



李政道研究所
TSUNG-DAO LEE INSTITUTE

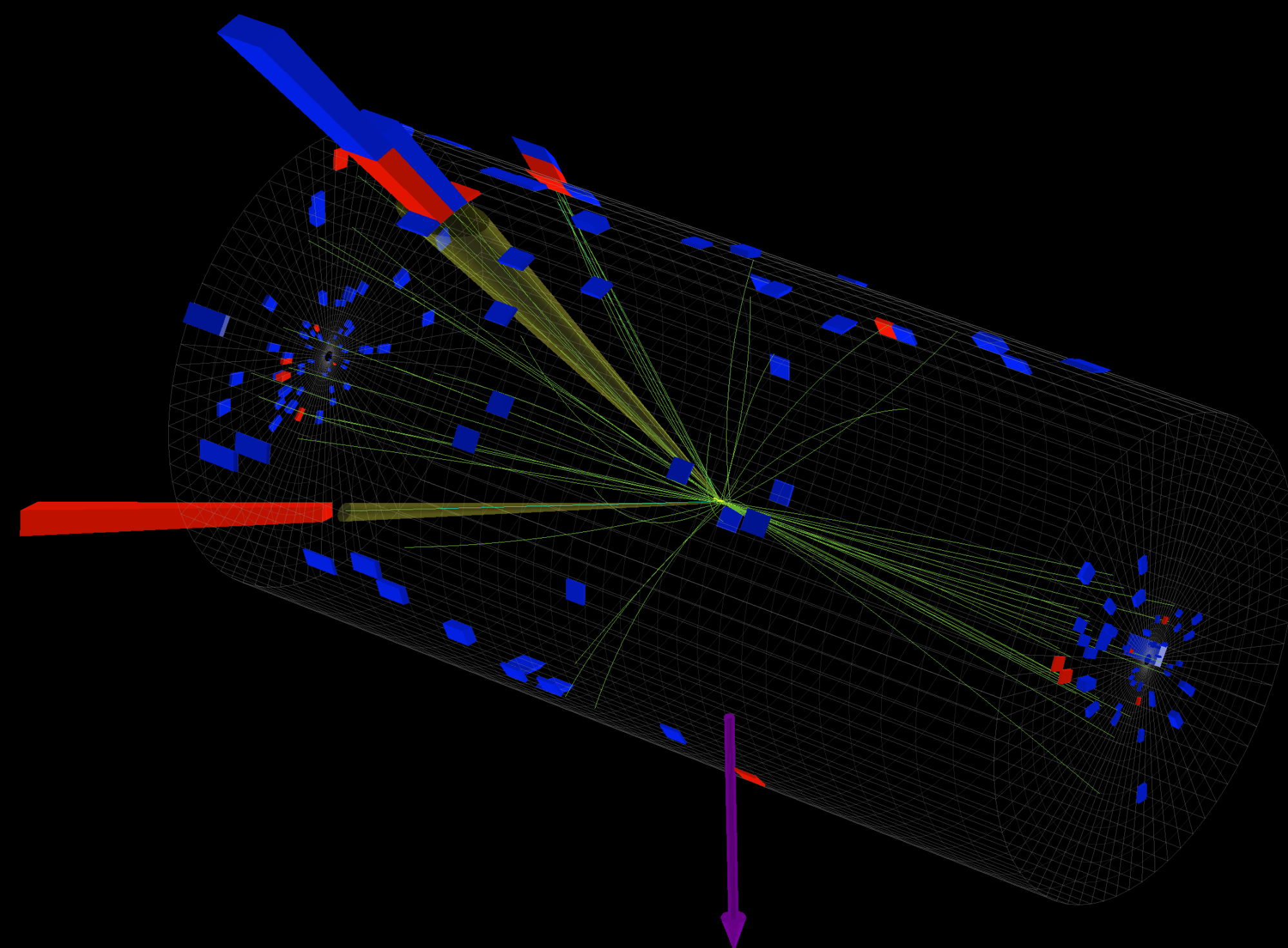
AI For HEP

— *A Representation Learning Perspective*

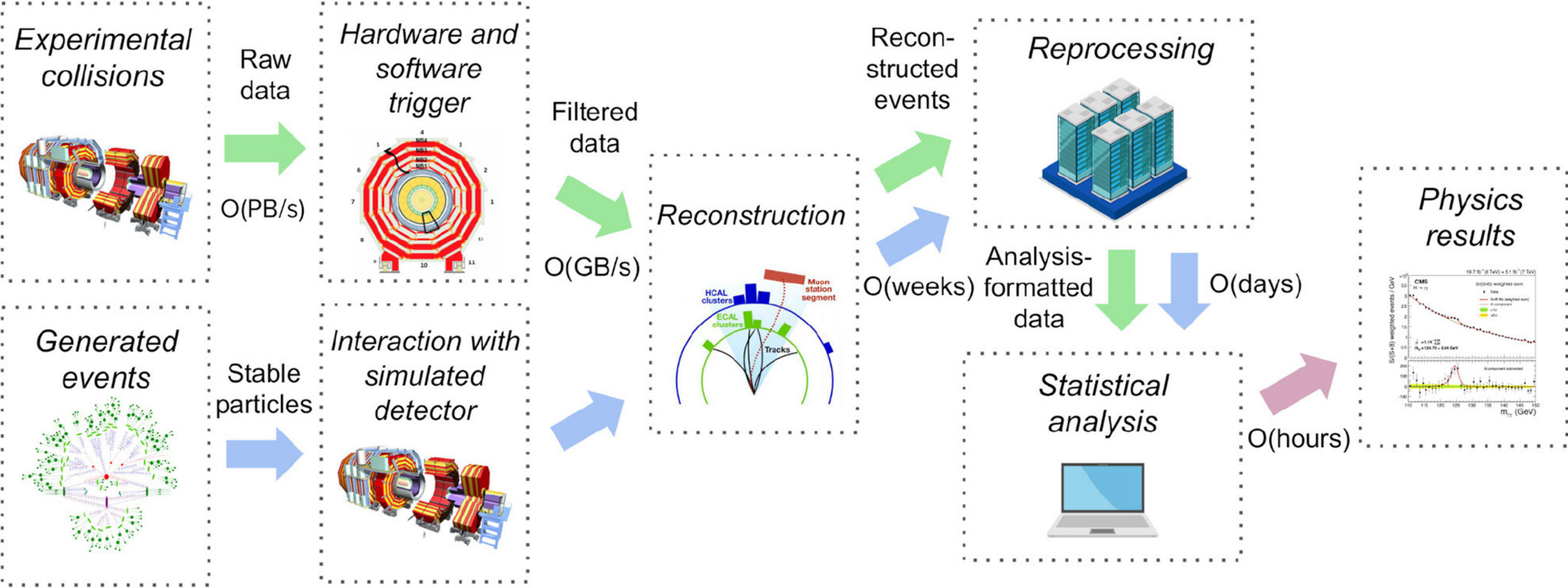
Huilin Qu (曲慧麟)

中国物理学会高能物理分会
第十二届全国会员代表大会暨学术年会

July 16, 2026

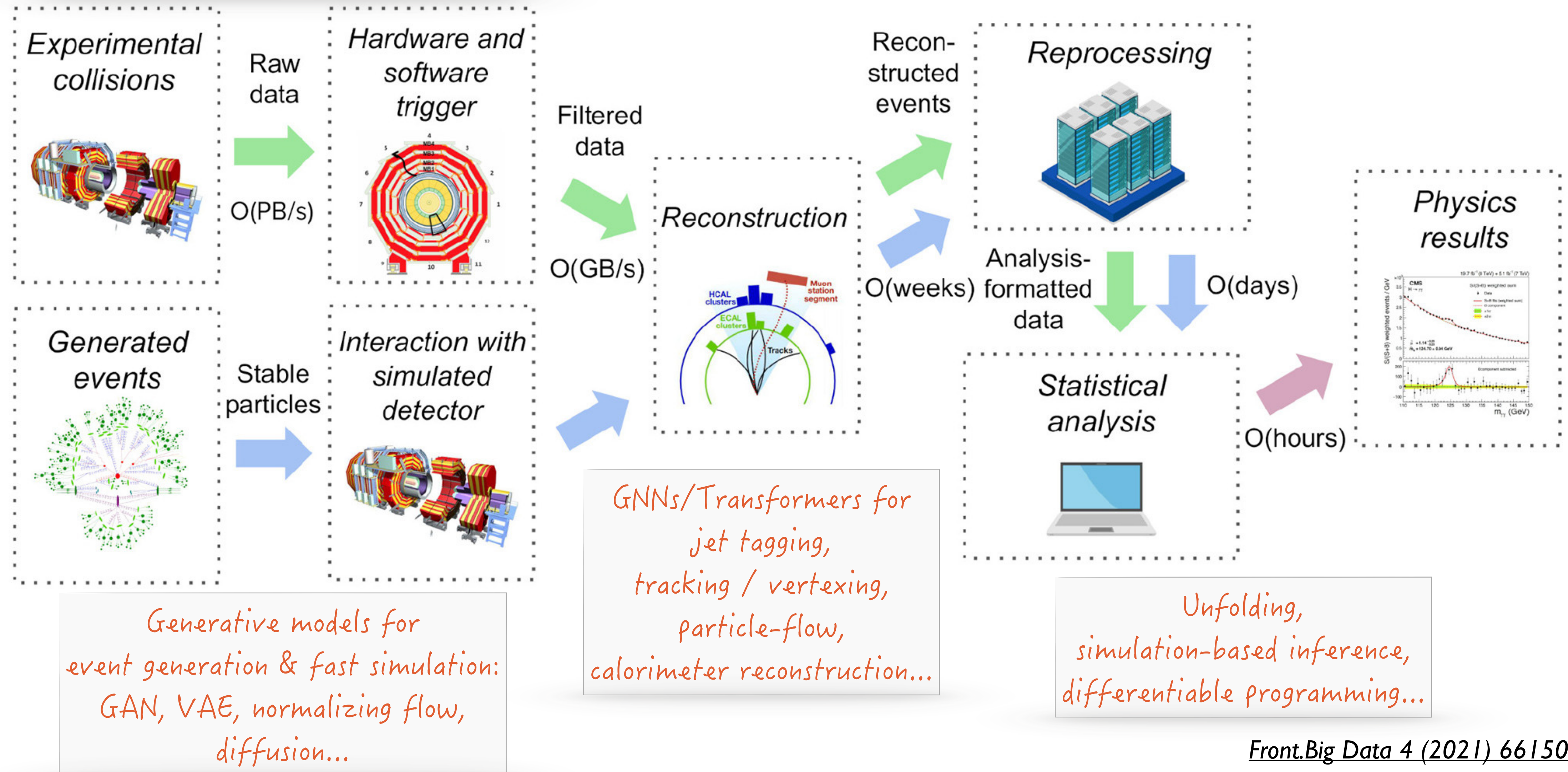


HEP DATA FLOW...



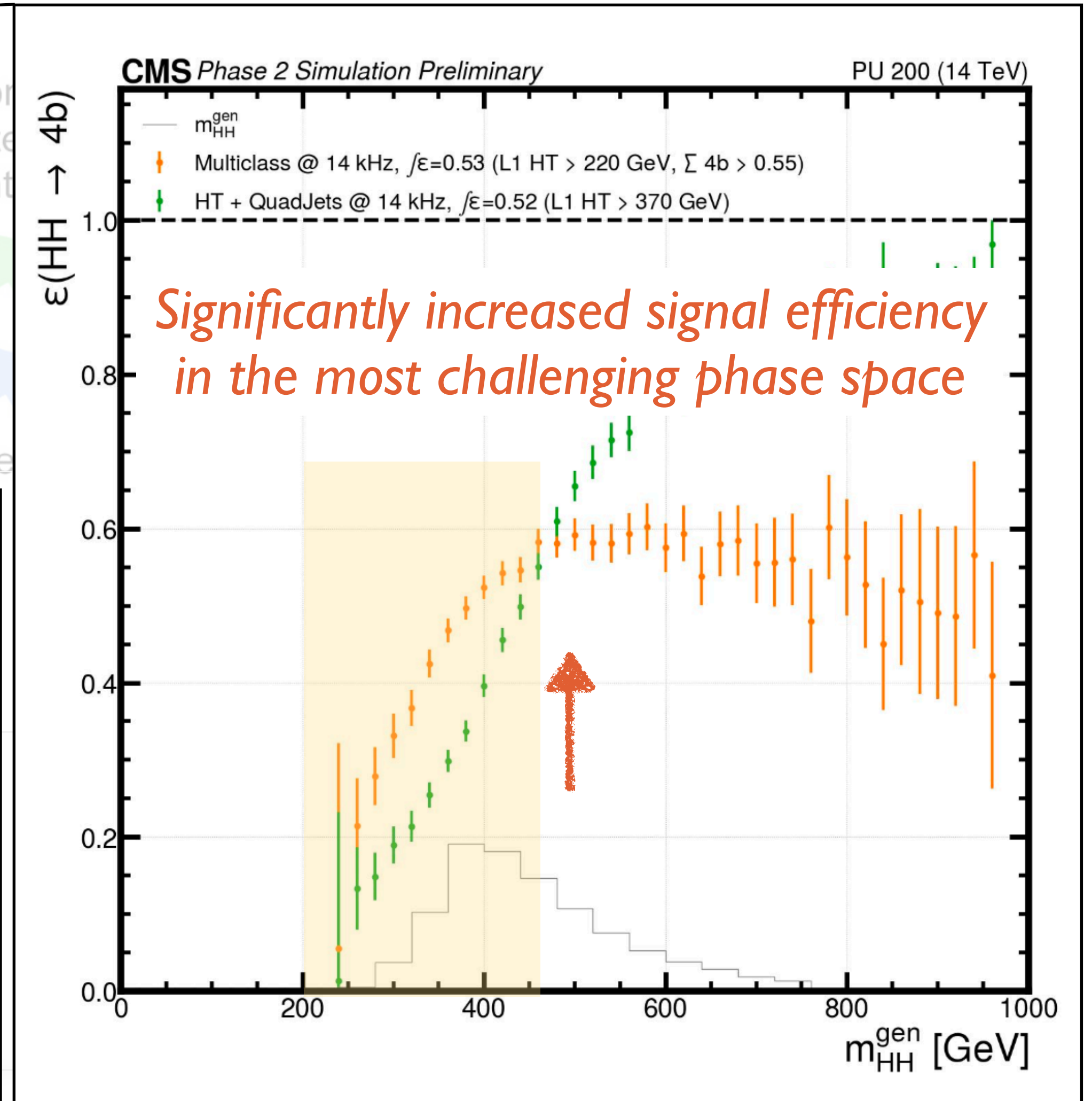
HEP DATA FLOW... MATCHED WITH ML!

Ultrafast inference (FPGA/ASIC),
data compression, anomaly detection...



AI for HEP - July 16, 2026 - Huilin Qu (TDLI)

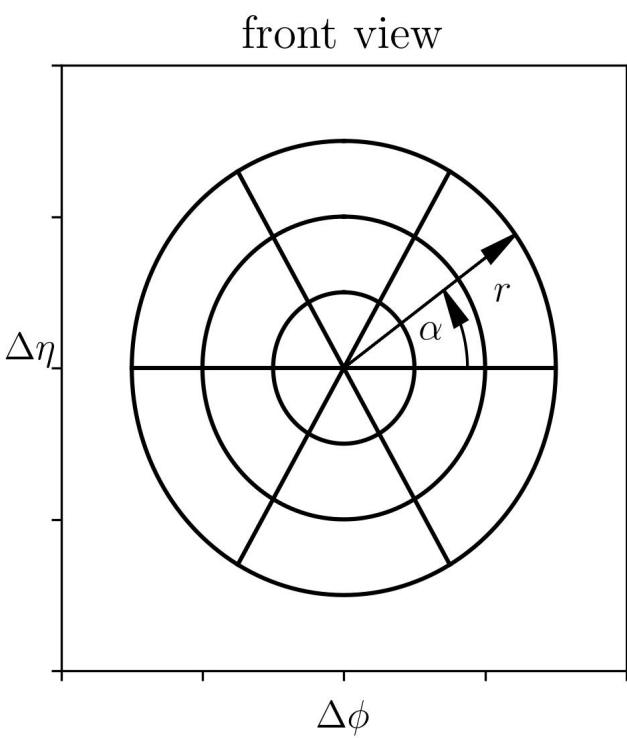
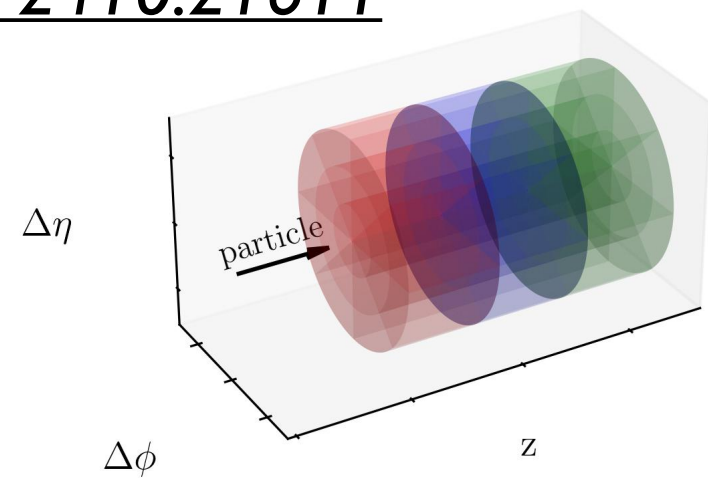
CMS-DP-2025-032



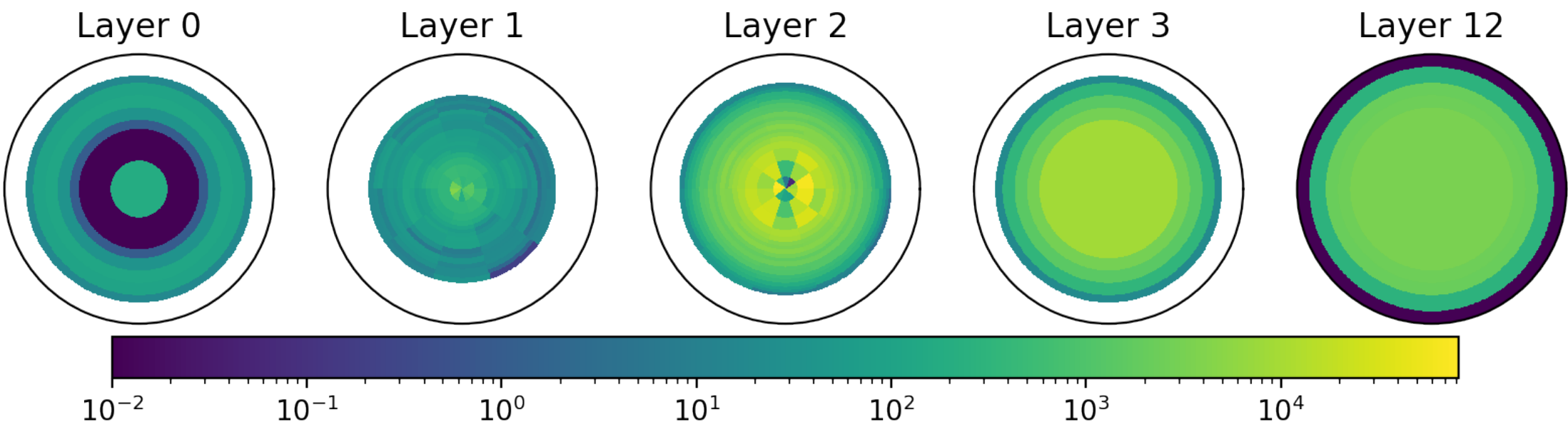
HEP DATA FLOW... MATCHED WITH ML!

Fast calorimeter simulation
using generative models

arXiv: 2410.21611



Photon shower at E = 1048.6 GeV



Generated
events



Stable
particles

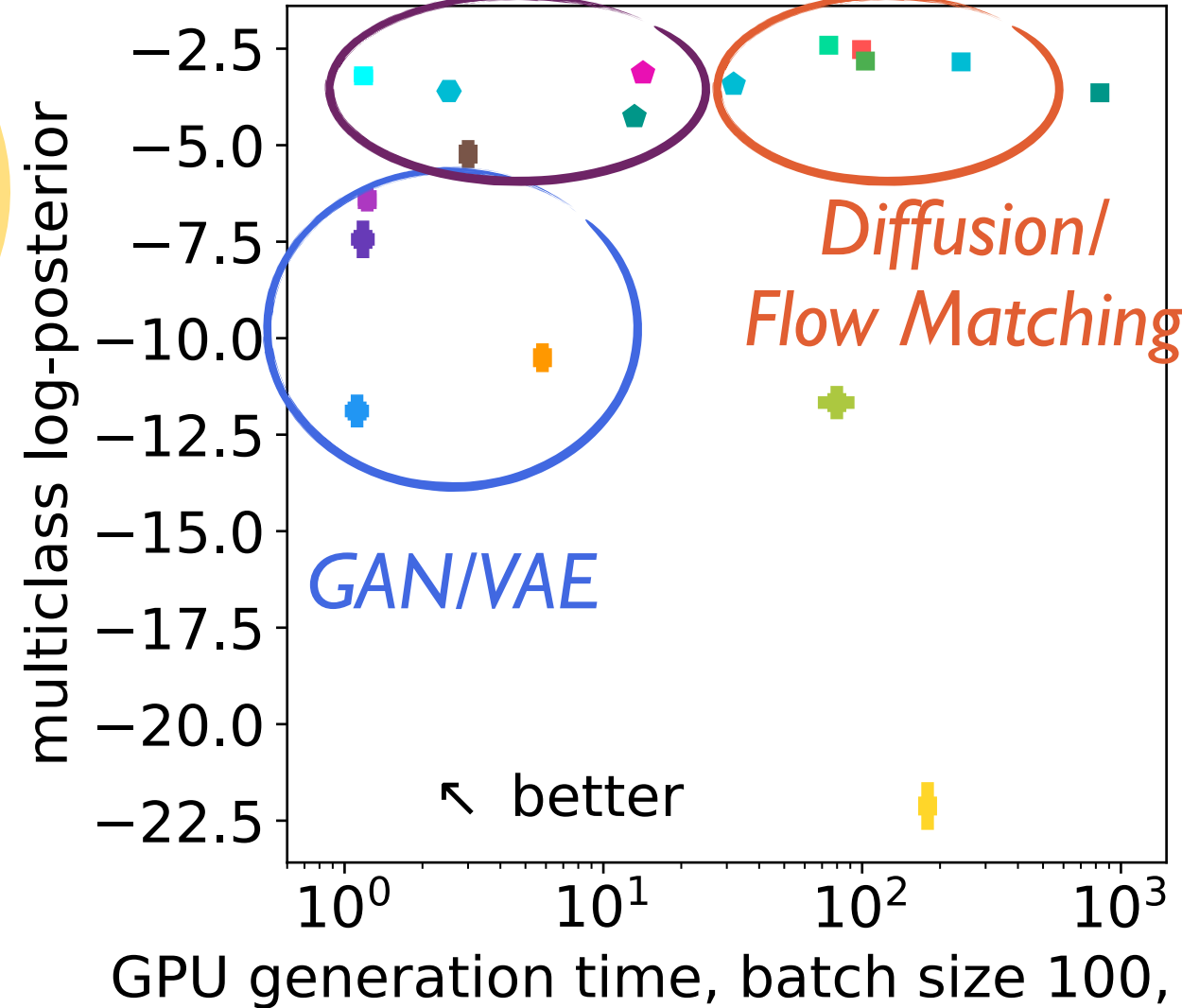
Interaction with
simulated
detector



Generative models for
event generation & fast simulation:
GAN, VAE, normalizing flow,
diffusion...

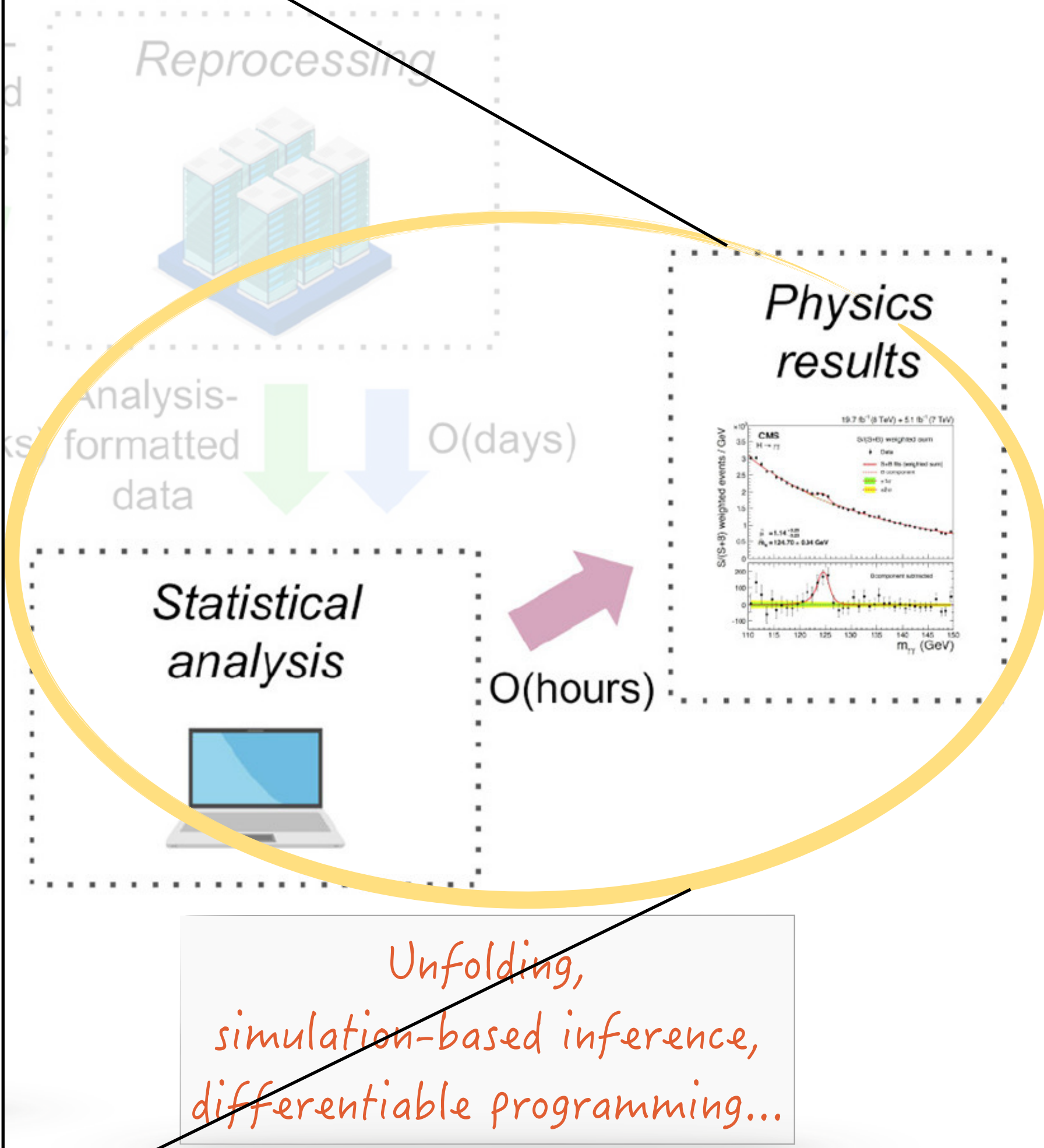
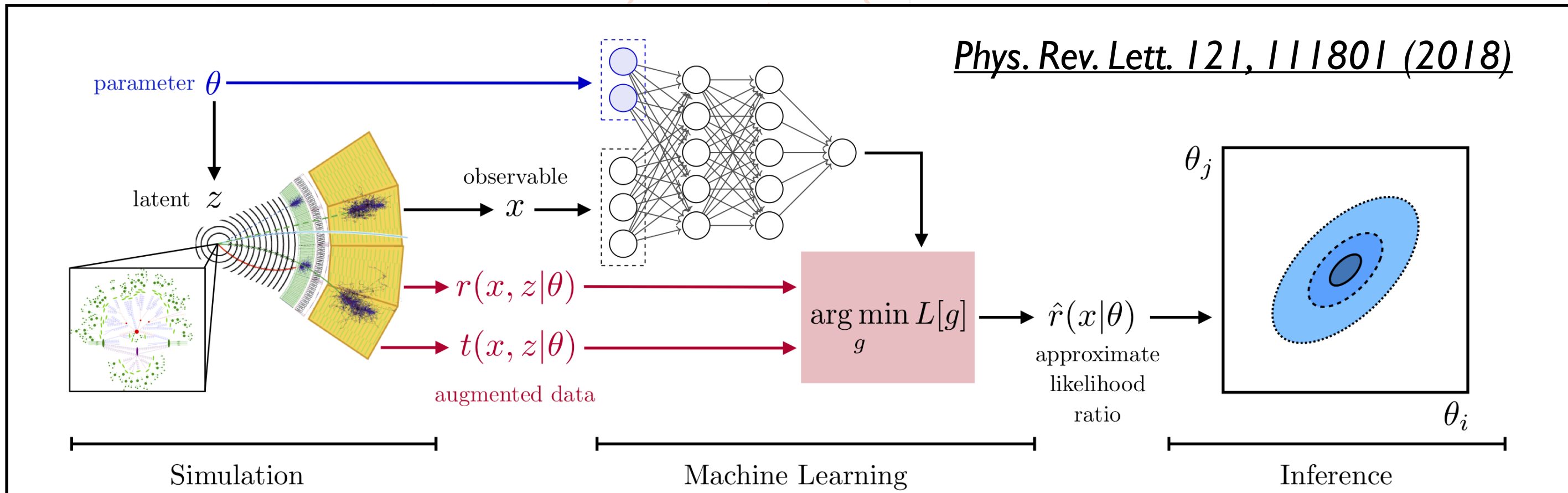
Normalizing
Flow

Pareto front: shower generation time on a GPU vs. multiclass log-posterior, dataset 2

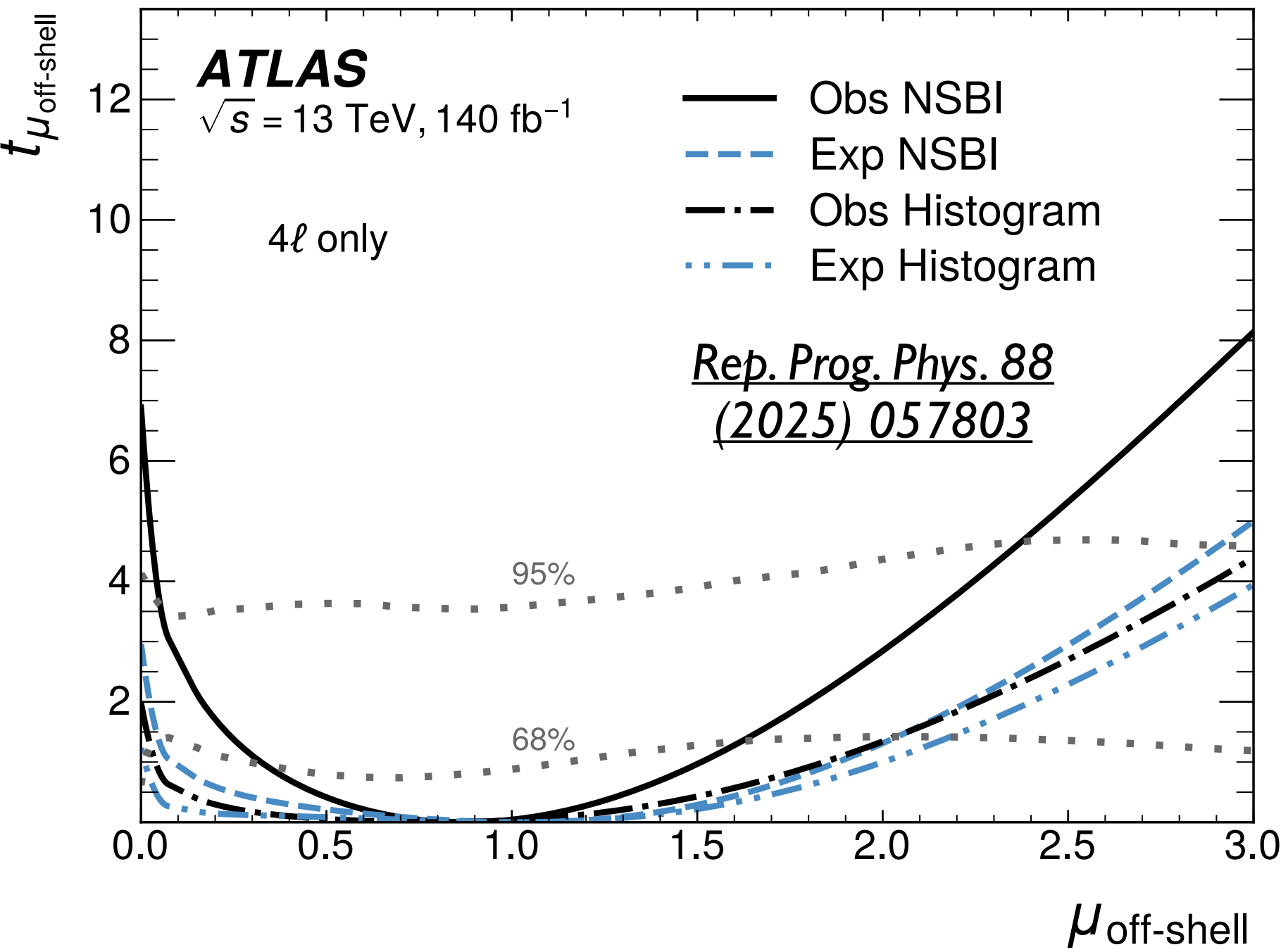


- | | |
|-----------------------|-------------------|
| CaloDiffusion | iCaloFlow student |
| conv. L2LFlows | SuperCalo |
| CaloINN | DeepTree |
| MDMA | CaloPointFlow |
| Calo-VQ | CaloVAE+INN |
| CaloScore | CaloLatent |
| CaloScore distilled | CaloDiT |
| CaloScore single-shot | CaloDREAM |
| iCaloFlow teacher | |

HEP DATA FLOW... MATCHED WITH ML!



