AN INTRODUCTION TO MACHINE LEARNING:
NEURAL NETWORKS AND DARK MATTER SEARCHES

VIVIANA GAMMALDI

Associate Professor
University CEU San Pablo.

"Flip Physics 2024", May 21- 24, 2024 - Valencia, Spain.

Picture from the web

ABOUT ME

Why do stars shine in the dark?

2011 —

3 + 2 years - Bachelor Physics / Master Degree Astrophysics
Università degli Studi di Napoli “Federico II” (UNINA), Italy.

4 years - PhD Degree Theoretical Physics
Universidad Complutense Madrid (UCM), Spain.

2015 —

Collaboration grant
Universidad Nacional Autónoma de Mexico (UNAM), Mexico City.

3 years - Postdoc
Scuola Internazionale Superiore di Studi Avanzati (SISSA), Trieste, Italy.

2018 —

5 years - Juan de la cierva formación / incorporación
Universidad Autónoma Madrid (UAM) - Instituto de Fisica Teórica (IFT)

2024 —

Associate Professor
Universidad CEU San Pablo, Madrid.

Why dark matter is dark?

OUTLINE

- **INTRODUCTION TO MACHINE LEARNING**
- **CLASSIFICATION WITH NEURAL NETWORKS**
- **APPLICATION: DARK MATTER SEARCHES**

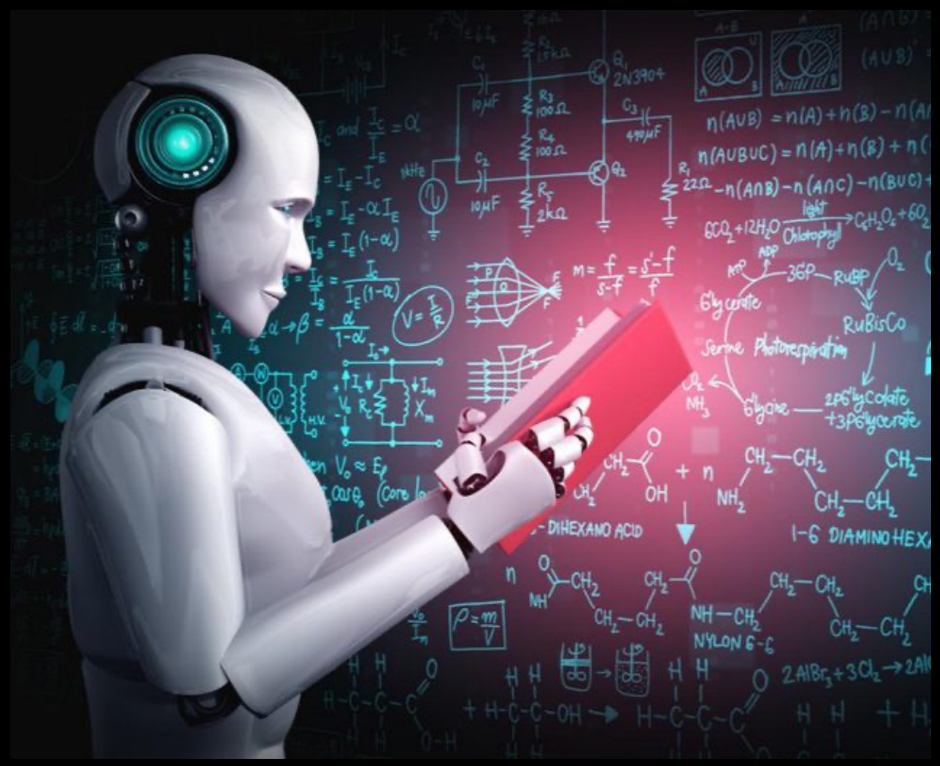
  V.G., B. Záldivar, M. A. Sánchez-Conde, J. Coronado-Blázquez, MNRAS 2023

OUTLINE

- **INTRODUCTION TO MACHINE LEARNING**
- **CLASSIFICATION WITH NEURAL NETWORKS**
- **APPLICATION: DARK MATTER SEARCHES**

  V.G., B. Záldivar, M. A. Sánchez-Conde, J. Coronado-Blázquez, MNRAS 2023

Artificial Intelligence (AI)



The name Artificial Intelligence (AI) was coined in 1956 at the Dartmouth Conference, United States, considered as the seed event of AI as a research field.

<u>The conference proposal:</u>

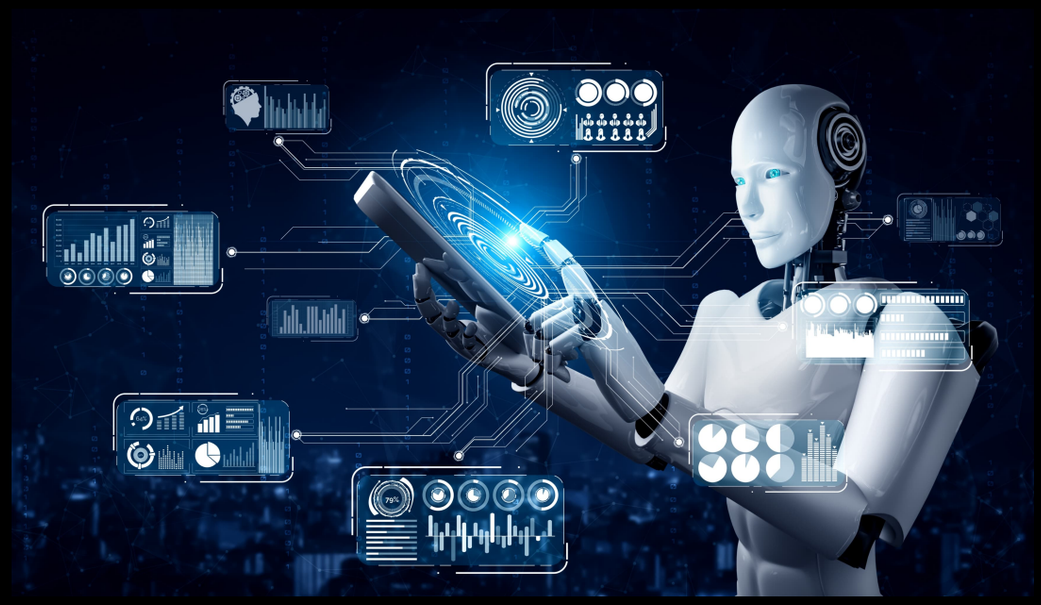
"… The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can be so precisely described that a machine can be made to simulate it. …"

Picture from the web

AI vs Machine Learning (ML)

AI has the main objective of replicating cognitive processes in machines.

In computer science, AI is the combination of algorithms, whose purpose is to create machines that mimic human intelligence to perform tasks and that can improve according to the information they collect.



It currently encompasses a wide variety of subfields, ranging from playing chess, proving mathematical theorems, writing poetry, learning disease diagnosis, etc…

Be careful with many ethical and legal aspects related to AI application!

Machine Learning (ML) is a skill to make systems capable of identifying patterns among the data to make predictions.

Picture from the web

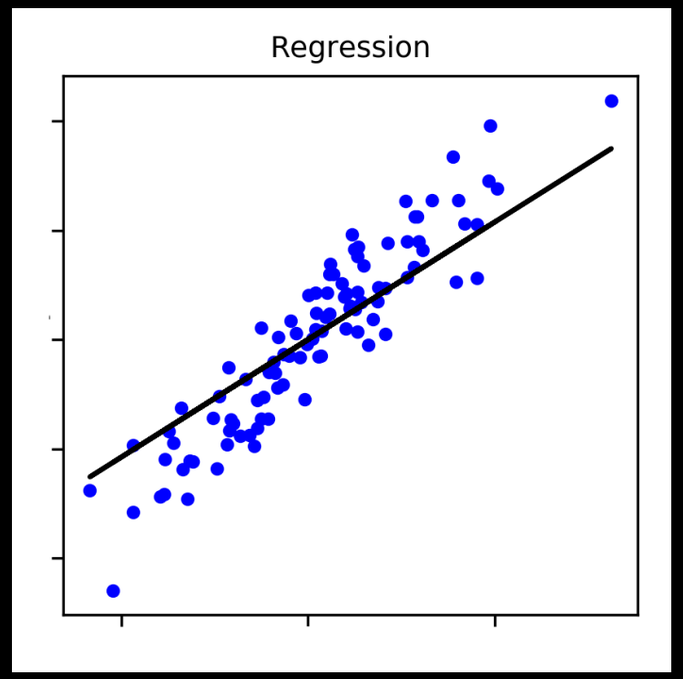
Machine Learning algorithms

- Supervised learning
- Unsupervised learning
- Reinforcement learning

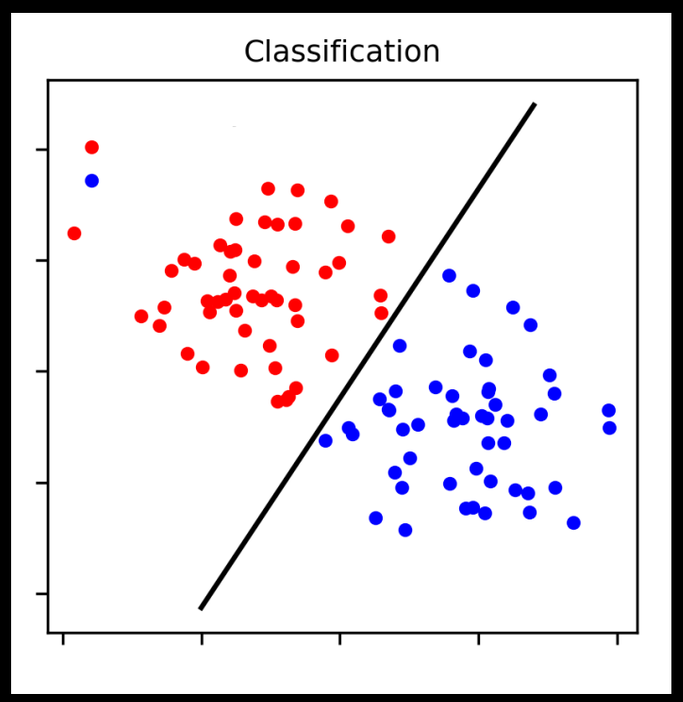
Supervised learning

Supervised learning: uses labeled datasets for training algorithms to predict outputs.

Can be grouped into two principal types:



Regression problem: uses algorithms to measure the relationship between a dependent variable and one or more independent variables. With regression models, the user can make predictions for new expectations, based on a set of data points.

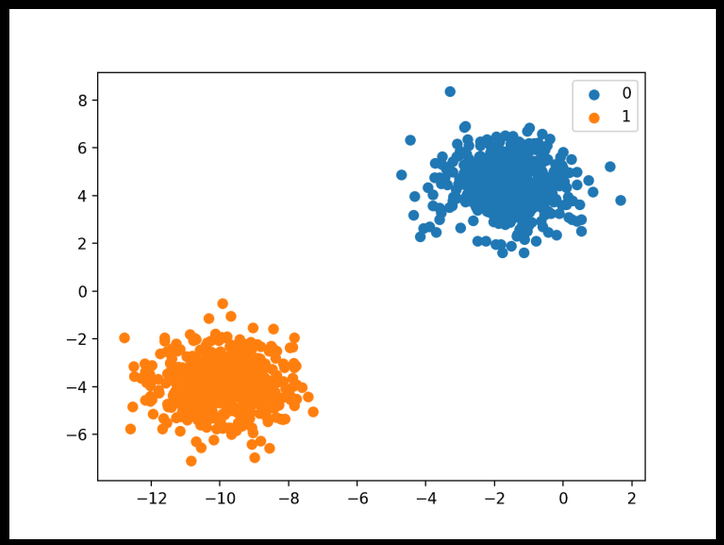


Classification problem: uses algorithms to classify data into different classes. With classification models, the user can make predictions of classification based on different known classes. An everyday example is an algorithm that helps reject spam for a primary e-mail inbox.

Classification problem

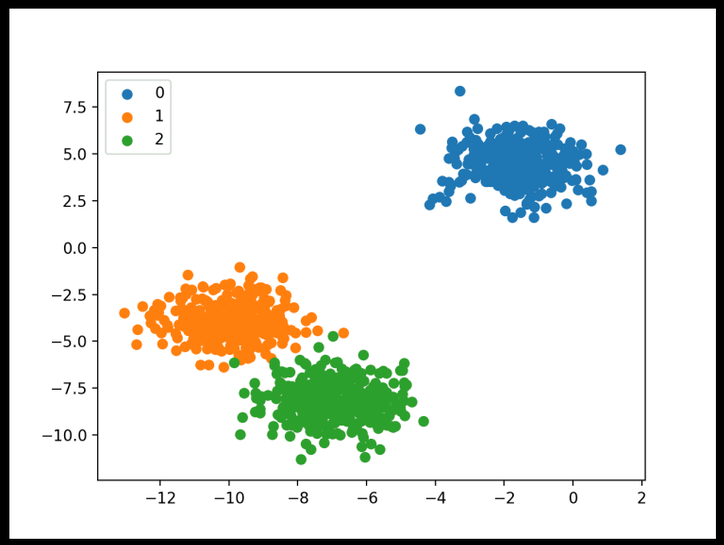
There are three main types of classification tasks that you may encounter:

Binary Classification refers to predicting one of two classes (Email spam detection: spam or not spam).



Multi-Class Classification involves predicting one of more than two classes (Email filter: spam, junk, inbox…).

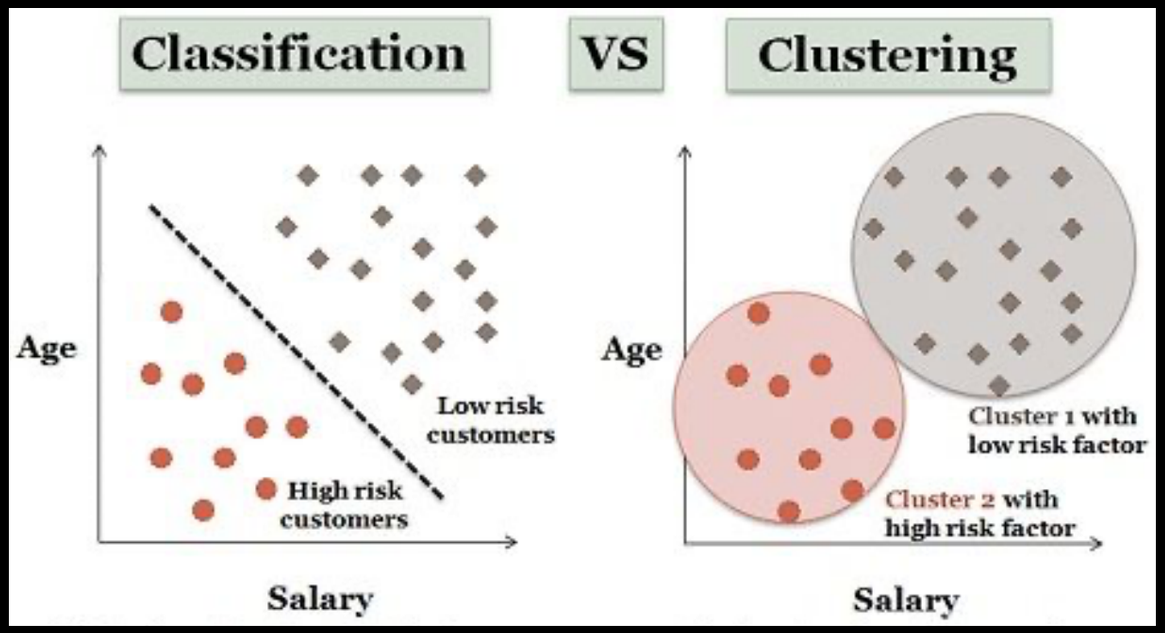
Multi-Label Classification involves more detailed classification, predicting one or more classes for each example (the email could contain text that may be considered vulgar, offensive, etc..).

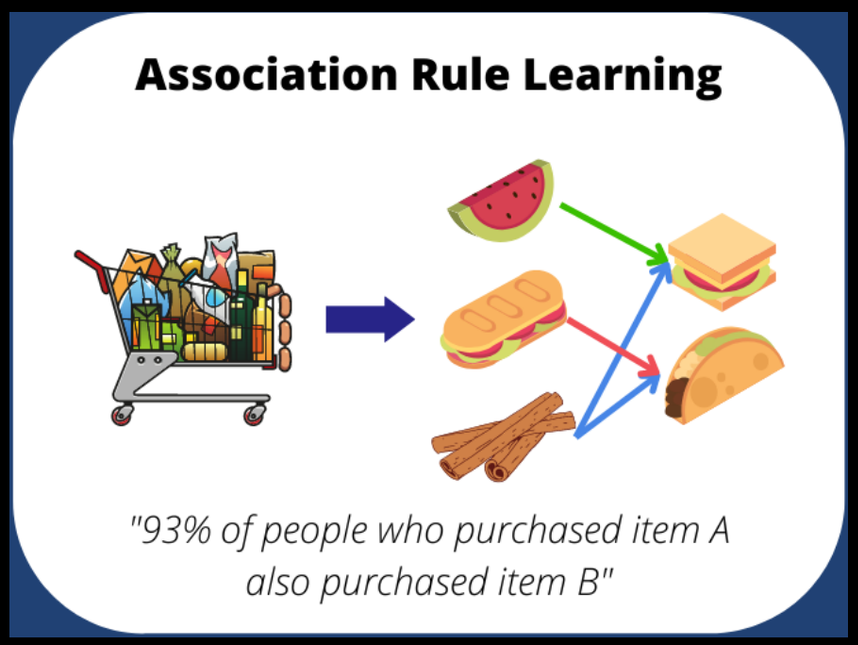


Unsupervised learning

Unsupervised Learning: allows to discover patterns and insights without any explicit guidance or instruction, i.e. reveal unknown patterns in data.

Clustering: Based on similarities or differences, unlabelled data is grouped using clustering techniques, i.e. the task of grouping a set of objects in such a way that objects in the same group (or **cluster**) are more similar to each other than to those in other groups (clusters).





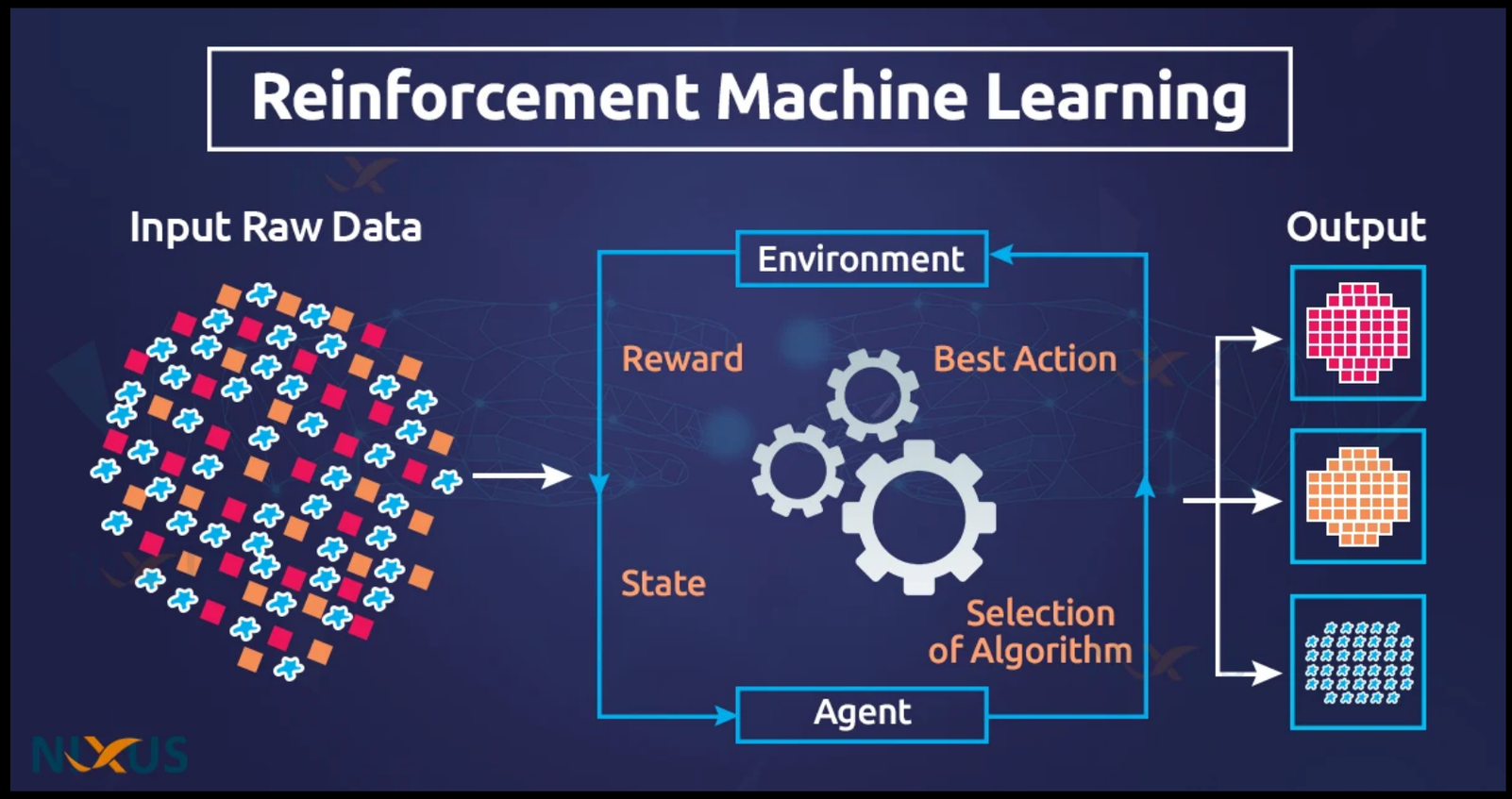
Association: used to identify relationships of variables within a dataset. This is the method used to create the prompt — "other customers also looked at".

If 15 customers purchased a new phone, and they also purchased the headphones to go with it. Therefore, the algorithms recommend headphones to all customers who put a phone in their shopping cart.

Reinforcement learning

Reinforcement Learning: improves continually by getting data from previous iterations.

Many reinforcement learning algorithms for this context use **dynamic programming techniques, i.e.** simplifying a complicated problem by breaking it down into simpler sub-problems in a recursive manner.



E.g.,advertising: the ads you receive are usually from companies whose websites you've visited before.

Machine Learning in Python

scikit-learn

Machine Learning in Python

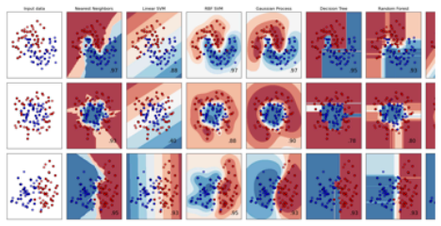
Getting Started | Release Highlights for 1.2 | GitHub

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying which category an object belongs to.

Applications: Spam detection, image recognition.
Algorithms: SVM, nearest neighbors, random forest, and more...

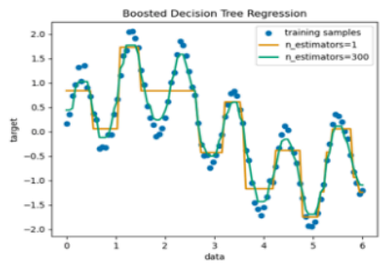


Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, nearest neighbors, random forest, and more...

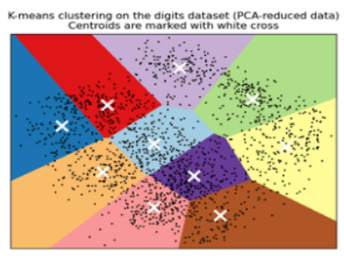


Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, and more...

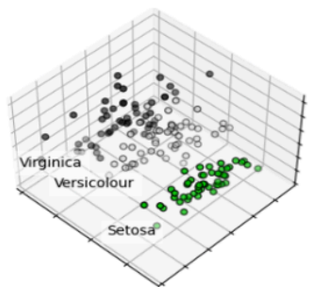


Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization, and more...

