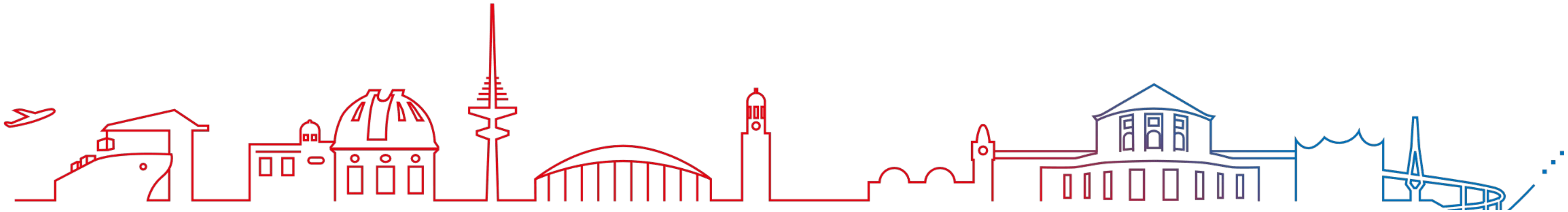
Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CLUSTER OF EXCELLENCE
QUANTUM UNIVERSE

PUNCH 4NFDI



Beyond Language: Foundation Models for Collider Physics Data

OMNIJET-α: The first cross-task foundation model for particle physics (2403.05618)

Anna Hallin, Joschka Birk, Gregor Kasieczka
anna.hallin@uni-hamburg.de

ACAT 2024

Why are foundation models interesting?

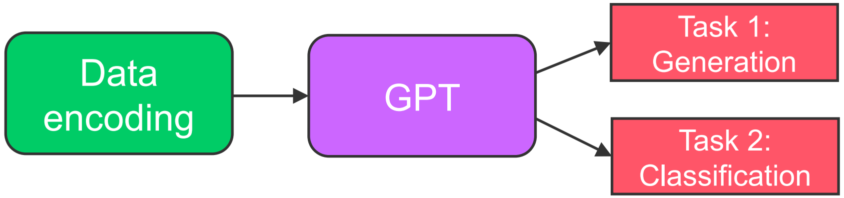
- Foundation models **pre-train** on a certain (large) dataset for a certain task, **fine-tune** to perform on a different dataset or a different task
- Promising avenue for particle physics:
  - use **pre-trained larger models** (trained on data) to fine-tune for specific tasks, instead of training every task from scratch
  - Saves compute and human **resources**
  - Pre-trained models need **less data**
  - Potential of **sharing** models and architectures within an experiment, across collaborations, and with the theory community

Towards foundation models in particle physics

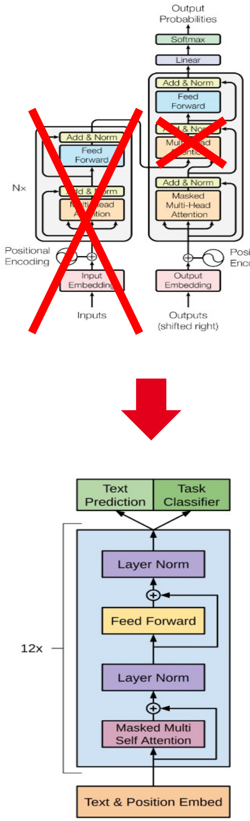
- Two examples:
  - **ParT** (2202.03772) learns classification on one dataset and can be finetuned on another (different) dataset
  - **MPM** (2401.13537) trains on a surrogate task to improve the performance of a classifier
  - In both cases, the pre-training results in better performance of the downstream task than training that task from scratch
- However, until now, no model has been able to **task-switch** between **full jet generation** and **classification**
- **OmniJet-α** is a foundation model for jets, built on **generative pretraining** and able to task-switch to classification

Generative pre-training

- Idea: while learning to generate, a model also learns aspects of the data useful for other tasks
- The transformer architecture is commonly used in natural language processing for generative pre-training
- We choose the original GPT-1 architecture [1], which is based on the decoder part of the transformer



[1] Radford *et al*, "Improving language understanding by generative pre-training," (2018)

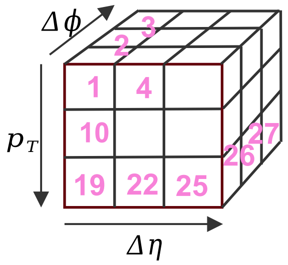


1706.03762

[1]

