Research Software Story - BALER

Abstract

Modern scientific experiments generate massive datasets, straining available storage and bandwidth. BALER is developed by a diverse team primarily composed of early-career researchers (undergraduate and postgraduate students and postdocs). BALER is a prototype tool (considered “Tier 2” software) that applies machine learning (autoencoder models) for domain-specific lossy data compression. All code is managed on GitHub, and contributions are submitted via pull requests for peer review. New developers are onboarded through hands-on tasks, tutorials, and example code. BALER adheres to the FAIR principles for research software. BALER’s documentation is primarily hosted on its GitHub repository and wiki. The BALER team emphasises sustainability by preserving knowledge and following structured processes.

1 The Problem

Modern scientific experiments generate massive datasets[LHC_STORAGE], straining available storage[LHC_ARCHIVAL] and bandwidth. Scientific fields such as particle physics, astrophysics, and computational fluid dynamics (CFD) produce data faster than it can be stored or transmitted. Traditional compression techniques[TRADITIONAL_COMPRESSION_HOSSEINI] are insufficient to cope with the exponential increase of these specialised datasets, and new methods are required[NEURAL_COMPRESSION_YANG_ET_AL] to drastically shrink data size without losing important scientific information.

2 User Community

BALER[WEBSITE] is developed by a diverse team primarily composed of early-career researchers (undergraduate and postgraduate students and postdocs). Contributors often join through their studies or programmes like Google Summer of Code[GSOC]. Collaborators often have limited time and varied experience levels - many members are new to machine learning or to the specific scientific domains of the data. To maintain consistency, experienced mentors hold regular meetings and guiding the development process, and define short, targeted projects.

Users of BALER come from multiple disciplines: particle physics, astrophysics, CFD, and even industry. Most users have basic knowledge of Python and machine learning. The software is designed to be straightforward to use out-of-the-box, while still allowing advanced users to customise or extend functionality as needed.

3 Technical Aspects

BALER is a prototype tool (considered “Tier 2” software) that applies machine learning (autoencoder models[AUTOENCODERS]) for domain-specific lossy data compression. It is implemented in Python and builds on the PyTorch library for neural network training.

BALER can run on standard CPUs and GPUs, and deployment on Field Programmable Gate Arrays (FPGAs) is under development for real-time use cases. The software is also distributed via Docker containers[DOCKER_CONTAINERS] to simplify deployment on local machines or High-Performance Computing (HPC) clusters. The entire codebase (including example datasets[DATASETS]) is roughly 400 MB and is publicly available on GitHub[GITHUB].

4 Libraries and Systems

- PyTorch[LIBRARY_PYTORCH]: Used to implement and train the autoencoder neural networks (with full GPU support).
- Docker[TOOL_DOCKER]: Used to containerise BALER for consistent and reproducible deployment across different systems.
- **FPGA toolchains**: Integrates with tools like hls4ml[LIBRARY_HLS4ML] and Xilinx Vivado[TOOL_XILINX_VIVADO] HLS to deploy the compression model on FPGAs for real-time data and processing.

5 Software Quality Practices

BALER's development follows standard software engineering practices. All code is managed on GitHub[GITHUB], and contributions[CONTRIBUTING] are submitted via pull requests[PULLREQUESTS] for peer review. Continuous Integration (via GitHub Actions[GITHUB_ACTIONS]) runs automated tests and enforces code style (using Black[TOOL_BLACK]) on each change to maintain quality. Docker containers are only built if syntax/code quality rules are satisfied, which is confirmed via a repository badge.

Every contribution is reviewed by senior developers, which helps ensure robustness and facilitate knowledge transfer. BALER recently changed its license to Apache 2.0[LICENSE_APACHE2] (from MIT) to encourage wider use (including in commercial contexts). Major releases are archived on Zenodo[ZENODO] with DOIs to provide citable versions for research. This is made visible to users via a CITATION.cff resource[CFF].

