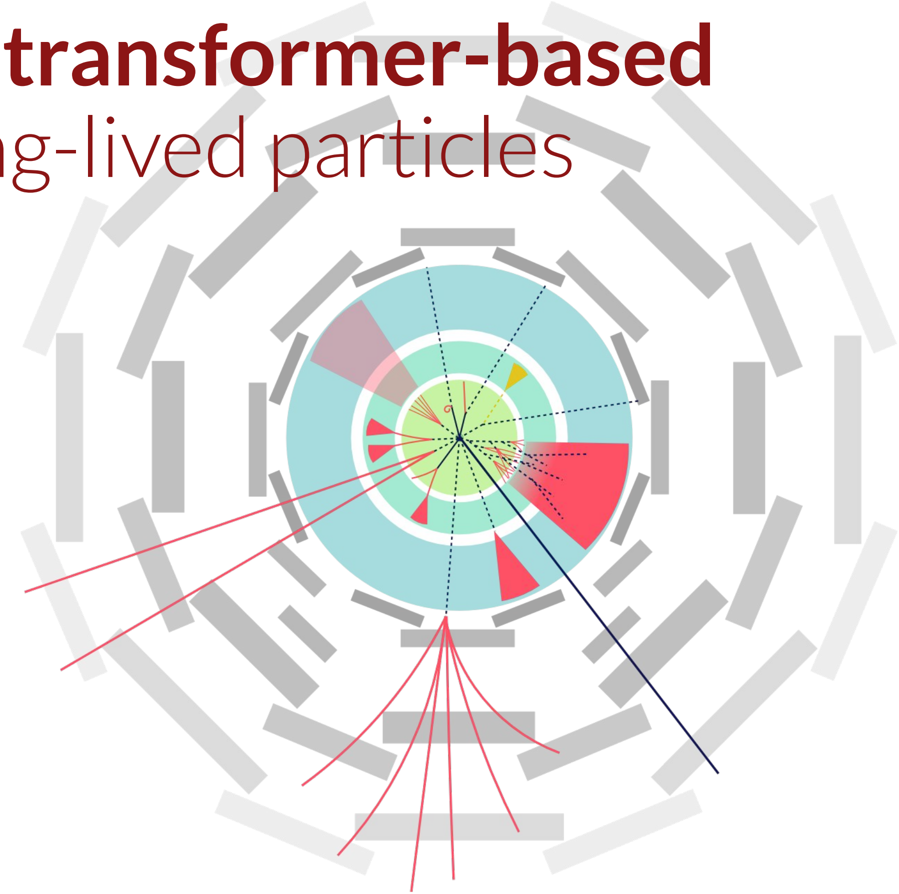
Fast pileup synthesis and **transformer-based anomaly detection** for long-lived particles



SLAC ATLAS Group Meeting

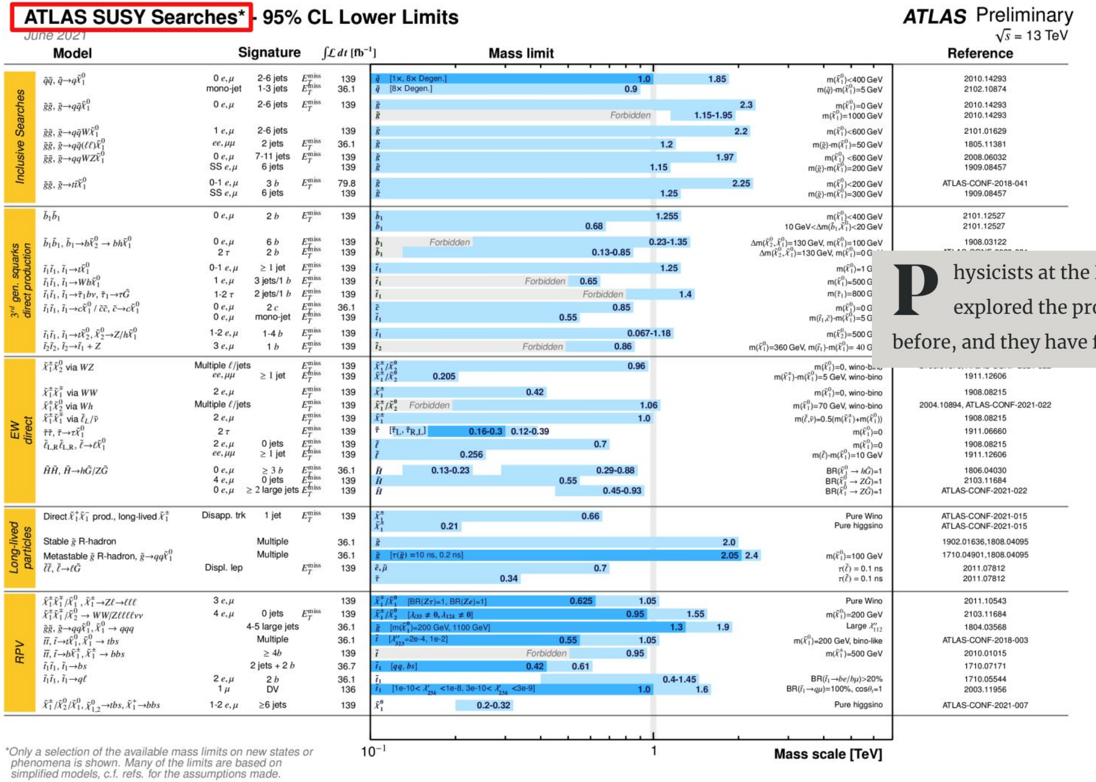
Mar. 22, 2024

Sam Young, rotation student

Stanford University

SLAC NATIONAL ACCELERATOR LABORATORY

ATLAS' rich search program for new physics



Physicists at the Large Hadron Collider (LHC) in Europe have explored the properties of nature at higher energies than ever before, and they have found something profound: nothing new.

quanta magazine, 2016

ATLAS’ rich search program for new physics

Perhaps we’re looking in the wrong spots or for the wrong models?

