MACHINE LEARNING, NUCLEAR PHYSICS, AND ALGORITHM DEVELOPMENT FOR DATA ANALYSIS IN NUCLEAR RESEARCH

MICHELLE KUCHERA
DAVIDSON COLLEGE

JOINT ICTP-IAEA WORKSHOP ADVANCED SCHOOL ON COMPUTATIONAL NUCLEAR SCIENCE

23 MAY 2022

ALPhA
DAVIDSON COLLEGE

CERN

CMS

FRIB

Jefferson Lab

DAVIDSON

NSCL

SESSION 2 TOPICS

- Convolutional Layers
- Classification
- Pre-trained models
- Unsupervised Methods
- Best practices
- Hot topics in ML research

MICHELLE KUCHERA
DAVIDSON COLLEGE

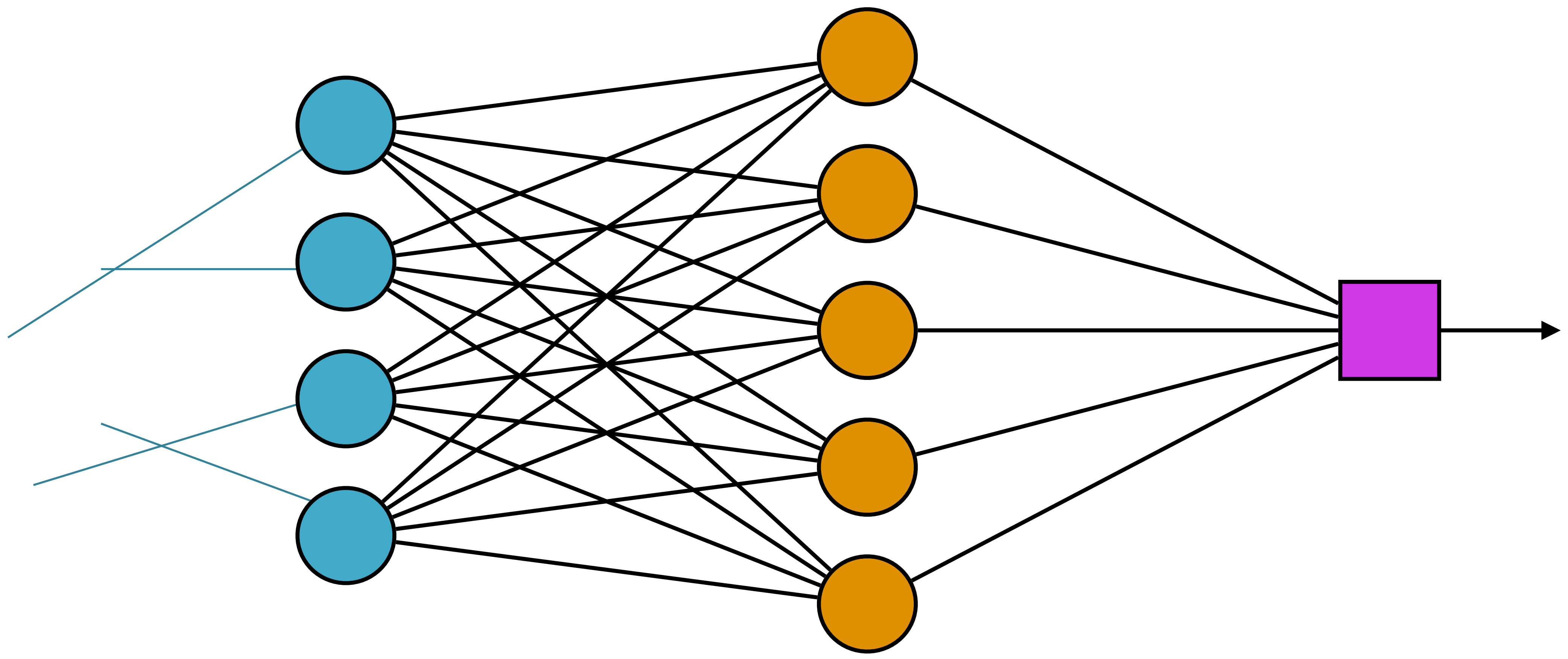
JOINT ICTP-IAEA WORKSHOP ADVANCED SCHOOL ON COMPUTATIONAL NUCLEAR SCIENCE

23 MAY 2022

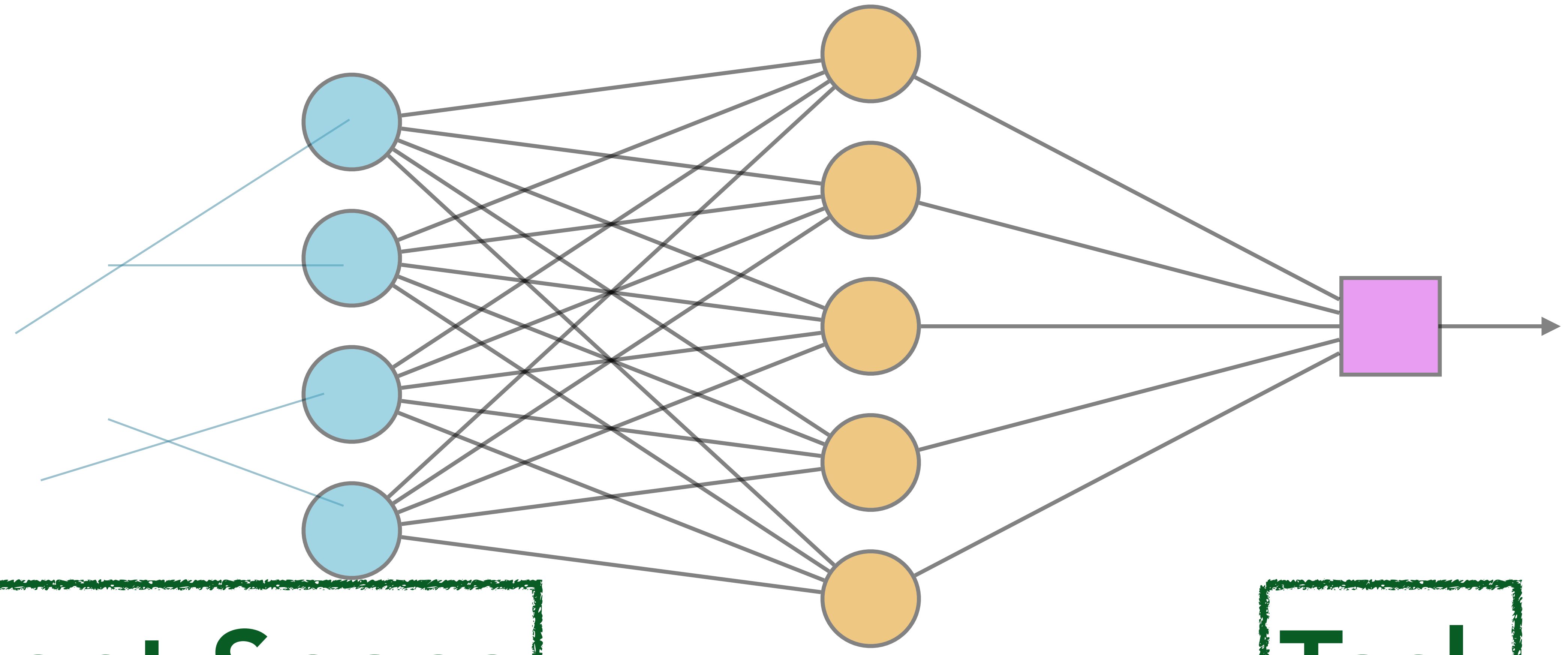
CONVOLUTIONAL NEURAL NETWORKS

CLASSIFICATION

CONVOLUTIONAL NEURAL NETWORKS



CONVOLUTIONAL NEURAL NETWORKS

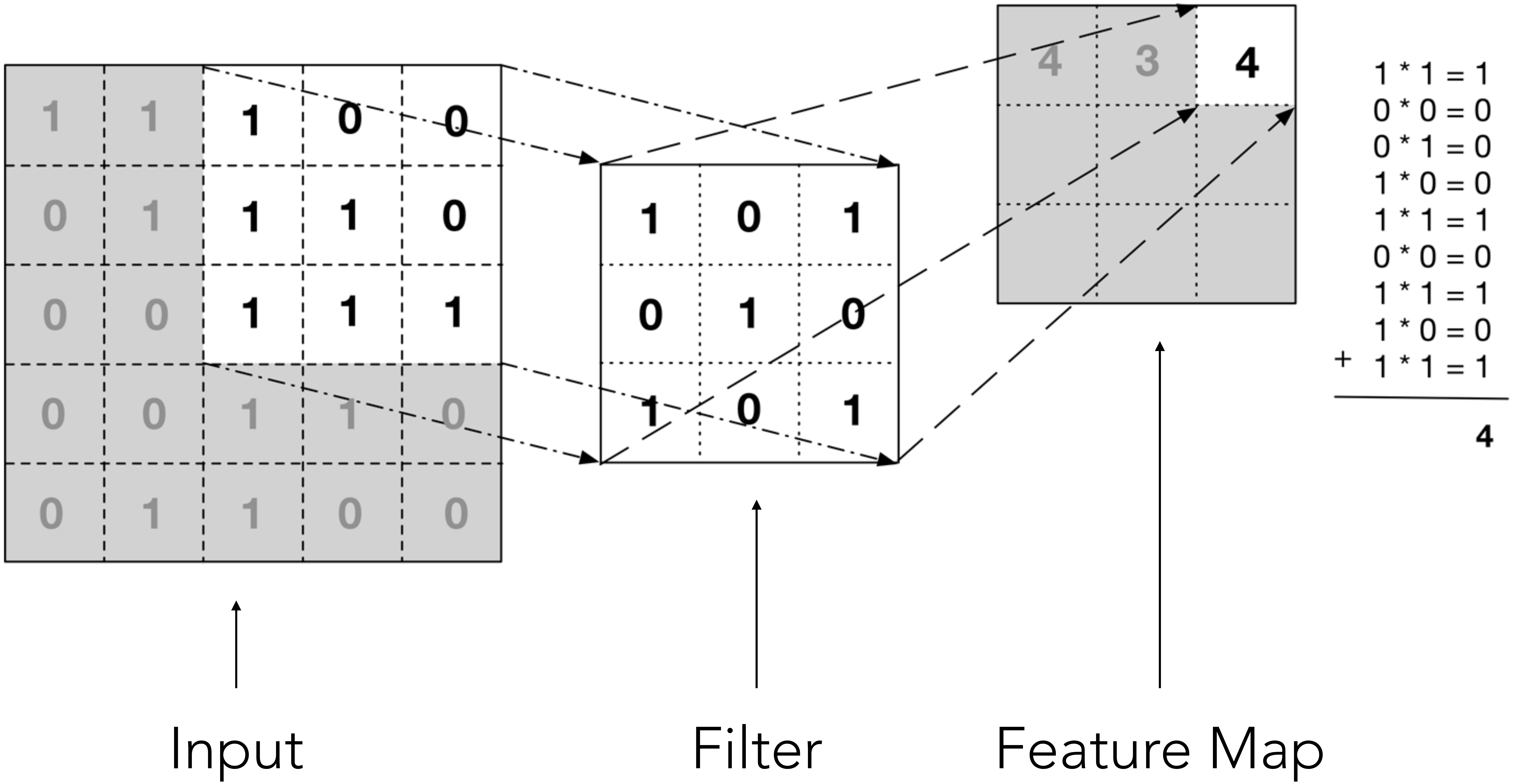


Feature Extraction

Latent Space

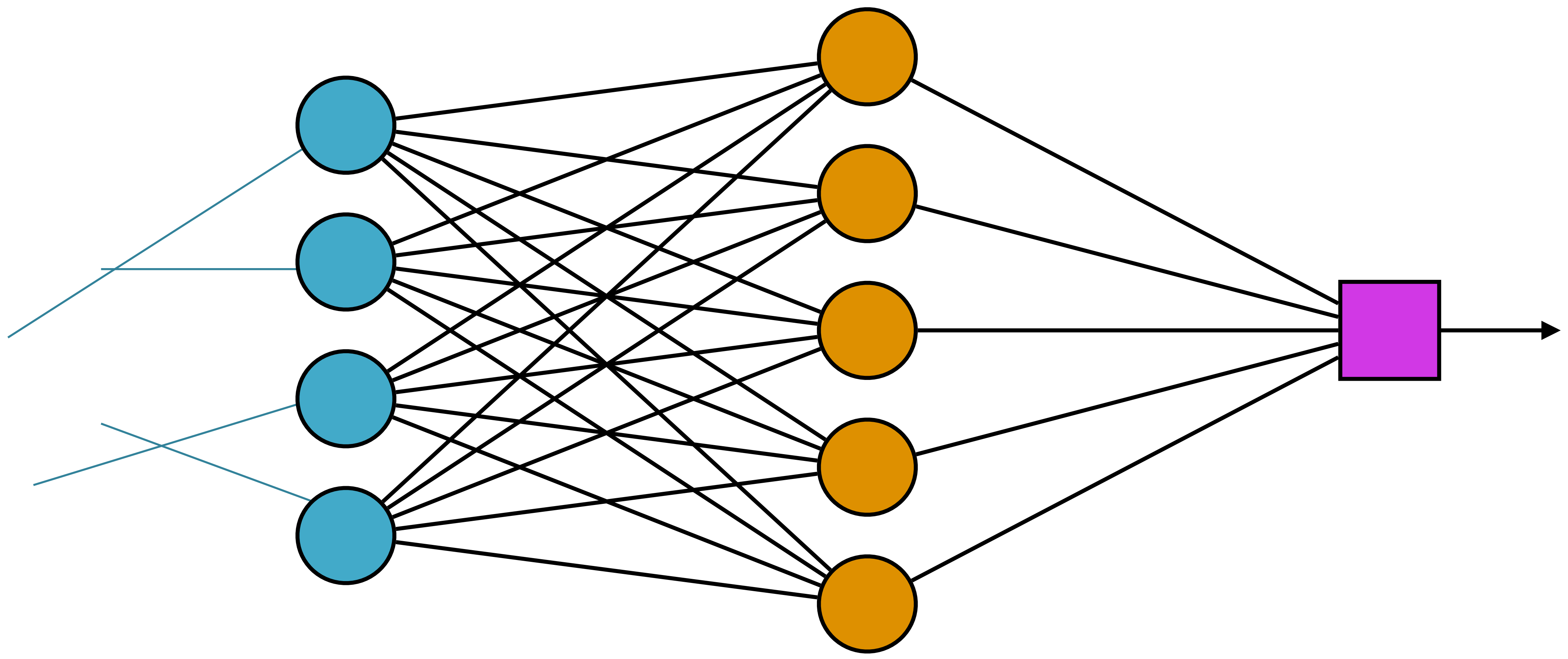
Task

DISCRETE CONVOLUTION



ADAPTED FROM *DEEP LEARNING*, ADAM GIBSON & JOSH PATTERSON

CONVOLUTIONAL NEURAL NETWORKS

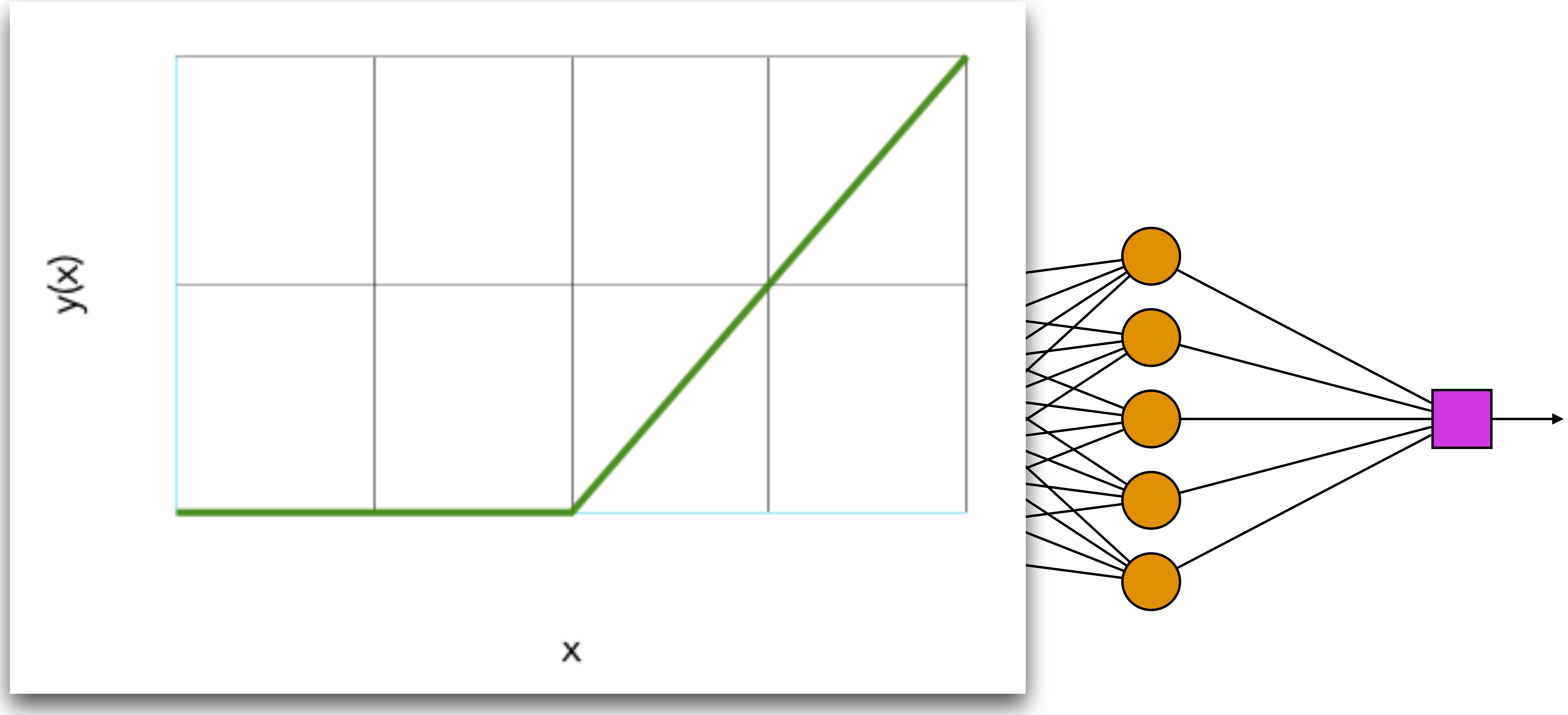



1156

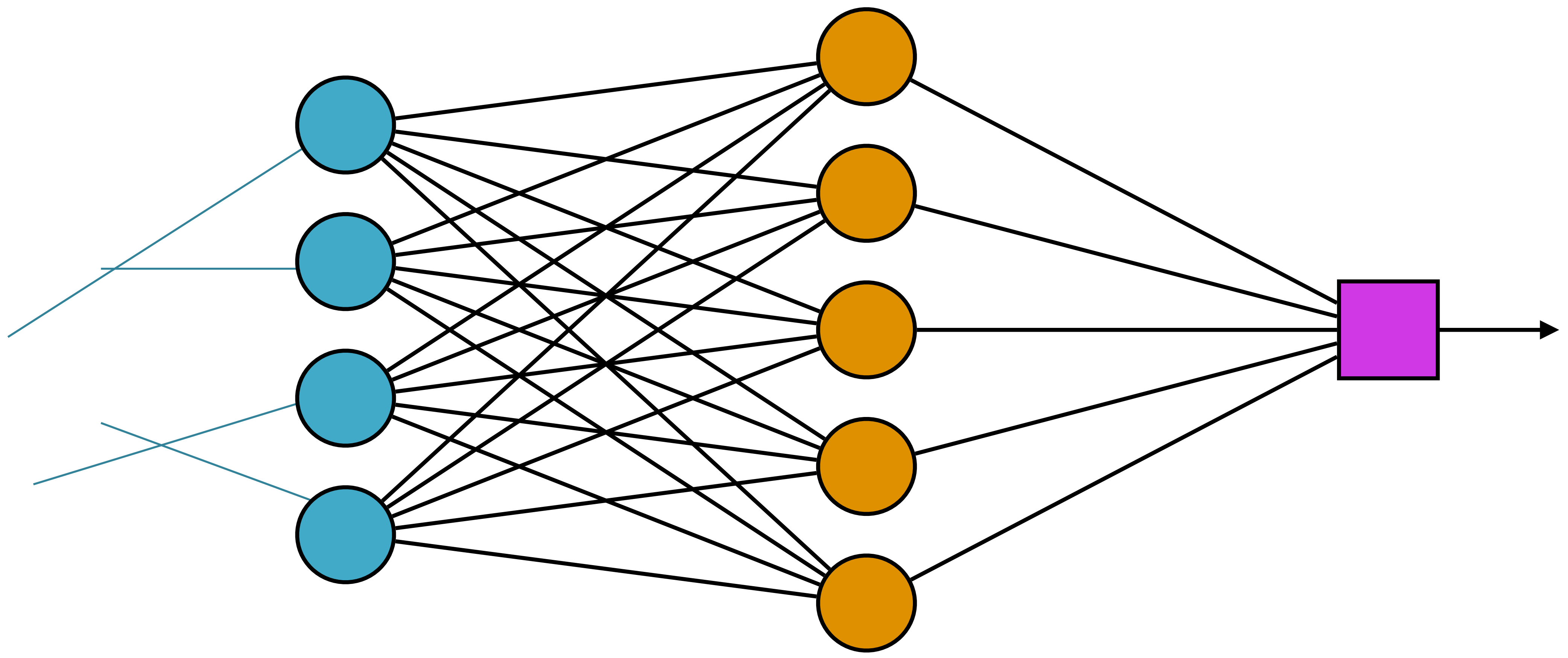
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
5	5	5	5	5
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1

