

# Scalable Deep Convolutional Neural Networks for Sparse, Locally Dense Liquid Argon Time Projection Chamber Data

Laura Domine (Stanford / SLAC)  
Kazuhiro Terao (SLAC)

[arXiv:1903.05663](https://arxiv.org/abs/1903.05663)

ACAT 2019



# Outline

1. LArTPC data & semantic segmentation
2. Submanifold sparse convolutions
3. Benchmark: Sparse vs dense U-ResNet

# Outline

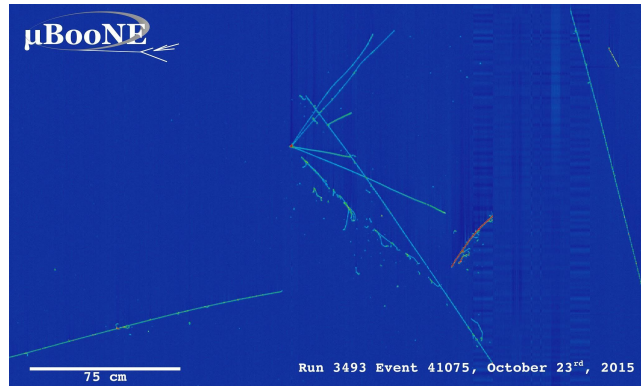
1. LArTPC data & semantic segmentation
2. Submanifold sparse convolutions
3. Benchmark: Sparse vs dense U-ResNet

# Particle Image Analysis with LArTPCs

Liquid Argon Time Projection Chamber (**LArTPC**) = particle imaging detector

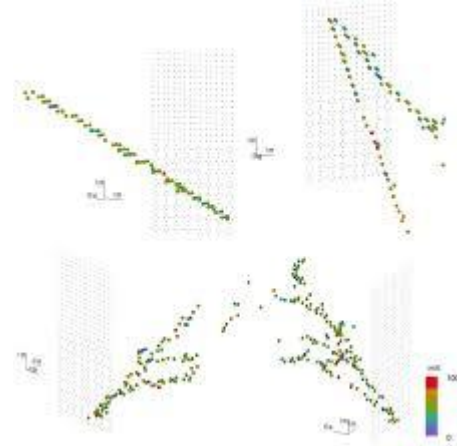
~3mm resolution

**Wire LArTPC (2D projections)**



Neutrino interaction candidate from MicroBooNE experiment @ Fermilab

**Pixel LArTPC (native 3D)**

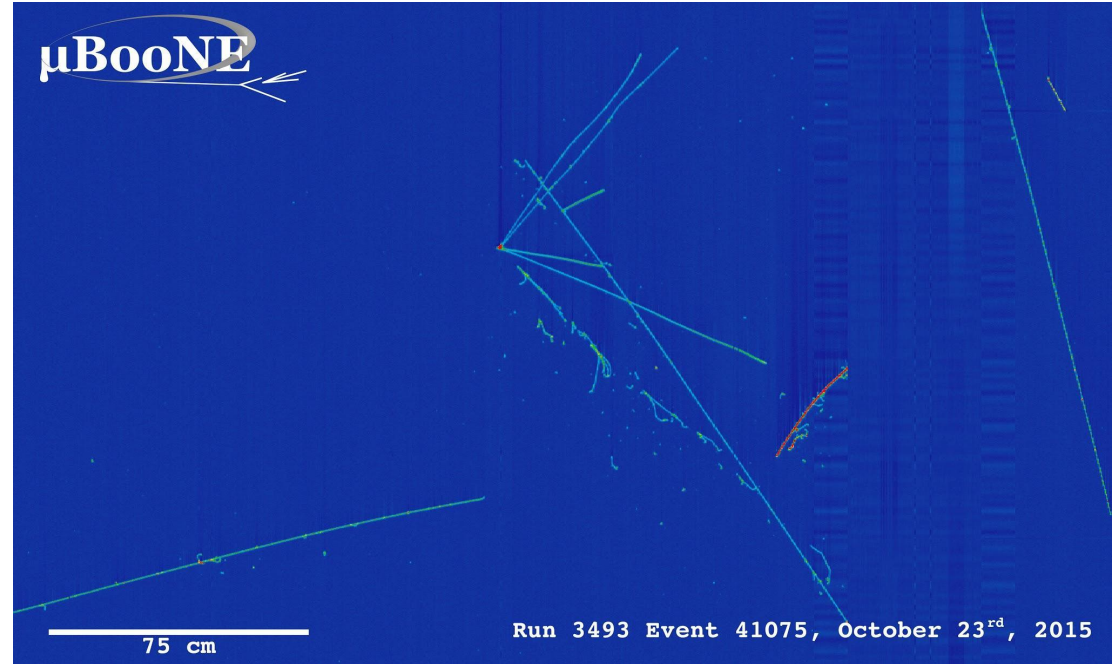
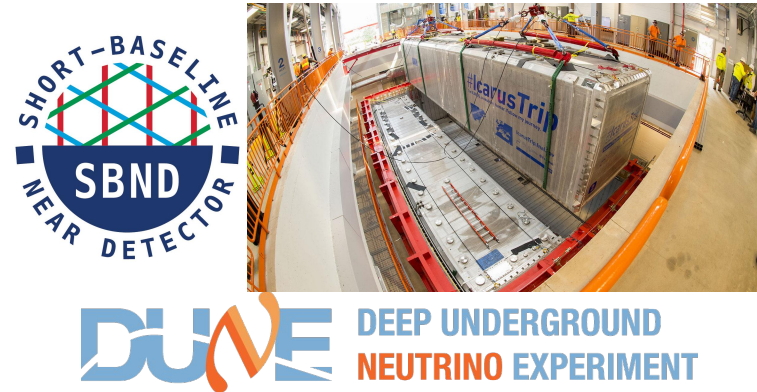


Cosmic rays in a 3D LArTPC charge readout (arxiv:1808.02969) @ LBNL

# Particle Image Analysis with LArTPCs for **neutrinos**

Neutrino detectors & LArTPCs

Goal: Extract  $\nu$  flavor + energy

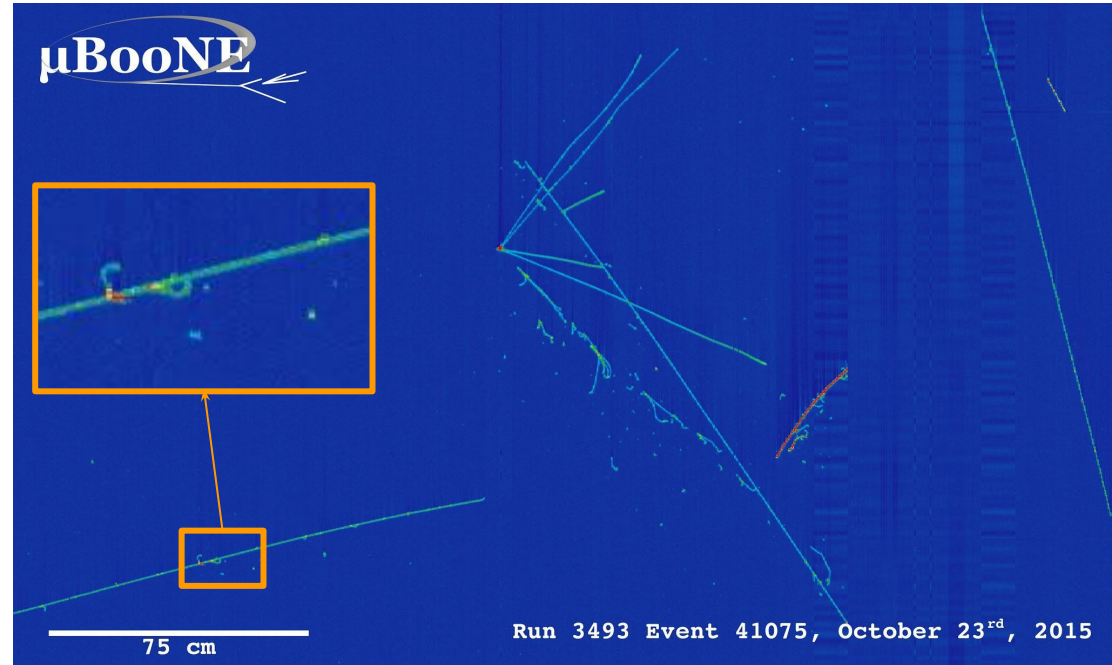
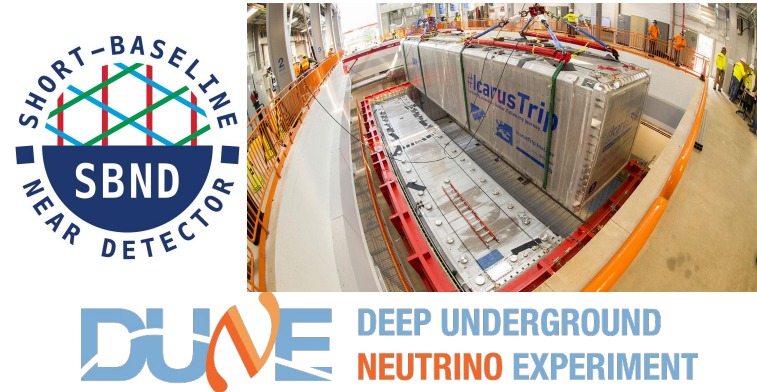




# Particle Image Analysis with LArTPCs for neutrinos

Neutrino detectors & LArTPCs

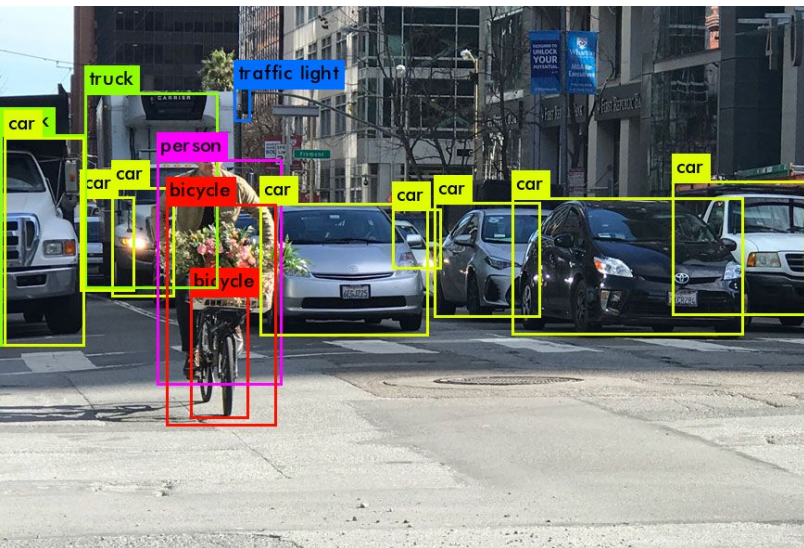
Goal: Extract  $\nu$  flavor + energy



# Convolutional Neural Networks

Now state-of-the-art technique in computer vision for complex image analysis tasks:

Object detection & classification



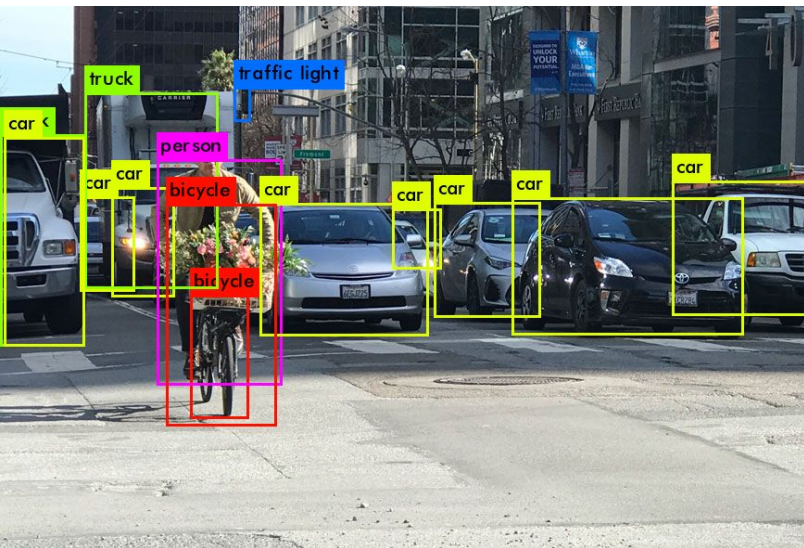
Semantic segmentation



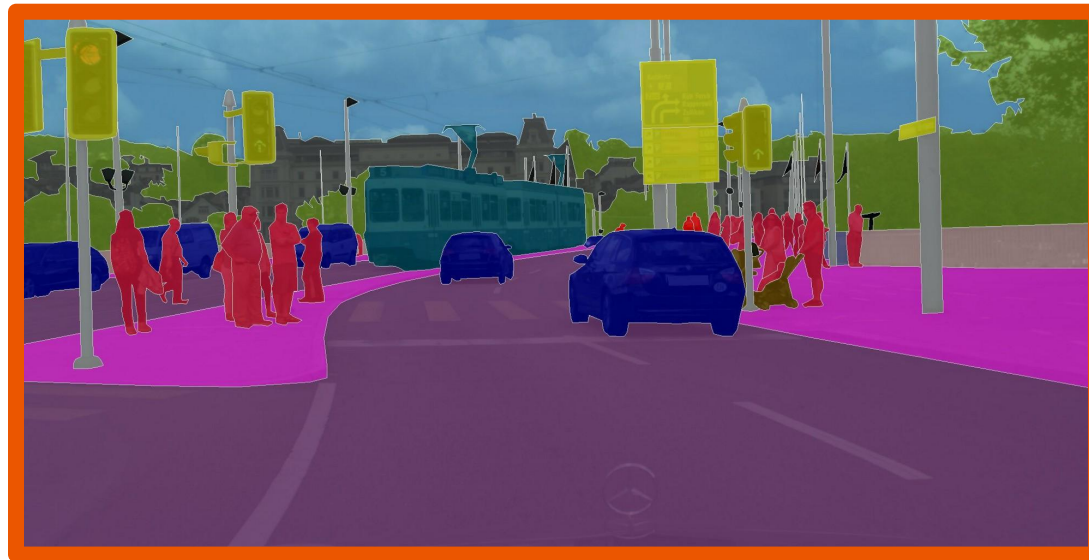
# Convolutional Neural Networks

Now state-of-the-art technique in computer vision for complex image analysis tasks:

