Generative acceleration of molecular dynamics simulations for solid-state electrolytes

Juno Nam [1] Sulin Liu [1] Gavin Winter [1] Rafael Gómez-Bombarelli [1]

Abstract

We introduce LiFlow, a generative acceleration framework designed for efficiently simulating diffusive dynamics in solids, particularly lithium-based solid-state electrolytes (SSEs). LiFlow consists of two components: *Propagator* and *Corrector*, which utilize a conditional flow matching scheme to predict atomic displacements and perform denoising, respectively. Our model achieves a Spearman's rank correlation of approximately 0.7 for the lithium mean squared displacement (MSD) on test set based on composition and temperature splits and offers a substantial speedup compared to reference molecular dynamics (MD) simulations using machine learning interatomic potentials (MLIPs). This framework facilitates high-throughput virtual screening for electrolyte materials and holds promise for the optimization of the kinetic properties of crystalline solids.

1. Introduction

Solid-state electrolytes (SSEs) are an emerging class of materials for electrical energy storage, offering a safer and more stable alternative to the liquid electrolytes commonly used in lithium-ion batteries (Bachman et al., 2016). The study and design of SSEs require fast and accurate atomistic simulation techniques to model the intricate ionic diffusion behaviors in such materials. The standard method, *ab initio* molecular dynamics (AIMD), involves costly density functional theory (DFT) calculations for each propagation step. Hence, their application is limited to small spatiotemporal scales and a few simulations, often insufficient for characterizing diffusive dynamics or screening candidate materials. Recently, universal machine learning interatomic potentials (MLIPs), trained on large-scale DFT calculations, have emerged as a promising alternative (Friederich et al., 2021; Ko & Ong, 2023). Still, even with MLIPs, dynamics must be discretized at sufficiently small time steps to ensure stable and accurate propagation (Fu et al., 2023a), limiting the scalable analysis of large-scale materials databases.

In the context of general MD simulations, methods such as Timewarp (Klein et al., 2024), Implicit Transfer Operator Learning (Schreiner et al., 2024), and Score Dynamics (Hsu et al., 2024) have been proposed to tackle the challenge of accelerating the simulation. These methods leverage a generative modeling framework to propagate the conformational distribution from time τ to time $\tau + \Delta\tau$, where $\Delta\tau$ is much larger than the typical MD time steps. A similar approach has been applied to temporally coarse-graining polymer electrolyte simulations (Fu et al., 2023b).

Building upon these methods, this work aims to develop a tailored generative modeling framework, specifically designed for cost-effective simulation of diffusive dynamics in SSEs. Our primary objective is to construct a model that accurately reproduces relevant kinetic observables, such as mean squared displacement (MSD) and diffusivity of mobile ions, compared to long-time MD simulations utilizing MLIPs. Preferably, the model would exhibit chemical (encompassing electrolytes with diverse elemental compositions) and thermal (spanning various simulation temperatures) transferability, thereby offering a generalizable enhancement in simulation efficiency.

2. Methods

2.1. Overview

The scheme for LiFlow is depicted in Fig. 1. LiFlow models the distribution of atom positions in a system with a periodic boundary at time $\tau + \Delta\tau$ given the positions at time τ. Here, we fix the $\Delta\tau$ to 1 ps, which is 10^3 times larger than the MD time step $\delta\tau = 1$ fs. Inspired by Fu et al. (2023b), LiFlow consist of two modules, *Propagator* and *Corrector*, which are both flow matching generative models that generate a displacement vector for each atom in a system. Given atom positions $\mathbf{x}_\tau$ at time τ, *Propagator* samples a displacement vector $\mathbf{d}^P = \tilde{\mathbf{x}}_{\tau+\Delta\tau} - \mathbf{x}_\tau$. The displacement is then added to $\mathbf{x}_\tau$ to give a candidate atom positions $\tilde{\mathbf{x}}_{\tau+\Delta\tau}$ at

[1] Department of Materials Science and Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA. Correspondence to: Rafael Gómez-Bombarelli <rafagb@mit.edu>.

Accepted at the 1st Machine Learning for Life and Material Sciences Workshop at ICML 2024. Copyright 2024 by the author(s).

