# heptokens: Jet Tokenization

## Software, Scans, Scale-Out

Jeff Krupa, Sam Klein, Michael Kagan

TREASURE Workshop

April 28, 2026

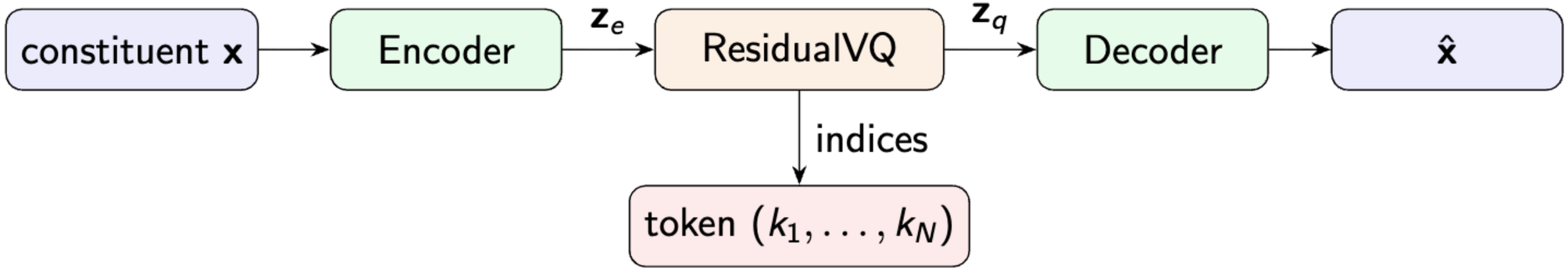
# Overview

1. VQ-VAE: in-context vs no-context encoding
2. heptokens tooling
3. Hyperparameter scans and classification
4. Scale-out

# VQ-VAE Tokenization

$$\mathcal{L} = \underbrace{\|\mathbf{x} - \hat{\mathbf{x}}\|^2}_{\text{recon.}} + \sum_{i=1}^{N} \Big( \underbrace{\|\mathrm{sg}[\mathbf{r}^{(i-1)}] - \mathbf{z}_q^{(i)}\|^2}_{\text{codebook}} + \beta \underbrace{\|\mathbf{r}^{(i-1)} - \mathrm{sg}[\mathbf{z}_q^{(i)}]\|^2}_{\text{commitment}} \Big)$$

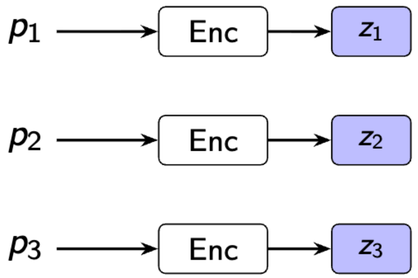
- Each constituent → tuple of N integer codes via residual tokenization
- $z_q$ → continuous vectors, indices are the tokens
- Main hyperparameters: encoder/decoder size, ResidualVQ parameters



# VQ-VAE Tokenization

$$\mathcal{L} = \underbrace{\|\mathbf{x} - \hat{\mathbf{x}}\|^2}_{\text{recon.}} + \sum_{i=1}^{N} \Big( \underbrace{\|\text{sg}[\mathbf{r}^{(i-1)}] - \mathbf{z}_q^{(i)}\|^2}_{\text{codebook}} + \beta \underbrace{\|\mathbf{r}^{(i-1)} - \text{sg}[\mathbf{z}_q^{(i)}]\|^2}_{\text{commitment}} \Big)$$

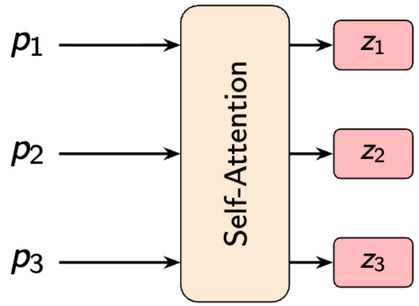
## No-context (MLP)




$$z_i = f_\theta(p_i)$$

Each token is independent

## In-context (transformer)




$$z_i = f_\theta(p_1, \ldots, p_N)_i$$

Tokens carry context of entire jet

# heptokens: overview

```
heptokens/
    src/heptokens/
      data/                    data loading
      models/  tokenizers, classifiers
      callbacks/           training hooks
      utils/                       helpers
    scripts/                 entry points
    configs/                Hydra configs
    workflow/                    Snakemake
```

**Tooling**

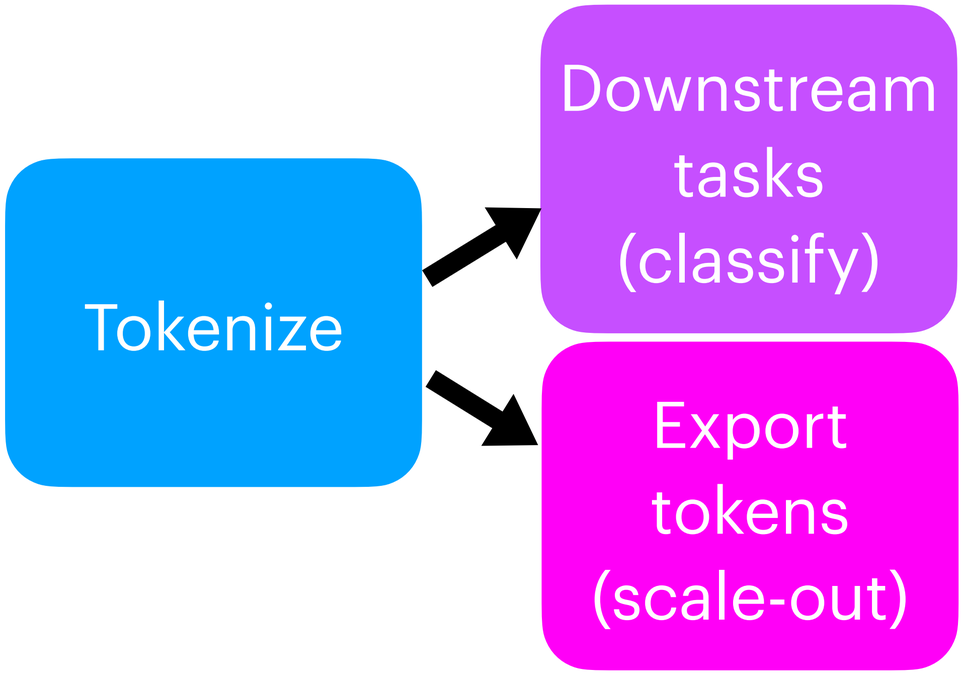
- PyTorch + Lightning
- vector-quantize-pytorch
- Hydra configs (CLI overrides)
- Slurm submission via Snakemake (DAG job scale-out)
- Monitoring via WandB
- Environment managed with Pixi