-1	-1	5	-1	-1
-1	-1	5	-1	-1
-1	-1	5	-1	-1
-1	-1	5	-1	-1
-1	-1	5	-1	-1

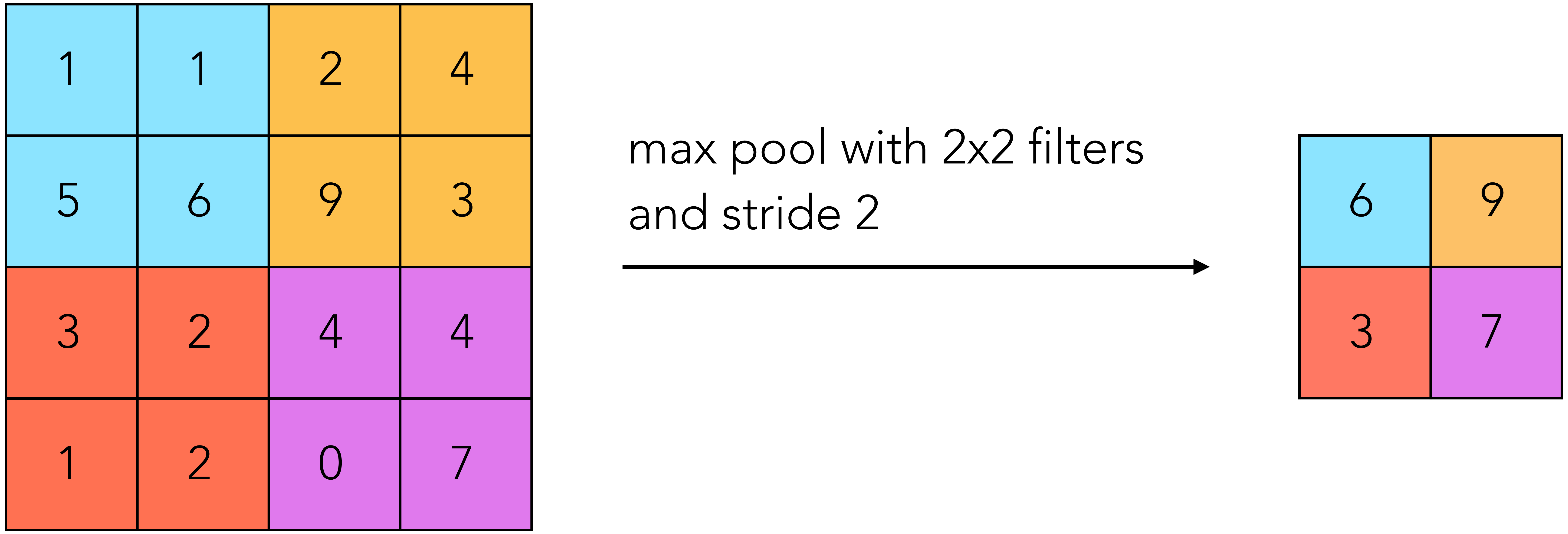
CONVOLUTIONAL NEURAL NETWORKS



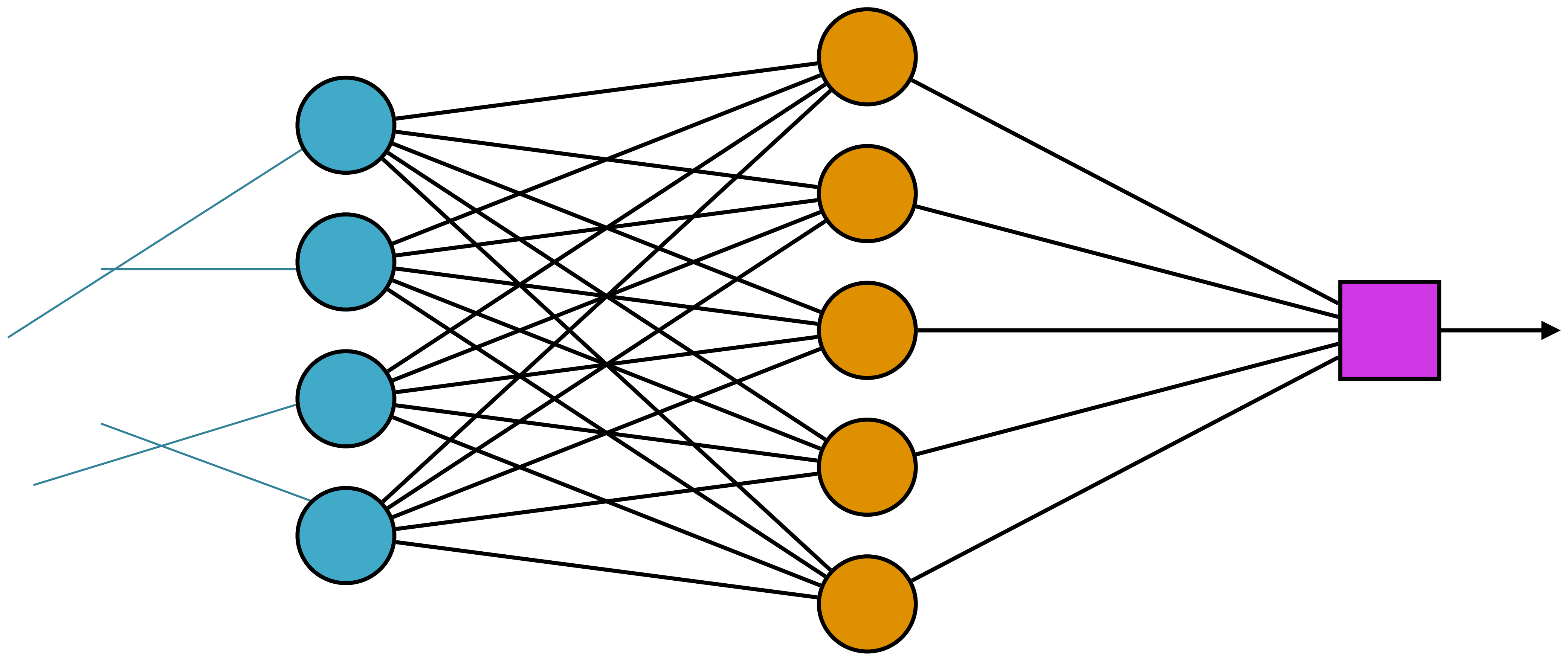
CONVOLUTIONAL NEURAL NETWORKS



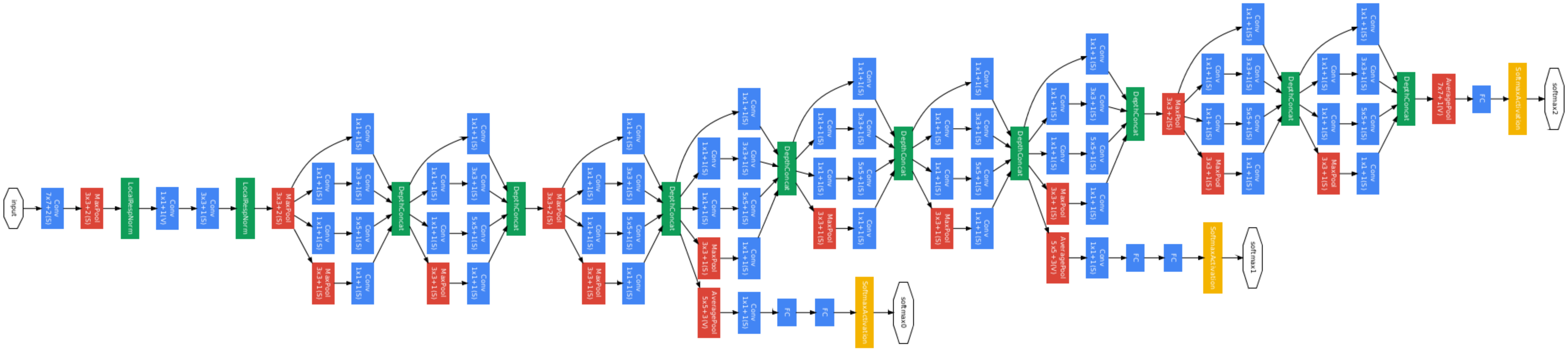
MAX POOLING


1 1 2 4
5 6 9 3
3 2 4 4
1 2 0 7
max pool with 2x2 filters
and stride 2
6 9
3 7

CONVOLUTIONAL NEURAL NETWORKS



“GoogLeNet network with all the bells and whistles”



CHRISTIAN SZEGEDY ET. AL. GOING DEEPER WITH CONVOLUTIONS.

CHOOSING AN ARCHITECTURE

HOW MANY LAYERS?

HOW MANY NODES PER LAYER?

LEARNING RATE

DROPOUT?

WHAT ACTIVATION FUNCTION(S)?

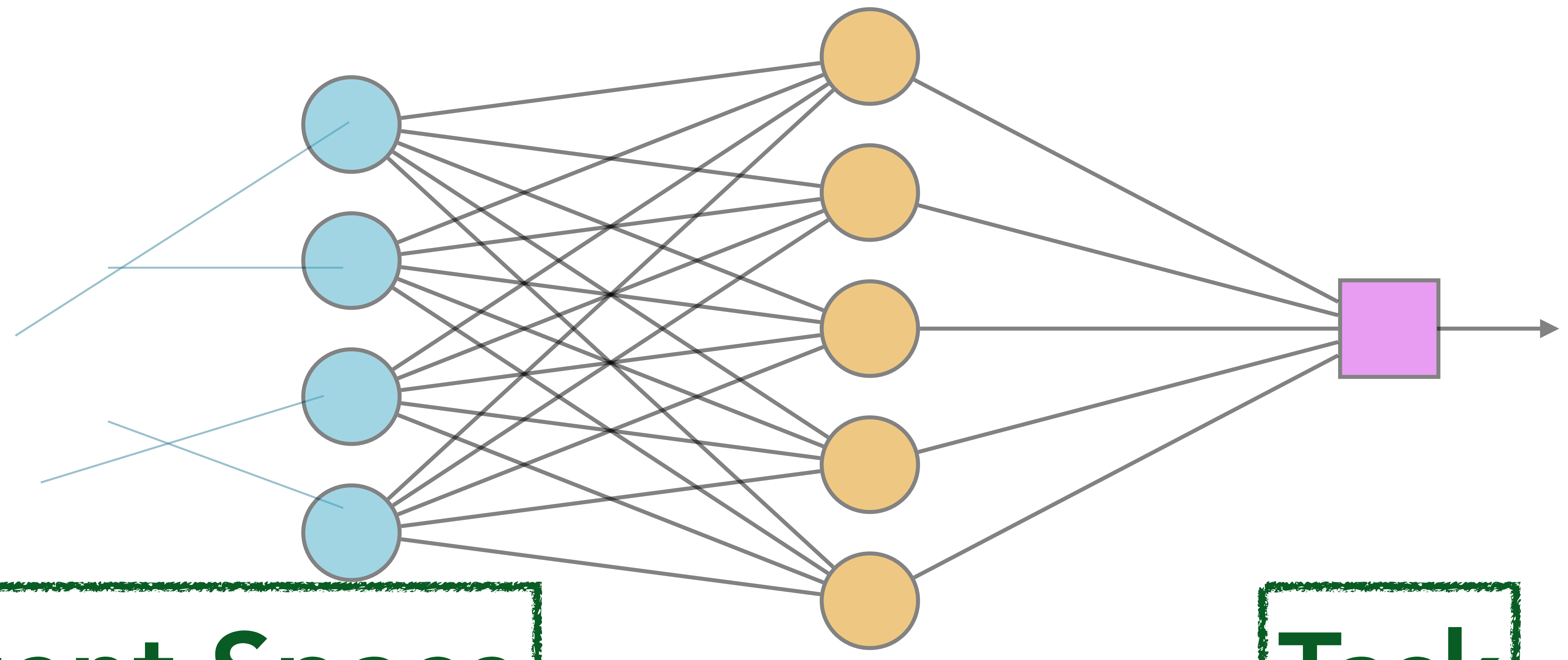
HOW MANY CONVOLUTION LAYERS?

FILTER SIZE?

STRIDE?

POOLING?

PRE-TRAINED MODELS

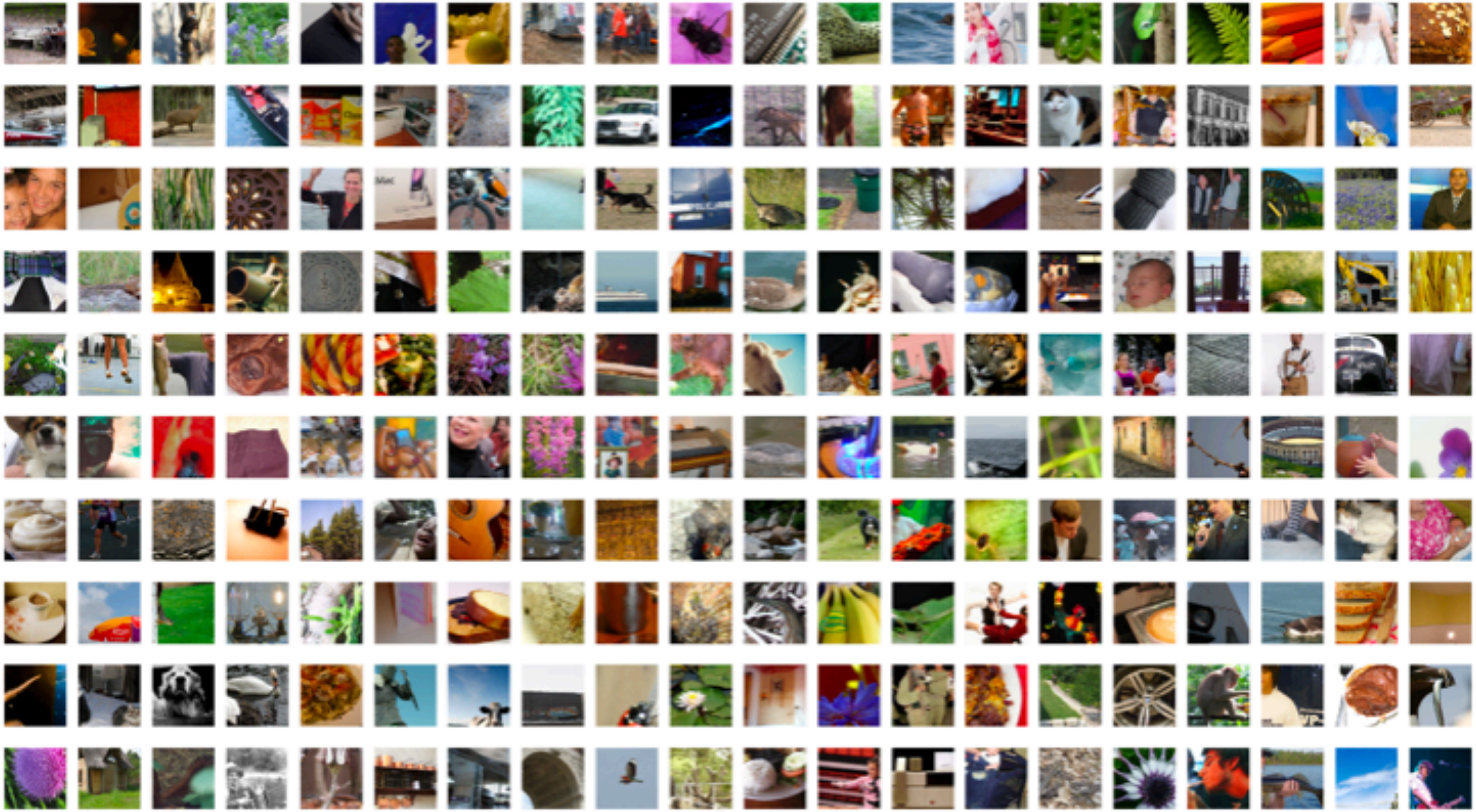


Feature Extraction

Latent Space

Task

PRE-TRAINED MODELS



PRETRAINED MODELS

PRETRAINED MODELS

ALEXNET

VGG

RESNETS

INCEPTION

XCEPTION

ALEXNET

$\boxed{\text{VGG}}$

RESNETS

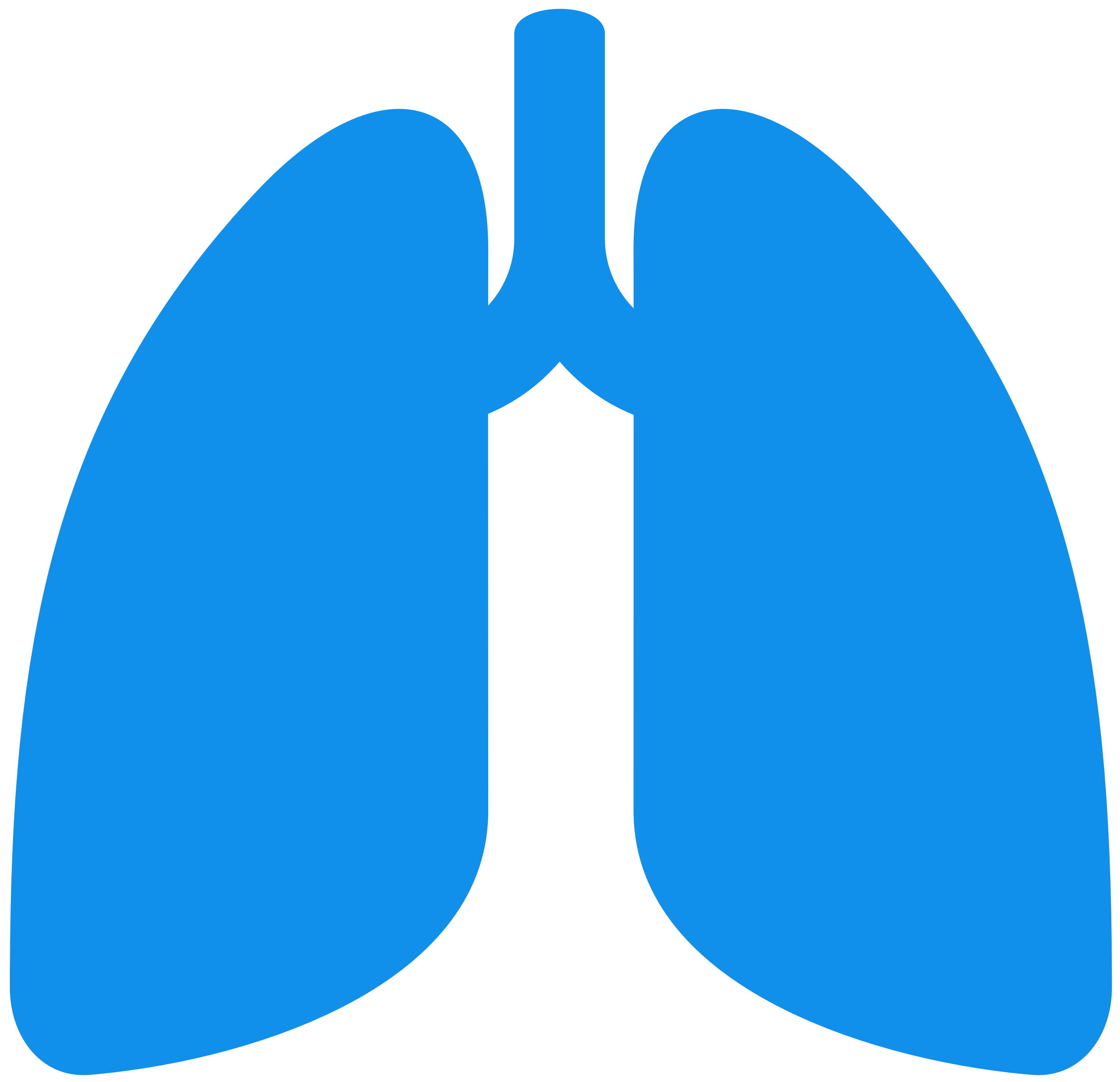
INCEPTION

XCEPTION

M.P. KUCHERA, ET. AL. NUCLEAR INSTRUMENTATION AND METHODS A (2019)

