A multi-task Large Language Model for Jets

Humberto Reyes-Gonzalez

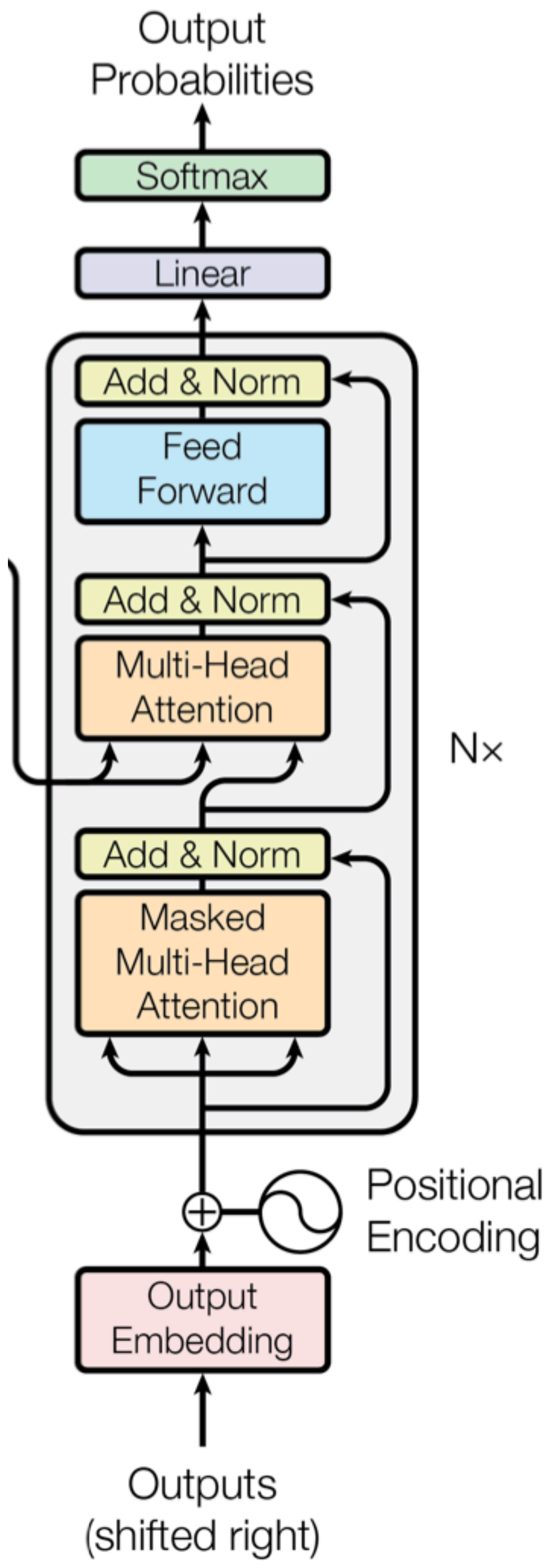
w. Michael Krämer, Alexander Muck.

BOOST 2024
Genova, Italia. 29/7/24

Motivation

- Describing the complex high-dimensional structure of jets is a complicated task.
- Necessary for jet reconstruction, tagging, simulation…
- Many efforts ongoing using Machine Learning.
- Self-supervision Transformers are a promising method .
- They can be used for diverse types of data with varying dimensionality and are easily tuned for specific tasks.
- In particular, they have yielded great advances in the field on Natural Language Processing.
- Here we will attempt to learn the *language* of Jet substructure with autoregressive transformers (based on arXiv:2303.07364) and test their capacity for generation and classification

Transformers



Multi-head attention…

Attention Is All You Need !

https://arxiv.org/pdf/1706.03762

…self-supervision

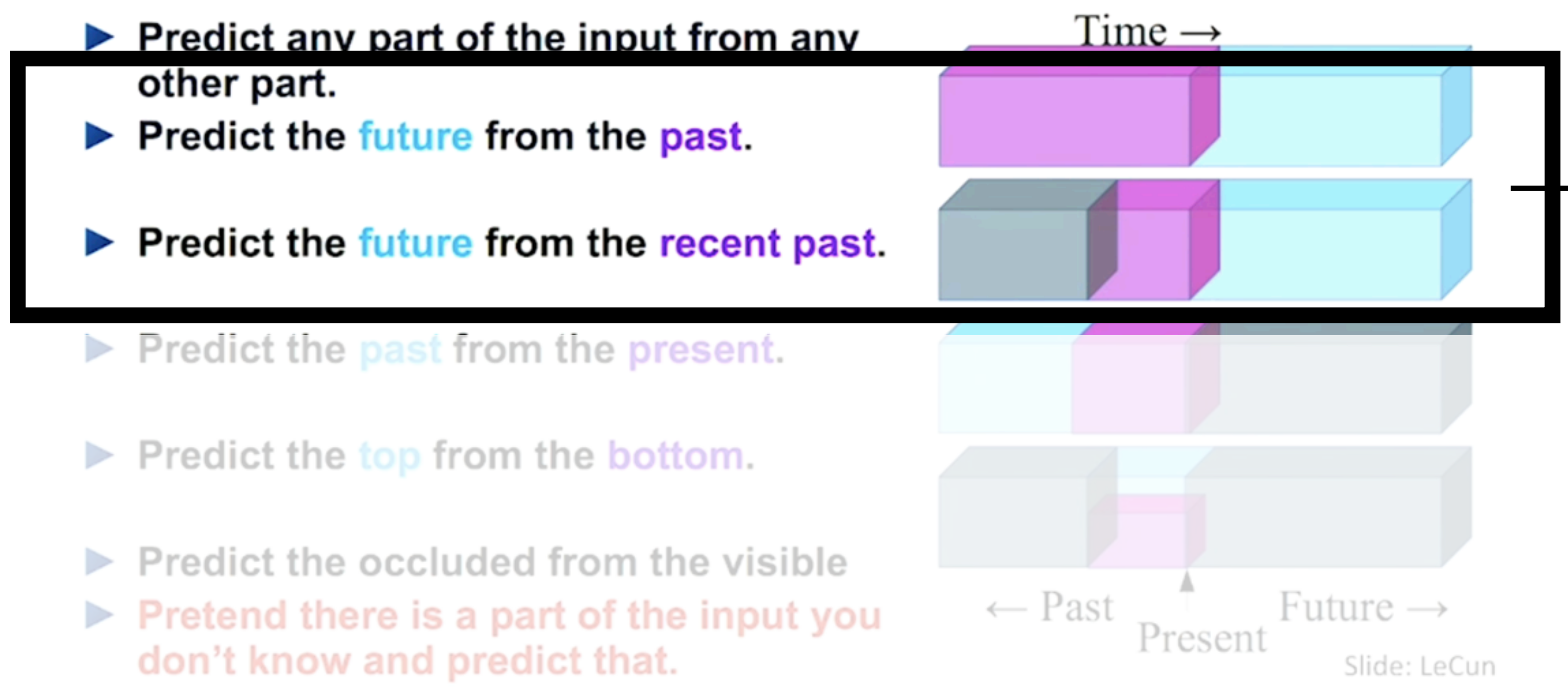
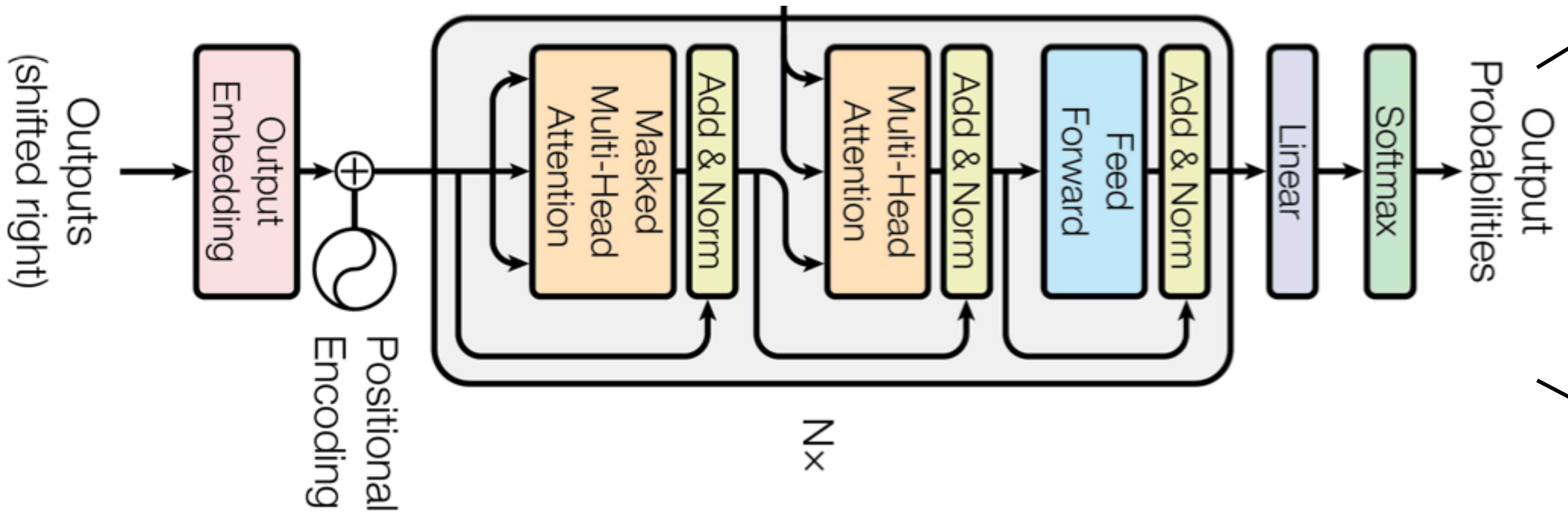


Fig. 1. A great summary of how self-supervised learning tasks can be constructed (Image source: LeCun's talk)

Autoregressive type

$$p(\mathbf{x}) = p(\mathbf{x_1})p(\mathbf{x_2}|\mathbf{x_1})\dots p(\mathbf{x_n}|\mathbf{x_1}\dots\mathbf{x_{n-1}}).$$

