A SEARCH FOR DARK MATTER AMONG FERMI-LAT UNIDENTIFIED SOURCES WITH SYSTEMATIC FEATURES IN MACHINE LEARNING

MNRAS 520, 1348–1361 (2023), [arXiv:2207.09307]

VIVIANA GAMMALDI

Juan de la Cierva - Incorporación
Departamento de Física Teórica, Universidad Autónoma de Madrid (UAM), Madrid, Spain
Instituto de Física Teórica (IFT UAM-CSIC), Madrid, Spain

In collaboration with: B. Zaldívar, M. A. Sánchez-Conde, J. Coronado-Blázquez

"Cosmology in Miramare 2023"
August 28- September 1, 2023 - Trieste, Italy.

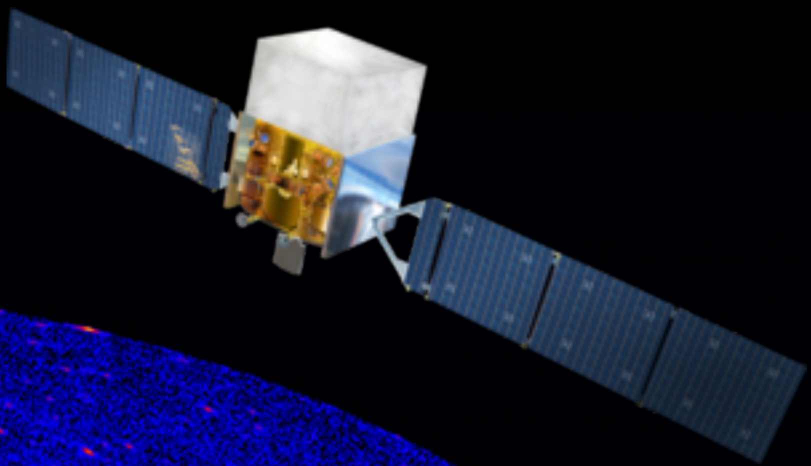
OUTLINE

- FERMI-LAT SATELLITE, GAMMA-RAY DATA & BETA-PLOT
- DARK MATTER & BETA-PLOT
- SYSTEMATIC FEATURES
- CLASSIFICATION ALGORITHMS AND SETUPS
- RESULTS

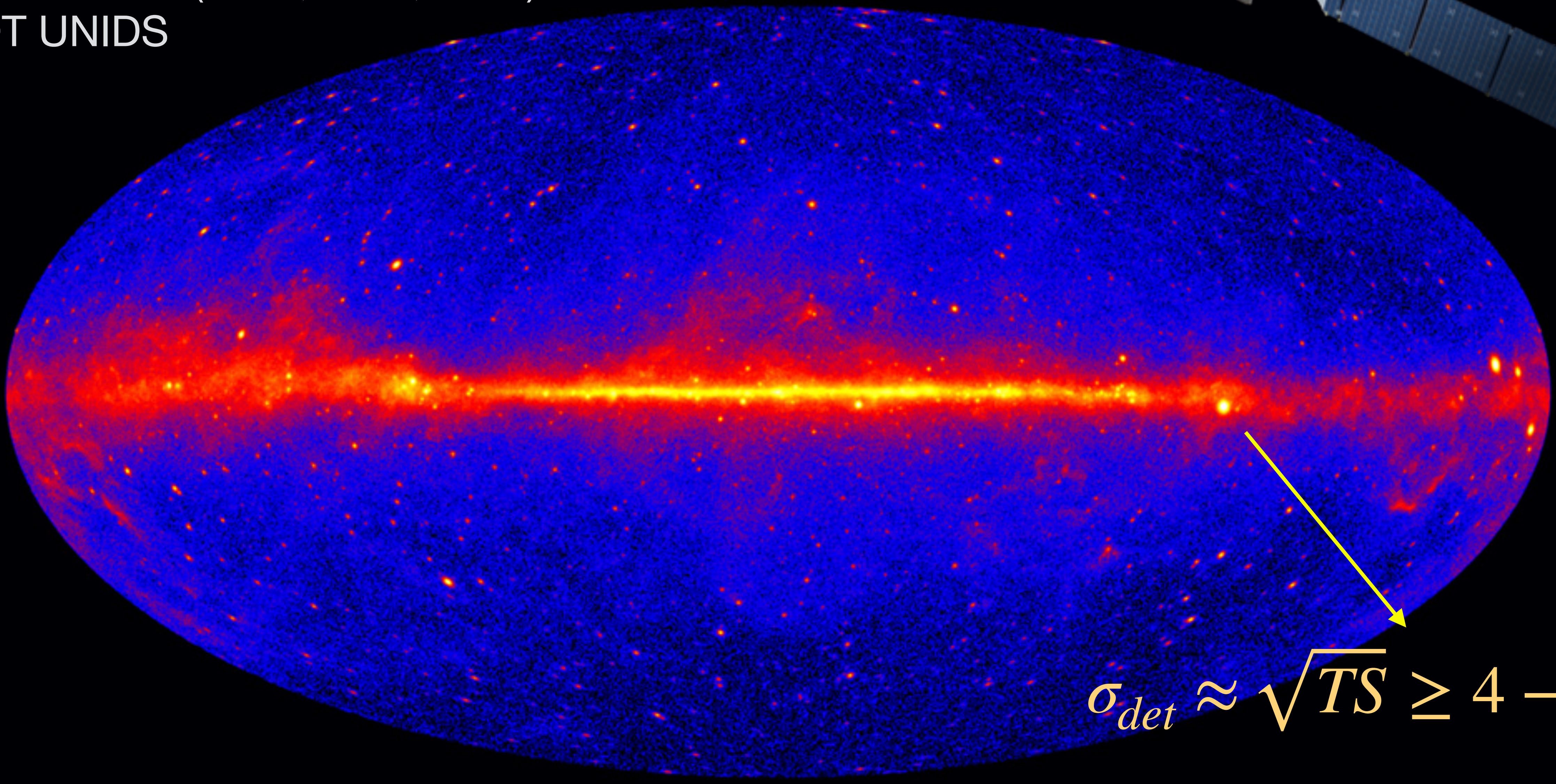
OUTLINE

- FERMI-LAT SATELLITE, GAMMA-RAY DATA & BETA-PLOT
- DARK MATTER & BETA-PLOT
- SYSTEMATIC FEATURES
- CLASSIFICATION ALGORITHMS AND SETUPS
- RESULTS

FERMI-LAT GAMMA-RAY DATA & BETA-PLOT



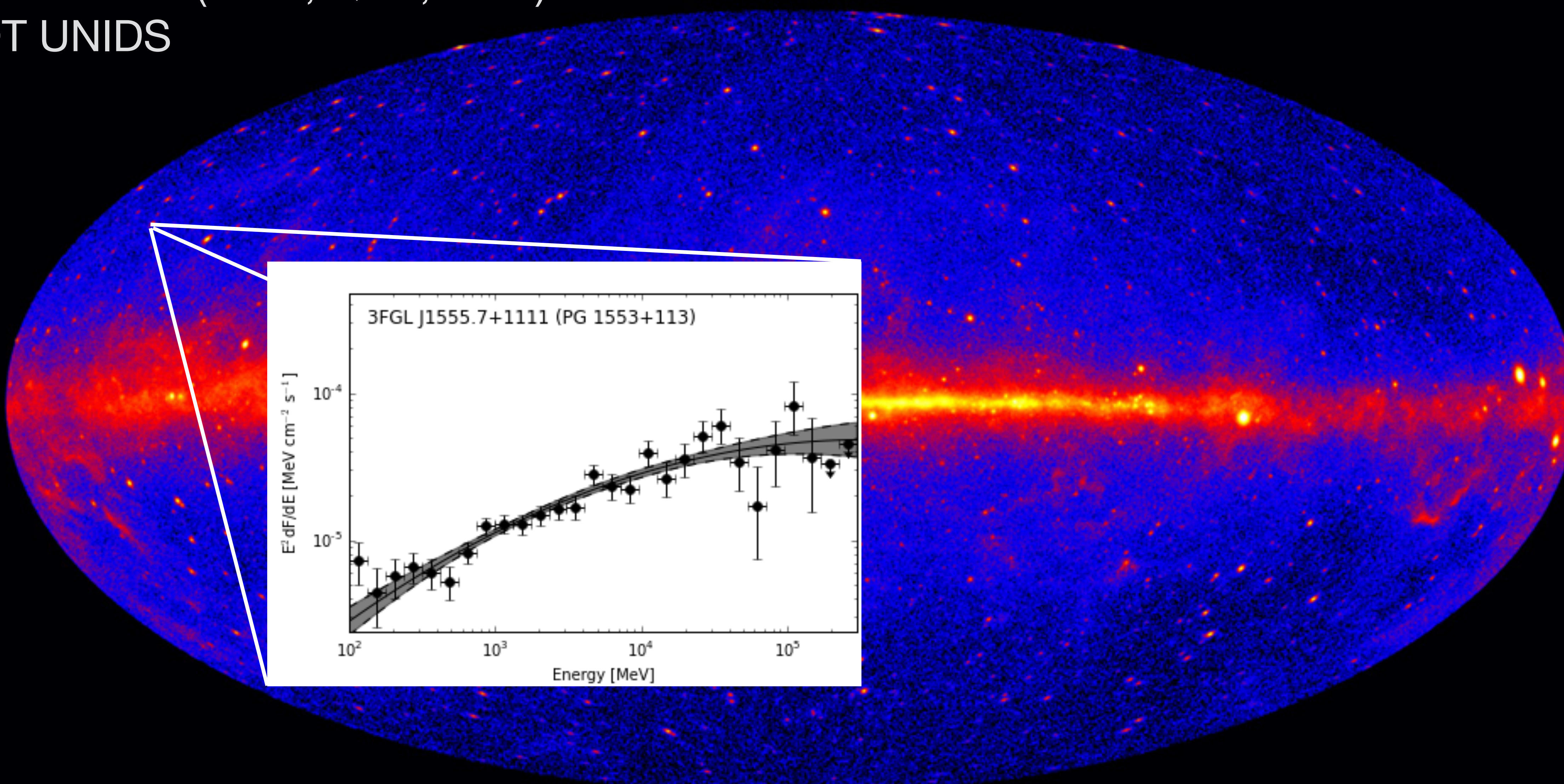
4FGL catalogue:
TOT ASTRO (PSR, QSR, BCU)
TOT UNIDS



$$\sigma_{det} \approx \sqrt{TS} \geq 4 - 5$$

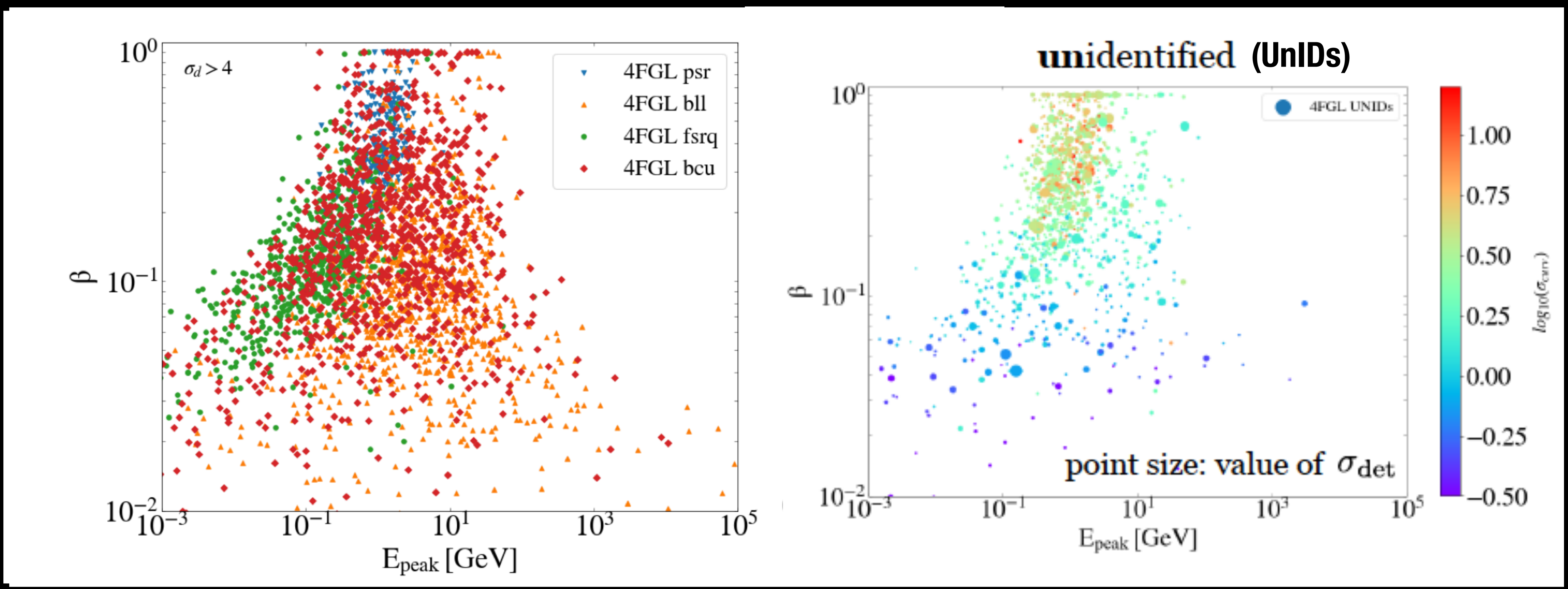
FERMI-LAT GAMMA-RAY DATA & BETA-PLOT

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



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

FERMI-LAT GAMMA-RAY DATA & BETA-PLOT

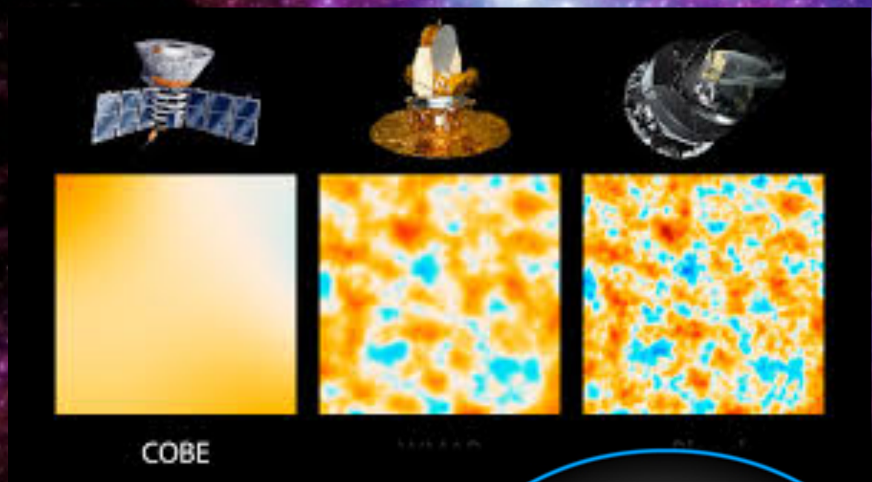


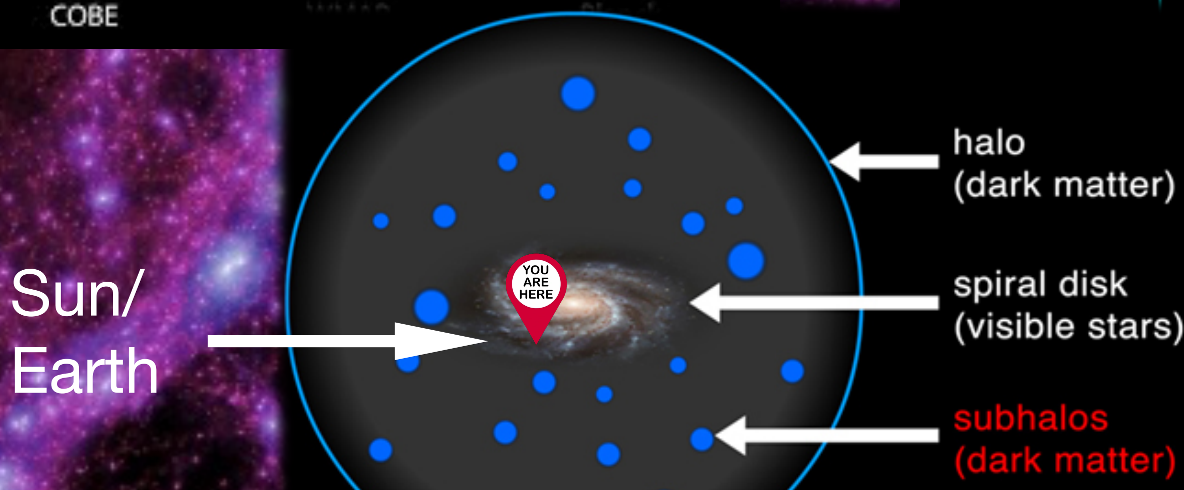
$$\frac{dN}{dE} = N_0 \left(\frac{E}{E_0}\right)^{-\alpha-\beta\cdot\log(E/E_0)} , \qquad E_{peak} = E_0 \cdot e^{\frac{2-\alpha}{2\beta}}$$

