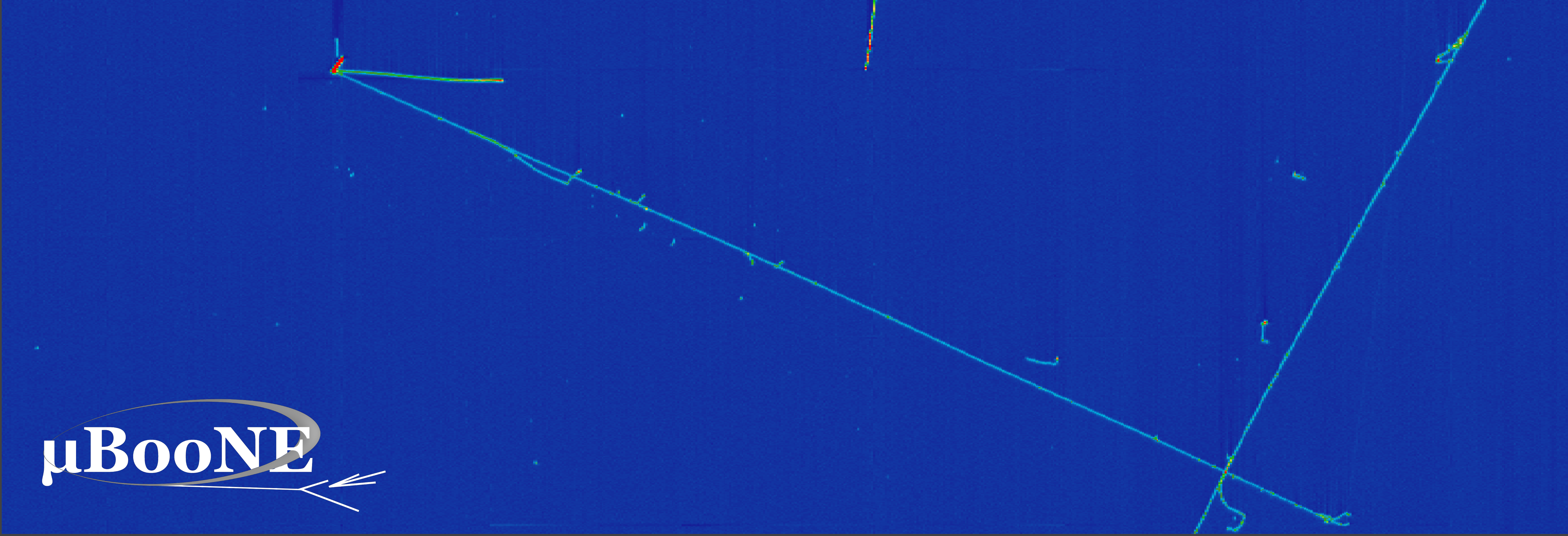


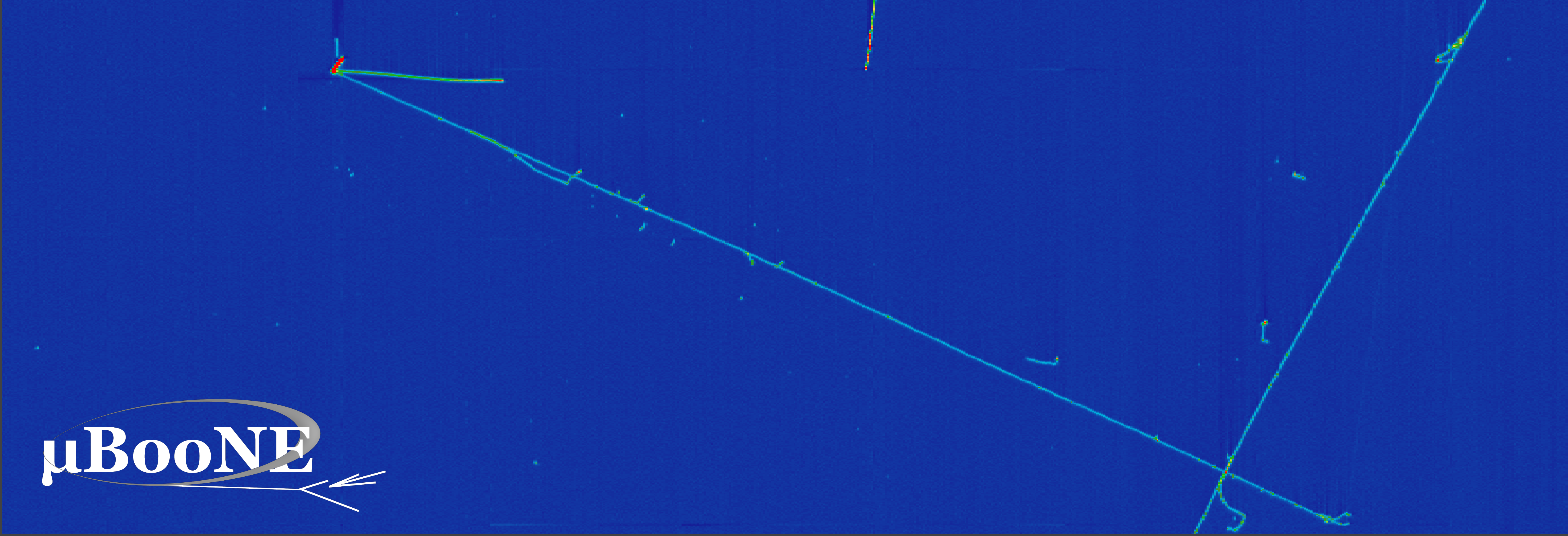

# Deep Neural Networks for Physics Data Reconstruction

*Kazuhiro Terao*
*SLAC National Accelerator Lab.*
*October 24, 2018*

U.S. DEPARTMENT OF ENERGY
Office of Science



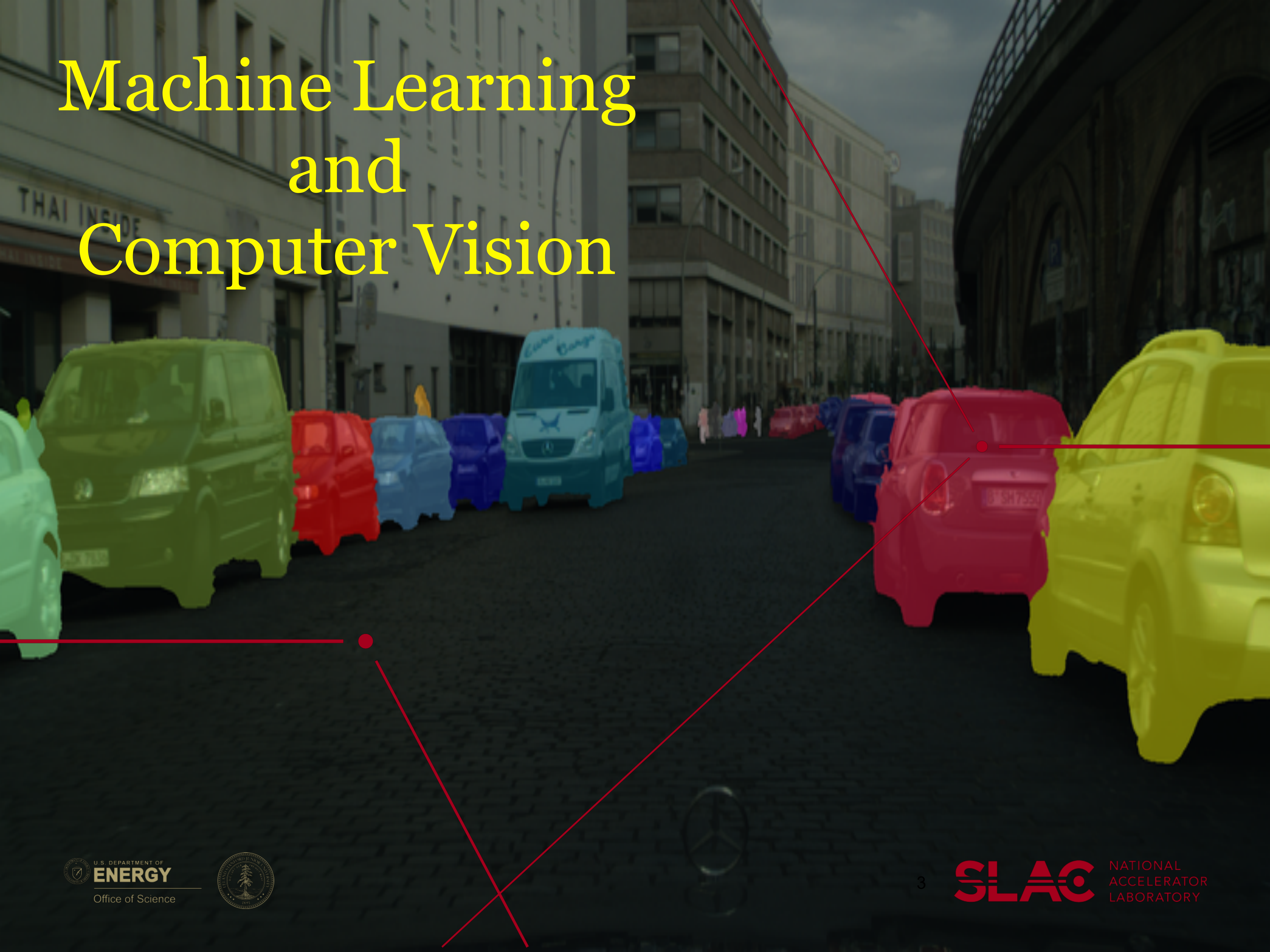
SLAC NATIONAL ACCELERATOR LABORATORY

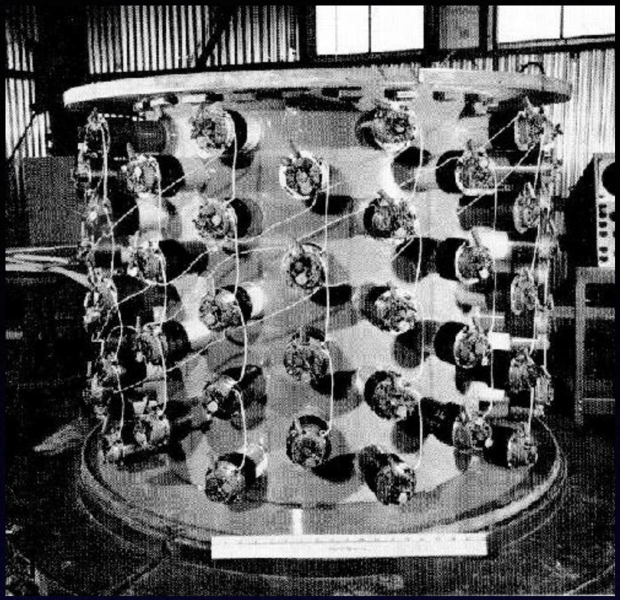


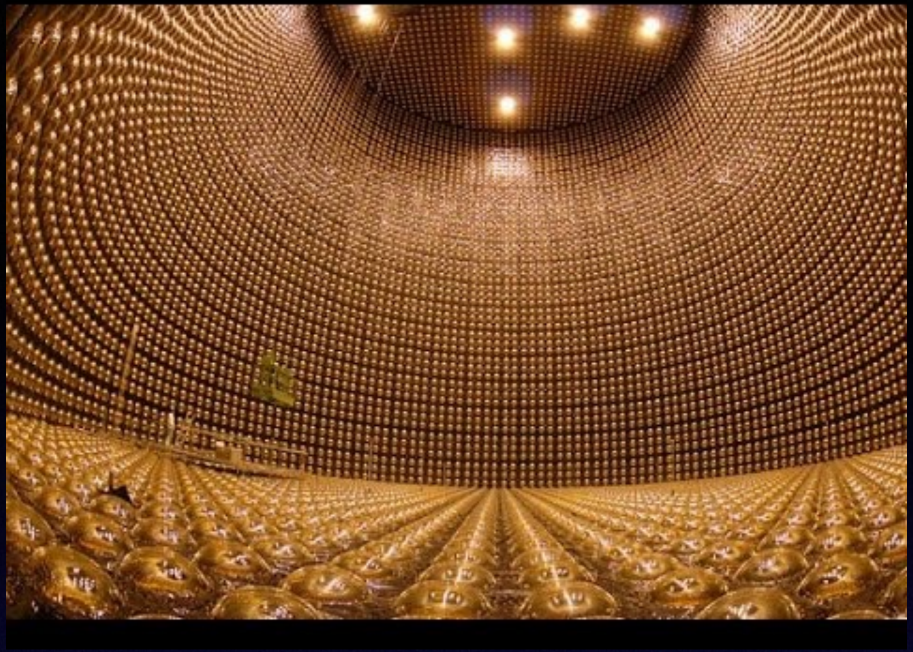

# Outline

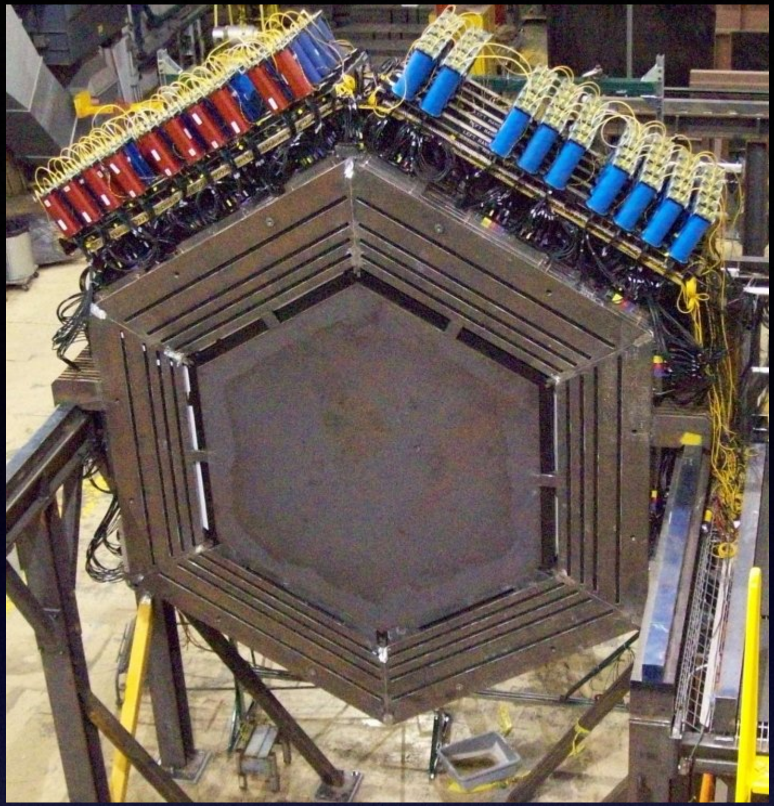
1. Machine Learning & Computer Vision
2. Applications in neutrino/NDK physics detectors
3. Wrap-up

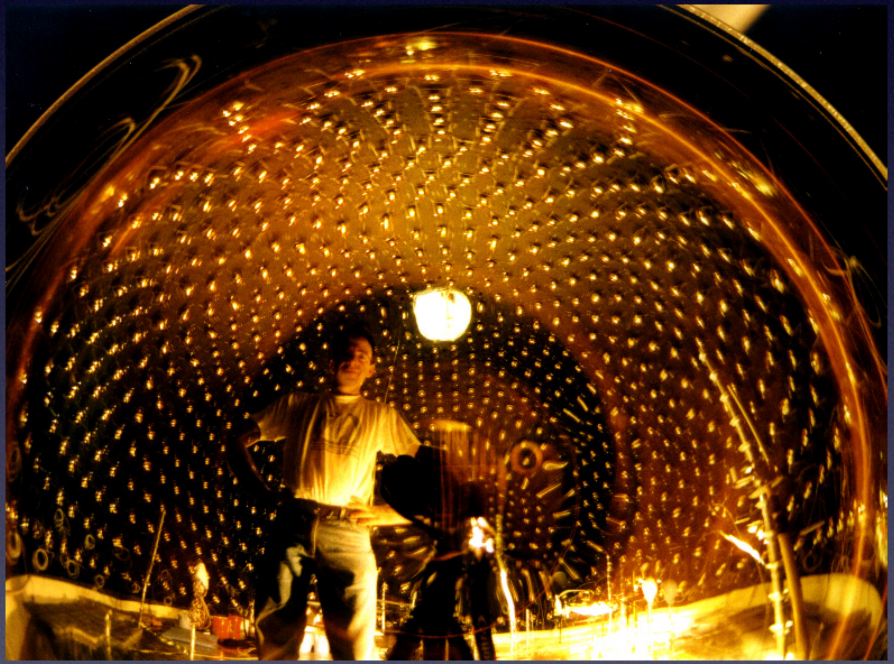
# Machine Learning and Computer Vision

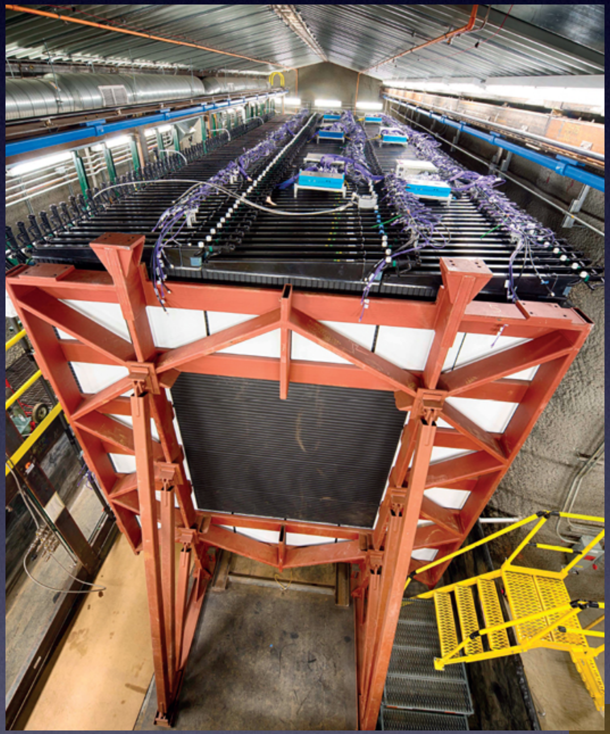


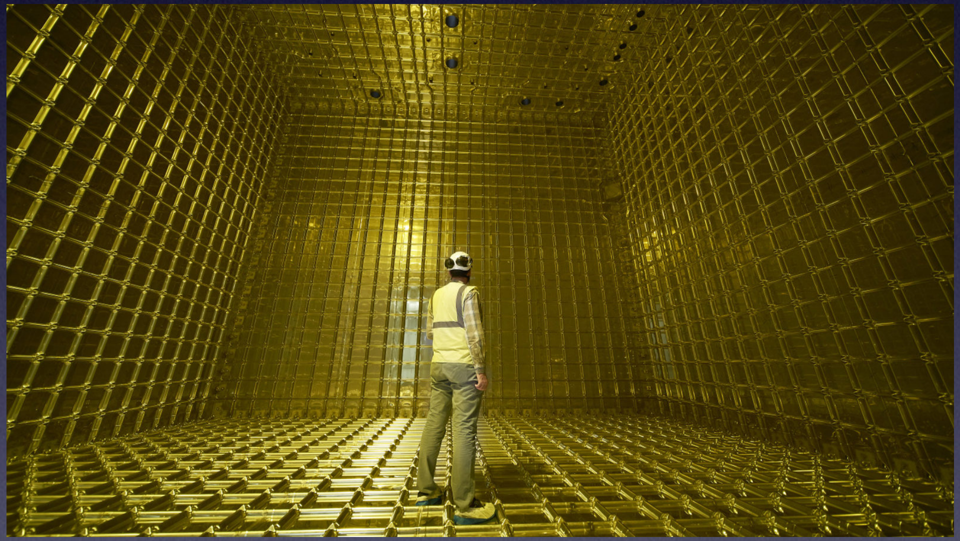


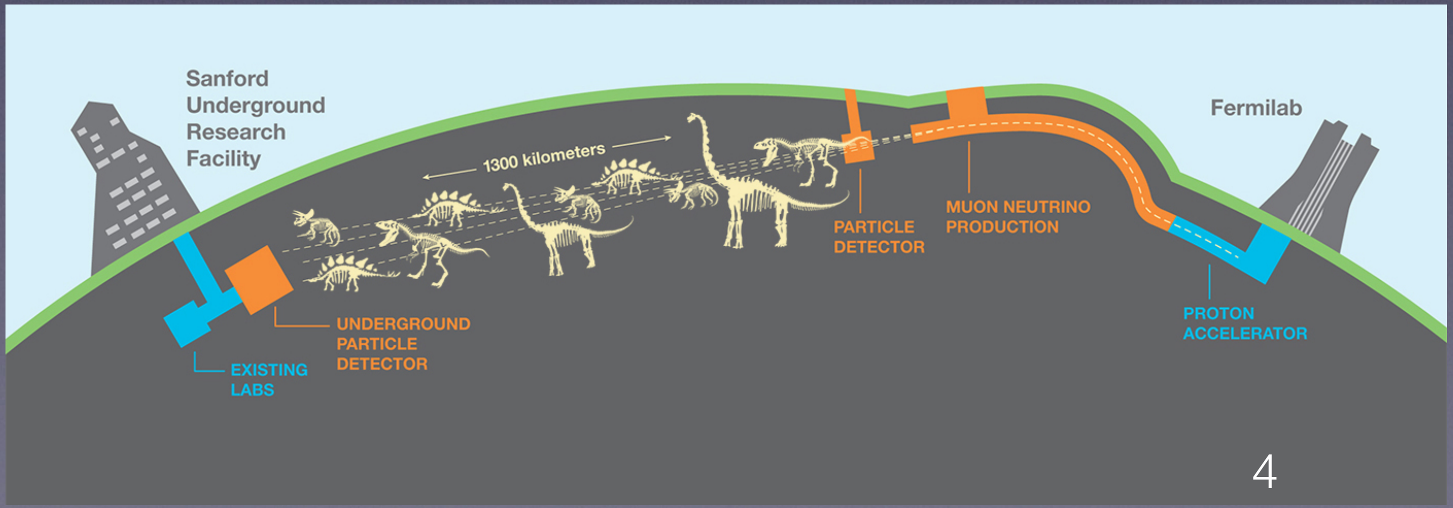


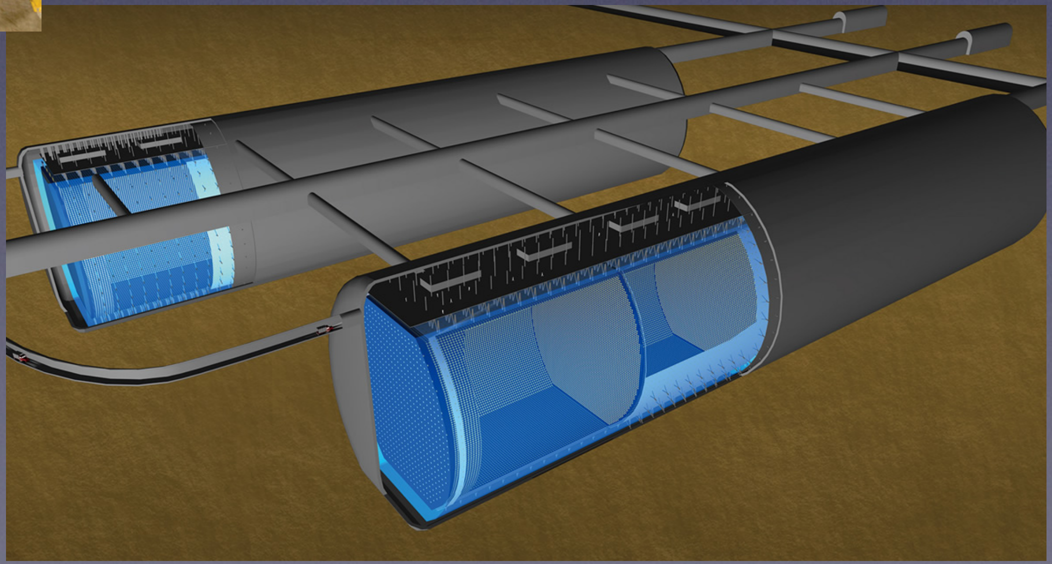


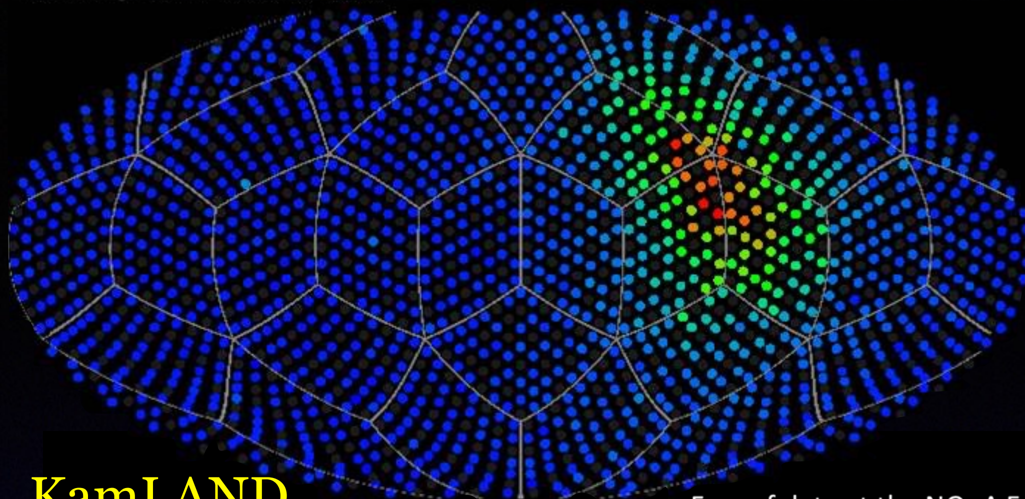


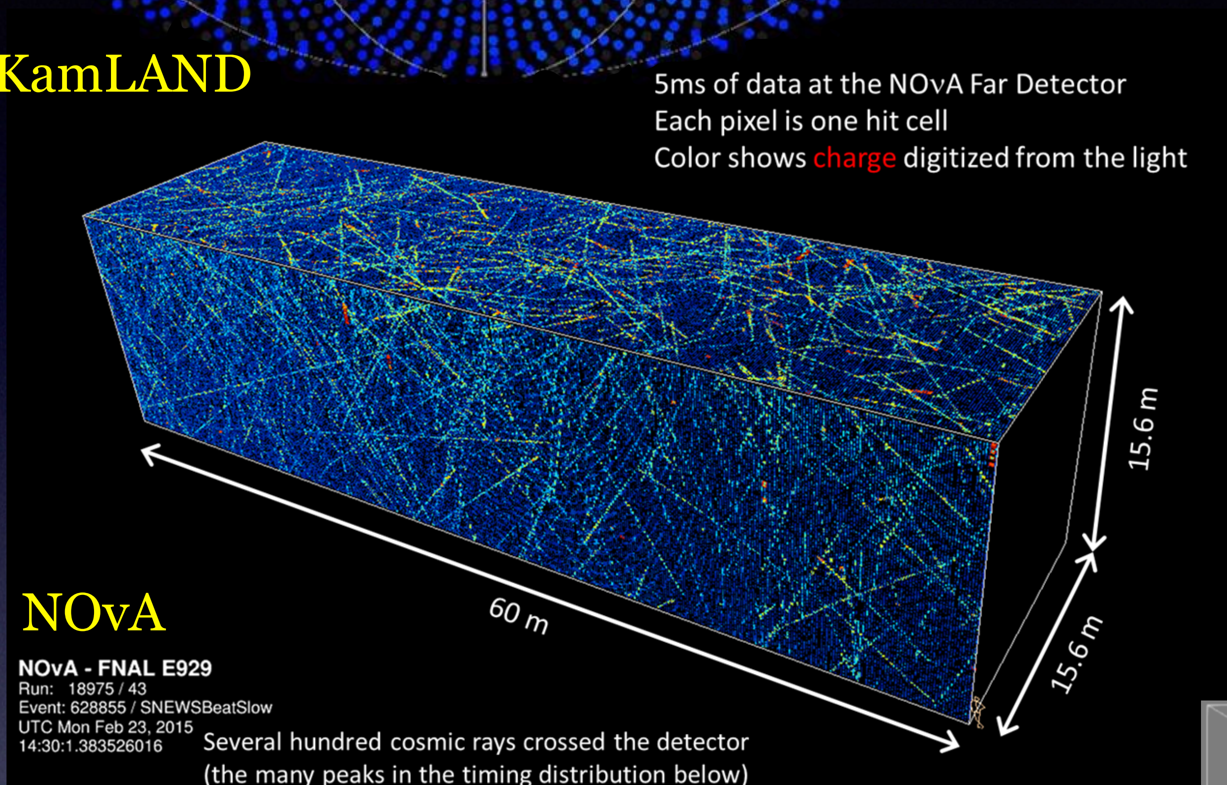


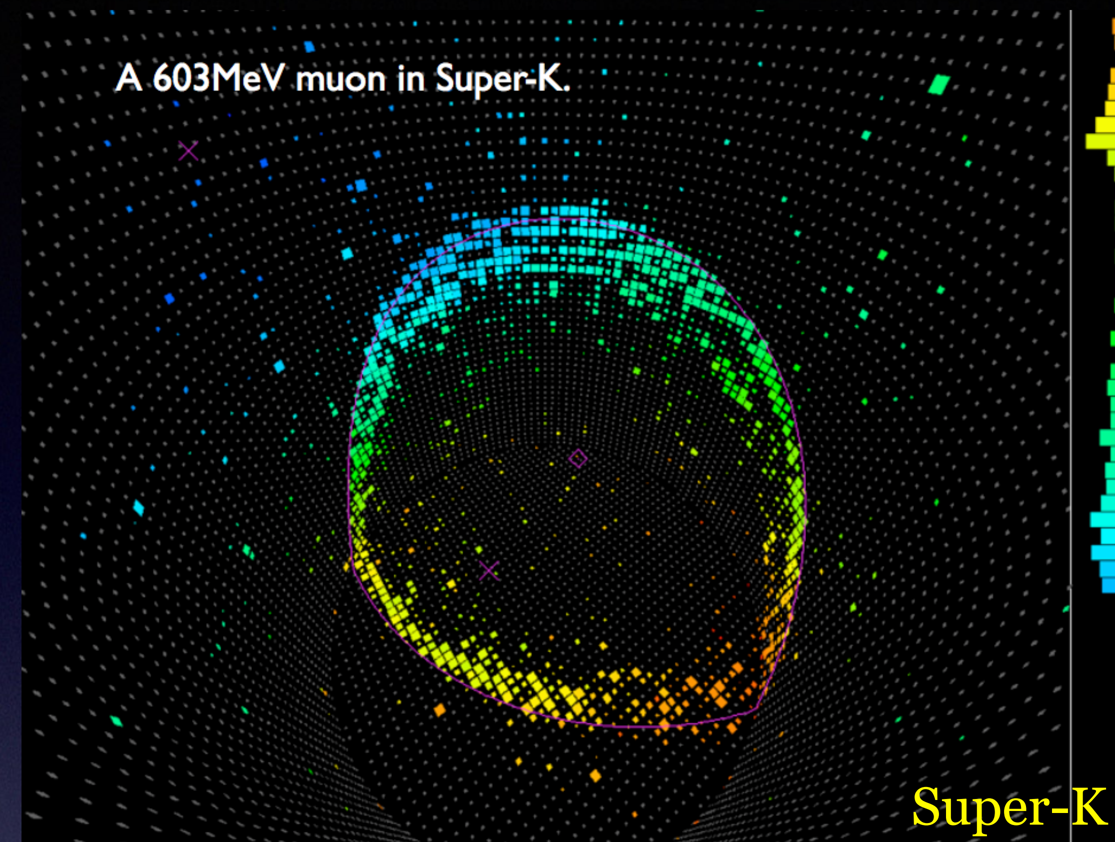


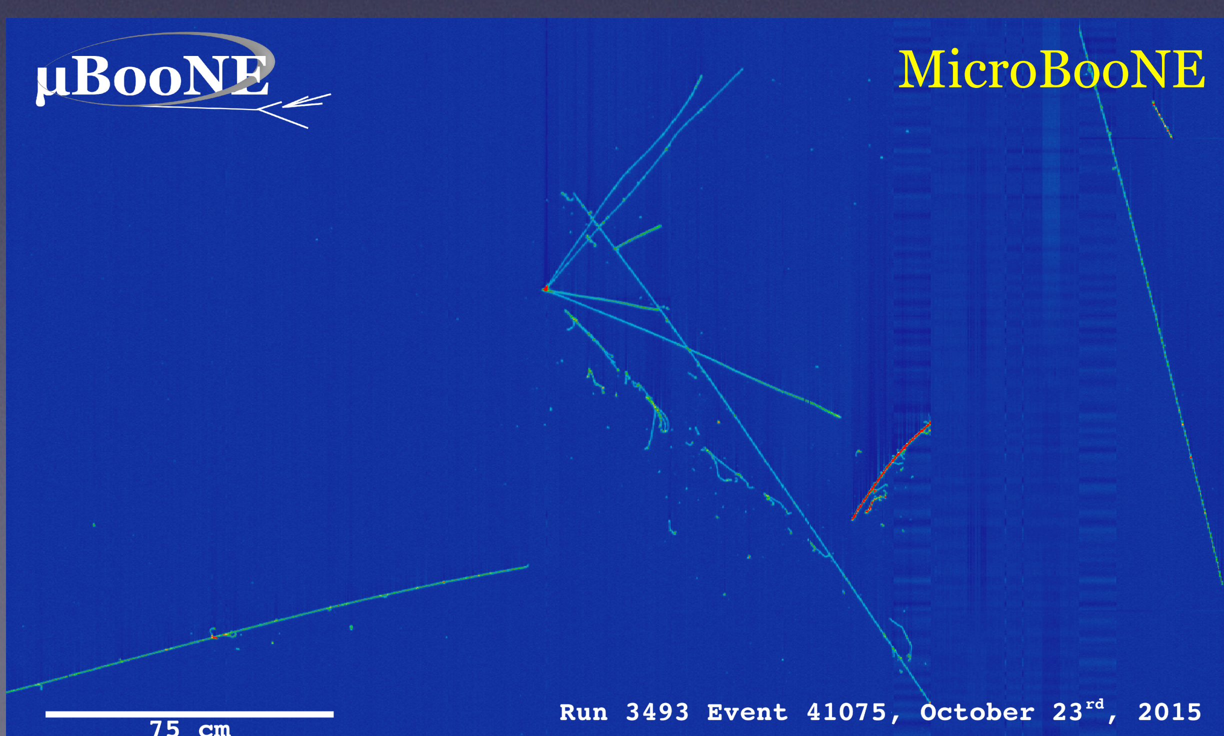

Sanford
Underground
Research
Facility
1300 kilometers
Fermilab
PARTICLE
DETECTOR
MUON NEUTRINO
PRODUCTION
UNDERGROUND
PARTICLE
DETECTOR
EXISTING
LABS
PROTON
ACCELERATOR