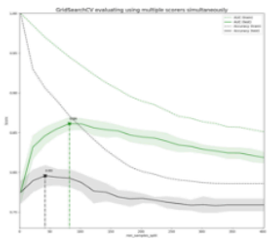


Examples

Model selection

Comparing, validating and choosing parameters and models.

Applications: Improved accuracy via parameter tuning
Algorithms: grid search, cross validation, metrics, and more...

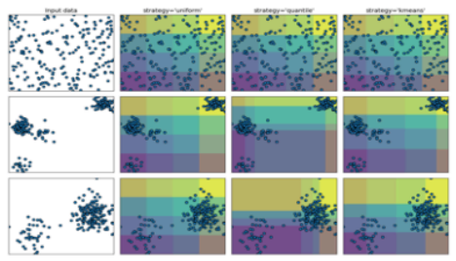


Examples

Preprocessing

Feature extraction and normalization.

Applications: Transforming input data such as text for use with machine learning algorithms.
Algorithms: preprocessing, feature extraction, and more...



Examples

OUTLINE

- **INTRODUCTION TO MACHINE LEARNING**
- **CLASSIFICATION WITH NEURAL NETWORKS**
- **APPLICATION: DARK MATTER SEARCHES**

  V.G., B. Záldivar, M. A. Sánchez-Conde, J. Coronado-Blázquez, MNRAS 2023

Linear Regression - 1 feature

1-Feature (1F) (x), n data

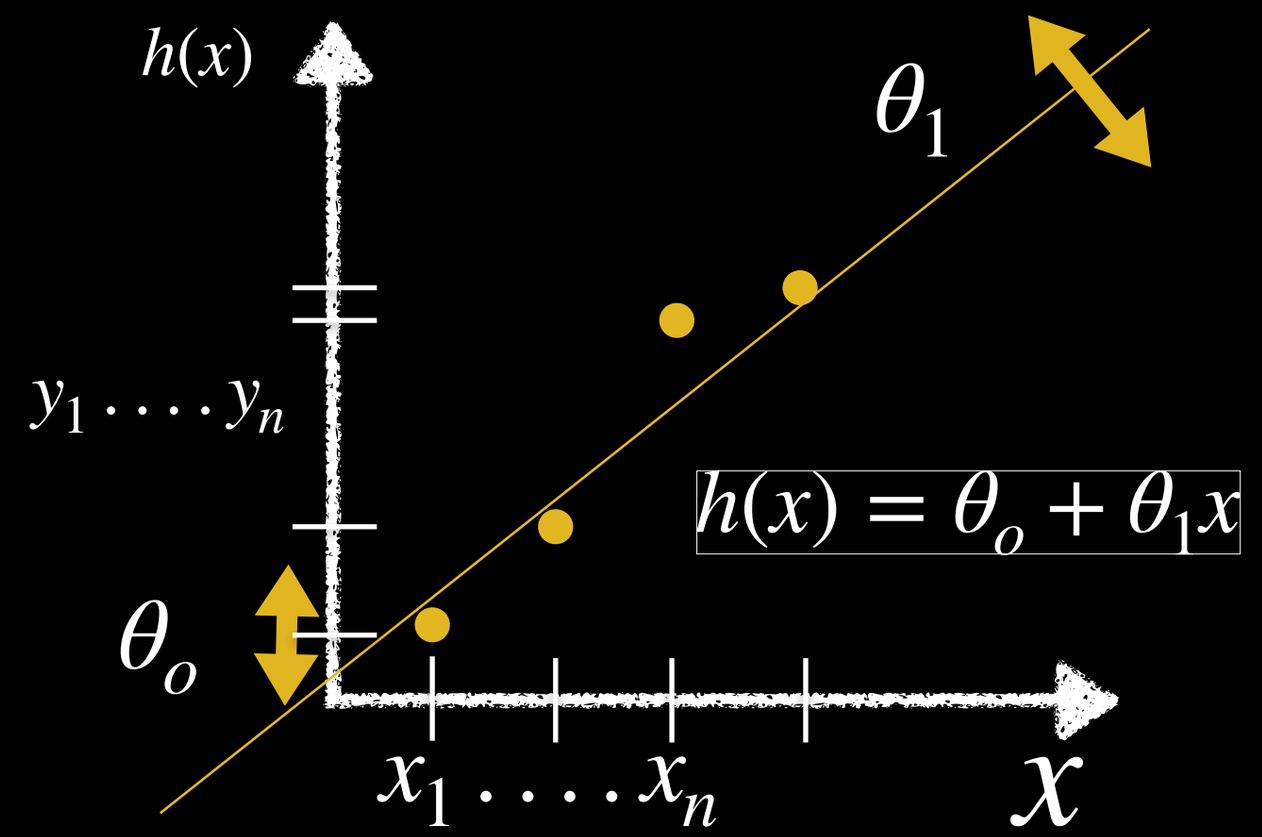
- Sample of n data set (x, y)
- Hypothesis : linear fit $h(x) = \theta_o + \theta_1 x$
  with parameters (θ_0, θ_1)
- In the matrix notation: $h(X) = \Theta^T X$

$$\mathbf{X}^T = \{x_1 \ldots . x_n\}$$

$$\mathbf{Y}^T = \{y_1 \ldots . y_n\}$$

$$\mathbf{\Theta}^i = \{\theta_o^i, \theta_1^i\}_{i=1\ldots n}$$

$$\mathbf{\Theta}^T = \{\mathbf{\Theta^1} \ldots \mathbf{\Theta^n}\}$$



Linear Regression - 1 feature

1-Feature (1F) (x), n data

Hypothesis: $h(x) = \theta_o + \theta_1 x \equiv \Theta^T X$

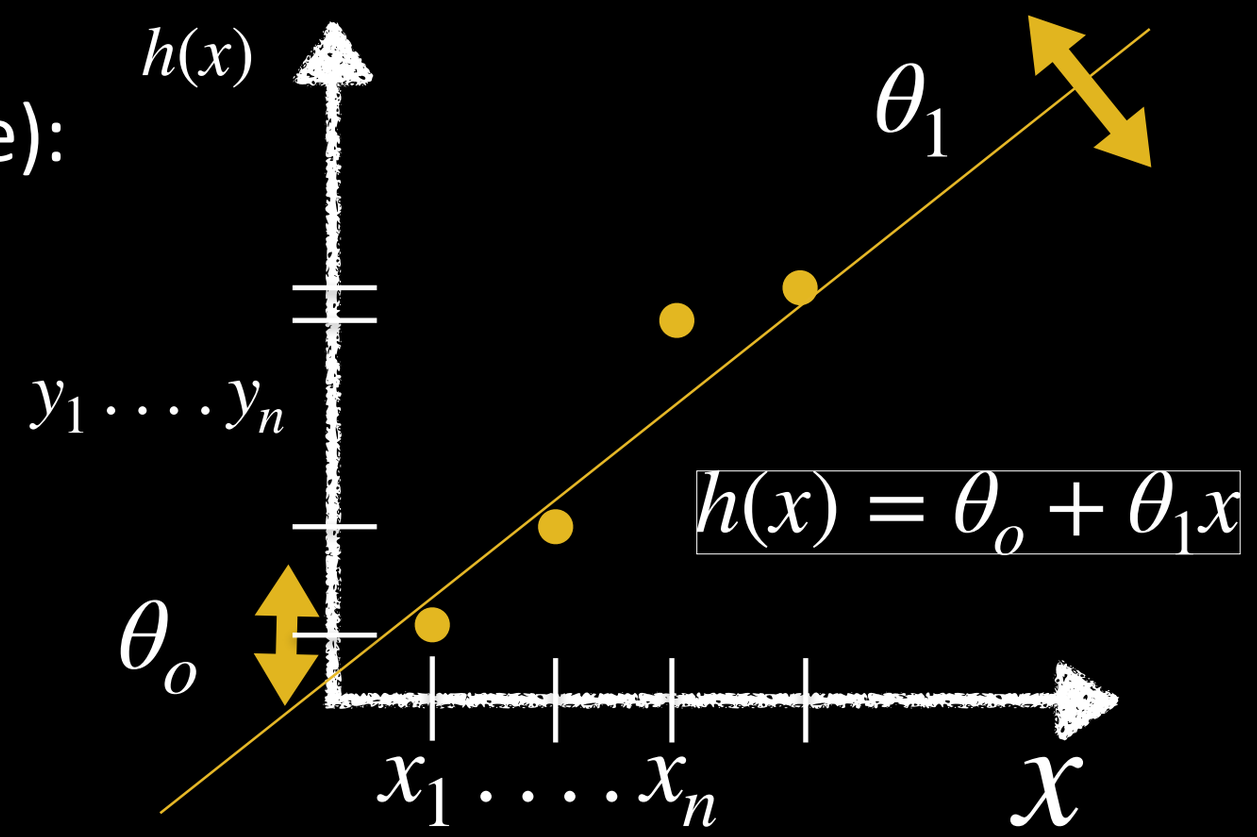
Parameters: θ_o, θ_1

Cost/Lost function (e.g. least square):

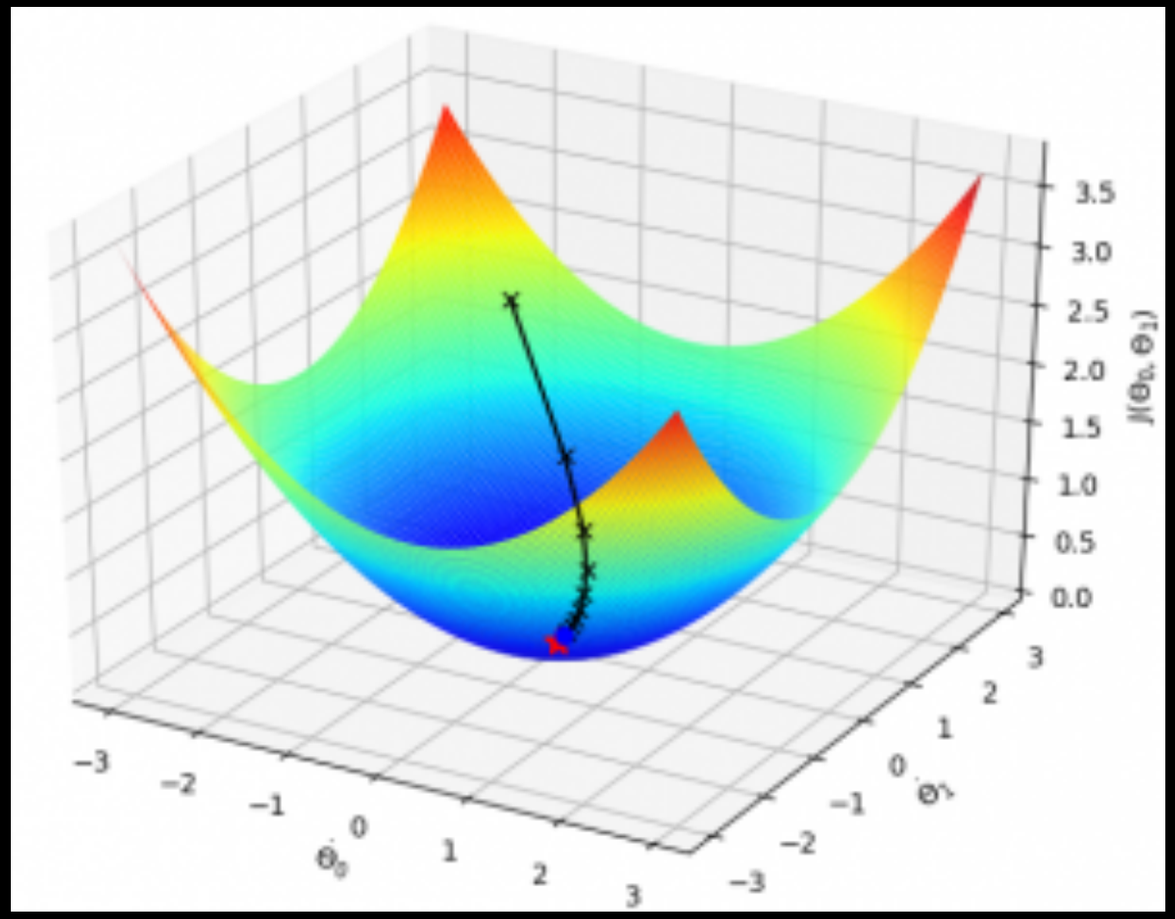
$$J(\theta) = \frac{1}{2n}\sum_{i=1}^{n}(h(x_i) - y_i)^2 \equiv \frac{1}{2n}\sum_{i=1}^{n}((\Theta^T \mathbf{X})_i - Y_i^2)$$

Minimize:

$$J(\Theta) = J(\theta_0, \theta_1)$$



Linear Regression - 1 feature



Hypothesis: $h(x) = \theta_o + \theta_1 x$

Parameters: θ_o, θ_1

Cost function (e.g. least square):

$$J(\theta) = \frac{1}{2n}\sum_{i=1}^{n}(h(x_i) - y_i)^2 \equiv \frac{1}{2n}\sum_{i=1}^{n}((\mathbf{\Theta}^T\mathbf{X})_i - Y_i^2)$$

Minimize: $J(\Theta) = J(\theta_0, \theta_1)$

Gradient descent:

Repeat until convergence

$$\theta_j =: \theta_j - \alpha\frac{\partial J(\theta_0, \theta_1)}{\partial \theta_j} \ (simultaneusly\ update\ j = 0\ , j = 1)$$

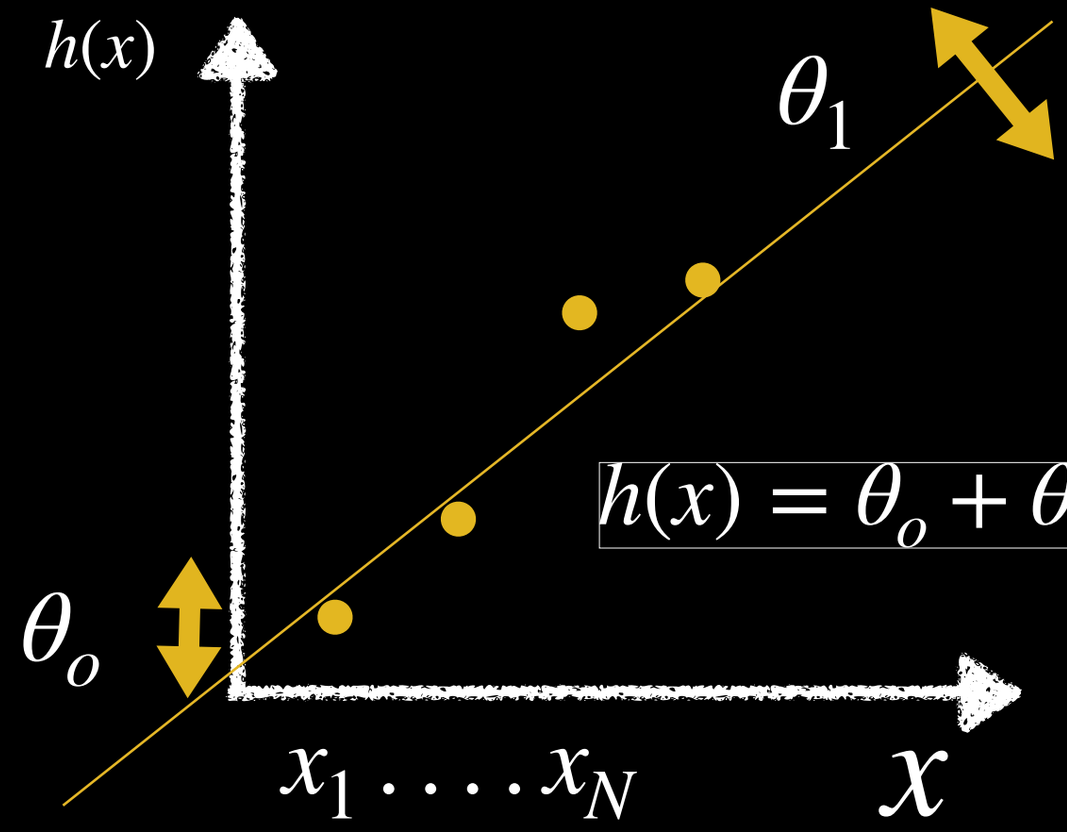
Learning rate

Linear Regression - multiple features

1-Feature (1-F) (x), n measurements

$$\mathbf{X}^T = \{x_1 \ldots . x_n\}$$

$$\theta^{\mathbf{i}} = \{\theta_o^i, \theta_1^i\}_{i=1\ldots n}$$



$$h(x) = \theta_o + \theta_1 x$$

f-Feature (f-F) (x), n measurements

$$[\mathbf{X}] = [n \times f]$$

$$X_i = \{x_1 \ldots . x_f\}_{i=1\ldots n}$$

$$X_j^T = \{X_1^T \ldots X_n^T\}_{j=1\ldots f}$$

$$\theta^i = \{\theta_o, \theta_1 \ldots . \theta_f\}_{i=1\ldots n}$$

$$h(\mathbf{X}) = \theta_o^i + \theta_1^i \mathbf{X}_i + \ldots . \theta_f^i \mathbf{X}_i^f = \theta^T \mathbf{X}$$

LR **cost function** e.g.,

$$J(\Theta) = \frac{1}{2}(h(x) - Y)^2 \equiv \frac{1}{2n}\sum_{i=1}^{n}((\Theta^T \mathbf{X})_i - Y_i^2)$$

Gradient descent - multiple feature

1 feature - n data

$$\theta_0 =: \theta_0 - \alpha \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_0} = \theta_0 - \alpha \frac{1}{n} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)}) \quad (x_{j=0} = 1)$$

$$\theta_1 =: \theta_1 - \alpha \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_1} = \theta_0 - \alpha \frac{1}{n} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)}) x^{(i)}$$

f features - n data

$$\theta_j =: \theta_j - \alpha \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_j} = \theta_j - \alpha \frac{1}{n} \sum_{i=1}^{n} \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_j} x_j^{(i)} \quad (j = 1 \ldots f)$$

Linear vs Logistic Regression

LINEAR REGRESSION

$$J(W) = \frac{1}{2n}\sum_{i=1}^{n}((\Theta^T \mathbf{X})_i - Y_i^2)$$

LOGISTIC REGRESSION

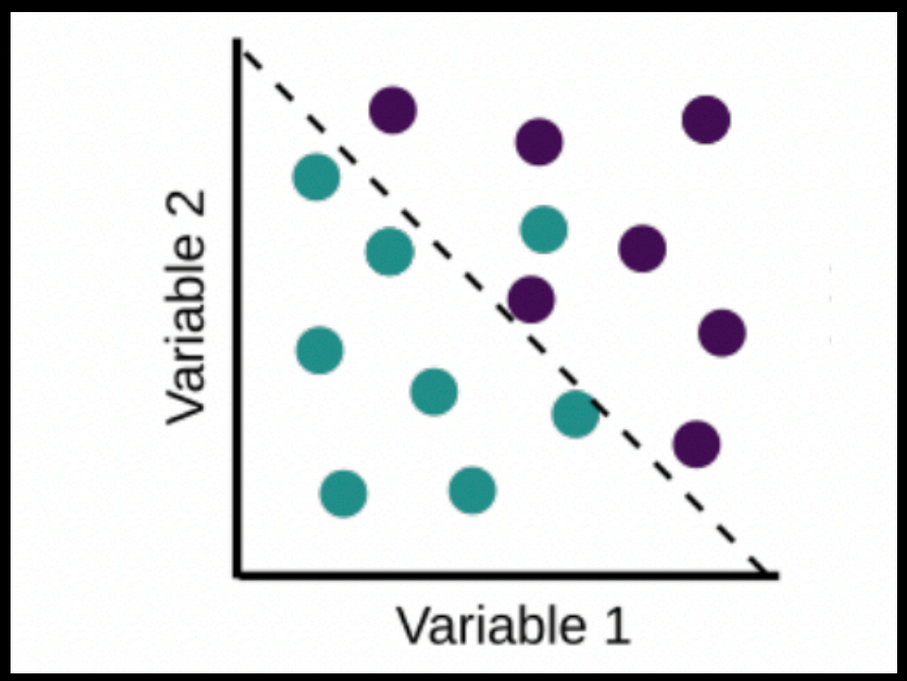
$$J(\Theta) = -\frac{1}{n}[\sum_{i=1}^{n} y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)})\log(1 - h_\theta(x^i))]$$

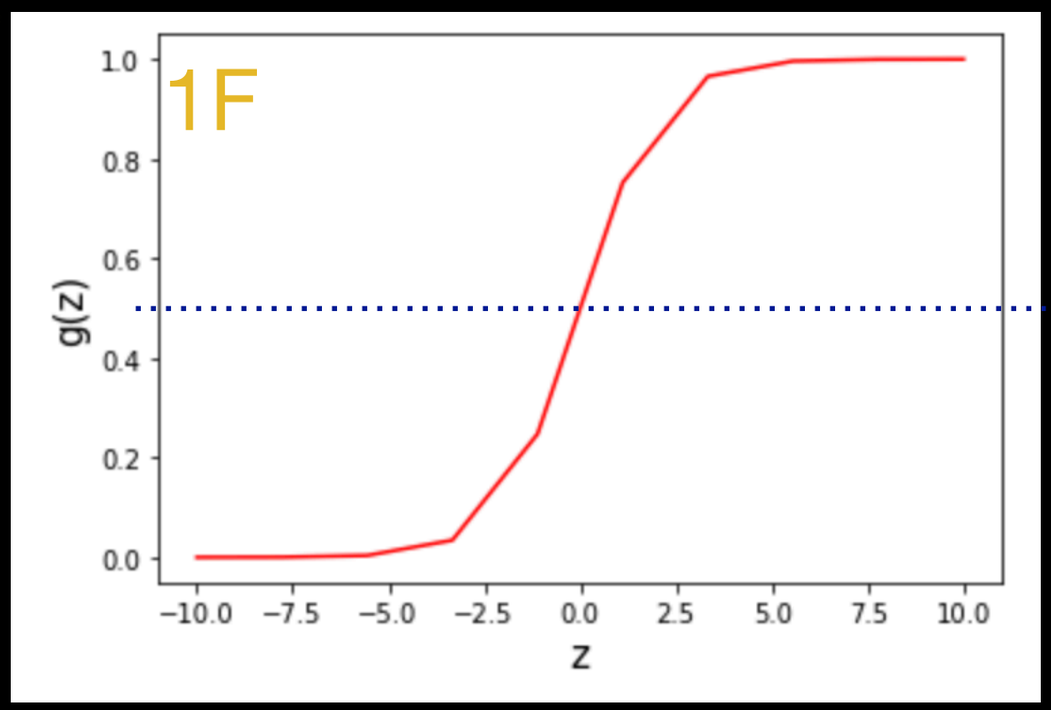
$$h(x) \rightarrow g(z) = \frac{1}{1 + e^{-z}} \quad z(x) = \theta_o + \theta_1 x$$

Activation function

Decision boundary

$$g(z) > 0.5 \rightarrow y^{(i)} = 1$$





$$g(z) \leq 0.5 \rightarrow y^{(i)} = 0$$

Gradient descent - logistic regression

For logistic regression, the algorithm is identical to linear regression

$$\theta_j =: \theta_j - \alpha \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_j} = \theta_j - \alpha \frac{1}{n} \sum_{i=1}^{n} \frac{\partial J(\theta_0, \theta_1)}{\partial \theta_j} x_j^{(i)} \quad (j = 1...f)$$

But the definition of the cost function changes:

$$J(\Theta) = -\frac{1}{n}[\sum_{i=1}^{n} y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^i))]$$

