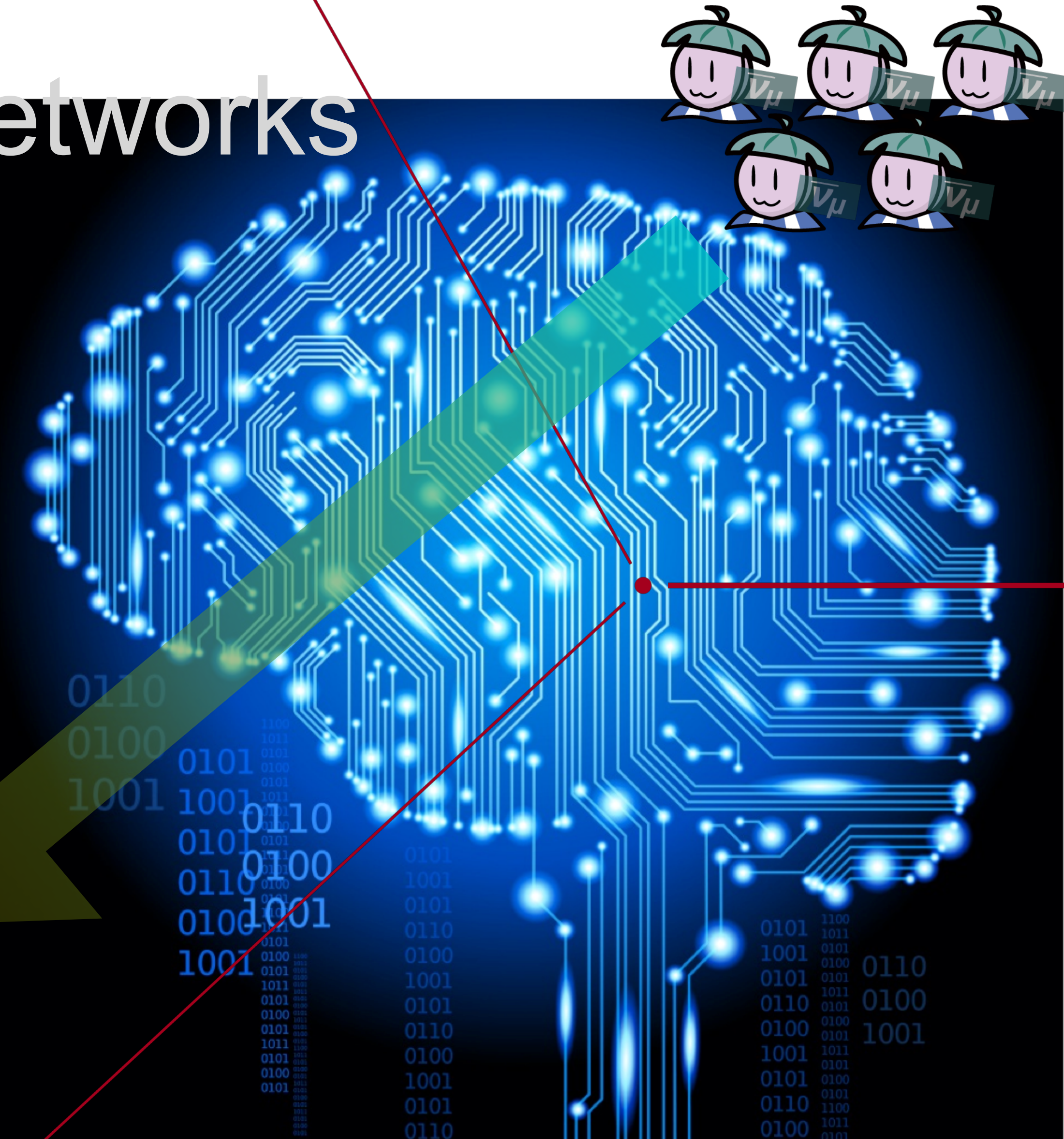
# Deep Neural Networks for 3D Data Reconstruction

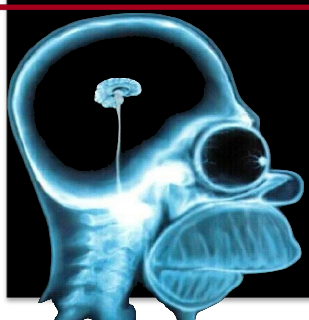


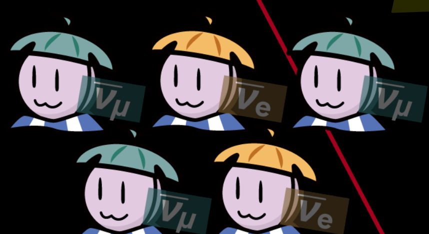

*Kazuhiro Terao*

*SLAC National Accelerator Lab.*


*POND² @ Fermilab*

*December 6th, 2018*





# LArTPC Data Reconstruction

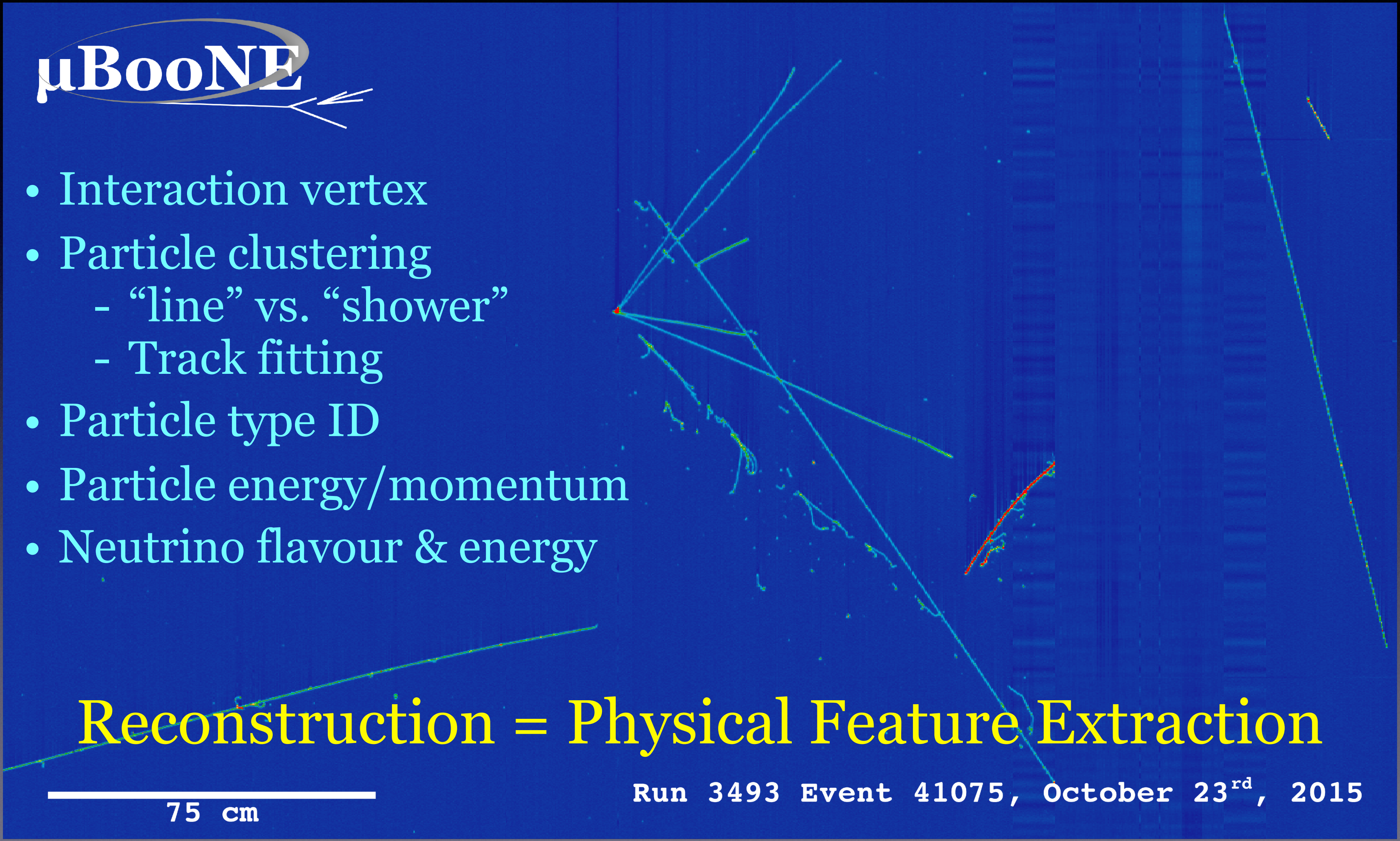

Particle imaging
= visually intuitive

# LArTPC Data Reconstruction

μBooNE

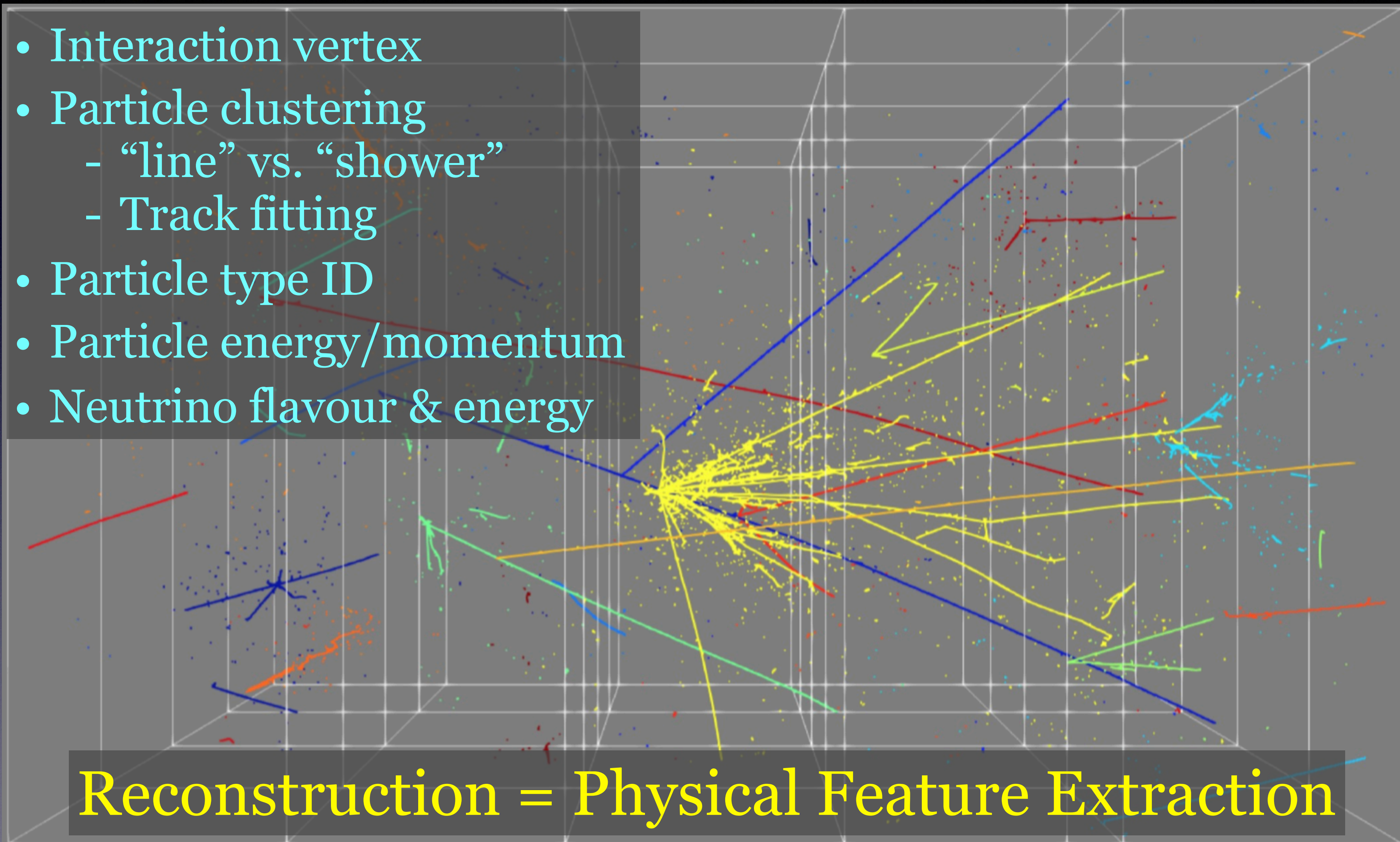
- Interaction vertex
- Particle clustering
  - "line" vs. "shower"
  - Track fitting
- Particle type ID
- Particle energy/momentum
- Neutrino flavour & energy

**Reconstruction = Physical Feature Extraction**



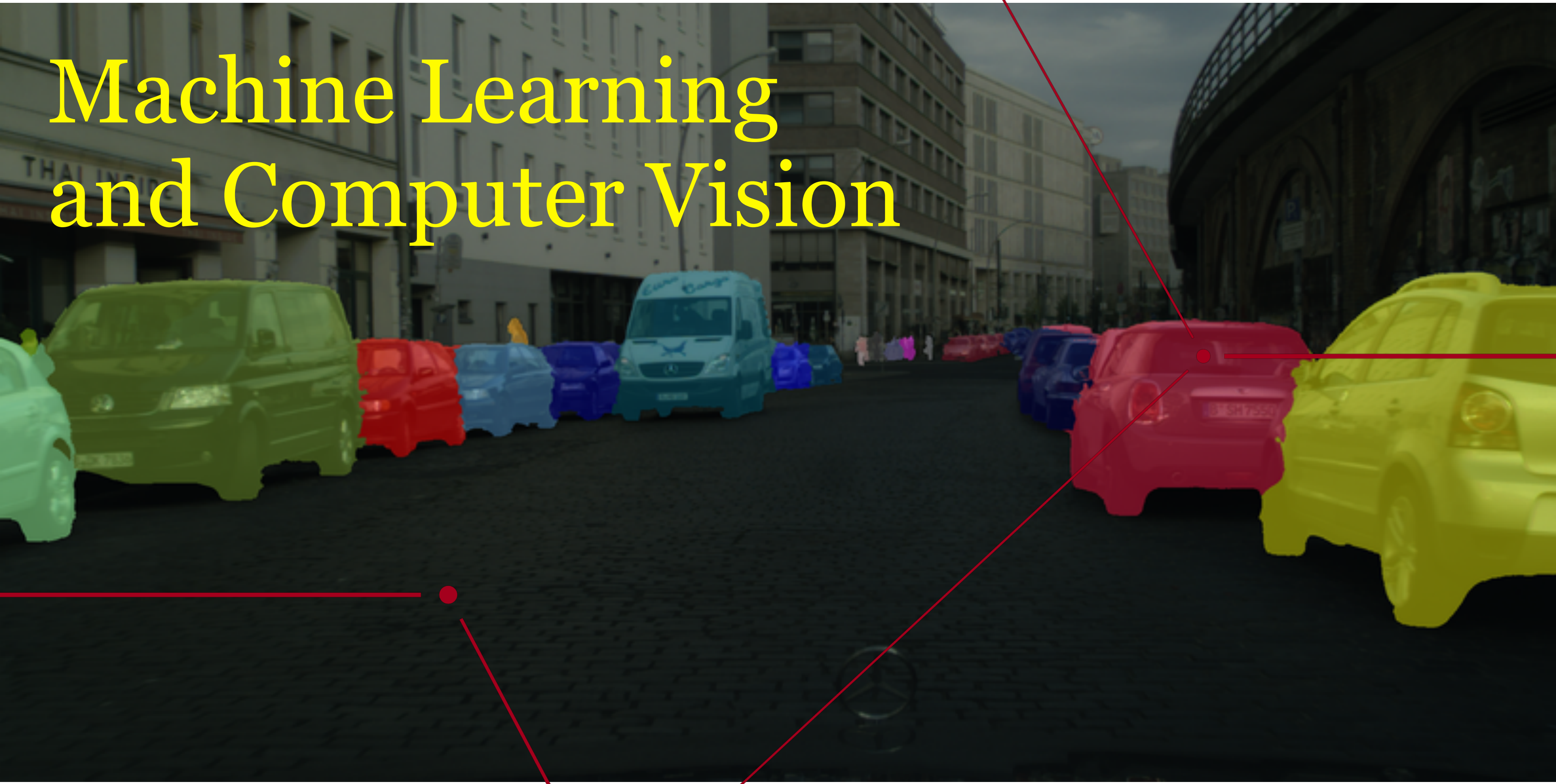
# LArTPC Data Reconstruction

- Interaction vertex
- Particle clustering
  - “line” vs. “shower”
  - Track fitting
- Particle type ID
- Particle energy/momentum
- Neutrino flavour & energy

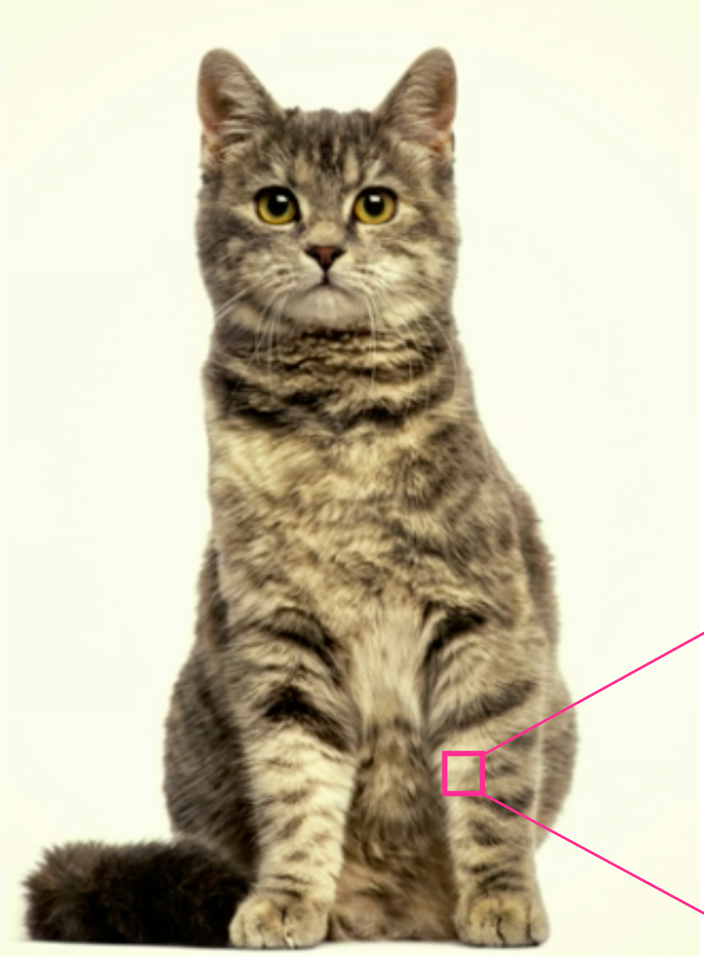


**Reconstruction = Physical Feature Extraction**

# Machine Learning and Computer Vision



# Challenge in Computer Vision



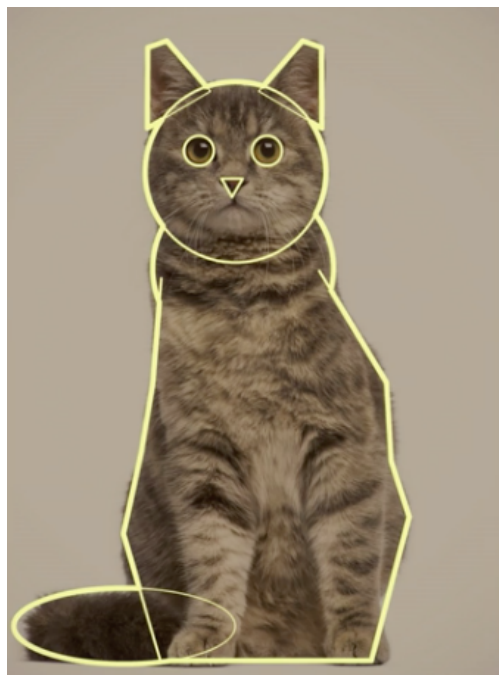
How to write an algorithm to identify a cat?

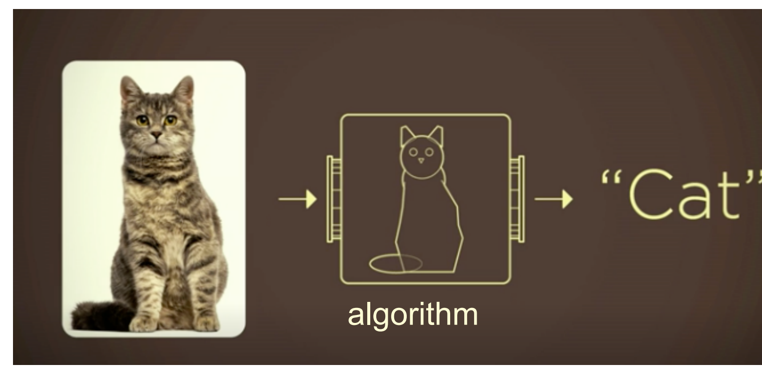
… very hard task …


Image credits: TED talk by Fei-Fei Li

## Development Workflow for non-ML algorithms

1. Write an algorithm based on basic (physics) principles






A cat = collection of certain shapes
(or, a neutrino)



Image credits: TED talk by Fei-Fei Li

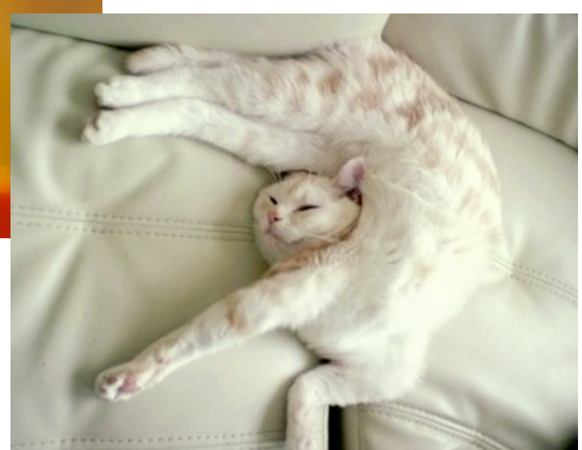
# Machine Learning
# Challenge in Computer Vision

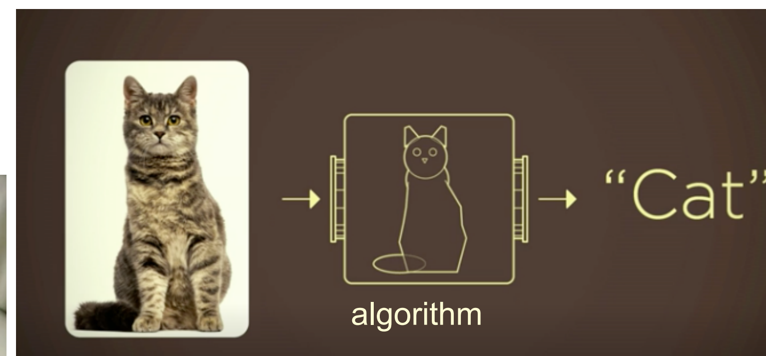
## **Development Workflow** for **non-ML** algorithms

1. Write an algorithm based on basic (physics) principles
2. Run on simulation/data samples
3. Observe failures, implement fixes/heuristics
4. Iterate over 2 & 3 till a satisfactory level is achieved
5. Chain multiple algorithms as one algorithm, repeat 2, 3, and 4.

**Partial cat**
(escaping muon)

**Stretching cat**
(Nuclear FSI)






A cat = collection of certain shapes
(or, a neutrino)



Image credits: TED talk by Fei-Fei Li

## Development Workflow for non-ML algorithms

1. Write an algorithm based on basic (physics) principles
2. Run on simulation/data samples
3. Observe failures, implement fixes/heuristics
4. Iterate over 2 & 3 till a satisfactory level is achieved
5. Chain multiple algorithms as one algorithm, repeat 2, 3, and 4.

## Machine Learning

- **Learn patterns from data**
  - automation of steps 2, 3, and 4
- **Chain algorithms & optimize**
  - step 5 addressed by design
- **"Deep Learning"**
  - Revolutions in computer vision using deep **neural networks**

Natural
Neural
Network

# Machine Learning
# Revolution with Deep Neural Networks

SLAC

**2012**

Public image categorization competition w/ 1.2M images, 1000 object categories.

IMAGENET

**"Deep" convolutional neural network** broke the past record by a large margin

**ImageNet Classification with Deep Convolutional Neural Networks**


Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

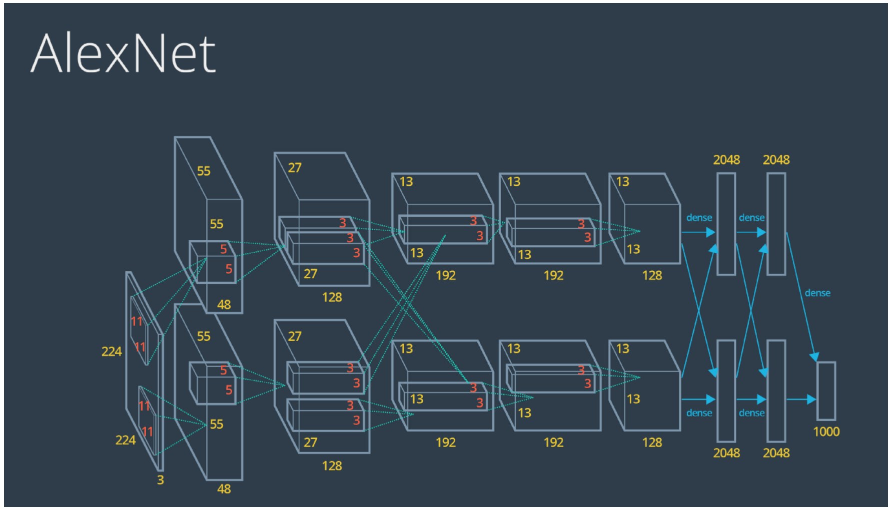
Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

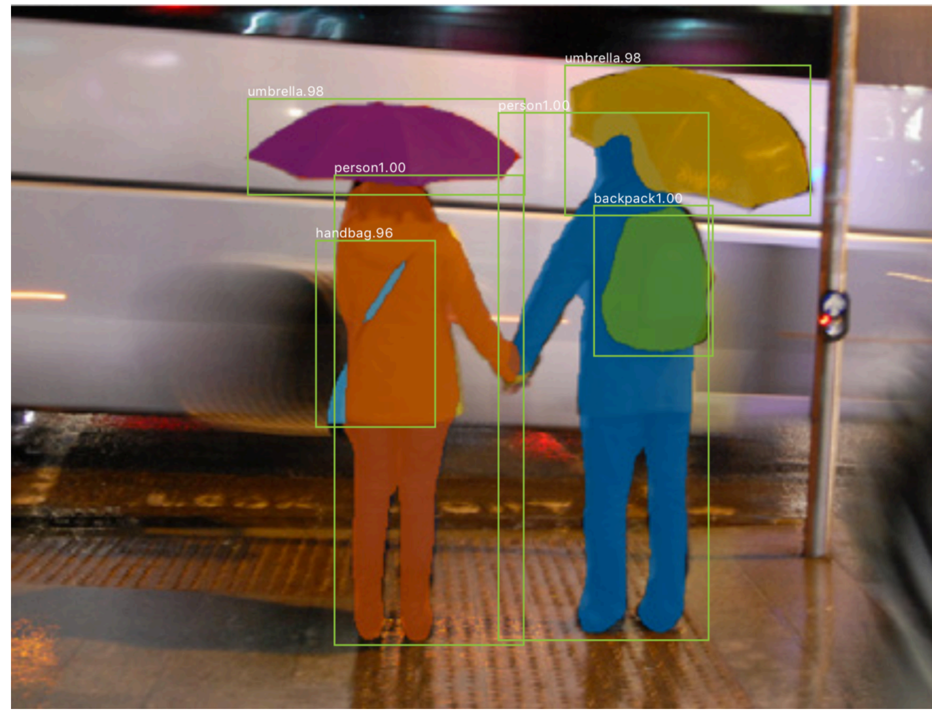

**> 30,000 citations**


**Abstract**

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

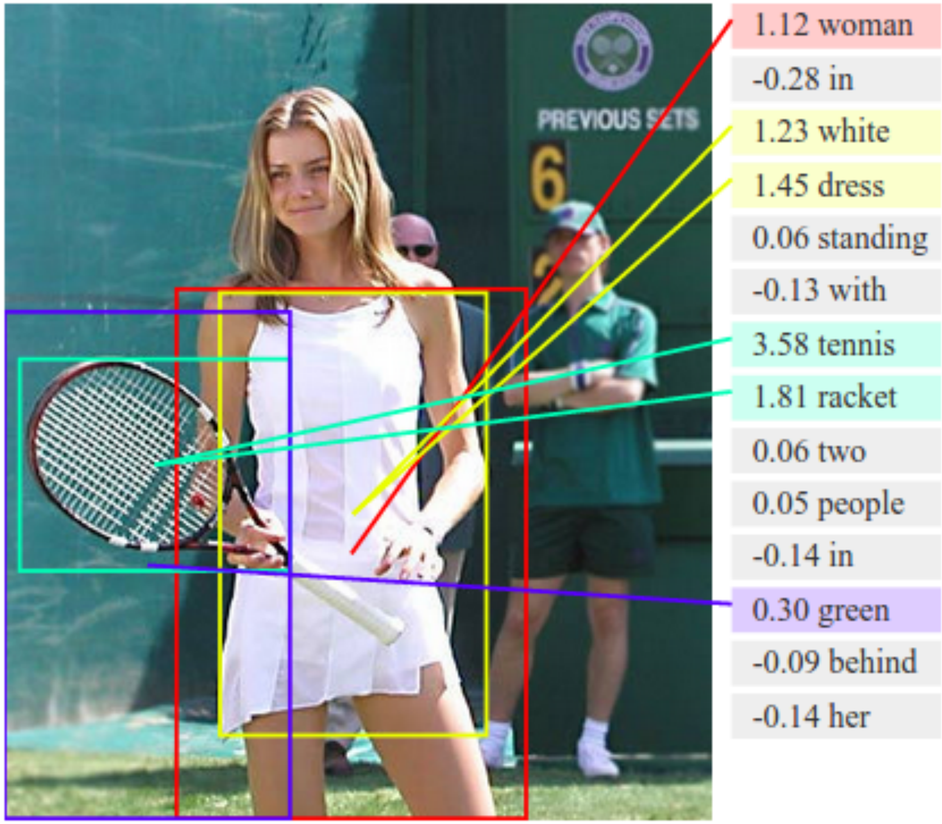




# Detection of Image Contexts

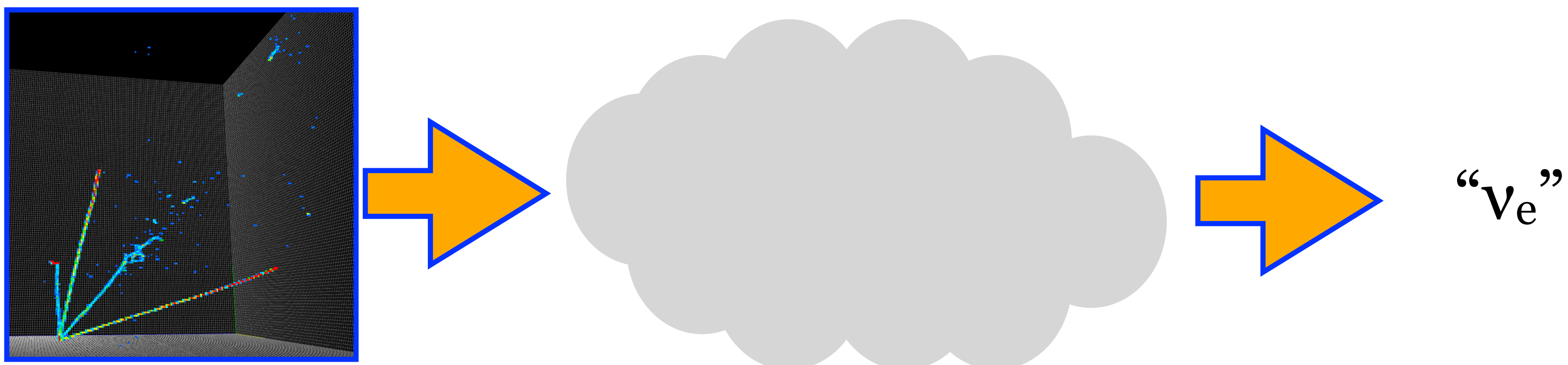
# Interpretation of Contexts' Correlation


"girl in pink dress is jumping in air."



# Machine Learning
# … for LArTPC Image Analysis

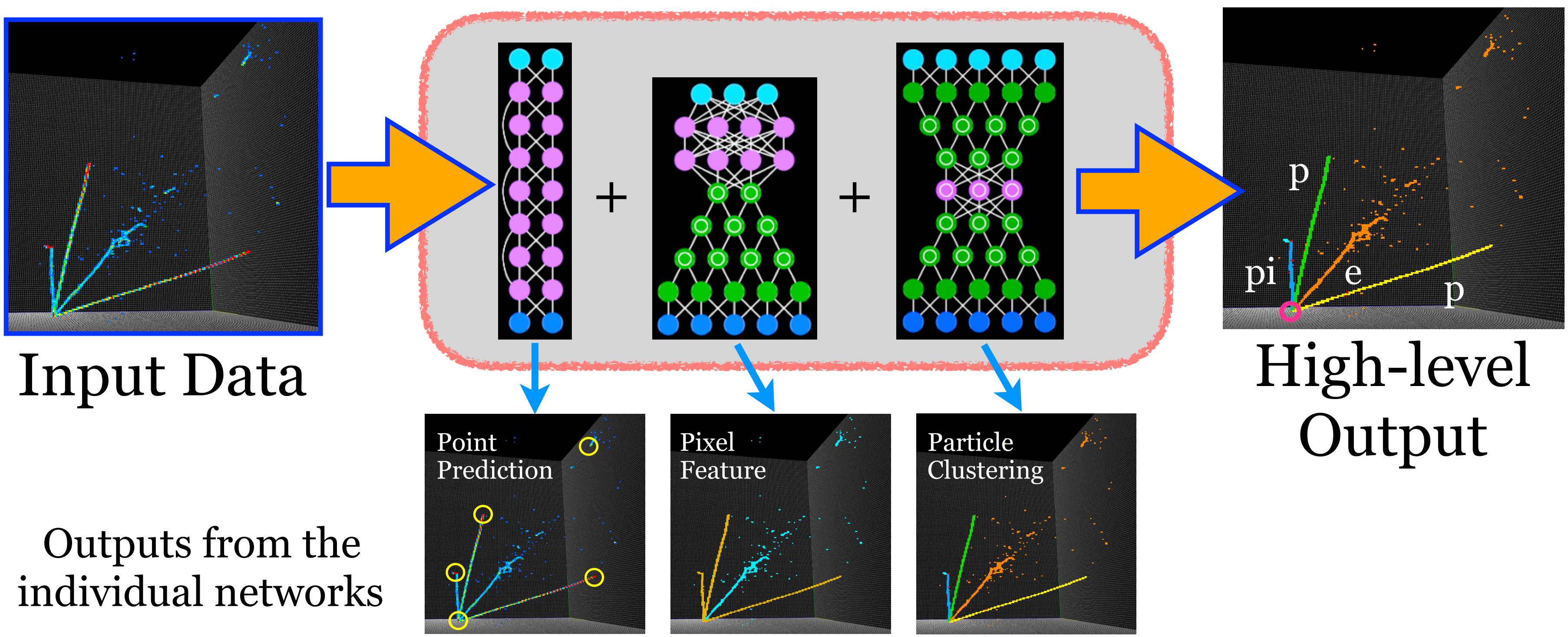
DNN is an efficient data transformation technique. “Image classification” transforms visually intuitive 2D/3D data into an array of discriminants (classes).




Input Data

# Multi-task Deep Neural Network

Introduce physical feature extraction tasks (reconstruction) to bias the data transformation. Implicitly introduce physics concepts + construct logic for the final output.



# Machine Learning
## … for LArTPC Data Reconstruction

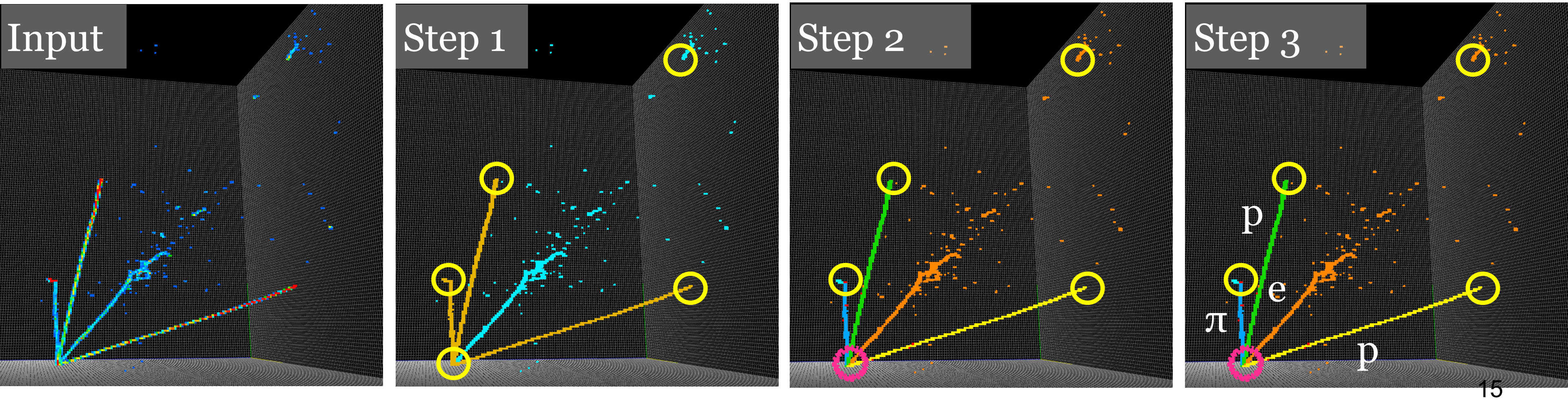
SLAC

## ML-based Full Data Reconstruction Chain

- A cluster of many task-specific networks in 2D & 3D
  - Vertex finding, clustering, particle ID, etc.

