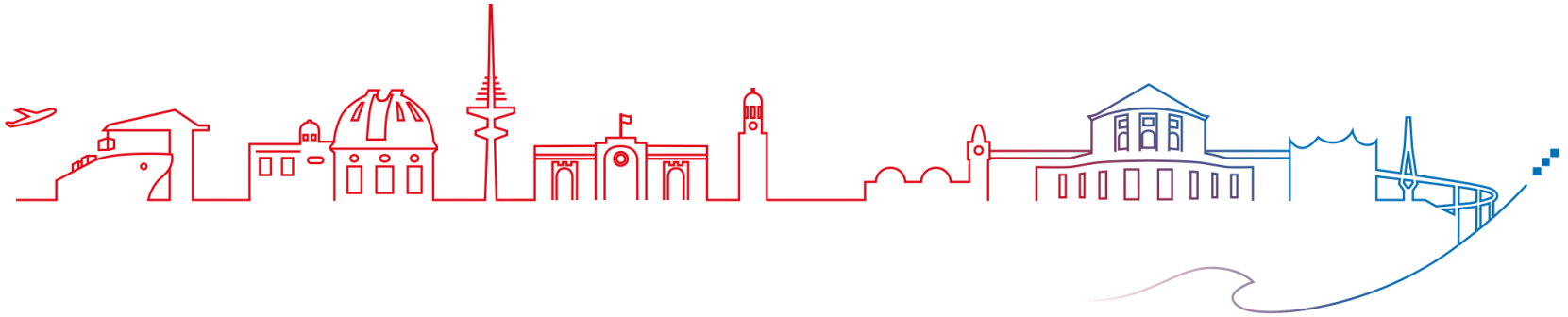




Universität Hamburg  
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CLUSTER OF EXCELLENCE  
QUANTUM UNIVERSE



# Foundation models for HEP

MIAPbP Build big or build smart workshop  
Munich, Sept 4 2025

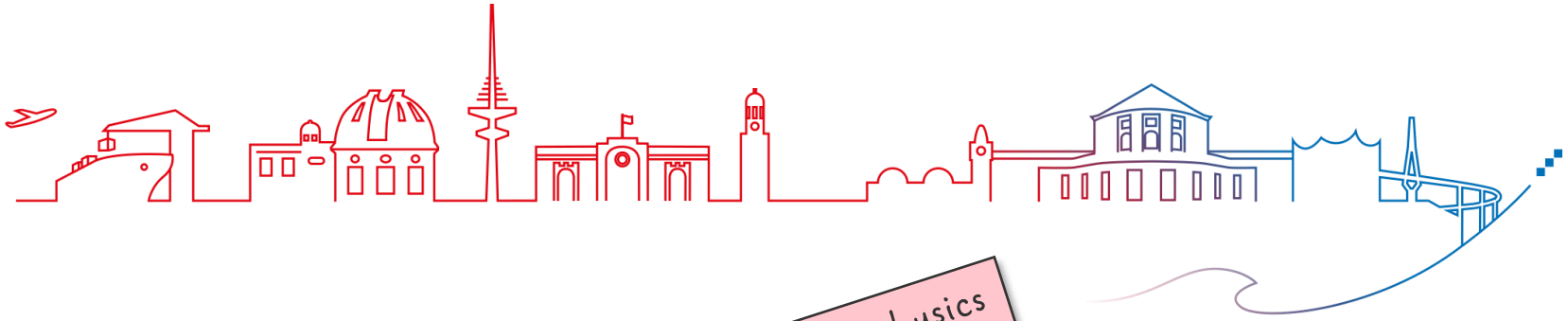
**Anna Hallin**

[anna.hallin@uni-hamburg.de](mailto:anna.hallin@uni-hamburg.de)



Universität Hamburg  
DER FORSCHUNG | DER LEHRE | DER BILDUNG

CLUSTER OF EXCELLENCE  
QUANTUM UNIVERSE



# Foundation models for HEP

MIAPbP Build big or build smart workshop  
Munich, Sept 4 2025

**Anna Hallin**

[anna.hallin@uni-hamburg.de](mailto:anna.hallin@uni-hamburg.de)

Collider physics

Without language models

# The (original) definition of foundation models

# On the Opportunities and Risks of Foundation Models

2022

Rishi Bommasani\* Drew A. Hudson Ehsan Adeli Russ Altman Simran Arora  
Sydney von Arx Michael S. Bernstein Jeannette Bohg Antoine Bosselut Emma Brunskill  
Erik Brynjolfsson Shyamal Buch Dallas Card Rodrigo Castellon Niladri Chatterji  
Annie Chen Kathleen Creel Jared Quincy Davis Dorottya Demszky Chris Donahue  
Moussa Doumbouya Esin Durmus Stefano Ermon John Etchemendy Kawin Ethayarajh

