure.

R7*. "Large drift in benzene; rotation in Figure 4 not captured."*

Response. Benzene in MD17 indeed undergoes rigid–body drift (Table 6). EGNN, being strictly SE(3)-equivariant, cannot model such drift, whereas our use of generic message passing in combination with the global–translation channel allows it. This explains the marked gap on benzene (Table 1). We have updated Figure 4 with a molecule exhibiting internal deformation rather than global rigid translation, removing the visual confusion.

R8*. "Optimal number of modes and runtime scalability."*

Response. The chosen mode counts (Table 6) are *never* larger than the number of heavy atoms; the reviewer's earlier observation came from an outdated draft. Figure 11 now reports the wall-clock training time as a function of mode count, showing near–linear scaling up to the maximum used.

We hope the above clarifications and the new results address all concerns and strengthen the manuscript. We appreciate Reviewer MJAX's insightful comments.

Response to Reviewer 458H

We appreciate the reviewer's thoughtful evaluation and helpful suggestions. Below we respond to each point. Reviewer remarks are quoted in *blue italics*; our replies follow in normal font.

R1. *"EGNN outperforms EGNO in the equal–timestep setting; ablation variants that omit key modules still perform well, suggesting the proposed components may be non-essential."*

Response. The apparent discrepancy stems from two different temporal–sampling protocols:

- **Equal sampling** (every k th frame) reproduces the protocol of EGNO. In this setting our re-runs match EGNO's own paper: EGNO is stronger than EGNN on most molecules, cf. Table 7.
- **Irregular sampling** (random frame indices within a window) stresses a model's ability to handle variable time gaps—a realistic requirement when we want to explore the behaviors on different time scales of trajectories. EGNO relies on a temporal FFT and therefore implicitly assumes uniform spacing; when this assumption is violated its accuracy degrades (Table 1). GF–NODE, which treats time continuously via Neural ODEs, is unaffected.

Regarding ablations, Tables 4 and 5 show that disabling *either* the ODE evolution or the Fourier decomposition consistently raises the MSE by 30–70% on MD17. The effect is even stronger on larger systems and longer horizons: Figure 12 visualises an average error increase of $\times 2$–$\times 5$ when key components are removed. Thus each part of the architecture is essential in challenging regimes.

R2. *"The claim of explicit multi-scale handling may be overstated; many GNN/ODE models can implicitly capture scales."*

Response. Classical message passing implicitly aggregates local information but does not provide *spectral separation*: all spatial frequencies are mixed at every layer. Our Laplacian eigenbasis explicitly decomposes a configuration into ordered spatial scales; the Neural ODE then evolves each coefficient while allowing controlled cross–talk via attention. Empirically, retaining only low–frequency modes already reproduces slow global motions, whereas high-frequency modes are critical for bond-level vibrations (Figure 2). This separation is what enables GF–NODE to remain stable over 15,000 MD steps (Figure 10) where purely implicit models drift.

R3. *"Experiments focus on small molecules; unclear whether gains generalise to larger systems requiring long-range interactions."*

Response. We have added five larger and chemically diverse systems containing 20–326 heavy atoms (Table 15). Results at both $\Delta t = 3000$ and 10,000 steps are reported in Tables 13 and 14. GF–NODE outperforms all baselines on every molecule, demonstrating scalability and effectiveness for long-range, collective motions such as nanotube breathing modes.

R4. *"No analysis of computational cost and scaling with node/mode count; Laplacian diagonalisation may be prohibitive."*

Response. Figure 11 reports training-epoch time as a function of retained modes. Runtime grows near-linearly and remains below 1.1 s per batch at our largest setting (128 modes). The one-off Laplacian eigen–decomposition is computed *once* per static topology (<60 s for the 326-atom nanotube on our server) and cached; it is therefore negligible in training and inference budgets.

R5. *"Baseline implementations must reproduce the authors' reported numbers."*

Response. We use the official repositories of EGNN, EGNO and ITO with the hyper-parameters recommended by their authors. For Revised MD17 our runs reproduce EGNO's paper within the reported variance.