## Object detection & classification



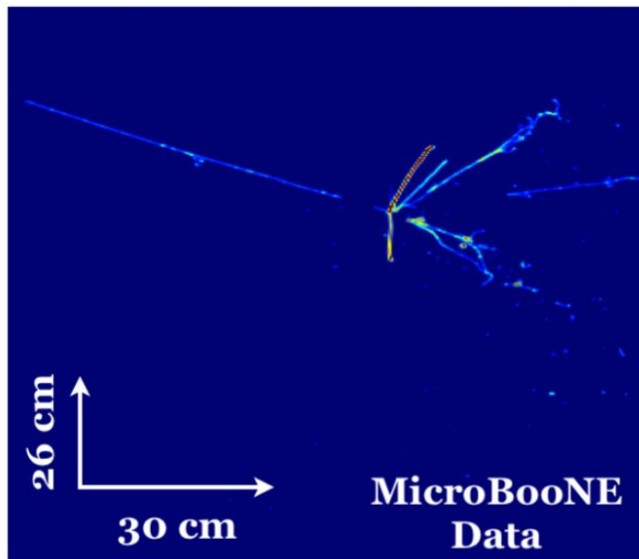
## Semantic segmentation



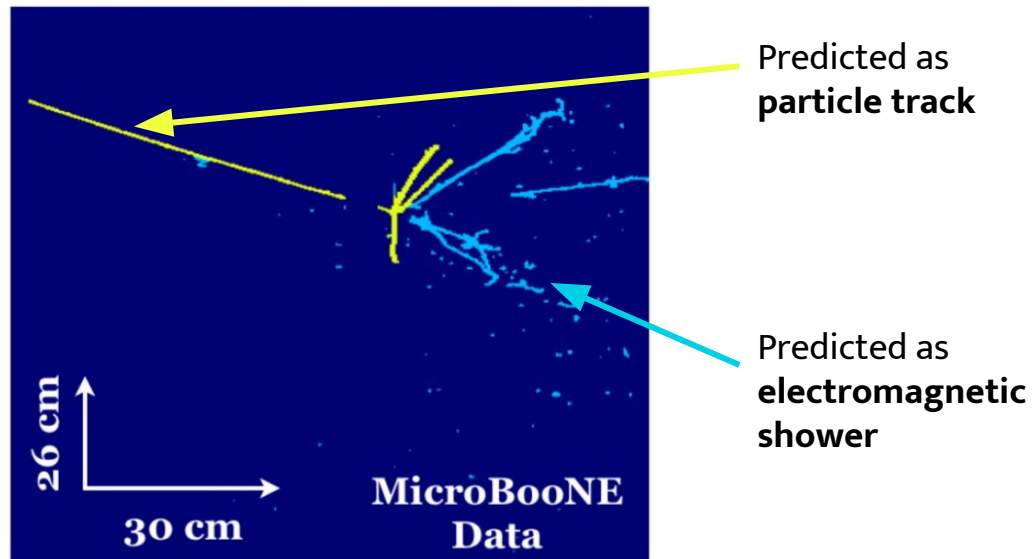


# Semantic segmentation of LArTPC data (2D)

Data (network's input)

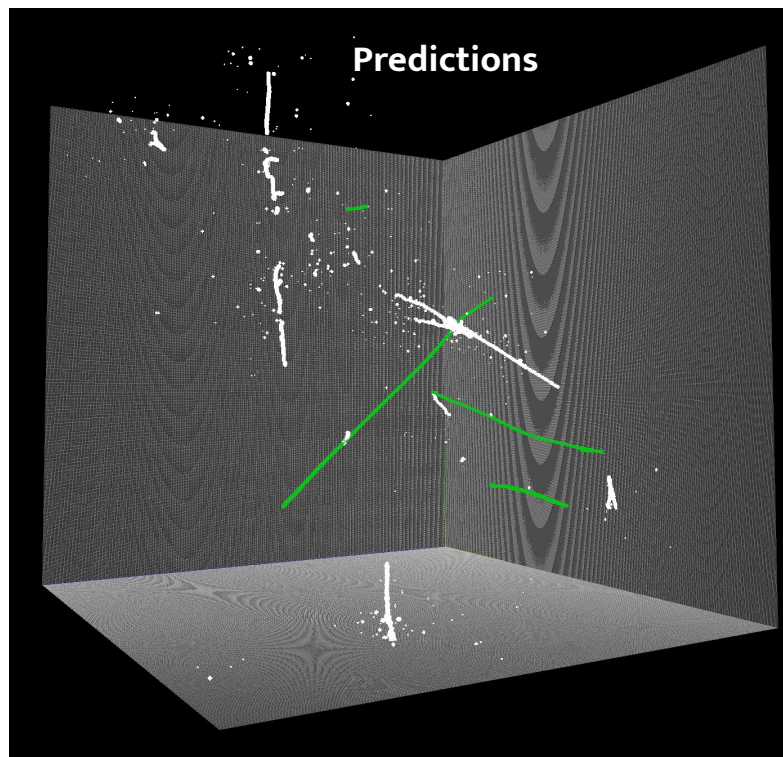
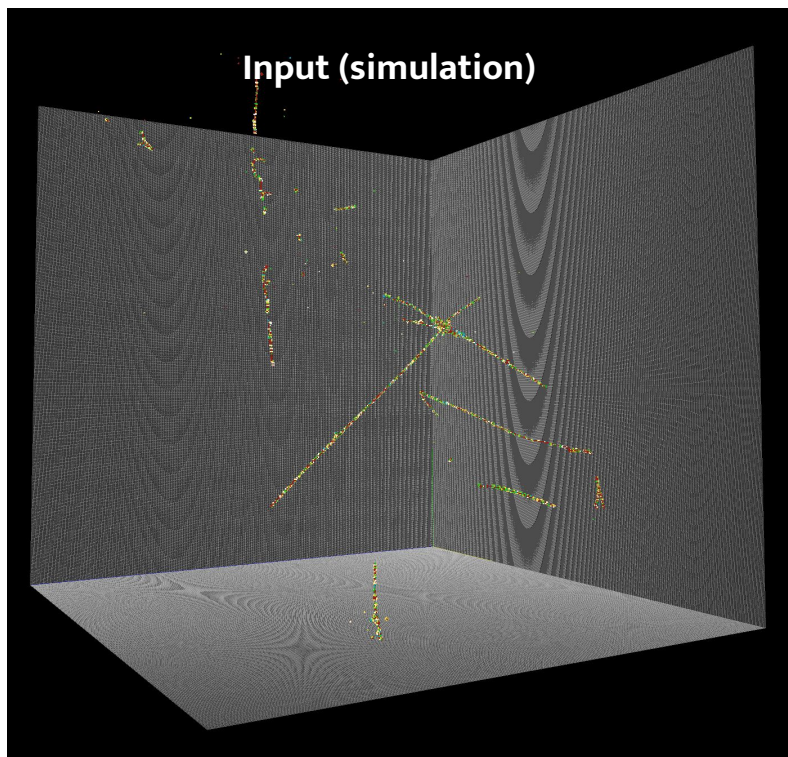


Predictions (network's output)



A Deep Neural Network for Pixel-Level Electromagnetic Particle Identification in the MicroBooNE Liquid Argon Time Projection Chamber. ([arxiv:1808.07269](https://arxiv.org/abs/1808.07269))

# Semantic segmentation of LArTPC data (**3D**)



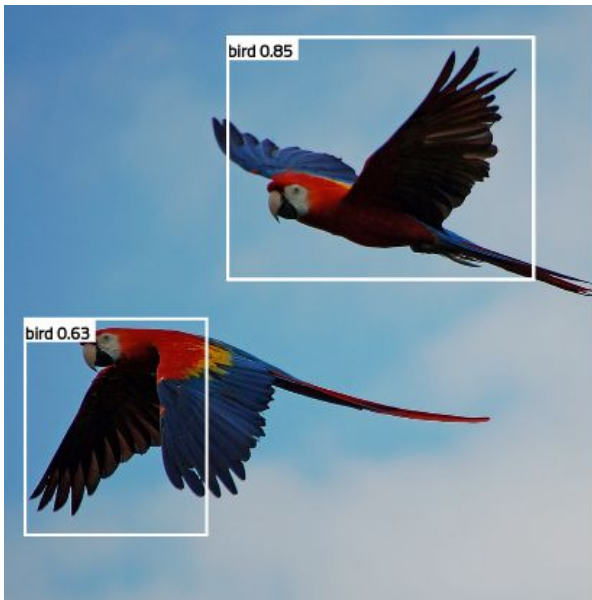
*2-classes (particle track vs electromagnetic shower) pixel-level **segmentation** on 512px 3D images.*

# Outline

1. LArTPC data & semantic segmentation
2. Submanifold sparse convolutions
3. Benchmark: Sparse vs dense UResNet

# LArTPC data is sparse, locally dense

Dense



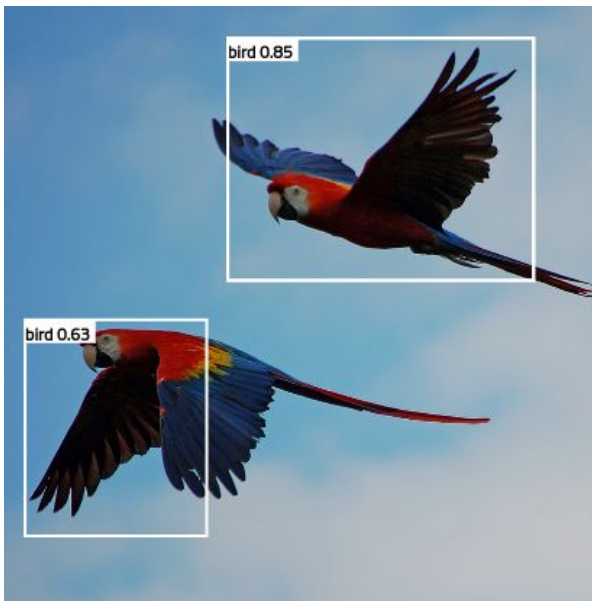
CNNs rely on  
dense matrix  
multiplications

All pixels are  
meaningful for  
CNNs.

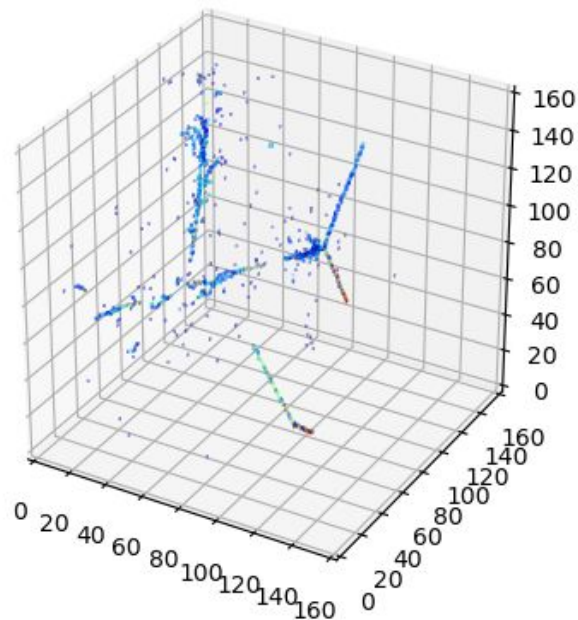


# LArTPC data is sparse, locally dense

Dense



Sparse (but locally dense)



<1% of voxels are non-zero in LArTPC data

Zero voxels are meaningless!

