Connecting Dots:
An AI Cookbook for Nuclear Physics

Aobo Li
Halıcıoğlu Data Science Institute
Department of Physics
UC San Diego

National Nuclear Physics Summer School, 07/22/2024

Why Do I Want To Give This Lecture

There is a “Gap” between Nuclear Physics and AI/ML

- AI/ML is a huge field with many different research directions
- As physicists, we prefer to approach problem in the physics way, but there is also an “AI/ML way” for the same problem

Lecture 1 sets up the foundation to understand more advanced AI/ML concepts

- Lots of technical details in Lecture 1, but not this lecture

- **Main Objective:** Connecting dots between AI/ML research directions and NP
  - “Not with all details, but I know there is an existing AI/ML methods that could solve my problem”

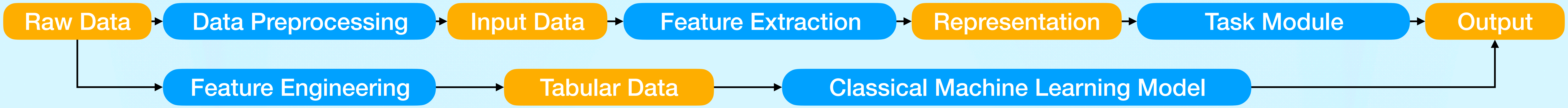
Nuclear Physics

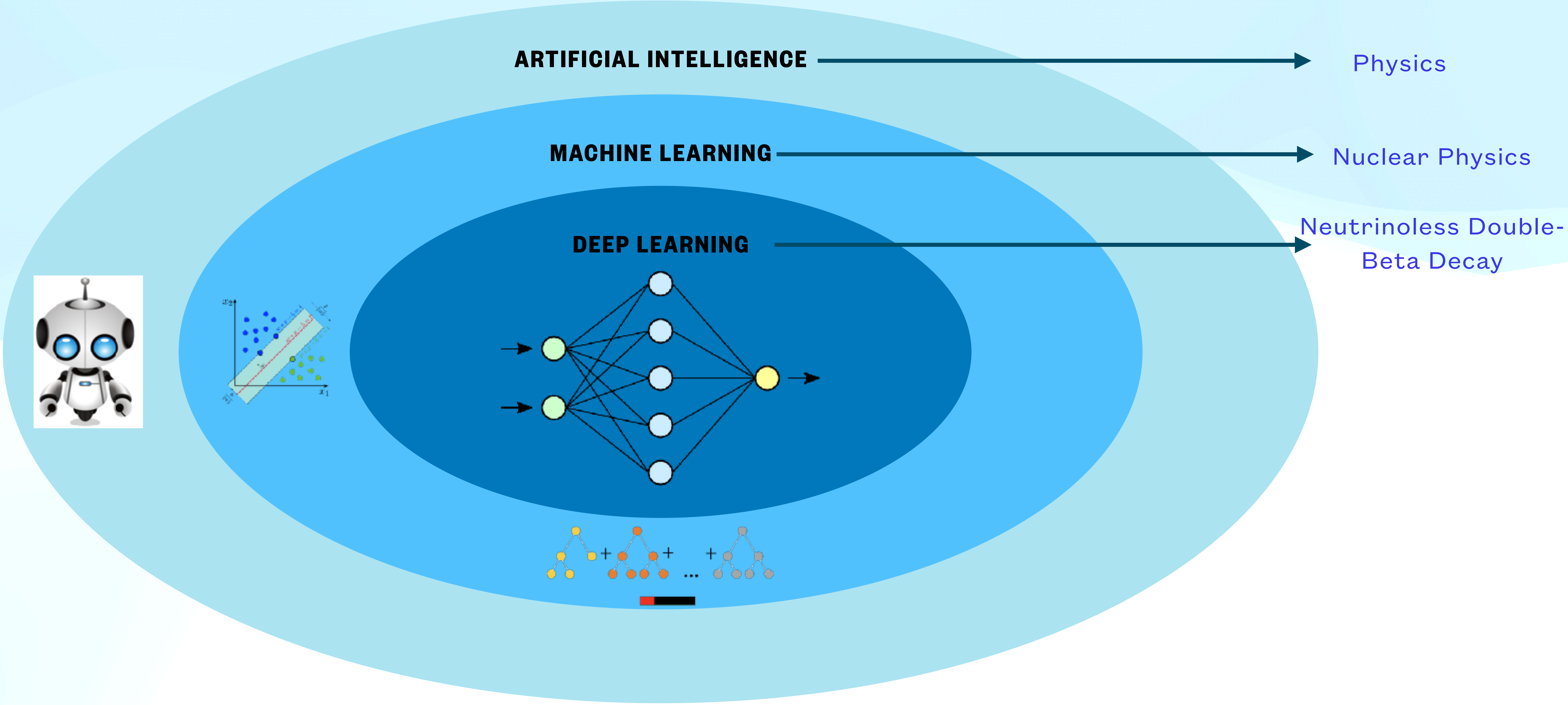
Questions I received from nuclear physics students

AI/ML

Research directions in AI/ML that could help solving NP challenges


1. Majorana Demonstrator Dataset and HPGe Detector
2. PyTorch Dataset Class
3. Data pre-processing: transform your data
4. Wrap dataset object to create a data loader
1. Task Module: Task Layer + Loss Function
2. Cross Entropy
3. Binary Classification 1-3
4. Multiclass Classification
1. Stochastic Gradient Descent
2. Backward Pass: Computational graph and Backpropagation
3. PyTorch implementation: one line
1. Data Preprocessing
2. Feature Extraction
3a. Classification
3b. Regression
4. Training
5. Evaluation
Model Building
1. Neural Network: linear layer and activation function
2. How to create a simple NN in PyTorch
3. Forward pass
MSE Loss, L1 Loss, and Smooth L1 Loss
1. Overfitting vs. Underfitting
2. Evaluate binary classication: ROC Analysis
3. Evaluate regression: reconstruction algorithm
Raw Data
Data Preprocessing
Input Data
Feature Extraction
Representation
Task Module
Output
Feature Engineering
Tabular Data
Classical Machine Learning Model


Raw Data
Data Preprocessing
Input Data
Feature Extraction
Representation
Task Module
Output
Feature Engineering
Tabular Data
Classical Machine Learning Model

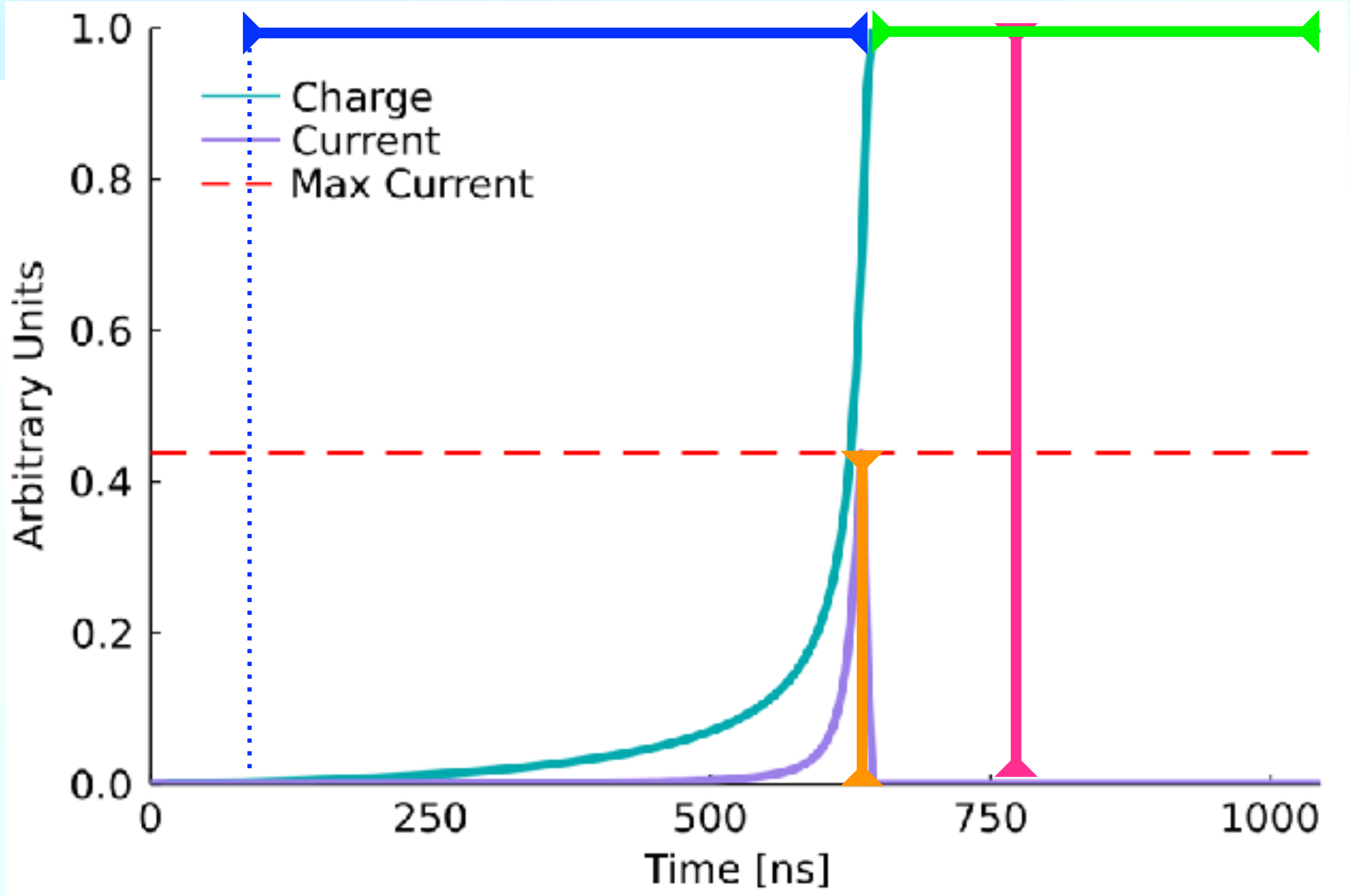

ARTIFICIAL INTELLIGENCE
MACHINE LEARNING
DEEP LEARNING
Physics
Nuclear Physics
Neutrinoless Double-
Beta Decay

Nuclear Physics

Q1: In Lecture 1, we started from raw MAJORANA DEMONSTRATOR waveforms, the lowest level of HPGe detector. Do we always have to start from **low level data**?

AI/ML

No, we can start from higher level parameters with a procedure called **Feature Engineering**



Single-Site Waveform

Tail Slope

For **surface background** rejection

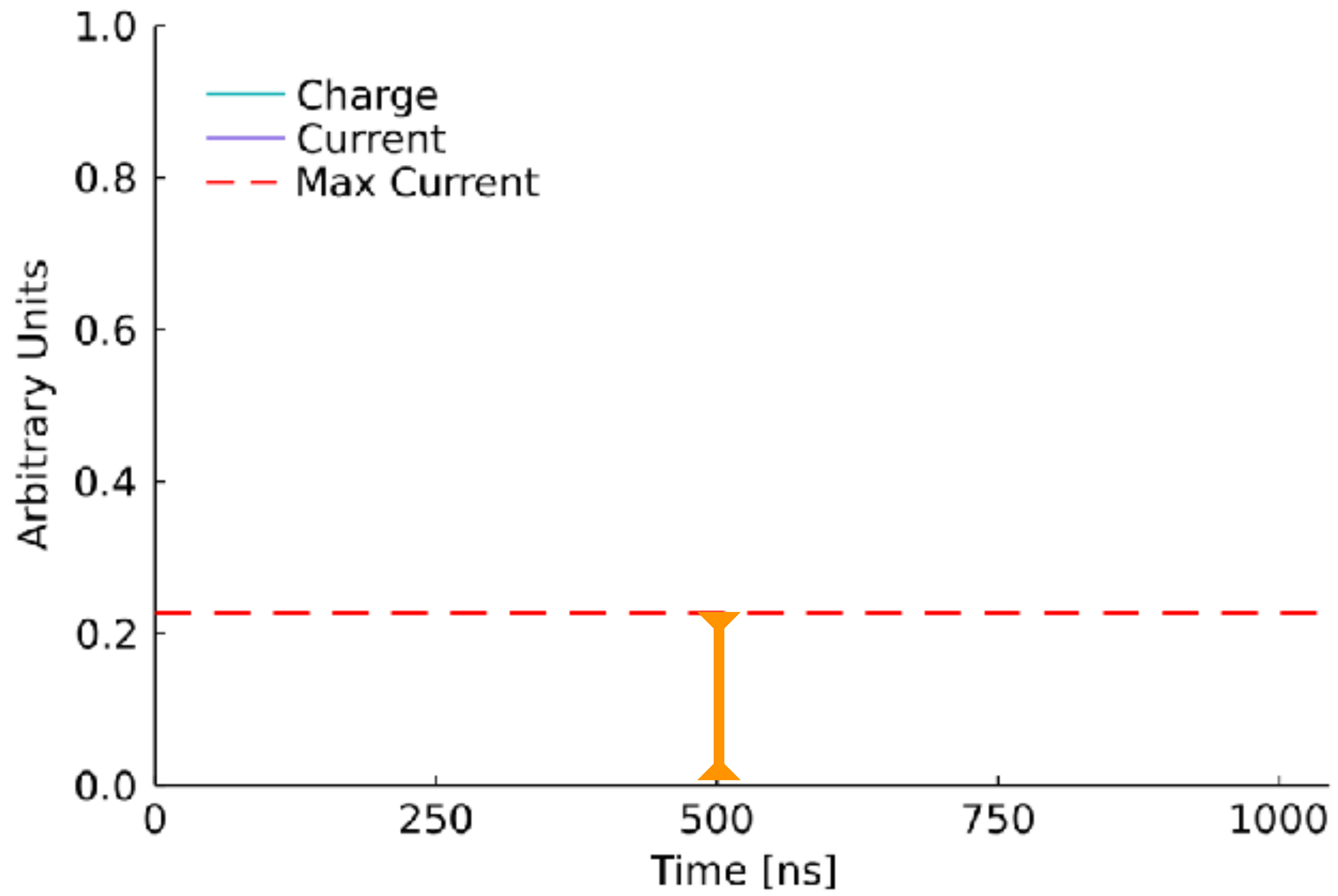
Energy

AvsE

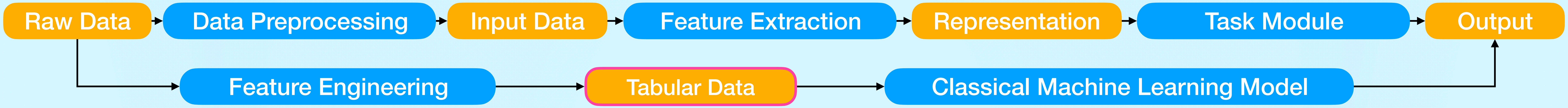
For **multi-site background** rejection

Drift Time

Reflect the location of incident particle



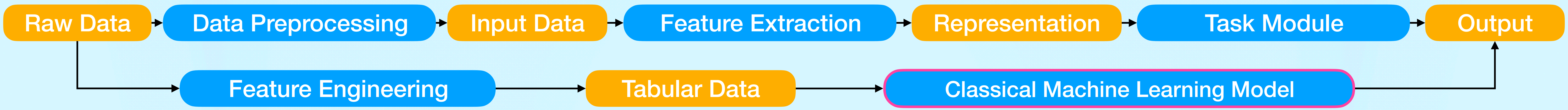
Multi-Site Waveform



	Tail Slope	Energy	AvsE	Drift Time
Waveform 1				
Waveform 2				
Waveform 3				
……				
Waveform 65,000				

Tabular Data

- Usually has **much lower dimension** than raw data
- Obtained through **Feature Engineering** process
  - Extracting useful/representative informations into a few quantitative parameters
  - **Prior knowledge** can be incorporated during this process
  - This means our understanding of **Nuclear Physics** can be incorporated

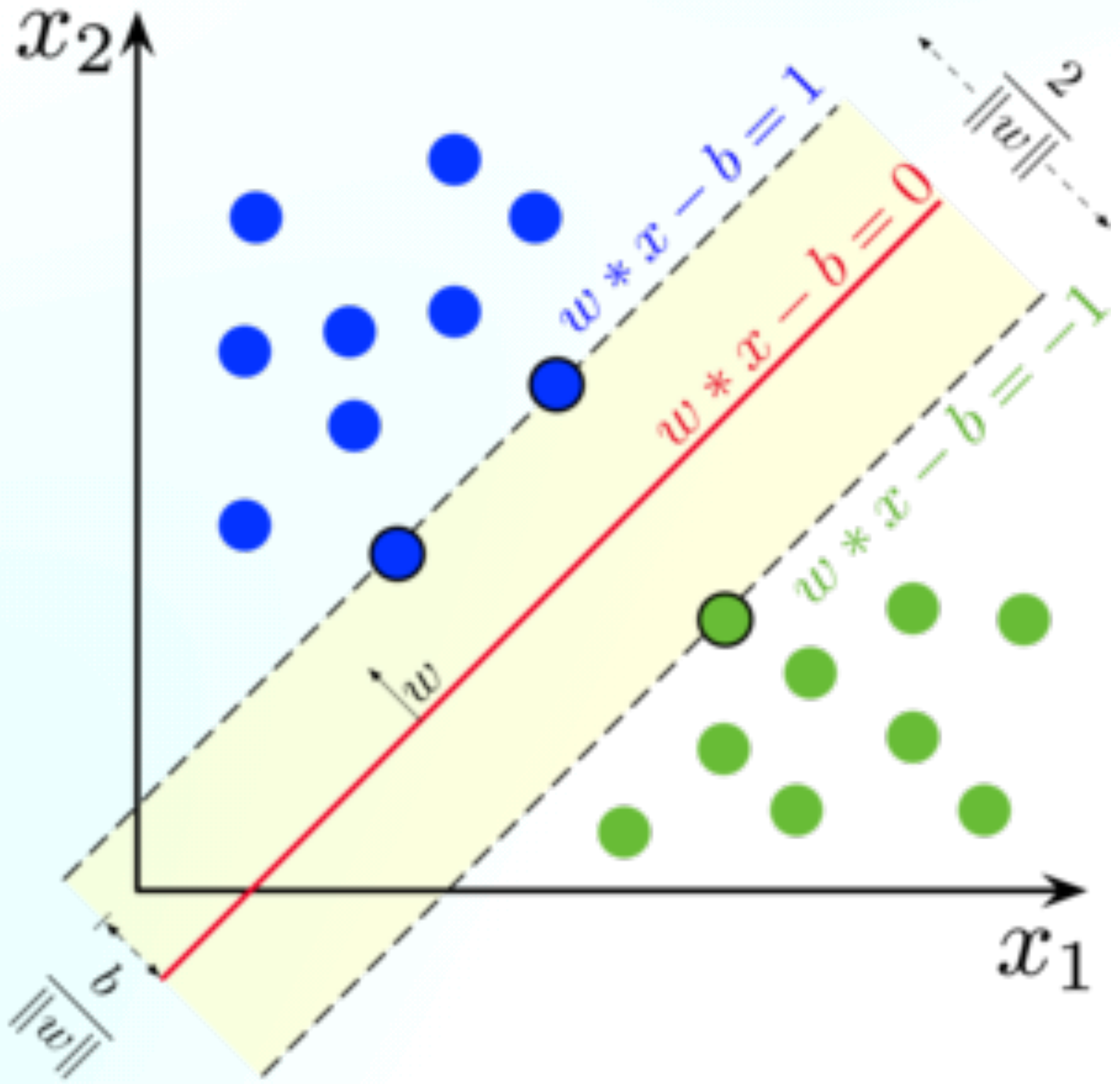


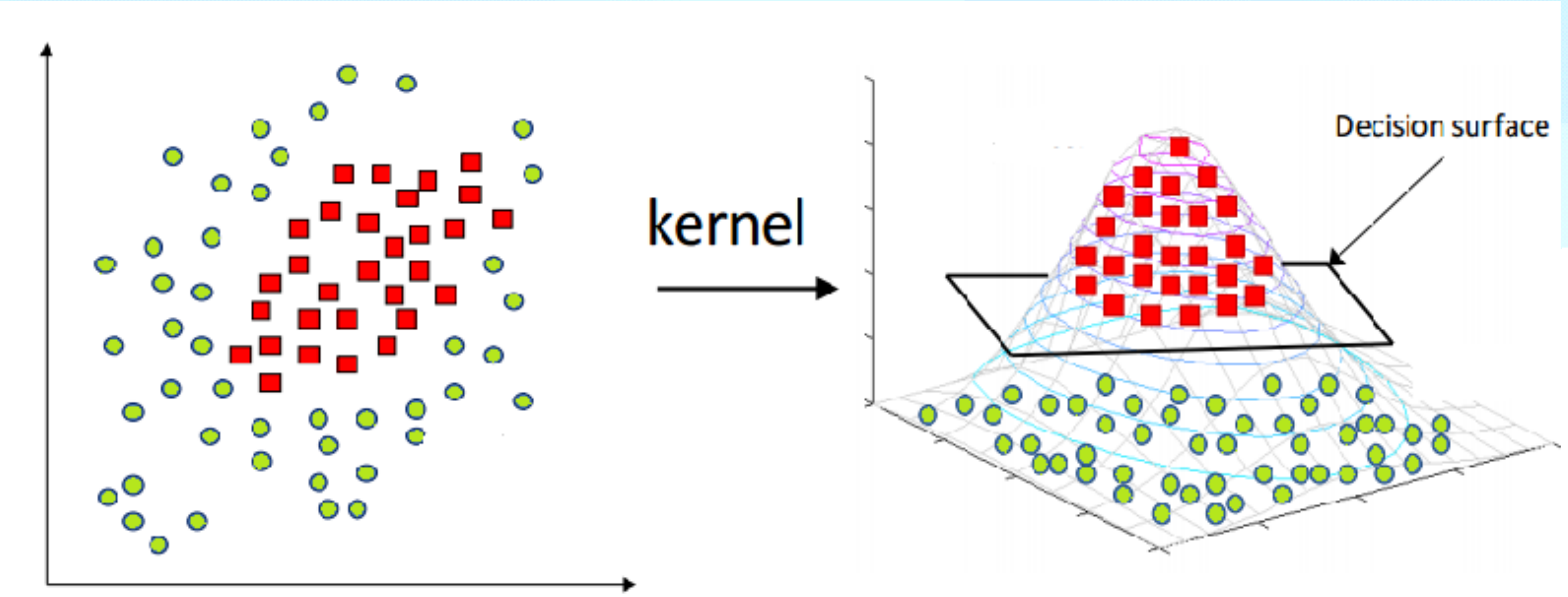
<u>Deep Neural Networks can be used to analyze Tabular Data, but it's usually not the best model…</u>

- DNN is particularly powerful for high dimensional data, but Tabular data is usually low dimensional
- DNN lacks some very useful features some other models have

Support Vector Machine (SVM)

- Draw a **hyperplane** between two clusters of tabular data
- Maximize the **margin** between hyperplane and the **support vector** (closest data point to the hyperplane)





Advantages of SVM

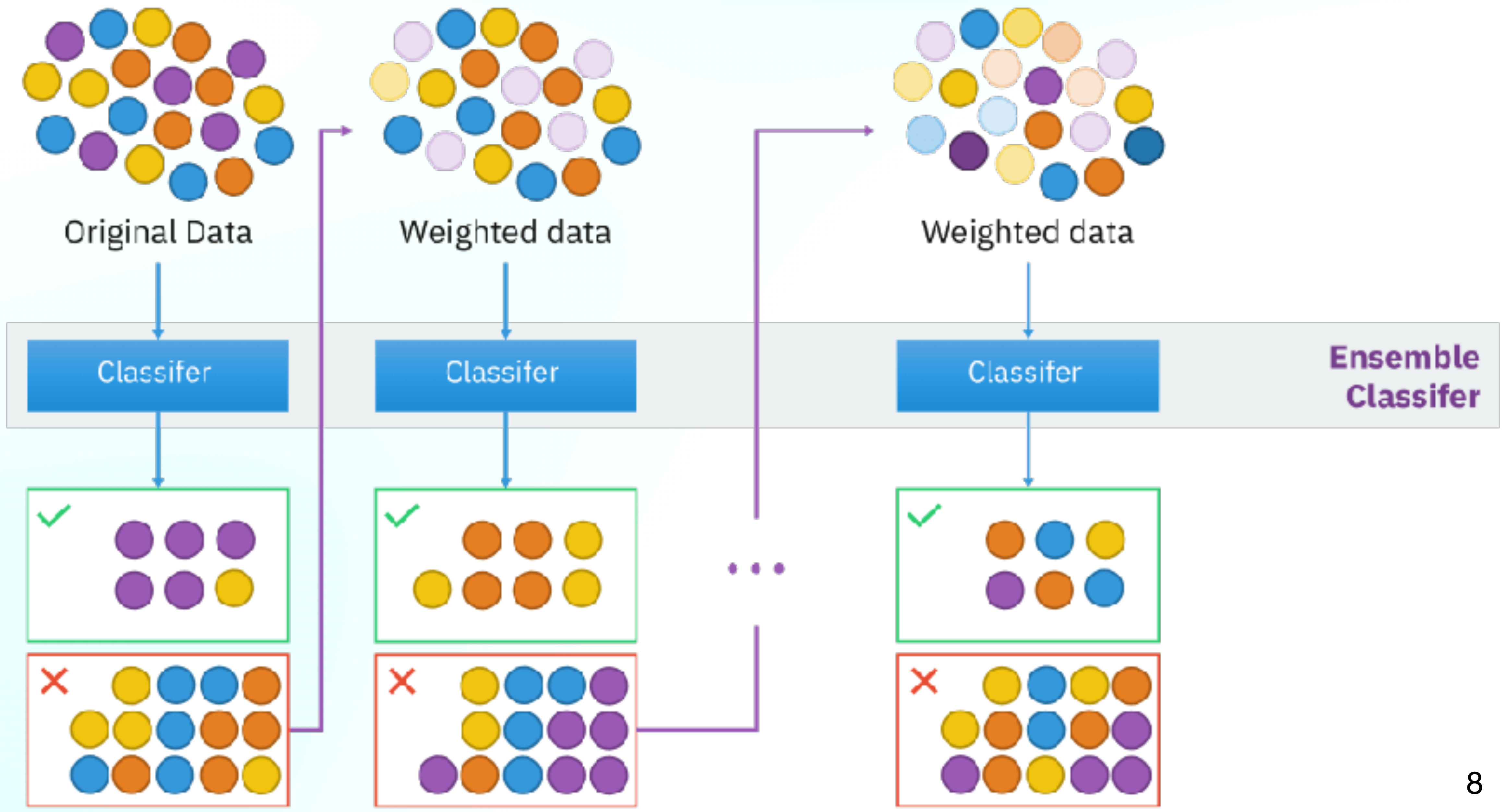
- Very clear and analytical **Decision Boundary** between signal and background
- Unlike DNN which is data-hungry, SVM is robust with **small amount of training data**
- **Kernel method** could transform the data into a space where they are linearly separable