Approximating complex likelihood using neural networks for faster and more optimal statistical inference.

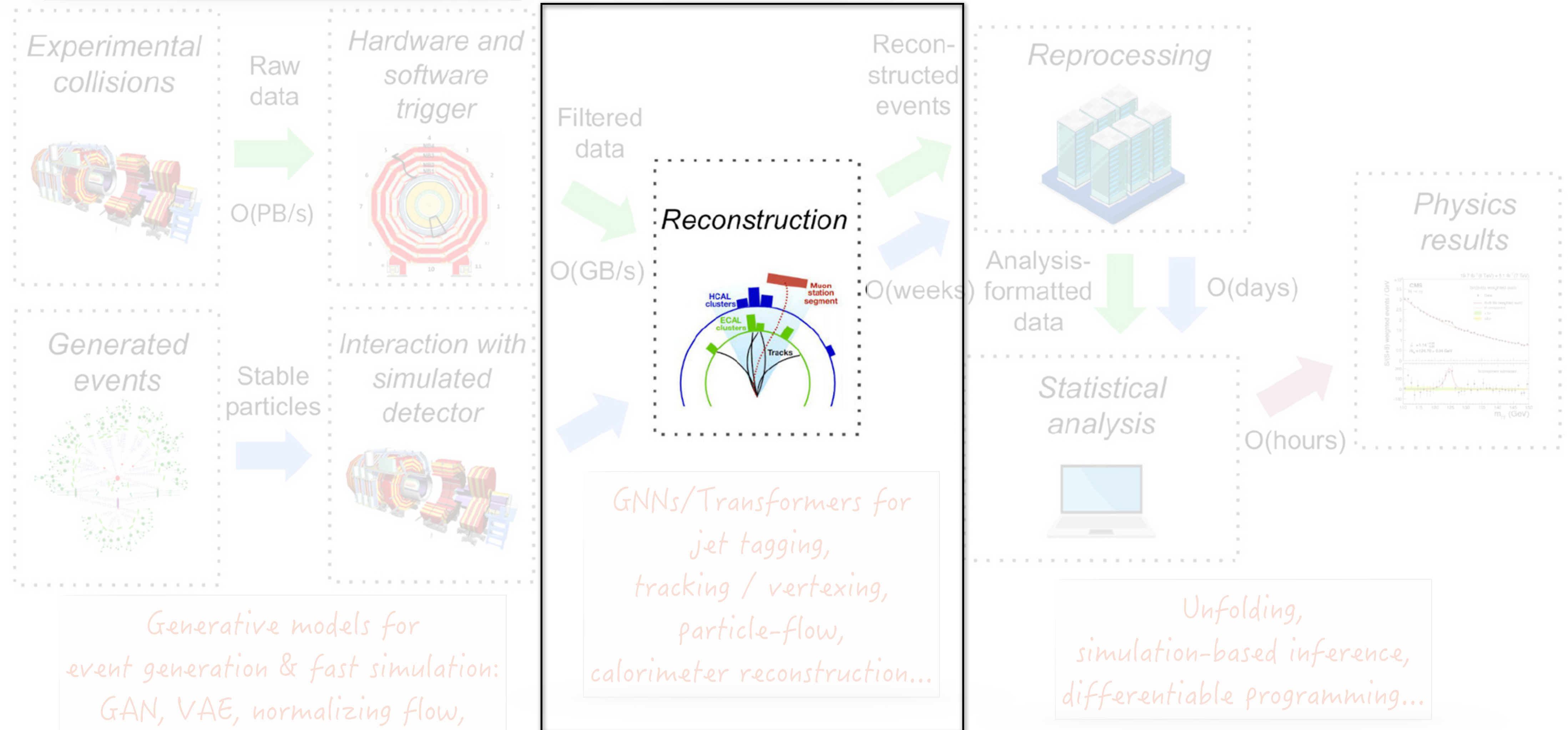


Unfolding, simulation-based inference, differentiable programming...

HEP DATA FLOW... MATCHED WITH ML!

Ultrafast inference (FPGA/ASIC),
data compression, anomaly detection...

Focus of today

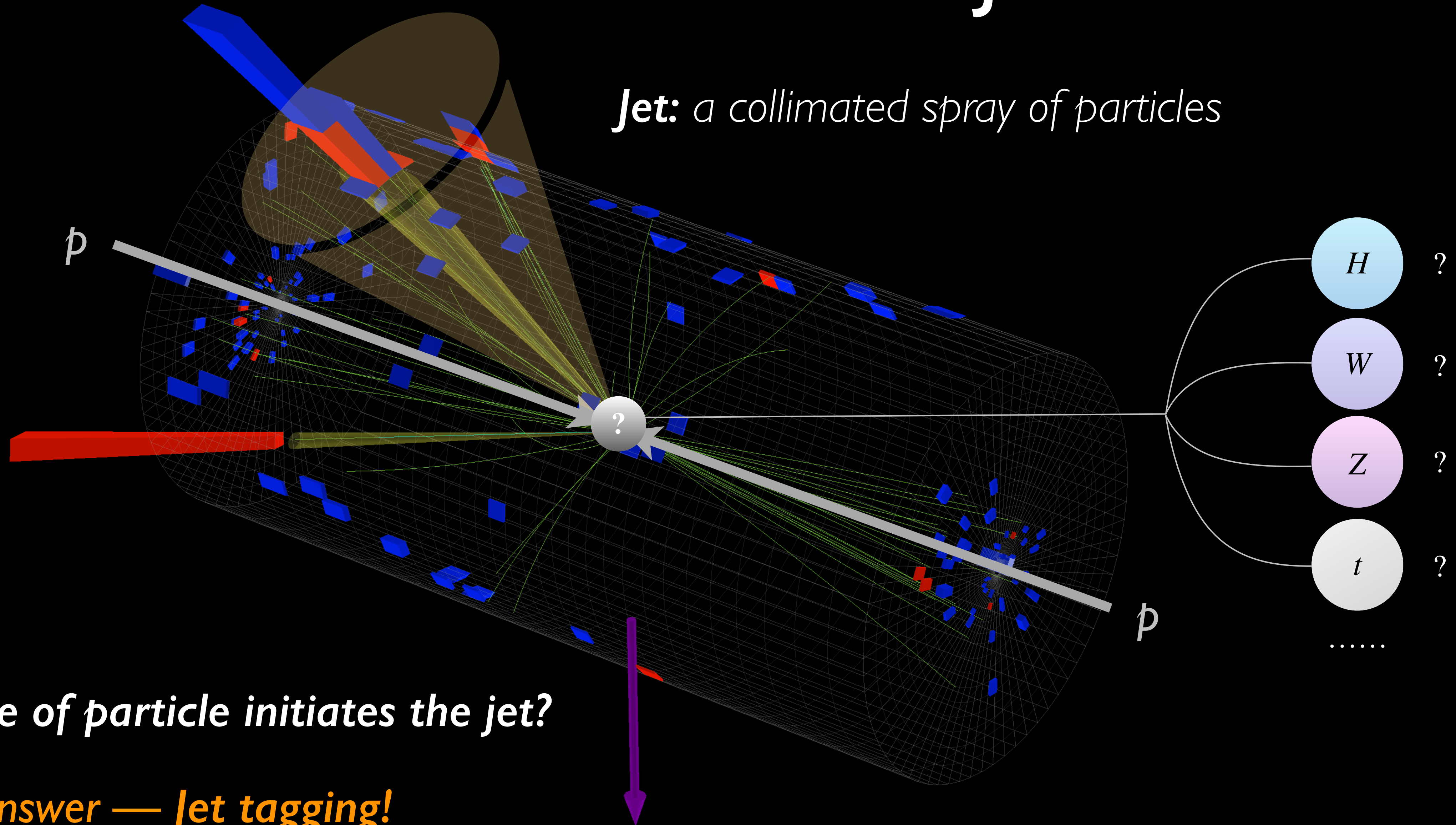




CMS Experiment at LHC, CERN
Data recorded: Sat Aug 5 15:32:22 2017 CEST
Run/Event: 300515 / 205888132

EXAMPLE: JET TAGGING

Jet: a collimated spray of particles

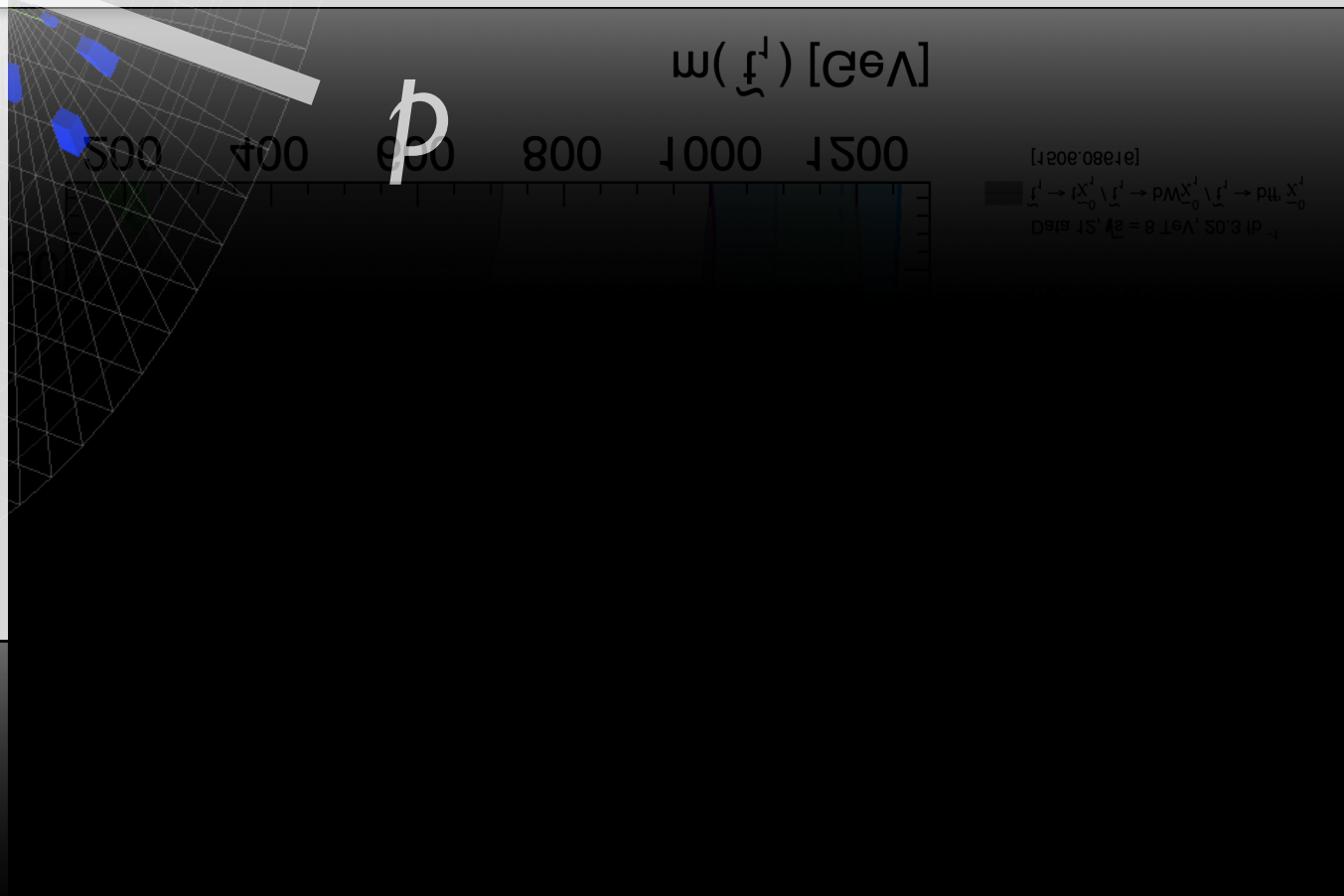
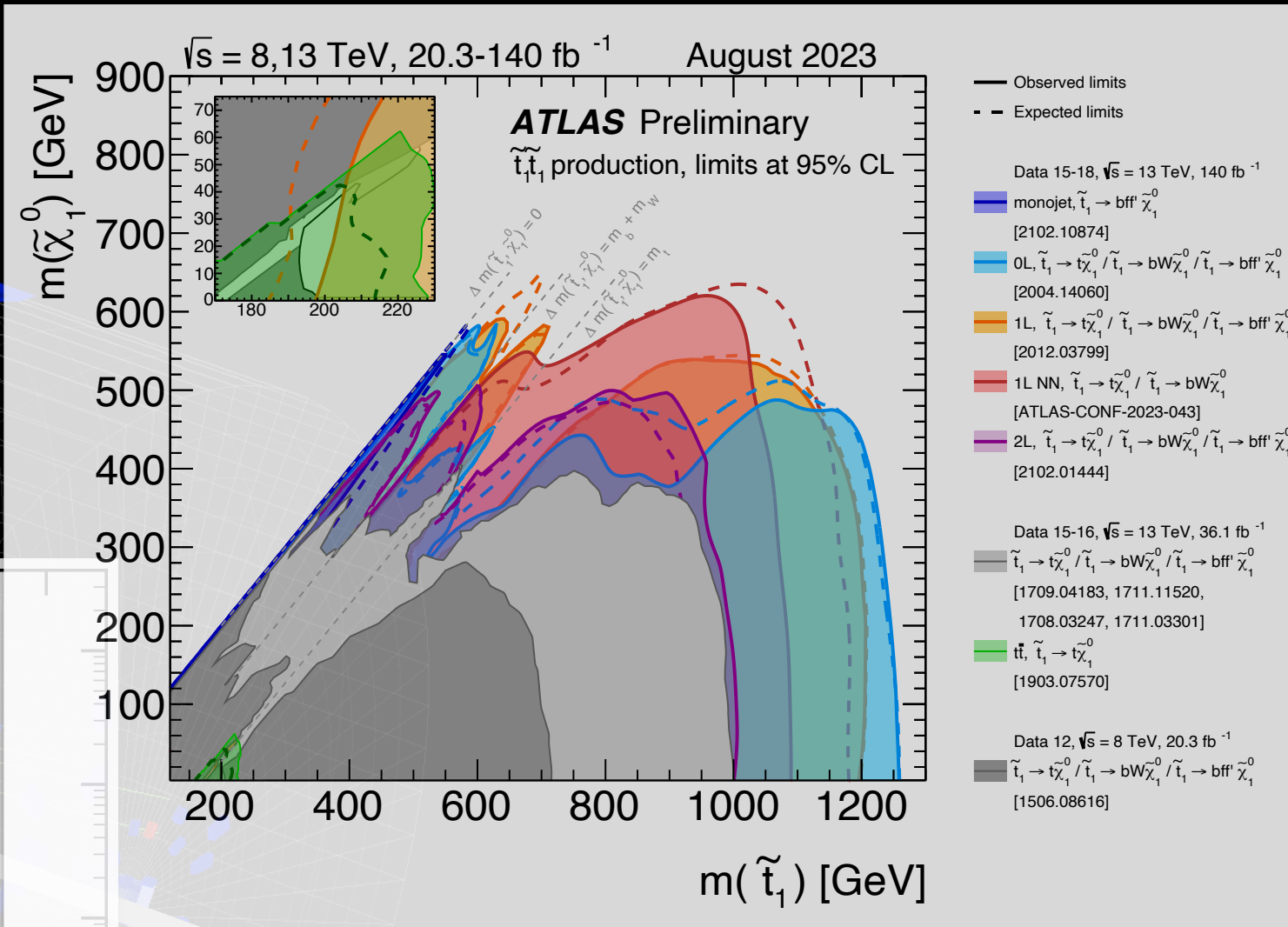
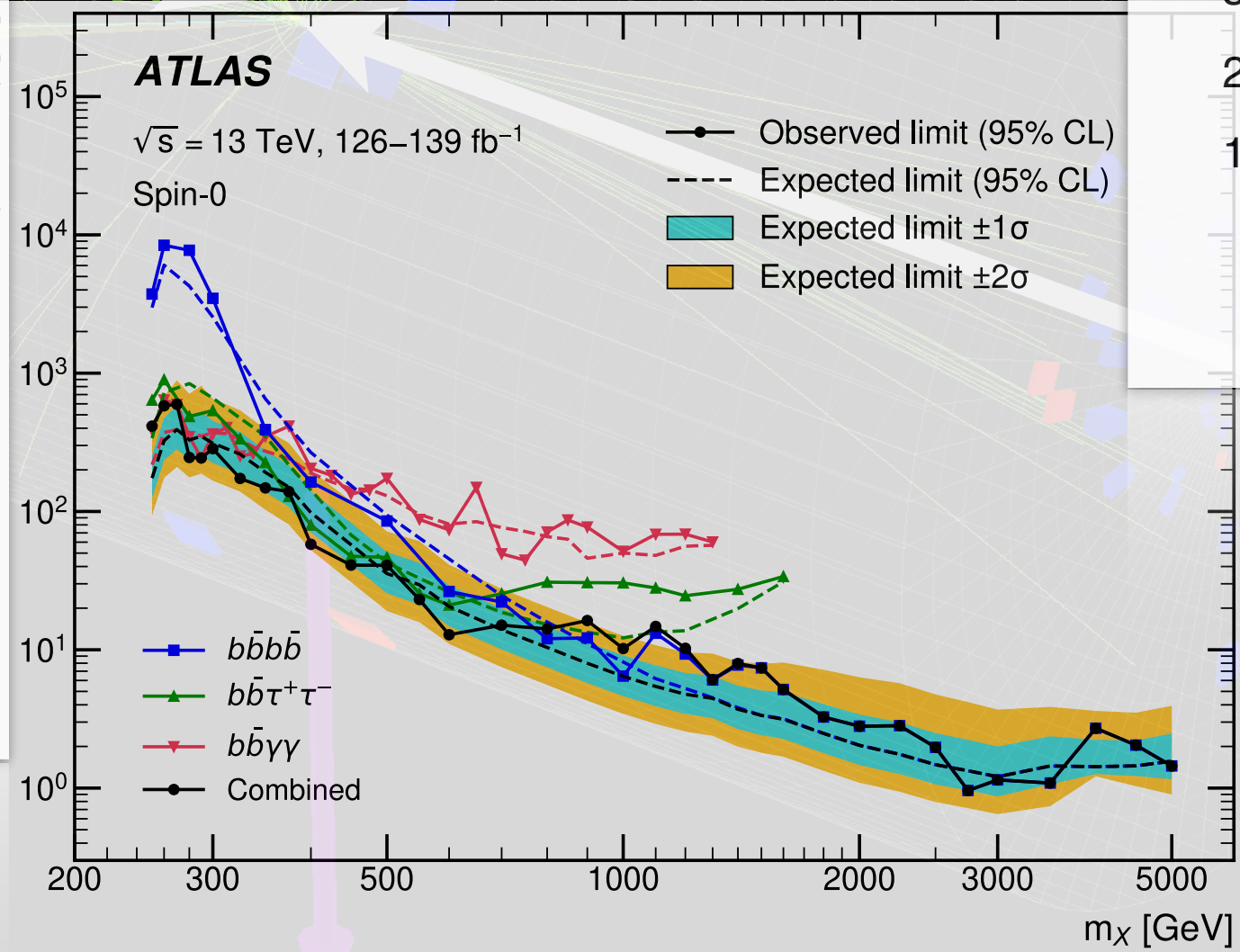
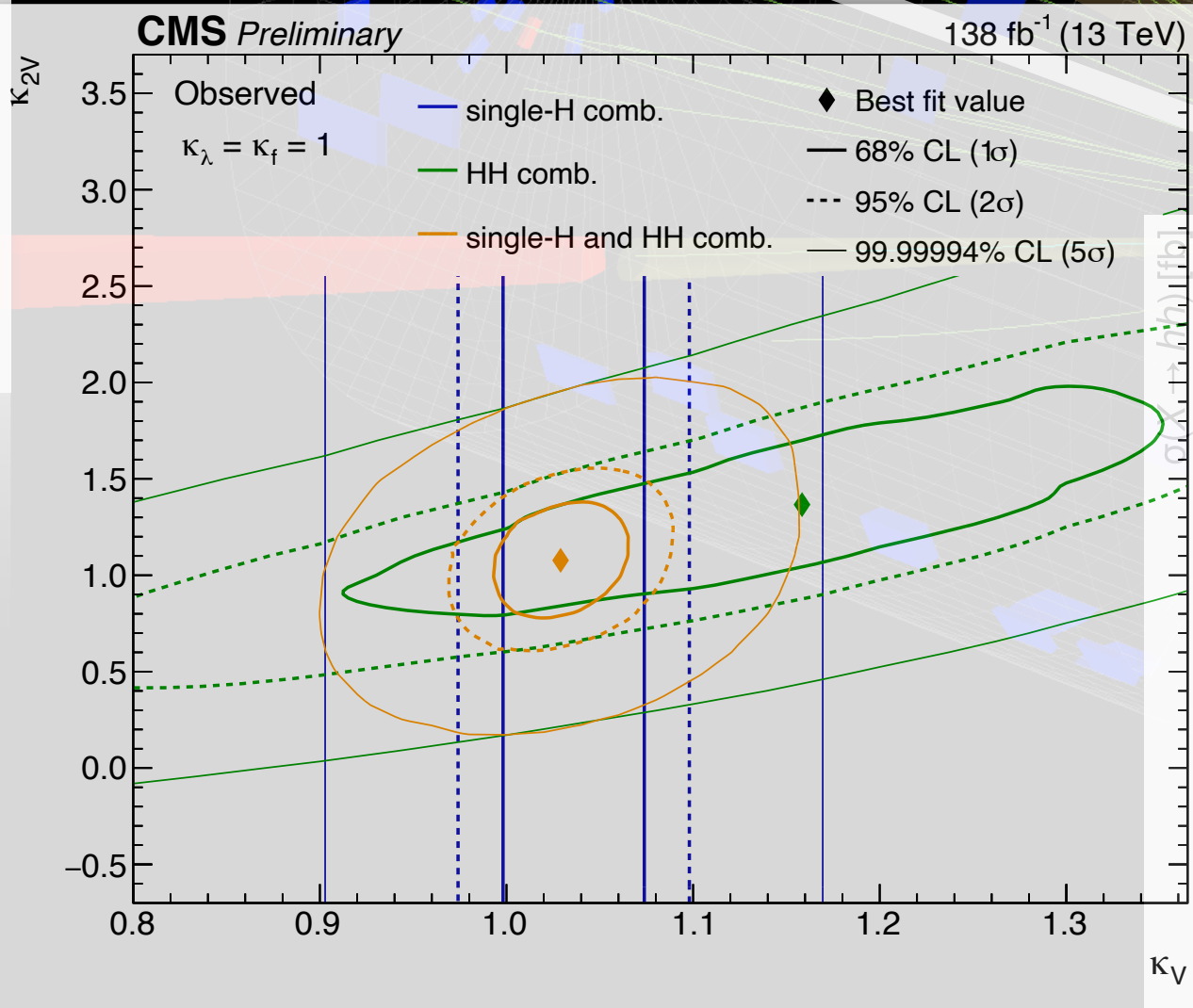
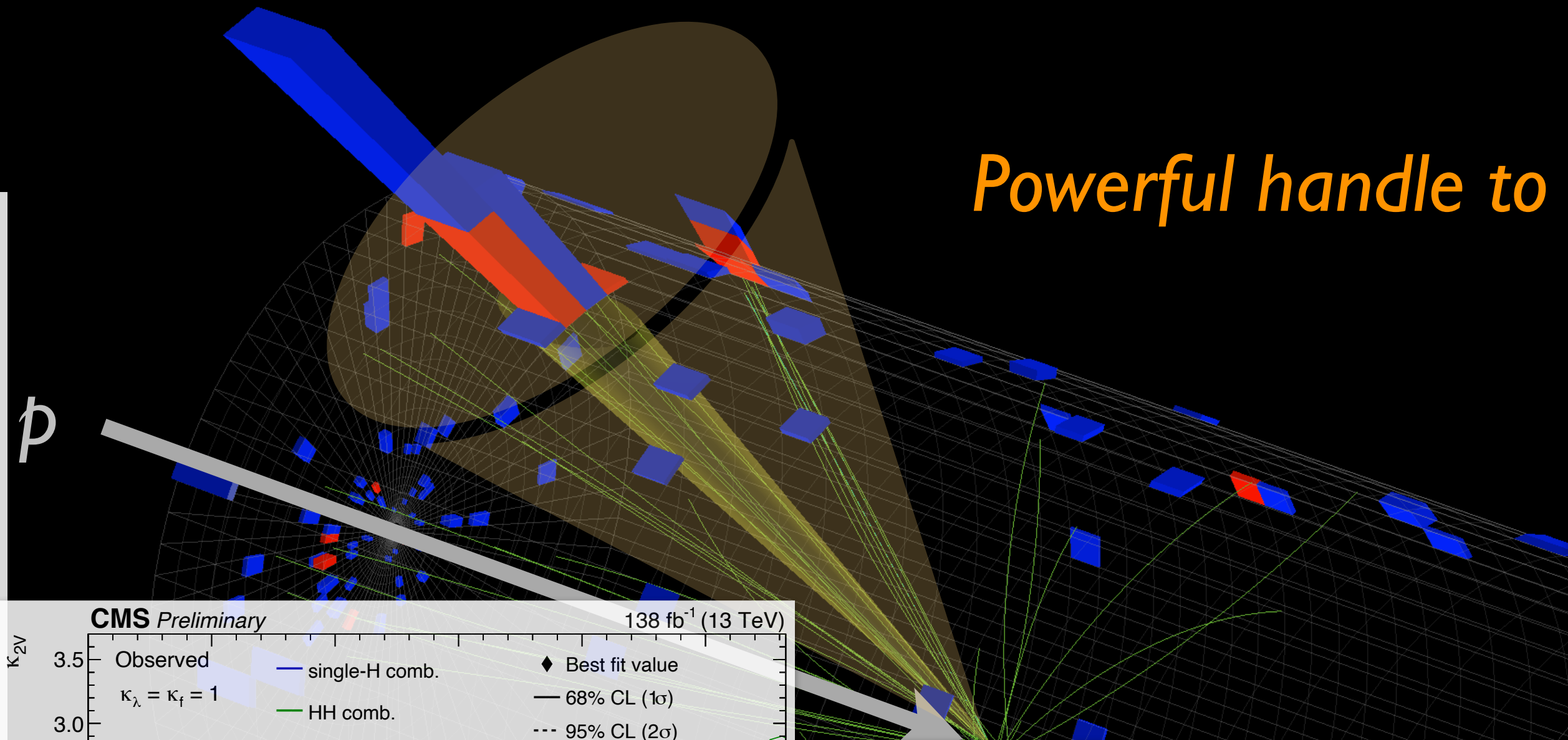
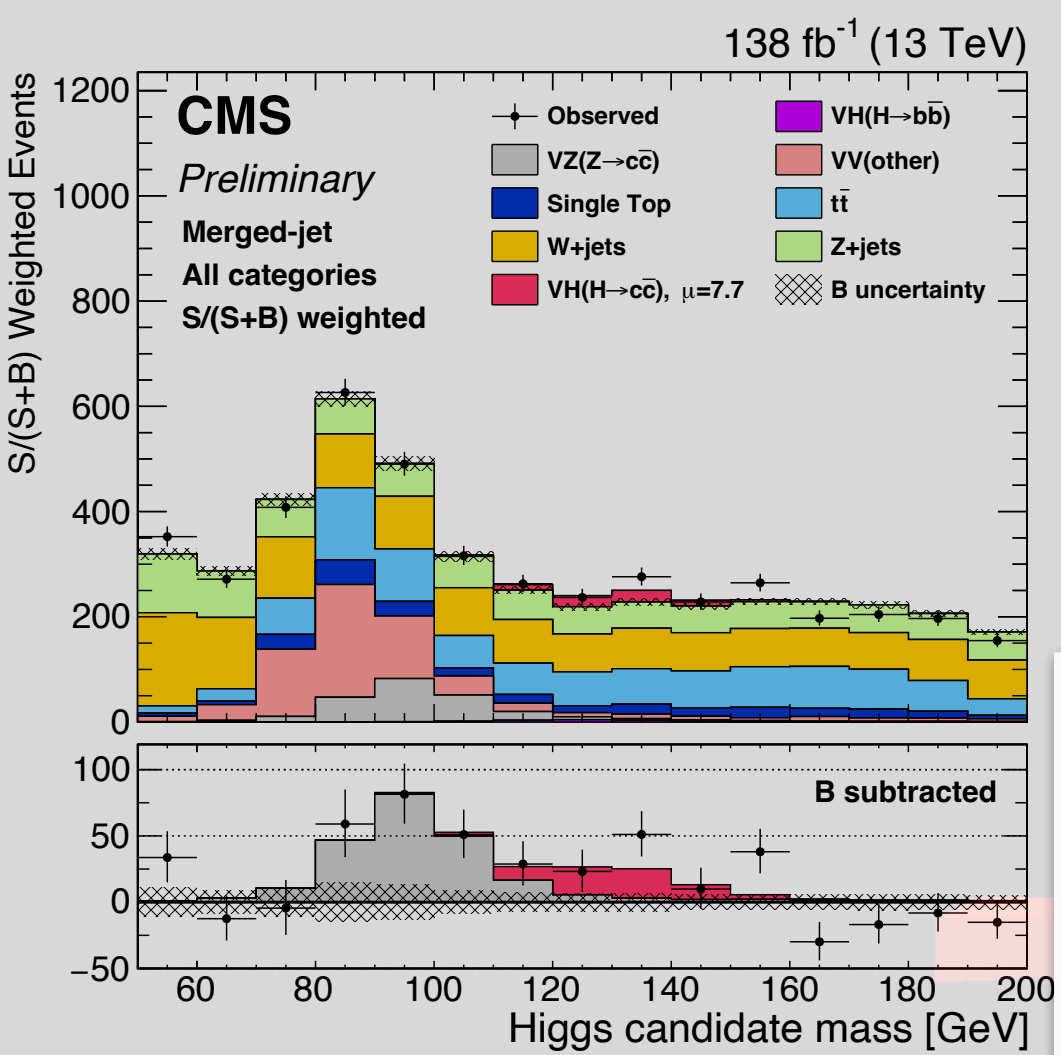




