ENPH 455/555 Final Report

Active Learning of Interatomic Potentials for Molecular Dynamics

by

William Little

A thesis submitted to the

Department of Physics, Engineering Physics and Astronomy

in conformity with the requirements for

the degree of Bachelor of Applied Science

Queen's University

Kingston, Ontario, Canada

February 28, 2025

Abstract

This project presents the design and implementation of an automated active learning framework for training machine-learned interatomic potentials, specifically the Moment Tensor Potential (MTP), using lithium as a case study. The framework integrates molecular dynamics (LAMMPS), density functional theory validation (Quantum ESPRESSO), and potential fitting (MLIP-3) to iteratively refine the MTP through uncertainty-driven selection of atomic configurations. Starting from a minimal training set of two-atom BCC lithium cells, the pipeline identifies and validates extrapolative configurations across six progressively larger atomic systems, with sizes ranging from 2 to 54 atoms. The trained MTP is validated against density functional theory benchmarks by computing physical properties such as the relaxed lattice parameter and equation of state. Results demonstrate that the framework is capable of generating accurate and generalizable potentials using modest computational resources, supporting future extension to multi-element systems and complex material behaviors. The project underscores the effectiveness of active learning in minimizing DFT evaluations while preserving model accuracy in large-scale materials simulations.

Contents

List of Tables

List of Figures

1. Introduction

Accurate calculation of interatomic potentials is the foundation of Molecular Dynamics (MD) simulations, a critical tool used widely across fields such as chemistry, physics, materials science, and biology. These simulations rely on interatomic potentials to model the interactions between atoms and predict the evolution of atomic systems over time.

1.1 Motivation

For small-scale systems (1–100 atoms), Density Functional Theory (DFT) is one of the most widely used and accurate computational methods [1]. It provides results that are very close to those obtained from full analytical quantum mechanics, making it highly reliable for capturing quantum mechanical effects at the atomic level. The primary limitation of DFT lies in its computational scaling. The method scales as $O(N^3)$, where N is the number of atoms in the system. This cubic scaling makes DFT computationally impractical for large-scale molecular dynamics simulations involving thousands to millions of atoms, which are often required to study realistic systems on long timescales.

A more efficient and classical approach to interatomic potential calculations is using empirical models. These models rely on predefined functional forms with parameters fitted to experimental or theoretical data, making them computationally inexpensive compared to *ab initio* methods like DFT. Empirical potentials approximate the potential energy surface of a system and are designed to balance computational efficiency with reasonable accuracy for specific systems or materials.

Examples of these models include the Lennard-Jones potential [2], Morse potential [3], and Buckingham potential [4], each describing pairwise atomic interactions and are often used for simple systems. For more complex materials, many-body potentials like the Embedded Atom Method (EAM) [5] are used to capture more complex phenomenon. These potentials can describe properties of materials but may require extensive parameter fitting to experimental or ab initio data to achieve accuracy.

A new generation of interatomic potentials, referred to as Machine Learning Potentials (MLPs), is revolutionizing the field of computational materials science and molecular simulations. These potentials bridge the gap between the computational

efficiency of classical models and the high accuracy of ab initio methods like DFT. By using machine learning algorithms, MLPs can learn the complex potential energy surfaces of atomic systems directly from quantum mechanical data, enabling them to replicate DFT-level accuracy while maintaining computational speeds comparable to or even beating those of traditional empirical models.

This advantage makes MLPs a highly significant area of research, as they provide scientists with a powerful tool for performing simulations that are both more accurate and computationally feasible. With MLPs, it becomes possible to study much larger systems or simulate longer timescales than was previously achievable with DFT, while still capturing the intricate quantum mechanical details often missed by classical potentials. As a result, MLPs are creating new opportunities for exploring complex phenomena in materials science, chemistry, and biology, paving the way for faster and more detailed simulations.

1.2 Problem Statement

The objective of this design project is to develop a software package that is capable of performing the active learning framework described in Section 2.2 to train an MTP and implement the functionality to validate and test the potential. The initial training must be performed on small-cell configurations of two-atom BCC lithium unit cells generated from a DFT software to minimize training cost before beginning the active learning process. Following the intital training, MD simulations will be performed ranging from 2 to 54 atom cells at varying temperatures to effectively capture behavior of the system. The trained MTP will then be validated by investigating the physical properties of lithium metal through large-scale MD simulations. These properties include the lattice parameter and equations of state. These serve as benchmarks to assess the accuracy and reliability of the MTP trained on small-cell configurations to capture large-scale material behavior.

