The International Joint Workshop on the Standard Model and Beyond 2024 &
3rd Gordon Godfrey Workshop on Astroparticle Physics
@ UNSW, Sydney, Australia
December 13, 2024

Weak Supervision Techniques in Collider Physics

Cheng-Wei Chiang
National Taiwan University
National Center for Theoretical Sciences

Refs:
Hugues Beauchesne, Zong-En Chen, and CWC, JHEP 02 (2024) 138
Zong-En Chen, CWC, and Feng-Yang Hsieh, 2412.00198

The International Joint Workshop on the Standard Model and Beyond 2024 &
3rd Gordon Godfrey Workshop on Astroparticle Physics
@ UNSW, Sydney, Australia
December 13, 2024

Weak Supervision Techniques in Collider Physics

Cheng-Wei Chiang
National Taiwan University
National Center for Theoretical Sciences

Refs:
Hugues Beauchesne, Zong-En Chen, and CWC, JHEP 02 (2024) 138
Zong-En Chen, CWC, and Feng-Yang Hsieh, 2412.00198

Outline

- Introduction
- Weak supervision
- Dark valley model
- Transfer learning
- Data augmentation
- Summary

Introduction

New Physics at LHC?

- We have been looking for new physics desperately at the LHC.
  ➠ only the SM-like Higgs was discovered

- Perhaps the sensitivity of traditional methods is not high enough?

- Can we utilize the **deep machine learning** technique to enhance the sensitivity so that we can better discover/constrain new physics?

Types of Machine Learning

- Supervised learning
  - Training data with labels (e.g., recognizing photos of cats and dogs)
- Unsupervised learning
  - Training data without labels (e.g., analyze and cluster unlabeled datasets)
- Reinforced learning
  - Data from interactions with the environment (e.g., chess and Go games)

 https://www.youtube.com/watch?v=Atg-Sl32vOo

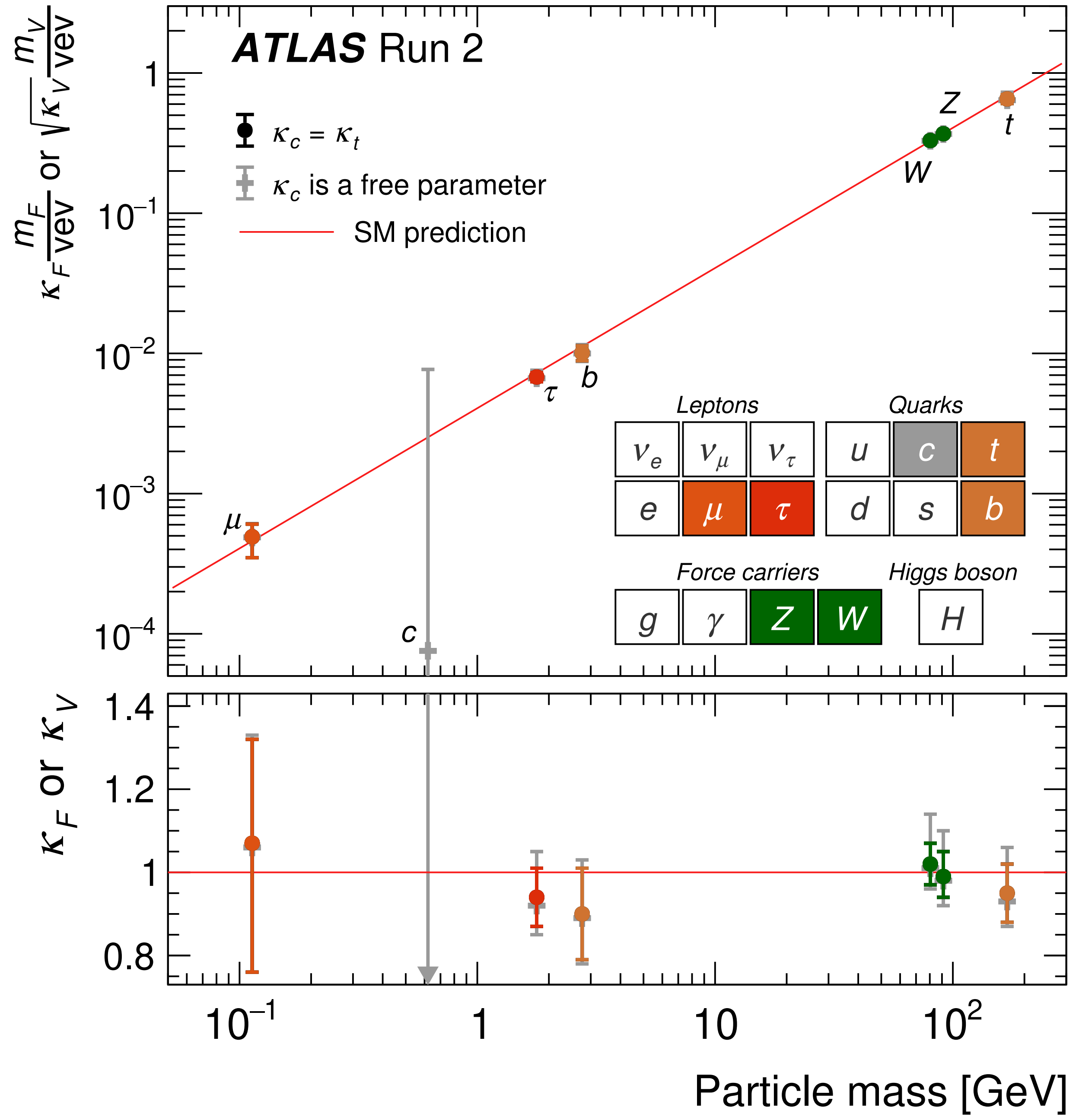
Types of Machine Learning

- Supervised learning
  - Training data with labels (e.g., recognizing photos of cats and dogs)
- Unsupervised learning
  - Training data without labels (e.g., analyze and cluster unlabeled datasets)
- Reinforced learning
  - Data from interactions with the environment (e.g., chess and Go games)
- **Weakly supervised learning**
  - When labeled data are *difficult* or *impossible* or *expensive* to obtain

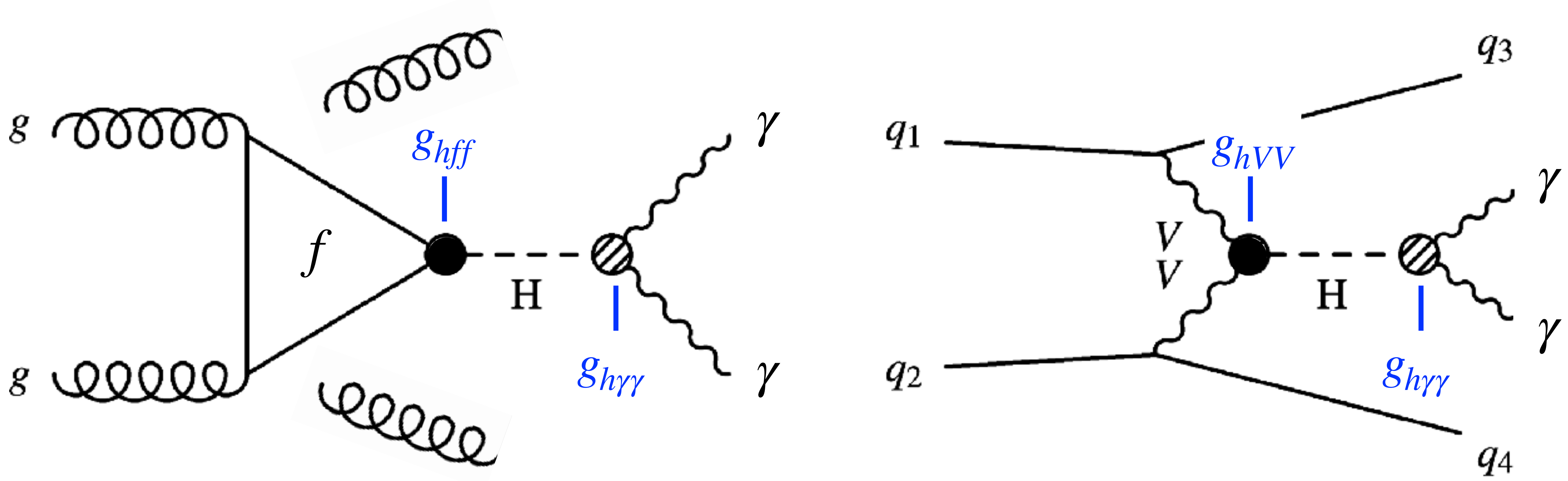
 https://www.youtube.com/watch?v=Atg-Sl32vOo

VBF/GGF Higgs Production

- Questions:
  - For each *detected* Higgs event, how can we *efficiently* and *correctly* determine/label its production mechanism?
  - Can it be *independent* of how the Higgs boson decays?



