Finetuning Foundation Model for Analysis Optimization

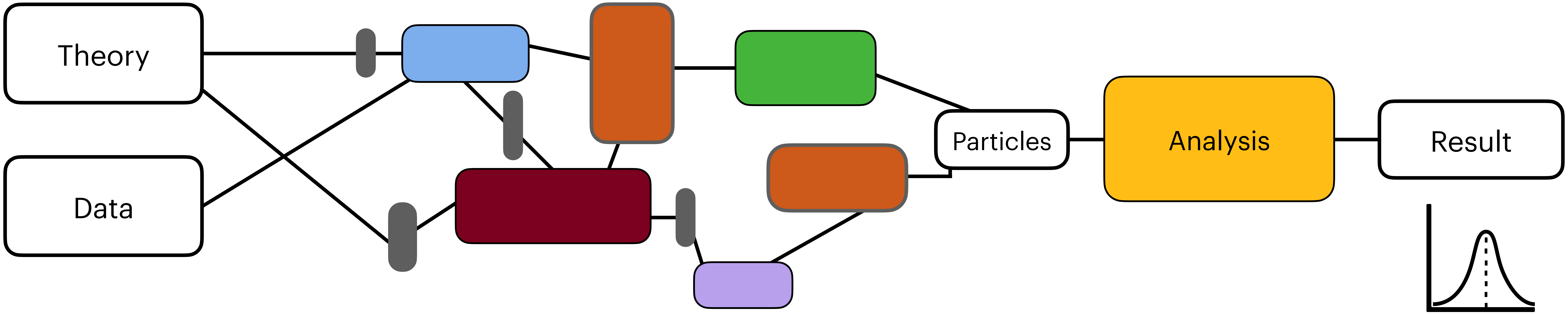
Lukas Heinrich, Nicole Hartman, Matthias Vigl

ML4Jets 2023 – November 6th

Analysis pipeline at the LHC

Lots of (also ML) components in our analysis pipeline

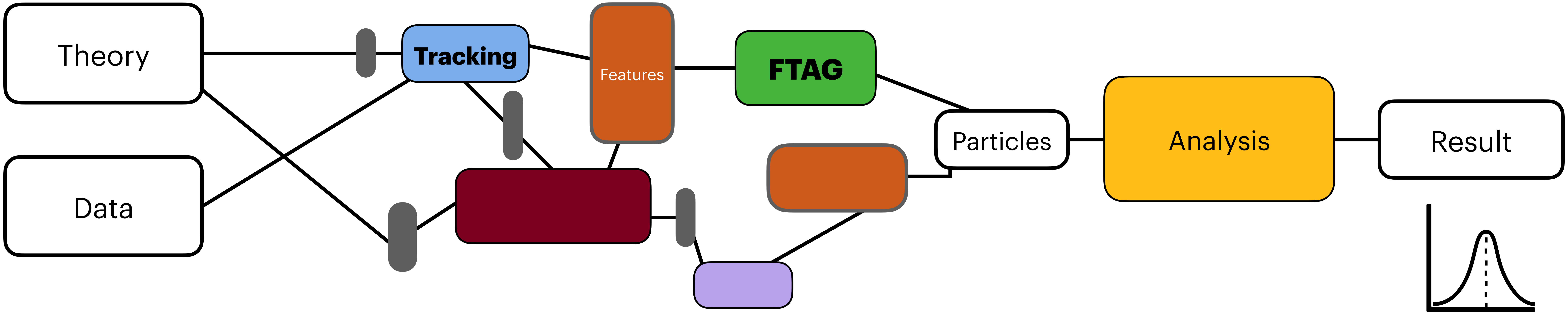
But each optimized separately and downstream components are optimized based on the steps prior to it



Analysis pipeline at the LHC

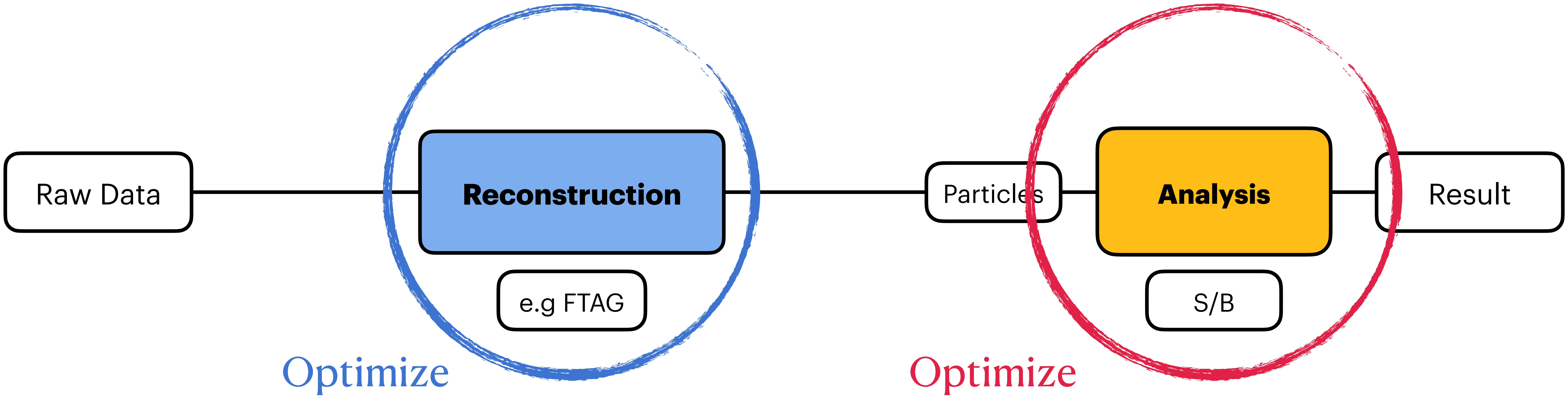
e.g. b-tagging can only be optimized after tracking, but we rarely re-optimize tracking for b-tagging

We optimize the parameters of the reconstruction and then freeze them



Analysis pipeline at the LHC

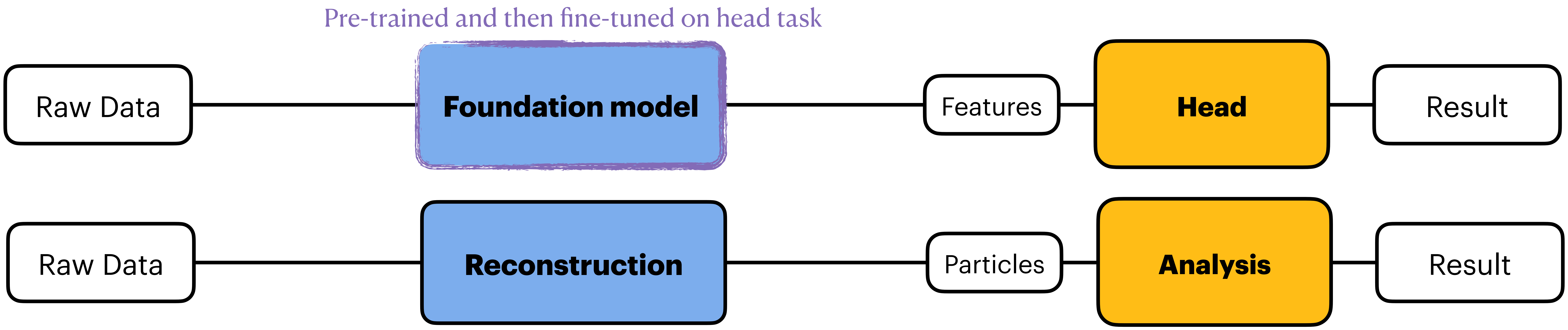
The optimization of the sensitivity is primarily the job of the **analysis**, given a fixed **reconstruction** - mostly common for all analysis



End-to-end Deep Learning

ML and HEP setups are fortunately very aligned

Also often split in two parts, but key difference is that backbone can be fine-tuned w/ gradient descent