CMS Experiment at LHC, CERN
Data recorded: Sat Aug 5 15:32:22 2017 CEST
Run/Event: 300515 / 205888132

EXAMPLE: JET TAGGING

Powerful handle to search for new phenomena

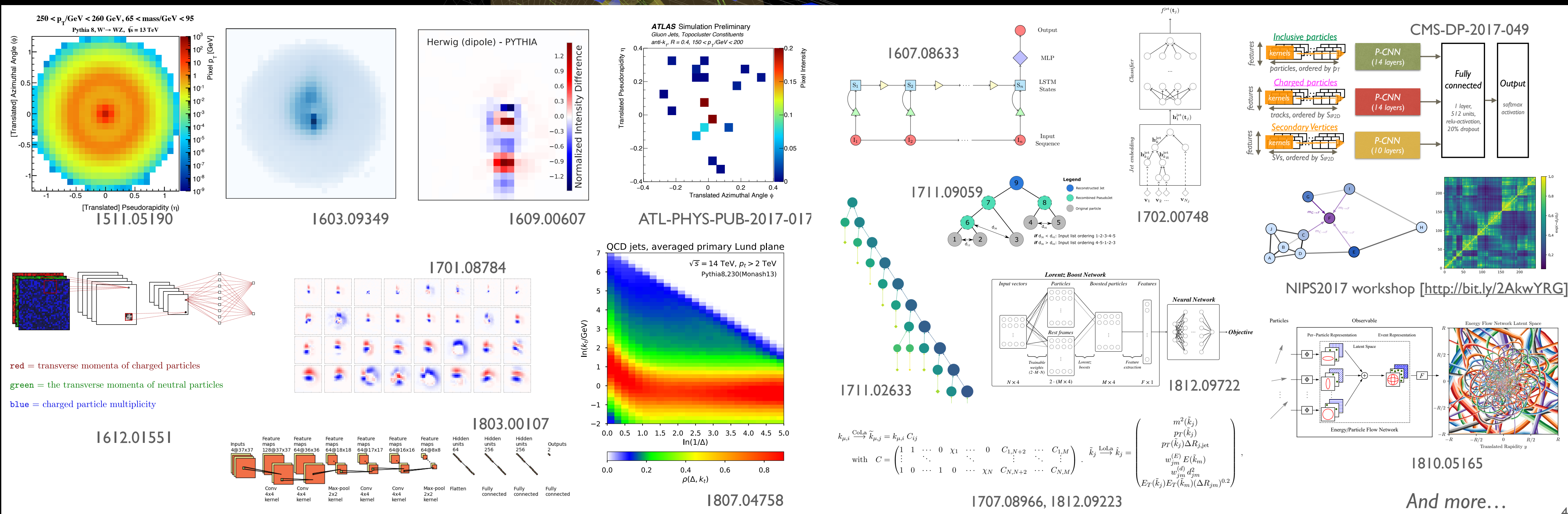




CMS Experiment at LHC, CERN
Data recorded: Sat Aug 5 15:32:22 2017 CEST
Run/Event: 300515 / 205888132

EXAMPLE: JET TAGGING

But also: excellent playground for machine learning

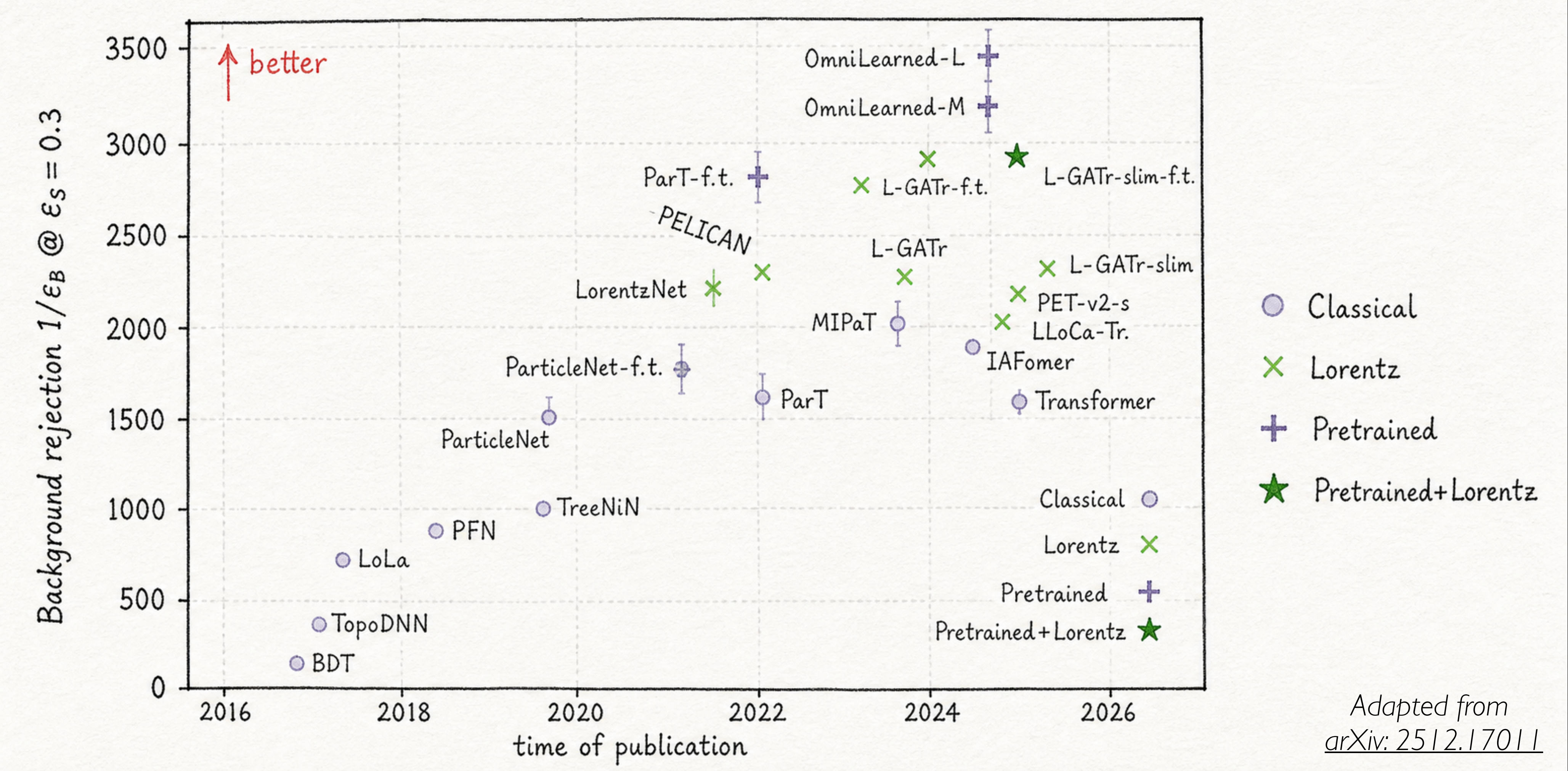




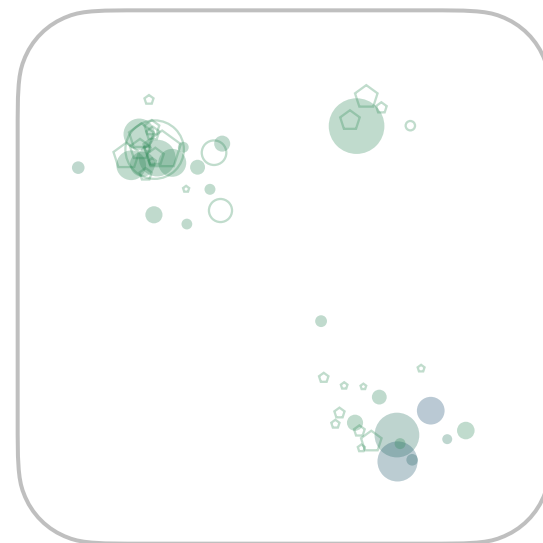
CMS Experiment at LHC, CERN
Data recorded: Sat Aug 5 15:32:22 2017 CEST
Run/Event: 300515 / 205888132

EXAMPLE: JET TAGGING

But also: excellent playground for machine learning

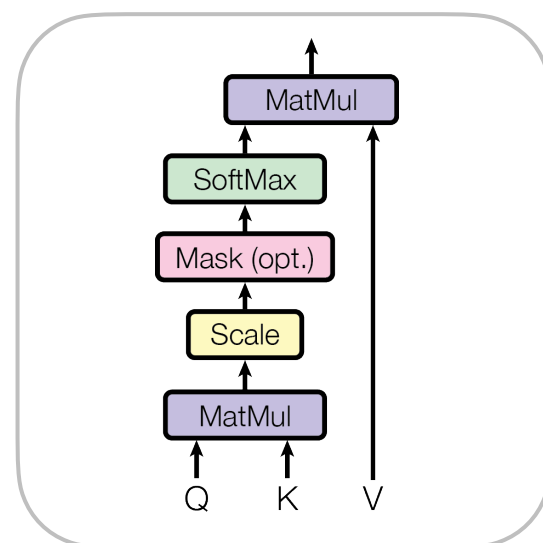


OUTLINE



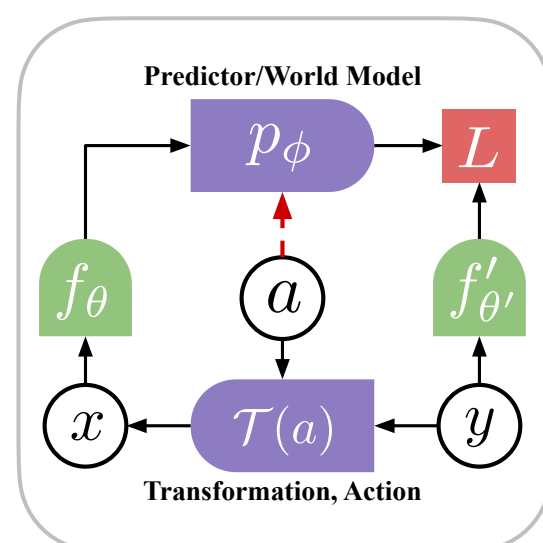
Evolution of Jet Representation

image, sequence, point cloud, ...



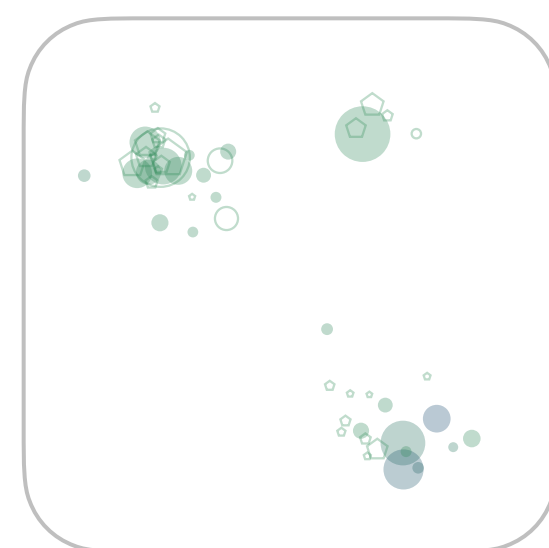
Attention Is All You Need?

or how physics can help...



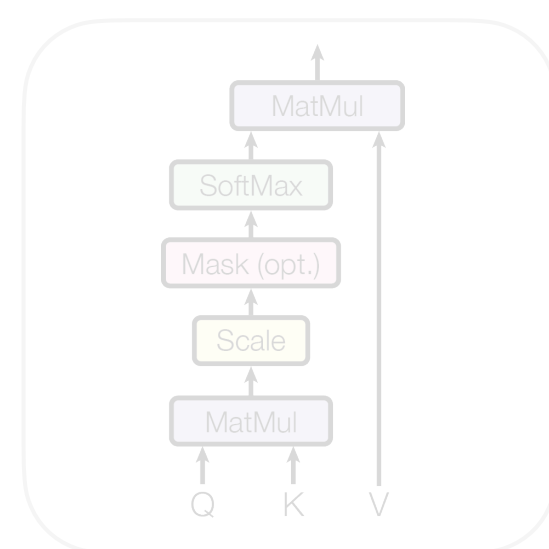
Towards a Foundation Model

scaling, self-supervision, and more...



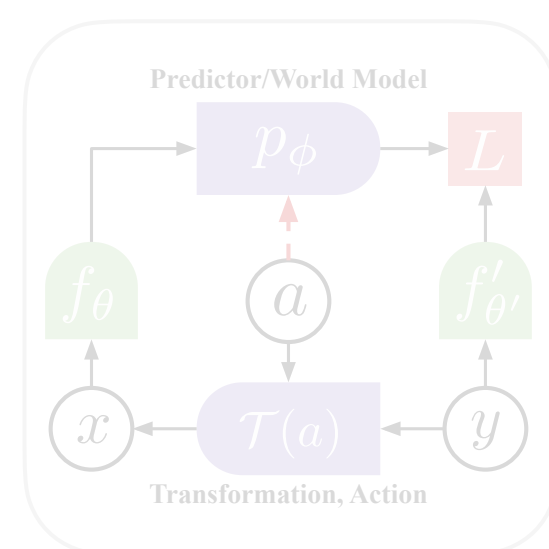
Evolution of Jet Representation

image, sequence, point cloud, ...



Attention Is All You Need?

or how physics can help...

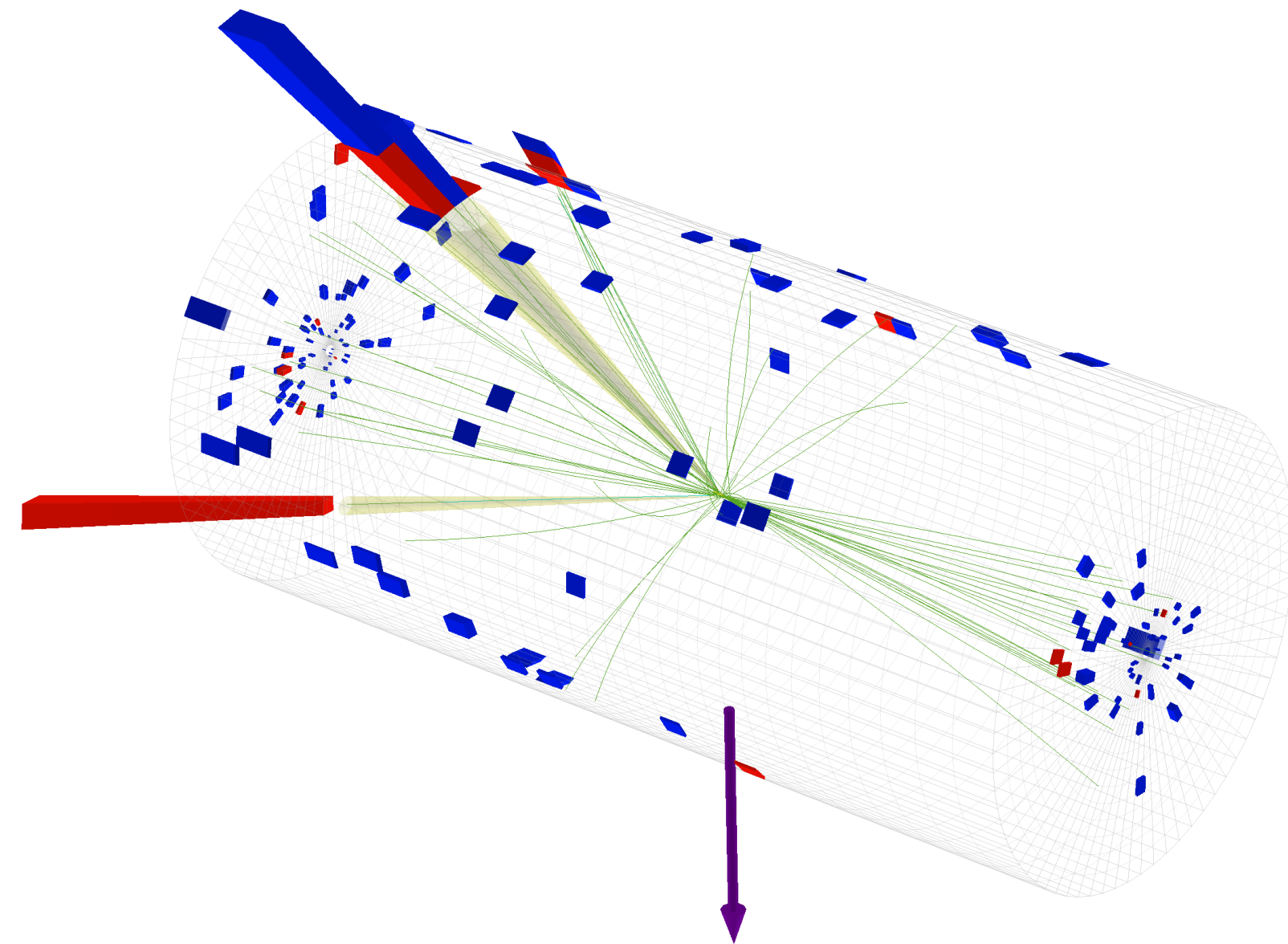


Towards a Foundation Model

scaling, self-supervision, and more...

DATA REPRESENTATION

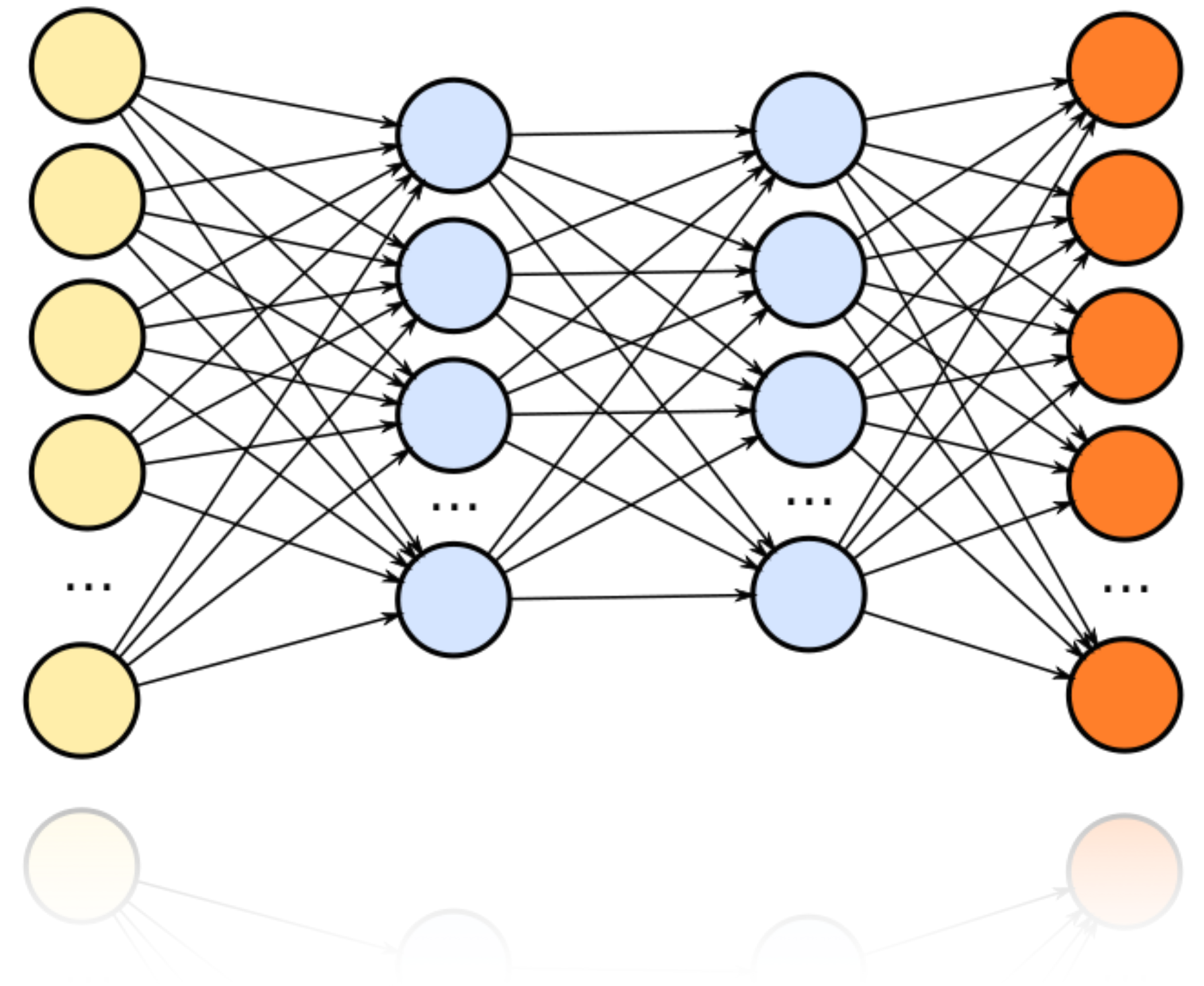
HEP



Collision events, detector hits, sensor arrays, ...

×

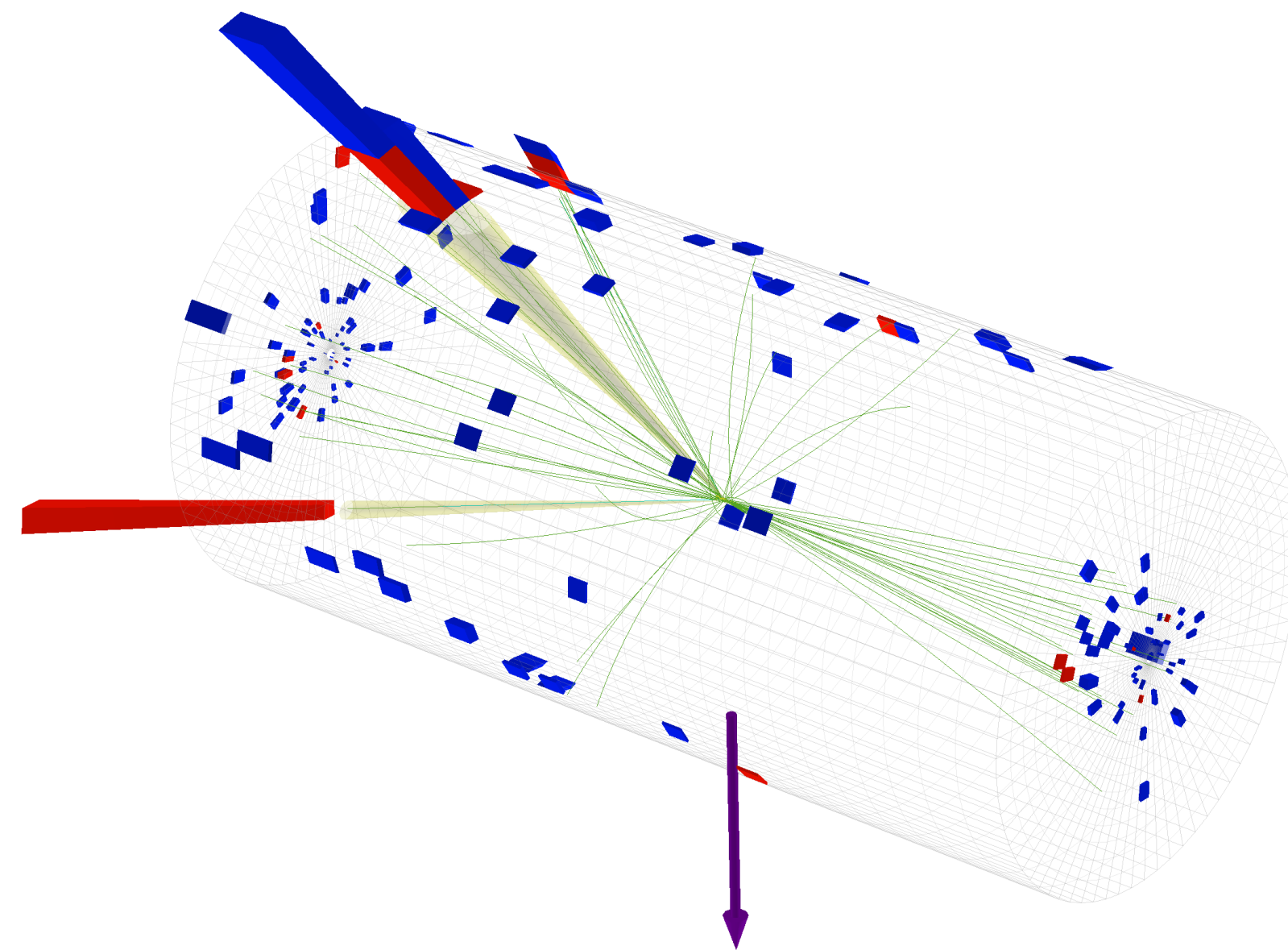
ML



First and foremost:
How to represent the data?

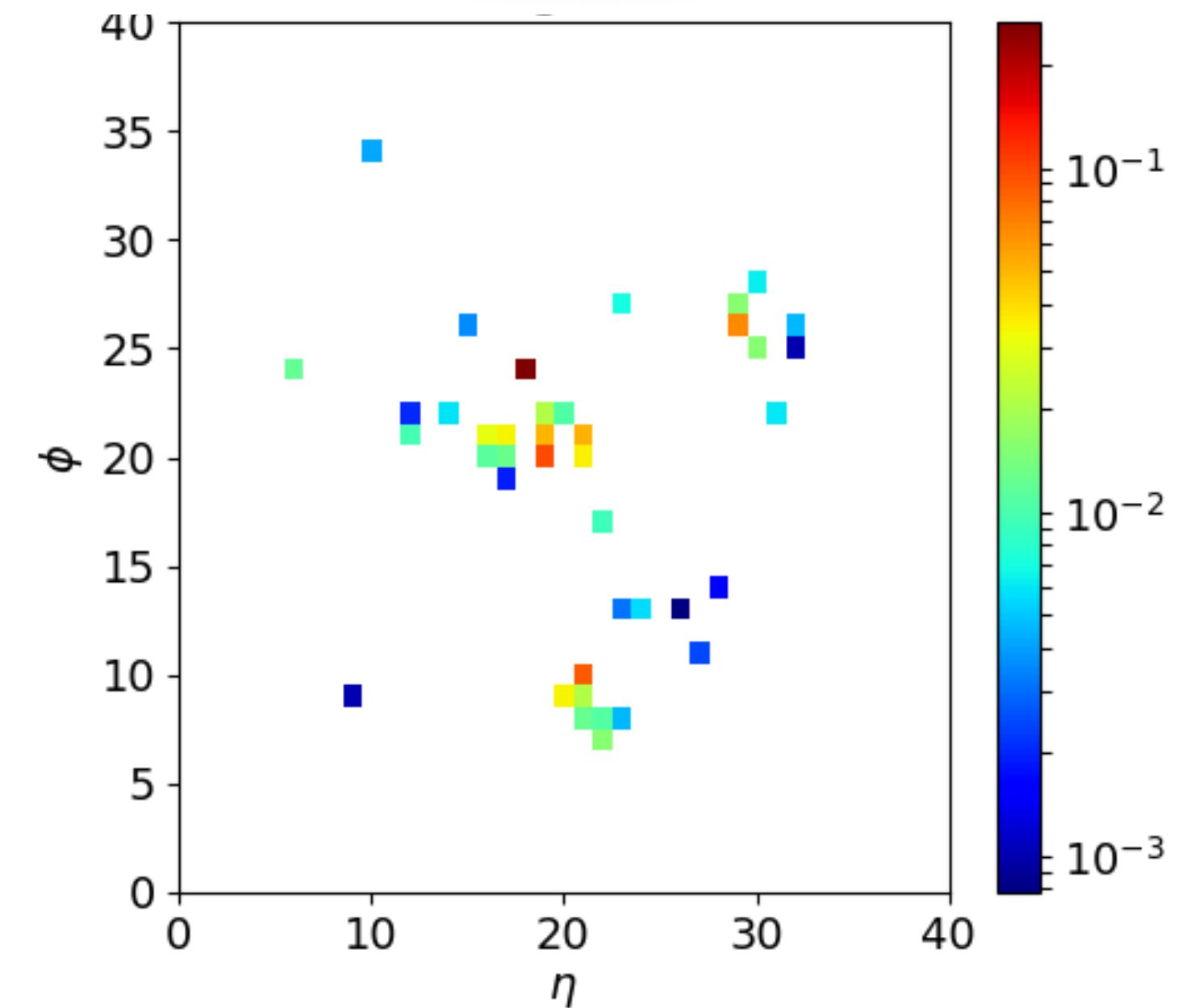
DATA REPRESENTATION: IMAGE

HEP



Collision events, detector hits, sensor arrays, ...