ATLAS 2019



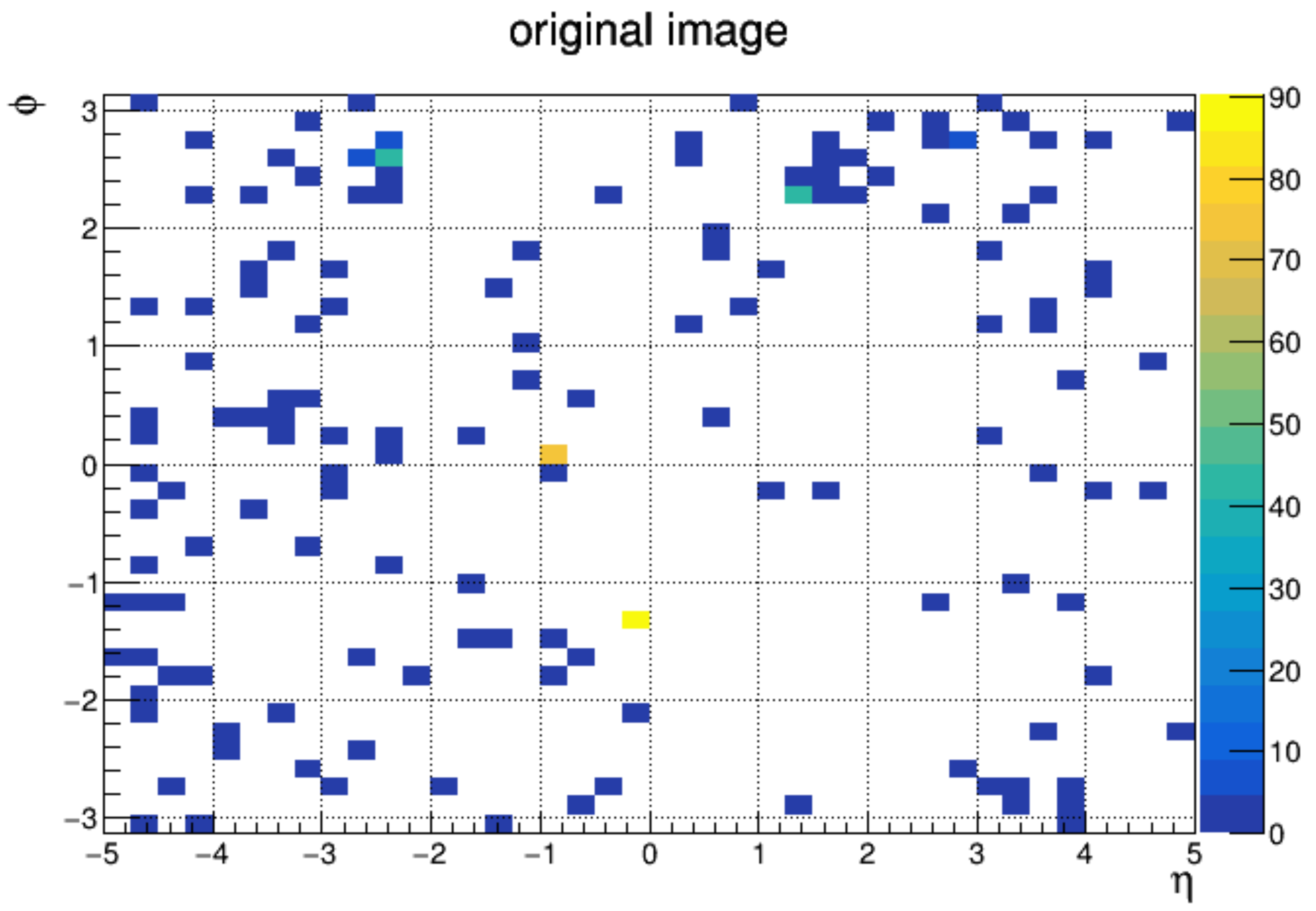
(a) ggF production

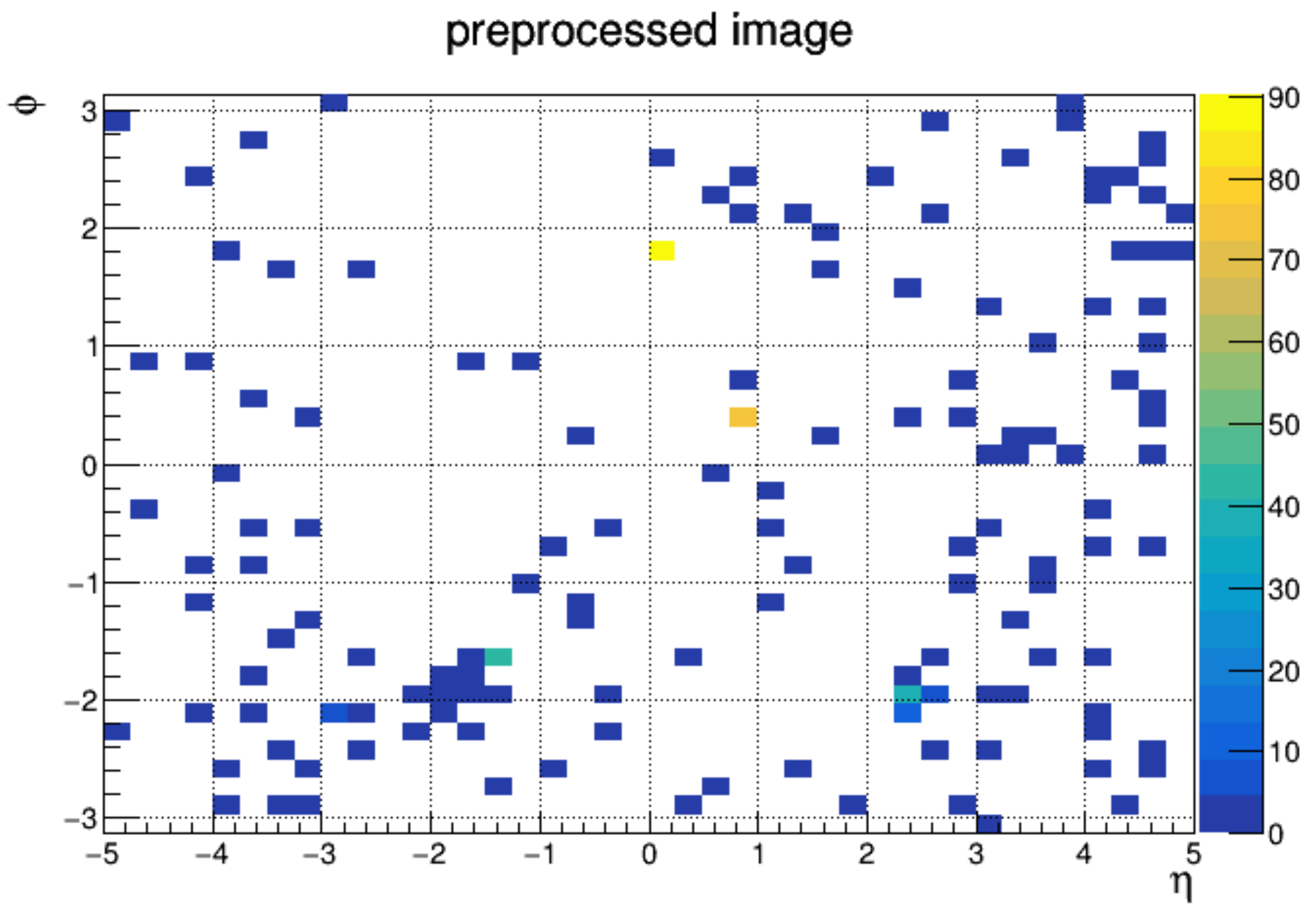
(b) VBF production

Event-CNN

- Train a convolutional neural network (CNN) by **full supervision** to discriminate the two production mechanisms by examining the final-state image.
- A successful training typically requires at least **tens of thousands** of samples.

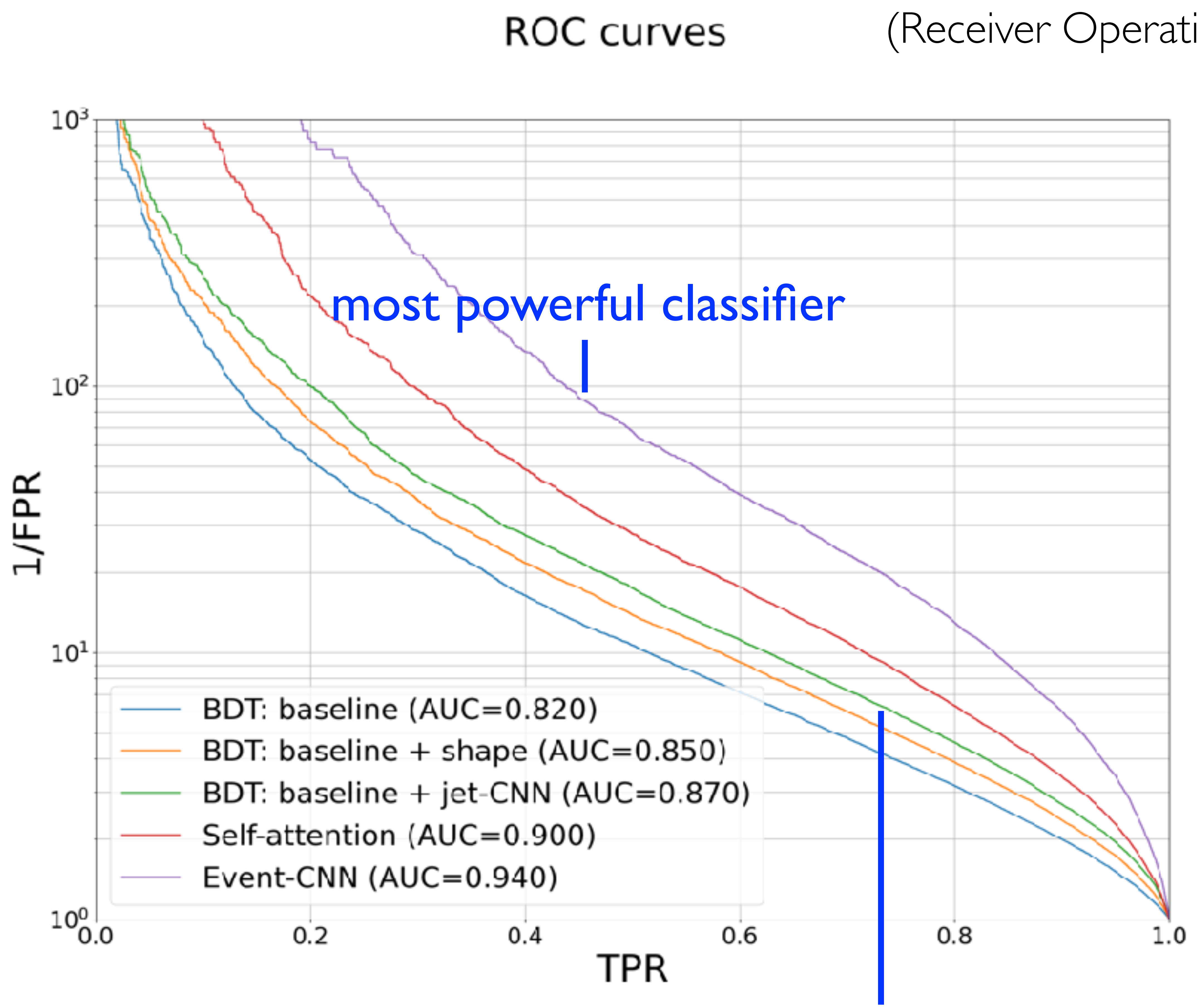
	training	validation	testing
VBF events	105k	26k	33k
GGF events	83k	21k	26k