→ Need to safeguard against missing new signs of physics

Reformulating the question

"Does this event look like BSM theory XYZ?"

↓

"Does this event look like the Standard Model?"

↓

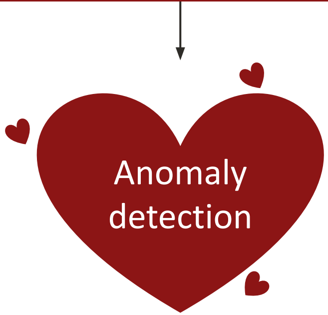
Anomaly detection

Reformulating the question

“Does this event look like BSM theory XYZ?”

Talk focus:

Can we correctly identity anomalous events containing long-lived particles versus regular SM events?

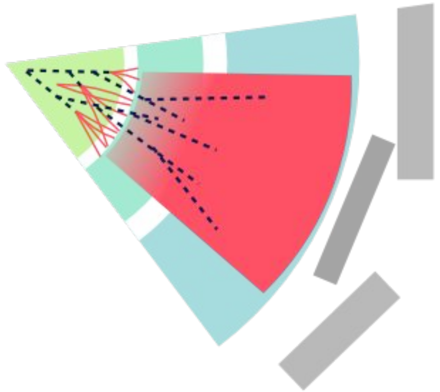


Outline

- Long-lived particles
- Dataset overview
- Fast pileup synthesis
- Event-level classification comparison (MLP v. Transformer)
- Future work

Long-lived particles

- Visible displacement of track or vertex
- >$O(10)$ µm decay length ($c\tau$)
- **DVs not saved by hardware trigger** → can we find a way to determine delineate QCD jets from BSM events using low-level information?



Can larger architectures model low-level data well?

- In a single event at the LHC:
  - $O(100)$ vertices
  - $O(1000)$ tracks
  - $O(10000)$ hits
- In recent years, more and more complex models like the transformer have been used to model ever-more-complex high-dimensional data to incredible success.
- **Goal: moving from high–level jets to low-level tracks, can we adapt these massive models to search for anomalous signals?**

Datasets

Signal (LLP)

- 200,000 total events
- p p $\longrightarrow \tilde{\chi}_3^0 \; \tilde{\chi}_3^0$
- With pileup (μ=60)
- Two $\tilde{\chi}_3^0$ rest masses: 100 & 500 GeV

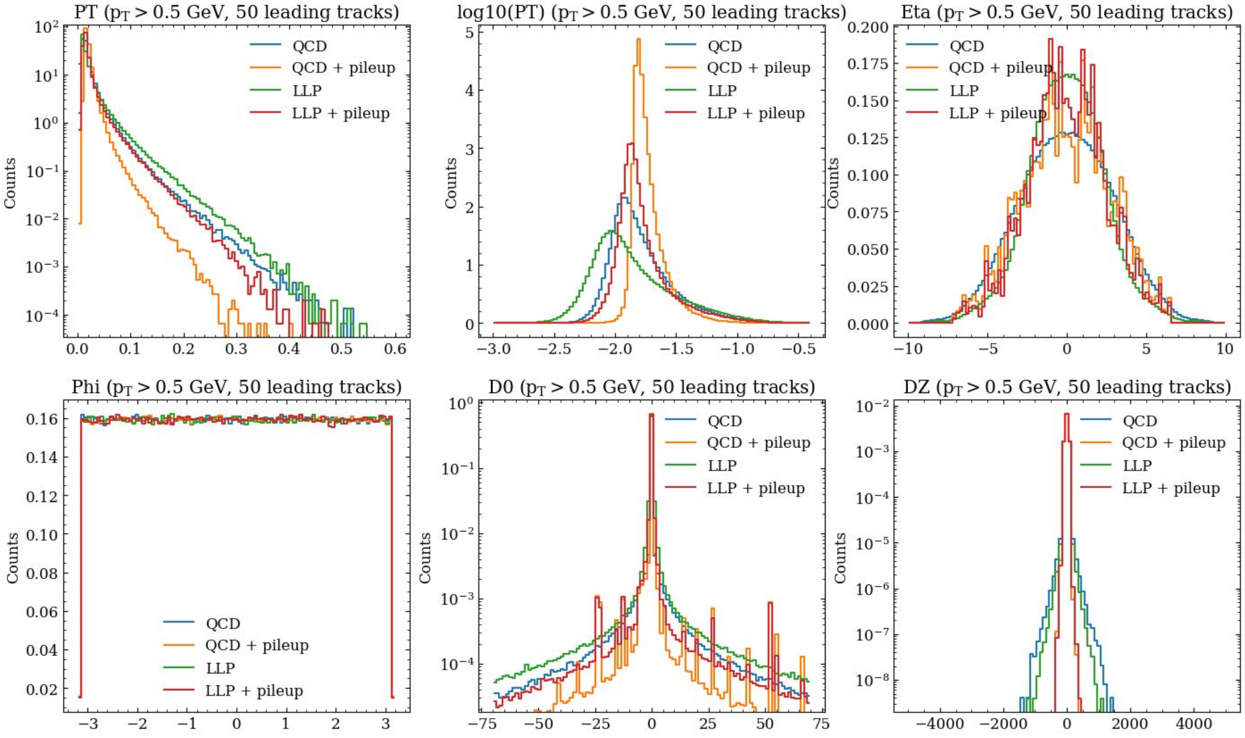
Background (SM)

- 200,000 total events
- p p $\longrightarrow$ 2-5 j (pure QCD)
- With pileup (μ=60)

Features

- Each event contains a number of tracks parametrized by (p_T,η,ϕ,d_0,d_z).

