Check all our meetings on indico: https://indico.cern.ch/category/13860/

13/01/2023

	WIP	Issues	Suggestions
Renato			
Kristina			
Kalliopi			
Piyush	Designing generative transformer model	1. Bringing stochasticity into the model following NLP-based approach 2. Conditional model	VQ-VAE to model dictionary or Diffusion models

[Here more details]

Piyush
Link for the presentation.
Slide 5, 3rd point: Another approach could be to add first token as random noise, and force the model to generate showers considering that random noise. How to exactly do that is not yet figured out.

Anna

1. CHEP abstract accepted - proposing Piyush as speaker
2. Summer projects: we could propose Openlab Student (possibly covered by IBM, to be seen - prediscussed with Sofia), and (one/two) GsoC students. Concrete proposals for independent GsoC projects should be shaped soon.
3. Follow-up with IBM on resources: Sofia will reach out again to John

20/01/2023

	WIP	Issues	Suggestions
Renato	Latent Space with Noise Generation Multiple tests on	No convergence with the current methods tested	

	different sizes of the model layers		
Kristina	Using just one GCN for all patches - trying pre-processing of the values to see if it helps.	Always goes to producing almost similar images (represents well the shape of an average shower) -> no variability	1. Focus on attention layers. 2. Look into the aggregation function - what is there, learnt aggregation? 3. Different graph construction.
Kalliopi	secondary learning task (regressor): reducing latent vector dimension to 1 and adding new term in loss function, calculating MSE between primary energy and regressor output	1. implementation errors (wrong total energy) 2. including events from primary energies of 64, 256	on pre-processing: dividing each cell energy by: 1. max energy within a shower 2. dividing by 99th percentile
Piyush	1. Diffusion reading material 2. Push the code	-	-

Piyush

1. Read some papers on diffusion
2. Did some code improvements
    a. Fixed previous E_cell plot
    b. Added more plots to weights and baises
3. Patch check for next layer prediction. Patch splits were 1x1x45. It worked well.
4. Tried 3d learned positional embeddings. No significant improvement in our case.
5. Tried downstream tasks by directly feeding showers to MLP w/ new preprocessing.
    a. Energy - overfitting, 99% train vs 65% validation
    b. Angle - 99%

Renato

1. Change VAE layers size for better generation results
    a. Use optuna for hyperparameter tunning
    b. add some skip layers

20/01/2023 - meeting with Mudhakar

Objective - A working generative model by CHEP (mid-April), accurate enough. So: around **2.5 months for dev**

Questions:

1. How AR and Diffusion compare wrt performance in general? Depends, on the data and maybe other things. For real images diffusion is better, but there is no clear winner.
2. Diffusion, how slower is it? Number of diffusion steps? Training time? -> Yes, slower, but there are works which make it faster. Training is also 4-5x slower.
3. Any relation between number of diffusion steps and complexity of the model? Also, the task for model is simpler compared to VAEs/GANs?
4. Conditions (Energy and Angle) - positional embeddings vs prompt? -> Prompt would be better vs adding constants to all patches.
5. Layer 0 - As a context essential? We can have it, but it'd better to not have it. -> Start with layer0 as context, then for the update, we can remove it. Concern: If layer 0 does not contain much information, is it of any use as a prompt? -> Layer 0 should have enough information.
6. Why VQ-VAE + Diffusion? Expensive? -> Yes
7. VQ-VAE w/ transformer architecture? -> Yes, better to have it cause of xi x xj terms
8. Different diffusion methods? Pros/Cons?
9. Noise schedule, cosine?
10. PyTorch or Tensorflow? -> PyTorch support is there, not Tensorflow.

Conclusions

1. We first train the VQ-VAE w/ transformer architecture.
2. We freeze the VQ-VAE and use it to go from shower space to latent space and vice-versa.
3. Transformer will be our autoregressive model that will only see this latent space and hence output the probabilities over the codebook vectors in this space.
4. We start with converting our code to PyTorch in the same repository but a different branch.
5. Also, perform an analysis of how much information is there in the 0th layer.

27/01/2023

	WIP	Issues	Suggestions
Renato	Continue debugging the Noise Generation Running optuna on the Vae with and without the Noise Generation / Sinkhorn Loss		Look into Diffusion Model implementation
Kristina			

Kalliopi			
Piyush	Implementation of autoregressive model	-	-

Dalila
Here are a few plots to show the applicability of layer 0 as context for the next layer generative task. We can see on slide 2 that the total energy in Layer 0 is different for showers

Piyush
Almost everything is converted now to PyTorch. The code (for now) can be found here.
Further discussed about the student's projects:

- Openlab student:
  - Generic description.
  - Could assign to explore preprocessing and loss or our ongoing tasks.
- GSoC:
  - Should be a standalone project.
  - Exploring custom attention + hierarchical architecture for our case.

03/02/2023

	WIP	Issues	Suggestions
Renato			
Kristina	Use of single GCN for all patches - normalizing A and features (layernorm), changing activ. function	Improvement but still not accurate enough.	Try selu (or other activ. f.), batchnorm, condition the GCN with primary energy
Kalliopi	run first experiments with new loss function (integrating primary energy "regressor")	check results / suggestions on optimization	
Piyush	Implementation of VQ-VAE (MLP) is	Lack of diversity in reconstructed	Verify posterior collapse by:

	done. VQ-VAE related experiments in progress.	showers, probably a case of posterior collapse.	• providing trivial latent • checking variance of activations • checking mutual information b/w latents and outputs Look into mitigating posterior collapse

10/02/2023

	WIP	Issues	Suggestions
Renato	Looking into diffusion models Generation of the noise process of diffusion models		
Kristina	Layer norm working better than batch norm. elu, selu, gelu very similar		Update GCN weights more often than MLM weights
Kalliopi	experiments with new loss function (integrating primary energy “regressor”), (code-wise: looking on how I can separate the two losses to plot them)		use latest preprocessing (power), display more physics plots
Piyush	More VQ-VAE experiments. Initial results lead to very good results.	Perplexity *might* not be consistent across different runs	-

Publish paper/workshop Neurips deadline 11.05.2023
ICML? ICCV?

17/02/2023

	WIP	Issues	Suggestions
Renato	Implementing the DDPM model using mlp layers	Using attention layers on the UNET architecture	remove the attention layers
Kristina	Running optuna on the current setup (optimizing no of layers, no of features, activ. function) Changing the code to update GCN more often (WIP).		
Kalliopi	work on the updated Pytorch code to integrate my task	issues with code execution (environment/node)	
Piyush	Implementation of autoregressive prior	Multiple design choices. Layer-wise progression does not seem scalable. Scalable approach might not follow layer-wise progression.	Explore all but scalable approach might be the way to go, even if it goes against our original motivation.

Piyush:
Design choices:

1. *(ARV1) Single token prediction* - VQVAE outputs multiple token for each layer. AR prior predicts each token one-by-one. This leads to 45 x 32 (number of tokens used to represent a layer) number of forward passes in the AR prior. Although simple design-wise, it's not scalable due to huge number of forward passes.
2. *(ARV2) Multi token prediction* - Small design tweak. Uses multiple heads for the classifier instead of single head. This enables prediction of all 32 tokens at once. Memory complexity inside the transformer remains the same as 1, since sequence length remains same. This might be hard to scale, as number of weights in the classifier increases 32x (This factor could increase as well). Not simple design-wise, just a patch on top.
3. *(ARV3) Generic VQVAE* - VQVAE sees the whole shower, not only a single layer. Hence, we only need 32 forward passes of AR prior. Obviously, using more than 32 tokens would a better idea. But the factor of 45 layers is not there. Plus, we can increase the token dimensions to tradeoff the number of tokens required. *Not*

completely sure that there is no way to integrate shower progression. Can we enforce what each token represents?

24/02/2023

	WIP	Issues	Suggestions
Renato			
Kristina			
Kalliopi			
Piyush	Experiments on ARV2 and ARV3	Epoch time ARV1 >> ARV2 > ARV3 High accuracy but shower observables not accurate. Due to sampling?	Try out GAN discriminator as a loss.

03/03/2023

	WIP	Issues	Suggestions
Renato			
Kristina			
Kalliopi	experiments with loss function on the updated torch code, using pre-processing power transformations	improved performance?	
Piyush	• Initial results for ARV3 are decent. • Further experiments needed. • Remove condition on	Transformer-based VQVAE suffers from no diversity	-

	first token.		

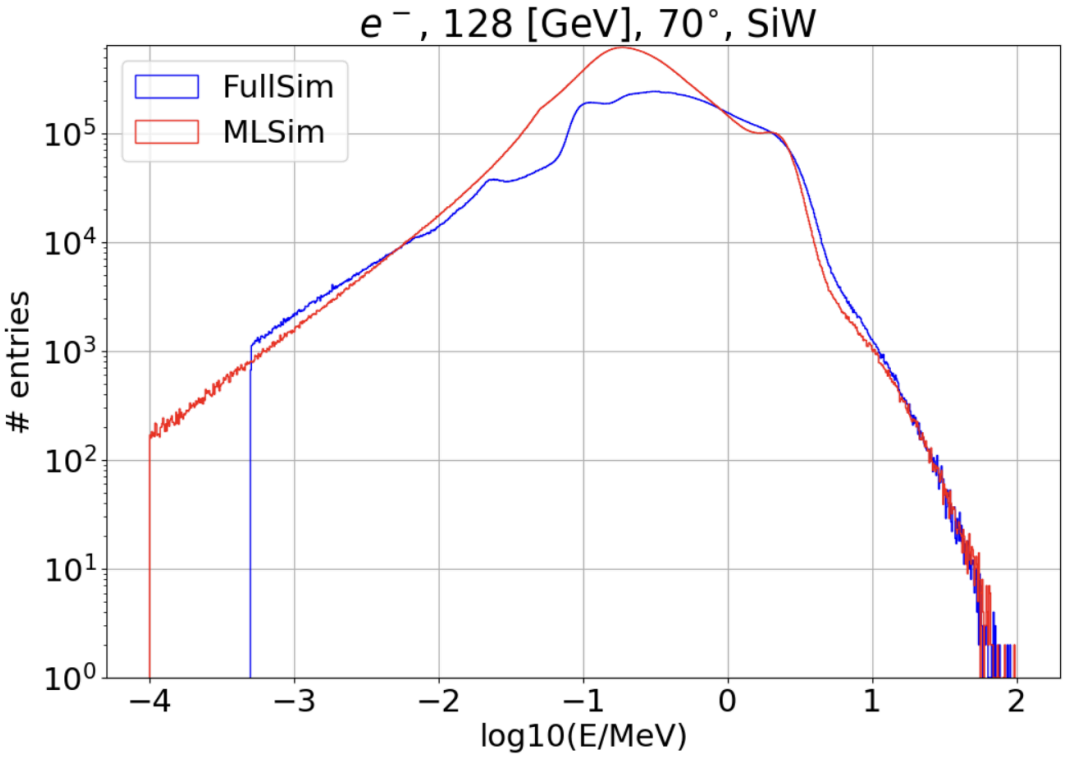
14/07/2023

	WIP (work in progress)	Issues	Suggestions
Renato			
Kristina			
Kalliopi			
Piyush	Why fixing attention gives worse results?	-	-
Chenguang			
Zeeshan			

