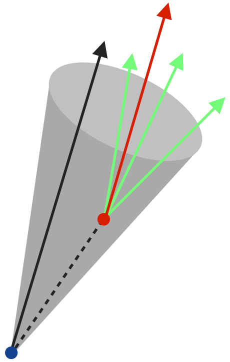


Adversarial training for b-tagging algorithms in CMS

Annika Stein[1]

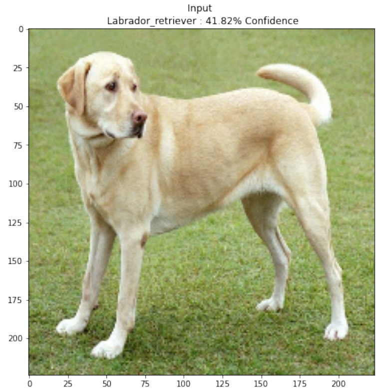
On behalf of the CMS Collaboration

ML4Jets Workshop 2022
03.11.2022

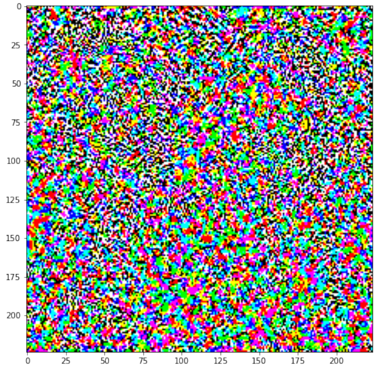
[1]

Why AI safety for jet tagging algorithms?

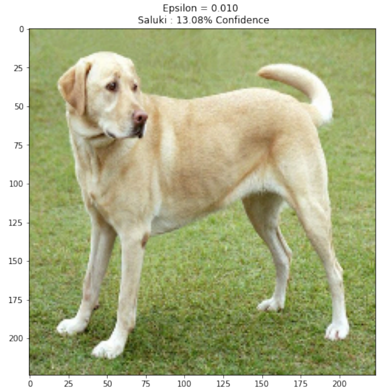
Computer Vision



$+ \ \epsilon \cdot$



$\rightarrow$



[1]

Classifier: labrador

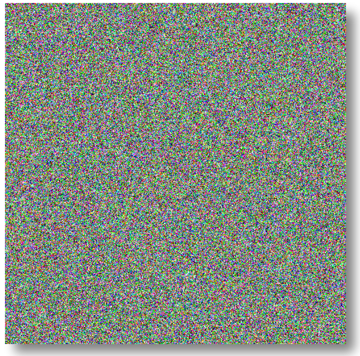
Classifier: saluki

Our case

Jet,
Track and
Secondary vertex
properties of a

b jet

$+ \ \epsilon \cdot$



$\rightarrow$

Slightly distorted
Jet,
Track and
Secondary vertex
properties of a

b jet

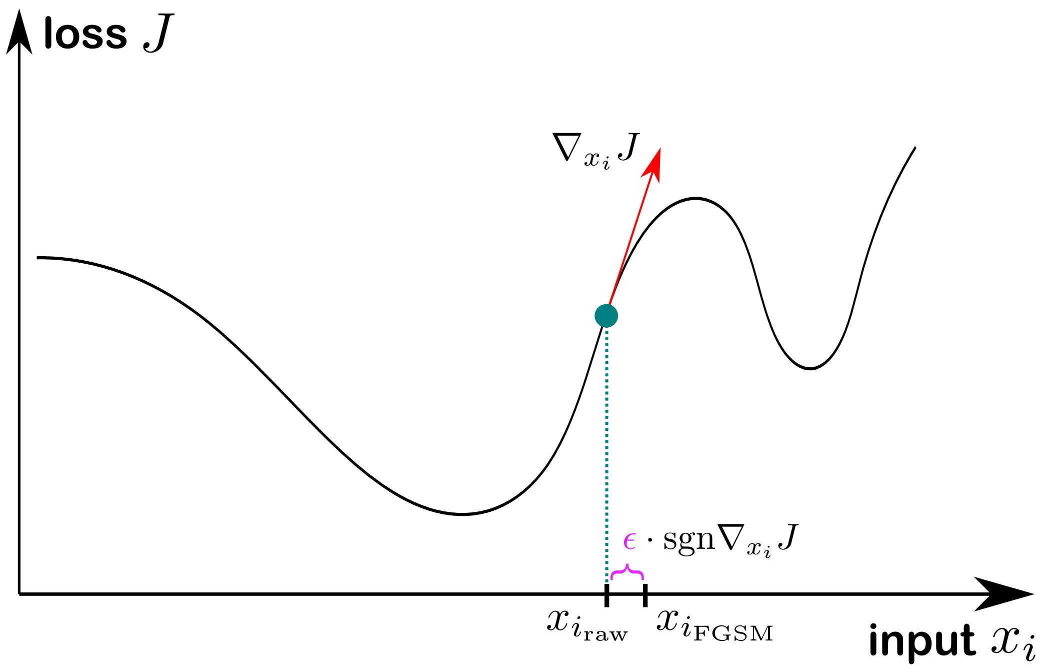
Classifier: b jet

Classifier: light jet

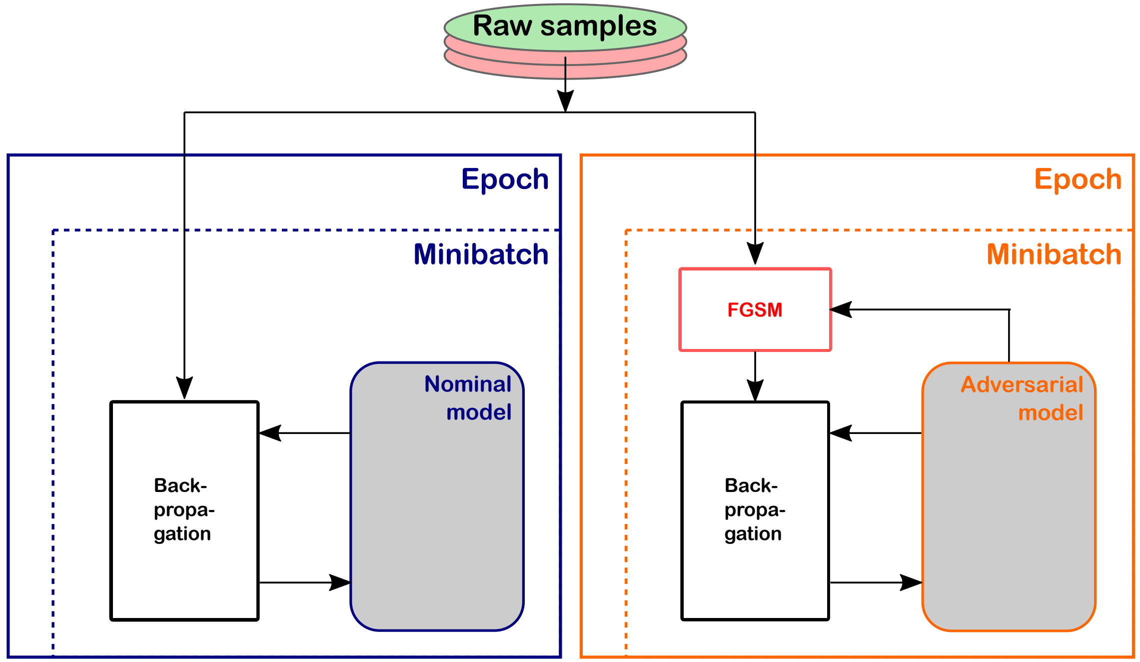
[1] Reproduced from work created and shared by Google and used according to terms described in the Creative Commons 4.0 Attribution License. (https://www.tensorflow.org/tutorials/generative/adversarial_fgsm). Labrador Retriever by Mirko CC-BY-SA 3.0 from Wikimedia Commons.

How? — Utilized methods

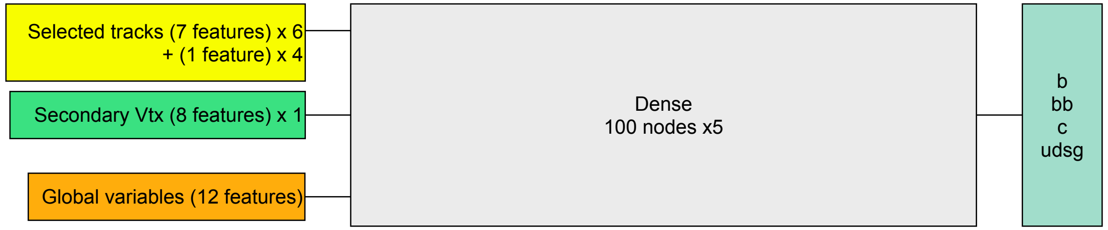
Adversarial attack — Fast Gradient Sign Method

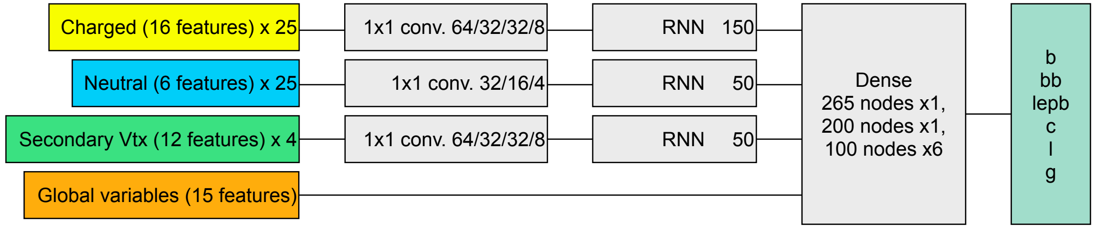


Defense — Adversarial training
(instead of nominal training)



