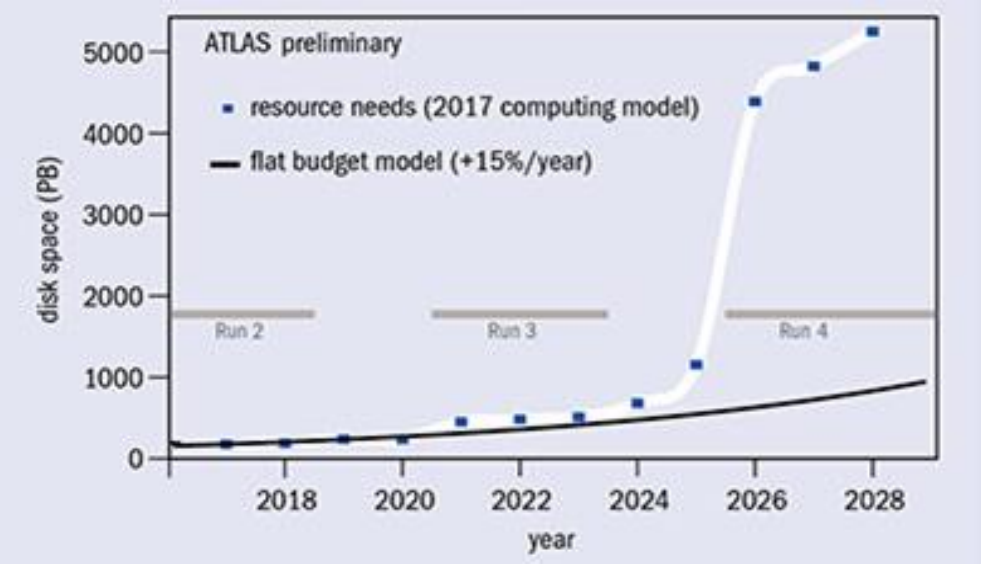
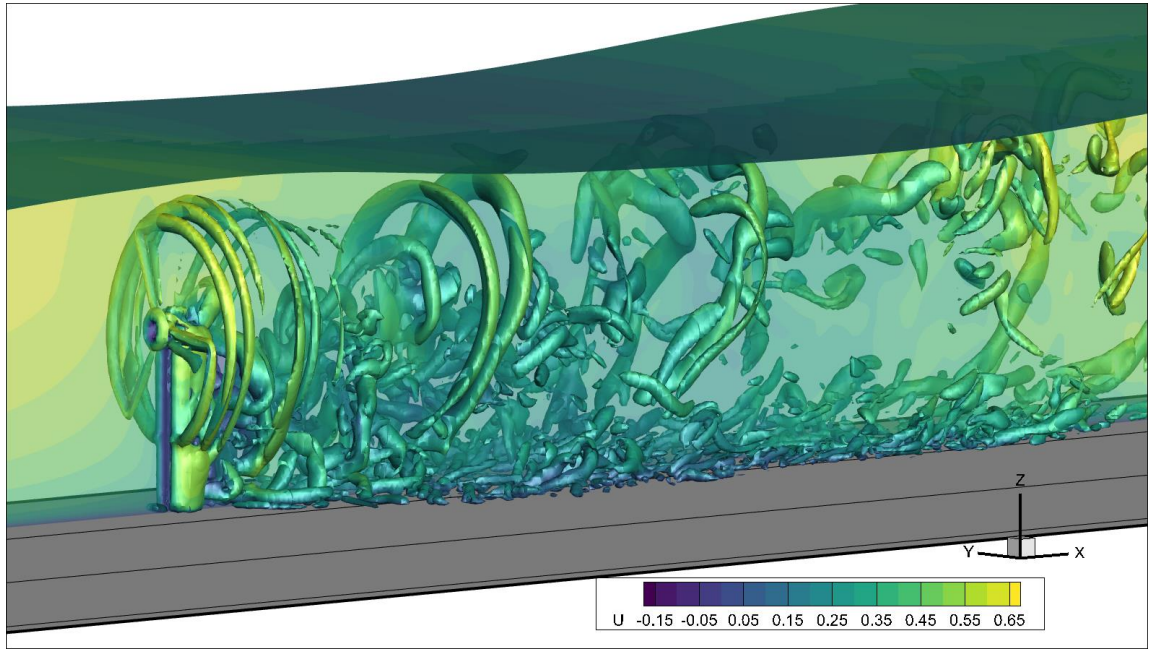
Case-by-case tailored data compression using machine learning algorithms trained on GPUs