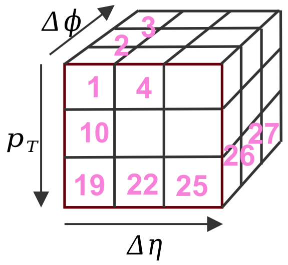
Tokenization

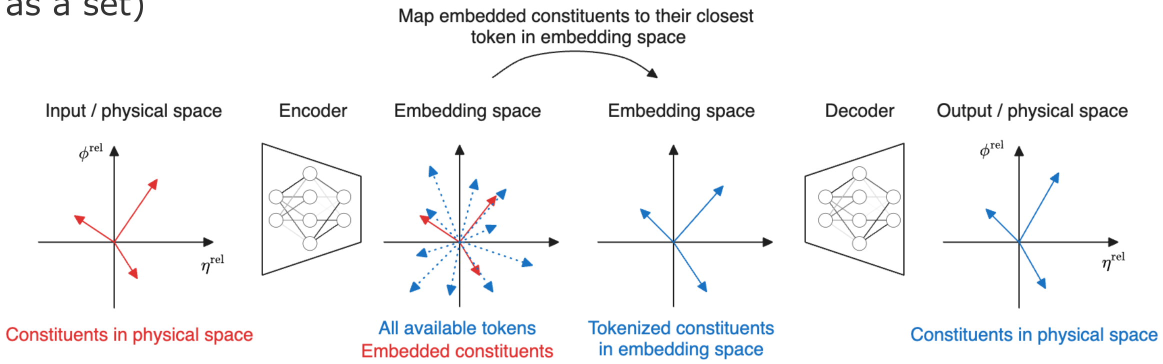
- The GPT model expects integer *tokens*, not continuous numbers
- Binning See eg. 2303.07364 for a generative model using binning



Tokenization

- The GPT model expects integer *tokens*, not continuous numbers
- Binning See eg. 2303.07364 for a generative model using binning
- Vector Quantized VAE (VQ-VAE, 1711.00937, 2305.08842) See also implementations in 2106.08254, 2401.13537
  - unconditional (vectors encoded individually)
  - conditional (vectors encoded as a set)