Jet tagging algorithms at CMS

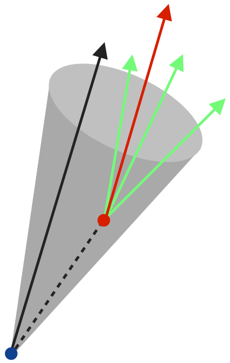


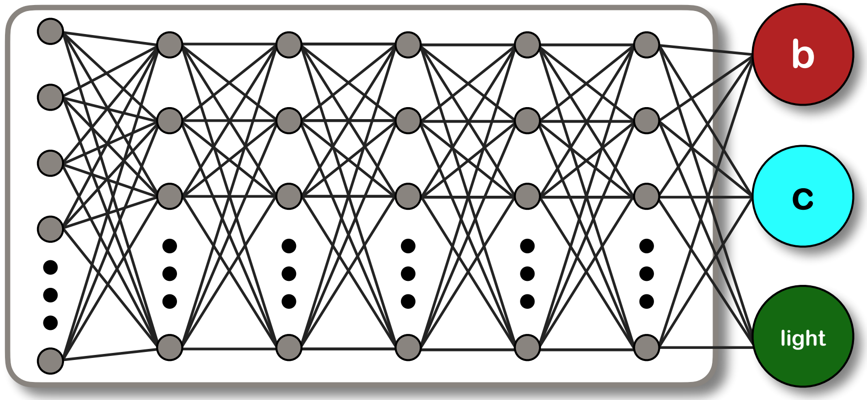


DeepCSV	DeepJet
Only fully-connected (**dense**) layers	**Convolutional** layers, recurrent layers (**LSTMs**), dense layers
66 features, from up to six tracks, one secondary vertex, and high-level jet features	**613** features, of which **many** are **low-level** features directly from up to 25 ParticleFlow candidates (charged & neutral) and four secondary vertices; high-level features from DeepCSV
Four outputs: b, bb, c, udsg	**Six outputs**: b, bb, lepb, c, uds, g

Typical workflow:
- train on **simulation**
- evaluate on simulation & **data**
- observe **differences** and correct by calibrating via scale factors

Previous and current studies





Deep Neural Network (DNN)

Investigate DeepCSV

(FCNN on `Delphes` simulation)

Investigate DeepJet

2020 — 2021

2021 — 2022

2022 —

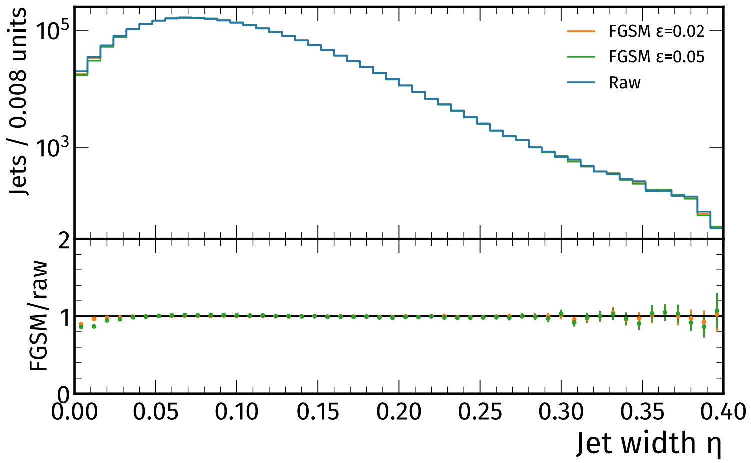
„Improving Robustness of Jet Tagging Algorithms with Adversarial Training“
A. Stein, X. Coubez, S. Mondal, A. Novak, and A. Schmidt, *Comput Softw Big Sci* 6 (2022) 15, **https://doi.org/10.1007/s41781-022-00087-1**, **arXiv:2203.13890**, Code available at: https://github.com/AnnikaStein/Adversarial-Training-for-Jet-Tagging

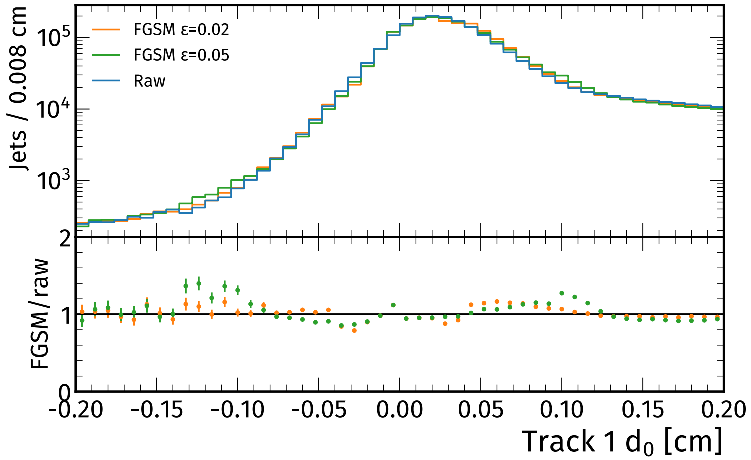
„Adversarial training for b-tagging algorithms in CMS“
CMS Collaboration, CMS DP-2022/049, https://cds.cern.ch/record/2839919?ln=en

From DELPHES sim

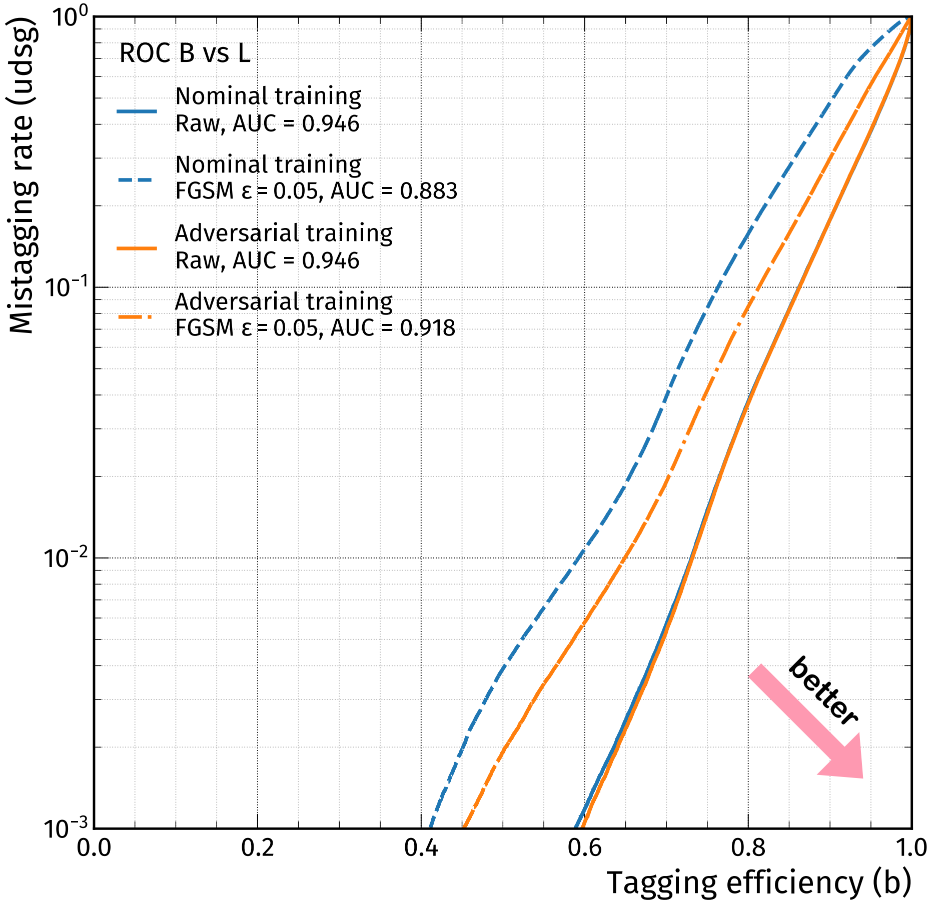
Applying adversarial attacks to jet classification

Effect on the inputs:
- within typical envelopes or **negligible**





Effect on performance:
- Suffers **dramatically** for **nominal** model
- **Adversarial** model: more **robust** (+ high performance)



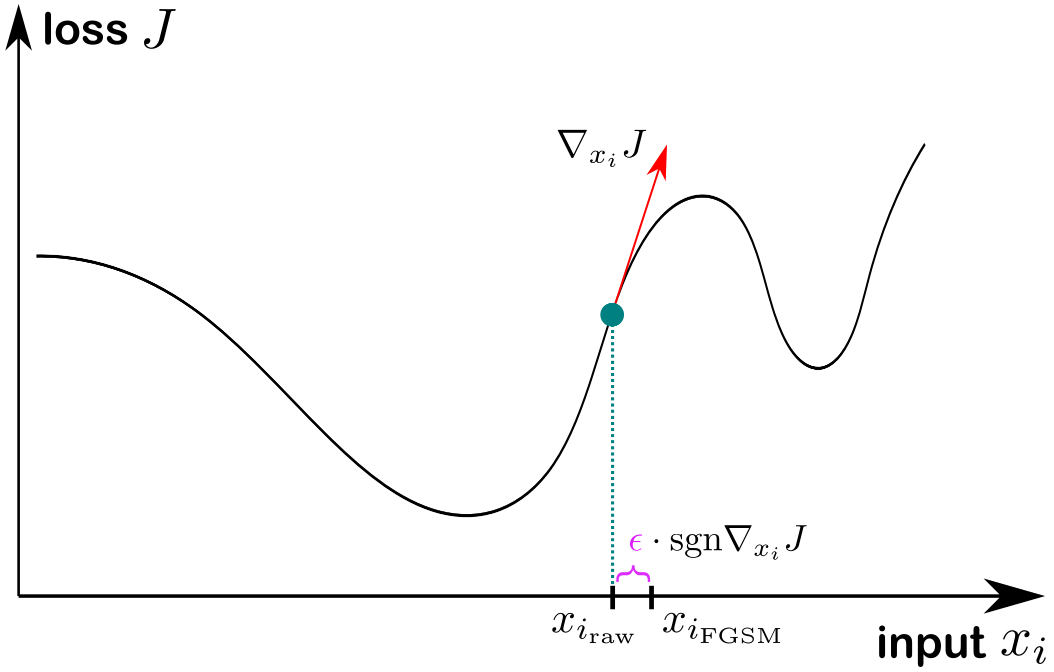
Physics Institute III A RWTH AACHEN UNIVERSITY

