Hunting dark matter signals with deep learning at the LHC

Dr. Andres D. Perez
IFLP-CONICET, Argentina

Work in colaboration with:
Ernesto Arganda, Anibal Medina,
and Alejandro Szynkman

I F L P
CONICET
U N L P

March 31, 2021
@ Feno-Exp

Plan

- **Models and sample generation**
  - Simplified models
  - Kinematic features
  - Benchmark models
- **Neural Network algorithms**
  - Event-by-even data
  - Data as 2D histograms → DNNs and CNNs
- **Flexibility of the method**
  - Performance invariance with the number of background events
  - Testing under incorrect training hypothesis:
    - coupling values
    - benchmark model
- **Multimodel Classifiers**
  - Binary classifier
  - Multiclass classifier
- **Conclusions**

What we want

- Study several simplified **dark matter** models and their signatures at the **LHC** using **Neural Networks**

  → monojet plus missing transverse energy channel.

- Determine the **viability** of deep learning methods and analyze **flexible** they are

  → different neural networks and different data representations

Models and sample generation

Simplified models
Kinematic features
Benchmark models

Simplified models

- **DM with a spin-0 mediator**
- **DM with a spin-1 mediator**
- **DM with a spin-2 mediator**

m_{DM}: DM mass
m_Y: mediator mass
DM-mediator couplings
SM-mediator coupling

- **Axion-Like Particle (ALP) as DM**

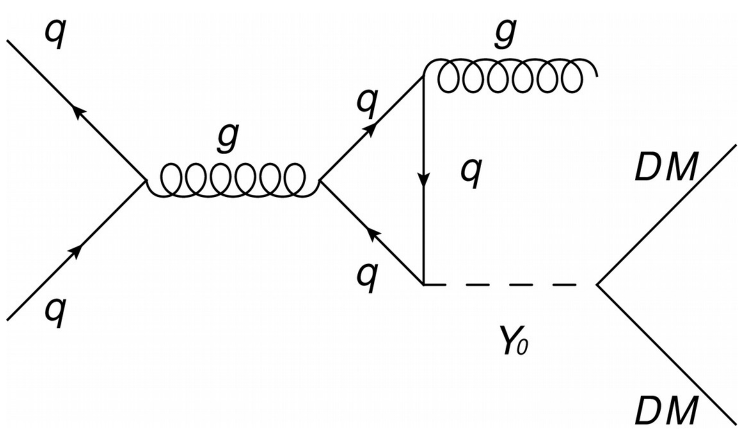
m_a: axion mass (DM)
f_a: axion scale
SM-axion couplings

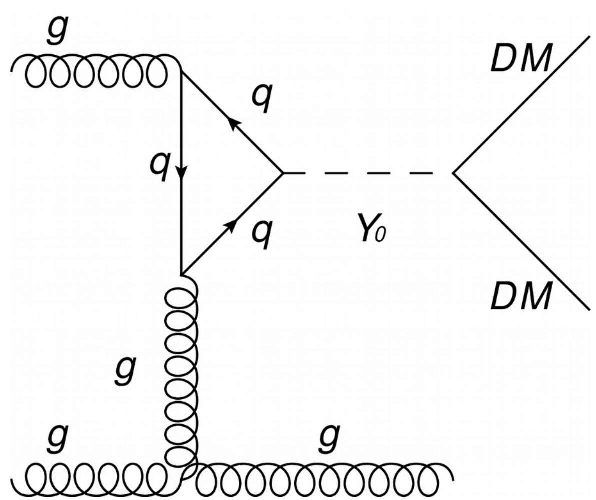
Simplified models

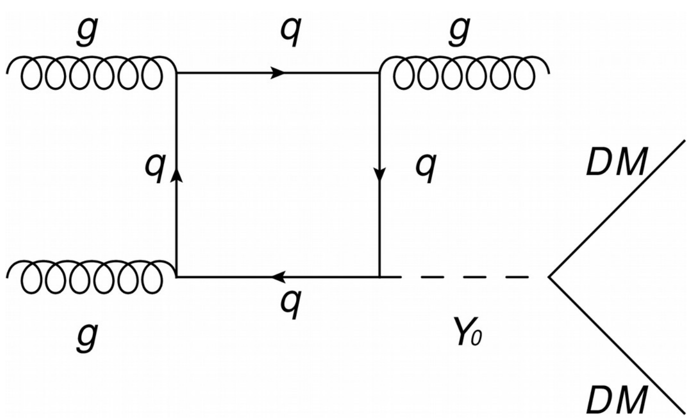
monoject plus
missing transverse
energy channel

- **DM with a spin-0 mediator**

$pp \rightarrow DM\ DM\ j$





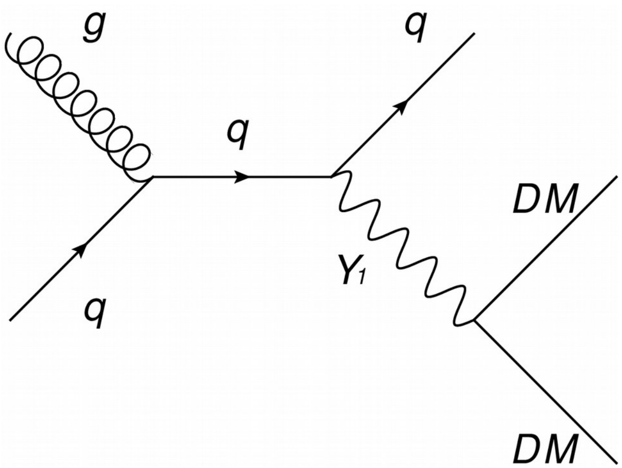


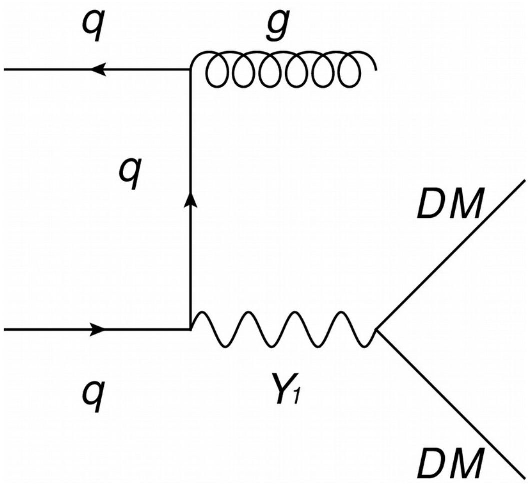
Simplified models

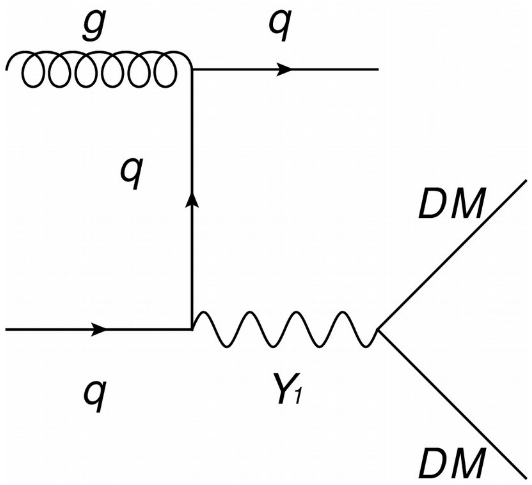
monojet plus
missing transverse
energy channel

- **DM with a spin-1 mediator**

$pp \rightarrow DM\, DM\, j$





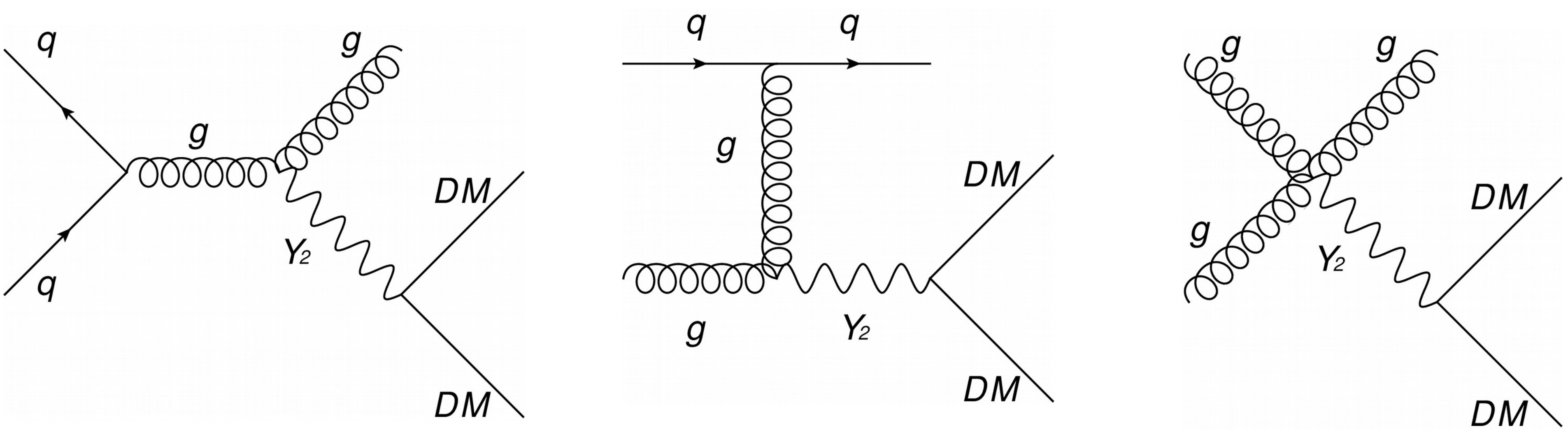


Simplified models

monojet plus
missing transverse
energy channel

- **DM with a spin-2 mediator**

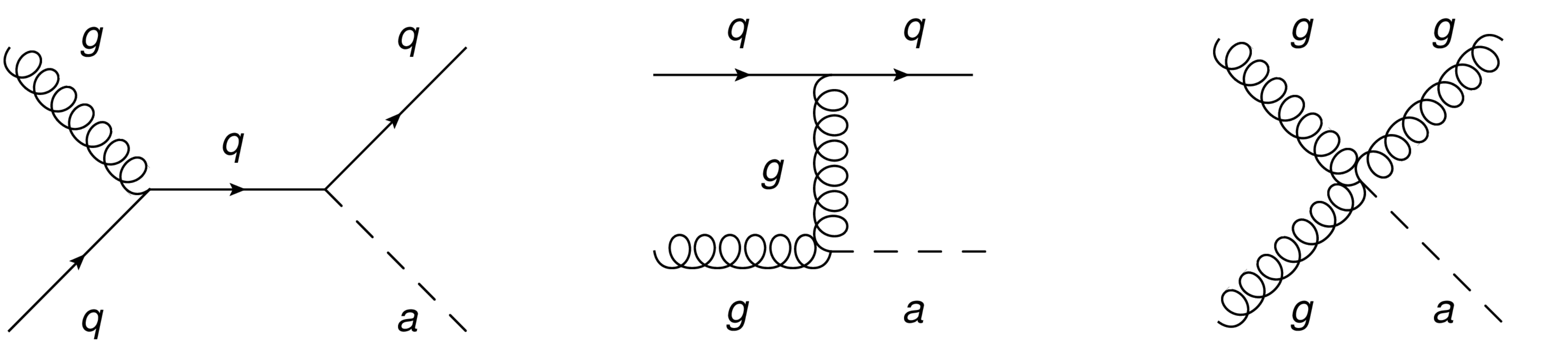
$pp \to DM\ DM\ j$



Simplified models

monoject plus
missing transverse
energy channel

- **Axion-Like Particle (ALP) as DM** $pp \to a\, j$

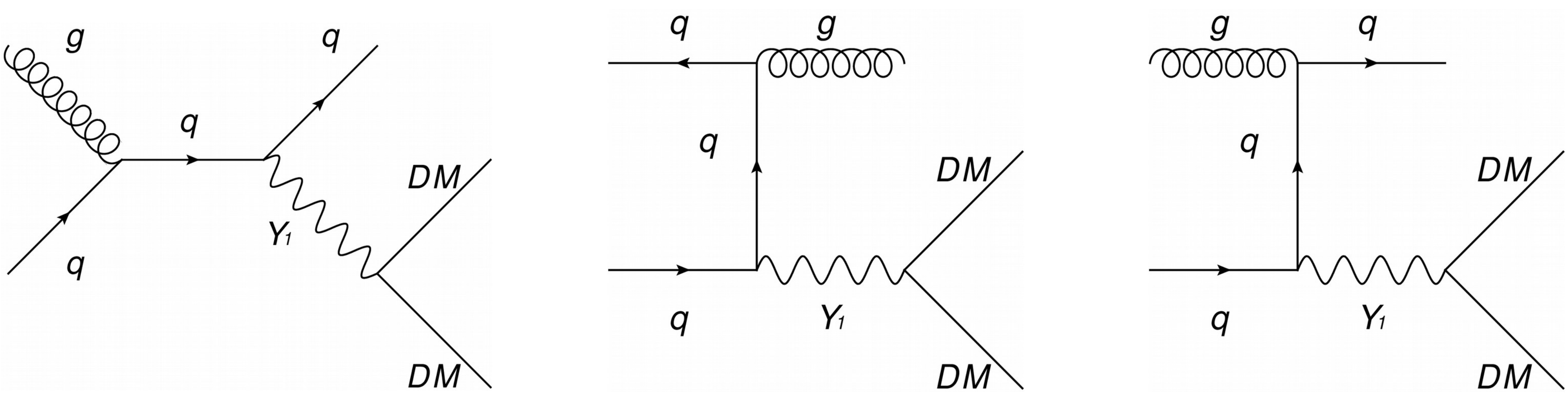


Simplified models

monojet plus
missing transverse
energy channel

- **SM background**

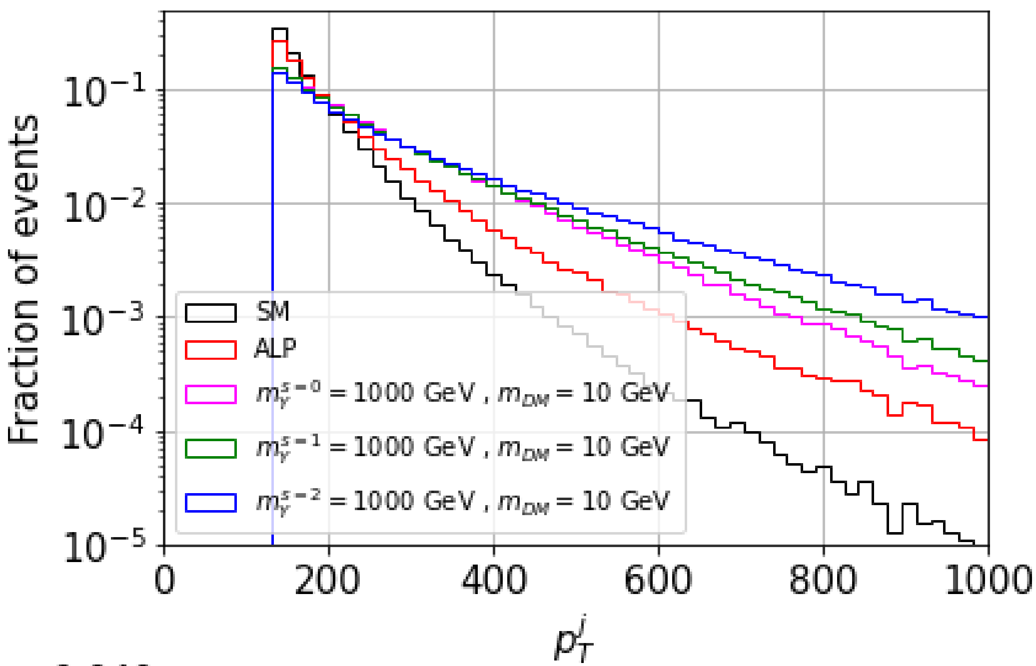
$pp \rightarrow Z\,j\ (Z \rightarrow \nu\nu)$

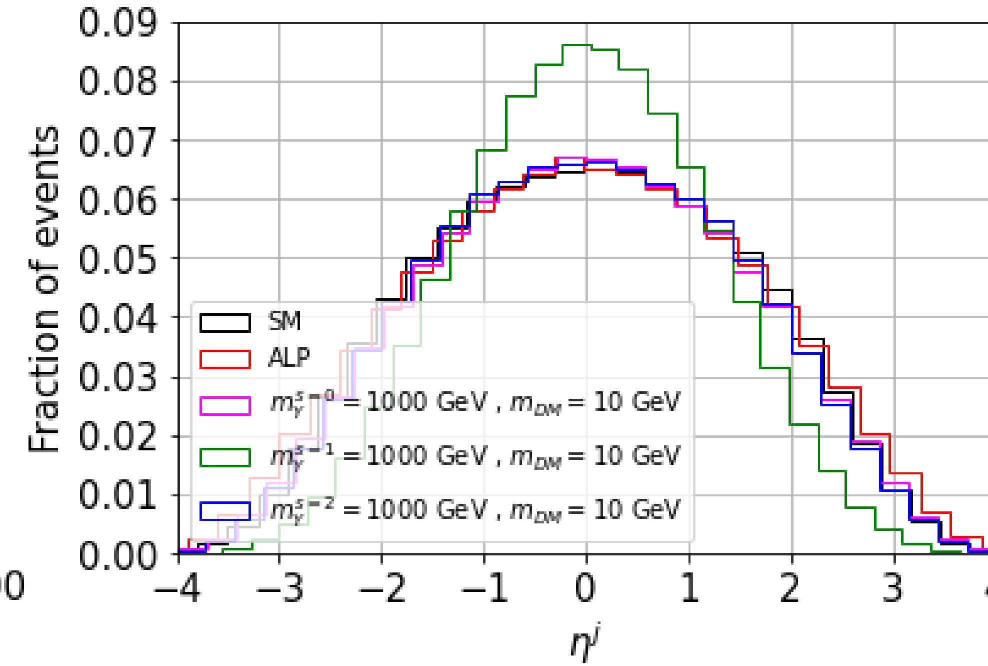


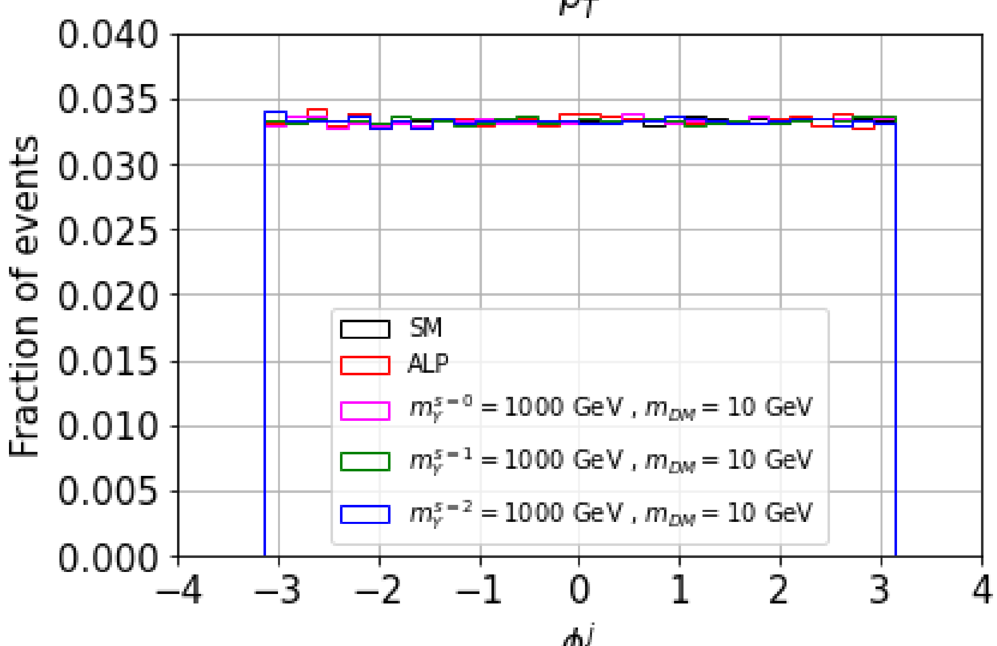
Y → Z
DM → ν

Kinematic distributions

Input data for our deep learning algorithms







1. The coupling values do not modify the kinematic distributions
→ *models with a mediator defined by* (m_{DM} , m_Y)

2. ALPs distributions are independent of m_a and f_a.

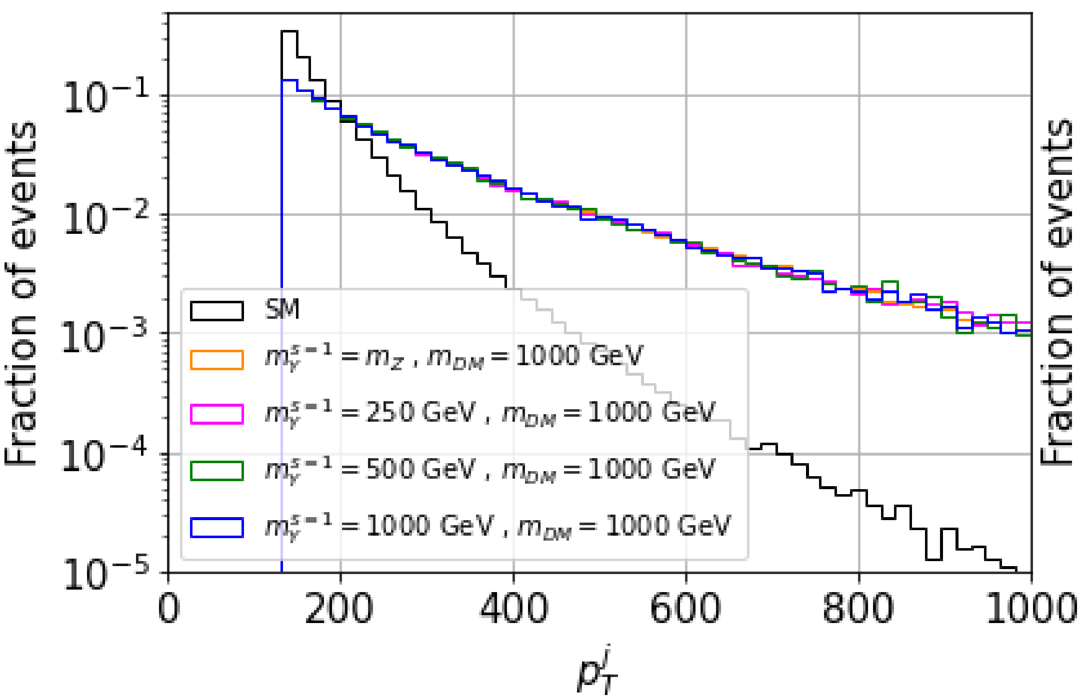
3. The azimuthal angle distribution does not show any useful structure.

