µBooNE

Deverlopment of 3D Data Reconstruction Chain using Deep Neural Network

DUNE LArTPC Pixel Workshop

Kazuhiro Terao

SLAC National Accelerator Laboratory

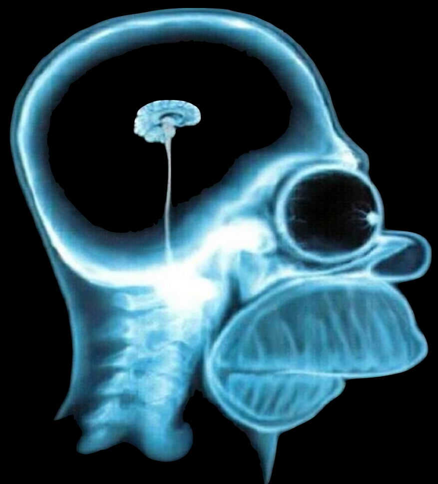
μBooNE

Deverlopment
of
3D Data Reconstruction Chain
using
Deep Neural Network

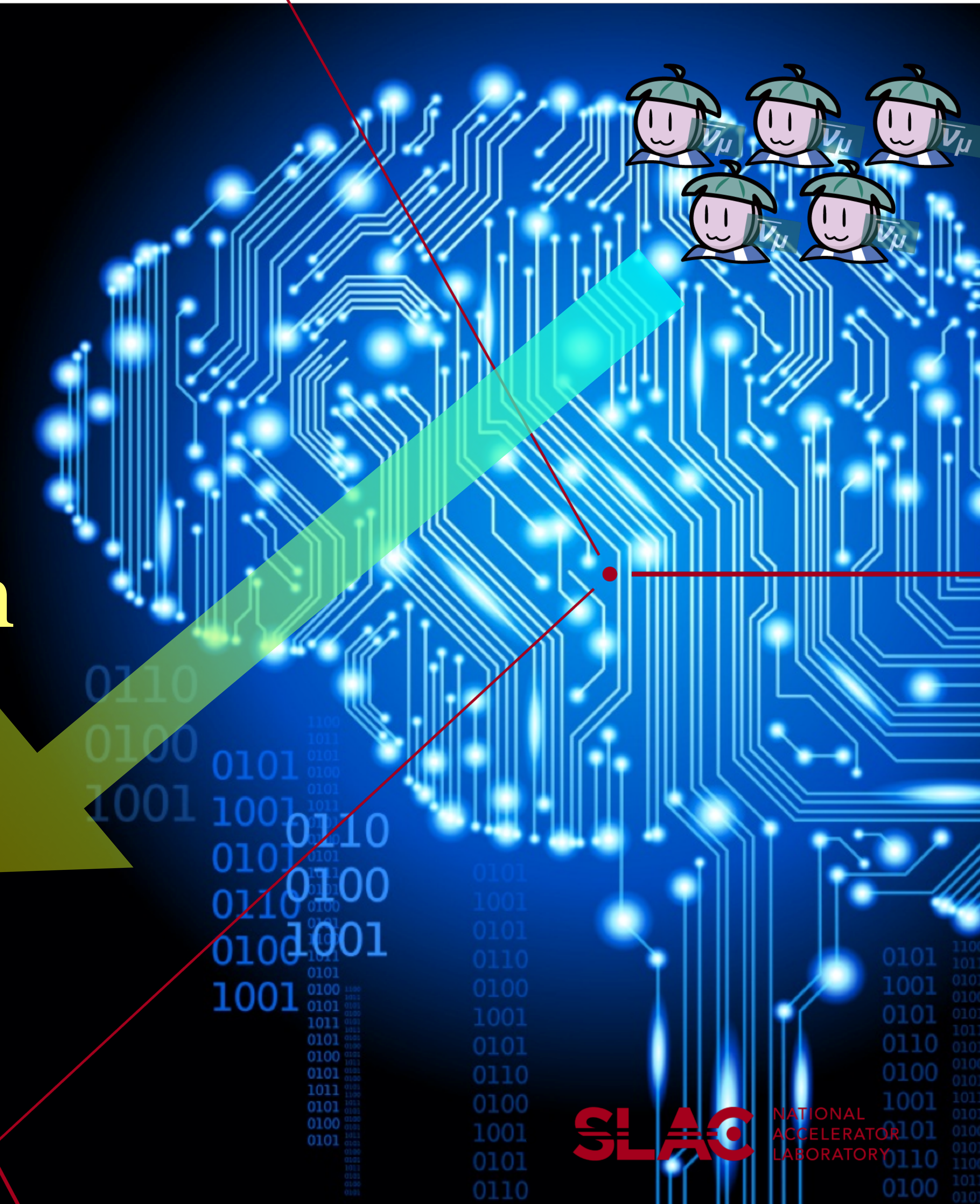
DUNE LArTPC Pixel Workshop

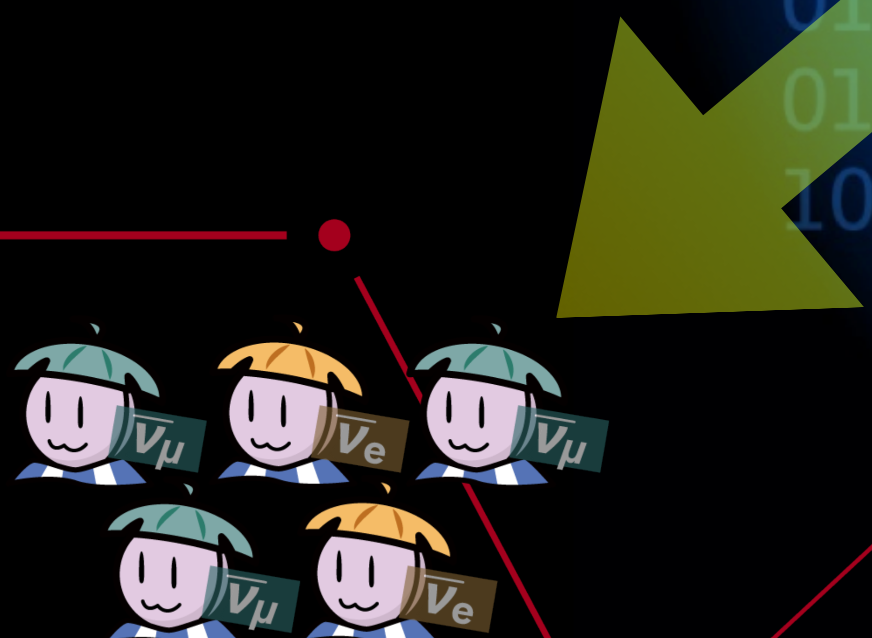
Outline

0. ML-based 3D Reconstruction
1. Progress & next steps
2. Challenges to be addressed



ML-based 3D Data Reconstruction





U.S. DEPARTMENT OF ENERGY
Office of Science

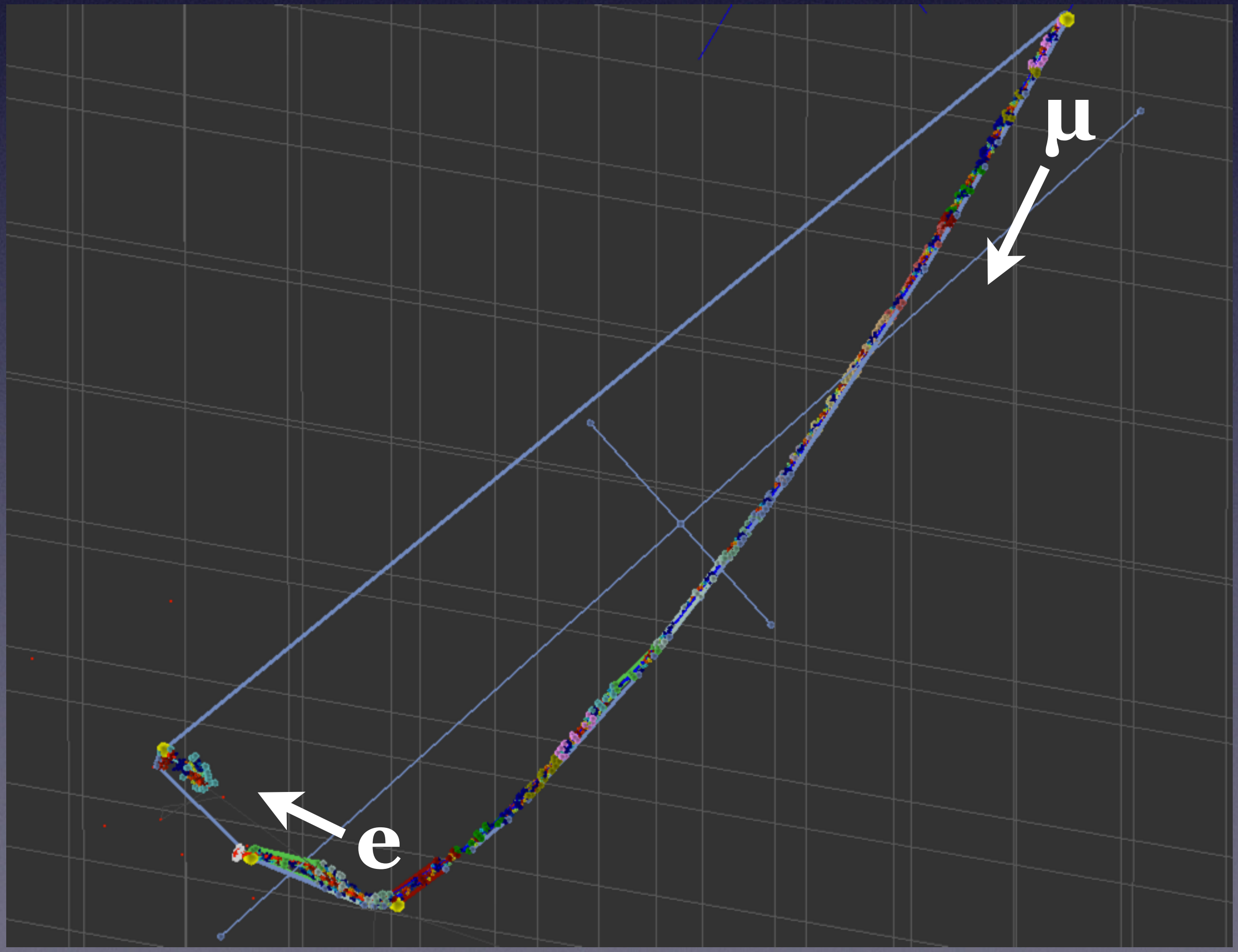
SLAC NATIONAL ACCELERATOR LABORATORY

3D Data Reconstruction @ SLAC

Our involvement: MicroBooNE/ICARUS/DUNE
Our history: long involvement in LAr reco

Tracy Usher: Cluster3D
- 3D point reconstruction
- 3D point clustering

Yun-Tse Tsai:
- 3D shower reco



... and long advocator for 3D pattern recognition, now moving into ML

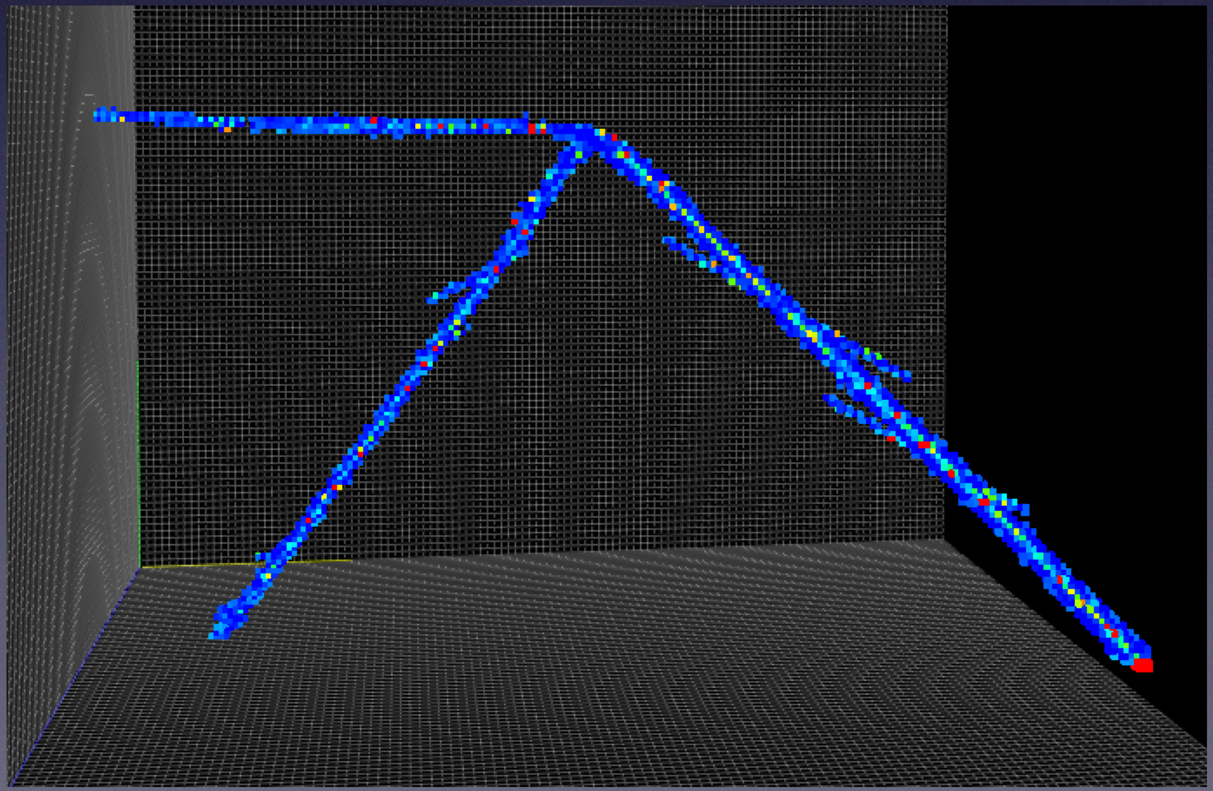
Me **(Kazu)**: 40 bounds ago
- LAr reconstruction
- Pioneered deep neural network for LAr

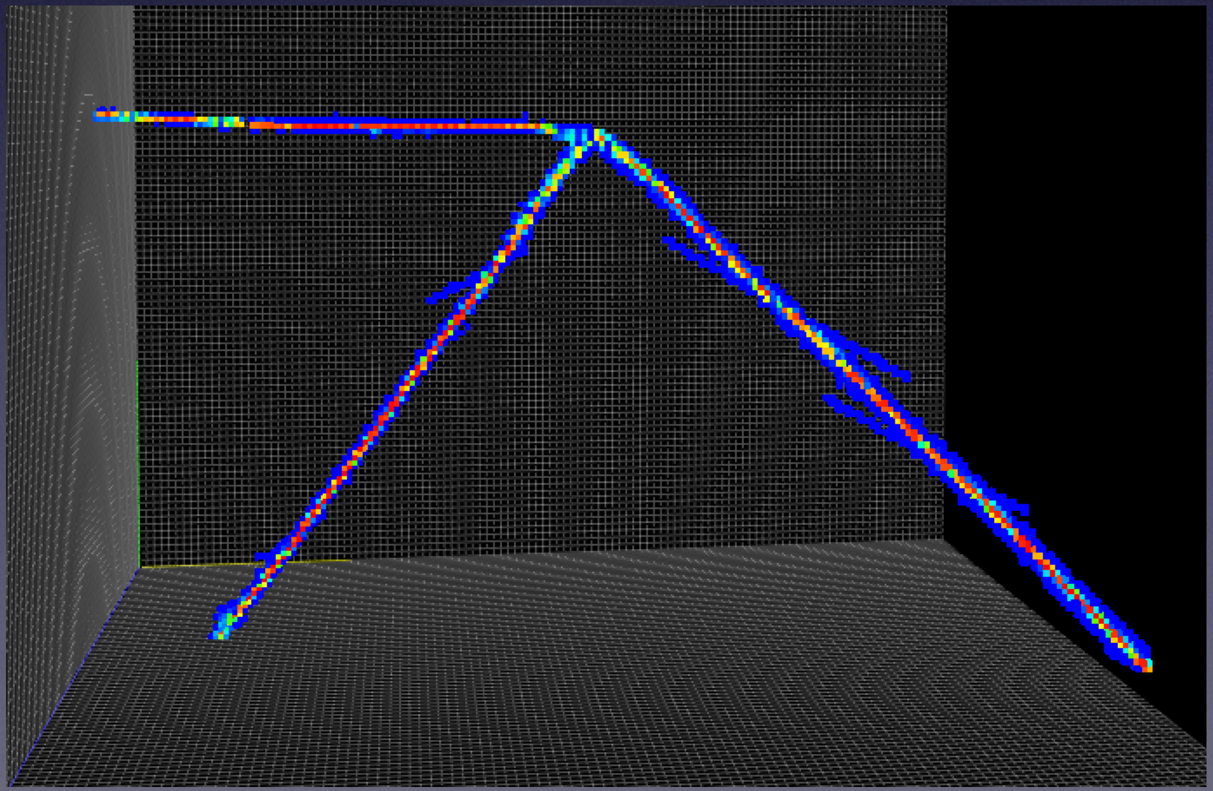
3D Data Reconstruction @ SLAC

Our involvement: MicroBooNE/ICARUS/DUNE
Our history: long involvement in LAr reco