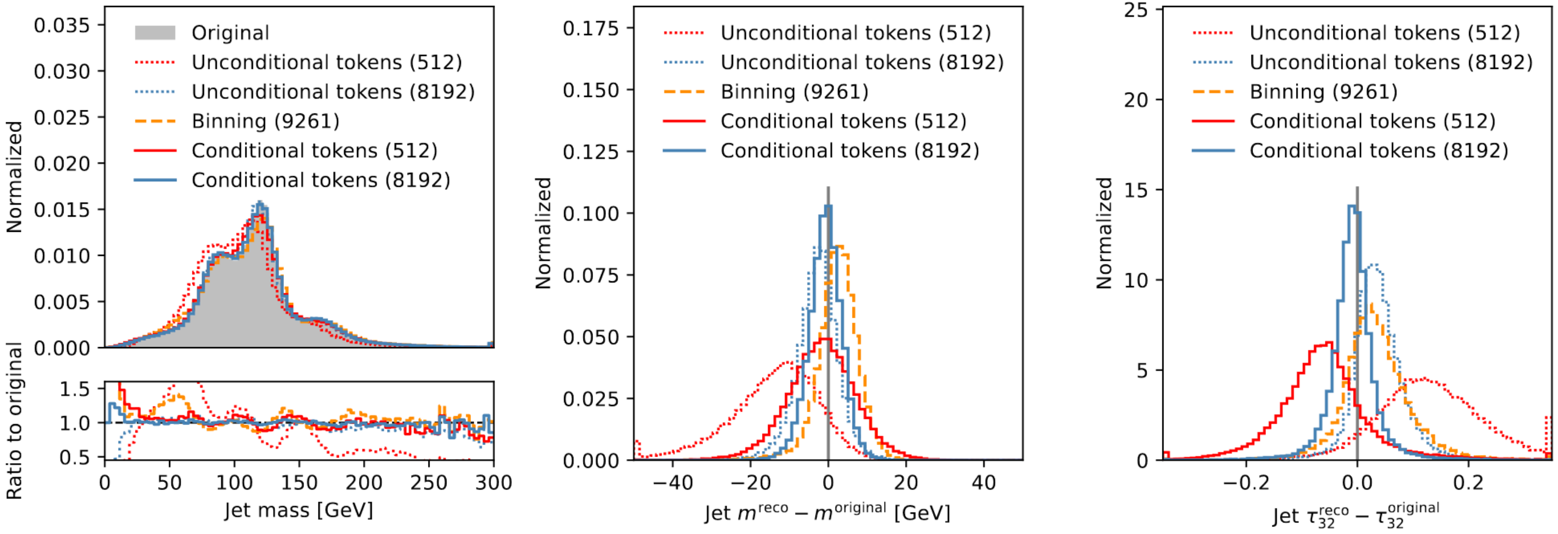


Dataset and tokenization approaches

- JetClass [1]
  - Tokenize all 10 classes to evaluate tokenization performance
  - For pretraining, generation and classification: use 10M *q/g* jets and 10M $t \rightarrow bqq'$ jets
- Use constituent features p_T, η^{rel}, φ^{rel} (rel = relative to the jet axis)
- Test 3 approaches:
  - Binning: 21x21x21 grid
  - VQ-VAE: unconditional (MLP for encoder/decoder) and conditional (transformer for encoder/decoder); codebook sizes 512 and 8192

[1] http://dx.doi.org/10.5281/zenodo.6619767

Tokenization results

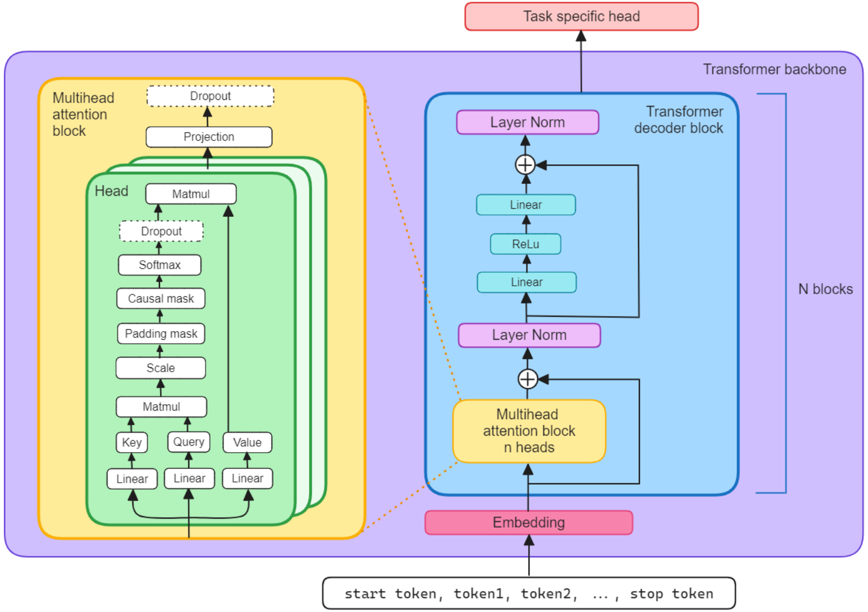


➡ We choose conditional tokens with codebook size 8192

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

The transformer backbone and task specific heads

- **Transformer backbone** takes tokens as input, outputs to task specific head.
- Causal mask prevents attention to future tokens
- Task specific heads
  - **Generation** – linear layer
  - **Classification** – linear layer, ReLU, sum, linear layer, softmax
- n heads = 8, N GPT blocks = 3
  - 6.7M parameters