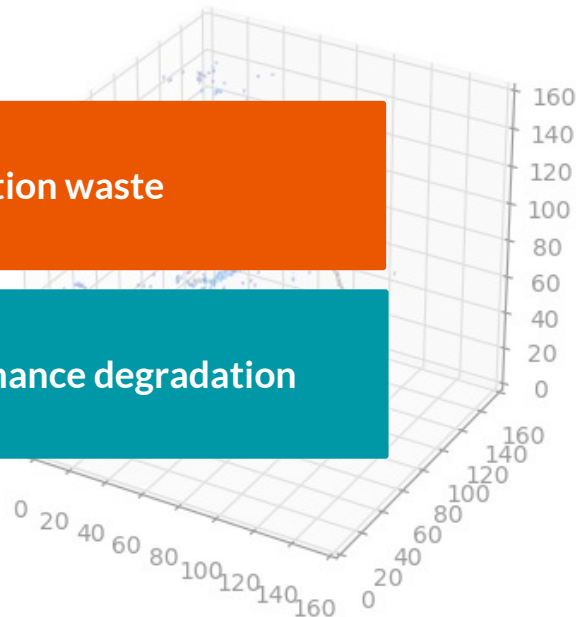
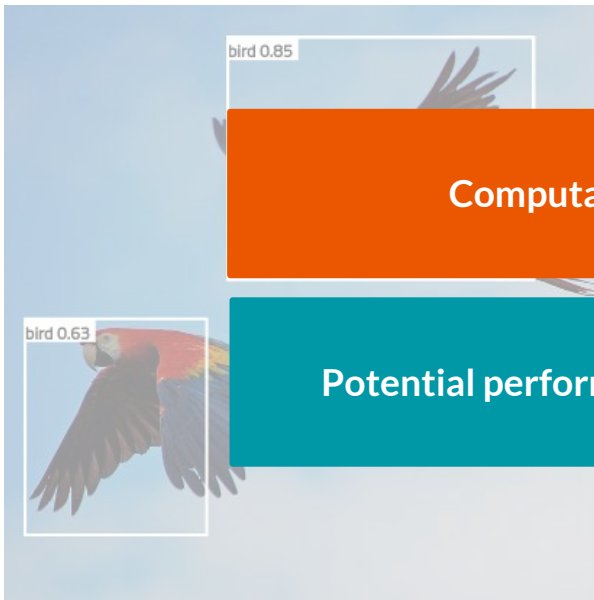
CNNs rely on dense matrix multiplications

All pixels are meaningful for CNNs.

# LArTPC data is sparse, locally dense

Dense

Sparse (but locally dense)



Computation waste

Potential performance degradation

<1% of voxels are non-zero in LArTPC data

Zero voxels are meaningless!

CNNs rely on dense matrix multiplications

All pixels are meaningful for CNNs.

# Submanifold Sparse Convolutions

Many possible definitions and implementations of *'sparse convolutions'*...

**Submanifold Sparse Convolutions** ([arxiv:1711.10275](https://arxiv.org/abs/1711.10275), CVPR2018):

<https://github.com/facebookresearch/SparseConvNet>

**State-of-the-art** on ShapeNet challenge (3D part segmentation)

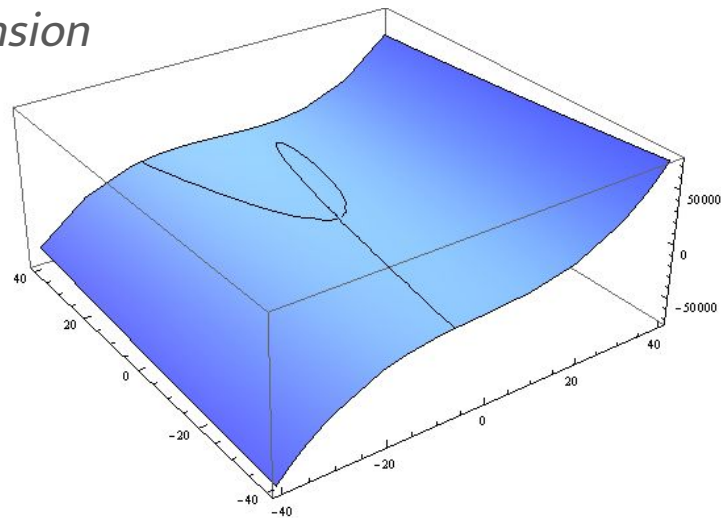
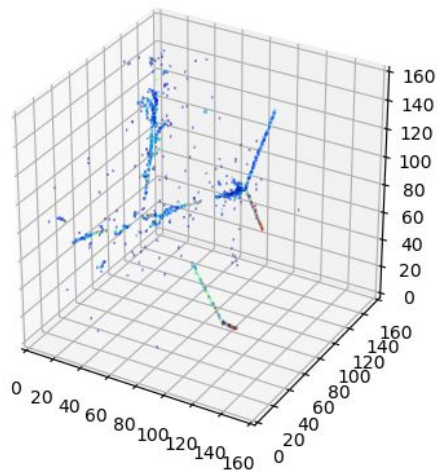


# Submanifold Sparse Convolutions

**Submanifold** = “input data with lower effective dimension than the space in which it lives”

Ex: 1D curve in 2+D space, 2D surface in 3+D space

Our case: the worst! **1D curve in 3D space...**





# Submanifold Sparse Convolutions *aim to solve...*

1. **Resources waste** of dense convolutions on sparse data
2. **Dilation problem**
  - 1 nonzero site leads to  $3^d$  nonzero sites after 1 convolution
  - How to keep the same level of sparsity throughout the network?



[3D Semantic Segmentation with Submanifold Sparse Convolutional Networks](#)  
(arxiv: 1711.10275)

# Outline

1. LArTPC data & semantic segmentation
2. Submanifold sparse convolutions
3. **Benchmark: Sparse vs dense UResNet**
  - a. **Dataset, task, metrics & network architecture**
  - b. Results

# Dataset & Task

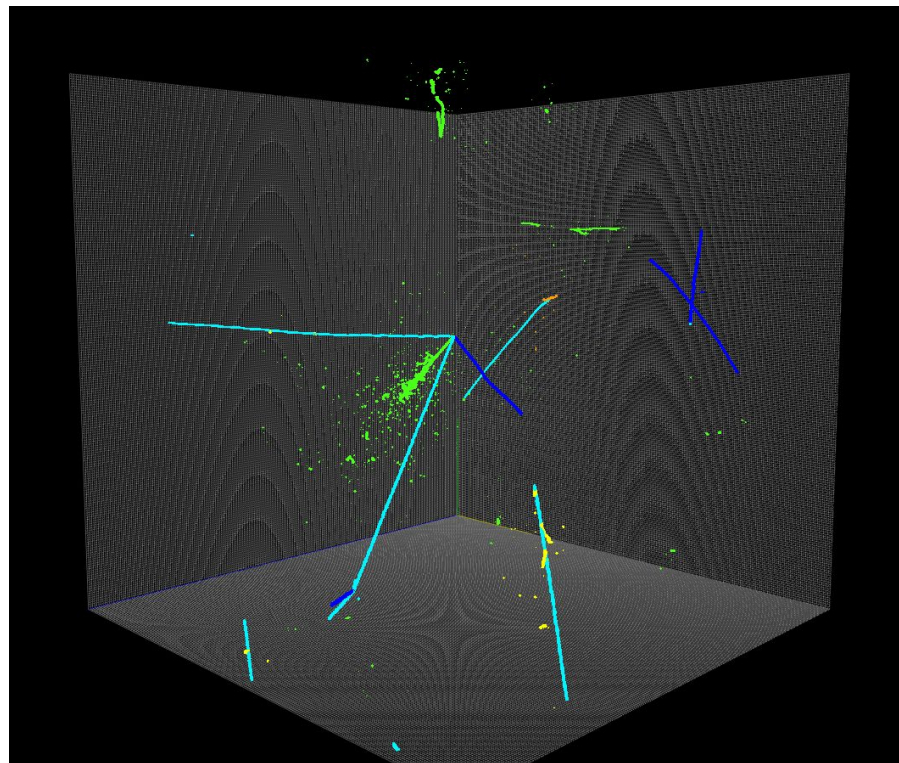
Total: 100,000 simulated 3D images

Spatial size: 192px / 512px / 768px (~3mm/pix)

## Semantic segmentation with **5 classes**

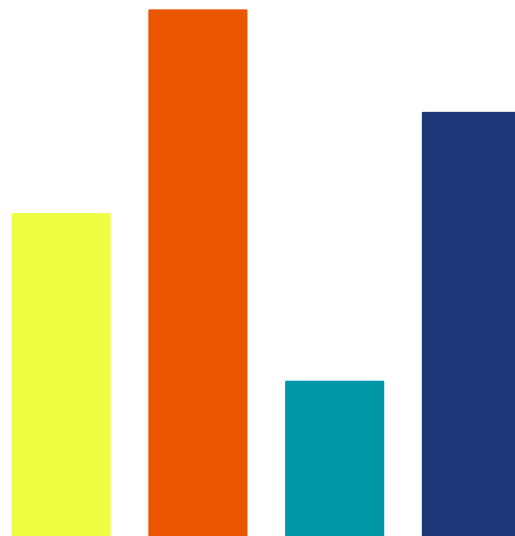
- Heavily ionizing particles (protons)
- Minimum ionizing particles (muons and pions)
- Electromagnetic shower
- Delta rays
- Michel electrons

Publicly available: <https://dx.doi.org/10.17605/OSF.IO/VRUZP>



# Metrics

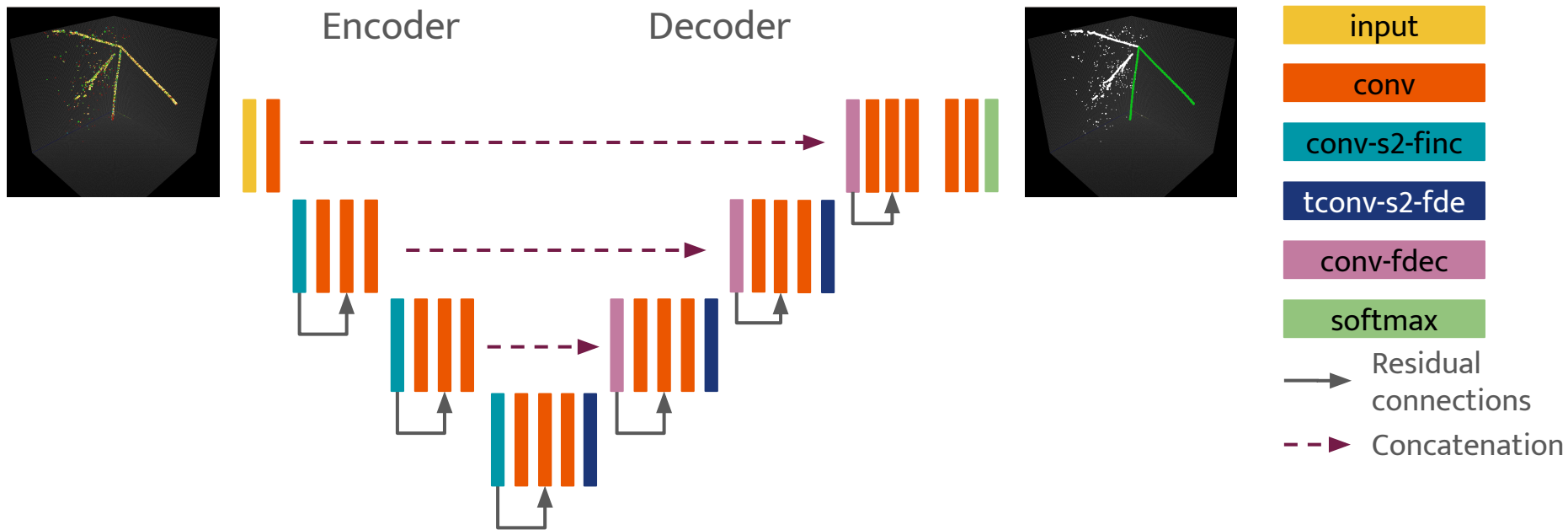
- **Non-zero accuracy:** fraction of correctly labeled pixels, i.e.  
 $\frac{\# \text{ nonzero voxels whose predicted label is correct}}{\# \text{ nonzero voxels}}$
- GPU memory (hardware limitation)
- Computation wall-time