Tracy Usher: Cluster3D
- 3D point reconstruction
- 3D point clustering

Yun-Tse Tsai:
- 3D shower reco



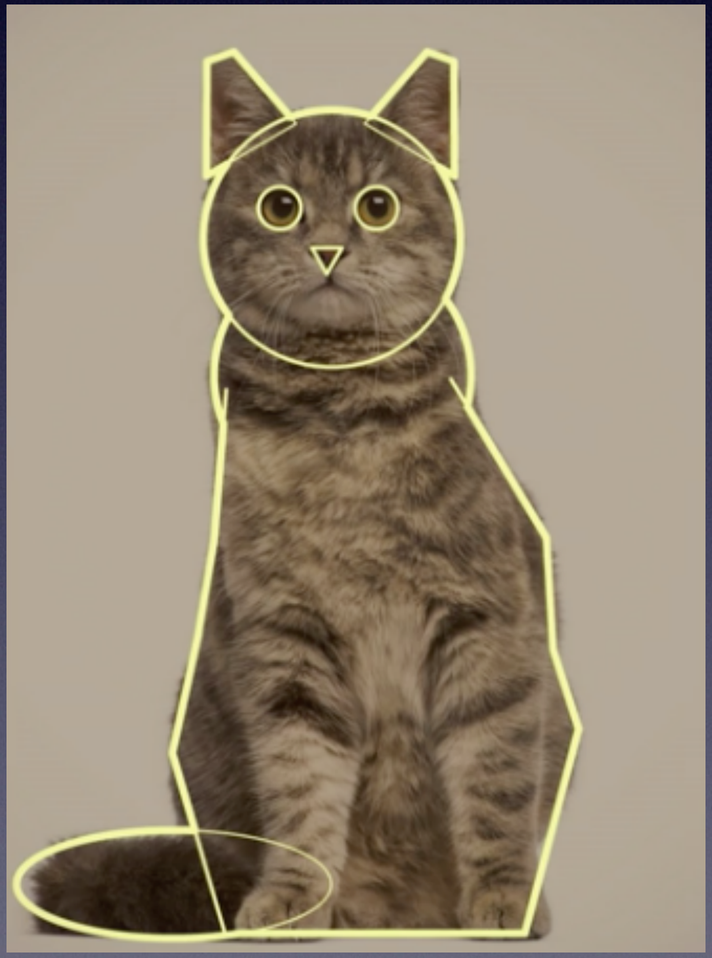


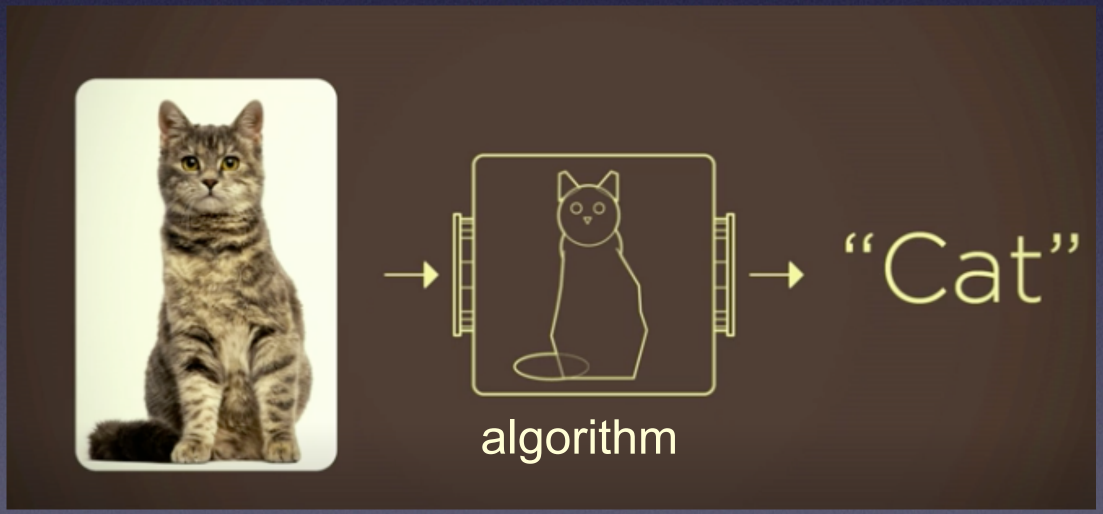
Tracy shows you can start ML above age of 60

Machine Learning & Computer Vision

Development Workflow for non-ML reconstruction

1. Write an algorithm based on physics principles





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

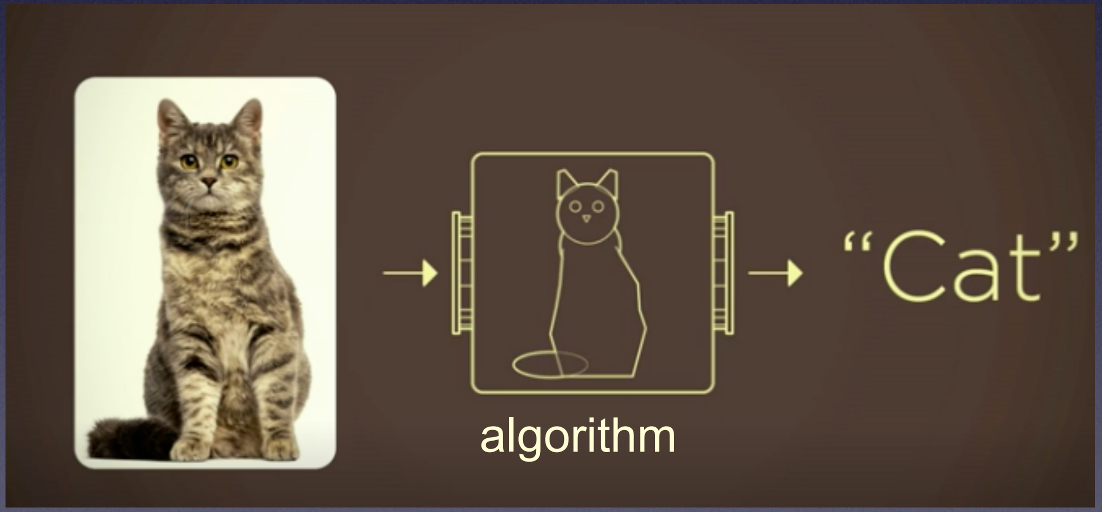
Machine Learning & Computer Vision

Development Workflow for non-ML reconstruction

1. Write an algorithm based on physics principles
2. Run on simulation and data samples
3. Observe failure cases, 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 fiducial volume)

Stretching cat
(DIS?)



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

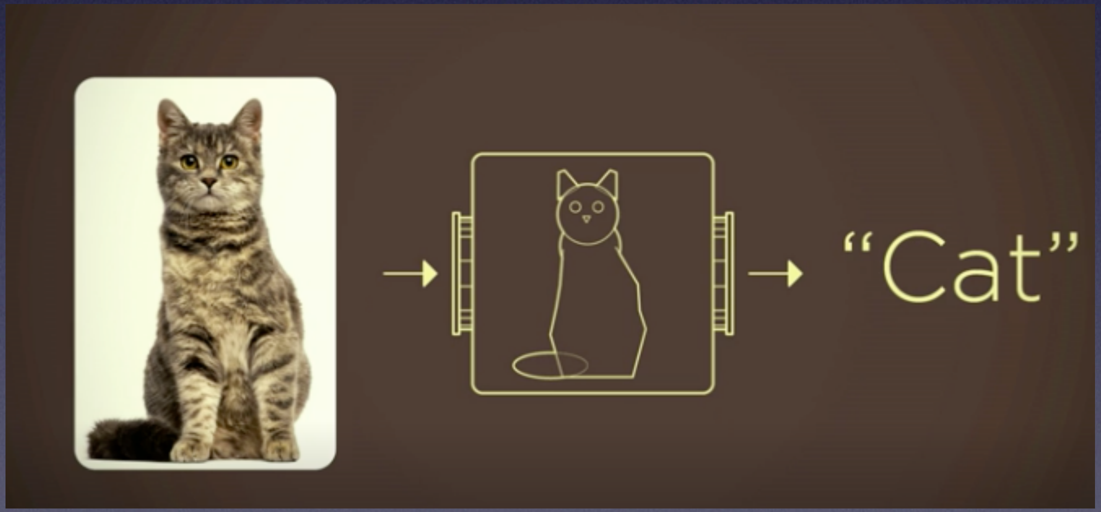
Machine Learning & Computer Vision

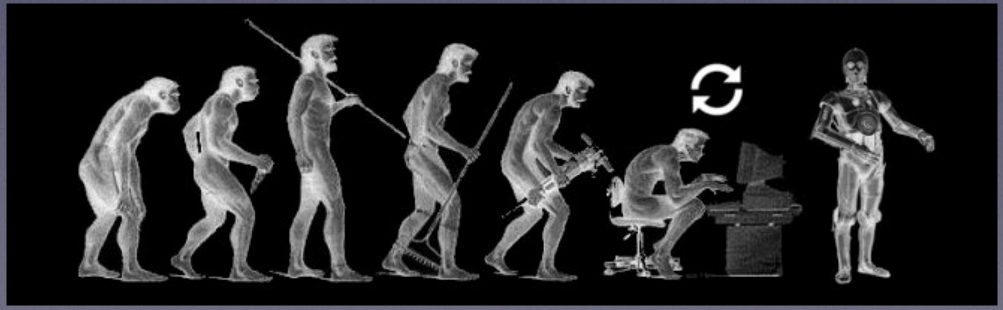
Development Workflow for non-ML reconstruction

1. Write an algorithm based on physics principles
2. Run on simulation and data samples
3. Observe failure cases, 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.

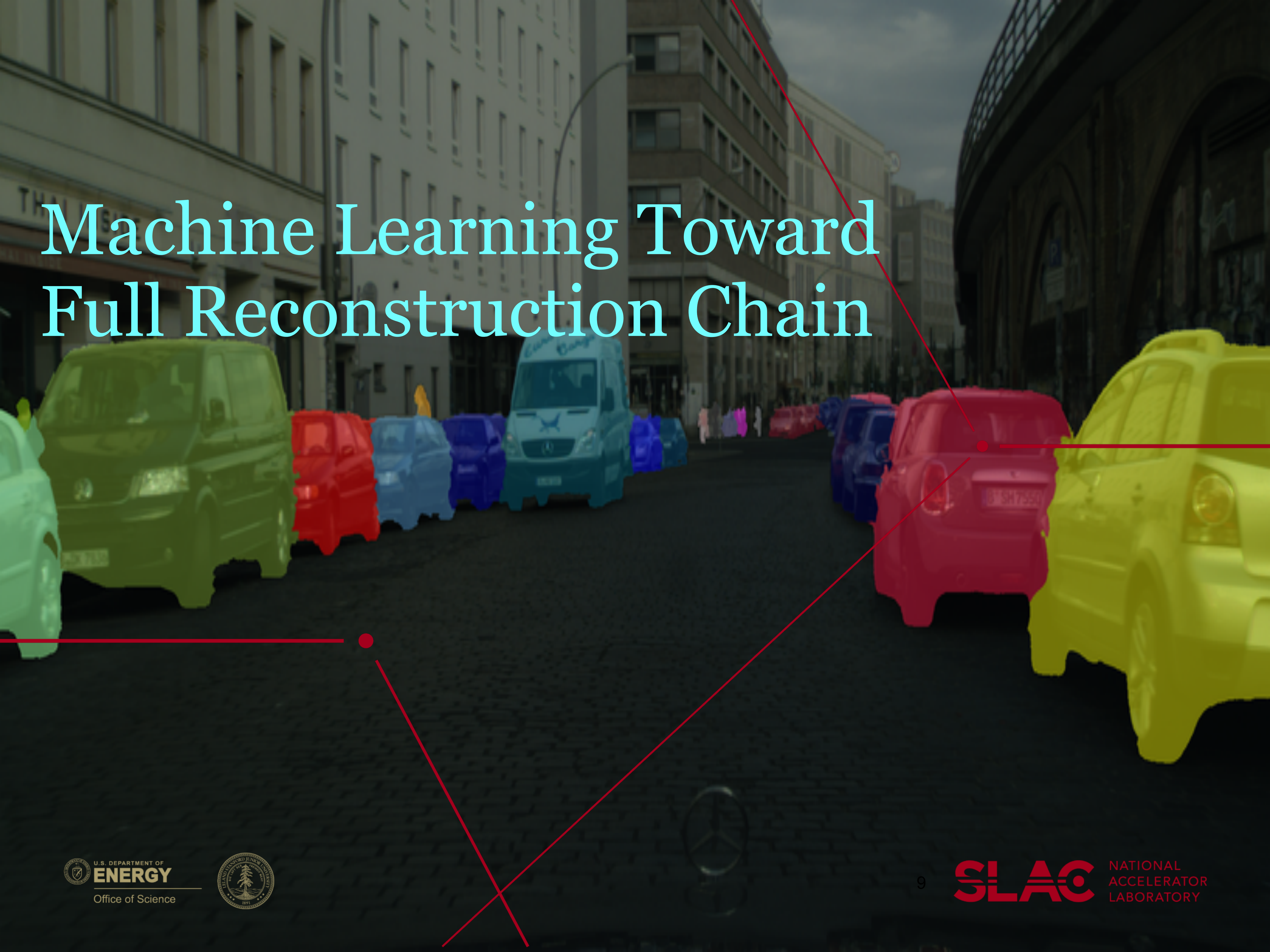
Machine Learning

- **"Learn patterns from data"**
  - automation of steps 2, 3, and 4
- **"Chain algorithms & optimize"**
  - step 5 addressed by design
- **"Deep Learning"**
  - ML algorithms using deep neural networks
  - now applying to LArTPC data analysis





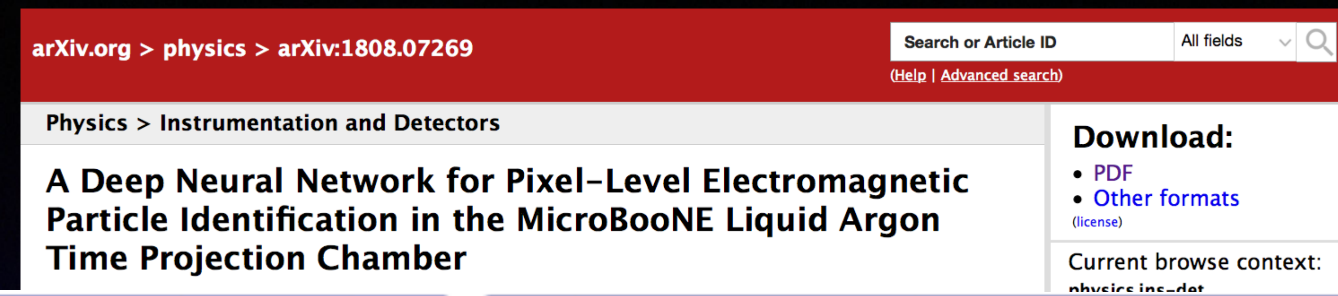
Machine Learning Toward Full Reconstruction Chain

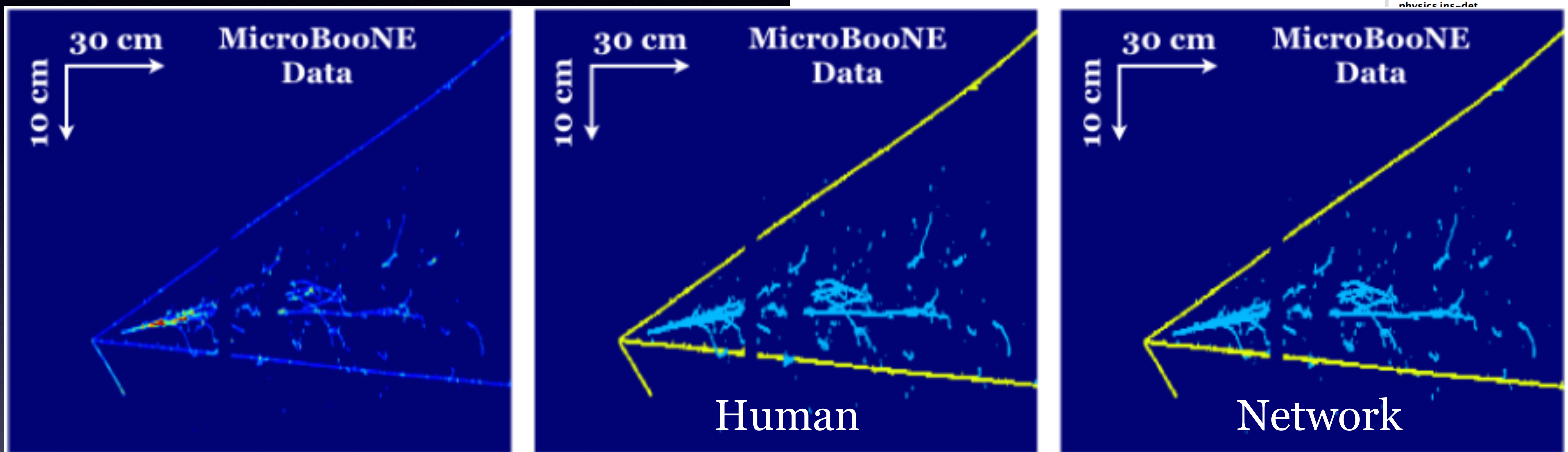


Machine Learning & Computer Vision

Demonstrations for LArTPC

arXiv:1611.05531, arXiv:1808.07269
(MicroBooNE)