OUTLINE

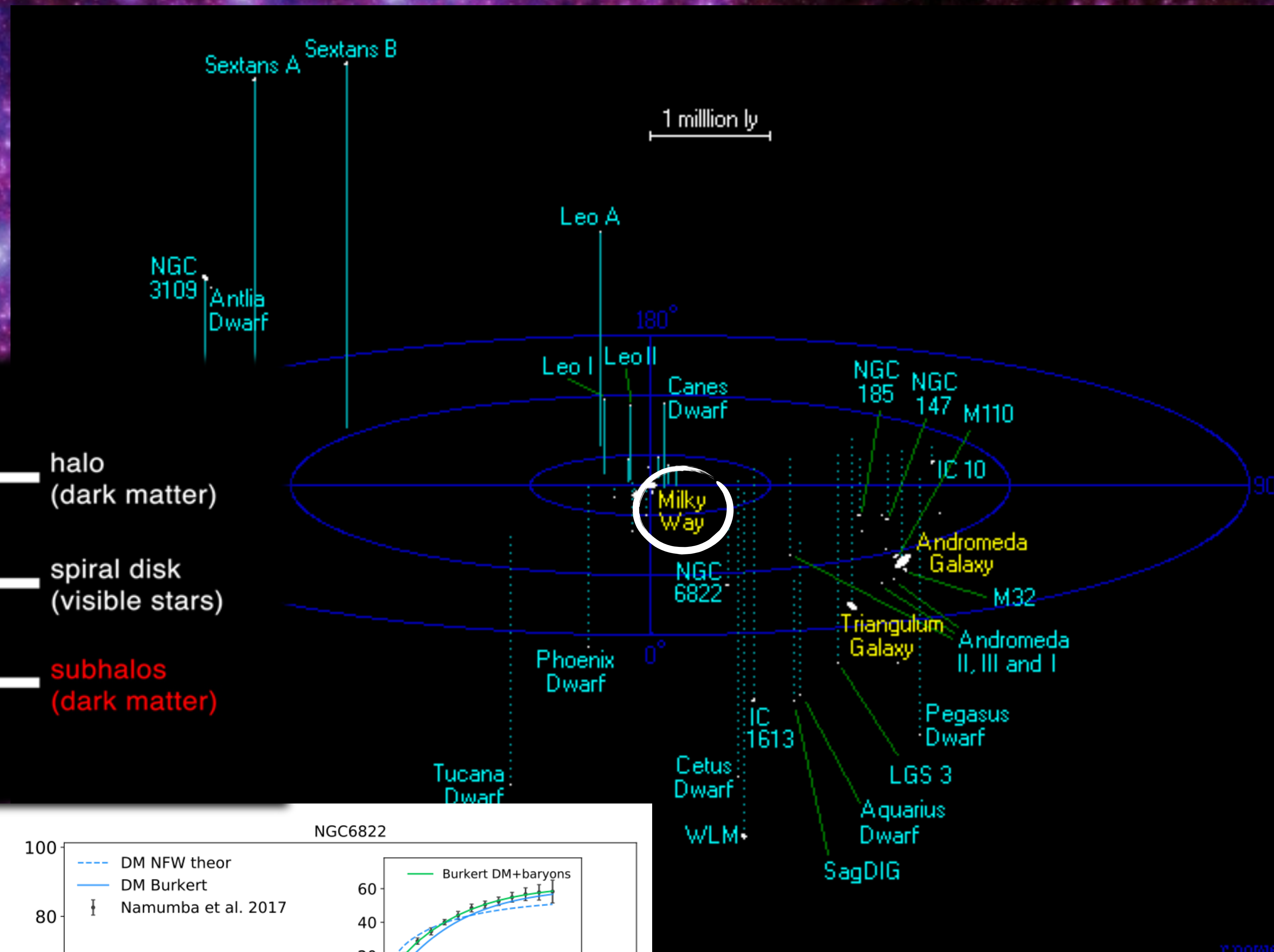
- FERMI-LAT SATELLITE, GAMMA-RAY DATA & BETA-PLOT
- DARK MATTER & BETA-PLOT
- SYSTEMATIC FEATURES
- CLASSIFICATION ALGORITHMS AND SETUPS
- RESULTS

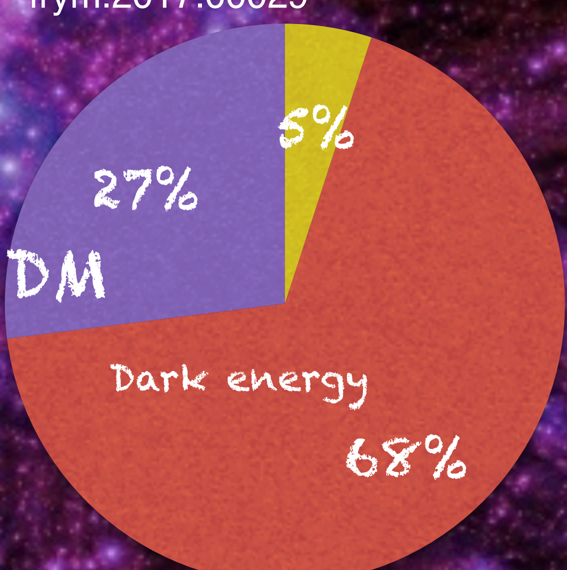
DARK MATTER

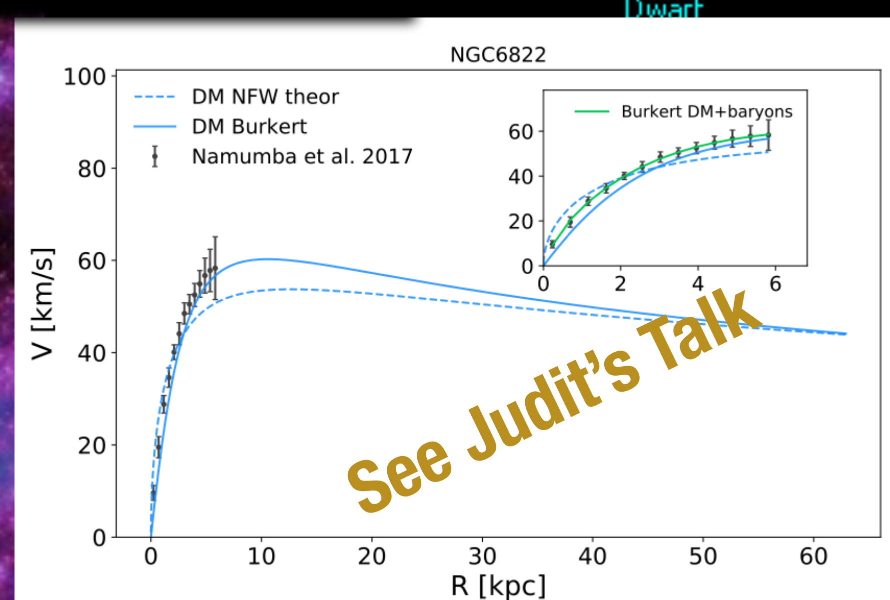


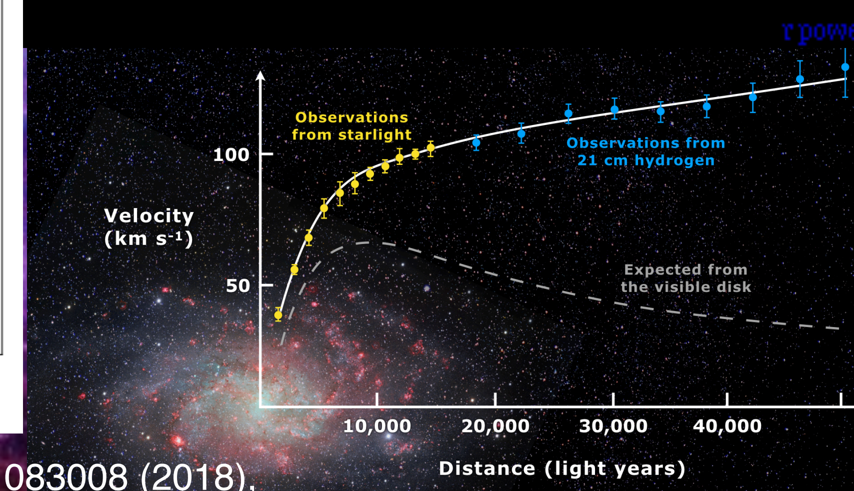


Ngan W (2017) Using Bright Streams to Learn about Dark Matter. Front. Young Minds. 5:29. doi: 10.3389/frym.2017.00029





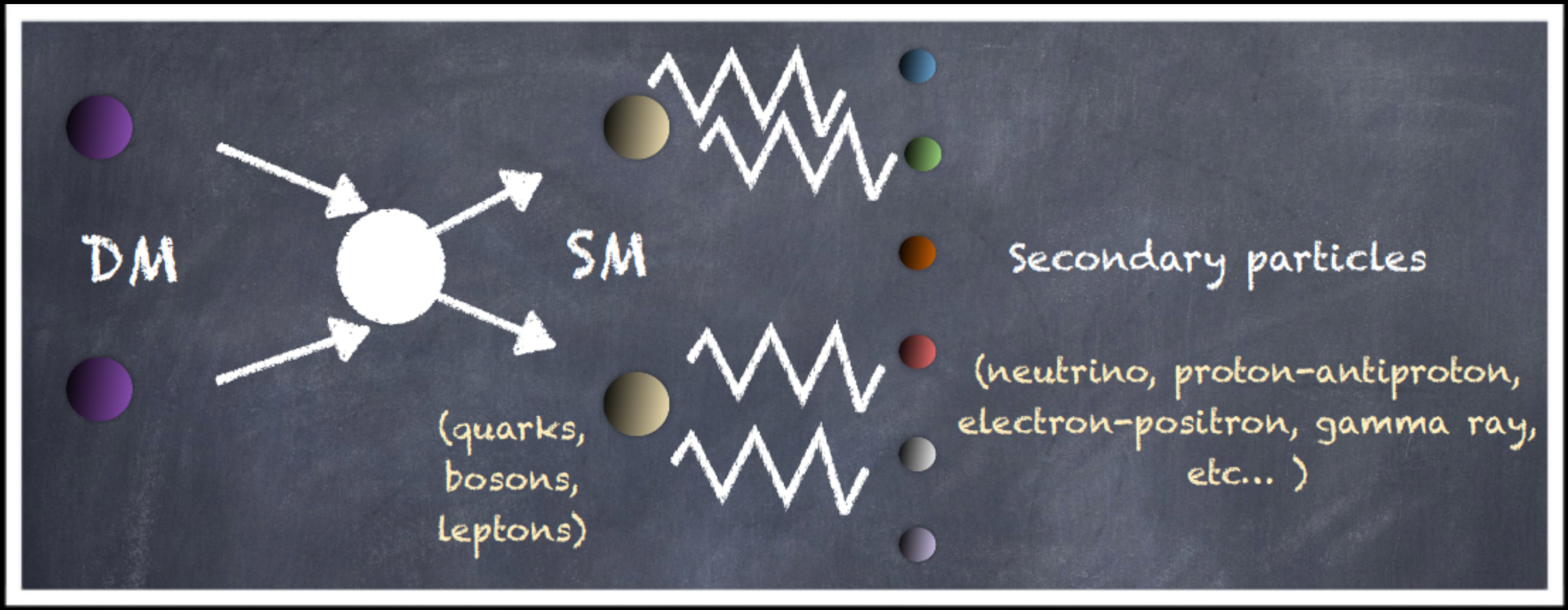




V. Gammaldi et al. PRD 98, 083008 (2018),
PRD 104, 083026 (2021)

DARK MATTER & BETA-PLOT

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

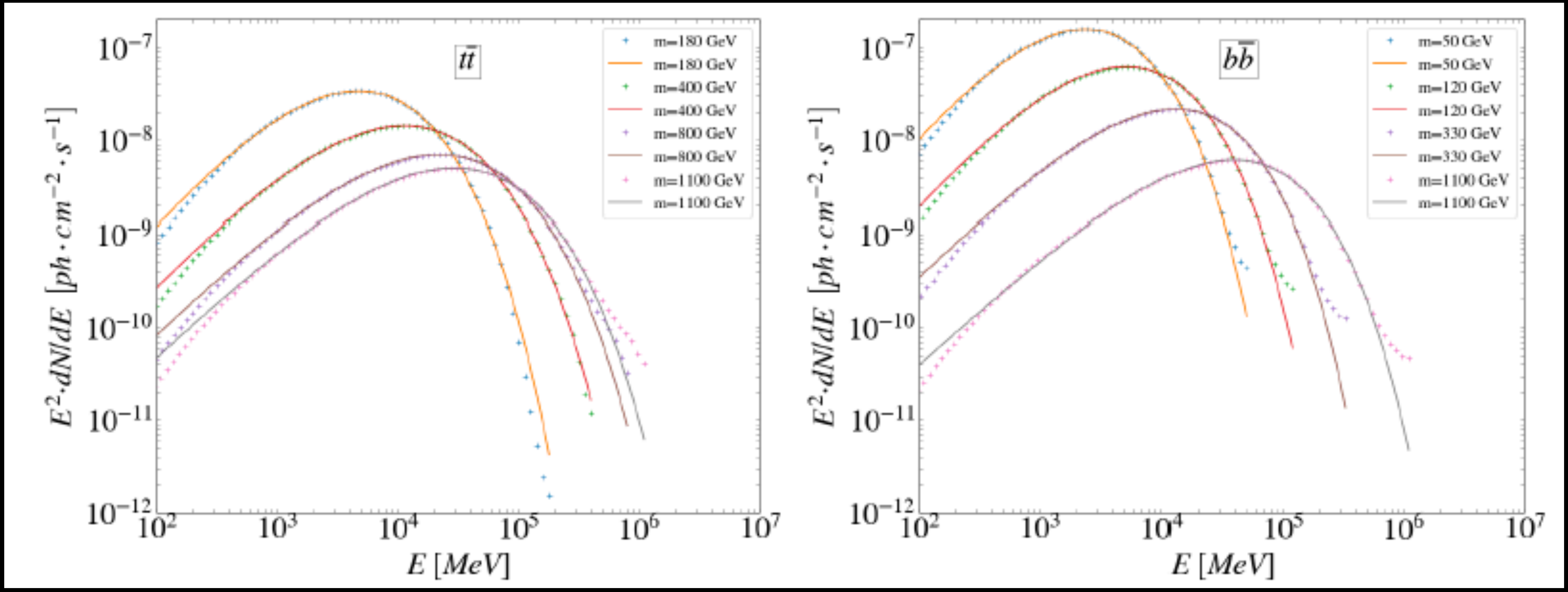


$$\frac{d\phi_\gamma}{dE}(E, \Delta\Omega, l\,.\,o\,.\,s) = \underbrace{\frac{1}{4\pi}\frac{\langle\sigma v\rangle}{\delta\, m_\chi^2}\frac{dN_\gamma}{dE}(E)}_{\text{Particle Physics}} \times \underbrace{J(\Delta\Omega, l\,.\,o\,.\,s)}_{\text{Astrophysics}}$$

DARK MATTER & BETA-PLOT

The gamma-ray flux expected by WIMP annihilation into Standard Model (SM) channels can be simulated via Monte Carlo event generator software (e.g. Pythia 8). The very well known PPPCB4DMID (Cirelli's) interpolation of those fluxes (for several WIMP masses and SM channels) makes those results user friendly for DM searches.

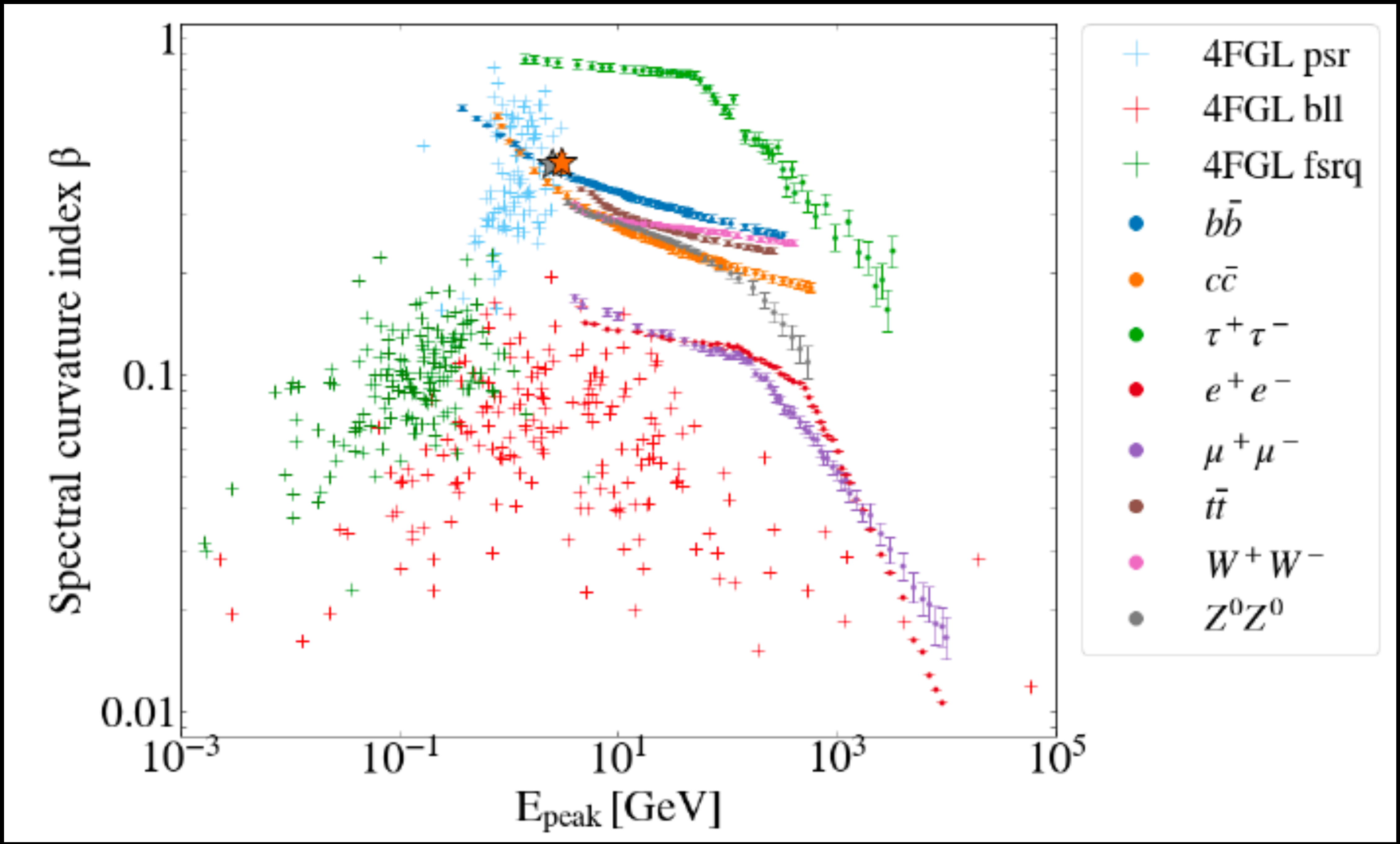
Here, we further fit those interpolations with a Log-Parabola (LP):