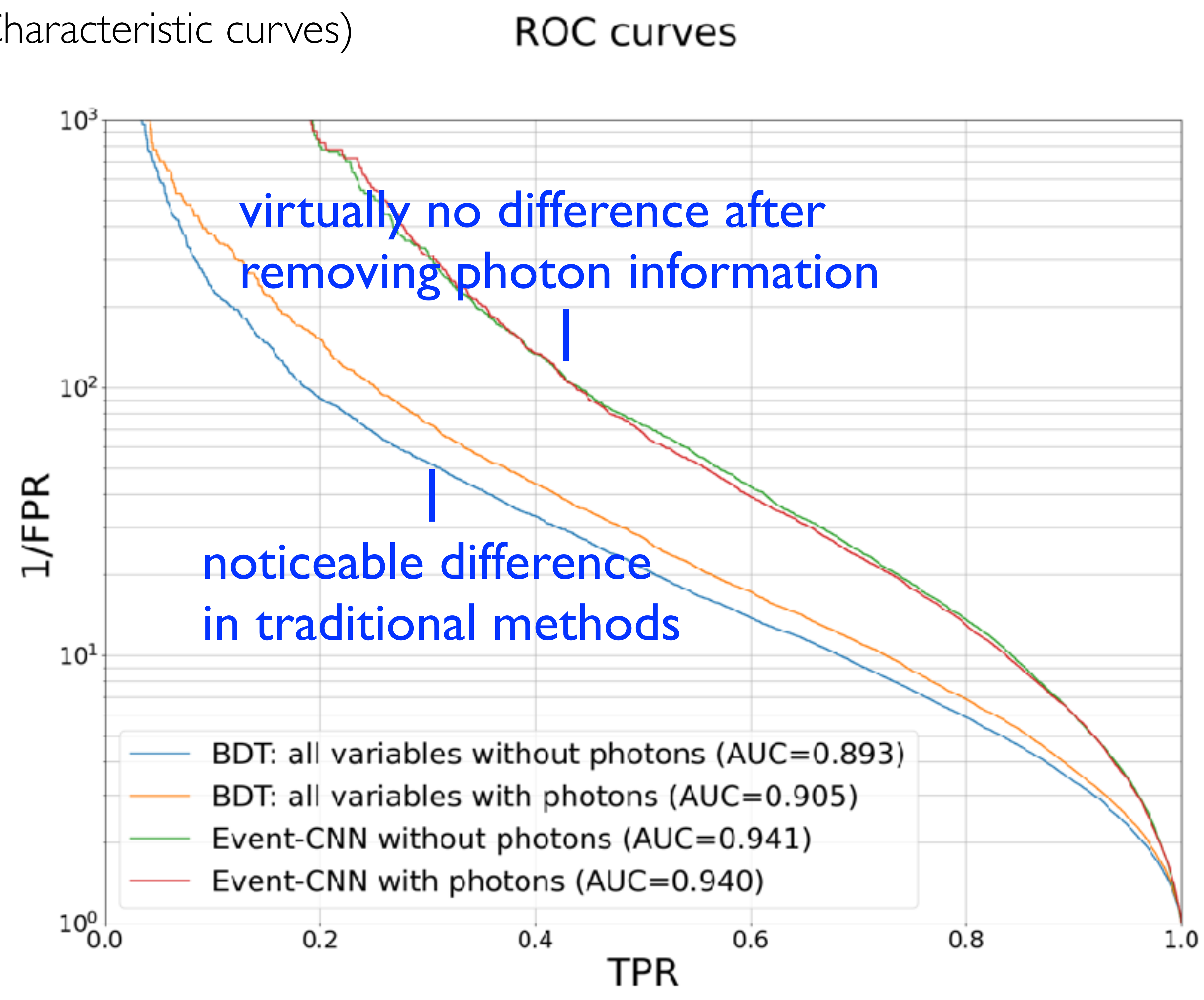




Comparison of Classifiers

(Receiver Operating Characteristic curves)





jet-CNN has learned the information contained in the human-engineered jet shape variables

CWC, Shih, Wei 2023

Requirements on Training Data

- **High-Quality Data**: The dataset should be representative of the problem domain and free of noise or irrelevant features. *Preprocessing* steps like removing outliers, handling missing values, standardization by utilizing symmetries, and balancing class distributions are crucial.

- **Sufficient Data**: Neural networks typically require large amounts of *labeled* data to learn meaningful patterns. When the dataset is small, techniques like *transfer learning* or *data augmentation* can mitigate data scarcity.

- **Data Diversity**: Samples in the datasets should be sufficiently *diverse* in properties in order to help the model *generalize* better and *avoid overfitting* to specific patterns.

Weak Supervision with CWoLa

Collider Simulations

Collider Simulations

- Particle experimentalists deal with **real data** collected by detectors around colliders.
  ➠ just like analyzing real images for CS people
  ➠ even current multivariate approaches for classification rely on simulations and must be corrected later on using data-driven techniques

Collider Simulations

- Particle experimentalists deal with **real data** collected by detectors around colliders.
  ➠ just like analyzing real images for CS people
  ➠ even current multivariate approaches for classification rely on simulations and must be corrected later on using data-driven techniques

Collider Simulations

- Particle experimentalists deal with **real data** collected by detectors around colliders.
  ➠ just like analyzing real images for CS people
  ➠ even current multivariate approaches for classification rely on simulations and must be corrected later on using data-driven techniques

- As particle theorists, we think we are simulating verisimilar data using various packages.
  ➠ in fact, we have been generating **fake data** all along
  ➠ problems: fixed-order in perturbation (e.g., CalcHEP, MadGraph), model-dependent showering/hadronization (e.g., Pythia, Herwig), crude detector simulations (e.g., Delphes, GEANT)

Collider Simulations

- Particle experimentalists deal with **real data** collected by detectors around colliders.
  ➠ just like analyzing real images for CS people
  ➠ even current multivariate approaches for classification rely on simulations and must be corrected later on using data-driven techniques

- As particle theorists, we think we are simulating verisimilar data using various packages.
  ➠ in fact, we have been generating **fake data** all along
  ➠ problems: fixed-order in perturbation (e.g., CalcHEP, MadGraph), model-dependent showering/hadronization (e.g., Pythia, Herwig), crude detector simulations (e.g., Delphes, GEANT)

Can We Be More Realistic?

Can We Be More Realistic?

- Use **adversarial networks** (so-called **GAN**). Louppe, Kagan, Cranmer 2016
  ➠ can alleviate model dependence during training, but at the cost of algorithmic performance and computational resources

Can We Be More Realistic?

