LiMemFlow: Memory-Enhanced Flow Matching for Lithium-Ion Transport in Solid-State Electrolytes

Anonymous authors
Paper under double-blind review

Abstract

Understanding and predicting long-horizon lithium-ion transport in solid-state electrolytes is important for screening high-conductivity materials, particularly near room temperature (300 K), where practical battery operation is most relevant. Although machine-learning interatomic potentials reduce force-evaluation costs, they still require sequential small-step integration, motivating large-step generative propagators for long-horizon trajectory prediction. Existing large-step propagators based on flow matching and diffusion models typically formulate trajectory generation as a position-only Markov transition, thereby discarding velocity-induced temporal dependence and collective slow processes that strongly affect diffusion statistics at 300 K. We introduce LiMemFlow, a memory-enhanced flow matching model for lithium-ion trajectory prediction. LiMemFlow combines a PET-based geometric backbone with two complementary memory modules: a point-wise history memory that encodes recent atomic displacement sequences and a pair-wise history memory that stores historical relative motion between neighboring atoms. These memory signals are injected into the equivariant propagator through residual cross-attention and updated during autoregressive rollout. We evaluate LiMemFlow on two trajectory datasets of lithium-containing crystalline structures: a newly computed 300 K dataset and 600–1200 K lithium-transport benchmarks. At 300 K, LiMemFlow reduces the mean absolute error (MAE) of $\log[\mathrm{MSD}_{\mathrm{Li}}]$ by 20.0% and the MAE of $\log[\mathrm{MSD}_{\text{Li-curve}}]$ by 17.0% compared with the strongest LiFlow baseline. Consistent with the expected role of memory in slow, path-dependent transport, explicit memory conditioning yields its clearest gains at 300–600 K, including the practically relevant near-room-temperature regime, while the gains diminish at higher temperatures and LiMemFlow remains competitive with the strongest LiFlow baseline at 1000–1200 K. The code and data are available at `https://anonymous.4open.science/r/limemflow-review-sub/`.

1 Introduction

As demand grows for safer and higher-performance energy storage, the rational design of lithium-containing battery materials increasingly depends on accurate prediction of ion diffusion and transport in crystalline solids. In solid-state lithium batteries, a candidate electrolyte must provide high ionic diffusivity and conductivity while maintaining structural stability at temperatures close to realistic operating conditions (Janek & Zeier, 2016; Famprikis et al., 2019; Jun et al., 2024a). Ab initio molecular dynamics (AIMD) is a reliable computational approach for studying ionic migration because it evaluates interatomic forces from electronic-structure calculations and directly simulates atomic transport (Car & Parrinello, 1985). However, AIMD requires repeated expensive density functional theory calculations at femtosecond time steps. This cost makes it difficult to obtain long-timescale transport statistics or to screen large materials databases. Graph neural network-based machine learning interatomic potentials (MLIPs), such as MACE (Batatia et al., 2022) and MatterSim (Yang et al., 2024), reduce this cost by replacing first-principles force evaluations. Despite this progress, MLIPs still rely on small-step numerical integration, so the overall simulation pipeline remains limited by sequential atomic propagation.

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

Recent generative models have made substantial progress in large-step molecular dynamics prediction. Approaches based on normalizing flows, flow matching, and diffusion models can predict atomic displacements over intervals much larger than conventional femtosecond-scale integration steps, which makes them promising tools for further accelerating molecular dynamics (Klein et al., 2023; Hsu et al., 2024; Nam et al., 2025). These methods typically observe only atomic positions and make a Markov approximation for trajectory prediction. Specifically, they learn an observed-state transition kernel, such as $p_\theta(X_{\tau+\Delta\tau} \mid X_\tau, T)$, which predicts the next atomic positions from the positions in the previous frame. In the full phase-space representation, molecular dynamics is Markovian when both atomic positions and momenta are explicitly retained. When a model observes only atomic positions at coarse time intervals, velocities are discarded, while collective slow processes, including phonon excitations, local cage relaxation, and environmental fluctuations, are implicitly integrated out (Grogan et al., 2020; Vroylandt et al., 2022; Lyu & Lei, 2023). The Mori–Zwanzig projection-operator formalism and generalized Langevin equation models show that such reduced dynamics generally exhibits memory effects and colored noise (Mori, 1965; Zwanzig, 1961). This limitation becomes more severe for room-temperature lithium-ion migration. At $300\,\mathrm{K}$ (room temperature), longer waiting times between ion hopping events allow local cage confinement, configurational backtracking, defect-restricted migration pathways, and framework response to influence subsequent motion across multiple coarse steps (Lee et al., 2024; Hanghofer et al., 2019; Morgan, 2021). Consequently, a one-step Markovian propagator conditioned only on the instantaneous atomic configuration may fit high-temperature or short-time positional evolution, yet fail to preserve the temporal correlations that determine room-temperature, long-horizon ion transport (Jun et al., 2024a; Sanchez-Gonzalez et al., 2020; Poletayev et al., 2024; 2022; López et al., 2024). This limitation is practically important because lithium batteries are generally expected to operate near room temperature. Therefore, accurate trajectory prediction at room temperature is more important for solid-electrolyte screening and battery materials design (Bachman et al., 2016; Manthiram et al., 2017).

Motivated by the effective non-Markovian nature of large-step, position-only configurational dynamics, we propose LiMemFlow, a history-conditioned flow matching model for lithium-ion trajectory prediction. As illustrated in Figure 1, LiMemFlow combines an equivariant PET backbone (Pozdnyakov & Ceriotti, 2023) with explicit memory conditioning. The equivariant backbone preserves geometric structure in velocity-field prediction for flow updates, while the memory module uses recent displacement history to represent dynamical context that cannot be inferred from the current atomic configuration alone. Point-wise memory captures atom-specific persistence and backtracking, while pair-wise memory encodes historical relative motion within local neighborhoods to represent cage relaxation, framework response, and correlated hopping. Together, they provide history-dependent context unavailable from position-only inputs.

Our contributions are summarized as follows: First, we formulate large-step, position-only lithium-ion trajectory prediction, especially at $300\,\mathrm{K}$, as a partially observed non-Markovian transport modeling problem rather than as standard one-step Markovian displacement prediction. Second, we introduce LiMemFlow, a history-conditioned flow matching model that incorporates recent displacement history into an equivariant backbone to encode latent dynamical state and improve the modeling of path-dependent lithium-ion migration. Third, we construct a $300\,\mathrm{K}$, $150\,\mathrm{ps}$ trajectory dataset covering 974 lithium-containing crystalline structures, and systematically benchmark LiMemFlow on both the $300\,\mathrm{K}$ room-temperature trajectory dataset and high-temperature datasets from 600 to $1200\,\mathrm{K}$. On the 300 K dataset, LiMemFlow reduces the $\log[\mathrm{MSD}_{\mathrm{Li}}]$ MAE by 20.0% and the $\log[\mathrm{MSD}_{\mathrm{Li\text{-}curve}}]$ MAE by 17.0% compared with the strongest LiFlow baseline. Across temperatures, the gains are largest at 300–600 K, diminish at 800 K, and become statistically indistinguishable from the strongest LiFlow baseline at 1000–1200 K. This trend is consistent with memory being most useful for slow, path-dependent transport(Mori, 1965; Zwanzig, 1961; Poletayev et al., 2024), particularly near room temperature, the regime most relevant to practical electrolyte screening (Lee et al., 2024; Bachman et al., 2016; Manthiram et al., 2017).

2 Related Work

Surrogate models for accelerating molecular dynamics. Recent work explores surrogate models as a way to reduce the reliance of molecular dynamics on femtosecond-scale numerical integration. These methods typically learn a finite-time transition operator $p_\theta(X_{\tau+\Delta\tau} \mid X_\tau, T)$, so that a

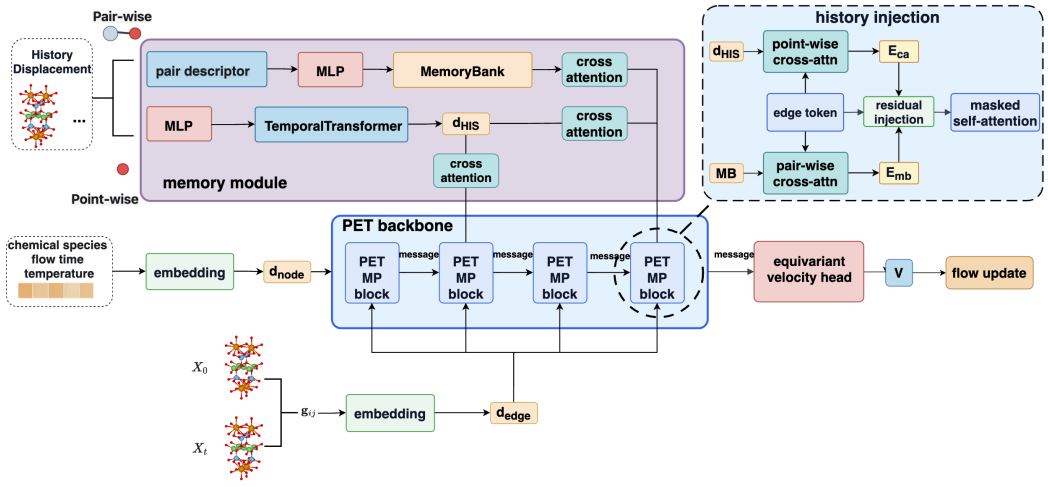


Figure 1: **Overview of LiMemFlow.** The architecture includes the PET backbone, memory module, and history injection.

single model step can predict molecular dynamics evolution over picosecond or longer timescales. ITO implements the implicit transfer operator as an SE(3)-equivariant denoising diffusion model, jointly learning transition kernels at multiple time resolutions and generating self-consistent dynamics across timescales under partially observed or coarse-grained representations (Schreiner et al., 2023). LiFlow targets lithium-ion transport in crystalline materials and learns a 1 ps atomic-displacement transition kernel conditioned on a fixed lattice and temperature (Nam et al., 2025). Similar ideas are also extended to protein dynamics: F3low performs frame-to-frame flow matching on the SE(3) manifold for coarse-grained protein backbones, autoregressively sampling subsequent conformations conditioned on the previous frame; EquiJump uses SO(3)-equivariant stochastic interpolants to bridge adjacent all-atom protein conformations directly, avoiding generation from a prior distribution far from the data manifold (Li et al., 2024; dos Santos Costa et al., 2025).

A second direction uses diffusion or generative objectives to learn effective force fields for coarse-grained dynamics. Two for One and Consistent Sampling and Simulation train diffusion models on coarse-grained conformational distributions and interpret the learned score as an approximate force field for coarse-grained MD (Arts et al., 2023; Plainer et al., 2025). In a related direction, Durumeric et al. combine force matching with denoising objectives, using atomistic force labels together with noise-force supervision to build stable machine-learned coarse-grained force fields with less atomistic training data (Durumeric et al., 2026).

A third group of force-free propagation models predicts future states directly in phase space or spacetime coordinates. TrajCast updates atomic positions and velocities autoregressively, bypassing explicit force evaluation and small-step integration (Thiemann et al., 2025). MDTrajNet predicts molecular coordinates from initial positions, initial velocities, and a target time, thereby learning a map from the initial phase-space state to the molecular structure at a future time (Ge et al., 2026). These methods broaden the design space of learned MD surrogates. Yet they usually condition on the current configuration, the initial phase-space state, or an equilibrium conformational distribution, and they mainly target enhanced sampling, coarse-grained force-field construction, protein dynamics, or small-molecule dynamics. They therefore do not directly address room-temperature lithium migration, where long rollouts must preserve directionally correlated hopping, backtracking, and framework response.

History-conditioned trajectory modeling. A natural way to recover non-Markovian behavior is to use history as model context. LSTM-based molecular dynamics models show that hidden states and memory cells can accumulate historical information, learn path probabilities, and reproduce thermodynamic and kinetic statistics (Tsai et al., 2020; 2022). ESTAG recasts physical simulation as history-conditioned spatio-temporal prediction, using past trajectories to recover non-Markovian interactions caused by unobserved environmental variables (Wu et al., 2023). The same idea also

appears in protein molecular dynamics generation. ConfRover uses a causal sequence model in latent space to aggregate historical frame information and generates continuous three-dimensional structures with an SE(3) diffusion decoder (Shen et al., 2025; Wang et al., 2024). STAR-MD further embeds joint spatio-temporal attention into a causal SE(3) diffusion transformer, using historical context during diffusion denoising to generate long-horizon trajectories (Shoghi et al., 2026). These works show that temporal context, sequence modeling, and geometric equivariance are important for stable trajectory generation.