- [ ] 1. Space point (track edges) + pixel feature annotation
- [ ] 2. Vertex finding + particle clustering
- [ ] 3. Particle type + energy/momentum
- [ ] 4. Hierarchy building

My ECA Program



# Machine Learning in ~~Computer Vision~~

## **High-Precision Detector Data Analysis**

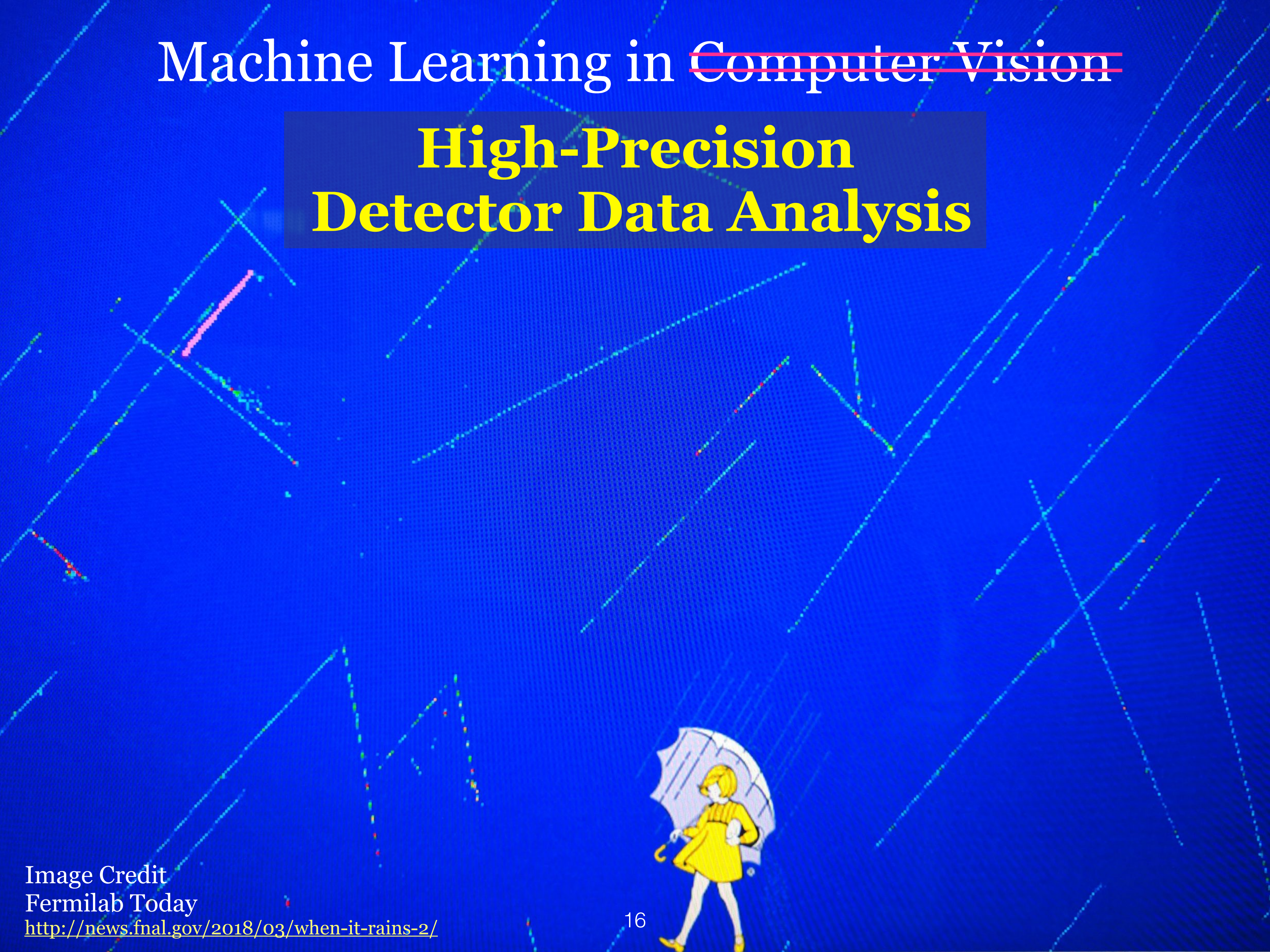
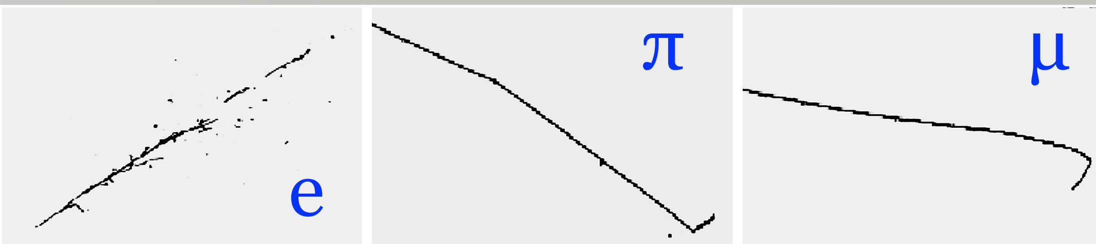
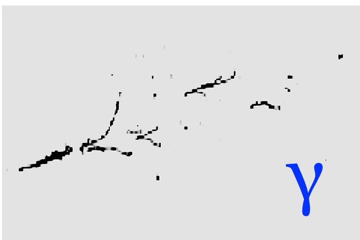


Image Credit
Fermilab Today
http://news.fnal.gov/2018/03/when-it-rains-2/

# Early Demonstrations
## Machine Learning for LArTPC Image Analysis

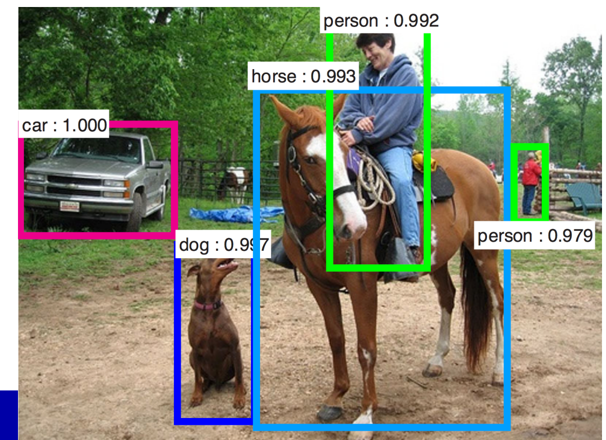


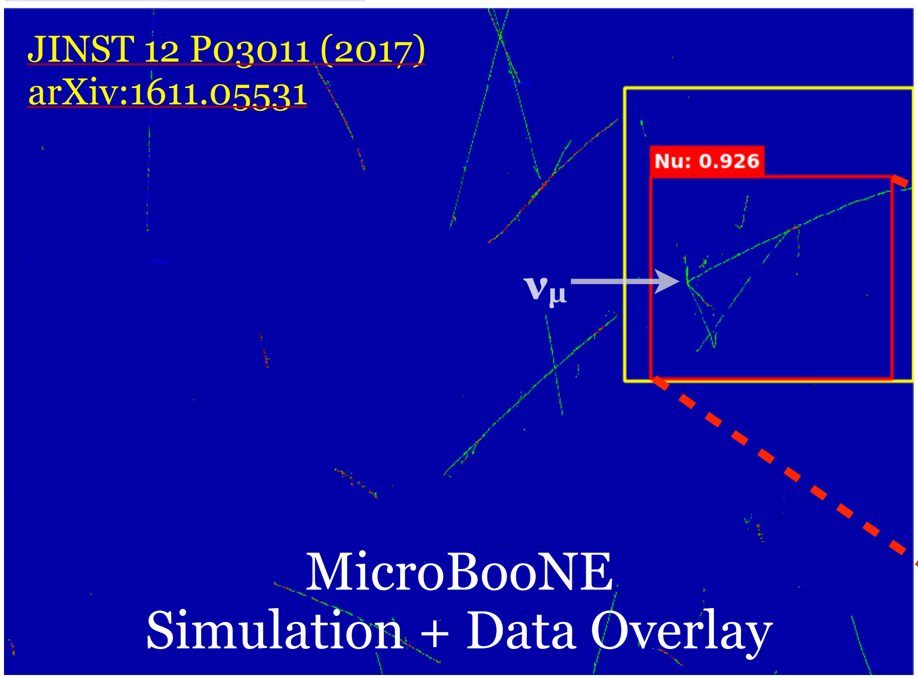




### ML Technique @ MicroBooNE LArTPC Detector

### Image Classification

- Classify a whole image into object categories
- particle type identification from an image
- signal/background selection



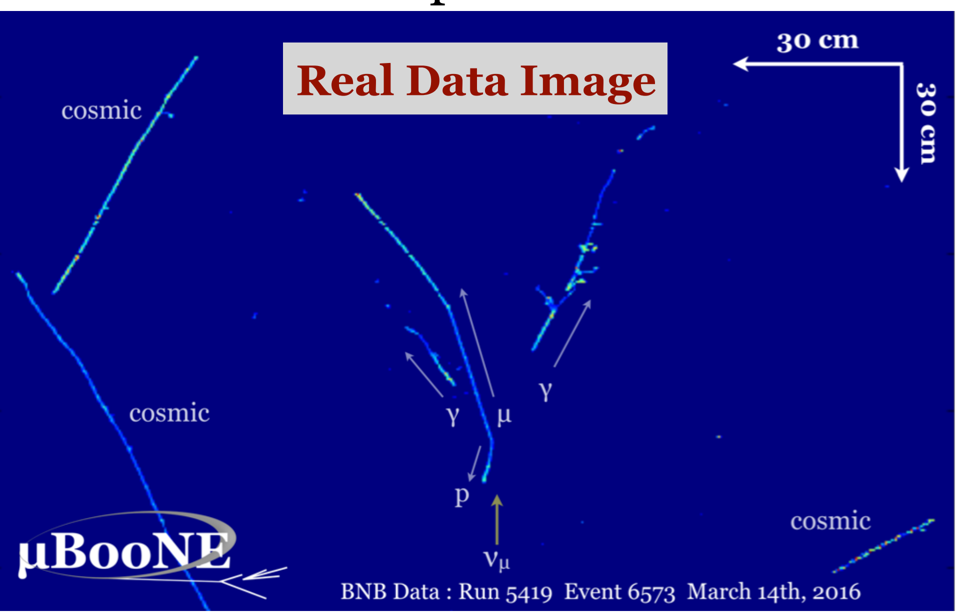




# Early Demonstrations
# Machine Learning for LArTPC Image Analysis

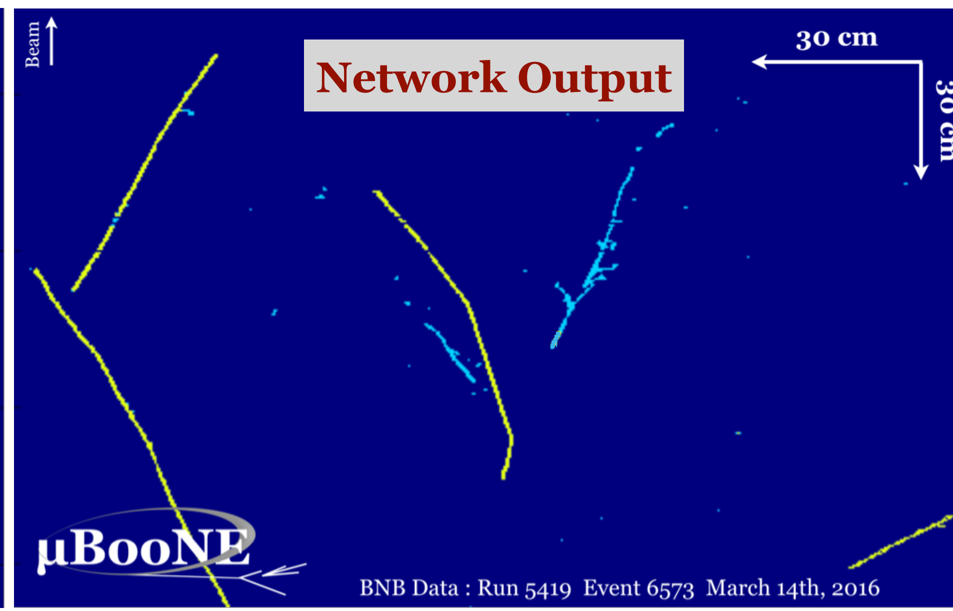
## Semantic Segmentation

- Recently published … arXiv:1808.07269
- Pixel-level object classification
  - Separation of EM-particle from other types
  - Key input information for particle clustering
- First time deep neural network validated on LArTPC data

## ML Technique @ MicroBooNE LArTPC Detector




Network Input

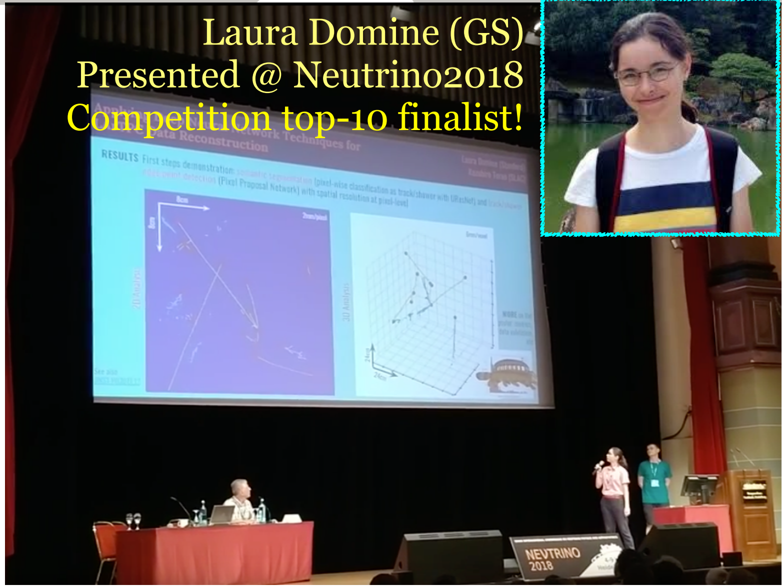


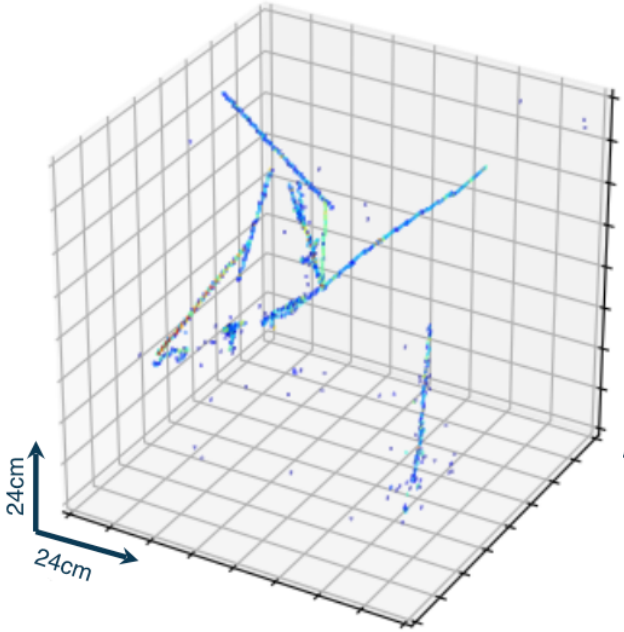

Network Output

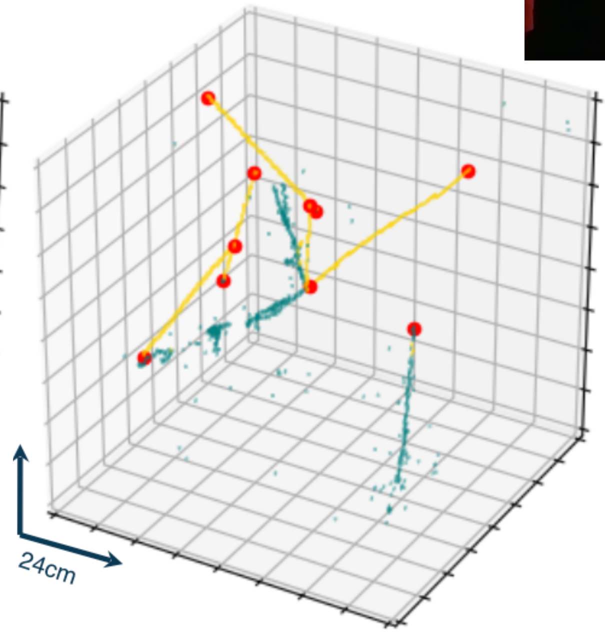
SLAC

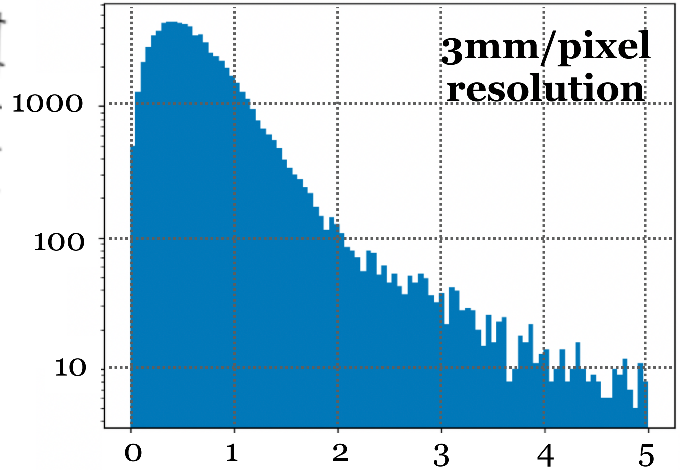
## Multi-Task Network Cascade

- **Chain of Segmentation + Detection**
  - Feature points: “shower start” and “track edges”
  - Classify each pixel into “shower” vs. “ track”
- **Extension to 3D data**
  - Change in tensor dimensions, identical algorithms



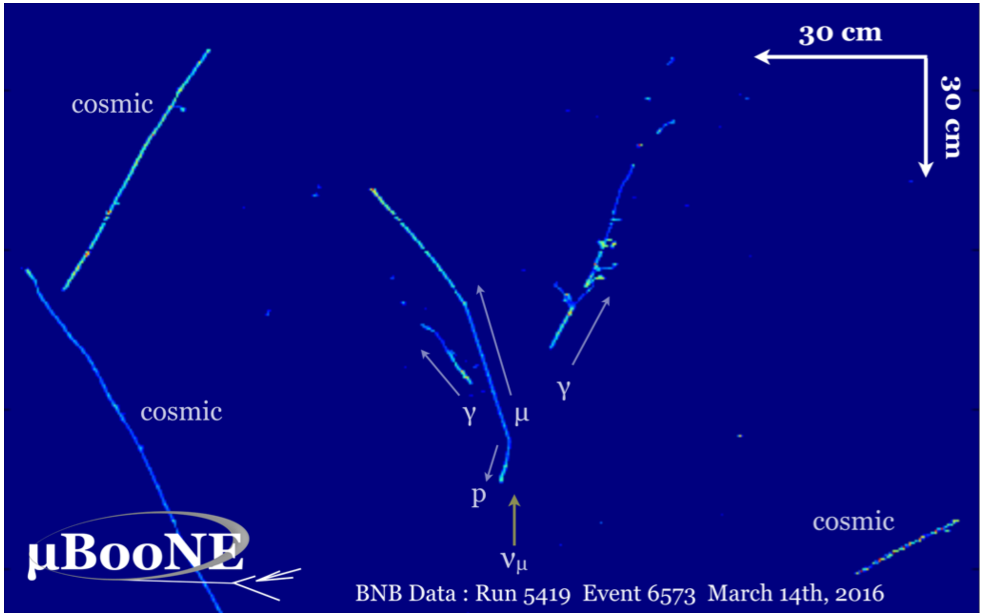

**Pixel distance** between the target truth point to the closest proposed point[19]

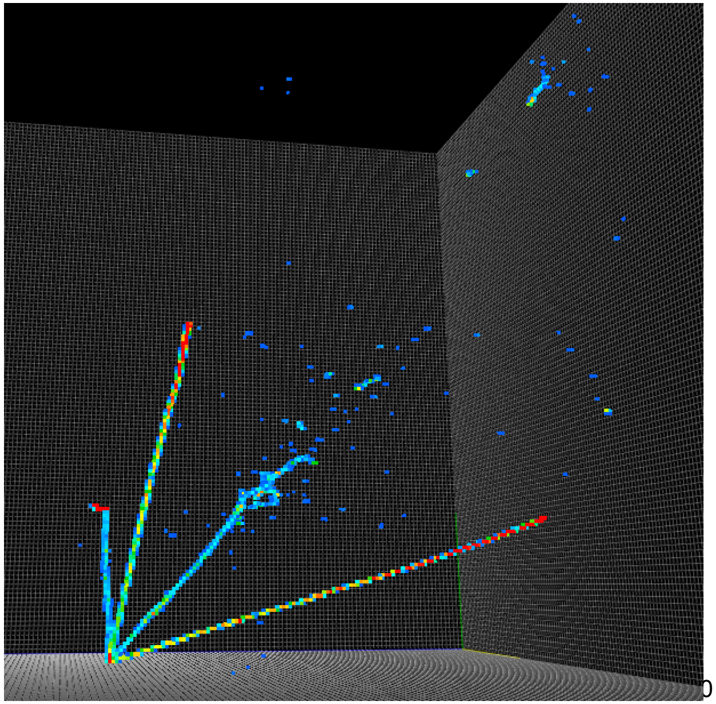
## Toward Full 3D Reconstruction Chain
## Machine Learning for Particle Image Analysis

SLAC

**“Applying for 3D” is simple, but is it scalable?**

- LArTPC data is generally sparse but locally dense
  - Mostly zero-filled matrix. CNN = dense matrix operation = bad!
  - Matrix size (volume) scales by power low, but non-zero pixels scales almost linearly (most particle trajectories are locally 1D line)



**"Applying for 3D" is simple, but is it scalable?**

- LArTPC data is generally sparse but locally dense
  - Mostly zero-filled matrix. CNN = dense matrix operation = bad!
  - Matrix size (volume) scales by power low, but non-zero pixels scales almost linearly (most particle trajectories are locally 1D line)
- CNN causes "blurring" which can be severe on sparse data

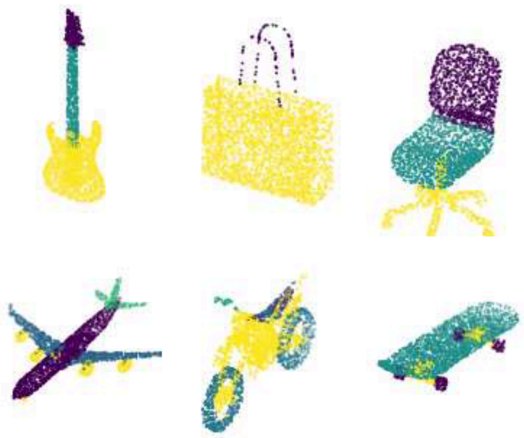
Input | After 1st convolution | After 2nd convolution

## "Applying for 3D" is simple, but is it scalable?

- LArTPC data is generally sparse but locally dense
  - Mostly zero-filled matrix. CNN = dense matrix operation = bad!
  - Matrix size (volume) scales by power low, but non-zero pixels scales almost linearly (most particle trajectories are locally 1D line)
- CNN causes "blurring" which can be severe on sparse data

## Solution: Sparse Submanifold Convolution

- **Submanifold** = "Input data with lower effective dimension than the space in which it lives"
- Can extract lower dimensional features effectively
  - Ideally suited for our problems
- Developed by [Facebook AI Research] / **Oxford**
  - CVPR2018, best 3D semantic segmentation record for ShapeNet (open 3D point-cloud dataset)



# Works amazingly well…

| | Dense U-ResNet | Sparse U-ResNet |
|---|---|---|
| Process time (forward path) | 40 ms/img | 13 ms/img |
| Memory (forward path) | 13 GB | 600 MB |
| Train time (20 epochs) | 9 days | 8 hours |

Preliminary

- Using 3D data with 192^3 pixels
- 256 image/GPU in forward path
- Trained to reach 99% accuracy in the segmentation task
- Paper w/ more details coming out

## This is a game changer…

Curse of dimensionality almost addressed = scalable to big data

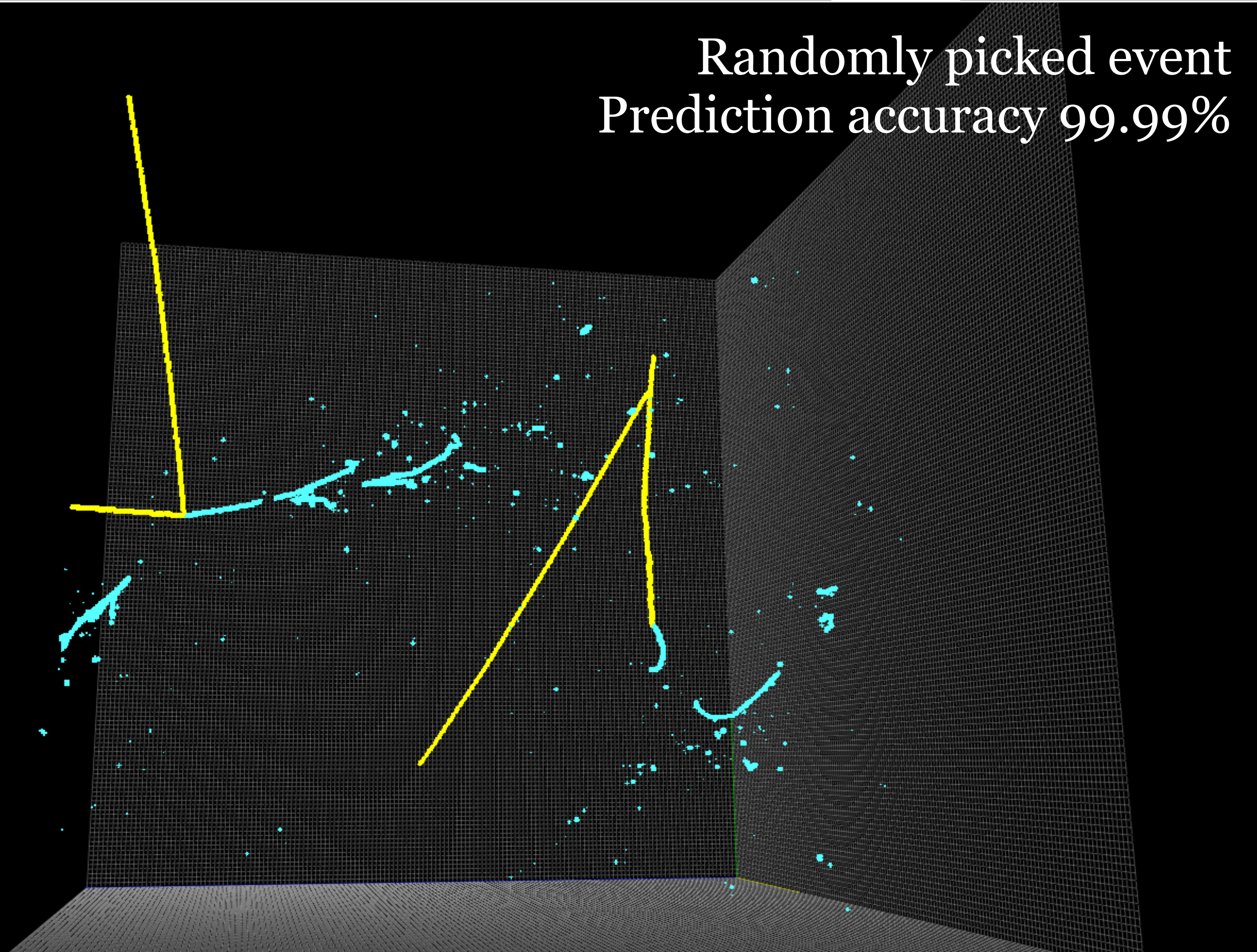
## What about accuracy?

It got better because the network can better focus on features

# Toward Full 3D Reconstruction Chain
## Machine Learning for Particle Image Analysis

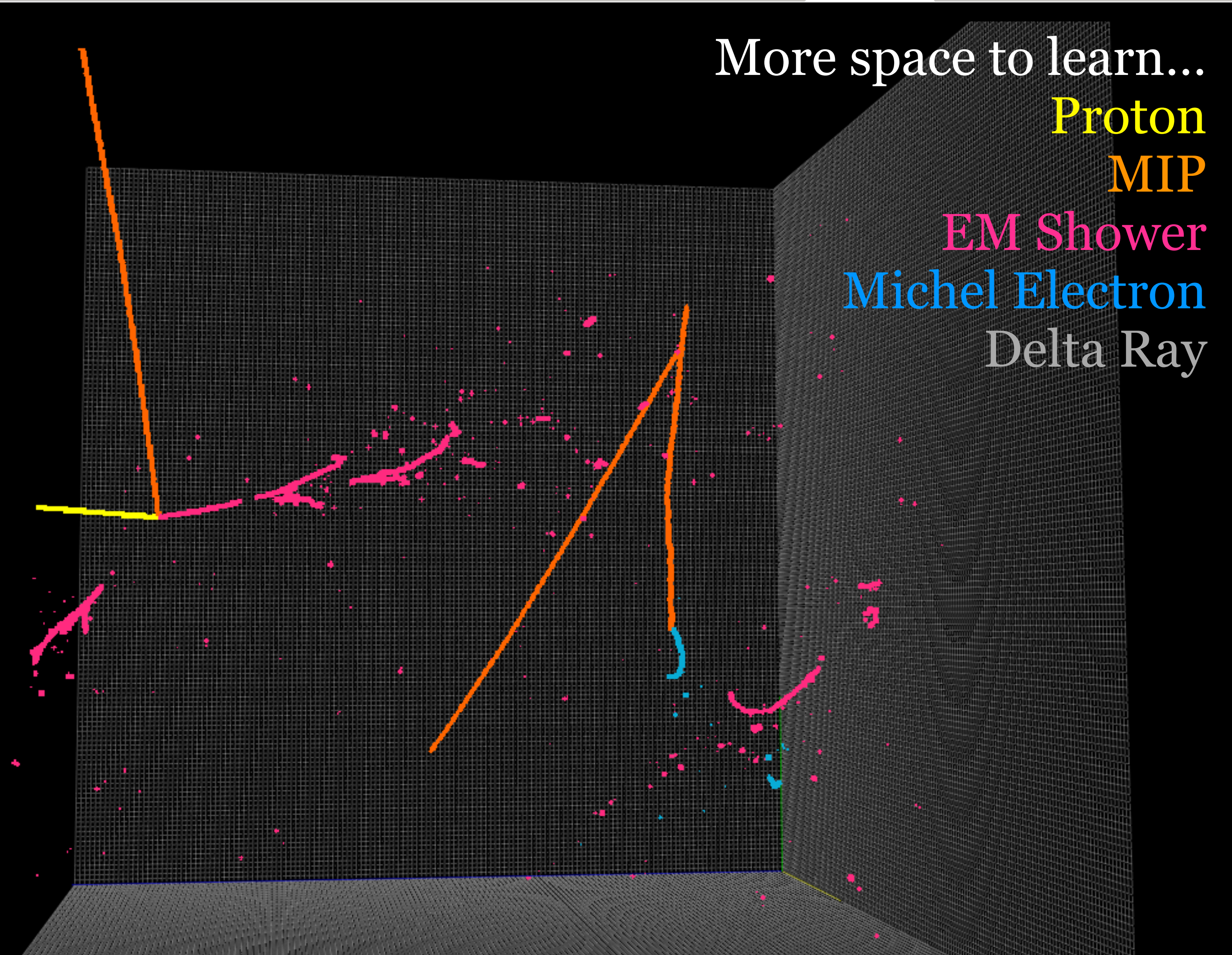
SLAC

Randomly picked event
Prediction accuracy 99.99%



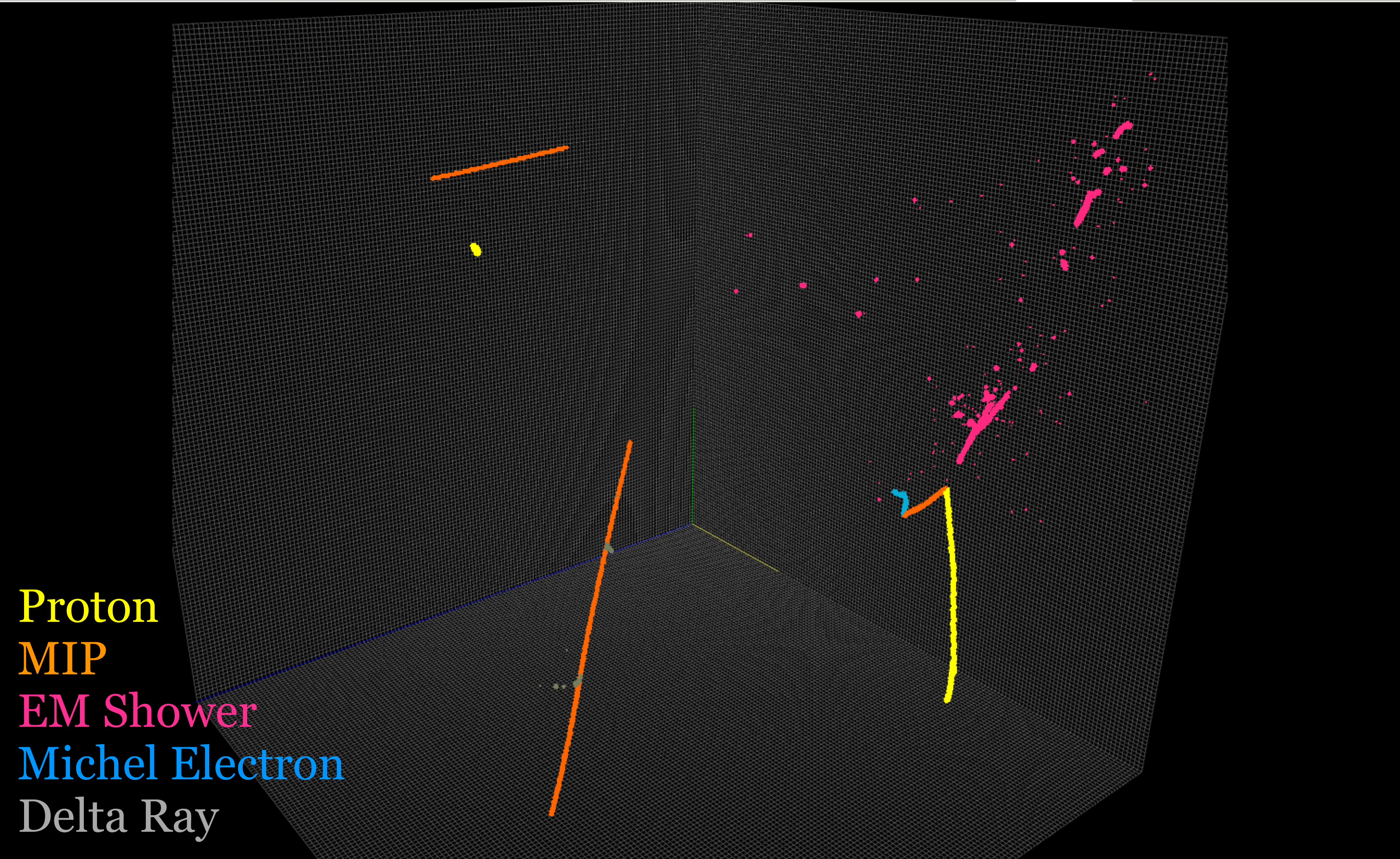
# Toward Full 3D Reconstruction Chain

## Machine Learning for Particle Image Analysis

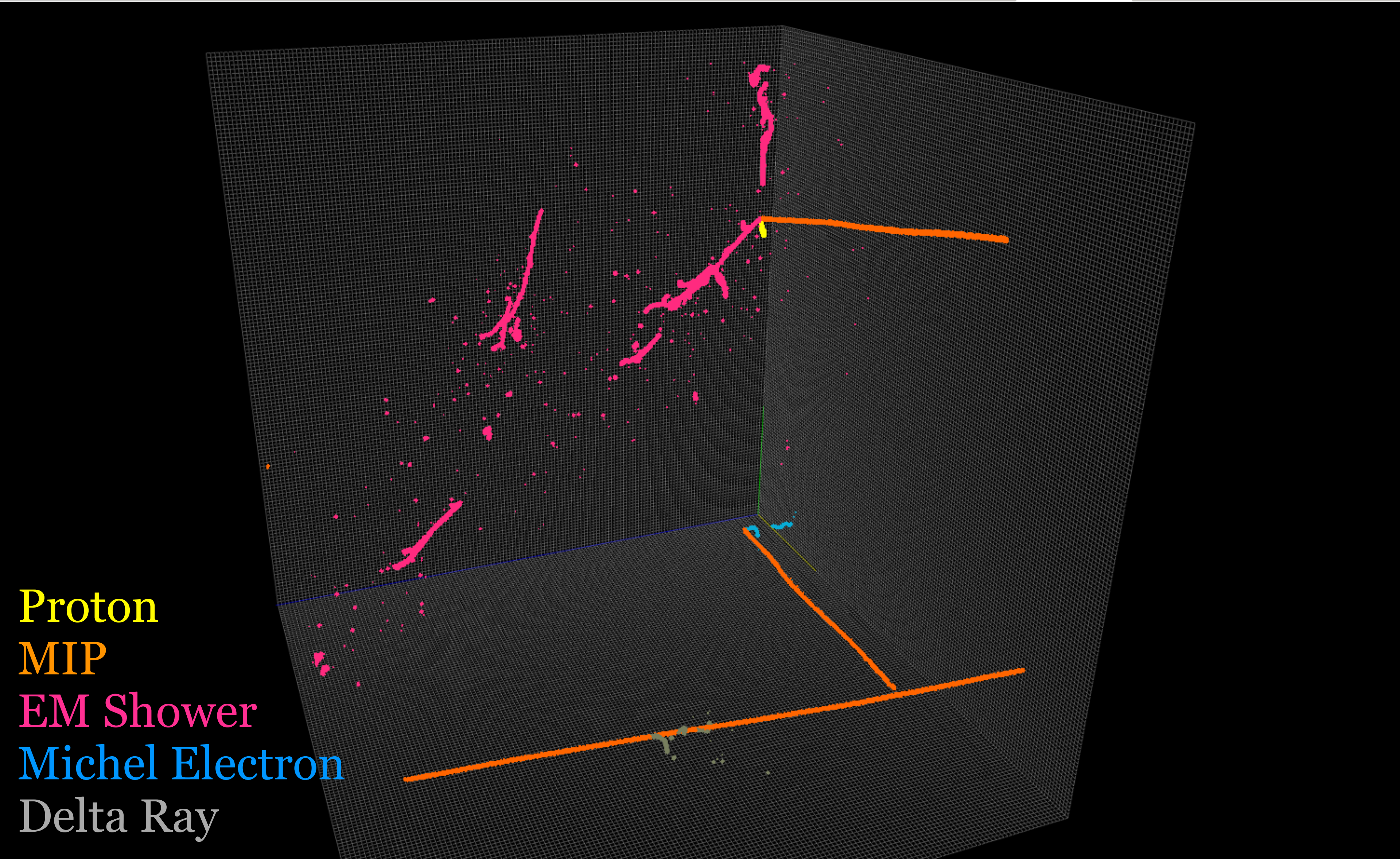


# Toward Full 3D Reconstruction Chain
## Machine Learning for Particle Image Analysis

SLAC



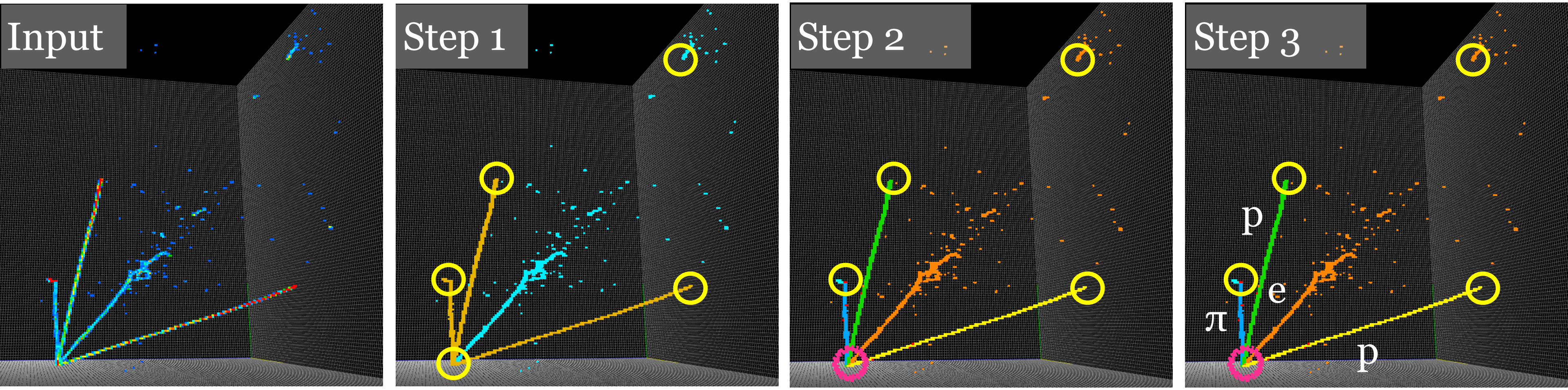
# Toward Full 3D Reconstruction Chain
## Machine Learning for Particle Image Analysis



**… wrapping up …**

# Where we are…

- ☑ 1. Space point (track edges) + pixel feature annotation
- ☑ 2. Vertex finding + particle clustering
- ☑ 3. Particle type + energy/momentum
- ☐ 4. Hierarchy building




Aiming to complete the full chain v.1 in early 2019, move to physics analysis applications

# Collaboration / Synergies

## Wire LArTPC

- WireCell team (BNL) on SBN/DUNE
- Cluster3D (SLAC) on SBN
- LArFlow (Tufts) on MicroBooNE

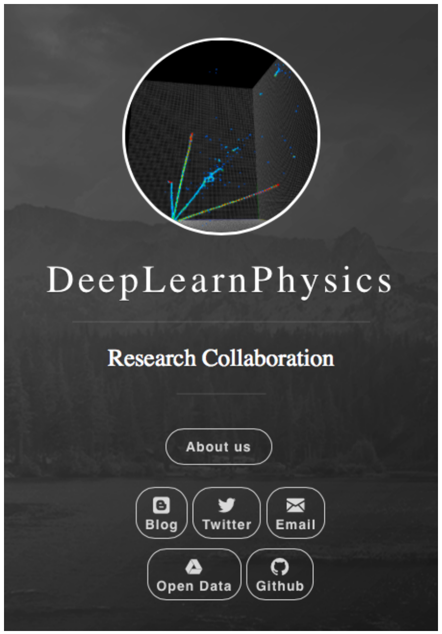
## Pixel LArTPC

- Interest from LBNL/UTA/Bern
  - Looking forward to 2x2 ArgonCUBE modules
  - Plan/Start working with students at LBNL and MSU specifically for DUNE ND

## Computing

- ANL demonstrating our code on distributed computing environment
- ALCC with FNAL+ORNL for DUNE ND study on Summit GPU HPC

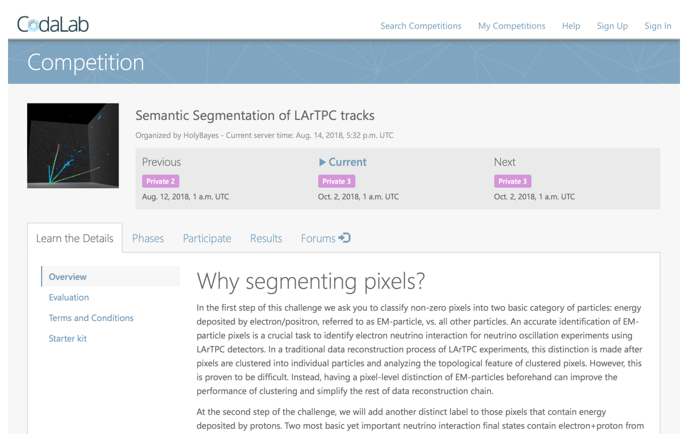
# Machine Learning & Broader Impact



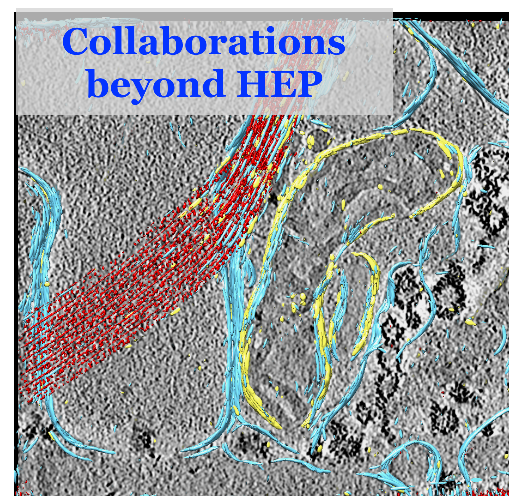

## DeepLearnPhysics (deeplearnphysics.org)

- **Collaboration** for ML technique R&D
  - ~70 members including HEP exp/theory, nuclear physics, BES (LCLS, SSRL), Cryo-EM, accelerator, AI/CS community
- **Open source software/tools, containers, open data**
  - our framework to collaborate & share reproducible results
- **Community building**
  - In-person tutorials (SLAC,LBNL,FNAL,BNL,VTech,MIT,Columbia…)
  - Sharing talk invitations, job/funding opportunities, etc.

