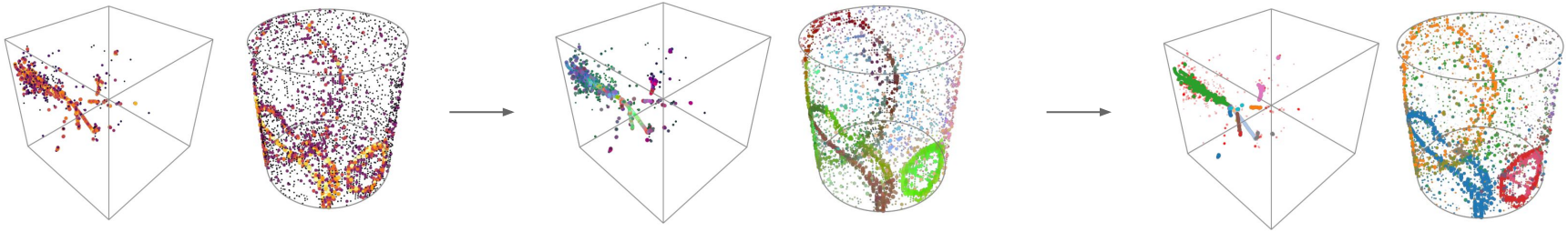
Toward Cross-experiment Pretraining of Point Cloud Foundation Models for Neutrino Detectors



Samuel Young (Stanford)
Kazuhiro Terao (SLAC)

NPML 2026 // based on arXiv:2512.01324

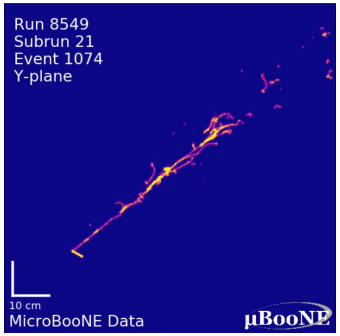
SLAC NATIONAL ACCELERATOR LABORATORY

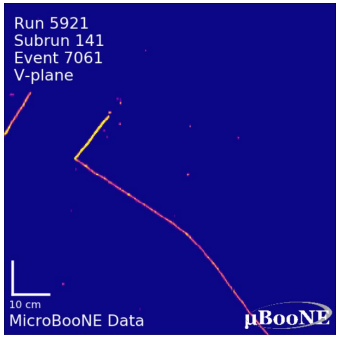
Outline

1. Foundation models for sensory data: what & why
2. Panda SSL in LArTPC
3. Panda SSL in Water Cherenkov
4. Lessons learned and challenges for the future

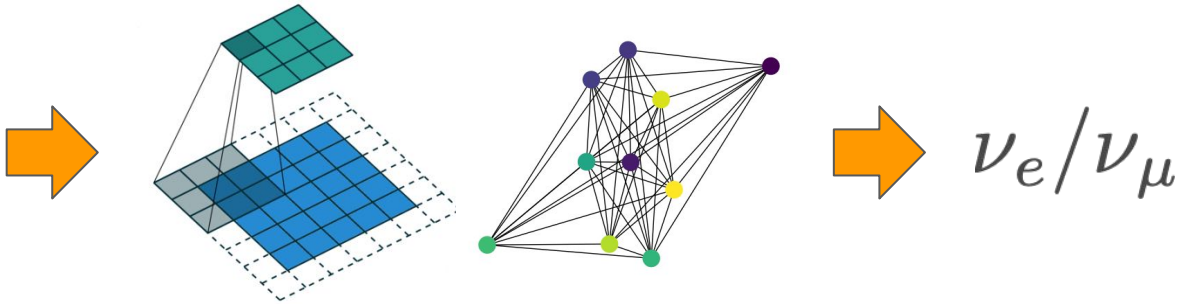
Deep learning in neutrino physics

Common approaches: deep neural network(s) trained using simulated datasets with "labels" via "supervised learning"





Deep Neural Nets



ν_e/ν_μ

"100 analyses, 100 AI models" (task-specific)

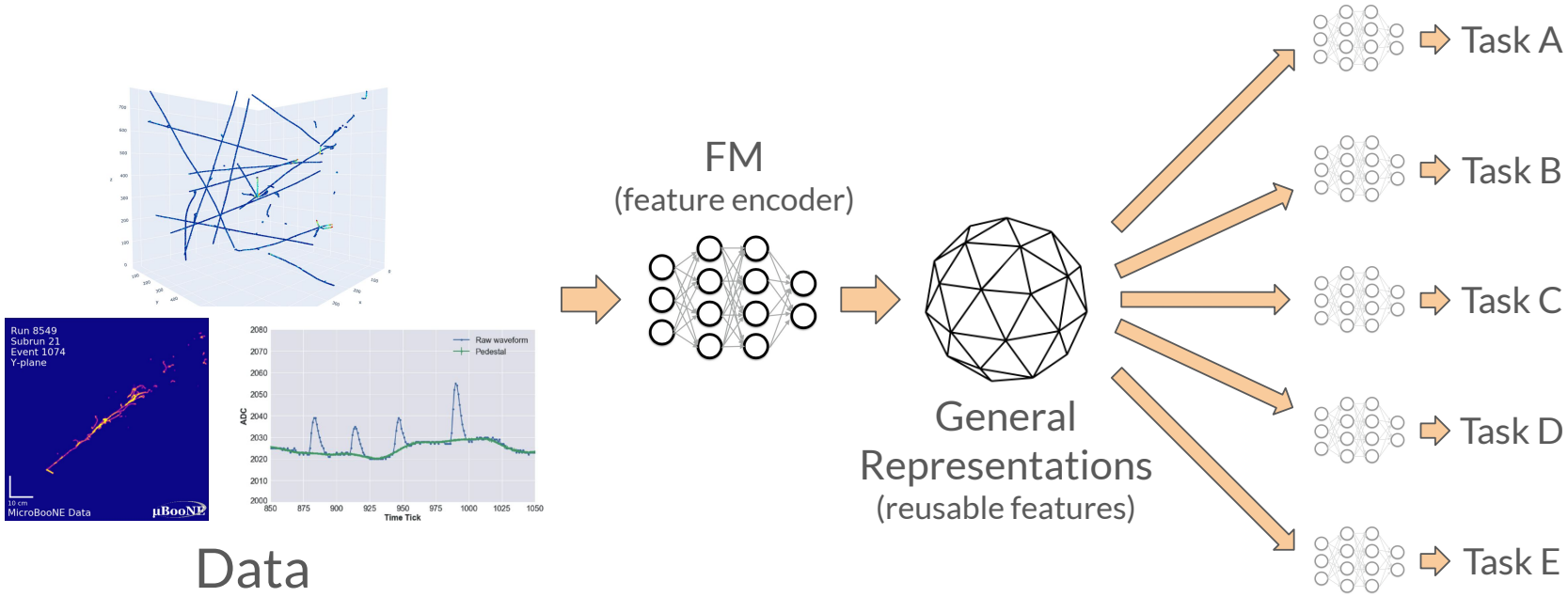
Poor reusability, cost for R&D/maintenance

Vulnerable against data-shift

High resolution + sensor-level ➔ hard to model.

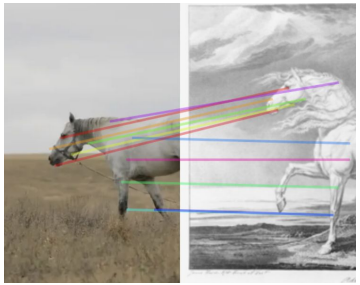
Foundation models = 2 training phases

1. **Pre-training** (Representation Learning) – computationally expensive
2. **Fine-tuning** (Adaptation) – computationally inexpensive (relatively speaking…)



Foundation models are generalists

Monocular depth estimation [1]

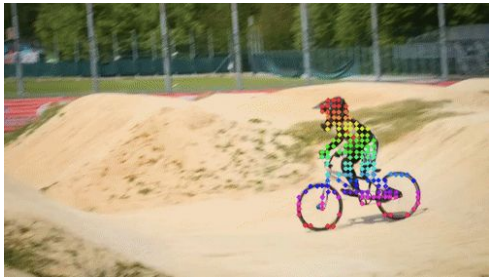


Point Correspondence [2]

self-supervised vision models are foundation models

DINO (SSL) [2]

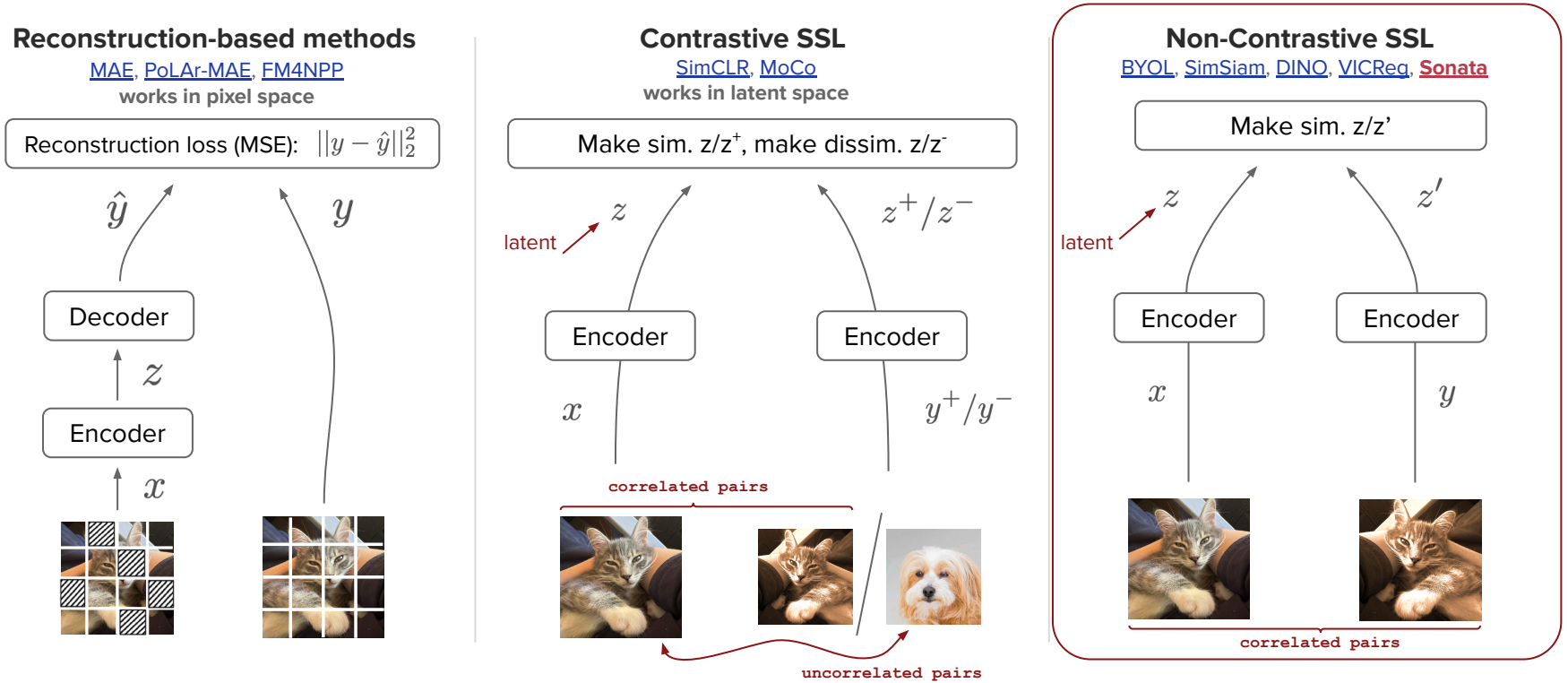
Segmentation [3]



Video Tracking [4]

Techniques in self-supervised learning

learning data representations without label supervision by comparing altered/partial views of the same input.



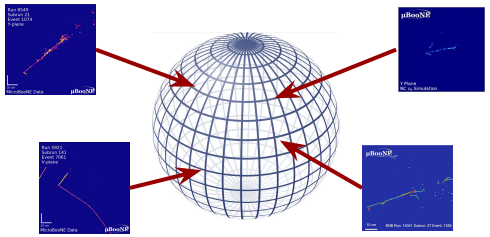
Panda: Self-distillation and hierarchy

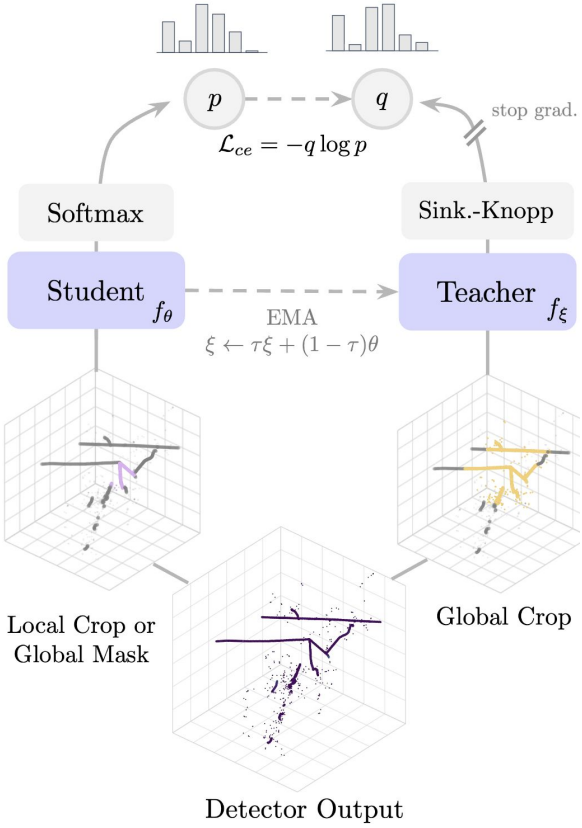
instead of **reconstructing masked portions of image directly**,

let's **predict where they would would end up on a unit sphere** (i.e., classify them),

by enforcing **consistency between global and local views** of the same image under **strong augmentations**.

this is **non-contrastive SSL.**



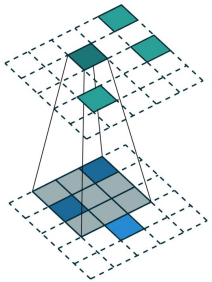


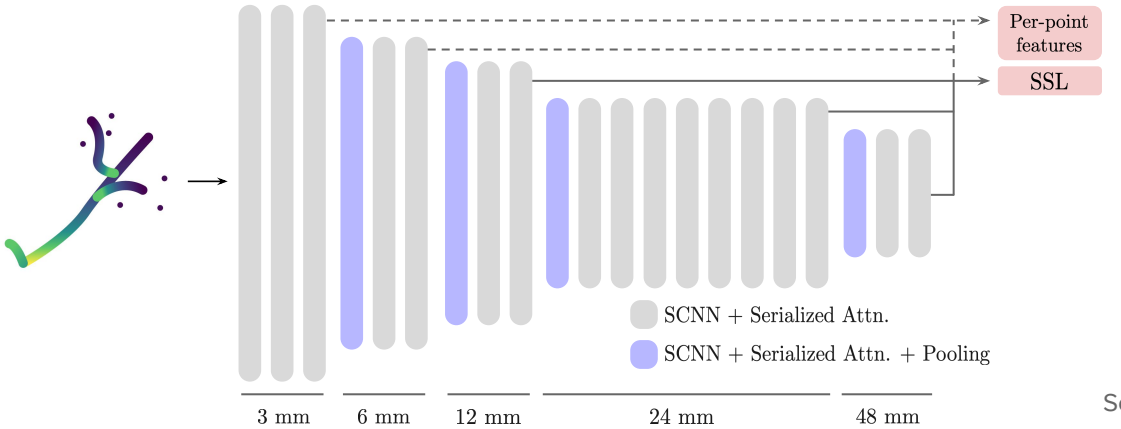
Panda: Self-distillation and hierarchy

Point Transformer V3 (CVPR ‘24)

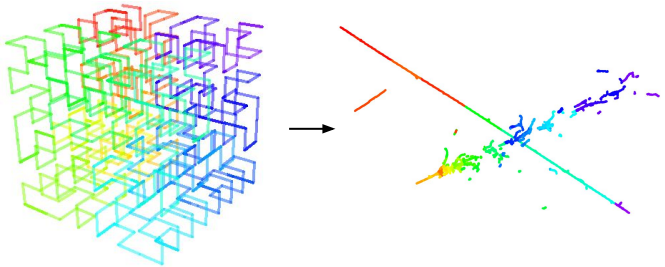
- Efficient serialization for approximate nearest neighbor (no kNN!)
- Local patch hierarchical attention
  - Pooling 5x, 48 to 512 features, patch size 256; 90M params
- Result: more expressive than Sparse UResNet, still scalable (compute scales linearly wrt # points)
- Dataset: PILArNet-M = 1M 3D point clouds (10k~100k/image)