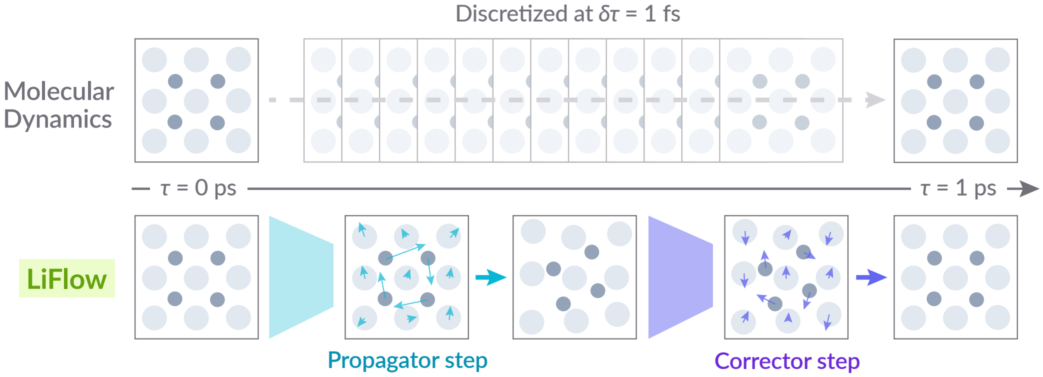

Figure 1. **LiFlow scheme.** LiFlow is a generative acceleration framework for SSE MD simulations, with *Propagator* and *Corrector* components leveraging a conditional flow matching scheme for accurate prediction of atomic displacements during time propagation.

the next time step. Since the displacements generated by *Propagator* model may not be accurate, we introduce an additional generative model, *Corrector*, which samples a displacement vector $\mathbf{d}^C = \mathbf{x}_{\tau+\Delta\tau} - \tilde{\mathbf{x}}_{\tau+\Delta\tau}$ to "denoise" or correct the atom positions. Similarly, the displacement is added to $\tilde{\mathbf{x}}_{\tau+\Delta\tau}$ to give a final prediction of the new atom positions $\mathbf{x}_{\tau+\Delta\tau}$.

Notations We denote the physical time by $\tau \in \mathbb{R}_{\geq 0}$ and flow matching time by $t \in [0, 1]$, which are associated with atomic positions $\mathbf{x}_\tau \in \mathbb{R}^{N\times 3}$ and displacements $\mathbf{d}_t^P, \mathbf{d}_t^C \in \mathbb{R}^{N\times 3}$ from the *Propagator* and *Corrector*, respectively. To featurize the input graph and propagate the dynamics, we additionally utilize lattice array $L \in \mathbb{R}^{3\times 3}$ and atom types $\mathbf{a} \in \mathbb{N}^N$, atomic masses $\mathbf{m} \in \mathbb{R}^N_{>0}$ and the temperature $T \in \mathbb{R}_{\geq 0}$. At inference time, we discretize the flow matching generation into N_{flow} steps, and iterate the LiFlow inference N_{step} times to generate the final atom positions at time $N_{\text{step}}\Delta\tau$.

2.2. Conditional flow matching models for time propagation and correction

Conditional flow matching. Flow matching (Lipman et al., 2023; Tong et al., 2024) is a generative modeling framework in which samples from the prior distribution $\mathbf{d}_0 \sim p_0(\mathbf{d})$ is transported to samples from the data distribution $\mathbf{d}_1 \sim p_1(\mathbf{d})$ by a time-dependent vector field $u_t(\mathbf{d})$. Conditional flow matching render the vector field tractable by conditioning the time-dependent probability path and the vector field on the data, i.e., $p_t(\mathbf{d}|\mathbf{d}_1)$ and $u_t(\mathbf{d}|\mathbf{d}_1)$, respectively. The marginal vector field model $v_t(\mathbf{d}; \theta)$ is parametrized by a neural network and learned by the following regression objective:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t,\mathbf{d}_1,\mathbf{d}} \left\| v_t(\mathbf{d}; \theta) - u_t(\mathbf{d}|\mathbf{d}_1) \right\|^2, \quad (1)$$

where $t \sim \mathcal{U}[0, 1]$, $\mathbf{d}_1 \sim p_1(\mathbf{d})$, and $\mathbf{d} \sim p_t(\mathbf{d}|\mathbf{d}_1)$. Following Jing et al. (2024), we choose the linear interpolation between the prior sample and the data sample as a conditional flow, i.e.,

$$u_t(\mathbf{d}|\mathbf{d}_1) = (\mathbf{d}_1 - \mathbf{d})/(1 - t). \quad (2)$$