# Network architecture: U-ResNet

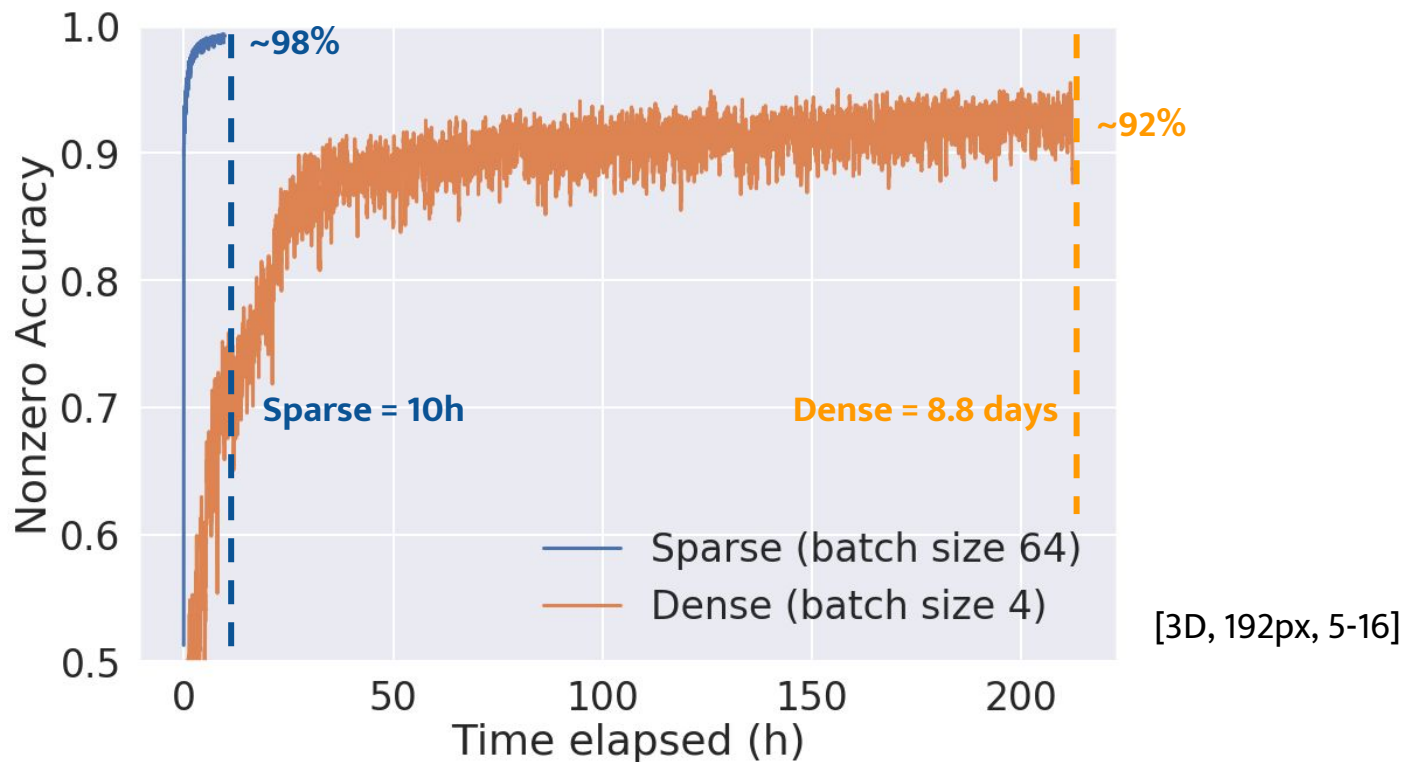
U-ResNet = U-Net + ResNet



# Outline

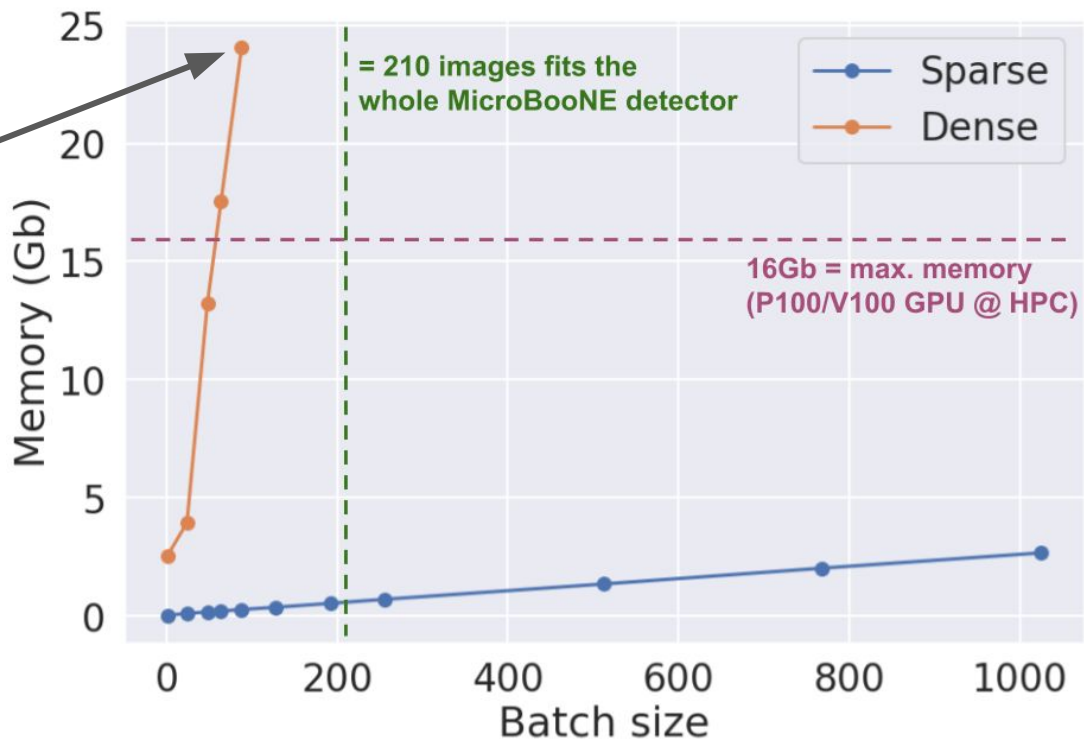
1. LArTPC data & semantic segmentation
2. Submanifold sparse convolutions
3. **Benchmark: Sparse vs dense UResNet**
  - a. Dataset, task, metrics & network architecture
  - b. Results**

# Sparse U-ResNet trains 119x faster

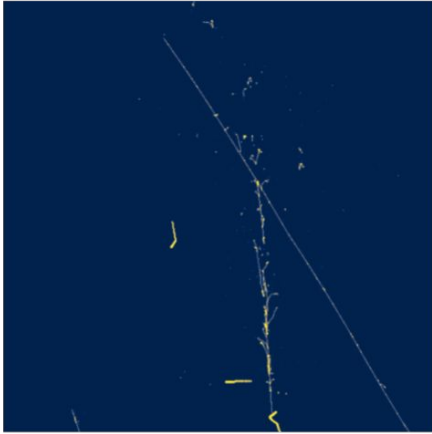


# Sparse U-ResNet allows larger batch sizes

@batch size 88  
sparse uses  
**93x less memory**  
than dense and  
computation is  
**3x faster**

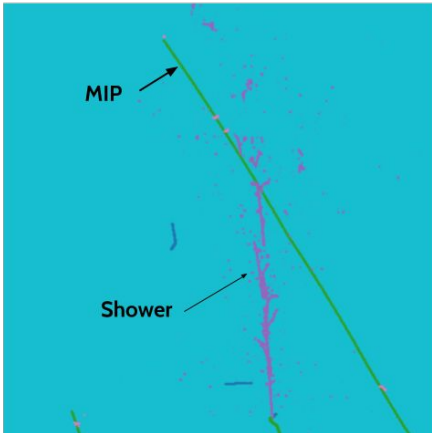


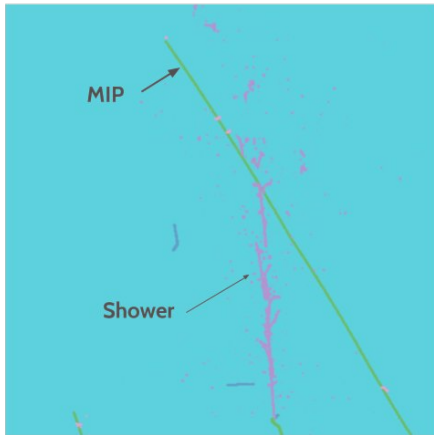
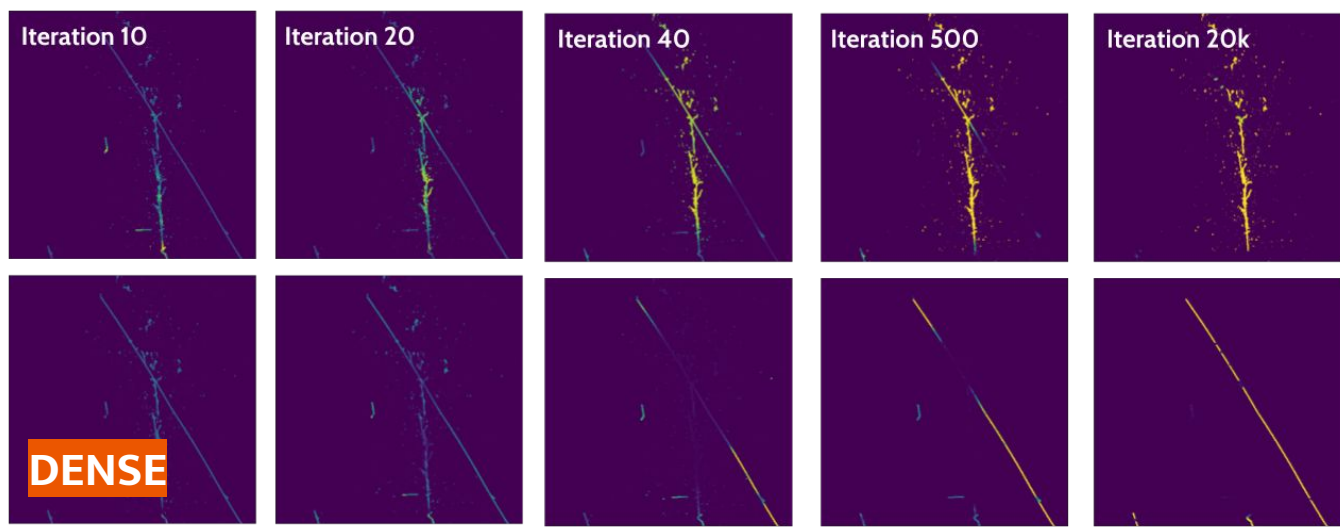
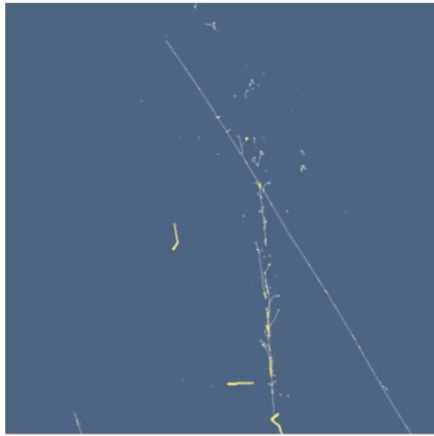
At train time  
[2D, 512px, 5-16]



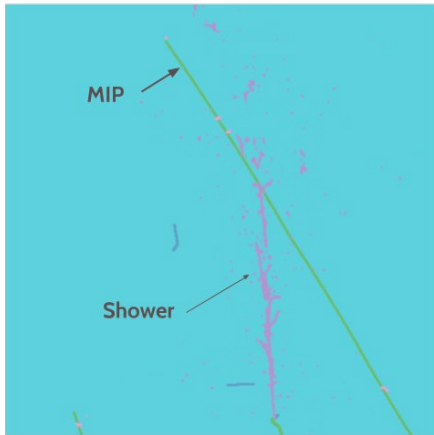
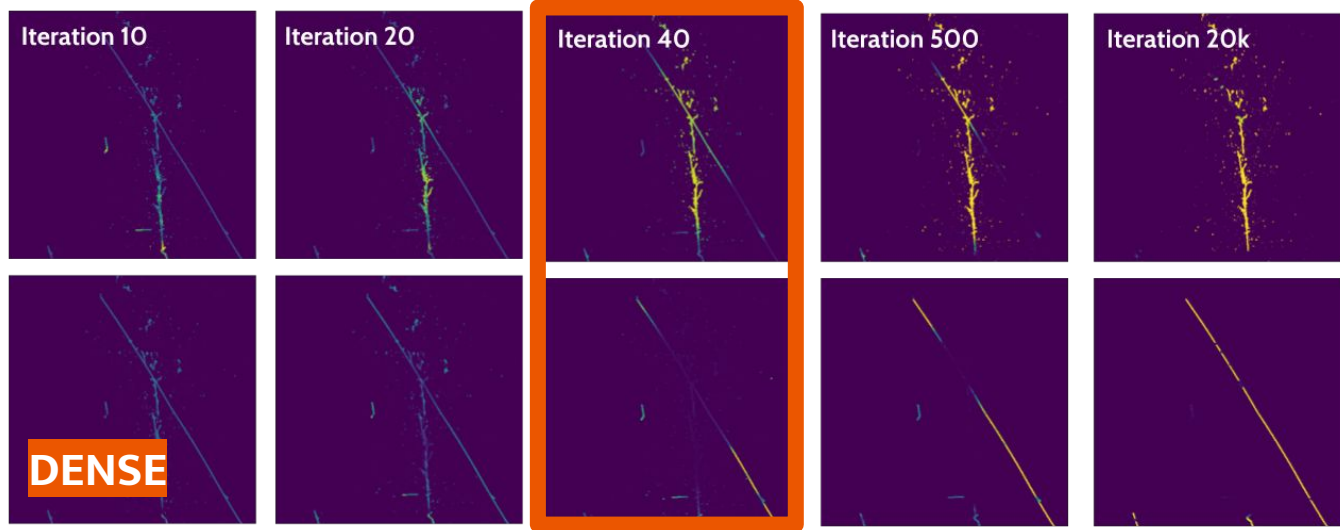
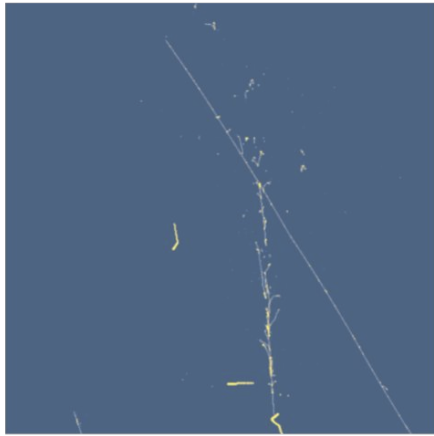
# Learning strategies