- Use **adversarial networks** (so-called **GAN**). Louppe, Kagan, Cranmer 2016
  ➠ can alleviate model dependence during training, but at the cost of algorithmic performance and computational resources

- It would be nice to train directly using real data.
  ➠ but real data are **unlabeled**…

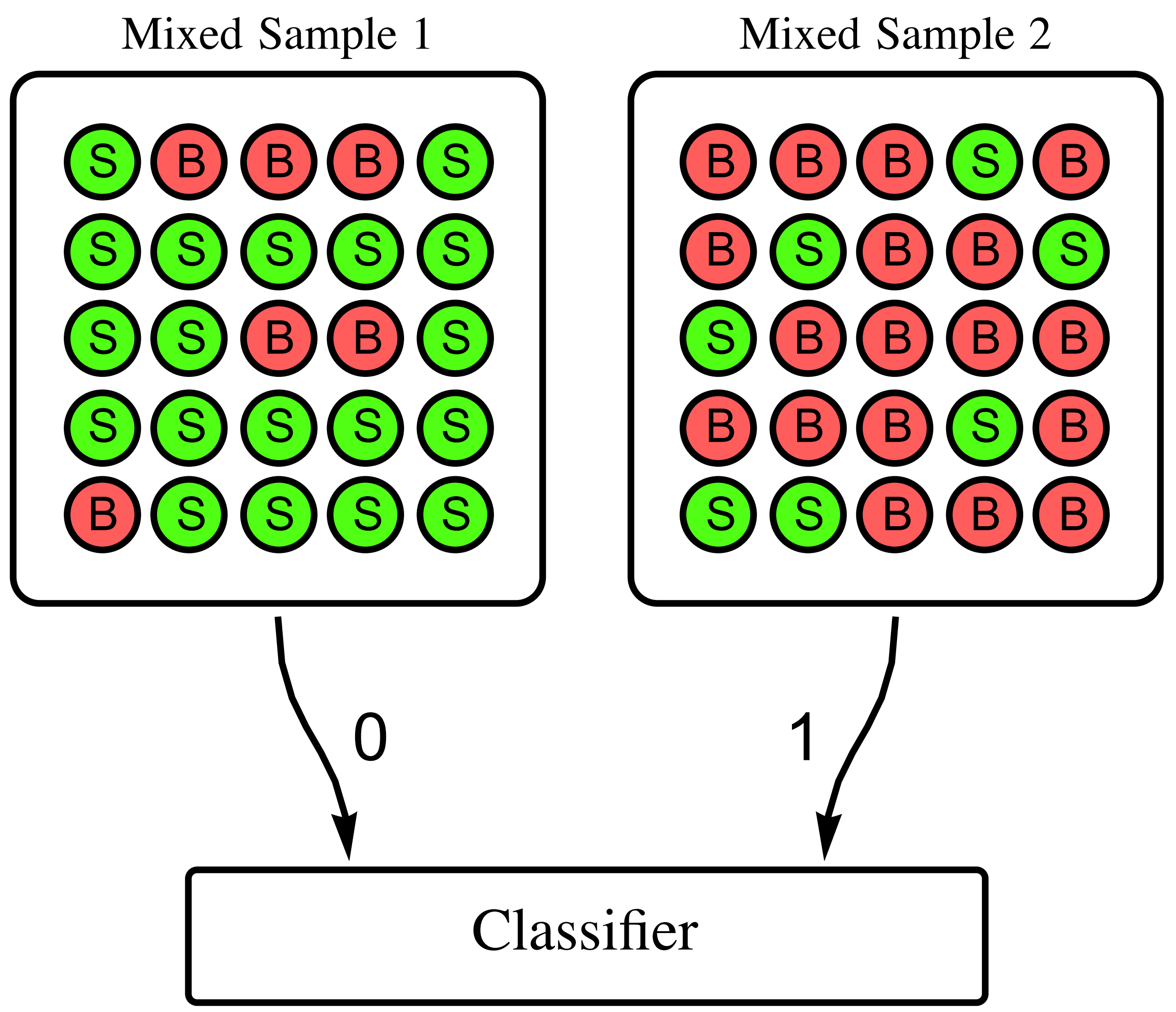
Can We Be More Realistic?

- Use **adversarial networks** (so-called **GAN**). Louppe, Kagan, Cranmer 2016
  ➠ can alleviate model dependence during training, but at the cost of algorithmic performance and computational resources

- It would be nice to train directly using real data.
  ➠ but real data are **unlabeled**…

- Introduce **classification without labels** (**CWoLa**, pronounced as koala).
  Metodiev, Nachman, Thaler 2017
  ➠ belonging to a broad framework called **weak supervision**, whose goal is to learn from *partially* and/or *imperfectly labeled* data Herna´ndez-Gonz´alez, Inza, Lozano 2016
  ➠ first weak supervision application in particle physics for *quark vs gluon* tagging using *only class proportions* during training; shown to match the performance of fully supervised algorithms Dery, Nachman, Rubbo, Schwartzman 2017

A Theorem for CWoLa



Metodiev, Nachman, Thaler 2017

- Let $\vec{x}$ represent a list of observables or an image, used to distinguish signal S from background B, and define:
    - $p_S(\vec{x})$: probability distribution of $\vec{x}$ for the signal,
    - $p_B(\vec{x})$: probability distribution of $\vec{x}$ for the background.
- Given mixed samples M_1 and M_2 defined in terms of pure events of S and B (both being *identical* in the two mixed samples) using

$$p_{M_1}(\vec{x}) = f_1 p_S(\vec{x}) + (1 - f_1)\, p_B(\vec{x})$$
$$p_{M_2}(\vec{x}) = f_2 p_S(\vec{x}) + (1 - f_2)\, p_B(\vec{x})$$

with *different* signal fractions $f_1 > f_2$, an *optimal classifier* (most powerful test statistic) trained to distinguish samples in M_1 and M_2 is also *optimal* for distinguishing S from B.

Remarks

- An important feature of CWoLa is that, unlike the learning from label proportions (LLP) weak supervision, the label proportions f_1 and f_2 are **not required** for training as long as they are *different*.
- This theorem only guarantees that the optimal classifier from CWoLa, if reached, is the same as the optimal classifier from fully-supervised learning.
- Just like most cases, successful training for CWoLa also requires **a large amount of samples**.
- What happens if available data for the mixed samples are **insufficient or limited**, as is often the case of **real data for BSM searches**?

Dark Valley Model

Dark Valley Model and Dark Jets

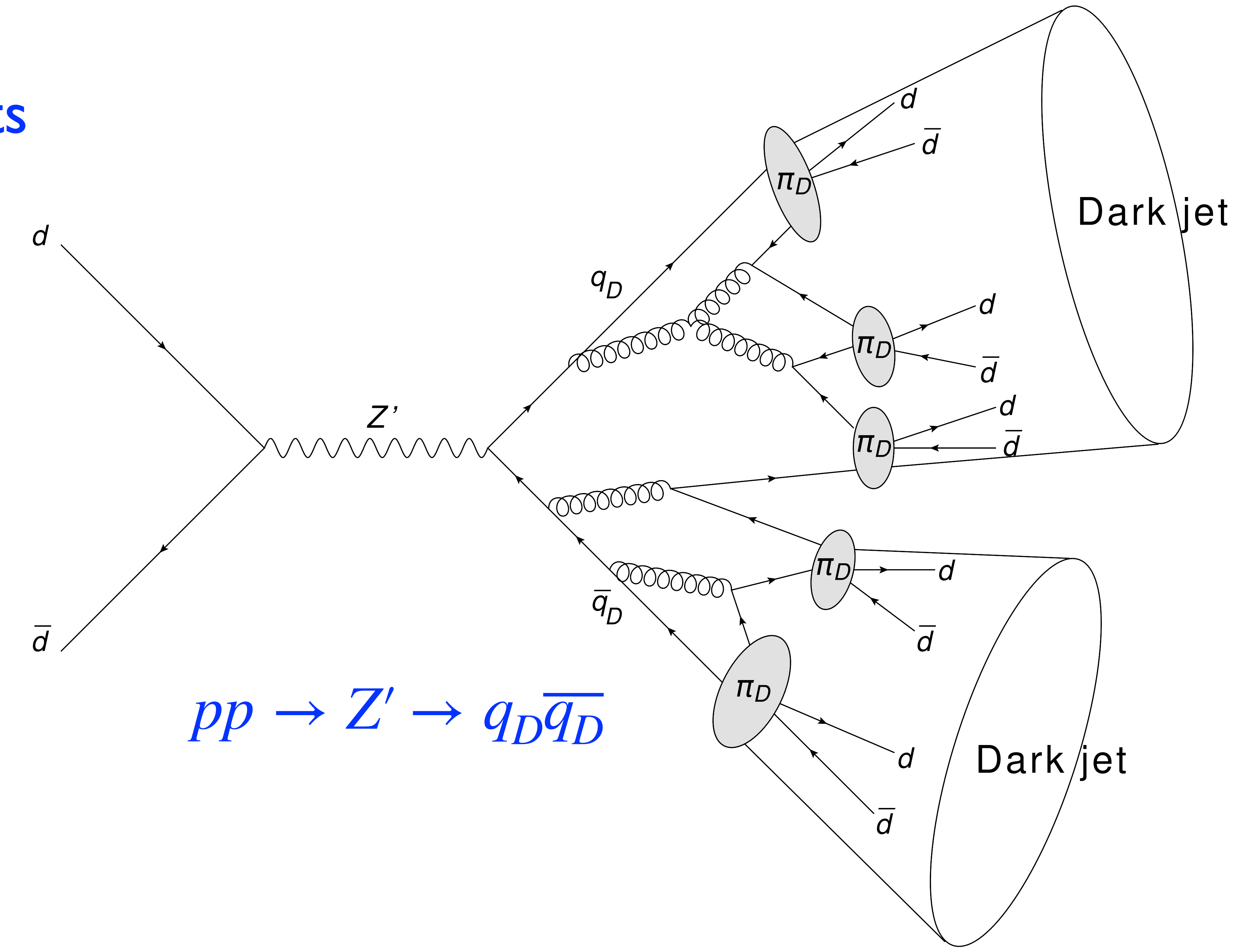
- Assume the existence of a **dark confining sector** that communicates with the visible sector via a **heavy Z' portal**:

dark quarks

$$\mathcal{L} \supset -Z'_\mu \left(g_q \overline{q_i} \gamma^\mu q_i + g_{q_D} \overline{q_{D\alpha}} \gamma^\mu q_{D\alpha} \right)$$

respective effective coupling constants

- For our purposes here, we
  - consider Z' couplings to the d-quarks only, though other SM particles are also possible;
  - give Z' a mass without specifying its source;
  - will not worry about such issues as anomaly cancellation and $Z - Z'$ mixing.