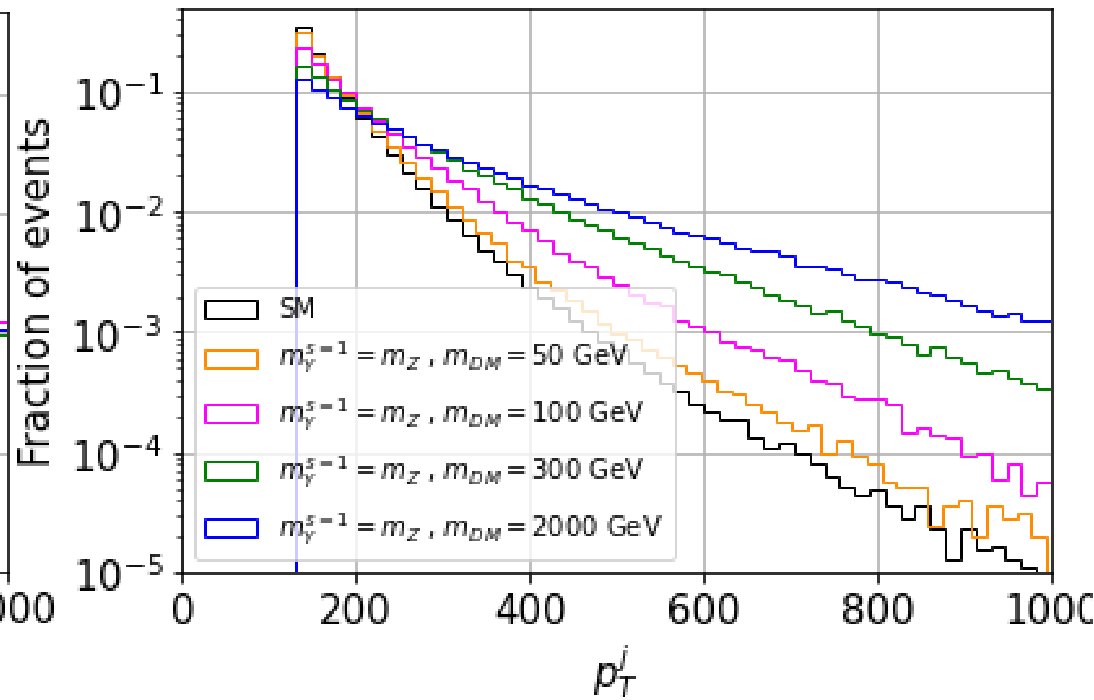
Kinematic distributions

Models with a mediator

1. The coupling values do not modify the kinematic distributions
→ *models with a mediator defined by* (m_{DM} , m_Y)

But if the parameter space is divided according to the mediator status:





Mediator status	Relevant parameter
Off-shell	m_{DM}
On-shell	m_Y
Off-shell PS (by phase space)	m_{DM}

Benchmark models

1.5M events

0.5M events

Benchmark Model		Label		
		Individual	Multimodel	
		Binary	Binary	Multiclass
SM only		0	0	0
ALPs		1	1	1
Spin-0 mediator	$m_Y^{s=0} = m_Z$, $m_{DM} = 300$ GeV	1	1	2
	$m_Y^{s=0} = 1000$ GeV , $m_{DM} = 10$ GeV	1	1	3
	$m_Y^{s=0} = 10$ TeV , $m_{DM} = 10$ GeV	1	-	-
Spin-1 mediator	$m_Y^{s=1} = m_Z$, $m_{DM} = 300$ GeV	1	1	4
	$m_Y^{s=1} = 1000$ GeV , $m_{DM} = 10$ GeV	1	-	-
	$m_Y^{s=1} = 10$ TeV , $m_{DM} = 10$ GeV	1	1	5
Spin-2 mediator	$m_Y^{s=2} = m_Z$, $m_{DM} = 300$ GeV	1	-	-
	$m_Y^{s=2} = 1000$ GeV , $m_{DM} = 10$ GeV	1	1	6
	$m_Y^{s=2} = 10$ TeV , $m_{DM} = 10$ GeV	1	1	7

MadGraph5_aMC@NLO to generate events with monojets plus missing energy at parton level.
Parton shower and hadronization are performed with **Pythia.**
Detector-level data is simulated using **Delphes** with the default ATLAS card.

$\sqrt{s}$= 14TeV generation level cuts: $p_j^T \geq 130$GeV and $|\eta_j| \leq 5$ for the leading jet.

Neural Networks algorithms

Event-by-even data
Data as 2D histograms → DNNs and CNNs

DNN: Deep Neural Networks

All the algorithms are constructed in **Python** using the library **Keras** along with **TensorFlow** for backend implementation

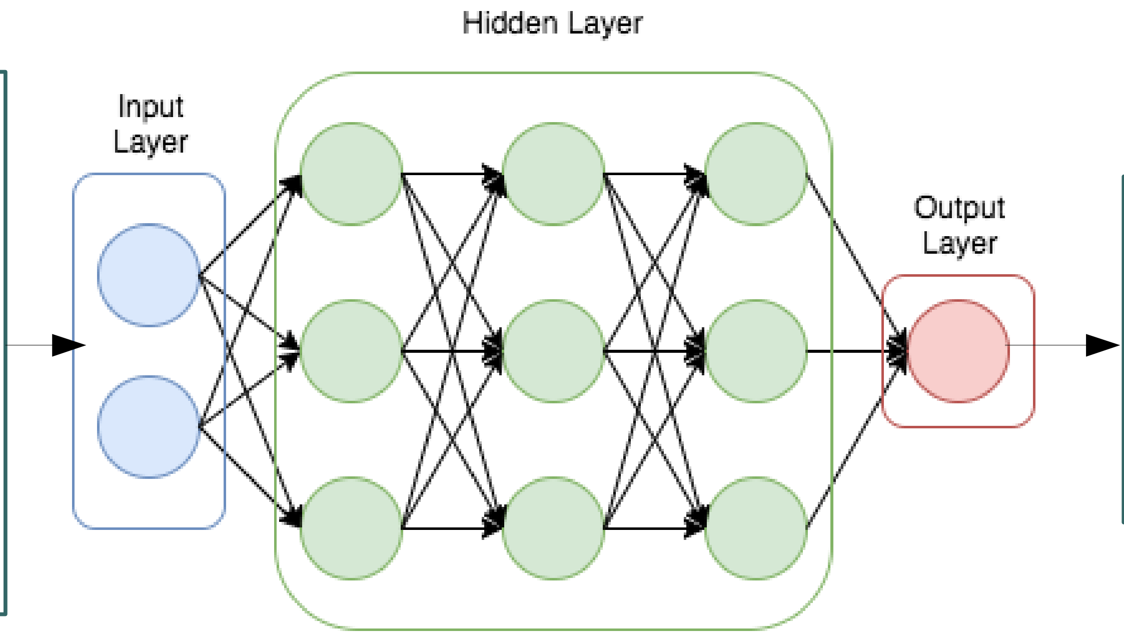
Supervised Learning

Training

2 Data sets:

No Signal → SM background
→ **labeled ‘0’**

Signal → New Physics
→ **labeled ‘1’**



Output

Number:

[0,1]

Trained to discriminate data samples with Signal and No Signal

Trained each benchmark model vs SM **individually**.

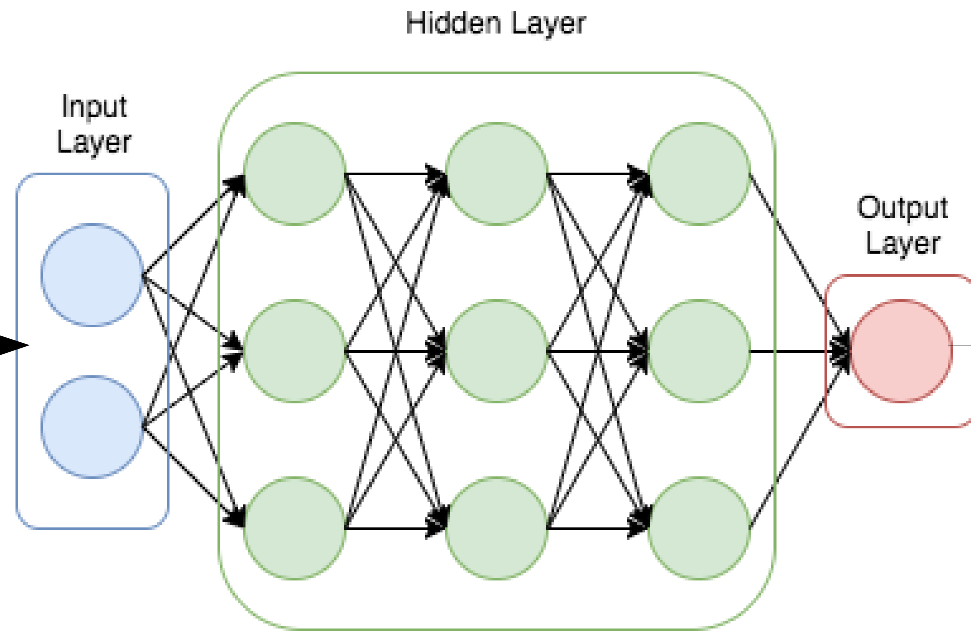
DNN: Deep Neural Networks

All the algorithms are constructed in **Python** using the library **Keras** along with **TensorFlow** for backend implementation

Supervised Learning

Testing

1 data sample
that I do not know if it is
Signal or No Signal:



Output

e.g.:

0.95

Threshold: 0.5 → Output > threshold

→ The DNN predicted **‘1’**
Signal → New Physics

DNN with Event-by-event data

We simulated **1.5M SM events** and **0.5M New Physics events**

Each event has the monojet kinematic information (p_T^j, η^j, Φ^j)

Input

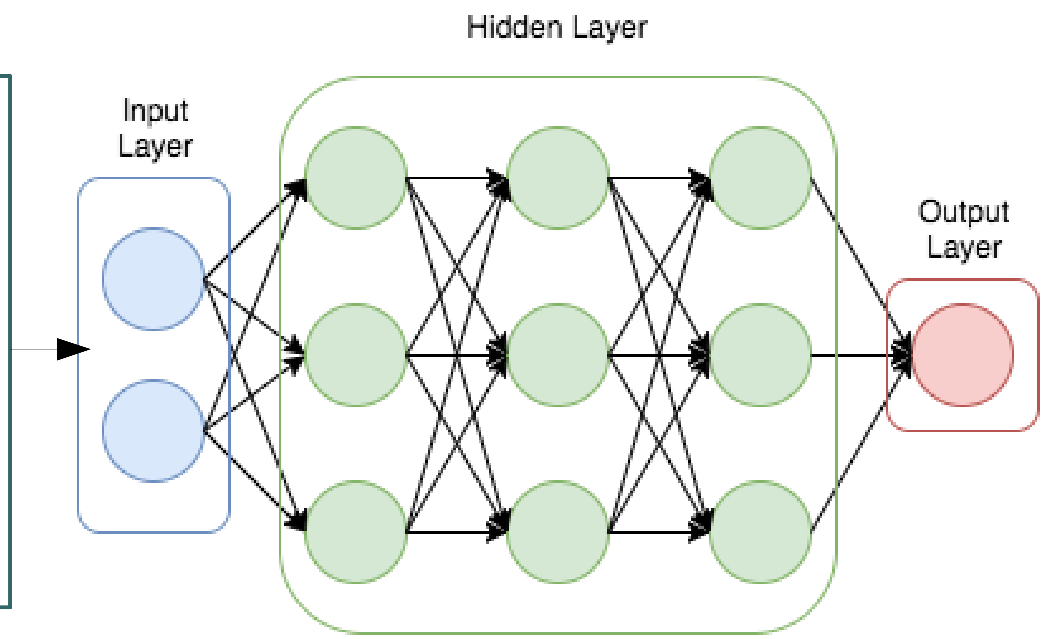
Each data sample is a single event:

Event 1 → (p_T^j, η^j, Φ^j)

Event 2 → (p_T^j, η^j, Φ^j)

...

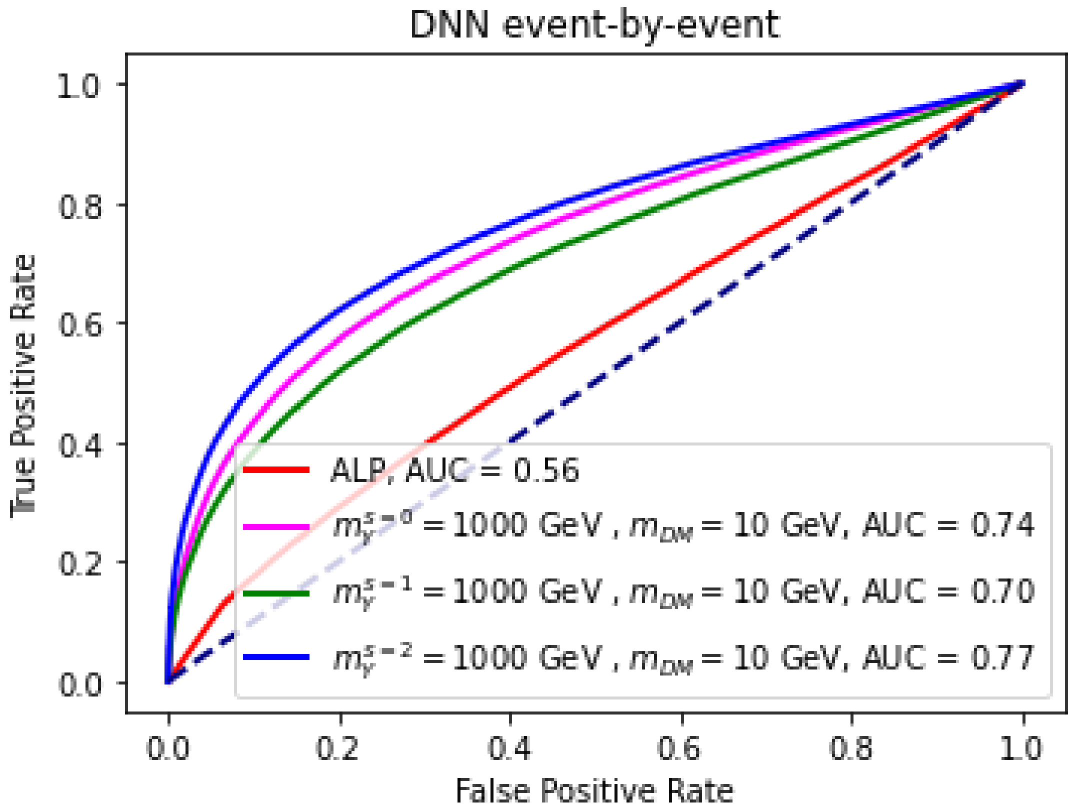
Event N → (p_T^j, η^j, Φ^j)



Data samples are divided with a 0.64:0.20:0.16 train-test-validation ratio

DNN with Event-by-event data

Receiver Operating Characteristic (ROC) curves:

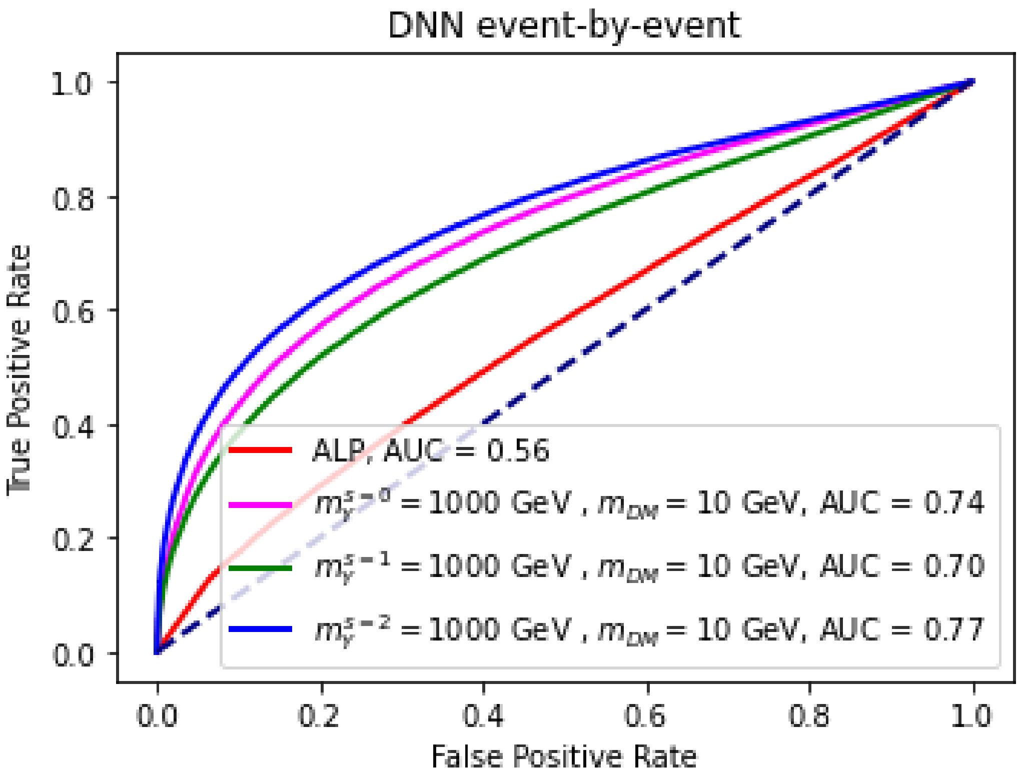


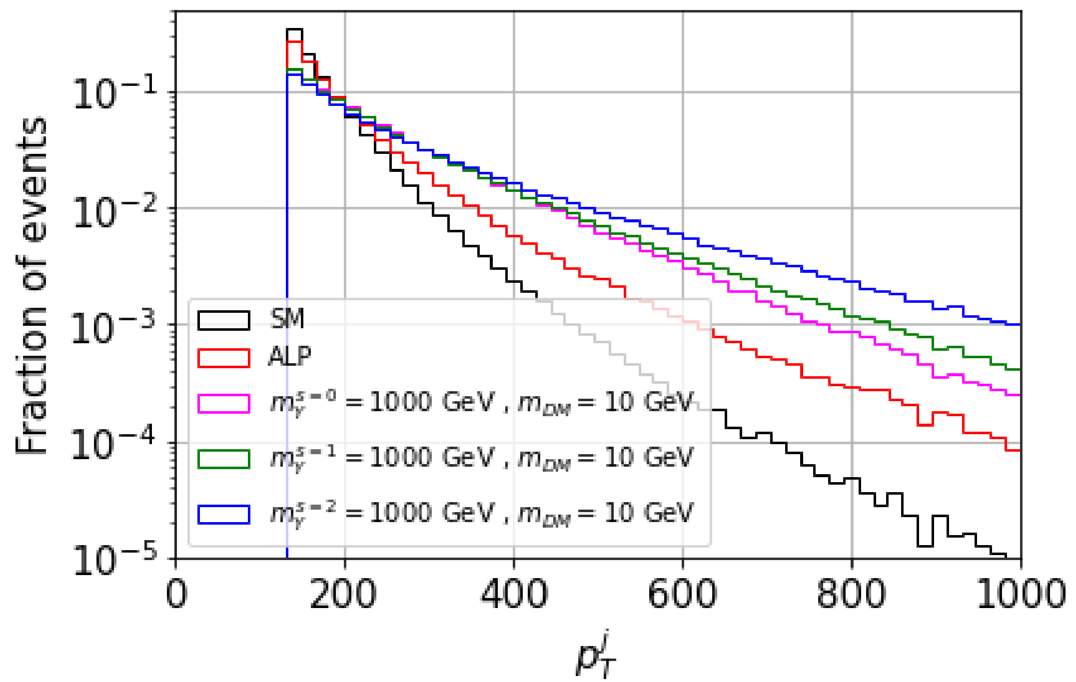
The area under the ROC curve (AUC), a conventional metric to test the performance of binary classifiers

AUC=1 is a perfect classifier, and **AUC=0.5** represents a random classifier

DNN with Event-by-event data

Receiver Operating Characteristic (ROC) curves:





The area under the ROC curve (AUC), a conventional metric to test the performance of binary classifiers

AUC=1 is a perfect classifier, and **AUC=0.5** represents a random classifier

DNN with data as 2D histograms

S: # NP events
B: # SM events

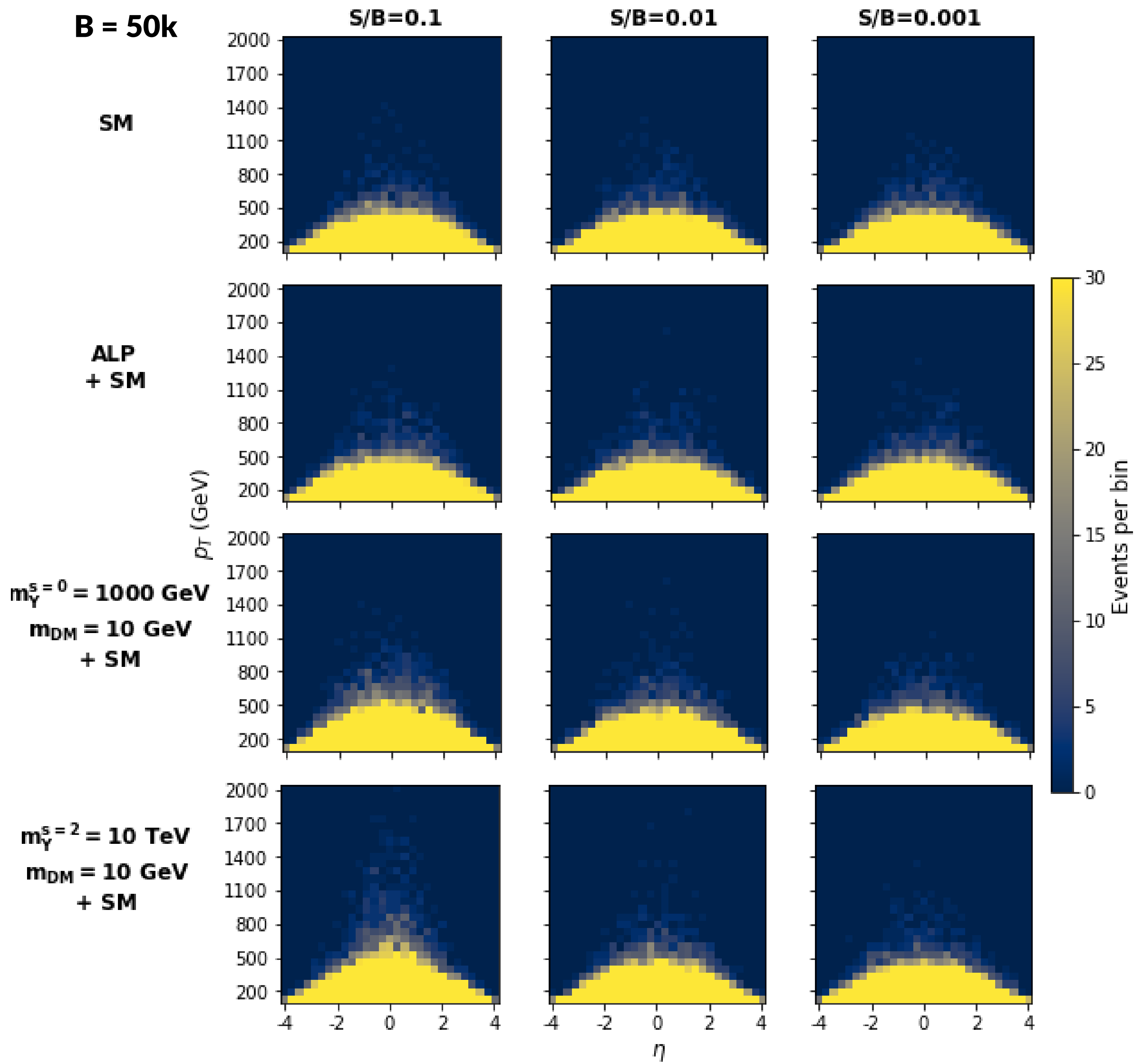
The jet azimuthal angle Φ^j does not provide any useful information.

