Modeling uncertainties with DCTR method

[arXiv:2411.03023]

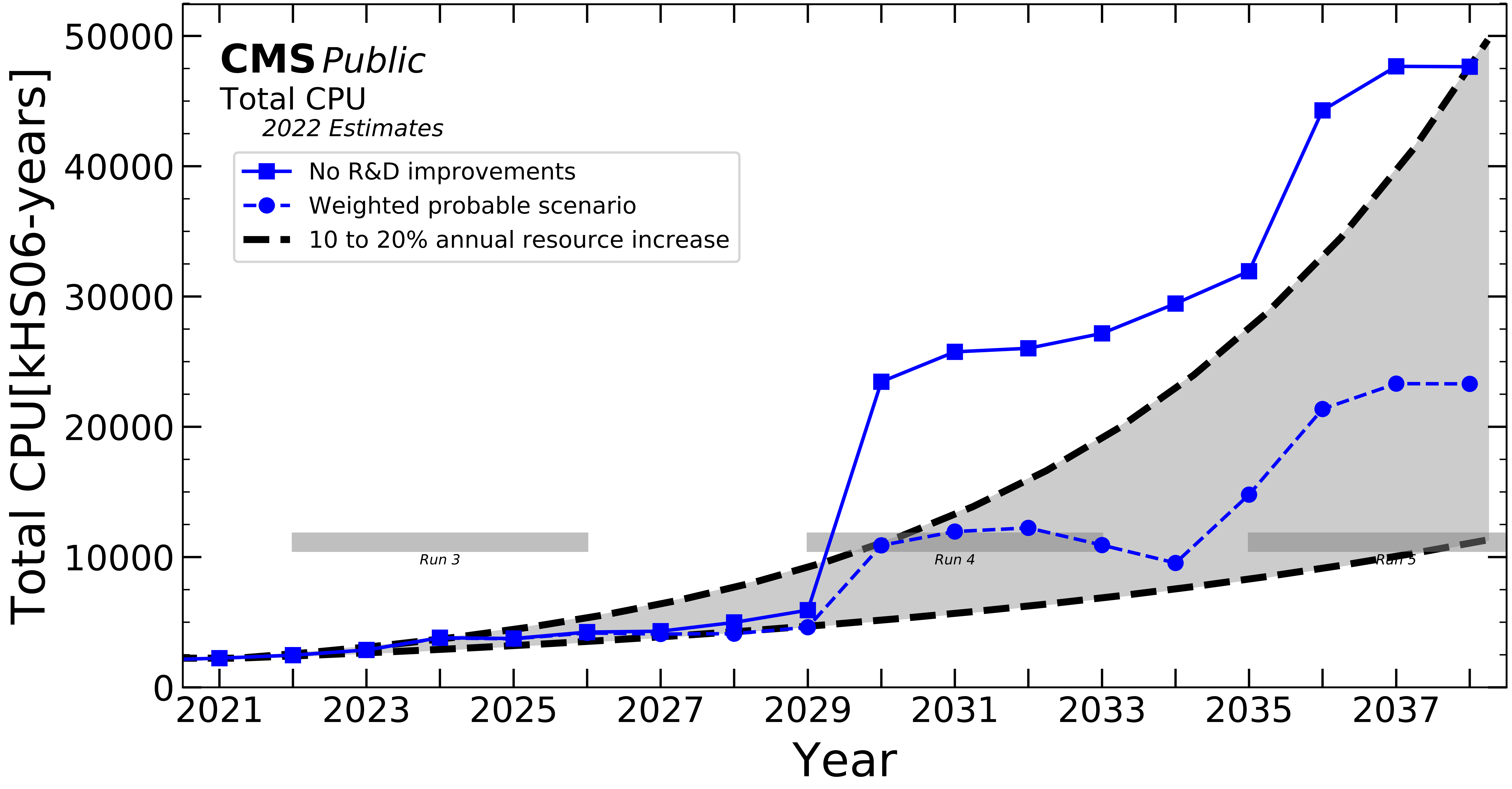
Valentina Guglielmi, Katerina Lipka, Roman Kogler, Simone Amoroso, Finn Puschman

Top workshop, Hamburg, 29.1.2025

HELMHOLTZ

Projections for the CMS Computing needs for HL-LHC

CMS-NOTE-2022-008: CMS Phase-2 Computing Model



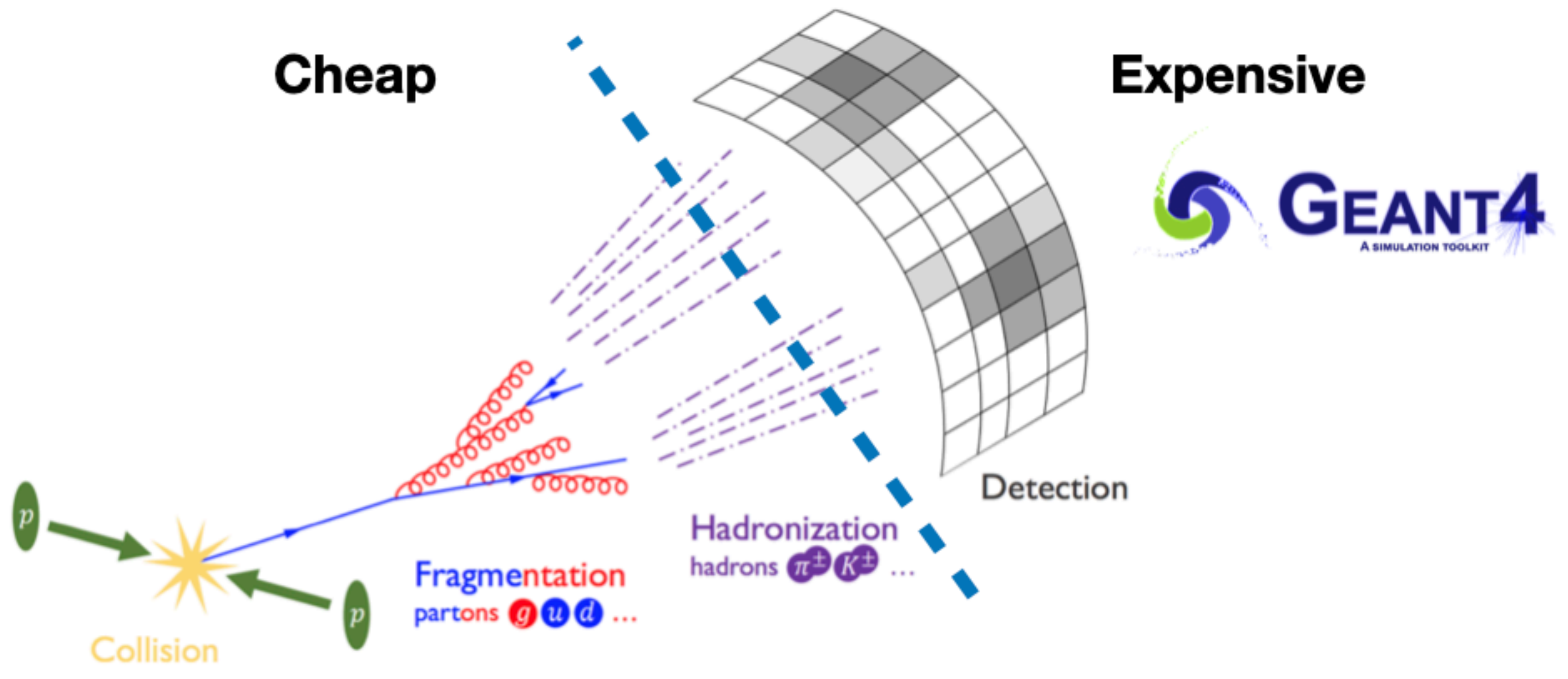
High-Luminosity LHC computing demands will be challenging even in optimistic scenarios

Monte Carlo simulations

Monte Carlo (MC) samples used to compare data to theory predictions

Workflow process:

- Generation of the physics event at NLO → Relatively cheap
- Simulation of the detector → Expensive



~10 different MC samples to take into account **systematic uncertainties**

- High computational cost
- Enlarge MC statistics (samples dedicated to systematic variations often produced with fewer events)

→ **Reweighting:** incorporate all the relevant variations in a single sample (avoid the need to simulate the detector response multiple times)

MC modelling uncertainties

Example: CMS $t\bar{t}$ systematics

Systematic	CMS
Nominal	PowhegPythia8
PDFs	PDF4LHC recommendations
NLO matching	Reweights top pT to NNLO
Initial State Radiation	7-point variations of μ_R^{ME} & μ_F^{ME} + indep vars of hdamp & $\mu_R^{PS,\,ISR}$
Final State Radiation	Variations of $\mu_R^{PS,\,FSR}$
B-fragmentation	Variations of r_B parameter in Pythia8
Hdamp	2-point variations hdamp
Top mass and width	6-point variation each
Underlying Event	Tune variations (CP5) + different CR models
Hadronization	Pythia6 vs Herwig++

← **Reweightable inside MC generator itself**

← **Not reweightable inside MC generator itself**

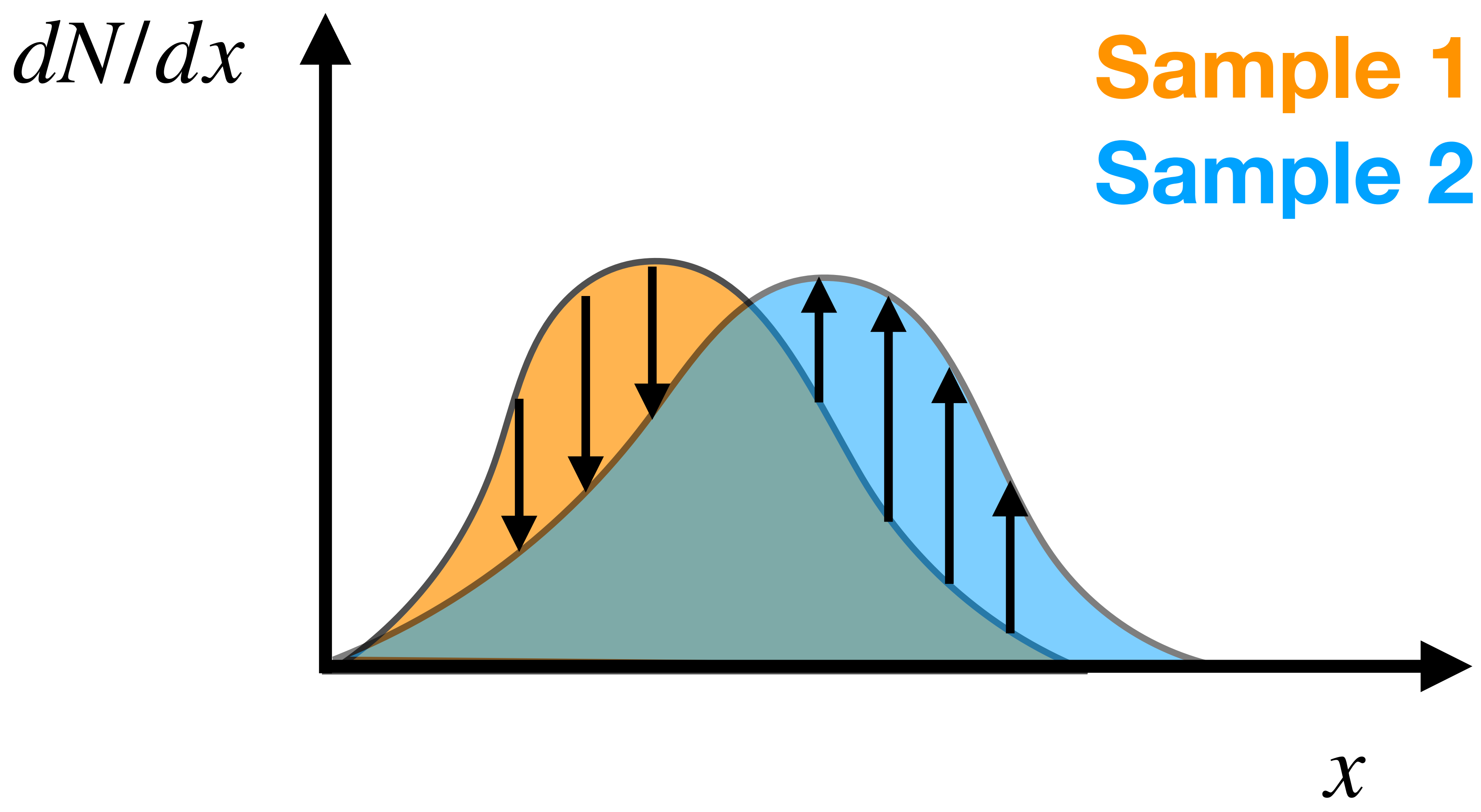
Reweighting prescription

Reweight the nominal MC sample to its variations using event weigths

Consider two MC samples, described by probability densities $p_0(x)$, $p_1(x)$ for $x \in \Omega$ (phase space):

- **Ideal event-level weight:** $w(x) = p_0(x)/p_1(x)$

$$dN/dx \rightarrow w(x) \times dN/dx$$



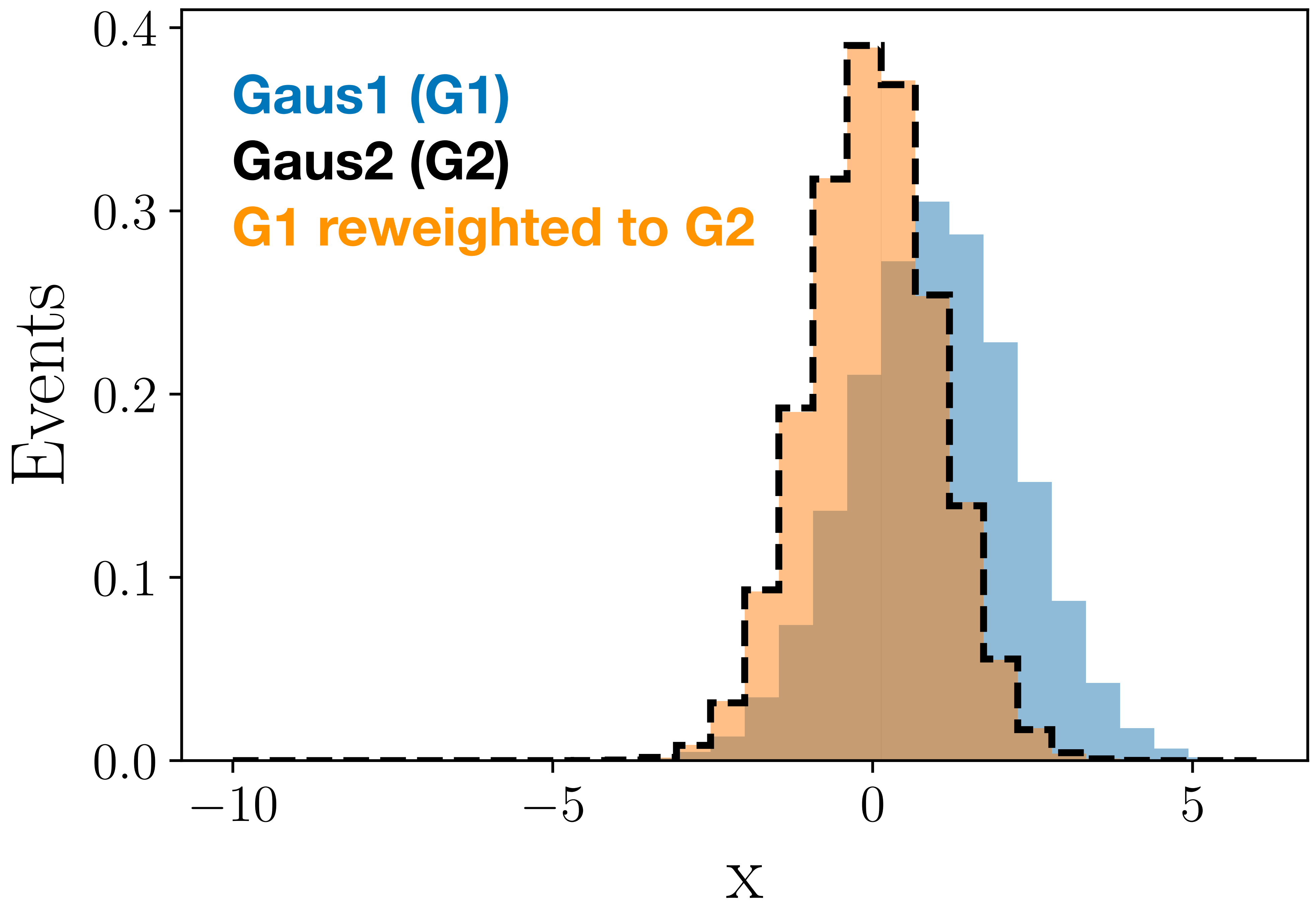
Standard reweighting → Ratio in bins of two distributions