Image

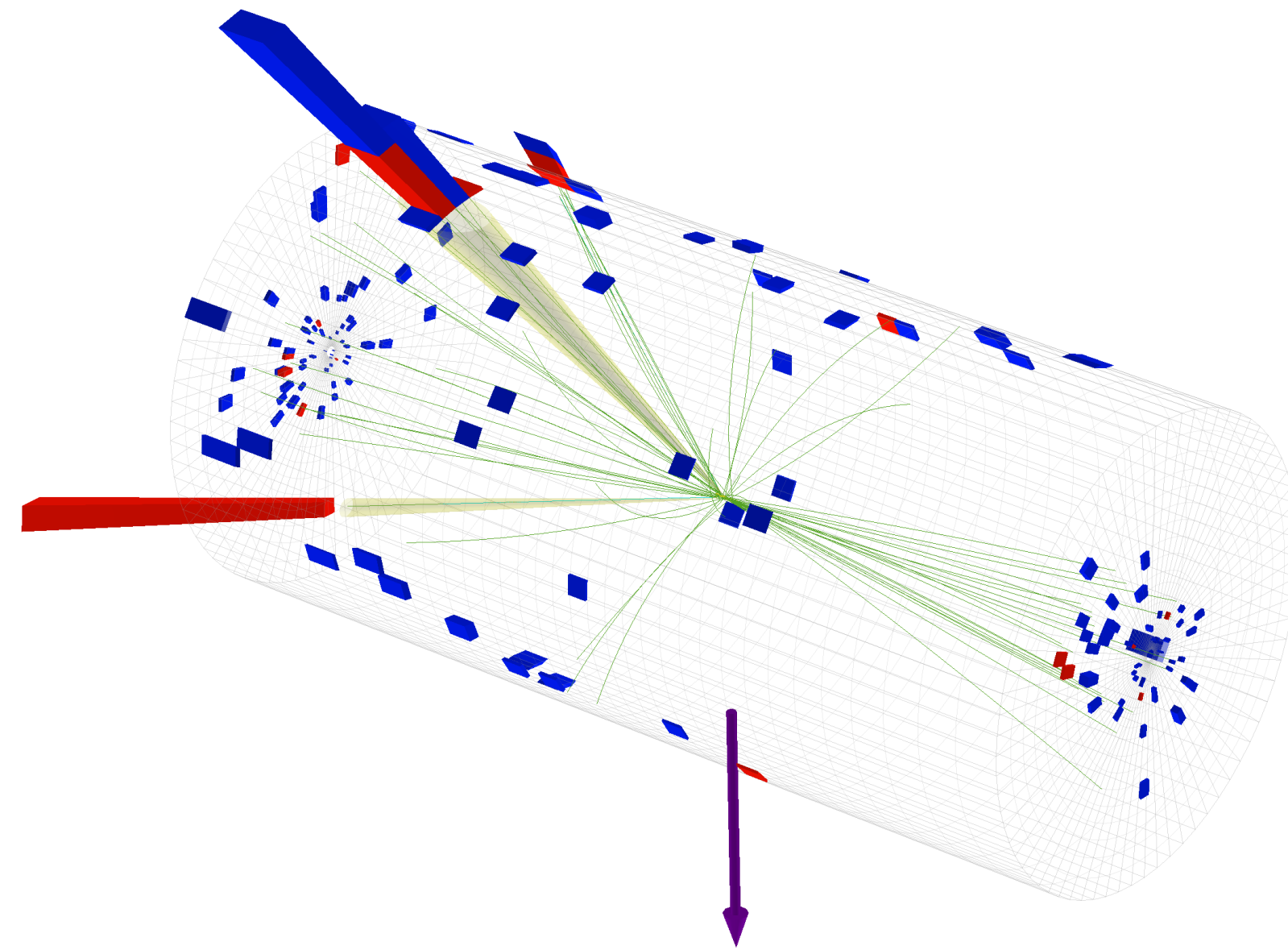


e.g., L. de Oliveira, M. Kagan, L. Mackey, B. Nachman and A. Schwartzman, arXiv: 1511.05190
(see also the review by Kagan, arXiv:2012.09719)

- Convert to 2D/3D image => **Computer vision**
 - then use convolutional neural networks (CNNs)
 - but:
 - inhomogeneous geometry, high sparsity, ...

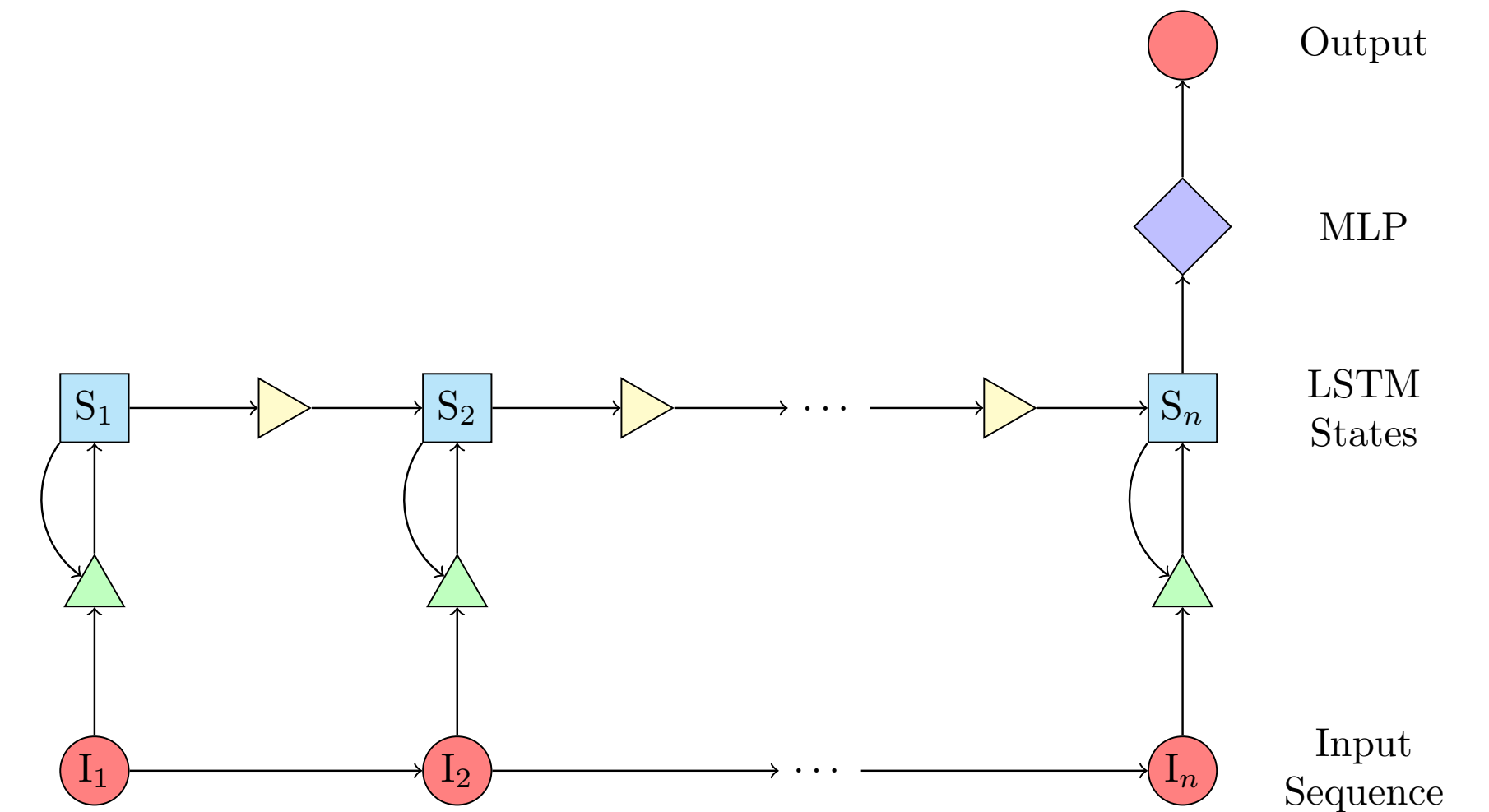
DATA REPRESENTATION: SEQUENCE

HEP



Collision events, detector hits, sensor arrays, ...

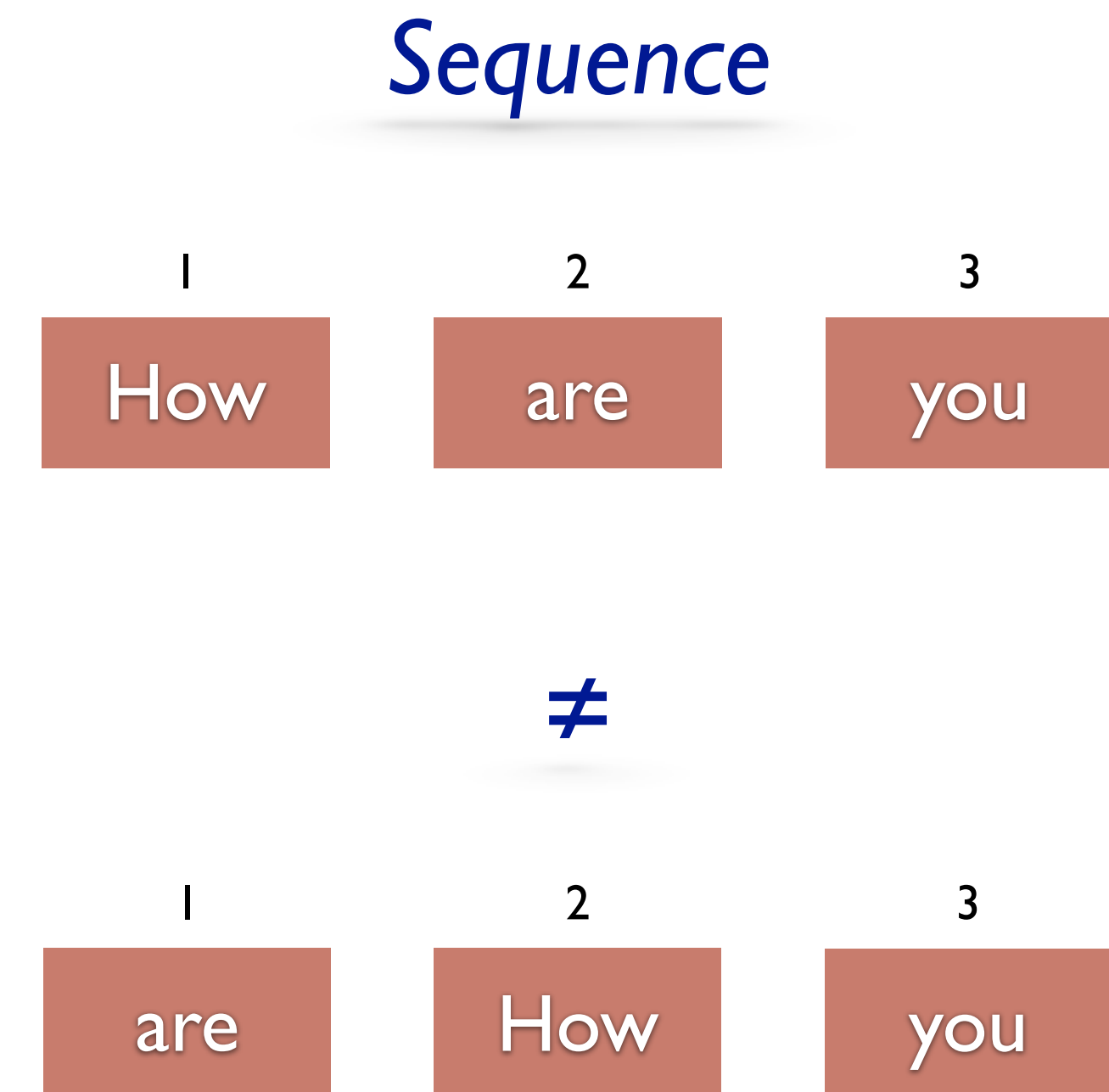
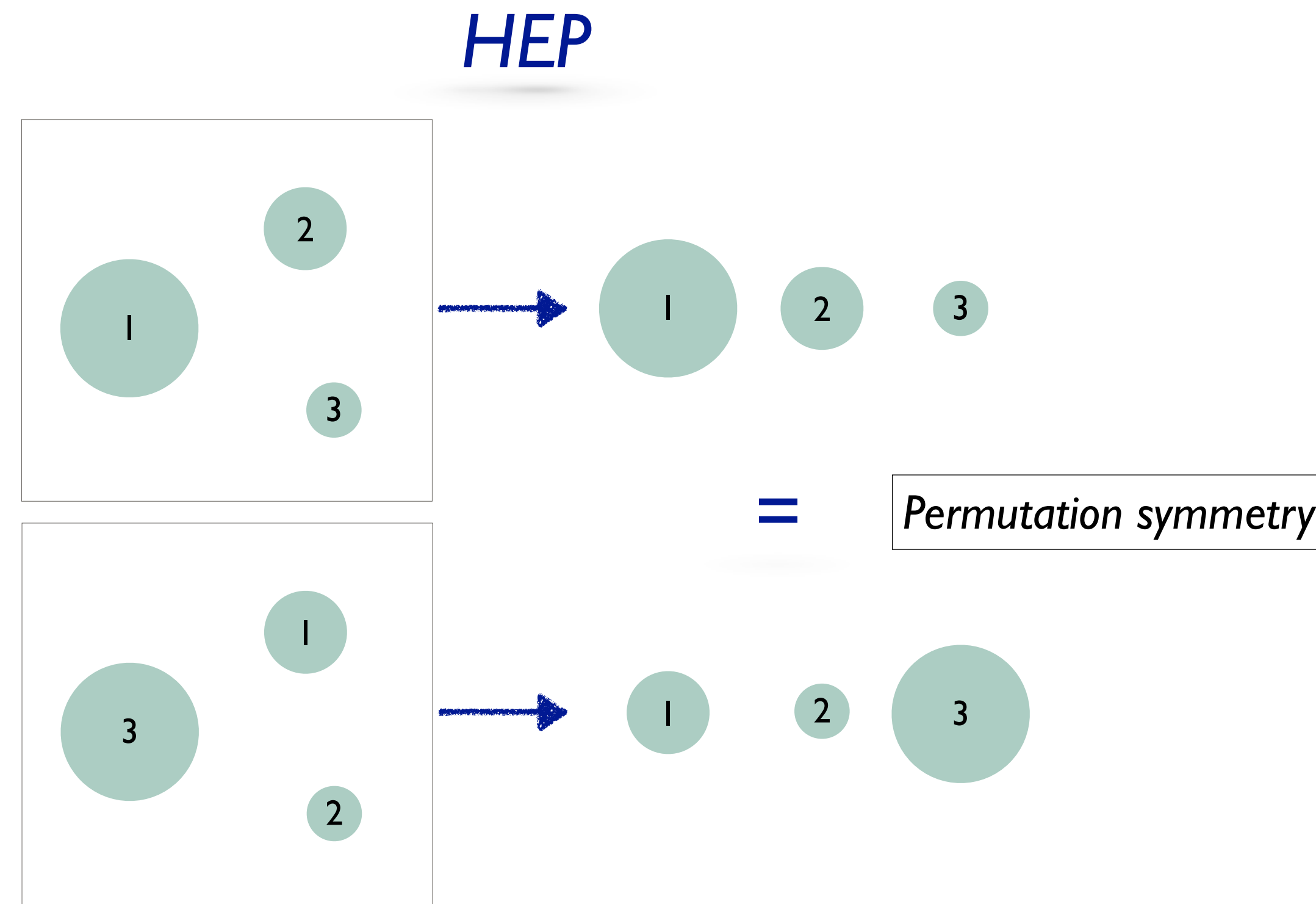
Sequence



e.g., Guest, Collado, Baldi, Hsu, Urban, Whiteson
arXiv: 1607.08633

- Convert to a sequence => **Natural language processing (NLP)**
 - recurrent neural network (RNN), e.g., GRU/LSTM; 1D CNNs; etc.

DATA REPRESENTATION: SEQUENCE?



- Convert to a sequence => **Natural language processing (NLP)**
 - recurrent neural network (RNN), e.g., GRU/LSTM; 1D CNNs; etc.
 - but:
 - must impose an **ordering** on the particles/hits, which can limit the learning performance

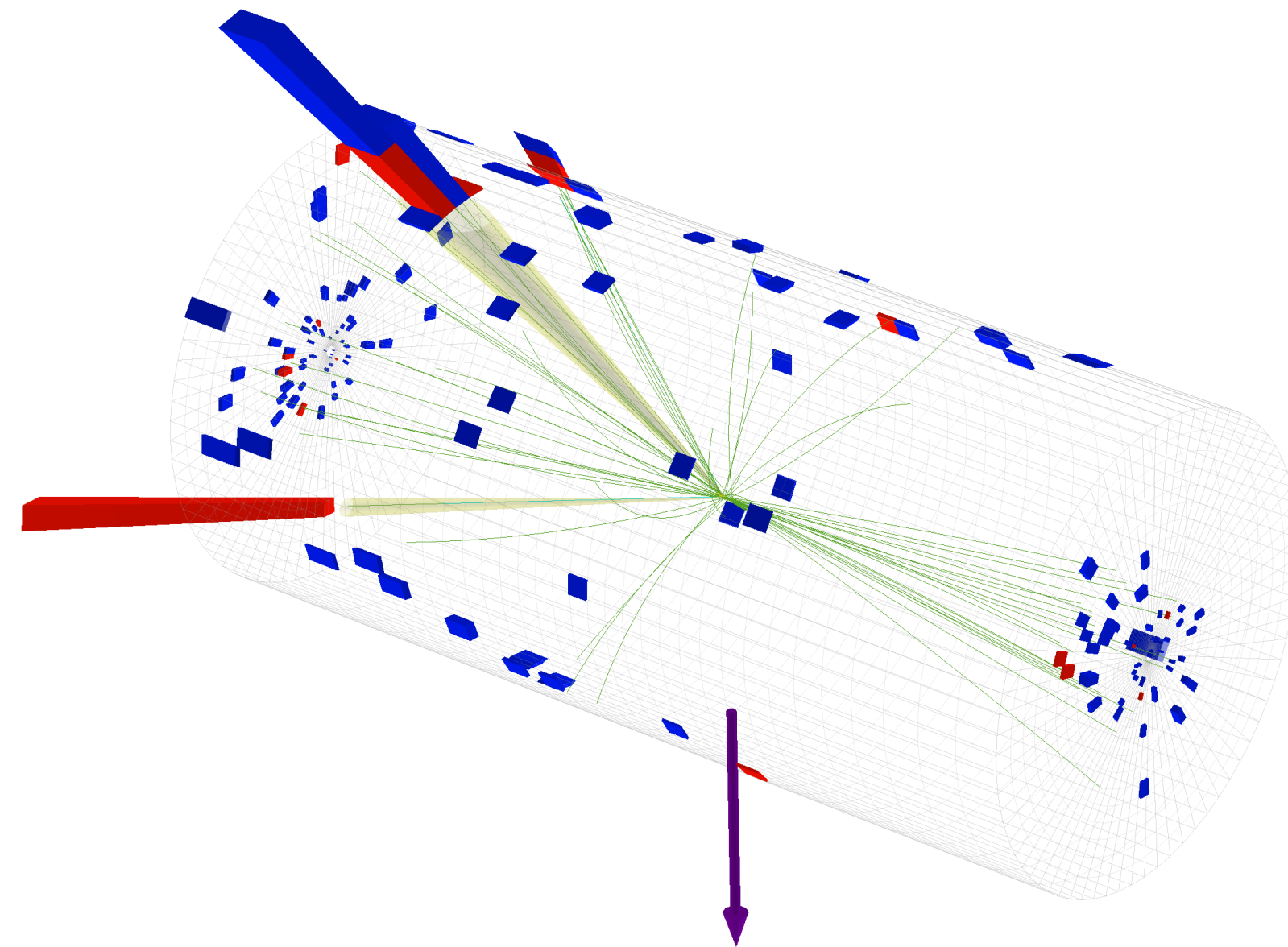
POINT CLOUD



*An unordered set of points in space
(e.g., produced by a LiDAR on self-driving cars)*

DATA REPRESENTATION: POINT CLOUD

HEP



Collision events, detector hits, sensor arrays, ...



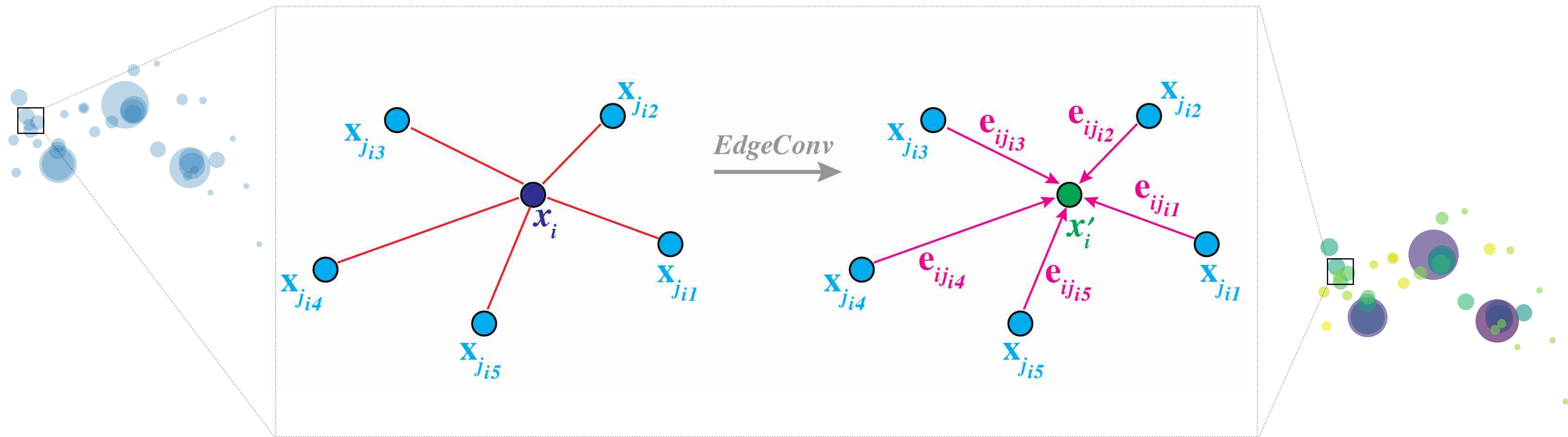
Point cloud



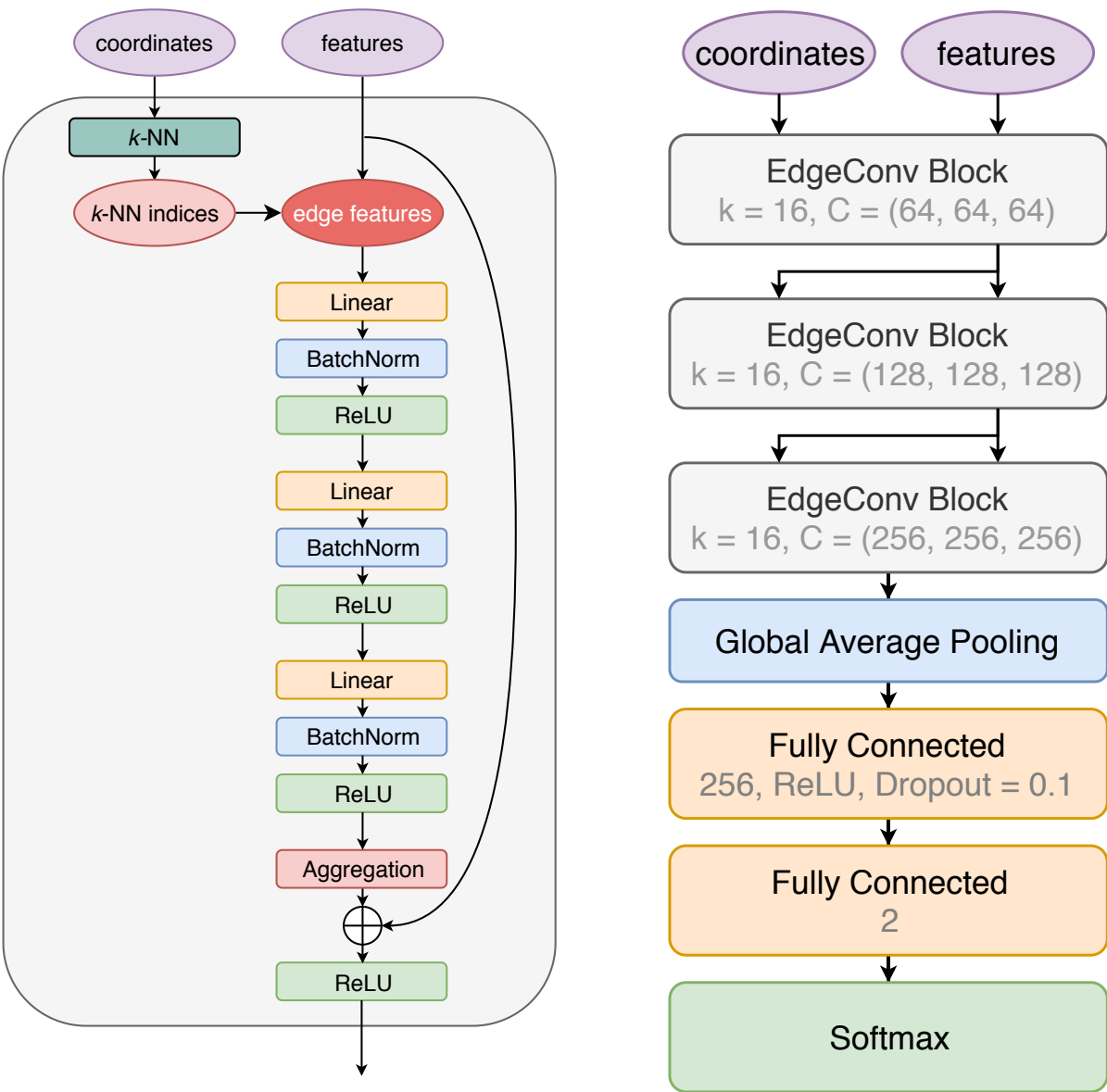
- HEP data as a point cloud
 - each particle / detector cell is a point in the cloud
 - for each point: (spatial) coordinates + any additional properties (energy/momentum, detector response, ...)
 - key feature: ***permutation symmetry*** – *algorithm output should not depend on the input ordering*

PARTICLENET

- ParticleNet: jet tagging via particle clouds
 - treating a jet as an **unordered set of particles**, distributed in the $\eta - \phi$ space
 - **graph neural network architecture**, adapted from DGCNN [arXiv:1801.07829]
 - treating a point cloud as a graph: each point is a vertex
 - for each point, a local patch is defined by finding its k-nearest neighbors
 - designing a permutation-invariant “convolution” function
 - define “edge feature” for each center-neighbor pair: $e_{ij} = \text{MLP}(x_i, x_j)$
 - aggregate the edge features in a symmetric way: $x'_i = \text{mean}_j e_{ij}$



HQ and L. Gouskos
Phys.Rev.D 101 (2020) 5, 056019



Performance on top quark tagging benchmark
[SciPost Phys. 7, 014 (2019)]

	$1/\epsilon_b$ at $\epsilon_s = 30\%$
ResNeXt-50	1147 ± 58
P-CNN	759 ± 24
PFN	888 ± 17
ParticleNet-Lite	1262 ± 49
ParticleNet	1615 ± 93

PARTICLENET

HQ and L. Gouskos
Phys.Rev.D 101 (2020) 5, 056019

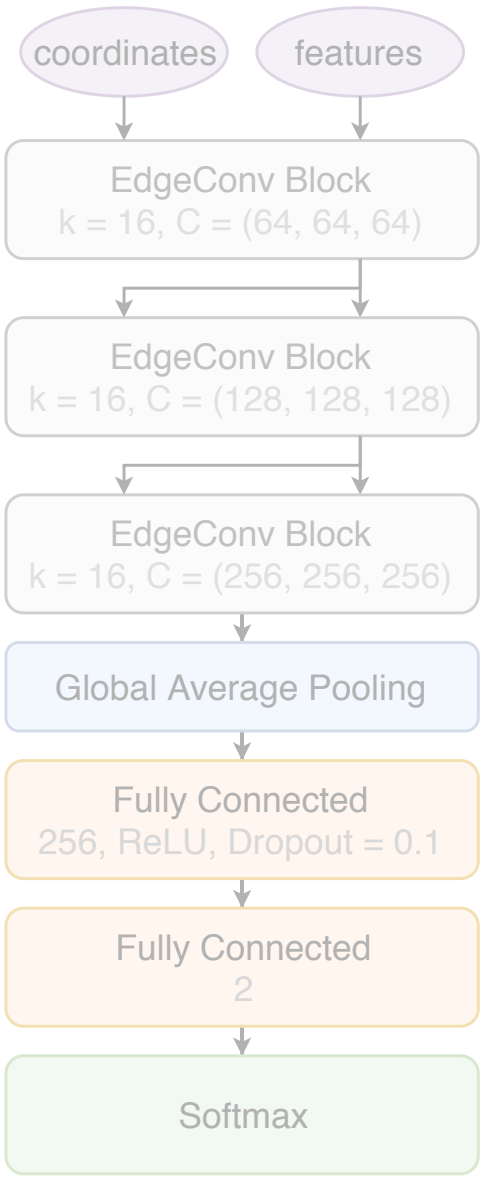
- ParticleNet: jet tagging via particle clouds

- treating a jet as an **unordered**
- graph neural network** architecture
 - treating a point cloud as a graph
 - for each point, a local patch is extracted
 - designing a permutation-invariant pooling operation
 - defined as “max over neighbors”
 - applied to the local patch

Performance on top quark tagging benchmark
[SciPost Phys. 7, 014 (2019)]

$1/\epsilon_b$ at $\epsilon_s = 30\%$

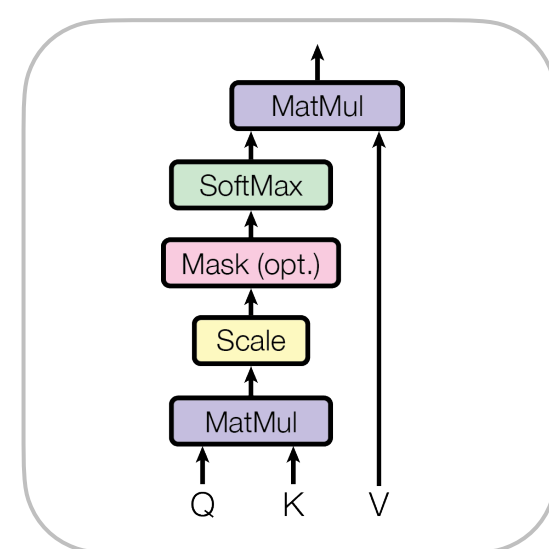
ResNeXt-50	1147 ± 58
P-CNN	759 ± 24
PFN	888 ± 17
ParticleNet-Lite	1262 ± 49
ParticleNet	1615 ± 93





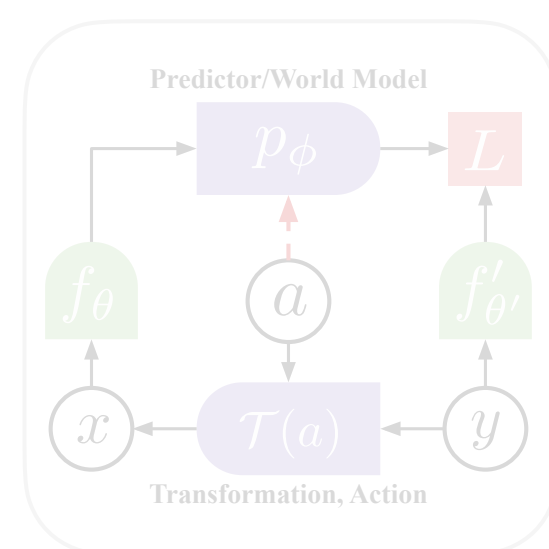
Evolution of Jet Representation

image, sequence, point cloud, ...



Attention Is All You Need?

or how physics can help...



Towards a Foundation Model

scaling, self-supervision, and more...