[2108.07258](#)

## 1 INTRODUCTION

This report investigates an emerging paradigm for building artificial intelligence (AI) systems based on a general class of models which we term *foundation models*.<sup>2</sup> A foundation model is any model that is trained on broad data (generally using self-supervision at scale) that can be adapted (e.g., fine-tuned) to a wide range of downstream tasks; current examples include BERT [Devlin et al. 2019], GPT-3 [Brown et al. 2020], and CLIP [Radford et al. 2021]. From a technological point of view, foundation models are not new — they are based on deep neural networks and self-supervised learning, both of which have existed for decades. However, the sheer scale and scope of foundation models from the last few years have stretched our imagination of what is possible; for example,

# On the Opportunities and Risks of Foundation Models

2022

Rishi Bommasani\* Drew A. Hudson Ehsan Adeli Russ Altman Simran Arora  
Sydney von Arx Michael S. Bernstein Jeannette Bohg Antoine Bosselut Emma Brunskill  
Erik Brynjolfsson Shyamal Buch Dallas Card Rodrigo Castellon Niladri Chatterji  
Annie Chen Kathleen Creel Jared Quincy Davis Dorottya Demszky Chris Donahue  
Moussa Doumbouya Esin Durmus Stefano Ermon John Etchemendy Kawin Ethayarajh

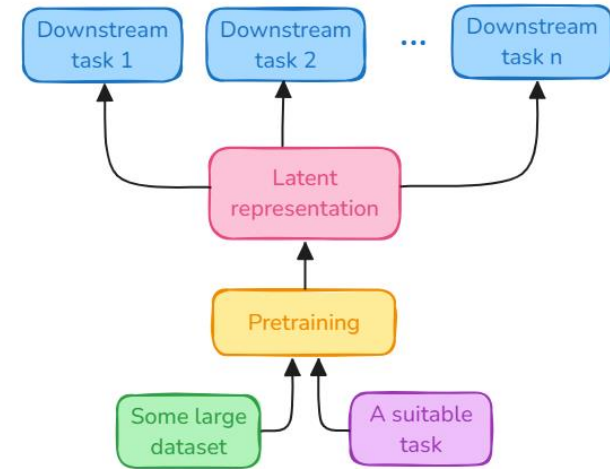
[2108.07258](#)

## 1 INTRODUCTION

This report investigates an emerging paradigm for building artificial intelligence (AI) systems based on a general class of models which we term *foundation models*.<sup>2</sup> A foundation model is any model that is trained on broad data (generally using self-supervision at scale) that can be adapted (e.g., fine-tuned) to a wide range of downstream tasks; current examples include BERT [Devlin et al. 2019], GPT-3 [Brown et al. 2020], and CLIP [Radford et al. 2021]. From a technological point of view, foundation models are not new — they are based on deep neural networks and self-supervised learning, both of which have existed for decades. However, the sheer scale and scope of foundation models from the last few years have stretched our imagination of what is possible; for example,

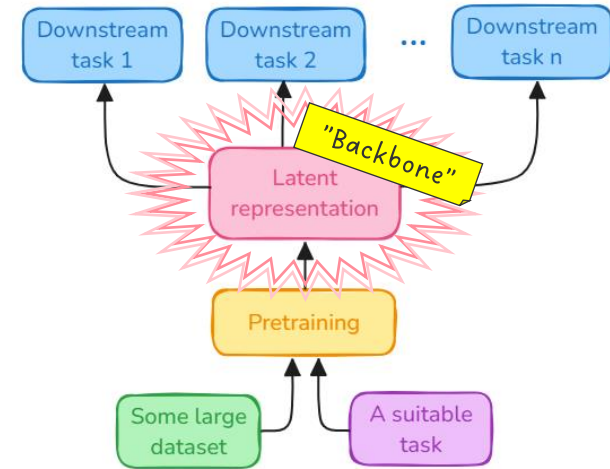
# Foundation models

- Definition:
  - A foundation model is a machine learning model that once **pretrained** can be **finetuned** to different downstream tasks (Bommasani 2021)
  - The **performance** of pretraining + finetuning is better than training on the downstream task from scratch



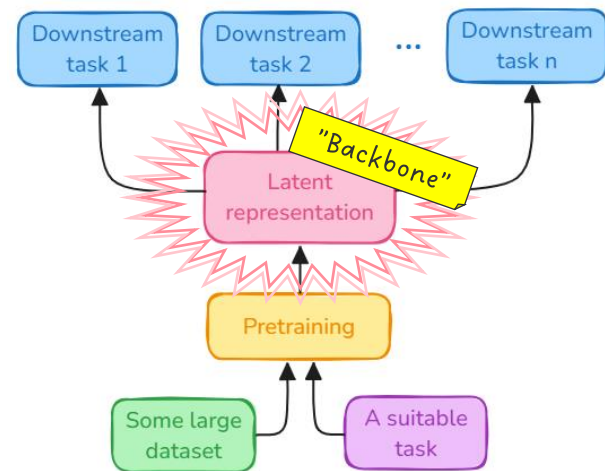
# Foundation models

- Definition:
  - A foundation model is a machine learning model that once **pretrained** can be **finetuned** to different downstream tasks (Bommasani 2021<sup>6</sup>)
  - The **performance** of pretraining + finetuning is better than training on the downstream task from scratch