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 & BETA-PLOT

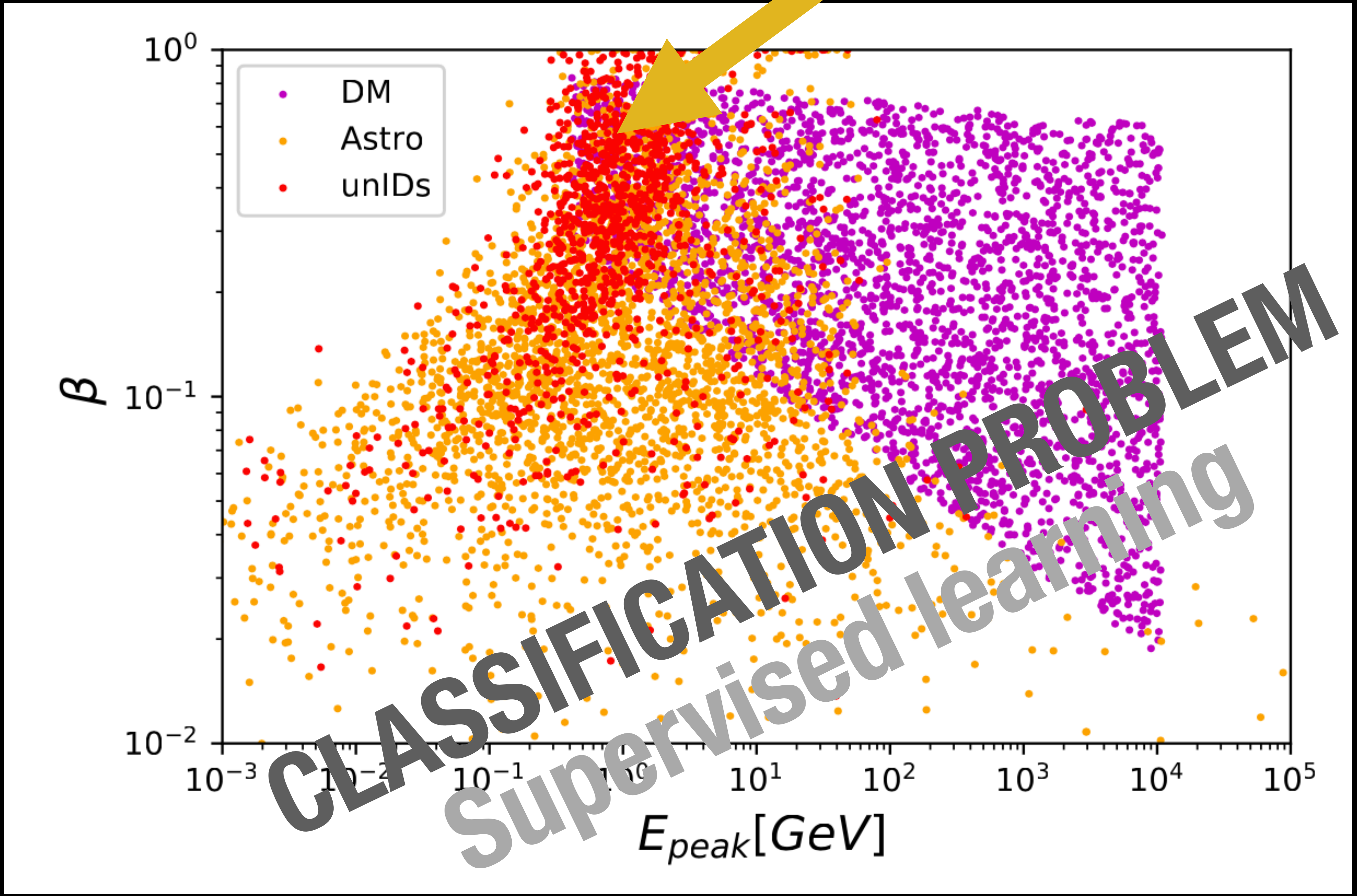


J. Coronado-Blázquez et al., JCAP11(2019)045

DARK MATTER & BETA-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



This work, V.G. et al. MNRAS 520, 1348–1361 (2023)

OUR STRATEGY

1. Training the classification algorithm on a sample of 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}.

- So far, the classification problem is based on two features (E_{peak}, β) .

- We expect the classification accuracy improves by including more independent features. Intuitively straightforward, but many observational features are not available for DM.

OUTLINE

- FERMI-LAT SATELLITE, GAMMA-RAY DATA & BETA-PLOT
- DARK MATTER & BETA-PLOT
- SYSTEMATIC FEATURES
- CLASSIFICATION ALGORITHMS AND SETUPS
- RESULTS

SYSTEMATIC FEATURES

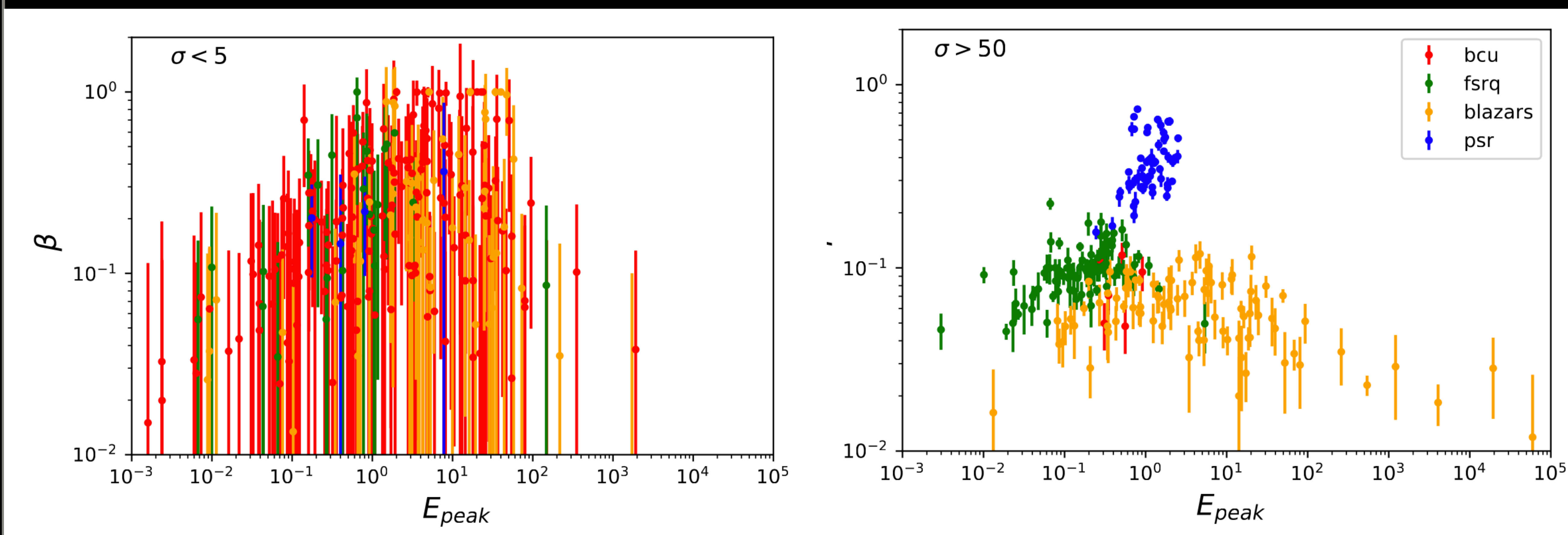
Indeed, we focused on two observational features:

1. Detection significance σ_d: obtained by the likelihood analysis, which determines if the source exists, its position, spectral parameters, etc. It depends e.g. on the background template (the diffuse model and emission model of all the other sources in the source region) among other systematics we are not able to model.

2. Uncertainty on the curvature β: we expect that a lower detection significance corresponds to a worst characterisation of the source spectrum, but this is not all….

SYSTEMATIC FEATURES: DETECTION SIGNIFICANCE

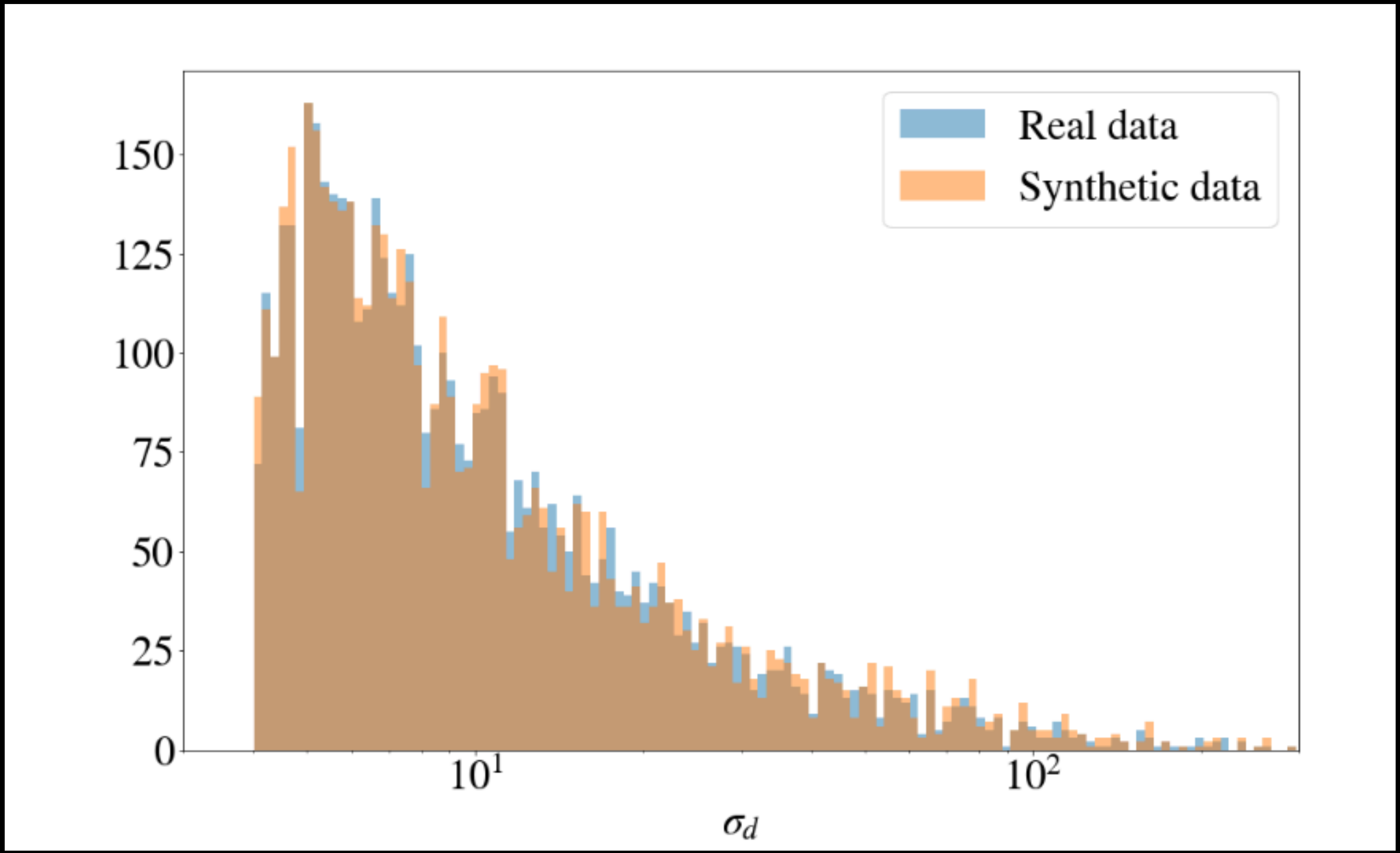
1. Detection significance σ_d: a source pre-selection based on detection significance improves the classification even only "by eye".



This work, V.G. et al. MNRAS 520, 1348–1361 (2023)

SYSTEMATIC FEATURES: DETECTION SIGNIFICANCE

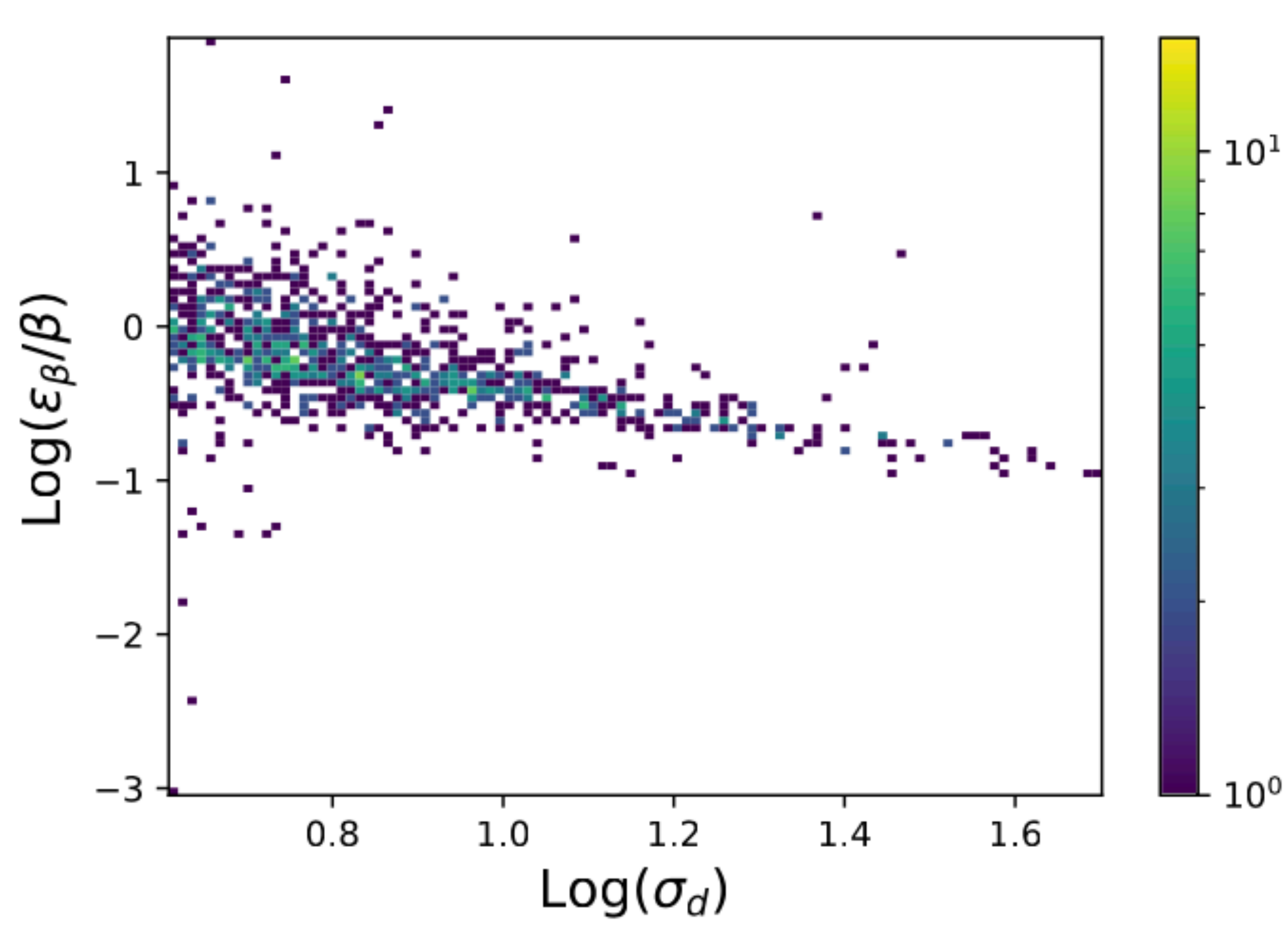
1. -> If any DM source has been detected so far, it is among the unIDs. We assume that all the DM candidates are unIDs. Indeed, we use for the DM sample the same distribution of detection significance as the unIDs.



This work, V.G. et al. MNRAS 520, 1348–1361 (2023)

SYSTEMATIC FEATURES: UNCERTAINTY ON β

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

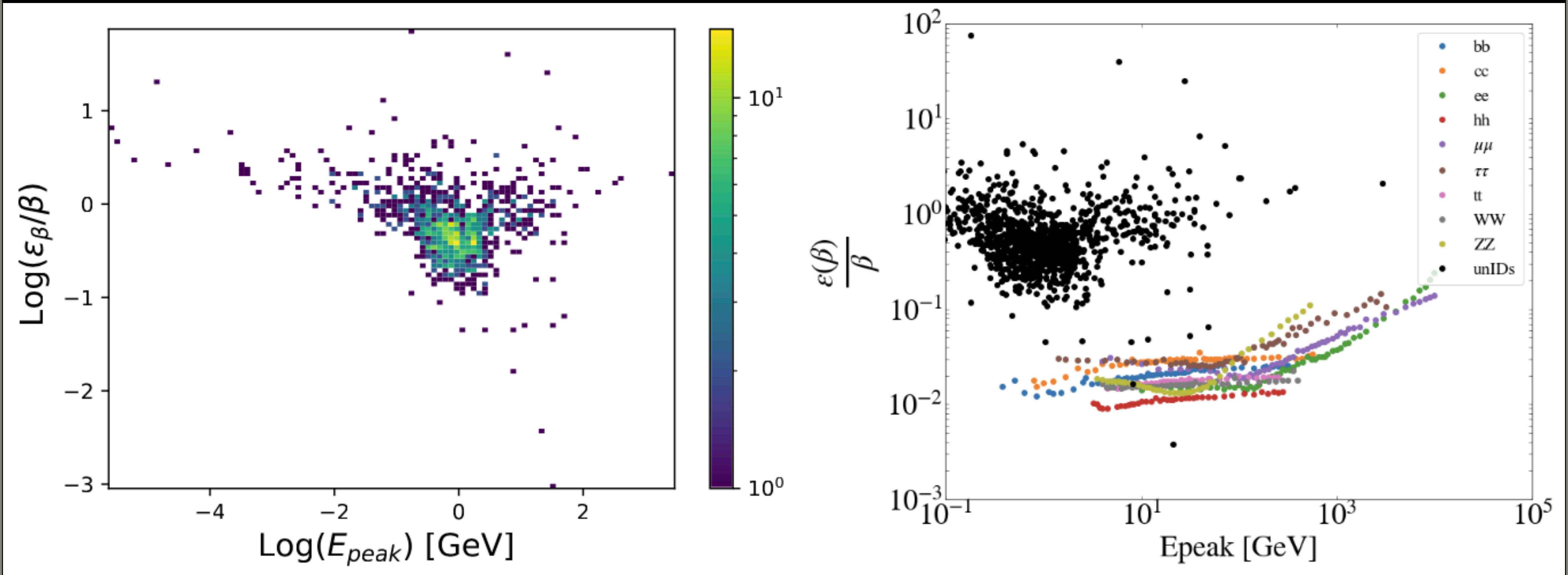


This work, V.G. et al. MNRAS 520, 1348–1361 (2023)

but this is not all….

