International Workshop on New Opportunities for Particle Physics 2025 @ IHEP
July 19, 2025

# Weak Supervision Techniques in Collider Physics

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

Refs:
CWC, David Shih and Shang-Fu Wei, PRD 107, 016014 (2023)
Hugues Beauchesne, Zong-En Chen, and CWC, JHEP 02 (2024) 138
Zong-En Chen, CWC, and Feng-Yang Hsieh, 2412.00198

# Outline

- Introduction
- Full supervision — an example
- Weak supervision — CWoLa
- Dark valley model — a physical model
- Transfer learning
- Data augmentation
- Summary

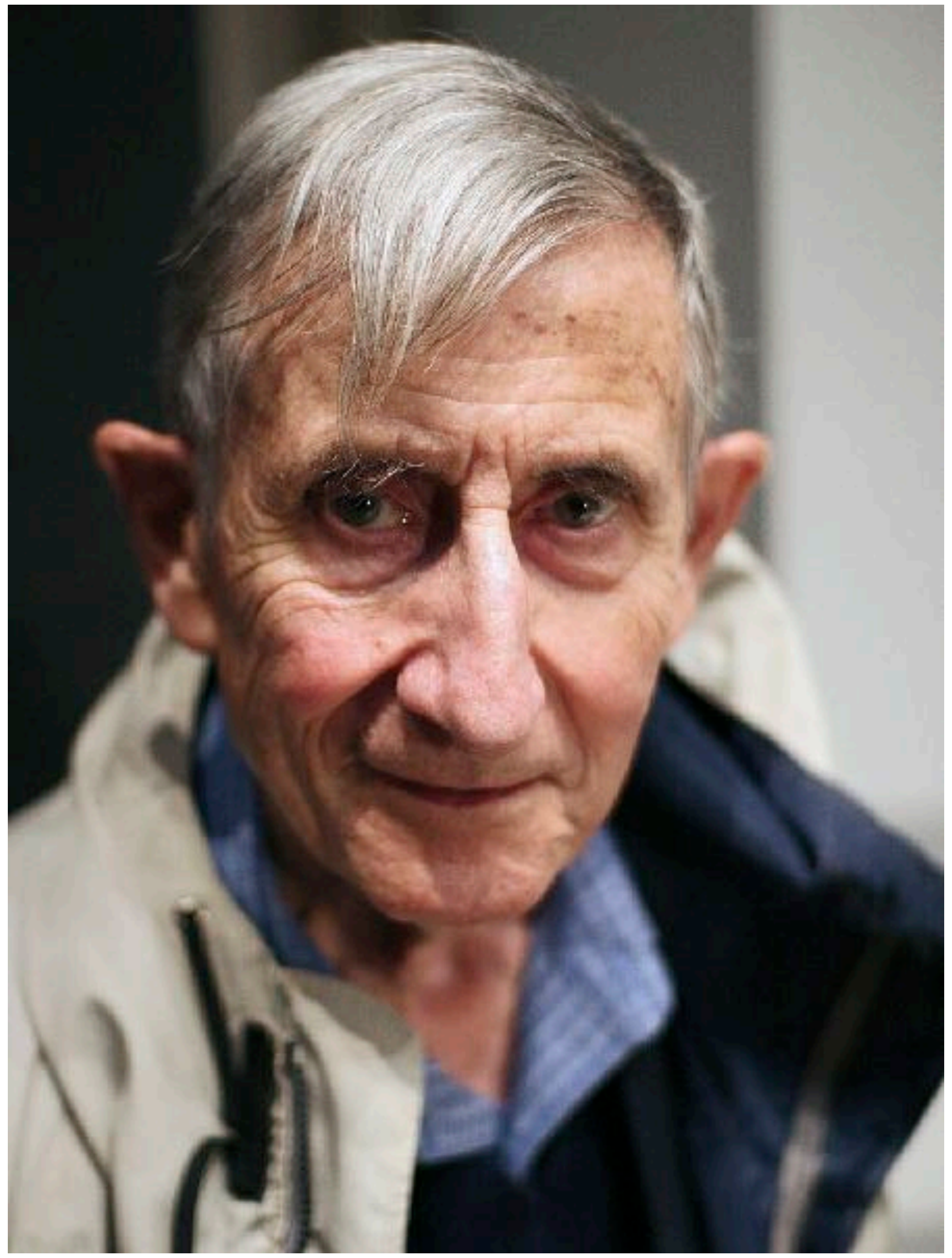
# Outline

- Introduction
- Full supervision — an example
- Weak supervision — CWoLa
- Dark valley model — a physical model
- Transfer learning
- Data augmentation
- Summary

# Revolution is Driven by New Tools

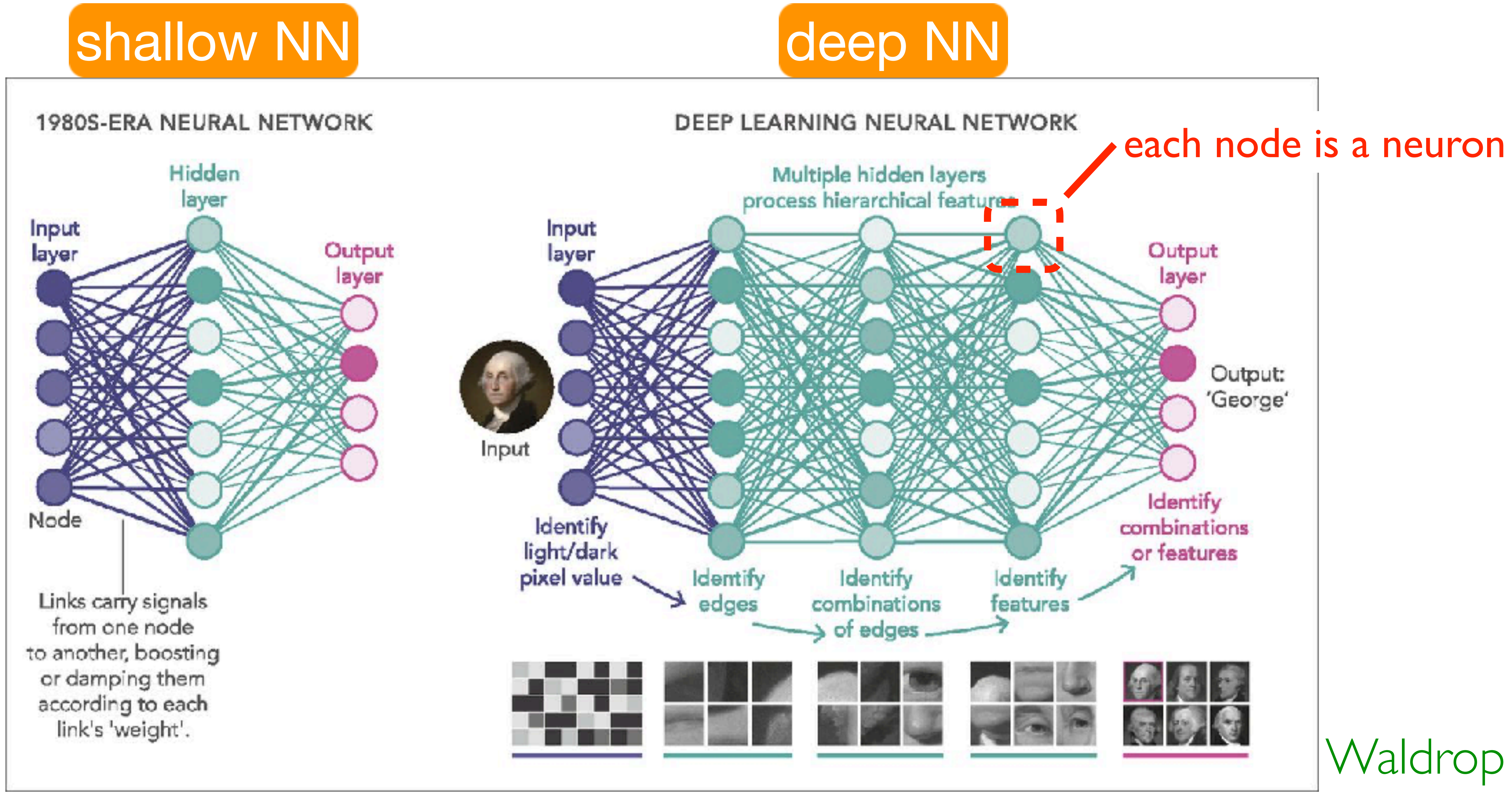
"New directions in science are launched by **new tools** much more often than by **new concepts**. The effect of a concept-driven revolution is to explain old things in new ways. The effect of a tool-driven revolution is to discover new things that have to be explained."

— Freeman J. Dyson, *Imagined Worlds*
Harvard University Press (1998)



# Machine Learning

- **Machine learning (ML)** is a new tool used for large-scale data processing and well-suited for complex datasets with huge numbers of **variables** and **features** (patterns and regularities), especially for **deep learning neural networks (NNs)**.
- **The Universal Theorem**: any function can be approximated by a neural network with at least one hidden layer.




Waldrop 2019

# Types of Machine Learning

- **Fully supervised learning**
  - Training data with labels (e.g., recognizing photos of cats and dogs)
- **Unsupervised learning**
  - Training data without labels (e.g., analyzing and clustering unlabeled datasets)
- **Reinforced learning**
  - Data from interactions with the environment (e.g., chess and Go games)

# Types of Machine Learning

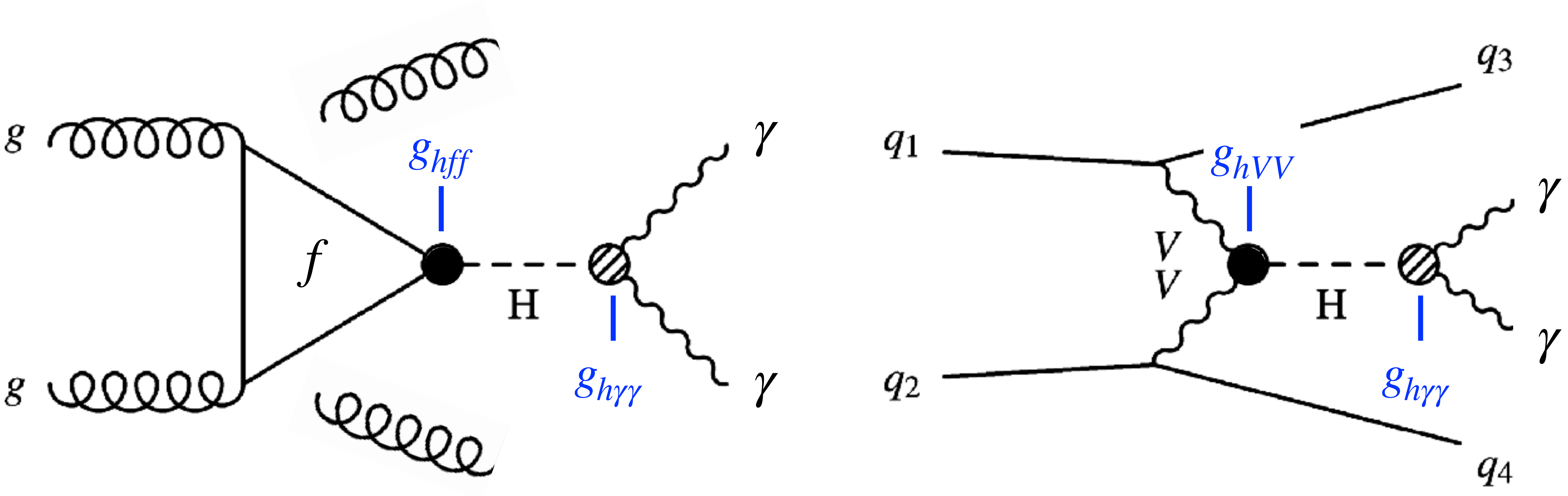
- **Fully supervised learning**
  - Training data with labels (e.g., recognizing photos of cats and dogs)
- **Unsupervised learning**
  - Training data without labels (e.g., analyzing and clustering unlabeled datasets)
- **Reinforced learning**
  - Data from interactions with the environment (e.g., chess and Go games)
- **Weakly supervised learning**
  - Training data whose labeling is *infeasible*, *imperfect*, *difficult*, or *expensive* (e.g., medical imaging, identifying celestial objects from low-quality telescope images, anomaly searches)

# Outline

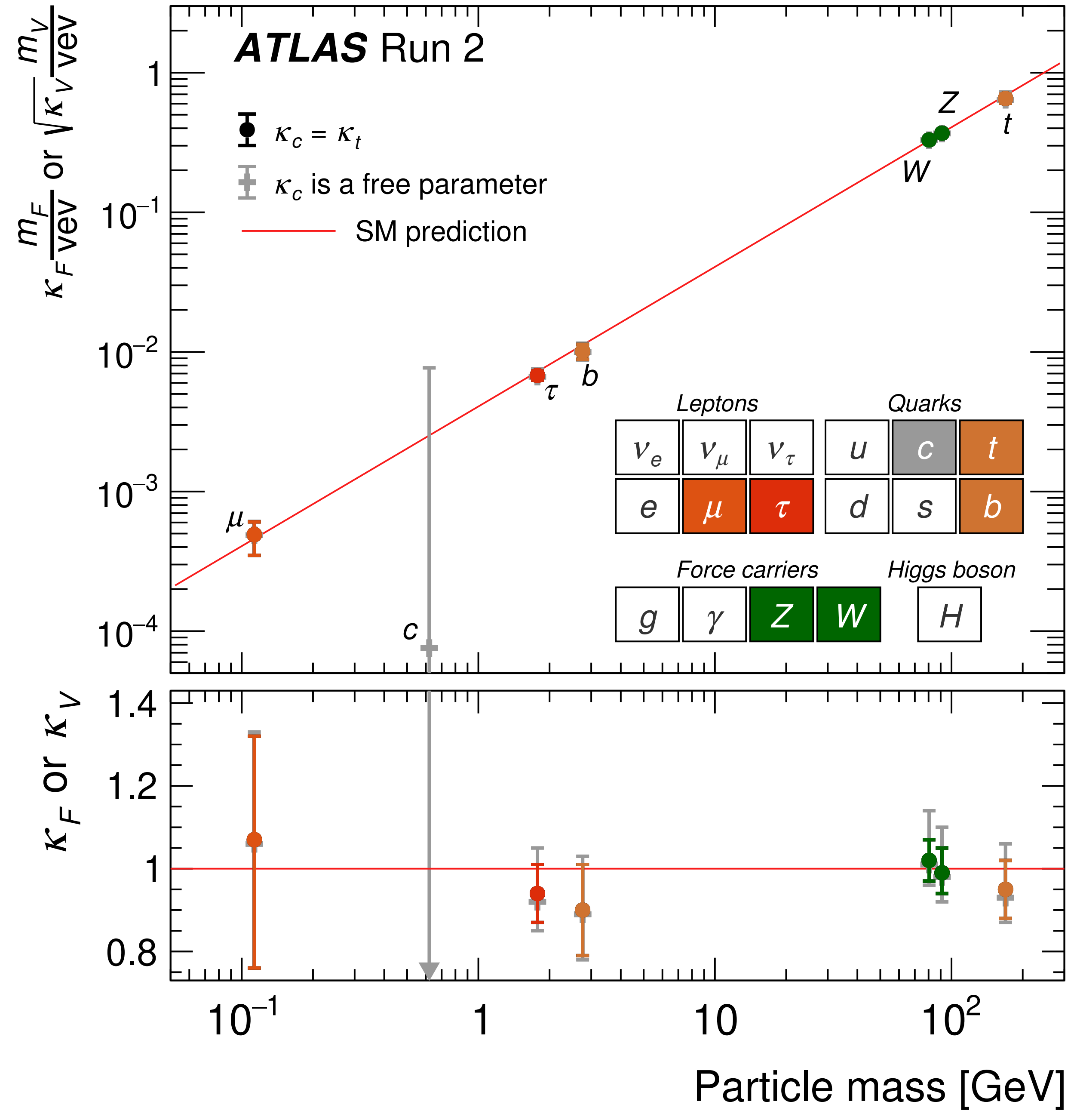
- Introduction
- Full supervision — an example
- Weak supervision — CWoLa
- Dark valley model — a physical model
- Transfer learning
- Data augmentation
- Summary