SCNN: Sparse CNN





Approximate nearest neighbor via locality-sensitive hashing (Hilbert curve, Z-curve)



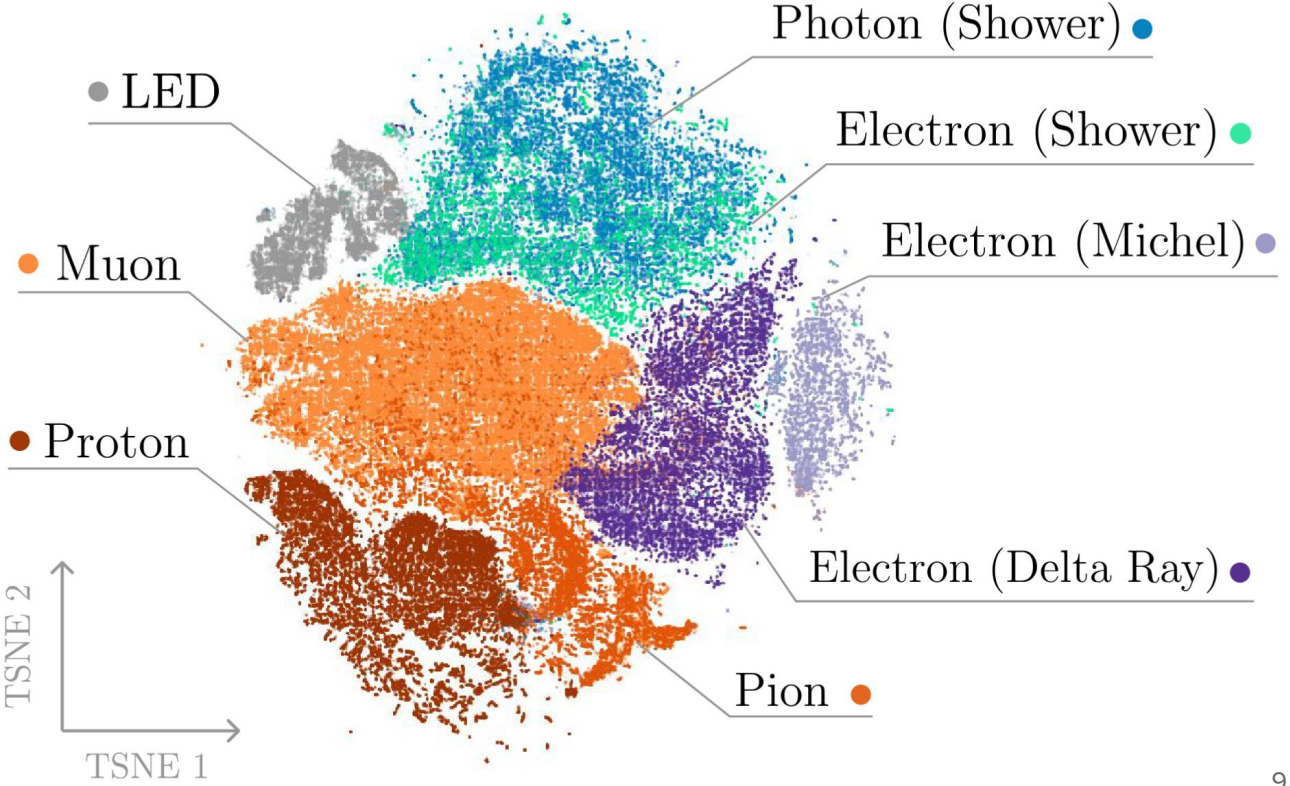
Serialize ➙ cut into non-overlapping patches ➙ windowed attention

(Note: muon length O(1 m) ⇒ ~20 coarse voxels / muon)

Learned representations

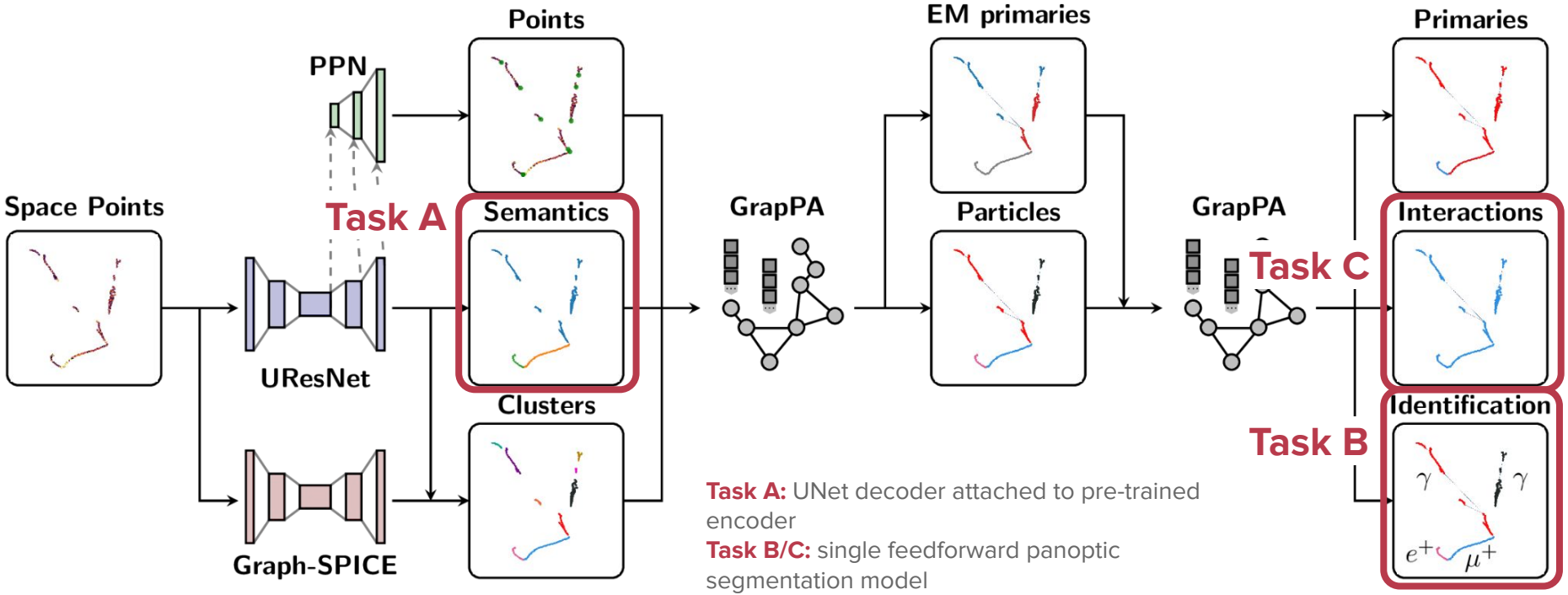
DSet: PILArNet-M

- 1M simulated MPVMPR (particle bomb) events
- 768^3 image
- ~ 2-30k points/event
- ~ 4 +/- 3 interactions/img
- ~ 11 +/- 8 particles/img



Three tasks

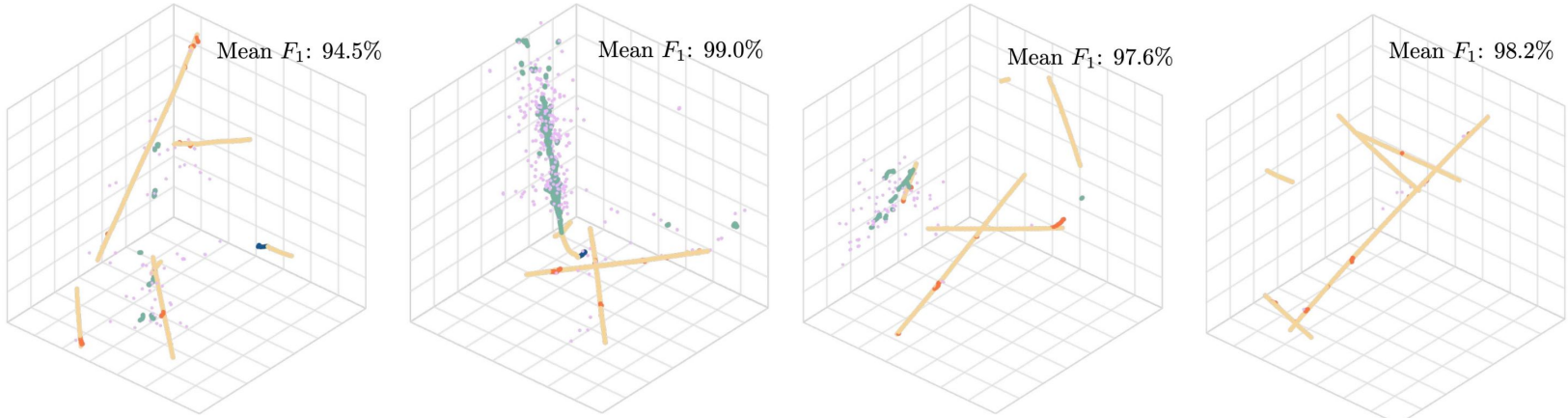
semantic segmentation (A), particle identification (B), interaction identification (C)



Task A: UNet decoder attached to pre-trained encoder

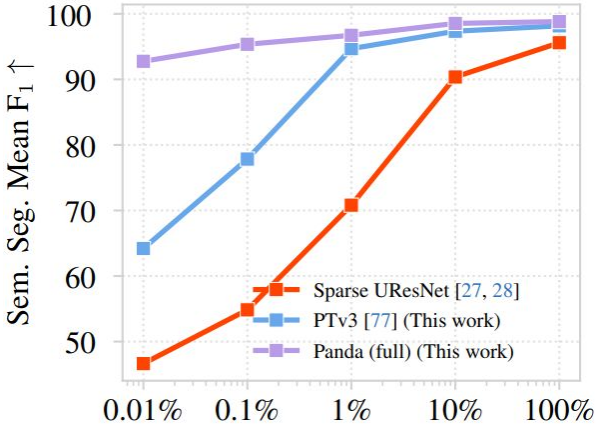
Task B/C: single feedforward panoptic segmentation model

Task A: Semantic Segmentation

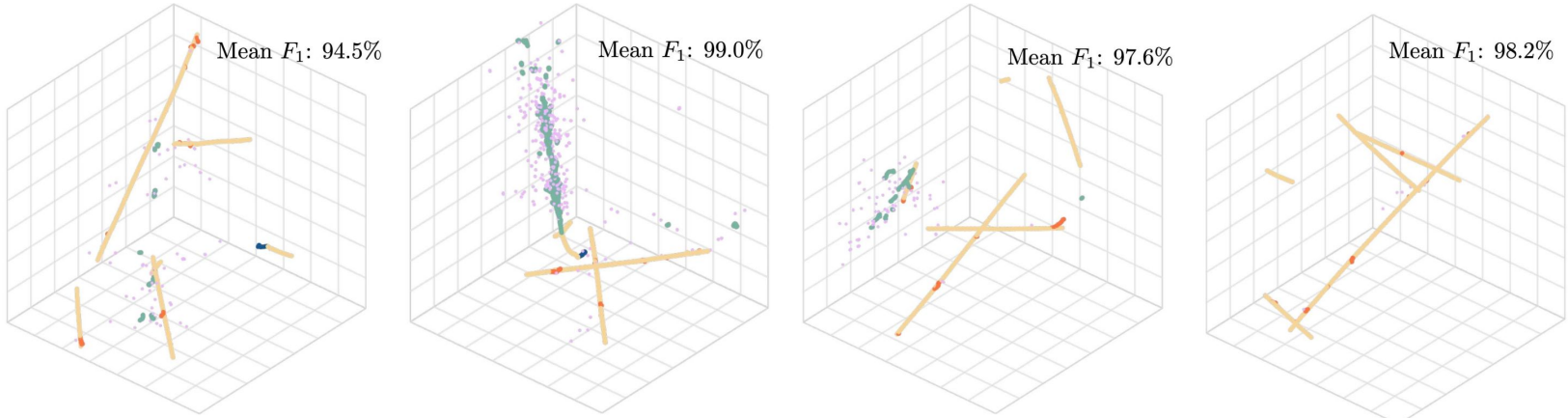


Semantic Segmentation	$PT_{K\%} + FT_{K\%}$					$PT_{100\%} + FT_{K\%}$				
Method / $K\%$	0.01%	0.1%	1%	10%	100%	0.01%	0.1%	1%	10%	100%
Supervised										
○ UResNet [27, 28]	46.6	54.8	70.8	90.4	95.6	46.6	54.8	70.8	90.4	95.6
● PTv3 [77]	64.2	77.8	94.7	97.3	98.2	64.2	77.8	94.7	97.3	98.2
Self-supervised										
● Panda (lin.)	79.1	90.1	93.2	93.7	93.9	92.2	93.6	93.8	93.9	93.9
● Panda (dec.)	83.5	92.8	95.9	97.3	98.2	92.6	95.2	96.2	97.8	98.2
● Panda (full)	**85.2**	**93.7**	**96.4**	**97.7**	**98.8**	**92.8**	**95.4**	**96.7**	**98.5**	**98.8**