3 METHOD

LiMemFlow models large-step, position-only propagation as an effective non-Markovian process by augmenting a dual-geometry equivariant PET backbone with point-wise and pair-wise displacement-history memories. Here, τ/t denote physical/flow time, $\Delta\tau$ is the propagation interval, and the remaining notation is summarized in Appendix F.

3.1 NON-MARKOVIANITY IN LARGE-STEP POSITION-ONLY PROPAGATION

We study lithium-ion trajectory generation under large-step, position-only observation. The full molecular dynamics process is not inherently non-Markovian. Let the complete state be

$$\Gamma_\tau = (Q_\tau, P_\tau, \Xi_\tau), \tag{1}$$

where Q_τ denotes all atomic positions, P_τ denotes momenta, and Ξ_τ denotes phonons, local environmental responses, and other unresolved degrees of freedom. Given this complete phase-space state, the dynamics approximately follow a Markovian evolution:

$$p(\Gamma_{\tau+\Delta\tau} \mid \Gamma_\tau, \Gamma_{<\tau}, T) = p(\Gamma_{\tau+\Delta\tau} \mid \Gamma_\tau, T). \tag{2}$$

A learned large-step propagator, however, usually does not have access to the complete state. Under position-only observation, velocities, momenta, local lattice relaxation, lithium backtracking paths, and heat-bath degrees of freedom are marginalized out. The future configuration therefore cannot, in general, be determined from X_τ (conditioning configuration) alone:

$$p(X_{\tau+\Delta\tau} \mid X_\tau, X_{0:\tau}, T) \neq p(X_{\tau+\Delta\tau} \mid X_\tau, T). \tag{3}$$

Thus, this work does not model Markovian MD in the full phase space. It models an effective non-Markovian process induced by projecting the dynamics onto position variables.

This reduction is formalized by the Mori–Zwanzig projection-operator framework (Mori, 1965; Zwanzig, 1961). Let $A(\tau) = \Pi(\Gamma_\tau)$ be an observable variable obtained by projecting the complete phase-space state, such as the atomic configuration used in this work. The Mori–Zwanzig formalism shows that, after projection, the evolution of a high-dimensional Markovian system can generally be written in the form of a generalized Langevin equation:

$$\frac{dA(\tau)}{d\tau} = F(A(\tau)) + \int_0^\tau K(\tau - s)A(s)\, ds + \eta(\tau). \tag{4}$$

Here, $F(A(\tau))$ is the instantaneous term determined by the current observed state, $K(\tau - s)$ is a memory kernel that describes the persistent influence of past states, and $\eta(\tau)$ is a random force induced by eliminated degrees of freedom. Eq. 4 therefore provides the theoretical motivation for introducing history conditioning into position-only propagation.

At 300 K, longer waiting times strengthen path dependence through cage effects, backtracking, defect-constrained pathways, and framework response (Jun et al., 2024a; Poletayev et al., 2024; 2022; López et al., 2024). A current-state propagator may therefore fit one-step displacements while missing correlations that govern long-horizon MSD and structural stability.

3.2 LIMEMFLOW ARCHITECTURE

History-conditioned flow parameterization. Following the non-Markovian formulation in Section 3.1, LiMemFlow models large-step lithium-ion propagation as

$$p_\theta(X_{\tau+\Delta\tau} \mid X_\tau, H_\tau, T), \tag{5}$$

where X_τ is the conditioning configuration at physical time τ, $X_{\tau+\Delta\tau}$ is the future configuration after a physical interval $\Delta\tau$, and H_τ denotes a finite history before X_τ. For atom i, we represent this history as a sequence of consecutive displacement increments:

$$H_{\tau,i} = \{u_i^{(m)}\}_{m=1}^{M_i}, \qquad u_i^{(m)} = x_i(\tau_{m+1}) - x_i(\tau_m). \tag{6}$$

The history frames are sampled at a fixed interval, and the last valid historical displacement ends at X_τ. Compared with absolute coordinates, displacement history more directly represents coarse-step local motion, including migration direction, correlated backtracking, intra- and inter-cage hopping, and framework-coupled relaxation (Vogel, 2003; He et al., 2017; Banerjee & Tkatchenko, 2025; Jun et al., 2024b; Ding et al., 2025).

LiMemFlow builds on flow matching to learn a velocity field from an intermediate flow state X_t toward the target configuration $X_{\tau+\Delta\tau}$ (Lipman et al., 2023). It extends this velocity field to the history-conditioned form

$$\widehat{V} = P_\theta(X_t, X_\tau, H_\tau, T, t) \in \mathbb{R}^{N\times 3}. \tag{7}$$

Here, $t \in [0,1]$ denotes flow time rather than physical MD time τ. The history H_τ can be viewed as a finite-dimensional learnable approximation to the memory integral in Eq. 4. Given an intermediate state X_t sampled from the chosen probability path and the corresponding target velocity $V_t^\star$, LiMemFlow optimizes the history-conditioned velocity matching objective

$$\mathcal{L}_{\mathrm{FM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1),\, X_t\sim p_t(\cdot|X_\tau, X_{\tau+\Delta\tau})}\left[\left\|\widehat{V} - V_t^\star\right\|_F^2\right], \tag{8}$$

where $\|\cdot\|_F$ denotes the Frobenius norm over atoms and Cartesian coordinates. LiMemFlow does not explicitly estimate an analytic physical memory kernel $K(\cdot)$. Instead, it learns from finite historical displacements which past motions remain informative for the current transition.

Dual-geometry equivariant backbone. LiMemFlow uses an equivariant PET backbone to predict the velocity field (Pozdnyakov & Ceriotti, 2023). For the same neighbor edge (i, j), the model encodes both the relative geometry of this edge in the intermediate flow state X_t and its relative geometry in the conditioning configuration X_τ. In other words, the backbone observes the local structure at the current flow state together with its reference state at the propagation origin. This dual-geometry representation allows the model to compare local offsets between X_t and X_τ while keeping a fixed neighborhood structure. PET blocks update node and edge representations through local masked attention, and the final velocity is read out from equivariant functions of edge vectors. This design provides a geometry-aware flow matching propagator while preserving translation invariance and rotation equivariance. Additional implementation details of the dual-geometry graph construction, PET message-passing blocks, and equivariant velocity readout are provided in Appendix E.

Point-wise history memory. The first memory pathway constructs an independent temporal memory for each atom, which describes its recent motion before the current conditioning configuration. For atom i, each historical displacement is first projected into a base history token:

$$\bar{c}_i^{(m)} = \phi_{\mathrm{hist}}\left(u_i^{(m)}, \|u_i^{(m)}\|_2, \frac{m-1}{\max(M_i-1,1)}, T\right), \qquad \bar{C}_i = [\bar{c}_i^{(1)}, \ldots, \bar{c}_i^{(M_i)}]. \tag{9}$$

Each token contains the three-dimensional displacement, displacement magnitude, relative historical position, and temperature condition, so it represents motion direction, motion strength, and thermodynamic context. A lightweight temporal Transformer then contextualizes these tokens along the history dimension:

$$C_i = \mathrm{Transformer}_{\mathrm{hist}}\left(\bar{C}_i, \mu_i^{\mathrm{hist}}\right). \tag{10}$$

Here, μ_i^{hist} is a history-validity mask that removes padded historical steps. After the temporal Transformer, each historical displacement is interpreted in relation to nearby historical segments rather than in isolation. For example, consecutive displacements in a similar direction can indicate persistent migration, whereas reversed displacements may correspond to backtracking or local cage confinement.

In selected PET layers, current edge tokens centered at atom i read C_i through cross-attention:

$$\Delta E_{i,\mathrm{ca}}^{(\ell)} = \mathrm{CrossAttn}\left(\widetilde{E}_i^{(\ell)}, C_i\right). \tag{11}$$

The current edge tokens serve as queries, while the history tokens serve as keys and values. The model can therefore select relevant historical segments according to the current local geometry. This pathway provides each edge token with recent motion memory of the center atom, allowing the model to identify persistent migration direction, renewed hopping after stagnation, backtracking, and latent dynamical state implied by historical velocity trends.

Pair-wise history memory. Point-wise history memory describes single-atom motion, but lithium-ion transport is not determined by the history of one ion alone. Whether an ion continues to migrate, backtracks, or remains confined within a local cage is influenced by local Li/vacancy occupancy, diffusion-channel connectivity, and framework-coupled relaxation (Chen et al., 2017; Banerjee & Tkatchenko, 2025; Ding et al., 2025). To represent such local relative historical motion, LiMemFlow constructs a local spatio-temporal memory bank:

$$B_i = \mathrm{Bank}\left(\{u_j^{(m)} - u_i^{(m)} : j \in \mathcal{N}(i),\, m = 1, \ldots, M_i\}\right). \tag{12}$$

Here, $u_j^{(m)} - u_i^{(m)}$ denotes the motion of neighbor j relative to center atom i at historical step m. Unlike point-wise history, this representation encodes local pair dynamics through relative displacement histories, allowing the propagator to condition its predictions on relative relaxation, cooperative-motion patterns, and direction-dependent local constraints. Each pair-time token in the memory bank corresponds to a specific neighbor and a specific historical step, so B_i preserves both neighbor identity and temporal information.

Current edge tokens retrieve pair-level historical information by reading B_i:

$$\Delta E_{i,\mathrm{mb}}^{(\ell)} = \mathrm{CrossAttn}\left(\widetilde{E}_i^{(\ell)}, B_i\right). \tag{13}$$

This attention readout allows the current edge (i, j) to attend not only to its own historical relative motion, but also to the historical responses of other neighbors around the same center atom. This readout allows the model to incorporate historical patterns of local cage relaxation, neighboring framework response, correlated lithium-ion motion, and direction-dependent migration into its displacement predictions. The pair-wise pathway complements the point-wise history pathway: the point-wise pathway emphasizes single-atom temporal trends, while the memory bank emphasizes historical interactions between the local environment and the center atom. LiMemFlow injects point-wise and pair-wise memory into selected PET layers through residual cross-attention.

4 Experiments

Datasets. We evaluate LiMemFlow on two lithium-containing molecular dynamics (MD) trajectory datasets that cover room-temperature and high-temperature transport regimes. The first dataset is a newly generated 300 K room-temperature dataset containing 974 lithium-containing crystalline structures. The initial structures are selected from the Materials Project under three stability- and transport-oriented criteria: the atomic fraction of lithium exceeds 10%, the band gap is larger than 2 eV, and the energy above the convex hull is below $0.1\,\mathrm{eV/atom}$. Because generating 150 ps MD trajectories with `mace-mpa0-medium` is computationally expensive, we further sample 974 structures from the filtered candidates according to both structural and compositional diversity. The resulting dataset spans diverse chemical compositions and crystal structures; additional dataset statistics are provided in Appendix D. For each selected structure, we construct either a $3\times3\times3$ or $4\times4\times4$ supercell and run MD simulations in the NPT ensemble using the general-purpose machine-learning interatomic potential (MLIP) `mace-mpa0-medium`. The integration time step is 1 fs, and each trajectory has a total length of 150 ps, corresponding to 150,000 MD steps, with atomic positions recorded every 10 steps. Constructing this room-temperature dataset required approximately 10,000 A100 GPU-hours.

The second dataset follows the universal lithium-transport benchmark introduced by LiFlow (Nam et al., 2025). It contains trajectories for 4,186 lithium-containing crystalline structures simulated at 600 K, 800 K, 1000 K, and 1200 K. Each trajectory has a total length of 25 ps.