Piyush

1. **Attention implementation:**
    a. It had a typo where softmax was being taken over wrong dimension (queries instead of keys). However, fixing it is resulting in worse performance. We don't get the diversity in total energy, first/second moments.
    b. Tried different values for the commit loss weight (from prior encounters), didn't work. Tried different values for vq loss weight, didn't work. PyTorch implementation of the TransformerEncoderLayer is also resulting in similar performance.
    c. For TransformerMLM, custom as well as PyTorch implementation, both seems to work well.
    d. Will now try to play with the architecture.
2. **Better cell energy distribution**:
    a. An attempt to get better cell energy distribution via various loss/activation function and pre/post-processing.
    b. Tried different combinations of these:
        i. Divide by E, Sigmoid, BCE (default)
        ii. + Logit transformation, Linear, L1
        iii. Divide by E, Hardsigmoid, BCE
    c. What worked was:
        i. Divide by E, replace zeros by -c (e.g., 1e-4)
        ii. Merge Linear and Sigmoid keeping the slope consistent

iii. Use BCE within appropriate range, L1 outside
iv. This improved cell energy (following figure), but rest of the shower observables were worse.



21/07/2023

Piyush

1. **Attention softmax dim=3** **(VQVAE with transformer, no AR)**
    a. After several experiments with the combination of custom attention & transformer blocks, PyTorch's attention & transformer blocks, the problem seems to be with the VQVAE framework.
    b. After trying various values for old/new attention, the conclusion is that for old attention (softmax dim=2), the values for the commitment loss & vq loss weights were actually the good ones but unfortunately these don't work for some different configuration (e.g., data-based init of codebook, codebook init w/ or w/o norm, attention softmax dim=3, etc.)
    c. What's happening is that the perplexity (different tokens used) remains low, thus no diversity.
    d. For the case in which we are getting diversity, vq/commit loss does not consistently decrease. This points towards that the losses should be comparable, so that the losses can be traded off easily.
    e. Two experiments:
        i. Set commitment loss and vq loss weights based on mean value of losses after feeedforwarding whole dataset (without updates). Hence, COEFF_COMMITMENT = commit_loss / bce_loss

COEFF_VQ = vq_loss / bce_loss
(old_attn_codebook_init w/o norm, old_attn_codebook_init w/ norm)

ii. Dynamically do the above after every few epochs (e.g., 10 epochs). So the COEFF_COMMITMENT & COEFF_VQ is reset after every 10 epochs. **(Need to reinit optimizer as well?)**
(old_attn_no_norm, old_attn_norm, new_attn_no_norm, new_attn_norm, new_attn_no_data_based_init)

f. The second experiment worked, for old attention w/ & w/o data_init of codebook, as well as for the new attention. **But can be unstable, difficult to decide when the model actually converges.** Perplexity can rise and fall at any time.

g. 3rd experiment: Scale losses for each optimizer step (Bad for Adam?) (Running)

h. As a byproduct in solving this issue, also implemented Gumbel-Softmax

quantizer, and VQEmbedding with EMA updates.

2. **CaloChallenge Dataset 3**
   a. Experiments running (vqvae)
   b. Most of the things as before, attention softmax dim=2, sigmoid, BCE
   c. ~0.3% of voxels were > 1 after preprocessing (i.e., division by incident energy)
   d. Changed the preprocessing to divide by 4300 (rounded off to maximum value in the dataset)
3. **Next steps:**
   a. Dataset 3 experiments:
      i. train VQVAE,
      ii. train AR with old attention, &
      iii. train AR with new attention
   b. ONNX conversion of these models
   c. CaloChallenge paper

Chenguang Guan

1. **Attention Softmax dim=3 & Positional Embedding**
   I set six experiments with attention-based ARV3-VQVAE on the SiW-90 dataset. The results are saved in the Positional_Embedding project.
   1. Attention Softmax dim: 2 or 3;
   2. Positional Embedding: 1D or 3D or None;

   a. The softmax dim =3 results are the same as Piyush's results. In the first/second moment of Long/Lat profile and Energy distribution of each layer, the distributions are peaked around the peak value of FullSim. This means that there are no variances.

   b. In the softmax dim=2 experiments, the none pos embedding case has slightly better performances than 3D pos embedding (mainly in energy distribution of each layer).

2. **MLP-based mixers and Unparameterized Fourier Transform**
   I replace multi-head attention with two kinds of MLP-based mixers as well as an unparameterized Fourier transform mixer. All the experiments are saved in the New_Mixer project.

   MLP-mixer:

The MLP-mixer comes from https://arxiv.org/abs/2105.01601. The architecture is: Data (Batch * Patch * Embedding) → Transpose (dim-1 and dim-2) → fist MLP (mixing information across patch) → Transpose → second MLP (mixing information across embedding dimensions) → Next Layer…

We can have comparable performances with the attention mechanism (same layers, ARV3-VQVAE). Based on this result, maybe we can guess that the low variance (of softmax dim-3) is mainly due to VQVAE part? (Anna's comment: and not the transformer)

ResMLP:

The ResMLP comes from https://arxiv.org/abs/2105.03404, which is a variant of MLP-mixer. The architecture is: Data (Batch * Patch *Embedding) → [Cross-patch sublayer] → [Cross-channel sublayer] → Next Layer…

Cross-Patch sublayer: [Affine Transformation → Transpose → Linear Transformation → Transpose → Affine Transformation]
Cross-Channel sublayer: [Affine → Linear → Gelu → Linear → Affine]

The ResMLP results are bad. We also found the low variance in the first/second moment of Long/Lat profile and Energy distribution of each layer, which is very similar to the softmax dim =3 case.

Fourier Transform mixer:

This idea comes from https://arxiv.org/abs/2105.03824. (Piyush's comments: the performances seem not so good compared with other architecture)

3. **Next Step:**

    a. (High priority) figure out the reason for low variance in the softmax dim=3 case.

    Piyush's suggestion: Experiments about loss coefficient in ResMLP, MLP-mixer, and Attention (softmax = 3, if time permitted).

    b. play with other mixer mechanism

    VAE instead of VQVAE

Zeeshan Memon

1) Debugging softmax preprocessing scaling issue:
    a) Tried with gradient scaling
    b) Different coefficients combination
2) Overviewed Results with Max Energy per layer Scaling https://wandb.ai/foundation-models/Max%20Energy%20Scaling/runs/2g69dt4a?workspace=user-zeeshan-memon

    a) Further investigation is required - whether results are due to variation of Max energy or model predictions
    b) [anna's comment: division by max value seems to be working much better than division by total energy per layer]
3) To Do:
    a) Try MMD loss and compare this with current performance
    b) [Anna+Piyush suggestion: try to go back to div by total E per layer because it is a much more meaningful input, but use it together with some scaling factor that allows to make the cell E distrib better]

Kristina Jaruskova

- more reading
    - Graph Variational Autoencoder for Detector Reconstruction and Fast Simulation in High-Energy Physics (https://arxiv.org/pdf/2104.01725.pdf)
        - Ali Hariri, Darya Daychkova, Sergei Gelyzer
        - simulation of top quark pair events
        - graph - hit ~ node, features ~ coordinates and hit energy
        - GraphSAGE to embed node features into latent space, based on message-passing (aggregates information from neighborhood) but

randomly samples nodes from the neighborhood (https://snap.stanford.edu/graphsage/)
    - min-cut pooling for dowsampling/upsampling of the number of nodes (https://arxiv.org/pdf/1907.00481.pdf)
    - overall architecture based on VGAE (https://arxiv.org/pdf/1611.07308.pdf)
    - missing some details, no public code

- issues with CUDA on the node - ticked submitted, waiting for response

28/07/23

Piyush

Calochallenge

1. Trained VQVAE - old attention
2. Trained AR - old attention
3. Trained AR - new attention
4. Not such a significant difference between b & c
5. (In progress) Training of VQVAE new attention dynamic weights
6. ONNX conversion done using 1 & 2 (takes an hour!)
7. Verified ONNX model via shower observables
8. (TODO) Push the code to upstream, new things:
    a. PyTorch to ONNX conversion
    b. Generation of showers via ONNX model
9. (Waiting on input from Claudius) I/O for ONNX model. Need to update and re-convert.

Chenguang Guan

Readings:

1. I reviewed four MLP-based mixing architectures more carefully: MLP-mixer, ResMLP, gMLP (https://arxiv.org/pdf/2107.10224.pdf) , cycle MLP (https://arxiv.org/pdf/2105.08050.pdf). I have implemented two of them (MLP-mixer and ResMLP) before and showed the results in the last meeting.