- Sensitive to the binning chosen
- Going beyond a small number of input dimensions is difficult

Machine Learning for reweighting

Neural network learns to approximate the likelihood ratio $w = p_0(x)/p_1(x)$ (arXiv:1506.02169)

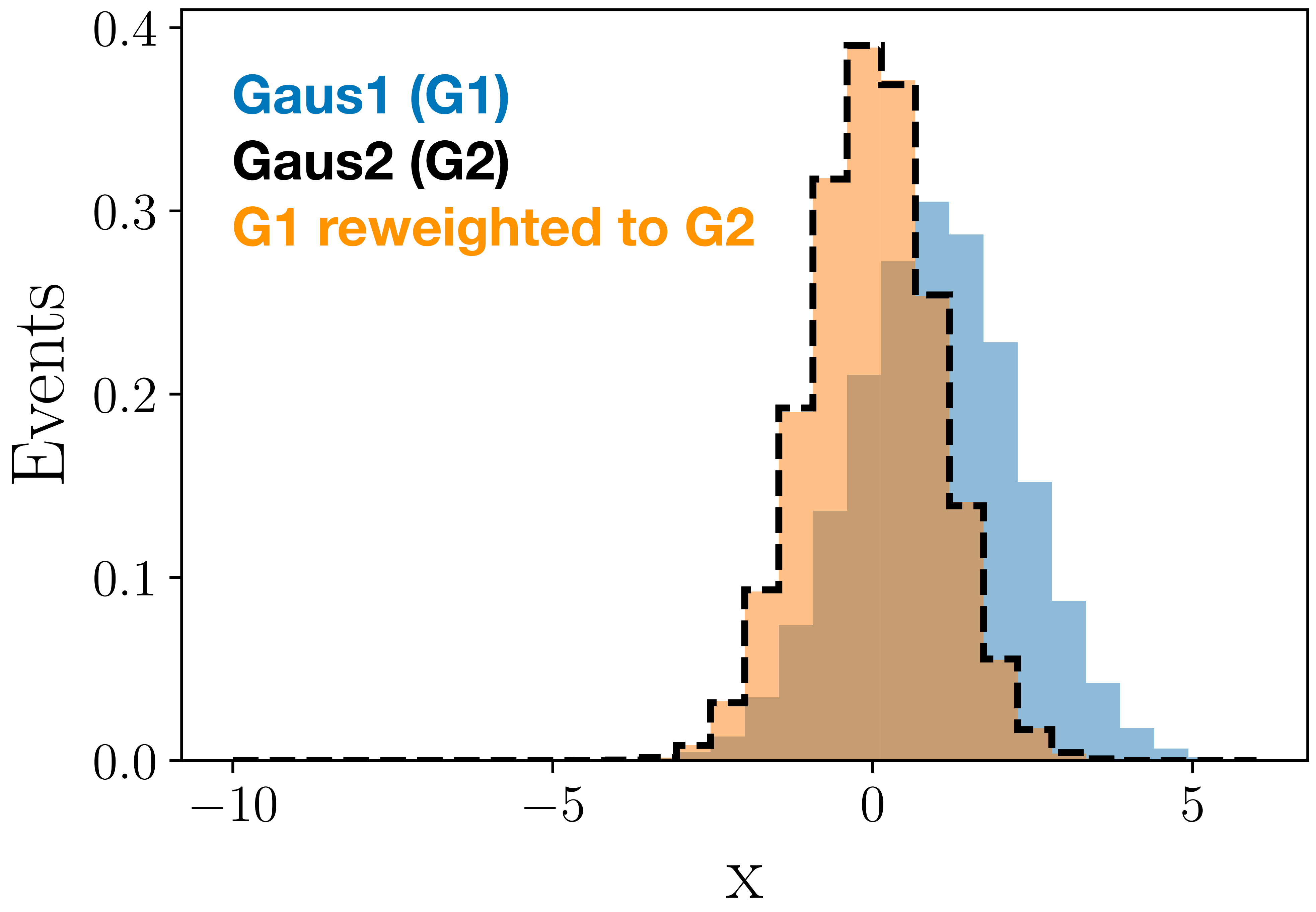
- Naturally takes **multidimensional** and **unbinned** inputs
- **Continuous** as a function of MC parameters



Machine Learning for reweighting

Neural network learns to approximate the likelihood ratio $w = p_0(x)/p_1(x)$ (arXiv:1506.02169)

- Naturally takes **multidimensional** and **unbinned** inputs
- **Continuous** as a function of MC parameters



- *Boosted Decision Tree:* JPC (2016) 762

- Neural Network: arXiv:1506.02169, PRD 101 (2020) 091901, PRD 105 (2022) 076015

- *Input convex neural networks:* arXiv:1609.07152

- *Normalising flow:* Commun. Pure Appl. Math. 66 (2013) 145, Comm. Math. Sci. 8 (2010) 217

The Method: DCTR

Deep neural network using Classification for Tuning and Reweighting

- Developed by A. Andreassen and B. Nachman (PRD 101 (2020) 091901)

Why DCTR?

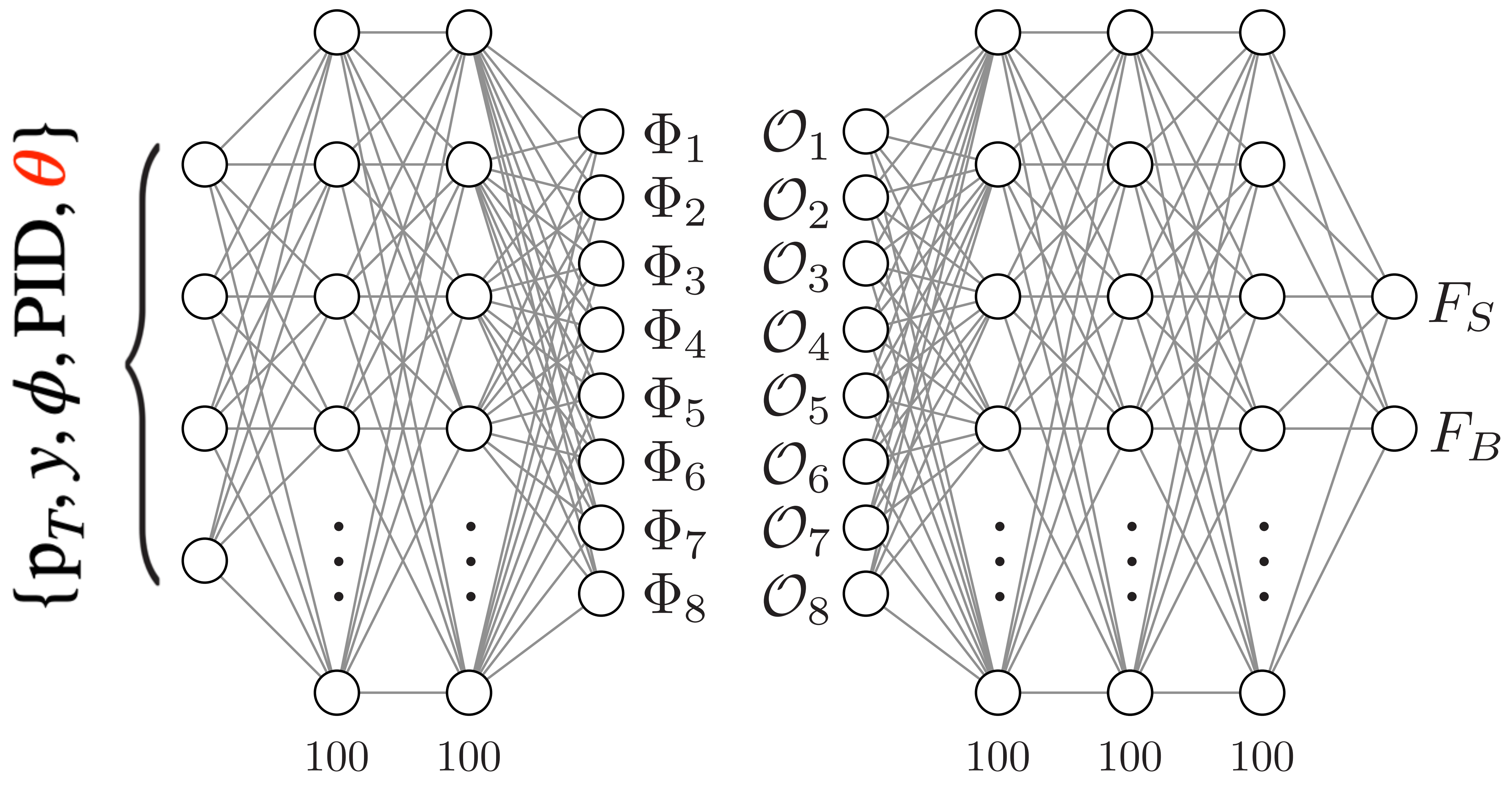
- Particle 4-vector and PID as inputs

  → **Full phase space reweighting**

- NN parametrised with reweighting parameter θ

  → **Continuous reweighting possible**

Particle Flow Network (PFN) (JHEP 01 (2019) 121)



Outlook

We used DCTR method to reweight MC samples of top quark production in CMS

- **Reweighting of MC parameters** → Systematic variations
  - h_{damp} parameter at parton level in POWHEG HVQ
  - B quark fragmentation at particle level in PYTHIA
- **Reweight MC to higher-accuracy theory predictions** → Model reweighting
  - NLO POWHEG HVQ → NNLO MiNNLO

The method can be used in CMS software framework (CMSSW)

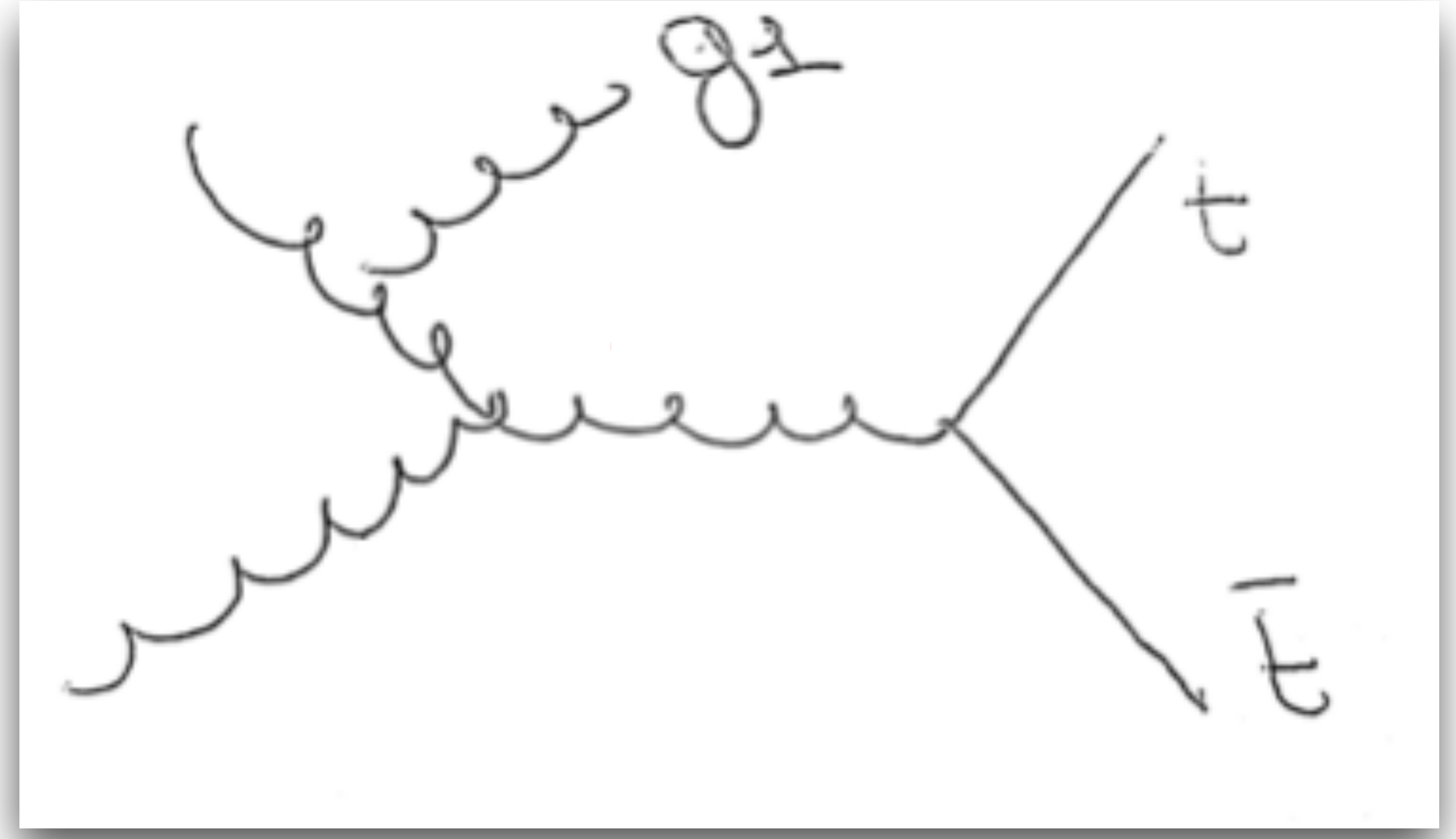
- Every analysis involving top quarks can use the already trained models
- The method can be generalised to any physics case after a dedicated retraining

- All the results and implementation in the CMSSW can be found in [arXiv:2411.03023]

Powheg h_{damp} parameter in top pair production

- Important parameter in nominal $t\bar{t}$ MC sample
- **Damping parameter**, regulating 1^{st} high-pt emission of POWHEG hvq generator
- **Variations of h_{damp}** considered by CMS/ATLAS to assess **ME-PS matching uncertainty**
- In CMS, samples dedicated to h_{damp} variations generated with **less than half** events of nominal sample