SYSTEMATIC FEATURES: UNCERTAINTY ON β

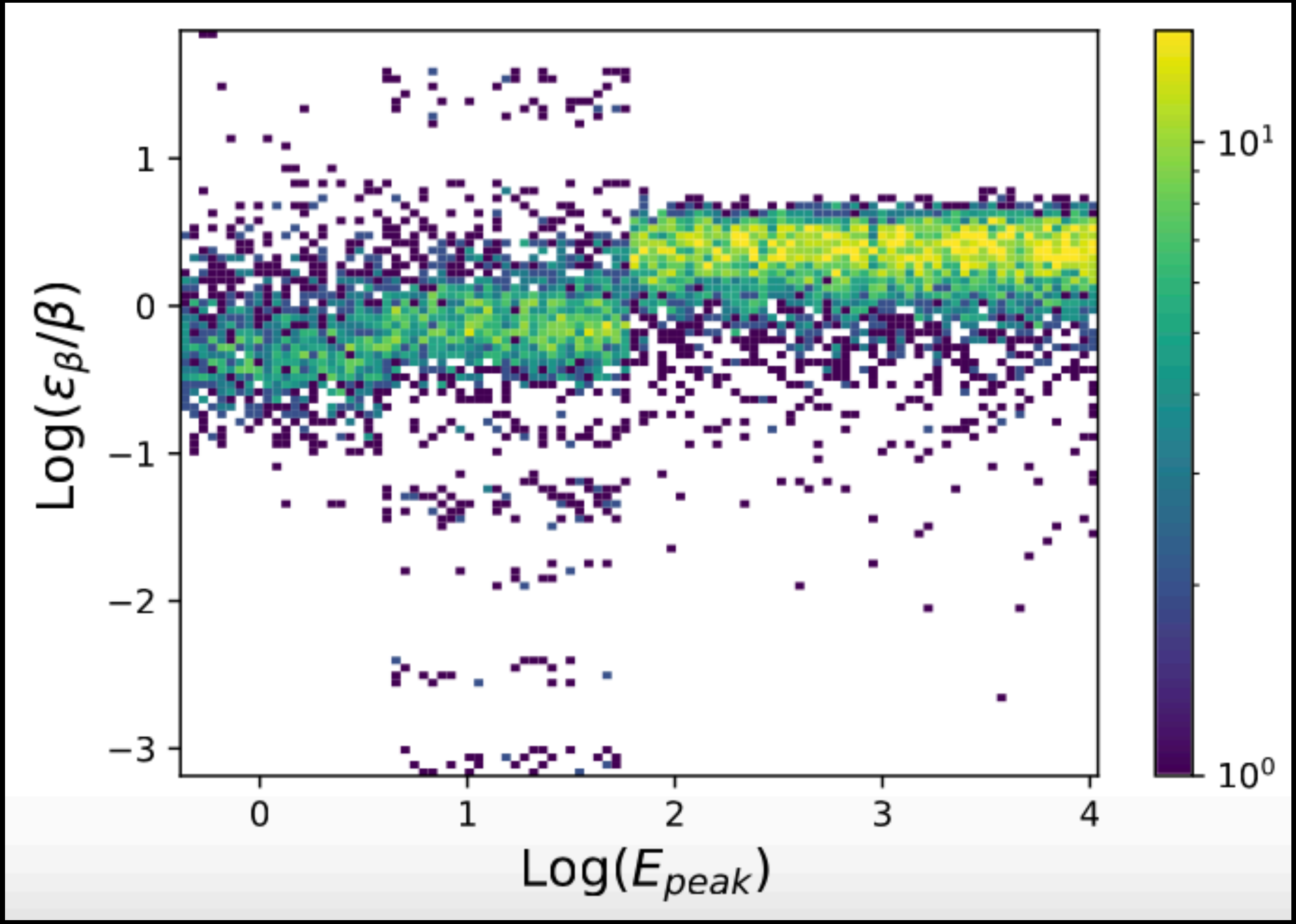
2. Uncertainty on β: we expect that a lower detection significance corresponds to a worst characterisation of the source spectrum, but the latter also depends on the E_{peak}



This work, V.G. et al. MNRAS 520, 1348–1361 (2023)

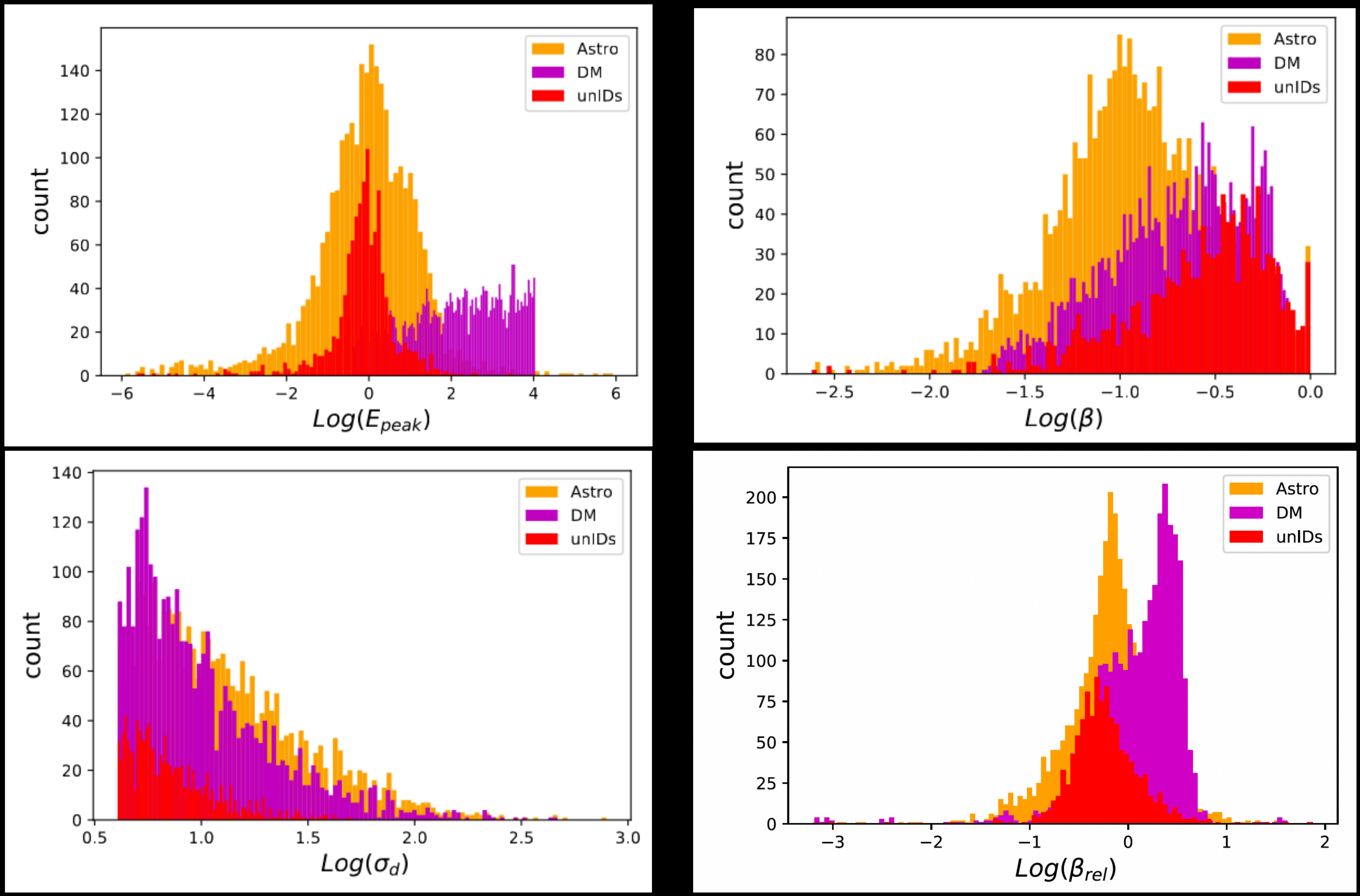
SYSTEMATIC FEATURES: UNCERTAINTY ON β

2. -> we use for the DM sample a distribution of $\beta_{rel} = \epsilon_\beta / \beta$ which statistically depends on the E_{peak}, in agreement with the observed unIDs population



This work, V.G. et al. MNRAS 520, 1348–1361 (2023)

FROM TWO TO FOUR FEATURES



This work, V.G. et al. MNRAS 520, 1348–1361 (2023)

OUTLINE

- FERMI-LAT SATELLITE, GAMMA-RAY DATA & BETA-PLOT
- DARK MATTER & BETA-PLOT
- SYSTEMATIC FEATURES
- CLASSIFICATION ALGORITHMS AND SETUPS
- RESULTS

CLASSIFICATION ALGORITHMS

- **LOGISTIC REGRESSION** (LR) (SCIKIT-LEARN)
- **ARTIFICIAL NEURAL NETWORK** (NN) (SCIKIT-LEARN)
- **NAIVES BAYES** (NB) (PYTHON)
- **GAUSSIAN PROCESS** (GP) (TENSOR FLOW)

CLASSIFICATION ALGORITHMS

LINEAR REGRESSION

Best fit: getting the θ^i parameters which minimize the (cost/lost) function (least square, likelihood, etc.)

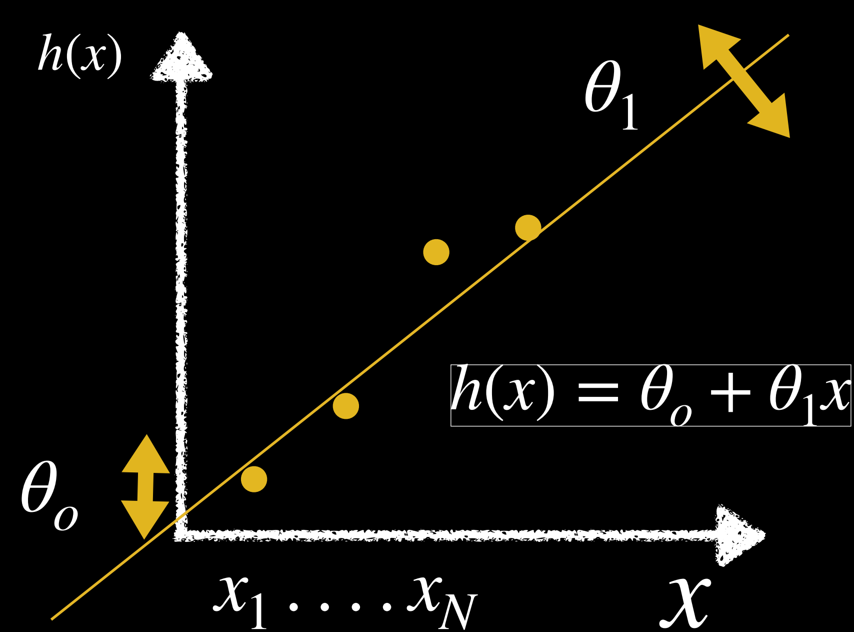
$$J(\Theta) = \frac{1}{2}(h(x) - Y)^2 \equiv \frac{1}{2n}\sum_{i=1}^{n}((\Theta^T \mathbf{X})_i - Y_i^2)$$

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

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

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

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



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

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

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

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

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

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

$$J(\Theta) = J(\theta_0^i, \theta_1^i, \ldots \theta_f^i)_{i=1..n}$$

CLASSIFICATION ALGORITHMS

LOGISTIC REGRESSION

Getting the θ^i parameters which minimize the cost/lost function

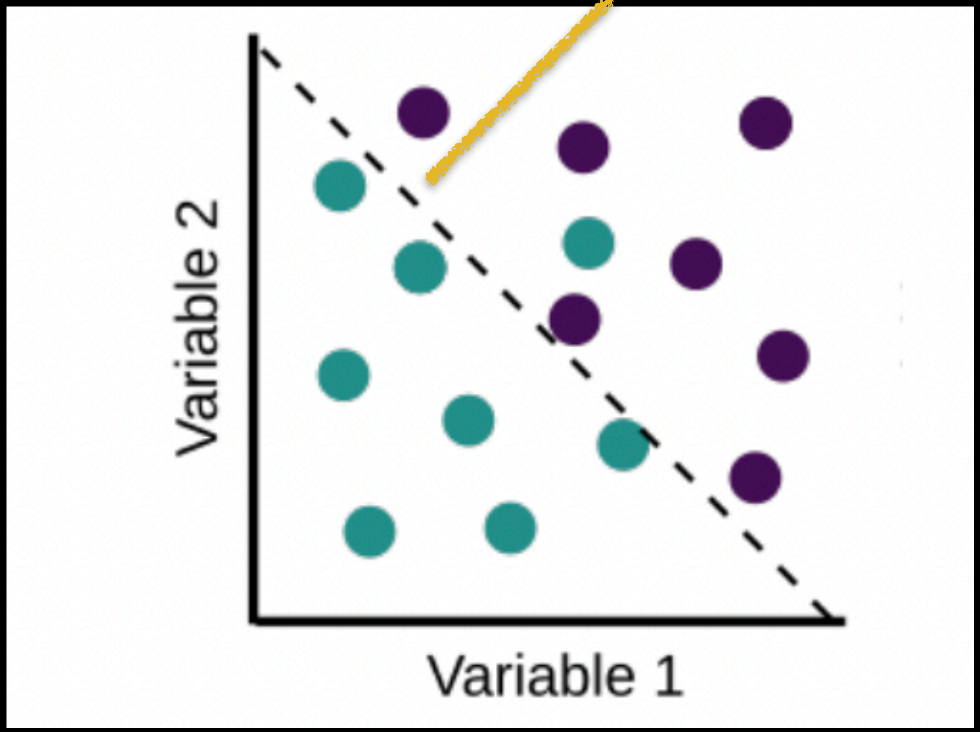
$$J(\Theta) = -\frac{1}{n}[\sum_{i=1}^{n} y^{(i)} \log(h_\theta(x^{(i)})) + (1-y^{(i)})\log(1-h_\theta(x^i))]$$

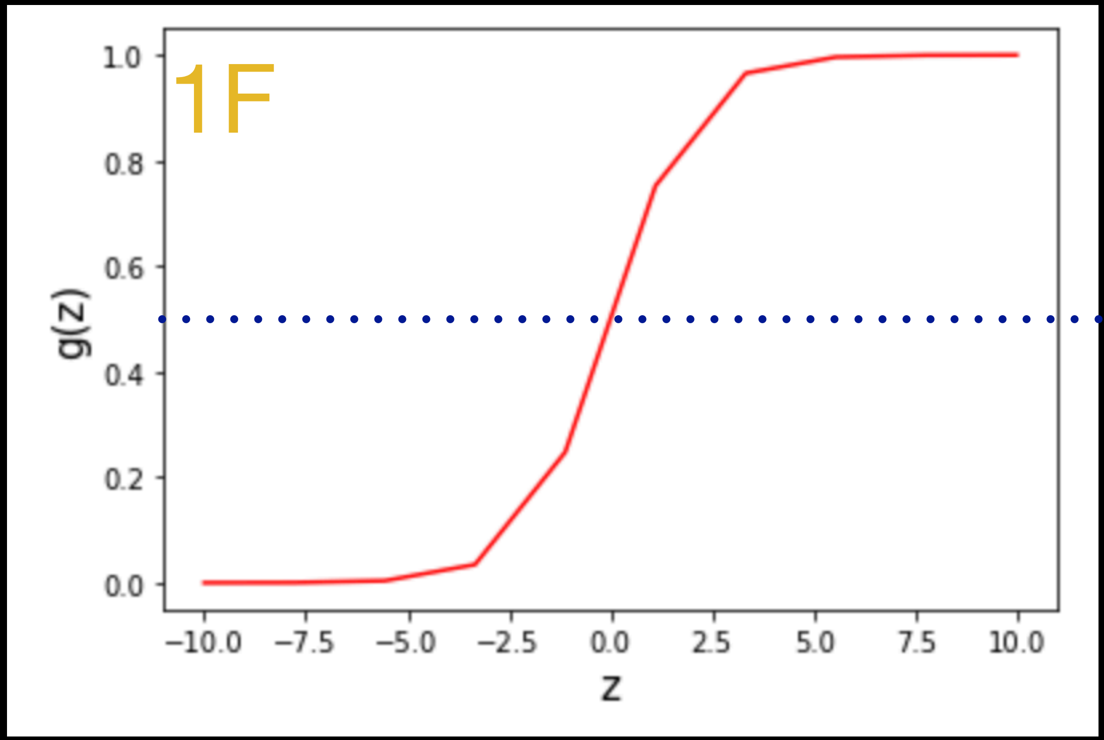
$$h(x) \rightarrow g(z) = \frac{1}{1+e^{-z}} \quad z = \theta_o + \theta_1 x + \ldots (+\ \theta_2 x^2 \ldots)$$

Activation function

Decision boundary

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



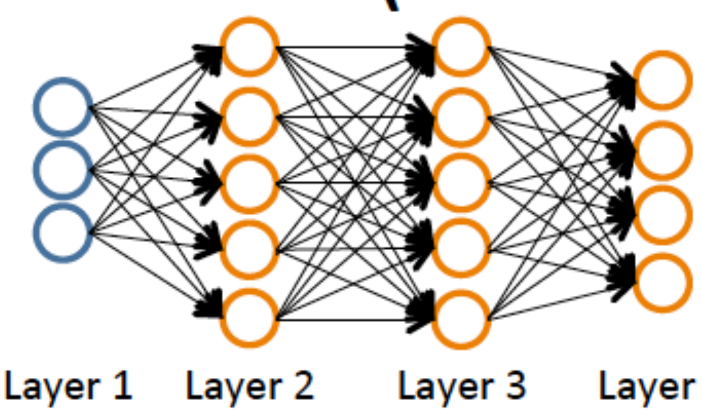


$g(z) \leq 0.5 \rightarrow y^{(i)} = 0$ (e.g. Astro)

CLASSIFICATION ALGORITHMS

ARTIFICIAL 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

$y = 0 \text{ or } 1$

1 output unit

Multi-class classification (K classes)

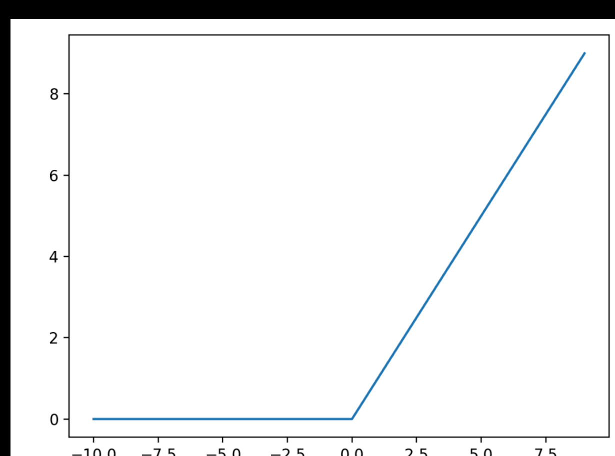
$y \in \mathbb{R}^K$ E.g. $\begin{bmatrix}1\\0\\0\\0\end{bmatrix}$, $\begin{bmatrix}0\\1\\0\\0\end{bmatrix}$, $\begin{bmatrix}0\\0\\1\\0\end{bmatrix}$, $\begin{bmatrix}0\\0\\0\\1\end{bmatrix}$

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.