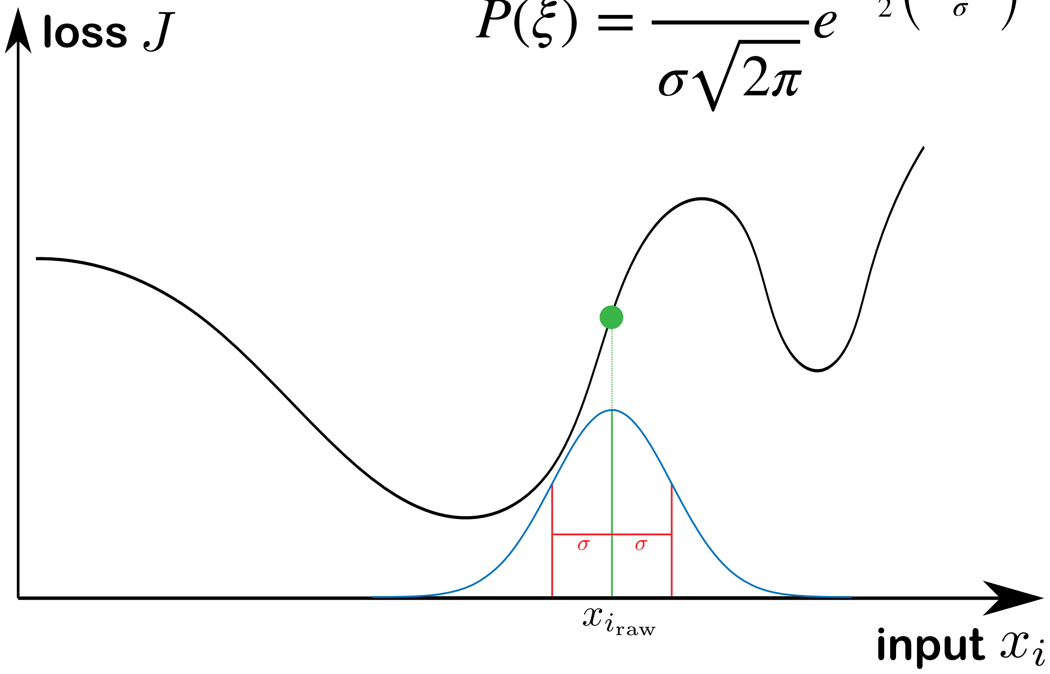
FGSM attack versus Gaussian noise

$$x_{\mathrm{FGSM}} = x_{\mathrm{raw}} + \epsilon \cdot \mathrm{sgn}\left(\nabla_{x_{\mathrm{raw}}} J(y, x_{\mathrm{raw}}) \right)$$

$$x_{\mathrm{noise}} = x_{\mathrm{raw}} + \xi, \qquad \xi \sim \mathcal{N}\left(\mu, \sigma^2\right)$$

$$P(\xi) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\xi-\mu}{\sigma}\right)^2}$$





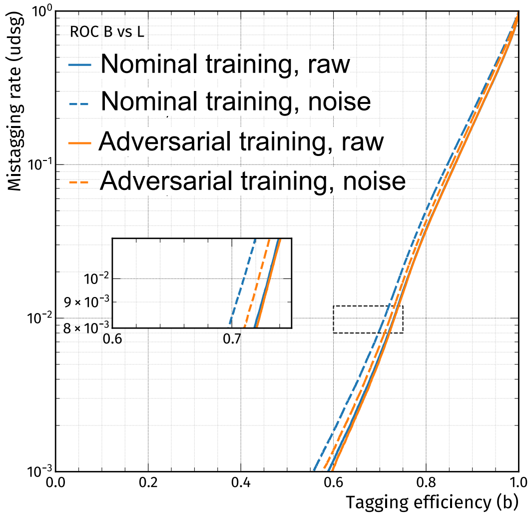
From DELPHES sim

Robustness under typical mismodeling scenarios

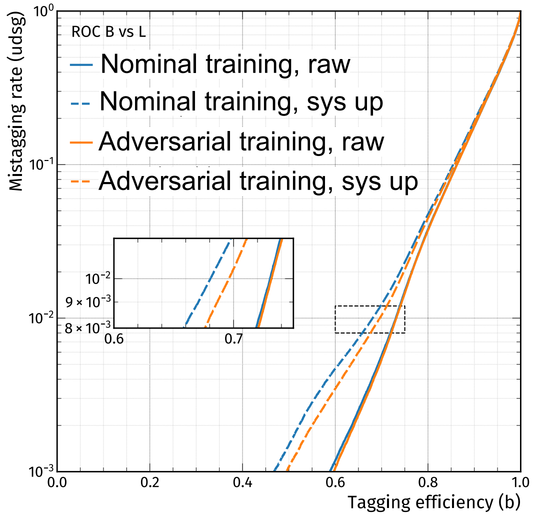
Being robust against FGSM is great, but: **detector effects** or **simulation artifacts** don't know how the loss surface looks like!

➔ Test other scenarios (some of which are closer to **physics / systematic uncertainties**)

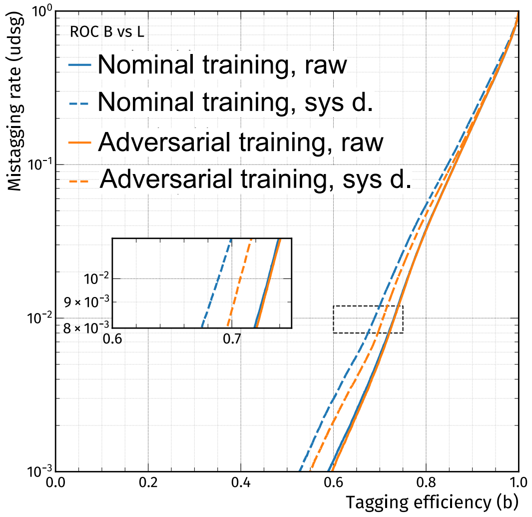
Gaussian noise



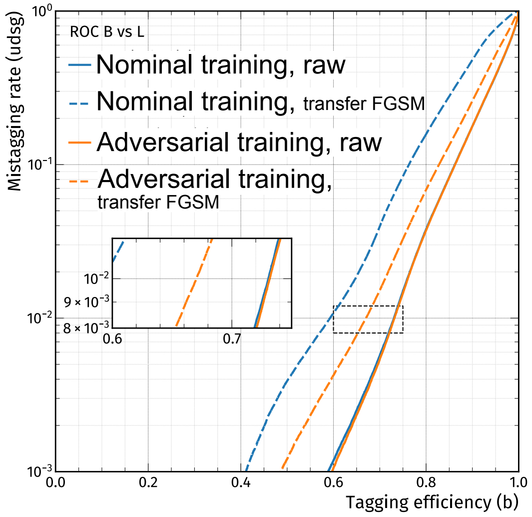
All inputs systematically up



All inputs systematically down



Transfer FGSM from A→B



- In all cases, adversarial training is more robust than nominal training

From DELPHES sim

The relation between robustness and performance

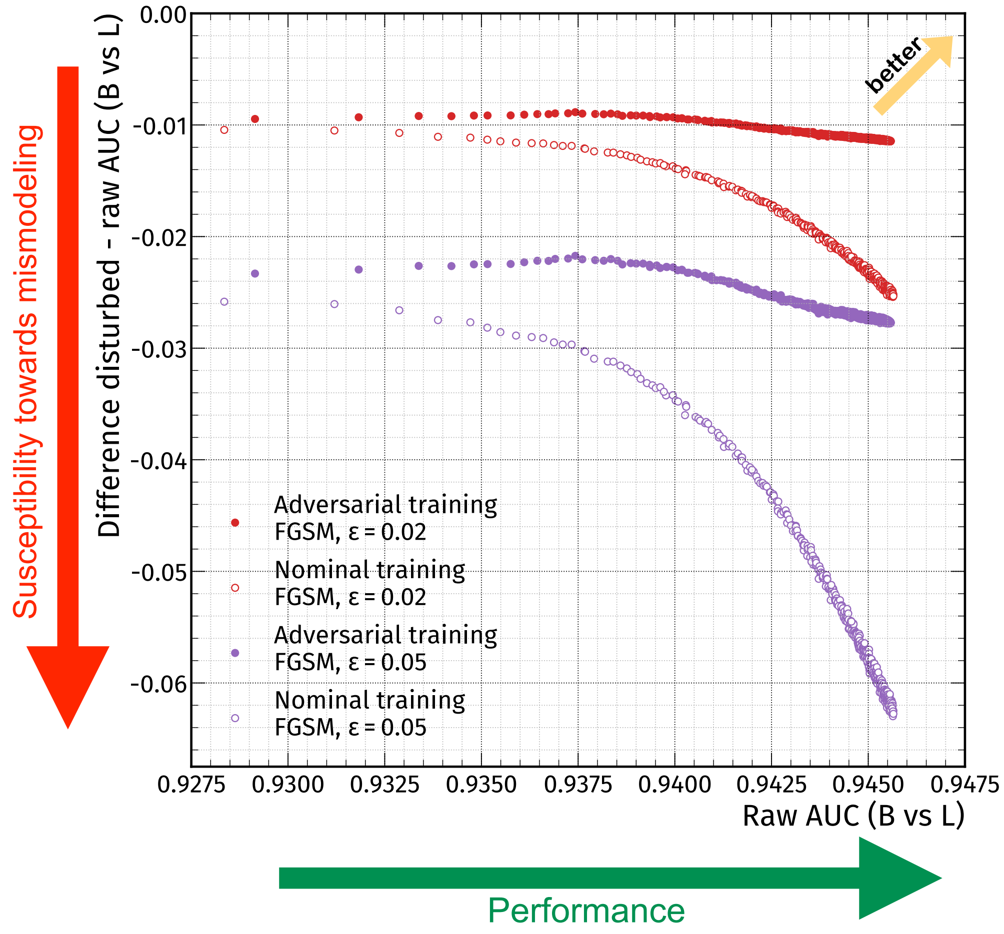
- Track the **evolution** of the model's **performance** after every iteration through the full training sample (= set a checkpoint per epoch)

- Compare the **original test set** with a **systematically distorted** one, specifically crafted for this epoch

Nominal training:
- Gain in **performance** ➠ **less robust** against FGSM

Adversarial training:
- **Maintain** high performance, even for FGSM samples!

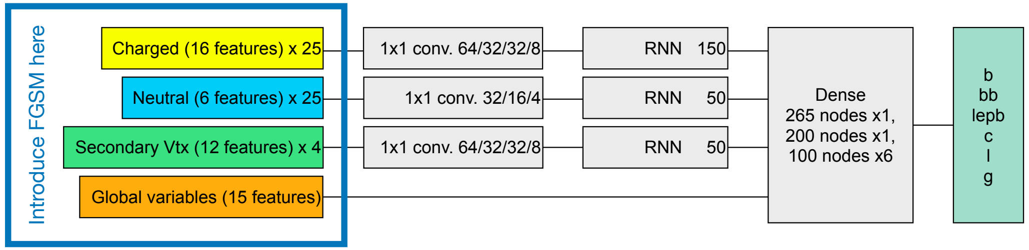
Goal: apply these techniques to DeepJet as well!