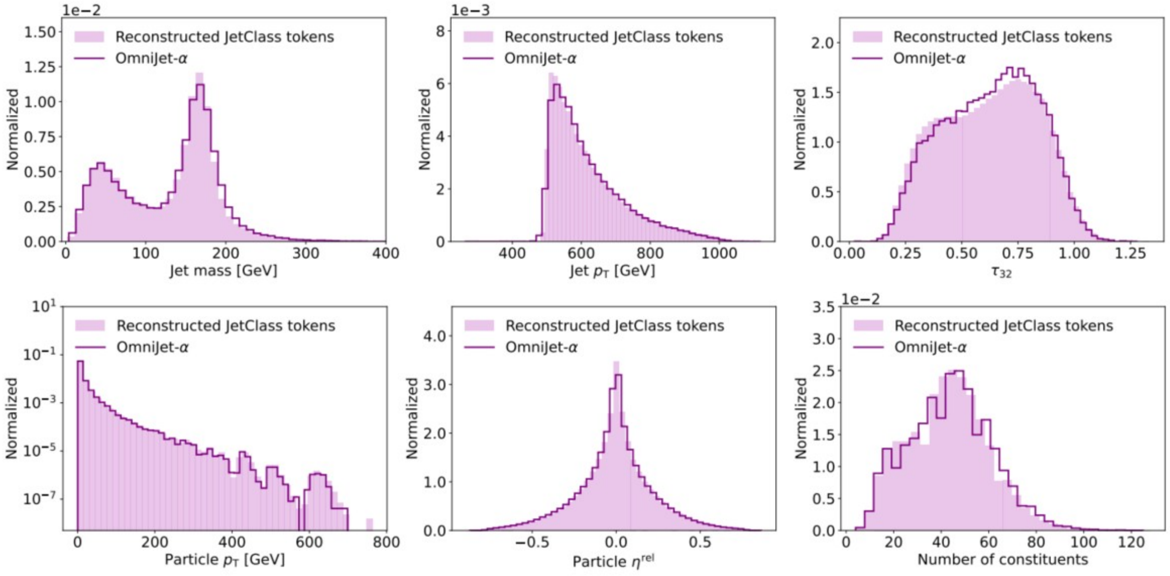
Train with generative head

- Add start and stop token
  - `<start token>, token 1, …, token n, <stop token>`
- Combine *q/g* and $t \rightarrow bqq'$ jets, no labels are passed to the model.

- To generate autoregressively from the trained model:
  - Model has learned $p\left(x_j | x_{j-1}, \ldots, x_1, \texttt{start\_token}\right)$
  - Model recieves `<start token>` and starts generating
  - Model stops if `<stop token>` is generated or the maximum sequence length is reached

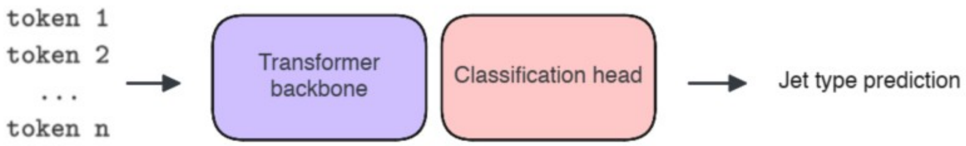
Generative results – reconstructed tokens

- Generally good agreement
- Constituent pT spectrum tail has few events → the limited codebook size shows up as bumps
- A simple classifier is unable to distinguish generated events from the original reconstructed tokens



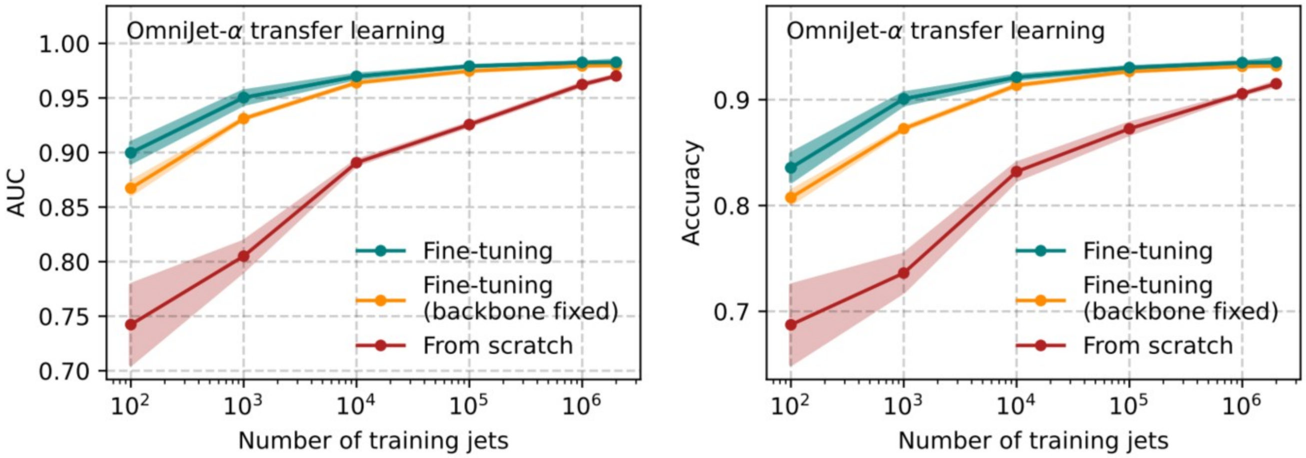
Transfer learning: classify *q/g* vs *t→bqq′*

