Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CLUSTER OF EXCELLENCE
QUANTUM UNIVERSE

DESY.

anuranjan.sarkar@desy.de

To Mask, Predict, or Classify?

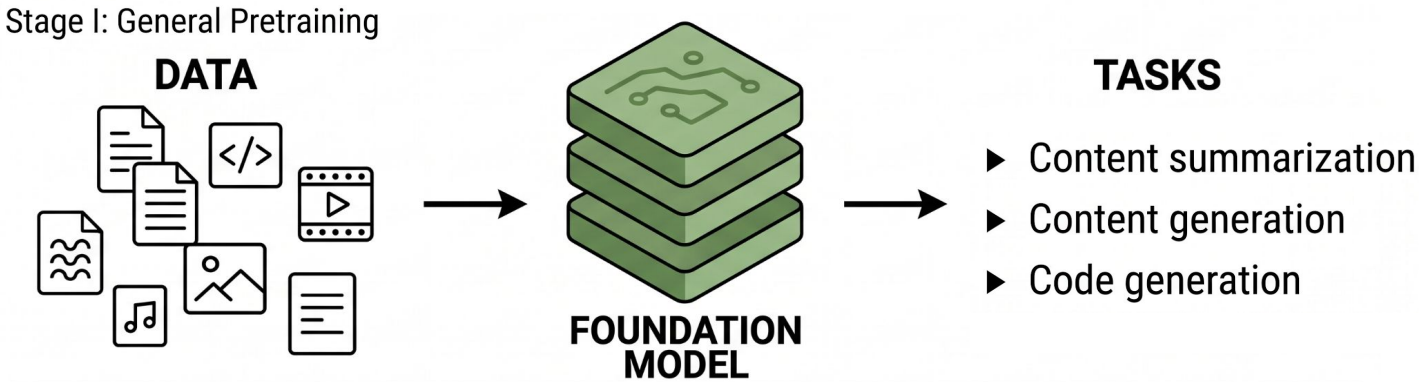
Comparative Analysis of Pre-Training Strategies for Jet Foundation Models

Anuranjan Sarkar, Joschka Birk, Anna Hallin, Sarah Heim, Gregor Kasieczka

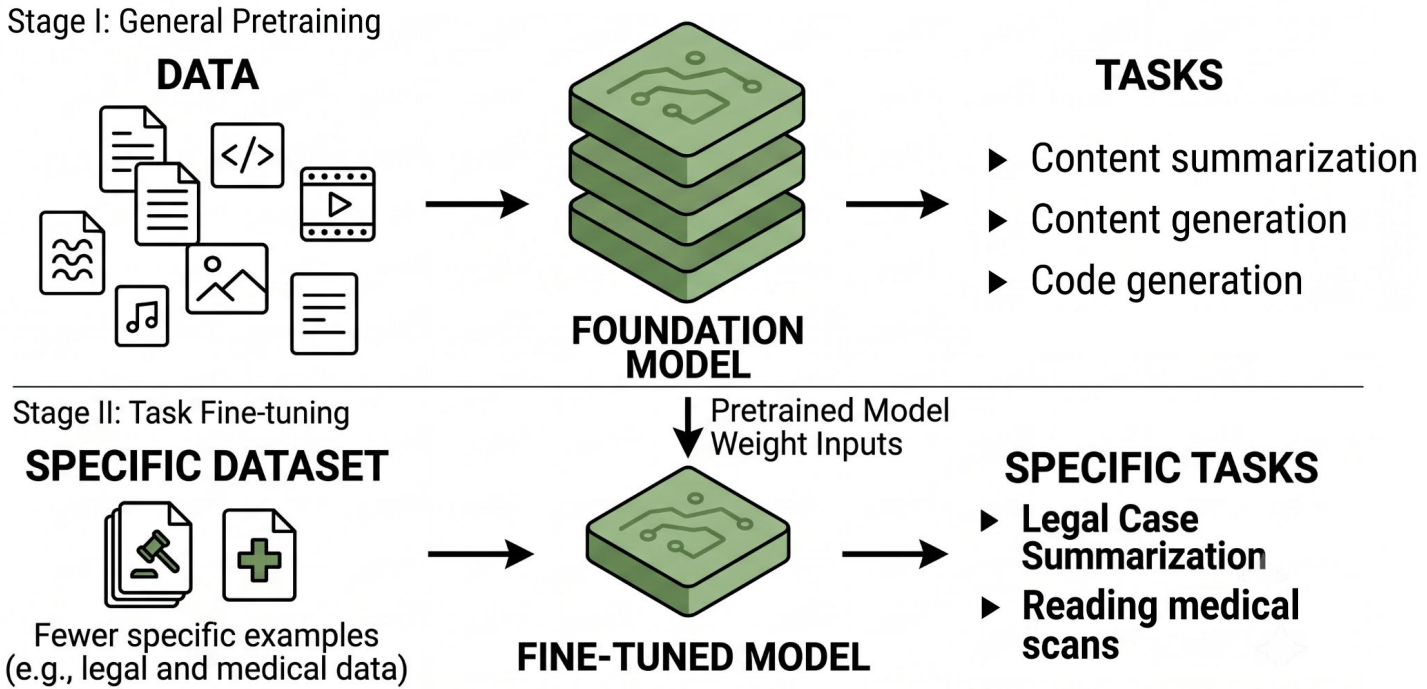
DESY · Universität Hamburg · Cluster of Excellence Quantum Universe

ML4Jets 2026 · 16 September 2026

What are Foundation Models? Pre-training?

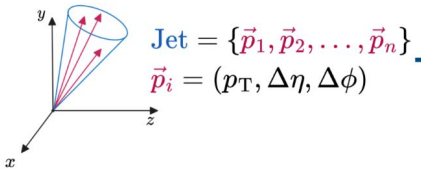


What are Foundation Models? Pre-training?



Jet Foundation Model: OmniJet-α

OmniJet-α [1] uses **next token prediction as a pre-training task** for a jet tagger

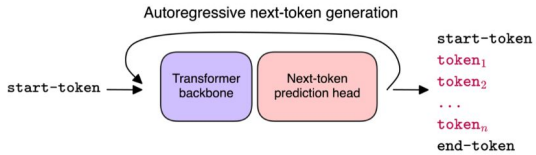


$$\text{Jet} = \{\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_n\}$$
$$\vec{p}_i = (p_\text{T}, \Delta\eta, \Delta\phi)$$

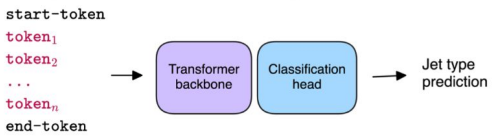
Constituents are tokenized with a VQ-VAE [2-4]

$$\text{Jet} = \{\texttt{start-token}, \texttt{token}_1, \ldots, \texttt{token}_n, \texttt{end-token}\}$$
$$\texttt{token}_i = \text{integer value} \in [1, \ldots, 8192]$$

Self-supervised pre-training of transformer backbone on generative task (next token prediction)