Transformers



CLASSIFICATION

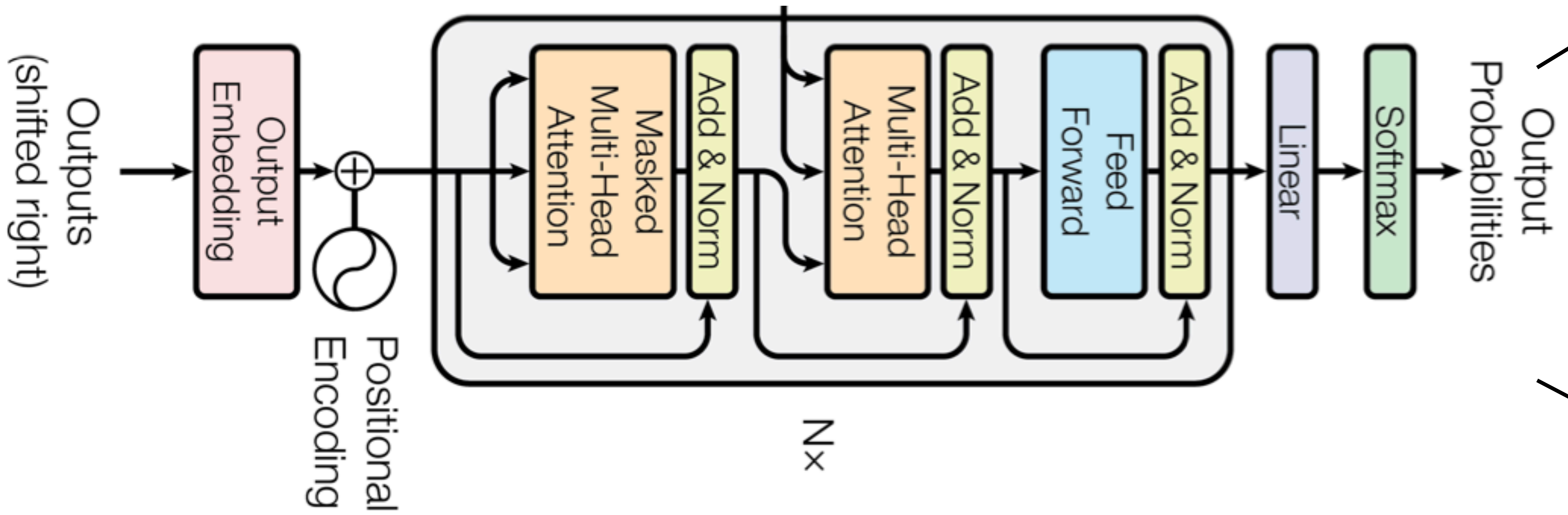
GENERATION

Transformers

p_T

$\Delta\eta$

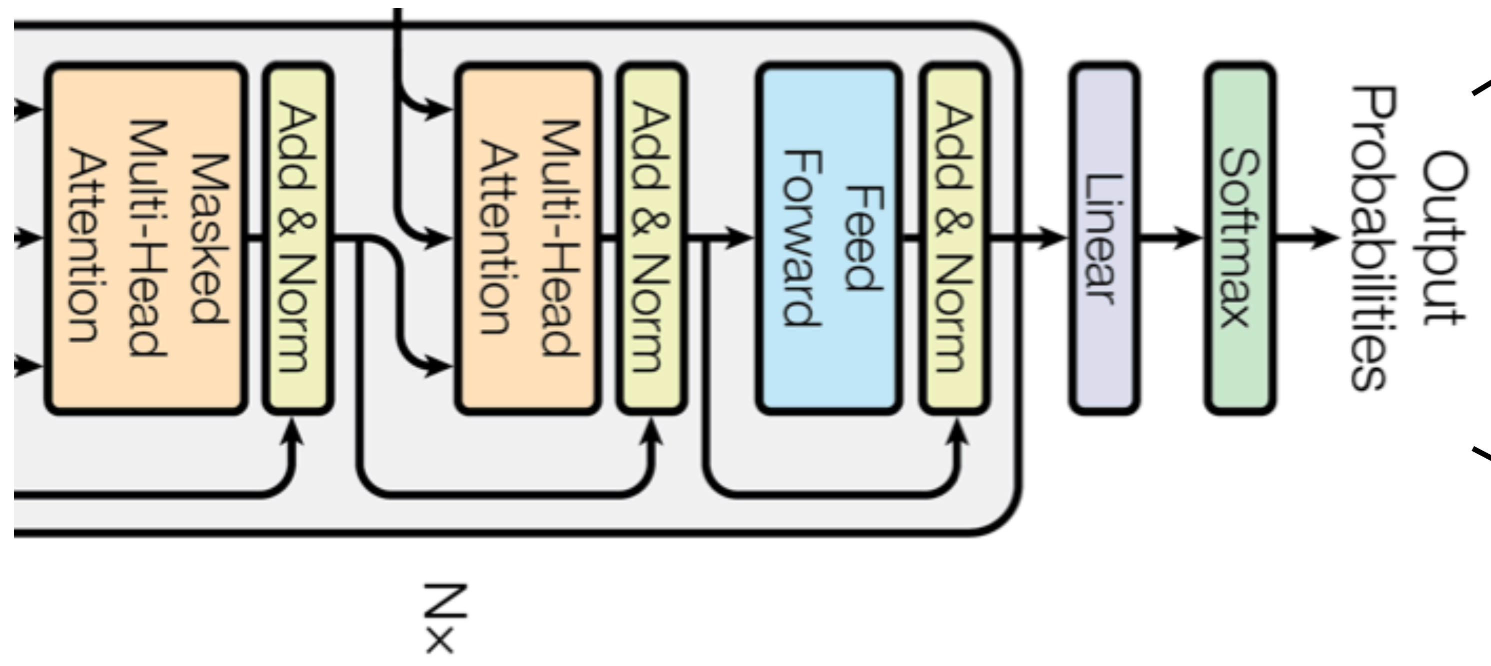
$\Delta\phi$



CLASSIFICATION

GENERATION

Many potential applications for Jets:



CLASSIFICATION

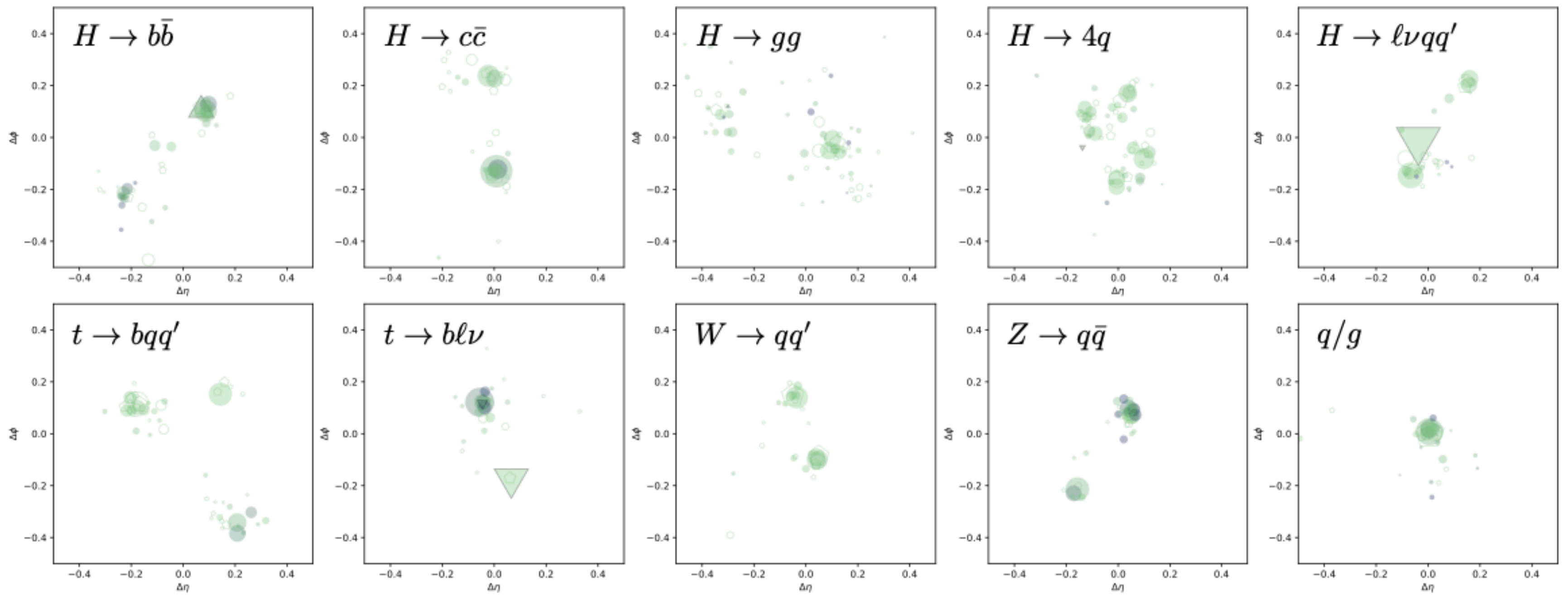
- Jet-tagging
- Re-weighting
- Anomaly detection

GENERATION

- Anomaly detection
- Simulation
- Density estimation

Data

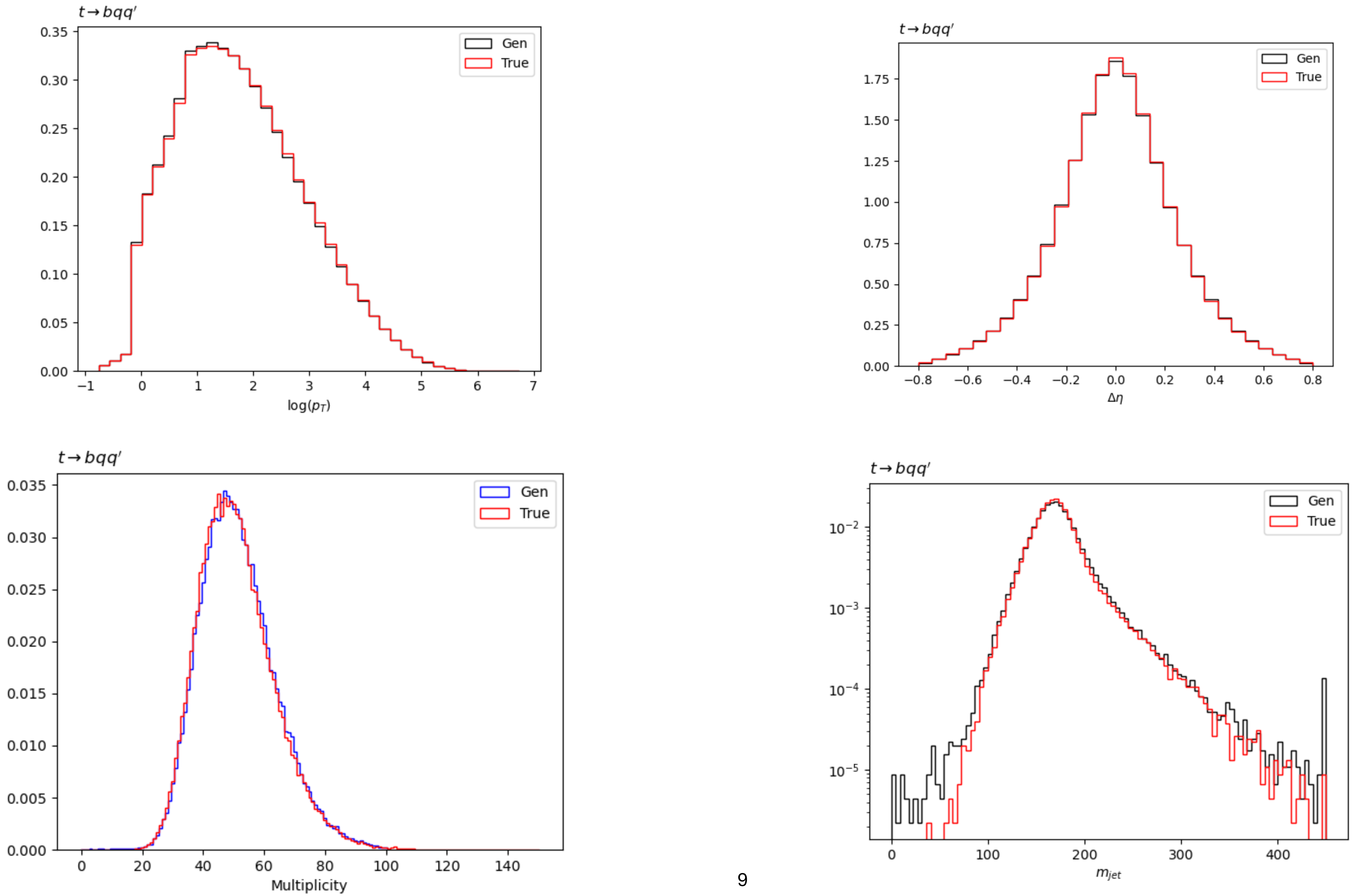
The developers of the PartT jet tagger arXiv:2202.03772 provided The **JetClass Dataset**: 10 million training samples for 10 classes of Jets, plus test and validation data.



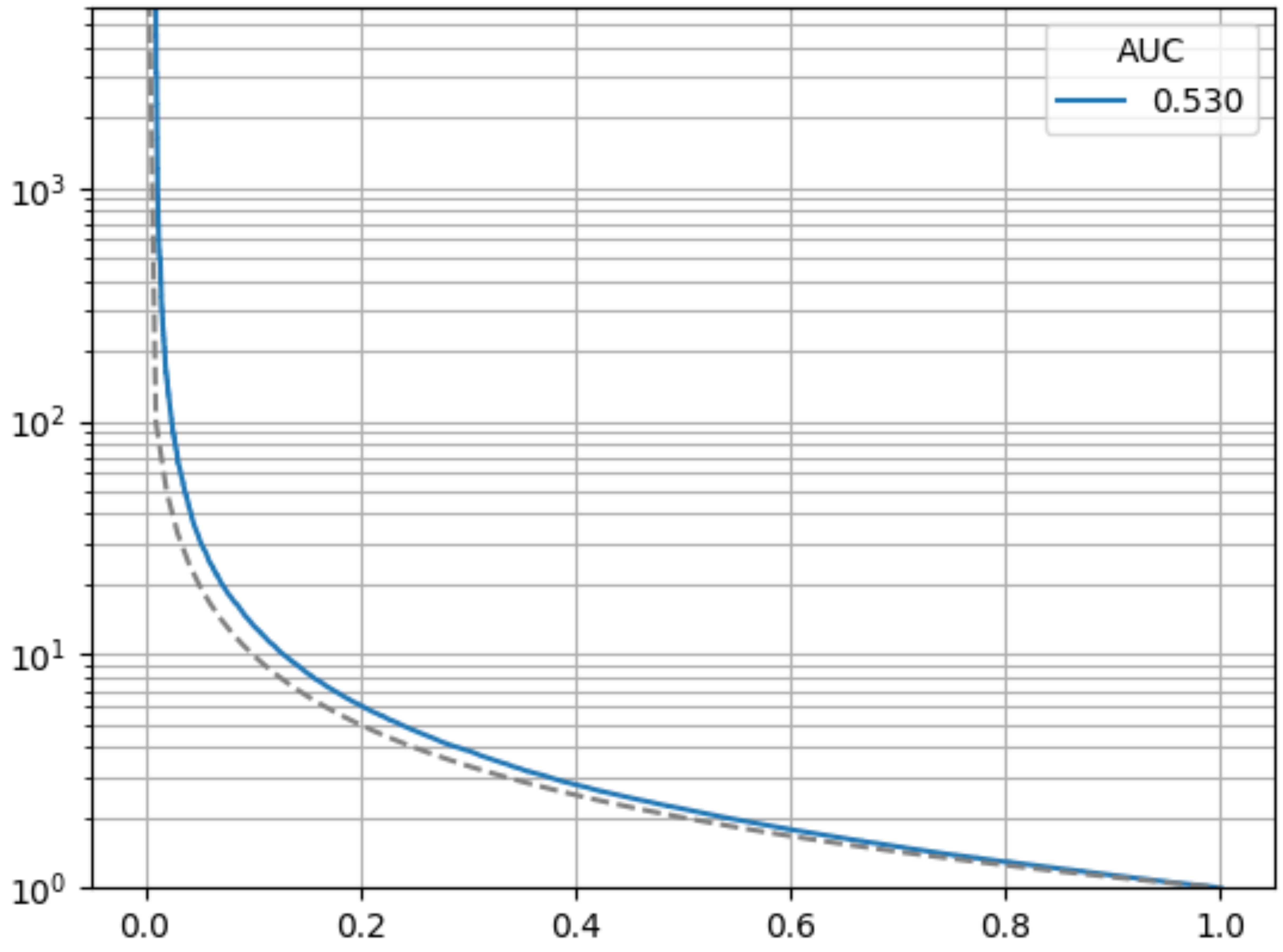
- We represent each jet constituent i with $\Delta\eta_i = \eta_i - \eta_{jet}$, $\Delta\phi_i = \phi_i - \phi_{jet}$ and p_{T_i}
- Then, we $\log p_{T_i}$.
- The features are **discretized (tokenized)** as integers: $p_T \in [0,40]$, $\Delta\eta \in [0,30]$ $\Delta\eta \in [0,30]$