We can construct 2D histograms made from the pair **(p_T^j, η^j)**

. 20k histograms
with only SM events

. 20k histograms
with NP + SM events

per benchmark model and per S/B ratio

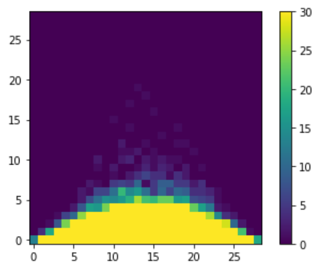


DNN with data as 2D histograms

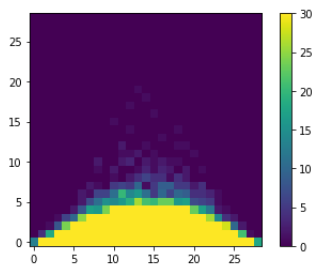
We simulated
20k SM only histograms and
20k New Physics + SM histograms (per benchmark model and per S/B ratio)

Input

Each data sample is a single histogram:

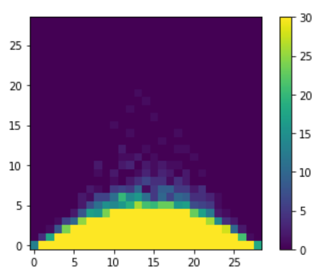


Data sample 1
SM only
labeled '0'

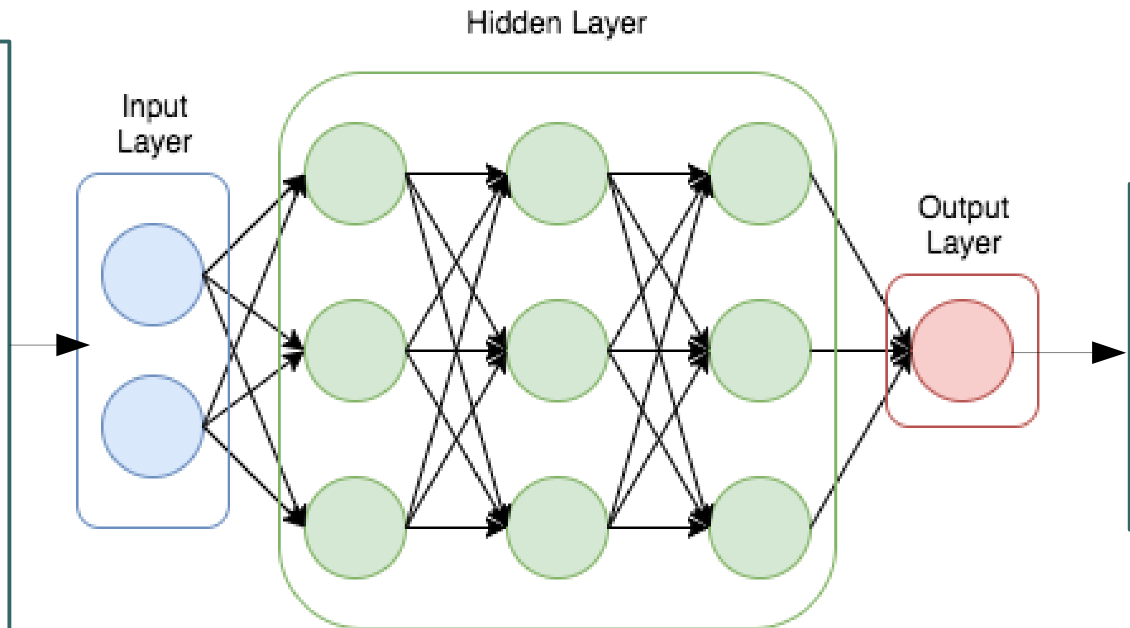


Data sample 2
NP + SM
labeled '1'

... ...



Data sample N
SM only
labeled '0'



Output

Number:

[0,1]

DNN trained to discriminate:
histograms with SM only events vs
histograms with NP+SM events

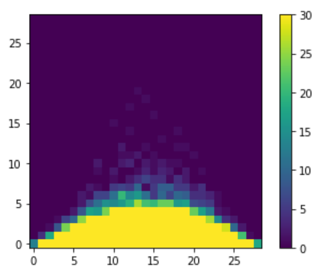
Trained each benchmark model vs SM **individually**.

DNN with data as 2D histograms

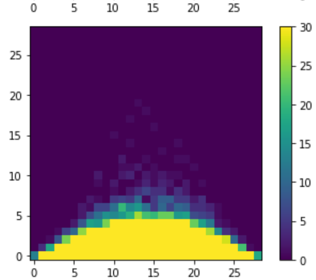
We simulated
20k SM only histograms and
20k New Physics + SM histograms (per benchmark model and per S/B ratio)

Input

Each data sample is a single histogram:

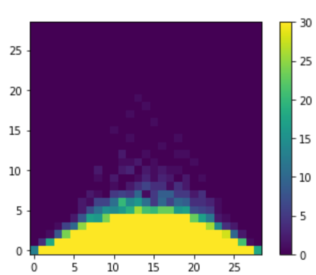


Data sa
SM
label

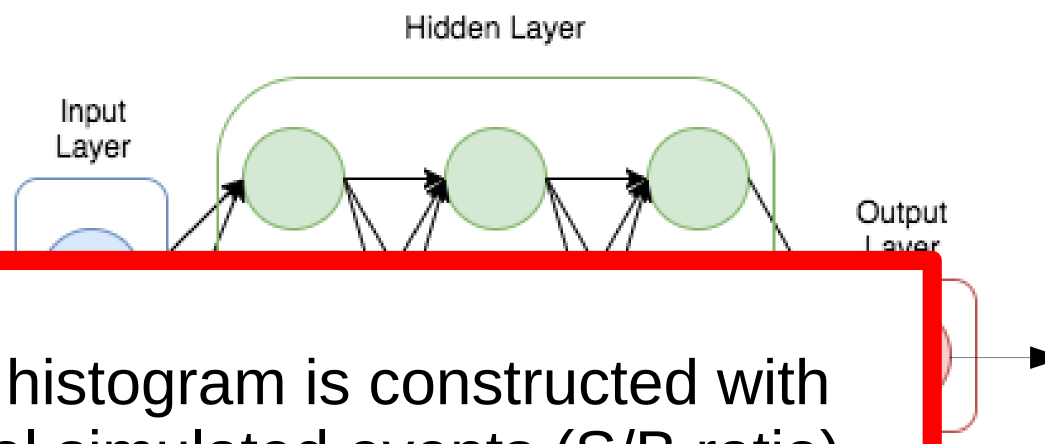


Data sa
NP +
label

...



Data sample N
SM only
labeled '0'



Output

Number:

[0,1]

Each histogram is constructed with several simulated events (S/B ratio)

BUT now the DNN considers each histogram as a single entity

1 data sample = 1 histogram

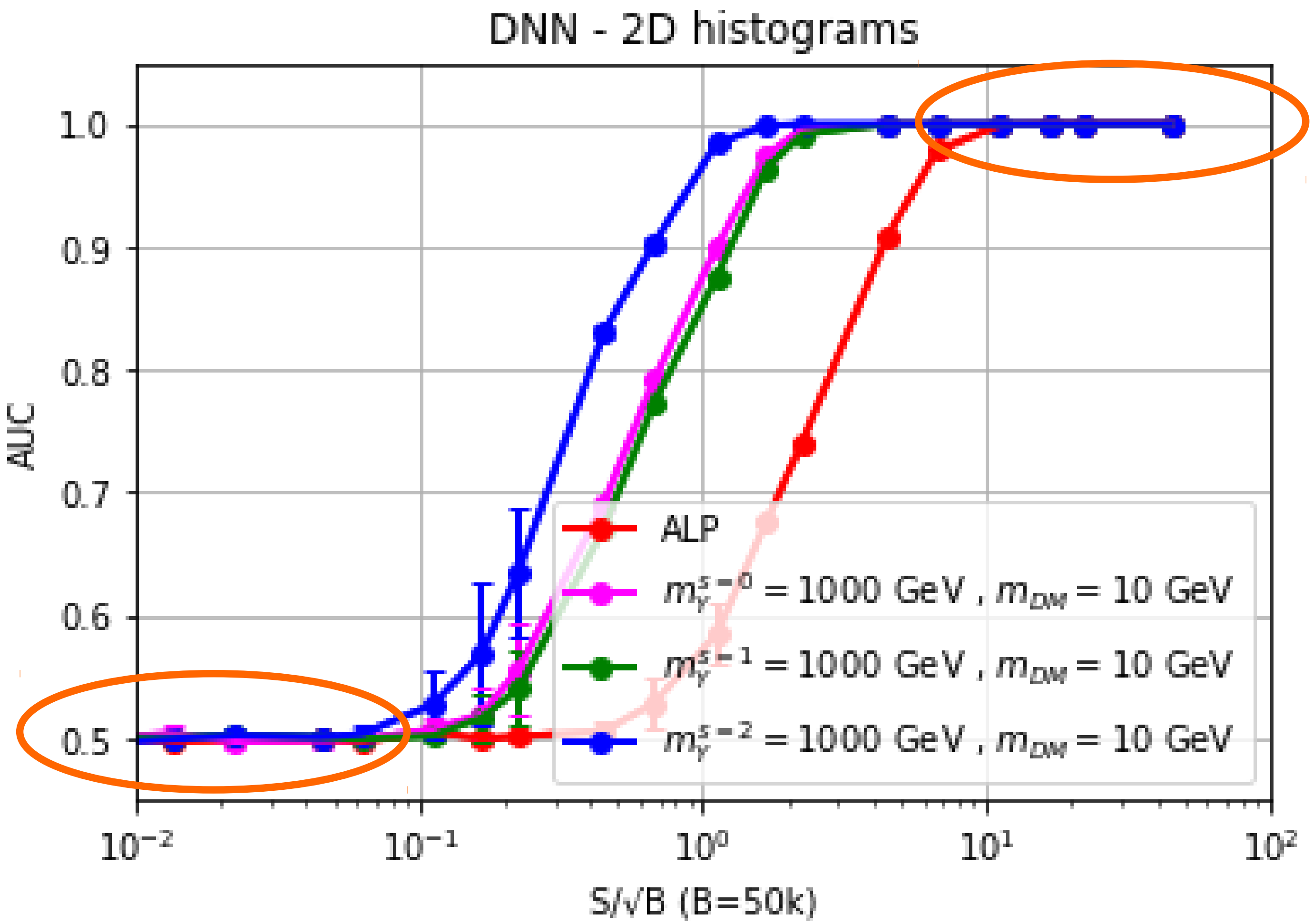
nate:
vents vs
events

Trained each benchmark model vs SM **individually**.

DNN with data as 2D histograms

Each point represents a DNN trained with a data set with a specific benchmark, S and B

Low signal-to-background ratio → poor performance

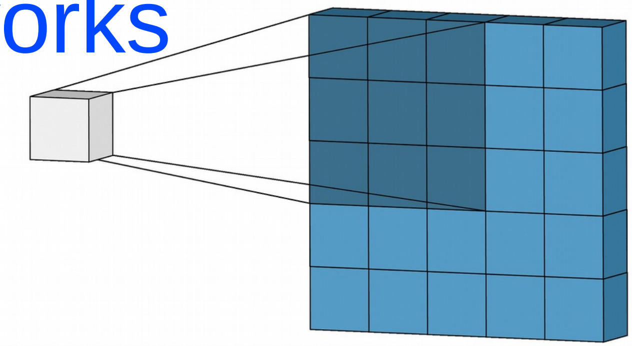


High signal-to-background ratio → excellent performance

Huge performance boost w.r.t. event-by-event representation!

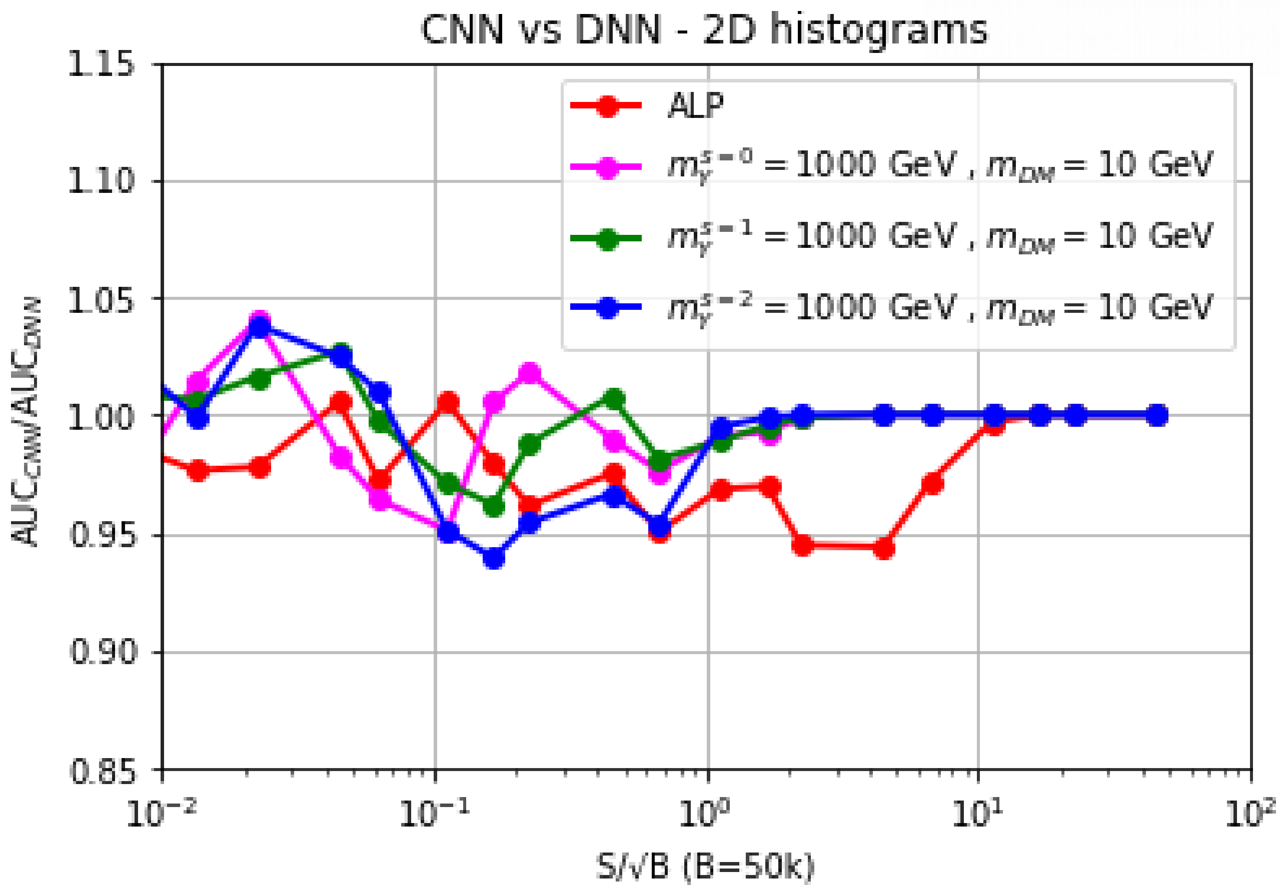
AUC=1 is a perfect classifier, and **AUC=0.5** represents a random classifier

CNN: Convolutional Neural Networks



- Image recognition.
- Have other structure (kernel layers, etc).
- Scan the image by parts looking for patterns.
- Demand significantly more computing time than DNN

$$\frac{AUC_{CNN}}{AUC_{DNN}} \rightarrow$$



Differences are within ∼5% for all the models considered

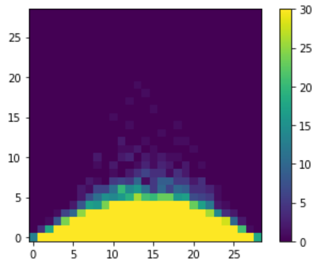
→ no improvement with respect to the DNN is found

Flexibility of the method

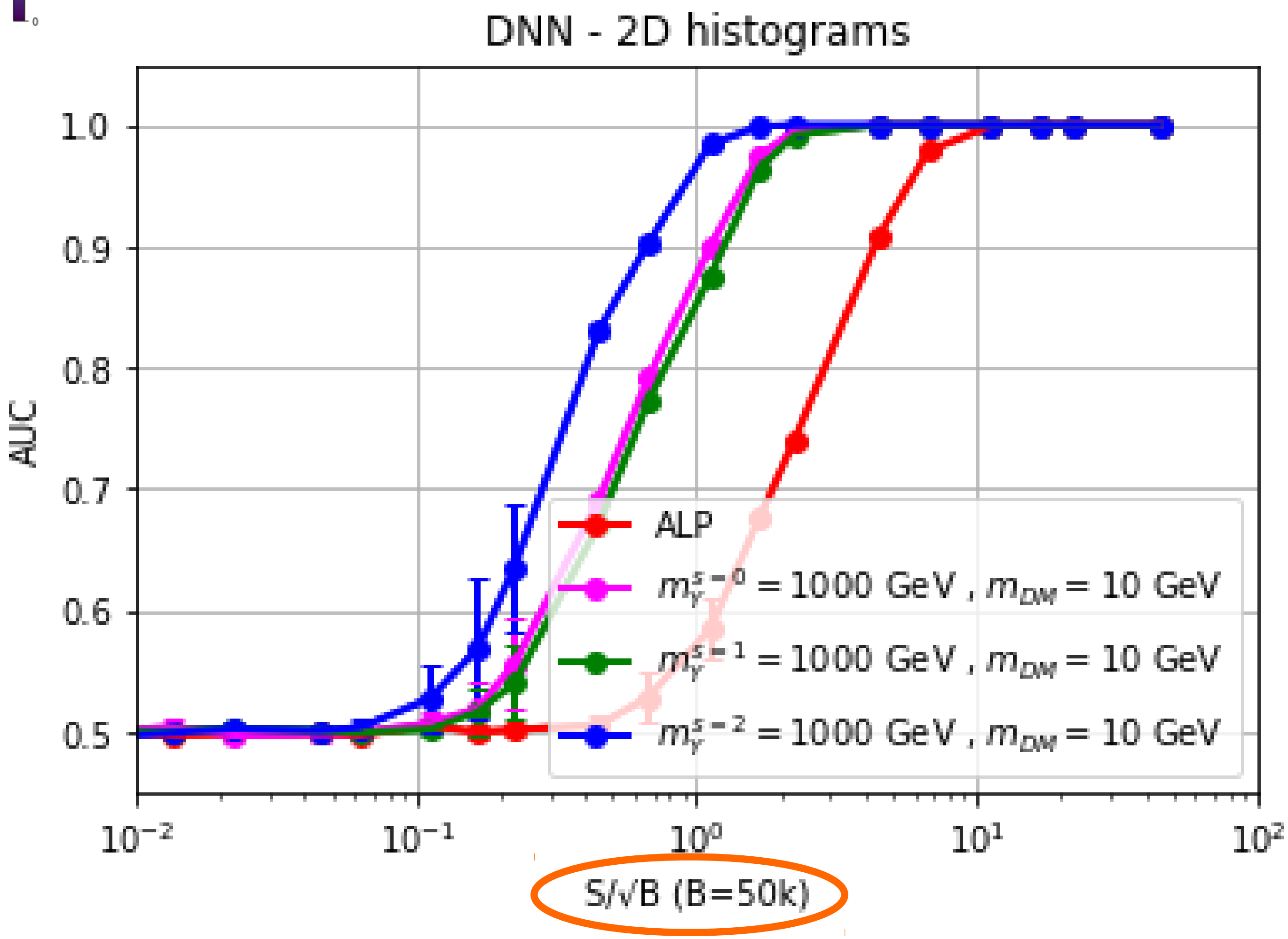
Performance invariance with the number of background events
Testing under incorrect training hypothesis: - coupling values
- benchmark model