○ Prev. SOTA ● This work

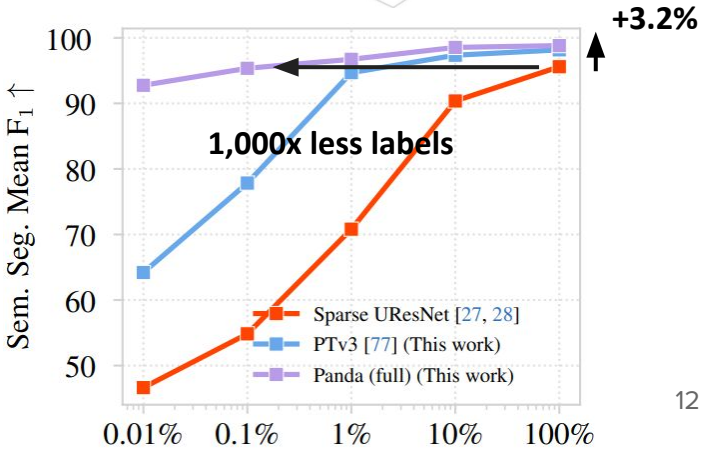


Task A: Semantic Segmentation

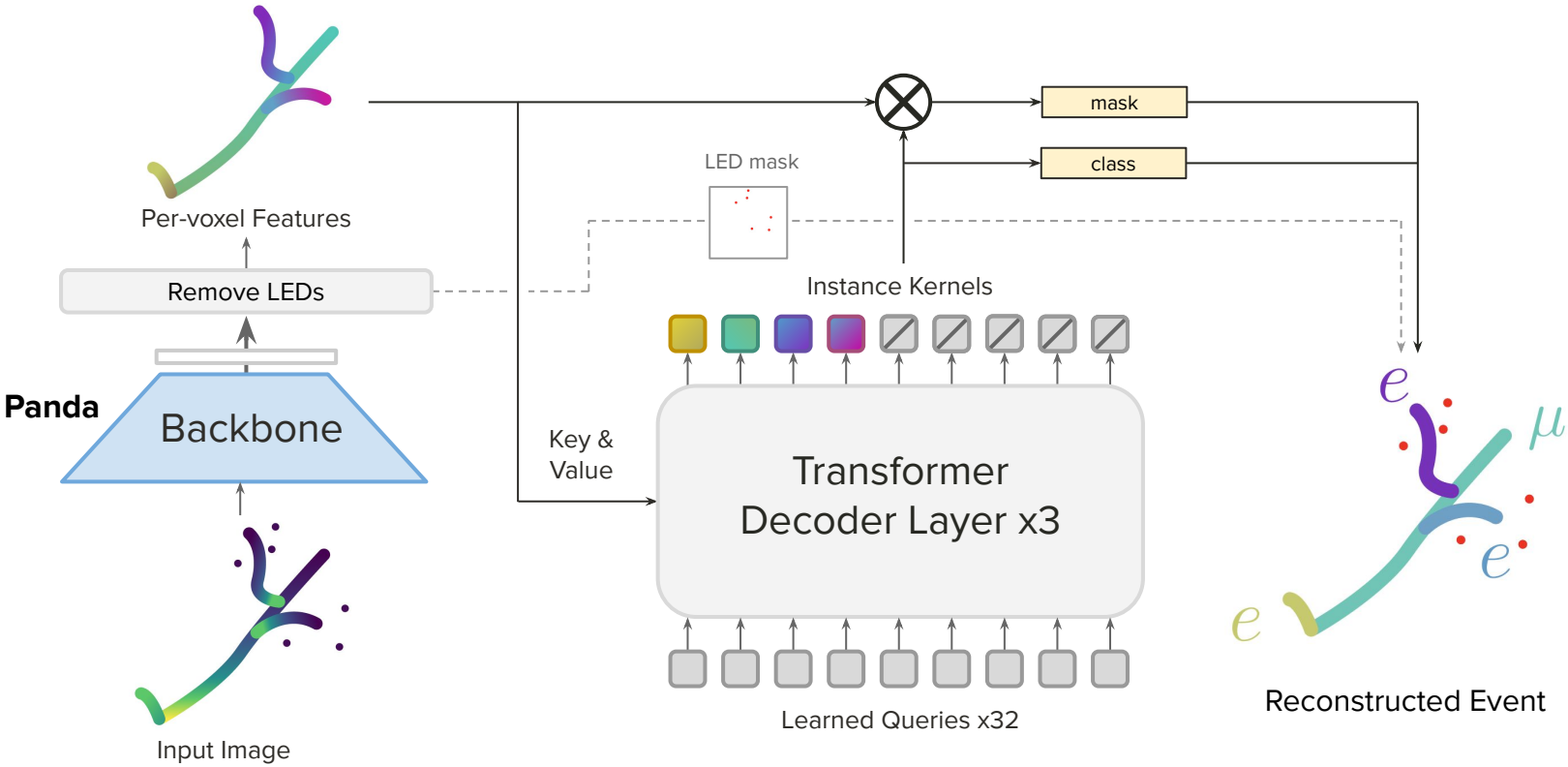


Semantic Segmentation	$PT_{K\%} + FT_{K\%}$					$PT_{100\%} + FT_{K\%}$				
Method / $K\%$	0.01%	0.1%	1%	10%	100%	0.01%	0.1%	1%	10%	100%
Supervised										
○ UResNet [27, 28]	46.6	54.8	70.8	90.4	95.6	46.6	54.8	70.8	90.4	95.6
● PTv3 [77]	64.2	77.8	94.7	97.3	98.2	64.2	77.8	94.7	97.3	98.2
Self-supervised										
● Panda (lin.)	79.1	90.1	93.2	93.7	93.9	92.2	93.6	93.8	93.9	93.9
● Panda (dec.)	83.5	92.8	95.9	97.3	98.2	92.6	95.2	96.2	97.8	98.2
● Panda (full)	**85.2**	**93.7**	**96.4**	**97.7**	**98.8**	**92.8**	**95.4**	**96.7**	**98.5**	**98.8**

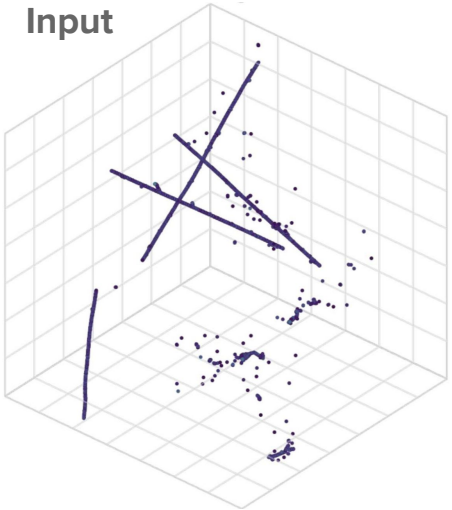
○ Prev. SOTA ● This work

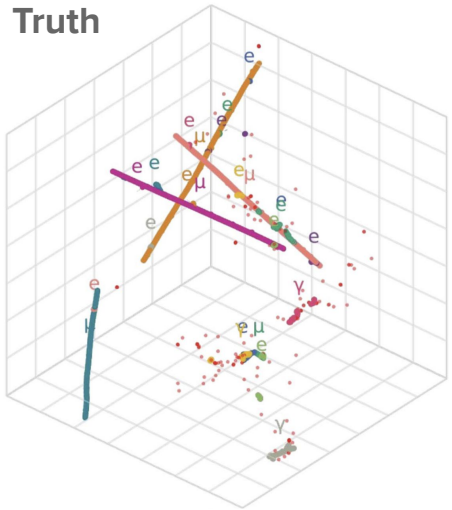


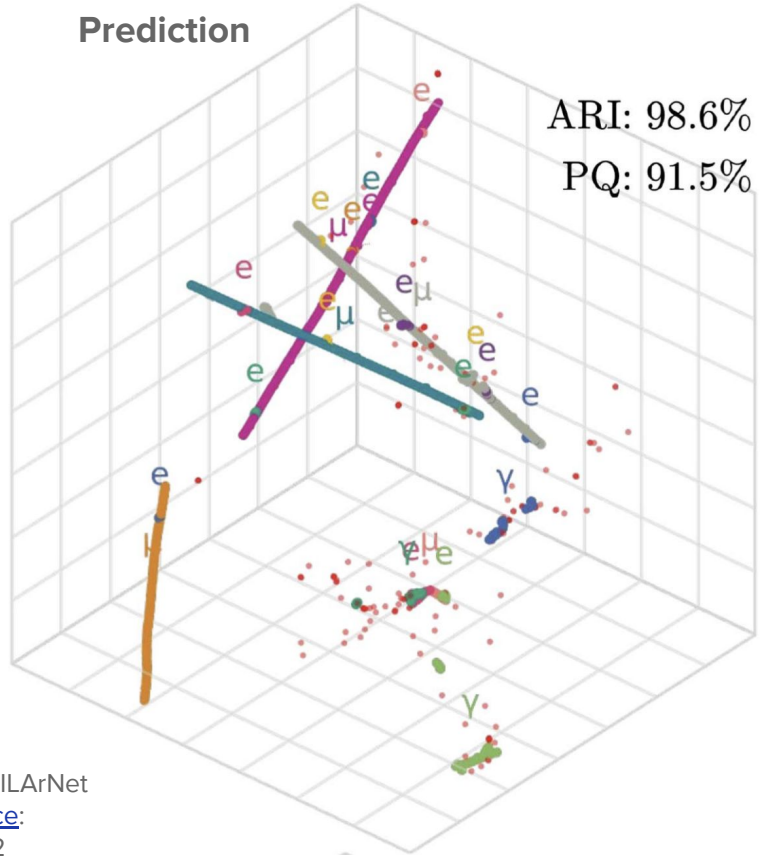
Instance Segmentation: separating particles from one another



Task B: Particle Identification





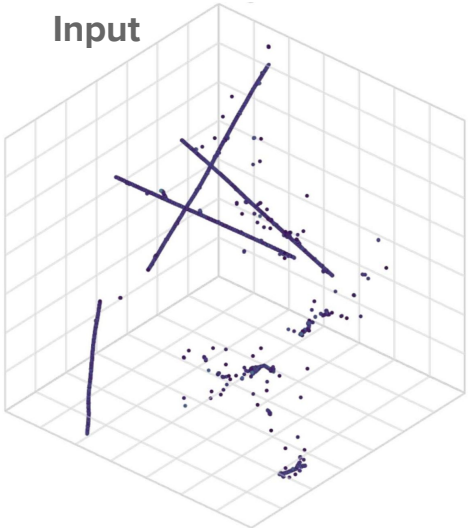


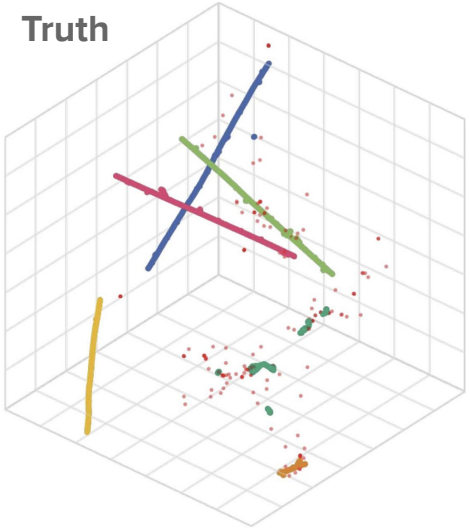
Instance Seg.	Param.	Interaction Id.		Particle Identification						
Method		PQ	ARI	PQ	ARI	γ	e	μ	π	p
Supervised										
• PTv3 [77]	95.1 M	96.3	93.7	86.9	96.6	93.5	94.9	98.6	94.2	98.1
Self-supervised										
• Panda (dec.)	4.4 M	96.6	94.5	89.5	97.3	96.2	96.3	99.1	95.7	98.4
• Panda (full)	95.1 M	**97.6**	**94.9**	**92.5**	**98.0**	**98.4**	**97.2**	**99.3**	**96.0**	**98.6**

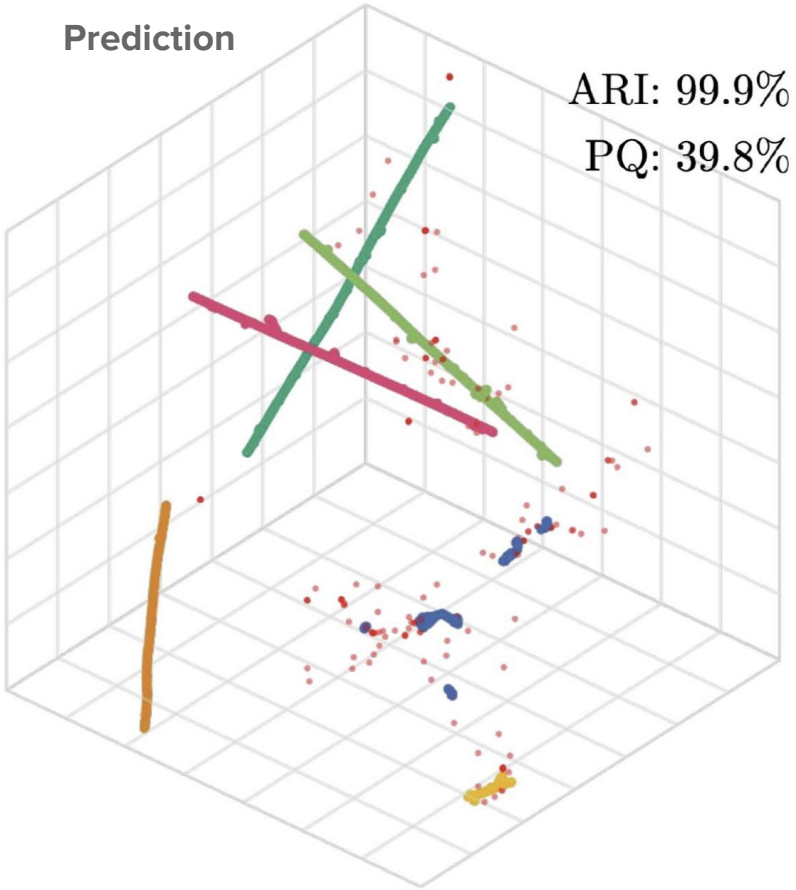
• This work

SPINE PILArNet reference:
ARI 98.2

Task C: Interaction Identification





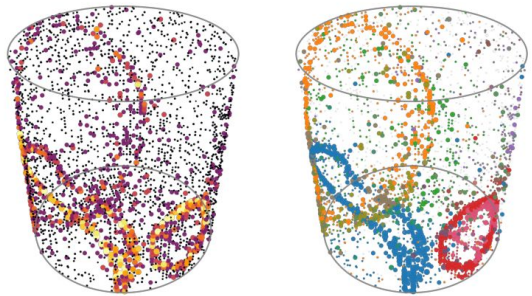


Instance Seg.	Param.	Interaction Id.		Particle Identification						
Method		PQ	ARI	PQ	ARI	γ	e	μ	π	p
Supervised										
• PTv3 [77]	95.1 M	96.3	93.7	86.9	96.6	93.5	94.9	98.6	94.2	98.1
Self-supervised										
• Panda (dec.)	4.4 M	96.6	94.5	89.5	97.3	96.2	96.3	99.1	95.7	98.4
• Panda (full)	95.1 M	**97.6**	**94.9**	**92.5**	**98.0**	**98.4**	**97.2**	**99.3**	**96.0**	**98.6**

• This work

WAND: “Water-cherenkov Annotated Neutrino Dataset”

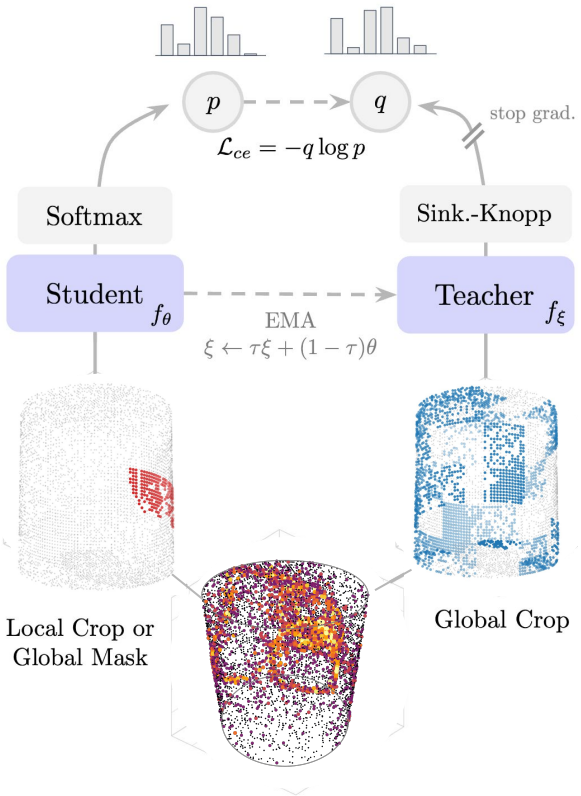
- New open dataset! Full WCh detector sim with:
  - (GENIE +) Geant4 + LUCiD
  - Incl.:
    - Relevant scattering, absorption, etc. all wavelength dependent where applicable
    - PMT sensor response (TTS, QE, AoI acceptance, etc.)
- SuperK-like geometry: (~11k PMTs, 16.9m cylinder)
- Single particle (mu, e, pi+, pi0) as well as multi-ring and GENIE nue/numu
- Isotropic particle guns, uniform 1-2000 MeV
- For this work, ~120k total events, ~10k from each config
- See Cesar’s talk tomorrow for in-depth information.