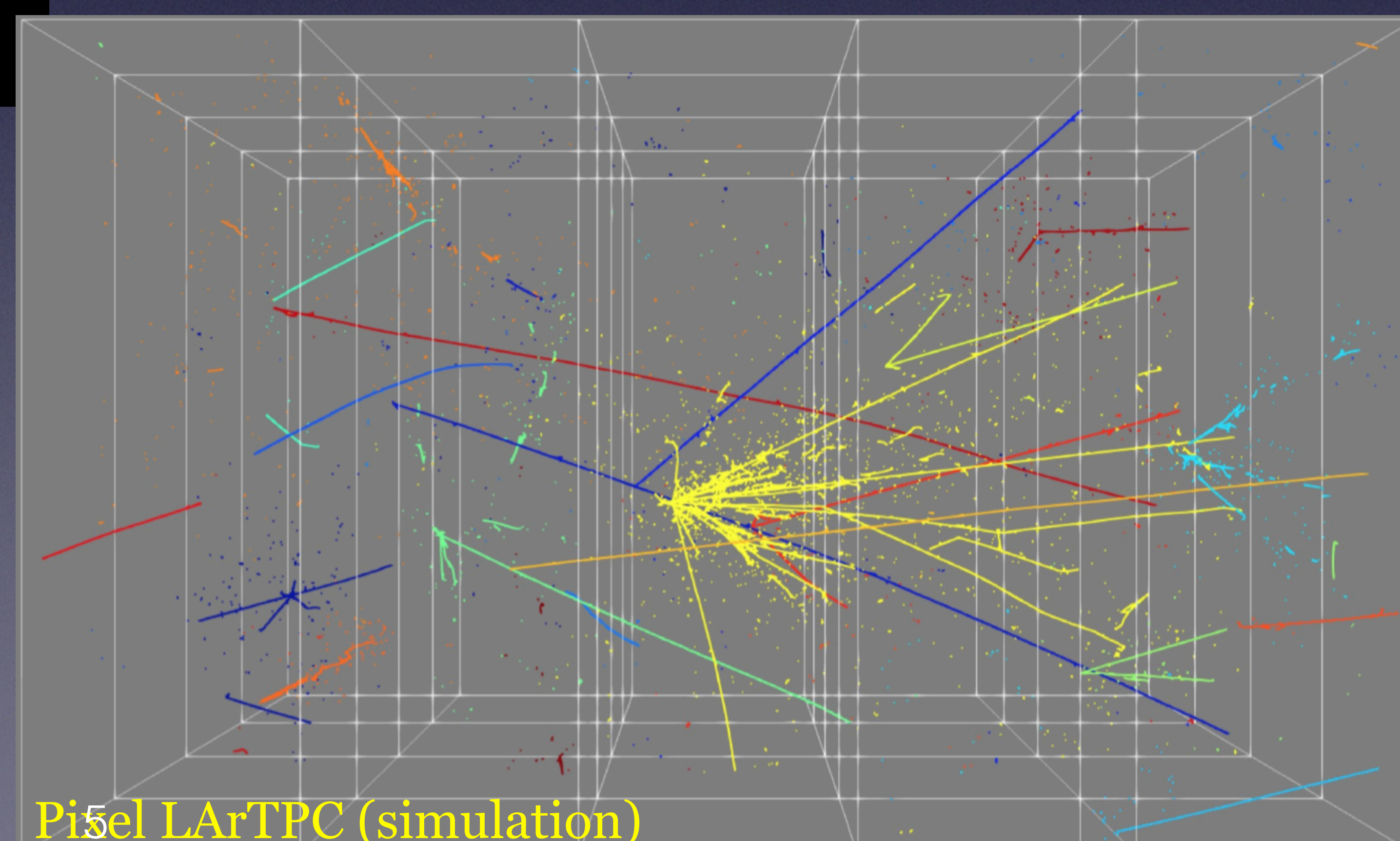



KamLAND

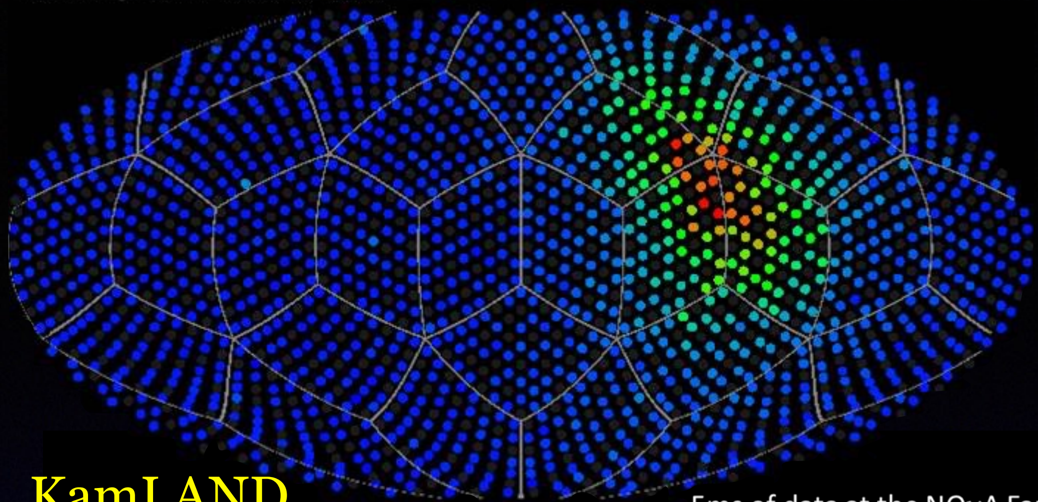

5ms of data at the NOvA Far Detector
Each pixel is one hit cell
Color shows charge digitized from the light
15.6 m
15.6 m
60 m
NOvA
NOvA - FNAL E929
Run: 18975 / 43
Event: 628855 / SNEWSBeatSlow
UTC Mon Feb 23, 2015
14:30:1.383526016
Several hundred cosmic rays crossed the detector
(the many peaks in the timing distribution below)

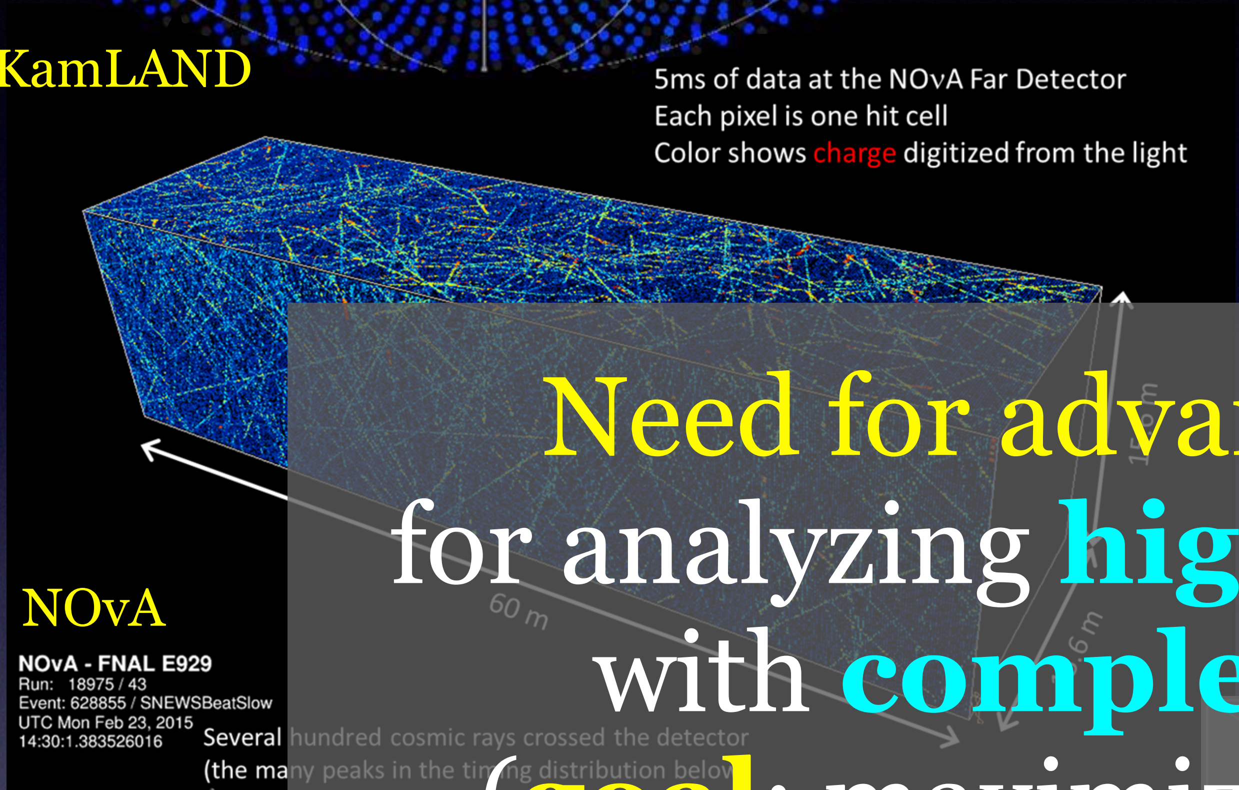

A 603MeV muon in Super-K.
Super-K

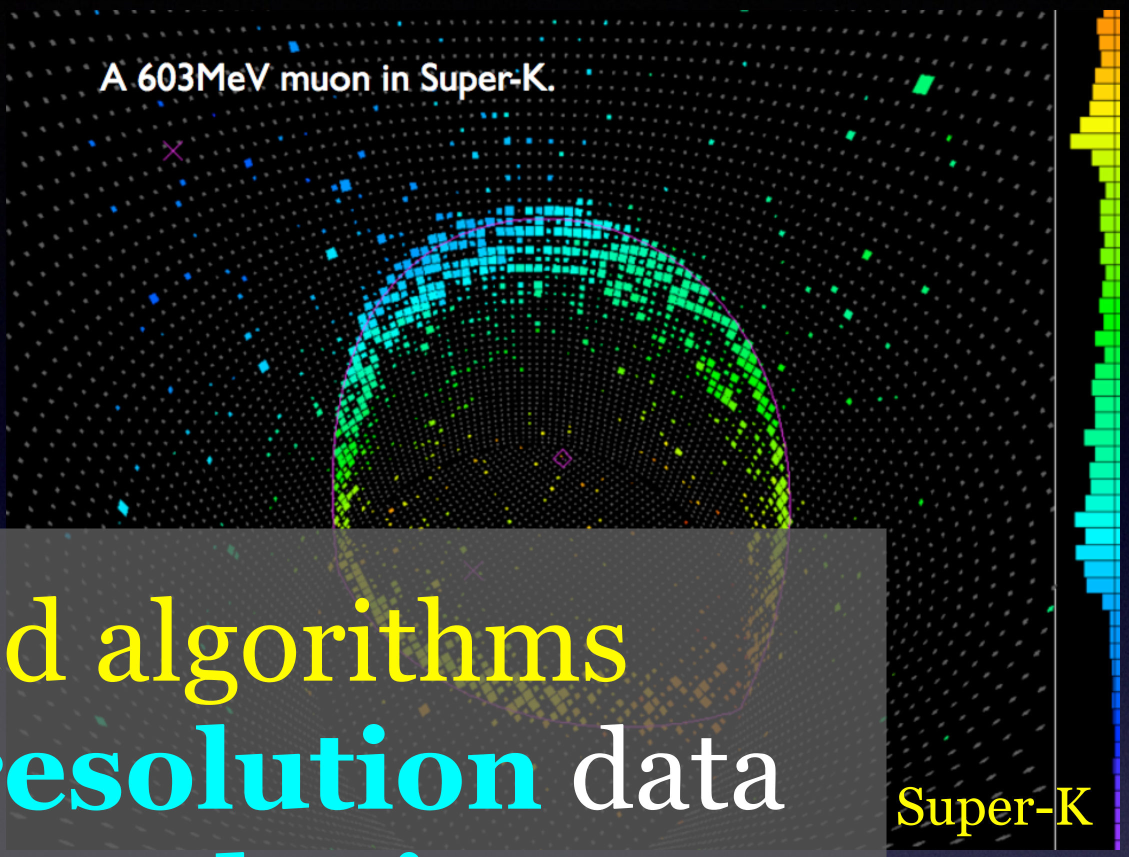

μBooNE
MicroBooNE
75 cm
Run 3493 Event 41075, October 23rd, 2015


Pixel LArTPC (simulation)

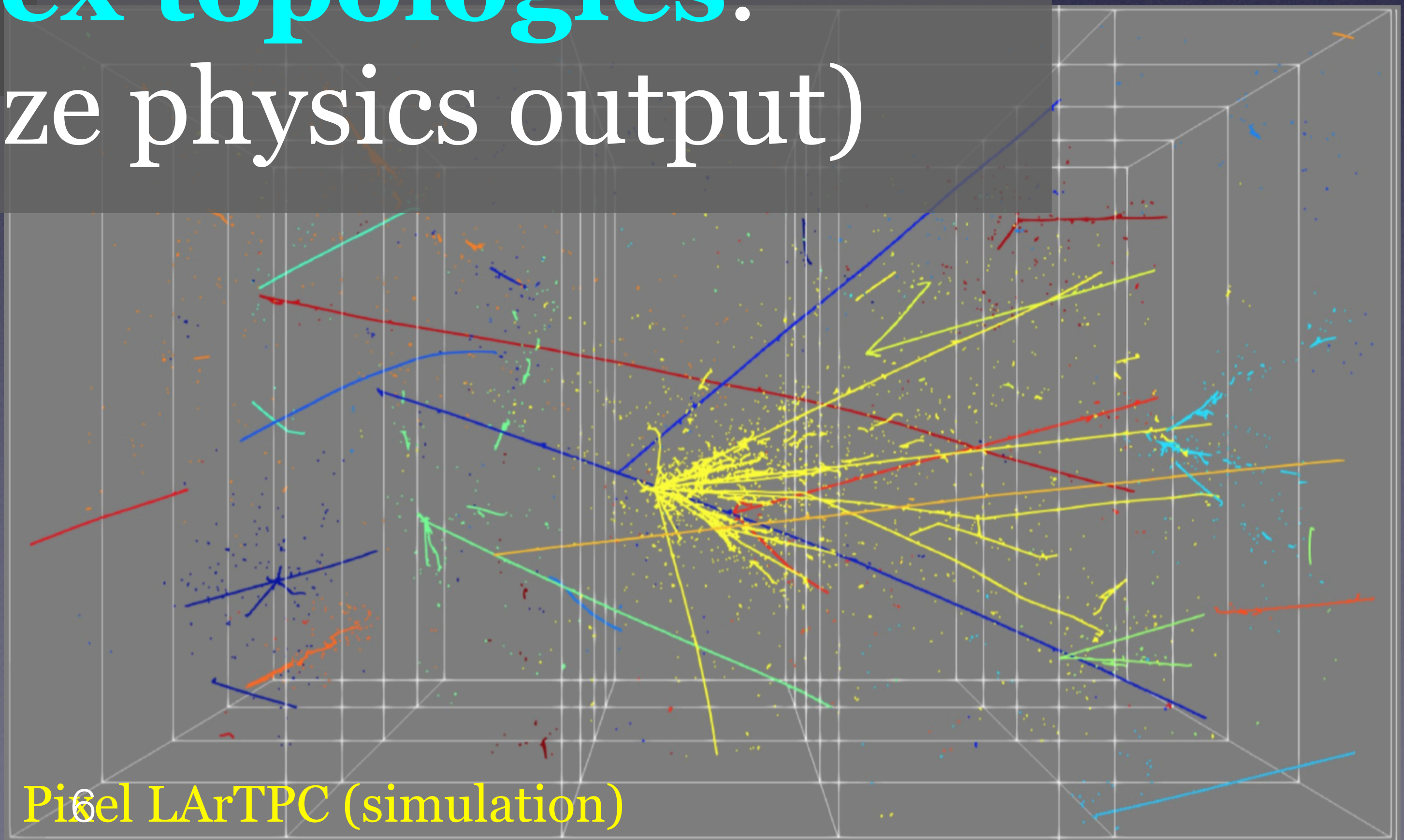
Need for advanced algorithms
for analyzing **high resolution** data
with **complex topologies**.
(**goal**: maximize physics output)



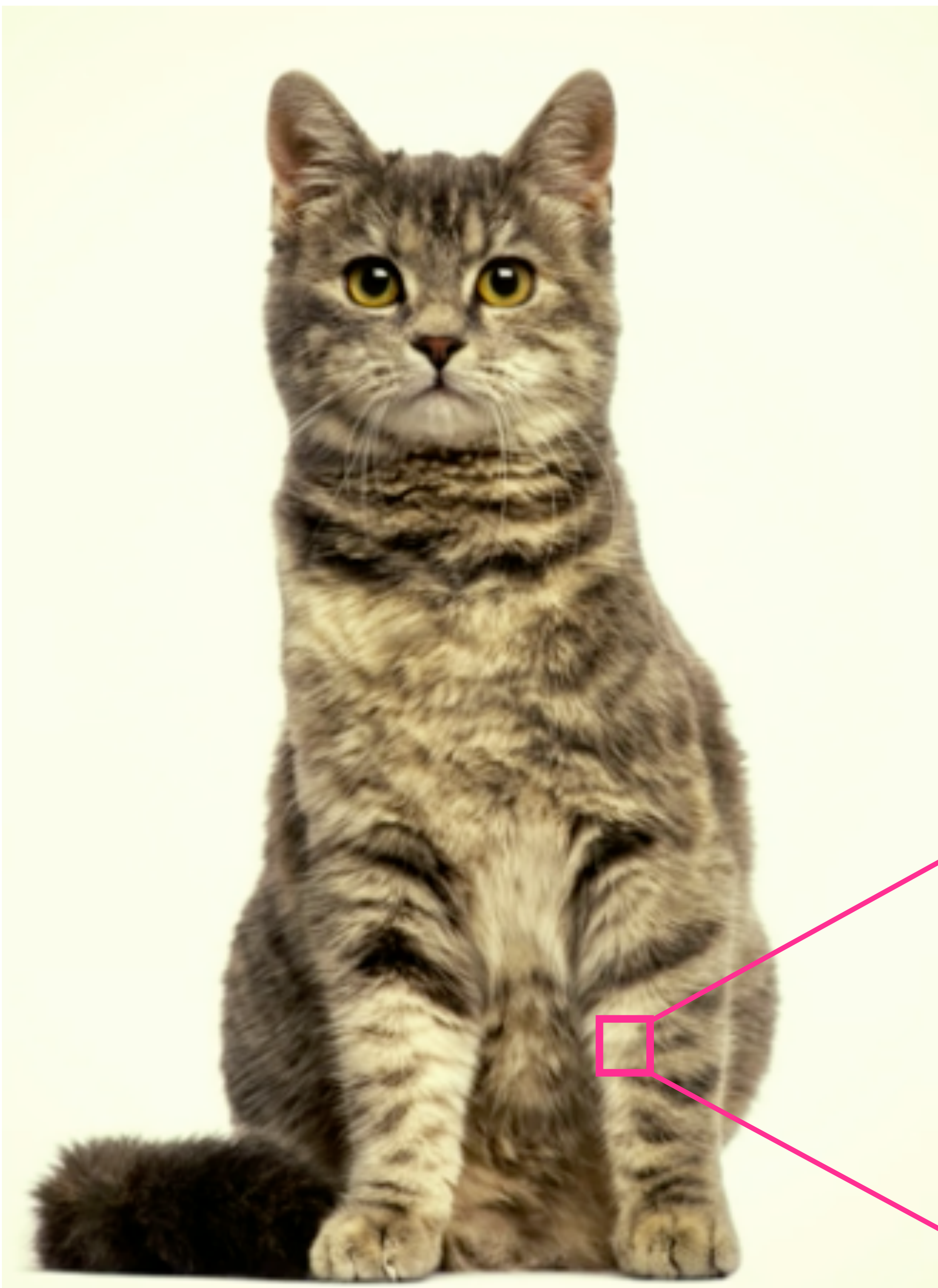







## Challenge in Computer Vision



How to write an algorithm to identify a cat?

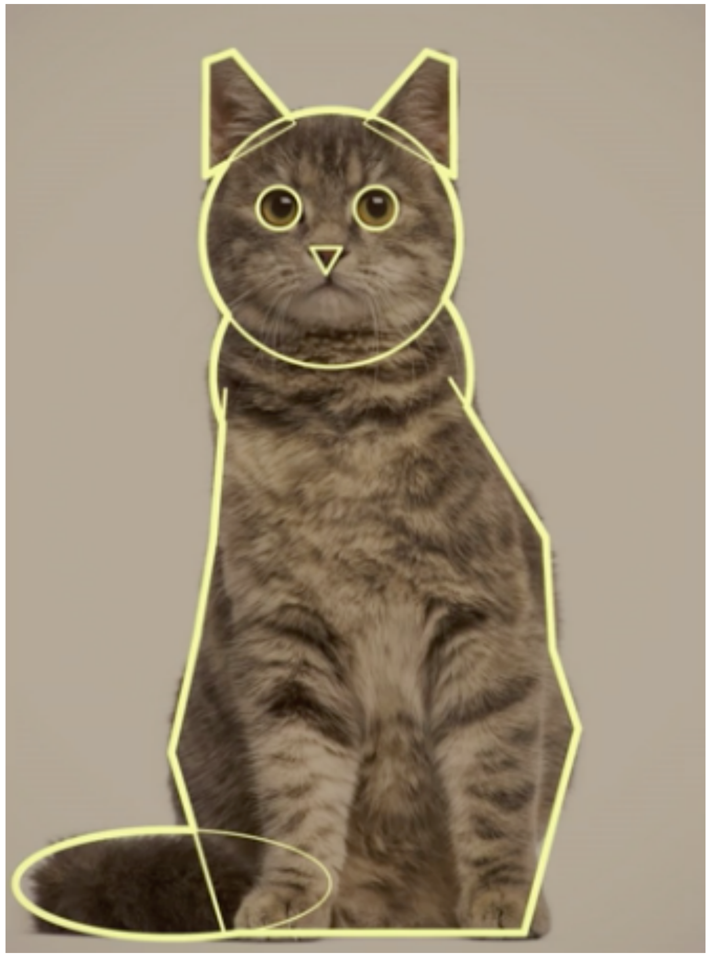
... very hard task ...

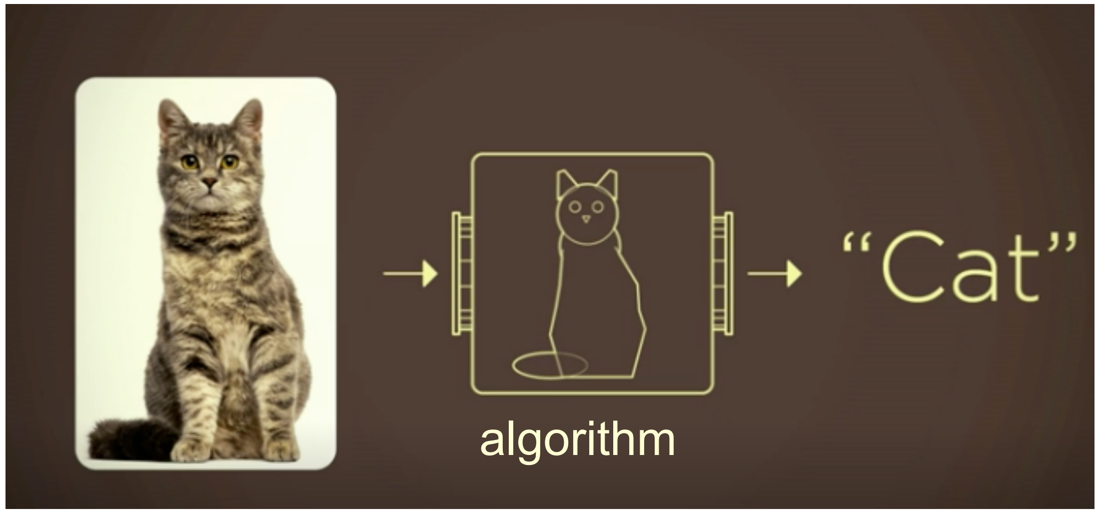
| | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 16 | 08 | 67 | 15 | 83 | 09 | 40 | 19 | 40 | 11 | 31 | 35 | 60 | 43 | 66 | 14 | 48 | 08 | 60 | 13 |
| 37 | 52 | 77 | 23 | 22 | 74 | 09 | 90 | 36 | 12 | 29 | 39 | 78 | 31 | 71 | 73 | 22 | 50 | 92 | 35 |
| 35 | 42 | 48 | 72 | 85 | 27 | 79 | 08 | 41 | 31 | 09 | 53 | 05 | 40 | 04 | 31 | 91 | 56 | 26 | 85 |
| 68 | 36 | 43 | 54 | 21 | 33 | 81 | 30 | 72 | 06 | 79 | 34 | 39 | 59 | 70 | 03 | 24 | 91 | 03 | 40 |
| 79 | 60 | 10 | 25 | 54 | 71 | 24 | 50 | 87 | 88 | 47 | 68 | 31 | 42 | 09 | 77 | 40 | 07 | 26 | 73 |
| 18 | 55 | 38 | 73 | 50 | 47 | 22 | 21 | 88 | 78 | 02 | 95 | 19 | 59 | 60 | 93 | 73 | 40 | 67 | 99 |
| 54 | 07 | 67 | 38 | 55 | 51 | 26 | 81 | 43 | 66 | 89 | 69 | 92 | 94 | 50 | 08 | 94 | 63 | 33 | 66 |
| 71 | 95 | 38 | 46 | 63 | 07 | 66 | 68 | 41 | 49 | 34 | 33 | 66 | 76 | 68 | 97 | 53 | 18 | 72 | 21 |
| 38 | 64 | 86 | 66 | 06 | 68 | 13 | 01 | 89 | 00 | 80 | 70 | 21 | 27 | 14 | 90 | 80 | 95 | 31 | 68 |
| 04 | 28 | 93 | 88 | 02 | 97 | 92 | 41 | 21 | 54 | 24 | 33 | 97 | 10 | 33 | 47 | 24 | 08 | 12 | 76 |
| 75 | 37 | 62 | 42 | 88 | 15 | 02 | 57 | 20 | 43 | 09 | 71 | 54 | 73 | 29 | 57 | 23 | 81 | 99 | 41 |
| 29 | 28 | 57 | 02 | 84 | 20 | 31 | 97 | 41 | 73 | 19 | 29 | 17 | 28 | 99 | 16 | 23 | 19 | 53 | 53 |
| 95 | 05 | 34 | 86 | 46 | 18 | 95 | 65 | 62 | 28 | 62 | 95 | 35 | 84 | 18 | 22 | 81 | 45 | 10 | 12 |
| 69 | 18 | 34 | 46 | 77 | 60 | 28 | 62 | 16 | 61 | 72 | 19 | 88 | 14 | 43 | 23 | 64 | 43 | 35 | 00 |
| 76 | 15 | 68 | 89 | 13 | 74 | 48 | 90 | 12 | 59 | 02 | 31 | 14 | 34 | 77 | 47 | 04 | 69 | 99 | 66 |
| 70 | 01 | 05 | 77 | 88 | 20 | 63 | 57 | 41 | 50 | 68 | 04 | 30 | 62 | 09 | 67 | 61 | 86 | 31 | 43 |
| 36 | 76 | 07 | 95 | 11 | 52 | 04 | 91 | 58 | 59 | 30 | 09 | 46 | 95 | 31 | 71 | 43 | 26 | 48 | 19 |
| 81 | 01 | 86 | 71 | 64 | 31 | 49 | 99 | 60 | 63 | 97 | 61 | 43 | 86 | 36 | 53 | 82 | 31 | 00 | 52 |
| 63 | 78 | 18 | 10 | 79 | 39 | 77 | 28 | 39 | 17 | 76 | 81 | 93 | 35 | 02 | 78 | 10 | 30 | 35 | 75 |
| 71 | 73 | 71 | 85 | 86 | 24 | 93 | 75 | 35 | 70 | 30 | 16 | 07 | 35 | 08 | 61 | 82 | 85 | 95 | 22 |

Taken from slides by Fei-Fei’s TED talk

## Development Workflow for non-ML algorithms

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






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



Taken from slides by Fei-Fei's TED talk

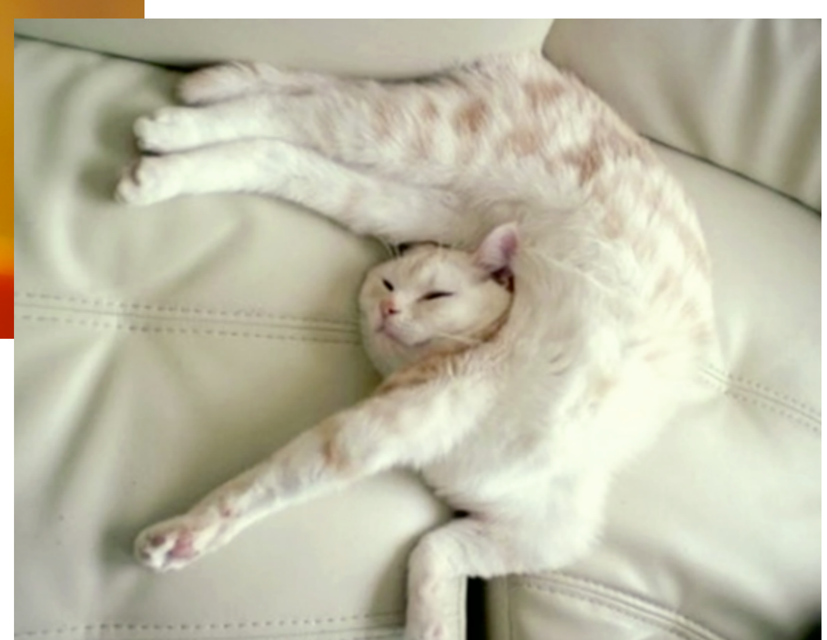
# Machine Learning Overview
## Challenge in Computer Vision

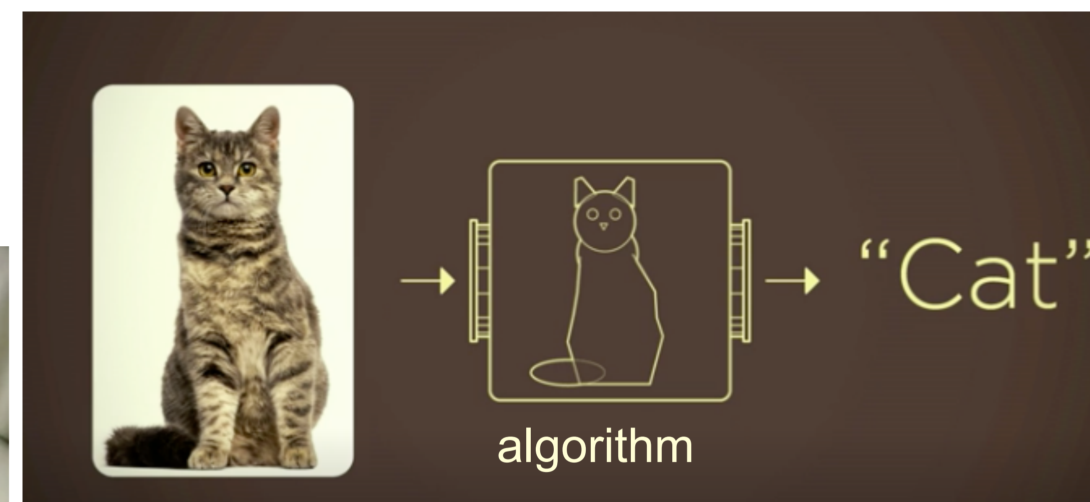
### **Development Workflow** for **non-ML** algorithms

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**
(particle escaping
fiducial volume)

**Stretching cat**
(Nuclear FSI)






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



Taken from slides by Fei-Fei's TED talk

# Challenge in Computer Vision

SLAC

## 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**

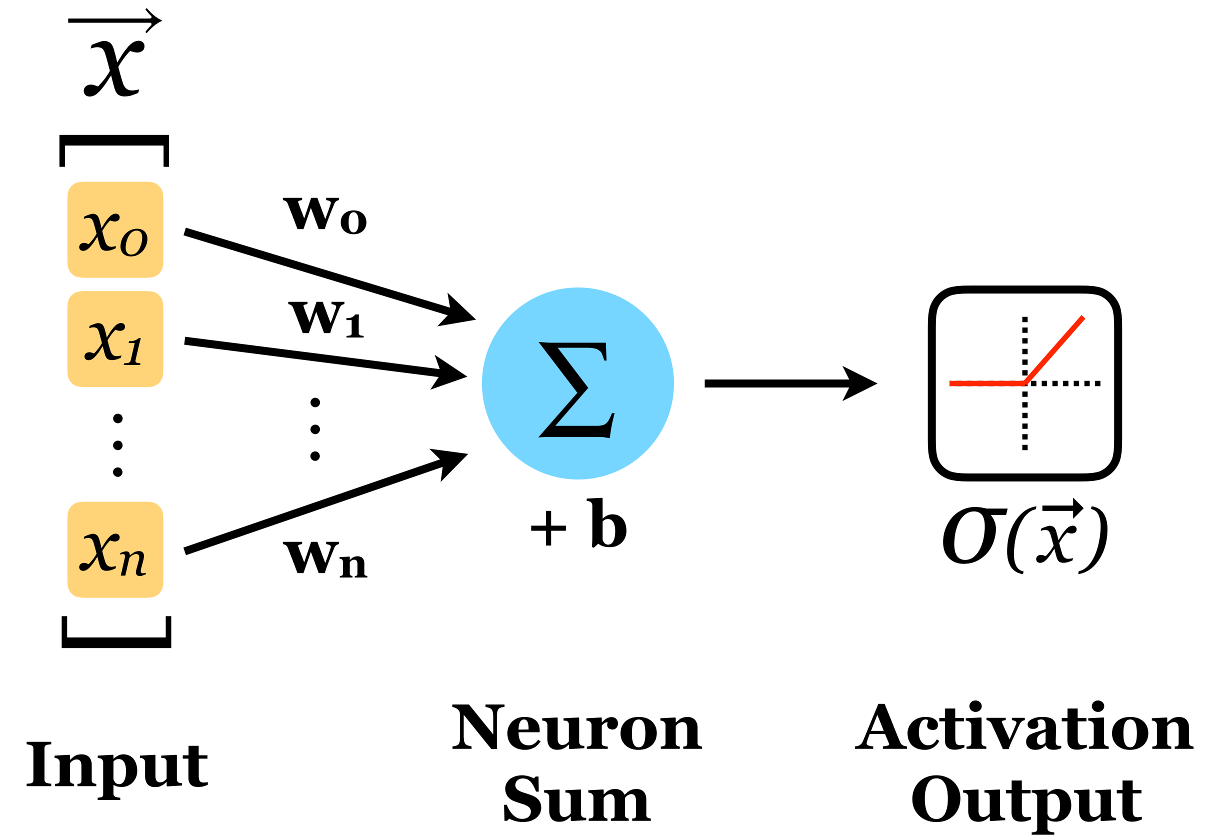
Natural Neural Network

# Machine Learning Overview

## 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.

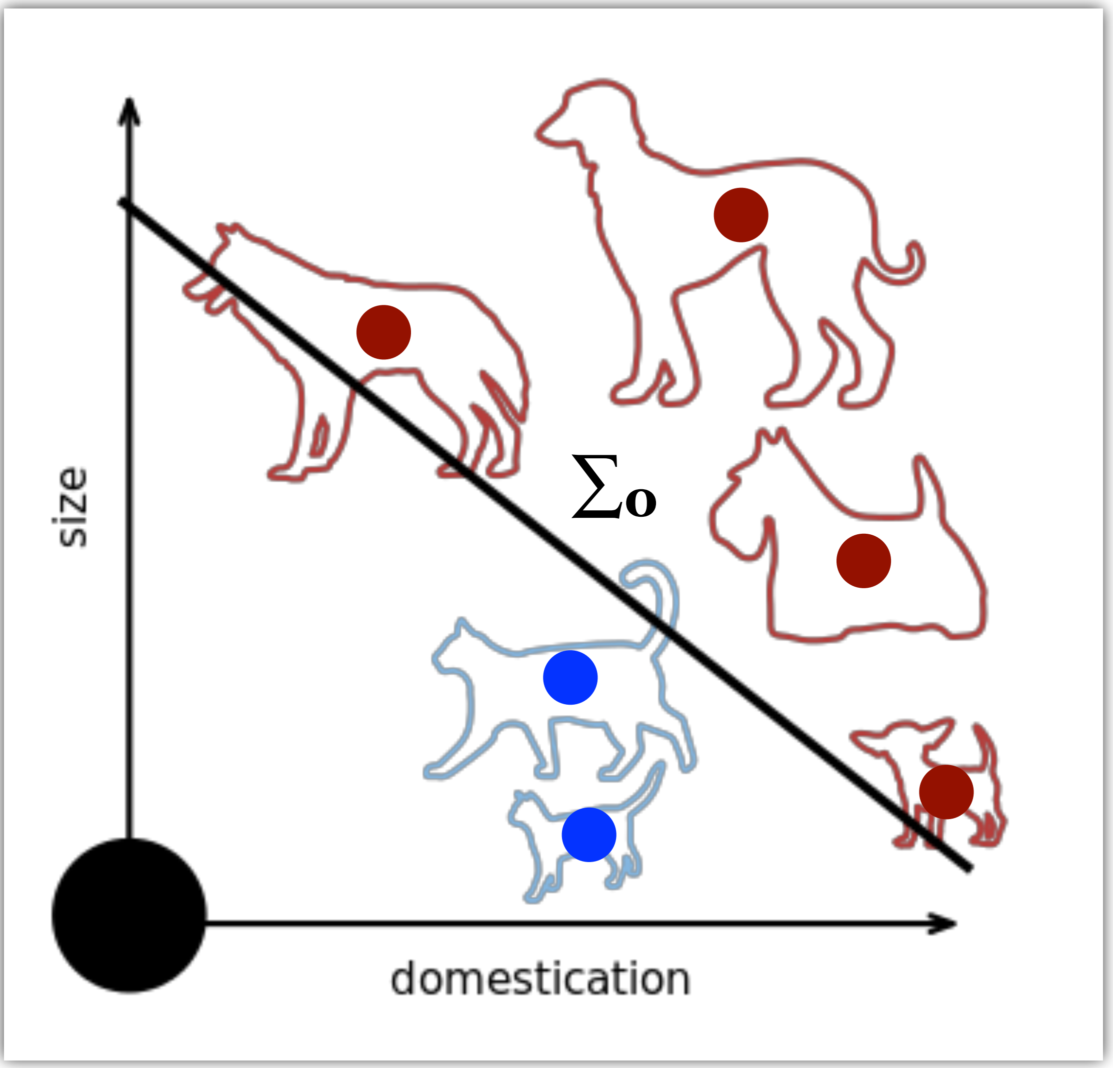



$$\sigma(\vec{x}) = \begin{cases} \vec{w}_i \cdot \vec{x} + b_i & \vec{w}_i \cdot \vec{x} + b_i \geq 0 \\ 0 & \vec{w}_i \cdot \vec{x} + b_i < 0. \end{cases}$$