Baselines. We compare LiMemFlow with representative large-step molecular trajectory generation methods. LiFlow(P) denotes the Propagator-only variant of LiFlow, which uses flow match-

ing to generate atomic displacements conditioned on the current crystal structure and temperature (Nam et al., 2025). This baseline is especially important because it is architecturally closest to LiMemFlow: both models use a flow-matching propagator for large-step atomic displacement generation, while LiFlow(P) does not use explicit history conditioning. LiFlow(P+C) denotes the full LiFlow model with both the Propagator and the Corrector. The Corrector reduces accumulated geometric errors during autoregressive rollout. We also compare with MDTrajNet (Ge et al., 2026), a direct time-conditioned trajectory prediction model. The original MDTrajNet pipeline uses a machine-learning interatomic potential for potential-energy-surface correction during inference, but we do not include this correction in our baseline rollout. LiMemFlow does not explicitly model energy or force-field information, so applying an additional potential-energy correction only to MDTrajNet would make the comparison reflect more than the trajectory generator itself. To keep the baselines comparable and avoid confounding from an extra physics model, we evaluate MDTrajNet in its direct prediction form.

Evaluation metrics. $\log[\mathrm{MSD}_{\mathrm{Li}}]$ MAE is defined as the mean absolute error in log space between the final lithium MSD of the generated trajectory and that of the reference trajectory (Nam et al., 2025). This metric measures whether the model reproduces the overall magnitude of lithium transport at the end of rollout.

$\log[\mathrm{MSD}_{\text{Li-curve}}]$ MAE measures the similarity between the lithium-ion MSD-time curve produced by the model rollout and the corresponding curve from the reference trajectory. In molecular dynamics studies of battery materials, the MSD curve is a standard trajectory-level physical statistic for ion transport: the slope in its long-time regime gives the diffusion coefficient through the Einstein relation, and the diffusion coefficient is further connected to ionic conductivity through the Nernst–Einstein relation (Maginn et al., 2018; He et al., 2018; Baktash et al., 2020). Matching the full Li MSD curve is therefore important for assessing whether a generated trajectory preserves the transport behavior of the reference simulation. Specifically, we compute the absolute error between the generated and reference log MSD values at the sampled time points and average it over all time points. This metric is stricter than comparing only the final MSD, because it requires the model to match not only the final diffusion magnitude but also the temporal evolution of lithium-ion diffusion.

The stable-trajectory metric reports the percentage of generated trajectories that complete the prescribed rollout horizon. Spearman Rank Correlation (Li) measures whether the model preserves the relative ordering of lithium mobility across different materials (Nam et al., 2025).

4.1 Temperature-dependent Long-Horizon Rollout Across Temperatures

LiMemFlow, LiFlow(P), and LiFlow(P+C) generate atomic-position frames at 1 ps intervals on both the 300 K and 600–1200 K datasets, while MDTrajNet predicts atomic-position frames at 0.2 ps intervals. On the 300 K task, LiMemFlow, LiFlow(P), and LiFlow(P+C) are rolled out for 150 steps, whereas MDTrajNet is rolled out for 750 steps, so all methods produce trajectories spanning 150 ps. On the 600–1200 K tasks, LiMemFlow, LiFlow(P), and LiFlow(P+C) are rolled out for 25 steps, whereas MDTrajNet is rolled out for 125 steps, giving the same total rollout length of 25 ps. We then evaluate the generated trajectories against the reference trajectories using the four metrics described above. Additional details on model training and inference are provided in Appendix G.

Table 1 summarizes the long-horizon rollout performance on the 300 K dataset and the 600–1200 K datasets. The improvement is the largest at 300 K, which is the most important regime for room-temperature lithium-based solid-state electrolyte screening. Compared with LiFlow(P+C), LiMemFlow reduces the $\log[\mathrm{MSD}_{\mathrm{Li}}]$ MAE from 0.544 to 0.435, a 20.0% reduction, and reduces the $\log[\mathrm{MSD}_{\text{Li-curve}}]$ MAE from 0.547 to 0.454, a 17.0% reduction.

On the 600–1200 K tasks, LiMemFlow achieves its clearest improvement at 600 K, while the numerical advantage diminishes as temperature increases. At 800 K, the point estimate remains lower than that of the best LiFlow baseline, but the improvement is reduced. At 1000 K and 1200 K, the differences from the strongest LiFlow baseline are small and do not show a clear advantage. This temperature dependence is consistent with the non-Markovian motivation in Section 3.1: explicit history is most useful for slow, path-dependent low-temperature transport and becomes less beneficial as thermal hopping becomes more frequent.

Table 1: Main rollout results on 300 K and 600–1200 K datasets. Values are reported as mean ± sample standard deviation over three independent rollout seeds, with each seed-level metric computed over the full test set. MDTrajNet inference is deterministic, so the standard deviation is zero. * Results reproduced using the checkpoint released by the original authors.

Temp.	Model	$\log[\mathrm{MSD_{Li}}]$ MAE ↓	$\log[\mathrm{MSD_{Li\text{-}curve}}]$ MAE ↓	Stable (%) ↑	Spear Rank ↑
300 K	MDTrajNet	2.183 ± 0.00000	1.588 ± 0.00000	95.74 ± 0.0	0.297 ± 0.00000
	LiFlow(P)	1.092 ± 0.01366	0.744 ± 0.00684	97.9 ± 2.1	0.345 ± 0.00125
	LiFlow(P+C)	0.544 ± 0.01033	0.547 ± 0.00673	98.9 ± 1.1	0.579 ± 0.00858
	LiMemFlow	**0.435** ± 0.00120	**0.454** ± 0.00442	**100.0** ± 0.0	**0.659** ± 0.00164
600 K	MDTrajNet	1.151 ± 0.00000	0.827 ± 0.00000	100.0 ± 0.0	0.120 ± 0.00000
	LiFlow(P)*	0.377 ± 0.00549	0.362 ± 0.00094	99.8 ± 0.2	0.704 ± 0.00628
	LiFlow(P+C)*	0.375 ± 0.00070	0.389 ± 0.00114	100.0 ± 0.0	0.755 ± 0.00194
	LiMemFlow	**0.298** ± 0.00947	**0.308** ± 0.00690	**100.0** ± 0.0	**0.783** ± 0.01406
800 K	MDTrajNet	0.971 ± 0.00000	0.769 ± 0.00000	100.0 ± 0.0	0.206 ± 0.00000
	LiFlow(P)*	0.390 ± 0.00260	0.360 ± 0.00216	99.6 ± 0.1	0.752 ± 0.01430
	LiFlow(P+C)*	0.396 ± 0.00524	0.394 ± 0.00039	100.0 ± 0.0	0.788 ± 0.01152
	LiMemFlow	**0.339** ± 0.00346	**0.338** ± 0.00094	**100.0** ± 0.0	**0.816** ± 0.00398
1000 K	MDTrajNet	0.850 ± 0.00000	0.790 ± 0.00000	100.0 ± 0.0	0.230 ± 0.00000
	LiFlow(P)*	0.450 ± 0.00813	0.403 ± 0.00450	97.4 ± 0.4	0.751 ± 0.00816
	LiFlow(P+C)*	0.460 ± 0.00470	0.419 ± 0.00377	99.7 ± 0.3	0.759 ± 0.00372
	LiMemFlow	**0.449** ± 0.00793	**0.401** ± 0.00392	**100.0** ± 0.0	**0.776** ± 0.00271
1200 K	MDTrajNet	0.883 ± 0.00000	0.917 ± 0.00000	100.0 ± 0.0	0.274 ± 0.00000
	LiFlow(P)*	0.409 ± 0.00902	0.382 ± 0.00609	98.4 ± 0.1	0.808 ± 0.00685
	LiFlow(P+C)*	0.405 ± 0.00947	0.391 ± 0.00497	99.6 ± 0.5	0.830 ± 0.00341
	LiMemFlow	**0.397** ± 0.00508	**0.381** ± 0.00206	**100.0** ± 0.0	**0.838** ± 0.00458

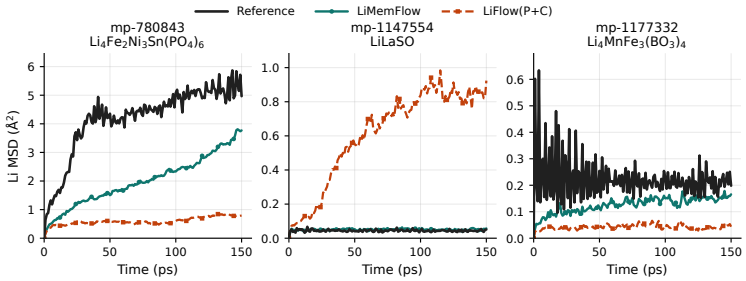


Figure 2: **Representative 300 K Li MSD-t curves for three test structures.** Black curves denote reference MD trajectories, green curves denote LiMemFlow rollouts, and red curves denote LiFlow(P+C) rollouts.

4.2 Room-Temperature MSD Curves and Screening Implications

Figure 2 shows representative 300 K Li MSD–time curves. LiMemFlow follows the reference magnitude and growth trend more closely than LiFlow(P+C); for `mp-780843`, LiFlow(P+C) predicts localized vibrations, whereas the reference trajectory shows Li migration. The long-time slope of these curves gives the tracer diffusivity D^* (Maginn et al., 2018). Ionic conductivity is determined by the collective charge diffusivity D_σ through the Einstein relation. However, because D_σ contains cross-ion correlations and is estimated from a single collective displacement, its finite-trajectory estimate has higher statistical variance and requires longer MD trajectories to converge; this problem is particularly pronounced in systems with sparse lithium hopping events (He et al., 2018; Baktash et al., 2020). We therefore use the tracer diffusivity D^* as a screening proxy. At a fixed horizon t, the endpoint metric in Table 1 equals the mean absolute log error of $D^*(t) = \mathrm{MSD_{Li}}(t)/(6t)$; reducing it from 0.544 to 0.435 lowers the geometric-mean fold error from 3.5× to 2.7×. In retrospective screening (Appendix B), LiMemFlow raises recall from 0.677 to 0.778 and average precision from 0.600 to 0.790 relative to LiFlow(P+C).

4.3 Memory Component Analysis

We ablate each memory pathway under the same rollout protocol. **Pair-wise history memory.** Removing the pair-wise memory bank raises $\log[\mathrm{MSD}_{\mathrm{Li}}]$ MAE from 0.435 to 0.545 at 300 K and worsens the curve-level MAE. It also degrades the collective-transport and return-hop diagnostics in Appendix C, showing that pair-wise history provides complementary local dynamical context. **Point-wise history memory.** Removing the point-wise pathway causes a larger degradation, raising $\log[\mathrm{MSD}_{\mathrm{Li}}]$ MAE from 0.435 to 0.951. It therefore provides the primary memory contribution to room-temperature rollout, consistent with its larger effect on collective diffusivity and return-hop statistics in Appendix C. **Effect on ranking metrics.** Memory improves MAE but has mixed effects on Spearman correlation. At 1200 K, it provides negligible MAE gains and slightly lowers rank correlation. At 300 K, LiMemFlow and w/o Memory have comparable Spearman scores (0.659 vs. 0.667), whereas LiMemFlow attains higher AP and recall in retrospective screening, indicating better recovery of high-mobility candidates. Further ablations and screening results are provided in Appendices A and B.

Table 2: Ablation study of LiMemFlow memory components on 300 K and 600–1200 K datasets.

