BERKELEY LAB

NERSC

Foundational Models for AD


vmikuni@lbl.gov


vinicius-mikuni

Vinicius M. Mikuni

Foundational Models

BERKELEY LAB

what is the definition of a foundational model?

A **foundational model** is a large-scale machine learning model trained on massive, diverse datasets, typically using self-supervised learning, to serve as a general-purpose foundation for various downstream tasks. These models are designed to be adaptable and can be fine-tuned or prompted to perform specific tasks with minimal additional training.

Key Characteristics:

1. **Pretrained at Scale** – Trained on vast amounts of data to capture broad patterns.
2. **General-Purpose** – Can be adapted for various applications (e.g., text, images, code).
3. **Self-Supervised Learning** – Often trained without explicit human labels.
4. **Transfer Learning** – Can be fine-tuned for specific tasks with smaller datasets.
5. **Emergent Capabilities** – Often exhibit unexpected abilities beyond their initial training data.

Foundational Models

BERKELEY LAB

what is the definition of a foundational model?

A
ty
d
to

K

How a **Foundational Model** for Science would look like?

2. **General-Purpose** – Can be adapted for various applications (e.g., text, images, code).
3. **Self-Supervised Learning** – Often trained without explicit human labels.
4. **Transfer Learning** – Can be fine-tuned for specific tasks with smaller datasets.
5. **Emergent Capabilities** – Often exhibit unexpected abilities beyond their initial training data.

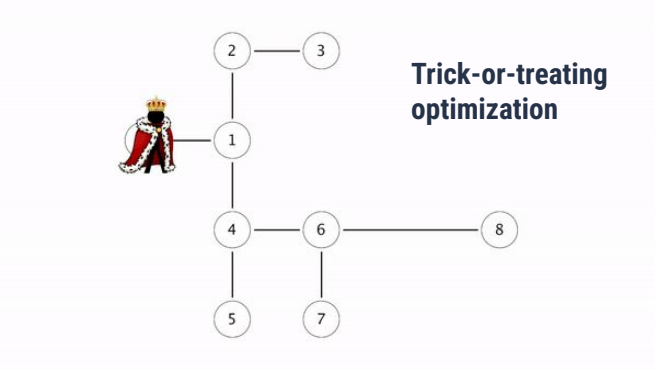
BERKELEY LAB

How to teach AI about Particle Collisions?