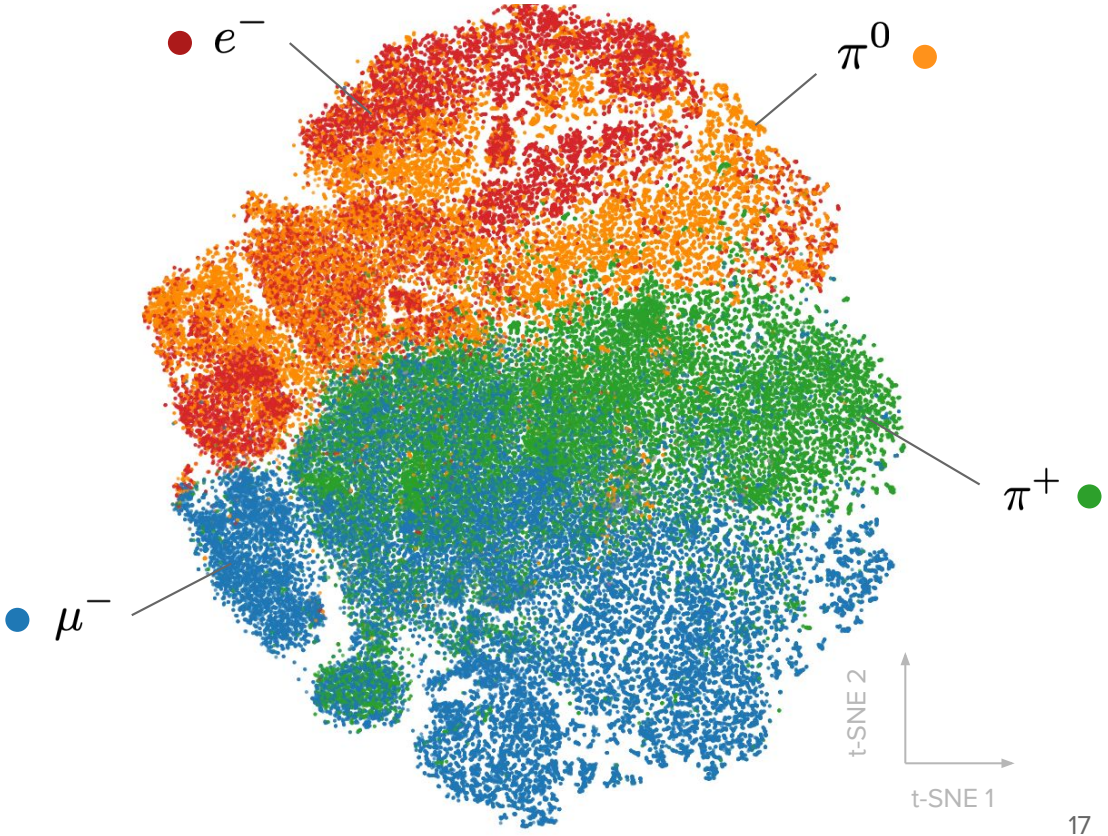


Particle content
μ^-
π^+
e^-
π^-
π^0 ($\rightarrow\gamma\gamma$)
e^- (low-energy)
μ^-, π^+
e^-, π^+
e^-, π^0 ($e+\gamma\gamma$)
μ^-, π^+, π^0
μ^-, π^+, π^-
e^-, π^+, π^0
ν_μ interaction (CC/NC)
ν_e interaction (CC/NC)
$\mu^- + \pi^+$ gun $\oplus$ ν_μ
$\mu^- \oplus \pi^+$ (2 vtx)
$\nu_\mu \oplus \nu_e$
2 bombs, 1–5 particles each

WORK IN PROGRESS

Learned representations

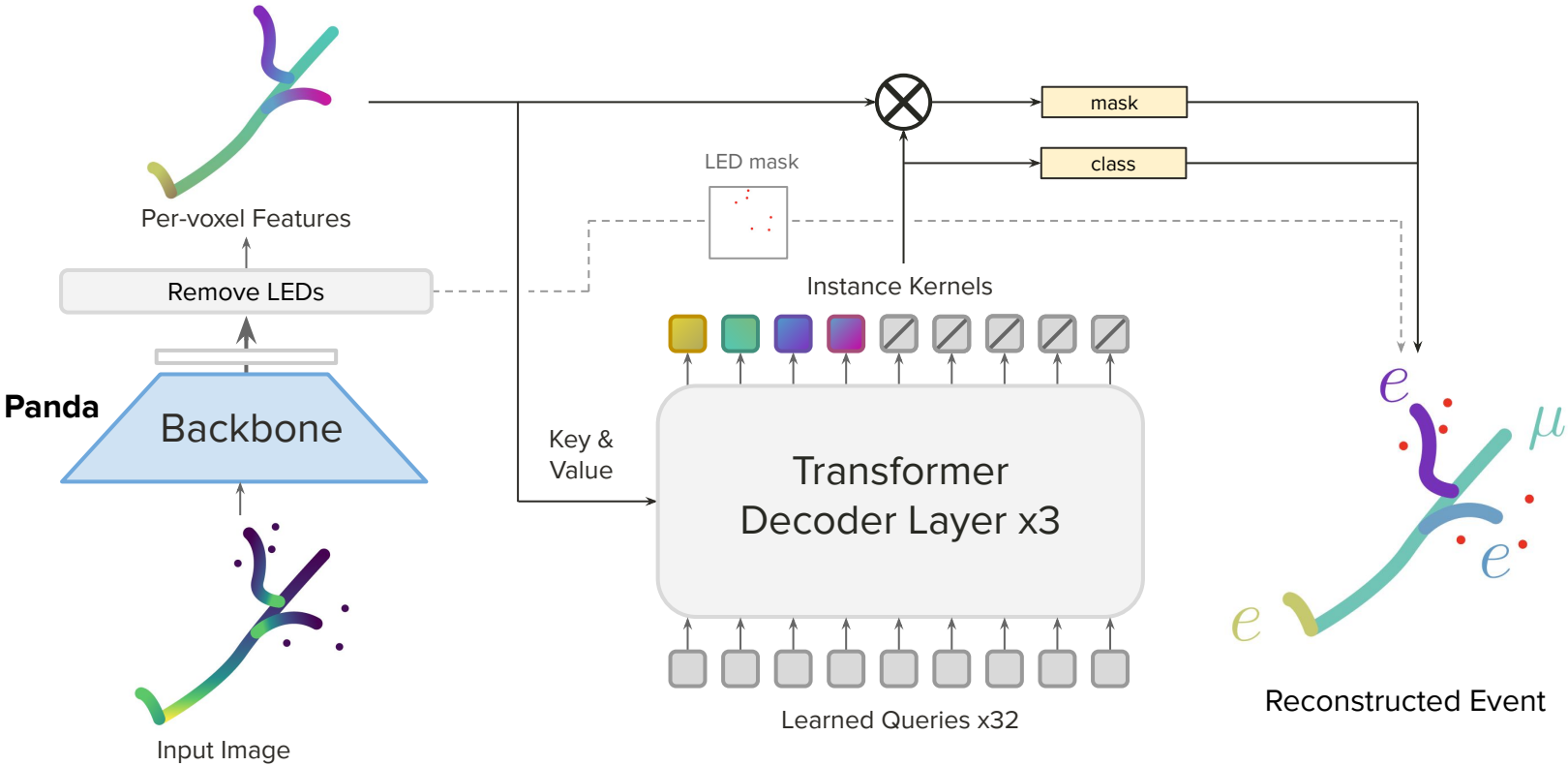




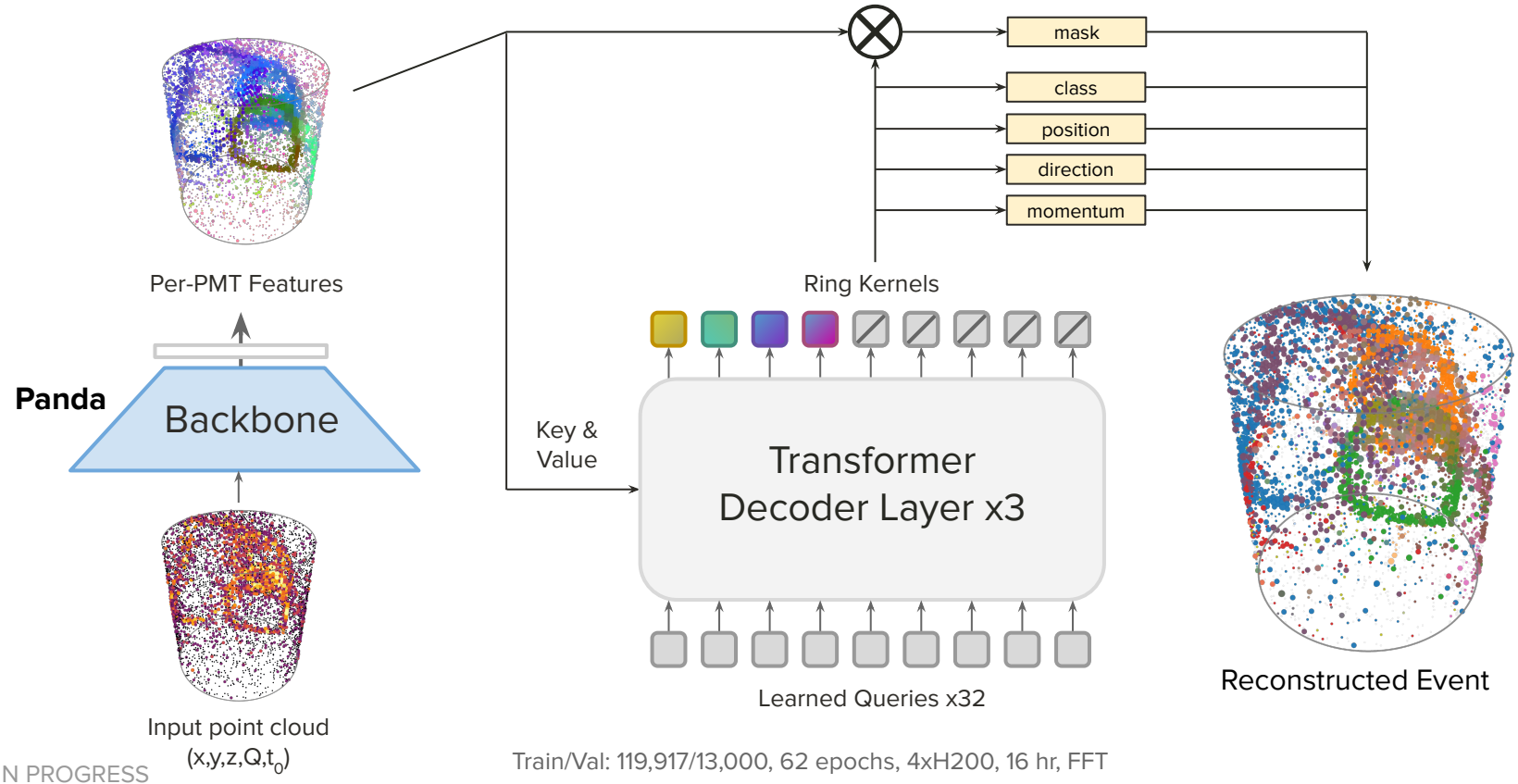
WORK IN PROGRESS

Pre-train: 119,917 events, 50 epochs, 4xH200, 36 hr

Instance Segmentation: separating particles from one another



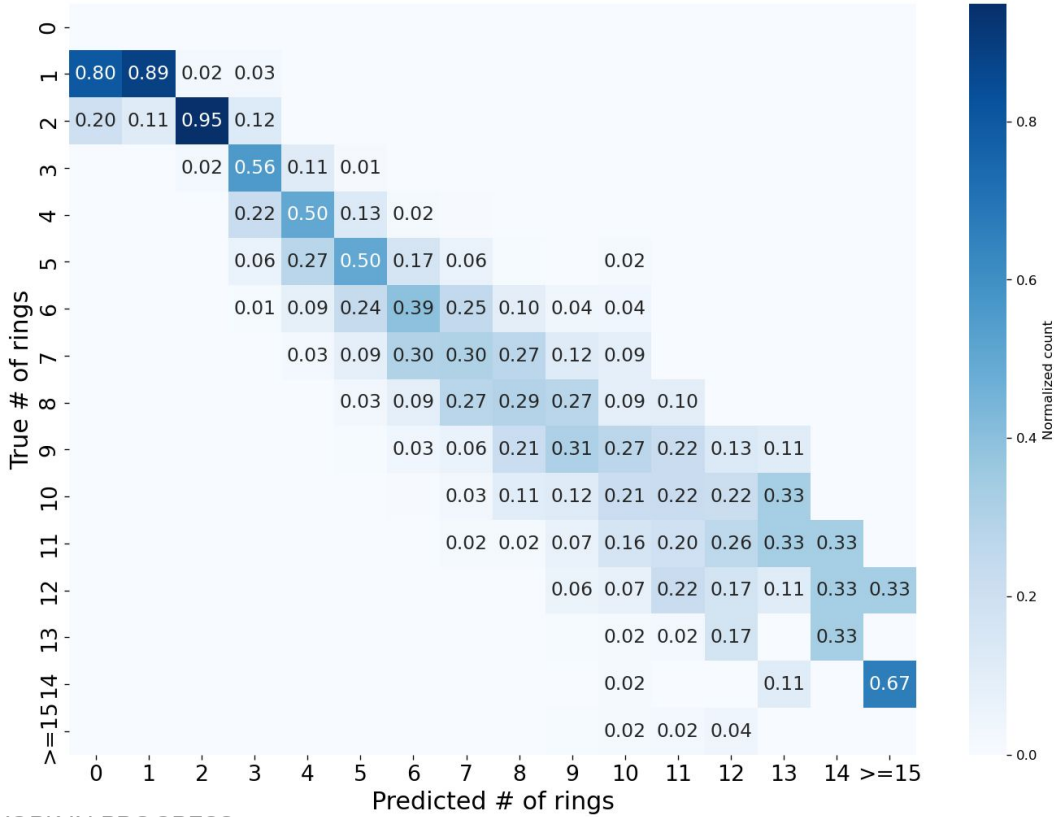
Instance Segmentation: separating ~~particles~~**rings** from one another



Train/Val: 119,917/13,000, 62 epochs, 4xH200, 16 hr, FFT

WORK IN PROGRESS

General ring counting



Note:

- **Using entire validation set**
- **Ring = any particle that has recorded PMT hits**

True Number of Rings	fiTQun Reconstruction 1R	2R	≥ 3R
True 1R	95.0%	4.64%	0.41%
True 2R	27.8%	66.7%	5.56%
True ≥3R	7.04%	25.5%	67.5%

FiTQun on fully contained atmospherics (not apples to apples) (arXiv:1901.03230)

WORK IN PROGRESS

Single particle reconstruction

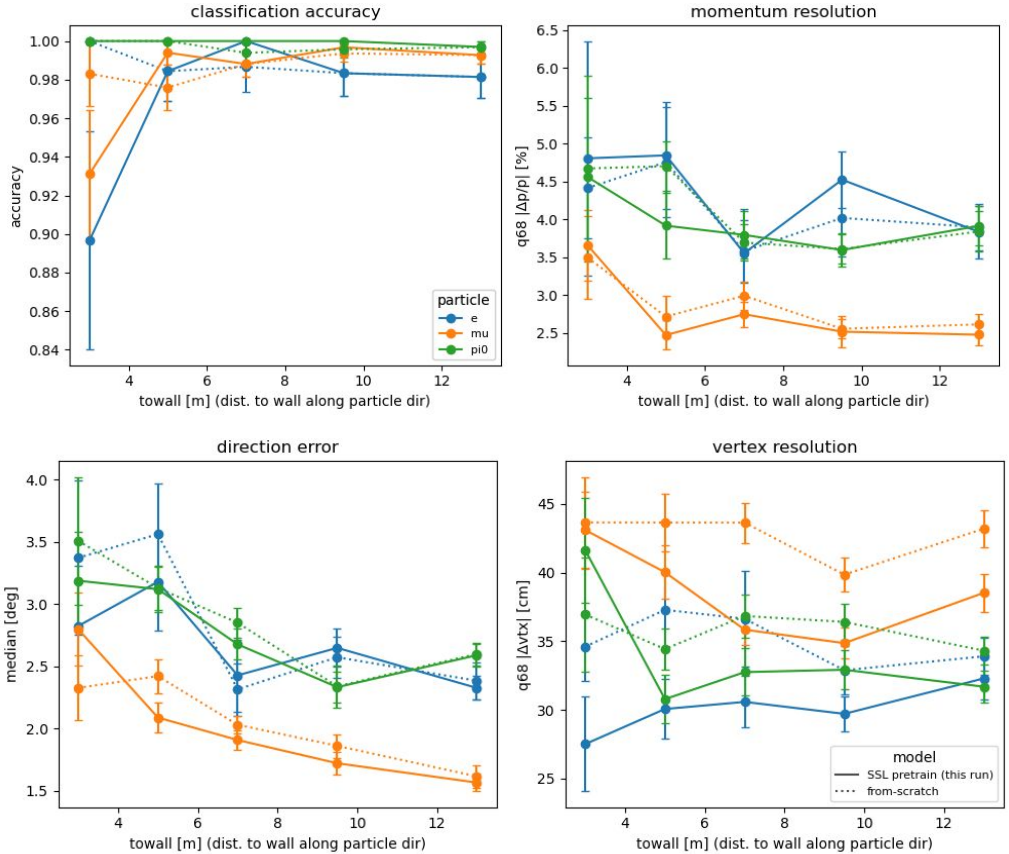
Cuts:
dist→wall > 2 m
E_{vis} < 1330 MeV
Fully contained