# 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?




(a) ggF production (b) VBF production

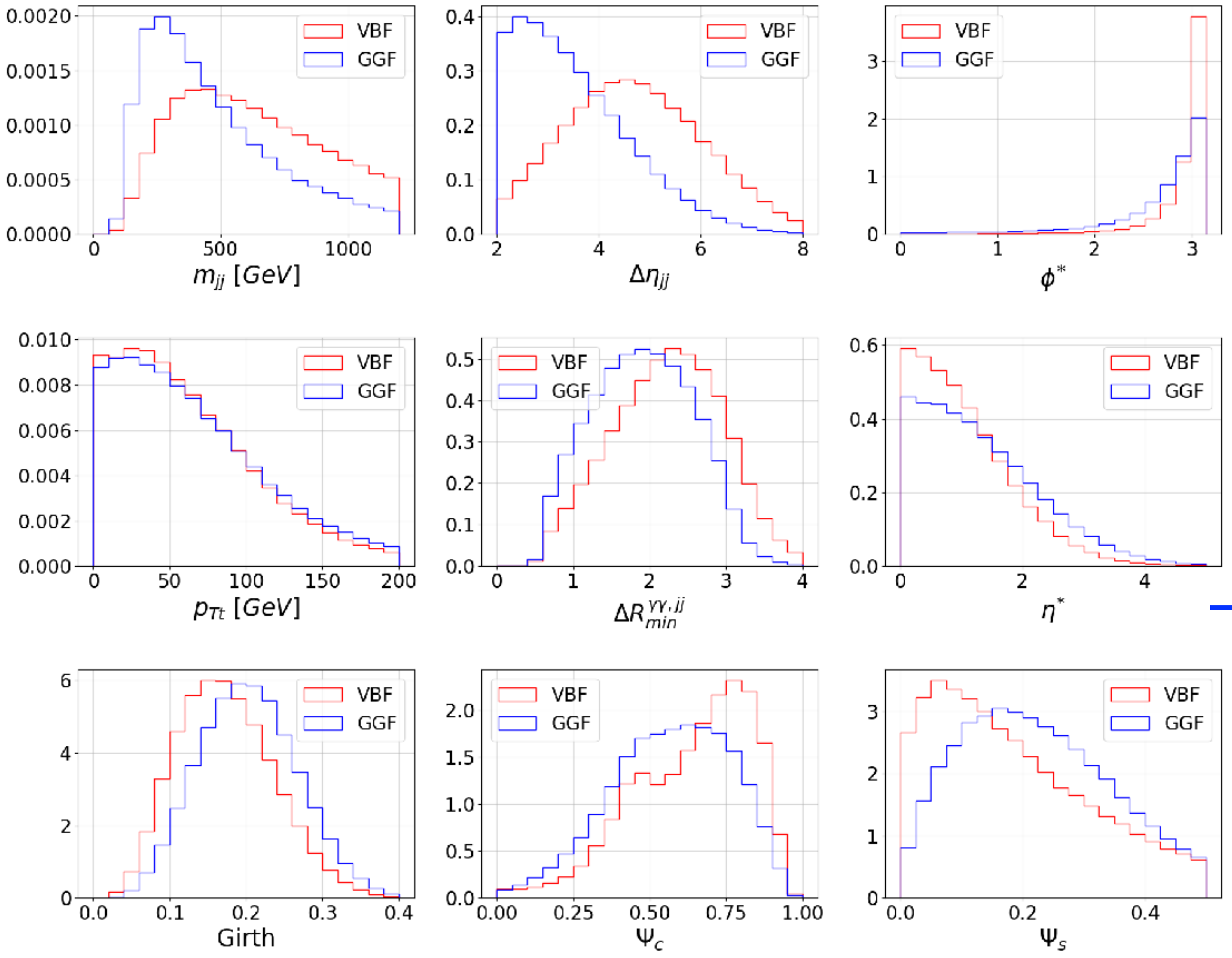



ATLAS 2019

# Distributions of BDT Input Variables

- Cut-based methods cannot reach high purity.
- BDT-based methods can achieve a purity of about 70% for the VBF sample, depending on the decay channel. ATLAS 2019

baseline

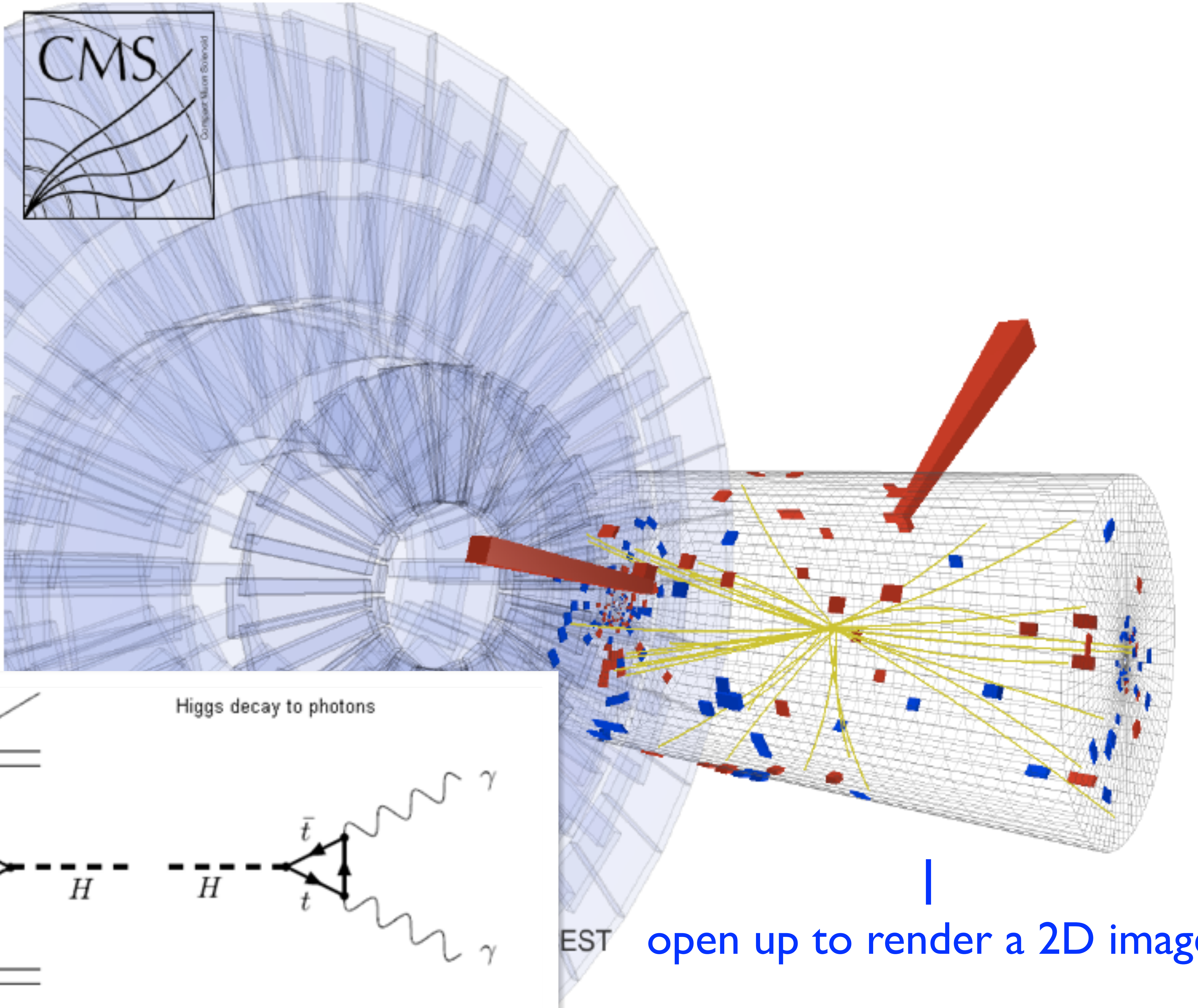



shapes

human-engineered kinetic variables

all histograms normalized to have unit area under the curves

# A Higgs to Diphoton Event




Event parameters:

$M_{\gamma\gamma}$ = 125.9 GeV

$p_T^{\gamma 1}$ = 89.8 GeV

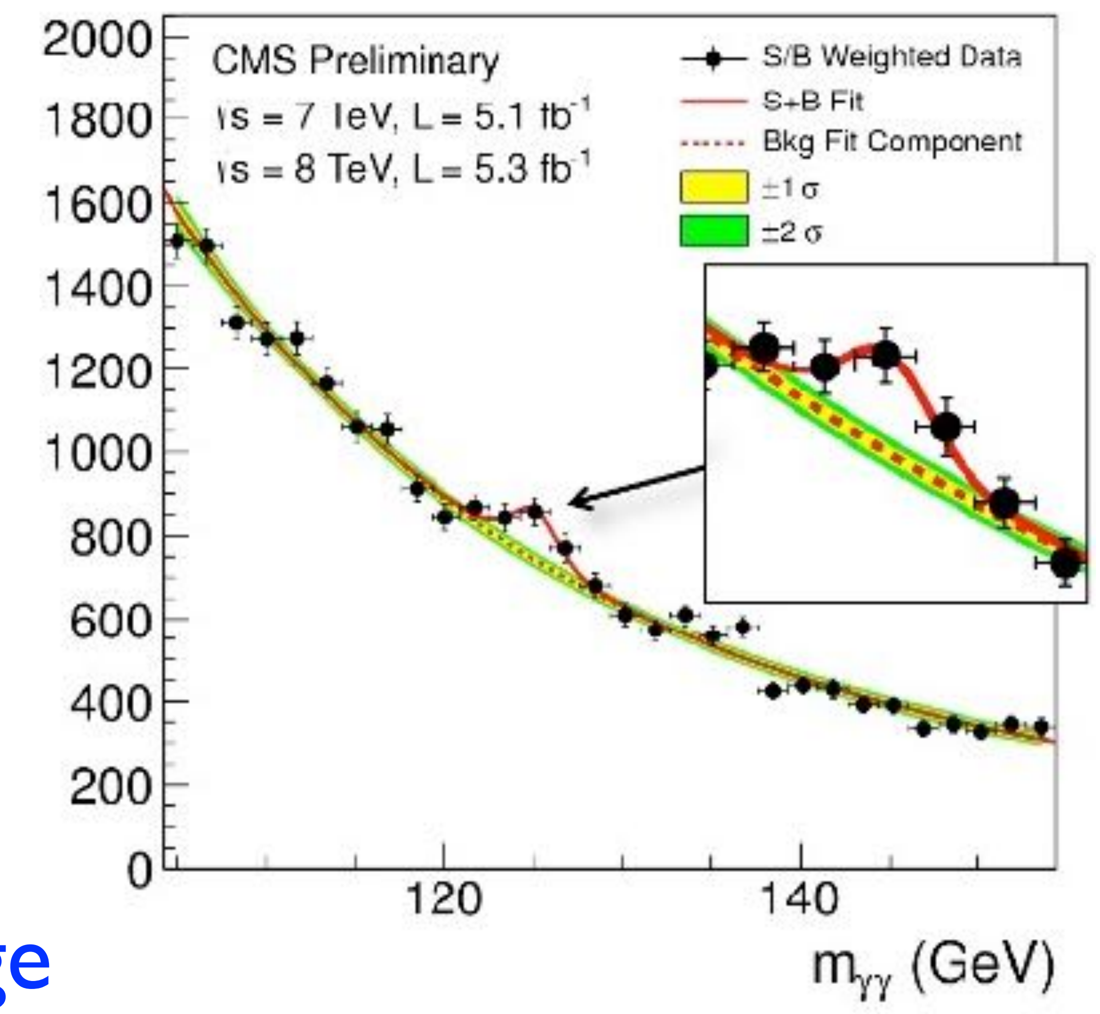
$p_T^{\gamma 2}$ = 46.5 GeV

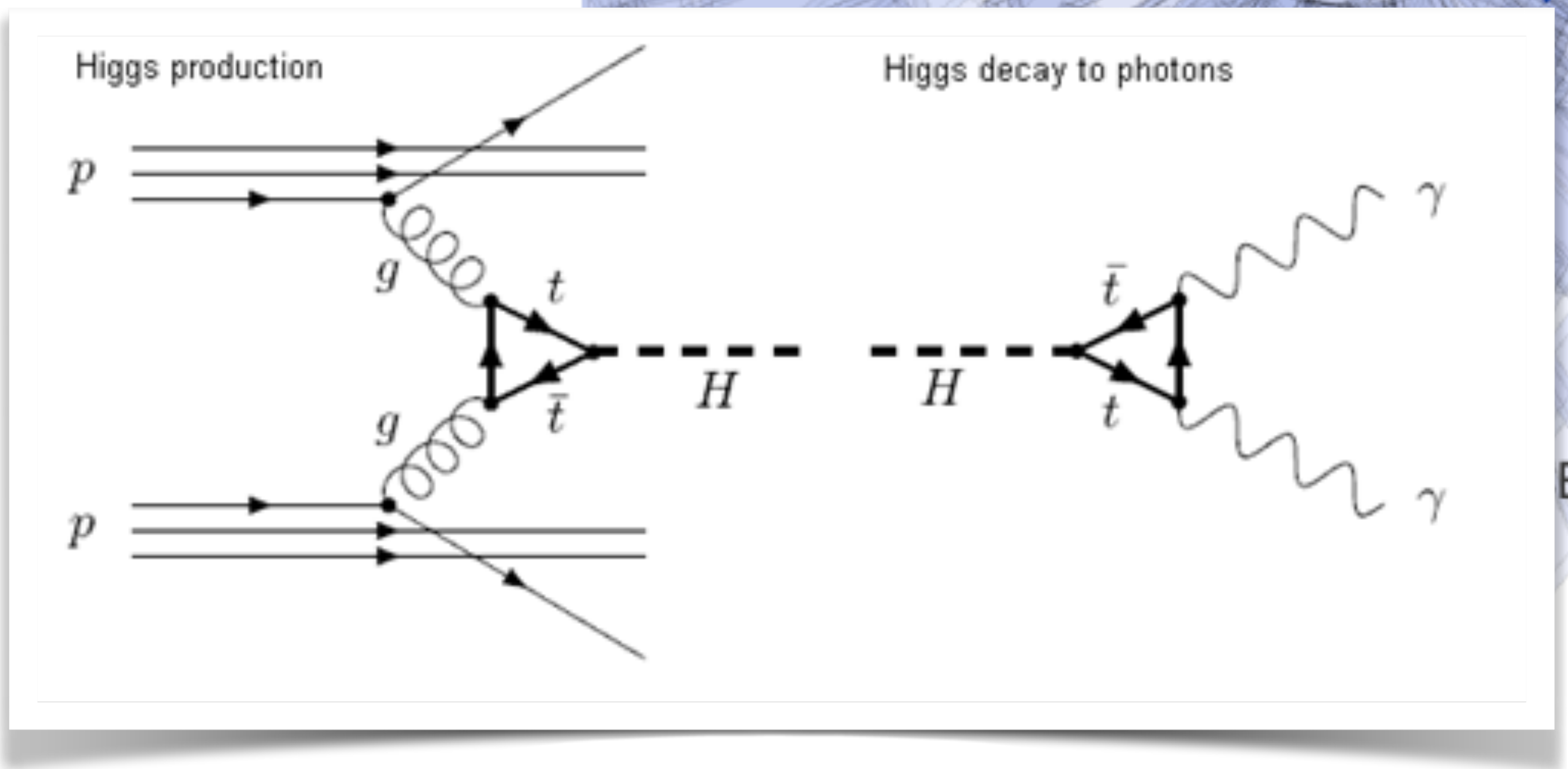
$\eta_{\gamma 1}$ = 0.06

$\eta_{\gamma 2}$ = -0.81

$\sigma_M/M$ =0.89%

$p_T^{\gamma\gamma}$ =78.4 GeV



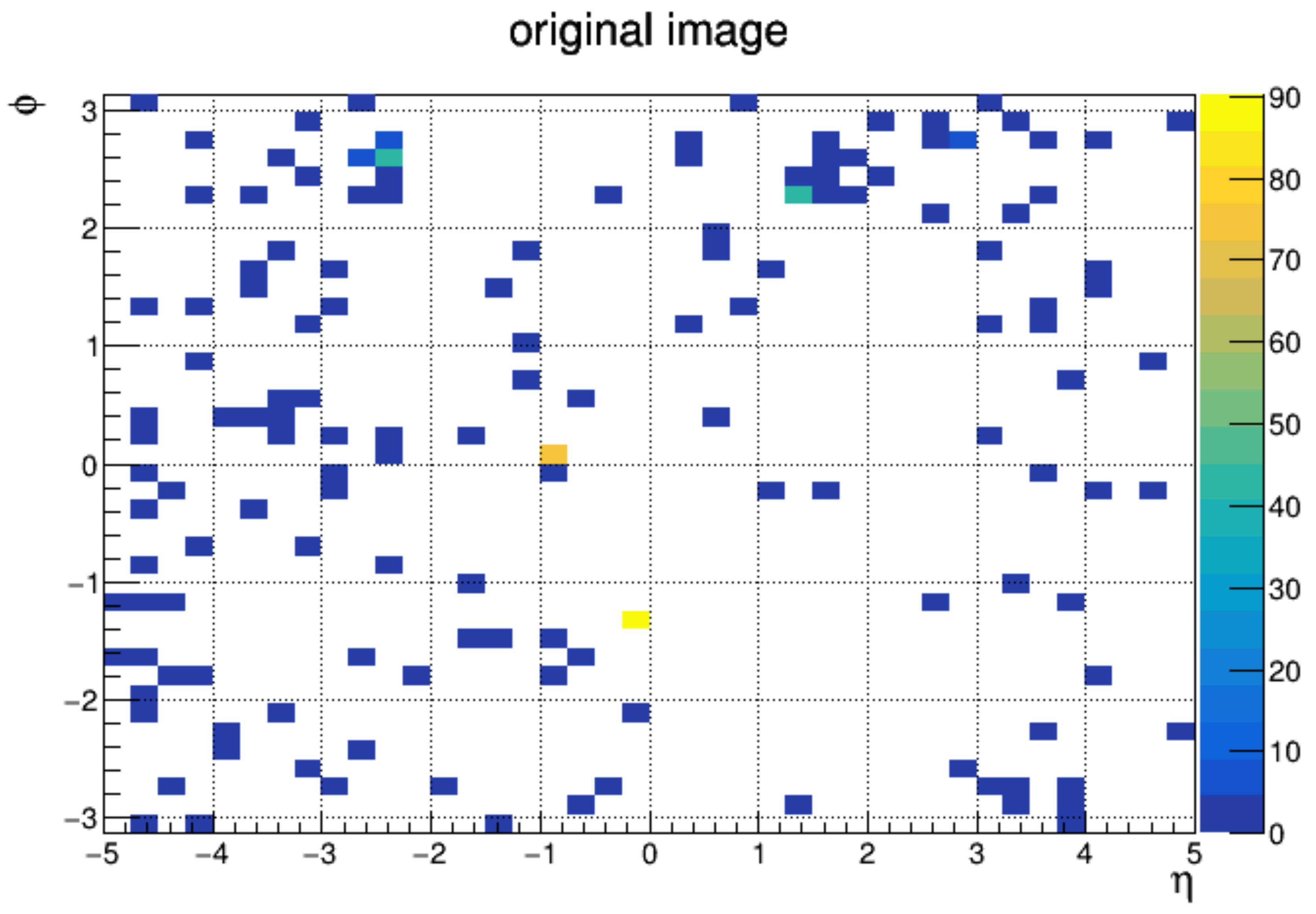



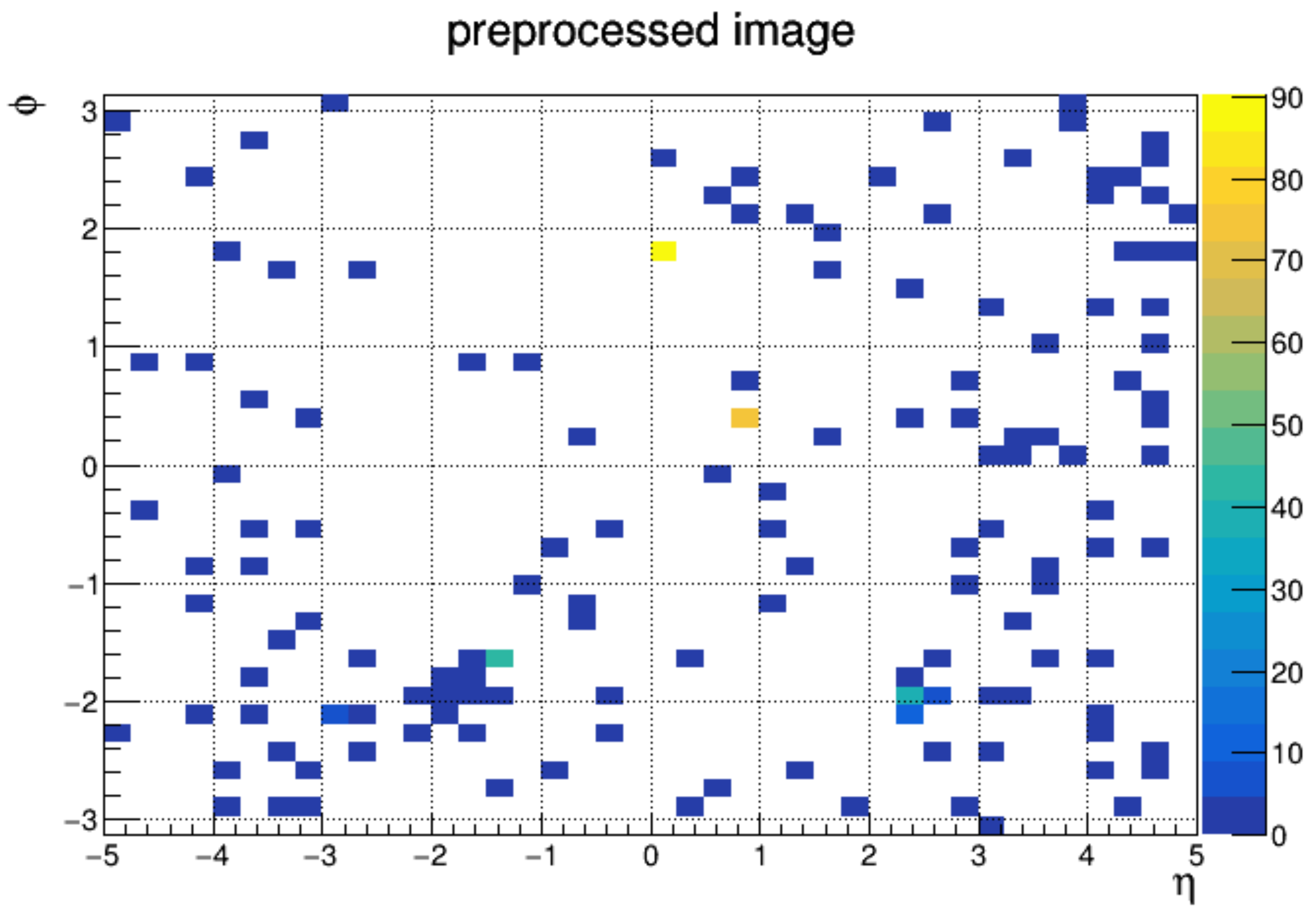

open up to render a 2D image

# 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.

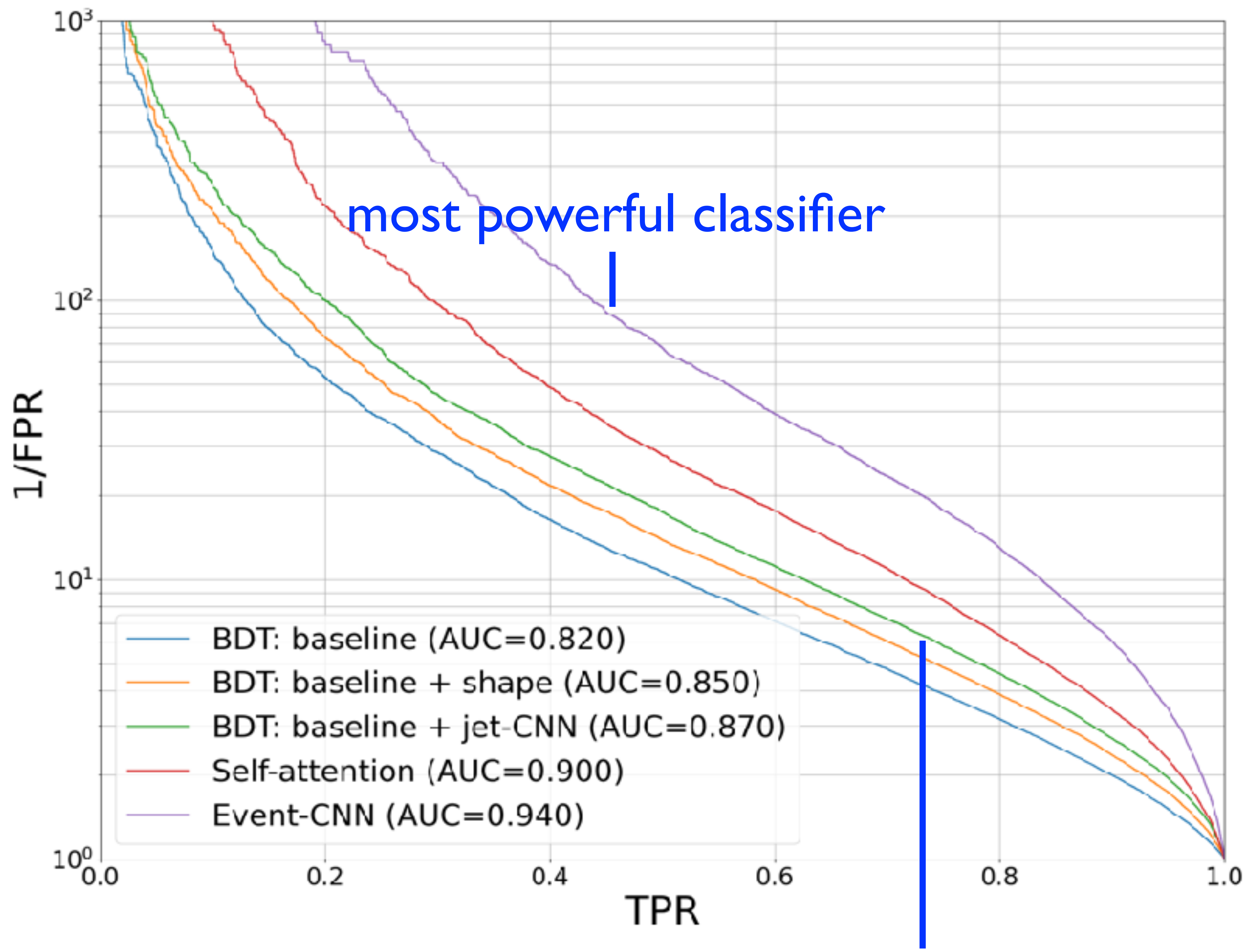
| | training | validation | testing |
|---|---|---|---|
| VBF events | 105k | 26k | 33k |
| GGF events | 83k | 21k | 26k |

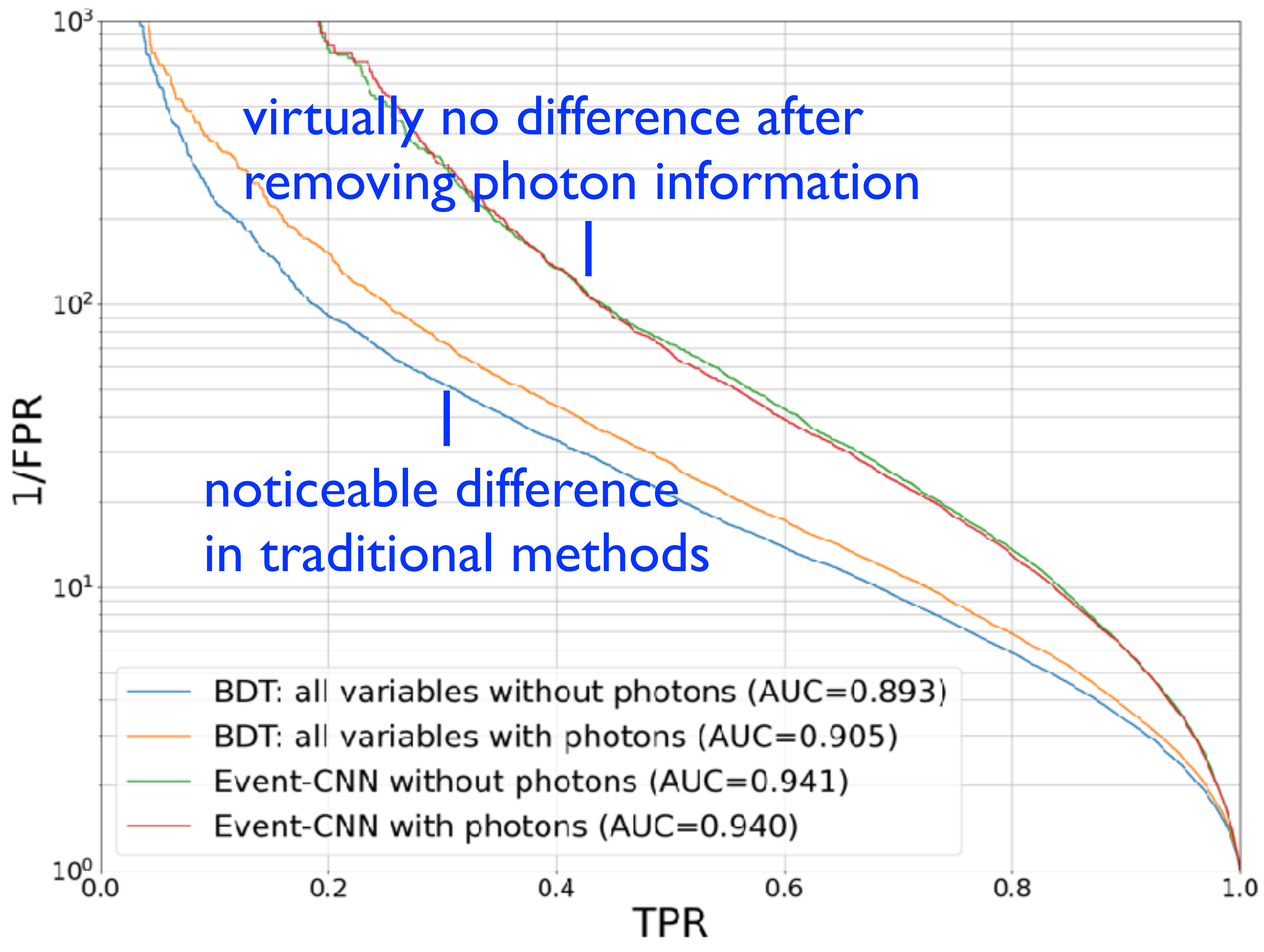


# Comparison of Classifiers

ROC curves (Receiver Operating Characteristic curves) ROC curves




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

CWC, Shih, Wei 2023

# Outline

- Introduction
- Full supervision — an example
- Weak supervision — CWoLa
- Dark valley model — a physical model
- Transfer learning
- Data augmentation
- Summary