1.3 Project Scope and Requirements

This project is scoped to train and validate MTPs on atomic configurations containing only lithium. This simplification serves as a proof of concept for the integration of active learning with machine-learned interatomic potentials. Lithium was selected due to its relevance in nuclear fusion applications, where accurate large-scale

simulations of lithium metals under extreme thermodynamic conditions are essential for material design and performance prediction.

A list of functional requirements for the final software are described in Table 1.1.

Table 1.1: Functional requirements of the active learning framework.

Requirement	**Description**
1	Interface with LAMMPS to run molecular dynamics (MD) simulations using a trained MTP.
2	Identify extrapolative configurations using MTP uncertainty measures and save them in a standardized format.
3	Convert preselected configurations into quantum mechanical (QM) input files and update the training set based on validated results.
4	Retrain the MTP iteratively, repeating the cycle until no further extrapolative configurations are found.
5	Support at least six progressively larger lithium cells to improve scalability and generalization of the MTP model.

Software projects also contain several important non-functional requirements. The requirements tailored for this project are defined in Table 1.2.

A key motivation and constraint in the development of this software was the availability of computational resources. All design, testing, and validation were conducted on a consumer Apple M2 chip, which features four efficiency cores (2.42GHz) and four performance cores (3.49GHz). While modest by research computing standards, this setup emphasizes the framework's lightweight nature and accessibility. The use of such hardware ensures the workflow can be reproduced on widely available, non-specialized machines—lowering the barrier to entry for researchers or students without access to high-performance computing clusters.

Initial training data relies on 20 DFT calculations to construct the first MTP, although due to the computational constraint this initial training set must be on small-cell configurations (2 atoms unit cell) considering the $O(N^3)$ time complexity of DFT. The framework is currently limited to lithium and does not support complex compounds without substantial modification.

Table 1.2: Non-functional requirements of the active learning software package.

Title	**Description**
Accuracy	The trained MTP must maintain energy and force errors within an acceptable threshold relative to the QM reference (e.g., MAE $<$ 10 meV/atom).
Efficiency	The MD and active learning loop must run with minimal overhead, enabling model improvement within reasonable time on a typical workstation or cluster node.
Modularity	The system should be modular to allow extensions for multi-element systems or alternative potential models in the future.
Reproducibility	The software should be fully scriptable and version-controlled, ensuring that all experiments can be reproduced exactly.
Visualization	The software must generate output files suitable for post-processing and visualization in OVITO or similar tools.

2. Background

This software is built off two primary scientific and computational models, the Moment Tensor Potential and the active learning training framework.

2.1 Moment Tensor Potential

The Moment Tensor Potential (MTP) is a functional form of an MLIP introduced by Alexander Shapeev in 2016 [6]. MTPs are systematically improvable and are designed to balance computational efficiency with high accuracy, making them perform well for modeling complex atomic interactions.

This software implementation focuses on the MTP framework due to its combination of efficiency and accuracy, making it the ideal choice for studying interatomic interactions of materials. One of the key features of MTPs is linear parameterization, which enables gradient descent training using data derived from quantum mechanical calculations, such as energies, forces, and stresses from a DFT software like Quantum ESPRESSO [7]. This characteristic makes MTPs highly adaptable for a wide range of applications in molecular dynamics simulations and materials modeling.