Courtesy of Hugues Beauchesne

- The LHC signature is **a pair of dark jets** with invariant mass consistent with $m_{Z'}$.

Dark Sector Parameter Choices

- The Z' **mass** is fixed at 5.5 TeV, and its **width** is fixed at 10 GeV.
  ➠ invariant mass of the two leading jets being around 5.2 TeV (with some constituents falling outside the reconstructed jets)

- The **dark confining scale** $\Lambda_D \in \{1,\ 5,\ 10,\ 20,\ 30,\ 40,\ 50\}$ GeV.

- Dark vector ρ_D and pseudoscalar π_D masses and two (prompt) decay scenarios:

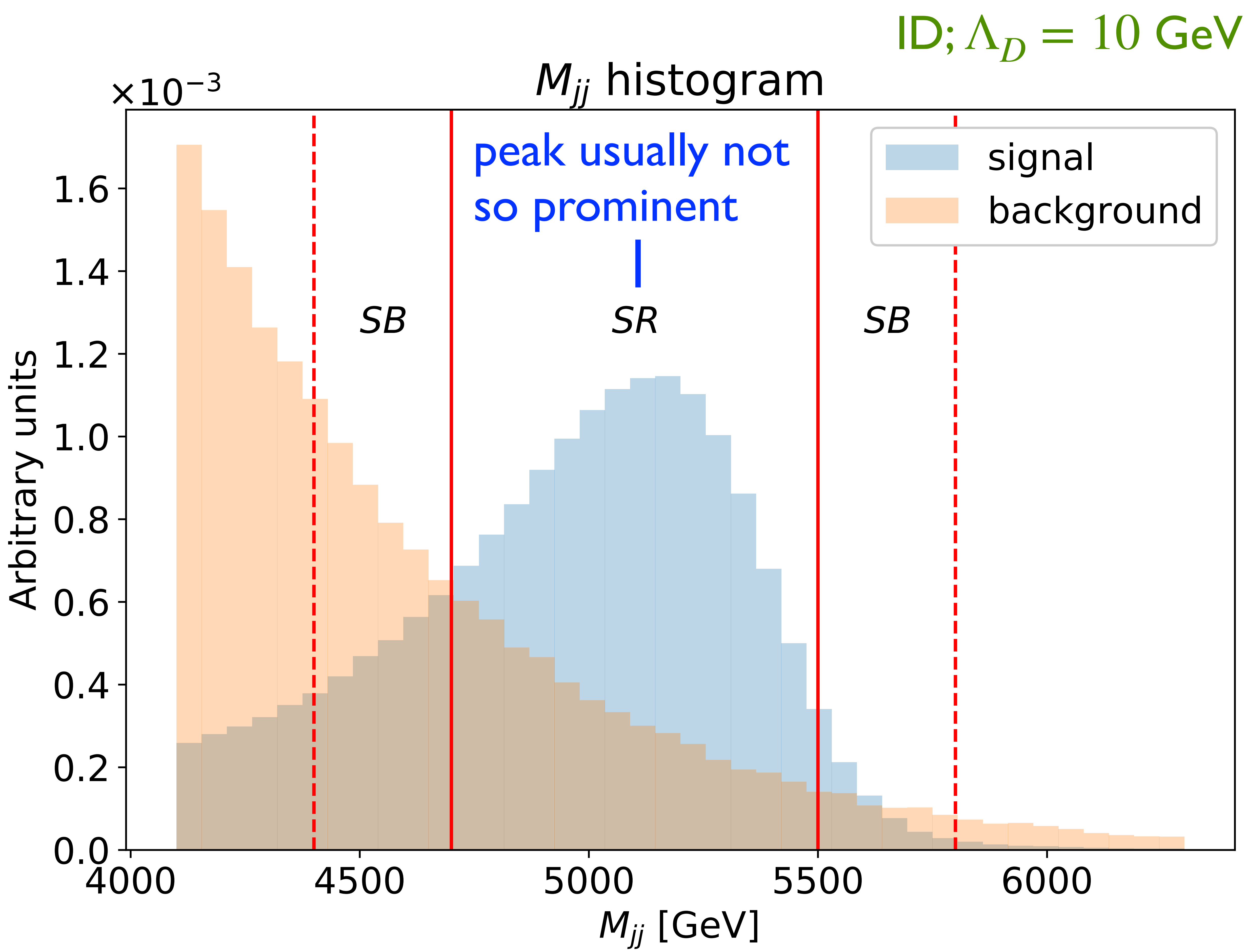
$$\frac{m_{\rho_D}}{\Lambda_D} = \sqrt{5.76 + 1.5\frac{m_{\pi_D}^2}{\Lambda_D^2}}$$

Albouy et al 2022

  - **Indirect Decay (ID)**: $\rho_D \to \pi_D \pi_D$ followed by $\pi_D \to d\bar{d}$ for $m_{\pi_D}/\Lambda_D = 1.0$
  - **Direct Decay (DD)**: $\rho_D,\ \pi_D \to d\bar{d}$ for $m_{\pi_D}/\Lambda_D = 1.8$

Dijet Invariant Mass Distributions

- Madgraph 2.7.3 with PDF = NN23LO1
- Pythia 8.307 with default settings
- Delphes 3.4.2 with default CMS card and jet radius $R = 0.8$



SR: signal region
SB: side-band region
➠ two mixed samples (M_1 and M_2) with different signal/background fractions

Signal and background events are assumed to be the same in both SR and SB, which should be valid to a good approximation.

Figure 1. Dijet invariant mass distributions for the indirect decaying scenario with $\Lambda_D = 10\,\text{GeV}$ and for the SM background. Distributions are normalized to unity. Both signal and background satisfy the selection criteria of table 1(b) except for the SR or SB conditions.

CNN + Dense Layers

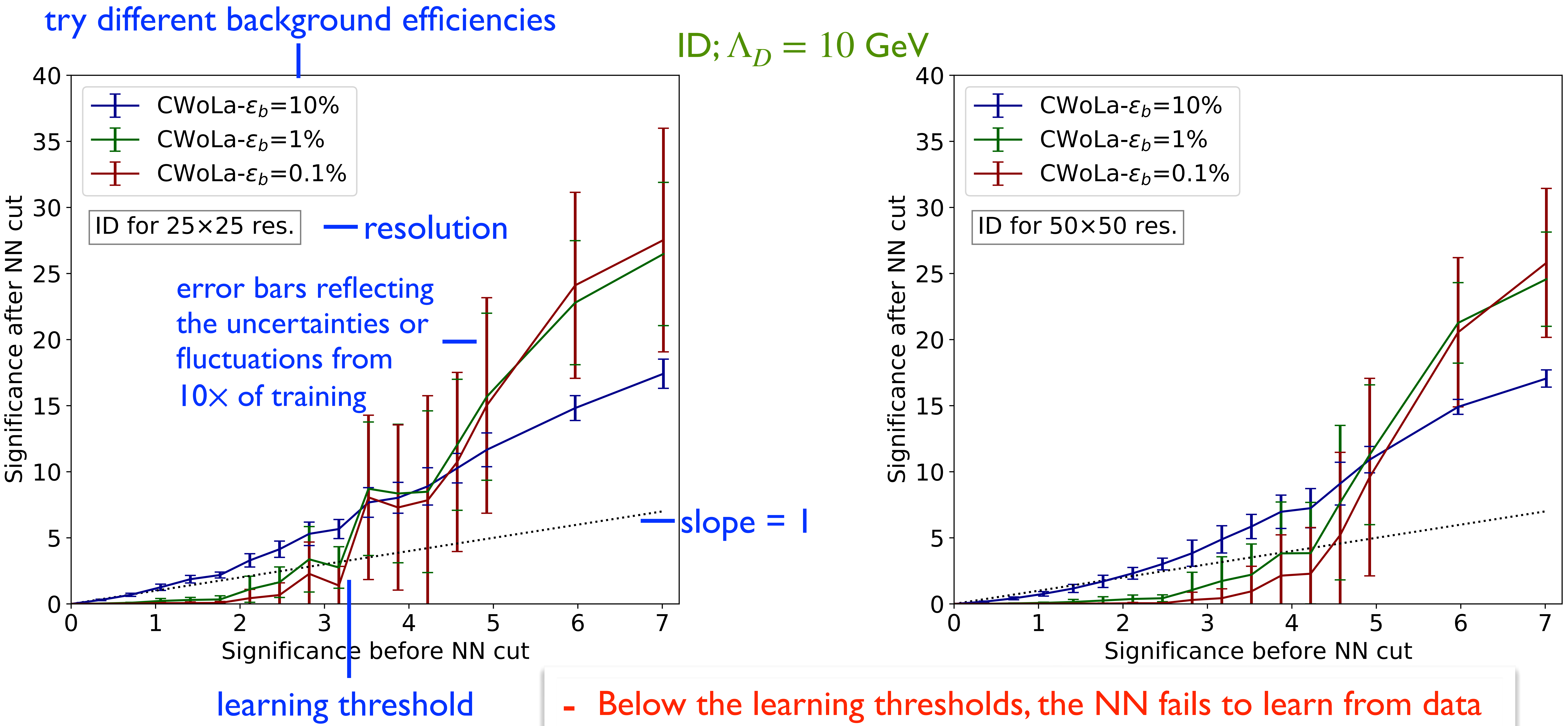
- Prepare each jet image in **three resolutions**: 25×25, 50×50, 75×75.
- Use the **images of the two leading jets** as input data.
- Pass each image through a **common** CNN*, and each returns a score $\in [0,1]$.
- Take the **product** of these two scores as the output of the full NN.