**Models**

- MLP encoder VQ-VAE
- Transformer encoder VQ-VAE
- k-means baseline
- Others in development

# Hyperparameter scans with Snakemake

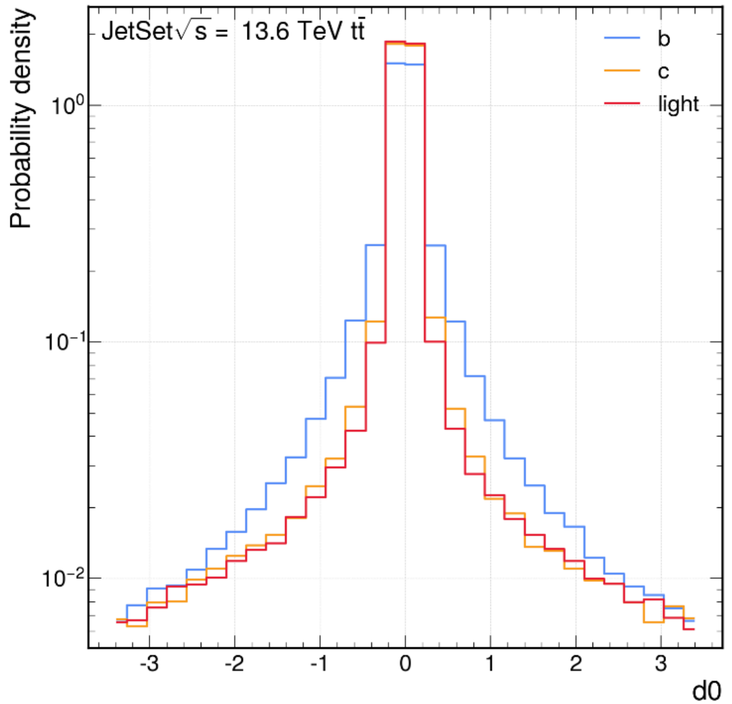
Snakemake facilitates **multi-stage workflows**

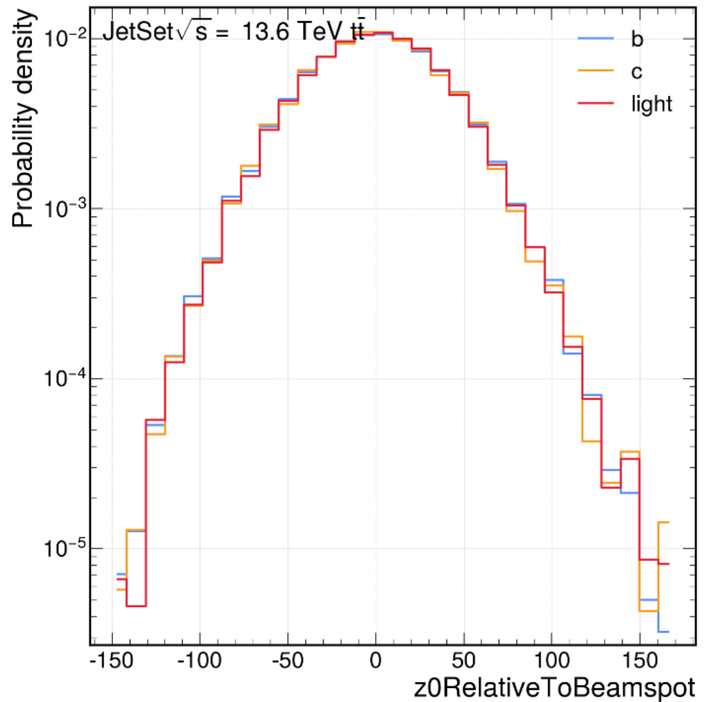



```
# parameters to scan
ENCODER         = ["mlp","transformer"]
ENCODER_SIZES   = ["small","medium","large"]
CODEBOOK_SIZES = [512, 8192, 32768]
CODEBOOK_DIMS   = [4,8,16]
NUM_QUANTIZERS = [1,2,3,4]
```

# Dataset: JetSet

- Large-scale ATLAS MC dataset
  - 200M *tt* at 13.6 TeV
  - 20 track features, jet features
  - Truth labels: b, c, light, tau
  - Up to 40 tracks per jet
- Developed by flavor tagging group in ATLAS for GN2 + ML algorithm benchmarking
- We use it for tokenizer R&D (10M training baseline)

