Lang2MLIP: End-to-End Language-to-Machine Learning Interatomic Potential Development with Autonomous Agentic Workflows

Wenwen Li [1] **Yuki Orimo** [1] **Nontawat Charoenphakdee** [1]

Abstract

Developing machine learning interatomic potentials (MLIPs) for complex materials systems remains challenging because it requires expertise in atomistic simulations, machine learning, and workflow design, as well as iterative active learning procedures. Existing automated pipelines typically assume a fixed sequence of stages or depend on domain experts, which limits their adaptability to heterogeneous materials systems where the optimal curriculum is not known in advance. To lower the barrier to developing MLIPs for non-experts, we propose **Lang2MLIP**, a multi-agent framework that takes natural-language input and formulates end-to-end MLIP development as a sequential decision-making problem solved by large language models (LLMs). At each step, a decision-making agent observes the current dataset, model, evaluation results, and execution log, and then automatically selects an appropriate action to improve the model. This removes the need for a predefined pipeline and enables the agent to self-correct by revisiting earlier subsystems when new failures arise. We evaluate this approach on a solid electrolyte interphase (SEI) system with multiple components and interfaces. These results suggest that LLM-based multi-agent systems are a promising direction for automating MLIP development and making it more accessible to non-experts.

1. Introduction

Machine learning interatomic potentials (MLIPs) have become an important tool for atomistic simulation, offering near first-principles accuracy at substantially lower computational cost than *ab initio* methods (Behler & Parrinello, 2007; Bartók et al., 2010; Drautz, 2019; Behler, 2016; Deringer et al., 2019; Unke et al., 2021). Recent advances in model architectures have further broadened the scope of MLIPs across molecules, materials, and interfaces (Schütt et al., 2018; Batatia et al., 2022; Batzner et al., 2022; Chen & Ong, 2022; Deng et al., 2023; Takamoto et al., 2022; Zhang et al., 2024; Yang et al., 2024; Rhodes et al., 2025).

Despite recent progress, developing an MLIP for a complex materials system still requires domain expertise across many stages, including structure design, simulation protocol engineering, error analysis, and active learning, as well as repeated debugging and workflow refinement (Zuo et al., 2020; Morrow & Deringer, 2022; Dai et al., 2025). These challenges are even more pronounced in heterogeneous materials systems, where diverse chemical species and local environments complicate data generation and model validation. A representative example is the battery solid electrolyte interphase (SEI) (Wang et al., 2018; Li et al., 2023; Wu et al., 2025), which contains chemically distinct components and interfacial regions whose local environments can change substantially during simulation. Training data for such systems must cover bulk-like structures, disordered configurations, interfaces, and thermally perturbed states. Existing MLIP workflows for these settings typically rely on manually designed pipelines and fixed active learning heuristics, making them difficult for non-experts to use (Novikov et al., 2020; Gong et al., 2025; Wang et al., 2025a; Lahouari et al., 2025).

In this paper, we present **Lang2MLIP**, a language-driven multi-agent framework that formulates MLIP development as a sequential decision problem solved by a tool-using LLM agent. Given only a natural-language task specification, the agent must decide at each step which action to take, how to parameterize it, and when to terminate. We identify workflow orchestration as the primary bottleneck in practical MLIP development. Lang2MLIP divides the overall process into two phases. In the *interactive preparation phase*, specialized agents interpret the task description, request missing information from the user when needed, generate initial structures, and optionally prepare a reference molecular dynamics (MD) workflow. In the *autonomous training phase*, a central decision-making agent coordinates iterative model development at each iteration by selecting from a set

[1]Preferred Networks, Inc., Tokyo, Japan. Correspondence to: Wenwen Li <wenwenli@preferred.jp>.

Accepted by ICML 2026 AI for Science Workshop, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

of actions: sampling, training, dataset selection, evaluation, pruning, reference calculation, and termination. Instead of following a fixed active-learning schedule, the agent chooses each action based on the current dataset, model state, evaluation results, and execution logs. Unless termination is selected, the decision-making agent then initiates interaction with the corresponding action-specific agent to execute the chosen operation.

We evaluate Lang2MLIP on a solid electrolyte interphase (SEI) system in lithium batteries with four layers, six components, and six elements, a setting whose multi-scale structure makes the optimal curriculum difficult to specify in advance. From natural-language interactions, the agent autonomously constructs a three-stage curriculum that progresses from basic components through binary interfaces to the full multilayer SEI, reallocating sampling effort in response to observed model failures. These results provide evidence that LLM-based sequential decision making is a viable substitute for hand-engineered active-learning pipelines in a non-trivial scientific setting.

2. Related Work

2.1. Training Dataset Preparation in MLIP Development

A central challenge in MLIP development is constructing training datasets that adequately cover the relevant configuration space, since insufficient coverage can lead to poor transferability or unreliable predictions (Novikov et al., 2020; Zuo et al., 2020; Morrow & Deringer, 2022; Dai et al., 2025). In practice, dataset preparation is therefore iterative, involving structure generation, model evaluation, and retraining. Active learning is a common strategy for improving sample efficiency, typically by alternating between candidate generation, informative sample selection based on uncertainty or error signals, and retraining (Novikov et al., 2020; Gong et al., 2025; Wang et al., 2025a; Vinod & Zaspel, 2025). These pipelines are typically structured as a fixed sequence of stages with hand-tuned criteria, and the selection among them is usually made by a human expert.

2.2. LLM for Scientific Discovery and Simulation

Recent work has explored LLMs as autonomous agents for scientific discovery, where they interact with tools, reason over intermediate outputs, and execute multi-step workflows from natural language instructions. Early systems such as ChemCrow (M. Bran et al., 2024) and Coscientist (Boiko et al., 2023) showed that tool-augmented LLMs can support or automate chemistry tasks ranging from retrosynthesis to robotic experimentation. In molecular simulation, related frameworks have been developed for molecular dynamics, polymer simulation, density functional theory (DFT), and quantum chemistry (Campbell et al., 2026; Zhao et al., 2026; Orimo et al., 2025; Wang et al., 2025b; Liu et al., 2025; Zou et al., 2025; Shi et al., 2025; Wei et al., 2025).

These systems demonstrate the potential of LLM-based agents for scientific workflows, but mainly focus on simulation setup and execution. In contrast, our work targets the iterative workflow of MLIP development, which requires dataset construction, adaptive sampling, and repeated model refinement.

2.3. Automatic Frameworks for MLIP development

Closest to our work are recent frameworks that connect natural-language instructions to trained MLIPs. AMLP (Lahouari et al., 2025) uses a multi-agent LLM system to automate DFT setup, *ab initio* molecular dynamics sampling, and fine-tuning of a MACE foundation model from a short user prompt. Nevertheless, its automation is primarily focused on task execution within a human-driven active learning framework, where users still specify key decisions such as simulation conditions, filtering thresholds, initial structure, and training hyperparameters. QUASAR (Yang & Evans, 2026) uses a three-agent architecture that coordinates DFT, MLIP training, and molecular dynamics in a unified pipeline. PFD (Wang et al., 2025a) introduces a fixed pre-training, fine-tuning, and distillation workflow that generates material-specific MLIPs from a pretrained universal model, but without an LLM-based controller.

Lang2MLIP differs from prior automated MLIP pipelines in a structural way. Existing systems such as AMLP (Lahouari et al., 2025) and QUASAR (Yang & Evans, 2026) couple LLM agents to a predefined sequence of stages (parameter recommendation, input generation, QM execution, dataset formatting, training) and retain a human in the loop for active-learning decisions. Lang2MLIP removes this dependency. Once the task is clarified, the autonomous phase selects actions via an LLM conditioned on the current state, operating without further human involvement. Our central premise is that workflow-level coordination over a set of actions is sufficient to support effective autonomous MLIP development, and that the resulting behavior, including self-correction and adaptive curriculum construction, emerges from decision making rather than from hand-engineered stage transitions.

3. Lang2MLIP Framework

3.1. Overview

Figure 1 provides an overview of Lang2MLIP, a multi-agent framework with two operational phases: an *interactive preparation phase* and an *autonomous training phase*.

In the first phase, specialized agents gather task informa-

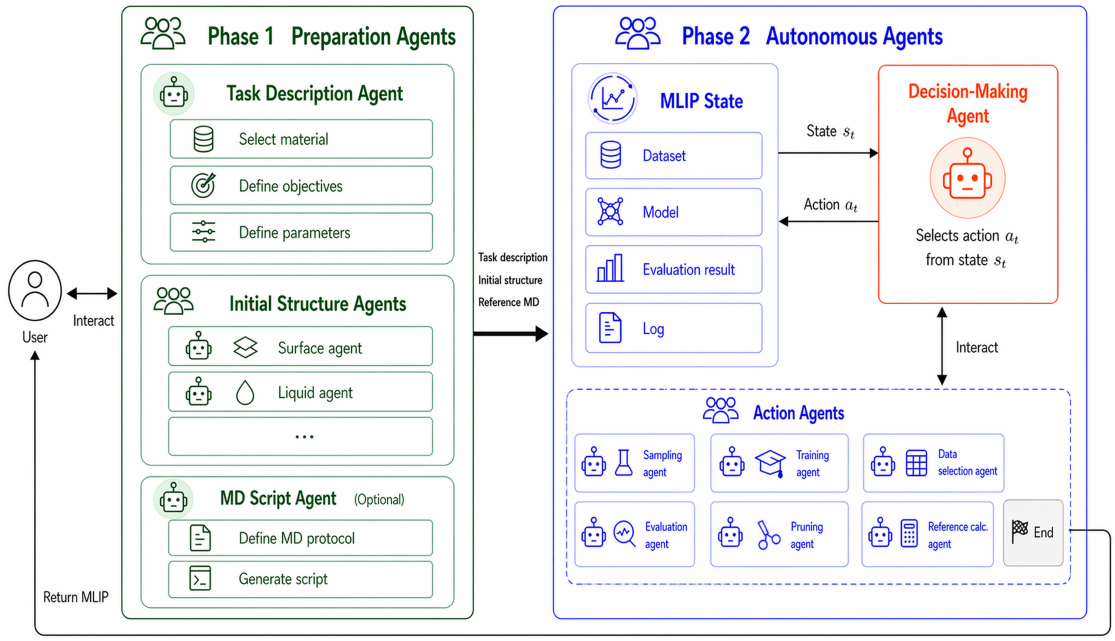

Figure 1. Overview of the Lang2MLIP framework. The robot icon indicates a single agent, while the group-of-people icon indicates multiple agents within a submodule.

tion, generate initial structures, and, optionally, prepare reference MD scripts for advanced simulation settings (e.g., non-equilibrium MD or metadynamics). This stage supports human-in-the-loop interaction, allowing users to inspect and refine task specifications with language.

In the second phase, agents collaboratively and iteratively improve the MLIP under the control of a central decision-making agent. At each step, this agent assesses the current state of the model and selects the most appropriate action to improve performance. The process continues until the model meets the desired criteria, at which point the workflow terminates and returns the MLIP to the user. This phase is designed for fully automated, long-running execution, encompassing training, failure analysis, data prioritization, and iterative refinement without manual intervention.

3.2. Multi-Agent Architecture

All agents are defined via role-specific prompts using the Claude Agent SDK [1] with Claude Opus 4.6 [2]. For example, the decision-making agent is prompted as a top-level controller responsible for determining and outputting the next task to execute (see Appendix E for a full prompt).

[1] https://code.claude.com/docs/en/agent-sdk/overview

[2] https://www.anthropic.com/news/claude-opus-4-6

3.2.1. Interactive Preparation Phase

This phase consists of three specialized agents shown in Figure 1 (left): Task Description agent, Initial Structure Agents, MD Script Agent, where Initial Structure Agents have several sub-agents. This phase is interactive because user intent and scientific scope are often ambiguous at the start of an MLIP project, and iterative clarification is essential for defining a well-posed training objective.

Task Description Agent identifies the material system, simulation objectives, and task parameters through interactive dialogue with the user.

Initial Structure Generation Agents are crucial because initial dataset quality and coverage strongly influence downstream MLIP performance. The agent first analyzes the task to identify the most informative configuration types for training. For systems with solid–liquid interfaces, for instance, it considers bulk solid, bulk liquid, and interfacial configurations. This decomposition matters because MLIPs learn from local atomic environments, e.g., interfacial regions exhibit coordination patterns not well represented by bulk phases alone. Based on this analysis, the agent coordinates a set of sub-agents (see Table 1) to construct candidate structures, systematically varying parameters such as lattice size, density, composition, and interfacial geometry, and applying reusable utility functions to improve diversity and physical plausibility. The process is interactive: the agent queries the user for missing or task-specific information (e.g., Materials Project IDs, SMILES strings, composition

Table 1. Initial structure generation agents

Agent	Scope
`Solid`	Crystalline bulk solids, including solid solutions.
`Amorphous`	Non-crystalline inorganic materials (e.g., glasses and melts).
`Molecule`	Organic molecular systems such as liquids, mixtures, and small molecules.
`Solid-surface`	Crystal surfaces and slab models.
`Cluster`	Finite clusters or nanoparticles (non-periodic).
`Solid-solid`	Interfaces between solid phases.
`Solid-molecule`	Solid-molecule interfaces and adsorption.
`Liquid-liquid`	Interfaces between molecular liquids.
`Polymer`	Long-chain molecules (e.g., polymers).
`Other`	Other structures.

details, or simulation constraints), enabling context-aware generation without requiring fully specified inputs upfront.