Hands-on workshop @ SLAC/Stanford




Public challenge (collab. w/ LHC)

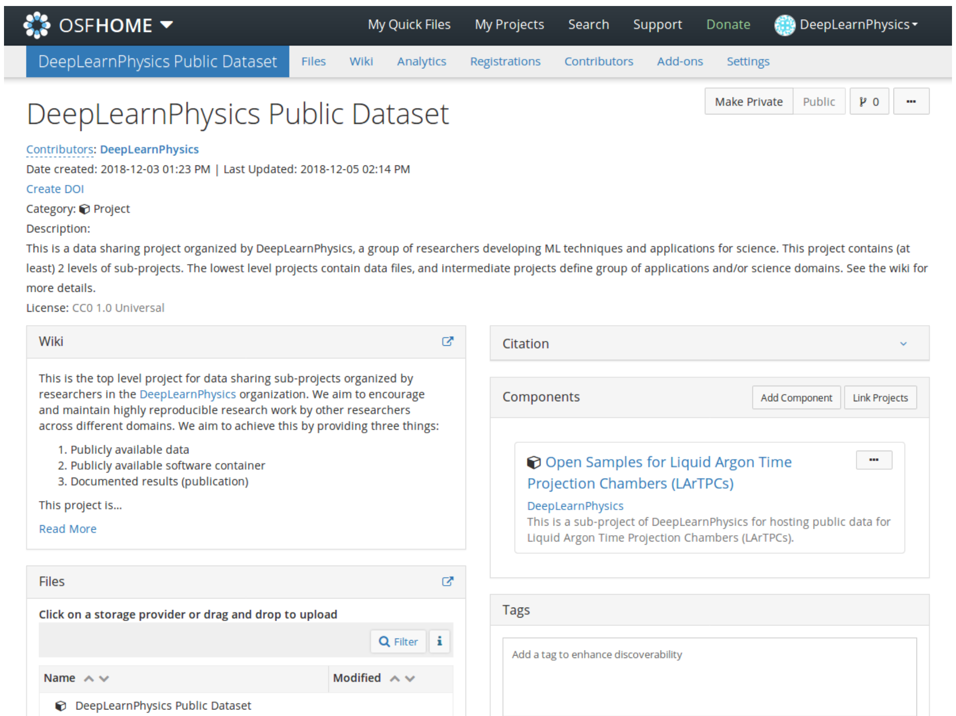


Collaborations beyond HEP

# Sharing Our R&D
# Machine Learning & Broader Impact

SLAC

## Public Data Set: OSF

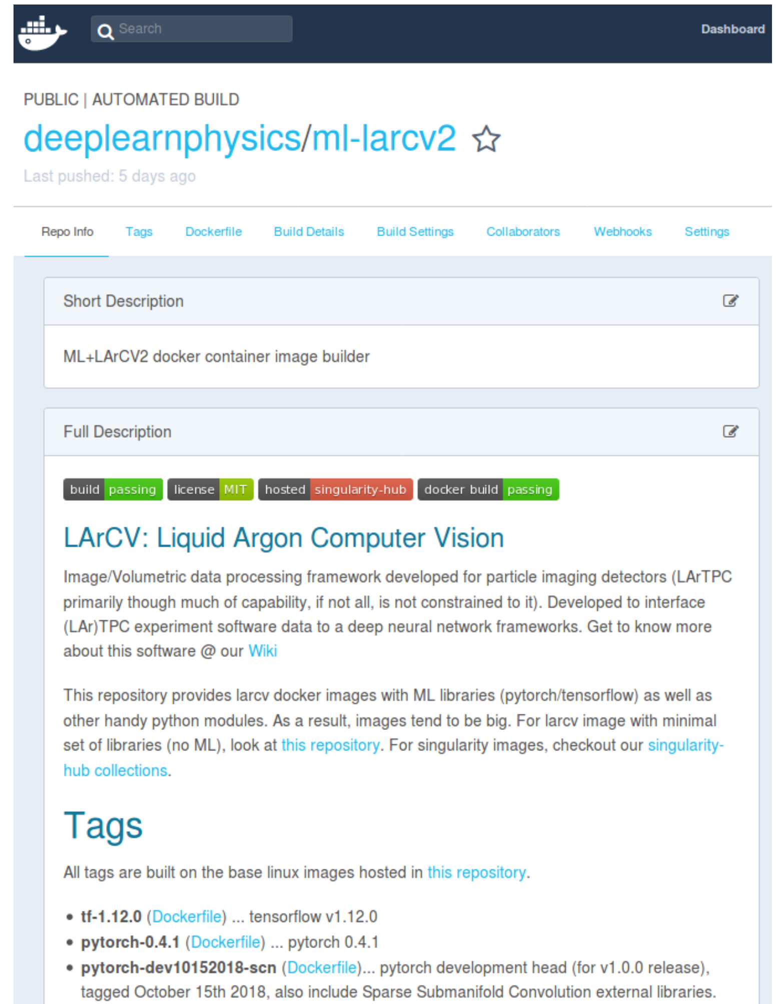



Travis CI

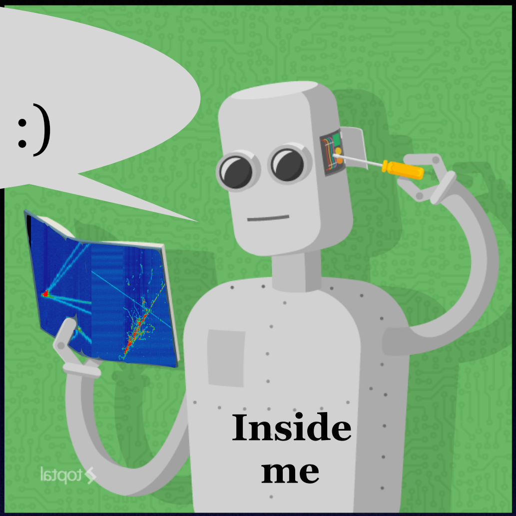
GitHub

docker

## Software Containers



**Thank you!**
for your attention :)




# Take Away Messages

1. **Deep neural networks (DNNs)** are **efficient image feature extraction techniques** developed in computer vision

2. **Sparse Submanifold Convolution** is **suited for LArTPC data and allows scalable DNN development**

3. **Full reconstruction chain** is **almost there, then physics!**

# Would love to collaborate?


**Projects I am responsible for…**

- Deep learning techniques R&D for LArTPCs
- HEP cross-frontier ML techniques R&D

**Other projects I work on…**

- Fermilab/ORNL for distributed ML algorithm optimization
- LBNL/Fermilab/CalTech for graph NN for particle clustering
- Fermilab for accelerated ML using edge computing devices
- LBNL/BNL for ML-based 3D pattern recognition
- MIT/Columbia for ML-based SBN data reconstruction
- Cryo-EM/SSRL (SLAC) for 3D tomogram analysis (biomedical)
- NASA-Ames/SLAC for pure Anomaly detection, computer vision

# Back-up Slides

# Liquid Argon Time Projection Chambers

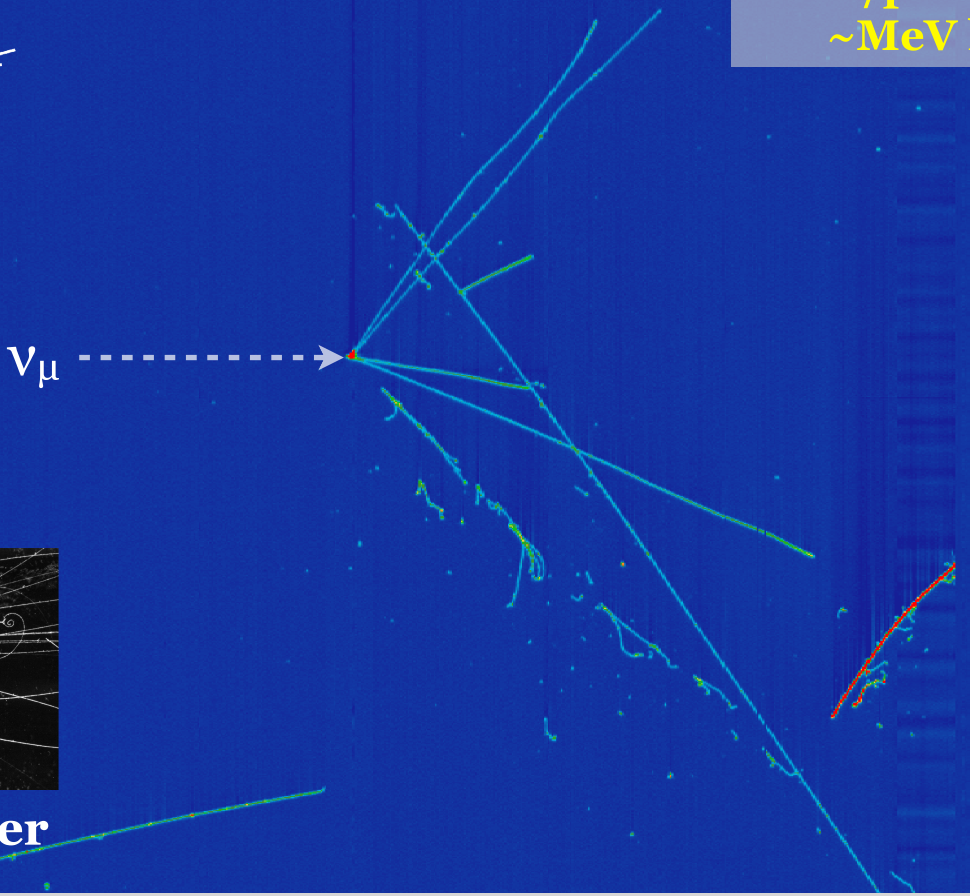
**Outline**

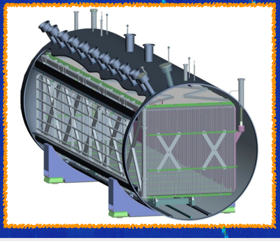
1. Neutrino Experiments & Detectors
2. **Liquid Argon Time Projection Chambers**
3. Machine Learning & Computer Vision
4. Applications in Data Reconstruction
5. Wrap-up

# Next Neutrino Detectors?

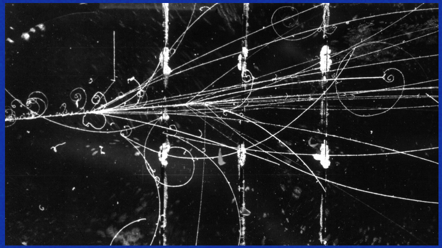
μBooNE

~mm/pixel spatial resolution
~MeV level sensitivity



MicroBooNE
~87 ton (school bus size)



Bubble Chamber

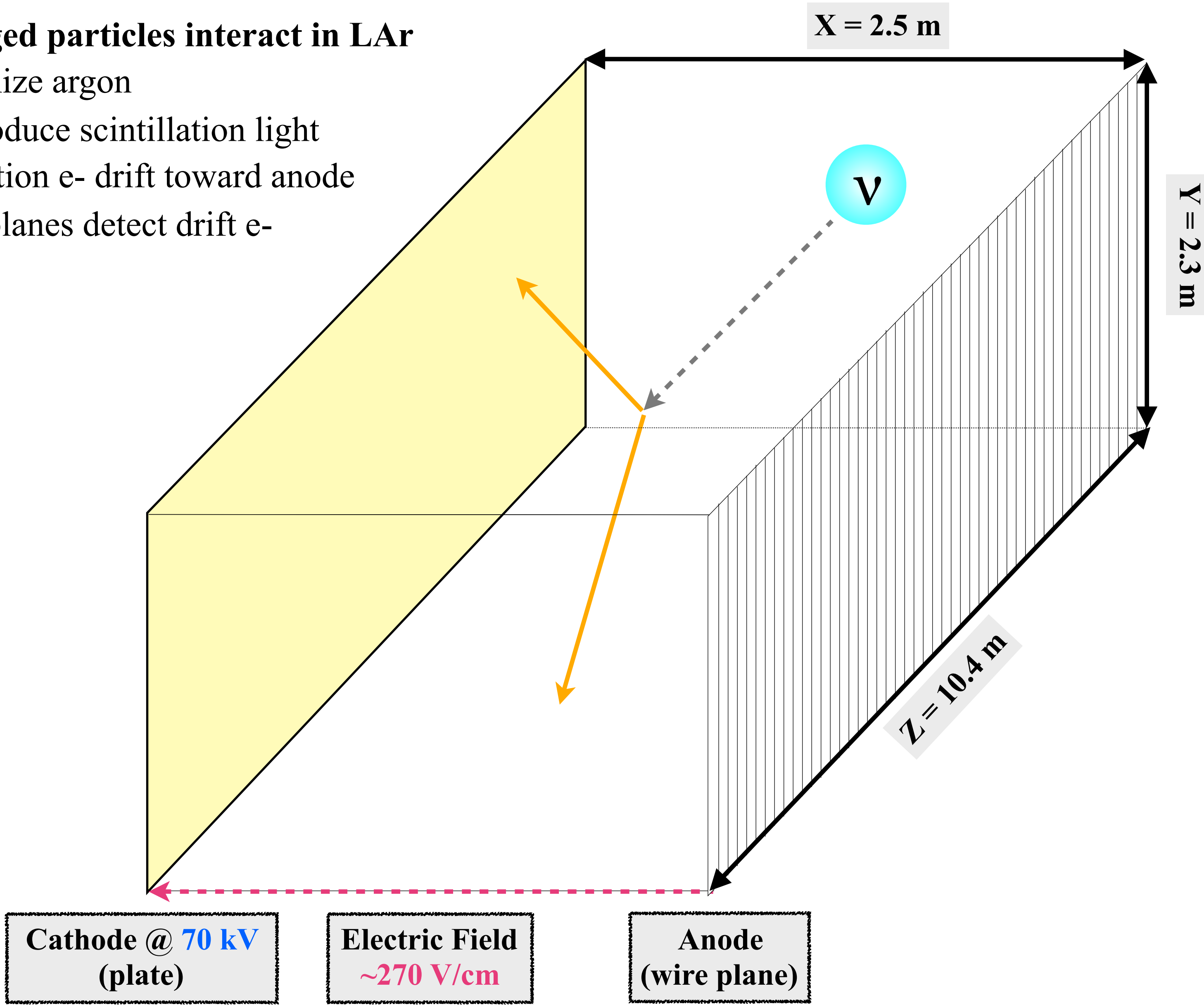
## Liquid Argon Time Projection Chamber

- Chamber-like images: digitized electronics readout
- Calorimetric measurement + scalability to a large mass

2015

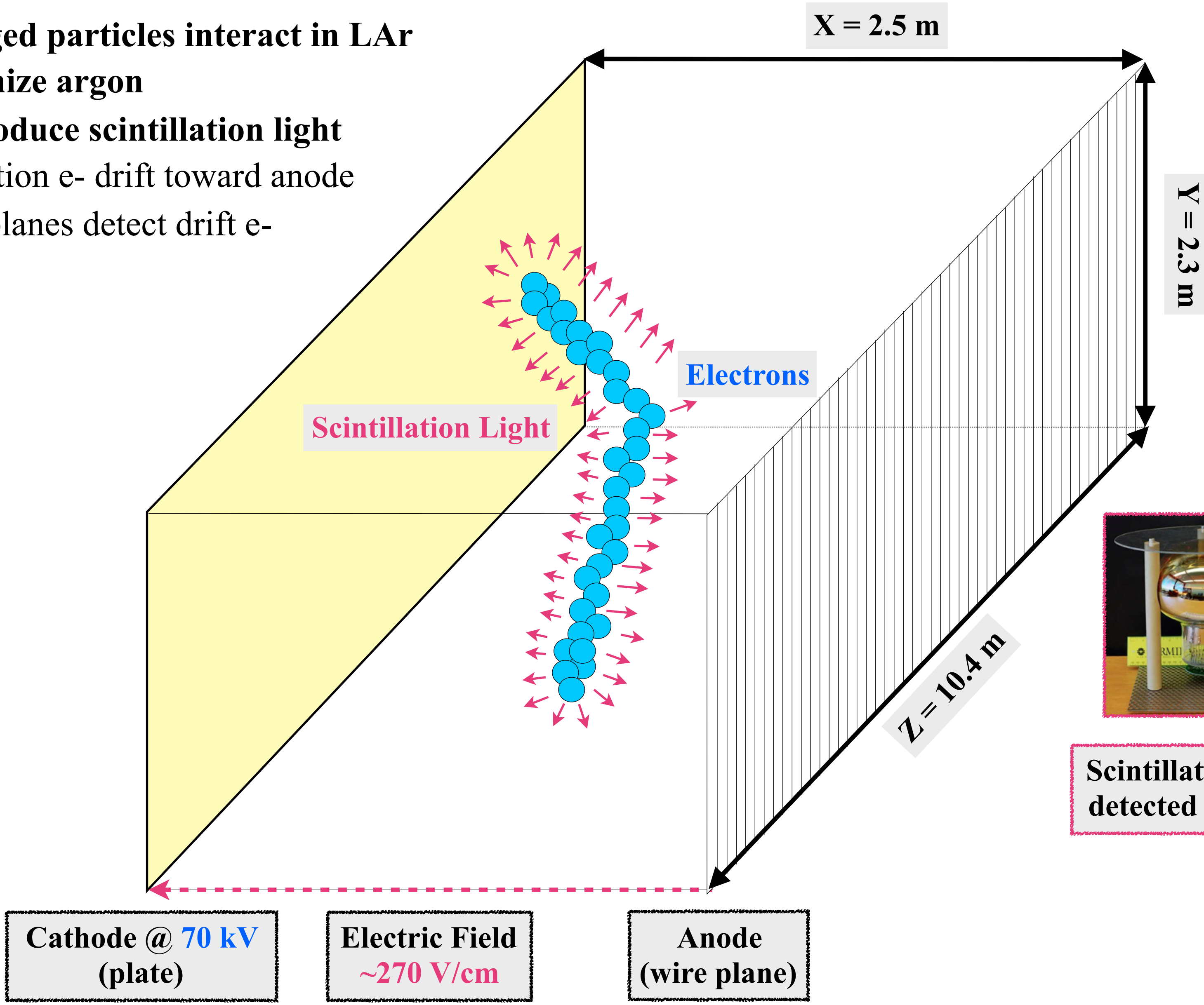
# How Wire LArTPC (MicroBooNE) Work (I)

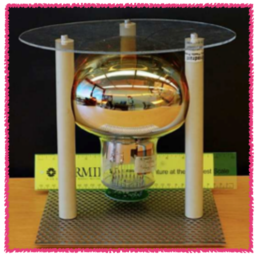
1. **Charged particles interact in LAr**
    - Ionize argon
    - Produce scintillation light
2. Ionization e- drift toward anode
3. Wire planes detect drift e-



# How Wire LArTPC (MicroBooNE) Work (I)

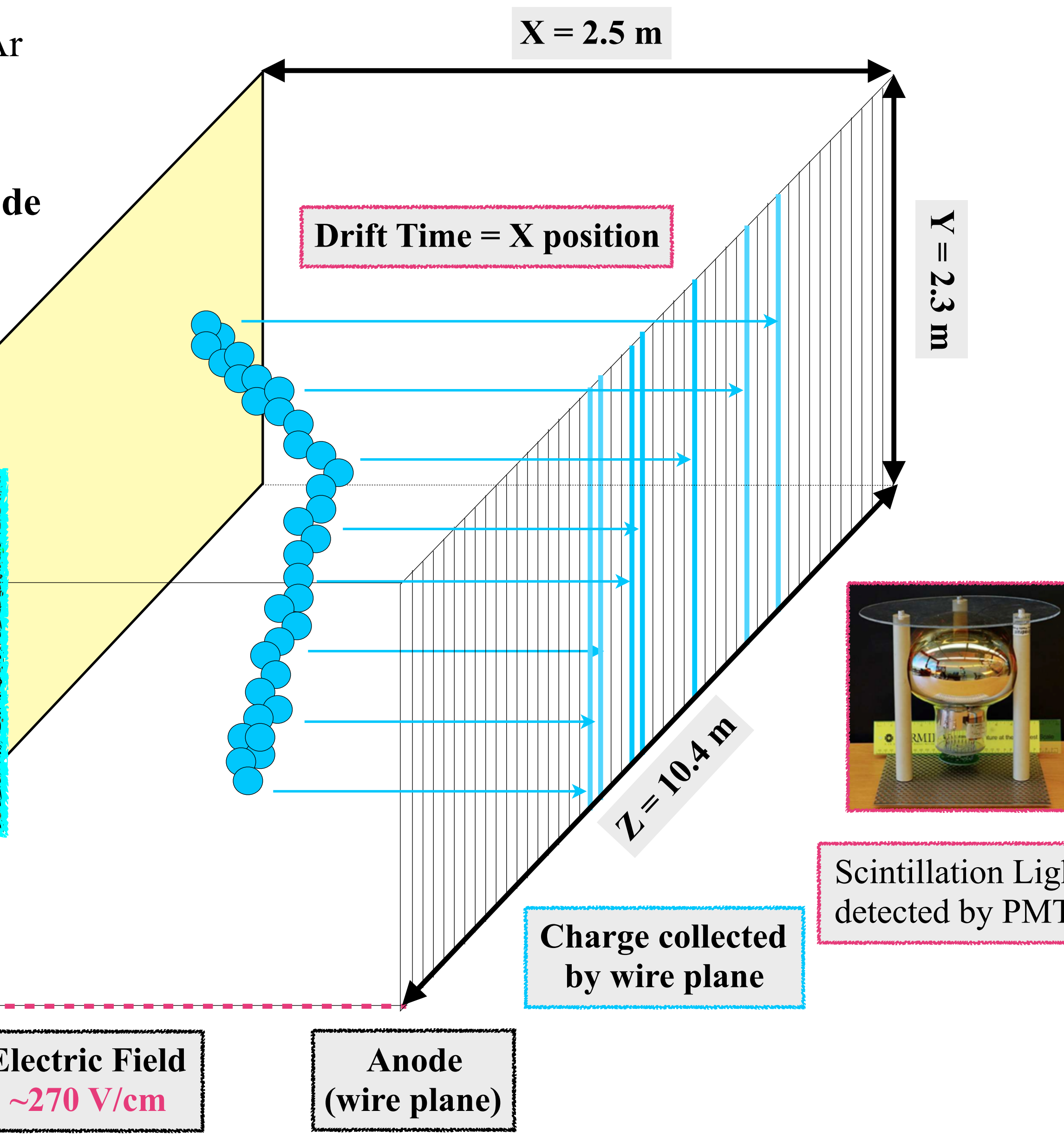
1. **Charged particles interact in LAr**
   - **Ionize argon**
   - **Produce scintillation light**
2. Ionization e- drift toward anode
3. Wire planes detect drift e-



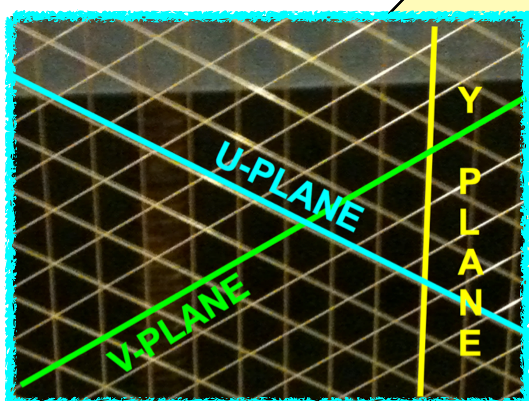



# How Wire LArTPC (MicroBooNE) Work (I)

1. Charged particles interact in LAr
   - Ionize argon
   - Produce scintillation light
2. **Ionization e- drift toward anode**
3. **Wire planes detect drift e-**




**Three Wire Planes**

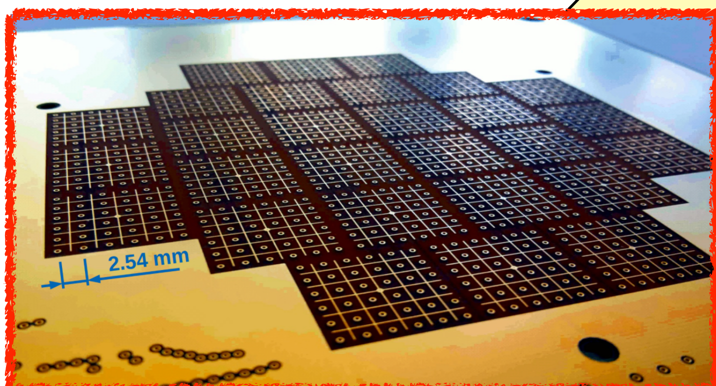



Scintillation Light detected by PMTs

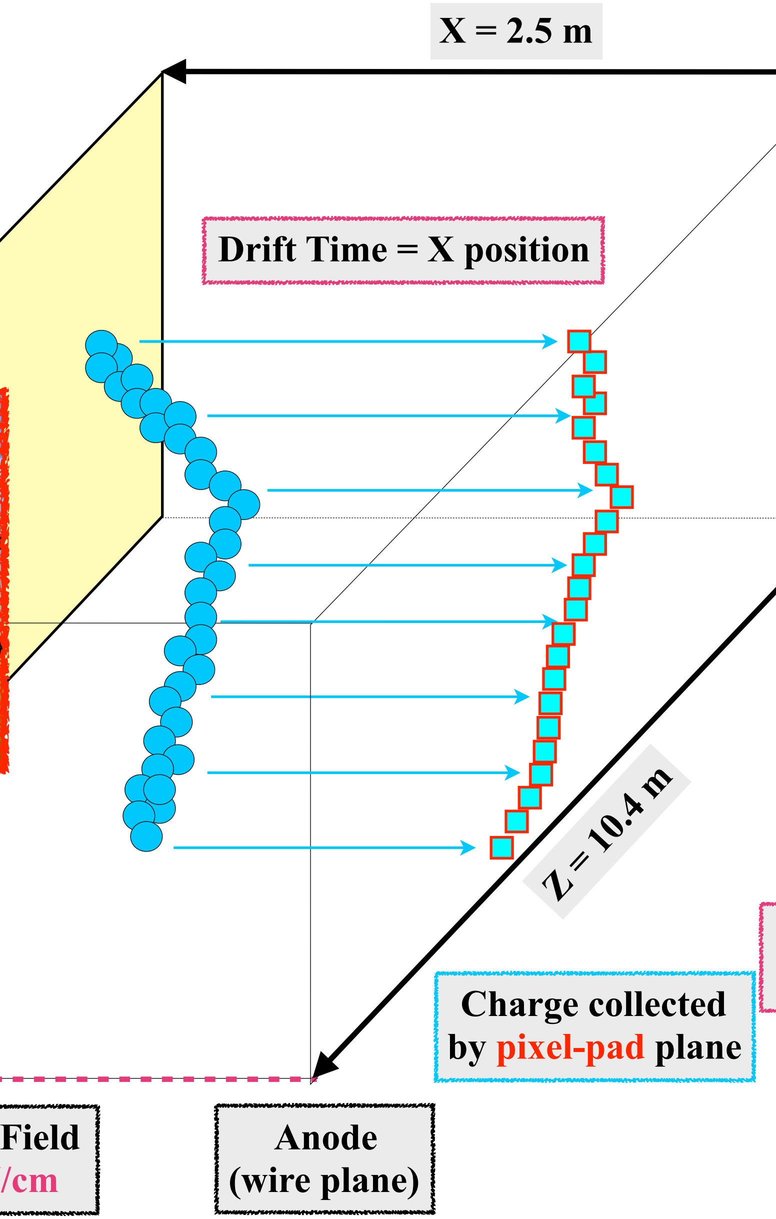
# How ~~Wire~~ Pixel LArTPC (~~MicroBooNE~~ DUNE-ND) Work (I)

1. Charged particles interact in LAr
    - Ionize argon
    - Produce scintillation light
2. **Ionization e- drift toward anode**
3. **~~Wire planes~~ detect drift e-**

**pixel detector**



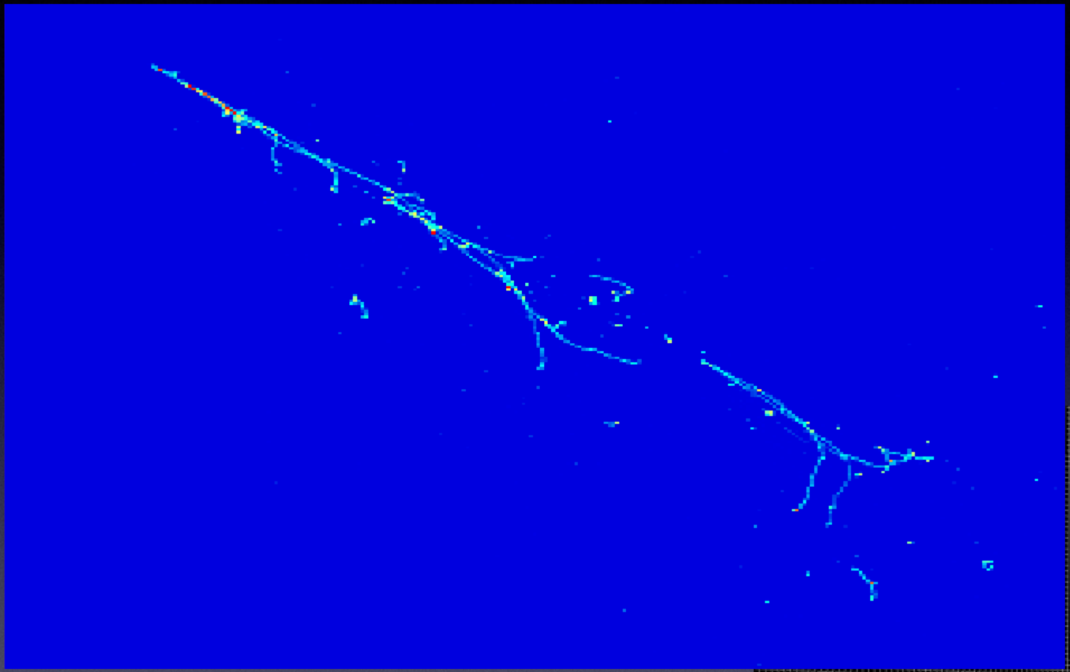

J. Assadi et al. arxiv 1801.08884




Scintillation Light detected by PMTs

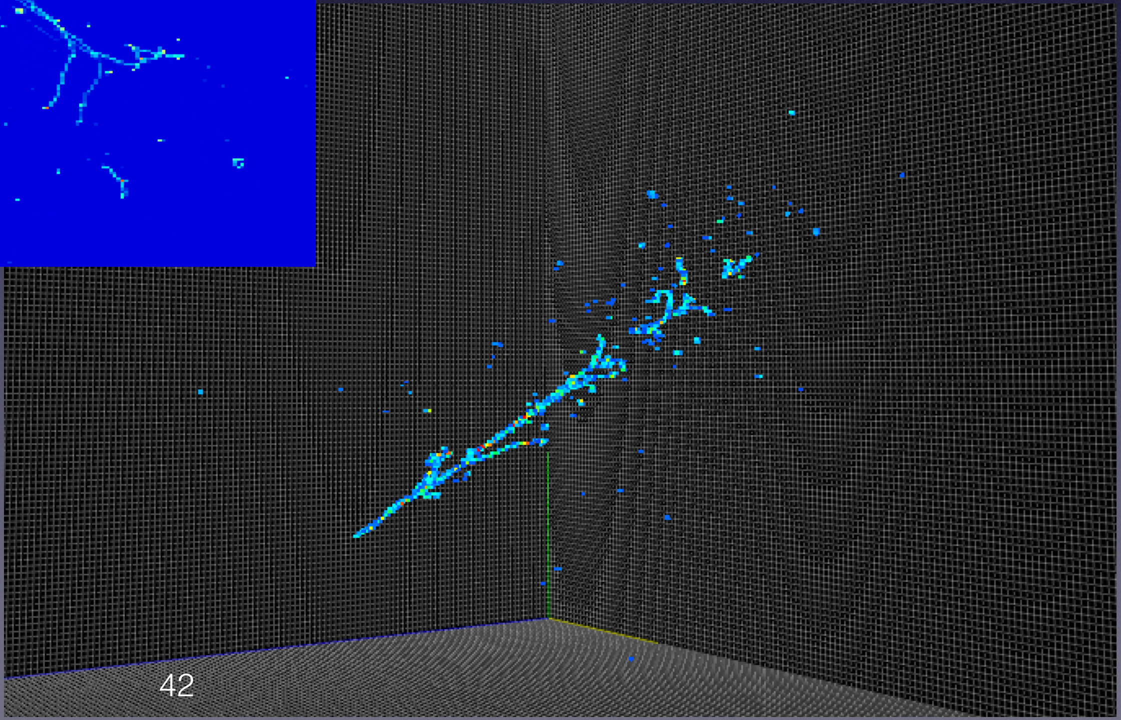
# LArTPC: Particle Imaging Detector

... when things work ...



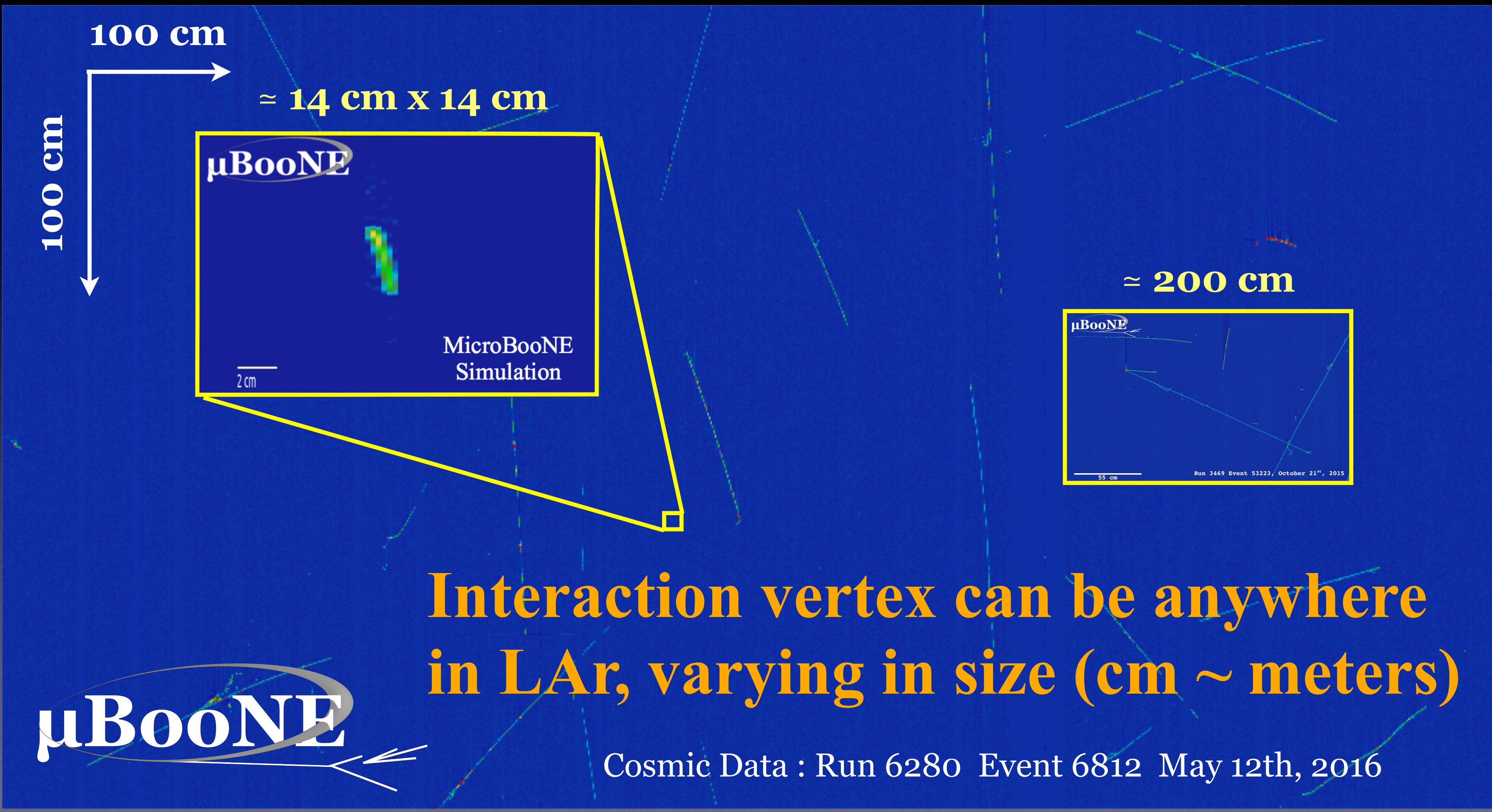
**2D Projection**
(Wire Detector)

**3D Imaging**
(Pixel Detector)



# Challenges in Data Analysis?

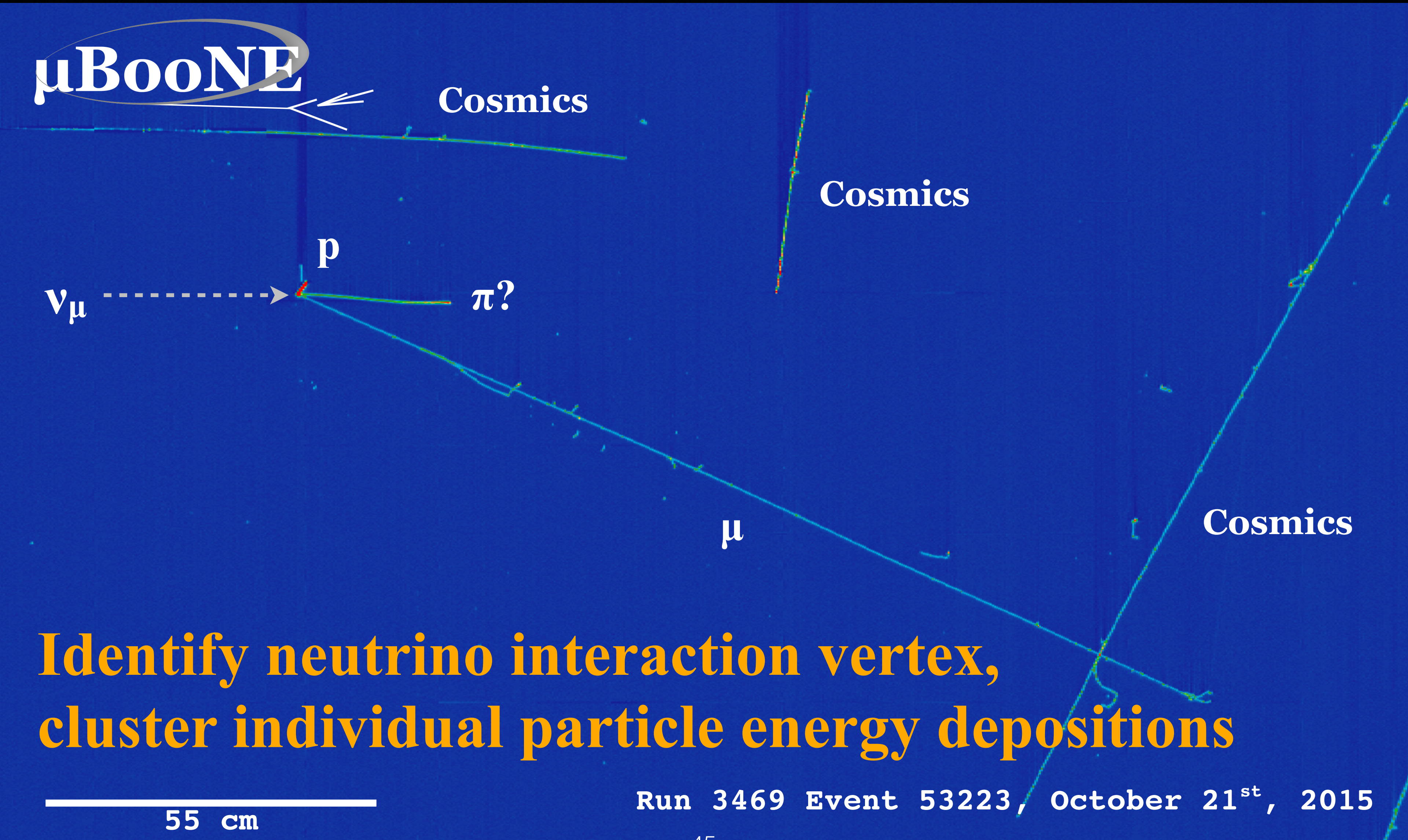
# Challenges in Data Analysis?




**Interaction vertex can be anywhere in LAr, varying in size (cm ~ meters)**

Cosmic Data : Run 6280 Event 6812 May 12th, 2016

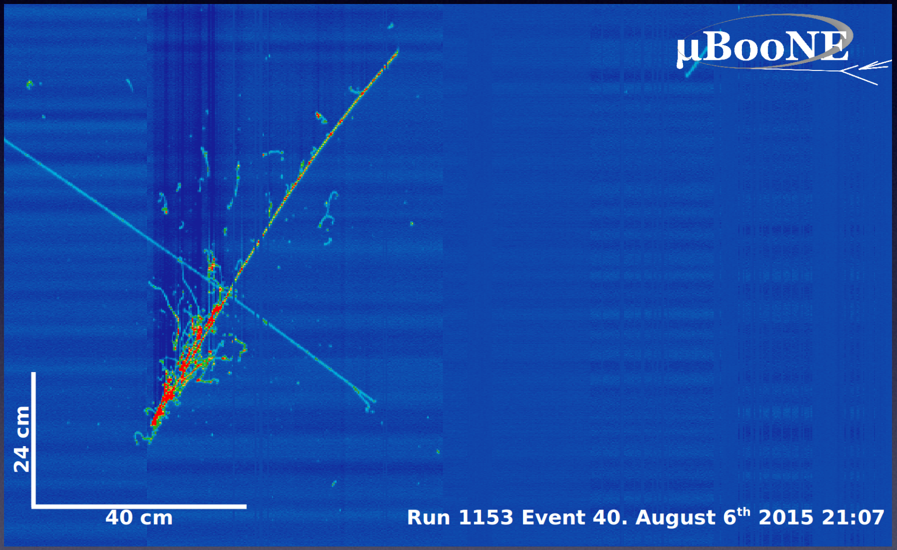
# Challenges in Data Analysis?