- "From scratch": all weights are initialized from scratch, no pre-training is used
- Fine-tuning: load weights of the pre-trained generative model, continue the training with the classification head instead of the generative head
  - regular fine-tuning: all weigths can change
  - backbone fixed: weights of the pre-trained transformer backbone are held fixed



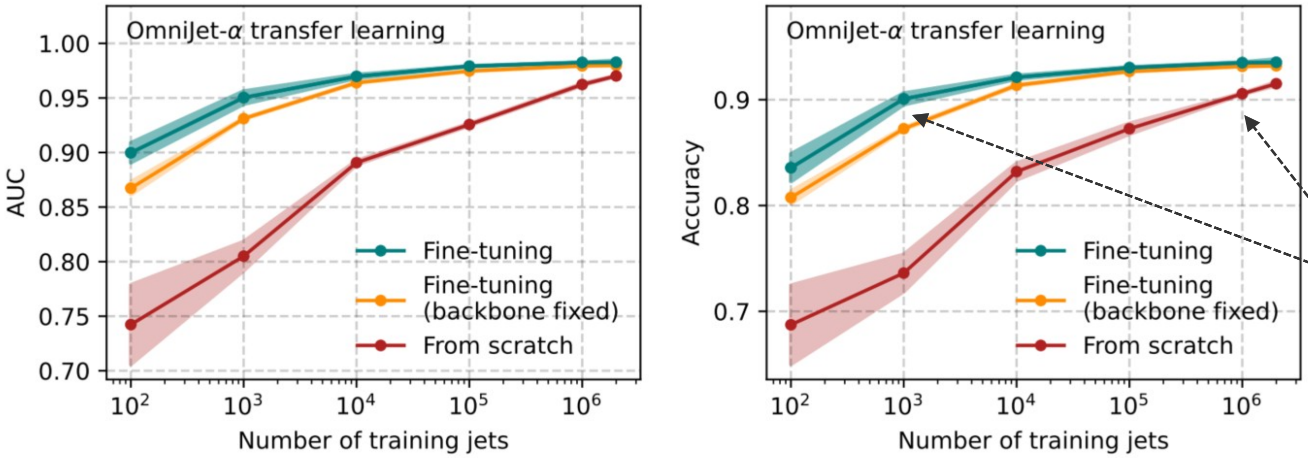
Transfer learning results

- Significantly better result when using pre-training
- Full fine-tuning slightly better than backbone fixed



Transfer learning results

- Significantly better result when using pre-training
- Full fine-tuning slightly better than backbone fixed



Pre-trained model requires only 1000 training events to reach the same accuracy level that the ”from scratch” model reaches with 1M events