# 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

https://www.catbreedslist.com/stories/what-breed-of-cat-is-garfield.html

# 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

https://www.catbreedslist.com/stories/what-breed-of-cat-is-garfield.html

- 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)

# 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

https://www.catbreedslist.com/stories/what-breed-of-cat-is-garfield.html

- 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)

https://en.wikipedia.org/wiki/Garfield_(character)

# Can We Be More Realistic?

# Can We Be More Realistic?

- Use a **generative adversarial network** (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 a **generative adversarial network** (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**…

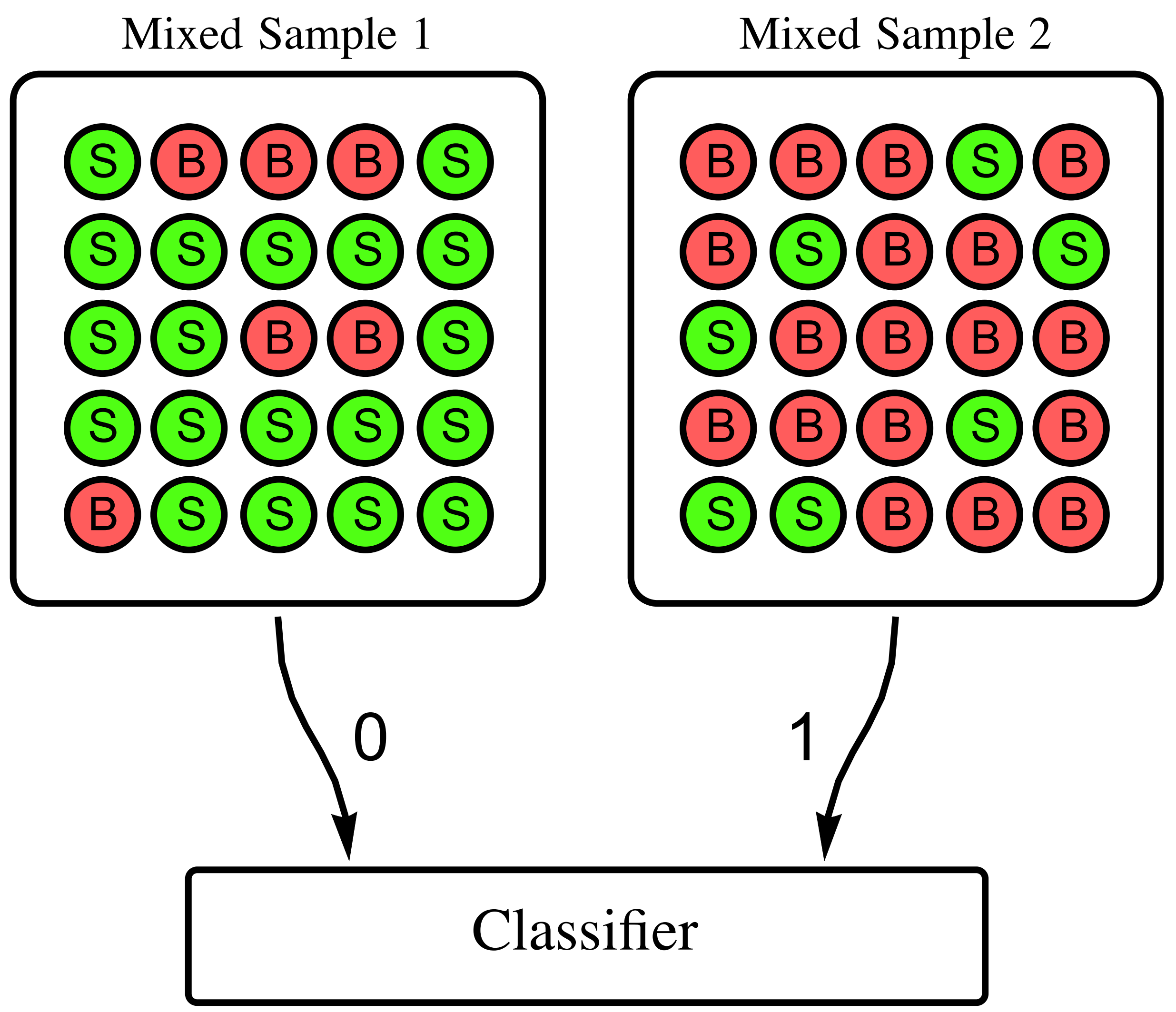
# Can We Be More Realistic?

- Use a **generative adversarial network** (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**). 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 $B$.

# Proof

- The *optimal classifiers* to distinguish examples drawn from $p_{M_1}$ and $p_{M_2}$ and to distinguish examples drawn from $p_S$ and $p_B$ are, respectively, the likelihood ratios

$$L_{M_1/M_2}(\vec{x}) = \frac{p_{M_1}(\vec{x})}{p_{M_2}(\vec{x})} \quad \text{and} \quad L_{S/B}(\vec{x}) = \frac{p_S(\vec{x})}{p_B(\vec{x})}$$

— Neyman-Pearson lemma

- Where $p_B$ has support, these two likelihood ratios are related:

$$L_{M_1/M_2} = \frac{p_{M_1}}{p_{M_2}} = \frac{f_1 p_S + (1 - f_1)\, p_B}{f_2 p_S + (1 - f_2)\, p_B} = \frac{f_1 L_{S/B} + (1 - f_1)}{f_2 L_{S/B} + (1 - f_2)} = \frac{f_1 \left(L_{S/B} - 1\right) + 1}{f_2 \left(L_{S/B} - 1\right) + 1}$$

which is a *monotonically increasing* function of $L_{S/B}$ as long as $f_1 > f_2$, since

$$\frac{\partial L_{M_1/M_2}}{\partial L_{S/B}} = \frac{f_1 - f_2}{\left(f_2 L_{S/B} - f_2 + 1\right)^2} > 0$$

- If $f_1 < f_2$, then one obtains the *reversed* classifier.

➠ $L_{S/B}$ and $L_{M_1/M_2}$ are **effectively equivalent classifiers**

this can be trained with full supervision

# 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**?

# Outline

- Introduction
- Full supervision — an example
- Weak supervision — CWoLa
- Dark valley model — a physical model
- Transfer learning
- Data augmentation
- Summary

# Dark Valley Model and Dark Jets

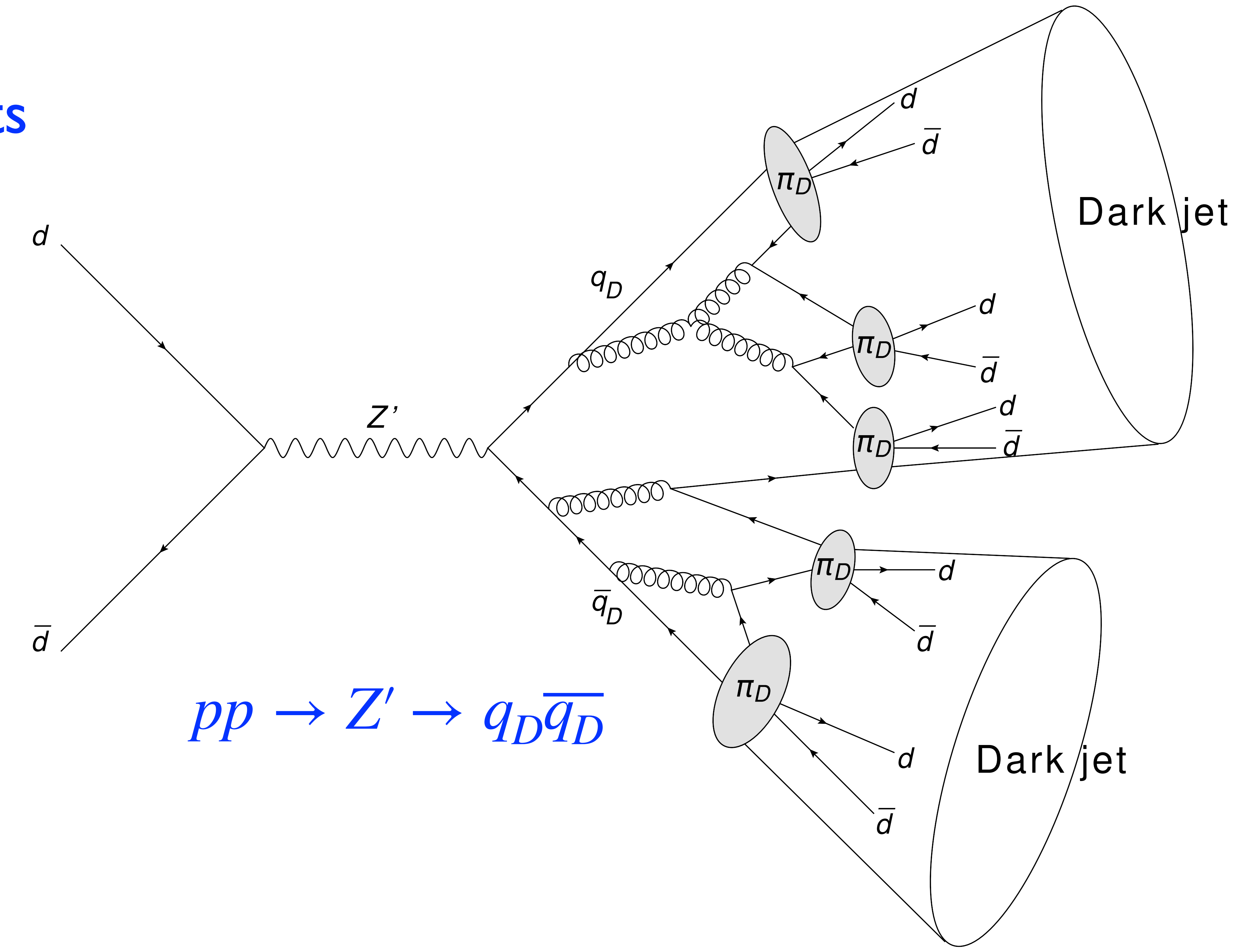
- Assume the existence of a **dark confining sector** that communicates with the visible sector via a **heavy $Z'$ portal**:

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 $\rho_D$ and pseudoscalar $\pi_D$ masses and two (prompt) decay scenarios:

Albouy et al 2022

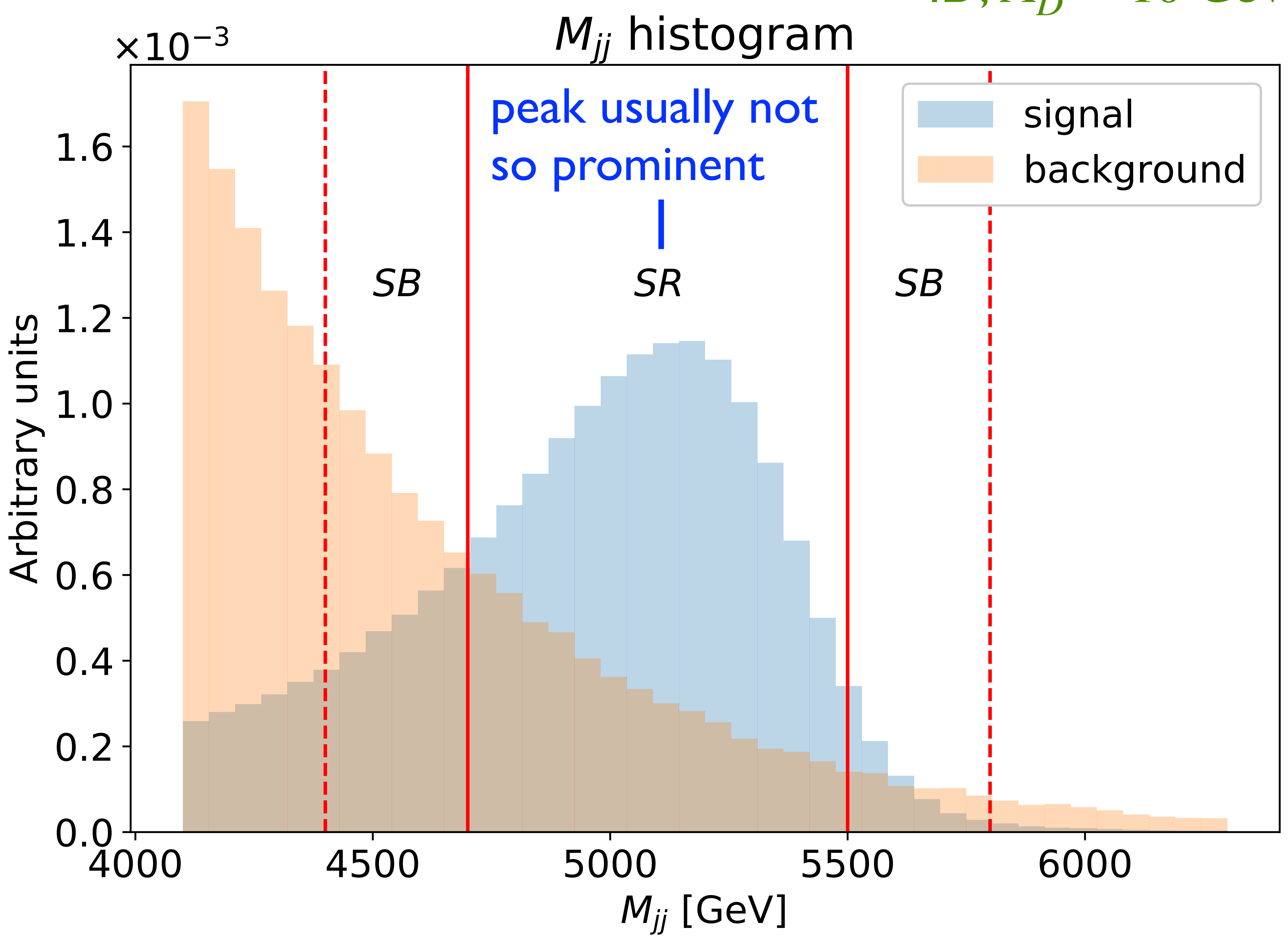
$$\frac{m_{\rho_D}}{\Lambda_D} = \sqrt{5.76 + 1.5\frac{m_{\pi_D}^2}{\Lambda_D^2}}$$

  - **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 $m_{\pi_D}/\Lambda_D = 1.8$

- Totally **14 "models"** from different combinations of the above parameters.

