



FTorch - lowering the technical barrier of incorporating ML into Fortran models

Dominic Orchard^{1,2}, **Elliott Kasoar**^{3,1}, Jack Atkinson¹, Thomas Meltzer¹, Simon Clifford⁴, and Athena Elafrou⁵

¹Institute of Computing for Climate Science, University of Cambridge, Cambridge, United Kingdom

²School of Computing, University of Kent, Canterbury, United Kingdom

³Science and Technology Facilities Council, United Kingdom

⁴Research Computing Services, University of Cambridge, Cambridge, United Kingdom

⁵NVIDIA

Across geoscience, numerical models are used for understanding, experimentation, and prediction of complex systems. Many of these models are computationally intensive and involve sub-models for certain processes, often known as parameterisations. Such parameterisations may capture unresolved sub-grid processes, such as turbulence, or represent fast-moving dynamics, such as gravity waves, or provide a combination of the two, such as microphysics schemes.

Recently there has been significant interest in incorporating machine learning (ML) methods into these parameterisations. Two of the main drivers are the emulation of computationally intensive processes, thereby reducing computational resources required, and the development of data-driven parameterisation schemes that could improve accuracy through capturing ‘additional physics’.

Integrating ML sub-models in the context of numerical modelling brings a number of challenges, some of which are scientific, others computational. For example, many numerical models are written in Fortran, whilst the majority of machine learning is conducted using Python-based frameworks such as PyTorch that provide advanced ML modelling capabilities. As such there is a need to leverage ML models developed externally to Fortran, rather than the error-prone approach of writing neural networks directly in Fortran, missing the benefits of highly-developed libraries.

Interoperation of the two languages requires care, and increases the burden on researchers and developers. To reduce these barriers we have developed the open-source FTorch library [1] for coupling PyTorch models to Fortran. The library is designed to streamline the development process, offering a Fortran interface mimicking the style of the Python library whilst abstracting away the complex details of interoperability to provide a computationally efficient interface.

A significant benefit of this approach is that it enables inference to be performed on either CPU or GPU, enabling deployment on a variety of architectures with low programmer effort. We will

report on the performance characteristics of our approach, both in the CPU and GPU settings and include a comparison with alternative approaches.

This approach has been deployed on two relevant case studies in the geoscience context: a gravity-wave parameterisation in an intermediate complexity atmospheric model (MiMA) based on Espinosa et al. [2], and a convection parameterisation in a GCM (CAM/CESM) based on Yuval et al. [3]. We will report on these applications and lessons learned from their development.

[1] FTorch <https://github.com/Cambridge-ICCS/FTorch>

[2] Espinosa et al., Machine Learning Gravity Wave Parameterization Generalizes to Capture the QBO and Response to Increased CO₂, GRL 2022 <https://doi.org/10.1029/2022GL098174>

[3] Yuval et al., Use of Neural Networks for Stable, Accurate and Physically Consistent Parameterization of Subgrid Atmospheric Processes With Good Performance at Reduced Precision, GRL 2021 <https://doi.org/10.1029/2020GL091363>