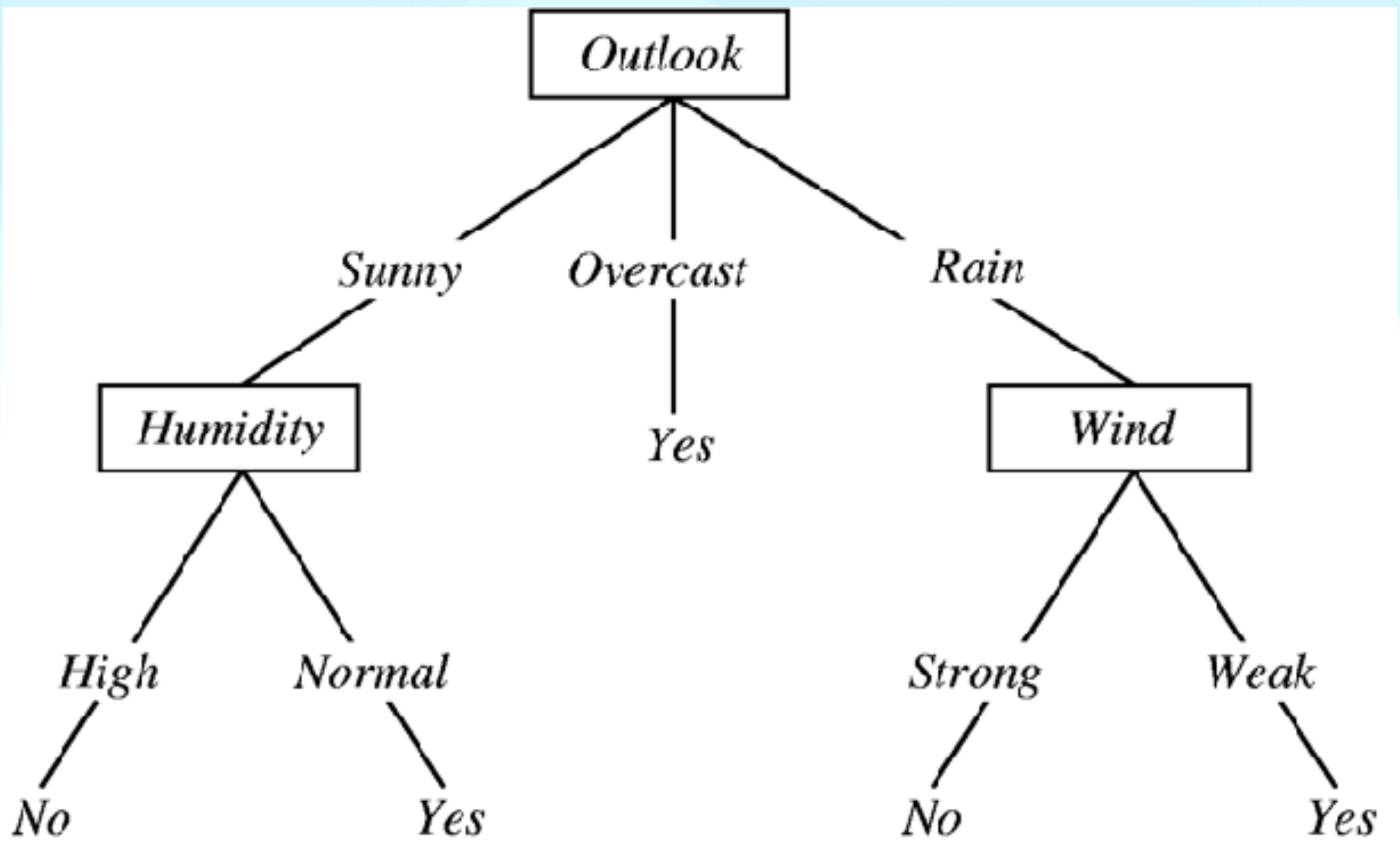
Deep Neural Networks can be used to analyze Tabular Data, but it's usually not the best model…

- DNN is particularly powerful for high dimensional data, but Tabular data is usually low dimensional
- DNN lacks some very useful features some other models have

Boosted Decision Tree (BDT)

- **Decision Tree:** Make decision by asking a series of binary questions
- **Boosting Algorithms:** iteratively grow many weak classifier and aggregate them to create a strong classifier





Advantages of BDT

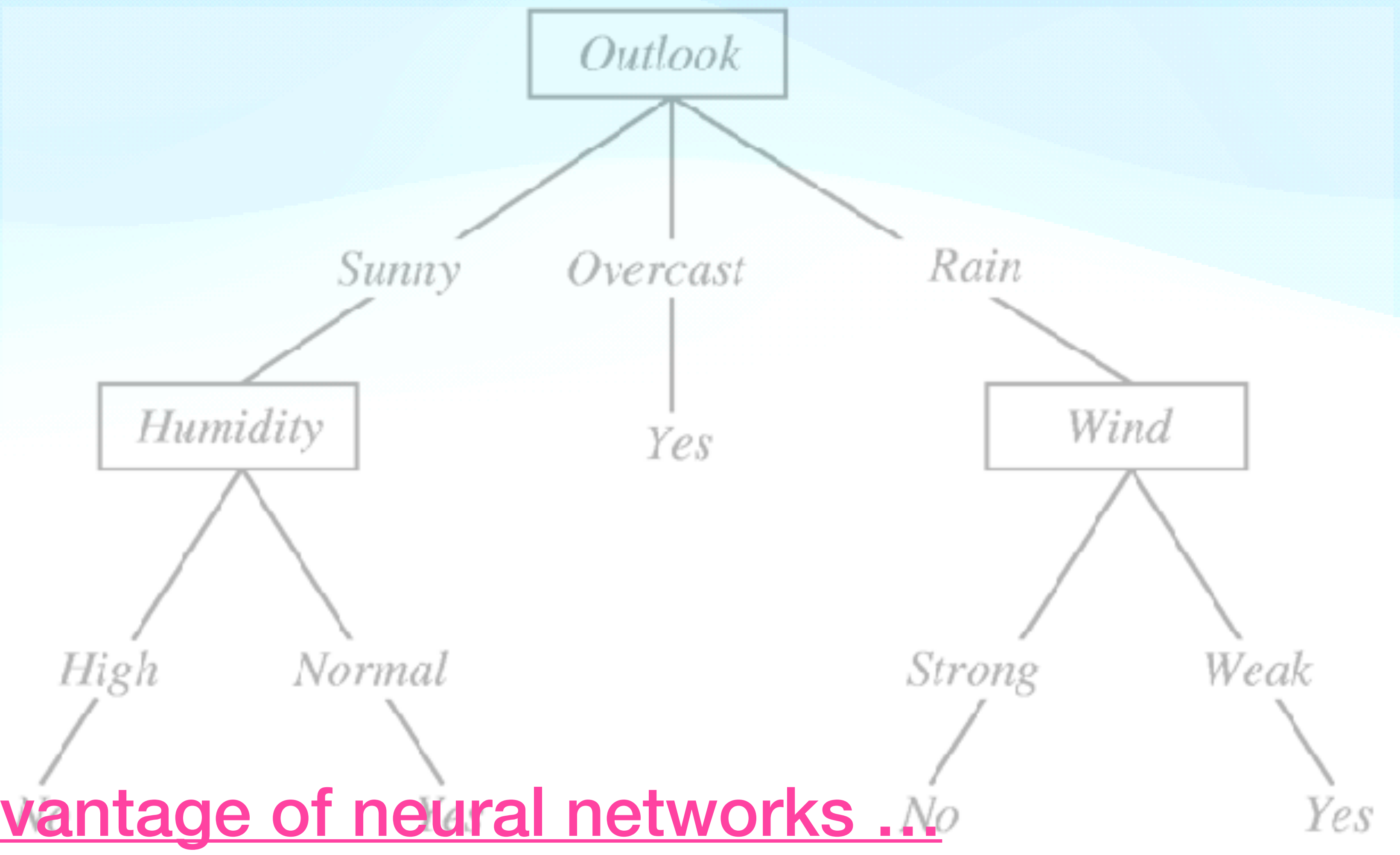
- **Ensemble Learning:** BDT and other ensemble learning methods usually achieves the best performance in Kaggle challenges
- Naturally come with some level of interpretability

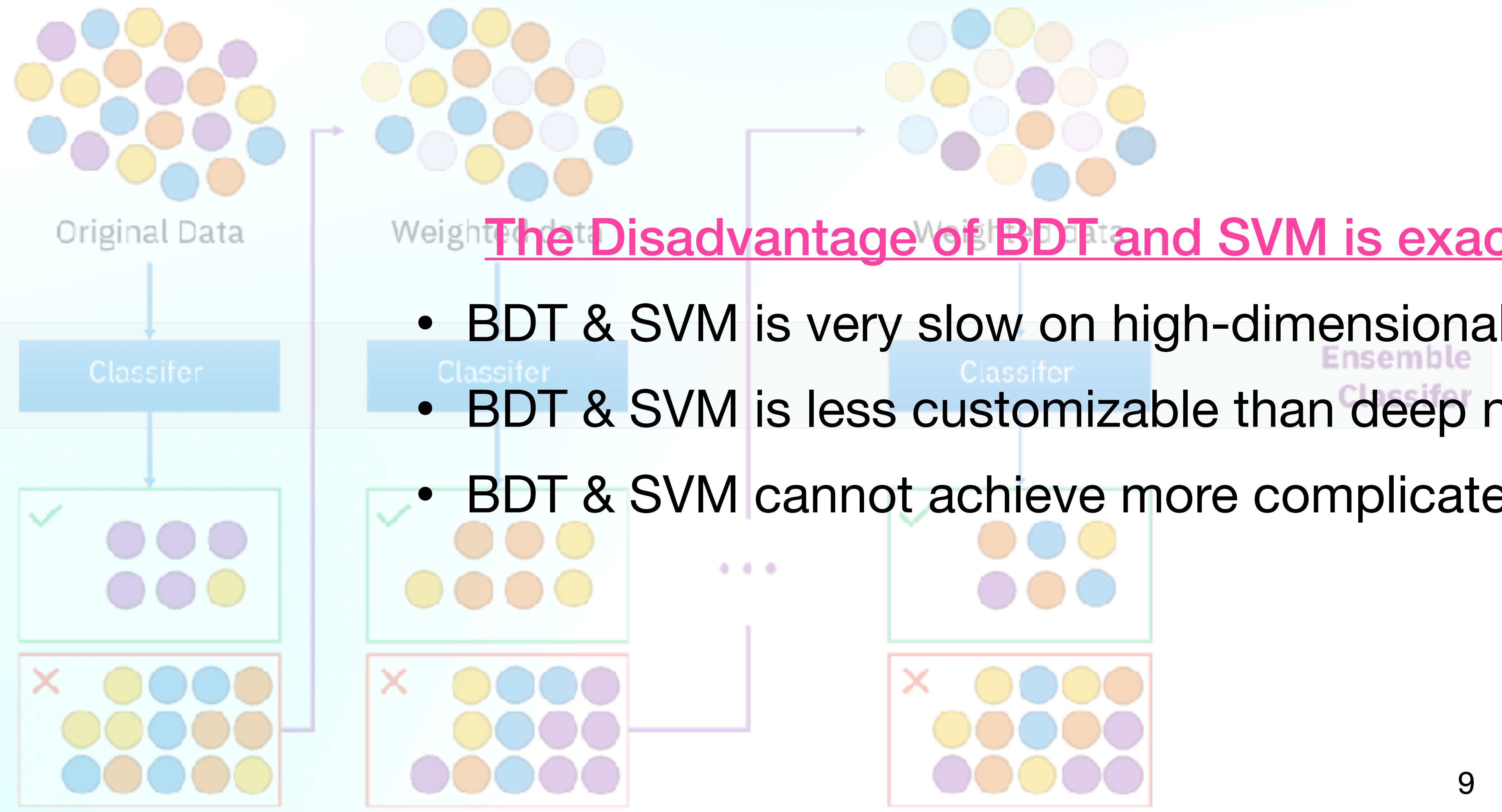
Deep Neural Networks can be used to analyze Tabular Data, but it's usually not the best model…

- DNN is particularly powerful for high dimensional data, but Tabular data is usually low dimensional
- DNN lacks some very useful features some other models have

Boosted Decision Tree (BDT)

- **Decision Tree:** Make decision by asking a series of binary questions
- **Boosting Algorithms:** iteratively grow many weak classifier and aggregate them to create a strong classifier





The Disadvantage of BDT and SVM is exactly the advantage of neural networks …

- BDT & SVM is very slow on high-dimensional data (i.e. raw detector output)
- BDT & SVM is less customizable than deep neural networks
- BDT & SVM cannot achieve more complicated tasks like data generation

Advantages of BDT

- **Ensemble Learning:** BDT and other ensemble learning methods usually achieves the best performance in Kaggle challenges
- Naturally come with some level of interpretability

Nuclear Physics

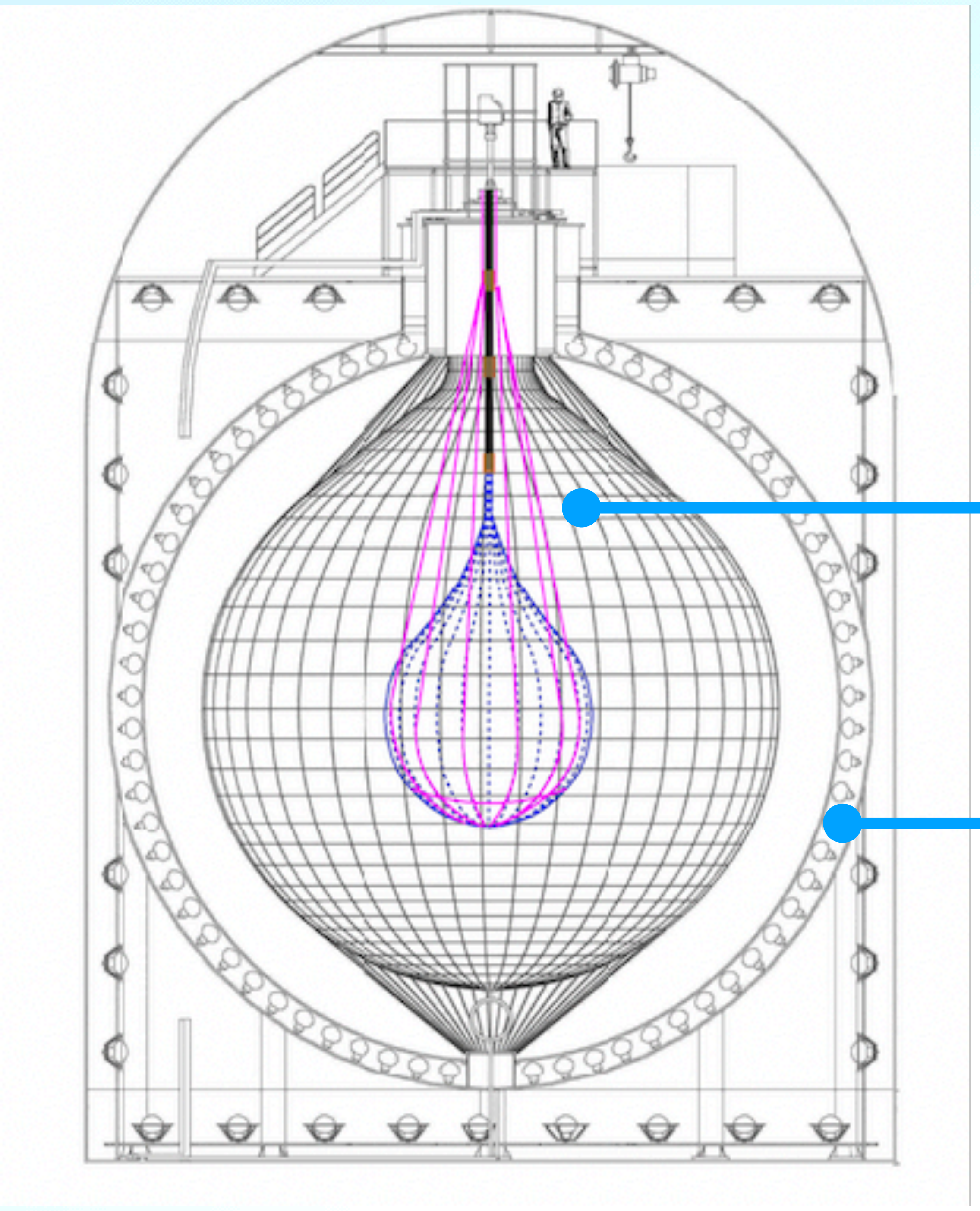
Q2: My experiment does not produce short waveforms/time series data like MAJORANA DEMONSTRATOR does, it produces more complicated **high-dimensional data**. what should I use as my feature extraction network?

AI/ML

The exact model to use depends on how you pre-process your data into the input format

Raw Data → Data Preprocessing → Input Data → Feature Extraction → Representation → Task Module → Output

Raw Data → Feature Engineering → Tabular Data → Classical Machine Learning Model → Output



Liquid Scintillator

Inner Detector PMTs

1325 17inch + 554 20inch

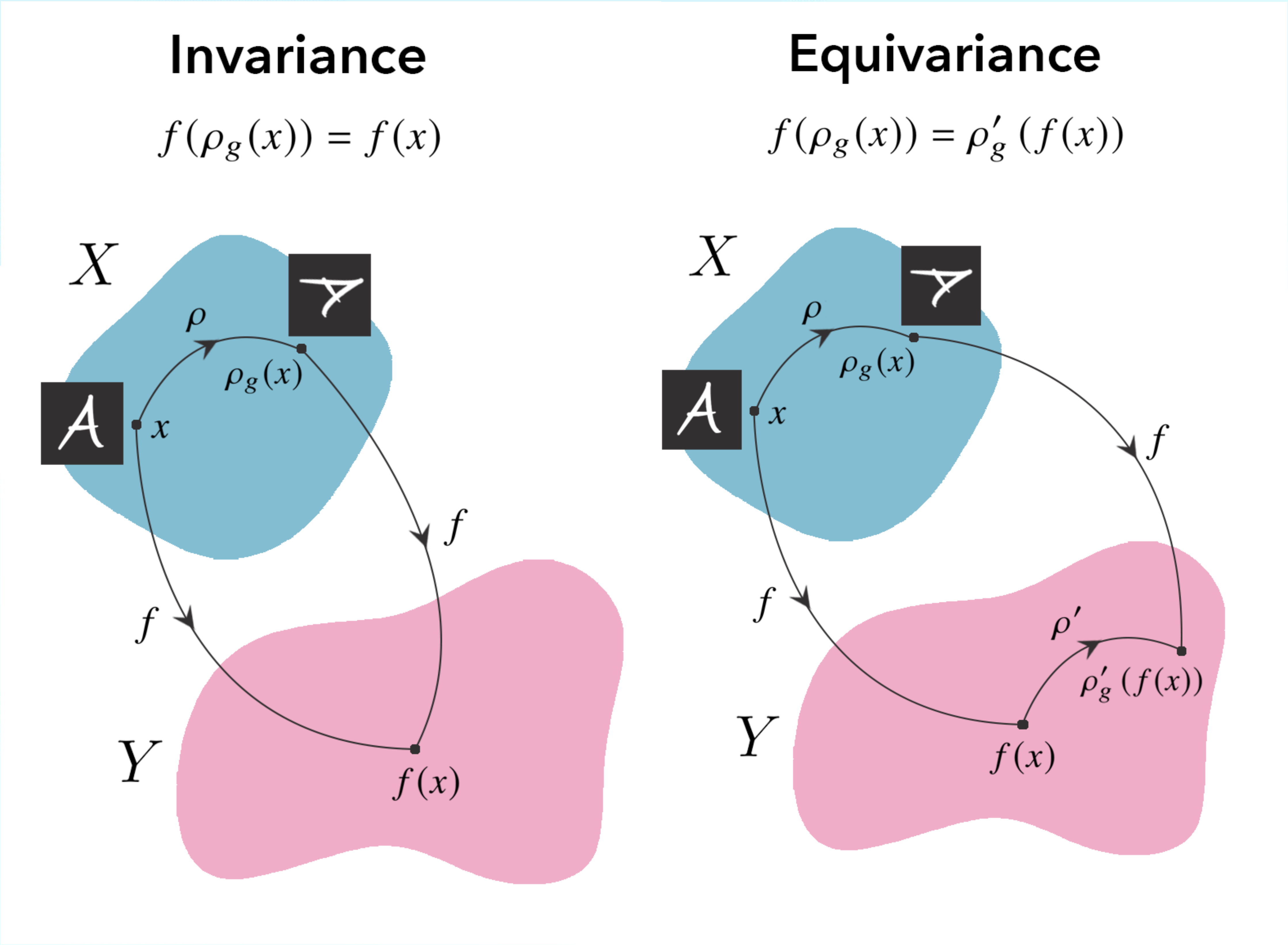

Raw Data
Data Preprocessing
Input Data
Feature Extraction
Representation
Task Module
Output
Feature Engineering
Tabular Data
Classical Machine Learning Model
Triggered PMT
(R, θ, ϕ, t, q)
2. 2D Image
3. Spatiotemporal Data (2D Movie)
t
1. Time Series
KamNet Event Window
Normalized Amplitude
Proper Hit Time [ns]
4. Point Cloud Data

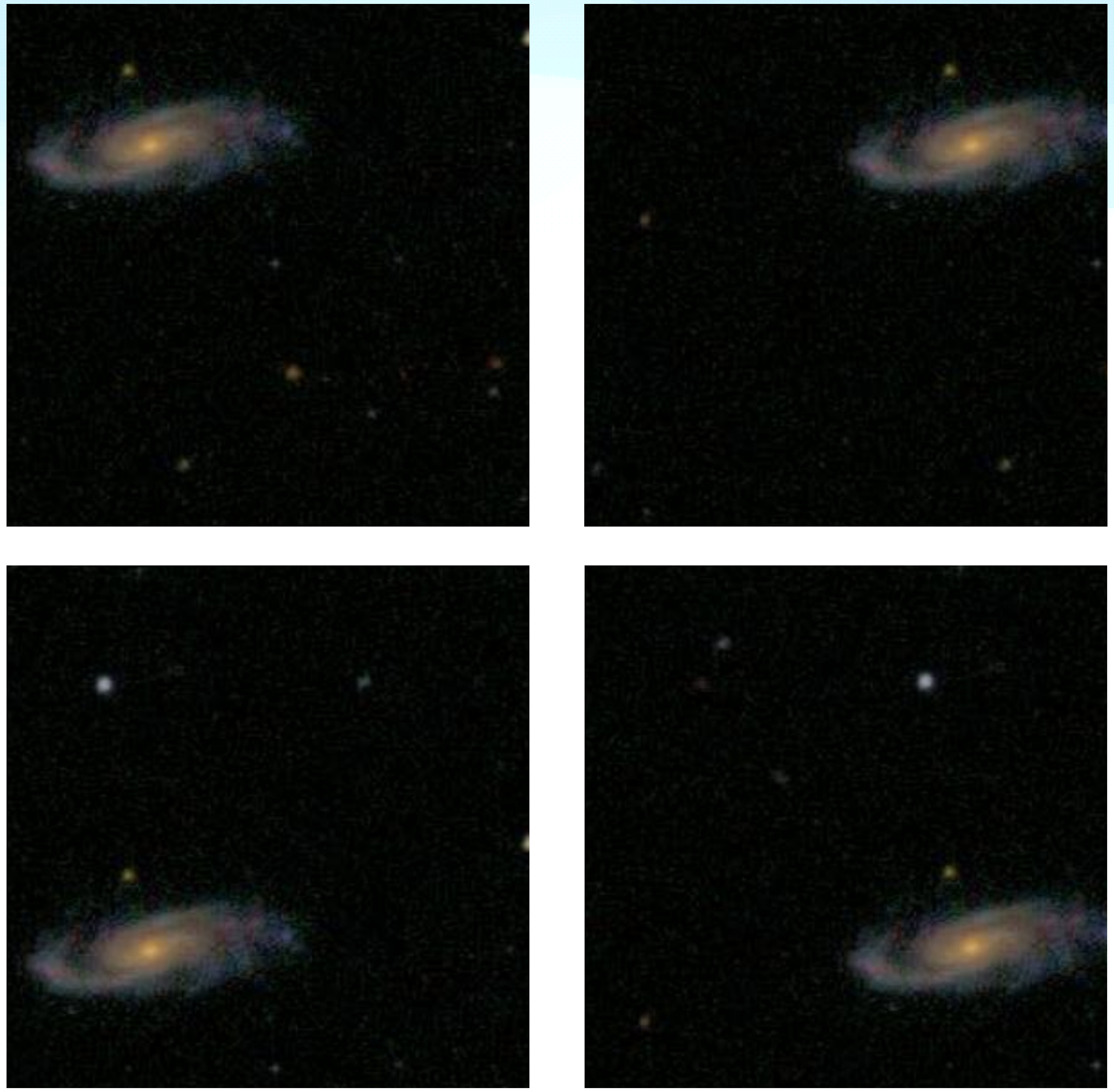
Raw Data → Data Preprocessing → Input Data → Feature Extraction → Representation → Task Module → Output

Raw Data → Feature Engineering → Tabular Data → Classical Machine Learning Model → Output

AI/ML

- The exact model to use depends on how you pre-process your data into the input format.
- **Convolutional Neural Network (CNN)** is a good model for multiple data types in general.