S: # NP events
B: # SM events

Performance invariance with B

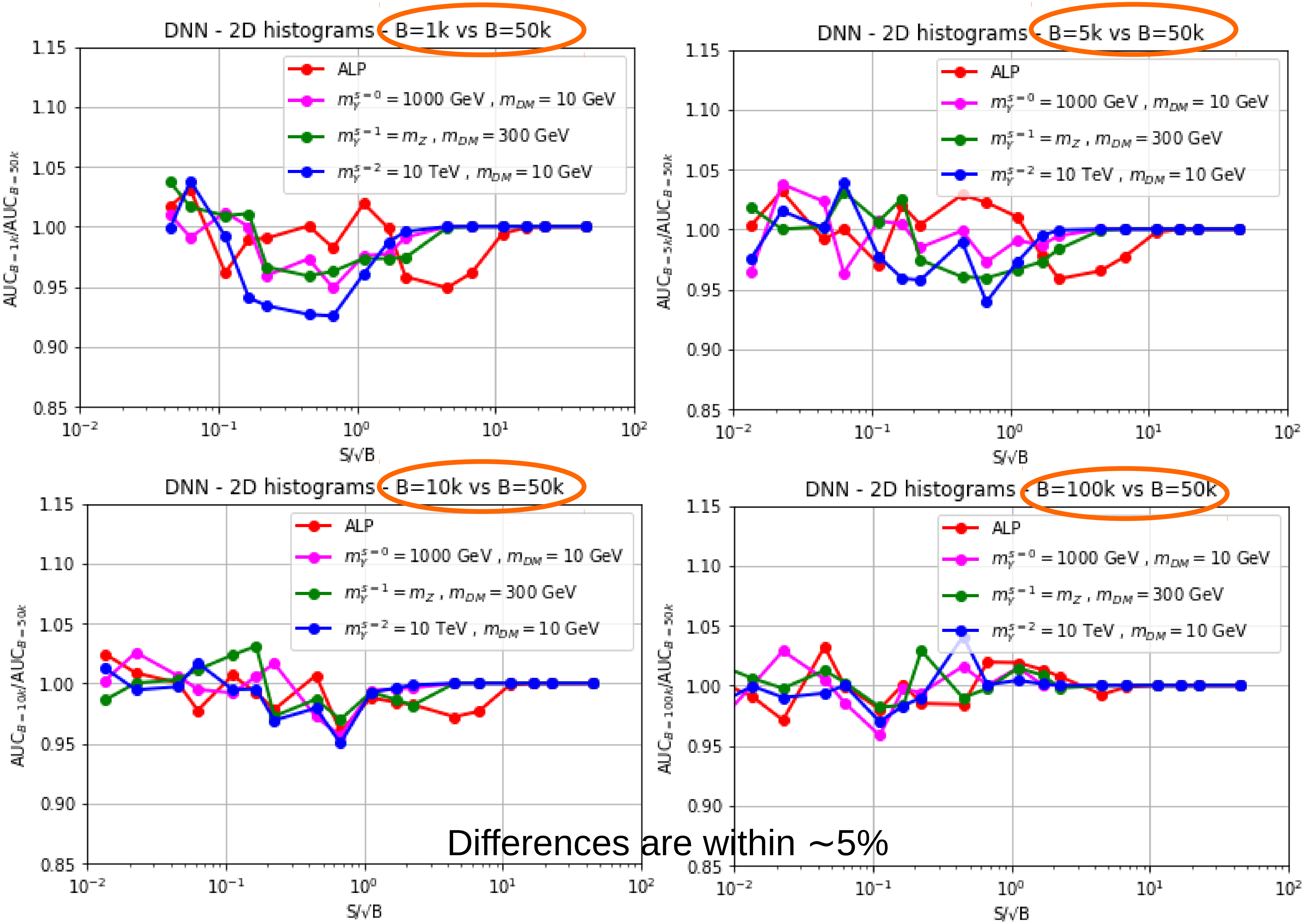


Before, each 2D histogram constructed with
B = 50k SM events + **S** New Physics events



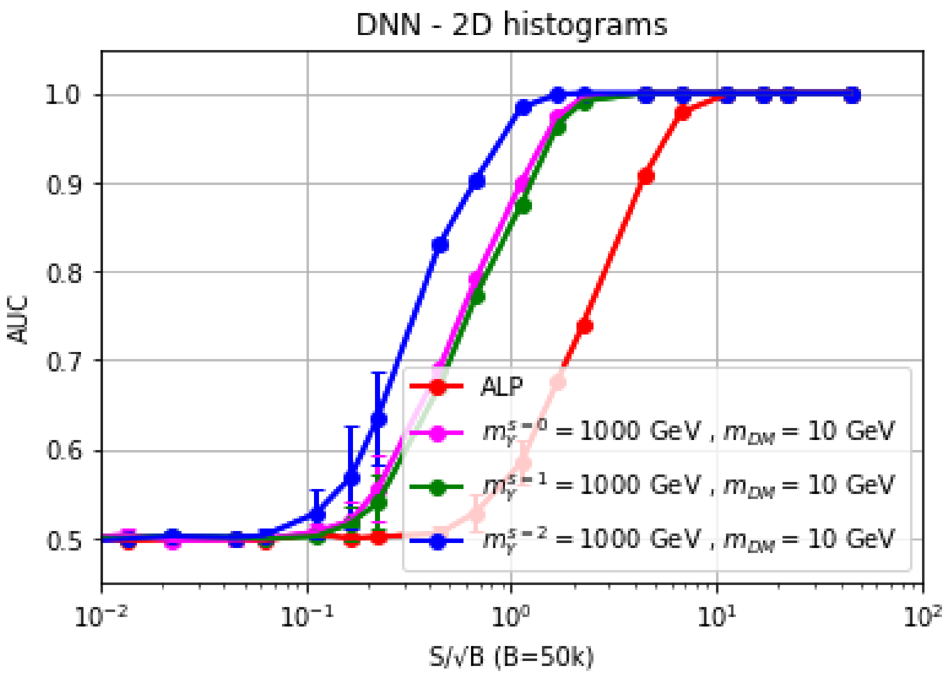
Performance is not modified significantly for different values of B, if the results are presented as a function of **S/√B**.

Performance invariance with B



Differences are within ~5%

Performance invariance with B



To know if a DNN or CNN with 2D histograms could distinguish a particular new physics model from the SM background, we only need to:

- **Identify the curve of the corresponding benchmark model**
- **Calculate the model cross section for the chosen couplings**
- **Calculate the SM background cross section**
- **Calculate S/√B for any luminosity, and check the corresponding AUC**

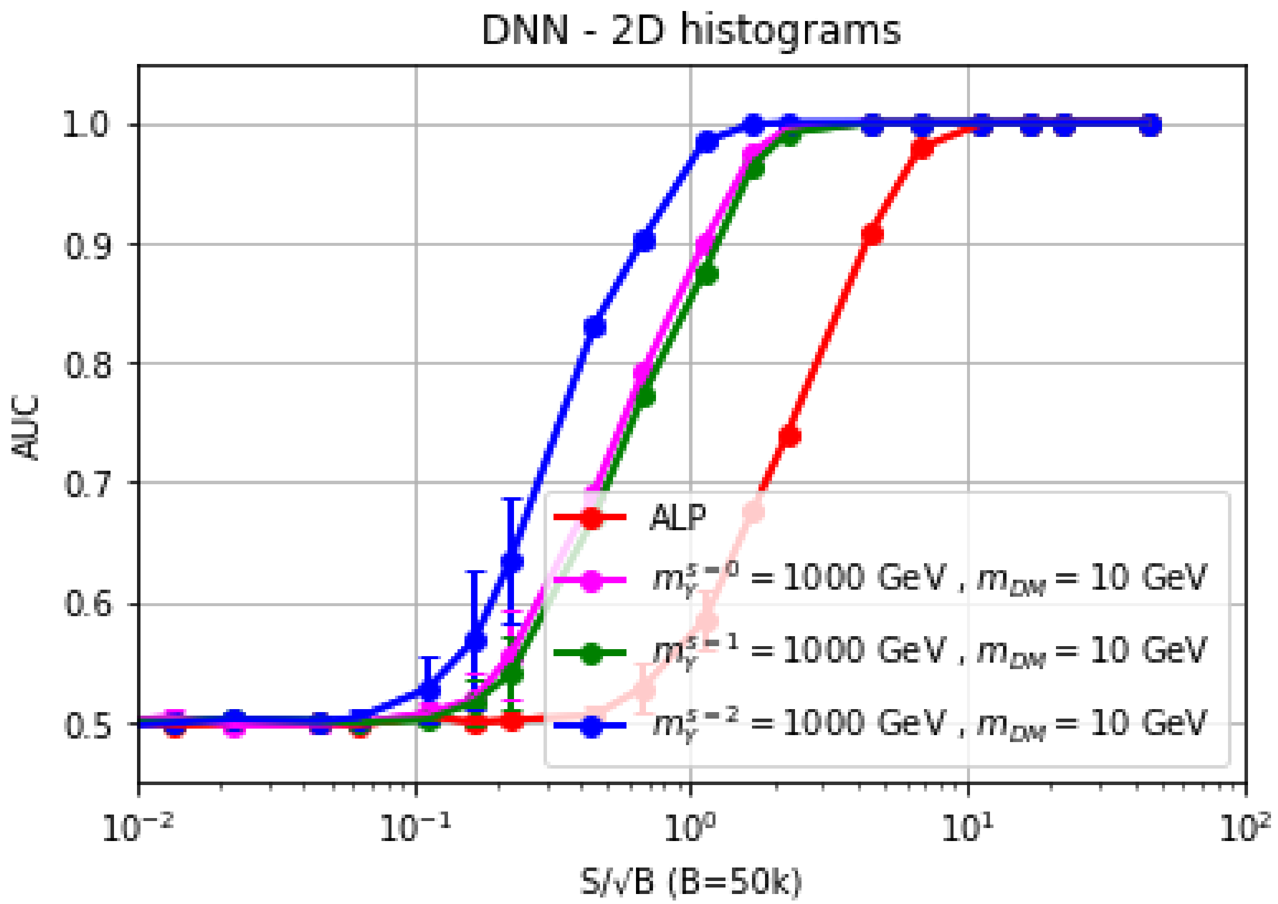
Also, we can have an idea of the luminosity needed to obtain a given efficiency. Change the last step for:

- **Identify the S/√B value for the corresponding AUC you would like to get and calculate the luminosity needed**

events = cross section * luminosity * detector efficiency

Performance invariance with B

S: # NP events
B: # SM events



S/√B is a histogram parameter

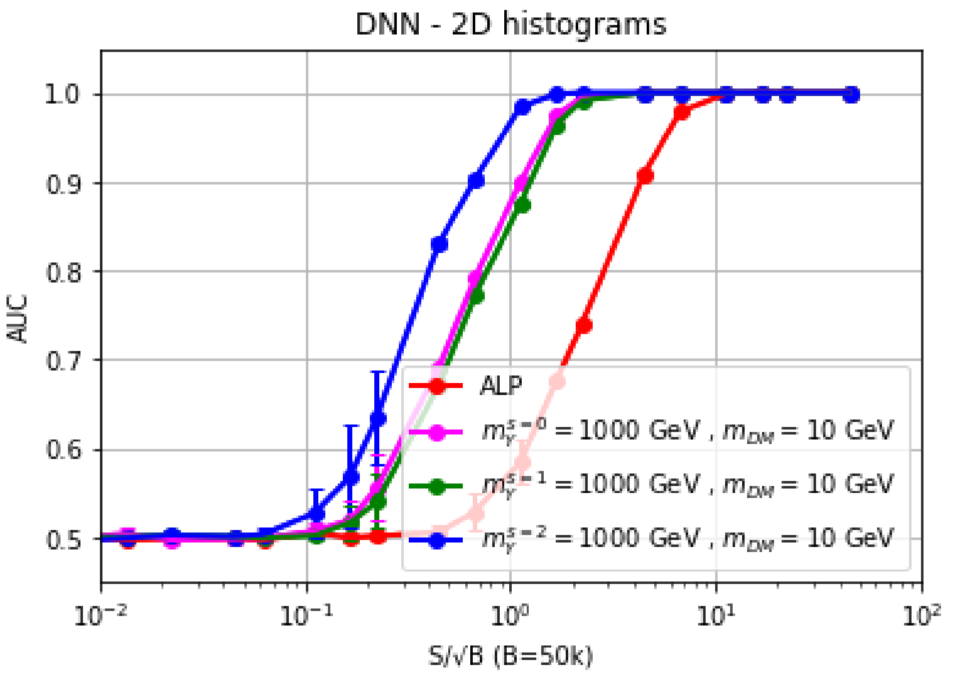
↓

each 2D histogram constructed with
B SM events + **S** New Physics events

It is not the significance in an usual counting experiment

(1 data sample = 1 histogram)

Preguntas abiertas



¿Cuál es la relación entre AUC y la significancia de excluir la hipótesis nula?

¿Cómo estimar la significancia del método?

¿Cómo es el proceso de aplicación del método?

¿Tiene sentido construir más de 1 histograma con los datos experimentales?

Disminuye S/√B → i.e. disminuye el AUC de la red, pero ¿tengo más estadística?

Testing under incorrect training hypothesis

A DNN is prepared to handle the same kind of data that it has been trained with, i.e. matching train and test data samples (same underlying model).

What is the performance when we apply a test data set to a DNN trained with a different underlying model?

DNN results with matching data samples are called AUC_{true}

DNN results with non-matching data samples are called AUC_{wrong}

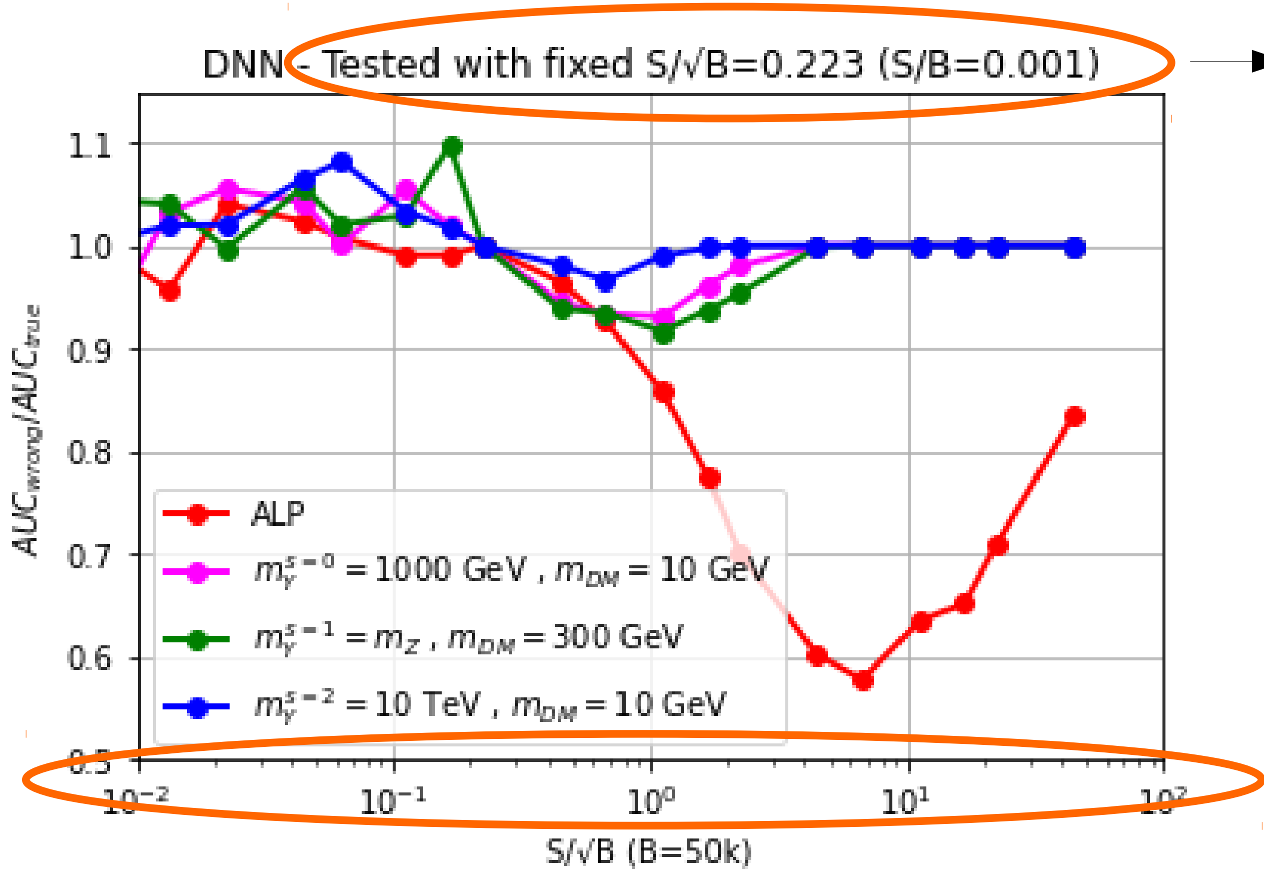
Testing under incorrect training hypothesis: ***coupling values***

Train and test data samples are generated with the **same benchmark models**, but with **different S/B ratios**, i.e. a same model with **different coupling values.**

Non-matching samples

$$\frac{AUC_{wrong}}{AUC_{true}}$$

Matching samples



Test data (from “collider”)

Low signal-to-background ratio

Train data: DNNs prepared for this signal-to-background ratio

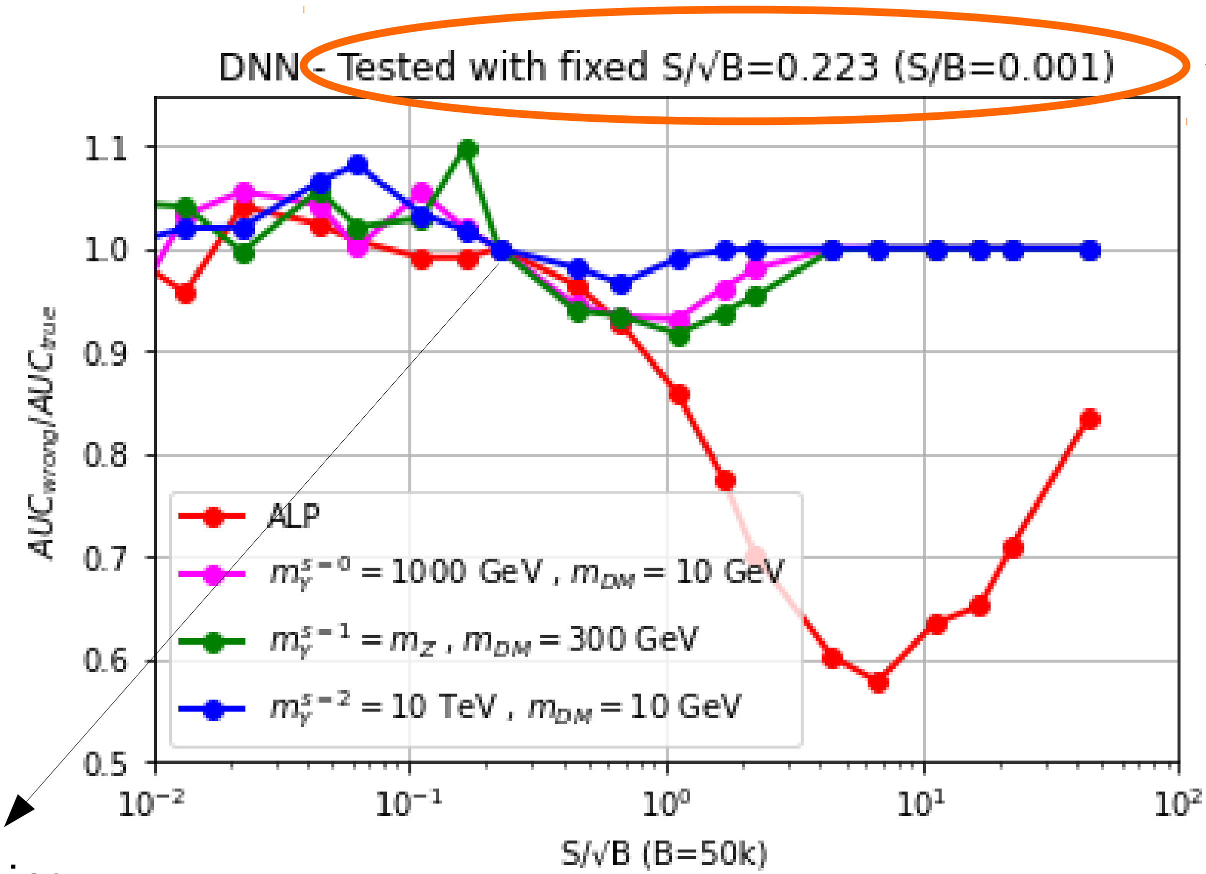
Testing under incorrect training hypothesis: ***coupling values***

Train and test data samples are generated with the **same benchmark models**, but with **different S/B ratios**, i.e. a same model with **different coupling values.**

Non-matching samples

$$\frac{AUC_{wrong}}{AUC_{true}}$$

Matching samples



Test data (from “collider”)

Low signal-to-background ratio

Matching test-train samples

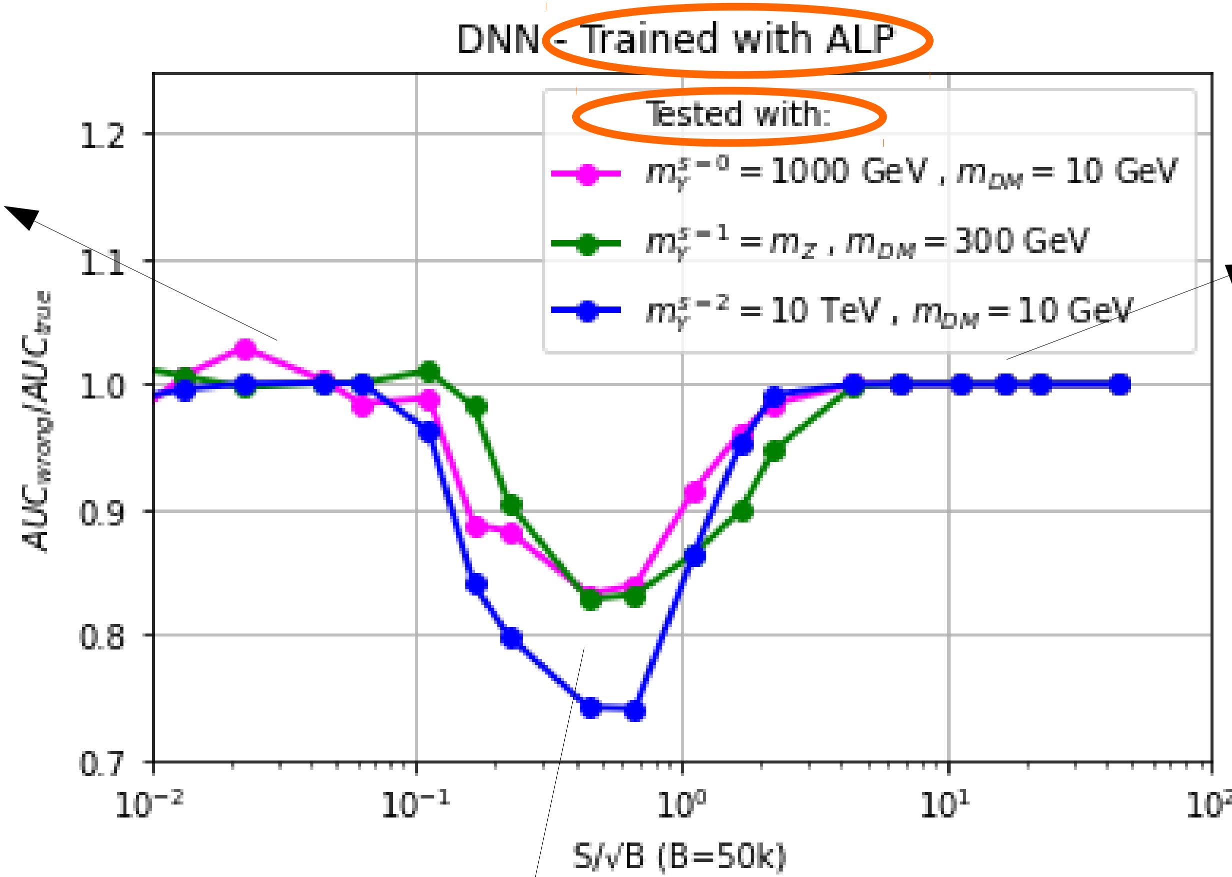
Around the matching S/√B, the DNN performance is close to the “true” DNN → some error or variation in the coupling values is acceptable

Testing under incorrect training hypothesis: ***benchmark model***

Train and test data samples are generated with **different benchmark models**, but with the **same S/B ratios**.