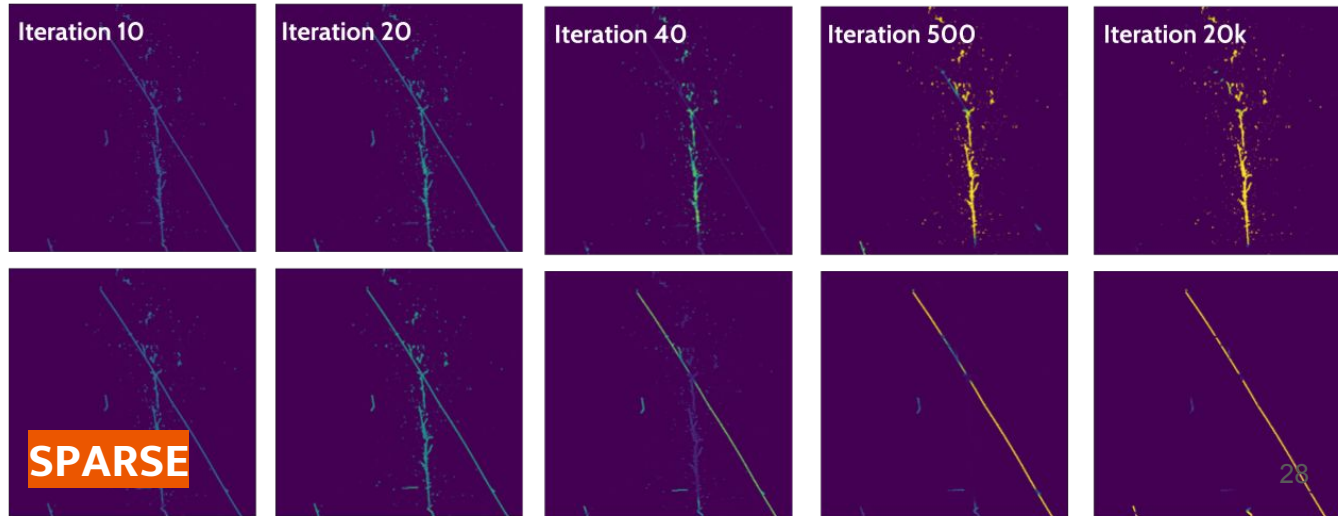
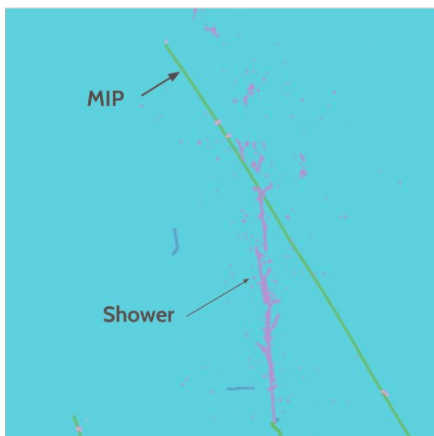
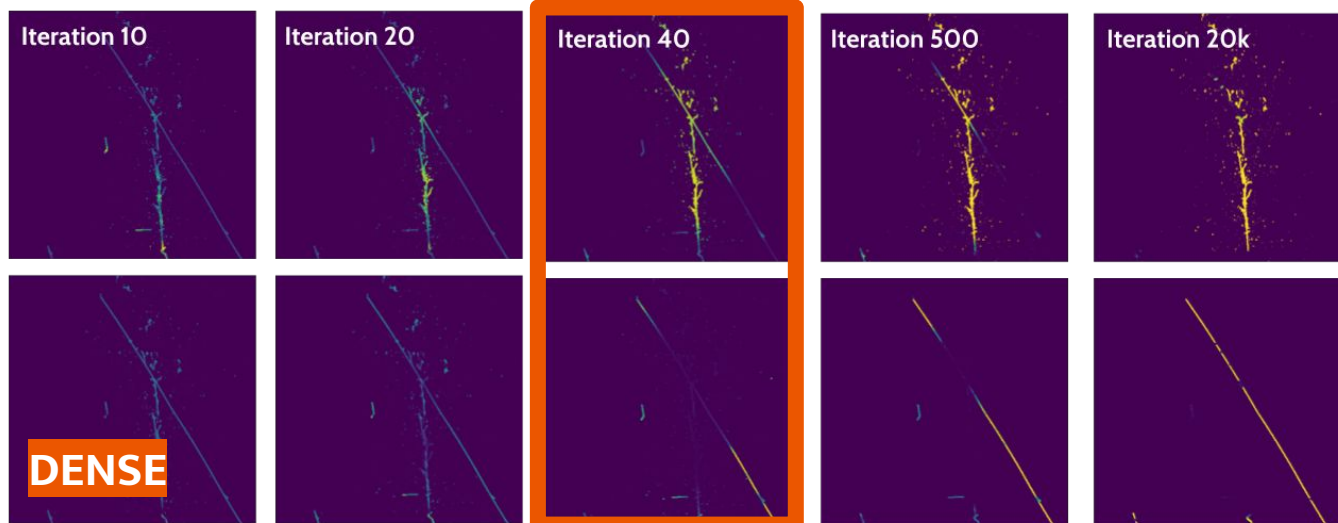
Sparse U-ResNet learns more uniformly across pixels than dense U-ResNet

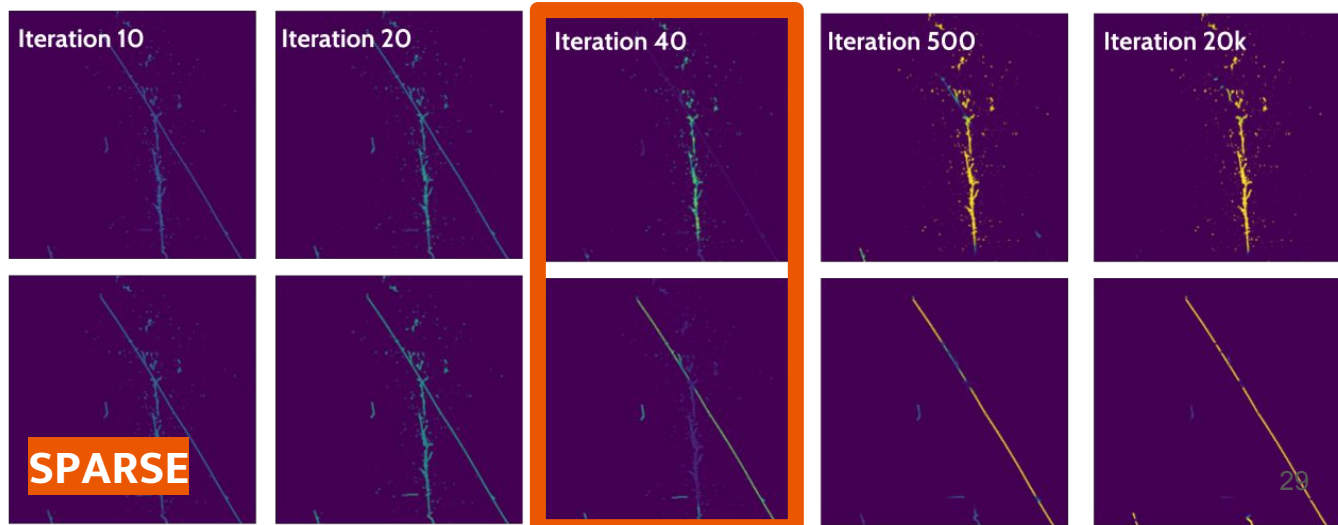
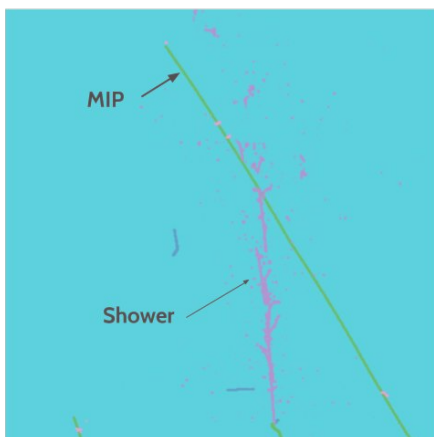
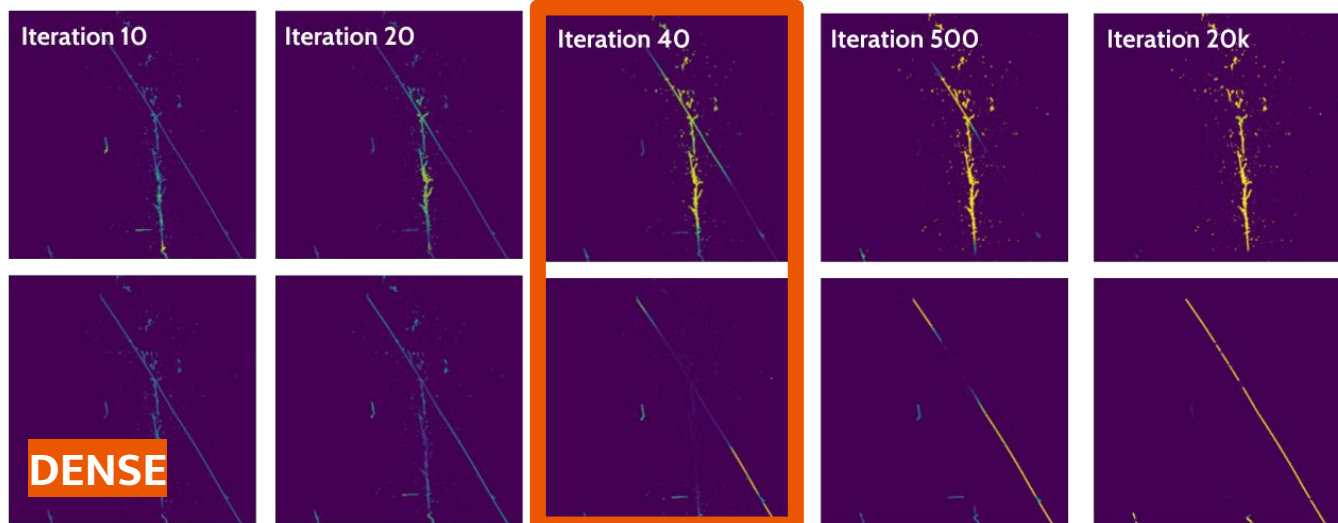












# Summary

Submanifold sparse convolutions...