# Foundation models

- Definition:
  - A foundation model is a machine learning model that once **pretrained** can be **finetuned** to different downstream tasks (Bommasani 2021<sup>6</sup>)
  - The **performance** of pretraining + finetuning is better than training on the downstream task from scratch
- **Transfer learning** per se is nothing new, and it was already known from image models that earlier layers learn general properties of the data, and final layers focus on the specifics (Yosinski et al 2014<sup>6</sup>)
- The introduction of **transformers** and **scaling** up of datasets and models is what led to the era of foundation models
- 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.

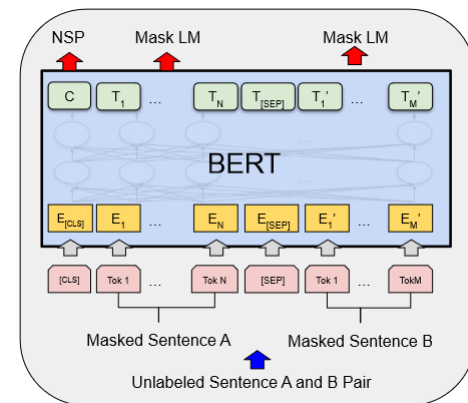


# Why would we want to use them?

- 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) 

# Pretraining

- Can be useful in itself, or a **surrogate task**
- Example of surrogate tasks: BERT
  - Input: 2 sentences with masked out parts
  - **Masked language modeling** in addition to **next sentence prediction**
    - Masking out tokens allows bidirectional training: sees both previous and future words in order to capture the **context within a sentence**
    - Next sentence prediction captures **context between sentences**: does sentence B follow sentence A?
  - Finetuning for different question-answer tasks or sentence pairing tasks



1810.04805

Devlin et al, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv [1810.04805](https://arxiv.org/abs/1810.04805)

# Scale

Foundation models become powerful because of **scale: data, architecture, compute**

- Example GPT-3: 300B tokens, 175 billion parameters, estimated thousands of GPUs trained over several weeks ( $\sim 10^{23}$  flops)

# Scale

Foundation models become powerful because of **scale**: **data**, **architecture**, **compute**

- Example GPT-3: 300B tokens, 175 billion parameters, estimated thousands of GPUs trained over several weeks ( $\sim 10^{23}$  flops)
- Parameter scale example: Parti (Pathways Autoregressive Text-to-Image model)

**Parti-350M**



A portrait photo of a kangaroo wearing an orange hoodie and blue sunglasses standing on the grass in front of the Sydney Opera House holding a sign on the chest that says Welcome Friends!

2206.10789

Yu et al, *Scaling Autoregressive Models for Content-Rich Text-to-Image Generation*. arXiv [2206.10789](https://arxiv.org/abs/2206.10789)

# Scale

Foundation models become powerful because of **scale: data, architecture, compute**

- Example GPT-3: 300B tokens, 175 billion parameters, estimated thousands of GPUs trained over several weeks ( $\sim 10^{23}$  flops)
- Parameter scale example: Parti (Pathways Autoregressive Text-to-Image model)



A portrait photo of a kangaroo wearing an orange hoodie and blue sunglasses standing on the grass in front of the Sydney Opera House holding a sign on the chest that says Welcome Friends!

2206.10789

Yu et al, *Scaling Autoregressive Models for Content-Rich Text-to-Image Generation*. arXiv [2206.10789](https://arxiv.org/abs/2206.10789)

# Scale

Foundation models become powerful because of **scale: data, architecture, compute**

- Example GPT-3: 300B tokens, 175 billion parameters, estimated thousands of GPUs trained over several weeks ( $\sim 10^{23}$  flops)
- Parameter scale example: Parti (Pathways Autoregressive Text-to-Image model)



A portrait photo of a kangaroo wearing an orange hoodie and blue sunglasses standing on the grass in front of the Sydney Opera House holding a sign on the chest that says Welcome Friends!