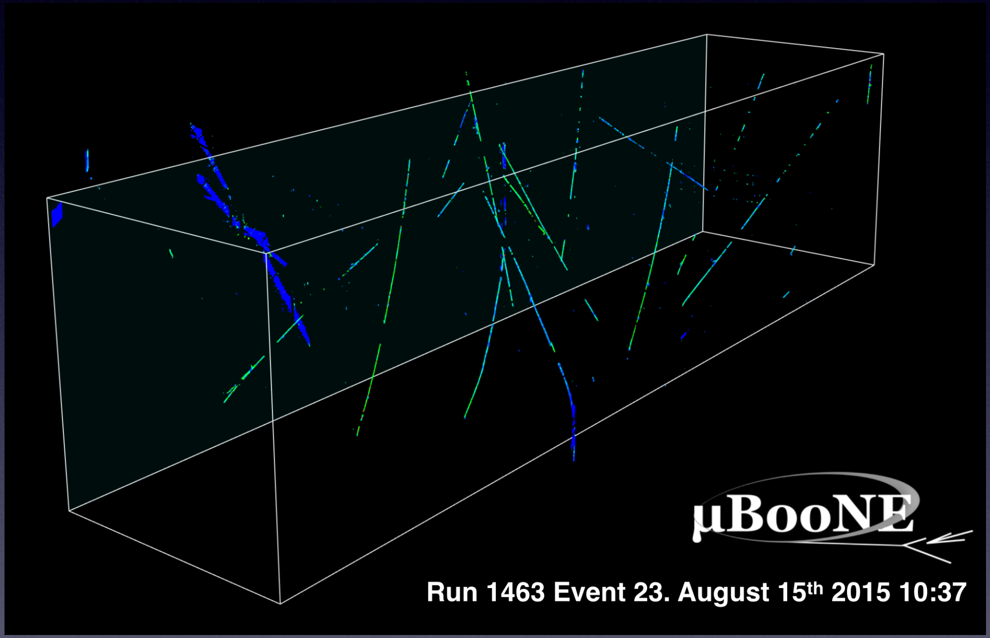



**Identify neutrino interaction vertex, cluster individual particle energy depositions**

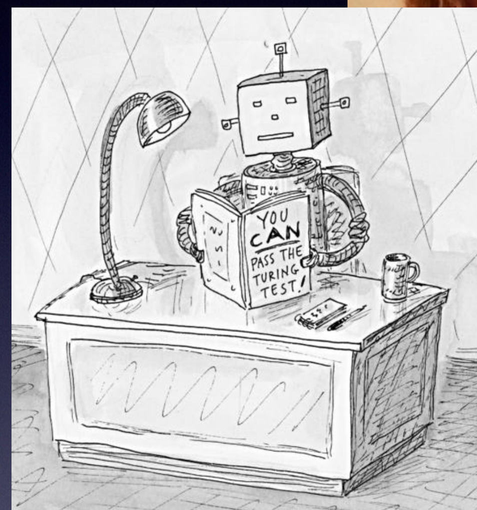
# Challenges in Data Analysis?

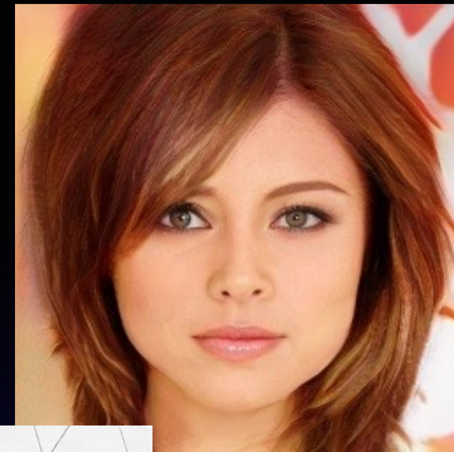
## Deal with optical illusions in 2D projections + pattern recognitions in 3D




"Physics features" look obvious to human physicists (eyes) but hand-engineering algorithms to extract them turned out challenging…



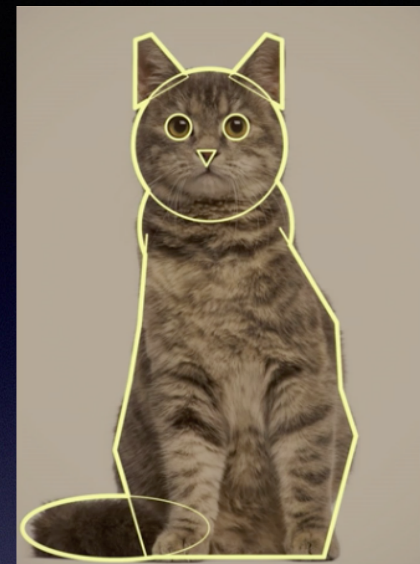



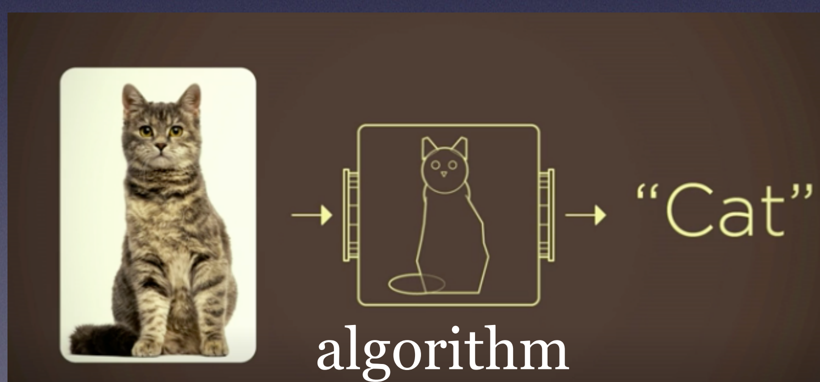
"Fake" celebrity images generated by DNN in 1024 x 1024 resolution

How may I help LArTPCs?

# Recent Innovations in Computer Vision and A.I.

# Classic Problem: Image Categorization






A cat
= collection of
certain shapes



Taken from slides by Fei-Fei's TED talk

# Classic Problem: Image Categorization

## … how about these?

Partial cat
(escaping fiducial volume)

Stretching cat
(DIS?)

Outliers
(axions/dark matter)

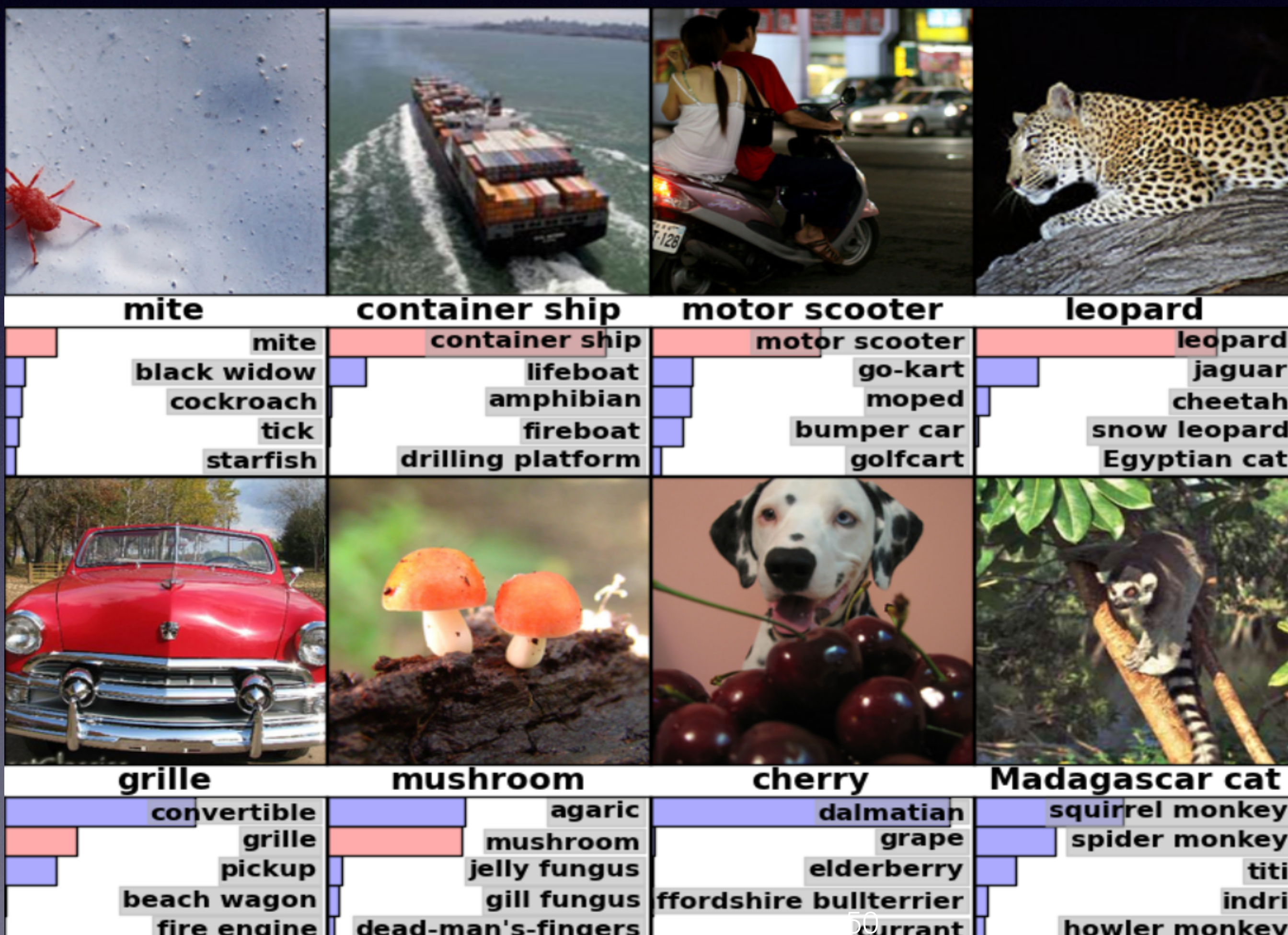


Taken from slides by Fei-Fei's TED talk

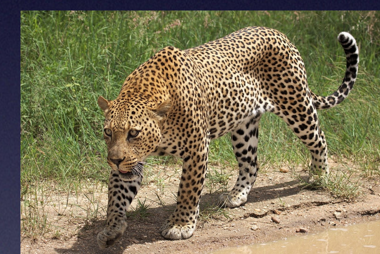
# Breakthrough in Computer Vision in 2012

***AlexNet*: 8-layers deep neural network**
***Birth of "Deep Learning"***

**> 20,000 citations!**



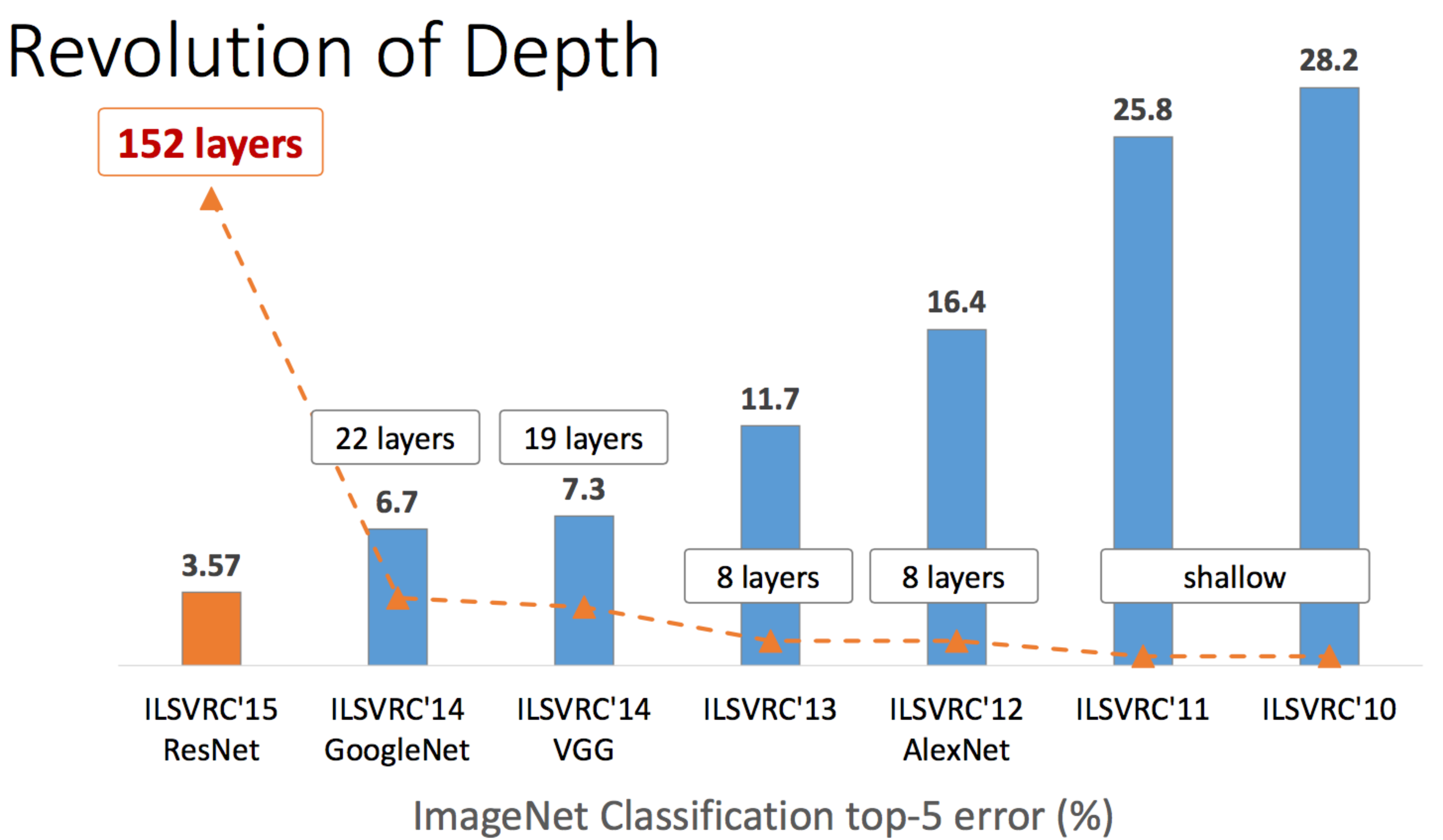

For my reference



Leopard

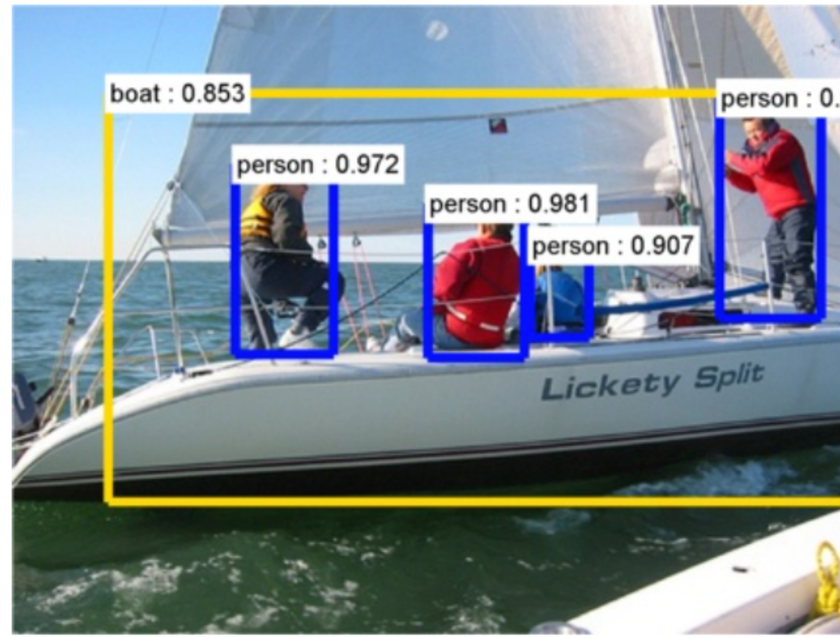
Jaguar

# “Super-human” Performance in 4 years



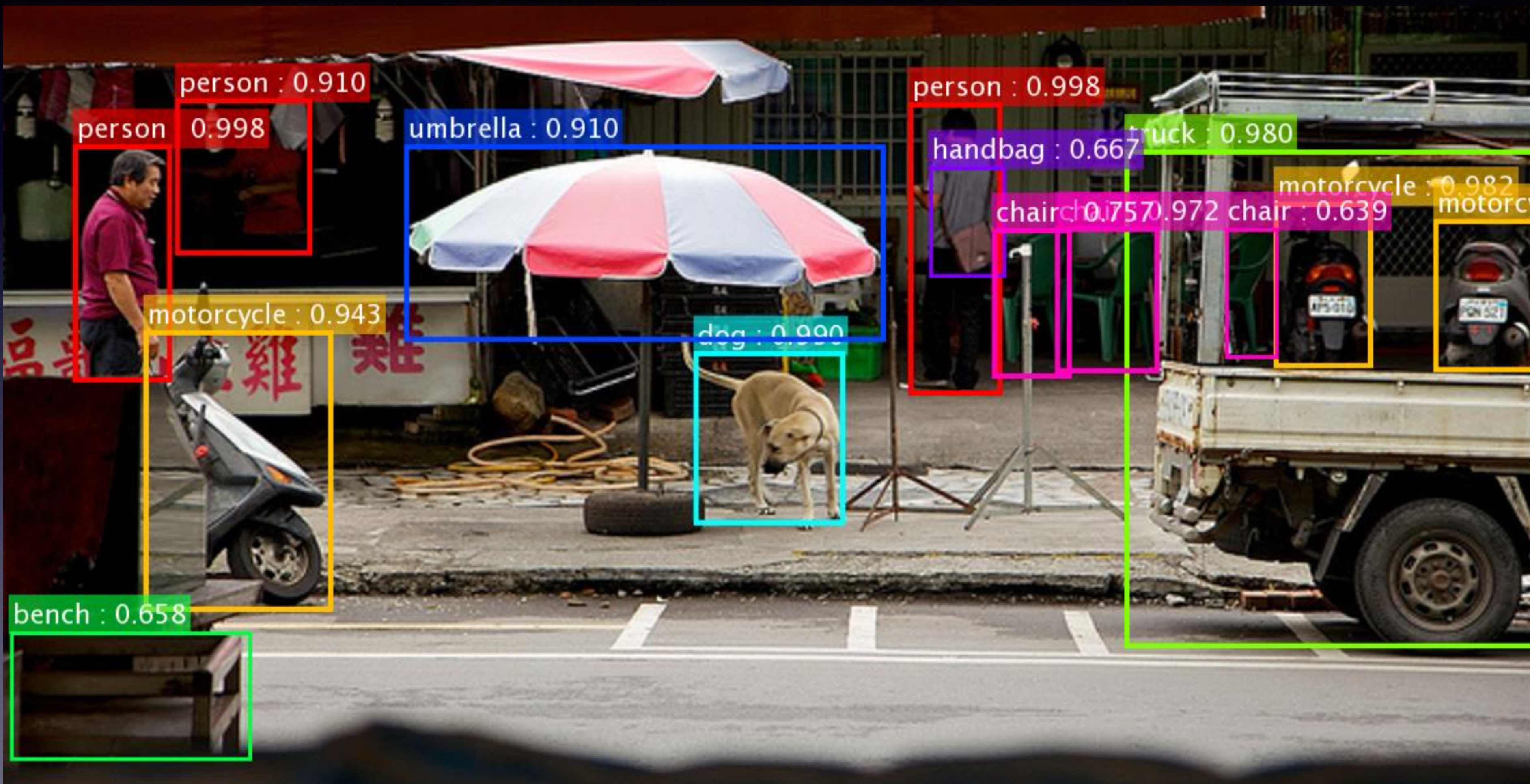
# Beyond Image Classification
# ~ Object Detection ~


Image Classification
(what?)




Object Detection
(what + where?)

# Beyond Image Classification
# ~ Object Detection ~



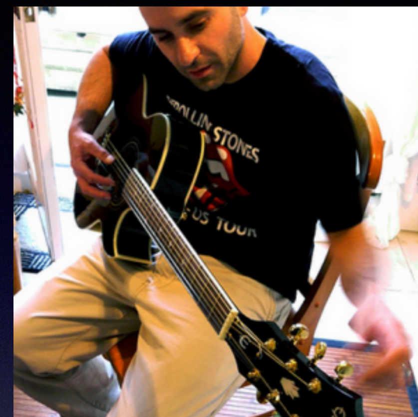
# Beyond Image Classification
# ~ Pixel Segmentation ~




High precision donuts detection

# Beyond Image Classification
# ~ Caption Generation ~

“girl in pink dress is jumping in air”



“man in black shirt is playing guitar”

“black and white dog jumps over bar”

“construction worker in orange safety vest is working on road”

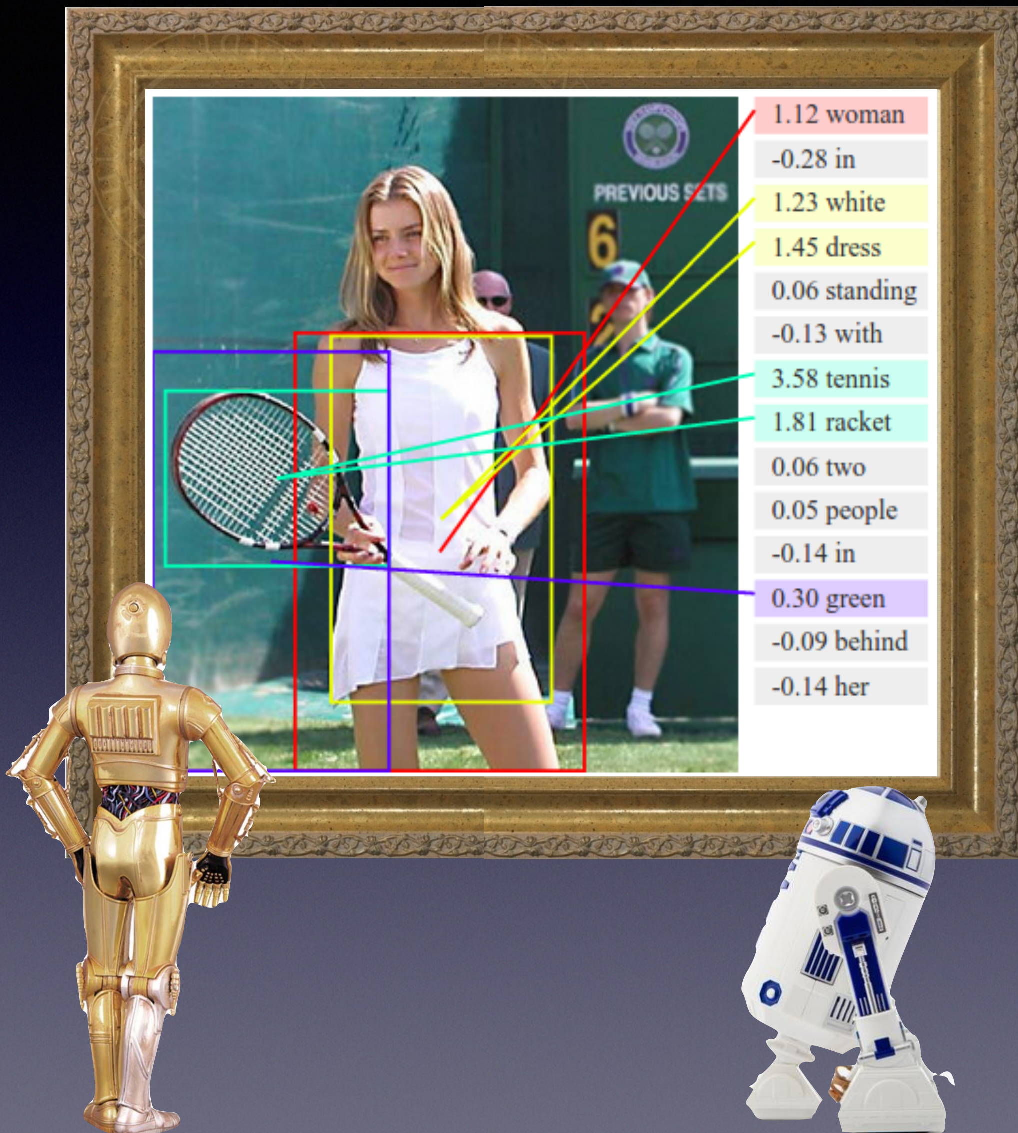
CVPR 2015 A. Karapathy, L. Fei-Fei

# Beyond Image Classification
# ~ Image Generation ~

*"fake celebrity"*
Photo-realistic human face generation

ICLR 2018 T. Karras, T. Aila, S. Laine, J. Lehtinen

## Image context analysis




## "Pose" detection


# Convolutional Neural Network

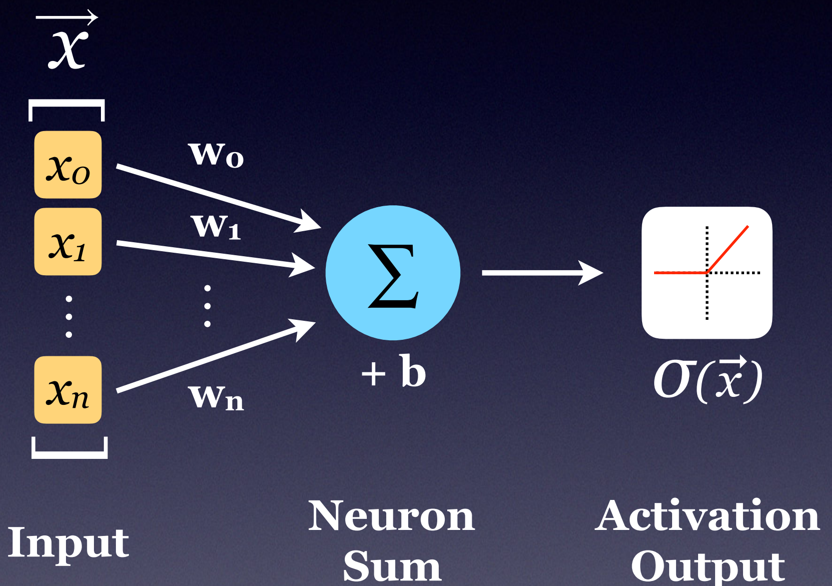
*~ How does it work? ~*

# How a Simple Perceptron Works

## Background: Neural Net

The basic unit of a neural net is the ***perceptron*** (loosely based on a real neuron)

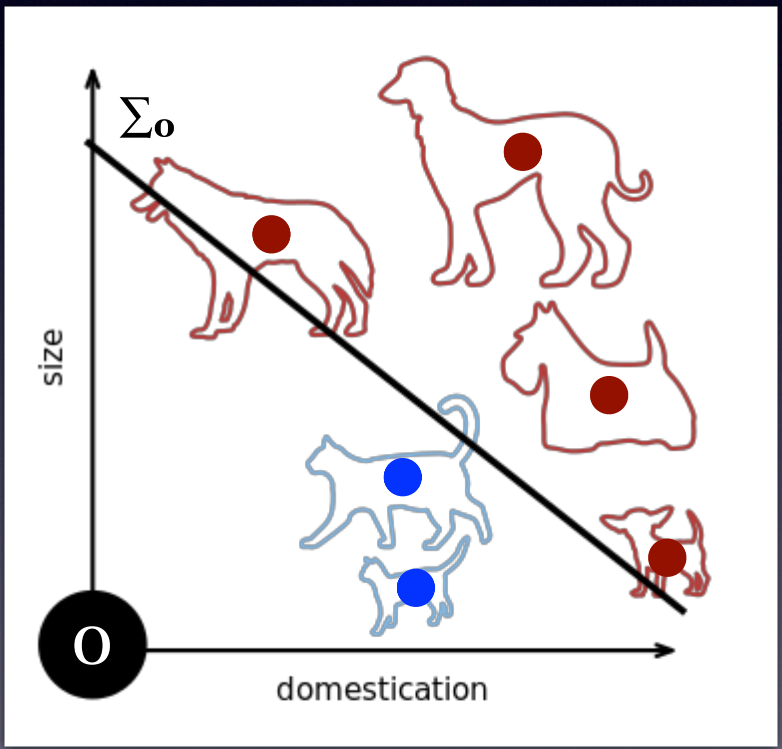
Takes in a vector of inputs ($x$). Commonly inputs are summed with weights ($w$) and offset ($b$) then run through activation.




$$\sigma(\vec{x}) = \begin{cases} \vec{w}_i \cdot \vec{x} + b_i & \vec{w}_i \cdot \vec{x} + b_i \geq 0 \\ 0 & \vec{w}_i \cdot \vec{x} + b_i < 0. \end{cases}$$

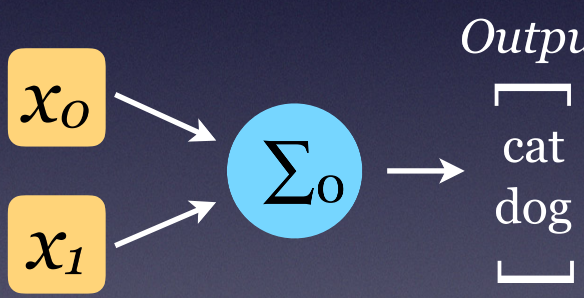
# How a Simple Perceptron Works
## Perceptron 2D Classification

Imagine using two features to separate cats and dogs




from [wikipedia]

$$\sigma(\vec{x}) = \begin{cases} \vec{w}_i \cdot \vec{x} + b_i & \vec{w}_i \cdot \vec{x} + b_i \geq 0 \\ 0 & \vec{w}_i \cdot \vec{x} + b_i < 0. \end{cases}$$

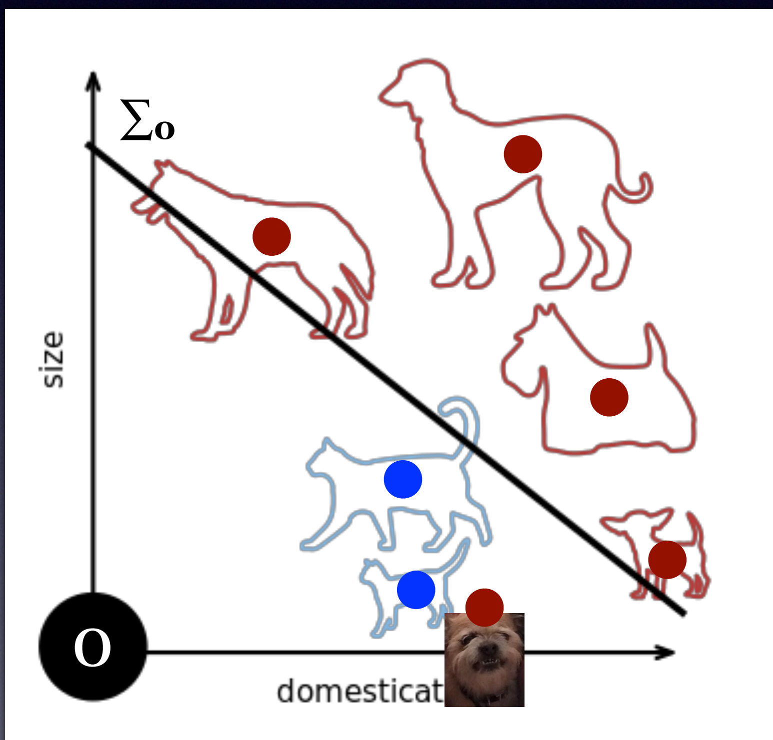



By picking a value for w and b,
we define a boundary
between the two sets of data

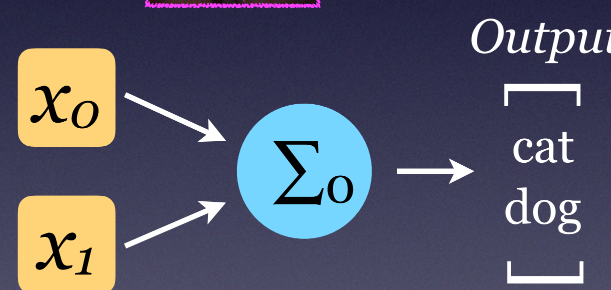
# How a Simple Perceptron Works

# Perceptron 2D Classification

Maybe we need to do better: assume a new data point (small but not as well behaved)



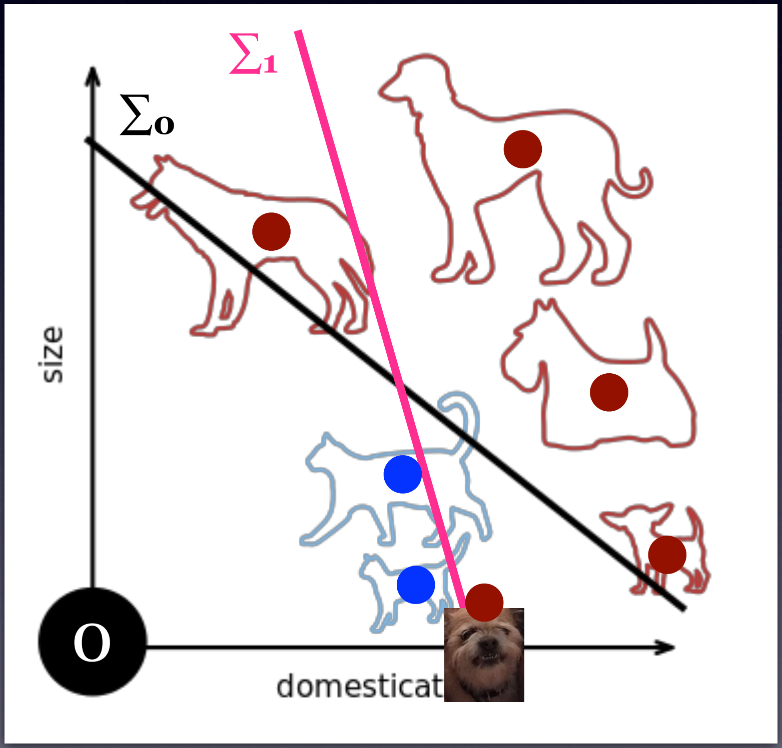

from [wikipedia]



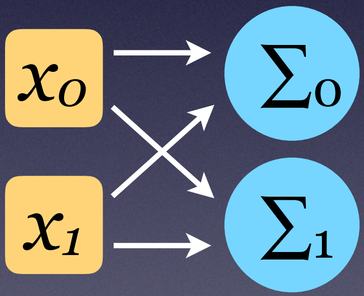
# How a Simple Perceptron Works

# Perceptron 2D Classification

Maybe we need to do better: assume a new data point (small but not as well behaved)




from [wikipedia]

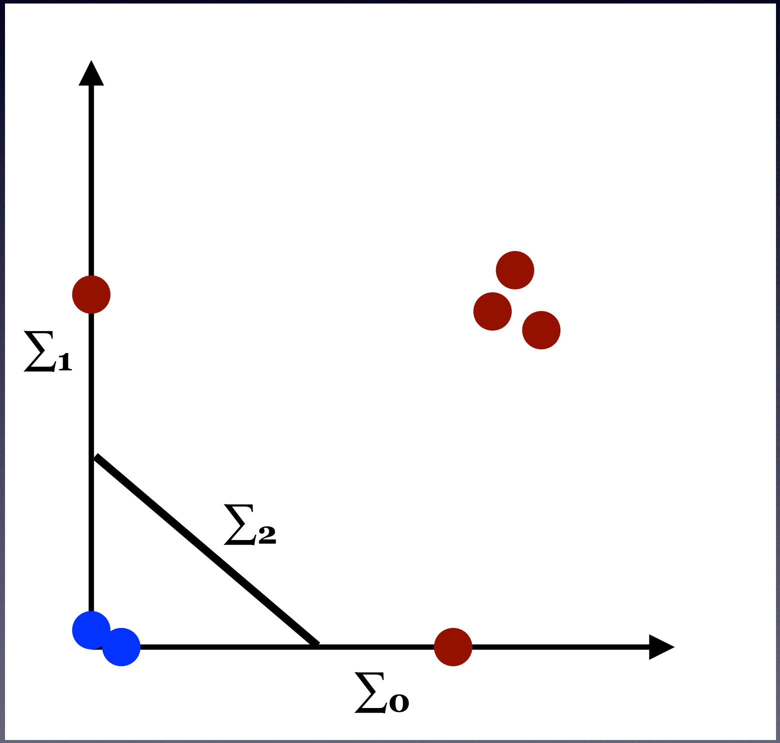


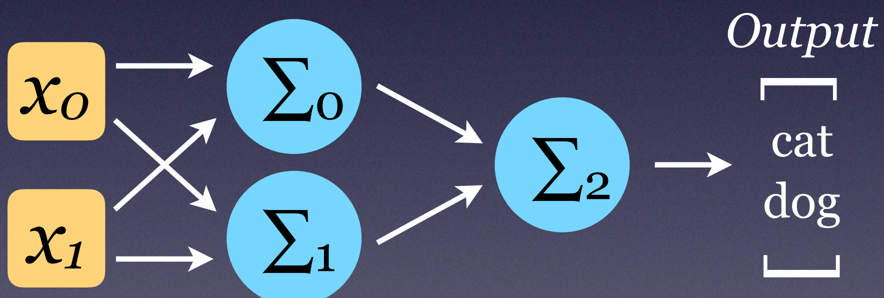

We can add another perceptron to help (but does not yet solve the problem)

# How a Simple Perceptron Works

## Perceptron 2D Classification

Maybe we need to do better: assume a new data point (small but not as well behaved)

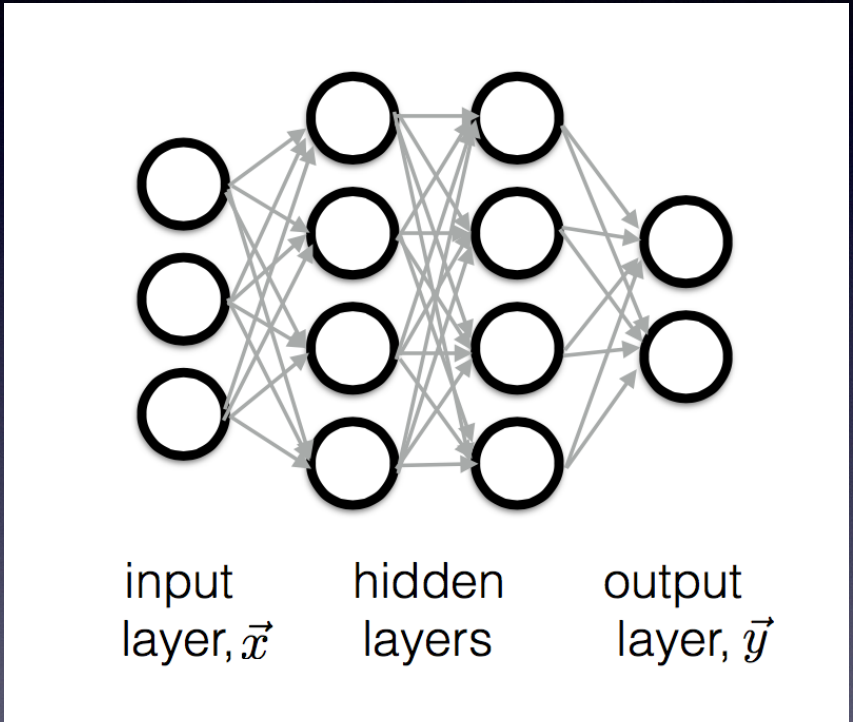



Another layer can classify based on preceding feature layer output

# “Classical” Neural Net

## **Fully-Connected, Feed-forward, Multi-Layer Perceptrons**



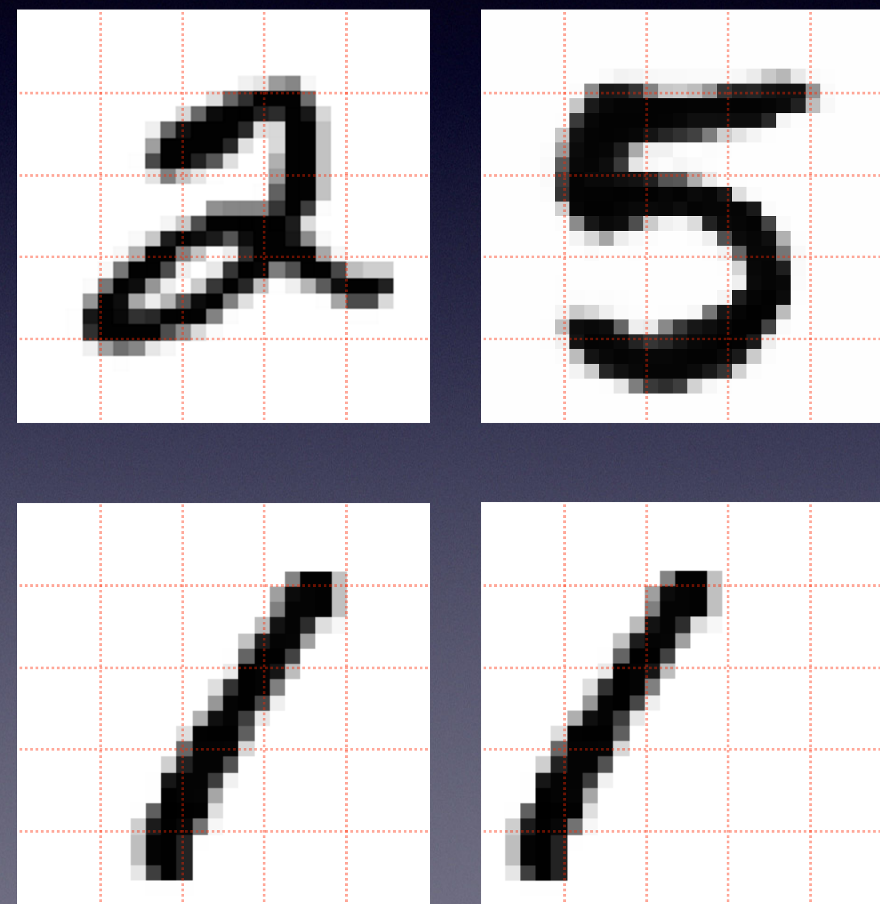

A traditional neural network consists of a stack of layers of such neurons where each neuron is ***fully connected*** to other neurons of the neighbor layers

# “Classical” Neural Net

## … is not ideal for image classification …

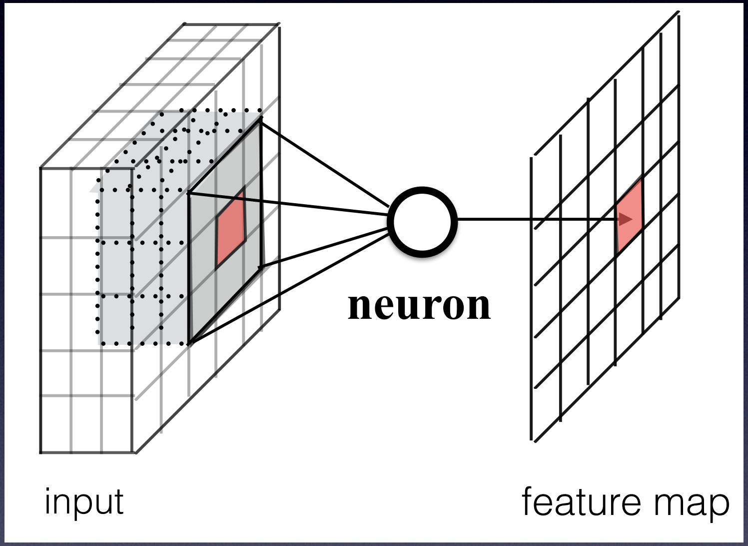
### Image classification

- **What is input neurons?**
  - Every pixel value
- **How many weights?**
  - # of pixels in an image!
- **Fully connected?**
  - translation variant!