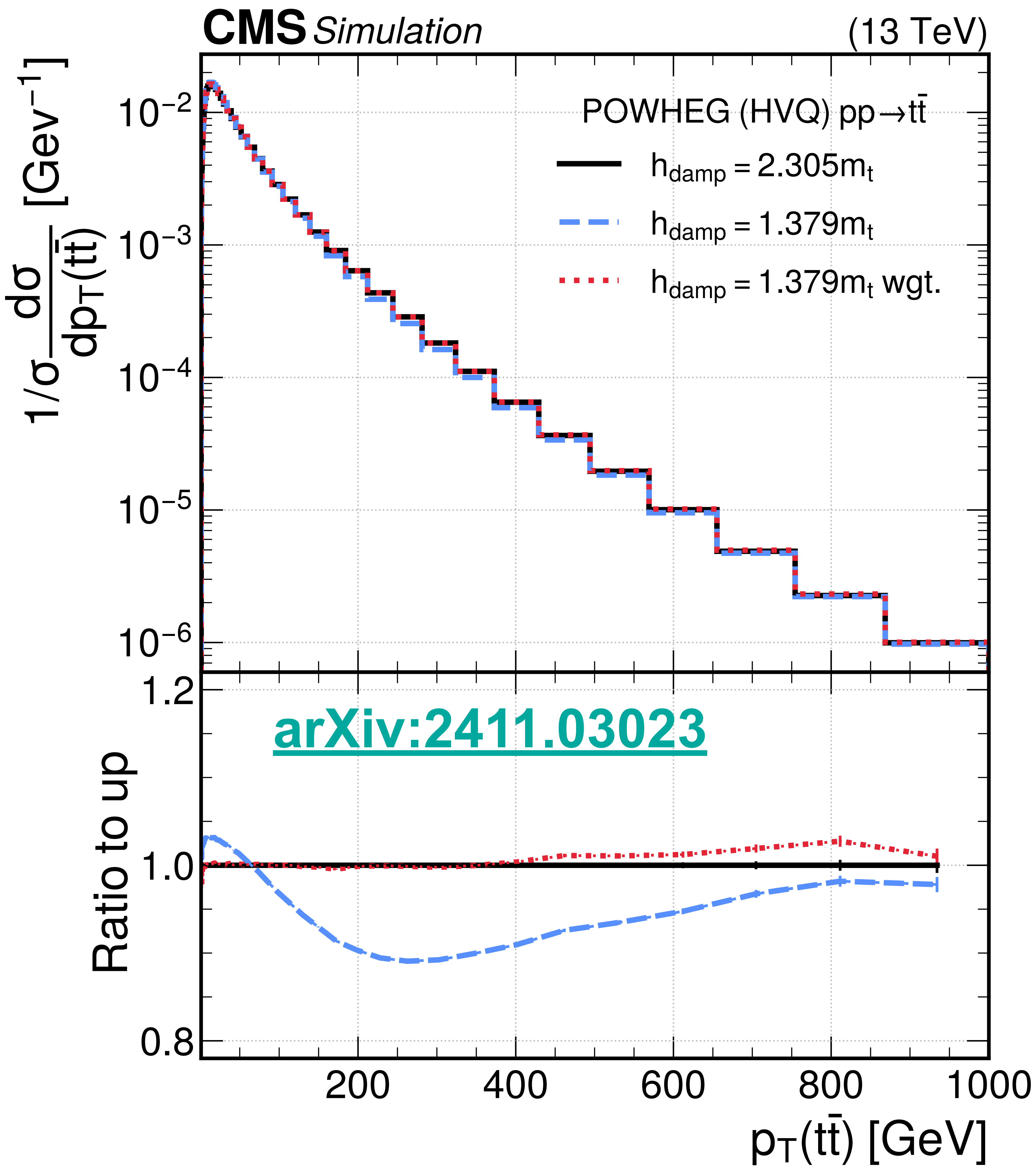
$$\boxed{F = \frac{h_{damp}^2}{p_T^2 + h_{damp}^2}, h_{damp} = h * m_t}$$



Ideal example of systematic variation reweighting

h_{damp} reweighting results

- 2 NN models to reweight CMS nominal sample to the two CMS variations of h_{damp} ($h_{damp} = 1.379m_t \rightarrow h_{damp}^{up} = 2.305m_t$)
- **Parton level (LHE) information as input to the PFN:**
  - 4-vector (p_T, y, ϕ, m) and PID [top, antitop]
- **Before reweighting:** ratio between nominal and up variation sample of h_{damp}
- **Method closure within ~2%:** ratio between reweighted sample and the target one



Pythia B-fragmentation parameter in top pair production

B-fragmentation uncertainty: variations of r_b parameter of Lund-Bowler function in PYTHIA8

$$\boxed{f_B(z) \propto \frac{1}{z^{1+br_b m_B^2}}(1-z)^a exp(-bm_B^2/z)}$$

m_t , m_b: top & b quark mass

a, b: terms related to light quarks

r_b: term related to b quark

a, b, r_b free parameters to be tuned to data

In CMS only the sample with PYTHIA nominal $r_b = 0.855$ produced, no variations

$\rightarrow$ Crucial to use a reweighting method to produce the variations

Example of continuous reweighting

B-fragmentation discrete reweighting

- **B-hadron momentum fraction respect to b-quark x_b as input to PFN:** 1D variable comprising entire event information

$$x_b = \frac{2p_B \cdot q}{m_t^2}/(1-w),\ w = m_W^2/m_t^2$$

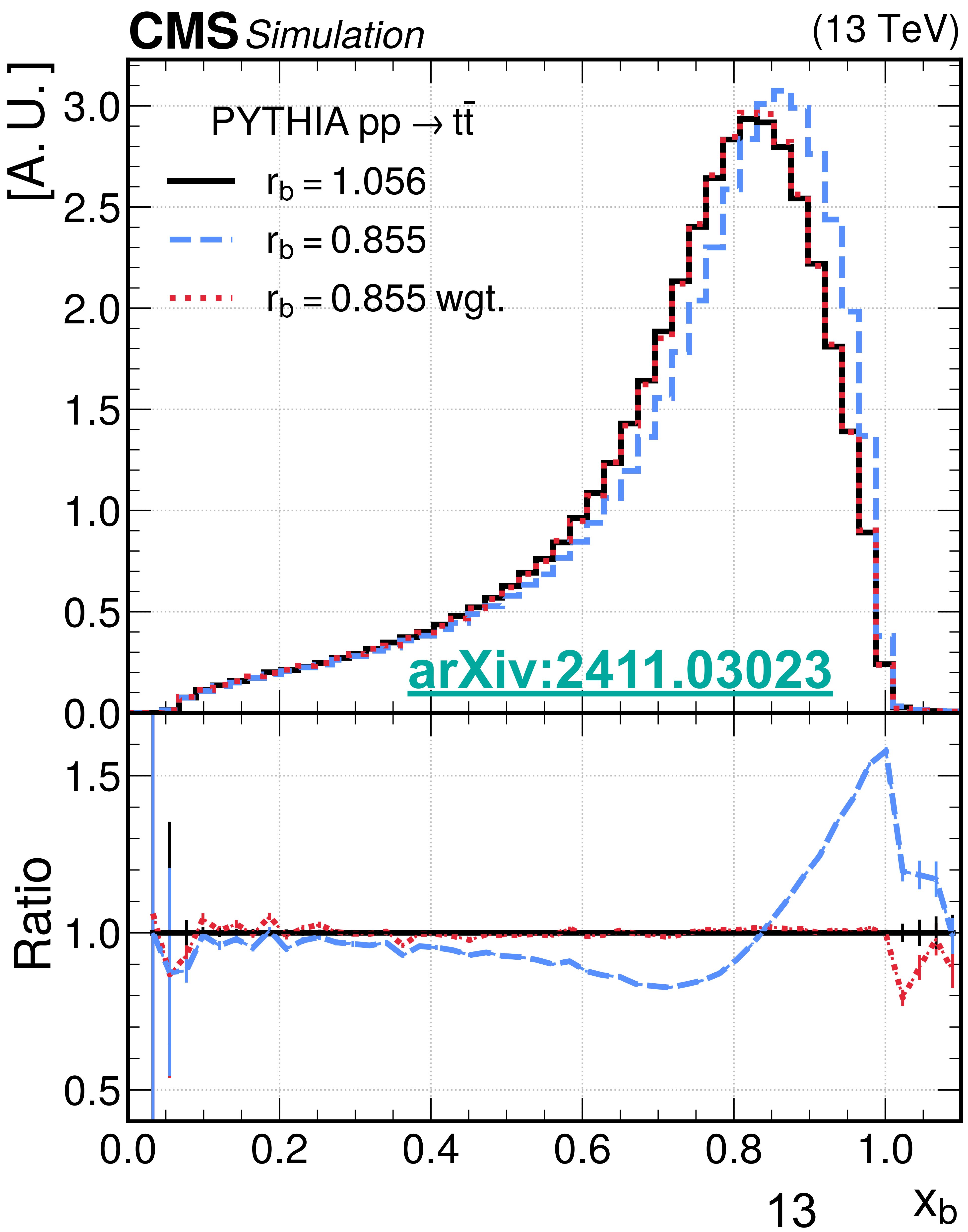
q: four-vector top p_B: four-vector B-hadron

m_t: top mass m_W: W-boson mass

- **2 NN models** to reweight CMS nominal sample to the two CMS variations of r_b

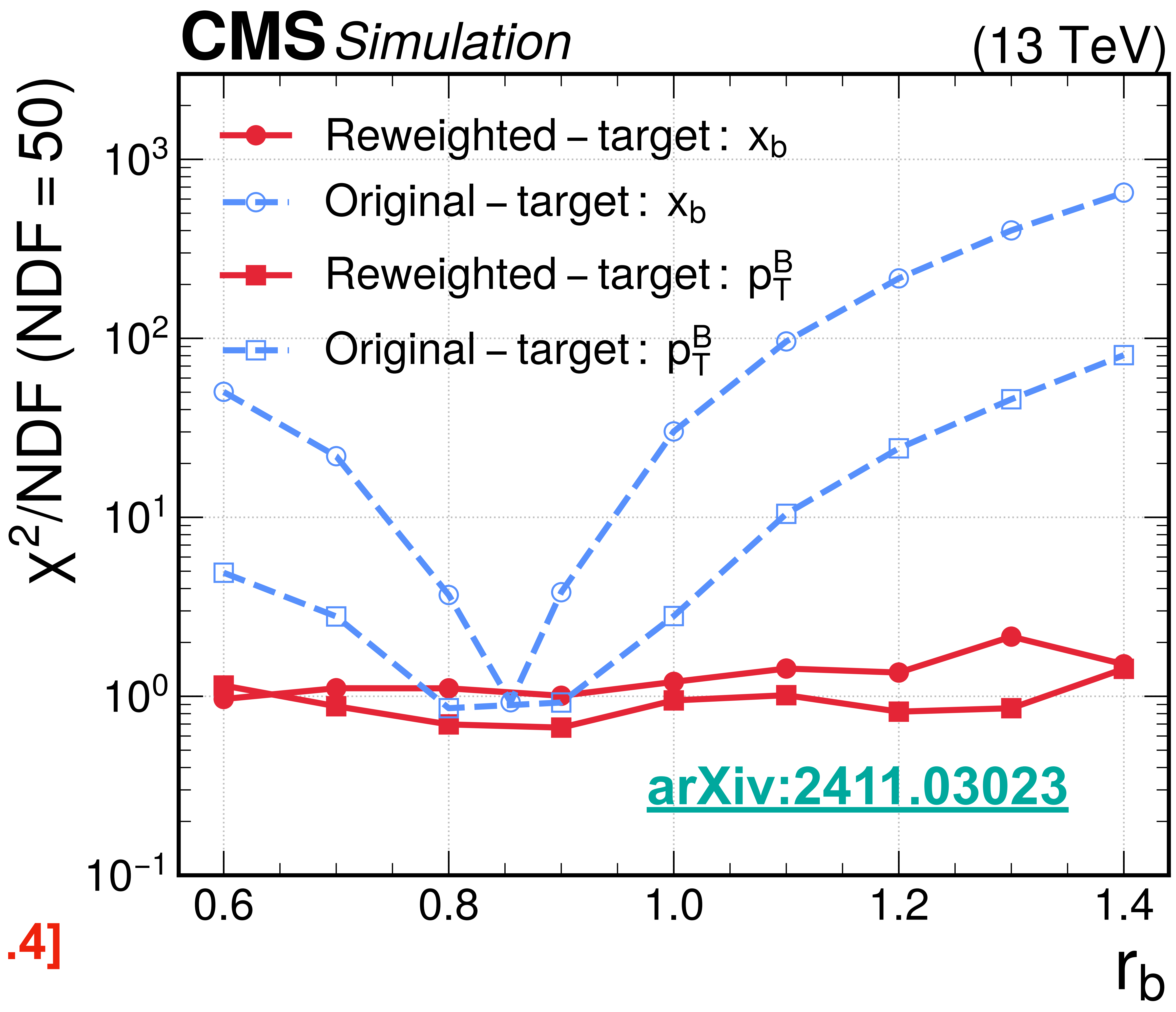
e.g. $r_b = 0.855 \ \rightarrow \ r_b = 1.056$

- **Reweighting closure within 2% up to $x_b < 1$**



B-fragmentation continuous reweighting

- Trained one single NN model to reweight:
  - **Whatever value of r_b in [0.6, 1.4] to** $r_b = 0.855$
  - NN parametrised in θ (i.e. r_b)



- **The method works well in all the range r_b = [0.6, 1.4]**

Model reweighting

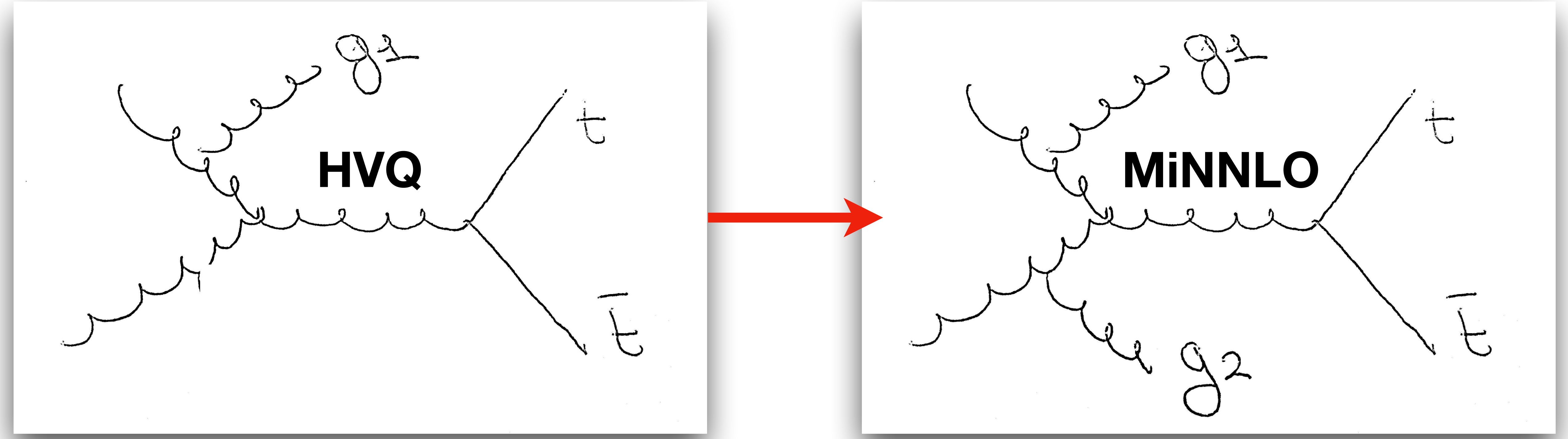
Generator/Predictions increasingly accurate and available (e.g. NNLOPS: MiNNLO$_{PS}$)

- But difficult (and slow) to regenerate and resimulate all the MC samples

Temporary solution:

→ Reweighting of Parton Level MC Simulations to higher-accuracy theory predictions

NLOPS: POWHEG hvq (JHEP 06 (2010) 043) → NNLOPS: MiNNLO (JHEP 05 (2020) 143)

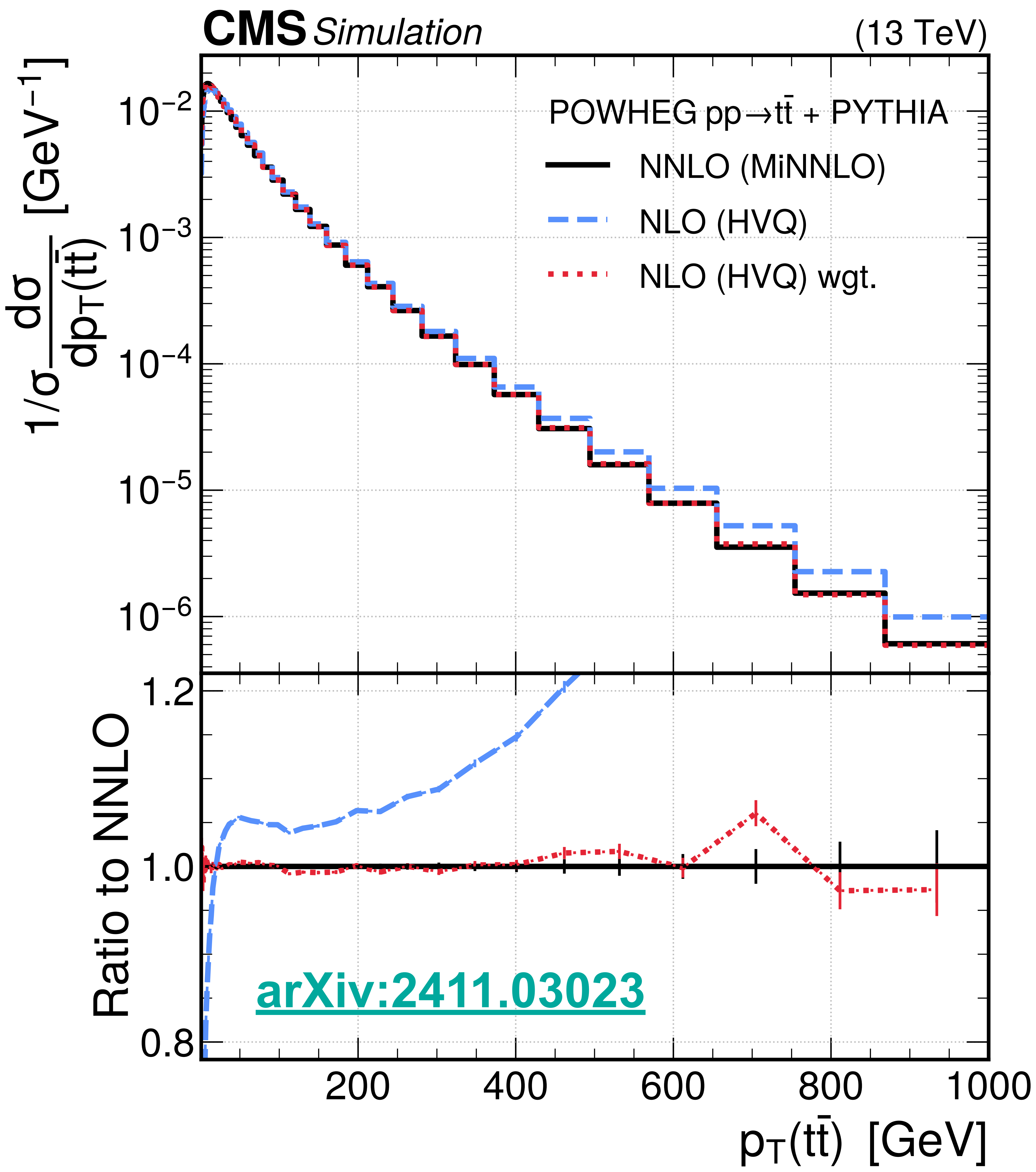


Both interfaced with PYTHIA 8, since the shower effect acts differently on the two generators

