BALER: Machine-Learning-Based Compression of Scientific Data in Real Time

James Smith
University of Manchester
IOP HEPP APP NPP 2024

MANCHESTER 1824
The University of Manchester

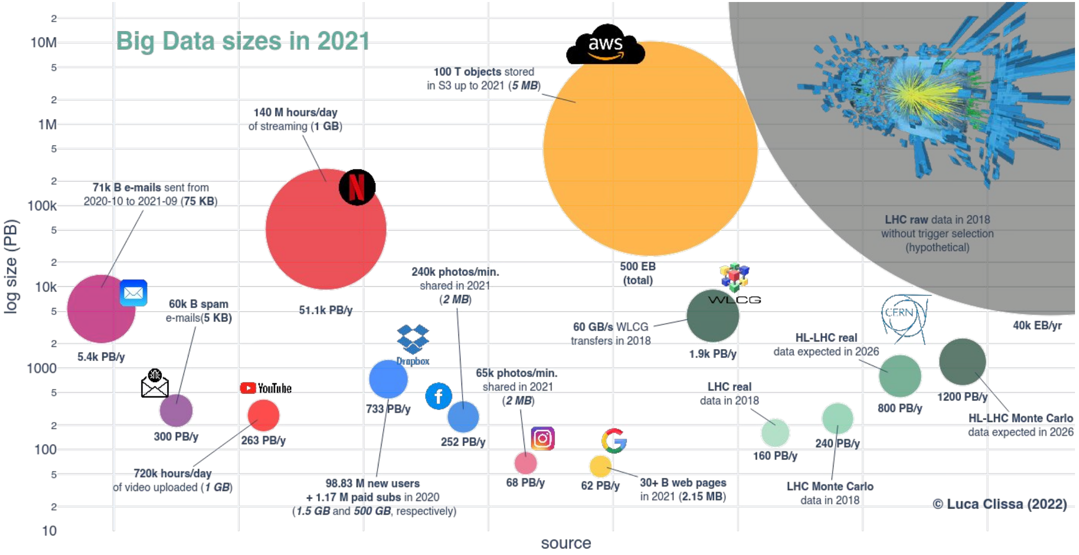
European Research Council
Established by the European Commission

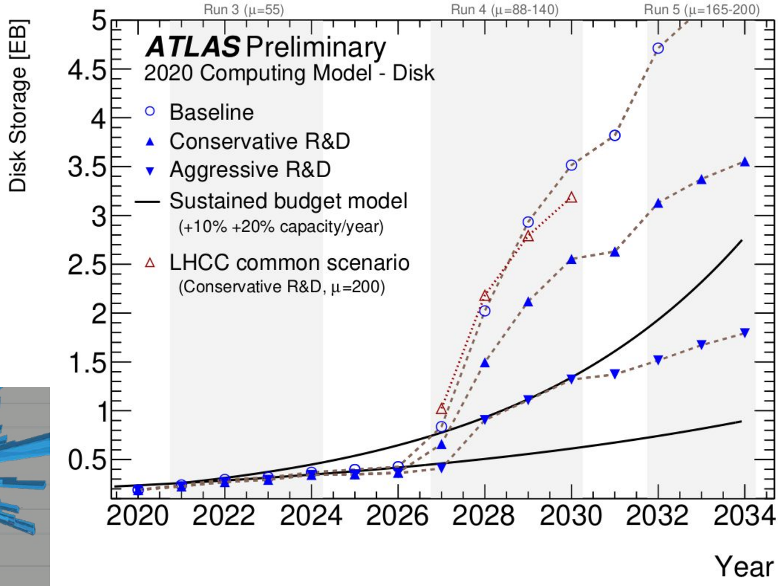
WARWICK
THE UNIVERSITY OF WARWICK

LUND UNIVERSITY

The Problem

- **Too much data, too little storage**
- Not unique to LHC Experiments
- High demand for compression





ATLAS HL-LHC Computing Conceptual Design Report
Calafiura, P ; Catmore, J ; Costanzo, D ; Di Girolamo, A
http://cds.cern.ch/record/2729668/

https://cloud.datapane.com/reports/dkjK28A/big-data-2021/ - Image by Luca Clissa

A Solution

- One approach: Lossy compression
- One problem: Lossy compression needs to be tailored
- Solution: **Lossy Machine Learning based compression**

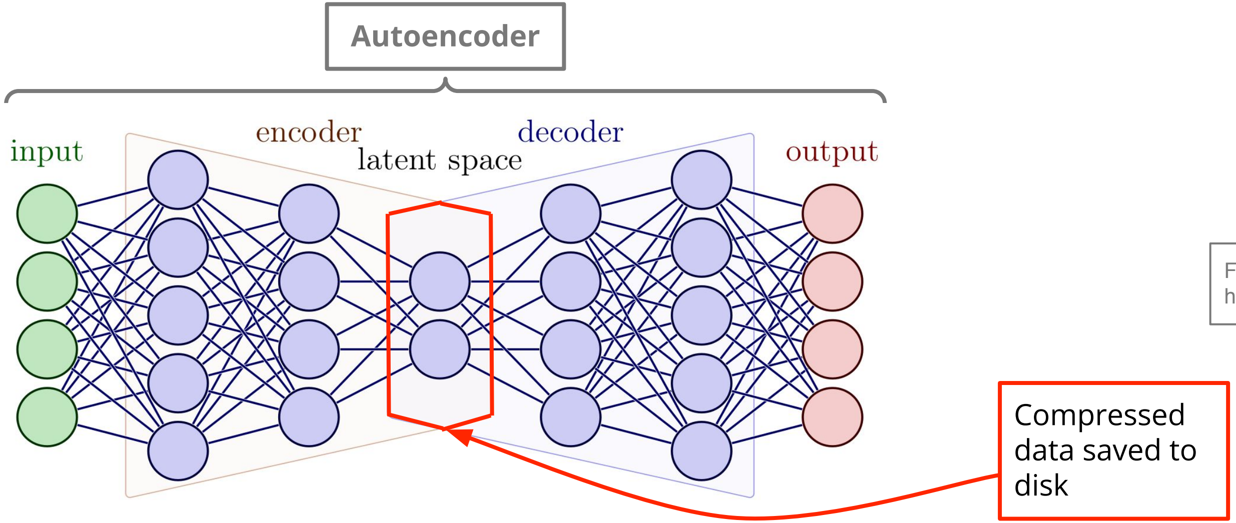


Figure modified from: https://tikz.net/neural_networks/

Our Tool: “Baler”

- We have created a tool called “**Baler**” to help investigate the viability of this compression

- **Multidisciplinary** tool

- Distributed and developed as an **open source project**
  - https://github.com/baler-collaboration/baler

- Simple to install as a **pip** package or as command line tool
  - `pip install baler-compressor`
  - `Poetry run python baler --project=CMS --mode=train`
  - **Docker** also available

arXiv:submit/4872497 [hep-ex] 3 May 2023

Baler - Machine Learning Based Compression of Scientific Data

F. Bengtsson[1] C. Doglioni[2] P.A. Ekman[1] A. Gallén[1] P. Jawahar[2] A. Orucevic-Alagic[1] M. Camps Santasmasas[2] N. Skidmore[2] O. Woolland[2]

[1] *Lund University*

[2] *University of Manchester*

Abstract: Storing and sharing increasingly large datasets is a challenge across scientific research and industry. In this paper, we document the development and applications of Baler - a Machine Learning based data compression tool for use across scientific disciplines and industry. Here, we present Baler's performance for the compression of High Energy Physics (HEP) data, as well as its application to Computational Fluid Dynamics (CFD) toy data as a proof-of-principle. We also present suggestions for cross-disciplinary guidelines to enable feasibility studies for machine learning based compression for scientific data.

