The Journal of Open Source Software

itwinai: A Python Toolkit for Scalable Scientific Machine Learning on HPC Systems

Matteo Bunino [1], **Jarl Sondre Sæther** [1], **Linus Maximilian Eickhoff** [1], **Anna Elisa Lappe** [1], **Kalliopi Tsolaki** [1], **Killian Verder** [1], **Henry Mutegeki** [1], **Roman Machacek** [1], **Maria Girone** [1], **Oleksandr Krochak** [2], **Mario Rüttgers** [2,3], **Rakesh Sarma** [2], and **Andreas Lintermann** [2]

1 European Organization for Nuclear Research (CERN), Espl. des Particules 1, 1217 Genève, Switzerland **2** Simulation and Data Lab Fluids & Solids Engineering, Jülich Supercomputing Center, Forschungszentrum Jülich, Germany **3** Data-Driven Fluid Engineering (DDFE) Laboratory, Inha University, Incheon, Republic of Korea

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 02 September 2025
Published: unpublished

Summary

The integration of Artificial Intelligence (AI) into scientific research has expanded significantly over the past decade, driven by the availability of large-scale datasets and Graphic Processing Units (GPUs), in particular at High Performance Computing (HPC) sites.

However, many researchers face significant barriers when deploying AI workflows on HPC systems, as their heterogeneous nature forces scientists to focus on low-level implementation details rather than on their core research. At the same time, the researchers often lack specialized HPC/AI knowledge to implement their workflows efficiently.

To address this, we present `itwinai`, a Python library that simplifies scalable AI on HPC. Its modular architecture and standard interface allow users to scale workloads efficiently from laptops to supercomputers, reducing implementation overhead and improving resource usage.

Statement of need

Integrating machine learning into scientific workflows on HPC systems remains complex. Researchers must often invest substantial effort to configure distributed training, manage hyperparameter optimization, and analyze performance, while adapting to varied system architectures.

`itwinai` is a Python library that streamlines this process by providing a unified framework for scalable AI workflows. It offers consistent interfaces for distributed training, supports large-scale hyperparameter optimization, and includes tools for profiling and code scalability analysis.

Developed within the interTwin project to support Digital Twin applications in physics and environmental sciences, `itwinai` is designed to be extensible and reusable. By consolidating core functionality into a single framework, it lowers the technical barrier to HPC adoption and enables researchers to focus on scientific objectives.

Package features

The main features offered by the `itwinai` library are:

Configuration for reproducible AI workloads: a declarative, hierarchical, composable, and CLI-overrideable YAML-based configuration system that separates experimental parameters from implementation code.

Distributed training and inference: PyTorch-DDP, DeepSpeed, Horovod, and Ray(Moritz et al., 2018) distributed ML training frameworks are suppported.

Hyperparameter optimization (HPO): model performance can be improved by automatically traversing the hyperparameter space.

Ray integration provides two HPO strategies: (i) assigning multiple workers to a single trial or (ii) running many trials concurrently (Figure 1).

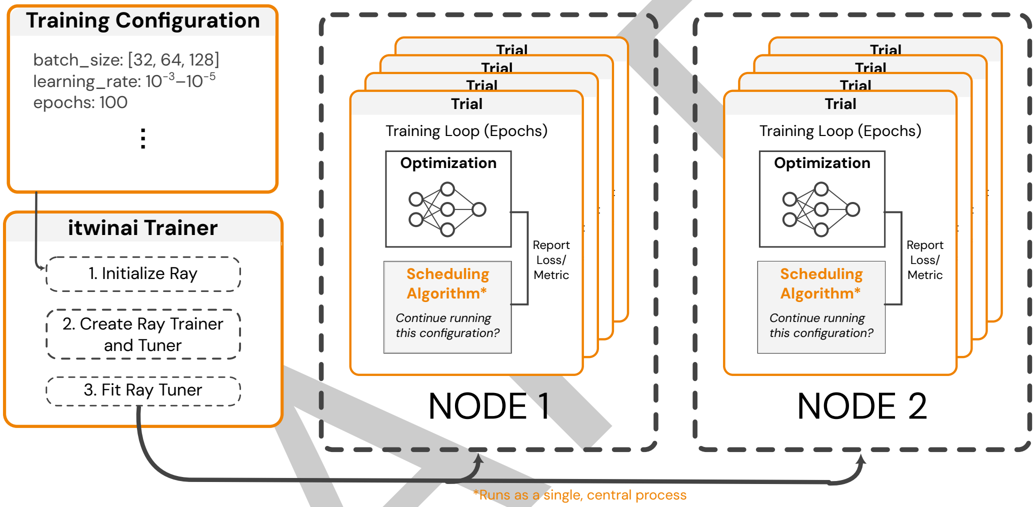

Figure 1: Conceptual representation of an HPO workflow in itwinai.

