Kobayashi-Maskawa Institute
Nagoya University

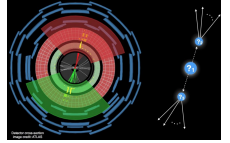
OmniLearn: A Method to Simultaneously Facilitate All Jet Physics Tasks and beyond

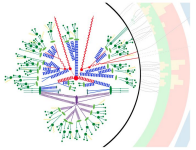

vmikuni@hepl.phys.nagoya-u.ac.jp

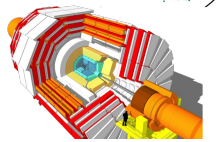

vinicius-mikuni

Vinicius M. Mikuni

Strategy



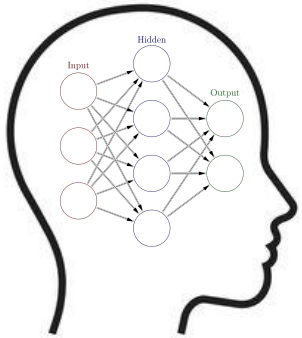


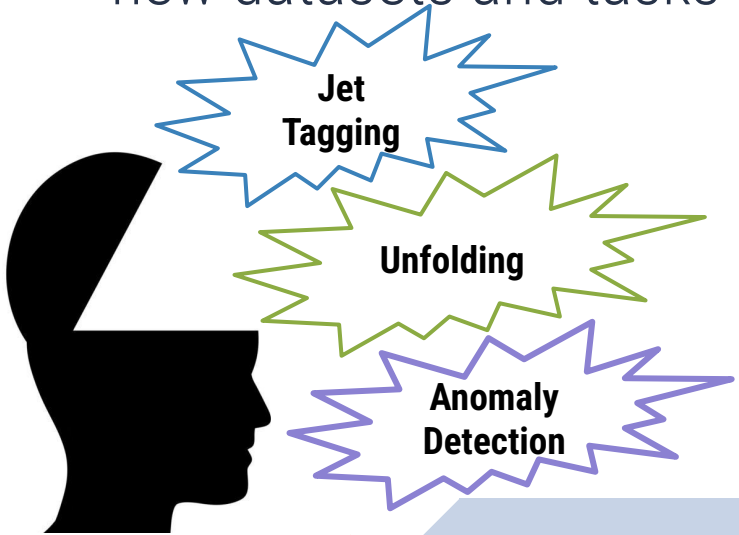


Model starts with random weights

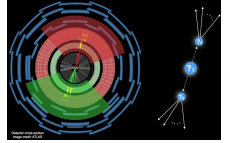
Ask the model to solve important tasks using particle collisions

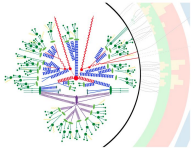
Fine-tune the model on new datasets and tasks

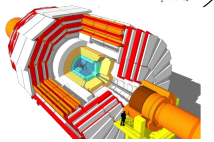




Jets

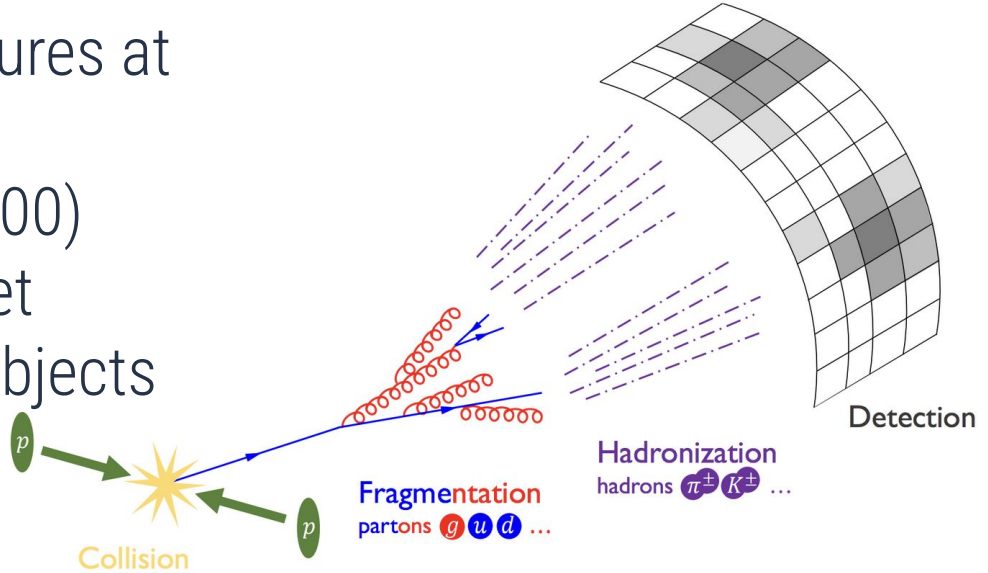






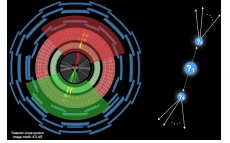
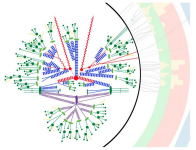
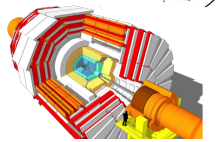
Jets are the most common signatures at the LHC

- Complicated signature: O(10-100) objects are clustered in each jet
- Choice of data: Particle Flow objects associated to jets



- Choice of data representation: **Point Clouds**

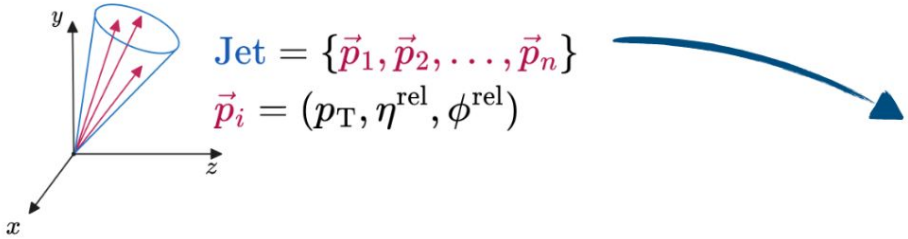
Data

Option 1: Tokenization

- Inspired by LLMs, tokenize the information of input particles

Jet constituents with **continuous features**



Constituents are tokenized with a VQ-VAE
(Similar to MPMv1 arXiv:2401.13537)
Since we use a language-model like approach, where the data is represented as a sequence of integer tokens

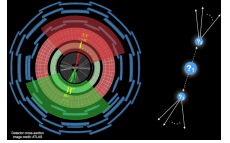
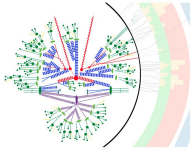
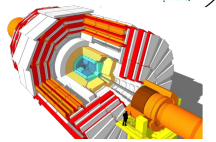
$$\text{Jet} = \{\texttt{start-token}, \texttt{token}_1, \ldots, \texttt{token}_n, \texttt{end-token}\}$$
$$\texttt{token}_i = \text{integer value} \in [1, \ldots, 8192]$$

From Joschka's slides

See:

- Kishimoto, T., Morinaga, M., Saito, M., & Tanaka, J. (2023). arXiv:2312.06909.
- Golling, T., Heinrich, L., Kagan, M., Klein, S., Leigh, M., Osadchy, M., & Raine, J. A. (2024). MLST, 5(3), 035074.
- Birk, J., Hallin, A., & Kasieczka, G. (2024). MLST. 5 (2024) 3, 035031
- Birk, J., Gaede, F., Hallin, A., Kasieczka, G., Mozzanica, M., & Rose, H. (2025). arXiv:2501.05534.
- Katel, S., Li, H., Zhao, Z., Kansal, R., Mokhtar, F., & Duarte, J. (2024). arXiv:2412.05333.

Data

Option 2: Point Cloud

- Features given as is: avoid loss of information, more natural for HEP

Is Tokenization Needed? *2409.12589*

Tokenizing enables **"binned" density estimation**

- K-Means clustering *(easier to train than VQ-VAE)*

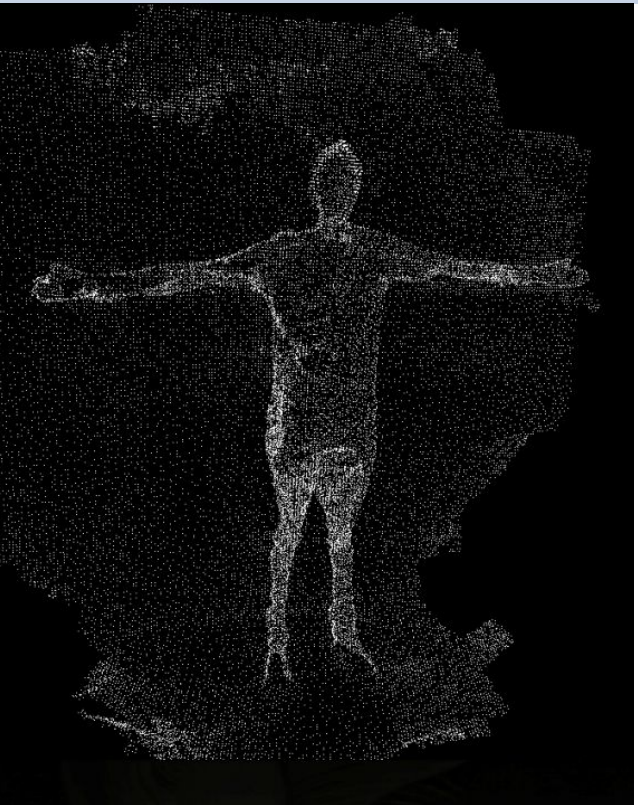
Can do **continuous density estimation** using **generative models** conditioned on unmasked data

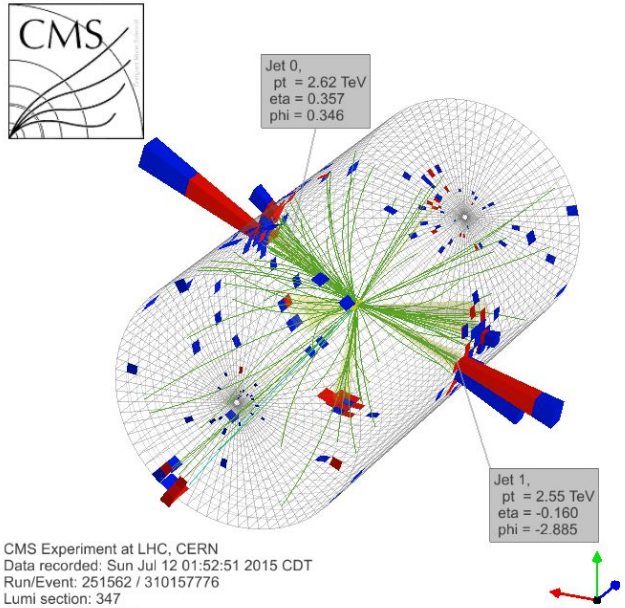
From Michael's slides

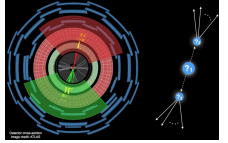
See:

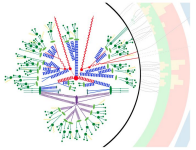
- Dillon, B. M., Kasieczka, G., Olischlager, H., Plehn, T., Sorrenson, P., & Vogel, L. (2022). SciPost Physics, 12(6), 188.
- **Mikuni, V.,** & Nachman, B. (2024). arXiv:2404.16091.
- Li, C., Agapitos, A., Drews, J., Duarte, J., Fu, D., Gao, L., ... & Li, Q. (2024). arXiv:2405.12972.
- Harris, P., Kagan, M., Krupa, J., Maier, B., & Woodward, N. (2024). arXiv:2403.07066.
- Vigl, M., Hartman, N., & Heinrich, L. (2024). arXiv:2401.13536.

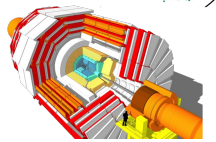
Point Clouds





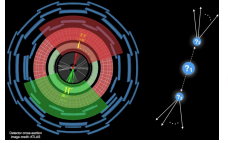


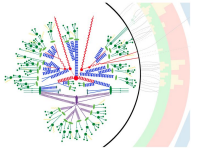


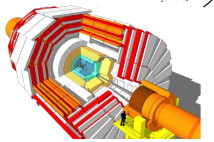


Particle collisions are naturally represented as **point clouds**: unordered set of objects in a metric space

Jets

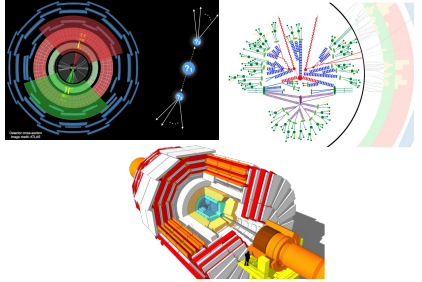


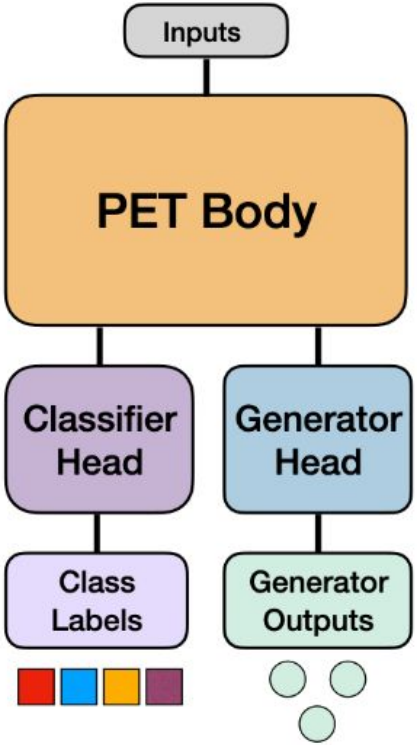




How to teach AI about jets?