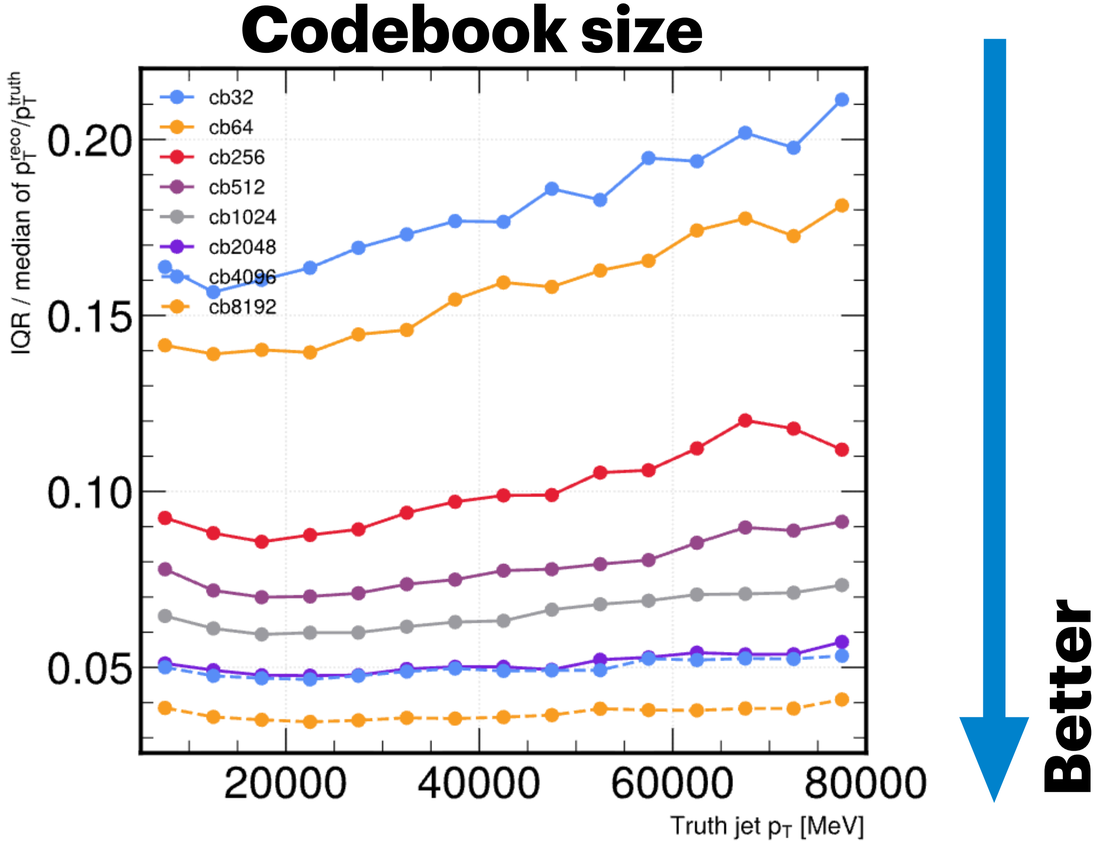


# Scan Overview

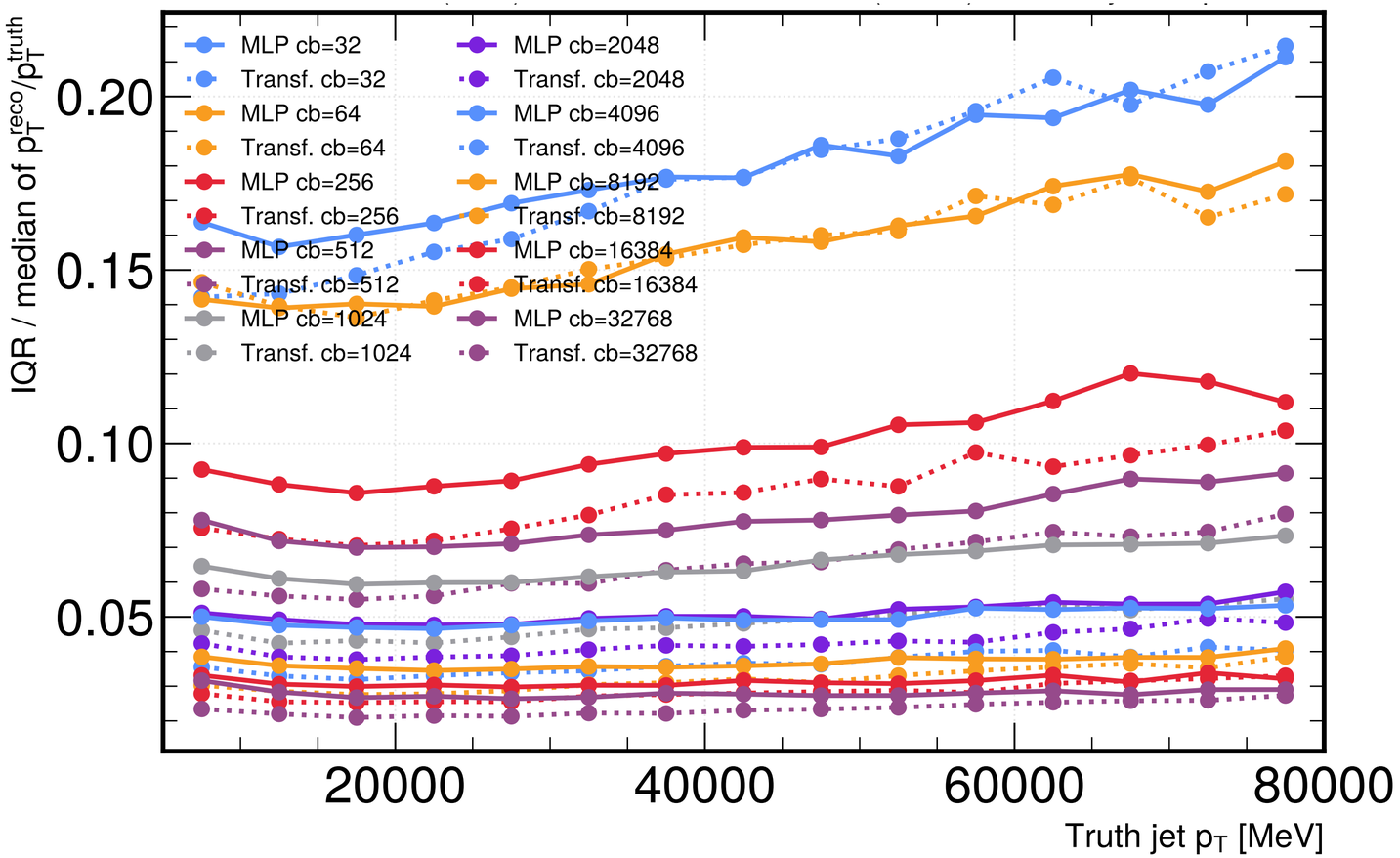
- Dataset: 10M JetSet (default)
- Training:
  - Up to 100 epochs + early stopping
  - LR scans
  - Various preprocessing supported
- Tokenizer scan axes: encoder type, encoder size, codebook size, codebook dimension, number of quantizers
- Also classifiers: encoder size, raw continuous features vs tokens

