university of groningen / faculty of science and engineering

University of Groningen

Automating Molecular Dynamics Simulations via a Large Language Model Agent

Research Proposal

under the supervision of
Prof. dr. Andrea Giuntoli (Zernike Institute for Advanced Materials, University of Groningen)
and
Prof. dr. Matthia Sabatelli (Artificial Intelligence, University of Groningen)

Junsheng Yin (s4774280)

June 6, 2024

1 Introduction

Large language models (LLMs),such as OpenAI's ChatGPT, have demonstrated remarkable capabilities for processing and generating human-like text. LLMs demonstrate their potential to have a significant impact on a wide range of disciplines. These models are trained on diverse data sets containing large amounts of human knowledge. They are good at tasks from very simple question answers to generate context-appropriate content. The evolvement of the AI tools offers a unique opportunity to solve some of the most challenging aspects of scientific research. Specifically, when it comes to automated tasks that have traditionally required significant human expertise, AI tools can be very potential.

Among the many domains of scientific research, molecular dynamics simulation [1] represents a key area of scientific research where AI can have a transformative impact. Molecular dynamics simulation is a computational method used to simulate the physical movements of atoms and molecules over time. Its development can date back from 1950s, when Alder and Wainwright [1] studied the behavior of hard spheres with early computer simulation. Next, Rahman develops the simulation of a liquid by considering Lennard-Jones interactions [2]. Since then, more and more algorithms and computing power has been incorporated. The application of molecular dynamics has then spread and provide valuable insights into the molecular mechanisms of various physical, chemical, and biological processes. Molecular dynamics can help identify key intermediates and transition states through simulating the motion of atoms and molecules over time. Therefore, it can provide useful information for optimizing reaction conditions especially when it comes to the choice of pressure and catalyst. This is also critical to advancing domains drug discovery, materials science, and understanding fundamental biological mechanisms that are not easily observed through experimental methods.

Despite their importance, molecular dynamics simulations are complex and resource-intensive. They require expertise in both the scientific field studied and computational techniques. The setup and analysis of these simulations involve a complex series of steps that must be carefully managed to ensure the accuracy and relevance of the results. This complexity limits the accessibility of MD simulations to a relatively small number of highly specialized researchers and becomes a bottleneck during research.

So far, in the study of the impact of LLMs on scientific discovery [3], the LLMs show significant potential in molecular dynamics simulations. GPT-4 is able to retrieve information, propose design principles, recommend suitable computational methods and software packages, generate code for various programming languages, and suggest further research directions or potential extensions. However, GPT-4 can have trouble generating accurate atomic coordinates for complex molecules, processing atomic coordinates, and performing precise calculations.

Our integration of LLMs into the MD simulation workflow intends to provide a promising solution to these challenges. By automating part of the simulation process, LLMs can make these powerful tools more accessible to the broader scientific community. In this context, the potential of LLMs lies not only in automating everyday tasks, but also in their ability to learn and adapt to the specific languages and scripts used in MD simulations. One of the widely used software in this field is *LAMMPS* (Large-scale Atomic/Molecular Massively Parallel Simulator).

LAMMPS is an open source simulation tool with extensive documentation and a strong user commu-

nity, which makes it an ideal candidate for integration with AI technologies. It is very popular because it provides a wide variety of particle interaction models for different materials. It is possible to run on platforms from a single CPU core to the largest supercomputers with accelerators. Through this way, users can control over simulation details, such as the input script, new interatomic potential, or other features needed for their models [4]. The capabilities still continue increasing from the contribution of people since its release in 2004. The goal of this project is to develop an LLM agent capable of interpreting scientific questions expressed in natural language and translating them into specialized scripts required for LAMMPS simulations.

In addition, LLM also can serve as a collaborative tool to help researchers explore new hypotheses or refine their experiments based on insights gleaned from preliminary data. For example, an LLM can suggest modifications to a simulation based on previous results, optimize parameters for a more accurate simulation, or identify potential errors in code that could lead to incorrect conclusions.