Profilers: `itwinai` integrates with multiple profilers, such as the py-spy profiler (Frederickson, 2018) and the PyTorch Profiler, and also logs metrics about training time, GPU utilization, and GPU power consumption.

ML logs tracking: `itwinai` integrates with existing ML logging frameworks, such as TensorBoard (TensorBoard Contributors, 2025), MIflow (Zaharia et al., 2018), Weights&Biases (wandb Contributors, 2025), and yProvML (Padovani & Fiore, 2025) logger, and provides a unified interface across all of them through a thin abstraction layer.

Offloading to HPC systems and cloud: To benefit from both cloud and HPC, interLink (Ciangottini et al., 2025) is used, which is a lightweight component to enable seamless offloading of compute-intensive jobs from cloud to HPC, performing an automatic translation from Kubernetes pods to SLURM jobs.

Continuous integration and deployment `itwinai` includes extensive tests (library and use cases). A Dagger pipeline builds containers on release, runs smoke tests on GitHub Actions (Azure runners: 4 CPUs, 16 GB)[1], offloads distributed tests to HPC systems via interLink, and publishes on success.

Use-case integrations

There is a wide range of scientific use cases currently integrated with `itwinai` via its plug-in architecture. Earth-observation plugins cover hydrological forecasting, drought prediction, and

[1]GitHub hosted runners define the type of machine that will process a job in your workflow. Find more here (Accessed on 2025-08-14).

The Journal of Open Source Software

climate/remote-sensing pipelines; physics plugins include high-energy physics, radio astronomy, lattice quantum chromodynamics (QCD), and gravitational-wave/glitch analysis. Packaging these as `itwinai` plugins enables reproducible, shareable workflows that run consistently on hardware ranging from personal computers to HPC systems. The full list of `itwinai` plugins can be found at this link.

Performance

`itwinai` provides tools to assess scalability and diagnose bottlenecks, enabling efficient and accountable use of HPC resources. Two complementary components are provided: scalability report generation and profiling.

Scalability report

For data-parallel training, adding more workers improves throughput, but as all-reduce communication costs grow, communication overhead eventually dominates, causing scaling to level-off or even decline. The report characterizes this trade-off across GPUs/nodes and backends, reporting wall-clock epoch time, relative speedup (Figure 2), GPU utilization (0–100%), energy (Wh), and compute-versus-other time, including collective communication and memory operations (Figure 3). Considered jointly, these metrics identify the most efficient configuration and distribution strategy, rather than relying on a single indicator. Figure 2 and Figure 3 show the scalability of the physics use case from INFN[2] targeting gravitational-wave analysis at the Virgo[3] interferometer (Tsolaki et al., 2025) (Sæther et al., 2025).

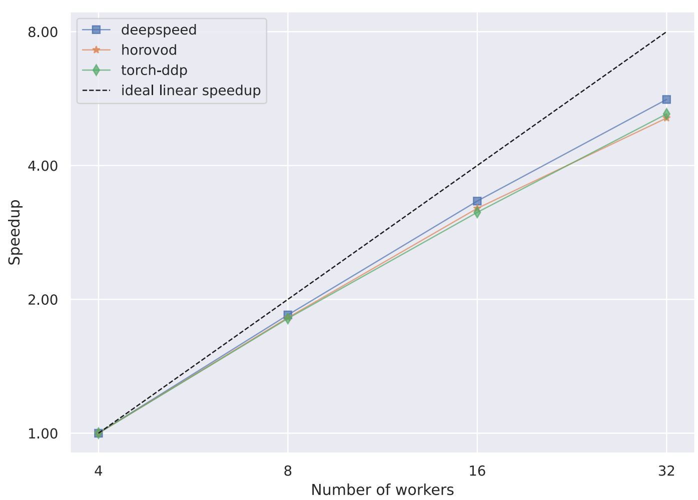

Figure 2: Relative speedup of average epoch time vs. number of workers for the Virgo use case.

[2]Istituto Nazionale di Fisica Nucleare infn.it (Accessed on 2025-08-14).
[3]Virgo Collaboration www.virgo-gw.eu (Accessed on 2025-08-14).