   A good summary of these four models is as follows:

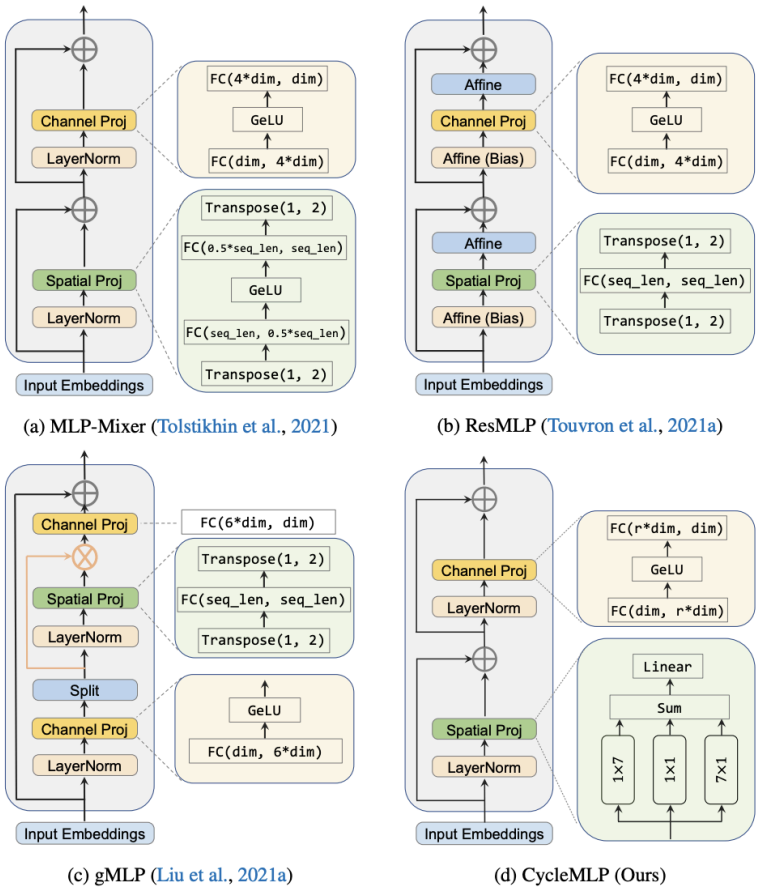
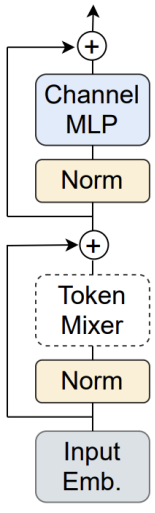


Figure 5: Comparison of MLP blocks in details.

.

The figure comes from the CycleMLP article.

2. There is another class of architecture: Linear Attention (Fourier Transform Transformer can be seen as a kind of linearization). I noticed that academia is tending to replace softmax attention with linear attention (but I am not completely sure).
3. All these models can be described as:

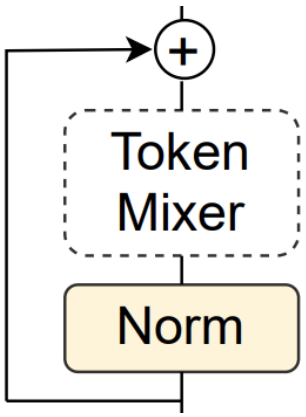


MetaFormer
(General Arch.)

(The figure comes from https://arxiv.org/pdf/2111.11418.pdf).

Debugging:

1. I standardized my codes (implementation of MLP-mixer, ResMLP, Fourier Transform mixer) and fixed all the following bugs/typos. All the codes are in the "autoregressive-dev" branch of my personal repo: https://gitlab.cern.ch/cguan/ml4fastsim/-/tree/autoregressive-dev?ref_type=heads .
2. Bug-1: I only initialized layernorm once in each sublayer (`self.ln = nn.LayerNorm(projection_dim)`) and used the same layernorm in the token mixer block and feedforward block. This may result in the layernorm of token mixer sharing parameters with feedforward block?
3. Bug-2: The starting point of skip connection (residual learning) should be set before the layer-norm. I previously set the starting point of the skip connection after the layer-norm.



← The correct skip connection.

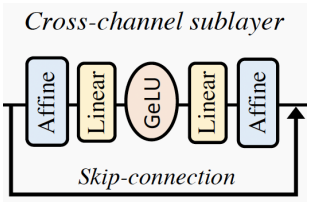
4. Bug-3: I used duplicated feedforward blocks (Channel Mixer) in each encoder layer.
5. Bug-4: I found that the pseudo-code in the ResMLP article missed the post-affine transform in the Fourier blocks and Feedforward blocks.
6. Bug-5: I applied FFT to all three spatial dims, but I forgot to use FFT on projection dim.

Fourier Transform based Mixer (Implementation & Experiments):

1. I mentioned this work in the last meeting: https://arxiv.org/abs/2105.03824 . I implemented this architecture (https://wandb.ai/foundation-models/New_Mixer/runs/inr4e7s1?workspace=user-chenguang-guan), but because there are some bugs in my code (mentioned above), the performances were even worse than the softmax-dim=3 case.
2. The original Fourier Transform based mixer is designed for NLP tasks, which is one dimensional. Therefore, I extended the 1+1-D FFT to 3+1-D FFT.

Some potential useful tricks:

1. Two layer norms (or Affine Transform) in one block: "pre-norm" + network + "post-norm".



2. Replacing layer-norm with Affine Transform.

3. In the Fourier Transform mixer article, there is only one channel mixer (feedforward block) in each encoder layer. We can add a patch mixer in each encoder layer.

Next Step:

1. Run experiments after debugging.
2. Apply these tricks.
3. Implement VAE Transformer.
4. I will write a note to cover the derivations and arguments of the Fourier Transform article.
5. If time permits, I will also write a note on MLP transformers.

04/08/2023

Kristina

Graph VAE

- in contact with Ali Hariri (author of a paper on graph VAE on used CMS data), need to agree on a date and time to talk
- I found the corresponding MSc thesis (link)
- found more pooling methods for graphs and a paper that compares some of them
  - minCUT pooling
  - DiffPool
  - SAGPool
  - SimPool
  - top-k pooling
  - comparison

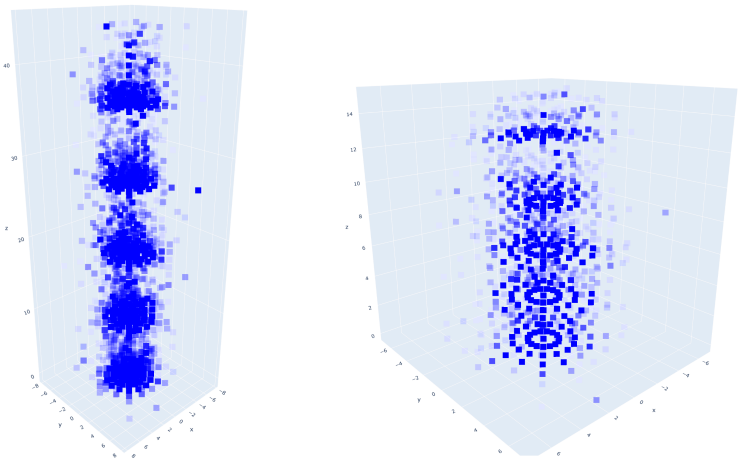
GCN + t-MLM image completion

- running again some of the experiments
- summarizing the results (link)

Renato

1. Datasets: https://calochallenge.github.io/homepage/

Warning for Calochallenge datasets: voxel are ordered differently, instead of r,phi,z -> z, phi, r
that explains weird bulks on vertical axis below



2. Two options for datasets:
    a. stay with dataset 2 of calo challenge and choose a range of E (check statistics!)
    b. go back to discrete energy dataset
3. https://indico.jlab.org/event/459/contributions/11736/attachments/9599/14176/CHEP23_CaloDiffusion.pdf

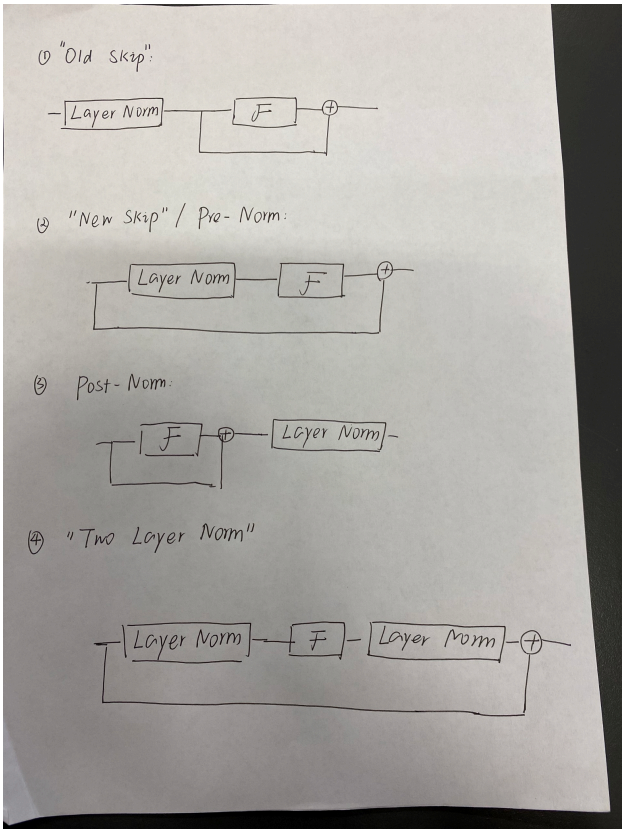
Piyush

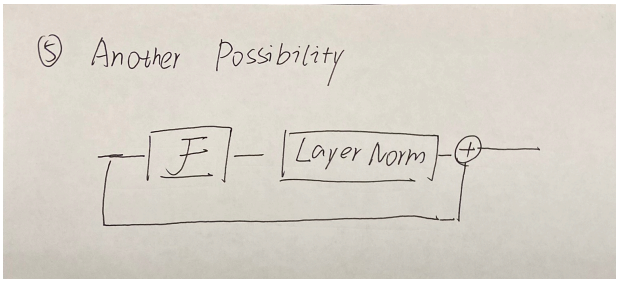
- Re-ran experiments wrt data_init (WIP), norm_embeddings after fixing softmax dim in attention. Turn off norm to increase expressive power of the vqvae.
- With data_init=False & norm=False, coeff_commit=0.01 worked once. Other values did not work, even dynamic weights. Unstable.
- ONNX inference to HDF5 done.
- For stabalising vqvae, this paper