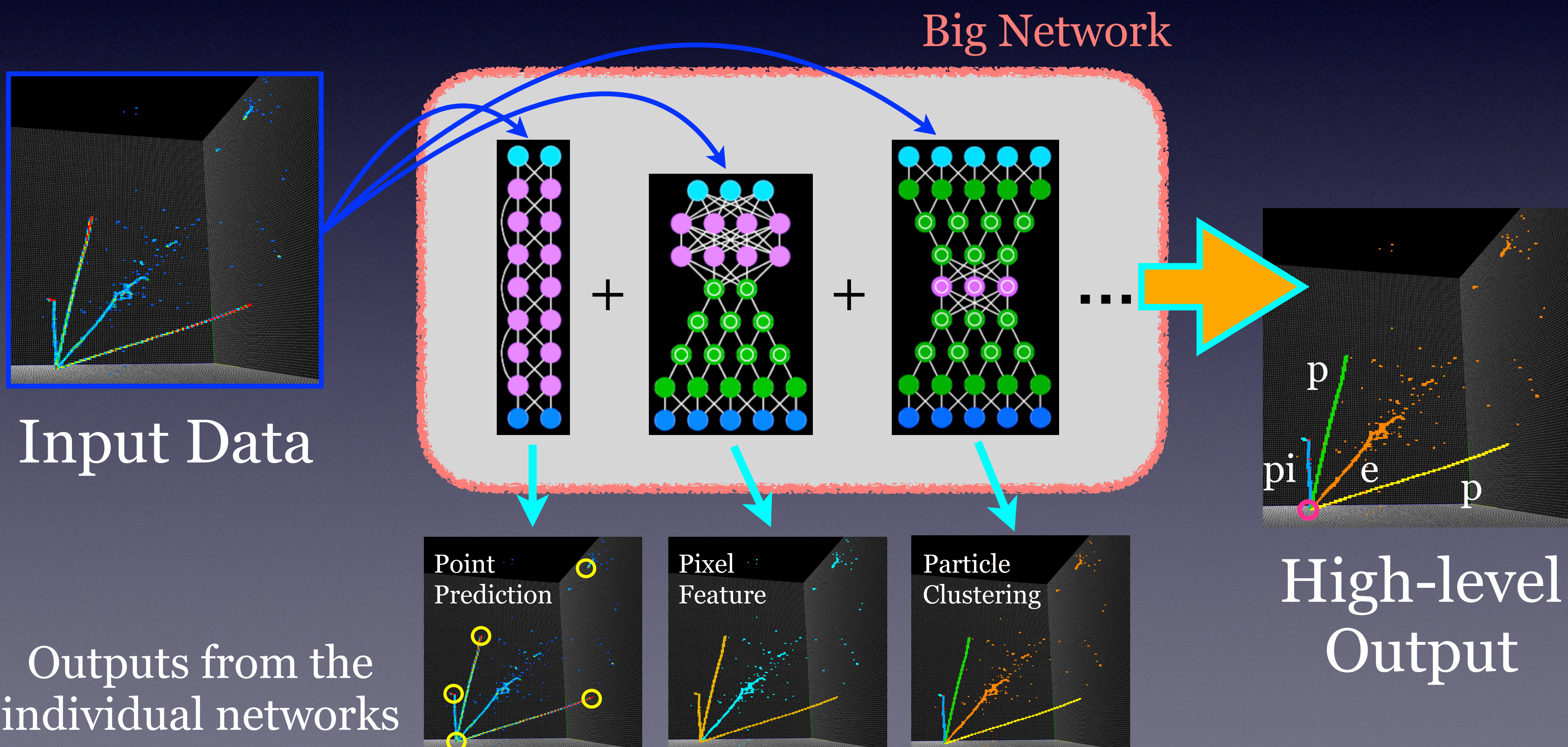




ML-based Data Reconstruction

Multi-task Deep Neural Network

- A cluster of many task-specific networks
  - Vertex finding, clustering, particle ID, etc.
  - The big network takes all informations used by individual network for a high level physics analysis task



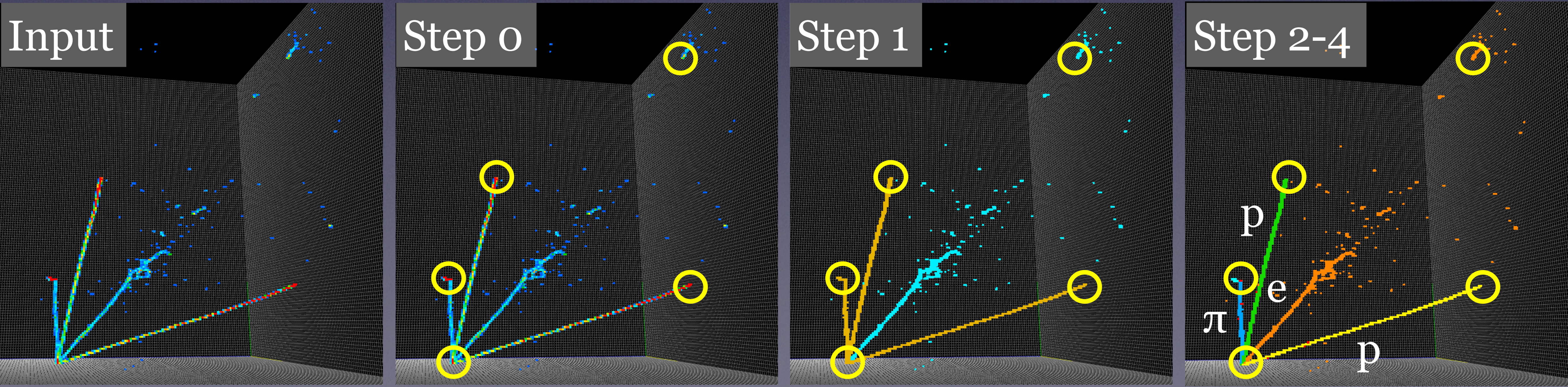
ML-based Data Reconstruction

Multi-task Deep Neural Network

- A cluster of many task-specific networks
  - Vertex finding, clustering, particle ID, etc.
  - The big network takes all informations used by individual network for a high level physics analysis task

☑ **0**. Feature space point (track edges, shower start)
☑ **1**. Basic topological classification
☐ **2**. Vertex finding
☐ **3**. Particle-wise clustering + type identification
☐ **4**. Hierarchy building

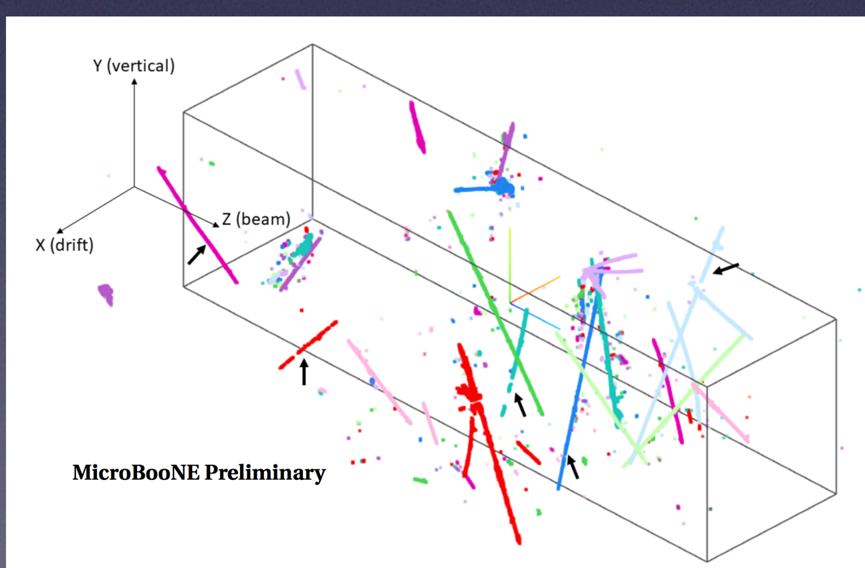
Early Career Award Program



ML-based Data Reconstruction

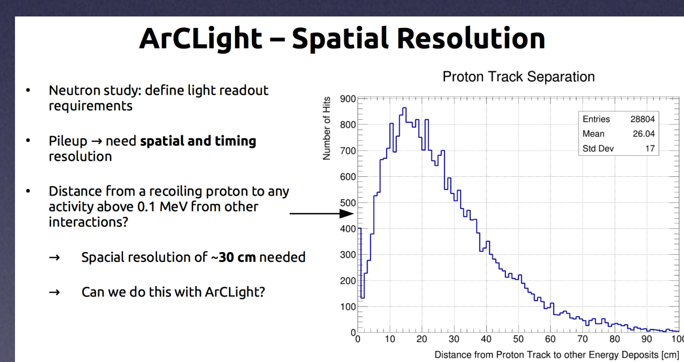
Synergy & Collaboration w/ other efforts

- **3D pattern recognition for wire LArTPCs**
  - **BNL-SLAC** for applying to WireCell/Cluster3D
- **3D trajectory fitting, calorimetry** (post-clustering)
  - Tools for track & shower reconstruction are wanted!
- **Physics analysis**
  - Policy on 3D data representations (w/ LArSoft, on-going)

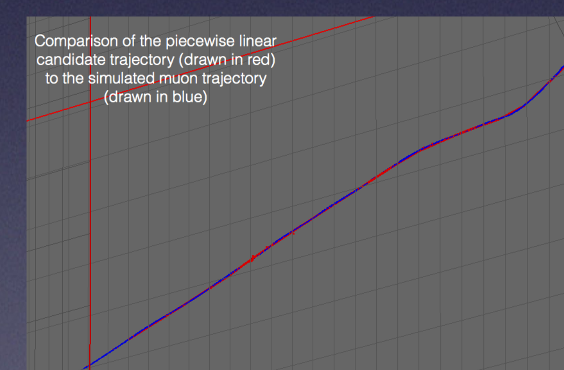


BNL (WireCell) Interaction Clustering

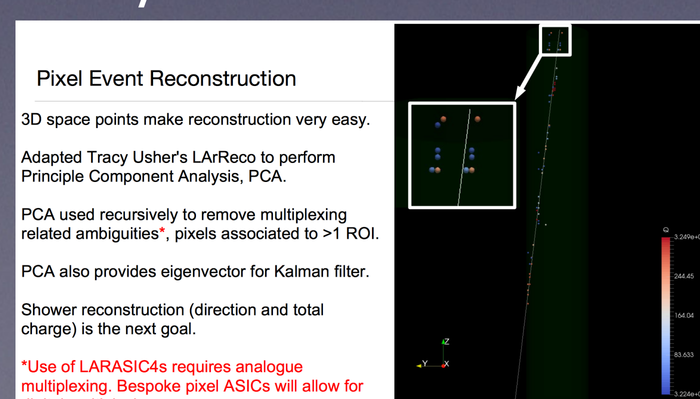
Bern ArCLight Analysis



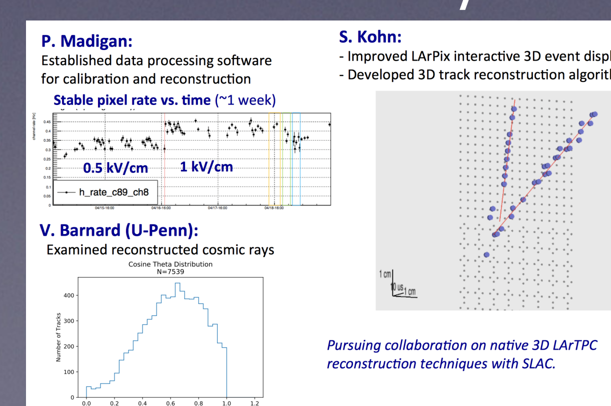
SLAC 3D Clustering



UTA/Bern PixLAr reco



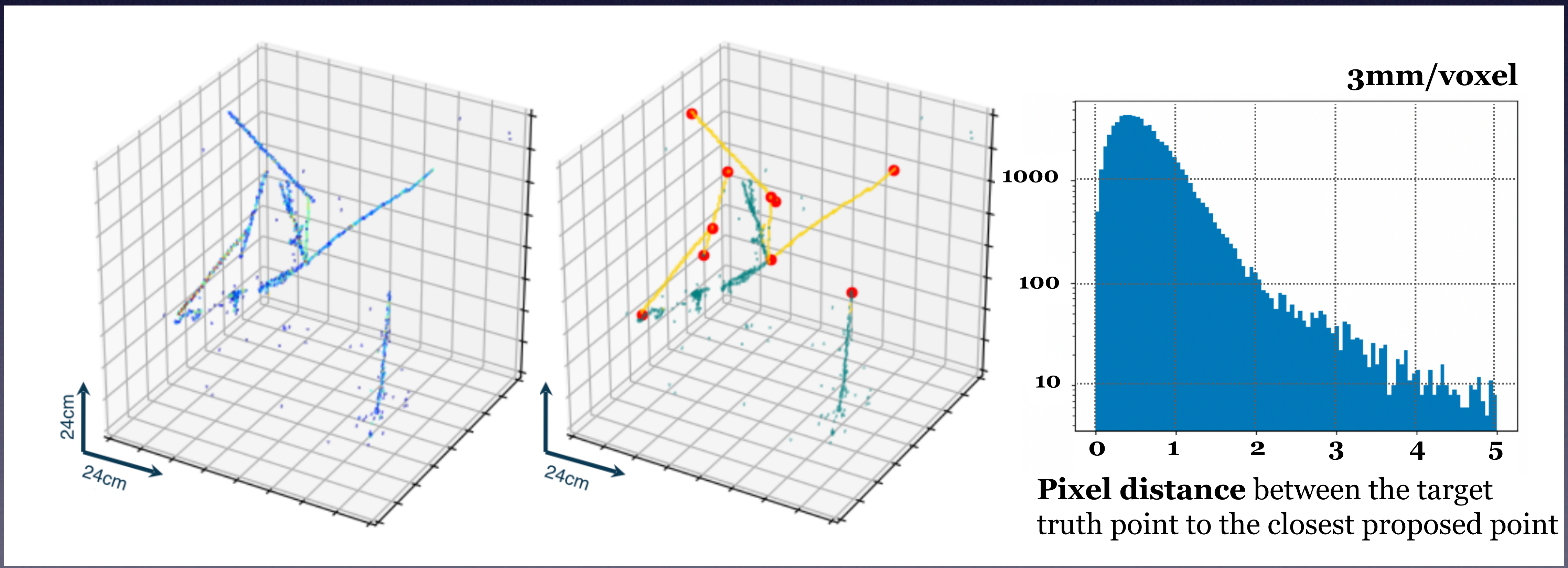
LBNL LArPix reco/calib/ana



Next Steps Toward Analysis

Analysis with currently existing tools

- **Vertex finding** + **particle multiplicity counting**
  - Pile-up disambiguation
- Preliminary **particle clustering + energy** reco
  - Range-based for tracks after fitting a trajectory
  - Calorimetry for showers after clustering



Likely ~3 physicists work for a few months. Discussion to initiate with **LBNL**. Minimal software overburden (can be all Python or C++ based, container for reproducibility).