Temp.	Model	$\log[\mathrm{MSD}_{\mathrm{Li}}]$ MAE ↓	$\log[\mathrm{MSD}_{\mathrm{Li\text{-}curve}}]$ MAE ↓	Stable (%) ↑	Spear Rank ↑
300 K	w/o Pair Wise	0.545 ± 0.00207	0.494 ± 0.00365	100.0 ± 0.0	0.575 ± 0.00734
	w/o Point Wise	0.951 ± 0.02469	0.616 ± 0.00807	100.0 ± 0.0	0.425 ± 0.02891
	w/o Memory	0.567 ± 0.01011	0.498 ± 0.00130	100.0 ± 0.0	**0.667** ± 0.00957
	LiMemFlow	**0.435** ± 0.00120	**0.454** ± 0.00442	100.0 ± 0.0	0.659 ± 0.00164
600 K	w/o Pair Wise	0.315 ± 0.00549	0.324 ± 0.00842	100.0 ± 0.0	0.780 ± 0.01177
	w/o Point Wise	0.331 ± 0.00509	0.335 ± 0.00347	100.0 ± 0.0	0.778 ± 0.01562
	w/o Memory	0.343 ± 0.00576	0.343 ± 0.00510	100.0 ± 0.0	0.771 ± 0.01074
	LiMemFlow	**0.298** ± 0.00947	**0.308** ± 0.00690	100.0 ± 0.0	**0.783** ± 0.01406
800 K	w/o Pair Wise	0.355 ± 0.00618	0.351 ± 0.00201	100.0 ± 0.0	0.814 ± 0.00341
	w/o Point Wise	0.358 ± 0.00111	0.352 ± 0.00398	100.0 ± 0.0	0.805 ± 0.00785
	w/o Memory	0.361 ± 0.00832	0.364 ± 0.00379	100.0 ± 0.0	0.812 ± 0.00374
	LiMemFlow	**0.339** ± 0.00346	**0.338** ± 0.00094	100.0 ± 0.0	**0.816** ± 0.00398
1000 K	w/o Pair Wise	0.454 ± 0.00533	0.409 ± 0.00533	100.0 ± 0.0	**0.784** ± 0.00325
	w/o Point Wise	0.458 ± 0.01474	0.408 ± 0.00813	100.0 ± 0.0	**0.784** ± 0.00372
	w/o Memory	0.463 ± 0.00766	0.414 ± 0.00978	100.0 ± 0.0	0.778 ± 0.00458
	LiMemFlow	**0.449** ± 0.00793	**0.401** ± 0.00392	100.0 ± 0.0	0.776 ± 0.00271
1200 K	w/o Pair Wise	0.400 ± 0.00404	0.385 ± 0.00244	100.0 ± 0.0	0.836 ± 0.00280
	w/o Point Wise	0.401 ± 0.00456	0.383 ± 0.00450	100.0 ± 0.0	0.838 ± 0.00229
	w/o Memory	0.398 ± 0.01491	0.385 ± 0.00735	100.0 ± 0.0	**0.859** ± 0.00465
	LiMemFlow	**0.397** ± 0.00508	**0.381** ± 0.00206	100.0 ± 0.0	0.838 ± 0.00458

5 Conclusions and Limitations

We introduced LiMemFlow, a flow-matching propagator whose point-wise and pair-wise displacement memories reduce the 300 K $\log[\mathrm{MSD}_{\mathrm{Li}}]$ and $\log[\mathrm{MSD}_{\mathrm{Li\text{-}curve}}]$ MAEs by 20.0% and 17.0% relative to the strongest LiFlow baseline, lowering the geometric-mean fold error in tracer diffusivity from $3.5\times$ to $2.7\times$ and improving retrieval of high-mobility candidates. The gains narrow with temperature and become statistically indistinguishable from that baseline at 1000–1200 K, consistent with the non-Markovian picture, in which path dependence strengthens at low temperature, and identifying low-temperature transport as the clearest use case for explicit memory.

LiMemFlow is designed as a pre-screening surrogate for MLIP-MD: it targets tracer transport as a screening proxy for ionic conductivity and is evaluated against its reference simulator rather than experiment. Collective diffusivity is captured better than by LiFlow but retains non-negligible errors, so shortlisted candidates still require verification by higher-fidelity MD and experiment. Because memory is learned from finite sampled history rather than an explicit kernel, accuracy remains sensitive to history length, sampling interval, and temperature; other mobile ions, broader chemistries, and longer horizons are left for future work. Lacking momenta, forces, and energy conservation, the model is a transport-oriented coarse propagator rather than a replacement for molecular dynamics.

AI USE STATEMENT

In this work, we used generative AI tools to support research execution through software code modification, debugging, and review. We did not use generative AI tools to generate synthetic datasets or to formulate or prove mathematical claims, and the remaining required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to assist with language editing and polishing of the manuscript. All AI-assisted textual edits were reviewed by the authors, and all AI-assisted code changes were manually inspected and tested before being used in the reported experiments. The authors verified all AI-assisted work and take full responsibility for the final content, code, experimental results, claims, and conclusions of this work.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of dataset construction and trajectory quality control in Appendix D, and document the model architecture and notation in Appendices E and F. Training, inference, validation-based hyperparameter selection, data splits, random-seed settings, and compute resources are reported in Appendix G. Additional ablations, screening protocols, and transport diagnostics are provided in Appendices A, B, and C. All evaluations use structure-level held-out test sets, with rollout seeds, sample sizes, and aggregation procedures specified in the corresponding sections and tables. Code and data are available through the anonymous repository linked in the abstract.

REFERENCES

Marloes Arts, Victor Garcia Satorras, Chin-Wei Huang, Daniel Zügner, Marco Federici, Cecilia Clementi, Frank Noé, Robert Pinsler, and Rianne van den Berg. Two for one: Diffusion models and force fields for coarse-grained molecular dynamics. *Journal of Chemical Theory and Computation*, 19(18):6151–6159, 2023.

John Christopher Bachman, Sokseiha Muy, Alexis Grimaud, Hao-Hsun Chang, Nir Pour, Simon F. Lux, Odysseas Paschos, Filippo Maglia, Saskia Lupart, Peter Lamp, Livia Giordano, and Yang Shao-Horn. Inorganic solid-state electrolytes for lithium batteries: Mechanisms and properties governing ion conduction. *Chemical Reviews*, 116(1):140–162, 2016.

Ardeshir Baktash, James C. Reid, Tanglaw Roman, and Debra J. Searles. Diffusion of lithium ions in lithium-argyrodite solid-state electrolytes. *npj Computational Materials*, 6:162, 2020.

Swastika Banerjee and Alexandre Tkatchenko. Non-local interactions determine local structure and lithium diffusion in solid electrolytes. *Nature Communications*, 16:1672, 2025.

Ilyes Batatia, Dávid Péter Kovács, Gregor N. C. Simm, Christoph Ortner, and Gábor Csányi. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. In *Advances in Neural Information Processing Systems*, volume 35, pp. 11423–11436, 2022.

Roberto Car and Michele Parrinello. Unified approach for molecular dynamics and density-functional theory. *Physical Review Letters*, 55(22):2471–2474, 1985.

Chi Chen, Ziheng Lu, and Francesco Ciucci. Data mining of molecular dynamics data reveals Li diffusion characteristics in garnet $Li_7La_3Zr_2O_{12}$. *Scientific Reports*, 7:40769, 2017.

Jingxuan Ding, Mayanak K. Gupta, Carolin Rosenbach, Hung-Min Lin, Naresh C. Osti, Douglas L. Abernathy, Wolfgang G. Zeier, and Olivier Delaire. Liquid-like dynamics in a solid-state lithium electrolyte. *Nature Physics*, 21:118–125, 2025.

Allan dos Santos Costa, Ilan Mitnikov, Franco Pellegrini, Ameya Daigavane, Mario Geiger, Zhonglin Cao, Karsten Kreis, Tess Smidt, Emine Kucukbenli, and Joseph Jacobson. EquiJump: Protein dynamics simulation via SO(3)-equivariant stochastic interpolants. In *ICLR 2025 Workshop on Integrating Generative and Experimental Platforms for Biomolecular Design*, 2025. Oral presentation.

Aleksander E. P. Durumeric, Yaoyi Chen, Aldo S. Pasos-Trejo, Frank Noé, and Cecilia Clementi. Learning data-efficient coarse-grained molecular dynamics from forces and noise. *Nature Communications*, 17:2493, 2026.

Theodosios Famprikis, Pieremanuele Canepa, James A. Dawson, M. Saiful Islam, and Christian Masquelier. Fundamentals of inorganic solid-state electrolytes for batteries. *Nature Materials*, 18:1278–1291, 2019.

Fuchun Ge, Yuxinxin Chen, and Pavlo O. Dral. Artificial intelligence for direct prediction of molecular dynamics across chemical space. *Journal of Chemical Theory and Computation*, 22(4): 1549–1562, 2026.

Francesca Grogan, Huan Lei, Xiantao Li, and Nathan A. Baker. Data-driven molecular modeling with the generalized Langevin equation. *Journal of Computational Physics*, 418:109633, 2020.

Isabel Hanghofer, Bernhard Gadermaier, and H. Martin R. Wilkening. Fast rotational dynamics in argyrodite-type Li_6PS_5X (X = Cl, Br, I) as seen by ^{31}P nuclear magnetic relaxation: On cation–anion coupled transport in thiophosphates. *Chemistry of Materials*, 31:4591–4597, 2019.

Xingfeng He, Yizhou Zhu, and Yifei Mo. Origin of fast ion diffusion in super-ionic conductors. *Nature Communications*, 8:15893, 2017.

Xingfeng He, Yizhou Zhu, Alexander Epstein, and Yifei Mo. Statistical variances of diffusional properties from ab initio molecular dynamics simulations. *npj Computational Materials*, 4:18, 2018.

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

Jürgen Janek and Wolfgang G. Zeier. A solid future for battery development. *Nature Energy*, 1: 16141, 2016.

KyuJung Jun, Yu Chen, Grace Wei, Xiaochen Yang, and Gerbrand Ceder. Diffusion mechanisms of fast lithium-ion conductors. *Nature Reviews Materials*, 9:887–905, 2024a.

KyuJung Jun, Byungju Lee, Ronald L. Kam, and Gerbrand Ceder. The nonexistence of a paddle-wheel effect in superionic conductors. *Proceedings of the National Academy of Sciences*, 121 (18):e2316493121, 2024b.

Leon Klein, Andrew Y. K. Foong, Tor Erlend Fjelde, Bruno Mlodozeniec, Marc Brockschmidt, Sebastian Nowozin, Frank Noé, and Ryota Tomioka. Timewarp: Transferable acceleration of molecular dynamics by learning time-coarsened dynamics. In *Advances in Neural Information Processing Systems*, volume 36, pp. 52863–52883, 2023.

Anastasia K. Lavrinenko, Theodosios Famprikis, Victor Landgraf, Jouke R. Heringa, Stef Smeets, Victor Azizi, Simone Ciarella, Marnix Wagemaker, and Alexandros Vasileiadis. Gemdat: a python toolkit for site-resolved diffusion analysis in solid-state molecular dynamics. *npj Computational Materials*, 12(1), May 2026. ISSN 2057-3960. doi: 10.1038/s41524-026-02133-7. URL https://doi.org/10.1038/s41524-026-02133-7.

Jiho Lee, Suyeon Ju, Seungwoo Hwang, Jinmu You, Jisu Jung, Youngho Kang, and Seungwu Han. Disorder-dependent Li diffusion in Li_6PS_5Cl investigated by machine-learning potential. *ACS Applied Materials & Interfaces*, 16:46442–46453, 2024.

Shaoning Li, Yusong Wang, Mingyu Li, Jian Zhang, Bin Shao, Nanning Zheng, and Jian Tang. F3low: Frame-to-frame coarse-grained molecular dynamics with SE(3) guided flow matching. In *GEM Workshop at International Conference on Learning Representations*, 2024.

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

Cibrán López, Riccardo Rurali, and Claudio Cazorla. How concerted are ionic hops in inorganic solid-state electrolytes? *Journal of the American Chemical Society*, 146(12):8269–8279, 2024.

Liyao Lyu and Huan Lei. Construction of coarse-grained molecular dynamics with many-body non-Markovian memory. *Physical Review Letters*, 131:177301, 2023.

Edward J. Maginn, Richard A. Messerly, Daniel J. Carlson, Daniel R. Roe, and J. Richard Elliot. Best practices for computing transport properties 1. Self-diffusivity and viscosity from equilibrium molecular dynamics. *Living Journal of Computational Molecular Science*, 1(1):6324, 2018.

Arumugam Manthiram, Xingwen Yu, and Shaofei Wang. Lithium battery chemistries enabled by solid-state electrolytes. *Nature Reviews Materials*, 2(4):16103, 2017.

Benjamin J. Morgan. Mechanistic origin of superionic lithium diffusion in anion-disordered Li_6PS_5X argyrodites. *Chemistry of Materials*, 33:2004–2018, 2021.