Chenguang

1. LayerNorm issue
   "New_Mixer" project:
   https://wandb.ai/foundation-models/New_Mixer?workspace=user-chenguang-guan
   "Hyperparameter Tuning" project:
   https://wandb.ai/foundation-models/Hyperparameter_Tuning?workspace=user-chenguang-guan

Old Skip Connection; Pre-Norm; Post-Norm; Two LayerNorm arch; A fifth possibility





Under the following four settings, the transformer vqvae with new attention (softmax-dim=3) will have perplexity and diversity:

a. TwoLayerNorm arch
b. PreNorm + w/o SoftMax
c. PreNorm + New Patch Combination (3,5,9)
d. A fifth Norm arch

Conclusion: However, the training process is not so stable. There might be some randomness in the final perplexity and final performances

2. MLP-based arch + Fourier Block:

“New_Mixer” project:
https://wandb.ai/foundation-models/New_Mixer?workspace=user-chenguang-guan

Replacing Affine Transform with Layer Norm in ResMLP, it will work.

Fixing all bugs: all worked and had perplexity and diversity;

Fourier Block + new attention (softmax-dim=3): worked (having perplexity and diversity) and had better performances

3. Norm Embedding (following Piyush's experiments):
https://wandb.ai/foundation-models/Hyperparameter_Tuning/runs/o1c9qtj2?workspace=user-chenguang-guan

Train perplexity ~ 120
Val perplexity ~ 60

4. Next Step:

Write documents (Logbook and notes) and give theoretical analyses

Go through all the codes again

Move to next stage (Swin Transformer, and other pyramid/hierarchical archs)

Zeeshan

- Fixed scaling issue with total energy scaling, it was that softmax activation was required along two dimensions instead of one, as we are noramlizing on layerwise.
- b h w d -> h d (h w) -> taking softmax along last dimension -> rearrange it back
- all other graphs are corrected, but no significinat improvement is recorded for required observales
- Next To Dos:
    - Experiment the same experiment with dynamic coefficient balancing
    - MMD loss experimations

11/08/2023

Prediscussed: Next meetings start 9:00

Piyush

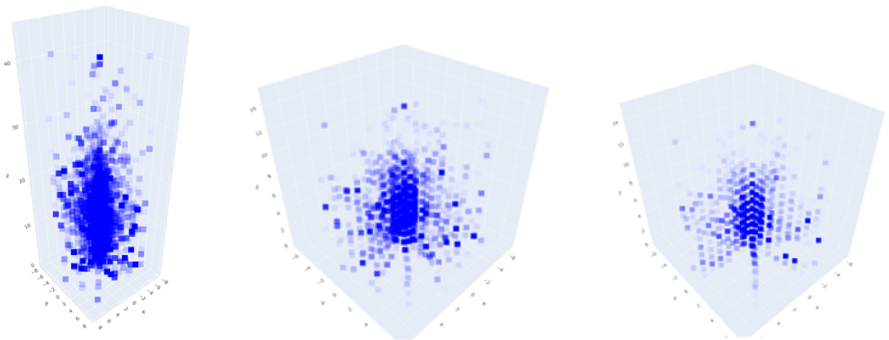
- Papers read:
    - Fast Decoding in Sequence Models Using Discrete Latent Variables
      - Multiple dictionaries corresponding to each token
    - Taming Transformers for High-Resolution Image Synthesis
      - No L2, but perceptual and adversarial loss with dynamic lambda (adv loss weight)
      - Transformer partial observability
    - Preventing Index Collapse in Discrete VAEs for Sentences
      - KMeans to initiate the codebook (multiple times)
      - AE -> Create codebook -> VQVAE -> Update codebook -> VQVAE -> …

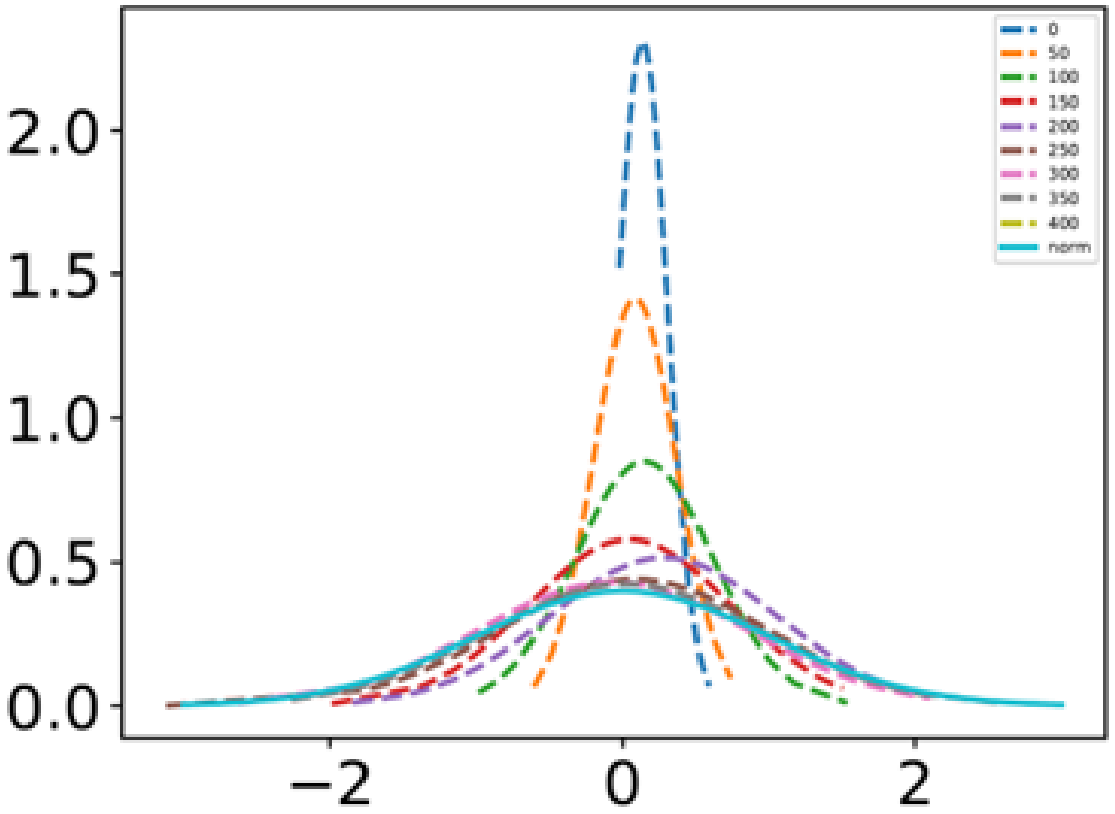
- Implementation done for the 1st and partially for the third.
- Went back to Autoencoder. Seems unstable. Trying experiments:
    - (DONE) Simpler arch
    - (DONE) Diverse data
    - Decouple latent
    - (Chenguang did it) Simpler patches
    - Diff optimizer
    - (Chenguang did it) Two layernorms
    - (DONE) Simpler arch Re
    - (DONE) Smaller arch L1
    - (DONE) Smaller arch L1 + leaky
    - (Crashed) Smaller arch SmoothL1 + leaky
    - (DONE) Smaller depermute
- VQVAE
    - Smaller arch diff coeff commit
    - Smaller norm off
    - Smaller data init off
    - Smaller norm and data init, both off

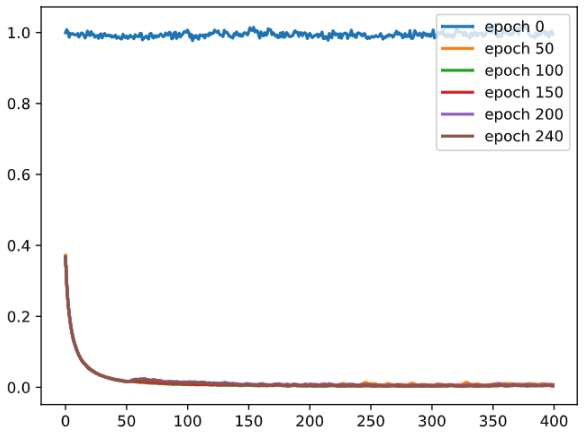
Zeeshan

- Readings:
    - InfoVAE - Combining MMD with VAEs https://arxiv.org/pdf/1706.02262.pdf
    - Implementation done, need to experiment
- Investigating baseline and total energy scaling comparison with AutoEncoders (AE)
    - Training is unstable, validated implementation
    - Following experiments
        - Default Preprocessing + AutoEncoder + BCE https://wandb.ai/foundation-models/default_study_name/runs/4hjbdfbc?workspace=user-zeeshan-memon
        - Total Energy Scaling + AutoEncoder + BCE

Renato

- Shape of downscale images (8x8x16(z)), 15 min per epoch
    - 

- Check single pixel distribution while adding noise (at different timesteps, for a single pixel somewhere in the middle and all the events)
  - 

- To plot: density plot of all pixels in the image for some event (1D histogram of cell values)
- Trained on all the energies and a subset the energies
  - 64 to 128 GeV (~10k events)
- Denoising: Check loss on each different diffusion timestep(x axis) (comparing noise added and noise predicted)
  - 

- Next Tests:
  - run optuna
  - try the cosine scheduler for the noise
  - Add the conditions for the layer and the radius
  - Train on just 50 timesteps to check reconstructed image