Encoding jet information



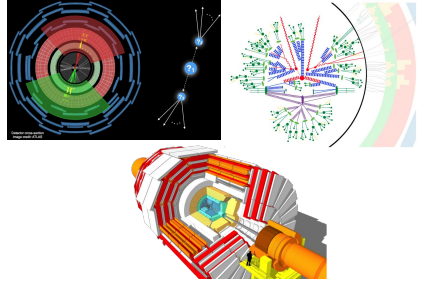


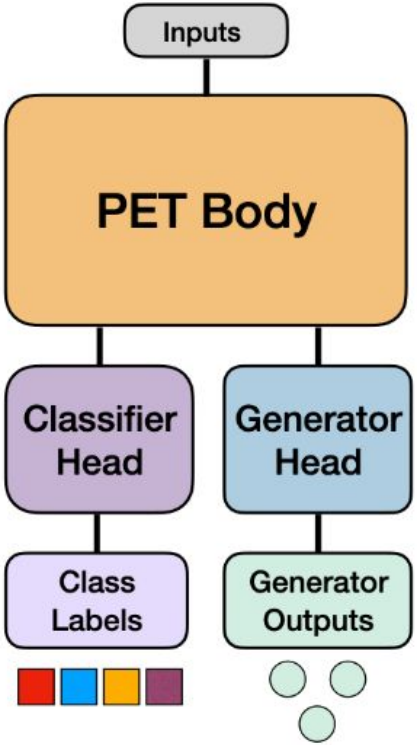
Create a neural network to accomplish 2 tasks:

- **Classify jets**: learns the difference in radiation signature between jet types
- **Generate jets**: implicitly learn the likelihood of jets for different initial partons

V. Mikuni, B. Nachman, Phys. Rev. D 111, L051504

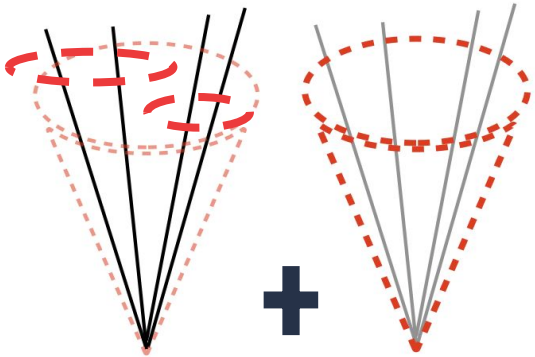
Encoding jet information



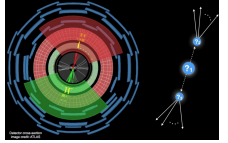


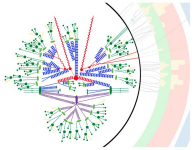
Point-**E**dge **T**ransformer (**PET**)

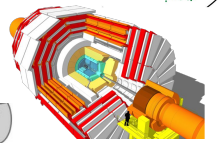
- Combine local information with graphs
- Learn global information with Transformers:
  3M parameters

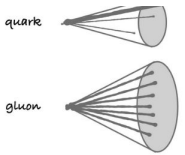


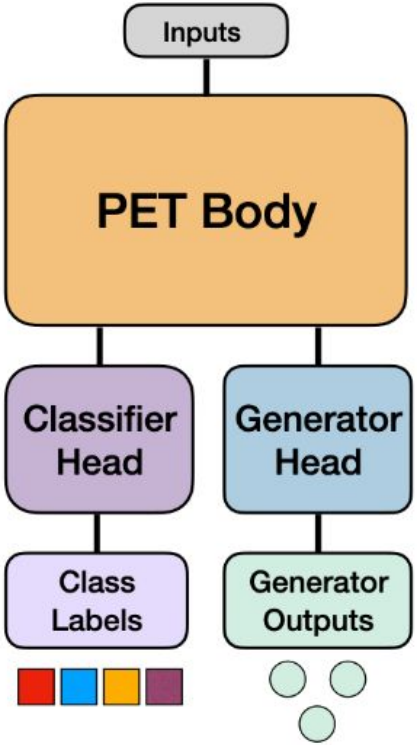
Encoding jet information









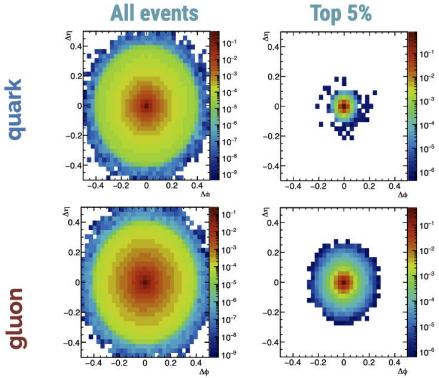


ABCNet application: classification of events

- **Higher background rejection** for the available signal thresholds compared to other methods

	Acc	AUC	$1/\epsilon_B$ ($\epsilon_S = 0.5$)	$1/\epsilon_B$ ($\epsilon_S = 0.3$)	Parameters
ResNeXt-50	0.821	0.960	30.9	80.8	1.46M
P-CNN	0.827	0.9002	34.7	91.0	348k
PFN	-	0.9005	34.7±0.4	-	82k
ParticleNet-Lite	0.835	0.9079	37.1	94.5	**26k**
ParticleNet	**0.840**	0.9116	39.8±0.2	98.6±1.3	366k
ABCNet	**0.840**	**0.9126**	**42.6±0.4**	**118.4±1.5**	230k

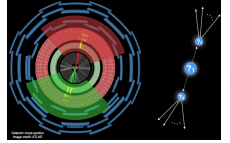
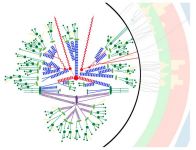
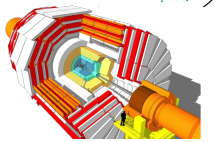
- Can we **look** at what **ABCNet** is **learning**?
  - Look at the **self-attention coefficients**
  - Only show the top **5%** **particles** with **highest self-attention coefficients**



Modern version of ABCNet, **the first transformer in HEP**

Mikuni, V., Canelli, F. *Eur. Phys. J. Plus* **135**, 463 (2020)

Training

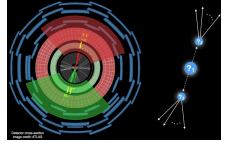
JetClass dataset used for training

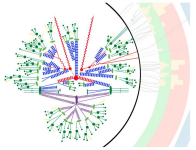
- 100M jets
- **10 different jet categories, AK8 jets simulated in pp collisions with Madgraph + Pythia8 with CMS Delphes detector simulation**

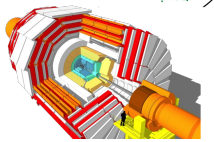
Use the pre-trained model as the starting point and fine-tune using different datasets

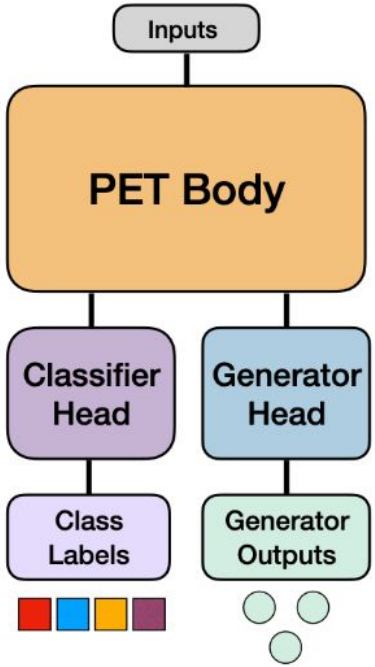
Huilin Qu, Congqiao Li, Sitian Qian, arXiv:2202.03772

OmniLearned





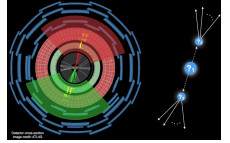
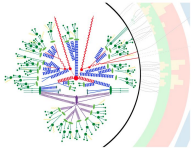
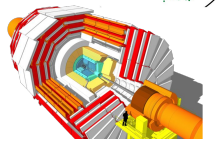




OmniLearned: Combine **Supervised** and **Unsupervised** Learning in the same model!

More details at: W. Bhimji, C. Harris, **V. Mikuni,** B. Nachman, arXiv:2510.24066

1 Billion Dataset

Dataset	Training	Validation	Test
JetClass [14]	100M	20M	5M
JetClass2 [33]	200M	600k	600k
Aspen Open Jets [34]	125M	25.7M	26.6M
ATLAS Top Tagging [35]	178M	20M	2.2M
H1 DIS	42.2M	872k	255k
CMS QCD	239M	17.5M	16M
CMS BSM	173.5M	17M	17M
Total	1057.7M	101.8M	67.6M

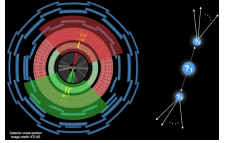
We combine multiple open datasets in HEP in a **single and ML-ready format**

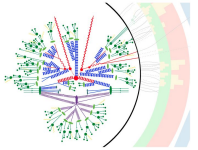
More than **1 Billion Jets** are accessible directly from the software package

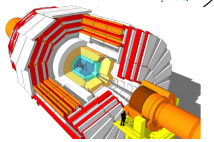
- 210 Jet Classes
- Mixture of simulations with real experimental data

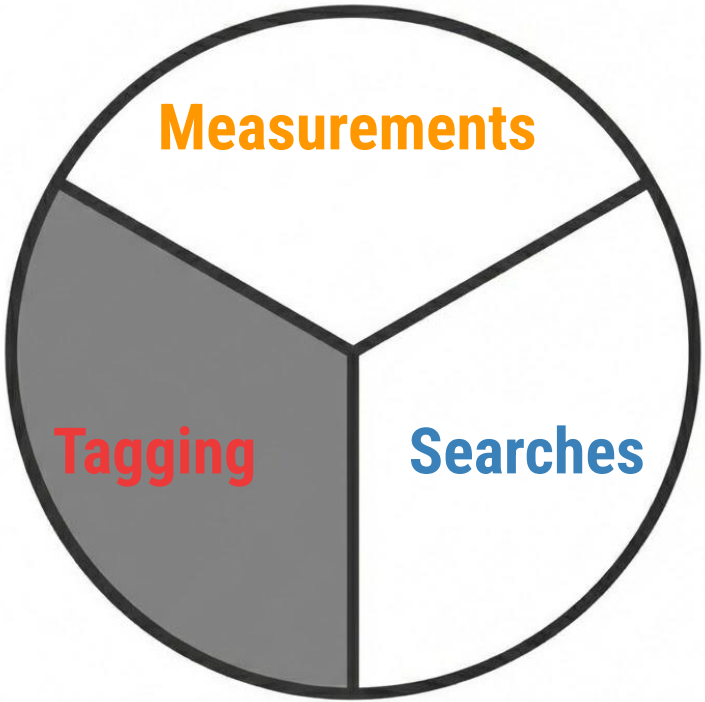
“Application Highlight

Applications

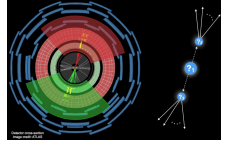


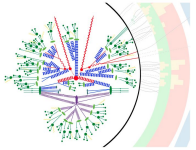


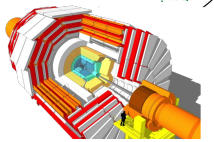




Jet Classification: Top Tagging



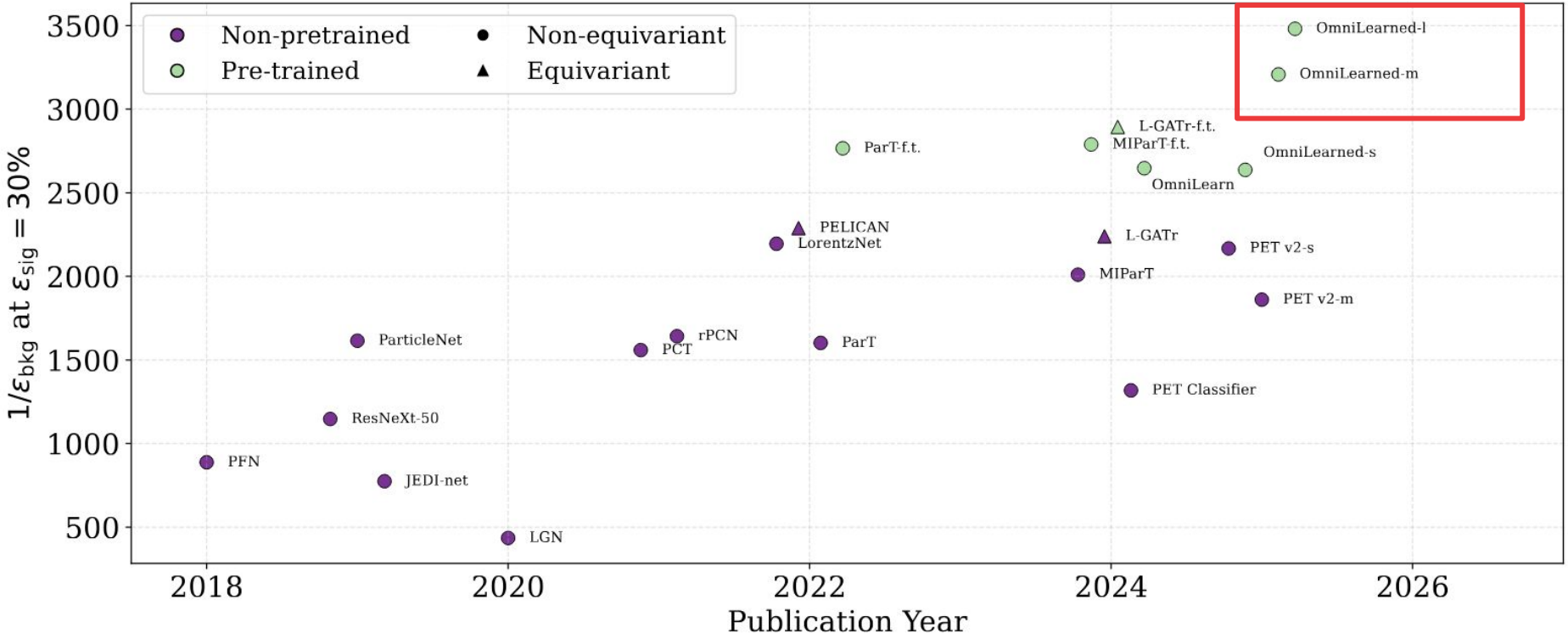




	Acc	AUC	$1/\epsilon_B$	
			$\epsilon_S = 0.5$	$\epsilon_S = 0.3$
ResNeXt-50 [19]	0.936	0.9837	302 ± 5	1147 ± 58
P-CNN [19]	0.930	0.9803	201 ± 4	759 ± 24
PFN [16]	-	0.9819	247 ± 3	888 ± 17
ParticleNet [19]	0.940	0.9858	397 ± 7	1615 ± 93
JEDI-net [18]	0.9300	0.9807	-	774.6
PCT [22]	0.940	0.9855	392 ± 11	1559 ± 98
LGN [63]	0.929	0.964	-	435 ± 95
rPCN [20]	-	0.9845	364 ± 9	1642 ± 93
LorentzNet [64]	0.942	0.9868	498 ± 18	2195 ± 173
PELICAN [65]	0.9425	0.9869	-	2289 ± 204
ParT [14]	0.940	0.9858	413 ± 16	1602 ± 81
ParT-f.t. [14]	**0.944**	0.9877	**691 ± 15**	2766 ± 130
Mixer [66]	-	0.9859	416	-
MIParT [26]	0.942	0.9868	505 ± 8	2010 ± 97
MIParT-f.t. [26]	0.944	0.9878	640 ± 10	2789 ± 133
L-GATr [67]	0.9423	0.9870	540 ± 20	2240 ± 70
L-GATr-f.t. [67]	**0.9446**	0.9879	651 ± 11	2894 ± 84
PET [11, 12]	0.938	0.9848	340 ± 12	1318 ± 39
OmniLearn [11, 12]	0.942	0.9872	568 ± 9	2647 ± 192
PET v2-s	0.9427	0.987	505 ± 14	2167 ± 153
OmniLearned-s	0.944	0.9875	565 ± 12	2637 ± 128
PET v2-m	0.9423	0.987	482 ± 11	1861 ± 61
OmniLearned-m	**0.944**	**0.9880**	656 ± 12	3208 ± 176
OmniLearned-l	**0.944**	**0.9880**	**688 ± 9**	**3486 ± 157**