# Initial Scan: MLP encoder takeaways



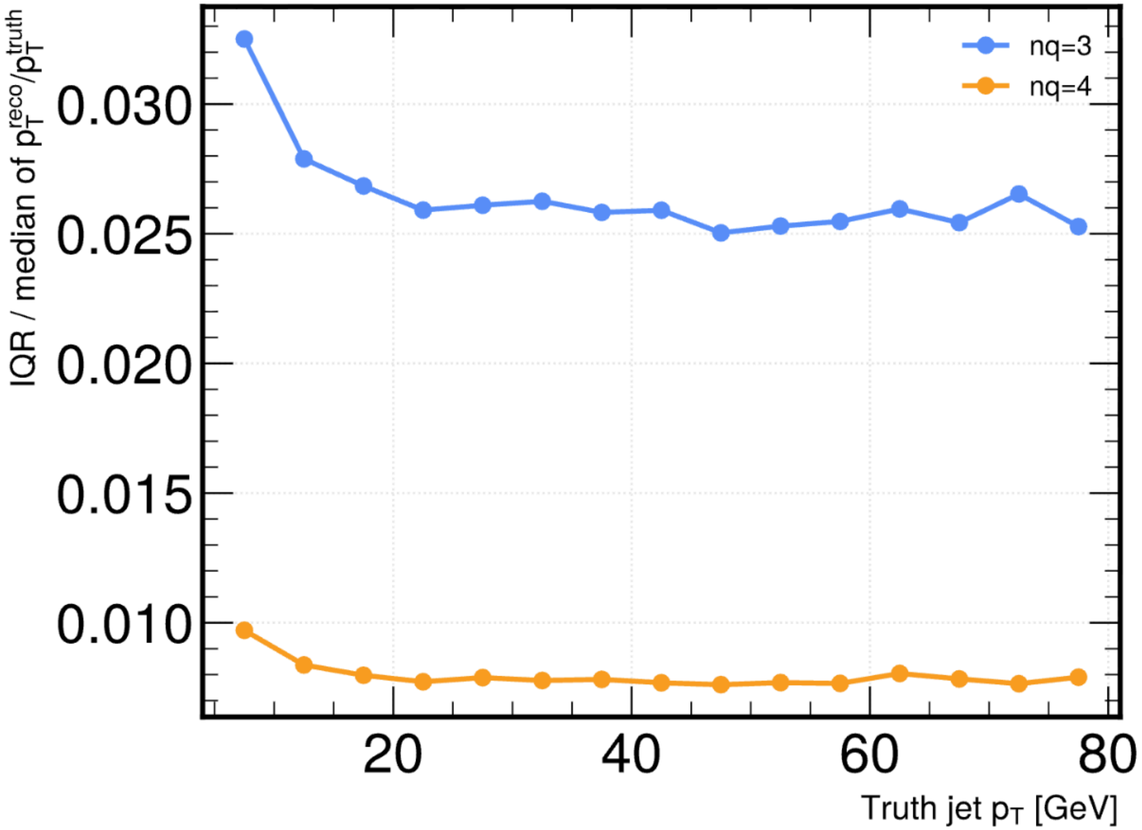

- Early observations:
  - Increased codebook size → **clear improvement**
  - Increased number of quantizers → **clear improvement**
  - Increased codebook dimension → **unclear**
- This motivates focused scan with larger codebooks

# In-context Encoding Improves Reconstruction



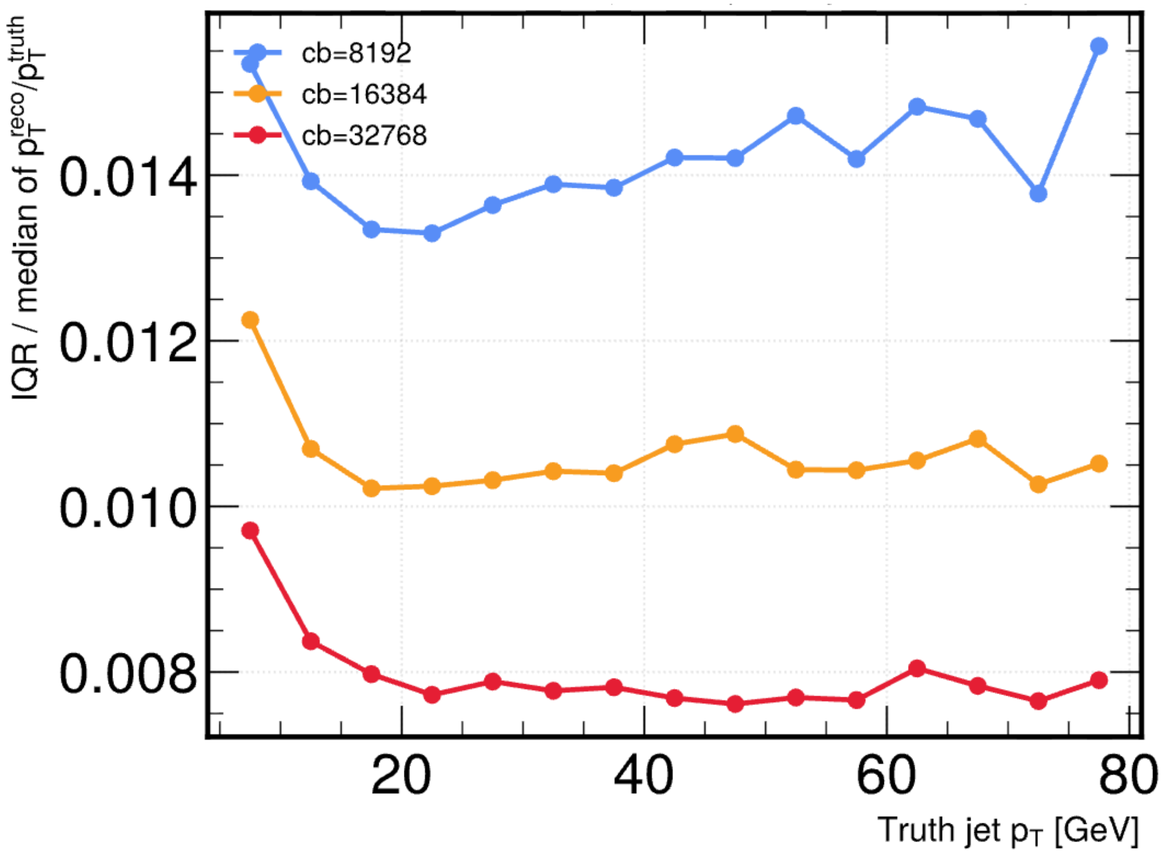

- Transformer outperforms MLP
- Both benefit from larger codebook size → push to 8192, 32768 codebook sizes