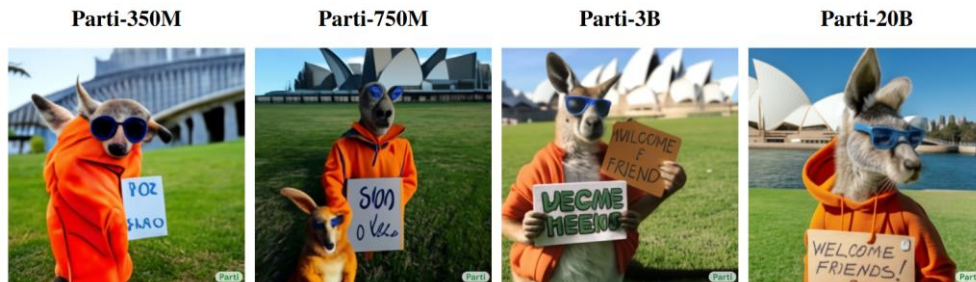
2206.10789

Yu et al, *Scaling Autoregressive Models for Content-Rich Text-to-Image Generation*. arXiv [2206.10789](https://arxiv.org/abs/2206.10789)

# Scale

Foundation models become powerful because of **scale: data, architecture, compute**

- Example GPT-3: 300B tokens, 175 billion parameters, estimated thousands of GPUs trained over several weeks ( $\sim 10^{23}$  flops)
- Parameter scale example: Parti (Pathways Autoregressive Text-to-Image model)



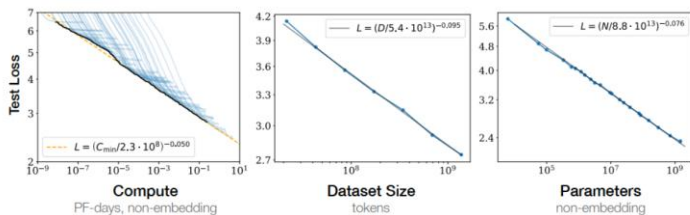
A portrait photo of a kangaroo wearing an orange hoodie and blue sunglasses standing on the grass in front of the Sydney Opera House holding a sign on the chest that says Welcome Friends!

2206.10789

Yu et al, *Scaling Autoregressive Models for Content-Rich Text-to-Image Generation*. arXiv [2206.10789](https://arxiv.org/abs/2206.10789)

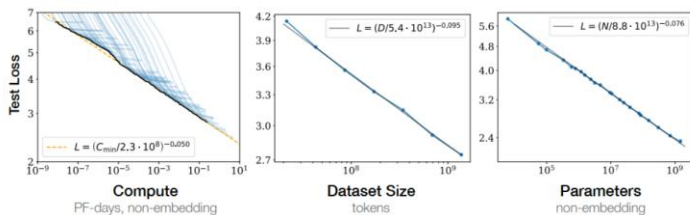
# Scaling laws

- Scaling compute, dataset size or parameters leads to a **predictable** decrease in **loss** (Kaplan et al 2020)

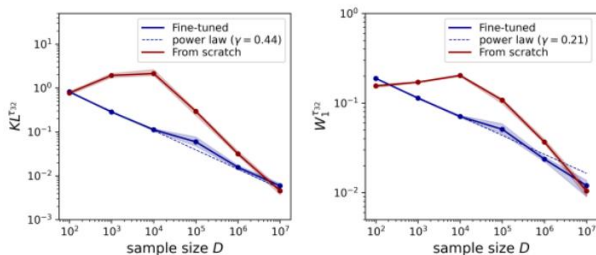


# Scaling laws

- Scaling compute, dataset size or parameters leads to a **predictable** decrease in **loss** (Kaplan et al 2020)



- Hint of scaling laws also seen in **other observables**, here the  $\tau_{32}$  of generated top jets (Amram, AH et al 2024)

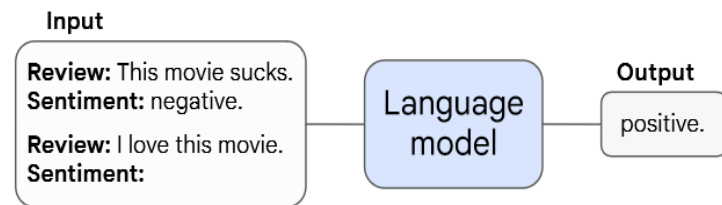


# Scale leads to emergent properties

A foundation model might be able to perform tasks that it was **not trained for**, and that were **not anticipated**. This behavior comes with **scale**.