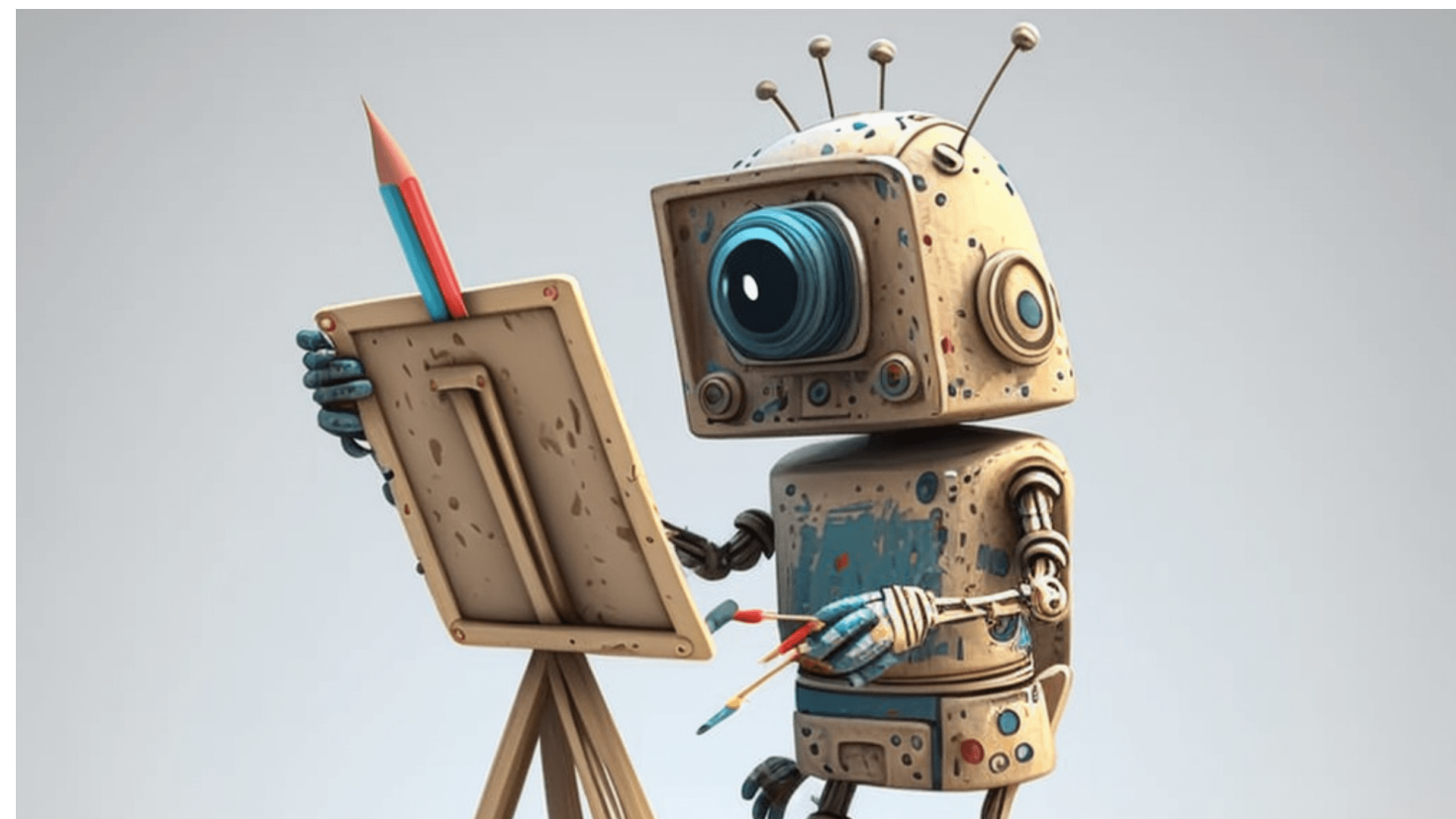
ATTENTION IS ALL YOU NEED?

- Transformers are taking over every task

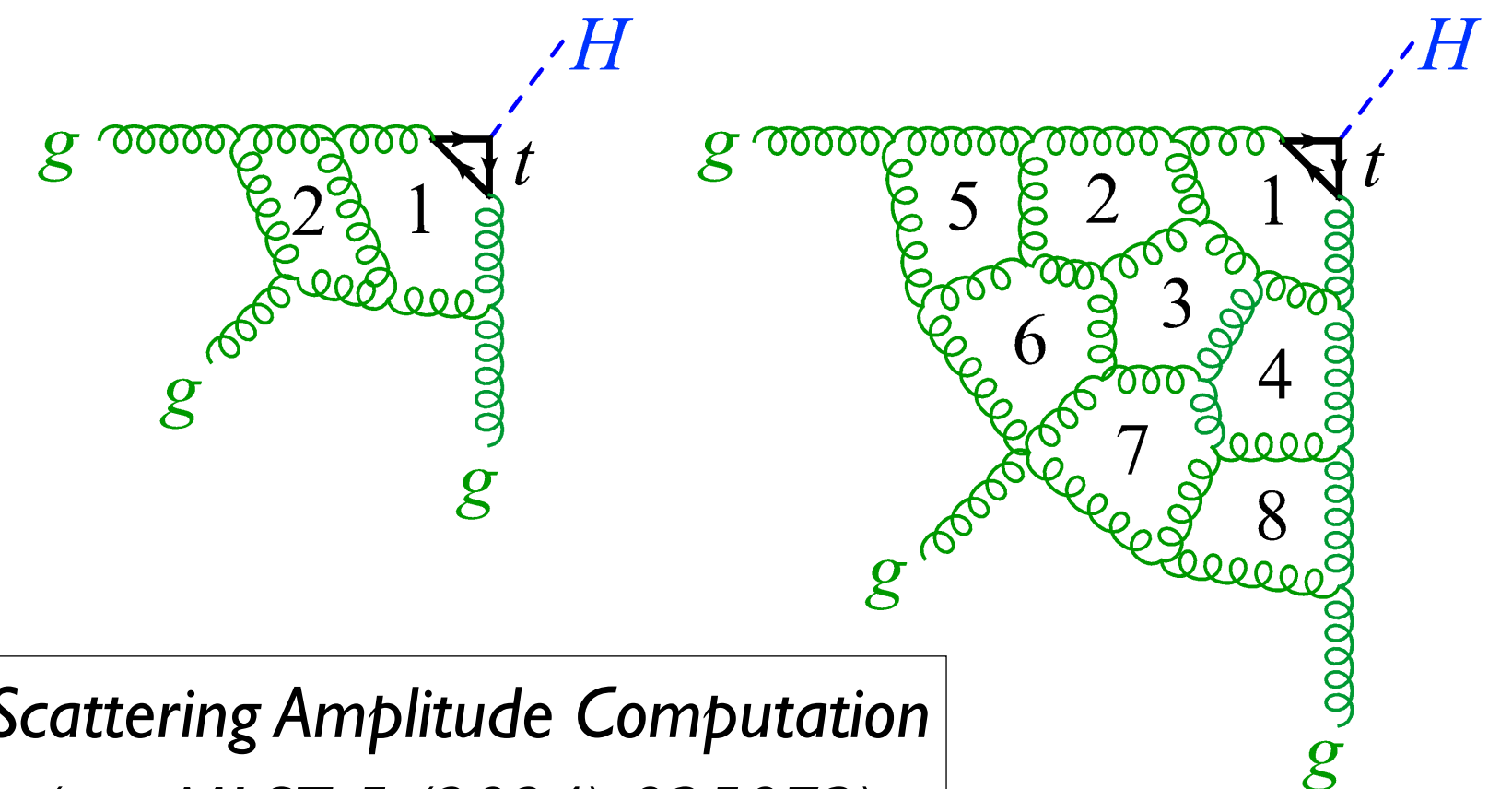
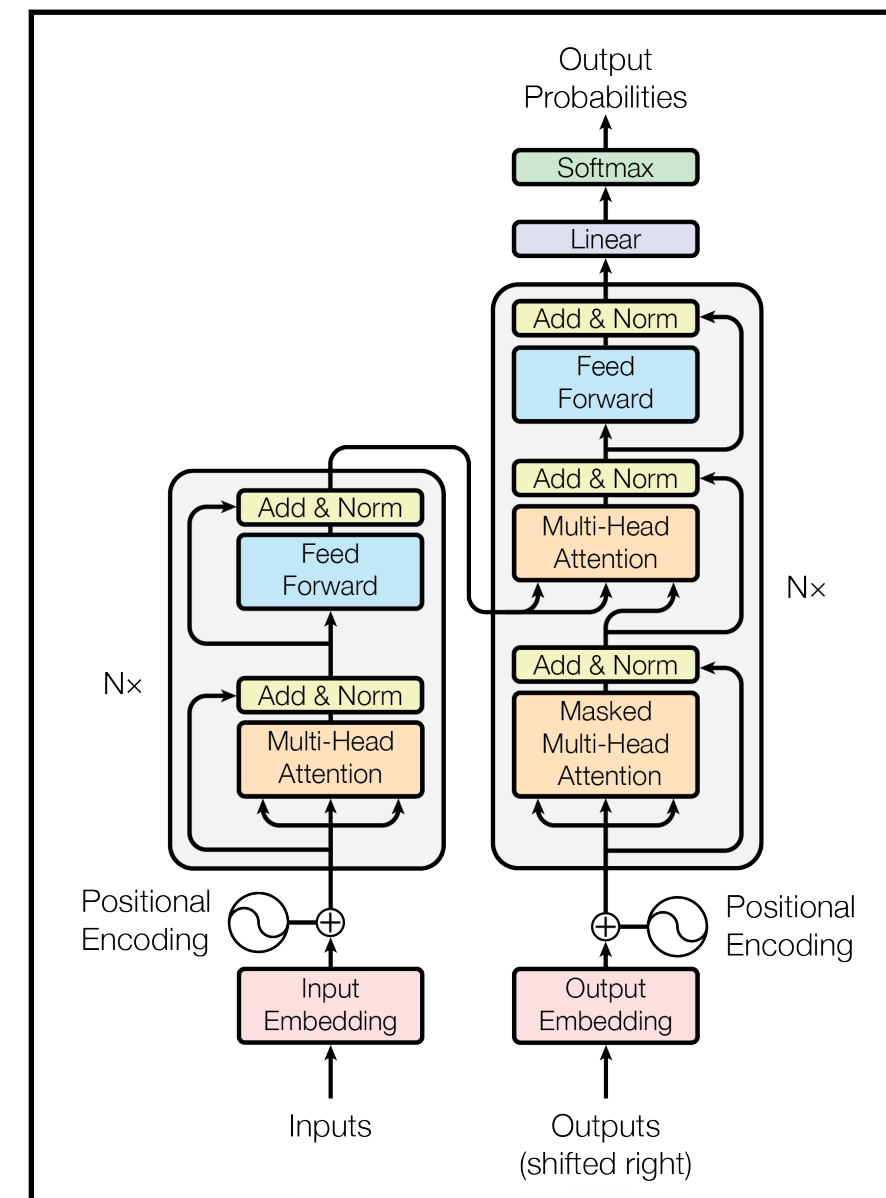
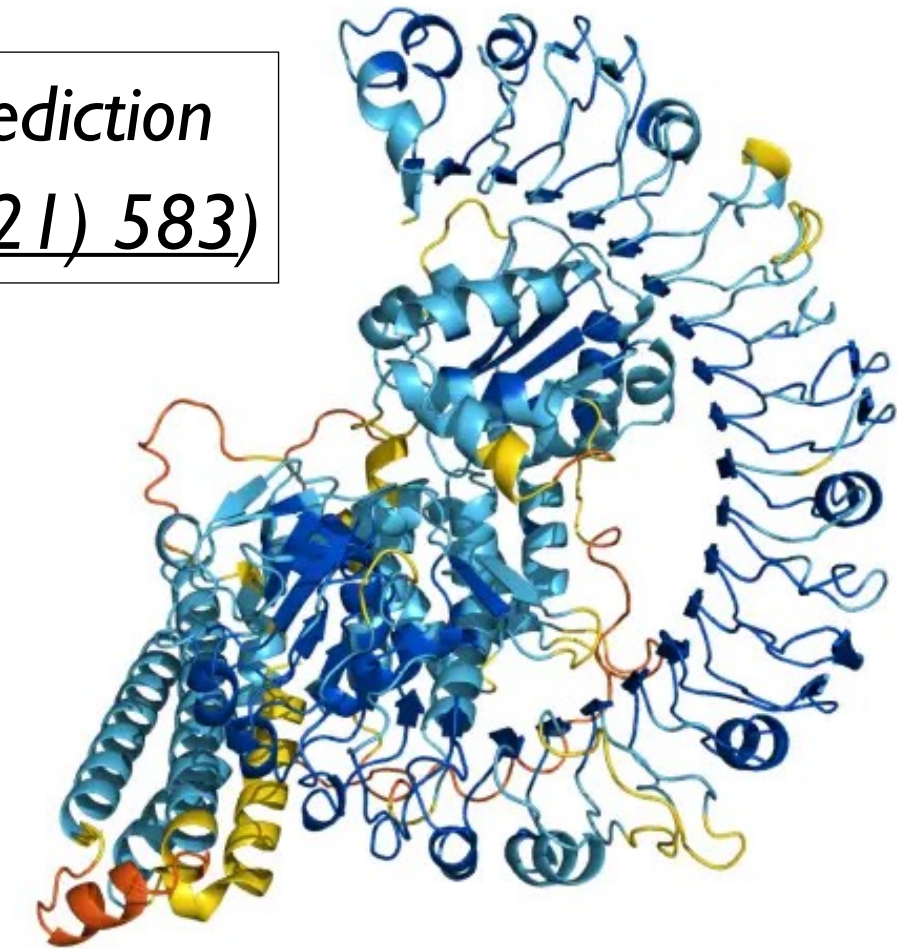
Large Language Models



Image/Video Generation



Protein Structure Prediction
(e.g. *Nature* 596 (2021) 583)



Scattering Amplitude Computation
(e.g. *MLST* 5 (2024) 035073)

ATTENTION IS ALL YOU NEED?

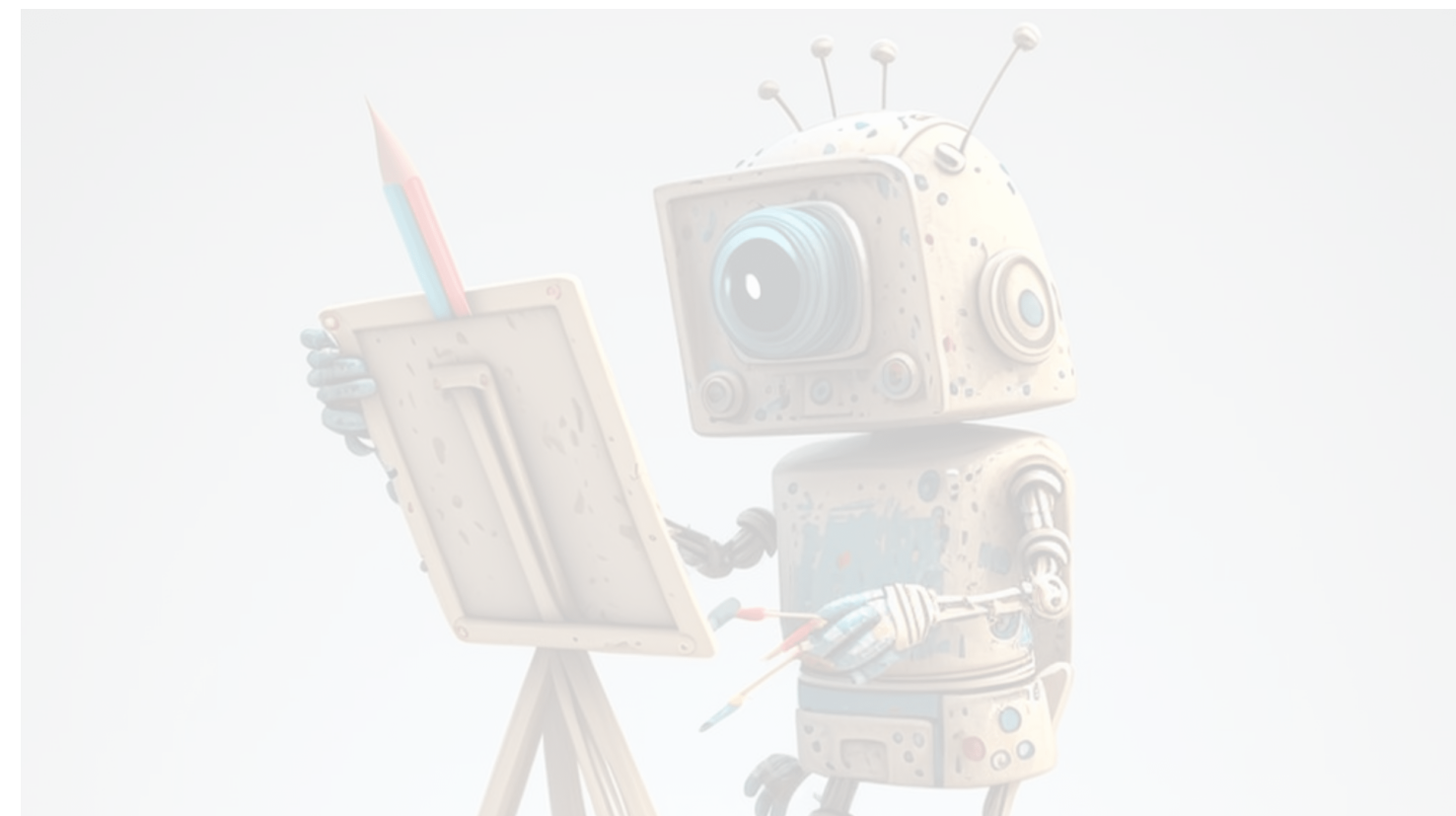
- Transformers are taking over every task

Large Language Models

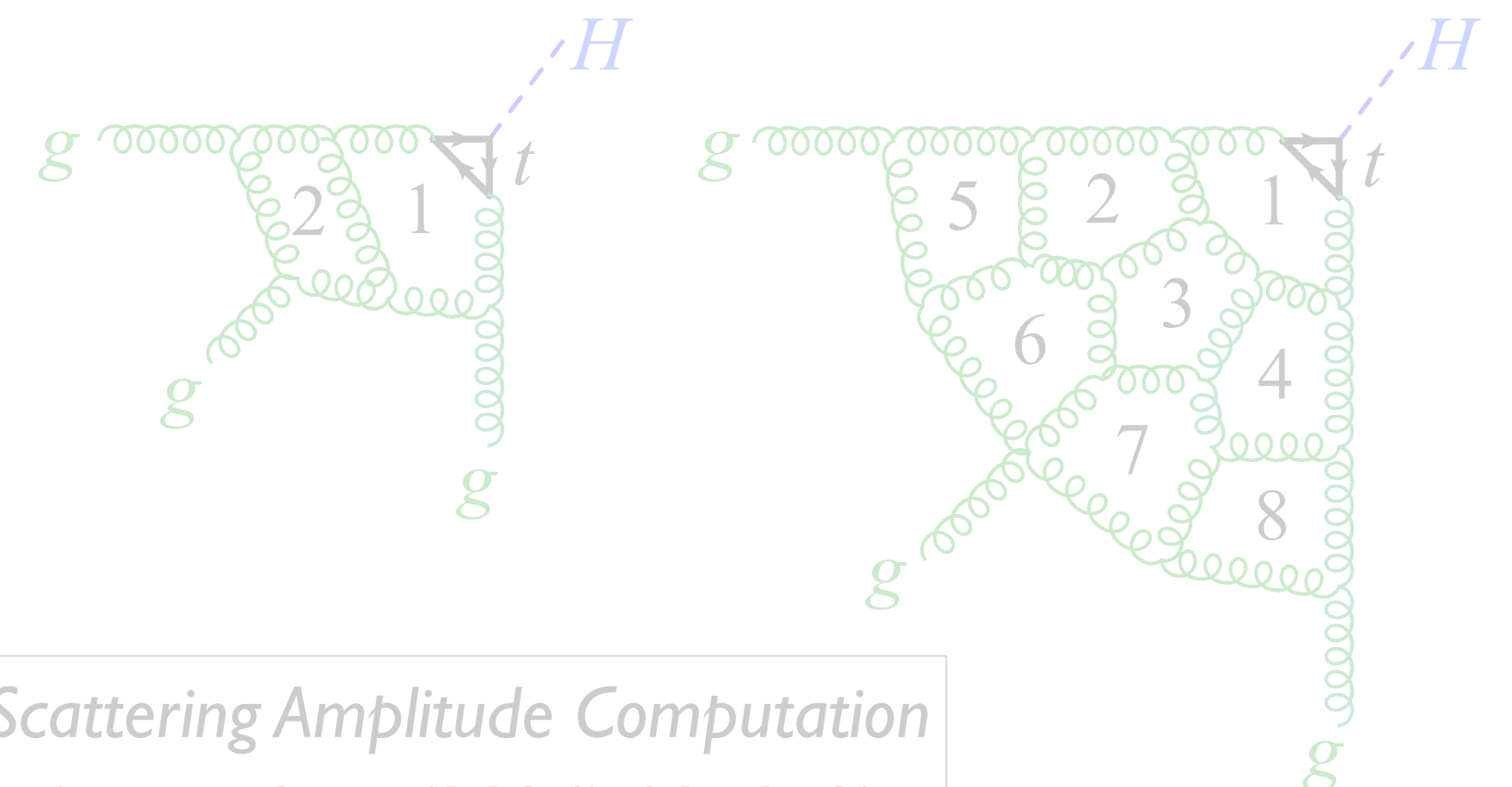
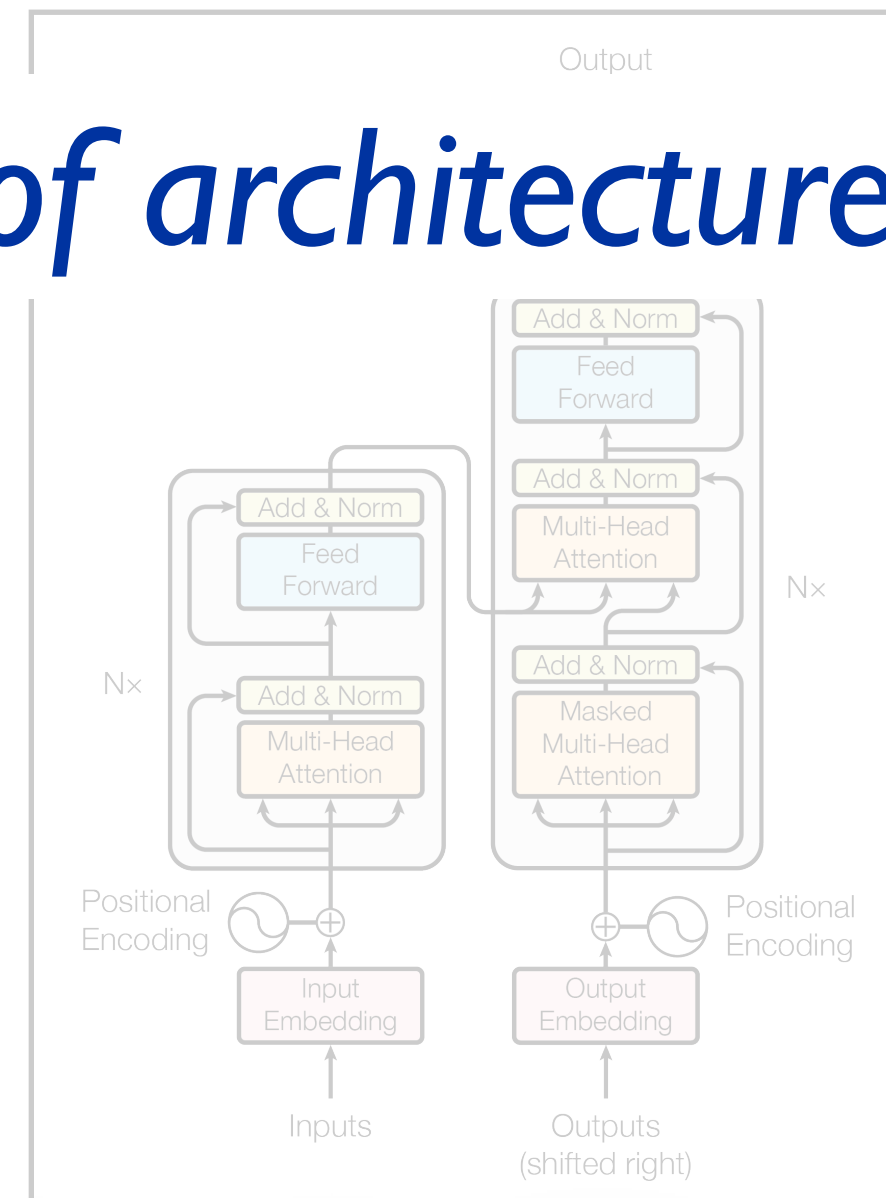
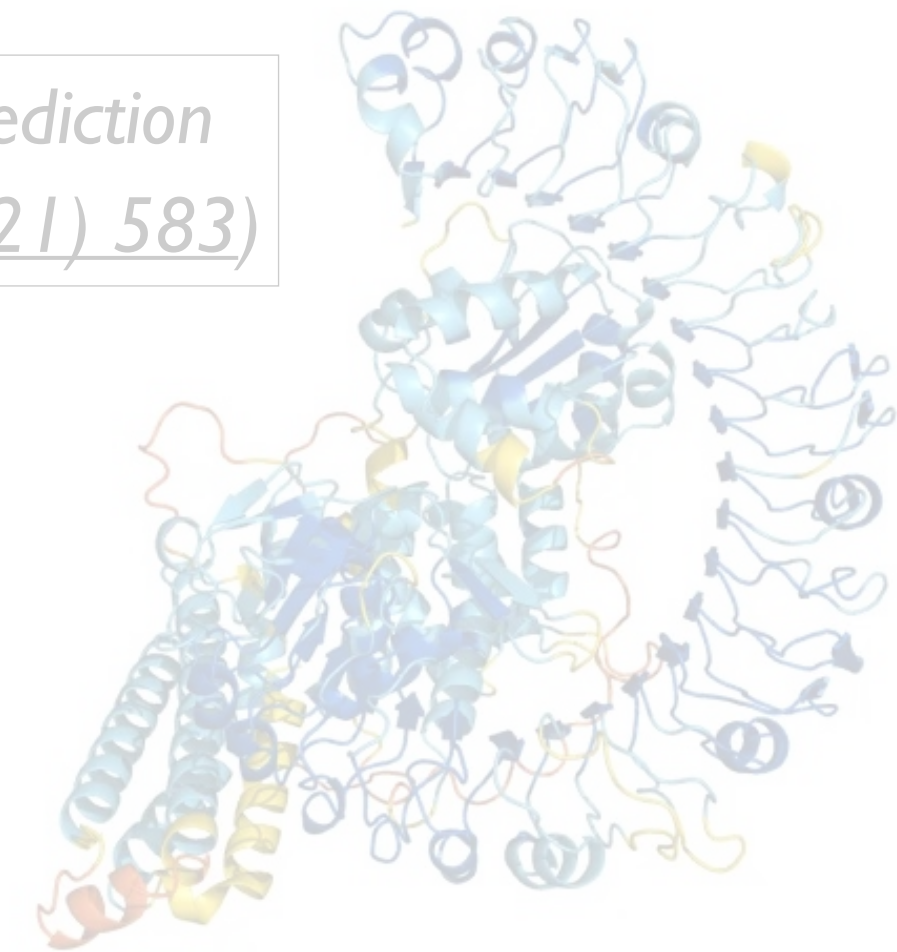


The end of architecture research?

Image/Video Generation



Protein Structure Prediction
(e.g. *Nature* 596 (2021) 583)



Scattering Amplitude Computation
(e.g. *MLST* 5 (2024) 035073)

ATTENTION IS ALL YOU NEED?

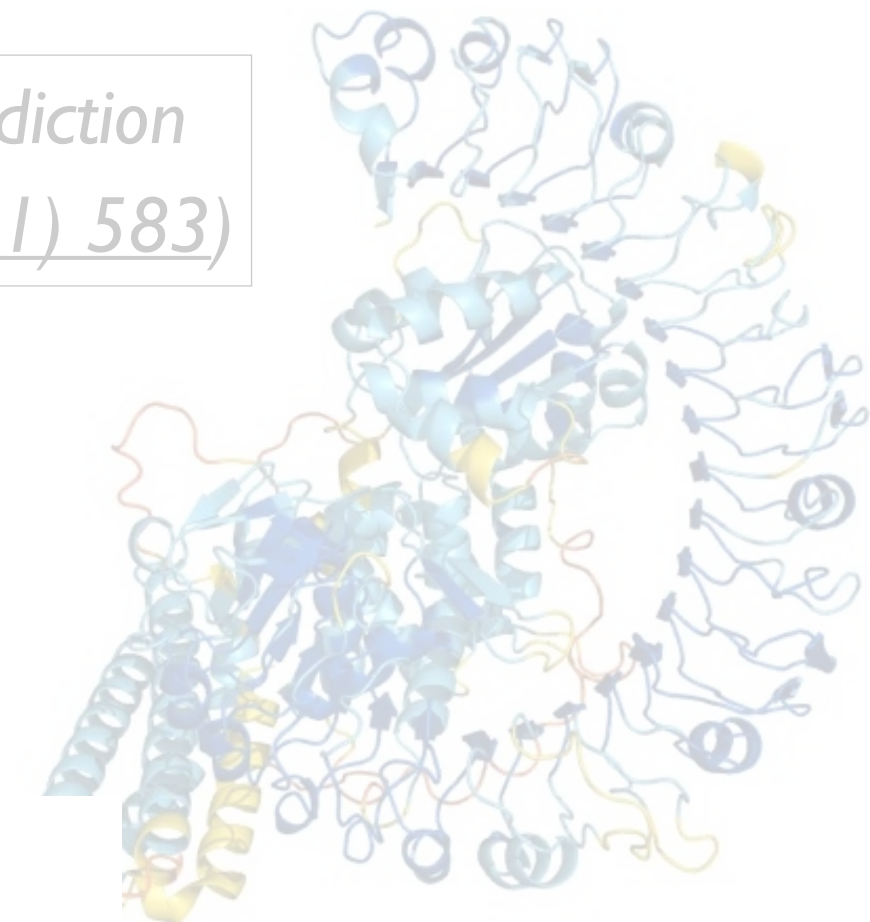
- Transformers are taking over every task

Large Language Models



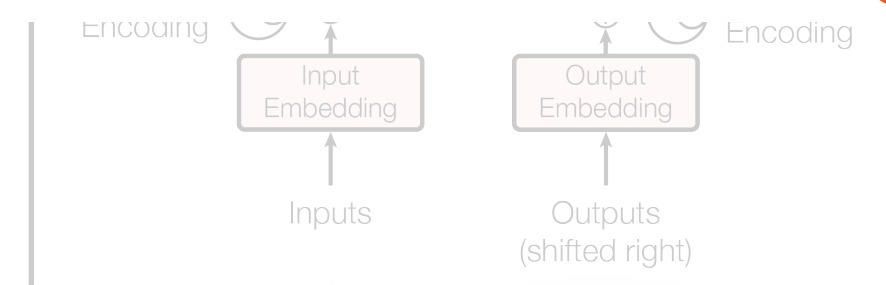
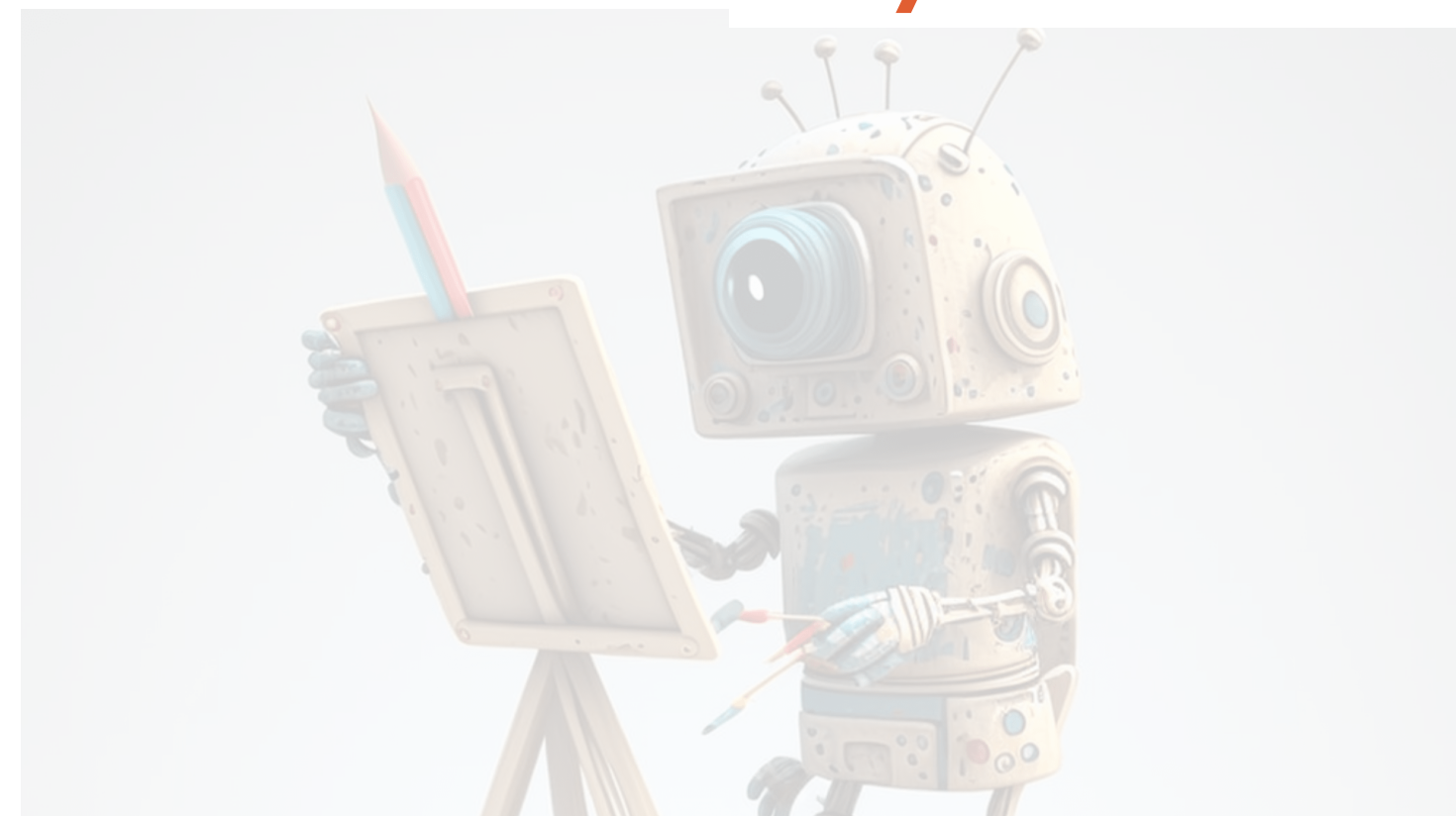
The end of architecture research?