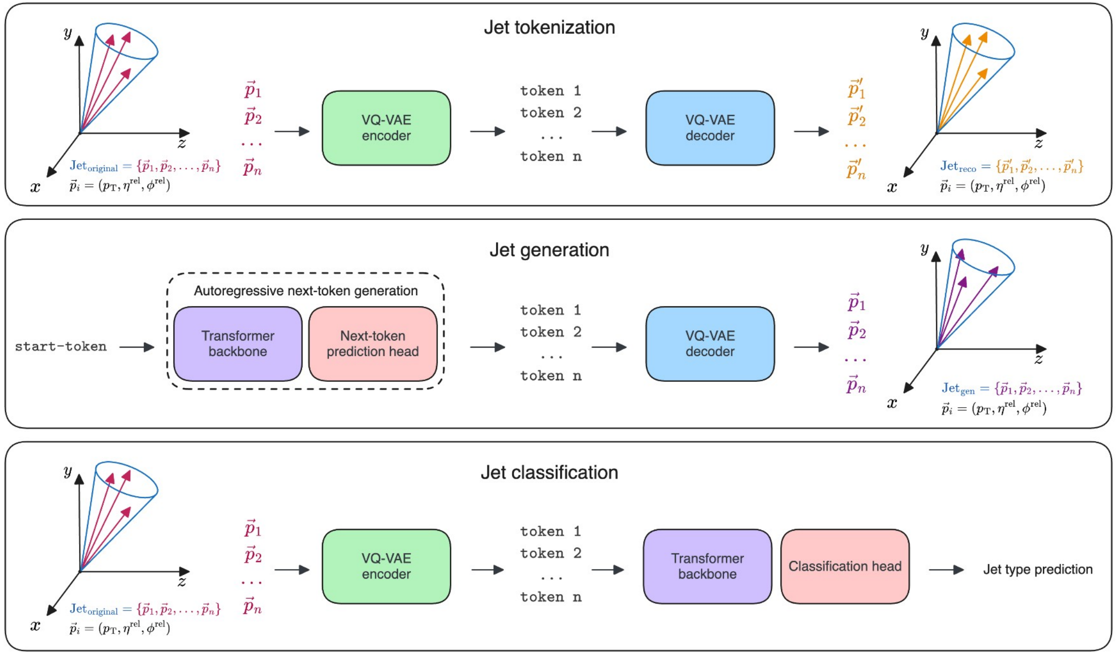
Conclusion

- OmniJet-α is the **first cross-task foundation model for particle physics**
- It is capable of both **generating full jets** and **classifying** q/g and $t \rightarrow bqq'$ jets
- Pre-training offers **significant improvements** in the classifier task compared to training from scratch
- Future work: explore different tokenization schemes, improve the generative model, expand to further tasks, include other features (eg. discrete), train on still larger datasets and more jet types

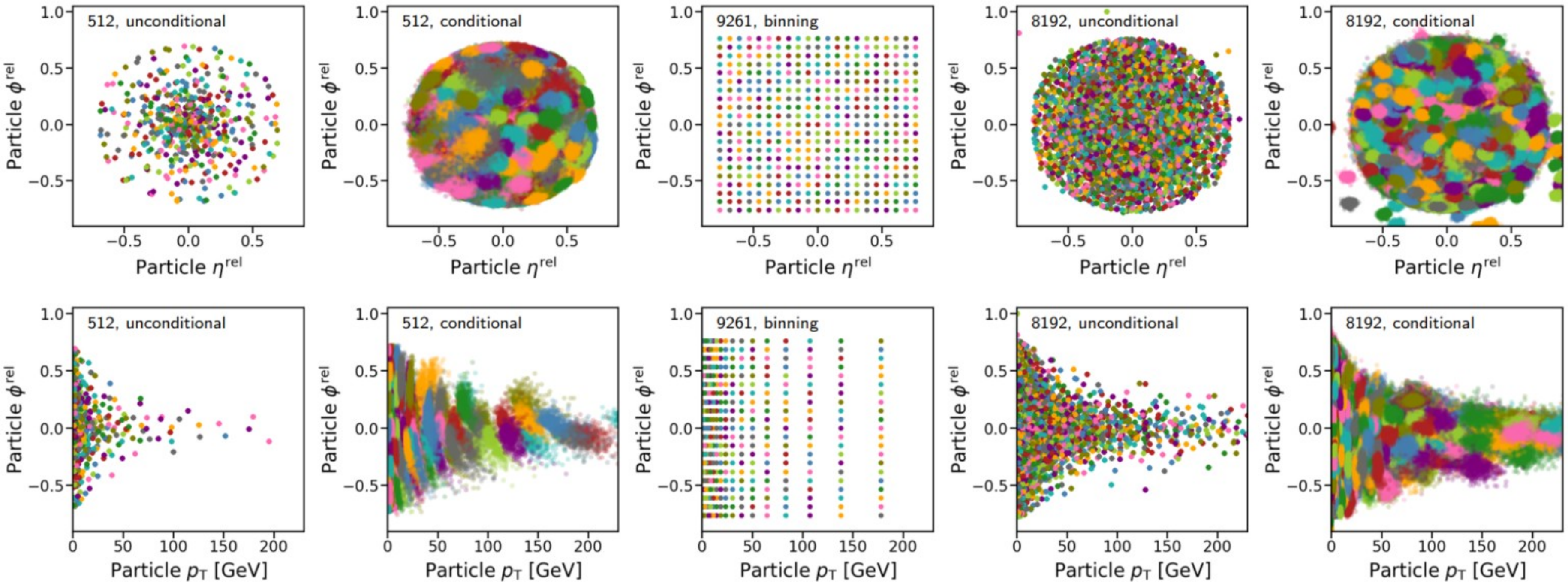
Backup

Workflow

- Jets are **tokenized**
- **Transformer backbone** is trained with the **generative head**
- Generation: autoregressive generation, then **decode** the generated tokens
- Classification: switch the generative head to a **classification head**