Track Parameter Distributions



Datasets

Signal (LLP)

- 200,000 total events
- p p $\longrightarrow \tilde{\chi}_3^0\ \tilde{\chi}_3^0$
- With pileup (μ=60)
- Two $\tilde{\chi}_3^0$ rest masses: 100 & 500 GeV

Background (SM)

- 200,000 total events
- p p ⟶ 2-5 j (pure QCD)
- With pileup (μ=60)

Features

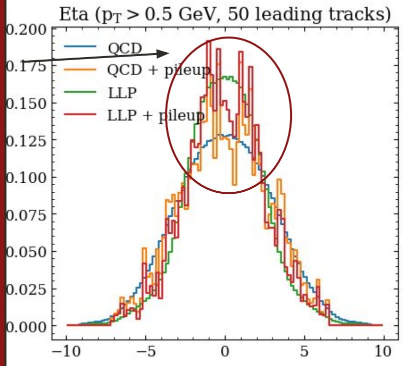
- Each event contains a number of tracks parametrized by (p_T,η, ϕ,d_0,d_z).

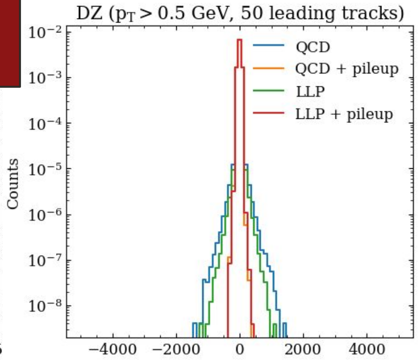
Background Track Parameter Distributions

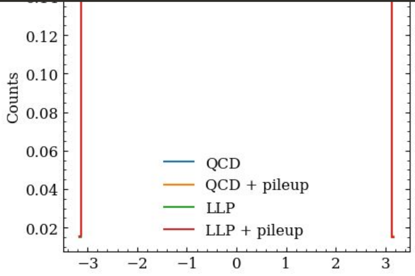
Delphes comes with just 1,000 pileup events to sample from, leading to oversampling by a factor of ~24,000x…

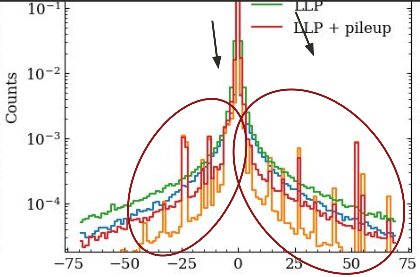
⟶ Need way to simulate ~24 million independent pileup events.
⟶ Simulating pileup is computationally complex. Is there a time- and compute-efficient way to create a synthetic pileup dataset?

⟶ We look into the use of hierarchical gaussian mixture models (HGMMS).









Gaussian Mixture Model (GMM)

- To capture event-level correlations, we model each **individual event** track parameter distribution by a weighted mixture of multivariate Gaussians parametrized by means, covariance matrices, and weights:

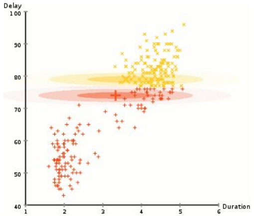
$$p(\vec{x}; \{\mu_i, \Sigma_i, w_i\}_{i=1}^{N}) = \sum_{i=1}^{N} w_i \mathcal{N}(\vec{x}; \mu_i, \Sigma_i, w_i), \quad \sum_{i=1}^{N} w_i = 1.$$

- We use a model selection heuristic Bayesian information criterion (BIC) to choose the number of Gaussians to model each event.

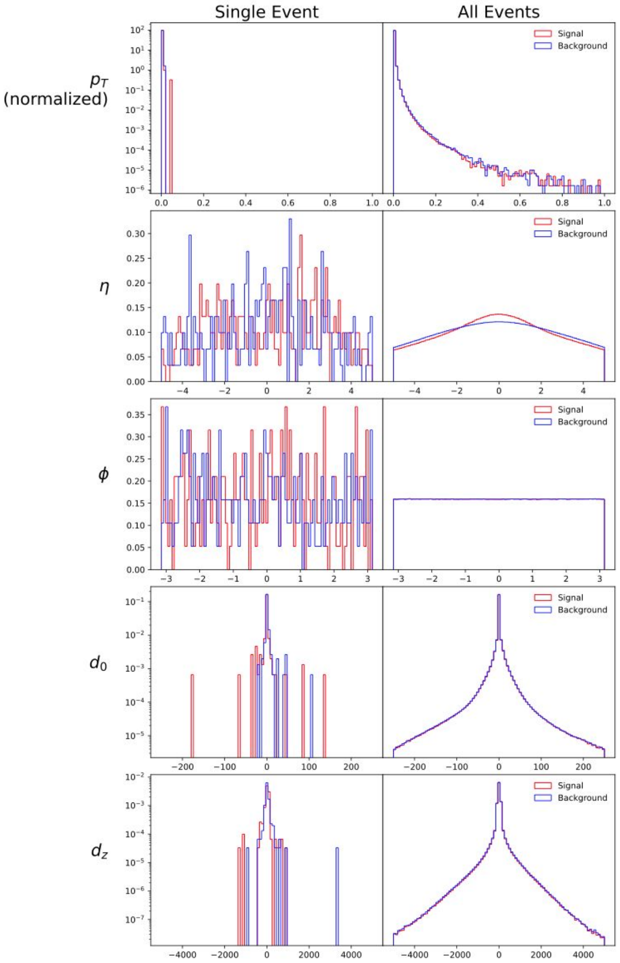
$$BIC = k \log n - 2 \log \hat{L}.$$

k is the number of model parameters, n is the number of tracks, and $\hat{L}$ is the maximized likelihood using the best-fit parameters.

- Balance model complexity with overall fit.