# Simple neural network (perceptron)



Imagine using two features to separate cats and dogs




**from wikipedia**

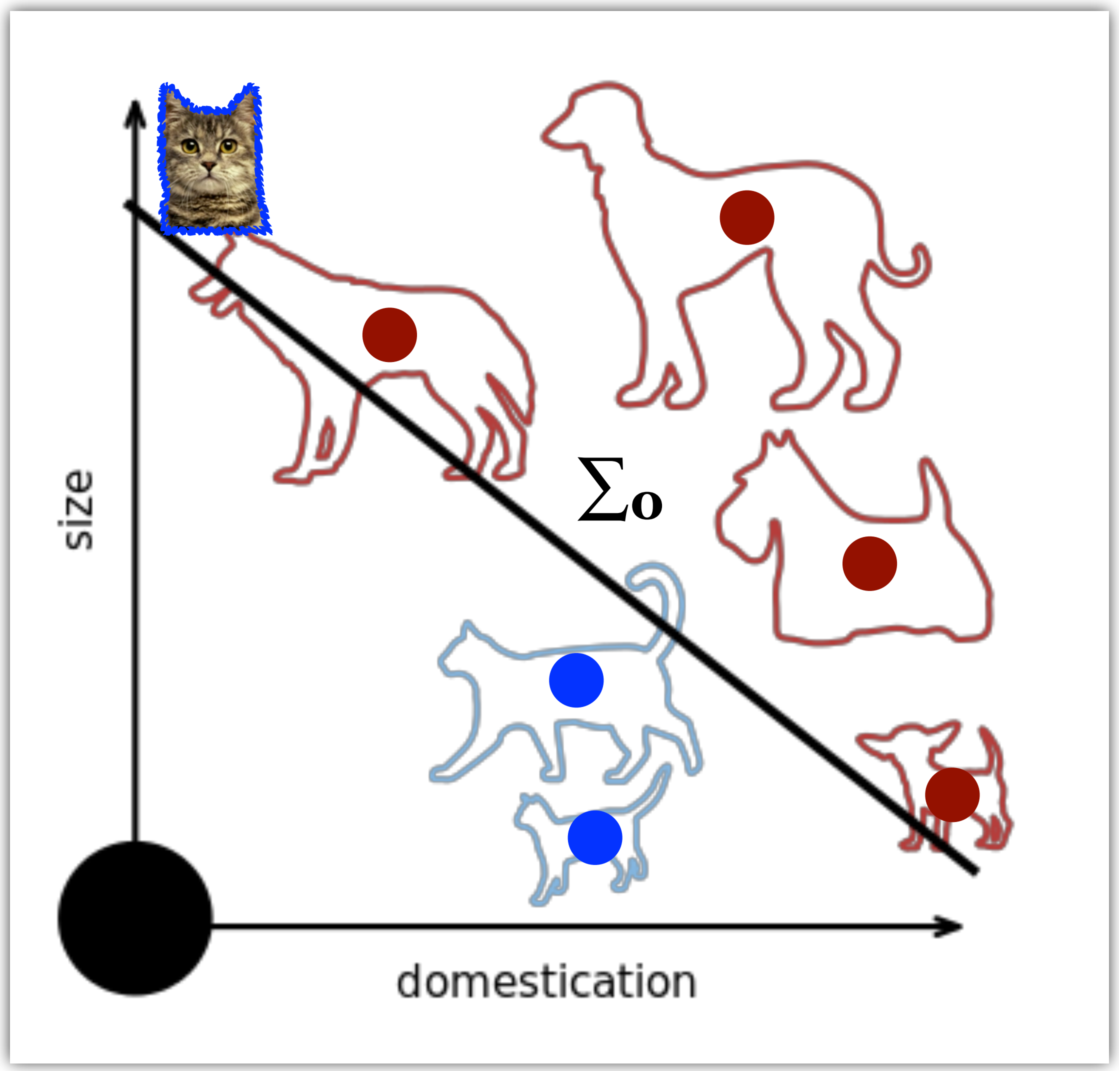
$$\sigma(\vec{x}) = \begin{cases} \vec{w}_i \cdot \vec{x} + b_i & \vec{w}_i \cdot \vec{x} + b_i \geq 0 \\ 0 & \vec{w}_i \cdot \vec{x} + b_i < 0. \end{cases}$$

$x_0$, $x_1$ → $\Sigma_o$ → *Output* $\begin{bmatrix} \text{cat} \\ \text{dog} \end{bmatrix}$

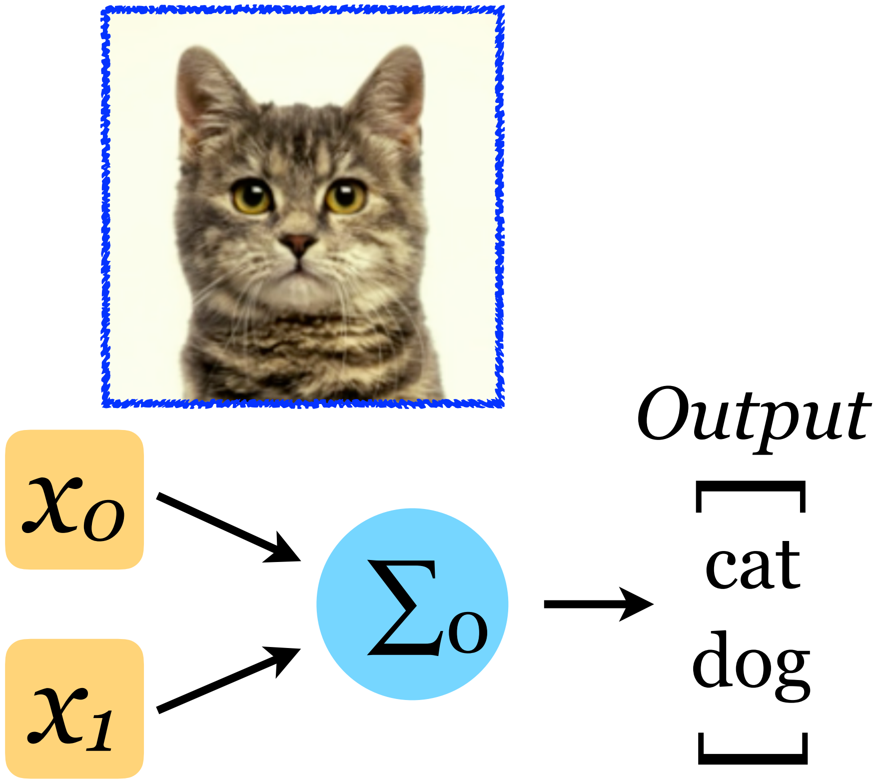
By picking a value for **w** and **b**, we define a boundary between the two sets of data

# Simple neural network (perceptron)

What if we have a new data point?

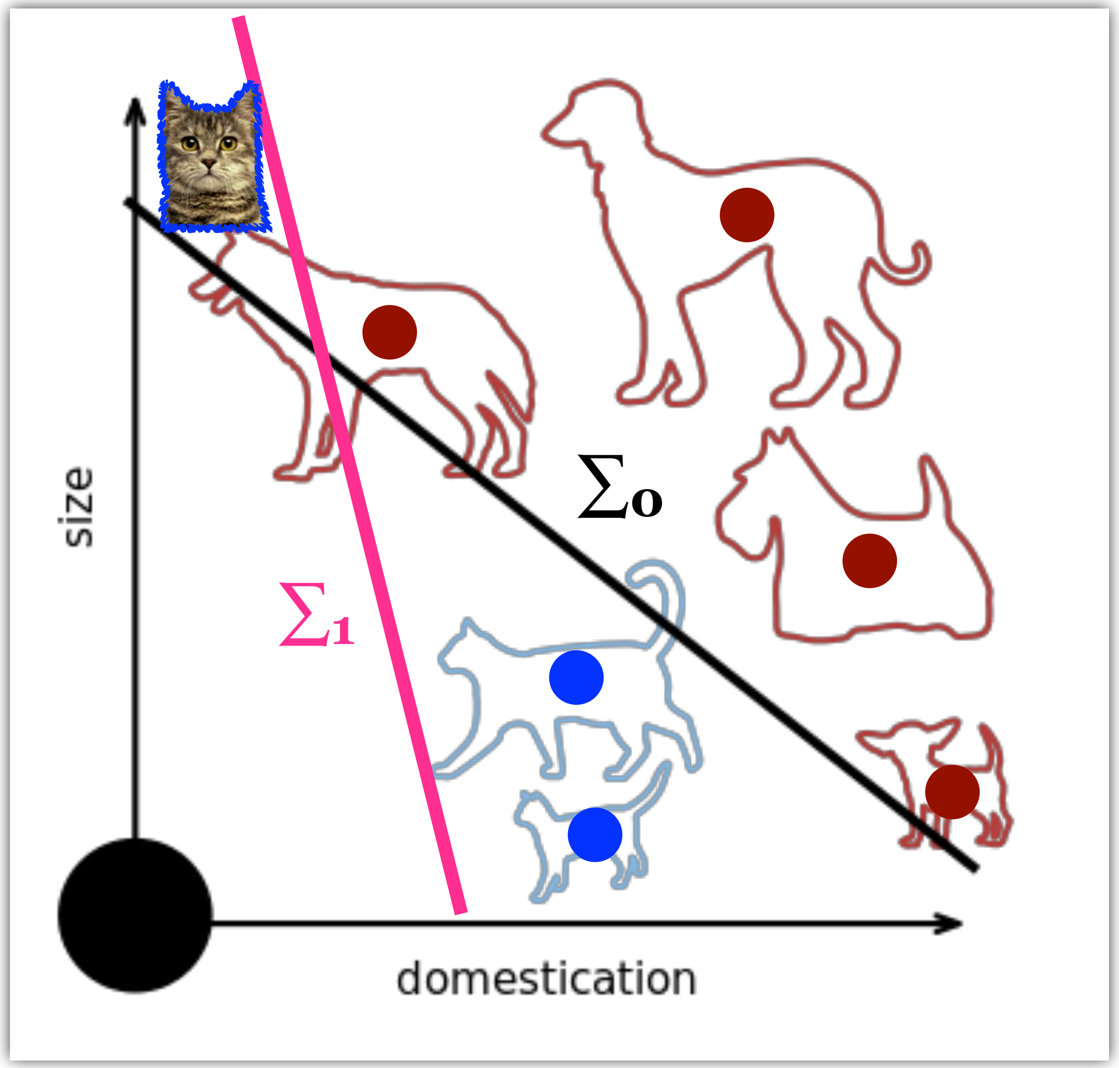



from [wikipedia]

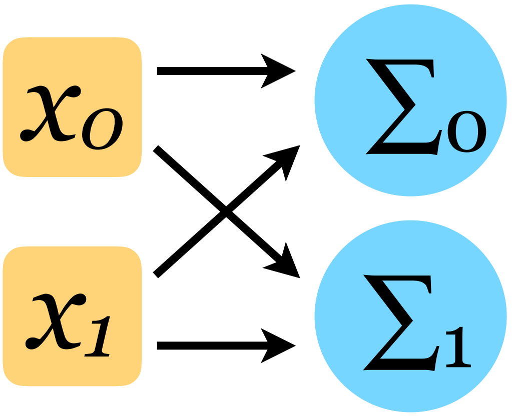


# Simple neural network (perceptron)

What if we have a new data point?



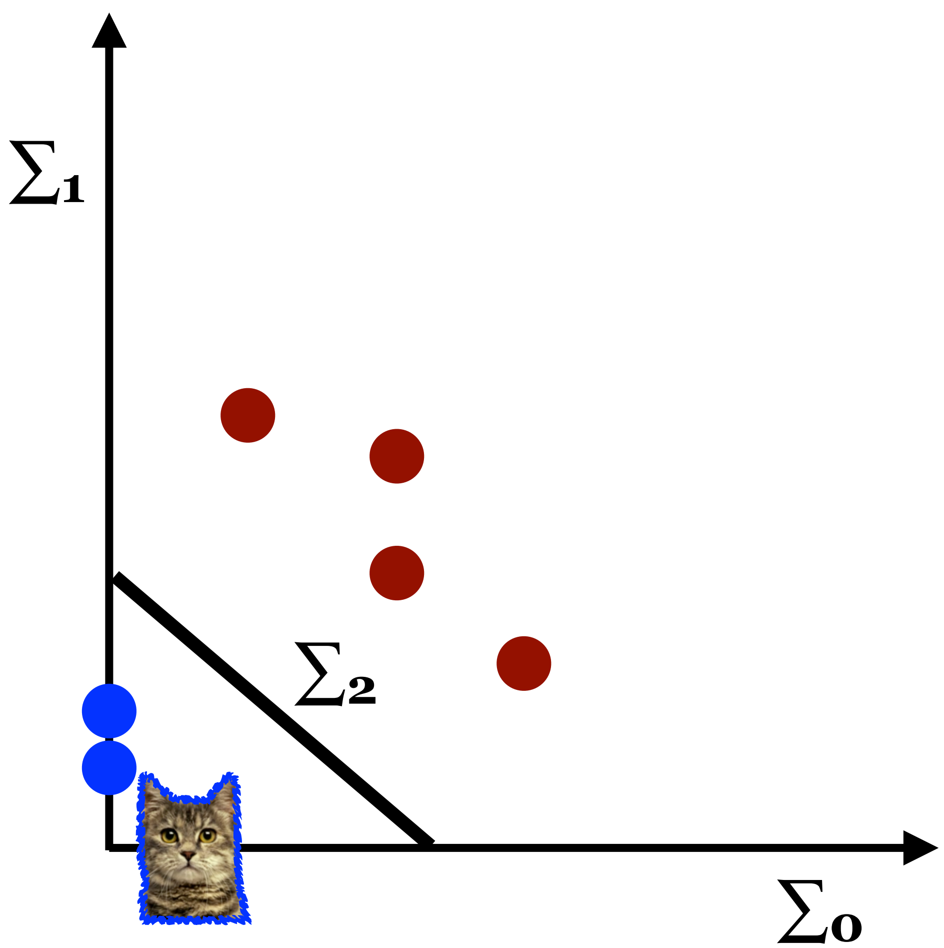

from [wikipedia](wikipedia)



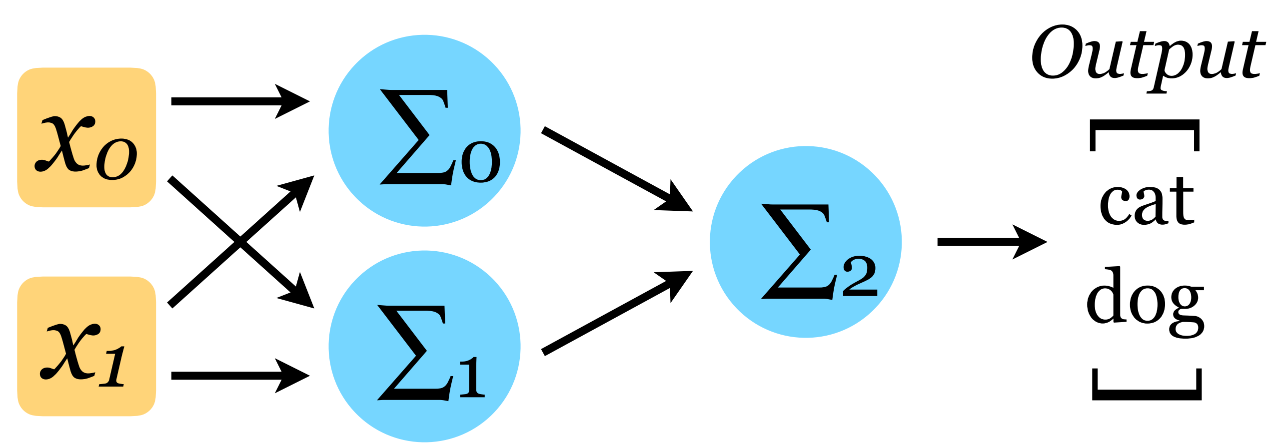

We can **add another perceptron** to help (but does not yet solve the problem)

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

What if we have a new data point?



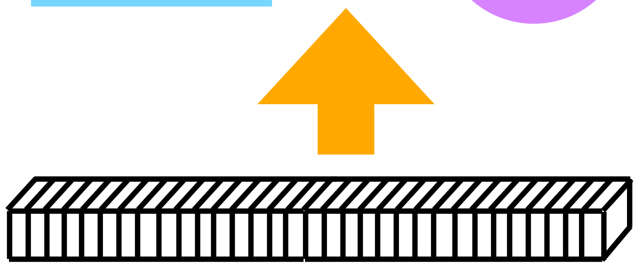

from [wikipedia](wikipedia)




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

# 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

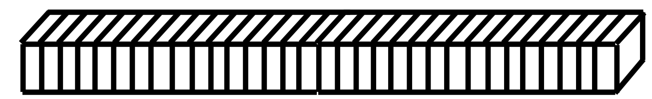
**How can we extract "features" from "image"?**

**Convolutional Neural Network**

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

SLAC

**Goal:** Dog or Cat



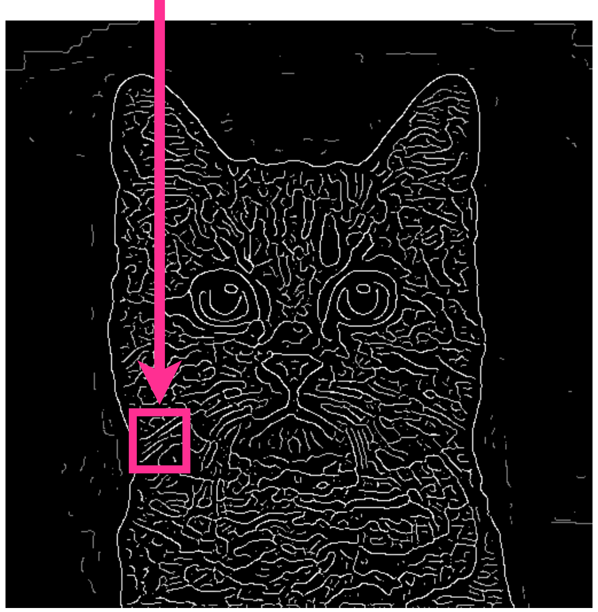
1D array of discriminants

convolutional filter **(kernel)**

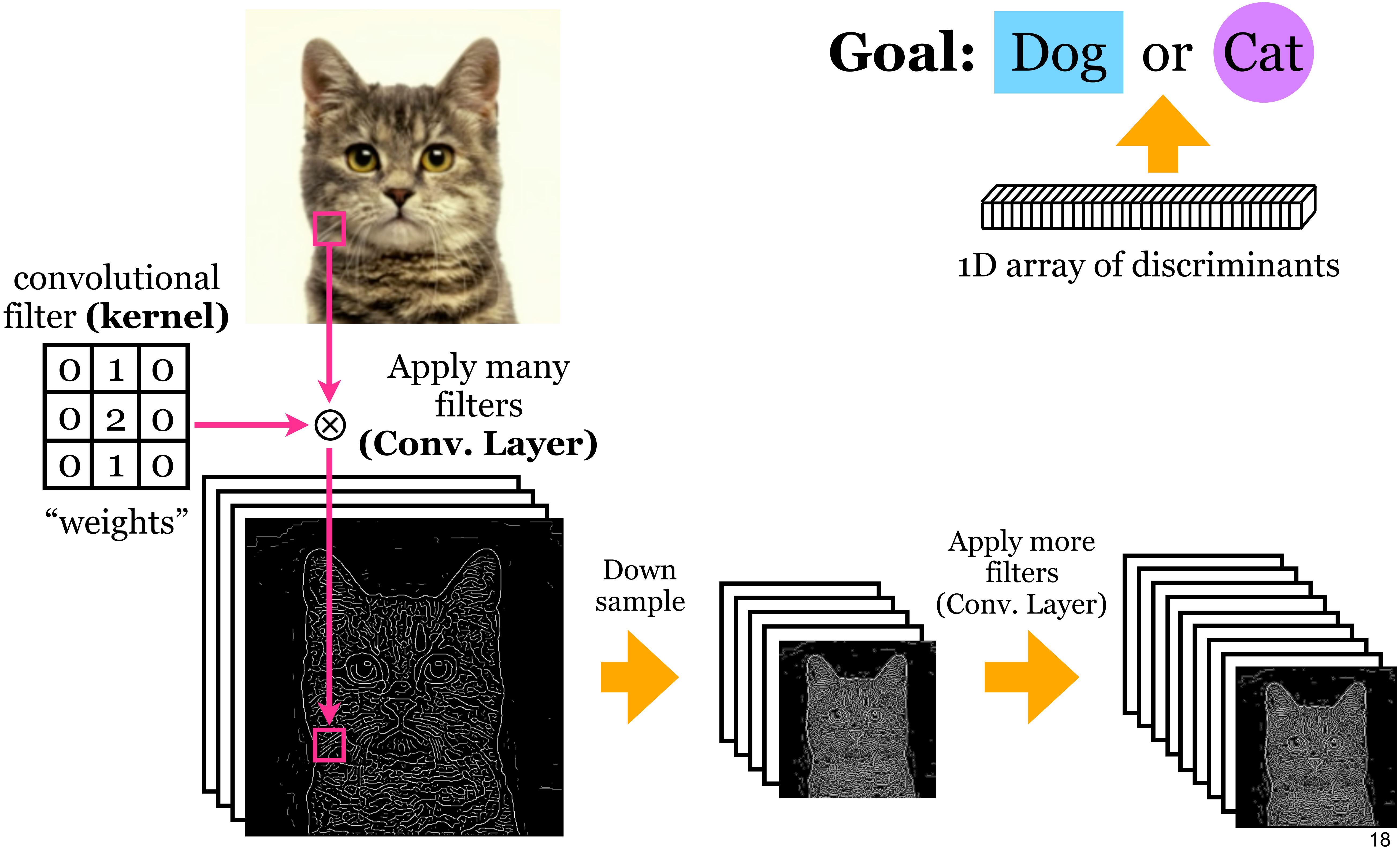
| | | |
|---|---|---|
| 0 | 1 | 0 |
| 0 | 2 | 0 |
| 0 | 1 | 0 |

"weights"

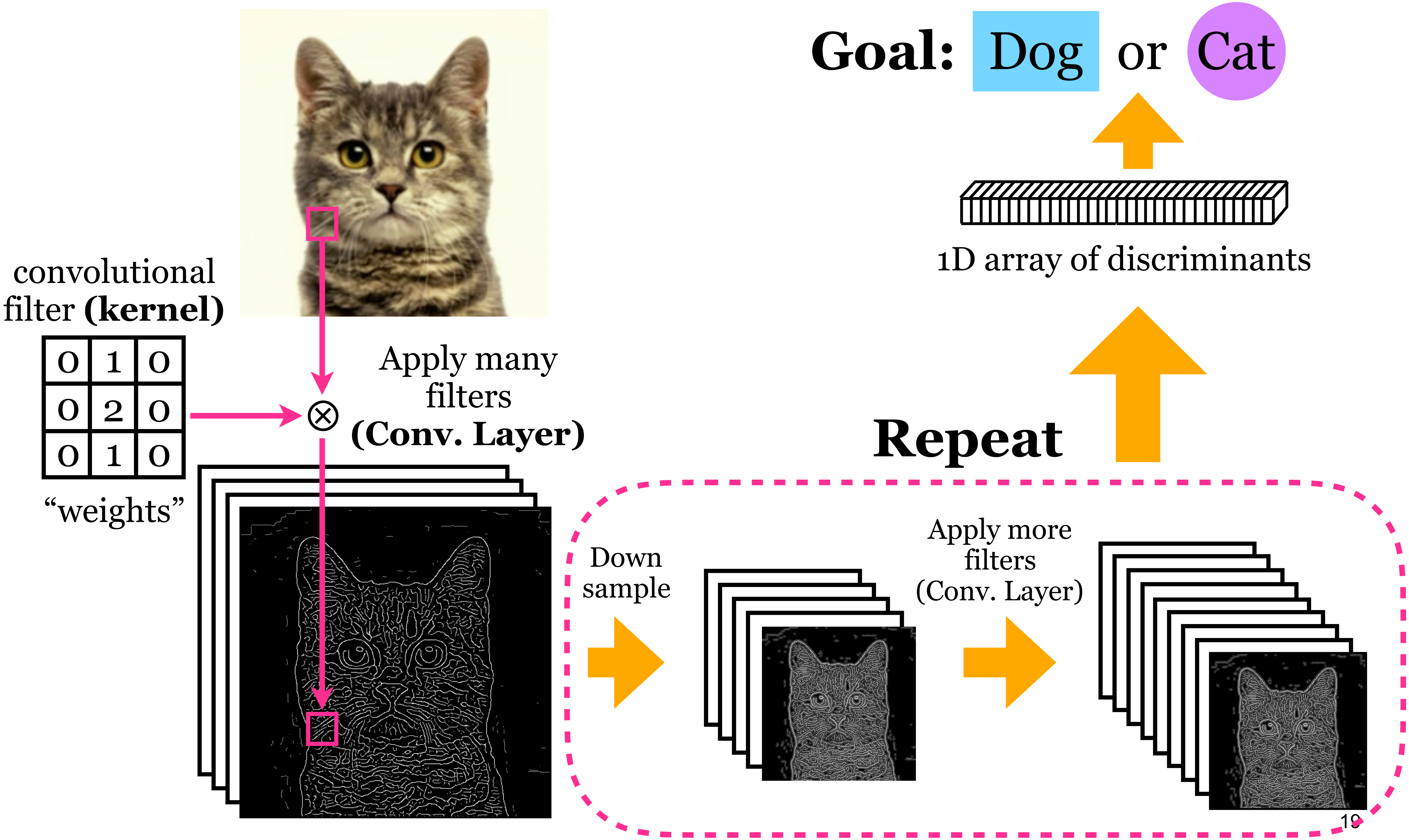
$\otimes$ **"neuron sum"**



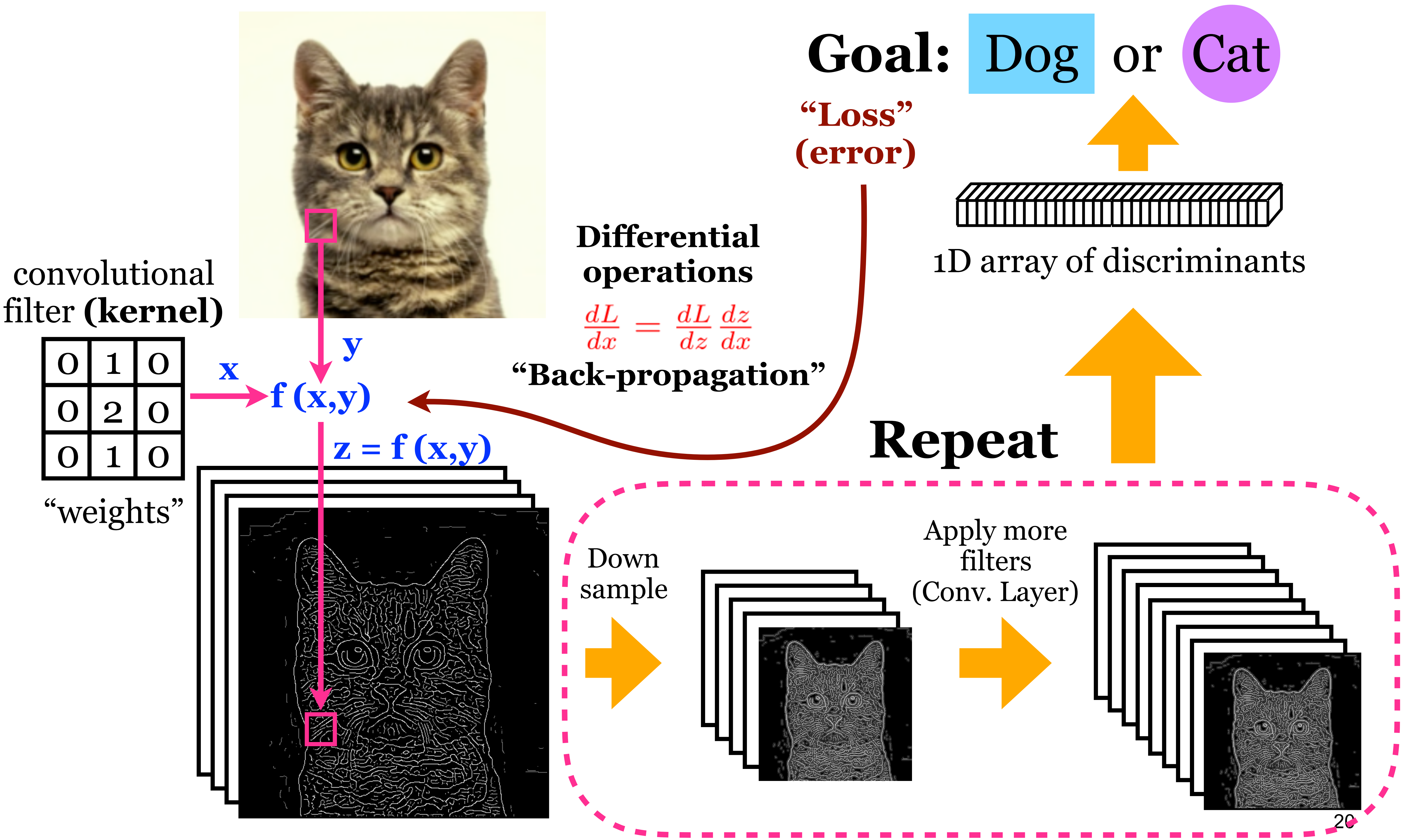
# Machine Learning Overview
## Convolutional Neural Network (CNN)



# Convolutional Neural Network (CNN)

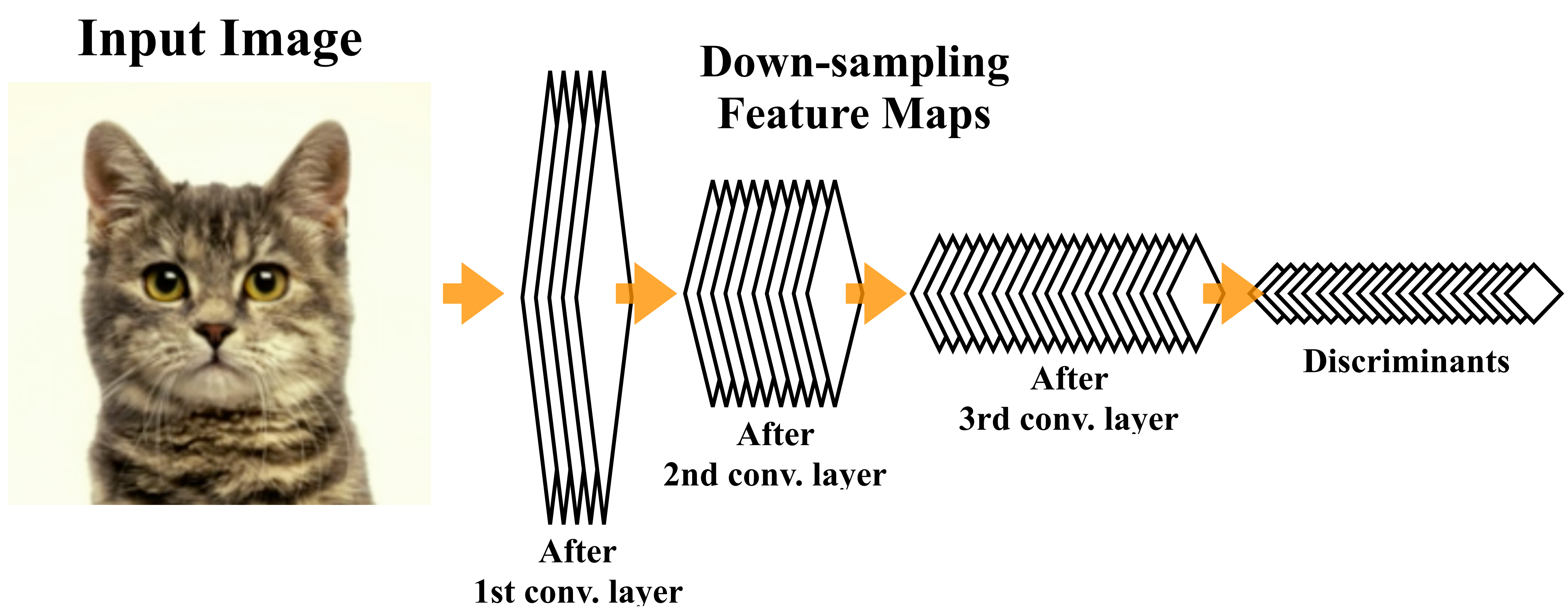


# Machine Learning Overview
## Supervised Training of CNN



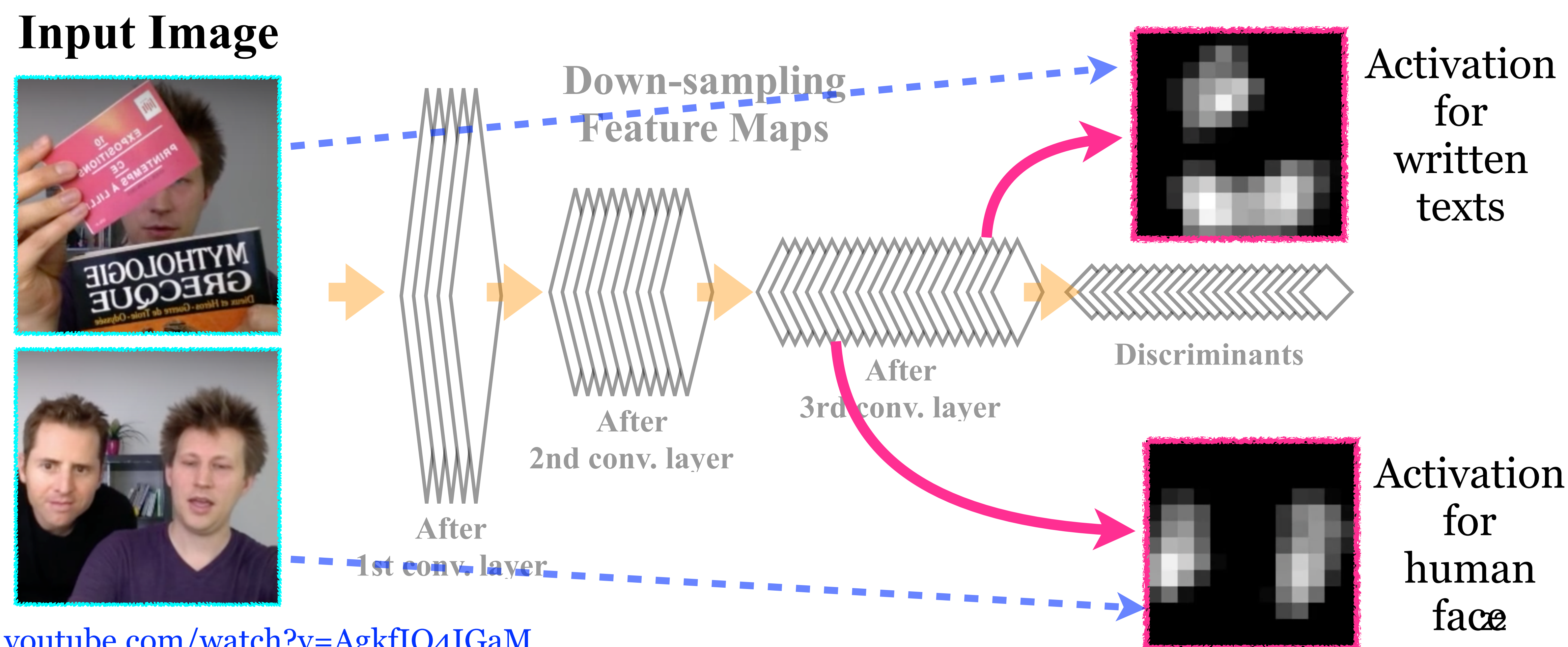
# Summarizing CNNs

- **CNNs are "feature extraction machine"**
  - Consists of a "convolution layer" with "kernels"
  - A chain of linear algebra operations = "massively parallel"
    - Suited for acceleration using many-core hardwares (e.g. GPUs)
- CNNs seen as **a geometrical data transformer**