Protein Structure Prediction
(e.g. *Nature* 596 (2021) 583)



Not really...

Physics + network “co-design” is the key.

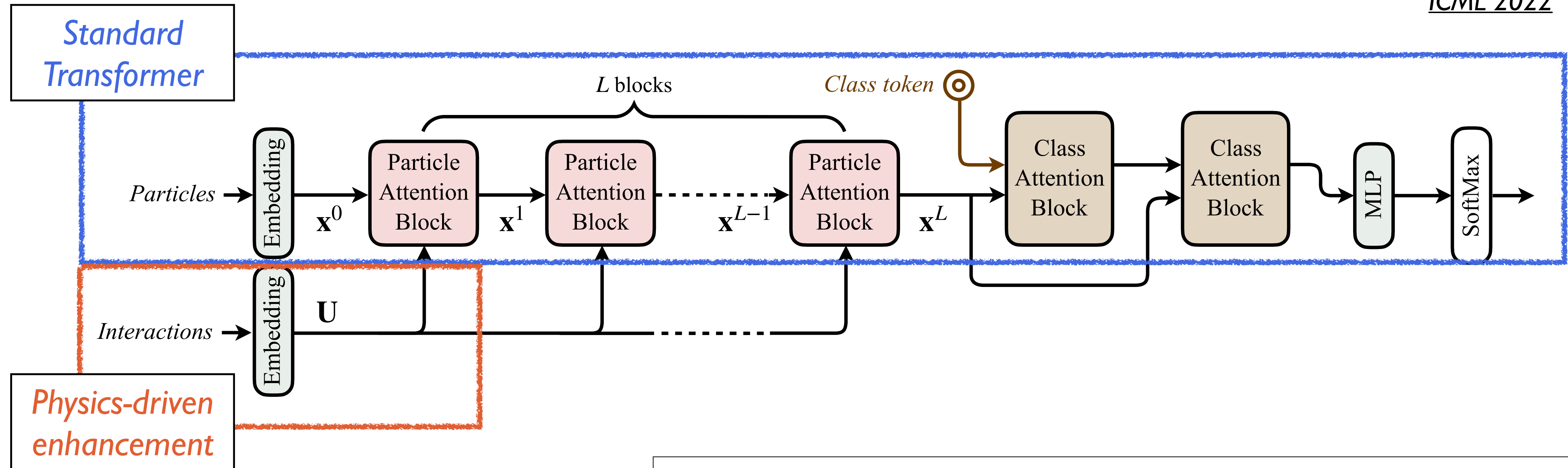


Scattering Amplitude Computation
(e.g. *MLST* 5 (2024) 035073)

PARTICLE TRANSFORMER

- **Particle Transformer (ParT):** Transformer model **tailored for particle physics**

HQ, C. Li, S. Qian,
ICML 2022



$$\Delta = \sqrt{(y_a - y_b)^2 + (\phi_a - \phi_b)^2},$$

$$k_T = \min(p_{T,a}, p_{T,b}) \Delta,$$

$$z = \min(p_{T,a}, p_{T,b}) / (p_{T,a} + p_{T,b}),$$

$$m^2 = (E_a + E_b)^2 - \|\mathbf{p}_a + \mathbf{p}_b\|^2,$$

$$\text{P-MHA}(Q, K, V) = \text{SoftMax}(QK^T / \sqrt{d_k} + \mathbf{U})V,$$

Injection of (physics-inspired) pairwise features to
“bias” the dot-product self-attention

PARTICLE TRANSFORMER: PERFORMANCE

	All classes		$H \rightarrow b\bar{b}$	$H \rightarrow c\bar{c}$	$H \rightarrow gg$	$H \rightarrow 4q$	$H \rightarrow \ell\nu qq'$	$t \rightarrow bqq'$	$t \rightarrow b\ell\nu$	$W \rightarrow qq'$	$Z \rightarrow q\bar{q}$
	Accuracy	AUC	Rej _{50%}	Rej _{50%}	Rej _{50%}	Rej _{50%}	Rej _{99%}	Rej _{50%}	Rej _{99.5%}	Rej _{50%}	Rej _{50%}
PFN	0.772	0.9714	2924	841	75	198	265	797	721	189	159
P-CNN	0.809	0.9789	4890	1276	88	474	947	2907	2304	241	204
ParticleNet	0.844	0.9849	7634	2475	104	954	3339	10526	11173	347	283
ParT	0.861	0.9877	10638	4149	123	1864	5479	32787	15873	543	402
ParT (plain)	0.849	0.9859	9569	2911	112	1185	3868	17699	12987	384	311

- Particle Transformer (ParT): significant performance improvement!
 - compared to the existing state-of-the-art, ParticleNet
 - 1.7% increase in accuracy
 - **up to 3x increase in background rejection** (Rej_{x%})
- ParT (plain): plain Transformer w/o interaction features
 - 1.2% drop in accuracy compared to full ParT
 - **Physics-driven modification of self-attention plays a key role!**

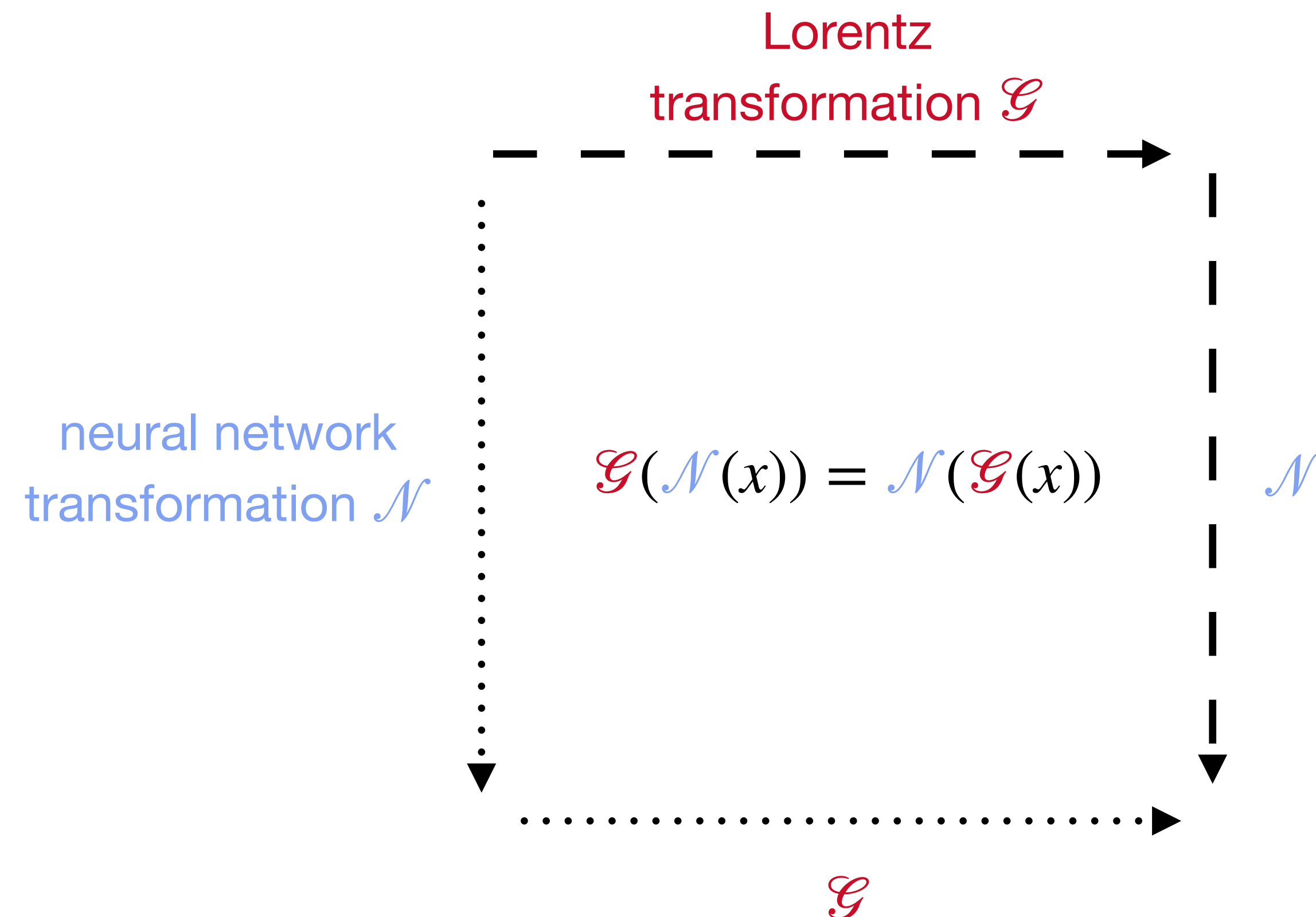
JETCLASS dataset (100M jets)

Model complexity

	Accuracy	# params	FLOPs
PFN	0.772	86.1 k	4.62 M
P-CNN	0.809	354 k	15.5 M
ParticleNet	0.844	370 k	540 M
ParT	0.861	2.14 M	340 M
ParT (plain)	0.849	2.13 M	260 M

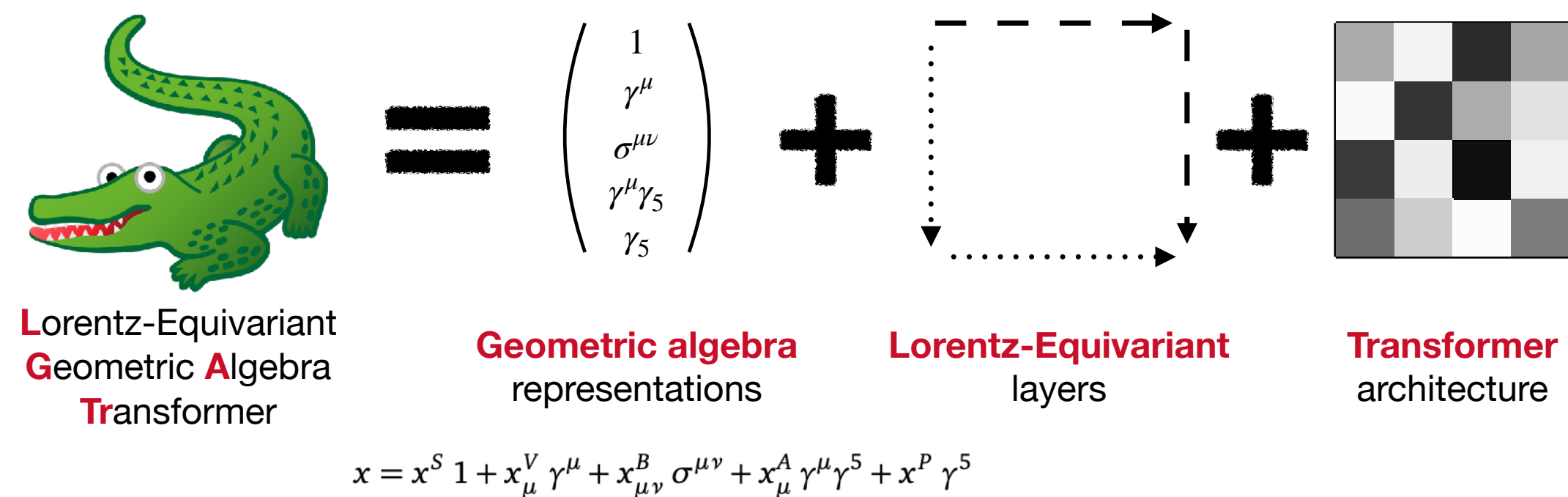
SYMMETRY-AWARE ARCHITECTURES

- One step further: symmetry-aware architectures
 - one important symmetry for particle physics: Lorentz symmetry
- Lorentz-equivariant architectures
 - equivariance (covariance) – preserve the symmetry by construction



L-GATr & LLoCa

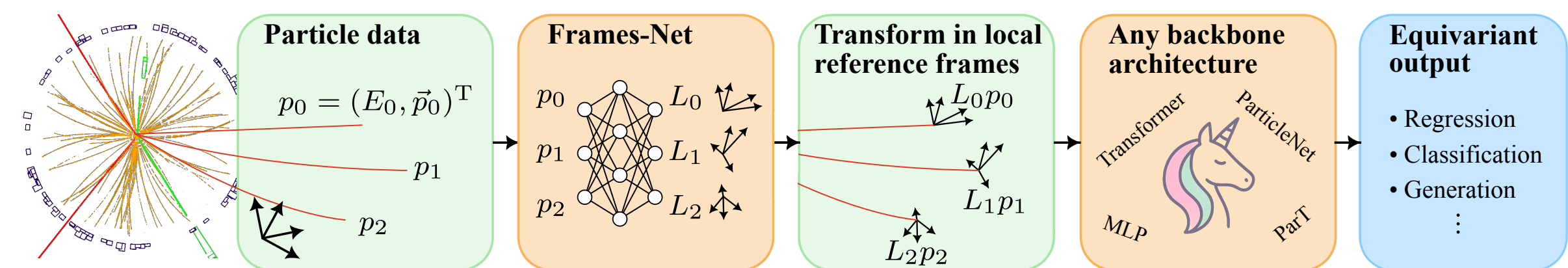
L-GATr: Lorentz-Equivariant Geometric Algebra Transformer



Specialized Lorentz-equivariant linear/attention/norm layers based on geometric algebra representations.

J. Brehmer, V. Bresó, P. Haan, T. Plehn, HQ, J. Spinner and J. Thaler, [arXiv: 2411.00446](https://arxiv.org/abs/2411.00446)

LLoCa: Lorentz Local Canonicalization



Learned local reference frames to transform the inputs into a (learned) canonical frame and then the outputs back. Can be applied to any architecture (GNN, transformer).

L. Favaro, G. Gerhartz, F.A. Hamprecht, P. Lippmann, S. Pitz, T. Plehn, HQ and J. Spinner, [arXiv: 2508.14898](https://arxiv.org/abs/2508.14898)

L-GATr & LLoCa PERFORMANCE

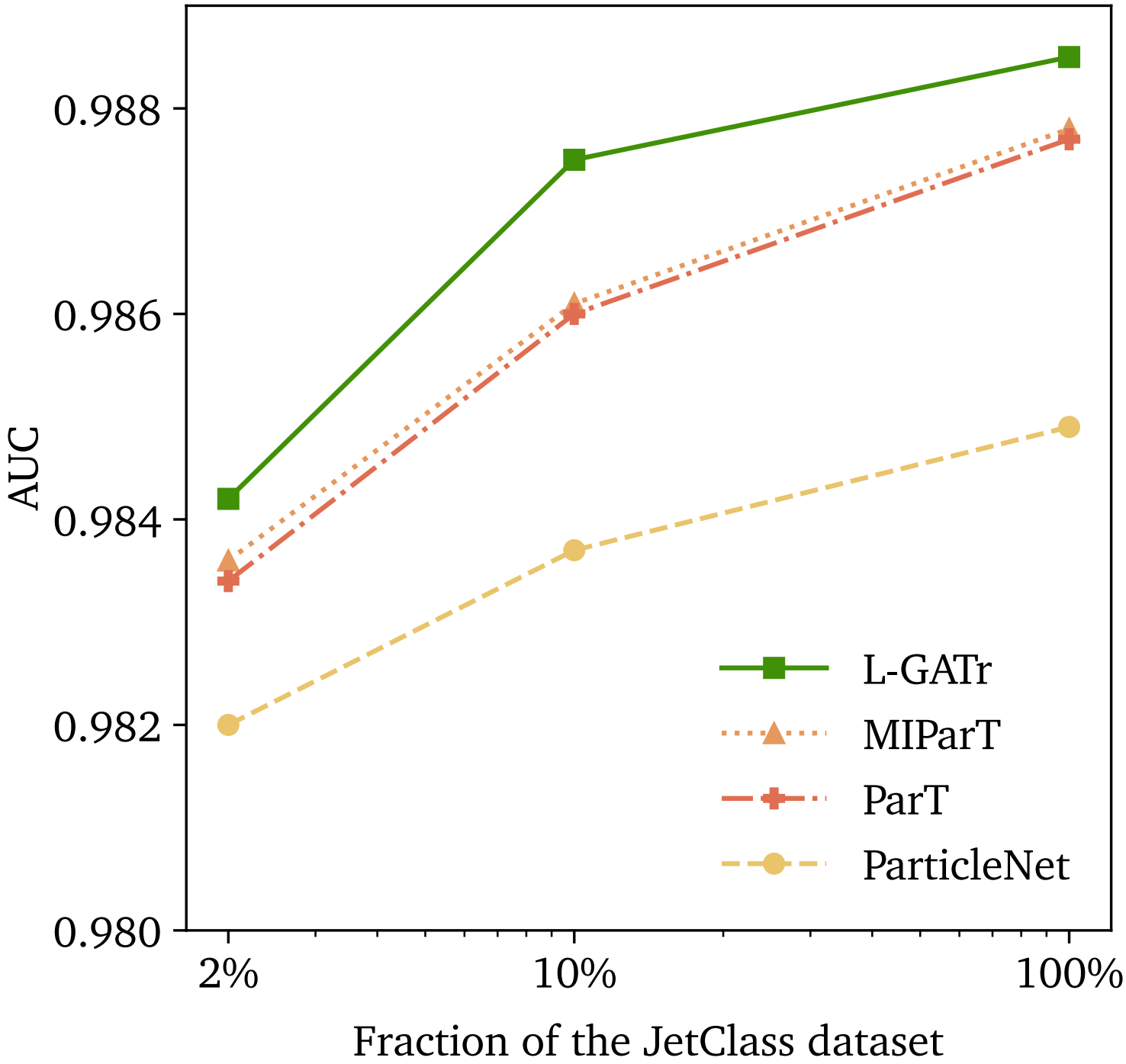
Performance on JetClass (100M)

Network	Accuracy	AUC	Time	FLOPs	Memory	Parameters
MIParT-L [26]	0.861	0.9878	43h	225M	53.6G	2380k
LorentzNet* [11]	0.847	0.9856	64h	676M	20.5G	223k
PELICAN-lite* [17]	0.851	0.9862	97h	1370M	27.4G	244k
ParticleNet [9]	0.844	0.9849	25h	413M	16.5G	366k
LLoCa-ParticleNet* [17]	0.848	0.9857	43h	517M	23.5G	385k
ParT [18]	0.861	0.9877	33h	211M	13.3G	2141k
LLoCa-ParT* [17]	0.864	0.9882	66h	315M	19.9G	2160k
Transformer [17]	0.855	0.9867	15h	210M	2.3G	1979k
LLoCa-Transformer* [17]	0.864	0.9882	28h	219M	4.1G	1980k
L-GATr* [16]	0.866	0.9885	166h	2060M	19.0G	1079k
L-GATr-slim*	0.866	0.9885	27h	329M	8.1G	2031k

* *L-GATr-slim*: a more efficient version of L-GATr implemented using only Lorentz scalar and vectors.

A. Petitjean, T. Plehn, J. Spinner and U. Köthe, [arXiv: 2512.17011](#)

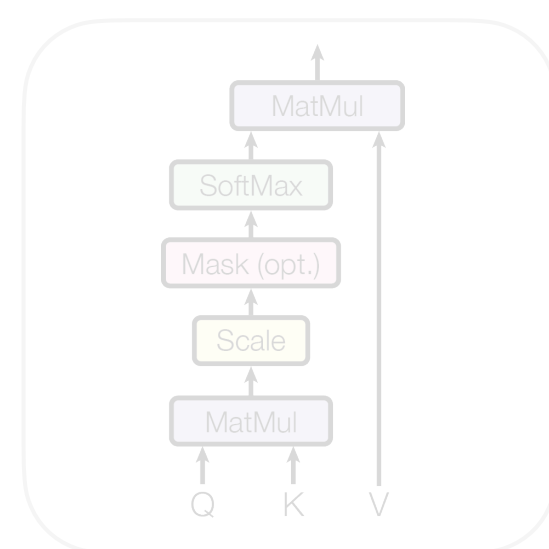
Scaling w/ training set size





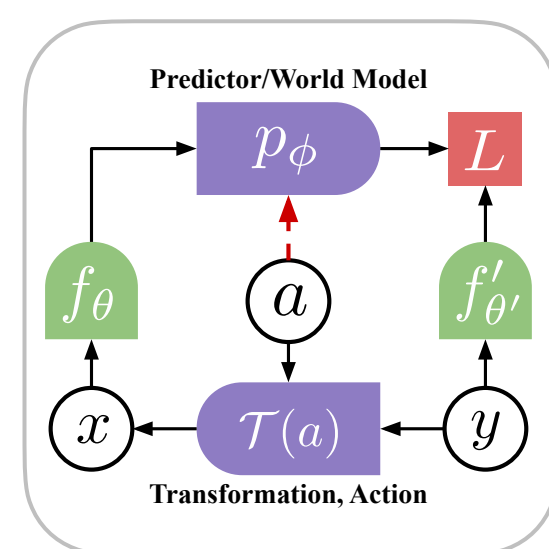
Evolution of Jet Representation

image, sequence, point cloud, ...



Attention Is All You Need?

or how physics can help...

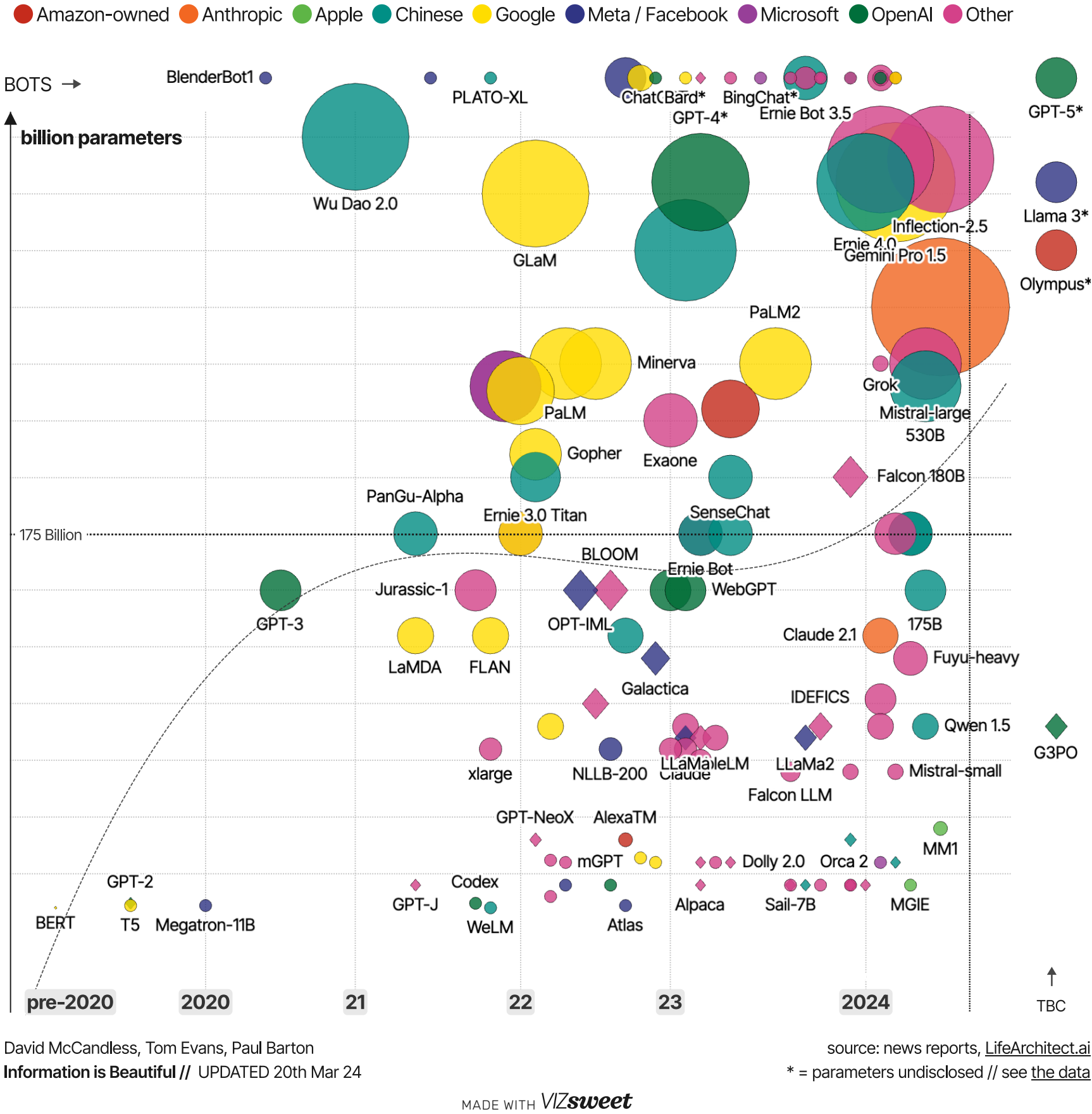


Towards a Foundation Model

scaling, self-supervision, and more...

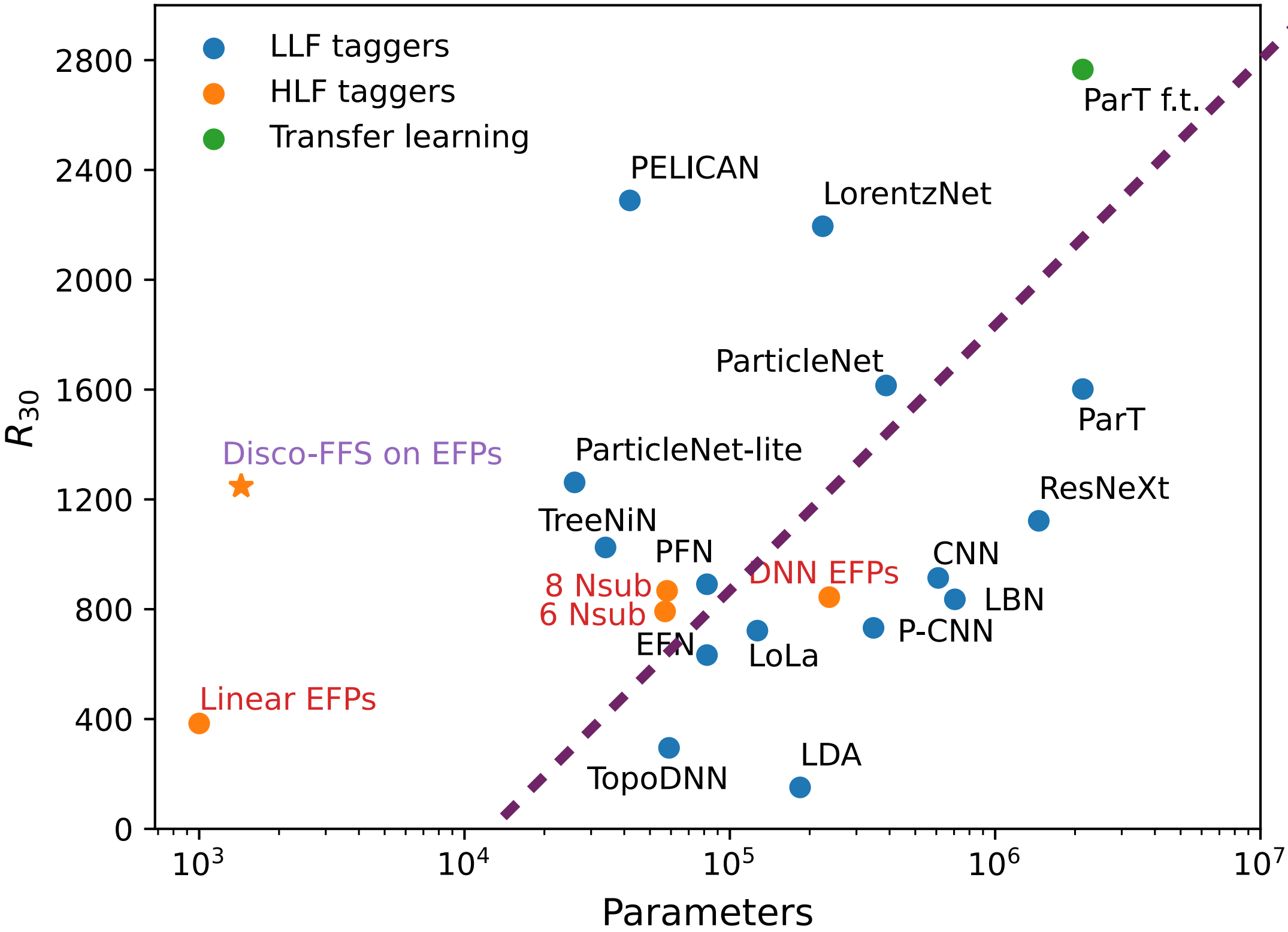
WHAT'S NEXT?

Natural language models



Source: informationisbeautiful.net

HEP models (e.g., jet tagging)



R. Das, G. Kasieczka and D. Shih, [PRD 109, 054009](#)

SCALING LAWS

- Neural scaling law [arXiv: 2001.08361, 2203.15556]

- **question:** for a fixed compute budget (C), how to determine the optimal dataset size (D) and model parameters (N), such that the loss is minimized?