Accordingly, we can alternatively predict $\mathbf{d}_1$ instead of the vector field v_t. By defining a final displacement predictor $\hat{\mathbf{d}}_1(\mathbf{d}, t; \theta)$, we parametrize the marginal vector field by

$$v_t(\mathbf{d}; \theta) = (\hat{\mathbf{d}}_1(\mathbf{d}, t; \theta) - \mathbf{d})/(1 - t). \quad (3)$$

The *Propagator* and *Corrector* in our scheme are hence defined as "$\mathbf{d}_1$ predictors," where we predict the desired displacements for propagation and correction and interpolate with the current displacement in each flow matching step.

Maxwell–Boltzmann prior. We design the LiFlow model to be transferable over different temperatures by introducing the temperature conditioning into the prior distribution of flow matching schemes. We use a scaled Maxwell–Boltzmann distribution as our prior, i.e., $\mathbf{d}_0 \sim \mathcal{N}(\mathbf{0}, (k_{\text{B}}T/\mathbf{m})\sigma^2)$, where k_{B} is the Boltzmann constant and σ is a hyperparameter for the model. This choice is motivated by the observation that atoms in the system diffuse faster when the particle masses are lighter and the temperature is higher. Note that the propagation time step $\Delta\tau$ is much larger than the velocity decorrelation time, so the direct physical relationship to the thermal velocities is not retained.

2.3. Model architecture

The input to the *Propagator* and *Corrector* models are current atomic positions $\mathbf{x}_\tau$ (or $\tilde{\mathbf{x}}_\tau$), current displacements $\mathbf{d}_t$, current flow matching time t, and atomic identities $\mathbf{a}$. Given these inputs, the models should predict the desired displacement $\mathbf{d}$. Since the atomic positions $\mathbf{x}_\tau$ represent their equivalent periodic images $\mathbf{x}_\tau + \mathbf{k}L^\top$ ($\mathbf{k} \in \mathbb{Z}^{N\times 3}$), the output displacements should be *invariant* to the periodic translation of each atom. We represent the periodic atomic system as a graph with a fixed edge cutoff distance, and use PaiNN (Schütt et al., 2021) architecture to predict the displacements. The atomic identities $\mathbf{a}$ and flow matching time t are embedded as node scalar features, and current displacements $\mathbf{d}_t$ are given as node vector features. The model output layer (gated equivariant block) is modified to predict a single vector for each node, which correspond to the final displacement prediction $\hat{\mathbf{d}}_1$.

2.4. Training and inference

***Propagator* training.** The *Propagator* model is trained on the time-separated pairs of atom positions in a MD trajectory, $(\mathbf{x}_\tau, \mathbf{x}_{\tau+\Delta\tau})$. We obtain $\mathbf{d}_t^P$ from the interpolation between a sampled prior displacement $\mathbf{d}_0^P$ and the true displacement $\mathbf{d}_1^P = \mathbf{x}_{\tau+\Delta\tau} - \mathbf{x}_\tau$, according to eq. 2. Then, the L_2 loss between the model prediction $\hat{\mathbf{d}}_1^P = \mathit{Propagator}(\mathbf{x}_\tau, \mathbf{d}_t^P, L, \mathbf{a}, t)$ and the true displacement $\mathbf{d}_1$ is minimized.