Axel Gallén*, Alexander Ekman*, Marta Camps Santasmasas', Nicola Skidmore', Tony Lujia Chen'

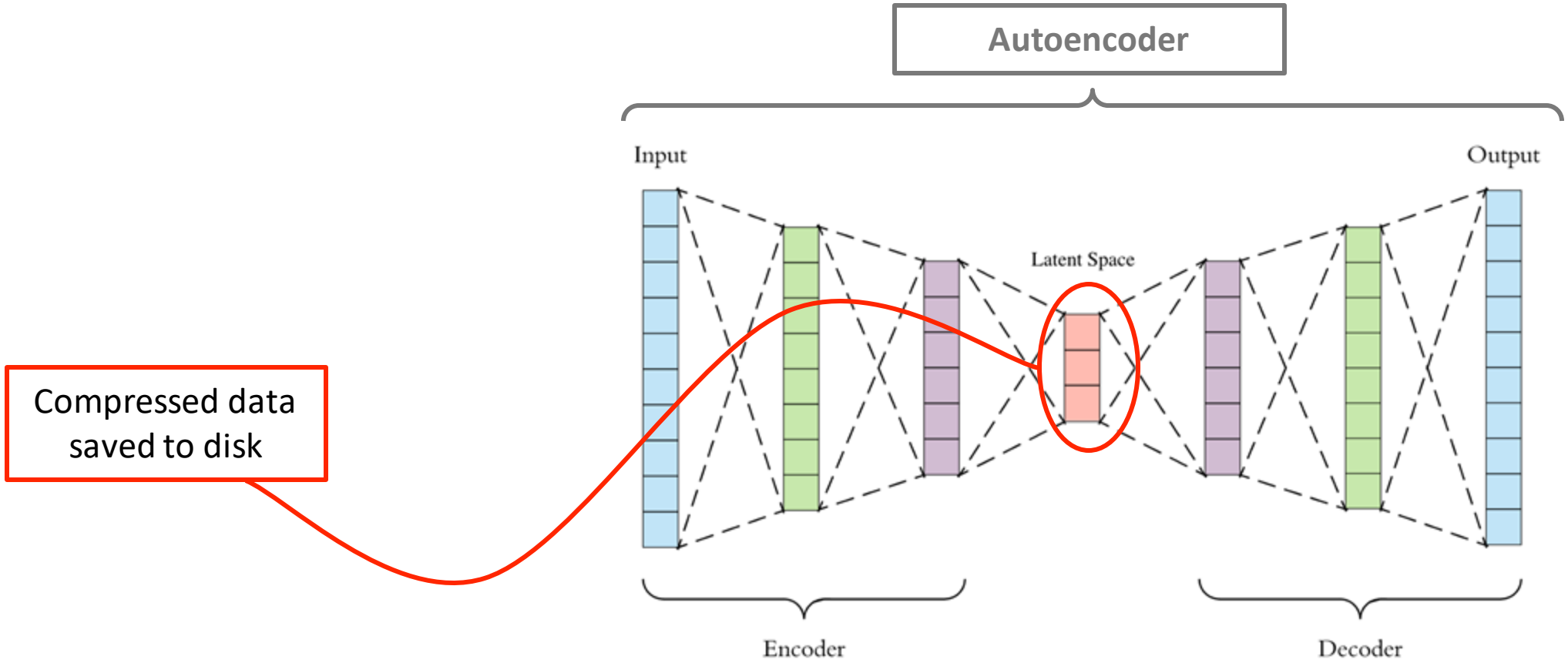
' University of Manchester, UK
* Lund University, Sweden





Lossy Machine Learning based compression

- Big size reduction at the cost of losing information.
- Lossy compression needs to be tailored to data type.
- **Autoencoder**: Neural network trained to optimally compress and decompress your data.