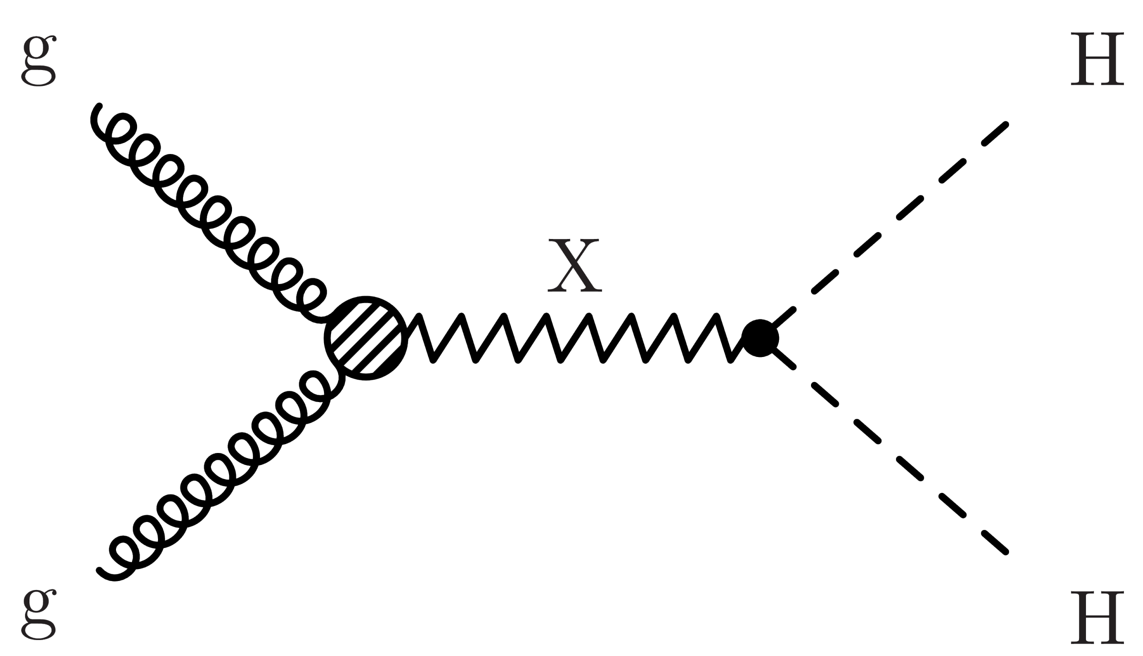
Q: Could this workflow also work in HEP?

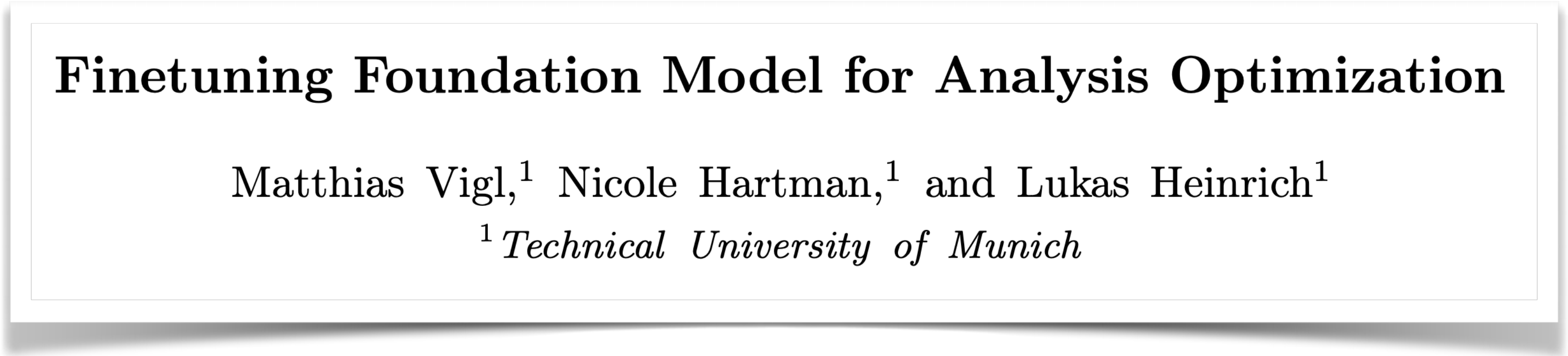
- fine-tuning is now standard in large-scale ML - introduced in HEP with e.g. neos and inferno

[https://arxiv.org/abs/2203.05570] [https://arxiv.org/abs/1806.04743]

A toy end-to-end Analysis

X → HH → 4b. Final state with Higgs/QCD Jets

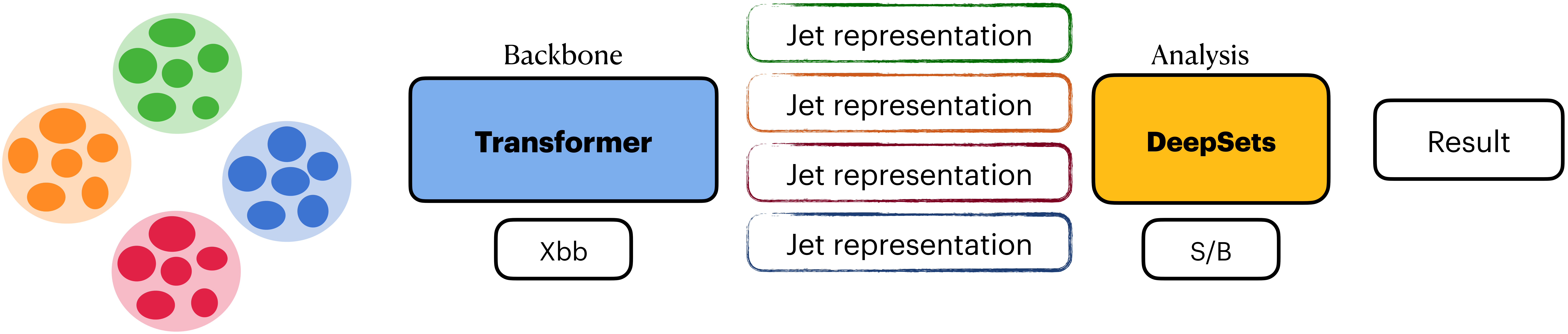



Finetuning Foundation Model for Analysis Optimization

Matthias Vigl,[1] Nicole Hartman,[1] and Lukas Heinrich[1]

[1] *Technical University of Munich*

to appear arxiv:23XX.XXXX



Setup: CMS open data and ParT

[http://opendata.cern.ch/record/12102]

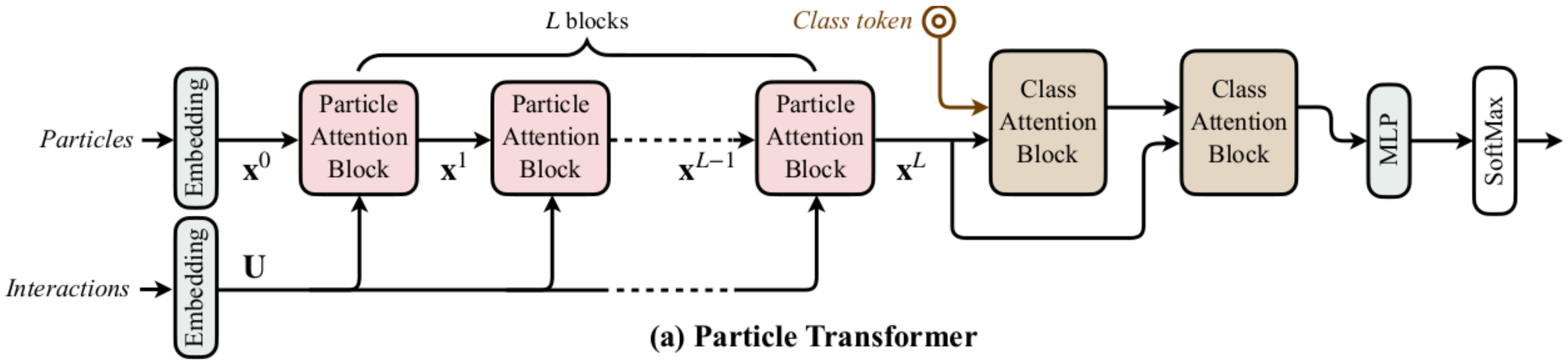
Jets are clustered using the anti-kT algorithm with R=0.8 from particle flow (PF) candidates

Constituents features:

- up to 100 PF per jet
- 17 features per PF

High-level features:

- Jet 4-momenta
- Xbb scores from ParT



(a) Particle Transformer

Particle transformer for FTAG [arXiv:2202.03772]