Neural Network

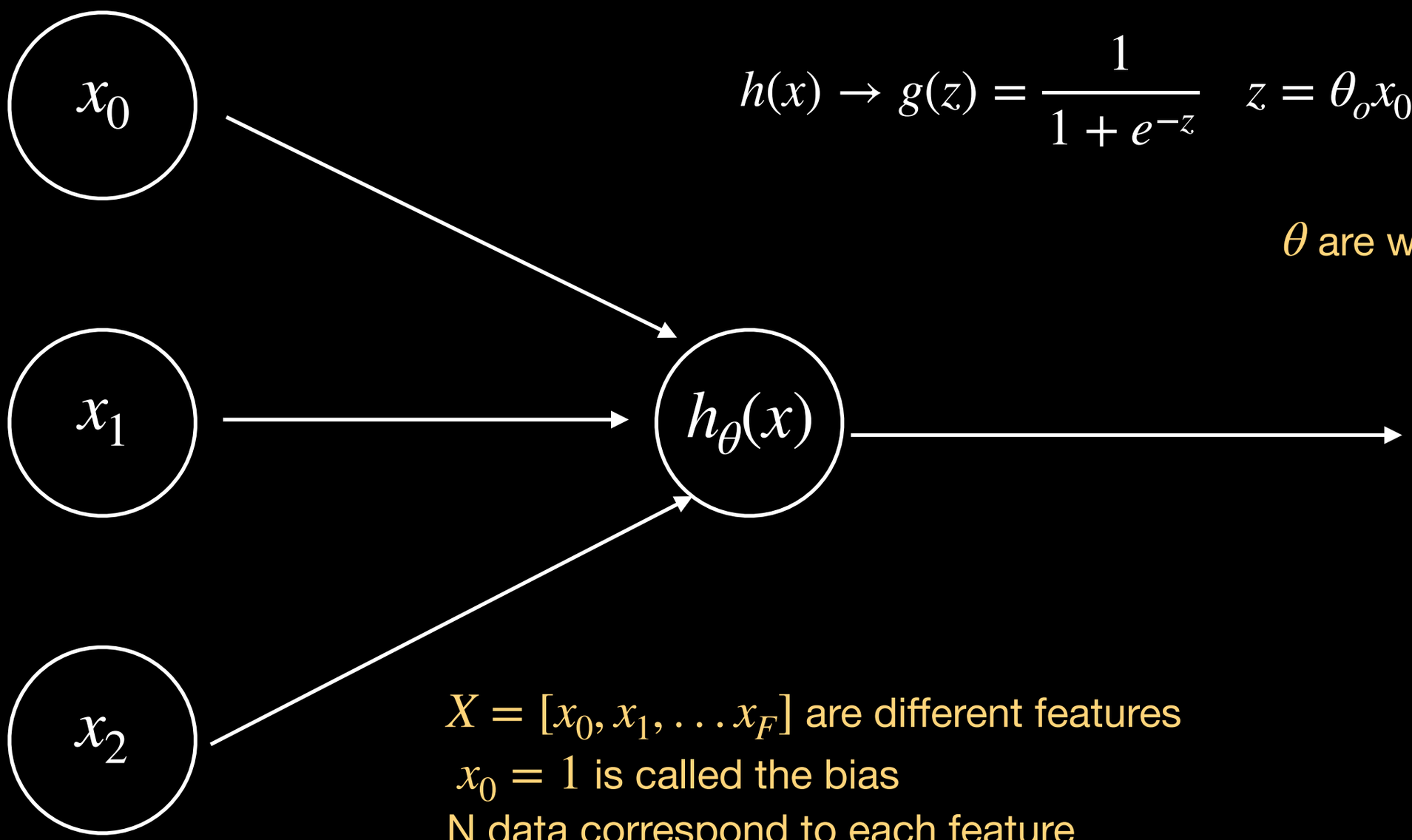
Logistic unit - neuron

Sigmoid (logistic) activation function

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$h(x) \to g(z) = \frac{1}{1 + e^{-z}} \quad z = \theta_o x_0 + \theta_1 x_1 + \theta_2 x_2^2 \ldots$$

θ are weights of the fit



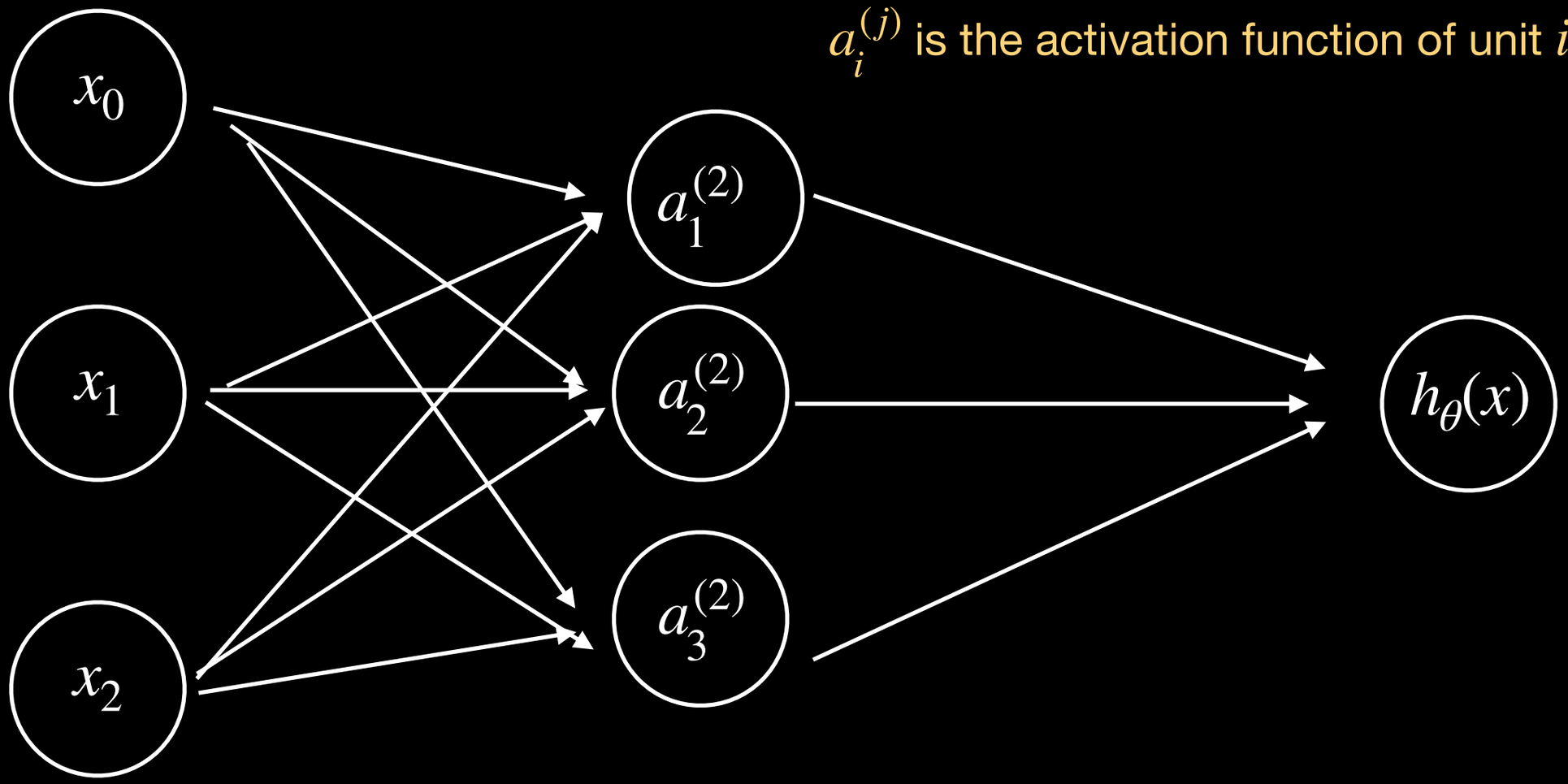
$X = [x_0, x_1, \ldots x_F]$ are different features

$x_0 = 1$ is called the bias

N data correspond to each feature

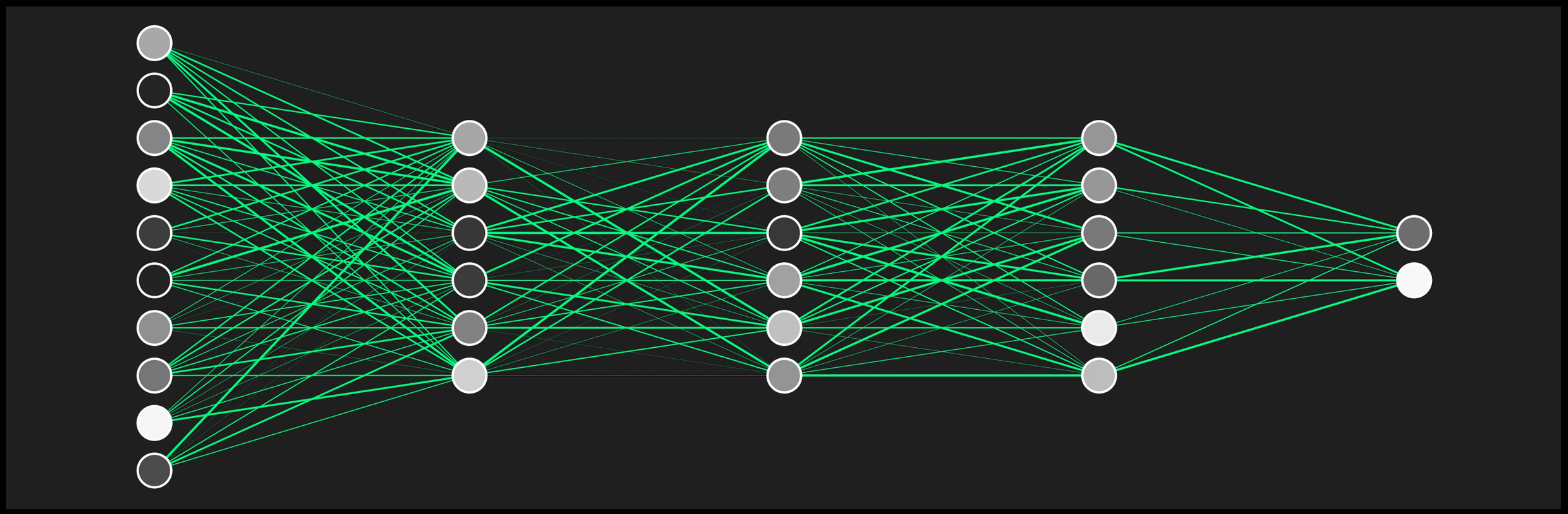
Neural Network

Each logistic unit is part of a more complex neural network



$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$
$$a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$
$$a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$
$$h_\Theta(x) = a_1^{(3)} = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

Neural Network



Layer 1 Layer 2 Layer 3 Layer 4 Layer 5

Input Hidden Layers Output

Neural Network

LOGISTIC REGRESSION

$$J(\theta) = -\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)})\log(1 - h_\theta(x^i))] + \frac{\lambda}{2m}\sum_{j=1}^{n} \theta_j^2$$

NEURAL NETWORK

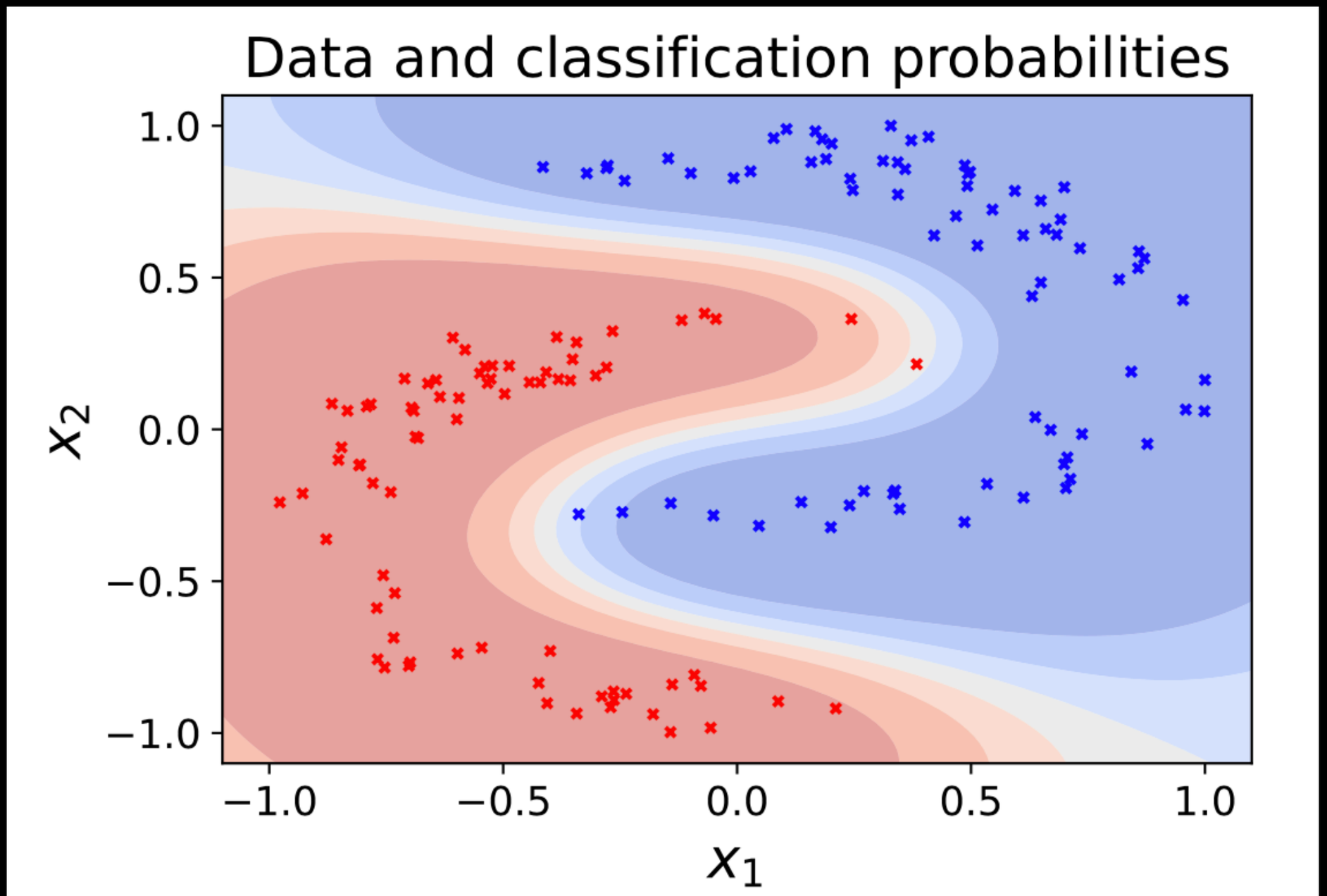
$h_\Theta(x) \in \mathbb{R}^k \; (h_\Theta(x))_i = i^{th}$ output

$$J(\Theta) = -\frac{1}{m}[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log(h_\Theta(x^{(i)}))_k + (1 - y_k^{(i)})\log(1 - h_\Theta(x^i))_k] + \frac{\lambda}{2m}\sum_{l=1}^{L-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}} (\Theta_{ji}^{(l)})^2$$

Implemented in software like scikit-learn

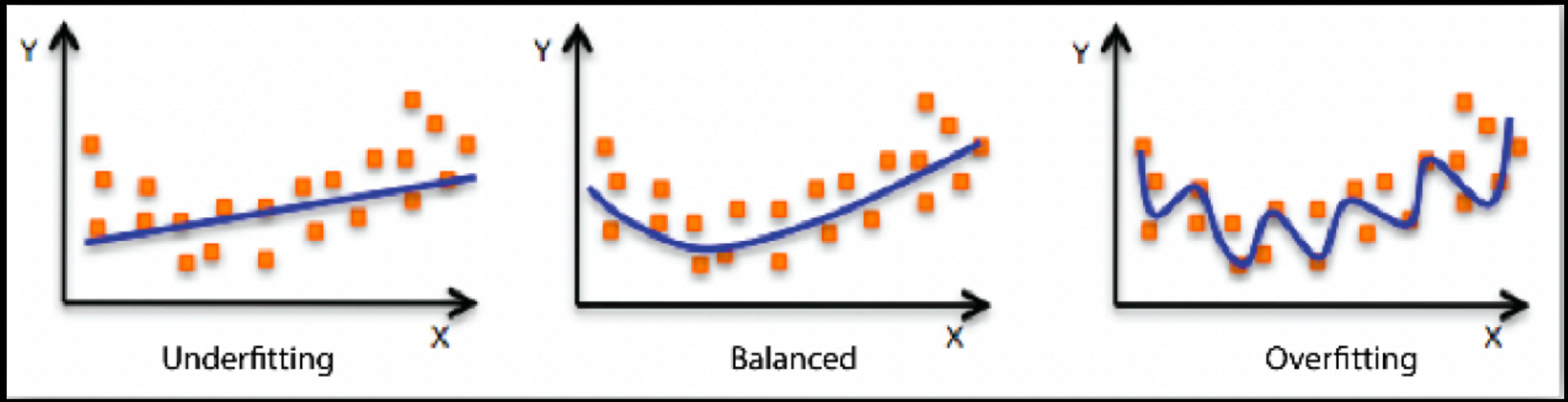
Neural Network

Non-linear hypothesis

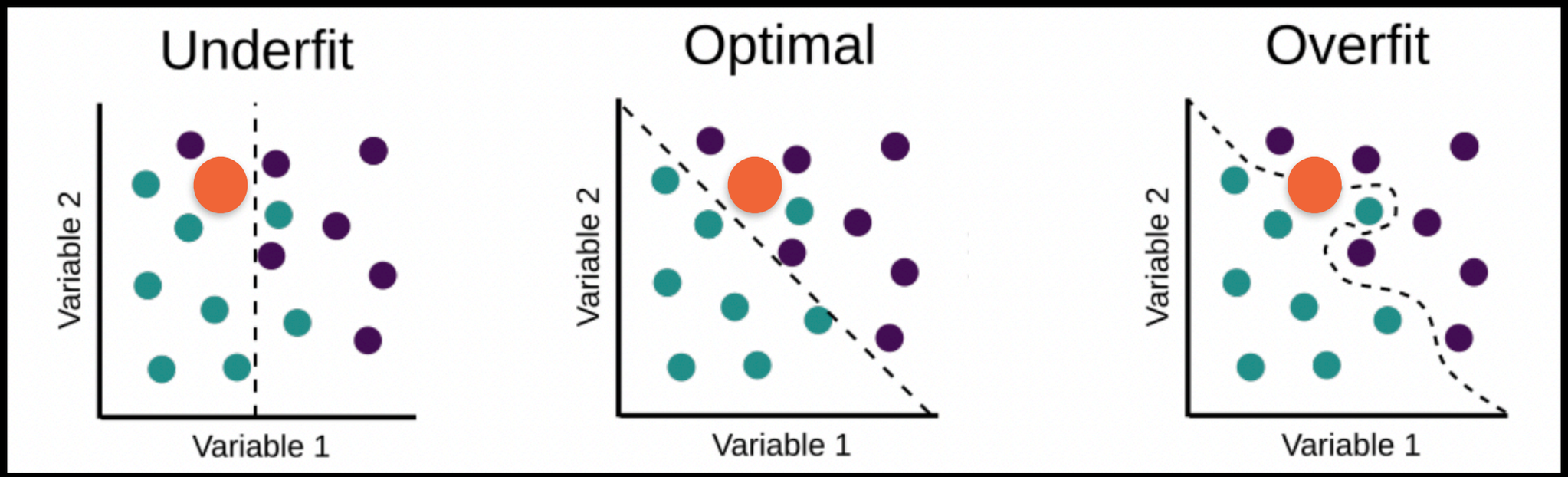


Overfitting

LINEAR REGRESSION



LOGISTIC REGRESSION

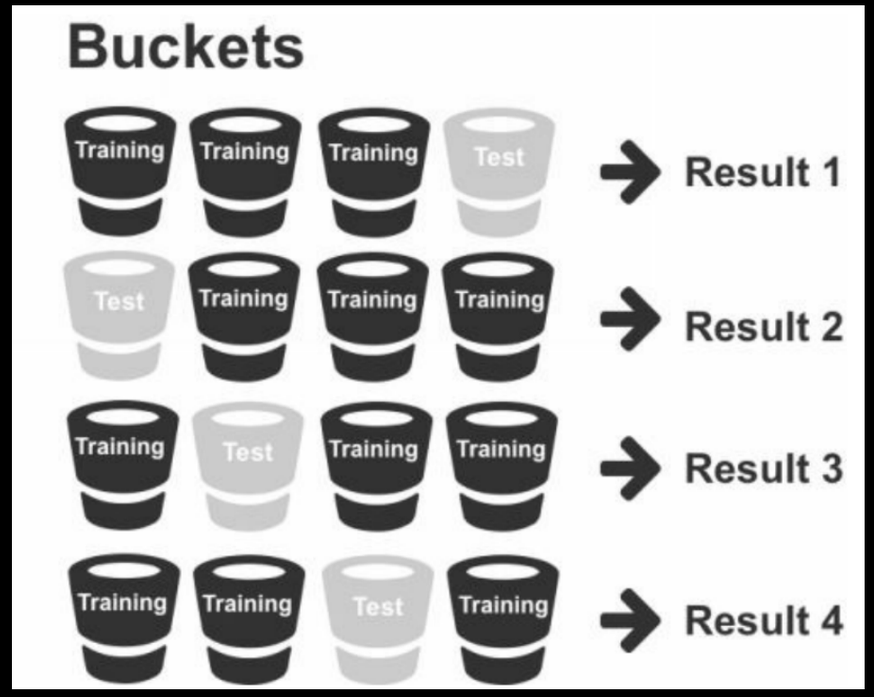


Classification accuracy: cross validation

Training/testing procedure:

4-folds

- 75% Training data -> To minimize cost function
- 25% Test Data ->To test your model