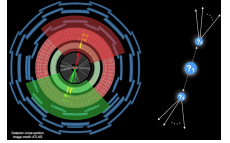
- **Fine-tune the pre-trained model** on the Top Tagging Community Dataset, a traditional benchmark dataset in HEP

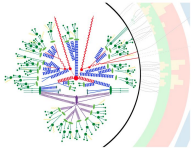
OmniLearned achieves SOTA performance

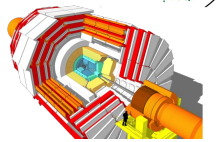


From: A. Petitjean, T. Plehn, J. Spinner, U. Köthe: arXiv:2512.17011

FastSim to FullSim: Top Tagging





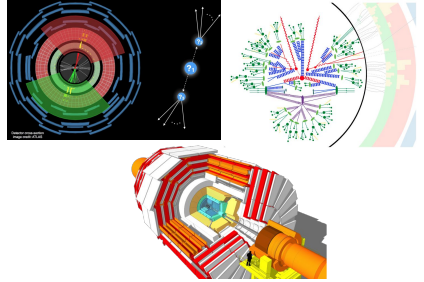


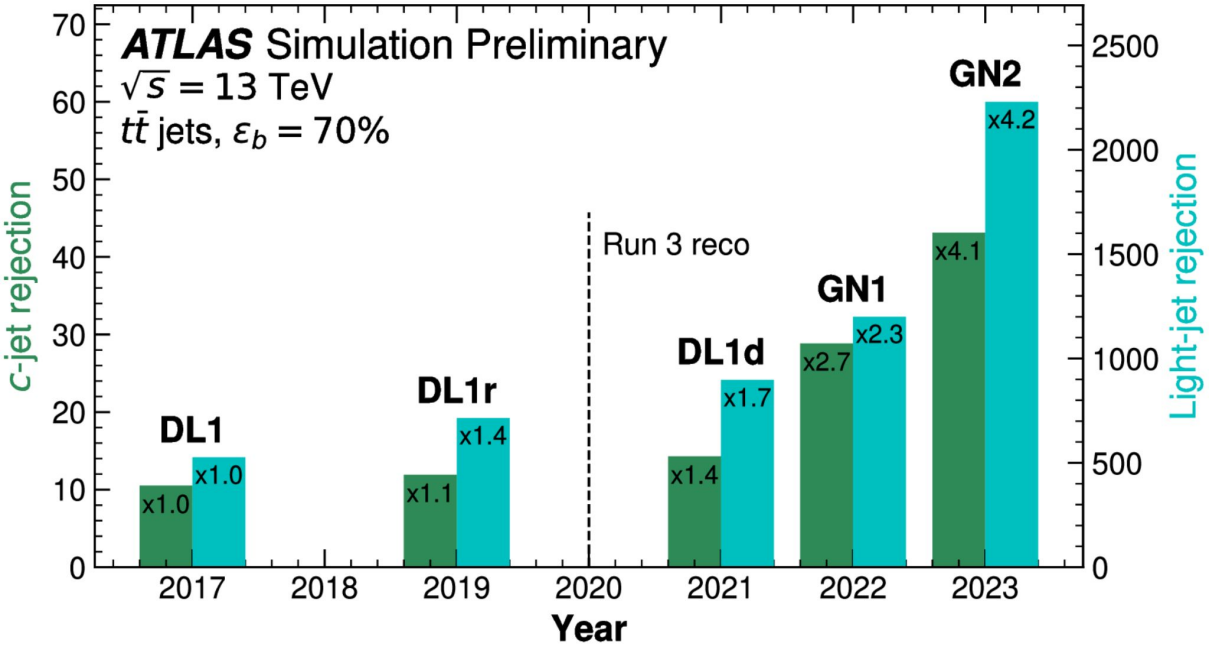
OmniLearn is trained on cheap Delphes simulations. Can we fine-tune to **Run 2 ATLAS** Full simulation + Reconstruction?

- Matches SOTA with **10%** of the data
- Improves on SOTA if all events are used

	AUC	Acc	$1/\epsilon_B$	
			$\epsilon_S = 0.5$	$\epsilon_S = 0.8$
ResNet 50	0.885	0.803	21.4	5.13
EFN	0.901	0.819	26.6	6.12
hlDNN	0.938	0.863	51.5	10.5
DNN	0.942	0.868	67.7	12.0
PFN	0.954	0.882	108.0	15.9
ParticleNet	0.961	0.894	153.7	20.4
PET classifier (4M)	0.959	0.890	146.5	19.4
OmniLearn (4M)	0.961	0.894	172.1	20.8
PET classifier (40M)	0.964	0.898	201.4	23.6
OmniLearn (40M)	**0.965**	**0.899**	**207.30**	**24.10**

Jet Classification: Flavor Tagging

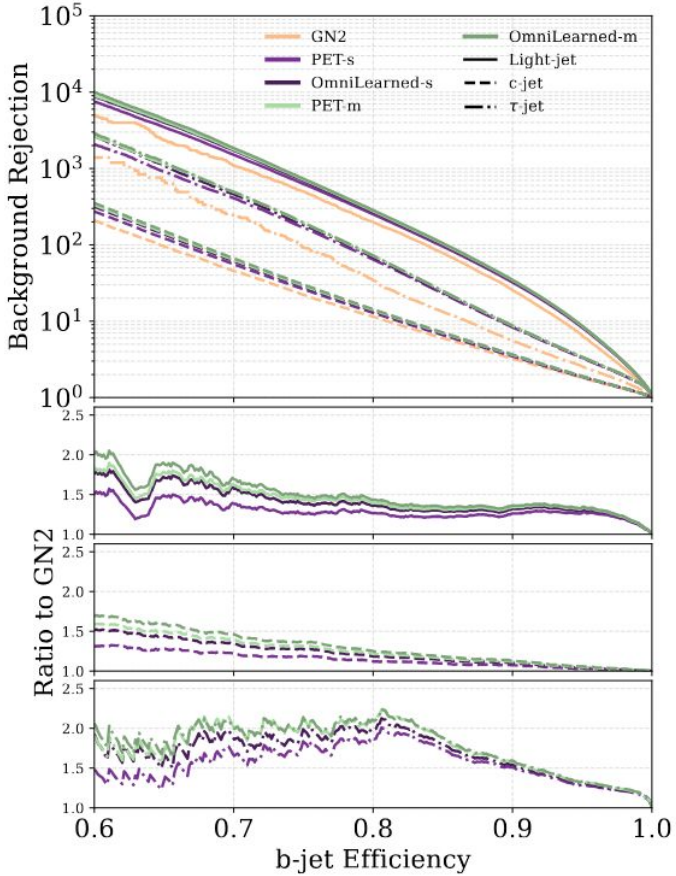


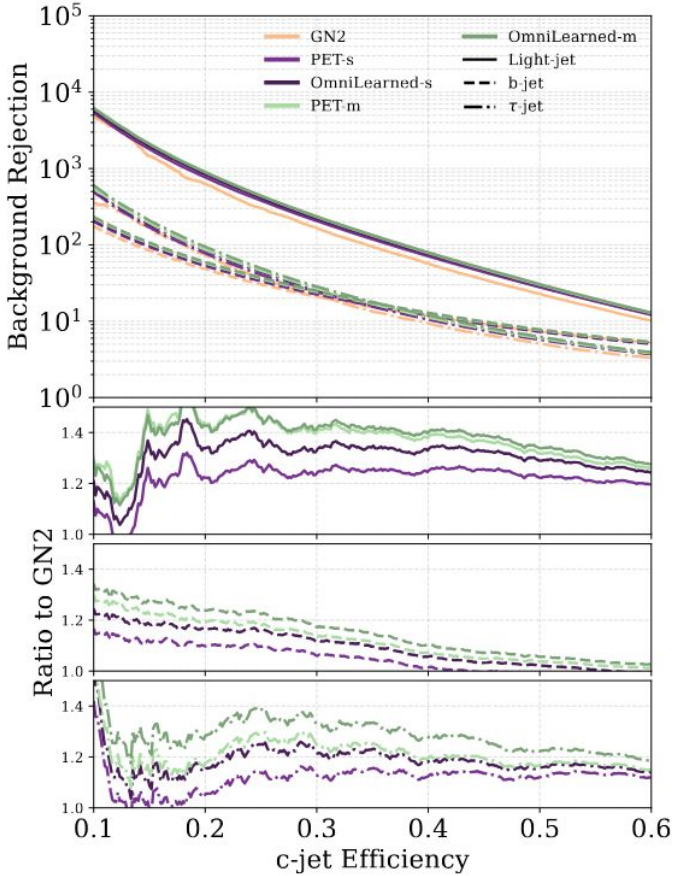


Fine-tune the pre-trained model on the ATLAS Flavour Tagging Dataset

- Non-trivial transfer with different detector and more **realistic simulation**

OmniLearned achieves SOTA performance

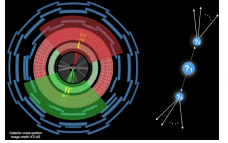


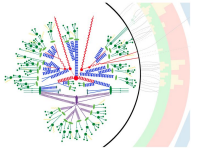


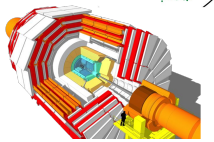
GN2 is the current algorithm used in ATLAS, trained on **168M Jets**

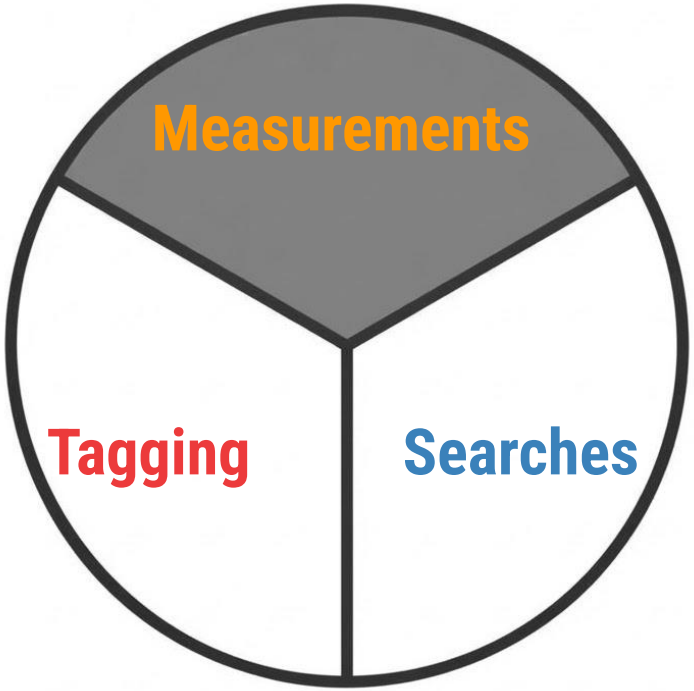
- **OmniLearned** still improves performance