Only events based on the kinematics of tt system reweighted, inclusive over additional ME + PS radiations

Model reweighting results

- **Parton level information as inputs to the PFN:**
  - 4-vector (p_T, y, ϕ, m) and PID [t, $\bar{t}$, $t\bar{t}$ system] of the showered events

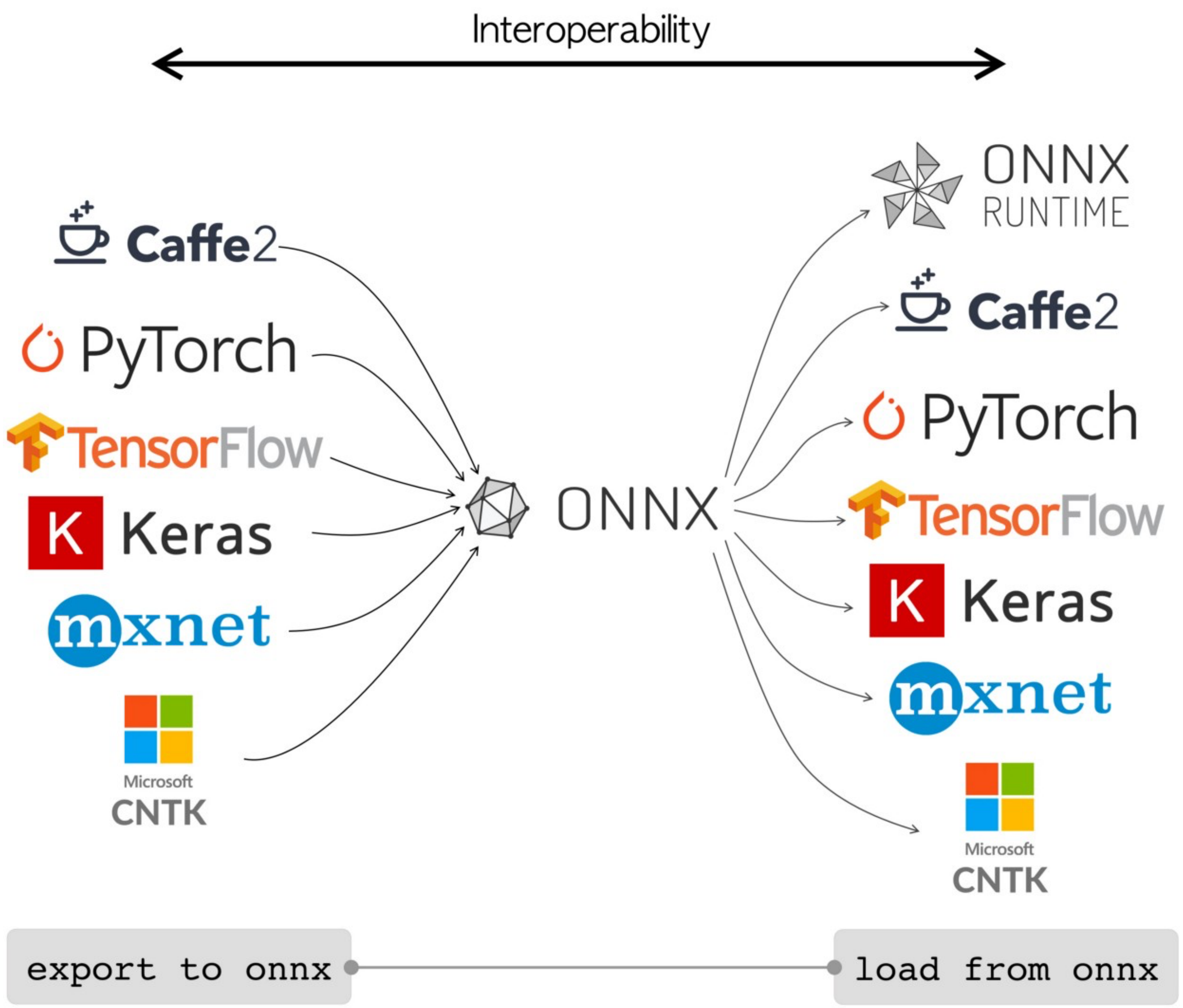


- **Before reweighting:** ratio between NLO and NNLO generators
- **Method closure** within ~2%: ratio between reweighted sample and the target one (NNLO)

Implementation in CMSSW

User doesn't need to retrain the model, it has just to load the model and compute the weigths to apply to its events

- **Trained model saved in ONNX universal format and can be used in CMSSW**
  - Facilitates sharing/usage of NN models across different frameworks



- **Weights can be add at whatever analysis stage**

GEN → SIM → RECO → NTUPLE

- **The method is generic, can be used by all analyses**

Summary and conclusions

- **Modelling uncertainties are already a major source of uncertainty at LHC**
  - Computational cost is a bottleneck (many alternative samples to be produced)
  - The current conditions will not be sustainable at HL-LHC
- **DCTR reweighting of MC samples solves the bottleneck**
  - First use of DCTR for a real CMS analysis application
  - Reweighting achieved high precision
- **Many other applications in any physics field can be investigated**
  - MC tuning at detector level
  - PYTHIA vs Herwig reweighting
  - Unbinned and full phase space unfolding
  - …

Thank you

Backup

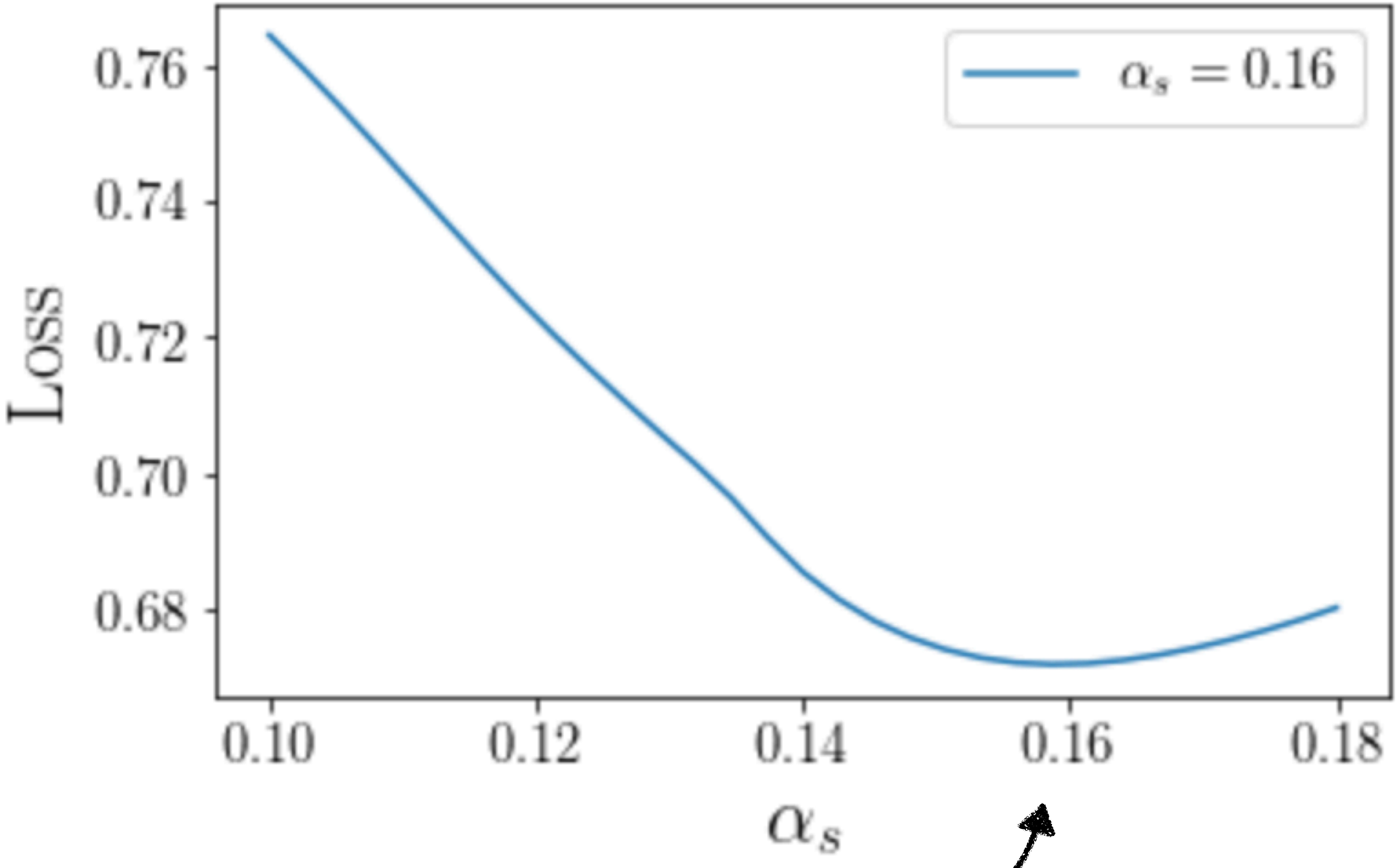
Other DCTR applications

1. DCTR **continuous as a function of MC parameter**

→ **Tune MC to match an unknown sample**, e.g. real data

arxiv1907.08209

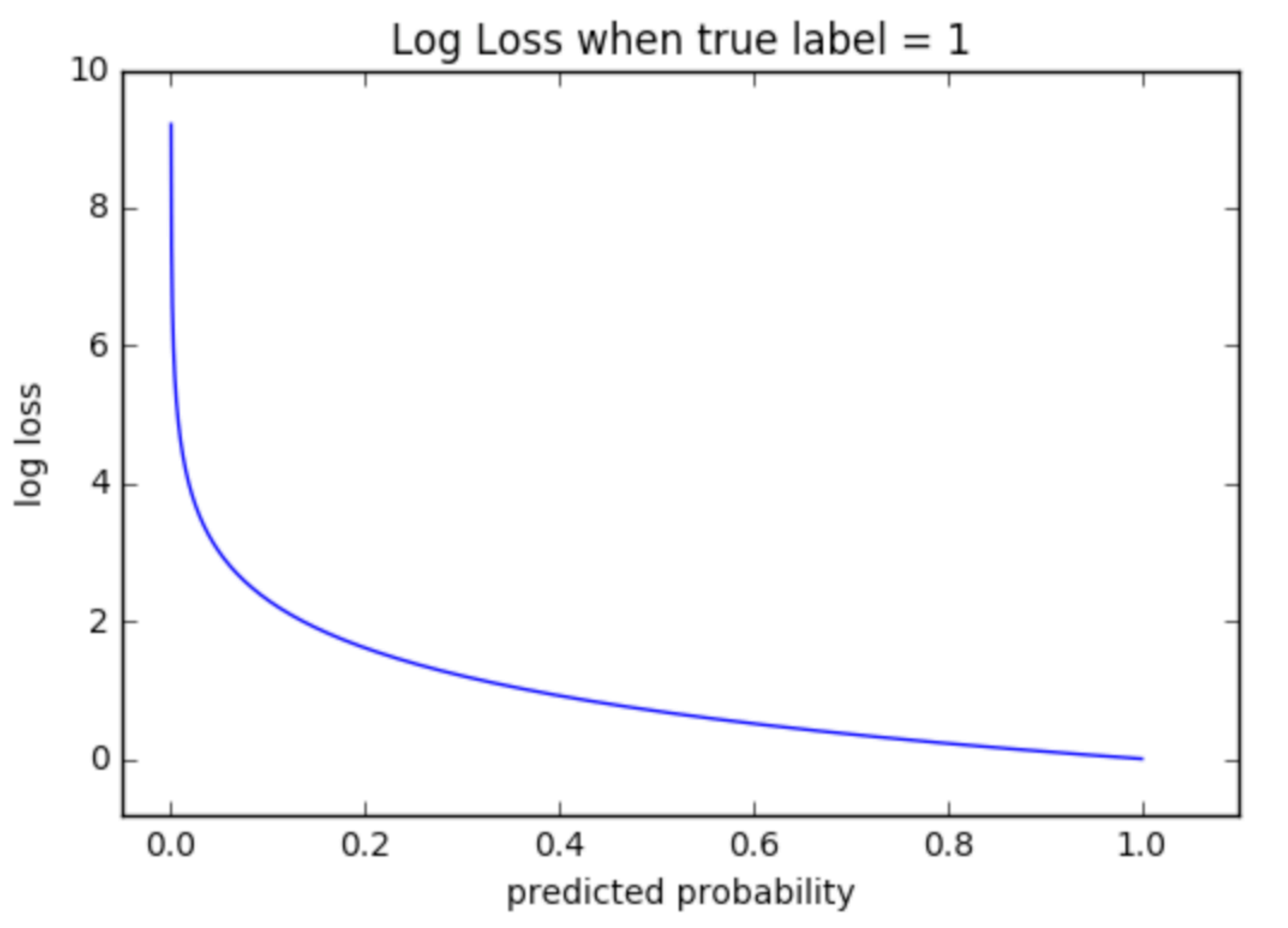
- Passing data to NN, MC parameter value extracted without generating all MC templates