Examples for a natural language model only trained to generate text:

- Translation
- Coding
- Basic arithmetic
- Sentiment analysis
- Few-shot and zero-shot learning

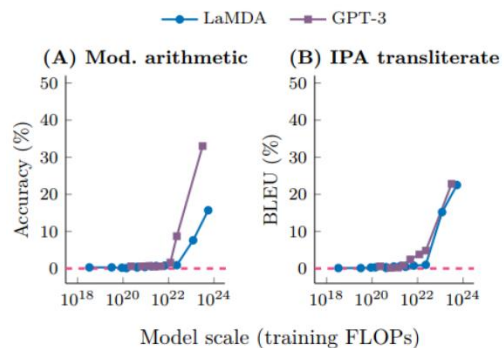


2206.07682

Bommasani et al, *On the Opportunities and Risks of Foundation Models*. arXiv [2108.07258](https://arxiv.org/abs/2108.07258)

# Emergent abilities

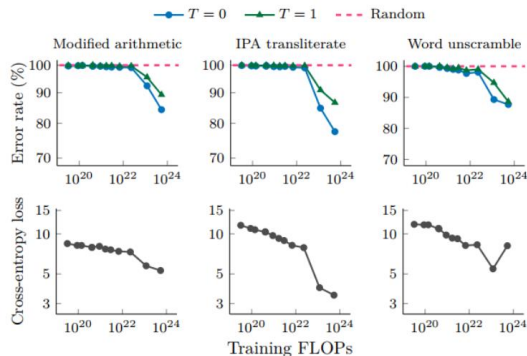
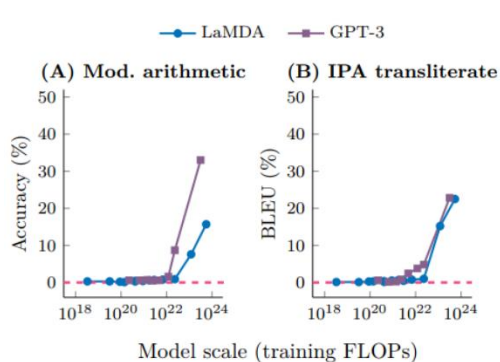
- A **phase-transition** in **downstream task abilities** that comes with scaling up (Wei et al 2022)



Mod. Arithmetic = 3-digit addition and subtraction and 2-digit multiplication; IPA transliterate = transliterating the phonetic alphabet

# Emergent abilities

- A **phase-transition** in **downstream task abilities** that comes with scaling up (Wei et al 2022)
- **Not** necessarily **correlated** with performance as measured by **loss**
- **Not** necessarily **predictable**

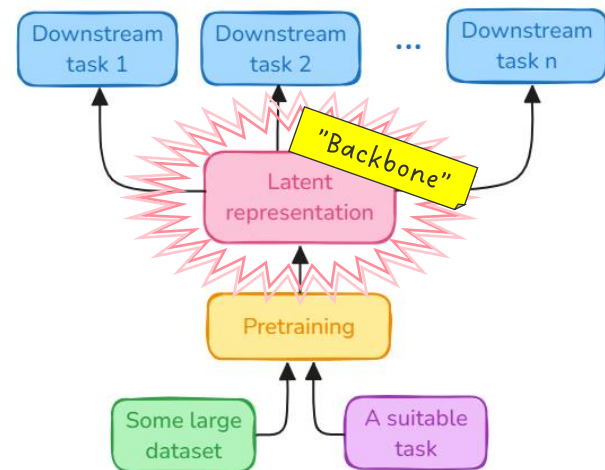


Note that for FLOPs <  $10^{22}$ , the **loss decreases** even though the model **performance** is still **random**

Mod. Arithmetic = 3-digit addition and subtraction and 2-digit multiplication; IPA transliterate = transliterating the phonetic alphabet

# A note on transformers, LLMs...

- Transformers are a common architectural choice for foundation models
  - But a transformer in itself is not a foundation model
- Large language models (LLMs) are foundation models
  - But not all foundation models are LLMs



# Approaches to foundation models in physics

- **Use** large language models for applications in physics
  - **AI Physicist**: an AI agent (multiple LLMs working together) trying to do anomaly detection (paper coming soon™)
  - Define the desired conditions for your experiment and let the LLM **tune your accelerator**

Diefenbacher, **AH**, Kasieczka, Krämer, Lauscher, Lukas, 2509.XXXX

Kaiser et al, *Large Language Models for Human-Machine Collaborative Particle Accelerator Tuning through Natural Language*. arXiv [2405.08888](https://arxiv.org/abs/2405.08888)

# Approaches to foundation models in physics

- **Use** large language models for applications in physics
  - **AI Physicist**: an AI agent (multiple LLMs working together) trying to do anomaly detection (paper coming soon™)
  - Define the desired conditions for your experiment and let the LLM **tune your accelerator**
- **Teach/adapt** large language models to do maths and physics
  - Symbolic maths: compute integrals and solve differential equations by treating equations and their solutions as a **translation** task
  - Number embedding in text: treat numbers as a **different entity** than text, to allow the model to "understand" numbers

Diefenbacher, [AH](#), Kasieczka, Krämer, Lauscher, Lukas, 2509.XXXX

Kaiser et al, *Large Language Models for Human-Machine Collaborative Particle Accelerator Tuning through Natural Language*. arXiv [2405.08888](#)

Lample and Charton, *Deep Learning for Symbolic Mathematics*. arXiv [1912.01412](#)