- The convolutional part of the NN is referred to as the **feature extractor**, and its weights and biases are collectively labeled as Θ.
  ➠ to be transferred later

- The weights and biases of the dense layers are collectively labeled as θ.
  ➠ to be fine-tuned later

* All NNs are implemented using `Keras` with `TensorFlow` backend. Also, using two distinct networks for the two jets would give slightly inferior results, possibly caused by the lack of signal.

Results of Regular CWoLa



- Below the learning thresholds, the NN fails to learn from data because it cuts background and signal indiscriminately, resulting in a significance even worse than without employing the NN.
- Increasing resolution tends to shift the thresholds higher because more parameters are to be learned inside the NN.

Transfer Learning

Introduction to Transfer Learning

- The phrase “**transfer learning (TL)**” comes from psychology.
  ➠ a learner new to a fresh topic (e.g., playing guitar or riding a motorcycle) typically has a higher learning threshold, while a learner experienced in related topics, even if different, (e.g., playing violin or riding a bicycle) usually has less difficulty in quickly picking it up
- As an ML technique, TL reuses a **pre-trained model** developed for one task as the starting point of a new model for a new task.
  ➠ transferring knowledge or experience extracted in the pre-trained model for a **source task/domain** to a new model for a **target task/domain**
  ➠ weights from the pre-trained model used to initialize those of the new model
- TL would only be successful when the features learned from the first model trained on its task can be *generalized* and *transferred* to the second task.
  ➠ dataset in the second training should be sufficiently similar to those in the first training

Pre-training and Fine-tuning

- **Pre-training**:
    - A neural network would first be trained on a *larger* dataset (source data) based upon *simulations*, which are only required to be sufficiently realistic but not necessarily faithful, to either learn certain concepts or become a more **efficient learner**.
- **Fine-tuning**:
    - The pre-trained model is subsequently trained on a *new* and possibly *smaller* dataset (target data), such as the actual collider data.

Transfer Learning by Pre-training and Fine-tuning

- **Step 1**: The NN is first trained to distinguish a sample of pure background from a pure combination of different signals, which includes all the models mentioned before (ID and DD, different values of Λ_D), except the benchmark on which the model will be tested.
  - ➠ **pre-training** on a large set of simulations as the **source data**
  - ➠ 200k S and 200k B events in the SR for training
    + 50k S and 50k B events for validation
  - ➠ training both Θ (from convolutional layers) and θ (from dense layers)

Layers of CNN subnetwork	$\begin{pmatrix} \text{convolutional 2D layer: 64 filters with } 5\times5 \text{ kernel size} \\ \text{maxpooling layer: } 2\times2 \text{ pool size} \end{pmatrix} \times 2$ convolutional 2D layer: 128 filters with 3×3 kernel size maxpooling layer: 2×2 pool size convolutional 2D layer: 128 filters with 3×3 kernel size flatten layer (dense layer: 128 units) $\times 3$ dense layer (output): 1 unit

Θ

θ

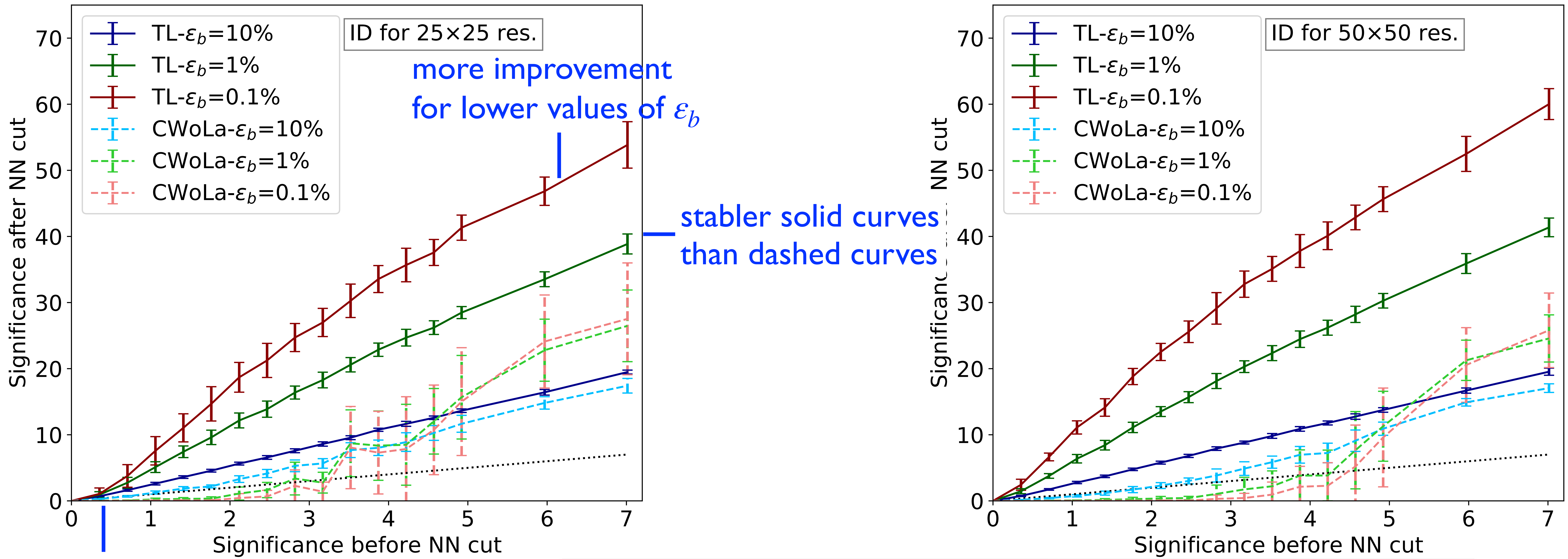
Transfer Learning by Pre-training and Fine-tuning

- **Step 2**: The NN is then trained to distinguish the mixed samples (i.e., the SR and SB regions) using the **actual** data of the benchmark signal (of the true model) plus the SM background.
  ➠ **fine-tuning** on the actual data as **target data**
  ➠ freezing Θ in the convolutional layers and reinitializing and training θ in the dense layers
  ➠ fixing the feature extraction part while training the classification part

Layers of CNN subnetwork	$\begin{pmatrix} \text{convolutional 2D layer: 64 filters with } 5 \times 5 \text{ kernel size} \\ \text{maxpooling layer: } 2 \times 2 \text{ pool size} \end{pmatrix} \times 2$	Θ
	convolutional 2D layer: 128 filters with 3×3 kernel size	
	maxpooling layer: 2×2 pool size	
	convolutional 2D layer: 128 filters with 3×3 kernel size	
	flatten layer	
	(dense layer: 128 units) $\times 3$	θ
	dense layer (output): 1 unit	

Transfer Learning vs Regular CWoLa

ID; $\Lambda_D = 10$ GeV



more improvement for lower values of ϵ_b

stabler solid curves than dashed curves

lower learning thresholds for TL

- The amount of signal necessary to claim a 5σ discovery can be reduced by a factor of a few, which is due to the fact that the NN can better keep the signals.
- Fluctuations in the significance are reduced, due to a smaller amount of trainable parameters and more successful learning.

Data Augmentation

Augmentation Methods

- While there are numerous augmentation methods in the field of computer vision, we focus on *physics-inspired* techniques related to our study.

Wang et al 2024
Dillon, Favaro, Feiden, Modak, and Plehn 2024

- Considering augmentations that capture the **symmetries** of the physical events and the experimental **resolution** or statistical **fluctuations** in the detector, we implement three methods:
    - p_{T} **smearing**;
    - **jet rotation**; and
    - a **combination** of the two.

- Additionally, we have applied $\eta - \phi$ **smearing** and **Gaussian noise** to jet images and observed essentially no improvement.

p_{T} Smearing Method

- The p_{T} smearing method is used to simulate **detector resolution/fluctuation** effects on the transverse momentum of jet constituents.

- This method resamples the transverse momentum p_{T} of jet constituents according to the **normal distribution**:

$$p'_{\mathrm{T}} \sim \mathcal{N}\left(p_{\mathrm{T}}, f\left(p_{\mathrm{T}}\right)\right), \quad f\left(p_{\mathrm{T}}\right) = \sqrt{0.052 p_{\mathrm{T}}^2 + 1.502 p_{\mathrm{T}}}$$

  where p'_{T} is the augmented transverse momentum, and $f\left(p_{\mathrm{T}}\right)$ is the **energy smearing function** applied by Delphes (with p_{T} normalized in units of GeV).

- The preprocessing is then applied after the p_{T} smearing augmentation.

- This augmentation helps the model consider the **detector effects**. It has the effect of making the training results more robust.