Physics Institute III A RWTH AACHEN UNIVERSITY

Application to DeepJet

<u>CMS samples for Training & Evaluation:</u>
see Backup



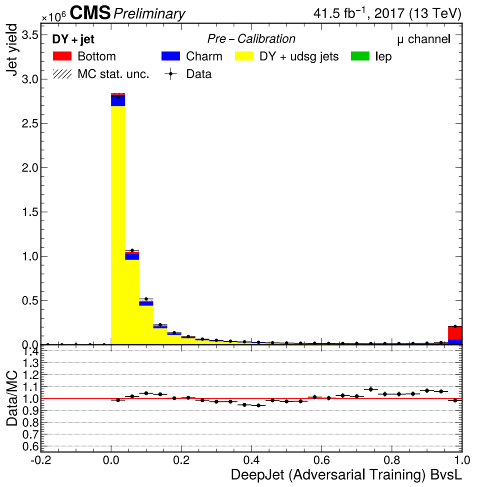
DeepNTuples

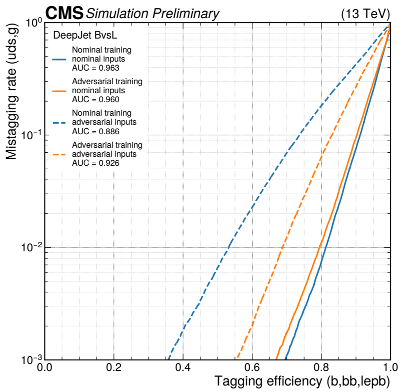
PFNano

Training &
MC performance

DeepJet

Evaluation,
Data/MC & SFs





DeepJetCore

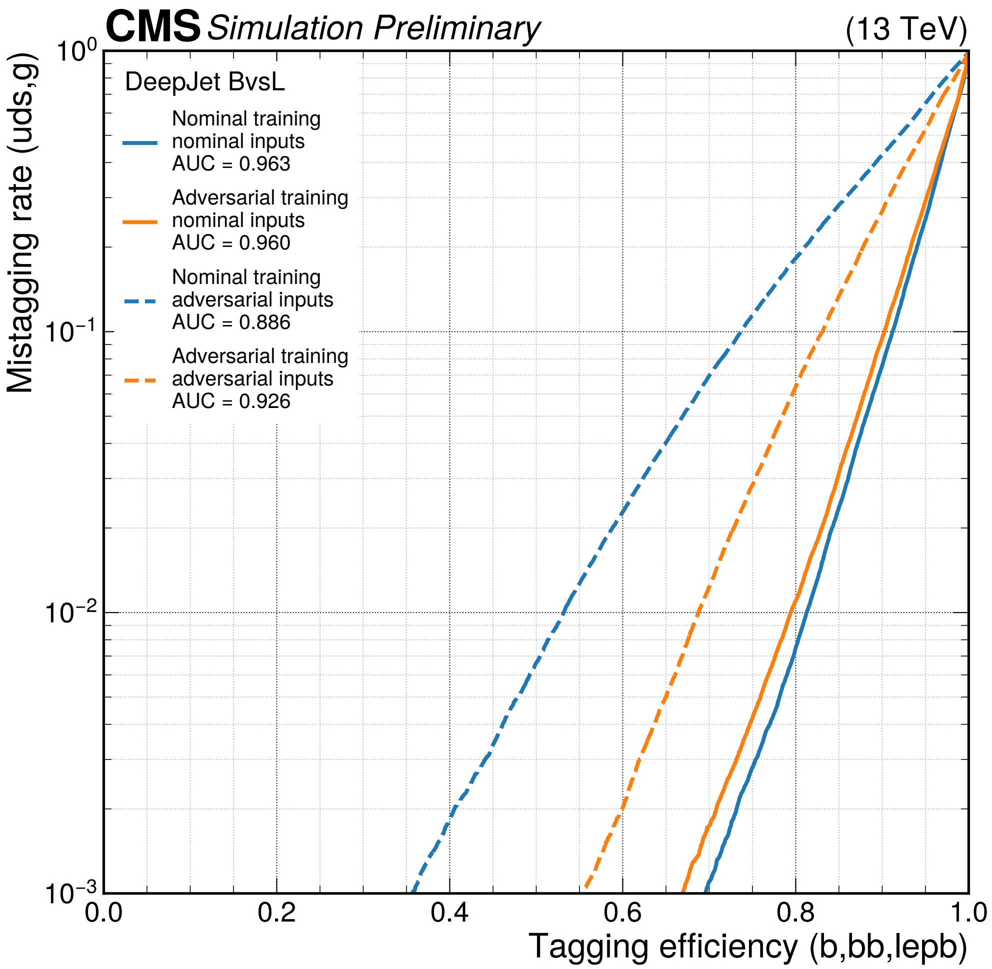
VHcc-
cTagSF

Performance in simulation

- Investigate nominal and adversarial training
  - **Setup** for adversarial training: **hyperparameter** $\epsilon = 0.01$
    - scaled per feature to match scale of each input distribution
    - integers as well as zero-padded elements: not modified at all
  - Adversarial inputs: same hyperparameter
    - and no input is shifted more than 20% of its original value

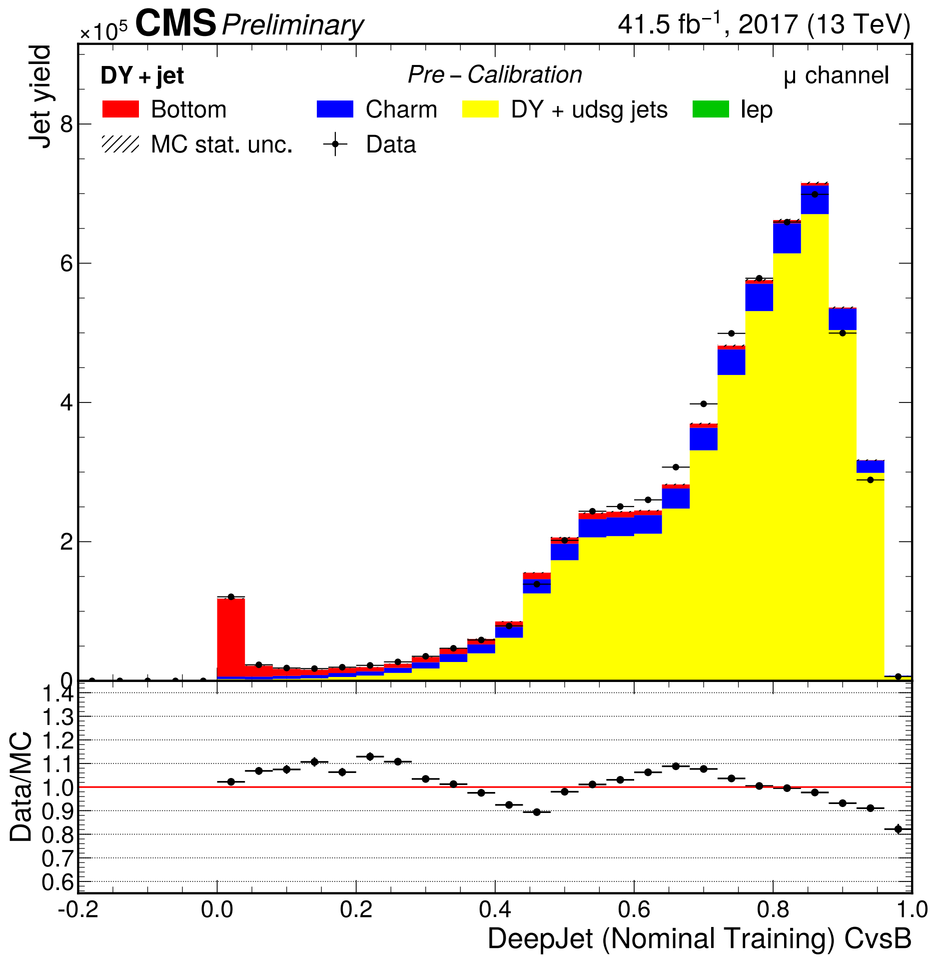
- **Tradeoff** observed: performance $\leftrightarrow$ robustness

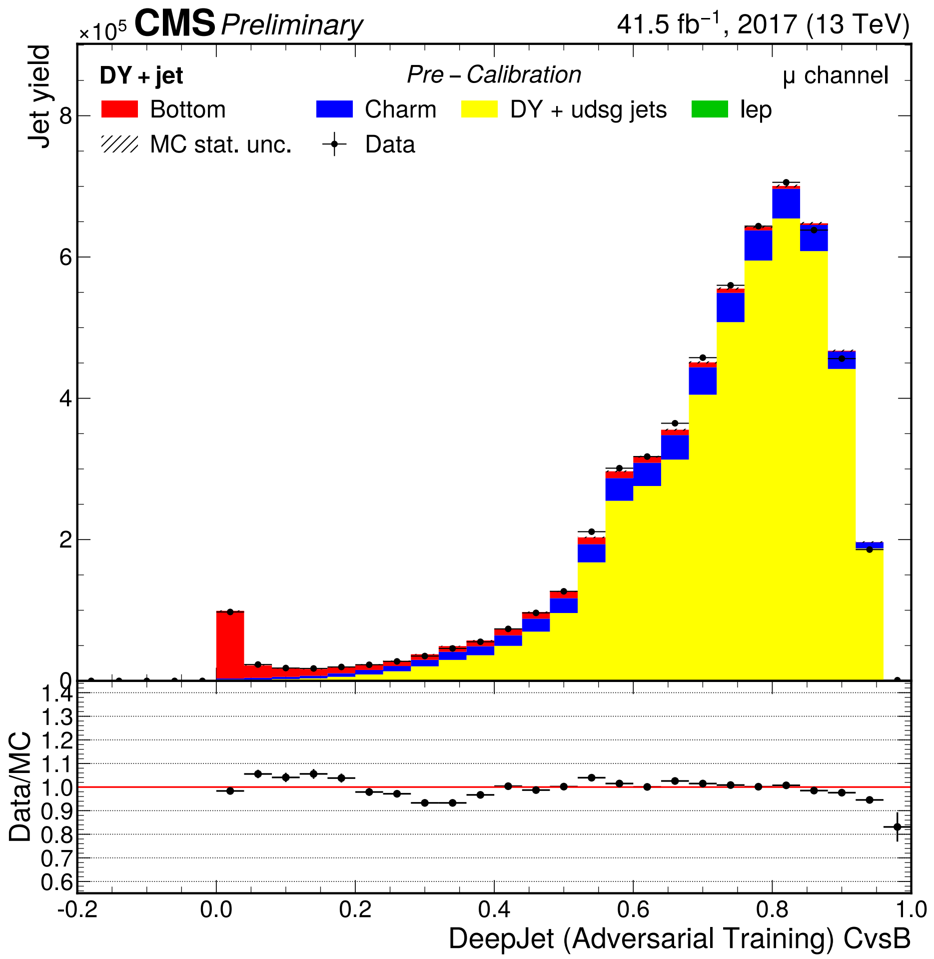
- Compared to previous example, impact of FGSM is more **severe** for DeepJet (several hundred **more input features!**)



($\rightarrow$ Similar results for other discriminators, see backup)

Comparing Data/MC agreement for nominal and adversarial training: light jet selection





➔ Agreement improves! More examples in backup.