Golkar et al, *xVal: A Continuous Number Encoding for Large Language Models*. arXiv [2310.02989](#)



# Approaches to foundation models in physics

- **Use** large language models for applications in physics
  - **AI Physicist**: an AI agent (multiple LLMs working together) trying to do anomaly detection (paper coming soon™)
  - Define the desired conditions for your experiment and let the LLM **tune your accelerator**
- **Teach/adapt** large language models to do maths and physics
  - Symbolic maths: compute integrals and solve differential equations by treating equations and their solutions as a **translation** task
  - Number embedding in text: treat numbers as a **different entity** than text, to allow the model to "understand" numbers
- Take **inspiration** from large language models and others, **build from scratch**
  - The remainder of the talk will focus on this approach

Diefenbacher, [AH](#), Kasieczka, Krämer, Lauscher, Lukas, 2509.XXXX

Kaiser et al, *Large Language Models for Human-Machine Collaborative Particle Accelerator Tuning through Natural Language*. arXiv [2405.08888](#)

Lample and Charton, *Deep Learning for Symbolic Mathematics*. arXiv [1912.01412](#)

Golkar et al, *xVal: A Continuous Number Encoding for Large Language Models*. arXiv [2310.02989](#)



# A particle physics foundation model example

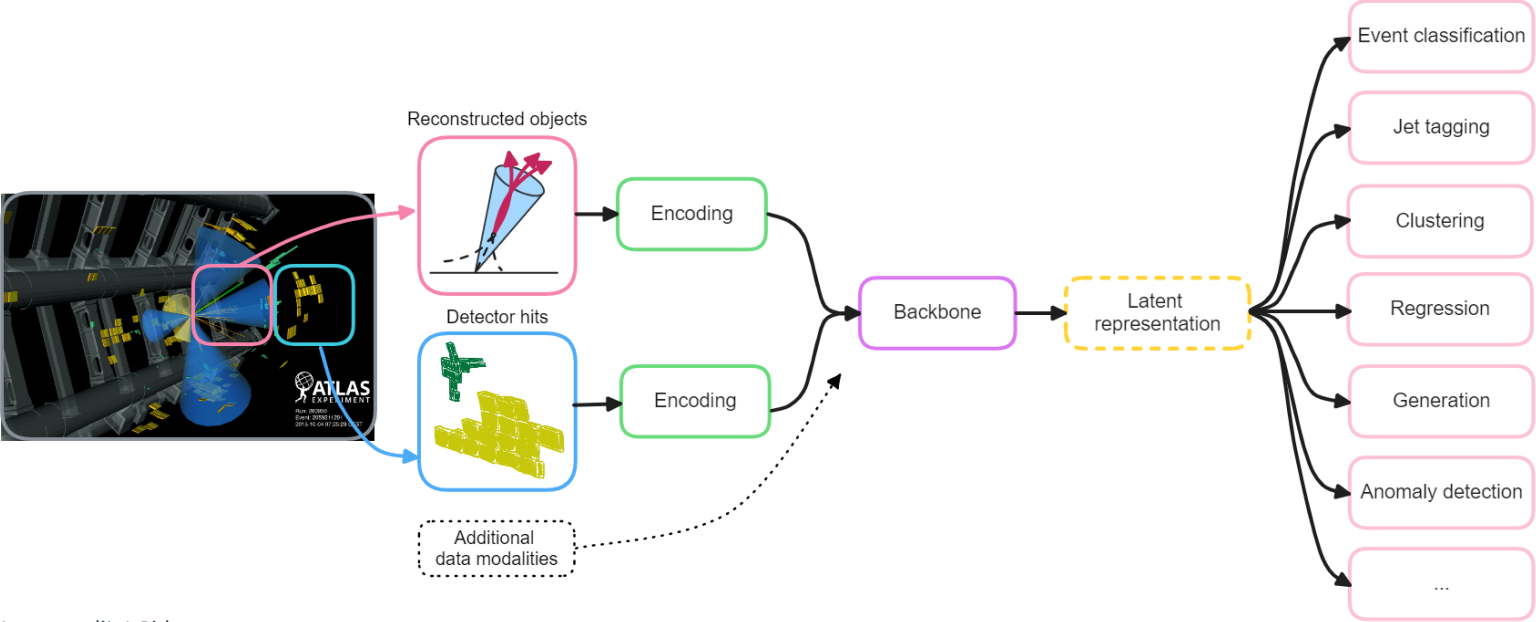
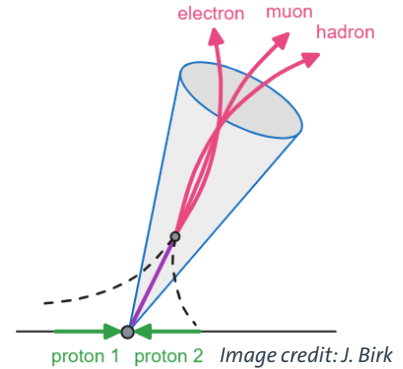


Image credit: J. Birk

# A survey of foundation models for jet physics\*

\* If I forgot your favorite model, please let me know!