Fine-tuning to classification task:
Swap model head and use pre-trained backbone



Three stages

1 Tokenize
constituents → discrete tokens with a VQ-VAE

2 Pre-train
transformer backbone, next-token prediction(NTP/Gen)

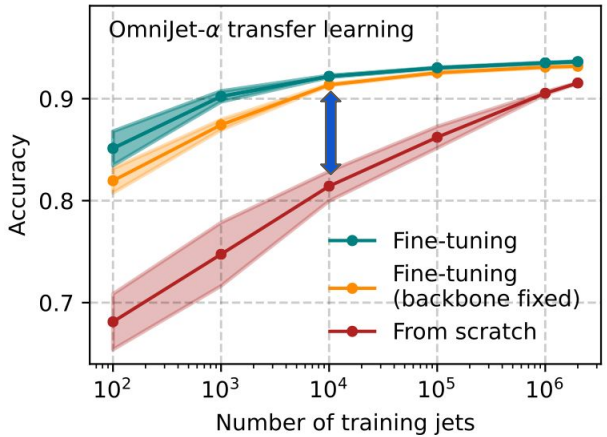
3 Fine-tune
swap the head, keep the backbone

[1] Birk et al. (2024). arXiv:2403.05618 [3] Huh et al. (2023). arXiv:2305.08842
[2] Oord et al. (2018). arXiv:1711.00937 [4] Golling et al. (2024). arXiv:2401.13537

– Slide by Joschka Birk

OmniJet-α: Developments

2024



OmniJet-α[1]

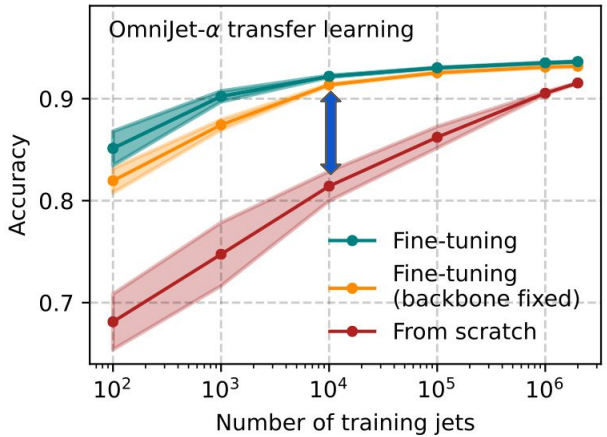
- Tokenized input
- p_T sorted constituents
- Kinematics only

[1] Birk et al. (2024). arXiv:2403.05618

OmniJet-α: Developments

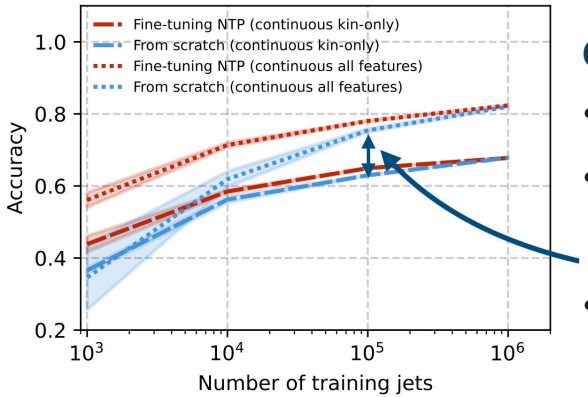
2024

2025



OmniJet-α[1]

- Tokenized input
- p_T sorted constituents
- Kinematics only



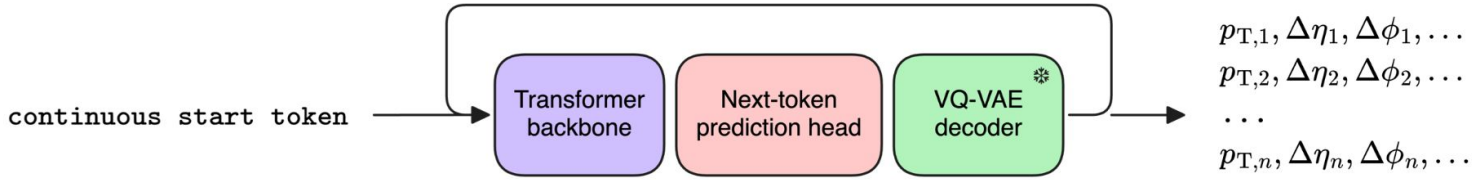
OmniJet-α v2[2]

- Shuffled sequences
- Continuous input, token targets + particle-ID, displacement
- Hybrid pre-training

[1] Birk et al. (2024). arXiv:2403.05618
[2] Birk et al. (2025). arXiv:2512.04149

What has changed since: Continuous extended features

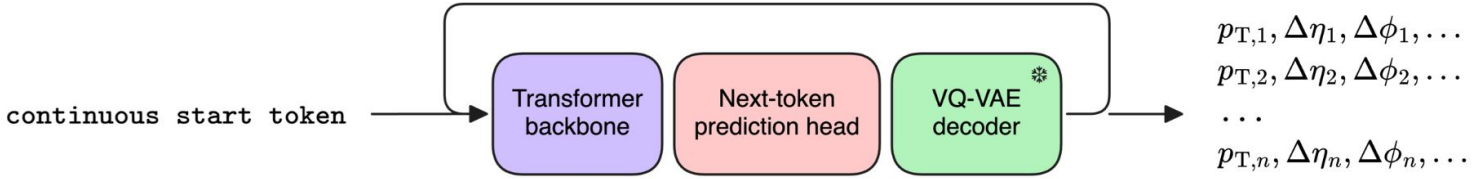
Switching to a **hybrid setup: continuous input, token target**



→ ***Allows for full-resolution continuous input in classification***

What has changed since: Continuous extended features

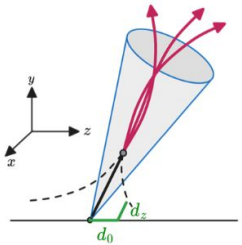
Switching to a **hybrid setup: continuous input, token target**



→ ***Allows for full-resolution continuous input in classification***

Extending the feature set:

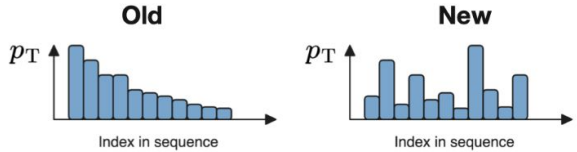
Adding particle-ID and trajectory displacement



- Kinematic features
- **Particle-ID (*new*)**
- **Trajectory displacement (*new*)**

Further changes compared to OmniJet-α:

- Using **shuffled particle sequences**

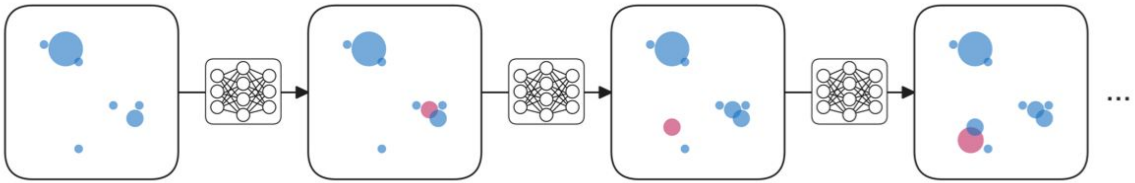


- Transformer-based model heads
- Condition on particle number

What has changed since: Hybrid pre-training objectives

Generation (next particle/token prediction)

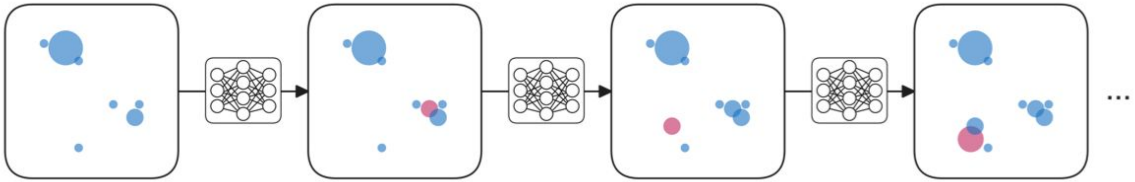
"Given this **sequence of previous particles**, what would you **add next** **to make it a reasonable jet**?"