MD Script Generation Agent is optionally used for complex simulation scenarios . While standard ensembles (e.g., NVT or NPT) can be managed reliably by downstream agents, more advanced protocols such as non-equilibrium molecular dynamics or meta-dynamics, remain difficult to specify correctly within the autonomous training phase. To address this limitation, the agent produces specialized reference MD scripts during the preparation stage, allowing users to validate them before execution. Once verified, these scripts guide subsequent sampling during MLIP training and improve coverage of configurations relevant to the target simulation conditions.

The outputs of the interactive preparation phase are (i) a structured task description, (ii) a set of initial structures, and (iii) an optional reference MD script.

3.2.2. Autonomous training phase

Once the preparation phase is completed, the workflow enters an autonomous phase consisting of a decision-making agent and several action agents responsible for sampling, training, evaluation, and data management.

The Decision-making Agent is the central controller of the autonomous workflow.

The process can be formalized as a sequential decision problem. At each step t, the decision-making agent observes a state $s_t = (\mathcal{D}_t, \mathcal{M}_t, \mathcal{E}_t, \mathcal{L}_t)$, where $\mathcal{D}_t$ denotes the current dataset, $\mathcal{M}_t$ the current MLIP, $\mathcal{E}_t$ the accumulated evaluation results (e.g. validation errors, stability diagnostics, comparison of basic properties with reference model), and $\mathcal{L}_t$ a log of prior commands, agent-agent discussions, and any failure signals. It then decides the next action, where possible actions are outlined in Table 2. After executing the action a_t, the system updates its artifacts (e.g., sampled trajectories, selected structures, models, or evaluation results) and moves to the next state s_{t+1}. The workflow terminates when the decision-making agent selects `End`.

Table 2. Available actions for the decision-making agent.

Action	Description
`Sample`	Generate new configurations using a reference (e.g. DFT) or the current MLIP model.
`Select_data`	Select informative structures from the pool and label them
`Train`	Train or fine-tune MLIP on selected datasets.
`Model_eval`	Evaluate MLIP on validation tasks.
`Prune`	Remove redundant components or low-quality configurations.
`Reference_calc`	Run the reference method on validation tasks for future MLIP evaluation.
`End`	Finish: return MLIP and final report.

Once an action is selected, the decision-making agent interacts with the corresponding **action agents** (outlined in Table 2) to execute it. For example, if sampling is selected, the decision-making agent provides the sampling agent with the task description and relevant parameters needed to carry out the job.

4. Experimental Results

In this section, we demonstrate the effectiveness of Lang2MLIP for a battery SEI. In this experiment, we adopt the Preferred Potential (PFP), a universal interatomic potential, as the reference data source due to its high reliability and significantly lower computational cost compared to DFT (Takamoto et al., 2022). The target MLIP is based on the moment tensor potential (MTP) architecture (Novikov et al., 2020). We note that both the choice of reference data source and the MLIP architecture are flexible and can be readily replaced by other alternatives. The only requirement is that the corresponding simulation workflows such as single-point energy calculations, MD simulations and model training command, are properly exposed and executable by the agents.

4.1. Battery Solid Electrolyte Interphase (SEI)

As illustrated in Fig. 2, the system exhibits a well-defined **four-layer hierarchical structure**. Starting from the top, the electrolyte region consists of a liquid solution (1 mol/L $LiPF_6$ in EC/DMC). Beneath the electrolyte lies the organic

Table 3. Action log for the Lang2MLIP workflow on the multilayer SEI system. See Appendix G for details.

Step	Action	Description
1	`Reference_calc`	Use PFP to run MD on validation structures for reference.
2	`Sample`	Use PFP to sample all non-validation structures.
3	`Train`	Train the first MLIP model using 3312 snapshots.
4	`Sample`	Use the current MLIP to sample basic components (graphite, Li_2CO_3, LEDC, electrolyte).
5	`Select_data`	Select the top 7.5% high-error snapshots.
6	`Train`	Fine-tune the model with 364 high-error samples.
7	`Sample`	Use the current MLIP to sample interface structures (graphite/Li_2CO_3, Li_2CO_3/LEDC).
8	`Select_data`	Select high-error samples, prioritizing graphite/Li_2CO_3.
9	`Train`	Fine-tune the model with 1457 interface samples.
10	`Sample`	Use the current MLIP to re-sample interfaces at improved high temperature.
11	`Select_data`	Select 7.5% high-error samples.
12	`Train`	Fine-tune the model with 1400 high-error interface samples.
13	`Model_eval`	Evaluate the model on interface structures by comparison with PFP.
14	`Sample`	Use the current MLIP to sample LEDC/electrolyte interface structures.
15	`Select_data`	Select 7.5% high-error samples.
16	`Train`	Fine-tune the model with 812 LEDC/electrolyte interface samples.
17	`Sample`	Use the current MLIP to sample the full multilayer SEI (graphite/Li_2CO_3/LEDC/electrolyte).
18	`Select_data`	Select high-error samples from four-layer SEI interface snapshots.
19	`Train`	Fine-tune the model with 181 four-layer SEI samples.
20	`Model_eval`	Evaluate the multilayer SEI model.
21	`End`	Workflow completed: return MLIP.

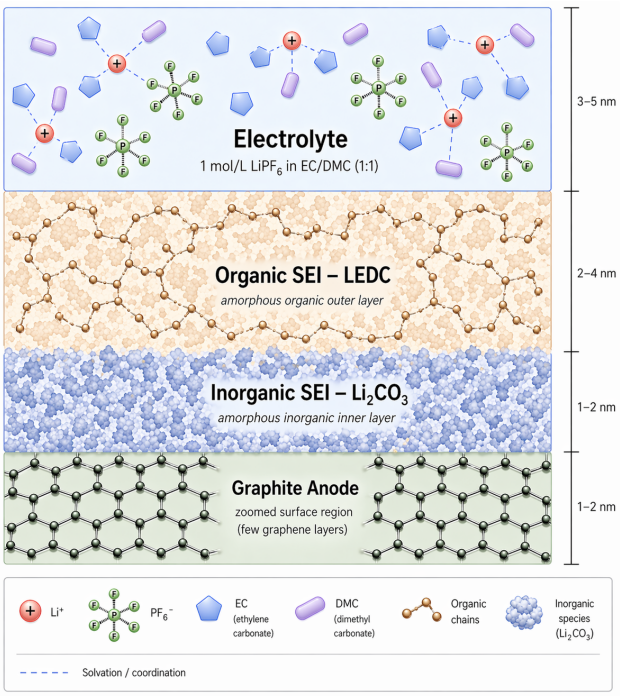

Figure 2. Schematic of the battery SEI system used in this study.

SEI outer layer, primarily composed of amorphous lithium ethylene dicarbonate (LEDC). This layer is relatively porous and mechanically soft, allowing partial transport of ions and solvent molecules. Below this is the inorganic SEI inner layer, dominated by amorphous inorganic compounds Li_2CO_3, which is denser and plays a critical role in electronically insulating the electrode while enabling Li^+ transport. Finally, the SEI is supported by the graphite anode, consisting of several stacked graphene layers forming the basal planes. This crystalline region provides the host structure for lithium intercalation. Overall, the system involves organic and inorganic components, amorphous and crystalline phases, and multiple heterogeneous interfaces. Such structural and chemical heterogeneity makes it significantly more challenging to model than conventional single-phase bulk materials, particularly for the construction of a unified MLIP.

4.2. Input Prompts

The initial user prompt is intentionally brief, i.e. "SEI of lithium battery." This prompt initiates the interactive preparation phase, during which the agent engages the user through targeted clarification questions to refine the system specification. These questions cover the system components, relevant phases, simulation methods and conditions, and the properties of interest. Once the agent determines that sufficient information has been collected, the workflow automatically transitions to the autonomous training stage, where the MLIP is developed without further human inter-

vention. The initial prompt and the subsequent user–agent interaction process are provided in the Appendix D.

4.3. Initial Structure Generation

Input structure generation constitutes the most critical and challenging component of the preparation stage. To address the complexity of the SEI system, multiple agents collaboratively generate a diverse set of candidate structures spanning different levels of structural complexity. Specifically, the generated structures cover (i) individual components, including graphite bulk, graphite $(10\bar{1}0)$ surface surfaces, amorphous Li_2CO_3, amorphous LEDC, EC/DMC electrolyte, and $LiPF_6$-EC-DMC systems; (ii) pairwise interfaces, such as Li_2CO_3/LEDC, graphite $(10\bar{1}0)$ surface/Li_2CO_3, and LEDC/electrolyte interfaces; and (iii) the complete multilayer SEI assembly.

In total, ten distinct structural categories are constructed, each corresponding to a specific phase or interfacial configuration. For each category, the agents further explore structural variations by systematically modifying parameters such as system size, density, and composition, resulting in 5–7 variants per category. Altogether, 60 initial structures are generated; details are provided in Appendix C. This hierarchical and diversified generation process ensures broad coverage of the relevant configurational space, which is essential for robust MLIP training. In addition, one structure from each category is selected as a validation structure for subsequent MLIP evaluation. To further understand the effect of reasonable initial structures for robust MLIP training, ablation studies are conducted in Section 4.5.

4.4. Autonomous Training Workflow

After the preparation agents summarize the task and generate the initial structures, the workflow enters the second stage, which is coordinated by a decision-making agent overseeing seven specialized sub-agents.

In this experiment, a total of 21 steps were carried out, with a concise summary provided in Table 3. Throughout the process, the agent performed six iterations of MLIP training, interleaved with operations such as data sampling and model evaluation. At the end of the workflow, the system produces a ready-to-use MTP model, along with corresponding validation results. Overall, this procedure closely resembles an active learning paradigm, while offering greater flexibility in decision-making and workflow adaptation.

4.5. Ablation Studies

To better understand which components of Lang2MLIP are responsible for the final performance, we conduct two ablation experiments on the SEI system. The first ablation eliminates the use of Lang2MLIP, using manually designed initial structures and a fully manual training process. The second ablation retains the agent-generated initial structures, but removes the autonomous closed-loop training procedure and replaces it with a simplified one-shot sampling-and-training workflow. These two settings allow us to separately examine the roles of the preparation phase and the autonomous training phase.

Ablation A: Manual initial structures without agent-assisted preparation. In the first ablation, we replace the preparation phase with a manually constructed initialization. Specifically, we create five initial structures corresponding to basic components of the SEI system, namely graphite bulk, graphite $(10\bar{1}0)$ surface surfaces, amorphous Li_2CO_3, amorphous LEDC and electrolyte. This choice captures the primary phases of the SEI system and is therefore physically meaningful. Such an approach reflects a setting with limited prior experience in MLIP construction: assembling a small set of known component structures and relying on MD sampling to generate additional configurations, without explicitly considering interfacial diversity or coverage. Starting from these structures, we perform sampling and subsequent MLIP training using the same reference potential as in the main experiment, but without the agent-driven preparation workflow and without the autonomous iterative refinement process. The total number of sampled structures is 3200, and the resulting model is evaluated on the same validation protocol as the full Lang2MLIP workflow.

Result During MD simulation, the trained MLIP exhibits severe instability: the system density rapidly decreases to nearly zero, and the generated structures collapse after a few picoseconds. This setting leads to a clear failure case.

These results show that a small manually designed set of basic structures is inadequate for the chemically and structurally heterogeneous SEI system, leading to poor configurational coverage and unstable downstream simulations. More broadly, the main challenge is not only training the potential, but constructing a sufficiently diverse and physically plausible starting configuration. In this context, the agent-assisted preparation phase helps lower the barrier to obtaining a trainable and stable MLIP setup, especially for users who may not know in advance which structures, interfaces, and compositions to include.

Ablation B: Agent-generated initialization without autonomous iterative refinement. In the second ablation, we keep the agent-generated initial structures from the preparation phase but remove the autonomous training loop. Concretely, the workflow starts from the same initial structures as the full Lang2MLIP pipeline, then applies only a simplified sampling-and-training procedure without iterative model evaluation, error-driven data selection, or adaptive resampling. The resulting training set contains 3312 struc-

tures, and the trained model is evaluated by comparing density, mean squared displacement (MSD), and radial distribution function (RDF) against the PFP reference and the Lang2MLIP model.

