

MPhil in Machine Learning and Machine Intelligence

Project Proposal Coversheet 2024 - 2025

Please fill in Part 1 of this form. Arrange for your Project Supervisor and Co-Supervisor (if applicable) to approve the proposal and either sign this form or email the Course Administrator (mimi-mphil-admin@eng.cam.ac.uk) that s/he is approving your project. Please attach the Project Proposal Coversheet to the front of your Project Proposal. **It is your responsibility to obtain the approval. Incomplete Project Proposals will not be accepted by Course Administrator.**

Name:

College:

CRSID:

Course:

Part 1

Title of Project:

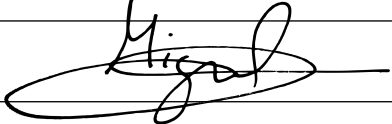
Speeding up Mace

Date of Submission:

Word counts: Proposal Workplan

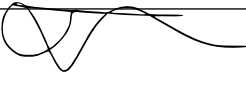
Project Supervisor:

Signature:



Co-Supervisor: Dr.

Signature:



The approval of the Project Supervisor and the Co-Supervisor (if applicable) is to be obtained by the Student. This form must be submitted to the Course Administrator by the published deadline.

Part 2

Approved (date)

Revise and resubmit by

Revision approved (date)

Assessor Notes:

Part 3

CA Date Received:

Revision Received:

CA Approved:

Thesis Project Proposal - Speeding Up MACE

Alexandre Benoit*
University of Cambridge
ab3149@cam.ac.uk

1 Project Proposal

This proposal outlines a plan to accelerate MACE [2] (Message Passing Neural Network for force-field modeling) by reducing computational overhead without sacrificing predictive performance. MACE has demonstrated excellent generalisation on datasets of energies and forces, allowing for accurate molecular dynamics simulations [1][4]. Unfortunately, its substantial computational cost can limit practical adoption, especially for large-scale or long-time-horizon simulations.

Previous work have shown that MACE achieves high accuracy on diverse molecular systems, putting forward that higher-order equivariant message passing are efficient. Recent advances demonstrate MACE's potential for foundation model such as atomistic materials chemistry [1], but also underscore its GPU-intensive nature. Meanwhile, efficient ways of training large neural networks such as knowledge distillation [3] and low-precision arithmetic [5] have gained traction in computer vision and natural language processing, which offers speed-ups with minimal accuracy loss. Applying these concepts to the MACE architecture has been largely unexplored so far and shows highly promising venues given the parallels to large-scale architecture like Transformers, which similarly face $O(n^2)$ scaling issues in certain tensor operations. A calculation that could be compared to the one in the message passing step which is known as the *convolutional tensor product*.

We will focus this problem on two key avenues of improvement:

1. **Model Distillation:** We will train a large MACE model and then transfer its learned representation to a smaller *student* model. By matching the student's predictions or intermediate features to those of the large *teacher*, we will aim to preserve the accuracy while reducing the computational footprint. The student MACE adopts the original's equivariant graph neural layers but in a compressed form which means that it will have fewer channels and thus cutting both inference and training cost.
2. **Low-precision Training:** Inspired by [5], we will integrate a reduced-precision arithmetic (using float16 or mixed-precision floats) throughout MACE's tensor product computations. Although equivariant tensor products require careful attention in the numerical calculations, the sparse Clebsh-Gordan coefficient tensors may make MACE more tolerant of quantization. We will test various low-precision training schemes to mitigate the usual reduction in accuracy.

In practice, we will try to do the following:

1. Implement the distillation method.
2. Make the MACE code working.
3. Reproduce low-precision results on a simpler MLP to establish best practices.
4. Implement a minimal MACE-like tensor product step from scratch.
5. Scale up to the full MACE pipeline in low precision

Those experiments will be traced with progress measurements and metrics such as:

- **Accuracy:** Mean Absolute Error on energies and forces, as well as downstream simulation quality such as stable molecular trajectories.
- **Speed:** Wall-clock time per training epoch and per inference step.

- **Memory Footprint:** GPU memory usage across training or inference.

More importantly, we would want to monitor how well the student MACE replicates the teacher’s predictions and how much precision reduction influences the stability, thus conducting a sensitivity analysis.

The criteria for success are relative but a significant reduction in MACE’s training and inference and minimal accuracy degradation is wanted. We want to assess MACE on various benchmark and see how robust it remains while delivering significant computational gains.