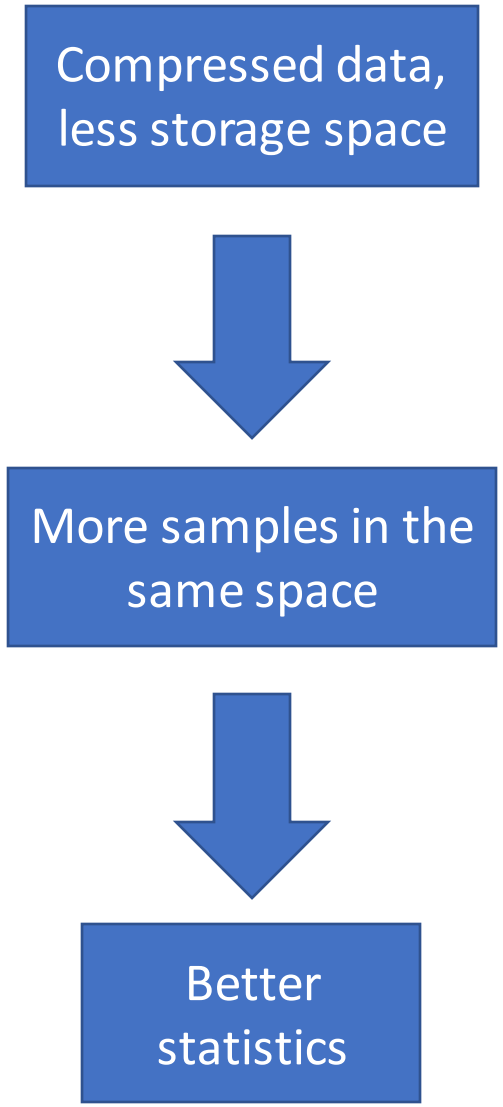


Autoencoder compression

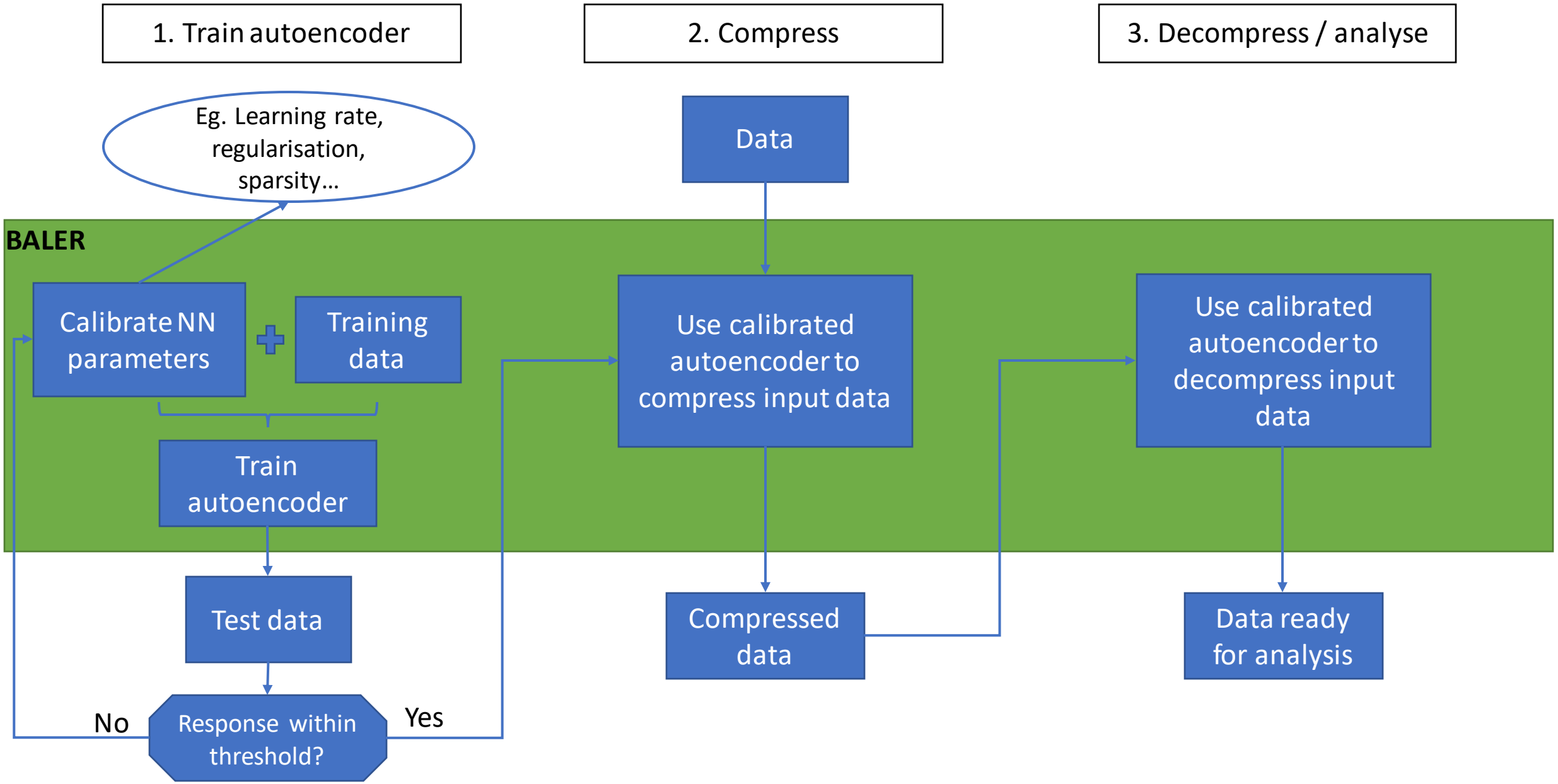
- Lossy compression comes with a price:
  - Decompressed data is not equal to original data
  - Lossy compression = bad ?