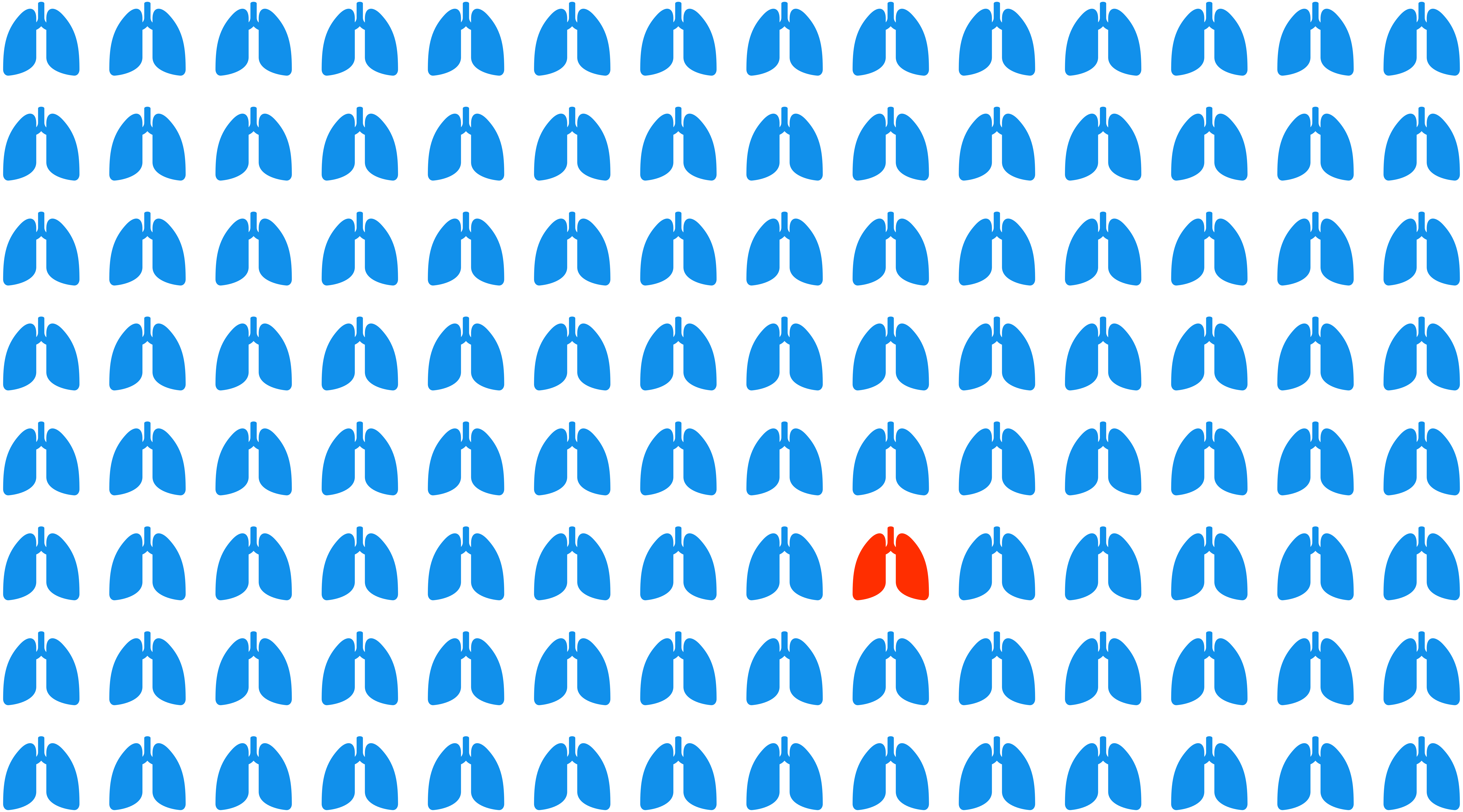
Application 2: Can we use machine learning to **accurately** classify events in detectors?

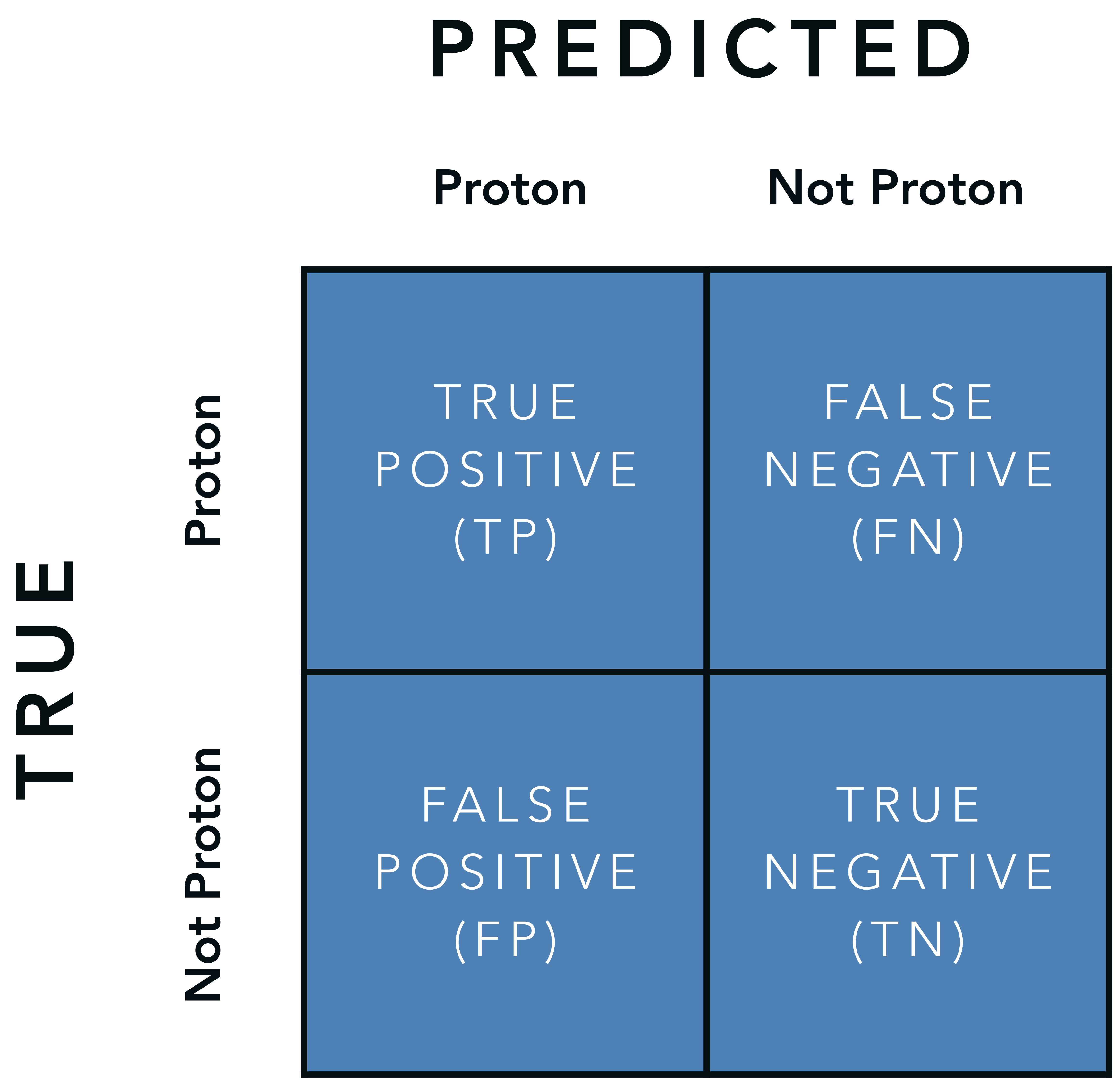
Metrics



Detect Lung Cancer

99% Accuracy




PREDICTED
Proton
Not Proton
TRUE
Proton
Not Proton
TRUE POSITIVE (TP)
FALSE NEGATIVE (FN)
FALSE POSITIVE (FP)
TRUE NEGATIVE (TN)

TRUE	PREDICTED: Proton	PREDICTED: Not Proton
Proton	TRUE POSITIVE (TP)	FALSE NEGATIVE (FN)
Not Proton	FALSE POSITIVE (FP)	TRUE NEGATIVE (TN)