Result Compared with Ablation A, this setting yields a substantially more stable model and physical property can be calculated. Figure 3 and Table 4 show the comparison with full Lang2MLIP method, where PFP is a reference method. MD simulations run normally without catastrophic structural collapse, and the predicted density agrees reasonably well with the PFP reference, reaching 1.500 g/cm3 compared with 1.485 g/cm3. However, clear discrepancies remain in more sensitive structural and dynamical observables. In particular, the mean squared displacement (MSD) of Li, C, and O atoms indicates one order of magnitude slower diffusion than in PFP. Moreover, the radial distribution functions (RDFs) still exhibit visible errors, especially for inorganic SEI and electrolyte parts. In contrast, the full Lang2MLIP model achieves closer agreement with PFP on these observables, as shown in Fig. 3 and Table 4. These results suggest that high-quality initialization alone is not sufficient for accurate MLIP construction, and that iterative refinement is necessary to improve quantitative fidelity in local structure and atomic dynamics. In this sense, the autonomous training agents fully automate iterative refinement, eliminating the need for expert-designed workflows such as active learning setup, and thereby lowering the barrier to reaching expert-comparable MLIP quality. Full validation results are given in Appendix B.

Discussion Taken together, the two ablations reveal a complementary division of labor between the two stages of Lang2MLIP. Without the preparation phase, the resulting MLIP cannot maintain stable MD trajectories, showing that initialization quality is a prerequisite for successful training. With agent-generated initialization but without autonomous closed-loop refinement, the model becomes stable but still underperforms on dynamical and local structural observables. Only the full Lang2MLIP workflow achieves both robust simulation stability and closer agreement with PFP across density, MSD, and RDF metrics.

These findings suggest that the benefit of Lang2MLIP does not arise from a single component in isolation. Instead, the preparation phase primarily determines whether the training problem is well-posed, whereas the autonomous training phase determines whether the final potential becomes quantitatively faithful. This complementary effect is especially important in complex interfacial systems such as SEI, where both broad configurational coverage and targeted iterative refinement are required.

Table 4. Ablation results on the SEI system. Diffusion coefficients are computed from the MSD curves. PFP is the reference method. The Ablation A method is omitted due to instability during MD simulations, which prevents reliable evaluation of physical properties. Lang2MLIP w/o active corresponds to method in Ablation B.

Metric	Lang2MLIP w/o active	Lang2MLIP	PFP (ref.)
Density (g/cm^3)	1.500	1.477	1.485
D_{Li} (cm^2/s)	-4.61×10^{-7}	8.00×10^{-7}	7.49×10^{-7}
D_O (cm^2/s)	4.16×10^{-7}	2.58×10^{-6}	2.29×10^{-6}
D_C (cm^2/s)	1.3×10^{-7}	1.3×10^{-6}	1.11×10^{-6}
D_P (cm^2/s)	8.55×10^{-6}	1.03×10^{-5}	1.59×10^{-5}

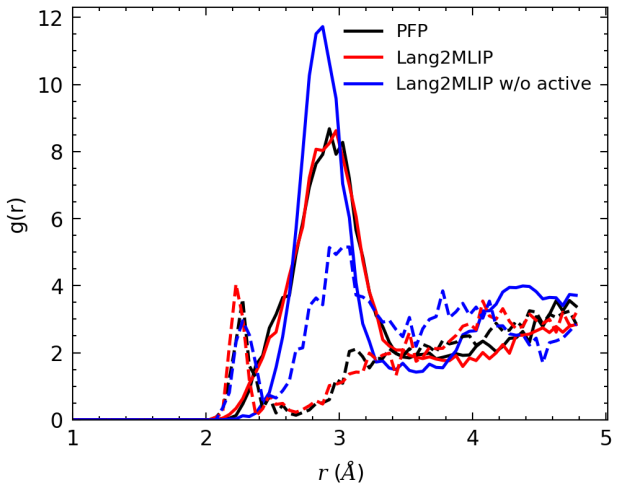

Figure 3. Radial distribution functions (RDFs) for selected atom pairs in the multilayer SEI system at 300 K. Solid lines show the Li–C pair in Li_2CO_3; dashed lines show the O–F pair in the electrolyte.

4.6. Emergent Adaptive Behaviors

Beyond overall task completion, we analyze the execution trace in Table 3 to understand how the agent organizes the MLIP development process. Despite operating over an action space without predefined stage transitions, the system exhibits several non-trivial adaptive behaviors that resemble expert-designed workflows.

Emergent curriculum over increasing complexity The agent autonomously organizes training into a three-stage progression: (i) basic components (graphite, Li_2CO_3, LEDC, electrolyte), (ii) pairwise interfaces (e.g., graphite/Li_2CO_3, Li_2CO_3/LEDC), and (iii) the full multilayer SEI system. This progression is not explicitly predefined, but emerges naturally from the sequential decision process, moving from simpler subsystems to more complex interfacial and multicomponent configurations – a strategy that reflects expert-level intuition in MLIP development (Li et al., 2025; Liang et al., 2025).

Failure-driven data acquisition The agent allocates sampling effort based not only on error magnitude but also

on the type and location of failure. During interface sampling (step 7), Li_2CO_3/LEDC systems remain stable across 300–800 K, whereas graphite/Li_2CO_3 simulations exhibit frequent instabilities, with 14 out of 30 runs terminating early due to structural collapse. In response, the agent prioritizes graphite/Li_2CO_3 by allocating a larger fraction of selected high-error samples (17.5% vs. 7.5%) at step 8. This behavior indicates that the agent distinguishes between subsystems and concentrates resources on those that limit model robustness, rather than treating all regions uniformly.

Adaptive exploration and iterative refinement The workflow exhibits iterative cycles of sampling, selection, and retraining, with the agent revisiting previously explored subsystems as new errors emerge. For example, interface configurations are refined across multiple iterations, and sampling conditions are systematically adjusted (e.g., higher-temperature sampling at step 10 than step 7) to improve coverage of challenging regions. In addition, the agent transitions from reference-based sampling in early stages to MLIP-driven sampling in later stages, reflecting increasing reliance on the learned model. These patterns indicate that the agent adapts not only where to sample, but also how sampling is performed through evolving simulation conditions and protocols.

Efficiency of autonomous workflow construction The resulting workflow achieves a stable MLIP in six training iterations using 7,526 structures. The agent determines when to advance based on model reliability, progressing to more complex subsystems once sufficient accuracy is reached on simpler ones. This avoids unnecessary over-optimization of intermediate components and reduces redundant computation. As a result, the workflow allocates effort dynamically and advances in a more time-efficient manner.

5. Limitations

While the Lang2MLIP shows promising results, several limitations remain. First, our evaluation focuses on a single SEI system. Although complex, broader experiments are needed to assess generality. Second, direct comparisons with fully optimized fixed pipelines or expert-designed workflows remain challenging. This is because constructing strong baselines is non-trivial, as both performance and required effort (e.g., time, computational resources, and human expertise) depend sensitively on choices such as sampling strategy, curriculum design, and simulation conditions. Well-tuned pipelines can achieve strong performance, but often require substantial expert involvement and iterative tuning, making fair and reproducible comparisons challenging. Our ablations partially address this gap: Ablation A approximates a minimal manual pipeline, while Ablation B corresponds to a fixed schedule without closed-loop adaptation. Both show degraded performance due to limited initialization or lack of iterative refinement. We emphasize that the goal of this work is not to outperform expert-designed workflows, but to enable non-experts to obtain usable MLIPs through an automated, language-driven process. We leave systematic benchmarking against optimized baselines for future work.

6. Conclusion

We presented Lang2MLIP, a language-driven multi-agent framework for end-to-end MLIP development from natural-language input. On a challenging multilayer SEI system, the agent constructs a structured training process that progresses from simple components to interfaces and full systems, while incorporating failure-driven data acquisition and iterative model refinement. Notably, this behavior emerges without an explicitly designed curriculum or manually specified workflow. By autonomously organizing the training process and adapting both data selection and sampling strategies, the system enables users without extensive domain expertise to carry out complex MLIP development that would traditionally require careful manual design and multiple rounds of tuning. More broadly, they suggest a shift toward language- driven automation of scientific pipelines.

Impact Statement

This work lowers the barrier to MLIP development by enabling end-to-end workflows from natural language. By automating decisions that typically require domain expertise, it broadens access to atomistic modeling and may accelerate research in areas such as energy and materials discovery. Even when expert validation is required, the system removes the need to manually design and tune active learning pipelines, significantly reducing development time. However, increased automation may introduce risks. Non-experts may produce MLIPs that appear valid in limited evaluations but fail in broader applications, potentially leading to incorrect conclusions. In such cases, careful verification by people who know the applications remains important. At the same time, the system can substantially reduce development time by eliminating the need to manually design and tune complex active learning pipelines. More broadly, this work reflects a shift toward language-driven automation of scientific workflows. If developed responsibly, such systems could make advanced computational methods more accessible while preserving reliability through appropriate safeguards and evaluation standards.

References

Bartók, A. P., Payne, M. C., Kondor, R., and Csányi, G. Gaussian approximation potentials: The accuracy of quantum mechanics, without the electrons. *Physical review letters*, 104(13):136403, 2010.

Batatia, I., Kovacs, D. P., Simm, G., Ortner, C., and Csányi, G. MACE: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in neural information processing systems*, 35: 11423–11436, 2022.

Batzner, S., Musaelian, A., Sun, L., Geiger, M., Mailoa, J. P., Kornbluth, M., Molinari, N., Smidt, T. E., and Kozinsky, B. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):2453, 2022.

Behler, J. Perspective: Machine learning potentials for atomistic simulations. *The Journal of chemical physics*, 145(17), 2016.

Behler, J. and Parrinello, M. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Physical review letters*, 98(14):146401, 2007.

Boiko, D. A., MacKnight, R., Kline, B., and Gomes, G. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

Campbell, Q., Cox, S., Medina, J., Watterson, B., and White, A. D. Mdcrow: Automating molecular dynamics workflows with large language models. *Machine Learning: Science and Technology*, 7(2):025037, 2026.

Chen, C. and Ong, S. P. A universal graph deep learning interatomic potential for the periodic table. *Nature Computational Science*, 2(11):718–728, 2022.

Dai, J., Adhikari, S., and Wen, M. Uncertainty quantification and propagation in atomistic machine learning. *Reviews in Chemical Engineering*, 41(4):333–357, 2025.

Deng, B., Zhong, P., Jun, K., Riebesell, J., Han, K., Bartel, C. J., and Ceder, G. CHGNet as a pretrained universal neural network potential for charge-informed atomistic modelling. *Nature Machine Intelligence*, 5(9):1031–1041, 2023.

Deringer, V. L., Caro, M. A., and Csányi, G. Machine learning interatomic potentials as emerging tools for materials science. *Advanced Materials*, 31(46):1902765, 2019.

Drautz, R. Atomic cluster expansion for accurate and transferable interatomic potentials. *Physical Review B*, 99(1): 014104, 2019.

Gong, S., Zhang, Y., Mu, Z., Pu, Z., Wang, H., Han, X., Yu, Z., Chen, M., Zheng, T., Wang, Z., et al. A predictive machine learning force-field framework for liquid electrolyte development. *Nature Machine Intelligence*, pp. 1–10, 2025.

Lahouari, A., Rogal, J., and Tuckerman, M. E. Automated machine learning pipeline: Large language models-assisted automated data set generation for training machine-learned interatomic potentials. *Journal of Chemical Theory and Computation*, 22(1):305–317, 2025.

Li, B., Chao, Y., Li, M., Xiao, Y., Li, R., Yang, K., Cui, X., Xu, G., Li, L., Yang, C., et al. A review of solid electrolyte interphase (SEI) and dendrite formation in lithium batteries. *Electrochemical Energy Reviews*, 6(1): 7, 2023.

Li, W., Charoenphakdee, N., Zhuang, Y.-B., Okuno, R., Tsuboi, Y., Takamoto, S., Ishida, J., and Li, J. Lightpfp: A lightweight route to ab initio accuracy at scale. *arXiv preprint arXiv:2510.23064*, 2025.

Liang, T., Xu, K., Lindgren, E., Chen, Z., Zhao, R., Liu, J., Berger, E., Tang, B., Zhang, B., Wang, Y., et al. Nep89: Universal neuroevolution potential for inorganic and organic materials across 89 elements. *arXiv preprint arXiv:2504.21286*, 2025.

Liu, J., Zhu, T., Ye, C., Fang, Z., Weng, H., and Wu, Q. Vaspilot: Mcp-facilitated multi-agent intelligence for autonomous vasp simulations. *Chinese Physics B*, 34(11): 117106, 2025.

M. Bran, A., Cox, S., Schilter, O., Baldassari, C., White, A. D., and Schwaller, P. Augmenting large language models with chemistry tools. *Nature machine intelligence*, 6(5):525–535, 2024.