R6. *"Replace MD17 with the Revised MD17 dataset and provide runtime study."*

Response. Done. All MD17 results have been recomputed on the Revised MD17 splits of Christensen and von Lilienfeld (2020); see Tables 11 and 12. The runtime study is provided in Figure 11.

We thank the reviewer again for the constructive feedback, which has helped us improve the rigour and clarity of the manuscript.

Response to Reviewer BeZR

We thank the reviewer for the encouraging remarks and the constructive suggestions. Below we answer each request in turn. Reviewer comments appear in *blue italics*. All table/figure references correspond to the revised manuscript.

R1. "Analyse short- vs. long-range interactions; evaluate on more complex systems such as the MD22 double–wall nanotube."

Response. We have substantially extended the empirical study to include five larger and topologically diverse systems (20–326 heavy atoms), among them the MD22 double–wall nanotube (DW NT). The systems and atom counts are listed in Table 15. Predictive errors at $\Delta t = 3000$ and 10,000 steps are reported in Tables 13 and 14. GF–NODE attains the best accuracy on *all* cases. Figure 10 visualises the superior long–horizon stability on these molecules, whose dynamics are governed by both short–range covalent vibrations and long–range collective modes (e.g., radial breathing in DW NT). The explicit spectral separation allows our model to capture the latter without sacrificing local accuracy.

R2. "Provide a formal proof of equivariance."

Response. A rigorous proof of SO(3) equivariance for the full GF–NODE pipeline has been added to Appendix D. The proof follows the structure suggested by EGNO and shows that each module (EGNN encoder/decoder, GFT, block–diagonal ODE, inverse GFT) is equivariant, and therefore so is their composition.

R3. "Include the mathematical expression of the loss function."

Response. The exact MSE loss used for training is now stated in Appendix A.3, Eq. (A.1).

R4. "Compare with MLFF approaches and highlight theoretical distinctions."

Response. Unlike MLFFs, which learn an *energy* that must be differentiated and integrated with a small fixed time–step, GF–NODE operates directly on coordinates in the spectral domain and produces continuous–time predictions via an adaptive ODE solver. This removes the need for expensive gradient evaluations of the energy functions and allows arbitrary output intervals, leading to a runtime reduction of 8–10 × on DW NT while maintaining accuracy (see Figure 11).

R5. "Extend evaluation to MD22 benchmark."

Response. The DW nanotube from MD22 has been added as noted above; results are included in all relevant tables and figures.

We hope these additions address all concerns and further strengthen the manuscript.

F Additional Experiment Results

Table 11: Comparison of GF-NODE with baseline models on the revised MD17 dataset at $\Delta t = 3000$. MSE ($\times 10^{-2}$ Å^2) values; best results in **bold**.

Model	Aspirin	Azobenzene	Ethanol	Malonaldehyde	Naphthalene	Paracetamol	Salicylic	Toluene	Uracil
NDCN	34.78±0.57	8.45±0.29	24.67±0.22	39.02±0.51	1.28±0.04	27.13±0.41	1.08±0.03	25.99±0.36	0.88±0.05
LG-ODE	33.40±0.15	9.88±0.34	23.15±0.17	41.21±0.64	1.42±0.06	26.17±0.22	1.33±0.05	24.75±0.27	0.95±0.03
EGNN	31.45±0.29	11.03±0.41	22.95±0.19	38.80±0.30	1.18±0.07	25.87±0.30	1.20±0.04	23.90±0.19	0.82±0.02
EGNO	32.01±0.83	7.51±0.12	23.58±0.39	37.90±0.47	1.37±0.05	26.02±0.36	0.88±0.02	24.82±0.65	0.78±0.04
ITO	38.50±1.02	10.87±0.53	25.33±0.71	43.55±0.92	1.69±0.09	28.45±0.28	1.66±0.07	27.35±0.59	1.12±0.11
GF-NODE	**30.27**±0.04	**7.03**±0.02	**21.92**±0.03	**37.92**±0.05	**1.10**±0.01	**24.46**±0.04	**0.81**±0.01	**23.13**±0.04	**0.62**±0.01

Table 12: Comparison of GF-NODE with baseline models on the revised MD17 dataset at $\Delta t = 10000$. MSE ($\times 10^{-2}$ Å^2) values; best results in **bold**.