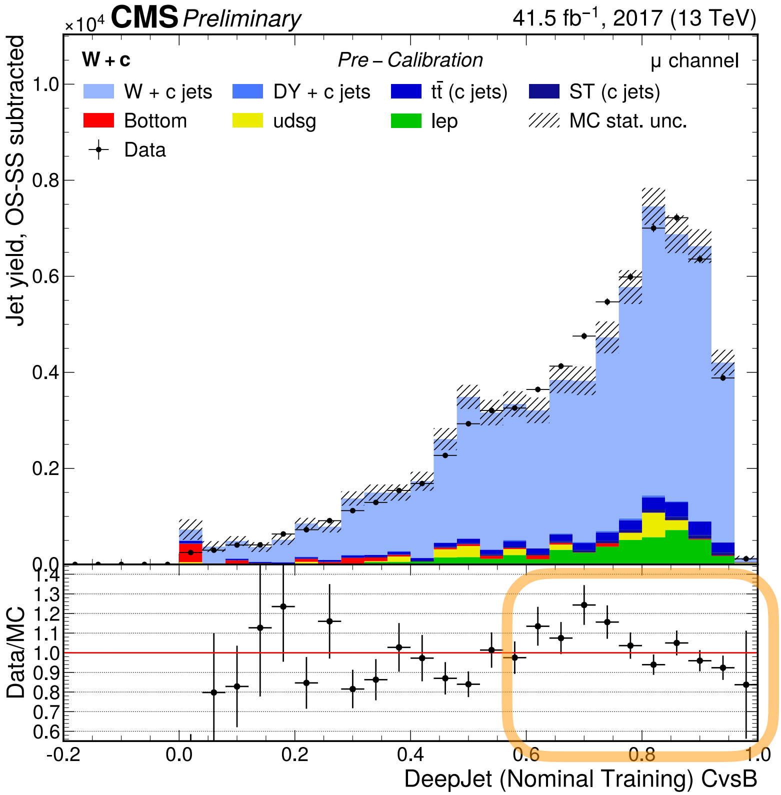
Quantifying Data/MC agreement

- Measure the agreement with the help of the **Jenson-Shannon (JS) divergence**, a „**distance** between two distributions“; here: **data and simulation**
- Compare agreement in three different phase spaces, enriching particular **flavours** (light/c/b)

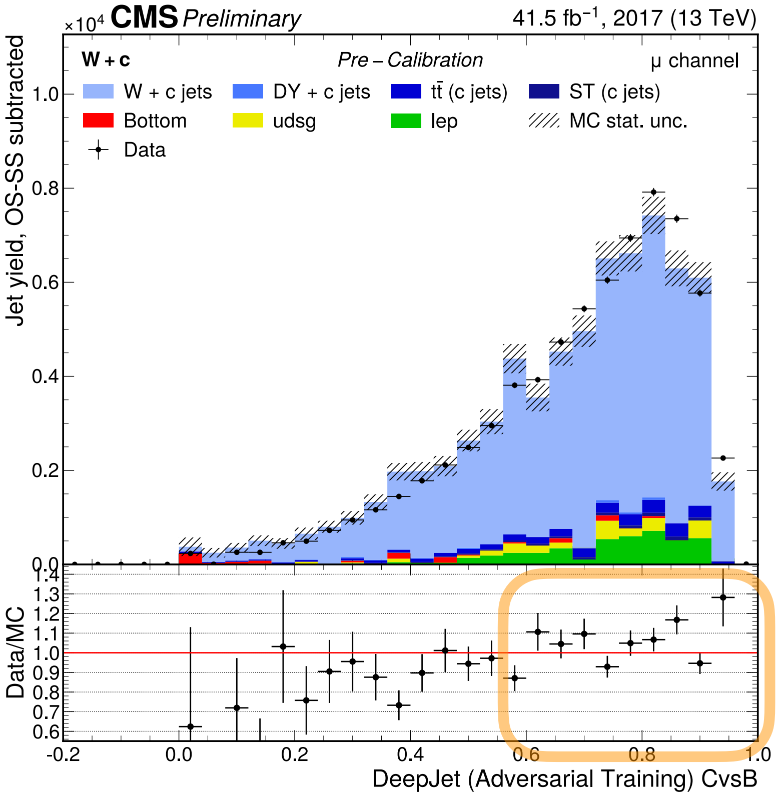
JS divergence (a.u.) DeepJet 2017 (13 TeV)	**udsg jets**			**c jets**			**b jets**		
	BvsL	**CvsB**	**CvsL**	**BvsL**	**CvsB**	**CvsL**	**BvsL**	**CvsB**	**CvsL**
Nominal training	0.000358	0.000353	0.000947	0.002632	**0.002350**	0.002263	0.003506	**0.002528**	0.004820
Adversarial training	**0.000063**	**0.000058**	**0.000466**	**0.001887**	0.003074	**0.001766**	**0.003329**	0.003005	**0.002924**

Comparing Data/MC agreement for nominal and adversarial training: charm jet selection

- **Disentangle** performance and agreement: Look at one of the „bad“ examples, where **JS does not improve** when applying adversarial training
  - Good performance in simulation, bad in data can lead to bad agreement between the two domains
  - … but so can good generalization to data (= the other way around)



Negative slope

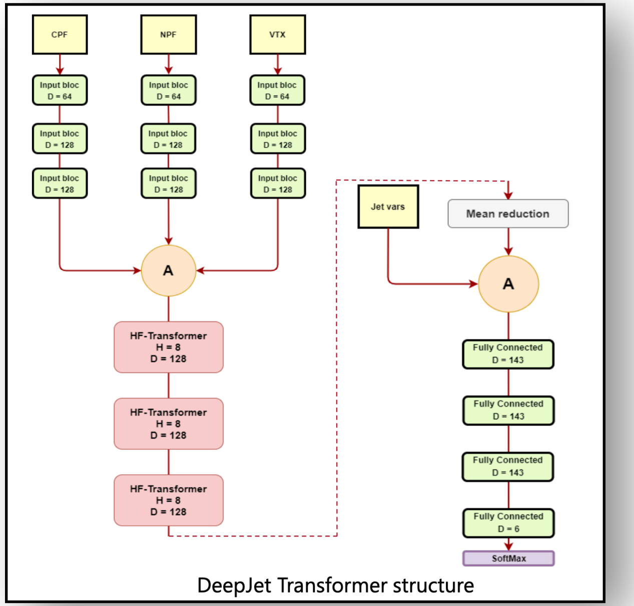


Positive slope

- Introduce a cut and compare efficiencies for data & mc
- adversarial training → actually a *higher* performance to tag charm jets as charm jets in data than doing the same for mc ➟ **generalizes better to data!**

Outlook

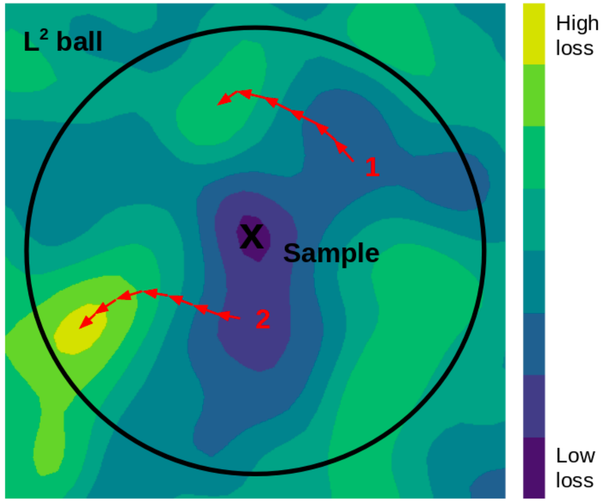
- Many directions to explore:
  - other **taggers** (e.g. ► DeepJet Transformer, GNN)



DeepJet Transformer structure

* by Alexandre de Moor

  - other **attacks and defenses**, or different **strategy** altogether (Domain adaptation? Training on data? PGD?)