1 Introduction

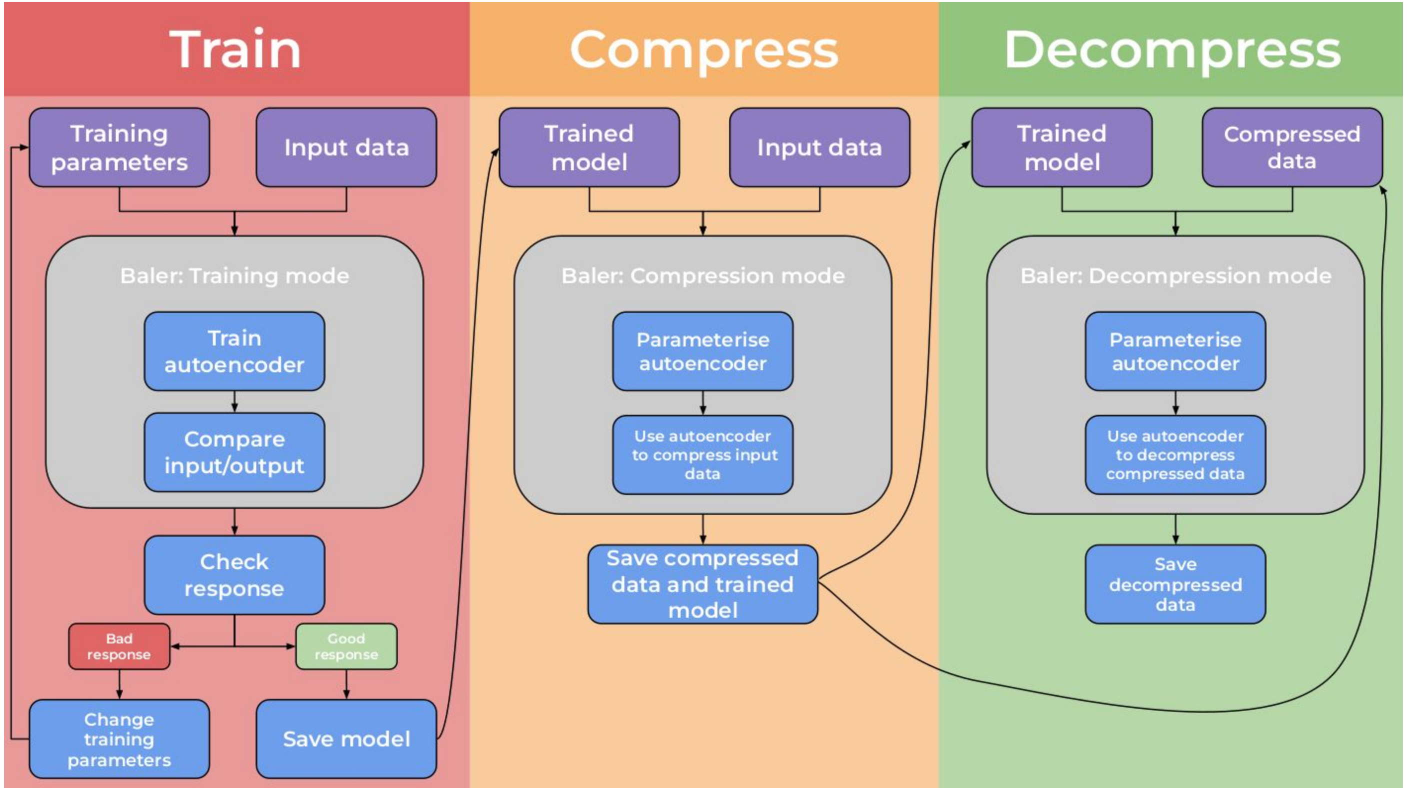
Many different fields of science share a common issue; storing ever-growing datasets. By the end of the next decade, the Large Hadron Collider (LHC) experiments will have over an order of magnitude more data to analyze than currently [1–3]; the Square Kilometre Array (SKA) experiment is expected to record 8.5EB of data over its 15-year lifespan [4] and fields such as Computational Fluid Dynamics (CFD) rely on TB-sized simulation samples that need to be stored and shared. Without significant R&D, the datasets expected to be collected by big-data science experiments are projected to exceed the available storage resources (see e.g. Fig. 2 of Ref. [1] for the case of the ATLAS experiment at the LHC). This cross-disciplinary issue is not limited to scientific research and extends to industrial operations [5].

1.1 Lossy data compression in high energy physics

A common mitigation strategy to this problem involves compressing data using lossless algorithms, see e.g. Refs. [6–8]. Once the storage limit is reached, one is forced to discard parts of the dataset, or only save certain features of the data. Generally, this can be done without impacting the overall scientific program of the experiments, for example by using a data selection system called *trigger* that only stores data satisfying certain pre-determined characteristics that ensure the dataset will be aligned with the experiment's main scientific goals. However, saving only a subset of data is not ideal for processes where additional