18/08/2023

We changed the time to 9:00 for our meetings

Mini-hackathon on Kubeflow

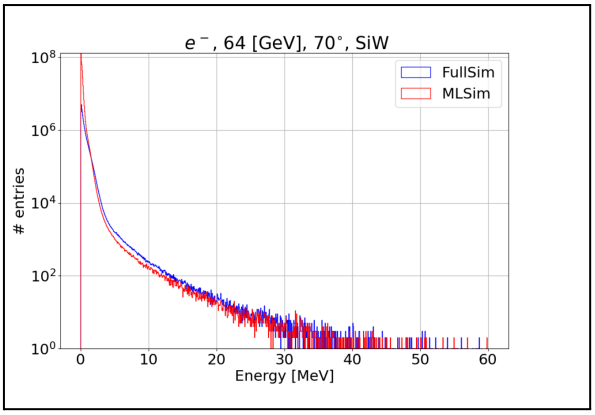
Took place on Tuesday 9-12,
https://docs.google.com/document/d/1hIwAODam-N5f8H6Sdfkd3RdJ-2rk_CbCU4qZFEpML-M/edit?usp=sharing
live notes contain all the material.
The main points:

- It is very unclear it Kubeflow can even be used, none of us got the resources (GPU)
- Otherwise the simple path to just use the resources from the terminal/notebook is clear
- To use AutoML (probably) or a clickable pipeline (certainly) a little more configuration is needed - a yaml config
- Sent our main concerns and questions to Ricardo and likely a meeting next week should get us more answers (Dalila and Renato said they are available and will note down the answers/minutes in the notes above)

Zeeshan

Tried following loss functions, to achieve high cell energies:

- Weighted Binary Cross Entropy - To penalize further
- Exponentially Weighted Binary Cross Entropy



Disturbed other profiles
https://wandb.ai/foundation-models/AutoEncoder%20Study/runs/kg0pex7h?workspace=user-zeeshan-memon
wrt Total Energy Scaling

https://wandb.ai/foundation-models/AutoEncoder%20Study/runs/g4kpn0xs?workspace=user-zeeshan-memon

- Weighted Binary Cross Entropy + L1
- MMD on latent representation - didn't work
  BIB AE [link] applied to output space , code
  Inputs - >Encodings
  Generated shower -> Encodings_hat
  Optimizing MMD between encodings and encodings_hat

wrt Default Preprocessing and Total Energy Scaling

To DOs:

- Compile all results for report
- Quick check for accuracy of Total Energy Predictions based on conditions(incident energy, angle, geometry)
- MMD on output space
- MMD with VAEs (in place of KL divergence) - inspired by InfoVAE

Link for Wednesday's Summer Student (Zeeshan Memon) Presentation.
https://indico.cern.ch/event/1309692/

Kristina

GCN + t-MLM image completion

- trying with enhancing node features - better results only for one energy, neglects that the images have different levels of energies (plots added to the summary slides link)
- adding loss term on energy classification (optuna to optimize params - loss weights)

Graph VAE

- started implementing graph encoder with GCN layers and mincut pooling
  - trying Spektral package for graph networks

Piyush

Implementation

- Add E_tot/E_inc graph

AE Experiments

- Loss
  - L1/SmoothL1 loss with leaky_relu -> underestimates the energy
  - (sigmoid) L1 + (sigmoid) BCE -> L1 made it worse
  - (leaky_relu) L1 + (sigmoid) BCE -> Nope
  - L1 proportional to voxel magnitude-> Nope
  - Same as above but weights sum up to 1-> Nope
- How latent space is connected to encoder & decoder?
  - Global (the default one)
  - Decoupled - shared projection -> Does not work, cannot get long prof
  - Decoupled - individual projection -> Works well, no overfitting even with same arch
  - Decoupled - shared projection w/ position info [TODO]
  - Number of tokens == number of patches (150)

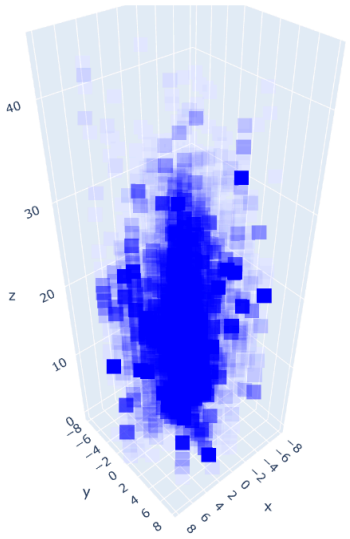
VQVAE Experiments

- Decoupled individual projection - smaller arch -> worked well! Constant increasing perplexity, medium entropy, stable! High cell energy dist. bad modelling. Need more expressive power?
- Tried a different commit coeff of the above 0.25 -> worked but entropy was higher, stable
- Global latent, 150 & 64 num_cb_vectors -> both worked, medium entropy for 64 but low perplexity & slow
- Global latent, no data init -> worked, slow, unstable
- Decoupled individual, smaller arch + GAN -> Better cell energy modelling, more expressive
- Same (w/ GAN) with L1 loss -> Need more training?

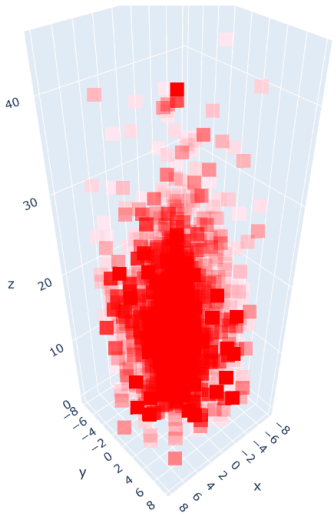
AR Experiments

- (Running) Effect of entropy - VQVAE 1st v/s 2nd?

Renato



Generated Shower



Original Shower

- Next steps:
    - Test with the whole energy range (condition on it)
    - Add physics verifications
    - Switch to dataset 3?
    - Read Oz's and Kevin's paper https://arxiv.org/pdf/2308.03876.pdf
    - Simplify network

Chenguang

Hierarchical structure:

- Pyramid Vision Transformer: implemented, experiments needed; https://arxiv.org/abs/2102.12122
- Swin Transformer: not suitable for generative models, good ideas needed; https://arxiv.org/abs/2103.14030
- Other methods: Neural Renormalization Group

25/08/2023

canceled due to many absences

01/09/2023

We start again at 9:00!

Kubeflow (short update)

https://docs.google.com/document/d/1hIwAODam-N5f8H6Sdfkd3RdJ-2rk_CbCU4qZFEpML-M/edit#bookmark=id.pk6kteppc2q2

Currently not a way to go for us, consider possibly for large scale training (but we'd need to make sure we can get those resources first!)

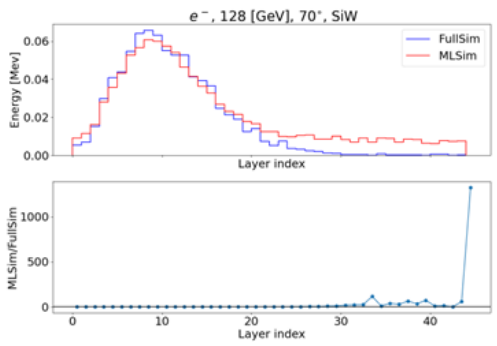
If we need more resources, we should rather try condor. However, current priv machine(s) and openlab's may be sufficient (till next summer).

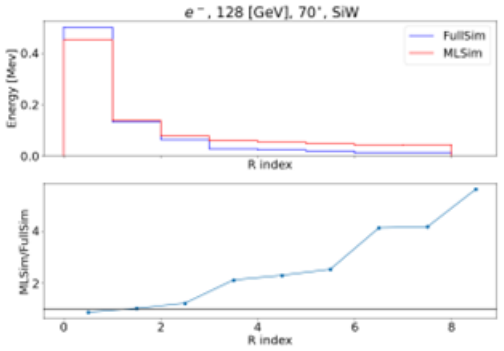
Piyush: CHEP [WIP]
Deadline: **23rd September**

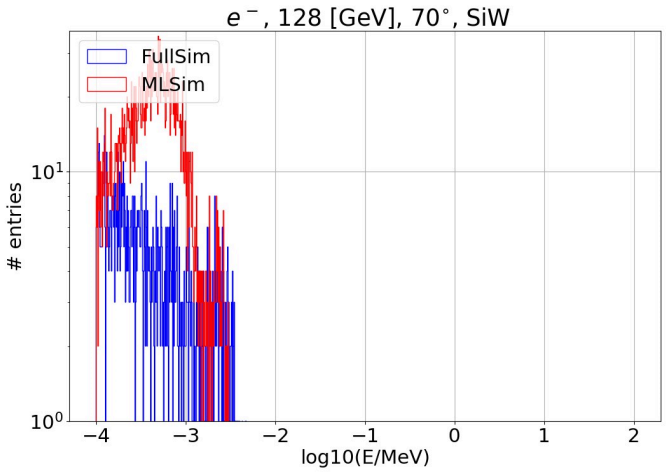
Trained AR models for two different VQVAEs, one w/ low entropy and the other with high entropy. Low entropy signifies low randomness, hence more information. Whereas, for high entropy VQVAE the probabilities over the vocab are more uniform. Yet, the results for AR are more or less the same. **With soft targets and high entropy VQVAE the AR model was a lot worse. But with hard targets, it was similar.**

Renato:

- Added the physics validation plots
- Added energy condition on input (Primary energy)
- Tested preprocessing methods (work in progress)
    - Initial: Pixel energy / max shower energy
    - Due to energy conditions: Pixel energy / Primary energy
    - Tested logit transformation (worst results)
    - Add normalization after logit transformation (how to get back from it?)