- **empirical power law:**

$$\hat{L}(N, D) \triangleq E + \frac{A}{N^\alpha} + \frac{B}{D^\beta}.$$

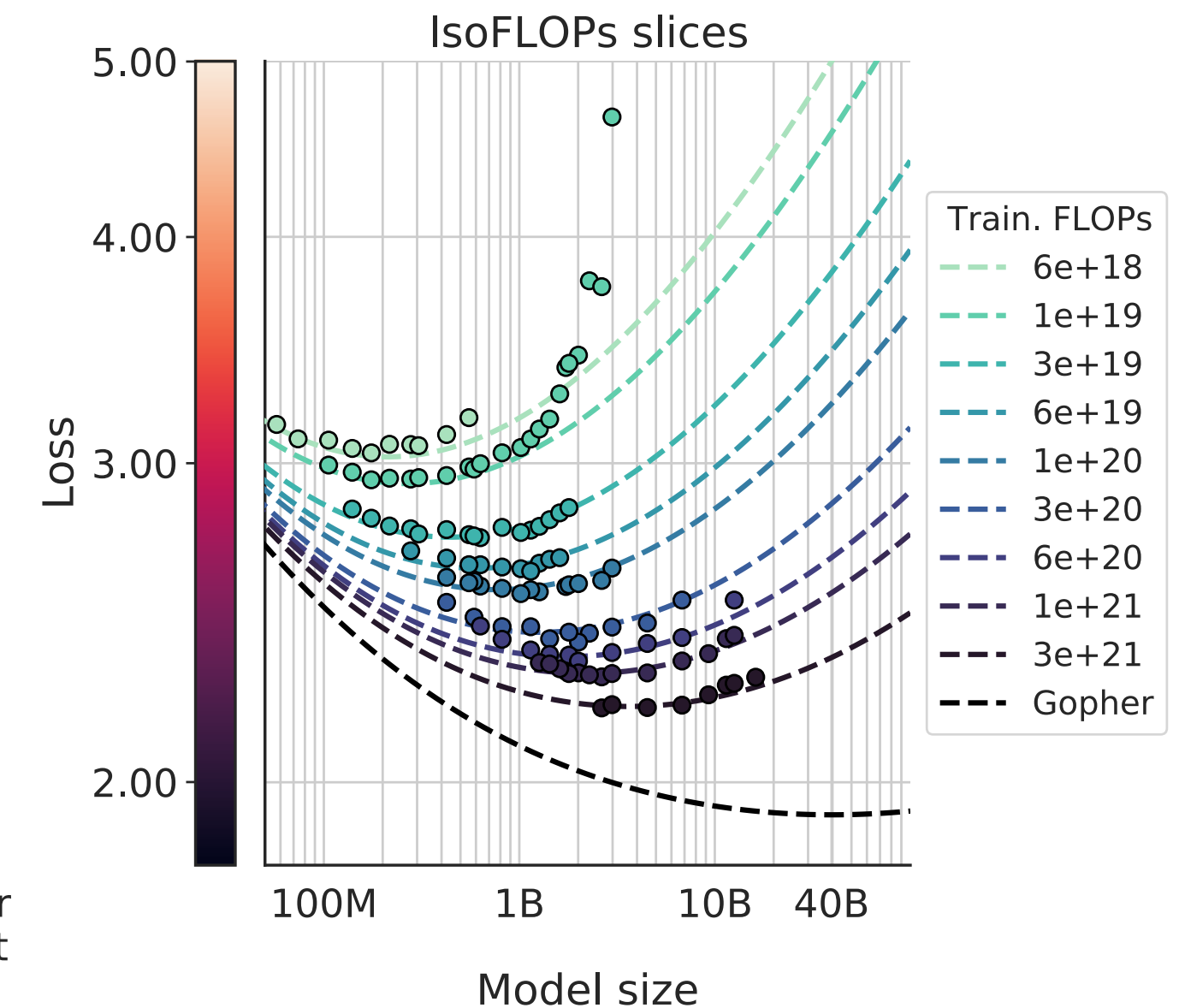
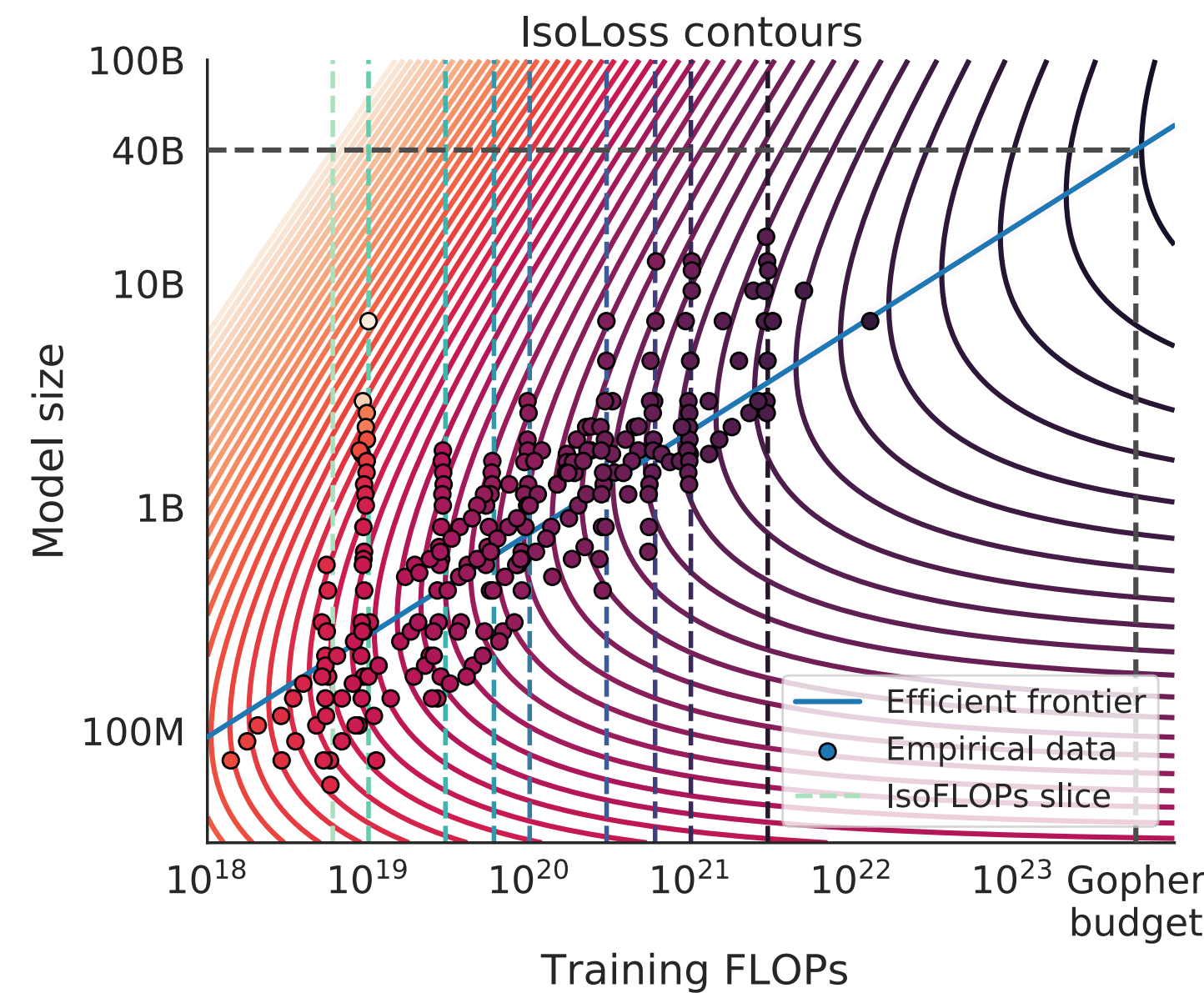
- for transformers:

$$\text{FLOPs}(N, D) \approx 6ND$$

- **compute-optimal scaling:**

$$N_{\text{opt}}(C) = G \left(\frac{C}{6} \right)^a, \quad D_{\text{opt}}(C) = G^{-1} \left(\frac{C}{6} \right)^b,$$

$$\text{where } G = \left(\frac{\alpha A}{\beta B} \right)^{\frac{1}{\alpha+\beta}}, \quad a = \frac{\beta}{\alpha+\beta}, \text{ and } b = \frac{\alpha}{\alpha+\beta}.$$



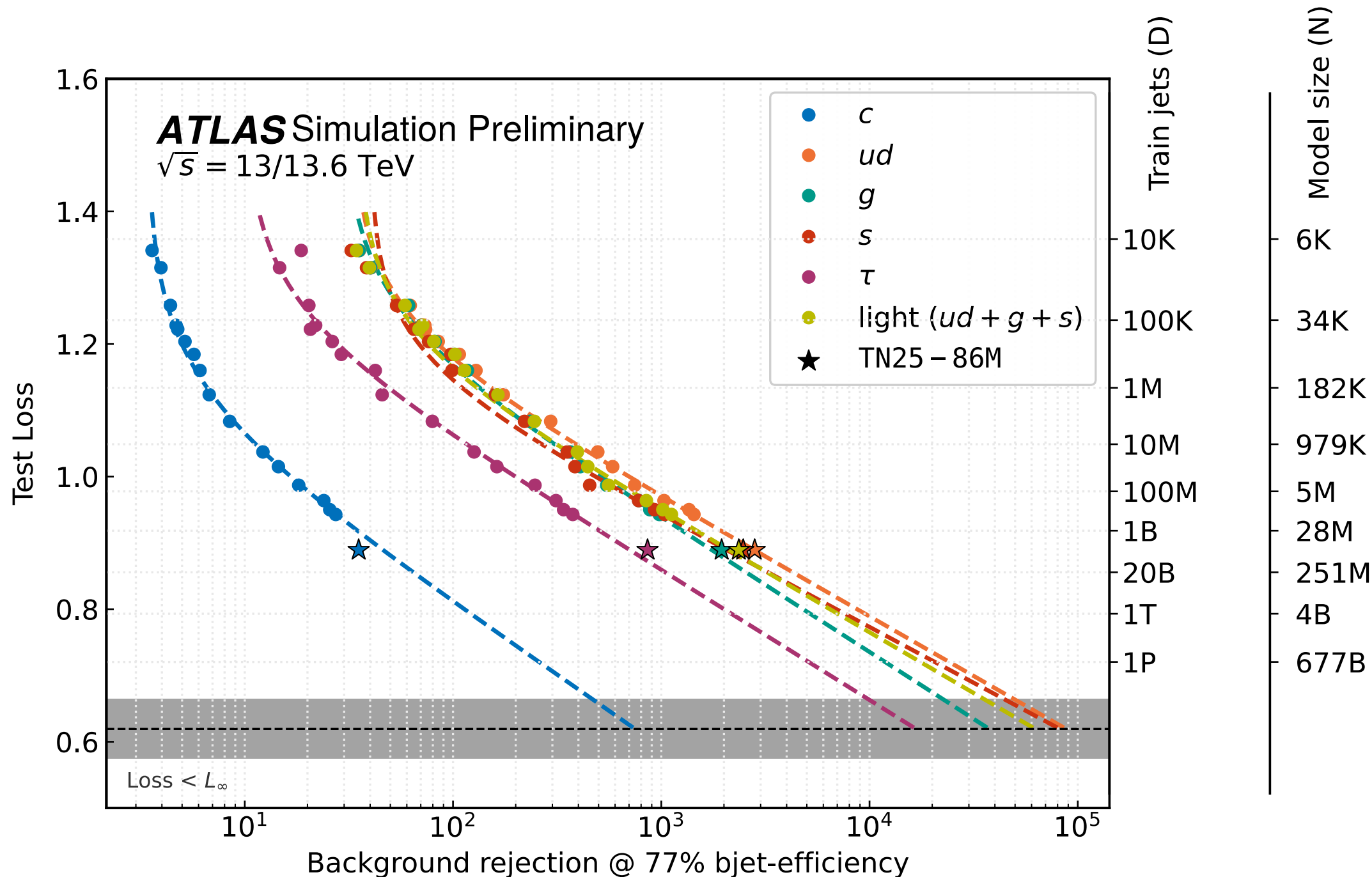
Interestingly, for LLMs, the optimal scaling is found to be $a \approx b \approx 0.50$ — same scaling for data and parameters.

SCALING LAWS FOR JETS

$$L(N, D) = L_\infty + \frac{A}{N^\alpha} + \frac{B}{D^\beta},$$
$$N \propto C^a, \quad D \propto C^{1-a}, \quad L \propto C^{-\gamma}, \quad \text{with } a = \frac{\beta}{\alpha + \beta}.$$

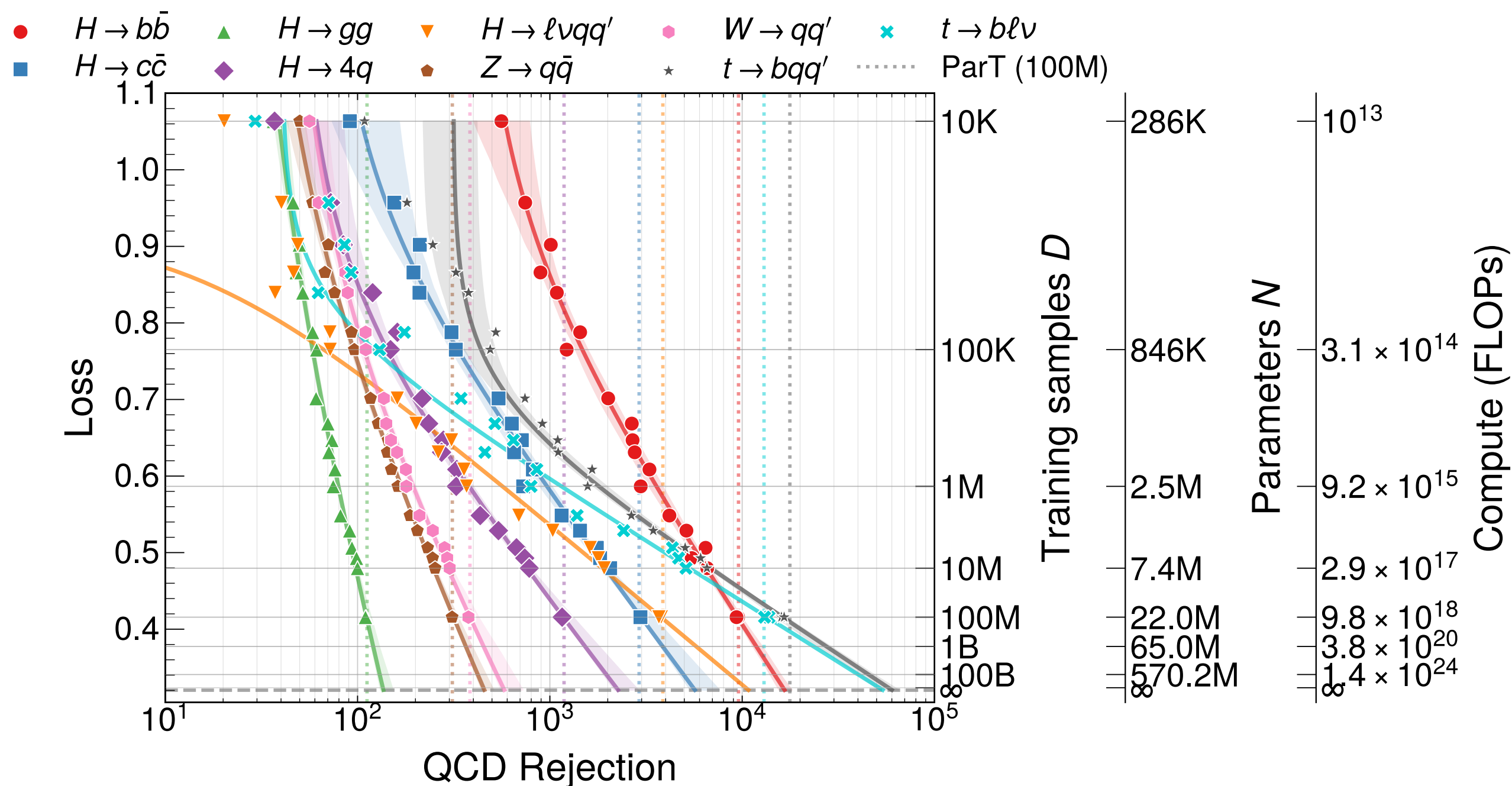
Jet flavor tagging (Small-R)

ATL-SOFT-PUB-2026-002



Boosted jet tagging (Large-R)

M.Vigl, N. Hartman, M. Kagan and L. Heinrich, [arXiv:2602.15781](#)



Quantity	Symbol	Description	Value
Irreducible loss	L_∞	$L(N \rightarrow \text{inf}, D \rightarrow \text{inf})$	0.619 (0.539, 0.671)
Model scaling exponent	α	$L_N \propto N^{-\alpha}$	0.677 (0.644, 0.764)
Data scaling exponent	β	$L_D \propto D^{-\beta}$	0.077 (0.065, 0.086)
$C = 6ND$ optimal exponent	γ	$L \propto C^{-\gamma}$	0.07 (0.064, 0.074)

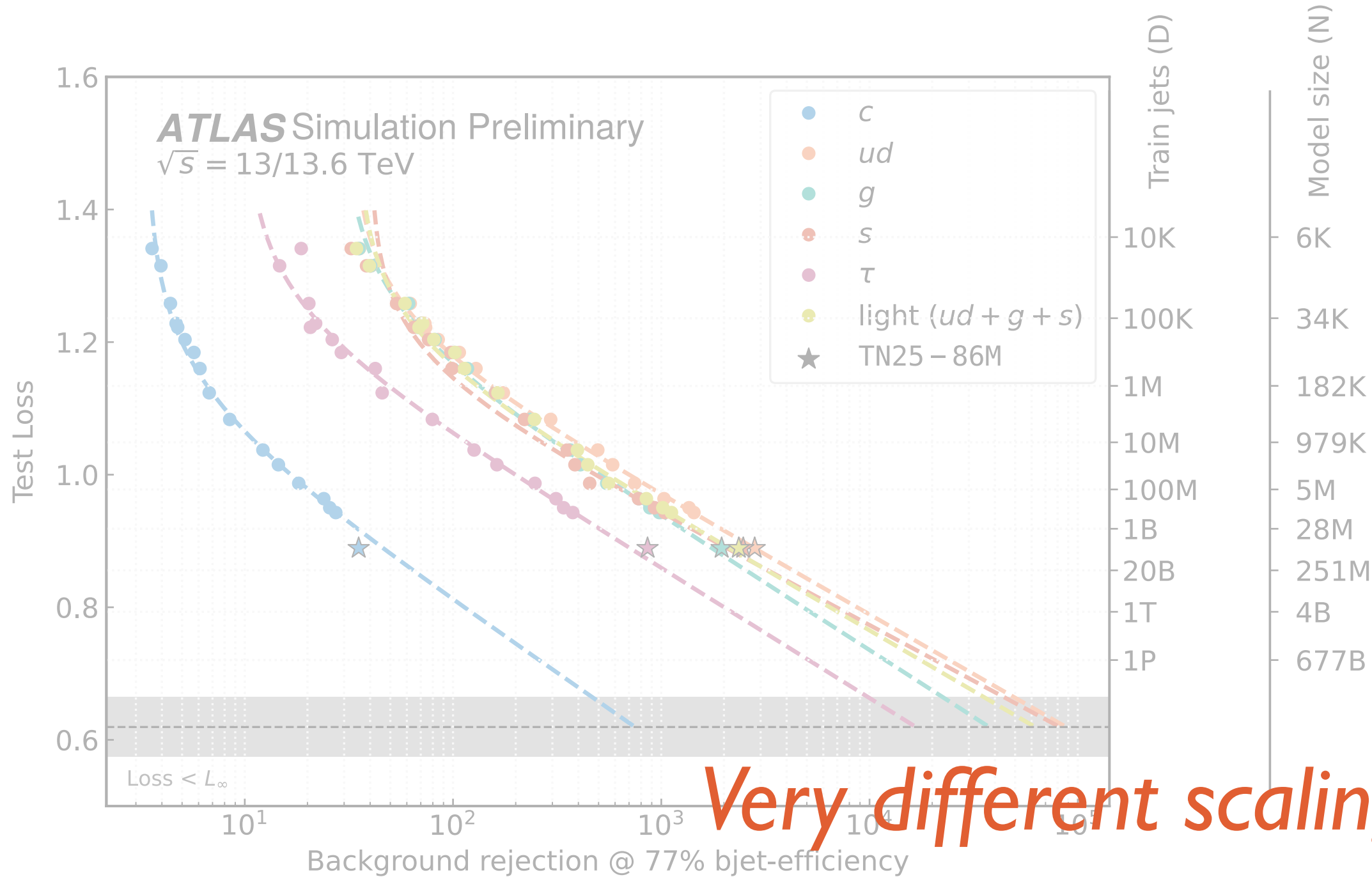
Parameter	Value (68% CI)
L_∞	0.32 (0.29, 0.34)
A	11.27 (8.01, 15.14)
α	0.44 (0.40, 0.48)
B	7.22 (6.66, 7.89)
β	0.22 (0.21, 0.23)
γ	0.15 (0.13, 0.16)

SCALING LAWS FOR JETS

$$L(N, D) = L_\infty + \frac{A}{N^\alpha} + \frac{B}{D^\beta},$$
$$N \propto C^a, \quad D \propto C^{1-a}, \quad L \propto C^{-\gamma}, \quad \text{with } a = \frac{\beta}{\alpha + \beta}.$$

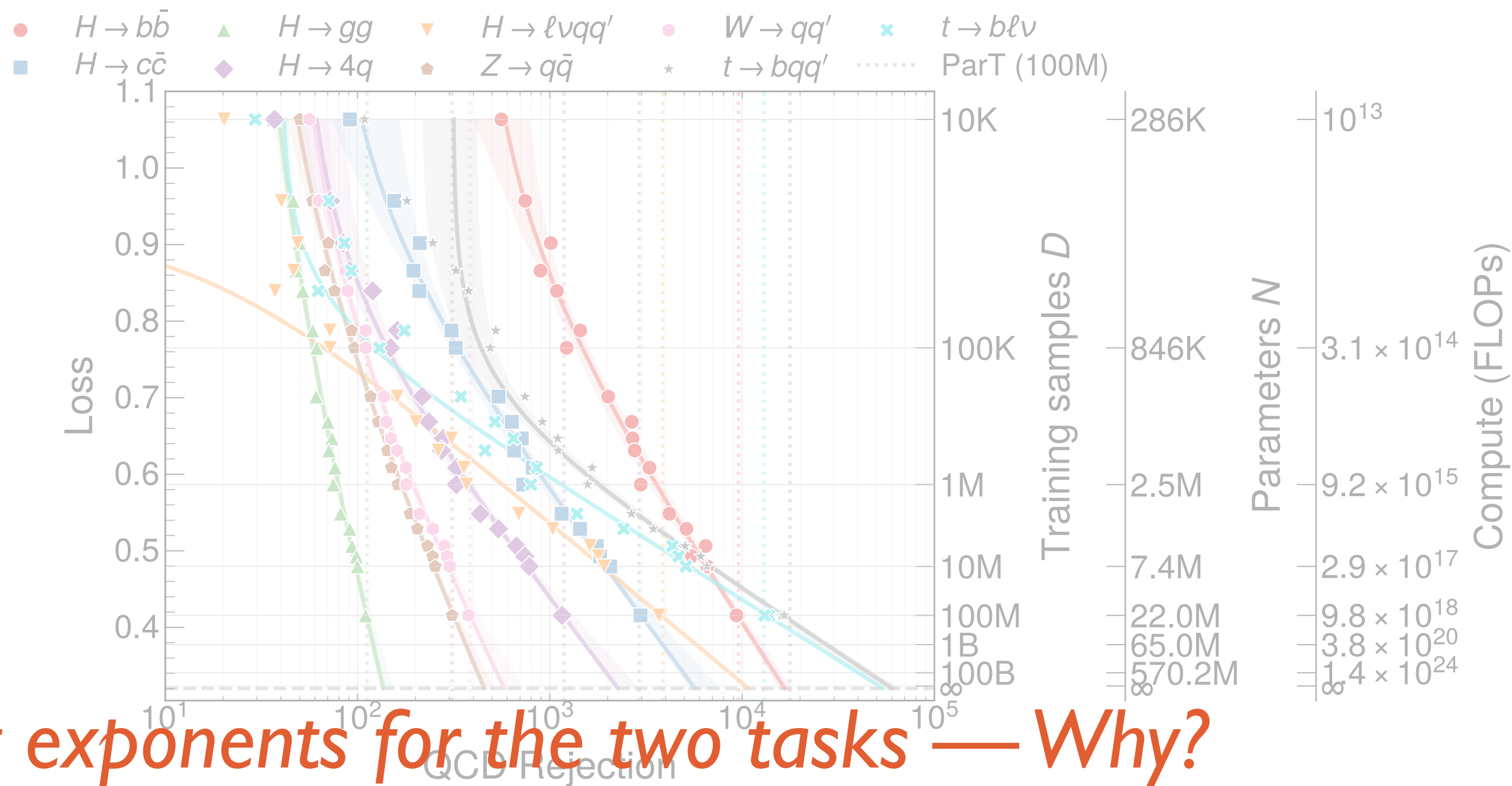
Jet flavor tagging (Small-R)

ATL-SOFT-PUB-2026-002



Boosted jet tagging (Large-R)

M.Vigl, N. Hartman, M. Kagan and L. Heinrich, [arXiv:2602.15781](#)



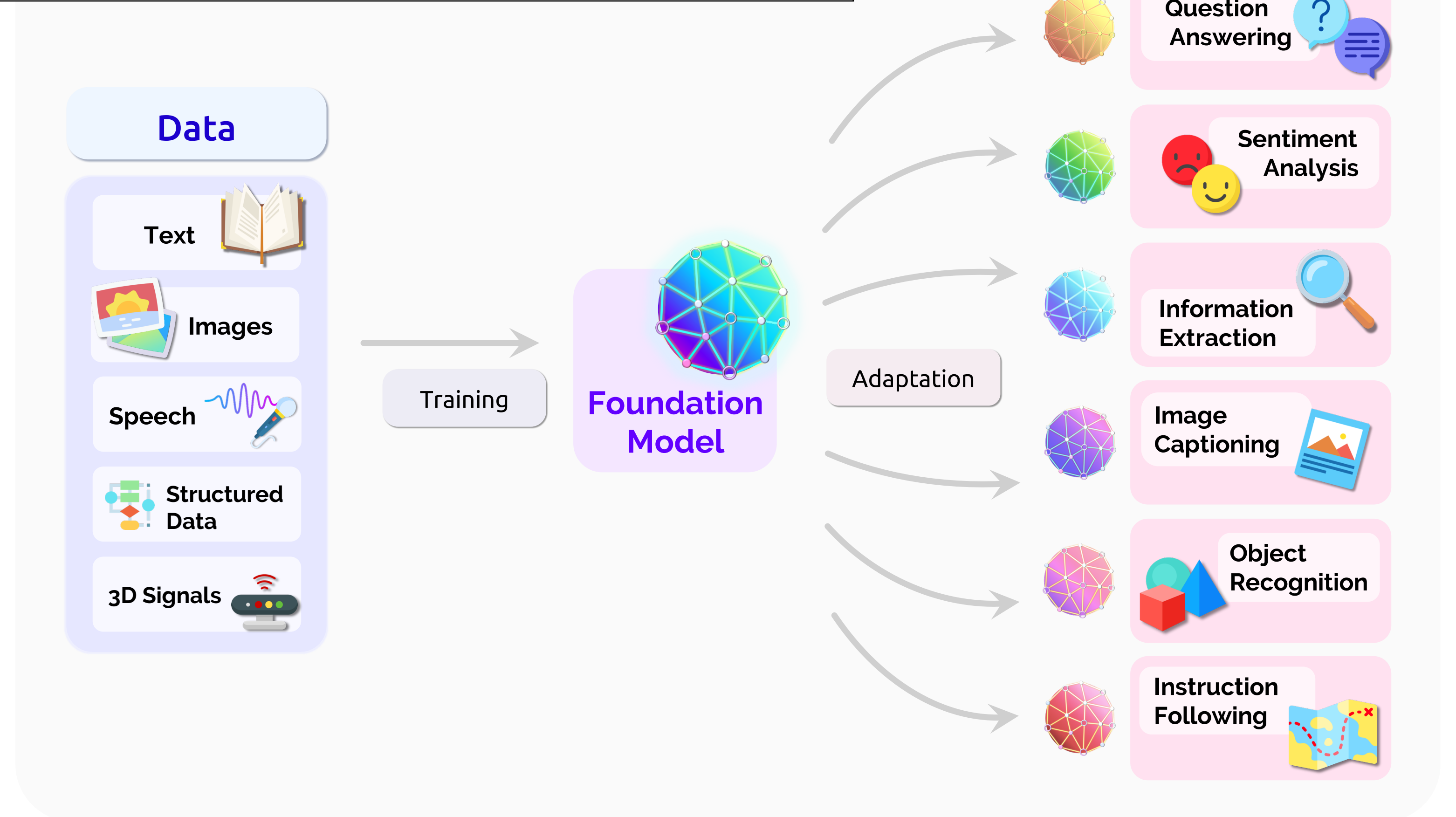
Very different scaling exponents for the two tasks — Why?

Quantity	Symbol	Description	Value
Irreducible loss	L_∞	$L(N \rightarrow \text{inf}, D \rightarrow \text{inf})$	0.619 (0.539, 0.671)
Model scaling exponent	α	$L_N \propto N^{-\alpha}$	0.677 (0.644, 0.764)
Data scaling exponent	β	$L_D \propto D^{-\beta}$	0.077 (0.065, 0.086)
$C = 6ND$ optimal exponent	γ	$L \propto C^{-\gamma}$	0.07 (0.064, 0.074)

Parameter	Value (68% CI)
L_∞	0.32 (0.29, 0.34)
A	11.27 (8.01, 15.14)
α	0.44 (0.40, 0.48)
B	7.22 (6.66, 7.89)
β	0.22 (0.21, 0.23)
γ	0.15 (0.13, 0.16)

FOUNDATION MODEL

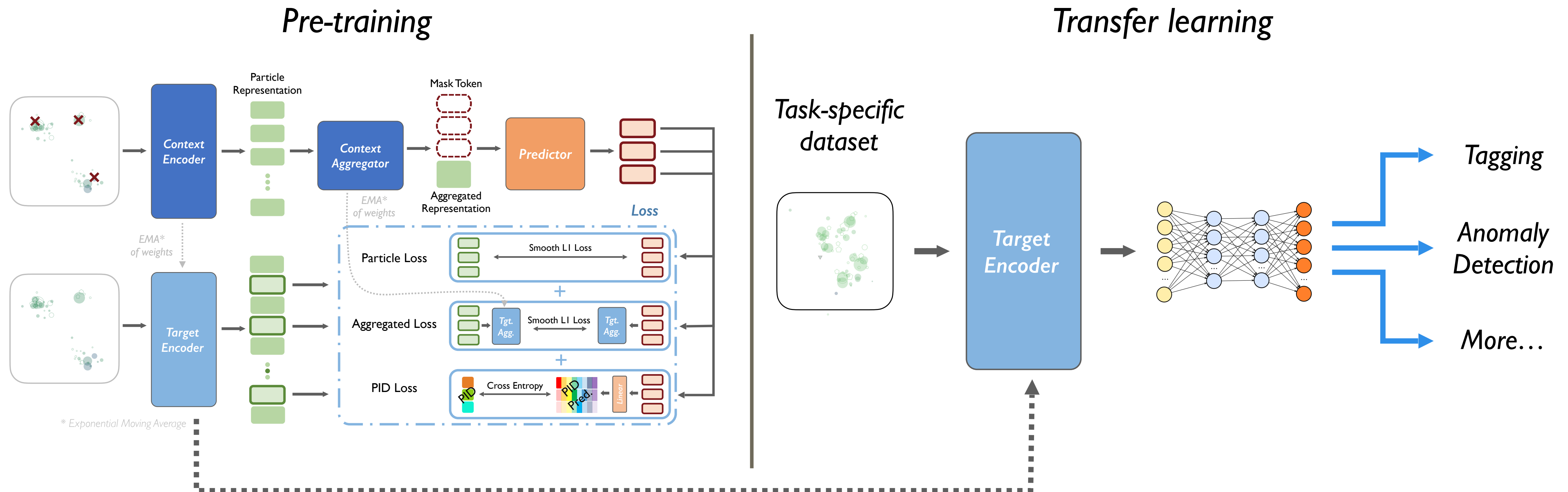
“A foundation model is any model that is trained on *broad data* (generally using self-supervision at scale) that can be adapted (e.g., fine-tuned) to *a wide range of downstream tasks*.”