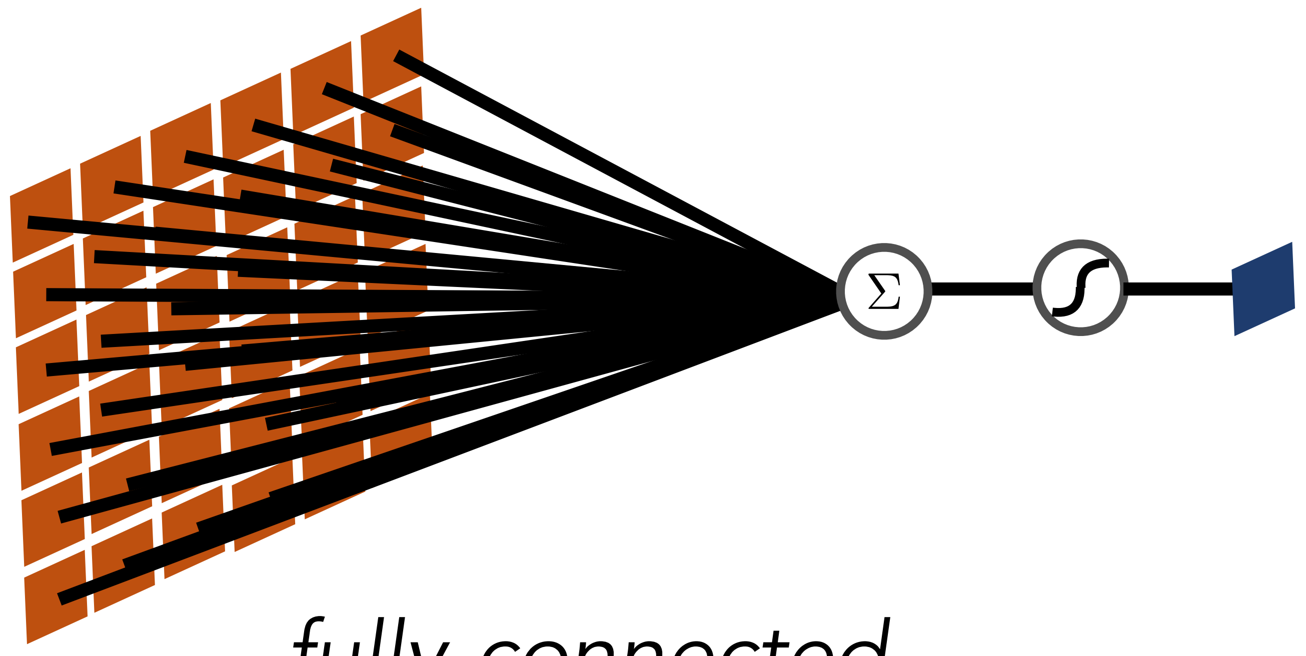
12

AI/ML

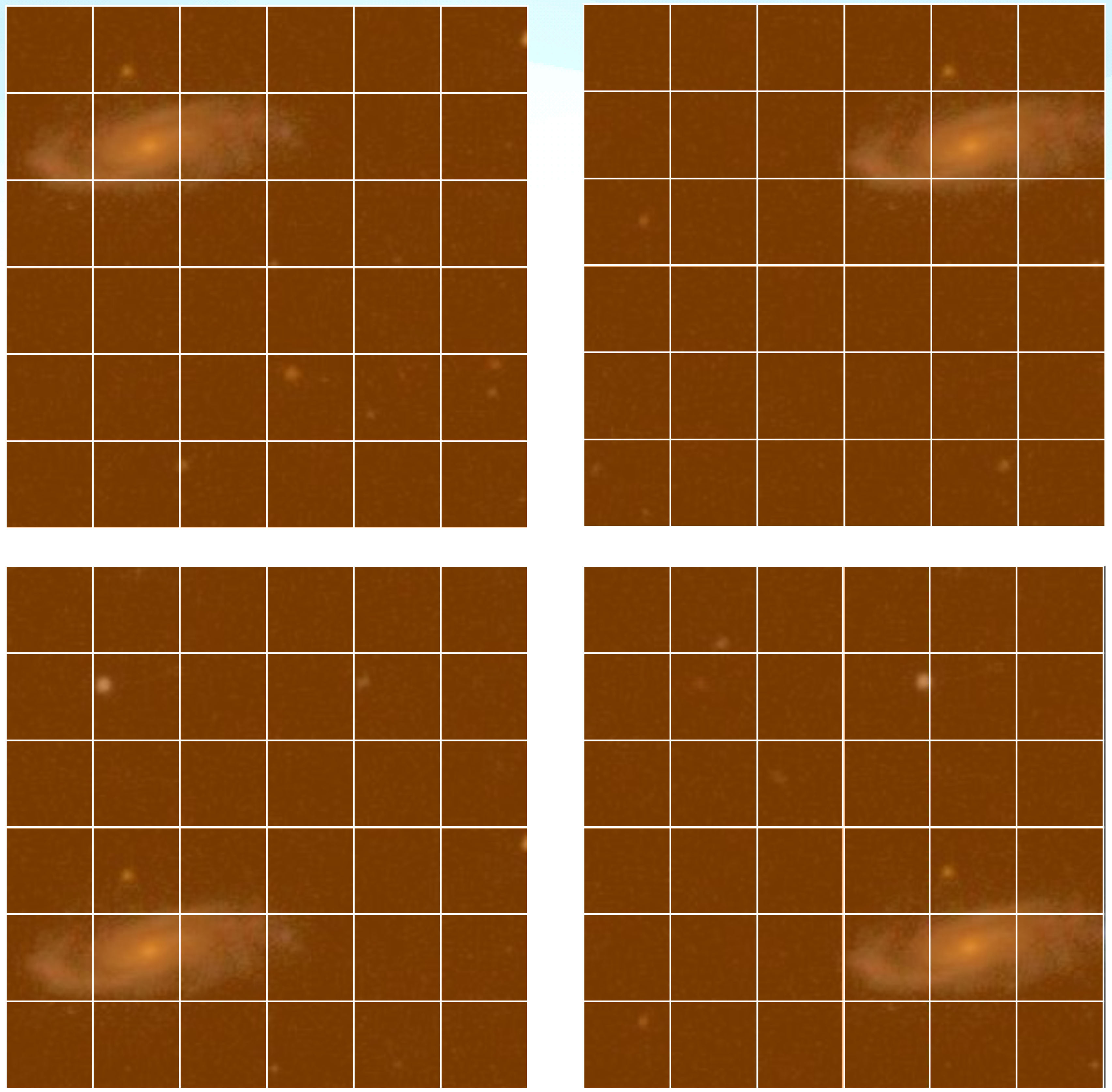
- The exact model to use depends on how you pre-process your data into the input format.
- **Convolutional Neural Network (CNN)** is a good model for multiple data types in general.

Fully Connected Neural Network (last lecture)

- Fully-connected neural networks are not **translation invariant**
- Also has a huge parameter space:
  - 2 million parameters for 1920×1080 image



fully-connected



13

AI/ML

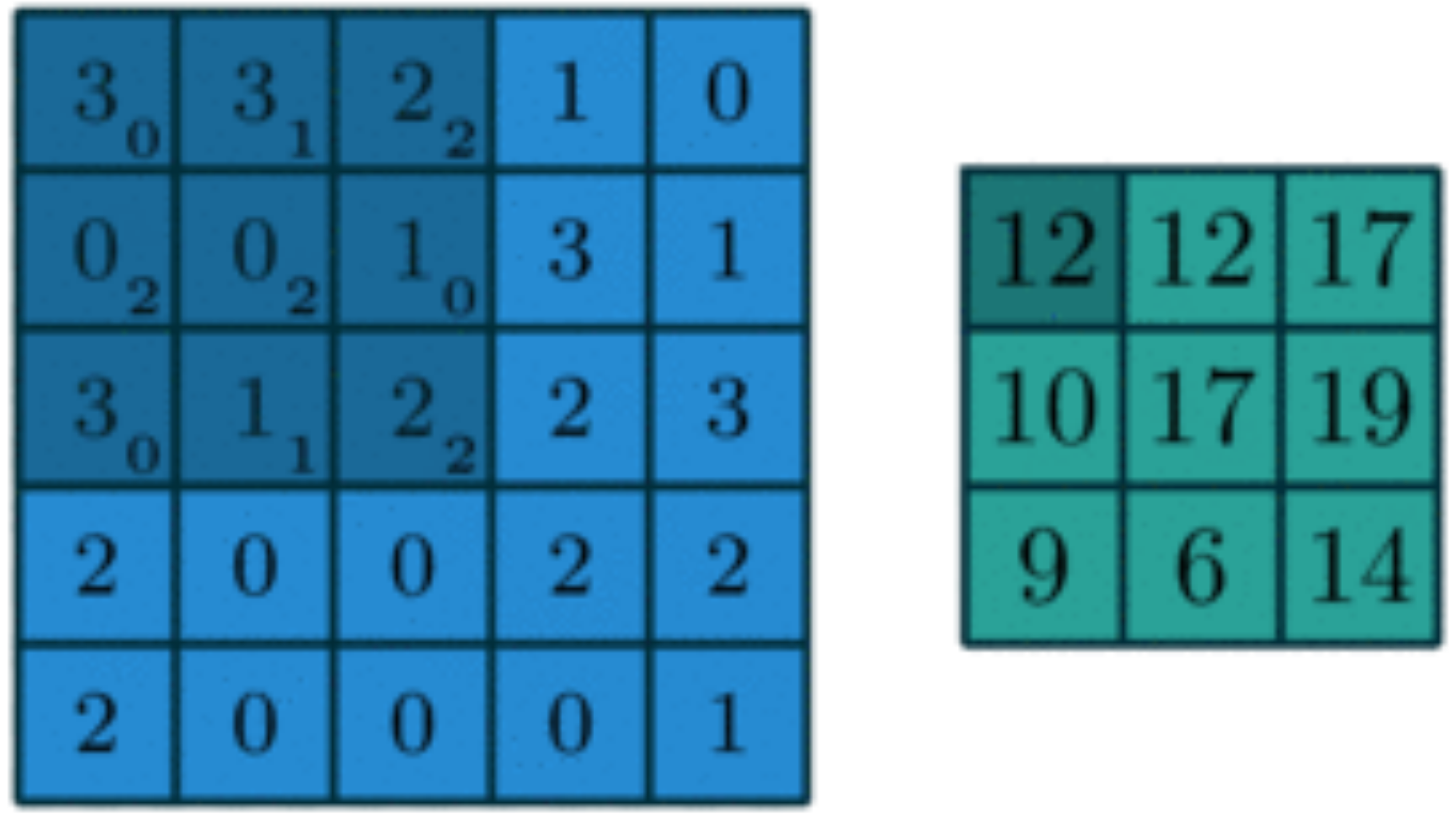
- The exact model to use depends on how you pre-process your data into the input format.
- **Convolutional Neural Network (CNN)** is a good model for multiple data types in general.

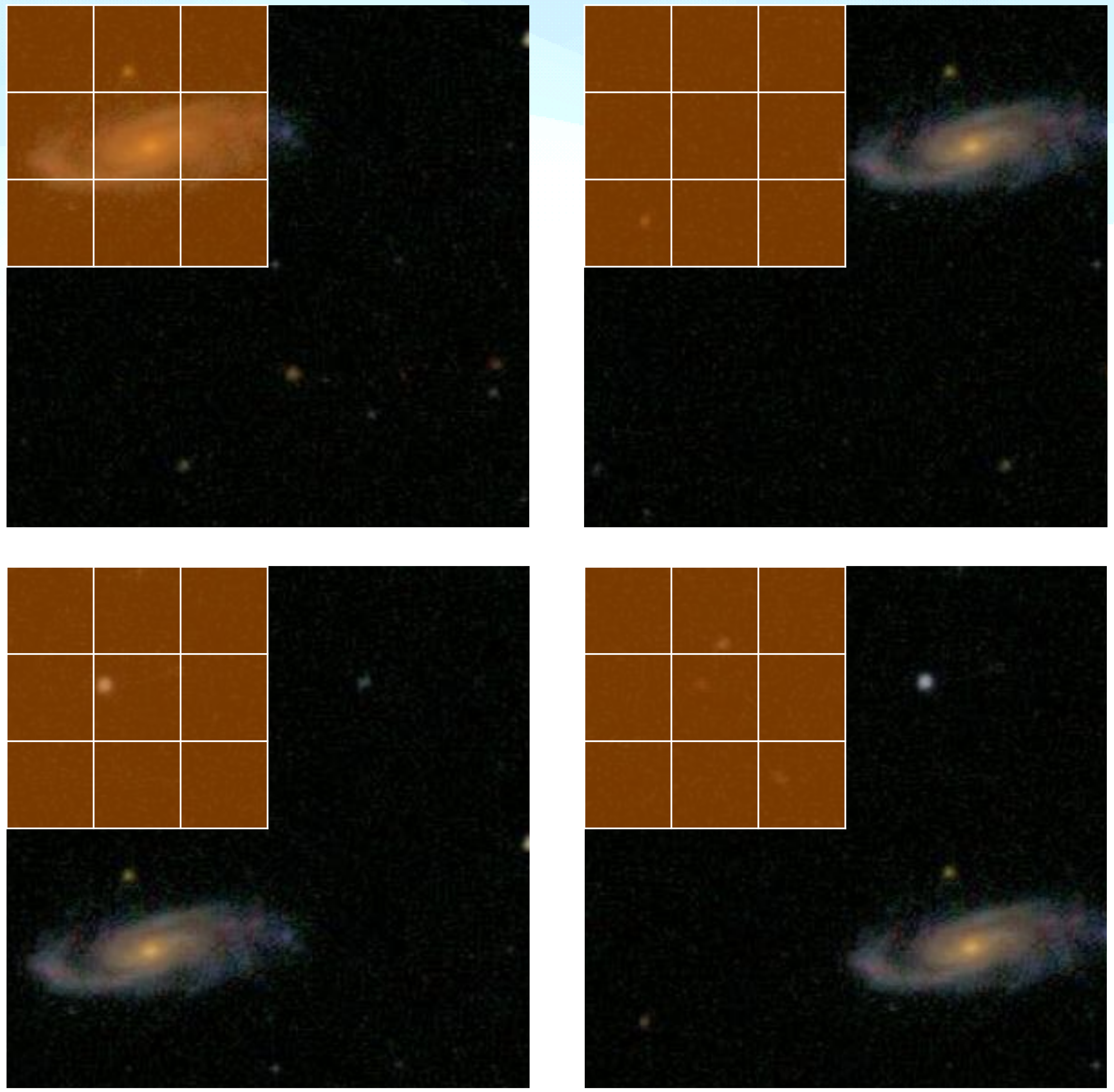
Convolution

- The only linear and translation-equivariant operations
- Scan n **filters** throughout the 2D images
  - n is the **channel** of CNN

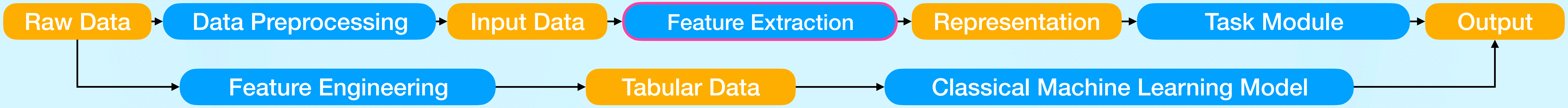
Filter weights:

$$\begin{pmatrix} 0 & 1 & 2 \\ 2 & 2 & 0 \\ 0 & 1 & 2 \end{pmatrix}$$



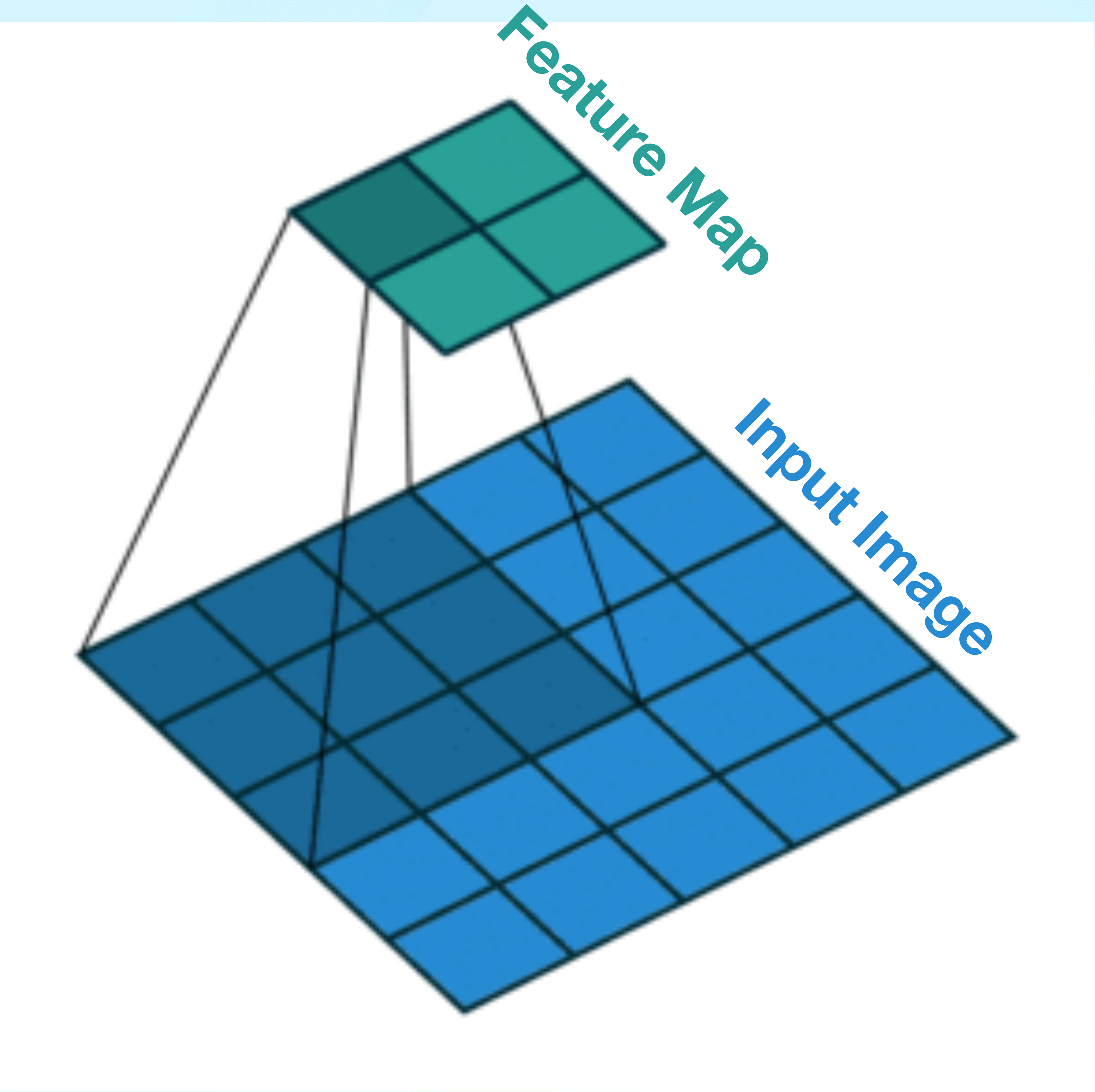


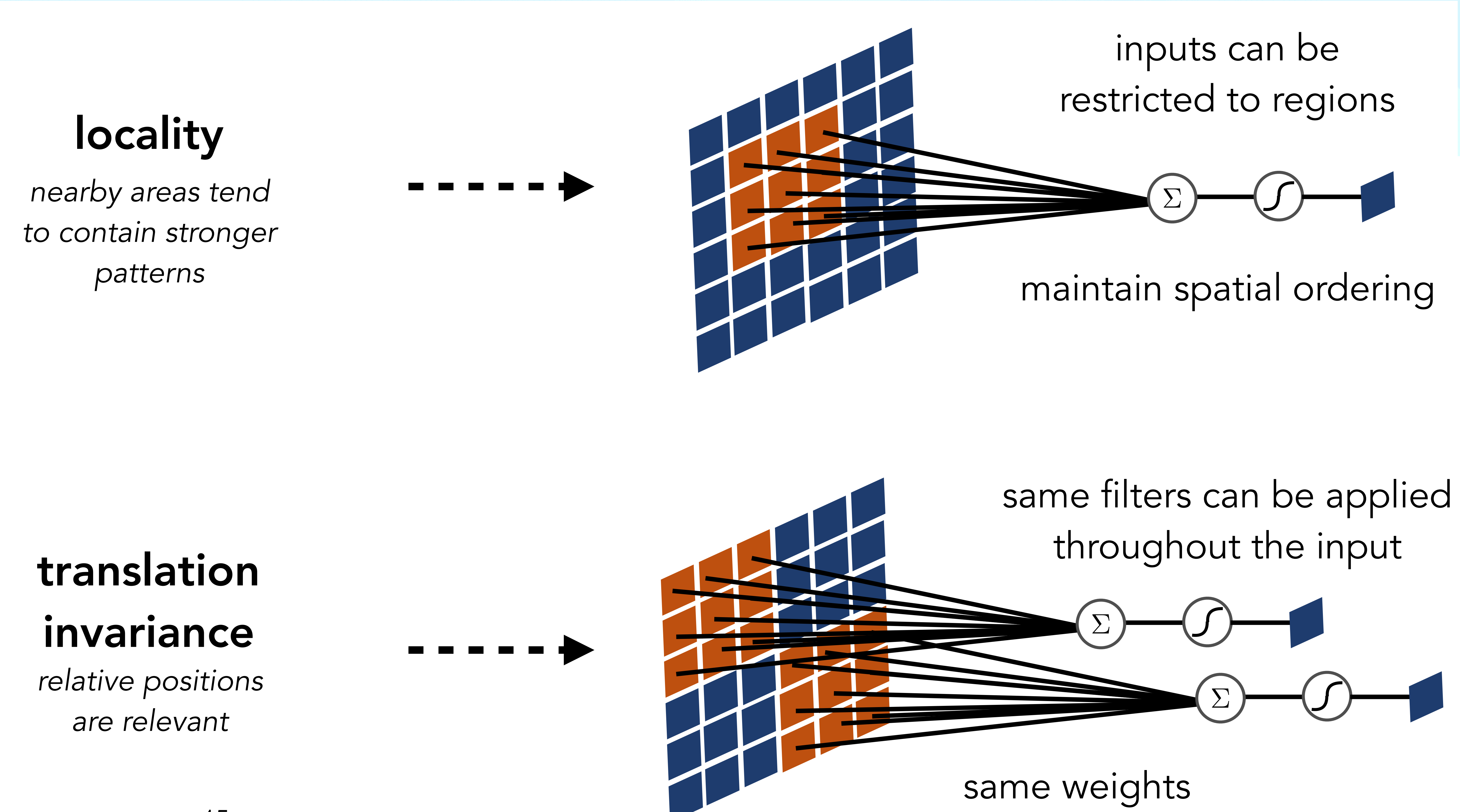
14



AI/ML

- The exact model to use depends on how you pre-process your data into the input format.
- **Convolutional Neural Network (CNN)** is a good model for multiple data types in general.





15

AI/ML

- The exact model to use depends on how you pre-process your data into the input format.
- **Convolutional Neural Network (CNN)** is a good model for multiple data types in general.

Convolution

Feature map smaller than Input Image

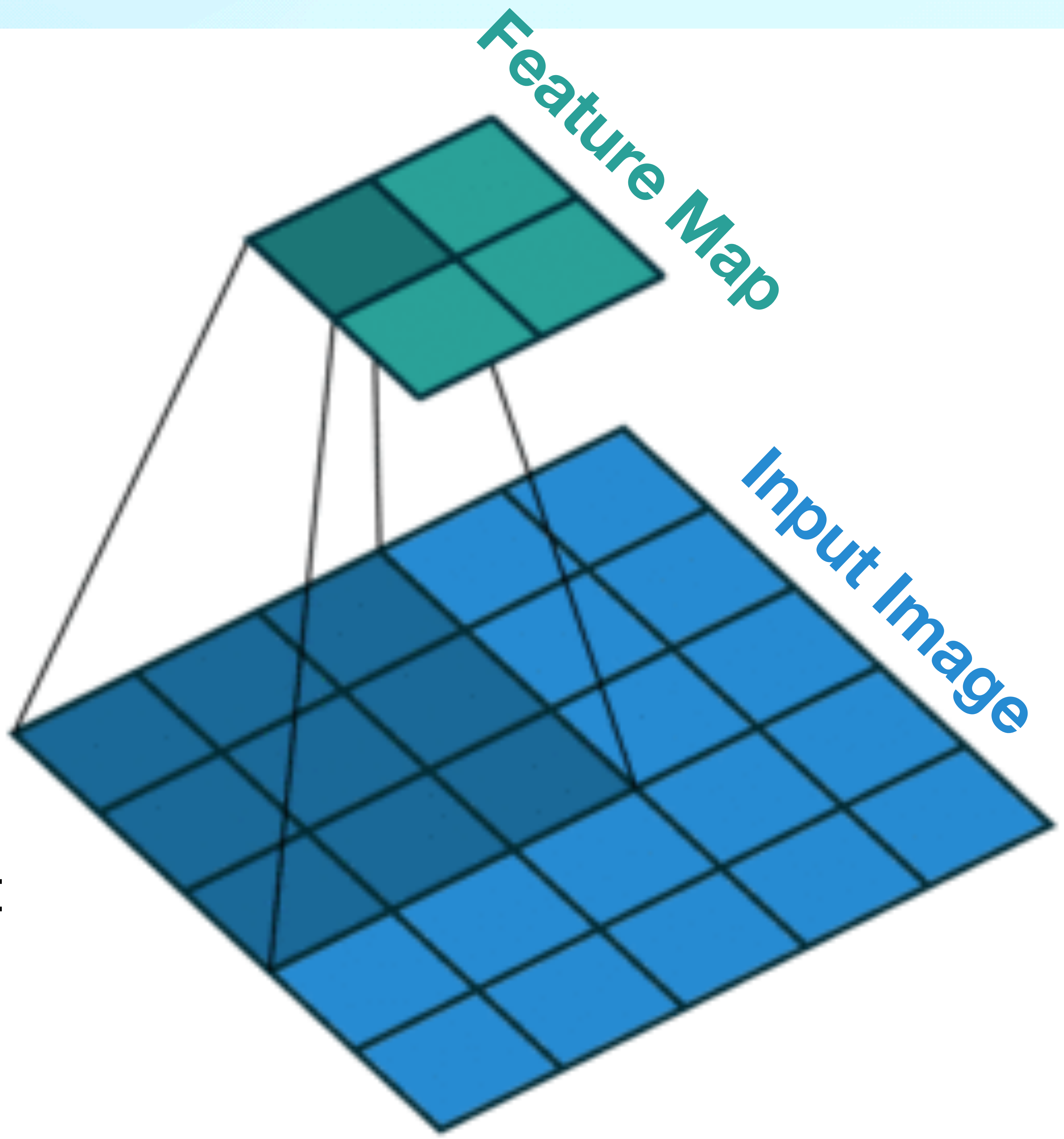
5×5 input

3×3 filter

2×2 stride

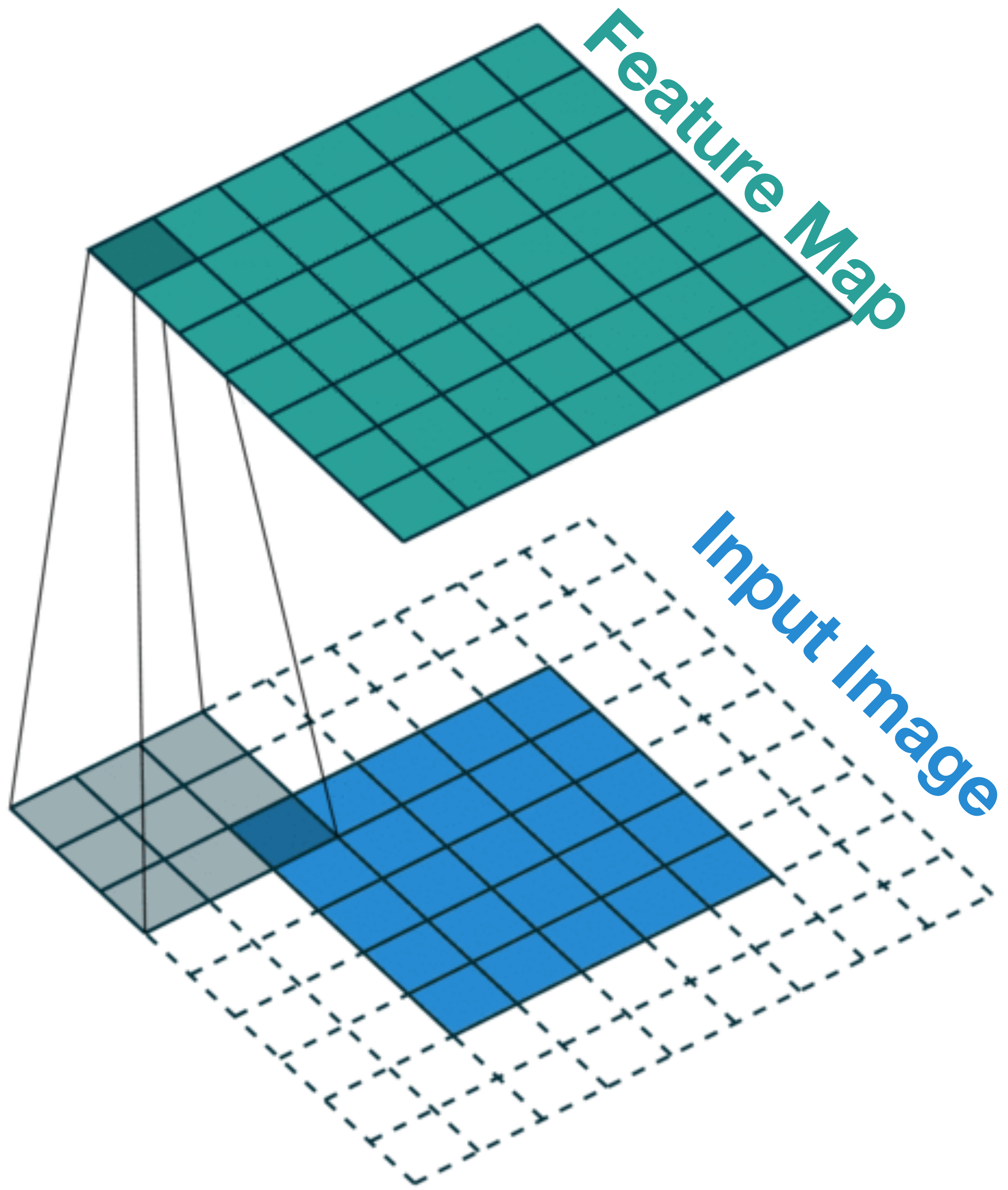
No zero padding

➡ 2×2 output



Deconvolution

Feature map larger than input image



5×5 input

3×3 filter

1×1 stride

- “Full” zero padding

➡ 7×7 output

16

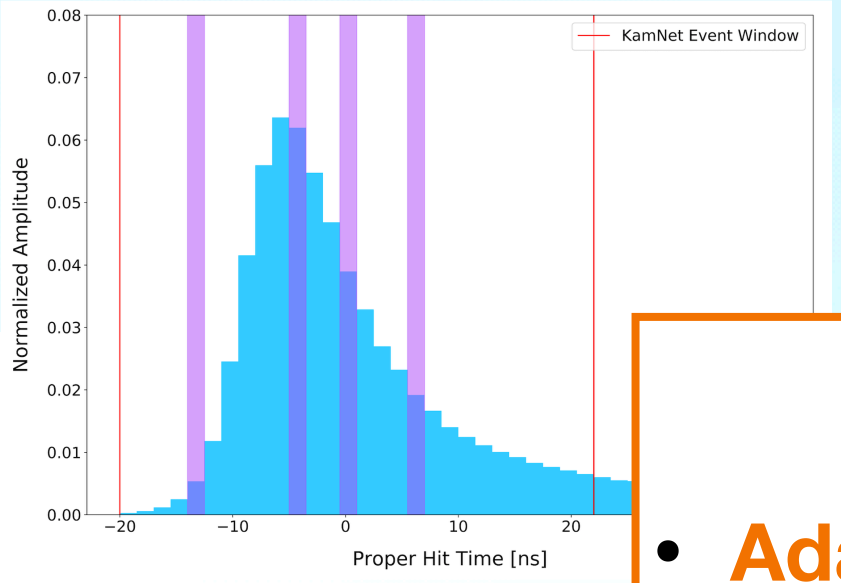
Raw Data → Data Preprocessing → Input Data → Feature Extraction → Representation → Task Module → Output