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

CLASSIFICATION ALGORITHMS

ARTIFICIAL NEURAL NETWORK

LOGISTIC REGRESSION: COST FUNCTION

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

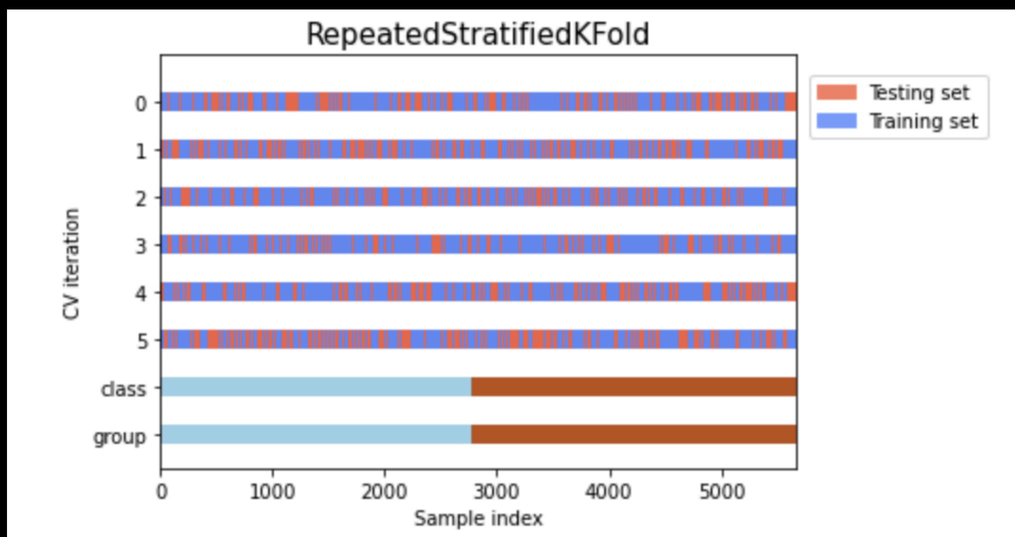
NEURAL NETWORK: COST FUNCTION

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

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

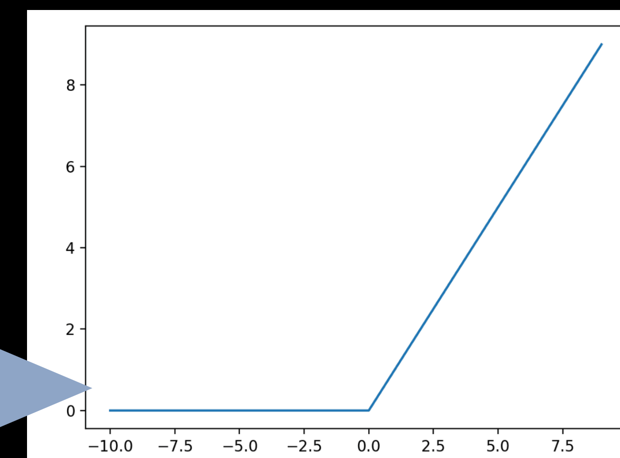
CLASSIFICATION ALGORITHMS

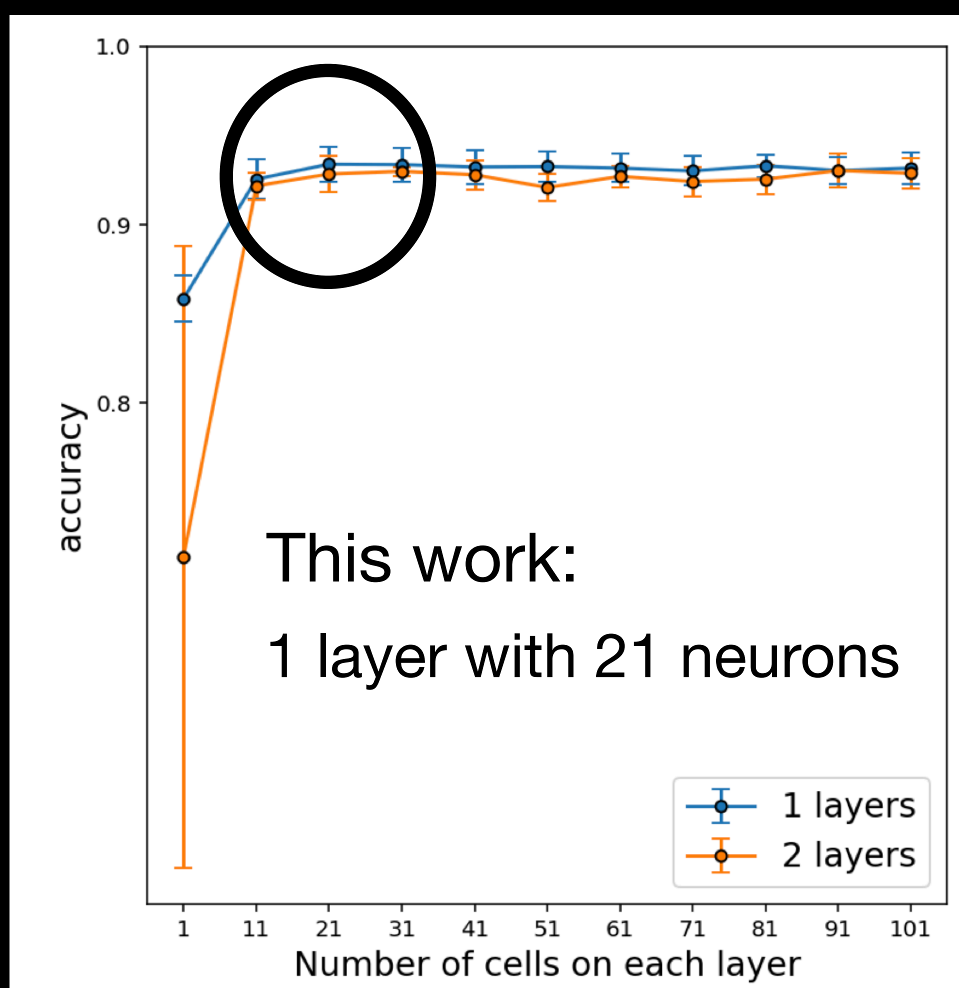
ARTIFICIAL NEURAL NETWORK: THIS WORK

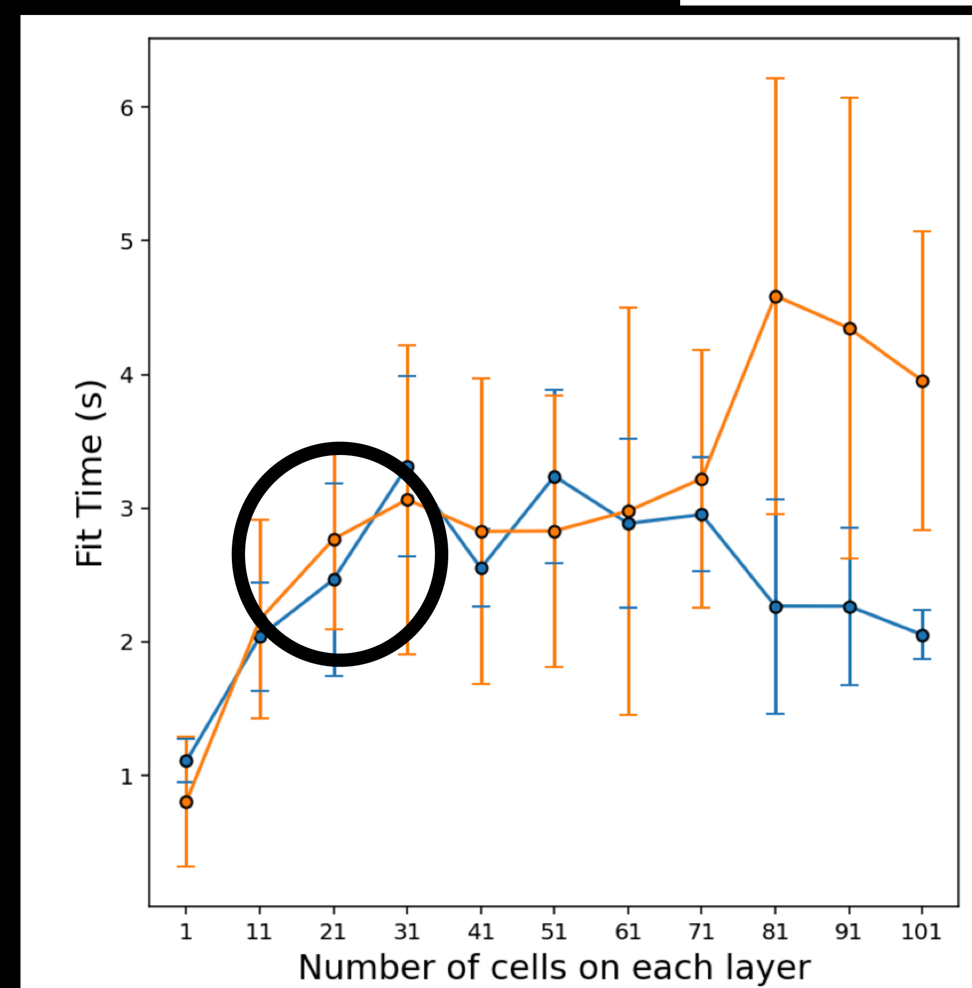


Activation function:

Rectified Linear Activation Function (ReLu)







SETUPS

- 2-FEATURES (2F) CLASSIFICATION (LR, NN, NB): Includes the 2-features introduced so far, indeed (E_{peak}, β)

- 3-FEATURES AUGMENTED (3F-A) (LR, NN, NB): An augmented dataset containing three features: $(E_{\text{peak}}, \beta_{\text{sampled}}, \sigma_d)$ Instead of incorporating the uncertainty β_{rel} as an extra feature, the strategy here is to augment the dataset by the following procedure: For each observation, we assume that the variable β follows a truncated Gaussian distribution, whose mean is precisely the observed value, and the standard deviation is precisely the observed uncertainty ϵ_β, but truncated such that $0 < \beta \leq 1$.

- 3F-B (GP): A dataset containing the three same features as above, i.e. $(E_{\text{peak}}, \beta, \sigma_d)$. However, now the uncertainties ϵ_β are included in the statistical model. Concretely, this setup will concern exclusively the NIMGP model mentioned above.

- 4-FEATURES (4F) CLASSIFICATION (LR, NN, NB): Includes the systematics uncertainty, by including two more features, that are: $(E_{\text{peak}}, \beta, \sigma_d, \beta_{\text{rel}})$ where $\beta_{\text{rel}} = \epsilon_\beta / \beta$

OUTLINE

- FERMI-LAT SATELLITE, GAMMA-RAY DATA & BETA-PLOT
- DARK MATTER & BETA-PLOT
- SYSTEMATIC FEATURES
- CLASSIFICATION ALGORITHMS AND SETUPS
- RESULTS

RESULTS

V.G. et al. arXiv: 2207.09307

OVERALL ACCURACY (OA): PERCENTAGE OF WELL CLASSIFIED DATA SET

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

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)

THE OVERALL CLASSIFICATION ACCURACY IMPROVES WITH THE INTRODUCTION OF SYSTEMATIC FEATURES FOR ALL THE TRAINED ALGORITHMS.

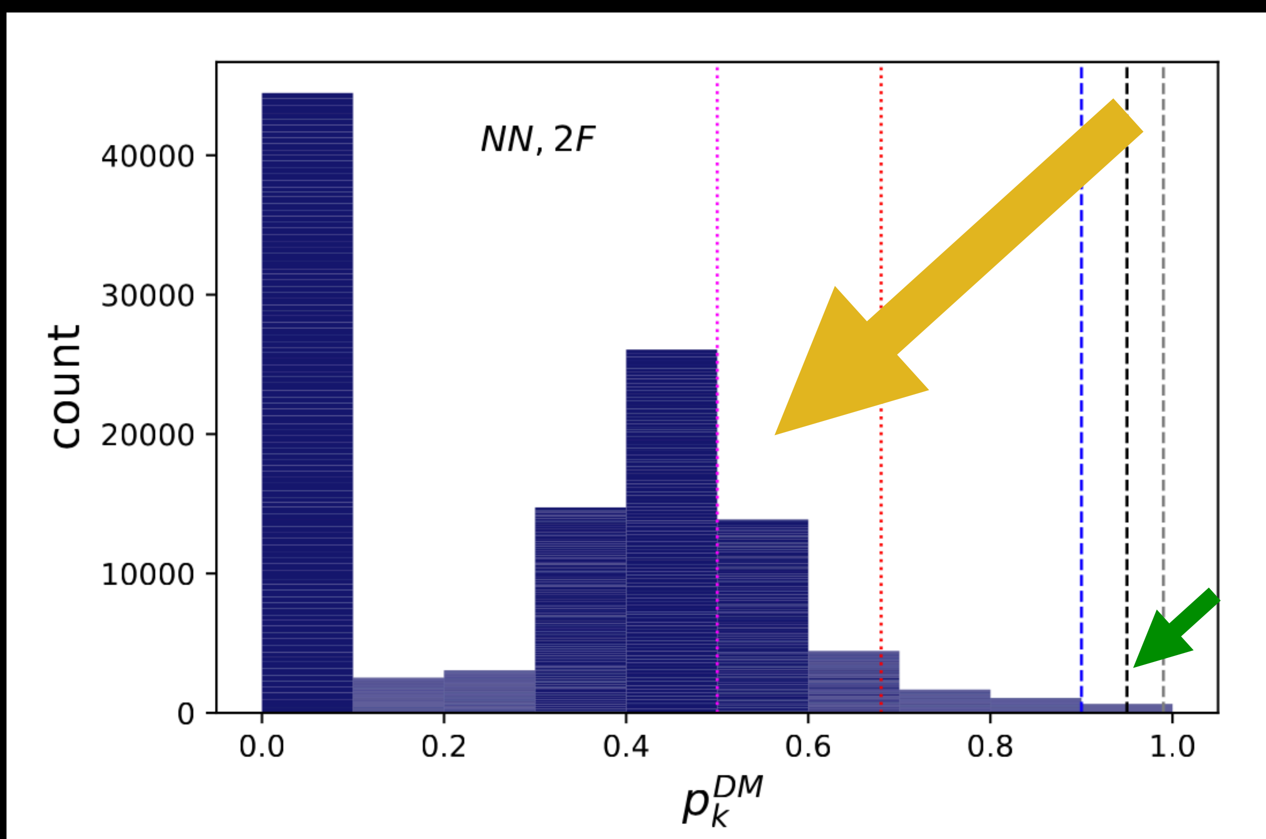
	OA(%)	TN (%)	TP (%)
LR			
2F	84.9 ± 0.8	85.4 ± 1.5	84.4 ± 1.4
3F-A	83.0 ± 0.1	85.0 ± 0.2	81.0 ± .0.2
4F	86.0 ± 0.9	86.7 ± 1.5	85.2 ± 1.3
NN			
2F	86.2 ± 0.8	86.1 ± 3.0	86.4 ± 3.4
3F-A	85.0 ± 0.2	87.9 ± 1.8	82.3 ± .1.8
4F	**93.3 ± 0.7**	**94.7 ± 1.7**	**91.8 ± 1.5**
NB			
2F	82.4 ± 1.5	83.9 ± 1.9	80.5 ± 2.5
3F-A	82.5 ± 0.3	83.7 ± 0.4	81.6 ± 0.3
4F	83.5 ± 1.0	86.2 ± 1.2	81.7 ± 1.2
GP			
3F-B	88.1 ± 0.2	89.6 ± 0.3	84.9 ± 0.2