# Convolutional Neural Networks

CNN introduce a ***limitation*** by forcing the network to look at only local, translation invariant features

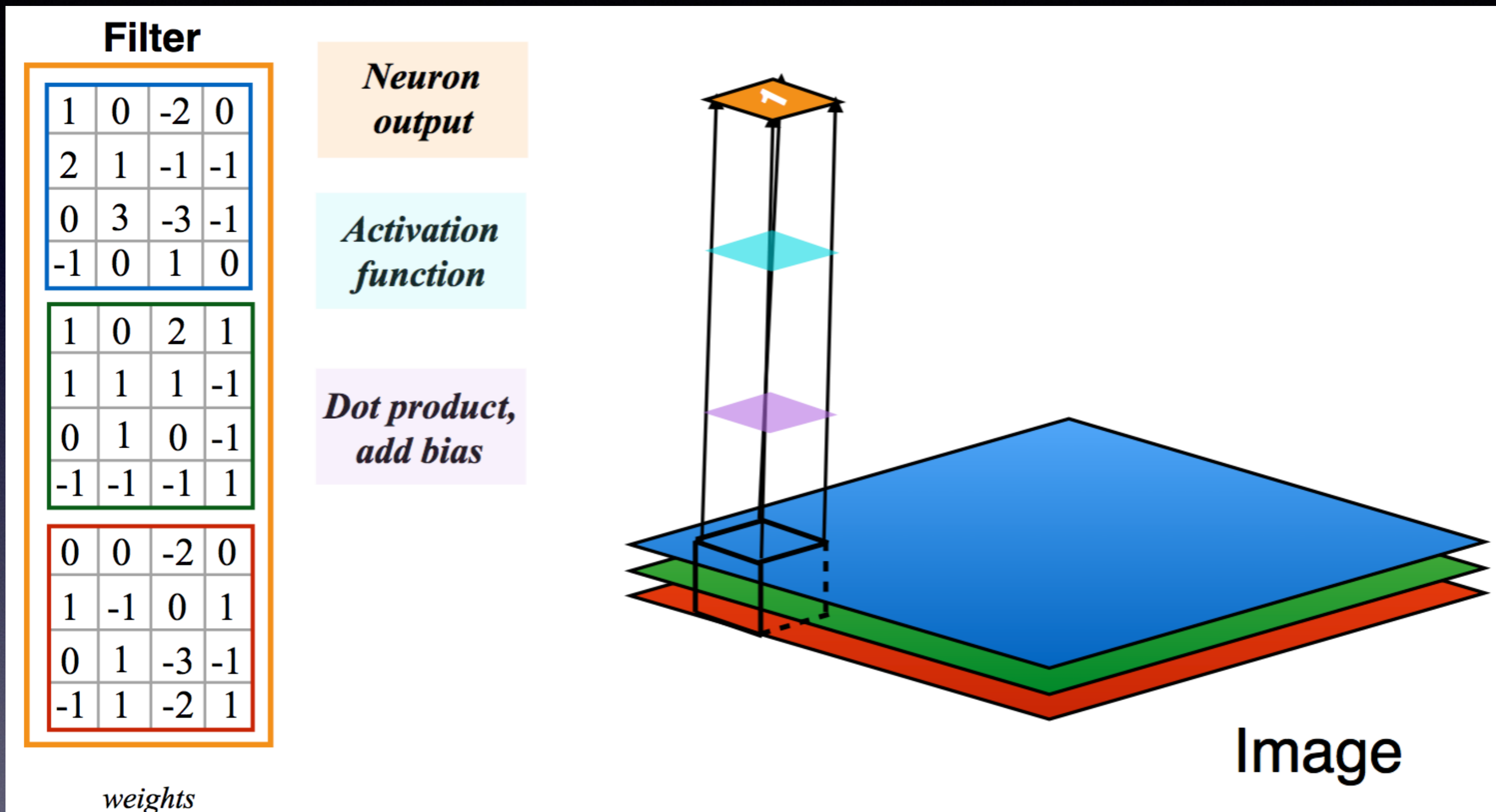



$$f_{i,j}(X) = \sigma\left(W_i \cdot X_j + b_i\right),$$

Activation of a neuron depends on the element-wise product of 3D weight tensor with 3D input data and a bias term

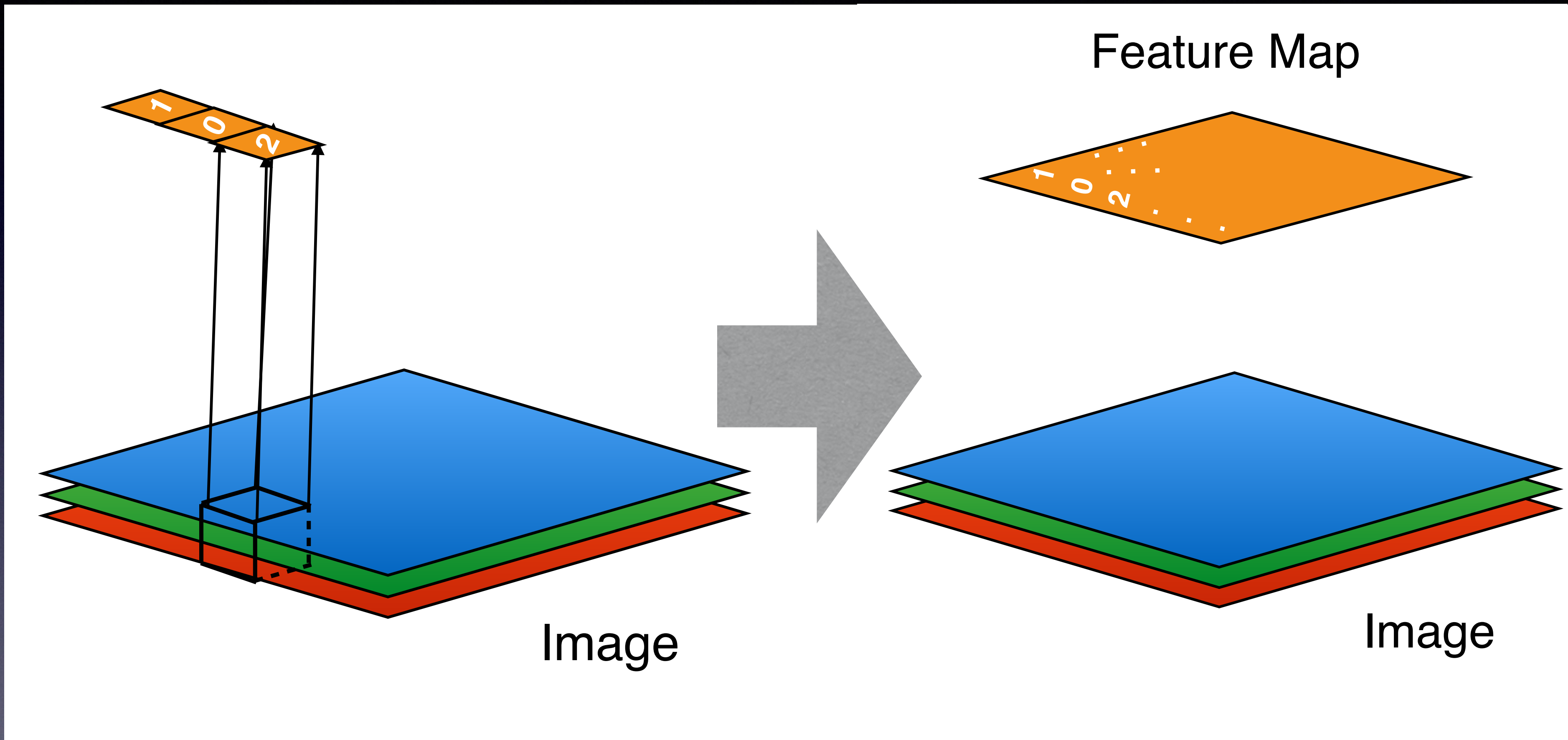
- Translate over 2D space to process the whole input
- Neuron **learns translation-invariant features**
  - Suited for a "***homogeneous***" detector like LArTPC
- **Output**: a "feature-enhanced" image (***feature map***)

# Convolutional Neural Networks



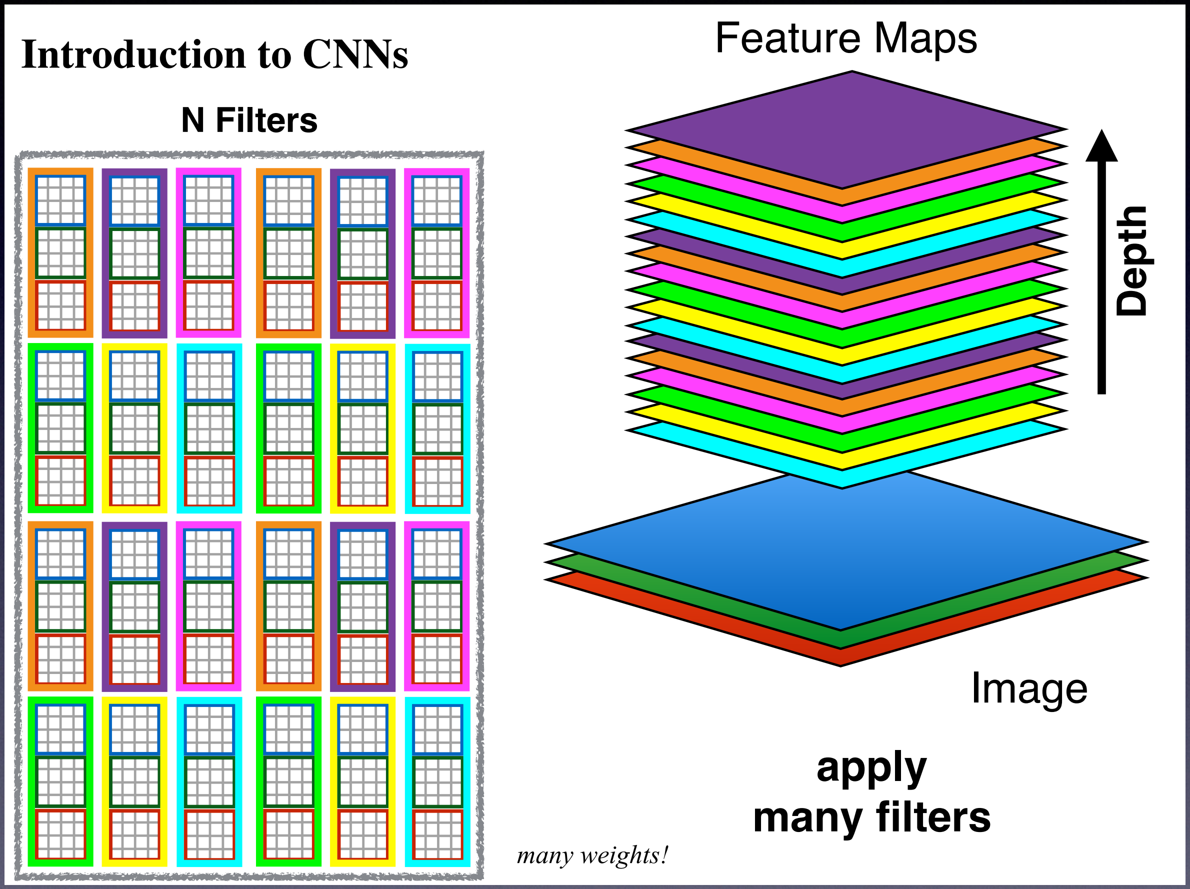

Toy visualization of the CNN operation

# Convolutional Neural Networks




Toy visualization of the CNN operation

# Convolutional Neural Networks

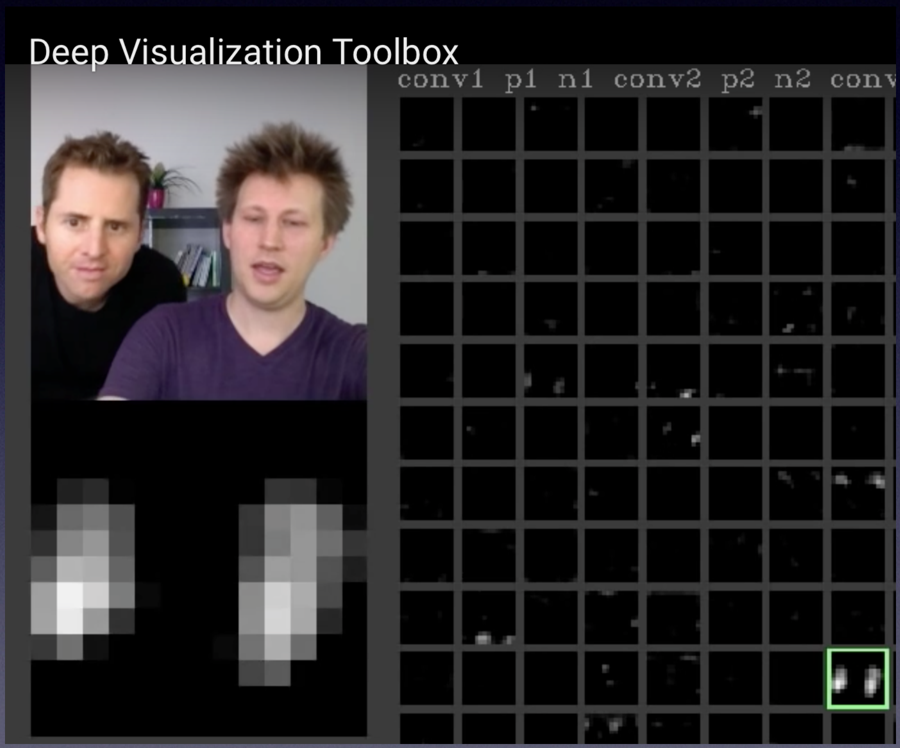


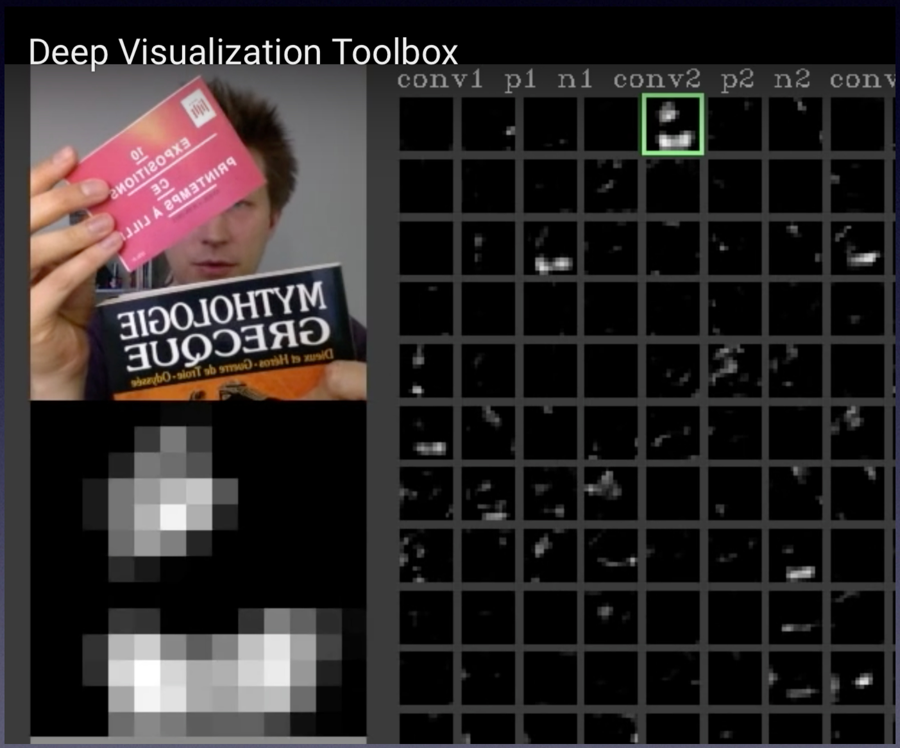

Toy visualization of the CNN operation

# How Image Classification Networks Work

## Feature map visualization example

- https://www.youtube.com/watch?v=AgkfIQ4IGaM




Neuron concerning face

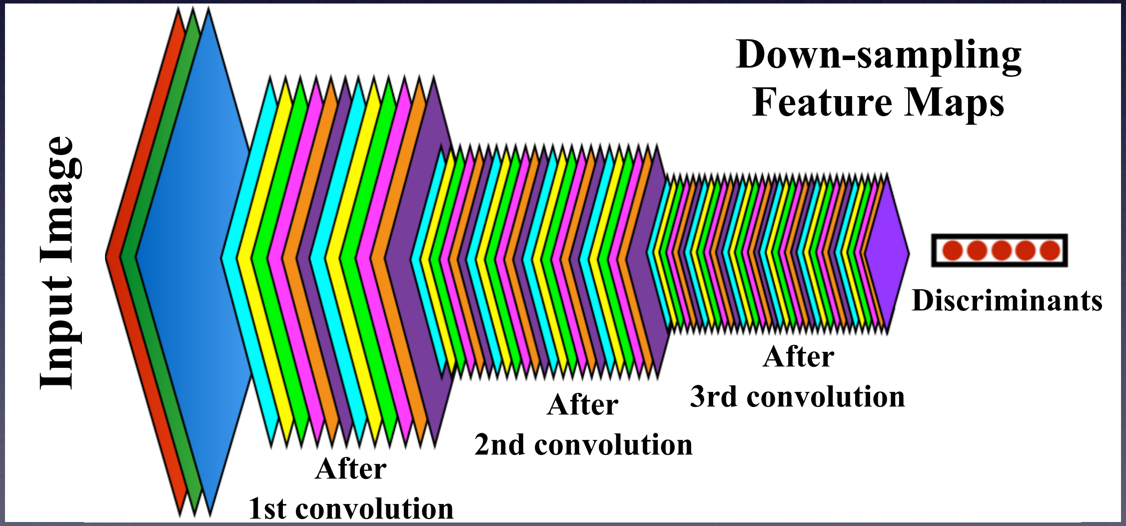
Neuron loving texts

(and don't care about your face)

# How Image Classification Networks Work

**Goal**: extract features to give "single label" to an image

1. **Convolution operation**
2. **Down-sampling**

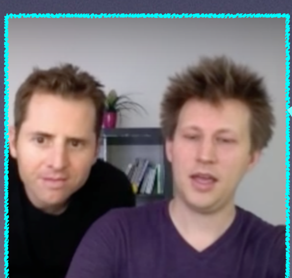


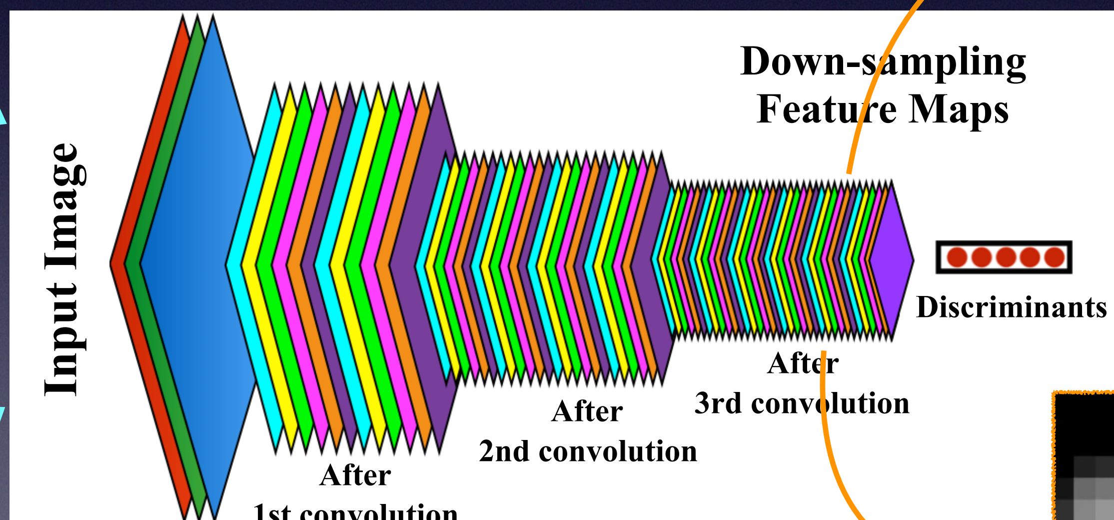

Series of convolutions
+ down-sampling

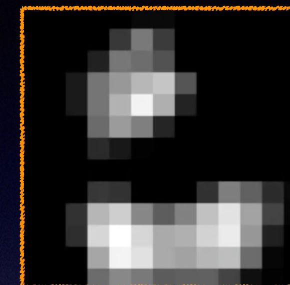
# How Image Classification Networks Work

**Goal**: extract features to give "single label" to an image

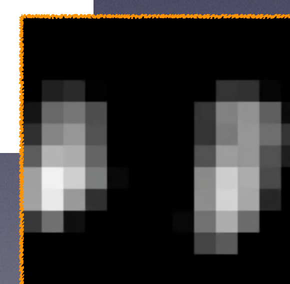
1. **Convolution operation**
2. **Down-sampling**





"Written Texts"
feature map



"Human Face"
feature map

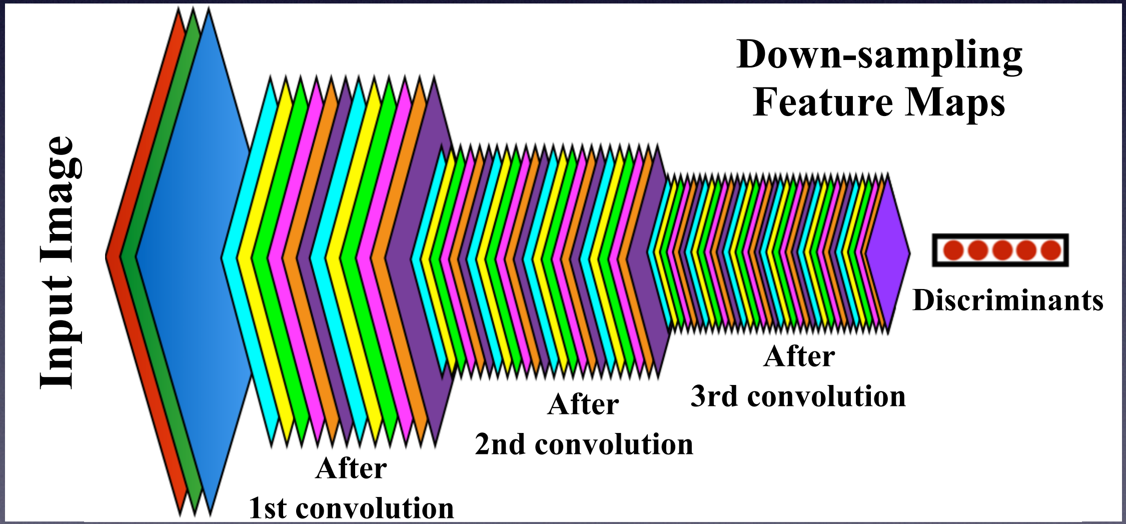
Series of convolutions
+ down-sampling

Image credit: DeepVis @ youtube

# How Image Classification Networks Work

**Goal**: extract features to give "single label" to an image

1. **Convolution operation**
2. **Down-sampling**

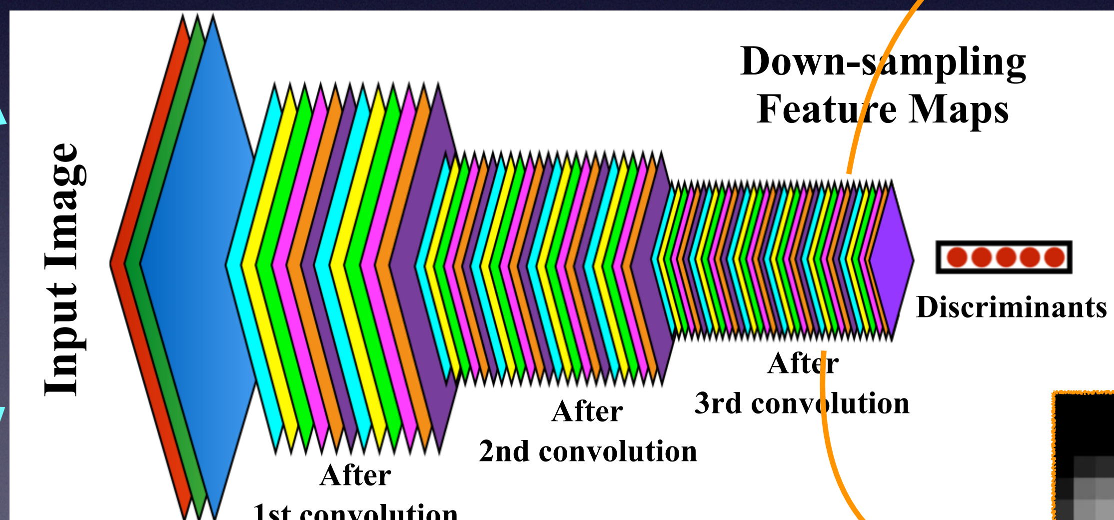


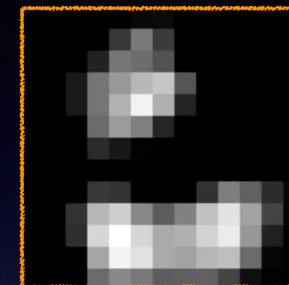

Series of convolutions
+ down-sampling

# How Image Classification Networks Work

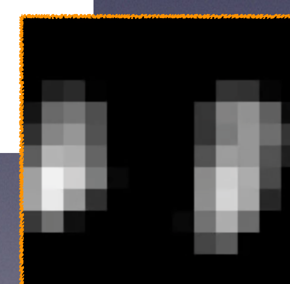
**Goal**: extract features to give "single label" to an image

1. **Convolution operation**
2. **Down-sampling**



"Written Texts"
feature map



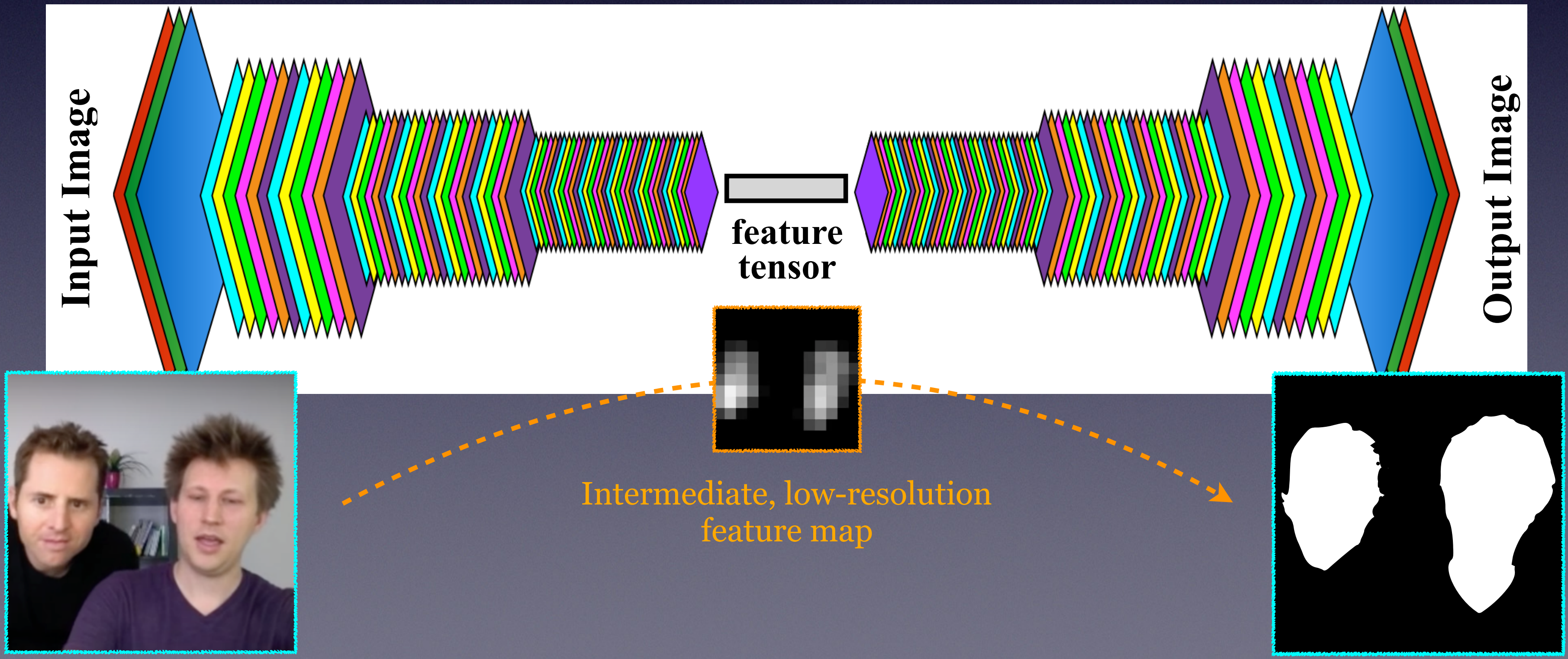
"Human Face"
feature map

Series of convolutions
+ down-sampling

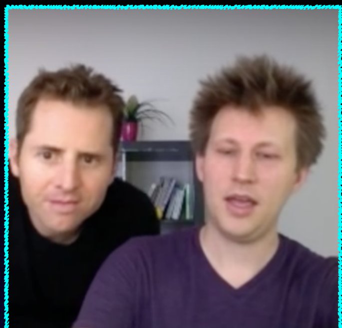
# How SSNet Works

**Goal**: recover precise, pixel-level location of objects

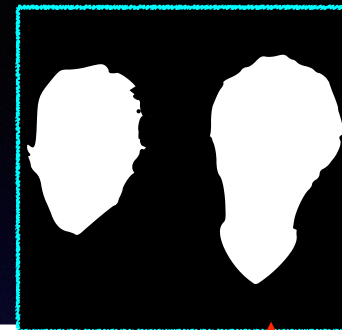
1. **Up-sampling**
   - Expand spatial dimensions of feature maps
2. **Convolution**
   - Smoothing (interpolation) of up-sampled feature maps

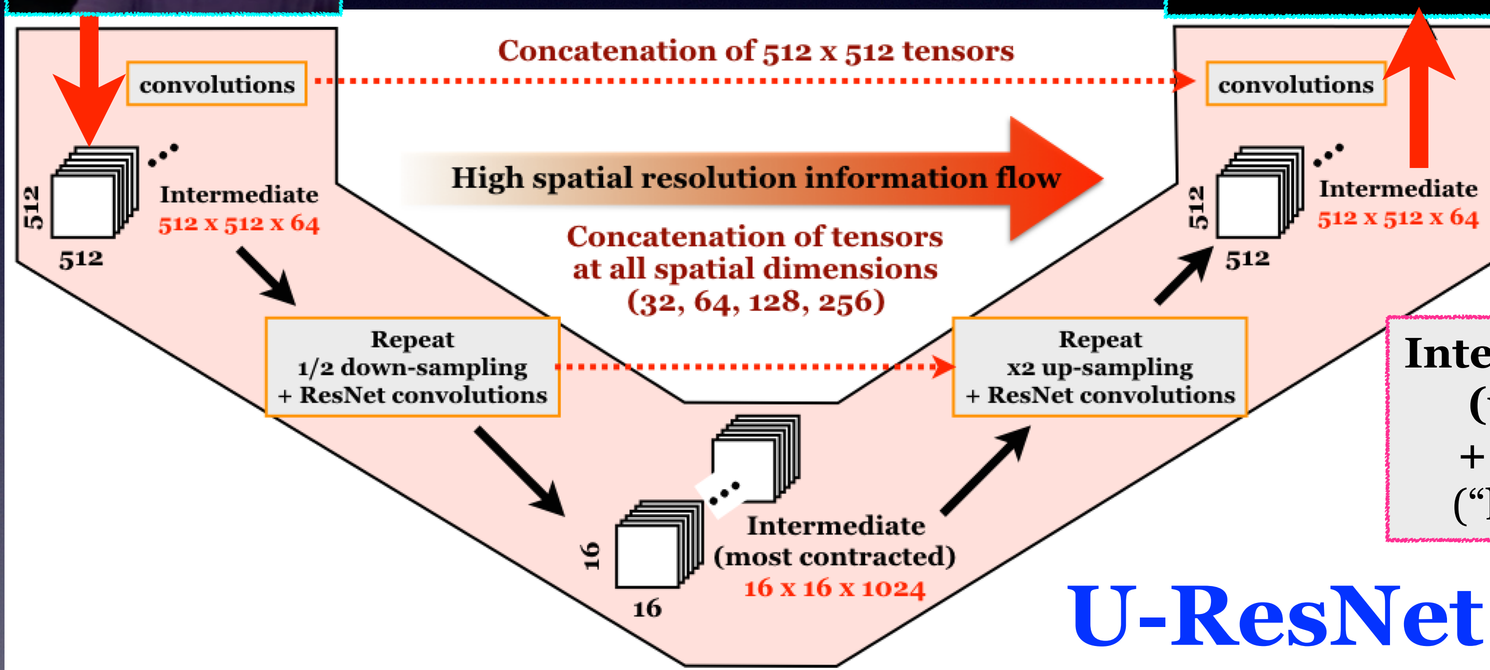


# DNN for LArTPC Data Reconstruction

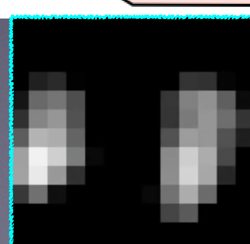


## How does U-ResNet Work?






**Interpolation filters**
**(up-sampling)**
**+ Convolutions**
("learnable" filter)



**Down sampling + Convolutions to identify**
highly abstract features (e.g. "human face")

# Validation with real data

# Benchmarking SSNet w/ Real Data

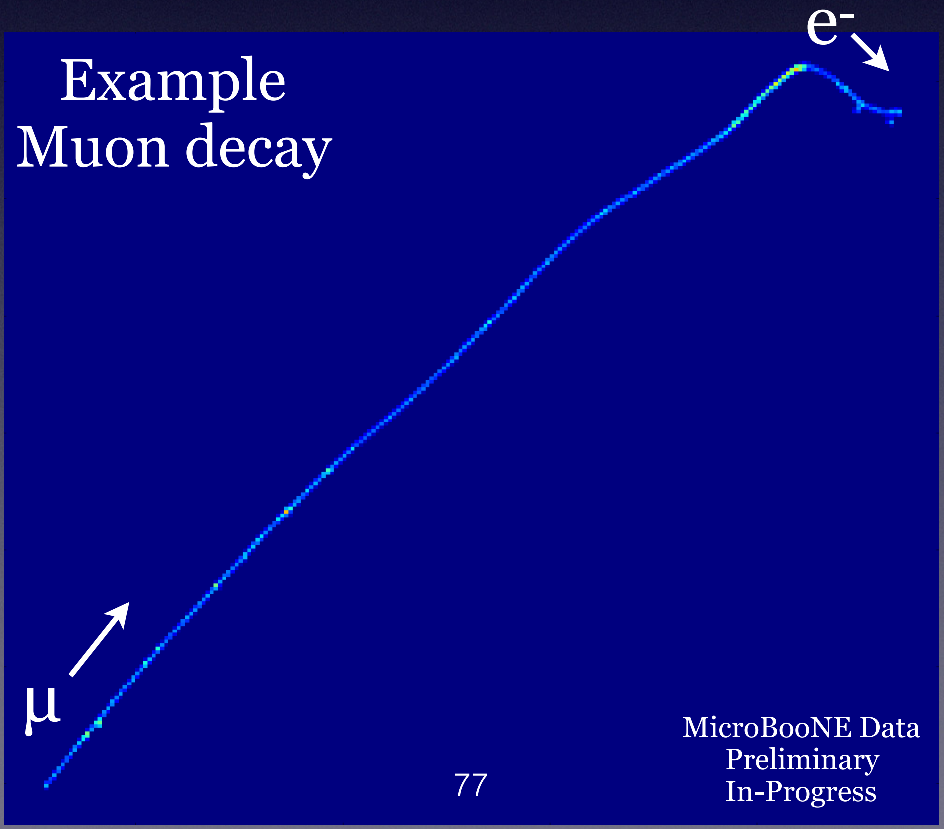
## Samples (100 images per sample per sim/data)

### A cosmic ray muon decay

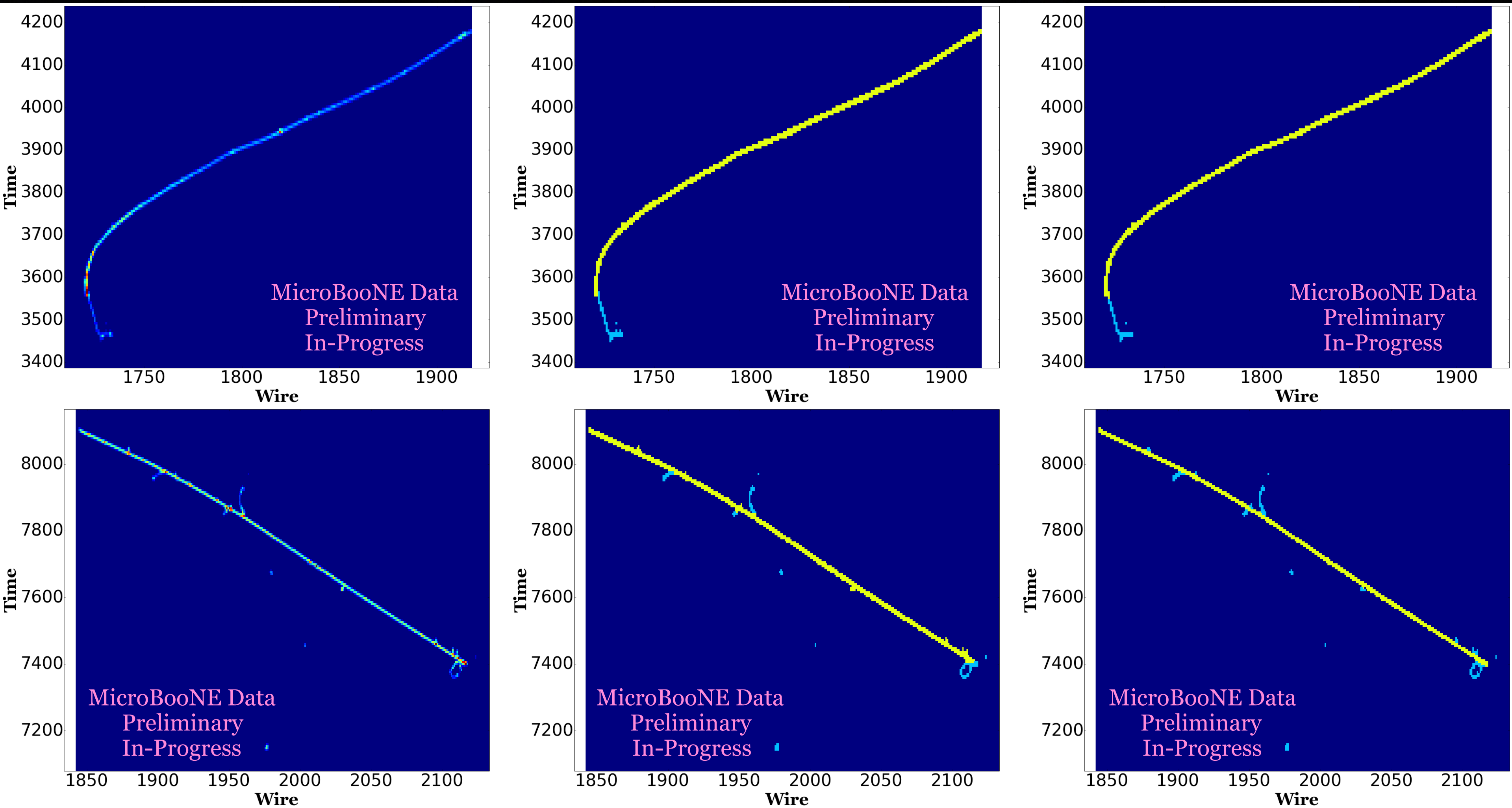
- Involves both "track" and "shower", simple and intuitive.