Hazime Mori. Transport, collective motion, and Brownian motion. *Progress of Theoretical Physics*, 33(3):423–455, 1965.

Juno Nam, Sulin Liu, Gavin Winter, KyuJung Jun, Soojung Yang, and Rafael Gómez-Bombarelli. Flow matching for accelerated simulation of atomic transport in crystalline materials. *Nature Machine Intelligence*, 7:1625–1635, 2025.

Michael Plainer, Hao Wu, Leon Klein, Stephan Günnemann, and Frank Noé. Consistent sampling and simulation: Molecular dynamics with energy-based diffusion models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Andrey D. Poletayev, James A. Dawson, M. Saiful Islam, and Aaron M. Lindenberg. Defect-driven anomalous transport in fast-ion conducting solid electrolytes. *Nature Materials*, 21:1066–1073, 2022.

Andrey D. Poletayev, Matthias C. Hoffmann, James A. Dawson, Samuel W. Teitelbaum, Mariano Trigo, M. Saiful Islam, and Aaron M. Lindenberg. The persistence of memory in ionic conduction probed by nonlinear optics. *Nature*, 625:691–696, 2024.

Sergey N. Pozdnyakov and Michele Ceriotti. Smooth, exact rotational symmetrization for deep learning on point clouds. In *Advances in Neural Information Processing Systems*, volume 36, pp. 79469–79501, 2023.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 8459–8468, 2020.

Mathias Schreiner, Ole Winther, and Simon Olsson. Implicit transfer operator learning: Multiple time-resolution models for molecular dynamics. In *Advances in Neural Information Processing Systems*, volume 36, pp. 36449–36462, 2023.

Yuning Shen, Lihao Wang, Huizhuo Yuan, Yan Wang, Bangji Yang, and Quanquan Gu. Simultaneous modeling of protein conformation and dynamics via autoregression. In *Advances in Neural Information Processing Systems*, 2025.

Nima Shoghi, Yuxuan Liu, Yuning Shen, Rob Brekelmans, Pan Li, and Quanquan Gu. Scalable spatio-temporal SE(3) diffusion for long-horizon protein dynamics. In *International Conference on Learning Representations*, 2026.

Fabian L. Thiemann, Thiago Reschützegger, Massimiliano Esposito, Tseden Taddese, Juan D. Olarte-Plata, and Fausto Martelli. Force-free molecular dynamics through autoregressive equivariant networks. *arXiv preprint arXiv:2503.23794*, 2025.

Sun-Ting Tsai, En-Jui Kuo, and Pratyush Tiwary. Learning molecular dynamics with simple language model built upon long short-term memory neural network. *Nature Communications*, 11: 5115, 2020.

Sun-Ting Tsai, Eric Fields, Yijia Xu, En-Jui Kuo, and Pratyush Tiwary. Path sampling of recurrent neural networks by incorporating known physics. *Nature Communications*, 13:7231, 2022.

Michael Vogel. Complex lithium ion dynamics in simulated $LiPO_3$ glass studied by means of multitime correlation functions. *Physical Review B*, 68(18):184301, 2003.

Hadrien Vroylandt, Ludovic Goudenège, Pierre Monmarché, Fabio Pietrucci, and Benjamin Rotenberg. Likelihood-based non-Markovian models from molecular dynamics. *Proceedings of the National Academy of Sciences*, 119(13):e2117586119, 2022.

Yan Wang, Lihao Wang, Yuning Shen, Yiqun Wang, Huizhuo Yuan, Yue Wu, and Quanquan Gu. Protein conformation generation via force-guided SE(3) diffusion models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 56835–56859. PMLR, 2024.

Liming Wu, Zhichao Hou, Jirui Yuan, Yu Rong, and Wenbing Huang. Equivariant spatio-temporal attentive graph networks to simulate physical dynamics. In *Advances in Neural Information Processing Systems*, volume 36, pp. 45360–45380, 2023.

Han Yang, Chenxi Hu, Yichi Zhou, Xixian Liu, Yu Shi, Jielan Li, Guanzhi Li, Zekun Chen, Shuizhou Chen, Claudio Zeni, Matthew Horton, Robert Pinsler, Andrew Fowler, Daniel Zügner, Tian Xie, Jake Smith, Lixin Sun, Qian Wang, Lingyu Kong, Chang Liu, Hongxia Hao, and Ziheng Lu. MatterSim: A deep learning atomistic model across elements, temperatures and pressures. *arXiv preprint arXiv:2405.04967*, 2024.

Robert Zwanzig. Memory effects in irreversible thermodynamics. *Physical Review*, 124(4):983–992, 1961.

A Additional Ablations and Performance Analysis

This section evaluates the sensitivity of LiMemFlow to its history configuration and tests whether its improvement extends across mobility regimes and trajectory statistics beyond single-particle MSD. Unless otherwise stated, all experiments use the 300 K test set and rollout seed 3. The mobility-stratified experiment instead reports the mean and sample standard deviation over three independent rollout seeds.

A.1 Sensitivity to History Length

To isolate the contribution of multi-step history, we retrained LiMemFlow with history lengths $M \in \{1, 2, 4, 8\}$. All variants use the same architecture, data split, training protocol, and training random seed as the $M = 6$ model; history length is the only changed hyperparameter. We evaluate all five checkpoints using the same rollout protocol and rollout seed.

Table 3: History-length ablation at 300 K. Every variant is retrained from scratch, and all entries use rollout seed 3.

History length M	$\log[\mathrm{MSD}_{\mathrm{Li}}]$ MAE ↓	$\log[\mathrm{MSD}_{\text{Li-curve}}]$ MAE ↓	Spearman ρ ↑	Stable (%) ↑
1	0.825	0.589	0.569	100.0
2	0.509	0.569	0.575	100.0
4	0.458	0.475	0.609	100.0
6 (LiMemFlow)	**0.433**	**0.459**	**0.659**	100.0
8	0.451	0.506	0.635	100.0

Increasing the history length from $M = 1$ to $M = 2$ reduces the endpoint log-MSD MAE from 0.825 to 0.509, indicating that the most recent displacement alone does not provide sufficient dynamical context in this setting. Performance improves further through $M = 6$, whereas $M = 8$ does not improve any of the three accuracy metrics. Among the tested values, $M = 6$ therefore gives the best observed performance while using a shorter context than $M = 8$. This controlled sweep supports both the use of multi-step history and the empirical choice of $M = 6$ for the 300 K model.

A.2 Sensitivity to the History Sampling Interval

The main model samples history at 1 ps intervals, matching its 1 ps autoregressive propagation step. To test this design choice during training, we retrained variants with history intervals of 0.5, 2, and 3 ps using the same data split and rollout protocol. The 0.5 ps model cannot access 0.5 ps generated frames at inference because the propagator produces one frame every 1 ps; it is therefore evaluated with the smallest available inference interval of 1 ps. The 2 and 3 ps models are evaluated with matching inference intervals.

Table 4: Training-time history-interval ablation at 300 K. All variants are retrained from scratch and evaluated using rollout seed 3. HI denotes the history interval.

History configuration	$\log[\mathrm{MSD}_{\mathrm{Li}}]$ MAE ↓	$\log[\mathrm{MSD}_{\text{Li-curve}}]$ MAE ↓	Spearman ρ ↑	Stable (%) ↑
HI = 1 ps (LiMemFlow)	**0.433**	**0.459**	**0.659**	100.0
HI = 0.5 ps; inference HI = 1 ps	1.112	0.640	0.438	100.0
HI = 2 ps	0.486	0.493	0.570	100.0
HI = 3 ps	0.507	0.515	0.605	100.0

The 1 ps interval gives the best endpoint MSD, MSD-curve, and rank-correlation results among the tested settings. Training with a 0.5 ps interval creates a train–inference mismatch under the current 1 ps propagator, while the 2 and 3 ps intervals discard intermediate generated states. The experiment therefore supports aligning the history interval with the autoregressive propagation step. The 0.5 ps result should be interpreted as evidence about this alignment, rather than evidence that finer temporal information is intrinsically harmful.

A.3 Memory-Injection Layer Placement

The main architecture injects point-wise history memory at PET layers 2 and 4 and retrieves pair-wise memory-bank information at layer 4. We compare this placement with two alternatives: moving pair-wise retrieval to layer 2 while retaining point-wise injection at layers 2 and 4, and retaining pair-wise retrieval at layer 4 while removing point-wise injection from layer 2.

Table 5: Memory-injection layer-placement ablation at 300 K using rollout seed 3. PW and PA denote point-wise and pair-wise memory, respectively.

Memory placement	$\log[\mathrm{MSD_{Li}}]$ MAE ↓	$\log[\mathrm{MSD_{Li\text{-}curve}}]$ MAE ↓	Spearman ρ ↑	Stable (%) ↑
PW [2,4], PA [4] (LiMemFlow)	**0.433**	**0.459**	**0.659**	100.0
PW [2,4], PA [2]	0.635	0.477	0.649	100.0
PW [4], PA [4]	0.595	0.509	0.618	100.0

The main placement performs best on all reported accuracy metrics. Removing the layer-2 point-wise update degrades both endpoint and curve-level MSD errors, indicating that intermediate point-wise conditioning helps historical information influence subsequent message passing. Moving pair-wise retrieval from layer 4 to layer 2 also worsens performance, consistent with retrieving pair-wise history after the edge representations have accumulated broader structural context.

A.4 Mobility-Stratified Performance

To determine whether the aggregate gain is concentrated in a particular transport regime, we divide the 94 structures in the 300 K test set into tertiles according to the endpoint reference $\log[\mathrm{MSD_{Li}}]$. The resulting low-, medium-, and high-mobility bins contain 32, 31, and 31 structures, respectively. Table 6 reports the mean and sample standard deviation over three rollout seeds.

Table 6: Mobility-stratified performance at 300 K. Values are mean ± sample standard deviation over three rollout seeds. Bins are defined using tertiles of the endpoint reference $\log[\mathrm{MSD_{Li}}]$. The best result within each bin is shown in bold.

Mobility bin	Model	$\log[\mathrm{MSD_{Li}}]$ MAE ↓	$\log[\mathrm{MSD_{Li\text{-}curve}}]$ MAE ↓
Low ($n = 32$)	LiFlow(P+C)	0.4453 ± 0.0147	0.4153 ± 0.0025
	w/o Memory	0.4363 ± 0.0181	0.3726 ± 0.0089
	LiMemFlow	$\mathbf{0.3432 \pm 0.0165}$	$\mathbf{0.3340 \pm 0.0038}$
Medium ($n = 31$)	LiFlow(P+C)	0.7507 ± 0.0206	0.6957 ± 0.0122
	w/o Memory	0.7557 ± 0.0107	0.6372 ± 0.0052
	LiMemFlow	$\mathbf{0.5556 \pm 0.0151}$	$\mathbf{0.5374 \pm 0.0038}$
High ($n = 31$)	LiFlow(P+C)	0.4392 ± 0.0070	0.5373 ± 0.0120
	w/o Memory	0.5119 ± 0.0114	$\mathbf{0.4883 \pm 0.0043}$
	LiMemFlow	$\mathbf{0.4095 \pm 0.0097}$	0.4948 ± 0.0125

Compared with LiFlow(P+C), LiMemFlow reduces endpoint log-MSD MAE by 22.9%, 26.0%, and 6.8% in the low-, medium-, and high-mobility bins, respectively; the corresponding reductions in curve MAE are 19.6%, 22.8%, and 7.9%. Thus, the gain is not confined to the lowest-mobility structures and is largest in the medium-mobility bin. Relative to the memory-free ablation, LiMemFlow gives lower endpoint error in every bin and lower curve error in the low- and medium-mobility bins. The memory-free model has a slightly lower curve MAE in the high-mobility bin, although its endpoint error is substantially larger.

B Screening-Oriented Evaluation at 300 K

Setup We conduct a retrospective in-silico screening evaluation by treating the 94 held-out structures at 300 K as a candidate pool. A structure is operationally labeled as a high-mobility candidate

Table 7: Retrospective screening performance on the 94 held-out structures at 300 K. FN denotes the number of missed high-mobility candidates, and k_{80} is the number of top-ranked candidates required to recover at least 80% of the reference positives. Results are averaged over three rollout seeds. Best mean values are shown in bold.