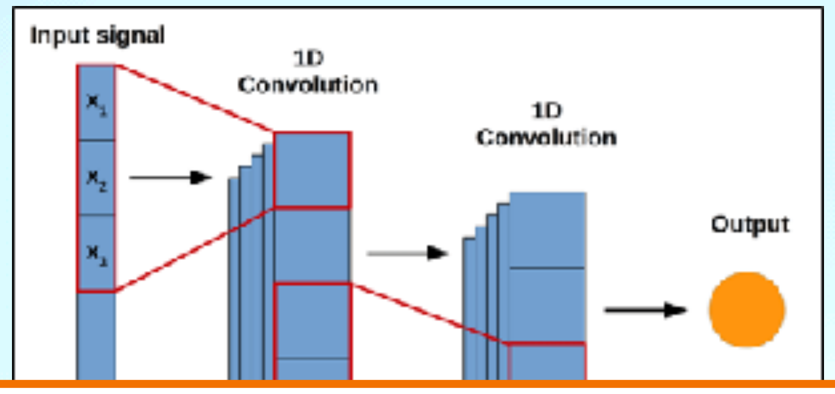
Raw Data → Feature Engineering → Tabular Data → Classical Machine Learning Model → Output

AI/ML

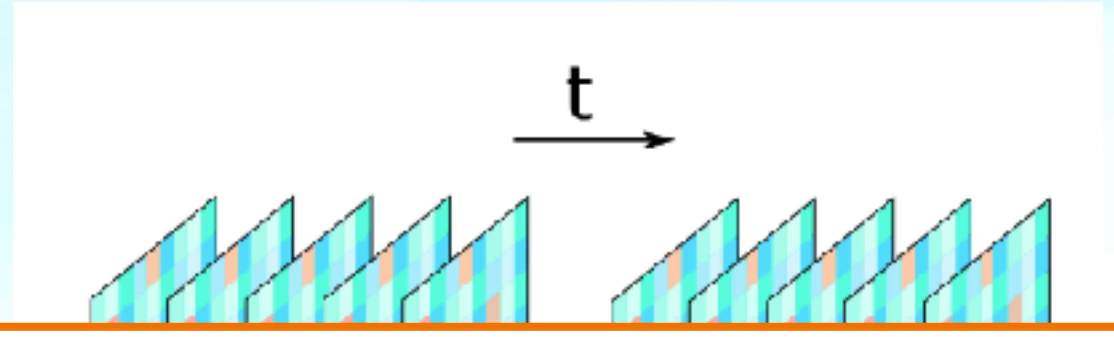
- The exact model to use depends on how you pre-process your data into the input format
- **Convolutional Neural Network (CNN)** is a good model for multiple data types in general

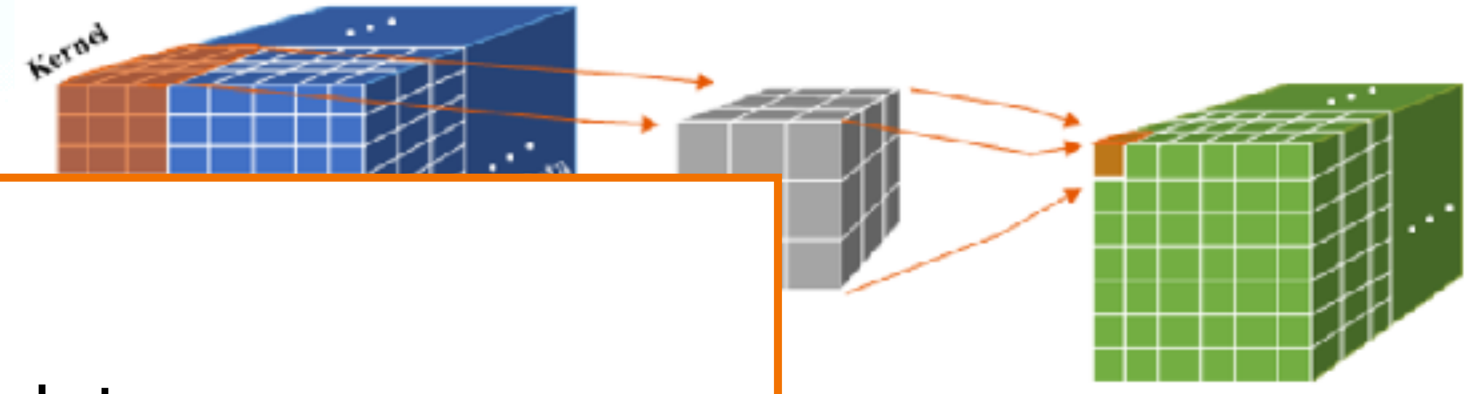
1. Time Series → 1D CNN





3. Spatiotemporal Data → 3D CNN

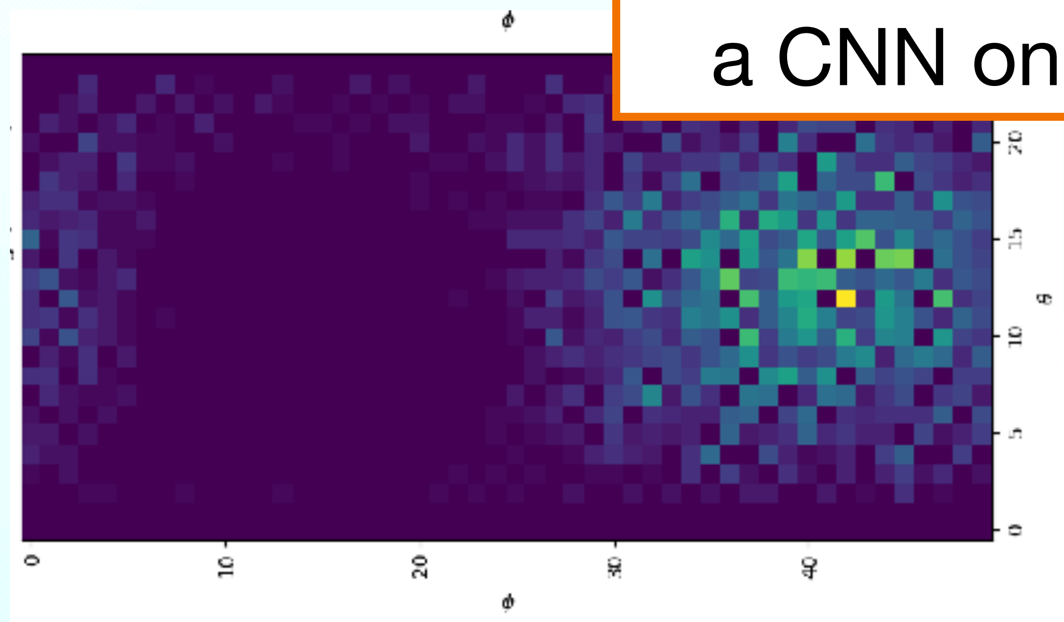


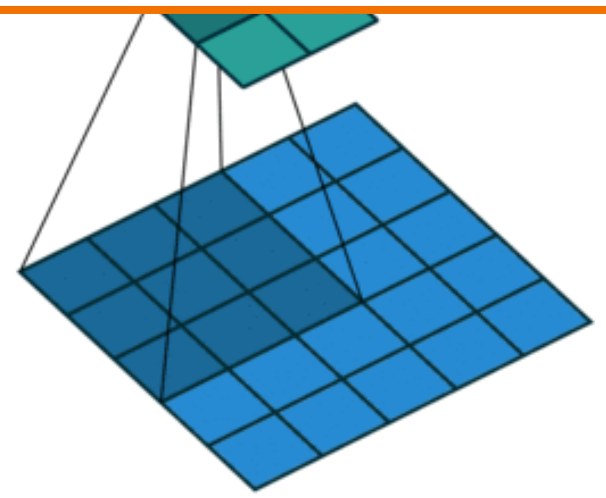


CNN is a general-purpose model

- **Adaptability:** can serve as the feature extractor for 1D, 2D, 3D, or even 4D data
  - For 4D, using 3D CNN + channel as additional dimension
- **Baseline Model:** when starting your own machine learning project, it's always good to first build a CNN on your data to produce a baseline classification/regression result

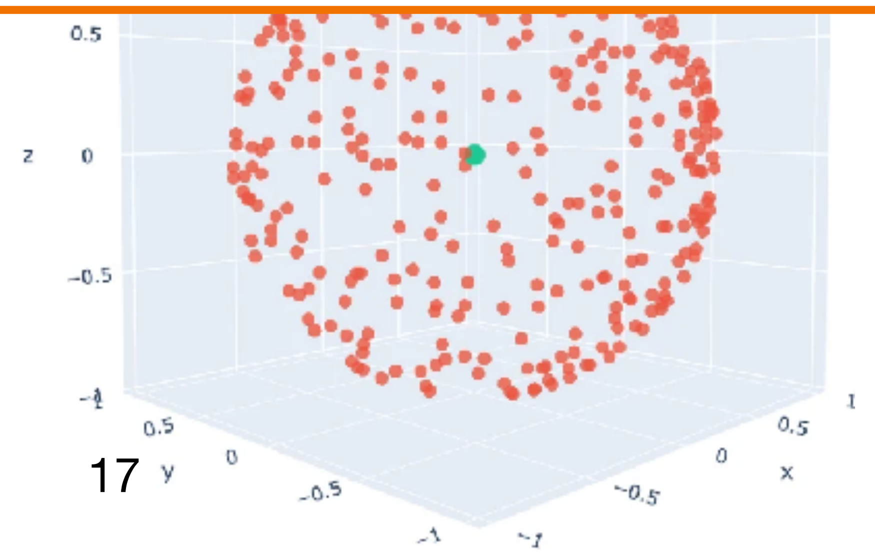
2. 2D Image

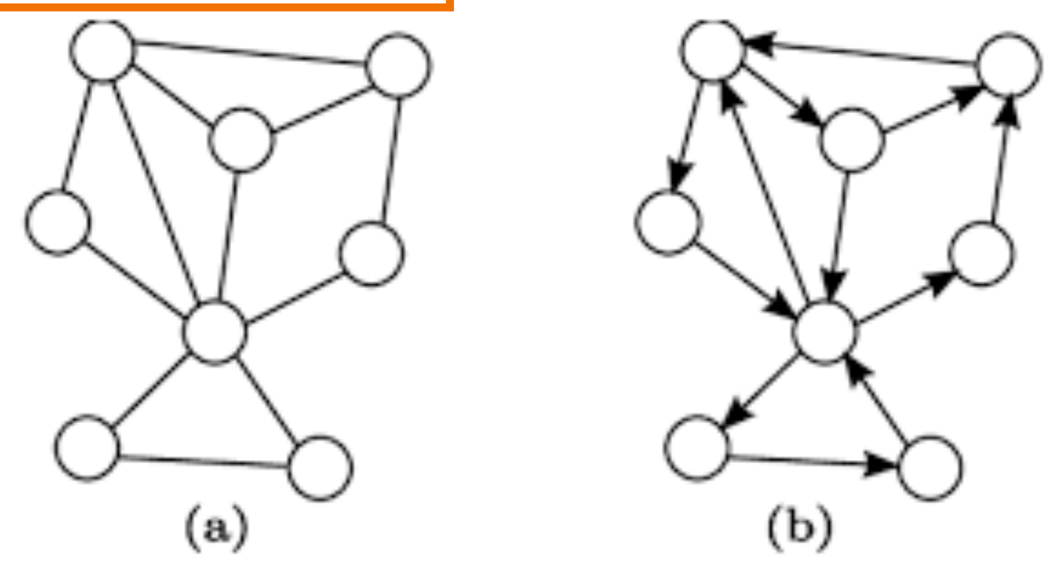




se CNN

etwork are better options





(a) An example of undirected graph and (b) an example of directed graph

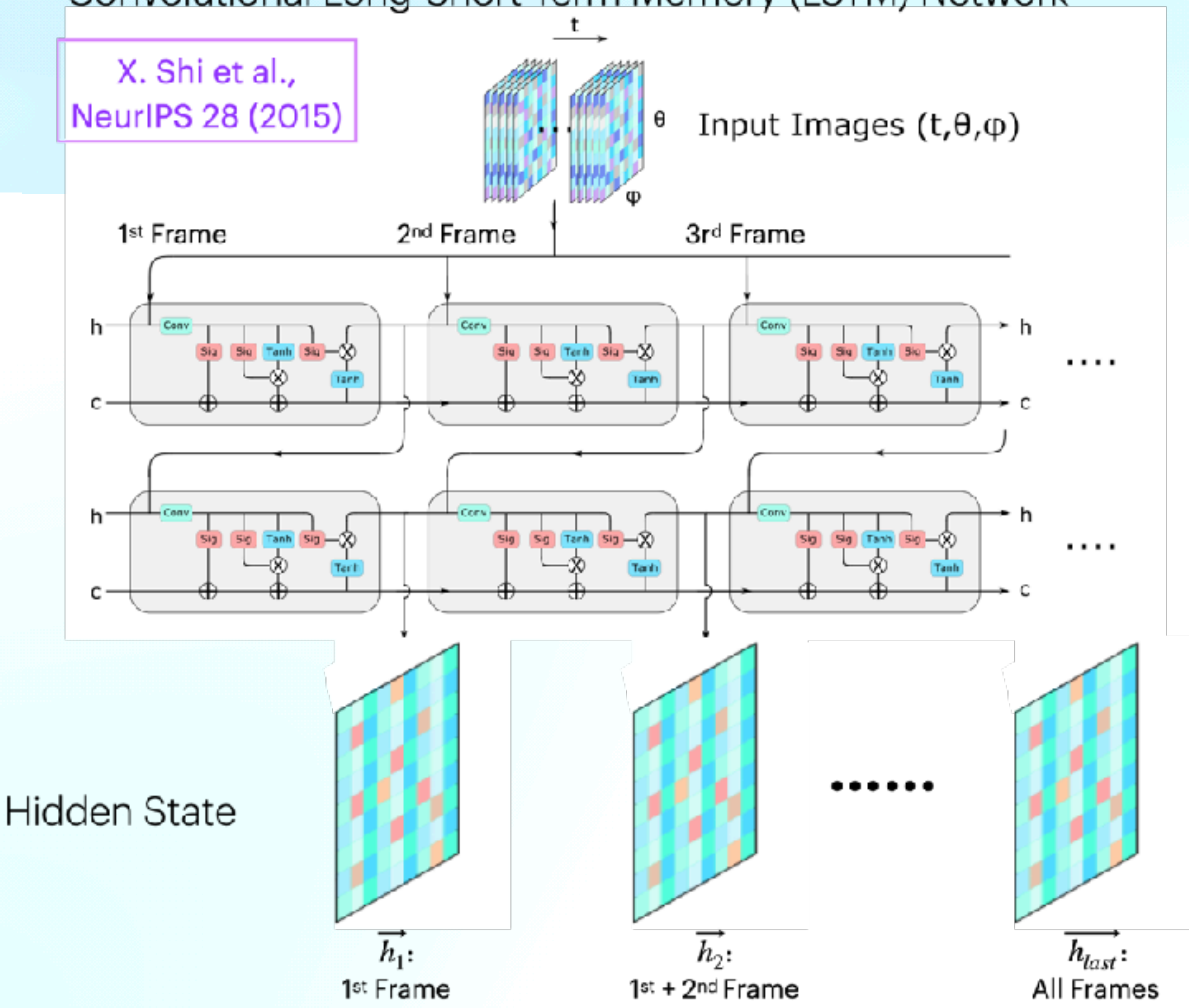
17

AI/ML

- The exact model to use depends on how you pre-process your data into the input format
- **Convolutional Neural Network (CNN)** is a good model for multiple data types in general
- Enhance neural network's performance by encoding symmetries with **Geometric Deep Learning**

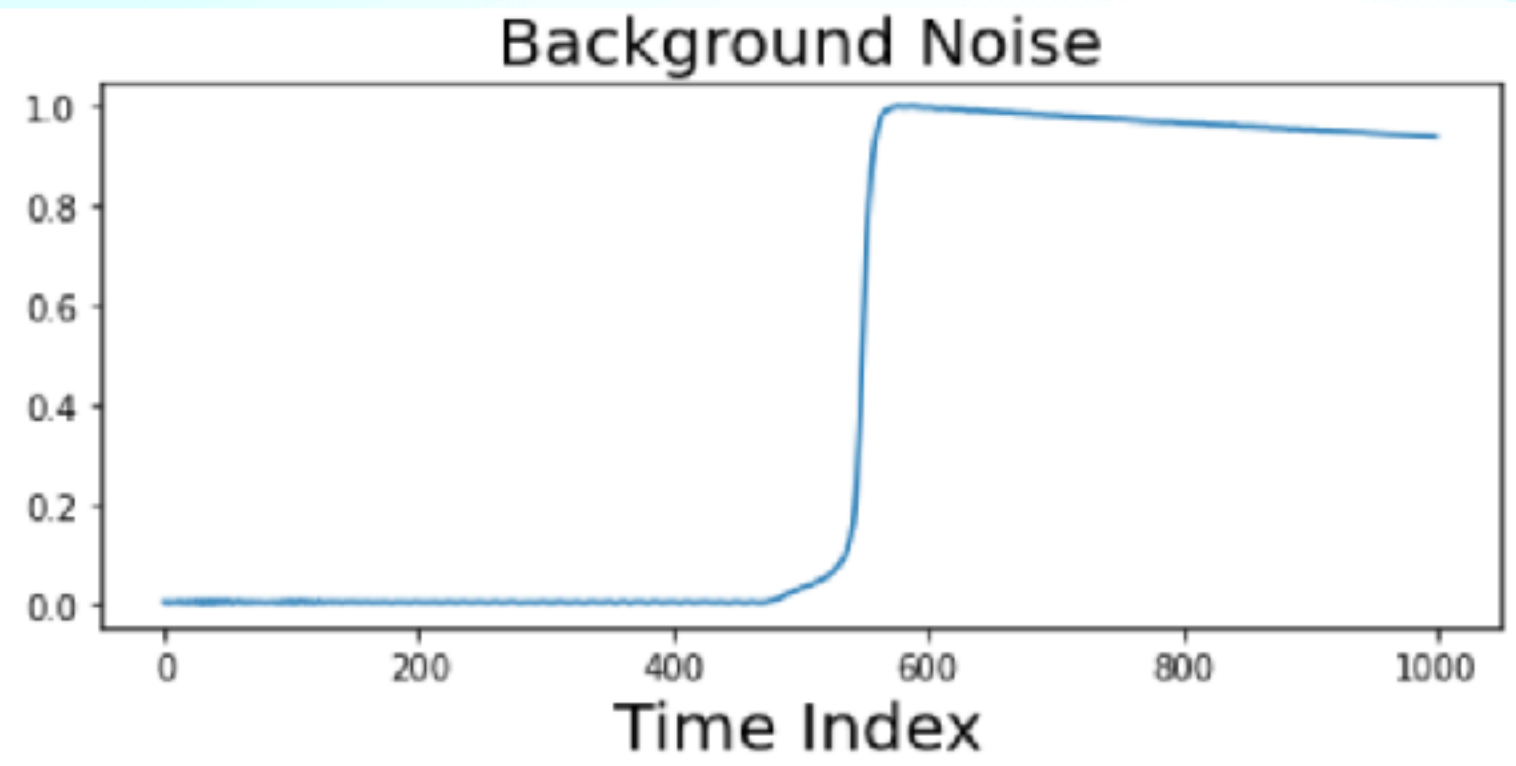
ConvLSTM

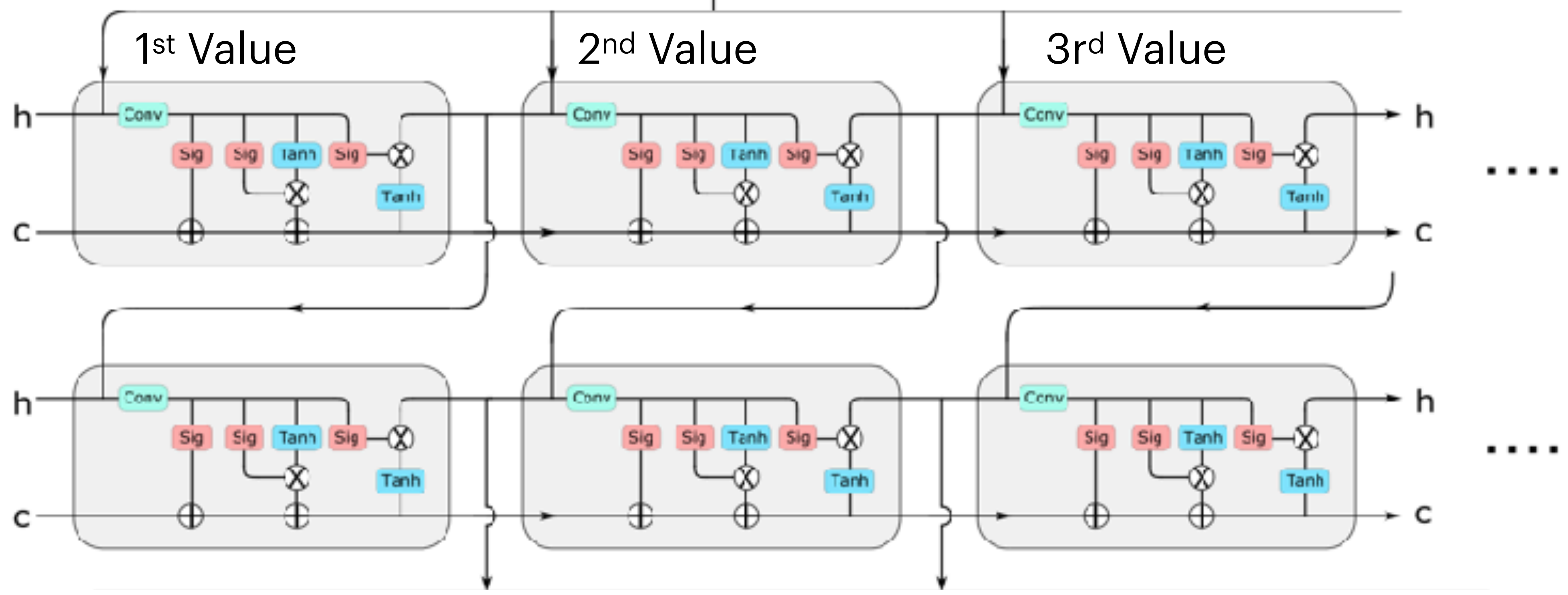
Convolutional Long-Short Term Memory (LSTM) Network



LSTM

LSTM (without Convolution) is efficient against time series data for similar reasons





