TA3 Highlights

from WP3

(slides from Anna Hallin)

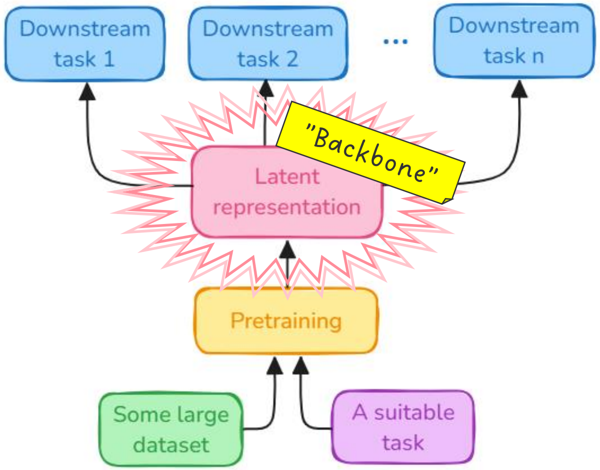
and WP4

(slides from Lorenz Gärtner)

Thomas Kuhr

Foundation models

- Definition:
  - A foundation model is a machine learning model that once **pretrained** can be **finetuned** to different downstream tasks
  - The **performance** of pretraining + finetuning is better than training on the downstream task from scratch
- Large language models (LLMs) like Chat-GPT made foundation models famous, but the concept is not limited to this type of models.
- Foundation models do not need to be based on transformers, although most are.



Why would we want to use them?

- Foundation models may be expensive to train, but once pre-trained, downstream tasks require **less resources**
  - Human resources 👩‍💻
  - Compute resources 🖥️
- Can leverage the pretraining to **boost performance on small datasets**
  - The model learns the general structure of the data during pretraining 📊
  - Can focus on the details during finetuning 🔍
- **Sharing** pre-trained models can provide others with access to resources that are normally not accessible for them (data, computing resources) 🤝