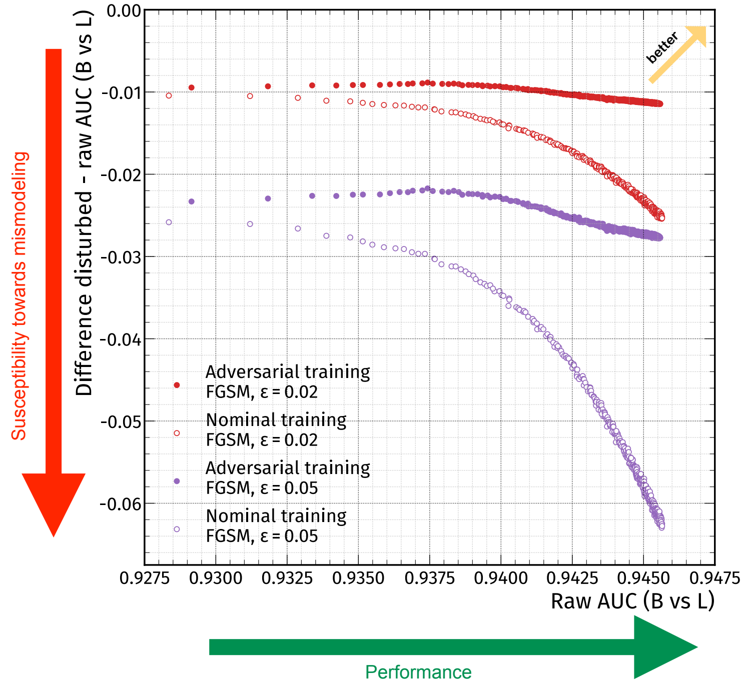


Source: https://towardsdatascience.com/know-your-enemy-7f7c5038bdf3

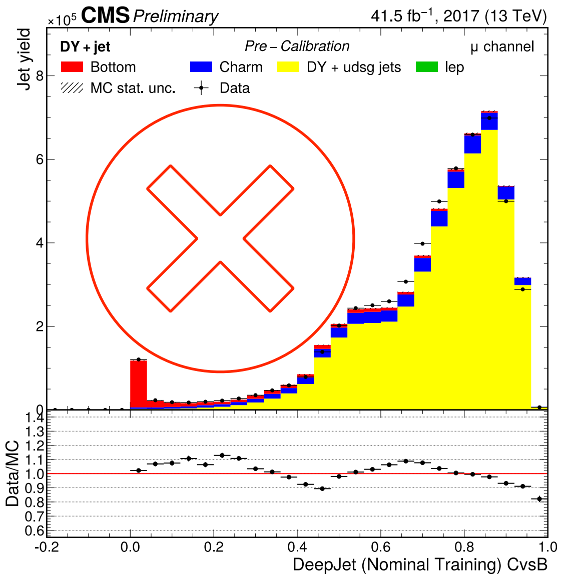
  - impact of **systematic variations**

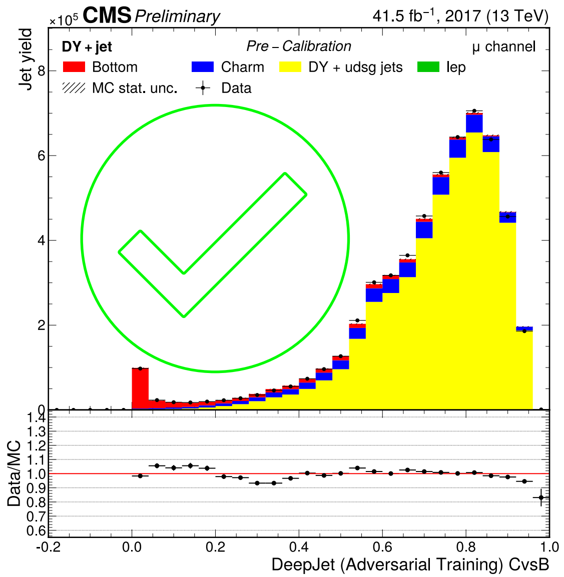
Summary

- Adversarial training has been identified as a method that improves **robustness** of jet tagging algorithms while maintaining high **performance**
- Preliminary studies done with a typical DNN + public dataset and **confirmed** with **CMS dataset** & state-of-the-art algorithm (**DeepJet**)
- Robustness and **data/MC agreement** are closely related



- Adversarial techniques could become an important ingredient of new algorithms to be used for **Run3** and beyond, thus (hopefully!) contributing to smaller uncertainties on scale factors, allowing more **precise** measurements of SM properties





Thank you!

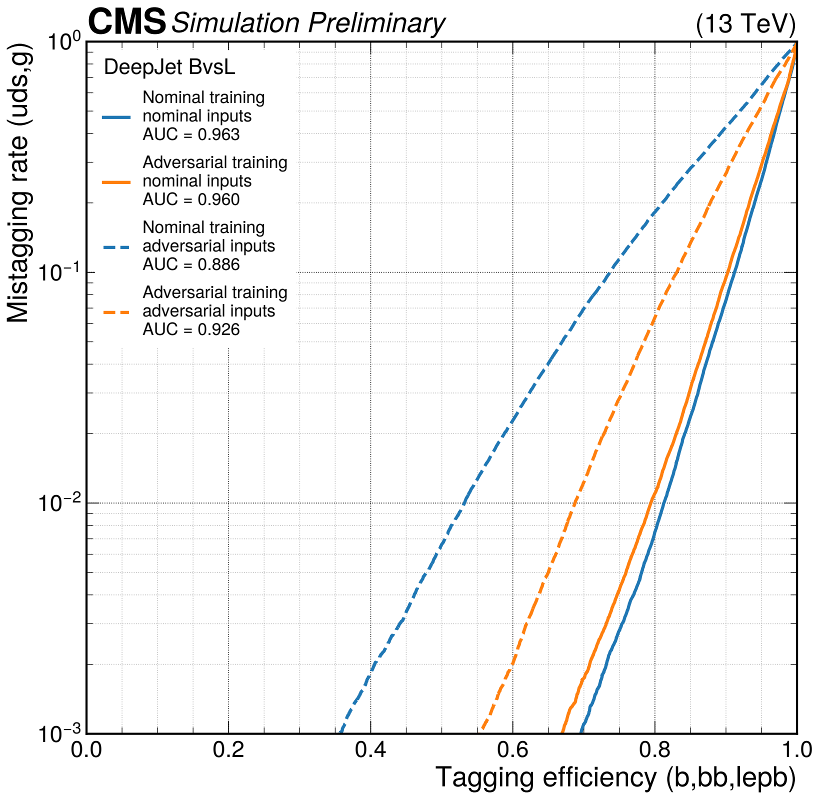
Backup

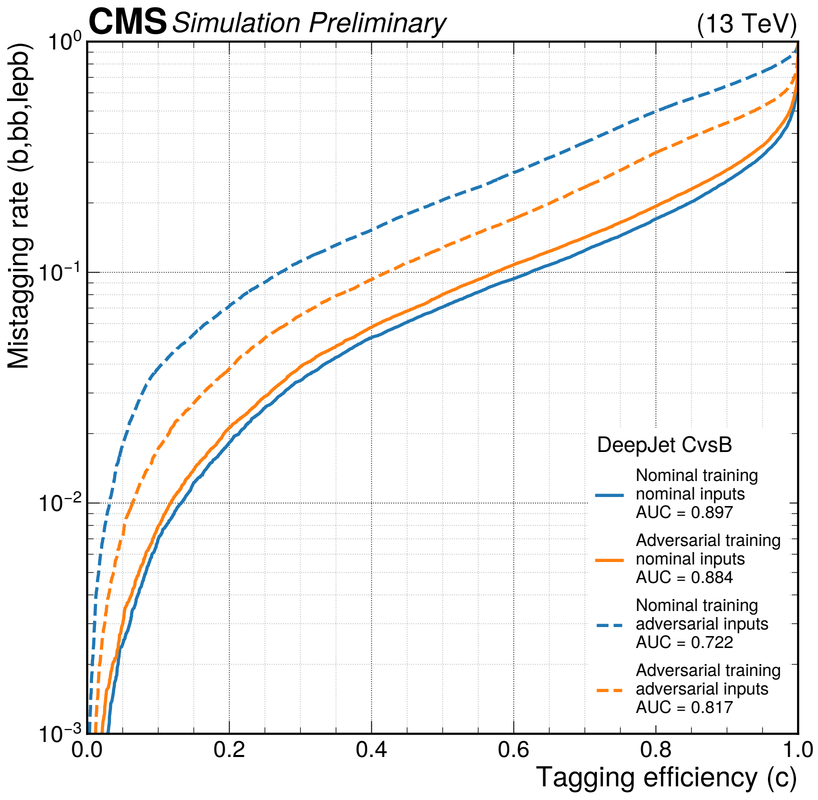
Samples and configuration

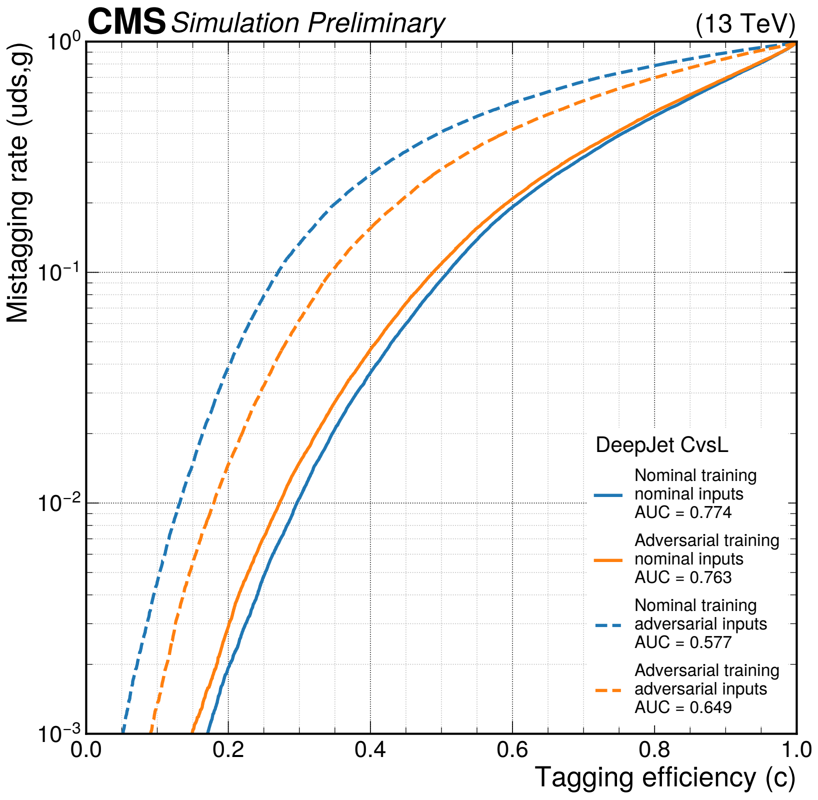
- Training:
  - performed on a mixture of QCD and ttbar MC so that there's enough stat. available for both light and heavy flavors
  - reweighted to reference (p_T, η) distribution of b-jets

- Evaluation:
  - performance in data is evaluated in the single muon and di-muon final states
  - MC: ttbar (dileptonic, semi-leptonic, hadronic), single top, W+jets, inclusive DY+Jets