In an MTP, the energy of an atom is a contribution of all neighboring atoms within a specified cutoff radius R_{cutoff}. This is formally represented by Equation 2.1.

$$E^{MTP} = \sum_{i=1}^{n} V(n_i) \tag{2.1}$$

Where N is the number of neighboring atoms within the local cutoff radius. The functional form of $V(n_i)$ is constructed using moment tensors, which encode atomic environment. These moment tensors shown by 2.2 are determined based on the relative positions of neighboring atoms and their distances from the central atom. The moment tensors are then contracted into scalar basis functions. The key idea of performing this is to ensure the model is invariant under translations, rotations, and permutations of atoms.

$$M_{\mu,\nu}(n_i) = \sum_{j} f_{\mu}(|r_{ij}|, z_i, z_j) r_{ij}^{\otimes \nu} \tag{2.2}$$

In Equation 2.2, the f_μ component is the radial component of the moment, dependent on the absolute distance between the two atoms, and the atomic species z_i and z_j, defined by the μ parameter. The $r_{ij}^{\otimes\nu}$ component is angular component, defined by the tensor product ν times.

The level of the MTP is characterized by μ and ν, often denoted as lev_{max}, representing the maximum order of tensor contractions used to describe the atomic environment. It determines the complexity of the descriptors of radial and angular interactions within the local atomic neighborhood. Higher levels correspond to higher-order tensors, which are ultimately more computationally expensive due to the increased tensor contractions. An optimal level of MTP must be determined that compensates for the accuracy and computational training requirements. The level is formally defined as Equation 2.3.

$$\text{lev}M_{\mu,\nu} = 2 + 4\mu + \nu \tag{2.3}$$

An MTP requires several hyperparameters. These include:

- **Max Level:** (lev_{max}) Determines the basis functions used to represent atomic environments
- **Inner and Outer Cutoff Radius**: (R_{cutoff}) Defines the spatial range within which neighboring atoms contribute to the energy calculation
- **Contribution Weights**: (C_{energy}, C_{stress}, C_{force}) These weights control the relative importance of matching each property to DFT data

2.2 Active Learning

Active learning is a machine learning training method that iteratively improves the training dataset by selectively adding new configurations that maximize the model's accuracy and generalization. Unlike traditional learning, which relies on a fixed training set, active learning dynamically identifies and uses underrepresented configurations during simulations, improving potential performance.

In the context of MTPs, active learning begins with an initial reference dataset generated from a DFT software and is used to train the preliminary MTP. The potential is then deployed in MD simulations During these simulations, an *extrapolation grade* is computed to assess how well the current MTP can predict each atomic

configuration it encountered in the simulation. If the extrapolation grade exceeds a user-defined threshold, it indicates that the configuration lies outside the model's trained ability and had low confidence. These configurations are used as candidates for the next training dataset. Their energies, forces, and stresses are recalculated using DFT and added to the reference set for retraining the MTP.

This iterative process ensures that the MTP is continuously trained on a diverse dataset, minimizing errors in unknown regions of configurational space while avoiding unnecessary computational costs associated with redundant data. By focusing computational resources on configurations that improve model accuracy, active learning enables MTPs to achieve near DFT-level precision while maintaining computational efficiency suitable for large-scale simulations.

3. Design & Implementation

3.1 Software Dependencies

Three primary software packages are required to implement the active learning pipeline for DFT calculations, MD simulations, and to implement the MTP model. Each software and their associated purpose in the pipeline is outlined in Table 3.1.

Table 3.1: Software package dependencies and their roles in the active learning pipeline.

Software	Purpose
Quantum ESPRESSO	Performs quantum mechanical (QM) density functional theory (DFT) calculations to validate preselected atomic configurations by computing accurate energies and forces. These serve as ground truth for retraining the MTP.
LAMMPS	Executes molecular dynamics (MD) simulations using the moment tensor potential (MTP). It is also responsible for generating new atomic trajectories and identifying extrapolative regions based on the current model.
MLIP-3	A software package for training, evaluating, and deploying machine-learned interatomic potentials—specifically the MTP. It handles model fitting, uncertainty quantification, and integration with LAMMPS for active learning.

3.2 System Architecture

The architecture of the proposed active learning framework is designed as a modular and iterative pipeline, composed of software components that communicate through standardized file formats and logical control flow. The primary objective is to automatically train an MTP for lithium metal by exploring configuration space using uncertainty-driven selection and quantum mechanical (QM) validation. The software used in this section are listed in Table 3.1.

Figure 3.1 illustrates the flow of the system architecture.