Low signal-to-background ratio

Can NOT discriminate NP vs SM regardless the underlying model.
“Few signal”



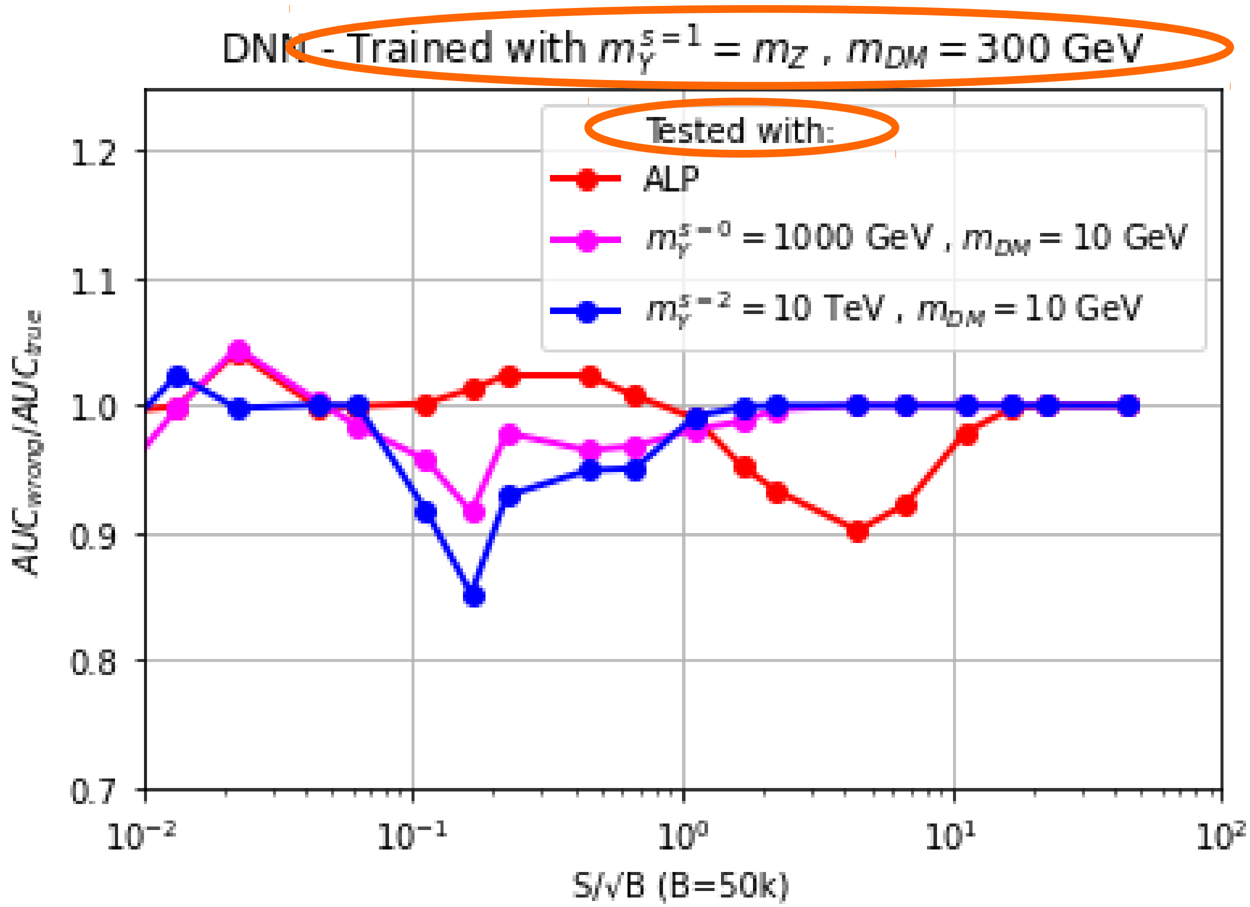
High signal-to-background ratio

Can discriminate NP vs SM regardless the underlying model.
“A lot of signal”

Intermediate S/√B values discrepancies can be huge, up to 25%

Testing under incorrect training hypothesis: ***benchmark model***

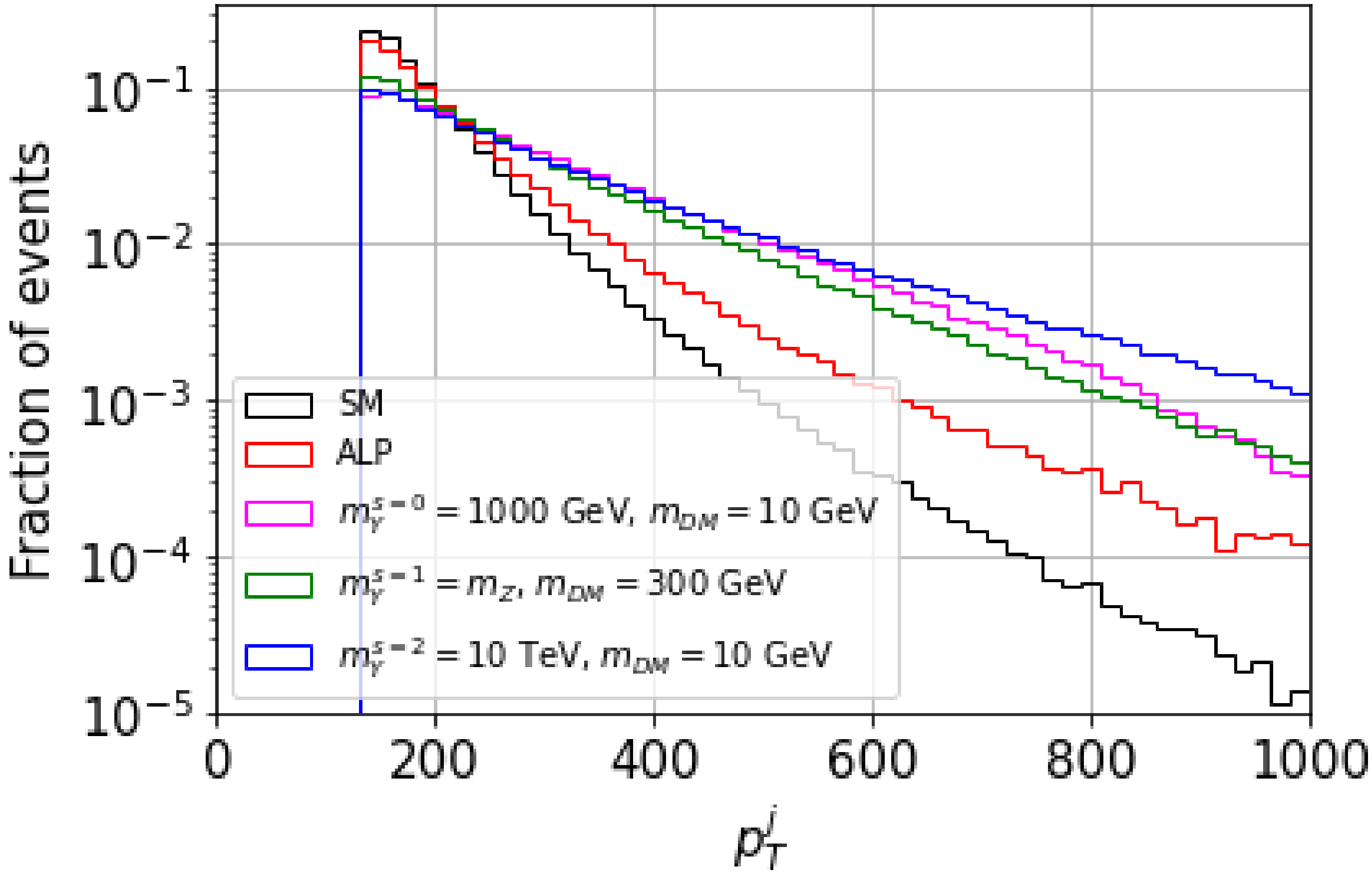
Another example



Performance variations within ~10%

The DNN classifies “kinematic distributions” not “models”

The DNN classifies “kinematic distributions” not “models”



Benchmark models with “similar” kinematic distributions result in “similar” DNN performances

Multimodel classifiers

Binary classifier
Multiclass classifier

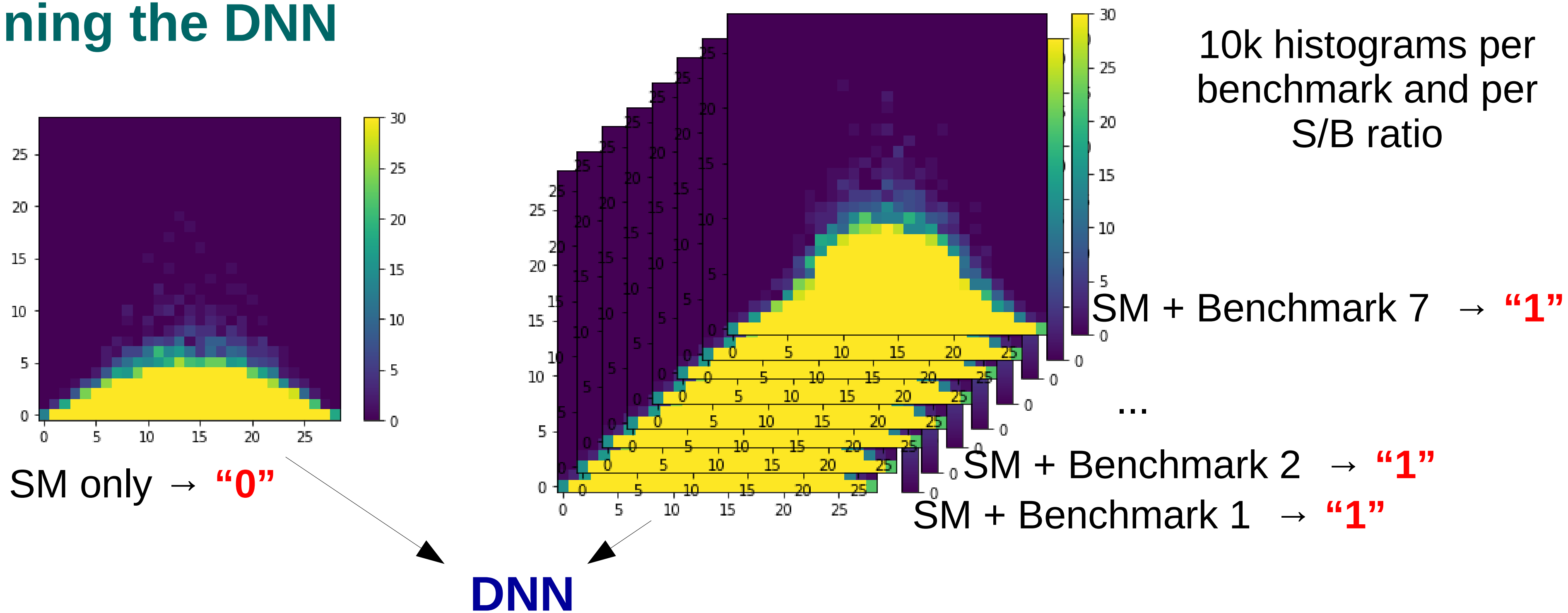
Binary classifier

A single DNN per S/B value to discriminate, regardless the underlying model, between

SM background (labeled '0') vs **non-SM processes** (labeled '1')

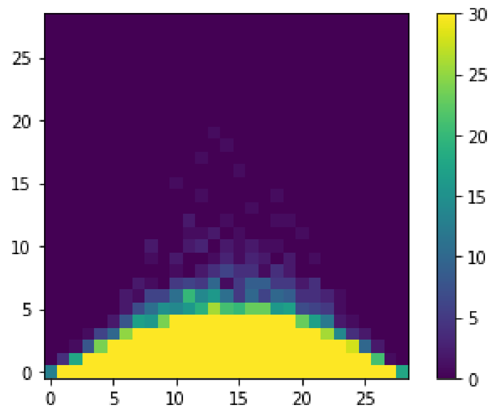
One DNN trained with several benchmark models

Training the DNN

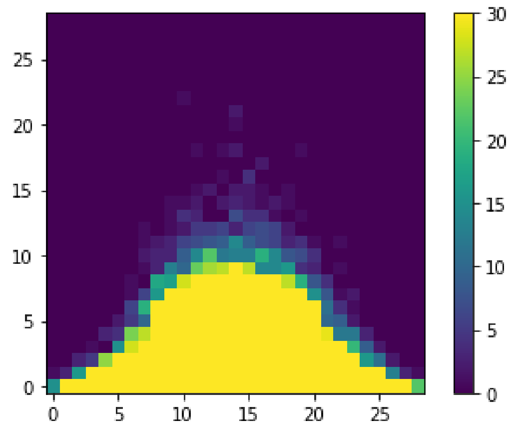


Binary classifier

Test the DNN



SM only → **"0"**



SM + Benchmark i → **"1"**

Test the network with only **one benchmark model at a time** to evaluate its performance, denoted **$AUC_{7\text{-models}}$**

DNN → Value between **[0,1]**

NP + SM
or
SM only

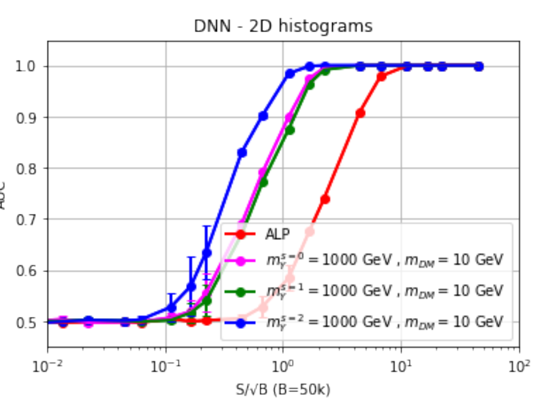
ROC curve → AUC as performance

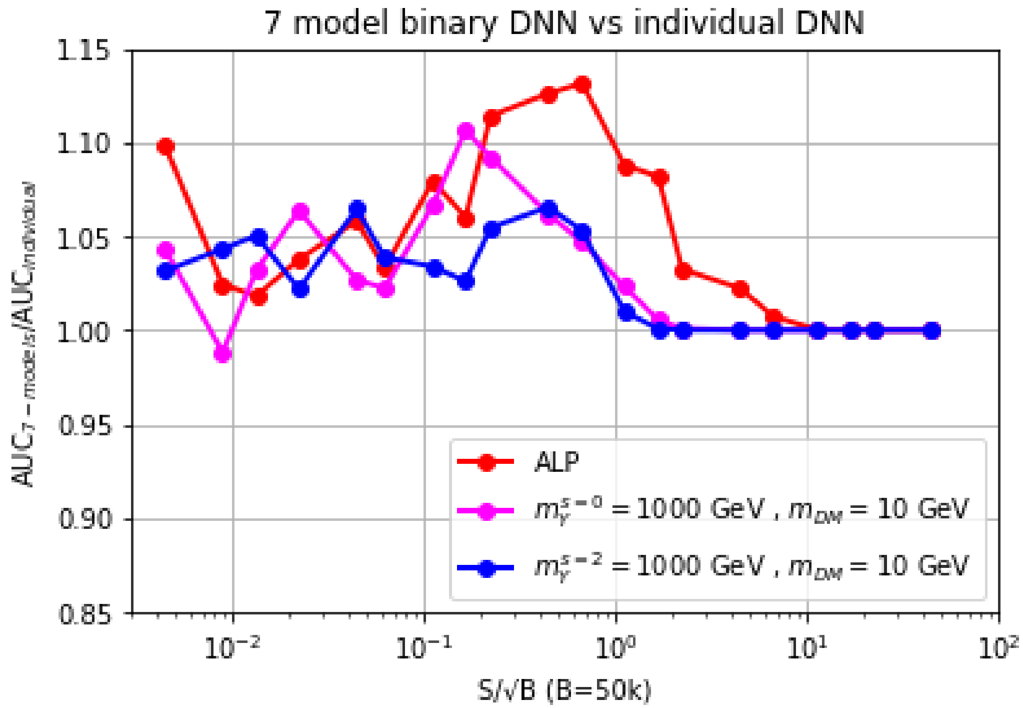
Binary classifier

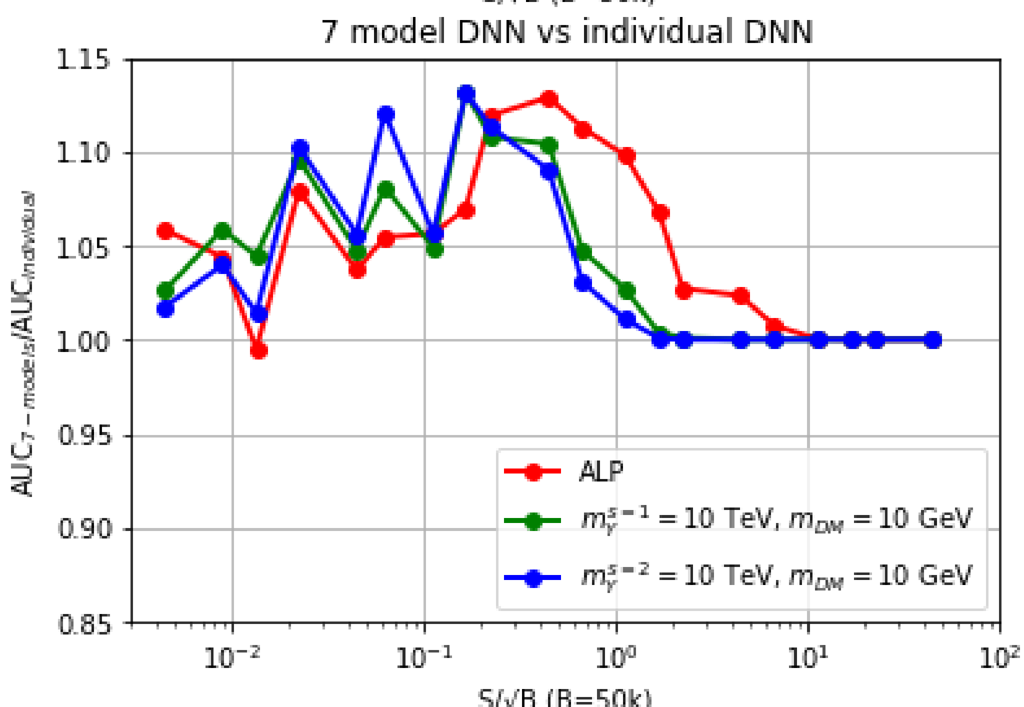
Trained with 7 benchmark models, tested with only one

$$\frac{AUC_{7\text{-models}}}{AUC_{individual}}$$

Trained and tested with only 1 benchmark model







Testing with training models

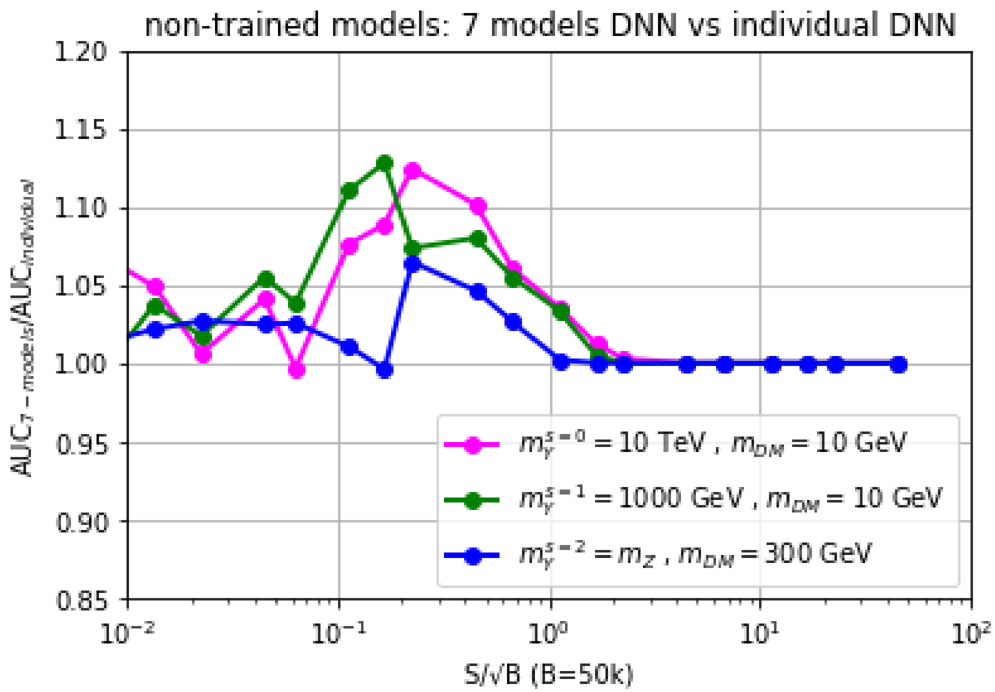
Some **improvement**, up to ~15%, w.r.t. individual DNNs, but in regions with **low AUC**

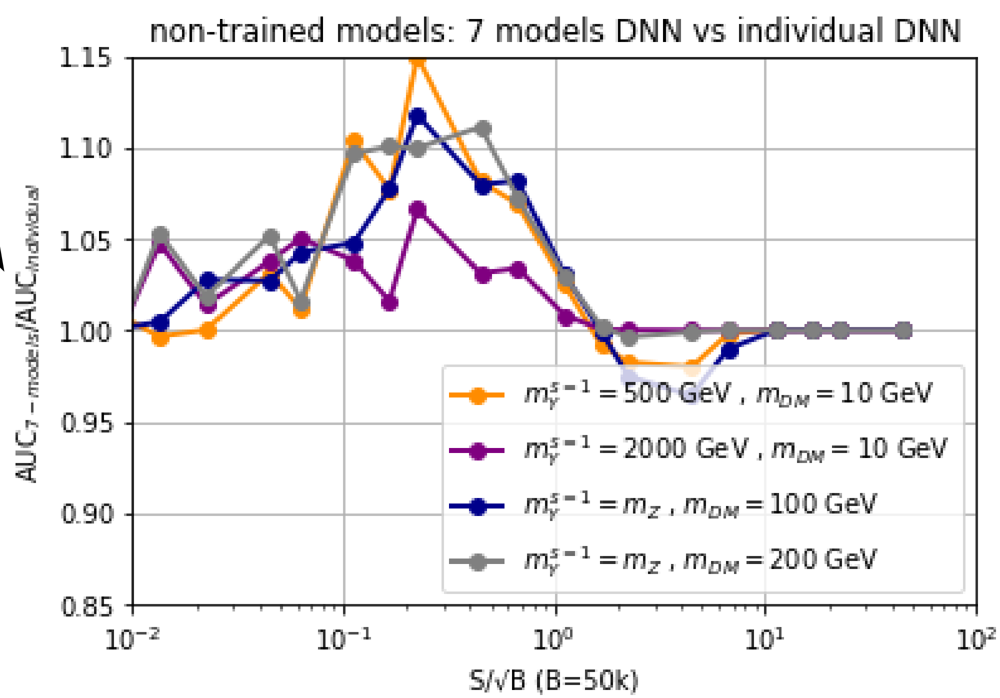
Binary classifier

Trained with 7 benchmark models, tested with only one

$$\frac{AUC_{7\text{-models}}}{AUC_{individual}}$$

Trained and tested with only 1 benchmark model





Models completely new to the DNN

Testing with non-training models

Some **improvement**, up to ~15%, w.r.t. individual DNNs, but in regions with **low AUC**