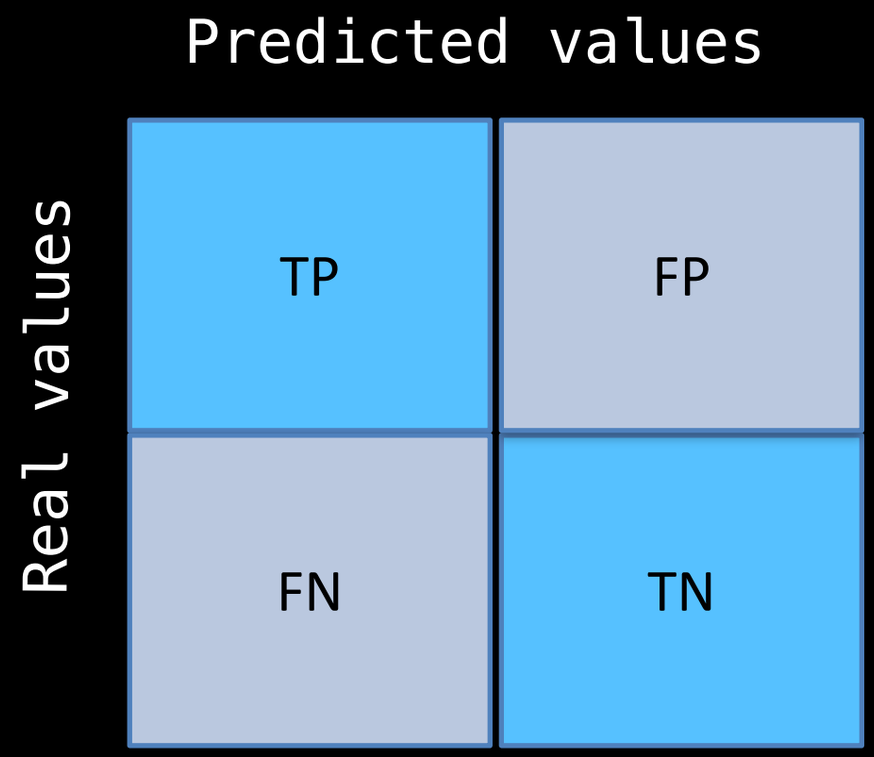
OVERALL ACCURACY:
PERCENTAGE OF WELL CLASSIFIED DATA SET

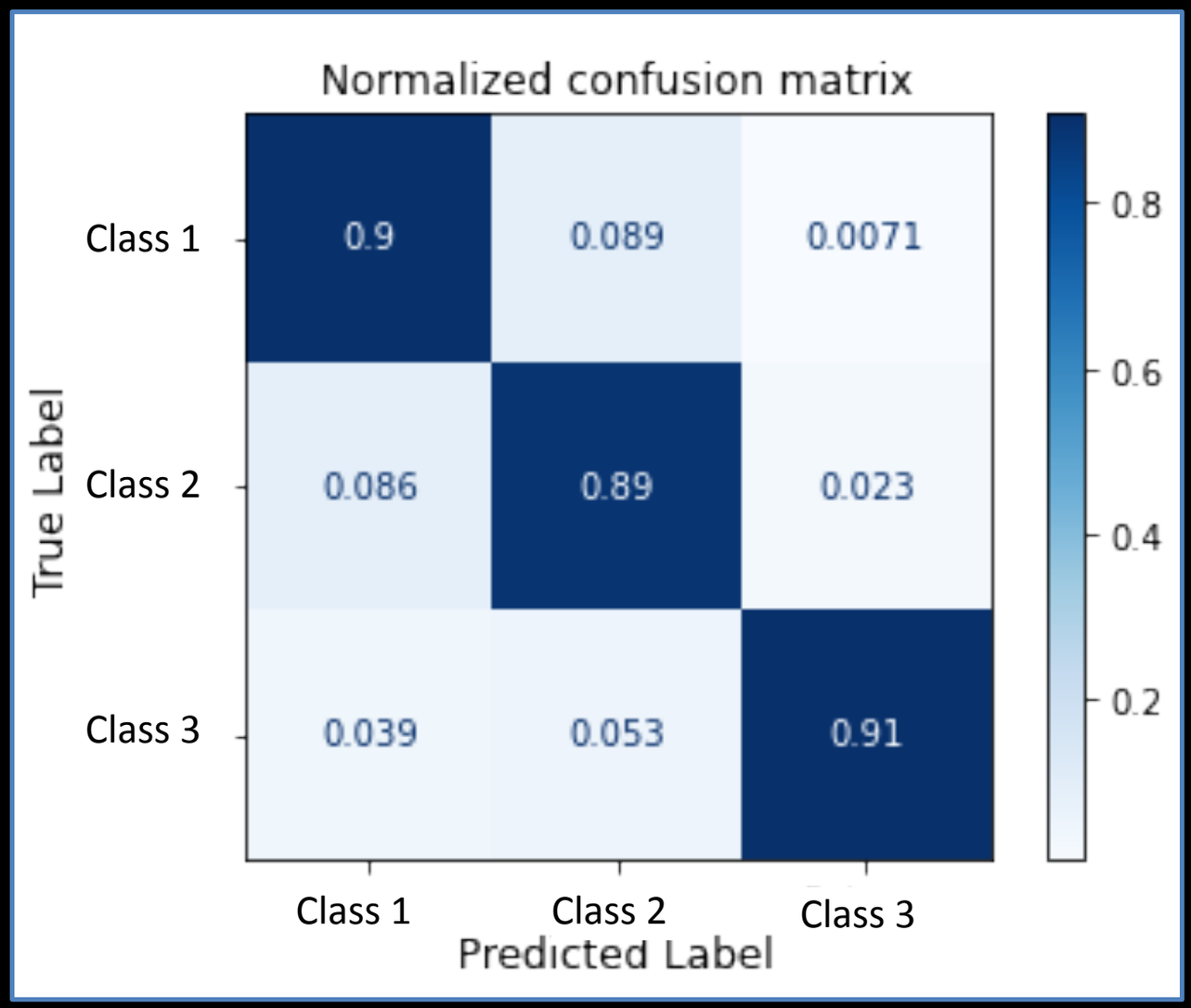
$$\textbf{Overall accuracy (OA)}(y, \hat{y}) = \frac{1}{n\textbf{samples}} \sum_{i=0}^{n\textbf{samples}-1} 1(\hat{y}_i = y_i)$$

Classification accuracy: confusion matrix

True Negative (TN): percentage of well classified Astro sources (normalised to the total number of astro sources)

True positive (TP): percentage of well classified Dark Matter sources (normalised to the total number of DM sources)



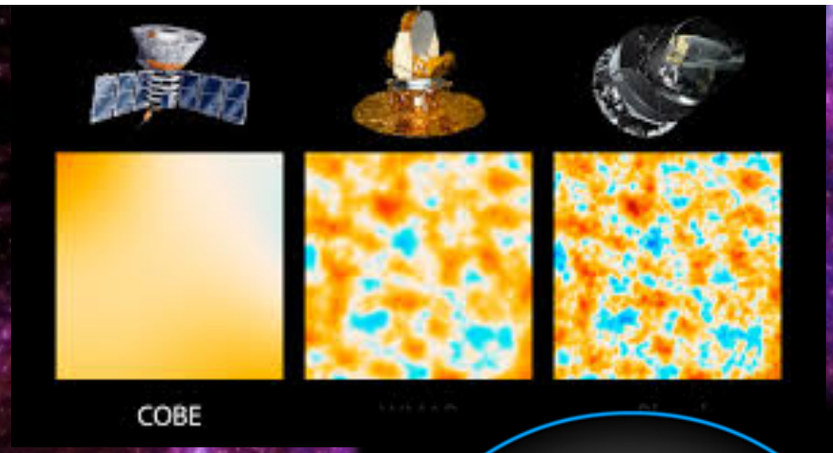


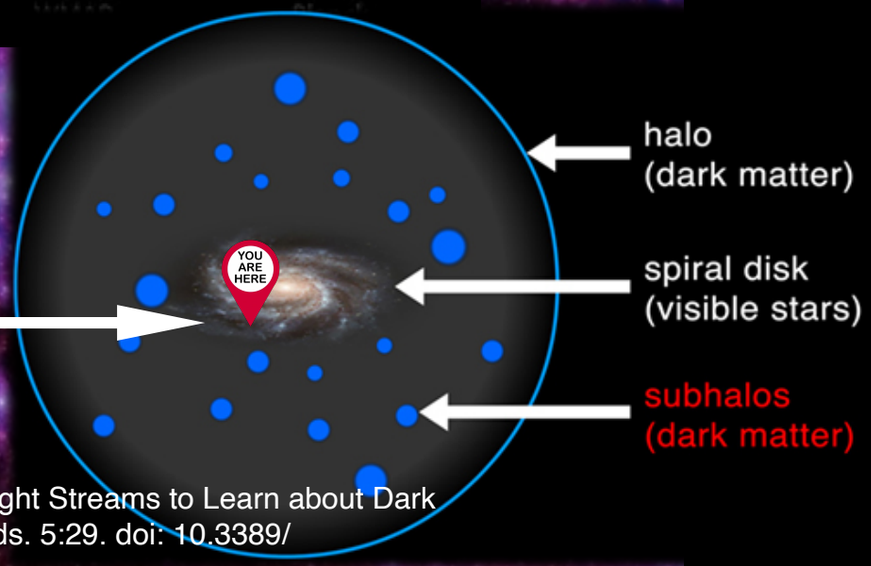
OUTLINE

- **INTRODUCTION TO MACHINE LEARNING**
- **CLASSIFICATION WITH NEURAL NETWORKS**
- **APPLICATION: DARK MATTER SEARCHES**

  V.G., B. Záldivar, M. A. Sánchez-Conde, J. Coronado-Blázquez, MNRAS 2023

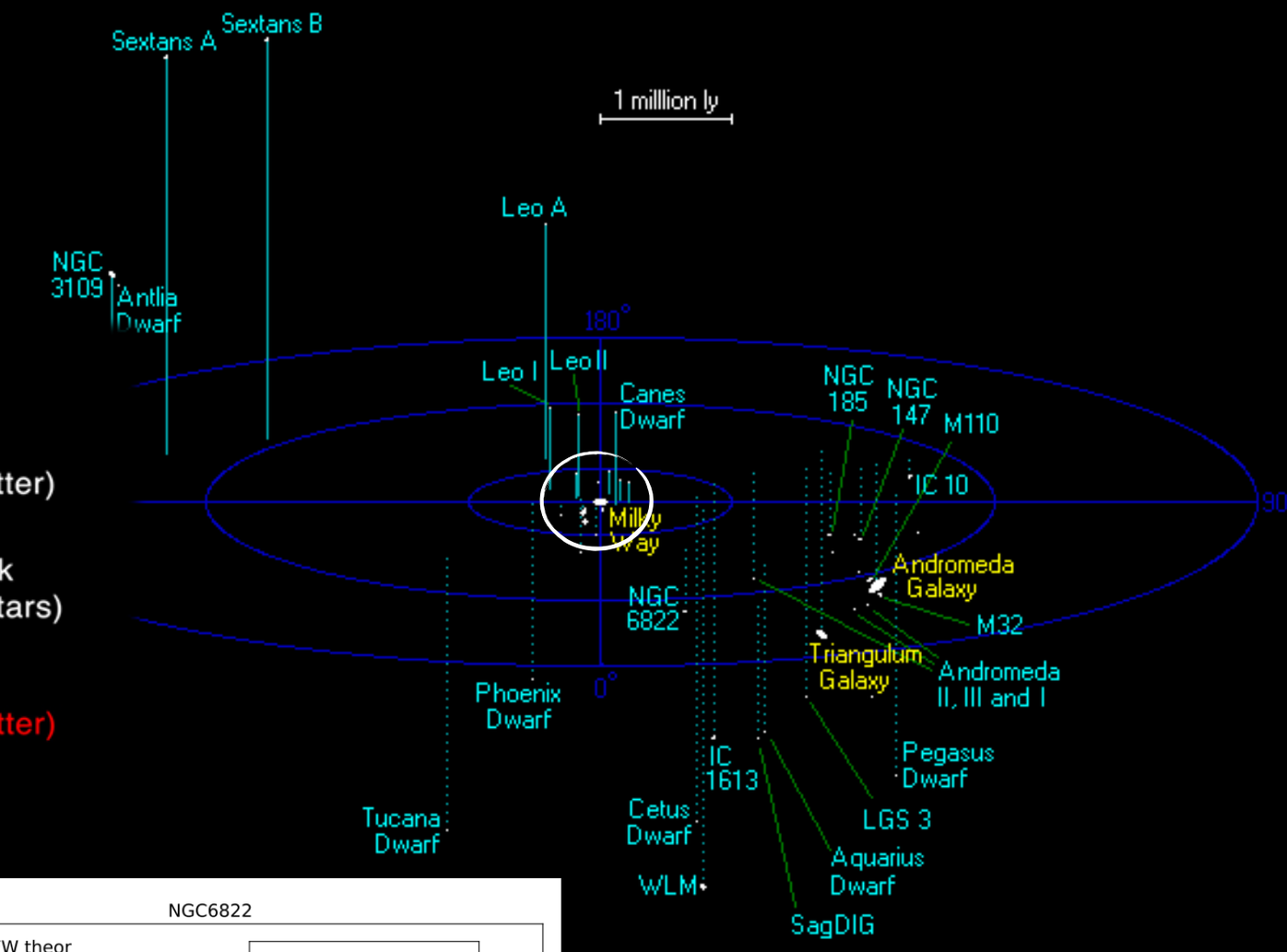
Dark Matter

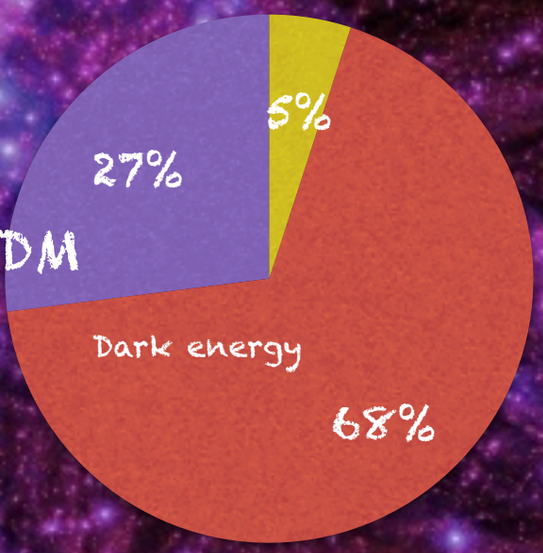


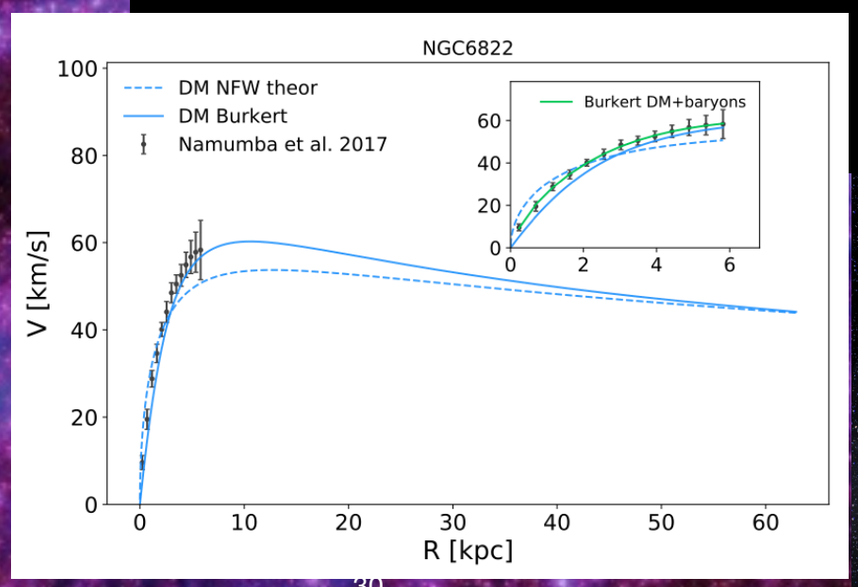


Sun/
Earth

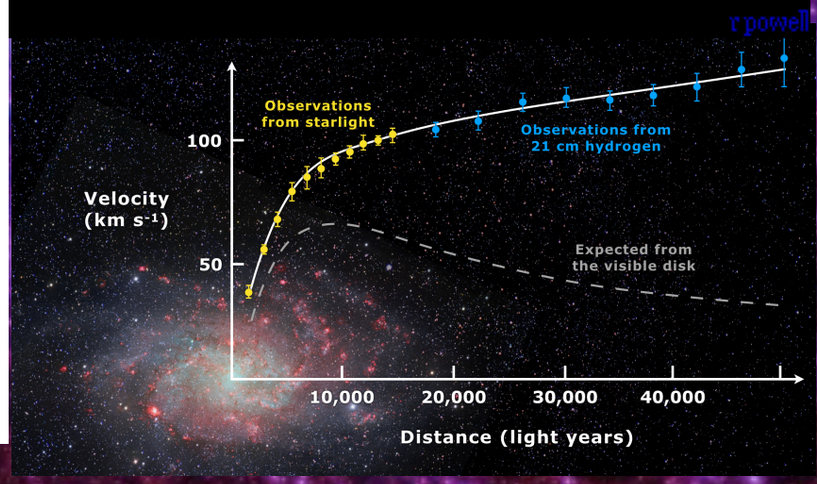
Ngan W (2017) Using Bright Streams to Learn about Dark Matter. Front. Young Minds. 5:29. doi: 10.3389/frym.2017.00029





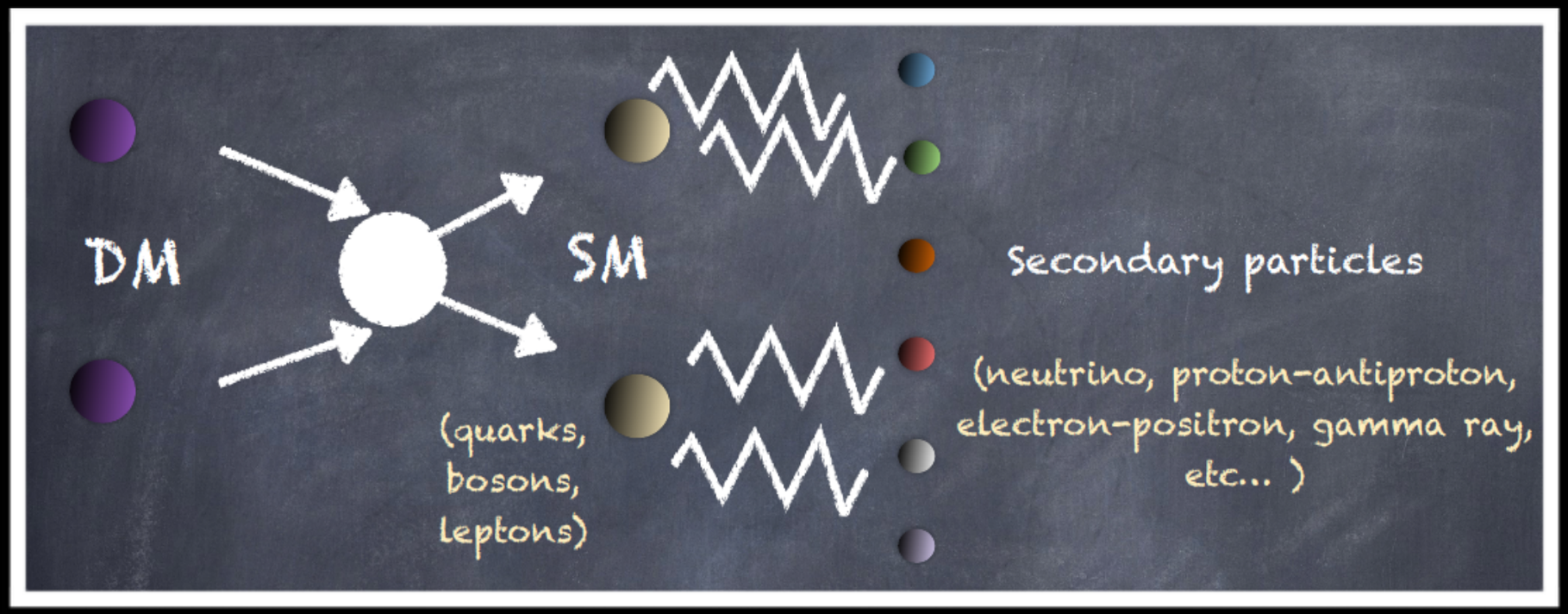


V. Gammaldi et al. Phys. Rev. D 104, 083026 (2021)



Dark Matter: indirect search

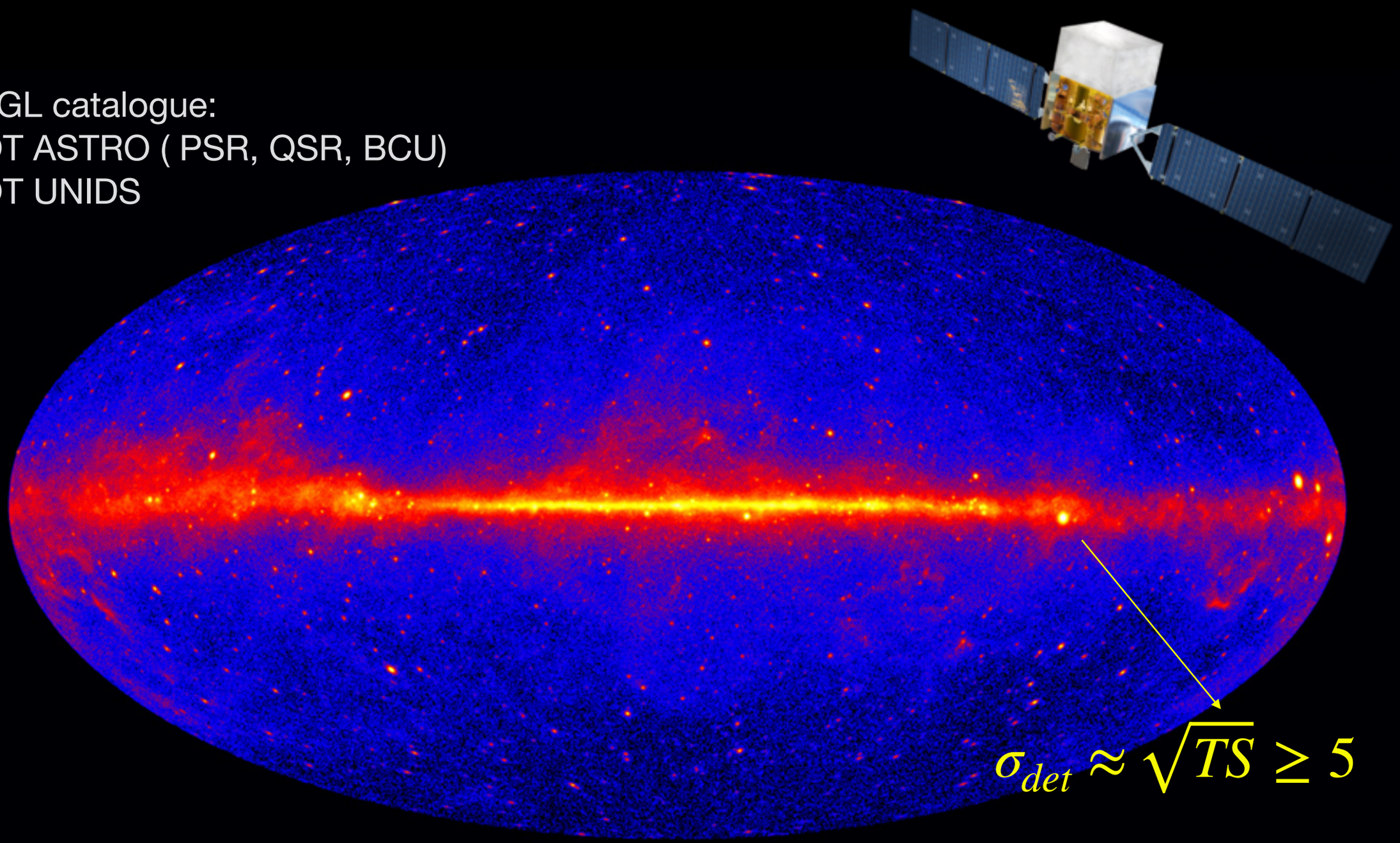
Context: Indirect searches for Dark Matter (DM)
Weakly Interacting Massive Particles (WIMPs)



$$\frac{d\phi_\gamma}{dE}(E, \Delta\Omega, l.o.s) = \underbrace{\frac{1}{4\pi}\frac{\langle\sigma v\rangle}{\delta\, m_\chi^2}\frac{dN_\gamma}{dE}(E)}_{\text{Particle Physics}} \times \underbrace{J(\Delta\Omega, l.o.s)}_{\text{Astrophysics}}$$