The system begins by generating a small number (20) of initial small-cell atomic configurations via Quantum ESPRESSO (2-atom BCC unit cell), which are used to

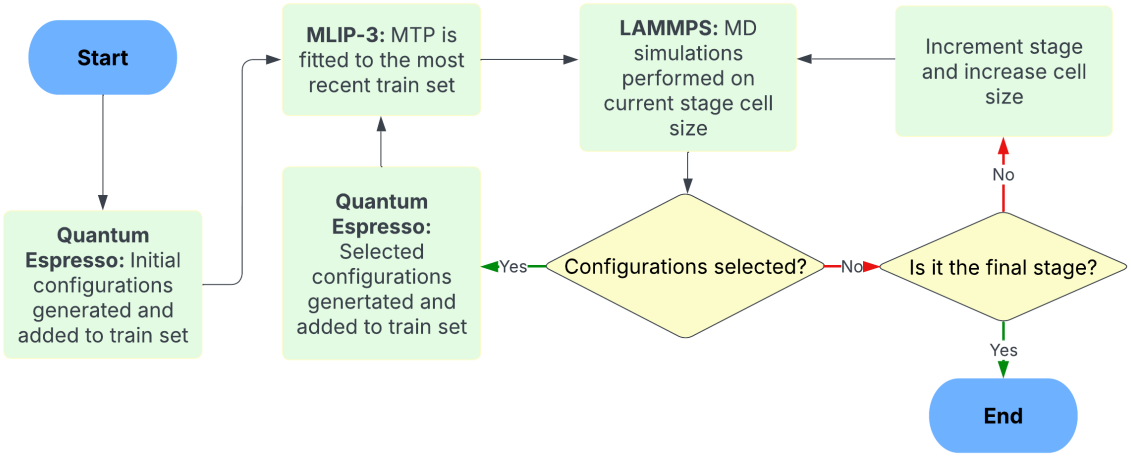

Figure 3.1: Active learning system architecture for training an MTP potential.

create the first MTP model. The MTP model is trained with MLIP-3. LAMMPS is then used to simulate a small lithium cell. If extrapolative configurations are found during the simulation, they are saved locally. These flagged configurations are passed back to Quantum ESPRESSO for high-fidelity QM calculations. The new Energy, Force, and Stress (EFS) results are appended to the training set, and the MTP is retrained. This loop repeats until no new configurations are flagged.

At this point, the framework increments to a larger atomic system (denoted as next stage in this software) to improve the generalization of the potential. In this software, the stages range from 2-54 atoms and each stage runs an MD simulation at 100 K and 500 K. These stages are shown in Table 3.2. Once the final stage is reached and no more uncertain configurations are found, the process terminates. This modular loop structure allows for easy substitution of tools like DFT calculation engine, machine learning potential architecture, etc.

3.3 Software Design

The software for this project is designed with modularity and automation as its core principles. The system is built to execute a fully autonomous active learning pipeline with minimal user intervention while ensuring flexibility to modify key components, such as the DFT engine, the interatomic potential type, or the molecular dynamics engine.

The pipeline is organized into three main stages: initialization, active learning, and

Table 3.2: Stages used in the active learning pipeline. Each stage increases the number of atoms and complexity to improve model generalization.

Stage	BCC Arrangement	# Atoms	k-space Grid
1	$1 \times 1 \times 2$	2	$4 \times 4 \times 2$
2	$1 \times 1 \times 4$	4	$4 \times 4 \times 2$
3	$1 \times 1 \times 6$	6	$4 \times 4 \times 2$
4	$1 \times 2 \times 4$	8	$4 \times 4 \times 2$
5	$2 \times 2 \times 2$	16	$4 \times 4 \times 4$
6	$3 \times 3 \times 3$	54	$2 \times 2 \times 2$

validation. Each of these is implemented using standalone Bash scripts and Python utilities that interact through a shared file structure. This separation of concerns allows for the easy substitution or extension of individual stages.

Three key Bash scripts run the main stages of the active learning iterative process:

- `initialize_mtp.sh`: Initializes the Moment Tensor Potential (MTP) by training on a small set of DFT-calculated configurations (usually one or two atoms). The trained model is saved as `pot_V0.almtp`.
- `run_md_simulations.sh`: Runs LAMMPS molecular dynamics simulations using the current MTP. It automatically selects the appropriate LAMMPS input file for the stage, checks for extrapolative configurations using MLIP uncertainty metrics, and stores outputs in stage-specific directories.
- `li_active_learning.sh`: Acts as the main script for the entire pipeline. It loops over every stage, checks for extrapolative configurations, calls the Python conversion and training tools, and moves to the next stage once the model is converged.