Info about many models → helps the discrimination of NP + SM vs SM only

Multiclass classifier

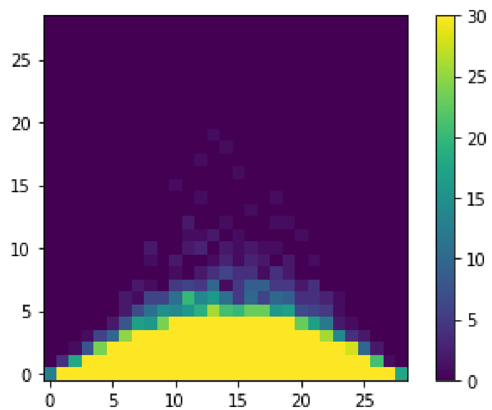
A single DNN per S/B value to discriminate between:

SM only vs **Benchmark 1 + SM** vs … vs **Benchmark N + SM**

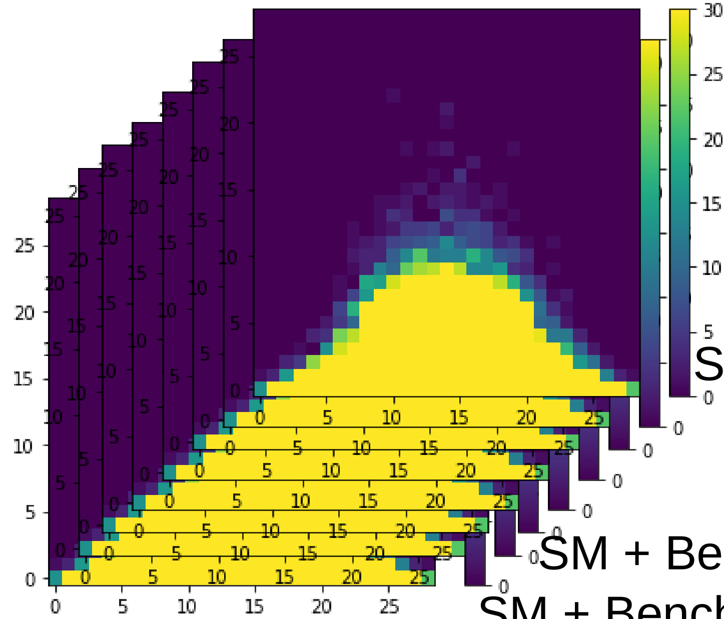
(labeled '0') **(labeled '1')** **(labeled 'N')**

a single DNN trained with **several** new physics models

Training the DNN



SM only → **"0"**



10k histograms per benchmark and per S/B ratio

SM + Benchmark 7 → **"7"**

...

SM + Benchmark 2 → **"2"**

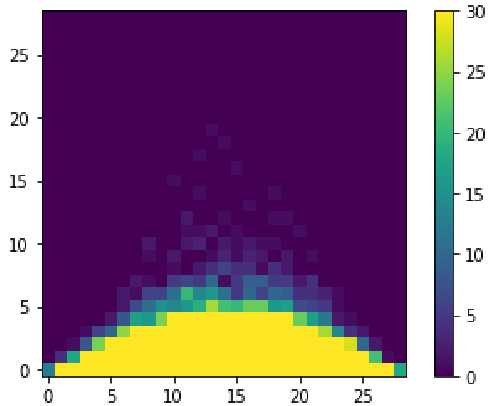
SM + Benchmark 1 → **"1"**

DNN

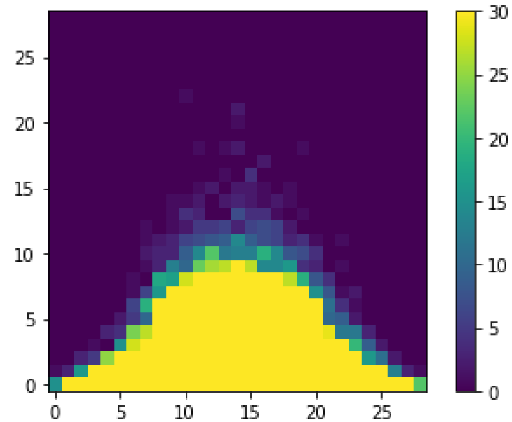
(In this work, 7 NP+SM models and SM only)

Multiclass classifier

Test the DNN



SM only → **"0"**



SM + Benchmark i → **"1"**

Test the network with only **one benchmark model at a time** to evaluate its performance, denoted **$AUC_{7\text{-models}}$**

(trained with **7 NP+SM** models and **SM only**)

DNN

Array of 8 elements between **[0,1]**

Each element is the probability that the histogram belongs to a training model

For example

Label	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Array	[0.09	0.03	0.01	**0.84**	0.005	0.005	0.01	0.01]

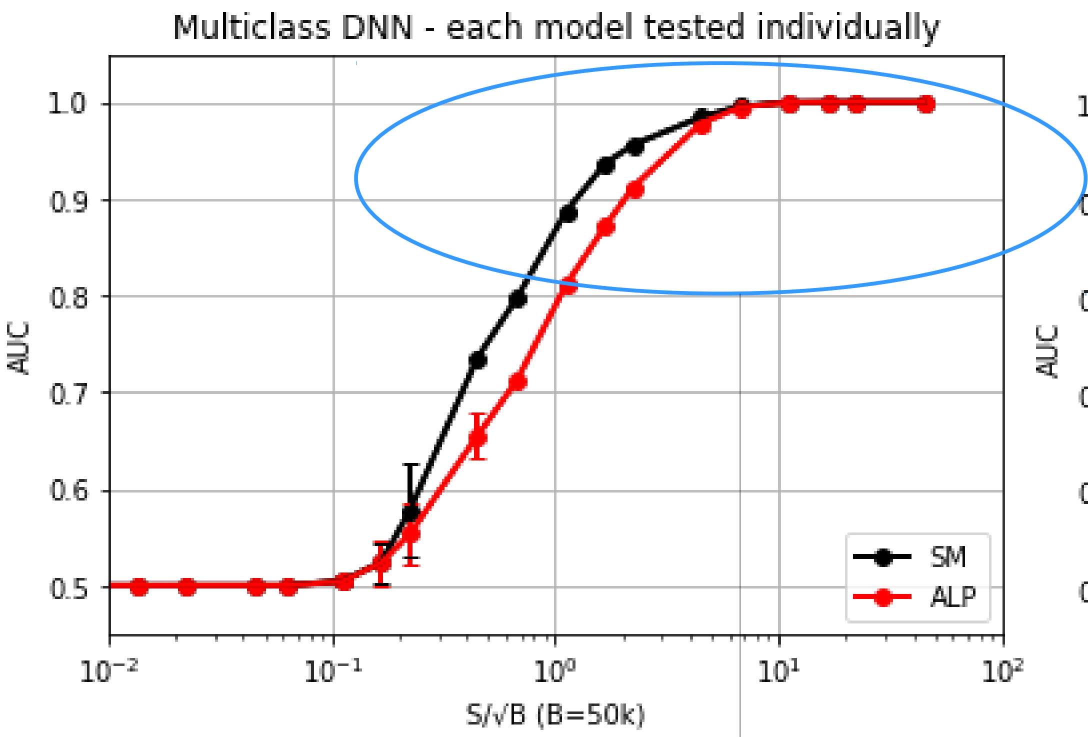
Prob(SM)

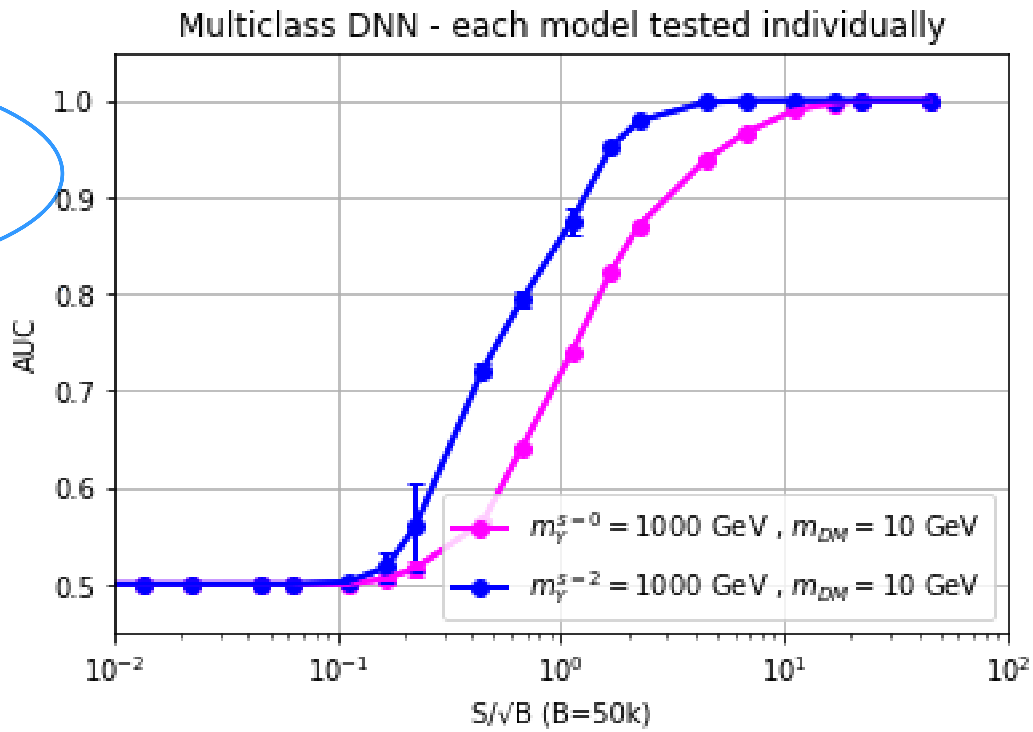
Prob(benchmark model 3 + SM)

Multiclass classifier

Convert the output array in a **binary result**:
positive class → the element corresponding to the model we are testing,
negative classes → the other elements

The discrimination power is between a particular model and the rest of the 7 selected models





Can discriminate **SM only** histograms
(or **ALP + SM** histograms) from the rest

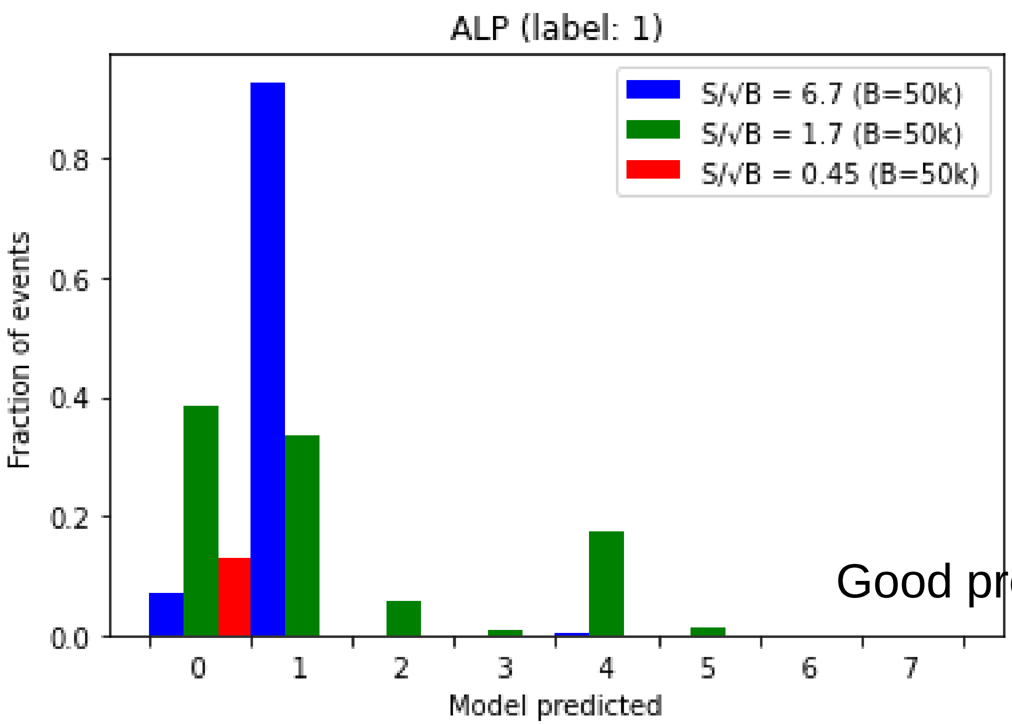
Multiclass classifier

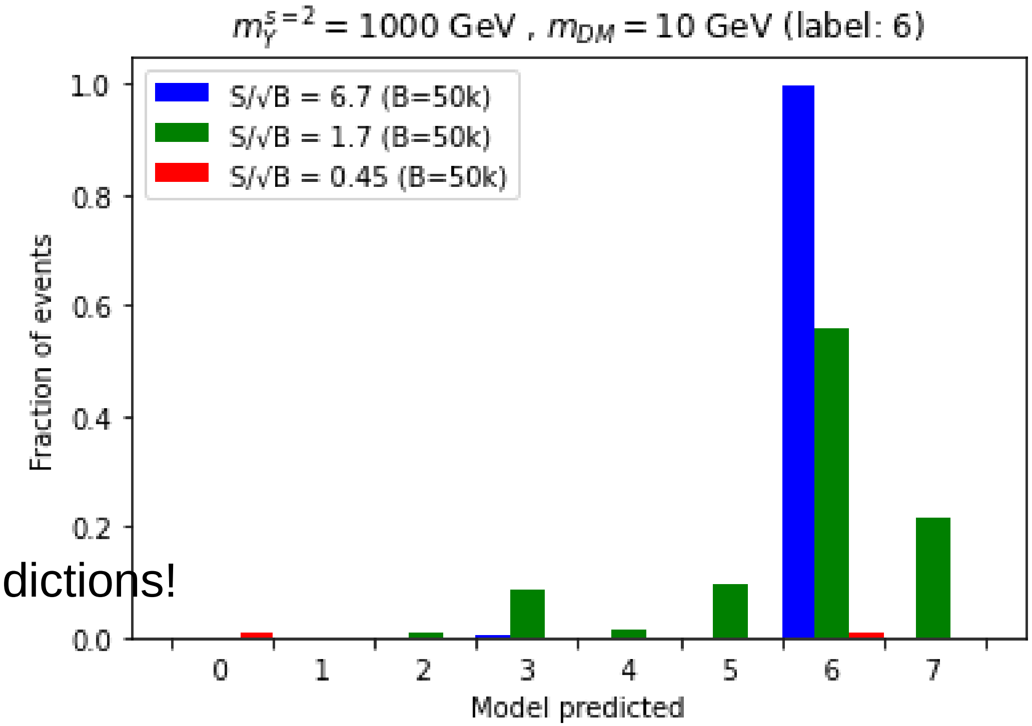
One count is assigned to the output array element if its probability fulfills two conditions:

1. it is the **element with the highest value**,
2. its **value is above** a threshold equal to **0.25**, defined as two times the probability that would be obtained for a completely random classifier.
(there are 8 training models → 2 * 1/8)

Then, a histogram of the frequency of occurrence can be constructed

Testing with training models



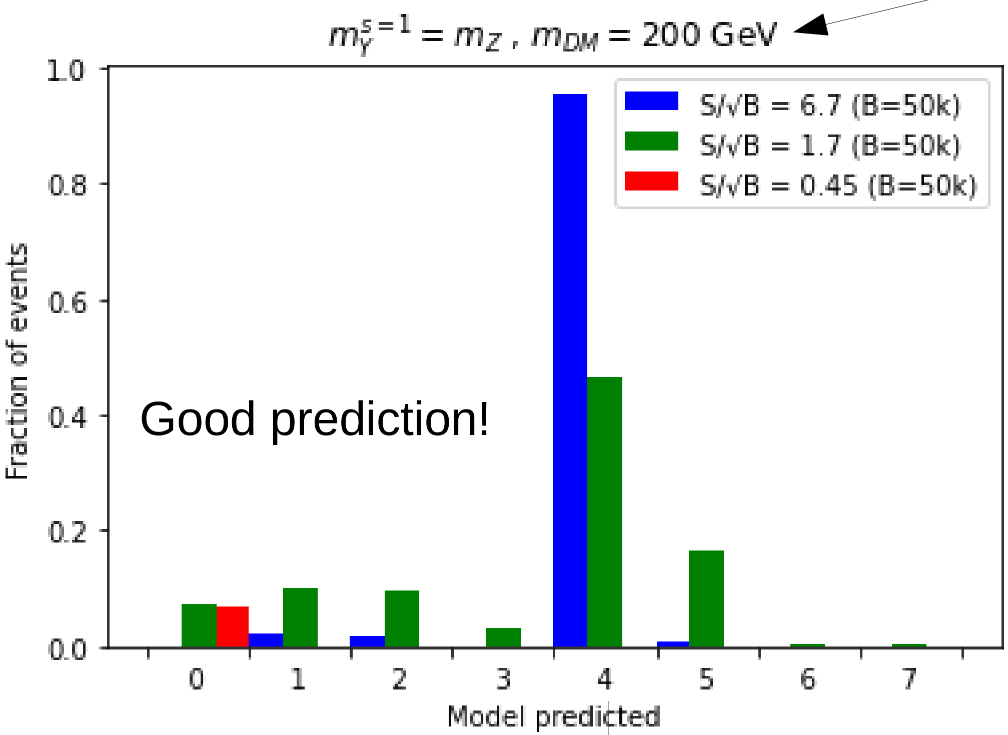


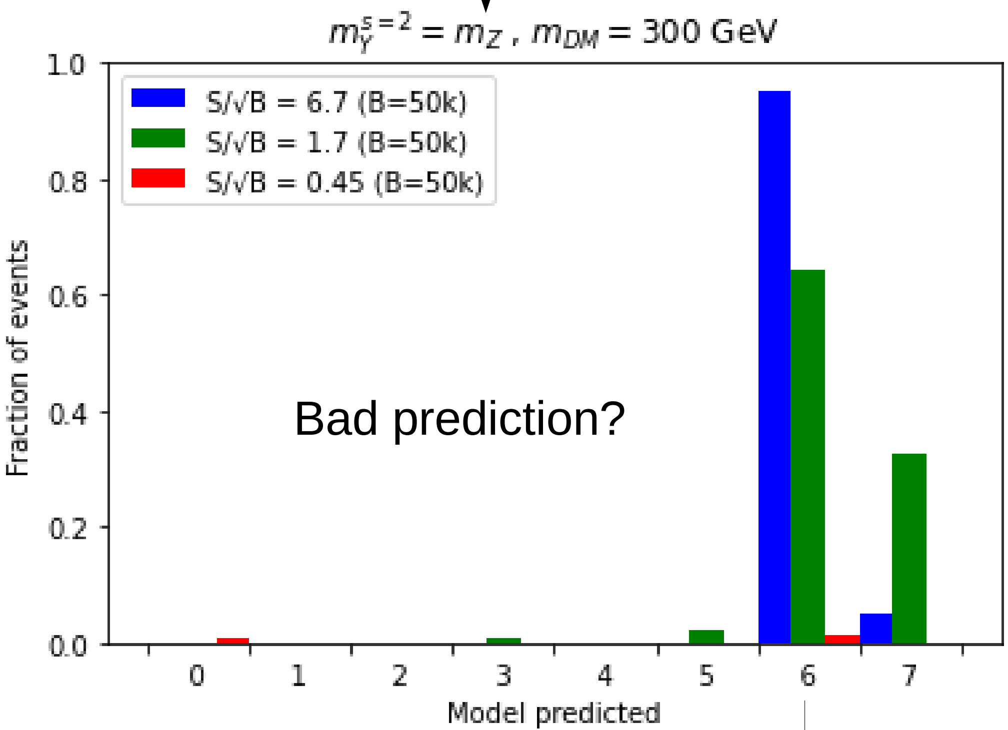
Good predictions!