Training: QCD vs Higgs jets

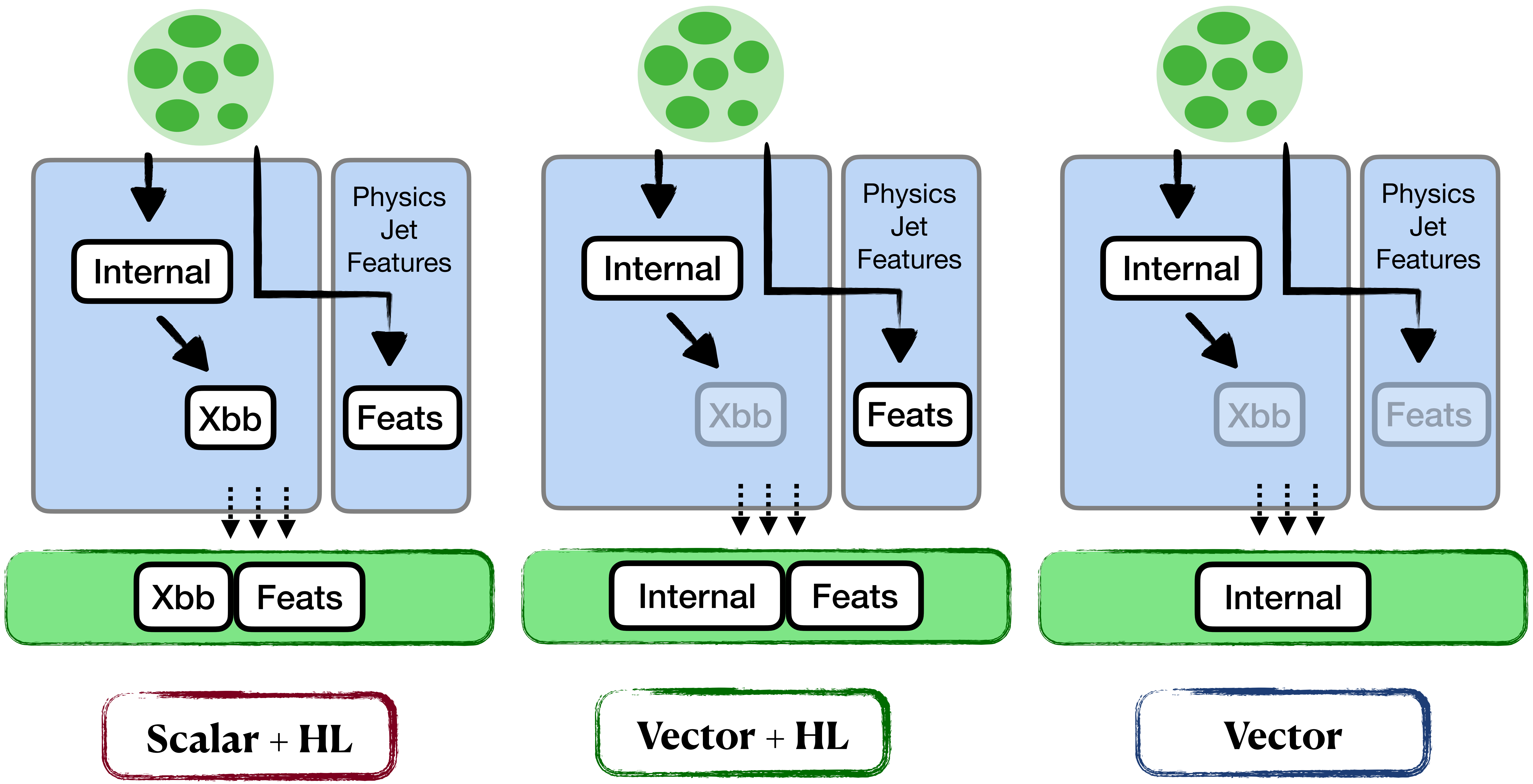
10M events / 22M jets

Backbone Jet representation

Analysis would typically use Xbb + HL features

ParT comes up with its own Internal representation (128 dim) when learning about jet flavour

Q: Do high-dim embeddings hold more (useful) info than Xbb+HL features?

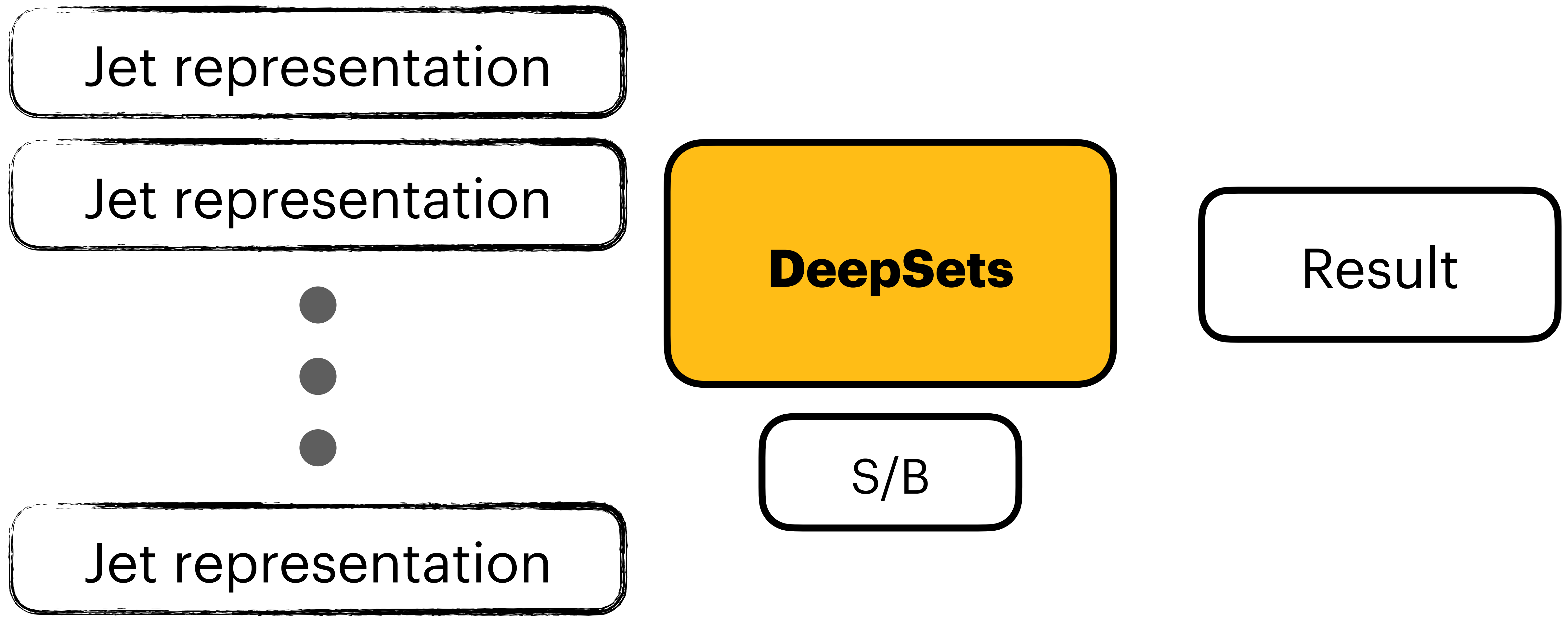


Analysis head

The head is trained for S/B discrimination with Jet representations from backbone as inputs

Variable number of jets per event + Permutation Invariance -> DeepSets

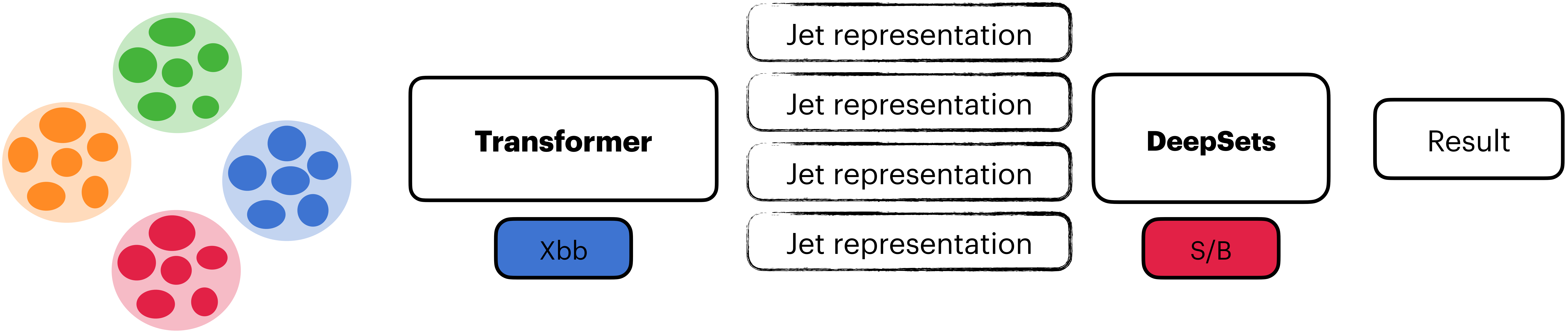
Q: Does fine-tuning the jet representation help?