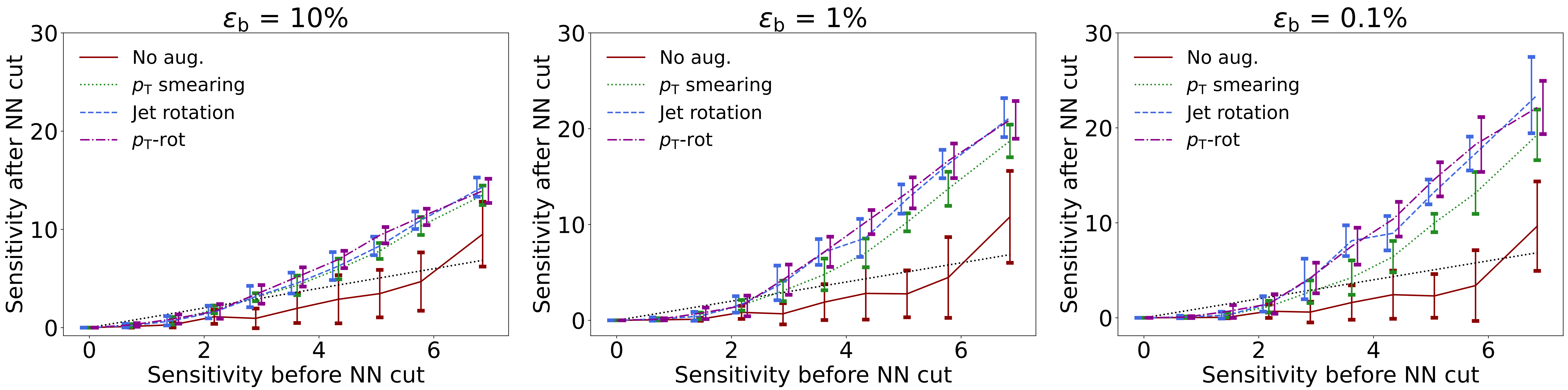
Jet Rotation Method

- The jet rotation method rotates each jet with respect to its center by a random angle $\theta \in [-\pi, \pi]$ to enlarge the **diversity** of training datasets.
- More specifically, the (η', ϕ') coordinates of a jet constituent after preprocessing are rotated as follows: $\eta'' = \eta' \cos\theta - \phi \sin\theta$ and $\phi'' = \eta' \sin\theta + \phi' \sin\theta$, where (η'', ϕ'') are the rotated coordinates.
- We allow the two leading jets in an event to be rotated by **different** angles, thereby further increasing the diversity of the training dataset.
- The complete workflow for preparing jet images with this augmentation is: translation, orientation, flipping, jet rotation, followed by pixelation.
- We have tested other ranges of jet rotation angles, including $[-\pi/6,\pi/6]$, $[-\pi/3,\pi/3]$, and $[-\pi/2,\pi/2]$.

  ➠ the training performance improves as the range of rotation angles increases

Sensitivity Improvement

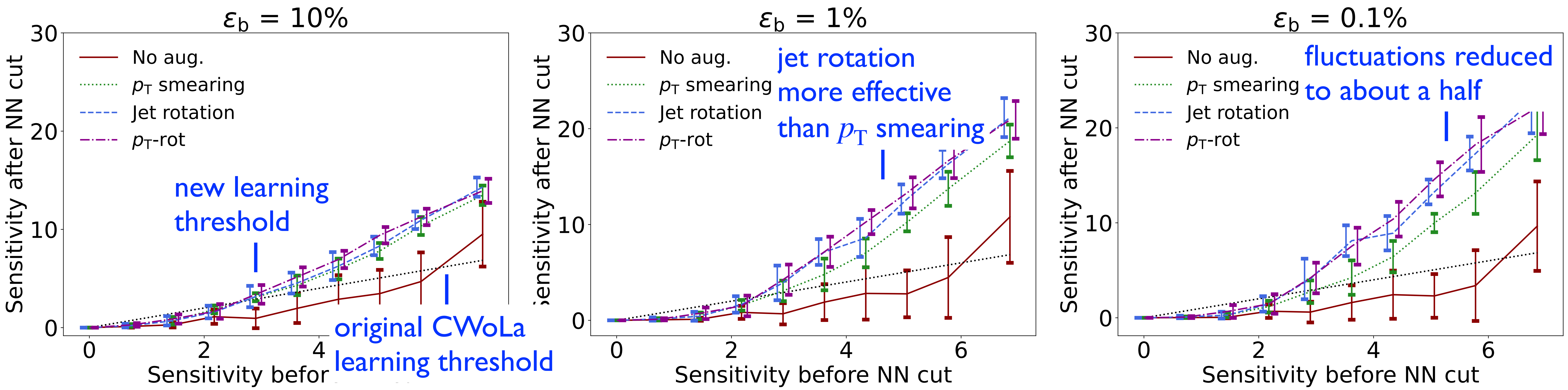
- Here we consider the "**+5 augmentation**," which means that the training dataset consists of the original data plus 5 augmented versions.
- The model's performance improves significantly even with just +5 augmentation.



ID; $\Lambda_D = 10$ GeV

Sensitivity Improvement

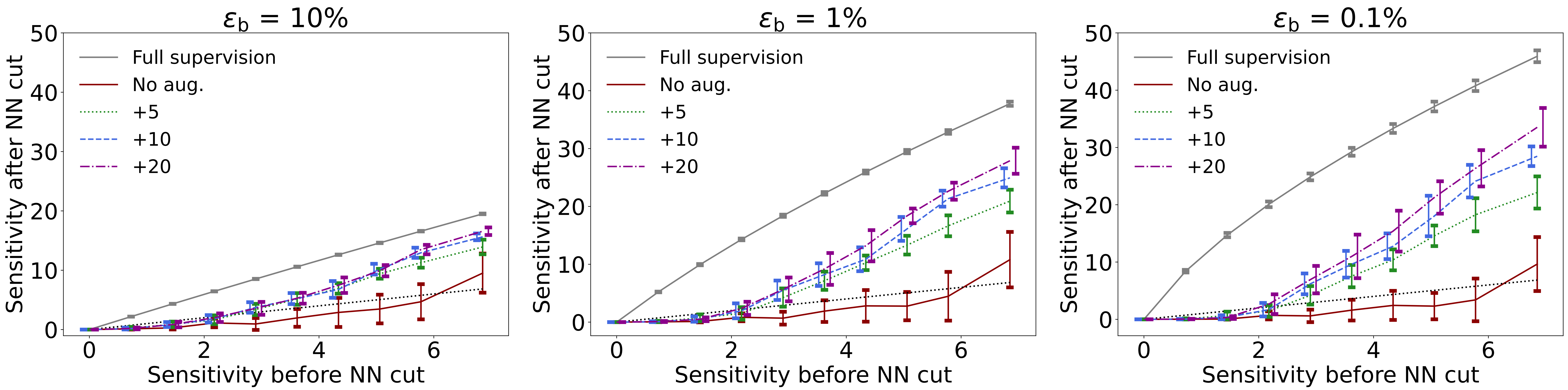
- Here we consider the "**+5 augmentation**," which means that the training dataset consists of the original data plus 5 augmented versions.
- The model's performance improves significantly even with just +5 augmentation.



ID; $\Lambda_D = 10$ GeV

Dependence on Augmentation Size

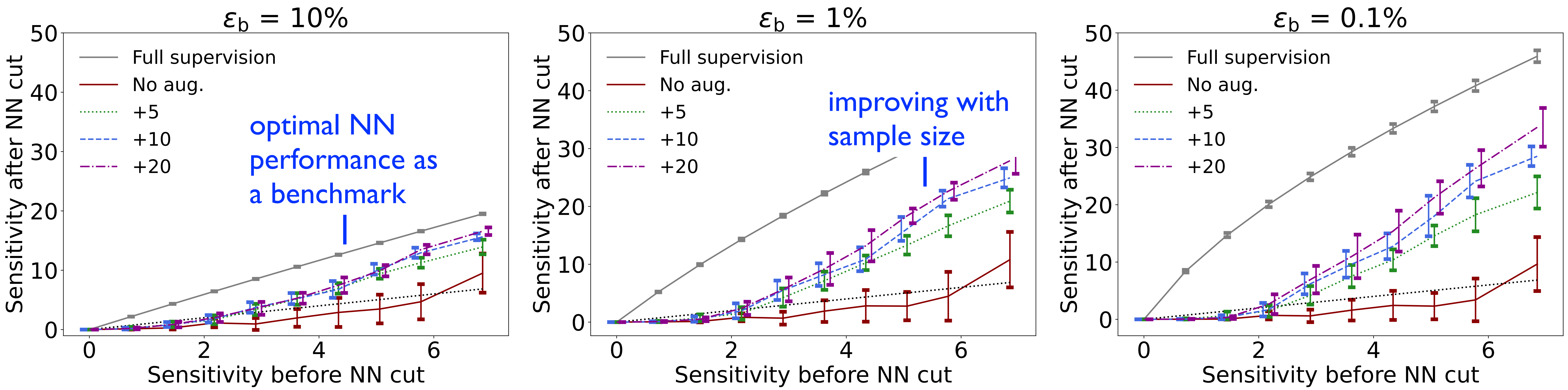
- Here, we focus on the "p_{T} smearing + jet rotation" augmentation method.
- The performance improvement is *not linear* in the augmentation size.
  ➠ "+5 augmentation" is already pretty effective