***Corrector* training.** For atom position $\mathbf{x}_\tau$ sampled in a MD trajectory, we add a noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2)$ where σ is sampled from $\mathcal{U}[0, 0.75]$ (units in Å) to obtain a noisy position $\tilde{\mathbf{x}}_\tau = \mathbf{x}_\tau + \boldsymbol{\epsilon}$. To address potential instabilities at very short distances, we resolve collisions by separating atom pairs within a cutoff distance (0.3 Å) along their pairwise directions until all pairs exceed the cutoff distance. Then, the *Corrector* model is trained similarly to the *Propagator*, with the model prediction obtained as $\hat{\mathbf{d}}_1^C = \mathit{Corrector}(\tilde{\mathbf{x}}_\tau, \mathbf{d}_t^C, L, \mathbf{a}, t)$ and the true displacement $\mathbf{d}_1^C = -\boldsymbol{\epsilon}$.

LIFLOW inference. Given the initial atom positions $\mathbf{x}_0$, we alternate between the *Propagator* and *Corrector* inference for N_step steps to predict the final positions $\mathbf{x}_\tau$ at time $\tau = N_\text{step}\Delta\tau$. The inference for two flow matching models start with sampling a prior displacement $\mathbf{d}_0$ from Maxwell–Boltzmann distribution, predicting final displacement $\hat{\mathbf{d}}_1$, and interpolating with the current displacement to follow the marginal vector field in eq. 3. The flow is discretized to $N_\text{flow} = 10$ steps at inference time. The detailed inference algorithm is shown in Alg. 1.

Algorithm 1: LIFLOW Inference

```
Input: Initial position $\mathbf{x}_0$, lattice $L$, atom types $\mathbf{a}$,
       atomic masses $\mathbf{m}$, temperature $T$
Output: Predicted position $\mathbf{x}_\tau$ at $\tau = N_\text{step}\Delta\tau$
for $i_\tau \leftarrow 0$ to $N_\text{step} - 1$ do
    $\tau \leftarrow i_\tau \Delta\tau$ and $\tau' \leftarrow (i_\tau + 1)\Delta\tau$
    Sample $\mathbf{d}_0^P \sim \mathcal{N}(\mathbf{0}, (k_\text{B}T/\mathbf{m})\sigma_P^2)$
    for $i \leftarrow 0$ to $N_\text{flow} - 1$ do
        $t \leftarrow i/N_\text{flow}$ and $t' \leftarrow (i+1)/N_\text{flow}$
        $\hat{\mathbf{d}}_1^P \leftarrow \mathit{Propagator}(\mathbf{x}_\tau, \mathbf{d}_t^P, L, \mathbf{a}, t)$
        $\mathbf{d}_{t'}^P \leftarrow \mathbf{d}_t^P + (\hat{\mathbf{d}}_1^P - \mathbf{d}_t^P)/(1-t)N_\text{flow}$
         (eq. 3)
    $\tilde{\mathbf{x}}_{\tau'} \leftarrow \mathbf{x}_\tau + \mathbf{d}_1^P$   /* Propagated x */
    $\tilde{\mathbf{x}}_{\tau'} \leftarrow \text{ResolveCollision}(\tilde{\mathbf{x}}_{\tau'})$
    Sample $\mathbf{d}_0^C \sim \mathcal{N}(\mathbf{0}, (k_\text{B}T/\mathbf{m})\sigma_C^2)$
    for $i \leftarrow 0$ to $N_\text{flow} - 1$ do
        $t \leftarrow i/N_\text{flow}$ and $t' \leftarrow (i+1)/N_\text{flow}$
        $\hat{\mathbf{d}}_1^C \leftarrow \mathit{Corrector}(\tilde{\mathbf{x}}_{\tau'}, \mathbf{d}_t^C, L, \mathbf{a}, t)$
        $\mathbf{d}_{t'}^C \leftarrow \mathbf{d}_t^C + (\hat{\mathbf{d}}_1^C - \mathbf{d}_t^C)/(1-t)N_\text{flow}$
         (eq. 3)
    $\mathbf{x}_{\tau'} \leftarrow \tilde{\mathbf{x}}_{\tau'} + \mathbf{d}_1^C$   /* Corrected x */
```

3. Experiments

3.1. Datasets

To train a compositionally transferable generative model for time-shifting conformational distributions, we require long-time simulation trajectories that encompass diverse compositional spaces of solid-state materials. We fetched 4,186 lithium-containing structures from Materials Project (Jain et al., 2013) with the criteria of (1) more than 10% of the atoms are lithium, (2) band gap > 2 eV, and (3) energy over the convex hull < 0.1 eV/atom. These criteria are designed to sample various modes of lithium ion dynamics