Applications

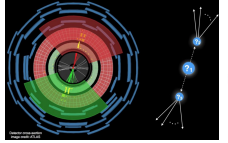


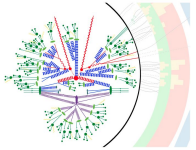


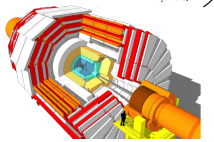




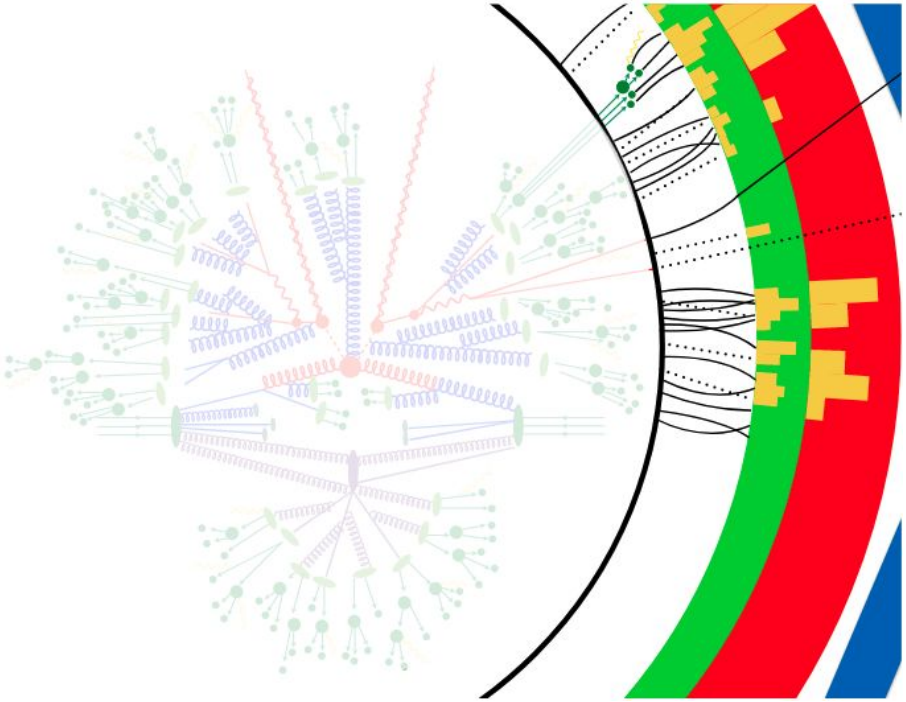
Unfolding



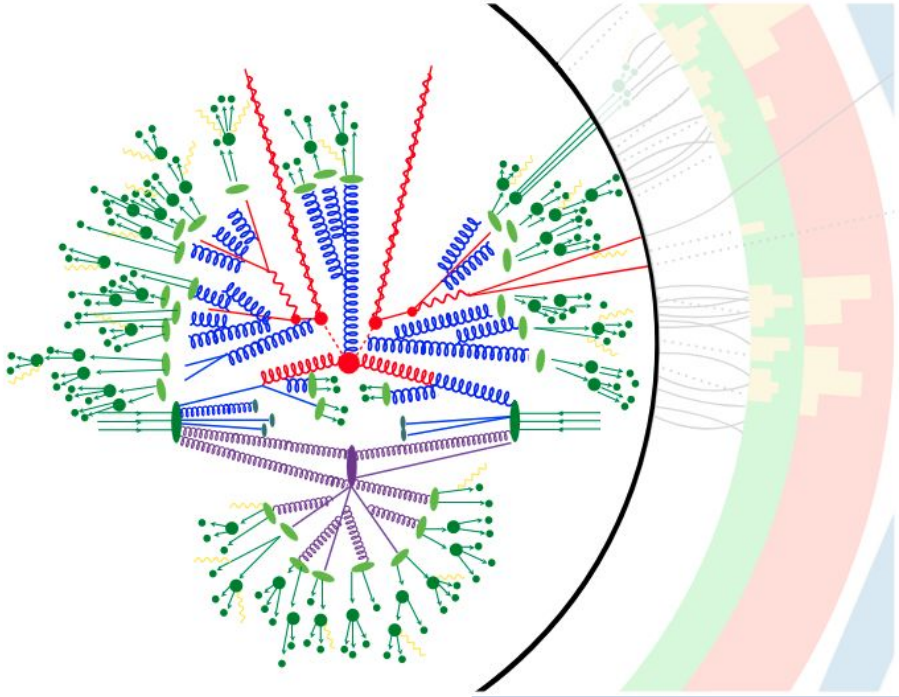




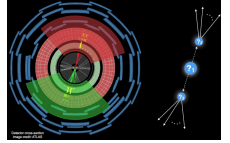
What we measure

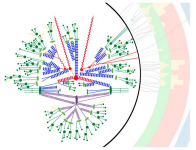


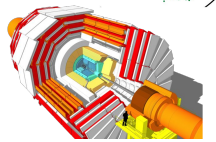
What we want

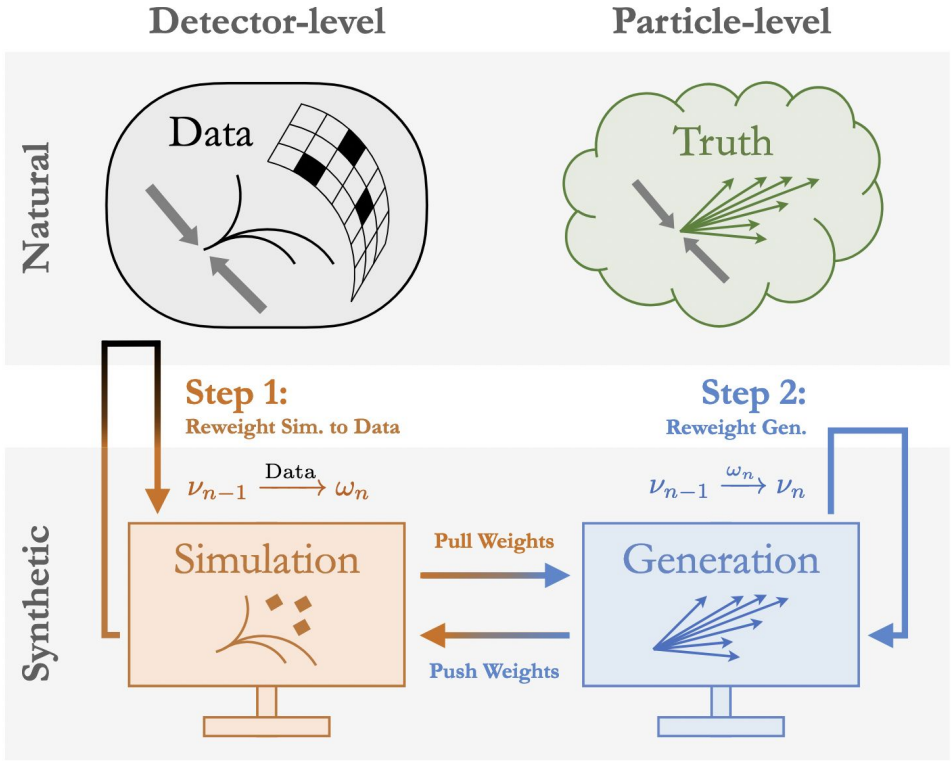


OmniFold









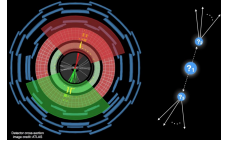
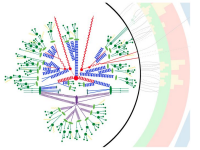
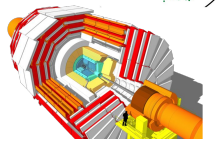
2-step iterative process

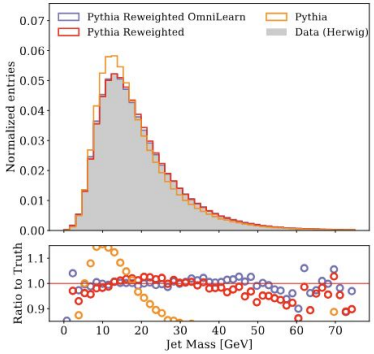
- **Step 1**: Reweight simulations to look like data
- **Step 2**: Convert learned weights into functions of particle level objects

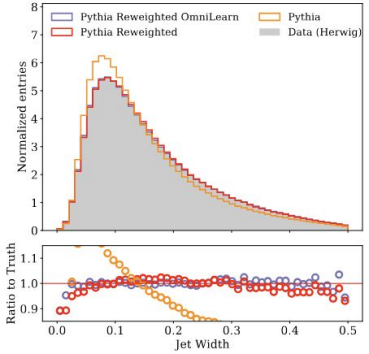
See more details in **Kyle's Talk!**

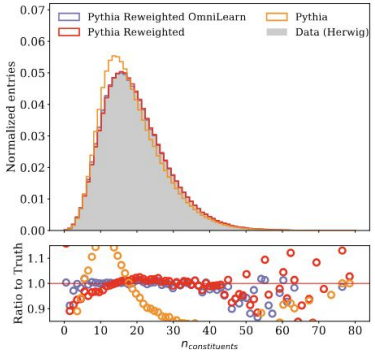
Source: Andreassen et al. PRL 124, 182001 (2020)

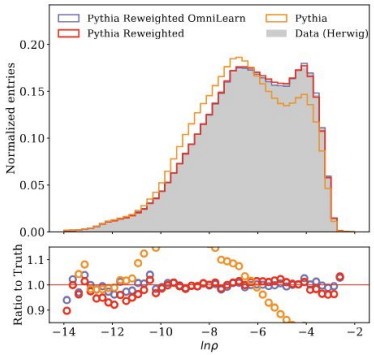
ATLAS OmniFold analysis

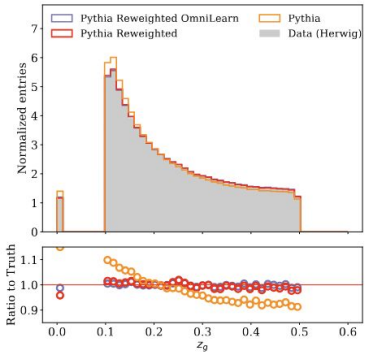




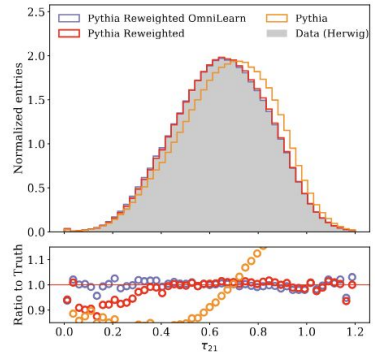






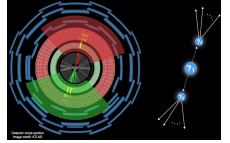


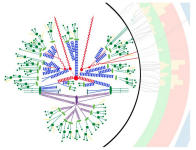


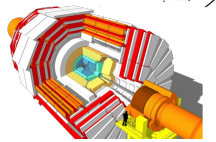


OmniFold dataset consisting of Z($\nu\nu$) + Jets events. Unfold the particles directly and then build the jet observables

Unfolding

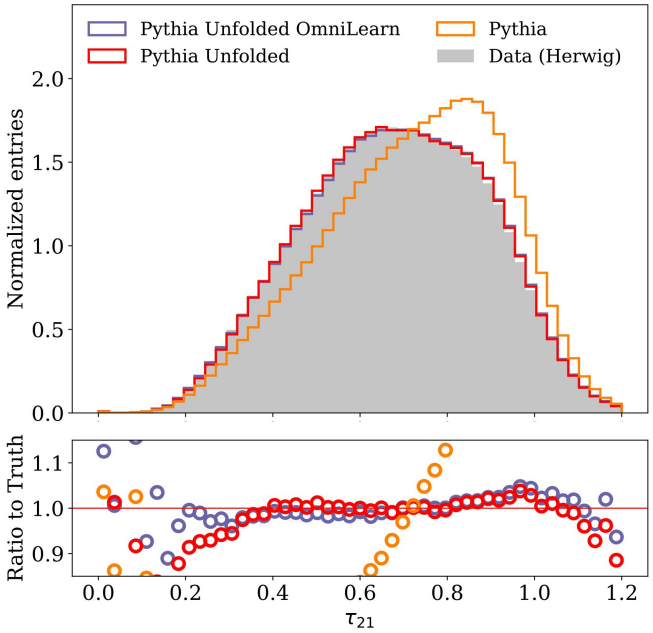






Unbinned Unfolding using the OmniFold workflow. More **precise** than traditional unfolding and more **efficient** than previous ML models

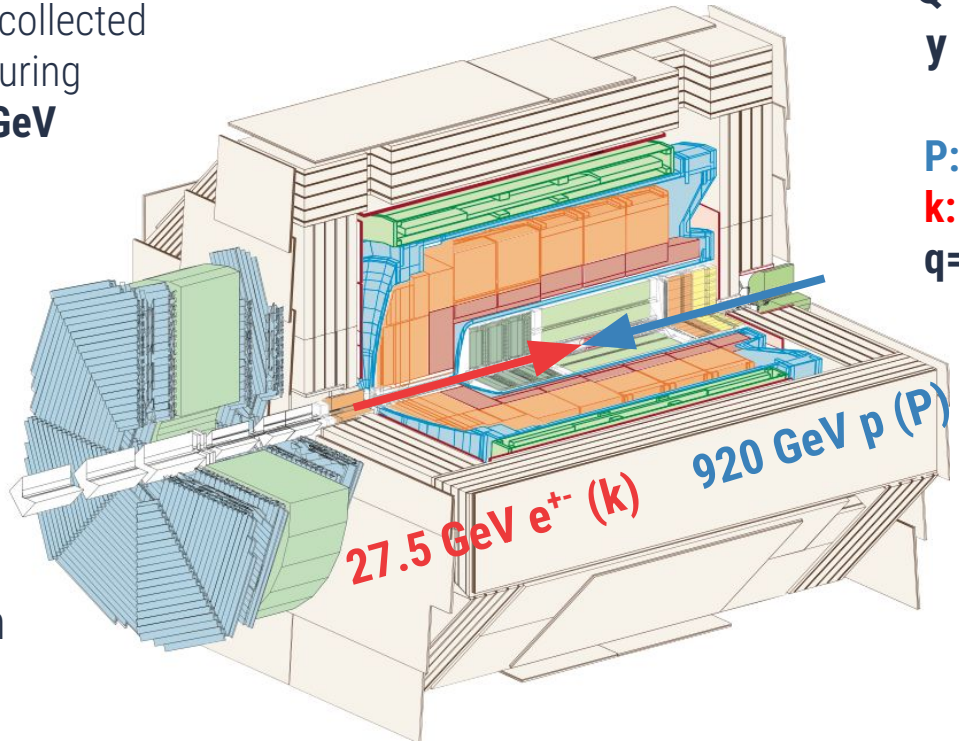
Metric	MultiFold	UniFold	IBU	OmniFold		
				DeepSets	PET classifier	OmniLearn
Jet mass	3.80	8.82	9.31	**2.77**	**2.8±0.9**	**2.6±0.8**
N	0.89	1.46	1.51	**0.33**	0.50±0.15	**0.34±0.1**
Jet Width	0.09	0.15	0.11	0.10	0.09±0.02	**0.07±0.01**
$\log \rho$	0.37	0.59	0.71	0.35	0.23±0.07	**0.14±0.03**
τ_{21}	0.26	1.11	1.10	0.53	0.13±0.03	**0.05±0.01**
z_g	**0.15**	0.59	0.37	0.68	0.19±0.03	0.21±0.04



Experimental setup

Using **228 pb^{-1}** of data collected by the **H1 Experiment** during **2006** and **2007** at **318 GeV center-of-mass energy**

Goal: Include the information of **all reconstructed particles + scattered lepton** in the collision