Data

Model

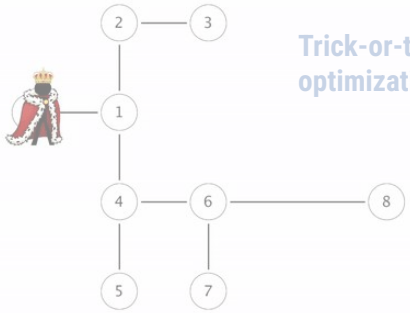
Learning



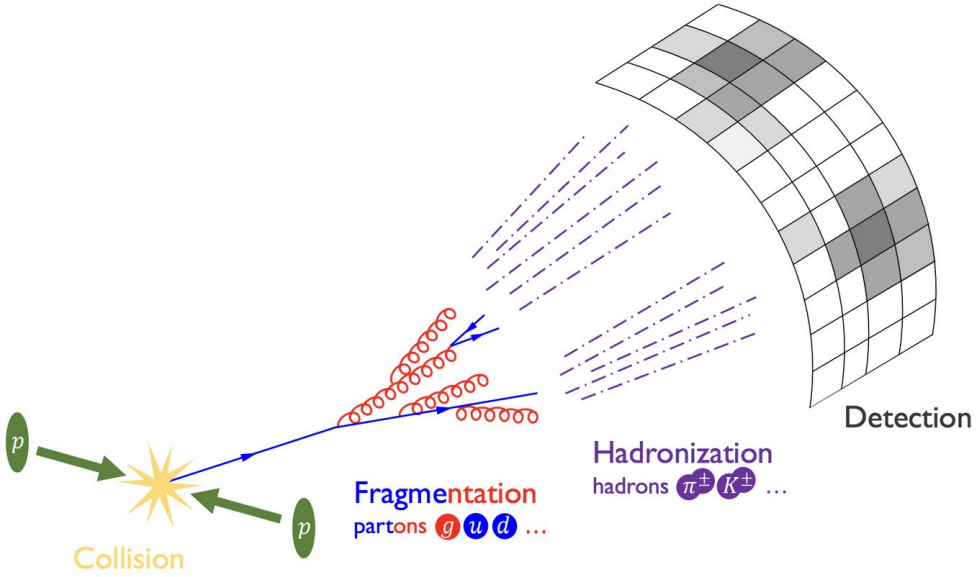
Data

Model

Learning



Jets



Jets are the most common signatures at the LHC

- Complicated signature: O(10-100) particles are clustered in each jet
- Everywhere: Jets are used in almost any analysis at the LHC

Training data

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**

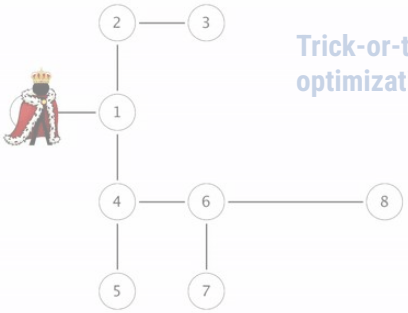
Use the pre-trained model as the starting point and fine-tune using different datasets

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

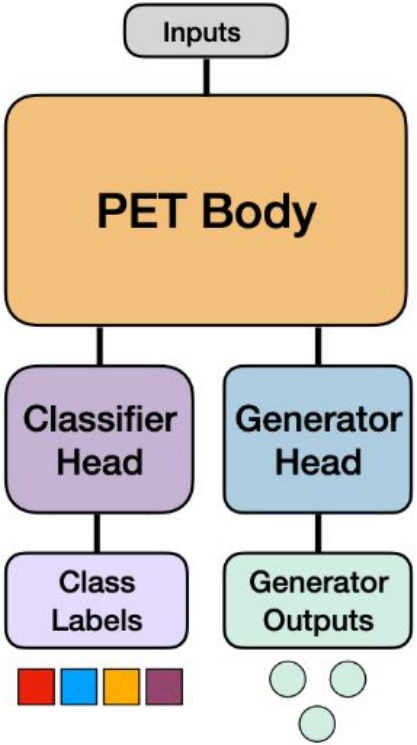
Data

Model

Learning

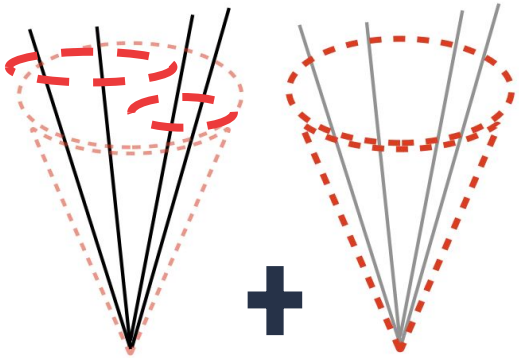


Encoding jet information



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

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



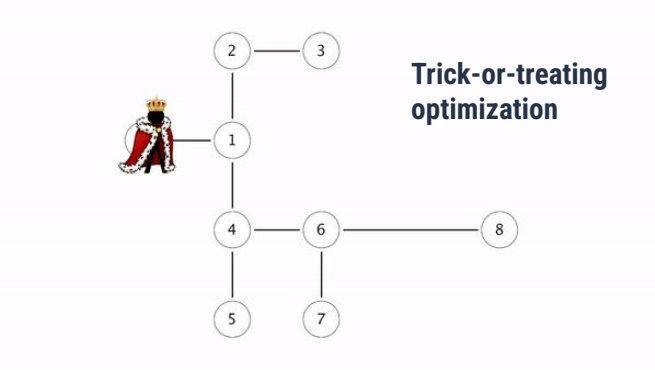
See also:

- 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.