Model	Recall ↑	FN ↓	MCC ↑	AP ↑	k_{80} ↓	Within-10× ↑
LiMemFlow	**0.778**± 0.017	**7.3**± 0.6	**0.594**± 0.023	**0.790**± 0.009	**38.0**± 1.0	**0.908**± 0.006
LiFlow(P+C)	0.677± 0.017	10.7± 0.6	0.476± 0.016	0.600± 0.009	41.0± 1.0	0.837± 0.006
w/o Memory	0.758± 0.000	8.0± 0.0	0.573± 0.011	0.753± 0.015	40.3± 2.1	0.830± 0.011

when its reference endpoint lithium mean-squared displacement satisfies $\mathrm{MSD}_{\mathrm{Li}}^{\mathrm{ref}}(150\,\mathrm{ps}) > 1\,\text{Å}^2$. This criterion identifies 33 positive and 61 negative structures. For threshold-based screening, the same criterion is applied to the predicted endpoint MSD.

The count of 33 positives differs from the 20 positives obtained by applying the same threshold to the precomputed `msd_Li` field in the original `test_300K.csv` index. All 20 positives under the original index definition are included among the 33 positives under the trajectory-recomputed definition, with 13 additional structures classified as positive. This difference arises from the two preprocessing paths rather than from different candidate pools or thresholds. We use the trajectory-recomputed `msd_Li_ref` as the primary screening target because it is temporally aligned with the 150 ps generated rollout and applies the same MSD definition to the prediction and reference. The original 20-positive index definition is retained only as a legacy sensitivity check.

Recall and the false-negative count (FN) measure the ability to avoid missing high-mobility candidates, while the Matthews correlation coefficient (MCC) measures balanced classification performance. For ranking-based screening, candidates are ordered by their predicted endpoint lithium MSD. We report average precision (AP) and k_{80}, where k_{80} is the number of top-ranked candidates that would need to be verified by reference MLIP-MD to recover at least 80% of the 33 high-mobility candidates. Finally, Within-10× denotes the fraction of structures satisfying

$$\left|\log_{10}\mathrm{MSD}_{\mathrm{Li}}^{\mathrm{pred}} - \log_{10}\mathrm{MSD}_{\mathrm{Li}}^{\mathrm{ref}}\right| \leq 1.$$

Reported values are averaged over three rollout seeds.

Results Under the endpoint-MSD threshold, LiMemFlow achieves a recall of 0.778, compared with 0.677 for LiFlow(P+C) and 0.758 for the memory-free ablation. It reduces the mean number of missed high-mobility candidates from 10.7 and 8.0 to 7.3, respectively, and achieves the highest MCC of 0.594. The ranking-based results show the same trend: LiMemFlow obtains an AP of 0.790 and requires verification of 38.0 candidates to recover at least 80% of the reference positives, compared with 41.0 for LiFlow(P+C) and 40.3 for the memory-free model. LiMemFlow also places 90.8% of its endpoint-MSD predictions within one order of magnitude of the reference values.

These results show that the improved trajectory-level accuracy of LiMemFlow translates into better mean threshold classification, candidate ranking, and verification efficiency relative to LiFlow(P+C). The improvements over the memory-free ablation are smaller but consistent across all reported metrics, supporting an additional screening-level benefit from history conditioning. This retrospective evaluation uses MLIP-MD trajectories as the reference and does not constitute validation against experimental conductivity.

C Collective-Transport and Return-Hop Diagnostics

The main evaluation focuses on single-particle lithium MSD. We additionally compare collective diffusivity, the Haven ratio, and site-level return hops to test trajectory statistics that depend on correlated or path-dependent motion. The collective-transport analysis uses all 94 structures in the 300 K test set. Reference trajectories are coarse-grained to the same 1 ps grid as generated trajectories, using the same unwrapping and drift-removal procedures. These statistics are computed using GEMDAT (Lavrinenko et al., 2026).

Let D^* denote tracer diffusivity and D_σ collective diffusivity. We define the Haven ratio as $H_R = D^*/D_\sigma$. The first two rows of Table 8 report the mean structure-wise absolute log error over the 94 paired structures. The site-level analysis uses a common subset of 30 structures for which reference-based site construction succeeds and the reference and all generated trajectories contain valid consecutive hop pairs. Li sites are identified exclusively from each reference trajectory and then held fixed when mapping every trajectory to a site sequence. A back jump is an $A \to B$ hop immediately followed by $B \to A$; the back-jump ratio is computed per structure and then averaged over the common subset. All reported transport diagnostics use rollout seed 3.

Table 8: Collective-transport and site-level statistics at 300 K. The collective errors are averaged over 94 paired test structures. Back-jump ratios are averaged over a common subset of 30 structures shared by all compared trajectories. For the back-jump ratio, closeness to the reference value is desired; lower is not intrinsically better. All generated-trajectory results use rollout seed 3.

Metric	Reference MD	LiMemFlow	LiFlow(P+C)	w/o Point Wise	w/o Pair Wise	w/o Memory
$\lvert\Delta \log_{10} D_\sigma\rvert \downarrow$	–	**1.29**	2.15	1.92	1.35	1.43
$\lvert\Delta \log_{10} H_R\rvert \downarrow$	–	**1.19**	1.95	1.37	1.34	1.48
Back-jump ratio	0.94	**0.91**	0.86	0.76	0.86	0.80

LiMemFlow achieves the lowest collective-diffusivity and Haven-ratio errors among the evaluated models, and its mean back-jump ratio is closest to the reference value. Removing the point-wise pathway produces the largest degradation, increasing the collective-diffusivity error from 1.29 to 1.92 and shifting the back-jump ratio from 0.91 to 0.76. Removing the pair-wise pathway also consistently worsens all three diagnostics: the collective-diffusivity error increases from 1.29 to 1.35, the Haven-ratio error increases from 1.19 to 1.34, and the back-jump ratio decreases from 0.91 to 0.86. These results indicate that point-wise history provides the primary memory contribution, while pair-wise history supplies complementary local relative-motion information for correlated and path-dependent trajectory statistics. The non-additive behavior across the ablations further suggests that the two memory pathways interact rather than contribute independently.

Scope. Compared with LiFlow(P+C), LiMemFlow reduces the collective-diffusivity error from 2.15 to 1.29 decades, although the remaining absolute error is still large (Table 8). In comparison, its endpoint tracer log-MSD MAE is 0.435 (Table 1; Section 4.1). At the available trajectory length, the collective-displacement estimator used for D_σ converges much more slowly than the single-particle MSD, particularly in systems with sparse lithium hopping events (He et al., 2018; Baktash et al., 2020). LiMemFlow is therefore positioned as a simulation-level, tracer-based model for candidate prioritization, whose screening performance is evaluated in Appendix B (Table 7).

D 300 K Dataset Details

Trajectory quality control. We initially generated molecular dynamics trajectories for $1{,}013$ lithium-containing crystalline structures, and then screened the trajectories individually to remove numerically unstable or physically unreasonable simulations. First, we used a coordinate-based diagnostic script to check whether each trajectory diverged at the late stage of the simulation. Specifically, we tracked the maximum absolute atomic coordinate in each frame. The script also computed adjacent-frame atomic displacements,

$$\Delta r_{t,i} = \|\mathbf{r}_{t+1,i} - \mathbf{r}_{t,i}\|_2, \tag{14}$$

and reported the median, 95th percentile, 99th percentile, and maximum values of both the per-frame maximum displacement and the per-frame mean displacement. For the largest abnormal adjacent-frame jump, the script further reports the corresponding time step, atom index, and atomic coordinates before and after the jump. To distinguish periodic-boundary wrapped jumps from true simulation divergence, we use the following heuristic criterion: if a trajectory contains a large isolated adjacent-frame jump, i.e., $\max_{t,i} \Delta r_{t,i} > 5$ Å, while the median adjacent-frame maximum displacement is below 1 Å, and the global coordinate magnitude remains below 100 Å, the trajectory is more likely to contain a periodic-boundary wrapped jump. In contrast, if the maximum absolute coordinate exceeds 10^3 Å, or if the coordinate scale of the final frame is more than ten times larger

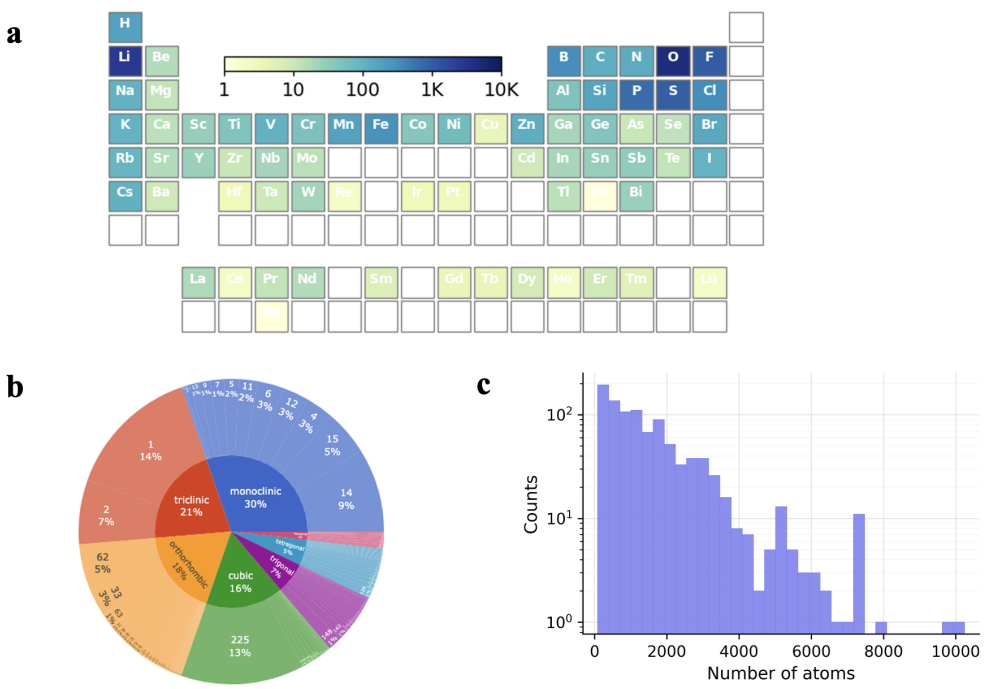


Figure 3: **Dataset statistics.** (a) Elemental count distribution across the unit cells of the structures in the dataset. (b) Space group distribution of the structures. (c) Distribution of atom counts (in the constructed supercell) per structure.

than that of the initial frame, the trajectory is regarded as late-stage divergent. Trajectories with divergent behavior are removed from the dataset.

We then apply two additional physical consistency checks. Since the simulations are performed in the NPT ensemble, the unit-cell volume should evolve smoothly along the trajectory. Therefore, for each trajectory, we examine the volume change between adjacent saved frames; if the adjacent-frame volume change never exceeds 0.1 throughout the trajectory, we regard its volume evolution as physically reasonable. Finally, we compare the long-time mean squared displacement (MSD) of lithium atoms and framework atoms. Since lithium atoms are light and are typically the primary mobile species in these materials, the framework MSD is expected to be smaller than the lithium MSD. Therefore, for trajectories where both the lithium MSD and framework MSD are larger than 10 Å^2, we remove the trajectory if the framework MSD is larger than the lithium MSD, as this indicates unphysical framework motion. When both MSD values are below 10 Å^2, however, a slightly larger framework MSD can arise from numerical or finite-sampling errors, so we do not remove trajectories based on this criterion. After the above quality-control procedure, 974 trajectories are retained to form the final 300 K dataset.

E PET Backbone Architecture

This appendix gives the implementation details of the dual-geometry PET backbone used in LiMemFlow. We use the same notation as Section 3.2: X_τ denotes the conditioning configuration, X_t denotes the intermediate flow state, H_τ denotes the finite displacement history, T denotes temperature, and $t \in [0, 1]$ denotes flow time. The memory pathways are described in Section 3.2; here they appear only as selected-layer residual hooks inside the PET trunk.