# Number of quantizers




- Fixed codebook size = 32768, codebook dims = 8
- More residual quantizers → finer reconstruction
- *N* = 4 reaches <1% jet $p_T$ resolution

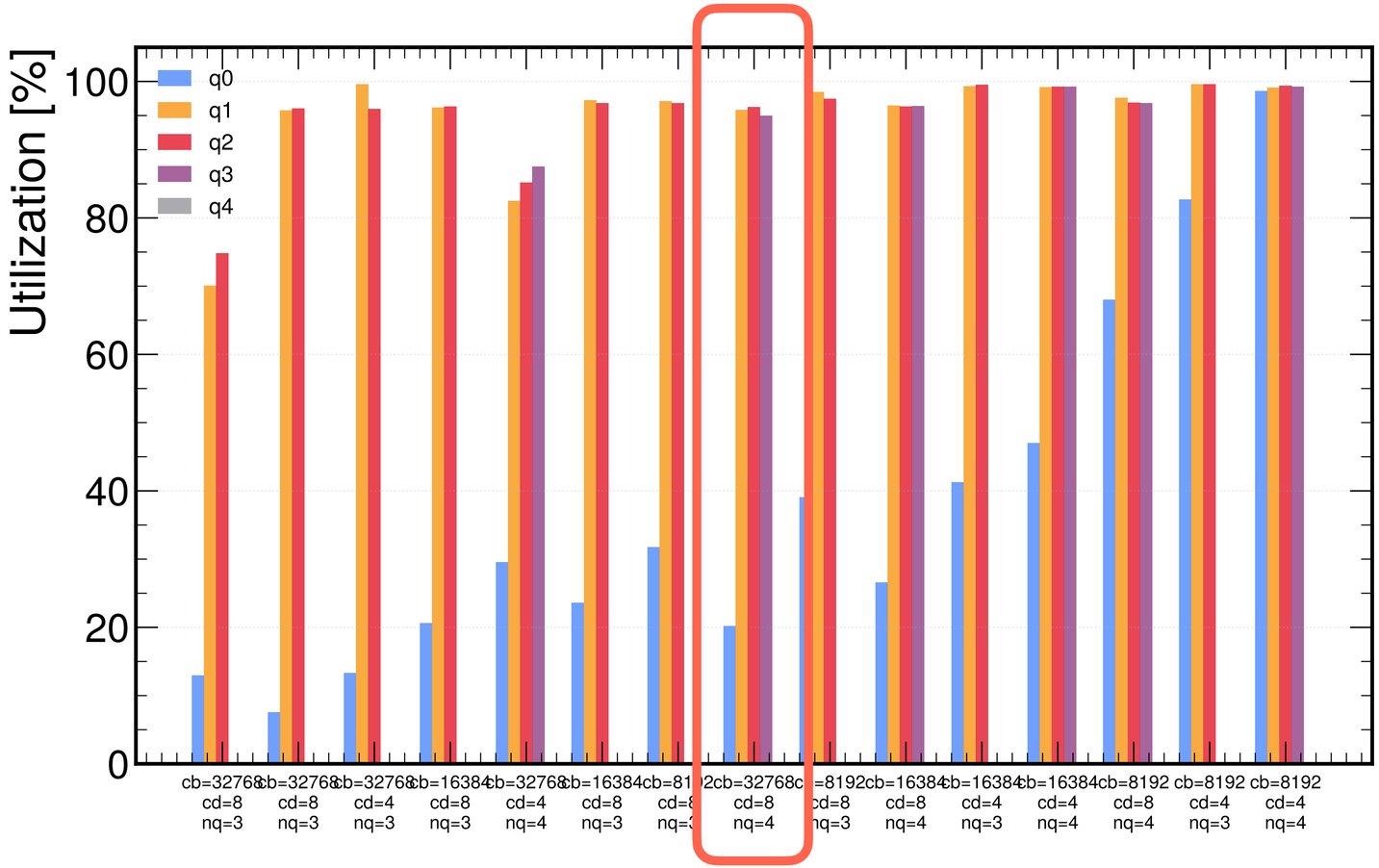
# Codebook size




- Fixed codebook dims = 8, number of quantizers = 4
- Larger codebooks → finer reconstruction
- Systematic improvement from 512 to 32768, 32768 reaches <1% jet $p_T$ resolution