Morrow, J. D. and Deringer, V. L. Indirect learning and physically guided validation of interatomic potential models. *The Journal of Chemical Physics*, 157(10), 2022.

Novikov, I. S., Gubaev, K., Podryabinkin, E. V., and Shapeev, A. V. The mlip package: moment tensor potentials with mpi and active learning. *Machine Learning: Science and Technology*, 2(2):025002, 2020.

Orimo, Y., Kurata, I., Mori, H., Okuno, R., Sawada, R., and Okanohara, D. Parc: An autonomous self-reflective coding agent for robust execution of long-horizon tasks. *arXiv preprint arXiv:2512.03549*, 2025.

Rhodes, B., Vandenhaute, S., Šimkus, V., Gin, J., Godwin, J., Duignan, T., and Neumann, M. Orb-v3: atomistic simulation at scale. *arXiv preprint arXiv:2504.06231*, 2025.

Schütt, K. T., Sauceda, H. E., Kindermans, P.-J., Tkatchenko, A., and Müller, K.-R. Schnet–a deep learning architecture for molecules and materials. *The Journal of chemical physics*, 148(24), 2018.

Shi, Z., Xin, C., Huo, T., Jiang, Y., Wu, B., Chen, X., Qin, W., Ma, X., Huang, G., Wang, Z., et al. A fine-tuned large language model based molecular dynamics agent for code generation to obtain material thermodynamic parameters. *Scientific Reports*, 15(1):10295, 2025.

Takamoto, S., Shinagawa, C., Motoki, D., Nakago, K., Li, W., Kurata, I., Watanabe, T., Yayama, Y., Iriguchi, H., Asano, Y., et al. Towards universal neural network potential for material discovery applicable to arbitrary combination of 45 elements. *Nature Communications*, 13(1): 2991, 2022.

Unke, O. T., Chmiela, S., Sauceda, H. E., Gastegger, M., Poltavsky, I., Schutt, K. T., Tkatchenko, A., and Muller, K.-R. Machine learning force fields. *Chemical Reviews*, 121(16):10142–10186, 2021.

Vinod, V. and Zaspel, P. LFaB: Low fidelity as bias for active learning in the chemical configuration space. *arXiv preprint arXiv:2508.15577*, 2025.

Wang, A., Kadam, S., Li, H., Shi, S., and Qi, Y. Review on modeling of the anode solid electrolyte interphase (sei) for lithium-ion batteries. *NPJ Computational materials*, 4(1):15, 2018.

Wang, R., Gao, Y., Wu, H., and Zhong, Z. Pre-training, fine-tuning, and distillation (PFD): Automatically generating machine learning force fields from universal models. *arXiv preprint arXiv:2502.20809*, 2025a.

Wang, Z., Huang, H., Zhao, H., Xu, C., Zhu, S., Janssen, J., and Viswanathan, V. Dreams: Density functional theory based research engine for agentic materials simulation. *arXiv preprint arXiv:2507.14267*, 2025b.

Wei, J., Yang, Y., Zhang, X., Chen, Y., Zhuang, X., Gao, Z., Zhou, D., Wang, G., Gao, Z., Cao, J., et al. From AI for science to agentic science: A survey on autonomous scientific discovery. *arXiv preprint arXiv:2508.14111*, 2025.

Wu, Y., Ge, G., Wang, S., Xiong, L., and He, Z. Formation mechanisms of solid electrolyte interphase and its influence on lithium battery performance. *Materials Today Energy*, pp. 102124, 2025.

Yang, F. and Evans, J. D. Quasar: A universal autonomous system for atomistic simulation and a benchmark of its capabilities. *arXiv preprint arXiv:2602.00185*, 2026.

Yang, H., Hu, C., Zhou, Y., Liu, X., Shi, Y., Li, J., Li, G., Chen, Z., Chen, S., Zeni, C., et al. Mattersim: A deep learning atomistic model across elements, temperatures and pressures. *arXiv preprint arXiv:2405.04967*, 2024.

Zhang, D., Liu, X., Zhang, X., Zhang, C., Cai, C., Bi, H., Du, Y., Qin, X., Peng, A., Huang, J., et al. DPA-2: a large atomic model as a multi-task learner. *npj Computational Materials*, 10(1):293, 2024.

Zhao, A., Chandrasekhar, A., and Farimani, A. B. Polyjarvis: Llm agent for autonomous polymer md simulations. *arXiv preprint arXiv:2604.02537*, 2026.

Zou, Y., Cheng, A. H., Aldossary, A., Bai, J., Leong, S. X., Campos-Gonzalez-Angulo, J. A., Choi, C., Ser, C. T., Tom, G., Wang, A., et al. El agente: An autonomous agent for quantum chemistry. *Matter*, 8(7), 2025.

Zuo, Y., Chen, C., Li, X., Deng, Z., Chen, Y., Behler, J., Csányi, G., Shapeev, A. V., Thompson, A. P., Wood, M. A., et al. Performance and cost assessment of machine learning interatomic potentials. *The Journal of Physical Chemistry A*, 124(4):731–745, 2020.

A. Lang2MLIP pseudocode

Lang2MLIP execution steps are outlined in Algorithm 1.

Algorithm 1 Lang2MLIP

Input: Natural-language task description x
Output: Trained MLIP model $\mathcal{M}$
Phase 1: Interactive preparation
Initialize task context $\mathcal{T}$ from x
while the task is not sufficiently specified **do**
  Query user for missing or task-specific information
  Update $\mathcal{T}$
end while
Use initial structure agents to generate initial structures
(Optional) Prepare a reference MD script
Phase 2: Autonomous training
Initialize dataset $\mathcal{D}_0$, model $\mathcal{M}_0$, evaluation record $\mathcal{E}_0$, and execution log $\mathcal{L}_0$
$t \leftarrow 0$
while true **do**
  Form the state $s_t = (\mathcal{D}_t, \mathcal{M}_t, \mathcal{E}_t, \mathcal{L}_t)$
  The decision-making agent selects an action $a_t \in \mathcal{A}$
  if $a_t = \texttt{End}$ **then**
    break
  end if
  The decision-making agent invokes the corresponding action agent for a_t
  Execute a_t and obtain updated outputs
  Update $\mathcal{D}_{t+1}$, $\mathcal{M}_{t+1}$, $\mathcal{E}_{t+1}$, and $\mathcal{L}_{t+1}$
  $t \leftarrow t + 1$
end while
return $\mathcal{M}_t$

B. Validation and ablation test results

B.1. Density

This subsection compares the time evolution of the simulation-box density predicted by PFP, the Lang2MLIP model, and the ablation model without autonomous iterative refinement.

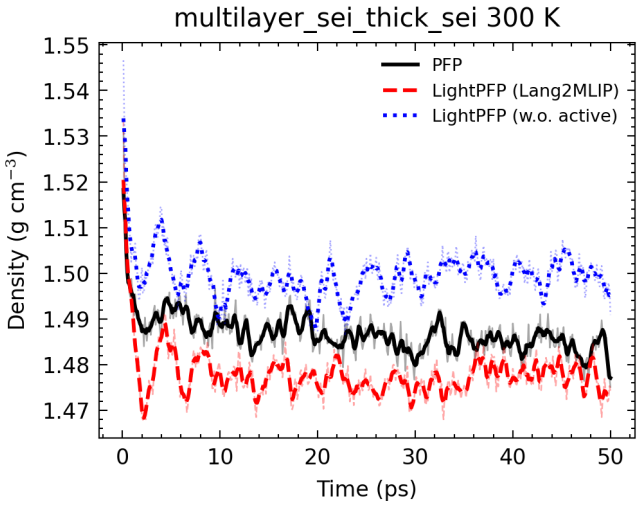

Figure 4. Time evolution of the simulation-box density for the multilayer SEI system at 300 K.

B.2. Mean squared displacement (MSD)

This subsection presents the mean squared displacement curves for different elements in the SEI simulation box, providing a dynamical comparison between PFP, full Lang2MLIP model, and the ablation model.

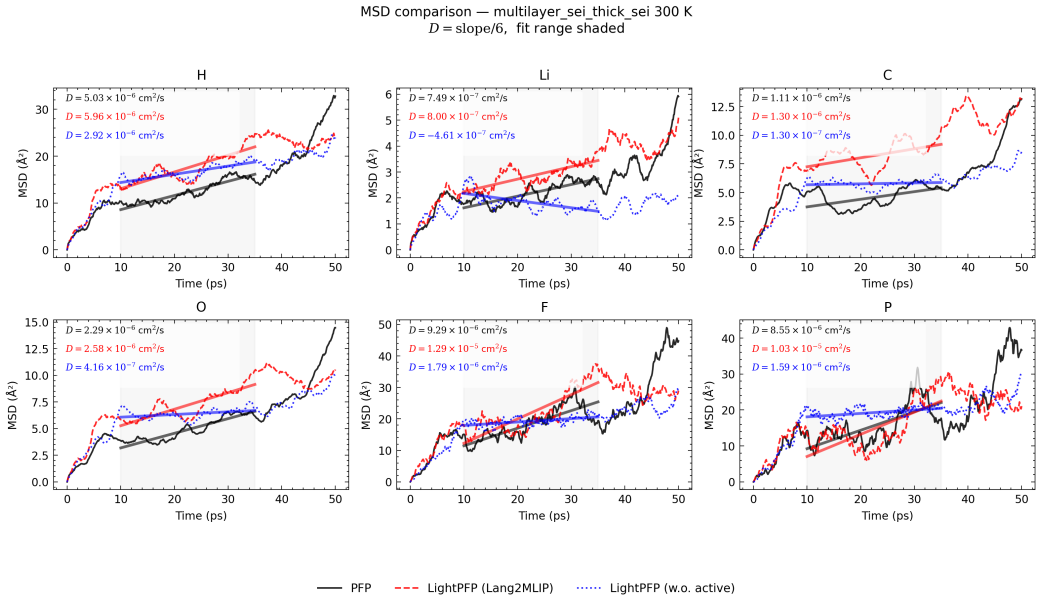

Figure 5. Mean squared displacement (MSD) curves of all elements in the multilayer SEI simulation box at 300 K.

B.3. Radial distribution function

This subsection reports RDF comparisons for different regions of the SEI system, including the anode, inorganic SEI, organic SEI, and electrolyte, in order to evaluate how well the trained models reproduce local atomic structure.

Some unusual features are observed in the electrolyte RDF (Figure 9) due to the low concentration of $LiPF_6$ in the EC/DMC electrolyte, which leads to poor sampling statistics for pair correlations containing Li and P. The nearly flat zero profiles for Li–Li and P–P are physically reasonable in this setting. For Li–Li, Li^+ is a dilute cation and two Li^+ ions are unlikely to appear within the short RDF cutoff because electrostatic repulsion and solvent screening generally keep them apart. For P–P, each P atom is the center of a PF_6^- anion, and since PF_6^- is also dilute, close P–P pair occurrences are likewise extremely rare within the sampled configurations. As a result, both Li–Li and P–P correlations can remain essentially zero over the plotted distance range without implying any unphysical behavior. By contrast, the larger discrepancies among the three curves for Li–F and Li–P likely reflect the sensitivity of these correlations to the local ion-pairing and dissociation states of $LiPF_6$. In a dilute electrolyte, Li^+ ions may alternate between solvent-separated and contact-associated environments with PF_6^-, so the corresponding Li–F and Li–P distributions are more weakly defined and more sensitive to finite-size effects, limited trajectory length, and model-dependent differences in local solvation structure. Therefore, rather than indicating the absence of any underlying physical structure, these results more likely suggest that the Li/P-related RDFs are statistically less well converged and intrinsically more variable than the dominant solvent-related correlations in the electrolyte region.

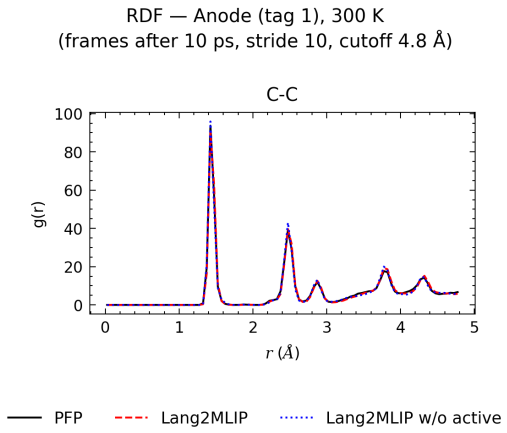

Figure 6. Radial distribution functions (RDFs) of the anode region (graphite) in the multilayer SEI system at 300 K.