- **Next Steps:**
    - Add more elements to the loss / network (extrapolate primary energy, conndition on layer and radius)

Kristina
GCN+t-MLM

- reducing GCN batch size (i.e. update GCN 16x more often, before 8x) - overall better results after 30 epochs - currently running for more
- adding loss component - classification of primary energy - improvement
- moving masking before the GCN - also slightly better
- combining all of these changes together - results in NaNs…
    - need to find out what is the problem
- will update the slides

GraphVAE

- got response from Ali Hariri - scheduling meeting
- started working on funcs to create graphs (feature and adjacency matrix) from the images

No meeting on 8.09 as we meet with DESY on Wednesday at 15:00

Meeting with IBM 14/09/2023

New IBM experts from time series group (Jayant, Kyongmin, Nam)

Question on custom transformer block in pytorch: Mudakhar pointed to the grouped query attention in pytorch

Mudakhar: more data -> perplexity can go down
some problems can go away just by using more hardware

Nam on mixers (TS mixer?) Mudakhar recommends it as a first try with mixers

Mudakhar: asks for target links to the code (some particular aspects: AE,vqvae, ar ?)

pointers from Mudakhar: accelerator transformers, flash attention (from pytorch 2.0), stpe (?) -> targets gpu hardware (io/memory)

Instability:

- Mudakar: bad initialization, preserving variance. Simple check on running const input and checking at each transformer layer the output (variance)
- Nam: design of loss

Entropy:

- homework for ibm:)

KQV dim -> we use 16x8
IBM 4k for lang, 5-12k ts

Make experiments with more data!

15/09/2023

To check: e-group of foundation models permissions. Sofia created a ticket.

Piyush: CHEP paper and AE to VQVAE experiments, no conclusion yet.

CERN-IBM meeting 21/09/23

- Vijay: asked about the CNN, something to revise?
- Kyongmin : mentioned hierarchical approach geometry information (CNN based? something to check)
- Kyongmin proposed the idea of the learnable scale (introducing a bias in the input) -> to think about
- As far as I can tell the TSMixer is an MLP mixer with these 3 successive blocks of rotations with gated attention (which features are important at one time) + res connection-> learn correlations across different dimensions -> seems pretty straightforward to apply to our case
- We need to check the model on hugging face

- Nam: Question on empty voxels, photon appearing.
- Kyingmin: Capturing information across 6 orders of magnitude is a hard task. Need some processing like log, but disadvantage is that it treats 1 and 100 at similar level. Adding learnable bias would help.

Another note: if everything is positive, energy keeps increasing. More prone in resnets.
- Vijay: Question on if flat vs non-flat latent space makes a difference. Meaning, does locality matters?
- Kyongmin: Need to think about rollover in phi. (Should have mentioned cylindrical conv!)
-

29/09/23

Piyush

1. AE - modifying latent space
    a. Previously - 128x450 latent space, might be too large to learn trivial representation link
    b. Reduced to 32x450, some drop in performance link
    c. Trying to increase performance w/ the smaller latent space
2. VQVAE - Nearest neighbour quantization
    a. Default quantization so far
    b. Need to balance commit weight. 0.05 is better than 0.1 and 0.01
    c. Smaller codebook vector dimension eases optimization, converges fast and to a better minima.
    d. Large codebook size helps. 5000 better than 1000. Not sure if 5000 is being fully utilized.
    e. Above things were wrt local latent space. Global latent space converges to lower loss than local latent space, but overfits.
    f. Looking for something in the middle by having local projection and then attention.
3. VQVAE - Gumbel softmax quantization
    a. I tried this before, but was not able to train the model well.
    b. Mudhakar said this one works and scales better usually.
    c. Two terms, classification and kl divergence.
    d. kl divergence set to 0 works better, need to figure out why kldiv is needed.
    e. classification term contains tau (temperature), needs to be carefully tuned to even start learning. As of now, linear schedule to take tau from 1 to 1/16 (DallE). If tau close to 0, means one-hot distribution, if close to 1 (>1?), means uniform distribution.
    f. Training loss goes low if tau is managed properly, but validation loss increases. This is not overfitting! Training forward pass and validation forward pass is different. Validation forward pass always need to use one-hot vectors. Training forward pass have a choice (I think?). Most implementation use soft-forward pass. Other choice, below:
    g. Tweaked with adding a straight-through estimator trick. This uses one-hot vector during forward pass, but treats as soft-vector (with given tau) during backward pass to calculate gradients.
    h. Having straight-through enabled bridges the gap b/w training and validation loss. But then training loss does not decreases much beyond a certain point.

Renato
Restructure of the code and architecture following hugging face architecture:

- https://huggingface.co/blog/annotated-diffusion
- Using linear attention between downsample and upsample and standard attention layers between the mid section of the model

Testing with different hyperparameters/loss
Dataset 2 incident energy vs max shower energy

06/10/23

Piyush
vqvae:

- Nearest-neighbour quantizer
    - Verified initialization
        - Xavier for attention (Check for W_v)
        - He for other layers in transformer
    - Dropout before latent layer helped
    - EMA updates w/ dropout helped more
    - Quantization => variance, soft-quantization?
    - Need to increase expressivity
- Gumbel-Softmax quantizer
    - Static coeff. didn't work
    - Cosine schedule >> Linear schedule for temperature, but still not good
    - Forward pass, one-hot or soft?
        - If soft, gap b/w training and validation
        - If hard, losses don't decrease much

Renato
https://github.com/facebookresearch/DiT

13/10/23

Piyush
VQVAE ideas to try:

1. Commit loss weight recheck (alpha/beta weighting instead of vq/commit)
2. EMA is commitment loss
3. Asses gap b/w AE and VQVAE
4. Data init over whole dataset/multiple batches
5. Follow diff. VQVAE training schemes from prior works (DallE, VQGAN, more, https://arxiv.org/pdf/2005.08520.pdf)

6. Look into making discrete gradient estimates better (Beyond STE) (https://arxiv.org/pdf/2304.08612.pdf, https://arxiv.org/pdf/2205.07547.pdf, https://proceedings.mlr.press/v202/huh23a/huh23a.pdf)
7. Codebook replacement (of unused vectors)
8. Quantization => variance, soft-quantization?
    a. Replace argmin by softmin
    b. Another approach, gumbel-softmin instead of softmin? (cause softmin is not truly categorical?)
9. Along deep adversarial clustering (https://openaccess.thecvf.com/content_CVPR_2019/papers/Ghasedi_Balanced_Self-Paced_Learning_for_Generative_Adversarial_Clustering_Network_CVPR_2019_paper.pdf)

Started looking into diffusion:

1. VQDiffusion - Make AE work (fallback: Make AE w/ MLPMixer/DNN work) -> Use more data in global
2. Decide on the approach by next week

Kristina

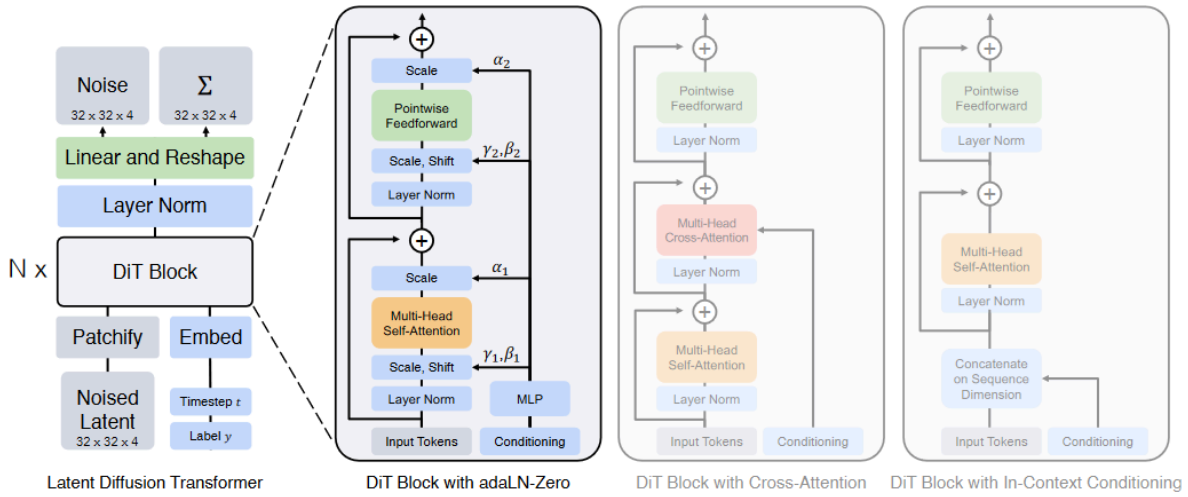
GCN+t-MLM

- trying other losses - combining tanh and ReLU, scaled sigmoid and ReLU
    - worse shower shapes, better cell energy
- GraphVAE
    - spektral package - not suitable because of data format
        - requires feeding data in a custom Dataset class and then using one of pre-defined modes
        - batch mode - requires adjacency matrix in dense format - too big even for one dataset
        - single mode - processes only one graph at a time - too slow to train something
    - In HEP - message-passing GAN for jets (Raghav Kansal - paper 2021)
        - jets with only 30 particles
        - found VAE implementation based on this model - for MNIST
            - does not train to anything reasonable
        - Raghav might be working on scaling up the GAN for larger graphs
            - call scheduled for Tuesday next week

Renato:

- Remaking the preprocessing and postprocessing
    - Add scaling factor (1.5) to compensate for shower being scaled 30%
    - Preprocess energy to be between 0 and 1
    - Corrected bugs on logit and normalization
    - Corrected bugs on validation
- Added cylindrical convolution to the network
- Changed downscale and upscale to use powers of 2
    - use output padding to get it back to the original size
- **Next:** Start looking into DiT

20/10/23



Latent Diffusion Transformer | DiT Block with adaLN-Zero | DiT Block with Cross-Attention | DiT Block with In-Context Conditioning

Renato:

- Initial implementation of the DiT model
    - Using forward process (noising) and sampling process (denoising) from the Unet implementation of the Diffusion process
    - Implmenting a patchify method using Conv3D
    - Implementation of the DiT block (block similar to the ViT block from vision transformers)
        - **Problem:** How to best pass in the conditions (time and energy)
    - Not estimating the variance as of yet

Piyush

- Implemented DiT Diffusion (ViT style w/ in-context conditioning, 3rd fig) for dataset 3 on shower space
- Issue of adding conditions:
    - Needs to be added as a token in each block, hence token number increment with each transformer block
    - Solution could be to have 4 extra tokens at the end (say for 4 blocks), and don't consider them for reconstruction.