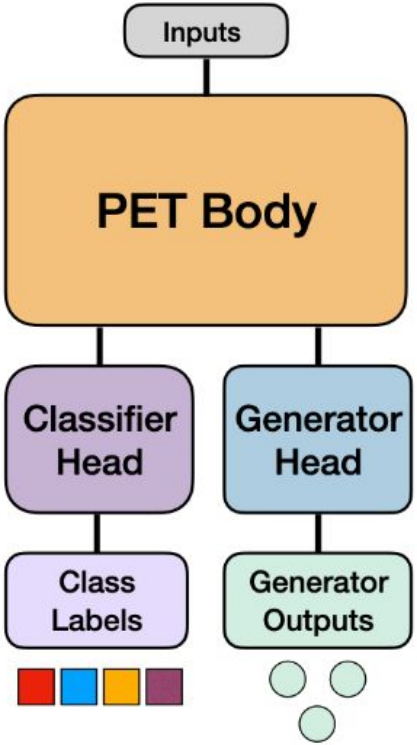
Data

Model

Learning



Encoding jet information



Couple the network with multiple experts:

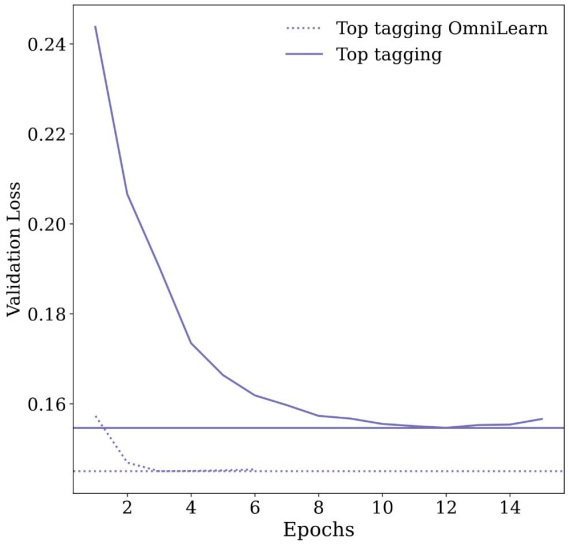
- **Classify jets**: learns the difference in radiation between jet types
- **Generate jets**: implicitly learn the likelihood of jets for different particles
- **Multi-objective** loss

Evaluation

BERKELEY LAB

2 different jet categories, AK8 jets simulated in pp collisions with ~~Madgraph +~~ Pythia8 with ATLAS Delphes detector simulation

	Acc	AUC	$1/\epsilon_B$	
			$\epsilon_S = 0.5$	$\epsilon_S = 0.3$
ResNeXt-50 [38]	0.936	0.9837	302 ± 5	1147 ± 58
P-CNN [38]	0.930	0.9803	201 ± 4	759 ± 24
PFN [35]	-	0.9819	247 ± 3	888 ± 17
ParticleNet [38]	0.940	0.9858	397 ± 7	1615 ± 93
JEDI-net [37]	0.9300	0.9807	-	774.6
PCT [41]	0.940	0.9855	392 ± 11	1559 ± 98
LGN [79]	0.929	0.964	-	435 ± 95
rPCN [39]	-	0.9845	364 ± 9	1642 ± 93
LorentzNet [10]	0.942	0.9868	498 ± 18	2195 ± 173
PELICAN [80]	0.9425	0.9869	-	2289 ± 204
ParT [42]	0.940	0.9858	413 ± 16	1602 ± 81
ParT-f.t. [42]	**0.944**	**0.9877**	**691 ± 15**	**2766 ± 130**
Mixer(HDBSCAN) [81]	-	0.9859	416	-
PET Classifier	0.938	0.9848	340 ± 12	1318 ± 39
OmniLearn	0.942	0.9872	568 ± 9	**2647 ± 192**



FastSim to FullSim

OmniLearn is trained on cheap Delphes simulations but the data structure learned is transferable to realistic datasets!