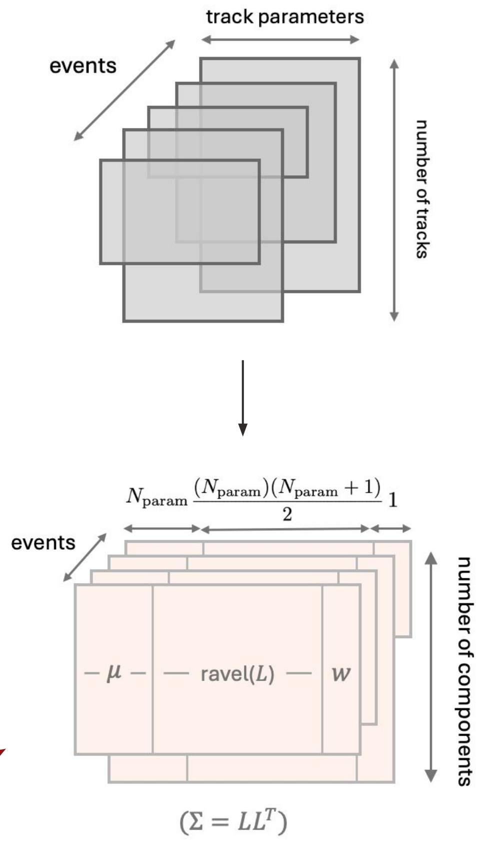


Example of two-component GMM being fit to a two-dim. dataset.



Introducing hierarchy

- After fitting each event's track probability distribution to Gaussian mixtures, we fit a **high-complexity Gaussian mixture to the distribution event-level track probability distributions** across all events.
- To synthesize new pileup events, we sample from this high-level Gaussian mixture to synthesize a new event-level probability distribution, which is then sampled from to create a variable distribution of particle tracks.



sample new track distributions from a GMM fit to all components

Hierarchical Gaussian Mixture Model (HGMM)

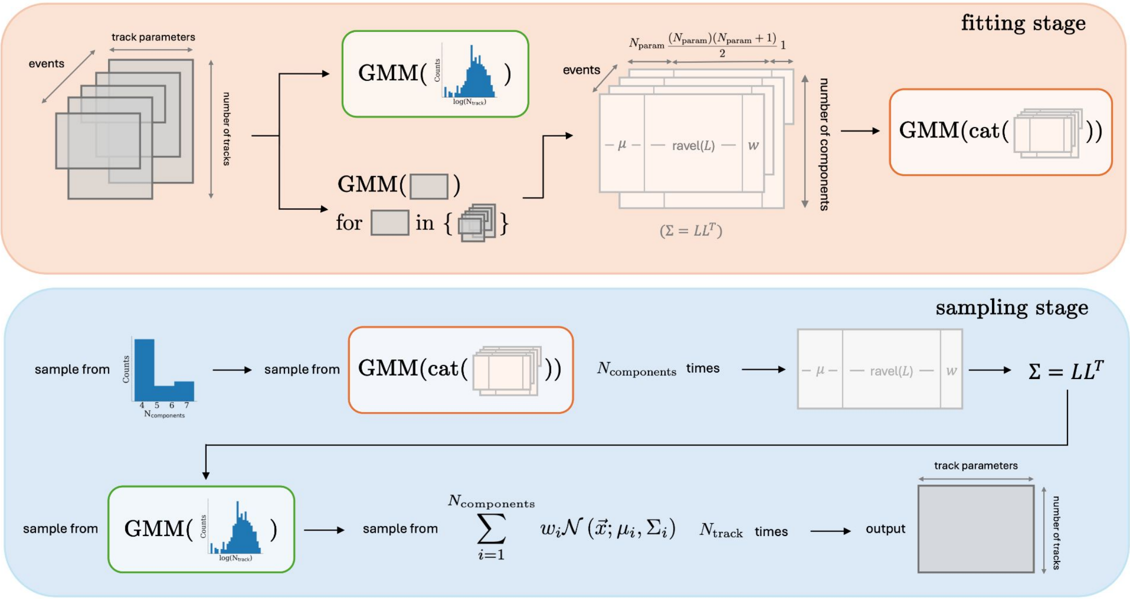
Pros

- Simple idea
- Relatively cheap to sample from compared to actual simulation and non-parametric methods (like KDE)