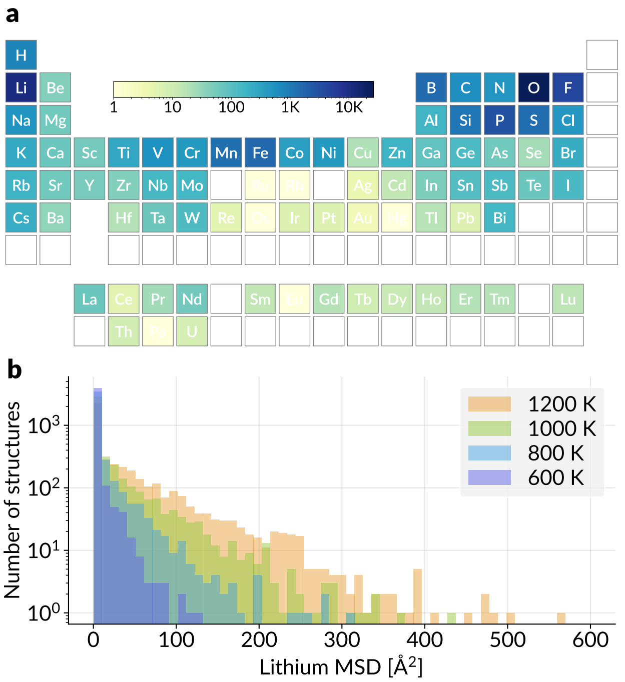

Figure 2. **Dataset statistics.** (a) Elemental count distribution across structures in the dataset. (b) Histogram of lithium MSD values from 25-ps MD simulations at different temperatures.

across different compositions, while maintaining minimal requirements for the solid-state electrolytes. After building a supercell of the structure in order to ensure that each dimension is larger than 9 Å and minimizing the structure, we conducted NVT MD simulations with MACE-MP-0 small model (Batatia et al., 2024) at 600, 800, 1000, and 1200 K for each structure. The initial velocities were assigned according to the temperature, and the system was propagated for 25 ps with the time step of 1 fs (25,000 steps) using Nosé–Hoover dynamics (Nosé, 1984; Hoover, 1985) as implemented in ASE (Larsen et al., 2017). We recorded the atom positions every ten steps.

The element distribution of the structures are shown in Fig. 2a, and it encompasses 77 elements over the periodic table. The lithium MSD per structure over the 25 ps trajectories are shown in Fig. 2b. The results indicate that the dataset encompasses a wide range of atomic environments and dynamics, including a sufficient amount of data points containing diffusive lithium atoms.

3.2. Experiment settings

Dataset split. We divided the structures based on their composition into training (90%) and testing (10%) sets. We conducted experiments using three types of splits.

1. Composition split: Trained on 800 K MD trajectories of training structures and compared the results with 800 K MD trajectories of testing structures.
2. Temperature split: Trained on MD trajectories of test structures at all temperatures except the one being tested (e.g., trained on 800, 1000, 1200 K and tested on 600 K) and compared with test MD trajectories at that T.
3. Composition + temperature split: Trained on training structures at all temperatures except the one being tested and compared with test MD trajectories at that T.

Inference setup. We conducted LiFlow inference iteratively for $N_{\text{step}} = 25$ steps to simulate dynamics over 25 ps with a time step of $\Delta\tau = 1$ ps. Each inference step involves $N_{\text{flow}} = 10$ flow matching iterations of both *Propagator* and *Corrector* models, resulting in a total of 20 forward passes of the PaiNN model per LiFlow step. During each inference process for a given structure, we terminated when either the maximum number of steps (N_{step}) was reached or the model prediction diverged due to instabilities. We then logged the number of stable propagation steps for each inference.