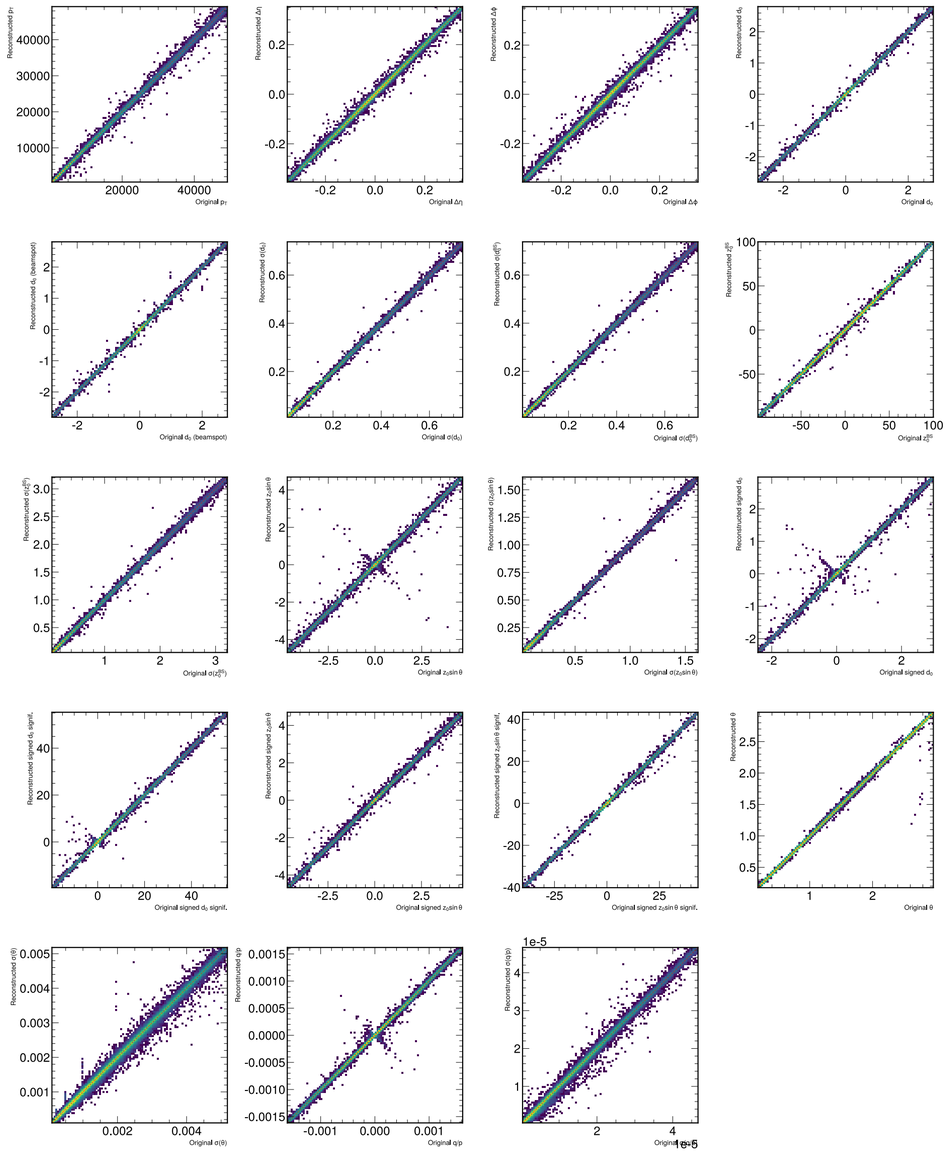
# Codebook utilization

Decreasing codebook size




- Per-quantizer utilization
- Later quantizers show more utilization than first layer

# Best Model



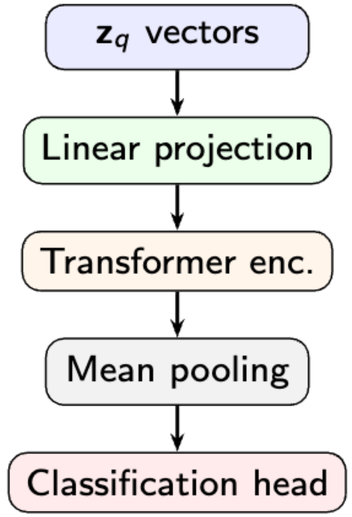
Best model:

- Transformer (in-context) VQ-VAE
- Codebook size 32768
- Codebook dimension 8
- Number of quantizers 4

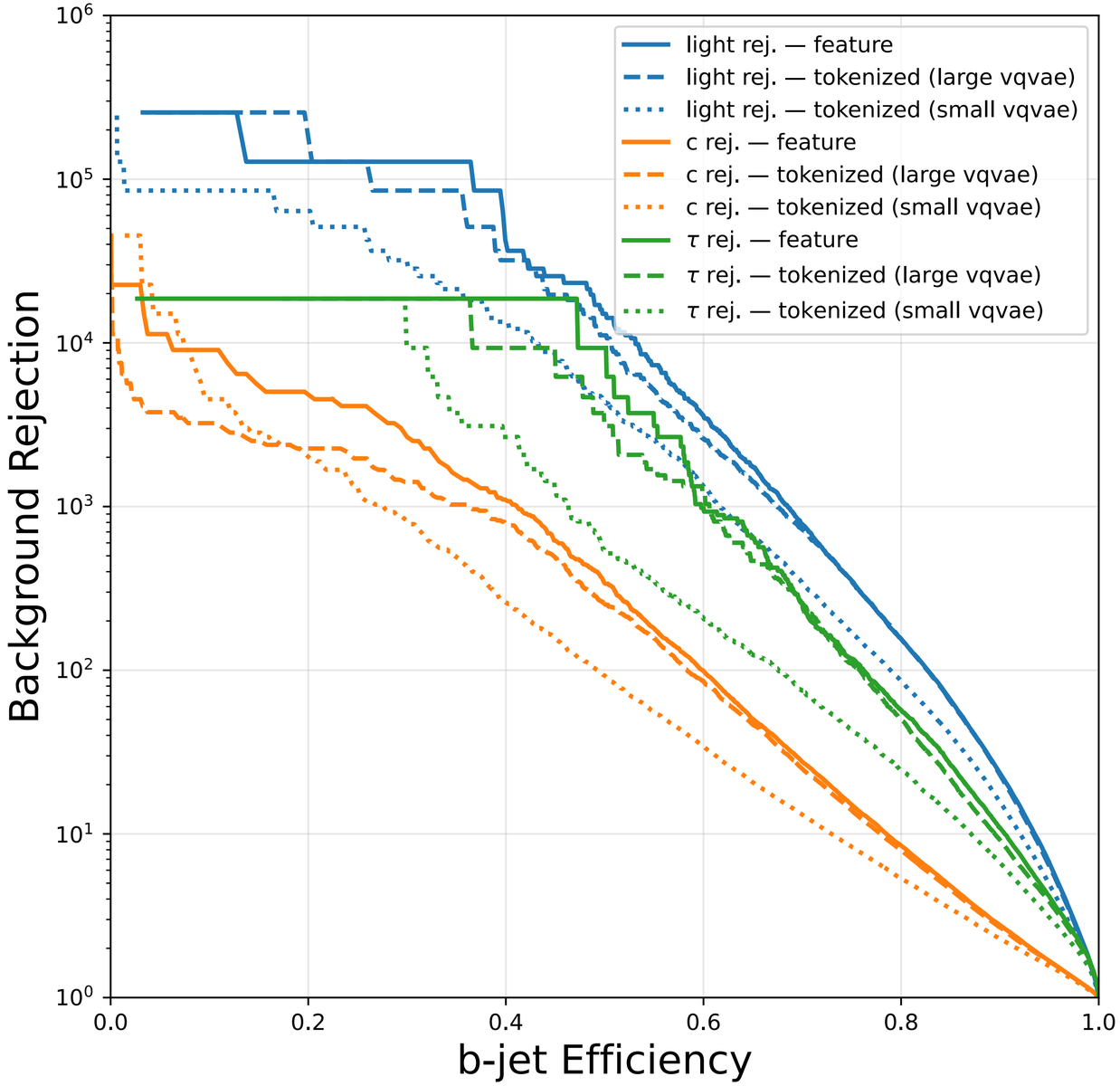
Sub-1% jet $p_T$ reconstruction
Reco-truth feature corr > 0.999