What a particle physics foundation model could look like

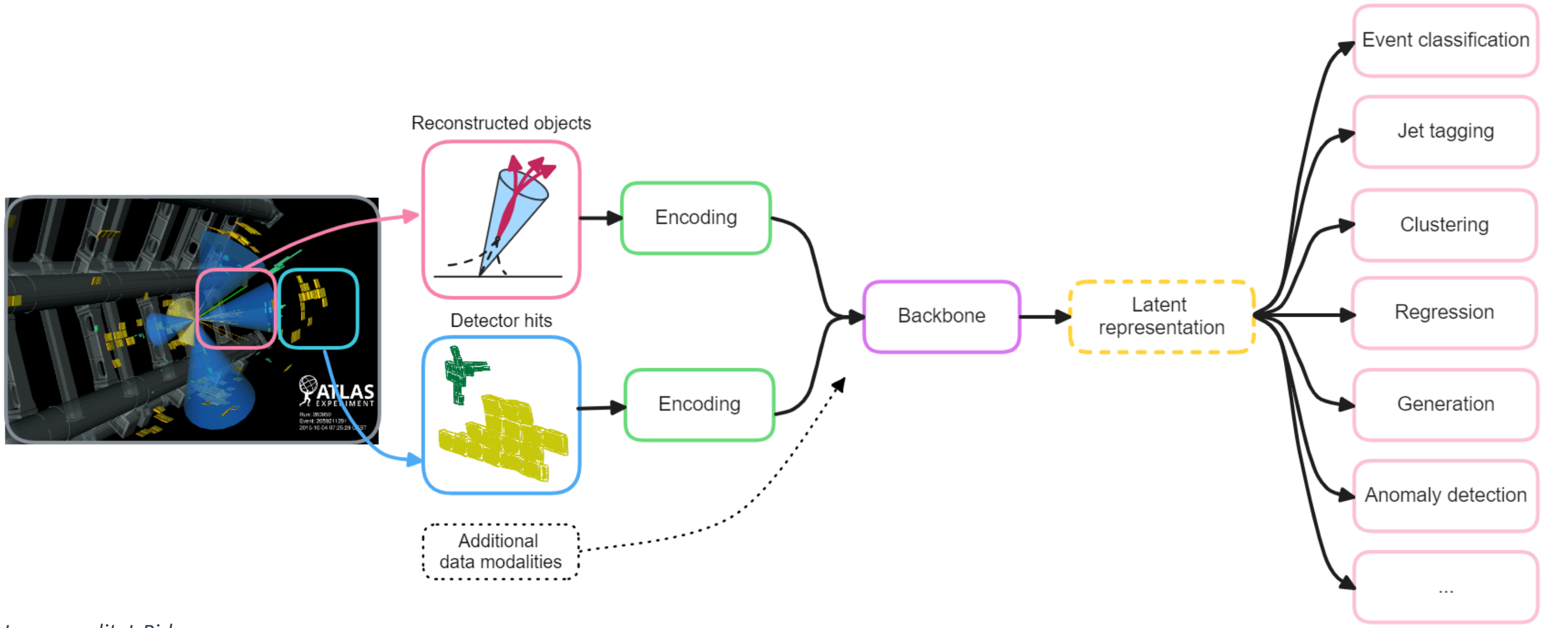


Image credit: J. Birk

A cross-task foundation model for jet physics

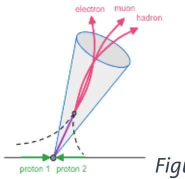


Figure: J. Birk

- **OmniJet-α** (Birk, **AH** et al, [*Mach.Learn.Sci.Tech.* 5 (2024) 3, 035031](); [github]()) was the first foundation model for jet physics that was able to switch tasks: from generation to classification
- **Unsupervised pretraining** on **generation**
  - A model that learns to generate should learn what a jet in general is supposed to look like
  - Unsupervised pretraining means that we **can use data** directly
  - Using low level constituent features only $(p_T, \Delta\eta, \Delta\phi)$
  - Particle features are **tokenized** and jets are represented as a sequence of integers: $p_i = \{p_T, \eta, \phi, \dots\} \rightarrow \text{token}_i$
  - Based on a modified GPT-1 architecture (Radford et al 2018) with **next token prediction** as target: $p(x_j | x_{j-1}, \dots, x_0)$

A cross-task foundation model for jet physics

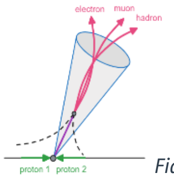
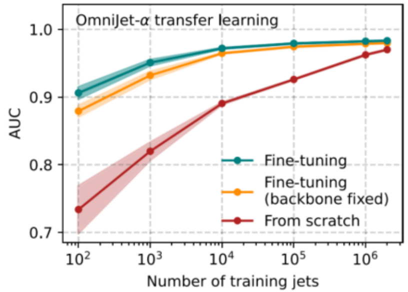
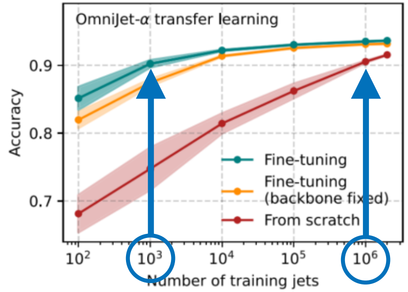