    - But the issue is the information usage is not guaranteed?
    - And the position change. Sometimes the condition token is 3rd one, but sometimes the 3rd token is shower patch. (But these are separate transformer block, so not an issue?)
- **TODO**:
    - (DONE) Push the code
    - Implement verification scripts (check forward diffusion, schedular, etc)
    - (DONE) Check adding condition token to each patch
    - Decide on “ideal” preprocessing, scheduler

Discussion

- In-context conditioning
    - Token increments, even if add and remove
    - Scaling issue in softmax, readjust scaling?
    - No position for condition token. How to add?
    - Instead just add condition token to each patch
- How to judge the backward diffusion process?
    - Calculate noise reconstruction loss for each timestep (Addresses schedular, no. of diffusion steps)
    - More diffusion steps, the better? Probably, our large range might make things difficult or the model learn to handle it. 1000 should be enough.
- While inference, diffusion steps can be less.
- Check distribution of x_t to verify forward diffusion process and schedular.
- Preprocessing
    - Normalization is essential

27/10/23

Anna

- Large statistics of the discrete dataset (100k per energy per angle) is almost done, on eos/. It is also produced with the new Geant4 (11.1.p02)
- progressing (slowly) on the NDA with IBM.

Piyush

- Going through diffusion papers
- Ran some experiments on dataset 3, no good results yet
    - ViT like 4 layer transformer arch
    - Adding conditions: add as token, add to all tokens
    - Schedulers: linear, linear w/ diff ranges, cosine, cosine with temperature
    - Different diffusion steps: 200, 400 (default), 1000
- Implemented some scripts: Monitoring loss for each diffusion timestep, forward diffusion plots

Renato

- Correction of bugs:
    - Patchify and Unpatchify
    - Attention
- Experiments on dataset 2:
    - Decrease of embed_dim to 144
- Going through cold diffusion paper
- **TODO:**
    - Run with dataset 3 - calo-challenge
    - Look into best methods for conditions

Kristina

Call with Raghav

- his usecase: MP-GAN for jets (used for 30 particles)
  - he used fully connected graph
  - https://arxiv.org/abs/2106.11535
- switched to transformers (currently on 150 particles per jet)
- link to a paper from CMS - graph network for reconstruction on HGCAL - clustering
  - approx 2000 nodes per graph
  - https://cds.cern.ch/record/2803236/files/2203.01189.pdf

Meeting with IBM 16/11/23

Slides are on slack,
Very nice intro to work done by IBM, with VAE from deterministic decoder to probabilistic, testing different functions as approximators of the second latent space, conclusion that a mix of all (Gamma, laplace, gaussian) may give the best results. At the next meeting we will talk more about the diffusion models, with updates from Renato but also some notes and background from IBM.

17/11/23

Ml4jets conference
https://indico.cern.ch/event/1253794/timetable/#20231106
Lots of contributions focusing on generative models.

Anna
I will make sure that we have even larger dataset, CaloChallenge one is almost done (also used for validation that Claudius does),
Discrete values - I can extend it even further, and also start with one angle

Renato
Runs to do:

- Run plots for all the energies
- The last priority: Full image space (40500 voxels) run instead of the decreased space
- [d2] run with larger statistics and try to improve scaling of energy by introducing loss to the variance estimation etc.
- [d2] also add plots for diff E
-

24/11/23

Renato:

01/12/23

Renato:

- Fixed problem with cosine scheduler and run with cosine scheduler
    - Run: https://wandb.ai/redacost/default_study_name/runs/11ldioec?workspace=user-redacost
    -
- Added variance estimation and run:
    - Run: https://wandb.ai/redacost/default_study_name/runs/3vm8j21y?workspace=user-redacost
- Added timestep sampling according to loss:
    - To be run
- To run a check on interpolation. Exclude region from 64GeV to 256 GeV and then ask for 128 GeV at validation
- dataset3 save for later, focus on d2
- First extend with angles on Par04 dataset, generate (Anna) cont angle distribution with phi and theta, generate total of 2.5M per detector, we will check if this stst is sufficient
- Always save output of preprocessing to save memory

Anna:

- continuous dataset is there for d2 and d3 (1M)
- discrete still requires a fix to h5 translation [WIP]
- License question sent to OSPO

Meeting (unusual) on Monday at 10:00

08/12/23

TODO: discuss ACAT abstract

Old code, Cosine (Piyush's run)
https://wandb.ai/foundation-models/debugging/runs/22x3c3cr/workspace?workspace=user-piyush_555

Renato:
2x2x2 cosine test:
https://wandb.ai/redacost/default_study_name/runs/1iqbi1x8?workspace=user-redacost
3x2x3 cosine continued:
https://wandb.ai/redacost/default_study_name/runs/13yeftnd?workspace=user-redacost
3x2x3 loss aware:
https://wandb.ai/redacost/default_study_name/runs/36d4viji?workspace=user-redacost
3x2x3 interpolation:
https://wandb.ai/redacost/default_study_name/runs/1wlx9lj4?workspace=user-redacost

TODO priority list:
DONE Piyush to give latest model
DONE Renato start with Piyush's branch, add generation code
DONE Piyush make a singularity image and test
DONE Renato generate h5 and run Calo Challenge validation
-> by afternoon today let's see where we are with all + check new epochs
if not improved, let's see if we can update over weekend, but let's make sure all of us have the files to send to Claudius (on g4fastsim afs or eos)

What we submitted to CaloChallenge: We miss the last R z layers due to even patching. This is now corrected but not submitted.

The rest:
paper CaloChallenge - WIP, added pic
wait for cos + var+loss, and run cos+loss to see the ultimate model →**for now we will use cosine only, we can keep in mind var+loss for later tests (e.g. with more noise steps), but for now they just make training longer**
then rerun with fully transformer based for dataset 2 for chosen combination (R)
[**Move to 2024**] Piyush to run on 200-300k d3 once we have the final choice
…
[2024, wait for optimised model] we introduce phi and theta, run d2 on it, check
[2024] then we produce ODD or FCCee and run on it with single theta and different phi
[2024] if all works we go to generalisation and multigeometry training

ZOOM chat 19.12.2023

Piyush to rerun d2, fully transformer on 1M, 128 batch size (or both 128 and 256)
Renato start hyperparam training on 1M samples (update Optuna):

- batch size 16 to 512 in pow of 2
- patch size (number) 2x2x2 to 4x4x4 and combinations in between
- num of noise steps 100-200-400-700-1000
- num of DiT blocks 1-2-4-8
- num of attention heads 4 to 128 (just pow of 2)
- embed dim (decouple it from att heads) 144/4=36 per head, go 16 to 128 in pow of 2
- learning rate (?) 1e-4 to 1e-2 continuous

Anna to rerun large stat with phi and theta so we can run multiple conditions in 2024
Papers: Piyush to write vqvae and check diffusion
CHEP 2023 abstract ->

“”

Recently, transformers have proven to be a generalized architecture for various data modalities, i.e., ranging from text (BERT, GPT3), time series (PatchTST) to images (ViT) and even a combination of them (Dall-E 2, OpenAI Whisper). Additionally, when given enough data, transformers can learn better representations than other deep learning models thanks to the absence of inductive bias, better modeling of long-range dependencies, and interpolation and extrapolation capabilities. Therefore, the transformer is a promising model to be explored for fast shower simulation, where the goal is to generate synthetic particle showers, i.e., the energy depositions in the calorimeter. The transformer should accurately model the non-trivial structure of particle showers, as well as quickly adapt to new detector geometries. Furthermore, the attention mechanism in transformers enables the model to better learn the complex conditional distribution of energy depositions in the detector. In this work, we will present how transformers can be used for accurate and fast shower simulation, as well as the know-how on transformer architecture, input data representation, sequence formation, and learning mechanism.

“”

Abstract ACAT 2024 -> change to diffusion and rewrite a bit (Renato), we need to check with IBM regarding authors.
Stress that there is many diffusion models now, but what we aim at a generalizable one.

22/12/23

Renato:

- Abstract for ACAT:
    - https://docs.google.com/document/d/16knj7G6ewoZQyYTV-mx4V48n0zmXVC644g6IgObc4Xs/edit?usp=sharing
- Set up hyperparameter search
    - Limits for the hyperparameters
    - Timeout?