Results (Always same architecture)

Generation: $t \to bqq'$

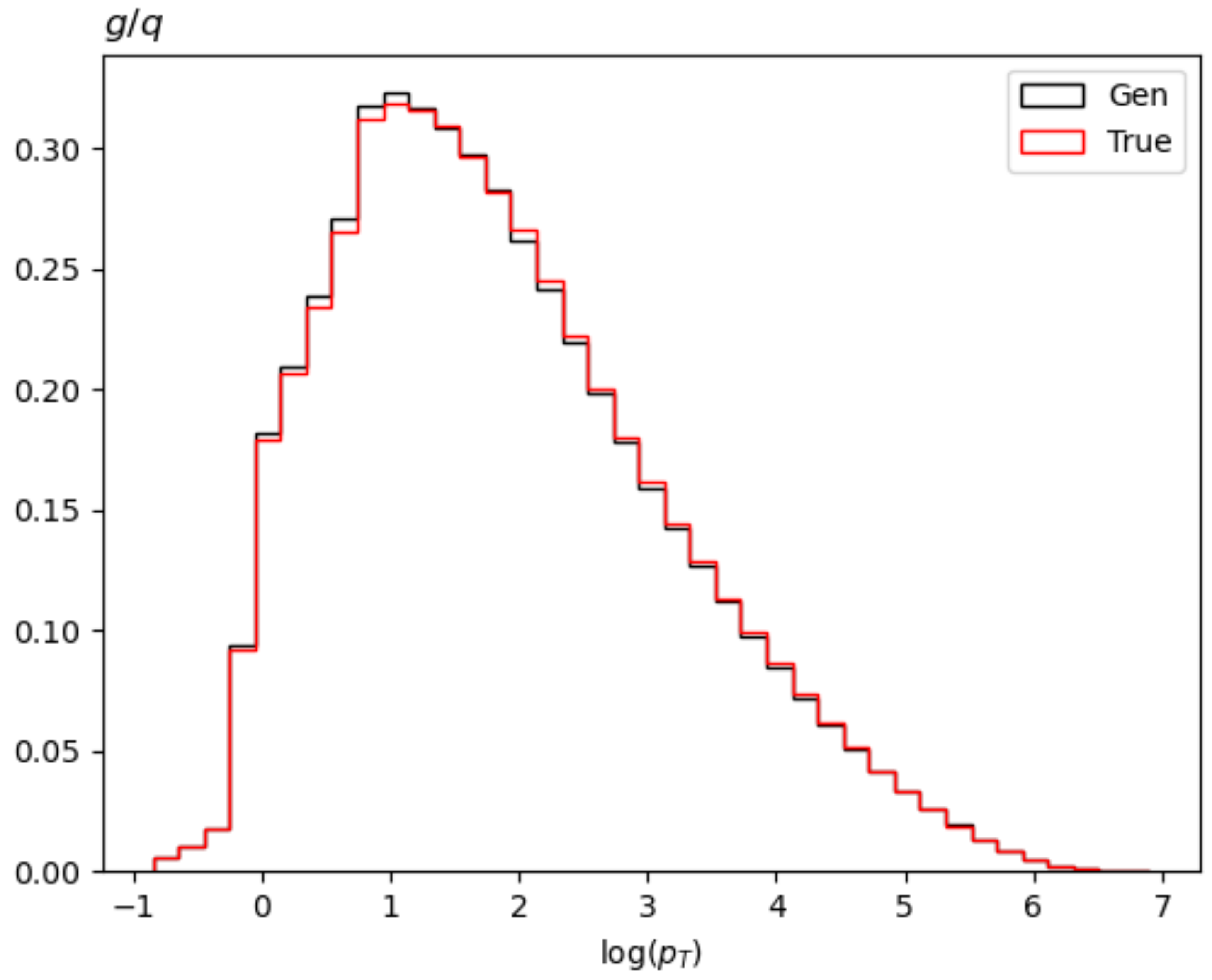


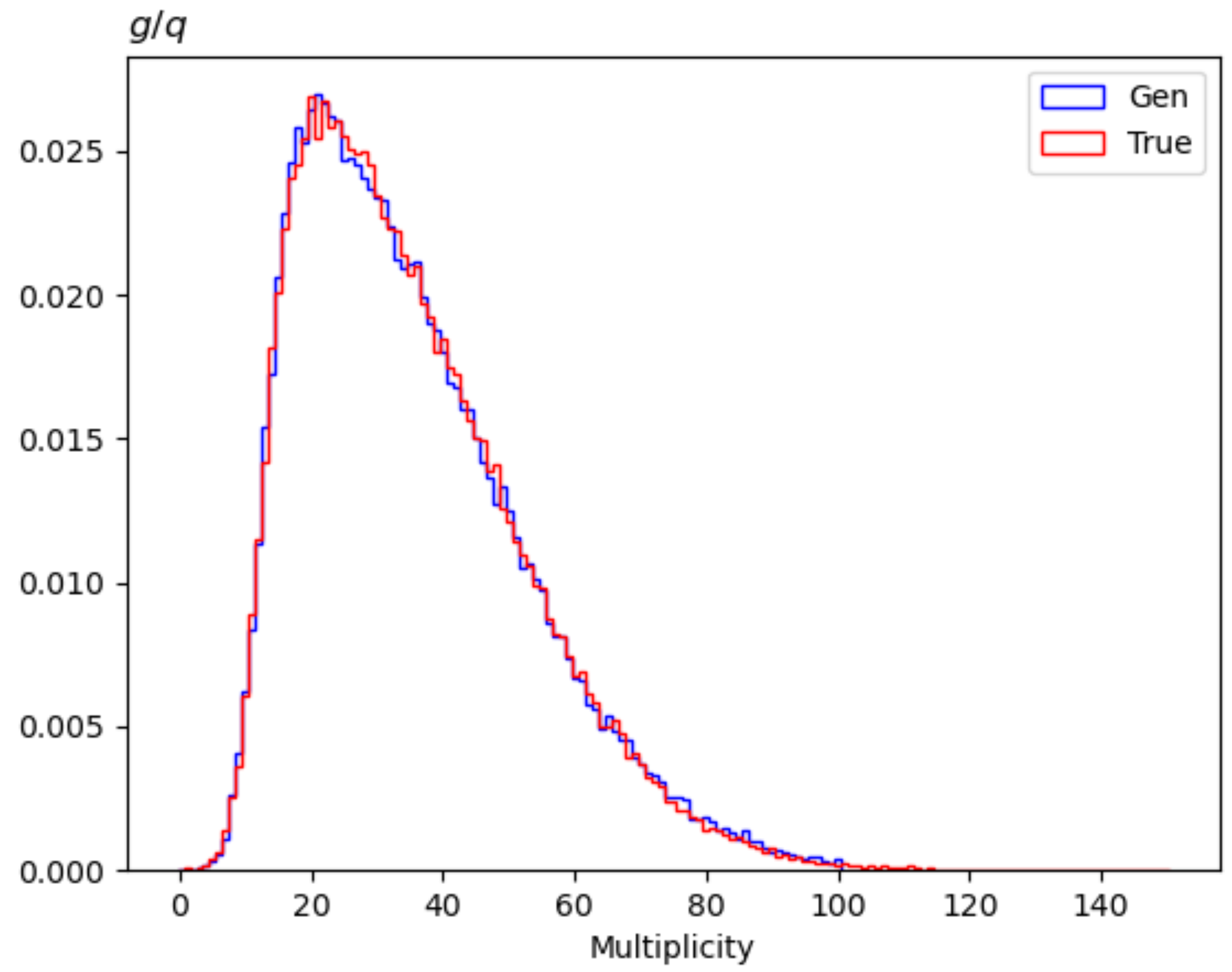
Generation: $t \to bqq'$

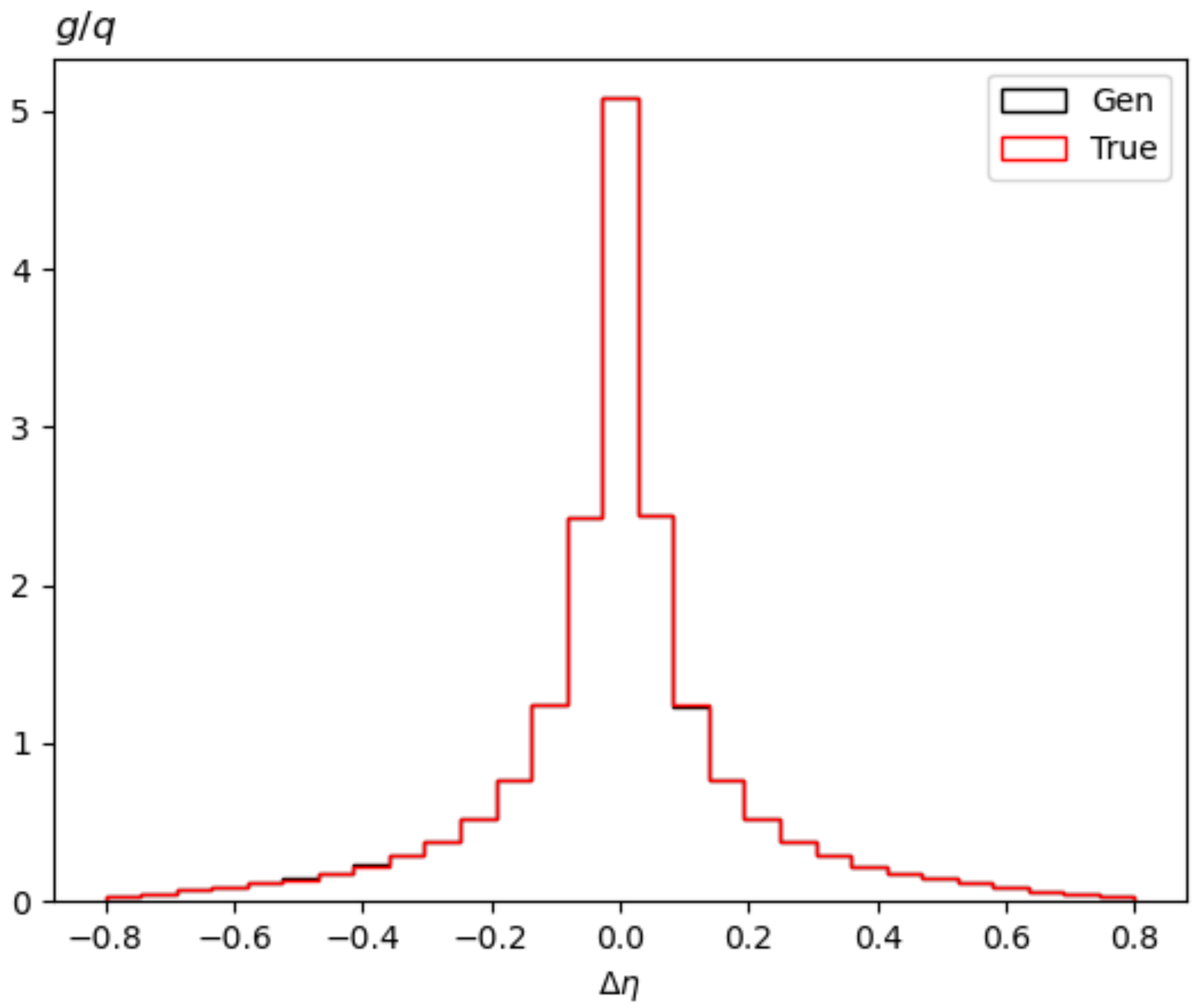


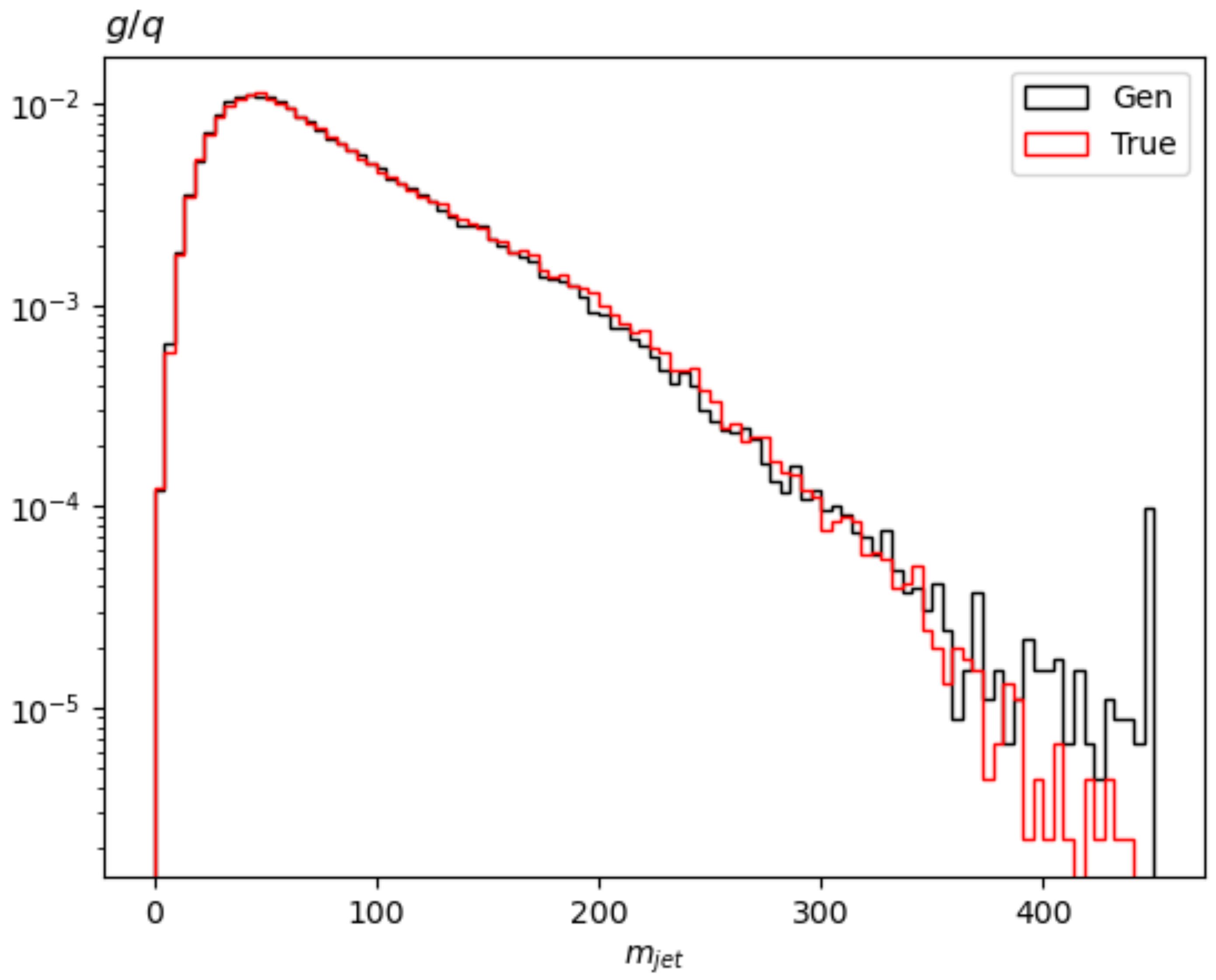
True vs generated classification. Trained with ParticleNet. arXiv:1902.08570

