20200901 - meeting with Caterina

Go through google checklist

- Public link: an example from 2017
    - https://summerofcode.withgoogle.com/archive/2019/projects/6581712404873216/
    - Will link to the PDF of the report
        - Think about Zenodo link with code version at the end
- Decide what to make public of the google drive
    - Index:
        - Project has taken place from June 2020 to September 2020
        - [google short project description]
        - [link to the project] - https://summerofcode.withgoogle.com/projects/#5677663735250944
        - [link to the report]
        - Index of files uploaded
            - Proposal that we had initially (and why we "strayed" from it: we wanted to test event-level before anything else)
            - Work log
            - Meeting log
            - Timeline and deliverables
            - HTCondor instruction + GPU
            - Full presentation
            - Presentation for OpenLab
            - Processed data and description
            - All the plots with a description of what the folders contain

Think about presentations

- Flash talk
- Others? Prepare a 15’ summary talk starting from OpenLab
    - Online conference on ML in particle physics → make an abstract
        - Deadline 18th of September
    - DarkMachines meeting on Fridays at 14:00 CERN time
        - Let’s see after this week / after the abstract has been submitted

Your CERNBox is in /eos/user/h/hgupta/

- Via the web interface, share with ‘doglioni’

20200814 - meeting with Caterina

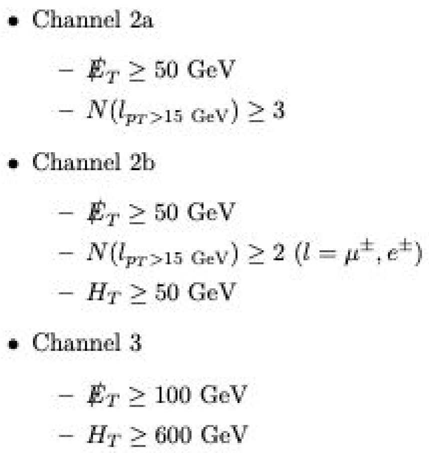
Plots:

- Things to add to new plots for the report
    - Add a version of the plots with the single particles overlaid instead of stacked

    - Add the overall mean / RMS on the plots
    - For each table, add a plot with the individual mean / RMS for each kind of particle

Next next:

- *Train on different particles other than jets and test on jets?*
    - If possible, use DarkMachines challenge data: Train on channel 2a and/or 2b to train as the “different” and then we test on channel 3 that is only jets



- Channel 2a
    - $\not{E}_T \geq 50$ GeV
    - $N(l_{p_T > 15 \text{ GeV}}) \geq 3$
- Channel 2b
    - $\not{E}_T \geq 50$ GeV
    - $N(l_{p_T > 15 \text{ GeV}}) \geq 2 \ (l = \mu^{\pm}, e^{\pm})$
    - $H_T \geq 50$ GeV
- Channel 3
    - $\not{E}_T \geq 100$ GeV
    - $H_T \geq 600$ GeV

    - If not possible (plan B) then use jet-trained network for channel 2a and/or 2b.
- *Start writing the final report for evaluation*
    - *Document findings and physics insights in a final report (3-4 pages)*
        - *On github in latex*
    - *Code release with instructions to reproduce most important plots*
- *If time allows, make an example network that uses all particles and maybe train it once → can be done by the next Master’s student*
    - *Not just 4 → 3*
    - *Network should use also the particle “label” = instead of having 4 variables, have 4 * number of particle variables, use 0-padding*
- *Evaluation 24th of August - submit the final project*

20200807 - meeting with Caterina, Simona

Looking at slides 55 onwards: https://docs.google.com/presentation/d/12_yRCl63H1VElDejEUteBqwXalWs0b9x3YcoKGFkkZA/edit#slide=id.g8dc56413ea_0_73

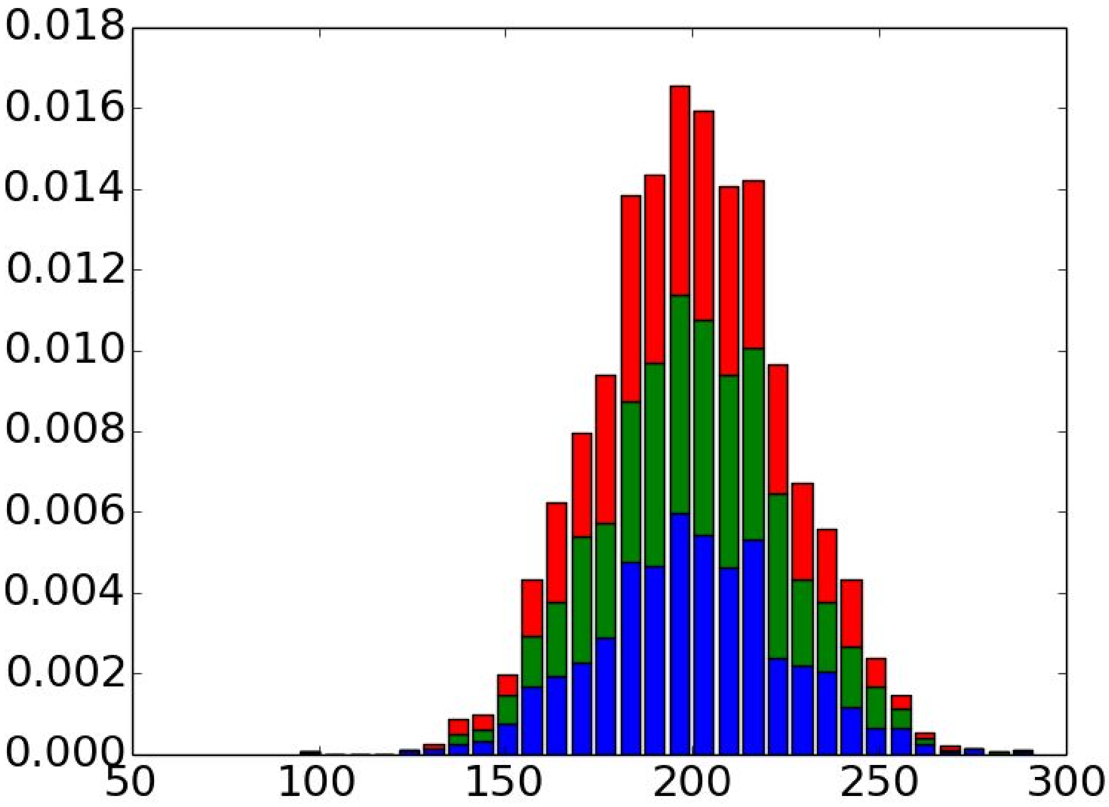
[make a meme about cake and jets] it’s all jets

→ most of the physics processes we’re using have many jets in the final state (80-90%), so it’s very likely our network will do well because it learned jets

How do we do with other kinds of particles?