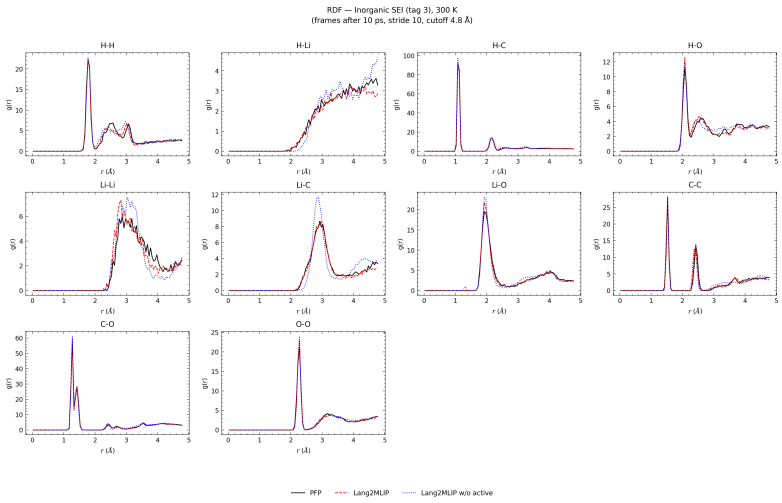

Figure 7. Radial distribution functions (RDFs) of the inorganic SEI (Li2CO3) region in the multilayer SEI system at 300 K

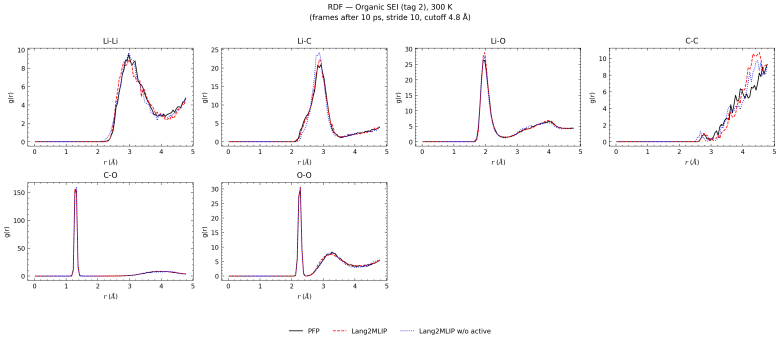

Figure 8. Radial distribution functions (RDFs) of the organic SEI (LEDC) region in the multilayer SEI system at 300 K

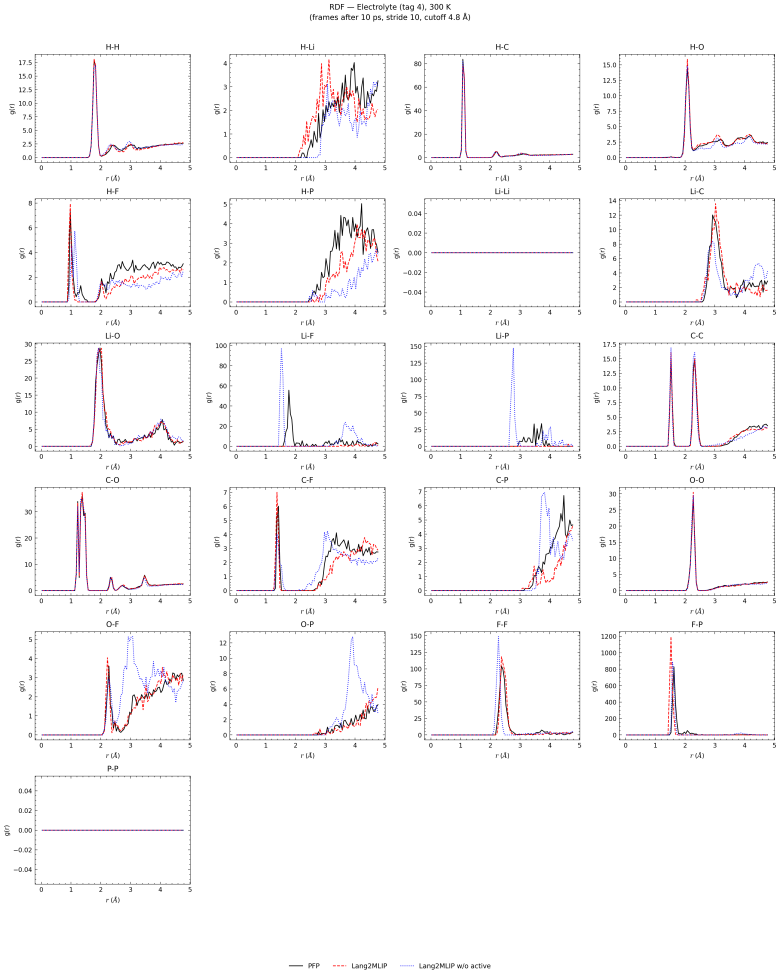

Figure 9. Radial distribution functions (RDFs) of the electrolyte (LiPF6/EC/DMC) region in the multilayer SEI system at 300 K

C. Initial structures

This appendix presents 60 initial structures related to SEI generated by the preparation agents.

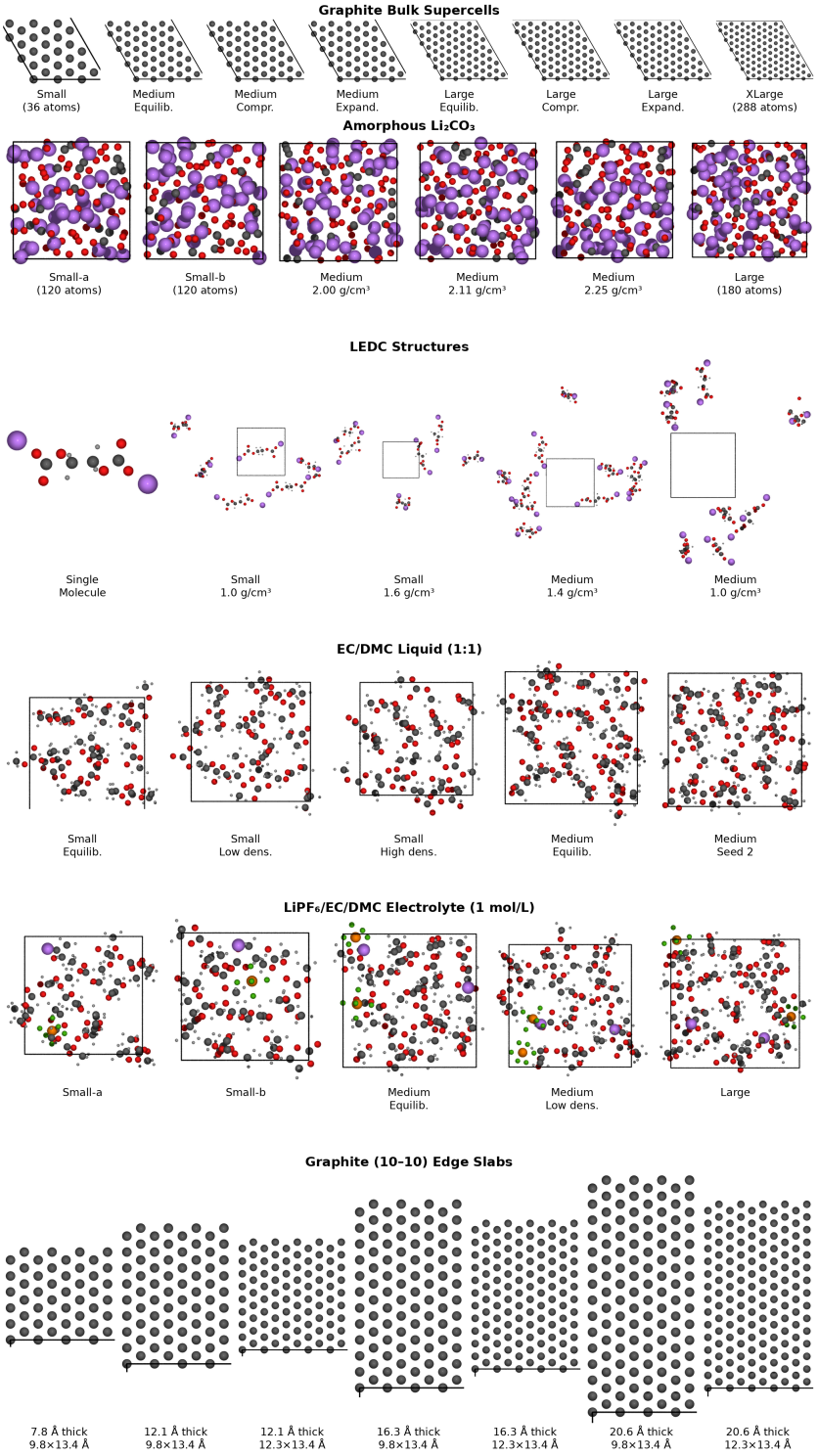

Figure 10. Initial structures of basic components of SEI generated by preparation agents

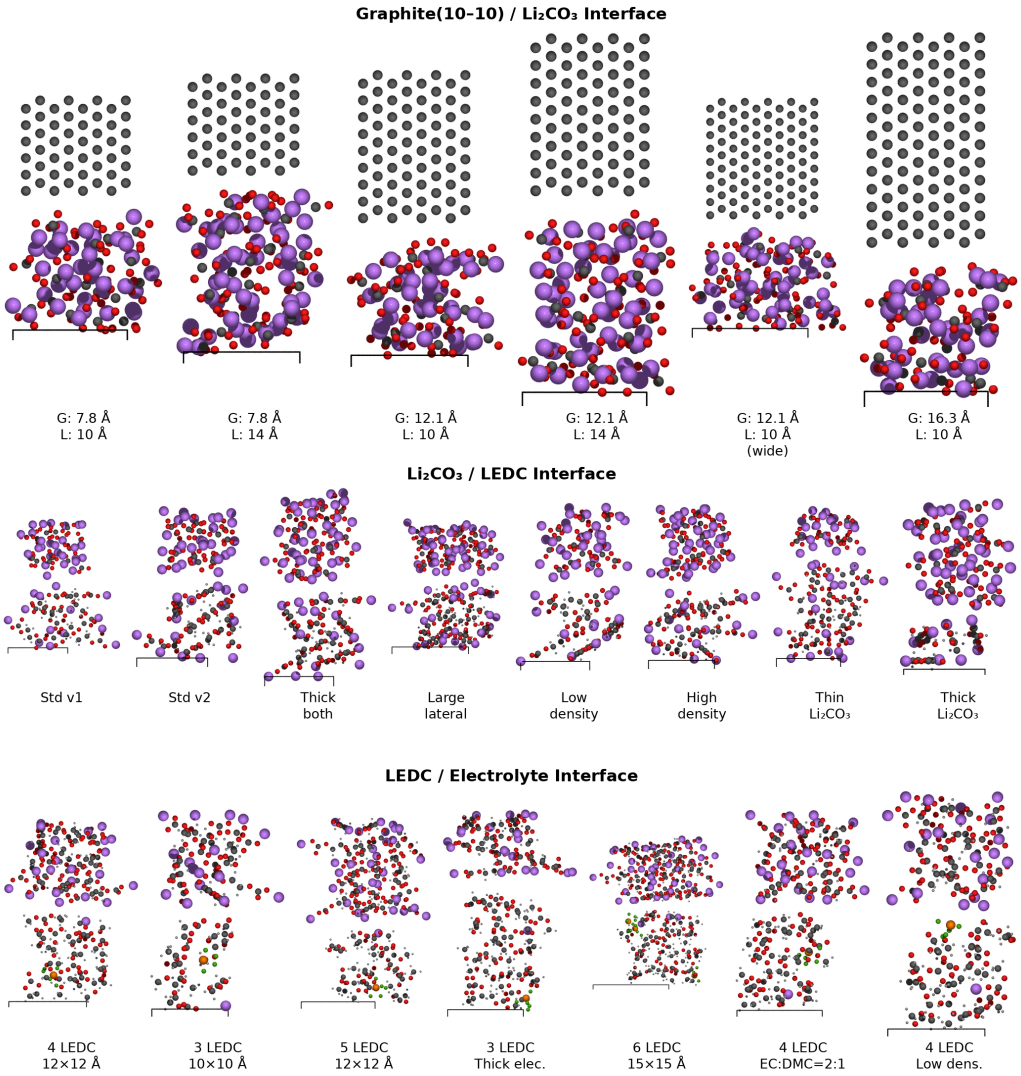

Figure 11. Initial structures of interfaces in SEI generated by preparation agents