Next Steps

Next Steps in Reco Development

Scaling for big data (… still planning only)

- Discussions with Gabe Perdue (Fermilab) to leverage Summit @ ORNL (GPU-based HPC), with Eric Church (PNNL) for compute distribution framework.
- Marcel/Zelimir (ANL) offer development for KNL-based HPC, possibly ideal for sparse data, sharable with NERSC
- Possible collaboration w/ Stanford CS + NVIDIA

Next Steps in Reco Development

Scaling for big data (… still planning only)

- Discussions with Gabe Perdue (Fermilab) to leverage Summit @ ORNL (GPU-based HPC), with Eric Church (PNNL) for compute distribution framework.
- Marcel/Zelimir (ANL) offer development for KNL-based HPC, possibly ideal for sparse data, sharable with NERSC
- Possible collaboration w/ Stanford CS + NVIDIA

Sparse data vs. Computing scalability

- Traditional com. vision ML = dense matrix linear algebra
- LArTPC data is extremely sparse = super inefficient

Next Steps in Reco Development

Scaling for big data (… still planning only)

- Discussions with Gabe Perdue (Fermilab) to leverage Summit @ ORNL (GPU-based HPC), with Eric Church (PNNL) for compute distribution framework.
- Marcel/Zelimir (ANL) offer development for KNL-based HPC, possibly ideal for sparse data, sharable with NERSC
- Possible collaboration w/ Stanford CS + NVIDIA

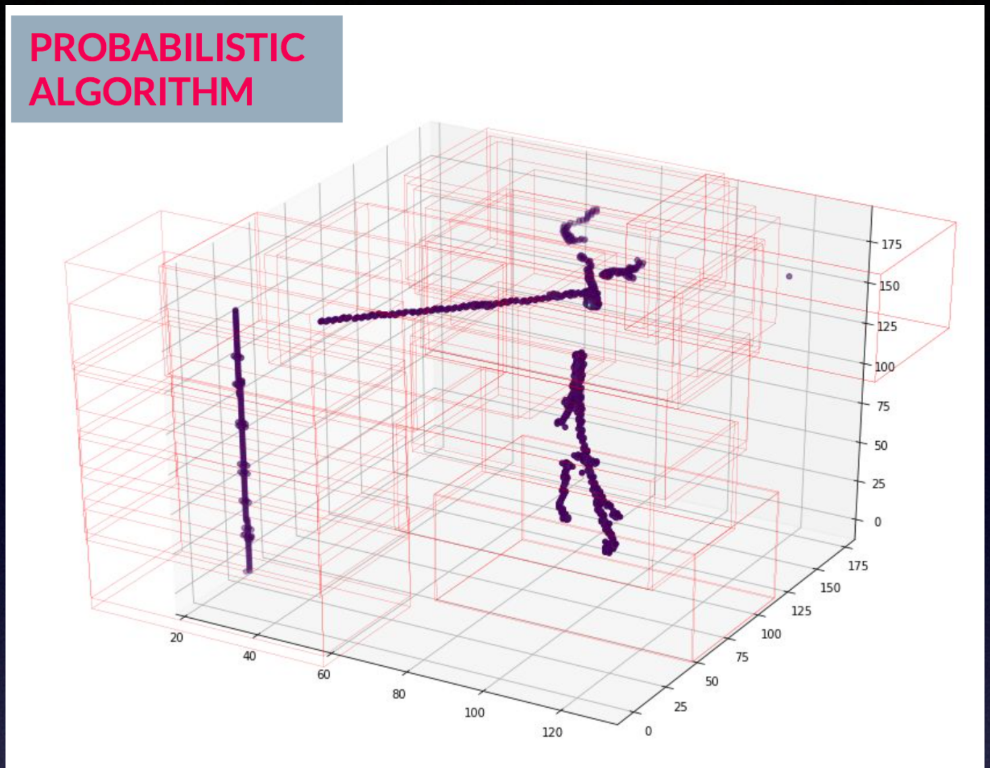
Sparse data vs. Computing scalability

- Traditional com. vision ML = dense matrix linear algebra
- LArTPC data is extremely sparse = super inefficient

Recent Progress & Plan

- More efficient method with dense matrix: ROI cropping
- Implementation of linear algebra for sparse matrix
- ML/CV techniques beyond in-grid (and sparse) data

Progress in Analyzing Sparse Data



ROI Cropping Technique

- Mitigation, not a solution
- 1/2 data reduction for 192^3 sample with 64^3 box crop
  - Speed up by ~x5 in algorithm training with NVIDIA V100, no performance loss
- Implemented in GPU kernel ops, now testing (expect another ~x5 speedup)

GPU hackathon @ BNL

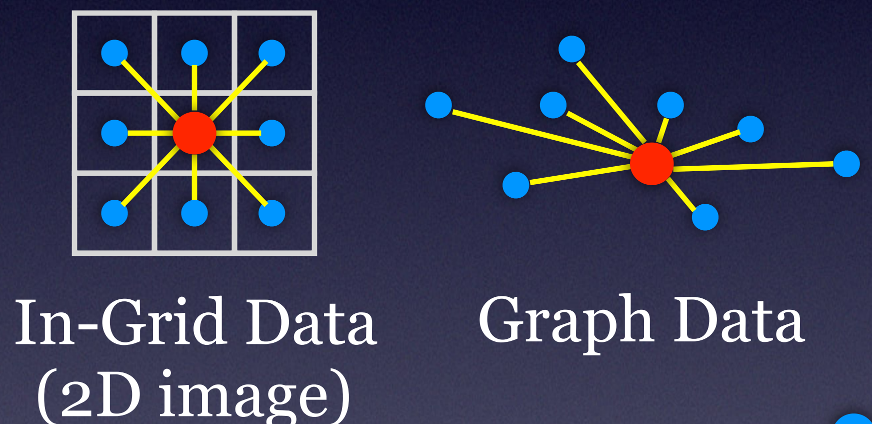
Sparse Linear Algebra

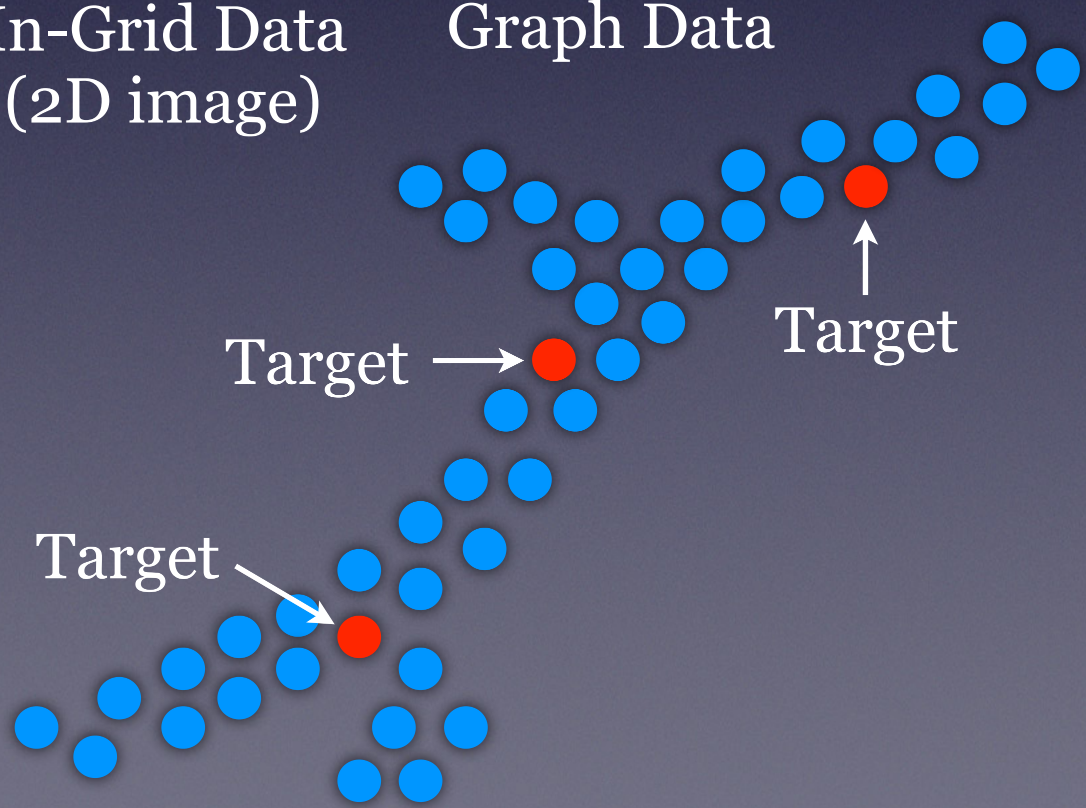
- GPU kernel ops (NVIDIA)
  - Started implementation & testing with NVIDIA experts, follow up in ~6 months
- Other venues?
  - Sparse matrix not optimal for GPUs, possibility for others such as many-core CPUs, etc? Need real expertise in distributed computing

Progress in Analyzing Sparse Data

ML in Computer Vision Beyond in-Grid Data

- **Graph Convolutional Neural Network** (GCNN)
  - Developed for social network analysis, treats data points as graph node and apply "convolution" analog operation
  - Computer vision application with **point cloud**
  - **Good for clustering, point (node) detection** (social media!)

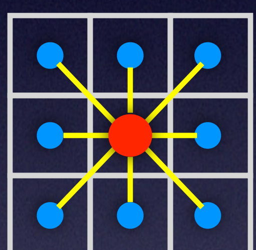




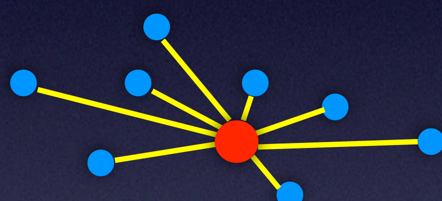
Progress in Analyzing Sparse Data

ML in Computer Vision Beyond in-Grid Data

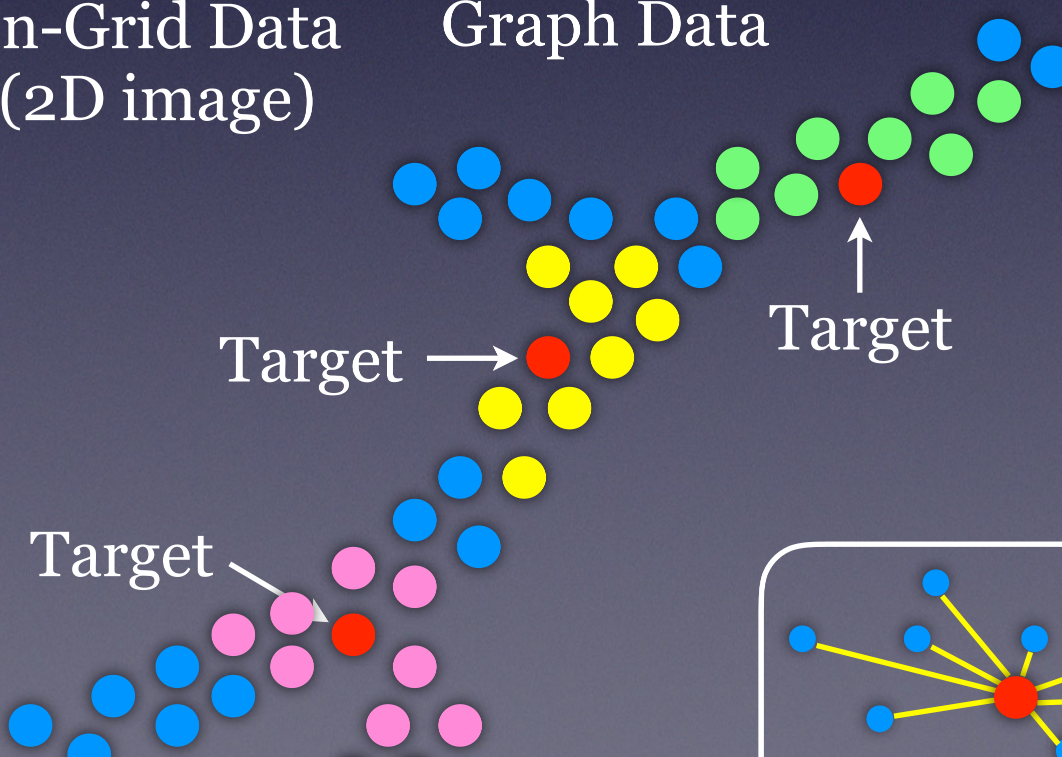
- **Graph Convolutional Neural Network** (GCNN)
  - Developed for social network analysis, treats data points as graph node and apply “convolution” analog operation
  - Computer vision application with **point cloud**
  - **Good for clustering, point (node) detection** (social media!)

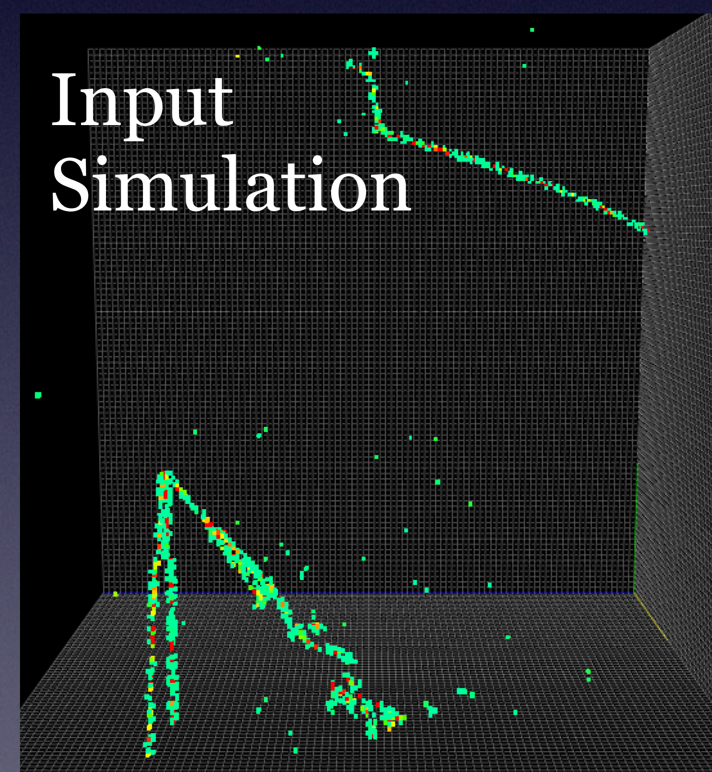


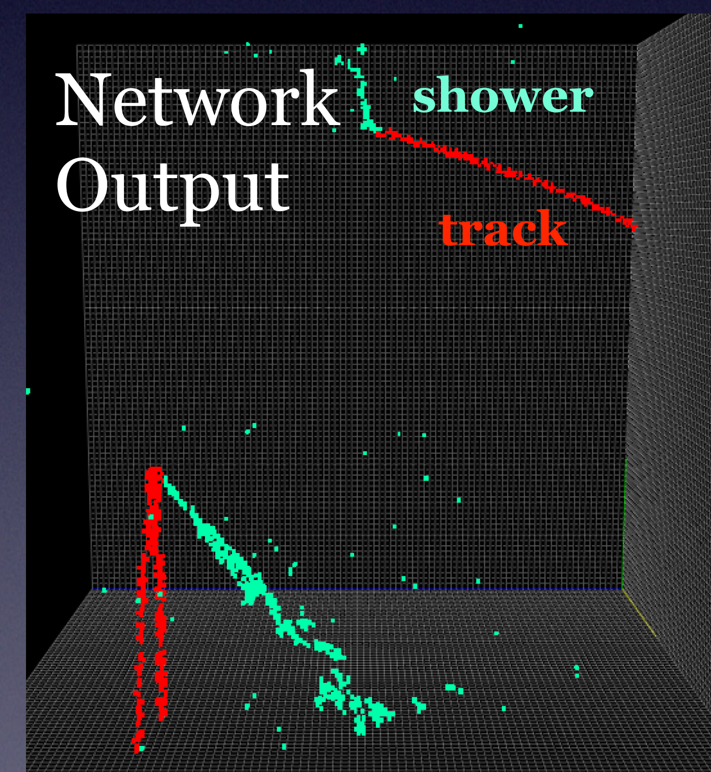
In-Grid Data (2D image)

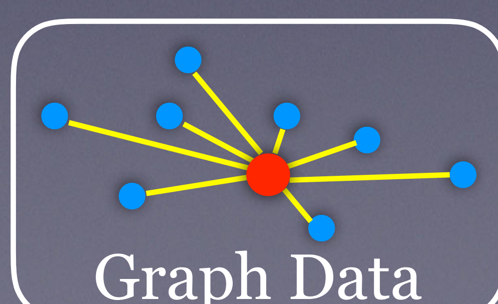


Graph Data





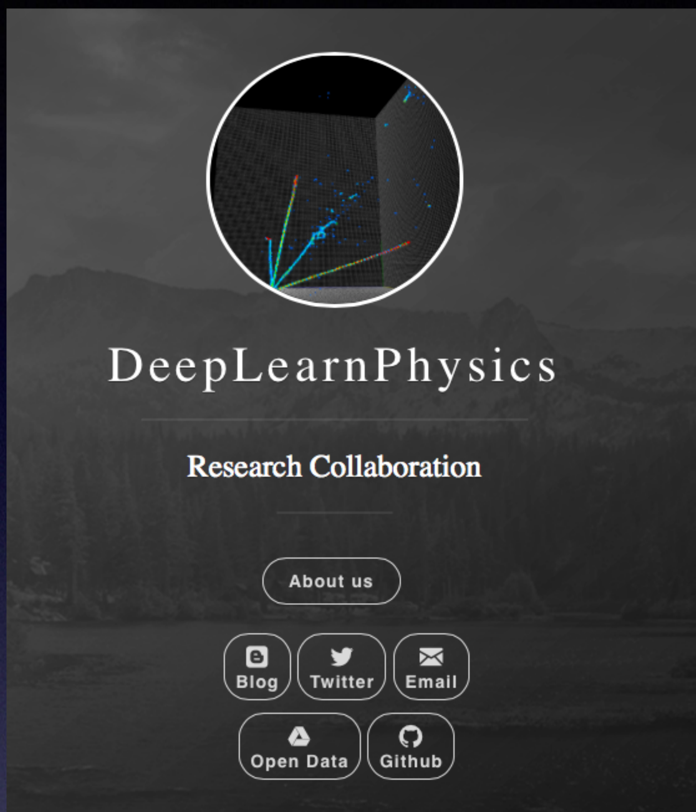




How many/far neighbors to reach? Implementing with LSTM (RNN) to incorporate “near” and “far” point correlations