$Q^2 = - q^2$

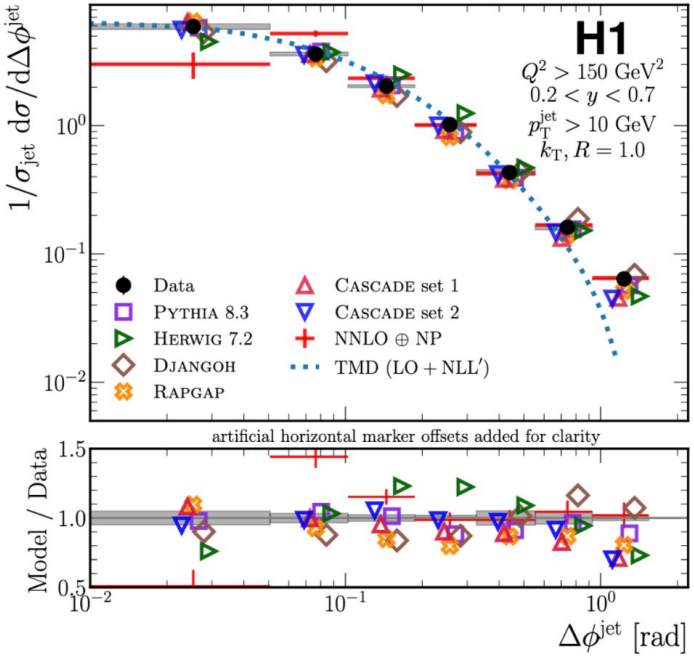
$y = Pq / pk$

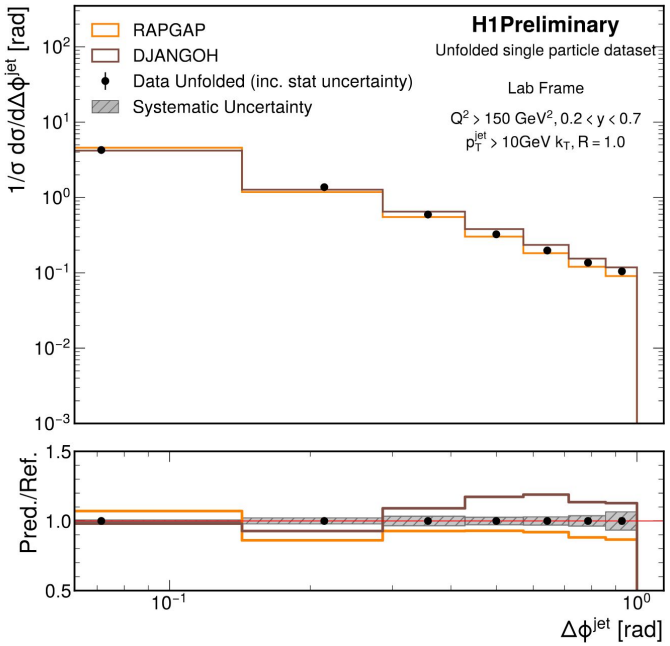
P: incoming proton 4-vector
k: incoming electron 4-vector
q=k-k' : 4-momentum transfer

Reconstructed hadrons using combined detector information: **energy flow algorithm**

Results

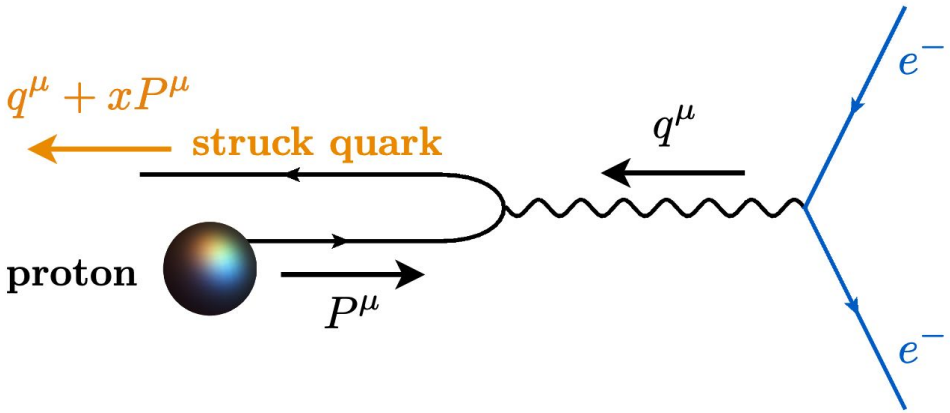
Cluster unfolded jets using kT algorithm with radius of 1.0
We are able to re-derive **past results**





[H1 Preliminary note]

Phys. Rev. Lett.(128) 132002

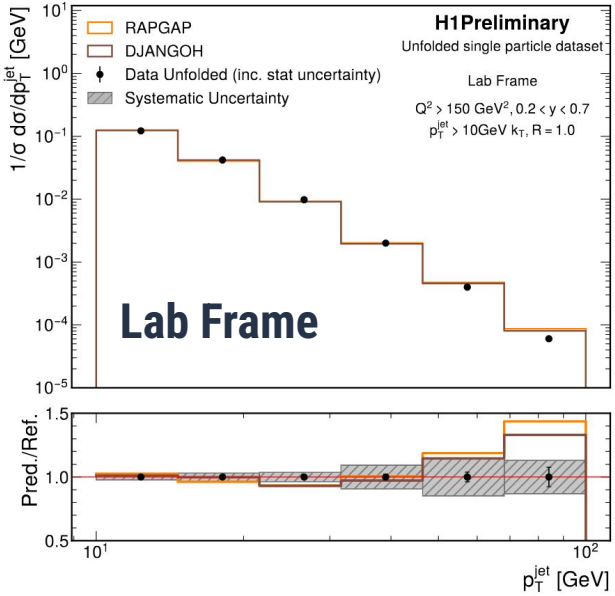


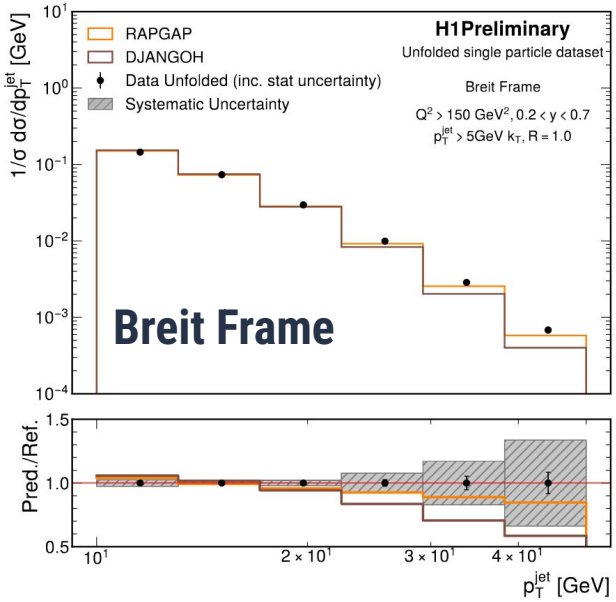
Breit Frame provides a natural frame to study ep collisions, where the struck quark forms a jet opposite from the proton beam: useful for jet and TMD studies

- Starting from the Lab frame, we need to boost the system: not trivial in terms of unfolding

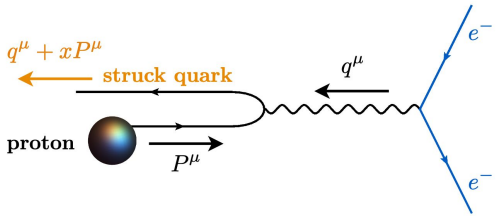
Results

Cluster jets using kT algorithm with radius of 1.0
We can study observables in **different frames**!

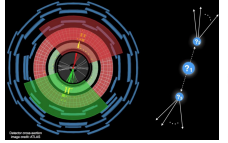


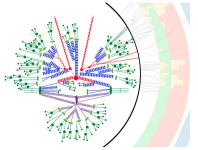


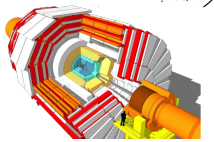
[H1 Preliminary note]

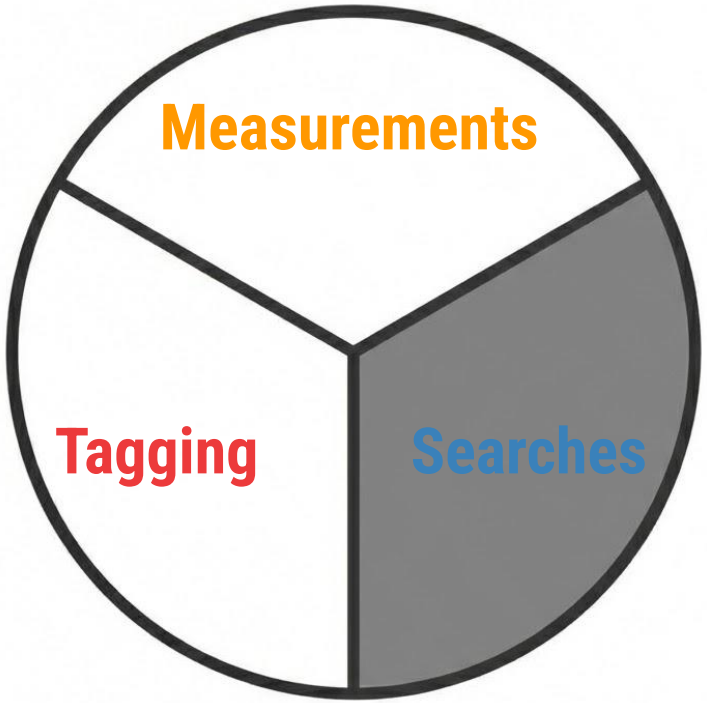


Applications



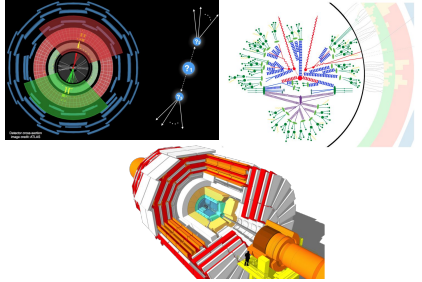






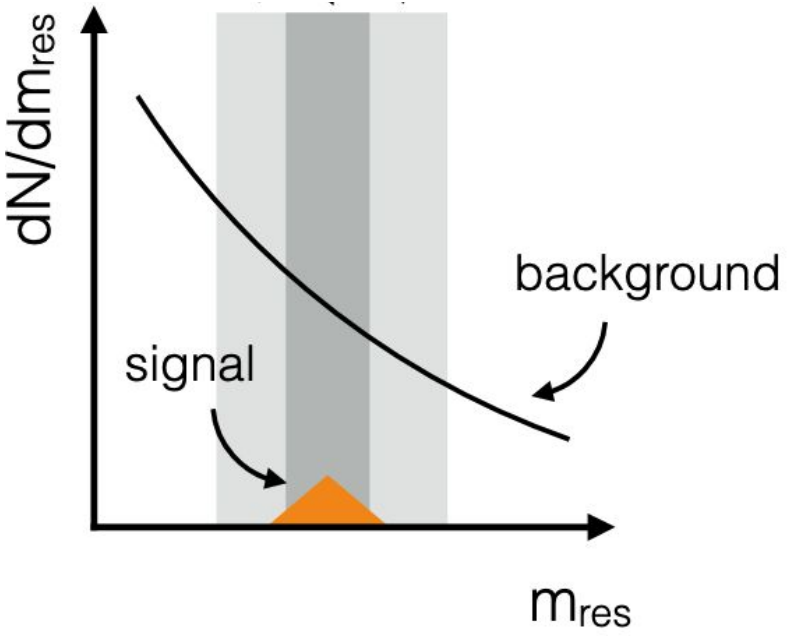
https://www.symmetrymagazine.org/article/december-2013/four-things-you-might-not-know-about-dark-matter

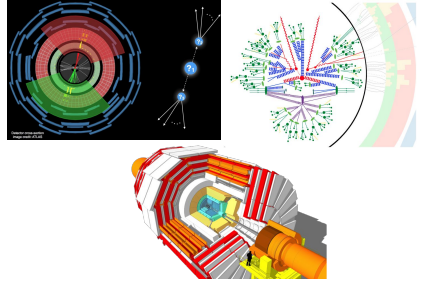
Anomaly Detection



Bump-hunting using ML:

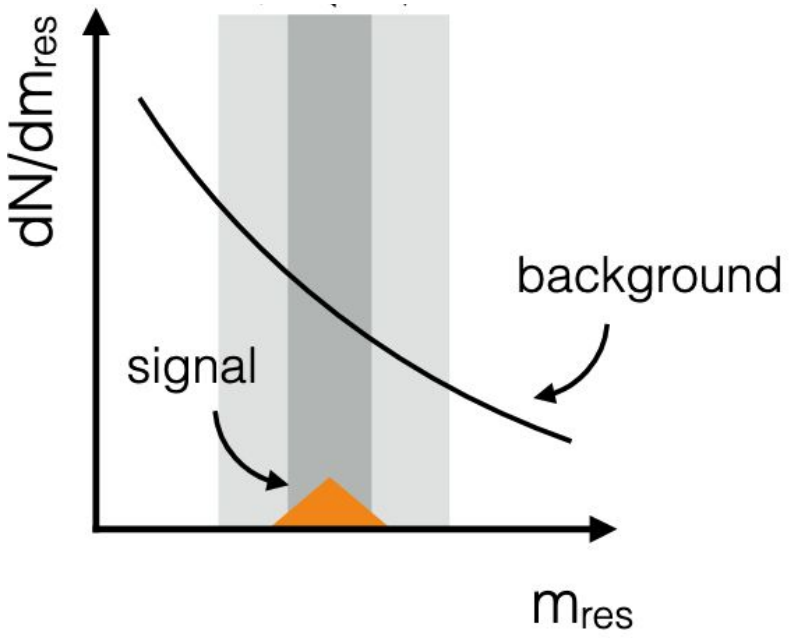
- Use the background in the sideband to estimate the background in the signal region
- Compare the estimated background with the data



