Baler: A ML-based Compression Tool

Introduction

One common issue in vastly different fields of research and industry is the ever-increasing need for more data storage. In 5 years, the ATLAS experiment at CERN is expected to need 3-7 times more storage than will be available [1]. A general cross-disciplinary compression algorithm is impossible to obtain as it needs to be domain specific. We present Baler, a machine learning based compression tool which derives a compression method tailored to your data.

The Problem

- Many different fields in science and industry struggle with having too much data and too little storage
- There is a high demand to effectively compress data more than conventional loss-less methods like gzip
- However, good compression methods require domain-specific knowledge and implementation

The Solution

- With lossy compression, one can achieve much higher data reduction than with loss-less methods [2]
- With machine learning, the method can be tailored to the user's data without much expertise
- Autoencoders are a type of neural network which are trained to compress and decompress your data

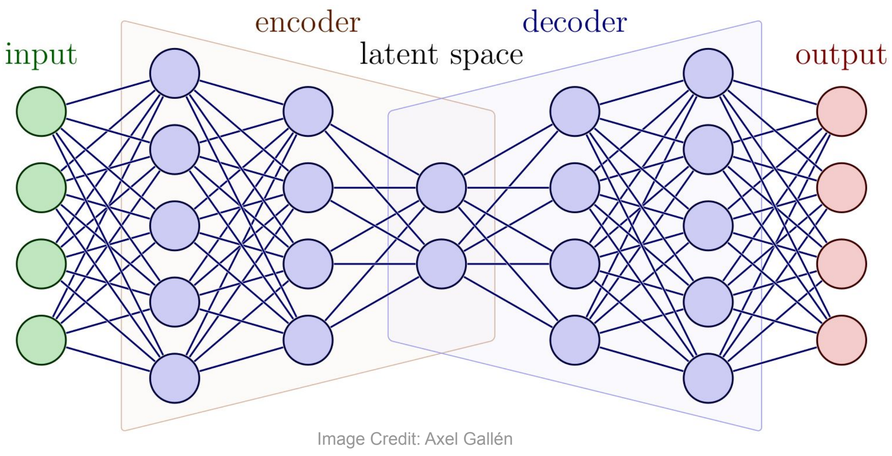


Image Credit: Axel Gallén

Baler

- In order to evaluate the feasibility of ML-based compression we developed a tool called “Baler”
- Baler provides 3 main modes of operation

