The Journal of Open Source Software

Salt: Multimodal Multitask Machine Learning for High Energy Physics

Jackson Barr [1], **Diptaparna Biswas** [2], **Maxence Draguet** [3], **Philipp Gadow** [4], **Emil Haines** [1], **Osama Karkout** [5], **Dmitrii Kobylianskii** [6], **Wei Lai** [1], **Matthew Leigh** [7], **Ivan Oleksiyuk** [7], **Nikita Pond** [1], **Sébastien Rettie** [4], **Andrius Vaitkus** [1], **Samuel Van Stroud** [1], and **Johannes Wagner** [9]

1 University College London, United Kingdom **2** University of Siegen **3** University of Oxford, United Kingdom **4** European Laboratory for Particle Physics CERN, Switzerland **5** Nikhef **6** Department of Particle Physics and Astrophysics, Weizmann Institute of Science, Israel **7** Université de Genève, Switzerland **8** Technical University of Munich, Germany **9** University of California, Berkeley

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

Summary

High energy physics studies the fundamental particles and forces that constitute the universe, often through experiments conducted in large particle accelerators such as the Large Hadron Collider (LHC) (Evans & Bryant, 2008). `Salt` is a Python application developed for the high energy physics community that streamlines the training and deployment of advanced machine learning (ML) models, making them more accessible and promoting shared best practices. `Salt` features a generic multimodal, multitask model skeleton which, coupled with a strong emphasis on modularity, configurabiltiy, and ease of use, can be used to tackle a wide variety of high energy physics ML applications.

Some key features of `Salt` are listed below:

- Based on established frameworks: `Salt` is built upon PyTorch (Paszke et al., 2019) and Lightning (Falcon & The PyTorch Lightning team, 2019) for maximum performance and scalability with minimal boilerplate code.
- Multimodal, multitask models: `Salt` models support multimodal inputs and can be configured to perform various tasks such as classification, regression, segmentation, and edge classification tasks. Any combination of these can be used to flexibly define models for multitask learning problems.
- Customisable and extensible: `Salt` supports full customisation of training and model configuration through YAML config files. Its modular design allows for the easy integration of custom dataloaders, layers, and models.
- Train at scale: `Salt` can handle large volumes of data with efficient HDF5 (The HDF Group, 1997) dataloaders. It also includes multi-GPU support from Lightning, enabling distributed training.
- Deployment ready: `Salt` facilitates ONNX (Bai et al., 2019) serialization for integrating models into C++ based software environments.

Statement of need

In high energy physics research the reliance on ML for data analysis and object classification is growing (Cagnotta et al., 2022; Guest et al., 2018). `Salt` meets this growing need by providing a versatile, performant, and user-friendly tool for developing advanced ML models. `Salt` was originally developed to train state of the art flavour tagging models at the ATLAS experiment (ATLAS Collaboration, 2008) at the LHC. Flavour tagging, the identification of jets from bottom and charm quarks, plays a crucial role in analysing ATLAS collision data. This process is key for precision Standard Model measurements, particularly in the characterisation of the Higgs boson, and for investigating new phenomena. The unique characteristics of hadrons containing bottom and charm quarks – such as their long lifetimes, high mass, and high decay multiplicity – create distinct signatures in particle detectors that can be effectively exploited by ML algorithms. The presence of hadrons containing bottom and charm quarks can be inferred via the identification of approximately 3-5 reconstructed charged particle trajectories from the weak decay of the heavy flavour hadron admist several more tracks from the primary proton-proton interaction vertex.

While initially developed for flavour tagging, `Salt` has evolved into a flexible tool that can be used for a wide range of tasks, from object and event classification, regression of object properties, to object reconstruction (via edge classification or input segmentation), demonstrating its broad applicability across various data analysis challenges in high energy physics.

Model Architecture

Salt is designed to be fully modular, but ships with a flexible model architecture that can be configured for a variety of use cases. This architecture facilitates the training of multimodal and multitask models as depicted in Figure 1, and is designed to take advantage of multiple input modalities. In the context of jet classification, these input modalities might include global features of the jet and varying numbers of jet constituents such as charged particle trajectories, calorimeter energy depositions, reconstructed leptons, or inner detector spacepoints. The architecture is described briefly below. First, any global input features are concatentated with the features of each constituent. Next, an initial embedding to a shared representation space is performed separately for each type of constituent. The different types of constituents are then projected into a shared representation space by a series of initialisation networks. The embedded constituents are then combined and fed into a encoder network that processes constituents of different modalities in a unified way. The encoder then outputs to a set of task-specific modules, each tailored to a specific learning objective. This architecture allows the model to leverage all the available detector information, leading to improved performance. A concrete example of this architecture is in use at ATLAS (*Graph Neural Network Jet Flavour Tagging with the ATLAS Detector*, 2022; *Transformer Neural Networks for Identifying Boosted Higgs Bosons decaying into $b\bar{b}$ and $c\bar{c}$ in ATLAS*, 2023).