Metrics. To quantify the prediction of kinetic observables, we compared the MSD of lithium atoms over 25 ps to the trajectories, calculated as $\text{MSD}_{\text{Li}} = (1/N_{\text{Li}}) \sum_{i,a_i=\text{Li}} \|\mathbf{x}_{i,\tau} - \mathbf{x}_{i,0}\|^2$ where N_{Li} is the number of lithium atoms. Since the raw MSD values span wide orders of magnitude, we compared the log MSD values to the reference trajectories. We report the mean absolute error (MAE) and Spearman's rank correlation (ρ) for log MSD predictions, as well as the number of stable propagation steps in LiFlow inference.

Regression baseline. We additionally trained a regression model for log MSD values with initial structure as inputs. We used the same PaiNN architecture as the *Propagator* and *Corrector* models, but switched the output layer to predict a single scalar for each node, which are then averaged to give a structure-level prediction.

3.3. Results

Reproducing kinetic observables. We report the test metrics on various splits in Table 1. Throughout our analyses, we found a consistent Spearman correlation around 0.7 across various split scenarios, except for the final split, which is a particularly challenging case. Given that the baseline model receives identical input (initial atom positions) and is directly trained to predict the log MSD values, it demonstrates better performance on composition splits and predictions at higher temperatures. Note that the regression baseline does not provide the dynamics of individual atoms. However, LiFlow model exhibits superior performance in generalizing to lower temperature predictions. This difference in generalizability across different temperatures may stem from different magnitudes of prior displacements, which introduces complexity to the generation process at higher temperatures. Lower temperature predictions may be effectively learned from data collected at elevated temperatures, as the prior distribution at lower temperatures could potentially be encompassed within those of higher temperatures. This suggests that the model could leverage information from higher temperature data to inform its predictions at lower temperatures, offering a potential explanation for the observed performance disparity.

When trained on trajectories at identical temperatures, the model mostly overestimate the MSD, resulting in errors associated with identifying "false positive" fast lithium ion conductors (also refer to Fig. 3). Given that the true MSD values are relatively small in these cases (less than 1 Å^2), the corresponding structures typically manifest dynamic behavior wherein most lithium atoms remain within their original crystallographic sites, undergoing vibrations around their equilibrium positions. This outcome underscores the potential benefits of integrating pre-trained MLIP features into the model inputs. These features are inherently informative about the stability of the current configuration, suggesting that their incorporation could significantly enhance the prediction accuracy for low-MSD structures.

Effect of the corrector model. As evidenced by the average number of stable steps in Table 1 and illustrated in Fig. 3,

Table 1. **Prediction results.** Mean Absolute Error (MAE) and Spearman rank correlation (ρ) for the log MSD values (MSD in units of Å^2) are displayed for all models, while the average number of stable steps is reported for LiFlow models. The best results for each split are highlighted in bold, with the best among LiFlow models underlined if not bolded.

	Composition split	Temperature split		Comp. + temp. split	
Model	800 K	600 K	1200 K	600 K	1200 K
REGRESSION PREDICTOR	logMSD MAE (↓) / logMSD Spearman ρ (↑)				
Structure input	**0.33 / 0.77**	0.77 / 0.76	0.55 / **0.87**	0.88 / 0.66	**0.69 / 0.75**
LIFLOW	logMSD MAE (↓) / logMSD Spearman ρ (↑) / Average num. stable steps (↑)				
Propagator only	0.62 / 0.57 / 12.5	0.36 / 0.76 / 20.6	0.80 / 0.56 / 8.4	0.41 / 0.67 / 20.8	0.91 / 0.22 / 5.2
Propagator + *Corrector*	0.70 / 0.69 / 24.3	**0.34 / 0.77** / 23.8	**0.55** / 0.68 / 21.1	**0.36 / 0.72** / 24.4	0.78 / 0.48 / 19.2

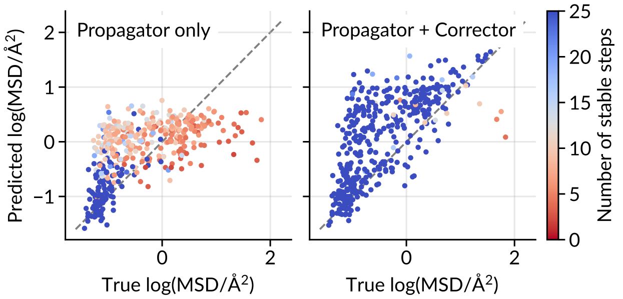

Figure 3. **Effect of the corrector model.** The scatter plot of the true and predicted log MSD values from the composition split experiment at 800 K, comparing the performance of the *Propagator*-only and *Propagator* + *Corrector* LiFlow models.