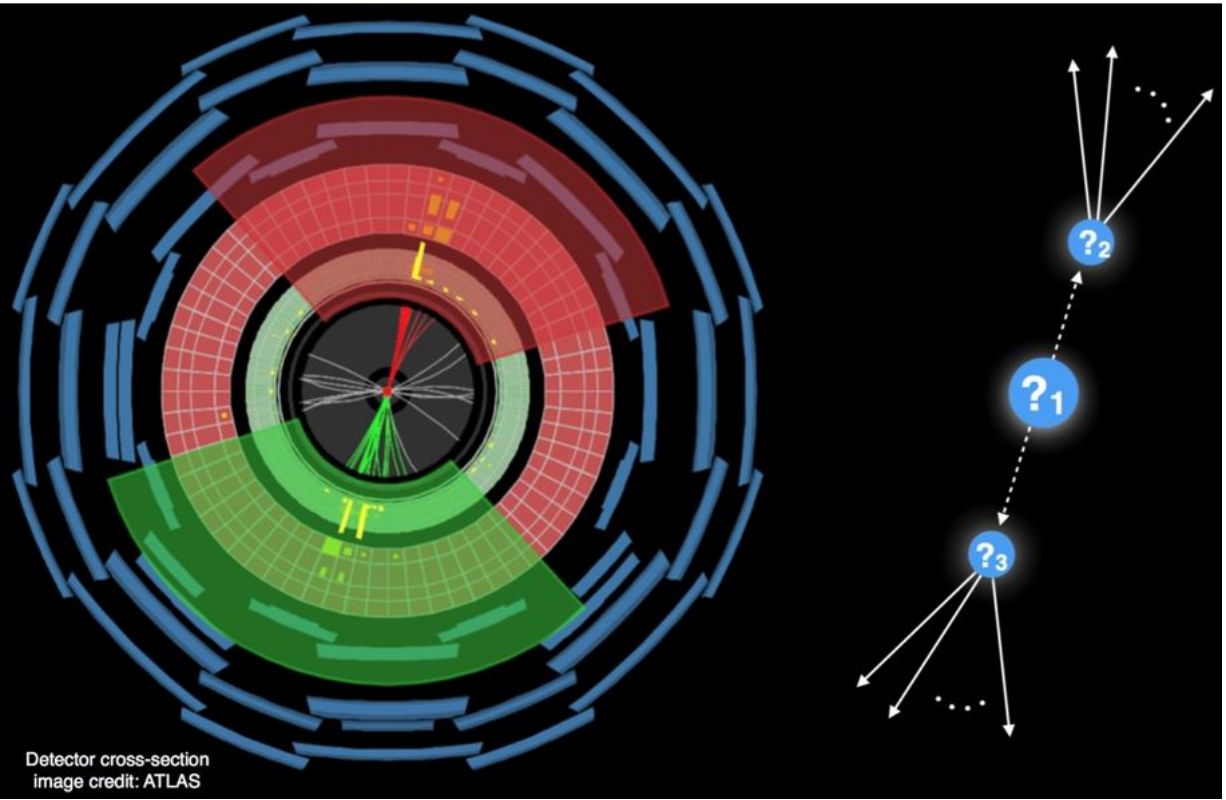


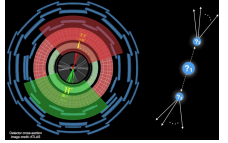
Bump-hunting using ML:

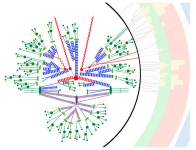
- **Generative Model**
- **Classifier**

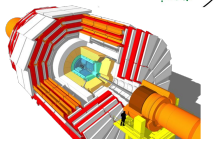


LHCO dataset





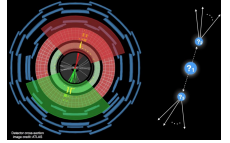
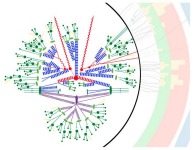
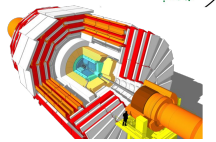


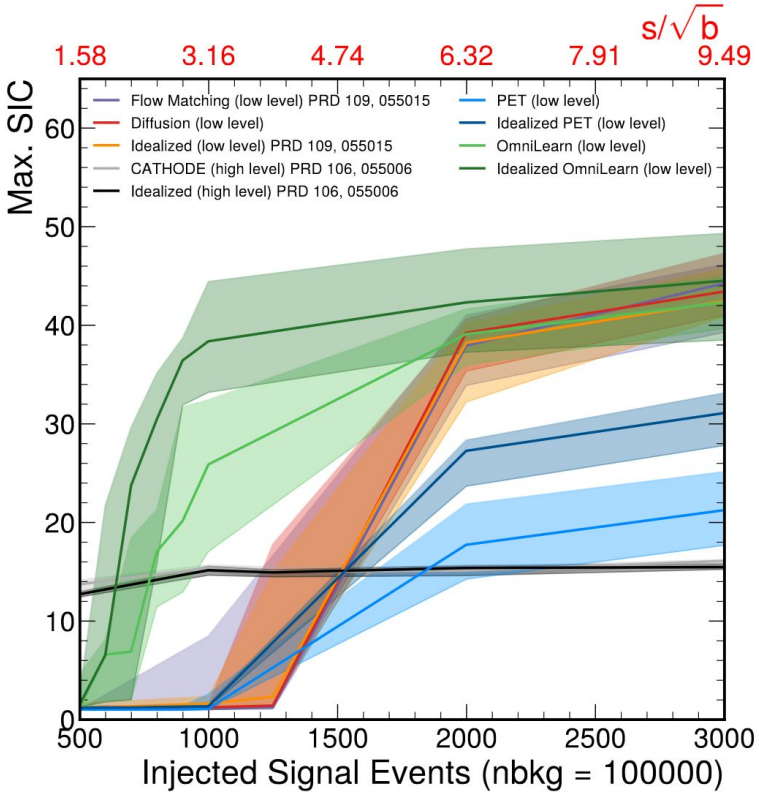


LHCO R&D dataset

- Resonant **dijet** final state: A->B(qq)C(qq) with m_A, m_B , m_C = 3.5, 0.5, 0.1 TeV

Anomaly Detection

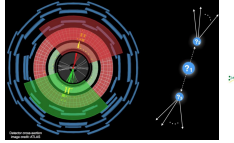
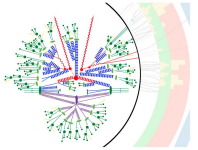
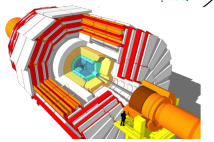




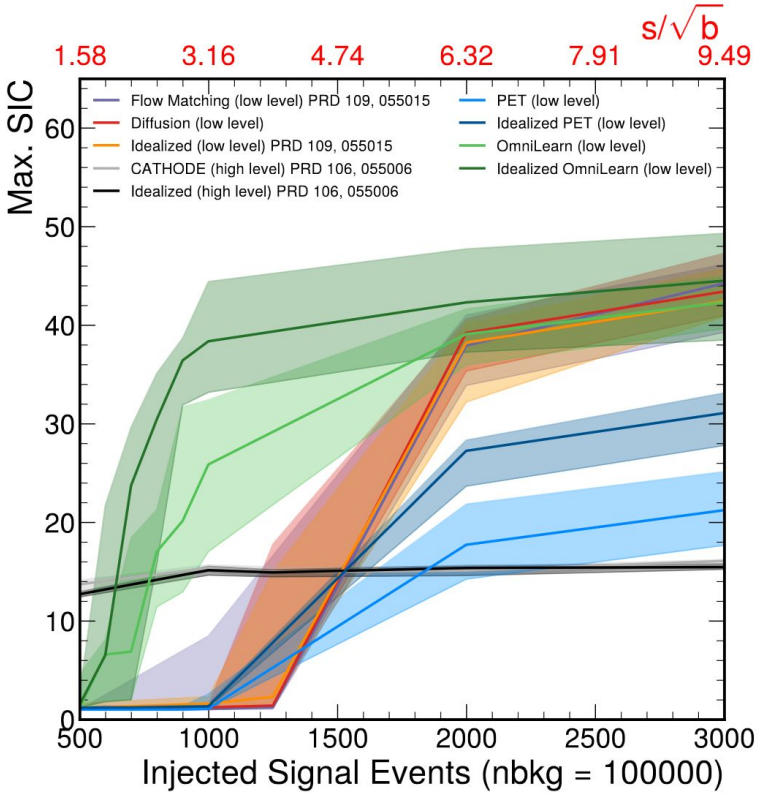


- **Generate** the full dijet system: 2*279*3 = **1674** numbers to generate
- **Classify** data from background

SIC = Significance Improvement Curve (TPR/sqrt(FPR) vs TPR) “By how much can I improve the significance of a particular signal given an initial significance.”

Anomaly Detection

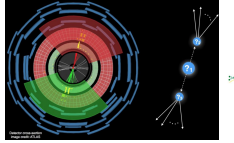


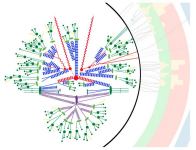
- **Generate** the full dijet system: 2*279*3 = **1674** numbers to generate
- **Classify** data from background

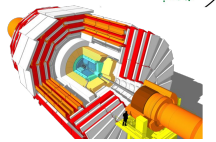
Previous results were limited by the amount of data in the SR: Only sensitive to NP when **S/B > 3% ~ 4σ**

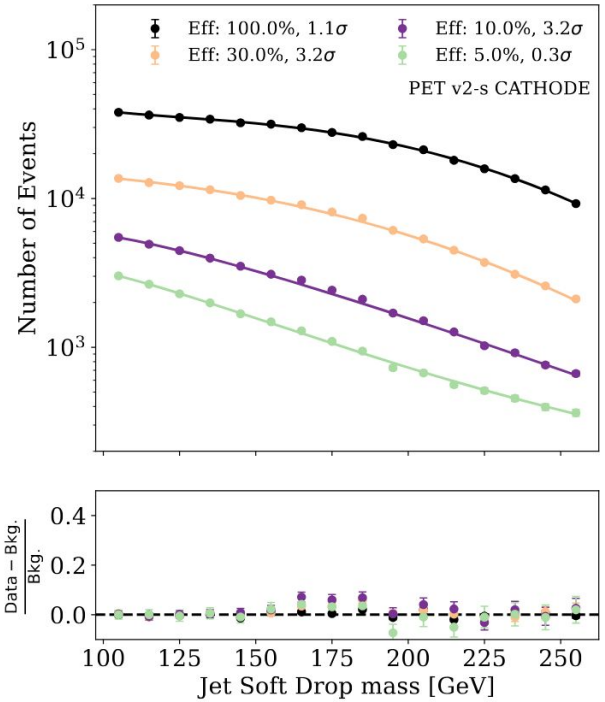
OmniLearn founds the NP with **S/B = 0.7% ~ 2σ**

Anomaly Detection





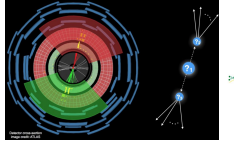
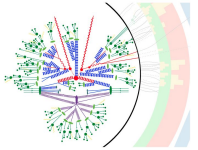
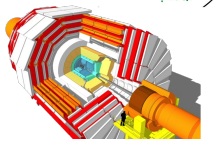


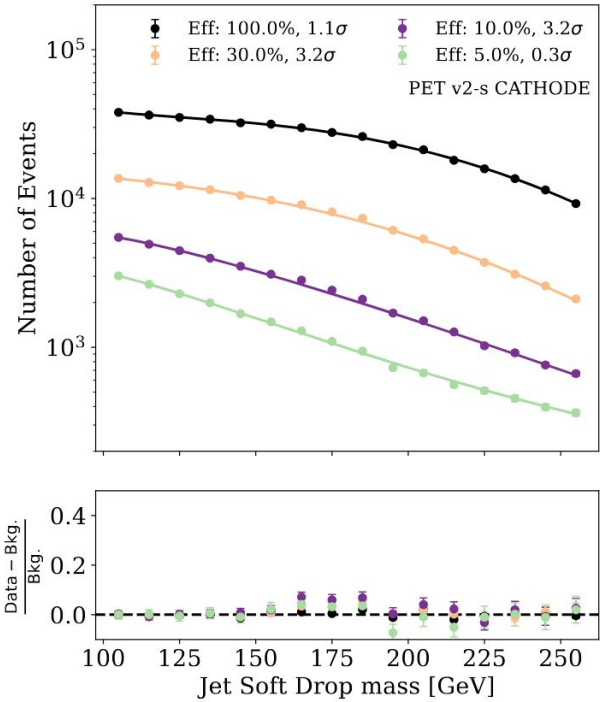


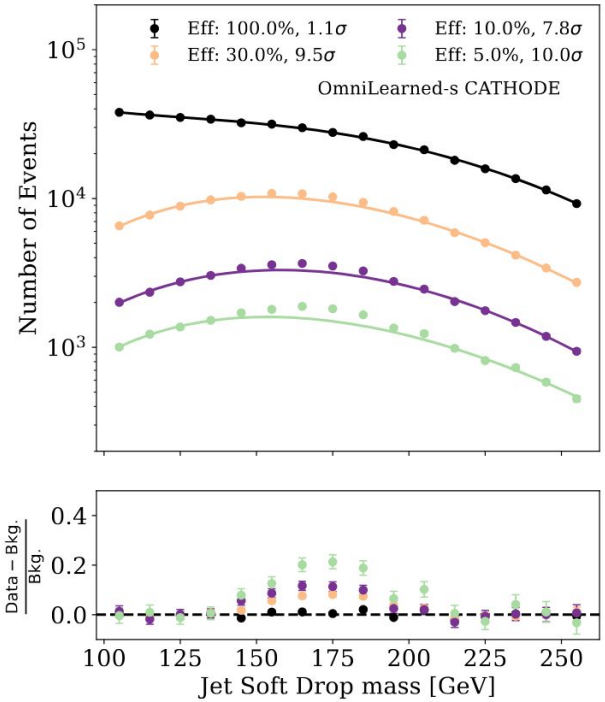
Use **experimental data** for this exercise

- **Signal**: top quarks, corresponding to 0.1% of the data
- Model trained from scratch **unable to find the anomaly**

Anomaly Detection

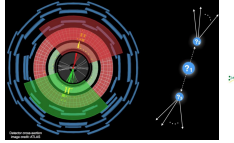
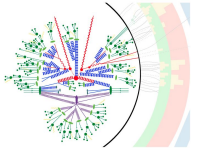
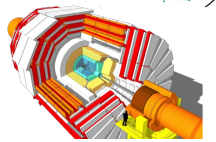


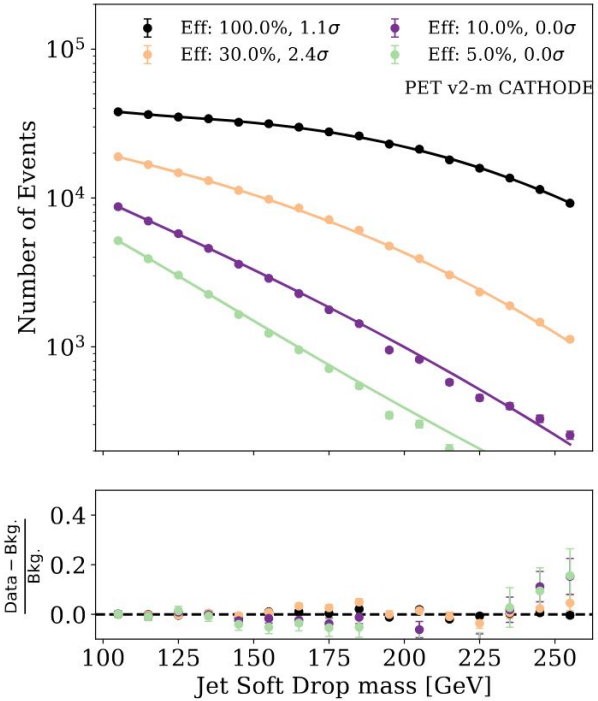


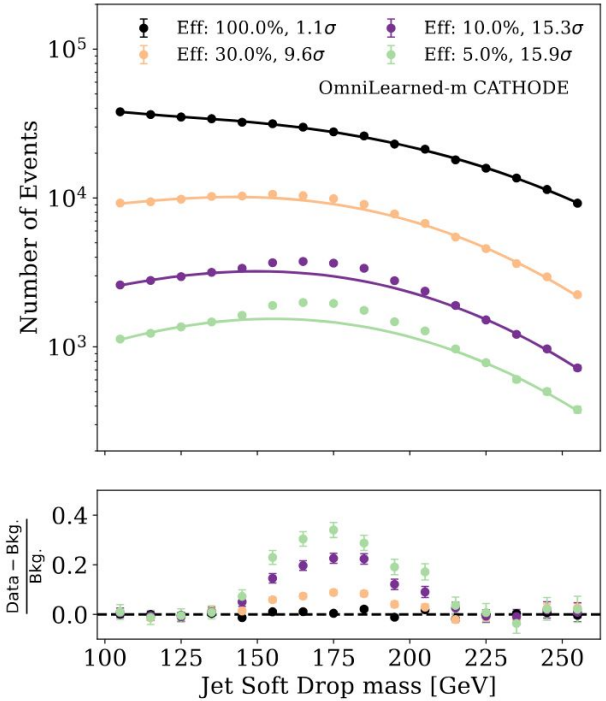
Use **experimental data** for this exercise

- **Signal**: top quarks, corresponding to 0.1% of the data
- **OmniLearned** finds the signal!