What has changed since: Hybrid pre-training objectives

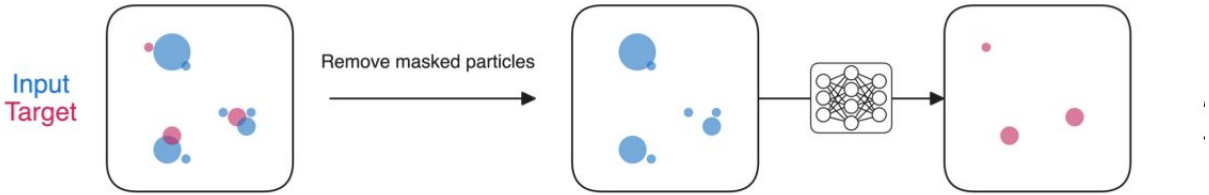
Generation (next particle/token prediction)

"Given this **sequence of previous particles**, what would you **add next** to make it a reasonable jet?"



Masked particle prediction* [1-2] **(MPM)**

"Given **this set of particles**, what would you **fill in at these positions** to make it a reasonable jet?"

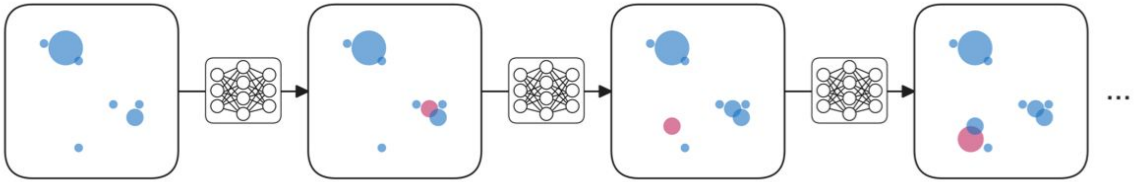


[1] Golling et al. (2024). arXiv:2401.13537
[2] Leigh et al. (2025). arXiv:2409.12589

What has changed since: Hybrid pre-training objectives

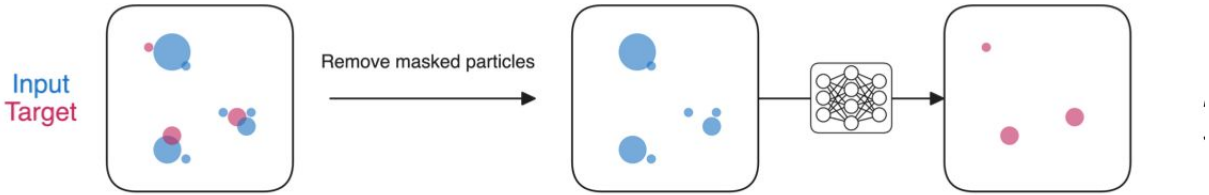
Generation (next particle/token prediction)

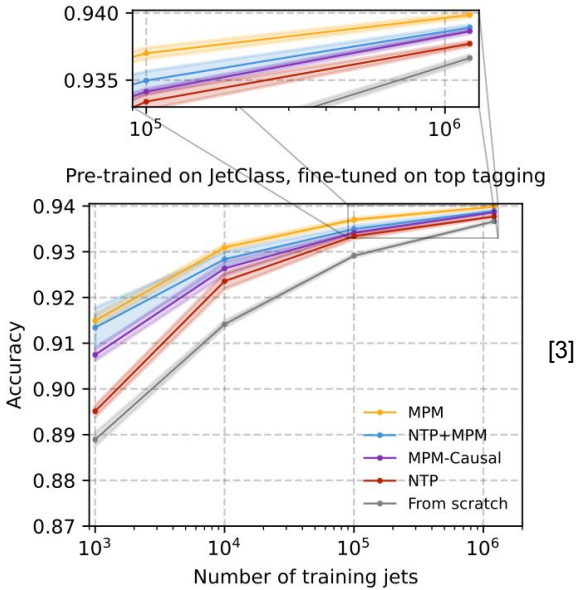
"Given this **sequence of previous particles**, what would you **add next** to make it a reasonable jet?"



Masked particle prediction* [1-2] **(MPM)**

"Given **this set of particles**, what would you **fill in at these positions** to make it a reasonable jet?"





[3]

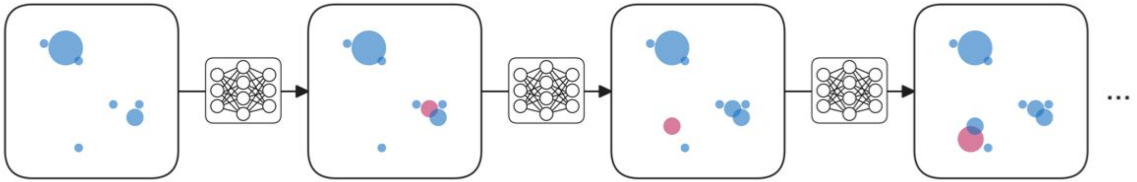
- **Hybrid pre-training Gen/NTP+MPM performs better than NTP**
- **MPM performs best as the pretraining objective to finetune for Top Tagging**

[1] Golling et al. (2024). arXiv:2401.13537
[2] Leigh et al. (2025). arXiv:2409.12589
[3] Birk et al. (2025). arXiv:2512.04149

What has changed since: Hybrid pre-training objectives

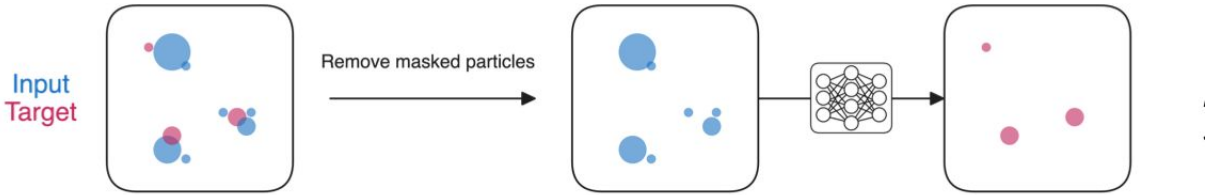
Generation (next particle/token prediction)

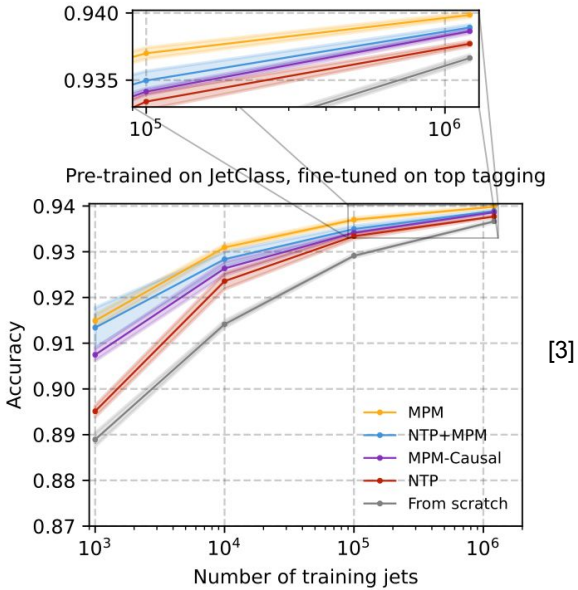
"Given this **sequence of previous particles**, what would you **add next** to make it a reasonable jet?"