### Neutrino interactions

- More complicated: varying particle types and multiplicity

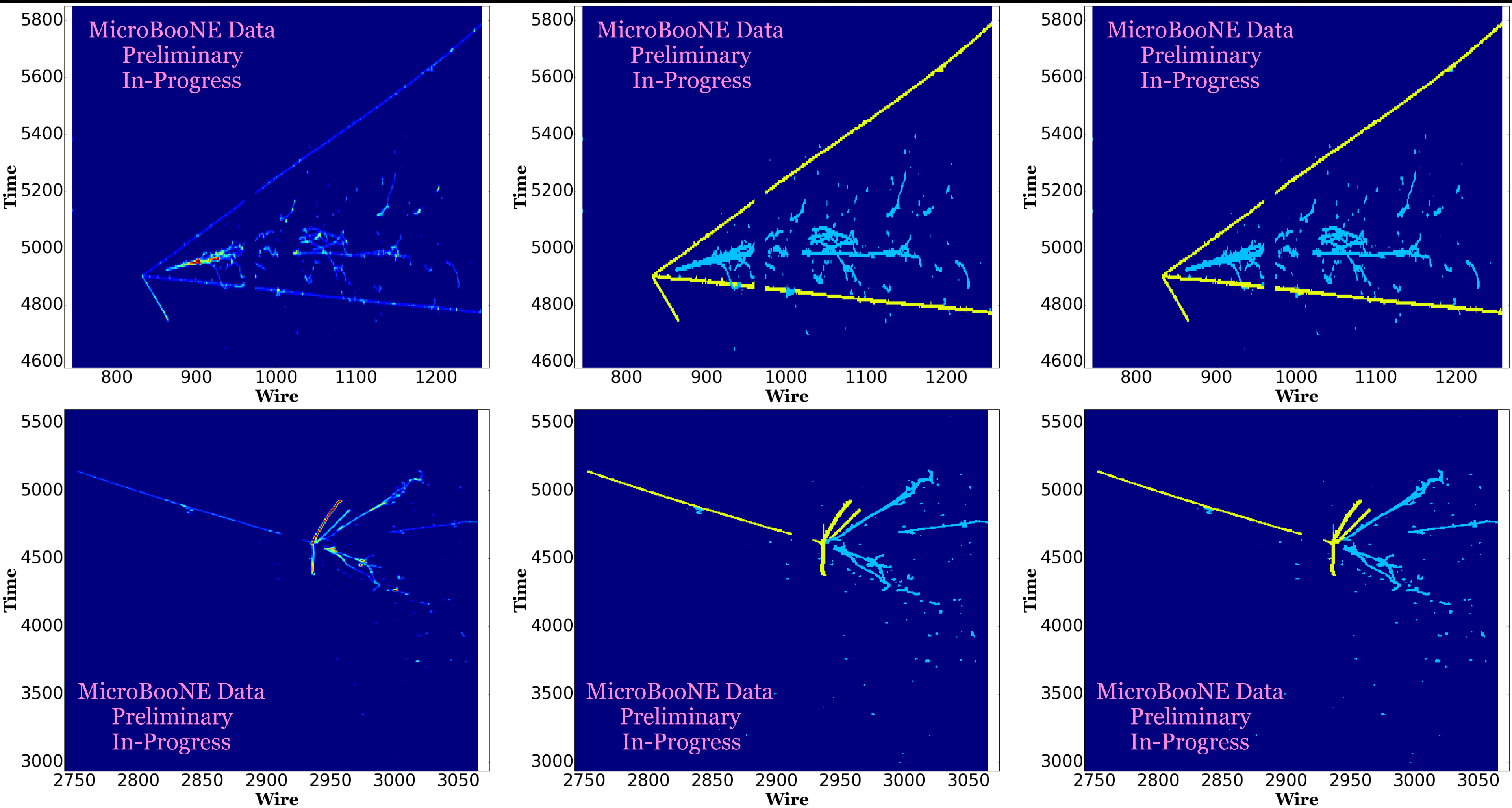


# Decay Muons: Example Displays




Input Image | Human Label | SSNet Label

# 4 Visually Picked “Busy Neutrino Events”




Input Image Human Label SSNet Label

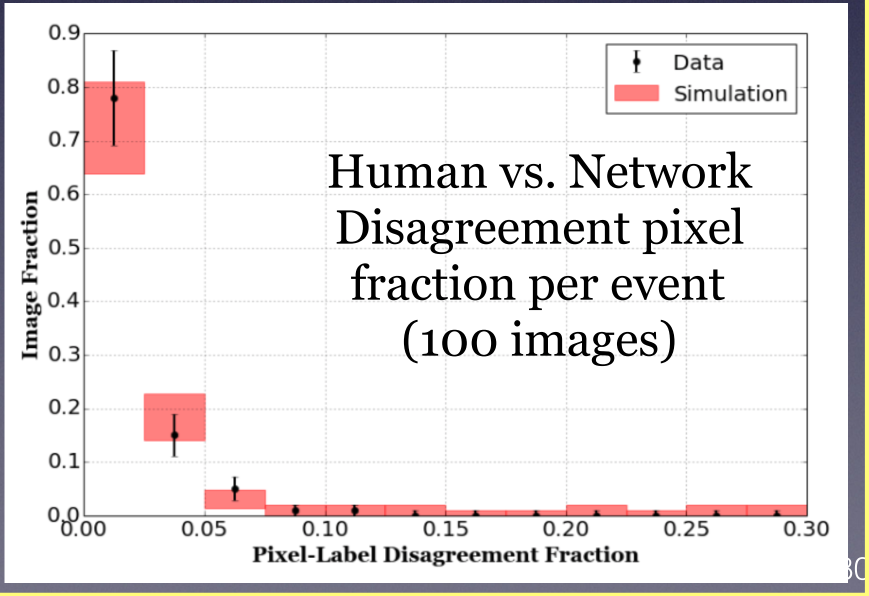
# Overall Performance

- **Data/Simulation agreement within statistical error**
  - No systematic error included
- **Network does better than a human** analyzer (sim.)

## Muon Decay

**Disagreement rate mean/std in %**

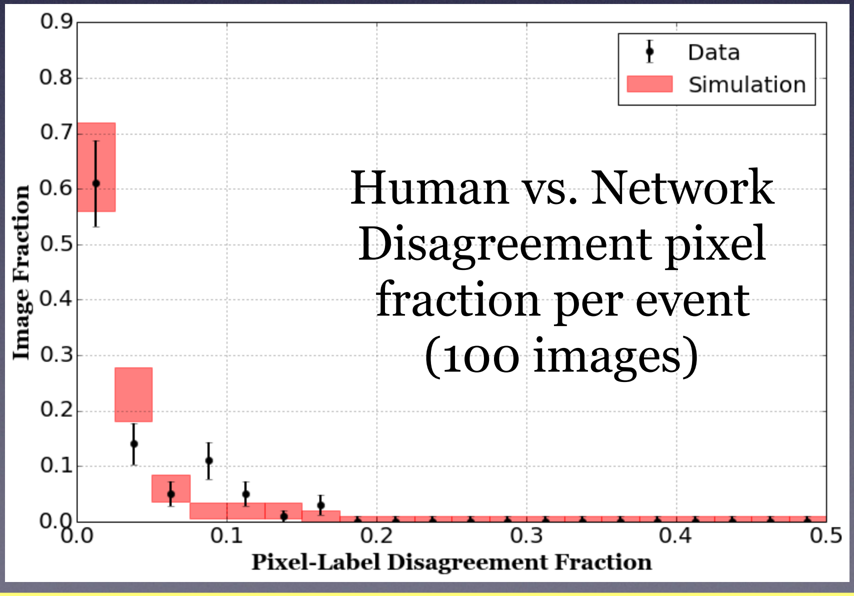
| Sample | Data | Simulation | Simulation | Simulation |
|---|---|---|---|---|
| Label | Physicist | Physicist | Simulation | Simulation |
| Prediction | U-Resnet | U-ResNet | U-ResNet | Physicist |
| ICPF mean | 1.8 | 2.6 | 2.5 | 2.3 |
| ICPF 90% | 3.3 | 4.4 | 4.5 | 3.1 |
| Shower | 6.2 | 5.7 | 4.0 | 3.9 |
| Track | 1.1 | 1.9 | 1.6 | 1.3 |




## Neutrino w/ Gamma

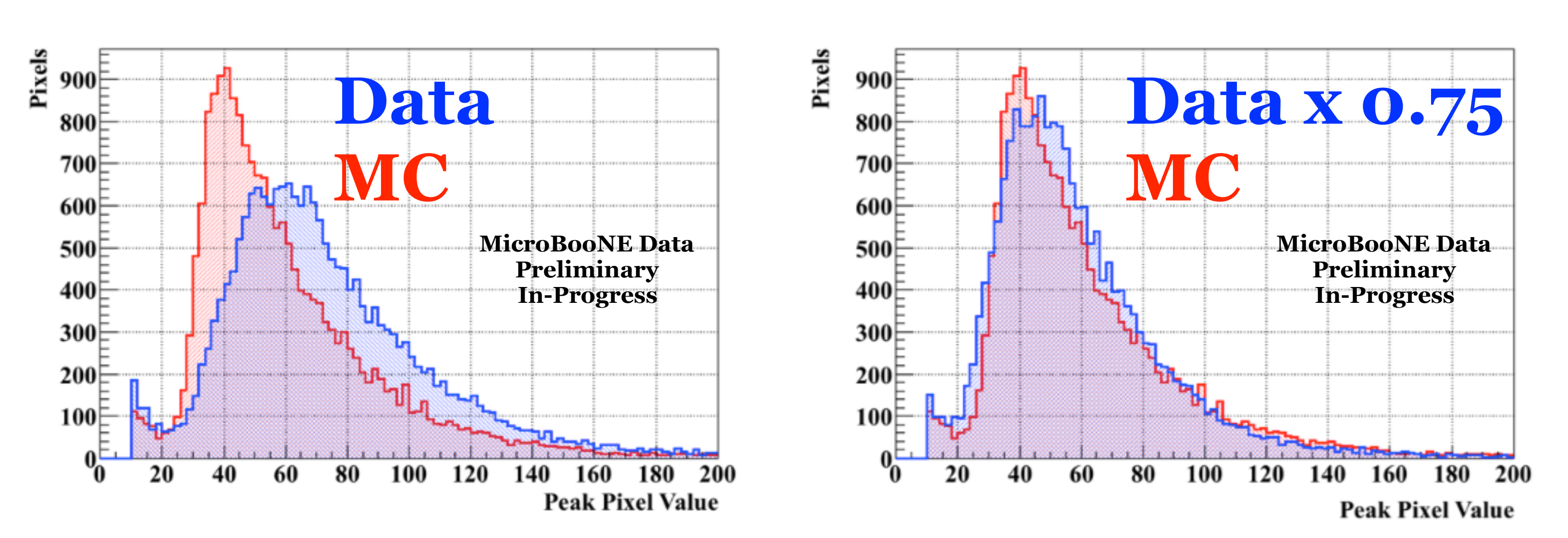
**Disagreement rate mean/std in %**

| Sample | Data | Simulation | Simulation | Simulation |
|---|---|---|---|---|
| Label | Physicist | Physicist | Simulation | Simulation |
| Prediction | U-ResNet | U-ResNet | U-ResNet | Physicist |
| ICPF mean | 3.4 | 2.5 | 1.8 | 2.0 |
| ICPF 90% | 9.0 | 5.7 | 4.6 | 4.8 |
| Shower | 4.8 | 3.4 | 3.0 | 2.6 |
| Track | 2.7 | 2.4 | 2.2 | 2.9 |



# Decay Muons: Pixel Value Variation

Studied how network performance varies when pixel values are scaled by a constant factor



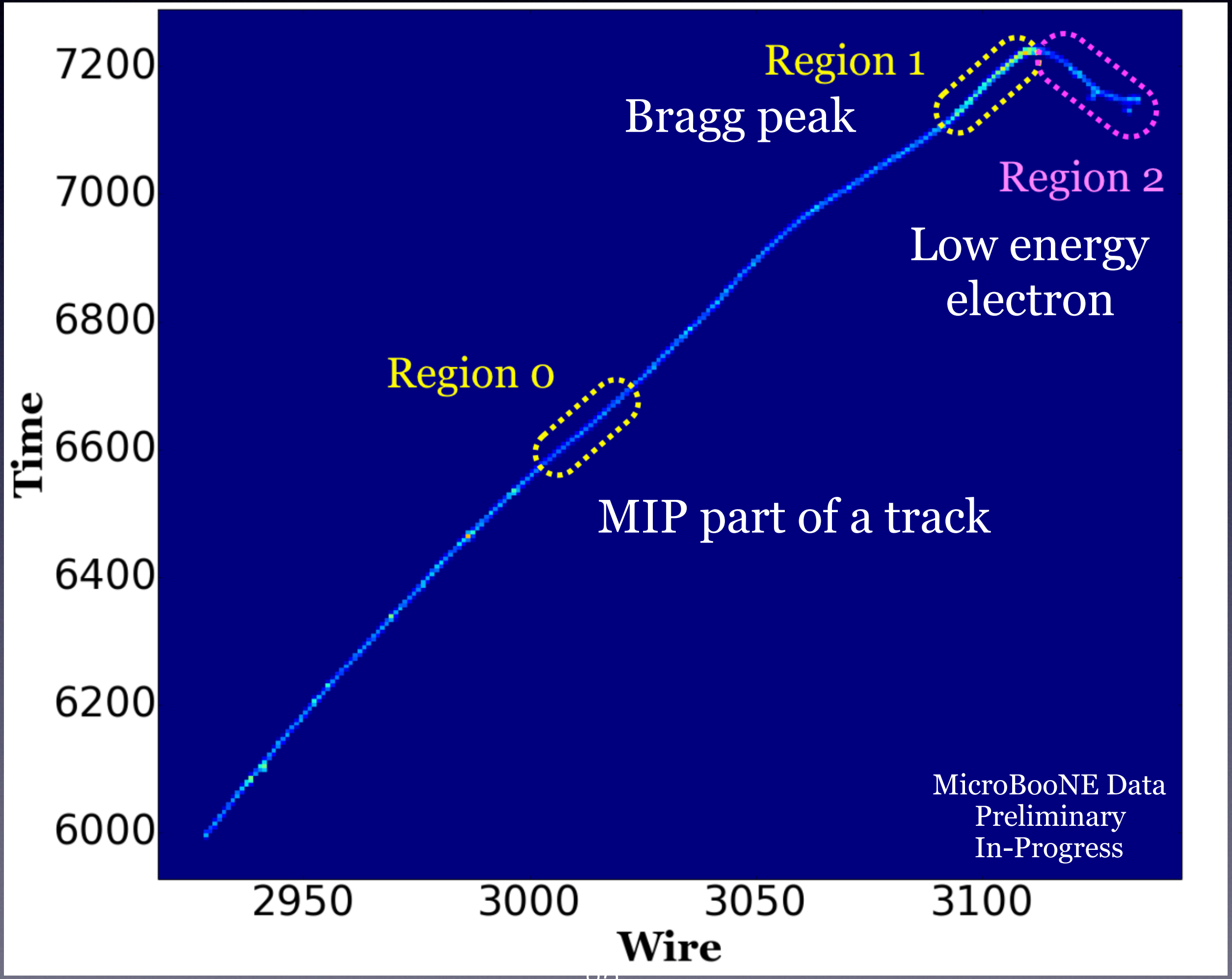

**No scaling**

| Scaling Factor | 0.75 | 0.80 | 0.85 | 0.90 | 0.95 | 1.00 | 1.05 | 1.10 | 1.15 | 1.20 | 1.25 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Track | 2.38 | 2.11 | 1.93 | 1.65 | 1.40 | 1.14 | 1.16 | 1.20 | 1.26 | 1.28 | 1.32 |
| Shower | 5.24 | 5.22 | 5.41 | 6.02 | 6.11 | 6.16 | 6.11 | 6.14 | 6.21 | 6.28 | 6.34 |
| Combined | 2.75 | 2.53 | 2.40 | 2.21 | 2.02 | 1.81 | 1.85 | 1.89 | 1.96 | 2.00 | 2.04 |

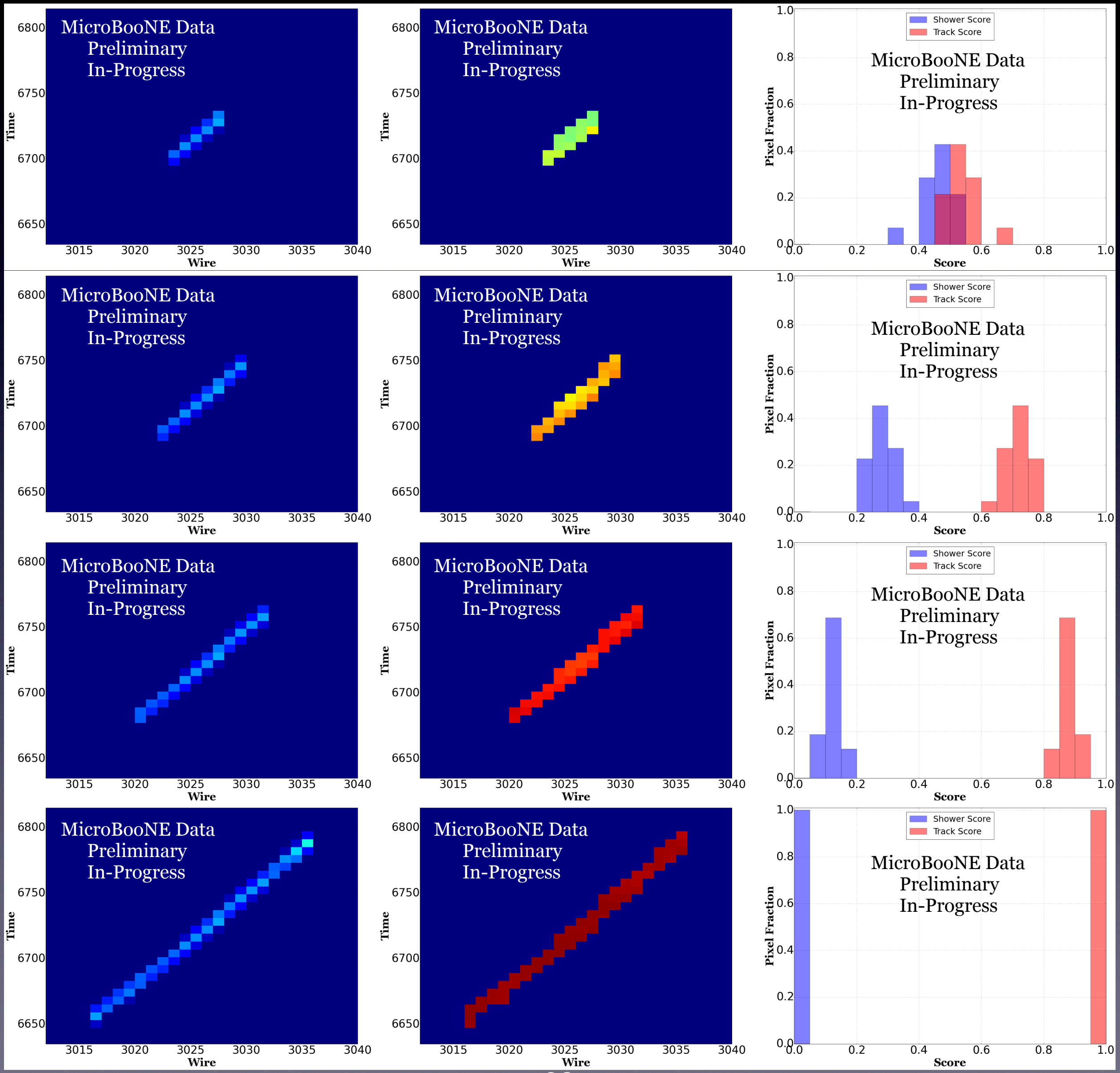
Change in the mean error rate is within 1% when pixel values are scaled within 20%, fairly robust

# Decay Muons: Inter-Pixel Correlation

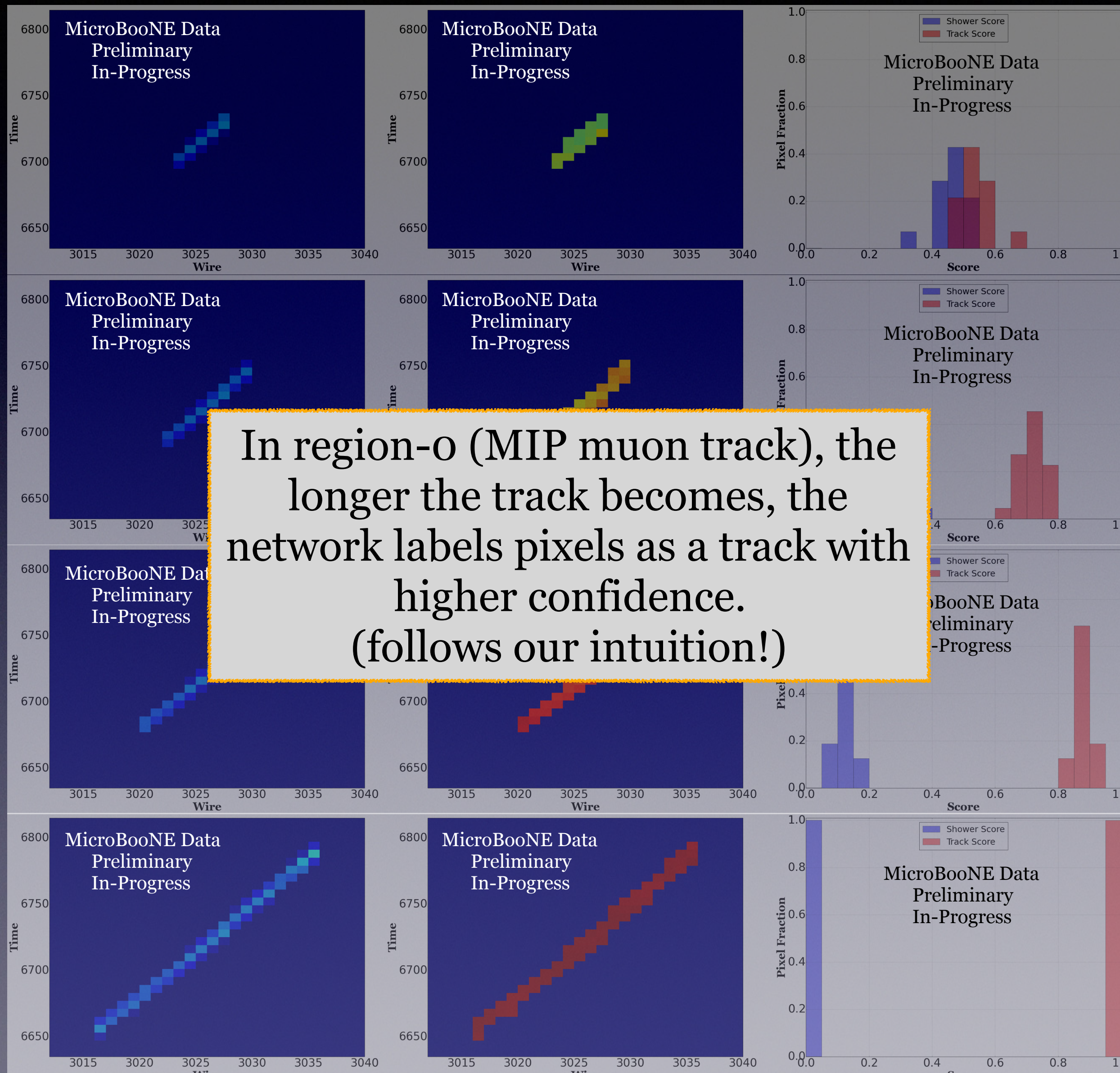
## Study, qualitatively, how network reacts to interesting portions of an image



# Decay Muon: Inter-Pixel Correlation … Region-0

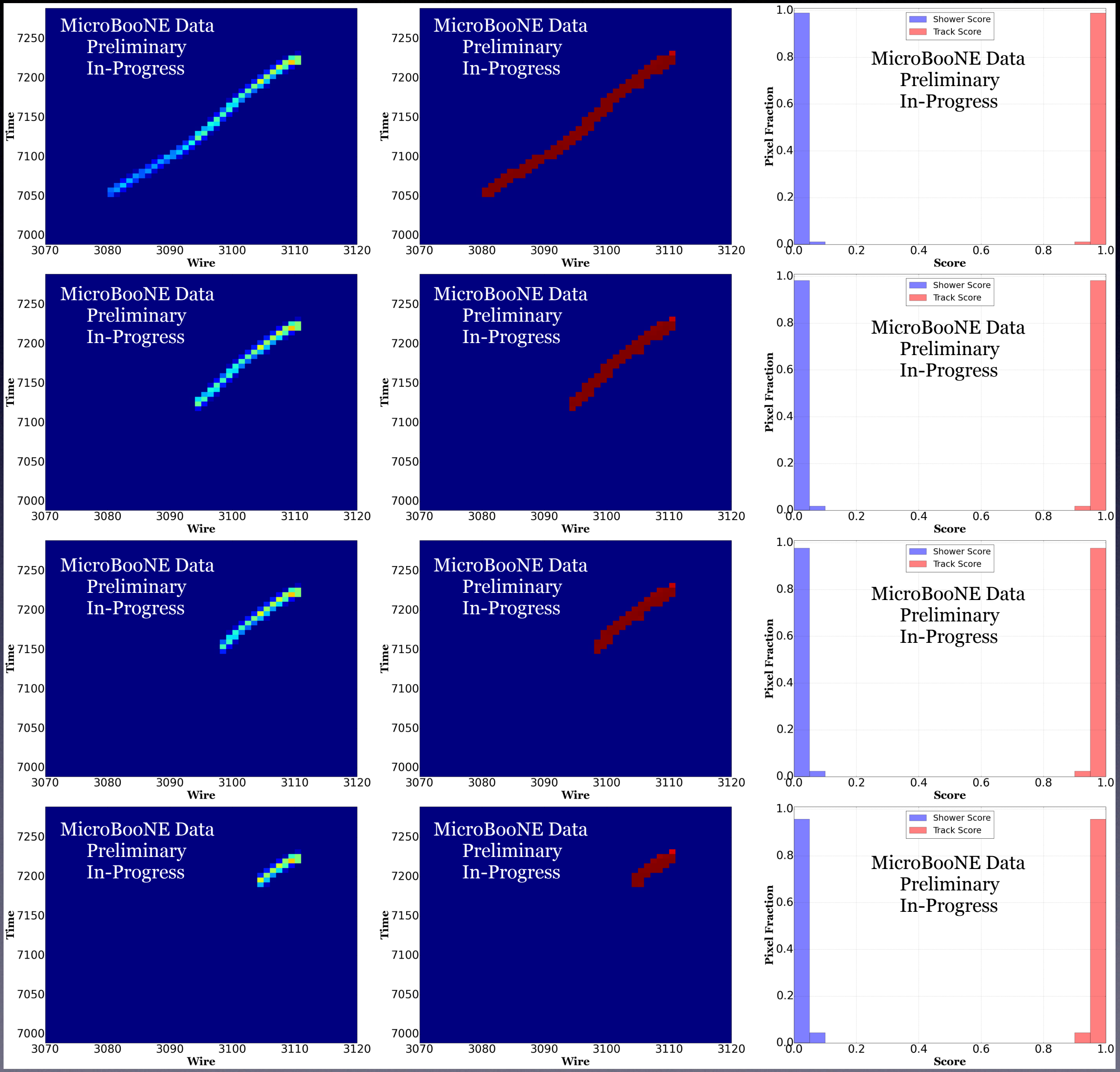


# Decay Muon: Inter-Pixel Correlation … Region-0

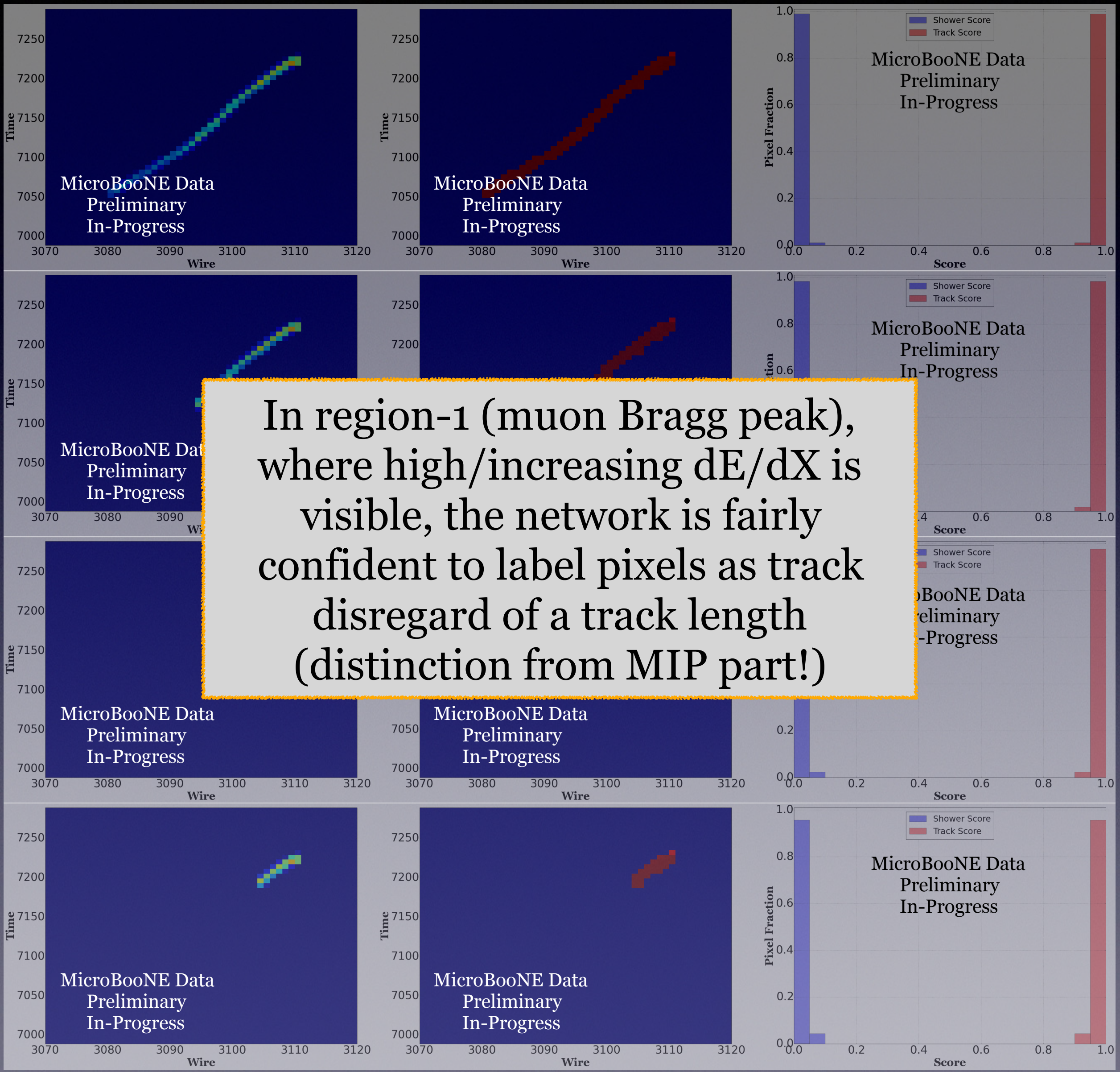



In region-0 (MIP muon track), the longer the track becomes, the network labels pixels as a track with higher confidence.
(follows our intuition!)

# Decay Muon: Inter-Pixel Correlation … Region-1

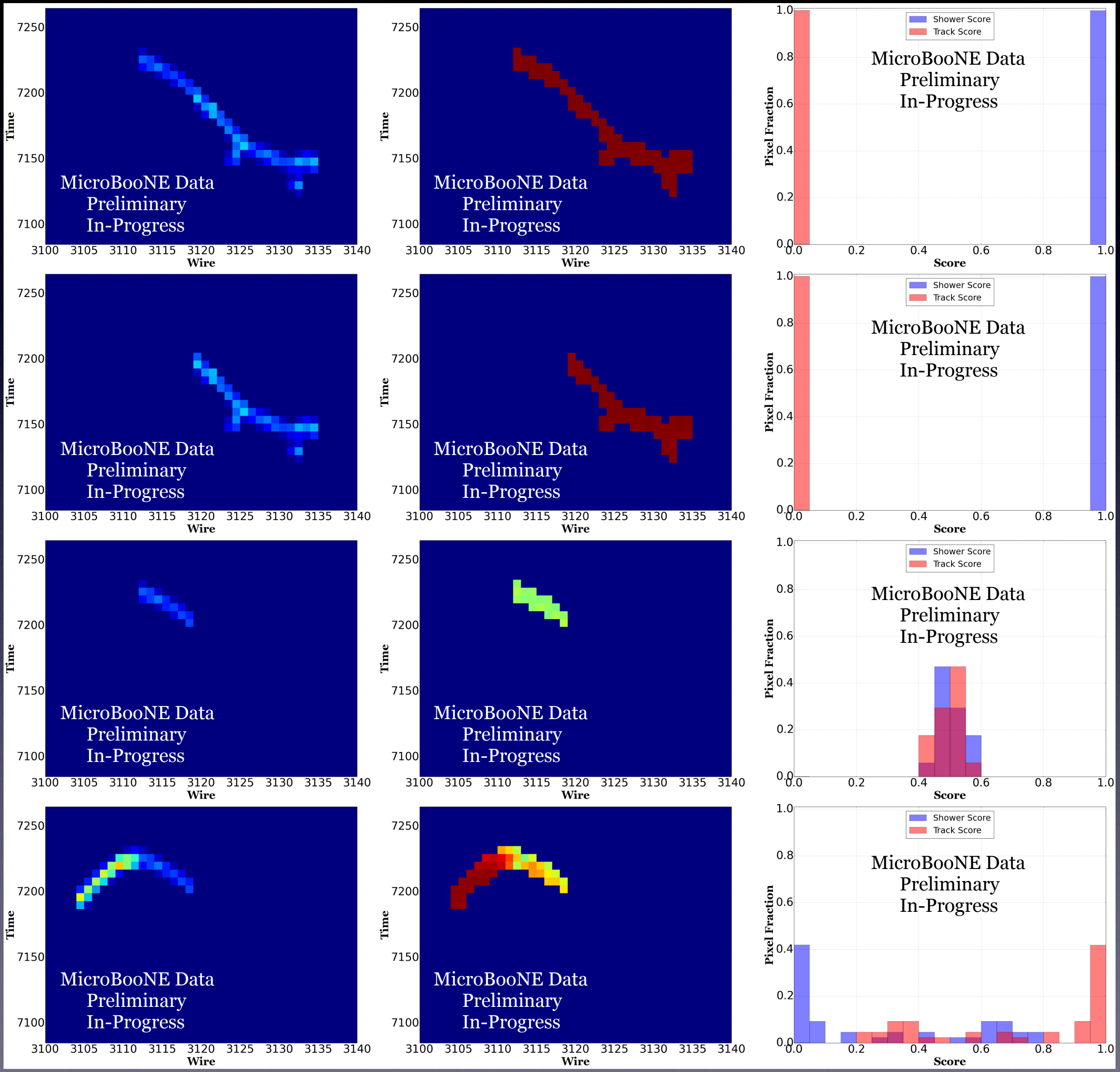


# Decay Muon: Inter-Pixel Correlation … Region-1

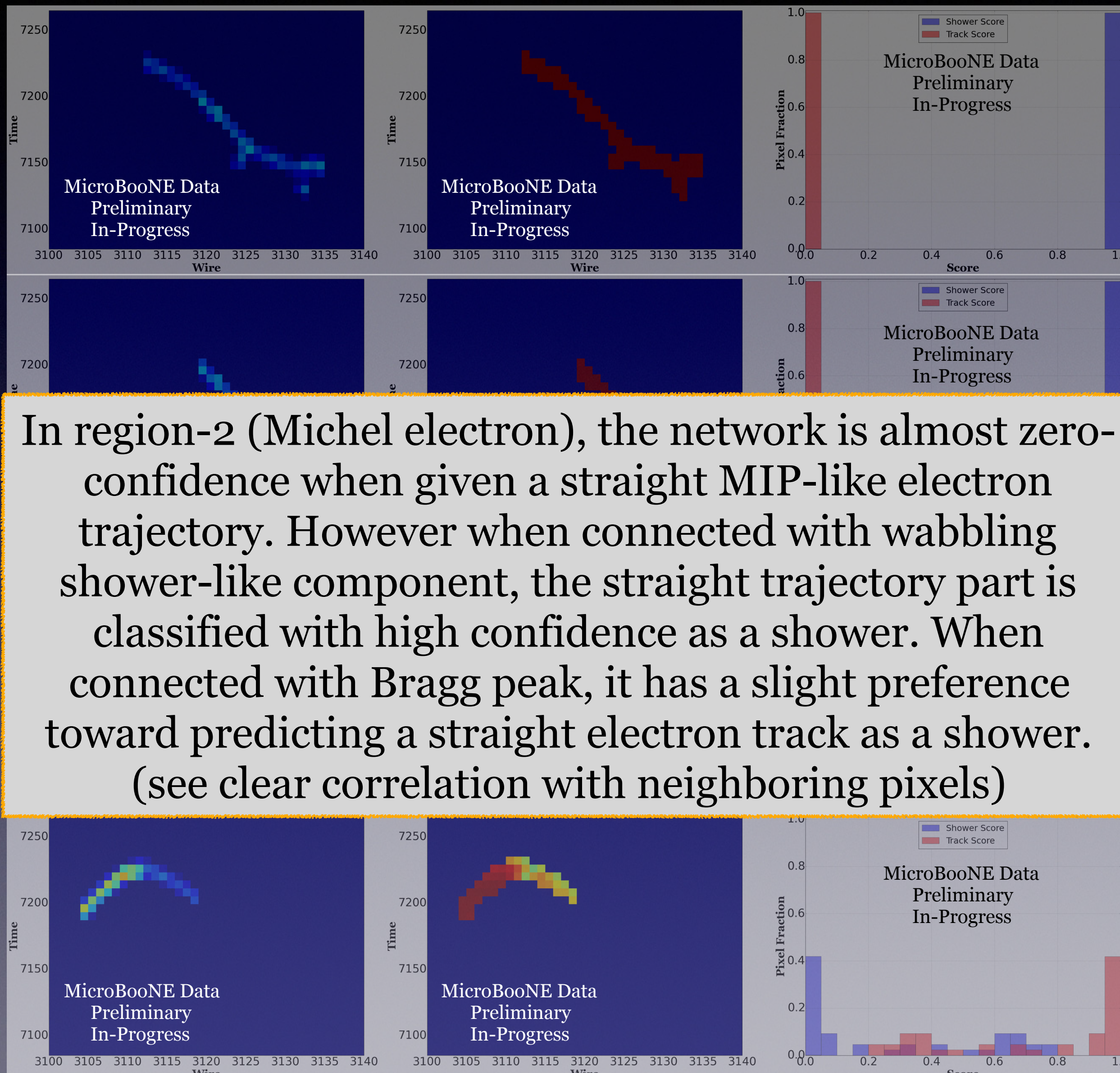



In region-1 (muon Bragg peak), where high/increasing dE/dX is visible, the network is fairly confident to label pixels as track disregard of a track length (distinction from MIP part!)