- 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**

DOI:10.7483/OPENDATA.ATLAS.FG5F.96GA

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

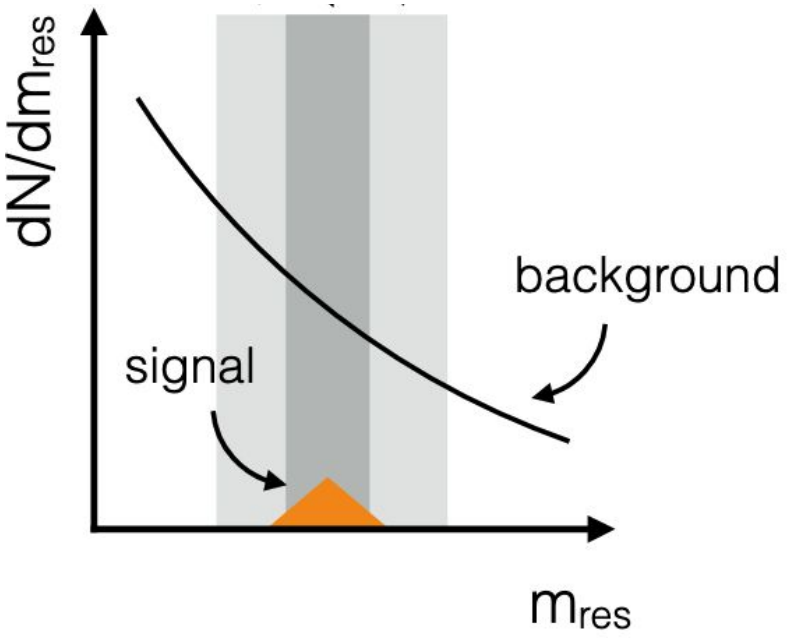
Anomaly Detection

BERKELEY LAB

Evaluation datasets: **9**

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

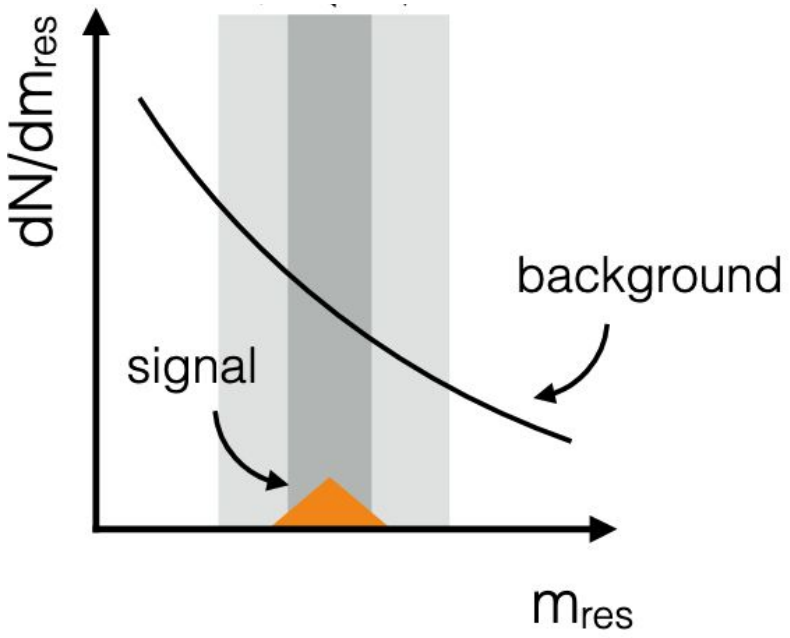


BERKELEY LAB

Evaluation datasets: **9**

Bump-hunting using ML:

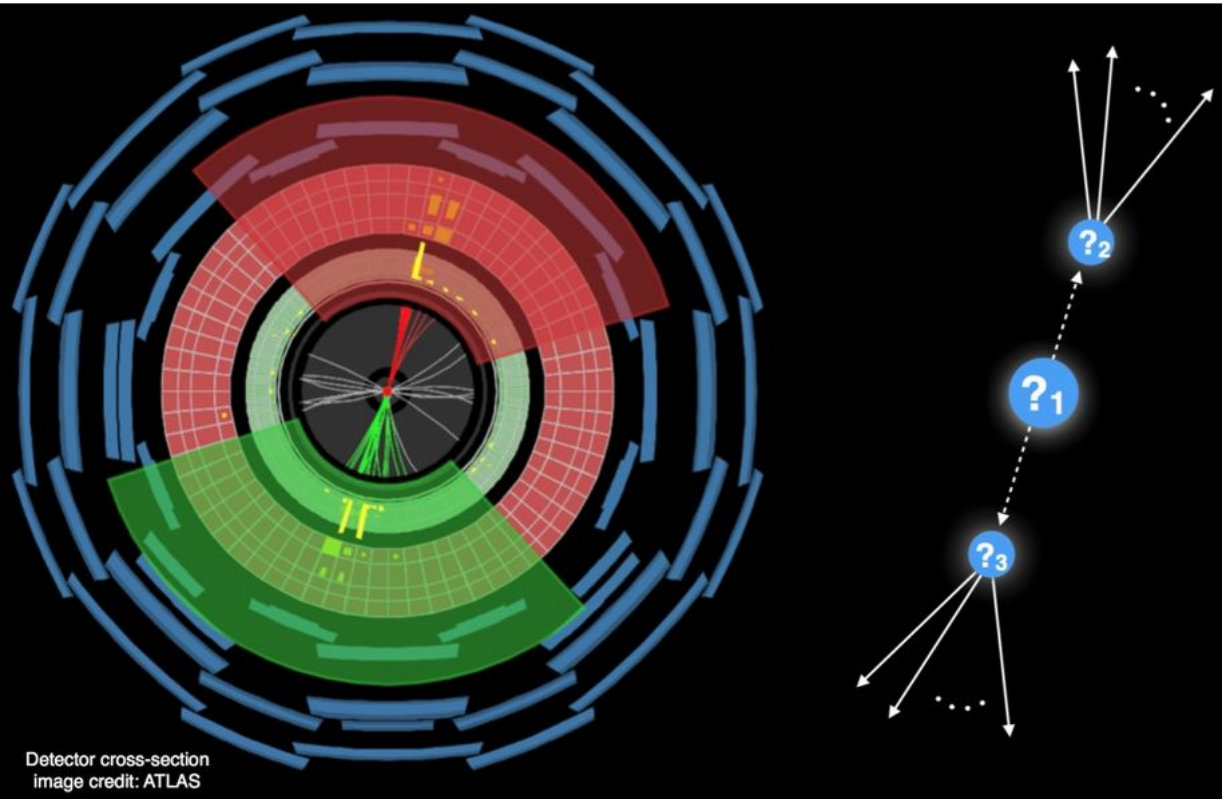
- **Generative Model**
- **Classifier**



LHCO dataset

BERKELEY LAB

Evaluation datasets: **9**

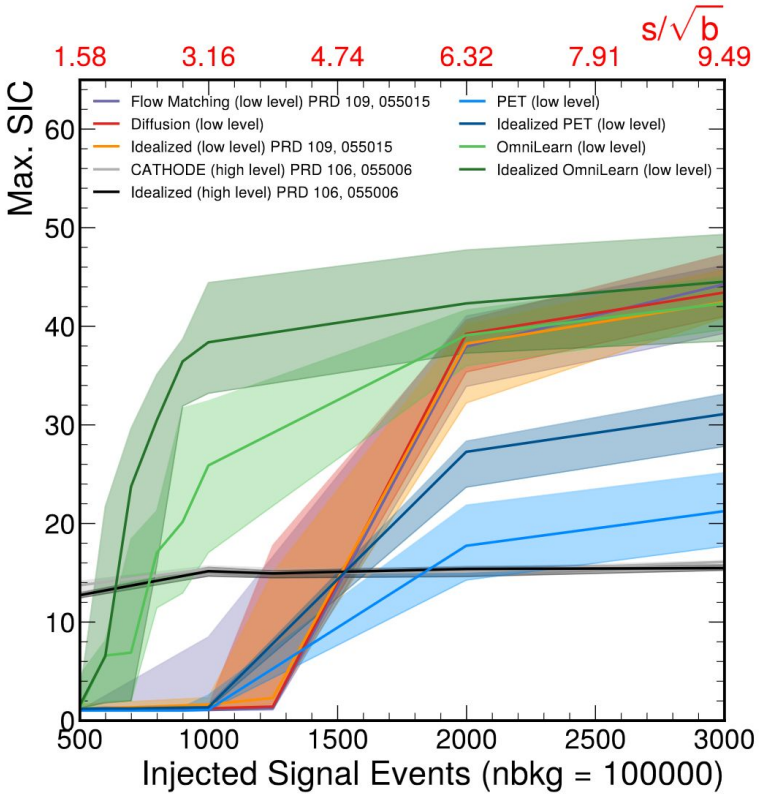


LHCO R&D dataset

- 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

Evaluation datasets: **9**

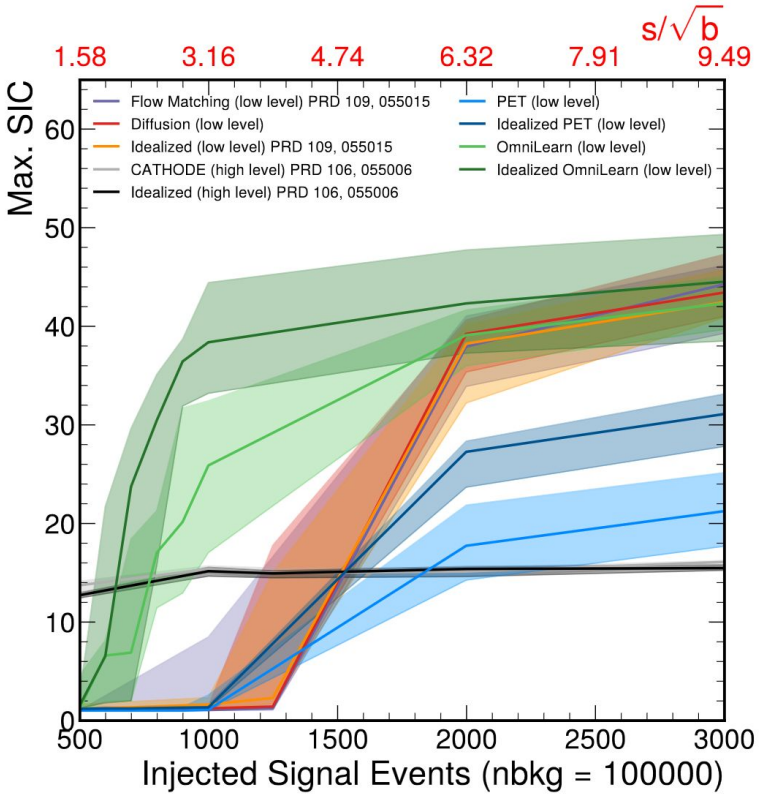


- **Generate** the full dijet system: 2*279*3 = **1674** numbers to generate