99.9% e- efficiency @ 99.9% mu- rejection

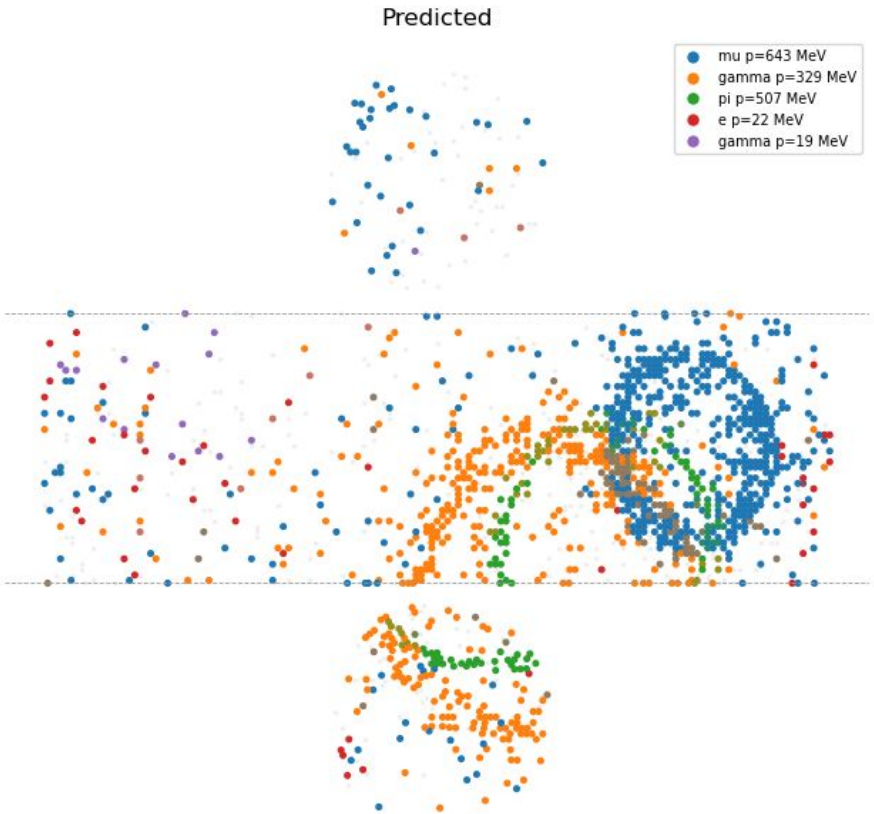
85.5% e- efficiency @ 95% pi0 rejection

e
mu
pi0

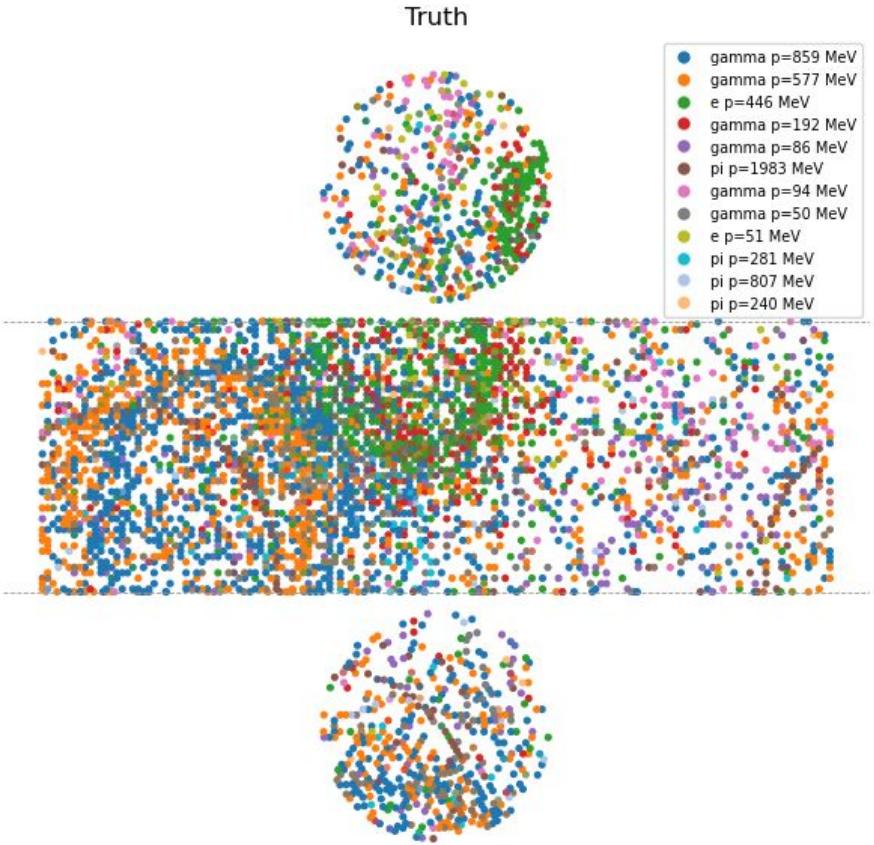
— SSL pretrain + finefine
- - from-scratch



Multi-ring reconstruction

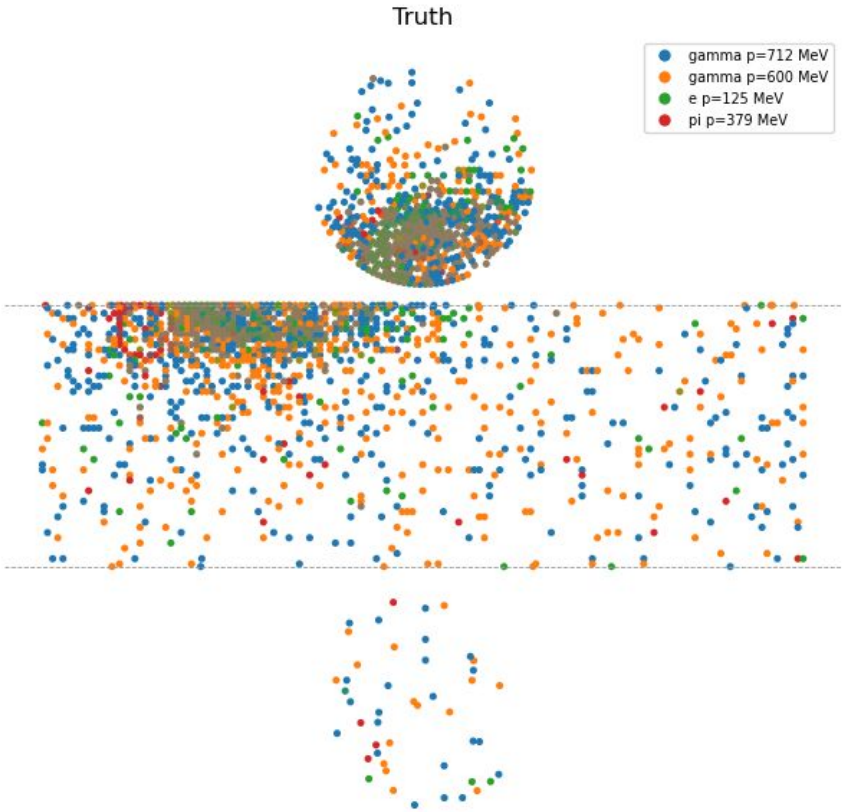


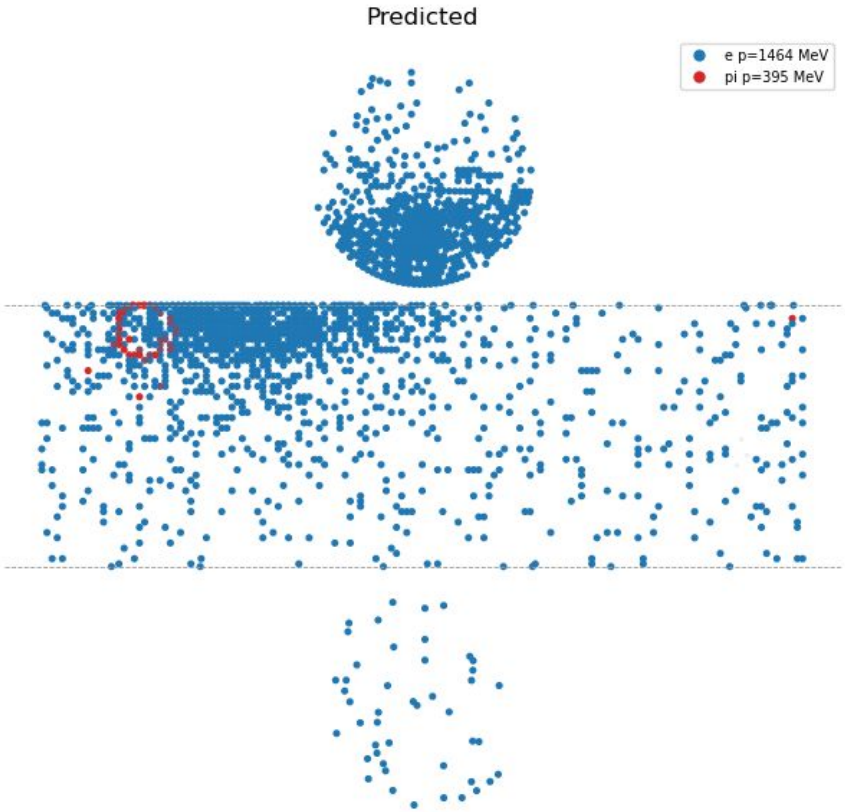
Multi-ring reconstruction



WORK IN PROGRESS

Multi-ring reconstruction





WORK IN PROGRESS

π^0 mass reconstruction using 1,000 e^- + π^0 events

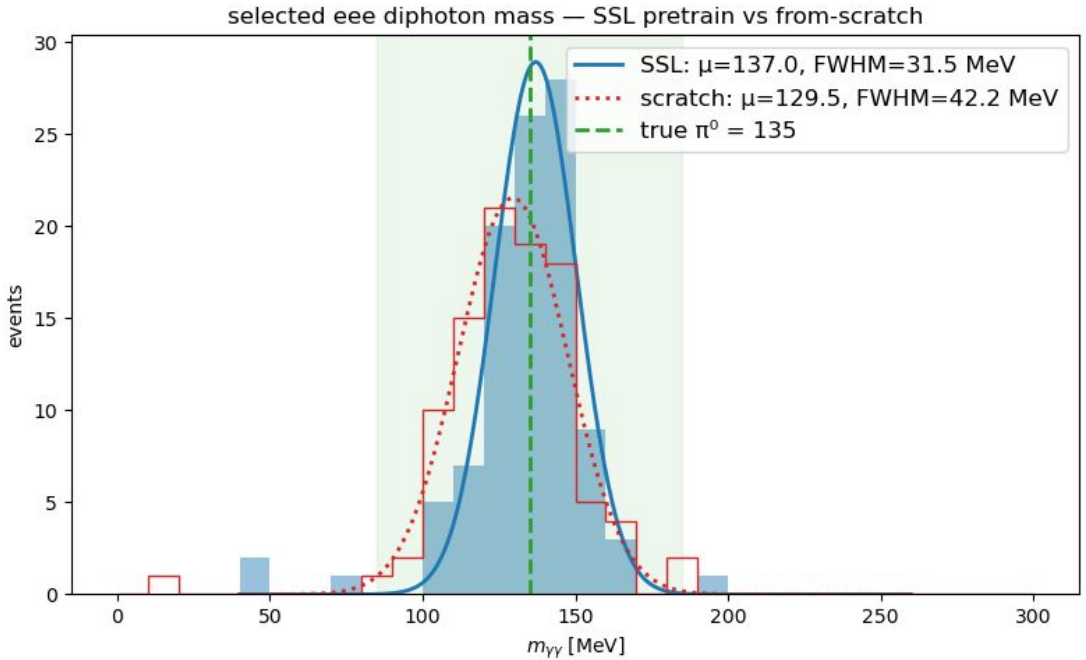
Cut (roughly following arXiv:1901.03230)

- Min points in ring: 5
- High confidence (> 0.9) PID only
- dist→wall > 2 m
- dist→wall along particle dir > 2 m
- E_{vis} < 1330 MeV
- 3 predicted rings only
- All rings e-like {e, gamma}

(1000 → ~100 events)

Reconstruction:

- PT+FT: 137.0 +/- 31.5
- Scratch: 129.5 +/- 42.2



Note: not CCν$_e$1π^0, so e and π^0 are uncorrelated

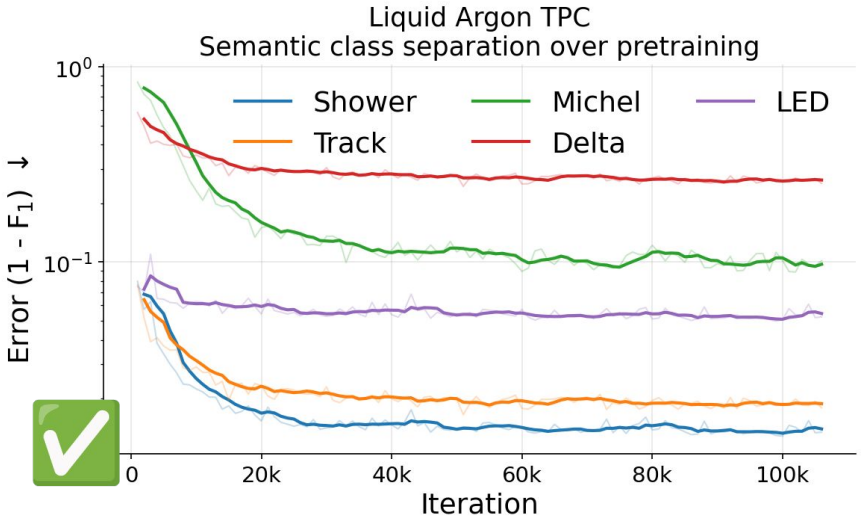
$$m_{\gamma\gamma} = \sqrt{2E_1E_2(1-\cos\theta_{12})}$$

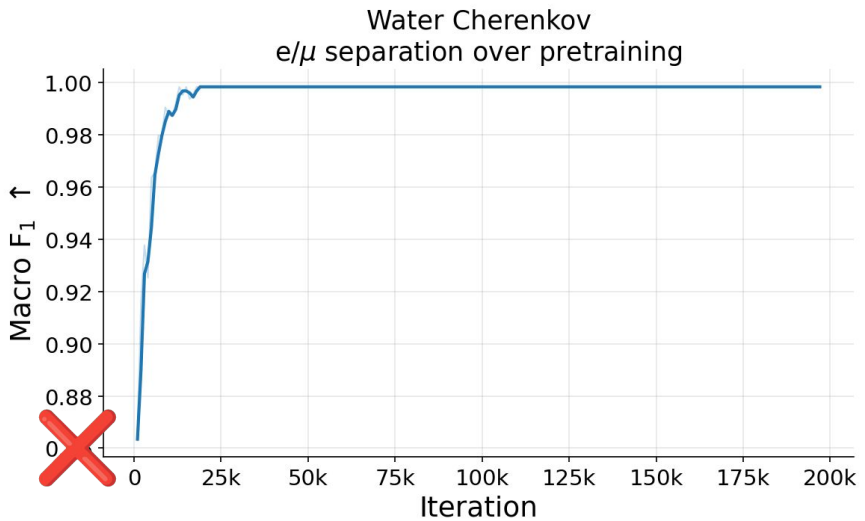
WORK IN PROGRESS

Lessons learned, and challenges ahead

1. Think of cheap, but difficult, probes that you can use to see how well SSL is learning over the course of training.

Getting SSL to work is extremely difficult and computationally time-intensive. However in these methods I've found that you can see whether it's "working" quite early in training if you run linear probes on weight checkpoints.





Lessons learned, and challenges ahead

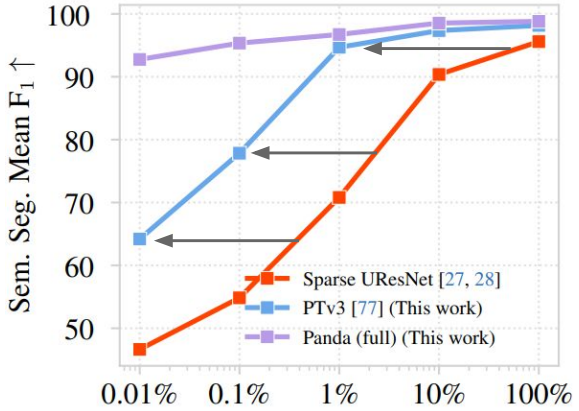
2. Data augmentations can contribute to an effective increase in dataset size by 1 OOM, generally (arXiv:2106.10270)

- Unphysical augmentations (e.g., random rotations in a LArTPC) can be extremely beneficial for pattern recognition tasks. Think of it as jointly learning the “superset” of transformations, including physical + non-physical.
- For places where you need exact physics (e.g., deriving space charge effects), small fine-tuning models should (imo) be sufficient (but this is not proven).