# Decay Muon: Inter-Pixel Correlation … Region-2

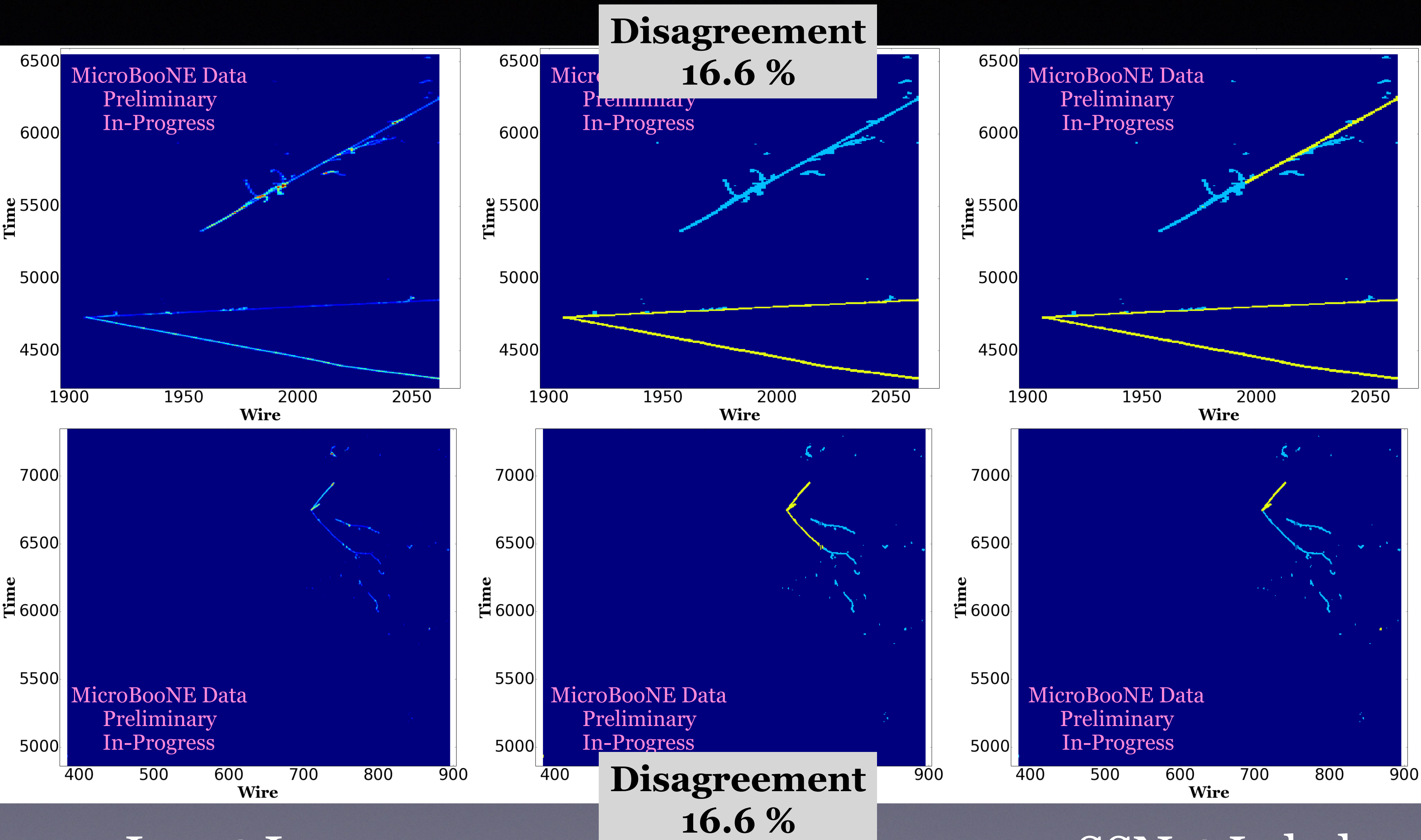


# Decay Muon: Inter-Pixel Correlation … Region-2




In region-2 (Michel electron), the network is almost zero-confidence when given a straight MIP-like electron trajectory. However when connected with wabbling shower-like component, the straight trajectory part is classified with high confidence as a shower. When connected with Bragg peak, it has a slight preference toward predicting a straight electron track as a shower. (see clear correlation with neighboring pixels)

# Neutrino w/ Gammas: 4 Worst Events

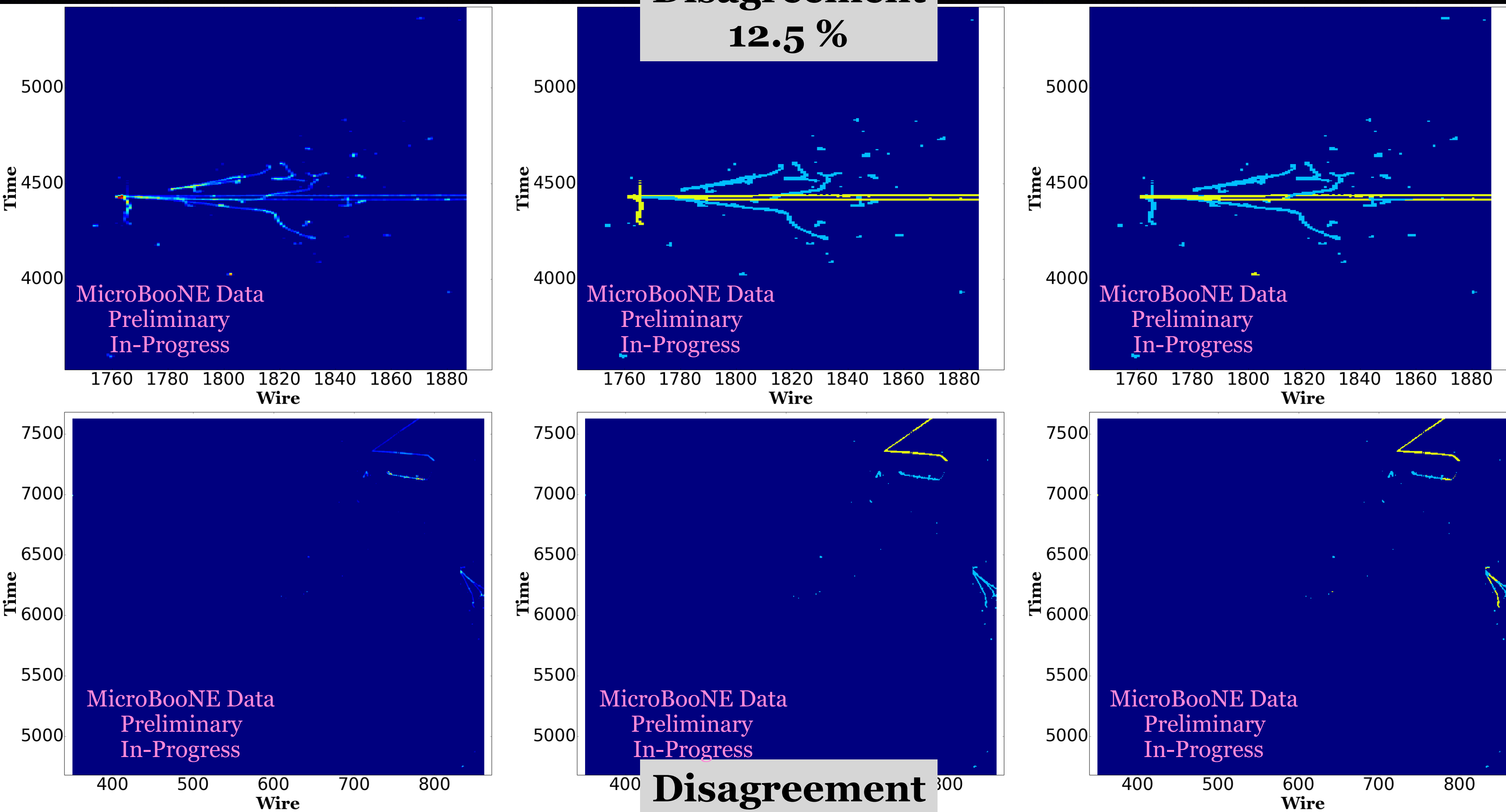



Input Image

SSNet Label

# Neutrino w/ Gammas: 4 Worst Events

**Disagreement 12.5 %**




**Disagreement 16.2 %**

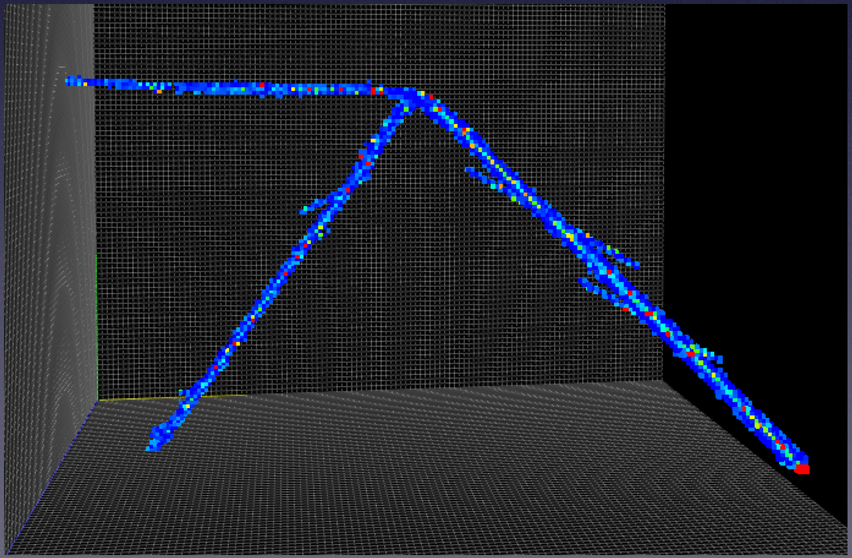
Input Image

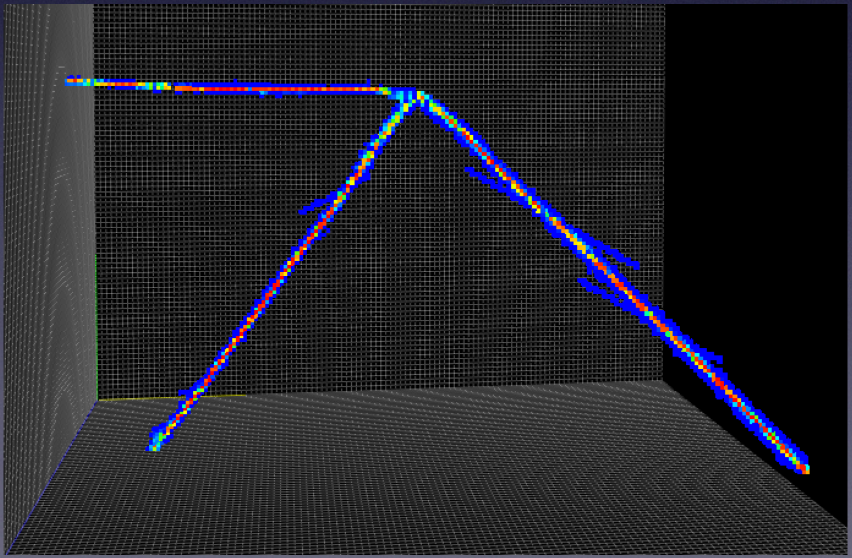
SSNet Label

# 3D Data Reconstruction @ SLAC

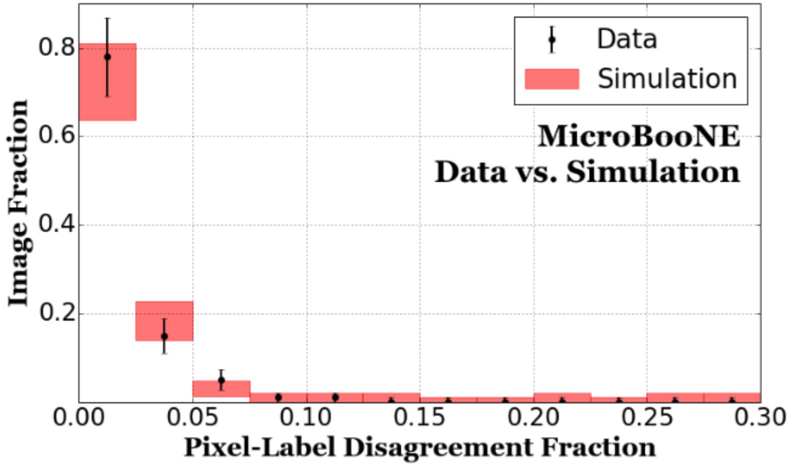
**Tracy Usher**

- Showing ML can be started above age of 60



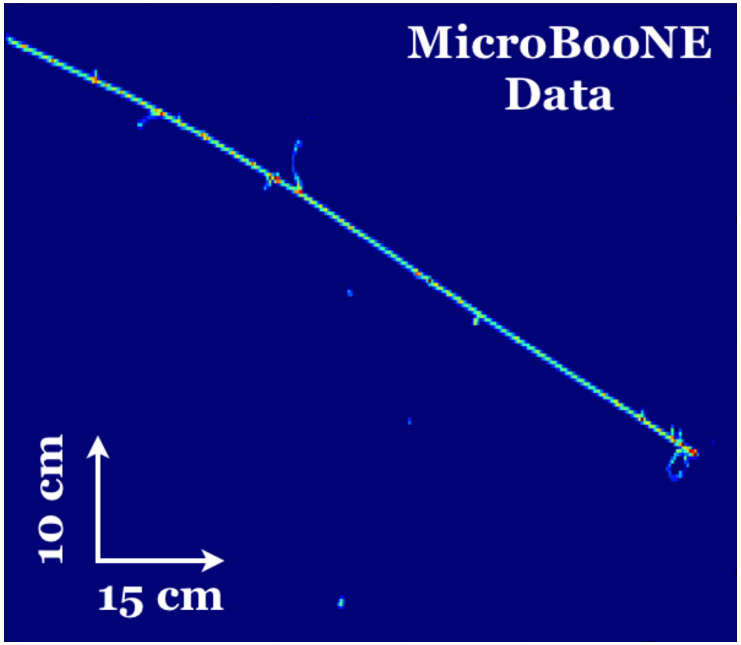


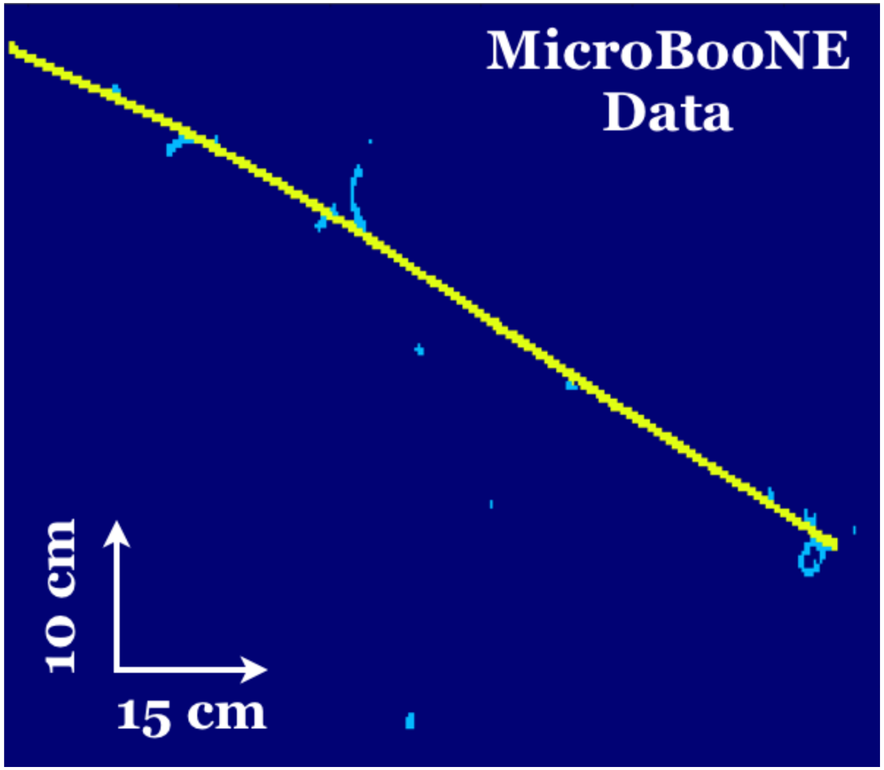
**Tracy shows you can start ML above age of 60**




## Technique Validation on Data

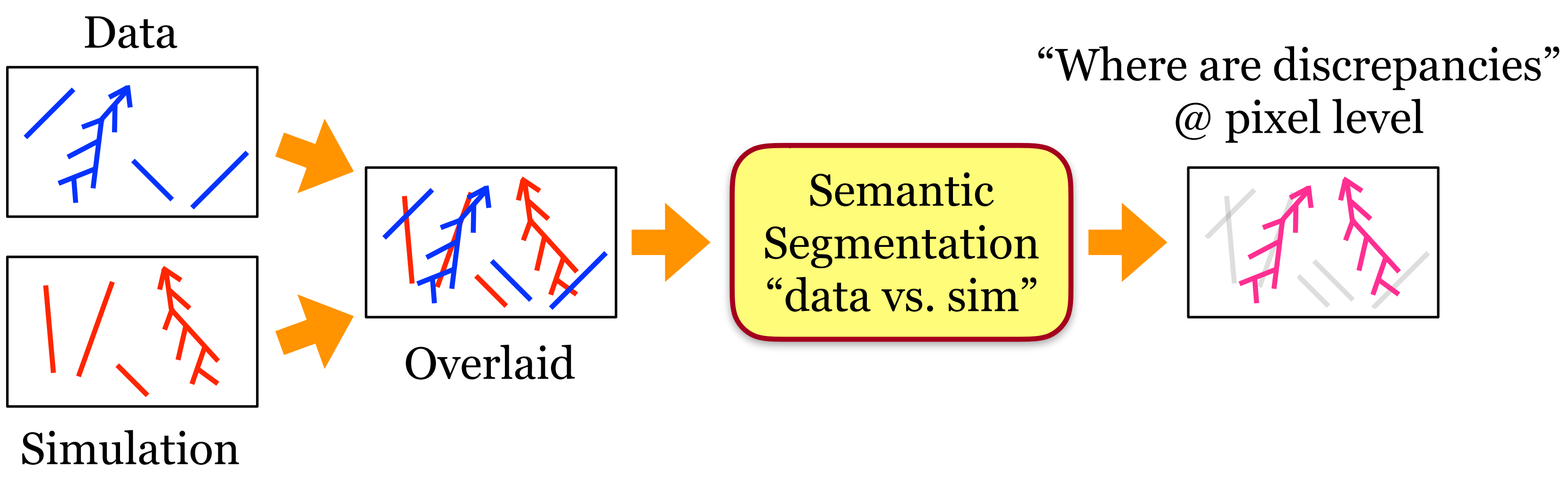
- Same paper … arXiv:1808.07269
  - Important for new techniques such as this
- Compared physicist vs. network predictions



## What can we do about imperfect simulation?

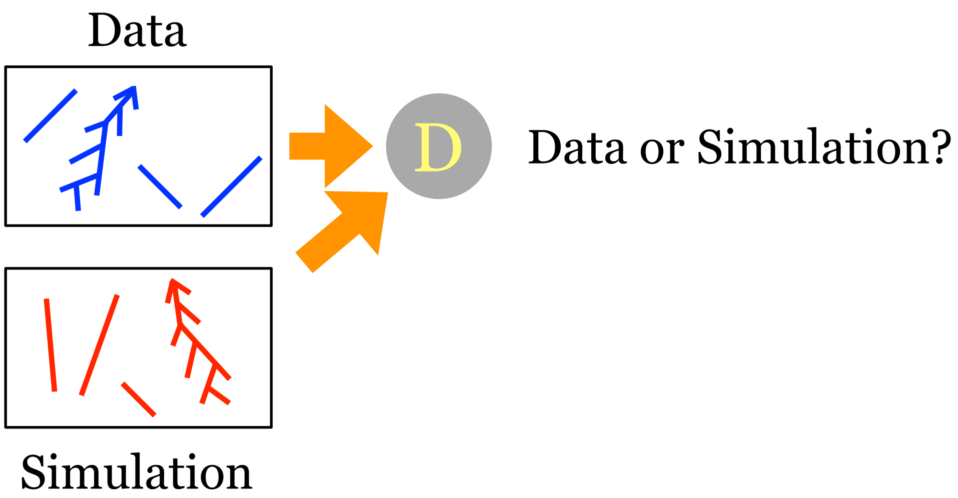
- **Problematic**: the "signal distribution" learnt by the algorithm may be different in two domains!
- **Mitigation techniques** in ML domain?
  - **Can** try CNN to "locate" where it is



# Drawbacks of supervised learning & mitigations

## What can we do about imperfect simulation?

- **Problematic**: the "signal distribution" learnt by the algorithm may be different in two domains!
- **Mitigation techniques** in ML domain?
  - **Can** try CNN to "locate" where it is
  - **Can** try CNN to "fix" the discrepancy

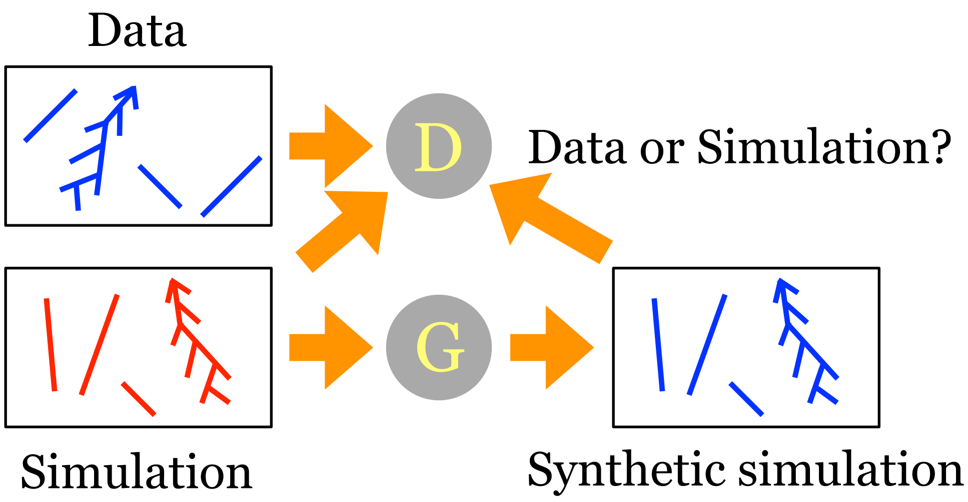


## Drawbacks of supervised learning & mitigations

SLAC

# What can we do about imperfect simulation?

- **Problematic**: the "signal distribution" learnt by the algorithm may be different in two domains!
- **Mitigation techniques** in ML domain?
  - **Can** try CNN to "locate" where it is
  - **Can** try CNN to "fix" the discrepancy



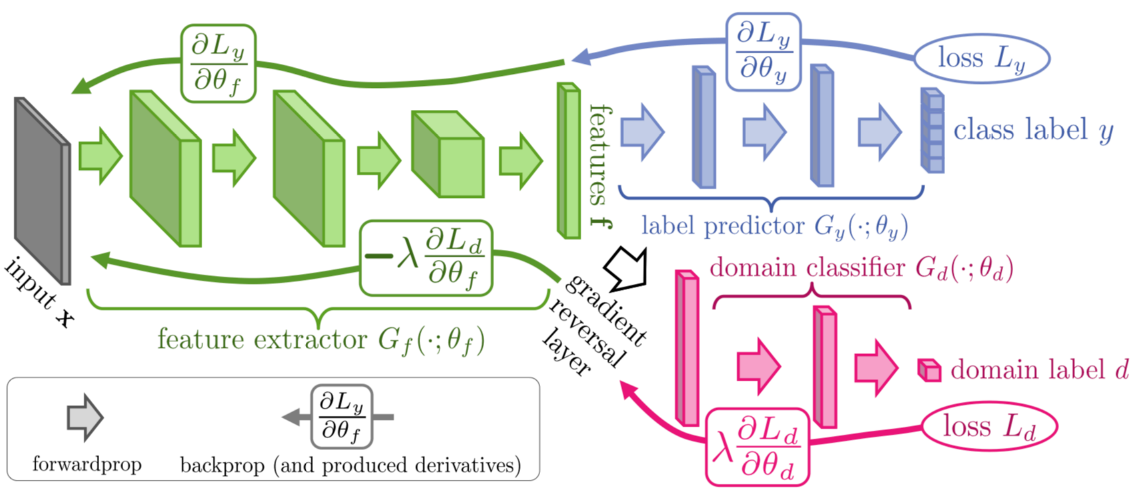

**Generative Adversarial Network**
Can learn the "mapping" between the data and simulation "distributions". The generator network can be used as a synthetic image generator to train different neural networks

# Future/On-going Projects
## Drawbacks of supervised learning & mitigations

SLAC

# What can we do about imperfect simulation?

- **Problematic**: the "signal distribution" learnt by the algorithm may be different in two domains!
- **Mitigation techniques** in ML domain?
  - **Can** try CNN to "locate" where it is
  - **Can** try CNN to "fix" the discrepancy
  - **Can** try a training technique to minimize the effect




Maximize the loss for discriminate data vs. simulation, feature extractors are penalized to key on simulation specific information

**Minerva Paper** arXiv:1808.08332

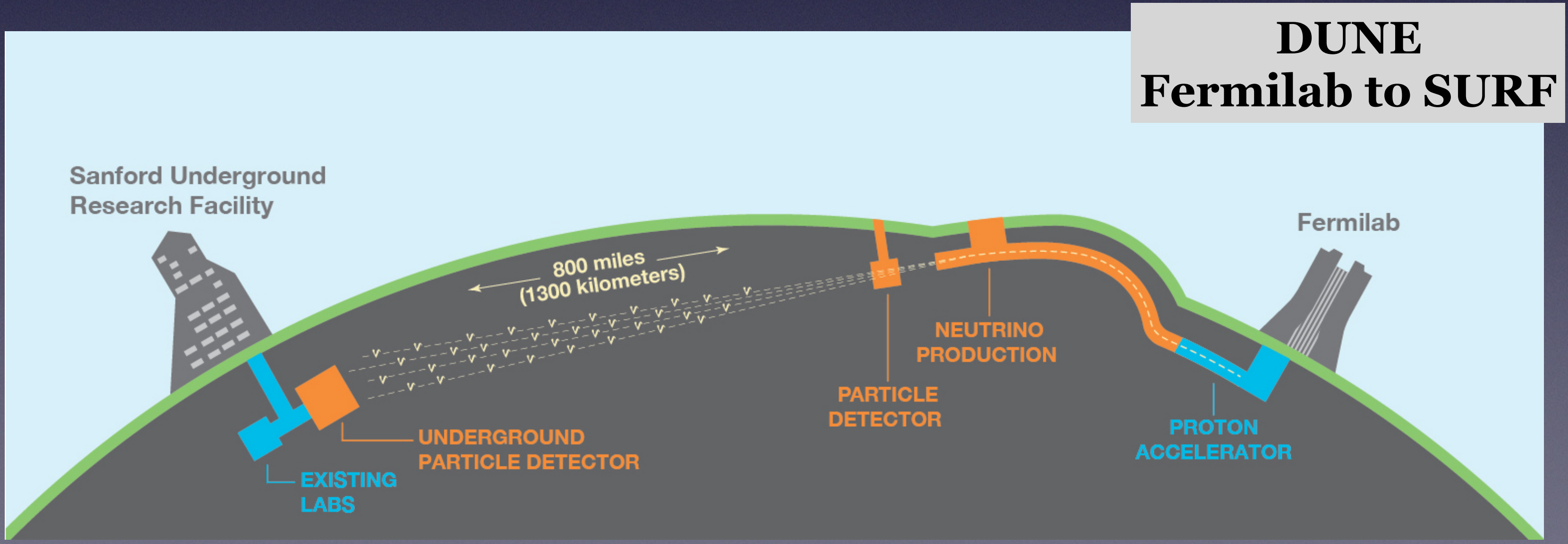
Domain-Adversarial Training of Neural Networks
J. Mach. Learn. Res. 17 (2016)

# LArTPC Experiments

## Accelerator-based oscillation experiments

**DUNE**: long baseline program (first beam expected @ 2026)

- Measure mass hierarchy and CP violation ($\nu_\mu \rightarrow \nu_e$ vs. $\bar{\nu}_\mu \rightarrow \bar{\nu}_e$)
- Rare physics processes (proton decay, n-n, Supernova neutrinos)



# LArTPC Experiments

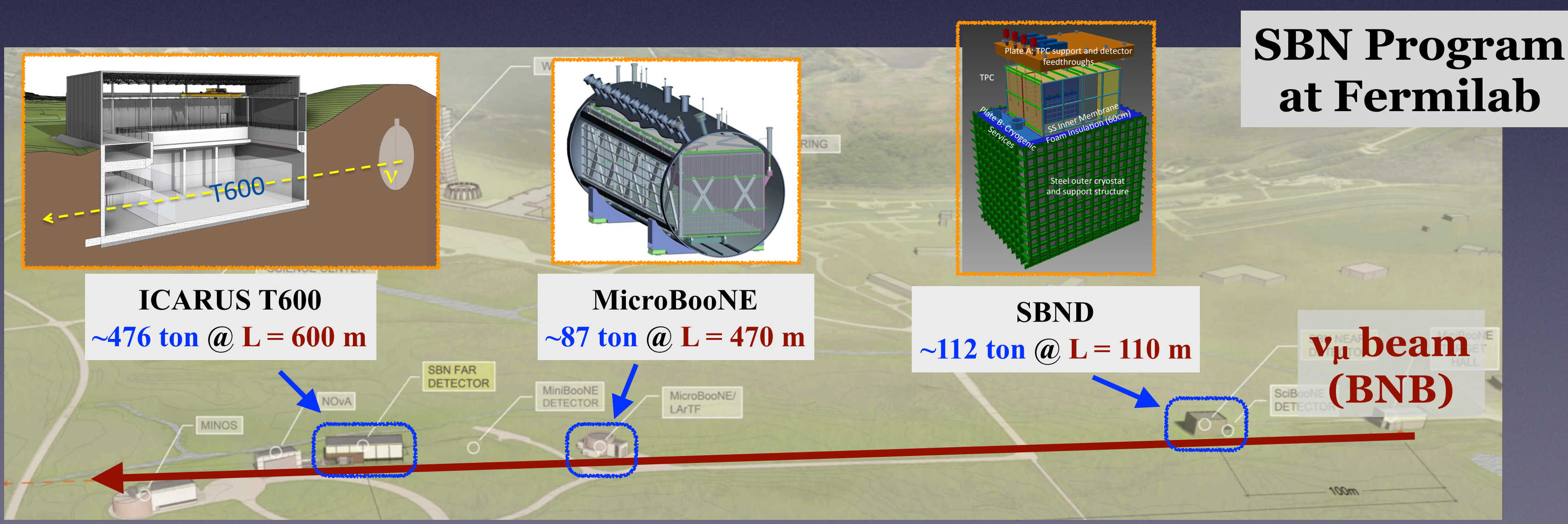
## Accelerator-based oscillation experiments

**DUNE**: long baseline program (first beam expected @ 2026)

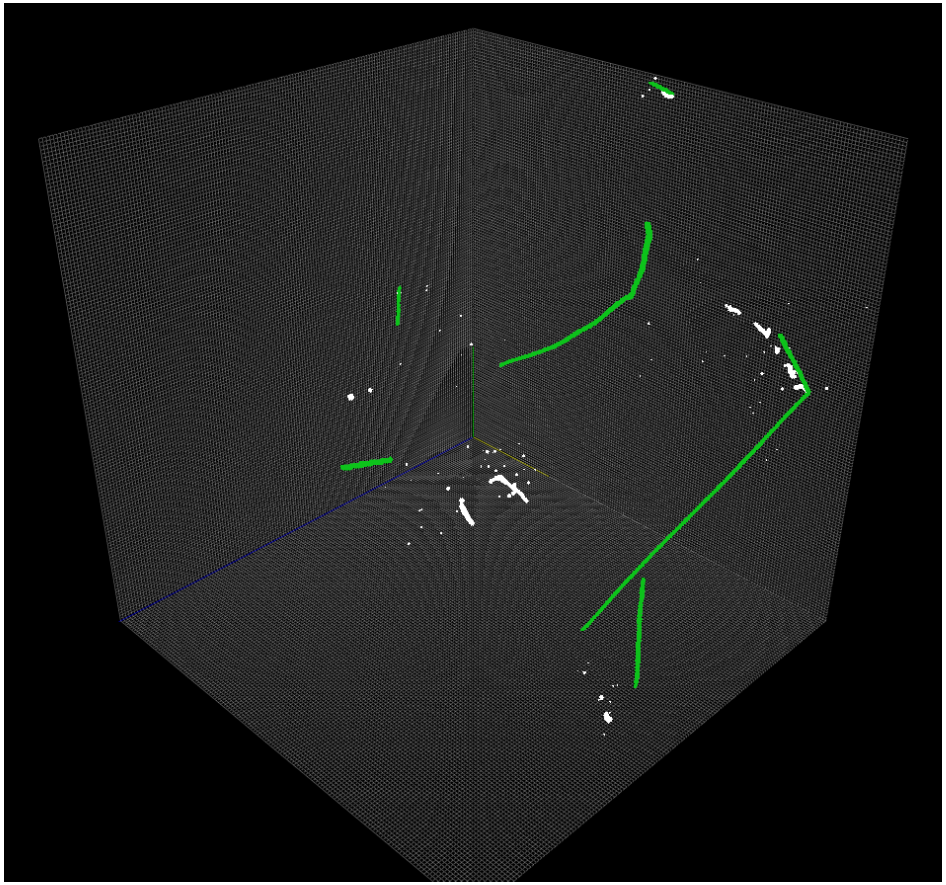
- Measure mass hierarchy and CP violation ($\nu_\mu \to \nu_e$ vs. $\bar{\nu}_\mu \to \bar{\nu}_e$)
- Rare physics processes (proton decay, n-$\bar{n}$, Supernova neutrinos)

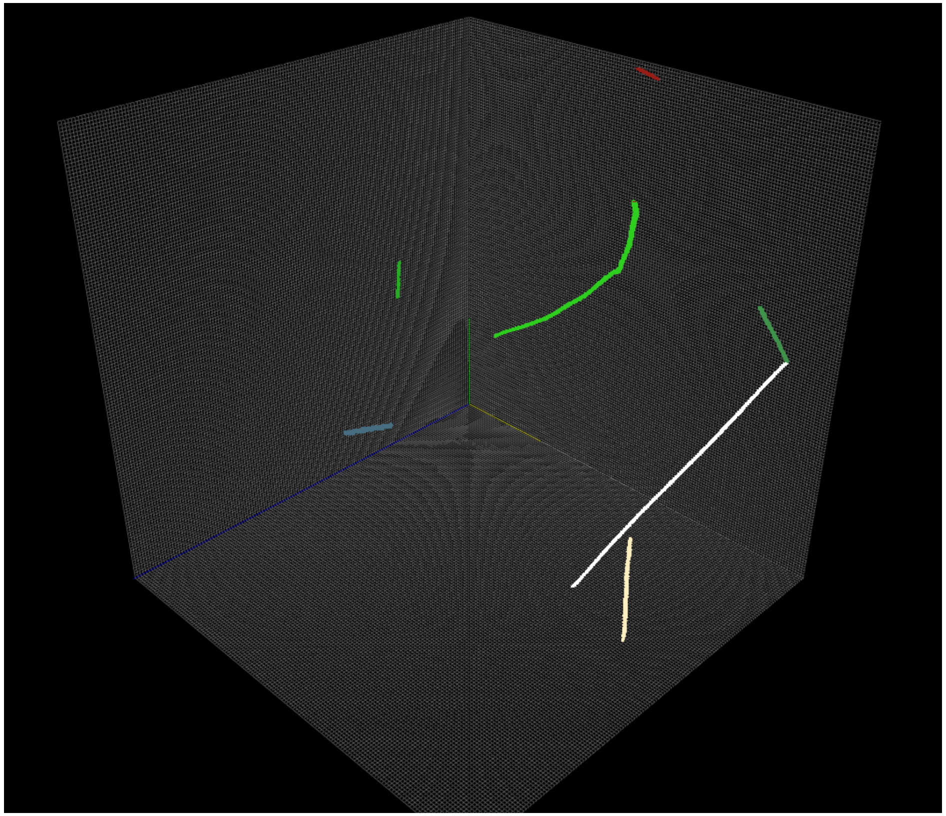
**SBN**: short baseline program (2015 ~)

- Measure $\nu_\mu \to \nu_e$ to investigate possible sterile neutrino oscillation
- Employs three LArTPC detectors at different baselines
- LArTPC R&D for DUNE



**Clustering** … on-going work
(**Left**: track/shower separation output)
(**Right**: track pixel clustering using graph NN)

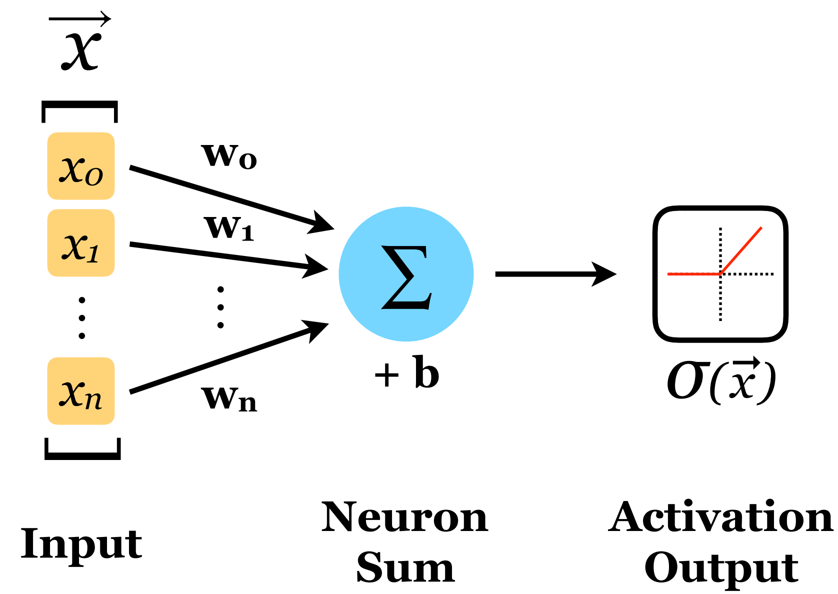




# Simple neural network (perceptron)

The basic unit of a neural net is the *perceptron* (loosely based on a real neuron)

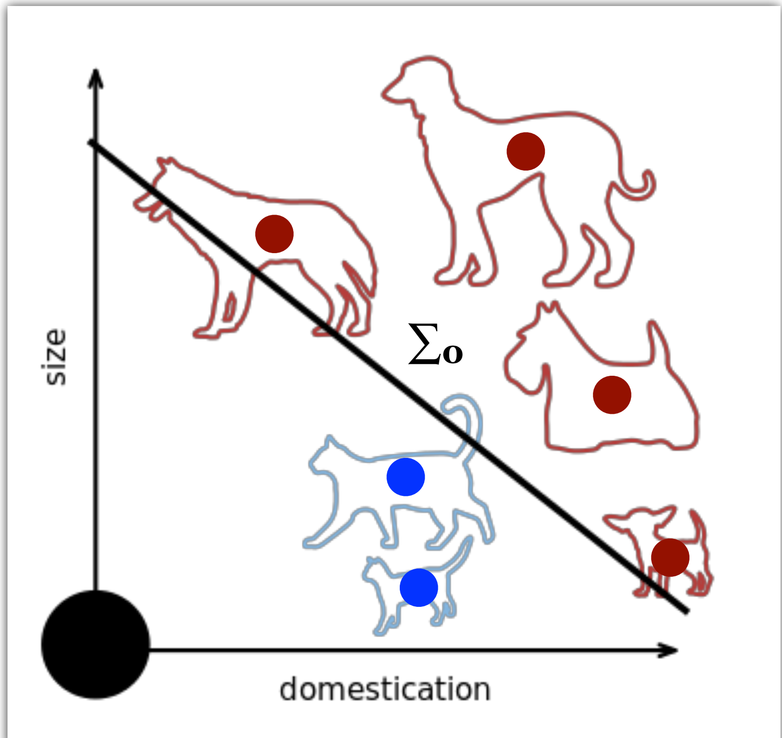
Takes in a vector of inputs ($x$). Commonly inputs are summed with weights ($w$) and offset ($b$) then run through activation.




$$\sigma(\vec{x}) = \begin{cases} \vec{w}_i \cdot \vec{x} + b_i & \vec{w}_i \cdot \vec{x} + b_i \geq 0 \\ 0 & \vec{w}_i \cdot \vec{x} + b_i < 0. \end{cases}$$

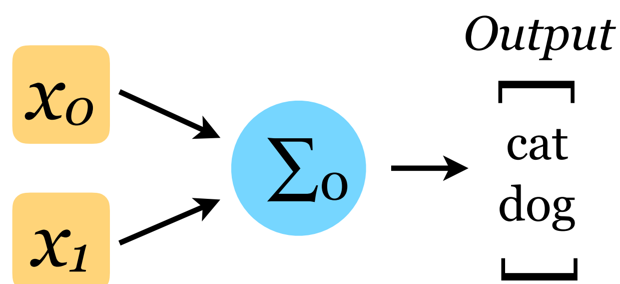
# Simple neural network (perceptron)

Imagine using two features to separate cats and dogs




from [wikipedia]

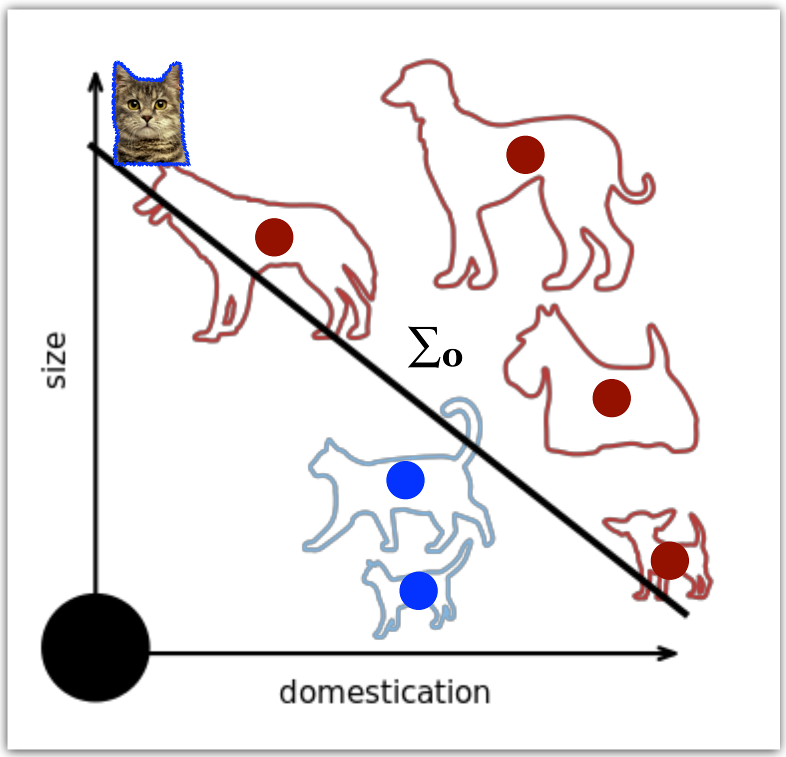
$$\sigma(\vec{x}) = \begin{cases} \vec{w}_i \cdot \vec{x} + b_i & \vec{w}_i \cdot \vec{x} + b_i \geq 0 \\ 0 & \vec{w}_i \cdot \vec{x} + b_i < 0. \end{cases}$$



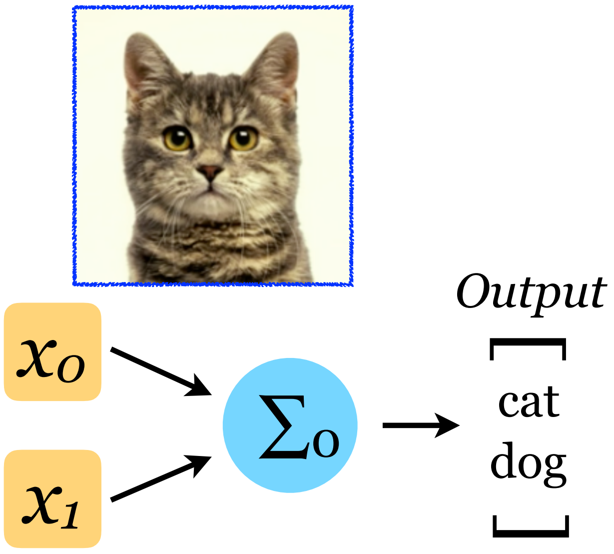

By picking a value for **w** and **b**, we define a boundary between the two sets of data

# Machine Learning Overview
## Simple neural network (perceptron)

What if we have a new data point?