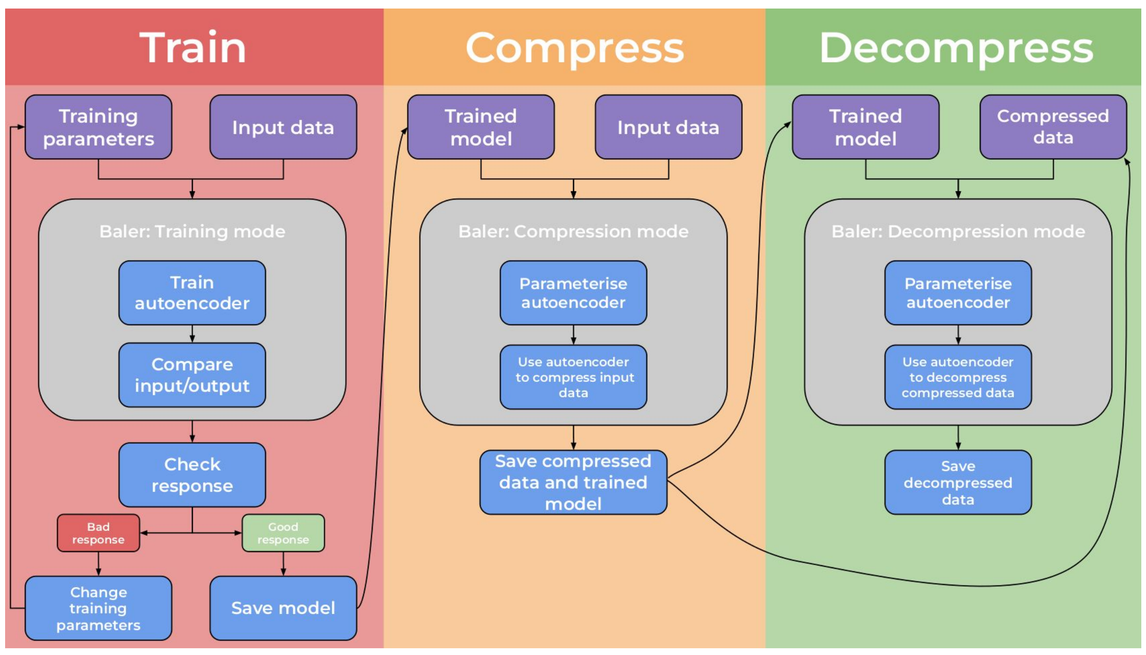


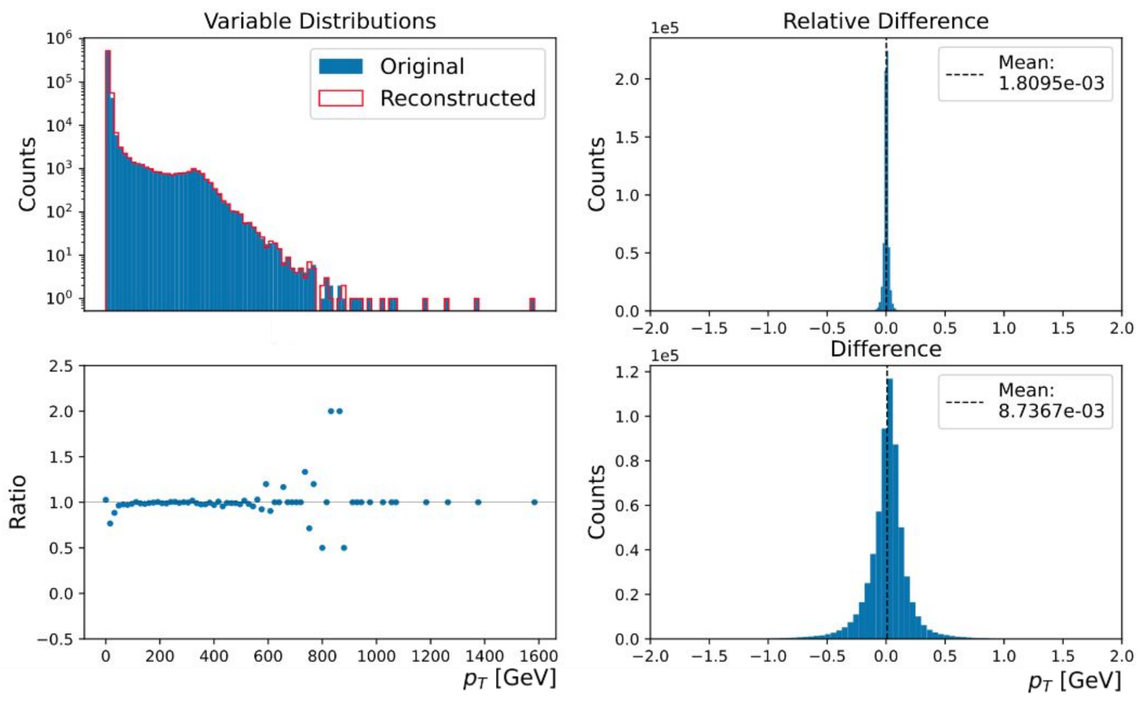
Image Credit: Oliver Woolland

- The project started as a Jupyter notebook but has in 6 months grown into a collaboration of 10 contributors
- The source code and simple tutorials are provided in our GitHub repository:
  - https://github.com/baler-collaboration/baler

```
poetry run python baler --project firstWorkspace firstProject --mode train
```

Results

- For particle physics data from the CMS experiment, we achieve good data reconstruction by compressing the file to 71% of the original file size

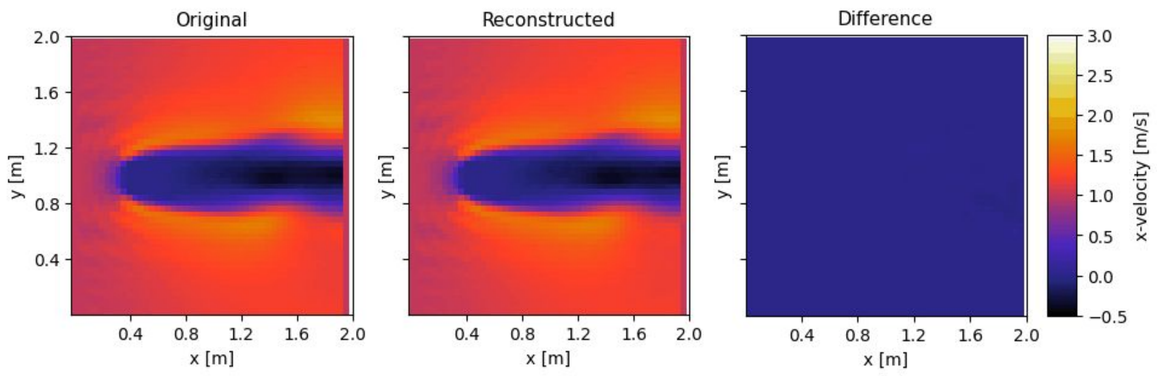


- The dilemma is that conventional loss-less methods like gzip can compress to 25% of the original file size
  - Because the data contains a lot of repeating values
- If Baler is applied closer to the detector the data values will be more unique and Baler more competitive

- Results explained in greater detail here:
  - https://arxiv.org/abs/2305.02283

Results in Other Disciplines

- Baler was designed as a cross-disciplinary tool
- Performs well when applied to a fluid simulation compressing to 0.5% original file size



- The dilemma is that the decoder used to decompress the file is 0.6 GB when the original file size is 1.2 MB
- We are hopeful to improve soon as other researchers have shown results with negligible decoder sizes [2]

Future Work

- Apply Baler on detector-level particle physics data
- Create light-weight models for fluid dynamics
- Implement error-bound compression
- Add capability for files larger than RAM
- Recruit researchers from other disciplines
  - Medicine, biology, atomic physics, solid-state physics, and industry

Help us beat the algorithm!
Fork, star, and follow our GitHub repository

Authors: Alexander Ekman
alexander.ekman@hep.lu.se

Axel Gallén
ax6264ga-s@student.lu.se

References:
[1] P. Calafiura, J. Catmore, D. Costanzo, and A. Di Girolamo, ATLAS HL-LHC Computing Conceptual Design Report, tech. rep. (CERN, Geneva, 2020)
[2] Tao Lu et al. “Understanding and Modeling Lossy Compression Schemes on HPC Scientific Data”. In: 2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS). 2018, pp. 348–357. doi: 10.1109/IPDPS.2018.00044.

The University of Manchester

Lund University