Generation: g/q

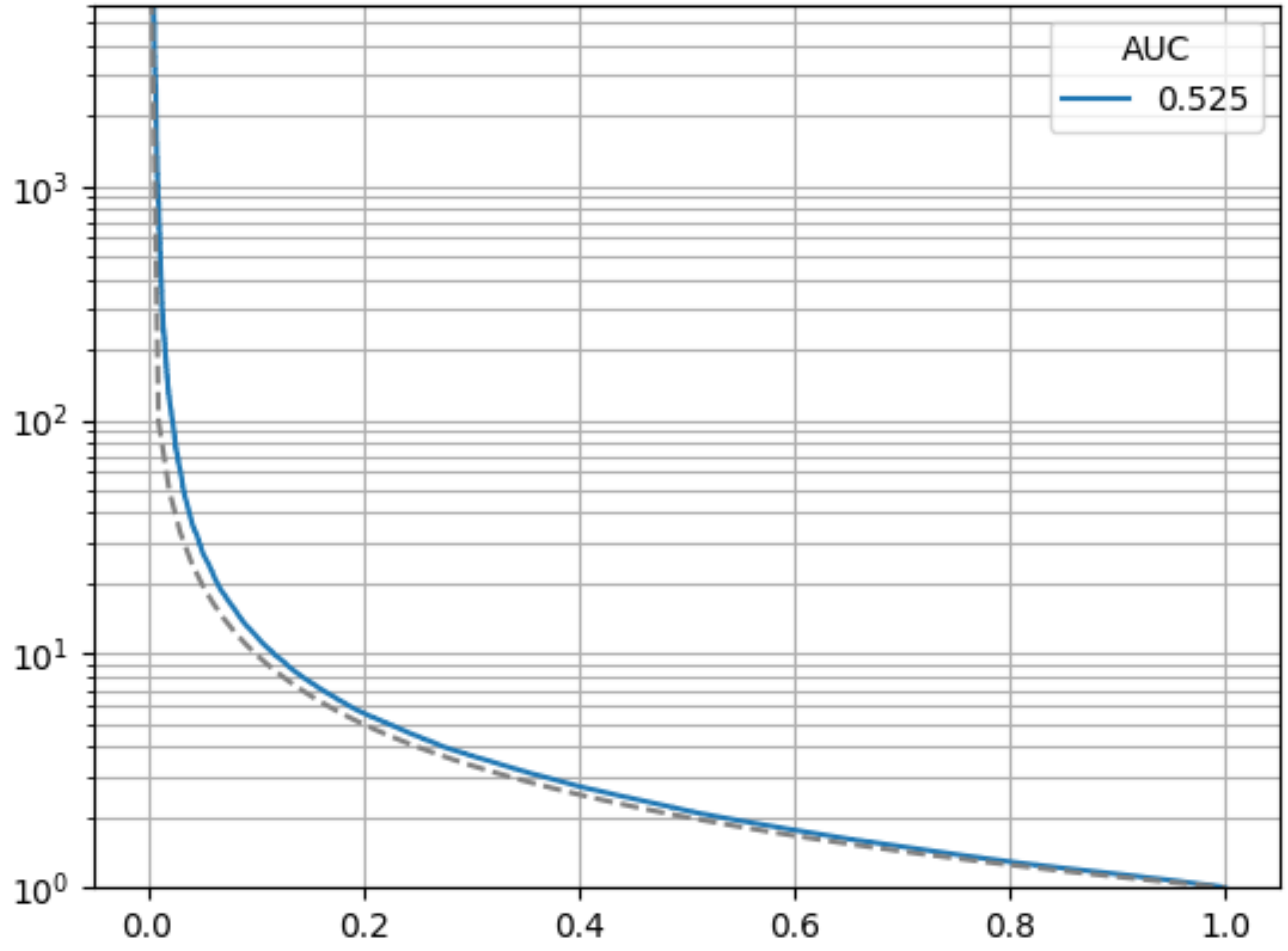








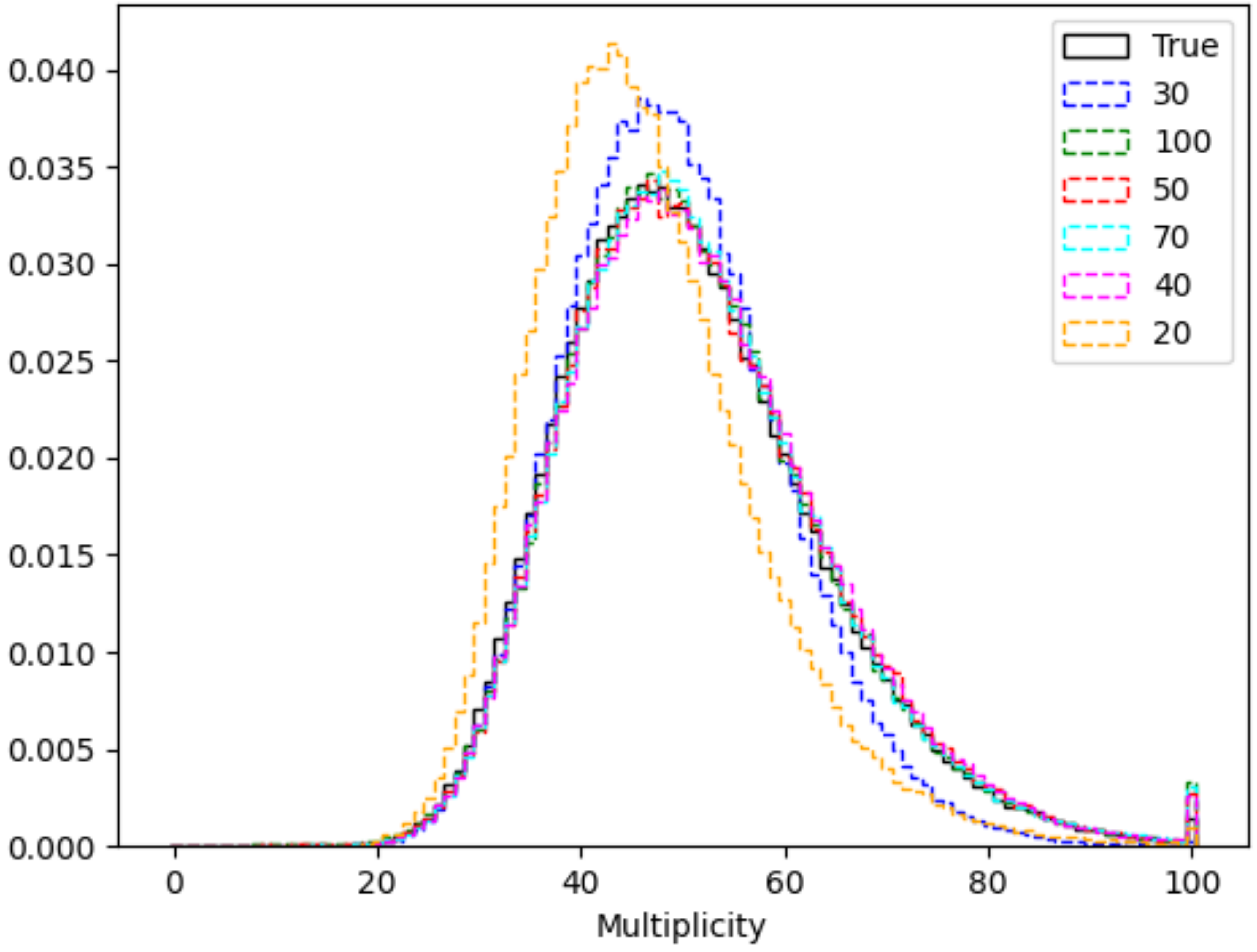
Generation: g/q



True vs generated classification. Trained with ParticleNet. arXiv:1902.08570

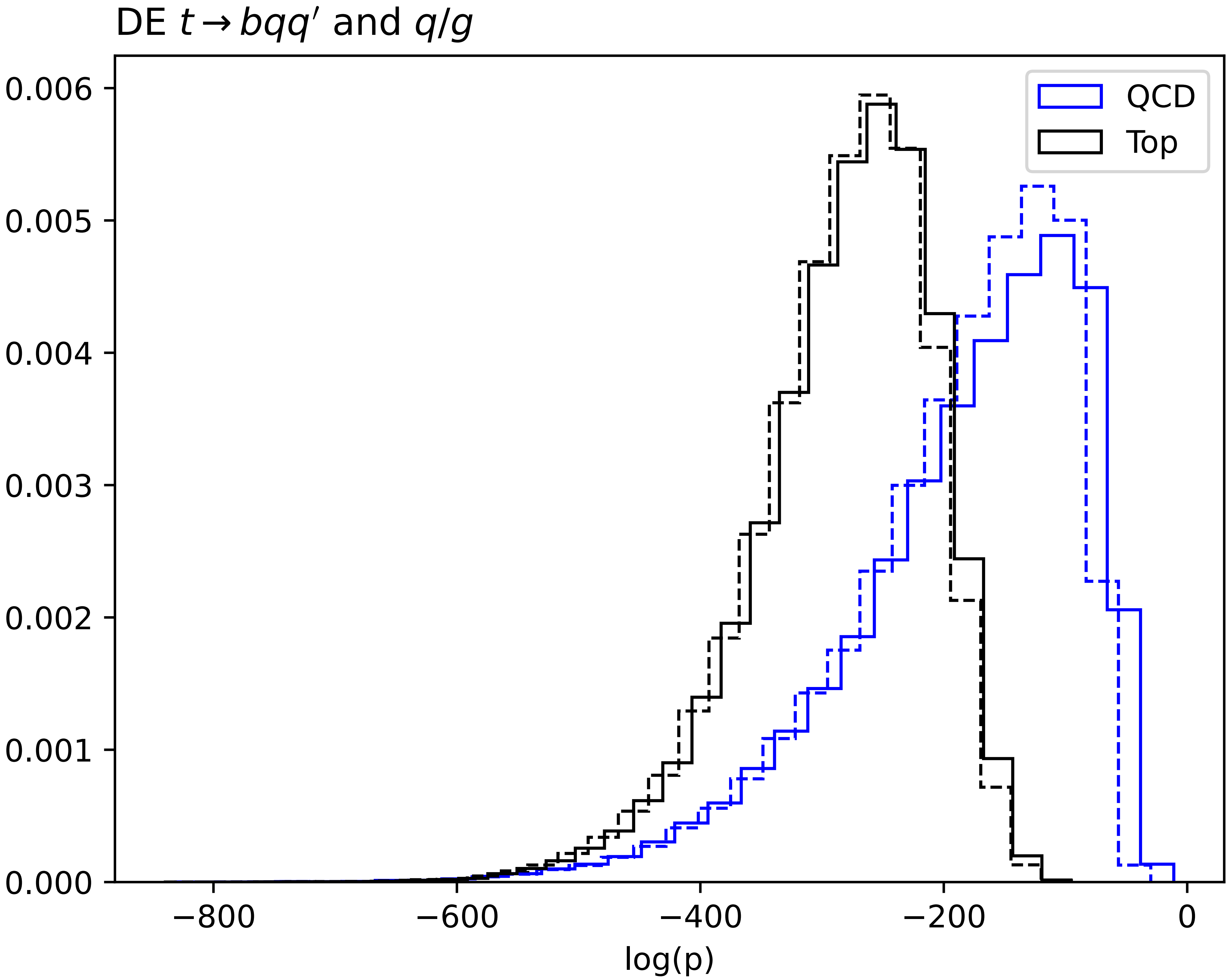
A couple of cool properties…

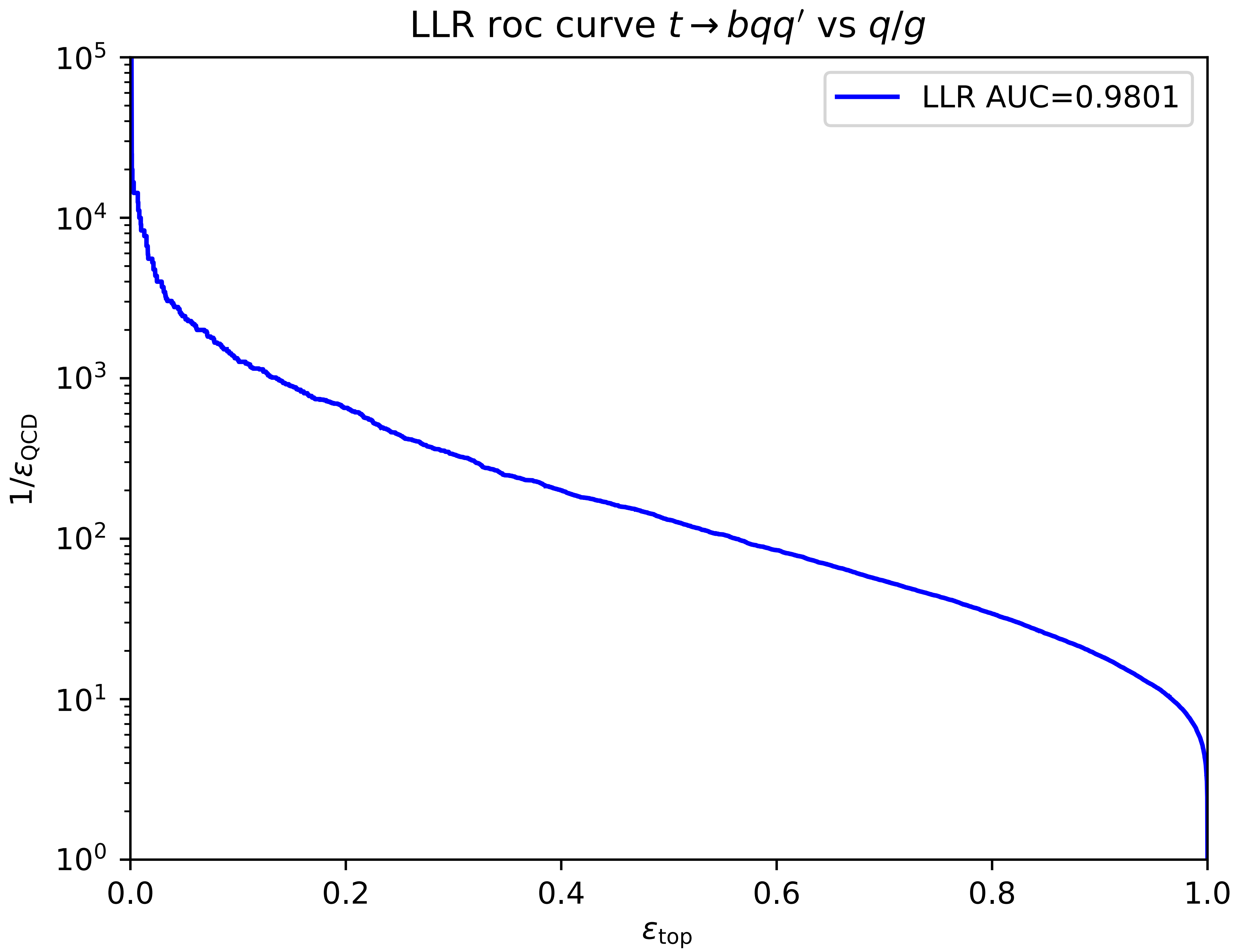
Multiplicity extrapolation



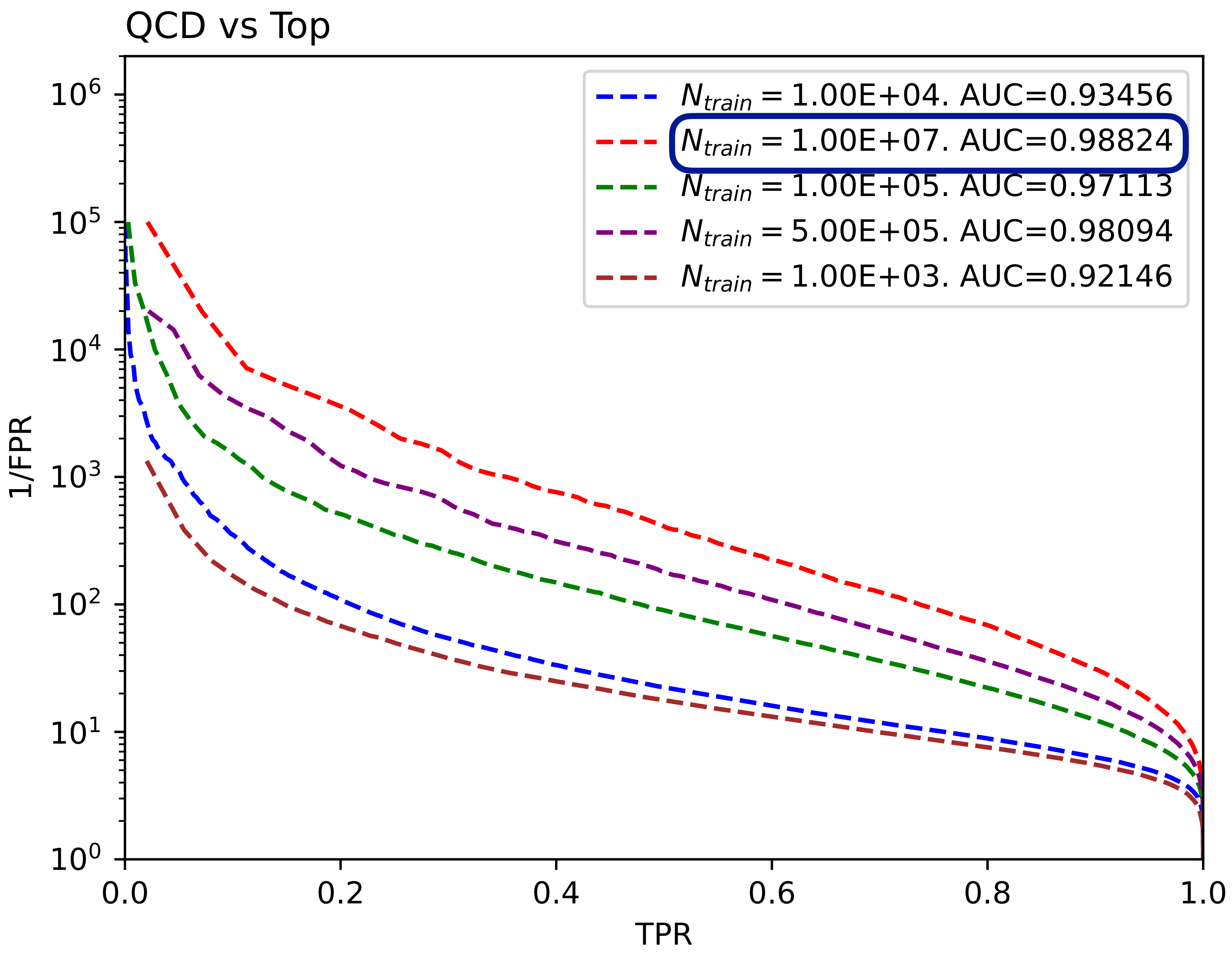
A couple of cool properties…

Density estimation