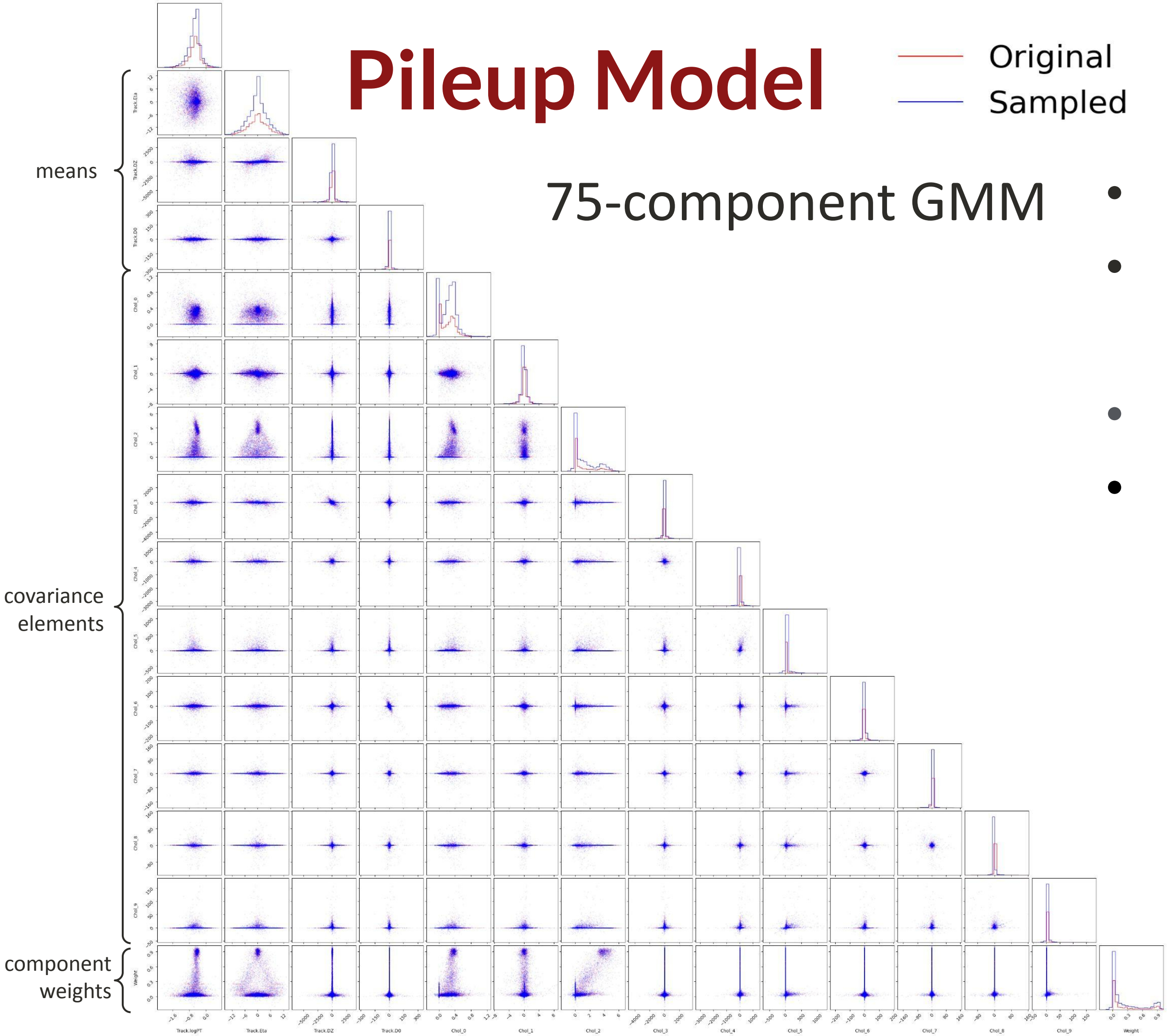
Cons

- Assumes track dist's are linear sums of a few multivariate Gaussians (extreme simplification)