6 Developer Community

New developers are onboarded through hands-on tasks, tutorials, and example code. Regular project meetings and mentorship from senior members create a supportive environment for them to learn and contribute effectively.

For users, BALER provides interactive Jupyter notebooks[COLLAB_NOTEBOOK] and tutorials[COLLAB_NOTEBOOK_DEMO] to help them get started quickly. An early version of BALER is also available on the ESCAPE Virtual Research Environment (VRE), allowing anyone to experiment with it online without complex setup.

BALER addresses the common need for efficient data compression that preserves essential data quality. For example, high-energy physics experiments can use BALER to compress huge volumes of collision data, and astronomers can apply it to reduce streaming telescope data in real time before transmission or storage.

7 Tools

- GitHub Actions[GITHUB_ACTIONS]: Automates continuous integration and testing for every update.
- Black[TOOL_BLACK]: Automatically formats the code to enforce a consistent style.
- Docker[TOOL_DOCKER]: Ensures reproducible runtime environments for testing and deployment.

Using these tools improves the software's reliability and maintainability, making it easier for contributors to collaborate and for users to deploy BALER.

8 FAIR & Open

BALER adheres to the FAIR principles for research software[NATURE_FAIR4RS]:

- **Findable**: Code and documentation are publicly available on GitHub[GITHUB], with major releases archived on Zenodo[ZENODO], each assigned a DOI to facilitate discoverability and academic citation.
- **Accessible**: Code, documentation, and examples are openly accessible on GitHub under an Apache 2.0 license[LICENSE_APACHE2], allowing unrestricted access. The code is described using CITATION.cff metadata files[CFF_FILES].
- **Interoperable**: BALER produces compressed outputs as NumPy arrays[LIBRARY_NUMPY] and models stored in standard machine learning formats, enabling integration with various scientific analysis tools. At present this data requires the correct ML model for decode.
- **Reusable**: Documentation[DOCUMENTATION], demo tutorials[COLLAB_NOTEBOOK_DEMO], and modular software design encourage reuse across diverse scientific domains. Docker containers[DOCKER_CONTAINERS] also assist reproducibility by providing a known good deployment.

BALER is open source and this openness enables community adoption, contribution, and transparency, enabling impact across research fields.

Additionally, the team evaluated BALER against a FAIR software checklist and scored well on open licensing and accessibility. They identified some areas to improve, such as refining versioning practices and making outputs more interoperable (currently, compressed outputs are NumPy arrays[LIBRARY_NUMPY] that require the corresponding ML model to decode).

The BALER developers also engage in the broader community on software sustainability and FAIR practices. For instance, they participate in an ESCAPE working group focused on sustainable and FAIR machine learning[SUSTAINABLE_FAIR_ML_WG] tools.

9 Documentation

BALER's documentation is primarily hosted on its GitHub repository[GITHUB] and wiki[WIKI]. The main README provides instructions for installation and basic usage. Internal documentation, reports and presentations are also stored on shared cloud storage.

The team also archives key documents on Zenodo[ZENODO_RECORDS], and the project wiki links to these resources. By layering the documentation (README, tutorials[COLLAB_NOTEBOOK_DEMO], wiki[WIKI], archived reports), BALER ensures that knowledge is preserved through short-term projects and collaborators. This makes it easier for new users and developers to learn the tool and contribute.

10 Sustainability

The BALER team emphasises sustainability by preserving knowledge and following structured processes. All code is under version control ({% tool "git" %}), and major versions are archived on Zenodo[ZENODO] for long-term access and reproducibility. Documentation and design notes are routinely published online so that information remains available over time.

Project governance includes oversight by postdoctoral researchers with guidance from senior academic staff, which provides stability and a long-term vision. Since many contributors are students who may only work on BALER for a short period, the team aligns development goals with the academic calendar to maintain steady progress.