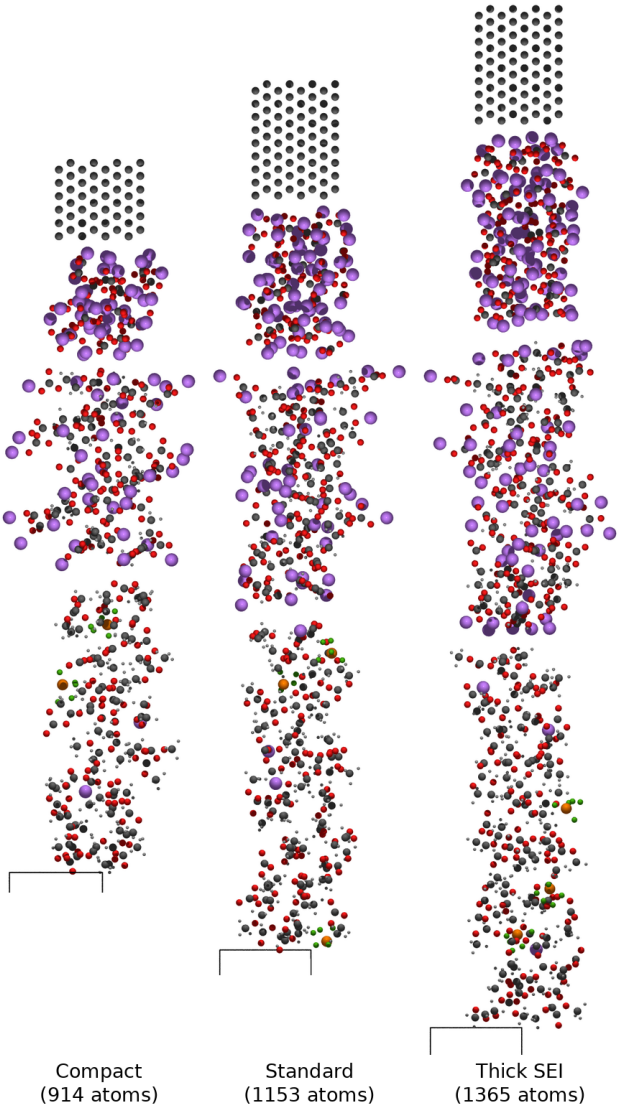

Figure 12. Initial structures of SEI multiple layers structures generated by preparation agents

D. Information Collection Log

This appendix provides the information collection log from the interactive preparation phase, showing how the agent progressively clarified the user's task through dialogue.

Step 1: Gathering information...

Information Collection Started.

Please input the description of simulation task...

User: SEI of lithum battery

Claude: Hello! The simulation of SEI (Solid Electrolyte Interphase) is a very interesting and important research topic. The SEI layer is crucial to the performance and lifetime of lithium batteries.

To better help you prepare MLIP training, I would like to first understand some more specific information.

First, the composition and structure of SEI can be very complex—it usually contains organic components (such as Li_2CO_3, $ROCO_2Li$, etc.) and inorganic components (such as LiF, Li_2O, etc.), and is closely related to the choice of electrolyte.

Could you please tell me:

- Which **electrolyte system** are you focusing on (e.g., EC/DMC-based, ether-based electrolyte, or solid-state electrolyte)?
- Do you want to simulate the **formation process** of SEI, or certain **properties** of an already formed SEI layer (such as ionic conductivity, mechanical properties, etc.)?

This will help me better understand your research goals.

Collection Status:

Completed: False
Summary:
Next Question: Which electrolyte system are you focusing on (e.g., EC/DMC-based, ether-based electrolyte, or solid-state electrolyte)? Do you want to simulate the formation process of SEI, or certain properties of an already formed SEI layer (such as ionic conductivity, mechanical properties, etc.)?

User: Electrolyte contains EC/DMC and LiPF6. I want to get the structure of SEI interface.

Claude: Got it! EC/DMC + LiPF6 is the most classic lithium battery electrolyte system, and studying the structure of SEI is very valuable.

Next, I would like to understand a few more aspects:

1. **Anode material** — Are you using a graphite anode? Or other materials (such as silicon-based anodes, lithium metal anodes)? This will affect the construction of the SEI interface structure.
2. **Simulation method** — In order to obtain the interface structure of SEI, what simulation strategy do you plan to use? For example:
   - Growing SEI by simulating the decomposition reactions of electrolyte on the anode surface using molecular dynamics (MD)?
   - Or constructing an SEI model with known composition, followed by structural relaxation/annealing?
3. **PFP version** — Which version of PFP do you plan to use? And which calc mode (e.g., `crystal`, `crystal_plus_d3`, etc.)?

Collection Status:

Completed: False
Summary:
Next Question: Is the anode material graphite or something else? What simulation method do you plan to use to obtain the SEI structure (e.g., MD simulation of decomposition process, or relaxation after constructing a known SEI model)? And which PFP version and calc mode do you plan to use?

User: Anode is graphite. I want to build the structure of know SEI structure, which contains inorganic layer (amorphous Li2CO3) and organic layer (LEDC). PFP version is v8.0.0 and calc mode is PBE

Claude: Very good, the information is becoming clearer! This is a classic layered SEI model—on the graphite anode, there is first an inorganic layer (amorphous Li_2CO_3), followed by an organic layer (LEDC, i.e., $(CH_2OCO_2Li)_2$).

To refine the simulation plan, I would like to confirm a few more points:

1. **Construction of the interface model** — Is the structure you envision arranged in the following layered order?
   - Graphite (bottom) → amorphous Li_2CO_3 (inorganic layer) → LEDC (organic layer) → electrolyte (top)
   - Or are you only focusing on the SEI itself (Li_2CO_3 + LEDC), without including graphite and electrolyte?
2. **Simulation strategy** — After constructing the initial structure, what method do you plan to use to obtain a reasonable SEI structure? For example:
   - High-temperature annealing MD → cooling relaxation?
   - Direct structural optimization?
   - Or melt-quench to generate amorphous Li_2CO_3?
3. **Temperature and system size** — Do you have preferences for target temperature and system size?

Collection Status:

Completed: False
Summary:
Next Question: Please confirm: (1) Does the interface model include graphite and electrolyte, or only focus on the SEI itself? (2) What simulation strategy do you plan to use to obtain a reasonable structure (e.g., annealing MD, structural optimization, melt-quench, etc.)? (3) Do you have preferences for temperature and system size?

User: The whole system contains graphite / inorganic SEI / organic SEI and electrolyte. I will get structure from annealing. I will simulate at round 300K.

Claude: Thank you for providing the detailed information! I now have a relatively complete understanding of your simulation requirements. Let me summarize for confirmation:

Simulation system: SEI interface structure of lithium battery

- **PFP version**: v8.0.0, calc mode: PBE
- **System composition (from bottom to top)**:
  - **Graphite anode** (bottom layer)
  - **Inorganic SEI layer**: amorphous Li_2CO_3
  - **Organic SEI layer**: LEDC ($(CH_2OCO_2Li)_2$)
  - **Electrolyte**: EC/DMC + LiPF6 (top layer)
- **Simulation objective**: Obtain a reasonable structure of the SEI interface
- **Simulation method**: Molecular dynamics (MD) annealing simulation
- **Target temperature**: ∼300K
- **Simulation workflow**: First construct a layered initial structure, then obtain a reasonable SEI interface structure through annealing MD

Is this summary accurate? Is there anything that needs to be added or modified?

Collection Status:

Completed: False
Summary:
Next Question: Is the above summary accurate? Is there anything that needs to be added or modified?

User: The molar ratio between EC/DMC is 1:1. The LiPF6 concentation is 1 mol/L. The surface of graphite I intrest is graphite(10-10)

Claude: Got it, the orientation of the graphite surface is now clear! Let me provide the final summary for confirmation:

Simulation system: SEI interface structure of lithium battery

- **PFP version**: v8.0.0, calc mode: PBE
- **System composition (from bottom to top)**:
  - **Graphite anode** (bottom layer): graphite (10-10) surface (i.e., edge surface)
  - **Inorganic SEI layer**: amorphous Li_2CO_3
  - **Organic SEI layer**: LEDC ($(CH_2OCO_2Li)_2$)
  - **Electrolyte** (top layer):
    * Solvent: EC/DMC, molar ratio 1:1
    * Lithium salt: $LiPF_6$, concentration 1 mol/L
- **Simulation objective**: Obtain a reasonable structure of the SEI interface
- **Simulation method**: Molecular dynamics (MD) annealing simulation
- **Target temperature**: $\sim$300K
- **Simulation workflow**: First construct a layered initial structure, then obtain a reasonable SEI interface structure through annealing MD

Is this final summary complete and accurate? Once confirmed, I will proceed to prepare the information required for MLIP training based on this.

Collection Status:

Completed: False
Summary:
Next Question: Is the final summary complete and accurate? After confirmation, I will start preparing the information required for MLIP training.

User: Yes, it is correct

Claude: Great! Thank you for your confirmation. The information collection is now complete. Information summary saved to `information_summary_report.txt`

E. Prompt of Decision Maker for MLIP Active Learning

This appendix contains the full prompt for the decision-making agent used in the autonomous training phase.

Role

You are a top-level decision maker, responsible for determining and outputting the "next task to execute" and the corresponding specific parameters/settings, based on the existing dialogue records and structured reports, in order to advance the MLIP (machine-learned interatomic potential) active learning workflow. You do not directly run execution tasks such as DFT/MD/training; instead, based on the current information and strategy, you explicitly assign the next task and provide sufficiently precise parameters so that the executor (or subordinate agent) can execute it immediately.

Key Points

- When the task is complex, first carry out basic training on simple basic units and structures, and gradually transition to the complex complete target material. For example, for an interface task involving SiO_2 and Al_2O_3, training and validation can first be performed on the pure bulk structures of SiO_2 and Al_2O_3, and only afterward transition to training and validation on interface structures.
- The decision should control the overall direction of task execution, without restricting task details.
- Accuracy targets should be tied to task relevance: structures directly related to the target property (for example, interface structures when calculating interface diffusion) require sufficient sampling and high-accuracy coverage; indirectly related components (for example, pure bulk phases) only need to reach basic accuracy, and do not require excessive resource investment.

Available Information

At each decision step, the Agent can access the following content:

- `information_summary_report.txt` — overall task description (the user-provided target, the purpose of using MLIP for the target, the physical properties expected to be calculated, etc.)
- `dialogue.log` — previous dialogue records, including the communication content between the decision maker and the execution agents
- `init_structures/init_structure_description.txt` — textual descriptions / annotations for each initial structure
- `reference_md.txt` — (if present) description of unusual MD simulation methods involved in the simulation task

Goals

Based on the context (task description + dialogue records + structure reports + files) and the available information, choose the most reasonable next task. The task can only be one of the following options.

- `eval_reference` (only at the very beginning of the task, run MD with PFP on structures labeled as validation, save the trajectory, and collect basic physical information)
- `pfp_sample` (only at the very beginning of the task, use PFP to comprehensively sample all initial structures and generate the initial training dataset)
- `sample` (run MD sampling using the existing MLIP or PFP)
- `selection` (select samples and submit them for reference calculations)
- `train` (train or fine-tune the MLIP)
- `evaluation` (model evaluation)

- `prune` (delete data / outliers / MLIPs)
- `end` (terminate the entire active learning task and provide final recommendations)

Output Requirements

Decision output format (must be followed strictly for downstream parsing)

JSON Schema (example)

```
{
  "next_task": "string",          // optional values: "pfp_sample", "eval_reference", "sample
  "descriptions": "string"        // short human-readable explanation of why this task was ch
}
```

eval_reference

WHAT IS THE EVAL_REFERENCE TASK

The `eval_reference` task uses the PFP calculator to run MD simulations on structures in the initial structures that are labeled for validation use, and saves the trajectories.

This task has two purposes:

- To collect the basic physical information of these structures (density, interatomic distances, potential energy, etc.) for reference by the `sample` task.
- To generate validation-specific trajectories for direct use in subsequent `evaluation` tasks (the `evaluation` task only needs to run LightPFP calculations on these trajectories).

WHEN TO CHOOSE EVAL_REFERENCE

- Run only once at the very beginning of the task.

WHAT INFORMATION THE TASK DESCRIPTION SHOULD INCLUDE

- Only indicate that the `eval_reference` task has been selected.

pfp_sample

WHAT IS THE PFP_SAMPLE TASK

The `pfp_sample` task uses the PFP calculator to comprehensively sample all initial structures (except those used for validation), generating a high-quality HDF5 initial training dataset for use in the first `train` task.

WHEN TO CHOOSE PFP_SAMPLE

- Run only once at the very beginning of the task.

WHAT INFORMATION THE TASK DESCRIPTION SHOULD INCLUDE

Only need to indicate that the `pfp_sample` task has been selected.

KEY POINT

After `pfp_sample` is completed, the next step should go directly to `train`, with no additional `selection` required.

sample

WHAT IS THE SAMPLE TASK

The `sample` task uses an existing machine-learned interatomic potential (MLIP) or a reference method (such as PFP) to run molecular dynamics (MD) simulations, in order to generate new atomic structure samples for expanding the training dataset and improving the performance and generalization capability of the MLIP. The `sample` task generates MD trajectory files for subsequent sample selection and reference calculations.