Make a “stack” plot of the response after passing the samples through the network trained on jets where each kind of particle is a separate histogram (of a different color) and we stack them (https://stackoverflow.com/questions/18449602/matplotlib-creating-stacked-histogram-from-three-unequal-length-arrays)

Plot the particles with the least percentages first, so we plot with a logarithmic y scale and those are highlighted.



Which datasets to make these plots (+ inputs to the network)?

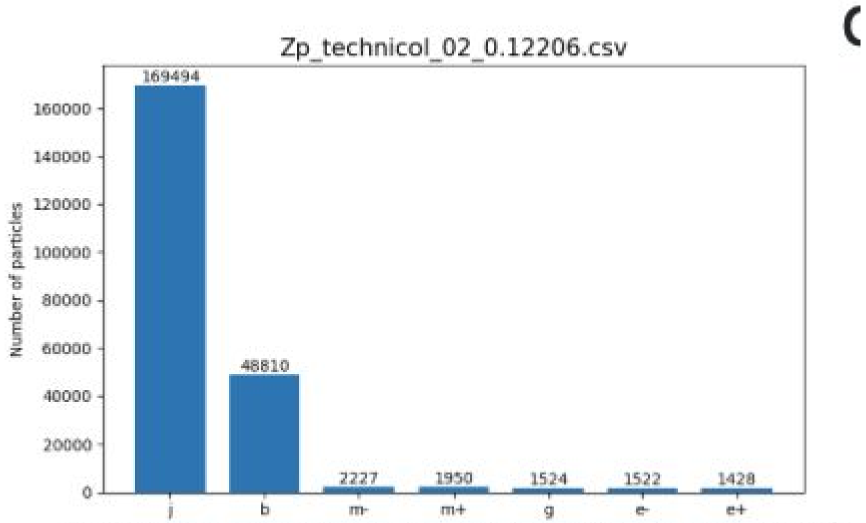
- The ones you already have because they’re already there
- The different DarkMachines channels (→ will need to be put through the network)
    - Should be smaller than the ones that we already have
    - https://zenodo.org/record/3961917#.Xy0XZS-ZN3M channels 2a/2b (with leptons) and 3 (mostly jets)

From the last meeting:

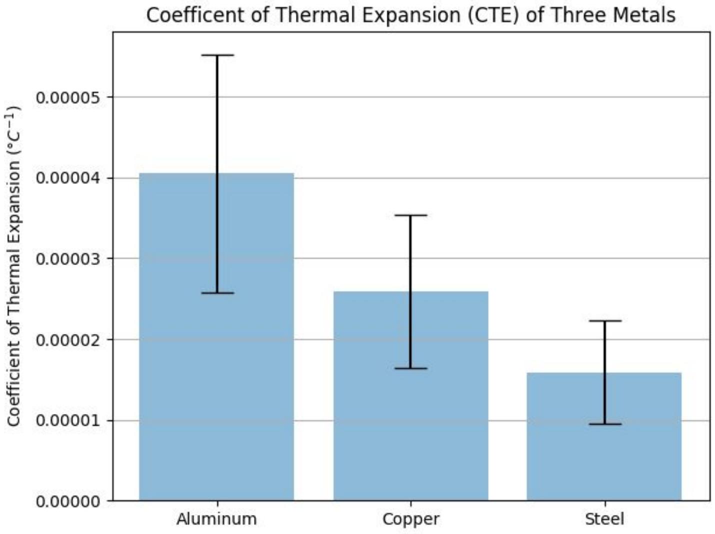
During this process, think about:

- *Change eta and phi to difference instead of residual [remake the ones that are already there later as a lower priority]*
- *Automating as much as possible*

- *Making summary plots (we will make those off the individual distributions for the variables of each particle:)*
    - *Y axis: mean or resolution of a certain variable for a given dataset*
    - *X axis: kind of particle*



```
{'j': '74.68', 'b': '21.51', 'm-': '0.98', 'm+': '0.86', 'g': '0.67', 'e-': '0.67', 'e+': '0.63'}
```

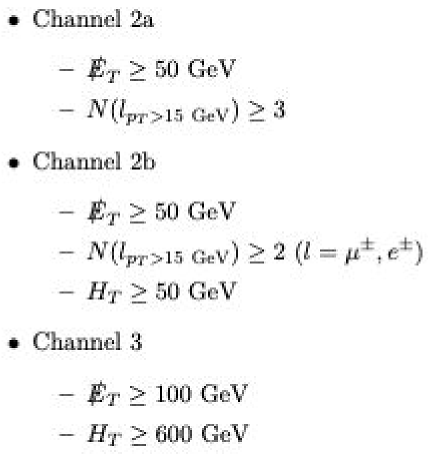


- *[optional] Next iteration: Fitting the resolution for summary plots? Or using the [IQR variable]*
- *[optional] If some particles are outliers, look at them and understand why (make individual plots for the kinematics of these particles) and thinking of anomaly detection score [after the stack plots]*

Next next:

- *Train on different particles other than jets and test on jets?*
    - Use DarkMachines challenge data

    - Train on channel 2a and/or 2b to train as the “different” and then we test on channel 3 that is only jets



- Channel 2a
  - $\not{E}_T \geq 50$ GeV
  - $N(l_{p_T>15\ \mathrm{GeV}}) \geq 3$
- Channel 2b
  - $\not{E}_T \geq 50$ GeV
  - $N(l_{p_T>15\ \mathrm{GeV}}) \geq 2$ $(l = \mu^{\pm}, e^{\pm})$
  - $H_T \geq 50$ GeV
- Channel 3
  - $\not{E}_T \geq 100$ GeV
  - $H_T \geq 600$ GeV