ID; $\Lambda_D = 10$ GeV

Dependence on Augmentation Size

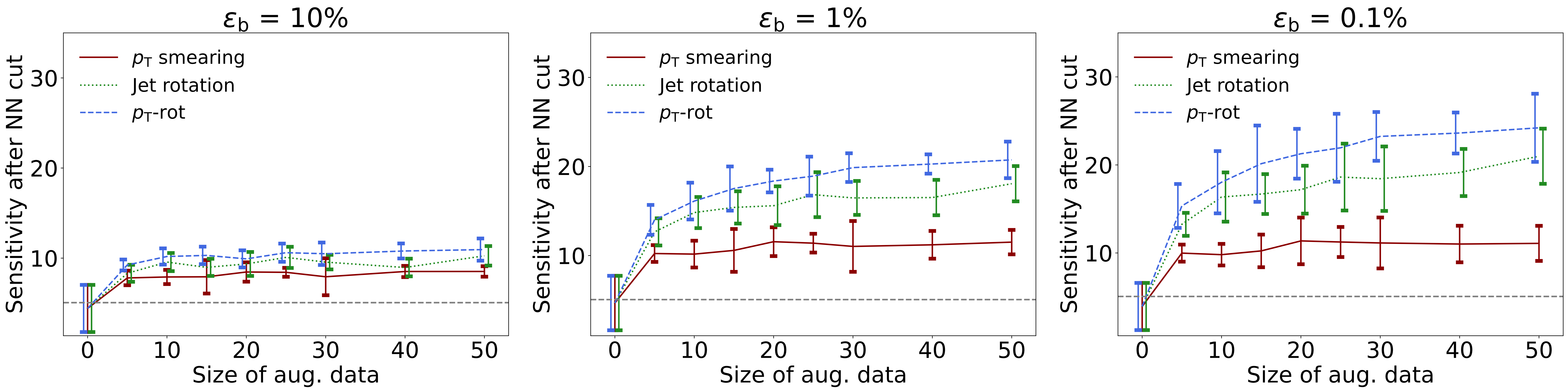
- Here, we focus on the "p_T smearing + jet rotation" augmentation method.
- The performance improvement is *not linear* in the augmentation size.
  ➠ "+5 augmentation" is already pretty effective



ID; $\Lambda_D = 10$ GeV

Asymptotic Behavior of Augmentation Size

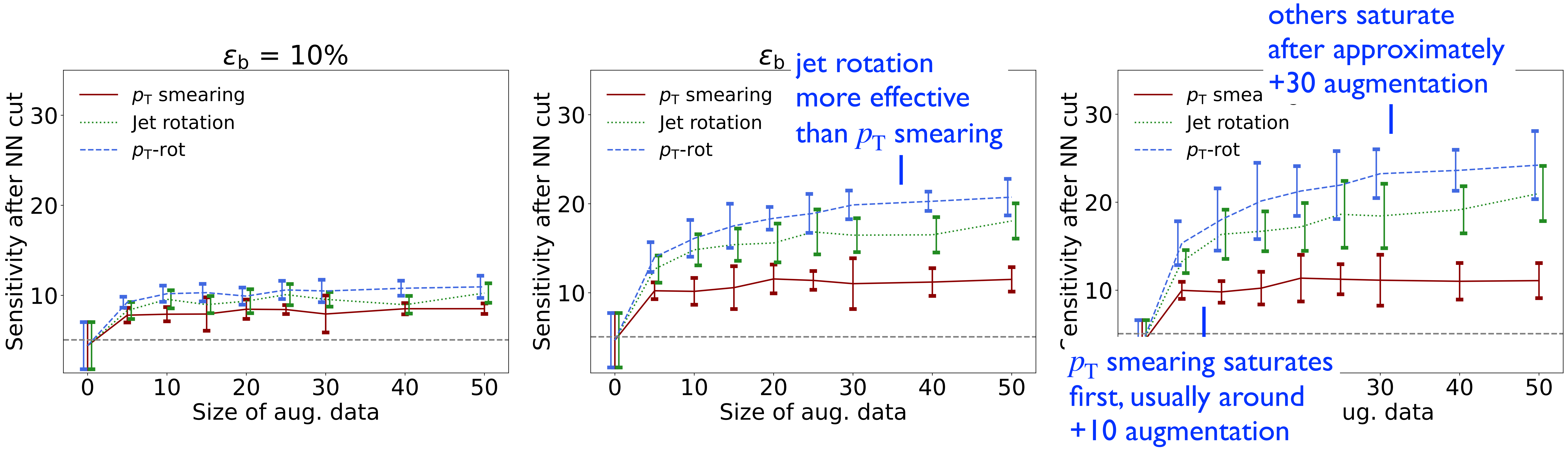
- Set the signal sensitivity to 5 before applying the NN selection.
- *A small sample augmentation* can already boost the sensitivity significantly, and there is no point in enlarging the dataset indefinitely.



ID; $\Lambda_D = 10$ GeV

Asymptotic Behavior of Augmentation Size

- Set the signal sensitivity to 5 before applying the NN selection.
- *A small sample augmentation* can already boost the sensitivity significantly, and there is no point in enlarging the dataset indefinitely.

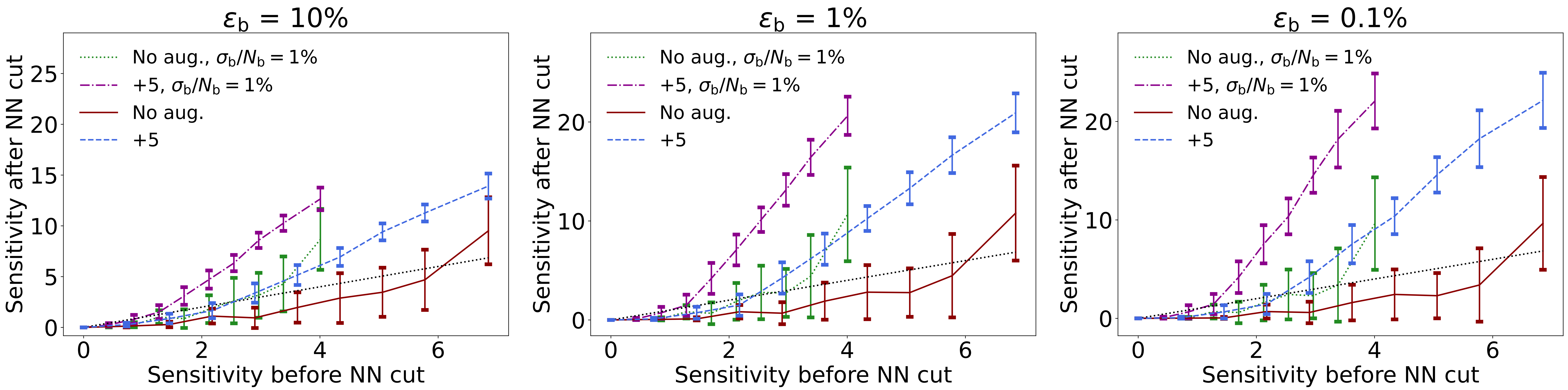


ID; $\Lambda_D = 10$ GeV

Impacts of Systematic Uncertainty

- Here, we consider a *relative background uncertainty* of 1% for illustration purposes, though the typical relative uncertainty is 5%.
- Data augmentation still significantly enhances the performance of NNs.

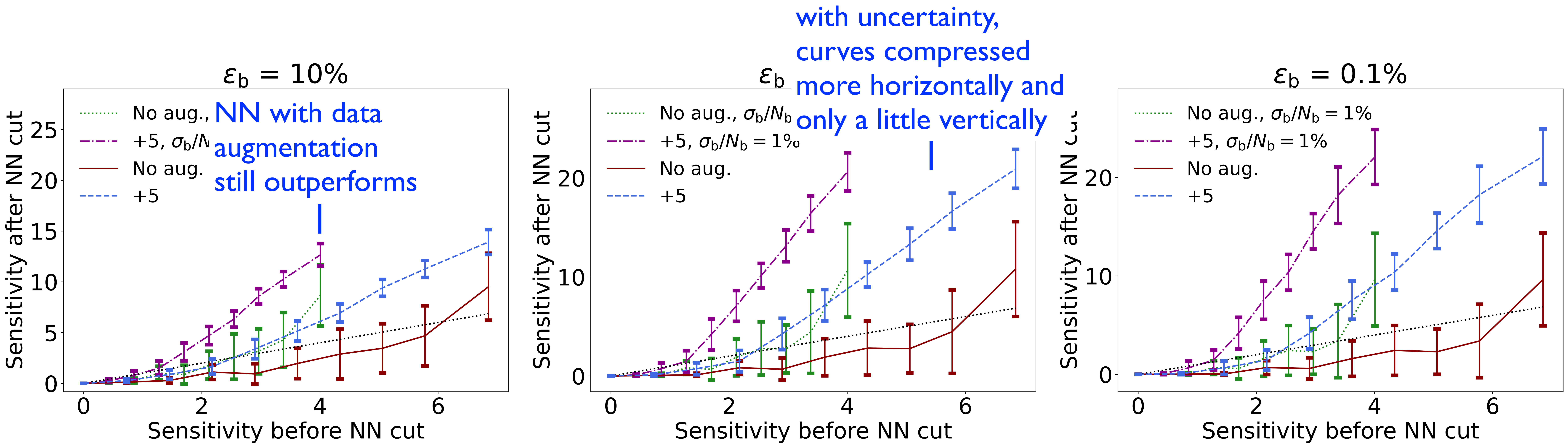
CMS 2020



Impacts of Systematic Uncertainty

- Here, we consider a *relative background uncertainty* of 1% for illustration purposes, though the typical relative uncertainty is 5%.
- Data augmentation still significantly enhances the performance of NNs.

CMS 2020



Summary

- **Weak supervision** (CWoLa) have the advantages of being able to **train on real data** and of exploiting distinctive signal properties.
  ➠ ideal tools for **anomaly searches**
  ➠ fail when signals are **limited**

- We propose to use the **transfer learning** (TL) technique and show that it can **drastically improve** the performance of CWoLa searches, particularly in the **low-significance region**, and that the amount of signal required for discovery can be reduced by a factor of a few (because of better identification of signals).

- We also propose to use the **data augmentation** technique and show that jet rotation is more effective than p_{T} smearing, that a mere +5 augmentation can already achieve great results, and that the NN still outperforms even when systematic background uncertainty is considered.

Thank You!