Masked particle prediction* [1-2] **(MPM)**

"Given **this set of particles**, what would you **fill in at these positions** to make it a reasonable jet?"





[3]

- **Hybrid pre-training Gen/NTP+MPM performs better than NTP**
- **MPM performs best as the pretraining objective to finetune for Top Tagging**

Q. Since Top tagging is a classification task, does a pretrained class head along with a pretrained backbone help us transfer better?

[1] Golling et al. (2024). arXiv:2401.13537
[2] Leigh et al. (2025). arXiv:2409.12589
[3] Birk et al. (2025). arXiv:2512.04149

Question: Does class head transfer better with finer supervised pretraining(198 labels)?

OmniJet-α uses Masking (MPM) and Next-token prediction(NTP/Gen) which are both self-supervised

Prior work	Pre-training	What is missing
Sophon[1]	188-class supervised, JetClass-II	No self-supervised pretraining
OmniLearned[2]	1B Jets including JetClass-I+II jointly	Always uses Class+Gen pretraining
OmniLearned Pre-training study[3]	Similar combinations of pre-training study	JetClass-I (only 10 labels)

[1] Li et al. (2024). arXiv:2405.12972
[2] Bhimji et al. (2026). arXiv:2510.24066
[3] Elsharkawy et al. (2026). arXiv:2606.14870

Question: Does class head transfer better with finer supervised pretraining(198 labels)?

OmniJet-α uses Masking (MPM) and Next-token prediction(NTP/Gen) which are both self-supervised

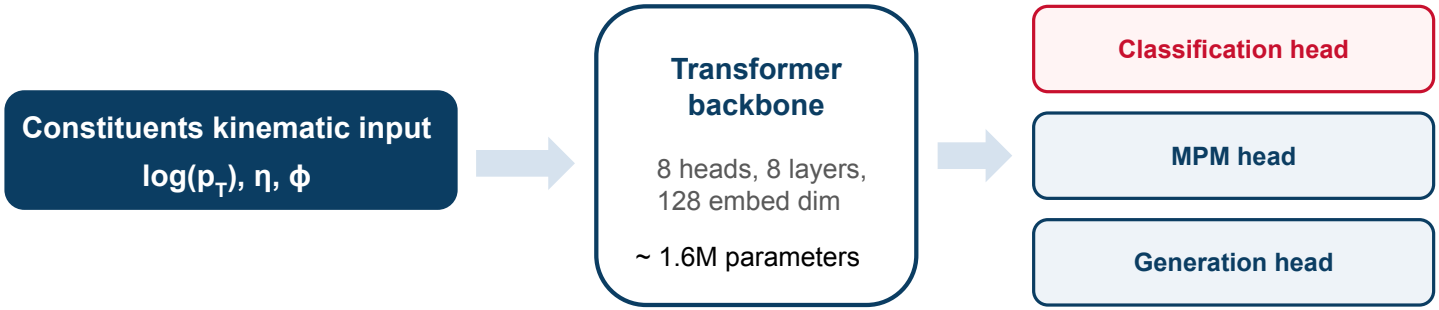
Prior work	Pre-training	What is missing
Sophon[1]	188-class supervised, JetClass-II	No self-supervised pretraining
OmniLearned[2]	1B Jets including JetClass-I+II jointly	Always uses Class+Gen pretraining
OmniLearned Pre-training study[3]	Similar combinations of pre-training study	JetClass-I (only 10 labels)

This work

- Incorporating **classification pretraining** along with the self-supervised pretrainings in OmniJet-α.
- Improving our framework to use **multiple datasets** with **different feature inputs** for pretraining and finetuning.
- Pre-trained on JetClass-I(10 labels) and JetClass-II(188 labels), whose **198 labels are far finer-grained**.
- How does **pretraining on fine-grained labels** for classification transfer to **Top tagging** and **Flavor tagging**?

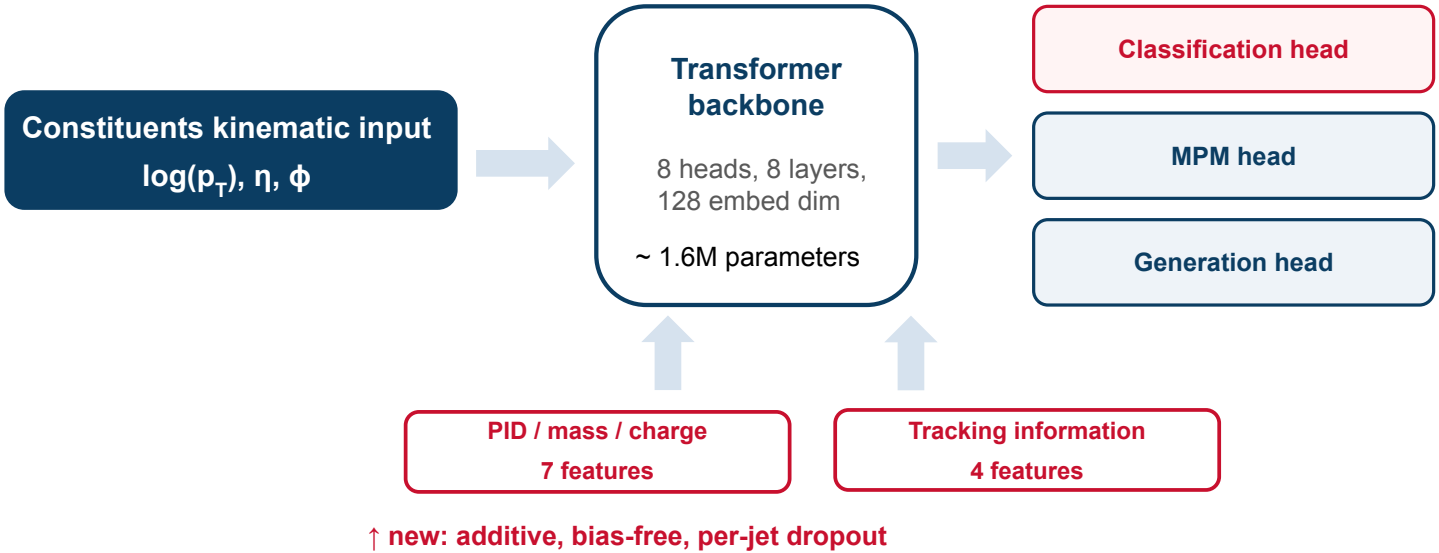
[1] Li et al. (2024). arXiv:2405.12972
[2] Bhimji et al. (2026). arXiv:2510.24066
[3] Elsharkawy et al. (2026). arXiv:2606.14870