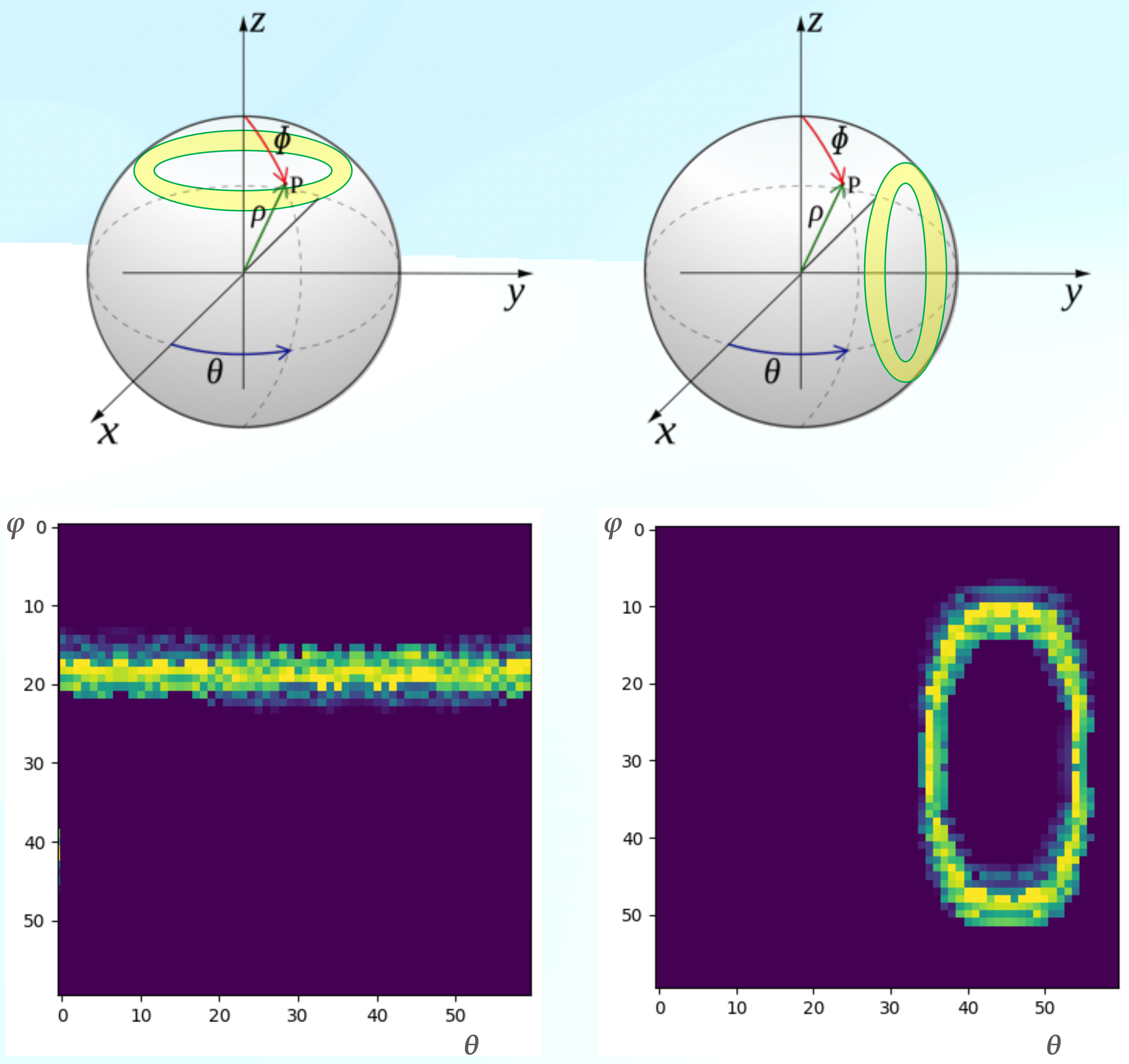
18

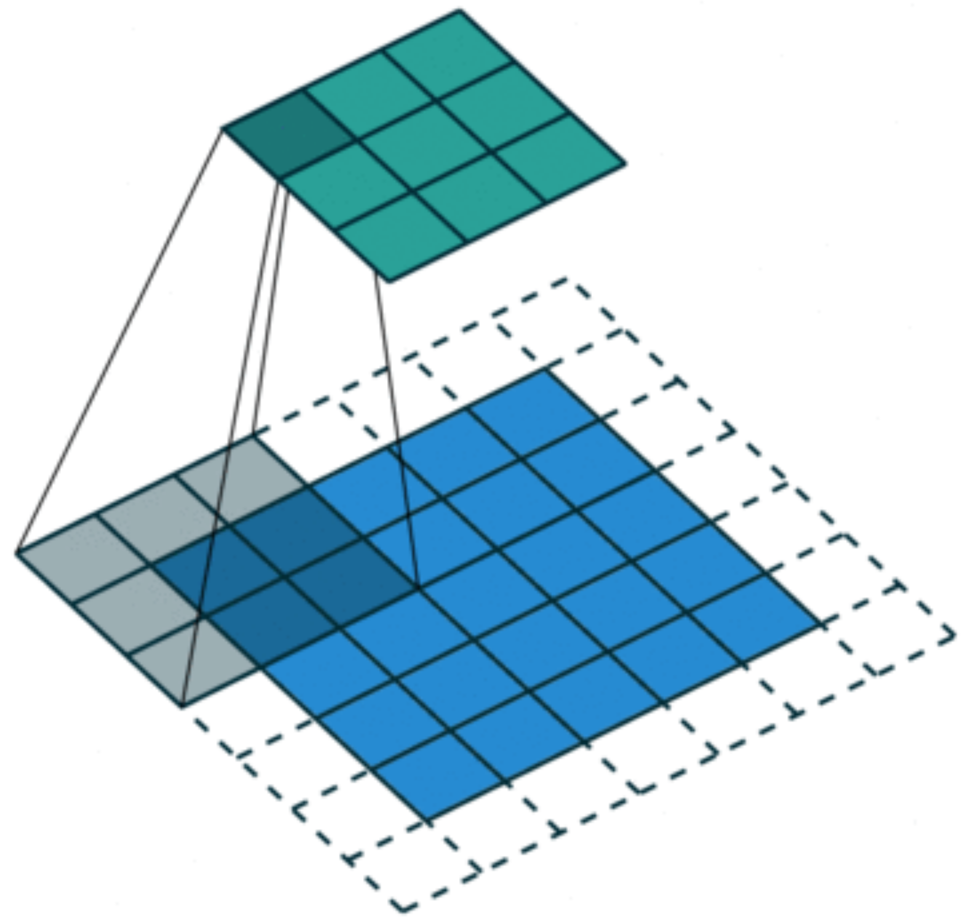
Raw Data → Data Preprocessing → Input Data → Feature Extraction → Representation → Task Module → Output

Raw Data → Feature Engineering → Tabular Data → Classical Machine Learning Model → Output

AI/ML

- The exact model to use depends on how you pre-process your data into the input format
- **Convolutional Neural Network (CNN)** is a good model for multiple data types in general
- Enhance neural network's performance by encoding symmetries with **Geometric Deep Learning**



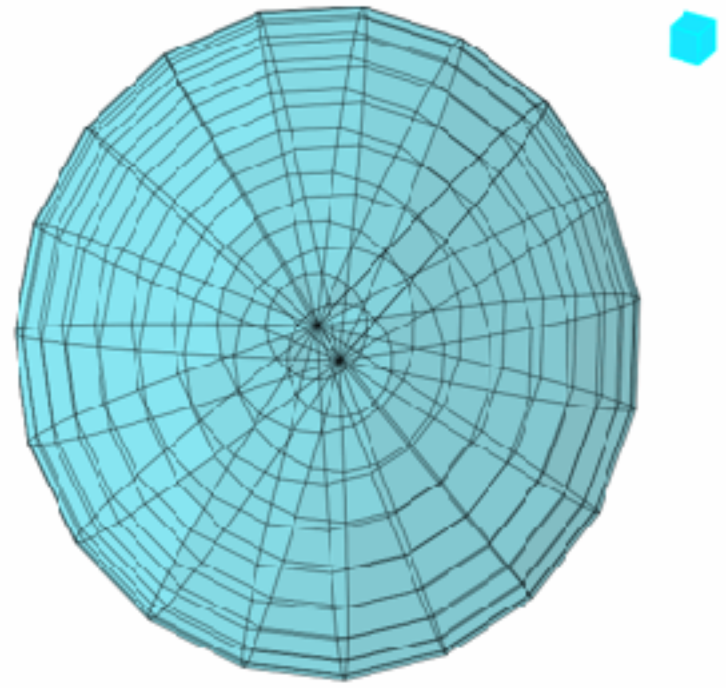


Spherical CNN

SO(3) symmetry & **rotational invariance**

Similar model exists for cylindrical detectors or other geometries

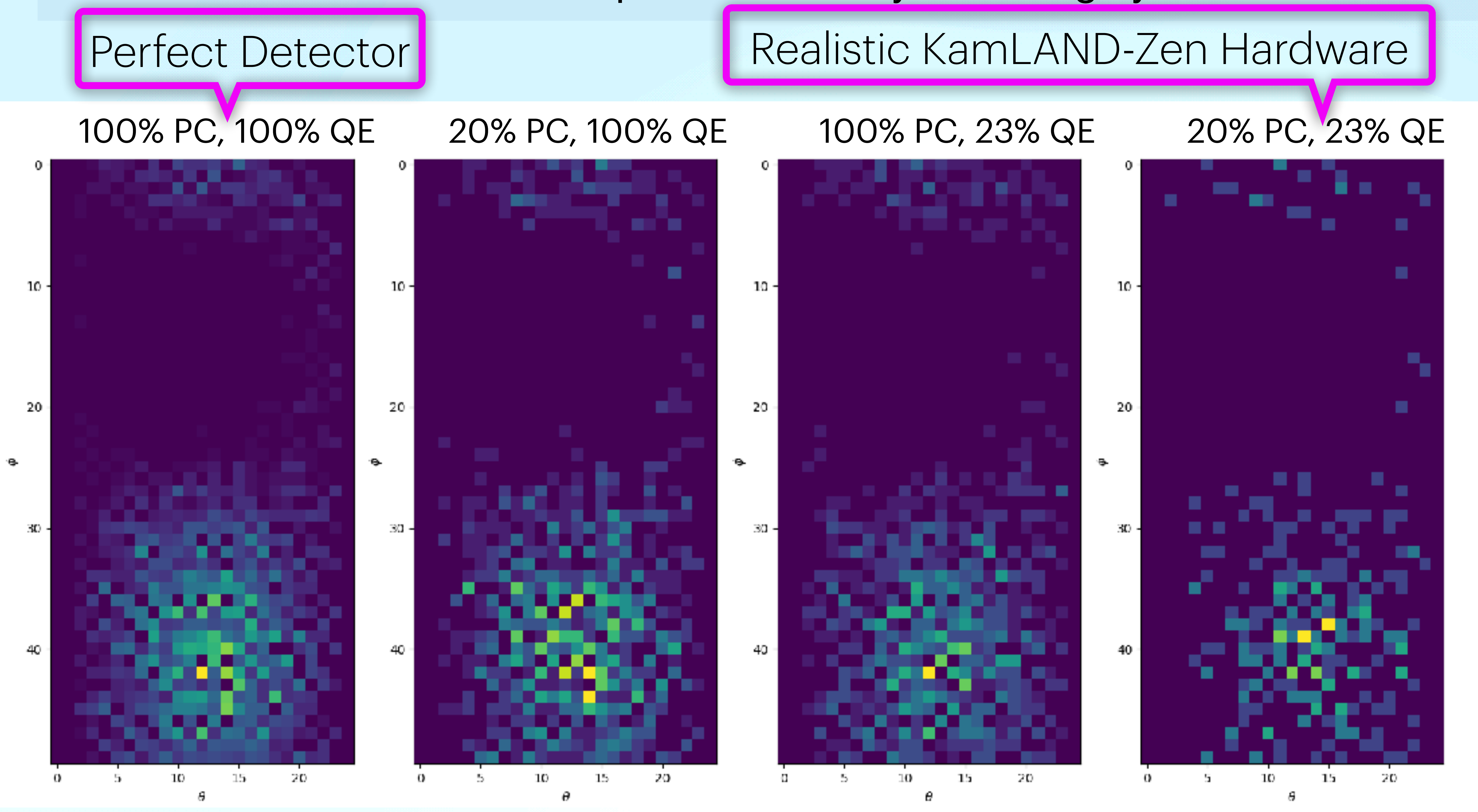
Cohen, Taco et al. "Spherical CNNs." ICLR 2018



19

AI/ML

- The exact model to use depends on how you pre-process your data into the input format
- **Convolutional Neural Network (CNN)** is a good model for multiple data types in general
- Enhance neural network's performance by encoding symmetries with **Geometric Deep Learning**



A. Li et al.,
DOI: 10.1016/j.nima.2019.162604

Computer simulation for neutrinoless double beta decay signal and background events

Wrote PMT model that allows us to vary two **Information Parameters**:

- **Photocoverage (PC)**
- **Quantum Efficiency (QE)**

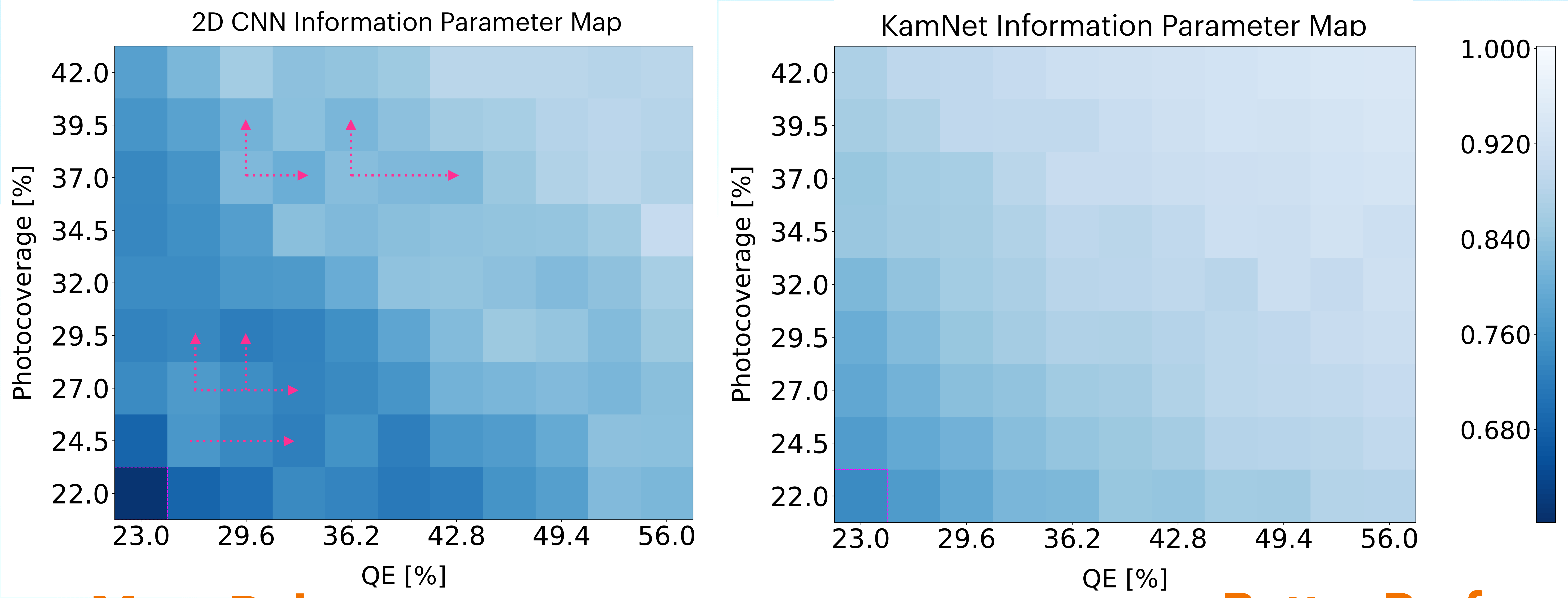
Benchmark model performance under different input conditions

better detector, more information in data

20

AI/ML

- The exact model to use depends on how you pre-process your data into the input format
- **Convolutional Neural Network (CNN)** is a good model for multiple data types in general
- Enhance neural network's performance by encoding symmetries with **Geometric Deep Learning**



A. Li et al, Phys. Rev. C **107**, 014323 (2023)

More Robust

Smoother transition from low to high information parameters
Every bit of additional information is absorbed by KamNet

Better Performance

Across entire map, 61% → 74% background rejection at KamLAND-Zen hardware configuration

21

Nuclear Physics

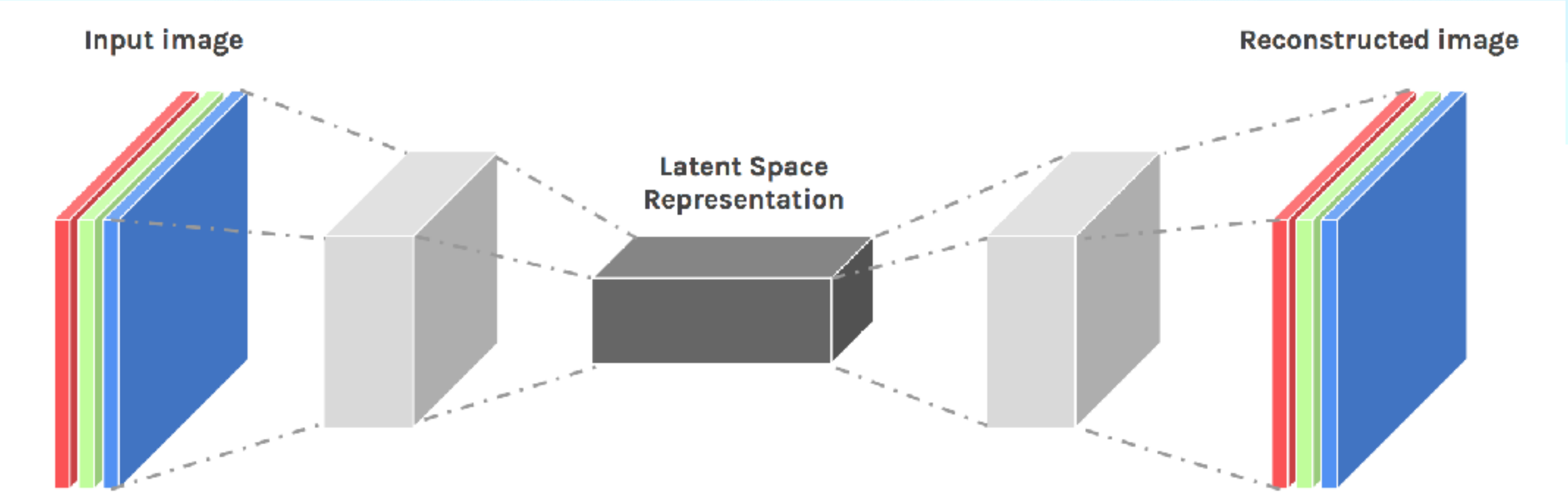
Q3: Can I use deep learning methods for **event simulation**?

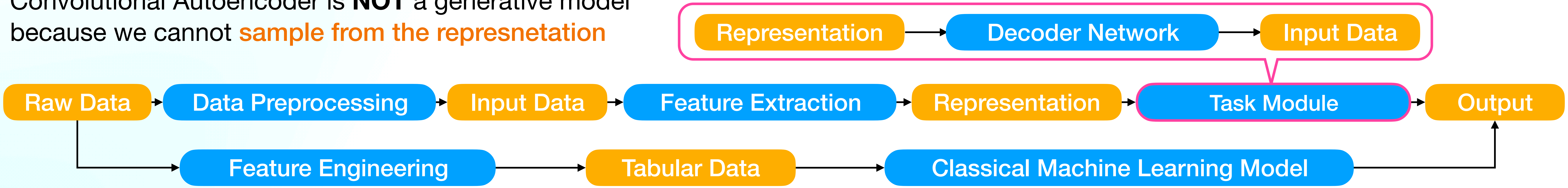
AI/ML

Yes! Use Generative Models: Variational Autoencoder (VAE), Generative Adversarial Network (GAN), or Diffusion Model

Convolutional Autoencoder

- Concatenating two CNNs back-to-back
  - Encoder: convolution layers
  - Decoder: deconvolution layers
- Train by minimizing MSE loss between input image and reconstructed image
- Each image is encoded into the representation, and can be reconstructed from the representation by the decoder
- Convolutional Autoencoder is **NOT** a generative model because we cannot sample from the represnetation





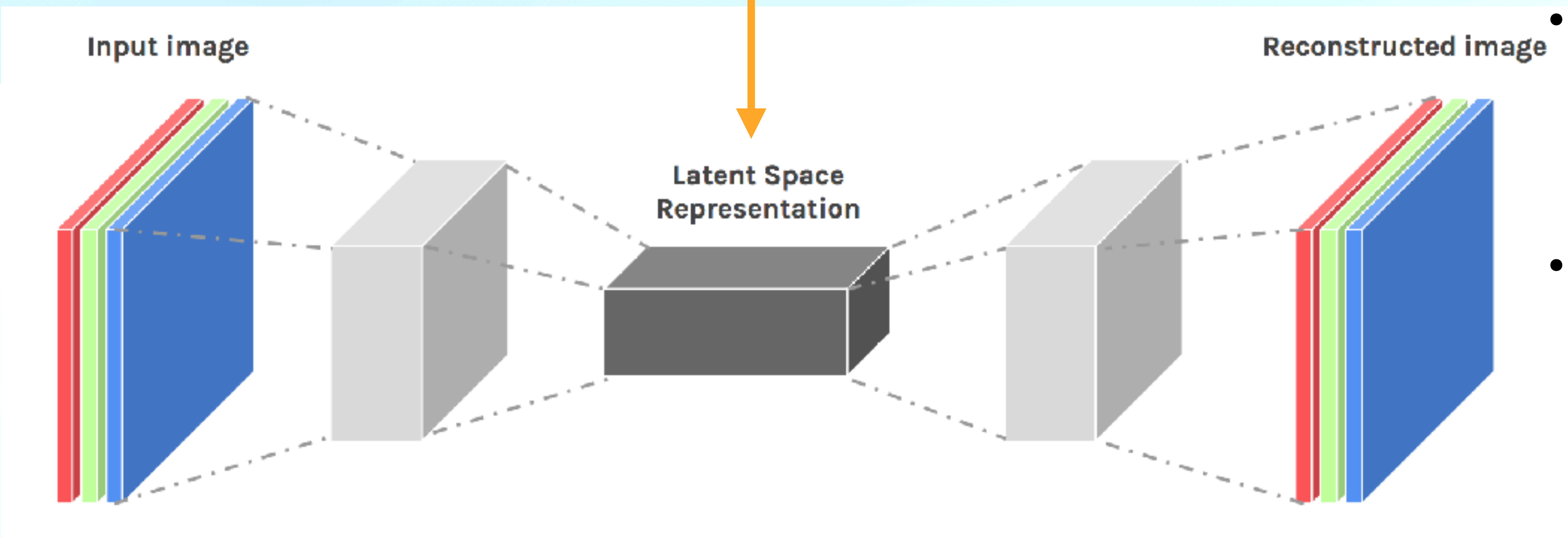
Nuclear Physics

Q3: Can I use deep learning methods for **event simulation**?

AI/ML

Yes! Use Generative Models: Variational Autoencoder (VAE), Generative Adversarial Network (GAN), or Diffusion Model

KL Divergence Loss: $D_{KL}(\mathcal{N}(x|\mu_1,\sigma_1)||\mathcal{N}(x|0,1)) = -log(\sigma_1) + \sigma_1^2 + \mu_1^2 - 0.5$



MSE Loss

Variational Autoencoder

- In Convolutional Autoencoder, the representation contains all information needed to reconstruct an image
- But the representation does not follow any particular distribution
  - This means we cannot sample from it to generate new events
- Variational Autoencoder add another loss to regulate the latent space vector
  - KL Divergence: measuring the distance between two probability distributions
  - Additional loss term that regulates the representation to follow a Gaussian with 0 mean and 1 standard deviation

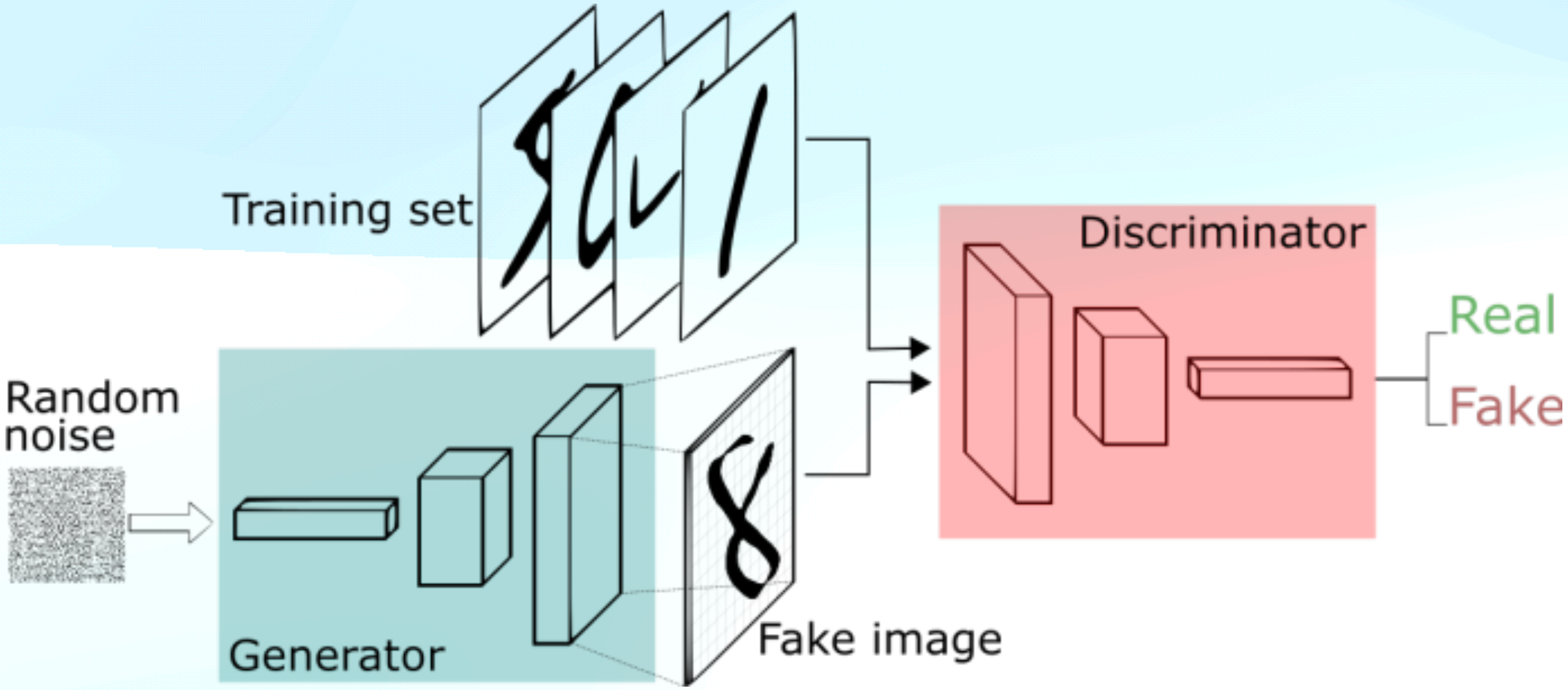
Input Data → Encoder → Representation → Sample → Decoder ⋯→ Input Data

Nuclear Physics

Q3: Can I use deep learning methods for **event simulation**?

AI/ML

Yes! Use Generative Models: Variational Autoencoder (VAE), Generative Adversarial Network (GAN), or Diffusion Model



Generative Adversarial Networks

- A Decoder/Generator will generate a fake data from sampled random noise
- A Encoder/Discriminator will classify whether the input image is real or fake
- Adversarial Training: generator and discriminator fight each other during training
  - $E_x(log(D(x)) + E_z(1 - D(G(z)))$

Nuclear Physics

Q4: Now I train a machine learning classifier with with simulated events (either with GEANT4 or generative model). But my **simulated event** looks different from **real detector event**. What should I do?