Figure: J. Birk

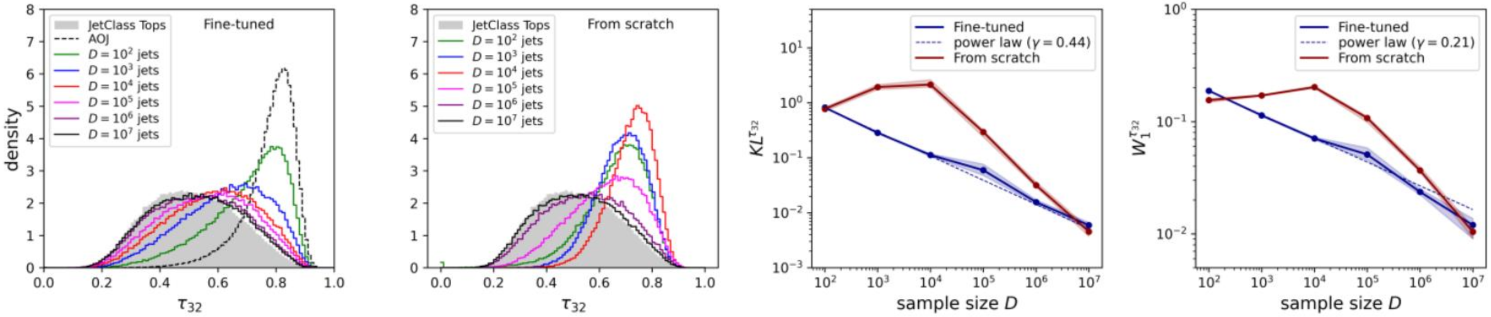
- **OmniJet-α** (Birk, **AH** et al, [*Mach.Learn.Sci.Tech.* 5 (2024) 3, 035031](); [github]()) was the first foundation model for jet physics that was able to switch tasks: from generation to classification
- **Unsupervised pretraining** on **generation**
  - A model that learns to generate should learn what a jet in general is supposed to look like
  - Unsupervised pretraining means that we **can use data** directly
  - Using low level constituent features only $(p_T, \Delta\eta, \Delta\phi)$
  - Particle features are **tokenized** and jets are represented as a sequence of integers
  - Based on a modified GPT-1 architecture (Radford et al 2018) with **next token prediction** as target: $p\left(x_j \middle| x_{j-1}, \dots, x_0\right)$
- **Finetune** on **supervised classification**
  - Demonstrated that the pretrained model **outperformed** the model trained from scratch, in particular on **very small datasets**





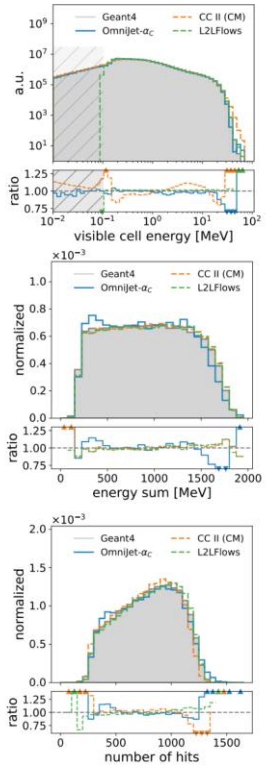
Training on real data

- **Aspen Open Jets** (Amram, **AH** et al, *Mach.Learn.Sci.Tech. 6 (2025) 3, 030601*; github)
  - Derived an **unlabeled** ML-friendly dataset from CMS Open data, containing 180M jets and made it public: fdr.uni-hamburg.de/record/16505
  - Expected to contain mostly QCD jets, and ~10^5 top jets
- **Pretrain** OmniJet-α on this dataset, then **finetune** on generation of hadronically decaying top jets (simulation) → better performance than training from scratch
- Having seen **QCD jets** is apparently **helpful** in order to generate top jets, also (or perhaps particularly) for quantities that are difficult to model, for example the n-subjettiness