    - *Make an example network that uses all particles and maybe train it once*
        - *Not just 4 → 3*
        - *Network should use also the particle “label” = instead of having 4 variables, have 4 * number of particle variables, use 0-padding*
- *Start writing the final report for evaluation*
    - *Document findings and physics insights in a final report (3-4 pages)*
        - *On github in latex*
    - *Code release with instructions to reproduce most important plots*
- *Evaluation 24th of August - submit the final project*

Big picture: https://www.pnas.org/content/116/28/13825 - https://www.sciencedaily.com/releases/2019/06/190626133800.htm → she is trying to understand how/why it works and getting physical laws out of that

20200728 - meeting with Caterina, Baptiste, Simona

Looking at slides 36 onwards: https://docs.google.com/presentation/d/12_yRCl63H1VEIDejEUteBqwXalWs0b9x3YcoKGFkkZA/edit#slide=id.g8dc56413ea_0_73

Conclusions:

- Maybe need to look at how to load more data in memory, not so far
    - For now use “full dataset” network to train and use that network to test others
- Network performs well on “jetty” signals
    - top/antitop quark+gamma
    - ttbar

Next: stress-test the network with non-jetty things

- SM (plot/write the % of particle of each kind per dataset)
    - Zz_10fb
    - Zw_10fb
    - ww_10fb

        - 4top
        - Single_higgs_10fb.csv [ask Melissa about it]
    - BSM signals? [something with leptons]

During this process, think about:

- Automating as much as possible
- Making summary plots
    - Y axis: mean or resolution of a certain variable
        - When plotting the sigma (=resolution), plot sigma/mean for pT/energy
        - Change eta and phi to difference instead of residual
    - X axis: kind of dataset
- Fitting the resolution for summary plots? Or using the [IQR variable]
- Looking into the outliers and thinking of anomaly detection score

Next next:

- Have a mix of different signals and train on that, then test on all these samples
- Add more particles while training

20200721 - meeting with Caterina

Looking at slides 32 onwards:
https://docs.google.com/presentation/d/12_yRCl63H1VEIDejEUteBqwXalWs0b9x3YcoKGFkkZA/edit#slide=id.g8dc56413ea_0_73

Normalization, training from scratch the 4D models.
Custom:

- data['eta'] = data['eta'] / 5
- data['phi'] = data['phi'] / 3
- data['E'] = np.log10(data['E'])
- data['pt'] = (np.log10(data['pt'])

Custom normalization performs better (more tails in E/pT for standard normalization)

Training with whole or half the training set:

- Training set bigger = takes more than a day to train it, and then the job quit…
- Half the training set: the models with a smaller training set seem to have a better performance, but we can’t see the MSE loss as the plots weren’t saved for the job that quit
- Is something like this happening (overfitting)?
    - Usually “more data” fixes it but we already have it

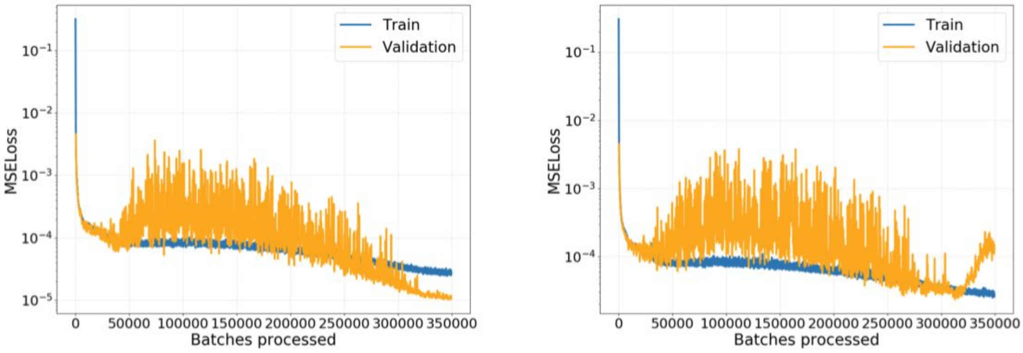


Figure 2.14: Comparison of two training runs of the 18D latent space AE using the same hyperparameters as the best performing network from the grid search in Fig. 2.13. (Left) The results from a NN that achieved the same end performance as the NN from the grid search. (Right) The network that resulted in roughly twice as high validation loss.

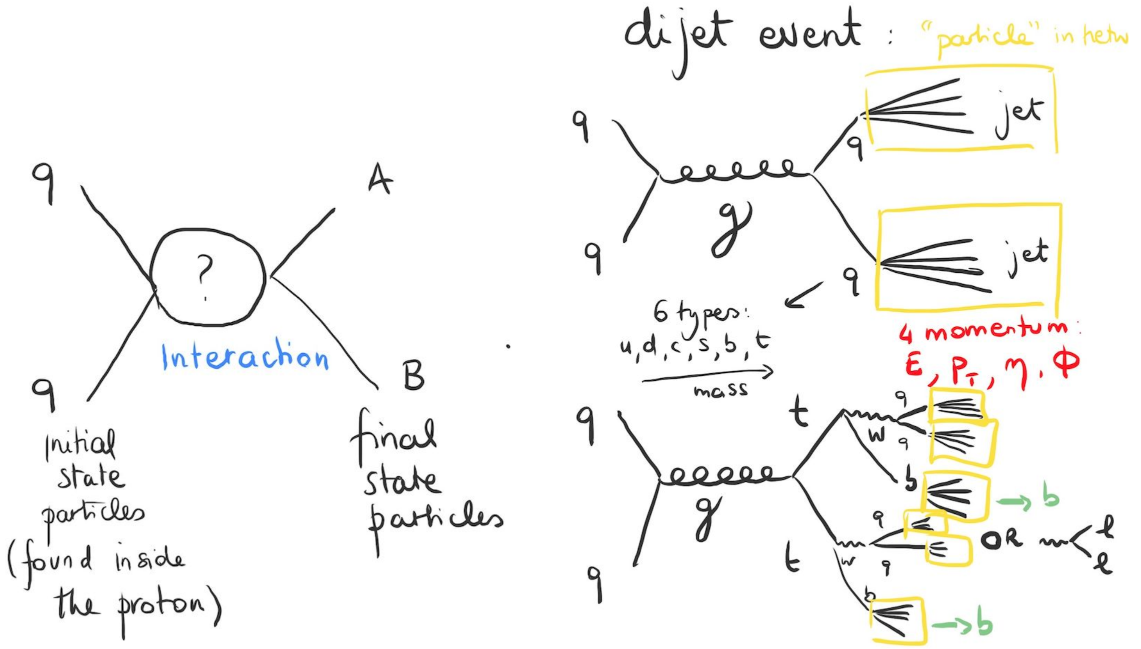
- Now it’s training for 2 days, and will modify the code to save the plots every 5 epochs
- Next thing to understand: why does half training set have better performance than full training set?