Dual-geometry graph construction. For each propagation step, the backbone predicts

$$\widehat{V} = P_\theta(X_t, X_\tau, H_\tau, T, t) \in \mathbb{R}^{N\times 3}. \tag{7}$$

The radius graph is constructed from the conditioning configuration X_τ, and the same directed edge set $\mathcal{E}_\tau$ is used throughout the flow update. This fixed graph allows the backbone to compare the current flow geometry against the reference geometry at the propagation origin without changing the local neighborhood structure.

For each directed edge $(i, j) \in \mathcal{E}_\tau$, let s_{ij} be the periodic-boundary shift vector(periodic-boundary-condition) . The relative edge vectors in the intermediate flow state and in the conditioning configuration are

$$r_{ij}^{(t)} = x_{t,j} - x_{t,i} + s_{ij}, \qquad r_{ij}^{(\tau)} = x_{\tau,j} - x_{\tau,i} + s_{ij}, \tag{15}$$

with distances

$$d_{ij}^{(t)} = \|r_{ij}^{(t)}\|_2, \qquad d_{ij}^{(\tau)} = \|r_{ij}^{(\tau)}\|_2. \tag{16}$$

We further define the dual-geometry offset

$$\Delta r_{ij} = r_{ij}^{(t)} - r_{ij}^{(\tau)}. \tag{17}$$

Thus, every edge token observes both the local geometry of the current flow state and its deviation from the conditioning geometry on the same periodic graph.

Edge and node tokenization. For each edge (i, j), LiMemFlow constructs the 12-dimensional dual-geometry descriptor

$$g_{ij} = \left[r_{ij}^{(t)}, d_{ij}^{(t)}, r_{ij}^{(\tau)}, d_{ij}^{(\tau)}, \Delta r_{ij}, \|\Delta r_{ij}\|_2 \right] \in \mathbb{R}^{12}. \tag{18}$$

A learnable linear projection maps this descriptor into the hidden dimension:

$$e_{ij}^{\text{geo}} = W_e g_{ij} \tag{19}$$

where W_e denotes a learnable linear projection from the 12-dimensional geometric descriptor to the PET hidden dimension. Invalid padded neighbor slots are excluded by a neighbor-validity mask χ_i.

The initial node token combines atom type, flow time, and temperature:

$$h_i^{(0)} = \text{Emb}(z_i) + \gamma_t(t) + \gamma_T(T), \tag{20}$$

where $\text{Emb}(\cdot)$ is a learnable element embedding and γ_t, γ_T are Gaussian Fourier embeddings. The backbone also includes a lightweight displacement-conditioning branch. Let

$$\delta x_i = x_{t,i} - x_{\tau,i}. \tag{21}$$

Each Cartesian component is projected into the hidden space and averaged:

$$q_i = \frac{1}{3} \sum_{c=1}^{3} W_{\text{disp}}\, \delta x_{i,c}. \tag{22}$$

The node token is then updated by

$$h_i^{(0)} \leftarrow h_i^{(0)} + \phi_{\text{node}}(q_i). \tag{23}$$

This branch gives the PET trunk direct access to the atom-wise offset between X_t and X_τ.

PET message-passing blocks. The backbone contains L stacked PET blocks. Let $m_{ij}^{(\ell-1)} \in \mathbb{R}^D$ be the incoming edge-message state before block ℓ, initialized as $m_{ij}^{(0)} = 0$. In the first PET block, the edge token is compressed from the geometric embedding and the incoming message:

$$\tilde{e}_{ij}^{(\ell)} = \phi_{\text{cmp}}^{(\ell)} \left([e_{ij}^{\text{geo}}, m_{ij}^{(\ell-1)}] \right), \qquad \ell = 1. \tag{24}$$

For deeper blocks, the neighbor element embedding is also included, where z_j denotes the element type of the neighbor atom j:

$$\tilde{e}_{ij}^{(\ell)} = \phi_{\text{cmp}}^{(\ell)} \left([e_{ij}^{\text{geo}}, \text{Emb}(z_j), m_{ij}^{(\ell-1)}] \right), \qquad \ell \geq 2. \tag{25}$$

For center atom i, the compressed edge tokens are collected as

$$\widetilde{E}_i^{(\ell)} = \{ \tilde{e}_{ij}^{(\ell)} : j \in \mathcal{N}(i) \}. \tag{26}$$

In selected PET layers, the memory pathways from Section 3.2 are applied before local self-attention:

$$\widetilde{E}_i^{(\ell)} \leftarrow \Psi_{\mathrm{mem}}^{(\ell)}\left(\widetilde{E}_i^{(\ell)}, C_i, B_i, \mu_i^{\mathrm{hist}}, \chi_i\right). \tag{27}$$

Equivalently, this residual hook may contain the point-wise update $\Delta E_{i,\mathrm{ca}}^{(\ell)}$, the memory-bank update $\Delta E_{i,\mathrm{mb}}^{(\ell)}$, or both, depending on the selected layer. The details of C_i, B_i, and their cross-attention readout are not repeated here.

Each PET block forms a local token sequence from one center-node token and its edge tokens:

$$s_i^{(\ell,0)} = [h_i^{(\ell-1)}, \widetilde{E}_i^{(\ell)}]. \tag{28}$$

A masked Transformer stack updates this sequence:

$$\bar{s}_i^{(\ell,r)} = \mathrm{LN}\left(s_i^{(\ell,r-1)} + \mathrm{MHSA}(s_i^{(\ell,r-1)}, \chi_i)\right), \tag{29}$$

$$s_i^{(\ell,r)} = \mathrm{LN}\left(\bar{s}_i^{(\ell,r)} + \mathrm{FFN}(\bar{s}_i^{(\ell,r)})\right), \qquad r = 1, \ldots, R. \tag{30}$$

The mask is defined over the center token and the K edge-token slots,

$$\chi_i = [1, \chi_{i1}, \ldots, \chi_{iK}], \tag{31}$$

so attention is applied only to valid neighbor edges. After the Transformer stack,

$$h_i^{(\ell)}, \{e_{ij}^{(\ell)}\}_{j\in\mathcal{N}(i)} = \mathrm{PET}^{(\ell)}(s_i^{(\ell,0)}). \tag{32}$$

LiMemFlow further refines each directed edge message using its reverse edge. Let $e_{ji}^{(\ell)}$ denote the PET output for the reverse direction. The reverse-edge correction is

$$\Delta m_{ij}^{(\ell)} = \phi_{\mathrm{rev}}^{(\ell)}\left(\mathrm{LN}([e_{ij}^{(\ell)}, e_{ji}^{(\ell)}])\right), \tag{33}$$

and the edge-message recursion is

$$m_{ij}^{(\ell)} = m_{ij}^{(\ell-1)} + e_{ij}^{(\ell)} + \Delta m_{ij}^{(\ell)}. \tag{34}$$

Equivariant velocity readout. After the final PET block, each final edge message is mapped to a scalar coefficient:

$$a_{ij} = w(m_{ij}^{(L)}). \tag{35}$$

The edge-based velocity for atom i is computed as

$$\widehat{v}_i^{\mathrm{edge}} = \sum_{j\in\mathcal{N}(i)} a_{ij} r_{ij}^{(t)}. \tag{36}$$

Because this readout is expressed through relative edge vectors, it is translation invariant and rotates consistently with the input geometry.

The displacement-conditioning branch produces an additional residual velocity:

$$\widehat{v}_i^{\mathrm{disp}} = \eta\, \phi_{\mathrm{out}}(q_i), \tag{37}$$

where η is a learnable residual scale. The final velocity is

$$\widehat{v}_i = \widehat{v}_i^{\mathrm{edge}} + \widehat{v}_i^{\mathrm{disp}}. \tag{38}$$

Collecting all atoms gives

$$\widehat{V} = [\widehat{v}_1, \ldots, \widehat{v}_N]^\top \in \mathbb{R}^{N\times 3}. \tag{39}$$

This dual-geometry PET backbone supplies the geometry-aware flow-matching propagator, while the point-wise and local spatio-temporal memory modules enter as residual conditioning paths in selected PET layers.

F NOTATION FOR THE METHOD

Symbol	Definition
$\tau, t, \Delta\tau, T$	Physical time, flow-matching time, physical propagation interval, and simulation temperature, respectively.
N, i, j, m, ℓ	Number of atoms; atom indices; historical-step index; and PET-layer index.
$\Gamma_\tau, \Gamma_{<\tau}$	Complete phase-space state at time τ and its history before τ.
Q_τ, P_τ, Ξ_τ	Atomic positions, atomic momenta, and unresolved environmental or lattice degrees of freedom.
p, p_θ, θ	Conditional transition density, learned transition density, and trainable model parameters.
$X_\tau, X_{\tau+\Delta\tau}, X_{0:\tau}, X_t$	Conditioning configuration, target configuration, observed position history, and intermediate flow state.
$A(\tau), \Pi$	Projected observable and projection operator from the complete state to the observed variables.
$F, K, s, \eta(\tau)$	Instantaneous term, memory kernel, integration variable, and random force in the generalized Langevin equation.
$H_\tau, H_{\tau,i}, M_i$	Global displacement history, atom-specific displacement history, and number of valid historical increments for atom i.
$x_i(\tau_m), \tau_m, u_i^{(m)}$	Position of atom i at the m-th historical sampling time, the sampling time itself, and the corresponding displacement increment.
$P_\theta, \widehat{V}$	LiMemFlow velocity predictor and its predicted $N\times3$ Cartesian velocity field.
$\mathcal{L}_{\mathrm{FM}}, \mathbb{E}, \mathcal{U}(0,1)$	Flow-matching objective, expectation operator, and uniform distribution over flow time.
$p_t, V_t^\star, \|\cdot\|_F$	Flow-matching probability path, target velocity field, and Frobenius norm over atoms and Cartesian coordinates.
$\phi_{\mathrm{hist}}, \bar{c}_i^{(m)}, \bar{C}i$	Learnable history-token encoder, base token for historical step m, and sequence of base history tokens.
$\|\cdot\|_2$	Euclidean norm of an atomic displacement vector.
$\mathrm{Transformer}_{\mathrm{hist}}, \mu_i^{\mathrm{hist}}, C_i$	Temporal history Transformer, history-validity mask, and contextualized point-wise history tokens.
$\widetilde{E}_i^{(\ell)}$, CrossAttn	Current edge-token set centered at atom i in PET layer ℓ, and the cross-attention operator.
$\Delta E_{i,\mathrm{ca}}^{(\ell)}$	Residual edge update retrieved from point-wise history through cross-attention.
$\mathcal{N}(i)$, Bank, B_i	Neighbor set of atom i, pair-time token construction operator, and pair-wise spatio-temporal memory bank.
$\Delta E_{i,\mathrm{mb}}^{(\ell)}$	Residual edge update retrieved from the pair-wise memory bank.

G TRAINING AND INFERENCE DETAILS

G.1 TRAINING DETAILS

We train LiMemFlow as a history-conditioned flow-matching propagator. For each training example, we sample a structure trajectory and randomly choose a starting frame X_τ. The target frame is $X_{\tau+\Delta\tau}$, where $\Delta\tau$ corresponds to 100 saved trajectory frames, i.e., a 1 ps propagation interval in our processed trajectories. The model is trained to predict the velocity field along a linear flow path from a prior-perturbed starting state to the target configuration. Specifically, we sample $t \sim \mathcal{U}(0,1)$ and define

$$X_t = (1-t)(X_\tau + \epsilon) + tX_{\tau+\Delta\tau}, \tag{40}$$

where ϵ is sampled from an adaptive Maxwell-Boltzmann prior (Nam et al., 2025). The target velocity is

$$V_t^\star = X_{\tau+\Delta\tau} - (X_\tau + \epsilon). \tag{41}$$

The training objective is the mean squared error between the predicted velocity field and $V_t^\star$ over atoms and Cartesian coordinates.

For history conditioning, LiMemFlow uses consecutive displacement increments before the current conditioning frame. If M history steps are available, the history tensor is

$$H_{\tau,i} = \{x_i(t_{m+1}) - x_i(t_m)\}_{m=1}^{M}, \tag{42}$$

where the last displacement ends at X_τ. When fewer than M historical displacements are available, we pad the history tensor with zeros and use a validity mask. During training, the history interval is set to 100 saved frames, matching the 1 ps transition interval.