As of July 2025 and funding and sustainability can and does change. As with many projects, long-term funding, as always, is a challenge and changing. BALER currently relies on a variety of short-term project grants and volunteer effort. By integrating BALER into larger initiatives, the team aims to secure more sustainable resources and support for continued development.

11 Key References

- BALER GitHub Repository: https://github.com/baler-collaboration/baler
- BALER: Bespoke data compression using autoencoders, WLCG/HSF Workshop 2024, 13-17 May 2024, https://indico.cern.ch/event/1369601/contributions/5883636/
- Baler - Machine Learning Based Compression of Scientific Data[PAPER]; Fritjof Bengtsson Folkesson, Caterina Doglioni, Per Alexander Ekman, Axel Gallén, Pratik Jawahar, Marta Camps Santasmasas, Nicola Skidmore; EPJ Web of Conf. 295 09023 (2024) DOI: 10.1051/epjconf/202429509023
- BALER v1.4.0 Software Release[140_RELEASE] (2023), Zenodo: DOI 10.5281/zenodo.10723669

12 References

- [AUTOENCODERS]: https://arxiv.org/abs/2201.03898#
- [140_RELEASE]: https://zenodo.org/records/10723669
- [CFF]: https://github.com/baler-collaboration/baler/blob/main/CITATION.cff
- [COLLAB_NOTEBOOK]: https://github.com/baler-collaboration/baler-tools/blob/main/MNIST_for_baler.ipynb
- [COLLAB_NOTEBOOK_DEMO]: https://github.com/baler-collaboration/baler-demo/blob/main/baler_demo.ipynb
- [CONTRIBUTING]: https://github.com/baler-collaboration/baler/blob/main/docs/CONTRIBUTING.md
- [DATASETS]: https://github.com/baler-collaboration/baler/tree/main/workspaces/public_datasets
- [DOCKER_CONTAINERS]: https://hub.docker.com/r/balercollaboration/baler
- [DOCUMENTATION]: https://github.com/baler-collaboration/baler/tree/main/docs/setup
- [GITHUB]: https://github.com/baler-collaboration/baler
- [GITHUB_ACTIONS]: https://github.com/baler-collaboration/baler/actions
- [PAPER]: https://doi.org/10.1051/epjconf/202429509023
- [PULLREQUESTS]: https://github.com/baler-collaboration/baler/pulls
- [WEBSITE]: https://baler-collaboration.github.io/
- [WIKI]: https://github.com/baler-collaboration/baler/wiki
- [ZENODO]: https://zenodo.org/records/10723669
- [ZENODO_RECORDS]: https://zenodo.org/communities/baler-compression/records
- [CFF_FILES]: https://citation-file-format.github.io/
- [GSOC]: https://summerofcode.withgoogle.com/
- [LHC_ARCHIVAL]: https://home.cern/science/computing/data-preservation
- [LHC_STORAGE]: https://home.cern/science/computing/storage
- [LIBRARY_HLS4ML]: https://fastmachinelearning.org/hls4ml/
- [LIBRARY_NUMPY]: https://numpy.org/
- [LIBRARY_PYTORCH]: https://pytorch.org/
- [LICENSE_APACHE2]: https://www.apache.org/licenses/LICENSE-2.0
- [NATURE_FAIR4RS]: https://www.nature.com/articles/s41597-022-01710-x
- [NEURAL_COMPRESSION_YANG_ET_AL]: https://arxiv.org/abs/2202.06533

- [SUSTAINABLE_FAIR_ML_WG]: https://eucaif.org/activities/
- [TOOL_DOCKER]: https://www.docker.com/
- [TOOL_XILINX_VIVADO]: https://www.amd.com/en/products/software/adaptive-socs-and-fpgas/vivado.html
- [TOOL_BLACK]: https://black.readthedocs.io/en/stable/
- [TRADITIONAL_COMPRESSION_HOSSEINI]: https://arxiv.org/abs/2506.10000