- **Classify** data from background

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

Evaluation datasets: **9**



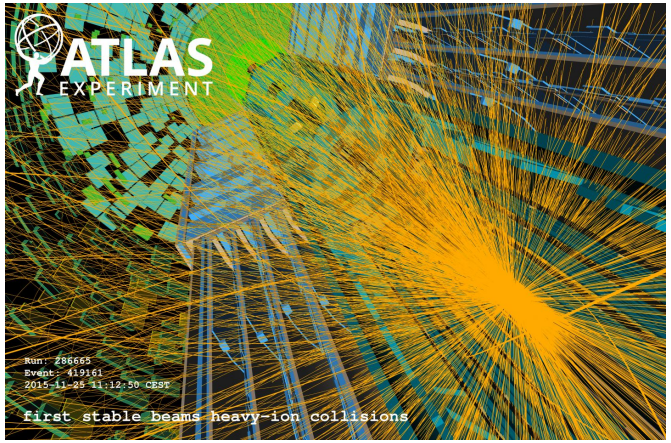
- **Generate** the full dijet system: 2*279*3 = **1674** numbers to generate
- **Classify** data from background

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

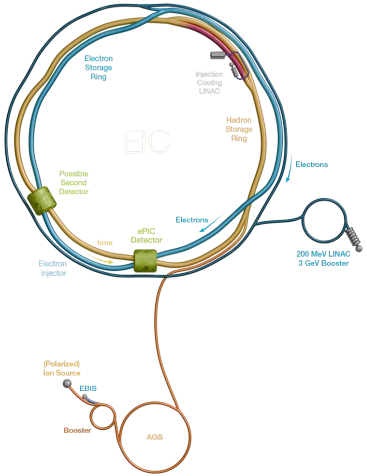
OmniLearn founds the NP with **S/B = 0.7% ~ 2σ**