Frozen training

ParT backbone trained on **Xbb** task and then frozen

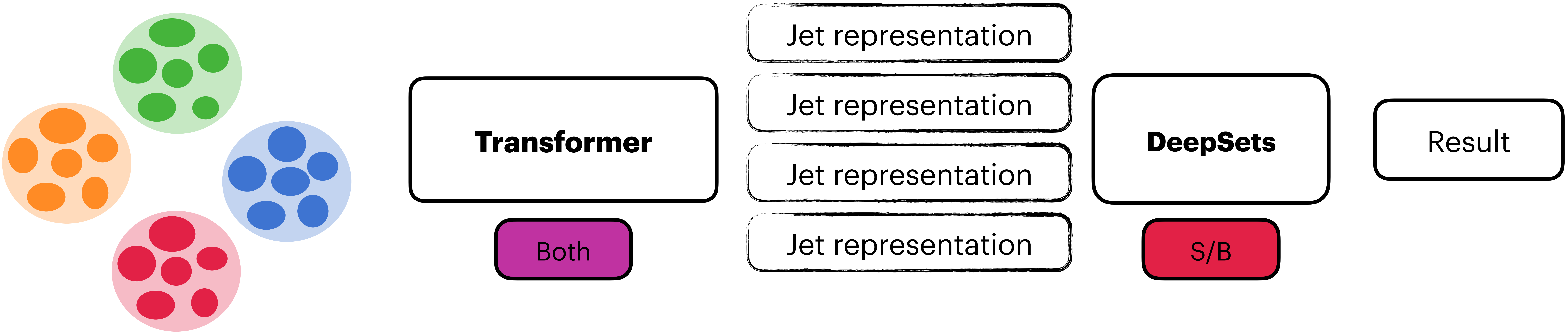
DeepSets + binary classification trained on **S/B**



Fine-tuned training

ParT backbone pre-trained on **Xbb** task

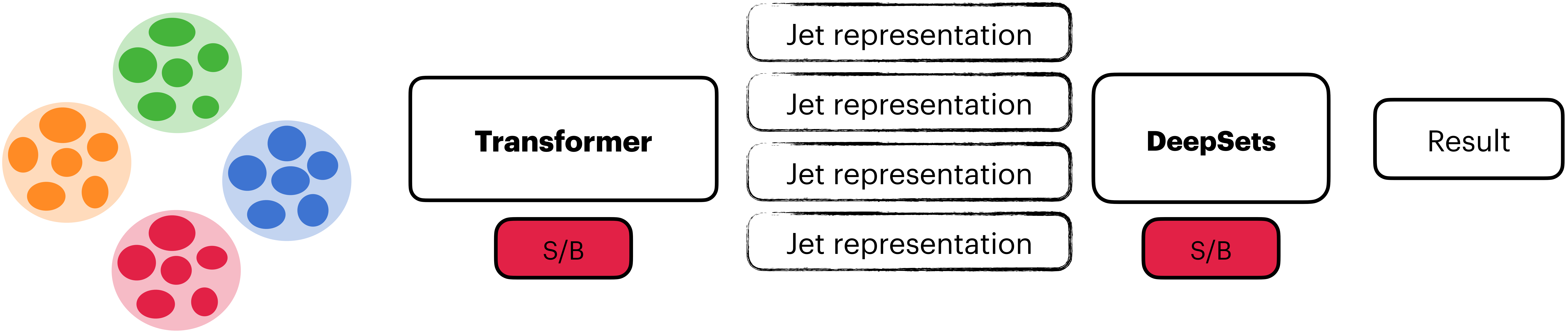
Then **fine-tuning** on **S/B**



From scratch training

No backbone pre-training

Backbone + head trained from scratch on **S/B**



Trainings in summary

Structural Autonomy

Representational Autonomy

Training \ Architecture	Scalar + HL	Vector + HL	Vector
Frozen	Standard HEP		Hope for sufficient stat
Fine-tuned		ML-assisted HEP	
From scratch	Inductive Bias is all you need		'Hits to Higgs'

Q: Could we just train from scratch? Does pre-training matter?

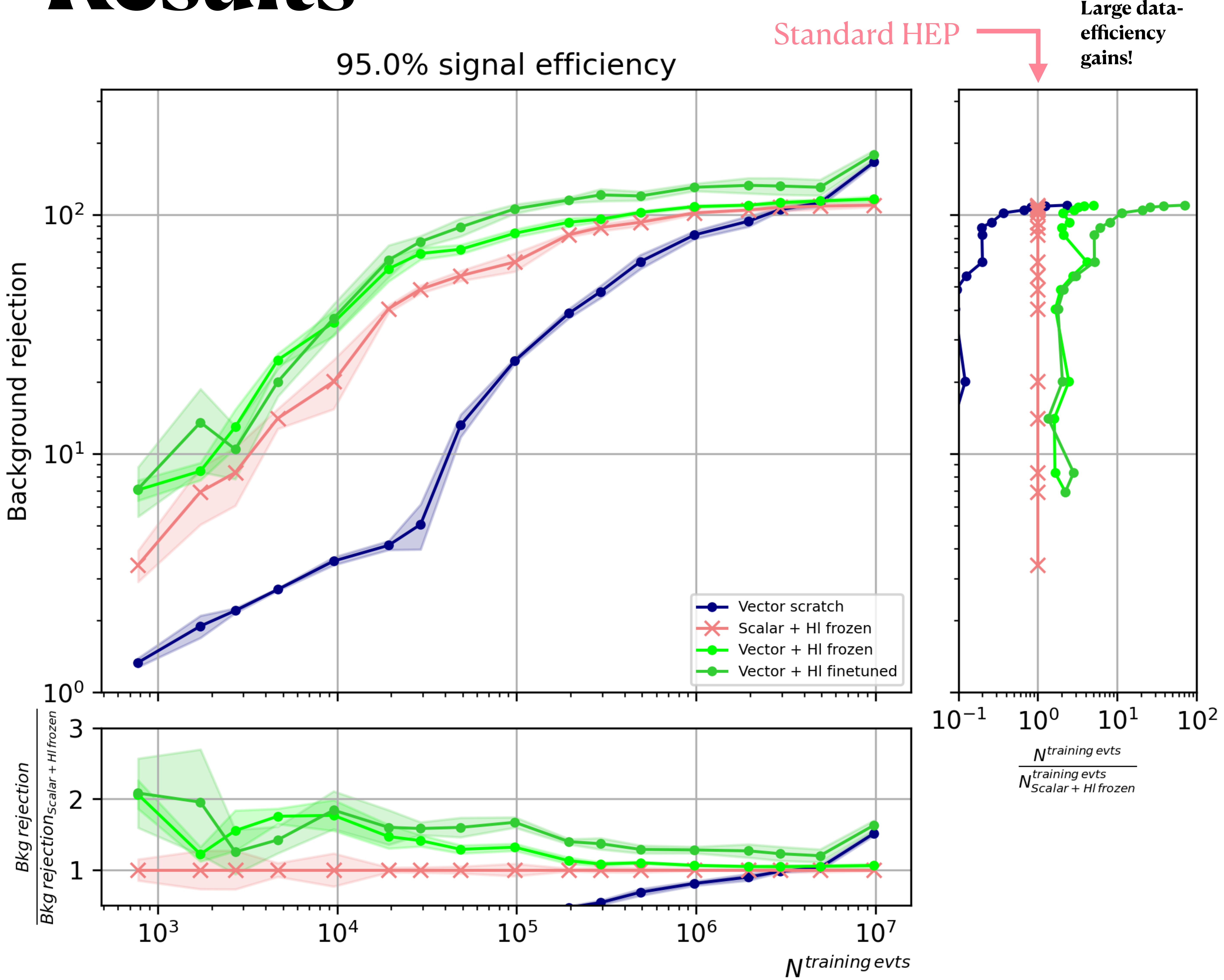
Q: Is fine-tuning as in modern ML worth it?

Q: Do we see benefits of scale & adjacent pre-training tasks?

Results

Well-known patterns from ML seem to hold also in HEP

- Fine-tuning for Analysis extracts more info than just pre-trained features
- Fine-tuning workflow helps in both **performance** & **data efficiency** (10-100x wrt standard hep)
- Higher-dim embeddings also seem to be useful