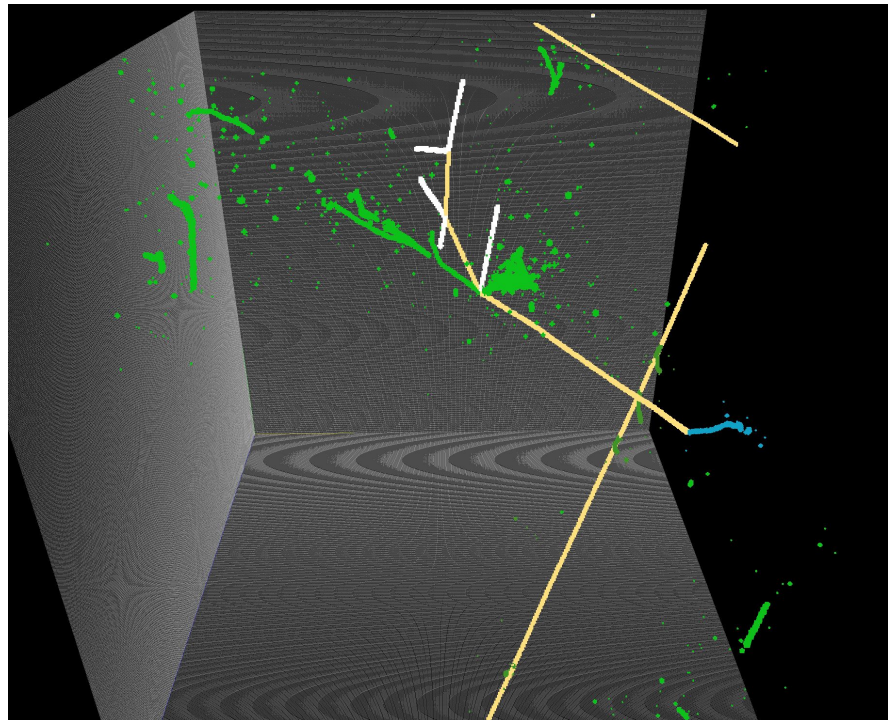
- Run faster
- Use less GPU memory
- ... and outperform standard convolutions.

**Better performance and better scalability!**

<https://arxiv.org/abs/1903.05663>

Reproduce our results / start using SSCN:

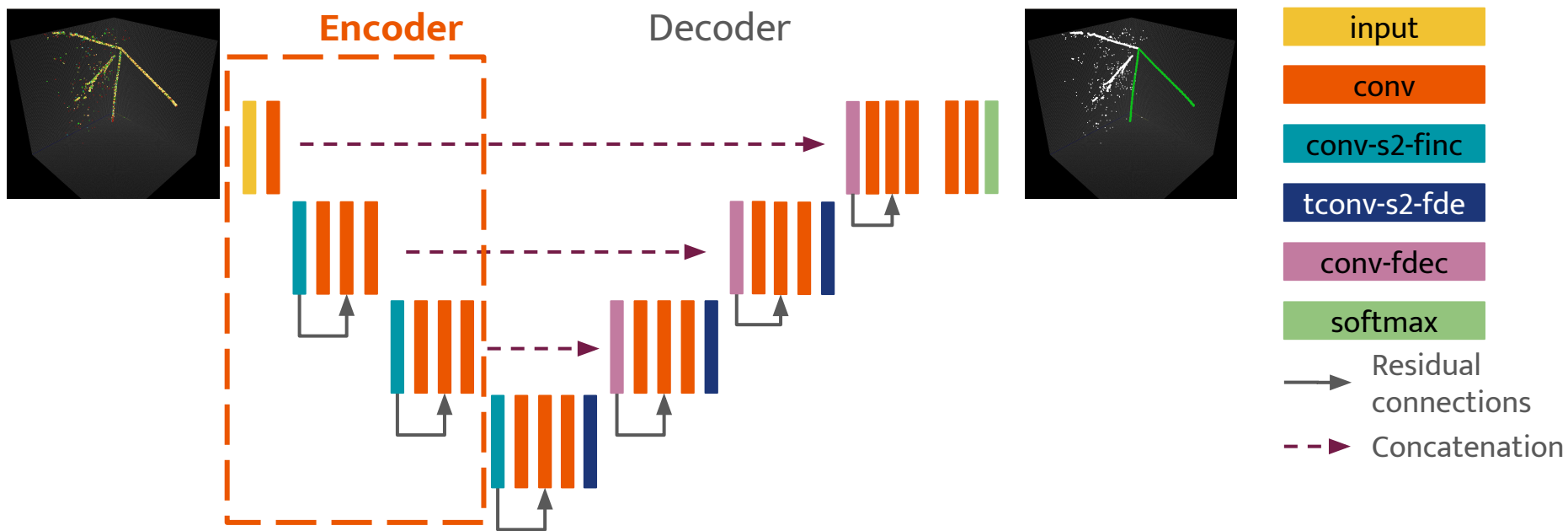
- Open dataset: <https://osf.io/vruzp/>
- Software containers: <https://www.singularity-hub.org/containers/6596>
- Code on Github: [https://github.com/Temigo/uresnet\\_pytorch](https://github.com/Temigo/uresnet_pytorch)



# Backup

# Network architecture: UResNet

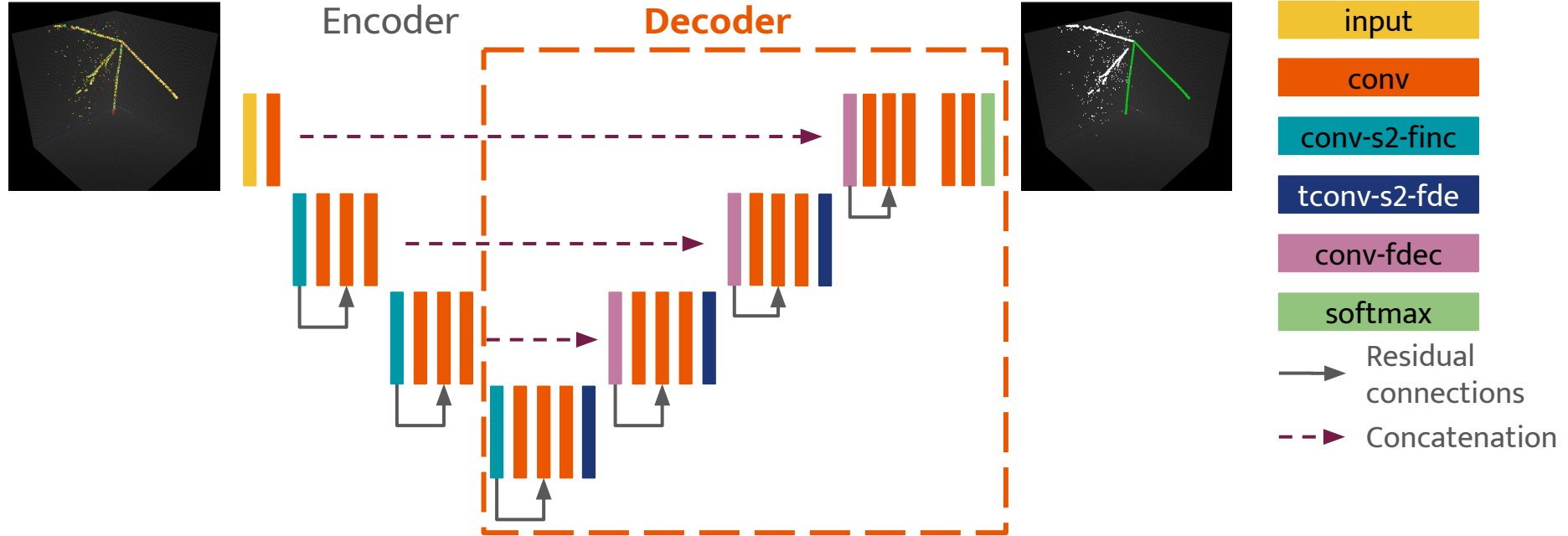
UResNet = U-Net + ResNet





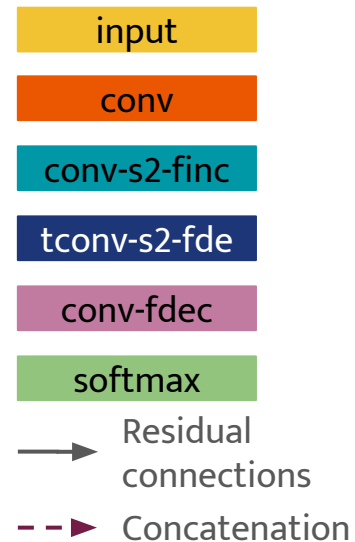
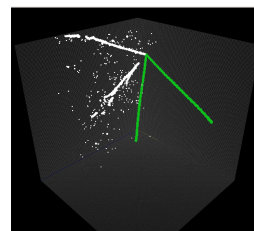
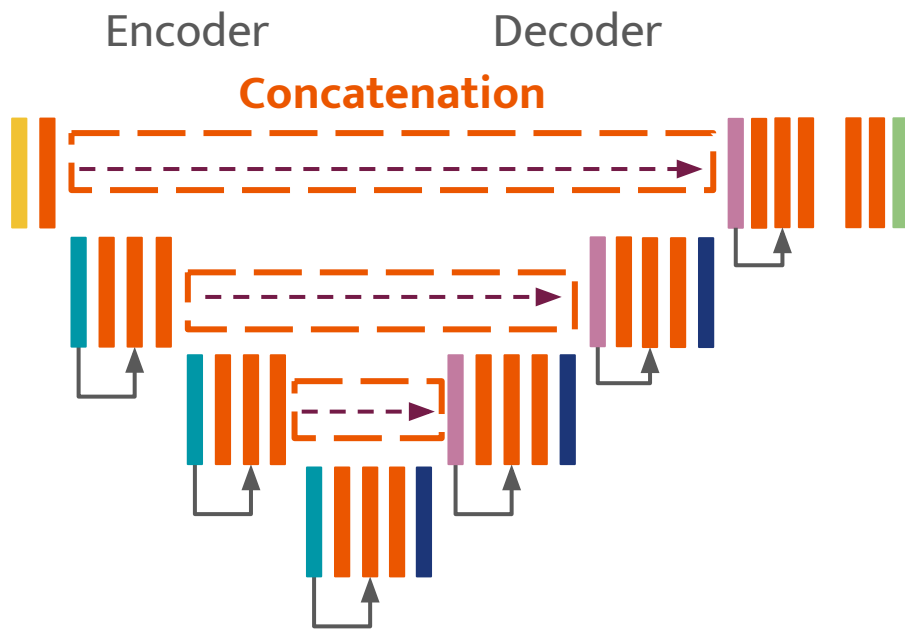
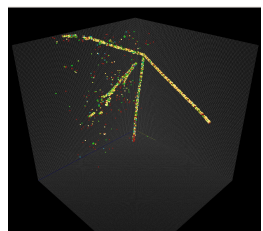
# Network architecture: UResNet