Minimum at the true value

Loss function and optimizer algorithm

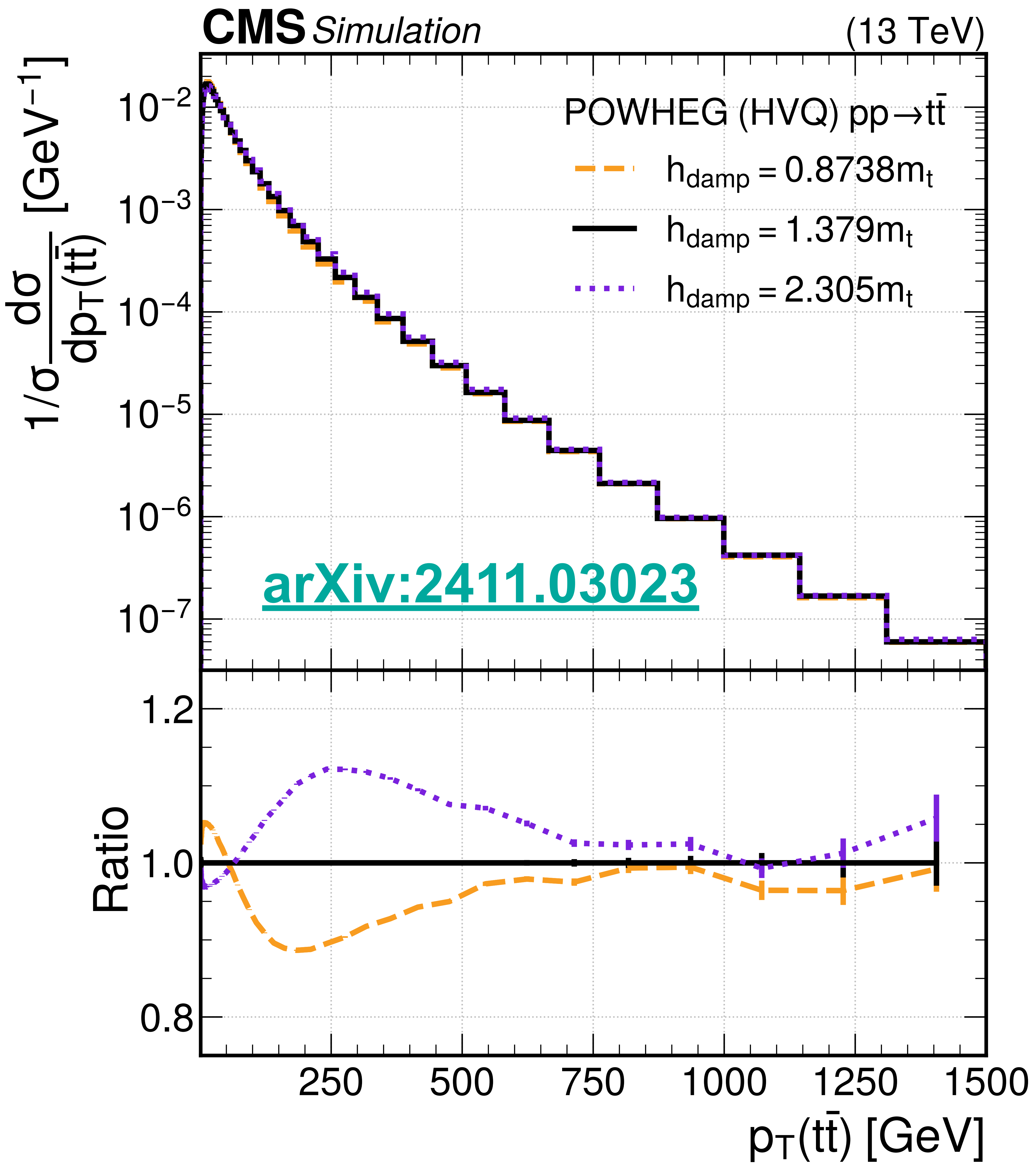
- Loss function: **"cross-entropy loss (or log loss)"**
  - It measures the **performance of** a classification **model** whose output is a probability value between 0 and 1
  - It increases as the predicted probability diverges from the actual label. A **perfect model** would have a **log loss** of **0**
- Optimizer algorithm: **"Adam"**
  - It permits to minimize the loss function
  - It is a stochastic gradient descent method, based on adaptive estimation of first-order and second-order moments



Powheg h_{damp} parameter in top pair production

Heavy quark process of Powheg (arxiv1002.2581):

- **Nominal CMS**: $h_{damp} = 1.379 \cdot m_t$
- **2 CMS variations:**
  - $h_{damp}^{down} = 0.8739 \cdot m_t \qquad h_{damp}^{up} = 2.305 \cdot m_t$



For computation reasons, variation samples produced with less than half the events of the nominal sample→ Decrease precision of analyses

→ Reweighting: same number of events in nominal and variations samples

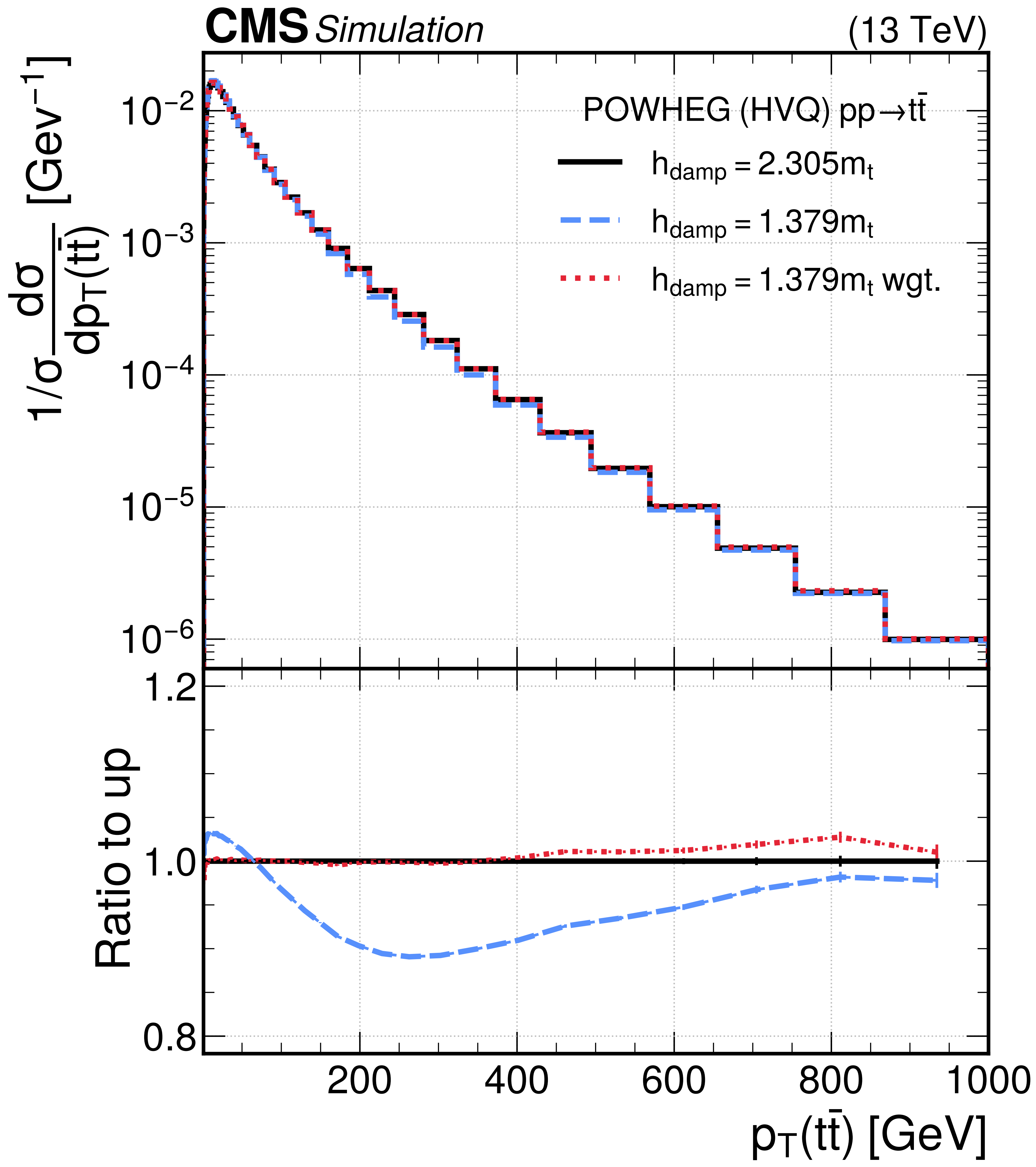
Powheg h_{damp} parameter reweighting

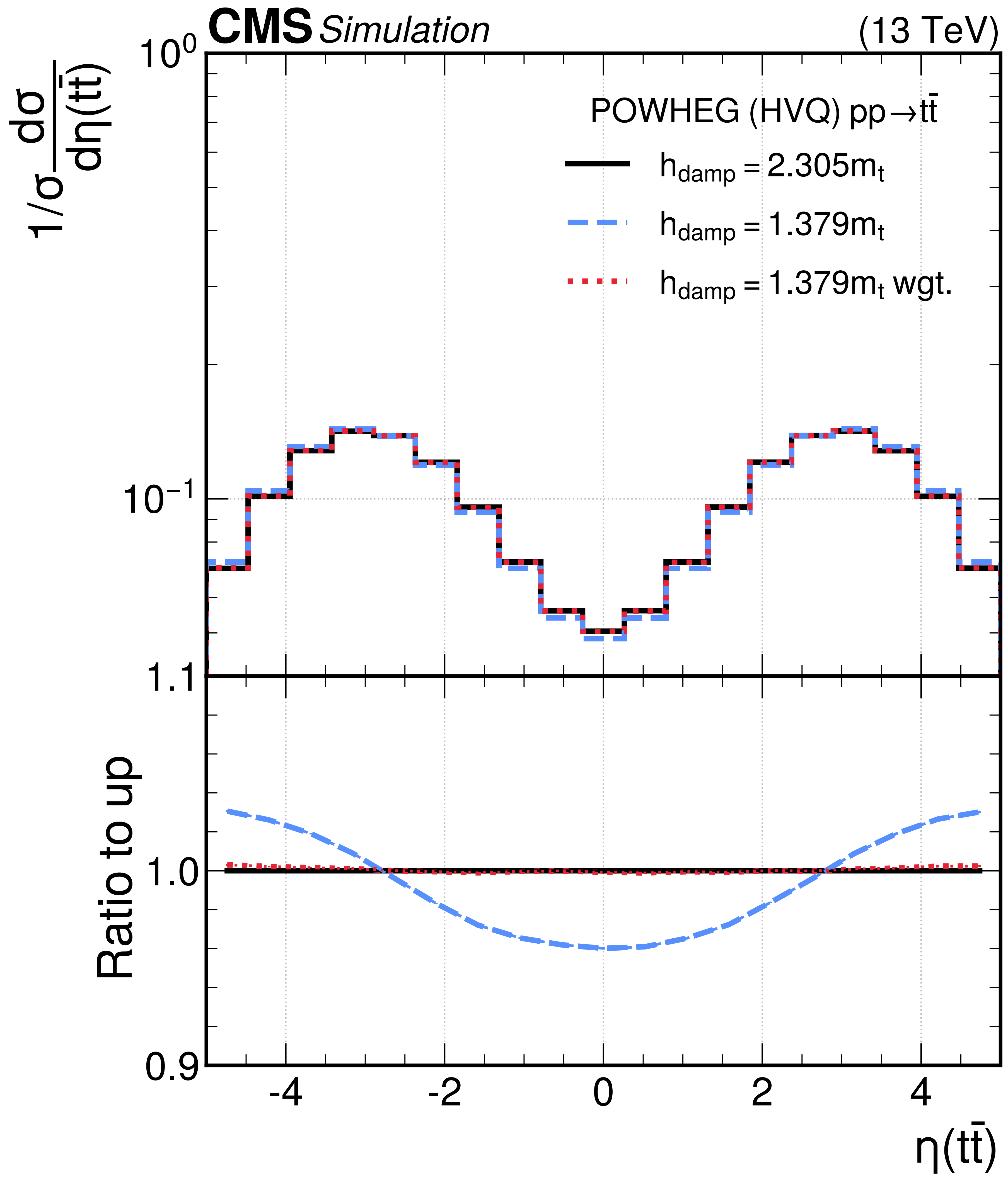
- **Discrete reweighting** trained on 2 values of h_{damp}
- **2 NN models** to reweight:
  - Nominal to up CMS variation of h_{damp}: $(1.379 \to 2.305) \cdot m_t$
  - Nominal to down CMS variation of h_{damp}: $(1.379 \to 0.8739) \cdot m_t$

- **Training & validation samples:**
  - 80M events
  - 75% used for training, 25% for validation

- **Inputs to PFN**
  - Parton level information from LHE files
  - 4-vector (p_T, y, ϕ, m) and PID of top and antitop

h_{damp} reweighting results

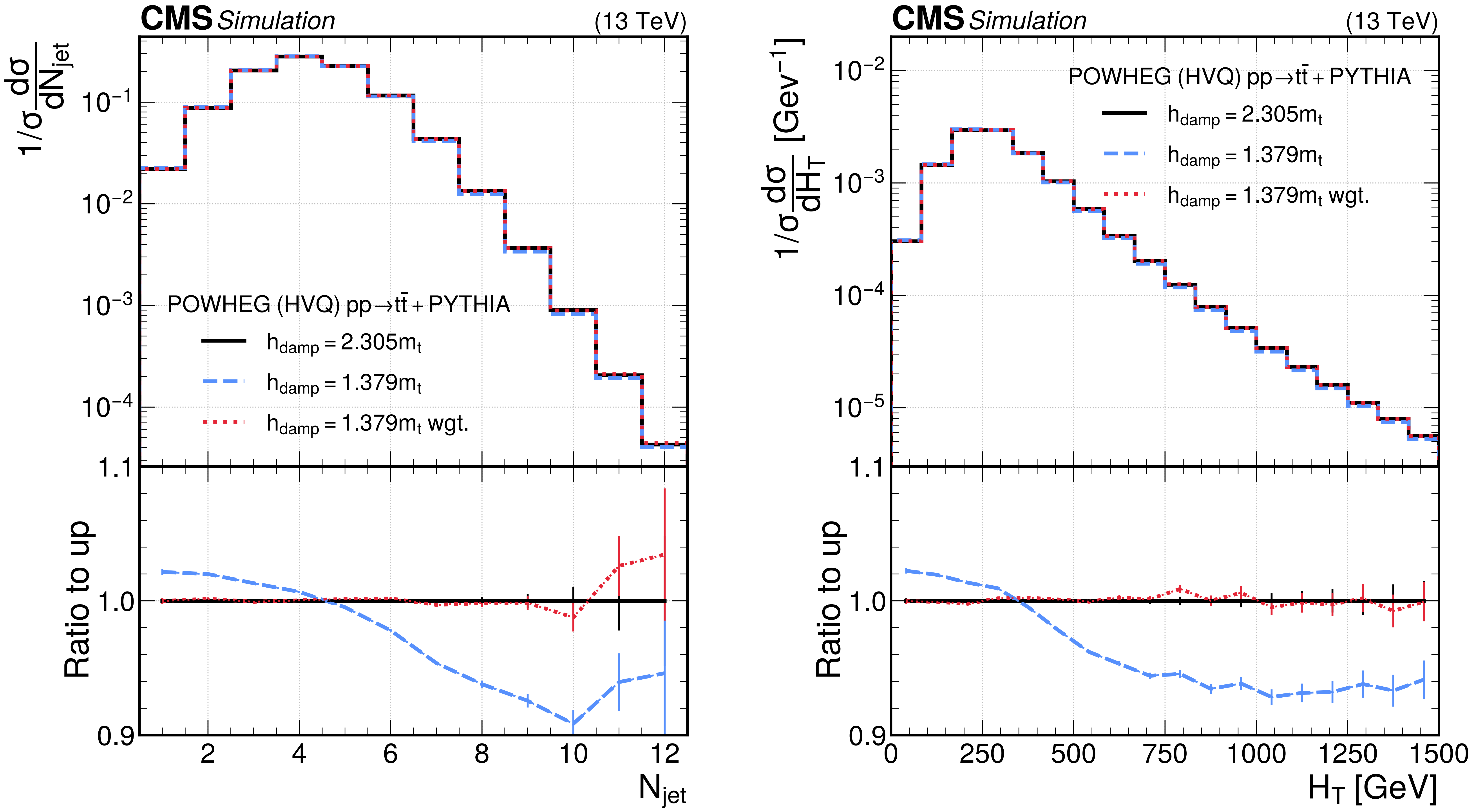




All results from arXiv:2411.03023

h_{damp} reweighting after the shower

- The model is trained at parton level using LHE information
- **The reweighting works well also after showering the events (hvq interfaced with PS generator Pythia)**



$$\hat{p}_T = \sum_{i=0}^{N_{jets}} p_T^i$$

With $p_T > 30$ GeV, $|\eta| < 2.4$

All results from arXiv:2411.03023

DESY.