2 Schedule

Weeks 1-2 (Early to mid June): Literature Review and Project Setup. : We will begin with a literature review focusing on equivariant graph GNNs with a particular focus on MACE as well as understanding model distillation and low-precision training techniques. We would like to understand the state-of-the-art and bottlenecks that different scholars have been pointing out and develop strategies for algorithmic improvements. At the end, we will try to setup the computing environment (acquiring the open-source MACE code and configuring necessary GPU resources) and establish baseline performance metrics.

Weeks 3–4 (Mid to End June): Baseline Reproduction and Preliminary Experiments. Reproduce the original MACE baseline and begin exploratory experiments. Test low-precision training on a simple MLP to understand numerical behavior. Start model distillation by training a high-capacity teacher MACE model and designing potential student architectures.

Weeks 5–6 (Early to Mid July): Custom Implementation and Optimization of Tensor Operations. Implement and evaluate a custom tensor product operation optimized for the $l=1$ message passing step, using sparse Clebsch–Gordan coefficients. Introduce low-precision arithmetic and benchmark its impact on efficiency and accuracy.

Weeks 7–8 (Mid–End July): Integration and System-Level Experiments. Integrate all improvements into the full MACE pipeline. Run comprehensive experiments on standard benchmarks to measure the trade-off between computational cost and predictive performance. Continue refining model distillation techniques.

Week 9 (Early August): Enhancement and Bonus Tasks. Explore enhancement tasks, such as tuning precision strategies, optimizing GPU usage, or testing alternative distillation losses. This week allows room for experimentation based on insights gained so far.

Weeks 10–11 (Mid–End August): Dissertation Draft Writing and Revisions. Focus on dissertation writing and revisions. Produce a full draft by mid-August, followed by edits and polishing to prepare for final submission by the end of the month.

3 Resource Declaration

3.1 Computing Resources

This project will use high-performance GPUs for training and evaluation of Neural Network models. We will utilize resources such as the Cambridge Service for Data Driven Discovery (CSD3) - the EPSRC Tier2 National HPC Services hosted by Research Computing Services at the University of Cambridge - or equivalent computing facilities, which offer NVIDIA A100 GPUs which were used in the original MACE experiments and are sure to provide enough computational capacity for training large-scale models with high body-order message passing.

3.2 Datasets

We will use the benchmark datasets used in the original MACE paper, along with additional datasets, to evaluate and compare the performance improvements of the *Speeded-up MACE* architecture:

- **rMD17**: A refined version of the MD17 dataset containing molecular dynamics trajectories of 10 small organic molecules. These data are computed using Density Functional Theory (DFT) and are free from noisy labels found in the earlier MD17 version.
- **3BPA**: A dataset of a flexible drug-like organic molecule sampled from molecular dynamics simulations at multiple temperatures (300K, 600K, and 1200K). It tests in-domain and out-of-domain extrapolation of learned force fields.
- **Acetylacetone (AcAc)**: Contains trajectory data of a reactive molecule sampled at 300K and 600K. The dataset probes the model’s ability to extrapolate to unseen chemical configurations, including bond torsions and proton transfer reactions.

Additional datasets may be incorporated as the research progresses, depending on the needs of future experiments.

3.3 Human Participants

No part of this project involves human subjects or human data. As a results, an ethics review form is not required.

References

- [1] Ilyes Batatia et al. *A foundation model for atomistic materials chemistry*. Dec. 2023. URL: <http://arxiv.org/abs/2401.00096>. 1
- [2] Ilyes Batatia et al. “MACE: Higher Order Equivariant Message Passing Neural Networks for Fast and Accurate Force Fields”. In: *Advances in Neural Information Processing Systems*. Vol. 35. 2022. 1
- [3] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. *Distilling the Knowledge in a Neural Network*. Mar. 2015. URL: <http://arxiv.org/abs/1503.02531>. 1
- [4] J. Harry Moore, Daniel J. Cole, and Gabor Csanyi. *Computing hydration free energies of small molecules with first principles accuracy*. May 2024. URL: <http://arxiv.org/abs/2405.18171>. 1
- [5] Christopher De Sa et al. *High-Accuracy Low-Precision Training*. Mar. 2018. URL: <http://arxiv.org/abs/1803.03383>. 1