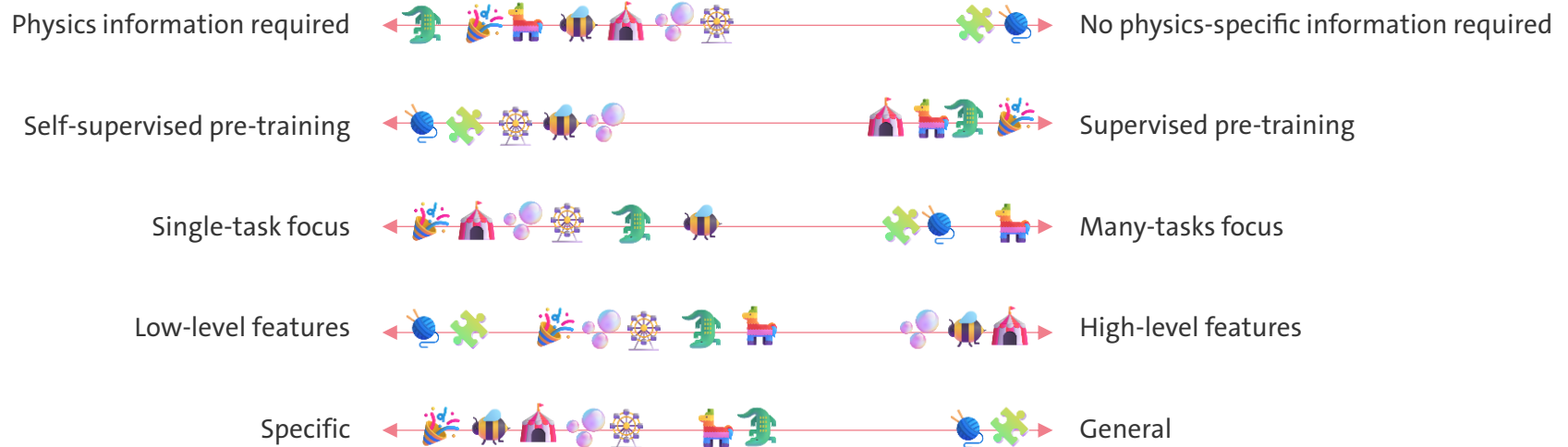
- 🎨 ParT (Qu et al 2022 [🔗](#)): classification (supervised) -> classification on different dataset
- 🧩 MPM (Golling et al 2024 [🔗](#)): masked prediction (self-supervised) -> classification/anomaly detection
- 🌐 OmniJet- $\alpha$  (Birk, [AH](#) et al 2024 [🔗](#)): generation (self-supervised) -> classification
- 🏠 OmniLearn (Mikuni et al 2024 [🔗](#)): generation and classification (supervised) -> classification, anomaly detection, unfolding
- 🌐 Joint-Embedding Predictive Architecture variants: HEP-JEPA (Bardhan et al 2025 [🔗](#)), J-JEPA (Katel et al 2024 [🔗](#)): prediction of target in latent space (self-supervised) -> classification, anomaly detection
- 🦜 L-GATr (Brehmer et al 2024 [🔗](#)): classification (supervised) -> classification on different dataset
- 🌐 RS3L (Harris et al 2024 [🔗](#)): contrastive learning (self-supervised) -> classification
- 🐝 Bumblebee (Wildridge et al 2024 [🔗](#)): masked prediction/event reconstruction (self-supervised) -> classification
- 🏠 Event classification (Ho et al 2024 [🔗](#)): multiclass and multilabel classification (supervised) -> binary classification



# Similarities and differences\*



\* Disclaimer: The arrangement for illustrative purposes only. Placements are approximate and not to scale. Different interpretations are possible. The ordering of models that appear in clusters is arbitrary.



# A cross-task foundation model for jet physics

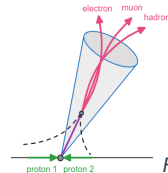




Figure: J. Birk

- **OmniJet- $\alpha$**  (Birk, **AH** et al 2024 ) 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

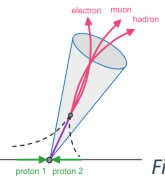


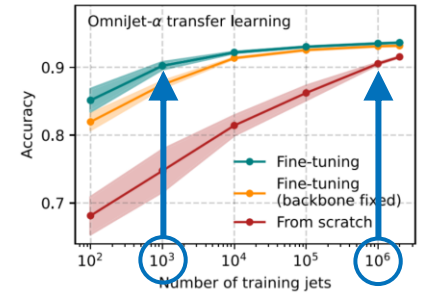
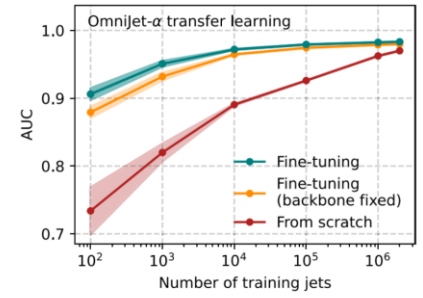