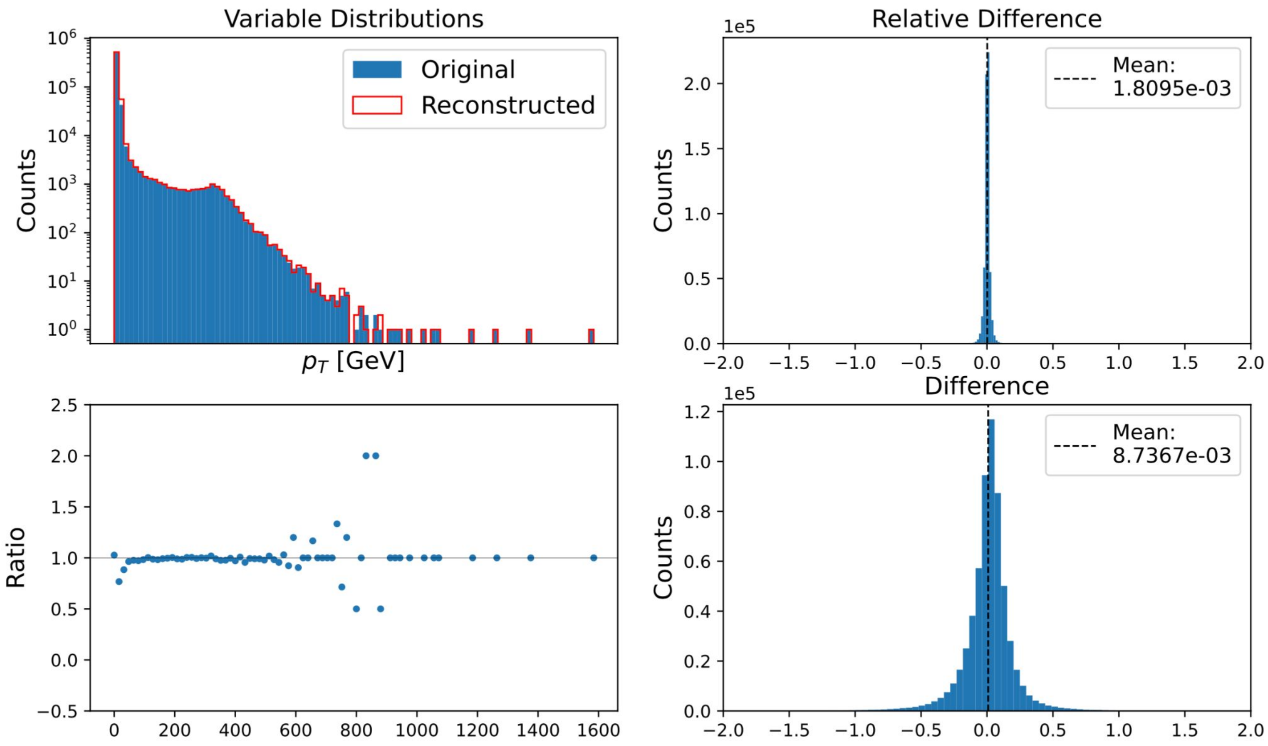
https://arxiv.org/abs/2305.02283

Workflow


Train
Compress
Decompress
Training parameters
Input data
Baler: Training mode
Train autoencoder
Compare input/output
Check response
Bad response
Good response
Change training parameters
Save model
Trained model
Input data
Baler: Compression mode
Parameterise autoencoder
Use autoencoder to compress input data
Save compressed data and trained model
Trained model
Compressed data
Baler: Decompression mode
Parameterise autoencoder
Use autoencoder to decompress compressed data
Save decompressed data

Results: Jet Transverse Momentum

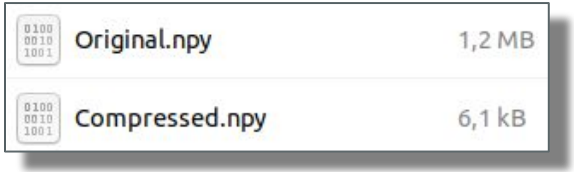
- Open CMS Data
  - ~ 600 000 jets
- 24 variables per jet compressed to 14 variables
- **58% original size**

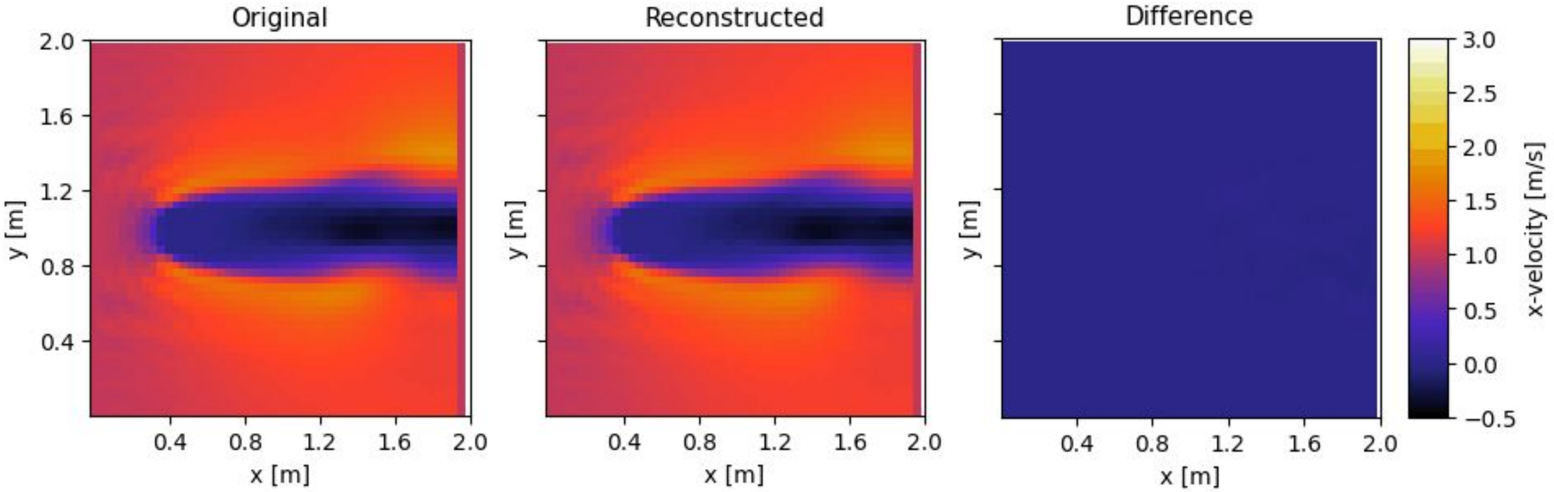


DOI:10.7483/OPENDATA.CMS.KL8H.HFVH

Results: CFD

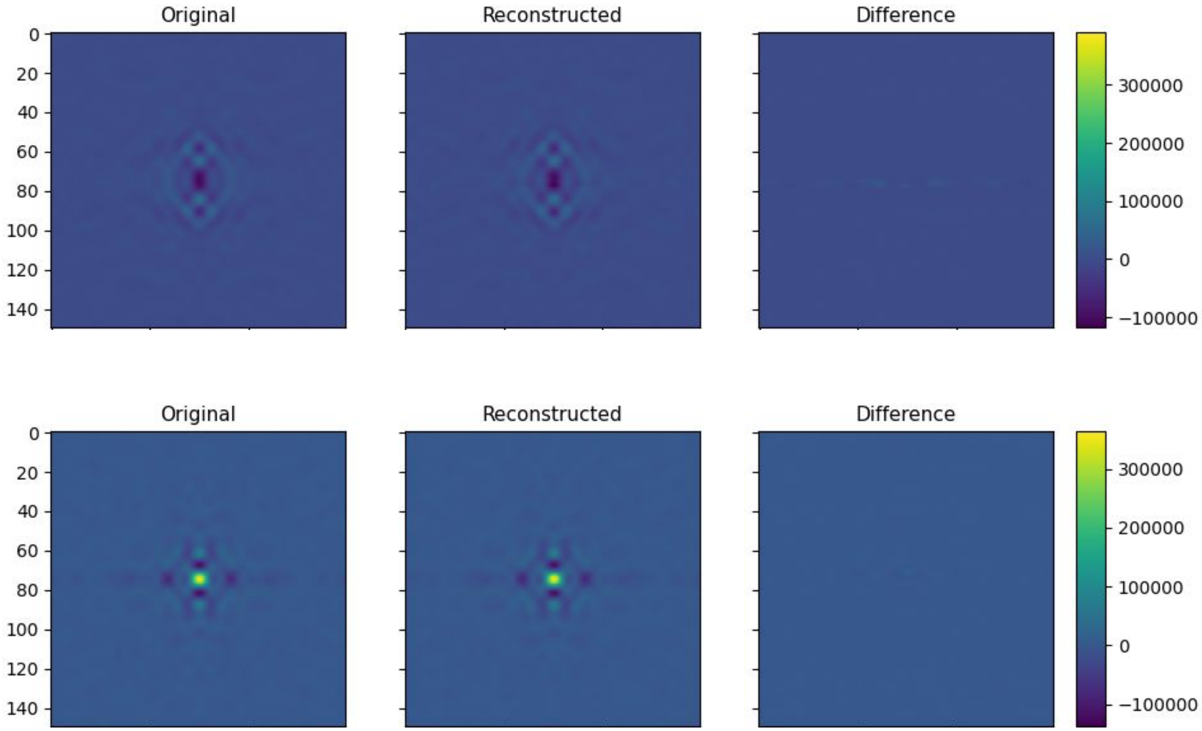
- Data consists of 2D slice of a liquid flowing over a cube
- The compressed file is **0.5%** the size of the input
- Model much larger…   (O(MB))