Multiclass classifier

Testing with non-training models

Models completely new to the DNN





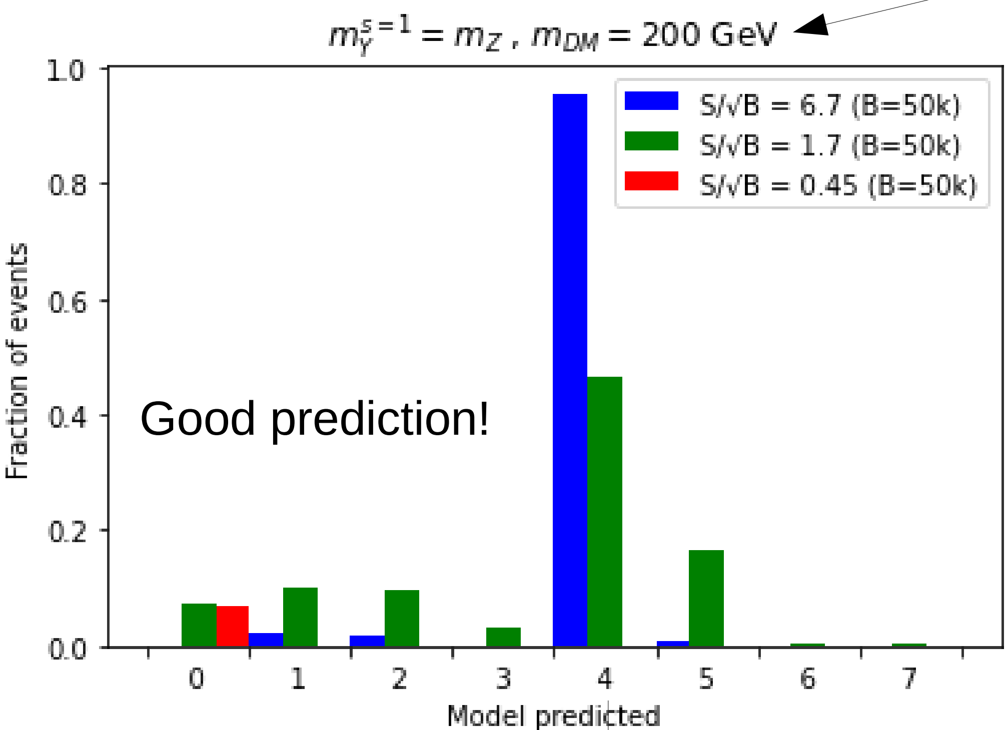
$m_Y^{s=1} = m_Z$, m_{DM}=300 GeV

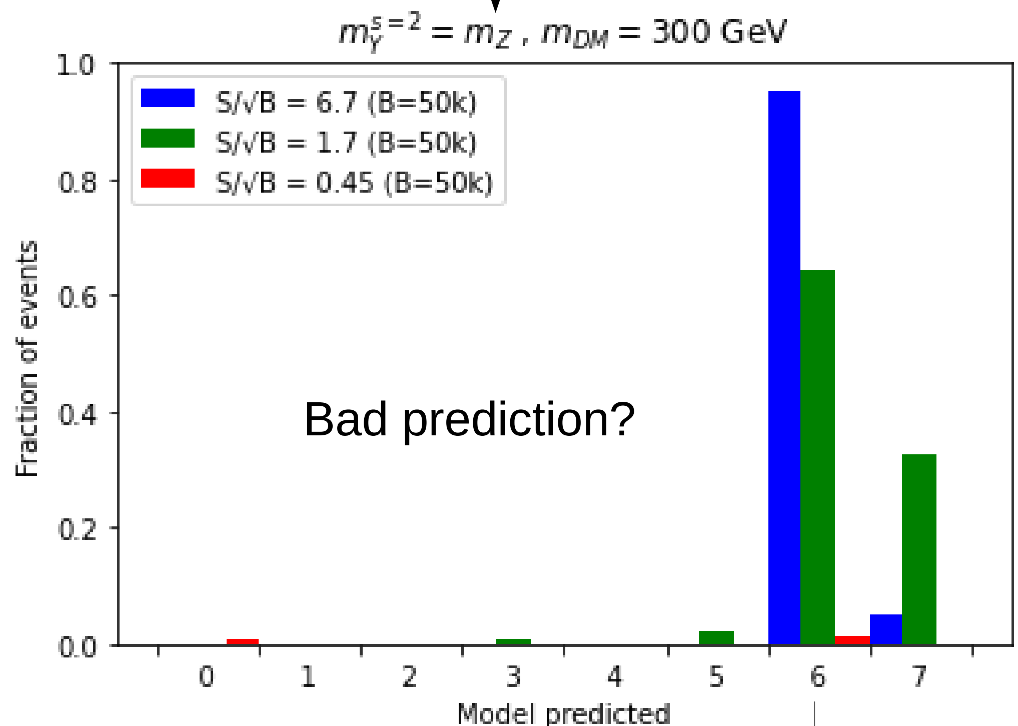
$m_Y^{s=2}$=1000 GeV , m_{DM}=10 GeV

Multiclass classifier

Testing with non-training models

Models completely
new to the DNN





$m_Y^{s=1} = m_Z$, m_{DM}=300 GeV

$m_Y^{s=2}$=1000 GeV , m_{DM}=10 GeV

The DNN classifies "kinematic distributions" not "models"

Predicts compatible kinetic distribution of the underlying model.

Conclusions

Conclusions

Models:

- Monojet plus missing transverse energy channel of four simplified dark matter frameworks:
  ALP and spin-0, spin-1, and spin-2 mediator models

- One usual drawback of supervised techniques: the need of a specific data set per model
  We take **kinematic features as input data**
  - *Independent of the coupling values*
  - In models with a mediator, *considering its status is key to reduce the free parameters*
  → **we describe a family of models with a single data set**

Neural Networks (individual classifiers):

- Discerning new physics signatures from SM background, two data representations:
  → event-by-event data → poor performance
  → **2D histograms** → **huge performance boost**

- CNN no significant improvement w.r.t. DNN results

- The **method is quite robust**.
  - The DNN can handle small errors in the coupling values,
  - and test an incorrect model as long as their kinematic distributions are similar,

Conclusions

- Histogram approach drawback: construct histograms for every S/B ratio
  → DNN performance independent of the number of background event
  if we choose $S/\sqrt{B}$ as variable

Easy to check if a DNN could discriminate a particular model from the SM, for any luminosity. Or to **estimate the luminosity needed** to achieve a certain performance level.

Multimodel classifiers:

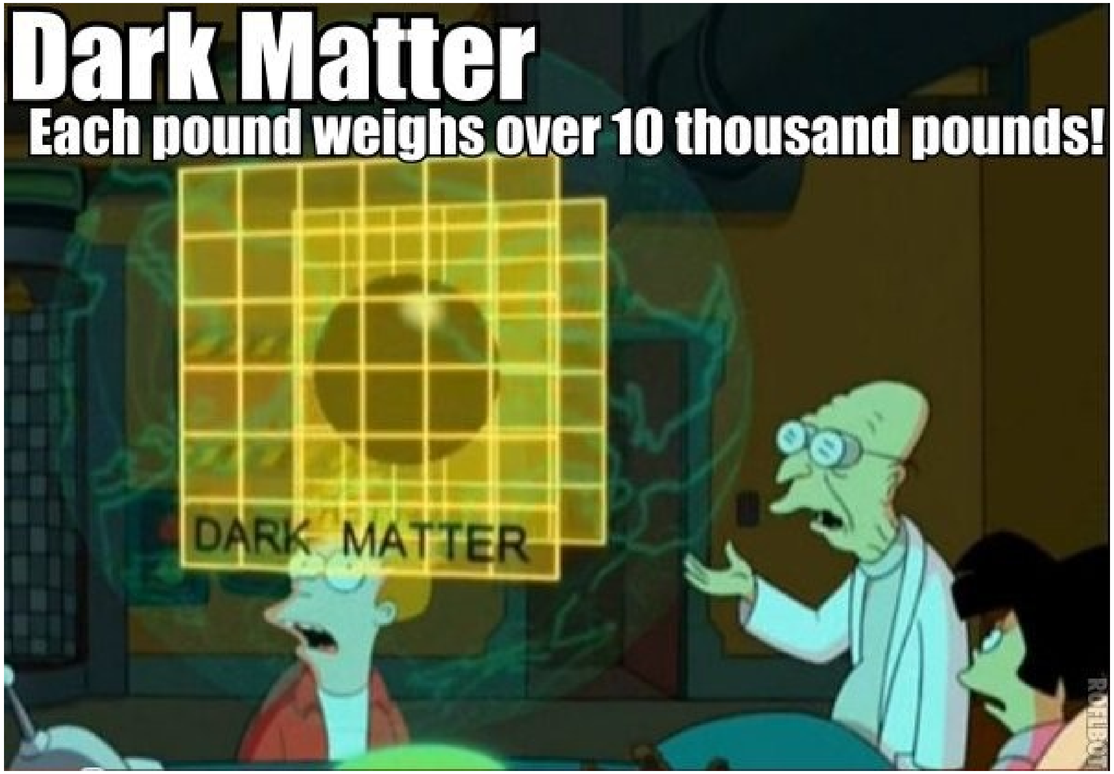
- Supervised algorithms but instead of training one benchmark model per DNN, several ones are used.
  - a **multimodel binary classifier**, prepared to discriminate between:
    SM only histograms vs SM background plus any kind of new physics events

  - a **multiclass classifier** prepared to identify between:
    SM only and several benchmark models, pointing out the most likely underlying model

- A more challenging task, but a good performance is achieved.

- misleading model properties can be predicted (e.g. incorrect dark sector masses or spin).
- But result points towards a compatible kinetic distribution, a key tool to guide further analysis

Thank you!


Dark Matter
Each pound weighs over 10 thousand pounds!
DARK MATTER

Back-ups

NNs
parameters

	DNN	CNN
Input data		
Format	event-by-event	2D histogram
Features	(p_T^j, η^j, ϕ^j)	p_T^j vs η^j
Sample size	400k per model	20k histograms per model
Train:Test:Validation	0.64:0.20:0.16	
Neural Network		
Layers	3 dense layers Hidden layers nodes = 20 Dropout in every hidden layer: 0.2	Convolutional 2D layer (3×3, 32) Max pooling 2D layer (2×2) Convolutional 2D layer (3×3, 64) Max pooling 2D layer (2×2) Flatten layer Dense layer
Activation function	Hidden layers: *relu* Output layer: *sigmoid*	
Compilation		
Loss function	*binary_crossentropy*	
Optimizer	*adam* (initial learning rate = 0.0001)	
Metric	*accuracy*	
Batch size	128	
Max epochs	1500-2500	
Patience	100-300 epochs	

SM background

Process	σ (pb)	D	PRC
$Zj\,(Z \to \nu\bar{\nu})$	53.6	0.78	1
$Wj\,(W \to \tau\nu)$	24.8	0.20	0.12
$Wj\,(W \to l\nu),\ l = e, \mu$	49.7	0.049	0.058
$Zj\,(Z \to ll),\ l = e, \mu, \tau$	19.1	0.013	0.006
$t\bar{t}$	217	0.021	0.11
		0.014 ($b_{tag} \leq 1$)	0.073
		0.004 ($b_{tag} = 0$)	0.021
diboson (WW, WZ, ZZ)	5.18	0.12	0.014