Fermi-LAT gamma-ray data and β-plot

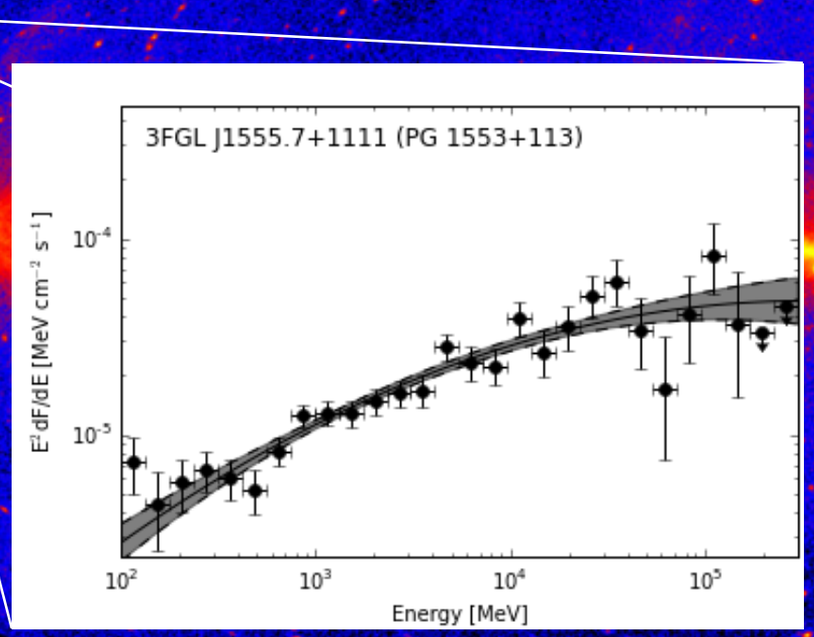
4FGL catalogue:
TOT ASTRO (PSR, QSR, BCU)
TOT UNIDS



$$\sigma_{det} \approx \sqrt{TS} \geq 5$$

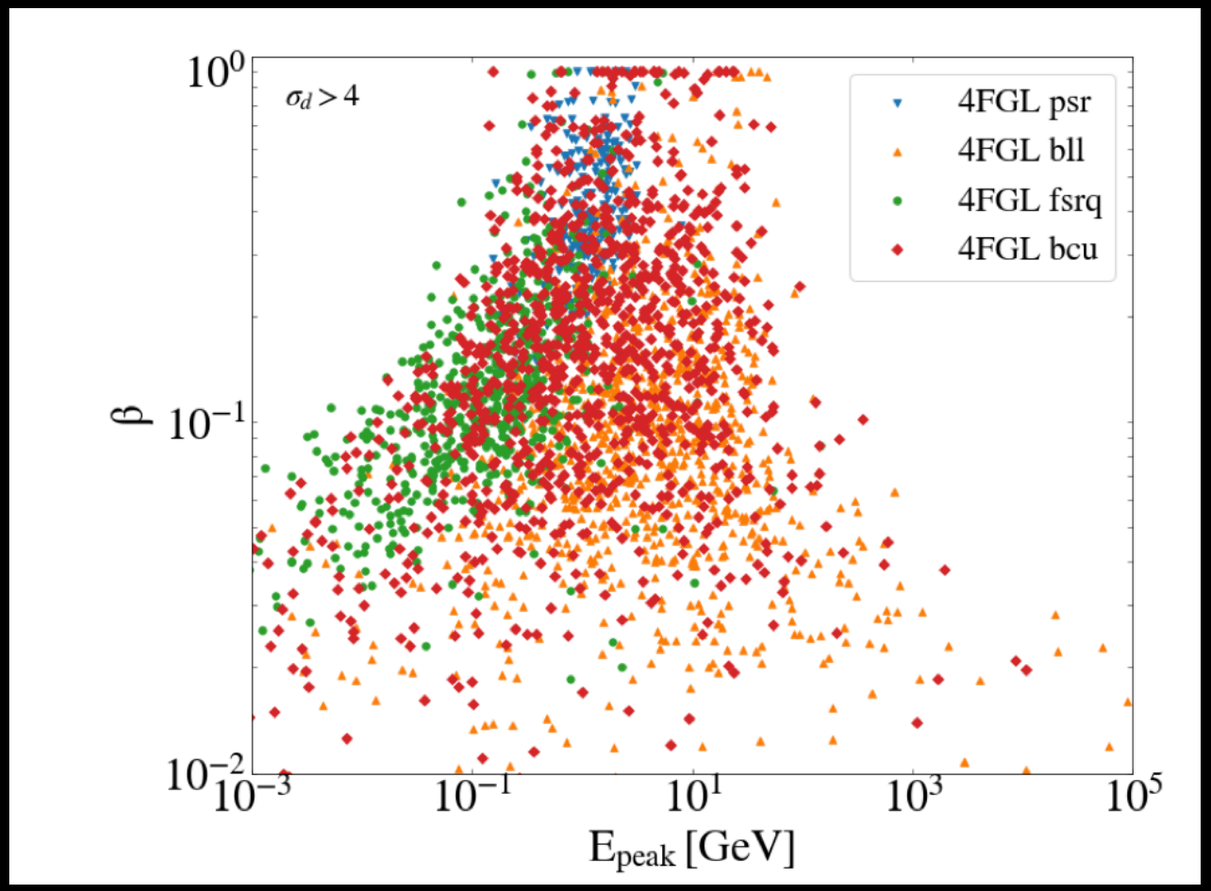
Fermi-LAT gamma-ray data and β-plot

4FGL catalogue:
TOT ASTRO (PSR, QSR, BCU)
TOT UNIDS



Log-Parabola: $\frac{dN}{dE} = N_0 \left(\frac{E}{E_0} \right)^{-\alpha - \beta \cdot \log(E/E_0)}$, $E_{peak} = E_0 \cdot e^{\frac{2-\alpha}{2\beta}}$

Fermi-LAT gamma-ray data and β-plot

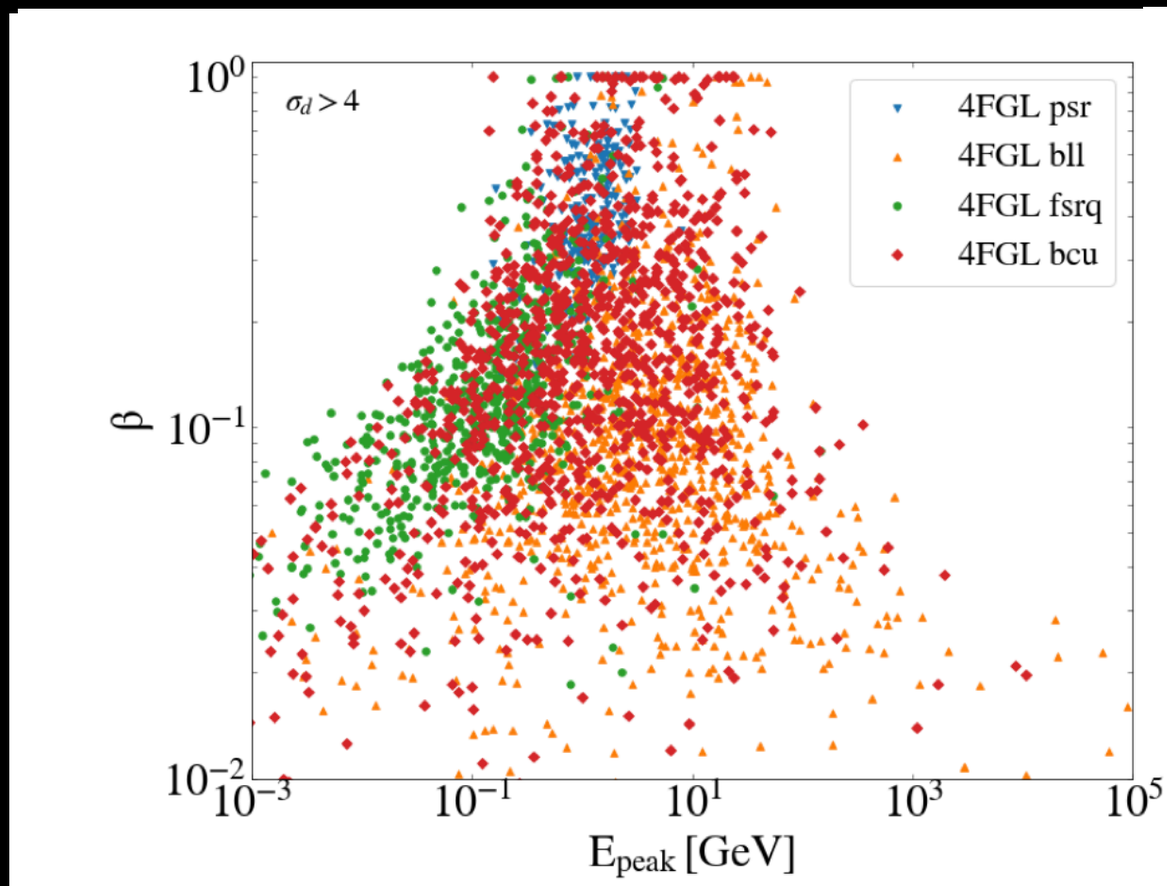


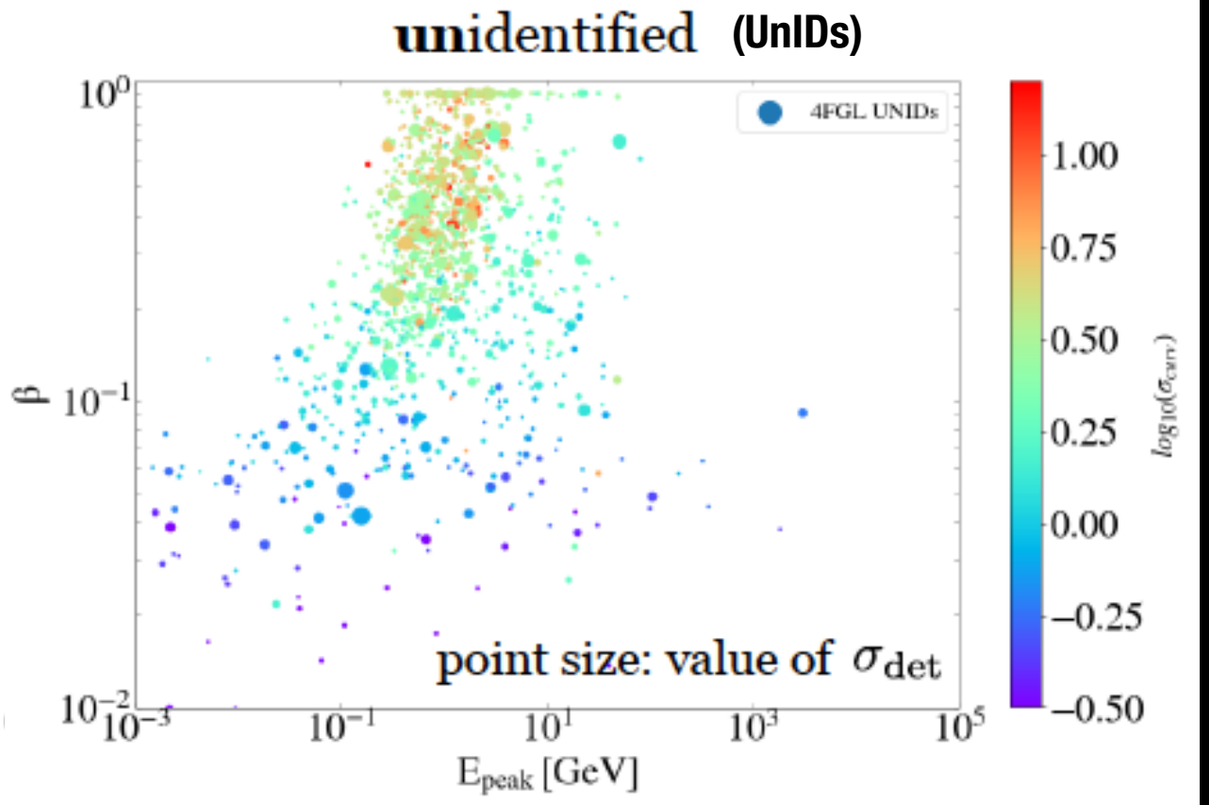
SOURCES

- Blazars: AGN with jet pointing towards us
  - BL Lacs: Weak emission lines, synchrotron peak energies.
  - Fsrq: Strong emission lines, no synchrotron peaks.
- Pulsars: Neutron stars rotating.

$$\frac{dN}{dE} = N_0 \left(\frac{E}{E_0} \right)^{-\alpha - \beta \cdot \log(E/E_0)} \quad , \quad E_{peak} = E_0 \cdot e^{\frac{2-\alpha}{2\beta}}$$

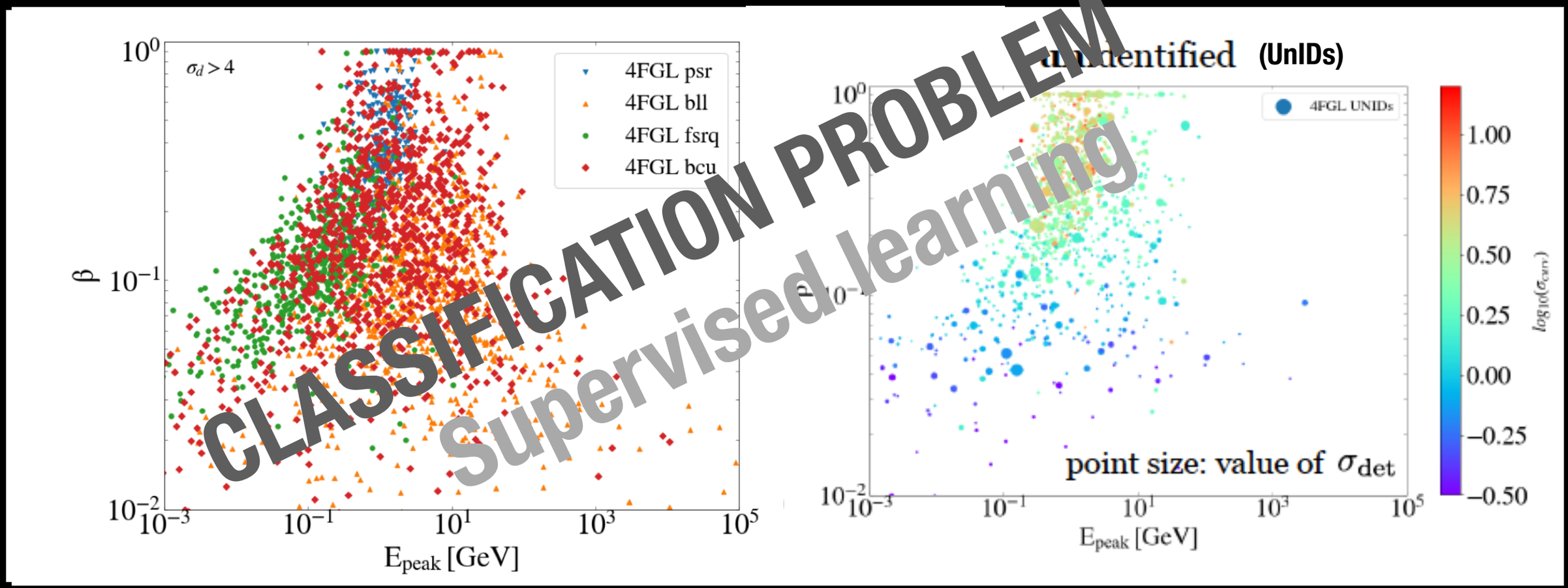
Fermi-LAT gamma-ray data and β-plot





$$\frac{dN}{dE} = N_0 \left(\frac{E}{E_0} \right)^{-\alpha - \beta \cdot \log(E/E_0)} \quad , \quad E_{peak} = E_0 \cdot e^{\frac{2-\alpha}{2\beta}}$$

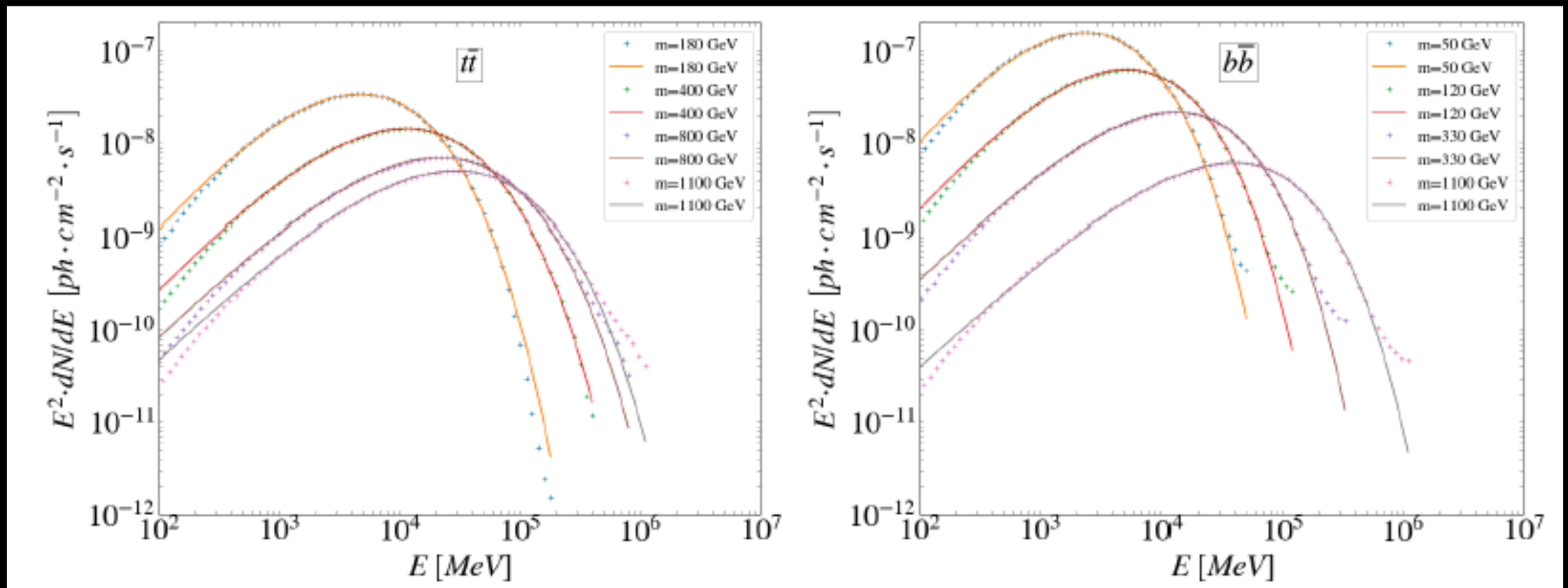
Fermi-LAT gamma-ray data and β-plot



$$\frac{dN}{dE} = N_0 \left(\frac{E}{E_0} \right)^{-\alpha - \beta \cdot \log(E/E_0)} \quad , \quad E_{peak} = E_0 \cdot e^{\frac{2-\alpha}{2\beta}}$$

Dark Matter and β - plot

The gamma-ray flux expected by a WIMP candidate annihilating in a Standard Model (SM) channel is predicted by Monte Carlo event generator software (e.g. Pythia 8). The PPPCB4DMID (Cirelli's) interpolation of those fluxes (for several WIMP masses and SM channels) makes those results user friendly. Here, we fit those interpolations with a Log-Parabola (LP):

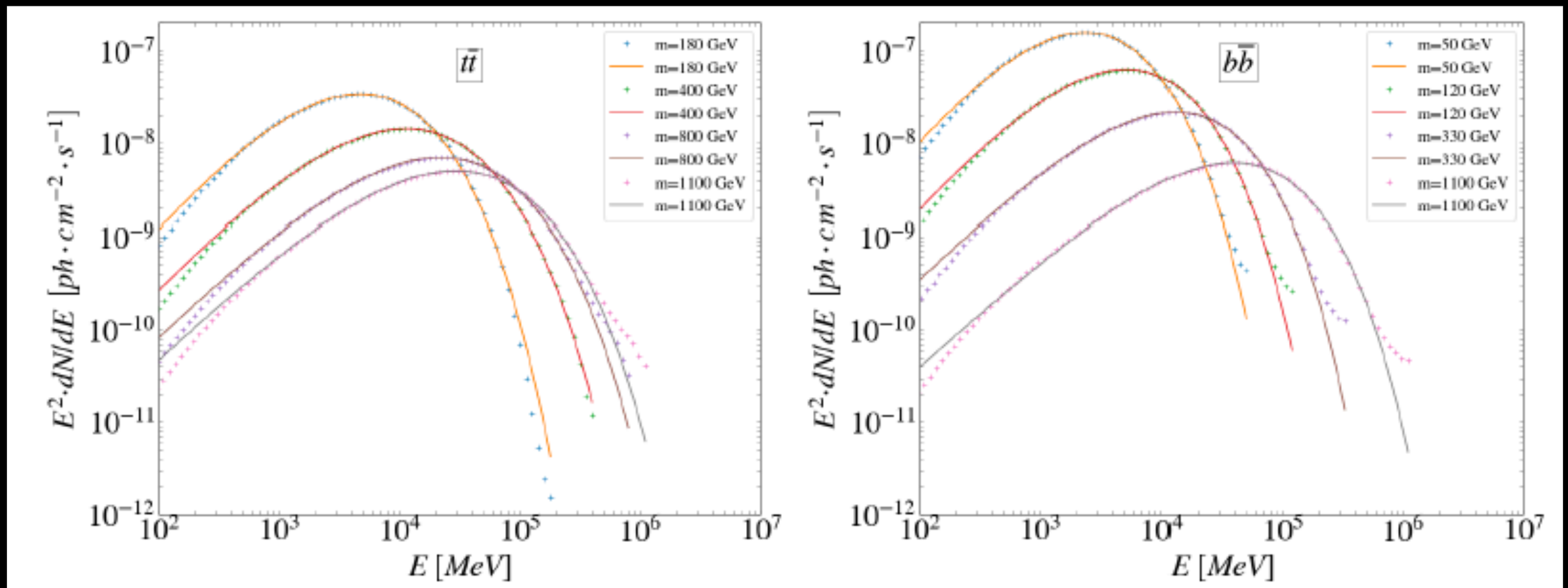


J.Coronado-Blazquez et al. JCAP07(2019)020

$$\frac{dN}{dE} = N_0 \left(\frac{E}{E_0} \right)^{-\alpha - \beta \cdot \log(E/E_0)} , \qquad E_{peak} = E_0 \cdot e^{\frac{2-\alpha}{2\beta}}$$

Dark Matter and β - plot

The gamma-ray flux expected by a WIMP candidate annihilating in a Standard Model (SM) channel is predicted by Monte Carlo event generator software (e.g. Pythia 8). The PPPCB4DMID (Cirelli's) interpolation of those fluxes (for several WIMP masses and SM channels) makes those results user friendly. Here, we fit those interpolations with a Log-Parabola (LP):

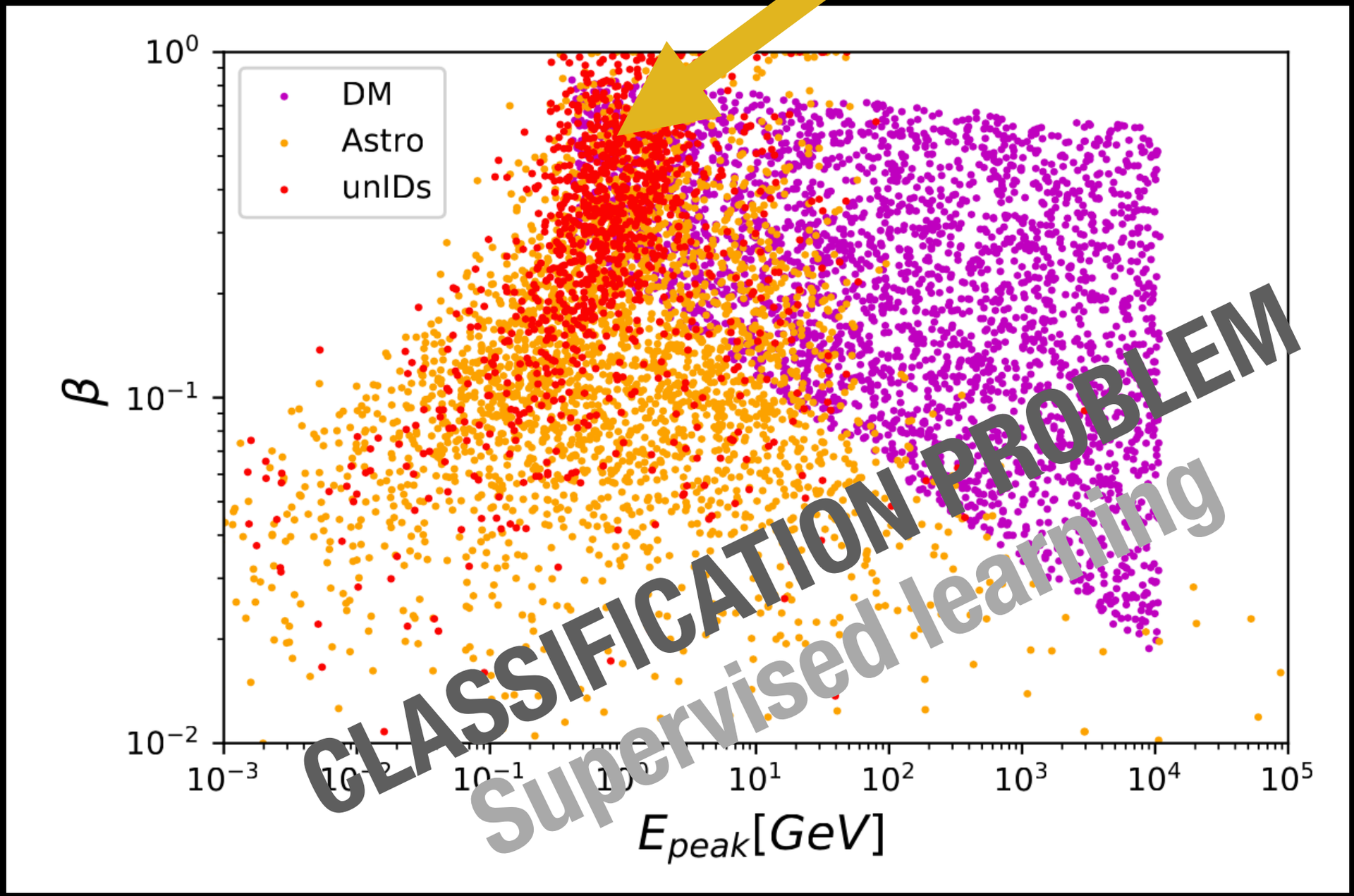


J.Coronado-Blazquez et al. JCAP07(2019)020

$$\frac{dN}{dE} = N_0 \left(\frac{E}{E_0} \right)^{-\alpha - \beta \cdot \log(E/E_0)} , \qquad E_{peak} = E_0 \cdot e^{\frac{2-\alpha}{2\beta}}$$

Dark Matter and β - plot

Degeneracy of pulsar and DM signal



V.G., et al. MNRAS 2023

Our strategy

1. Training the classification algorithm on a sample of both experimental (astrophysical - Astro) and expected (Dark Matter - DM) dataset.