Python scripts handle DFT input/output processing and training set updates. These scripts abstract away the complexity of file parsing and data transformation:

- `QEInputGenerator`: Converts selected atomic configurations into valid Quantum ESPRESSO input files for energy and force calculations. It applies necessary parameters such as k-point mesh and pseudopotentials.

- `QEOutputParser`: Parses Quantum ESPRESSO output files and extracts energy and force data. This data is used to generate formatted configuration files compatible with MLIP's training format.

- `ActiveLearningUtil`: Handles logic for moving configurations, checking for duplicates, appending to the training set, and invoking MTP retraining. It also handles cleanup and tracking of preselected configurations between iterations.

All configuration files, such as LAMMPS inputs and QE input templates, are stored in structured directories and are parameterized. The use of variables like stage number, system size, and model version allows the pipeline to adjust inputs automatically. The framework detects completion or failure at each step through return codes and file pattern checks, making the entire pipeline robust and restartable.

This modular and layered structure was critical in enabling a working prototype to be developed and tested using limited computational resources, and it lays a foundation for future improvements and scalability.

3.4 Active Learning Automation

The unique functionality of the project lies in the automation of the active learning cycle, which integrates MD simulation, extrapolation detection, DFT validation, and retraining of the interatomic potential. This automation ensures the efficient exploration of configuration space without any user supervision.

Upon execution of the active learning script (active_learning.sh) described in Section 3.3, the algorithm in Listing 3.4 is autonomously run until the model is converged at the end of the last stage.

Each step of the pipeline includes checks for safe execution:

- Every external tool use (LAMMPS, QE, MLIP-3) is followed by an exit code check to ensure successful completion.

- Standard output and error logs are redirected to stage-specific directories for debugging in the case of failure.

- Failures in any stage result in a clean exit with an informative error message.

Listing 3.1: Active Learning Algorithm

```
1. Initialize training set and train initial MTP
   (initialize_mtp.sh)
2. For each stage s:
    a. Run MD with current MTP (md_simulations.sh)
    b. Check for extrapolative configs
    c. If found:
        i. Convert to QE input.
        ii. Run DFT, add to training set.
        iii. Retrain MTP.
    d. If none found: move to next stage.
3. End when no extrapolative configs are found in final stage.
```

Progression through stages is managed using nested loops. An outer loop iterates through cell sizes (stages), and an inner loop handles iterations within a stage until no new extrapolative points are found. Iteration counters and backup files ensure that training history is preserved and retraining can resume from any stage if interrupted.

3.5 Computational Setup

All MD and quantum mechanical simulations were performed on a personal machine equipped with an Apple M2 chip. The M2 architecture has four efficiency cores clocked at 2.42 GHz and four performance cores at 3.49 GHz. While these are below average specifications compared to typical HPC clusters, the framework was purposely designed for compatibility with low-resource environments to ensure portability and reproducibility.

LAMMPS was compiled with MPI support to allow for parallel execution of MD simulations. All simulations were run using a single MPI task. For small systems (2-54 atoms), the parallel speed-up was minimal, but the MPI infrastructure ensures that the pipeline is scalable to larger systems if deployed on an HPC system.

MTP training was executed using the MLIP-3 [8] software package, which was compiled natively for ARM architecture. Retraining was performed in serial due to the small size of the training dataset and the fast convergence of the model on single-species lithium cells. Each training cycle used an iteration limit of 500 with the neighborhood-based active learning strategy.

DFT calculations were performed using Quantum ESPRESSO [7]. Single-point calculations were performed on preselected configurations using the pw.x executable with the Li.pbesol-s-kjpaw_psl.1.0.0.UPF pseudopotential from Standard Solid State PPs. A consistent cutoff energy and k-point mesh were used throughout the stages to ensure reliable force and energy evaluations, as shown in Table 3.2.