$$\text{accuracy} = \frac{TP + TN}{TP + FN + FP + TN}$$

$$\text{precision} = \frac{TP}{TP + FP}$$

$$\text{recall} = \frac{TP}{TP + FN}$$

$$\text{F1} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

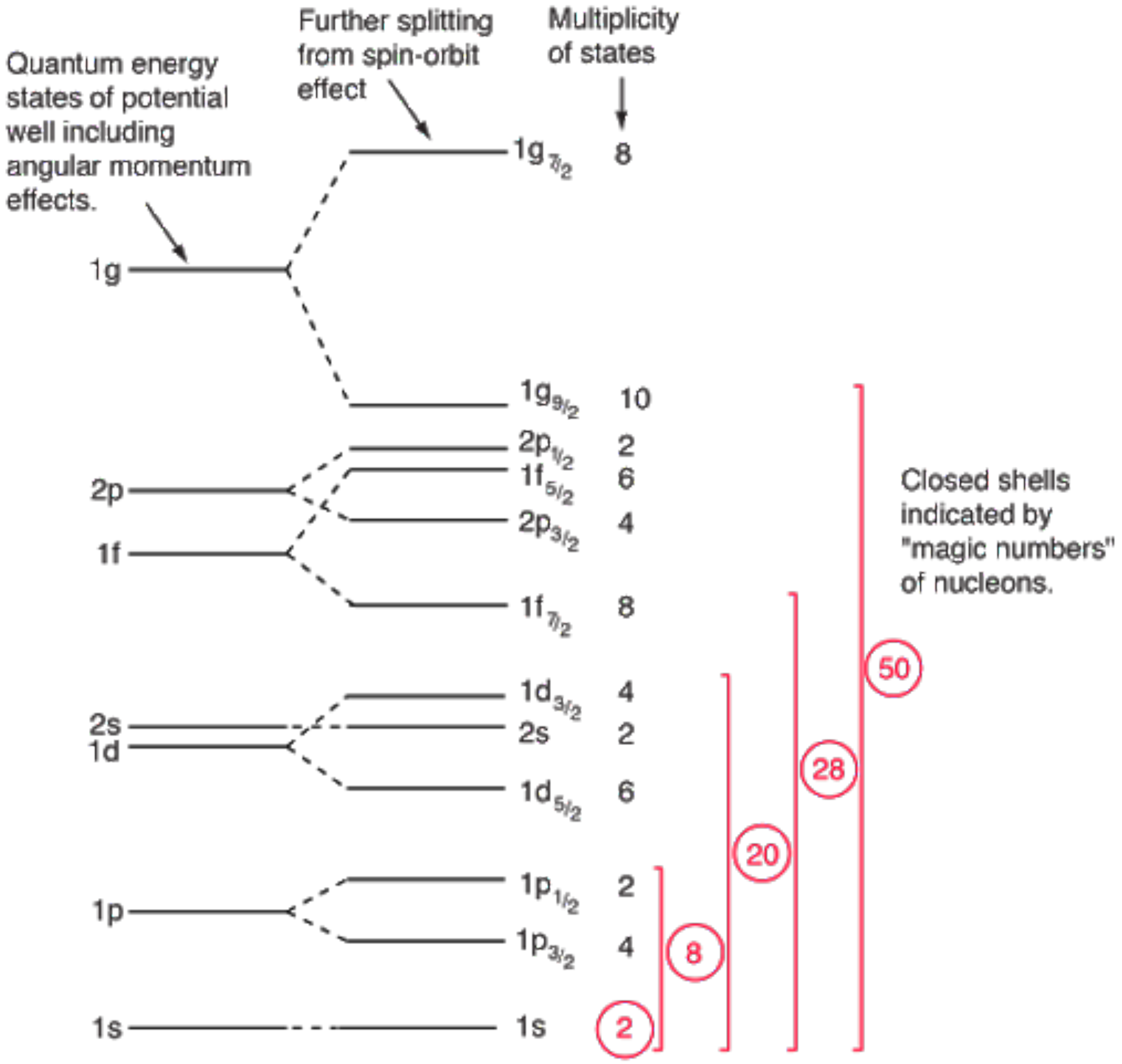
PREDICTED

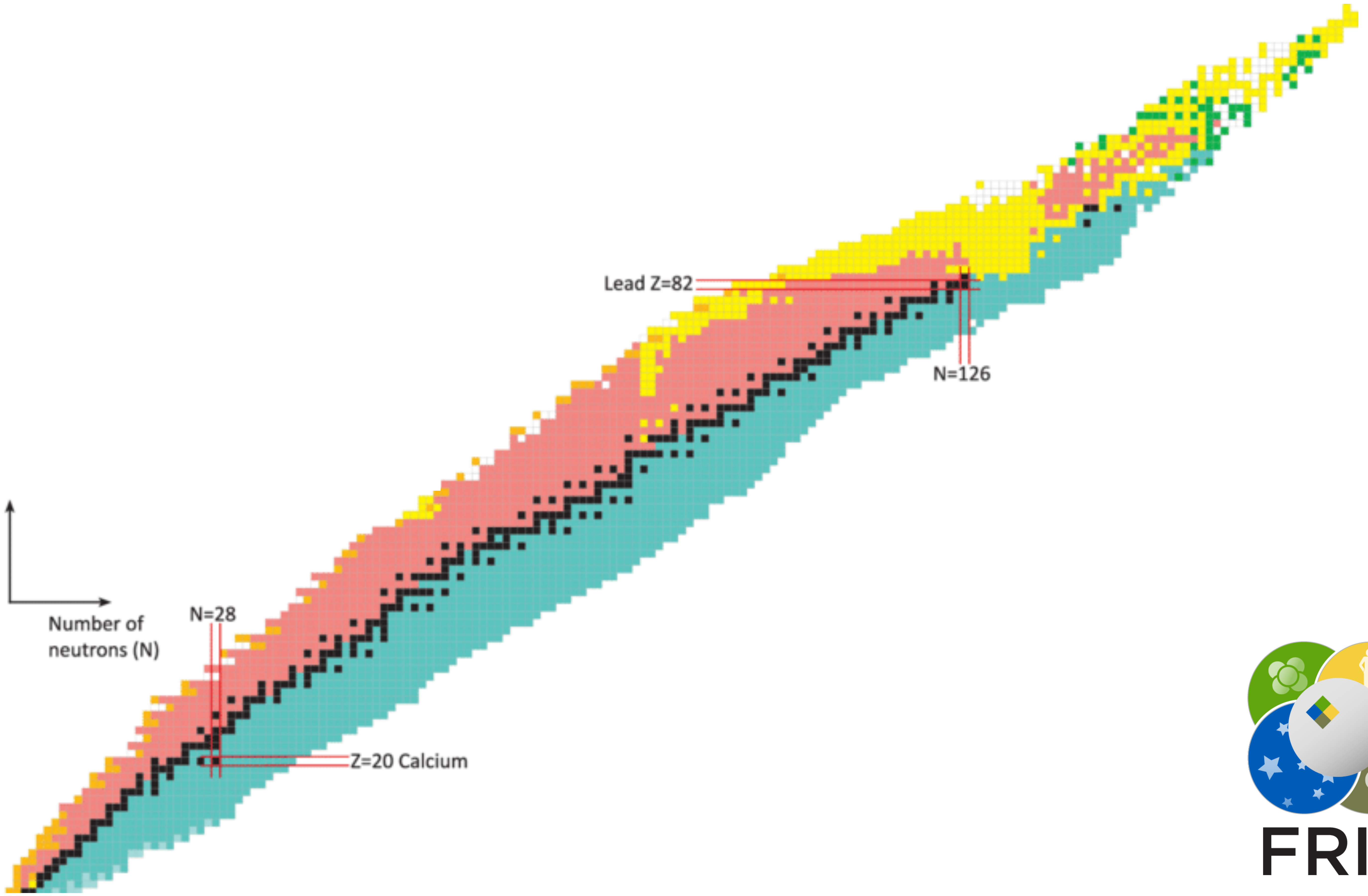
TRUE	Proton	Not Proton
Proton	TRUE POSITIVE (TP)	
Not Proton		TRUE NEGATIVE (TN)

PERFECT MODEL

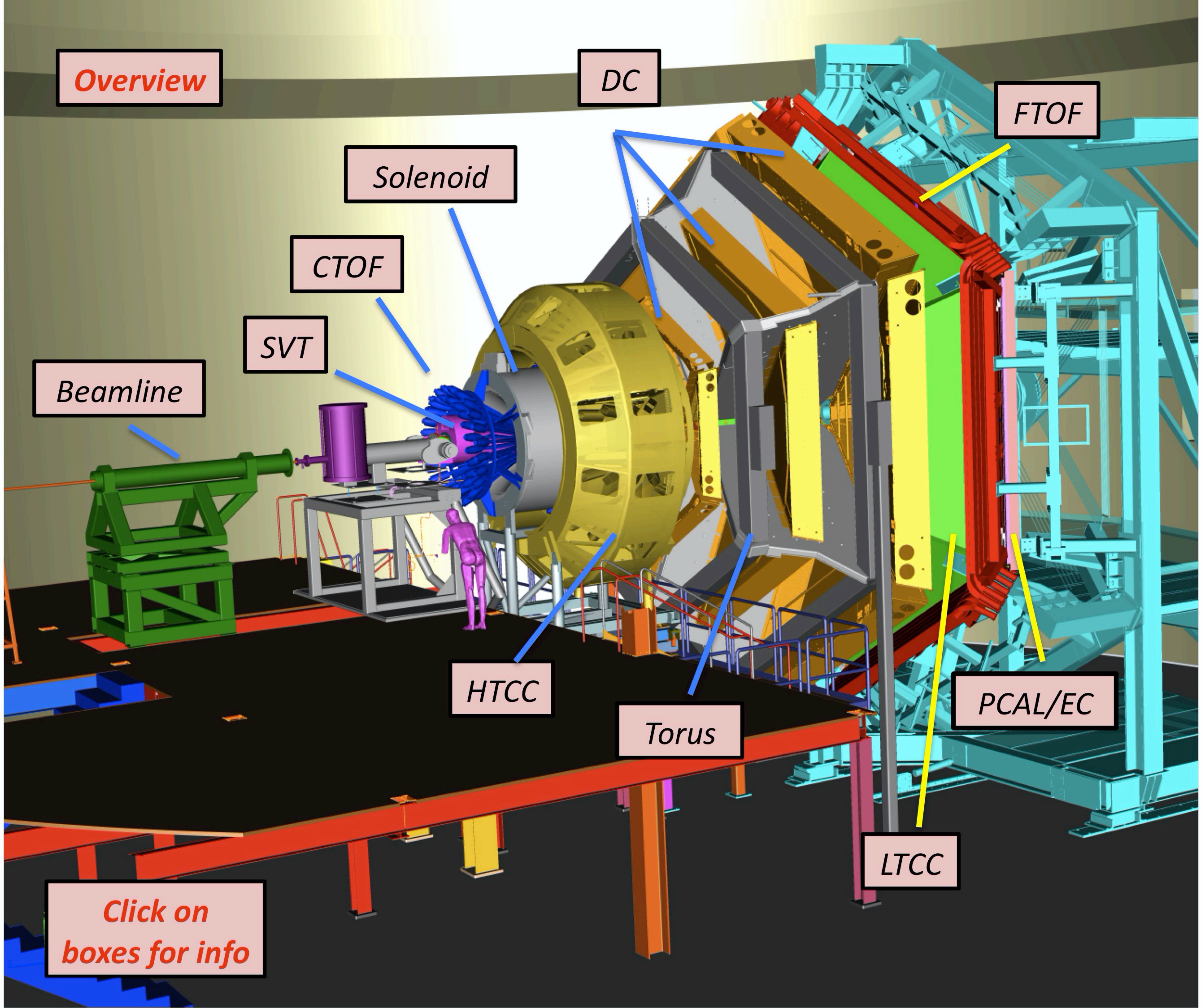
Application 2: Can we use machine learning to **accurately** classify events in detectors?

ACTIVE-TARGET TIME PROJECTION CHAMBER (AT-TPC)





FRIB

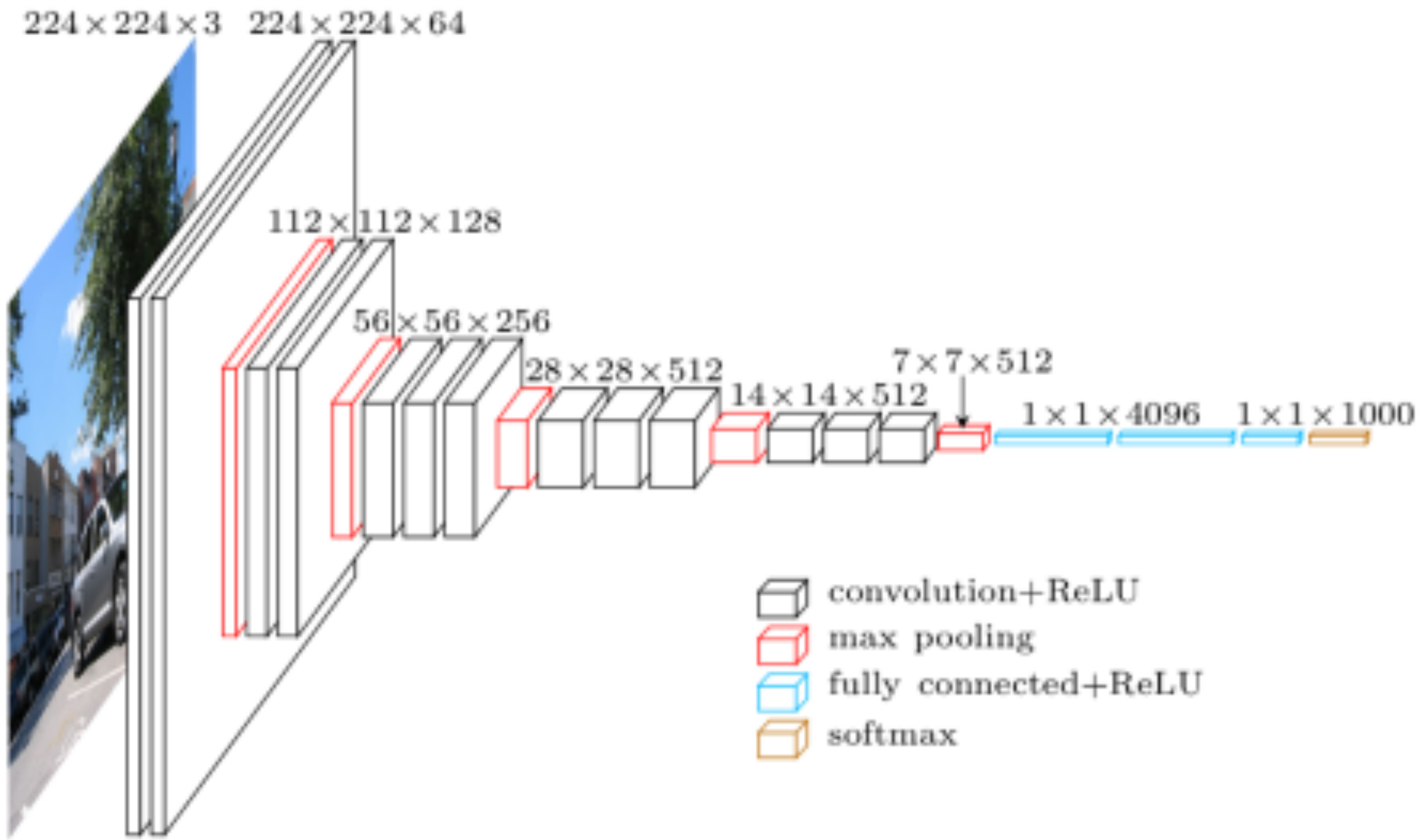

Overview
DC
FTOF
Solenoid
CTOF
SVT
Beamline
HTCC
Torus
PCAL/EC
LTCC
Click on boxes for info

EXPERIMENTAL DATA

AT-TPC

HALL B

VGG16 ARCHITECTURE



PRE-TRAINED ON IMAGENET DATA!