2. Testing the classification accuracy on a subsample of data;

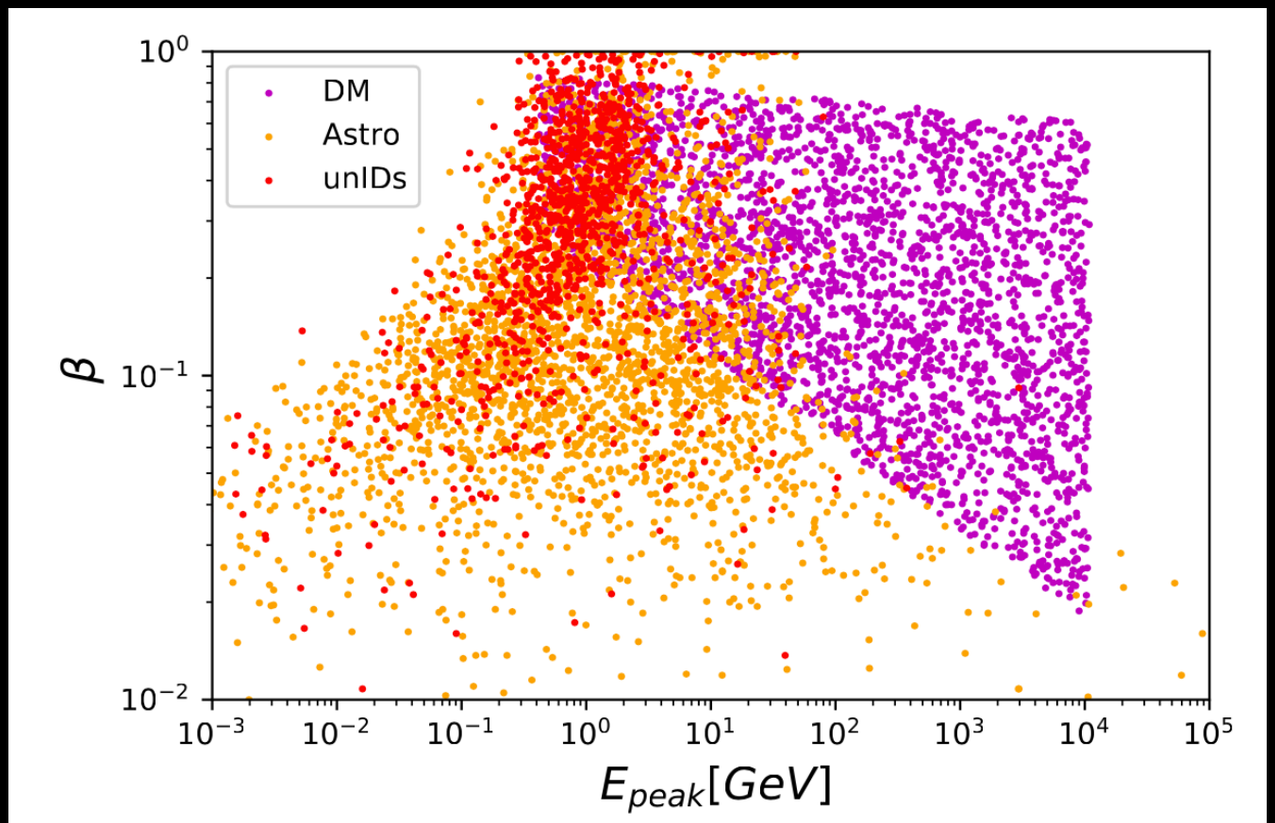
3. Predicting prospective DM-source candidates among the unIDs dataset, with assigned probability p^{DM}.

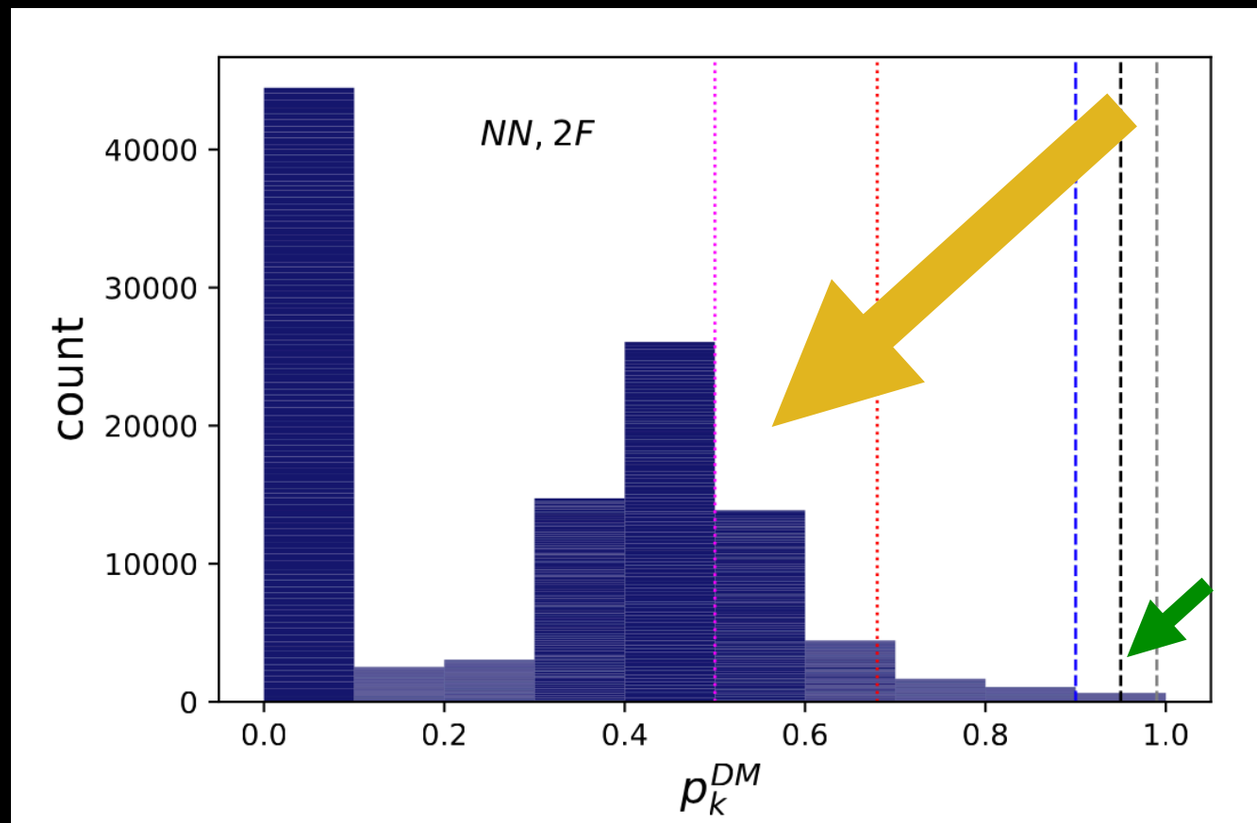
> The classification problem so far is based on two features (E_{peak}, β) .

UnIDs classification (DM): first results

Probability distribution for the full sample of 1125 unIDS and 100 classification runs

▸ 2-FEATURES (2F) CLASSIFICATION



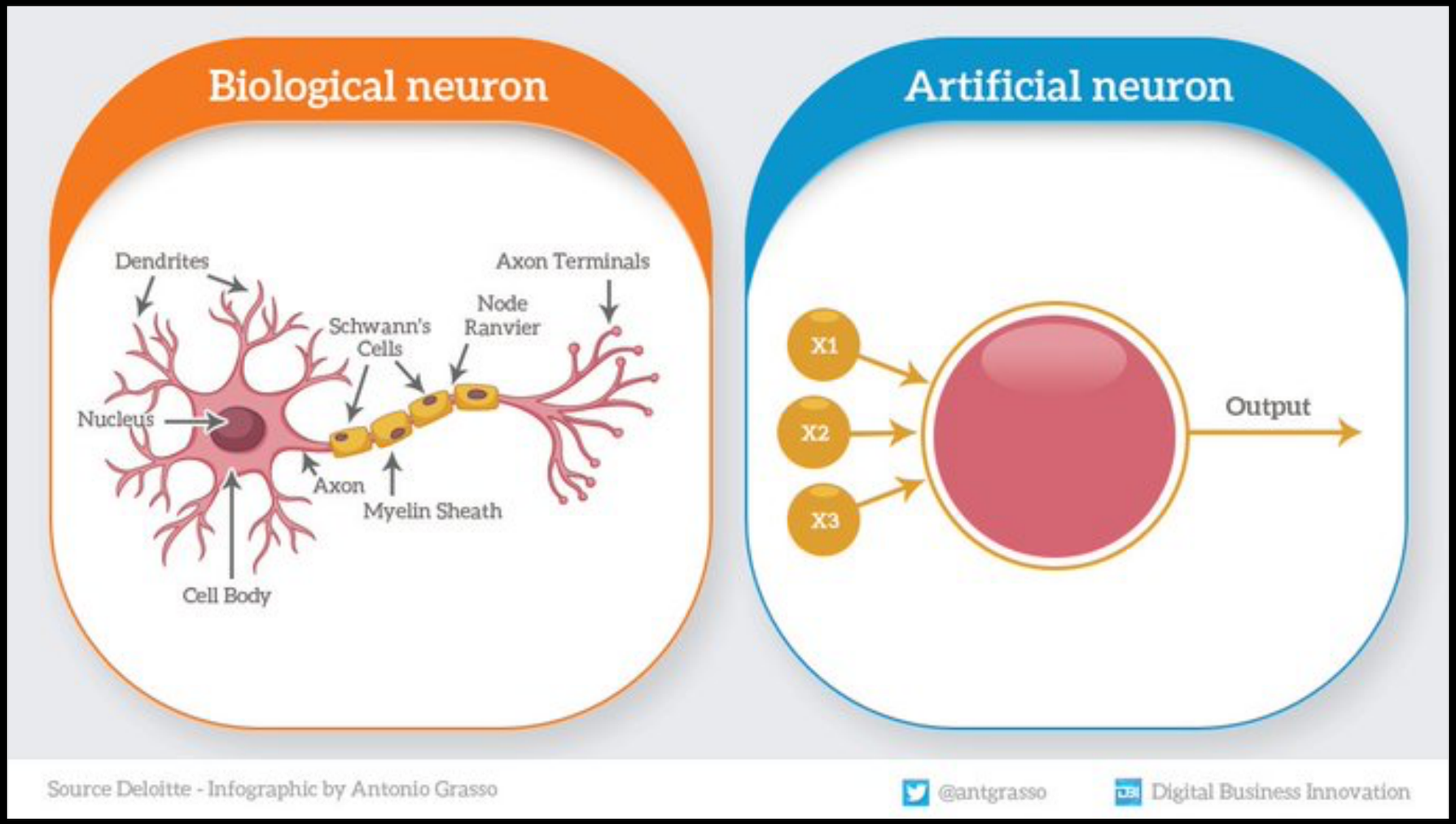


V.G. et al., MNRAS 2023

› The classification accuracy generally improves by using more features.

Neural Network

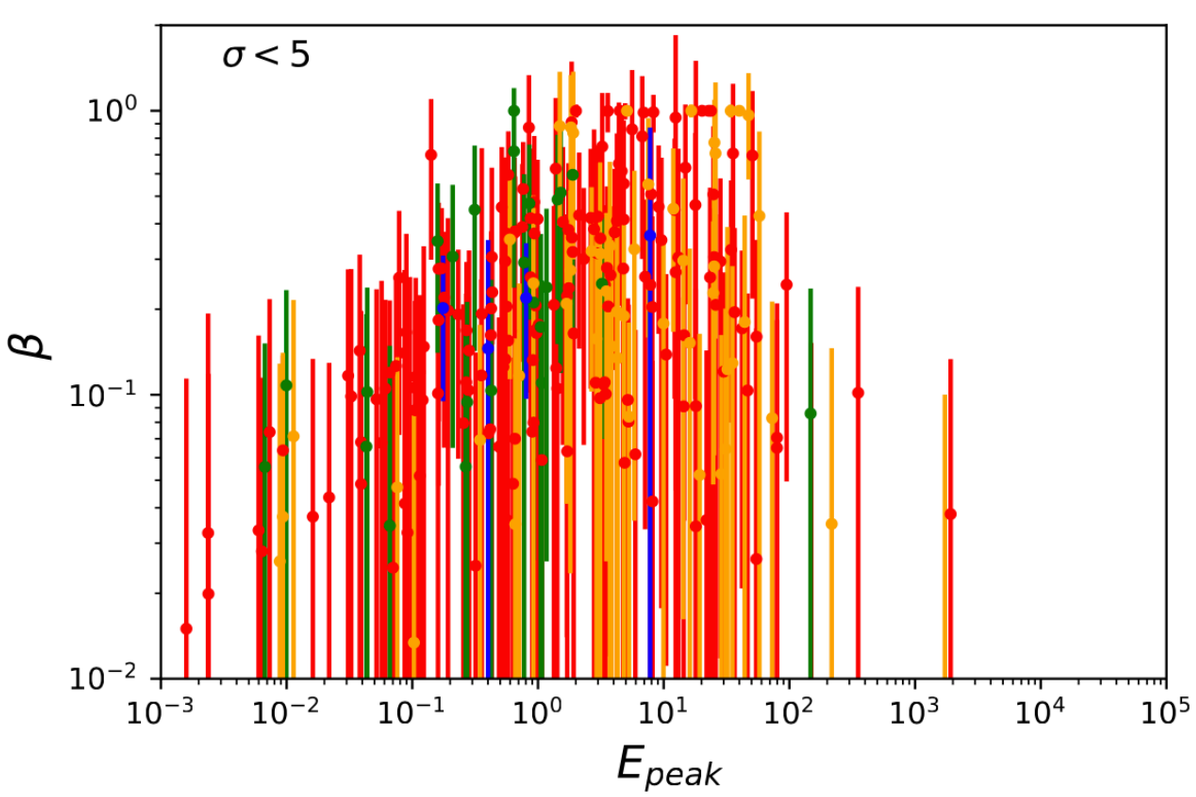
Fortunately, an artificial neuron is not like a biological one.

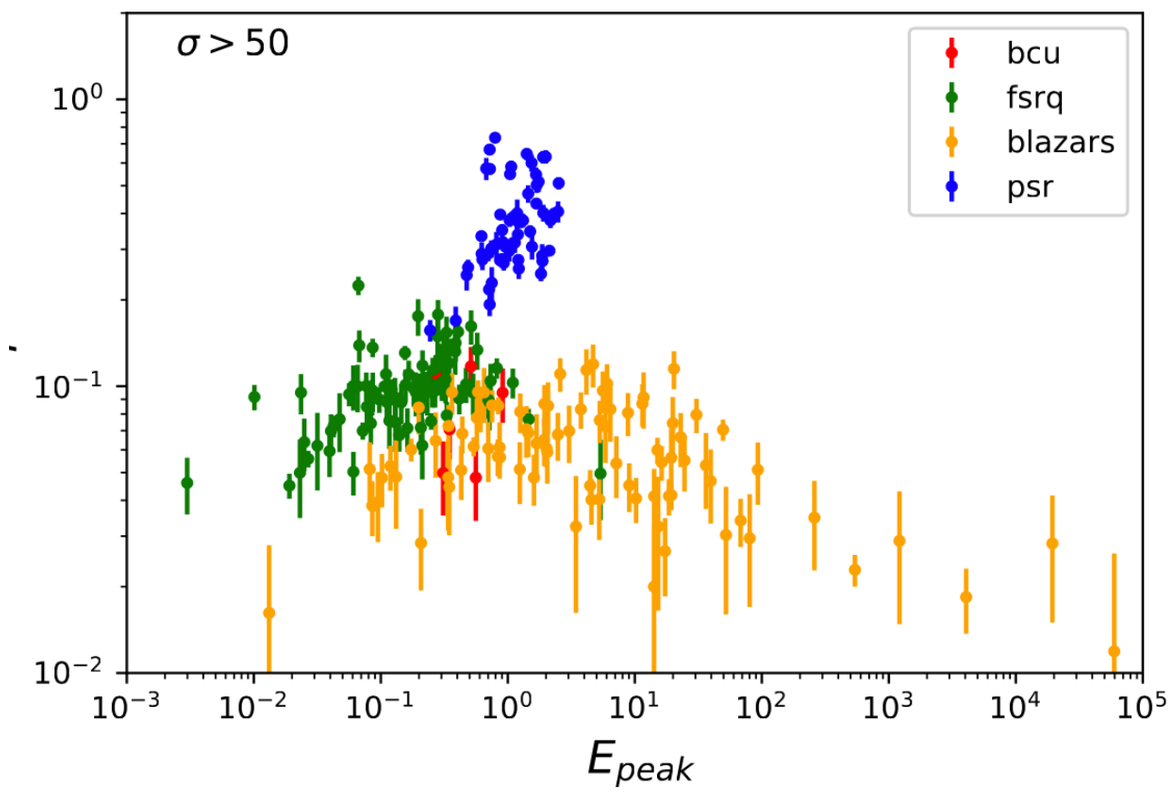


We are not automatised. Humans have much more that cannot be reproduced by an algorithm: empathy, freewill, intuition.

Systematic features

A source pre-selection based on detection significance σ_d improves the classification even only “by eye”





V.G. et al., MNRAS 2023

Systematic features

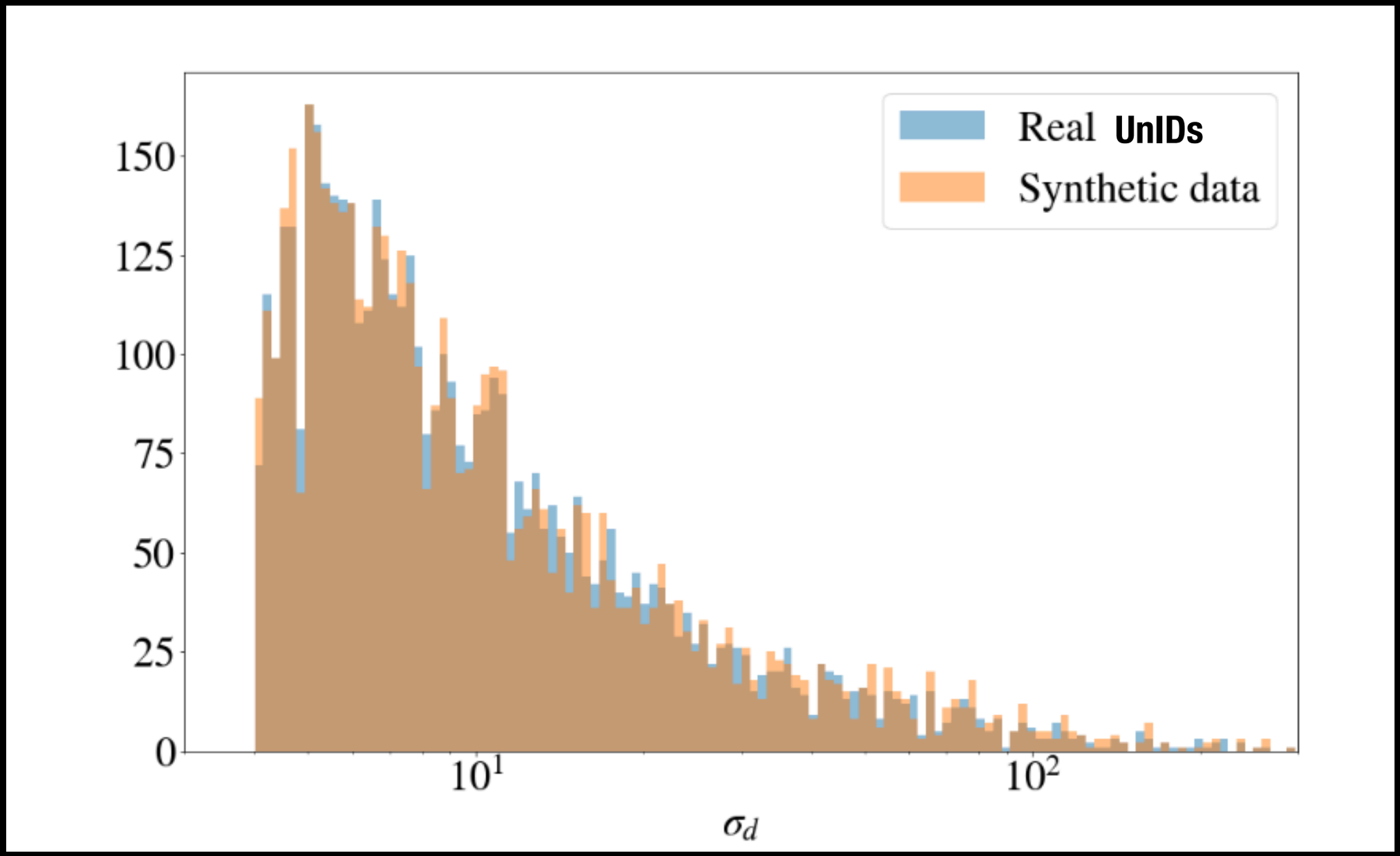
Experimental features:

1. Detection significance σ_d: given by the likelihood, i.e. depends on background template, i.e. both the diffuse model and emission model of all the other sources in the source region. The likelihood analysis determines if the source exists, its position and spectral parameters.