# Summarizing CNNs

- **CNNs are “feature extraction machine”**
  - Consists of a “convolution layer” with “kernels”
  - A chain of linear algebra operations = “massively parallel”
    - Suited for acceleration using many-core hardwares (e.g. GPUs)
- CNNs seen as **a geometrical data transformer**




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

## Revolution with Deep Neural Networks

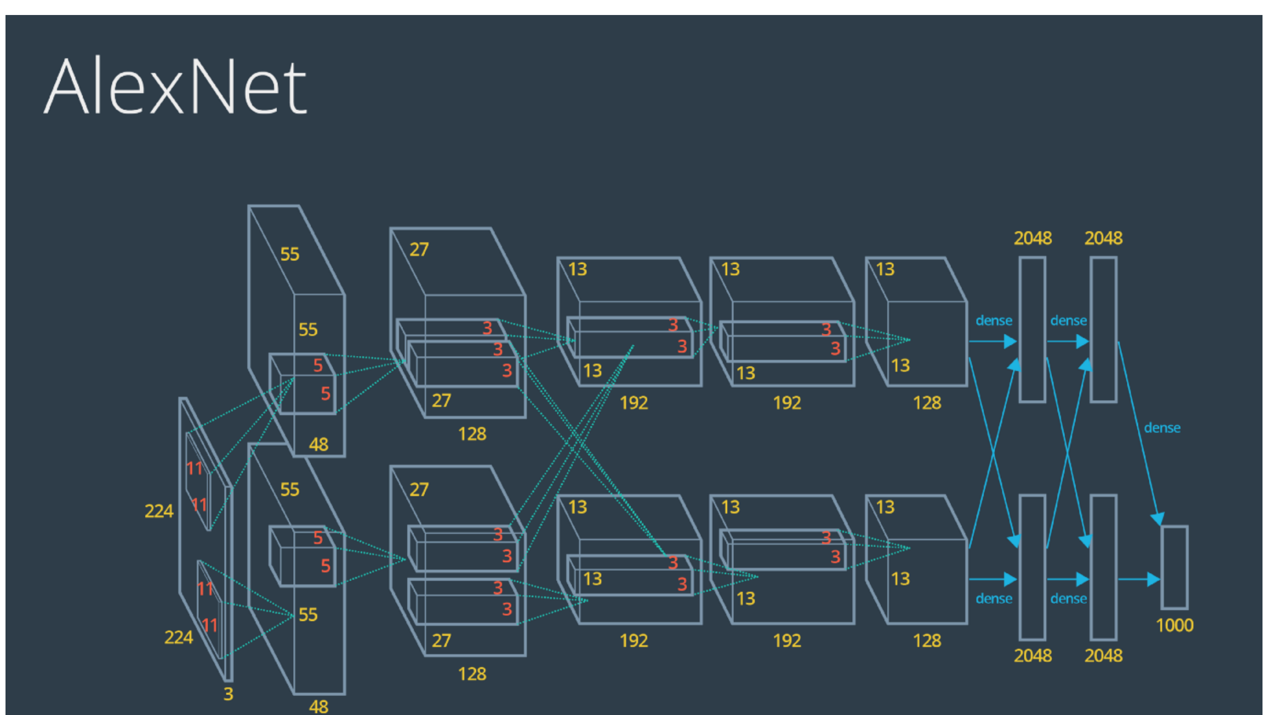
SLAC

**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**

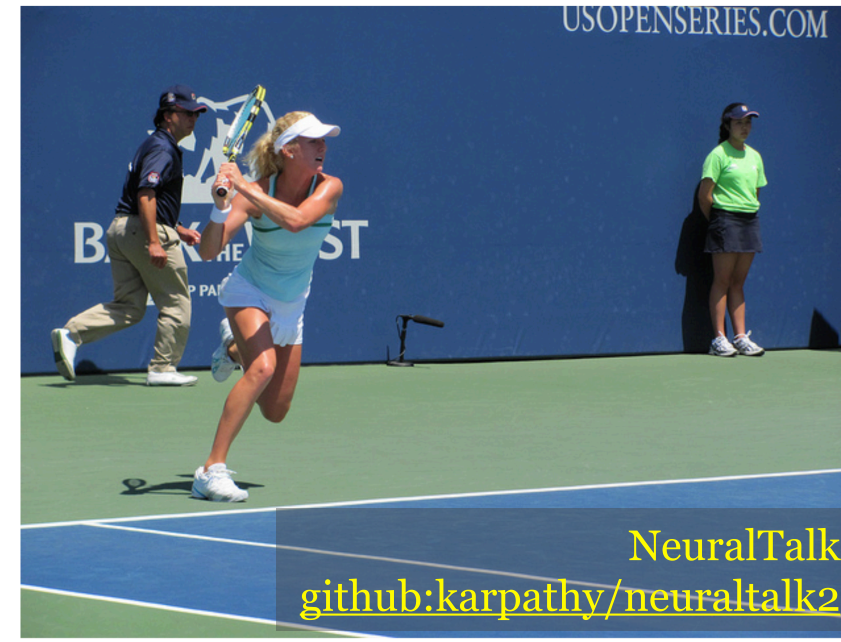

**Abstract**

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.

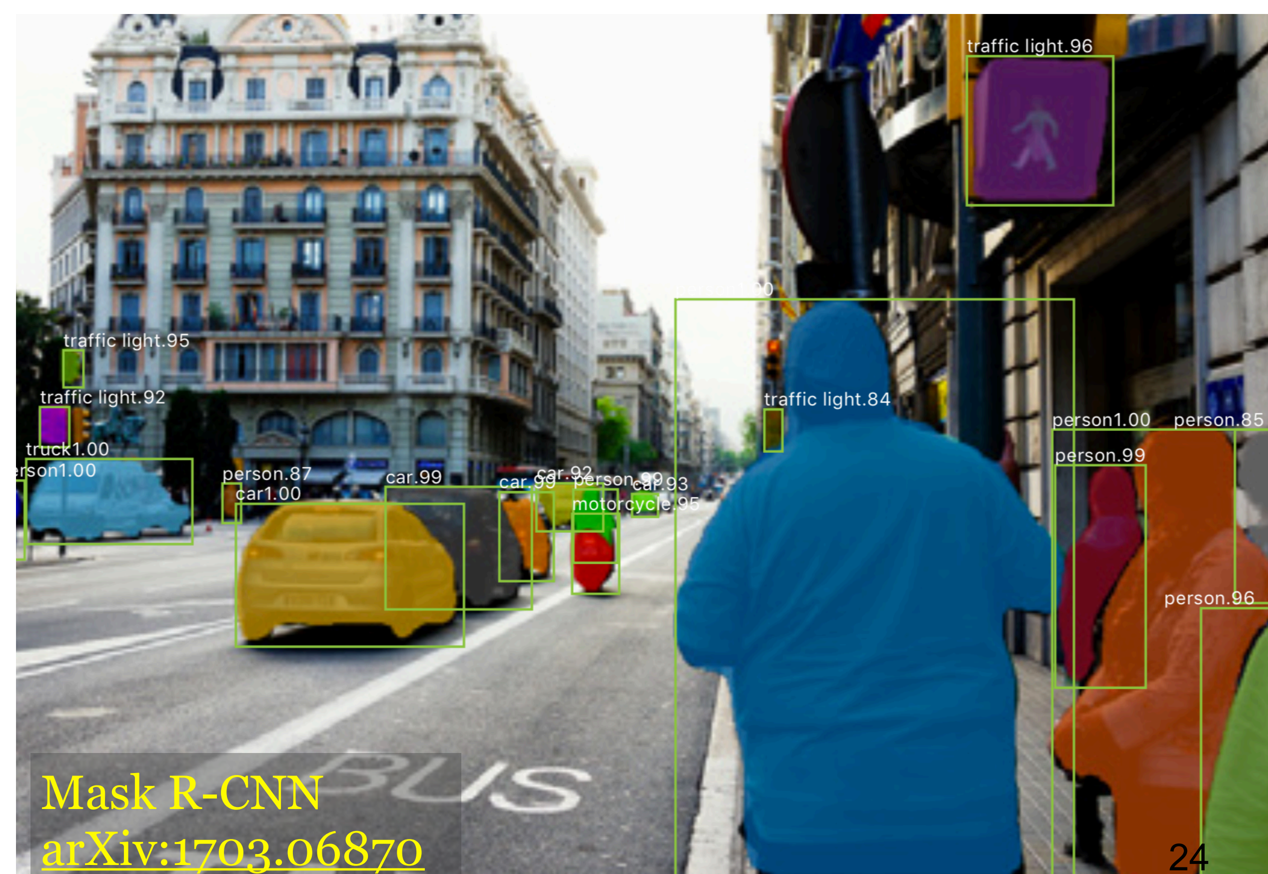

# Machine Learning Overview
# Revolution with Deep Neural Networks

SLAC


"girl in pink dress is jumping in air."




a woman is playing tennis on a tennis court

# Machine Learning in ~~Computer Vision~~

## High-Precision Detector Data Analysis

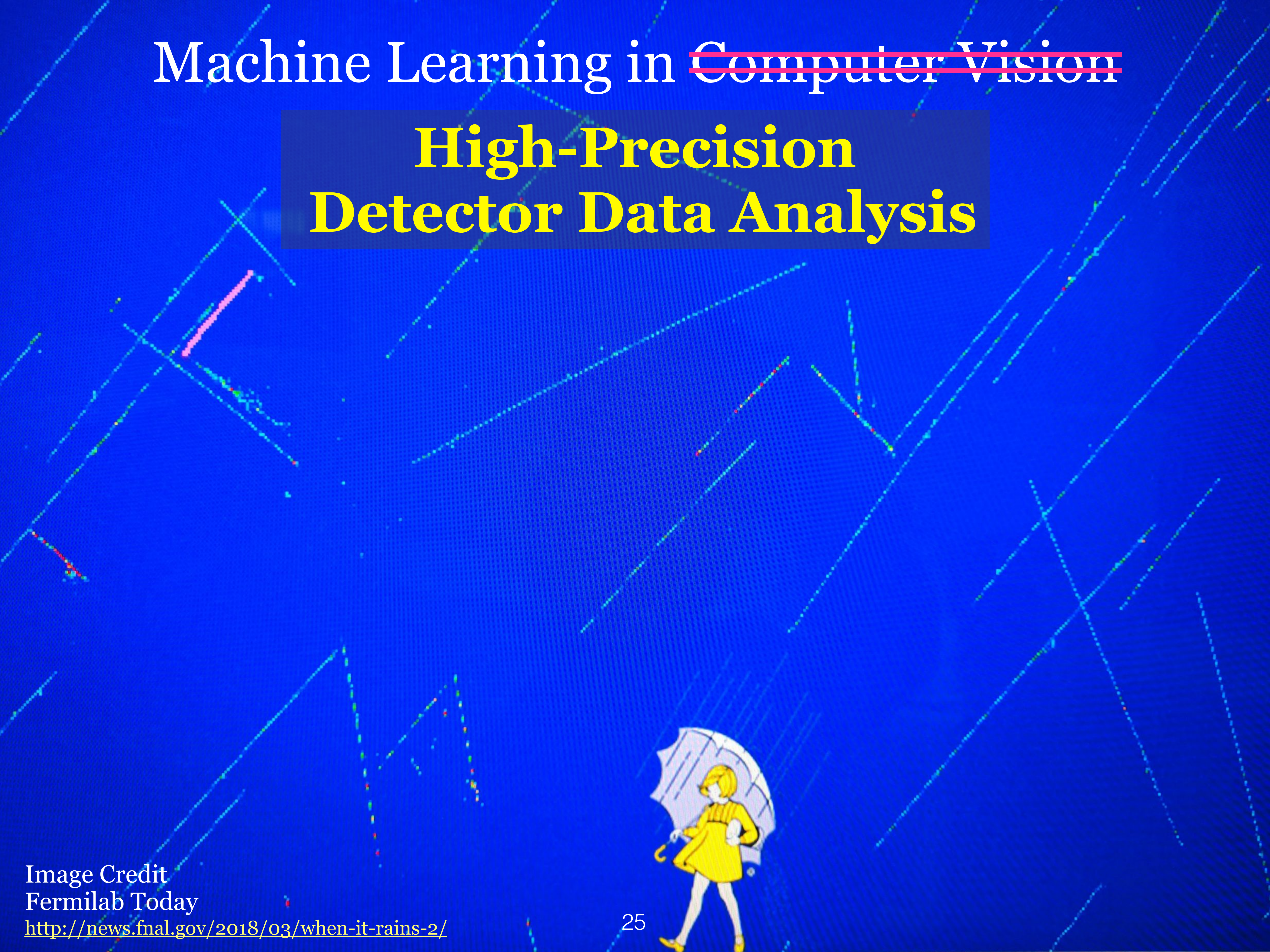
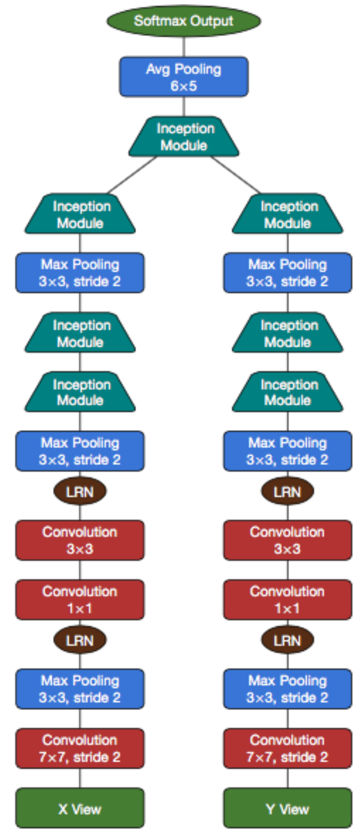


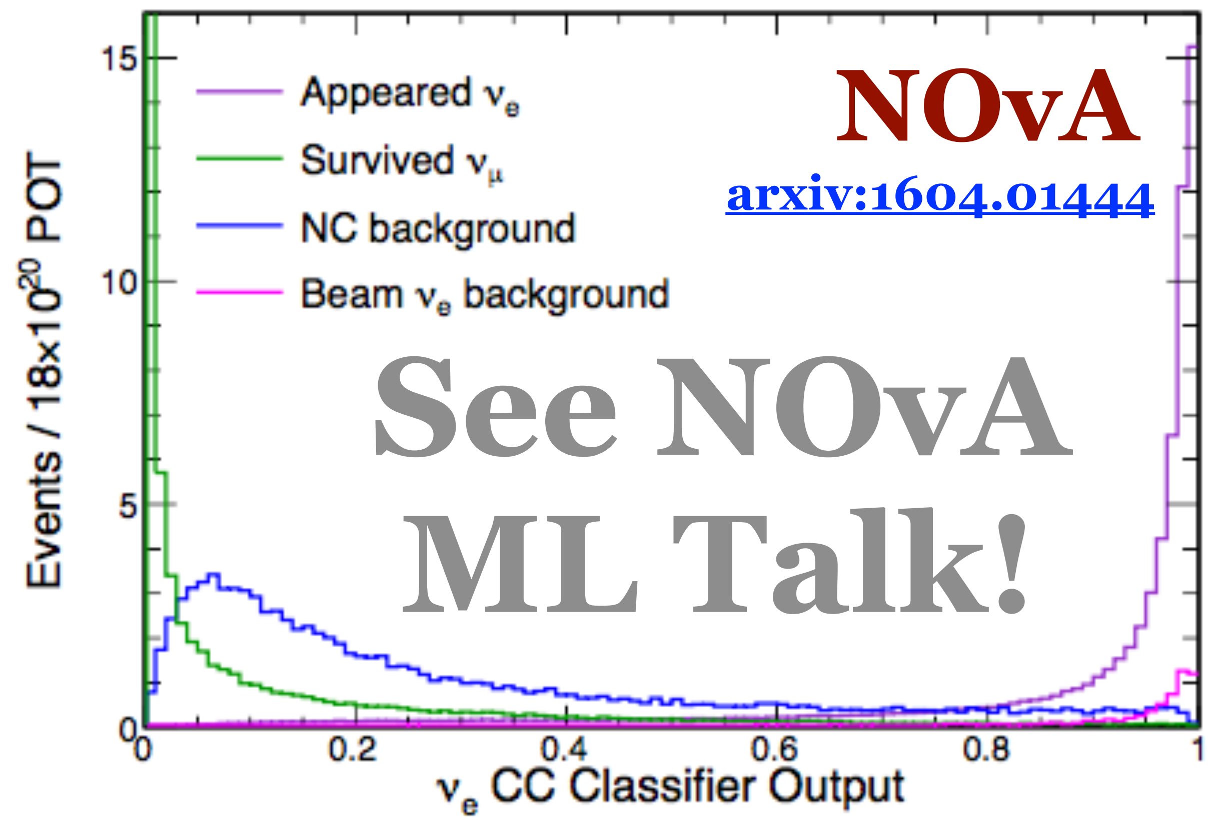
Image Credit
Fermilab Today
http://news.fnal.gov/2018/03/when-it-rains-2/

# Image Classification Application

## NOvA Neutrino Event Classifier

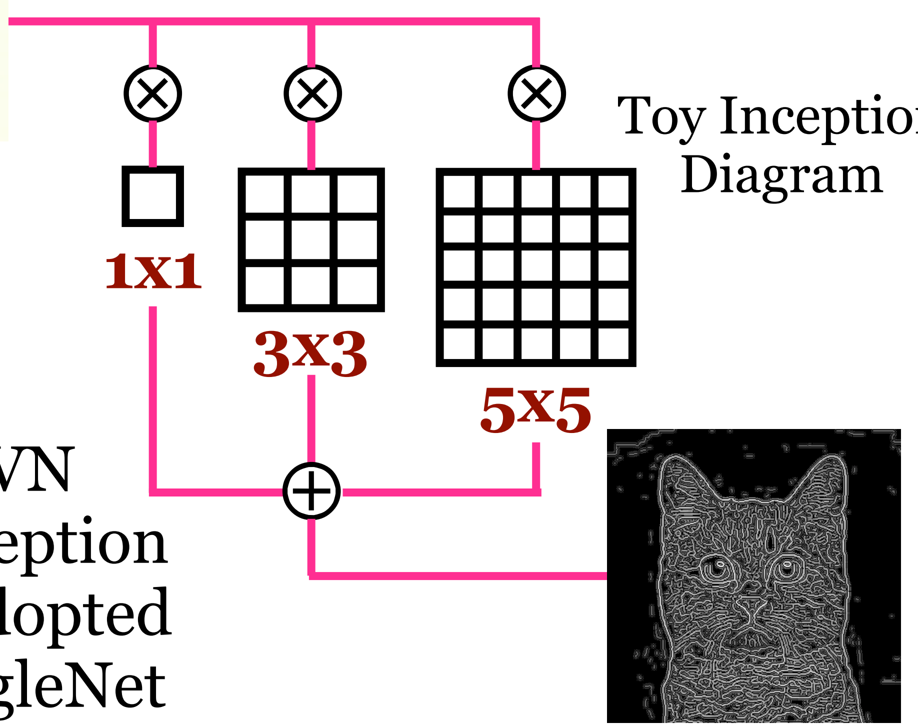
Neutrino event topology classification with 2D images



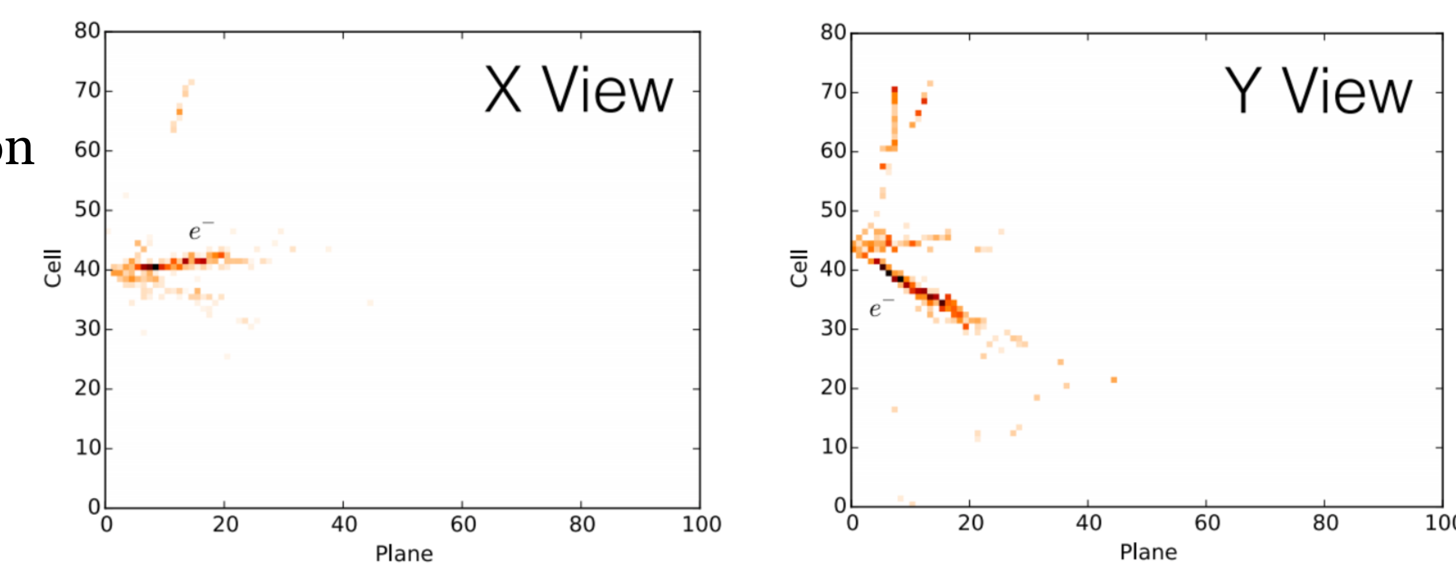

## "Inception Module"

A convolution with multiple sized kernels




Toy Inception Diagram

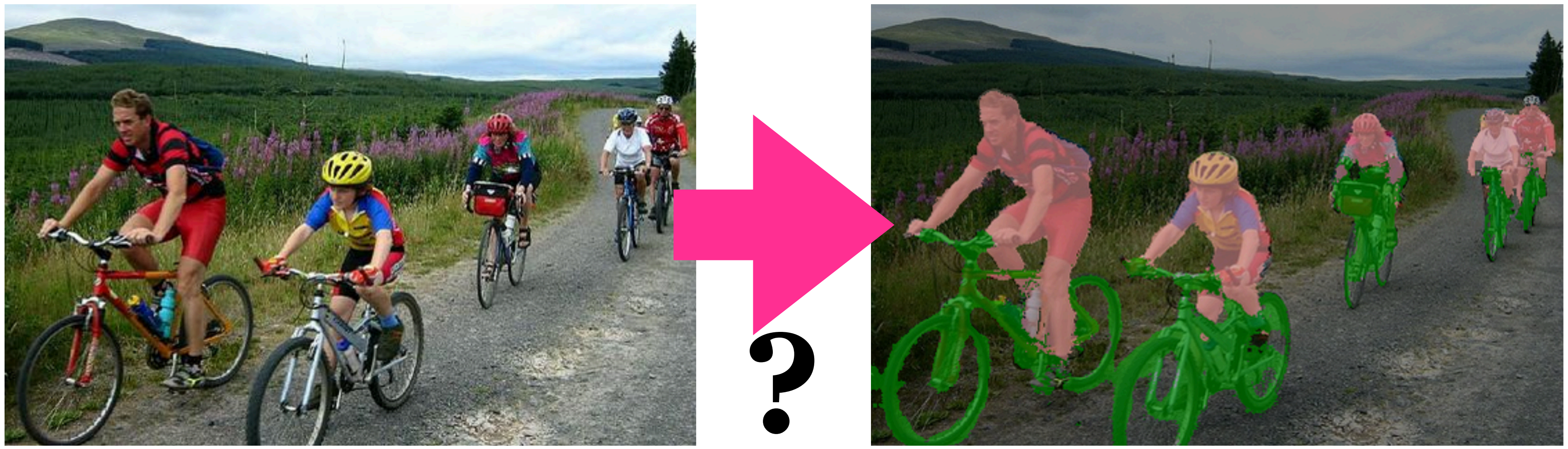
NOvA's CVN utilize inception module adopted from GoogleNet

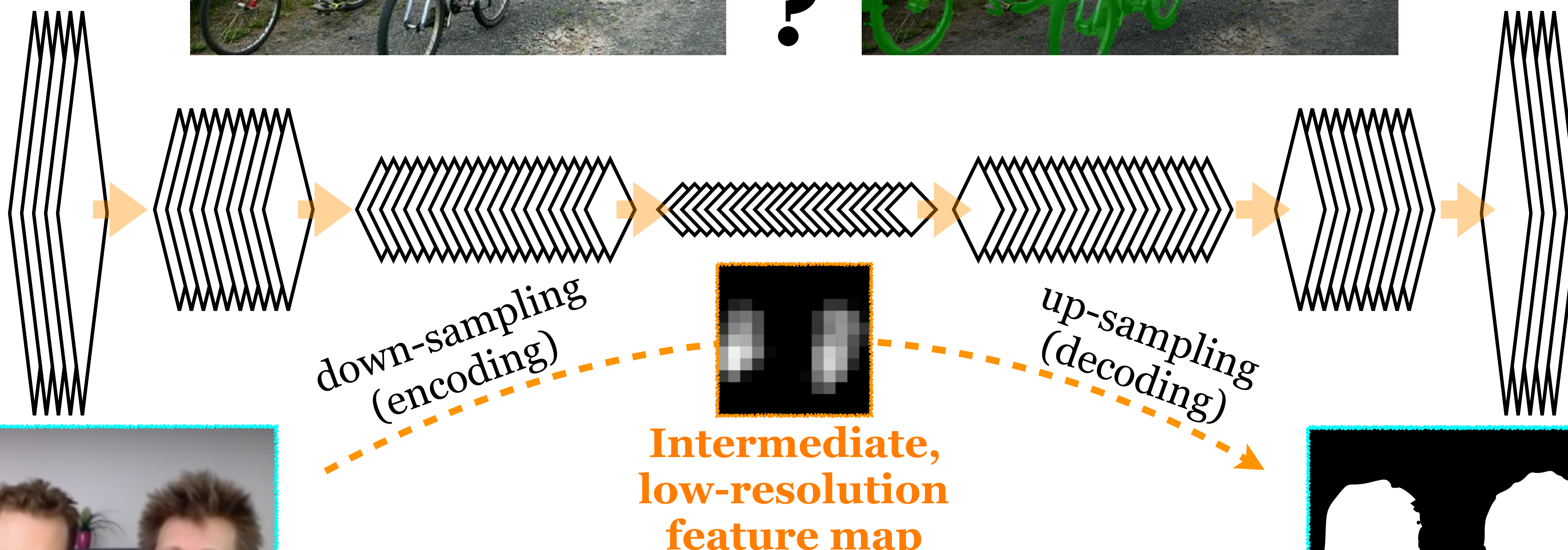


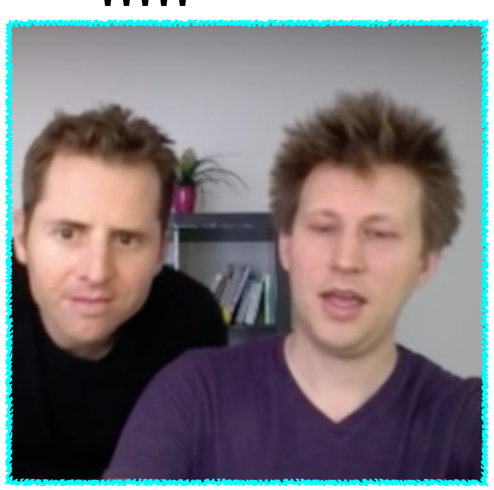

Input "images"

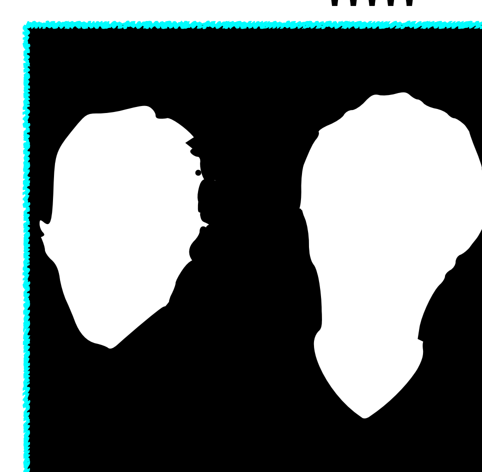
# Physics Applications
## Beyond image classification: pixel segmentation

# Beyond image classification: pixel segmentation



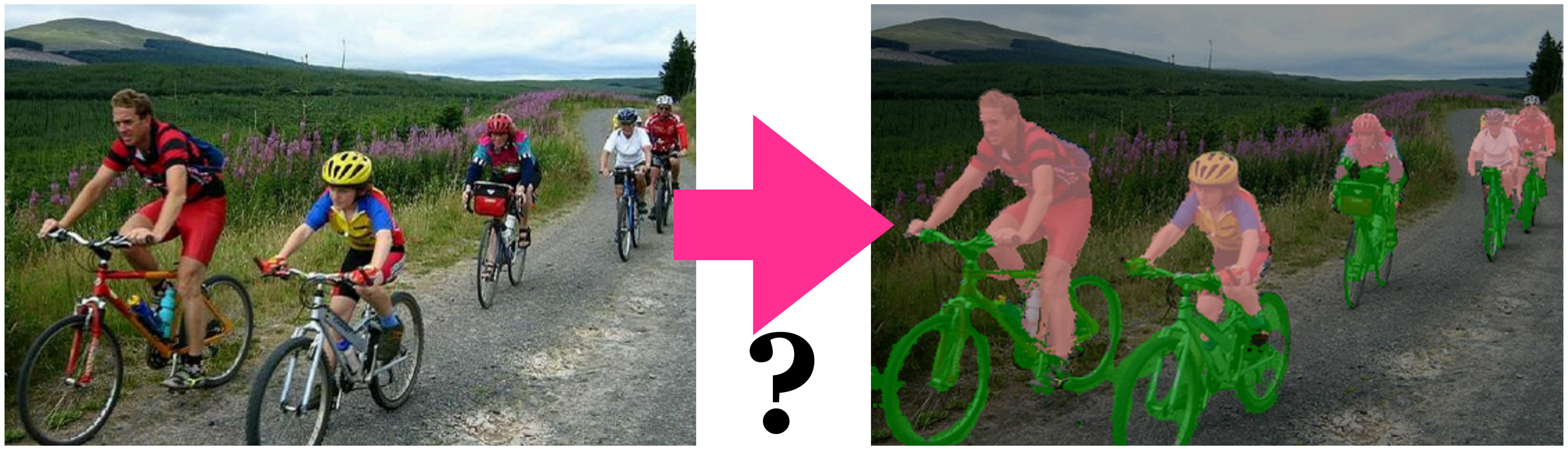





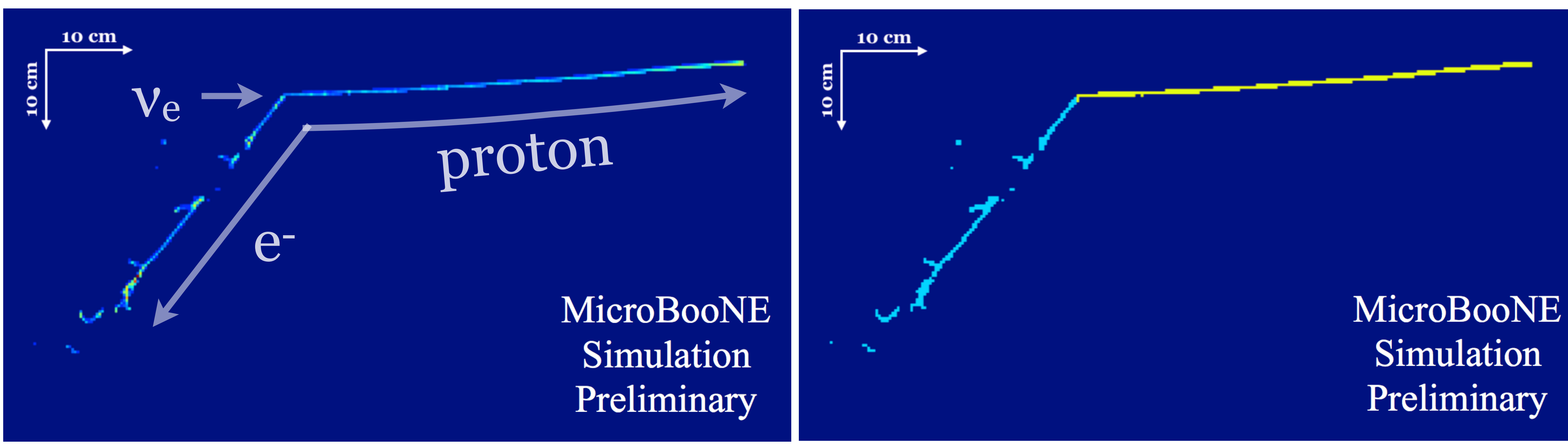
- Combine “up-sampling” + convolutions
- Outcome: “learnable” interpolation filters