- **Works well** in cases where **more data is better**
  - For example: Particle physics
    - More data compensate for the loss

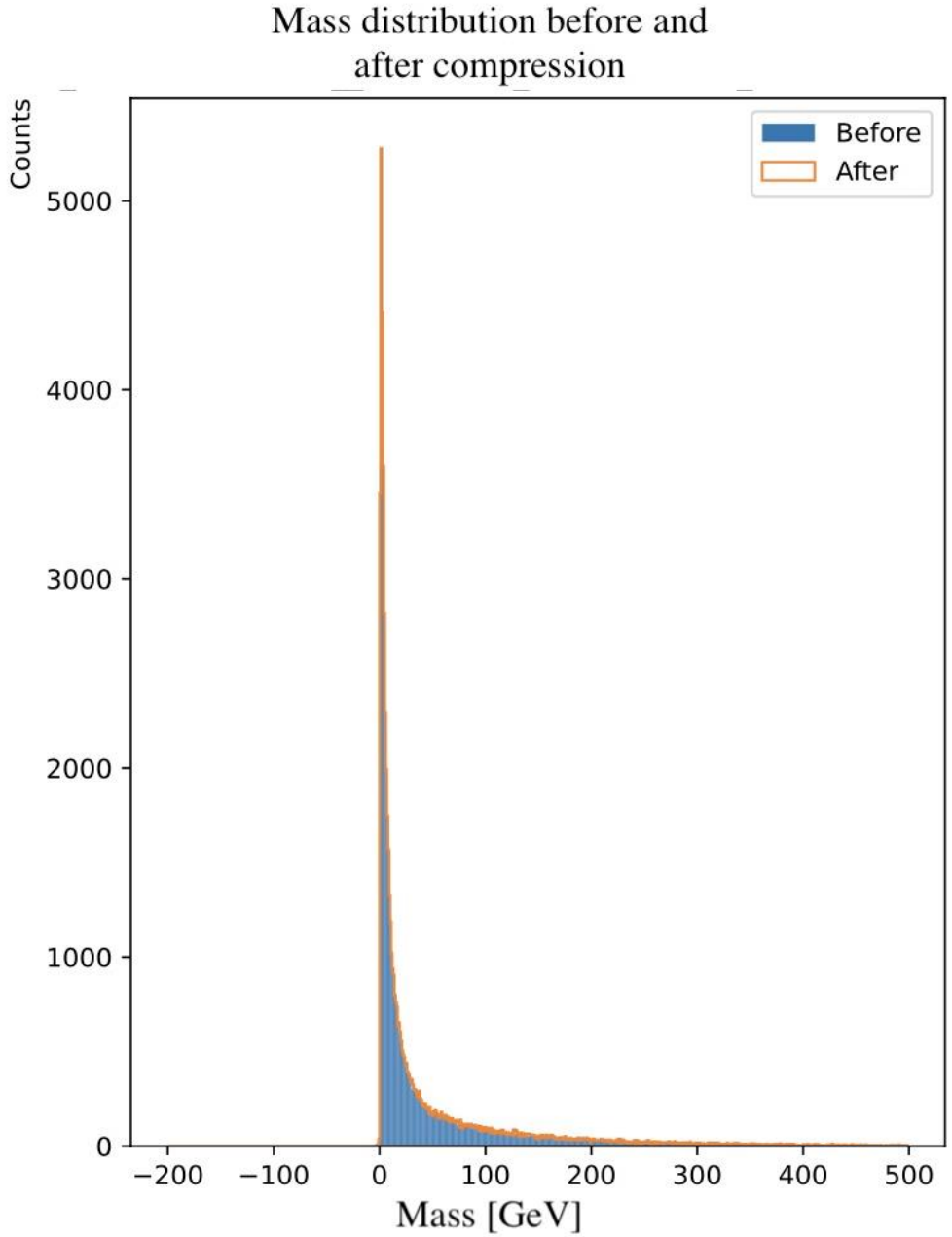
- **Current analysis** prototype and code available on the Virtual Research Environment (**VRE**) here