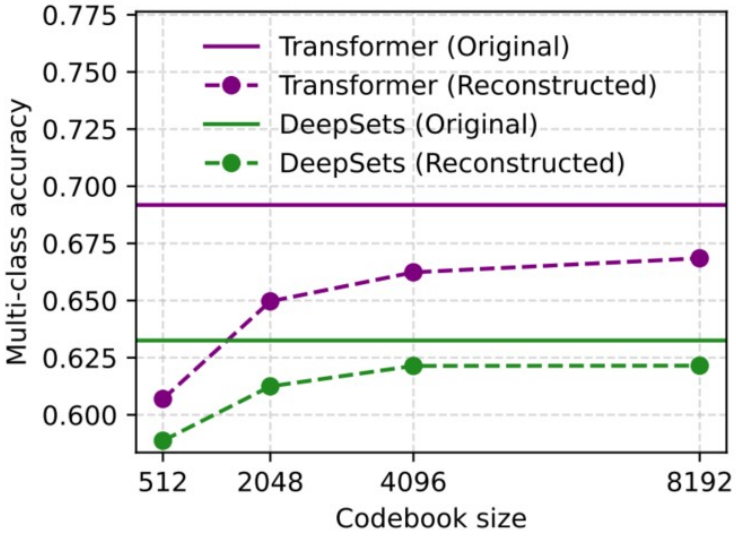


Token reconstruction space

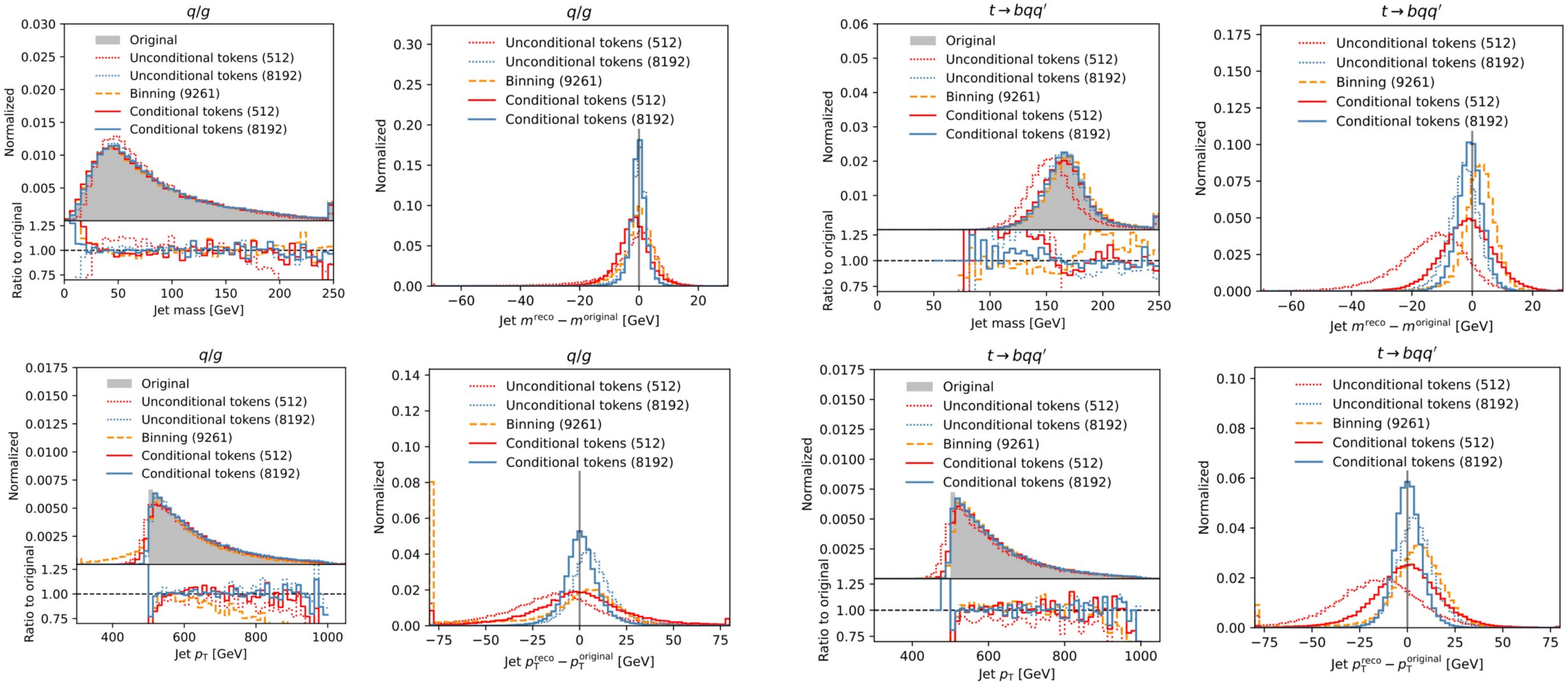


Quantifying tokenization information loss

- Train a multi-class classifier on all 10 classes of JetClass (note: this is not a reconstructed vs truth test)
- Two types of classifiers are tested: transformer and Deep sets
- Train on original JetClass data to obtain an upper limit
- Accuracy starts plateauing at a codebook size of 8192

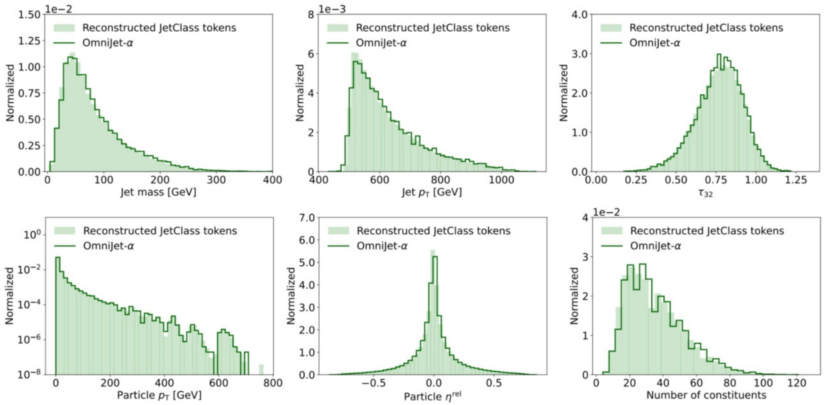