We use random rotation and inversion augmentation during training. The same transformation is applied consistently to atomic coordinates, prior samples, historical displacement tensors, periodic shifts, future positions, and lattice vectors. Validation is performed without augmentation. Model architecture hyperparameters are reported in Table 9, and training hyperparameters are reported in Table 10.

Table 9: Main LiMemFlow architecture hyperparameters. The 300 K model is trained on the room-temperature dataset, while the 600–1200 K model is trained jointly on the high-temperature LiFlow benchmark trajectories.

Hyperparameter	300 K model	600–1200 K model
PET hidden dimension	128	256
Number of PET/GNN layers	4	4
Attention layers per PET block	2	2
Number of PET attention heads	8	16
Radial cutoff	5.0 Å	5.0 Å
Hybrid displacement branch	enabled	enabled
Hybrid residual scale init.	0.1	0.1
Memory hidden dimension	128	256
History length during training	6	4
History interval during training	100 frames	100 frames
Temporal history projector	Transformer	Transformer
Temporal Transformer layers	2	2
Temporal Transformer heads	4	4
Temporal FF multiplier	2	2
Temporal residual gate init.	0.1	0.1
Positional encoding	disabled	disabled
Point-wise cross-attention layers	[2, 4]	[2, 4]
Point-wise cross-attention heads	4	4
Point-wise memory scale	0.05	0.05
Point-wise residual gain init.	1.0	1.0
Spatio-temporal memory-bank layers	[4]	[4]
Memory-bank attention heads	2	4
Memory-bank scale	0.03	0.03
Same-neighbor bias	enabled	enabled
Same-neighbor bias max.	0.2	0.2
Memory-bank reference temperature	1000 K	1000 K

G.2 Inference and Validation-Time Residual Scale Selection

LiMemFlow is evaluated by autoregressive rollout starting from the first frame of each reference trajectory. At each rollout step, the model predicts a 1 ps transition. The flow ODE is solved using Euler integration with 10 internal flow steps. Unless otherwise stated, the history buffer is updated using the model-generated trajectory, rather than reference frames, so that evaluation matches the fully autoregressive setting used at test time.

For each rollout step, the current atomic configuration is first perturbed by an adaptive Maxwell-Boltzmann prior. The flow model then transports this prior-perturbed state toward the next atomic configuration. After each generated step, we remove the mass-weighted center-of-mass displace-

Table 10: Training hyperparameters.

Hyperparameter	300 K model	600–1200 K model
Training temperatures	300 K	600, 800, 1000, 1200 K
Time-delay steps	100	100
Physical transition interval	1 ps	1 ps
Prior	Adaptive Maxwell-Boltzmann	Adaptive Maxwell-Boltzmann
Prior scales	[[1.0, 10.0], [0.316, 3.16]]	[[1.0, 10.0], [0.316, 3.16]]
Optimizer	Adam	Adam
Learning rate	1×10^{-4}	1×10^{-4}
Weight decay	0.0	0.0
Learning-rate scheduler	StepLR	StepLR
StepLR step size	30 epochs	30 epochs
StepLR decay factor	0.5	0.5
Validation flow time points	11	11
Maximum epochs	1000	500
Validation interval	every 5 epochs	every 5 epochs
Gradient clipping	1.0	1.0
Early stopping monitor	validation loss	validation loss
Early stopping patience	20	10
Checkpoint selection	lowest validation loss	lowest validation loss
Training samples per epoch	2000	4000
Validation samples	200	400
Batch size per device	1	4
Precision	FP32	FP32
Random seed	42	42

ment to avoid global drift. The generated displacement is appended to the history buffer and becomes part of the memory context for subsequent rollout steps.

Inference-time history residual scale. LiMemFlow injects historical memory into selected PET layers through residual memory updates. During training, the point-wise memory pathway and the local spatio-temporal memory bank are conditioned on ground-truth historical displacement sequences constructed from reference trajectories. During autoregressive inference, however, the history buffer must be updated using model-generated frames. This creates an exposure-bias effect: the memory module is trained on clean reference history, but is evaluated on generated history that may gradually deviate from the training distribution. To mitigate this train-test distribution drift, we adjust the strength of residual history injection at inference time, using a scalar hyperparameter λ_{hist} to control the magnitude of the residual memory update:

$$\tilde{E}^{(\ell)} = E^{(\ell)} + \lambda_{\mathrm{hist}} \Delta E^{(\ell)}_{\mathrm{point}} + \lambda_{\mathrm{hist}} \Delta E^{(\ell)}_{\mathrm{bank}}, \tag{43}$$

where $E^{(\ell)}$ is the edge representation at layer ℓ, $\Delta E^{(\ell)}_{\mathrm{point}}$ is the point-wise temporal memory update, and $\Delta E^{(\ell)}_{\mathrm{bank}}$ is the local spatio-temporal memory-bank update. This override changes only the inference-time memory injection strength; the model weights are not retrained.

The motivation for tuning λ_{hist} is that long-horizon rollout can be sensitive to the strength of memory feedback under generated-history conditioning. If the residual scale is too small, the model approaches a nearly Markovian propagator and may underuse useful historical information. If the residual scale is too large, errors accumulated in the generated history may be amplified during autoregressive rollout. We therefore select λ_{hist} using validation trajectories only, so that the final test evaluation uses a fixed residual scale chosen without accessing the test set.

Validation sweep protocol. We perform a grid search over candidate residual scales using a validation split constructed from the training set. Specifically, we select 10% of the training structures and use their complete trajectories as validation trajectories for residual-scale selection. These validation trajectories are used only to choose the inference-time residual scale; the test set is not used

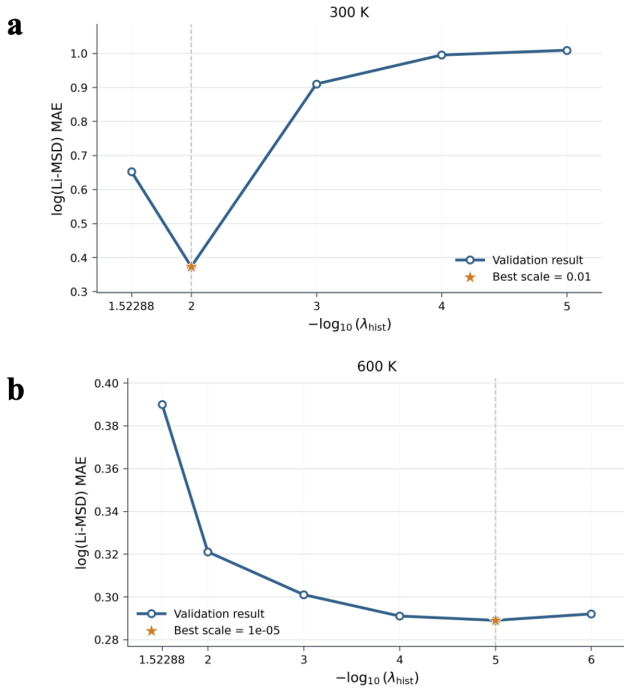


Figure 4: **Validation sweep for the inference-time history residual scale λ_{hist}.** (a) 300 K validation sweep results. (b) 600 K validation sweep results.

Table 11: Validation-time sweep used to select the inference history residual scale. The sweep changes only the inference-time memory residual strength; all model weights are fixed.

Setting	300 K model	600–1200 K model
History source	generated	generated
Solver	Euler	Euler
Flow steps per rollout step	10	10
Rollout steps during sweep	150	25
Physical rollout horizon	150 ps	25 ps
Inference history length	6	4
History interval	1 generated step	1 generated step
Number of sweep seeds	1	1
Candidate residual scales	$0.03, 0.01, 10^{-3}, 10^{-4}, 10^{-5}$	$0.03, 0.01, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$
Selection criterion	validation Li log-MSD error	validation Li log-MSD error
Selected residual scale	0.01	10^{-5}

during this selection process. For each candidate value, we load the same trained checkpoint and run the full validation rollout protocol with the same solver, rollout length, history source, and flow steps used at test time. We then compute validation transport metrics from the generated trajectories and select the residual scale that gives the lowest validation lithium $\log[\mathrm{MSD}_{\mathrm{Li}}]$ MAE. Figure 4 shows representative validation sweeps at 300 K and 600 K, where we evaluate $\log[\mathrm{MSD}_{\mathrm{Li}}]$ MAE under different values of λ_{hist}. The orange star marks the residual scale with the lowest validation error in each sweep, which is then fixed for the final test-set rollout. The detailed validation sweep settings are summarized in Table 11. For the 600–1200 K benchmark, we apply the same validation-sweep procedure separately at each temperature, so each temperature uses its own residual scale selected from the corresponding validation trajectories.

G.3 Experimental Protocol

All reported results are evaluated on structure-level held-out test sets. For both the 300 K dataset and the 600–1200 K dataset, we split structures at a 9:1 ratio during training and evaluation. No crystal structure appears in more than one split. During inference-time history residual-scale selection, we further select 10% of the training structures as validation structures for sweeping the residual-scale parameter. These validation structures are used only for selecting the inference-time history residual scale, while the held-out test structures are used only for the final reported evaluation.

For the 300 K experiments, LiFlow(P), LiFlow(P+C), and MDTrajNet are retrained on the 300 K dataset using the official code released by their respective authors. All methods are trained and evaluated using the same structure-level split, rollout horizon, and metric definitions as LiMemFlow. For the 600–1200 K experiments, the LiFlow(P) and LiFlow(P+C) results are reproduced using checkpoints released by the original LiFlow authors. MDTrajNet is retrained on the 600–1200 K dataset using the official code released by its authors. As described in Section 4, MDTrajNet is evaluated in its direct prediction form without the potential-energy-surface correction module, so that all compared methods are evaluated as trajectory generators without an additional force-field correction model.

G.4 Compute Resources and Runtime Comparison

All training and inference experiments were conducted on NVIDIA A100 GPUs with 80 GB of memory. The 300 K LiMemFlow model was trained on two A100 GPUs for approximately 24 hours, corresponding to approximately 48 A100 GPU-hours. The 600–1200 K model was trained on one A100 GPU for approximately 12 hours. Constructing the 300 K trajectory dataset required approximately 10,000 A100 GPU-hours. When multiple rollout seeds are used, the inference cost scales approximately linearly with the number of seeds.

Table 12 reports the per-structure wall-clock cost for a 150 ps trajectory at 300 K. The LiFlow(P+C), LiMemFlow, and MACE-MPA-0 timings are averages over the 300 K test set. Completing a 150 ps VASP AIMD trajectory is prohibitively expensive; we therefore measured the per-step runtime for ten structures sampled from the same dataset and extrapolated it to the 150 ps horizon. The VASP entry should consequently be interpreted as an extrapolated estimate rather than a completed end-to-end run.

Table 12: Wall-clock cost of producing a 150 ps trajectory for one 300 K structure on one NVIDIA A100 80 GB GPU. Relative cost is normalized to LiMemFlow. The VASP AIMD value is extrapolated from measured per-step runtimes on ten structures; the other values are measured averages over the 300 K test set.

Approach	Time per structure	Relative cost	Source
LiFlow(P+C)	0.67 min	$0.67\times$	measured
LiMemFlow	**1.00 min**	$1.00\times$	measured
MACE-MPA-0 (medium)	10 h	$\sim 600\times$	measured
VASP AIMD	3–4 months	$\sim 1.3 \times 10^5$–$1.7 \times 10^5\times$	extrapolated

The memory modules increase inference time by approximately $1.5\times$ relative to LiFlow(P+C), while retaining minute-scale trajectory generation. LiMemFlow is approximately $600\times$ faster than the evaluated MACE-MPA-0 MD calculation. The extrapolated comparison with VASP AIMD indicates a speedup of approximately $\sim (1.3\text{–}1.7) \times 10^5$, subject to the stated extrapolation protocol.

H Existing Assets and Licenses

We use crystal structures derived from the Materials Project and follow the Materials Project Terms of Use. We use the MACE-MPA-0 medium model, which is released under the MIT license. We also use the LiFlow lithium-transport benchmark and released checkpoints, and cite the original LiFlow paper; the LiFlow software is released under the MIT license.