Next steps (copied from last time):

- ✔️Try normalization after 20 GeV cut
- ✔️Train and plot the 4 → 3 network with half the dataset
- [investigating] Reduce the size of the dataset that is loaded to the minimum (check MSE loss with small dataset, then bigger dataset)
- After that
    - Use the same “dijet trained” network on a sample that is similar enough to jets [to check the physics process dependence of the network]
        - Use ttbar as test
            - First step: remove events where there is at least one lepton (so we select only jet decays of the top)
            - Second step: add back in those events, but don’t look at the lepton
            - Look at the leptons as well in this sample
        - Gluino (3 events??)
    - If performance not good compared to jets only, then retrain
- After that
    - Add one category of particle in the samples we know
        - Jets → add photons as objects (‘b’, ‘j’, ‘g’)
        - ttbar → add leptons as objects (‘b’, ‘j’, ‘e’, ‘mu’)
- After that
    - Mix all other physics processes and see what happens in using a network trained on jets / using a network trained on all physics processes
- Whenever Simona generated the “weird jet sample” try the network on this as well
    - If it compresses well, we are happy because we can apply the compression algorithm to the trigger and still retain those events

- If it compresses badly, we are also happy because we would be able to find those events in the trigger and put them away (anomaly detection)

Difference between ttbar and jet events:


q
A
?
Interaction
q
B
initial state particles (found inside the proton)
final state particles
dijet event : "particle" in hetw
q
jet
q
g
q
jet
q
6 types: u,d,c,s,b,t
mass
4 momentum: E, P_T, η, φ
q
t
W
q
q
g
b
→ b
q
t
W
q
OR
l
l
q
b
→ b

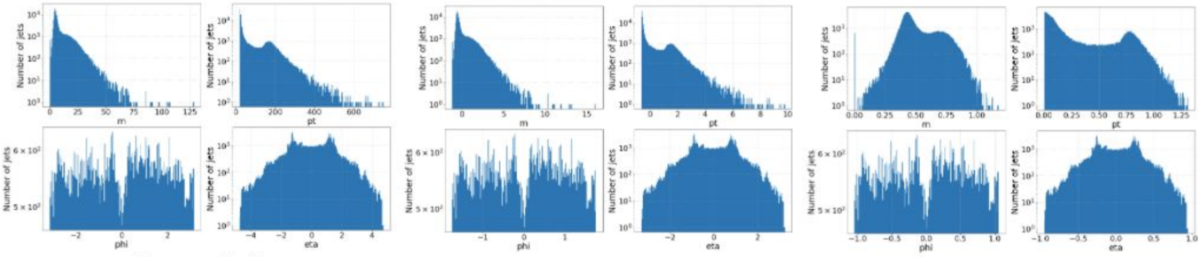
Last 10 minutes: tried https://openlab-fpp.web.cern.ch/ibmminsky/ibmminsky/ and found CD can't access the ATLAS folder yet, so wrote on the ticket here: https://cern.service-now.com/service-portal?id=ticket&table=u_request_fulfillment&n=RQF1603054

20200714 - meeting with Caterina, Simona, Baptiste

Honey:

- Started working on PhenoML data, how to read it and how to train it
    - Using pp->jj
- Yesterday, some problems on how to read it but now it works.
- Plots are here: [work log]
- Each event has has data in the format below
    - Label + 4 variables
- Code for reading reads the whole event and zero pads everything else.
    - Max number of particles for one event was 10, so it creates 10 columns
    - We're doing something different: we only want to look at the 4 variables of the jets and b-jets.
    - Juypter notebook: process_data_as_4D
        - Leaving the label as well, stored as metadata in a different file
        - Also storing event weight etc for later
        - The data is looking like the "GSOC evaluation" dataset

- Normalization: in the data we have, we have in the past used custom normalization
    - Custom norm for AOD data -
        - data['eta'] = data['eta'] / 5
        - data['phi'] = data['phi'] / 3
        - data['m'] = np.log10(data['m'] ~~+ 1) / 1.8~~
        - data['pt'] = (np.log10(data['pt']) ~~- 1.3) / 1.2~~
    - Plots: 10k sample
    - Previous dataset:

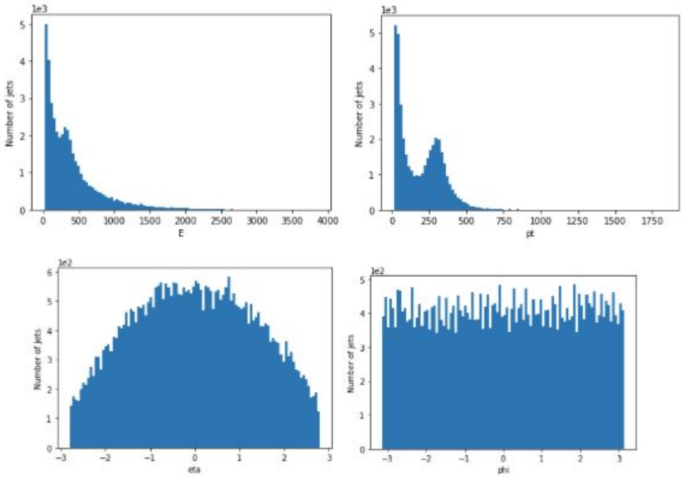


No normalization

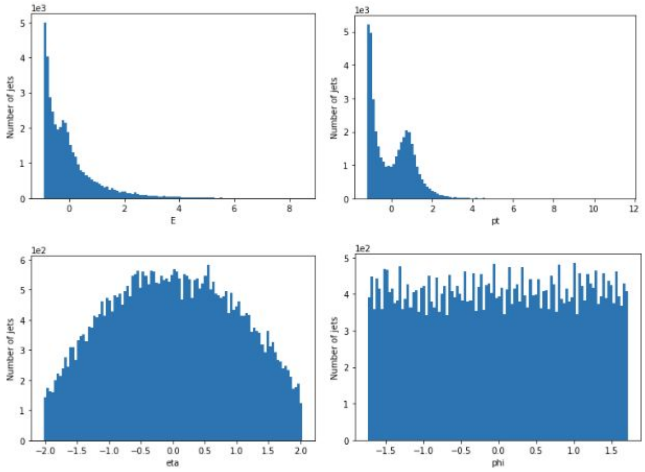
Standard normalization

Custom normalization

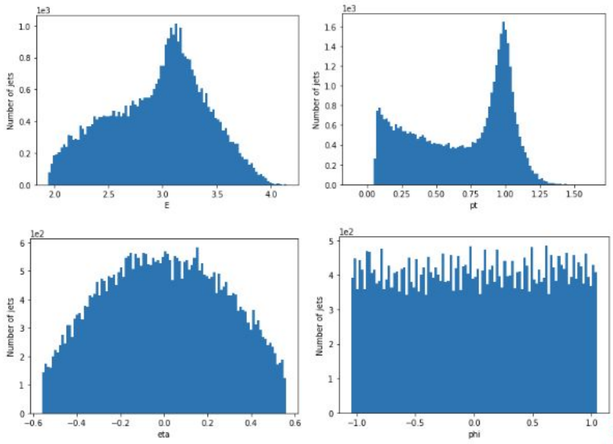
    - Unnormalized (divided by 1000)



    - Standard normalization



    - Custom normalization



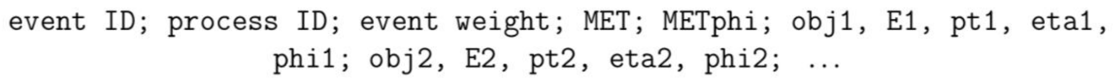
Custom normalization without additional factors: same thing

Next steps:

- Try normalization after 20 GeV cut
- Reduce the size of the dataset that is loaded to the minimum (check MSE loss with small dataset, then bigger dataset)
- Train and plot the 4 → 3 network
- After that
    - Use the same “dijet trained” network on a sample that is similar enough to jets [to check the physics process dependence of the network]
        - ttbar
        - Gluino (3 events??)
    - Add one category of particle in the samples we know
        - Jets → add photons as objects (‘b’, ‘j’, ‘g’)
        - ttbar → add leptons as objects (‘b’, ‘j’, ‘e’, ‘mu’)

20200713 - meeting with Caterina, Simona, Nathan

Current code will zero-pad the uneven arrays.
What we want is to make 4-vectors of jets only



```
event ID; process ID; event weight; MET; METphi; obj1, E1, pt1, eta1,
phi1; obj2, E2, pt2, eta2, phi2; ...
```

Obj1 = “jet” [label]

Stupid python code:

```
For i_event in events :
    Read in the event line
    Tokenize it (using .split(“;”))
    If “b” or “j” in token :
        Read the next 4 lines and save them as E, pT, eta, phi
```

OR

Pandas masking?

gluino_06	1	624432	-1.352227	*b,931728,230412,2.0741,2.37135*	*j,509750,209826,-1.53171,0.867619*	*j,174561,169253,-0.110392,-3.12433*	*b,589814,138939,-2.12443,0.495588*	*b,197622,110396,1.17779,0.874402*	*j,100801,98740.6,0.0511899,-2.37414*	*j,116107,94721.9,-0.650933,1.61946*	*b,88483.3,71695.6,0.609343,2.58646*	*j,63039.9,32330.5,1.28351,2.19988*

20200707 - meeting with Caterina, Baptiste and Joe

Honey's updates

See worklog

Review of OpenLab presentation (slides)

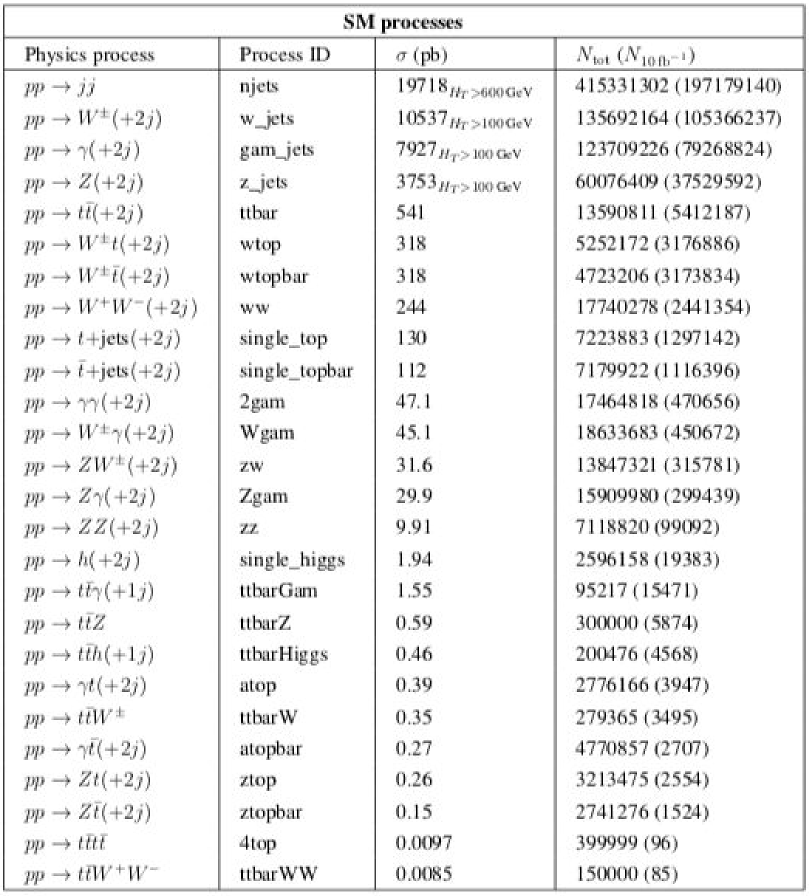
- Slides 14 and 33: maybe link to Eric Wulff's thesis?
- C: we shouldn't show plots from the non-public ATLAS dataset - can we make the presentation shorter? [will help offline]