	AT-TPC			HALL B		
Experiment	Precision	Recall	F1	Precision	Recall	F1
Experimental → Experimental	0.96	0.90	0.93	0.97	0.93	0.95
Simulated → Simulated	1.00	1.00	1.00			
Simulated → Experimental	0.90	0.60	0.72			

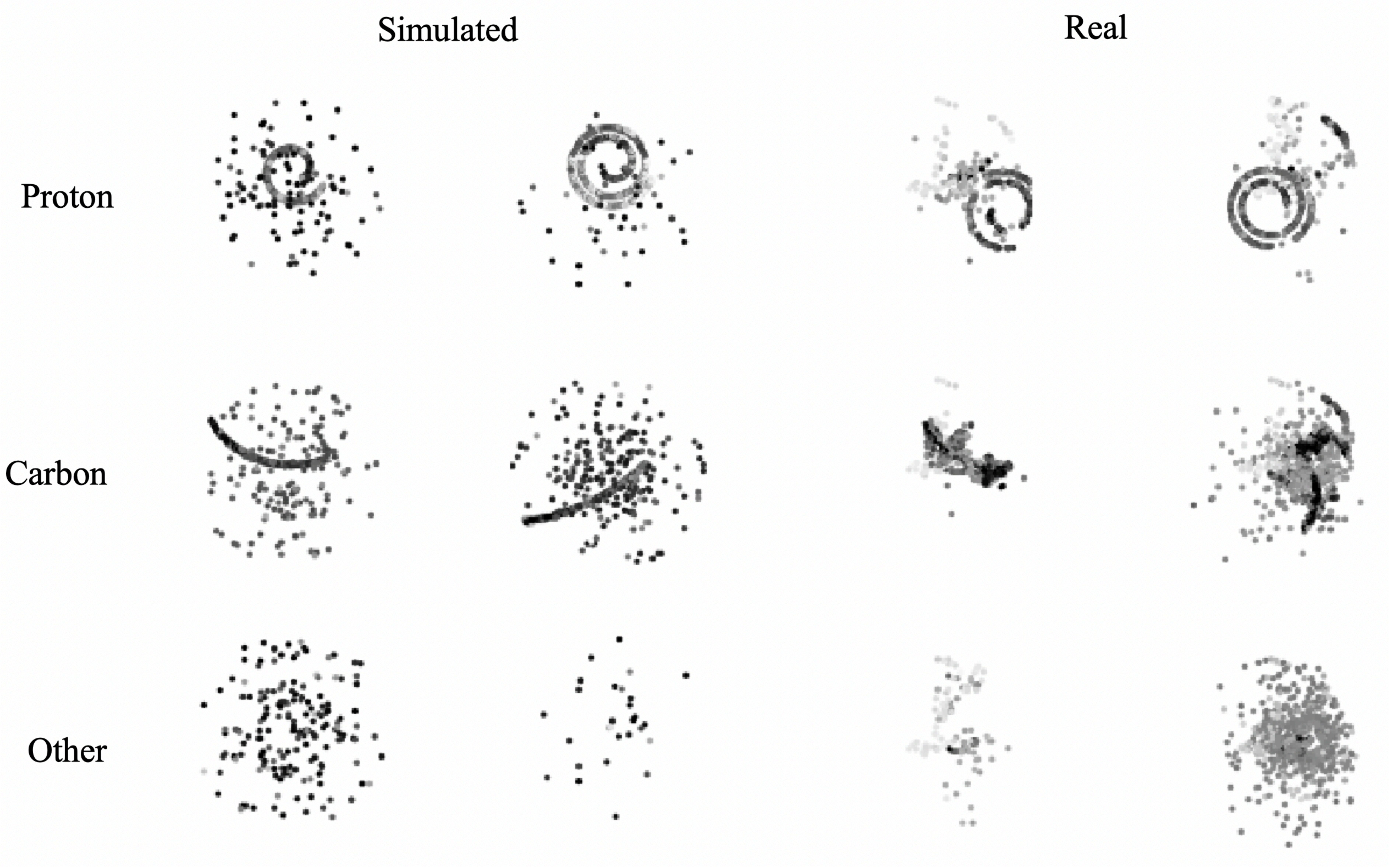
AT-TPC

Experiment	Precision	Recall	F1
Experimental → Experimental	0.96	0.90	0.93
Simulated → Simulated	1.00	1.00	1.00
Simulated → Experimental	0.90	0.60	0.72

HALL B

Precision	Recall	F1
0.97	0.93	0.95

6x faster!

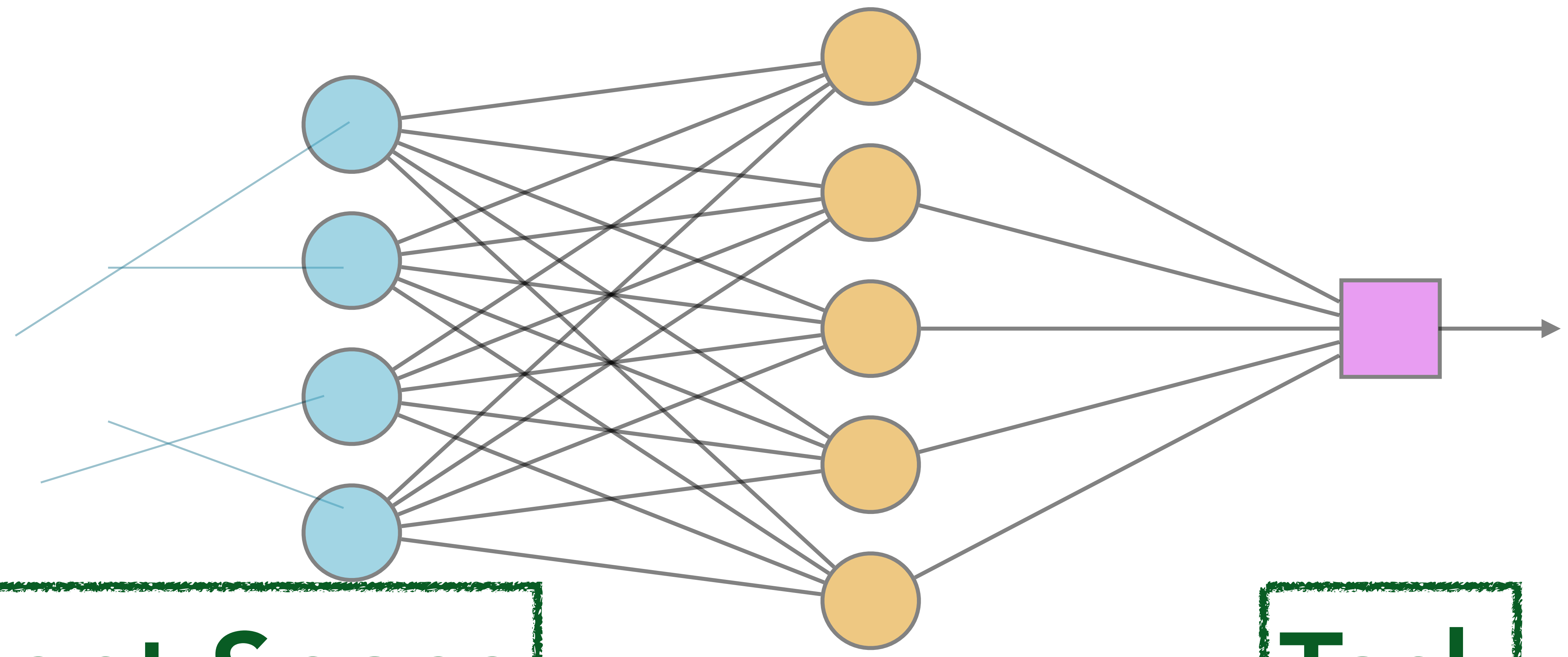


Simulated
Real
Proton
Carbon
Other

MACHINE LEARNING

UNSUPERVISED LEARNING

CONVOLUTIONAL NEURAL NETWORKS



Feature Extraction

Latent Space

Task

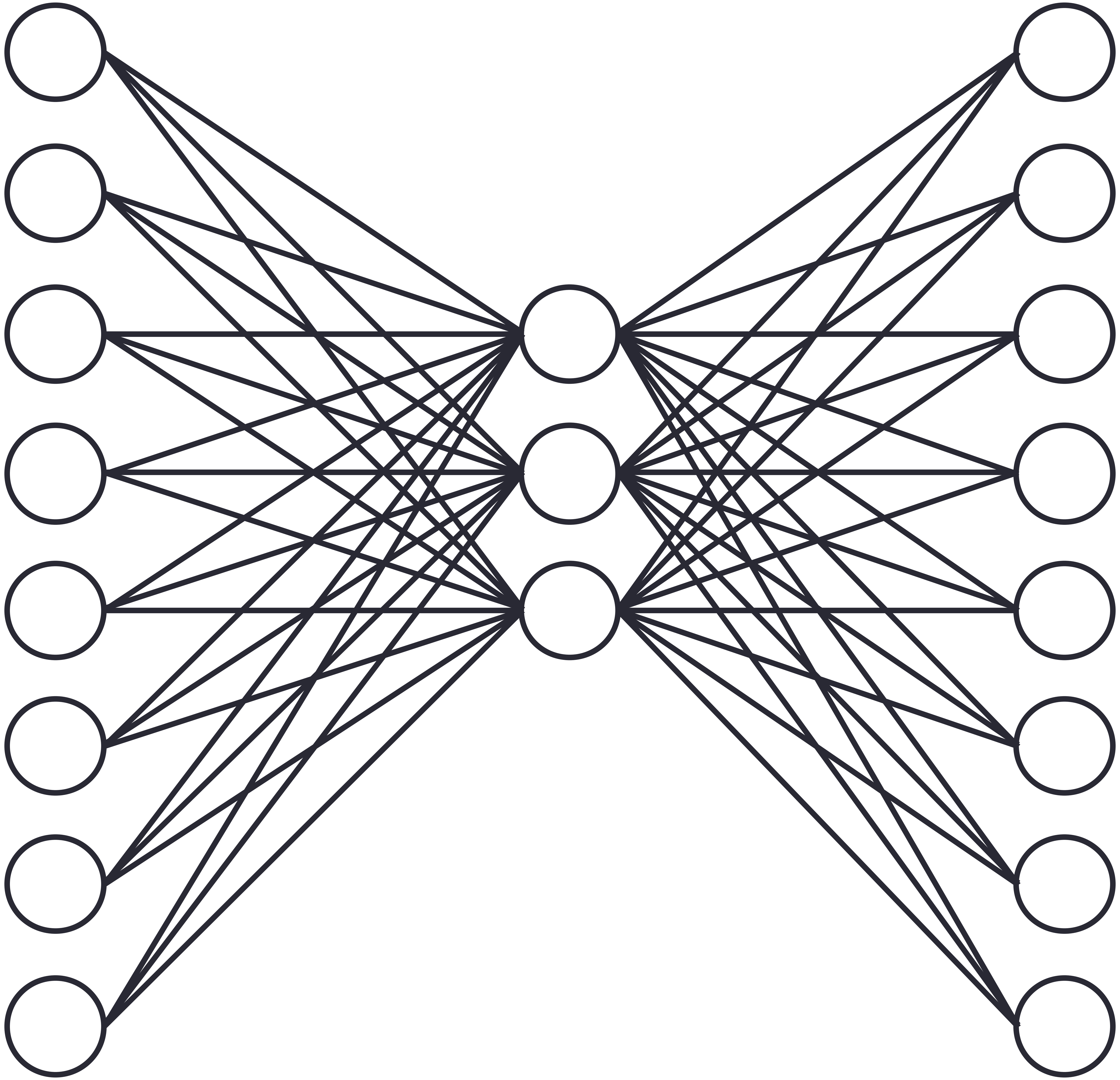
CLUSTERING — KMEANS

Goal: minimize pairwise distances between points in *same* cluster

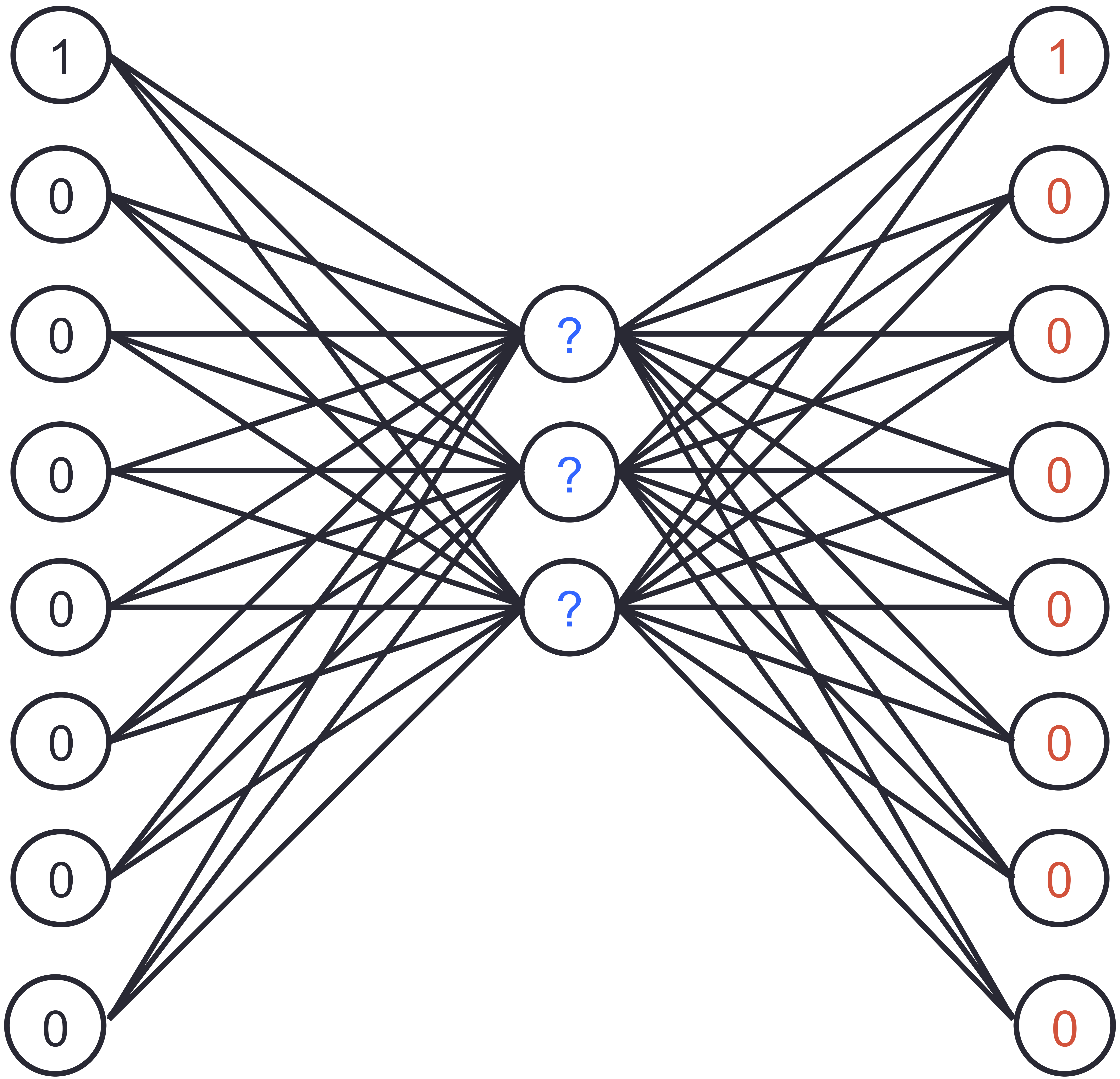
$$\min \sum_{i=1}^{k} \frac{1}{2N} \sum_{x,y,x\neq y}^{N} (\vec{x} - \vec{y})^2$$