Collaboration Model

For in-depth ML/application development…

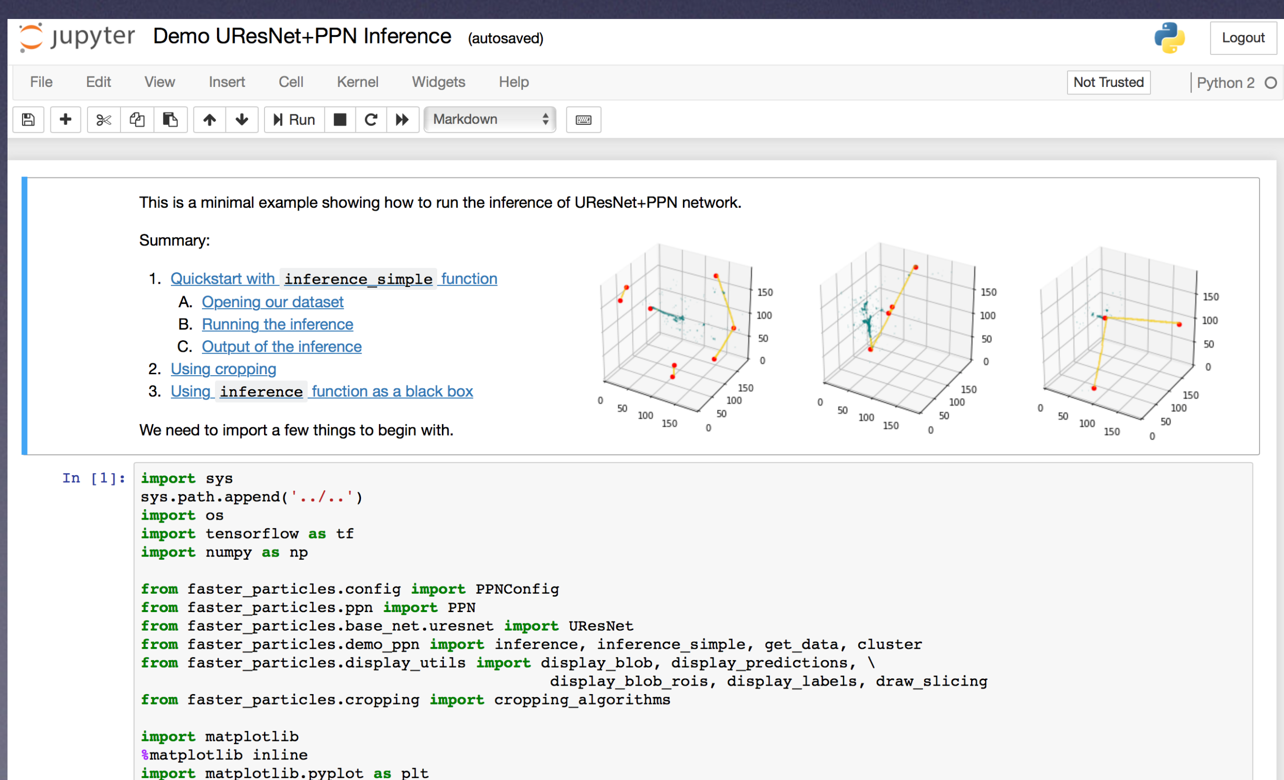


DeepLearnPhysics (deeplearnphysics.org)

- Group of ~70 physicists (in 8 months!) across national labs and universities
- ML & ML-application development, software and data sharing for reproducible results

SLAC resource

- 2.5 postdoc + student (DOE funding ECA + HEP ML)
- ~100 GPUs (~15 dedicated, 85 opportunistic)



For analysis development

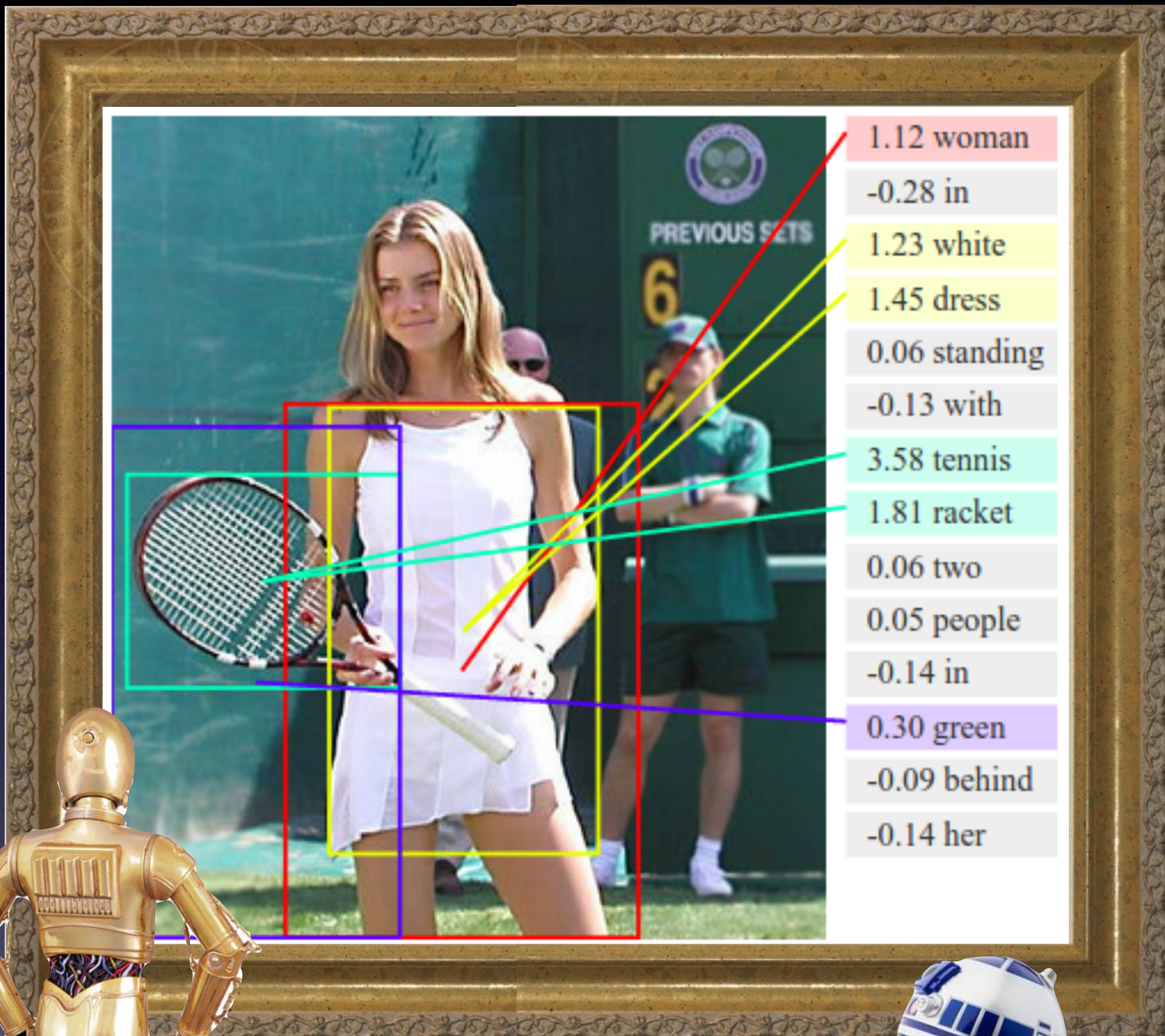
- **We provide completely reproducible workflow**
  - Contact me to get your hands on coding.
  - Tutorials exist & many people got started on their own. Workshops done/available.
- **We need your help!**

Five Messages

- Our research plan: **ML-based 3D reconstruction** chain for wire & pixel LArTPCs
- Current algorithms ready for some **design study**: **collaboration with LBNL** and beyond
- Have a working **model for collaboration**
- Will start working on a **large scale data processing**
- Exciting ideas to address **data sparsity challenges**

Back Up Slides

Image context analysis



"Pose" detection

Convolutional Neural Network

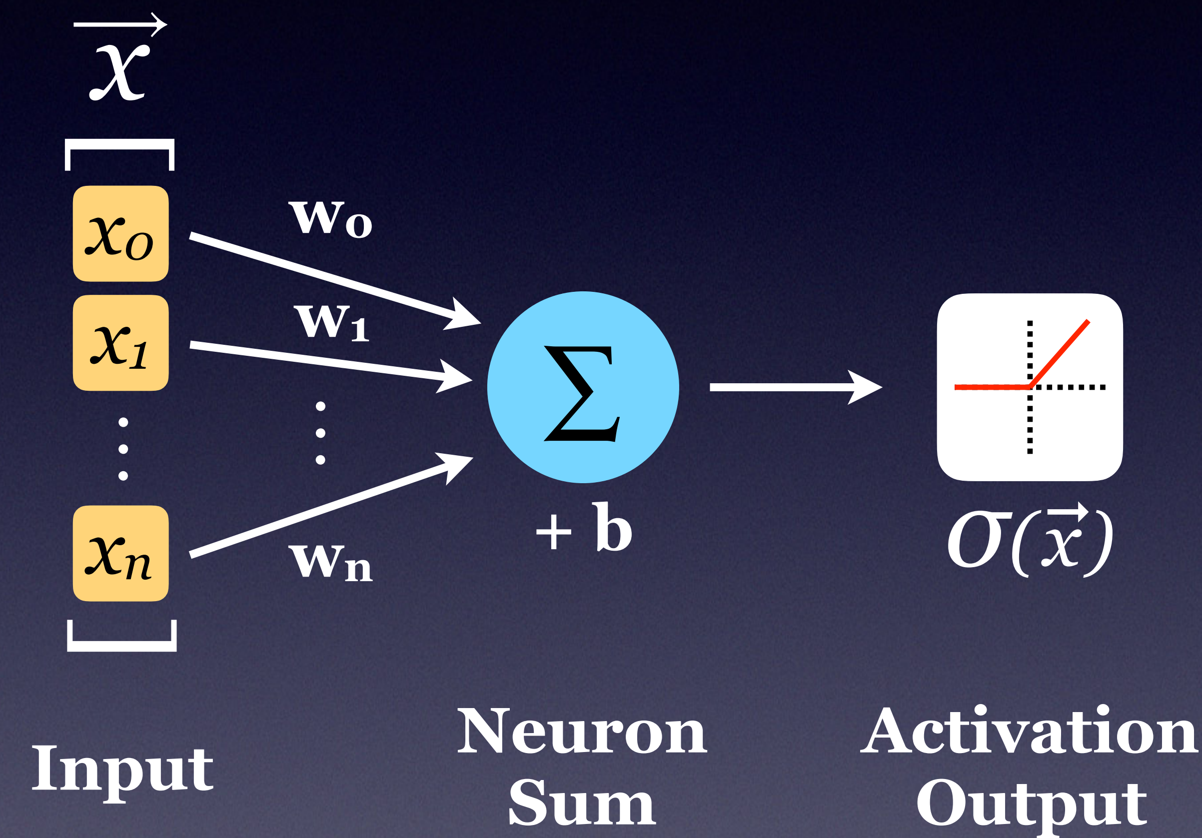
~ How does it work? ~

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.

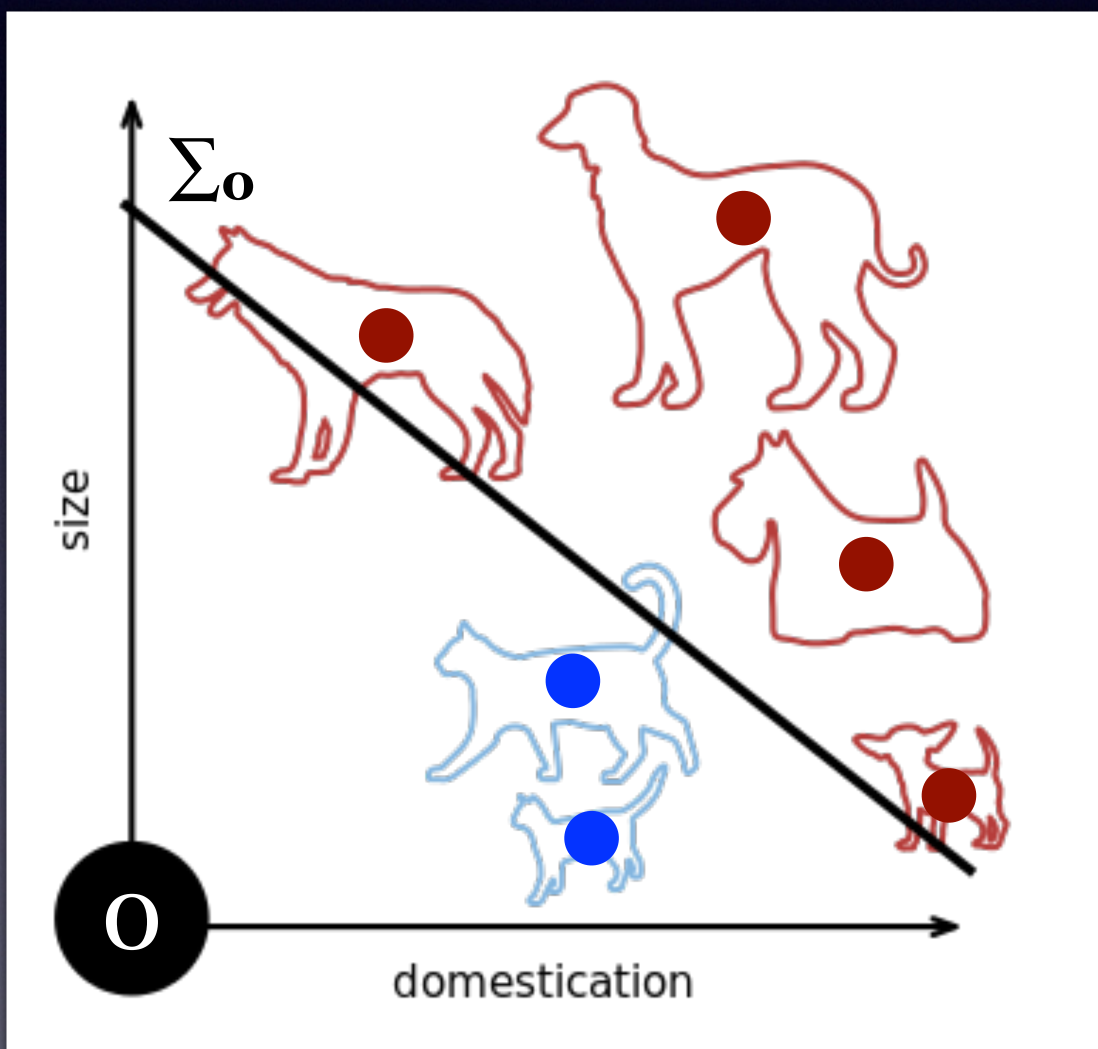


$$\sigma(\vec{x}) = \begin{cases} \vec{w}_i \cdot \vec{x} + b_i & \vec{w}_i \cdot \vec{x} + b_i \geq 0 \\ 0 & \vec{w}_i \cdot \vec{x} + b_i < 0. \end{cases}$$

How a Simple Perceptron Works

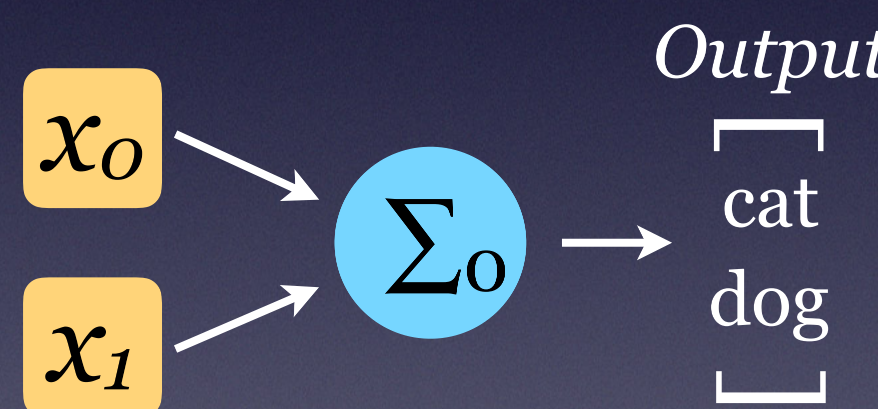
Perceptron 2D Classification

Imagine using two features to separate cats and dogs



from wikipedia

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

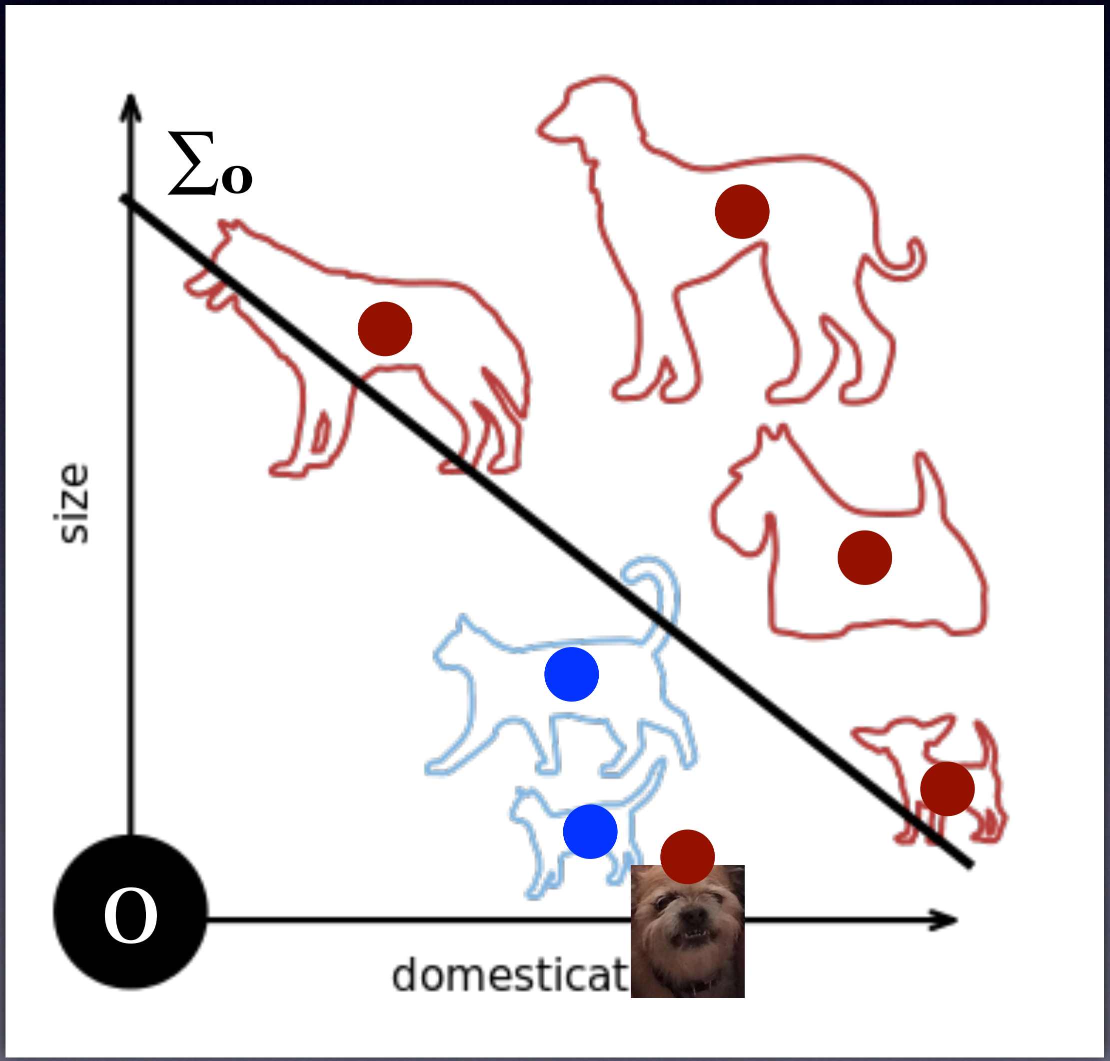


By picking a value for w and b, we define a boundary between the two sets of data