Anomaly Detection



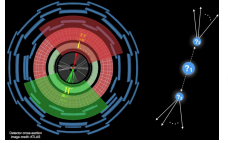


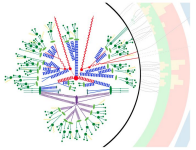
Use **experimental data** for this exercise

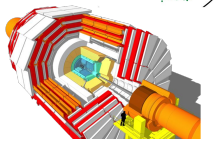
- **Signal**: top quarks, corresponding to 0.1% of the data
- **OmniLearned** finds the signal!
- **Bigger** model even more sensitive

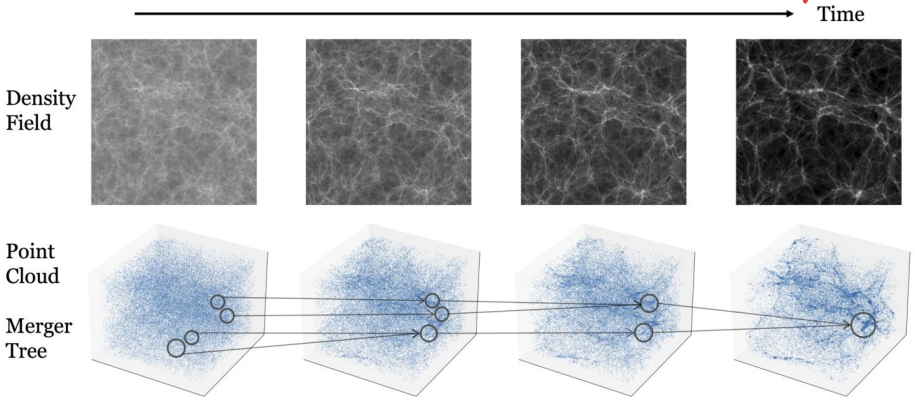
Bonus: OmniCosmos

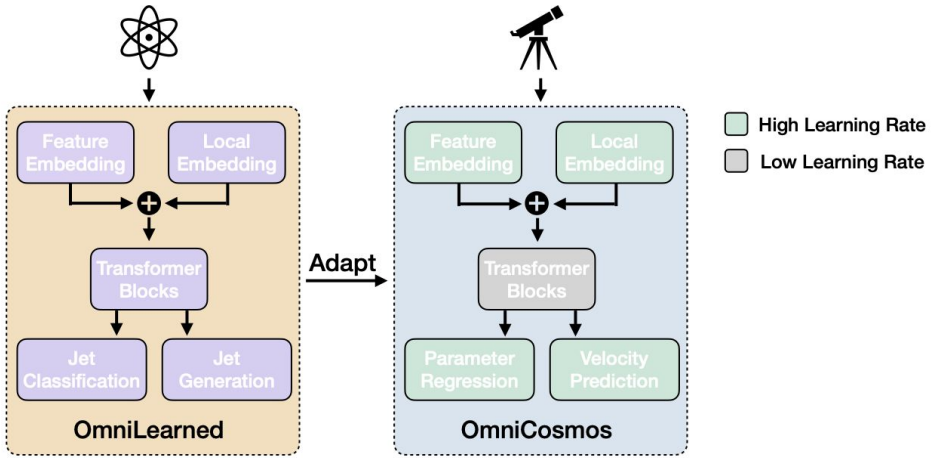
NEW









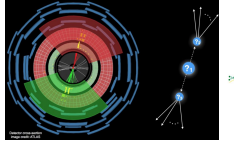


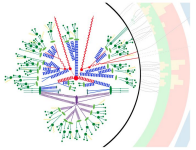
- Can we go **beyond Collider Physics**?
- Expensive to generate multiple universes
- Adapt OmniLearned to predict cosmological parameters and dark matter halo velocities: **OmniCosmos**

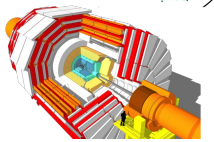
V. Mikuni, I. Elsharkawy, B. Nachman arXiv:2512.24422

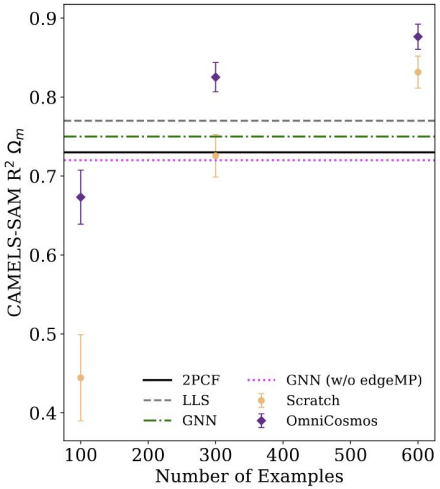
Bonus: OmniCosmos

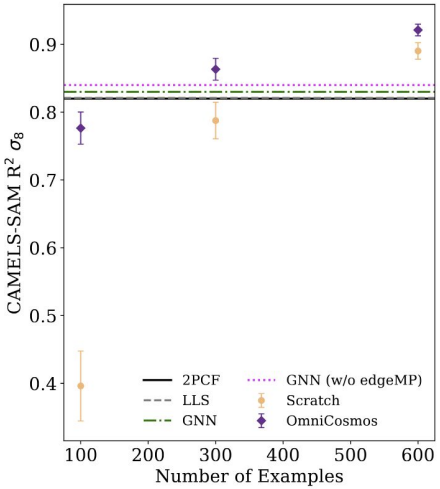
NEW

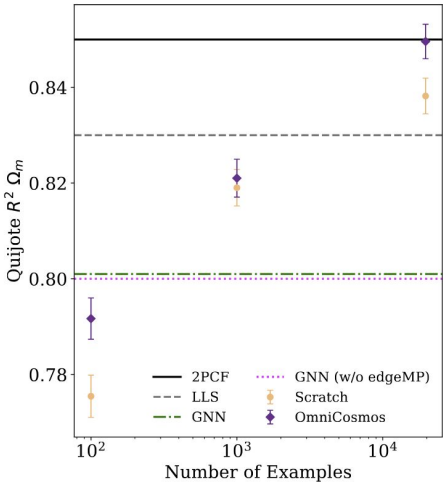


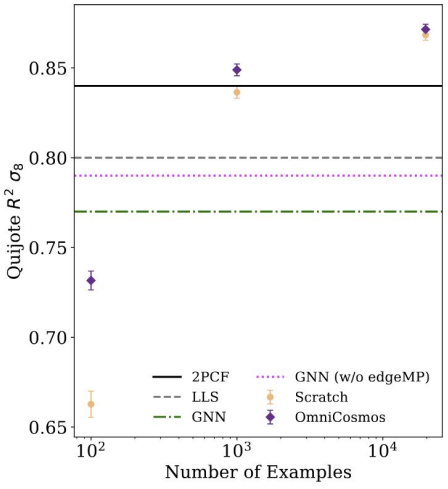










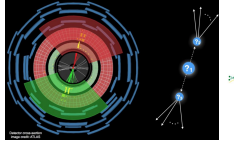


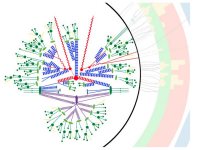
- Transfer learning benefits using **different types** of simulators
- Improve upon previous AI methods for cosmology

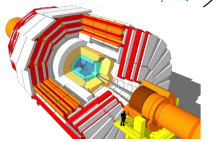
V. Mikuni, I. Elsharkawy, B. Nachman arXiv:2512.24422

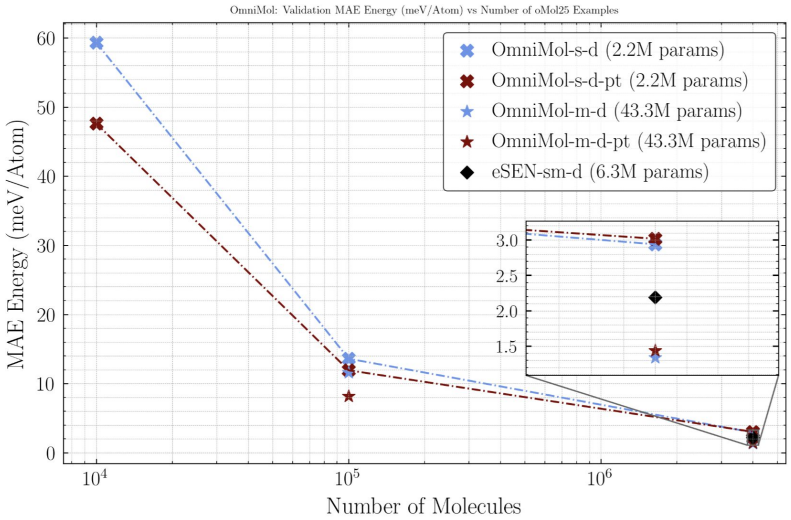
Bonus: OmniMol

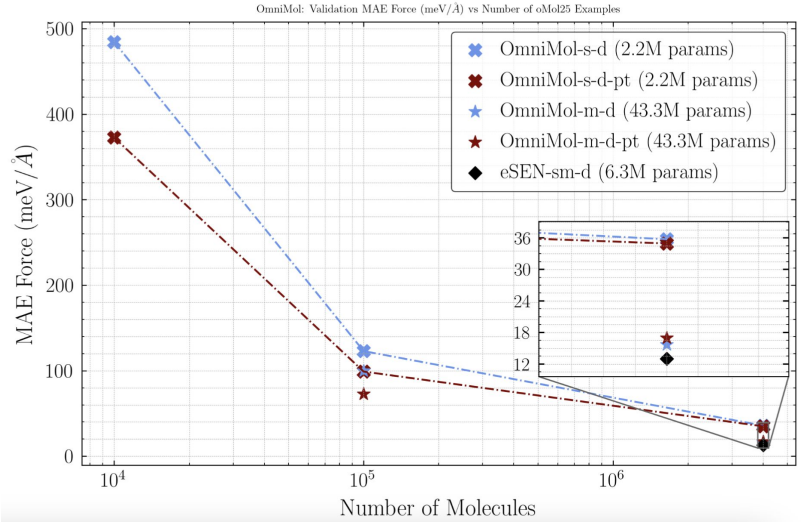
NEWER







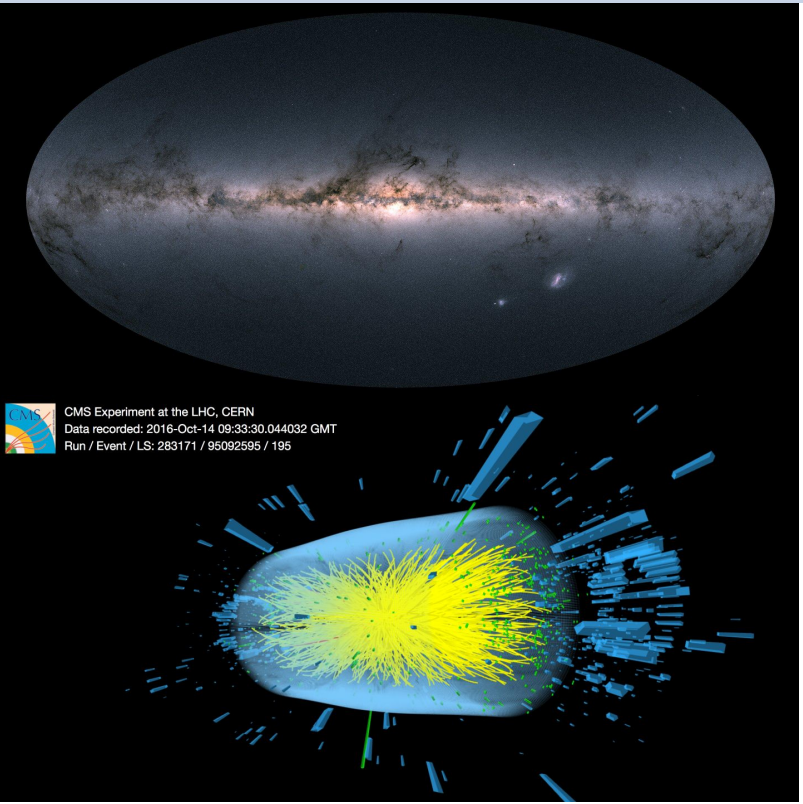




- Use OmniLearned to learn Molecular Dynamics: **OmniMol**
- Strong scaling with model and data size: Best performing transformer model in the **Open Molecules dataset**