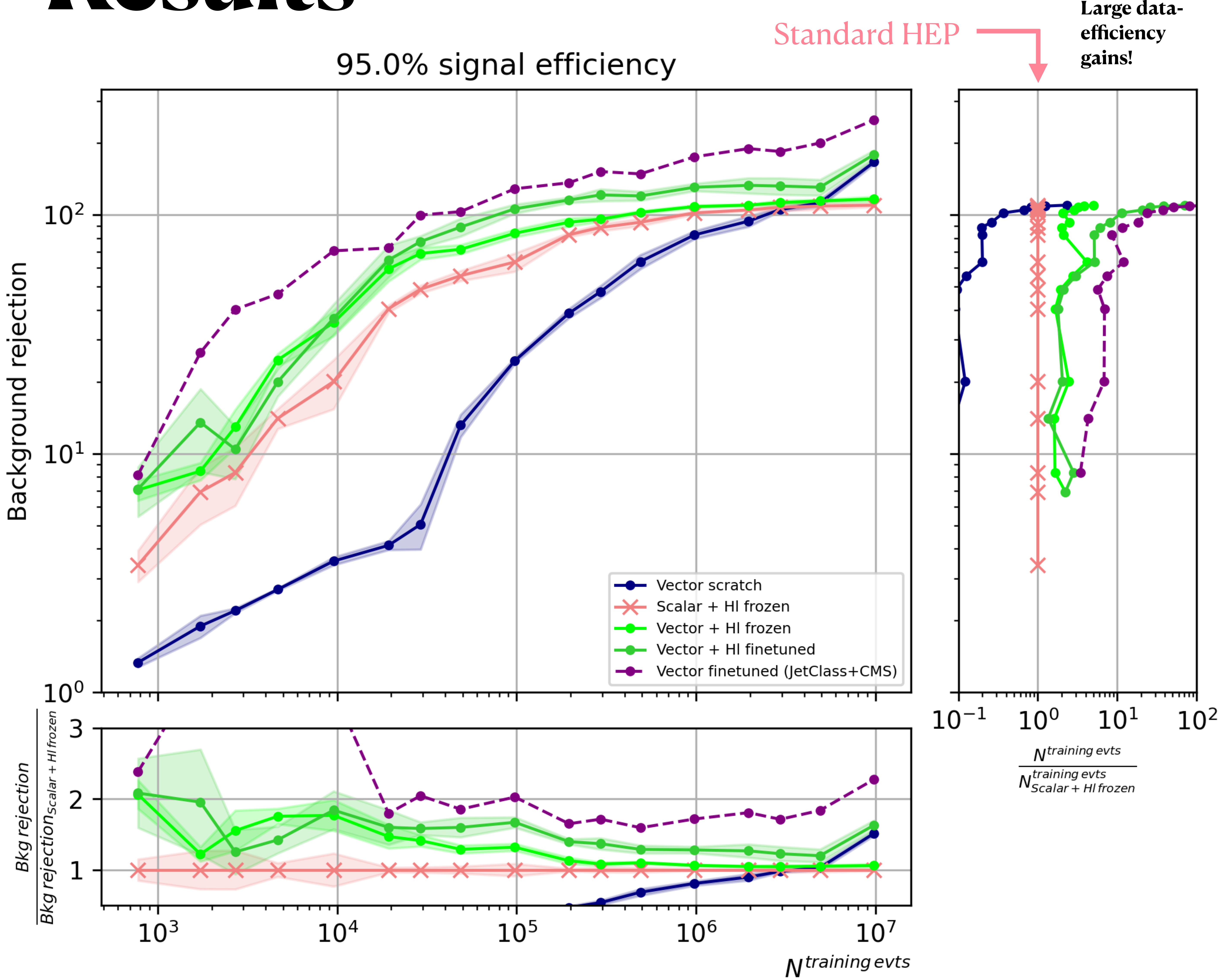


Results

Well-known patterns from ML seem to hold also in HEP

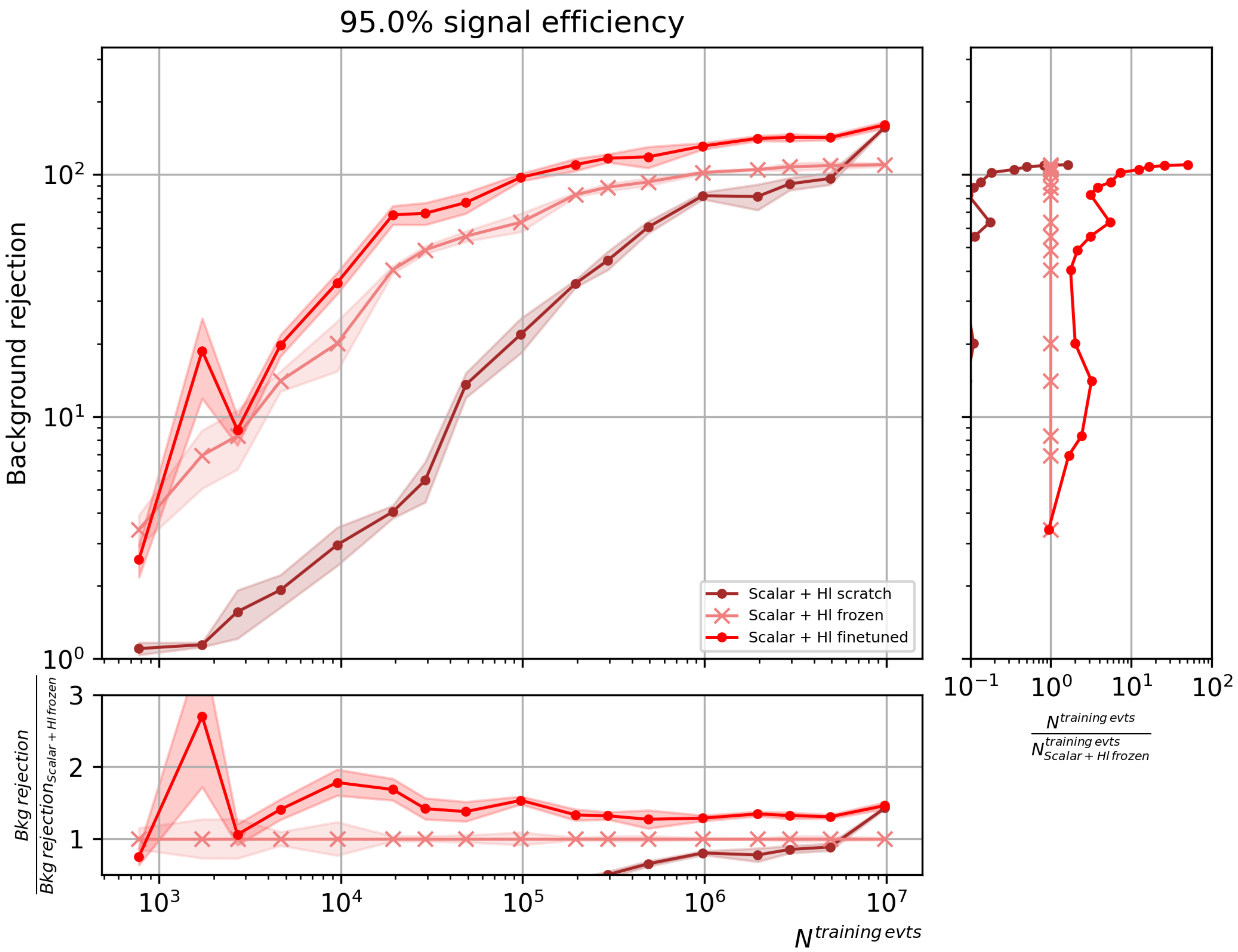
- More pre-training helps

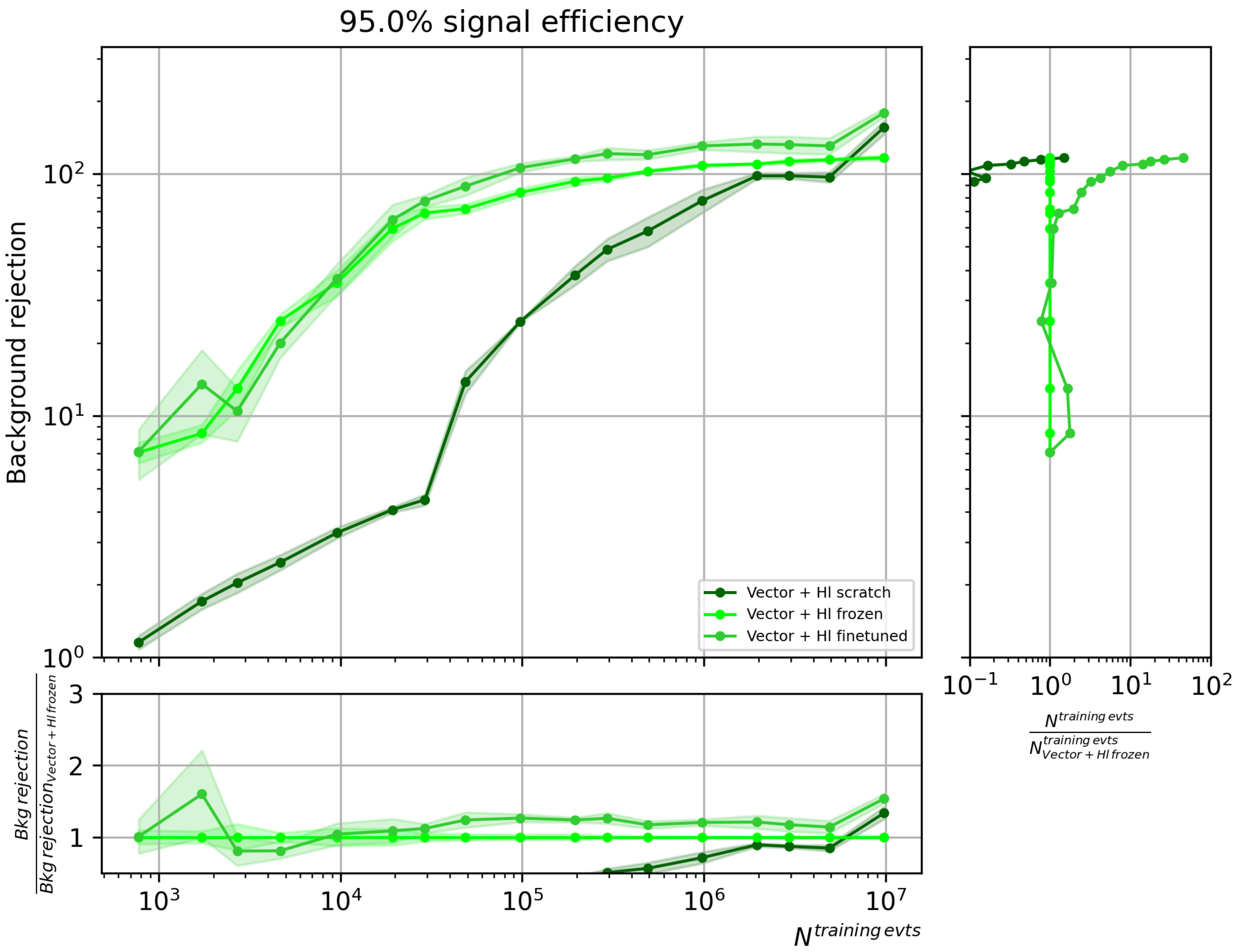
ParT backbone also pre-trained on a different dataset: JetClass (10 jet labels)