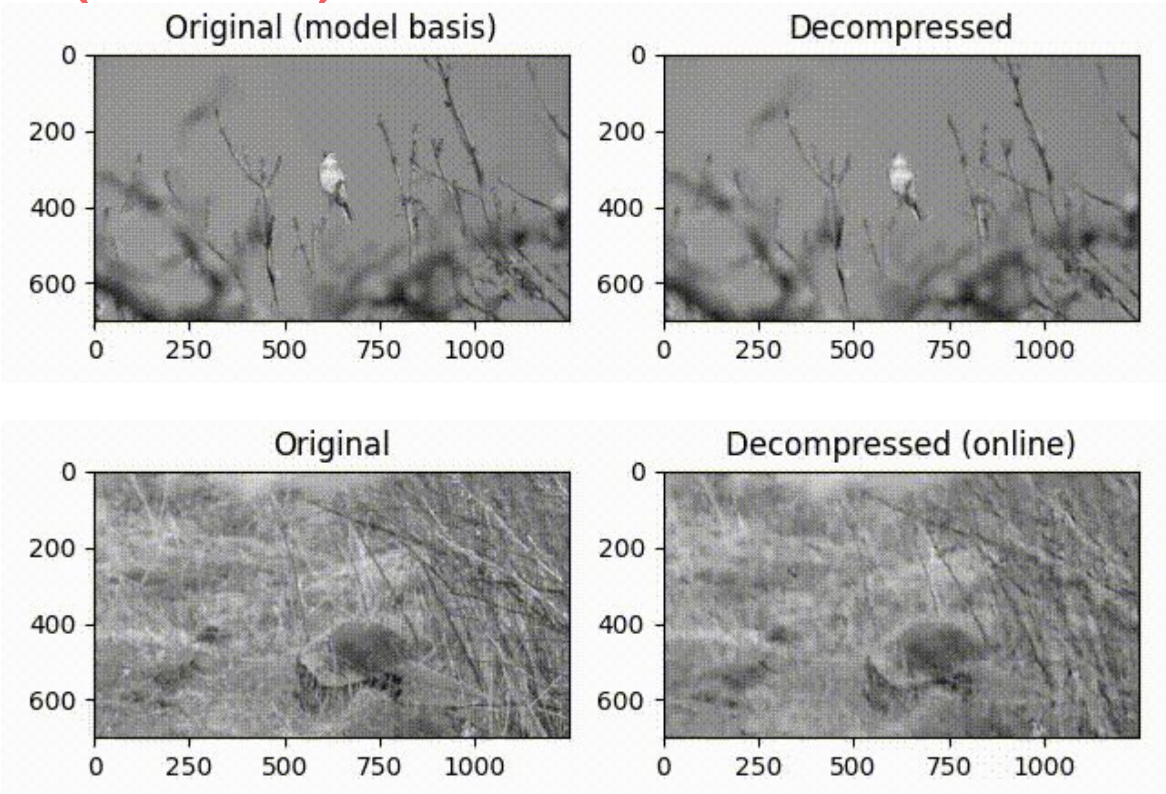
Online vs offline (X-Ray Diffraction)

- Previously applied model trained on one dataset to the same dataset (***offline***)
- Can also apply to similar but unseen datasets (***online***)
  - Eliminate the cost of the model size!
- Useful for compressing **live data** (triggers, networks, etc)



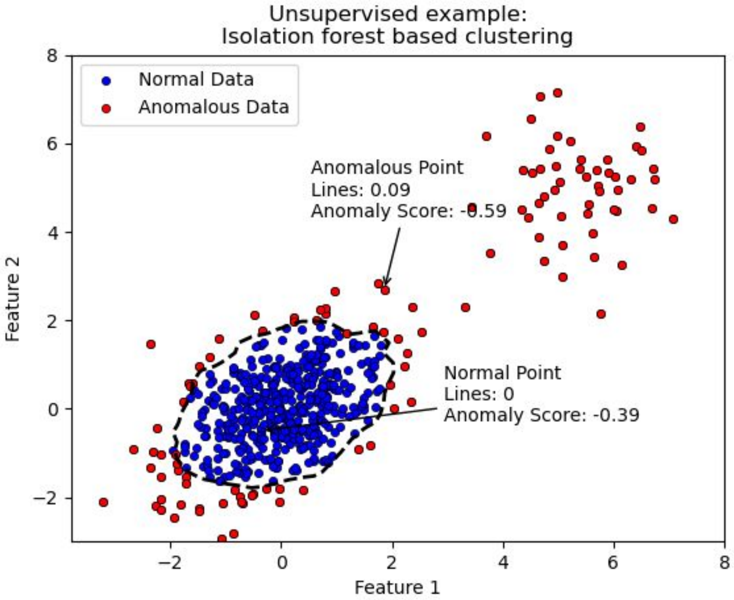
Online vs offline (Video)

- Previously applied model trained on one dataset to the same dataset (***offline***)
- Can also apply to similar but unseen datasets (***online***)
  - Eliminate the cost of the model size!
- Useful for compressing **live data** (triggers, networks, etc)



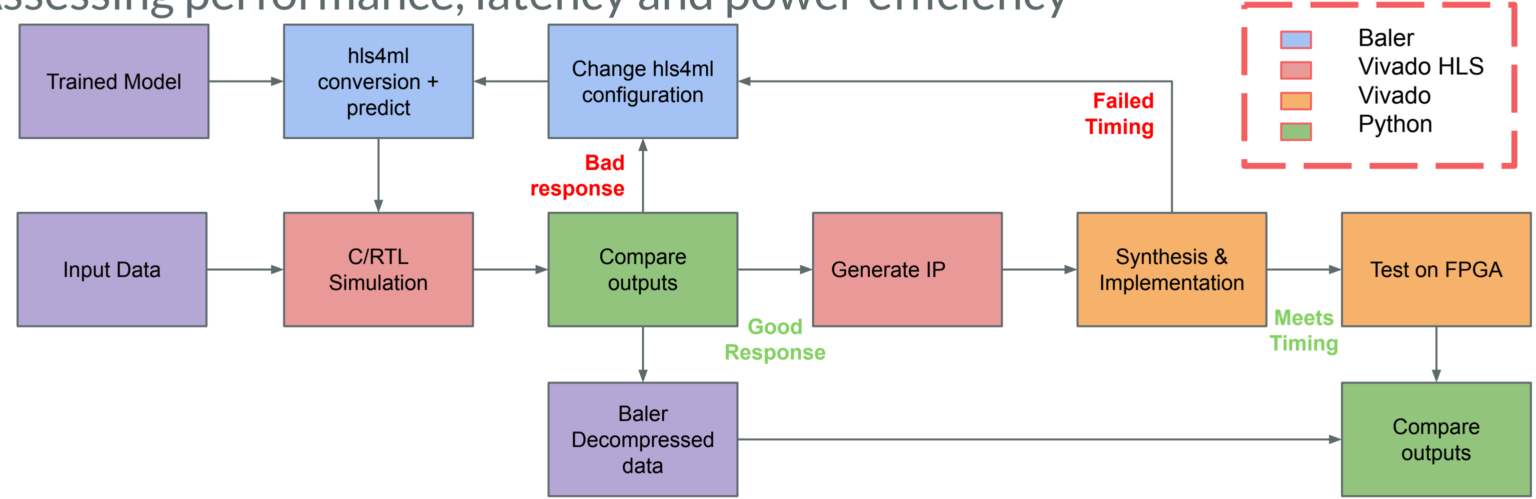
Anomaly Detection for Outlier Removal

- Online performance degraded by **outliers**
- Exploring use of **anomaly detection** to separate outliers
  - Outliers could be stored in full for further analysis
- Use a simplified version of BALER to build a **probability distribution** of points in latent space
- **Remove points** that significantly disagree, **iterate recursively**
- Performance **evaluation ongoing**