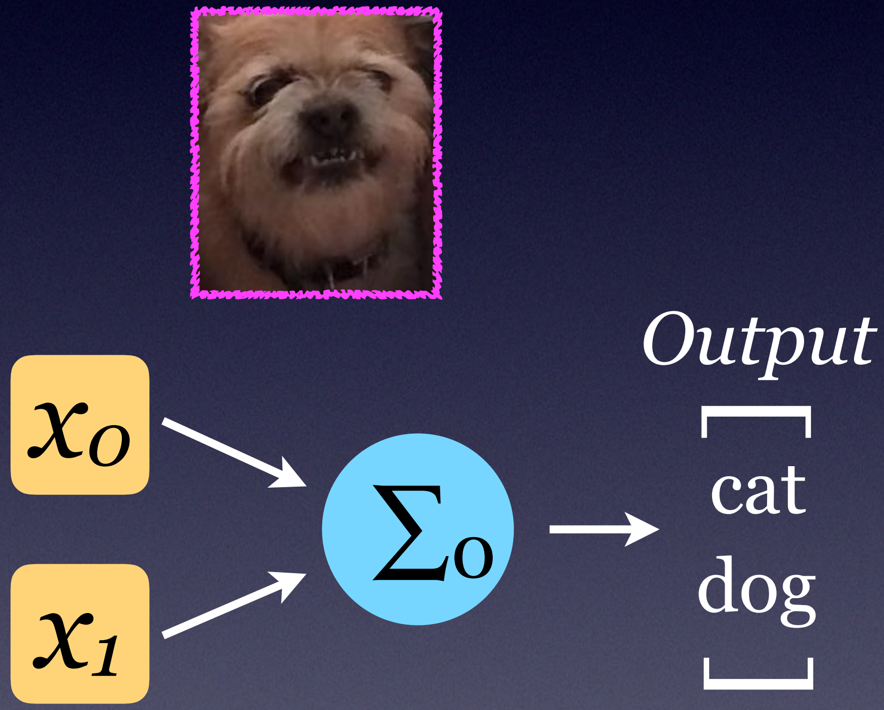
How a Simple Perceptron Works

Perceptron 2D Classification

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



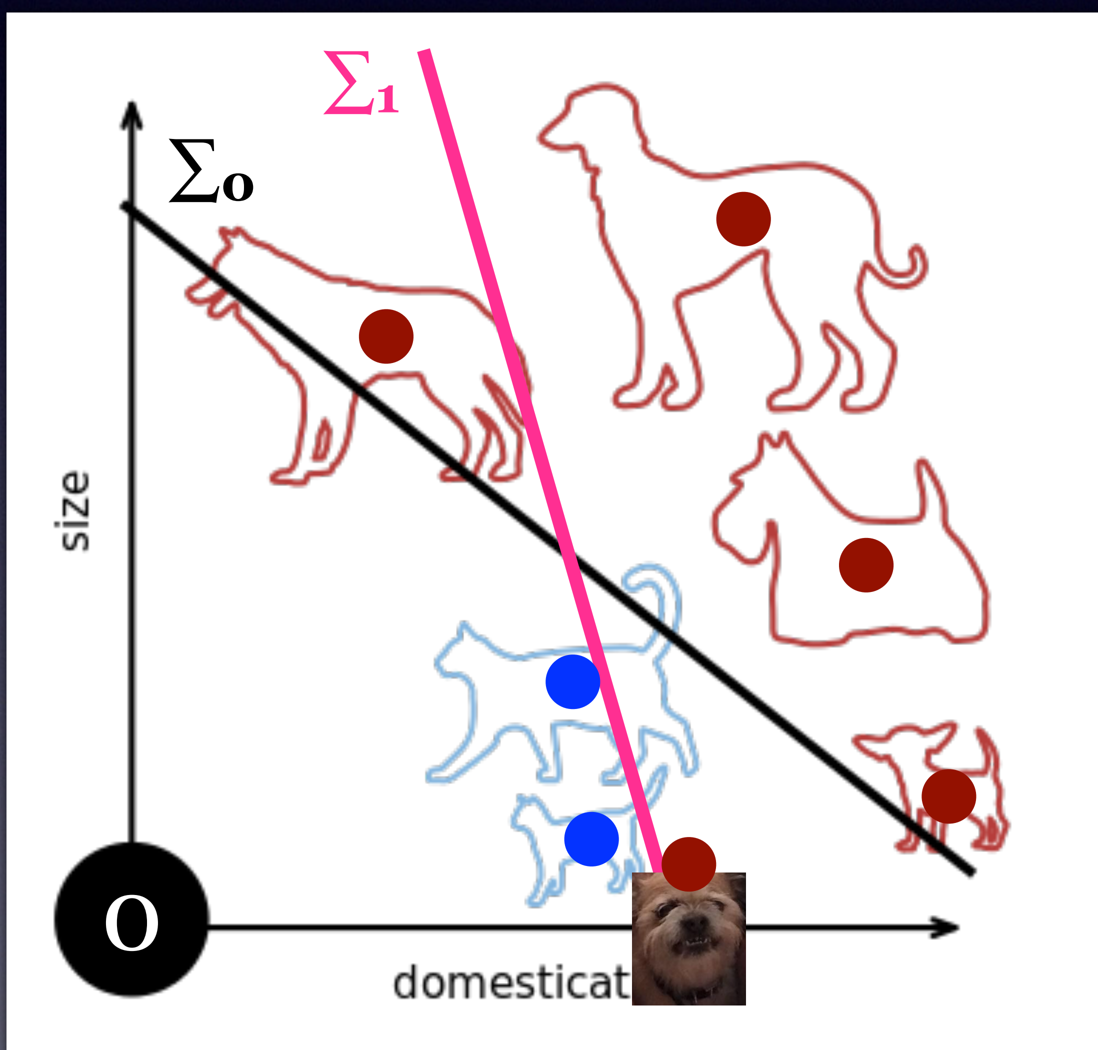
from [wikipedia]



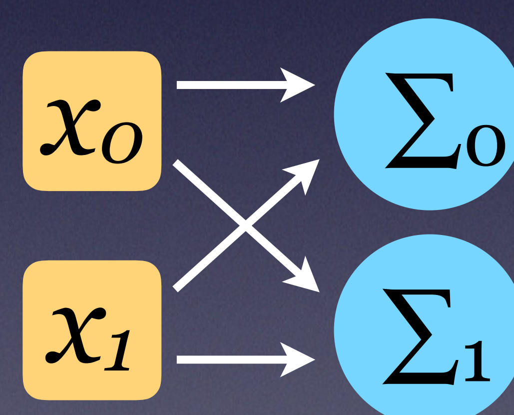
How a Simple Perceptron Works

Perceptron 2D Classification

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



from wikipedia

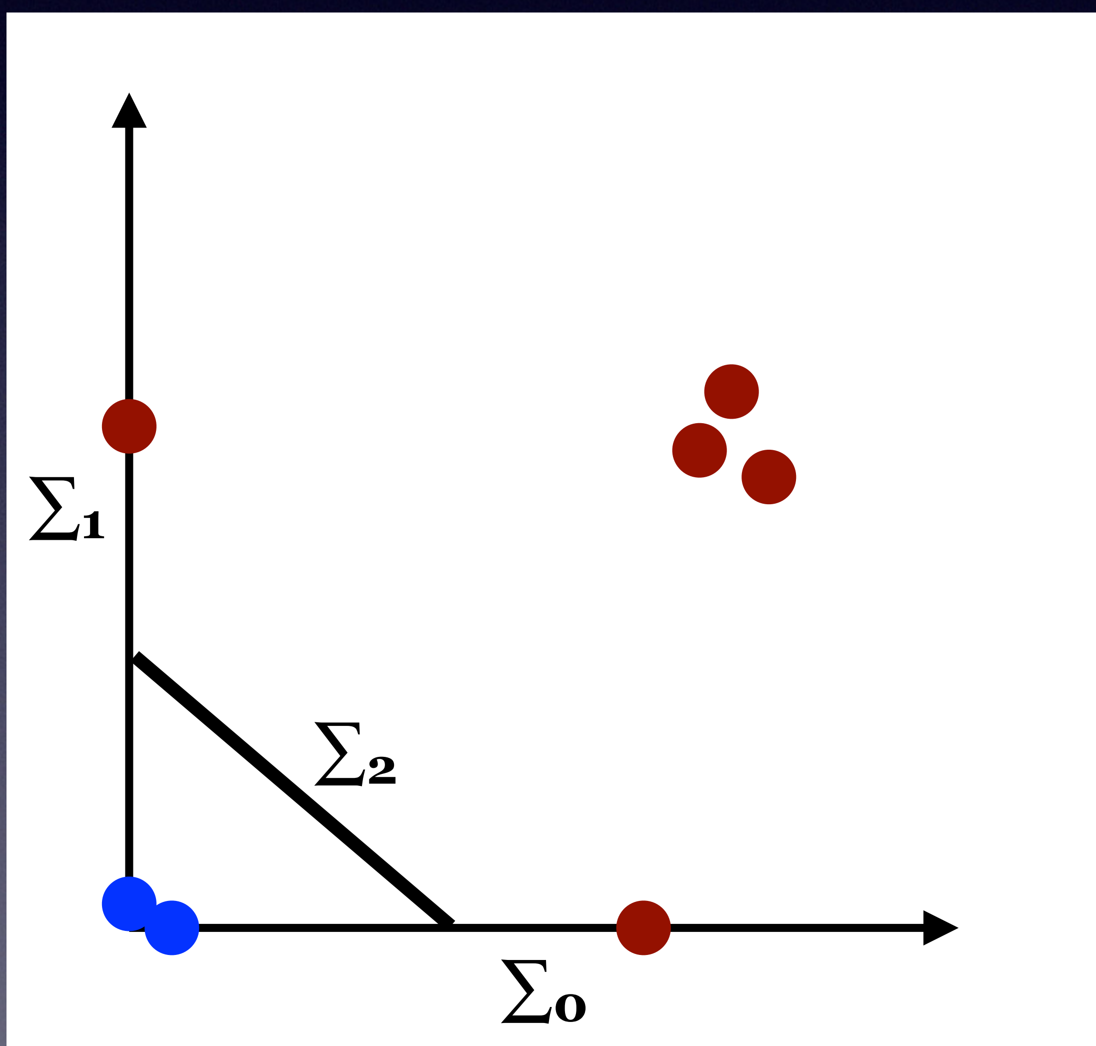


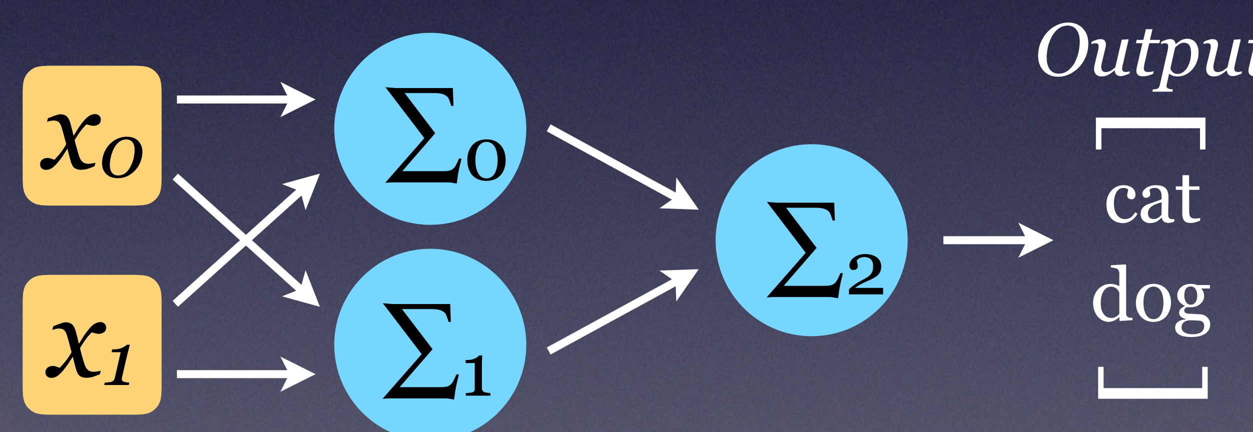
We can add another perceptron to help (but does not yet solve the problem)

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)