# Dijet Invariant Mass Distributions

ID; $\Lambda_D = 10$ GeV




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

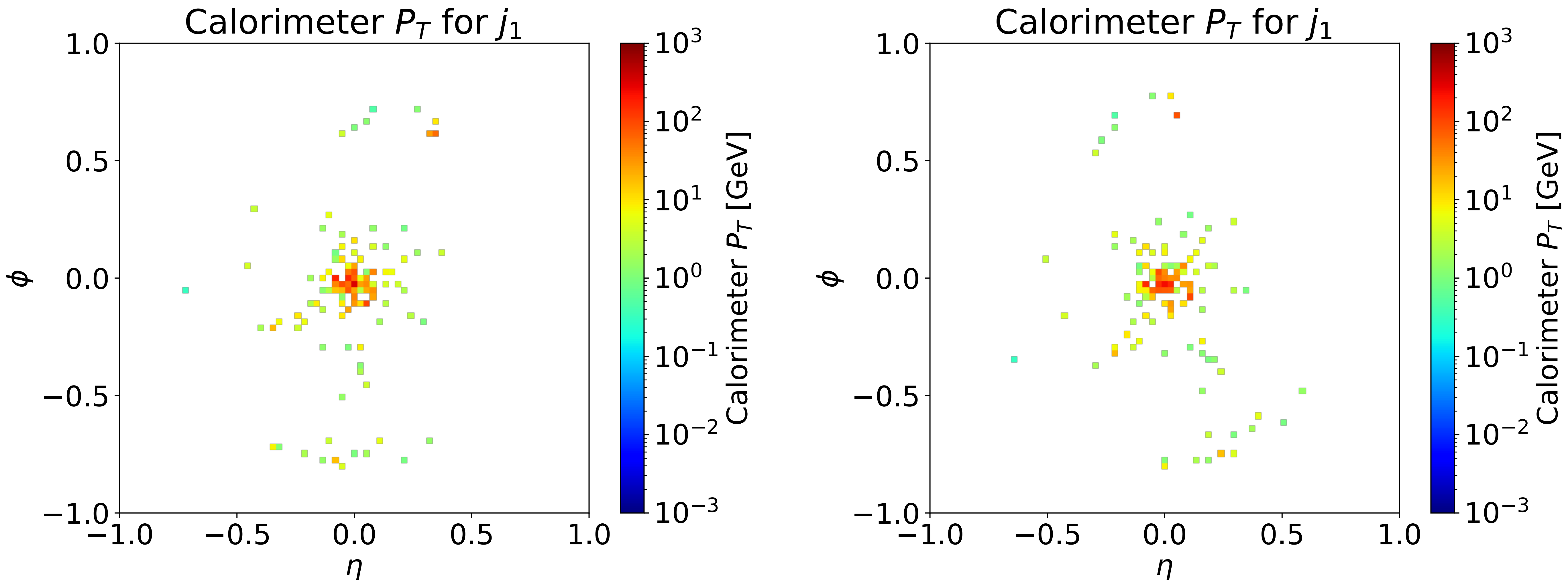
- `Madgraph 2.7.3` with PDF = `NN23LO1`
- `Pythia 8.307` with default settings
- `Delphes 3.4.2` with default CMS card and jet radius $R = 0.8$

Probability distributions of 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.

# Convolutional + Dense Layers

- Prepare each jet image in **three resolutions**: $25 \times 25$, $50 \times 50$, $75 \times 75$.
- 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.




(a) Before preprocessing. (b) After preprocessing.

Image of one signal jet in SR

$\Lambda_D = 10$ GeV

Resolution = $75 \times 75$

* 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.

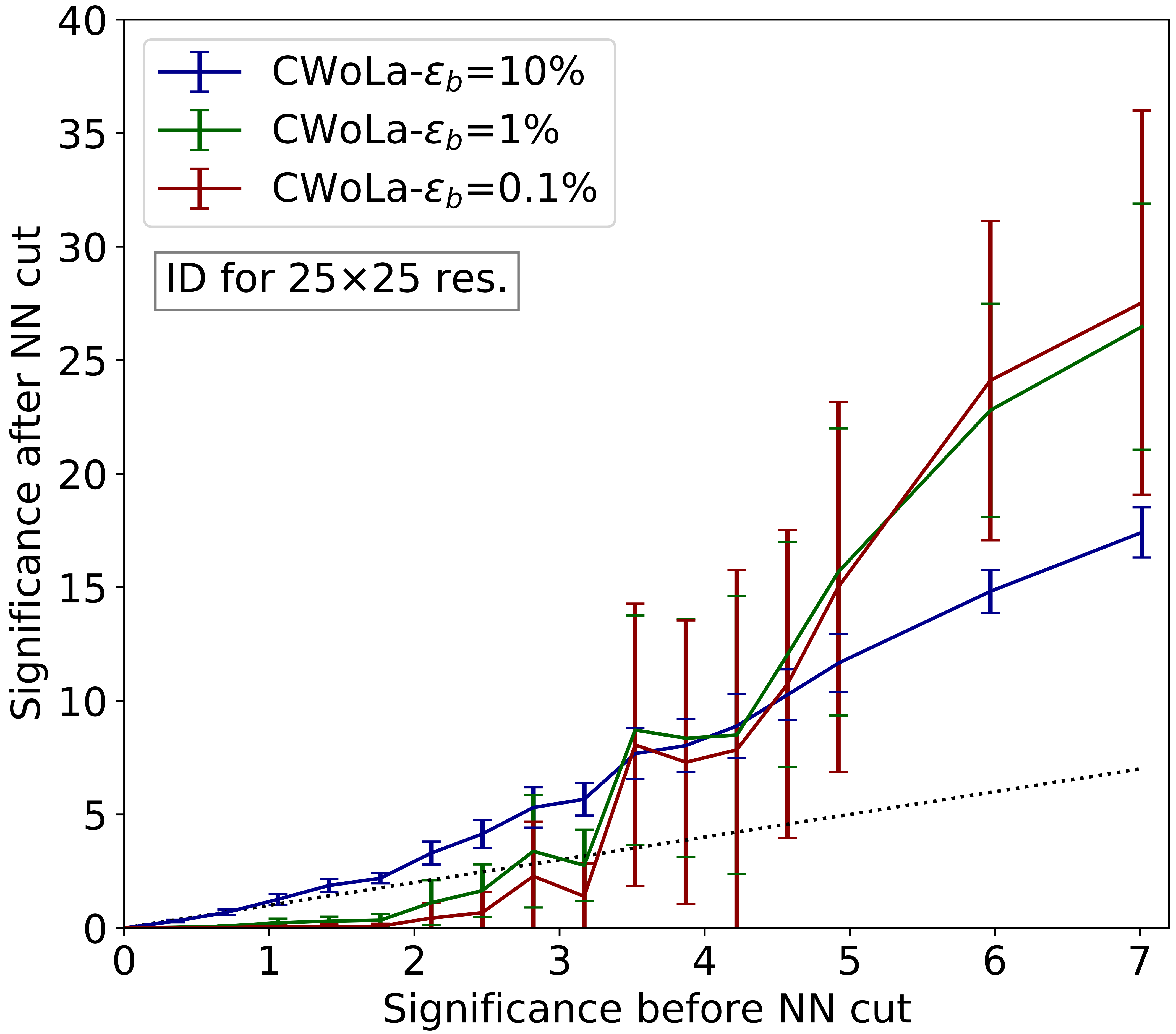
# Convolutional + Dense Layers

- The convolutional part of the NN is referred to as the **feature extractor**, and its weights and biases are collectively labeled as $\Theta$.
  ➠ to be **transferred** later
- The dense layer part of the NN is referred to as the **classifier**, and its weights and biases of the dense layers are collectively labeled as $\theta$.
  ➠ to be **fine-tuned** later

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

# Results of Regular CWoLa

Beauchesne, Chen, CWC 2024




ID; $\Lambda_D = 10$ GeV

# Results of Regular CWoLa

Beauchesne, Chen, CWC 2024

try different background efficiencies

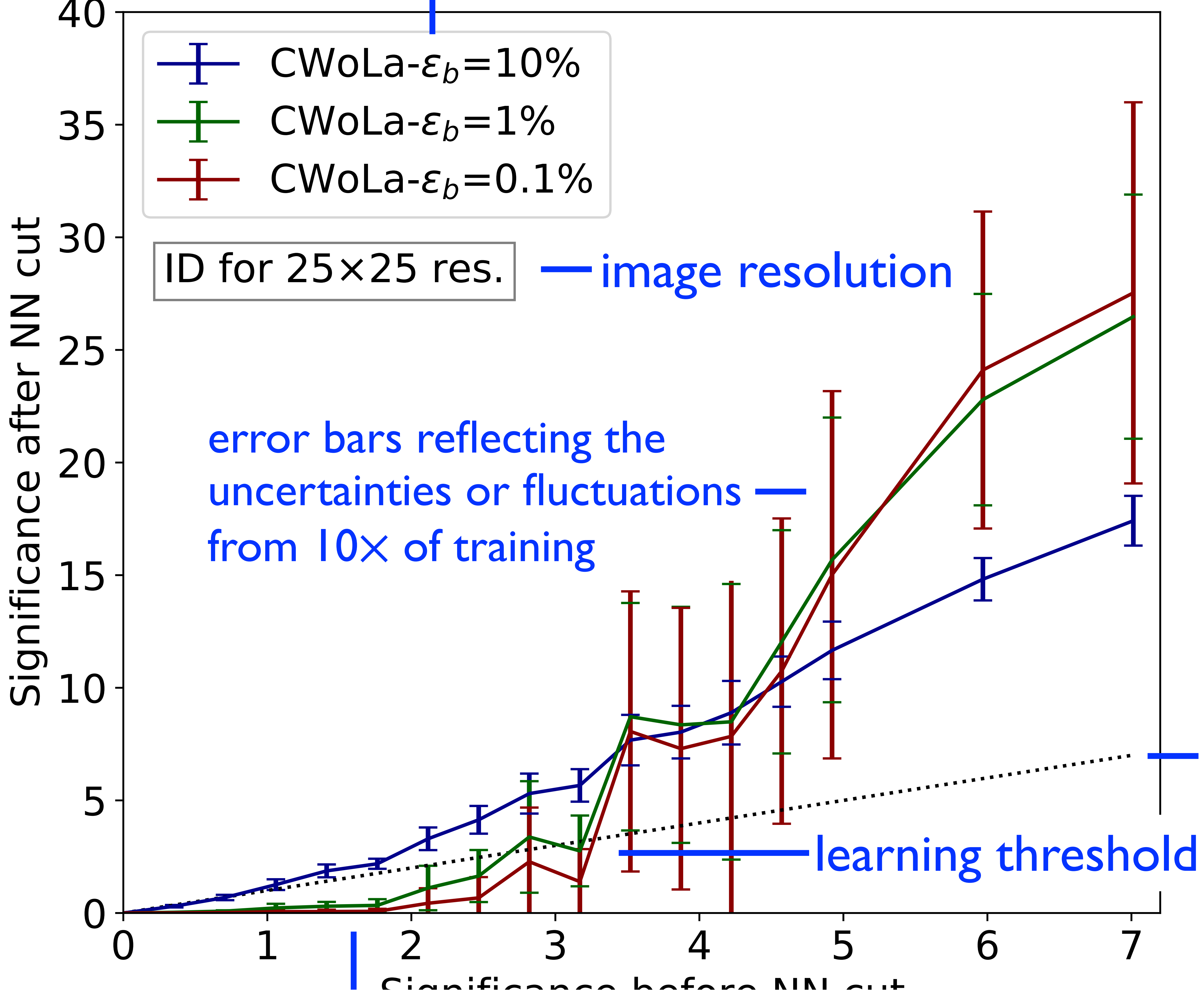



image resolution

error bars reflecting the uncertainties or fluctuations from 10× of training

slope = 1

learning threshold

below learning thresholds, NN fails to learn from data as it cuts background and signal indiscriminately

# Outline

- Introduction
- Full supervision — an example
- Weak supervision — CWoLa
- Dark valley model — a physical model
- Transfer learning
- Data augmentation
- Summary

# Introduction to Transfer Learning

- The phrase "**transfer learning (TL)**" comes from **psychology**.
  ➠ a learner new to a fresh topic (e.g., riding a motorcycle or playing guitar) typically has a higher learning threshold, while a learner experienced in related topics (e.g., riding a bicycle or playing violin) 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** and **fine-tuned** for the second task.

# 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 $\Lambda_D$), 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 $\Theta$ (from convolutional layers) and $\theta$ (from dense layers)

| | | |
|---|---|---|
| Layers of CNN subnetwork | $\begin{pmatrix} \text{convolutional 2D layer: 64 filters with } 5\times5 \text{ kernel size} \\ \text{maxpooling layer: } 2\times2 \text{ pool size} \end{pmatrix} \times 2$ | $\Theta$ |
| | convolutional 2D layer: 128 filters with $3\times3$ kernel size | |
| | maxpooling layer: $2\times2$ pool size | |
| | convolutional 2D layer: 128 filters with $3\times3$ kernel size | |
| | flatten layer | |
| | (dense layer: 128 units) $\times$ 3 | $\theta$ |
| | dense layer (output): 1 unit | |

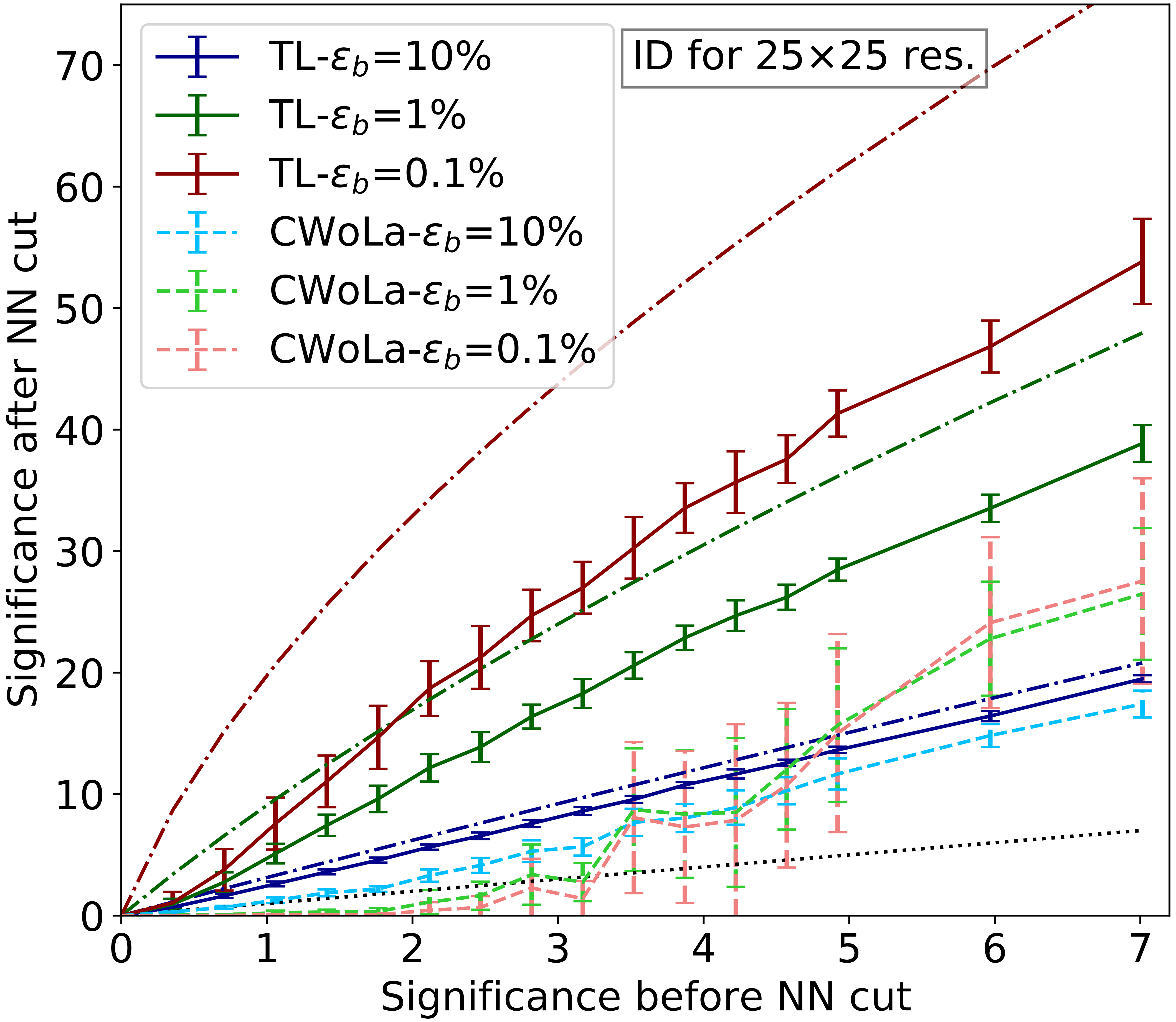
# Transfer Learning by Pre-training and Fine-tuning