2 Research problems

To achieve the objective, our project aims to address several specific research problems including:

1. How can an LLM effectively interpret and understand complex scientific problems described in natural language and accurately translate them into specialized LAMMPS scripts?

2. What strategies can be employed by an LLM to validate the correctness of the generated simulation code and optimize it for performance and then we can ensure that it runs efficiently on HPC(High-Performance Computing) clusters?

3. What methodologies can the LLM agent use to analyze and visualize the results of the MD simulations?

4. How can the LLM agent be designed to interact effectively with researchers? Furthermore, how to enhance intuitive interfaces and feedback mechanisms to improve usability?

3 Methodology

3.1 Phase 1: Model Selection and Adaptation

We will review some of the commonly used existing LLMs suitable for the project to focus on our models, such as CodeBERT, PolyCoder, Codex and PyMT5. We will evaluate each model's strengths and weaknesses in the context of interpreting scientific problems and generating LAMMPS scripts. Then we select the most suitable model based on its performance and compatibility with the project objectives.

The selected model will be fine-tuned on a specialized dataset consisting of LAMMPS scripts and associated scientific descriptions. Then we will evaluate fine-tuned model's ability to interpret scientific queries and generate appropriate LAMMPS scripts. Based on the results, hyperparameters will then be adjusted and the model retrained as necessary to enhance performance.

3.2 Phase 2: Development and Integration

The objective is to create a user-friendly interface that enables researchers to input scientific queries and interact with the LLM agent. The interface should feature robust input validation and feedback mechanisms. Therefore it can guide users in formulating precise and appropriate queries.

To ensure the accuracy and efficiency of the generated LAMMPS scripts, our project will be aimed at developing comprehensive validation routines to verify the correctness of the code. Then we will do the implementation of optimization techniques which is designed to enhance the performance of the scripts on HPC clusters.

We will develop methods for analyzing the simulation output data. Then we implement visualization tools to generate insightful plots and through this way we can enhance the understanding and interpretation of the results.

3.3 Phase 3: Testing and Validation

To ensure the robustness of the LLM agent, we will try to design a comprehensive set of test scenarios which covers a wide range of MD simulation problems. The agent's performance on these scenarios will be evaluated with a focus on accuracy, efficiency, and usability. Subsequently, we will compare the agent's performance against human-created solutions for the same scenarios. Then, areas for improvement will be identified, and finally the agent will be refined based on the evaluation results to enhance its overall performance.

3.4 Phase 4: Deployment and Monitoring

The LLM agent will be deployed within the university and in selected external research labs, accompanied by documentation and training materials to facilitate its adoption and use. The agent's performance and user feedback in real-world scenarios will be closely monitored, with regular updates and improvements made based on user input and observed performance.

4 Literature Review

The application of LLMs to scientific research has gained increasing attention in recent years. The following literature highlights key advancements relevant to the proposed project on automating Molecular Dynamics simulations using an LLM agent.

4.1 Large Language Models for Code Generation

4.1.1 A Systematic Evaluation of Large Language Models of Code

Xu et al. (2022) [5] provide a comprehensive evaluation of large language models for code generation. The study introduces *PolyCoder*, a specialized LLM focused on generating code across multiple programming languages. This research emphasizes the potential for LLMs to outperform human-written code in specific domains, particularly in languages like C. The insights from this study indicate the viability of LLMs in specialized code generation tasks. The model is relevant to the project's focus on generating LAMMPS scripts.

4.1.2 CodeBERT: A Pre-Trained Model for Programming and Natural Languages

Feng et al. (2020) [6] present *CodeBERT* This is a bimodal pre-trained model designed for both programming and natural languages. CodeBERT's ability to translate between natural language and code, as well as its state-of-the-art performance in tasks like code search and documentation generation, aligns closely with the project's objective of interpreting scientific problems and generating corresponding MD simulation scripts.