UResNet = U-Net + ResNet



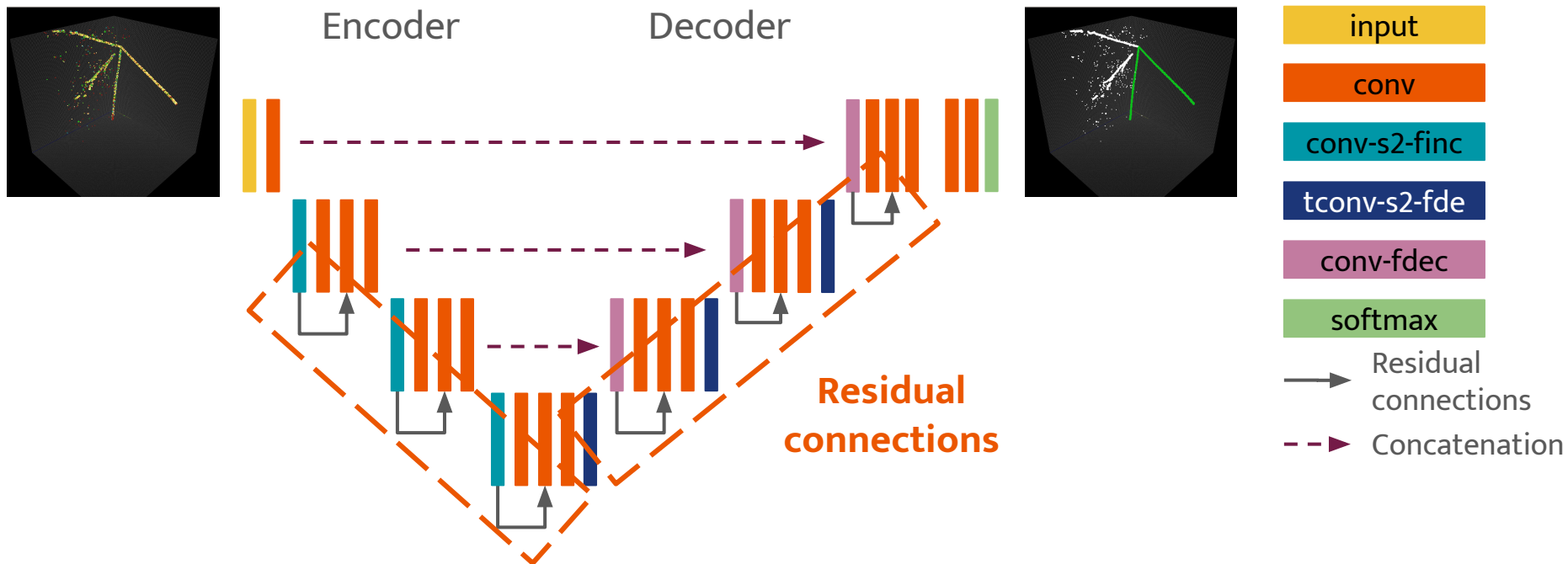
# Network architecture: UResNet

UResNet = U-Net + ResNet

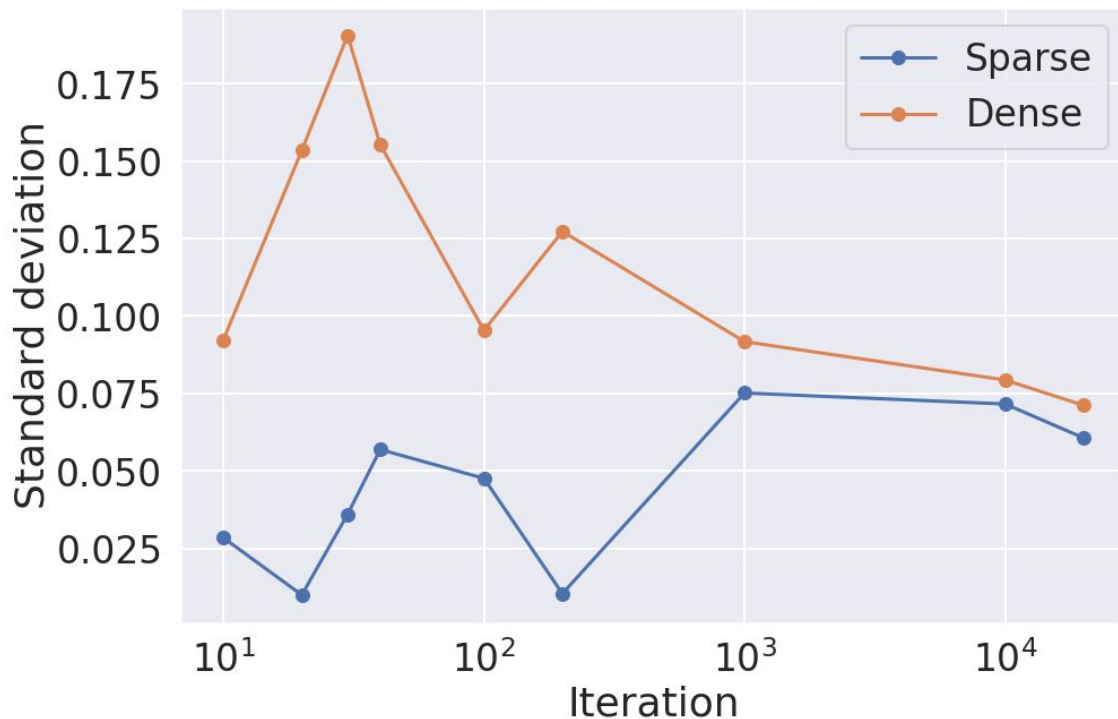


# Network architecture: UResNet

UResNet = U-Net + ResNet

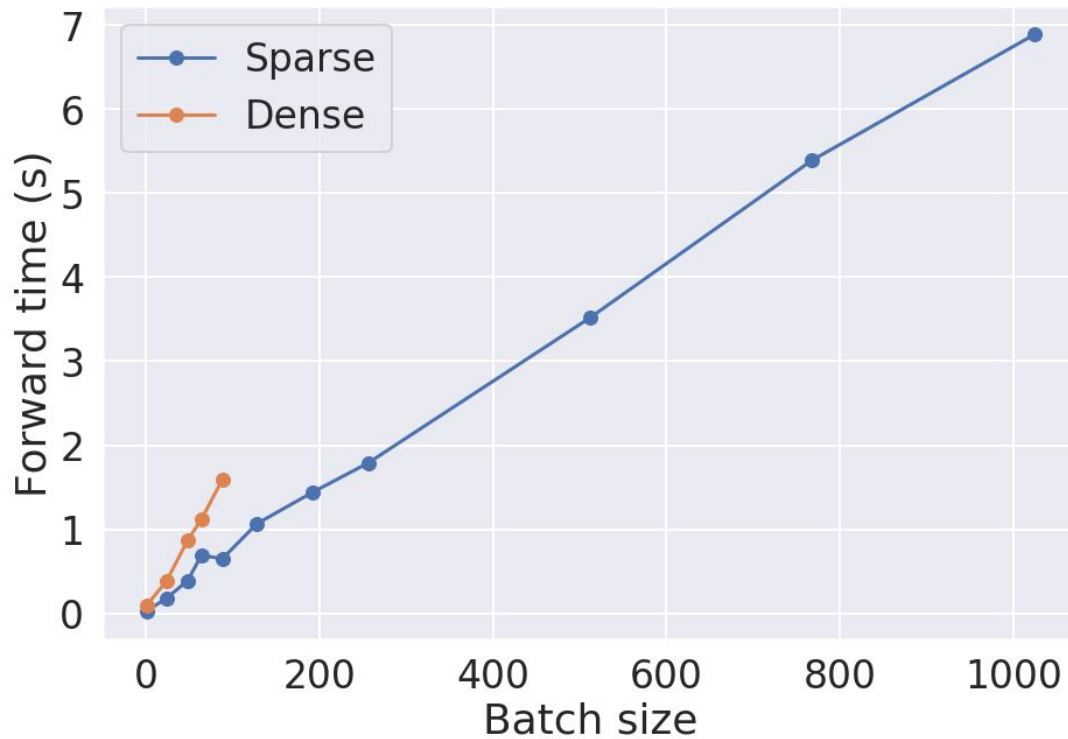


# Sparse U-ResNet learns more uniformly

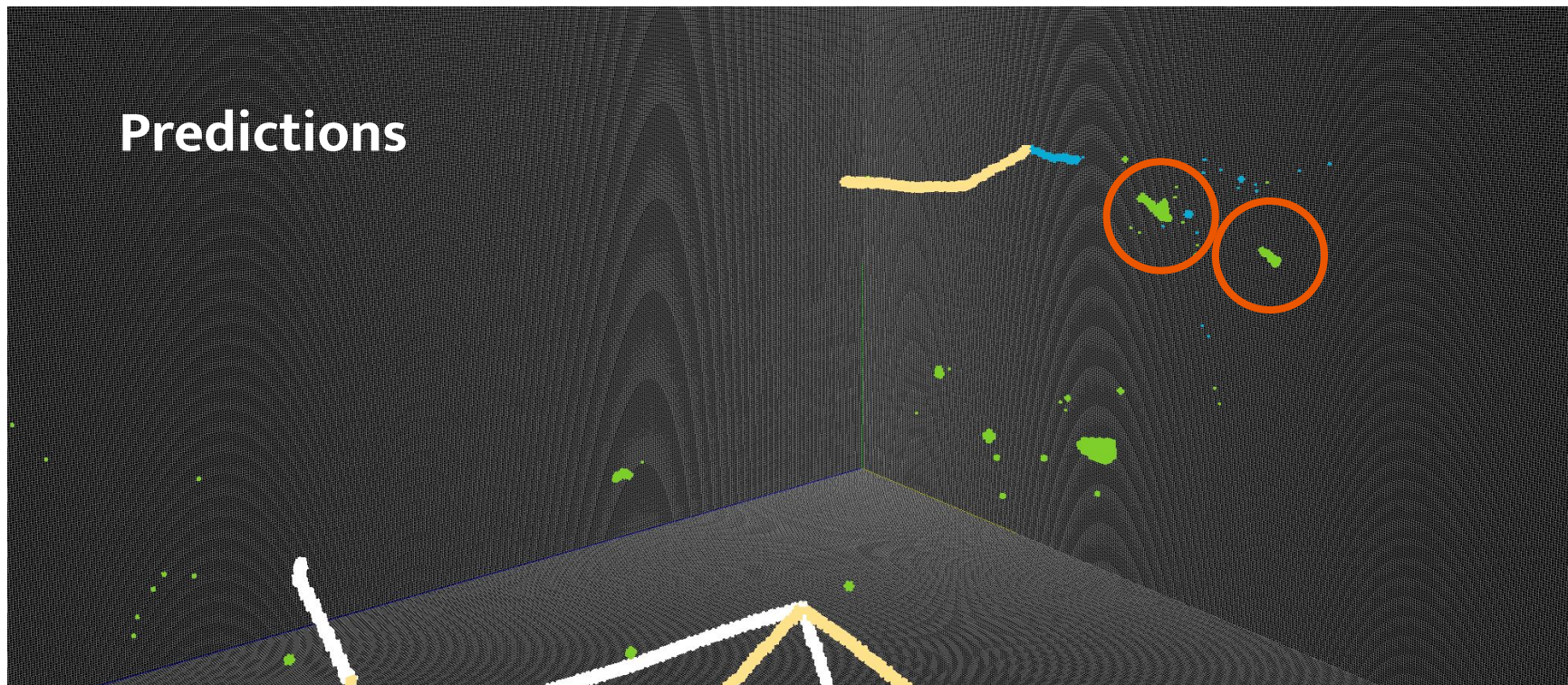


*Std of mean EM shower softmax score in an image, across training iterations*

# Inference duration

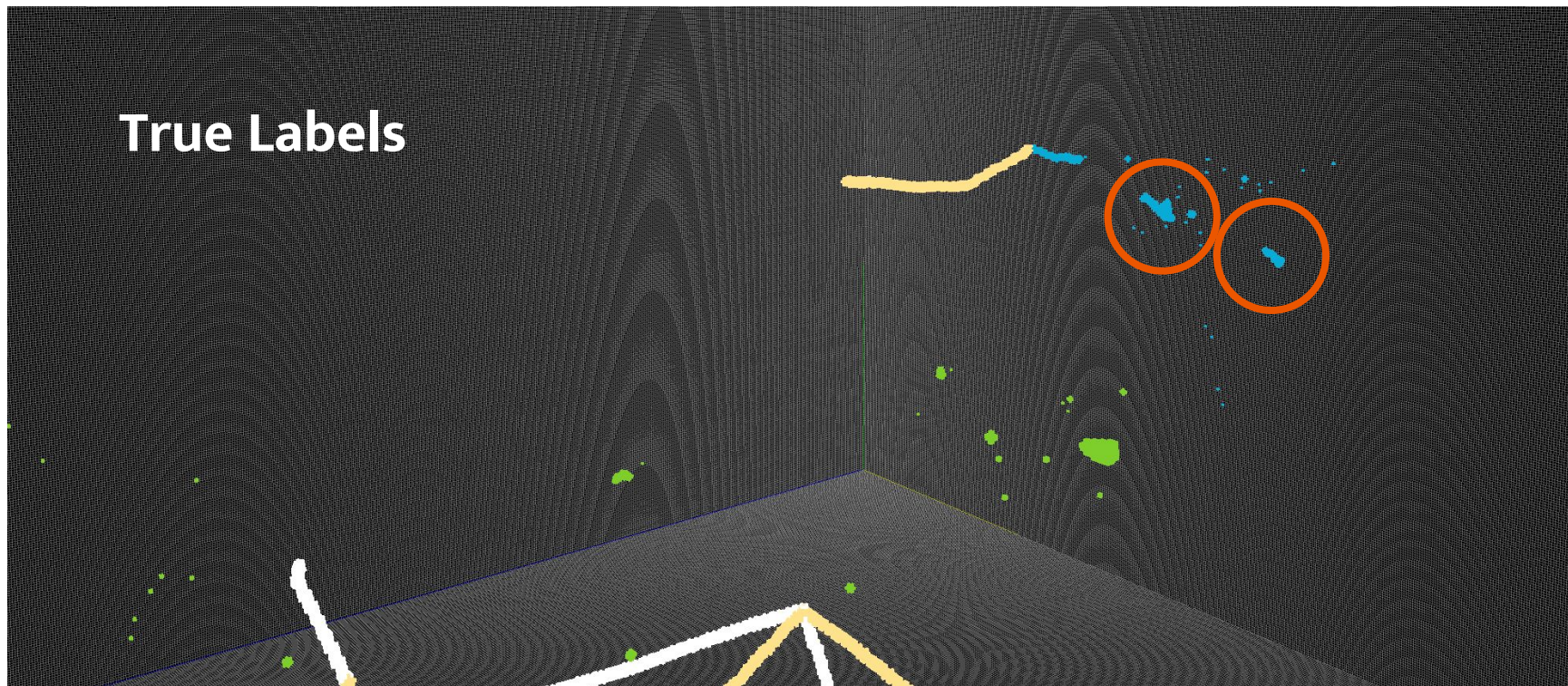


# Learning from mistakes: the case of Michel electrons



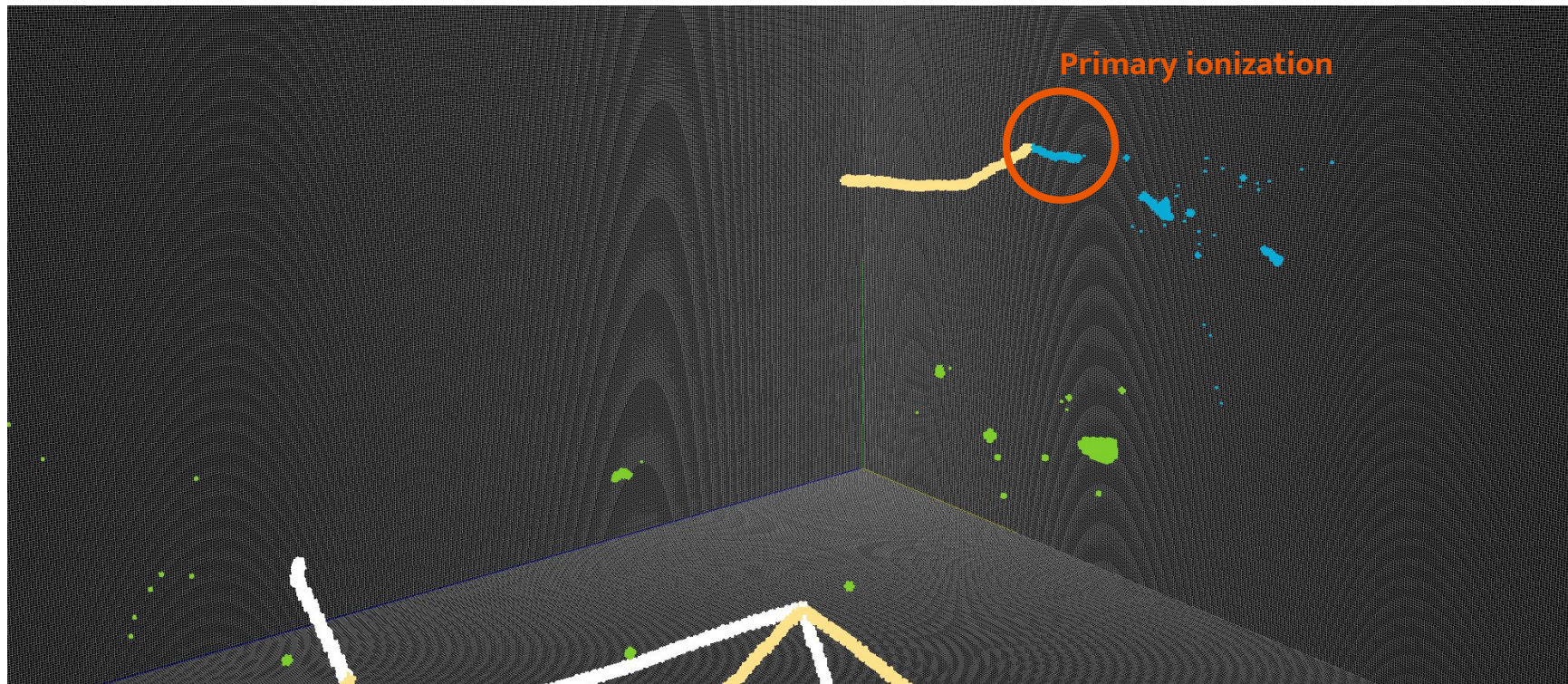


# Learning from mistakes: the case of Michel electrons



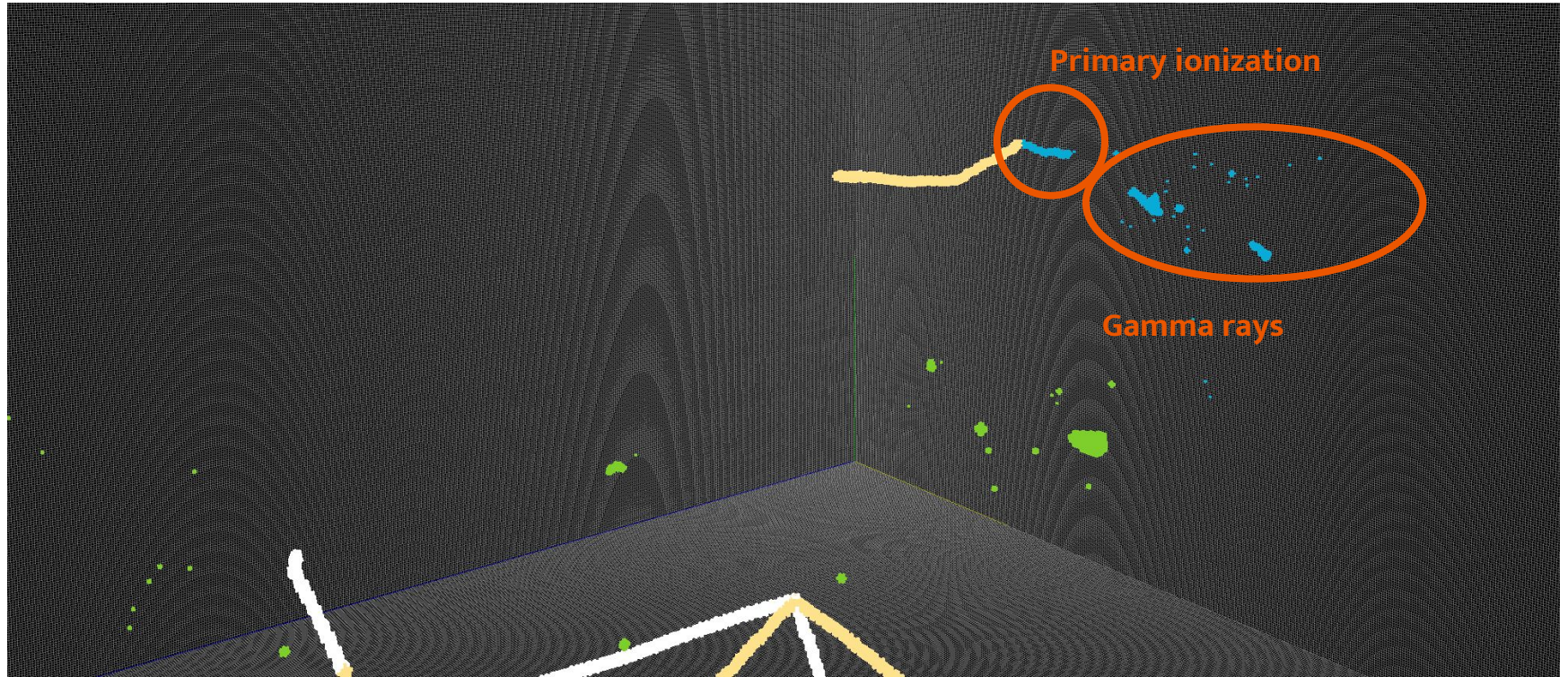


# Learning from mistakes: the case of Michel electrons





# Learning from mistakes: the case of Michel electrons

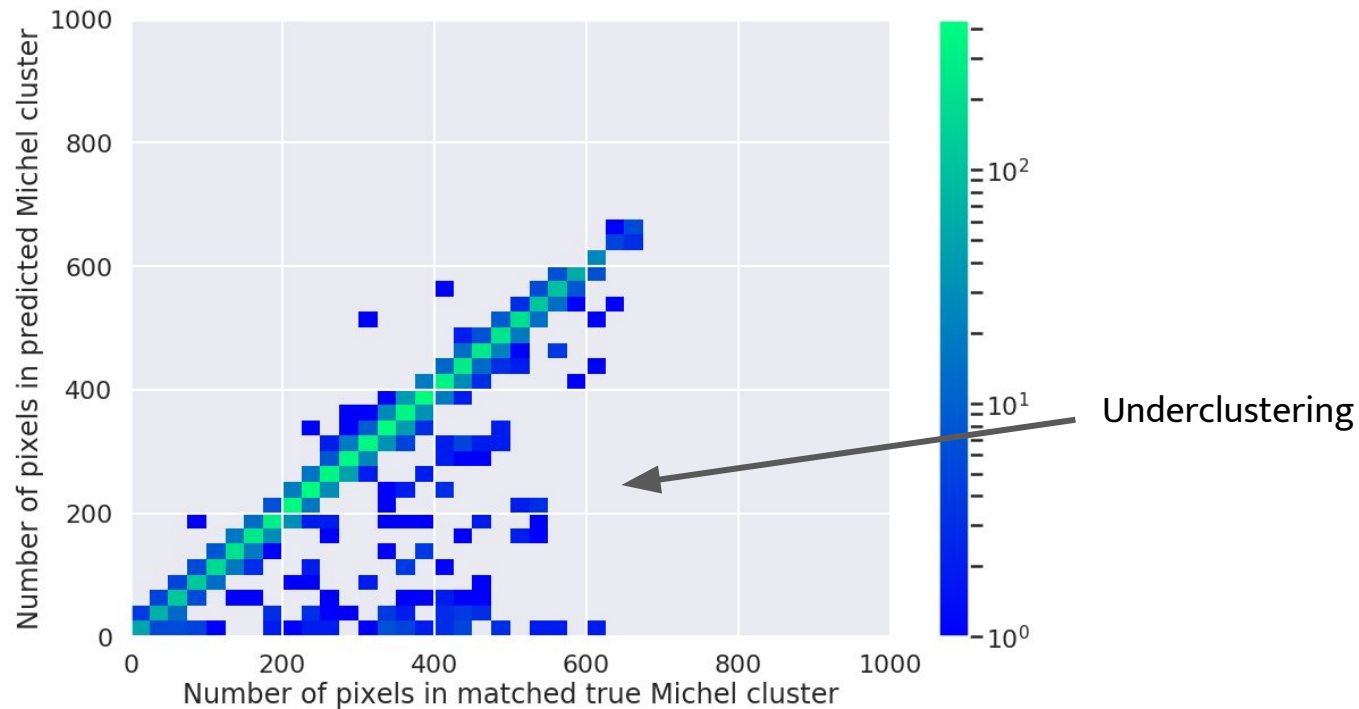