- **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 small set of actual data as **target data**
  ➠ freezing $\Theta$ in the convolutional layers and reinitializing and training $\theta$ in the dense layers
  ➠ fixing the feature extraction part while training the classification part

| | | |
|---|---|---|
| Layers of CNN subnetwork | $\left(\begin{array}{l}\text{convolutional 2D layer: 64 filters with } 5\times 5 \text{ kernel size}\\ \text{maxpooling layer: } 2\times 2 \text{ pool size}\end{array}\right)\times 2$ | $\Theta$ |
| | convolutional 2D layer: 128 filters with $3\times 3$ kernel size | |
| | maxpooling layer: $2\times 2$ pool size | |
| | convolutional 2D layer: 128 filters with $3\times 3$ kernel size | |
| | flatten layer | |
| | (dense layer: 128 units) $\times 3$ | $\theta$ |
| | dense layer (output): 1 unit | |

# Transfer Learning vs Regular CWoLa

Beauchesne, Chen, CWC 2024

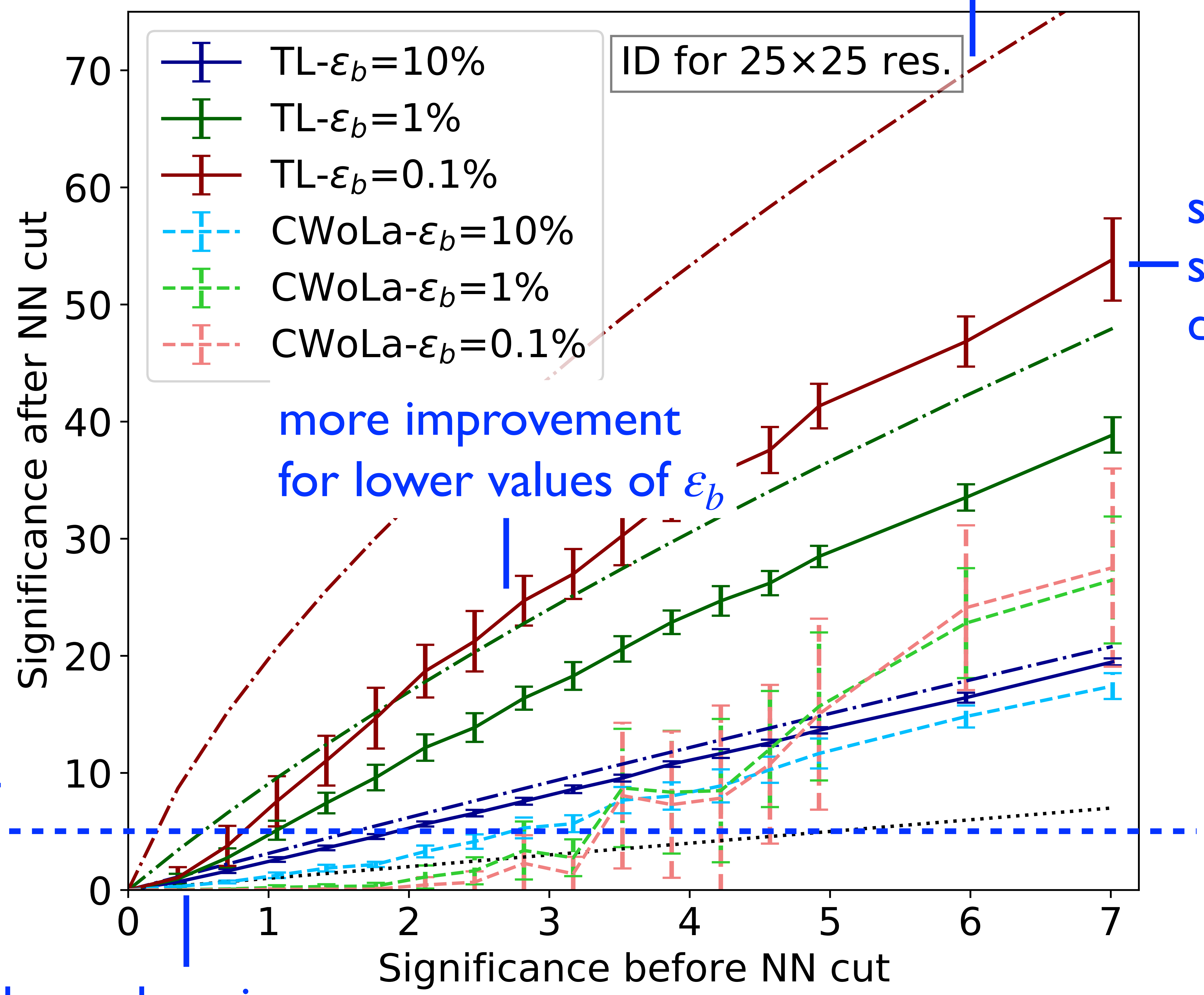



ID; $\Lambda_D = 10$ GeV

# Transfer Learning vs Regular CWoLa

Beauchesne, Chen, CWC 2024

dash-dotted curves
for full supervision




stabler solid curves and smaller fluctuations than dashed curves

more improvement for lower values of $\varepsilon_b$

amount of signal for a 5σ discovery reduced by a factor of a few, due to the fact that NN can better reject backgrounds

lower learning thresholds for TL

ID; $\Lambda_D = 10$ GeV

# Outline

- Introduction
- Full supervision — an example
- Weak supervision — CWoLa
- Dark valley model — a physical model
- Transfer learning
- Data augmentation
- Summary

# 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:
  - $p_{\mathrm{T}}$ **(transverse momentum) 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.

# $p_{\mathrm{T}}$ Smearing and Jet Rotation Methods

- The $p_{\mathrm{T}}$ smearing method is used to simulate **detector resolution/fluctuation** effects on the transverse momentum of jet constituents, achieved by resampling the $p_{\mathrm{T}}$ of jet constituents according to the **normal distribution**:

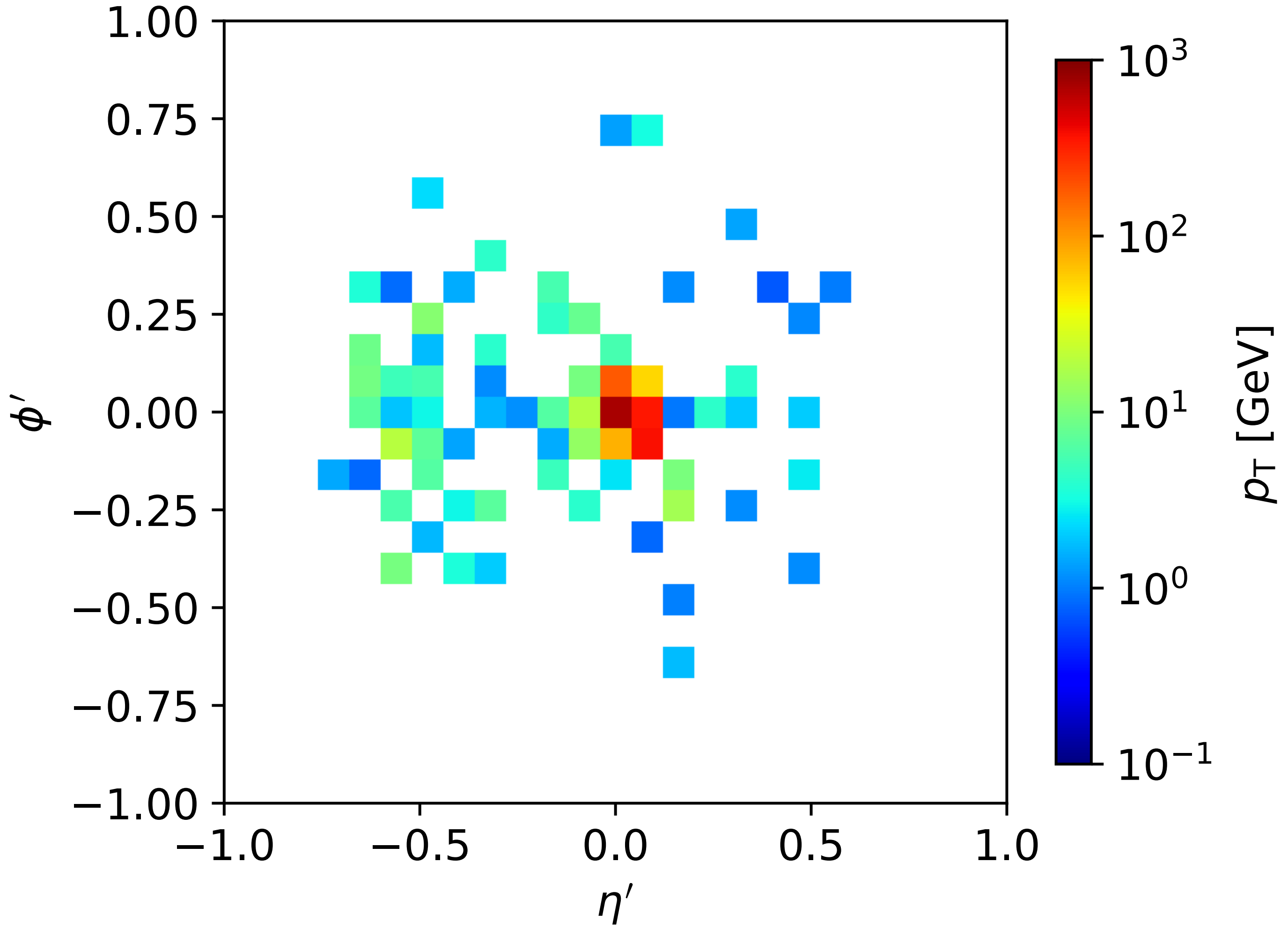
$$p'_{\mathrm{T}} \sim \mathcal{N}\left(p_{\mathrm{T}}, f\left(p_{\mathrm{T}}\right)\right), \quad f\left(p_{\mathrm{T}}\right) = \sqrt{0.052 p_{\mathrm{T}}^2 + 1.502 p_{\mathrm{T}}}$$

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

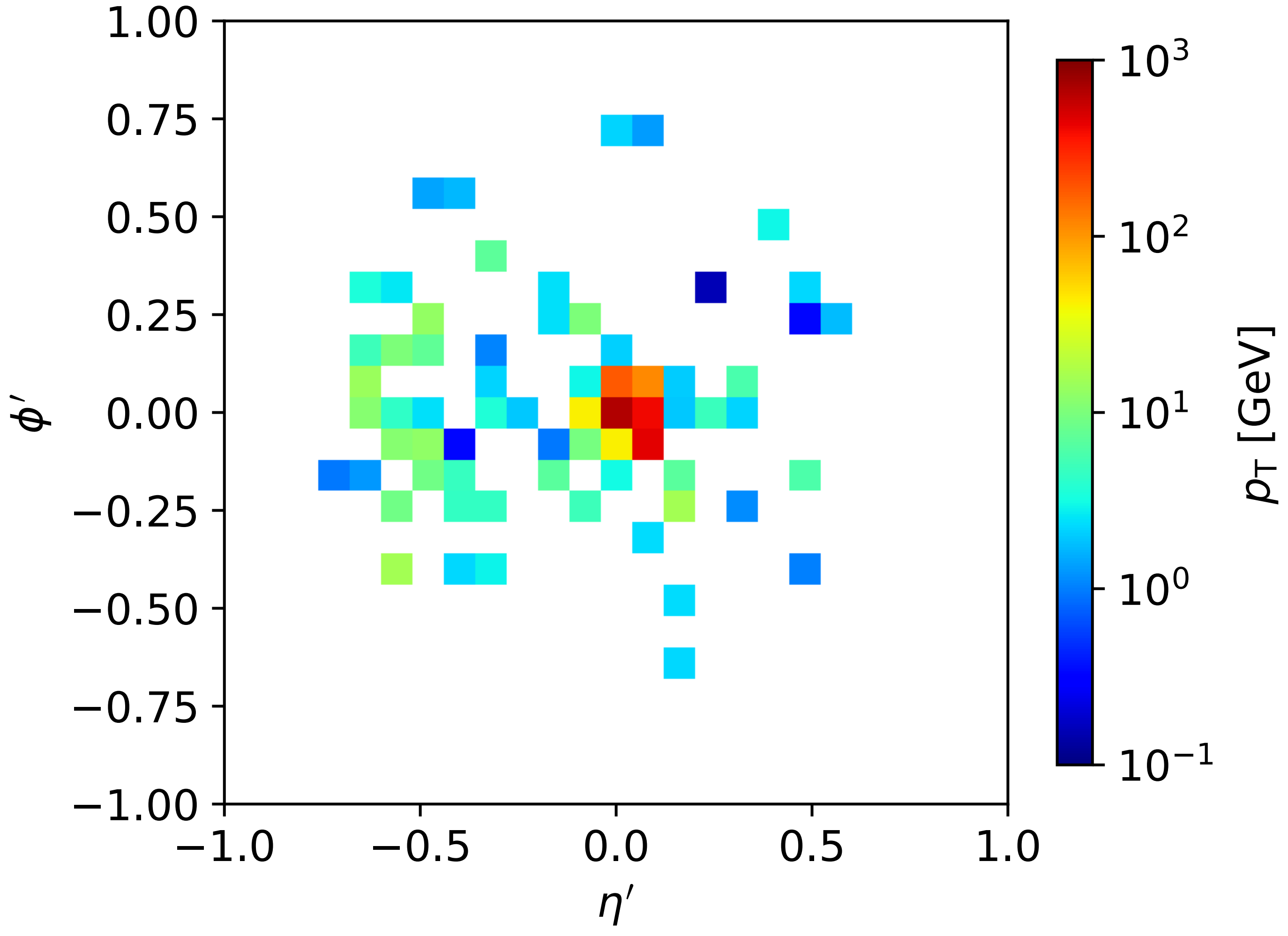
- The jet rotation method rotates each jet with respect to its center by a **random angle** $\theta \in [-\pi, \pi]$ to enlarge the **diversity** of training datasets.

