Master's Thesis Proposal

A fast implementation of deep neural-network potentials for molecular dynamics simulations of binary alloy

Background and Motivation

Machine-learned potentials, especially the Deep Potential (DP) models implemented in DeePMD-kit v2, offer a powerful alternative to classical force fields in molecular dynamics (MD) simulations. These models combine the high accuracy of ab initio methods DFT with the computational efficiency of classical MD, enabling realistic large-scale simulations. HALMD is a high-performance MD engine for Nvidia GPU.

This project extends DeePMD-trained potentials in HALMD to multi-component mixtures by incorporating type embeddings, allowing simulations of binary alloys and other mixed systems. Unlike previous work focused on single-species models (e.g., Cu), this thesis adds support for species-specific descriptors. Type embeddings introduce an additional one dimensional input to each atom type, enabling the model to distinguish between different elements during force and energy calculations.

Objectives

The thesis is structured into two main parts:

Part 1: Integration of Deep Potential Models into HALMD

1. Model Parameter Extraction

Develop standalone Python tool to directly extract and interpret the parameters (weights, biases, activation types, and cutoffs) from DeePMD-kit v2 trained models. This avoids limitations encountered when attempting to load the models using TensorFlow. The focus will be on models utilizing two-body embeddings (DeepPot-SE) in combination with type embeddings for multi-species systems.

The extraction pipeline will target publicly available .pb models trained using DeePMD-kit v2, including but not limited to:

- Cu model (DeepPot-SE with type embedding) from AISSquare: `https://www.aissquare.com/models/detail?pageType=models&name=Cu_fcc_slabs`
- Ag-Au nanoalloy model trained with DeepPot-SE using .pb format: AgAu-nanoalloy-model
  `https://www.aissquare.com/models/detail?pageType=models&name=AgAu-nanoalloy-model`

- Additional .pb models with verified two-body embeddings and type embeddings from AISSquare and DeepModeling repositories, selected based on availability and documentation of the descriptor settings.

2. Modular Potential Calculator in HALMD
Design and implement a modular component in HALMD that:

- Computes potentials from the extracted parameters,
- Supports two-body descriptor inference with type embedding integration for multi-component systems,
- Incorporates the additional input dimension introduced by type embeddings to distinguish atomic species,
- Handles periodic boundary conditions and neighbor lists consistently with DeePMD.

3. Validation and Benchmarking
Benchmark the accuracy of force and energy calculations by comparing HALMD outputs with reference results generated by DeePMD-kit for known systems.

Part 2: Material Properties from DeePMD-based MD Simulations

4. Thermodynamic Property Extraction
Using trajectories from Part 1, compute:

- Specific Heat Capacity (C_V, C_P), which quantifies the system's ability to store thermal energy. Under the canonical (NVT) ensemble, it can be estimated using energy fluctuations as:

$$C_V = \frac{1}{k_B T^2} \left(\langle E^2 \rangle - \langle E \rangle^2 \right)$$

  where:

  - C_V is the specific heat at constant volume,
  - k_B is the Boltzmann constant,
  - T is the absolute temperature,
  - E is the total energy (kinetic + potential),
  - $\langle E \rangle$ is the average energy over the simulation,
  - $\langle E^2 \rangle$ is the average of the squared energy.

- **Bulk Modulus**, which measures resistance to uniform compression:

  - **Thermodynamic Definition**:

$$K_T = -V \left(\frac{\partial P}{\partial V} \right)_T$$

    where:

* K_T is the isothermal bulk modulus,
* V is the system volume,
* P is the pressure,
* The derivative is taken at constant temperature T.

– **Microscopic Definition via Structure Factor (NVT Ensemble)**:

$$K_T = \frac{\rho k_B T}{S(k \to 0)}$$

where:

* ρ is the number density (N/V),
* k_B is the Boltzmann constant,
* T is the absolute temperature,
* $S(k \to 0)$ is the static structure factor in the long-wavelength limit.

5. Benchmarking with Reference Data

To validate the computed thermodynamic properties, specific heat capacity and bulk modulus will be used as primary benchmarking targets. Simulation results will be compared against experimental or reference values from the NIST Chemistry WebBook or other public thermophysical property databases under matching conditions (e.g., temperature, pressure). Accuracy will be assessed using standard error metrics.

Methodology

- Study the DeePMD v2 architecture in detail, especially two-body embedding descriptors (DeepPot-SE), type embeddings, and ResNet-based fitting networks.
- Write a custom parser to extract model parameters directly from the `.pb` files.
- Extend the existing C++ module in HALMD developed for single-species systems (e.g., Cu) to support multicomponent systems by incorporating type embeddings. This includes processing the additional input dimension introduced by atom-type-specific vectors and adapting the descriptor and network evaluation accordingly.
- Run MD simulations for NVT ensembles with the integrated potential model.
- Post-process observable data using statistical mechanics formulas to calculate macroscopic thermodynamic properties.

Expected Outcome

- A modular DeePMD-compatible extension to HALMD, supporting two-body/type-embedded potentials.
- Accurate validation of machine-learned force fields within a high-performance MD engine.
- A complete framework for computing and benchmarking thermodynamic properties (e.g., heat capacity, bulk modulus).
- Potential open source contributions to the HALMD and DeePMD-kit communities.

Supervision and Timeline

The thesis will be conducted over a period of five months. The work will be organized as follows:

Month 1:

- Study DeePMD-kit v2 architecture and model internals
- Analyze `.pb` model structure and design parameter extractor
- Implement Python parser to extract model parameters

Month 2:

- Begin HALMD integration of potential inference module
- Use a Python prototype to reproduce the calculations of the DeepMD-trained model using the extracted parameters
- Complete HALMD integration for two-body/type-embedded potentials

Month 3:

- Validate energy and force outputs against DeePMD reference data
- Run short MD simulations to test stability and consistency

Month 4:

- Perform production-scale MD simulations
- Post-process data to compute C_V and K_T
- Begin comparison against NIST or other reference databases

Month 5:

- Finalize validation, benchmarking, and documentation
- Write the thesis and prepare for defense
- Deliver and package code and results for submission or release