Discussion with Joe

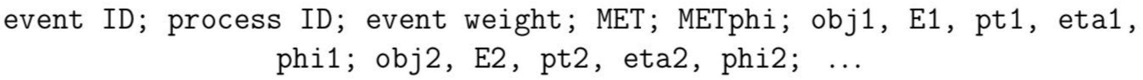
- We've shown we can compress jets with 4 or 27 input variables, can we move to event-level? → use phenoML dataset
- https://arxiv.org/abs/2007.01023 → it's possible to concatenate various compression networks
- Discussion
    - Example case: 10 kinds of particles, 4 variables / particle
    - Network case #1: network structure of 40 → 100 → 200 → 100 → 30 → 100 → 200 → 100 → 40 (symmetric, number have been made up)
    - Network case #2: each particle gets its own network 4 → 100 → 200 → 100 → 3 and then the losses are multiplied together (concatenation)

What next: read DarkMachines data! ~31 GB
https://zenodo.org/record/3685861#.XwQz5i-ZOJR

Start with multi-jets (pp → jj process), looking at this paper (page 198):
https://arxiv.org/pdf/2002.12220.pdf



SM processes			
Physics process	Process ID	σ (pb)	$N_{\rm tot}$ ($N_{10\,{\rm fb}^{-1}}$)
$pp \to jj$	njets	$19718_{H_T>600\,{\rm GeV}}$	415331302 (197179140)
$pp \to W^{\pm}(+2j)$	w_jets	$10537_{H_T>100\,{\rm GeV}}$	135692164 (105366237)
$pp \to \gamma(+2j)$	gam_jets	$7927_{H_T>100\,{\rm GeV}}$	123709226 (79268824)
$pp \to Z(+2j)$	z_jets	$3753_{H_T>100\,{\rm GeV}}$	60076409 (37529592)
$pp \to t\bar{t}(+2j)$	ttbar	541	13590811 (5412187)
$pp \to W^{\pm}t(+2j)$	wtop	318	5252172 (3176886)
$pp \to W^{\pm}\bar{t}(+2j)$	wtopbar	318	4723206 (3173834)
$pp \to W^{+}W^{-}(+2j)$	ww	244	17740278 (2441354)
$pp \to t$+jets$(+2j)$	single_top	130	7223883 (1297142)
$pp \to \bar{t}$+jets$(+2j)$	single_topbar	112	7179922 (1116396)
$pp \to \gamma\gamma(+2j)$	2gam	47.1	17464818 (470656)
$pp \to W^{\pm}\gamma(+2j)$	Wgam	45.1	18633683 (450672)
$pp \to ZW^{\pm}(+2j)$	zw	31.6	13847321 (315781)
$pp \to Z\gamma(+2j)$	Zgam	29.9	15909980 (299439)
$pp \to ZZ(+2j)$	zz	9.91	7118820 (99092)
$pp \to h(+2j)$	single_higgs	1.94	2596158 (19383)
$pp \to t\bar{t}\gamma(+1j)$	ttbarGam	1.55	95217 (15471)
$pp \to t\bar{t}Z$	ttbarZ	0.59	300000 (5874)
$pp \to t\bar{t}h(+1j)$	ttbarHiggs	0.46	200476 (4568)
$pp \to \gamma t(+2j)$	atop	0.39	2776166 (3947)
$pp \to t\bar{t}W^{\pm}$	ttbarW	0.35	279365 (3495)
$pp \to \gamma\bar{t}(+2j)$	atopbar	0.27	4770857 (2707)
$pp \to Zt(+2j)$	ztop	0.26	3213475 (2554)
$pp \to Z\bar{t}(+2j)$	ztopbar	0.15	2741276 (1524)
$pp \to t\bar{t}t\bar{t}$	4top	0.0097	399999 (96)
$pp \to t\bar{t}W^{+}W^{-}$	ttbarWW	0.0085	150000 (85)



```
event ID; process ID; event weight; MET; METphi; obj1, E1, pt1, eta1,
                    phi1; obj2, E2, pt2, eta2, phi2; ...
```

Planning

Step 1: reproduce the 4->3 network with DarkMachines jet data, just considering the jet's 4-vector.

- Read the data in
    - Scripts at the moment have events in columnar format, so this makes it harder to read (see https://github.com/Autoencoders-compression-anomaly/collider-unsupervised-learning/blob/master/phenom_data_read_in.py)
    - Filters: how to take into account? For now, let's not worry about this when we use jets only
    - Question on whether to identify the particles?
        - Probably not possible to do in network version #1, but we could apply a "weight" variable in the MSE loss to compress different particles differently

        - Question: can we read everything in and mask part of the dataset for the variables we are not using
            - Joe: probably
        - Question: can we use the labels either as input to the encoder or decoder?
            - That's a feature of VAEs (conditional VAE) - we don't sample the latent space in an AE
            - But let's think about this idea more...
        - CD: AwkwardArrays?

20200630 - meeting with Caterina and Baptiste and Rebeca

Honey's updates

See worklog

Review of presentation on work done (slides)

- To be presented at the next anomaly detection forum?
- Minor comments for the presentation, results look very good!
    - Went slide-by-slide, adding some clarifications for future readers.
    - Tasks clearly marked through the presentation
    - Message on each slide discussed, some to be highlighted a bit more for external readers
    - Suggested to add a conclusion slide and at least one lines on each plot-slide
- R: question on labeling of correlation plots, slide 14
    - H: it was a problem solved in further plots; will correct the plot
- B: do we understand the difference between training and validation loss on slide 16?
    - H: differences of order 10^-5 (left plot is "unzoomed") not worrying, attributable to test/train/validation split
    - (Would be interesting to see if this is a recurring feature)

Planning

- Caterina to put Honey in touch with Joe Davis from DarkMachines slack, who's working on a script to read in the phenoML data
- Use phenoML data as input for event-level AE
- Finalise instructions on how to run on HTCondor and make sure Sam can too

20200623 - meeting with Caterina and Baptiste and Rebeca

Honey's updates

See worklog

Coming up:

- Make plots for 10 and 100 epoch training and “complete” presentation on work done so far (also explaining the big picture) - if ready for Monday we can present it at the ATLAS anomaly detection forum (but no pressure)
- Add to HTCondor quickstart: “how to run and retrain the network”

Minsky tests

See Trello (mostly things to do by Caterina)

Planning

CERN webfest is a possibility this weekend: https://indico.cern.ch/event/923748/
29th June - 3 July: evaluation won’t take all this time
See timeline and deliverables google doc - Honey should choose option 1/2/3 for the main part of the work coming up now and let the supervisors know.