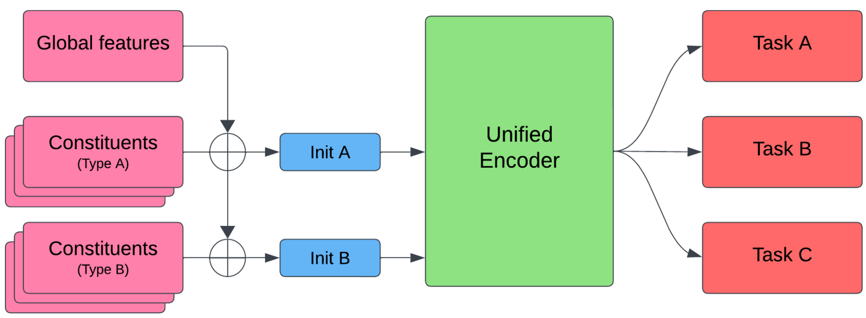

Figure 1: This diagram illustrates the flow of information within a generic model trained using `Salt`. In this example, global object features are provided alongisde two types of constituents. The model is configured with three training objectives, each of which may relate to the global object or the one of the constituent modalities. Concatenation is denoted by $\oplus$.

Related work

`Umami` (Barr & others, 2024) is a related software package in use at ATLAS. While `Salt` relies on similar preprocessing techniques as those provided by `Umami`, it provides several additional features which make it a more powerful and flexible tool for creating advanced ML models. Namely, `Salt` provides support for multimodal and multitask learning, optimised Transformer encoders (Vaswani et al., 2017), and distributed model training.

Acknowledgements

The development of `Salt` is part of the offline software research and development programme of the ATLAS Collaboration, and we thank the collaboration for its support and cooperation. This work is funded in part by the UK's Science and Technology Facilities Council via University College London's Centre for Doctoral Training in Data Intensive Science, and the Royal Society.

References

ATLAS Collaboration. (2008). The ATLAS Experiment at the CERN Large Hadron Collider. *JINST*, *3*, S08003. https://doi.org/10.1088/1748-0221/3/08/S08003

Bai, J., Lu, F., Zhang, K., & others. (2019). ONNX: Open neural network exchange. In *GitHub repository*. https://github.com/onnx/onnx; GitHub.

Barr, J., & others. (2024). Umami: A python toolkit for jet flavour tagging. In *GitHub repository*. https://github.com/umami-hep/umami-preprocessing; GitHub.

Cagnotta, A., Carnevali, F., & Iorio, A. D. (2022). Machine learning applications for jet tagging in the CMS experiment. *Applied Sciences*, *12*(20), 10574. https://doi.org/10.3390/app122010574

Evans, L., & Bryant, P. (2008). LHC Machine. *JINST*, *3*, S08001. https://doi.org/10.1088/1748-0221/3/08/S08001

Falcon, W., & The PyTorch Lightning team. (2019). *PyTorch Lightning* (Version 1.4). https://doi.org/10.5281/zenodo.3828935

Graph Neural Network Jet Flavour Tagging with the ATLAS Detector. (2022). CERN. https://cds.cern.ch/record/2811135

Guest, D., Cranmer, K., & Whiteson, D. (2018). Deep learning and its application to LHC physics. *Annual Review of Nuclear and Particle Science*, *68*(1), 161–181. https://doi.org/10.1146/annurev-nucl-101917-021019

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

The HDF Group. (1997). *Hierarchical Data Format, version 5*.

Transformer Neural Networks for Identifying Boosted Higgs Bosons decaying into $b\bar{b}$ and $c\bar{c}$ in ATLAS. (2023). CERN. https://cds.cern.ch/record/2866601

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention Is All You Need. *arXiv e-Prints*, arXiv:1706.03762. https://doi.org/10.48550/arXiv.1706.03762