# Classification

## Architecture




| Classifier | Macro AUC |
|---|---|
| Features (baseline) | 0.924 |
| Vector (cb512, nq3) | 0.907 |
| Vector (cb32768, nq4) | 0.922 |



# Classification

**Bkg rejection @80% b-tagging efficiency**

| | light | $c$ | $\tau$ |
|---|---|---|---|
| Raw features | **156** | **8** | **57** |
| VQ (cb512) | 86 | 5 | 25 |
| VQ (cb32768) | 155 | 8 | 50 |

- Residual VQ-VAE with a large codebook has competitive performance to continuous features

# Scale-out tokenization

**Pipeline**: `heptokens/scripts/export_tokens.py`

1. Load VQ-VAE checkpoint
2. Create data module with Hydra
3. Run predictions on GPU
4. Save indices + labels + codebooks

**Output format**

1. Indices [N, 40, $N_q$] (int)
2. Labels [N] (int)
3. Codebooks [$N_q$, codebook size, codebook dim] (float)

**Throughput (A100-40GB)**:

4.3k jets/s

~13 hours for 200M events

# Summary

**Software:**

1. `heptokens`: modular, config-driven tokenization
2. Supports reproducible environments, scalable workflows
3. Scales out tokenization pipelines
4. Next steps: extend to other datasets, modalities, tokenization schemes ...

**Results:**

1. Systematic residual VQ-VAE scan: architecture, codebook size, dim, quantizers
2. In-context encoding achieves superior reconstruction to no-context
3. Classifier with large codebook has competitive performance

# Backup

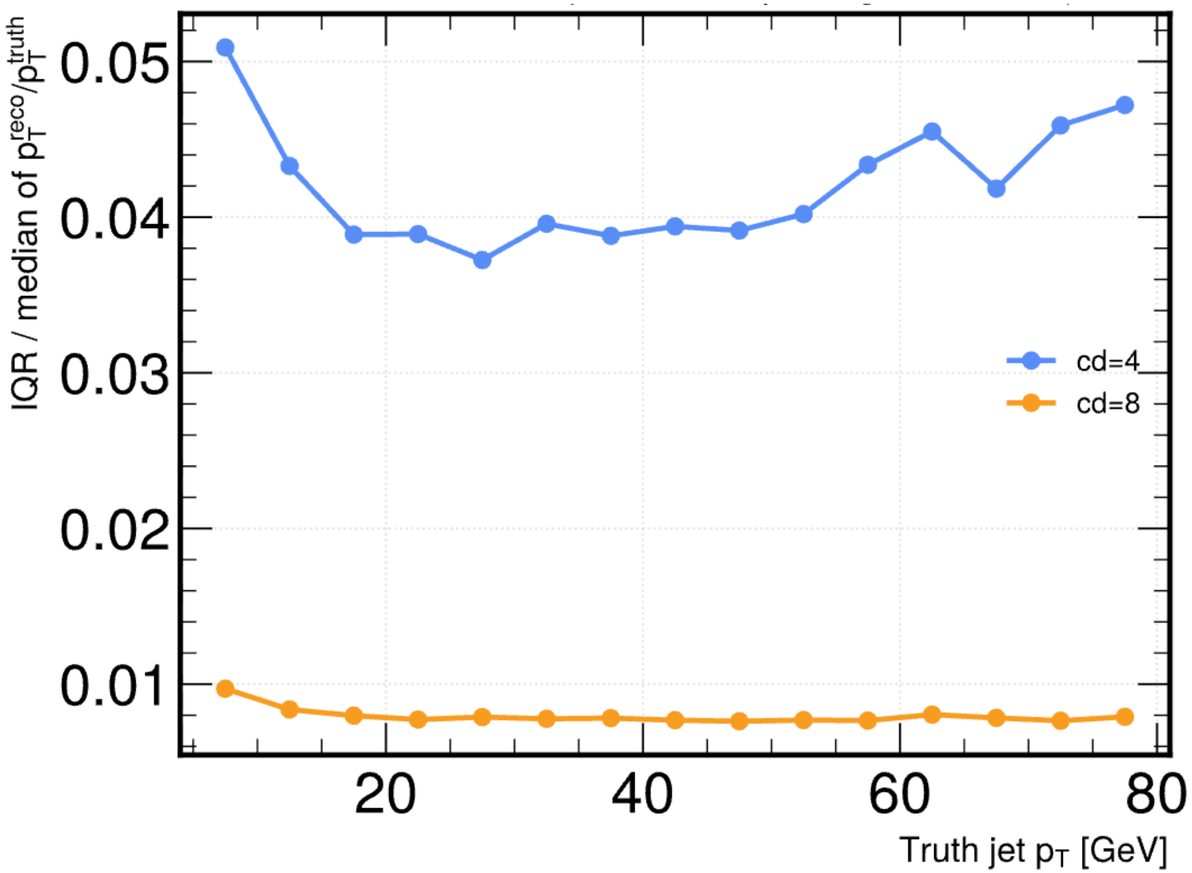
# Customizing with Hydra

- Customize anything at run-time with overrides
- Full config info saved

```
# Switch model, tune codebook
python scripts/train.py \
  model=vqvae \
  model.codebook_size=8192 \
  model.codebook_dim=8 \
  model.num_quantizers=4

# Scale dataset
python scripts/train.py \
  datamodule.num_jets=10_000_000
```