Pythia r_b parameter reweighting

Rederivation of rb using LEP results: $r_b^{nom} = 1.056$, $r_b^{up} = 1.252$, $r_b^{down} = 0.856$

- **Trained 2 NN models** to reweight:
  ✓ Nominal Pythia to nominal CMS CP5 tune value of r_b:
  $(0.855 \rightarrow 1.056)$

  ✓ Nominal Pythia to up CMS CP5 tune variation of r_b:
  $(0.855 \rightarrow 1.252)$

  ✗ Nominal Pythia to down CMS CP5 tune variation of r_b:
  $(0.855 \rightarrow 0.856) \rightarrow$ NOT sensitive to this small change

B-fragmentation parameter reweighting

- **Continuous reweighting** trained on 10 values of r_b

- One single **NN model** to reweight:
  - 9 different r_b values to the CMS nominal one ($r_b = 0.855$)

- **Training & validation samples:**
  - 10M events
  - 90% used for training, 10% for validation

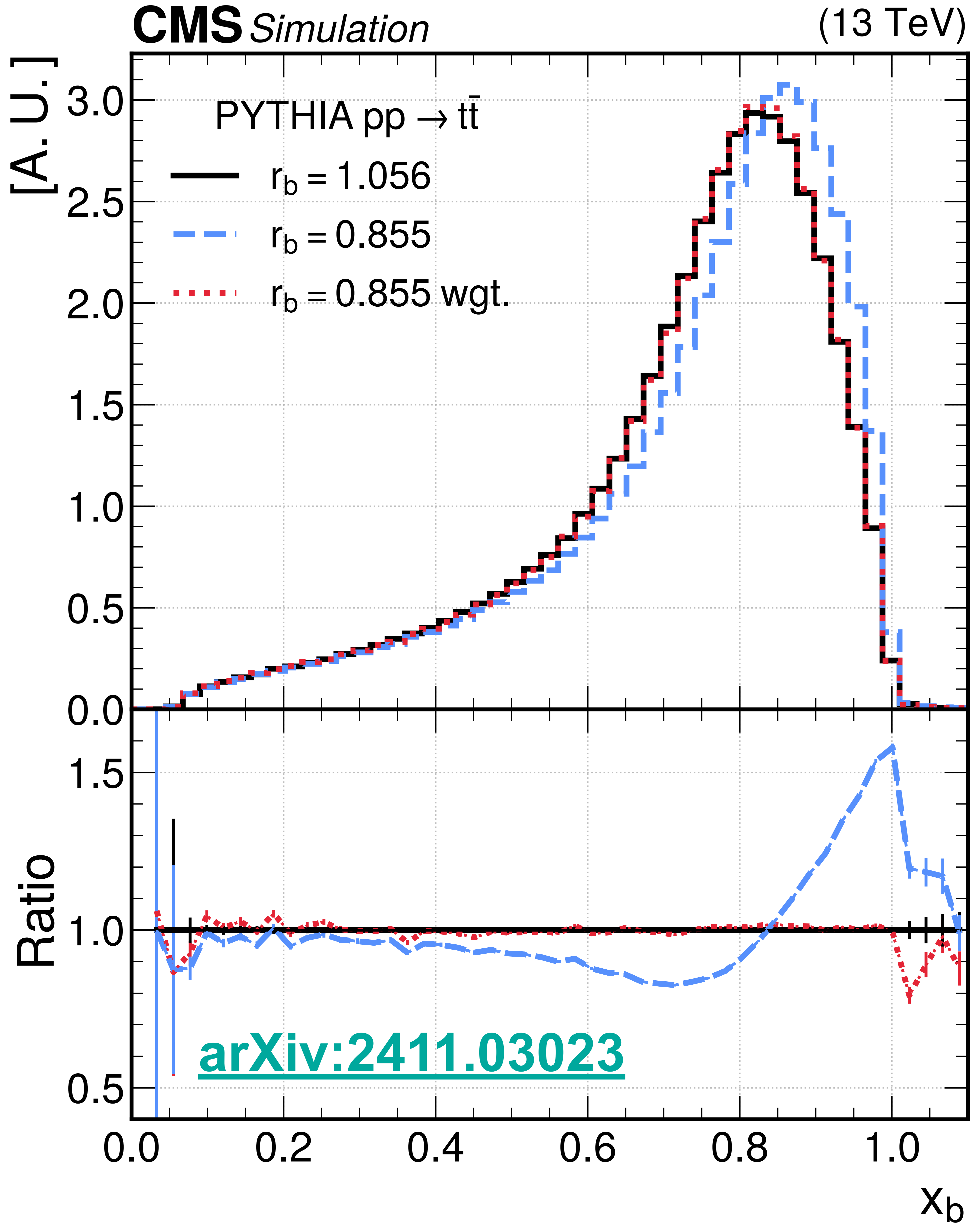
- **Inputs to PFN**
  - Particle level information from Pythia samples:
  - 2 x_b from **2 b-quarks** decaying from tops

r_b parameter reweighting results

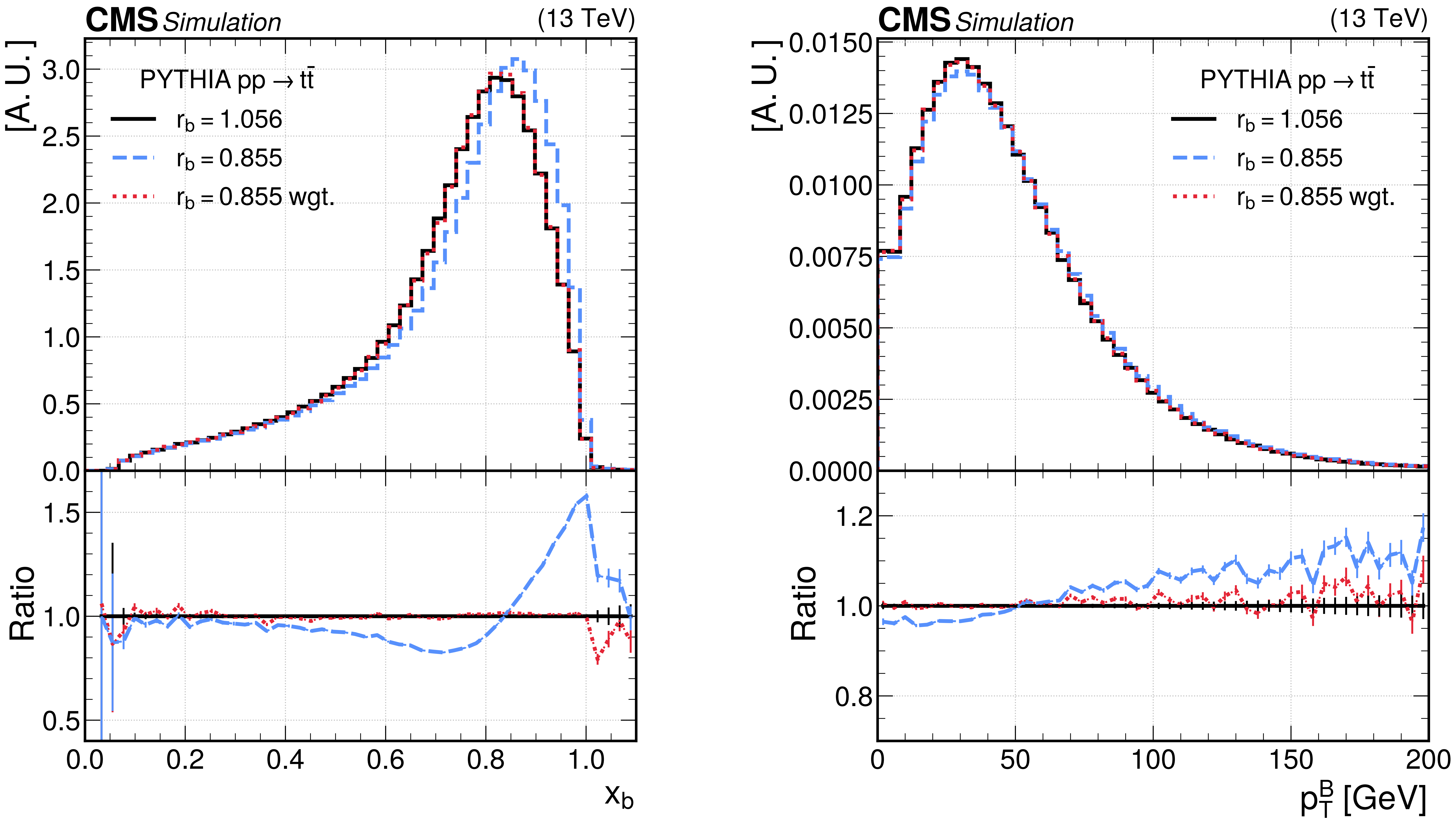
- **Goodness of reweighting checked with a reweighting closure:**
  - Comparison between reweighted and target sample
    - Target: sample generated with $r_b = 1.056$
    - Reweighted sample: sample generated with $r_b = 0.855$ and reweighted to $r_b = 1.056$ using a test sample

- **Test sample:** 500k events generated for each r_b value, orthogonal to trained and validation samples

- **Reweighting closure within 2% up to $x_b < 1$**

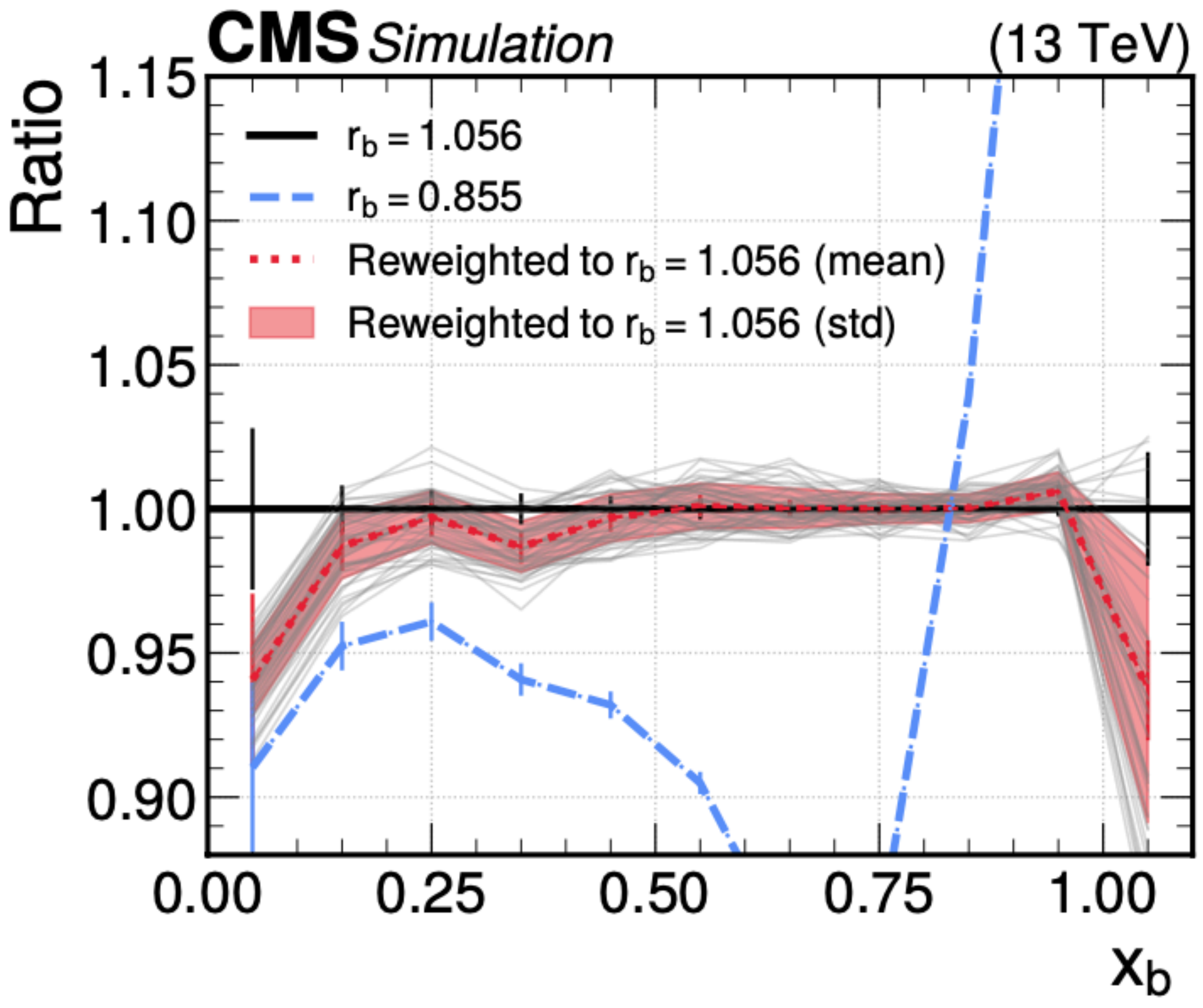


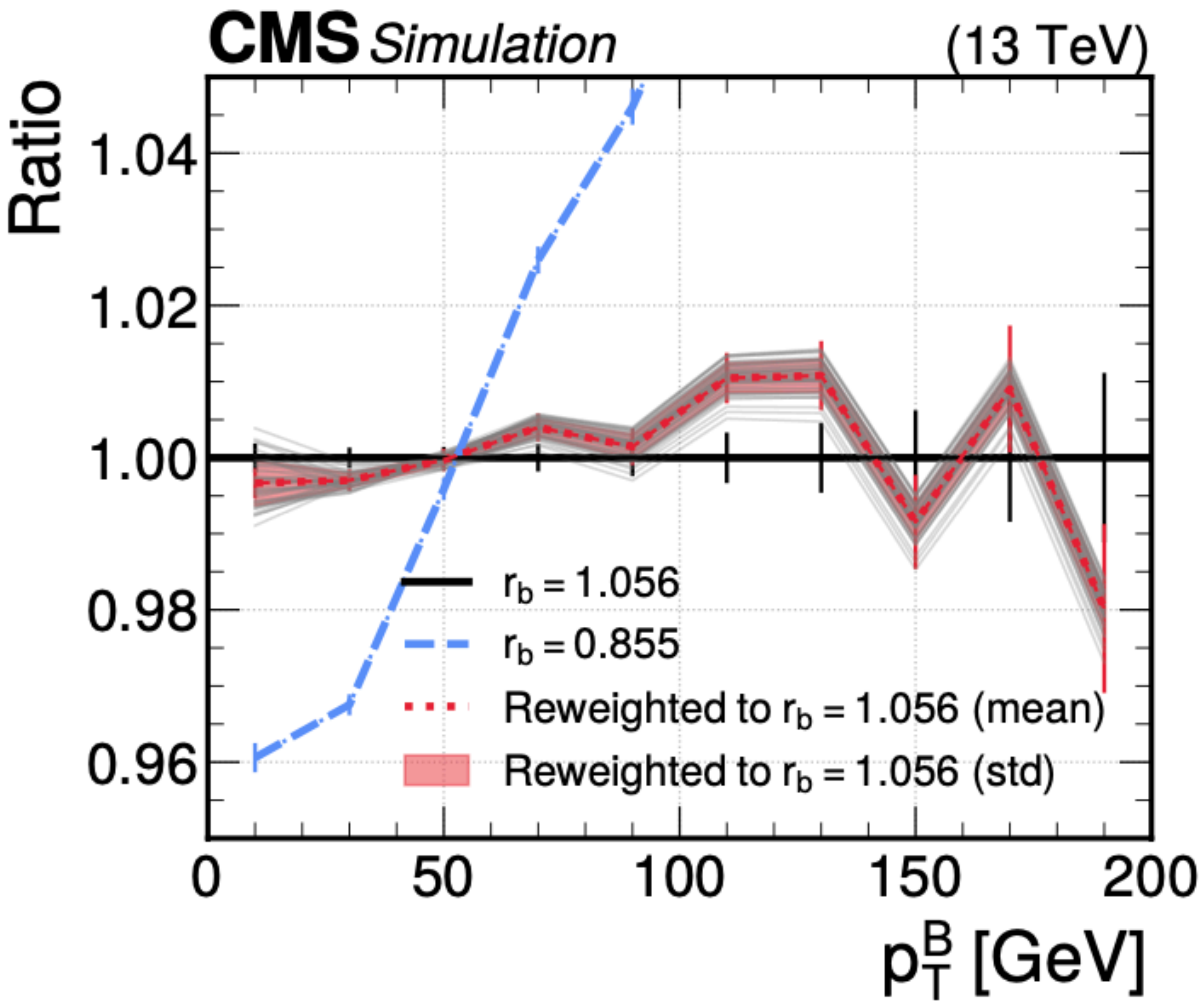
r_b parameter reweighting results



All results from arXiv:2411.03023

r_b parameter reweighting results





All results from arXiv:2411.03023

Model reweighting

- **Training & validation samples:**
  - 10M events at NLO, 10M events at NNLO
  - ~75% used for training, ~25% for validation