2. Uncertainty on β: a lower detection significance corresponds to a worst characterisation of the source spectrum.

DM Systematic features: detection significance

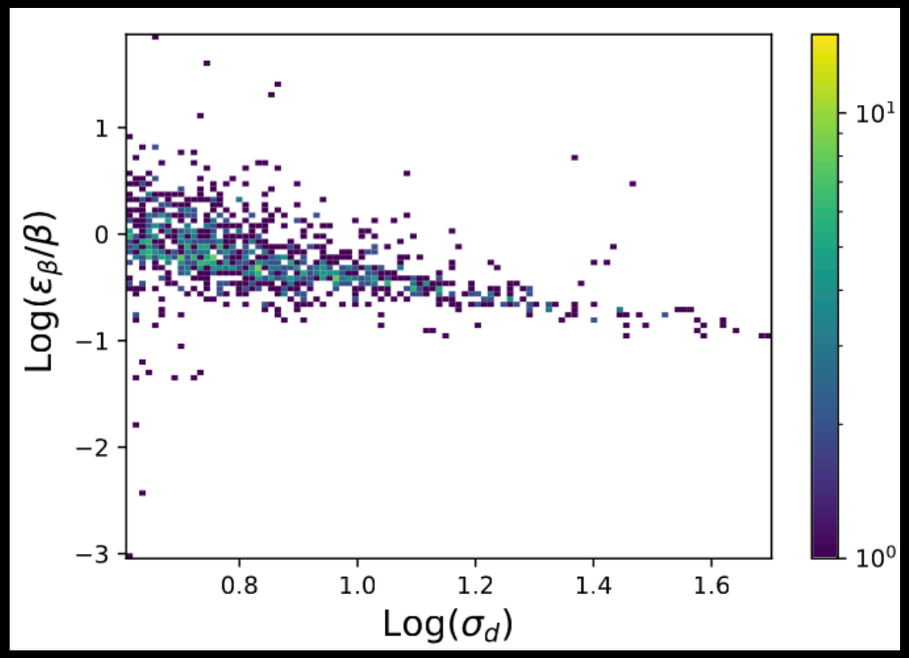
We assume that all the DM candidates are unIDs

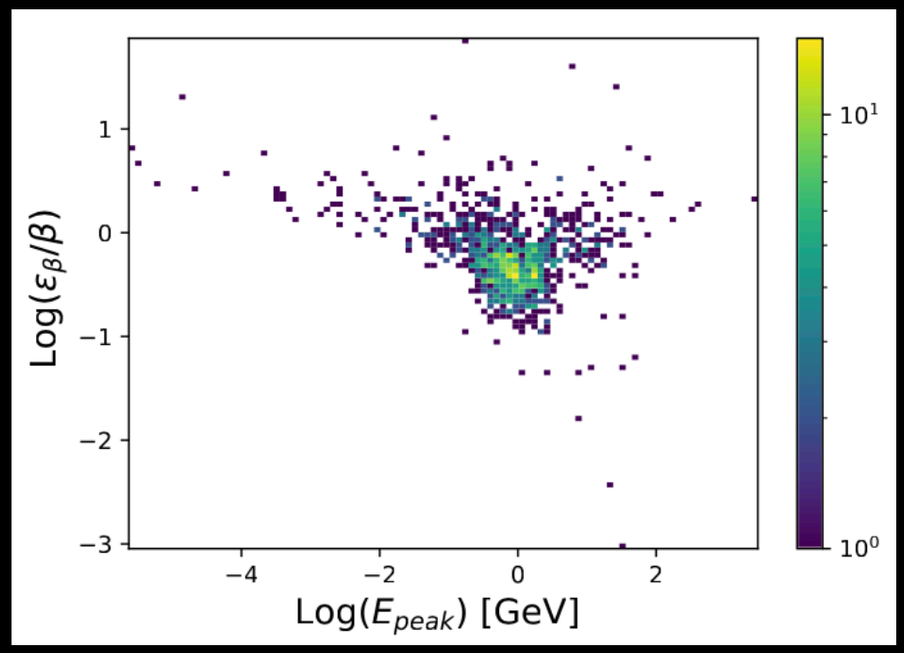


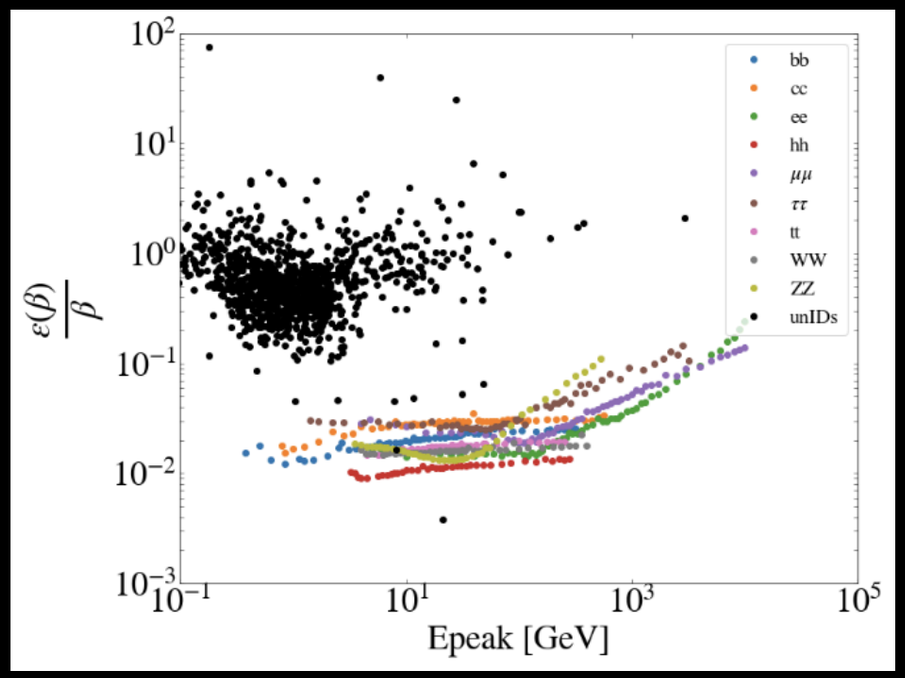
V.G. et al., MNRAS 2023

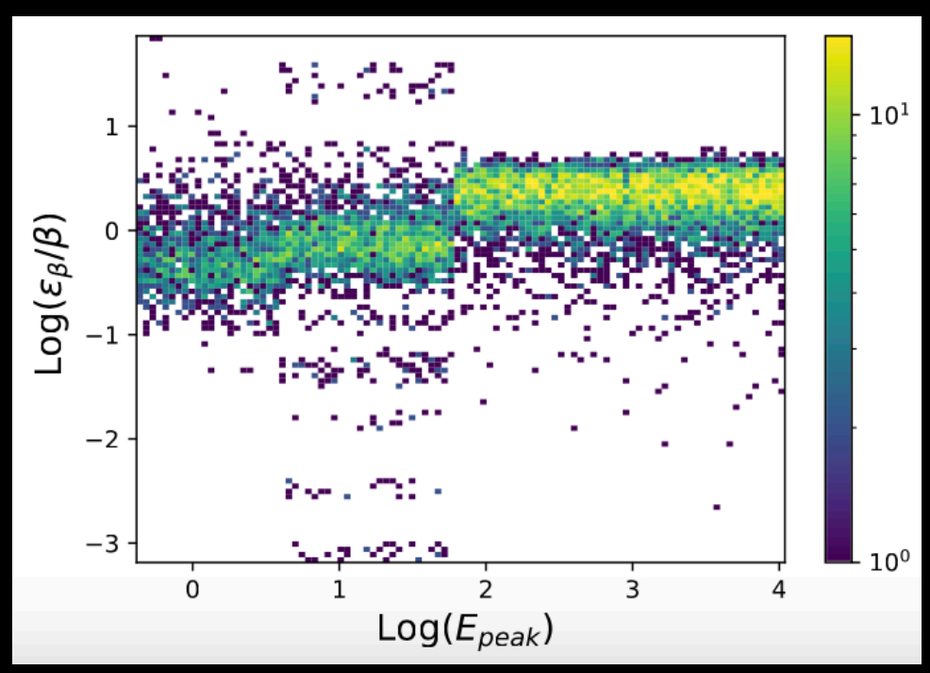
DM Systematic features: β uncertainty

We assume the β uncertainty
as for a prospective DM detection by Fermi-LAT



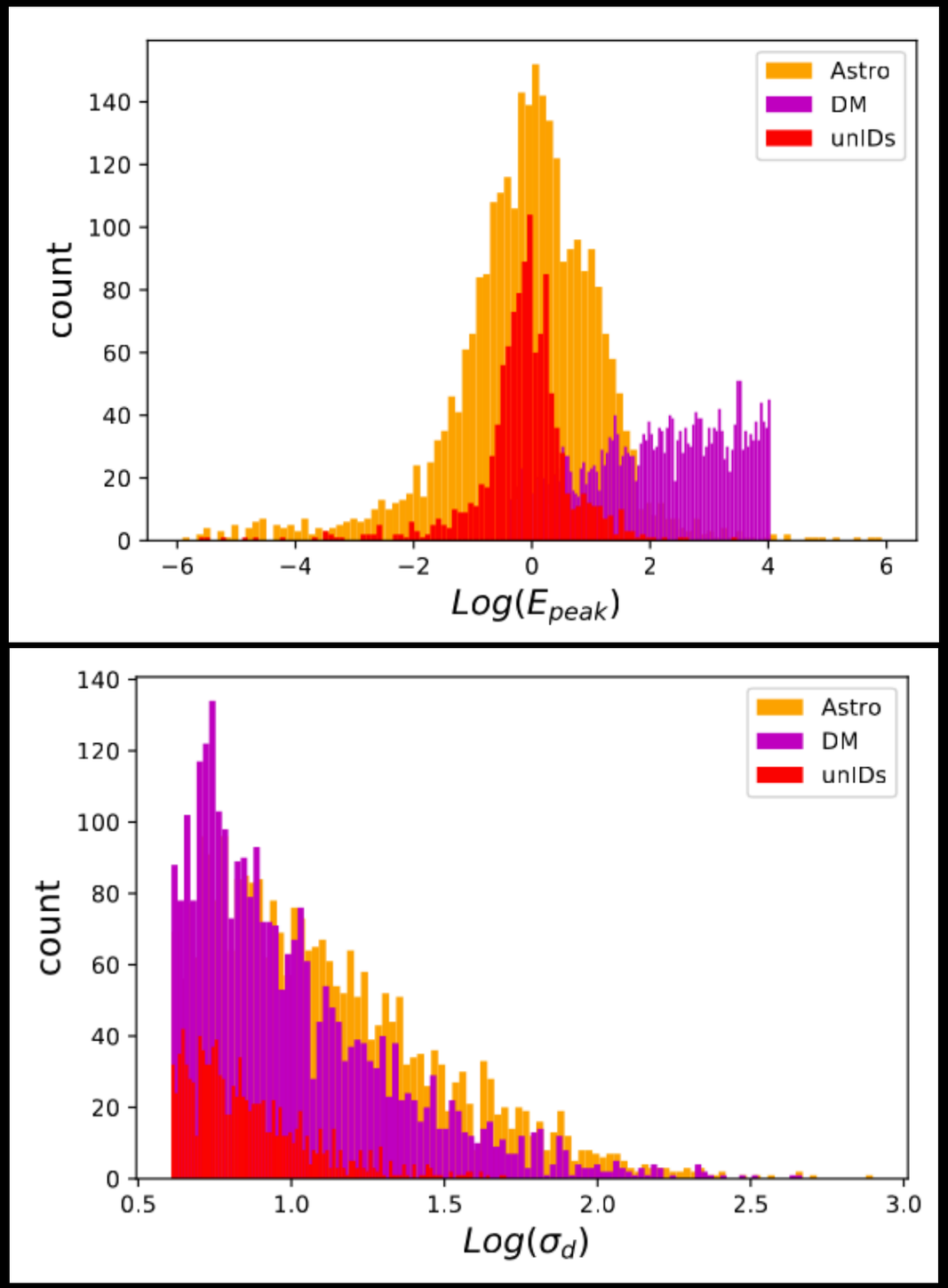


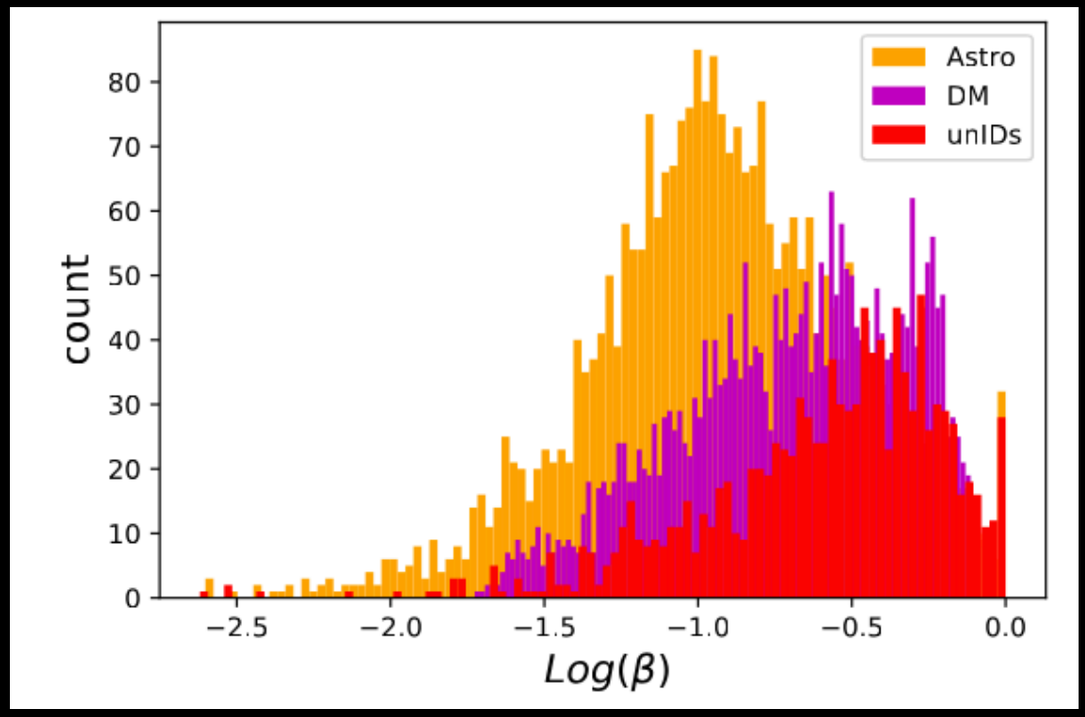


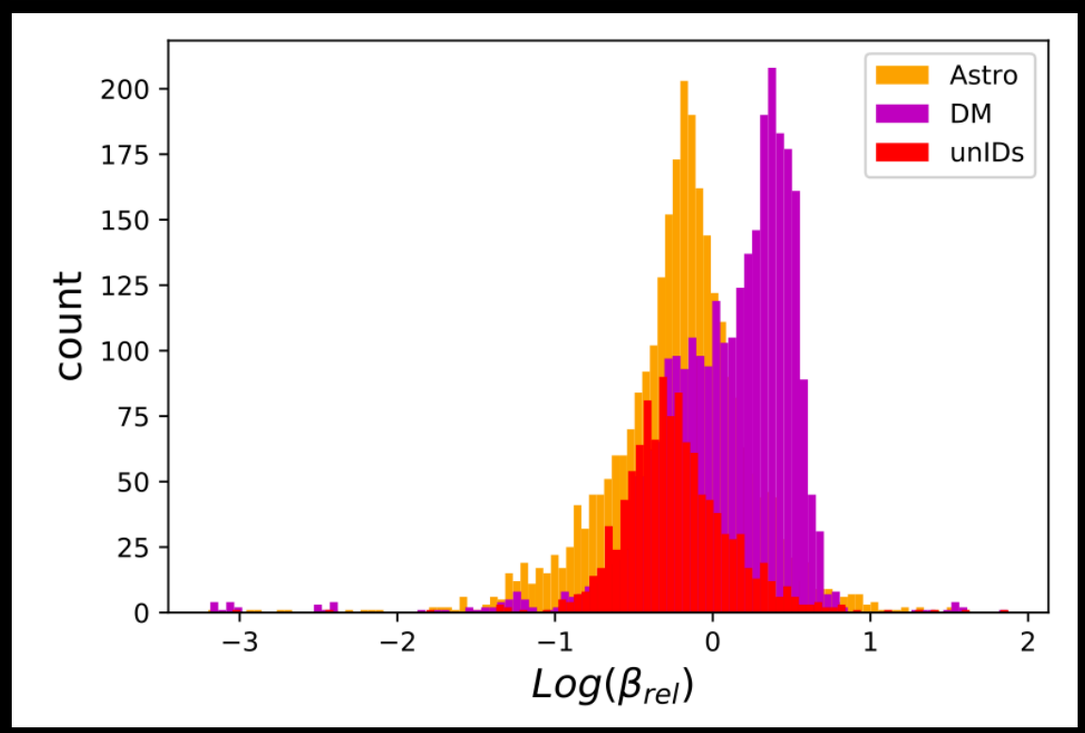


V.G. et al., MNRAS 2023

From 2F to 4F







V.G. et al., MNRAS 2023

Results

- **OVERALL ACCURACY:**
- **TRUE NEGATIVE (TN):**
- **TRUE POSITIVE (TP):**

	OA(%)	TN (%)	TP (%)
LR	**LOGISTIC REGRESSION**		
2F	84.9 ± 0.8	85.4 ± 1.5	84.4 ± 1.4
4F	86.0 ± 0.9	86.7 ± 1.5	85.2 ± 1.3
NN	**NEURAL NETWORK**		
2F	86.2 ± 0.8	86.1 ± 3.0	86.4 ± 3.4
4F	$\mathbf{93.3 \pm 0.7}$	$\mathbf{94.7 \pm 1.7}$	$\mathbf{91.8 \pm 1.5}$

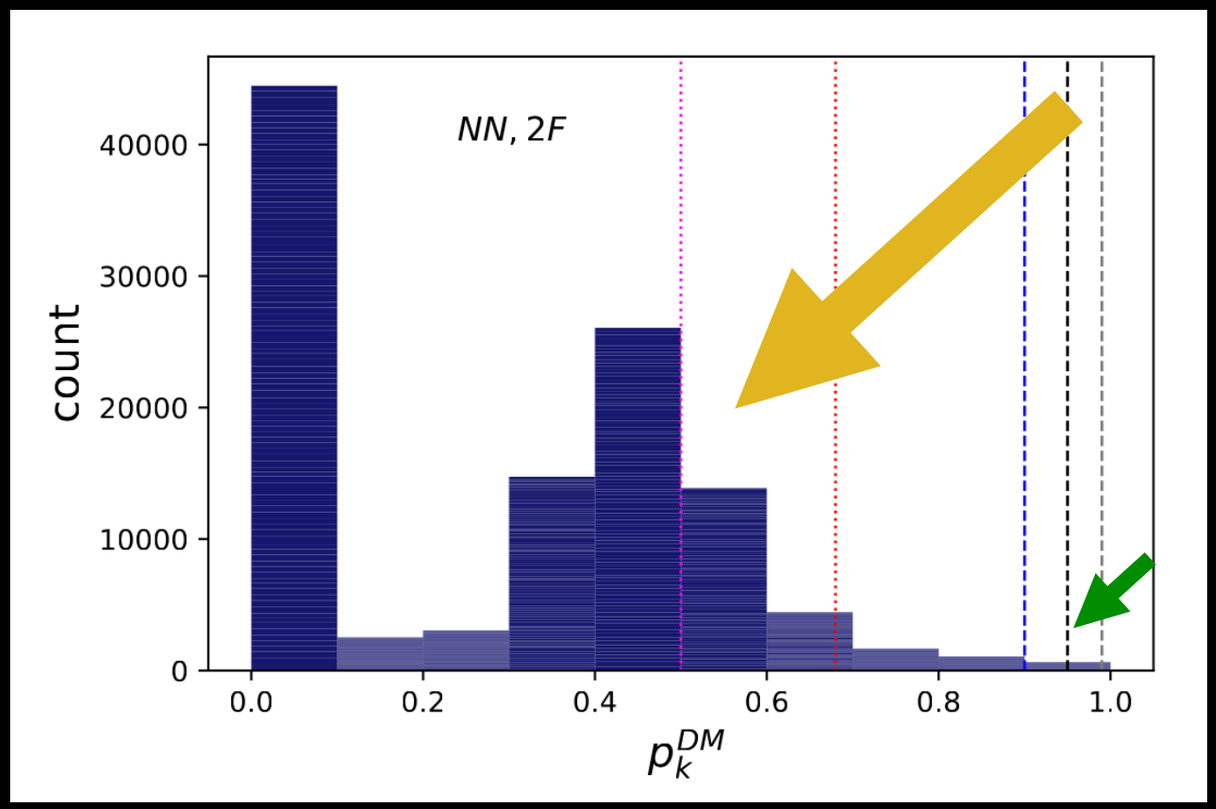
V.G. et al., MNRAS 2023

THE OVERALL CLASSIFICATION ACCURACY IMPROVES WITH THE INTRODUCTION OF SYSTEMATIC FEATURES.

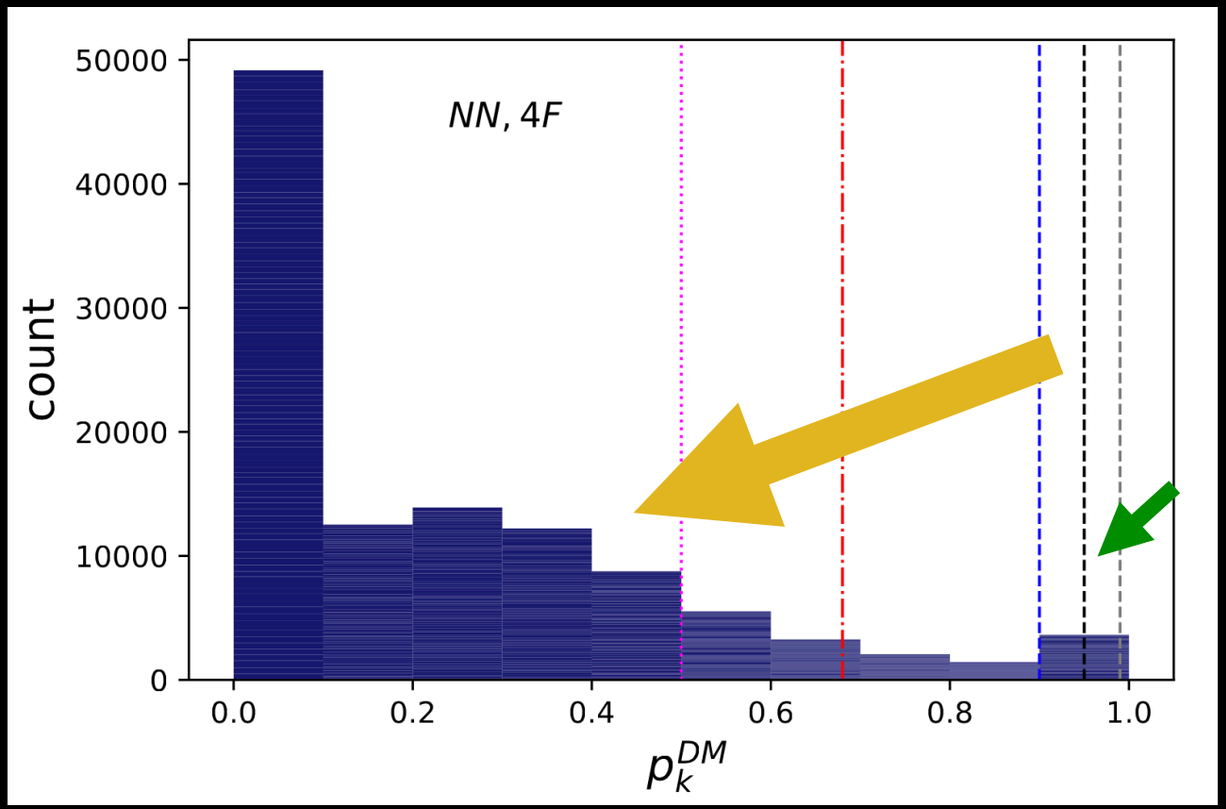
UnIDs classification (DM)

Probability distribution for the full sample of 1125 unIDS and 100 classification runs

- **2-FEATURES (2F) CLASSIFICATION**



- **4-FEATURES (4F) CLASSIFICATION**



V.G. et al., MNRAS 2023

The degeneracy problem is partially solved

Instituto de Física Teórica UAM-CSIC

EXCELENCIA SEVERO OCHOA

Home | Registration | Speakers | Program | Organizers | Sponsorship | Travel | ML-Poster | Photos | Conduct

AI goes MAD²

Home

This is the second version of the ML-AI 3-day workshop at the Instituto de Física Teórica (IFT).