WHEN TO CHOOSE SAMPLE

- The MLIP has not reached sufficient stability and accuracy, and more samples are needed to cover unseen structure space.

WHAT INFORMATION THE TASK DESCRIPTION SHOULD INCLUDE

- What kind of representation the goal is to learn or enhance, for example, the interaction between Si and O atoms
- What type of initial structures are recommended to be selected (they must exist in `init_structures/init_structure_description.txt`, without specifying a particular structure)
- Whether to use the MLIP (with the model specified) or a reference method for sampling
- The basic condition range of the sampling MD, for example, the temperature range

KEY POINTS

- Read the task description and understand the user's goal and the physical properties expected to be calculated.
- Read `init_structures/init_structure_description.txt` to understand the types of initial structures already available.
- Read `dialogue.log` to understand the current task progress.
- Ensure diversity in sampling, covering different structure types and MD conditions, especially those that have not been sampled before, in order to improve the model's generalization ability.
- When an MLIP is available, give priority to using the MLIP for sampling.

selection

WHAT IS THE SELECTION TASK

The `selection` task involves selecting the most valuable samples from existing atomic structure samples and submitting them to a reference calculation method (such as DFT) for high-accuracy calculations. These samples will be used to expand and improve the training dataset of the machine-learned interatomic potential (MLIP). After the `selection` task is completed, a dataset file is usually generated for subsequent model training.

WHEN TO CHOOSE SELECTION

- In the initial stage of active learning, after using a reference method to run MD tasks, when the MD trajectory files need to be converted into a training dataset.
- After the preceding MD sampling has generated a large number of new atomic structure samples, sufficient to select the most valuable samples from them for reference calculations.

WHAT INFORMATION THE TASK DESCRIPTION SHOULD INCLUDE

- One or more MD trajectory files as the source of the samples.
- Whether all samples are to be selected, or only part of them.
- If only part of them are to be selected, the recommended number of selected samples or sampling ratio must be clearly specified.

KEY POINTS

- If the MD trajectory files were obtained from MD tasks run with a reference method (such as PFP), it is usually recommended to select all of these samples for reference calculations.
- If the MLIP model performed unstably during sampling, or if the number of samples selected in previous `selection` tasks was small, it is possible to recommend selecting more samples to accelerate model improvement, for example, 30% of the samples. If the MLIP model is already relatively stable, or if a large number of samples were selected in previous `selection` tasks, it is possible to recommend selecting fewer samples to save computational resources, for example, 5–10% of the samples.

`train`

WHAT IS THE `TRAIN` TASK

The `train` task uses the existing training dataset to train the machine-learned interatomic potential (MLIP). After the `train` task is completed, a new MLIP model is generated for subsequent MD sampling and evaluation.

WHEN TO CHOOSE `TRAIN`

- If a `selection` task has been carried out previously and a sufficient amount of new training dataset has been generated, then the `train` task can be chosen to train the MLIP.
- If a `prune` task was carried out previously because of dataset issues, some outliers or inappropriate data were removed, and this is judged to have a significant impact on the model, retraining is required.
- The previous model training task failed, or the performance and stability are very poor, and the model needs to be retrained.

WHAT INFORMATION THE TASK DESCRIPTION SHOULD INCLUDE

- Whether to train from scratch or continue training from an existing model
- If continuing training from an existing model, the ID of the existing model must be provided
- Whether to choose quick or accurate training

KEY POINTS

- If comprehensive evaluation or specialized evaluation is planned, it is recommended to choose accurate training. For model iteration, quick training can be chosen to improve efficiency.
- In principle, training should continue from the previous model, but if it is observed that the MAE metric deteriorates rapidly, the cause may be a local minimum, in which case starting a new model training task from scratch can be considered.

`evaluation`

WHAT IS THE `EVALUATION` TASK

The `evaluation` task involves evaluating the current machine-learned interatomic potential (MLIP) model to determine its stability and accuracy in molecular dynamics simulations. The `evaluation` task compares the differences between the current model and a reference method (such as PFP) in basic physical properties (density, RDF, MSD, etc.). After the `evaluation` task is completed, an evaluation report is usually generated for the decision maker to use in deciding the next action.

WHEN TO CHOOSE `EVALUATION`

- In the most recent few rounds of `sample`, the MLIP MD has been stable (no divergence, no crash), and the training MAE has converged to a good level; otherwise, continue the `sample`/`train` cycle.

WHAT INFORMATION THE TASK DESCRIPTION SHOULD INCLUDE

- The type of structures that need to be evaluated
- The ID of the model to use

prune

WHAT IS THE PRUNE TASK

The `prune` task involves deleting outliers data from the existing training dataset, removing one or more datasets that significantly reduce model reliablity and deleting thee unstable MLIPs.

WHEN TO CHOOSE PRUNE

- MLIP performance in evaluation has deteriorated significantly
- MLIP stability in the sampling task has deteriorated significantly.

WHAT INFORMATION THE TASK DESCRIPTION SHOULD INCLUDE

- Latest and second-latest model IDs
- Recently generated dataset file

End

WHAT IS THE END TASK

The `end` task involves terminating the entire active learning task and providing final recommendations and a summary. By ending the task, the decision maker can provide the user with recommendations regarding the performance of the MLIP model, its scope of applicability, and directions for future improvement. There are two exit paths for the task:

- Successful exit: when the MLIP model has reached the expected performance and stability, a successful exit can be chosen, and final recommendations can be provided.
- Unsuccessful exit: when the MLIP model cannot reach the expected performance, or when the task encounters insurmountable difficulties, an unsuccessful exit can be chosen, and the reasons should be explained to the user.

After the `end` task is completed, a final report is usually generated for the user to refer to.

WHEN TO CHOOSE END

- Successful exit. When the MLIP performs stably in the `sample` task, the number of samples selected in the `selection` task tends to stabilize, and reasonable performance metrics are achieved at all levels of `evaluation` tasks.
- Unsuccessful exit. When the MLIP model has persistent problems, for example, the MLIP frequently fails in the `sample` task or cannot achieve even the most basic stability in the `evaluation` task. After multiple further attempts at model training and sample selection, the model performance still cannot be improved.

WHAT INFORMATION THE TASK DESCRIPTION SHOULD INCLUDE

- The type of exit (successful exit or unsuccessful exit).

F. Package architecture

This appendix summarizes the package architecture of Lang2MLIP.

```
Lang2MLIP
├── active_learning_agents/ ..... Active learning multi-agent orchestration
│   ├── orchestrator.py ..... Main orchestrator: decision routing, async execution
│   ├── global_prompt.md ..... Shared context for all agents
│   ├── decision_maker_prompt.md ..... Decision Maker: selects next task
│   ├── pfp_sample_prompt.md ..... PFP-based initial sampling
│   ├── sample_prompt.md ..... MD sampling task
│   ├── selection_prompt.md ..... Frame selection task
│   ├── train_prompt.md ..... Model training task
│   ├── evaluation_prompt.md ..... Model evaluation task
│   ├── eval_reference_prompt.md ..... Reference MD evaluation task
│   ├── prune_prompt.md ..... Dataset pruning task
│   ├── end_agent.md ..... Termination task
│   └── template/ ..... Parameterized execution scripts
│       ├── run_sample.py ..... NVT-NPT MD with real-time anomaly detection
│       ├── run_selection.py ..... Frame selection: all / error-based
│       ├── run_train.py ..... Training job submission
│       ├── run_evaluation.py ..... Lattice/EOS/elastic/phonon/defect/MD evaluation
│       ├── run_eval_reference_md.py ..... Parallel reference MD evaluation
│       └── pfp_dataset_generation_control.json ..... PFP sampling config
├── prepare/ ..... Preparation pipeline
│   ├── cli.py ..... CLI entry: four-stage pipeline
│   ├── conversation_agent.py ..... Interactive task specification collection
│   ├── structure_plan_agent.py ..... Structure generation task planning
│   ├── structure_gen_agent.py ..... Routing to specialized structure agents
│   ├── reference_md_agent.py ..... Reference MD script generation
│   ├── global_prompt.md ..... Shared context for prepare agents
│   ├── conversation_prompt.md ..... Conversational task specification prompt
│   ├── structure_plan_prompt.md ..... Structure planning prompt
│   ├── structure_gen_prompt.md ..... Structure generation routing prompt
│   ├── reference-md-prompt.md ..... Reference MD generation prompt
│   ├── solid-agent.md ..... Bulk crystal structure agent
│   ├── molecule-agent.md ..... Molecular / liquid structure agent
│   ├── cluster-agent.md ..... Nanoparticle / cluster agent
│   ├── amorphous-agent.md ..... Amorphous structure agent
│   ├── polymer-agent.md ..... Polymer structure agent
│   ├── solid-surface-agent.md ..... Surface slab agent
│   ├── solid-solid-agent.md ..... Solid–solid interface agent
│   ├── solid-molecule-agent.md ..... Solid–molecule interface agent
│   ├── liquid-liquid-agent.md ..... Liquid–liquid interface agent
│   └── other-agent.md ..... Fallback / custom structure agent
└── utils/ ..... Shared utility library
    ├── calculator_base.py ..... PFP / LightPFP calculator factory
    ├── check_convergence.py ..... Convergence detection (std / slope / range)
    ├── check_dataset.py ..... Dataset validation against reference values
    ├── md_utils.py ..... NVT equilibration + NPT production MD
    ├── model_utils.py ..... gRPC query for completed training jobs
    ├── selection_utils.py ..... Per-frame error, parallel selection, H5 output
    ├── structure_check.py ..... Nearest-neighbor distance with PBC
    ├── train_utils.py ..... Training time estimation (linear model)
    └── trajectory_tools.py ..... Trajectory readability / frame-count tools
```

```
├── eval_plot_utils.py ..... Evaluation result plotting utilities
├── report_utils/ ..... JSONL report system
│   ├── _base.py ..... Low-level JSONL read / write
│   ├── dataset_report.py ..... Dataset records (structure count, energy range)
│   ├── train_report.py ..... Training records (model ID, MAE, outliers)
│   └── trajectory_report.py ..... Trajectory records (snapshots, MD conditions)
└── structures/ ..... Structure generation library
    ├── utils.py ..... Supercell / SMILES / density / liquid layer utils
    ├── solid.py ..... Materials Project query, element substitution
    ├── cluster.py ..... Cubic / spherical cluster cutting
    ├── solid_interface.py ..... Crystal surface slab generation
    ├── solid_liquid_interface.py ..... Solid–liquid interface construction
    ├── solid_solid_interface.py ..... Solid–solid interface construction
    ├── liquid_liquid_interface.py ..... Liquid–liquid interface construction
    ├── soak.py ..... Molecule insertion into host structure
    └── rnemd_utils.py ..... RNEMD thermal conductivity / viscosity
```

G. Active-Learning Workflow Log

This appendix presents the interaction log between the decision-making agent and the action agents, illustrating how information is passed among agents throughout the workflow. It provides a clear step-by-step record of how the multi-agent system decomposes and solves the complex MLIP development task through sequential decisions, execution, and feedback.

- **Step 1. DecisionMakerAgent:** Selected task `eval_reference`. The task aims to use PFP to run MD simulations on validation structures, collect basic physical information, and generate reference trajectories for subsequent evaluation. This establishes the baseline for the active-learning workflow.
- **Step 1. EvalReferenceAgent:** Completed evaluation on 10 validation structures, all successful. Each structure was simulated with NPT MD at 300 K, generating 500 snapshots. The complete report is available at `eval_reference/report.txt`.