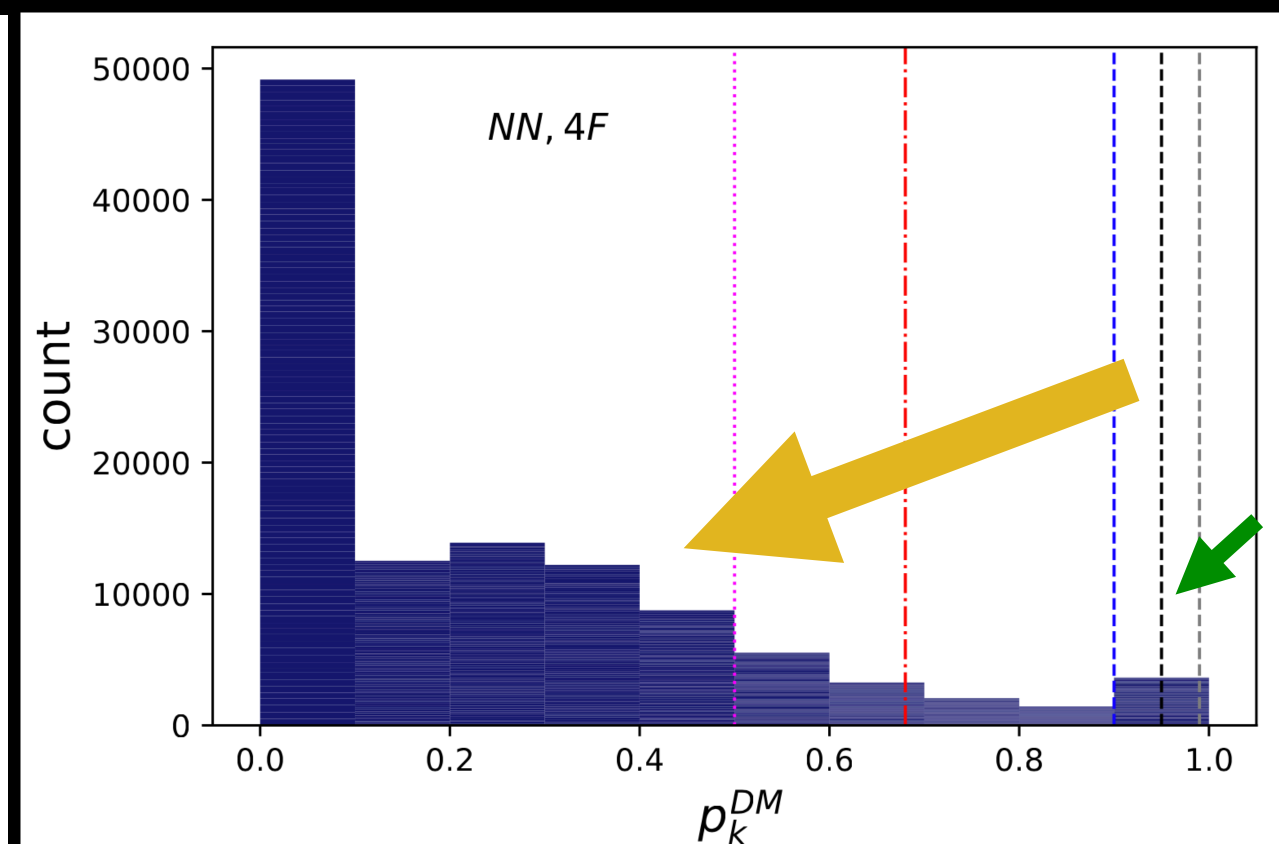
UNIDS CLASSIFICATION WITH NN

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

2-FEATURES (2F) CLASSIFICATION

4-FEATURES (4F) CLASSIFICATION





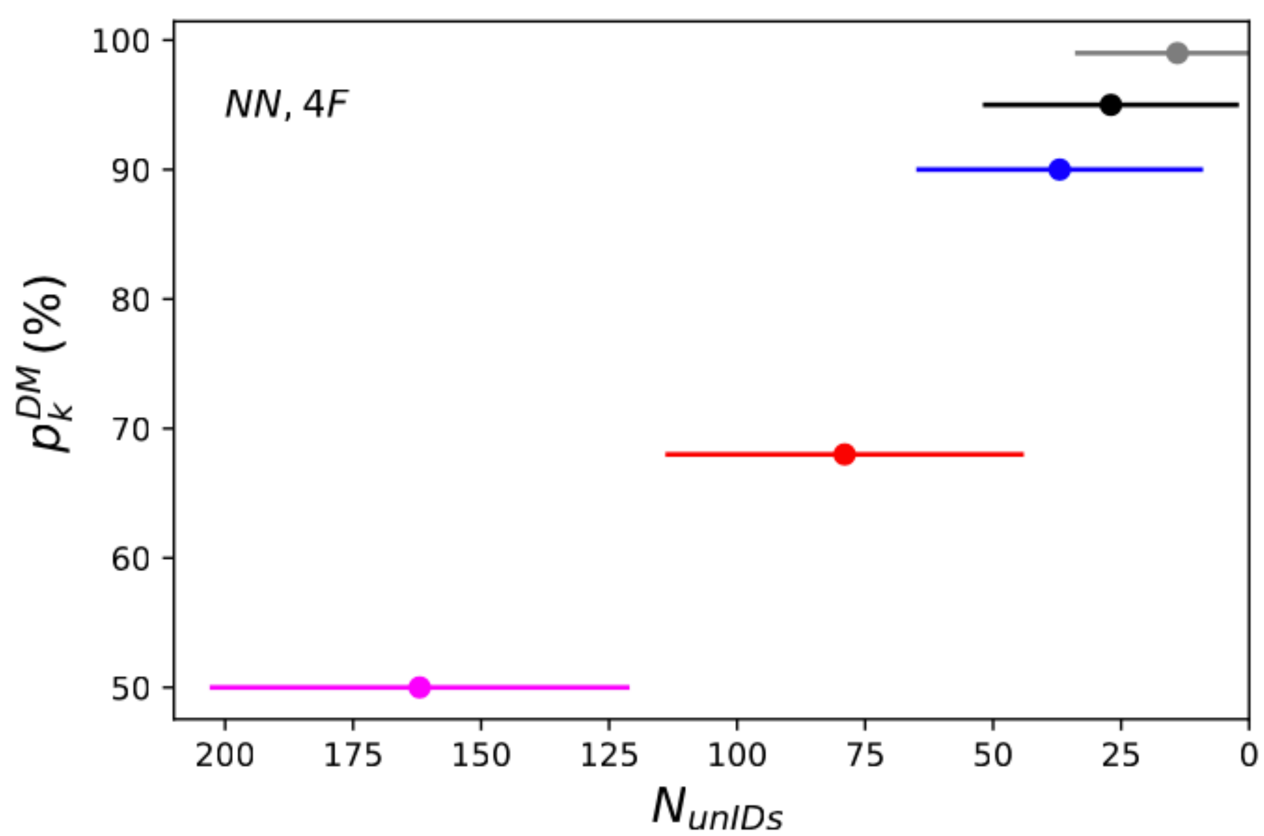
This work, V.G. et al. MNRAS 520, 1348–1361 (2023)

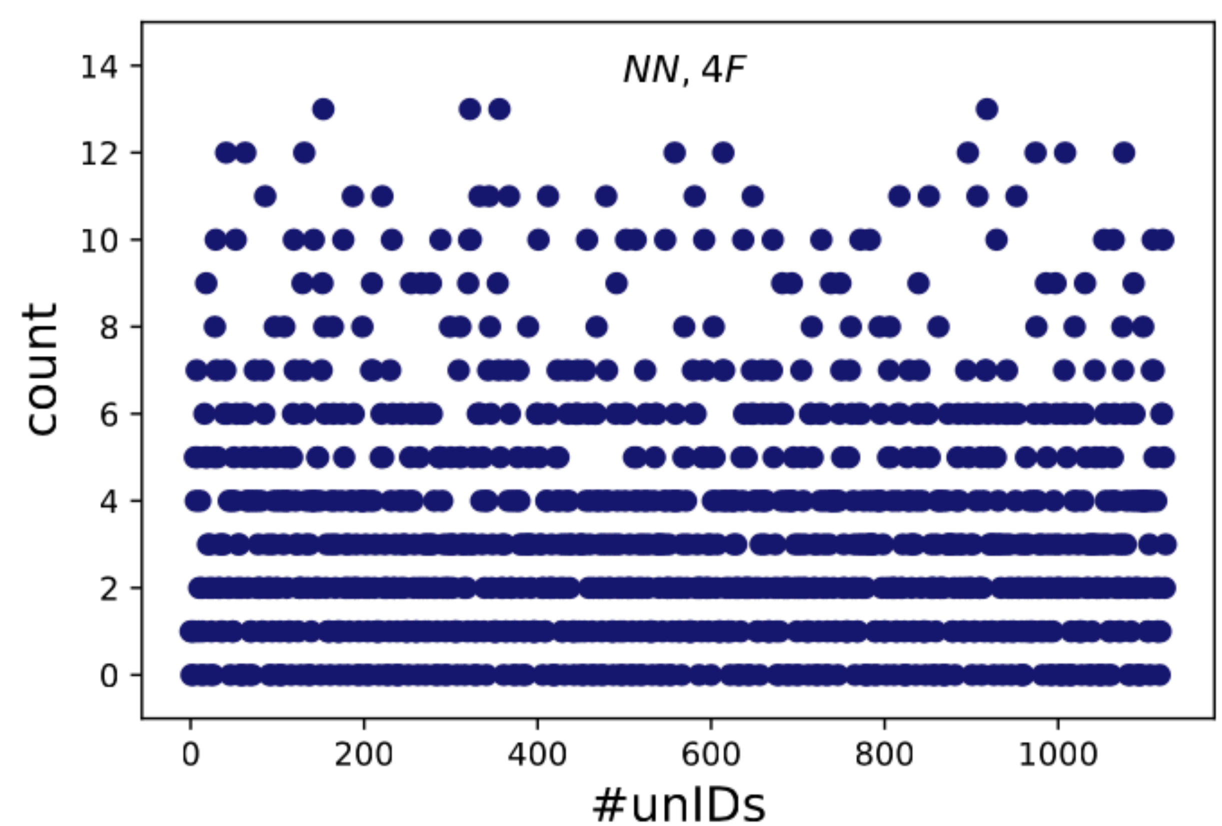
<u>The degeneracy is partially solved</u>

UNIDS CLASSIFICATION WITH NN-4F

Mean number of unIDs with $p_k^{DM} > 0.50, 0.68, 0.90, 0.99$ in the NN-4F classification. The error bars are calculated as the standard deviation on 100 classifications

The best candidates in the run have been classified with $p_k > 90\%$ 13 times over 100 classifications.





This work, V.G. et al. MNRAS 520, 1348–1361 (2023)

Due to statistical fluctuations,
we cannot claim for any robust DM candidate.

CONCLUSIONS

- WE WORK ON A DERIVED PARAMETER SPACE, DEFINED BY THE OBSERVATIONAL SPECTRAL FEATURES E_{peak} AND β. IN THE SAME SPACE, WE INTRODUCE THE WIMP CANDIDATES, BASED ON THEORETICAL EXPECTATIONS.
- WE TRAIN THE CLASSIFICATION ALGORITHM ON THE SAMPLE OF EXPERIMENTAL AND EXPECTED DATA. WE ASK TO THE ALGORITHM TO CLASSIFY UNIDENTIFIED SOURCES.
- WHITIN THE 2-FEATURES (E_{peak}, β) SET UP, THE DEGENERACY BETWEEN ASTRO AND DM SOURCES IS VISIBLE.
- WE INCLUDE SYSTEMATIC UNCERTAINTY (σ_d, β_{rel}) IN CLASSIFICATION PROBLEMS, IMPROVING THE OVERALL CLASSIFICATION ACCURACY FOR ALL THE ALGORITHMS.
- THE DEGENERACY OF ASTROPHYSICAL AND DARK MATTER SIGNAL IS PARTIALLY SOLVED.
- THE RESULTS ARE IN STATISTICAL AGREEMENT WITHIN DIFFERENT RANDOM SEEDS.
- NO ROBUST DARK MATTER CANDIDATES HAVE BEEN FOUND AMONG THE FERMI-LAT UNIDENTIFIED SOURCES.

WHAT'S THE NEXT?

- THIS IS ONLY THE FIRST ATTEMPT FOR APPLYING MACHINE LEARNING TO DM SEARCHES.
- WE CAN IMPROVE THESE RESULTS BY INCLUDING NEW FEATURES OR IMPLEMENTING DIFFERENT HYPOTHESIS AND APPROXIMATIONS, ETC.
- WE CAN IMPROVE THESE RESULTS BY APPLYING DIFFERENT MACHINE LEARNING TECHNIQUES.
- SAME METHODOLOGY MAY BE APPLIED TO SIMILAR DATA SETS.

THANK YOU
FOR YOUR ATTENTION

BACK-UP SLIDES

A SELECTION OF PREVIOUS WORKS

Previous works developed as machine learning application to the unidentified sources of the Fermi-LAT catalogues - focused on identifying astrophysical sources. See e.g.:

- "Artificial Neural Network Classification of 4FGL Sources." S. Germani et al., MNRAS (2021);
- "Machine Learning application to Fermi-LAT data: sharpening all-sky map and emphasizing variable sources." S. Sato et al., Astrophys.J. 913 (2021);
- "Searches for Pulsar-like candidates from unidentified objects in the third catalog of hard Frmi-LAT (3FGHL) sources with machine learning techniques." C.Y. Hui et al. MNRAS (2020);
- "3FGLzoo. Classifying 3FGL Unassociated Fermi-LAT Gamma-ray Sources by Artificial Neural Networks." David Salvetti et al., MNRAS (2017);

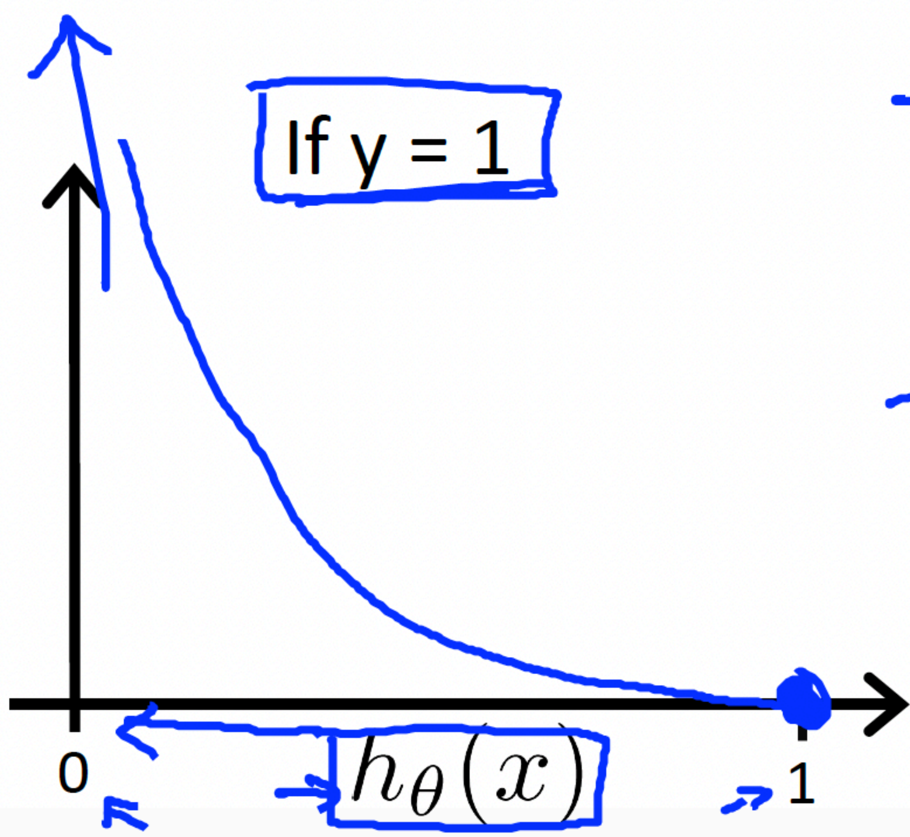
Previous works developed as dark matter searches among unidentified sources of the Fermi-LAT catalogues - adopt benchmark analyses (without machine learning. See e.g.:

- "Spectral and spatial analysis of the dark matter sub halo candidates among Fermi Large Area Telescope unidentified sources." J. Coronado-Blázquez et al., JCAP11(2019)045.
- "Unidentified gamma-ray sources as targets for indirect dark matter detection with the Fermi Large Area Telescope." J. Coronado-Blázquez et al., JCAP07(2019)020

Logistic Regression: minimization

Logistic regression cost function

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



- Cost $= 0$ if $y = 1, h_\theta(x) = 1$
  But as $h_\theta(x) \rightarrow 0$
  $Cost \rightarrow \infty$

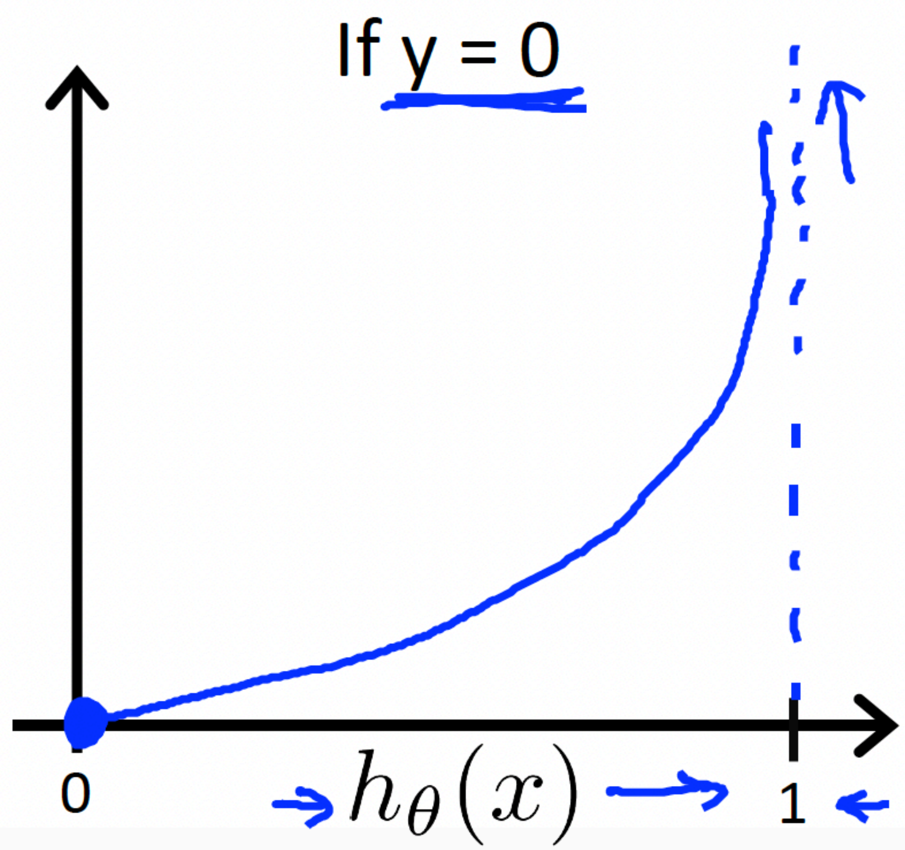
- Captures intuition that if $h_\theta(x) = 0$, (predict $P(y = 1|x; \theta) = 0$), but $y = 1$, we'll penalize learning algorithm by a very large cost.