- 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

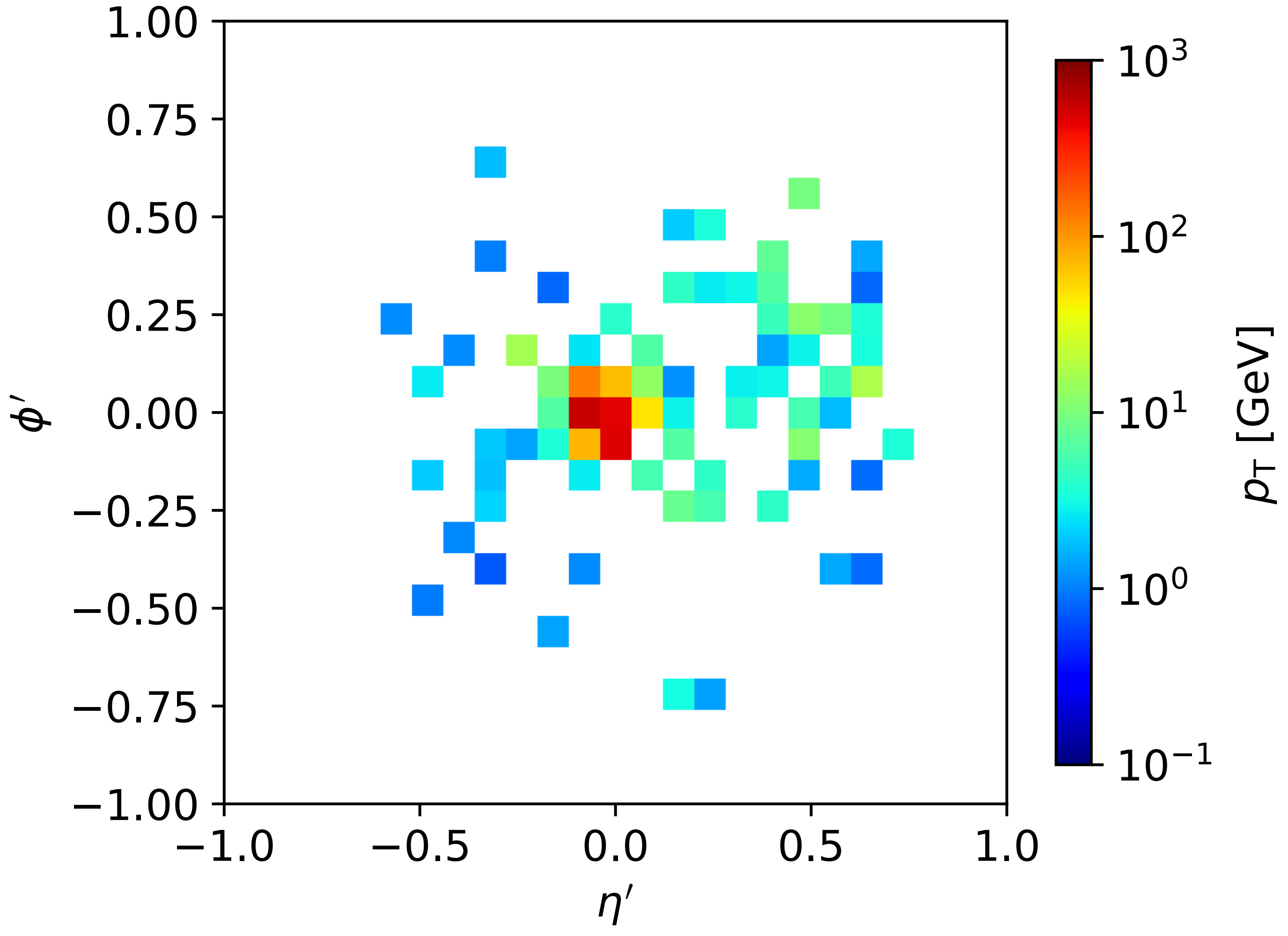
# Example of A Jet Image



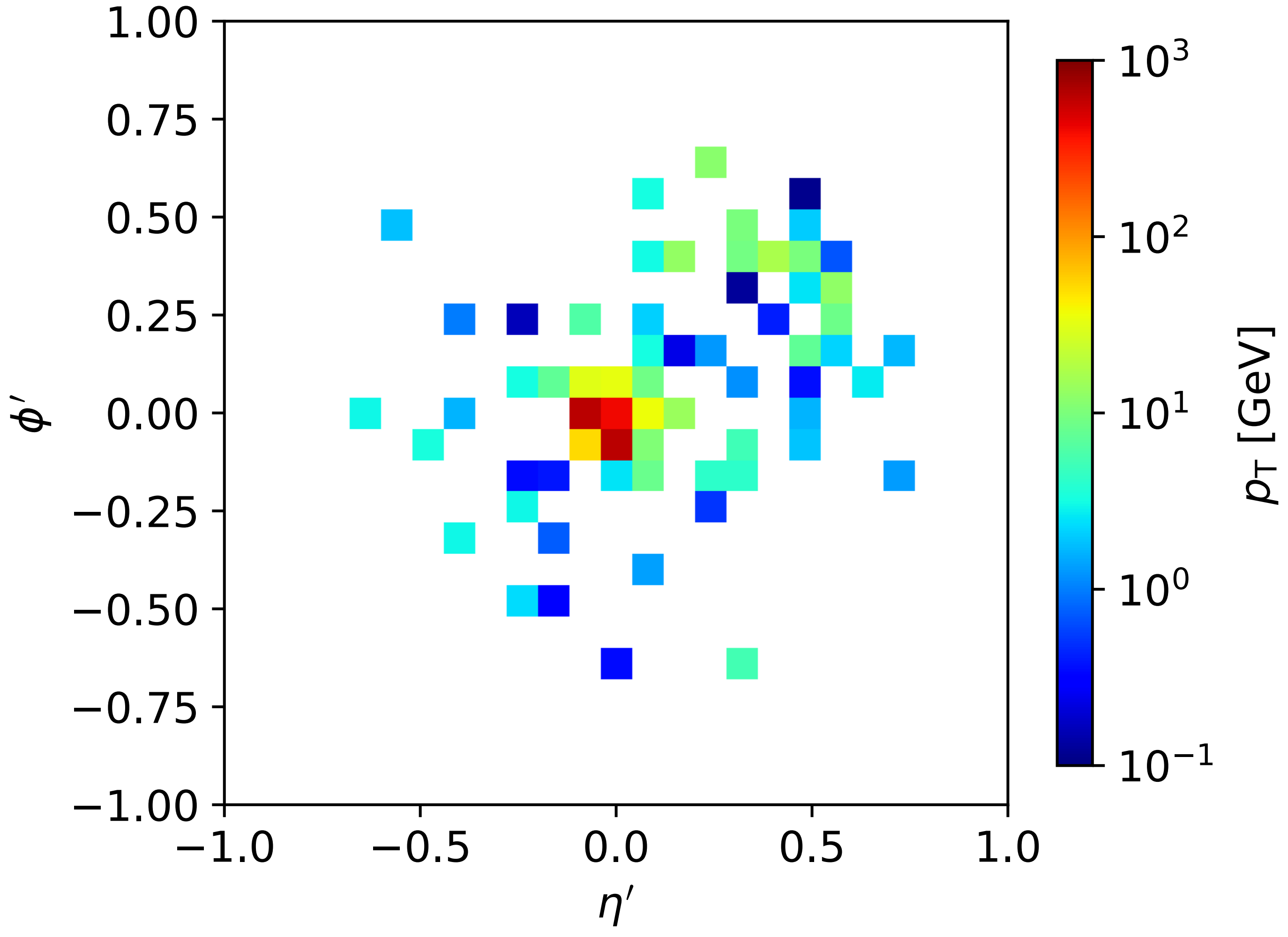

(a) Original jet image




(b) $p_T$ smearing

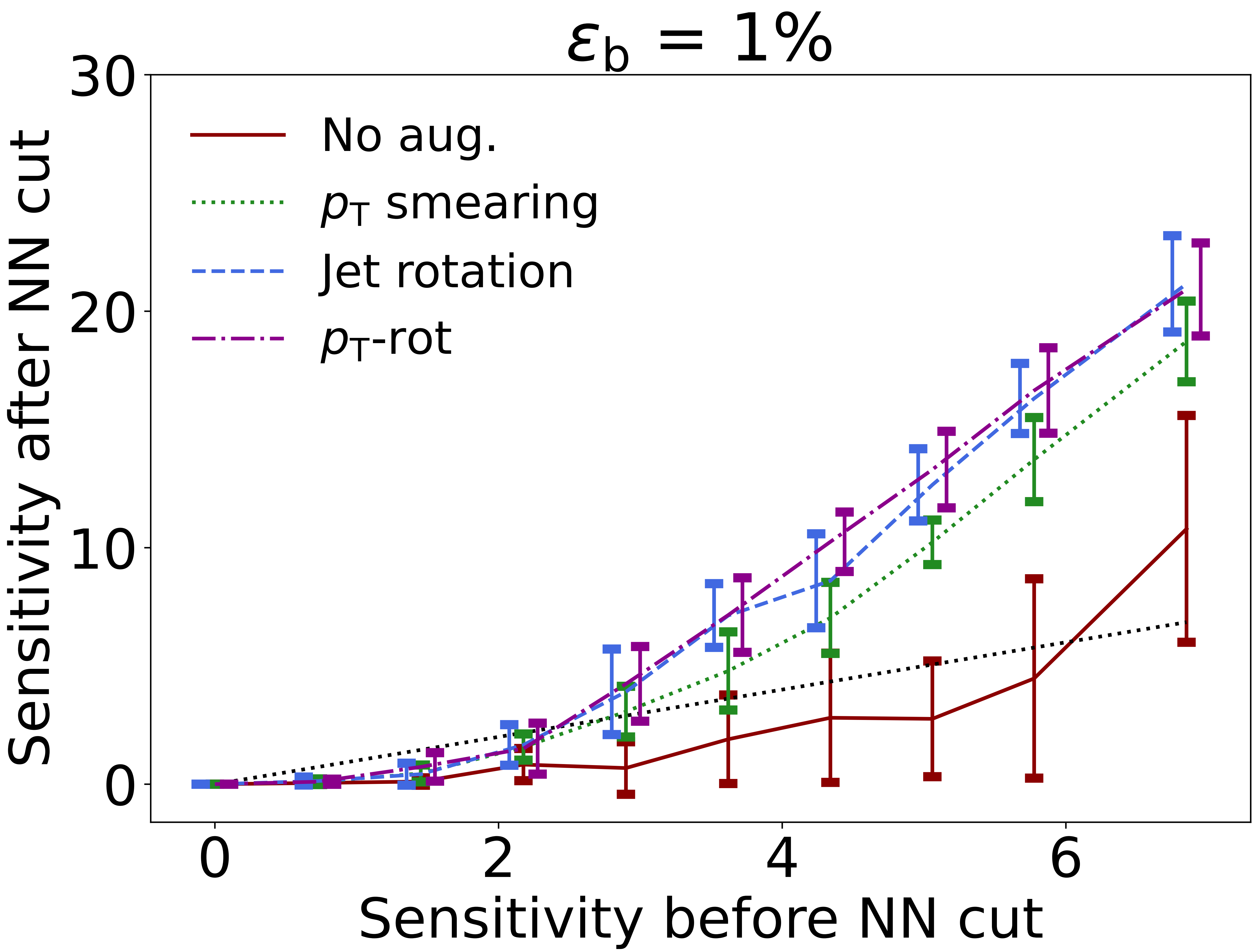



(c) Jet rotation




(d) $p_T$ smearing + jet rotation

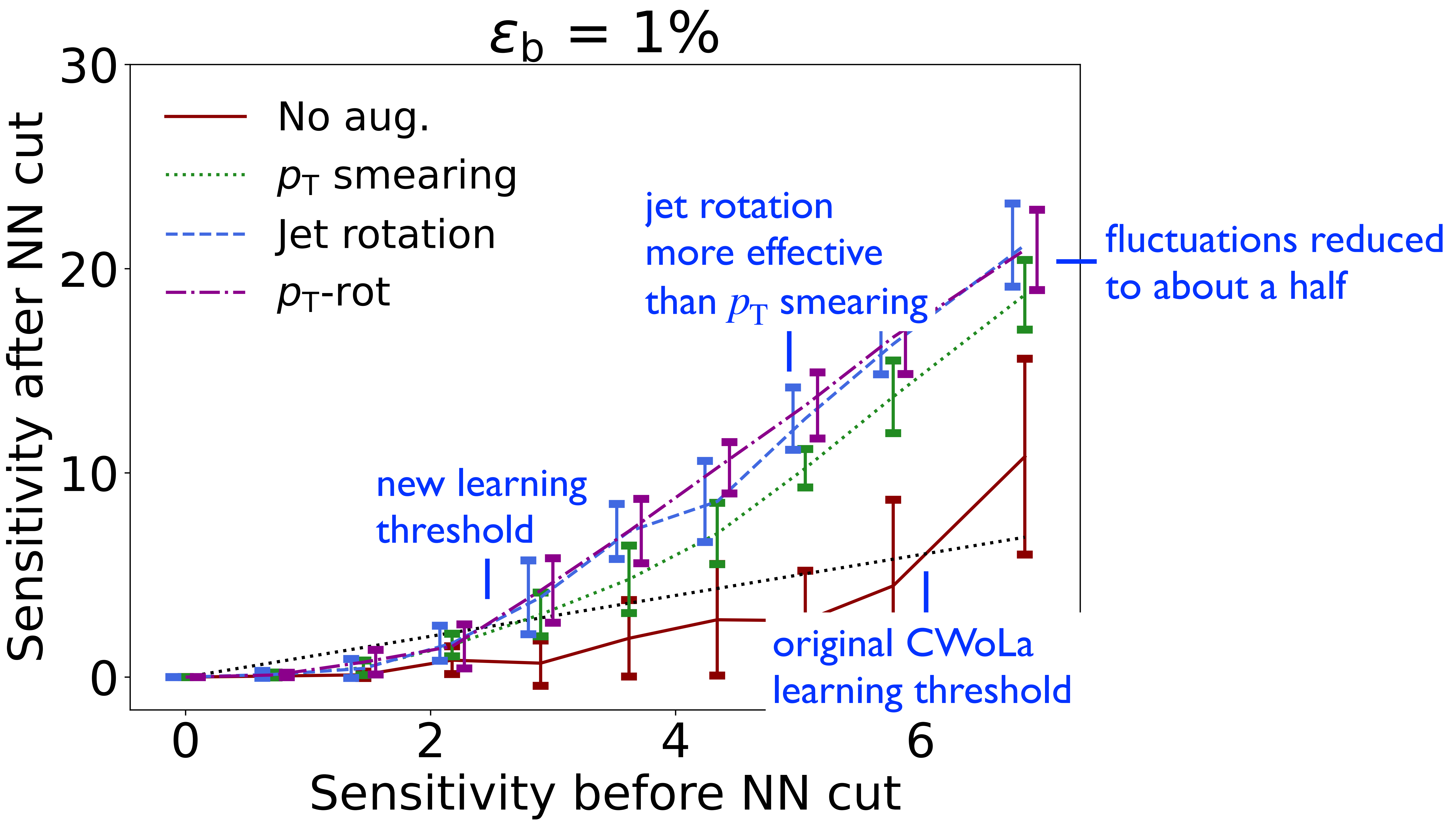
# Sensitivity Improvement




ID; $\Lambda_D = 10$ GeV

Chen, CWC, Hsieh 2024

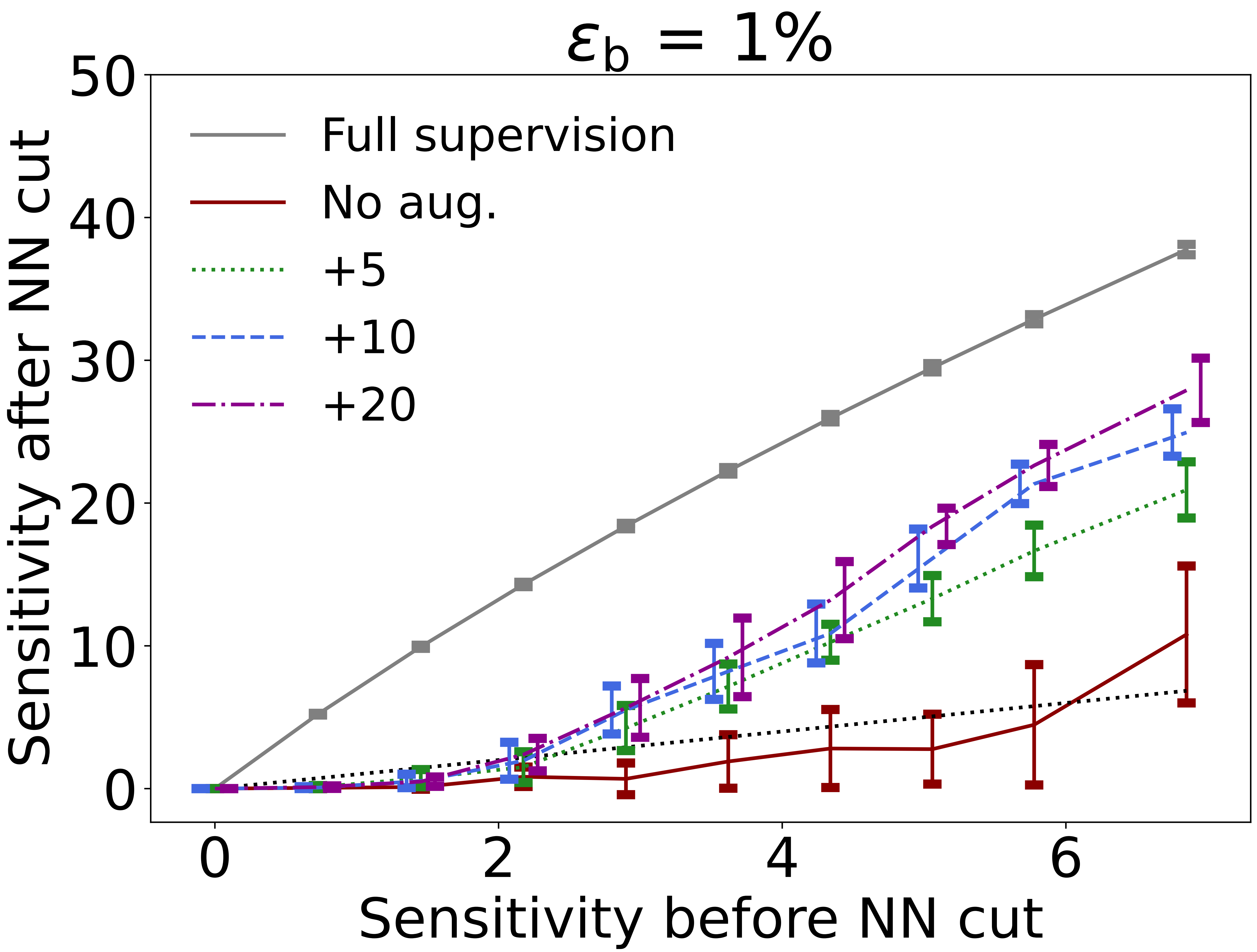
# Sensitivity Improvement




ID; $\Lambda_D = 10$ GeV

Chen, CWC, Hsieh 2024

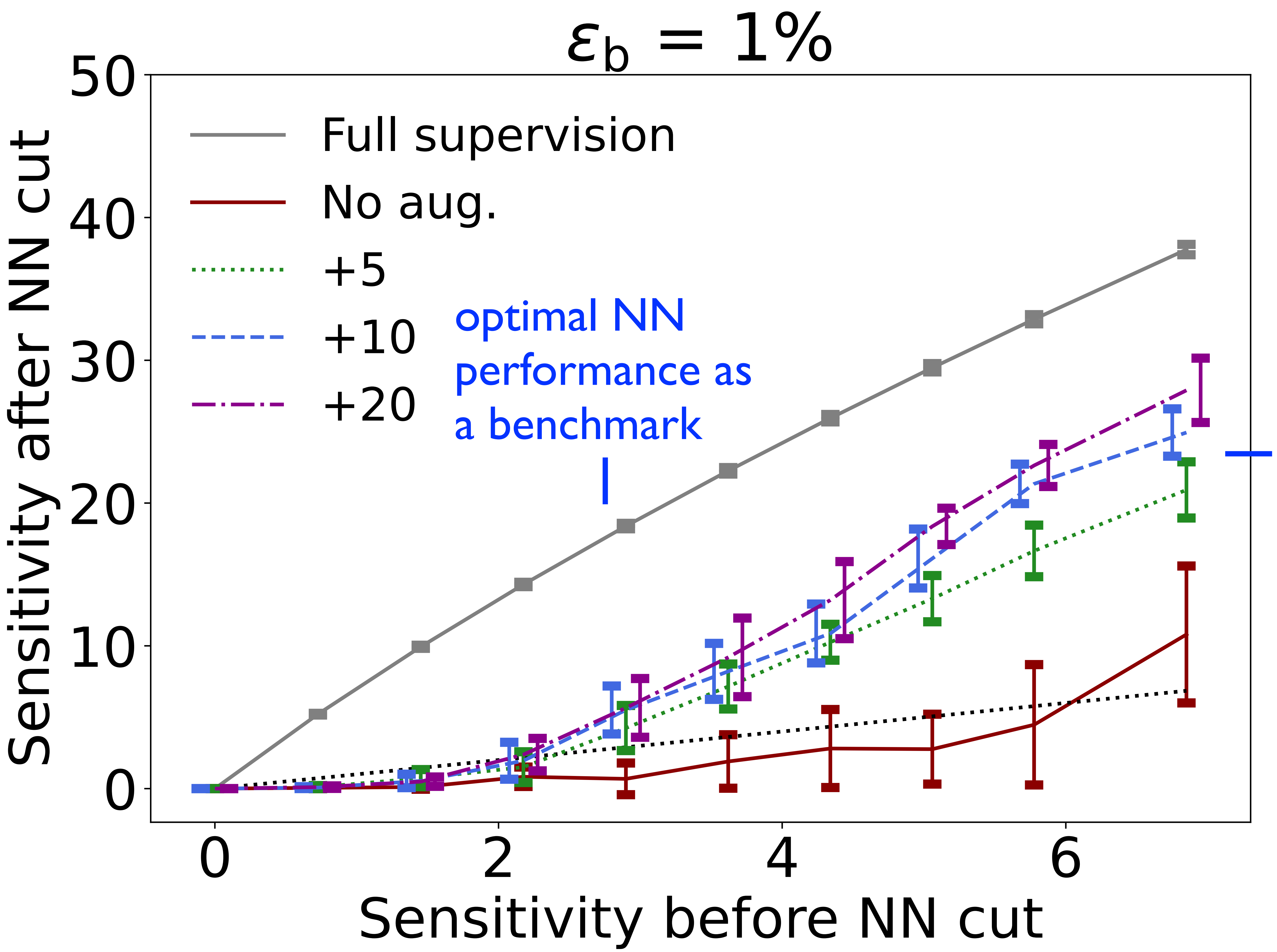
# Dependence on Augmentation Size




ID; $\Lambda_D = 10$ GeV

Chen, CWC, Hsieh 2024