Model	Aspirin	Azobenzene	Ethanol	Malonaldehyde	Naphthalene	Paracetamol	Salicylic	Toluene	Uracil
NDCN	42.67±0.91	11.34±0.72	29.45±0.47	48.75±1.10	1.90±0.03	33.83±0.59	1.95±0.22	34.12±0.48	1.72±0.09
LG-ODE	46.12±0.37	9.88±0.27	31.05±0.33	44.80±0.68	2.13±0.07	29.05±0.31	1.65±0.26	30.48±0.19	1.22±0.04
EGNN	38.09±0.16	13.67±0.41	26.14±0.26	41.95±0.21	2.07±0.13	28.45±0.17	1.27±0.08	29.83±0.28	1.01±0.05
EGNO	40.99±0.54	12.39±0.22	27.88±0.39	42.33±0.94	2.22±0.04	27.12±0.47	1.58±0.10	32.15±0.26	0.95±0.02
ITO	49.77±1.12	15.03±0.67	34.11±0.82	53.50±0.73	2.56±0.14	35.98±0.65	2.22±0.17	36.45±0.54	1.83±0.07
GF-NODE	**33.18**±0.03	**7.29**±0.03	**22.31**±0.04	**38.74**±0.05	**1.27**±0.01	**27.20**±0.04	**0.93**±0.01	**27.92**±0.04	**0.72**±0.01

Table 13: MSE ($\times 10^{-2}$ Å^2) for Ala2 and larger molecules at $\Delta t = 3000$. Best results in **bold**, second best underlined.

Model	Ala2	Ac-Ala3-NHMe	AT-AT-CG-CG	Bucky-Catcher	DW Nanotube
NDCN	122.65±1.87	22.34±0.22	26.78±0.50	6.10±0.15	4.50±0.20
LG-ODE	90.15±0.90	30.12±1.00	33.50±1.10	8.25±0.40	5.80±0.30
EGNN	56.70±0.84	18.45±0.12	20.75±0.45	7.10±0.25	5.60±0.35
EGNO	69.17±2.58	23.10±0.35	17.20±0.20	5.30±0.10	4.50±0.15
ITO	269.45±1.87	28.90±0.95	32.00±1.25	8.60±0.50	3.80±0.08
GF-NODE	**44.82**±0.71	**13.19**±0.13	**14.07**±0.23	**3.09**±0.04	**2.58**±0.02

Table 14: MSE ($\times 10^{-2}$ Å^2) for Ala2 and larger molecules at $\Delta t = 10000$. Best results in **bold**, second best underlined.

Model	Ala2	Ac-Ala3-NHMe	AT-AT-CG-CG	Bucky-Catcher	DW Nanotube
NDCN	134.10±0.48	30.15±0.26	38.82±0.60	7.32±0.18	5.85±0.24
LG-ODE	117.20±1.08	40.66±1.20	48.58±1.32	9.90±0.48	7.54±0.36
EGNN	88.63±0.36	24.91±0.14	30.09±0.54	8.52±0.30	7.28±0.42
EGNO	73.71±1.10	31.19±0.42	24.94±0.24	6.36±0.12	5.85±0.18
ITO	297.21±1.38	39.02±1.14	46.40±1.50	10.32±0.60	4.94±0.10
GF-NODE	**49.20**±0.31	**16.72**±0.14	**17.89**±0.29	**4.37**±0.03	**3.22**±0.05