# Physics Applications
## Beyond image classification: pixel segmentation




**See MicroBooNE ML Talk!**

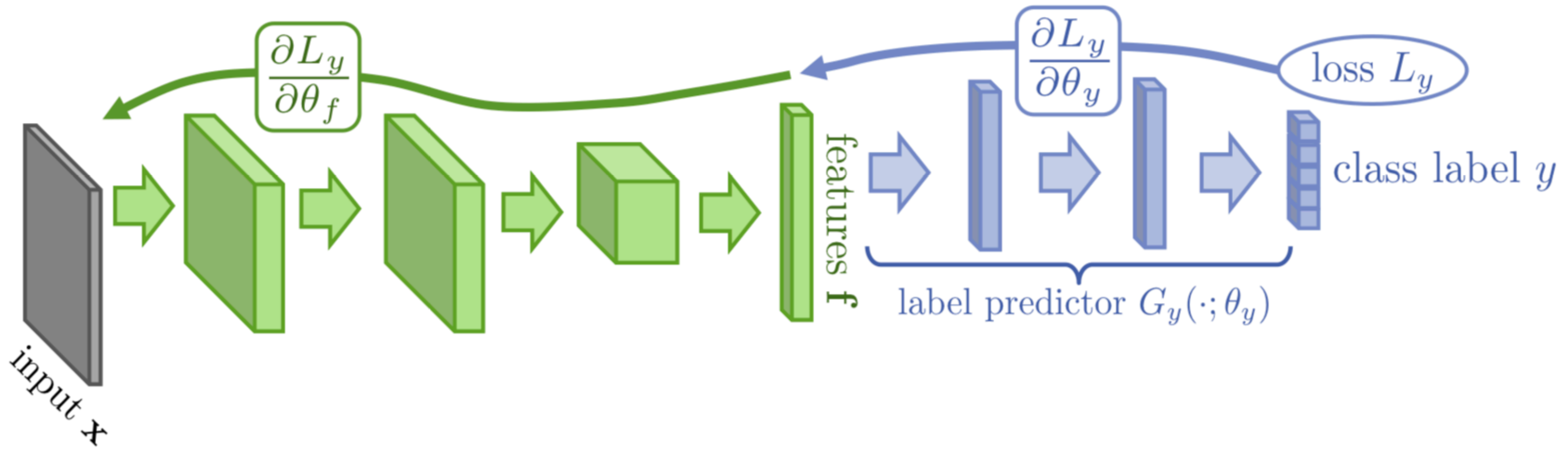
**MicroBooNE Paper**
arXiv:1808.07269



# Drawbacks of supervised training

## 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

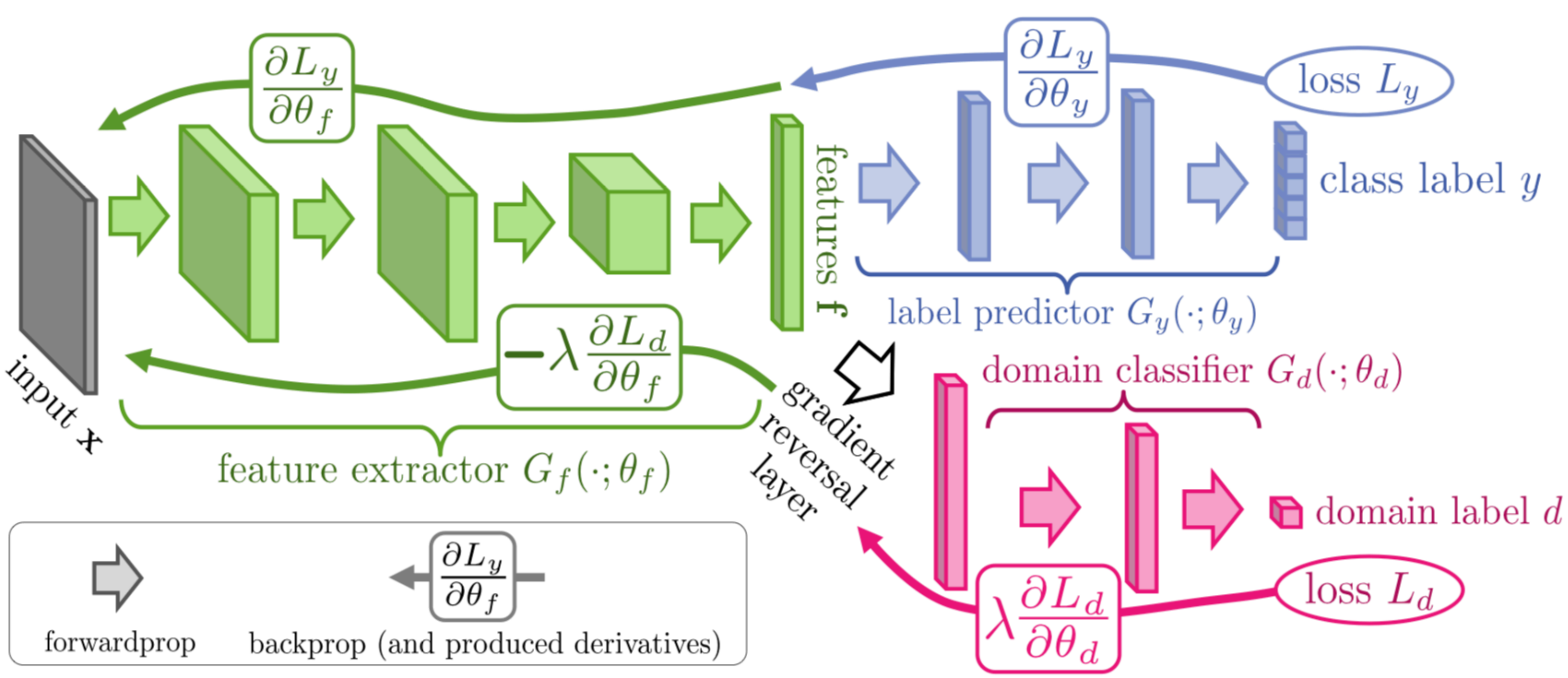


# Drawbacks of supervised training

## 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

**See Minerva ML Talk!**



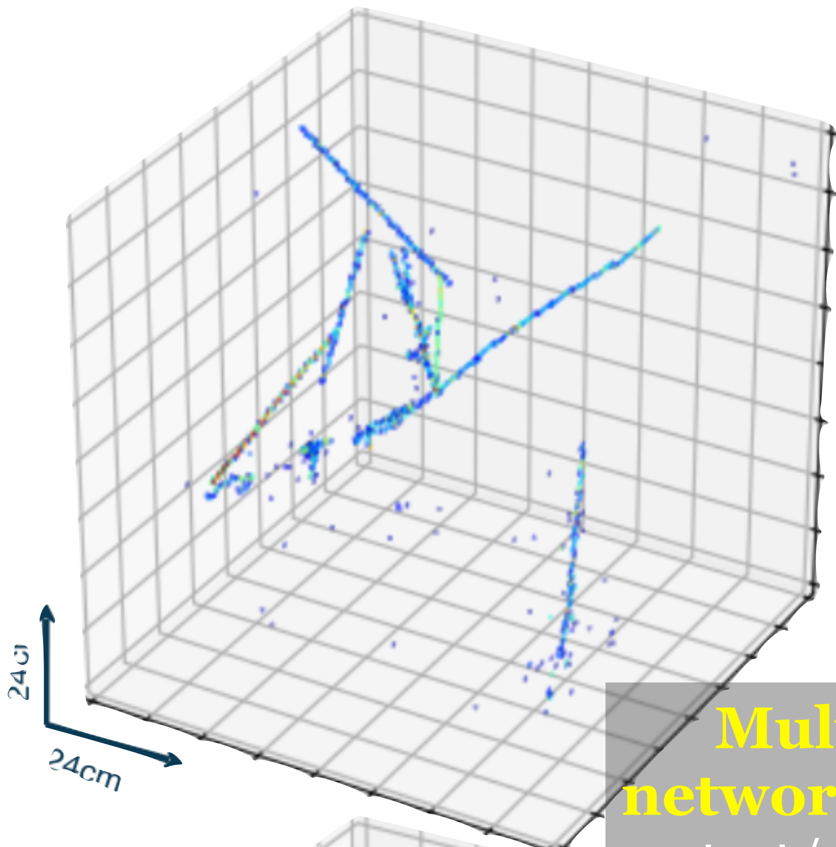

Maximize the loss for discriminate data vs. simulation, feature extractors are penalized to key on simulation specific information

**Minerva Paper** arXiv:1808.08332

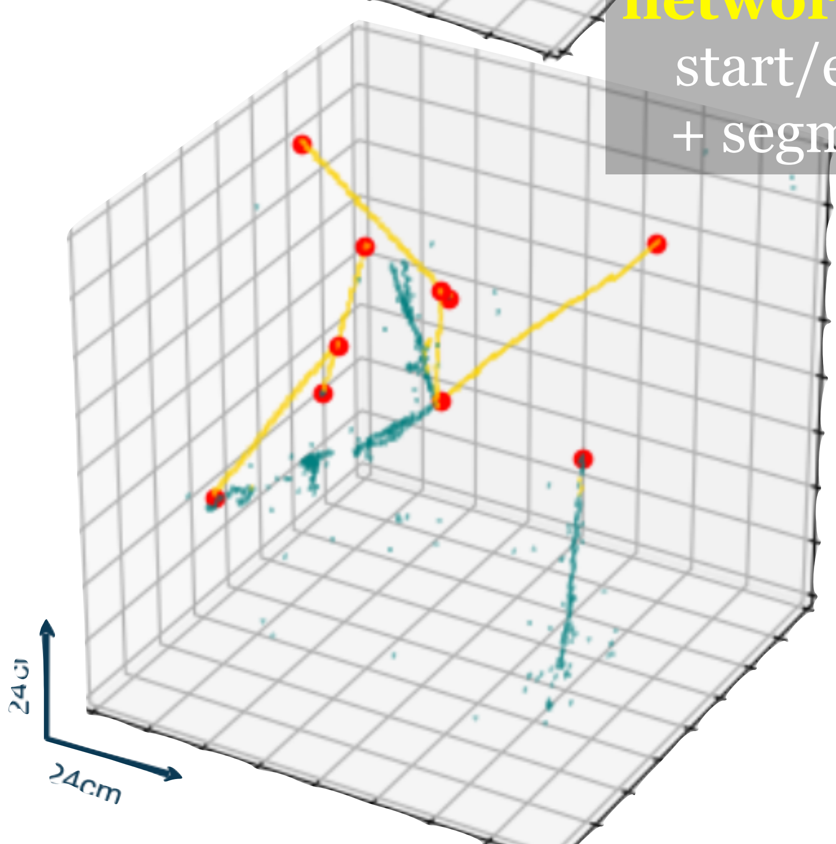
Domain-Adversarial Training of Neural Networks
J. Mach. Learn. Res. 17 (2016)

# Physics Applications
## DNNs for full reconstruction on "Big Data"

SLAC

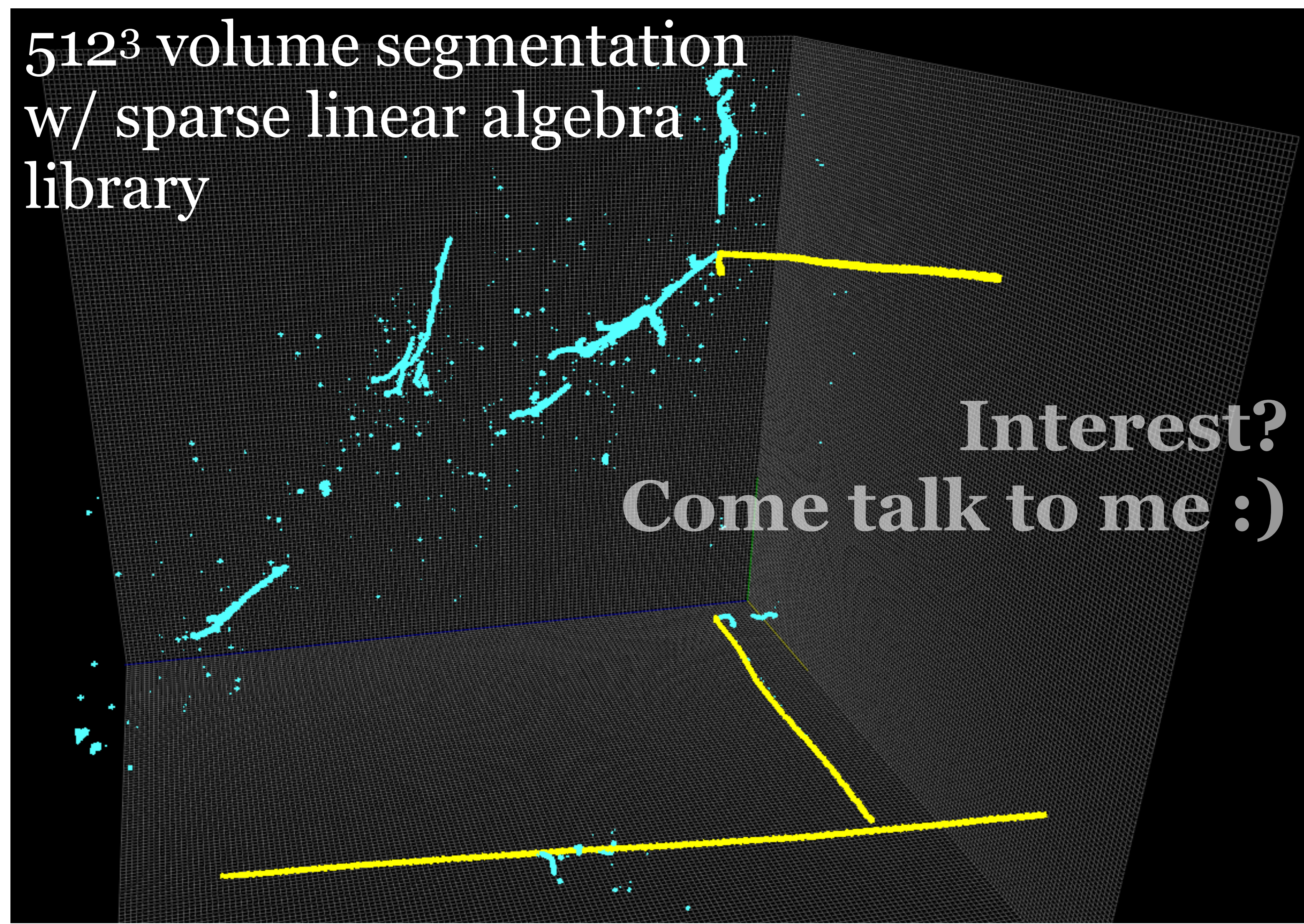



**Multi-task network cascade**
start/end point
+ segmentation

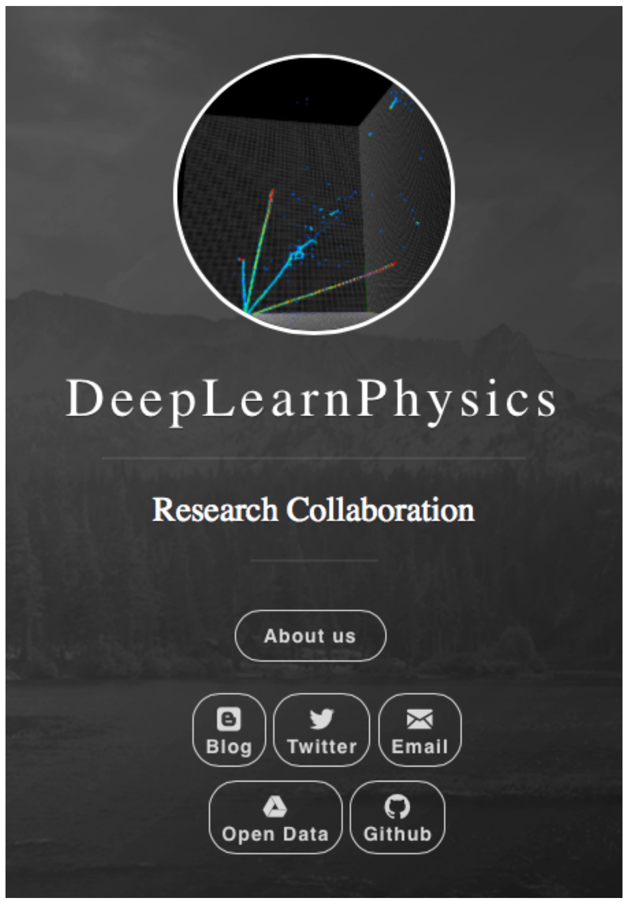



### Full reconstruction (multi-task network)
- 3D particle clustering + type ID
- Detection of vertex, particle "start/end"

### Sparse Big Data
- Locally dense, but overall <0.1% occupancy!



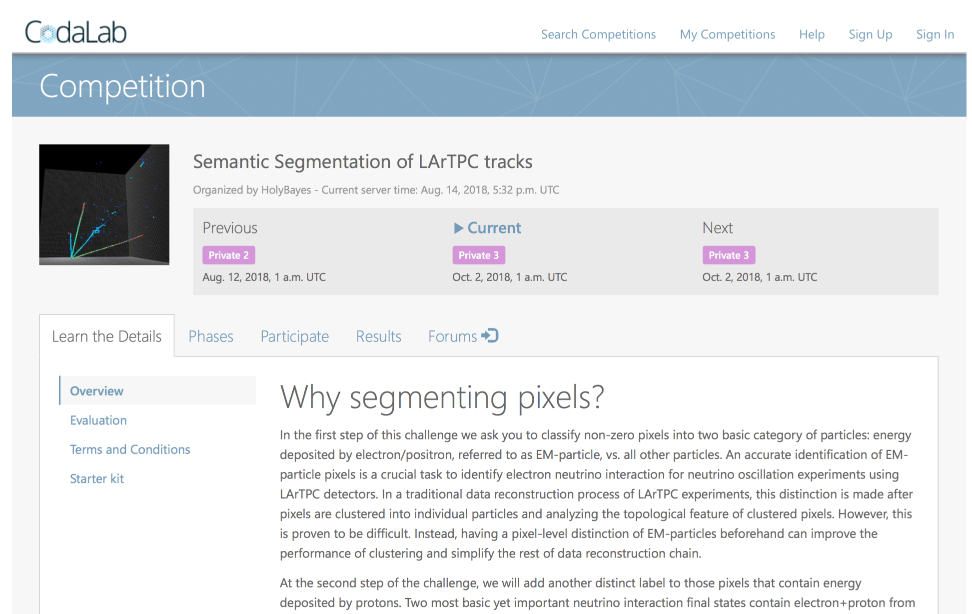
# Physics Applications
# Interested in? Let us work together!



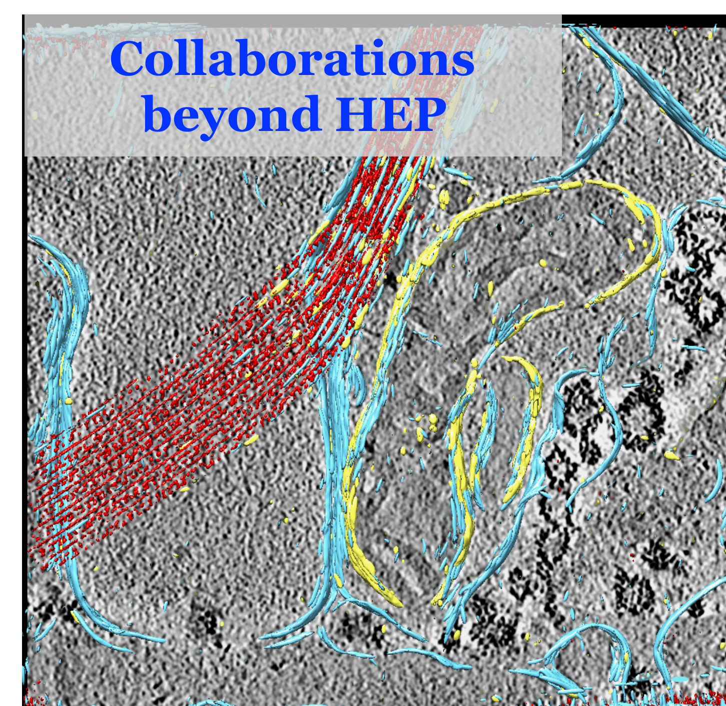

## DeepLearnPhysics (deeplearnphysics.org)

- **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**
  - Workshops (done at many universities/national labs)
  - Sharing opportunities (talks, jobs/fundings, etc.)

**Hands-on workshop @ SLAC/Stanford**

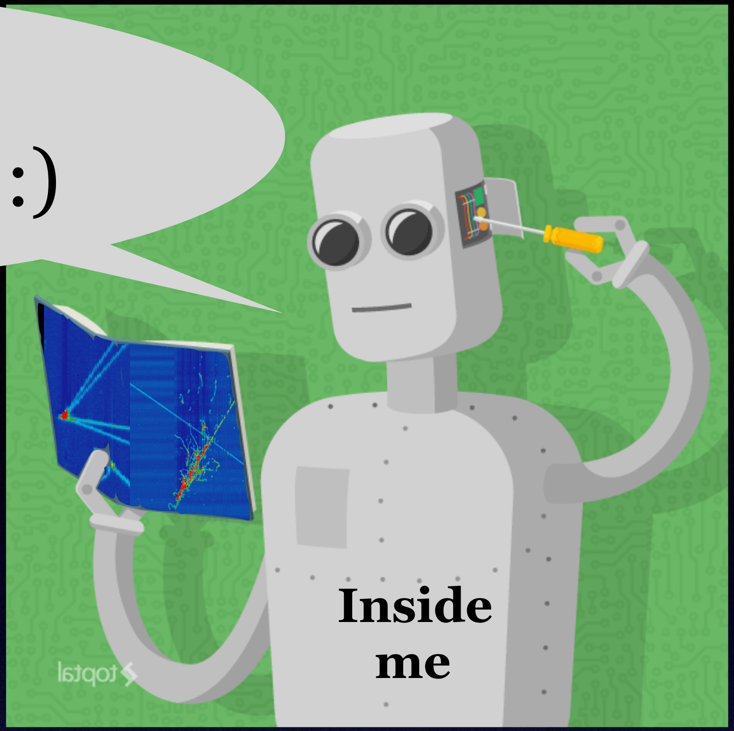



**Public challenge (collab. w/ LHC)**



**Collaborations beyond HEP**

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




# Take-away messages…

1. **CNNs** are **image feature extractors**

2. **CNNs** are **useful for many computer vision tasks** including…
   - **Image classification** & **Object detection** in an image
   - **Pixel segmentation** & **Clustering**

3. **DNNs** are **used in physics analysis & reconstruction tasks**

4. **DNNs/ML** are **becoming more popular, and we're learning…**
   **(join DLP!)**

# More Exciting Talks to Follow!

# **Things avoided in my talk** but popular…

- **Graph neural network**
  - Emerged from social network analysis, very popular
- **Generative models**
  - Including GANs (e.g. "Fast simulation" in LHC)
- **Recurrent neural network**
  - Sequence analysis (language as well as physics!)
- **Hyper-parameter optimizations**
- **ML on distributed systems** (HPCs)
- **Quantum X** (ML, neural network, algorithms, etc)
  - No idea, please don't ask me about this

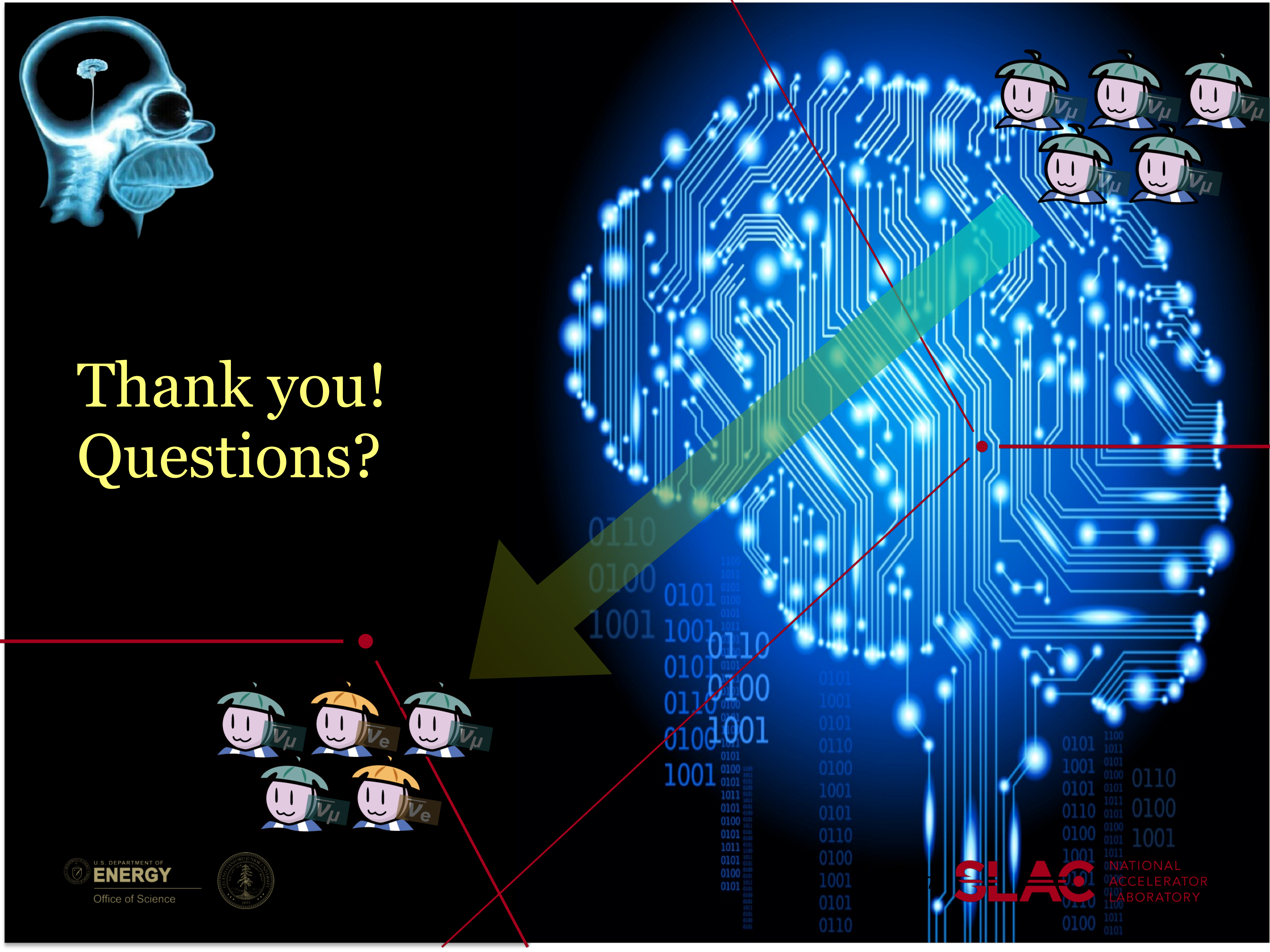
# Glossaries

## Convolutional Neural Networks (CNNs)

- A **convolution filter** (neuron) applies a small **kernel** to input data, and produces a **feature-map**.
- A **convolution layer** consists of **multiple filters**, and the output of a layer is another **"image" matrix data with many channels**
- CNNs are typically made of **successive down-sampling and convolution operations**, and "many" layers = "**deep**" CNNs.

## Applications

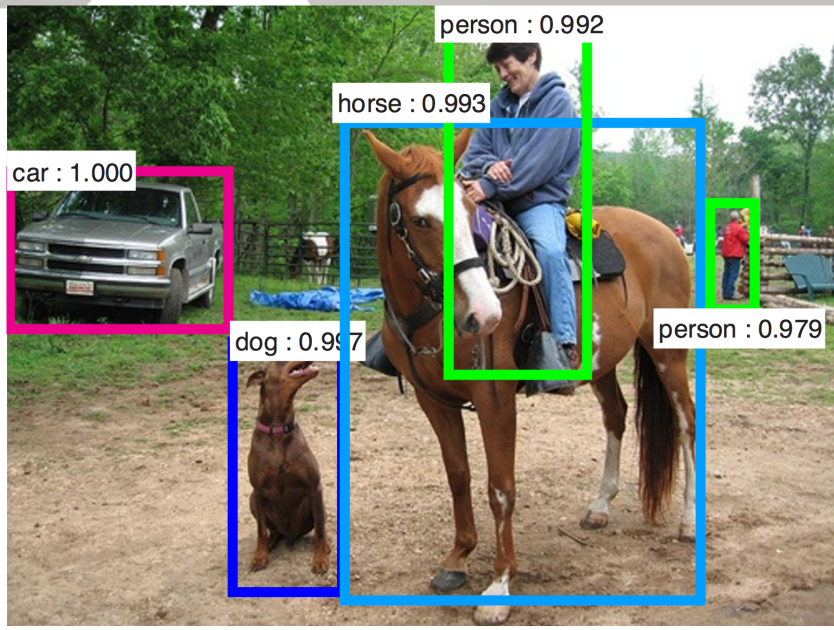
- Computer vision: **image classification**, **object detection**, **semantic segmentation**, **clustering**


Thank you!
Questions?
$\nu_\mu$
$\nu_e$
U.S. DEPARTMENT OF
ENERGY
Office of Science
SLAC
NATIONAL
ACCELERATOR
LABORATORY

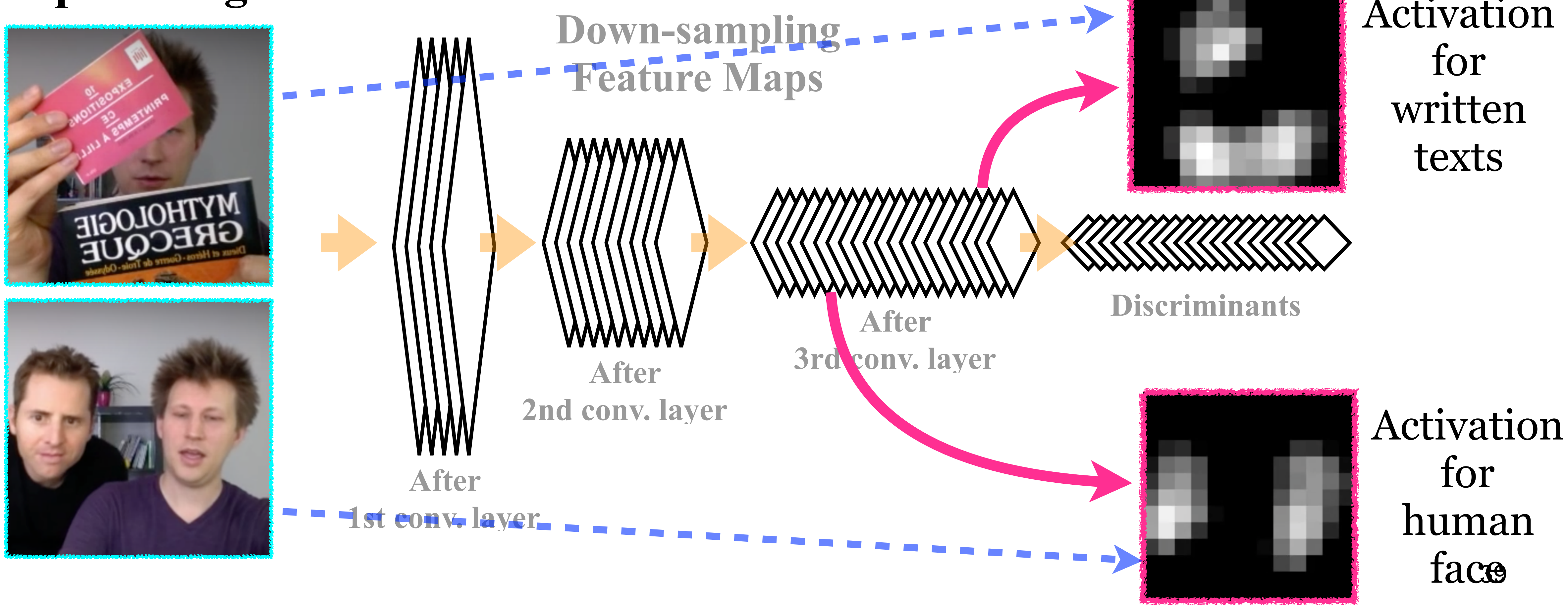
# Backup Slides

# 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!

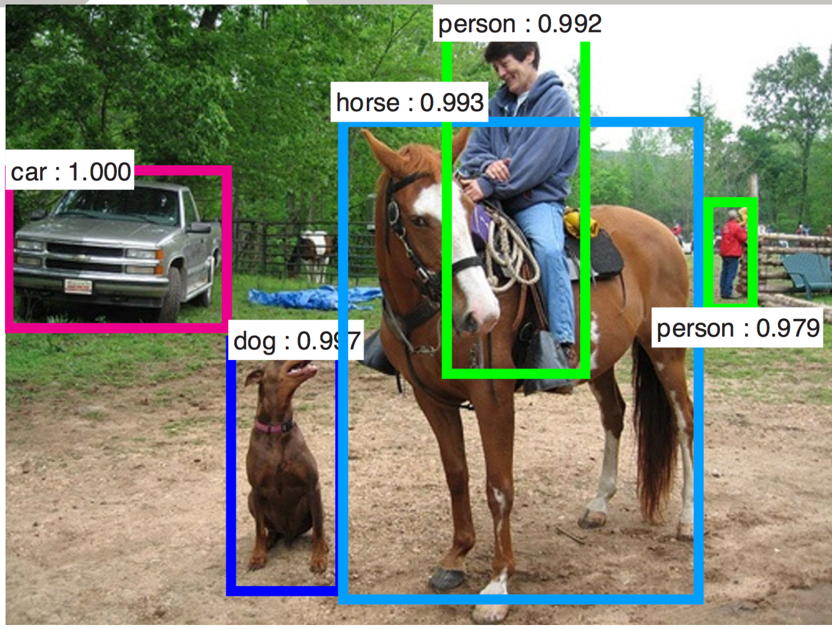



**Input Image**

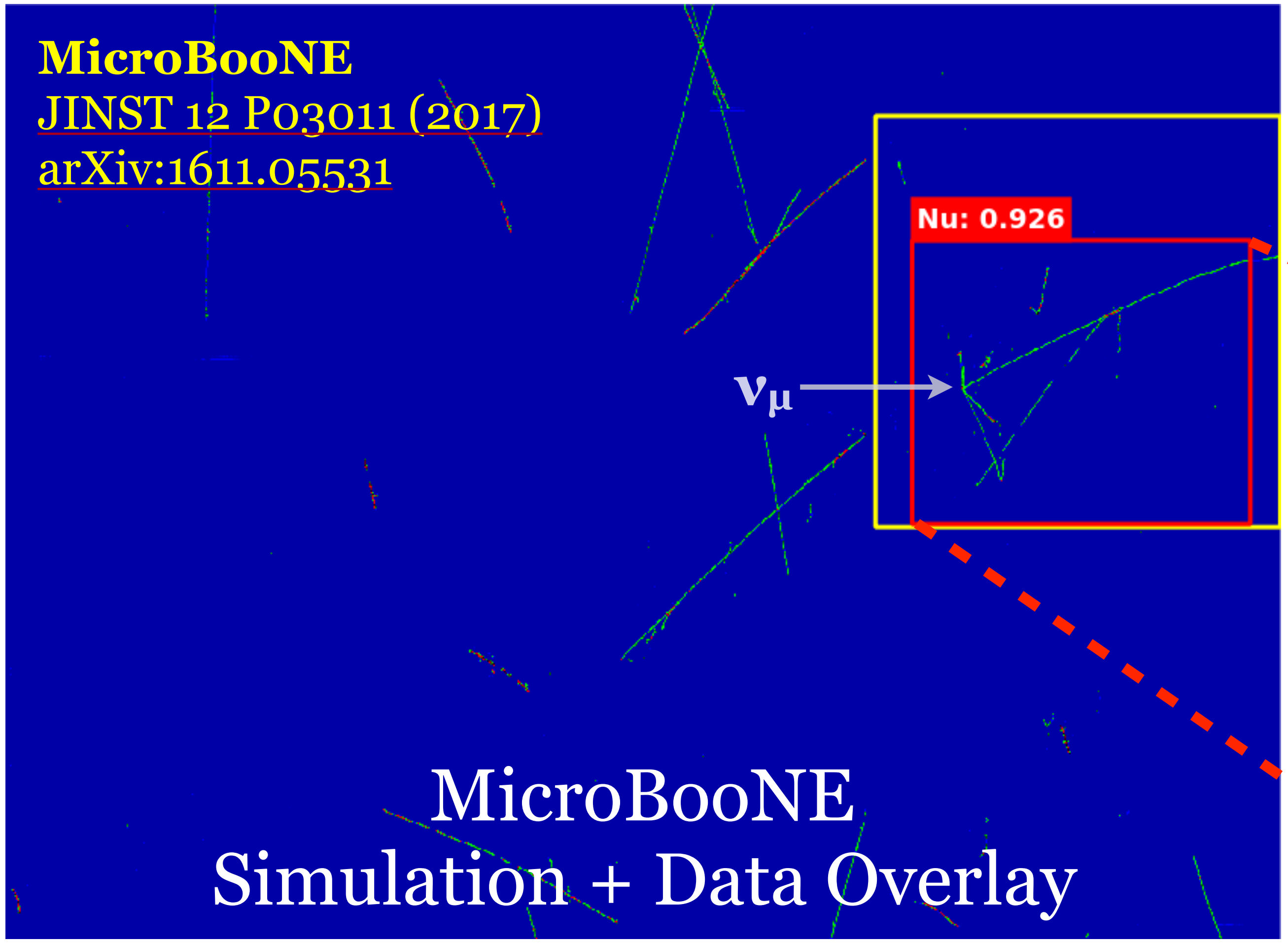


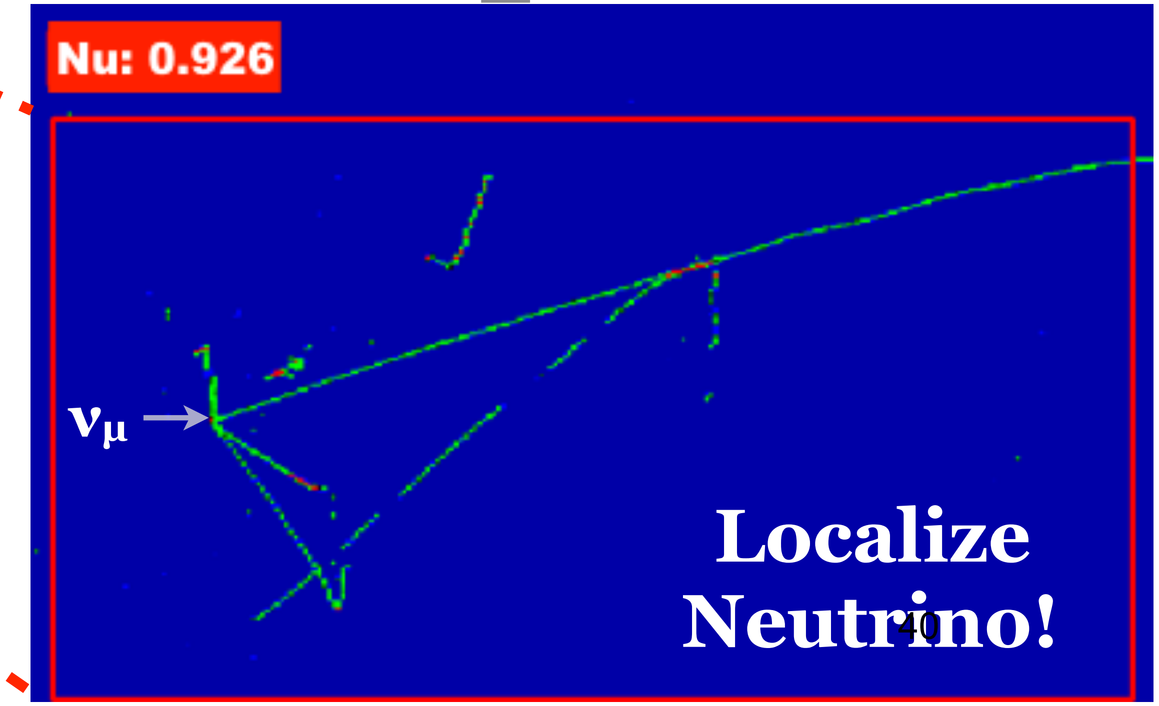
# Beyond image classification: object detection

- **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!