# Dependence on Augmentation Size

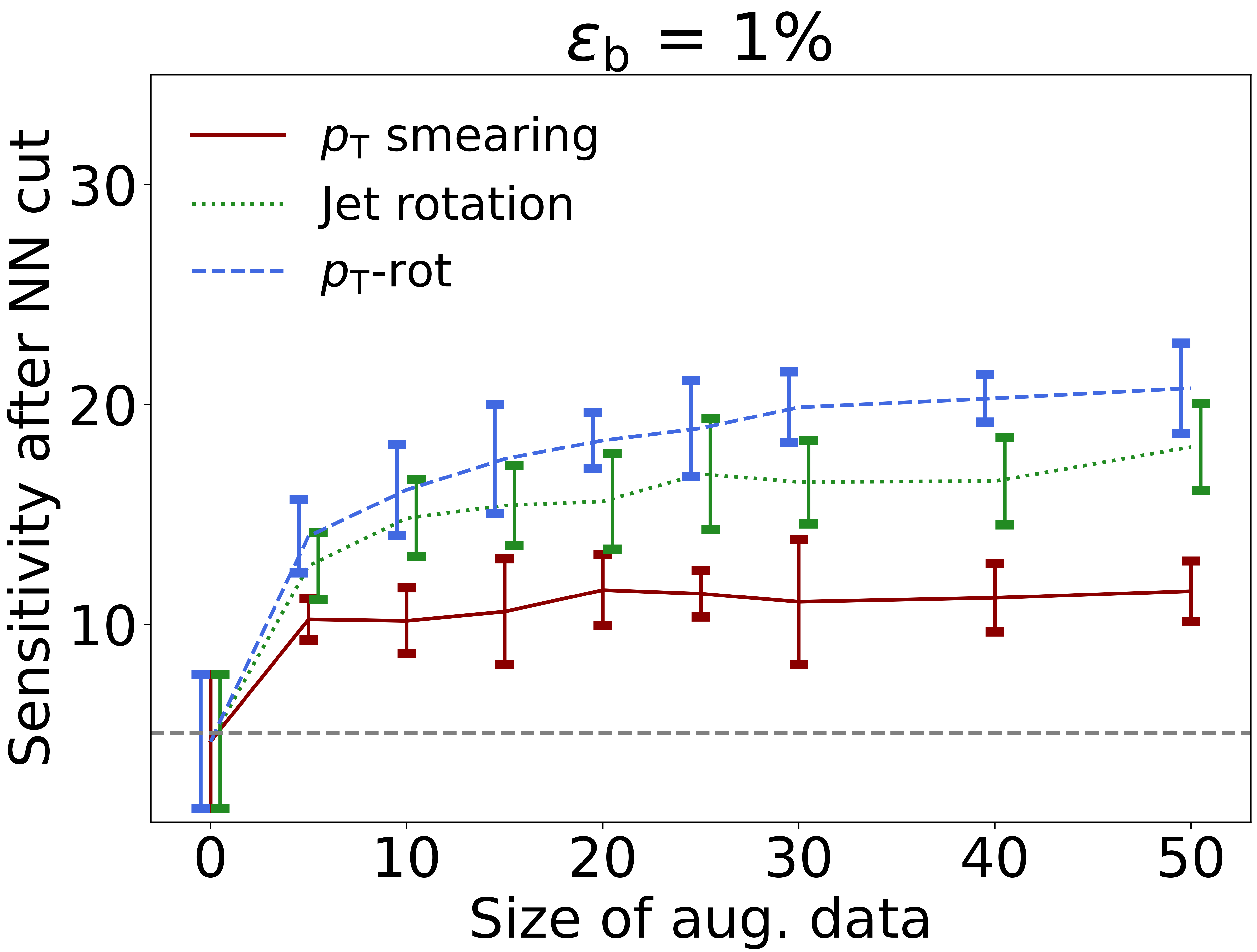



improving with sample size, but not linearly

ID; $\Lambda_D = 10$ GeV

Chen, CWC, Hsieh 2024

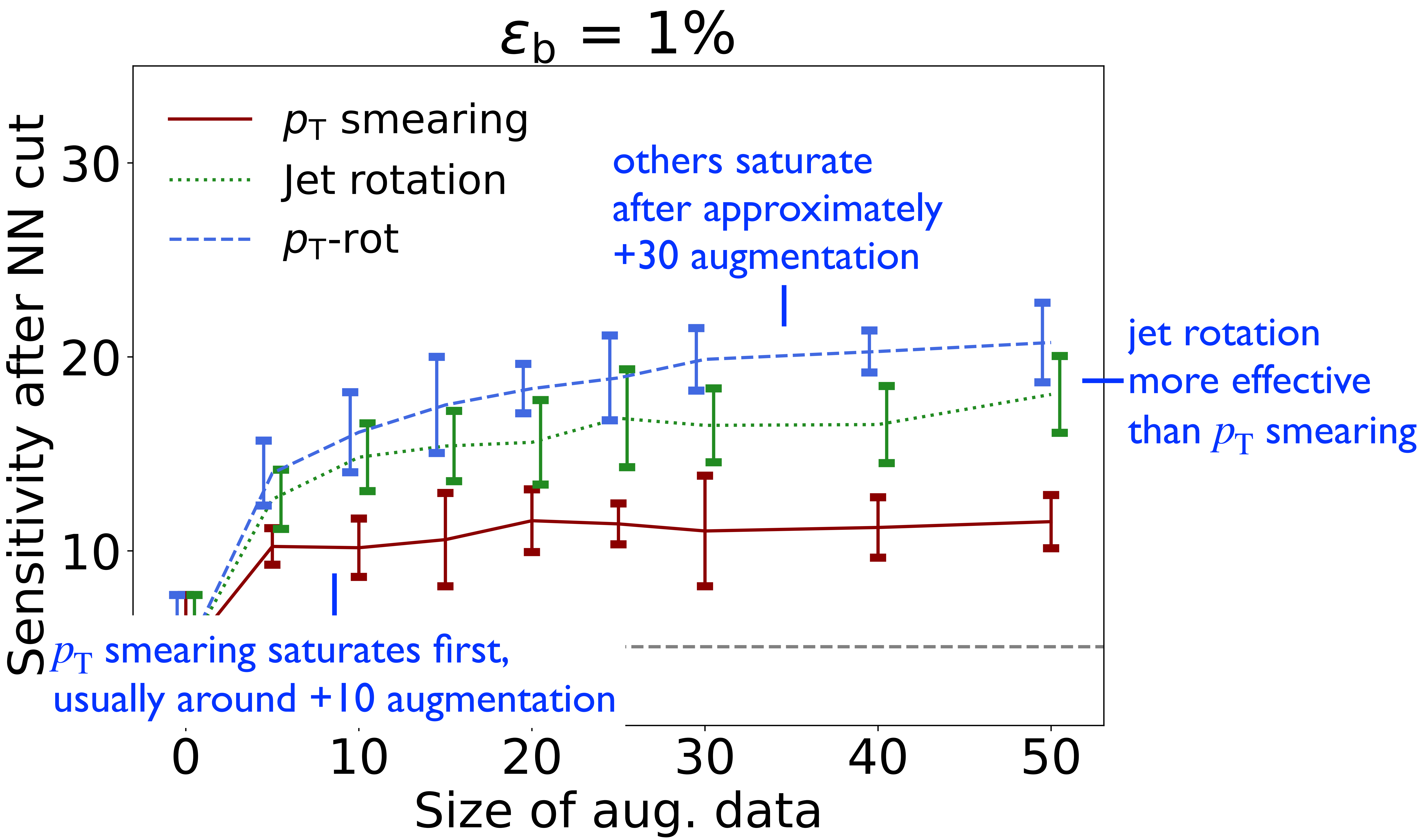
# Asymptotic Behavior of Augmentation Size




ID; $\Lambda_D = 10$ GeV

Chen, CWC, Hsieh 2024

# Asymptotic Behavior of Augmentation Size




ID; $\Lambda_D = 10$ GeV

Chen, CWC, Hsieh 2024

# Summary

- **Weak supervision** (e.g., CWoLa) has 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 using the **data augmentation** technique and show that **jet rotation** is more effective than $p_{\mathrm{T}}$ **smearing**, that a mere **+5 augmentation** can already achieve great results.

# Thank You!