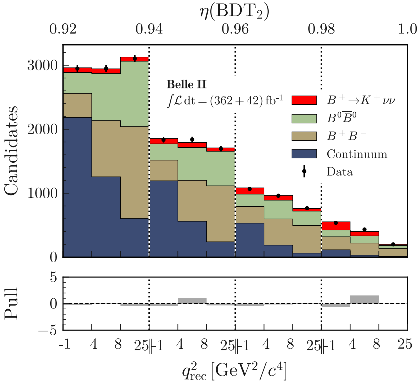
Beyond jets

- Can a foundation model deal with a completely different data type?
- **OmniJet-α_C** (Birk, **AH** et al, *JINST* 20 (2025) 07, P07007; github) applies the OmniJet-α architecture to **point-cloud calorimeter showers**
  - Possible since the model requires no physics knowledge and is **not dependent on any specific type of input** (sequence of integers)
  - No weights from the jet version of OmniJet-α are used: data types are presumably too different for the model to benefit from it
  - Generative training on photon showers shows good results
  - Learns the number of hits independently, no need to condition on it
- By re-using the architecture, we have shown a **hint of translatability**



Belle II has reported
"Evidence for $B^+ \to K^+\nu\bar{\nu}$ decays"

PRD 109.112006



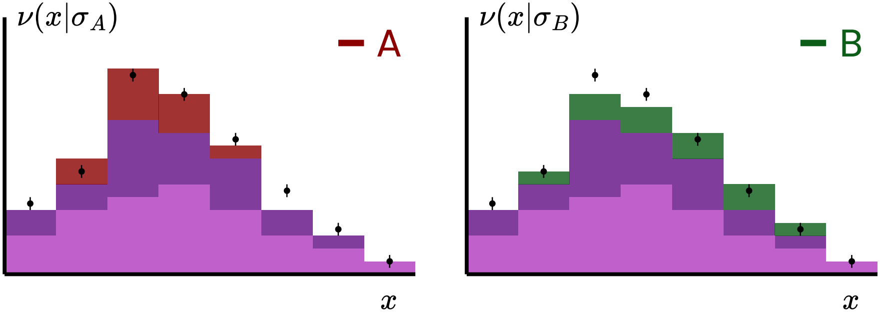
- Fit to kinematic distribution with SM signal scaled by μ
- $\mu = 4.6 \pm 1.0(\text{stat}) \pm 0.9(\text{syst})$
- 2.7σ above SM ($\mu = 1$)

$\rightarrow$ Hint for new physics (NP)?

$\rightarrow$ What kind of NP?

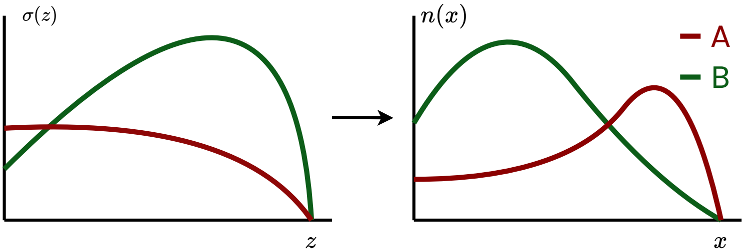
If there is NP kinematic distributions can change

$$p(n|\text{model A}) \neq p(n|\text{model B})$$



Templates from kinematic predictions

$$n(x|\sigma) = \int dz\, L\, \varepsilon(x|z)\, \sigma(z) = \int dz\, n_\sigma(x,z)$$



$z(= q^2)$ – kinematic d.o.f.

x – reconstruction / fitting variable(s)