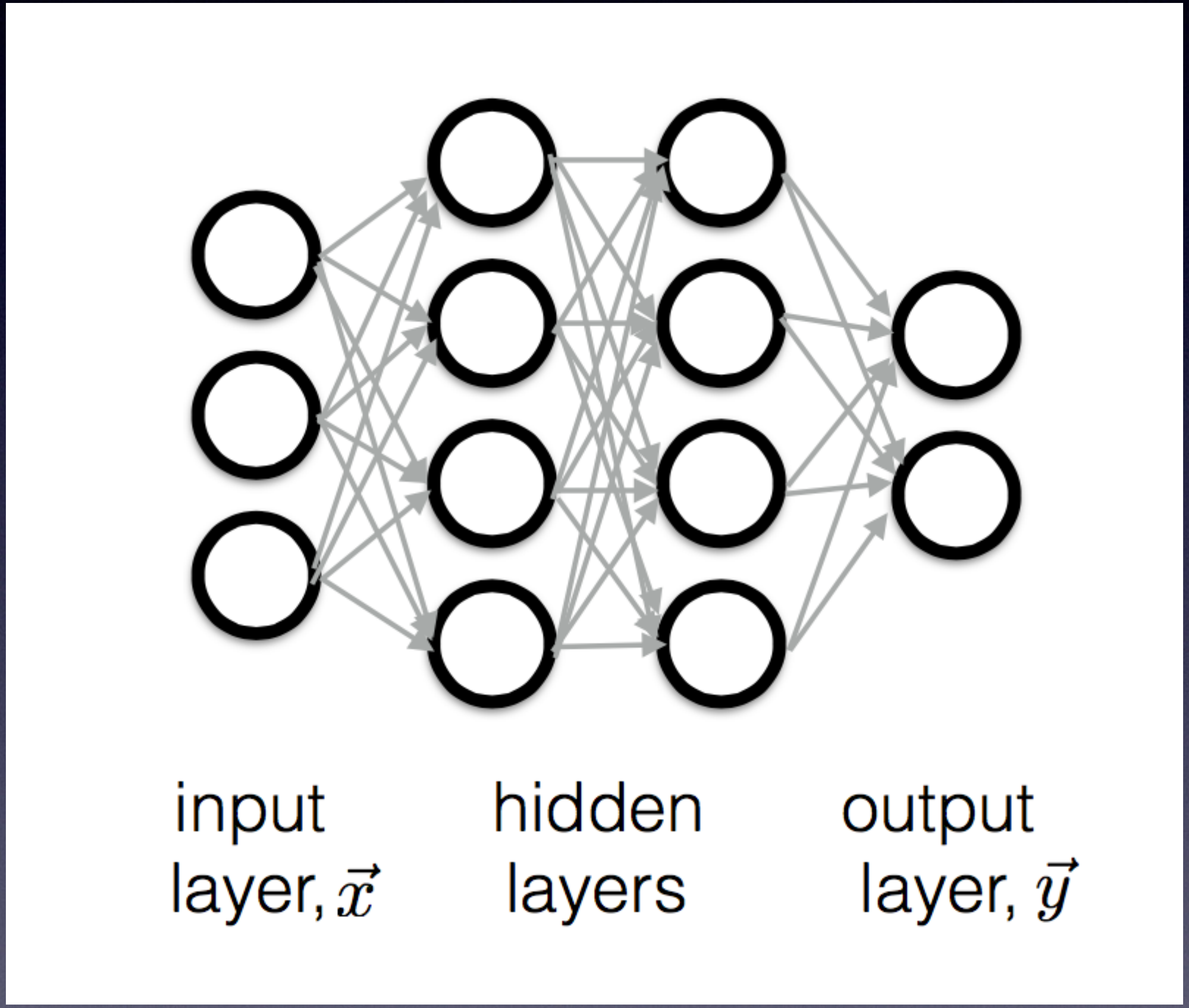




Another layer can classify based on preceding feature layer output

“Classical” Neural Net

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



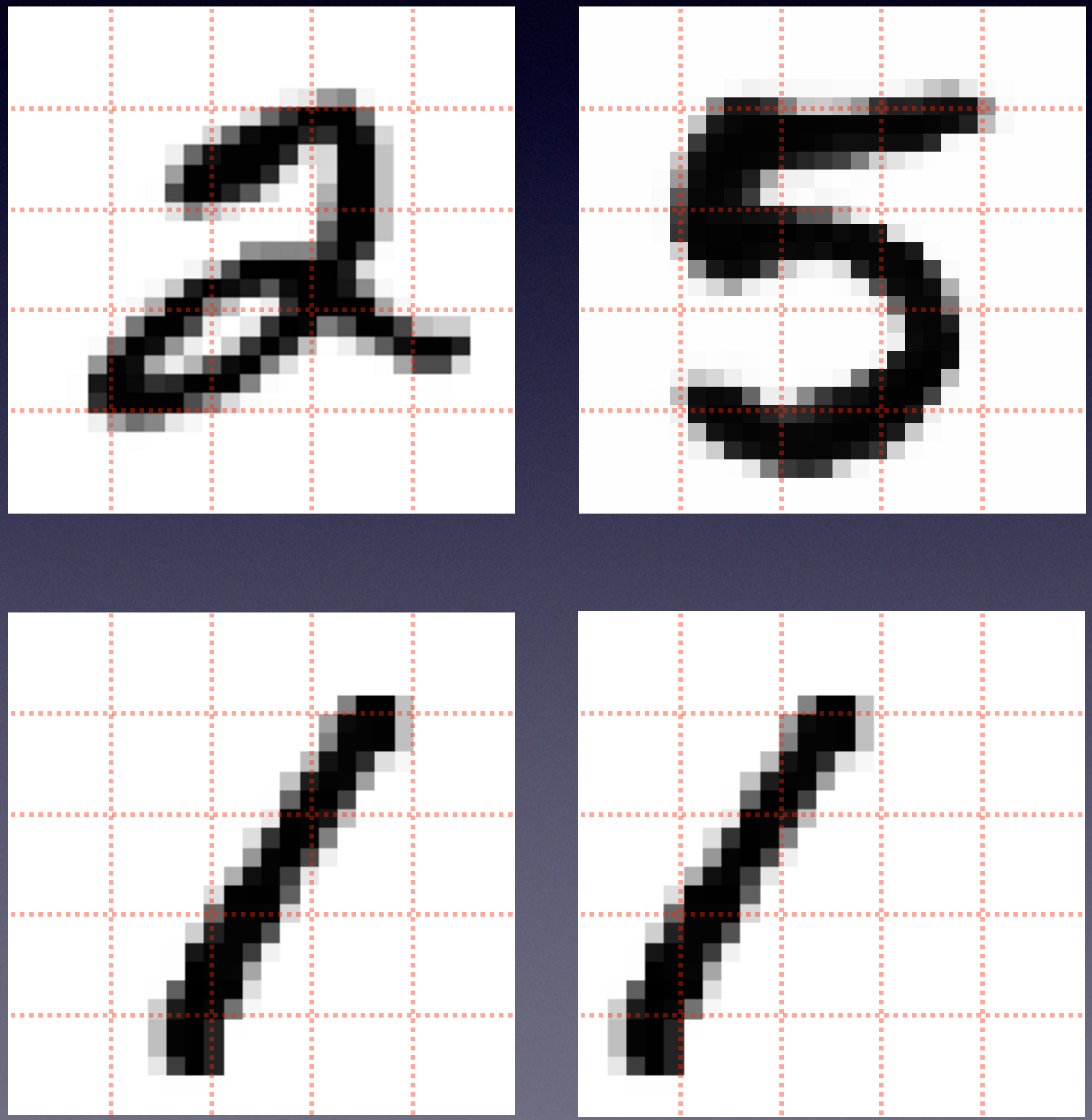
A traditional neural network consists of a stack of layers of such neurons where each neuron is ***fully connected*** to other neurons of the neighbor layers

“Classical” Neural Net

… is not ideal for image classification …

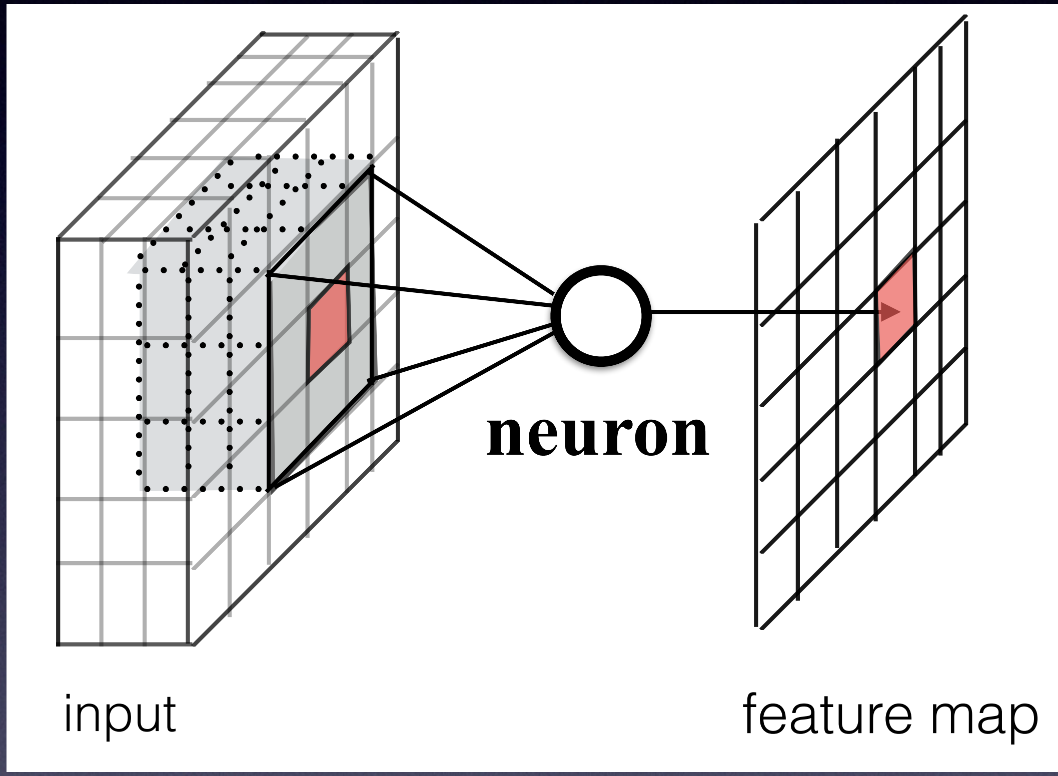
Image classification

- **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

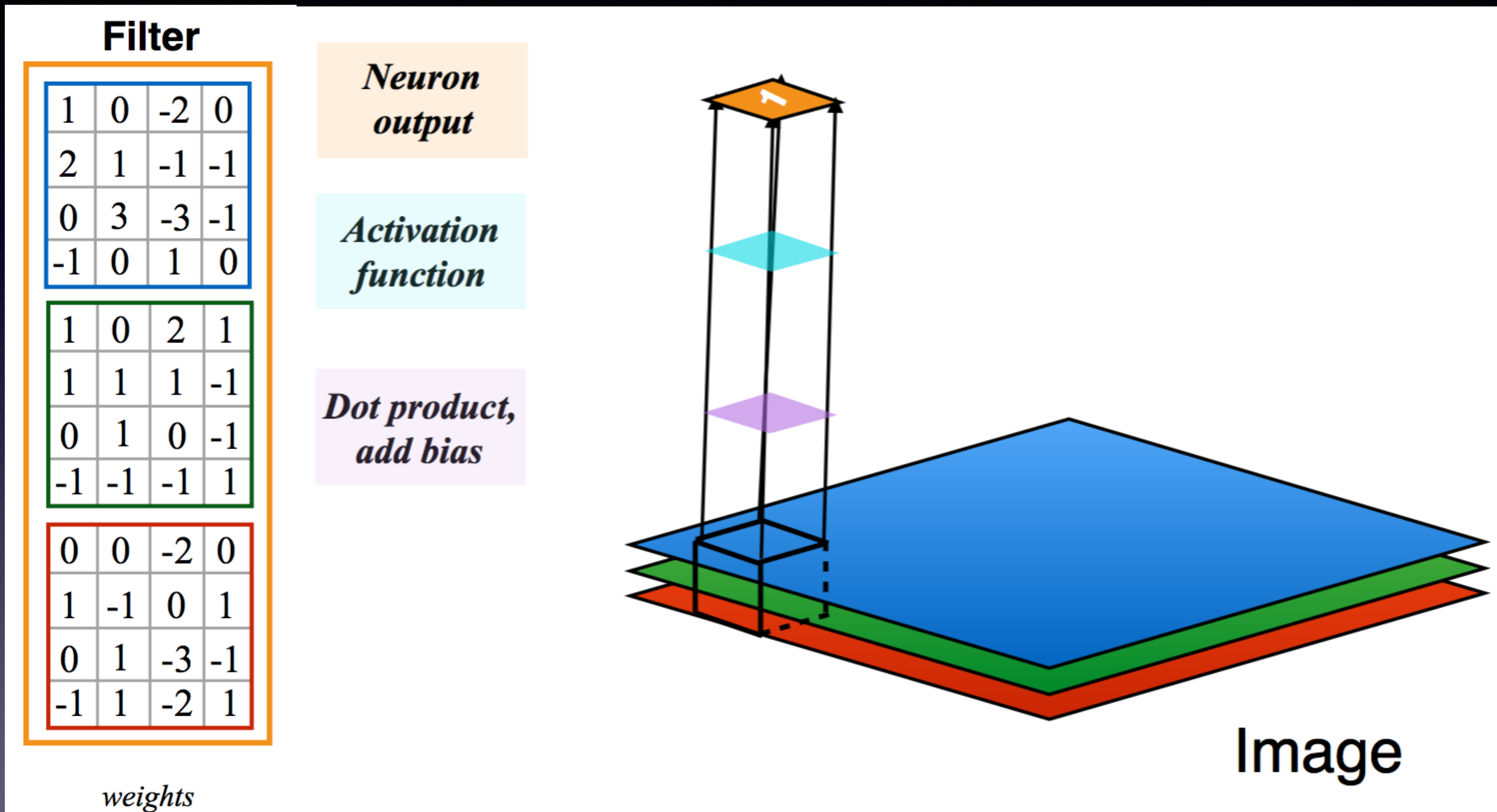


$$f_{i,j}(X) = \sigma\left(W_i \cdot X_j + b_i\right),$$

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

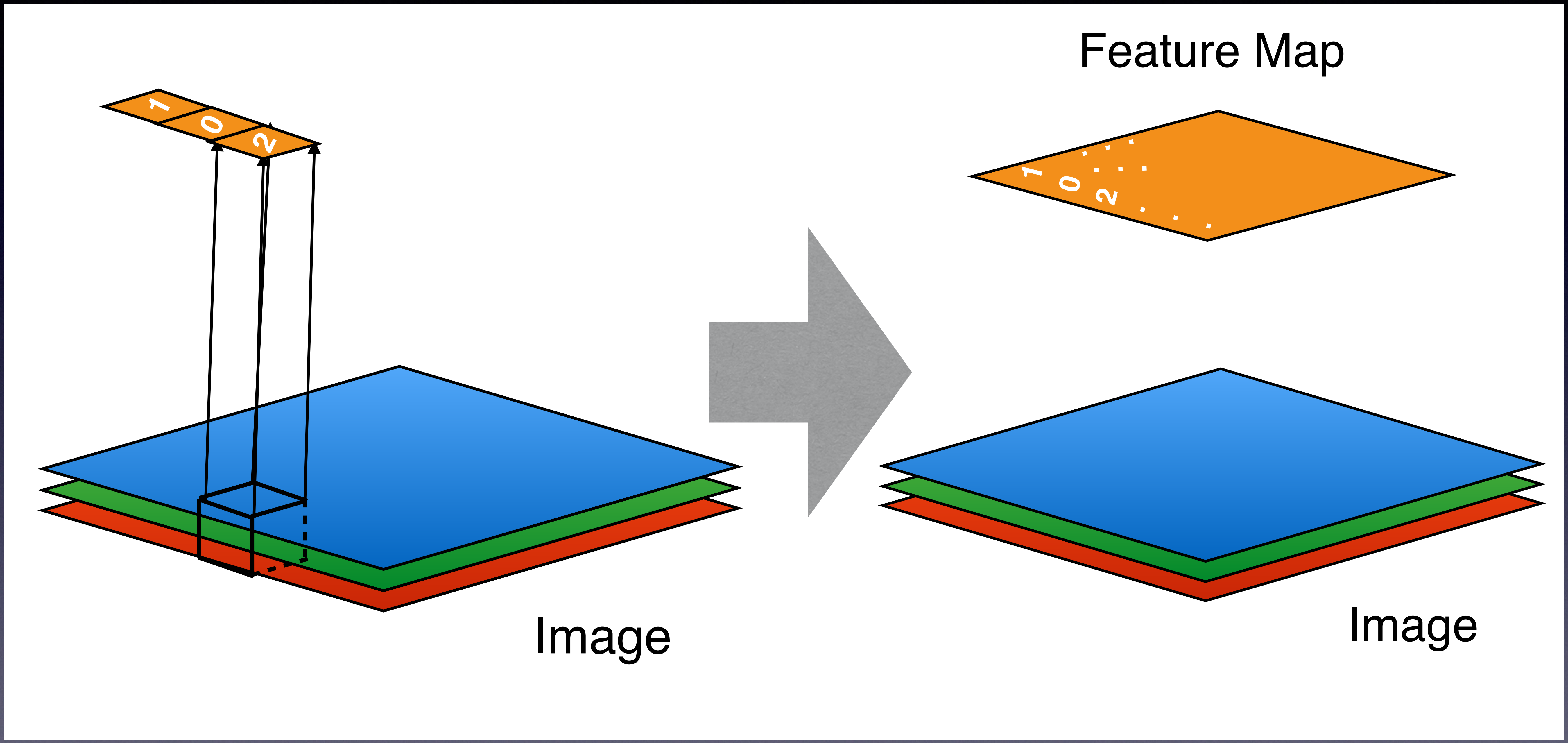
- Translate over 2D space to process the whole input
- Neuron **learns translation-invariant features**
  - Suited for a "***homogeneous***" detector like LArTPC
- **Output**: a "feature-enhanced" image (***feature map***)