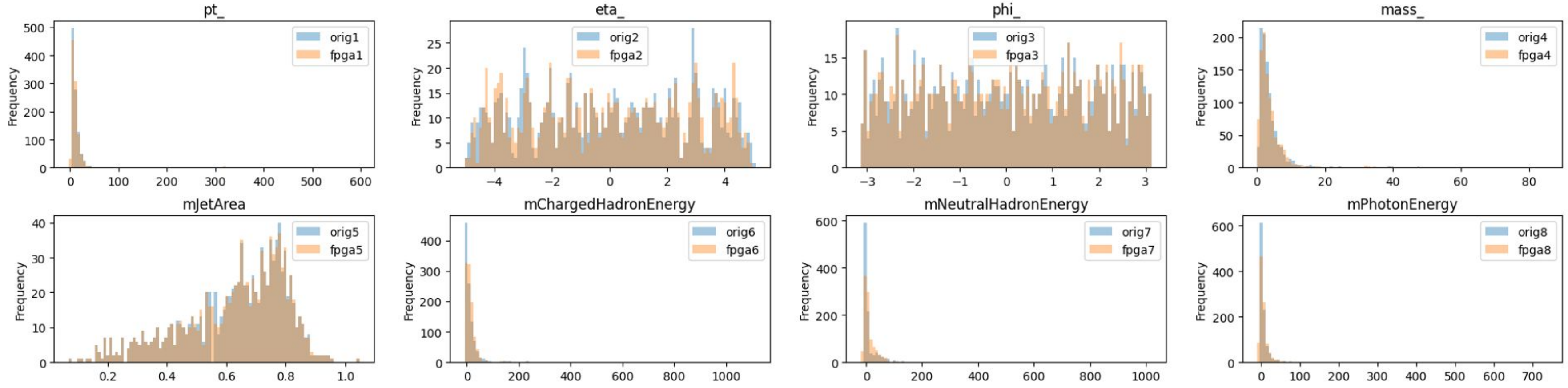
Baler on FPGA: Workflow

- Prototype version for developing and running BALER on an **FPGA**
  - Using vivado HLS code
- Useful in **bandwidth-restricted cases**
  - Network cards, detector readout, triggers, transmitters
- Assessing performance, latency and power efficiency



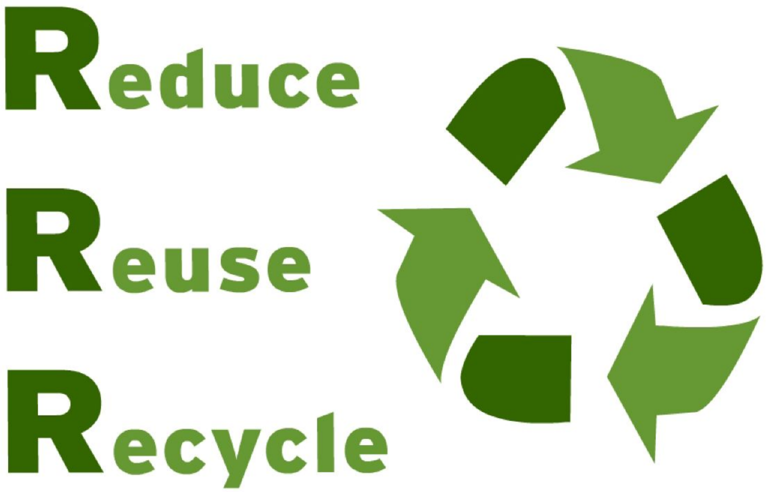
Baler on FPGA: Workflow

- Prototype version for developing and running BALER on an **FPGA**
  - Using vivado HLS code
- Useful in **bandwidth-restricted cases**
  - Network cards, detector readout, triggers, transmitters
- Assessing performance, latency and power efficiency



Software Sustainability

- Funded by **software sustainability** grants
- How can we improve climate impact?
  - **Reduce** software resource usage
    - Efficient software
    - Share cross-discipline expertise
  - **Reuse** software
    - Open-source
    - Well-written so it can be extended
    - Generic as possible
  - **Recycle** old software
    - Good documentation!
    - Good publicity
    - Preserve code and datasets (github, zenodo)



Summary

- **BALER** is a new toolkit for **compressing data** using **auto-encoders**
- Capable of **impressive** compression results, but requires saving a **large model**
- New developments targeting reusing models for **online lossy compression**
- New developments incorporating **anomaly detection** for outlier removal
- New developments targeting **FPGAs** for network or trigger applications

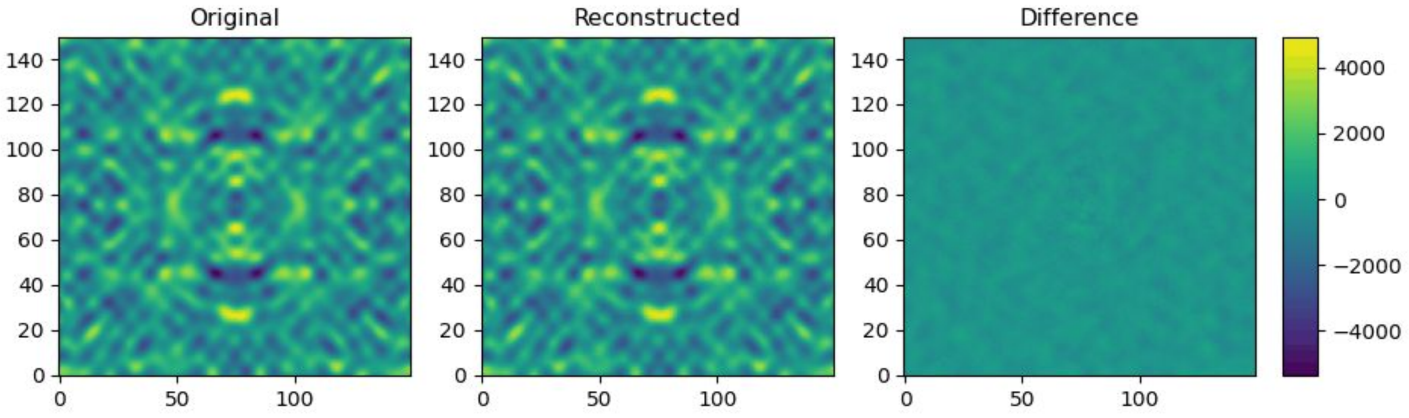
Interested? Contact us & Get involved!