Goal: maximize pairwise distances between points in *different* clusters

CLUSTERING — KMEANS



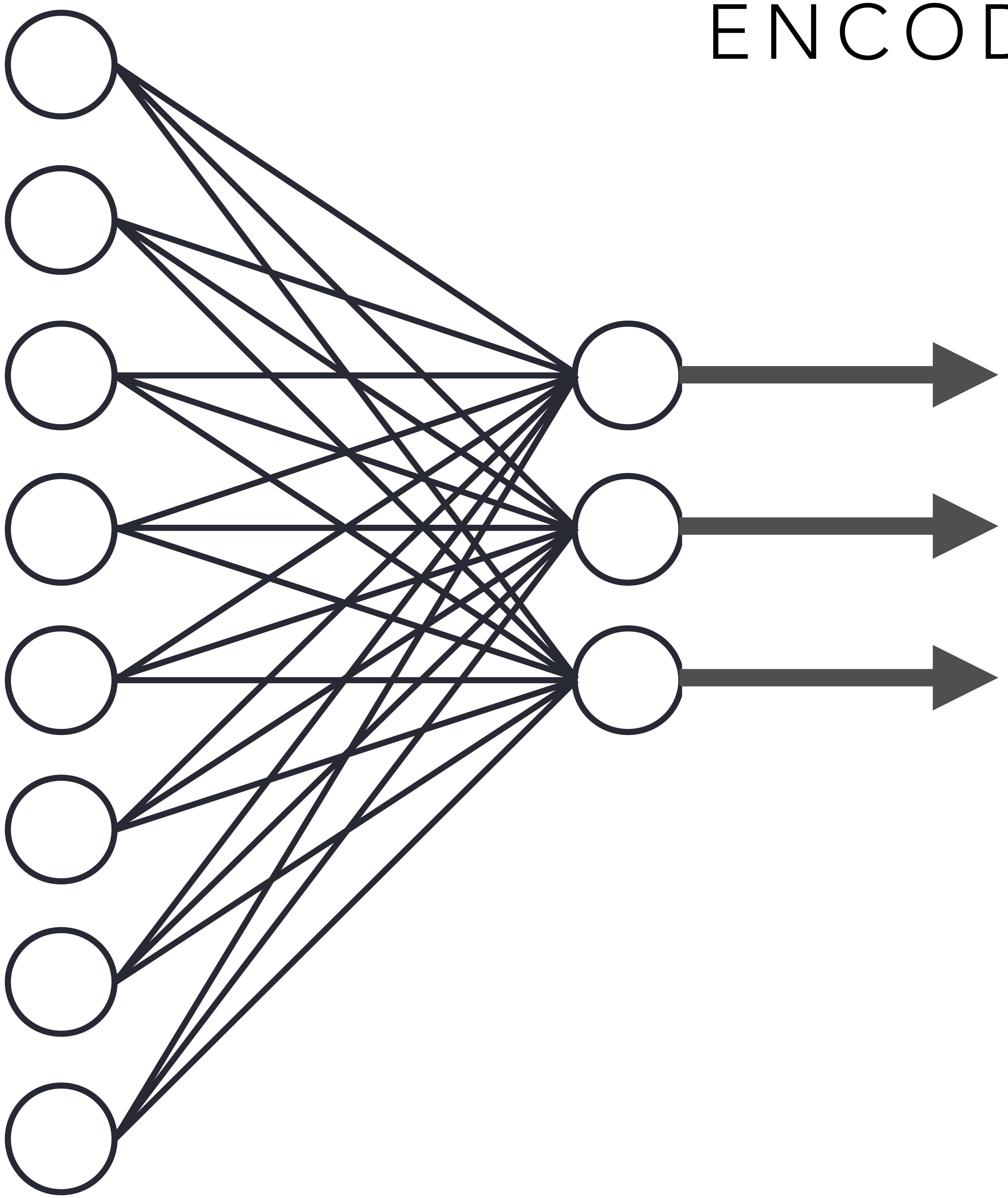
Input	Output
10000000	10000000
01000000	01000000
00100000	00100000
00010000	00010000
00001000	00001000
00000100	00000100


1
0
0
0
0
0
0
0
?
?
?
1
0
0
0
0
0
0
0

Input	Output
10000000	10000000
01000000	01000000
00100000	00100000
00010000	00010000
00001000	00001000
00000100	00000100

Input	A1	A2	A3	Output
10000000	0.9911	0.9869	0.0093	10000000
01000000	0.9892	0.0095	0.0124	01000000
00100000	0.0094	0.0283	0.0122	00100000
00010000	0.9840	0.9836	0.9900	00010000
00001000	0.0139	0.9904	0.0186	00001000
00000100	0.0128	0.9805	0.9868	00000100