# Relabeling study

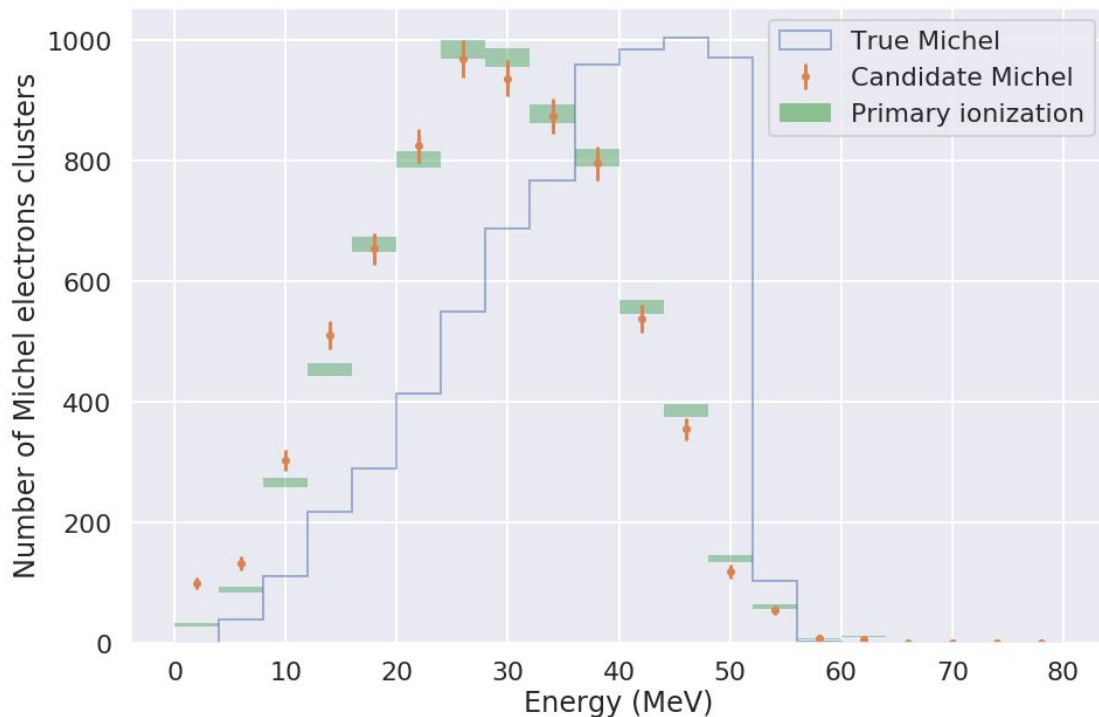
*Relabeled dataset = only primary ionization is labelled as Michel electrons*

Train data	Regular		Relabeled	Relabeled + Weighting
Test data	Regular	Relabeled		
HIP	98.0%	98.1%	98.1%	99.3%
MIP	99.4%	99.2%	99.4%	98.1%
EM shower	99.4%	97.9%	99.2%	99.2%
Delta rays	85.7%	94.8%	96.0%	97.2%
Michel electrons	<b>56.6%</b>	<b>94.4%</b>	<b>94.7%</b>	<b>95.7%</b>

# Number of pixels in candidate vs matched Michel cluster



# Michel electrons energy spectrum reconstruction



Sample size	7105
Identification purity	98.8%
Identification efficiency	93.9%
Cluster efficiency	96.1%
Cluster purity	97.3%

**First ML-based approach**  
Simulation only, next step is data!