4. Validation & Testing

4.1 Test Plan

An effective interatomic potential must accurately reproduce key physical properties of the system it is modeling. The MTP in this project is validated across multiple metrics, both quantitative and qualitative, using LAMMPS simulations and MLIP-3 testing features.

One of the primary validation metrics used is the **relaxed lattice parameter** of bulk lithium in a body-centered cubic (BCC) configuration. After each iteration of active learning, the potential is tested on a perturbed BCC unit cell by performing an energy minimization under zero pressure. The final relaxed lattice parameter is extracted and compared to known experimental and DFT-derived values. This test serves as a physically interpretable check on the ability of the potential to model equilibrium structures.

In addition to lattice relaxation, the **Equation of State (EOS)** testing is used. The potential is evaluated by computing the total energy of a BCC lithium cell as a function of lattice parameter. The resulting EOS curve is analyzed for its minimum and compared against DFT to verify the potential's accuracy at different compressions and expansions.

4.2 Simulation Results

Figure 4.1 shows how the predicted relaxed lattice parameter evolves with each iteration of active learning. Initially, the MTP overestimates the equilibrium lattice spacing, reflecting the limited data in the early training set. As new extrapolative configurations are identified and incorporated into the training set, the potential rapidly improves in accuracy. After just 10 iterations, the predicted lattice parameter converges to within 0.01 Å of the DFT reference, indicating the MTP has successfully learned the equilibrium structure of BCC lithium.

Figure 4.2 presents the Equation of State (EOS) for lithium, comparing total energies predicted by the trained MTP and reference DFT values across a range of lattice parameters. The MTP reproduces the DFT energy curve with good agreement, capturing both the minimum (equilibrium) energy and the overall curvature of the

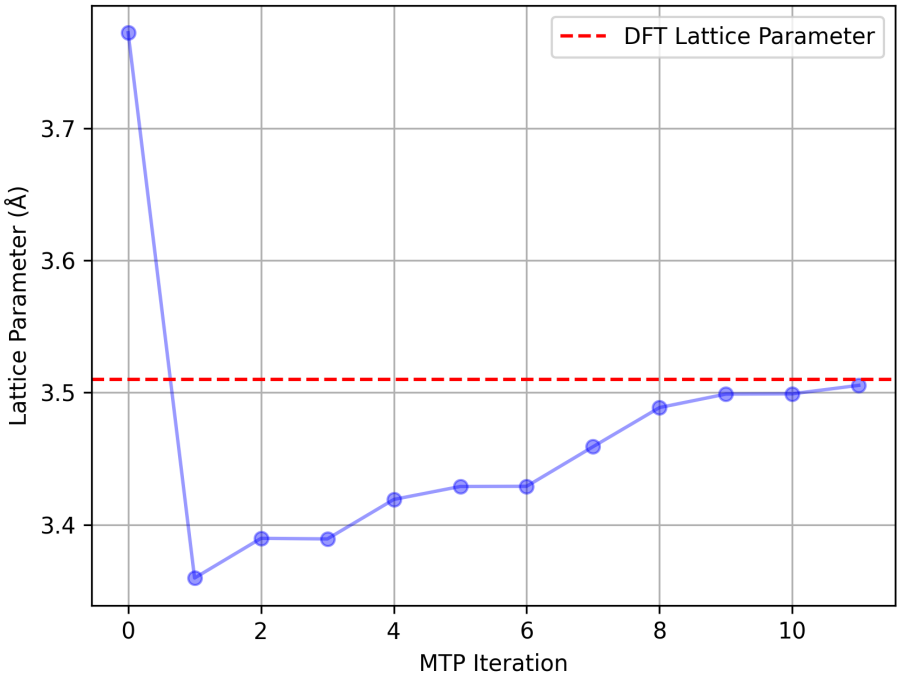

Figure 4.1: Plot of lattice parameter vs MTP iteration to validate models from each iteration of the active learning process by testing the relaxed lattice parameter of bulk BCC lithium.

EOS. Small deviations appear at some compressions and expansions, but the model remains physically reasonable throughout the tested range.

4.3 Discussion

The results demonstrate that the active learning software is capable of producing highly accurate and computationally efficient interatomic potentials with minimal compute on consumer hardware. A key strength of this framework is its adaptability. MTP training automatically detects high-error regions, leading to quick convergence even with limited initial data.