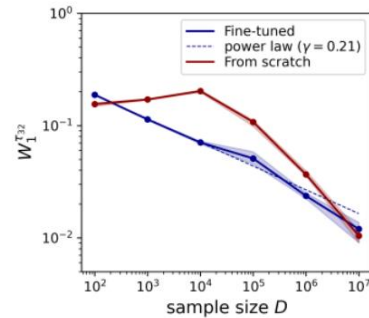
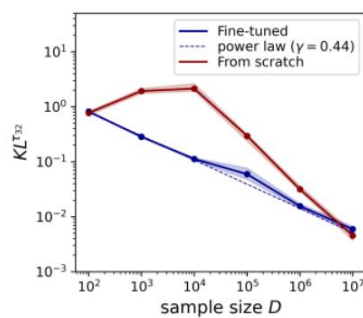
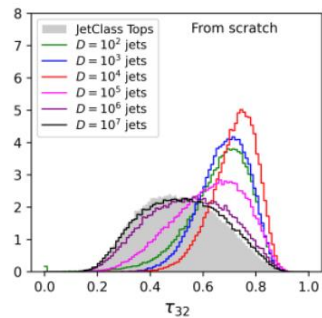
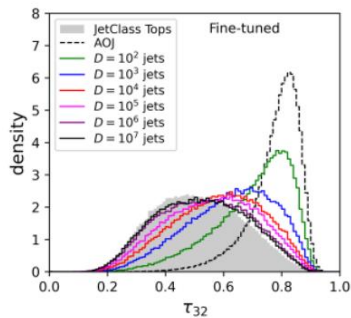
Figure: J. Birk

- **Omnijet- $\alpha$**  (Birk, AH et al 2024 ) 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:  $p(x_j | x_{j-1}, \dots, x_0)$
- **Finetune on supervised classification**
  - Demonstrated that the pretrained model **outperformed** the model trained from scratch, in particular on **very small datasets**



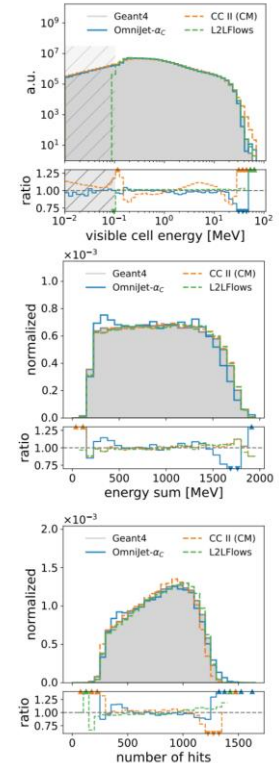
# Training on real data

- **Aspen Open Jets** (Amram, [AH](#) et al 2024 [🔗](#))
  - An **unlabeled** dataset [🔗](#) derived from CMS Open data, containing 180M jets
  - Expected to contain mostly QCD jets, and  $\sim 10^5$  top jets
- **Pretrain** OmniJet- $\alpha$  on this dataset, then **finetune** on generation of hadronically decaying top jets (simulation)  $\rightarrow$  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- $\alpha_c$**  (Birk, [AH et al 2024\(b\)](#)) applies the OmniJet- $\alpha$  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- $\alpha$  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**



# A vision for HEP: a common framework

- Beyond jets and calorimeters, what else can we do?
- Is it possible to create a **framework**, a **uniform interface** or a **common toolset**, that can be used for many different types of data and tasks?
- Even though some tasks may not be weight compatible, they might be **architecturally compatible**
- Some analysis tasks, like classification/tagging, may be performed as **finetunings** for specific analysis cases, avoiding double work and saving on compute resources
- Foundation models become powerful due to **scale** – are we there yet? If not, do we need more data, larger models, more compute? A common framework would make consolidation of compute resources and data possible, allowing us to scale it up

# Summary

- Foundation models are large models combining (usually) **self-supervised** training and **transfer learning**, trained on **large amounts of data** and providing a **rich representation** from which **downstream tasks** can be performed
- HEP as a field has already seen **several developments** of foundation models, most focusing on **classification**
- **Visions** range from modest (improving performance on a single task) to grandiose (a model that can do everything)
- Many **open questions** remain, eg regarding whether observations (eg. scale, emergence) from language models also apply to foundation models in HEP, and if yes, what that entails