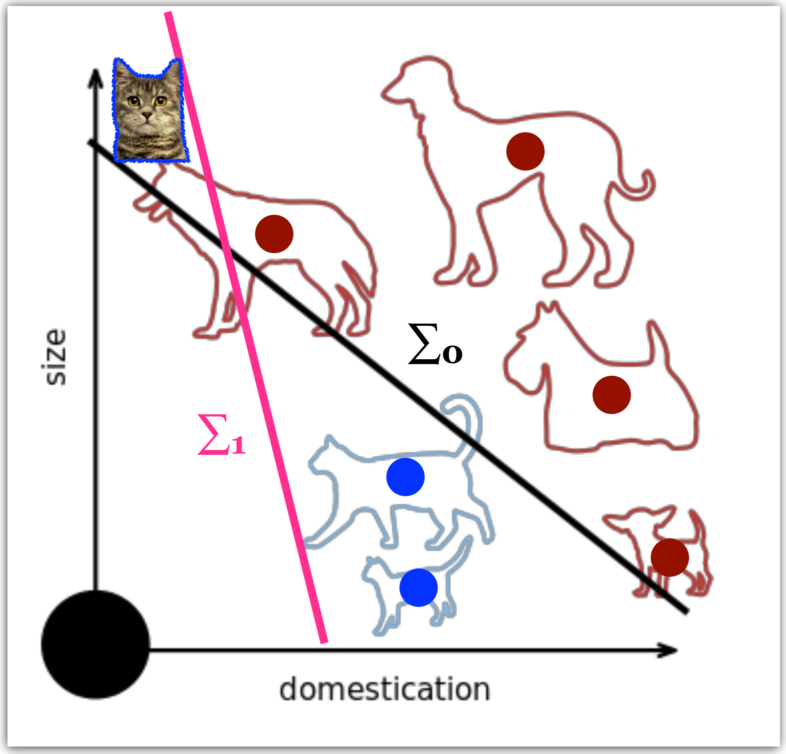



from [wikipedia]

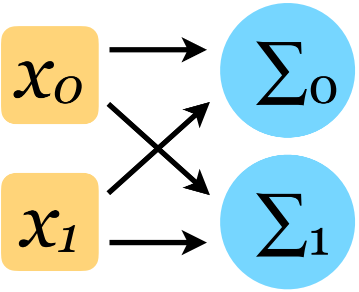


# Simple neural network (perceptron)

What if we have a new data point?



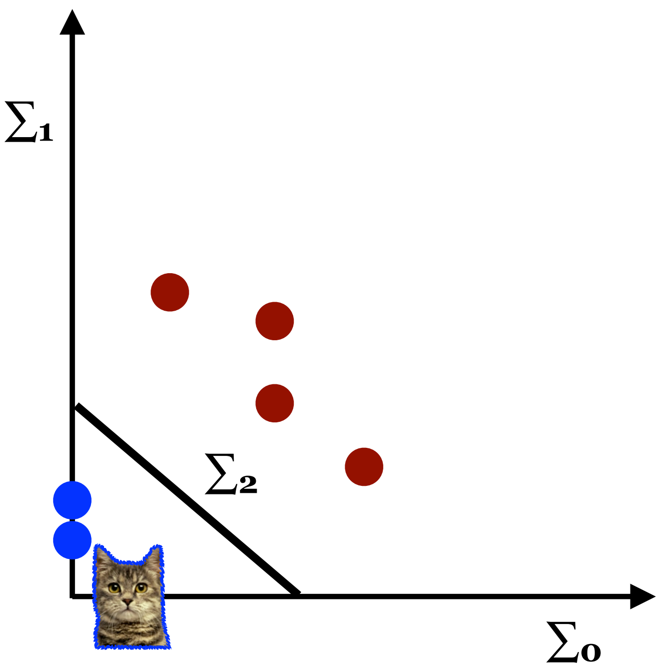

from [wikipedia]



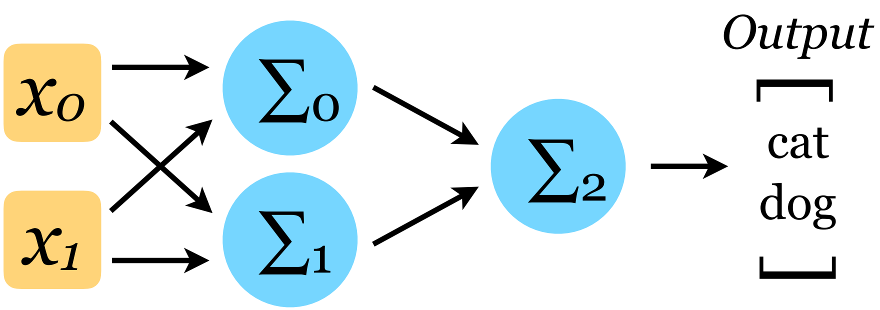

We can **add another perceptron** to help (but does not yet solve the problem)

# Simple neural network (perceptron)

What if we have a new data point?



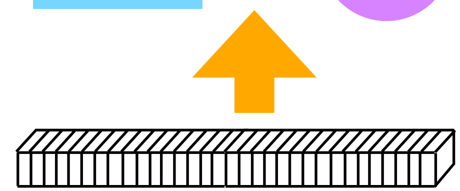

from [wikipedia]



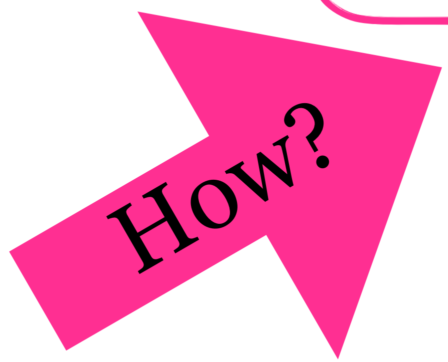

**Another layer** can classify based on preceding layer's output (of **non-linear activation**)

# Machine Learning Overview
## Back to analyzing a cat “image…”

**Goal:** Dog or Cat



1D array of discriminants




This part can be done with a classic (fully-connected) neural network

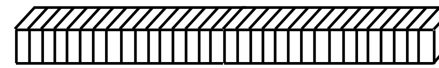
**How can we extract “features” from “image”?**

**… the hard part …**

**(where I have failed for long)**

# Back to analyzing a cat "image…"

**Goal:** Dog or Cat



1D array of discriminants

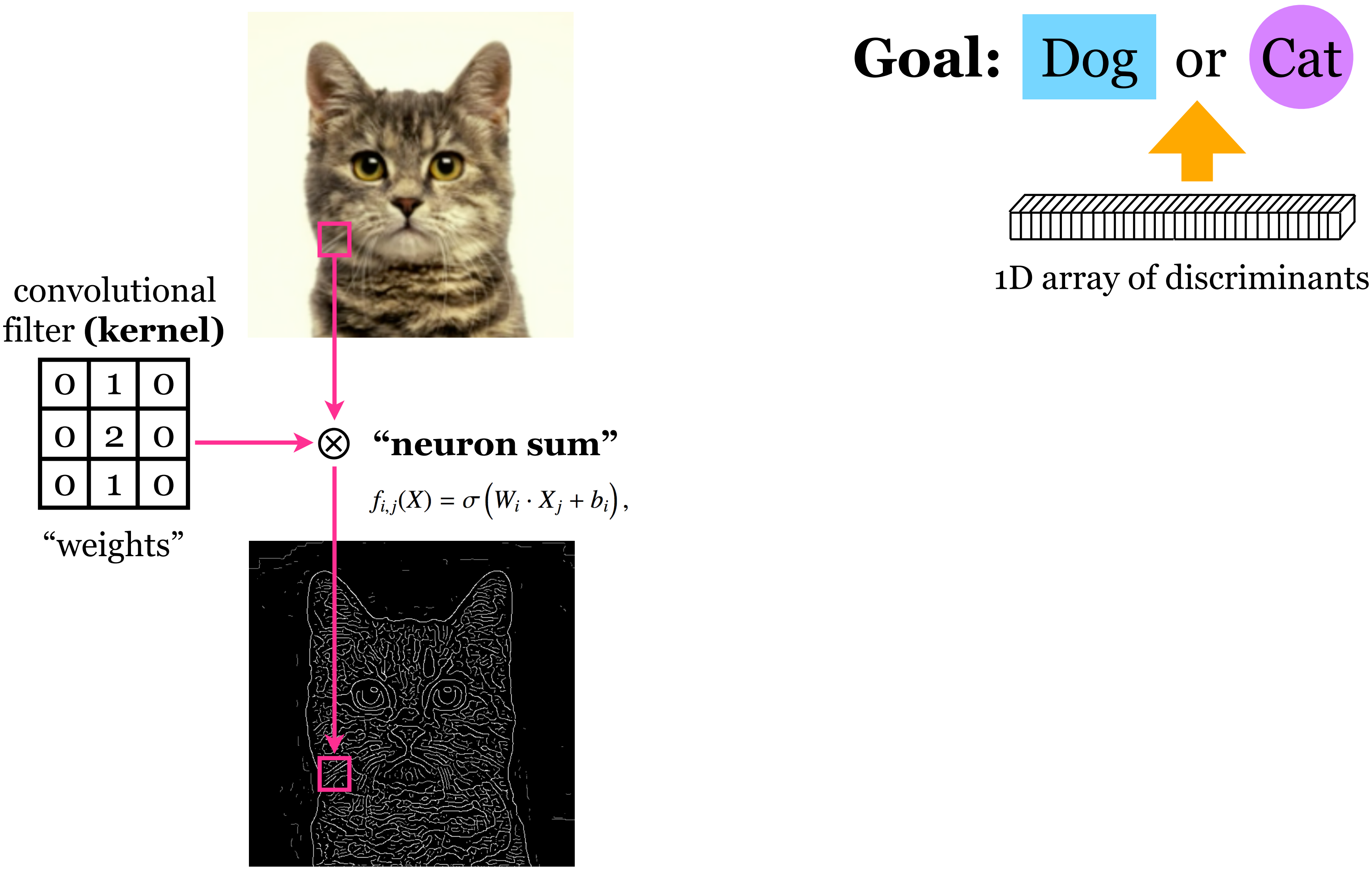
How?

This part can be done with a classic (fully-connected) neural network

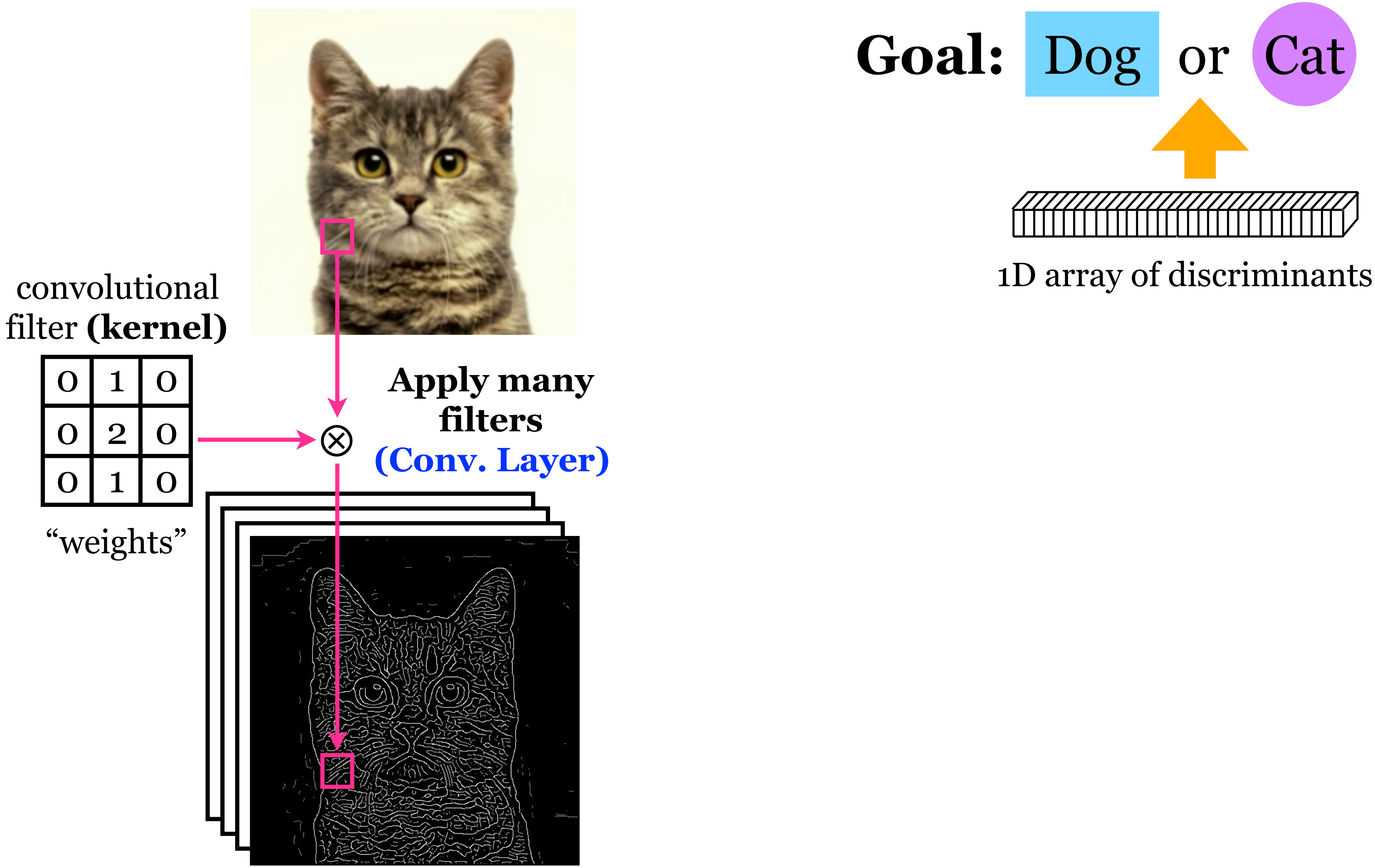
**How can we extract "features" from "image"?**

**Convolutional Neural Network**

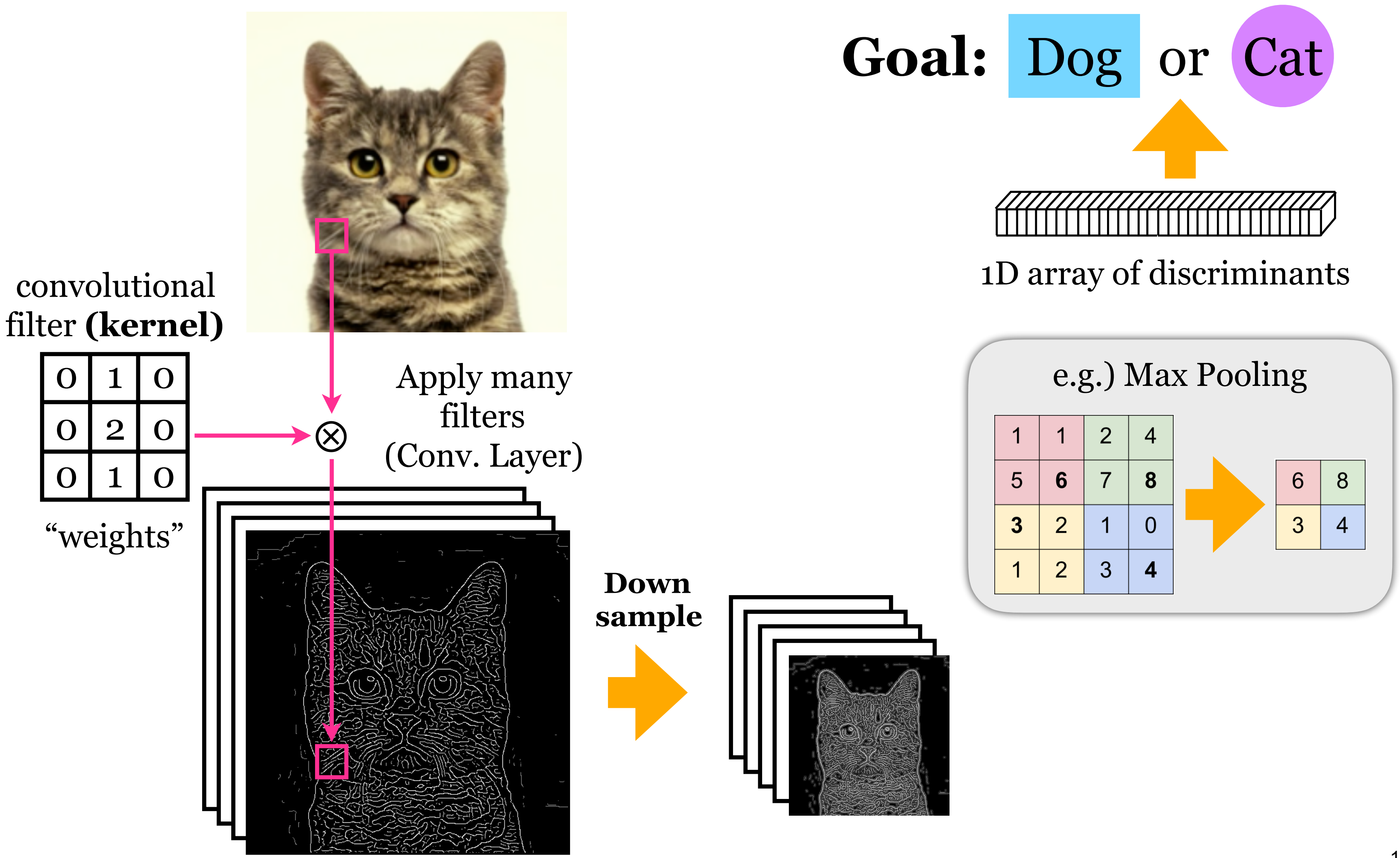
# Machine Learning Overview
## Convolutional Neural Network (CNN)



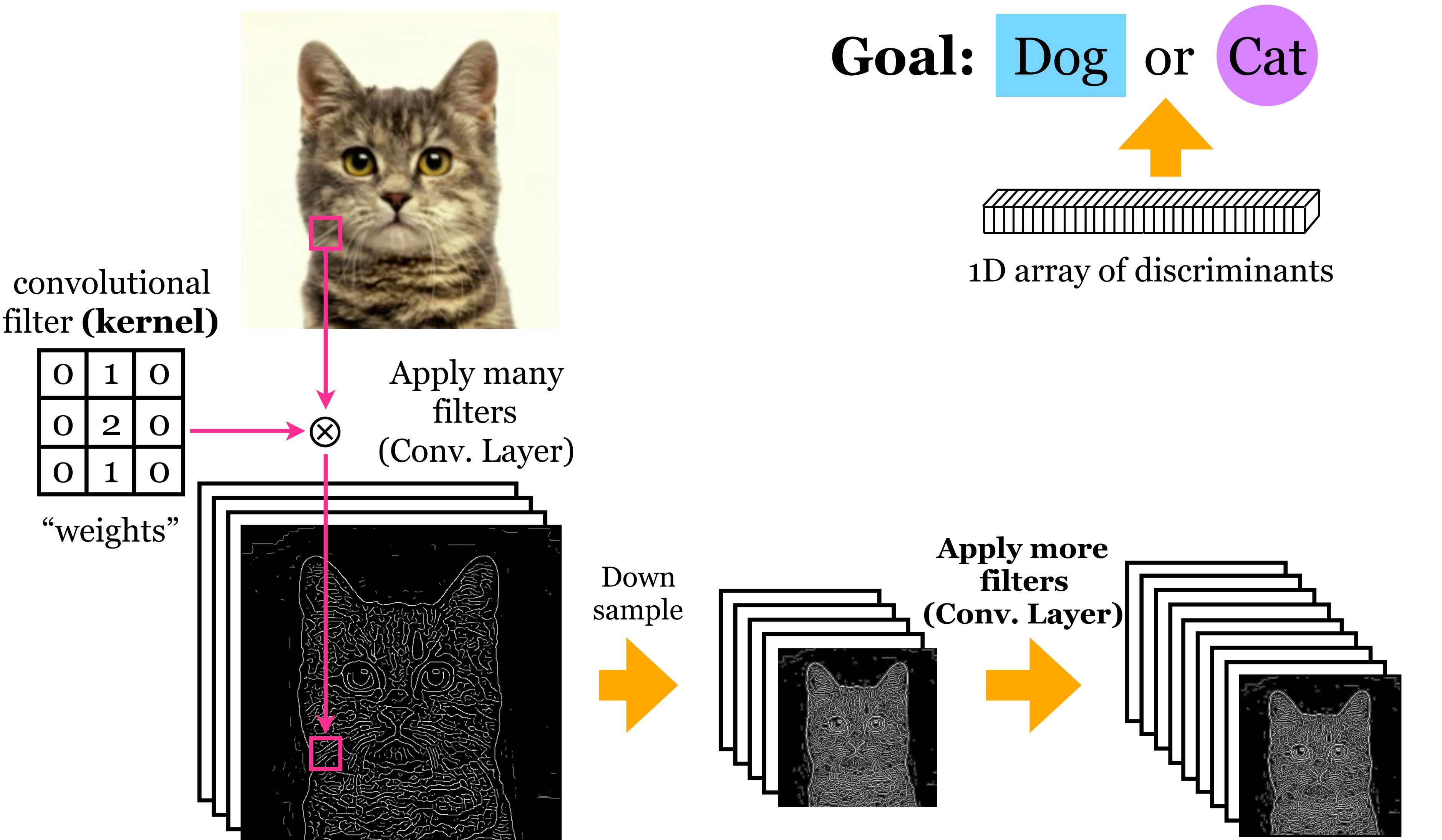
# Machine Learning Overview
## Convolutional Neural Network (CNN)



# Machine Learning Overview
## Convolutional Neural Network (CNN)

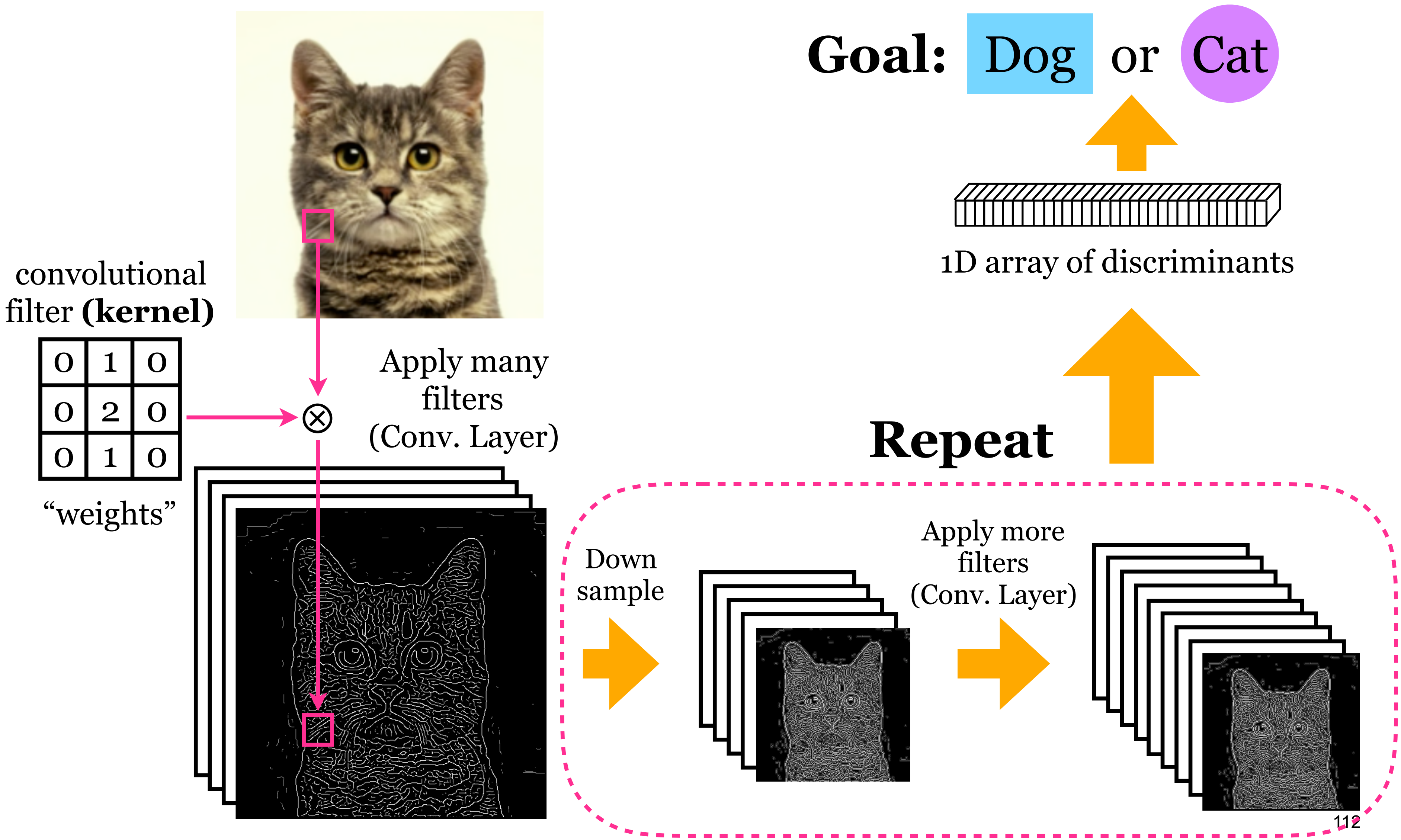


# Machine Learning Overview
## Convolutional Neural Network (CNN)

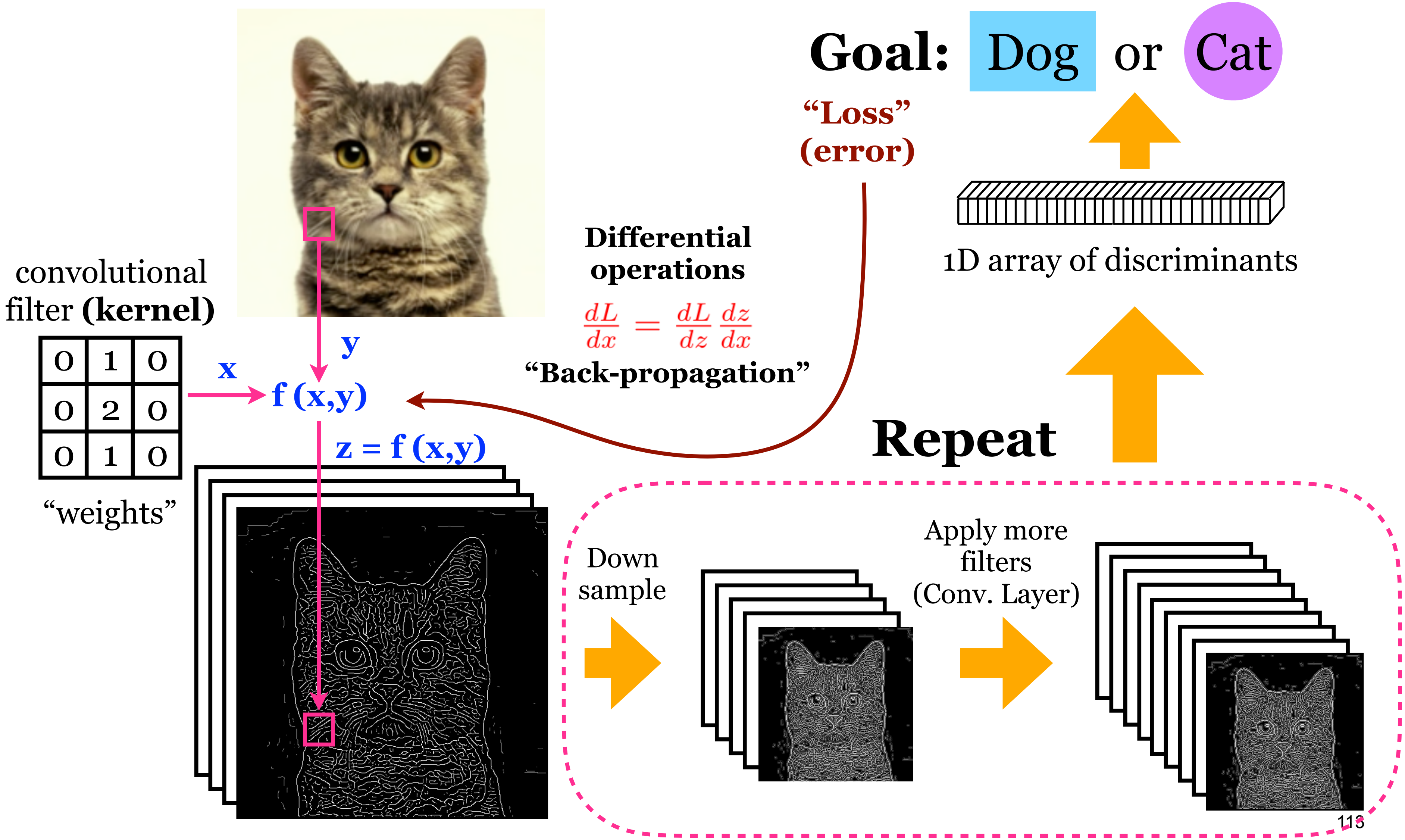


# Convolutional Neural Network (CNN)

SLAC

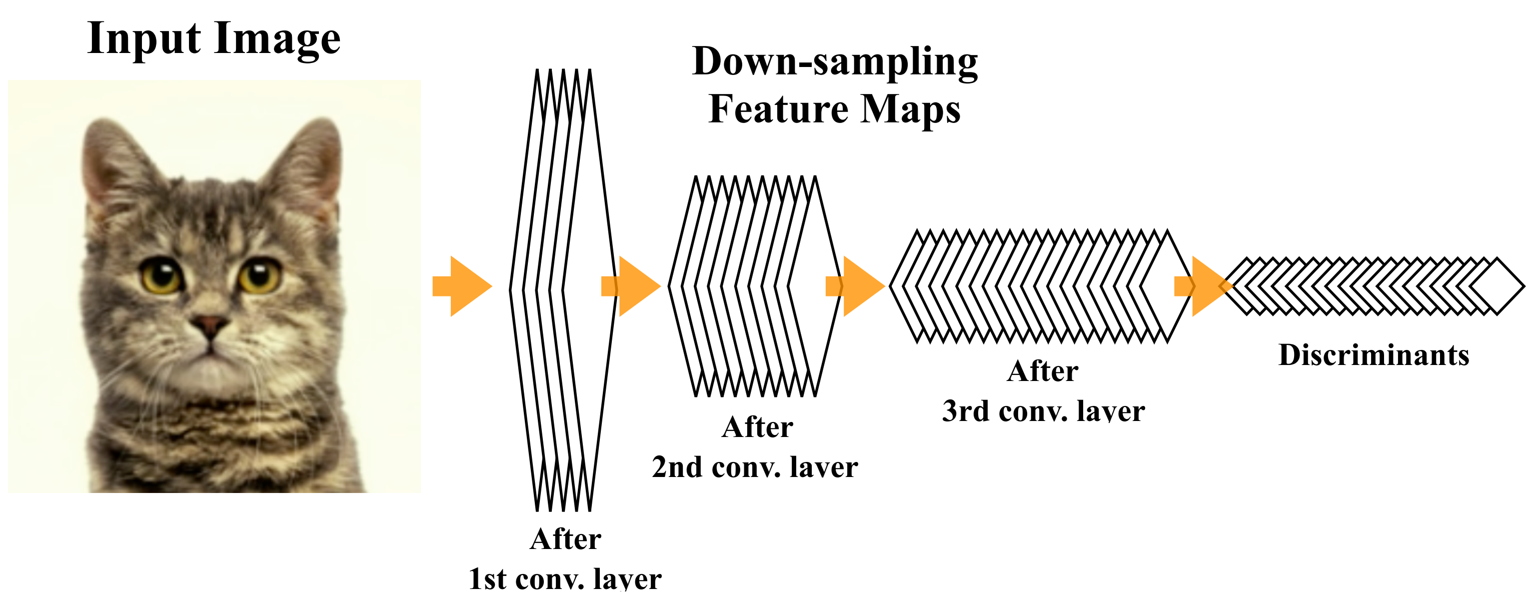


# Machine Learning Overview
## Supervised Training of CNN



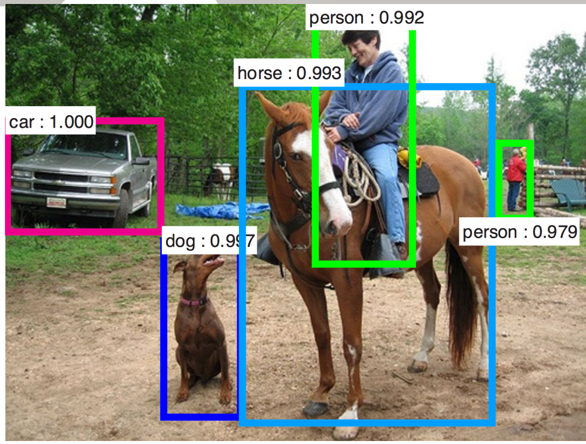
# Summarizing CNNs

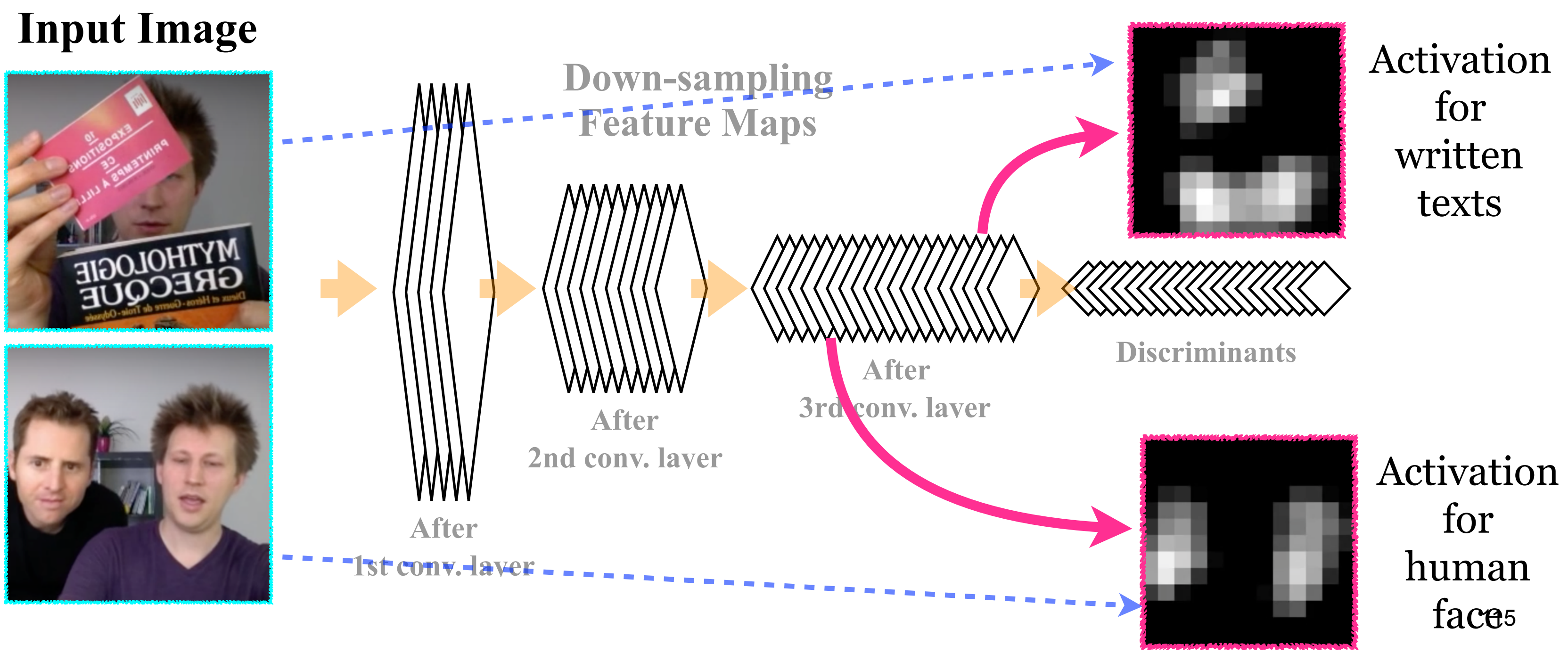
- **CNNs are "feature extraction machine"**
  - Consists of "convolution layers" with "kernels"
  - A chain of linear algebra operations = "massively parallel"
    - Suited for acceleration using many-core hardwares (e.g. GPUs)
- **CNN: data ⇔ distribution "Mapping"** (transformation)



# Beyond image classification: object detection

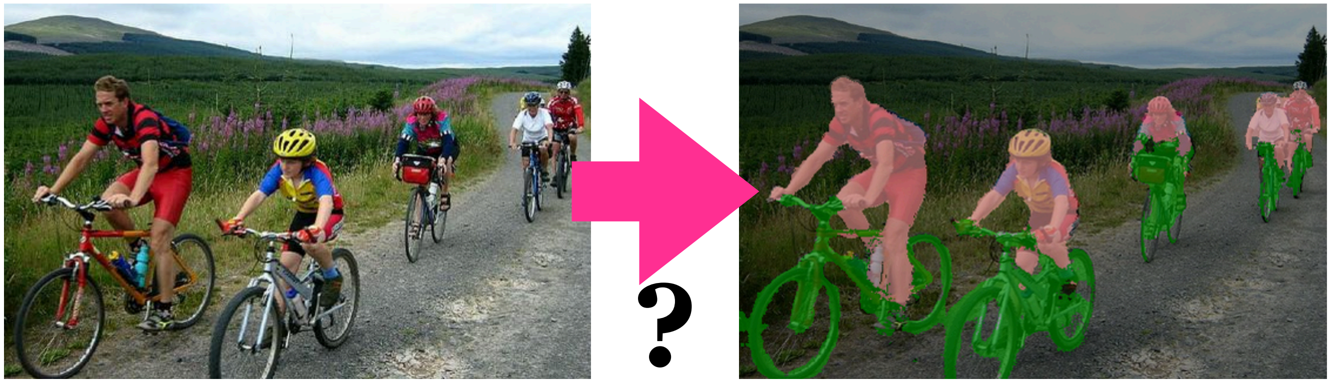
- **Object Detection**
  - Train CNN to regress "object location & size"
  - "sliding windows" to find "regions of interest"
    - With spatially contracted, feature-enhanced data, detection is much faster!

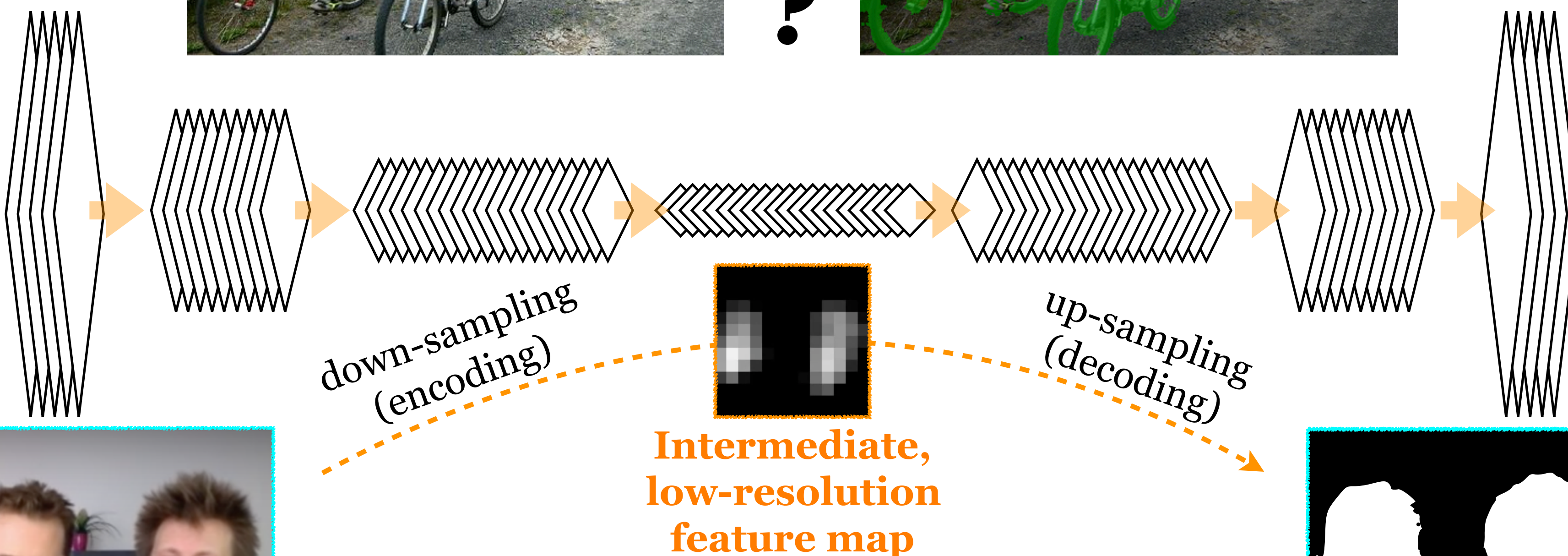


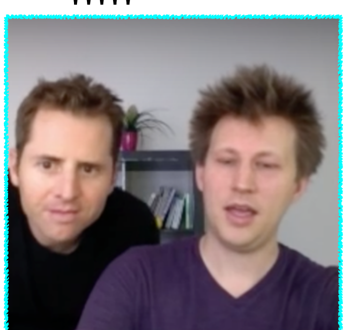



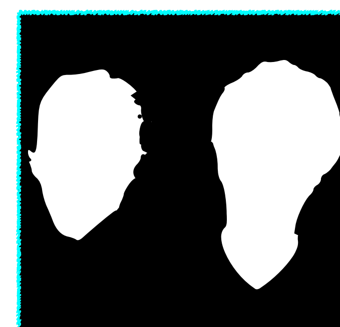
# Machine Learning Overview

## Beyond image classification: pixel segmentation





- Combine “up-sampling” + convolutions
- Outcome: “learnable” interpolation filters