Some challenges were observed in the extrapolation behavior of the early iteration models. This suggests that additional configurations in high-pressure environments could further improve accuracy.

Despite the limitations, the validation confirms that the designed pipeline is successful in generating a transferable, interpretable model for BCC lithium. The

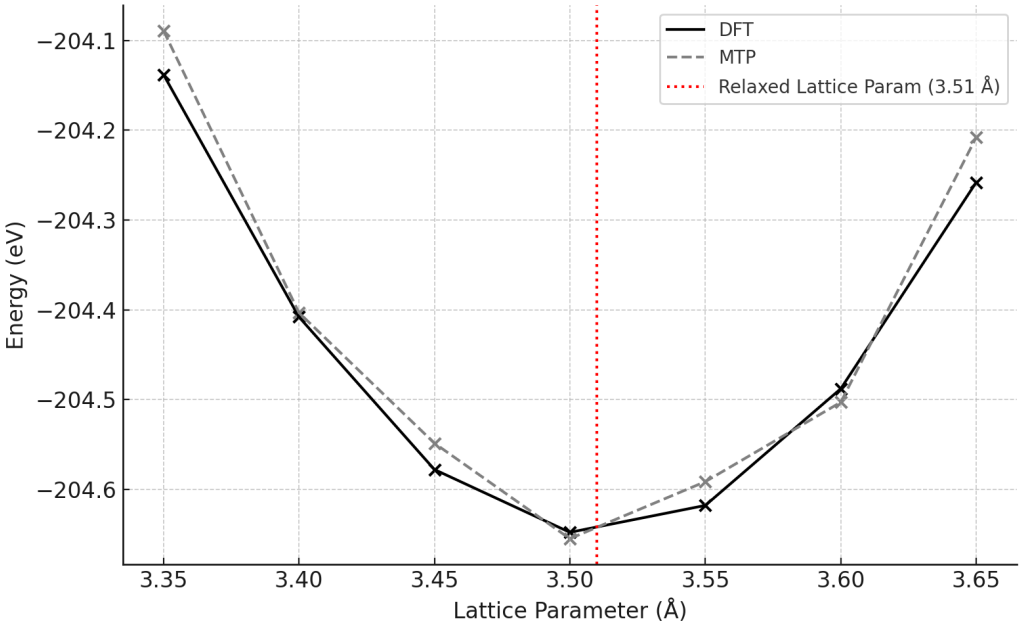

Figure 4.2: Equation of State comparison between DFT and MTP-predicted energies for various lattice parameters. The vertical dashed line indicates the relaxed lattice constant from MTP.

methodology is general enough to be extended to other elements and crystal structures with minimal changes, making it a powerful tool for future materials design efforts.

5. Ethics & Safety

This chapter outlines the ethical considerations and safety responsibilities associated with developing and deploying a machine learning-based interatomic potential within a high-performance materials modeling framework. Though the project is purely computational, it interfaces with physical phenomena particularly in the context of fusion applications.

5.1 Ethics

The development of interatomic potentials, especially when intended for use in high-consequence fields such as fusion energy, must abide by principles of responsible scientific conduct. This work uses open-source tools such as LAMMPS, Quantum ESPRESSO, and MLIP, all of which are widely accepted within the materials modeling community. By automating the active learning pipeline and documenting each design decision, this project enables reproducibility, which is essential for scientific validation.

Another ethical consideration is the environmental cost of computation. While training large models and performing thousands of DFT calculations can be energy-intensive, this project was purposely scoped to run on modest hardware (an Apple M2 chip) demonstrating that meaningful advances in materials modeling can be achieved without massive computational requirements. This aligns with efforts to make scientific tools more accessible and environmentally sustainable.

5.2 Safety

Though the framework operates exclusively in simulations, its implications extend to physical systems, with a primary motivation being the behavior of lithium in fusion reactor environments. Machine learning potentials like MTP can potentially poorly influence design decisions if deployed without careful validation. The nature of these systems demands careful observation of extrapolation control. To address this, the framework selects high-uncertainty configurations during MD runs, ensuring that only reliable predictions inform downstream tasks. Additionally, the staged system scaling from 2-atom to 54-atom cells allows the model to generalize throughout the active

learning process, avoiding instability that could arise from overfitting to small cells.