Difference between **orange** and **blue**

=

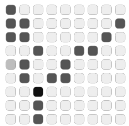
Improved architecture + data augmentation



Lessons learned, and challenges ahead

3. FM R&D requires designing systems that are general, extensible, and can scale.

Consider using **pimm** if starting out!



particle-imaging-models

Train, fine-tune, eval scripts of foundation models for neutrino physics

[github] https://github.com/deeplearnphysics/particle-imaging-models

Local

```
> pimm launch \
>       --resources.nproc_per_node 2 \
>       --config pretrain/panda.py
```

SLURM Cluster

```
> pimm submit \
>       --site nersc \
>       --resources.nnodes 2 \
>       --resources.nproc_per_node 4 \
>       --resources.time 24:00:00 \
>       --config pretrain/panda.py
```

Some features:

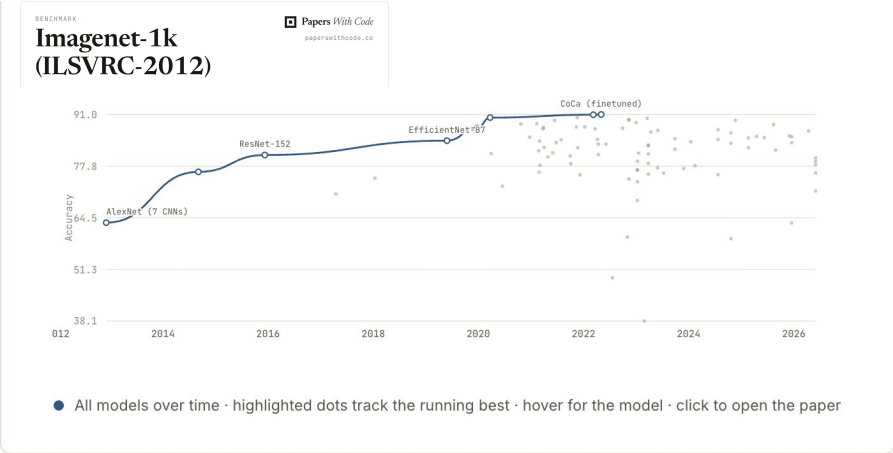
- Apptainer/Docker images
- Easy logging to Weights and Biases (wandb)
- Automatic mixed precision (AMP) training.
- Multi-node training using `torchrun`
- DDP for multi-GPU and multi-node training, with (far) future support for FSDP2.
- Fully deterministic checkpointing: start and stop runs as if it never stopped.
- Auto-export and loading models from HuggingFace.
- Validated on SLAC S3DF and NERSC Perlmutter using site-specific configurations.
- Point cloud only (for now–1D & 2D in the works)
- NVIDIA only

Lessons learned, and challenges ahead

1. Common benchmarks and reproducibility

- Deep learning has proliferated because of benchmarks (ImageNet, ScanNet, S3DIS, etc.). LHC relies on them heavily (JetClass, Aspen Open Jets, ATLAS Top Tagging, ...). We desperately need benchmarks. Otherwise, we cannot compare our methods together, which means slowed scientific progress on methods.
- ...and is aligned with Genesis mission
- Please consider contributing your models to the DORAEMON open data challenge (LArTPC, WCh, soon IceCube, etc...) leaderboards (see Kazu's talk tom) come Q4 2026, or at least open source your method with reproducibility instructions and dataset (MicroBooNE open data, MINERvA, PILArNet, PILArNet-M).

It's worth it!



Lessons learned, and challenges ahead

1. Common benchmarks and reproducibility

- Deep learning has proliferated because of benchmarks (ImageNet, ScanNet, S3DIS, etc.). LHC relies on them heavily (JetClass, Aspen Open Jets, ATLAS Top Tagging, ...). We desperately need benchmarks. Otherwise, we cannot compare our methods together, which means slowed scientific progress on methods.
- ...and is aligned with Genesis mission
- Please consider contributing your models to the DORAEMON open data challenge (LArTPC, WCh, soon IceCube, etc...) leaderboards (see Kazu's talk tom) come Q4 2026, or at least open source your method with reproducibility instructions and dataset (MicroBooNE open data, MINERvA, PILArNet, PILArNet-M).

It's worth it!

2. Inefficiency

Point cloud learning is famously an unsolved field. PTv3 is SOTA, but is inefficient by the field's standards.

- ~60-70% of parameters come from the massive 3D sparse CNN kernels, with the rest going to the actual attention blocks. **Ratio of kernel to attn block params = $3^3d^2/12d^2$ = 27/12 = 2.25**
- I found that forcing the sparse CNN to operate in low dimensions even in late depths reduces parameters by half and keeps the same performance.
- See also LitePT, Volume Transformer for very recent work on this issue.

Lessons learned, and challenges ahead

3. Scaling laws

- No understanding of predictive scaling laws in neutrino physics (see collider example here)
- IMO, pre-training recipes should be general enough to be applied easily across similar problems in nu phys. We do not have this. Panda took ~4-5 months of tuning to work on LArTPC, ~1-2 week for WCh.

4. Parameter inefficient architectures.

Unlike natural images/language, point cloud learning is an “unsolved” field. Point Transformer v3 is SOTA, but is extremely inefficient.

For example:

- e.g., ~60-70% of parameters come from the massive 3D sparse CNN kernels, with the rest going to the actual attention blocks.
    - I found that forcing the sparse CNN to operate in low dimensions even in late depths reduces parameters by half and keeps the same performance.
- See also LitePT (arXiv:2512.13689), Volume Transformer (arXiv:2604.19609)

Summary and outlook

My site: https://youngsm.com/

Summary:

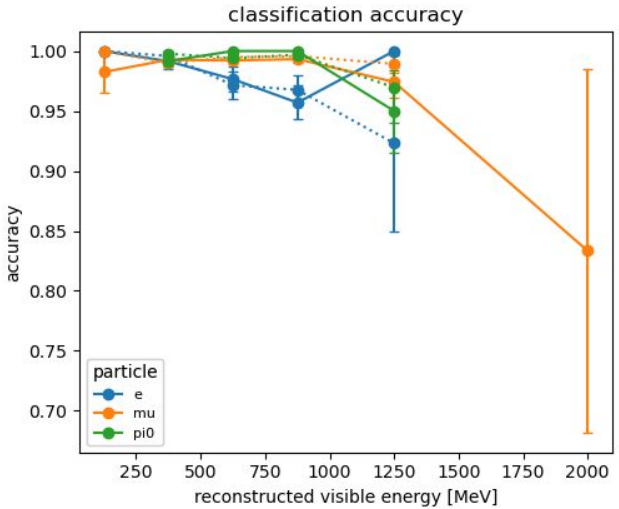
- Methods of label-free self-supervised learning in popular in computer vision can be applied successfully to HEP pattern recognition problems in different detection regimes (TPC, WCh) with minimal changes.
- The field is moving towards larger models, more data, and more compute.
  - High upfront cost of pretraining is largely amortized across downstream tasks.
  - The Genesis mission is also pushing heavily towards this…
- Efforts currently focus on efficient transfer learning across single-domain tasks, but…

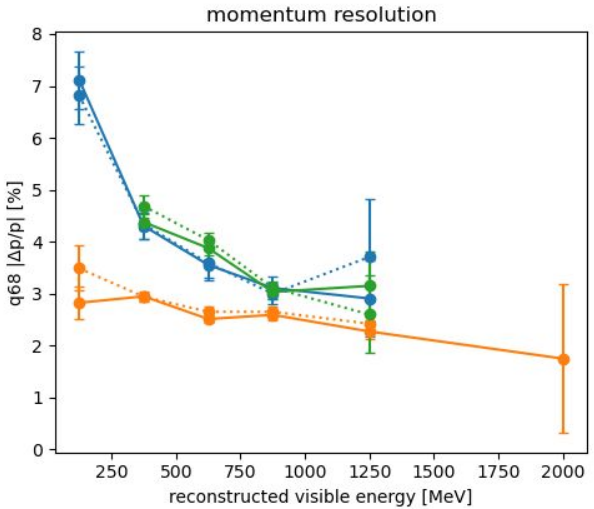
Outlook:

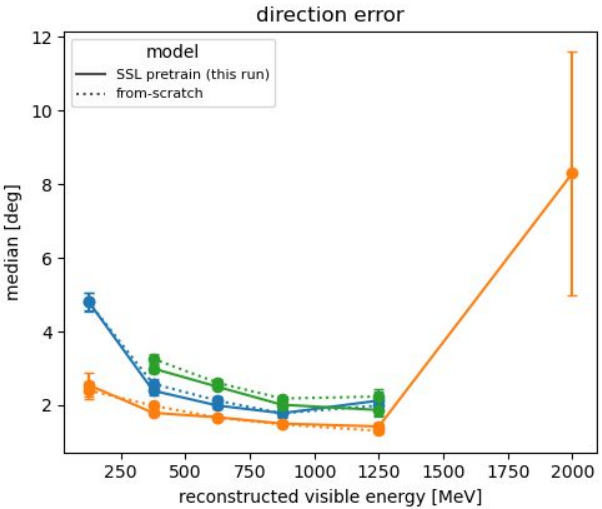
- There is current interest to experiment with cross-experimental FMs (e.g., FM for DUNE ND+FD).
- Multi-modal pre-training being investigated (e.g., charge + light)
- Looking into seeing “FM”-ness by fine-tuning on other open datasets.
- AI-ready 200 TB dataset with O(10M) LArTPC + Water Cherenkov events targeted for Q4 2026. (See Kazu’s talk and Cesar’s talk on Friday)
- Please contribute (or create) easily reproducible common benchmarks on important tasks.

Extras

Function of E_{vis}







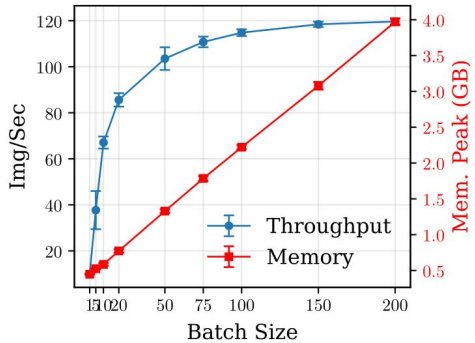
Needs!

Resources

- Compute: larger models, more compute, more data = better (FM scaling law)
  - Vision models are dominated largely by data memory (as opposed to a model)

Panda (A100)

Pre-training 10 epochs for 1M image dataset (~ 10 GPU-day)



Config.	Batch	Params	Wall-time [s/iter]	Data Xfer [s/iter]	VRAM [GB]
Pre-train	12	104M	0.895	0.005	34.3
SemSeg (dec)	12	16.3M	0.210	0.001	3.4
PanSeg (vtx-dec)	6	4.3M	0.292	0.005	5.3
PanSeg (pid-dec)	6	4.3M	0.321	0.004	12.7

Needs!

Resources

- Compute: larger models, more compute, more data = better (FM scaling law)
  - Vision models are dominated largely by data memory (as opposed to a model)
  - A tiny R&D models 5-10 A100-day, for a large model x10-x100
  - Fine-tuning (task-specific models) ~ 1 A100-day per task
- Storage: large data size
  - PILArNet-1M (3D point cloud) ~167GB
  - PILArNet-10M (+ TPC/Optical waveforms) ~200TB
  - Publishing “encoded, AI-ready dataset” … somewhere in between?

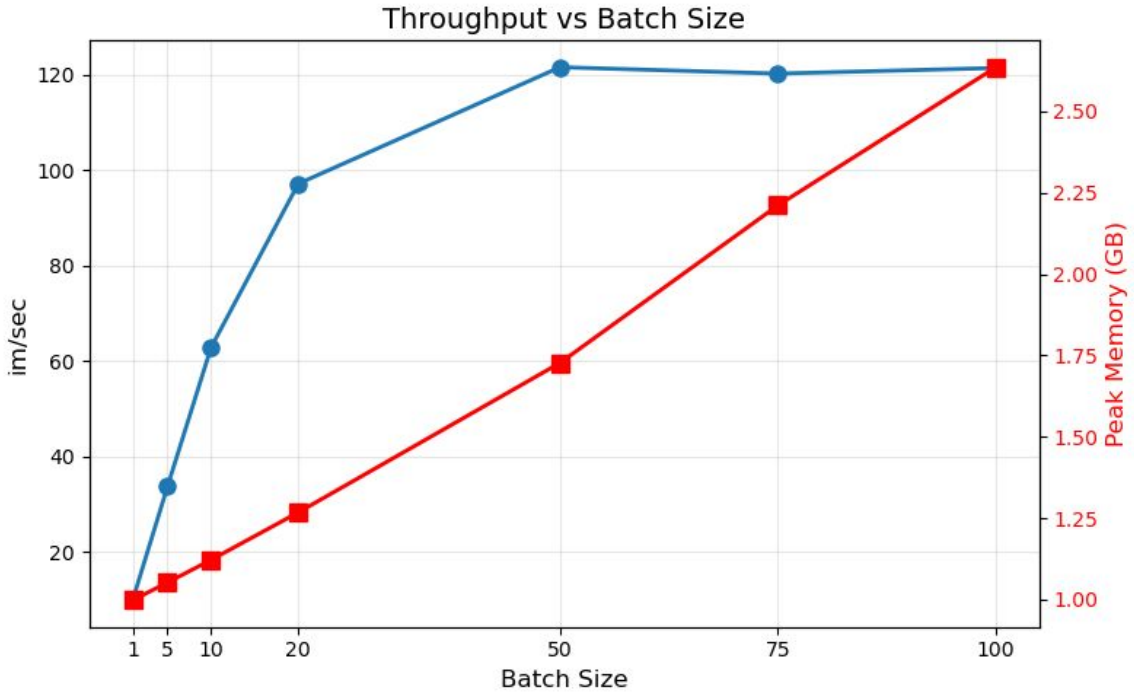
Needs!

Resources

- Compute: larger models, more compute, more data = better (FM scaling law)
  - Vision models are dominated largely by data memory (as opposed to a model)
  - A tiny R&D models 5-10 A100-day, for a large model x10-x100
  - Fine-tuning (task-specific models) ~ 1 A100-day per task
- Storage: large data size
  - PILArNet-1M (3D point cloud) ~167GB
  - PILArNet-10M (+ TPC/Optical waveforms) ~200TB
  - Publishing “encoded, AI-ready dataset” … somewhere in between?
- Software ecosystem - (always containers)
  - The core: pytorch, spconv, flash-attn, torch-scatter, pytorch-lightning, omegaconf (hydra), pybind11, hugging-face
  - Auxiliary: numba, cupy, jax, WandB, warpconvnet

Scalability

Semantic Segmentation (measured on single A100)

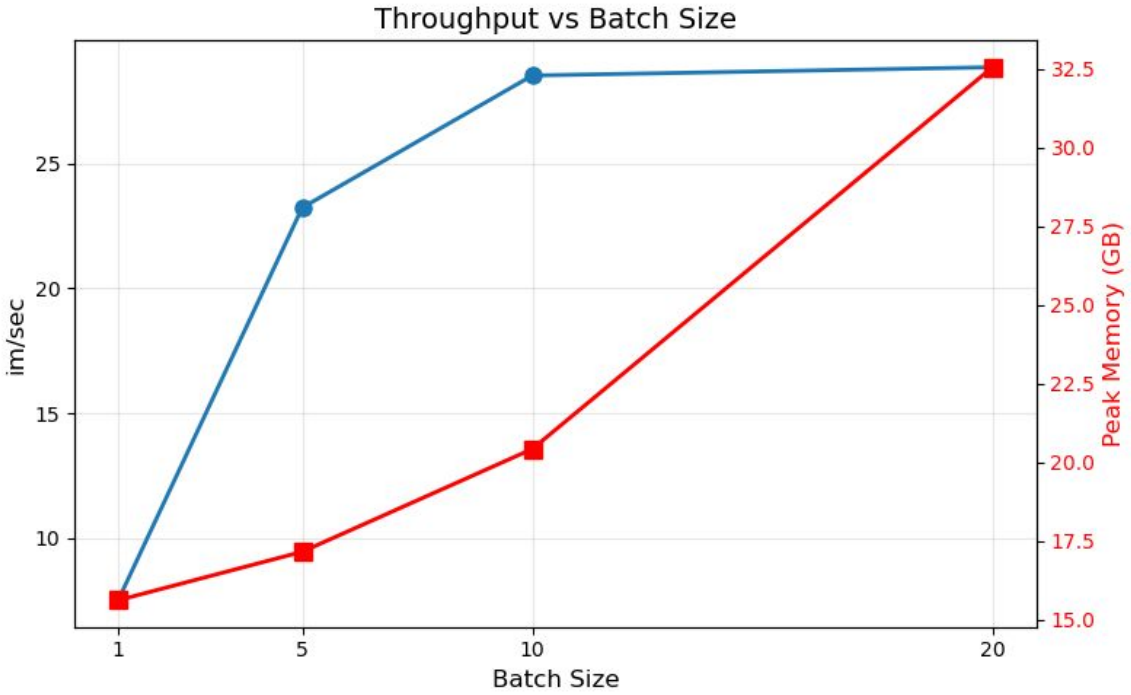


Peak throughput:
8.3 ms/image
120 img/sec

Mem@peak:
~1.75 GB

Scalability

Instance/Panoptic Segmentation (model forward + NMS post-processing)



Peak throughput:

34.5 ms / image

29 img/sec

Mem@peak:

~20 GB

NMS post-processing is serial, but parallelized via multiprocessing

Sharpening + Centering

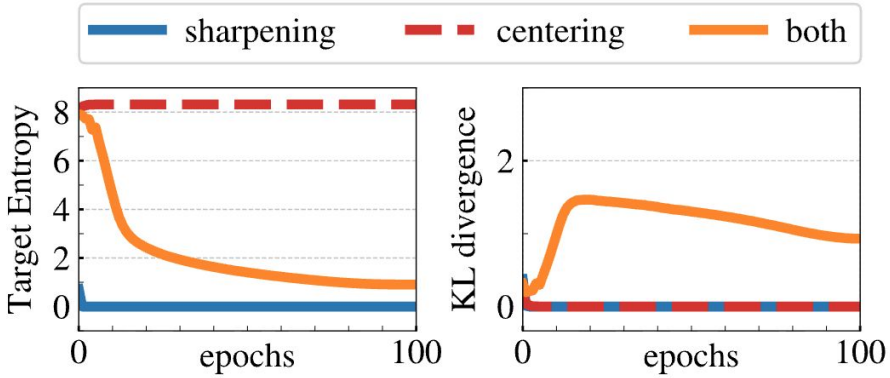


Figure 7: **Collapse study. (left)**: evolution of the teacher's target entropy along training epochs; **(right)**: evolution of KL divergence between teacher and student outputs.