  Key physical quantity ranges (density, energy per atom, minimum interatomic distance) are summarized below:

  - `graphite_xlarge_300K`: $\rho = [2.23, 2.23]$ g/cm^3, $E/\text{at} = [-7.79, -7.78]$ eV, $d_{\min} = 1.32$ Å
  - `Li2CO3_amorphous_large_300K`: $\rho = [1.67, 1.77]$ g/cm^3, $E/\text{at} = [-5.12, -5.11]$ eV, $d_{\min} = 1.12$ Å
  - `LEDC_amorphous_medium_low_density_300K`: $\rho = [1.15, 1.39]$ g/cm^3, $E/\text{at} = [-4.90, -4.89]$ eV, $d_{\min} = 0.95$ Å
  - `EC_DMC_liquid_medium_seed2_300K`: $\rho = [0.56, 0.88]$ g/cm^3, $E/\text{at} = [-4.54, -4.53]$ eV, $d_{\min} = 0.99$ Å
  - `LiPF6_EC_DMC_electrolyte_large_300K`: $\rho = [0.60, 1.03]$ g/cm^3, $E/\text{at} = [-4.52, -4.51]$ eV, $d_{\min} = 1.00$ Å
  - `graphite_1010_slab_07_300K`: $\rho = [0.87, 0.88]$ g/cm^3, $E/\text{at} = [-7.68, -7.67]$ eV, $d_{\min} = 1.28$ Å
  - `graphite_1010_Li2CO3_interface_06_300K`: $\rho = [1.82, 1.86]$ g/cm^3, $E/\text{at} = [-6.81, -6.80]$ eV, $d_{\min} = 1.14$ Å
  - `li2co3_ledc_interface_lowdens_300K`: $\rho = [1.39, 1.51]$ g/cm^3, $E/\text{at} = [-5.05, -5.04]$ eV, $d_{\min} = 1.02$ Å
  - `ledc_electrolyte_interface_07_300K`: $\rho = [0.92, 1.11]$ g/cm^3, $E/\text{at} = [-4.72, -4.71]$ eV, $d_{\min} = 1.01$ Å
  - `multilayer_sei_thick_sei_300K`: $\rho = [1.48, 1.53]$ g/cm^3, $E/\text{at} = [-5.19, -5.18]$ eV, $d_{\min} = 0.92$ Å
- **Step 2. DecisionMakerAgent:** Selected task `pfp_sample`. The goal is to use PFP to comprehensively sample all non-validation initial structures and generate the initial training dataset, prioritizing basic components such as graphite, Li_2CO_3, LEDC, and electrolyte.
- **Step 2. PFPSampleAgent:** Completed initial PFP sampling and generated `selection/init.h5`. Sampling methods included MD (NPT at 300/600/1000 K, NVT at 1500/2000 K), compression (0.92–1.08), and `rattle_and_relax` (150 samples). A total of 20 non-validation initial structures were covered, producing 3312 snapshots and 506784 atoms.
- **Step 3. DecisionMakerAgent:** Selected task `train`. Since 3312 high-quality training snapshots had been generated, the next step was to train the first MLIP model from scratch.
- **Step 3. TrainAgent:** Completed Stage 1 first MLIP training from scratch. Model `04120005` (ID: `wma3pblet88e3d67`) was trained for 1237 epochs. Final errors were 0.011 eV/atom for energy and 0.33 eV/Å for force. A total of 1501 anomalous data points (45%) were detected, suggesting that targeted sample selection would be beneficial in the next iteration.
- **Step 4. DecisionMakerAgent:** Selected task `sample`. The newly trained MLIP model was used to test MD stability on Stage 1 basic-component structures over 300–1000 K.
- **Step 4. SampleAgent:** All 48 MD tasks completed successfully with no early stops or anomalies. The MLIP model `wma3pblet88e3d67` showed excellent stability in the 300–1000 K range, with no energy divergence, structural collapse, density anomalies, or temperature runaway. A total of 4848 snapshots were collected from 48 trajectories. The sampled systems covered graphite (12 tasks, 300–1000 K), Li_2CO_3 (12 tasks, 300–1000 K), LEDC (12 tasks, 300–900 K), and $LiPF_6$/EC/DMC electrolyte (12 tasks, 300–800 K).

- **Step 5. DecisionMakerAgent:** Selected task `selection`. Since the model showed excellent stability, 5–10% of the sampled configurations were recommended for reference calculations.

- **Step 5. SelectionAgent:** Completed sample selection (task ID: `04120404`). From 4848 snapshots, the top 7.5% high-error samples were selected using the `max_force` criterion, resulting in `selection/dataset_04120404.h5` with 364 structures (40730 atoms). The energy range was -5.01 to -4.44 eV/atom (mean -4.75), and the maximum force was 16.94 eV/Å. Overall model performance was strong, with average energy MAE of 0.032 eV/atom and force MAE of 0.21 eV/Å.

- **Step 6. DecisionMakerAgent:** Selected task `train`. The existing model `wma3pblet88e3d67` was further refined with the newly selected 364 high-error samples.

- **Step 6. TrainAgent:** Completed MLIP fine-tuning. Model `04120407` (ID: `ubjddsyv65chhyfd`) was trained from `wma3pblet88e3d67` using 3676 structures (3312 initial + 364 new samples) for 2000 epochs. Final errors were 0.014 eV/atom for energy and 0.322 eV/Å for force. Six anomalous data points were found in `init.h5`, while no anomalies appeared in the new dataset.

- **Step 7. DecisionMakerAgent:** Selected task `sample`. Model `ubjddsyv65chhyfd` was used to begin Stage 2 interface sampling, focusing on graphite/Li_2CO_3 and Li_2CO_3/LEDC interfaces in the 300–800 K range.

- **Step 7. SampleAgent:** Sampling was partially completed, with 58 out of 72 tasks successful and 14 early stops. All 42 Li_2CO_3/LEDC interface tasks succeeded, whereas 14 out of 30 graphite/Li_2CO_3 tasks stopped early due to structural collapse at high temperatures; 9 of these occurred at temperatures ≥ 700 K. A total of 13149 snapshots were obtained.

- **Step 8. DecisionMakerAgent:** Selected task `selection`. Based on the sampling results, high-error samples and early-stop trajectories from graphite/Li_2CO_3 were prioritized.

- **Step 8. SelectionAgent:** Completed Stage 2 interface sample selection, producing two datasets:

  1. graphite/Li_2CO_3 (17.5% sampling): 824 structures, 273370 atoms, energy range -6.50 to -5.14 eV/atom, maximum force 379.86 eV/Å
  2. Li_2CO_3/LEDC (7.5% sampling): 633 structures, 162514 atoms, energy range -5.00 to -4.81 eV/atom, maximum force 15.34 eV/Å

  In total, 1457 high-error samples were selected.

- **Step 9. DecisionMakerAgent:** Selected task `train`. The model was fine-tuned using the 1457 interface high-error samples.

- **Step 9. TrainAgent:** Completed fine-tuning of model `04122324` (ID: `ryhw6lnfvitqypsn`) based on `ubjddsyv65chhyfd`. After 2000 epochs, the energy MAE decreased by 21.5% to 0.0107 eV/atom, and the force MAE decreased by 3.8% to 0.310 eV/Å. Eighteen anomalous data points were detected, 10 of which came from graphite/Li_2CO_3.

- **Step 10. DecisionMakerAgent:** Selected task `sample`. The updated model `ryhw6lnfvitqypsn` was used to re-sample Stage 2 interfaces, especially to test whether high-temperature stability at the graphite/Li_2CO_3 interface had improved.

- **Step 10. SampleAgent:** All 62 MD tasks completed successfully with no early stops or anomalies. The new model showed stable graphite/Li_2CO_3 behavior at 700–1000 K, in contrast to the previous 14/30 early stops. Li_2CO_3/LEDC remained stable over 300–1000 K. A total of 18662 snapshots were collected.

- **Step 11. DecisionMakerAgent:** Selected task `selection`. Since the model exhibited excellent stability, 5–10% of samples were recommended for reference calculations.

- **Step 11. SelectionAgent:** From 18662 snapshots, 1400 high-error samples (7.5%) were selected, covering both graphite/Li_2CO_3 and Li_2CO_3/LEDC interfaces over 300–1000 K. The maximum force error reached 39 eV/Å for graphite/Li_2CO_3 and 25 eV/Å for Li_2CO_3/LEDC.

- **Step 12. DecisionMakerAgent:** Selected task `train`. The model was further refined with these 1400 Stage 2 interface samples.
- **Step 12. TrainAgent:** Completed fine-tuning of model `04131346` (ID: `o5iksgkqtw89x4pt`). Training used all five datasets, including the 1400 newly selected interface samples, over 1580 epochs. Fifteen anomalous data points were detected (6 from `init.h5`, 9 from the graphite/Li_2CO_3 dataset). Final training errors were 0.0097 eV/atom for energy and 0.312 eV/Å for force.
- **Step 13. DecisionMakerAgent:** Selected task `evaluation`. The latest model `o5iksgkqtw89x4pt` was used to evaluate the graphite/Li_2CO_3 and Li_2CO_3/LEDC interfaces.
- **Step 13. EvaluationAgent:** Completed Stage 2 interface evaluation. For the graphite/Li_2CO_3 interface, the density deviation was 0.6%, and both RDF and energy curves were highly consistent with PFP. For the Li_2CO_3/LEDC interface, the density deviation was 7.6%, the RDF main-peak positions matched well, although the predicted volume was slightly smaller. No structural collapse was observed in either case.
- **Step 14. DecisionMakerAgent:** Selected task `sample`. The model `o5iksgkqtw89x4pt` was applied to the final Stage 2 interface type, namely the LEDC/electrolyte interface, in order to complete interface coverage.
- **Step 14. SampleAgent:** Sampling completed with 35 out of 36 MD tasks successful; one task stopped early at 600 K due to a sudden temperature increase to 1574 K. Overall success rate was 97%. A total of 10823 snapshots were collected from six LEDC/electrolyte interface structures over 300–800 K.
- **Step 15. DecisionMakerAgent:** Selected task `selection`. Because the model performed well, 5–10% of the sampled structures were recommended for reference calculations.
- **Step 15. SelectionAgent:** From 10823 snapshots, 812 high-error structures (7.5%) were selected. The resulting dataset `selection/dataset_04140121.h5` contained 448400 atoms. The energy range was -4.68 to -4.52 eV/atom, the average force MAE was 0.23 eV/Å, and the maximum force error was 15.98 eV/Å.
- **Step 16. DecisionMakerAgent:** Selected task `train`. The model was refined with these 812 LEDC/electrolyte samples.
- **Step 16. TrainAgent:** Completed fine-tuning of model `04140126` (ID: `ie4o21a3w3xxr0qo`) based on `o5iksgkqtw89x4pt`. Fourteen anomalous data points were found (4 from `init.h5`, 10 from graphite/Li_2CO_3). Final training errors were 0.0085 eV/atom for energy and 0.275 eV/Å for force, corresponding to an 11.8% improvement in force error over the base model.
- **Step 17. DecisionMakerAgent:** Selected task `sample`. The model `ie4o21a3w3xxr0qo` was used to sample complete multilayer SEI structures (graphite/Li_2CO_3/LEDC/electrolyte) over 300–600 K.
- **Step 17. SampleAgent:** All 8 MD tasks completed successfully with no early stops or anomalies. The model exhibited stable performance for the complete four-layer SEI system, with no energy divergence, structural collapse, or temperature runaway. A total of 2408 snapshots were obtained. The sampled systems included two multilayer SEI structures: a compact system with 914 atoms and a standard system with 1153 atoms.
- **Step 18. DecisionMakerAgent:** Selected task `selection`. Since the model performed stably, 5–10% of the snapshots were recommended for reference calculations.
- **Step 18. SelectionAgent:** Completed sample selection for the four-layer SEI interface, generating `selection/dataset_04140836.h5` with 181 structures (approximately 207k atoms). The energy range was -5.253 to -5.087 eV/atom. The maximum force was 10.647 eV/Å. The energy MAE ranged from 0.0034 to 0.0101 eV/atom, while the force MAE ranged from 0.1940 to 0.2176 eV/Å.
- **Step 19. DecisionMakerAgent:** Selected task `train`. The model was further refined using these 181 multilayer SEI samples.
- **Step 19. TrainAgent:** Completed fine-tuning of model `04140839` (ID: `vhdko7lyhyhtbqqd`) based on `ie4o21a3w3xxr0qo`. After adding the 181 multilayer samples and training for 1073 epochs, the force MAE decreased from 0.275 to 0.262 eV/Å (a 4.7% improvement), while the energy MAE remained at 0.0085 eV/atom. No anomalies were found in the new dataset.

- **Step 20. DecisionMakerAgent:** Selected task `evaluation`. The final model `vhdko7lyhyhtbqqd` was used for comprehensive evaluation on the complete multilayer SEI structure.
- **Step 20. EvaluationAgent:** Completed the comprehensive multilayer SEI evaluation. The MLIP model `vhdko7lyhyhtbqqd` was highly consistent with PFP, showing a density deviation of only 0.70%. MD remained stable over 50000 steps with no structural collapse. RDF curves for all atom pairs (e.g., H–H, Li–Li, C–O, and F–F) matched closely, and the MSD indicated reasonable diffusion behavior. Energy, density, and volume fluctuations were normal. The minimum interatomic distance remained above 0.93 Å, and the maximum force was below 5.9 eV/Å.
- **Step 21. DecisionMakerAgent:** Selected task `end`. The workflow was successfully completed. The final MLIP model `vhdko7lyhyhtbqqd` passed all Stage 1 and Stage 2 training and evaluation steps, achieving strong consistency with PFP on multilayer SEI structures. The final model metrics were 0.0085 eV/atom for energy MAE and 0.262 eV/Å for force MAE. The model was considered ready for Stage 3 Li^+ crossing metadynamics simulations. The recommended setup is to combine this model with Plumed in well-tempered metadynamics, using the Li^+ z-coordinate as the collective variable, Gaussian height 0.5–1.0 kJ/mol, width 0.02 nm, bias factor 15–20, deposition interval 200–500 fs, total simulation time 5–10 ns, and temperature 300 K.