Together, these safeguards ensure that both the software and its outputs remain trustworthy and aligned with professional engineering responsibilities.

6. Conclusion

This project presented the design and implementation of an automated active learning framework for training a machine learning interatomic potential, specifically MTP, validated on lithium-based atomic systems. Through a modular and script-driven pipeline, the molecular dynamics (LAMMPS), extrapolation detection (MLIP), and quantum mechanical validation (Quantum ESPRESSO) work in conjunction to iteratively improve the interatomic potential. The framework successfully demonstrated its ability to perform the following:

- Identify extrapolative configurations from molecular dynamics simulations.
- Convert and evaluate these configurations using density functional theory (DFT).
- Retrain the MTP and repeat this loop across six progressively larger BCC systems.
- Validate learned potentials via energy, force, and equation-of-state metrics.

Design Reflections

Several design lessons emerged through this work. First, the importance of modularity and script automation was critical in enabling easy iteration and debugging. Second, the decision to structure simulations as progressive stages greatly improved the generalization of the potential, mimicking real-world scaling scenarios. Finally, minimizing the framework's computational overhead by testing and developing on standard consumer hardware (Apple M2) demonstrated that meaningful materials modeling can be accessible, efficient, and reproducible.

Future Improvements

There are many impactful opportunities for extending this work:

- Adding more active learning stages with larger and more diverse atomic configurations.
- Incorporating real-time feedback mechanisms to adjust extrapolation threshold values on the fly.

- Generalizing the pipeline to support multi-element systems and arbitrary atomic structures.

Broader Applications

While this framework was validated on lithium in a BCC configuration, its architecture is meant to be generic and applicable to a wide range of elements and crystal systems. Future efforts could expand this tool into a general-purpose MTP training framework for materials discovery, assisting in the development of accurate, low-cost potentials for systems where high-throughput DFT is otherwise prohibitive. Ultimately, this work contributes to the growing field of tools at the intersection of machine learning and materials science, aiming to improve the speeds of simulation of materials under complex thermodynamic conditions.

Bibliography

[1] N. M. Harrison, "An introduction to density functional theory," *Department of Chemistry, Imperial College of Science Technology and Medicine, and CLRC, Daresbury Laboratory*, 2001.

[2] J. E. Lennard-Jones, "On the determination of molecular fields," *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 1924.

[3] P. M. Morse, "Diatomic molecules according to the wave mechanics. ii. vibrational levels," *Physical Review*, 1929.

[4] R. A. Buckingham, "The classical equation of state of gaseous helium, neon and argon," *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 1938.

[5] M. S. Daw and M. I. Baskes, "Embedded-atom method: Derivation and application to impurities, surfaces, and other defects in metals," *Physical Review B*, 1984.

[6] A. V. Shapeev, "Moment tensor potentials: A class of systematically improvable interatomic potentials," *Multiscale Modeling and Simulation*, 2016.

[7] P. Giannozzi, S. Baroni, N. Bonini, M. Calandra, R. Car, C. Cavazzoni, D. Ceresoli, G. L. Chiarotti, M. Cococcioni, I. Dabo, A. Dal Corso, S. Fabris, G. Fratesi, S. de Gironcoli, R. Gebauer, U. Gerstmann, C. Gougoussis, A. Kokalj, M. Lazzeri, L. Martin-Samos, N. Marzari, F. Mauri, R. Mazzarello, S. Paolini, A. Pasquarello, L. Paulatto, C. Sbraccia, S. Scandolo, G. Sclauzero, A. P. Seitsonen, A. Smogunov, P. Umari, and R. M. Wentzcovitch, "Quantum espresso: A modular and open-source software project for quantum simulations of materials," *Journal of Physics: Condensed Matter*, 2009.

[8] E. Podryabinkin, K. Garifullin, A. Shapeev, and I. Novikov, "Mlip-3: Active learning on atomic environments with moment tensor potentials," *arXiv preprint arXiv:2304.13144*, 2023. Available at https://doi.org/10.48550/arXiv.2304.13144.