Same diagrams replacing
Z → W
ν → l

$$PRC = \frac{N_{\text{process}}}{N_{pp \to Zj(Z \to \nu\bar{\nu})}},$$

$$N_{\text{process}} = L\,\sigma_{\text{process}}\,D_{\text{process}},$$

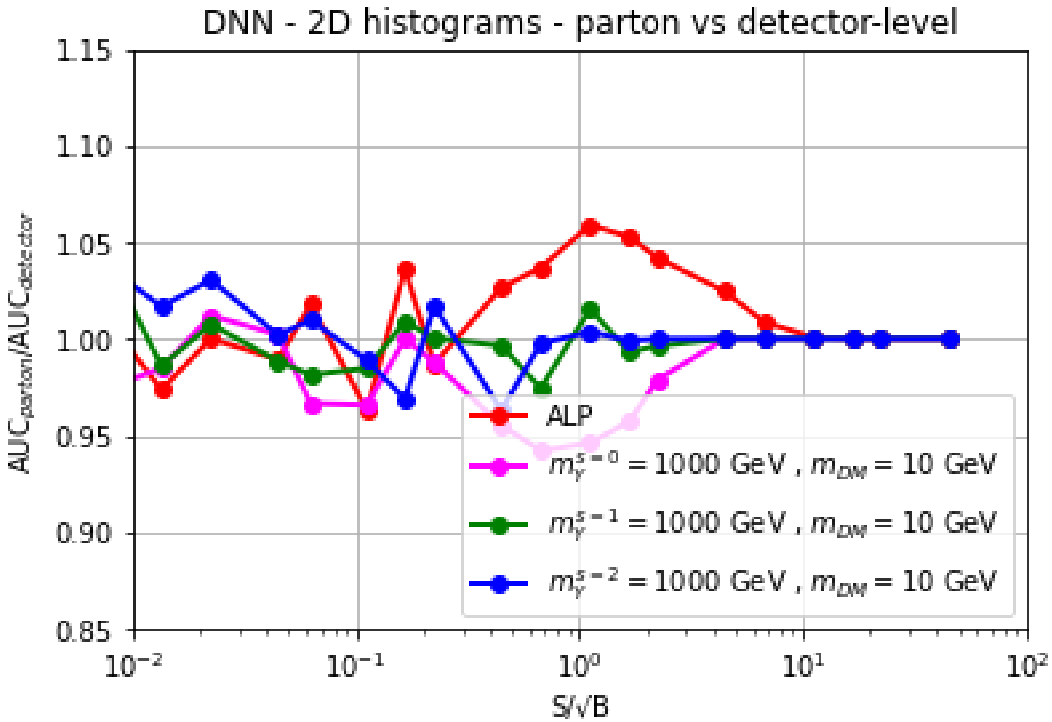
number of events

luminosity

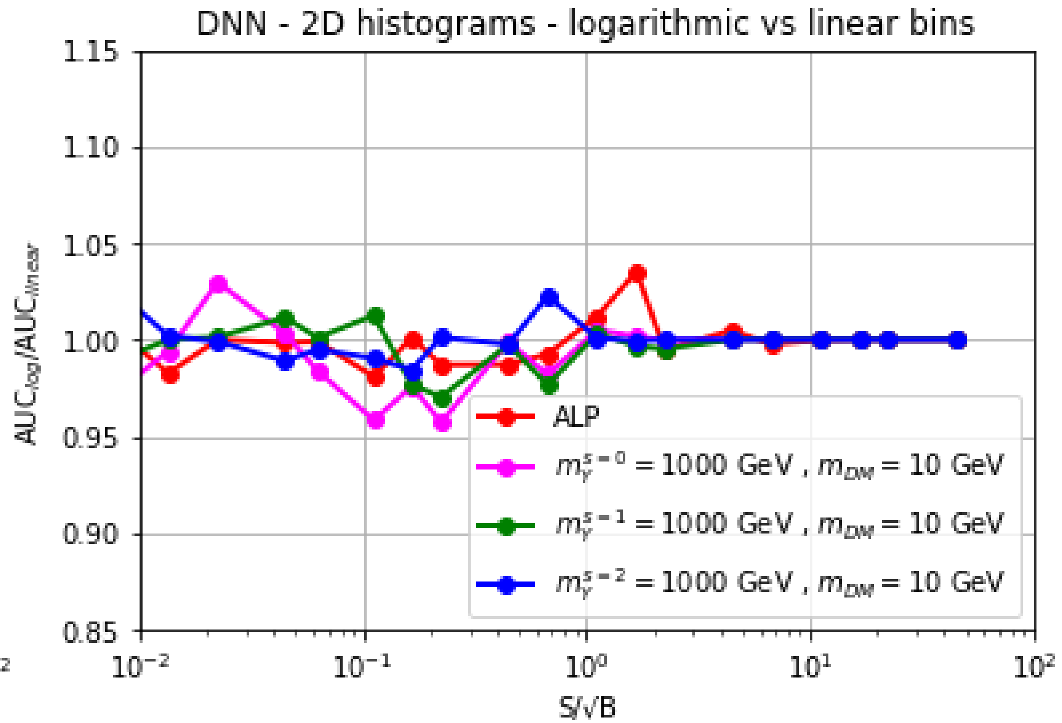
cross section

fraction of events identified as jets + missing E_T at detector level

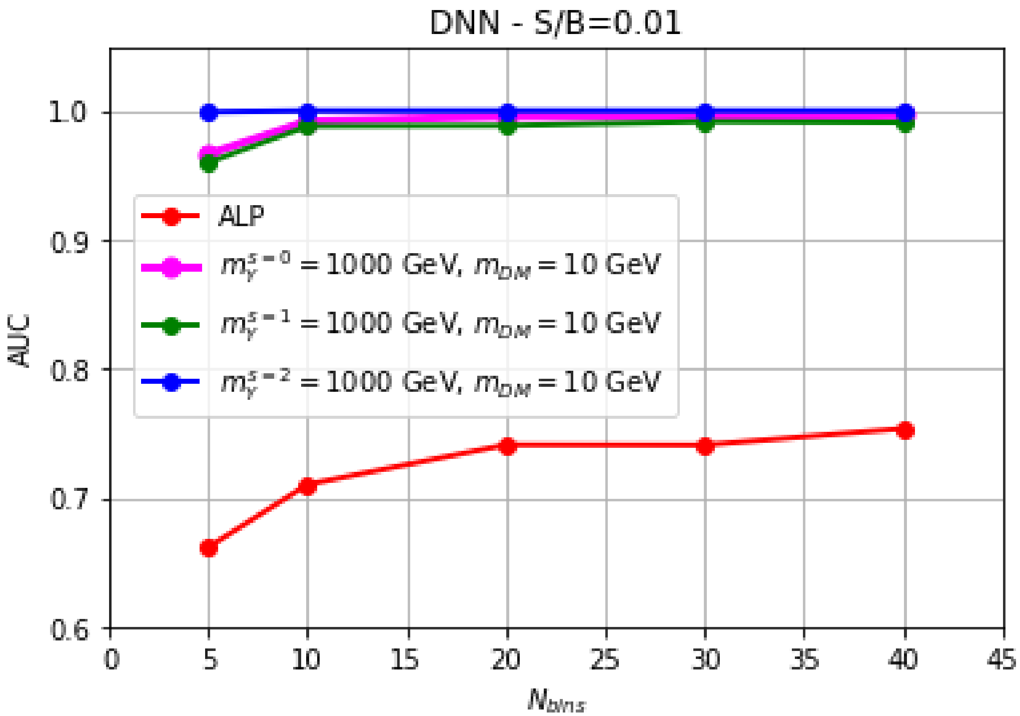
Parton vs detector-level data

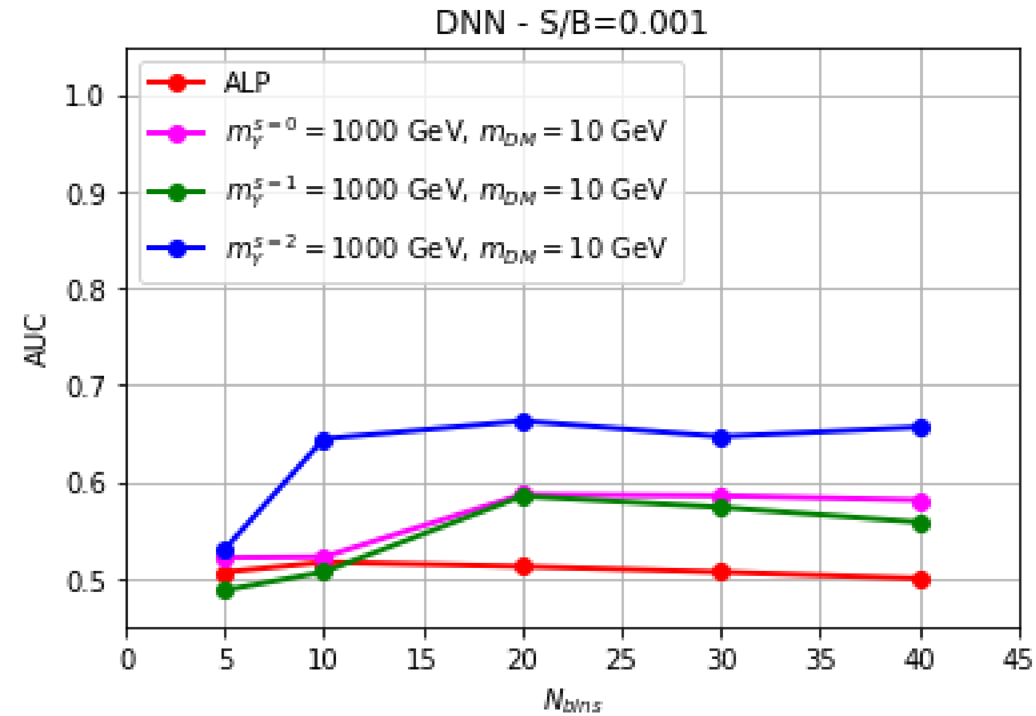


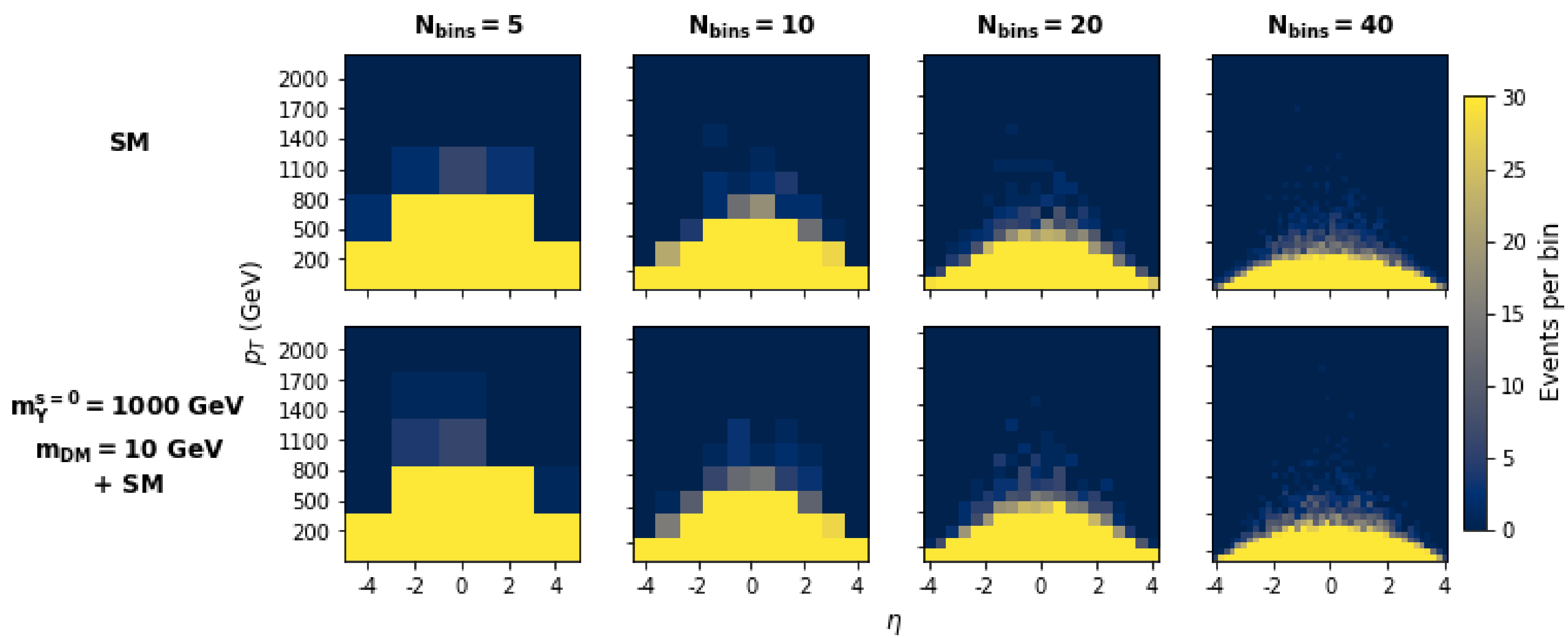
Logarithmic vs linear bins



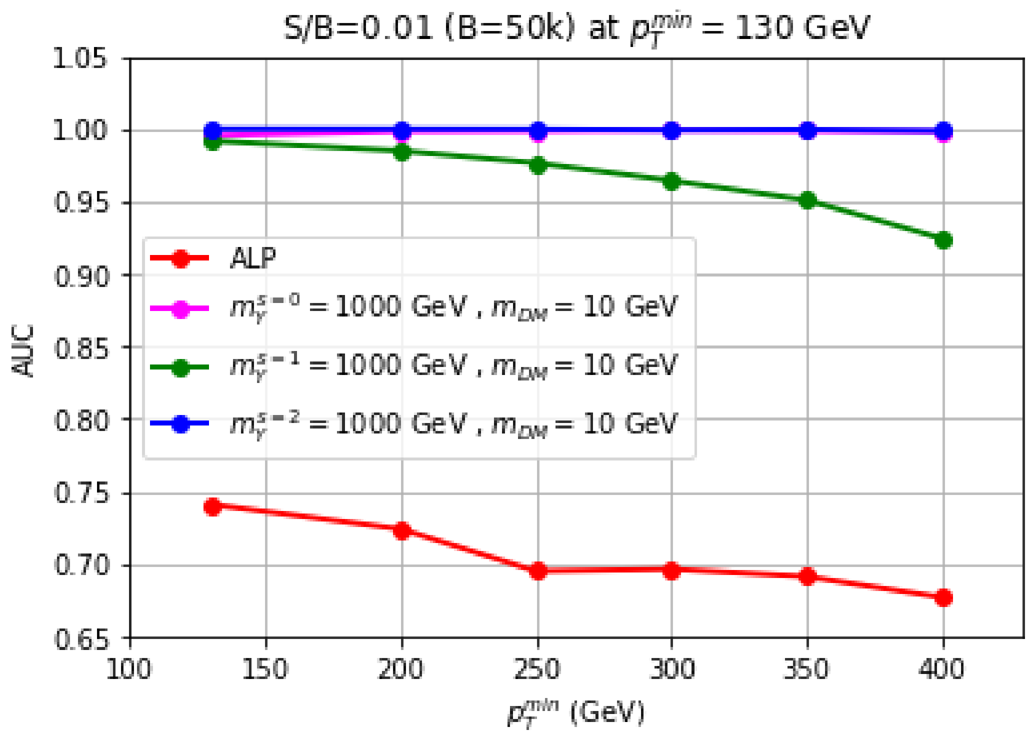
Bin number per histogram

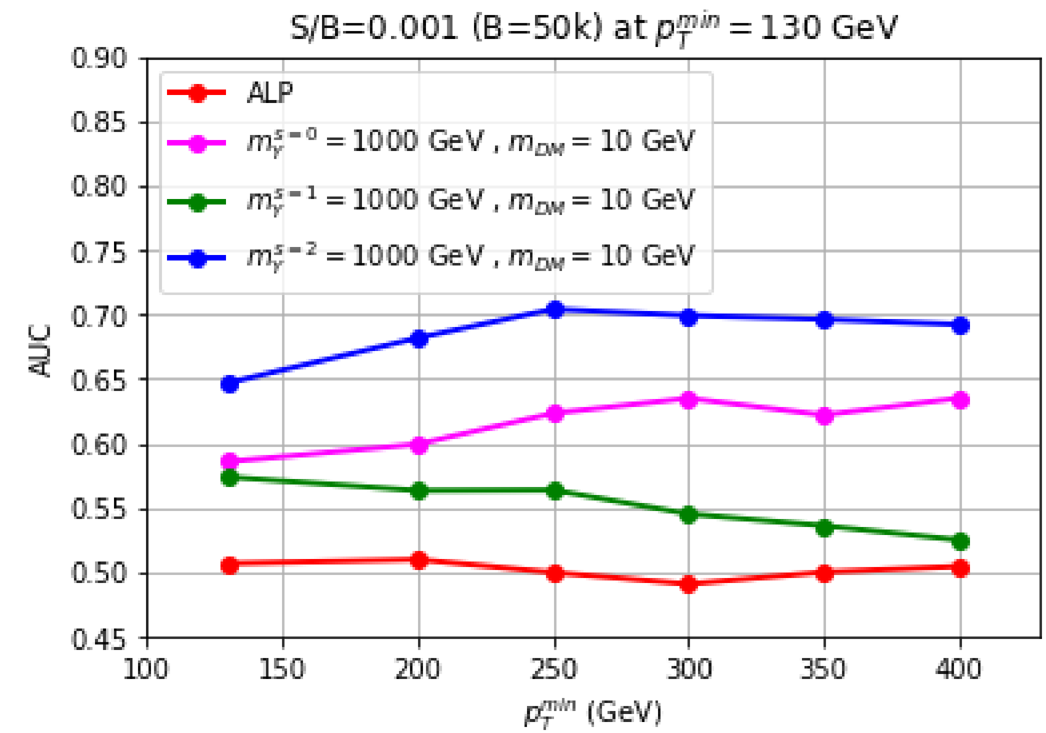


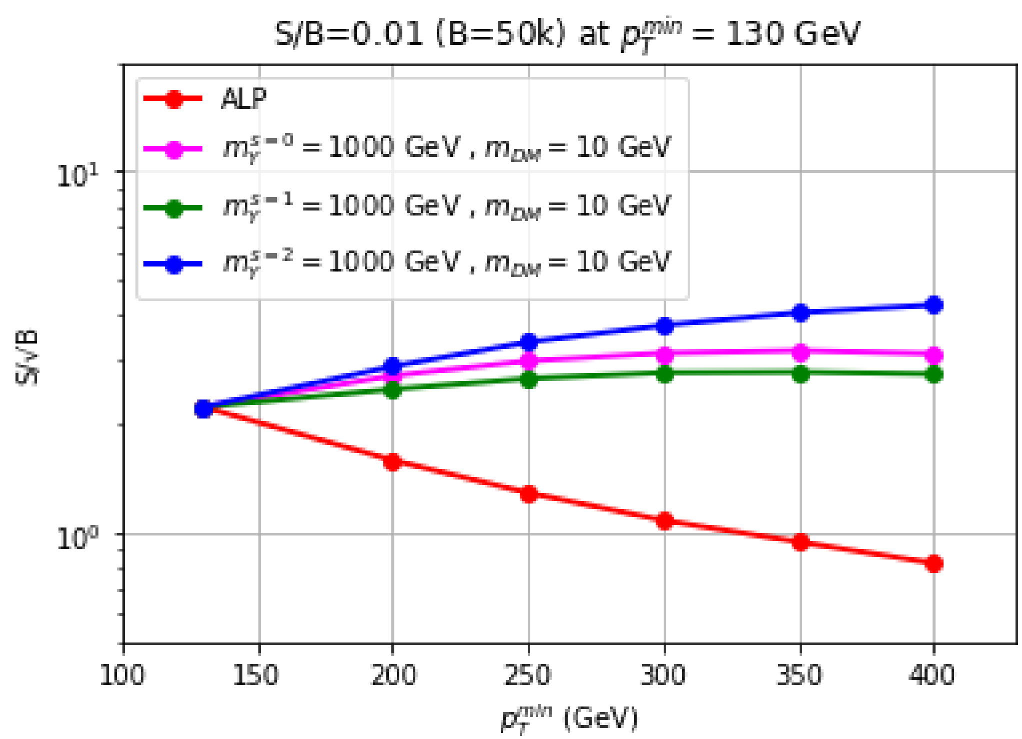


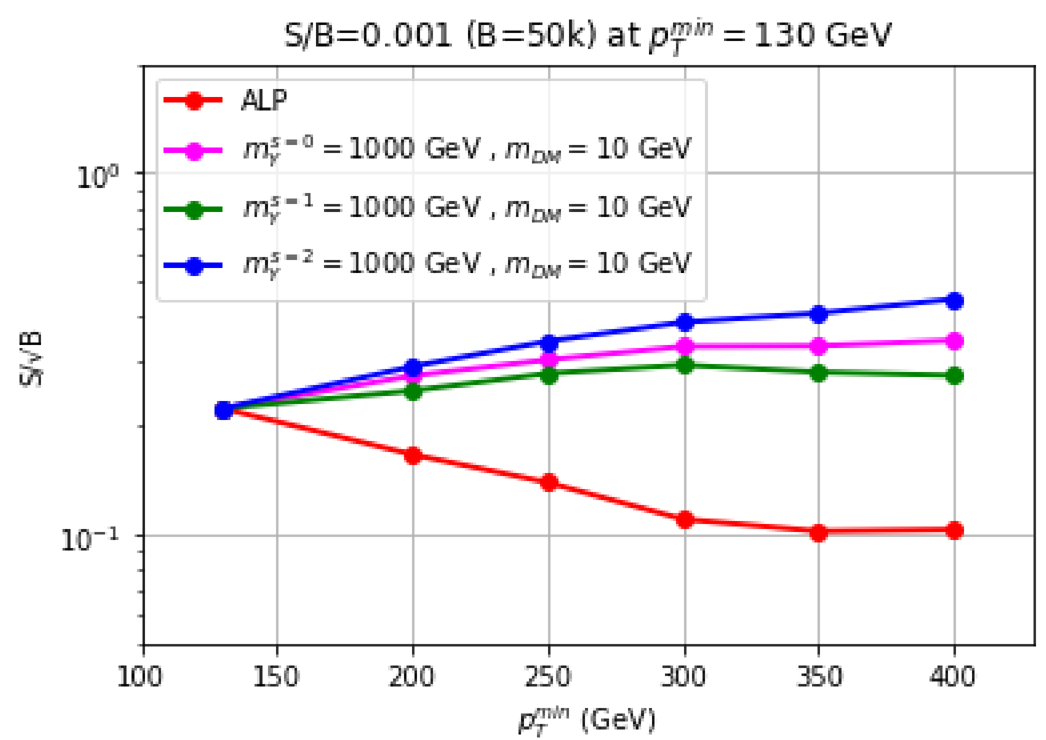


Minimum jet transverse momentum (p_T^j cut)









Discerning between DM models

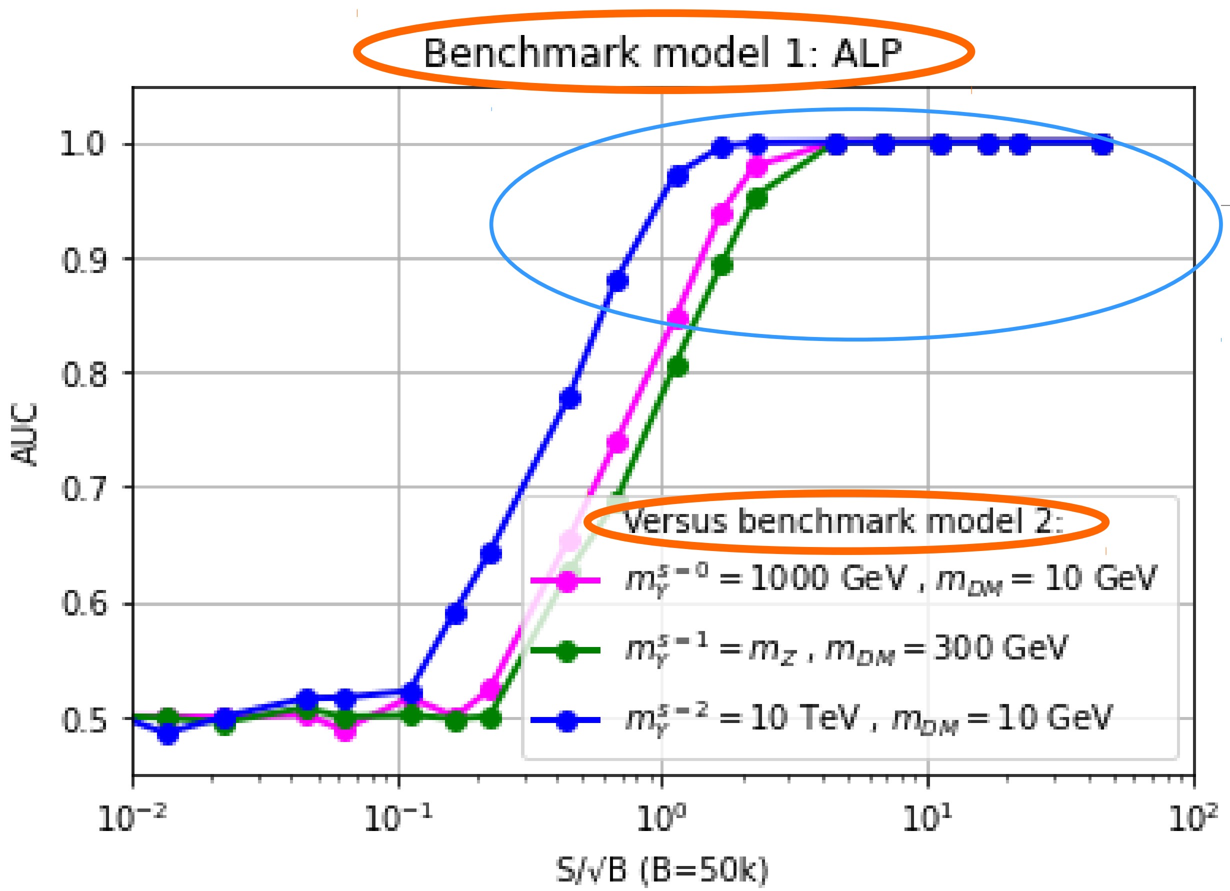
With individual DNNs

Discerning between DM models

If signal indicates the presence of non-SM processes, we would like to identify the underlying new physics model.

DNN performance to distinguish between histograms with:

Benchmark model 1 plus SM vs **Benchmark model 2 plus SM**

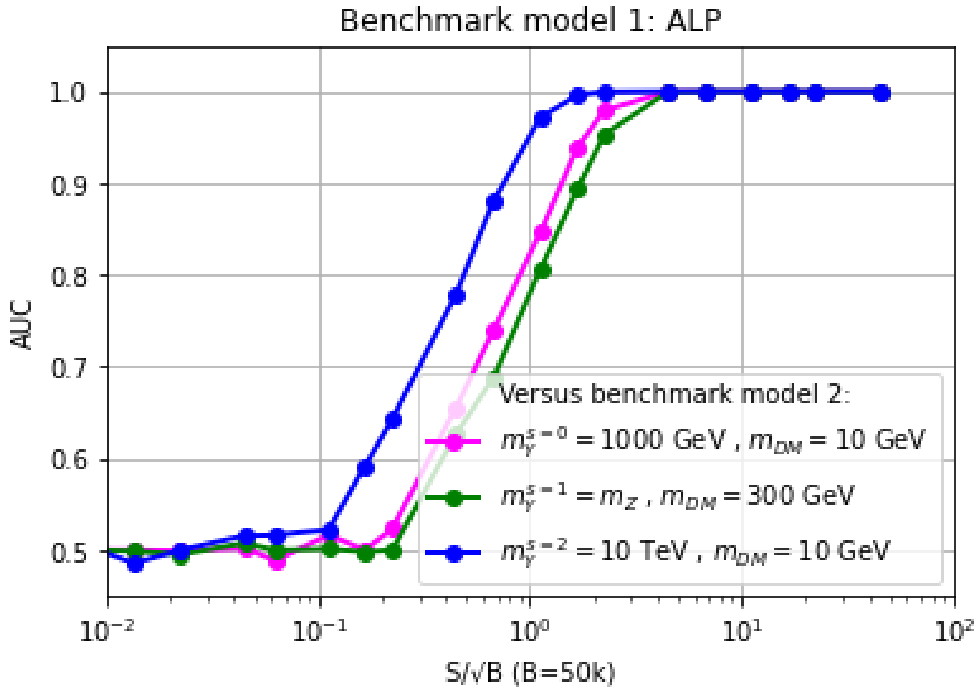


Can discriminate the ALP model from the others

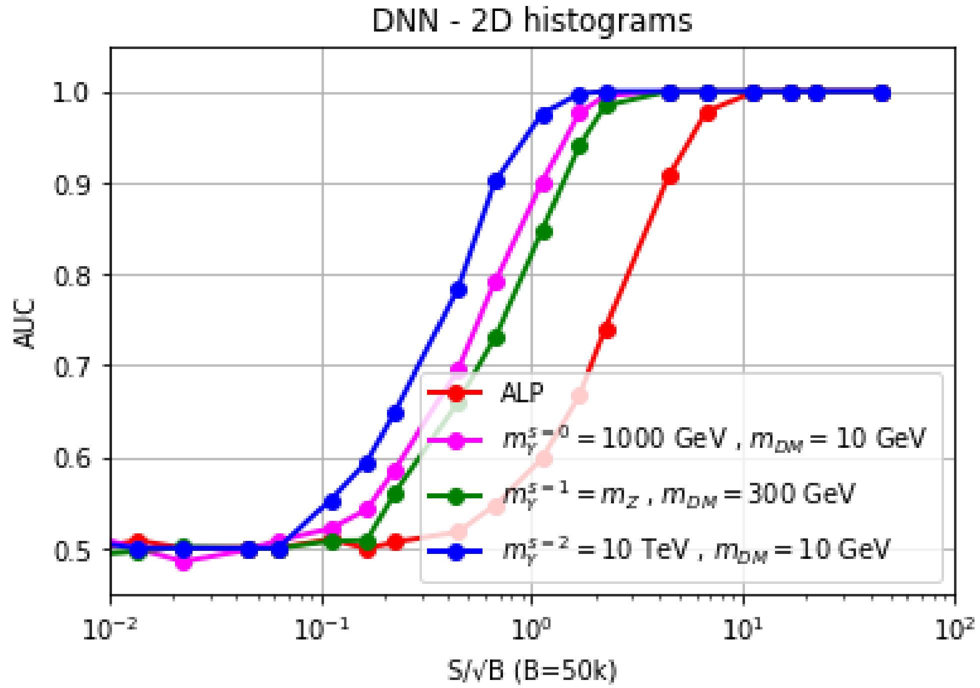
Discerning between DM models

ALP plus SM
vs
the other benchmark models plus SM

The ALP model takes the role of the SM, because both kinematic distributions are similar.



SM only
vs
Benchmark model plus SM

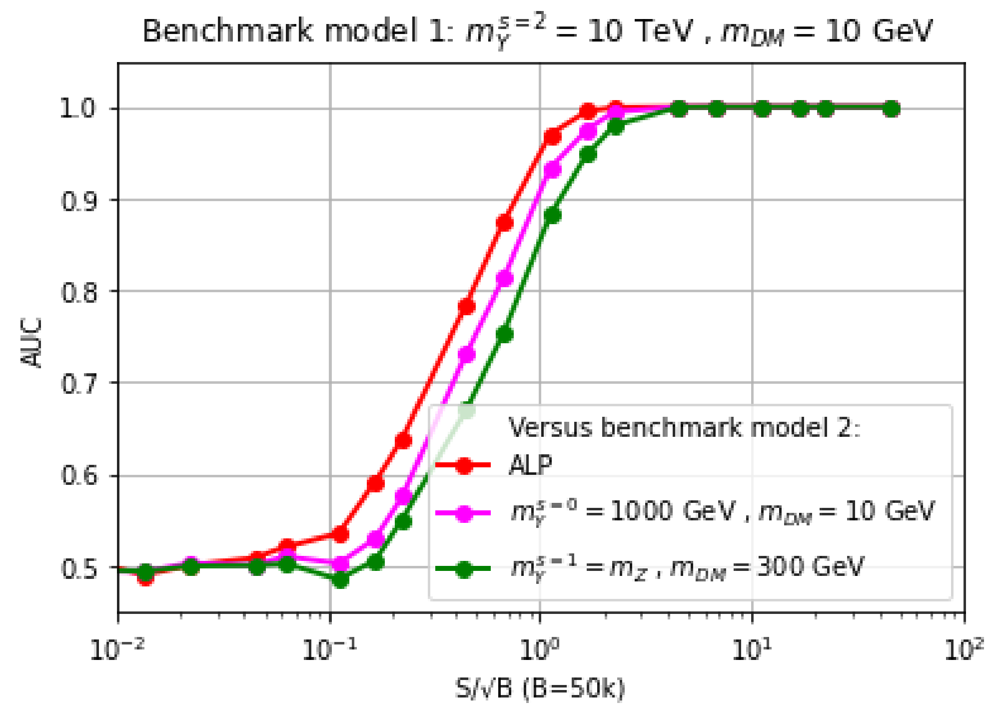


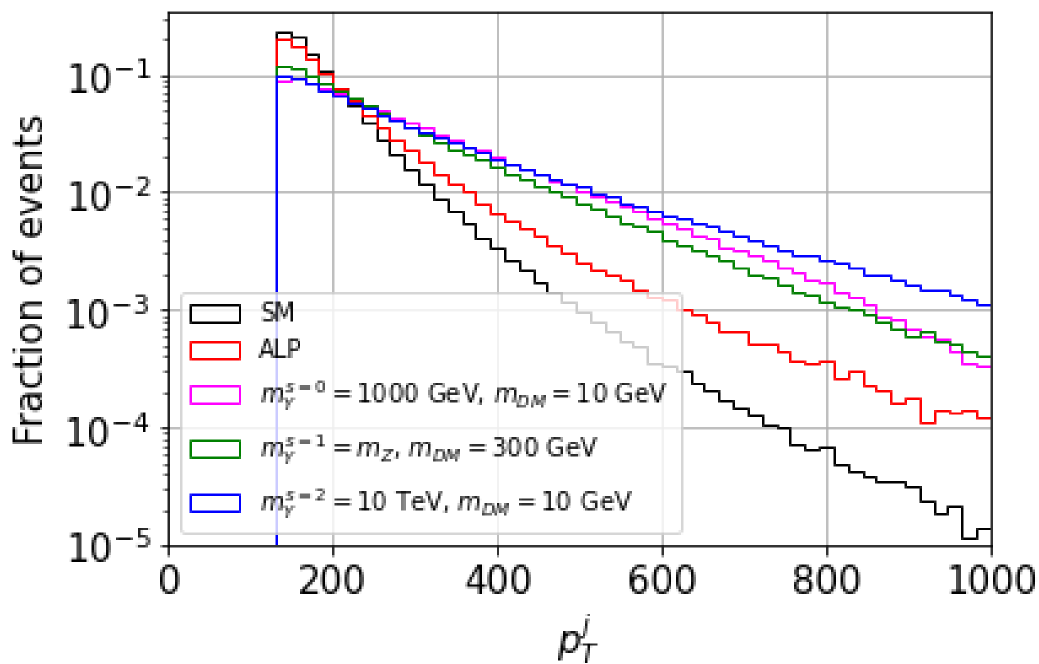
Discerning between DM models

Spin-2 mediator plus SM
vs
the other benchmark models plus SM

The **spin-2 model** has the hardest spectrum

→ highest efficiency achieve when we try to disentangle between the **spin-2 model** and the model with the softer spectrum, **ALP**.





Monojet and dijet + MET processes

Monojet → 3 kinematic variables: p_T^{j}, η^{j}, Φ^{j}

Dijet → 8 kinematic variables:

- p_T^{j1}, η^{j1}, (j1: leading jet)
- p_T^{j2}, η^{j2}, (j2: sub-leading jet)
- p_T^{MET} missing transverse momentum
- $\Delta\Phi^{j1\,j2}$, $\Delta\Phi^{j1}_{MET}$, $\Delta\Phi^{j2}_{MET}$