There are two forms of collapse: regardless of the input, the model output is uniform along all the dimensions or dominated by one dimension. The centering avoids the collapse induced by a dominant dimension, but encourages an uniform output. Sharpening induces the opposite effect. We show this complementarity by decomposing the cross-entropy H into an entropy h and the Kullback-Leibler divergence ("KL") D_{KL}:

$$H(P_t, P_s) = h(P_t) + D_{KL}(P_t|P_s). \quad (5)$$

A KL equal to zero indicates a constant output, and hence a collapse.

Sharpening:

$$P_s(x)^{(i)} = \frac{\exp(g_{\theta_s}(x)^{(i)}/\tau_s)}{\sum_{k=1}^{K} \exp(g_{\theta_s}(x)^{(k)}/\tau_s)}, \quad (1)$$

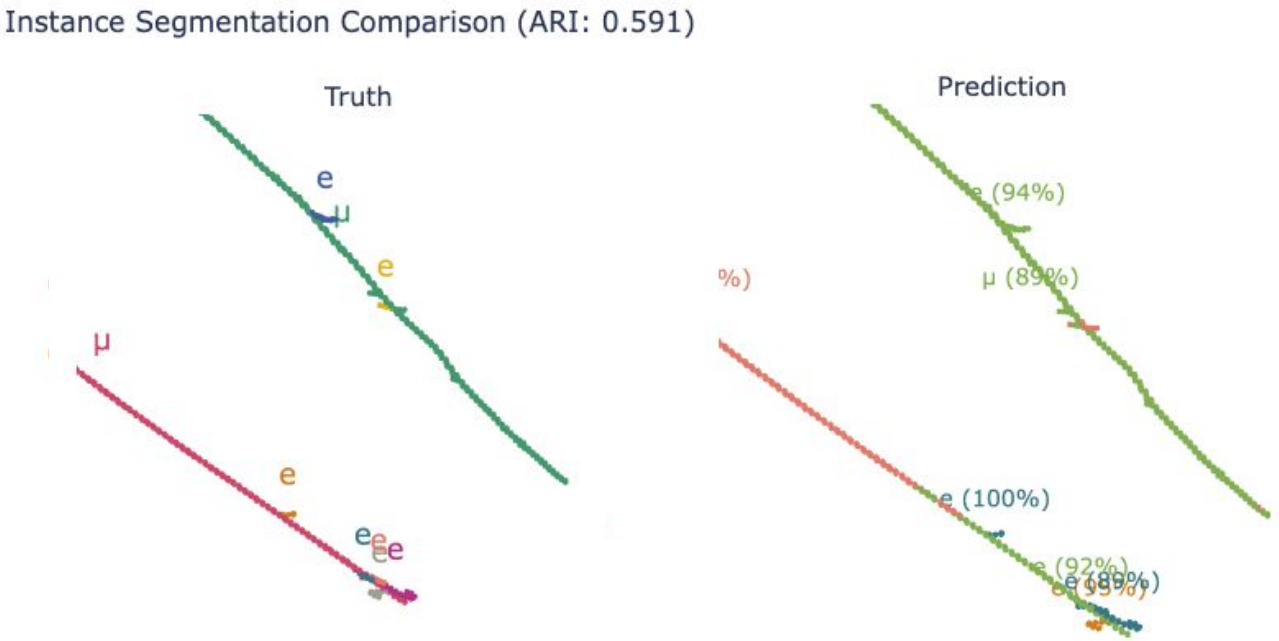
with $\tau_s > 0$ a temperature parameter

From *Emerging Properties in Self-Supervised Vision Transformers* (arXiv:2104.14294)

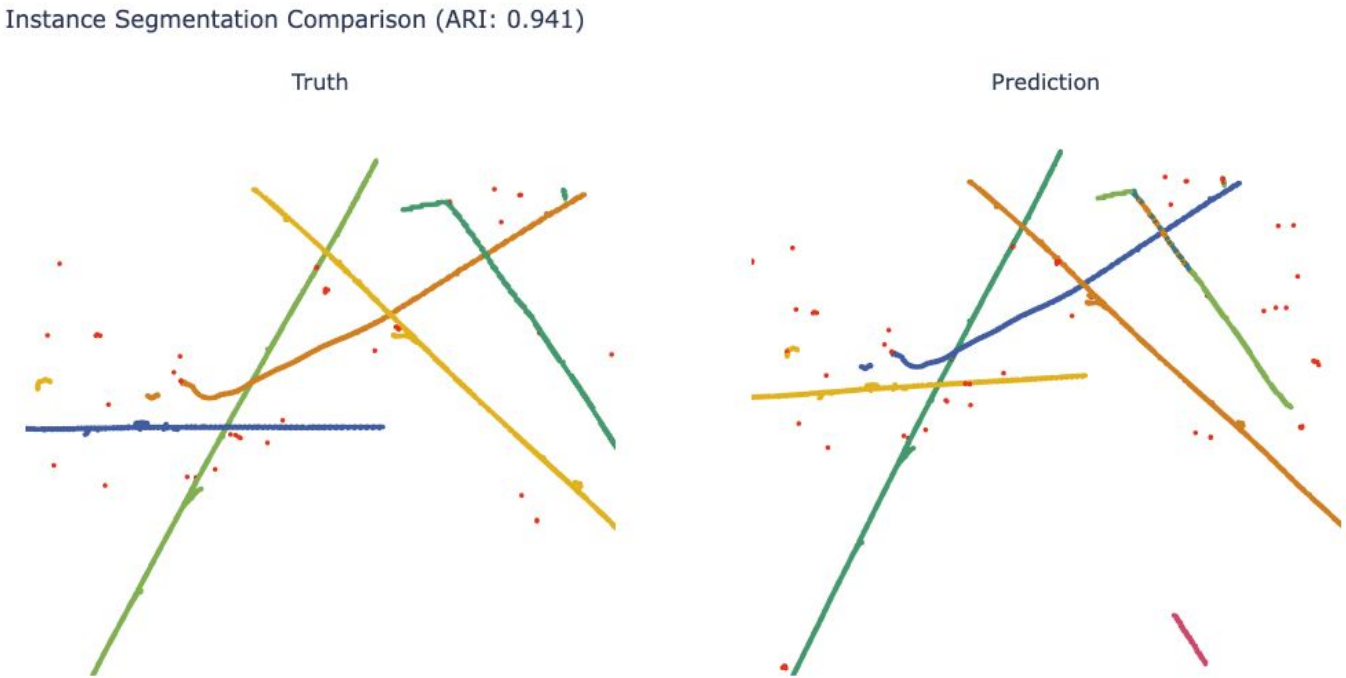
Augmentations

What about diffusion/attenuation?

Poor instance reconstruction – particle

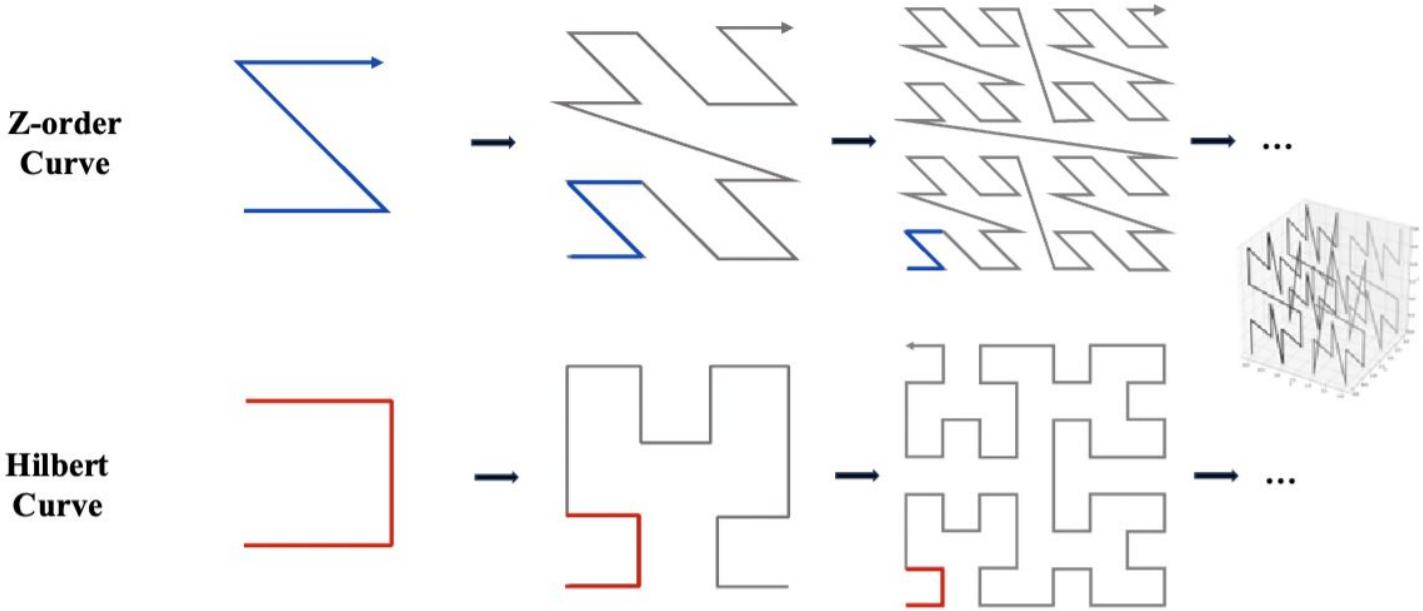


Poor instance reconstruction - interaction



Serialized Attention

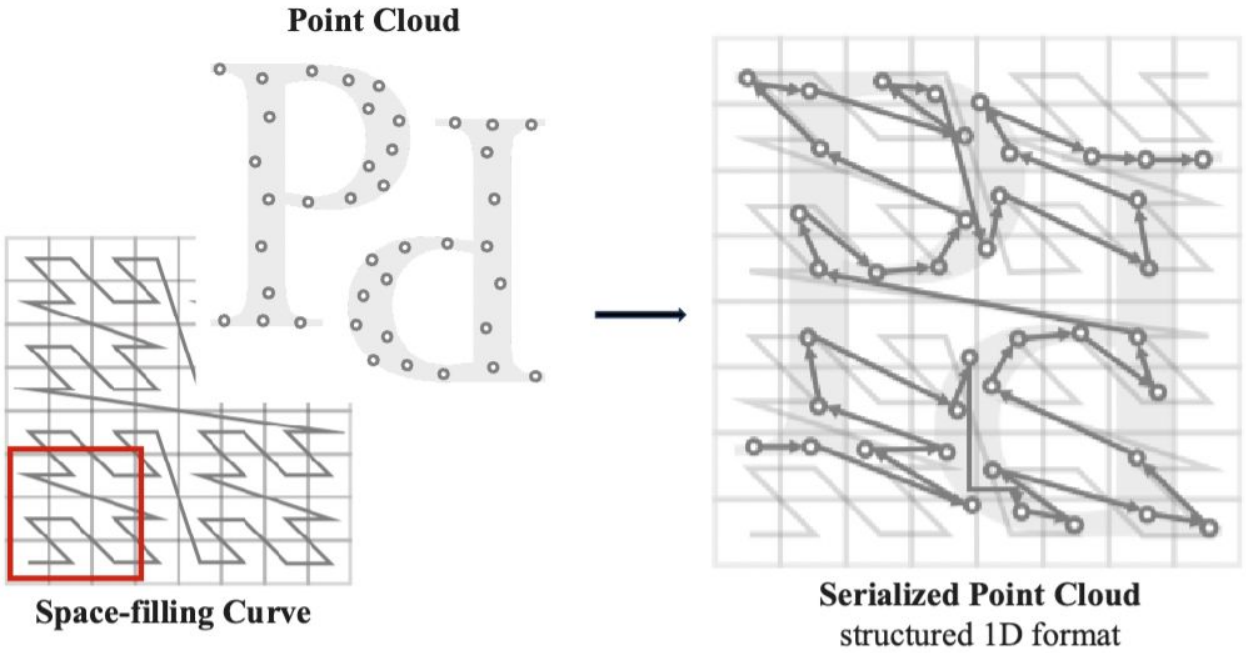
See Point Transformer V3 paper (arXiv:2312.10035) for more detail



From https://xywu.me/talk/20250609_Stanford_Spatial_Representations.pdf

Serialized Attention

See Point Transformer V3 paper (arXiv:2312.10035) for more detail

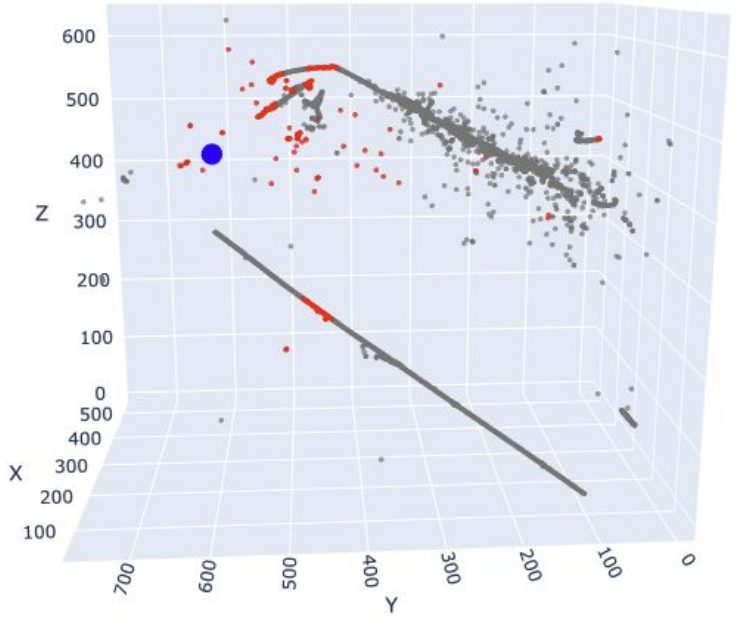


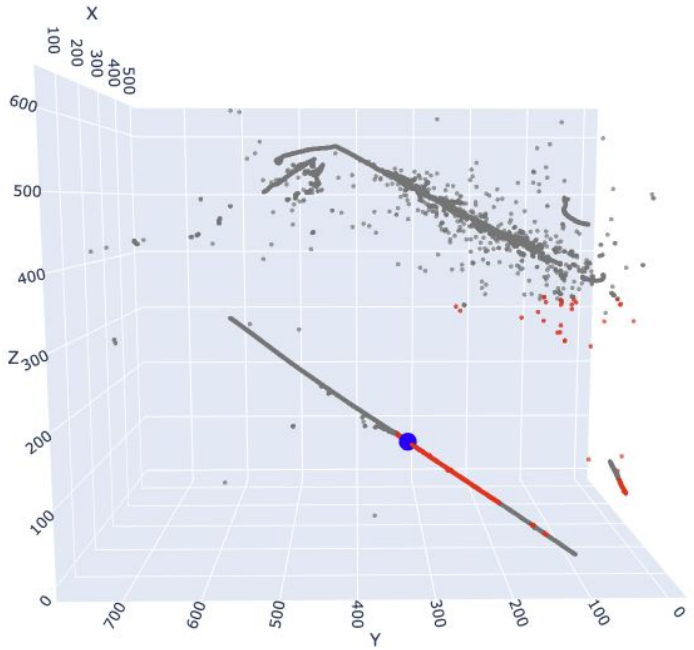
From https://xywu.me/talk/20250609_Stanford_Spatial_Representations.pdf

Serialized Attention – 256 voxel patches

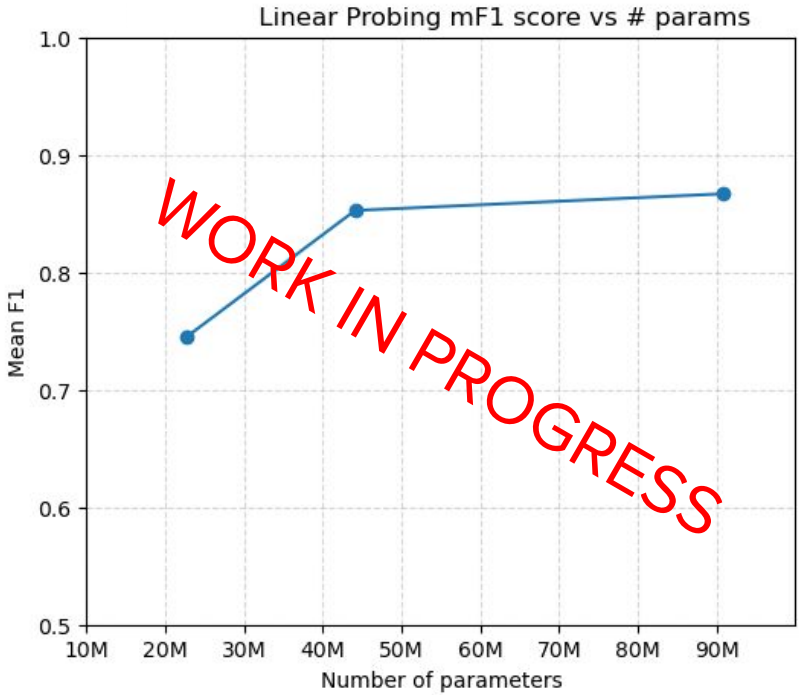
Receptive field at a single point at stage 0

- not perfect, but good enough with enough depth.