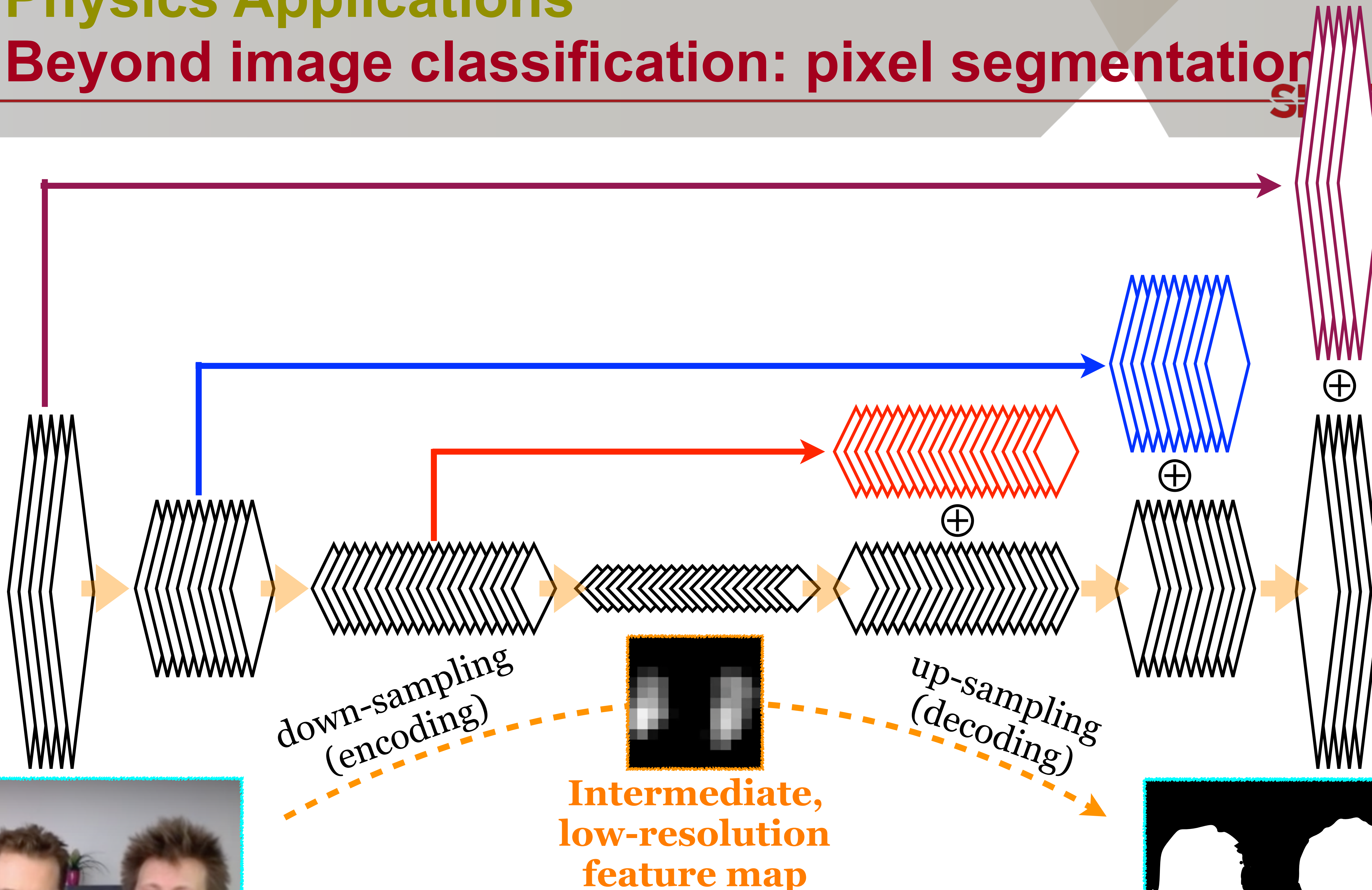


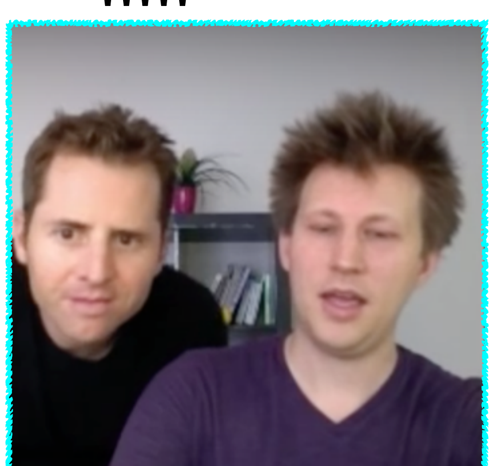

See MicroBooNE Paper!

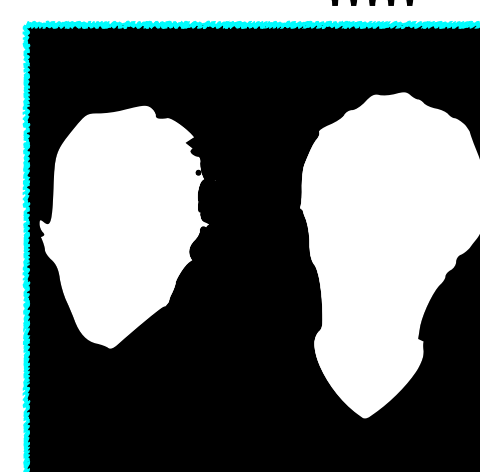


# Beyond image classification: pixel segmentation

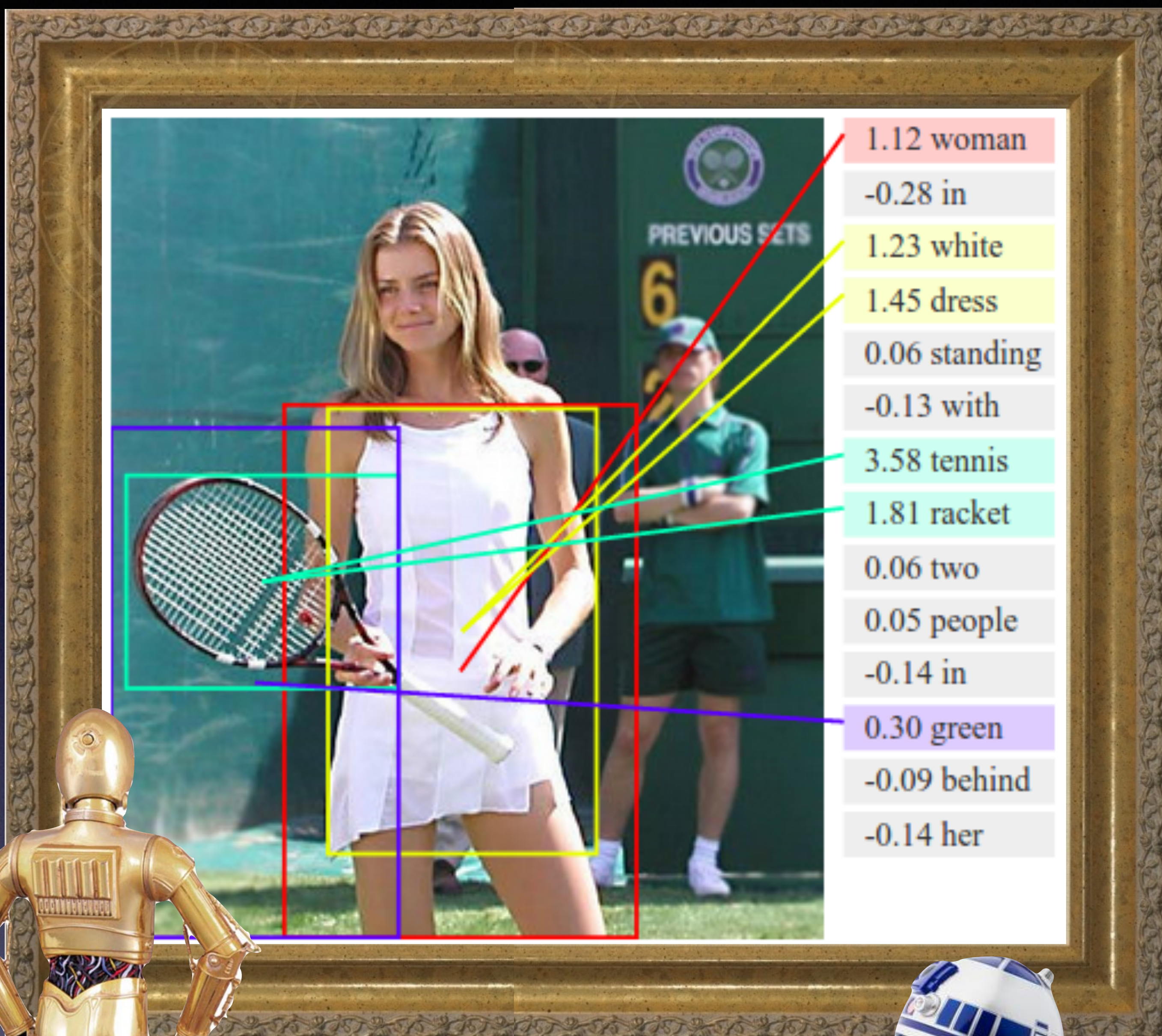




- Combine "up-sampling" + convolutions
- Outcome: "learnable" interpolation filters

## Image context analysis



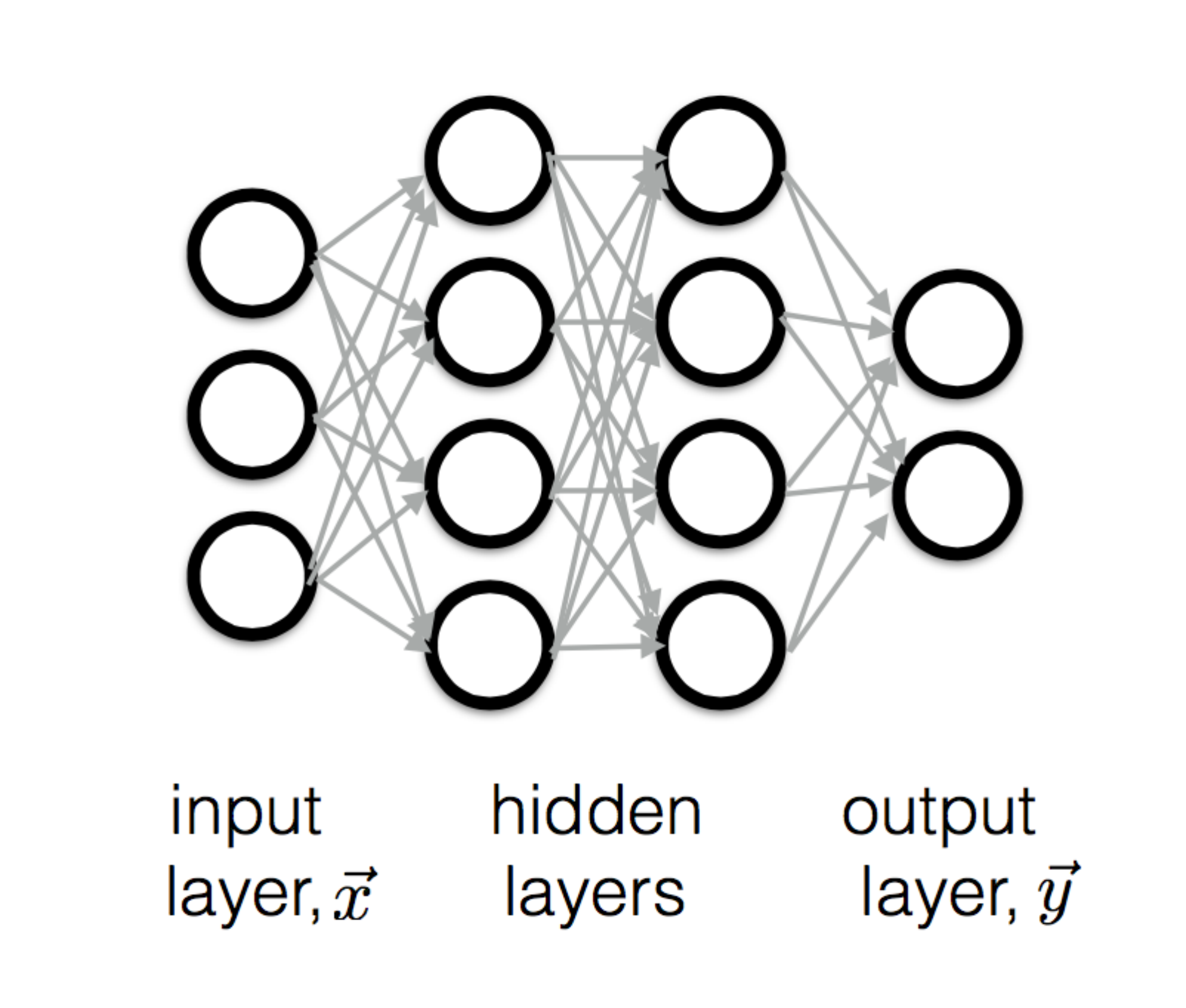

## “Pose” detection


# Convolutional Neural Network

*~ How does it work? ~*

# “Classical” Neural Net

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



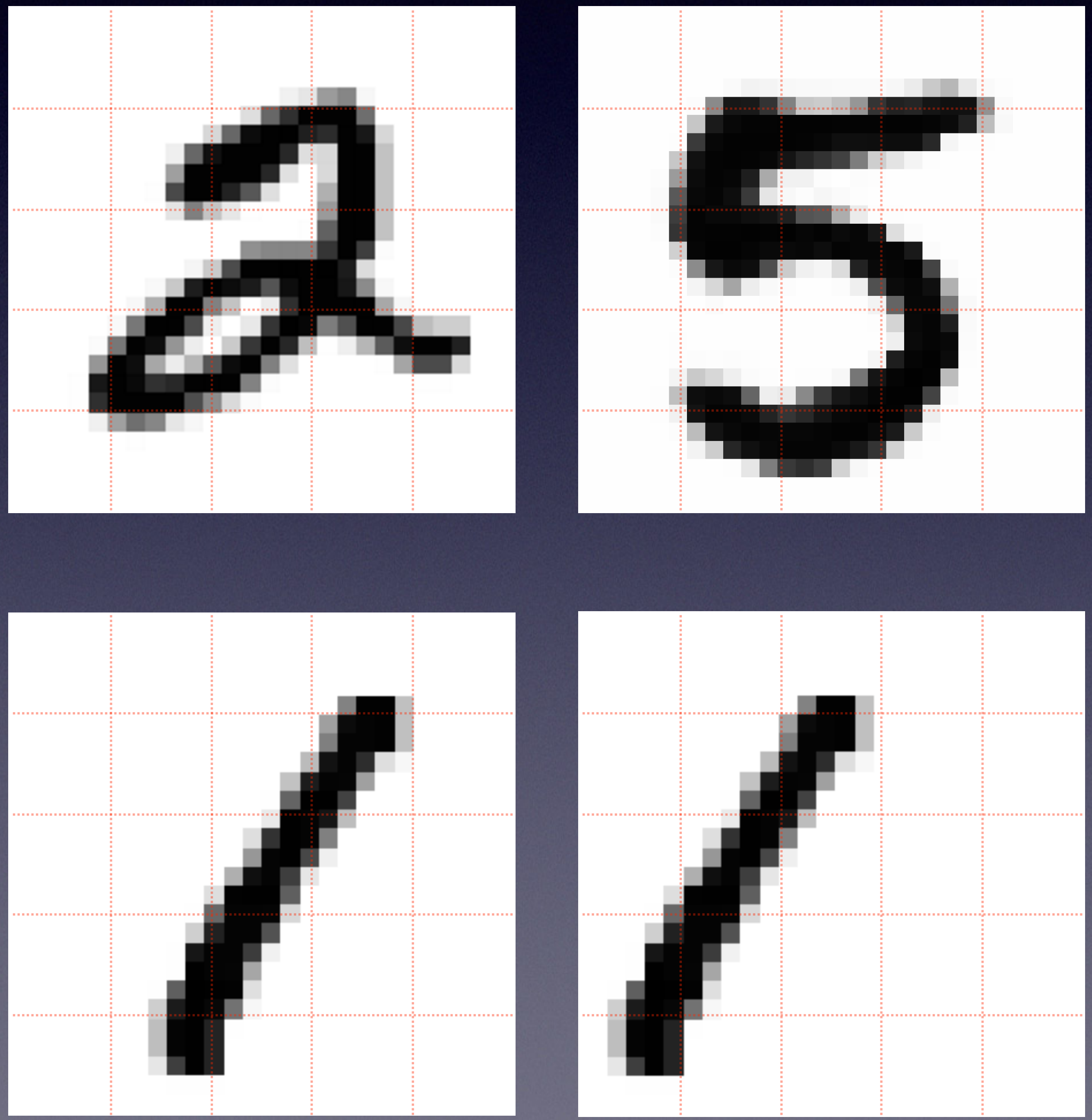

A traditional neural network consists of a stack of layers of such neurons where each neuron is ***fully connected*** to other neurons of the neighbor layers

# “Classical” Neural Net

## … is not ideal for image classification …

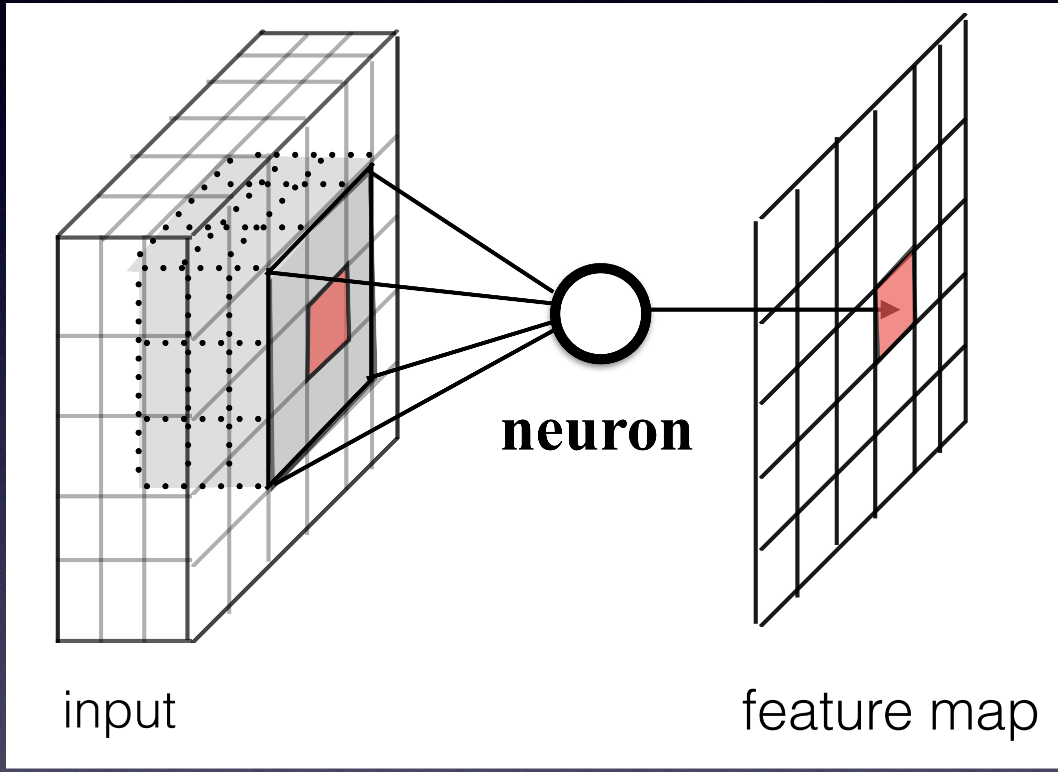
### Image classification

- **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

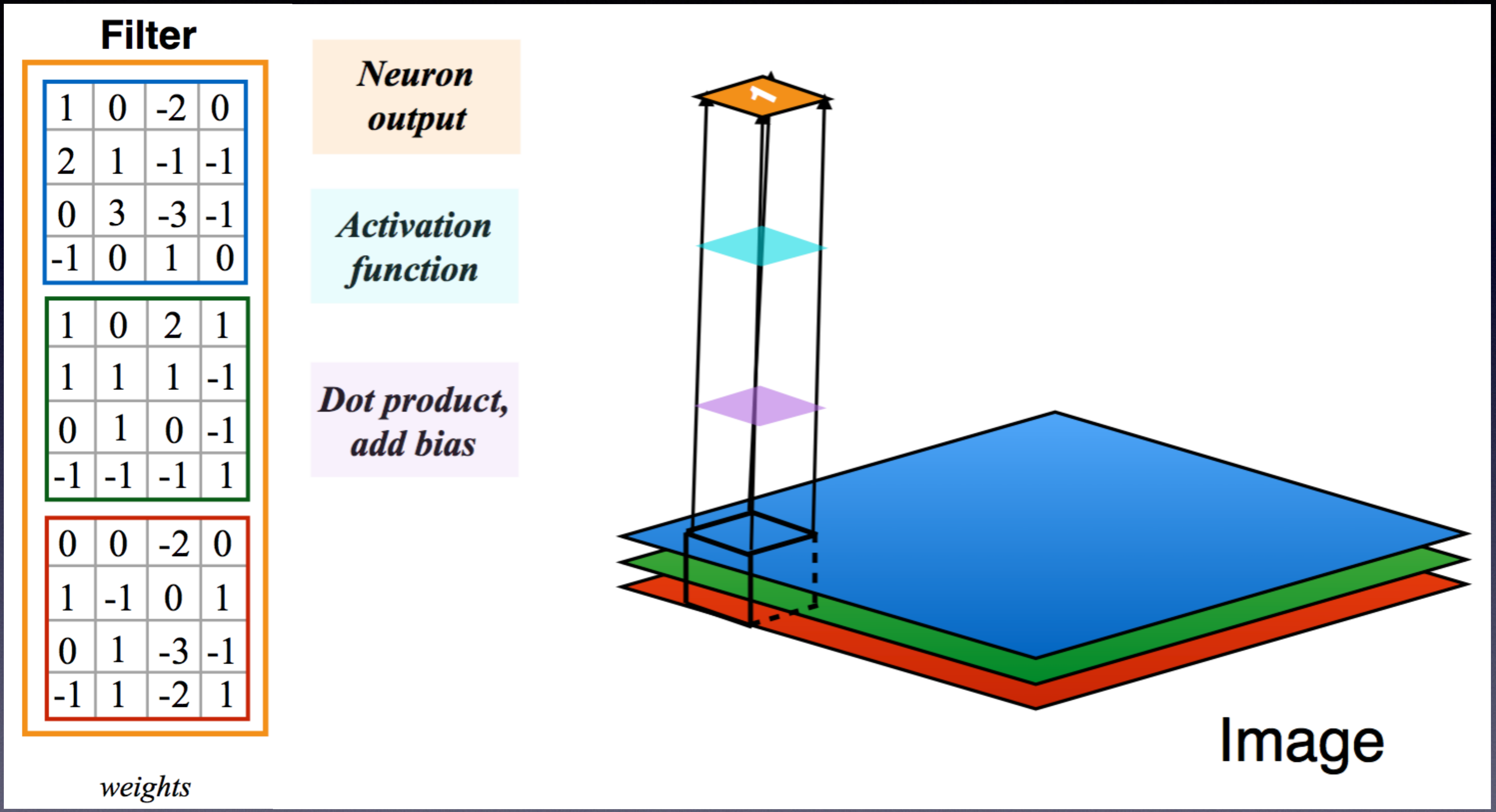



$$f_{i,j}(X) = \sigma\left(W_i \cdot X_j + b_i\right),$$

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

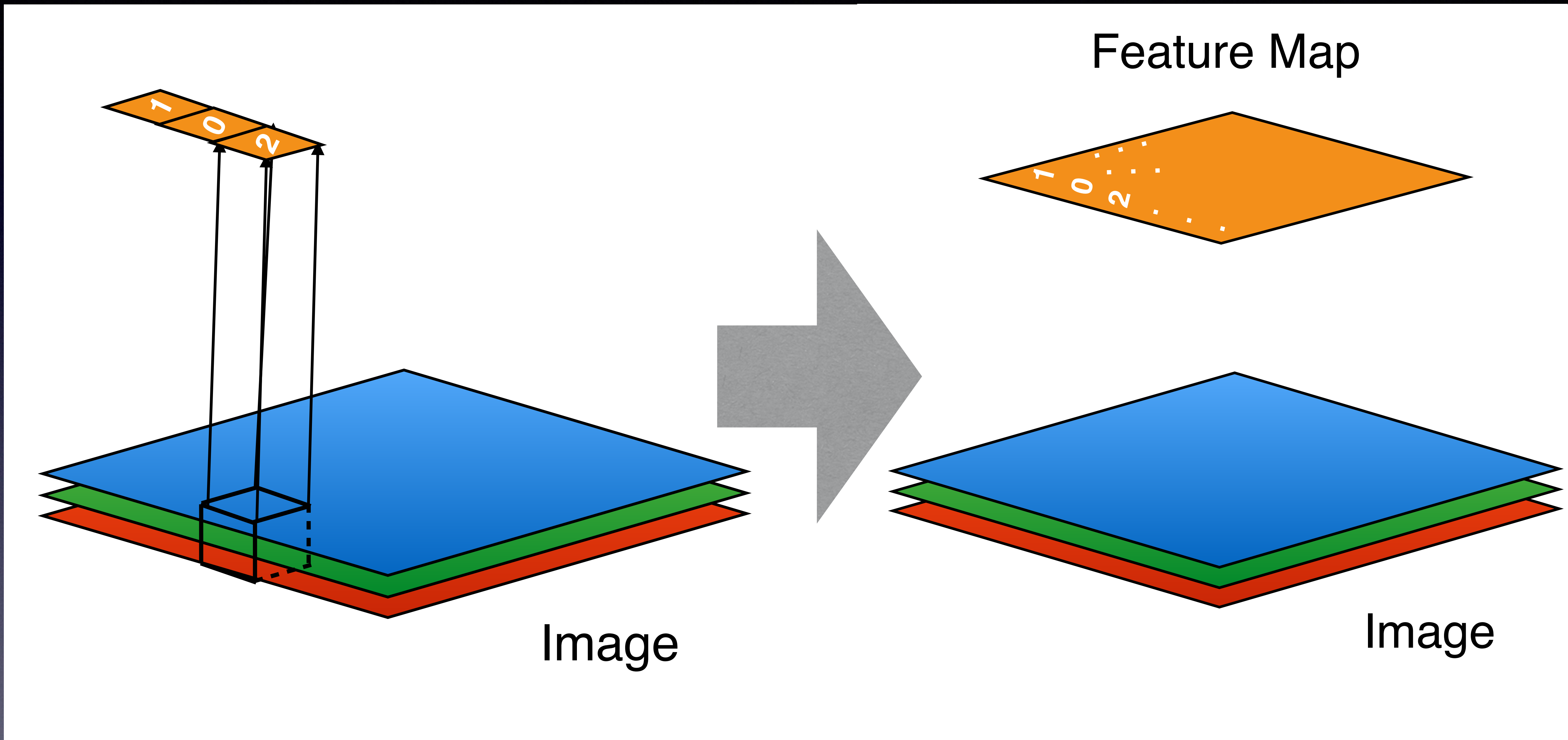
- Translate over 2D space to process the whole input
- Neuron **learns translation-invariant features**
  - Suited for a "***homogeneous***" detector like LArTPC
- **Output**: a "feature-enhanced" image (***feature map***)