Improvements to OmniJet-α: One backbone, All Datasets



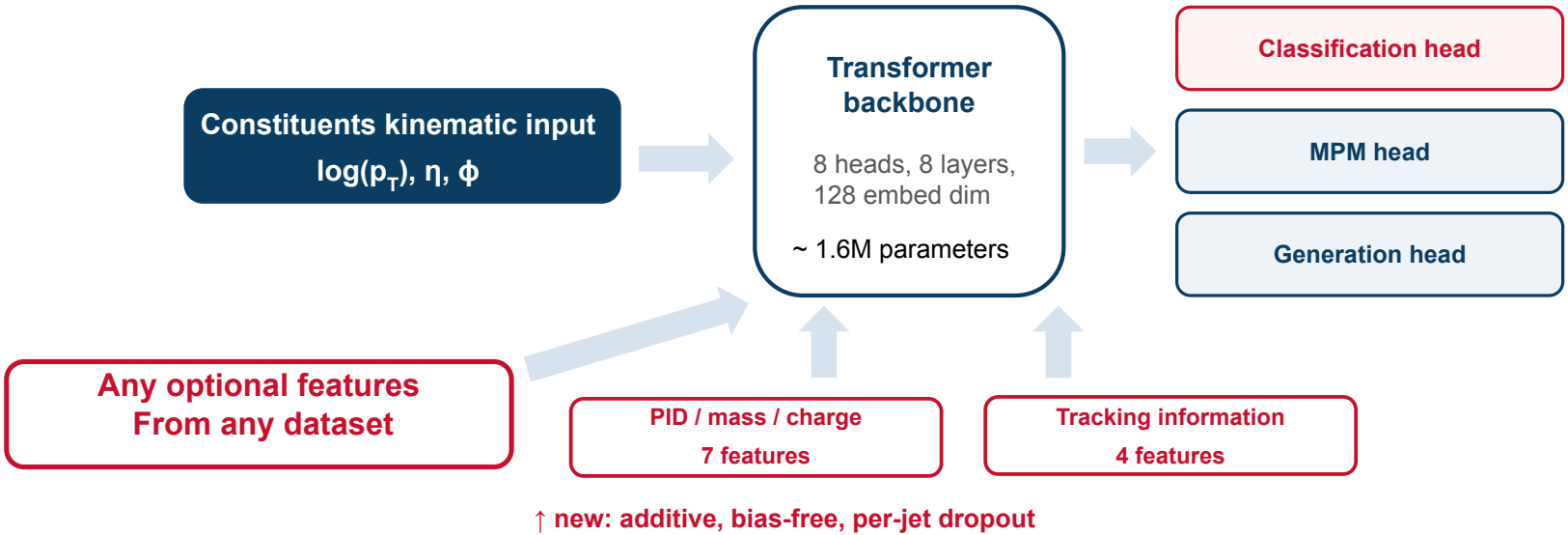
- **Backbone input** is only the **3 kinematic features** every dataset has.

Improvements to OmniJet-α: One backbone, All Datasets



- **Backbone input** is only the **3 kinematic features** every dataset has.
- Optional groups enter through **bias-free projections** that are added, an absent group contributes exactly zero
- Trained with **dropout** on those branches, so the backbone is **robust to datasets lacking in the PID/track information**.

Improvements to OmniJet-α: One backbone, All Datasets



- **Backbone input** is only the **3 kinematic features** every dataset has.
- Optional groups enter through **bias-free projections** that are added, an absent group contributes exactly zero
- Trained with **dropout** on those branches, so the backbone is **robust to datasets lacking in the PID/track information**.

One pre-trained model, any dataset with kinematic features, no shape change and no zero-padding

Pre-training setup

234M	10+188=198	65,536	~2.0M
jets pre-trained on	joint classes	VQ-VAE tokenizer codes	total parameters

Two pre-training datasets

JetClass-I

100M jets · 10 classes
5M jets · Validation
20M jets · Test

JetClass-II

134M jets · 188 classes
33.5M jets · Validation
33.5M jets · Test

- 4 Pre-training arms: **Class, Class+MPM, Class+MPM+Gen, Class+Gen**
- **Extended features** reach the backbone through the **optional branches**, never through the tokenizer.
- **Classification** head gets **raw continuous** inputs instead of tokenized inputs to **prevent loss of information**.
- All arms compared at 5 passes of the 234M dataset.

Downstream tasks

Task	Jets	Constituents	Classes
Top tagging (Landscape)	1.21M	Kinematics only	2 (Top/QCD)
ATLAS open flavour tagging	5.6M	Charged tracks only	4 (light/c/b/τ)

Feature availability

	Kin	PID	Trk
JetClass-I	●	●	●
JetClass-II	●	●	●
Top tagging	●	○	○
ATLAS FTAG	●	◌	●

Filled = present. The optional branches are what let one backbone absorb the gaps.

Downstream tasks

Task	Jets	Constituents	Classes
Top tagging (Landscape)	1.21M	Kinematics only	2 (Top/QCD)
ATLAS open flavour tagging	5.6M	Charged tracks only	4 (light/c/b/τ)

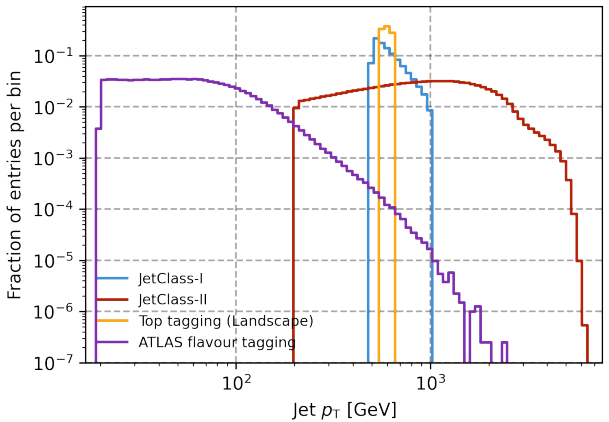
Feature availability

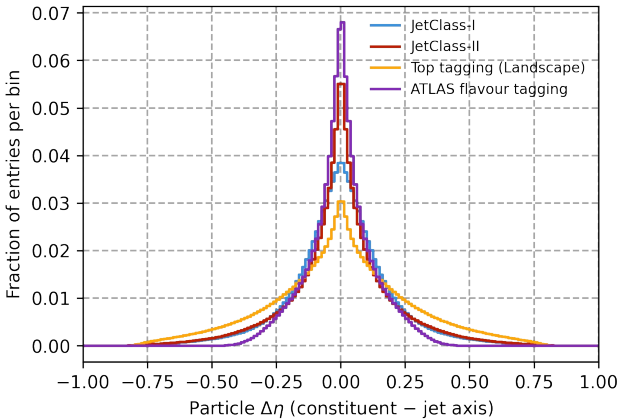
	Kin	PID	Trk
JetClass-I	●	●	●
JetClass-II	●	●	●
Top tagging	●	○	○
ATLAS FTAG	●	◌	●

Filled = present. The optional branches are what let one backbone absorb the gaps.