Pileup synthesis using a Hierarchical GMM



Pileup Model

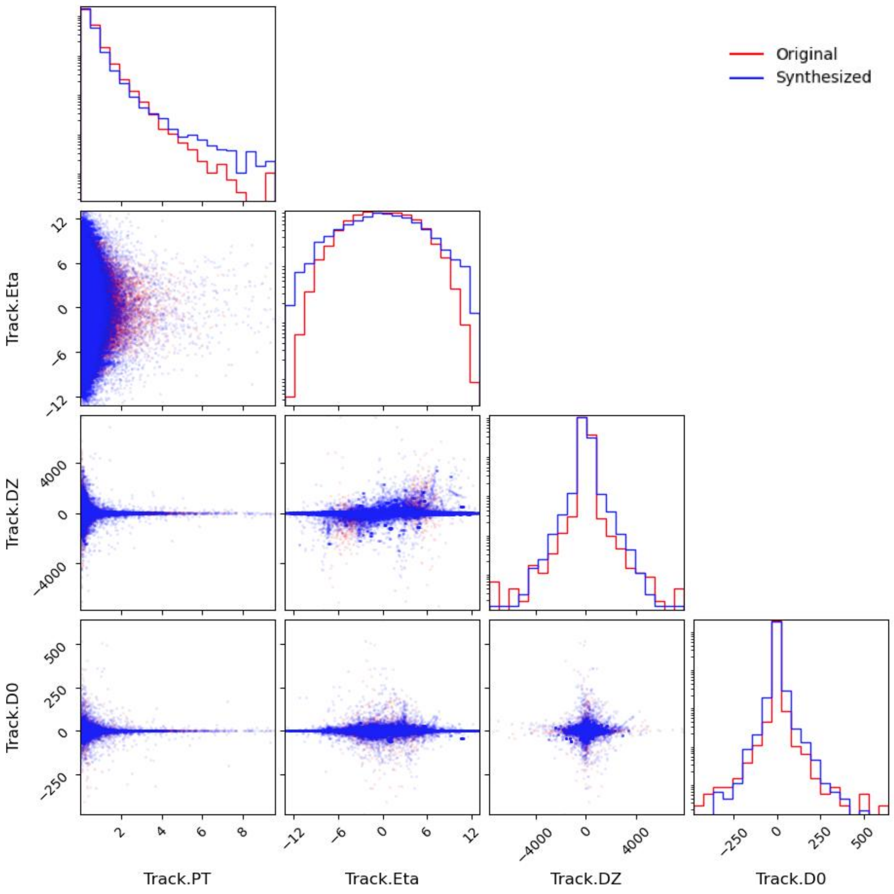


75-component GMM

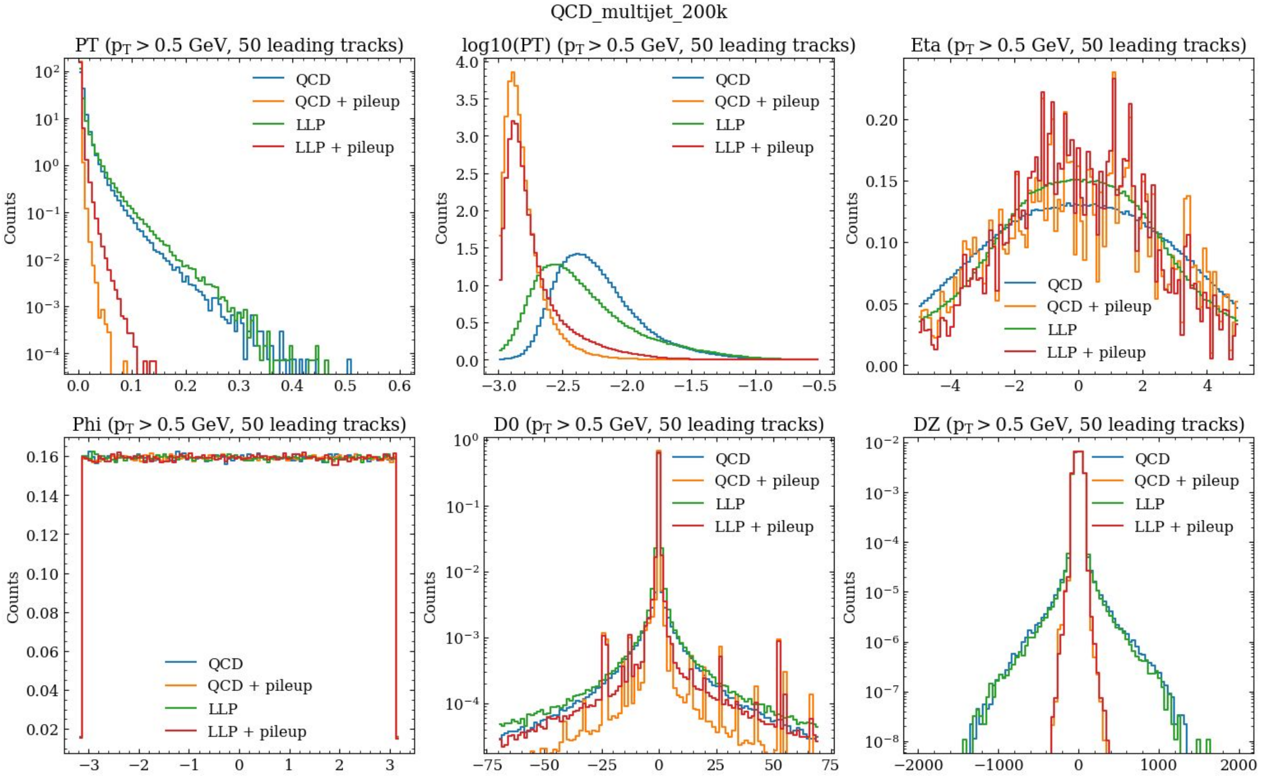
- Ignore histogram scaling (synthetic is doubly sampled)
- Covariance elements are Cholesky decomposed:
  - $\Sigma = LL^T$
  - 16 elements ⟶ 10 elements
  - Ensures positive semi-definite nature of Σ
- I have assumed ϕ is isotropic due to it being difficult to model over 1000 events.
- **The point: if we believe that Gaussians model event probability distributions well, we can very effectively model all possible event-level track probability distributions.**

Pileup model (cont.)

- Despite explicitly modeling it, the “global” track probability distributions are well-modeled
- However there are high-covariance “speckles” → overfitting, covariance allowed to be too small for some parameters.
  - Possible fix: scale all parameters to zero mean and unit variance and such that fit covariance matrices have the same scaling between parameters, then clip the covariance to $|\Sigma_{ij}| > \varepsilon$
  - Possible fix: fit HGMM to more pileup events.
- Nonetheless, we use this HGMM to synthesize pileup events for our signal/background datasets.

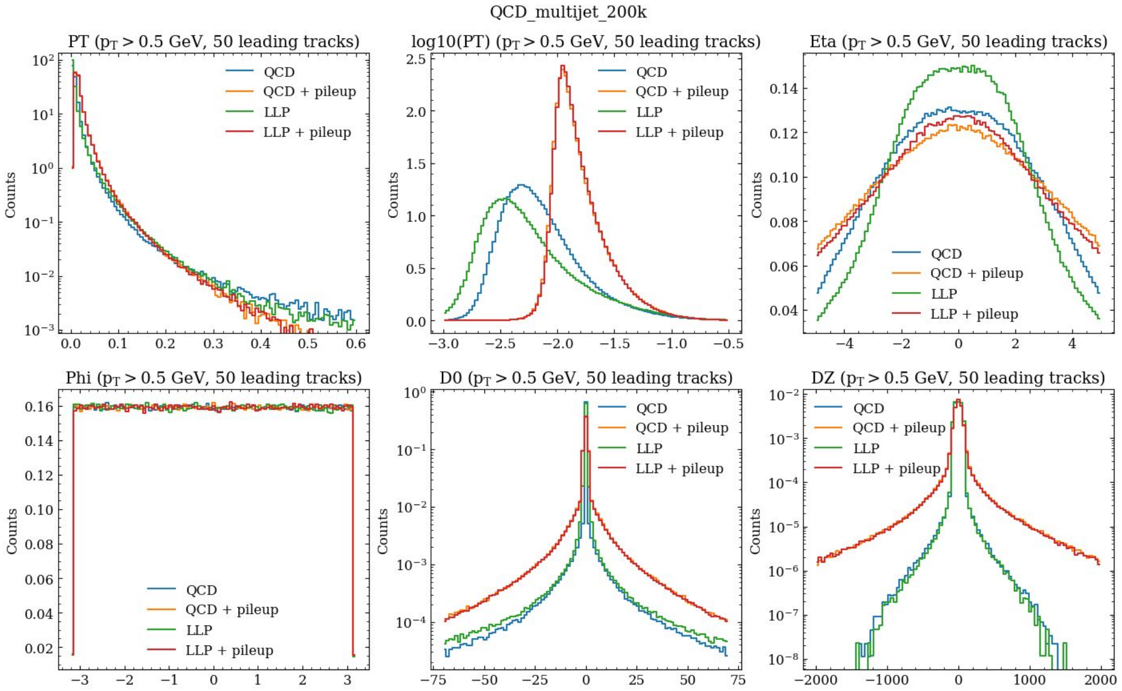


Before: input parameters + oversampled pileup



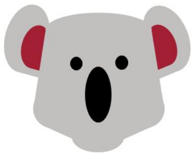
After: input parameters + synthetic pileup