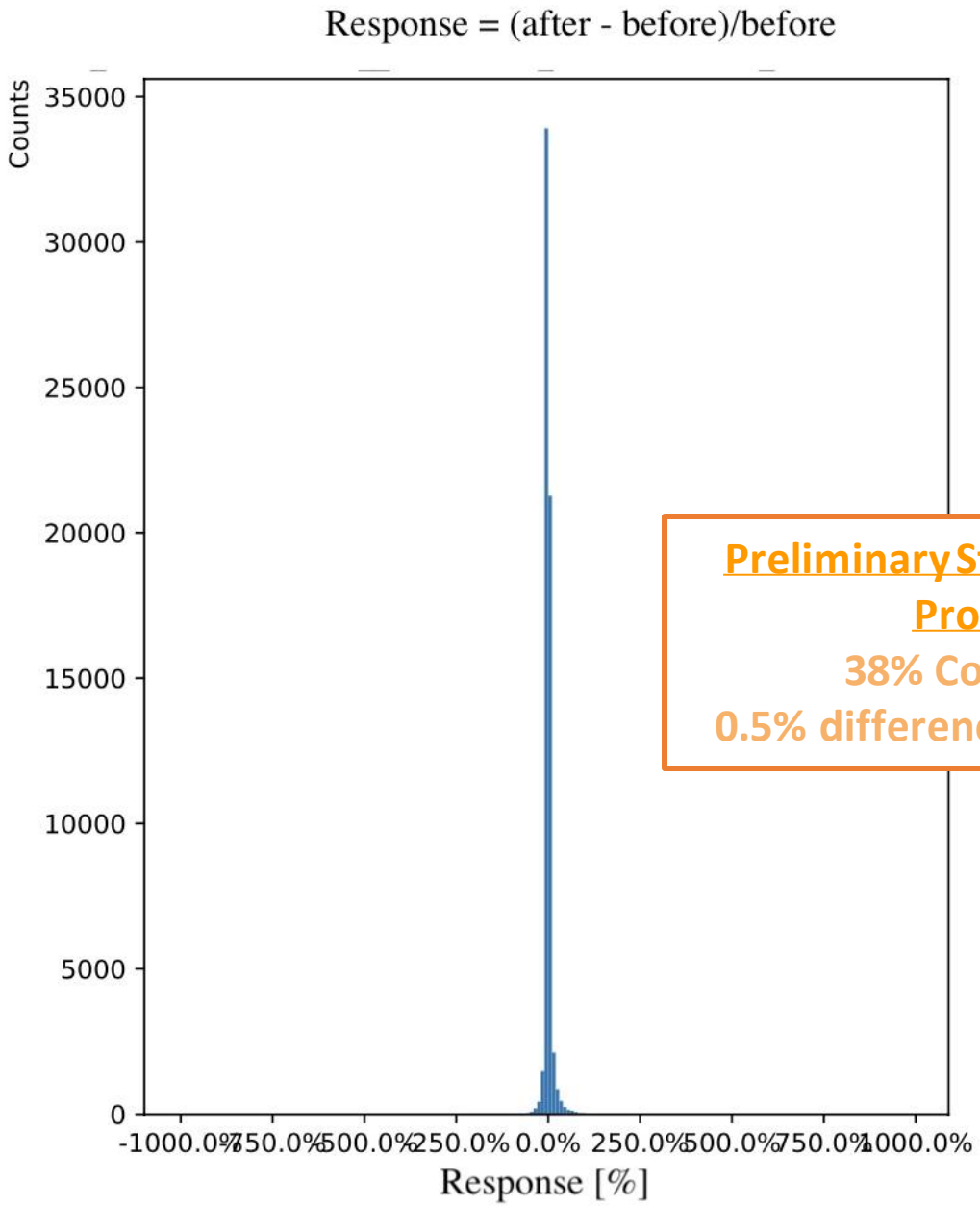


BALER: Python code. Train autoencoders, compress / decompress data.