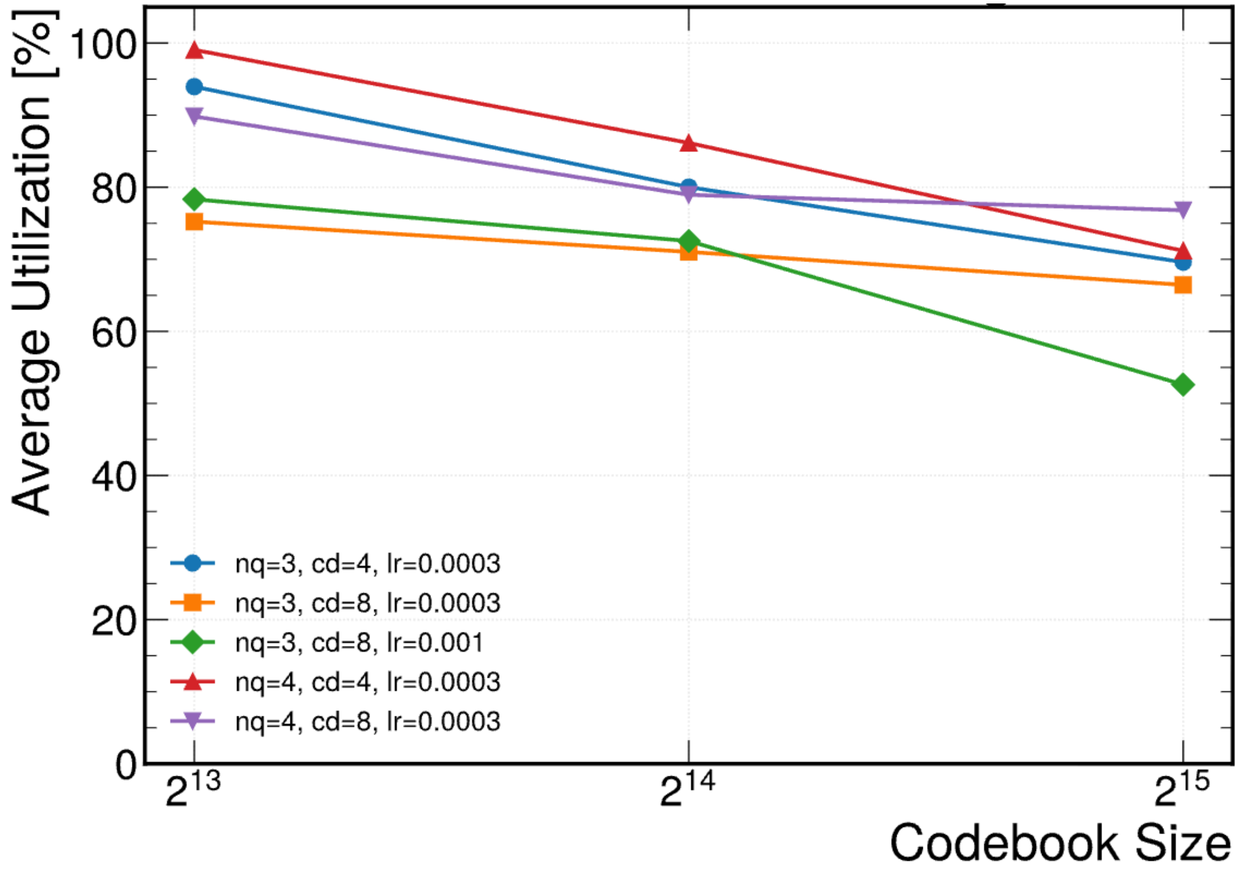
```
# configs/train.yaml
defaults:
  - callbacks: encode
  - datamodule: atlas_mappable
  - model: transformer_vqvae

trainer:
  max_epochs: 100
  precision: bf16-mixed
  gradient_clip_val: 1.0
```

# Codebook dimension




- Fixed codebook size = 32768, number of quantizers = 4
- Codebook size 8 gives best resolution

# Codebook utilization vs size




- Utilization averaged over quantizers decreases with codebook size

# Encoder size and dataset size scans

**Encoder size**

(cb=32768, cd=8, nq=4, 10M jets)

| Encoder | Avg. IQR/med |
|---|---|
| Small (d=64, L=2) | 0.94% |
| Large (d=256, L=6) | 0.80% |

Larger encoder gives ∼14% improvement

**Training dataset size**

(large transformer, cb=32768, cd=8, nq=4)

| Training jets | Avg. IQR/med |
|---|---|
| 1M | 0.80% |
| 10M | 0.80% |
| 20M | 0.83% |

Insensitive to training size — 1M sufficient

# Features

pt
deta
dphi
d0
d0RelativeToBeamspot
d0Uncertainty
d0RelativeToBeamspotUncertainty
z0RelativeToBeamspot
z0RelativeToBeamspotUncertainty
z0SinTheta
z0SinThetaUncertainty
lifetimeSignedD0
lifetimeSignedD0Significance
lifetimeSignedZ0SinTheta
lifetimeSignedZ0SinThetaSignificance
theta
thetaUncertainty
qOverP
qOverPUncertainty

pt_btagJes
GN2v01_pb
GN2v01_pc
GN2v01_pu
GN2v01_ptau
DL1dv01_pu
DL1dv01_pc
DL1dv01_pb
eta_btagJes
pt
mass
ptFromTruthJet
etaFromTruthJet
phiFromTruthJet
mFromTruthJet
deltaEtaToTruthJet
deltaPhiToTruthJet
PartonTruthLabelPt
eta
phi
PartonTruthLabelDR
isJvtPU
isJvtHS
matchedToTruthJet
HadronConeExclTruthLabelID
HadronConeExclExtendedTruthLabelID
HadronGhostTruthLabelID
HadronGhostExtendedTruthLabelID
PartonTruthLabelID
eventNumber