ENCODER

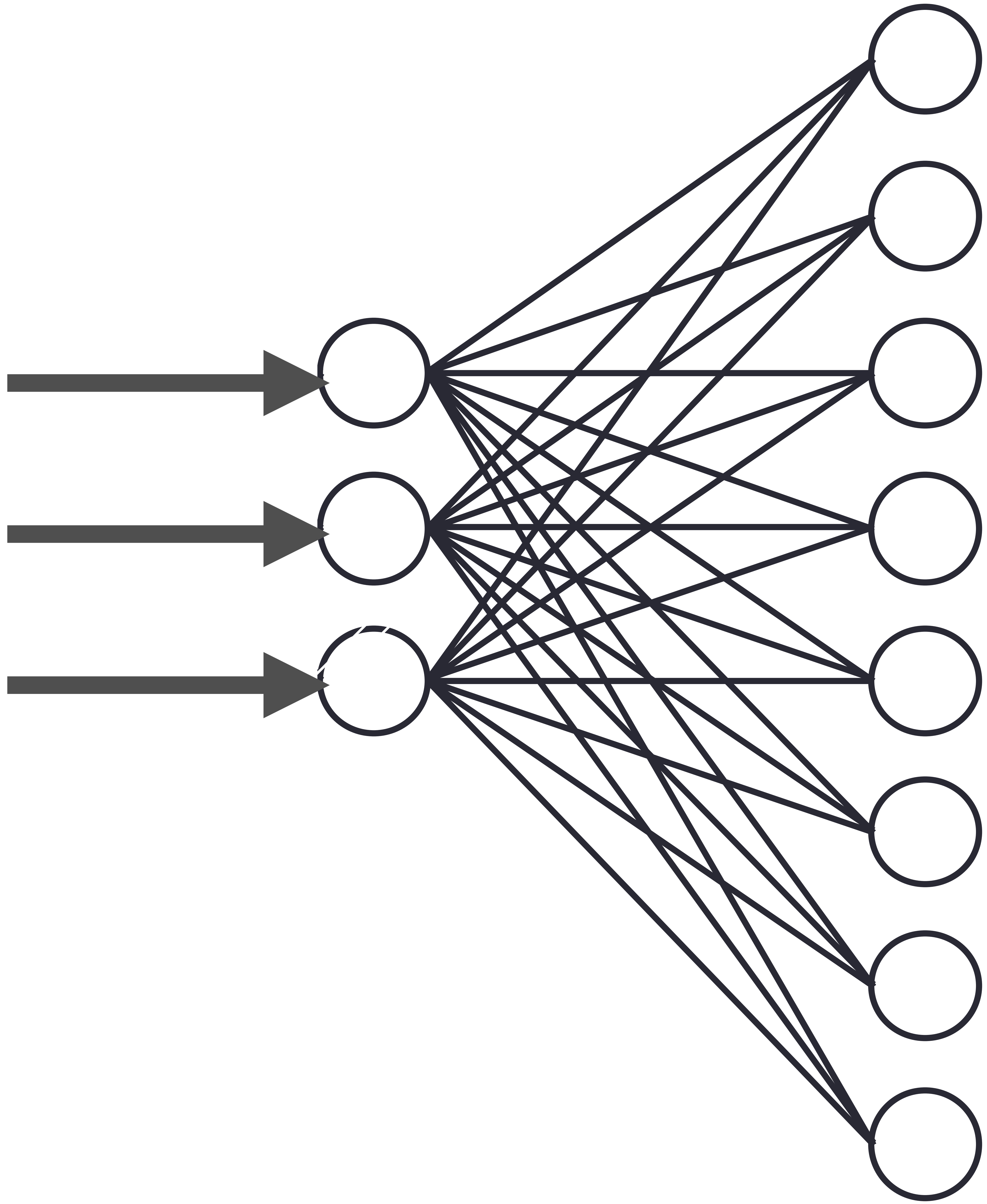


GENERATIVE MODELS

MICHELLE KUCHERA
DAVIDSON COLLEGE

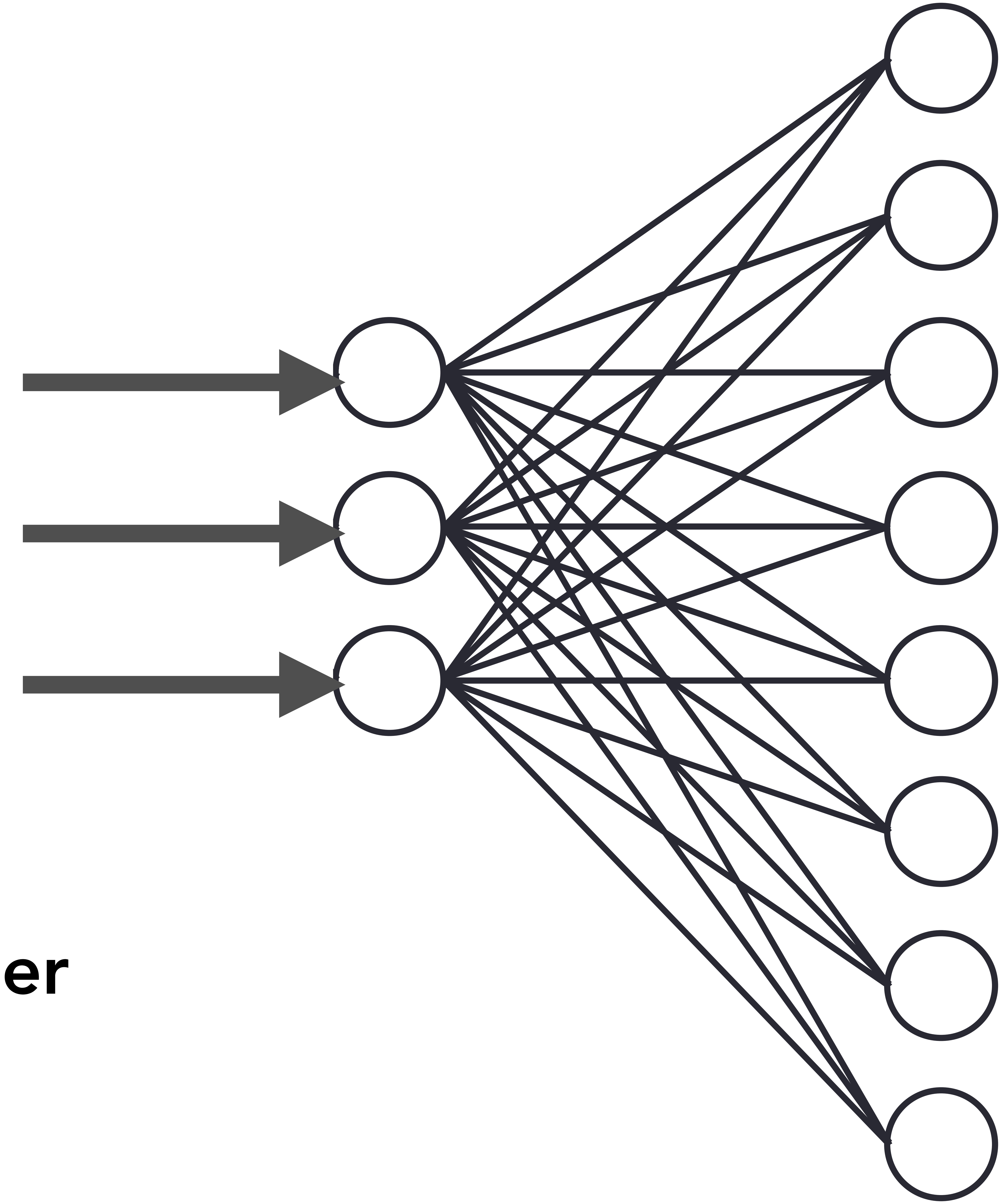
ECT* TALENT SUMMER SCHOOL
02 JULY 2020

DECODER



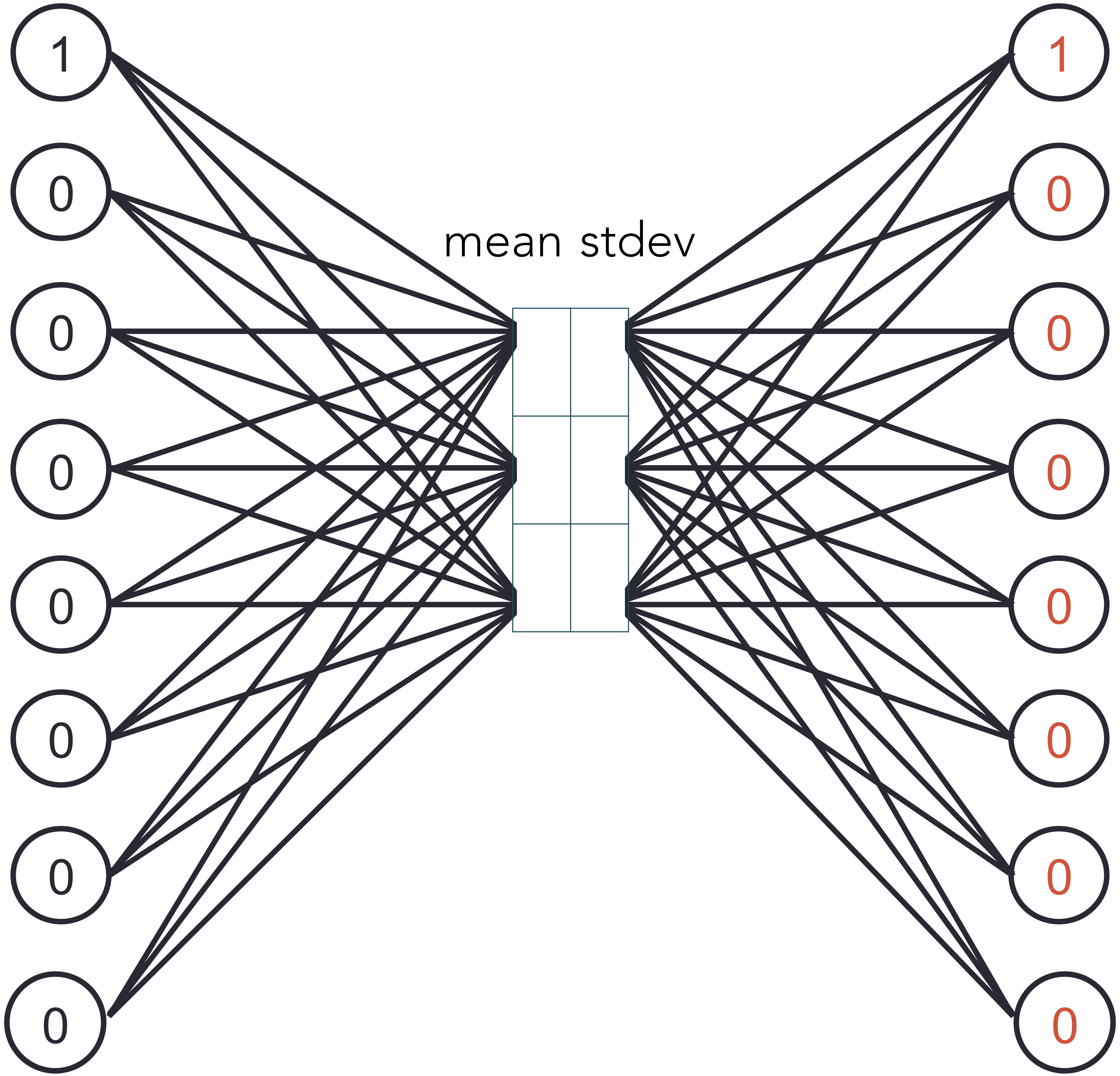
DECODER

How do we know that we are providing a latent vector that represents those seen in training?

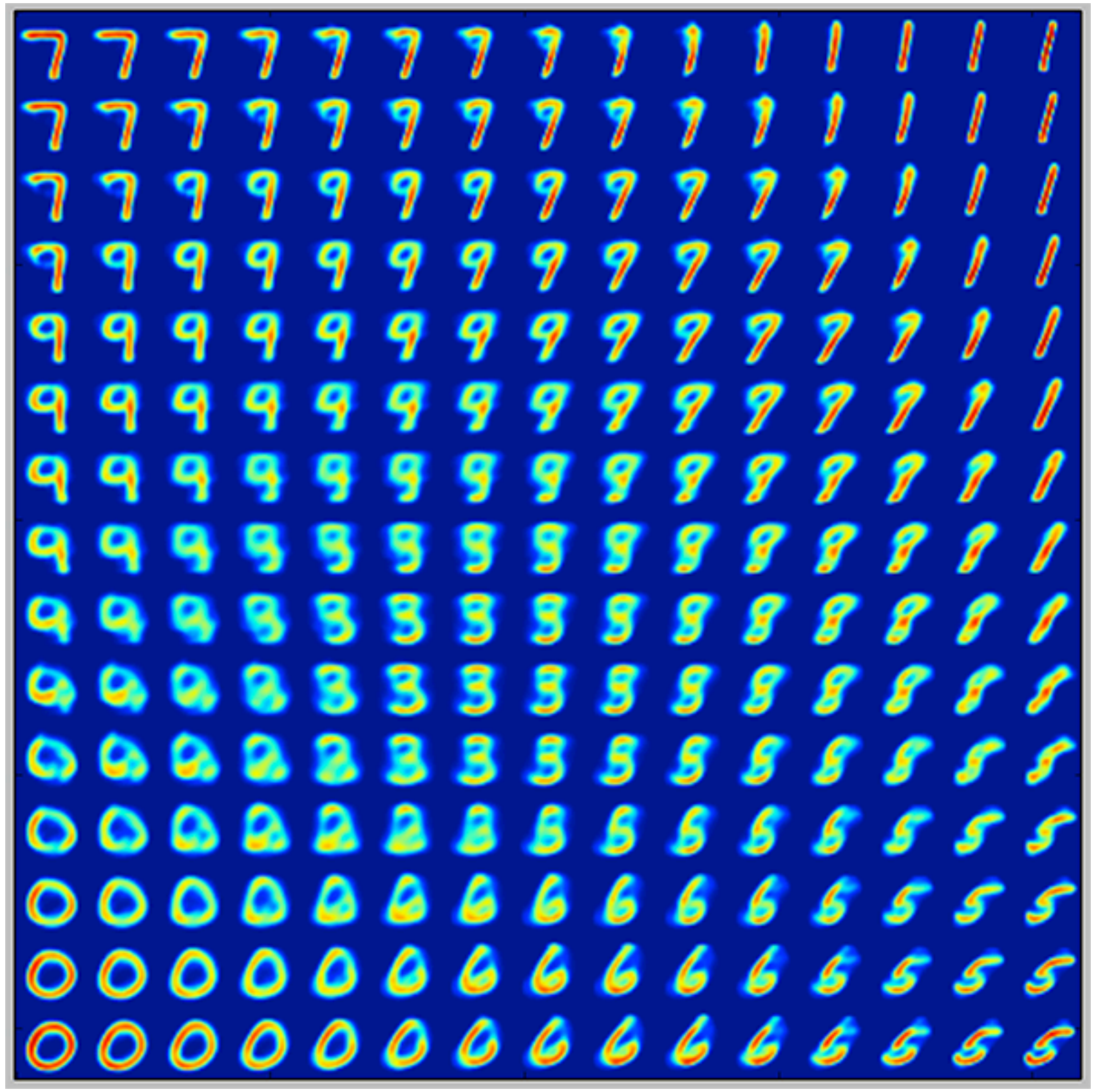