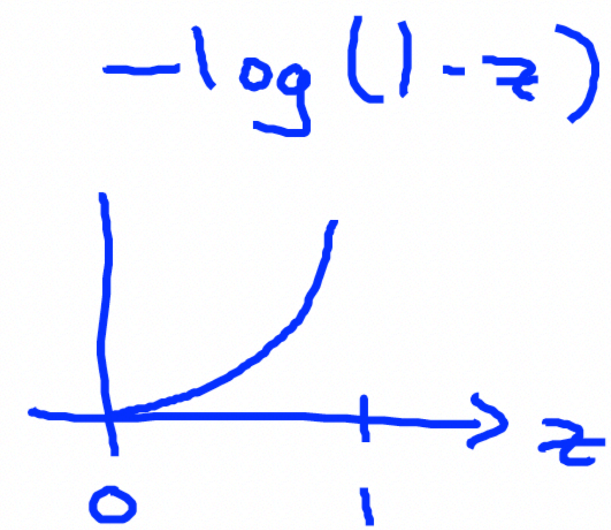
Coursera by Stanford - Andrew Ng

Logistic Regression: minimization

Logistic regression cost function

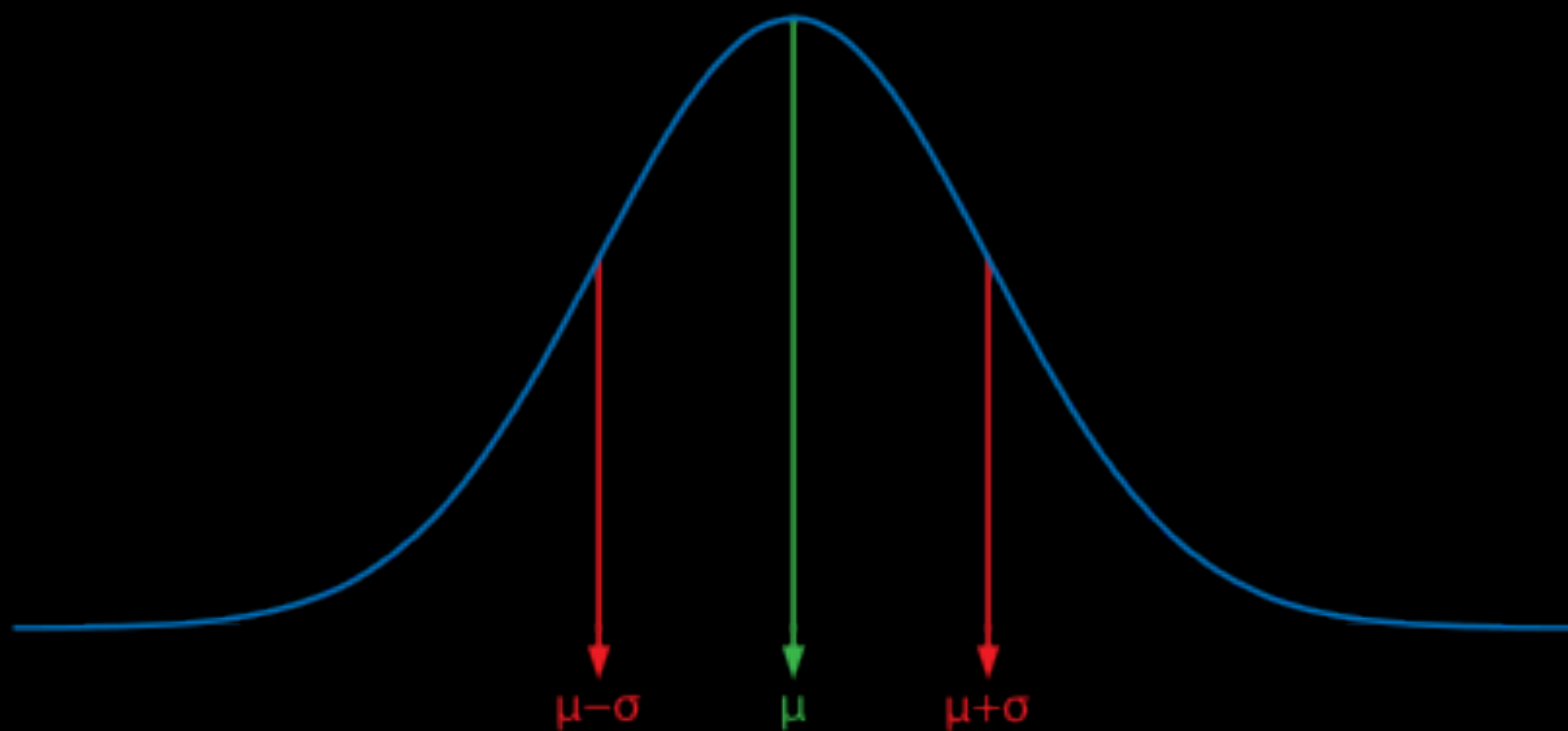
$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$





GAUSSIAN SAMPLING OF β UNCERTAINTY

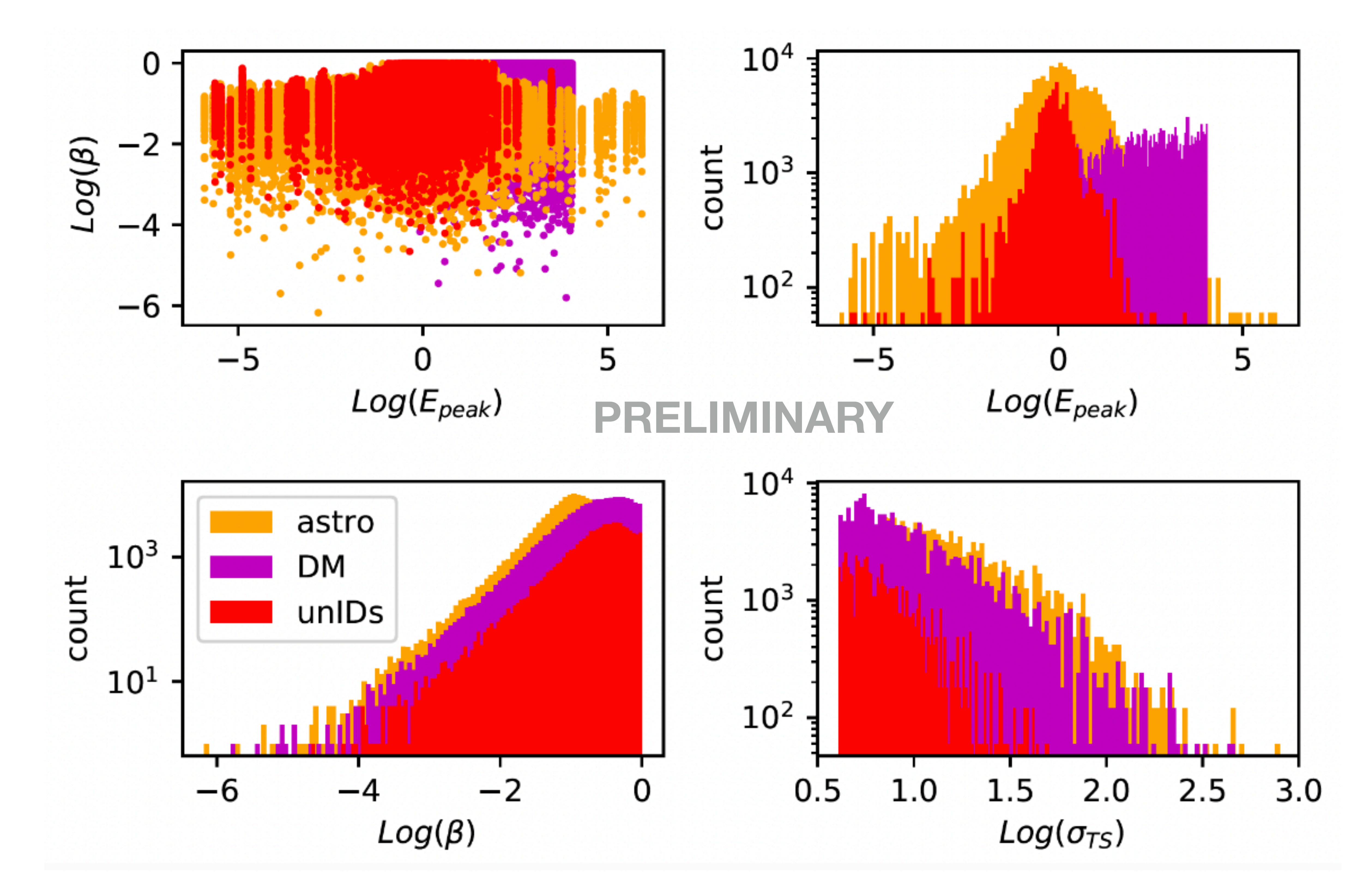
$M = 60$



$\beta - \epsilon_\beta$ β $\beta + \epsilon_\beta$

$0 < \beta \leq 1$ **Is required if β is small and ϵ_β is big**

GAUSSIAN SAMPLING OF β UNCERTAINTY



GAUSSIAN SAMPLING OF β UNCERTAINTY

Related issues:

- **Increasing the number of data** from N (Astro+DM datasets) to MxN makes the learning process slower;
- After the learning step and in order to classify the unIDs, the method would also require the **sample of the unIDs uncertainty**, that is useless for the classification intent itself.

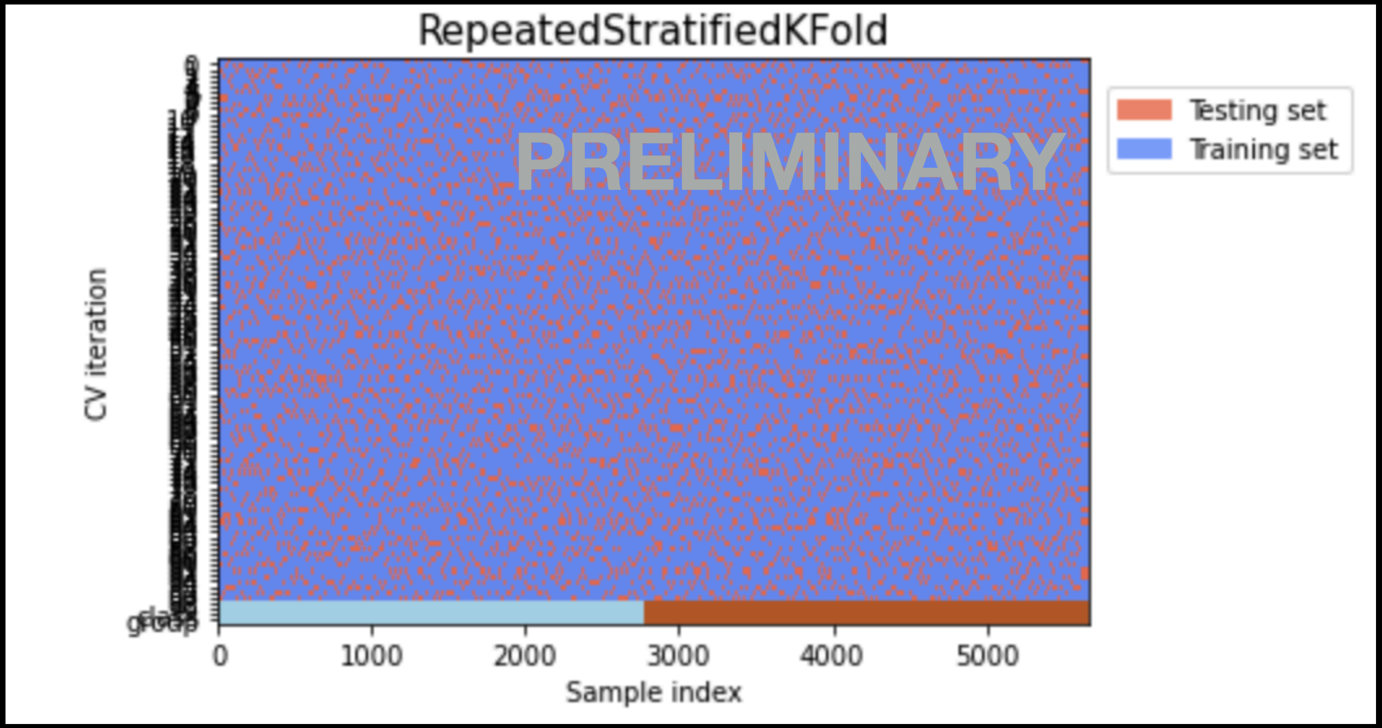
DATA PRE-PROCESSING

1. 10^(-3)GeV < E_peak < 10^6 GeV , reliable range of the Fermi-LAT sensitivity in energy
2. Balanced data: same number of DM and Astro
3. Log scale classification
4. Standardised data: each feature is normalised with respect to their medium values.
5. Training/Testing data set split:

RepeatedStratifiedKFold(n_splits=N_splits, n_repeats=N_Repeats)
Number of folds, N_splits=5 -> Train set = 4530 (80%) data Test set=1132 (20%)
Number of times cross-validator needs to be repeated, N_Repeats=20
N_class=N_splits x N_Repeats= 100

Stratified: The split into N_folds preserve the percentage of samples for each class and without repeated data in different folds.

Repeated: the cross-validation is repeated a number of times with different random seed

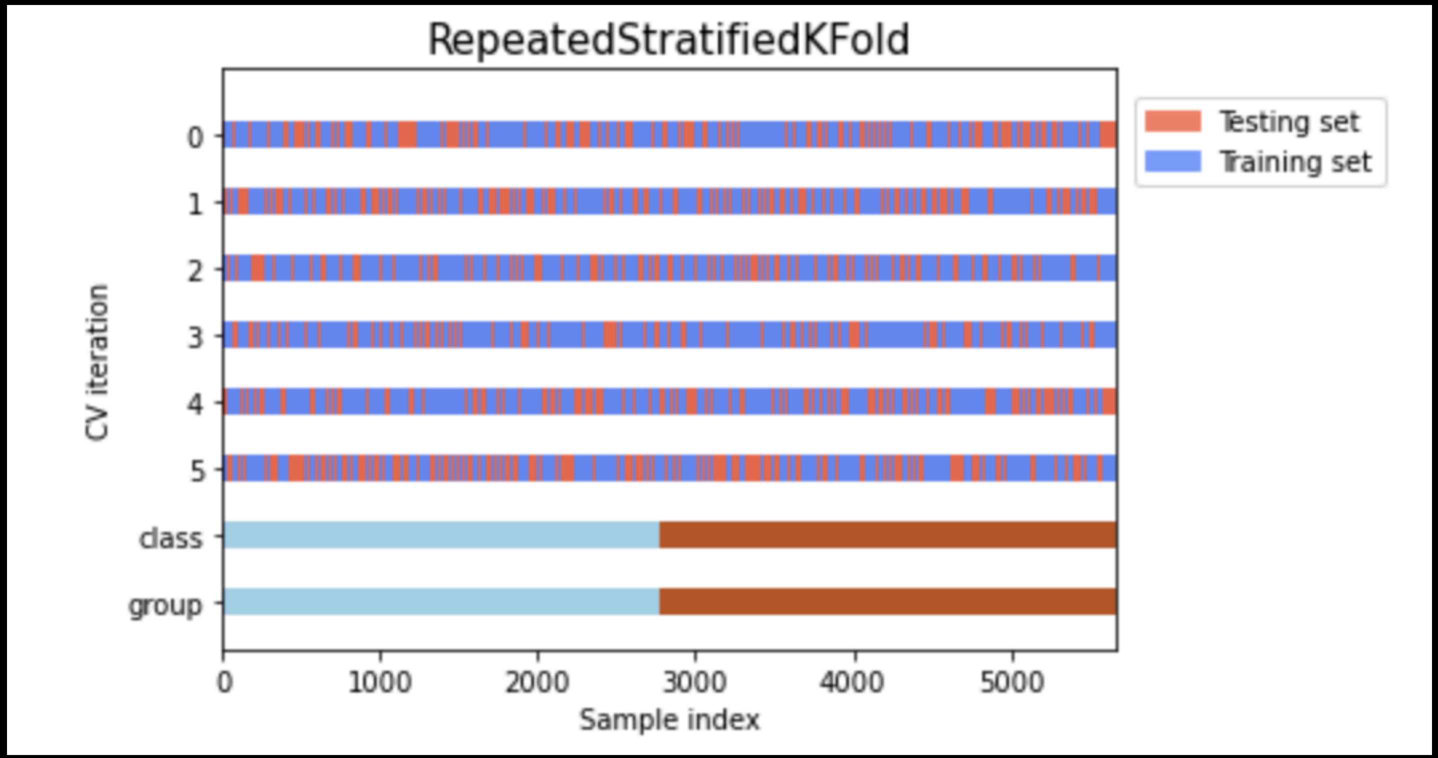


DATA PRE-PROCESSING: CHECK

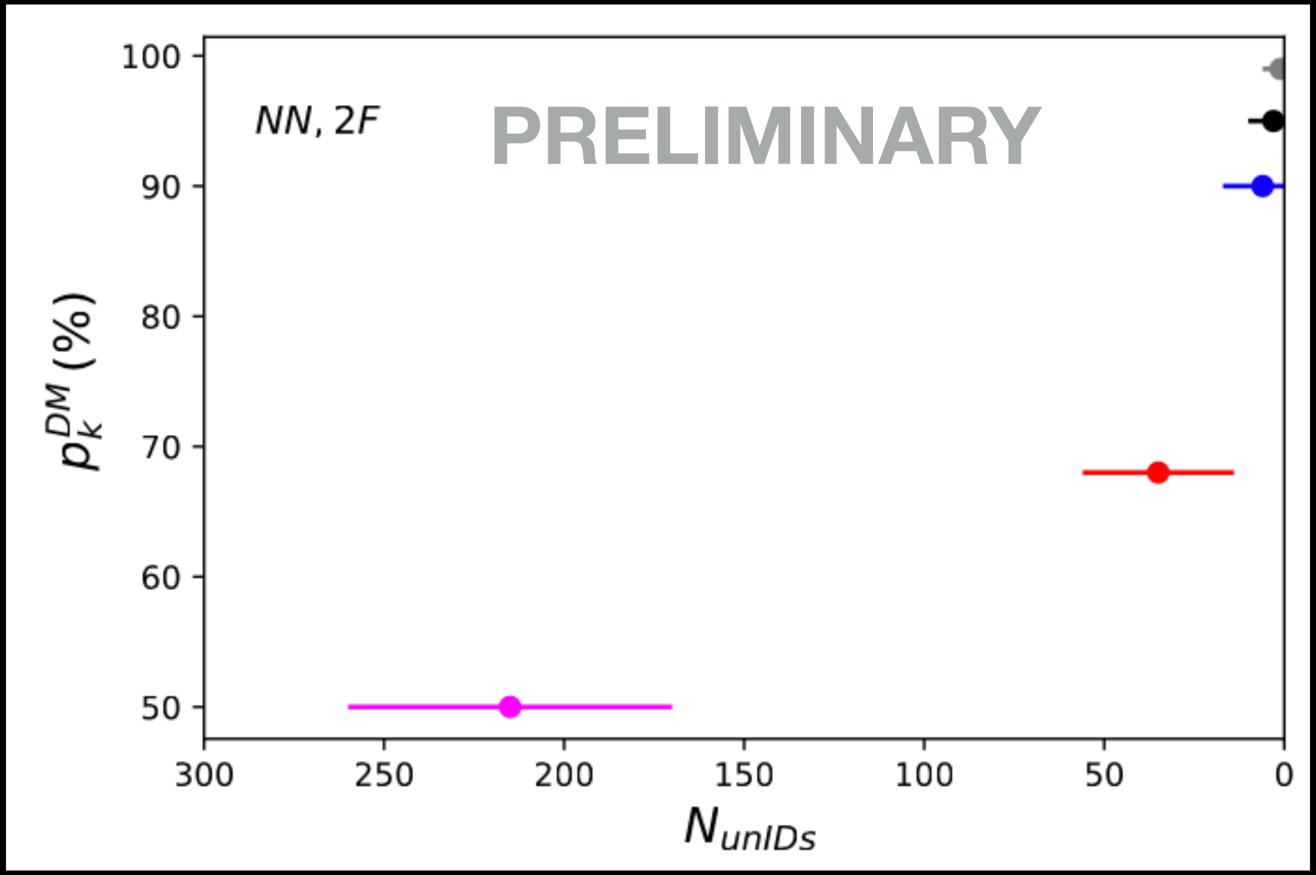
RepeatedStratifiedKFold(n_splits=N_splits, n_repeats=N_Repeats)
Number of folds, N_splits=3 -> Train set = 3774 (80%) data Test set=1888 (33%)
Number of times cross-validator needs to be repeated, N_Repeats=2
N_class=N_splits x N_Repeats= 6