we observed enhanced stability in propagation when utilizing the *Corrector* model. In contrast, the *Propagator*-only model struggled to propagate atomic positions over significant distances, resulting in log MSD values hovering below 1. The stochastic nature of the LiFlow *Propagator* model necessitates a substantial dataset size to adequately cover the distribution of potential atomic movements over extended time intervals. However, since the data collection relies on MD simulations using MLIPs across diverse materials stuctures, it is challenging to gather a sufficiently large amount of data, as with the biomolecular simulations using classical force fields (e.g., Klein et al. (2024) and Schreiner et al. (2024)). Consequently, errors in *Propagator* predictions are inevitable, compounded by the autoregressive nature of inference, leading to divergence in propagation over time. The *Corrector* model addresses this issue by mapping erroneous atom positions after propagation to align with thermally plausible distributions, thereby stabilizing propagation and enabling longer simulation steps.

Prediction speed. Experiments were performed using a single NVIDIA RTX A5000 GPU. For a $2 \times 2 \times 1$ supercell of LGPS ($Li_{10}GeP_2S_{12}$, mp-696128) containing 200 atoms, a 25,000-step NVT MD simulation requires 530 seconds. In contrast, LiFlow model inference with $N_{\text{step}} = 25$ completes in 1.8 seconds, resulting in approximately a 300× speedup compared to MD simulation.

4. Conclusion

We proposed the LiFlow model, a generative acceleration framework tailored for lithium-based solid-state electrolyte molecular dynamics (MD) simulations. The model is composed of *Propagator* and *Corrector* components, which utilizes a conditional flow matching scheme to predict atomic displacements for time propagation and denoising, respectively. Our model achieves Spearman's rank correlation of approximately 0.7 when reproducing mean squared displacement (MSD) values on compositionally and thermally split test structures. Remarkably, LiFlow achieves a speedup of about 300× compared to reference MD simulations with machine learning interatomic potentials (MLIPs). While our model exhibits a tendency to overestimate MSD values for non-diffusing structures, we aim to address this issue by incorporating features from pretrained MLIP models to develop energy-aware propagation models, enhancing the accuracy and robustness of our approach.

While the underlying assumption regarding the sufficient accuracy of electronic structure calculations and MLIP approximations generally holds, recent reports (Deng et al., 2024) indicate that MLIPs may smooth the potential energy landscape and lead to an overestimation of kinetic properties. Therefore, it's essential to consider the accuracy of the reference dynamics. However, the methodology employed in the current study is focused on rapidly screening potential electrolyte materials, prioritizing speed over absolute accuracy in generating dynamics. Beyond facilitating high-throughput virtual screening for electrolyte materials, we envision that based on an efficient and differentiable description of ionic transport, future application of our framework will enable inverse design and optimization of electrolyte materials.

References

Bachman, J. C., Muy, S., Grimaud, A., Chang, H.-H., Pour, N., Lux, S. F., Paschos, O., Maglia, F., Lupart, S., Lamp, P., et al. Inorganic solid-state electrolytes for lithium batteries: mechanisms and properties governing ion conduction. *Chemical reviews*, 116(1):140–162, 2016.