BALER: example for particle collision data





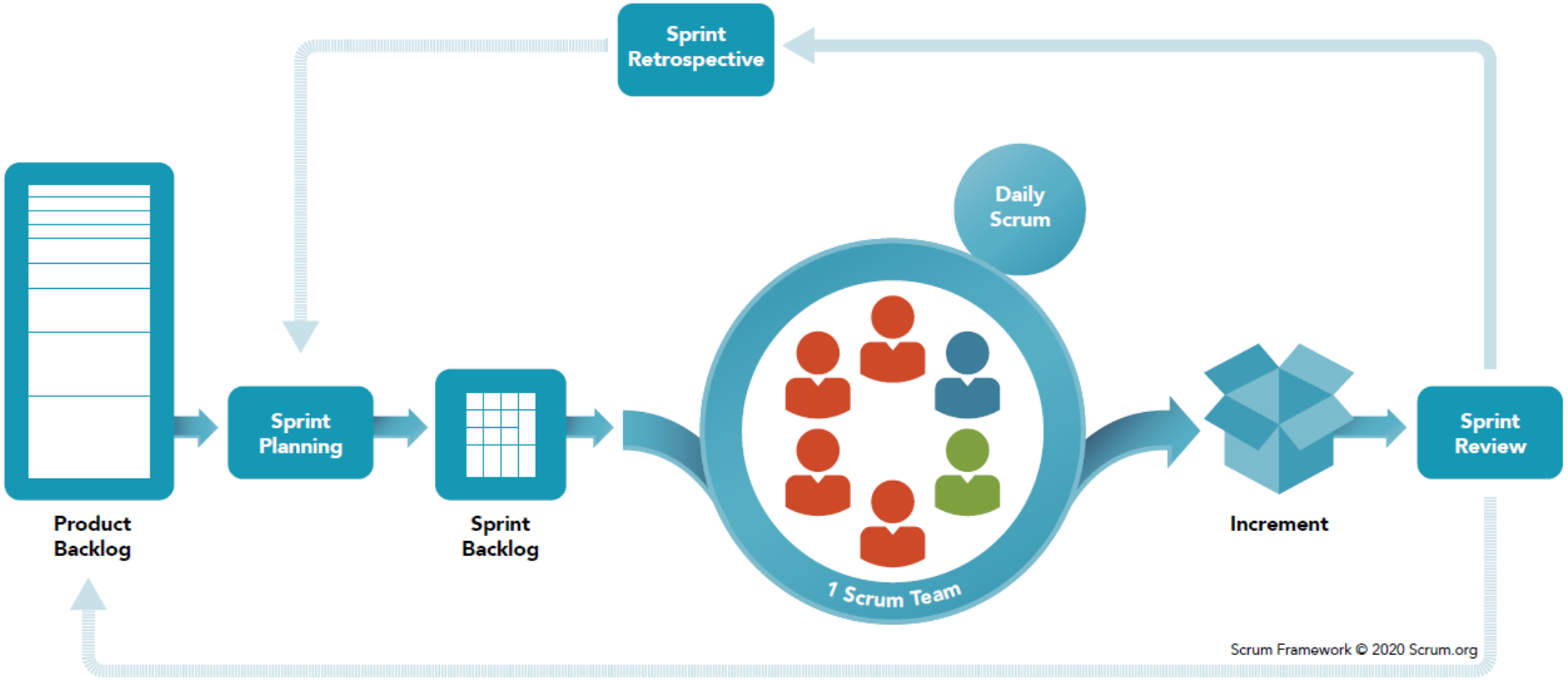
Preliminary Studies (From VRE Prototype)

38% Compression

0.5% difference to original data

BALER: Sustainable code development

- Development currently based on the **Scrum** method
  - Sprints which are 2 weeks long
    - Begins with Spring planning meeting
    - Ends with retrospective meeting
  - Good time planning & clear workload structure.
- Software management based on Jira management
  - Allows for easy-to-read backlogs
  - Integration with version control (Git in Bitbucket)
  - Easy to integrate new people to the project
- Modularity testing in progress



Objectives of the project: ongoing work

- Port BALER to GPU
  - Idle GPUs available, potential for faster compression.

- Package BALER for general usage
  - Researchers can apply BALER to their data. Benefit from lossy compression.

- Test case: Apply BALER to computational fluids dynamics data

Flow around a wall mounted cube