Performance comparison for three discriminators



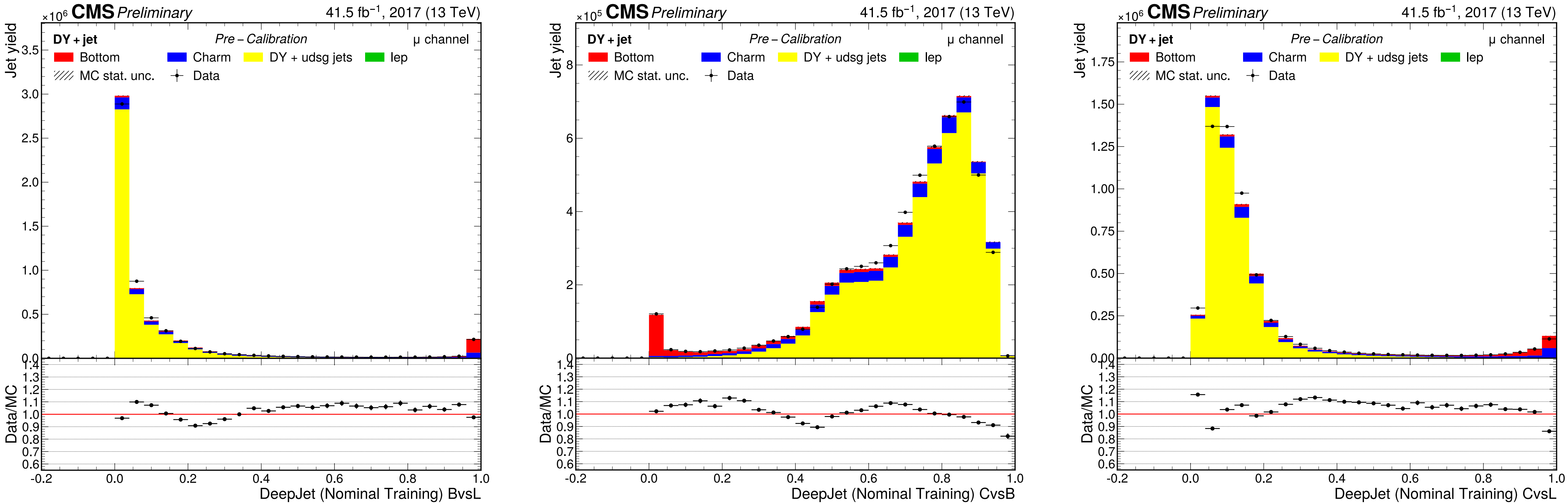




Pre-calibration, nominal training, no FGSM applied anywhere

Nominal training

udsg jets enriched

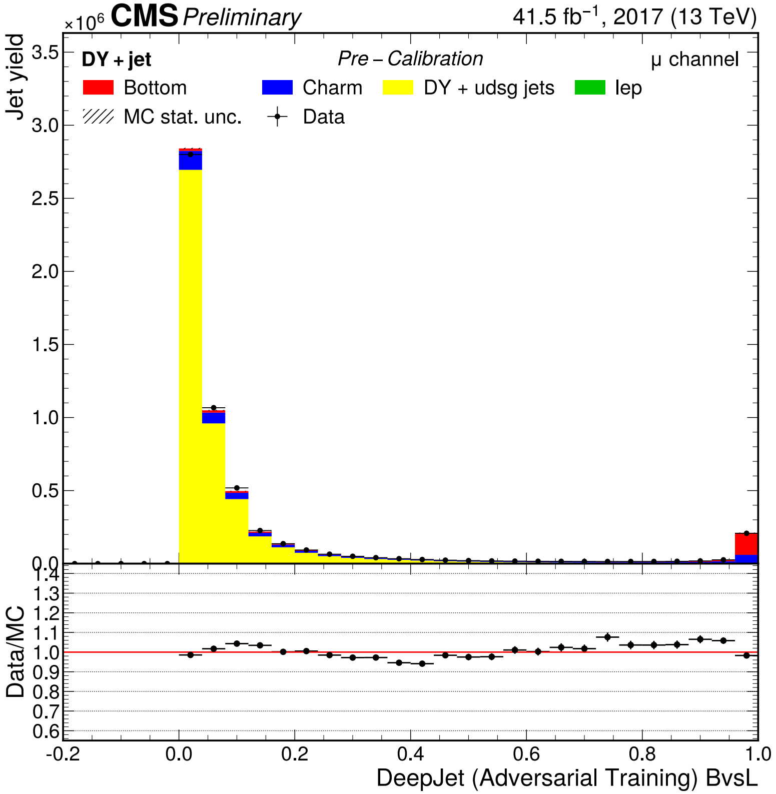


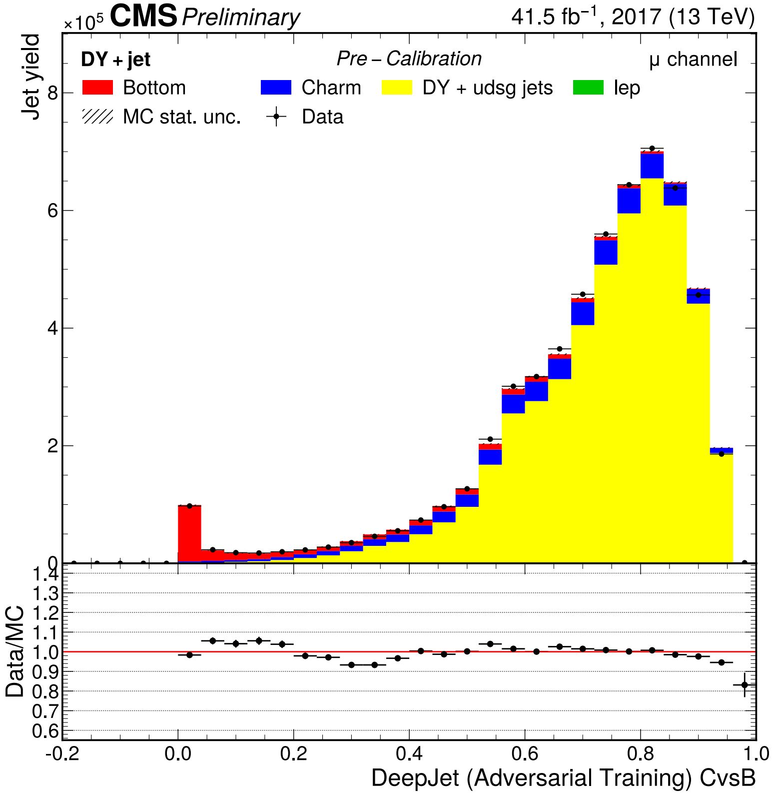
Data/MC agreement of three discriminators BvsL, CvsB, CvsL (left, center, right) for the light flavour-enriched selection, using the nominal model. Events with at least two isolated, oppositely charged muons are selected, additional requirements are placed on the invariant mass window of the reconstructed Z boson, and at least one jet is required that will be used as probe (see Ref. [3] for more details on the selection). Ratios between data and MC show oscillations and ranges with non-zero slope.

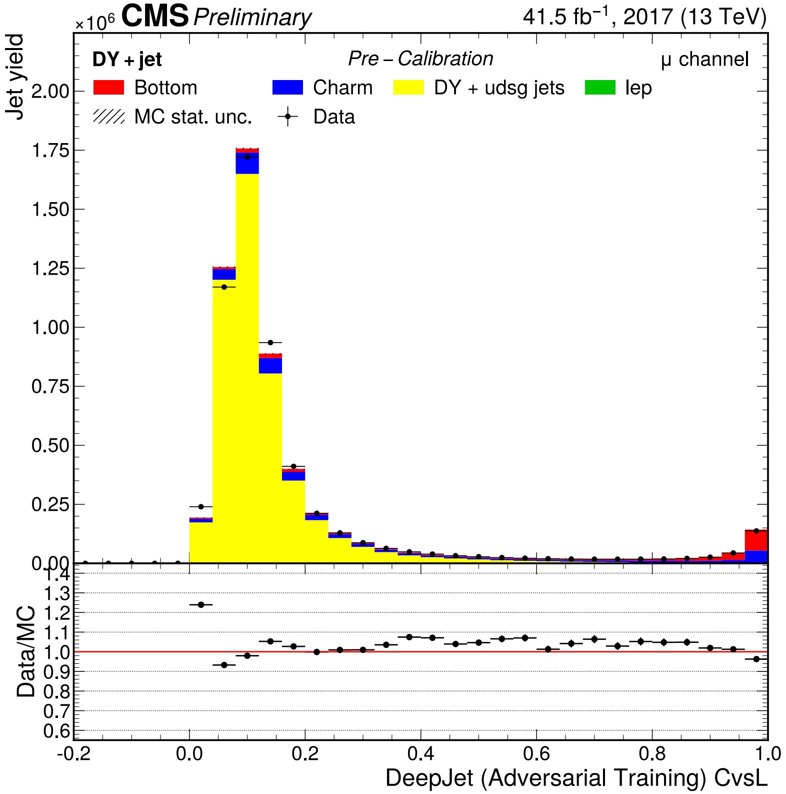
Pre-calibration, adversarial training, but no FGSM applied at evaluation