20200616 - meeting with Caterina and Baptiste and Rebeca

Introductions

Rebeca, Uppsala/ATLAS, with a PhD student

Mailing lists

See document

Progress last week

Looking into HTCondor for GPUs

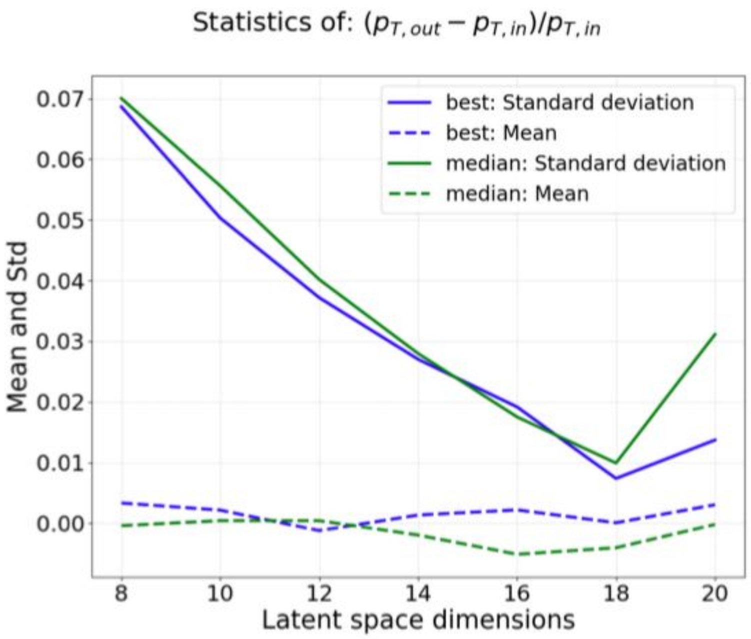
- Docker
    - Very slow in installation, went to hold and then error
    - Let’s not look into Docker errors too much yet
- Virtualenv
    - Not yet know how to take inputs/outputs, how do we deal with the data files?
        - Afs is probably not working
        - CERNBox?
    - Could start interactive nodes and can access afs/ (not work)
        - Was able to do some more debugging

Discussion with Lukas.

- 6 pm this Thursday [probably not good? 6 pm tomorrow]
- Time: TBC
- This week conference, so could do anytime → Caterina writes to Lukas

Tried on 27D data
Older checkpoint as 27->20 latent space.
150 epochs, 7-8 hours, but for 350 epochs, 1 day

How to decide latent space dimension?
http://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=9004751&fileOId=9004752



Minsky → [unnamed IBM computer cluster]

Meeting yesterday, summary on notes [link]

Honey: Write an email to Eric Aquaronne to be added to Slack (get address in DM)
CD: Write an email to Guillermo with our lxplus usernames to be added to the WMLA interface / machine

Datasets

https://docs.google.com/document/d/1D4Ld1ZB82ajeynwuqPiQlPuJdsp2mquOGH5av5_Az2Q/edit

Confirmed from Erik Wallin that the 27d datasets are what Eric Wulff used.

Timeline

20200609 - meeting with Caterina and Baptiste

Change in training function as practice with existing network

Last week: tried changing loss function from MSE to L1 on a 4D dataset, no re-training of the network.

Slides where this is discussed:
https://docs.google.com/presentation/d/12_yRCl63H1VEIDejEUteBqwXalWs0b9x3YcoKGFkkZA/edit?usp=sharing

MSE is better, but checking a loss function that is the same as the training will give results that are favourable to the training one.
From the relative errors and residuals it is not clear which one is better, in general L1 is lower. But we still may need to change in L1 + MSE or something different.

Looking at the corner plots we may learn something about the correlation between the variables. L1 seems to have a better behaviour? But hard to infer from correlations of residuals.

What is the “ultimate figure of merit”?

- MSE is the most straight-forward loss we are using (it’s effectively a combination of all the residual plots)
    - We could define a more optimized metric using the combined residuals of all the variables that we are accounting for, where some of the variables that we care about more have a bigger weight
    - Maurizio Pierini has a paper out about this: we should read and summarize https://hal.inria.fr/hal-02396279/document

Normalization of variables

Currently using custom normalization as in:
https://docs.google.com/presentation/d/1QMAuUOPh8tp32xdqTDh_LHtZjvoBFSvbh57SJ_rHzpo/edit?usp=sharing

Another problem brought up by Baptiste: intrinsic correlations between the four variables concerning a jet may be removed when normalizing one at a time.

Open questions:

- Normalization
    - The variables encode some physical quantity, if we normalize them one at a time, we lose the physics correlations between them
        - And then the network goes back to try and find those correlations…
    - Check papers: no huge problems, but an open question

Overall not clear what is better. Caterina can email about the Darkmachines community for resources about this and cc Honey, and we will share the results.

Other points

From Erik Wallin’s thesis: 4D datasets don’t train well. Could it be because m/E mismatch?
What we will do: proceed with 27D data as discussed below.

Where to do training

Ask Lukas about GPU on lxplus/condor and grid?

Next steps (to be moved to Trello)

1) [Honey] Learn how to use computing resources for training
    a) HTCondor - start with https://batchdocs.web.cern.ch/local/quick.html
    b) IBM Minsky [Caterina adds Honey and Baptiste to another Slack - process started, IBM needs to get back to me]
2) [Honey] Moving to 27 variable dataset, this is the goal for the week]
    a) Trying to run the network and make 1D/residual “corner” plots
    b) Trying to retrain and understand how long it takes
    c) How many epochs has Eric Wulff trained for?
        i) Some info on fastAI 1cycle (not yet implemented) on P20 of http://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=9004751&fileOId=9004752
        [Update, HG] For 4D data: 10@10^{-7}; 10@10^{-4} and 2000@10^{-6}.
        For 27D. he performed grid search. The plot shows that around 350k batches were processed, so this can be taken as a starting point.
        ii) Honey will come up with the “optimal” number of epochs
            (1) Consider range of loss function “good enough” and then start the testing
3) [Caterina, Baptiste] Informing ourselves on
    a) Normalization of variables
    b) Loss function choice
4) [Everyone] Thinking about metrics to use for anomaly detection
    a) Baptiste will send a paper about this → done

20200602 - meeting with Caterina

Discussed timeline [link]

Weekly meetings on Tuesdays at 9 am UK time / 10 am CERN time

20200526 - meeting with Caterina

How to get involved in the community:

HEP Software Foundation

https://hepsoftwarefoundation.org
https://hepsoftwarefoundation.org/future-events.html
I will in any case alert you of the meetings that are of interest for your project.