Scaling Model Params



Pretraining techniques

- Self-supervised learning
- Supervised learning

INVARIANCE & RECONSTRUCTION

- **Masked reconstruction** 1D · 2D · 3D · 1D+2D+3D
  Recover masked pixels, tokens, or local geometry from partial observations.
  MAE, BEiT, iBOT, Point-BERT, Point-MAE, PoLAr-MAE, Hiera / Point-M2AE, 4M
- **Non-contrastive alignment** 1D · 2D · 3D
  Teacher-student or twin-view agreement, usually without explicit negatives.
  BYOL, SimSiam, DINO + DINOv2, Sonata, Panda
- **Predictive latent modeling** 2D
  Predict semantic latent targets rather than reconstructing raw pixels or prototype distributions
  JEPA / LeJEPA / VICReg, data2vec, BYOL, point2vec
- **Contrastive alignment** 2D · 3D
  Bring two views of the same sample together and push different samples apart.
  SimCLR, MoCo, PointContrast, "Contrastive Learning for Robust Representations of Neutrino Data"

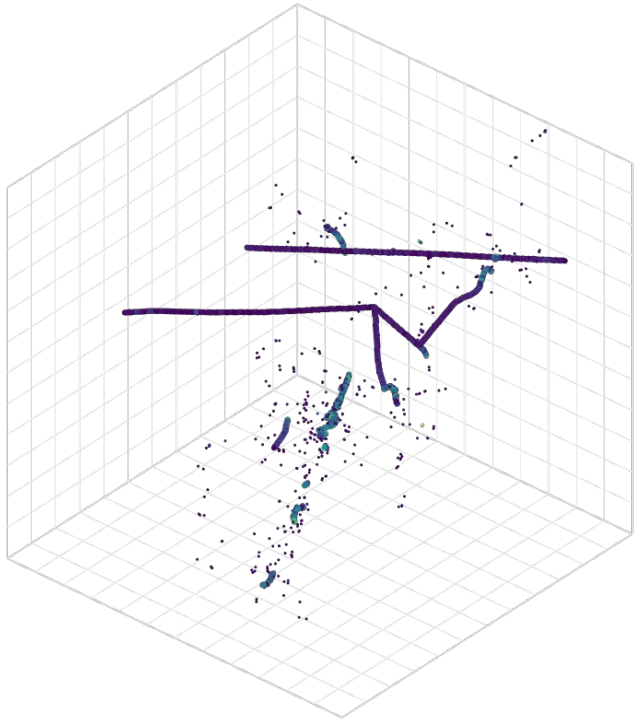
GENERATIVE, GEOMETRIC & CROSS-MODAL

- **Autoregressive token modeling** 2D
  Learn by next-token prediction over image tokens or rasterized sequences.
  PointGPT, PointMamba, FM4NPP, NEPA
- **Generative: denoising / flow** 2D · multimodal
  Learn representations through denoising, diffusion, or flow matching objectives.
  Diffusion-based repr. learning, Self-Flow, VAE, "Score-based Diffusion Models for Generating LArTPC Images"
- **Geometry & multi-view 3D** 2D→3D · 3D
  Use view consistency or scene geometry as the supervisory signal.
  VGGT, Dust3r/Mast3r
- **Cross-modal alignment** any modality (+text)
  Align representations across language, images, and point clouds.
  CLIP, Astro-CLIP, Perception Encoder, SigLIP, Concerto/Utonia
- **Labeled + multi-task supervision** supervised
  General-purpose encoders trained on broad label spaces, segmentation masks, or task mixtures.
  BiT / JFT, SAM, task mixtures, OmniLearn, L-GAtr

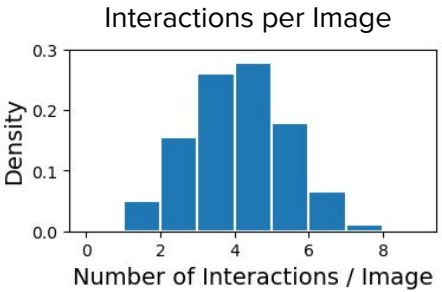
Dataset: PILArNet-Medium

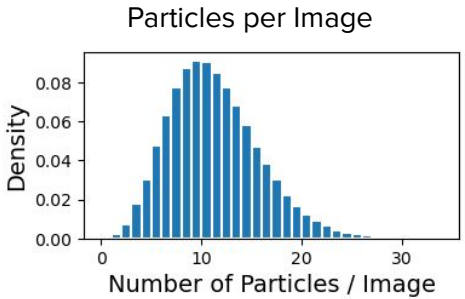
Open data!

- Simulated dataset of 1.2M 3D events
- $(2.3\ \text{m})^3$ cube $(768\ \text{px})^3$. ~5B non-zero voxels.
- +1M events on top of previous open dataset, PILArNet (2020).
- Simply 3D energy depositions, equivalent to “digital hits” from a LArTPC (e.g., DUNE Near Detector)
- 1024 - 30,000 voxels/event



Raw Depositions

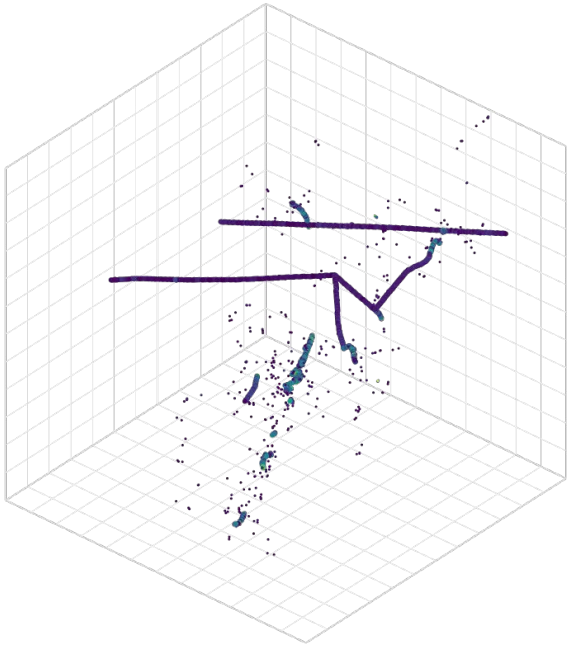




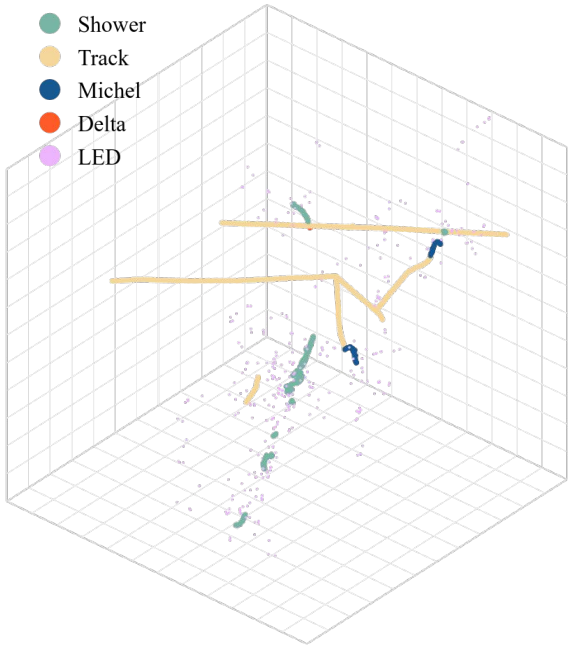
Young, S. et. al, *Particle Trajectory Representation Learning with Masked Point Modeling* (2025)

Semantic Segmentation Labels

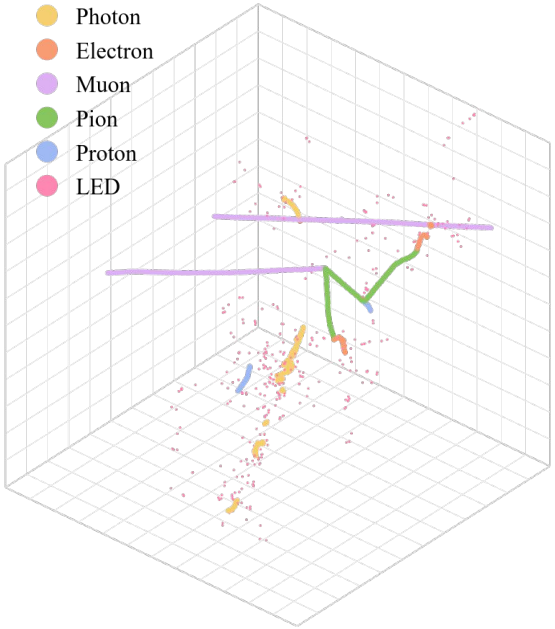
Open data!



Raw Depositions



Semantics Reconstruction

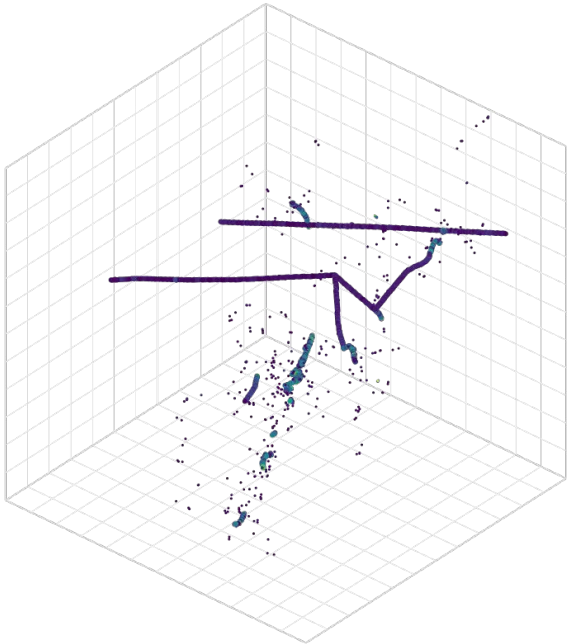


Particle ID Reconstruction

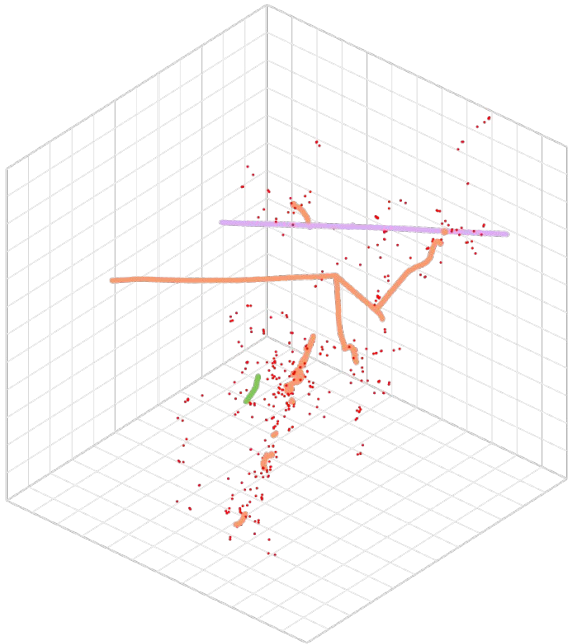
Young, S. et. al, *Particle Trajectory Representation Learning with Masked Point Modeling* (2025)

Instance Labels

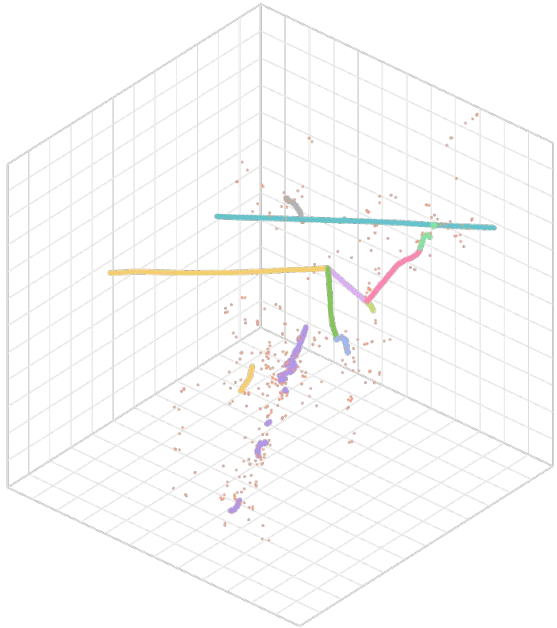
Open data!



Raw Depositions

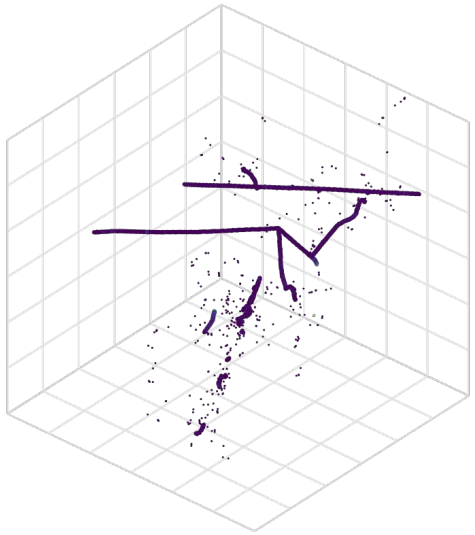


Interaction-level



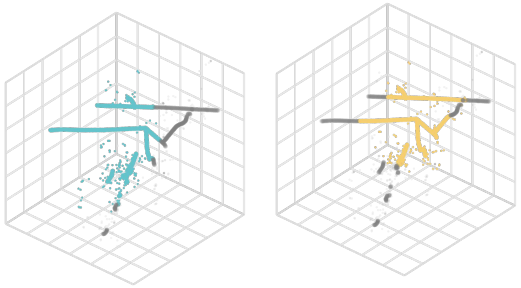
Particle-level

Young, S. et. al, *Particle Trajectory Representation Learning with Masked Point Modeling* (2025)

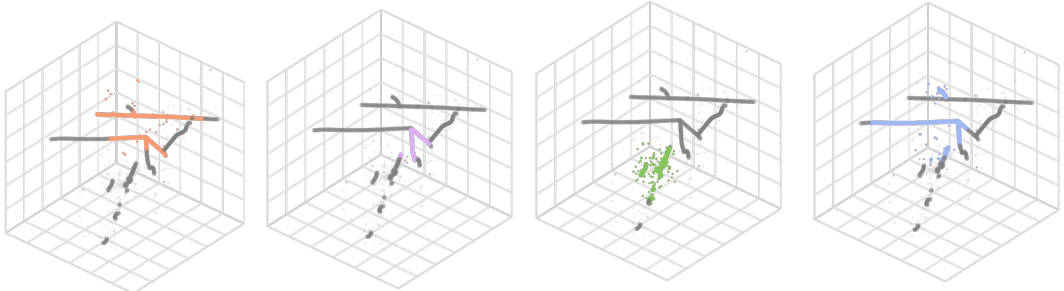


Full Image

Global Views



Local Views



Masked Views