AI/ML

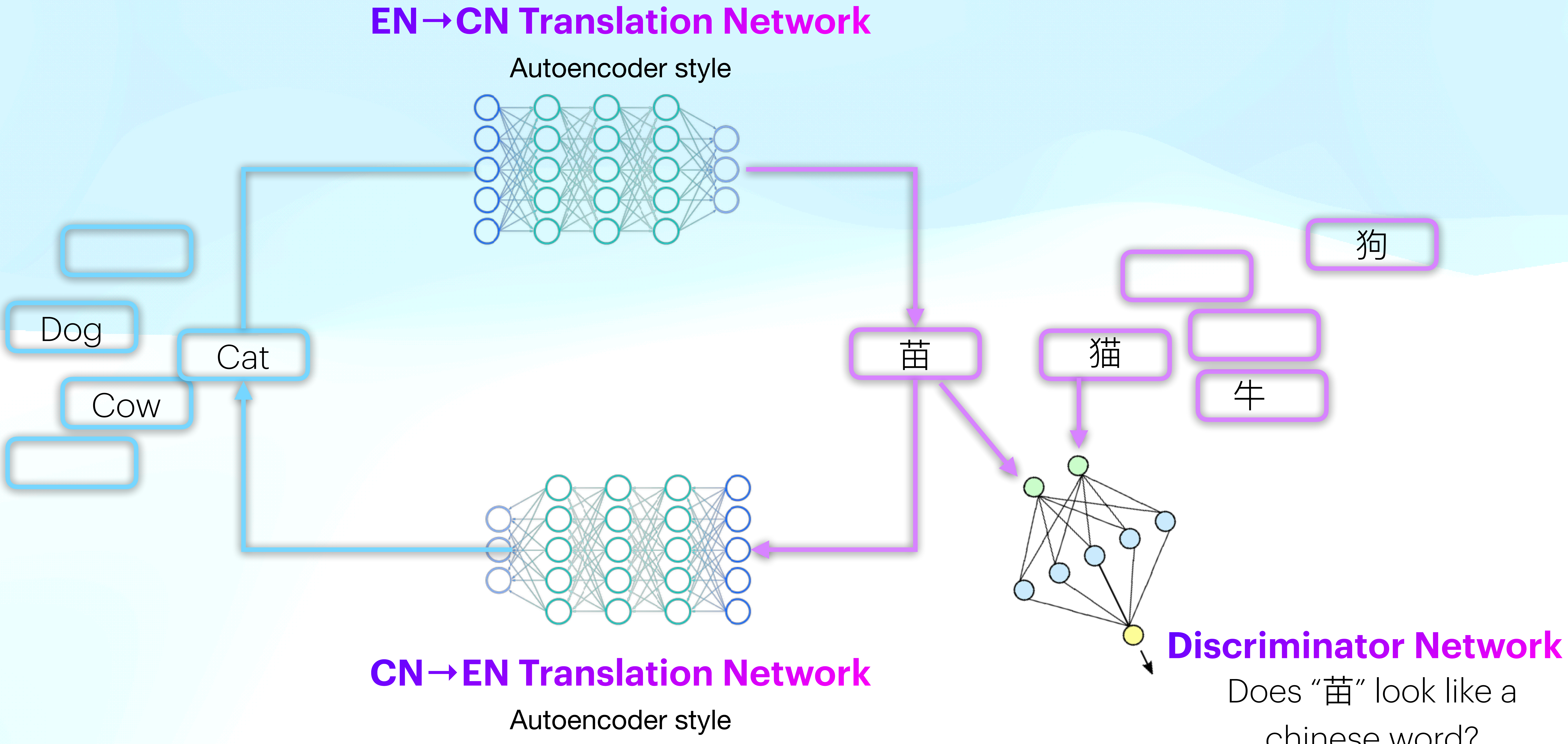
- Build a Cycle GAN to perform unpaired translation between simulation and data

Think of simulated events and real events as two different languages …

- Simulation tuning: building a model that translate simulated events to real detector events
- Ideally, we will train our translation model between paired events, but those pairs are difficult to obtain

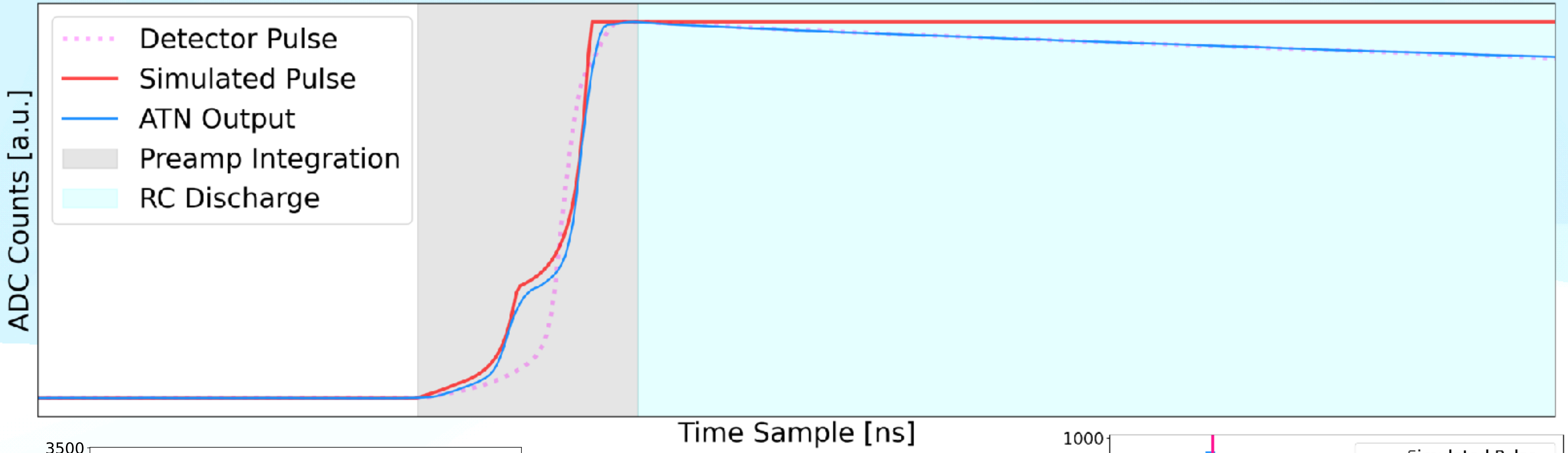
AI/ML

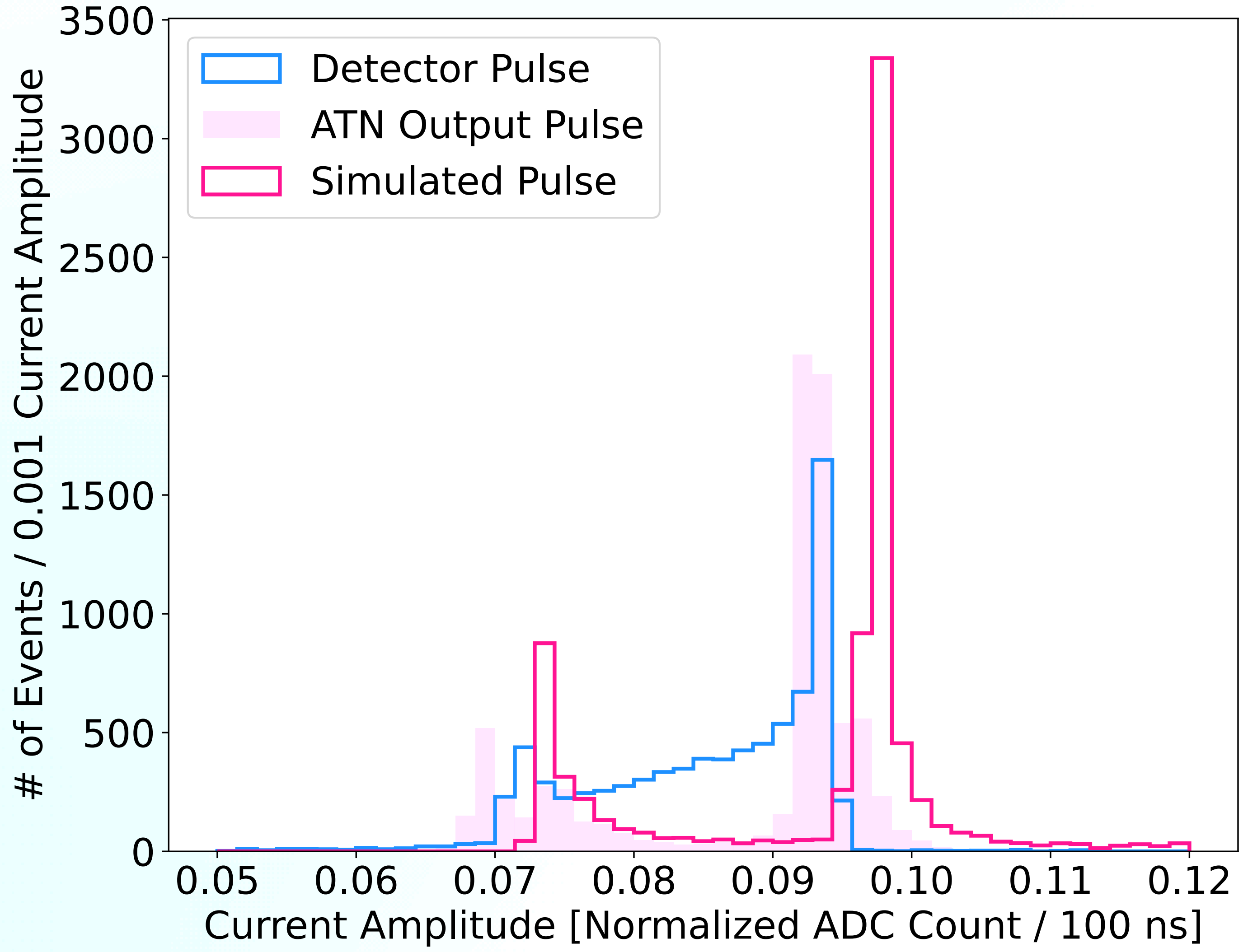
- Build a Cycle GAN to perform unpaired translation between simulation and data



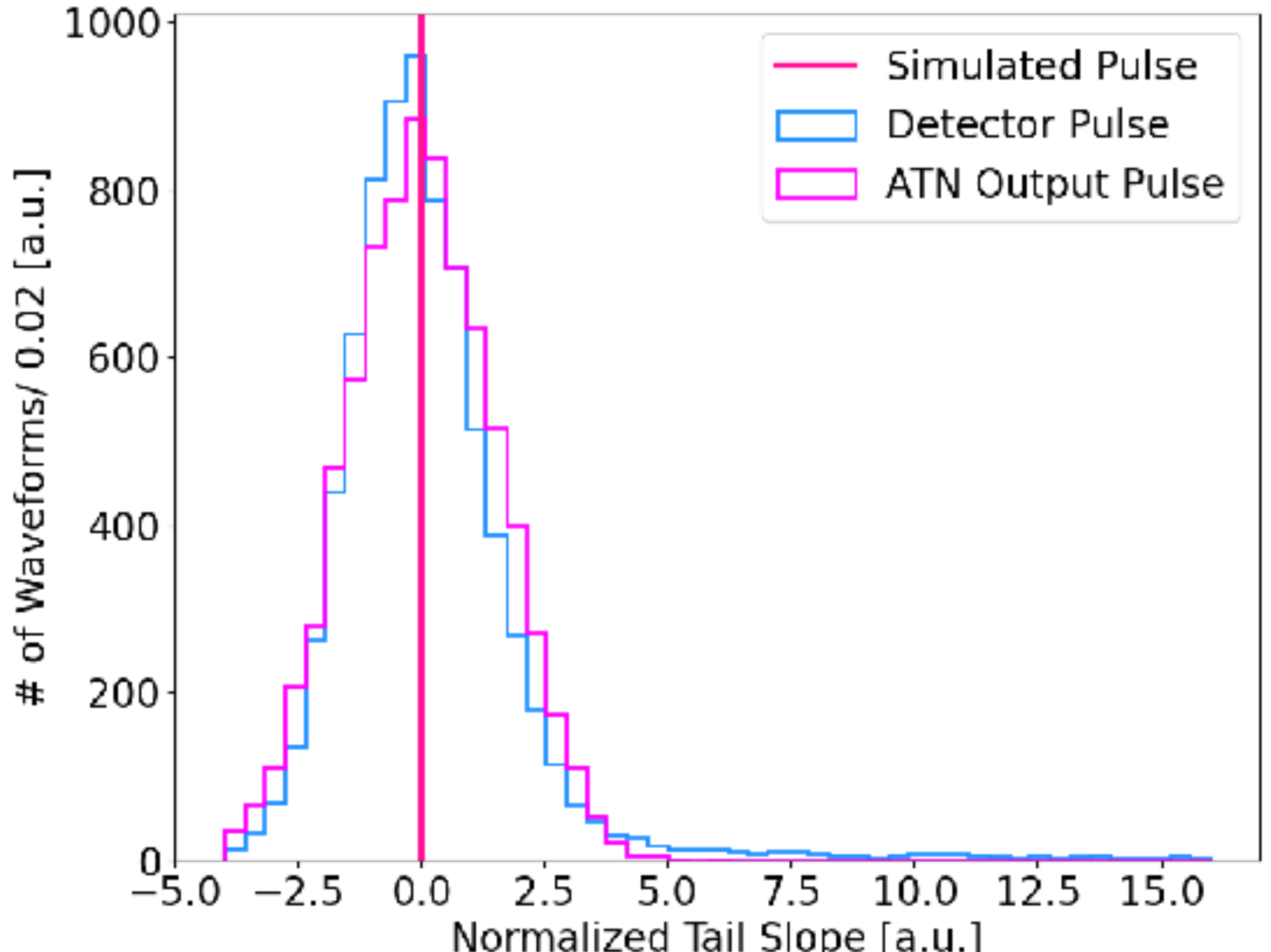
AI/ML

- Build a Cycle GAN to perform unpaired translation between simulation and data





A. Li, J.Gruszko, et al.
NeurIPS 22 ML4PS Workshop

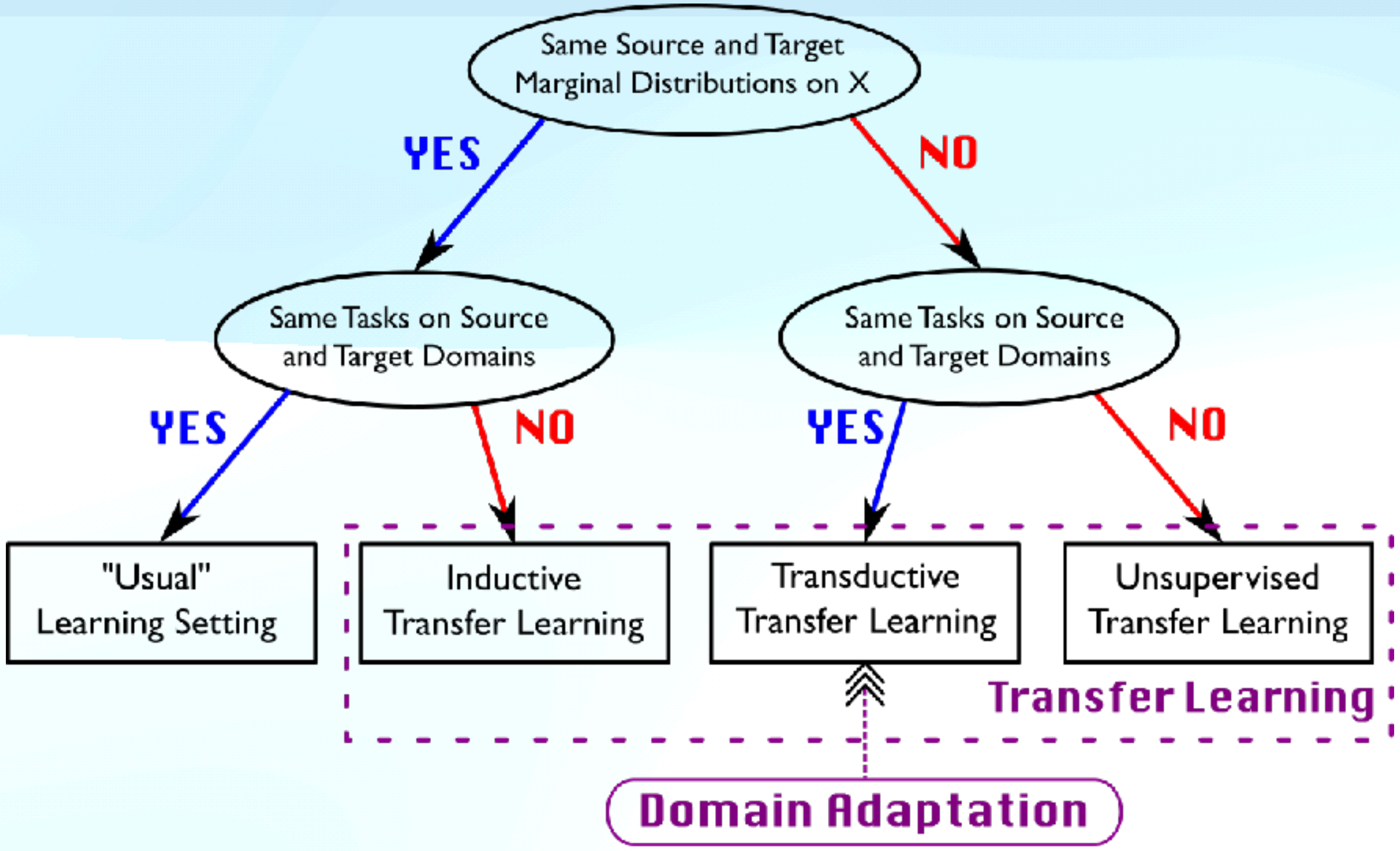


Nuclear Physics

Q4: Now I train a machine learning classifier with with simulated data (either with GEANT4 or generative model). But my **simulated data** looks different from **real detector data**. What should I do?

AI/ML

- Build a Cycle GAN to perform unpaired translation between simulation and data
- Domain Adaptation between simulated and real detector data



Transfer Learning

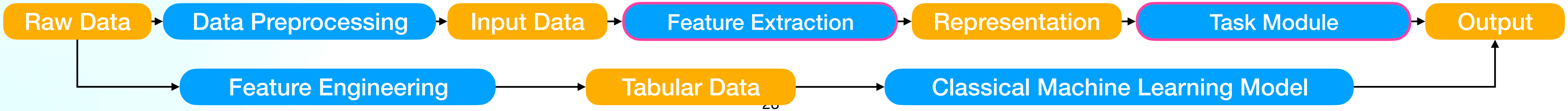
Source Domain: the domain from which the initial training data is drawn.

- Data are typically labelled
- Simulated data in context of NP

Target Domain: the domain to which the model needs to be adapted.

- Data are typically unlabelled and looks different from the source domain
- Real detector data in context of NP

Task: classification or regression or other tasks

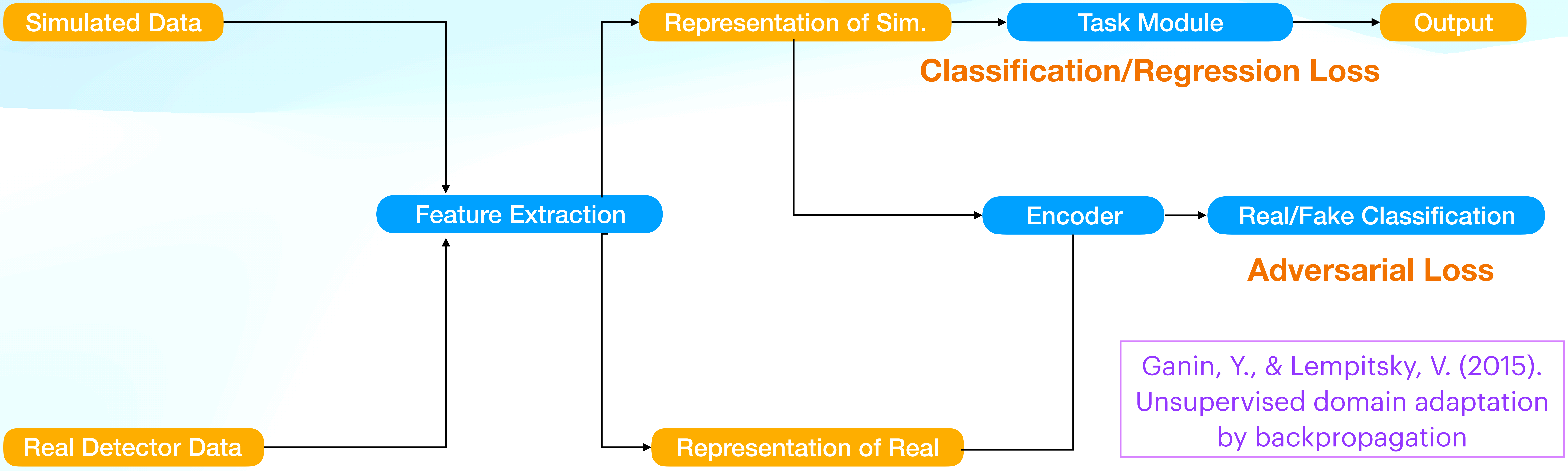


Nuclear Physics

Q4: Now I train a machine learning classifier with with simulated data (either with GEANT4 or generative model). But my **simulated data** looks different from **real detector data**. What should I do?

AI/ML

- Build a Cycle GAN to perform unpaired translation between simulation and data
- Domain Adaptation between simulated and real detector data



Ganin, Y., & Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation

Nuclear Physics

Q5: Can I directly train my model on **real detector data**?

AI/ML

Yes! But real detector data is oftentimes unlabelled. This means we have to adopt an unsupervised **representation learning** approach, which is quite different from what we have done before.

Supervised Learning

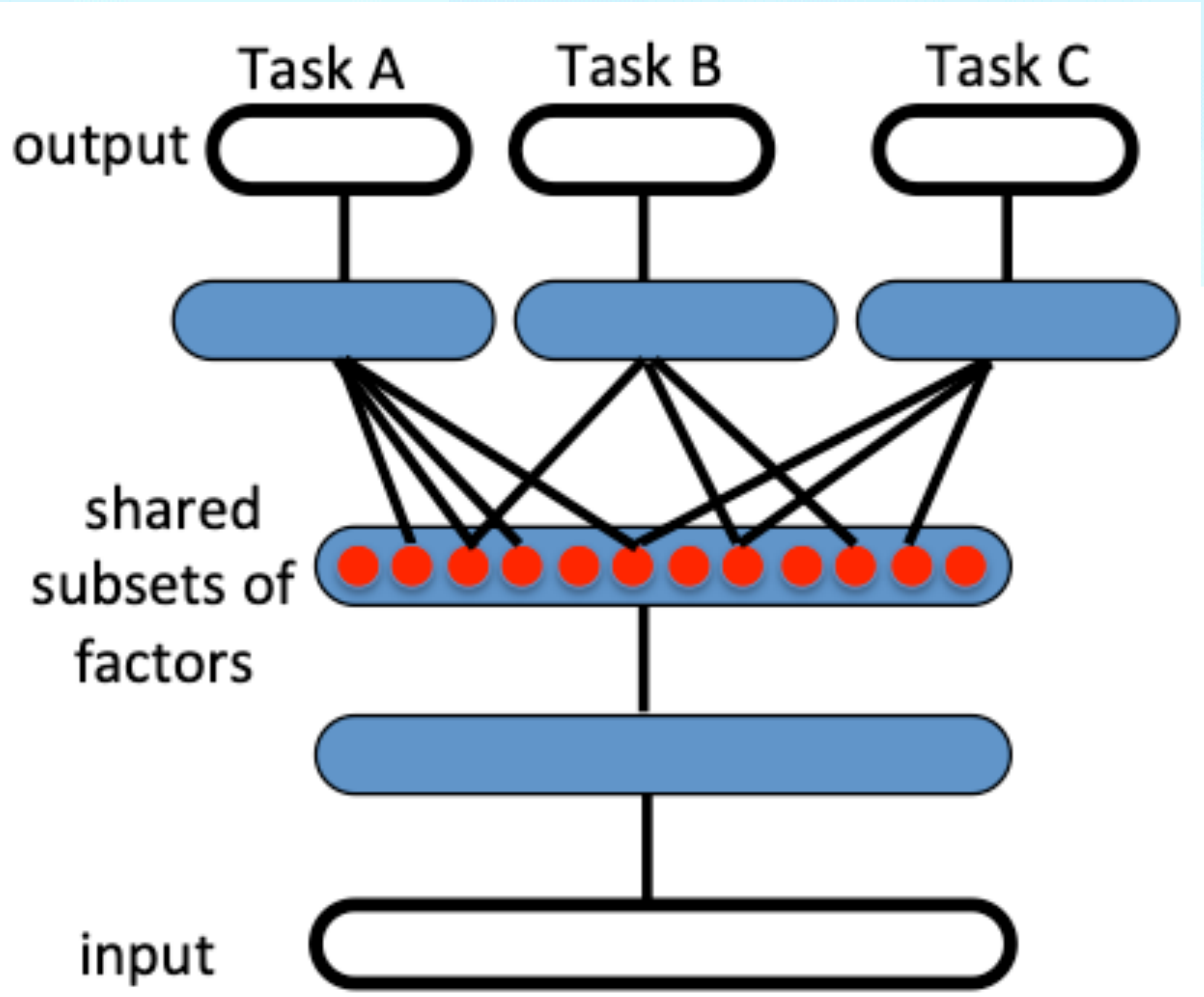
In this setup, the task is defined by the **label**

- With signal(1) vs. background(0) as label we can build a background cut
- With energy as label we can build a energy reconstruction fitter
- With position as label we can build a energy reconstruction fitter

Representation Learning

In this setup, since there is no label…

- The goal is to learn a good **representation** that encode important informations in our data
- This representation is **task-agnostic:** generalizable to different tasks

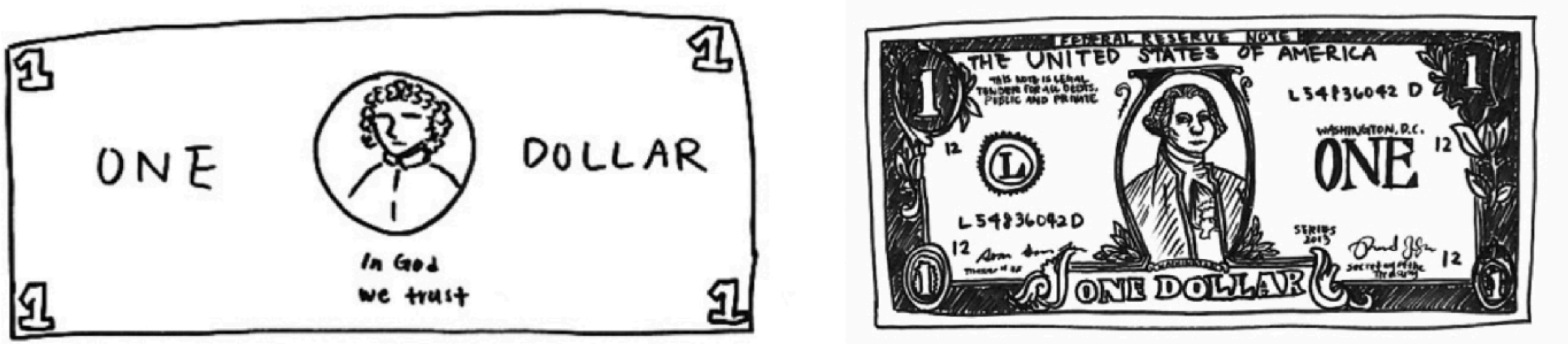


Nuclear Physics

Q5: Can I directly train my model on **real detector data**?

AI/ML

Yes! But real detector data is oftentimes unlabelled. This means we have to adopt an unsupervised **representation learning** approach, which is quite different from what we have done before.