V. Mikuni, I. Elsharkawy, B. Nachman, W. Bhimji arXiv:2601.10791

Conclusion



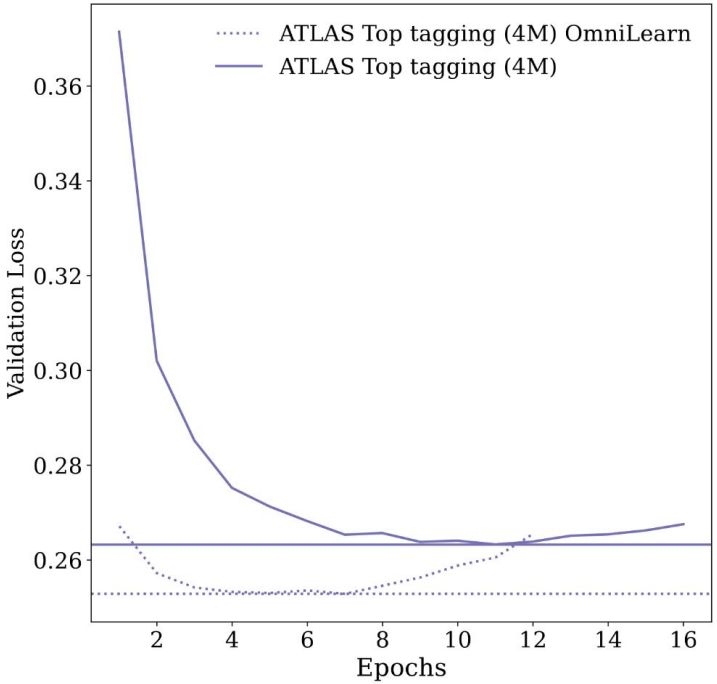
- **OmniLearn(ed)**: learn a rich representation of jets
- Improvements in **jet tagging**, **unfolding**, and **anomaly detection in a single model**
- **Going beyond**: Train models on large HEP data and adapt to data limited fields
- **Try yourself:** https://github.com/ViniciusMikuni/OmniLearned

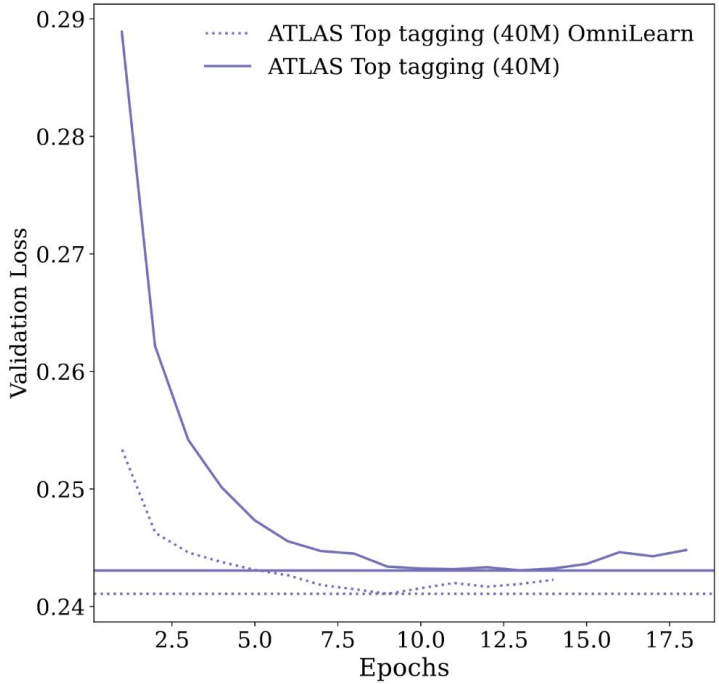
THANKS!

Any questions?

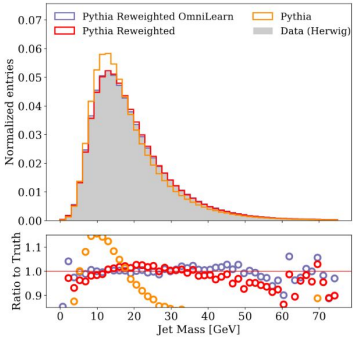
Backup

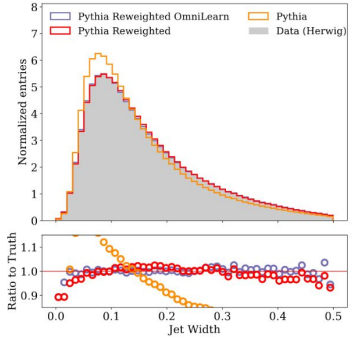
ATLAS Loss Curves

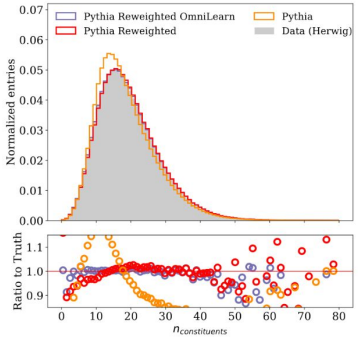


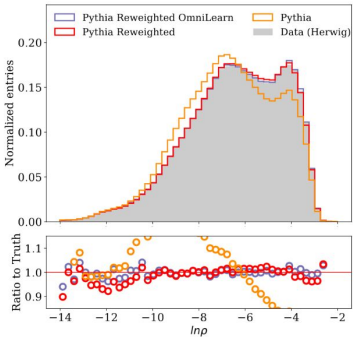


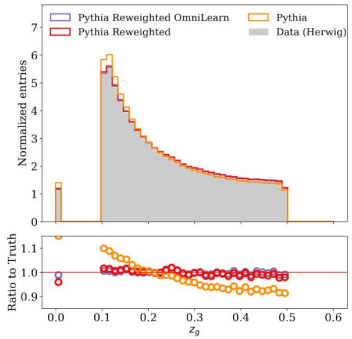
OmniLearn for reweighting

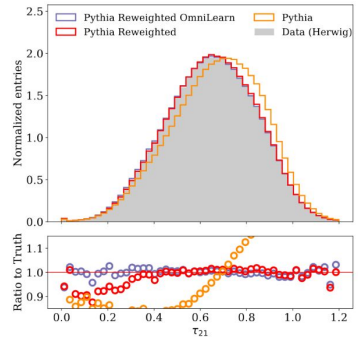


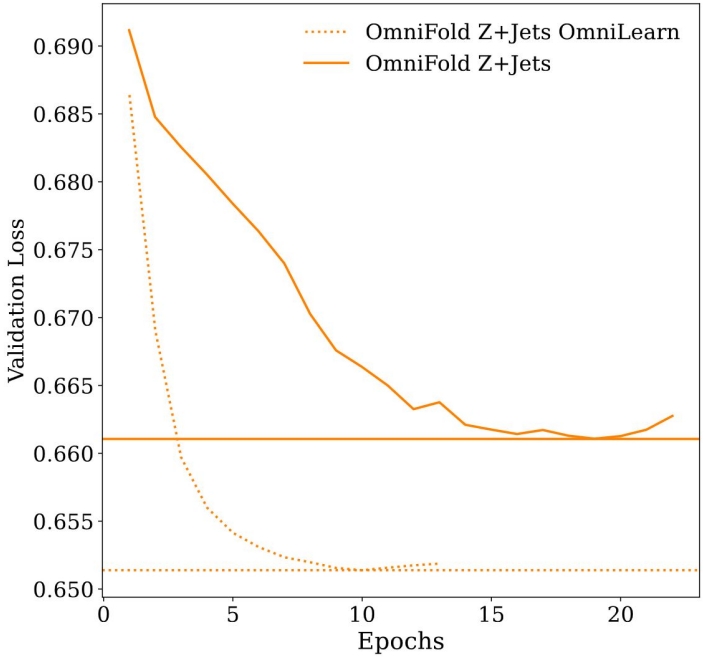




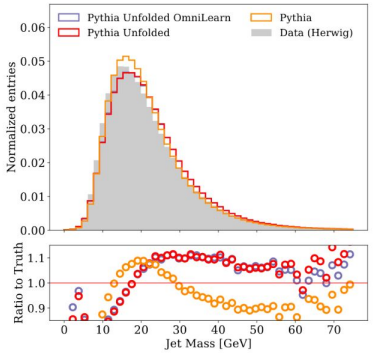


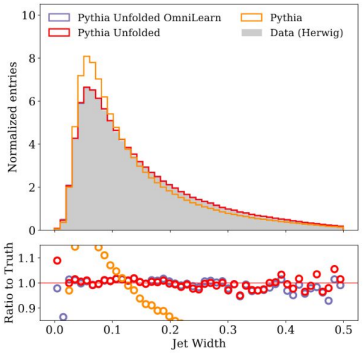


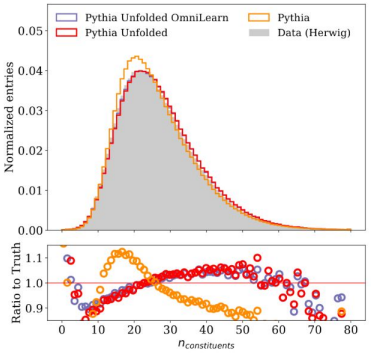


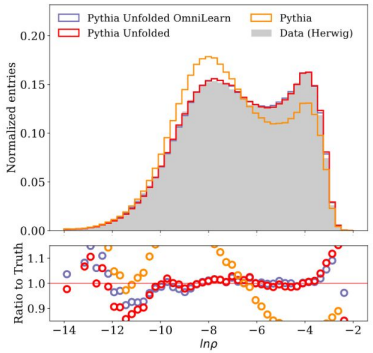


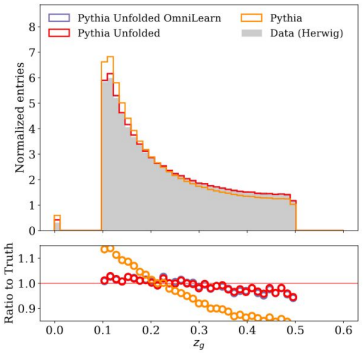
OmniLearn for Unfolding

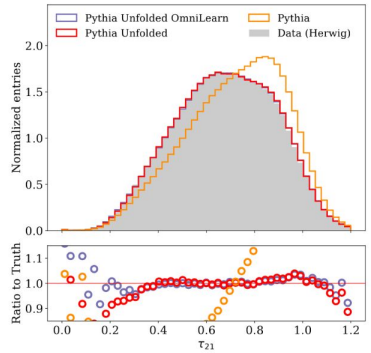




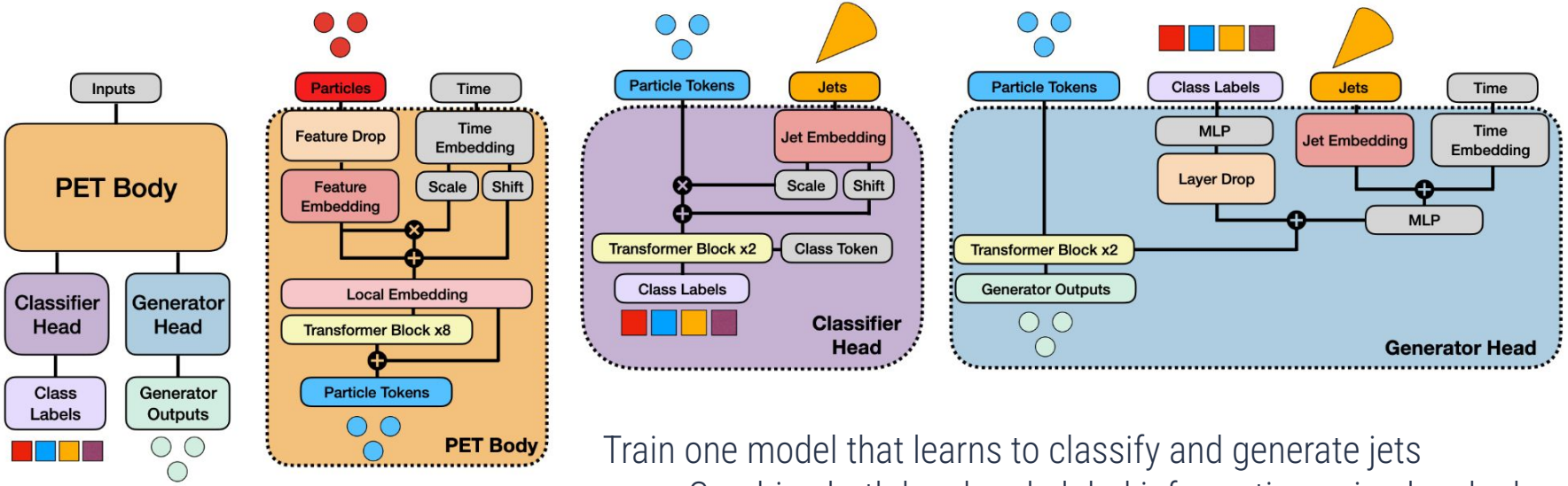








PET



Train one model that learns to classify and generate jets

- Combine both local and global information using local edges and a transformer: **P**oint-**E**dge **T**ransformer

More details at: https://arxiv.org/abs/2404.16091

Diffusion Generative Models

Forward SDE (data → noise)

$\mathbf{x}(0)$ — $\mathrm{d}\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\mathrm{d}t + g(t)\mathrm{d}\mathbf{w}$ → $\mathbf{x}(T)$

score function

$\mathbf{x}(0)$ ← $\mathrm{d}\mathbf{x} = \left[\mathbf{f}(\mathbf{x}, t) - g^2(t)\nabla_{\mathbf{x}} \log p_t(\mathbf{x})\right] \mathrm{d}t + g(t)\mathrm{d}\bar{\mathbf{w}}$ — $\mathbf{x}(T)$

Reverse SDE (noise → data)

Source:
https://yang-song.net/blog/2021/score/

Loss function

$$\mathcal{L} = \mathcal{L}_{\text{class}} + \mathcal{L}_{\text{gen}} + \mathcal{L}_{\text{class smear}}$$
$$= \text{CE}(y, y_{\text{pred}}) + \|\mathbf{v} - \mathbf{v}_{\text{pred}}\|^2 + \alpha^2 \text{CE}(y, \hat{y}_{\text{pred}})$$

Straightforward loss function:

- **Cross entropy** for each class
- Perturbed data prediction from the **diffusion loss**
- Classification over perturbed inputs: **data augmentation**!

More details at: https://arxiv.org/abs/2404.16091