- We are a friendly, cross-discipline team with significant involvement from **ECRs** and **industry**
- Master's and PhD projects very welcome and **can be supported**
- https://github.com/baler-collaboration/baler
- james.smith-7@manchester.ac.uk
- caterina.doglioni@manchester.ac.uk

Backup

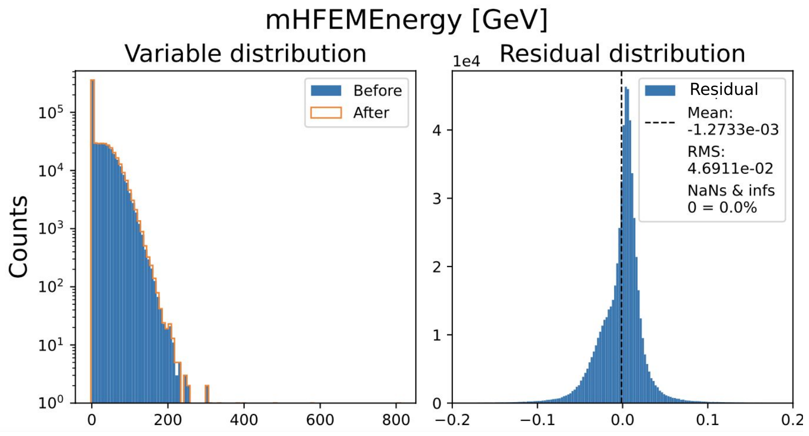
X-Ray Diffraction

- "4M simulated diffraction images of chaperone 3iyf"
  - In actuality 151x151x151 array, which I split into two 75x150x15 arrays
- Train on one half to compress down to 0.001% the original size
- Used for compression of the other half
  - Actually great performance

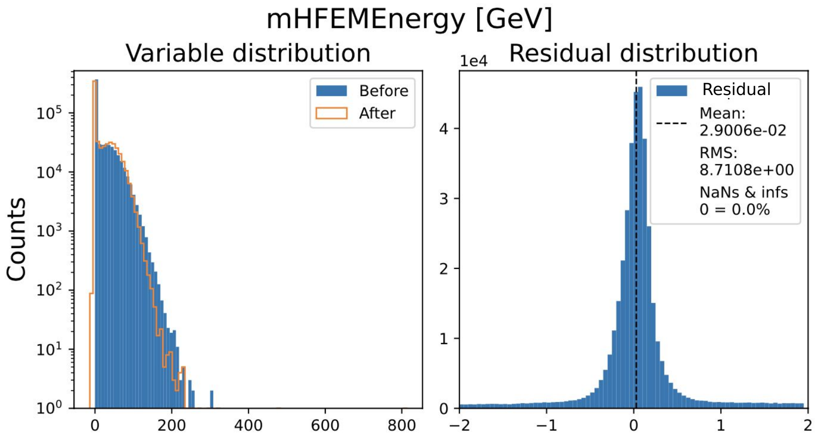


1.7x vs 6x compression

1.7x compression



6x compression

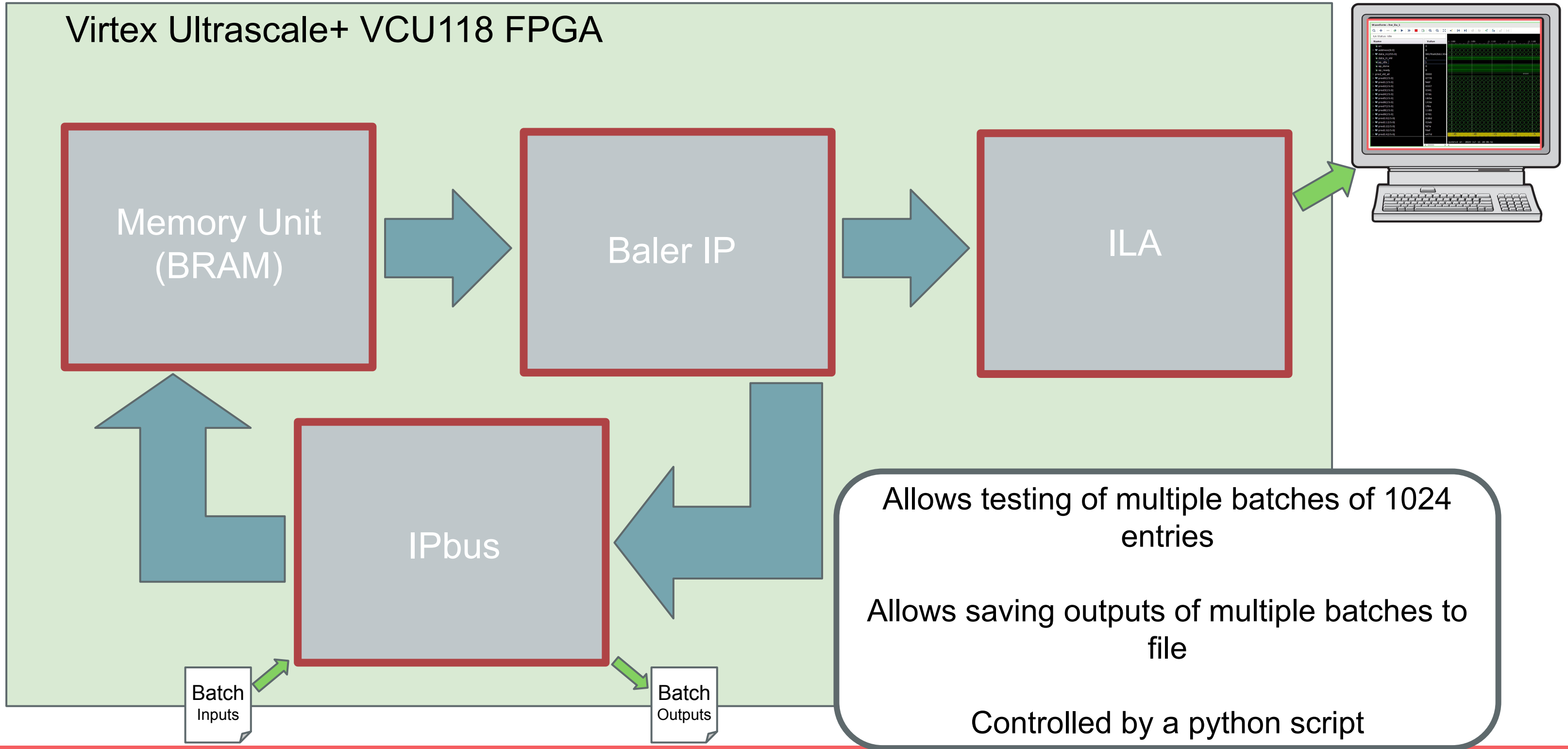


Full variable list (see https://arxiv.org/abs/2305.02283)

Table 2: Residual and Response distribution means and RMS values for all variables in the dataset. These values are presented at $R = 1.7$, and all values have been averaged over 5 runs, with an added statistical error of two standard deviations.