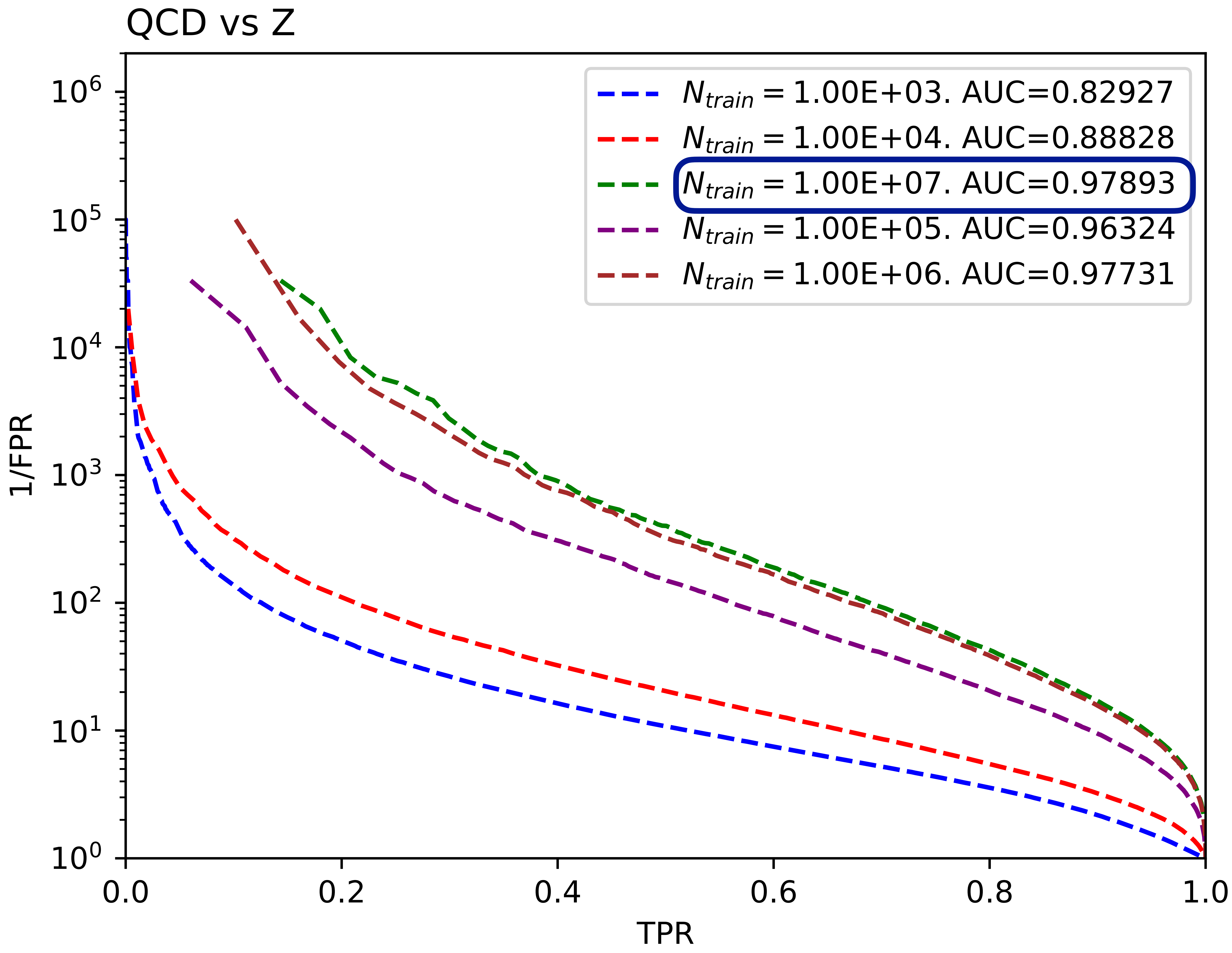




Classification

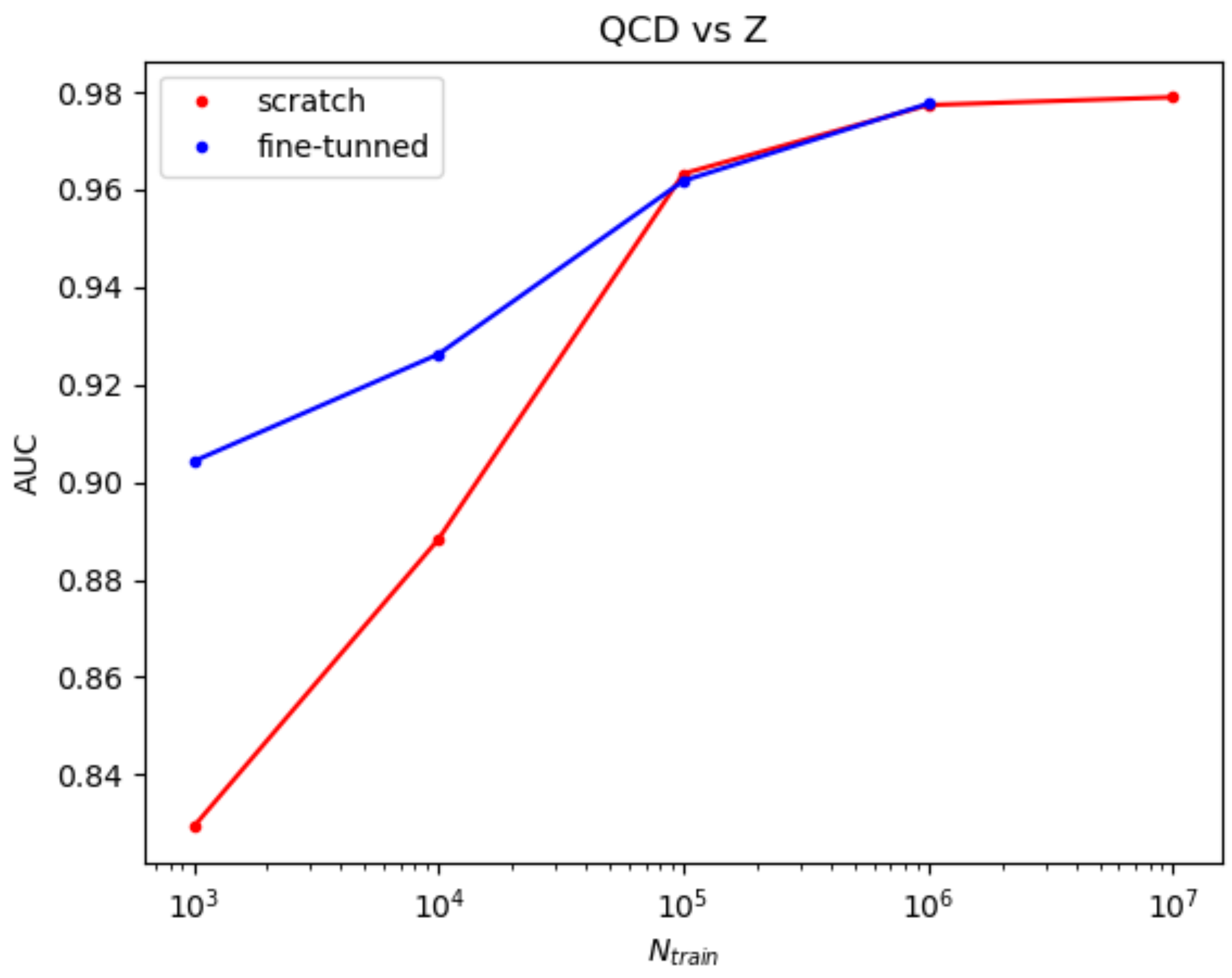


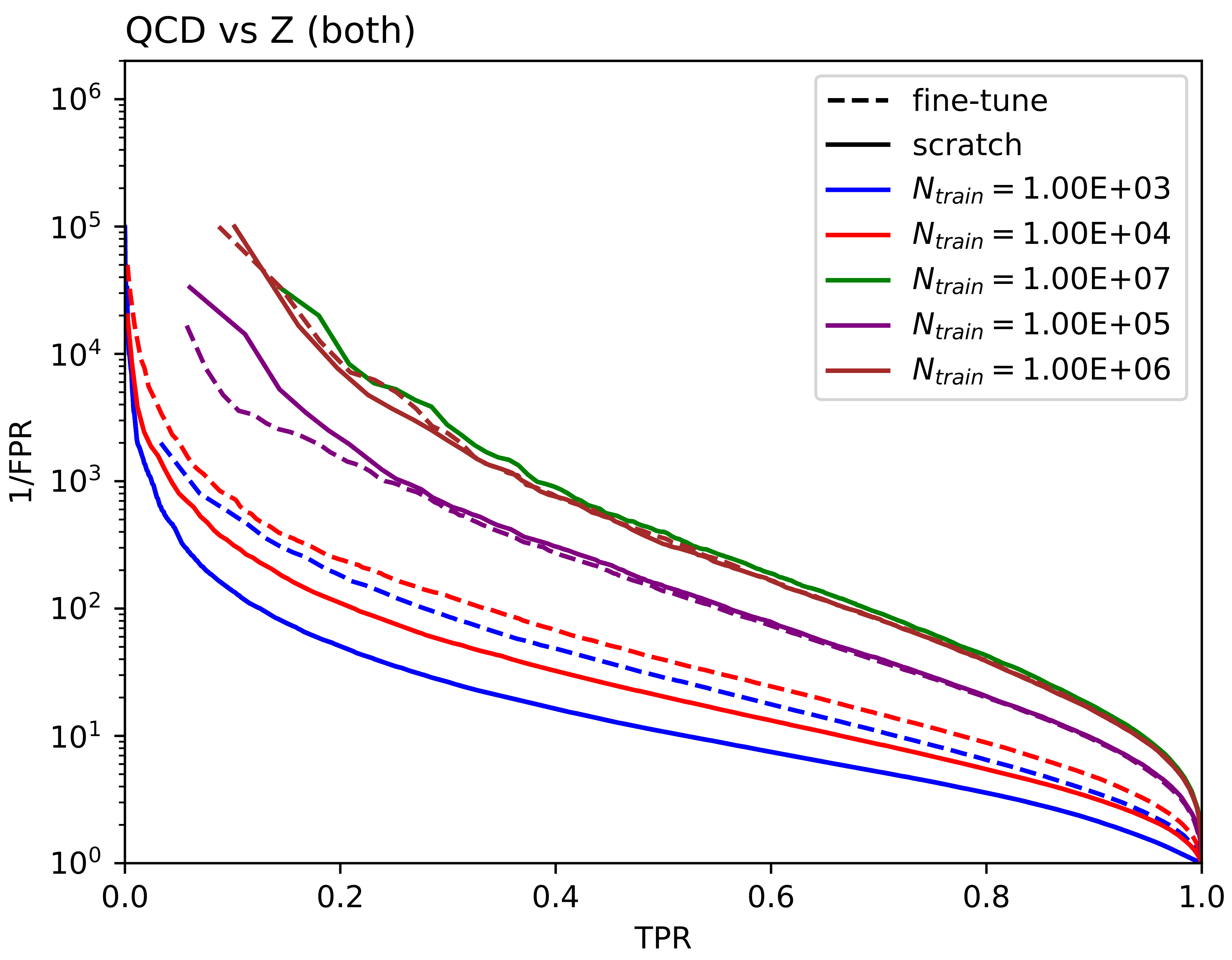
Classification



Transfer learning

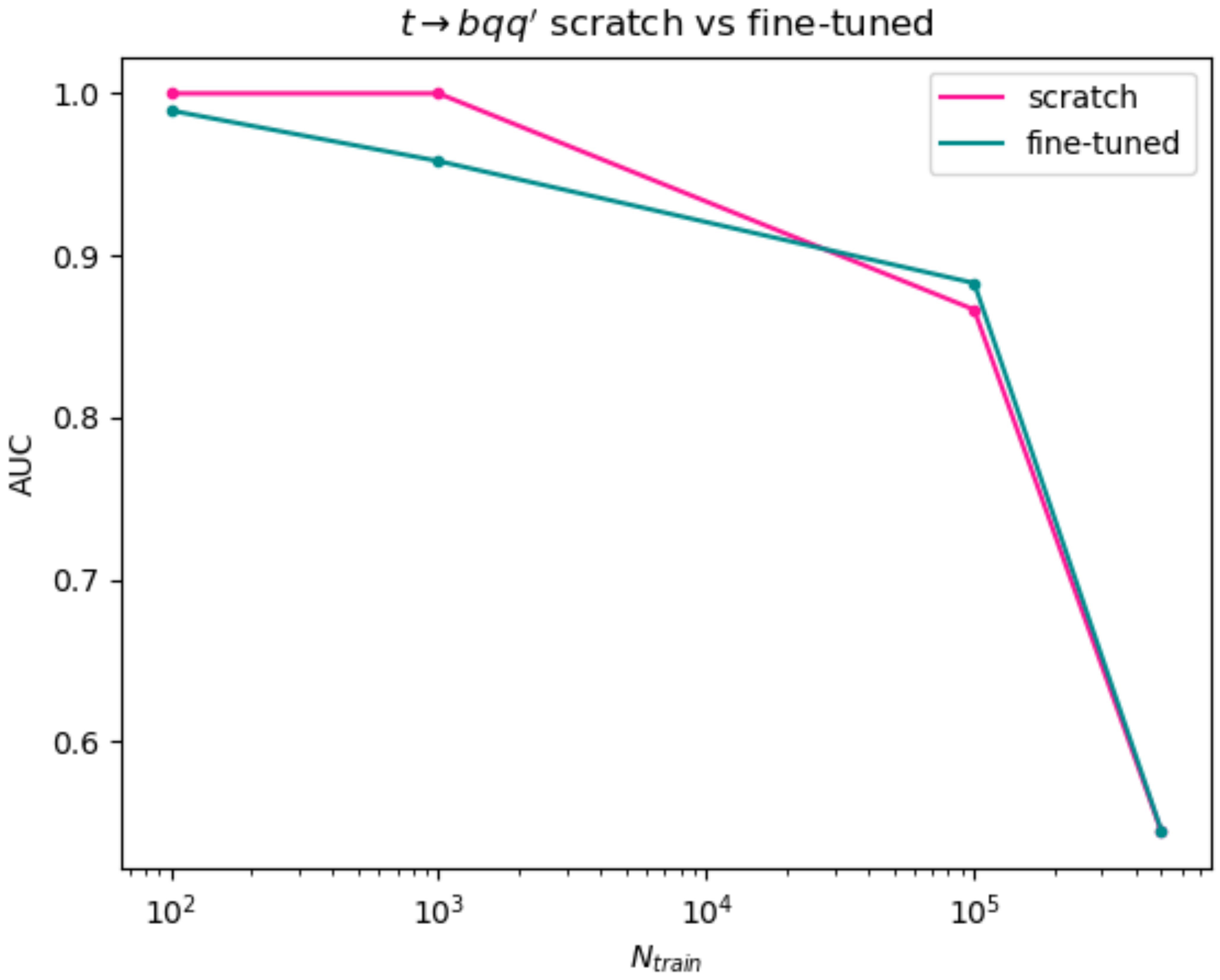
Classification
QCD vs Top -> QCD vs Z

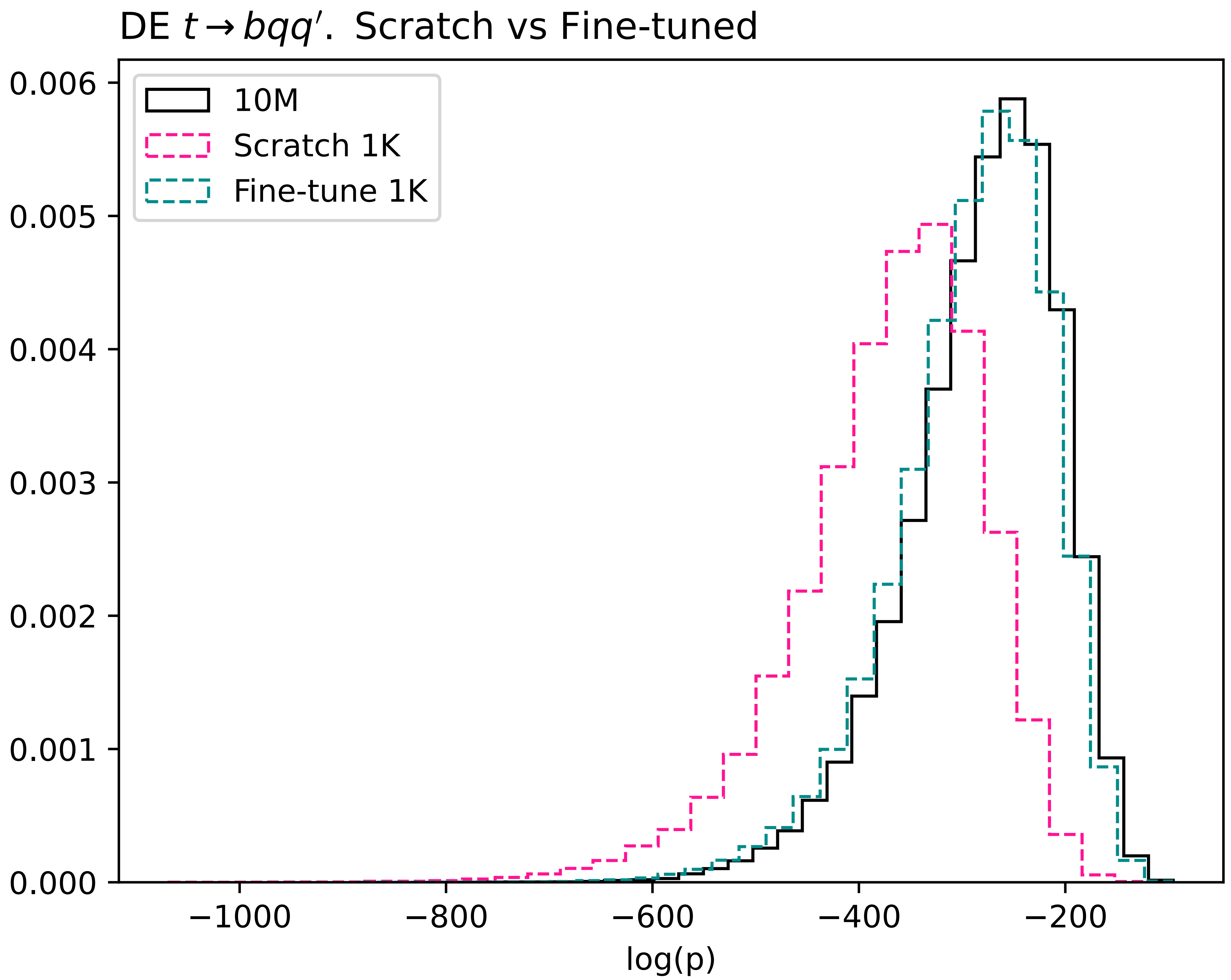




Transfer learning

Generation: QCD -> Top





Conclusions

- We find that multi-head attention **transformers are able to accurately build representations of jet substructure**.