# Convolutional Neural Networks



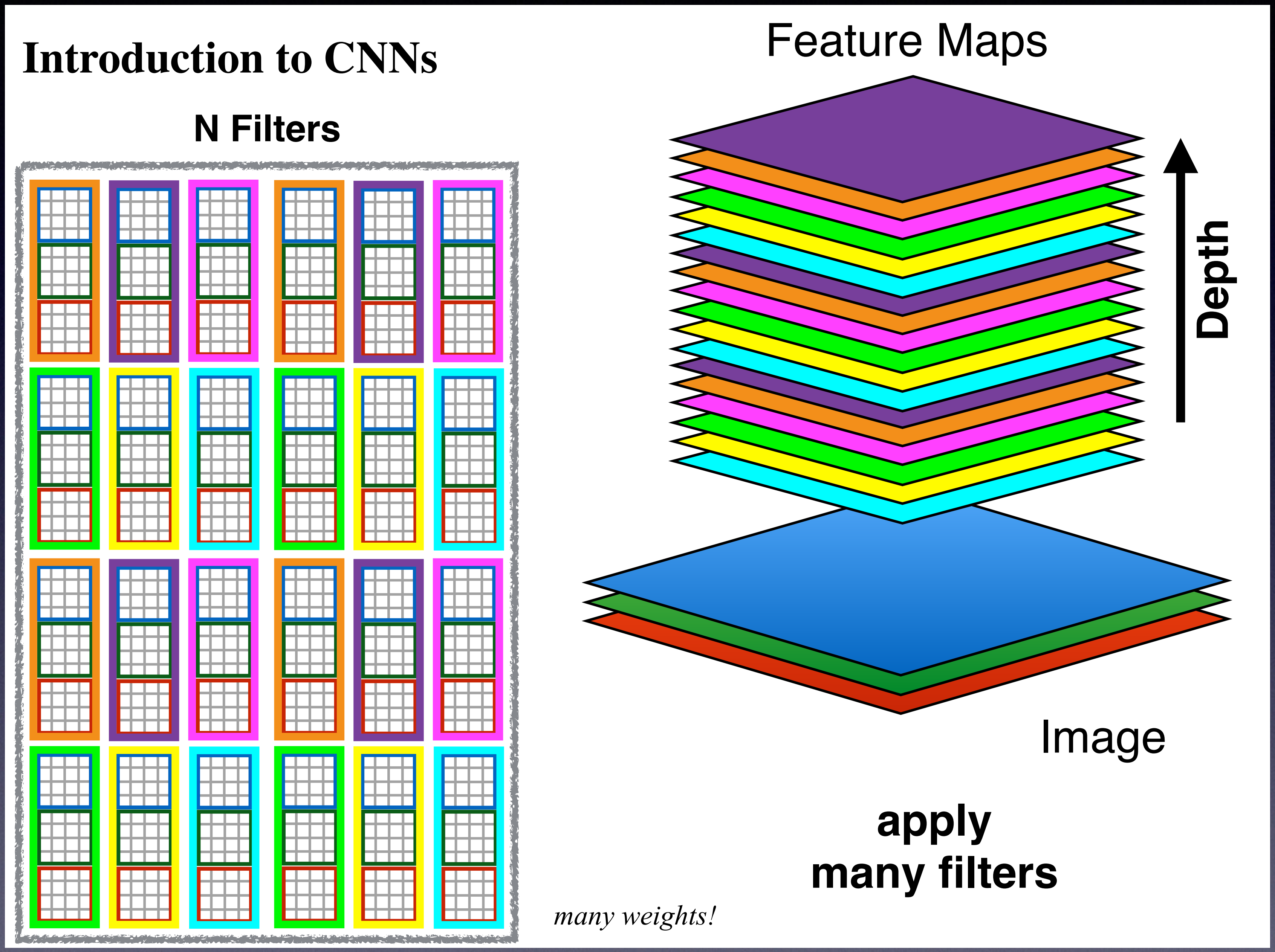

Toy visualization of the CNN operation

# Convolutional Neural Networks




Toy visualization of the CNN operation

# Convolutional Neural Networks

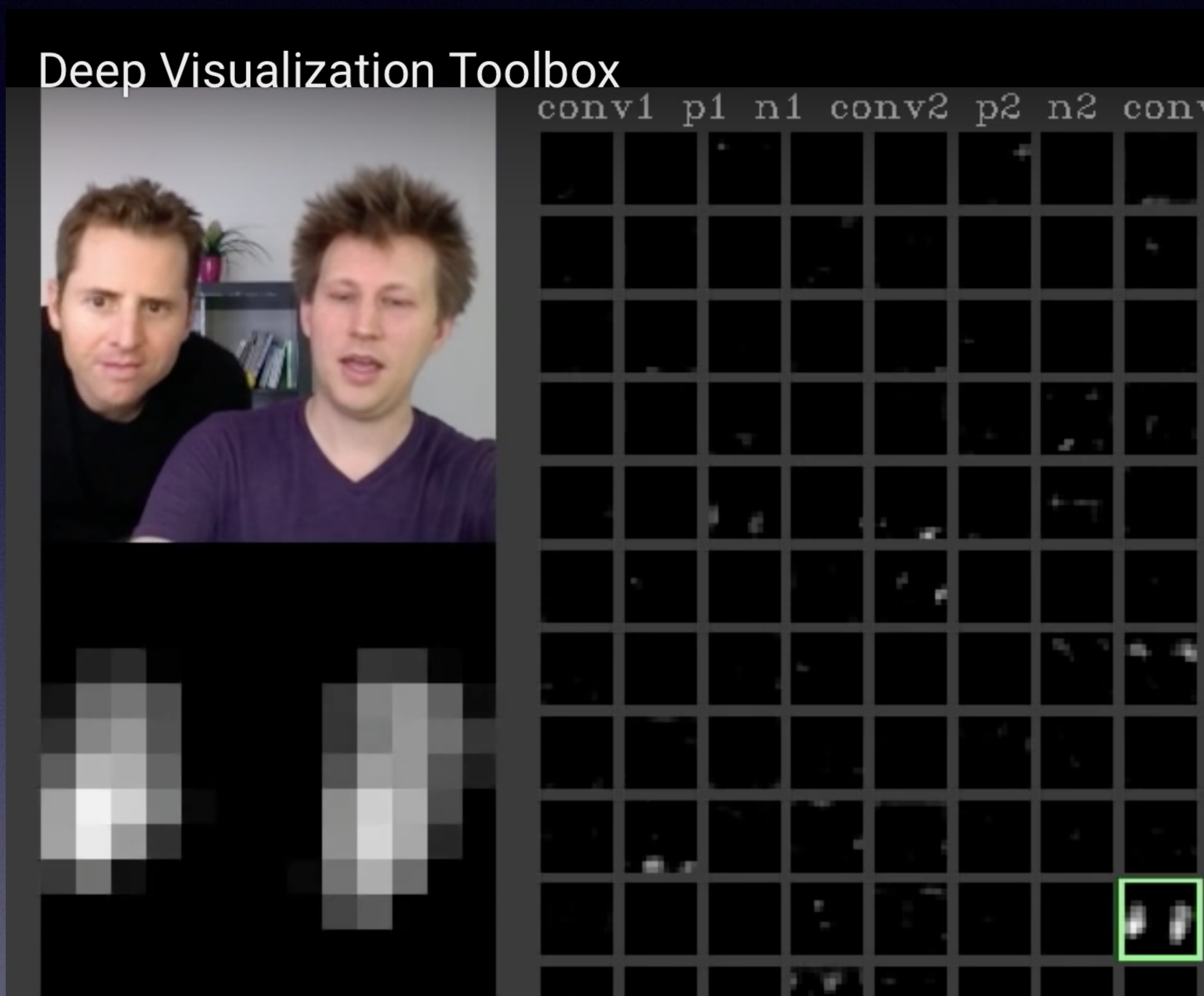



Toy visualization of the CNN operation

# How Image Classification Networks Work

## Feature map visualization example

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




Neuron concerning face

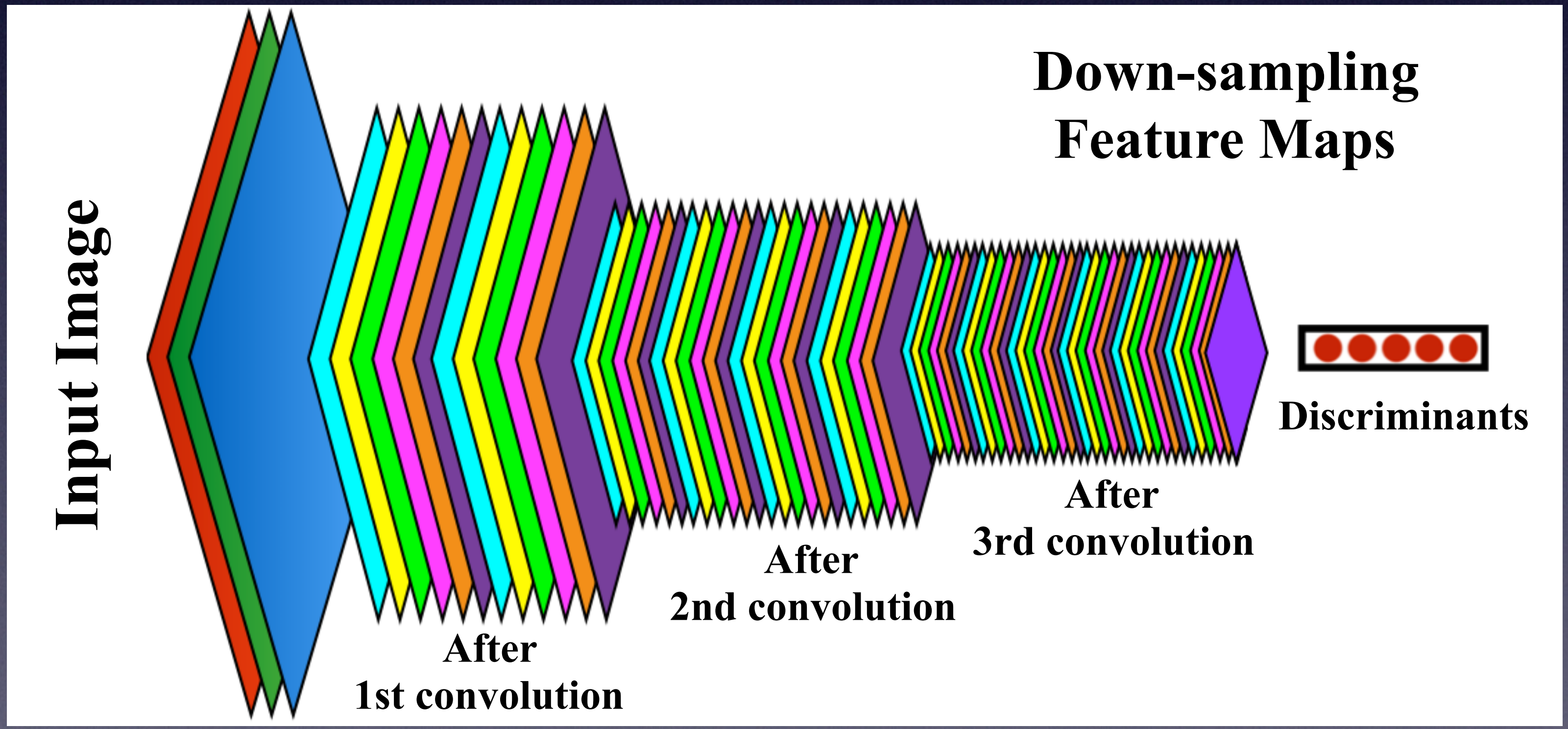

Neuron loving texts

(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**

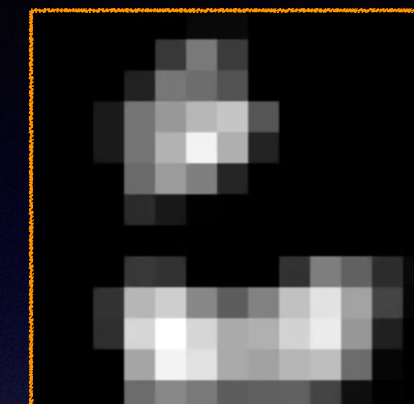


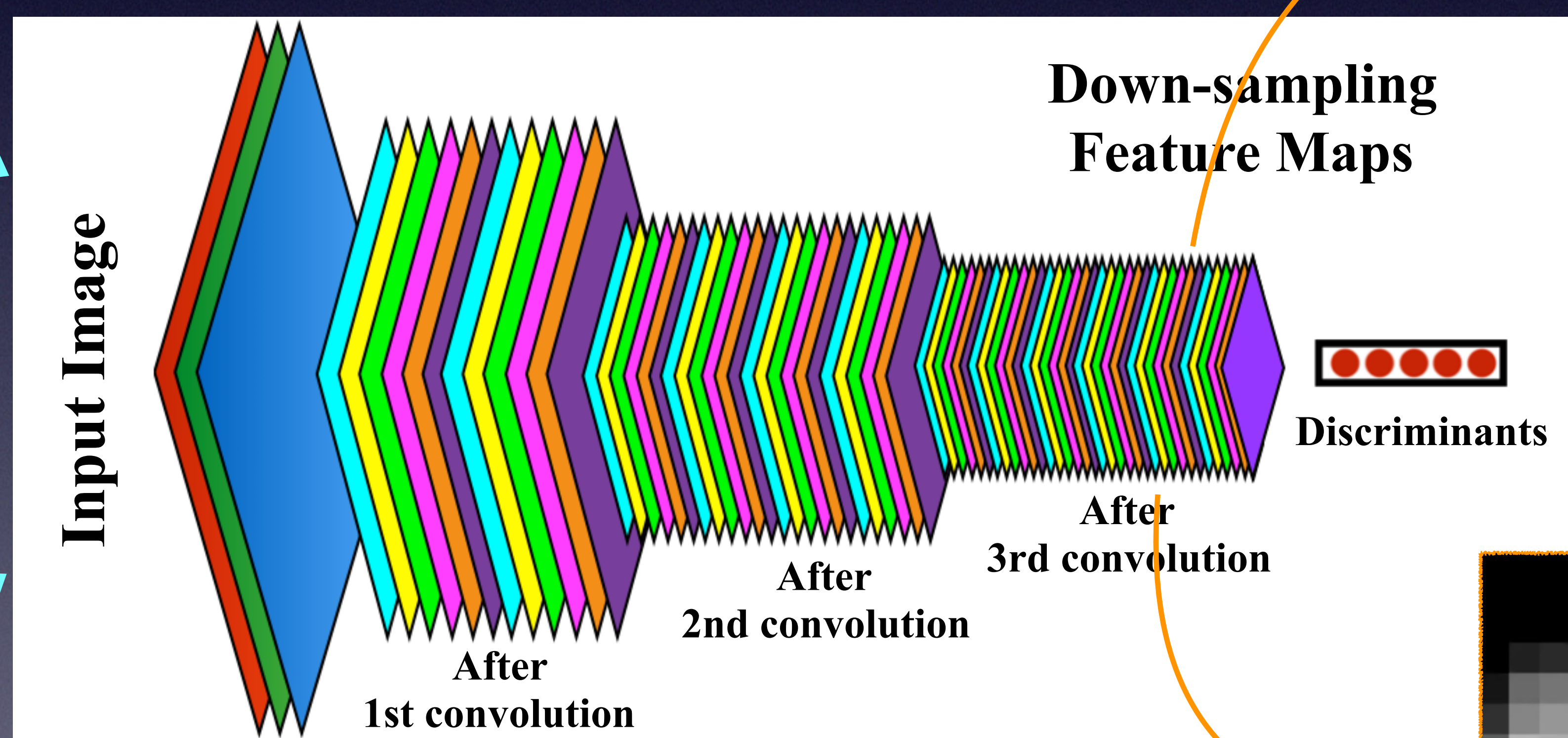

Series of convolutions
+ down-sampling

# How Image Classification Networks Work

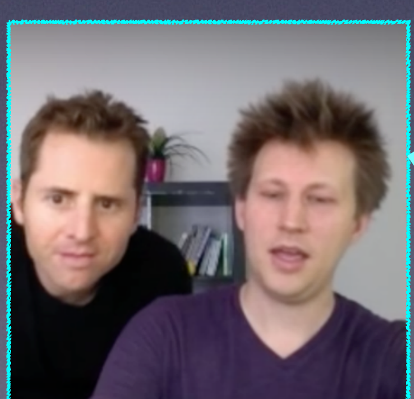
**Goal**: extract features to give “single label” to an image

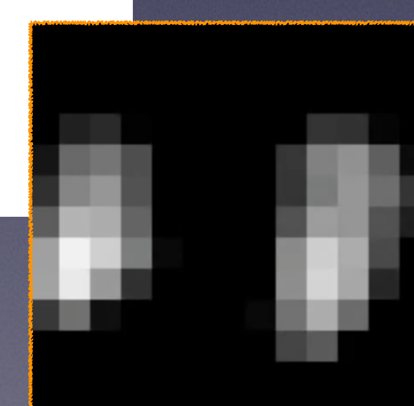
1. **Convolution operation**
2. **Down-sampling**




“Written Texts” feature map





“Human Face” feature map

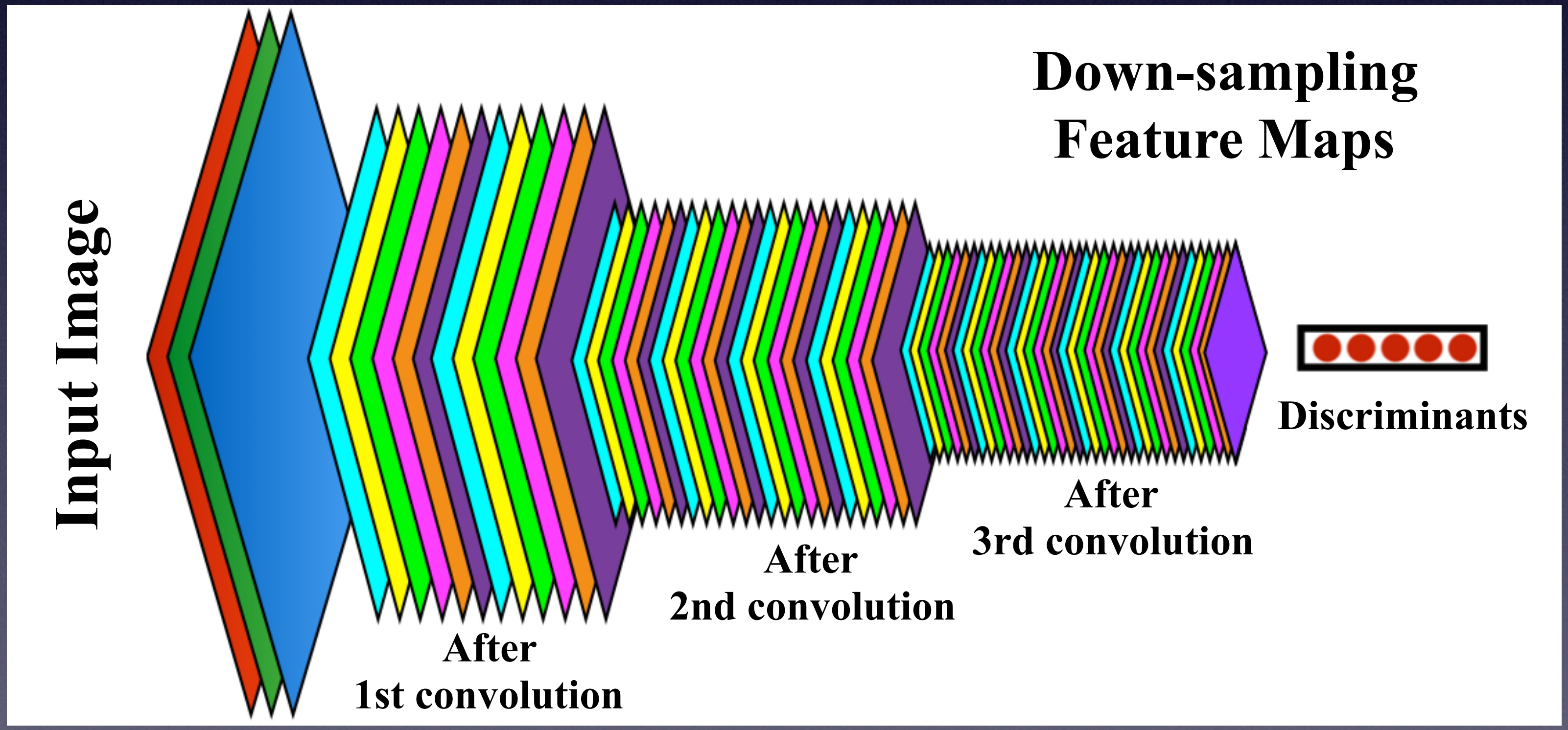
Series of convolutions
+ down-sampling

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**

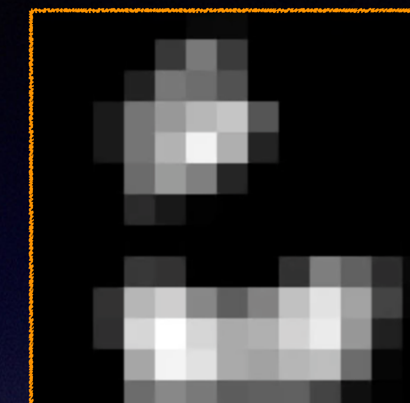



Series of convolutions
+ down-sampling

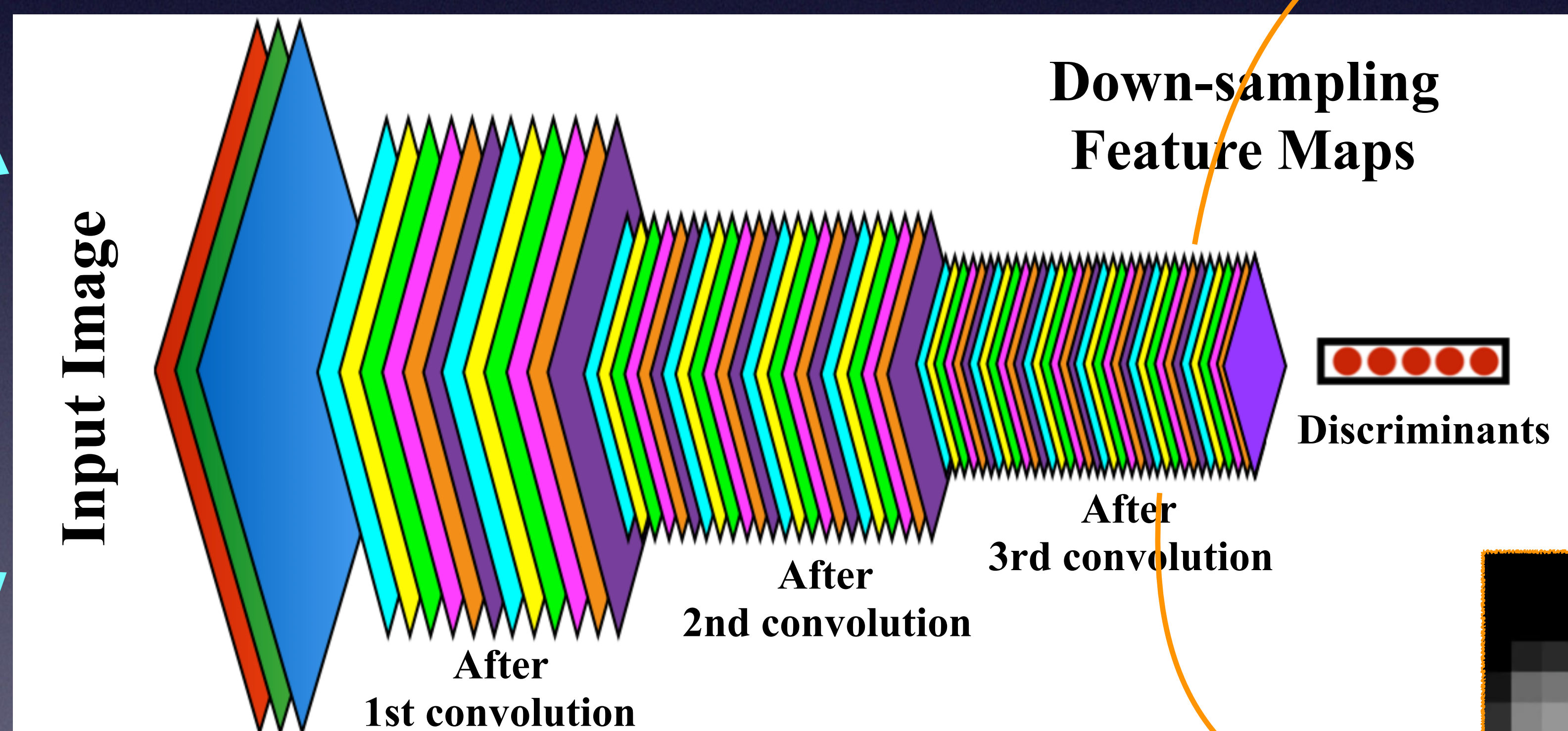
# How Image Classification Networks Work

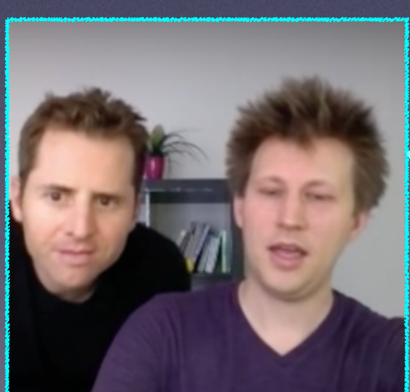
**Goal**: extract features to give “single label” to an image

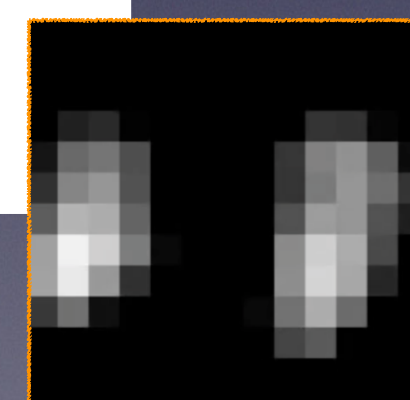
1. **Convolution operation**
2. **Down-sampling**



“Written Texts” feature map





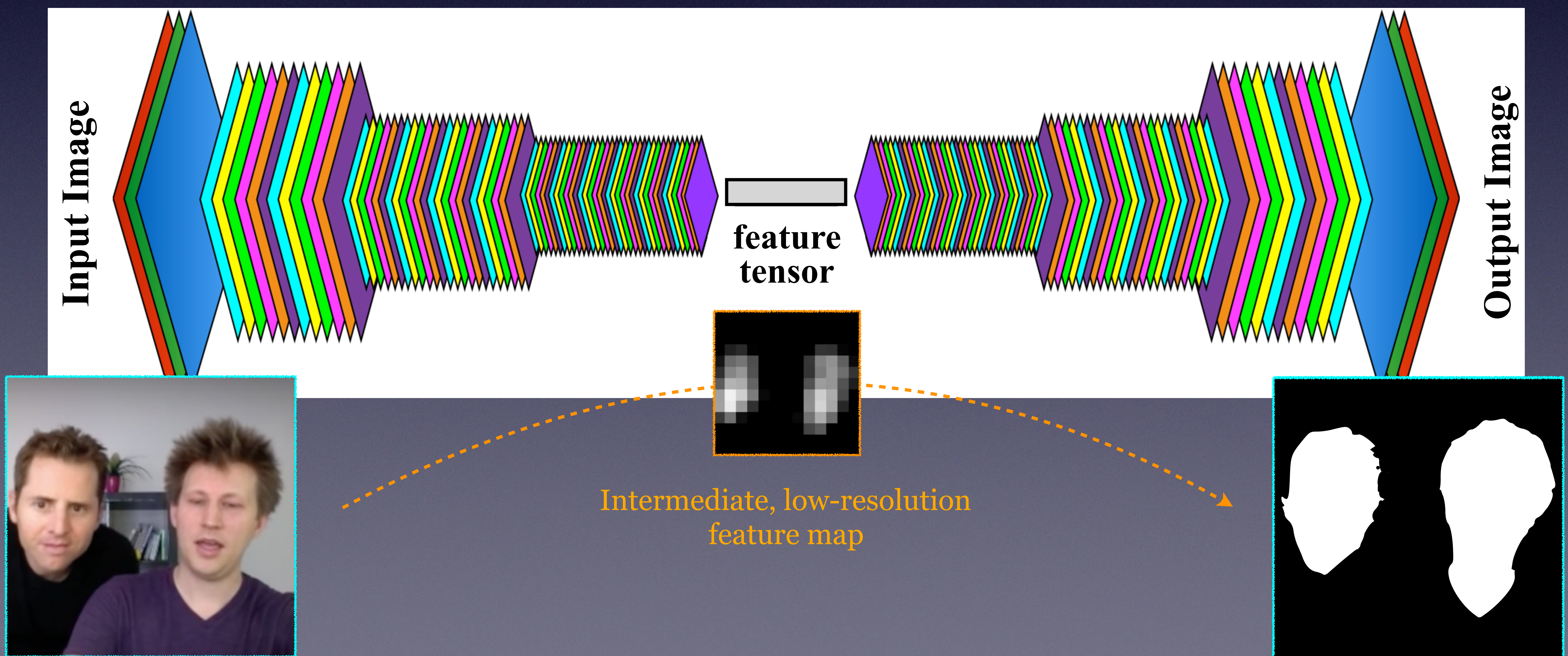
“Human Face” feature map

Series of convolutions
+ down-sampling

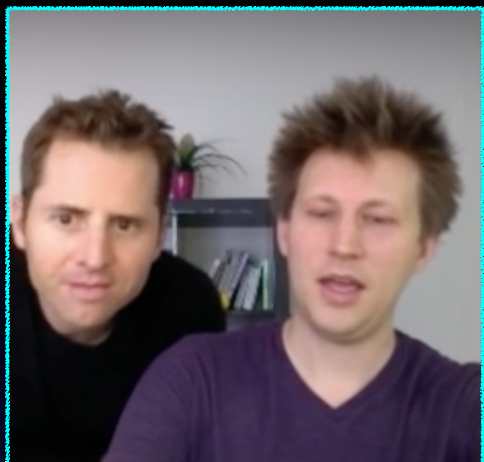
# How SSNet Works

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

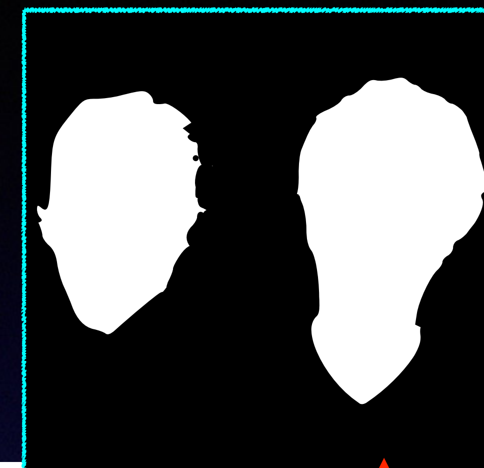
1. **Up-sampling**
   - Expand spatial dimensions of feature maps
2. **Convolution**
   - Smoothing (interpolation) of up-sampled feature maps

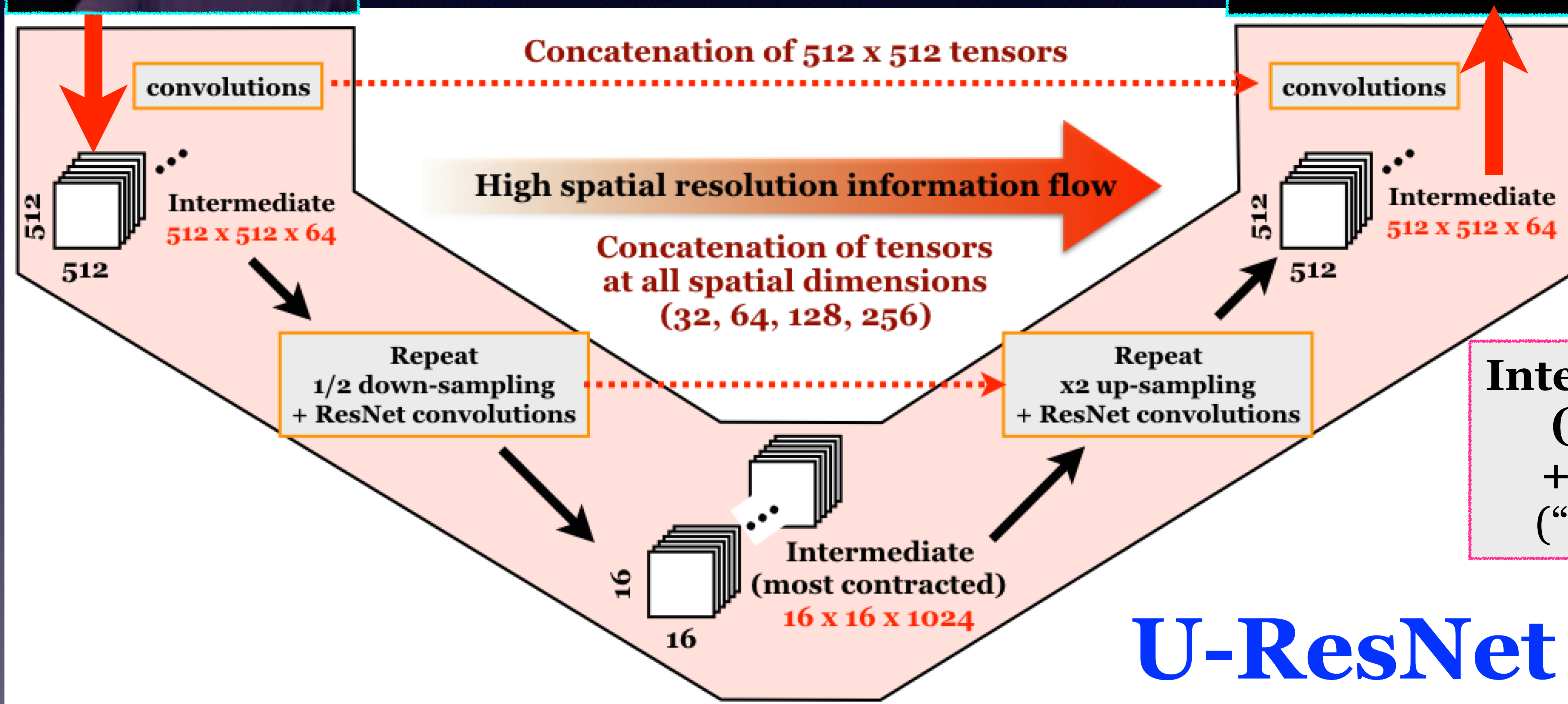


# DNN for LArTPC Data Reconstruction

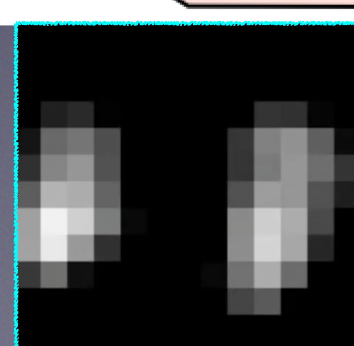


**How does U-ResNet Work?**

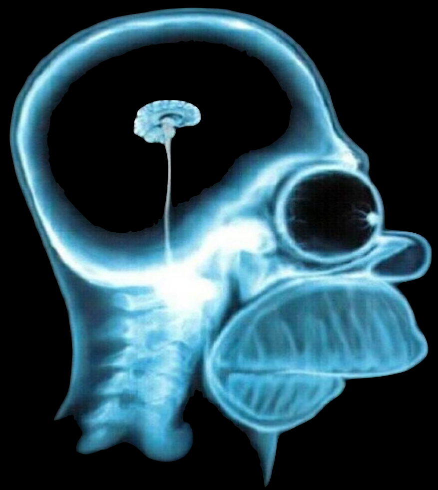




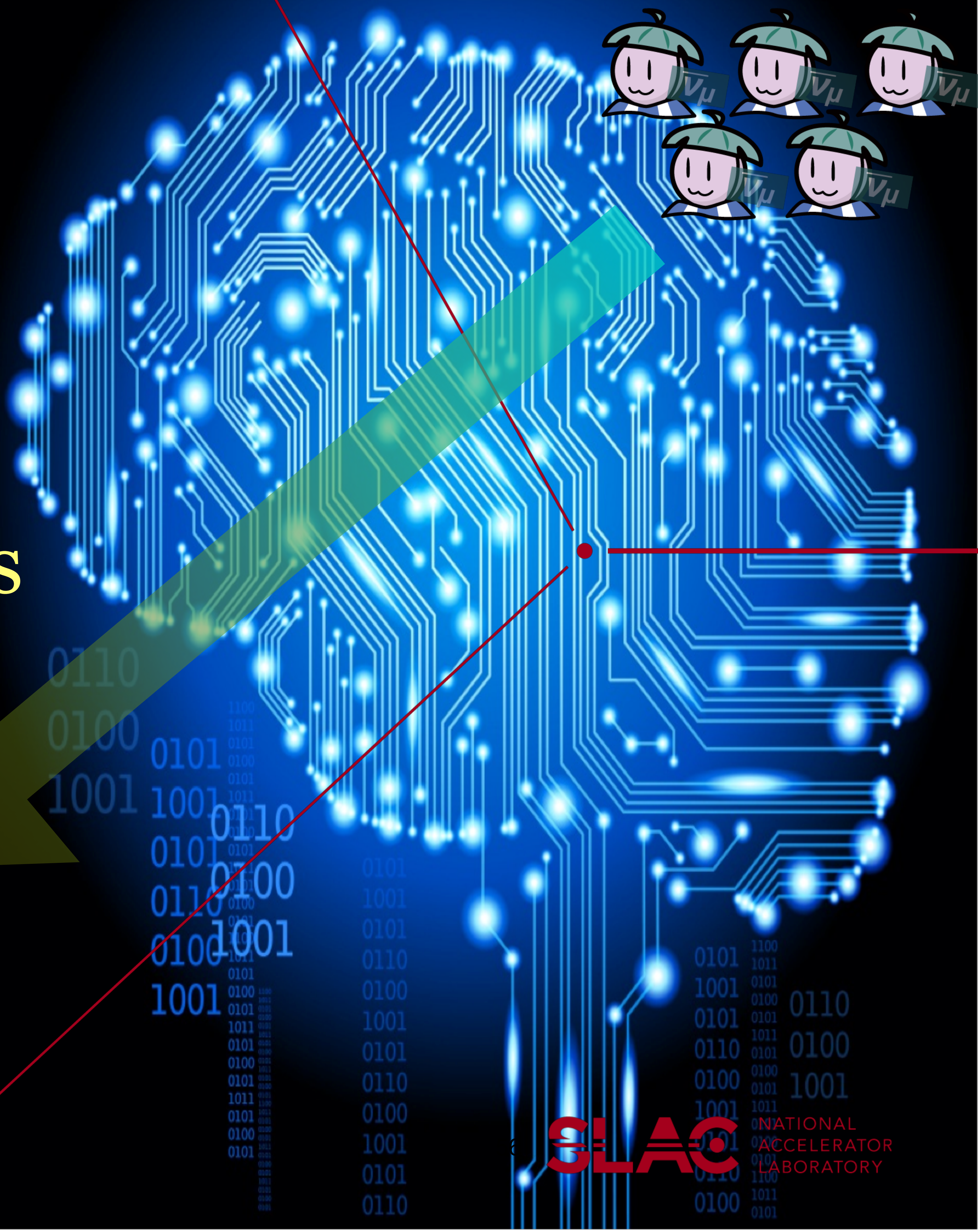

**Interpolation filters**
**(up-sampling)**
**+ Convolutions**
("learnable" filter)

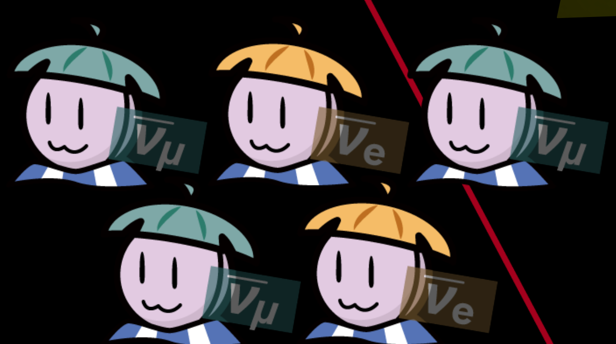


**Down sampling + Convolutions to identify**
highly abstract features (e.g. "human face")



# Path to Big Data Scalable Algorithms





U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

# Scalable ML Algorithms
# Machine Learning for LArTPC Image Analysis

## Scalable software is critical

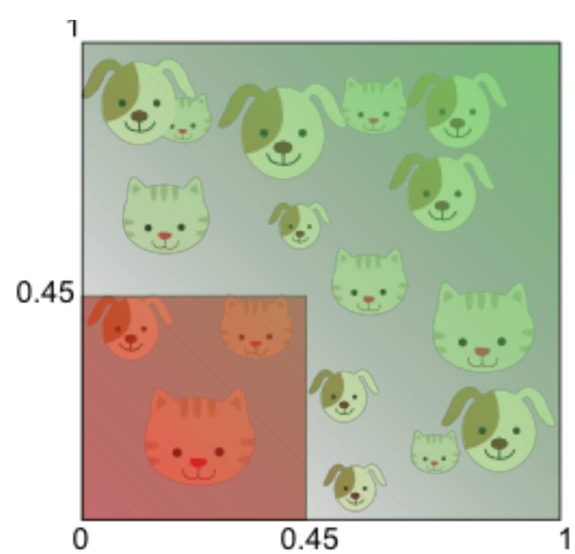
- **Big detector = big data** … can we run a data production?
- Combining more algorithms = more computing resource (CR)

## Challenges for ML on LArTPC data

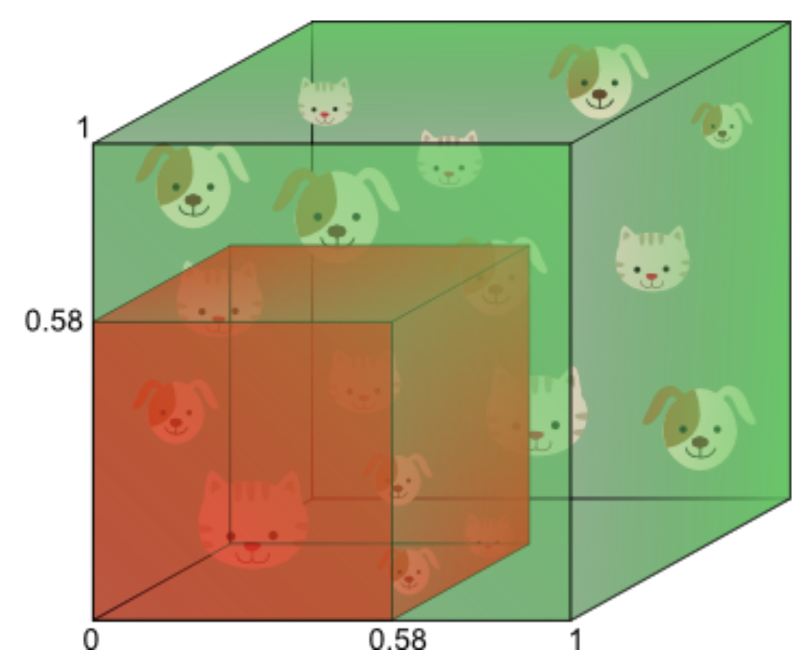
- ML in computer vision = linear algebra on matrix data
- CR for **scales by data size = power law** (^2 for 2D, ^3 for 3D)
- **LArTPC data** is unique: locally dense but **generally very sparse**
  - Only ~1% of pixels are non-zero in 2D images, and ~0.1% in 3D volumetric data
  - "Trajectory" is really 1D … **non-zero pixel count does not scale by power law!**