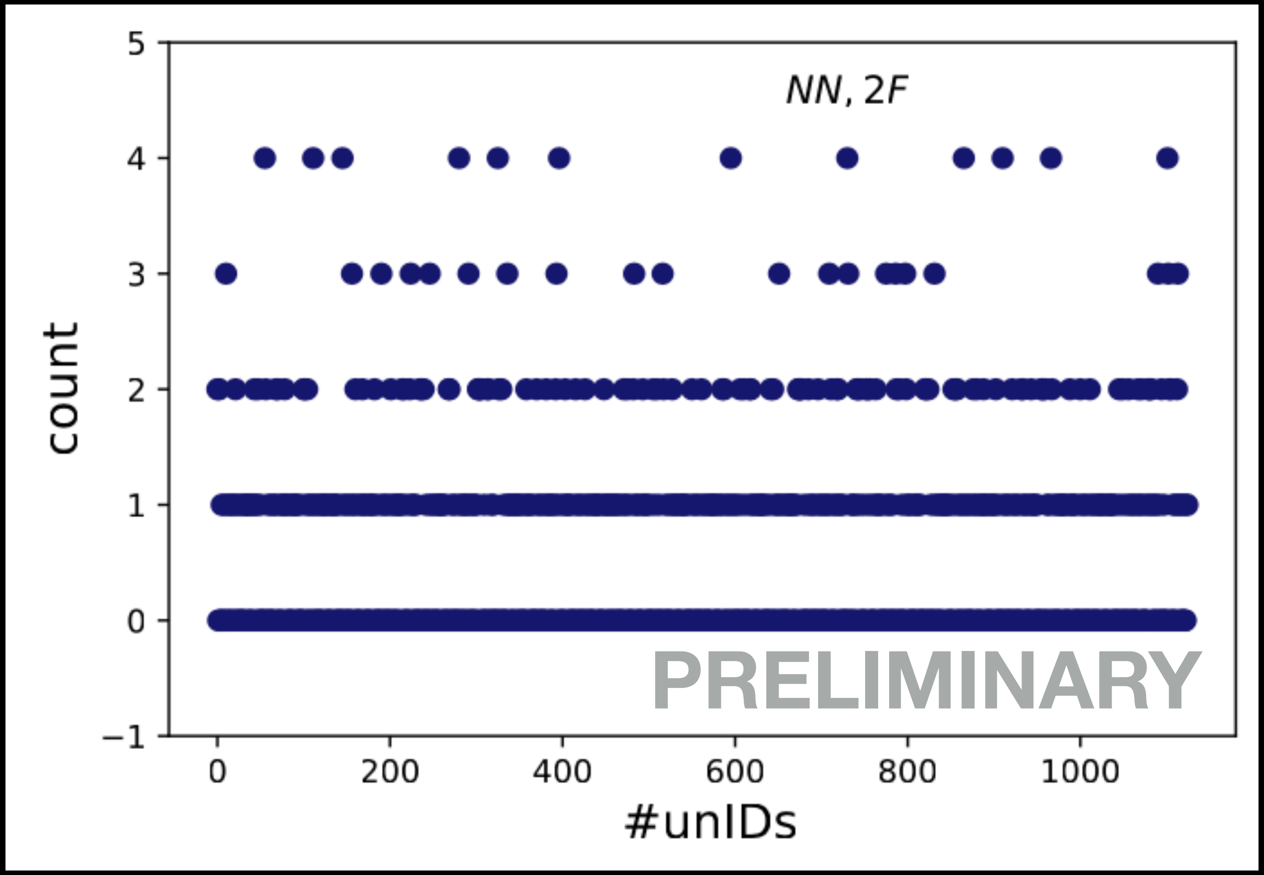
Stratified: The split into N_folds preserve the percentage of samples for each class and without repeated data in different folds.

Repeated: the cross-validation is repeated a number of times with different random seed



UNIDS CLASSIFICATION WITH NN-2F





CLASSIFICATION ALGORITHMS
NAIVE BAYES

Assuming the Bayes' theorem:

$$P(y\,|\,\mathbf{x}) = \frac{P(y)P(\mathbf{x}\,|\,y)}{P(\mathbf{x})}$$

$P(y)$ Prior on the class, e.g. $P(y_0)$ is the probability that a source is astro before to analyse the gamma-ray spectra

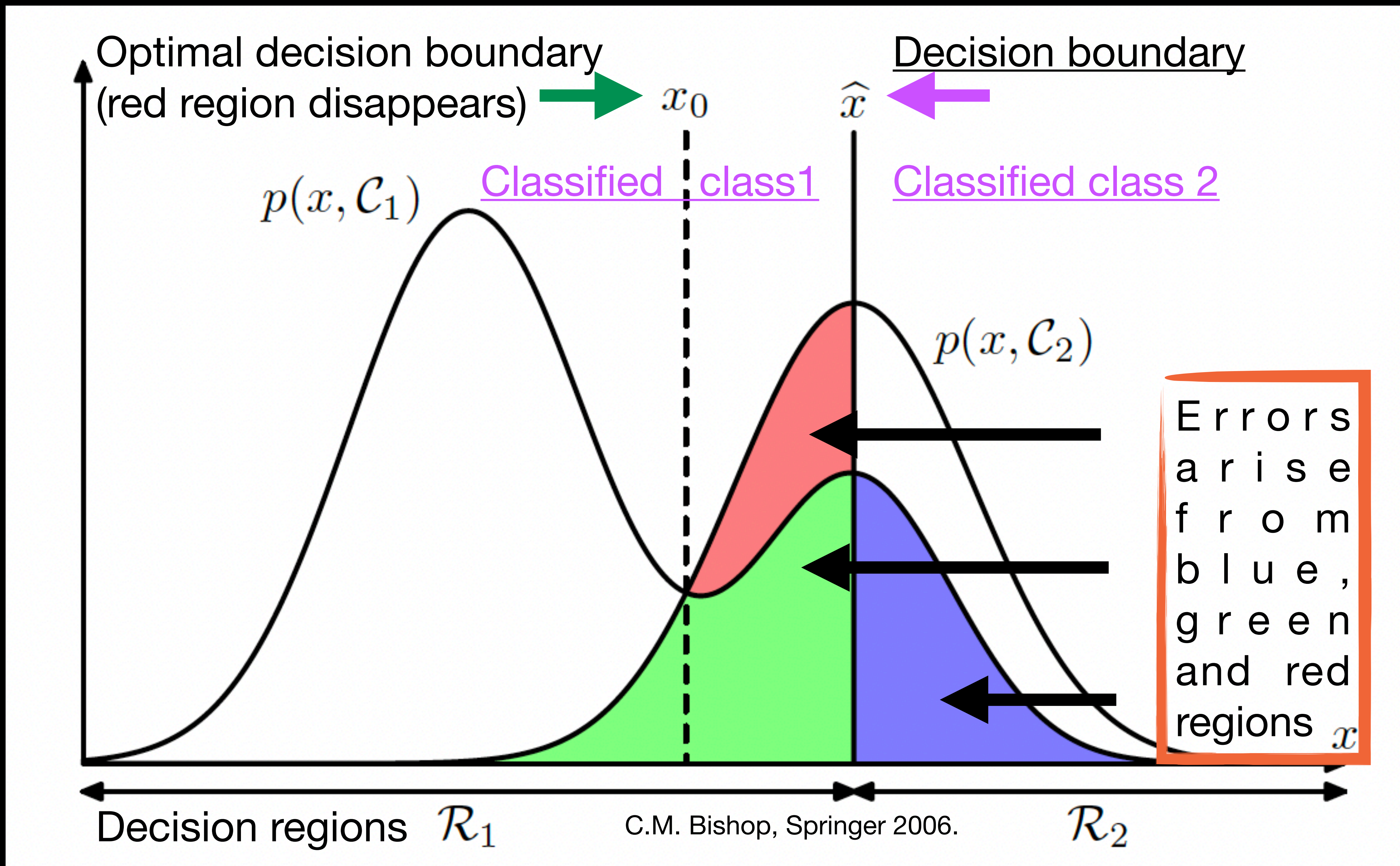
$P(y\,|\,\mathbf{x})$ Posterior: corresponding probability,
e.g. $P(\mathbf{x}\,|\,y_0)$ after the analysis of gamma-ray spectra (posterior)

$P(\mathbf{x}\,|\,y)$ Likelihood, i.e. the most complete probabilistic description of the scientific case

$$P(\mathbf{x}) = \sum_k p_k(\mathbf{x}\,|\,y)p(y)$$ Typically intratable

The "naive" assumption is the conditional independence between every pair of features given the value of the class variable. The solution is obtained by fitting the model for each class separately using the correspondingly labelled data.

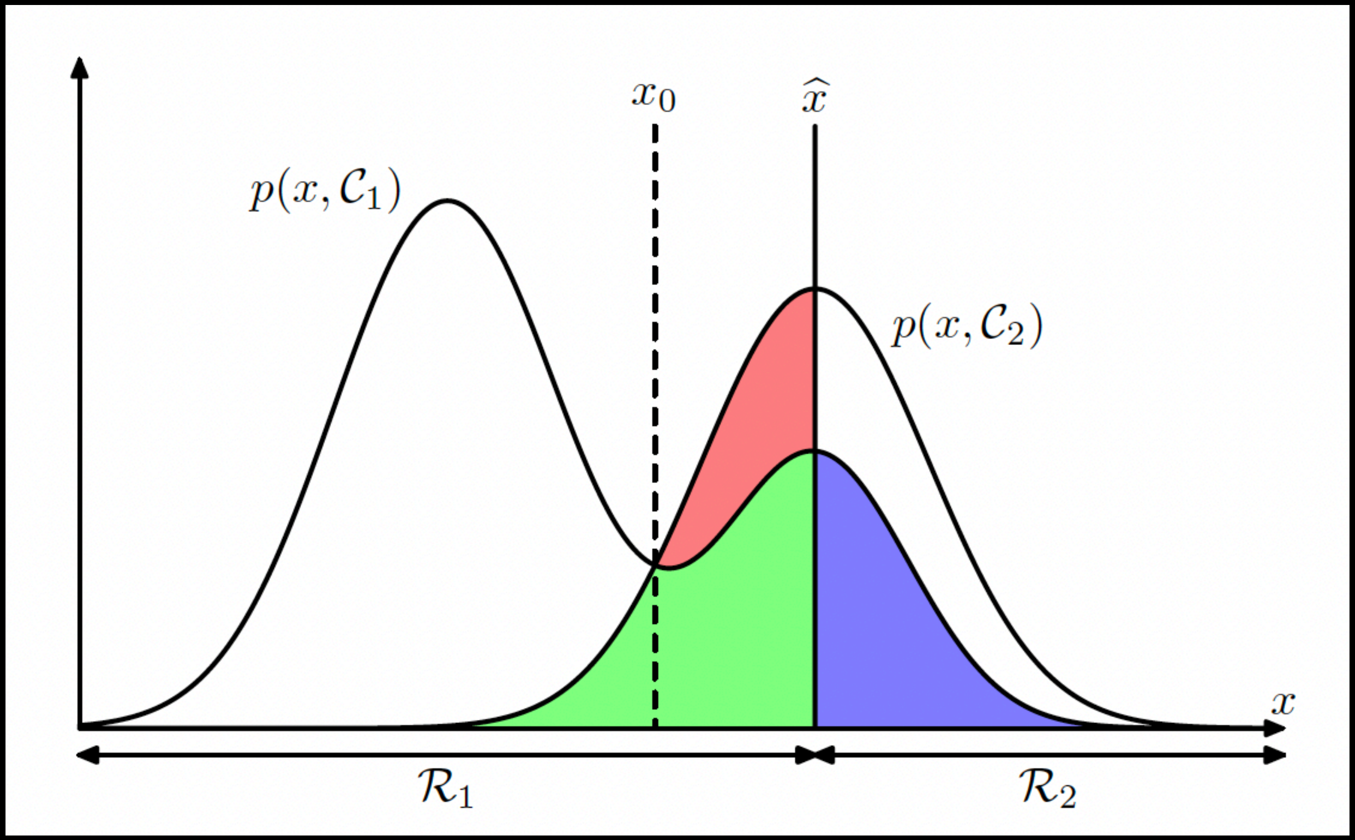
CLASSIFICATION ALGORITHMS
NAIVE BAYES

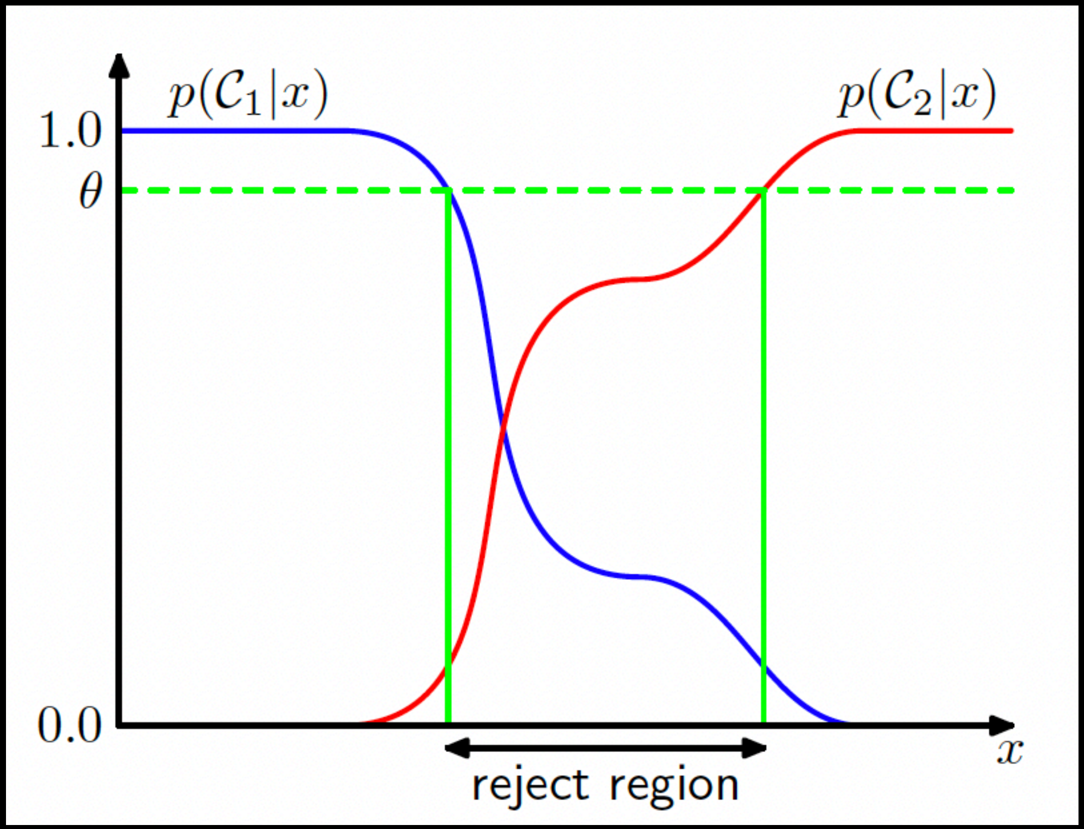


Schematic illustration of the joint probabilities $p(x \mid C_k)$ of each of the two classes plotted against x. $x = x_0$ is equivalent to the minimum misclassification rate decision rule, which assigns each value of $\mathbf{x}$ to the class having the higher posterior probability

CLASSIFICATION ALGORITHMS
NAIVE BAYES

Having found the posterior probabilities $p(C_k, x)$, we use decision theory to determine class membership for each new input x. If our aim is to minimize the chance of assigning x to the wrong class, then intuitively we would choose the class having the higher posterior probability (e.g. here astrophysical sources).





C.M. Bishop, Springer 2006.

Illustration of the reject option. Inputs such that the larger of the two posterior probabilities is less than or equal to some threshold will be rejected.

CLASSIFICATION ALGORITHMS
GAUSSIAN PROCESS WITH NOISY INPUTS

Based on:

Multi-class Gaussian Process Classification with Noisy Inputs

Autor (es): Villacampa-Calvo, Carlos; Zaldívar, Bryan; Garrido-Merchán, Eduardo C.; Hernández Lobato, Daniel

Entidad: UAM. Departamento de Ingeniería Informática

Editor: Microtome Publishing

Fecha de edición: 2021-01

Cita: Journal Of Machine Learning Research 22.36 (2021): 1−52

ISSN: 1532-4435 (print); 1533-7928 (online)

CLASSIFICATION ALGORITHMS
GAUSSIAN PROCESS WITH NOISY INPUTS

VILLACAMPA, GARRIDO, HERNÁNDEZ, AND BZ, JOURNAL OF ML RESEARCH, 2020

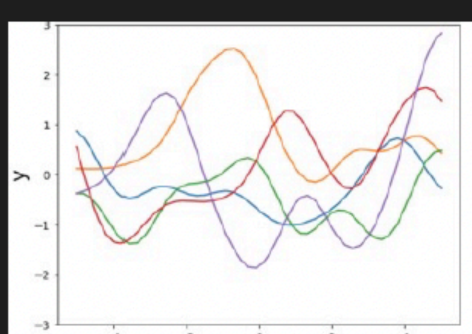
- Idea: Each input $\vec{x}_i = \underline{\vec{x}}_i + \Delta\vec{x}_i$ → Input uncertainty (assumed Gaussian)
  - $\vec{x}_i$: noisy observation
  - $\underline{\vec{x}}_i$: actual input (unknown) → To be modelled

  (in analogy to what we typically do for the dependent variable y)

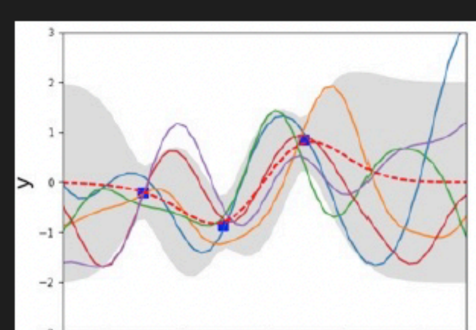
- Likelihood of data $p(Y, X \mid F, \underline{X}) = \prod_i^N p(y_i \mid f(\underline{\vec{x}}_i)) \cdot \mathcal{N}(\vec{x}_i \mid \underline{\vec{x}}_i, \Delta\vec{x}_i)$
  - F: latent variables for the output Y

- Y modelled as a Gaussian Process (GP) → Very popular Stochastic Process in ML, based on Gaussian distrib. [over functions]

GP prior



GP posterior after 3 obs.



- GP give analytical predictions in regression problems
- For classification the posterior should be approximated
  - ↳ nowadays typically using Variational Inference

Credit: Bryan Zaldivar

N-SPLITS TRAINING/TESTING SET