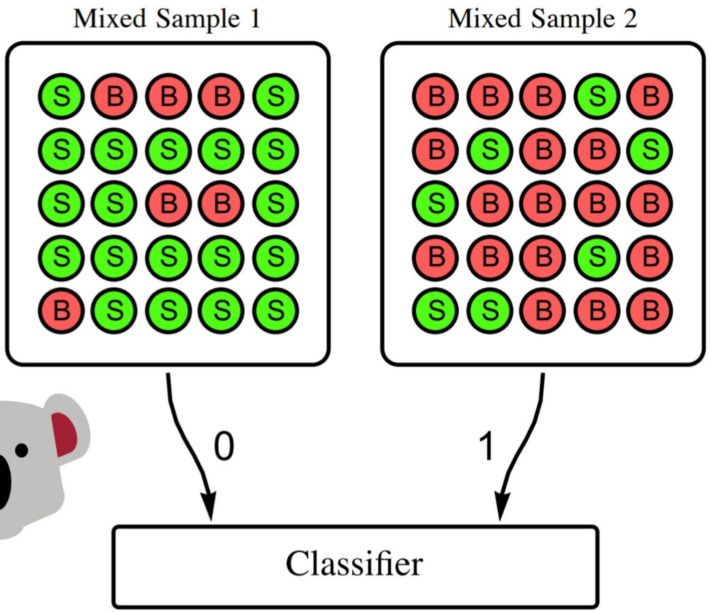
- Issue:
  - Clearly pileup distribution tails aren't well captured (esp. d_0, d_z)
- Possible reasons:
  - We are only fitting 1,000 events and `upscaling' it thousands of times over. Running more events into this model could improve it.



Dataset preprocessing:

- For each training, we apply three cuts:
  - $p_T > 500$ MeV
  - $|\eta| < 5$
  - Take the 80 tracks with highest p_T
- Events are labeled as either 1 (containing LLP) or 0 (pure QCD), meaning that our classifiers are actually being trained on **mixed samples** (a la CWoLA) of tracks.
  - This is what we'd actually see in the LHC, since there's no clear "truth" label anymore when working with tracks.





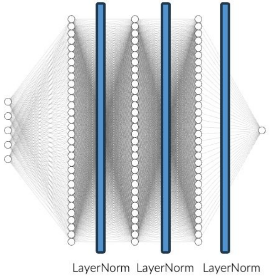
https://www.ericmetodiev.com/publication/classificationwithoutlabels/

Model architectures:

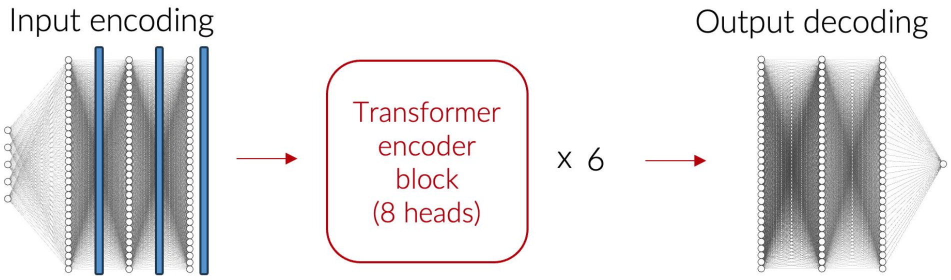
- **Goal: Classify an event of tracks as either containing LLP(s) or not.**
- We compare two models classifiers:
  - A simple multi-layer perceptron (MLP) with mean reduction along tracks.
  - Transformer-based model
    - MLP encoder (same arch as above)
    - Transformer encoder
    - MLP decoder
- Loss function: binary cross-entropy

$$BCE(X;Y) = \sum_{i=1}^{n} \left\{ y^{(i)} \log x^{(i)} + (1 - y^{(i)}) \log(1 - x^{(i)}) \right\}$$

MLP



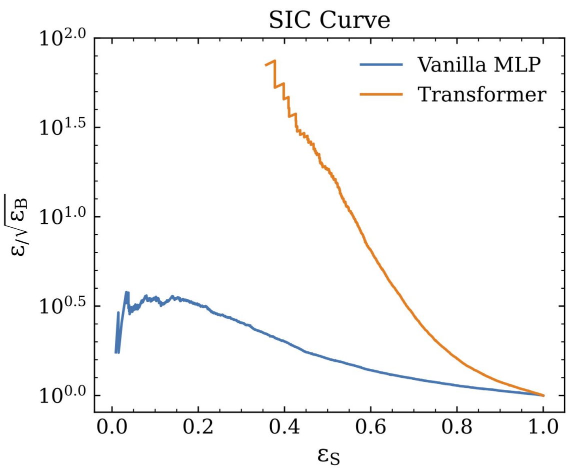
Transformer-based

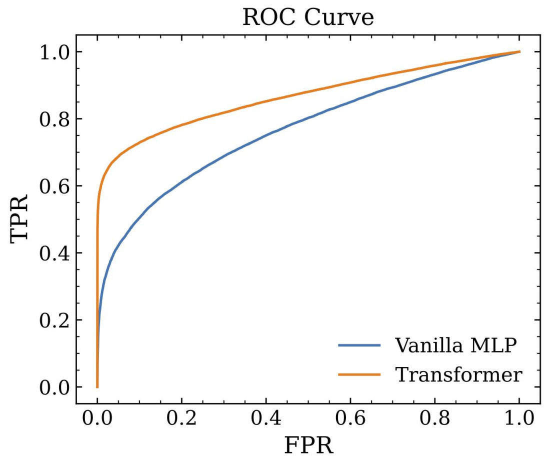


Model performance

- ➔ **MLP:**
  - ◆ 70.7% validation accuracy
  - ◆ 0.768 AUC
- ➔ **Transformer:**
  - ◆ **81.4% validation accuracy**
  - ◆ 0.860 AUC

- ➔ Side note:
  - ◆ Without pileup, classifying between both datasets is trivial (a simple p_T cut gives ~70% accuracy)