[arXiv: 2108.07258]

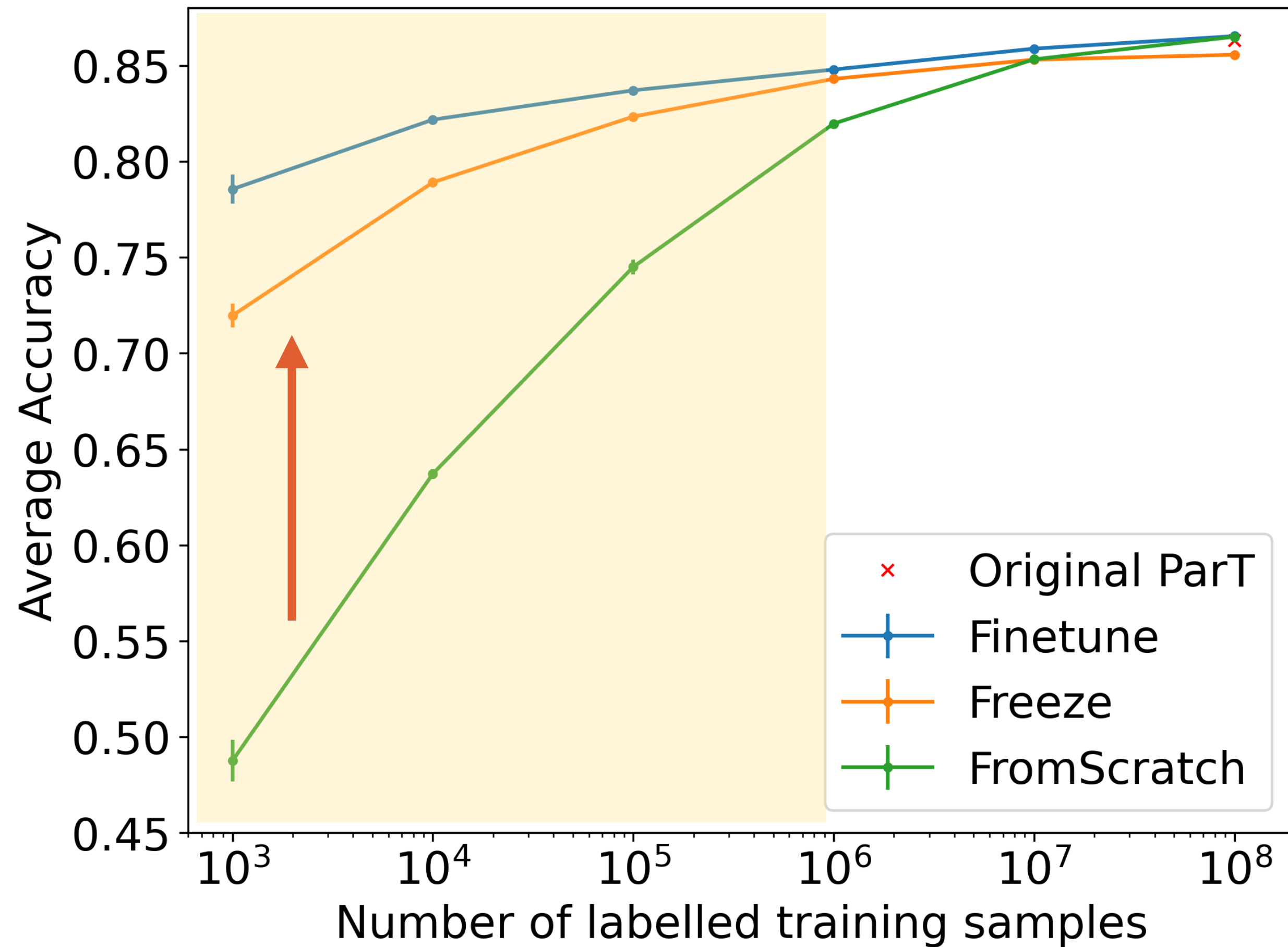
PRE-TRAINING AND TRANSFER LEARNING

- The pre-training of the P-JEPA model can be performed on large-scale real data
 - we demonstrate this by pre-training P-JEPA on the JetClass dataset (100M jets) **without using any truth labels**
- Once pre-trained, the **encoder** can be viewed as a **foundation model**
 - transfer learning to specific downstream tasks



TRANSFER LEARNING: JET TAGGING

- Benchmark: 10-class jet classification on JetClass



FineTune:

Encoder allowed to be slightly updated when trained with labelled jets for tagging

Freeze:

Encoder fixed when trained with labelled jets for tagging

FromScratch:

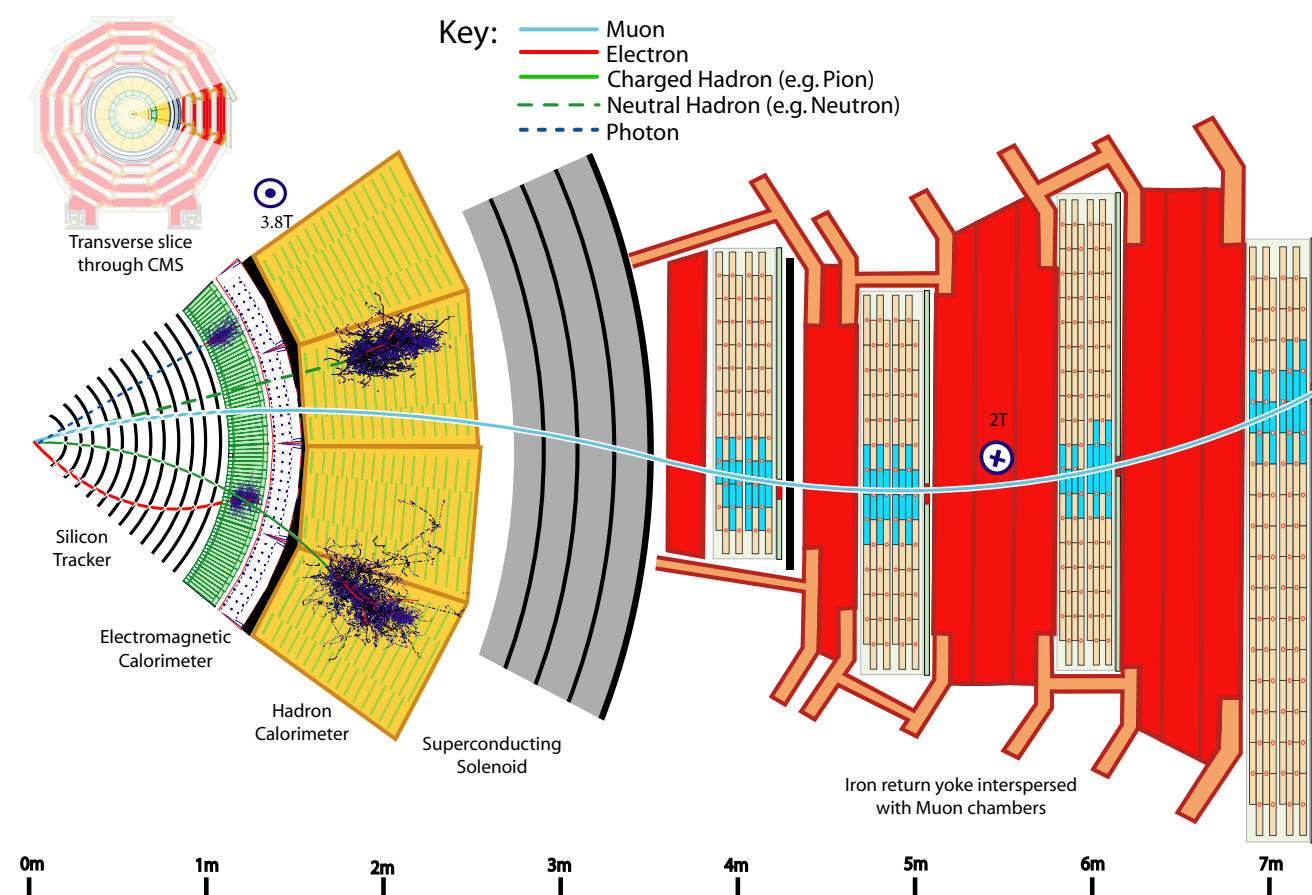
Same network architecture, but trained with labelled jets starting from randomly initialized weights

Pre-training + transfer learning shows a significant performance boost when **labelled samples are limited**.

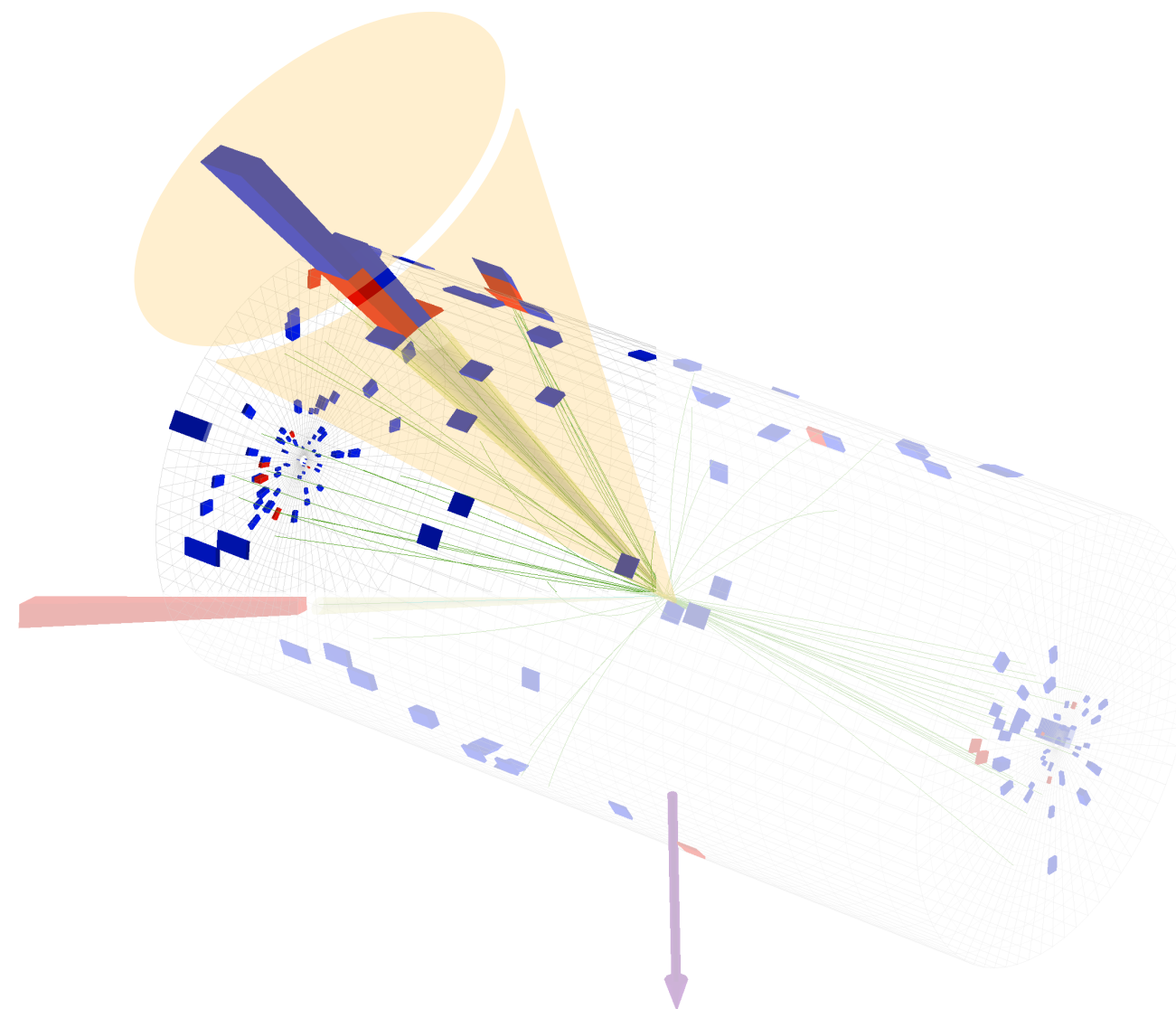
OUTLOOK: VERTICAL SCALING

- **At colliders:** event reconstruction usually follows a hierarchical way
 - due to heterogeneous sub-detectors and high particle multiplicity
- **Going beyond: a foundation model that aligns data representations at various levels and cover all the stages?**
 - requires a multi-modal approach and efficient architectures

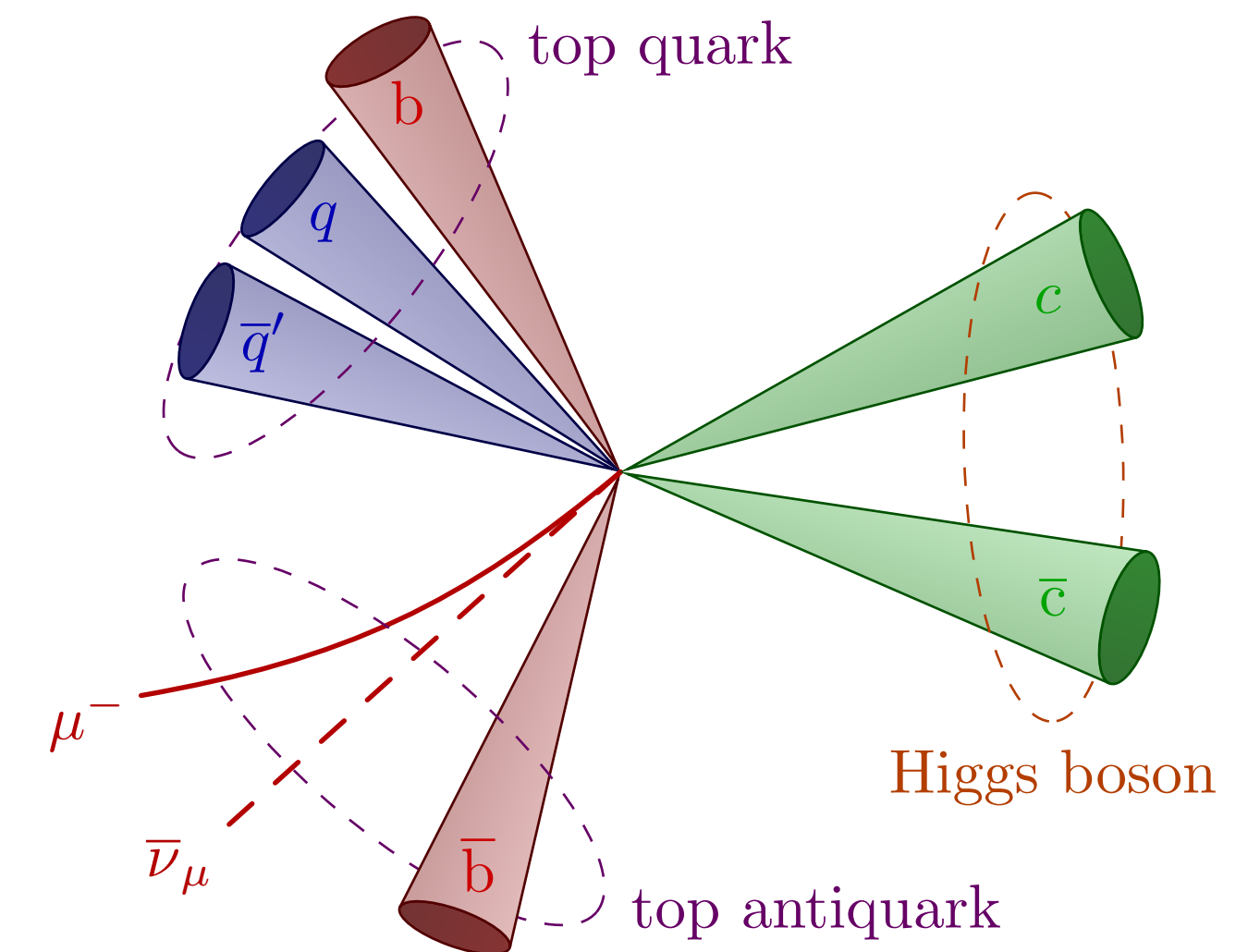
Detector/Particle-flow reconstruction



Jet/Object reconstruction



Event classification/interpretation

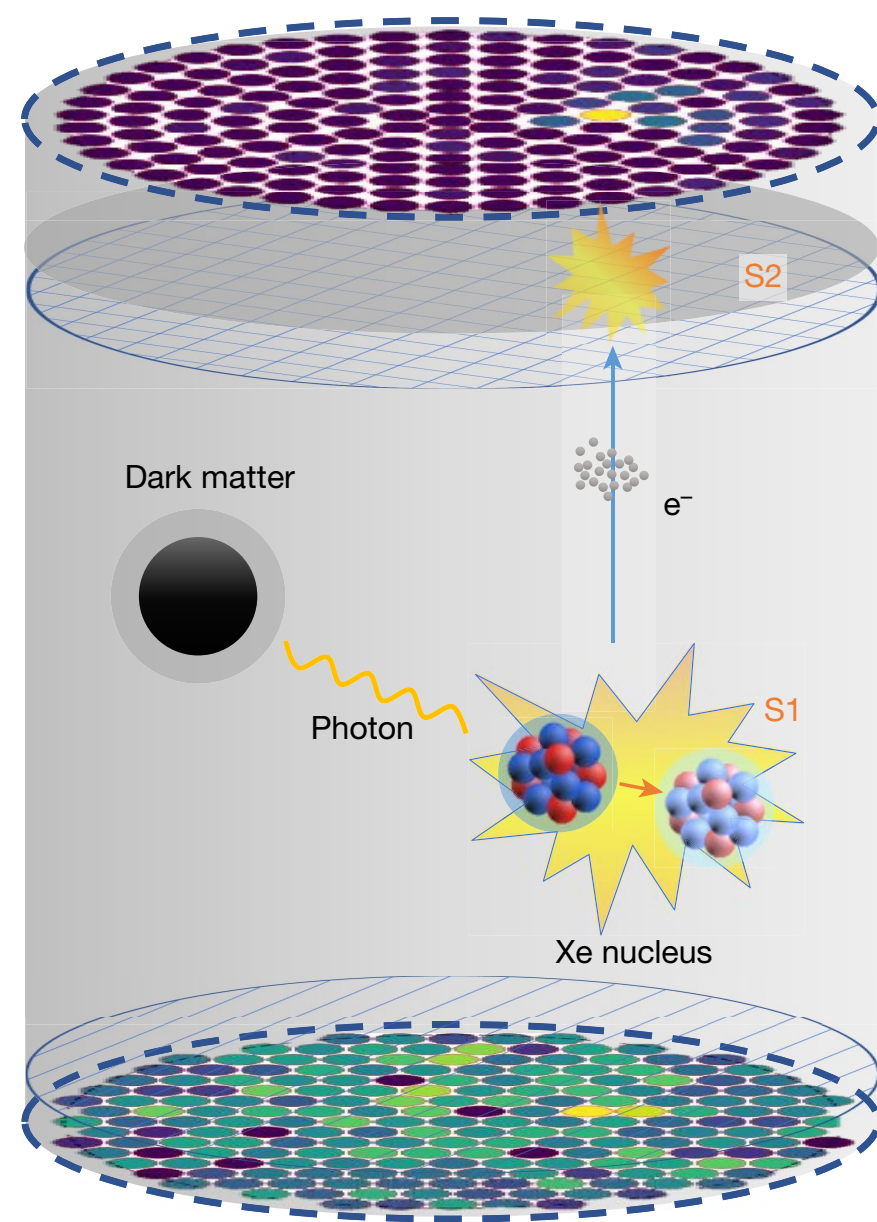


Raw detector hits => Tracks, Showers, Particles => Jets, leptons, taus => Signal vs background processes

OUTLOOK: HORIZONTAL SCALING

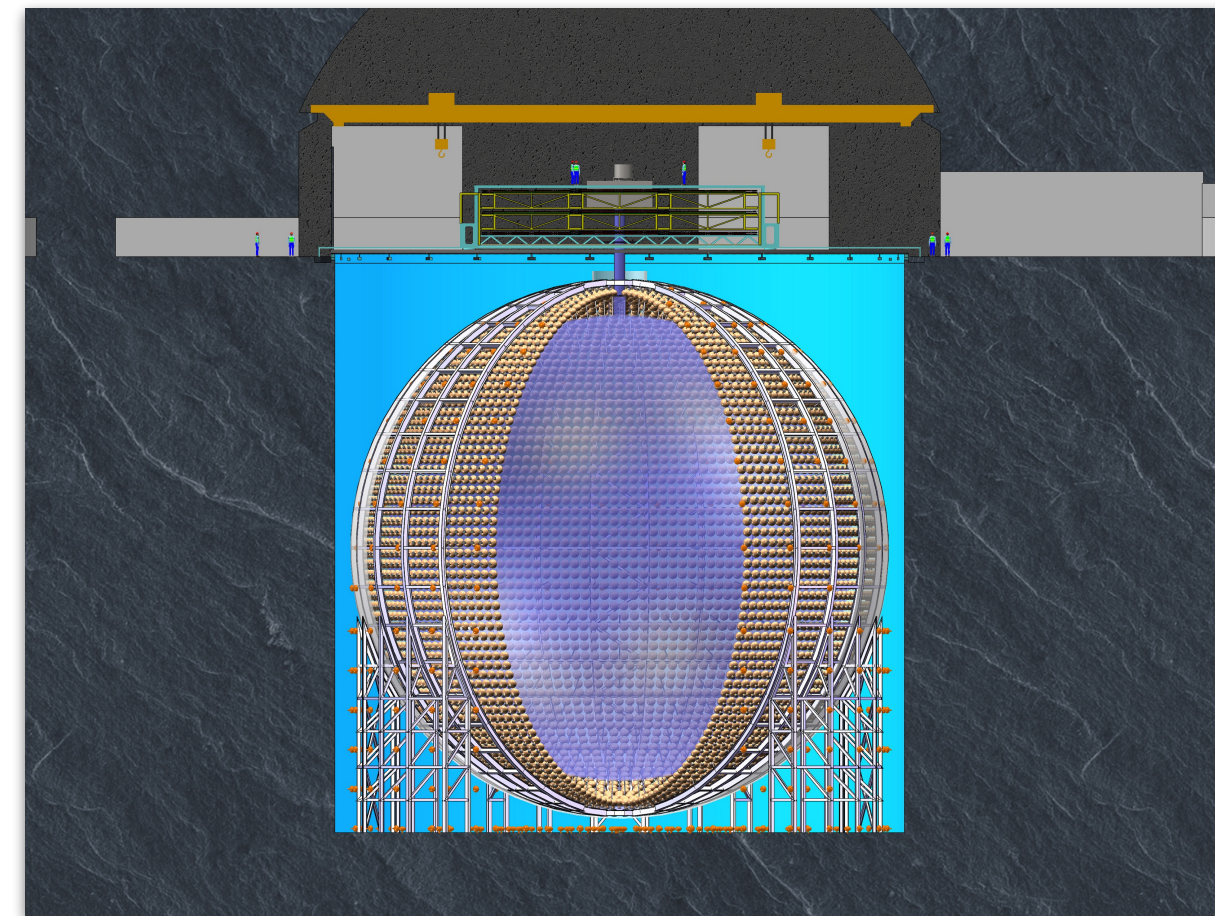
- **Dark matter/neutrino/cosmic ray experiments**
 - typically homogeneous detector — end-to-end ML reconstruction more promising
- **Going beyond: a foundation model that generalizes across experiments and covers all energy scales?**

PandaX

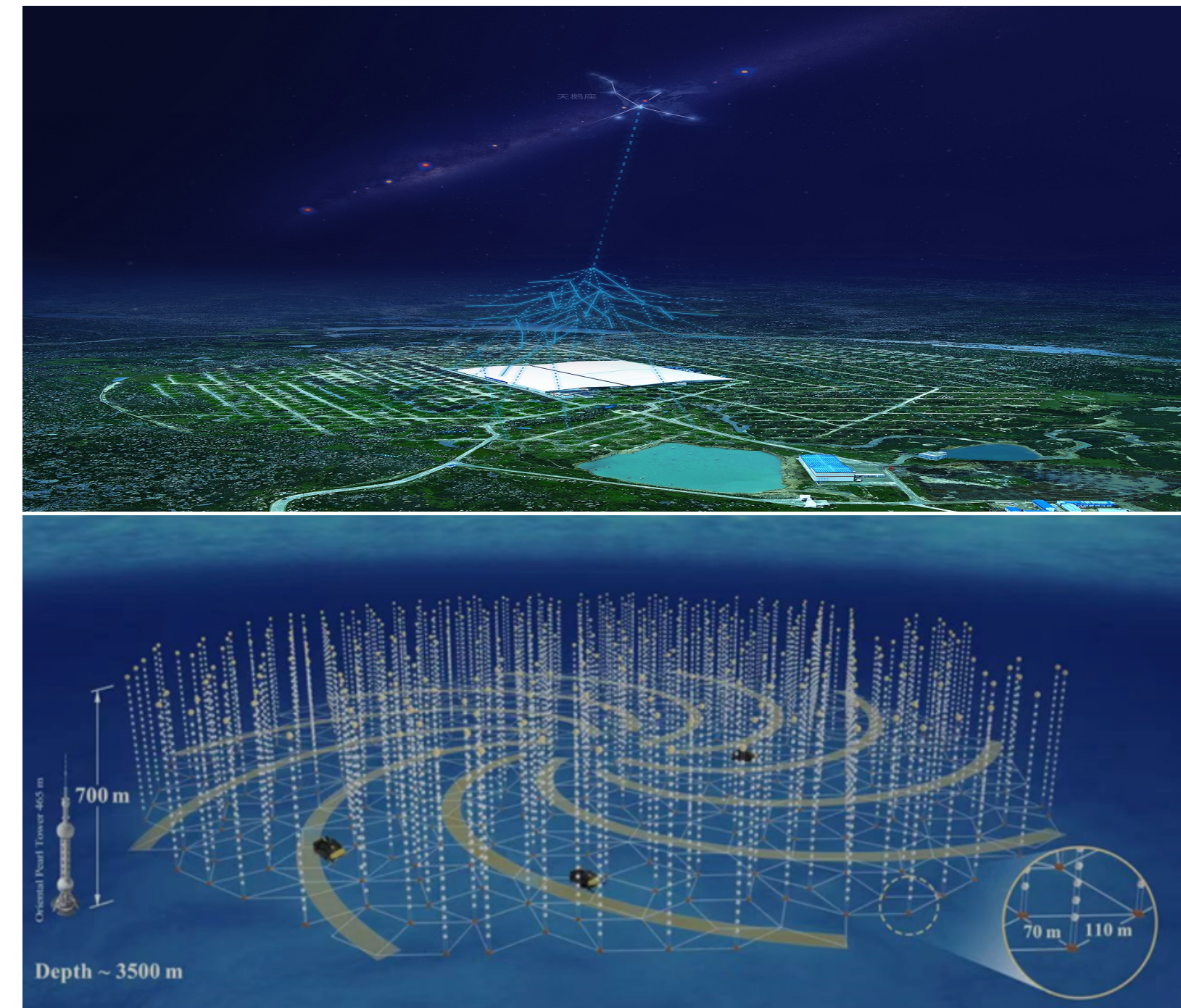


*Dark Matter/Neutrino
(keV~MeV)*

JUNO



*Neutrino
(MeV~GeV)*

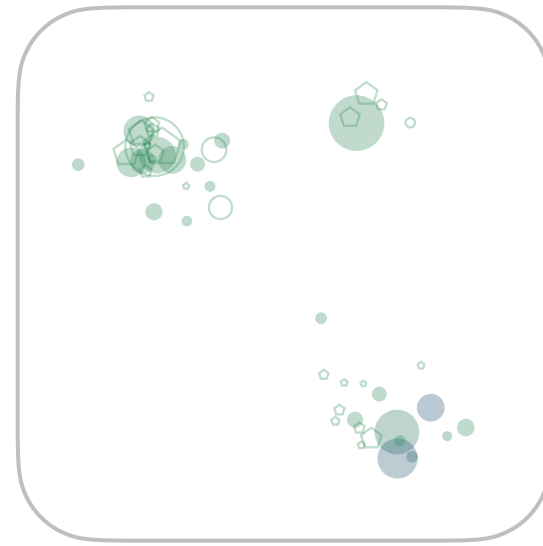


LHAASO

TRIDENT

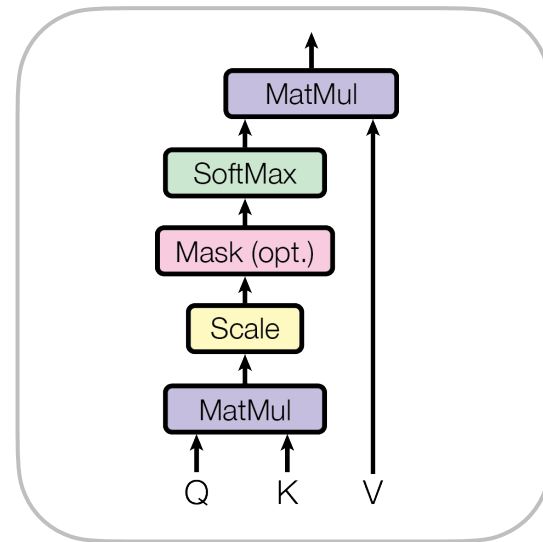
*Cosmic Rays/Neutrinos
(TeV~EeV)*

SUMMARY



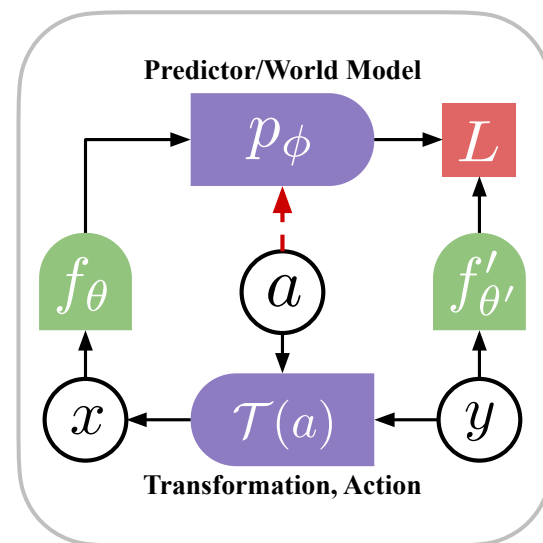
Evolution of Jet Representation

... leads to significant performance gains.



Attention Isn't All You Need

... more physics = fast learning & better generalization.



Towards a Foundation Model

... efficient and scalable representation learning is the key!