- **Test samples:**
  - 10M events at NLO, 10M events at NNLO

- **Inputs to PFN**
  - Parton level information after shower
  - 4-vector (p_T, y, ϕ, m) and PID of top and antitop, and ttbar system
  - 4-vector of additional partons not included, since the NN architecture is not suitable to reweight a 3D to a 4D phase space

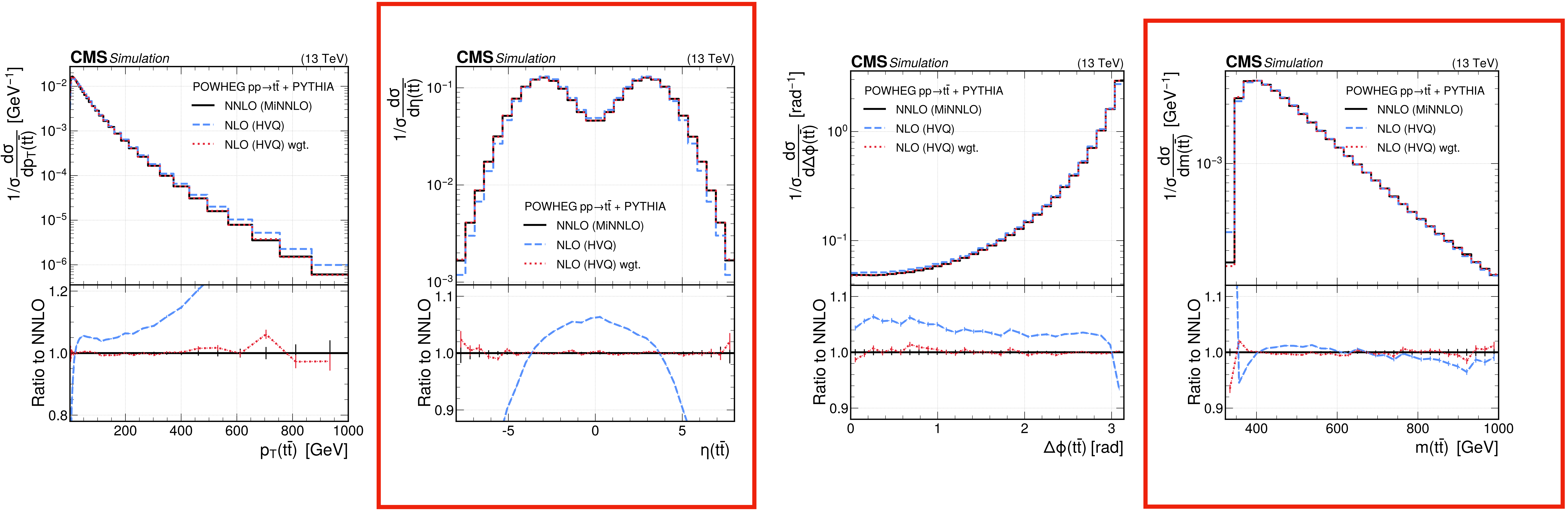
Negative Event weights

- **Negative event weight** comes from the cancellation of soft and collinear real emissions by corresponding virtual corrections
- <1% for NLO accuracy, 10% for NNLO accuracy
- The binary cross-entropy is negatively unbounded for negative event weights → The classification task can become impossible
- This effect can be mitigated by using of a large batch size, which reduces the risk of a single event dominating the loss function.
- This approach works for NLO simulations, not for NNLO ones

- **The Categorical Cross Entropy loss function can not learn negative weights**
- **Mean Square Error loss can learn negative weights when using enough large batch size**

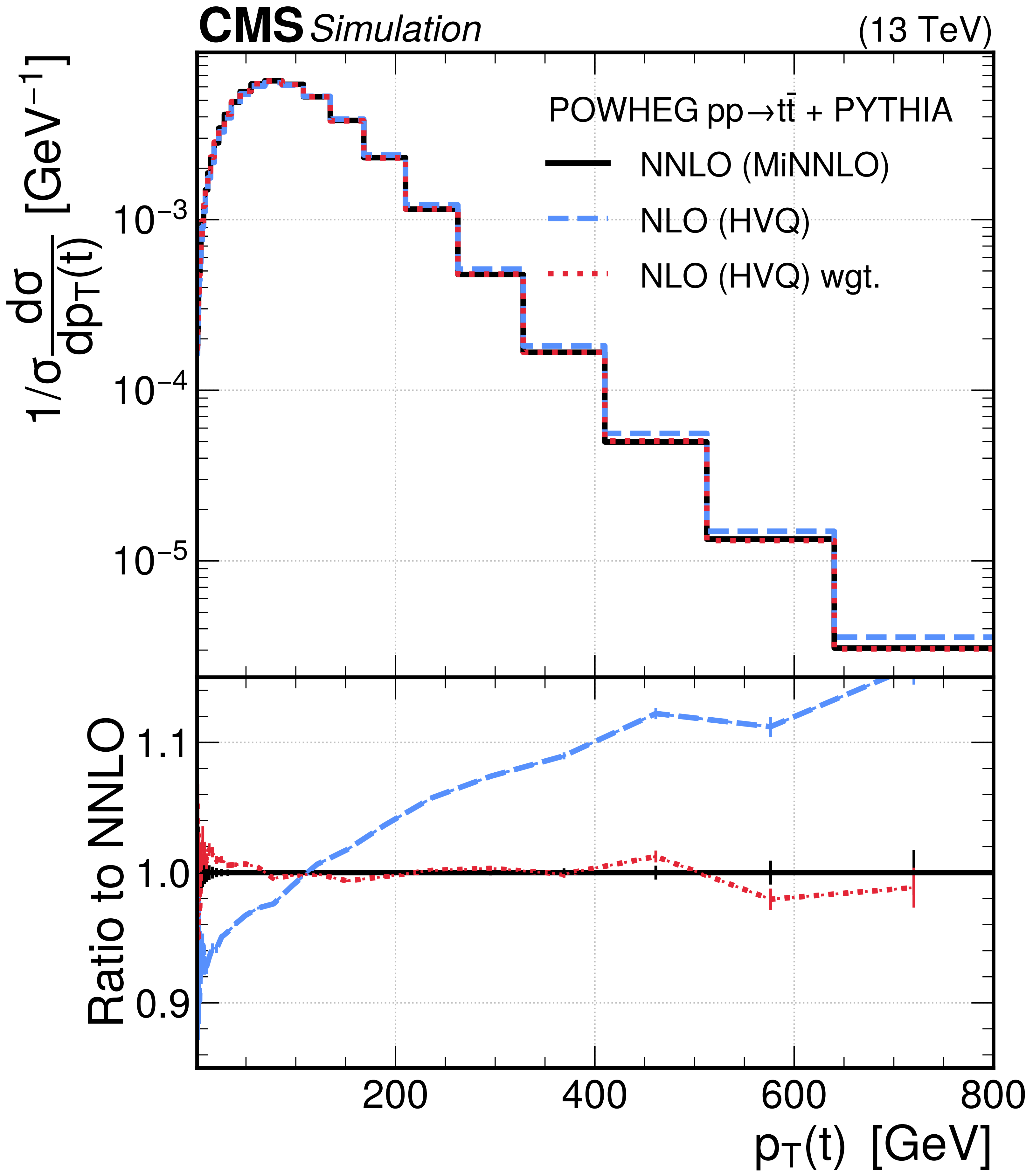
MiNNLO reweighting

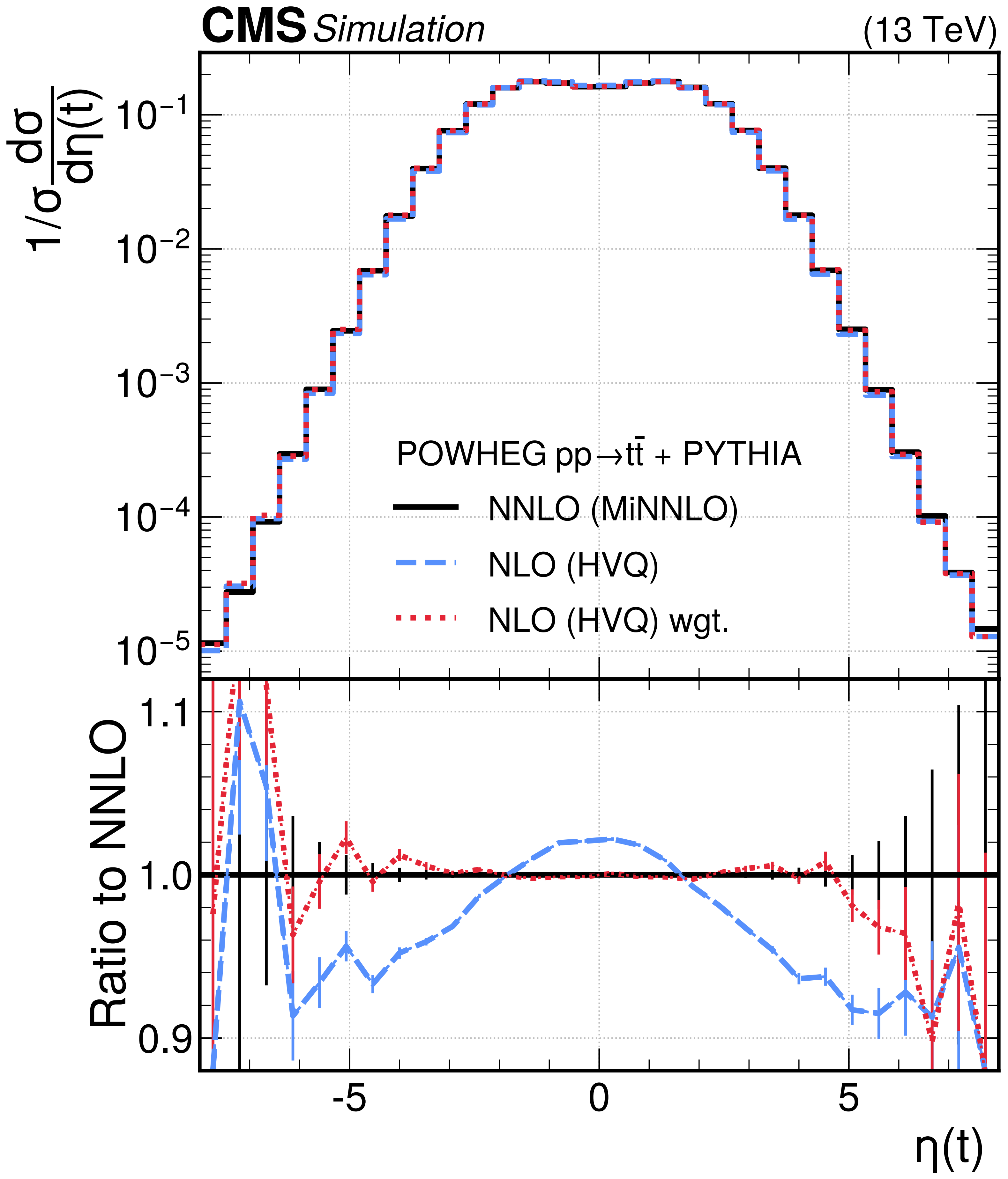
The method works well also on observable we didn't train on



All results from arXiv:2411.03023

MiNNLO reweighting: top observables



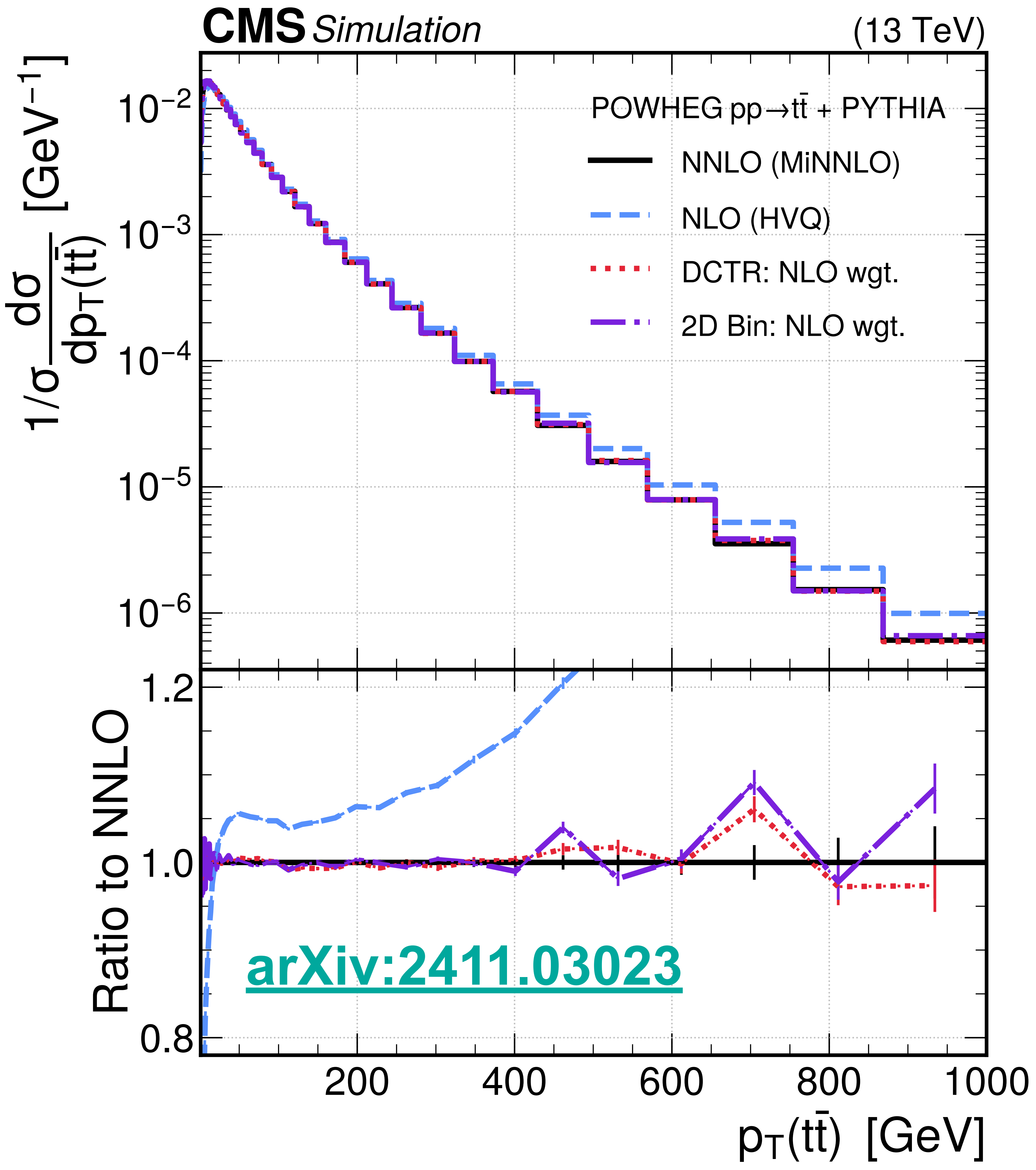


All results from arXiv:2411.03023

DCTR compared to 2D bin reweighting

Comparing DCTR to 2D bin reweighting

- The 2D reweighting is done with p_T and η of $t\bar{t}$ system
- Check the goodness of the two reweightings on $p_T(t\bar{t})$