Left: Drawing of a dollar bill from memory. Right: Drawing subsequently made with a dollar bill present. Image source: Epstein, 2016

Learning to generate pixel-level details is often unnecessary; learn high-level semantic features with pretext tasks instead

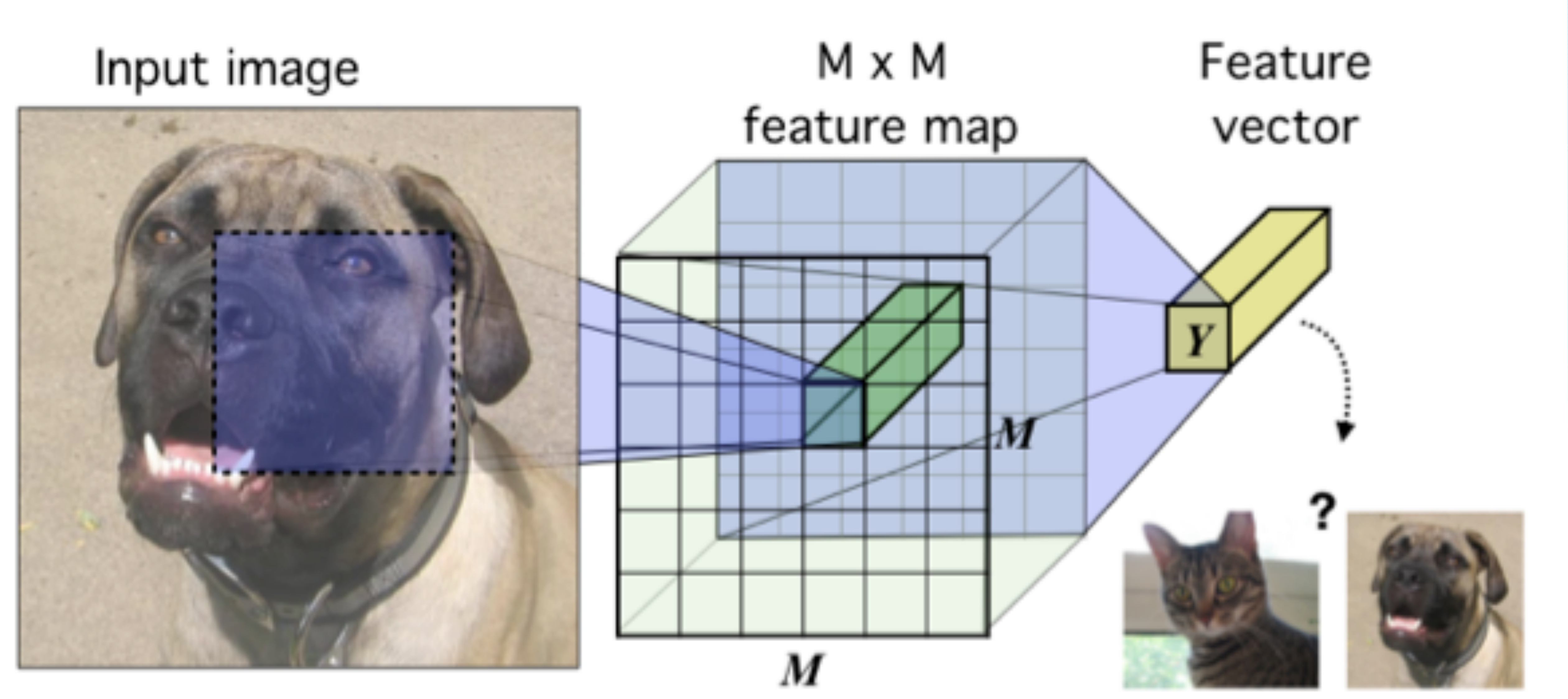
Source: Anand, 2020

Hjelm, R. D. et al. Learning deep representations by mutual information estimation and maximization. *International Conference on Learning Representations (ICLR)*.

① Feature Extractor

Using CNN to convert the image into a feature map

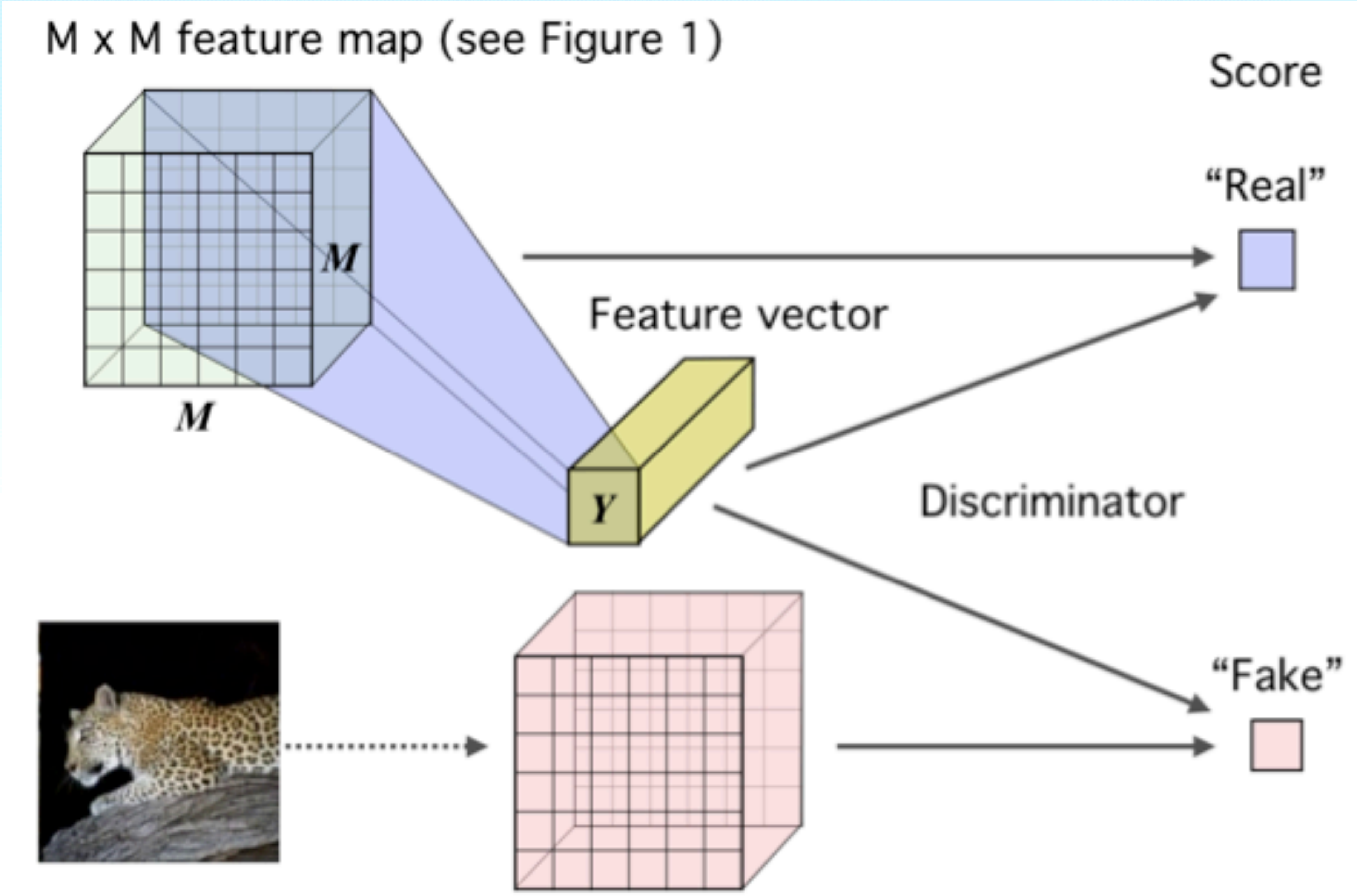
Using fully connected layer to summarize feature map into a feature vector



② Mutual Information

A measure of correlation between two probability distributions

In this model, we can calculate the MI between feature map and feature vector



Representation Space

Probability space where the feature vectors live in

Feature vector should contain high-level semantic information from feature map if well trained

③ Contrastive Training

Maximize MI if the input ( , ) are the same

Minimize MI if the input ( , ) are different

Fig. A→D: the length of the "band" is the time it takes for waveforms to reach maximum

Fig. D vs. Fig E: the width of the "band" represents the number of steps in waveforms

Fig. F: the "ring island" are slow-rounded-top waveforms caused by passivated surface

Nuclear Physics

Q6: I built a machine learning model within my collaboration and attempted to use it for my analysis. But my collaborators do not like it. They say that you cannot trust the decision of ML model since it's a **black box**. What should I do?

AI/ML

- **Uncertainty Quantification: bootstrapping** methods and **uncertainty-aware** machine learning models
-

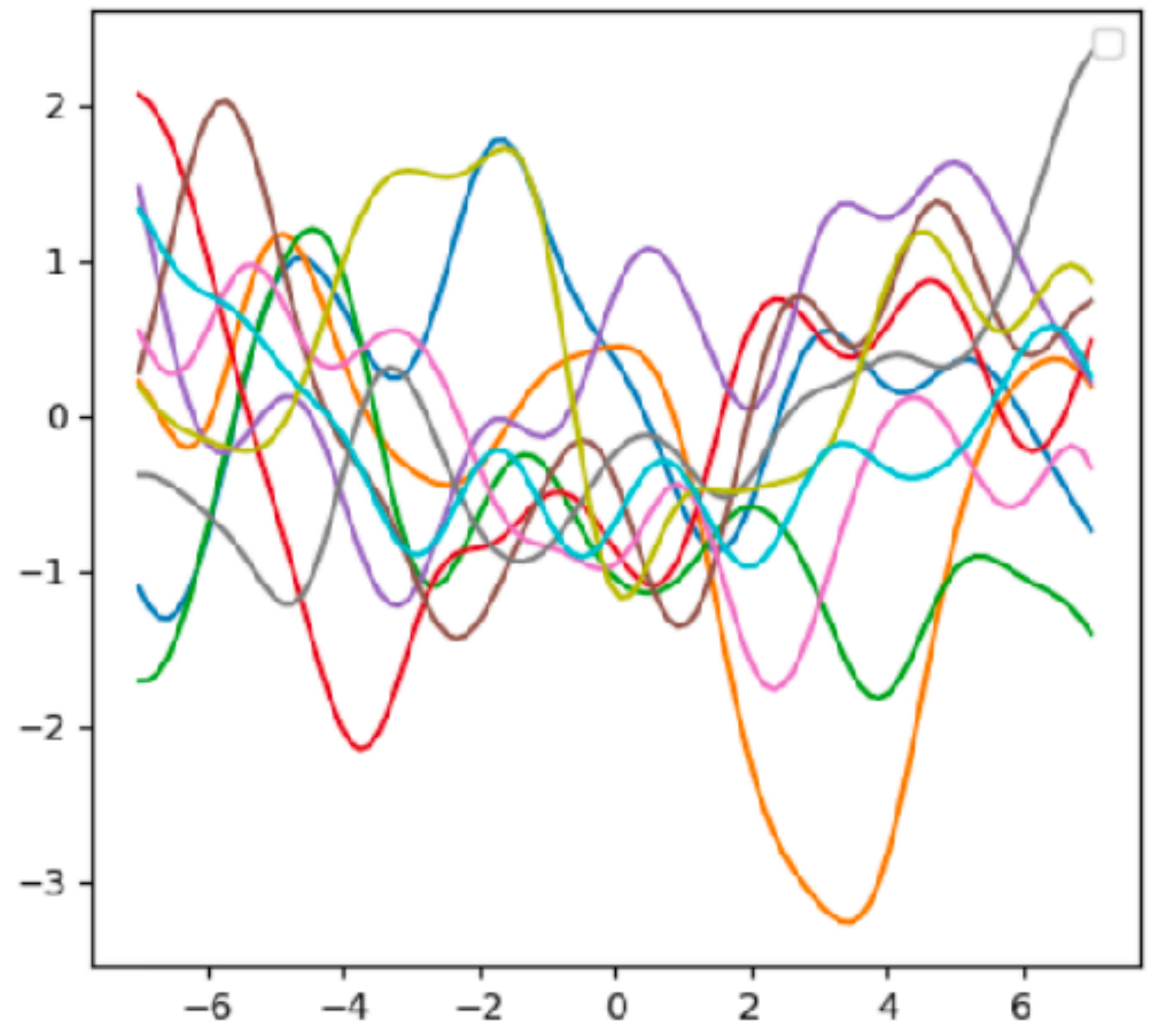
AI/ML

- **Uncertainty Quantification: bootstrapping** methods and **uncertainty-aware** machine learning models
-

predict the value of y_n for a new value of x_n where $f:\{x_n\}^N \rightarrow \{y_n\}^N$ maps the input space to the output space

Let's start with a distribution of all possible functions that, could have produced our data (without actually looking at the data!).

$$f(\cdot) \sim p(f(\cdot)) \sim \mathcal{N}(\mu(.), \sigma(.))$$



A Gaussian process is a probability distribution over possible functions that fit a set of points.

Credit: A. Shuetz

AI/ML

- **Uncertainty Quantification: bootstrapping** methods and **uncertainty-aware** machine learning models
-

A gaussian process needs two ingredients:

$$f(x) \sim GP(m(x), k(x, x'))$$

- a **mean function**

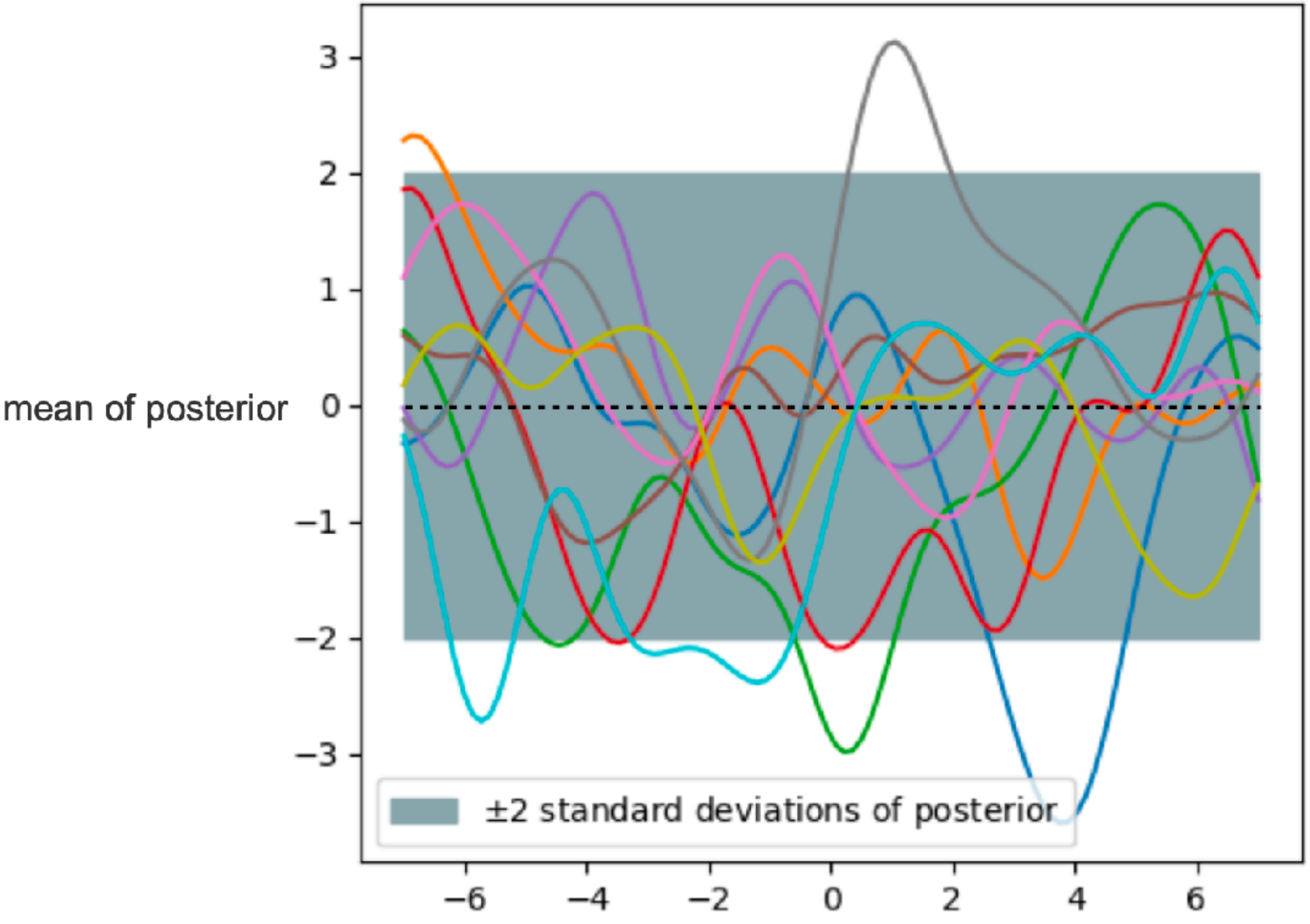
$$m(x) = \mathbb{E}[f(x)]$$

(mean at any point of the input space)

- a covariance function (**kernel**)

$$k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))^T]$$

(how likely it is the functions are similar)



Credit: A. Shuetz

Nuclear Physics

Q6: I built a machine learning model within my collaboration and attempted to use it for my analysis. But my collaborators do not like it. They say that you cannot trust the decision of ML model since it's a **black box**. What should I do?

AI/ML

- **Uncertainty Quantification: bootstrapping** methods and **uncertainty-aware** machine learning models
-

GP prior | GP posterior

Other uncertainty-aware machine learning models:

Bayesian Neural Network, Monte Carlo Dropout, Deep Ensemble, Quantile Regression

Credit: A. Shuetz

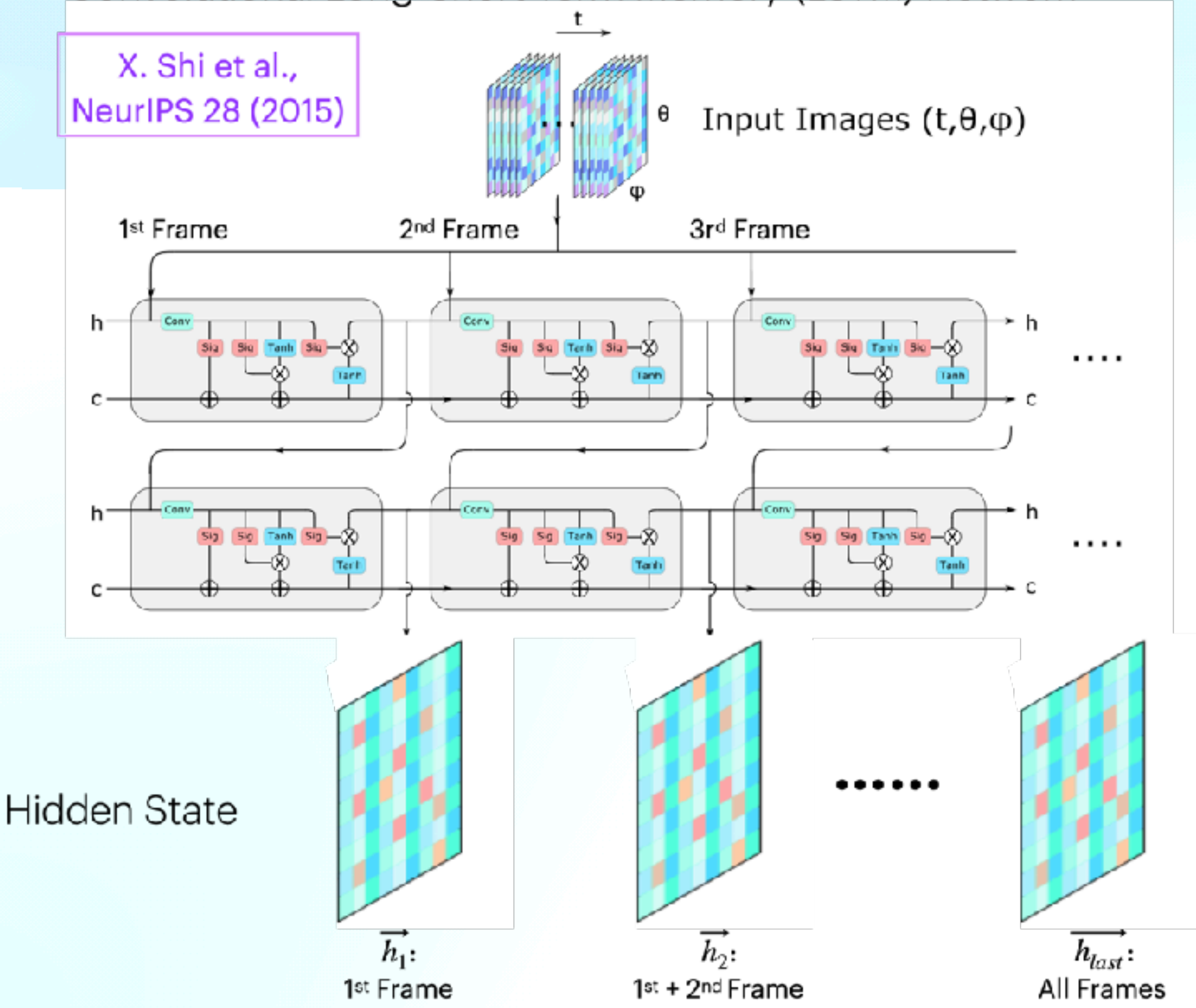
Nuclear Physics

Q6: I built a machine learning model within my collaboration and attempted to use it for my analysis. But my collaborators do not like it. They say that you cannot trust the decision of ML model since it's a **black box**. What should I do?

AI/ML

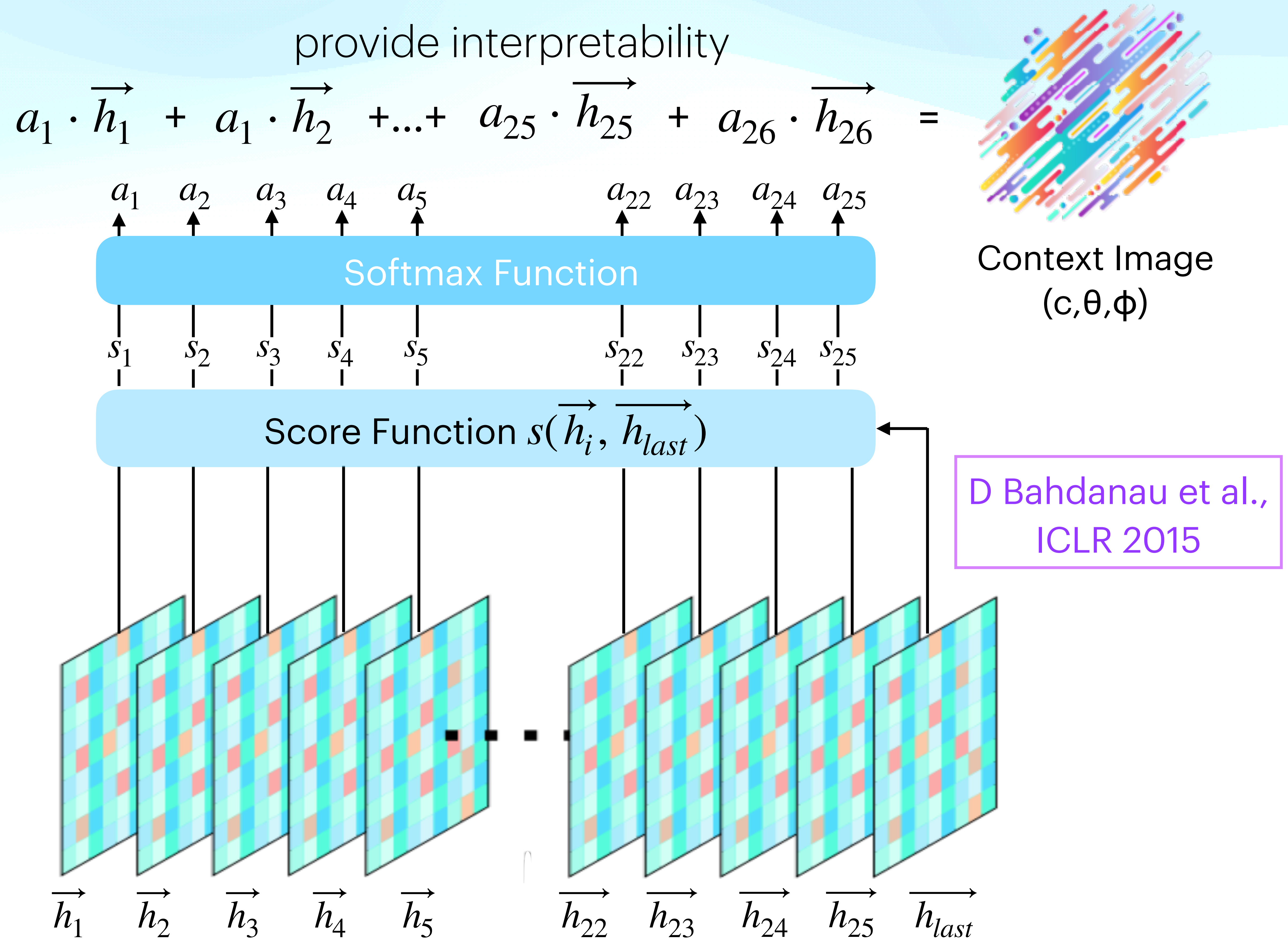
- **Uncertainty Quantification: bootstrapping** methods and **uncertainty-aware** machine learning models
- **Interpretability Study:** understanding the reason behind how neural network makes decisions

ConvLSTM



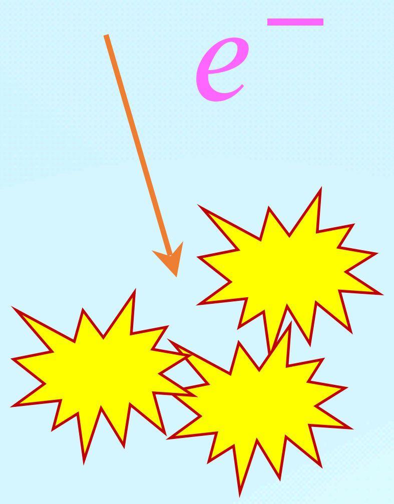
Attention Mechanism

provide interpretability

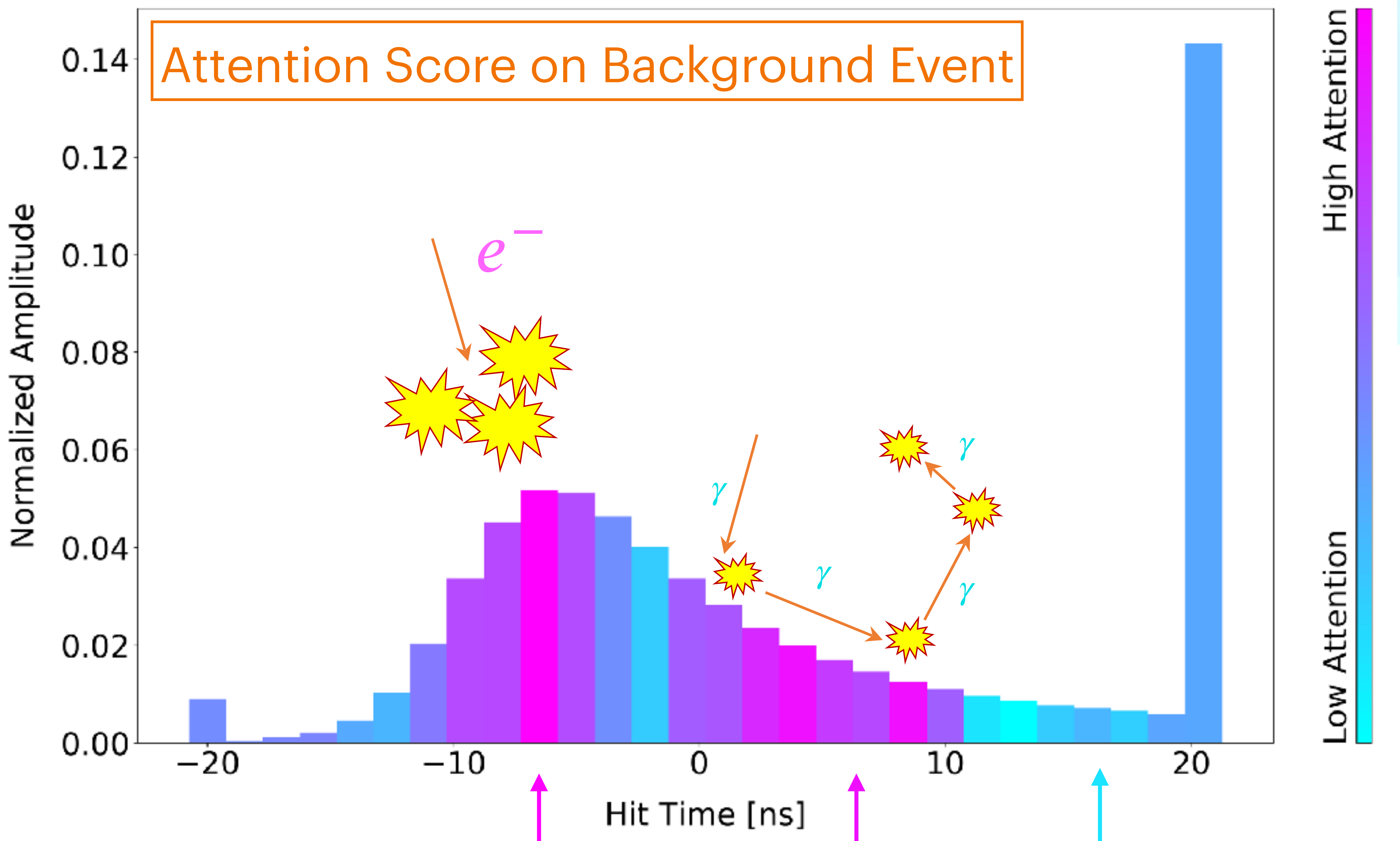


AI/ML

- **Uncertainty Quantification: bootstrapping** methods and **uncertainty-aware** machine learning models
- **Interpretability Study:** understanding the reason behind how neural network makes decisions