Results

From scratch training also works, it’s just slow





Conclusions

Fine-tuning workflow for end to end analysis works and is useful even for simple examples

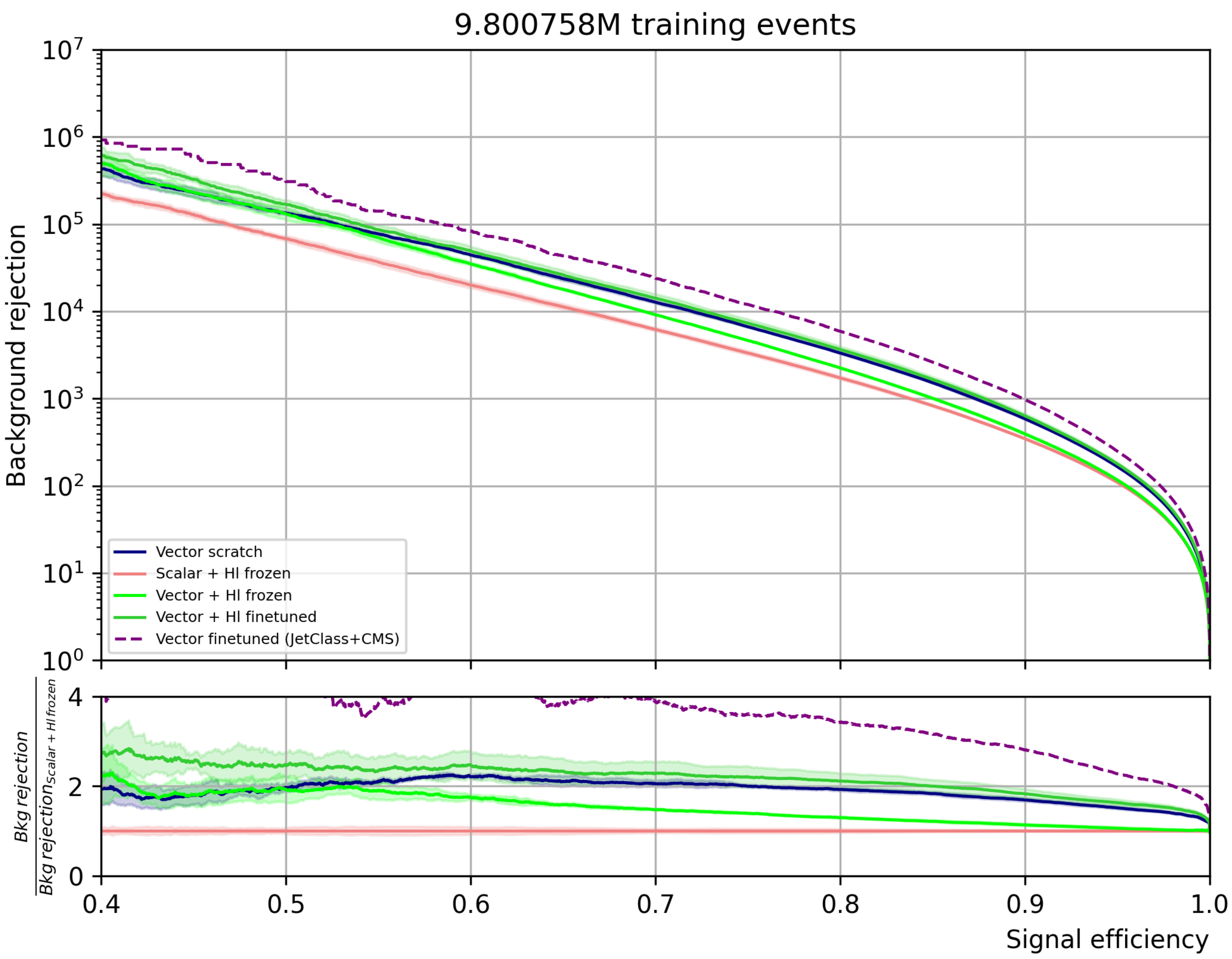
- **Gains in both data efficiency & performance** wrt standard HEP
    - 2x in background rejection
    - 10-100x in data efficiency
- There might be more to gain in complex topologies

Q: What's the best pre-training task?

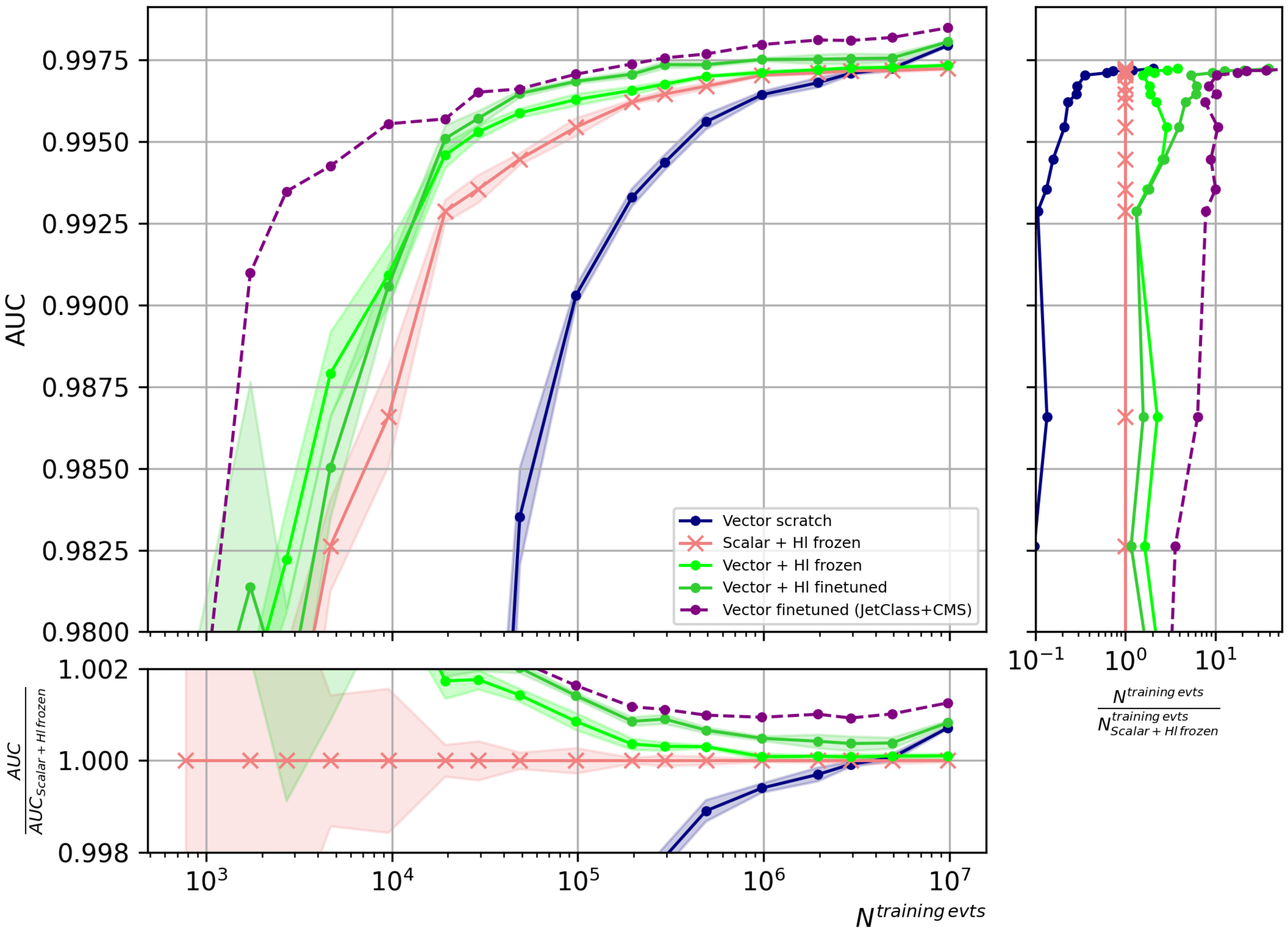
Q: How do we calibrate high-dim representation?

Thank You!