**"curse of dimensionality"**


**1D**



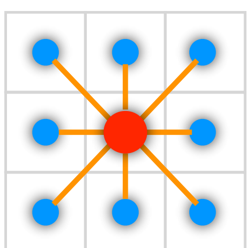

**2D**



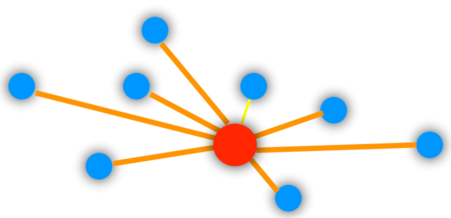

**3D**

## Two independent solutions pursued

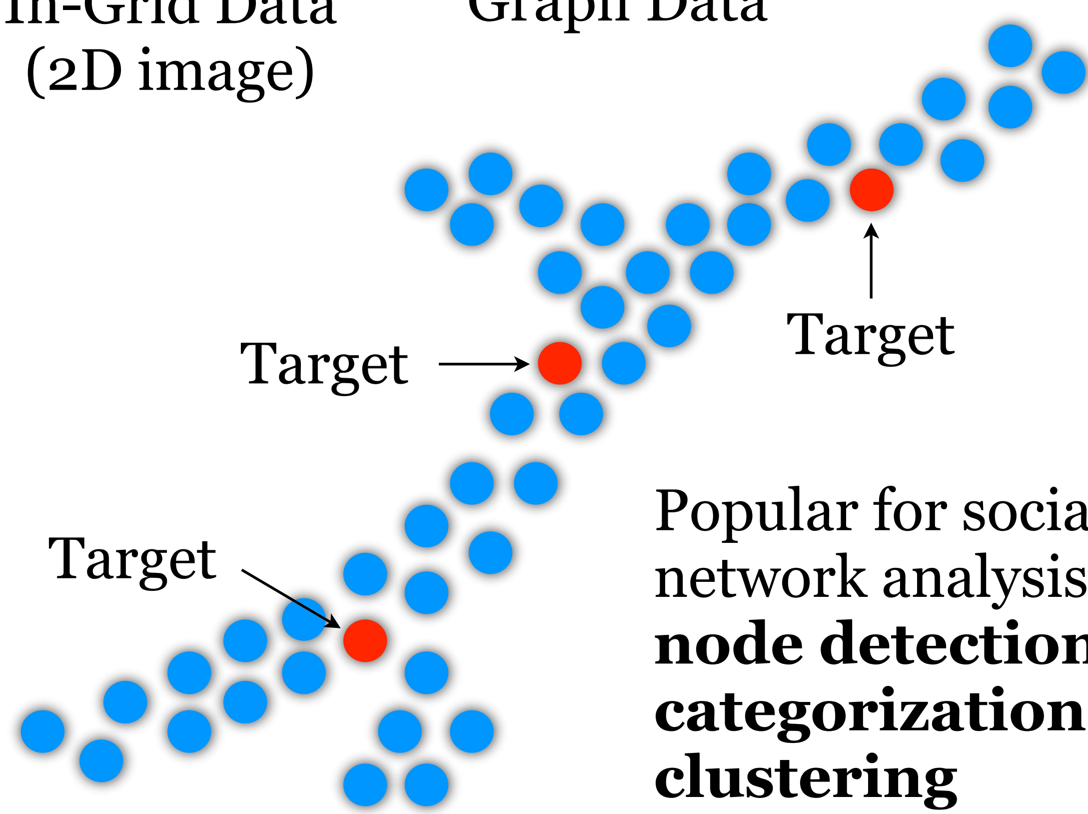
- **Sparse linear algebra** … efficient operation, ignores zero pixels (ZP)
- **Graph neural network** … efficient data representation, eliminate ZP



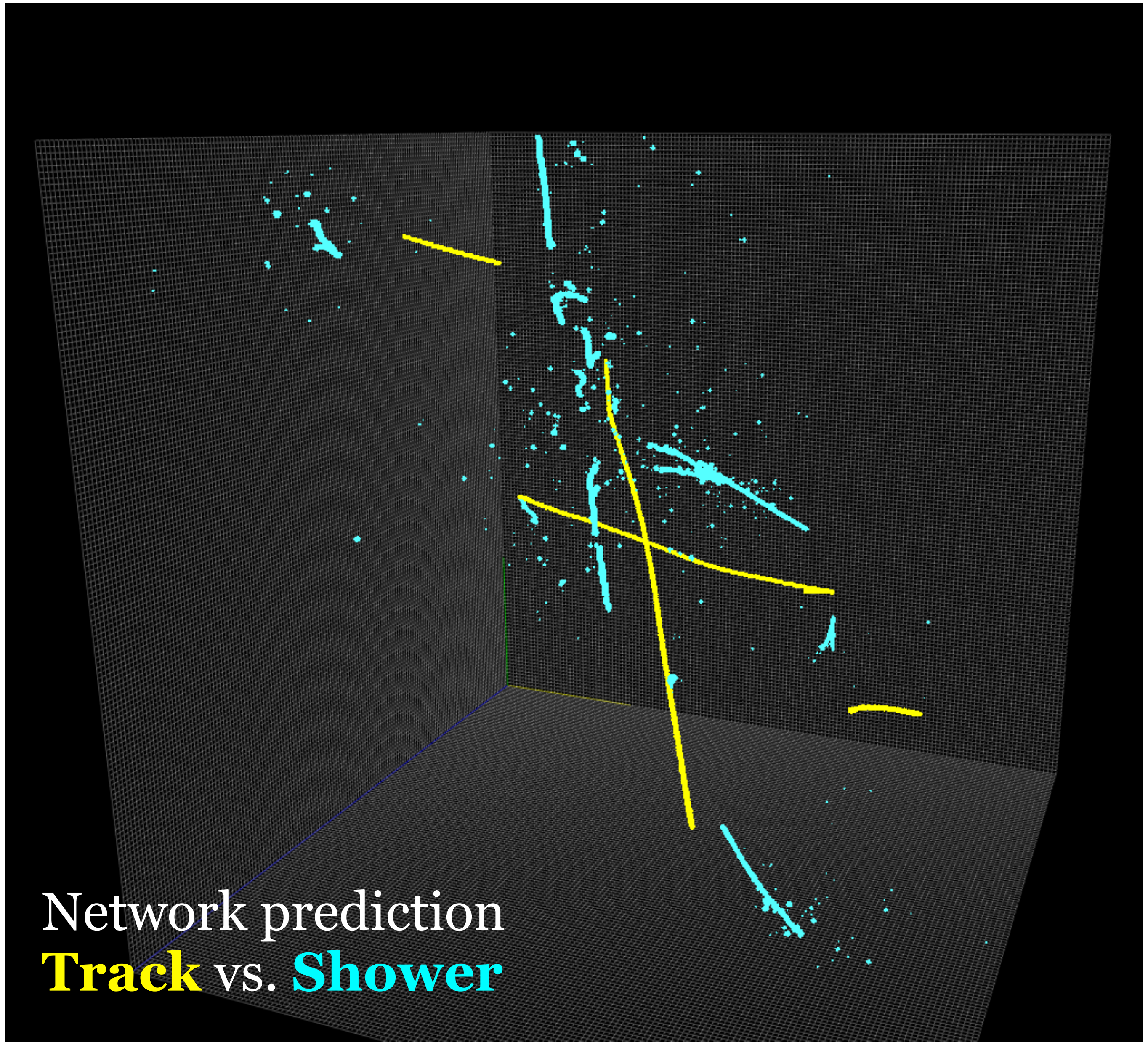
In-Grid Data (2D image)



Graph Data



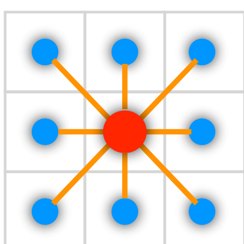

Popular for social network analysis, **node detection, categorization, clustering**



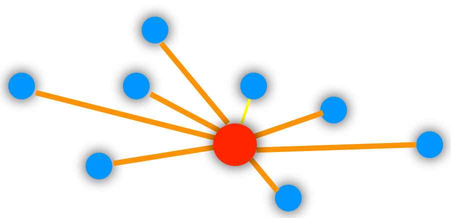
# Scalability Solution for Sparse Data
# Machine Learning for LArTPC Image Analysis

## Two independent solutions pursued

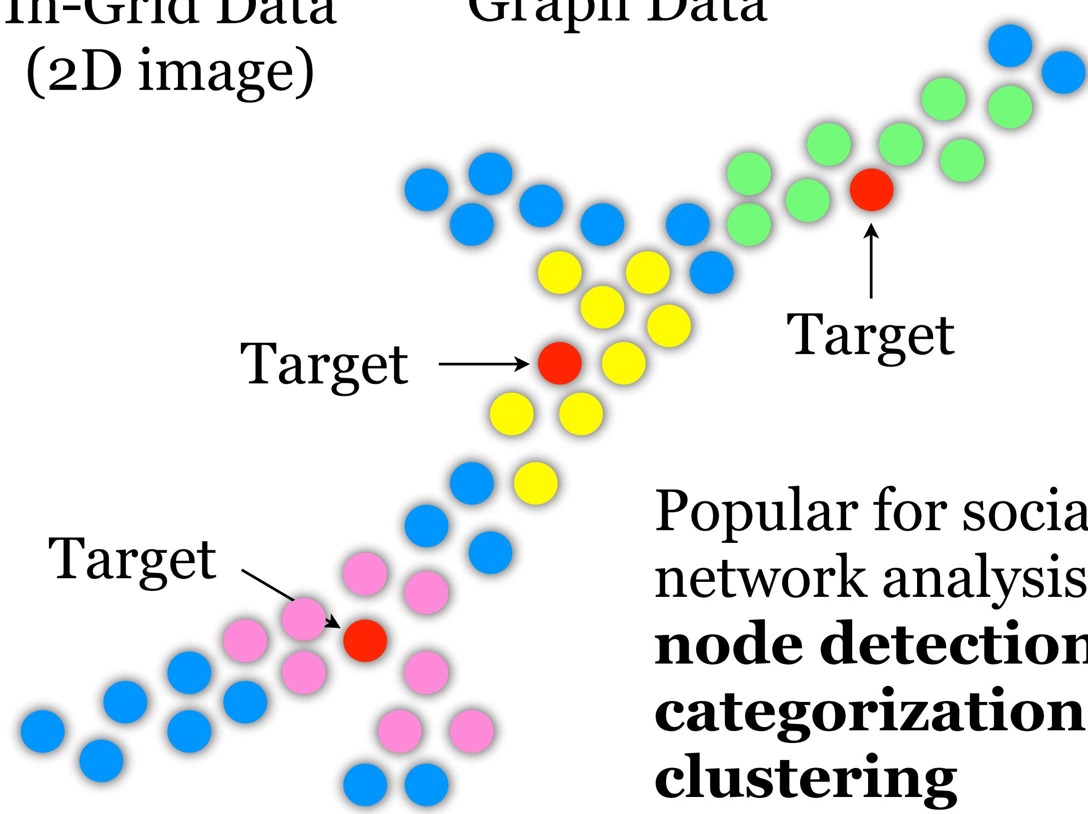
- **Sparse linear algebra** … efficient operation, ignores zero pixels (ZP)
- **Graph neural network** … efficient data representation, eliminate ZP



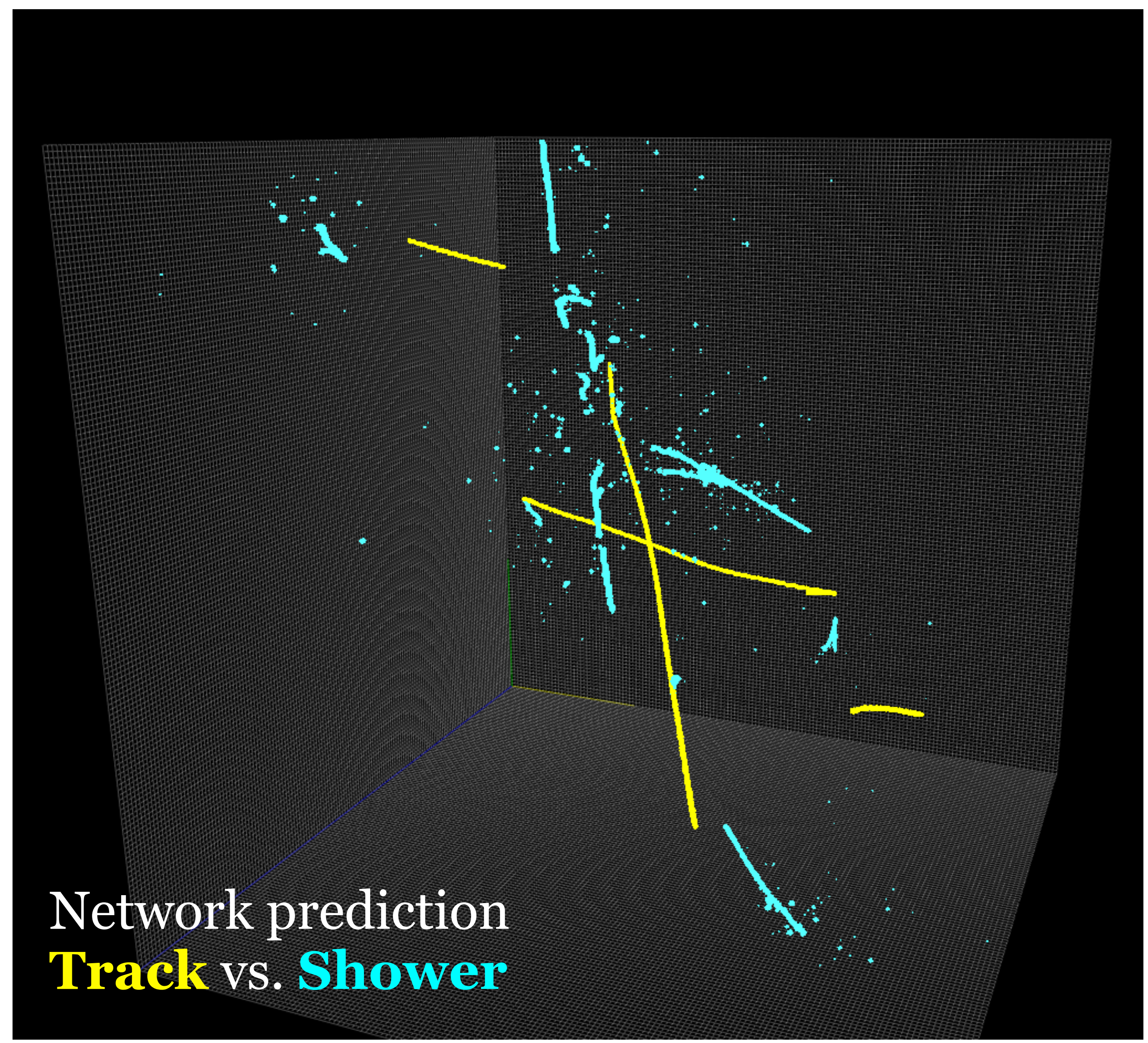
In-Grid Data
(2D image)



Graph Data




Popular for social network analysis, **node detection, categorization, clustering**



# Scalability Solution for Sparse Data
# Machine Learning for LArTPC Image Analysis

SLAC

## Two independent solutions pursued

- **Sparse linear algebra** … efficient operation, ignores zero pixels (ZP)
- **Graph neural network** … efficient data representation, eliminate ZP
- **Bottom line**: **both works great**

| | Dense U-ResNet | Sparse U-ResNet |
|---|---|---|
| Process time (forward path) | 4.5 s | 6.6 ms |
| Memory | 25.4 GB | 50 MB |
| Train time | 10 days | 15 min. |

Preliminary

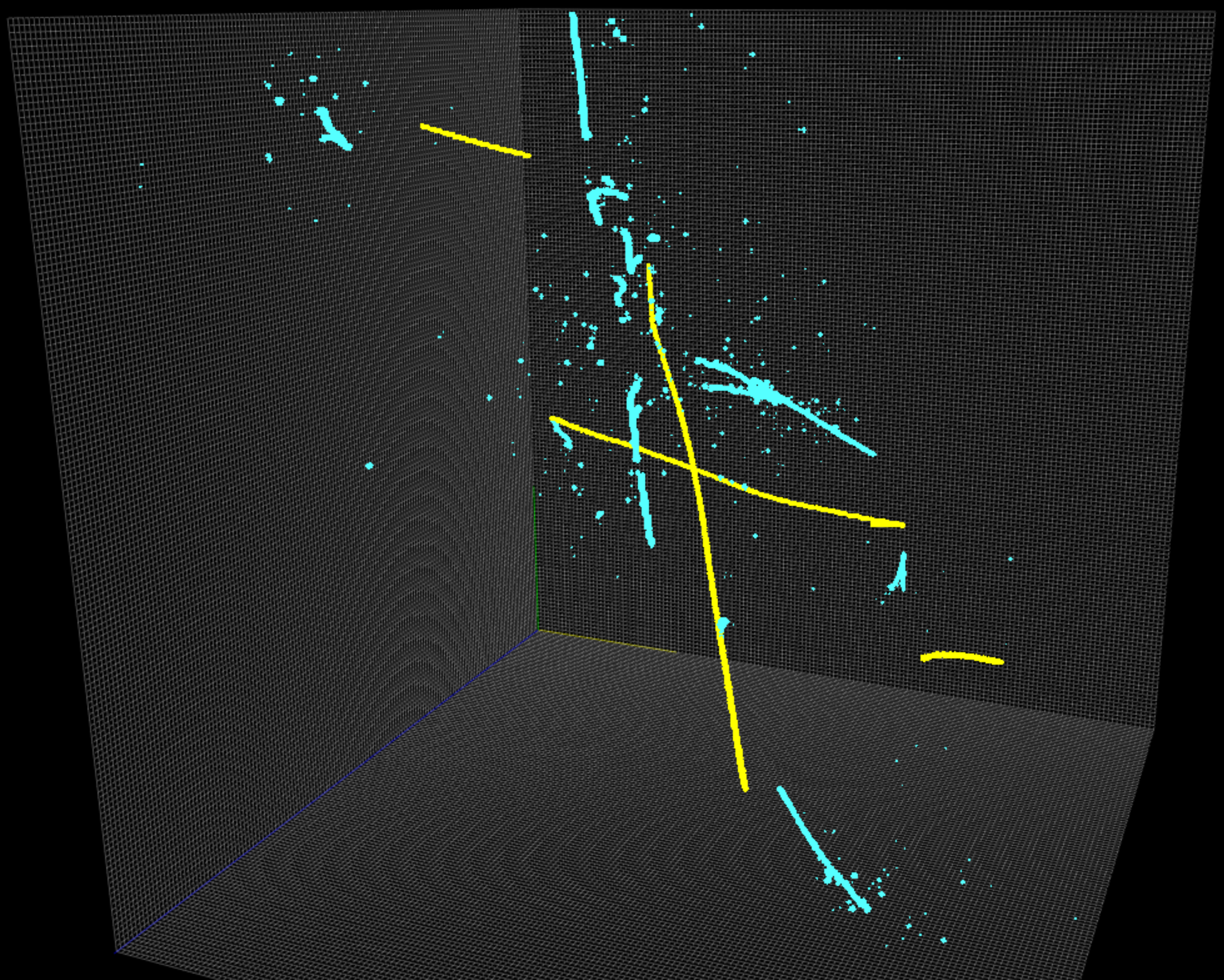
- Using 3D data with 192^3 pixels
- 2 events per GPU processing
- Trained to reach 98% accuracy in the segmentation task (defined in paper)
- 1/2 neurons compared to the published 2D U-ResNet (code)

**This is a game changer…**

Curse of dimensionality almost addressed = scalable to big data

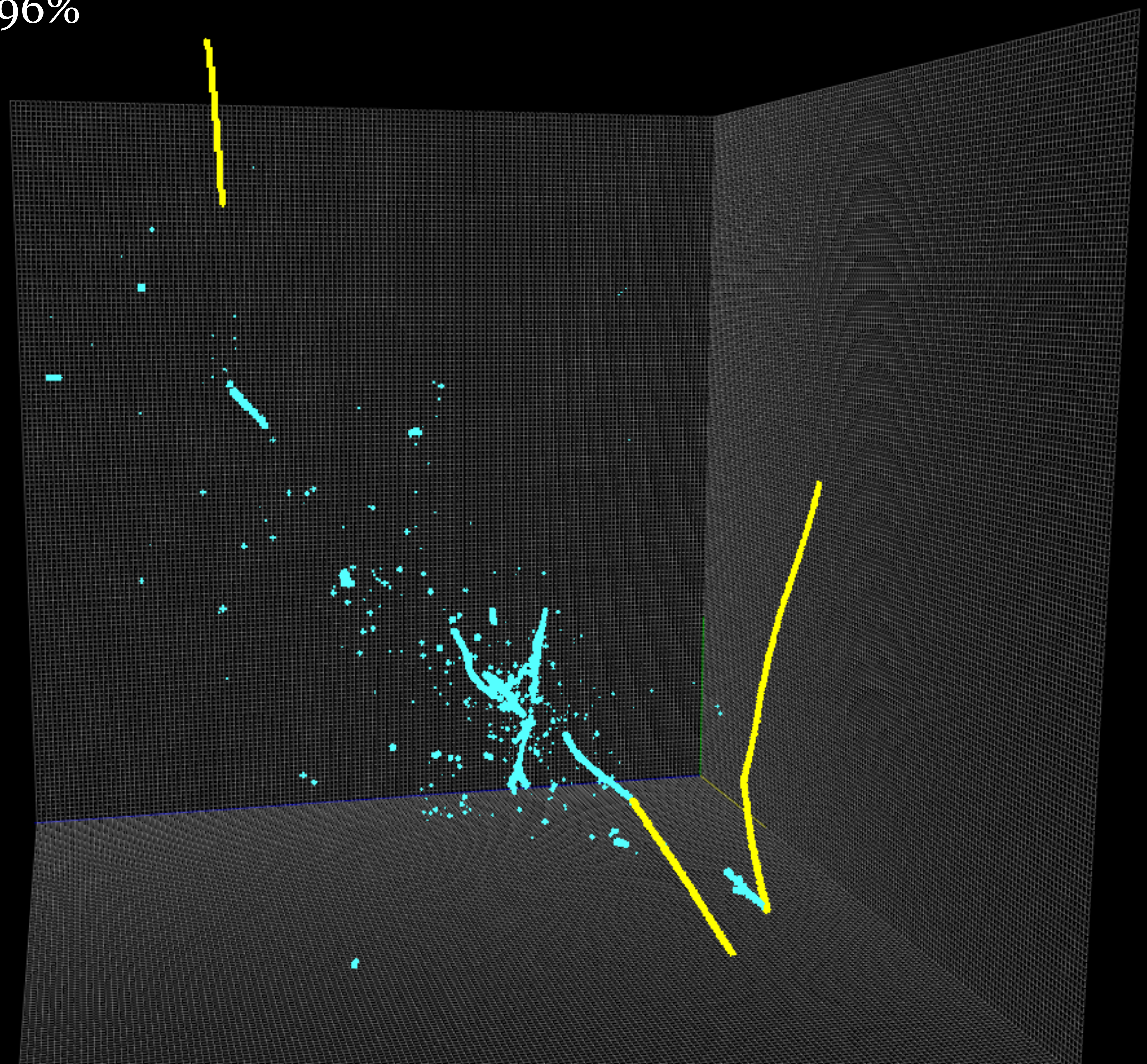
# Scalability Solution for Sparse Data
## Machine Learning for LArTPC Image Analysis

Randomly picked event
Prediction accuracy 99.93%

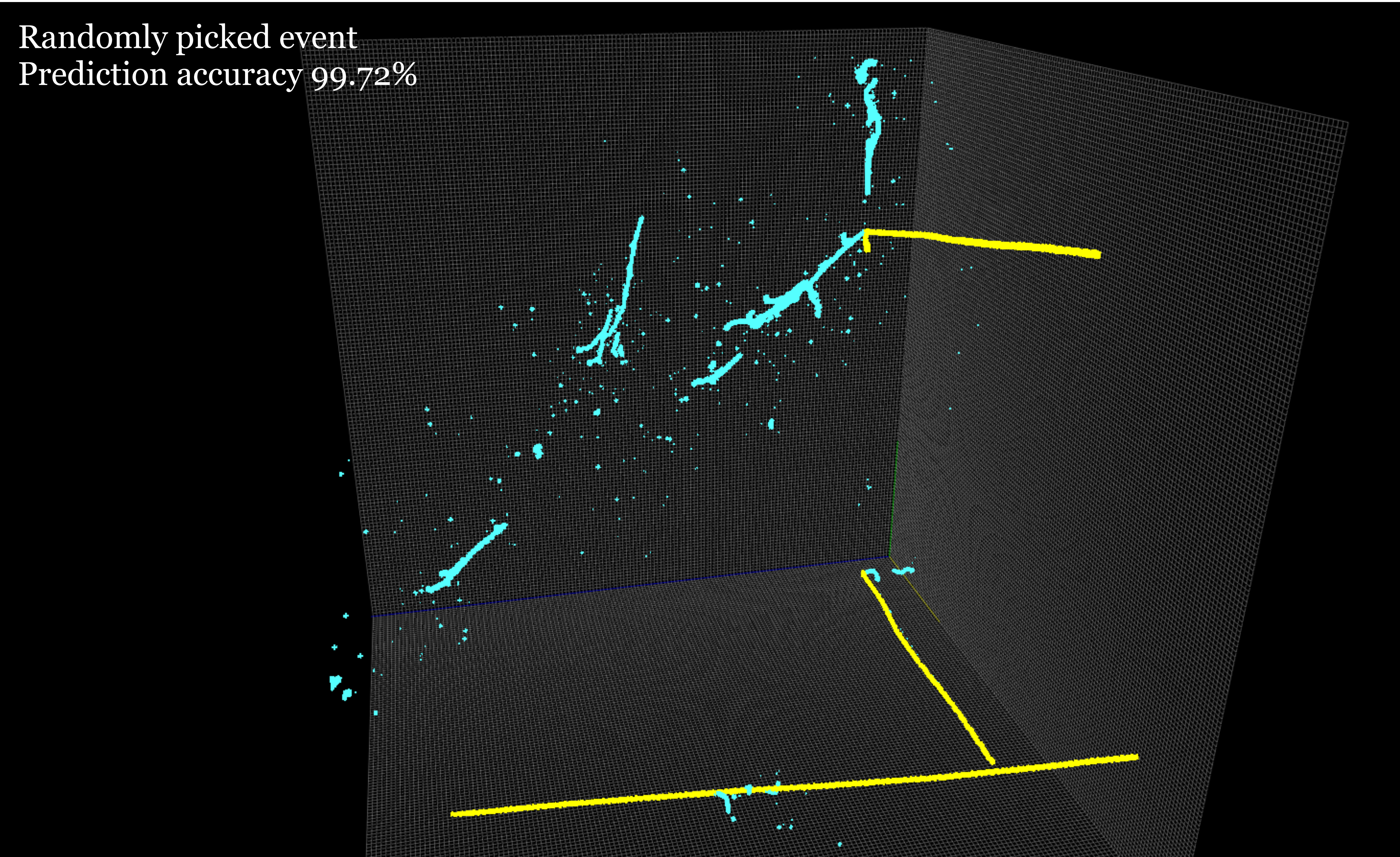


# Scalability Solution for Sparse Data
# Machine Learning for LArTPC Image Analysis

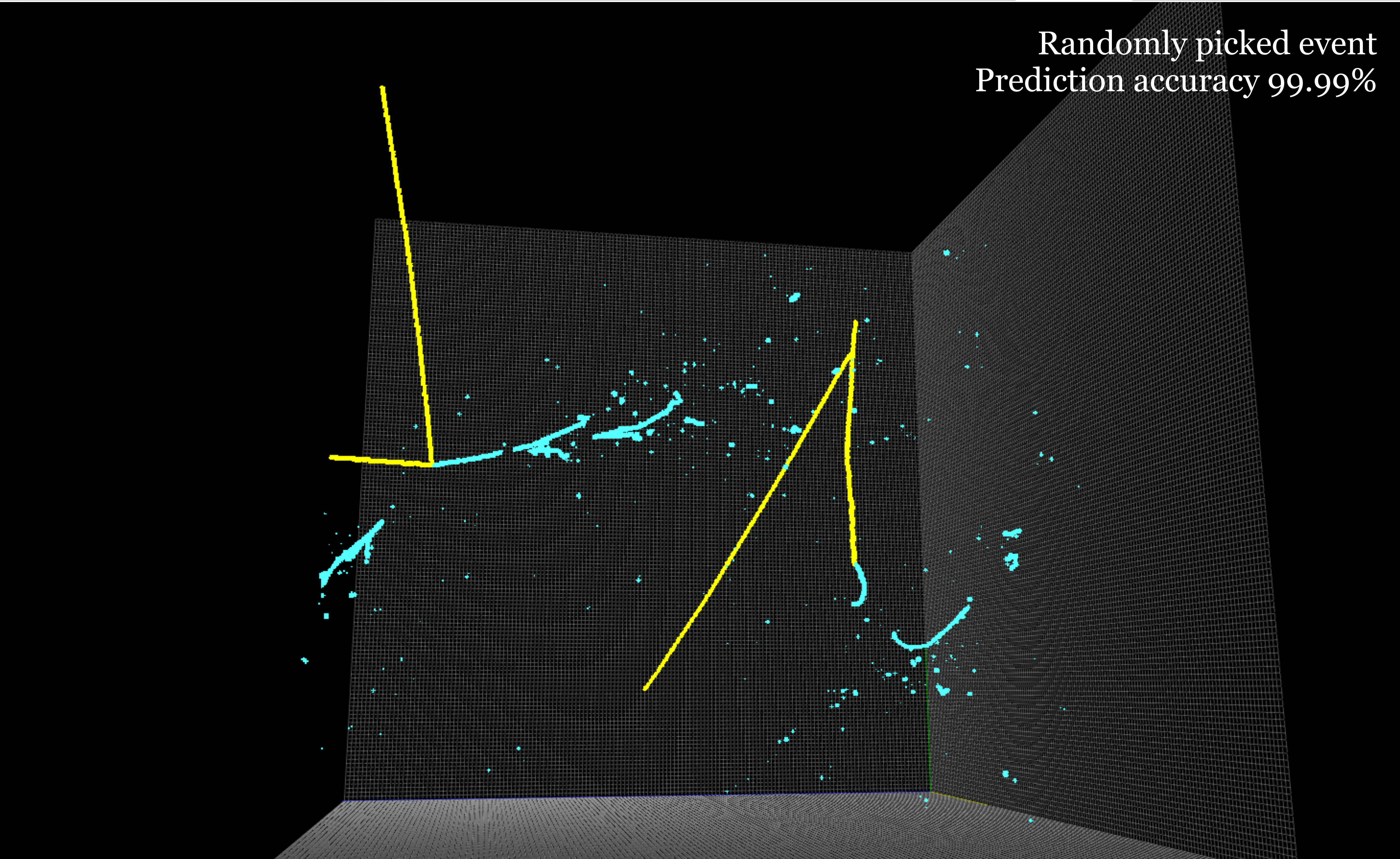
Randomly picked event
Prediction accuracy 99.96%



# Scalability Solution for Sparse Data
# Machine Learning for LArTPC Image Analysis

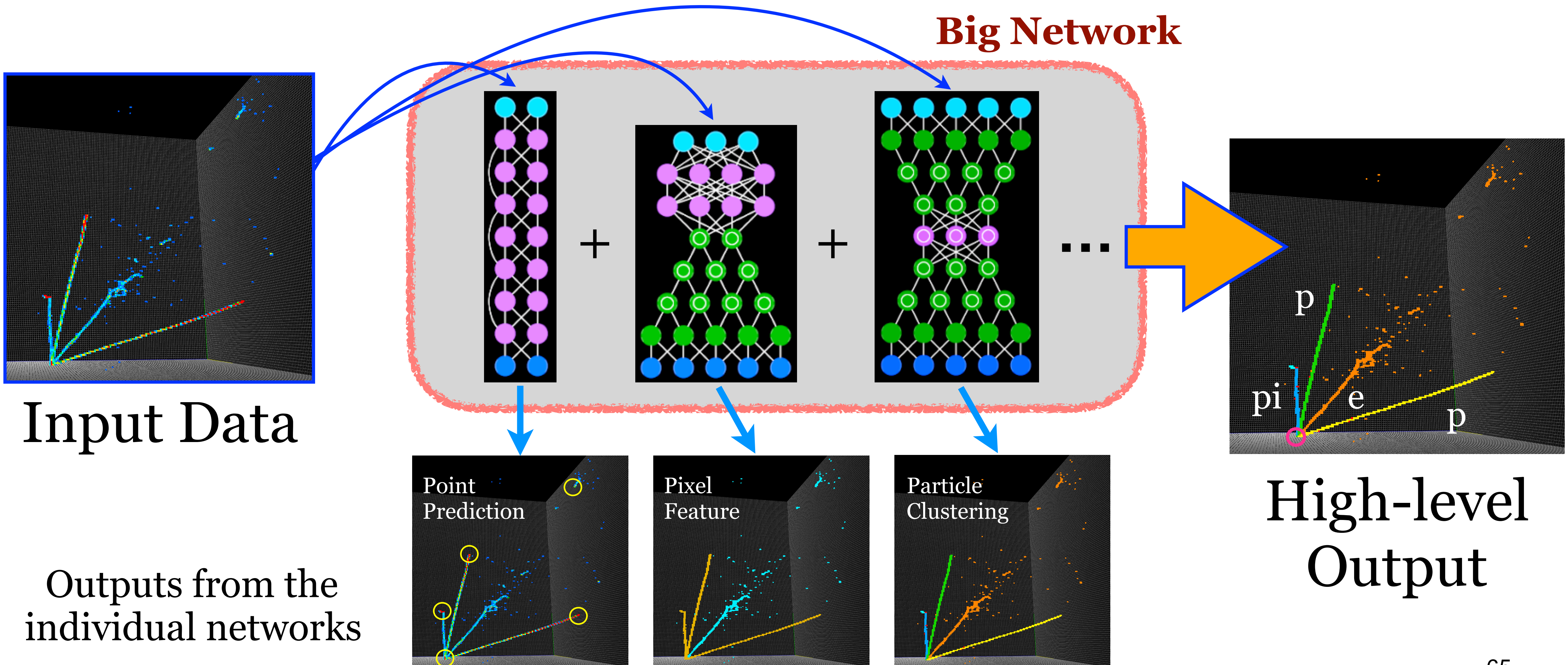


# Scalability Solution for Sparse Data
# Machine Learning for LArTPC Image Analysis



# Multi-task Deep Neural Network

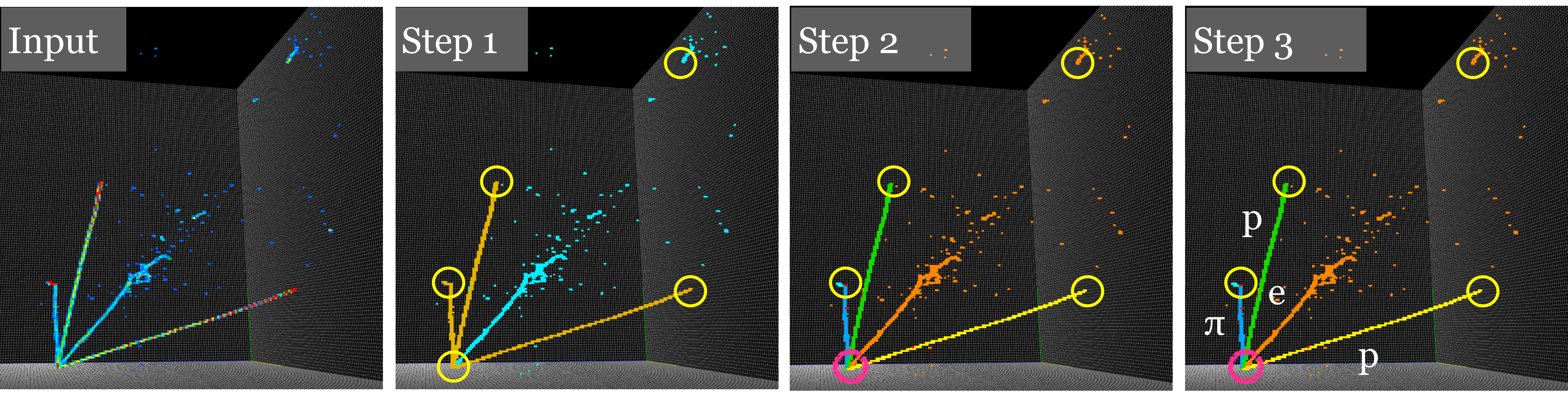
- A cluster of many task-specific networks in 2D & 3D
  - Vertex finding, clustering, particle ID, etc.



# Full Reconstruction Chain
# Machine Learning for LArTPC Image Analysis

## 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