Variable ($R = 1.7$)	Response		Residual	
	Mean	RMS	Mean	RMS
p_T	$-1.07 \times 10^{-3} \pm 1.34 \times 10^{-2}$	$2.09 \times 10^{-2} \pm 3.56 \times 10^{-3}$	$-1.44 \times 10^{-2} \pm 1.04 \times 10^{-1}$	$2.12 \times 10^{-1} \pm 5.29 \times 10^{-2}$
η	$3.75 \times 10^{-4} \pm 6.11 \times 10^{-4}$	$8.12 \times 10^{-1} \pm 1.17$	$-1.12 \times 10^{-3} \pm 2.67 \times 10^{-3}$	$2.09 \times 10^{-3} \pm 1.45 \times 10^{-3}$
ϕ	$3.44 \times 10^{-4} \pm 8.64 \times 10^{-4}$	$1.93 \times 10^{-1} \pm 4.32 \times 10^{-1}$	$2.45 \times 10^{-4} \pm 1.80 \times 10^{-3}$	$9.91 \times 10^{-4} \pm 1.12 \times 10^{-3}$
mass	$2.39 \times 10^{-1} \pm 7.87$	$4.38 \times 10^{3} \pm 4.47 \times 10^{3}$	$-8.05 \times 10^{-3} \pm 2.51 \times 10^{-2}$	$3.98 \times 10^{-2} \pm 1.42 \times 10^{-2}$
mJetArea	$6.12 \times 10^{-5} \pm 1.81 \times 10^{-4}$	$3.13 \times 10^{-4} \pm 1.48 \times 10^{-4}$	$3.21 \times 10^{-5} \pm 8.90 \times 10^{-5}$	$1.10 \times 10^{-4} \pm 5.77 \times 10^{-5}$
mChargedHadronEnergy	$1.58 \times 10^{-3} \pm 1.70 \times 10^{-2}$	$2.85 \times 10^{-2} \pm 1.30 \times 10^{-2}$	$1.68 \times 10^{-2} \pm 1.43 \times 10^{-1}$	$1.71 \times 10^{-1} \pm 7.33 \times 10^{-2}$
mNeutralHadronEnergy	$7.05 \times 10^{-2} \pm 9.88 \times 10^{-2}$	$2.22 \times 10^{-1} \pm 6.59 \times 10^{-2}$	$2.77 \times 10^{-1} \pm 5.23 \times 10^{-1}$	$6.94 \times 10^{-1} \pm 2.26 \times 10^{-1}$
mPhotonEnergy	$-2.75 \times 10^{-2} \pm 7.48 \times 10^{-2}$	$6.84 \times 10^{-2} \pm 1.09 \times 10^{-1}$	$-8.00 \times 10^{-2} \pm 1.87 \times 10^{-1}$	$1.52 \times 10^{-1} \pm 1.77 \times 10^{-1}$
mElectronEnergy	$-7.71 \times 10^{-2} \pm 1.05 \times 10^{-1}$	$1.44 \times 10^{-1} \pm 7.47 \times 10^{-2}$	$1.71 \times 10^{-2} \pm 5.32 \times 10^{-2}$	$8.40 \times 10^{-2} \pm 4.15 \times 10^{-2}$
mMuonEnergy	$1.29 \times 10^{-2} \pm 1.97 \times 10^{-2}$	$8.04 \times 10^{-2} \pm 9.77 \times 10^{-2}$	$1.18 \times 10^{-2} \pm 1.46 \times 10^{-2}$	$3.15 \times 10^{-2} \pm 7.05 \times 10^{-3}$
mHFHadronEnergy	$-1.10 \times 10^{-2} \pm 4.66 \times 10^{-2}$	$1.77 \times 10^{-1} \pm 2.48 \times 10^{-2}$	$-3.15 \times 10^{-1} \pm 1.07$	$1.85 \pm 7.31 \times 10^{-1}$
mHFEMEnergy	$1.78 \times 10^{-3} \pm 7.40 \times 10^{-3}$	$1.41 \times 10^{-2} \pm 3.63 \times 10^{-3}$	$1.22 \times 10^{-2} \pm 8.26 \times 10^{-2}$	$6.93 \times 10^{-2} \pm 5.54 \times 10^{-2}$
mChargedHadronMultiplicity	$-1.00 \times 10^{-3} \pm 5.04 \times 10^{-3}$	$4.48 \times 10^{-3} \pm 4.90 \times 10^{-3}$	$-3.13 \times 10^{-3} \pm 1.82 \times 10^{-2}$	$9.68 \times 10^{-3} \pm 1.50 \times 10^{-2}$
mNeutralHadronMultiplicity	$-1.22 \times 10^{-4} \pm 1.29 \times 10^{-3}$	$8.76 \times 10^{-4} \pm 9.42 \times 10^{-4}$	$-1.19 \times 10^{-4} \pm 1.51 \times 10^{-3}$	$9.89 \times 10^{-4} \pm 1.20 \times 10^{-3}$
mPhotonMultiplicity	$-1.14 \times 10^{-3} \pm 3.62 \times 10^{-3}$	$2.72 \times 10^{-3} \pm 4.14 \times 10^{-3}$	$-2.69 \times 10^{-3} \pm 7.44 \times 10^{-3}$	$4.92 \times 10^{-3} \pm 7.12 \times 10^{-3}$
mElectronMultiplicity	$1.07 \times 10^{-3} \pm 3.87 \times 10^{-3}$	$2.37 \times 10^{-3} \pm 2.37 \times 10^{-3}$	$-1.54 \times 10^{-5} \pm 9.96 \times 10^{-5}$	$2.11 \times 10^{-4} \pm 1.75 \times 10^{-4}$
mMuonMultiplicity	$1.12 \times 10^{-3} \pm 1.22 \times 10^{-3}$	$2.51 \times 10^{-3} \pm 6.69 \times 10^{-4}$	$5.67 \times 10^{-5} \pm 1.16 \times 10^{-4}$	$2.41 \times 10^{-4} \pm 6.35 \times 10^{-5}$
mHFHadronMultiplicity	$-1.34 \times 10^{-3} \pm 1.84 \times 10^{-3}$	$2.53 \times 10^{-3} \pm 1.94 \times 10^{-3}$	$-2.67 \times 10^{-3} \pm 3.33 \times 10^{-3}$	$4.44 \times 10^{-3} \pm 4.05 \times 10^{-3}$
mHFEMMultiplicity	$2.41 \times 10^{-4} \pm 2.51 \times 10^{-3}$	$1.98 \times 10^{-3} \pm 1.33 \times 10^{-3}$	$5.98 \times 10^{-4} \pm 4.16 \times 10^{-3}$	$3.08 \times 10^{-3} \pm 2.95 \times 10^{-3}$
mChargedEmEnergy	$-7.72 \times 10^{-2} \pm 1.05 \times 10^{-1}$	$1.44 \times 10^{-1} \pm 7.48 \times 10^{-2}$	$1.72 \times 10^{-2} \pm 5.30 \times 10^{-2}$	$8.40 \times 10^{-2} \pm 4.15 \times 10^{-2}$
mChargedMuEnergy	$1.29 \times 10^{-2} \pm 1.97 \times 10^{-2}$	$8.05 \times 10^{-2} \pm 9.78 \times 10^{-2}$	$1.18 \times 10^{-2} \pm 1.46 \times 10^{-2}$	$3.15 \times 10^{-2} \pm 7.07 \times 10^{-3}$
mNeutralEmEnergy	$-1.73 \times 10^{-2} \pm 5.42 \times 10^{-2}$	$5.89 \times 10^{-2} \pm 8.87 \times 10^{-2}$	$-6.70 \times 10^{-2} \pm 2.57 \times 10^{-1}$	$1.75 \times 10^{-1} \pm 1.81 \times 10^{-1}$
mChargedMultiplicity	$-9.83 \times 10^{-4} \pm 5.04 \times 10^{-3}$	$4.46 \times 10^{-3} \pm 4.88 \times 10^{-3}$	$-3.07 \times 10^{-3} \pm 1.83 \times 10^{-2}$	$9.74 \times 10^{-3} \pm 1.51 \times 10^{-2}$
mNeutralMultiplicity	$-8.97 \times 10^{-4} \pm 1.42 \times 10^{-3}$	$1.56 \times 10^{-3} \pm 1.93 \times 10^{-3}$	$-5.36 \times 10^{-3} \pm 7.37 \times 10^{-3}$	$7.34 \times 10^{-3} \pm 6.60 \times 10^{-3}$

Vivado Project – (in progress)