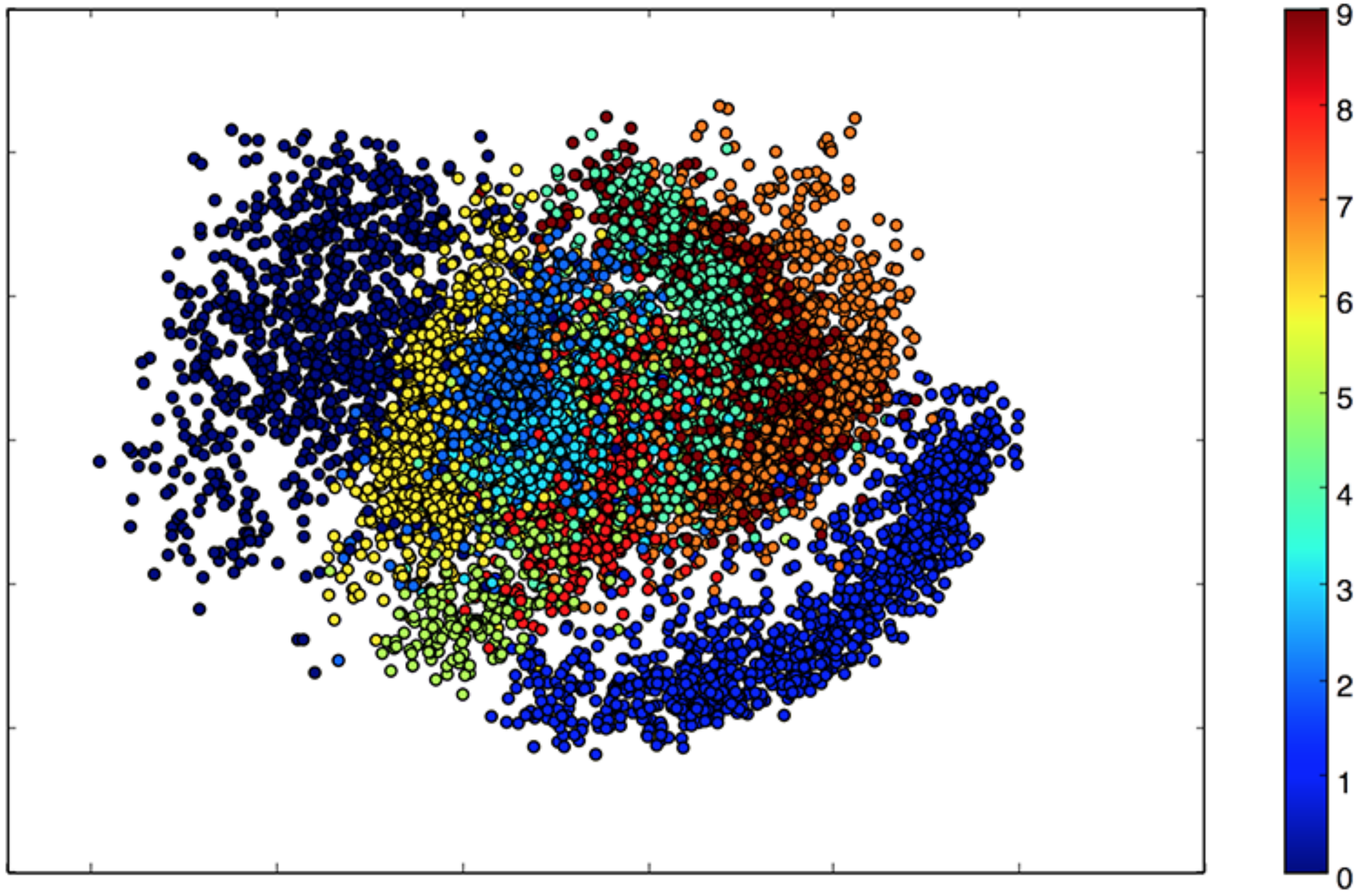
Variational Autoencoder

Encode to two outputs for each latent dimension: mean and stdev



Sample similar points in latent space, decode, and compare with regularization



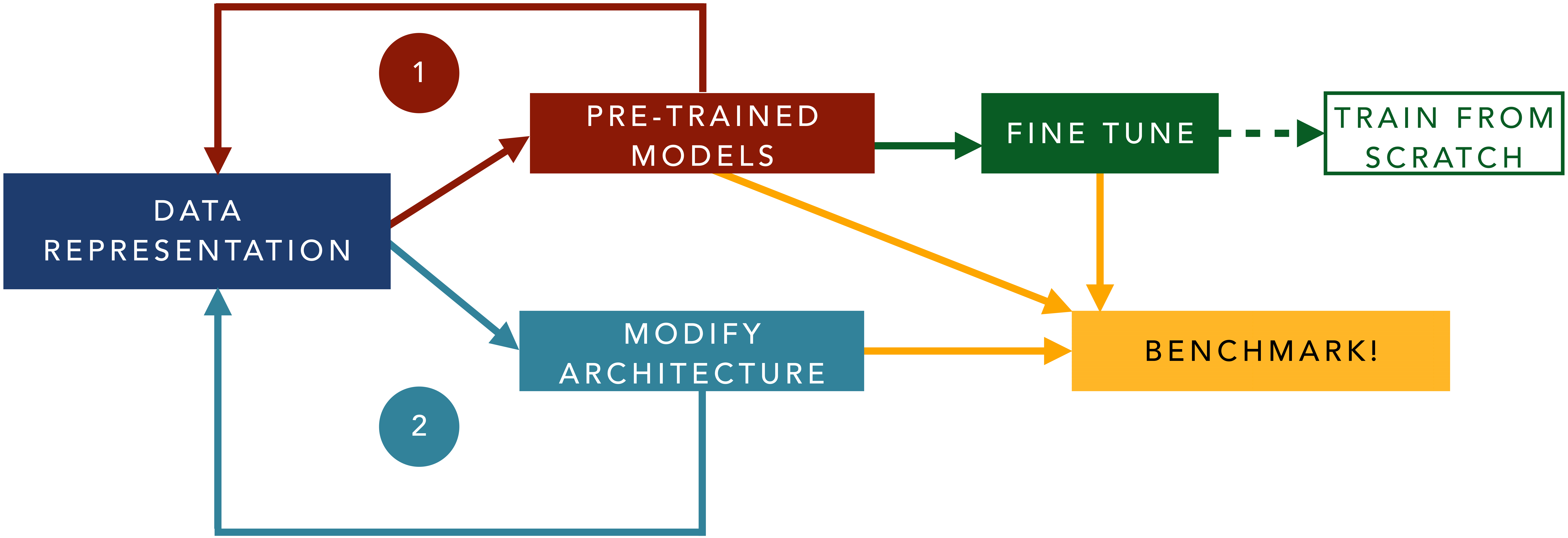


https://blog.keras.io/building-autoencoders-in-keras.html

GENERATIVE MODELS

SIMULATION

EXAMPLE WORKFLOW



EXAMPLE WORKFLOW