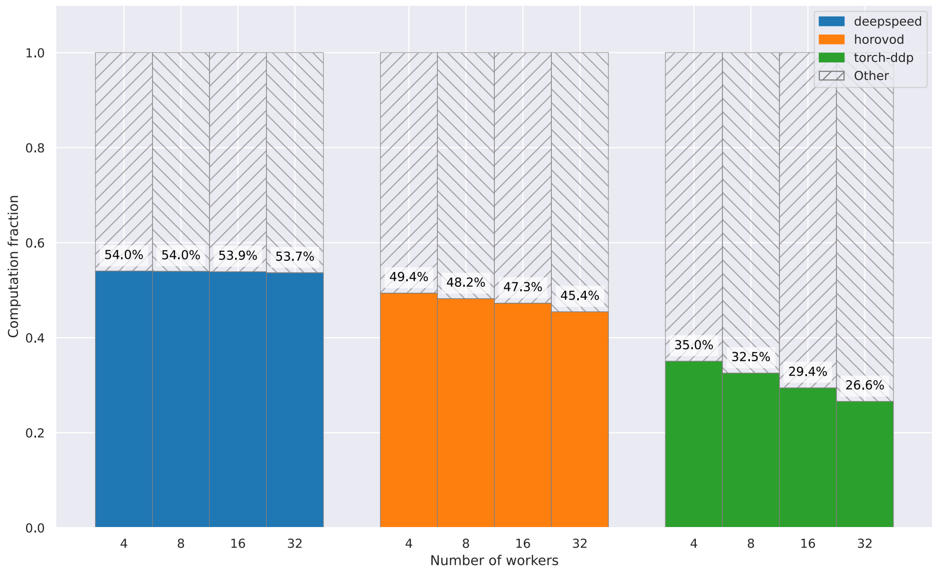

Figure 3: Proportion of time spent on computation versus other operations, such as collective communication, in the Virgo use case, broken down by number of workers and distributed framework.

Addressing bottlenecks via profiling

To explain why performance degrades, `itwinai` integrates low-overhead, sample-based profiling (e.g., py-spy (Frederickson, 2018)) and summarizes flame-graph data into actionable hotspots (e.g., data loading and I/O, kernel execution, host–device transfer, communication). These summaries guide targeted remedies such as adjusting batch size, data-loader parallelism, gradient accumulation, or backend/collective settings.

Outlook and future developments

`itwinai` provides ready-to-use ML tools that are applicable across a wide range of scientific applications. The development of the library is continued through projects ODISSEE[4] and RI-SCALE[5]. The future developments include the integration of new scientific use cases, exploring additional parallelism approaches, integrating advanced user interfaces, and adding other EuroHPC systems and performance optimization features.

Acknowledgements

This work has been funded by the European Commission in the context of the interTwin project, with Grant Agreement Number 101058386. In interTwin, `itwinai` has been actively developed on the HPC systems at JSC, such as on the HDF-ML and JUWELS Booster systems, and using EuroHPC resources, such as on the Vega HPC system. `itwinai` is an open-source Python library primarily developed by CERN, in collaboration with Forschungszentrum Jülich (FZJ).

[4]Online Data Intensive Solutions for Science in the Exabytes Era (ODISSEE): odissee-project.eu (Accessed on 2025-08-14).
[5]RI-SCALE project: riscale.eu (Accessed on 2025-08-14).

References

Ciangottini, D., Bianchini, G., & Spiga, D. (2025). interLink. In *GitHub repository*. GitHub. https://github.com/interLink-hq/interLink

Frederickson, B. (2018). py-spy: Sampling profiler for Python programs. In *GitHub repository*. GitHub. https://github.com/benfred/py-spy

Moritz, P., Nishihara, R., Wang, S., Tumanov, A., Liaw, R., Liang, E., Elibol, M., Yang, Z., Paul, W., Jordan, M. I., & Stoica, I. (2018). *Ray: A distributed framework for emerging AI applications*. https://arxiv.org/abs/1712.05889

Padovani, G., & Fiore, S. (2025). yProvML. In *GitHub repository*. GitHub. https://github.com/HPCI-Lab/yProvML

Sæther, J. S., Bunino, M., & Eickhoff, L. M. (2025). *Scalability analysis of GlitchFlow with itwinai*. Zenodo. https://zenodo.org/records/16882390

TensorBoard Contributors. (2025). TensorBoard. In *GitHub repository*. GitHub. https://github.com/tensorflow/tensorboard

Tsolaki, K., Vallecorsa, S., Vallero, S., Asprea, L., Sarandrea, F., Komijani, J., Ray, G. S., Pidopryhora, Y., & Campos, I. (2025). *interTwin D4.6 final architecture design of the DTs capabilities for high energy physics, radio astronomy and gravitational-wave astrophysics* (1 Under EC Review). Zenodo. https://doi.org/10.5281/zenodo.15120028

wandb Contributors. (2025). Weights & Biases. In *GitHub repository*. GitHub. https://github.com/wandb/wandb

Zaharia, M. A., Chen, A., Davidson, A., Ghodsi, A., Hong, S. A., Konwinski, A., Murching, S., Nykodym, T., Ogilvie, P., Parkhe, M., Xie, F., & Zumar, C. (2018). Accelerating the Machine Learning Lifecycle with MLflow. *IEEE Data Eng. Bull.*, *41*, 39–45. https://api.semanticscholar.org/CorpusID:83459546