- They can be easily turned for generation and classification.
- Which would lead to numerous applications: Tagging, simulation. Anomaly detection, etc.
- Quite some activity in HEP (see also talks this week): arXiv:2303.07364, arXiv:2401.13537, arXiv:2403.05618, arXiv:2404.16091, arXiv:2202.03772, arXiv:1804.09720, etc …

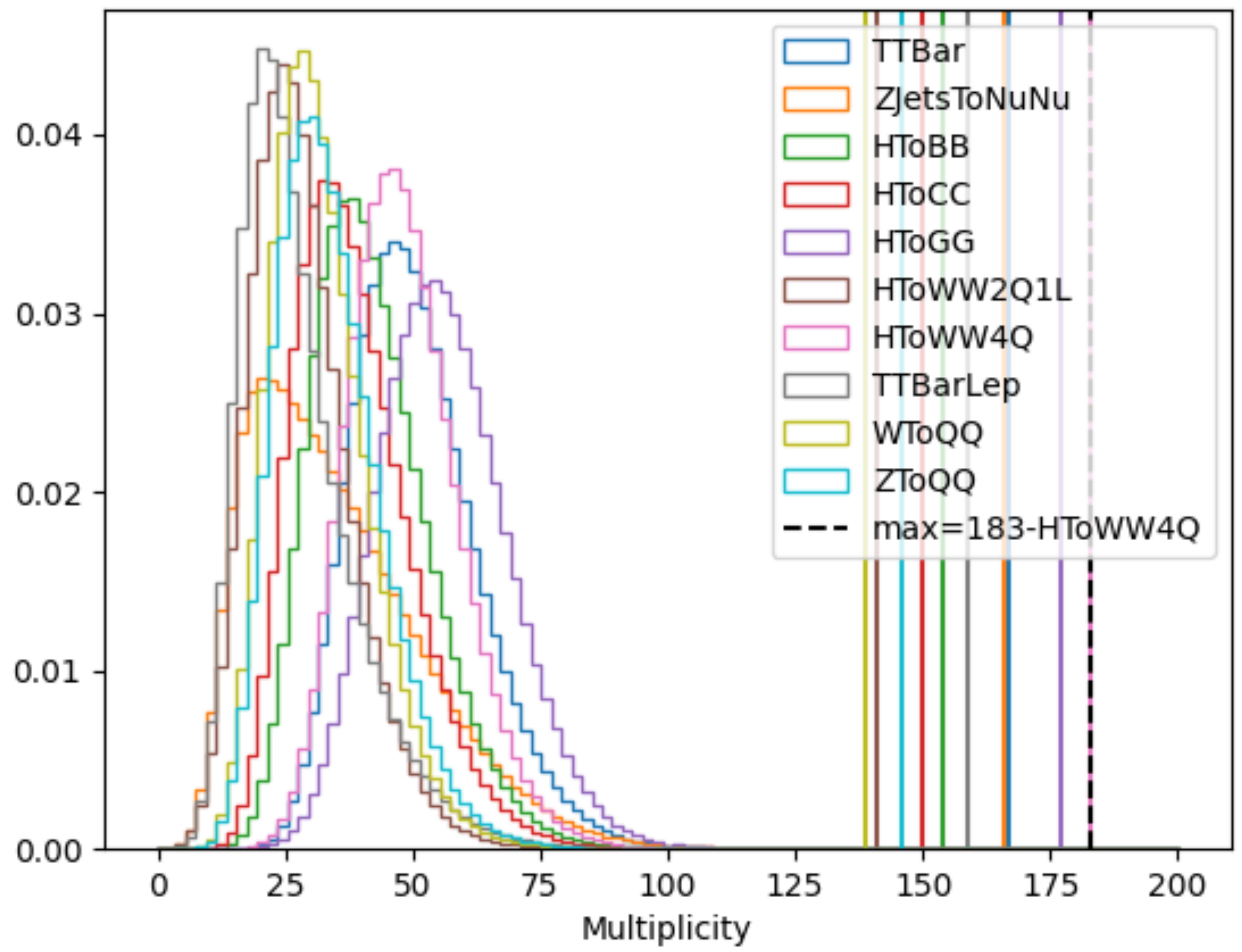
- **A lot of work to do,** for instance:

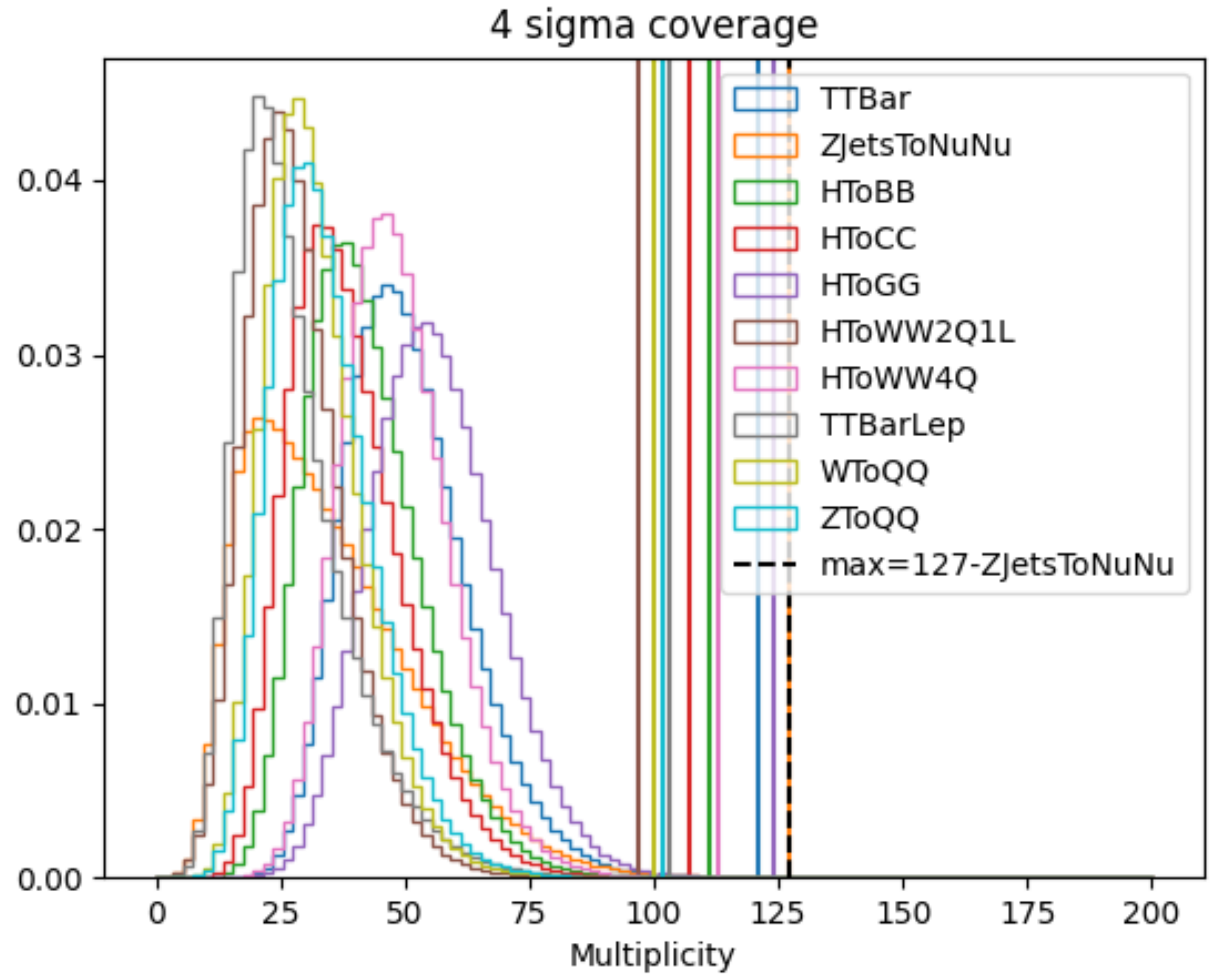
  - Find efficient tokenisation strategies (fine-grained bins, VQ-VAE, VQ-GAN, XVal,…)
  - Try more sophisticated *heads* for downstream tasks.
  - Train *backbones* with large and preferably diverse data.
  - How to train *backbone* for more efficente/accurate generalization?
  -How to test the accuracy and estimate the uncertainty of high dimensional generators?

Thank you!