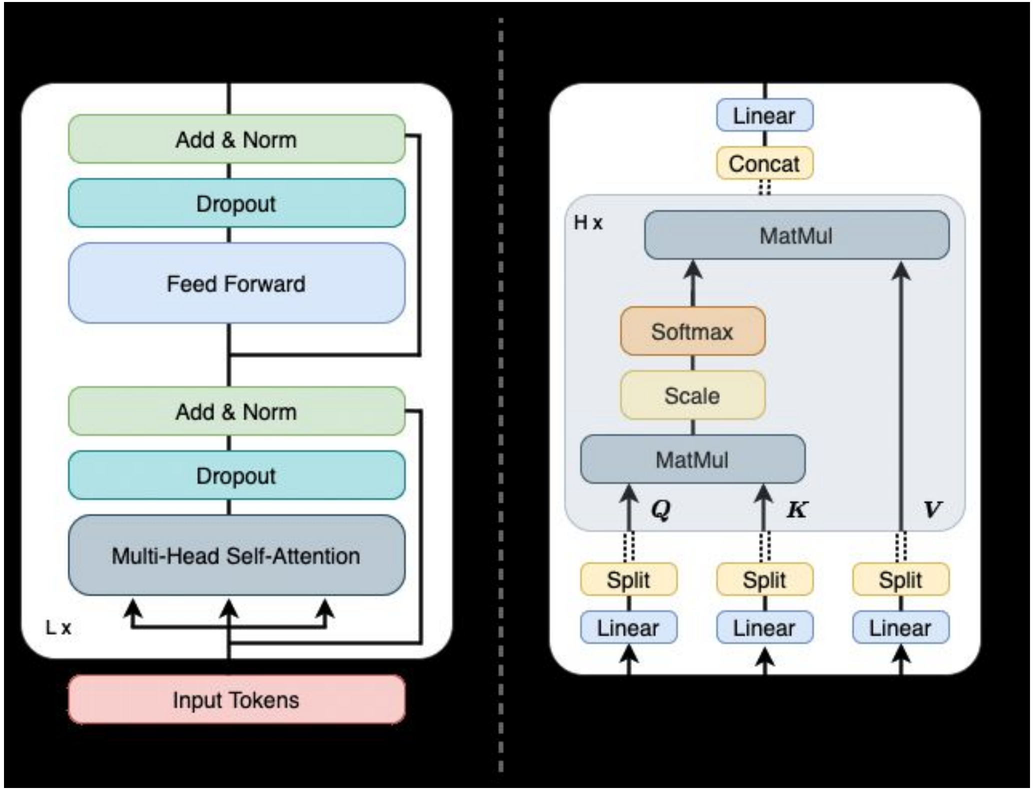
Summary/lessons learned

- Classifying QCD v. LLP events is trivial if there's no pileup
- Modelling pileup using multivariate GMMs is a simple idea but nontrivial in practice.
    - It's better to just simulate the extra events for smaller studies, but as of now simulating huge amount of pileup is hard.
- Transformers can outperform simple MLPs in a high-dimensional task like this.

Backup

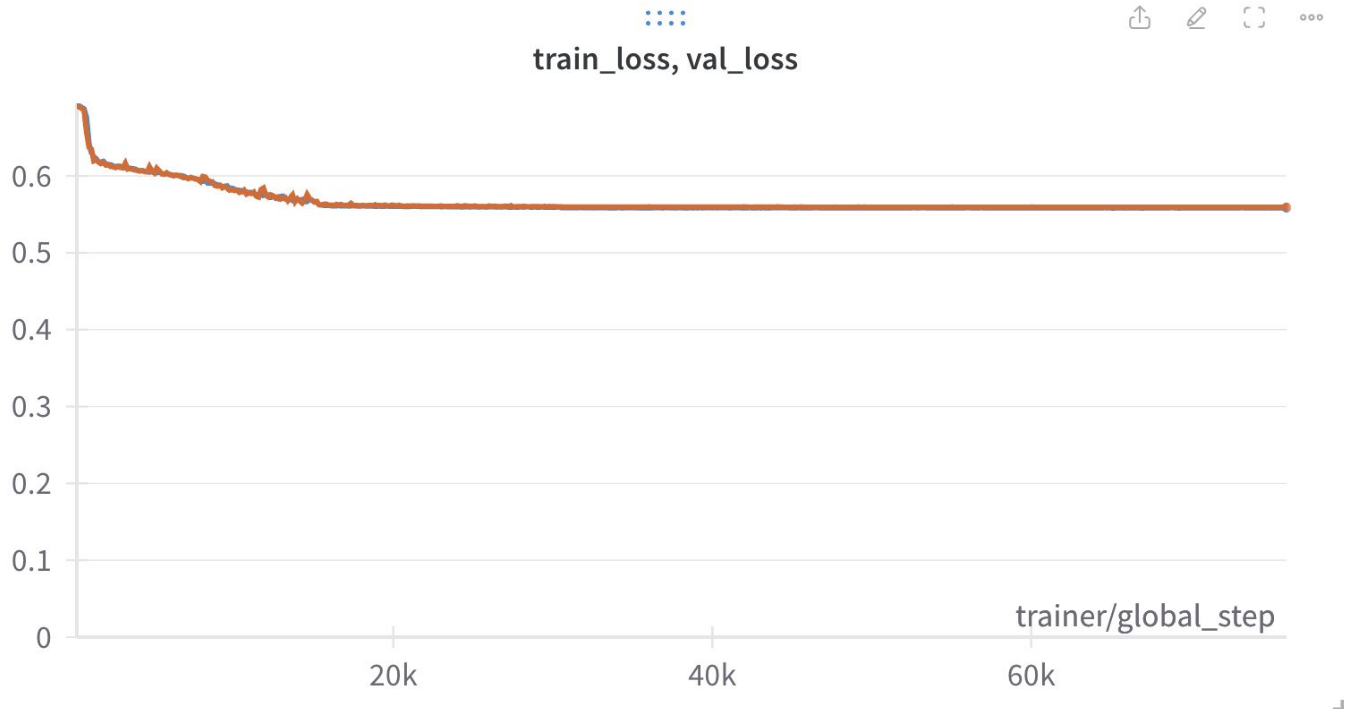
Transformer architecture

Encoder block



Multi-headed attention

CNN loss



train_loss, val_loss
0.6
0.5
0.4
0.3
0.2
0.1
0
trainer/global_step
20k
40k
60k

Transformer loss