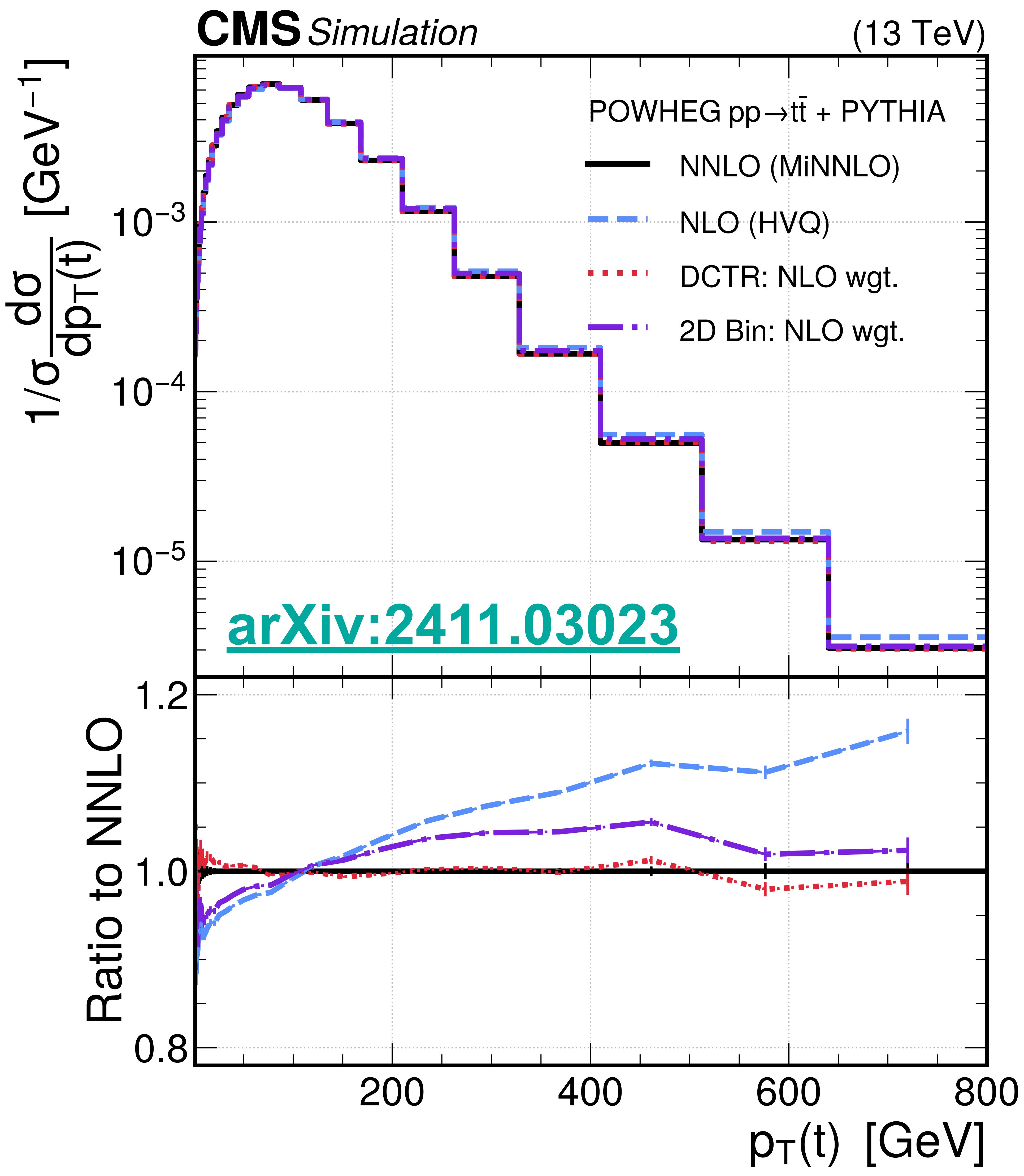


- **Both methods work well on variables used in the 2D reweighting**

DCTR compared to 2D bin reweighting

Comparing DCTR to 2D bin reweighting

- The 2D reweighting is done with p_T and η of $t\bar{t}$ system
- Check the goodness of the two reweightings on $p_T(t)$



- 2D reweighting improves $p_T(t)$ but still large deviations respect target
- DCTR uses the whole phase space for reweighting

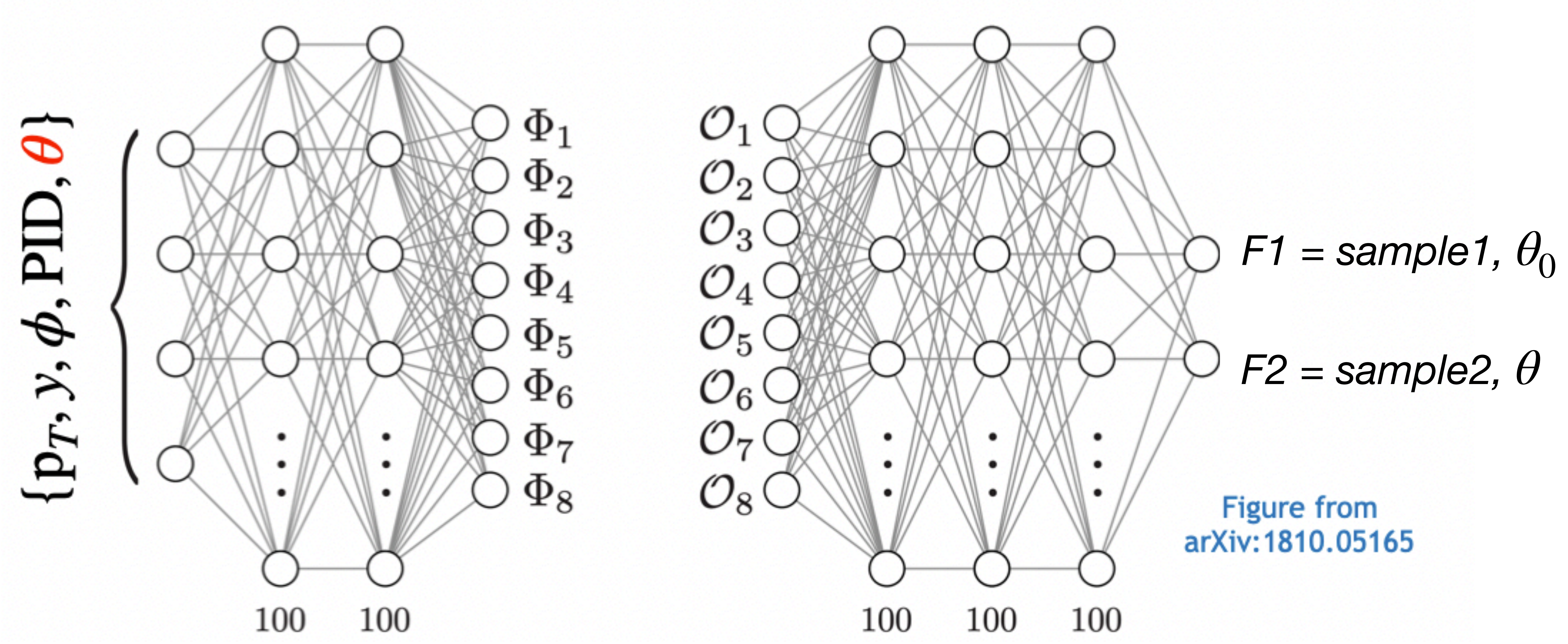
  →It works well on any projections

The method: NN architecture

Particle Flow Network (PFN) (arxiv1810.05165)

$$f(\{p_i\}) = F\left(\sum_{i=1}^{n} \Phi(p_i)\right)$$

p_i: properties of particle i and parameter to reweight θ



softmaxed discriminant

Φ processes each particle individually, providing a per-particle internal (latent) representation

$$\boxed{O_a = \sum_i \Phi_a(p_i)}$$

F takes the sum of Φ latent representations from all particles to produce an overall event-level representation

Dealing with negative Event weights

$$L_{BCE}(f) = -\frac{1}{N}\sum_{i}^{N} w_i^{MC}(y_i \cdot \log f(x_i) + (1 - y_i) \cdot \log(1 - f(x_i))$$

$$L_{MSE}(f) = \frac{1}{N}\sum_{i}^{N} w_i^{MC}(f(x_i) - y_i)^2$$

yi: true label of each event (between 0 or 1 according to which class it belongs to)

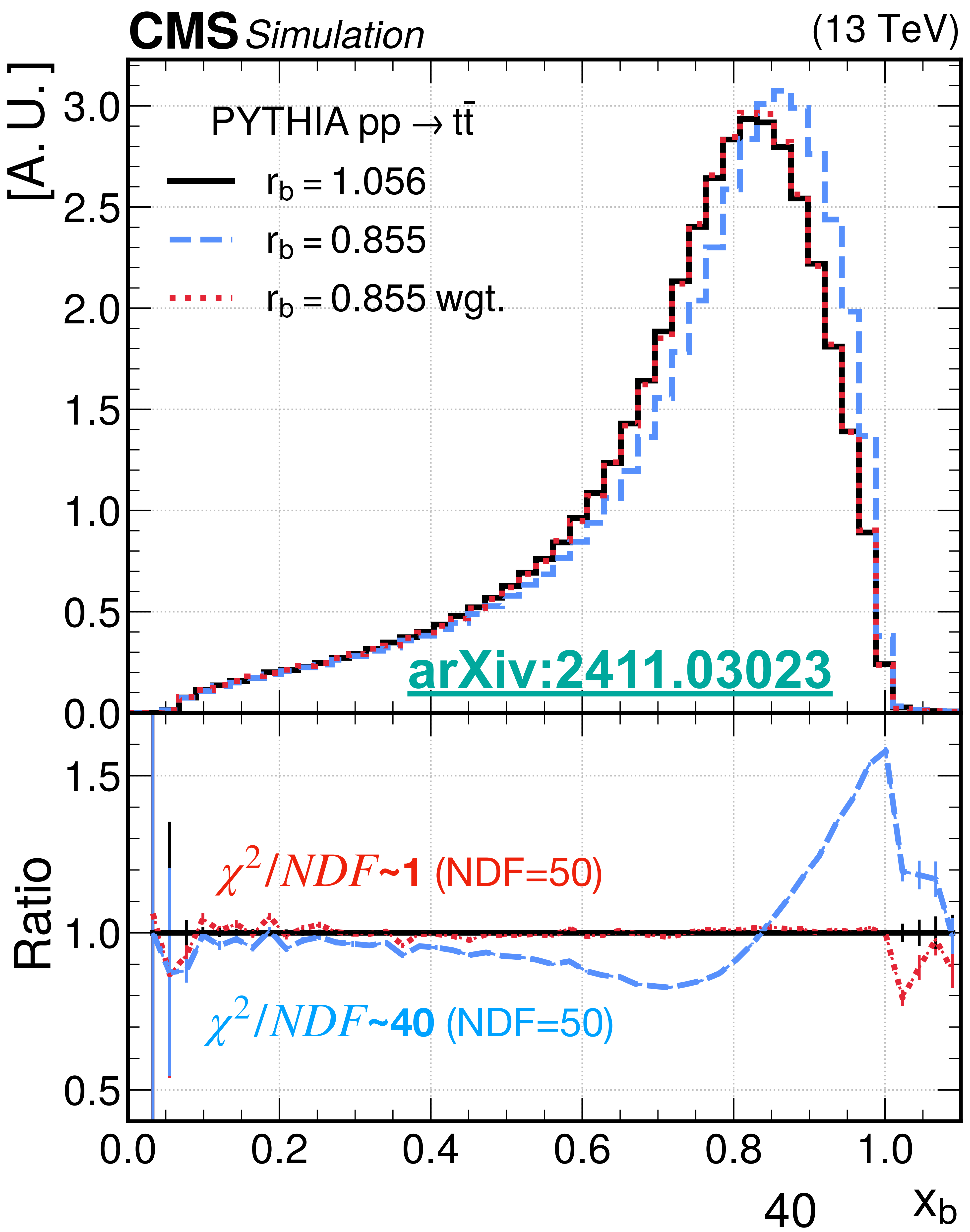
f(xi): predicted probability (between 0,1)

$\mathbf{w_i^{MC}}$: MC event weight

B-fragmentation discrete reweighting

Reduced χ^2 to test goodness of reweighting

- **Before reweighting:** difference between 2 samples with different r_b
- **After reweighting:** difference between reweighted sample and target one



Technical information: PFN architecture

All models are implemented in Keras with the Tensorflow backend

- **Technical details:**
  - **Latent space dimension:** l=128
  - **Activation func:** ReLu
  - **Classification output func:** softmax
  - **Loss func:** crossentropy loss
  - **Optimizer:** Adam***
  - **Learning rate:** 0.01**
  - **Early stopping with patience 10** ****

*** to update the NN parameters (weights and biases), to minimise the cross-entropy loss function for 100 epochs.

****To prevent overfitting

This architecture has been already optimised by the authors for particle physics.

Pythia B-fragmentation parameter in top pair production

B-fragmentation uncertainty: variations of r_b parameter of Lund-Bowler function in PYTHIA8

$$\boxed{f_B(z) \propto \frac{1}{z^{1+br_b m_B^2}}(1-z)^a exp(-bm_B^2/z)}$$

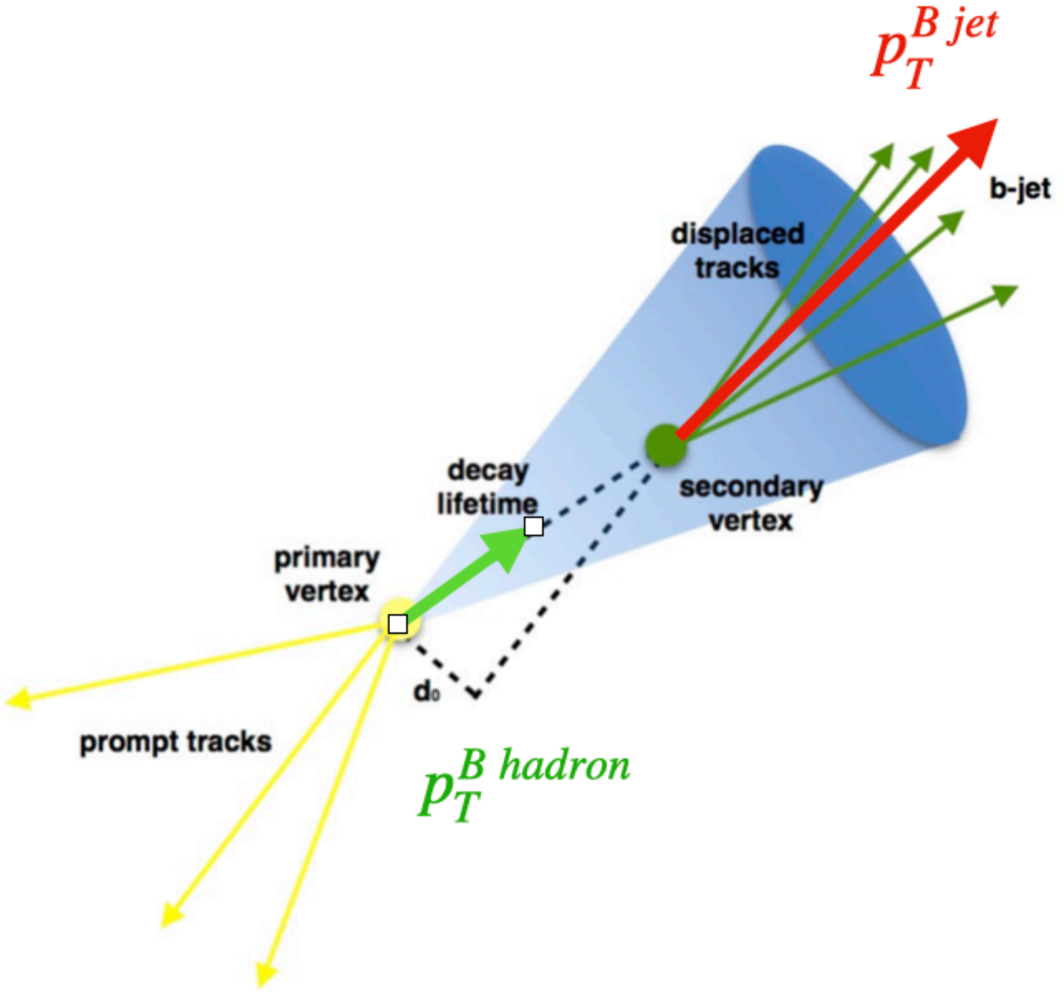
m_t , m_b: top & b quark mass
a, b: terms related to light quarks
r_b: term related to b quark
a, b, r_b free parameters to be tuned to data

In CMS only the sample with PYTHIA nominal $r_b = 0.855$ produced, no variations

→ Crucial to use a reweighting method to produce the variations

- **Variable to reweight systematic uncertainties in CMS:**
  - Ratio in bins of two distributions with different w at truth level

$$w = \frac{p_T^{B\ hadron}}{p_T^{B\ jet}}$$



DCTR reweighting → full phase space reweighting

Statistical uncertainty of the method

- Training is repeated 50 times bootstrapping the data
- The goodness of the reweighting with the 50 trained model is checked and the mean and the standard deviations of the models computed in each bin
- **Our model is compatible with the target one within the statistical uncertainty of the method**