The workshop will be held at the facilities of the IFT on the campus of the Universidad Autónoma de Madrid (UAM).

October 14th-16th 2024

Instituto de Física Teórica UAM-CSIC

EXCELENCIA SEVERO OCHOA

Home Registration Speakers Program Organizers Sponsorship Travel ML-Poster Photos Conduct

AI goes MAD²

Organizers

Name	Post	Institution	Email
Aguilar-Saavedra, Juan Antonio	Organizer	IFT-UAM/CSIC	jaas@ugr.es
Alestas, George	Organizer	IFT-UAM/CSIC	g.alestas@csic.es
Arganda, Ernesto	Organizer	IFT-UAM/CSIC	ernesto.arganda@csic.es
de los Rios, Martín	Organizer	IFT-UAM/CSIC	martin.delosrios@uam.es
Gammaldi, Viviana	Organizer	IFT-UAM/CSIC	viviana.gammaldi@uam.es
Nesseris, Savvas	Organizer	IFT-UAM/CSIC	savvas.nesseris@csic.es
Perez , Andres	Organizer	IFT-UAM/CSIC	perez.andres.daniel@gmail.com
Sánchez Conde, Miguel Ángel	Organizer	IFT-UAM/CSIC	miguel.sanchezconde@uam.es
Sandá Seoane, Rosa María	Organizer	IFT-UAM/CSIC	rosa.sanda@uam.es
Vergel, Monica	Secretary	IFT-UAM/CSIC	visitantes-eventos.ift@csic.es

10 results

Summary

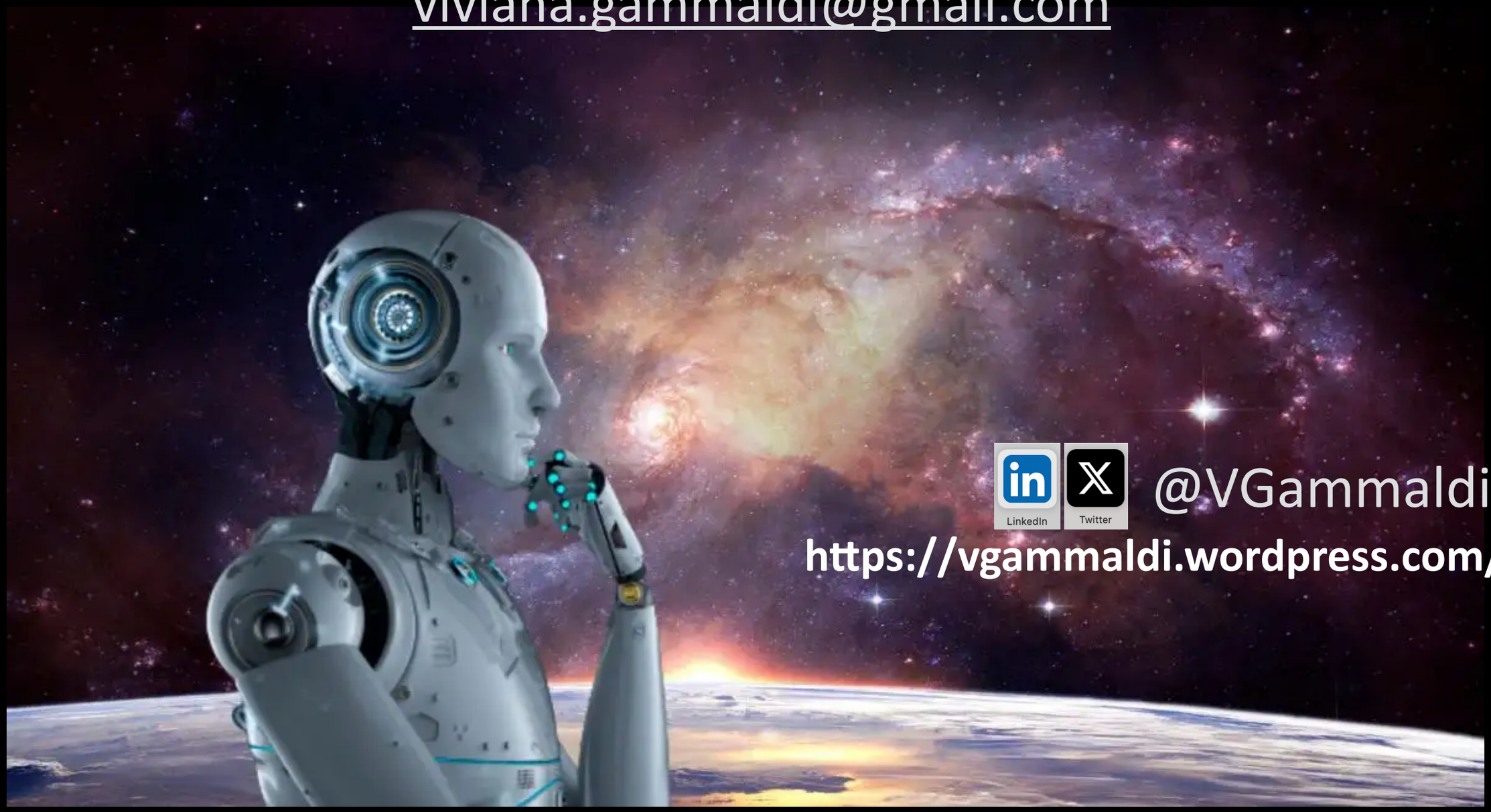
- Machine learning is a branch of artificial intelligence that allows machines to learn and identify patterns among data and make predictions.
- Different kinds of learning (supervised, unsupervised, reinforcement) allow to approach different problems (classification, regression, clustering, anomaly detection, etc.) and solve them.
- Scikit-learn is a user friendly software for developing machine learning in Python.
- Neural networks are a powerful algorithm to solve, e.g. nonlinear classification problems.
- Machine learning improved the classification accuracy of Fermi-LAT unidentified gamma-ray sources, partially solving the degeneracy problem between astrophysical and DM sources.
- Many other possible applications exist.

Thank you for your attention

For further questions or collaborations:

viviana.gammaldi@ceu.es

viviana.gammaldi@gmail.com



Picture from the web: El Universal

Back-up slides

Gradient descent - logistic regression

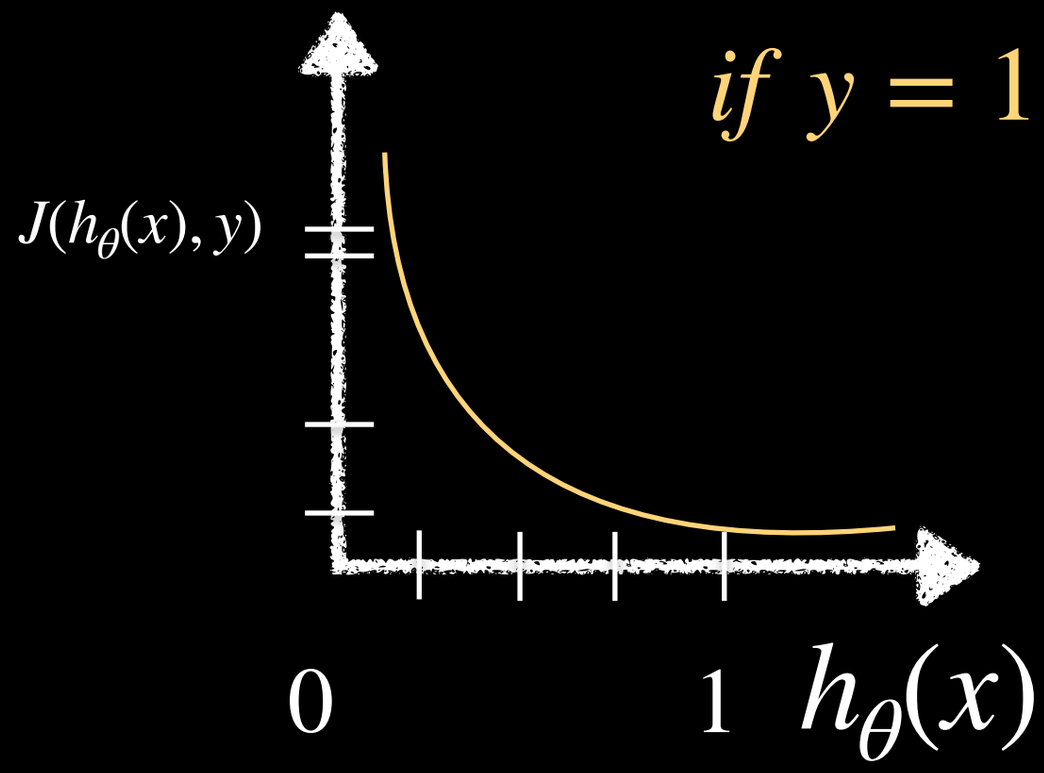
$$J(\Theta) = -\frac{1}{n}[\sum_{i=1}^{n} y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^{i}))]$$

$$J(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) \quad if\ y = 1 \\ -log(1 - h_\theta(x)) \quad if\ y = 0 \end{cases}$$

$$J(h_\theta(x), y) = \begin{cases} = 0 \quad if\ h_\theta(x) = 1\ and\ y = 1 \\ = 0 \quad if\ h_\theta(x) = 0\ and\ y = 0 \end{cases}$$

If $h_\theta(x) = 0, P(y = 1 \mid x; \theta) = 0$

If $h_\theta(x) = 1, P(y = 1 \mid x; \theta) = 1$



Gradient descent - logistic regression

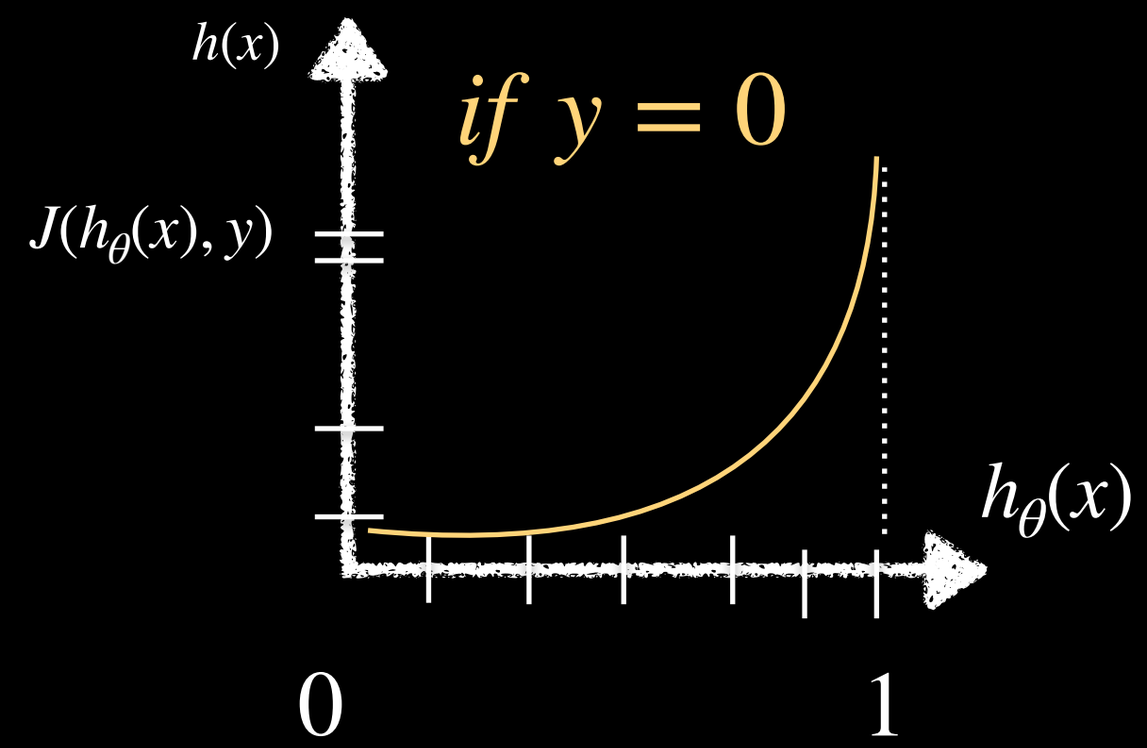
$$J(\Theta) = -\frac{1}{n}[\sum_{i=1}^{n} y^{(i)} \log(h_\theta(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_\theta(x^i))]$$

$$J(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)) & if\ y = 1 \\ -log(1 - h_\theta(x)) & if\ y = 0 \end{cases}$$

$$J(h_\theta(x), y) = \begin{cases} = 0 & if\ h_\theta(x) = 1\ and\ y = 1 \\ = 0 & if\ h_\theta(x) = 0\ and\ y = 0 \end{cases}$$

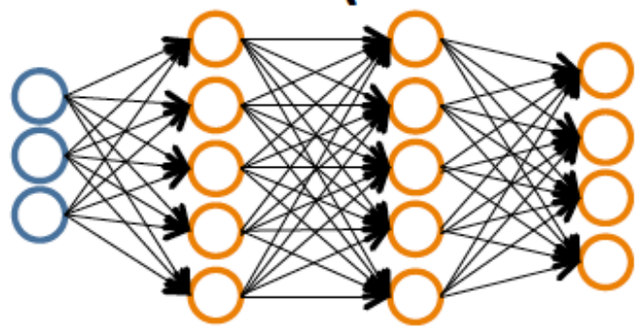
If $h_\theta(x) = 0, P(y = 0 \mid x; \theta) = 1$

If $h_\theta(x) = 1, P(y = 0 \mid x; \theta) = 0$



Neural Network

Neural Network (Classification)



$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})\}$

$L =$ total no. of layers in network

$s_l =$ no. of units (not counting bias unit) in layer l

Binary classification

$y = 0$ or 1

1 output unit

Multi-class classification (K classes)

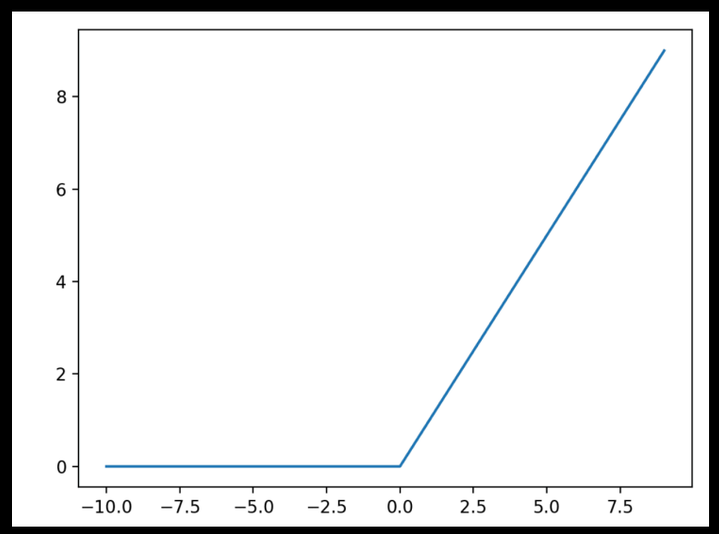
$y \in \mathbb{R}^K$ E.g. $\begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$

pedestrian car motorcycle truck

K output units

Rectified Linear Activation Function (ReLu)

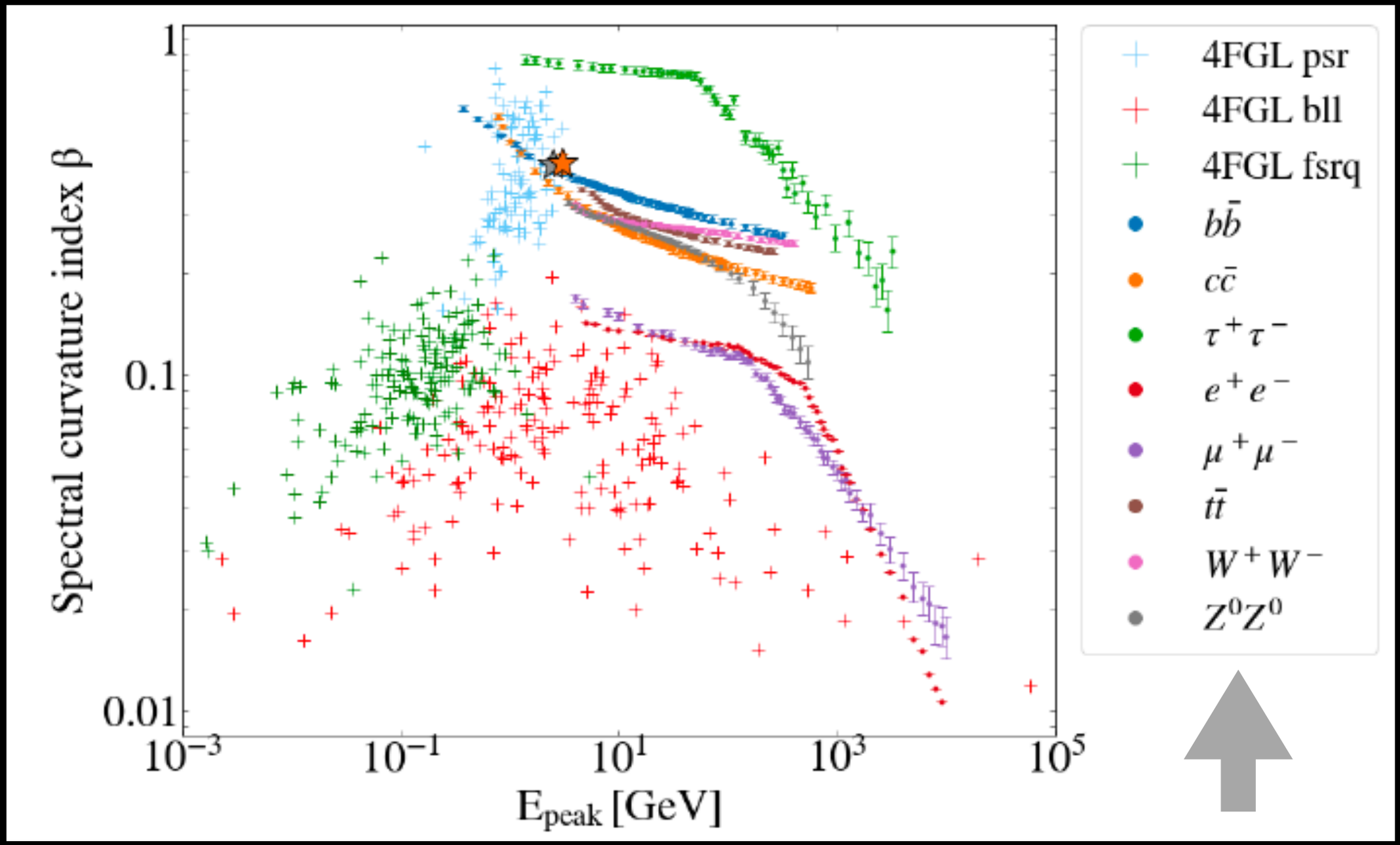
f(x) = max(0, x)



The **softmax function** converts a vector of *K* real numbers into a probability distribution of *K* possible outcomes.

$$P(y = j \mid \mathbf{x}) = \frac{e^{\mathbf{x}^T \theta_j}}{\sum_{k=1}^{K} e^{\mathbf{x}^T \theta_k}}$$

Dark Matter and β - plot



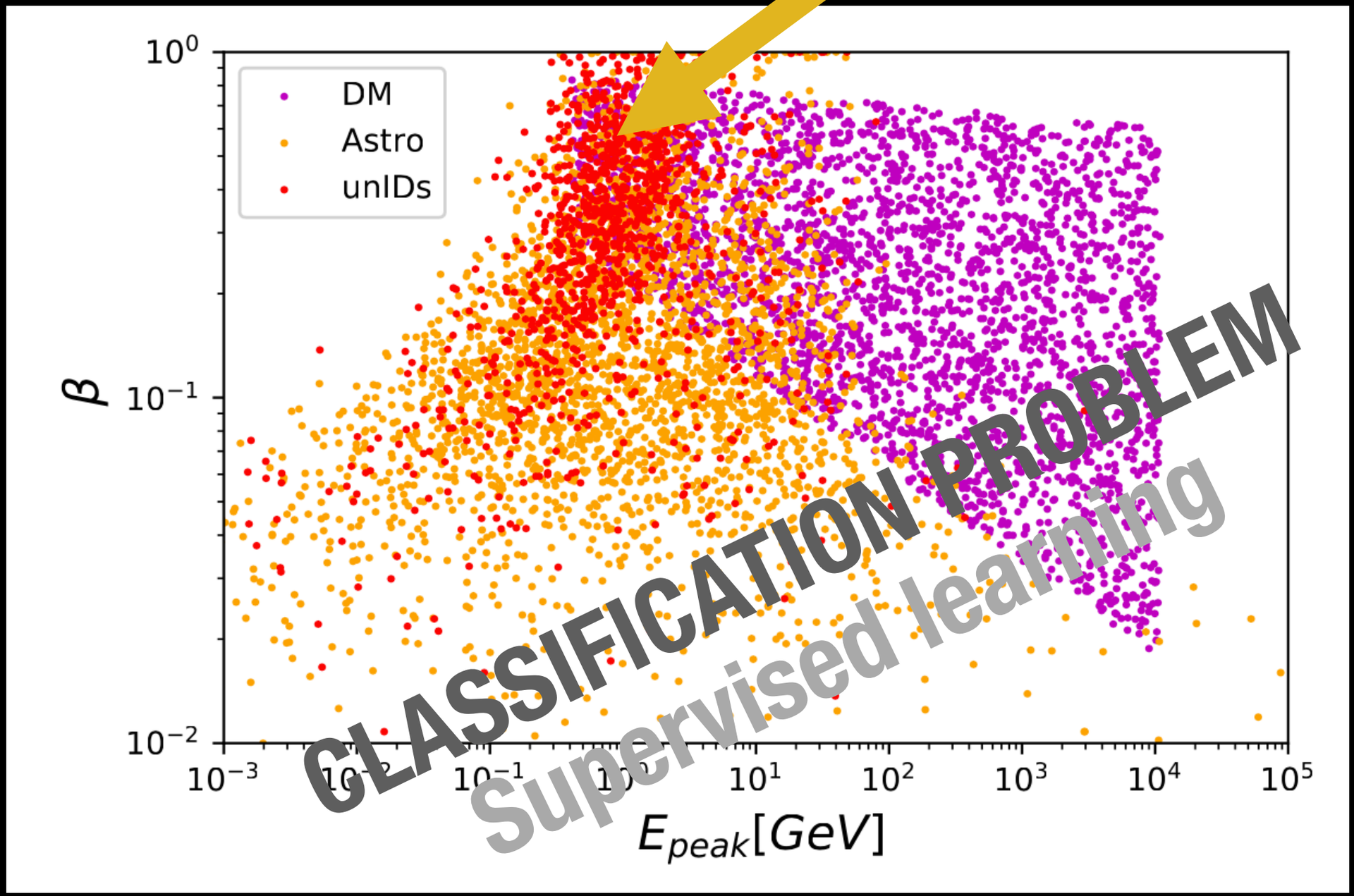
J. Coronado-Blázquez et al., JCAP11(2019)045

Model independent

Dark Matter and β - plot

$$\frac{dN}{dE} = B_r \left(\frac{dN}{dE}\right)_{C_1} + (1 - B_r)\left(\frac{dN}{dE}\right)_{C_2}$$

Degeneracy of pulsar and DM signal



V.G. et al., MNRAS 2023

Instituto de Física Teórica UAM-CSIC

EXCELENCIA SEVERO OCHOA

Home | Registration | Speakers | Program | Organizers | Sponsorship | Travel | ML-Poster | Photos | Conduct

AI goes MAD²

Speakers

Name	Institution	Talk
Cuesta-Lazaro , Carol	IAIFI/MIT	TBA
Gabrié , Marylou	École Polytechnique, CMAP	TBA
Garrafo , Cecilia	Harvard	TBA
Krenn , Mario	Max Planck Institute for the Science of Light	TBA
Thais , Savannah J.	Columbia	TBA
Wulzer , Andrea	IFAE	TBA

6 results