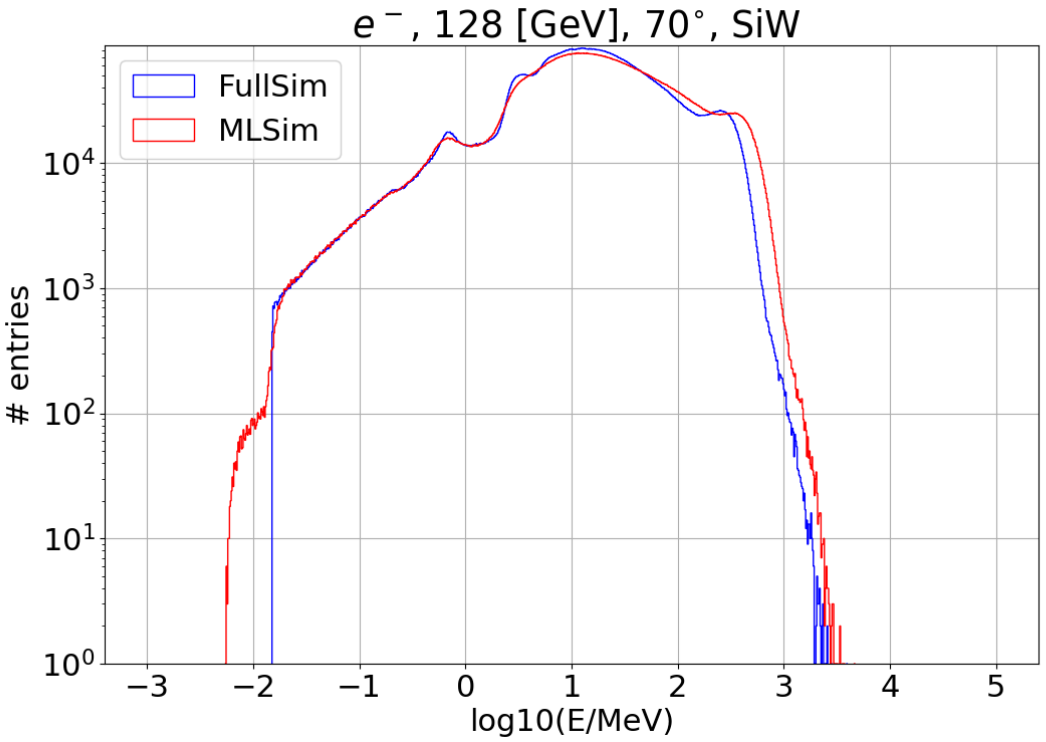
12/01/24

Renato:

- Timeout per epoch

Piyush

- VQVAE with Renato's model **(EDIT: This was AE due to a bug in the script)**
    - Difference is a that it has a lot of channels, thus huge projection for a patch. This enables multiple pathways for the model to learn varying representations (which is apparently essential for when the input has long range?). And also no bottleneck.
    - So the latent layer is 8x the input.
    - But as long as we can generate samples, it should be fine



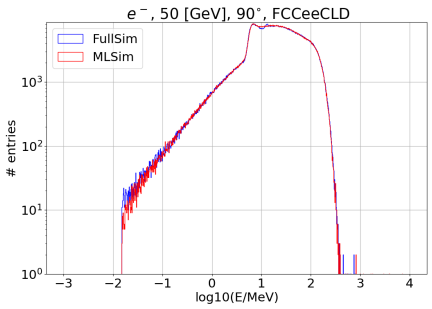
- TODO:
    - Transformer model VQVAE (DONE, not so good results)

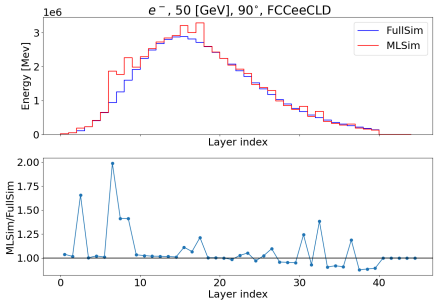
- Conv arch VQVAE
- AR on it
- VAE without bottleneck (take care of conditions) (Half done)

19/01/2024

Anna

- datasets
  - Par04 reimplemented in key4hep (ddodd), and validated, awaiting 1M events for eta -1 to 1, full phi, and energies 1 GeV to 1 TeV
  - FCCee CLD 1M generated
    `/eos/geant4/fastSim/ddodd/FCCeeCLD/1GeV100GeV_eta0_phi0/ddsim_mesh_FCCeeCLD_gamma_100kevents_1GeV100GeV_eta0_phi0_edm4hep_part1.h5`
    we have `part1 - part10` files of 100k showers each
  - FCCee ALLEGRO 0.5M to be checked if generated, needs validation and translation to h5
- training
  - I use condor! Need to confirm if I am able to fit 1M in RAM (I use 8 CPUs = 16 GB RAM for 500k showers now)
  - FCCeeCLD on 400k sample (+100k validation)
    cell energy v good, but profiles have spikes -> try running on 1M….





Piyush

- Great results with AE within just few epochs
- Failed to reproduce those for VAE or VQVAE
  - Tried diff latent space sizes
  - Tried diff arch
  - Tried diff hyperparameter for VQVAE and VAE
- Tried 1M data with VQVAE but could've been bad hyperparameters
- Can do:
  - Hyperparameter tuning on 1M data
  - Only GAN
  - Then add GAN to VQVAE
  - Global latent layer

26/01/2024

GSoC -> to be prepared on inference optimisation

Renato

- Run estimation on x and noise
    - https://arxiv.org/pdf/2107.00630.pdf
    - https://arxiv.org/pdf/2202.00512.pdf

Anna

o test float 16 on dataset and FCCeeCLD

Piyush

- Some directions along VQVAE
- Global latent space with no bottleneck improves things
- Gumbel softmax (GS) also works better now
- Custom quantization (GS with traditional VQ) increases codebook usage

TODO

test mix precision and/or float16 (for dataset and for model)

1. translate to h5 all Par04 data with 3 conditions
2. generate a big sample for ODD
3. add conditions and test
4. train on at least 2 geometries and adapt on 3rd → minimum for ACAT
5. implement bigger datasets re-read from disk (or sth)
6. Compare (total n samples=const, we change k*M where k is num of detectors) adaptation capability to a new detector eg in terms of steps/M or time

09/02/2024

GSoC - submitted

Piyush

TODO

[done almost] test mix precision and/or float16 (for dataset and for model)

for dataset it gives good results, we can use float16, then the RAM on CPU is halved and we can fit more data

mixed precision on GPU (model) did not give significant improvements, maybe we do not use float16 for too many weights

7. [done almost] translate to h5 all Par04 data with 3 conditions

    previous production had a bug in ddsim implementation, now it's re-runing on condor

8. generate a big sample for ODD
    a. to be done, once previous finishes
9. add conditions and test
    a. how to preprocess? theta as energy, for phi ensure continuity - sin and cos

    b. run a test on not normalising conditions since time is already from 0 to 400
10. train on at least 2 geometries and adapt on 3rd → minimum for ACAT
11. implement bigger datasets re-read from disk (or sth)
    a. Piyush is already investigating
12. Compare (total n samples=const, we change k*M where k is num of detectors) adaptation capability to a new detector eg in terms of steps/M or time
13. Peaks in the average profiles -> explore other schedulers (learnable as we did not see peaks in linear, we see it in cosine) or stretch cosine scheduler not to do the last step that possibly produces the peaks in the distribution

16/02/2024

Testing the angle conditioning:
[todo] modify validation

- add theta filter (and condition) : theta = 1.47-1.67 (how much full sim we have? make it smaller as long as we have 1k)
  theta = …[to be checked]
  phi = 0, phi=0.2 [to be checked, what is the num for ODD)
  E = 50, E = 500
  it's a matrix -> 8 validation points

Tests to run: [**IMPORTANT: STORE ALL CHECKPOINTS**]

1) Par04 with more data and all conditions
   a) RESULT: https://wandb.ai/foundation-models/Diffusion_Par04/runs/q03pka6c?workspace=user-piyush_555
   b) 500 GeV is worse than 50 geV (we did expect that, we will try to run training with flat distribution)
   c) 50 GeV looks like before, profiles have occasional spikes
   d) theta in the middle shows better results
2) ODD with all conditions
   a) result:
   b) same observations than before (50 vs 500)
   c) but profiles still do not look OK , large spikes
   d) if NOT fixed with training: let's try more data
3) Par04+ODD joint with a new one-hot-vector encoding for geo
   a) started
   b) do a second one with normalization per dataset
4) adaptation, e.g. FCCeeALLEGRO
   a) works great : cool, we can generalise; compare to training from scratch, training from the checkpoint of a most-similar detector, …
   b) does not work:
      i) check with CLD trained from scratch vs CLD started add checkpoint from odd (point 2), does it offer any speed improvement?
         (1) yes: we can release several trained models
         (2) no: we do not need to bother, we just release code in Par04
      ii) check other variations of diffusion, e.g. image estimation instead of noise, as well as hopefully we can test other models from ibm

01/03/2024

adding comments to the previous notes above (and colours), from 16.02
TODO

1. change titles of plots to correspond to detector etc
2. run Par04 flat training
3. scaling of energy in preprocessing - right now it's arbitrary, we need to change it to dataset ? let's see the result of the first joint training
4. Normalize wrt different geometries
5. Plots: zeros dist, lat, long, cell, tot

ACAT presentation: https://indico.cern.ch/event/1330797/contributions/5796591/

15/03/2024

5/04, 12/04 still on Friday, afterwards move to General ML meetings for SFT: starting 25/04 at 9:30 (biweekly for starters, maybe weekly with summer students).

TODO:

- do the preprocessing study indifferent datasets to figure out scaling
- merge flat production to check training

Next week hackathon https://indico.cern.ch/event/1307202/
(AIDAinnova)

5/04/2024

Flat vs power energy spectrum: **no significant difference!**
We decide then to use power spectrum since it should offer smaller simulation time (full sim).

Data validation of samples completed

- Par04 SiW (also 1M training data: flat and power)
- Par04 SciPb [potential for adaptation candidate]
- Par04 PbWO4
- ODD (also 1M training data: flat and power)
- FCCeeALLEGRO (1M samples for a single angle)
- FCCeeCLD

Repositories:

- CaloDiT: clean up our current repo, fix conversion using scripting
    - clean up master/main branch that can be passed to LHCb
    - move noise vs x etc to branches
    - document it all on the website
- IBM's repo -> to become our main working repo, but we need to clean up first
    - adding CaloDiT
    - changing the dataset to cont and all 3 conditions

Adaptation:

we need to understand first the preprocessing

12/04/2024

Technical meetings **will continue** on Fridays. On Thursdays we can present highlights.
TODO Piyush: book a room beyond 10.05 and ensure access to indico.

Flat vs Power: ODD to be verified, but as seen yesterday at IBM meeting is likely better (on Par04 500 GeV, profiles)

Renato:

x0 prediction on ODD does not reproduce dips in the profiles (whily noise prediction does)