13-17 July https://indico.cern.ch/e/pyhep2020

ATLAS

http://atlas.cern

Anomaly detection forum (community):
Meetings every other Monday at 13:00 CERN time → next week (think about introducing yourself & project at the June 8th using the presentation already prepared for evaluation task)

Attend ad-hoc introductory meetings (shared google calendar with other students and supervisors) + analysis meetings where someone doing similar things is presenting

ML Forum - what it is about
https://indico.cern.ch/event/545453/contributions/2214995/attachments/1301084/1942374/S2I2_20160629.pptx.pdf

https://atlas.cern/tags/machine-learning → to learn more about the “challenges”

16-17 July https://indico.desy.de/indico/event/25341/ → about anomaly detection

Autoencoders for compression

Weekly meetings with everyone involved who wants to join

Computing resources

After registration, will get email
Laptop w/1 GPU, basic prototypes (Ubuntu)
Lab resources but no access to it right now

- Lxplus cluster (CPU farm) → prototyping
- Minsky (?????? from IBM at CERN) → Spark
    - We’ll have a session with IBM people

Communication
Probably will use Slack from Ohio State, TBC - slack hn.gpt1@gmail.com

Lund

Thesis defense of Erik Wallin: June 1st, 10 am Lund time
https://lu-se.zoom.us/j/64891191813

Scrum?

Will have a chat with some people who did this with students during our first meeting
Daily meeting to update everyone about what we're doing

- https://indico.cern.ch/event/304944/contributions/1672228/attachments/578481/796612/agile_research.pdf

Honey's previous experience: had a google doc and a fixed meeting time every week and everyone updated the google doc before the meeting with what was done and what will be done in the coming week.

Physics questions

ATLAS and HEP software

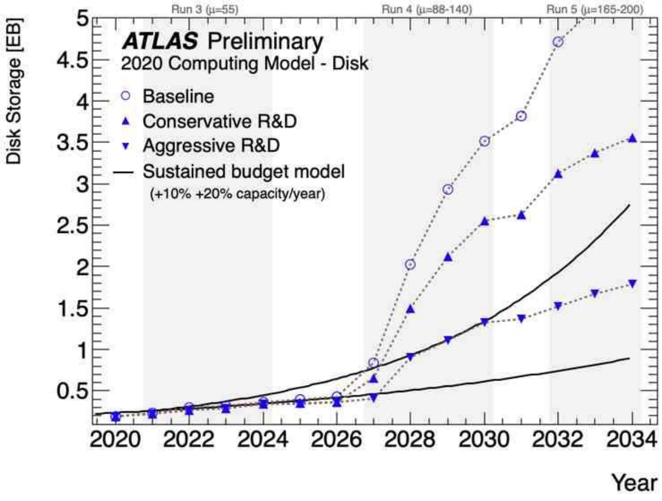
https://arxiv.org/pdf/1712.06982.pdf

https://twiki.cern.ch/twiki/pub/AtlasPublic/ComputingandSoftwarePublicResults/cpuHLLHC_comparison_2020_InputData_3April_CMSC.jpg

Mu = simultaneous proton-proton collisions within a bunch, we want this number to be high to be able to discover rare processes (most of the collisions will give us known processes).

Event = result of a proton bunch-proton bunch collision (LHC collides bunches of protons). An event contains all the 200 interactions of the 200 protons that collided.

Disk space needed increases with mu because there are more collisions simultaneously.



Challenges:

- Precision after compression
  - Very important but not essential to be lossless (our physics cases are robust against small fluctuations, 2-3% is tolerable, 5% still tolerable but maybe questionable…)
- Storage space (compression factor)
  - As much as possible without losing too much precision
- Computation speed (CPU needs)

    - Important but not critical because there are bigger “consumers”
        - Important to keep track of it:
            - Timing for training & hyperparameter scans, we do it one-off (more or less)
            - More important: inference time / unpacking time

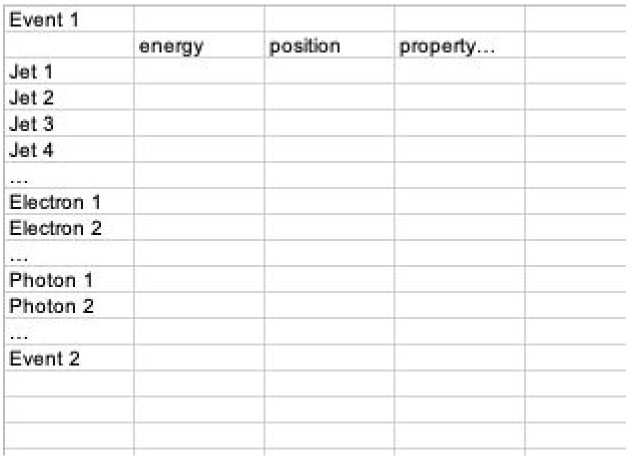
Which data?

- Simulation for new physics signals
- Data from 2018 LHC
- [open data] for things outside ATLAS

Jets → spray of particles in the detector, coming from quarks and gluons.
https://www.youtube.com/watch?v=df4LoJph76A

Start with jets, move on to event-level compression
[another student studying other compressions for jets]

Event = excel spreadsheet (ntuple)



Event 1				
	energy	position	property...	
Jet 1				
Jet 2				
Jet 3				
Jet 4				
...				
Electron 1				
Electron 2				
...				
Photon 1				
Photon 2				
...				
Event 2				

“IParticle” ← Jets OR electrons OR photon
Characteristics that are common to all (4-momentum):

- Position in the detector (px, py, pz)
- Energy (E)

In the evaluation task, 4-momentum was used. Next: add more variables (that may be vectors of vectors).

Timeline and deliverables

Let’s discuss this on Tuesday June 2nd, 7:30 - **8:15** UK time → 12:00 Indian time