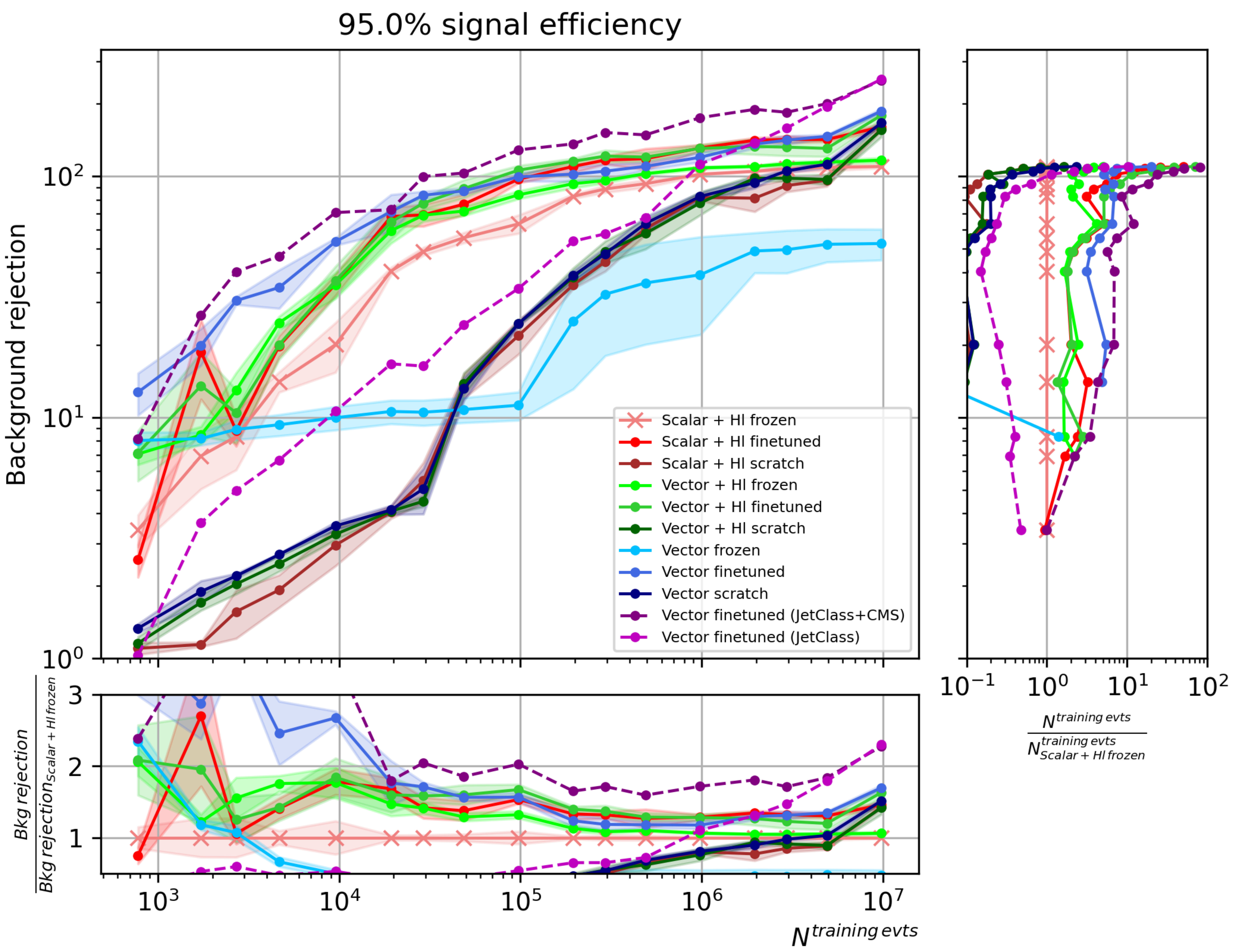
Backup



9.800758M training events
Background rejection
Vector scratch
Scalar + Hl frozen
Vector + Hl frozen
Vector + Hl finetuned
Vector finetuned (JetClass+CMS)
Bkg rejection / Bkg rejection Scalar + Hl frozen
Signal efficiency

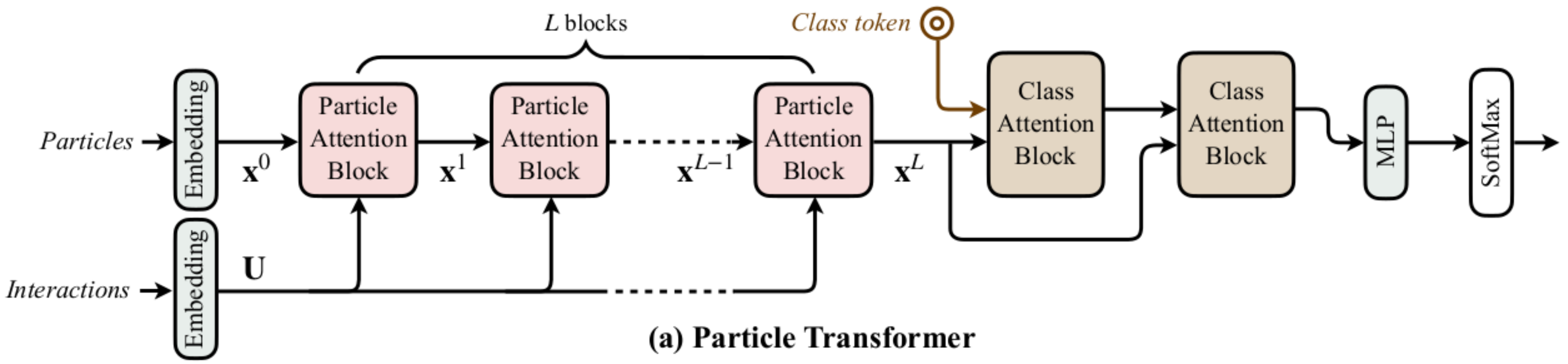


AUC
0.9975
0.9950
0.9925
0.9900
0.9875
0.9850
0.9825
0.9800
Vector scratch
Scalar + Hl frozen
Vector + Hl frozen
Vector + Hl finetuned
Vector finetuned (JetClass+CMS)
$\frac{AUC}{AUC_{Scalar + Hl\,frozen}}$
1.002
1.000
0.998
10^3 10^4 10^5 10^6 10^7
$N^{training\,evts}$
10^{-1} 10^0 10^1
$\frac{N^{training\,evts}}{N^{training\,evts}_{Scalar + Hl\,frozen}}$