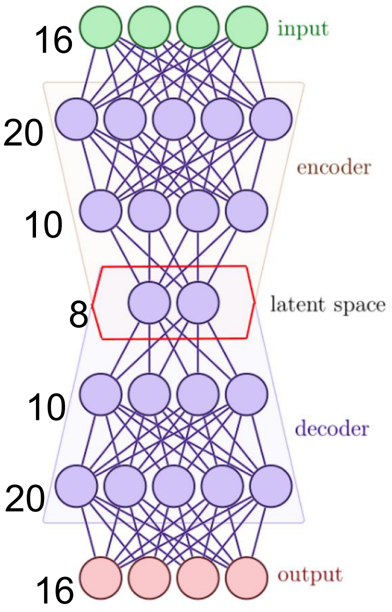


Allows testing of multiple batches of 1024 entries

Allows saving outputs of multiple batches to file

Controlled by a python script

Prototype Specifications



Synthesis Timing Estimation

Latency (cycles)		Latency (absolute)		Interval (cycles)		
min	max	min	max	min	max	Type
12	12	60.000 ns	60.000 ns	1	1	function

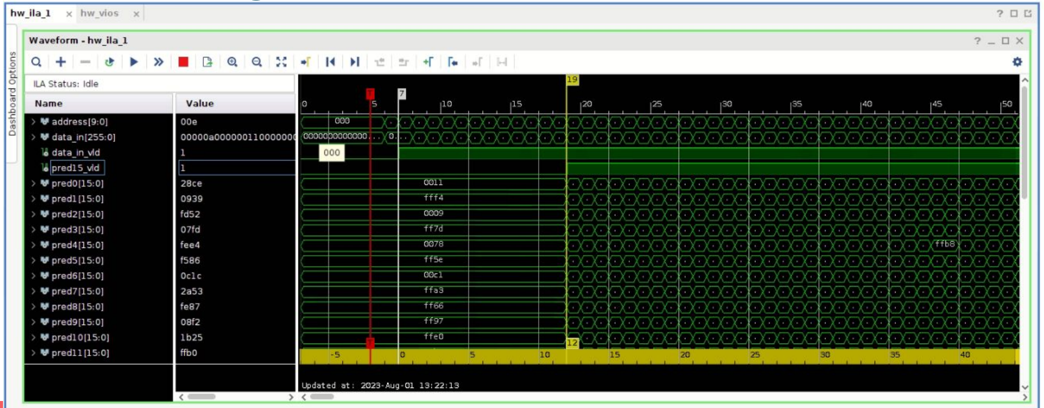
Resource Utilization

Messages | Log | Reports | Design Runs | Utilization | Timing | Power | Methodology | DRC | I/O Ports

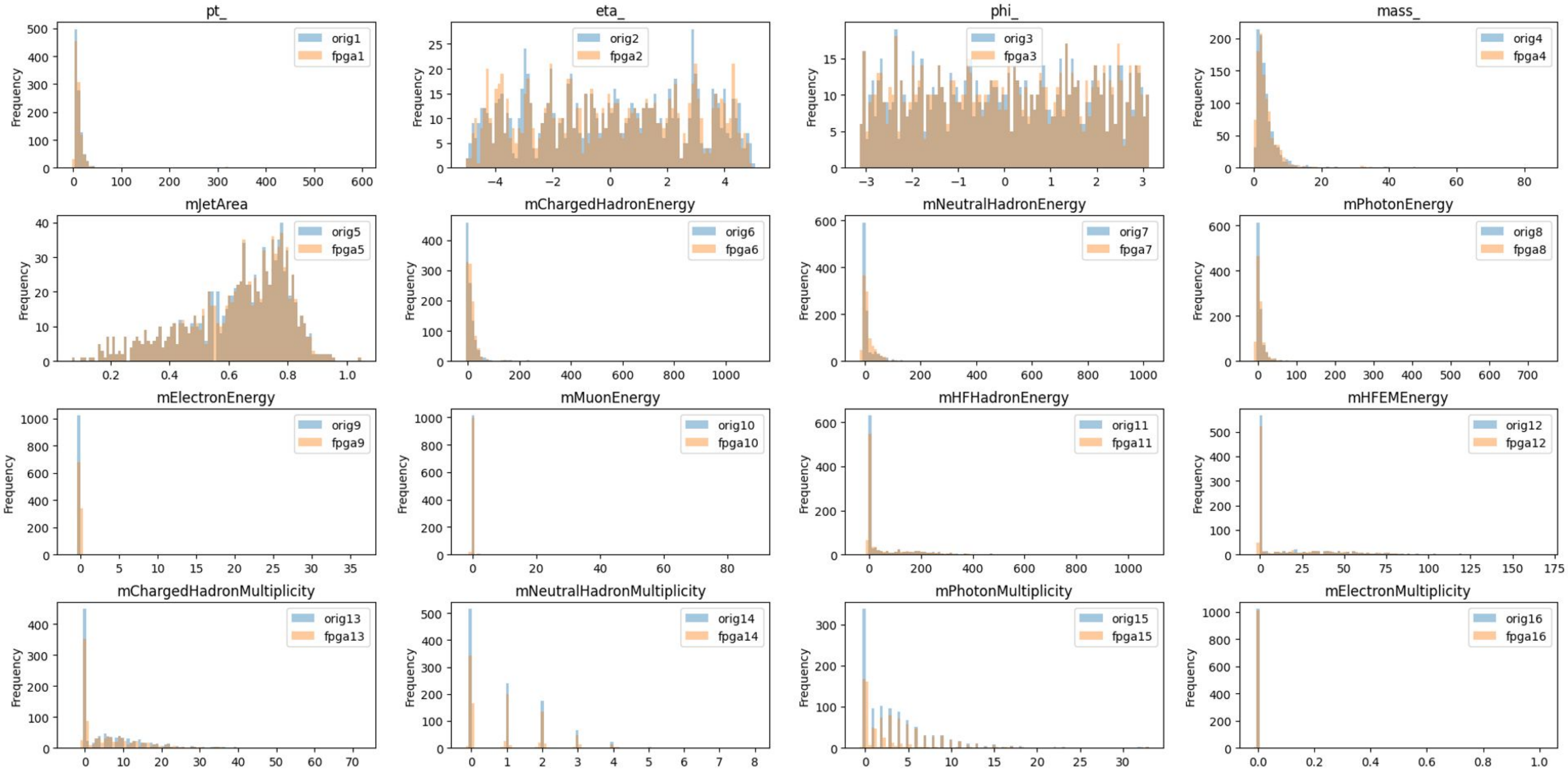
Hierarchy

Name	CLB LUTs (1182240)	CLB Registers (2364480)	CARRY8 (147780)	F7 Muxes (591120)	F8 Muxes (295560)	CLB (147780)	LUT as Logic (1182240)	LUT as Memory (591840)	Block RAM Tile (2160)	DSPs (6840)	Bonded IOB (832)	HPIOB_M (384)	HPIOB_S (384)	HPIOB DIFFINBUF (720)	GLOBAL CLOCK BUFFERs (1800)	MMCM (30)	BSCANE2 (12)
baler_top	24545	10229	2535	125	28	5129	23862	683	38	653	2	1	1	1	2	1	1
baler (tiny_model_0)	21889	4948	2462	0	0	4284	21889	0	0	653	0	0	0	0	0	0	0
clk_inst (clk_wiz_0)	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
dbg_hub (dbg_hub)	461	753	7	0	0	159	429	32	0	0	0	0	0	0	1	0	1
ila_inst (ila_0)	2080	4272	66	125	28	794	1429	651	30.5	0	0	0	0	0	0	0	0
mem_inst (blk_mem_	0	0	0	0	0	0	0	0	7.5	0	0	0	0	0	0	0	0
vio_inst (vio_0)	99	231	0	0	0	52	99	0	0	0	0	0	0	0	0	0	0

ILA Wave Diagram



Preliminary Results: Data vs FPGA



Preliminary Results: GPU vs FPGA