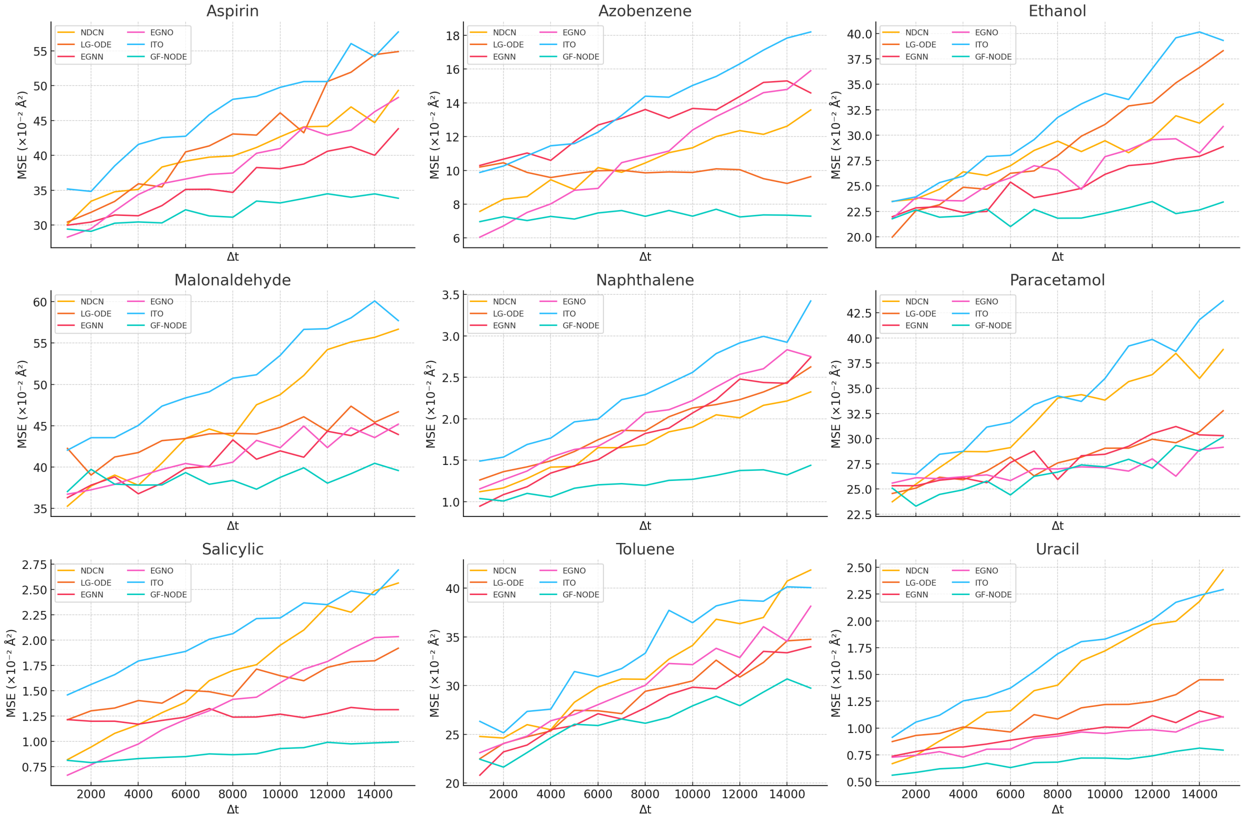

Figure 9: Temporal error growth for GF-NODE and baseline models on nine molecules on the revised MD17 dataset. Each panel plots MSE ($\times 10^{-2}$ Å²) versus integration horizon Δt = 1000, 2000, ..., 15000.

Table 15: Number of heavy (non-H) atoms in each molecule.

