Deliverables

- Python scripts to compress and decompress the data using different compression algorithms analyzed during the project
- Scripts to produce the plots for different metrics
- Summary of the algorithms, documentation of the findings and points for future development.

Proposed Timeline

- Before May 4, 2020
    - *Familiarise myself with existing ATLAS code, understand ROOT I/O and its functionalities.*
- May 4 - May 17, 2020
    - Read up on ATLAS trigger and data formats, evaluate the existing compression algorithms, with hands-on experience if possible.
- May 18 - May 31, 2020
    - Learn about the critical factors concerning the environment, the planned experimental upgrades in 2026 and which factors regarding the ATLAS data compression could be relevant while designing the deep-compression algorithm.
    - Document and discuss the possible directions.
- June 1 - June 28, 2020
    - Experiment and analyze the existing deep-compression algorithms or networks to compress the ATLAS data *in the context of a resource-constrained system such as the ATLAS trigger*.
        - Start with running the 27-variable version of this https://github.com/Autoencoders-compression-anomaly/AE-Compression-pytorch - should be able to do this on your laptop
            - Let me know if you need input files, not sure they are there already
        - First run the network, the retrain the network on 80% of the data and test it on 20% (→ git issue to Erik Wallin)
            - How long does it take on laptop? Decide how fast we want to get other resources
        - Check that this reproduces the results in Eric Wulff's thesis (27 variables) ~~or Erik Wallin's thesis (15 variables)~~ in terms of
            - Response plots
                - (Compressed variable - Original variable)/original variable = compressed variable / original variable (-1)
                - Response ~= 1 if all goes well with the compression

        - Quadrant/correlation plots
        - Every time you make plots, save them somewhere safe with a good naming convention that lets us understand where they come from (make a Beamer talk so one can compare easily?)
            - Convention could be: date_testingTopic_networkCharacteristics_variable
    - Try to implement your normalization functions and check if there are improvements with respect to the baseline
        - Keep the current ones for the test with Erik W's thesis, then revisit for event-level
    - Loss function thoughts
        - For now, keep using MSE
        - Once we have the 27D network set up, can try Maurizio's or something that weighs 4-vector more than other variables
    - Document the findings and their performances for the 1st evaluation.
        - Evaluation is a google form
            - More complete presentation with work in this first month
                - Includes work that you've done to document the HTCondor/modifications to the code
            - Summary of results
- (Optional) Perform hyperparameter tuning for a selected set of architectures. This is time-consuming and hence, will be performed if time permits. [contingent to getting a bigger cluster]

June 29 - July 3, 2020 (1st evaluation)

IMPORTANT DEADLINE for ALL MENTORS: First Evaluations open June 29 and are DUE before Friday, July 3rd 18:00 UTC

- July 4 - July 19, 2020
    - Explore autoencoder architectures and develop a network that compresses full events rather than individual jets.
        - Option 1: do things with the current workflow, use ATLAS data (ttbar simulation) and change file to ttbar
            - Pros: straightforward
            - Cons: can't publish outside ATLAS
        - Option 2: do things with the current workflow but use PhenoML data
            - Pros:
                - can publish outside ATLAS
                    - This would be our paper on "how autoencoders work for compression of high energy physics

variables" using a sample dataset with different physics objects and physics processes
        - link with DarkMachines community (who want a paper on anomaly detection by the end of the summer, but that is not our main goal)
        - Could brute-force the anomaly detection side (just make a different plot that tells us "is this autoencoder good for anomaly detection, as well as compression?" - maybe the answer is no)
    - Cons
        - Requires more time (~1 week) for data wrangling starting from existing scripts to feed those to our network
    - Possible work plan:
        - Outputs:
            - Zenodo Jupyter notebook
            - Contribution to DarkMachines
        - Work to be done
            - [undergraduate student could help] Understand and document what LHCPheno variables are, by talking to DarkMachines people and to Caterina
            - [undergraduate student could help] Make a Jupyter notebook that reads in the data and prints it out in simple histograms, and has descriptions of each variable
            - This will be published on Zenodo and linked to the data
            - Take scripts from DarkMachines and adapt them for our network
                - Participate in discussions about how to read the entire data in memory / in chunks
    - Option 3: investigate more optimal way of data representation
        - E. g. binary representation learning, fairly standalone [Baptiste also interested] - start with Eric's network and compress to a number of ints
        - Followed by option 1
- July 20 - July 26, 2020
    - Document the results and conclusions obtained from the experiments for the 2nd evaluation.

July 27 - 31, 2020 (2nd evaluation)

- Aug 1, Aug 15, 2020
    - Develop a deep-compression network that works on entire events and can run in resource-constrained systems such as the trigger system, based on the previous months' findings. Analyze its performance.
    - (Optional) Explore the possibility of anomaly detection using the designed autoencoder-network.

- Aug 16 - Aug 23, 2020
    - Document the findings and write a white paper if needed.
- Aug 24 - Aug 31, 2020
    - Code submission and final evaluations.