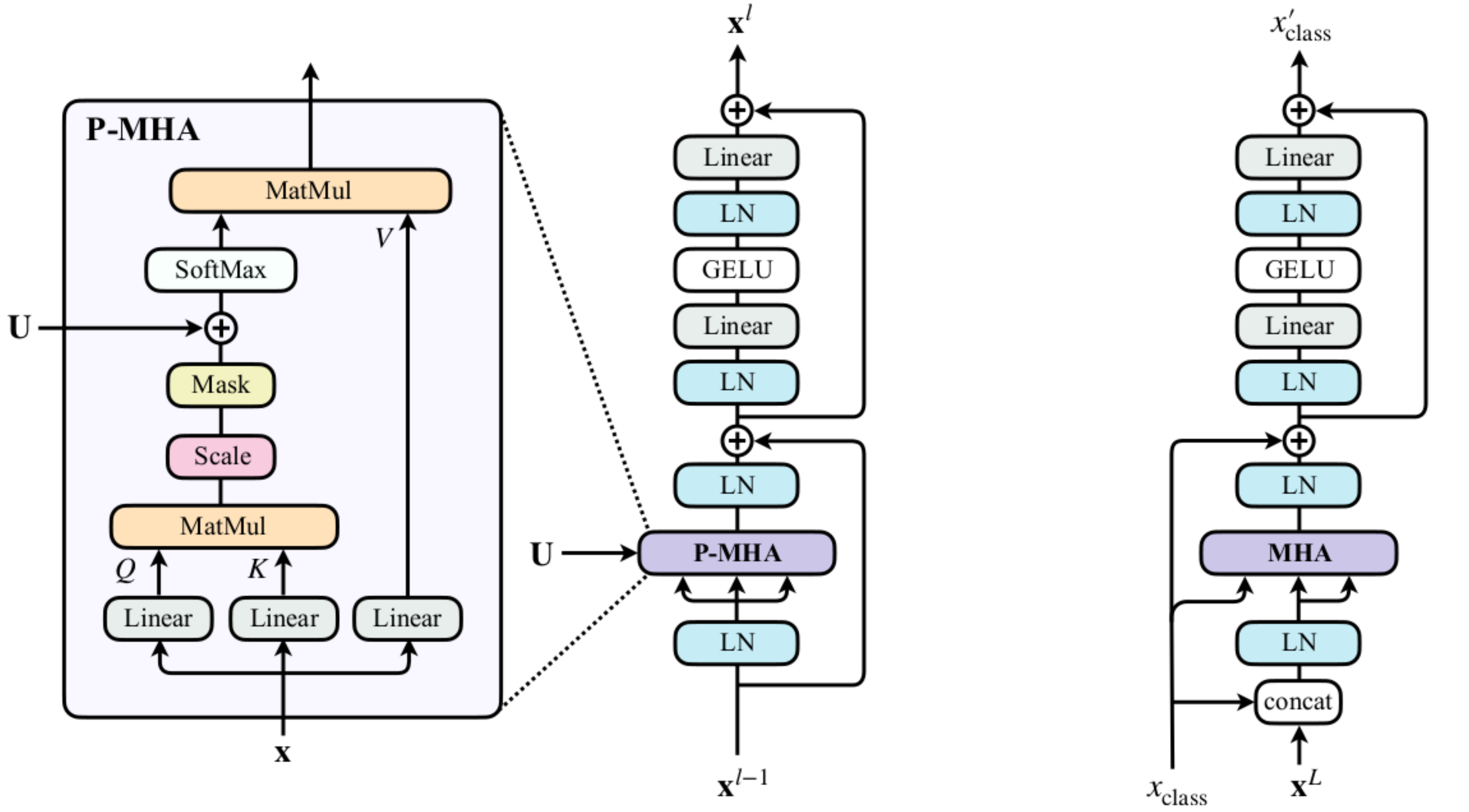


95.0% signal efficiency
Background rejection
Scalar + Hl frozen
Scalar + Hl finetuned
Scalar + Hl scratch
Vector + Hl frozen
Vector + Hl finetuned
Vector + Hl scratch
Vector frozen
Vector finetuned
Vector scratch
Vector finetuned (JetClass+CMS)
Vector finetuned (JetClass)
Bkg rejection / Bkg rejection_Scalar + Hl frozen
N^training evts
N^training evts / N^training evts_Scalar + Hl frozen

ParT



(a) Particle Transformer



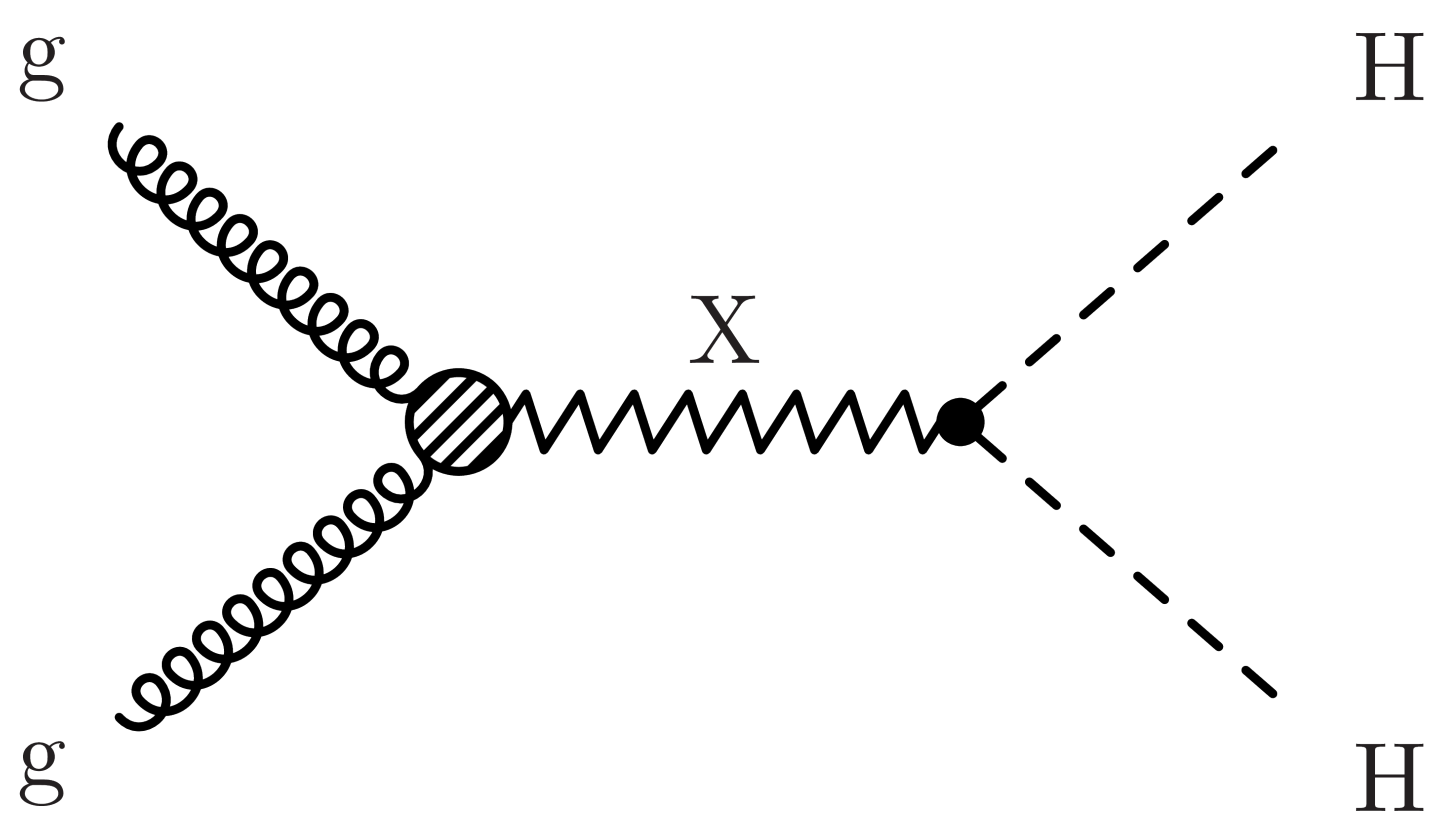
(b) Particle Attention Block

(c) Class Attention Block

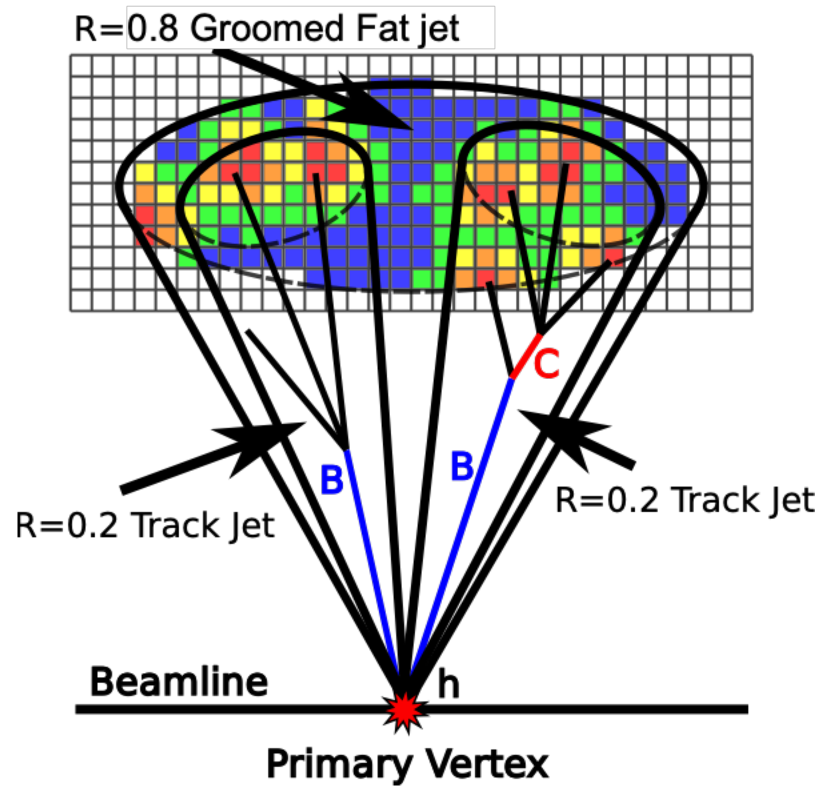
[arXiv:2202.03772]

CMS open data

- CMS simulated dataset:
- Sample with jet, track and secondary vertex properties for H(bb) tagging (http://opendata.cern.ch/record/12102)
- meant for jet tagging, up to 100 pf cand per jet - 17 feats each
- signal samples: 11 mass points - M_x from 600 GeV to 4500 GeV, bkg: QCD multijet
- ’fat jets’ (fj) 4-momenta and (old) Xbb score



10M events / 22M jets




```
    - ['pfcand_pt_log', null]
    - ['pfcand_e_log', null]
    - ['pfcand_logptrel', null]
    - ['pfcand_logerel', null]
    - ['pfcand_deltaR', null]
    - ['pfcand_charge', null]
    - ['pfcand_isChargedHad', null]
    - ['pfcand_isNeutralHad', null]
    - ['pfcand_isGamma', null]
    - ['pfcand_isEl', null]
    - ['pfcand_isMu', null]
    - ['pfcand_dz', null]
    - ['pfcand_dzerr', null]
    - ['pfcand_dz', null]
    - ['pfcand_dzerr', null]
    - ['pfcand_deta', null]
    - ['pfcand_dphi', null]
pf_vectors:
  length: 110
  pad_mode: wrap
  vars:
    - [pfcand_px, null]
    - [pfcand_py, null]
    - [pfcand_pz, null]
    - [pfcand_energy, null]
```


[http://cms-results.web.cern.ch/cms-results/public-results/publications/BTV-16-002/]