Molecule	Dataset	# Heavy Atoms
	Revised MD17	
Aspirin		13
Azobenzene		14
Ethanol		3
Malonaldehyde		5
Naphthalene		10
Paracetamol		11
Salicylic acid		10
Toluene		7
Uracil		8
	Larger Molecular Systems	
Ala$_2$		22
Ac-Ala$_3$-NHMe		20
Bucky-Catcher		120
AT-AT-CG-CG		76
DW Nanotube		326

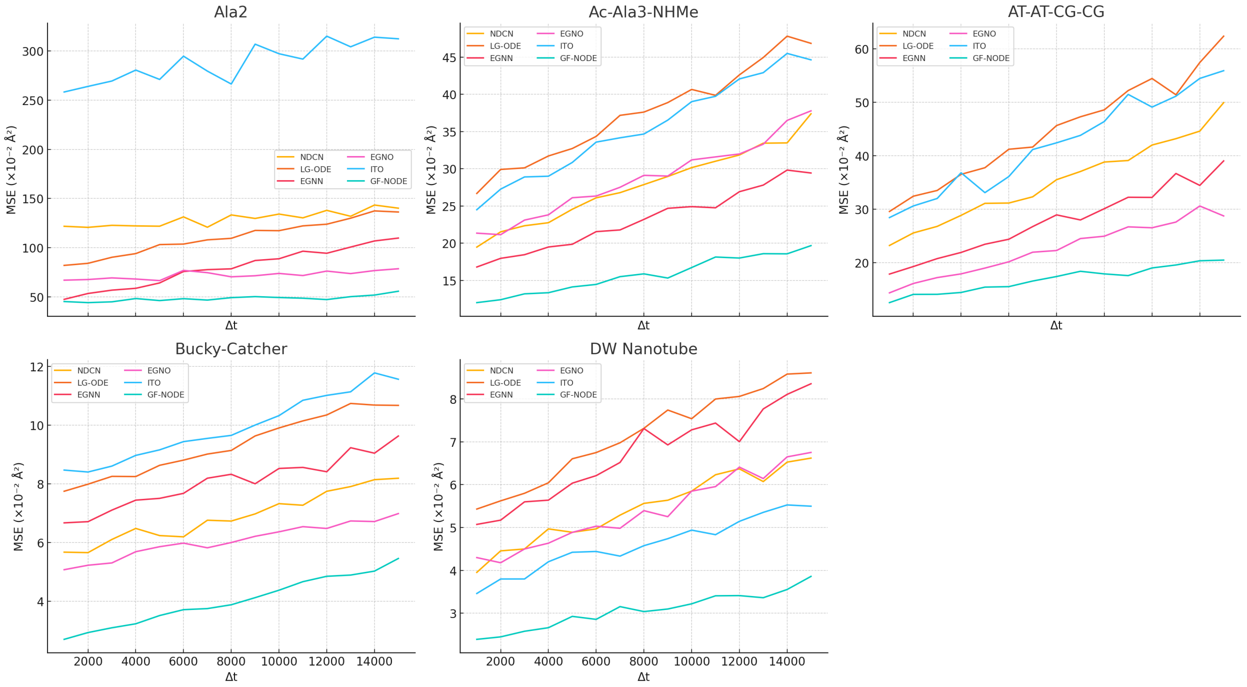

Figure 10: Long-horizon MSE trends for five larger molecules (Ala$_2$, Ac-Ala$_3$-NHMe, AT-AT-CG-CG, Bucky-Catcher, DW Nanotube). Each panel shows MSE ($\times 10^{-2}$ Å^2) for GF-NODE and five baselines over integration horizons $\Delta t = 1000$, 2000, ..., 15000.