Convolutional Neural Networks



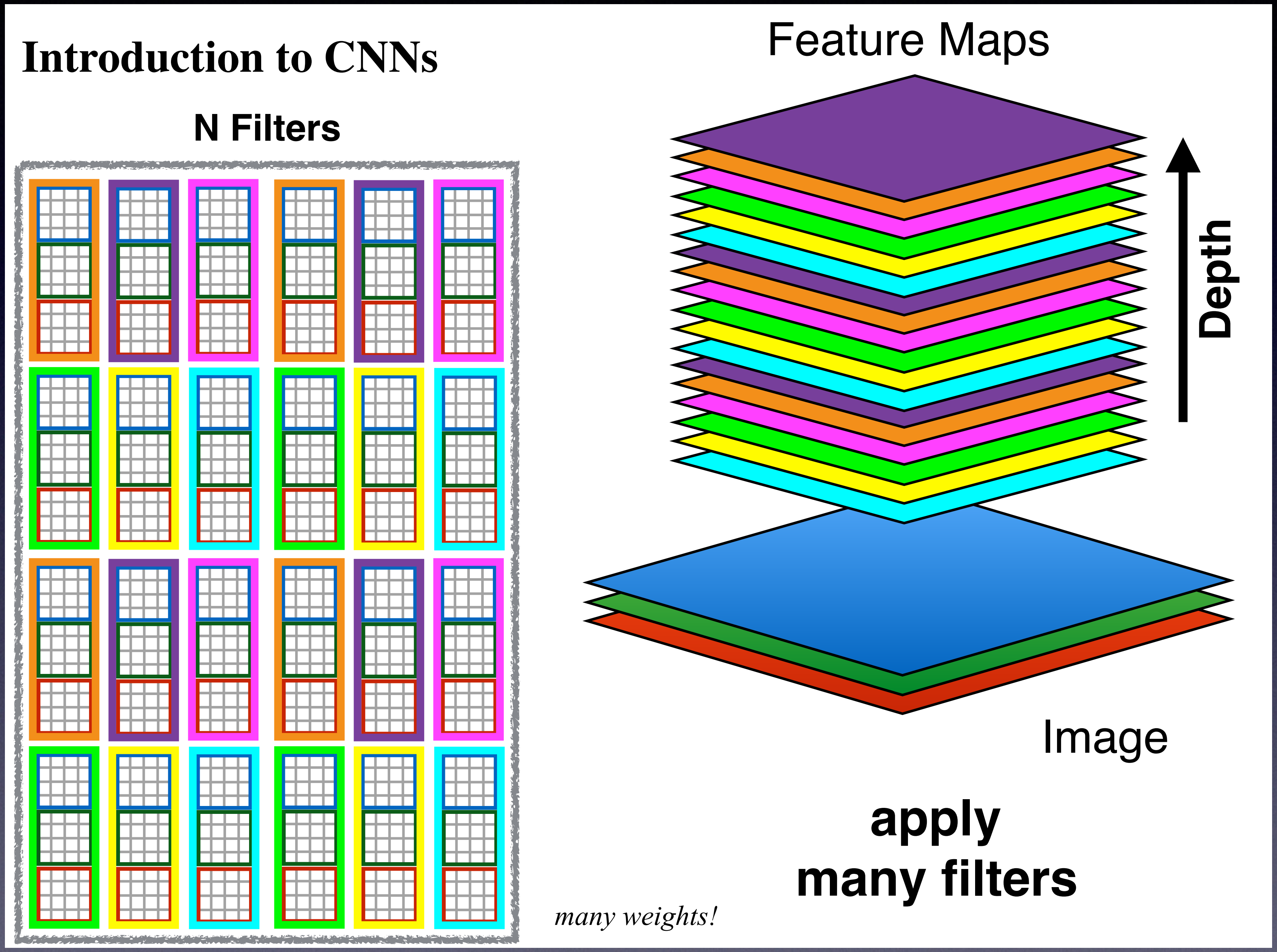
Toy visualization of the CNN operation

Convolutional Neural Networks



Toy visualization of the CNN operation

Convolutional Neural Networks

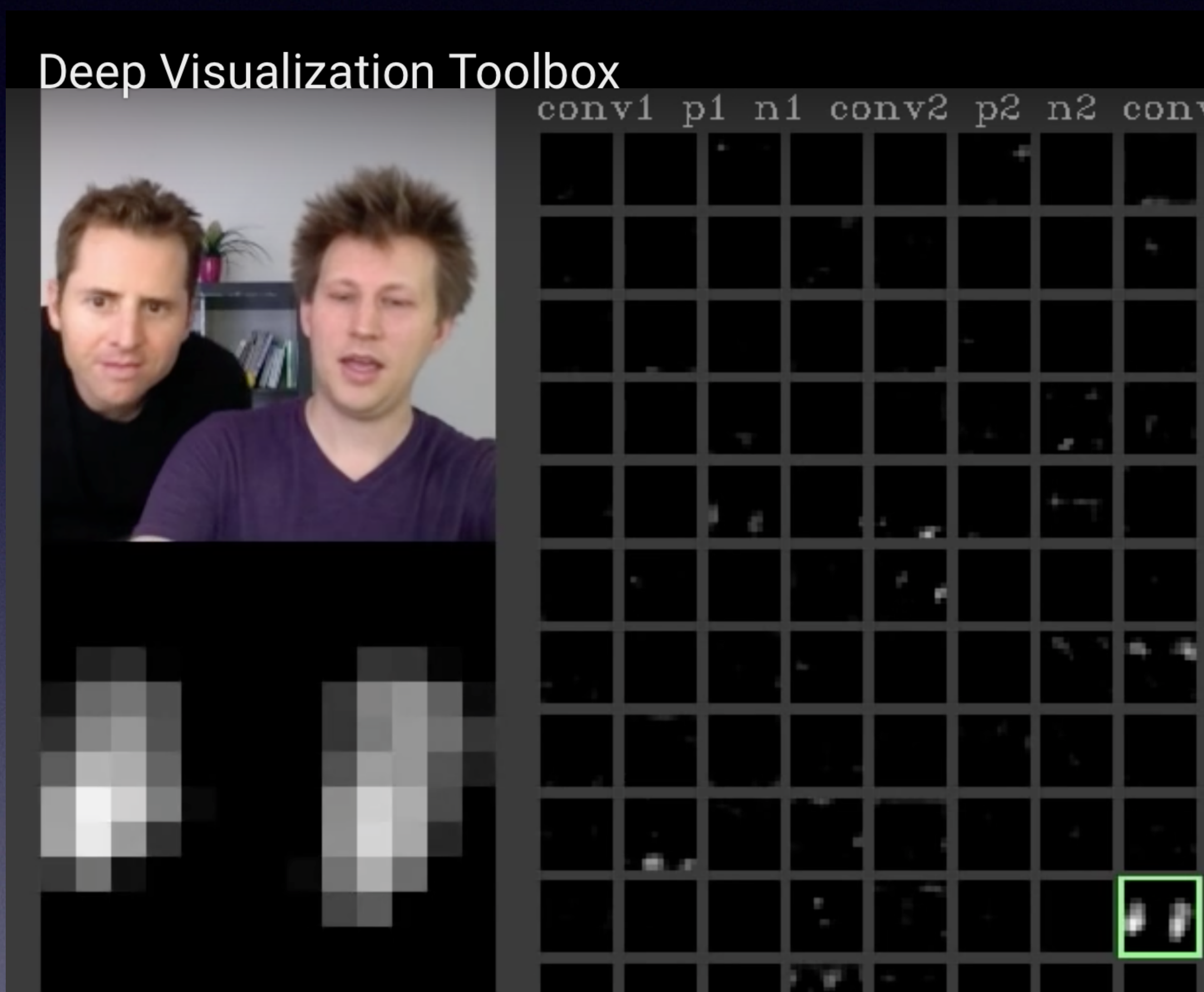


Toy visualization of the CNN operation

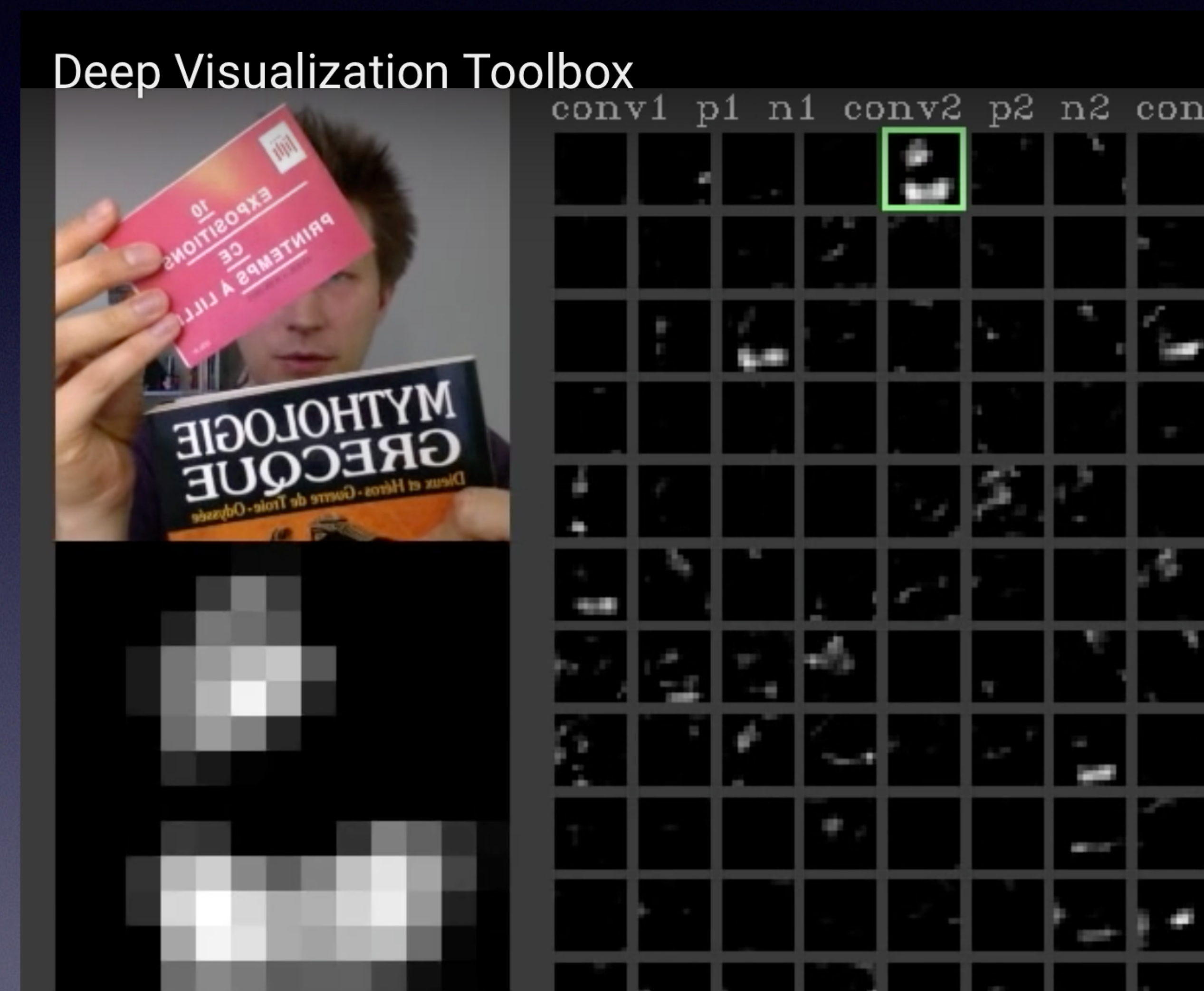
How Image Classification Networks Work

Feature map visualization example

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



Neuron concerning face

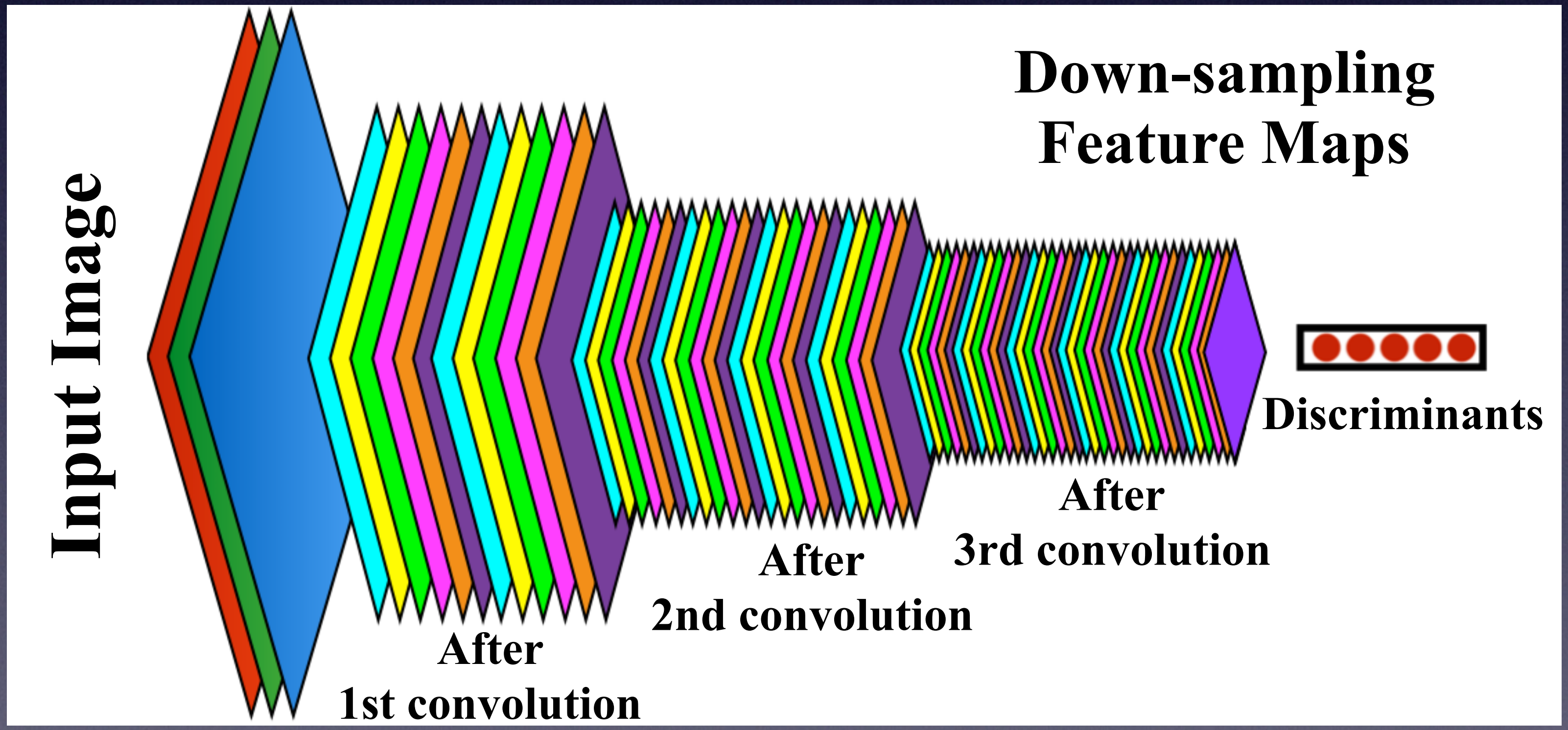


Neuron loving texts
(and don't care about your face)

How Image Classification Networks Work

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

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

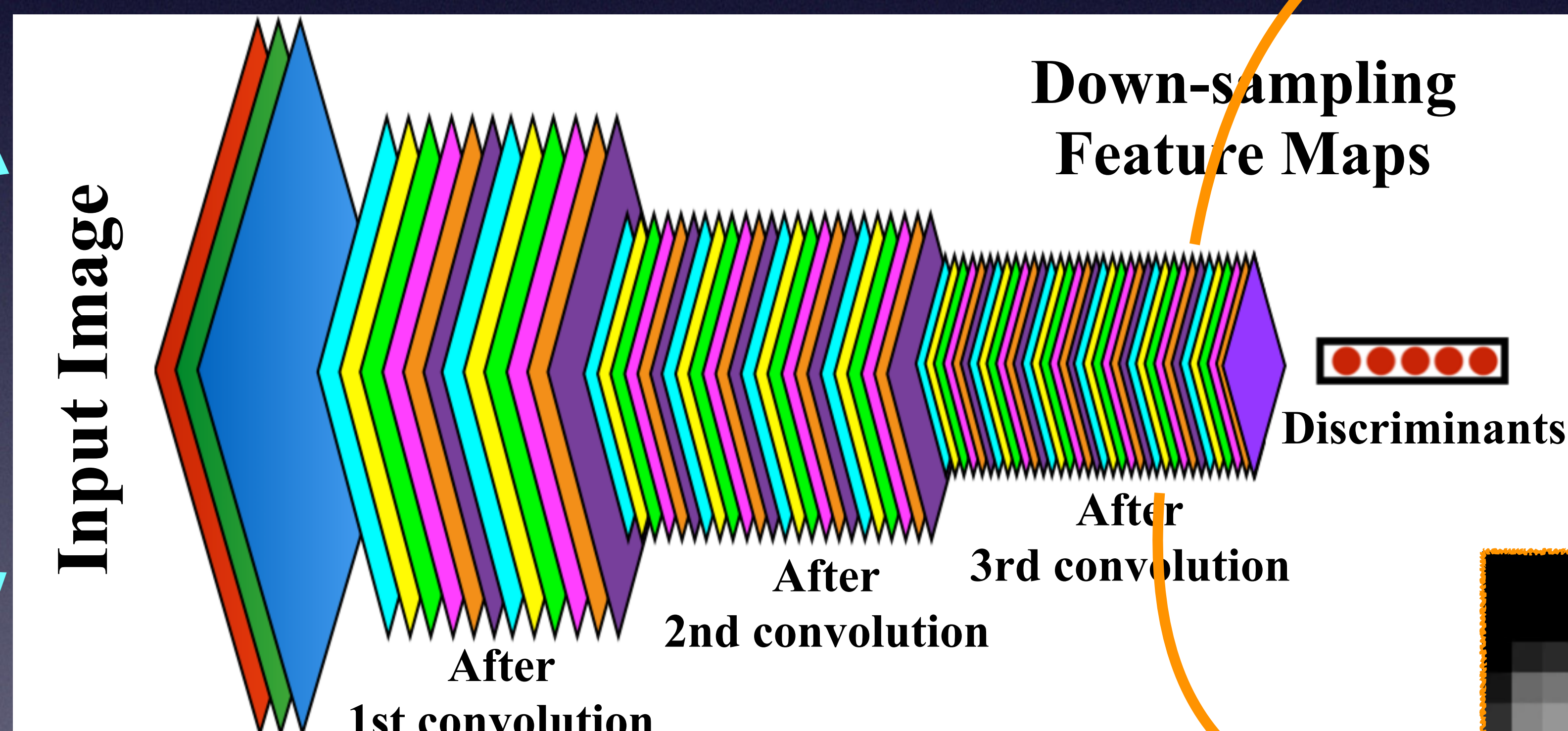


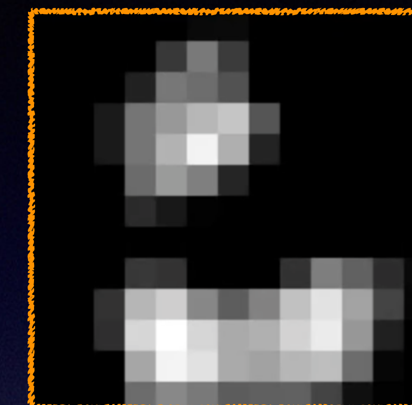
Series of convolutions
+ down-sampling

How Image Classification Networks Work

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

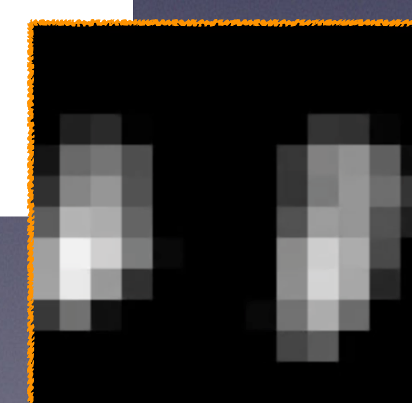
1. **Convolution operation**
2. **Down-sampling**





"Written Texts"
feature map

Series of convolutions
+ down-sampling



"Human Face"
feature map