Both downstream tasks differ from pre-training in other ways

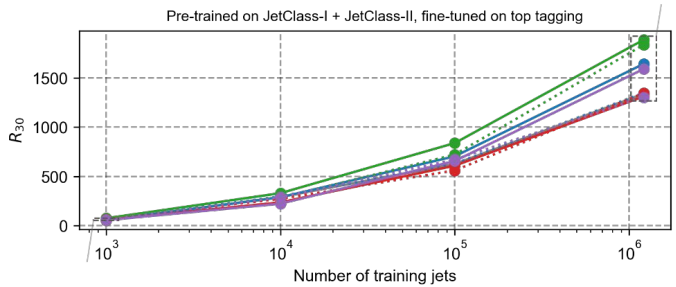
- Detector:
  - JetClass and JetClass-II use **CMS simulation**; R=0.8 for both
  - Landscape(R=0.8) and ATLAS flavour tagging(R=0.4) use **ATLAS simulation**
- Cone size: R = 0.8 in pre-training against R = 0.4 downstream(ATLAS FTag)





Top tagging: Does the pretrained Class head transfer better?

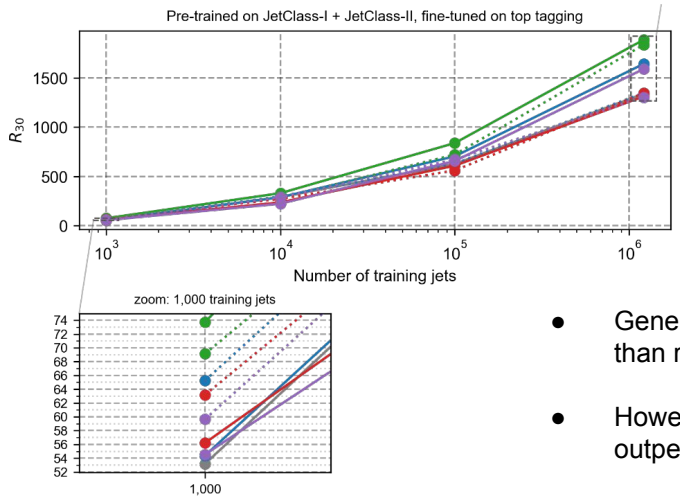
Preliminary plot: Rejection of QCD background at top signal efficiency of 30%



class
class+MPM
class+Gen
class+MPM+Gen
class (fresh head)
class+MPM (fresh head)
class+Gen (fresh head)
class+MPM+Gen (fresh head)
from scratch

Top tagging: Does the pretrained Class head transfer better?

Preliminary plot: Rejection of QCD background at top signal efficiency of 30%

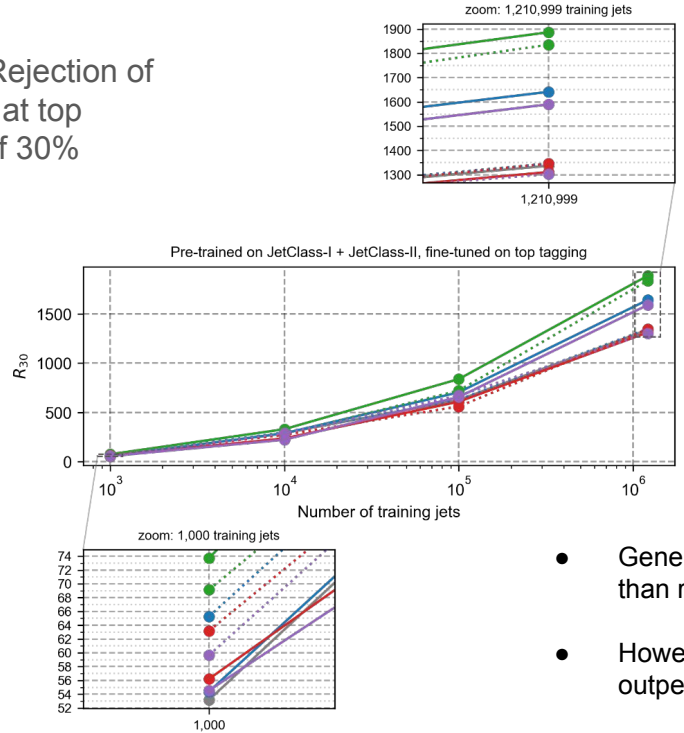


- Generally, **Fresh heads** performs better than reused heads for **1000 jets**.
- However, **Class+MPM** reused head outperforms all other arms.

class
class+MPM
class+Gen
class+MPM+Gen
class (fresh head)
class+MPM (fresh head)
class+Gen (fresh head)
class+MPM+Gen (fresh head)
from scratch

Top tagging: Does the pretrained Class head transfer better?

Preliminary plot: Rejection of QCD background at top signal efficiency of 30%



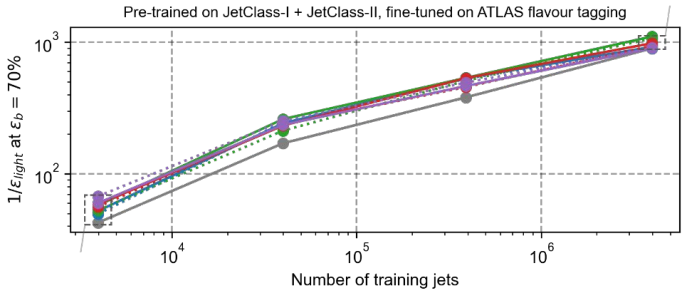
- **Reused heads** outperform their fresh heads counterparts at **1.2M jets**.
- **Class+MPM** reused head outperforms all other arms at 1.2M jets also.

- Generally, **Fresh heads** performs better than reused heads for **1000 jets**.
- However, **Class+MPM** reused head outperforms all other arms.

class
class+MPM
class+Gen
class+MPM+Gen
class (fresh head)
class+MPM (fresh head)
class+Gen (fresh head)
class+MPM+Gen (fresh head)
from scratch

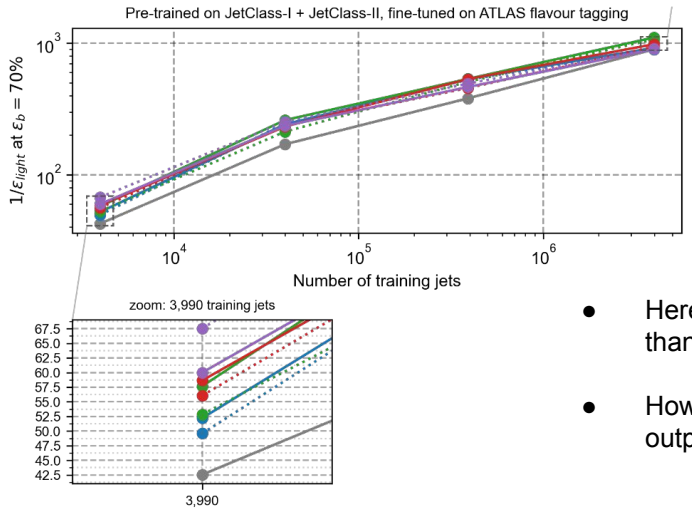
ATLAS flavour tagging: Pretrained class head transfers?

Preliminary plot: Light jet rejection at b-tagging efficiency of 70%



ATLAS flavour tagging: Pretrained class head transfers?