- Signal are strictly **single-vertex events**
  - All energy deposited almost immediately



- Most backgrounds are **closely-spaced multi-vertex events**
  - part of event energy is deposited by cascading γs that slightly alter event topology

High Attention: Important

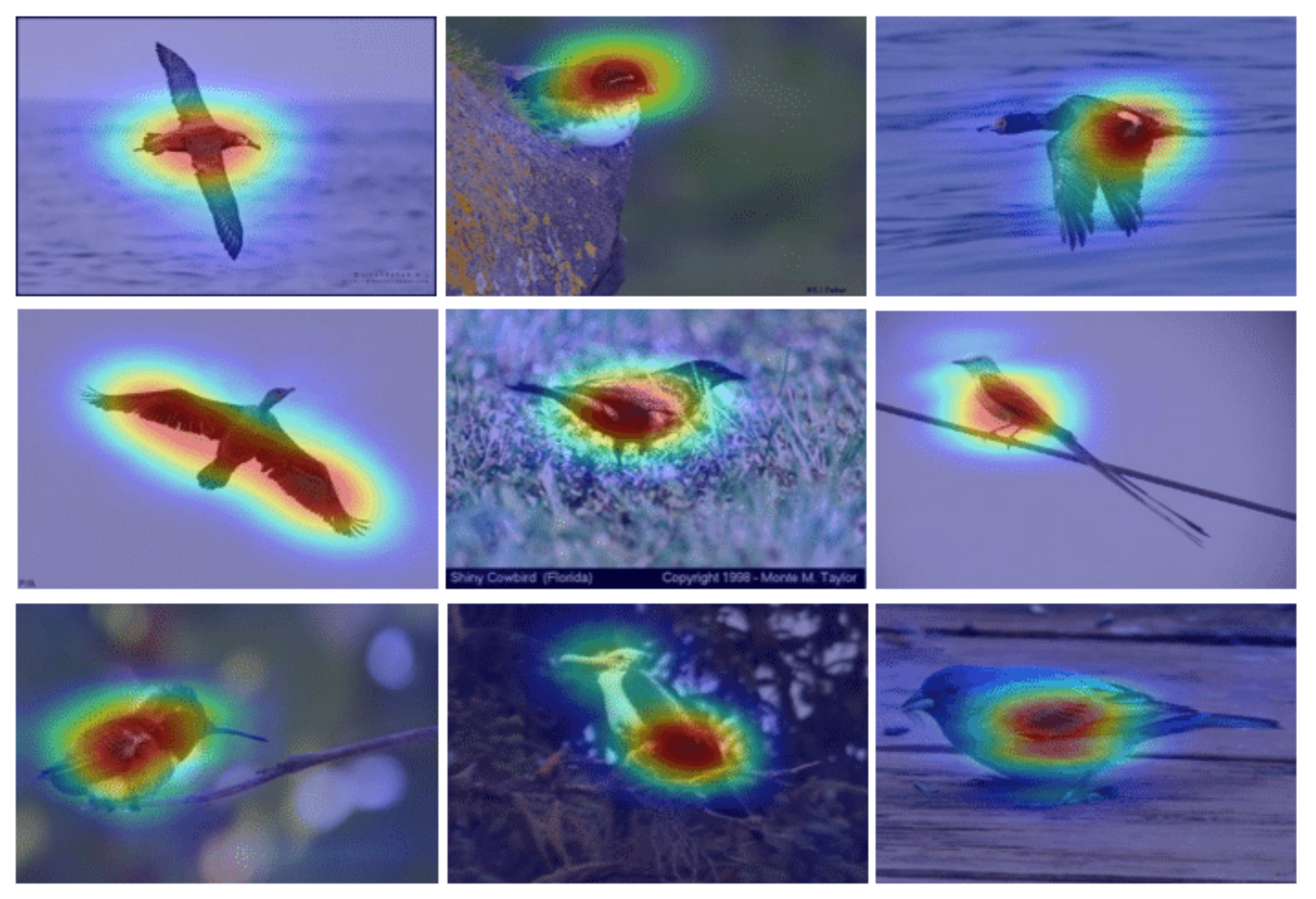
Low Attention: Unimportant

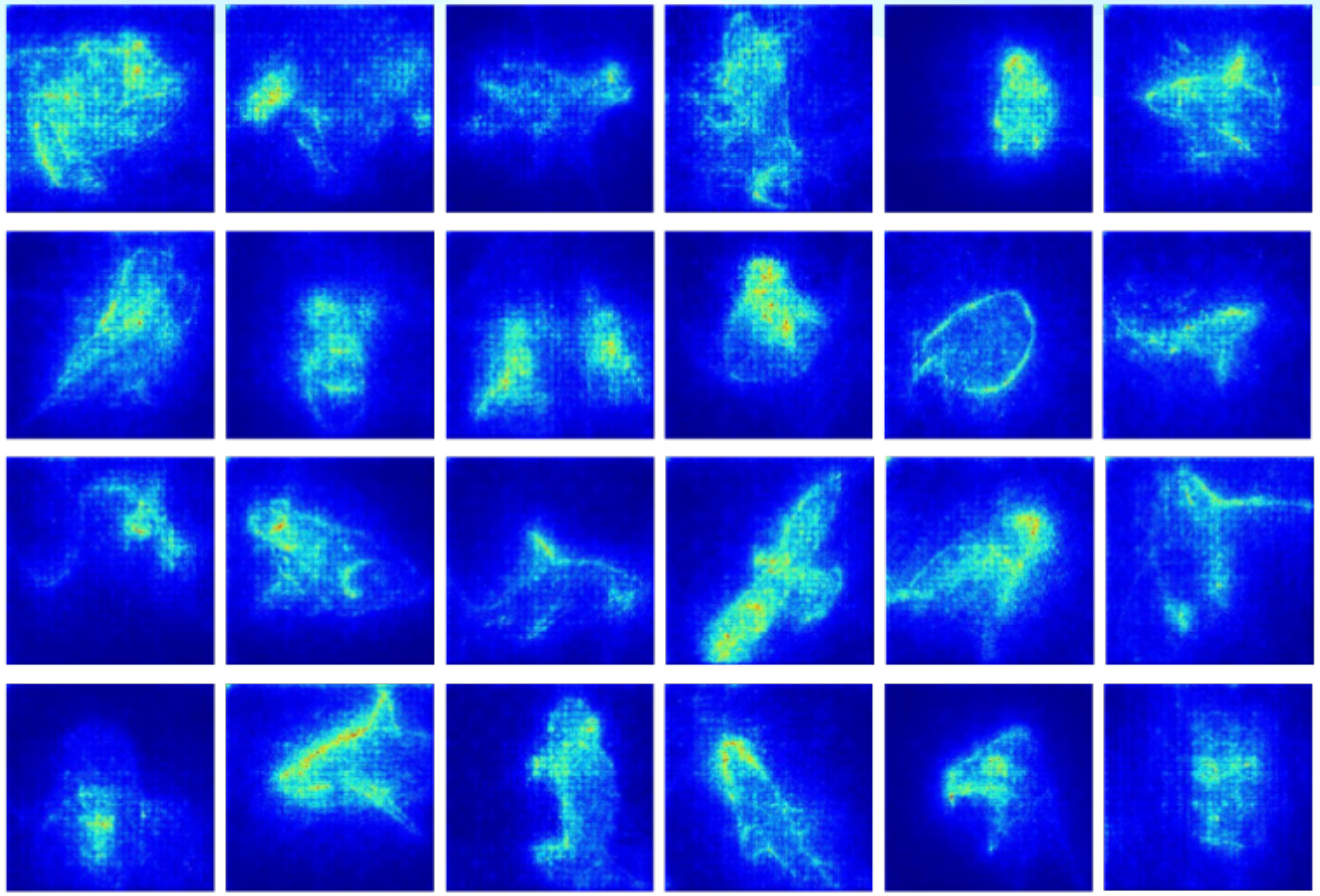
AI/ML

- **Uncertainty Quantification: bootstrapping** methods and **uncertainty-aware** machine learning models
- **Interpretability Study:** understanding the reason behind how neural network makes decisions

Saliency Map

- Gradient-based interpretability technique
- highlight the parts of an input that are most important for a neural network's prediction
- Most suitable for CNN

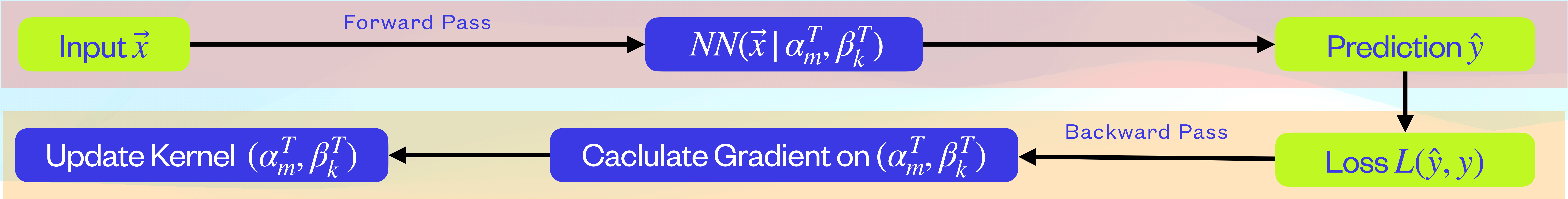




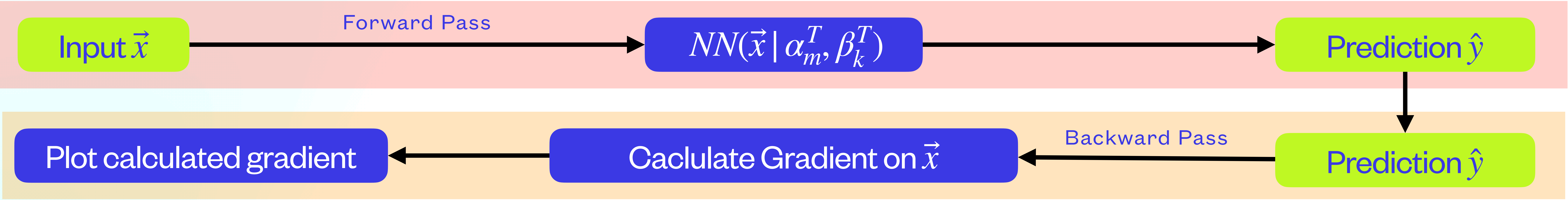
AI/ML

- **Uncertainty Quantification: bootstrapping** methods and **uncertainty-aware** machine learning models
- **Interpretability Study:** understanding the reason behind how neural network makes decisions

Training a neural network:

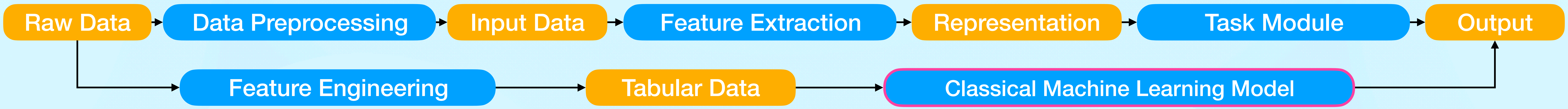


Compute **Saliency Map:**



AI/ML

- **Uncertainty Quantification: bootstrapping** methods and **uncertainty-aware** machine learning models
- **Interpretability Study:** understanding the reason behind how neural network makes decisions



SHAP interpretor:

- Black-box interpretor: can interpret ANY trained machine learning model
- Works better on classical ML model with low-dimensional inputs

Shap

Shapley value:

- Coalitional Game Theory concept
- Represent each player's contribution to the total surplus/deficit assuming they work collaboratively

$$\phi_i(v) = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!}(v(S \cup \{i\}) - v(S))$$

Force Plot:

- For each input event, the SHAP package produces a force plot, analogous to free body diagram
- Shapley value of each feature acts like a force drives the BDT decision to either higher (signal-like) or lower (background-like)
- The value at equilibrium position is then fed to a sigmoid function to produce BDT output

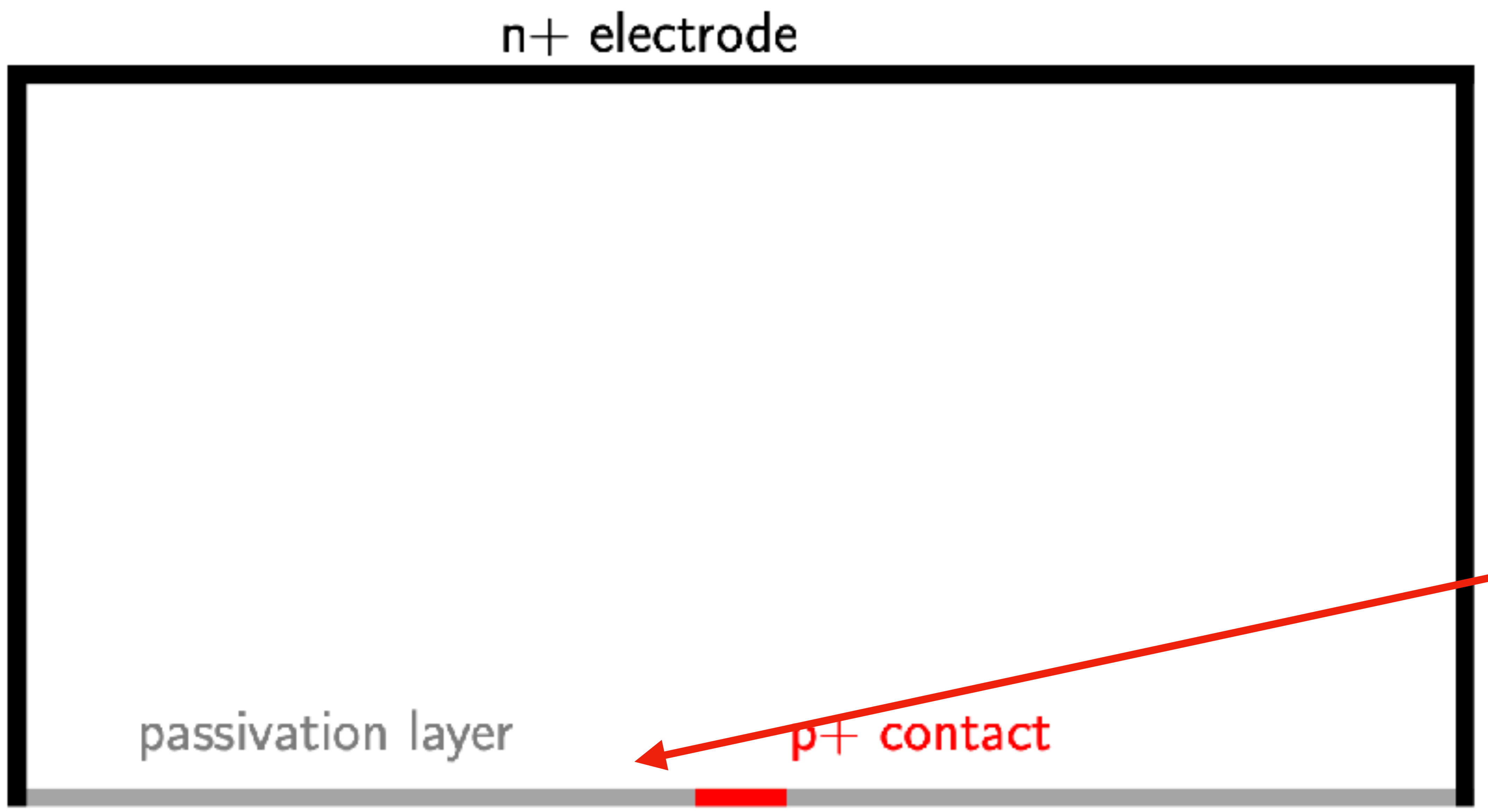
AI/ML

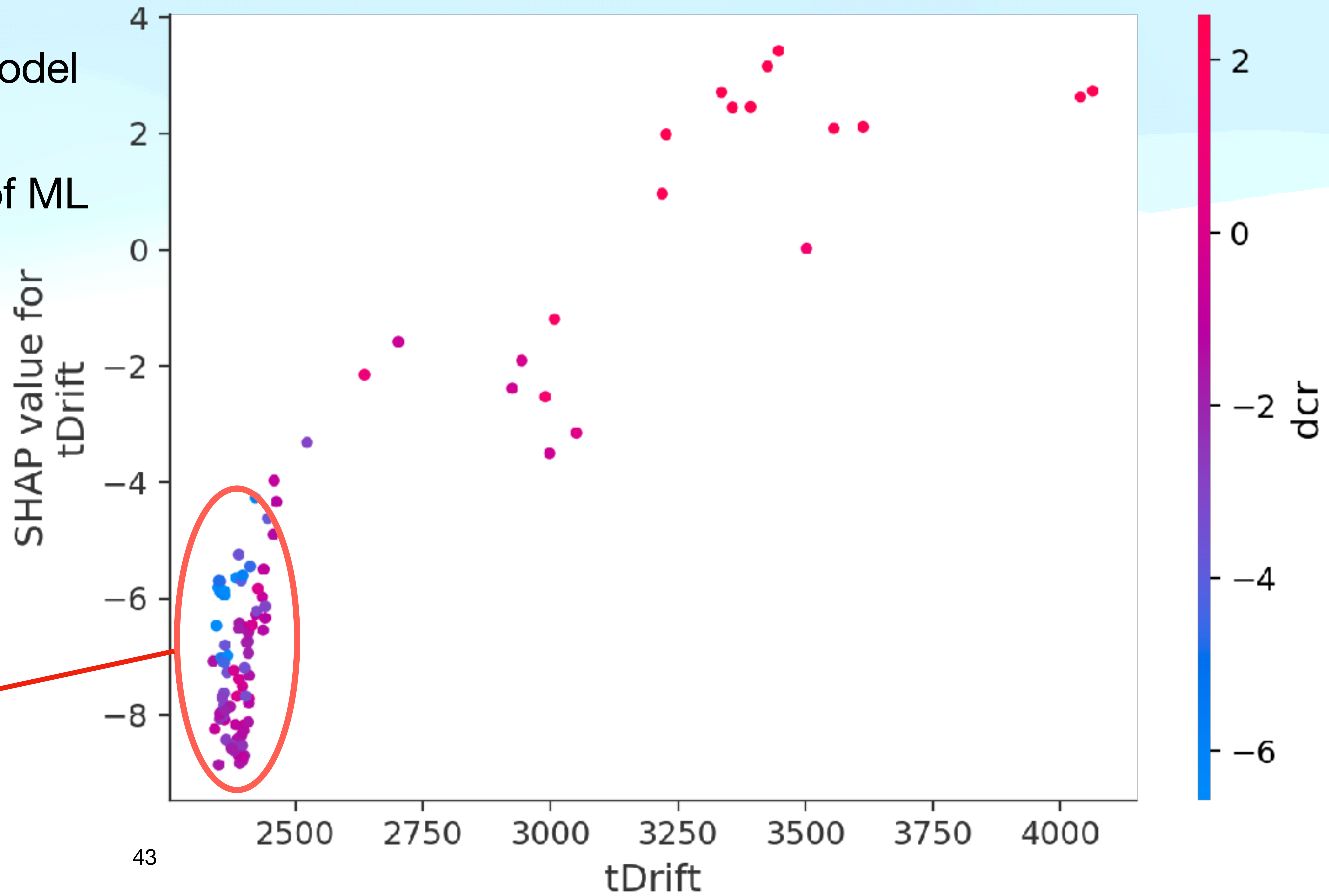
- **Uncertainty Quantification: bootstrapping** methods and **uncertainty-aware** machine learning models
- **Interpretability Study:** understanding the reason behind how neural network makes decisions

Phys.Rev.C 107 (2023) 1, 014321 ArXiv: 2207.10710

Learning from the Machine:

- Select data that are correctly classified by ML model but misclassified by traditional method
- Using Shapley value to study the driving factor of ML decision





Connecting Dots:

An AI Cookbook for Nuclear Physics

Nuclear Physics AI/ML

Q1: In Lecture 1, we started from raw MAJORANA DEMONSTRATOR waveforms, the lowest level of HPGe detector. Do we always have to start from **low level data**?

- No, we can start from higher level parameters with a procedure called **Feature Engineering**

Nuclear Physics AI/ML

Q2: My experiment does not produce short waveforms/time series data like MAJORANA DEMONSTRATOR does, it produces more complicated **high-dimensional data**. what should I use as my feature extraction network?

- The exact model to use depends on how you pre-process your data into the input format
- **Convolutional Neural Network (CNN)** is a good model for multiple data types in general
- Enhance neural network's performance by encoding symmetries with **Geometric Deep Learning**

Nuclear Physics AI/ML

Q3: Can I use deep learning methods for **event simulation**?

- Yes! Use Generative Models: Variational Autoencoder (VAE), Generative Adversarial Network (GAN), or Diffusion Model

Connecting Dots:

An AI Cookbook for Nuclear Physics

Nuclear Physics AI/ML

Q4: Now I train a machine learning classifier with with simulated events (either with GEANT4 or generative model). But my **simulated event** looks different from **real detector event**. What should I do?

- Build a Cycle GAN to perform unpaired translation between simulation and data
- Domain Adaptation between simulated and real detector data

Nuclear Physics AI/ML

Q5: Can I directly train my model on **real detector data**?

- Yes! But real detector data is oftentimes unlabelled. This means we have to adopt an unsupervised **representation learning** approach, which is quite different from what we have done before.

Nuclear Physics AI/ML

Q6: I built a machine learning model within my collaboration and attempted to use it for my analysis. But my collaborators do not like it. They say that you cannot trust the decision of ML model since it's a **black box**. What should I do?

- **Uncertainty Quantification: bootstrapping** methods and **uncertainty-aware** machine learning models
- **Interpretability Study:** understanding the reason behind how neural network makes decisions

Some Useful Links

Code

All lecture materials: Link
Jupyter Notebook Code: Link

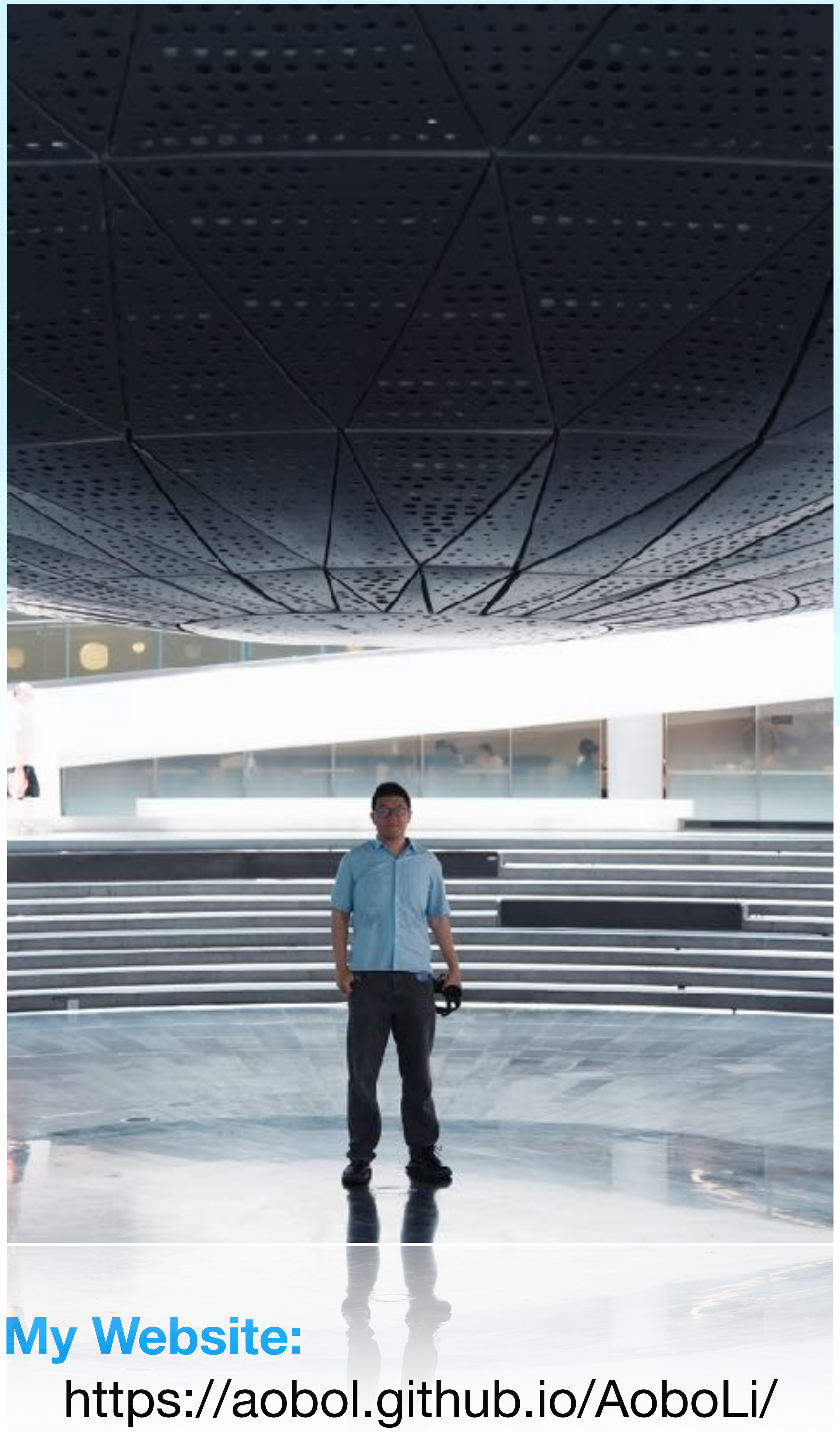
Concept

The Practical Machine Learning
https://pire.gemadarc.org/education/school21/#ai

2024 Summer Bootcamp on Deep Learning and Applications
https://ai-bootcamp2024.github.io/

MIT 6.S191 Introduction to Deep Learning
http://introtodeeplearning.com/

Andrew Ng: Deep Learning Specialization
Link



My Website:
https://aobol.github.io/AoboLi/

My Email Addresses:
aol002@ucsd.edu