Future

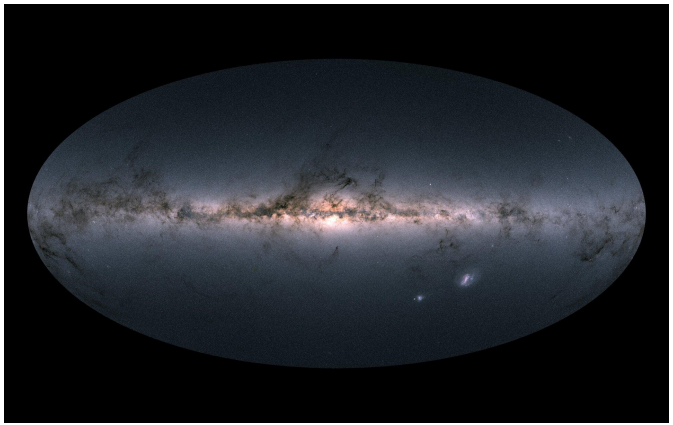
Encode full LHC Events



Encode other colliders



Encode the Universe



Conclusion?

- **Foundational Models for data**: learn a useful representation of the data
- Representation can be adapted to specific datasets through fine-tuning
- Smaller datasets are needed for the adaptation: **smaller datasets also benefit from large transformer models**
- **Try it out yourself**: https://github.com/ViniciusMikuni/OmniLearn/ and check out the paper: arXiv:2404.16091

Conclusion

- Interested in **developing models to understand the universe**? Apply for a Postdoc Position at the Kobayashi-Maskawa Institute (KMI) in Japan!
- https://academicjobsonline.org/ajo/jobs/30011
- Feel free to talk to me about it!

THANKS!

Any questions?

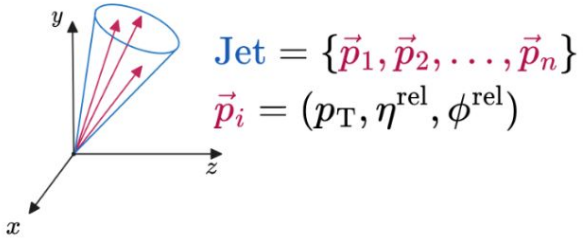
Backup

BERKELEY LAB

Option 1: Tokenization

- Inspired by LLMs, tokenize the information of input particles

Jet constituents with **continuous features**



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

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.

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

BERKELEY LAB

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

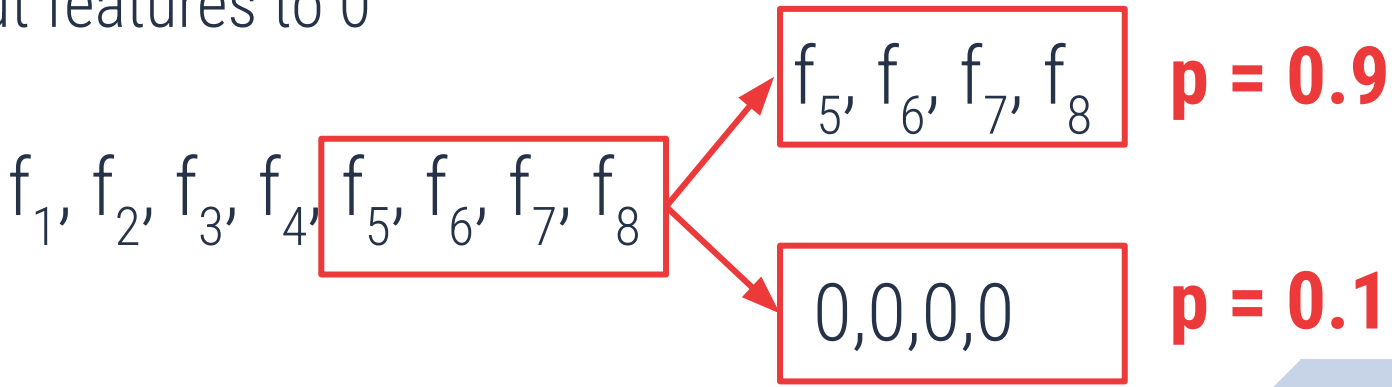
See:

- 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.