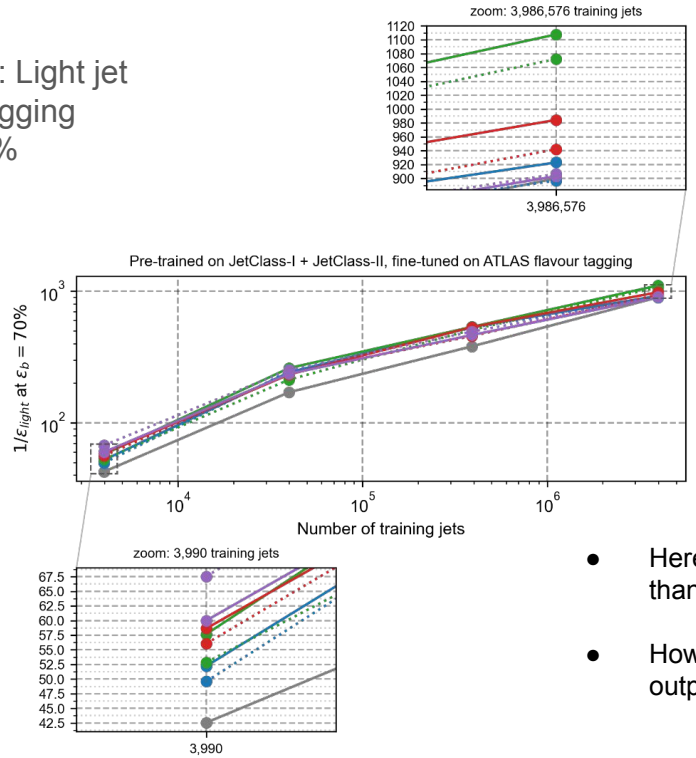
Preliminary plot: Light jet rejection at b-tagging efficiency of 70%



- Here, **reused heads** performs better than fresh heads for **4000 jets**.
- However, **Class+MPM+Gen** fresh head outperforms all other arms at 4000 jets.

ATLAS flavour tagging: Pretrained class head transfers?

Preliminary plot: Light jet rejection at b-tagging efficiency of 70%



- **Reused heads** outperform their fresh heads counterparts at **4M jets**.
- **Class+MPM** reused head outperforms all other arms at 4M jets also.
- Achieved a light jet rejection of **1110 with 4M jets** compared to ATLAS GN2 light jet rejection **1097 trained on 168M jets**.

- Here, **reused heads** performs better than fresh heads for **4000 jets**.
- However, **Class+MPM+Gen** fresh head outperforms all other arms at 4000 jets.

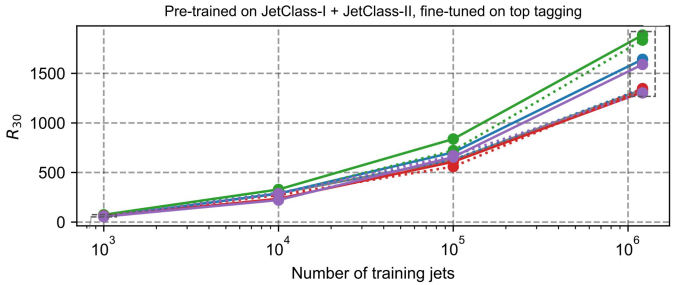
Summary

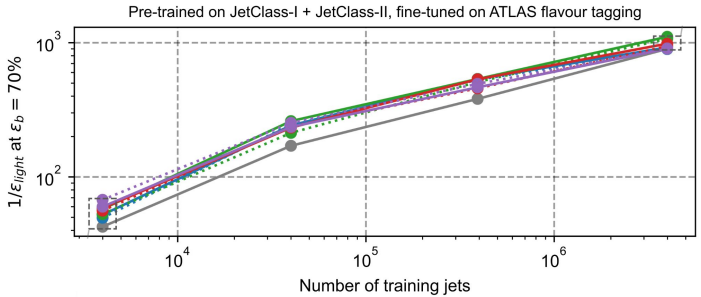
Improvements to OmniJet-α: Optional feature branches makes one backbone dataset-agnostic

Kinematic features trunk plus additive bias-free optional feature branches

Study of supervised classification with self-supervised objectives on fresh and reused classification heads

Class+MPM pretraining with reused class head perform better than other arms for Top tagging and Flavor tagging(comparable to GN2)



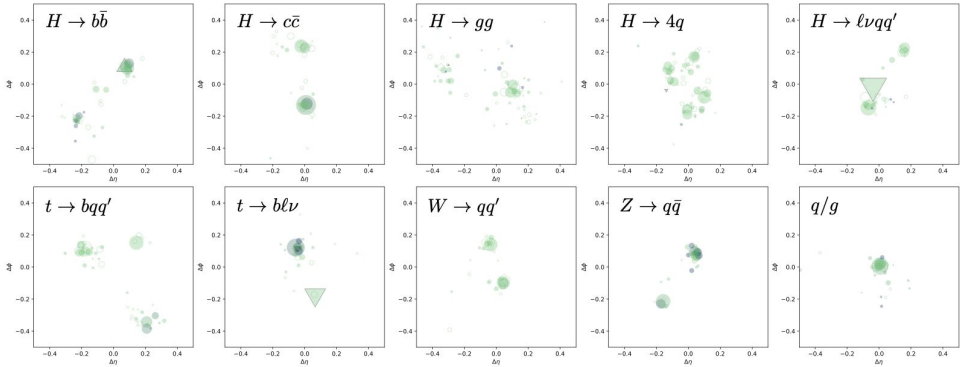


Outlook:

- Scaling the model up from 2M parameters
- One model across detectors and datasets by training on more exhaustive large datasets

Backup: JetClass-I+II labels

JetClass Labels



JetClass-II Labels

Major types	Index range	Label names
Resonant jets: $X \to 2$ prong	0–14	bb, cc, ss, qq, bc, cs, bq, cq, sq, gg, ee, $\mu\mu$, $\tau_h\tau_e$, $\tau_h\tau_\mu$, $\tau_h\tau_h$
Resonant jets: $X \to 3$ or 4 prong	15–160	$bbbb$, $bbcc$, $bbss$, $bbqq$, $bbgg$, $bbee$, $bb\mu\mu$, $bb\tau_h\tau_e$, $bb\tau_h\tau_\mu$, $bb\tau_h\tau_h$, bbb, bbc, bbs, bbq, bbg, bbe, $bb\mu$, $cccc$, $ccss$, $ccqq$, $ccgg$, $ccee$, $cc\mu\mu$, $cc\tau_h\tau_e$, $cc\tau_h\tau_\mu$, $cc\tau_h\tau_h$, ccb, ccc, ccs, ccq, ccg, cce, $cc\mu$, $ssss$, $ssqq$, $ssgg$, $ssee$, $ss\mu\mu$, $ss\tau_h\tau_e$, $ss\tau_h\tau_\mu$, $ss\tau_h\tau_h$, ssb, ssc, sss, ssq, ssg, sse, $ss\mu$, $qqqq$, $qqgg$, $qqee$, $qq\mu\mu$, $qq\tau_h\tau_e$, $qq\tau_h\tau_\mu$, $qq\tau_h\tau_h$, qqb, qqc, qqs, qqq, qqg, qqe, $qq\mu$, $gggg$, $ggee$, $gg\mu\mu$, $gg\tau_h\tau_e$, $gg\tau_h\tau_\mu$, $gg\tau_h\tau_h$, ggb, ggc, ggs, ggq, ggg, gge, $gg\mu$, bee, cee, see, qee, gee, $b\mu\mu$, $c\mu\mu$, $s\mu\mu$, $q\mu\mu$, $g\mu\mu$, $b\tau_h\tau_e$, $c\tau_h\tau_e$, $s\tau_h\tau_e$, $q\tau_h\tau_e$, $g\tau_h\tau_e$, $b\tau_h\tau_\mu$, $c\tau_h\tau_\mu$, $s\tau_h\tau_\mu$, $q\tau_h\tau_\mu$, $g\tau_h\tau_\mu$, $b\tau_h\tau_h$, $c\tau_h\tau_h$, $s\tau_h\tau_h$, $q\tau_h\tau_h$, $g\tau_h\tau_h$, $qqqb$, $qqqc$, $qqqs$, $bbcq$, $ccbs$, $ccbq$, $ccsq$, $sscq$, $qqbc$, $qqbs$, $qqcs$, $bcsq$, bcs, bcq, bsq, csq, $bce\nu$, $cse\nu$, $bqe\nu$, $cqe\nu$, $sqe\nu$, $qqe\nu$, $bc\mu\nu$, $cs\mu\nu$, $bq\mu\nu$, $cq\mu\nu$, $sq\mu\nu$, $qq\mu\nu$, $bc\tau_e\nu$, $cs\tau_e\nu$, $bq\tau_e\nu$, $cq\tau_e\nu$, $sq\tau_e\nu$, $qq\tau_e\nu$, $bc\tau_\mu\nu$, $cs\tau_\mu\nu$, $bq\tau_\mu\nu$, $cq\tau_\mu\nu$, $sq\tau_\mu\nu$, $qq\tau_\mu\nu$, $bc\tau_h\nu$, $cs\tau_h\nu$, $bq\tau_h\nu$, $cq\tau_h\nu$, $sq\tau_h\nu$, $qq\tau_h\nu$
QCD jets	161–187	$bbccss$, $bbccs$, $bbcc$, $bbcss$, $bbcs$, bbc, $bbss$, bbs, bb, $bccss$, $bccs$, bcc, $bcss$, bcs, bc, bss, bs, b, $ccss$, ccs, cc, css, cs, c, ss, s, others