4.1.3 PYMT5: Multi-Mode Translation of Natural Language and Python Code with Transformers

Clement et al. (2020) [7] introduce *PYMT5*, a transformer-based model for translating between Python code and natural language documentation. The research emphasizes the model's proficiency in generating and summarizing Python code. The model demonstrates its potential utility in automating coding tasks. Although it is focused on Python, the methodology and performance of PYMT5 offer valuable insights into utilizing LLMs for code generation and documentation and this could be adapted for LAMMPS scripting.

4.2 Molecular Dynamics Simulation Tools and Automation

Plimpton (1995) [8] introduces *LAMMPS*, a highly versatile tool for MD simulations. LAMMPS provides a flexible scripting environment for modeling atomic and molecular systems, widely used in physics, chemistry, and materials science. The extensive documentation and community support for LAMMPS make it an ideal platform for exploring automated code generation using LLMs, as proposed in this project.

4.3 Natural Language Processing in Scientific Contexts

4.3.1 Applications of NLP in Scientific Research

Min et al. (2023) [9] is a comprehensive survey of the use of large pre-trained transformer-based language models such as BERT in various NLP tasks. The paper emphasizes how these models have drastically changed the Natural Language Processing (NLP) field.

The authors review recent work that uses these large language models to solve NLP tasks via different approaches including pre-training then fine-tuning, prompting, or text generation. They also discuss the use of pre-trained language models for data generation for training augmentation or other purposes. This work could provide valuable insights into how large language models can be adapted and applied in various fields, including potentially molecular dynamics.

4.3.2 BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding

Devlin et al. (2019) [10] present *BERT*, a groundbreaking LLM that excels in understanding context and semantics in natural language. Just as pursued in the proposed project, BERT's architecture and pre-training methodology underpin subsequent developments like CodeBERT. In addition, it informs the technical foundation for developing specialized LLMs for scientific applications.

4.4 The impact of large language models on scientific discovery: a preliminary study using gpt-4

This recent paper [3] evaluate to generate LAMMPS input to run molecular dynamics simulations and get the atomic structures. It finds GPT-4 has a clear understanding of what LAMMPS requires in terms of format and functionality. However when asked to utilize a development package to generate LAMMPS data, GPT-4 didn't perform as well in grasping the intricacies of the complex code packages when asked to perform the task without relying on any packages, GPT-4 can provide a helpful workflow, outlining the necessary steps. The scripts it generates are generally correct, but certain details still need to be filled in manually or through additional instruction.

5 Expected Outcomes

1. **Efficient MD Simulation Code Generation:** The LLM agent is expected to accurately interpret scientific problems described in natural language and generate the corresponding LAMMPS simulation scripts. This outcome will demonstrate the ability of the LLM agent to bridge the gap between textual descriptions and specialized code and then enhance the efficiency of setting up MD simulations.

2. **Improved Code Validation and Optimization:** The LLM agent could be capable of validating and optimizing the generated LAMMPS scripts for correctness and performance. This will ensure that the simulation code is both functional and efficient. At the same time we will also try to minimize errors and enhance computational efficiency.