Batatia, I., Benner, P., Chiang, Y., Elena, A. M., Kovács, D. P., Riebesell, J., Advincula, X. R., Asta, M., Avaylon, M., Baldwin, W. J., Berger, F., Bernstein, N., Bhowmik, A., Blau, S. M., Cărare, V., Darby, J. P., De, S., Pia, F. D., Deringer, V. L., Elijošius, R., El-Machachi, Z., Falcioni, F., Fako, E., Ferrari, A. C., Genreith-Schriever, A., George, J., Goodall, R. E. A., Grey, C. P., Grigorev, P., Han, S., Handley, W., Heenen, H. H., Hermansson, K., Holm, C., Jaafar, J., Hofmann, S., Jakob, K. S., Jung, H., Kapil, V., Kaplan, A. D., Karimitari, N., Kermode, J. R., Kroupa, N., Kullgren, J., Kuner, M. C., Kuryla, D., Liepuoniute, G., Margraf, J. T., Magdău, I.-B., Michaelides, A., Moore, J. H., Naik, A. A., Niblett, S. P., Norwood, S. W., O'Neill, N., Ortner, C., Persson, K. A., Reuter, K., Rosen, A. S., Schaaf, L. L., Schran, C., Shi, B. X., Sivonxay, E., Stenczel, T. K., Svahn, V., Sutton, C., Swinburne, T. D., Tilly, J., van der Oord, C., Varga-Umbrich, E., Vegge, T., Vondrák, M., Wang, Y., Witt, W. C., Zills, F., and Csányi, G. A foundation model for atomistic materials chemistry, 2024.

Deng, B., Choi, Y., Zhong, P., Riebesell, J., Anand, S., Li, Z., Jun, K., Persson, K. A., and Ceder, G. Overcoming systematic softening in universal machine learning interatomic potentials by fine-tuning, 2024.

Friederich, P., Häse, F., Proppe, J., and Aspuru-Guzik, A. Machine-learned potentials for next-generation matter simulations. *Nature Materials*, 20(6):750–761, 2021.

Fu, X., Wu, Z., Wang, W., Xie, T., Keten, S., Gomez-Bombarelli, R., and Jaakkola, T. S. Forces are not enough: Benchmark and critical evaluation for machine learning force fields with molecular simulations. *Transactions on Machine Learning Research*, 2023a. ISSN 2835-8856. URL https://openreview.net/forum?id=A8pqQipwkt.

Fu, X., Xie, T., Rebello, N. J., Olsen, B., and Jaakkola, T. S. Simulate time-integrated coarse-grained molecular dynamics with multi-scale graph networks. *Transactions on Machine Learning Research*, 2023b.

Hoover, W. G. Canonical dynamics: Equilibrium phase-space distributions. *Physical review A*, 31(3):1695, 1985.

Hsu, T., Sadigh, B., Bulatov, V., and Zhou, F. Score dynamics: Scaling molecular dynamics with picoseconds time steps via conditional diffusion model. *Journal of Chemical Theory and Computation*, Article ASAP, 2024.

Jain, A., Ong, S. P., Hautier, G., Chen, W., Richards, W. D., Dacek, S., Cholia, S., Gunter, D., Skinner, D., Ceder, G., et al. Commentary: The materials project: A materials genome approach to accelerating materials innovation. *APL Materials*, 1(1):011002, 2013. doi: 10.1063/1.4812323.

Jing, B., Berger, B., and Jaakkola, T. Alphafold meets flow matching for generating protein ensembles, 2024.

Klein, L., Foong, A., Fjelde, T., Mlodozeniec, B., Brockschmidt, M., Nowozin, S., Noé, F., and Tomioka, R. Timewarp: Transferable acceleration of molecular dynamics by learning time-coarsened dynamics. *Advances in Neural Information Processing Systems*, 36, 2024.

Ko, T. W. and Ong, S. P. Recent advances and outstanding challenges for machine learning interatomic potentials. *Nature Computational Science*, 3(12):998–1000, 2023. doi: 10.1038/s43588-023-00561-9.

Larsen, A. H., Mortensen, J. J., Blomqvist, J., Castelli, I. E., Christensen, R., Dułak, M., Friis, J., Groves, M. N., Hammer, B., Hargus, C., et al. The atomic simulation environment—a python library for working with atoms. *Journal of Physics: Condensed Matter*, 29(27):273002, 2017.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Nosé, S. A unified formulation of the constant temperature molecular dynamics methods. *The Journal of chemical physics*, 81(1):511–519, 1984.

Schreiner, M., Winther, O., and Olsson, S. Implicit transfer operator learning: Multiple time-resolution models for molecular dynamics. *Advances in Neural Information Processing Systems*, 36, 2024.

Schütt, K., Unke, O., and Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9377–9388. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/schutt21a.html.

Tong, A., Fatras, K., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Wolf, G., and Bengio, Y. Improving and generalizing flow-based generative models with minibatch optimal transport, 2024.