Input Dropout

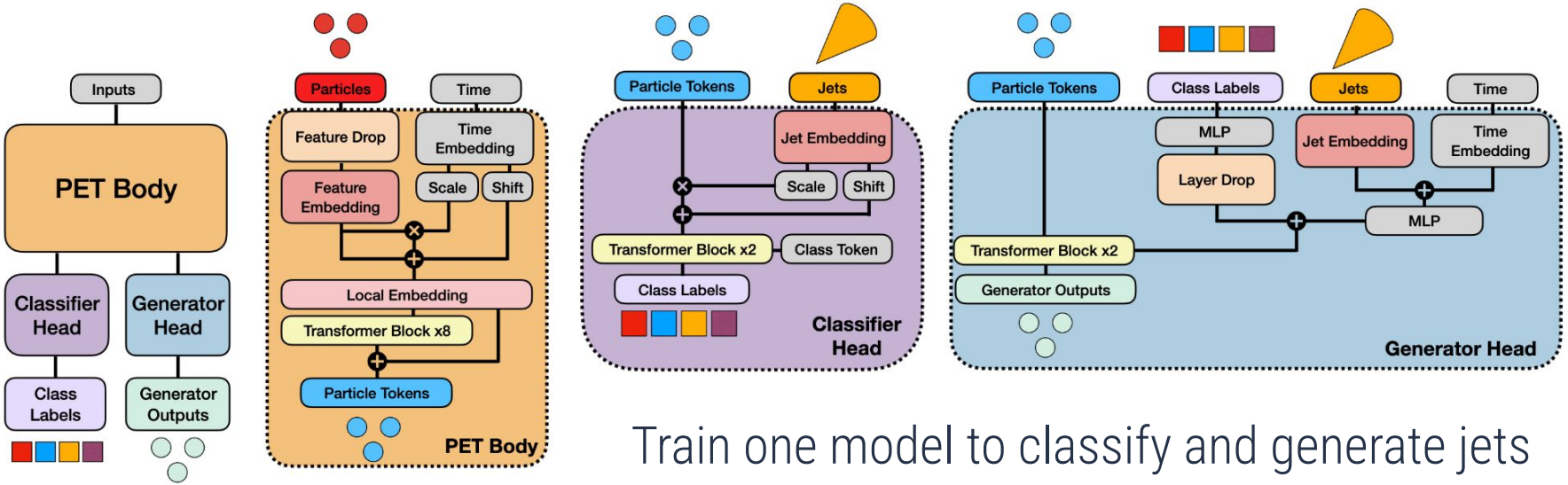
Not all datasets contain the same information:

- Let the model learn with and without some features
- **Feature Dropout**: With fixed probability, set some of the input features to 0



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

OmniLearn



Train one model to classify and generate jets

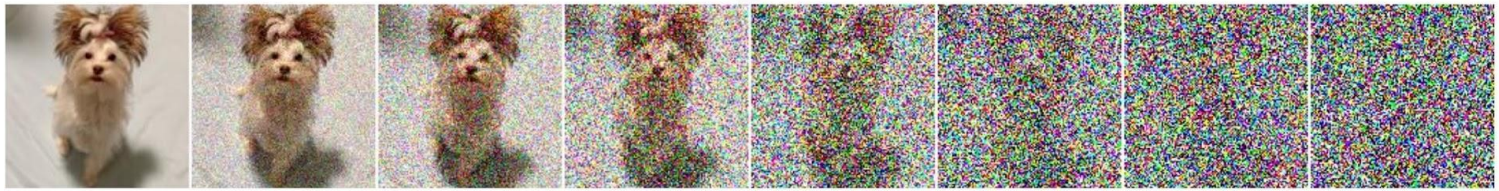
- Combine both local and global information: **OmniLearn**

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