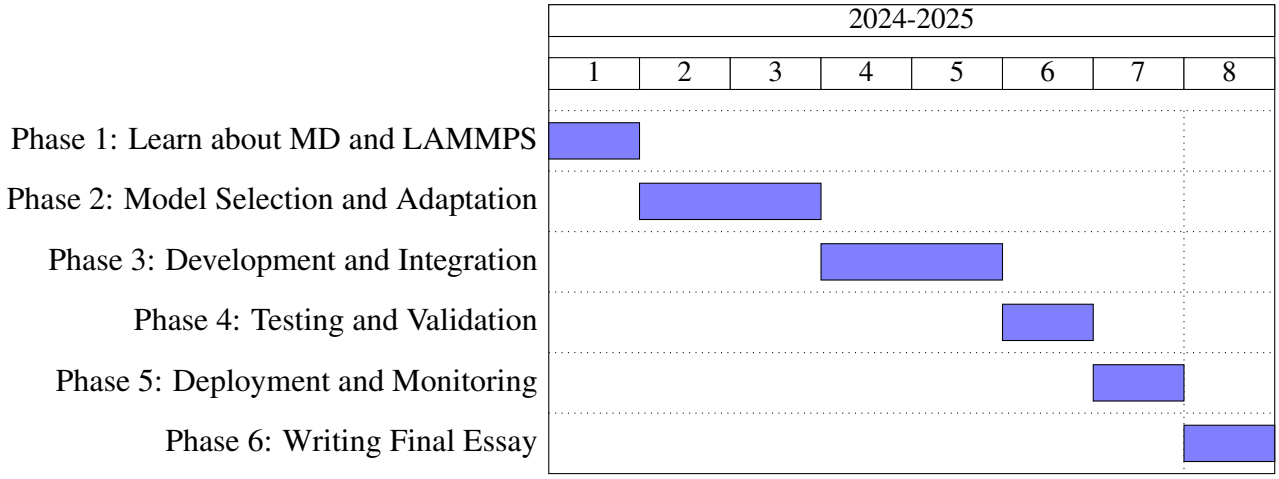
3. **Automated Job Submission and Management:** The LLM agent is expected to facilitate the submission of simulation jobs to our university's HPC cluster and handle potential errors or failures during execution. This outcome will streamline the process of running MD simulations on HPC systems.

4. **Insightful Data Analysis and Visualization:** The LLM agent should be capable of analyzing the output data from MD simulations, then generating insightful visualizations. This will enhance the interpretation of simulation results and provide researchers with valuable insights and promote scientific discovery.

5. **User-Friendly Interface and Feedback:** In addition, we expect our outcome to provide an intuitive user interface and robust feedback mechanisms. This will make it convenient to be used by researchers in different levels of computational expertise.

6 Plan of Work

The research project will be conducted over the course of eight months, divided into six main phases. Each phase will focus on different aspects of the project to ensure a logical progression and successful completion.



Bibliography

[1] Alder, B.J. and Wainwright, T.E., 1959. Studies in molecular dynamics. I. General method. The Journal of Chemical Physics, 31(2), pp.459-466.

[2] Rahman, A., 1964. Correlations in the motion of atoms in liquid argon. Physical review, 136(2A), p.A405.

[3] AI4Science, M.R. and Quantum, M.A., 2023. The impact of large language models on scientific discovery: a preliminary study using gpt-4. arXiv preprint arXiv:2311.07361.

[4] Thompson, A.P., Aktulga, H.M., Berger, R., Bolintineanu, D.S., Brown, W.M., Crozier, P.S., In't Veld, P.J., Kohlmeyer, A., Moore, S.G., Nguyen, T.D. and Shan, R., 2022. LAMMPS-a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. Computer Physics Communications, 271, p.108171.

[5] Xu, F., Li, P., He, J., Chu, C., & Zheng, L. (2022). A Systematic Evaluation of Large Language Models of Code. arXiv preprint arXiv:2202.13169.

[6] Feng, Z., Guo, D., Tang, D., Liu, N., Feng, X., & Zhou, M. (2020). CodeBERT: A Pre-Trained Model for Programming and Natural Languages. arXiv preprint arXiv:2002.08155.

[7] Clement, C., Guo, D., Hao, J., Rajamanickam, S., Swersky, K., Tarlow, D., ... & Huang, J. (2020). PYMT5: multi-mode translation of natural language and PYTHON code with transformers. arXiv preprint arXiv:2008.09593.

[8] Plimpton, S. (1995). Fast parallel algorithms for short-range molecular dynamics. Journal of computational physics, 117(1), 1-19.

[9] Min, B., Ross, H., Sulem, E., Veyseh, A.P.B., Nguyen, T.H., Sainz, O., Agirre, E., Heintz, I. and Roth, D., 2023. Recent advances in natural language processing via large pre-trained language models: A survey. ACM Computing Surveys, 56(2), pp.1-40.

[10] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4171-4186).