Token quality: distribution and resolution

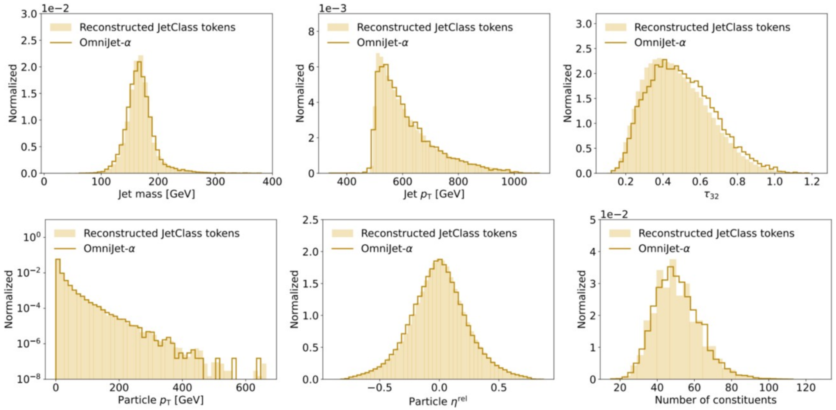


Generative results, single-jet type training

- *q/g* jets



- $t \rightarrow bqq'$ jets



Comparison of generation capabilities, *t* → *bqq'*

- EPiC-FM (2312.00123): flow matching, no tokenization
- Ratios compare OmniJet-α and EPiC-FM to their respective truths
- Both models are doing well
- OmniJet-α has a slightly higher discrepancy in the tails, except for constituent η^{rel} and number of constituents