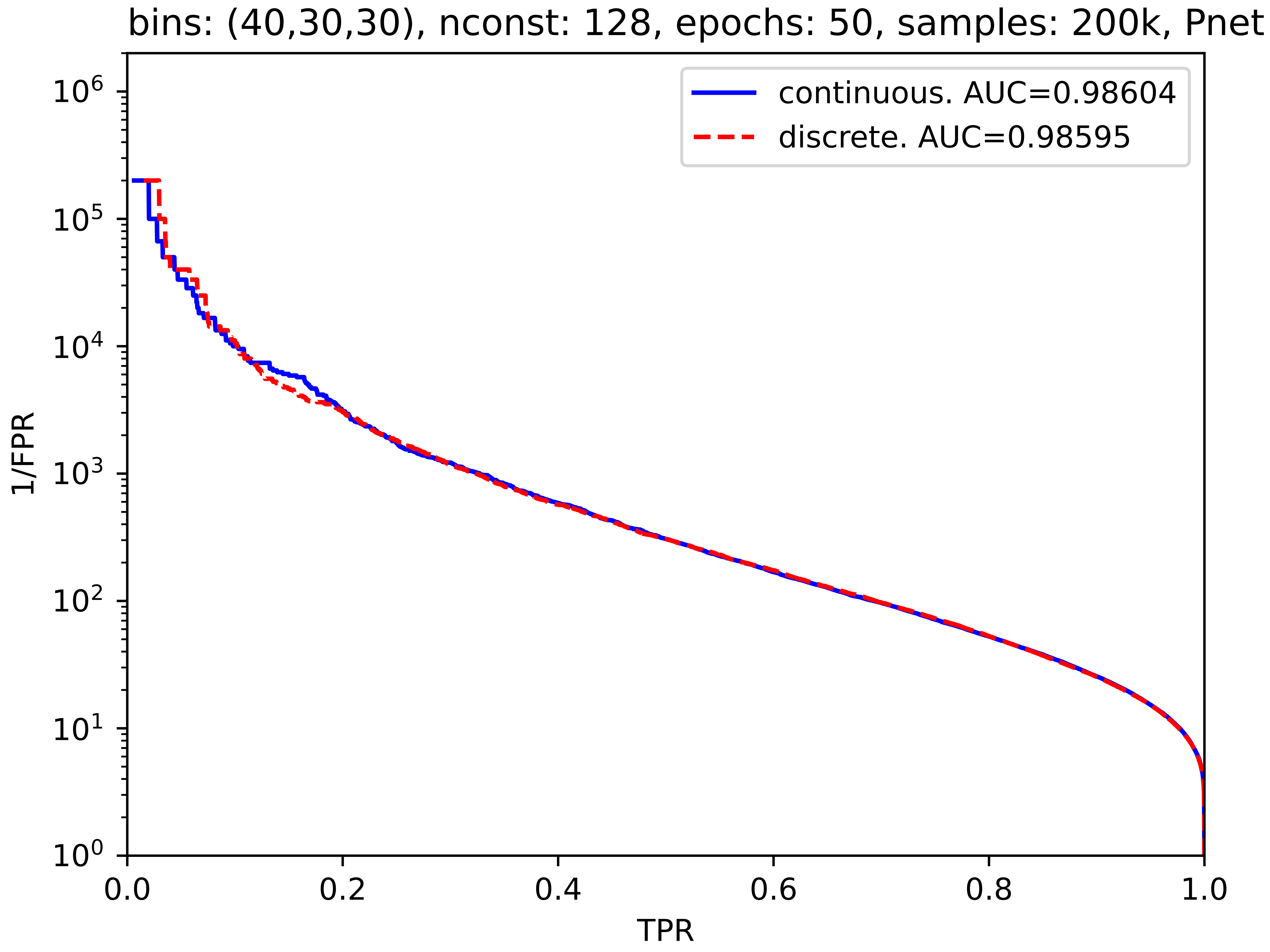
Back-up

Multiplicity



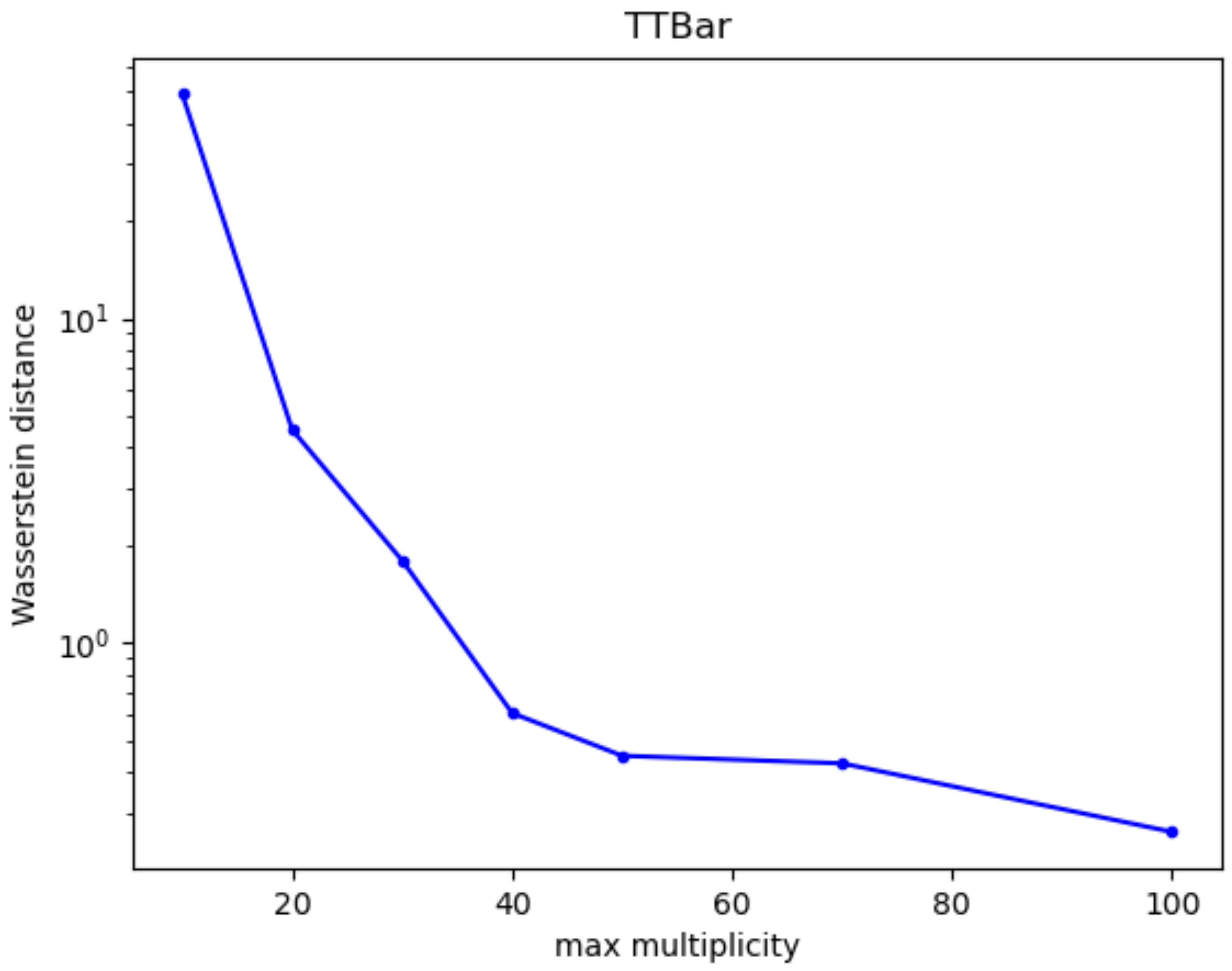


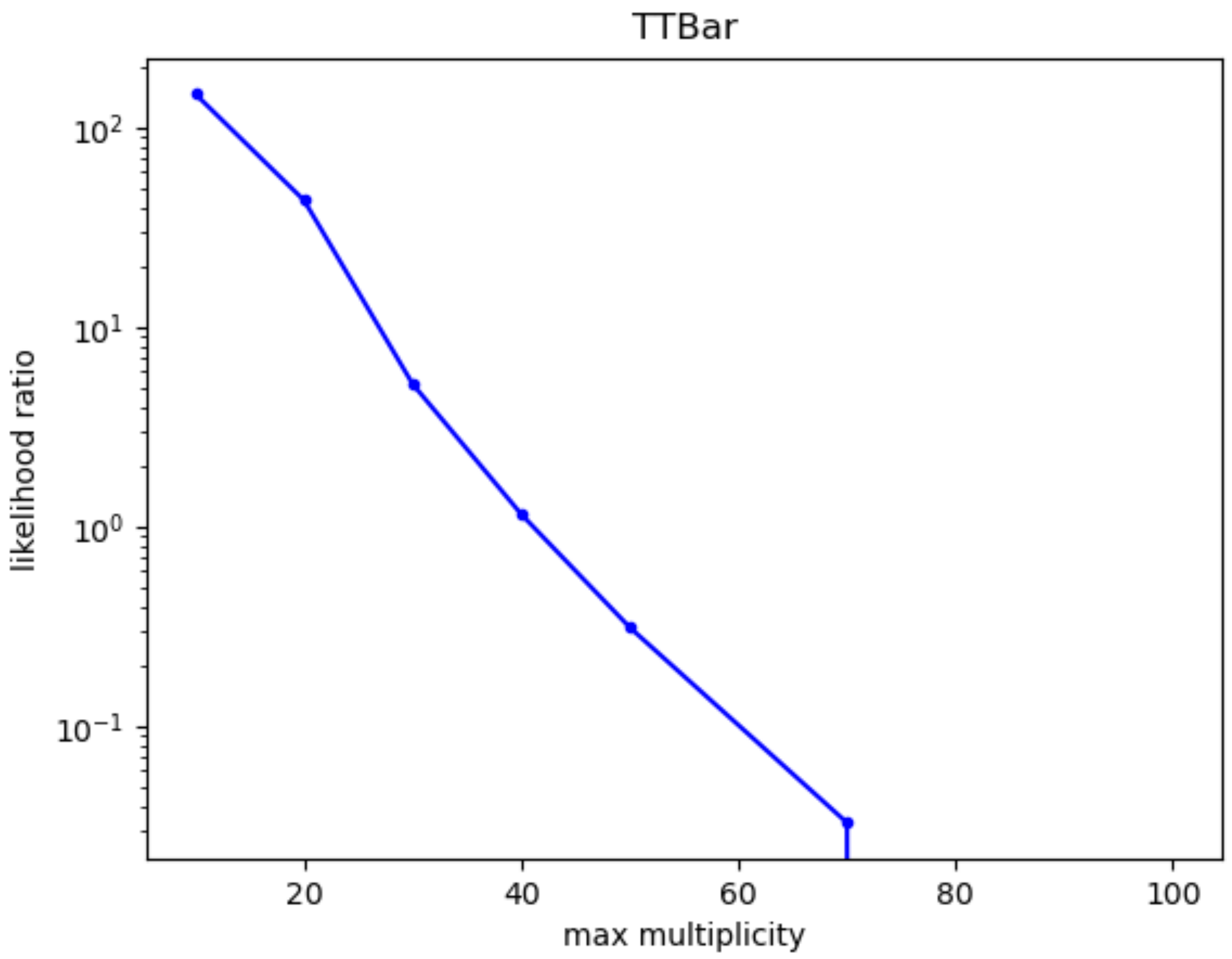
Information loss from discretisation?



Multiplicity dependence

$$\frac{\sum |\ (\log(p_{mult}) - \log(p_{100})\ |}{N}$$





num_const	w_distance_mul	LLR
10	49.116268	147.74804
20	4.50374210	43.339125
30	1.748110526	5.2046128125
40	0.626410526	1.15853242
50	0.49659473	0.3134186132
70	0.4536684	0.03294159
100	0.286457894	0.0

Best hyperparameters

No dropout, lr
001 is better

Table 1

name_sufix	dropout	lr	hidden_dim	num_layers	num_heads	w_distance_pt	w_distance_mj	w_distance_mul
7UJGYPA	0.0	.001	128	4	4	0.0026115	0.00128009	0.423864
TF28V0K	0.0	.001	256	8	4	0.0030855	0.0011687	0.4180666
GBOGYSH	0.0	.001	128	8	4	0.00311702	0.0011228	0.629232
43V6VKM	0.0	.001	128	4	2	0.0032688	0.00134409	0.579718
HNBPXHW	0.0	.001	128	8	2	0.0035342	0.00128216	0.597039
JS8IICS	0.0	0.0005	256	8	4	0.00365753	0.001216956	0.794875
RNNLY3D	0.1	0.0005	256	8	4	0.020951378	0.0033593	1.023535
1QPCWAZ	0.1	.001	256	8	4	0.03392		2.980099