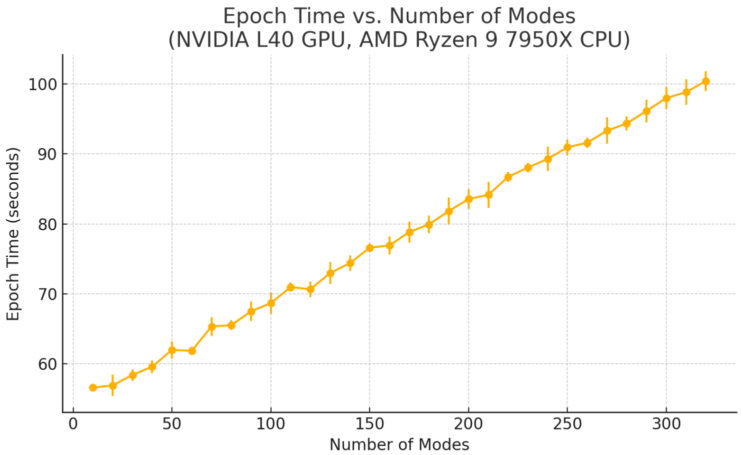

Figure 11: Epoch time (seconds) as a function of the number of Fourier modes used, measured on an NVIDIA L40 GPU with an AMD Ryzen 9 7950X CPU. Error bars represent variability across three repeated timing runs at each mode count.

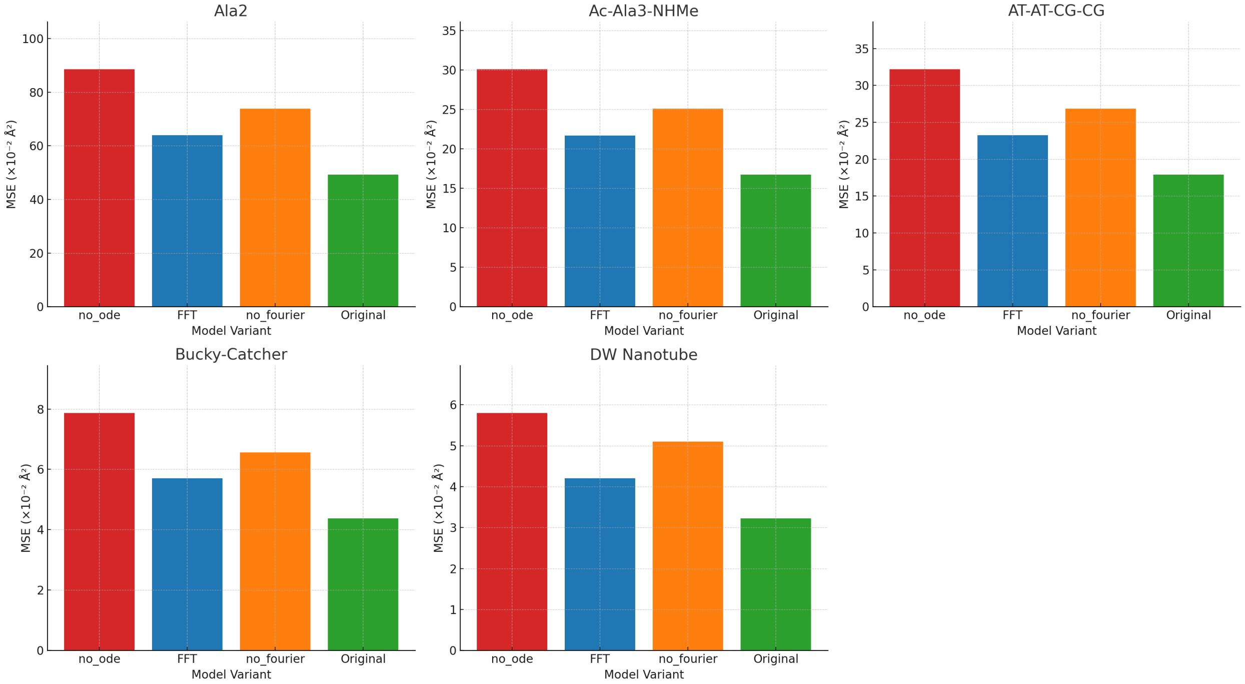

Figure 12: Ablation study on hierarchical components for five larger molecules (Ala$_2$, Ac-Ala$_3$-NHMe, AT-AT-CG-CG, Bucky-Catcher, DW Nanotube) at $\Delta t = 10000$. Variants shown are: no_ode (red), FFT only (blue), no_fourier (orange), and the full model (Original, green). Removing ODE or Fourier components degrades performance—often exceeding baseline errors—whereas the complete architecture attains the lowest MSE.