Backup: MPM+NTP

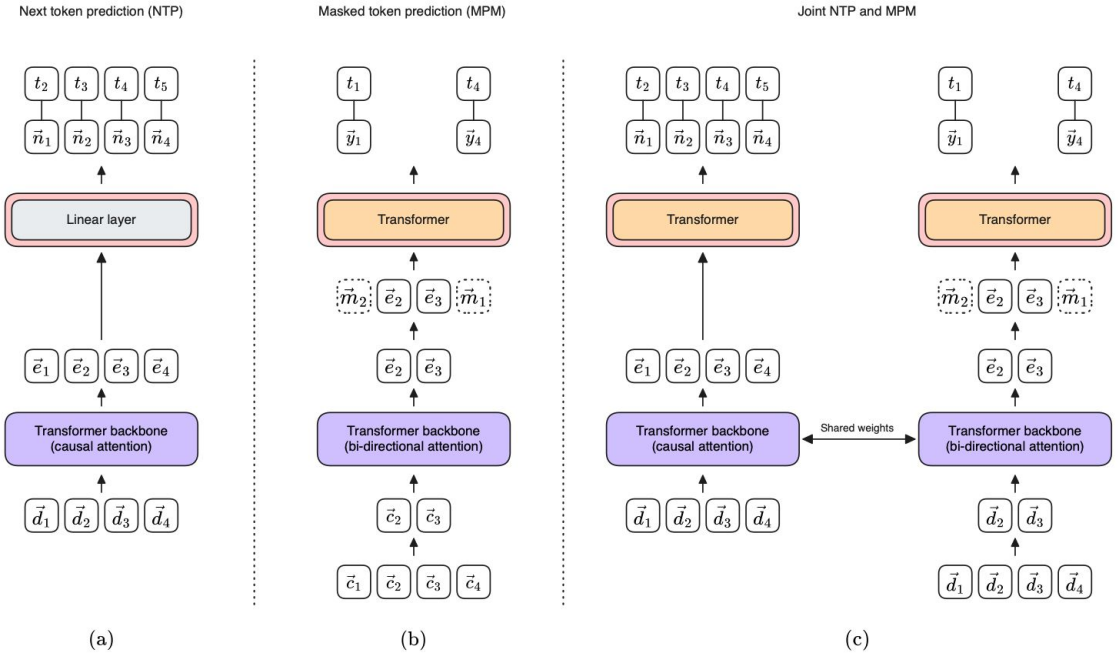


FIG. 2. Schematic overview of the different pre-training strategies: (a) next token prediction (NTP), (b) masked token prediction (MPM) [4, 5], and (c) joint NTP and MPM. Token-IDs are shown as t_i, continuous feature vectors as $\vec{c}_i$, pseudo-continuous feature vectors as $\vec{d}_i$, the mask embeddings as $\vec{m}_i$. The vectors $\vec{e}_i$ represent the backbone output and $\vec{n}_i$ and $\vec{y}_i$ represent the next and masked token predictions, respectively. The particles are not p_T-sorted, but with positional encoding the information of the p_T-order within the masked subset is recovered: in this example, the first masked particle (position 1) is assumed to have smaller p_T than the second masked particle (position 4), which is why they are re-introduced as $\vec{m}_2$ and $\vec{m}_1$ respectively when using positional encoding.

Backup: Transformer Backbone

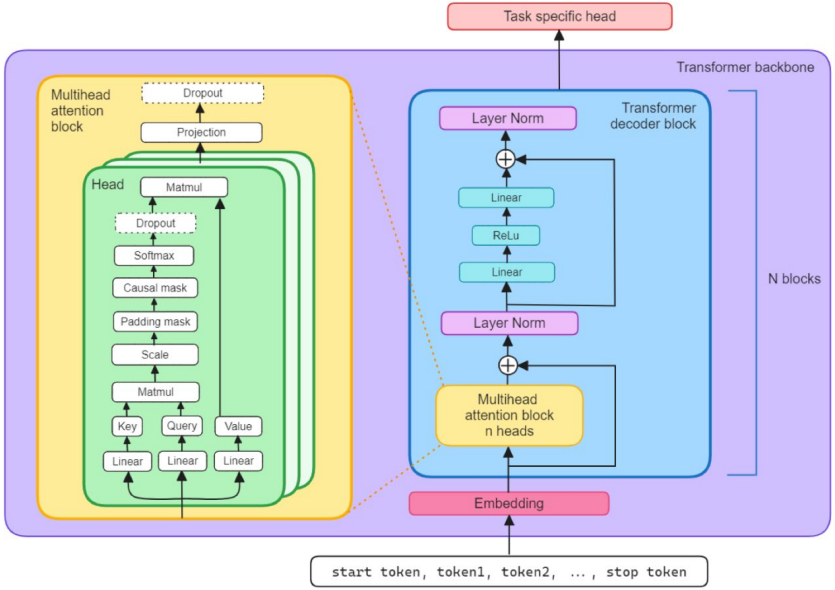


Figure 2: Architecture of the transformer backbone component of OmniJet-α. The data that has been encoded by the VQ-VAE is fed through an embedding layer, before it reaches the main part of the model which is based on the transformer decoder. The output of the transformer decoder blocks is passed to a task specific head, for either generation or classification tasks. Note that during inference of the generative model, the model does not receive complete token sequences, but only the start token. The model will then autoregressively generate the rest of the sequence, updating its input as it progresses, as described in the text.