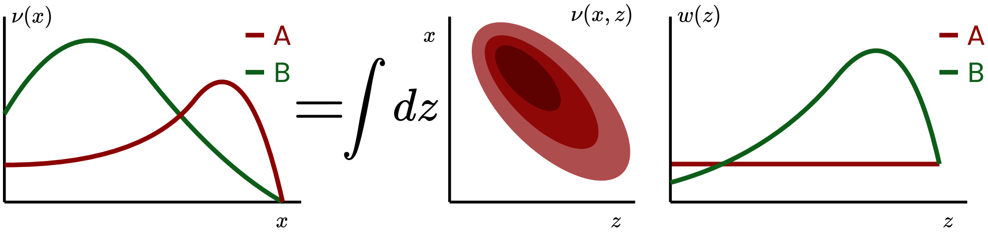
L – luminosity

$\varepsilon(x|z)$ – conditional efficiency

$n_\sigma(x,z)$ – joint number density

Reweight to new model

$$\boxed{n(x|B)} = \int dz\, L\, \varepsilon(x|z)\, \boxed{\sigma_B(z)} = \int dz\, L\, \varepsilon(x|z)\, \boxed{\sigma_A(z)}\, \frac{\boxed{\sigma_B(z)}}{\boxed{\sigma_A(z)}} = \int dz\, \underbrace{\boxed{n_A(x,z)}}_{\text{main object}}\, \boxed{w(z)}.$$



Discretization $\Rightarrow$ joint number density histogram

redist

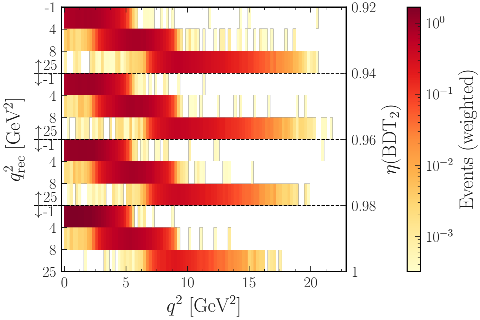
A reinterpretation framework

EPJC 84, 693 (2024)

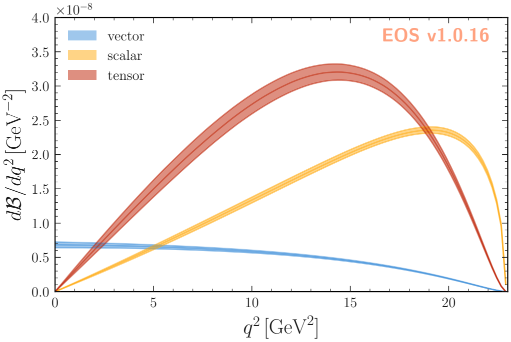
github.com/lorenzennio/redist

Application to Belle II $B^+ \to K^+\nu\bar{\nu}$ Result

Joint number density

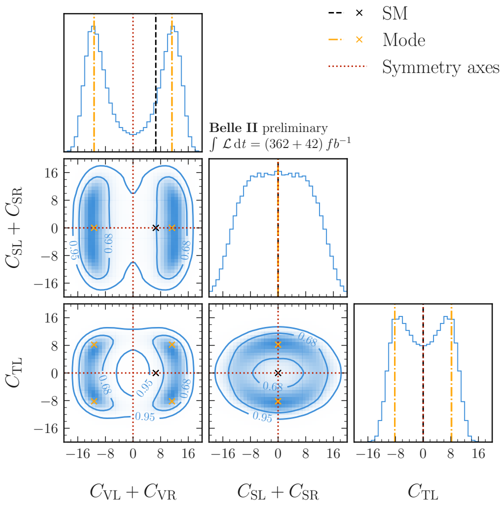


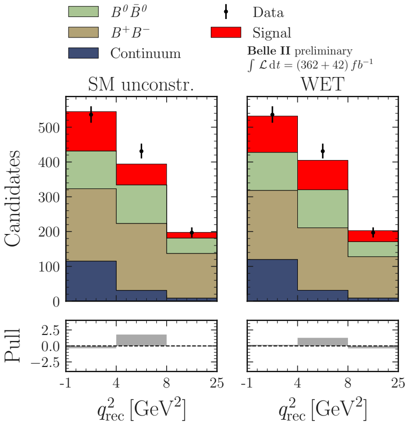
Weak Effective Theory (WET)



SM contains only *vector* contribution.

Result [arXiv:2507.12393]





First ever direct constraints on $b \to s\nu\bar{\nu}$ WET Wilson coefficients 🥳

Summary

TA3 is advancing FAIRness
with foundation models
and reinterpretation methods

→ Integration in SDP (in PUNCH 2.0)?