Adversarial training

udsg jets enriched





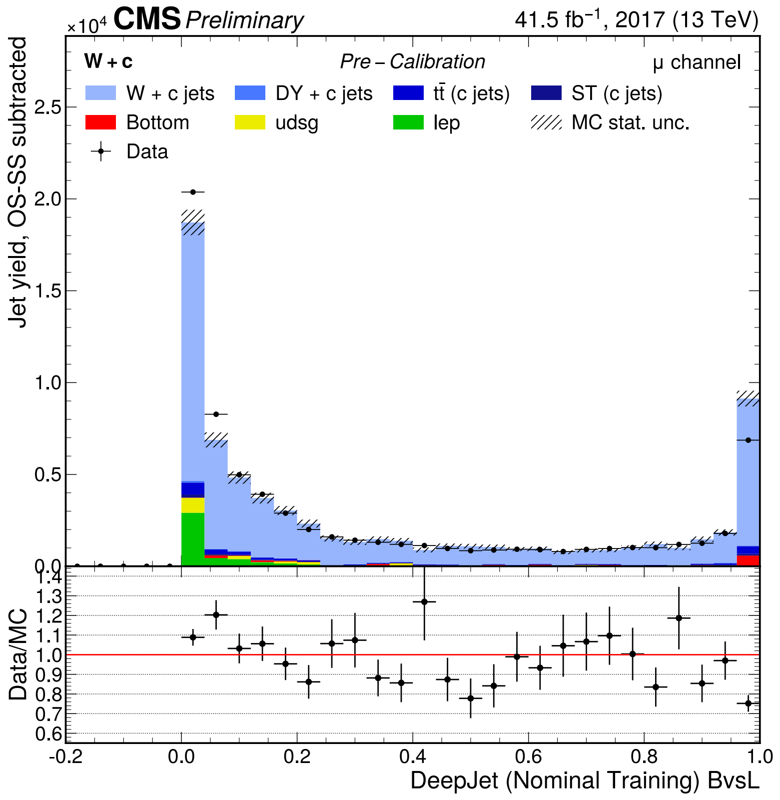


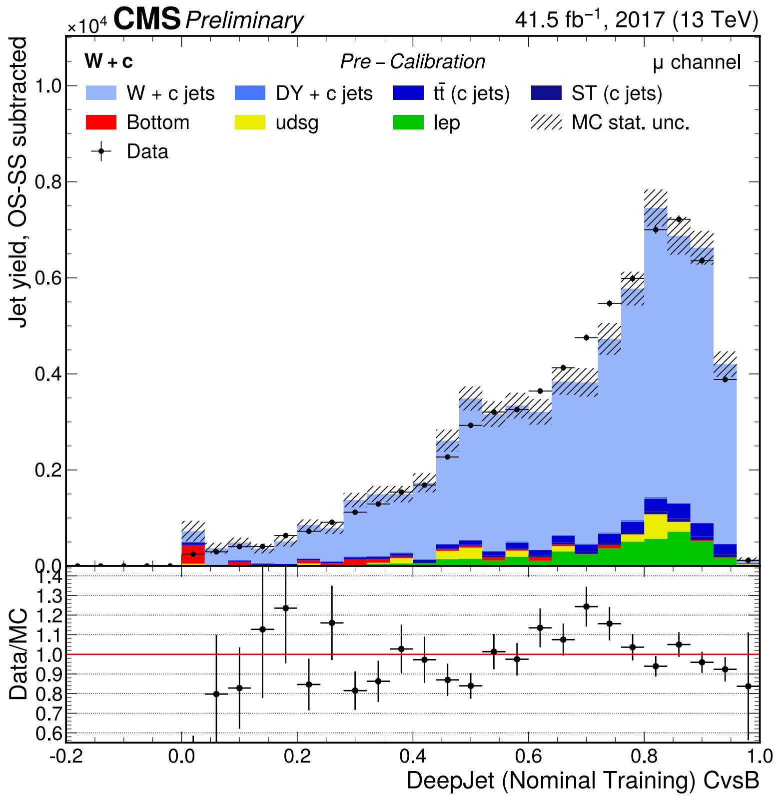
Data/MC agreement of three discriminators BvsL, CvsB, CvsL (left, center, right) for the light flavour-enriched selection, using the adversarial model. Agreement improves compared to the nominal training, with ratios between data and MC moving closer to 1 for all three discriminators. Adversarial training with the chosen hyperparameter of ϵ=0.01 leads to high performance not only for simulated samples, but also for data, and consequently, the resulting discriminator shapes agree well.

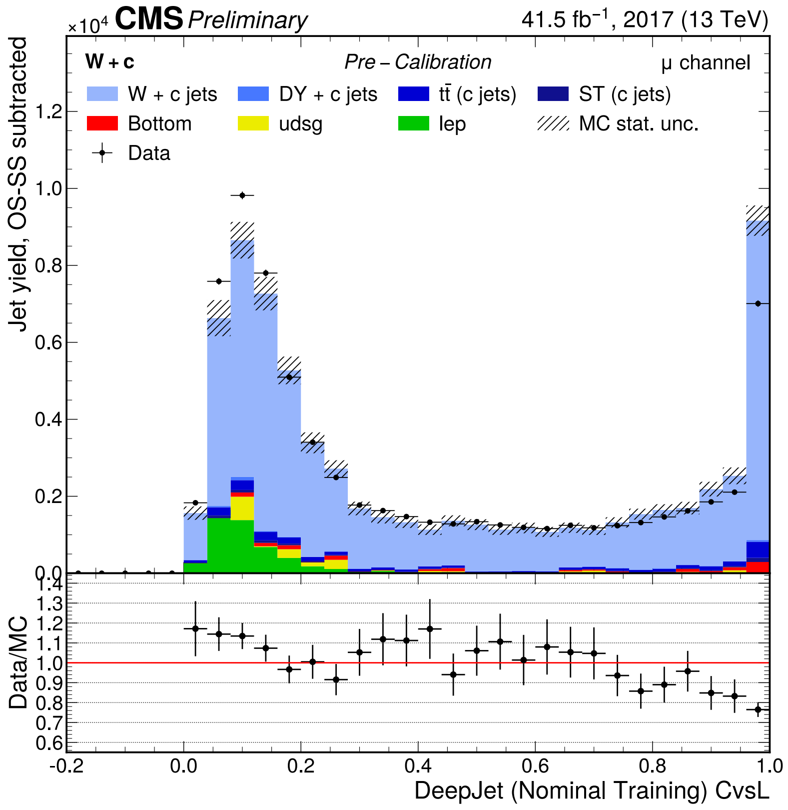
Pre-calibration, nominal training, no FGSM applied anywhere

Nominal training

c jets enriched





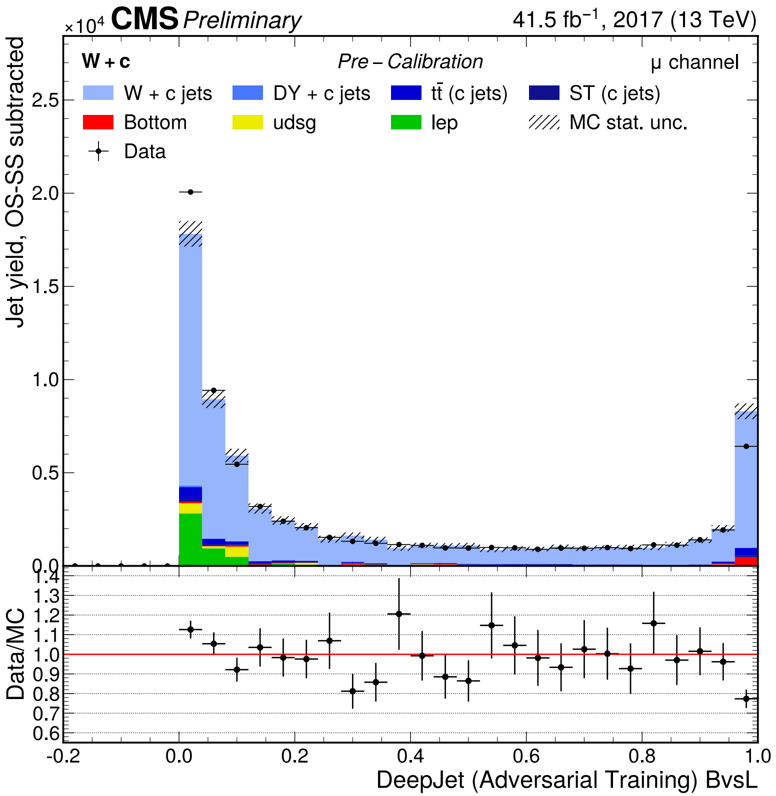


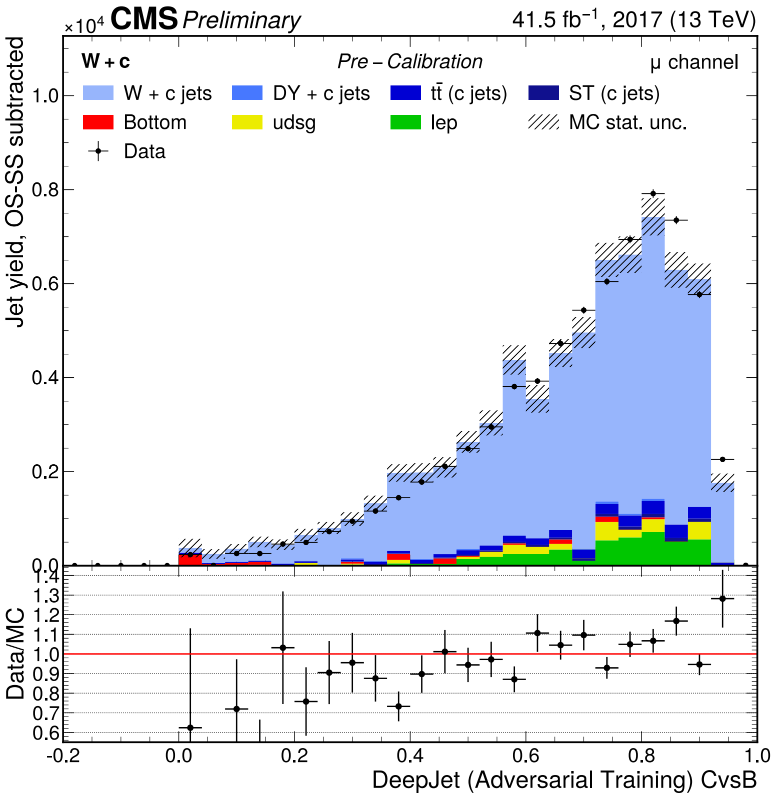
Data/MC agreement of three discriminators BvsL, CvsB, CvsL (left, center, right) for the charm flavour-enriched selection, using the nominal model. Events are selected by identifying an isolated charged lepton from the W boson decay, missing energy, and at least one jet with a soft, non-isolated, muon inside it (see Ref. [3] for more details on the selection). The observed ratios fluctuate, the minimum and maximum being 0.75 and 1.25, respectively, for CsvL, Data/MC ratios show a negative slope. Given that this is the charm jet selection, a negative slope at high values for any discriminator of the form CvsOther indicates a worse performance in data than in simulation which would need to be calibrated subsequently.

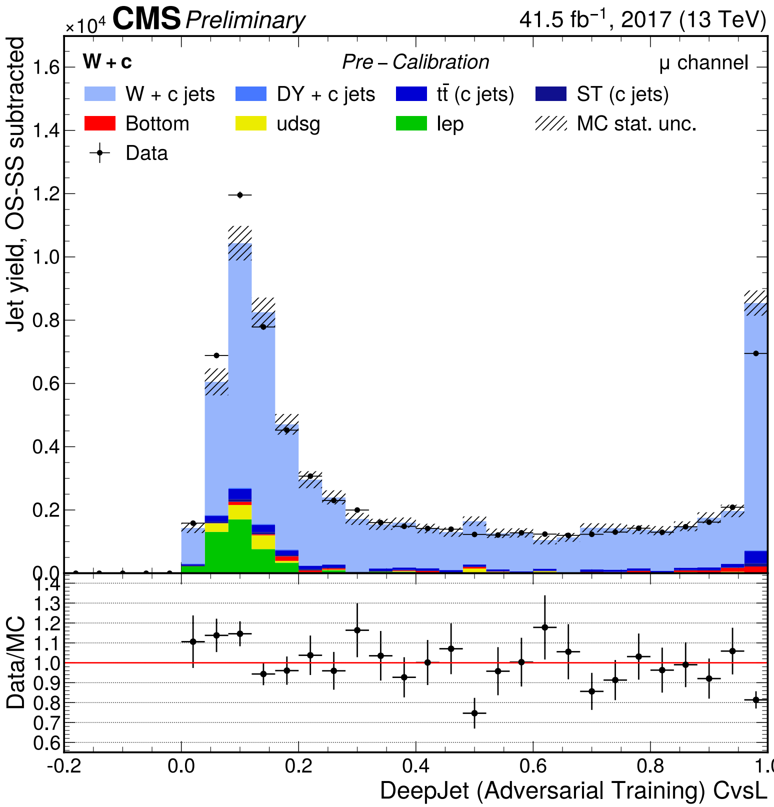
Pre-calibration, adversarial training, but no FGSM applied at evaluation

Adversarial training

c jets enriched





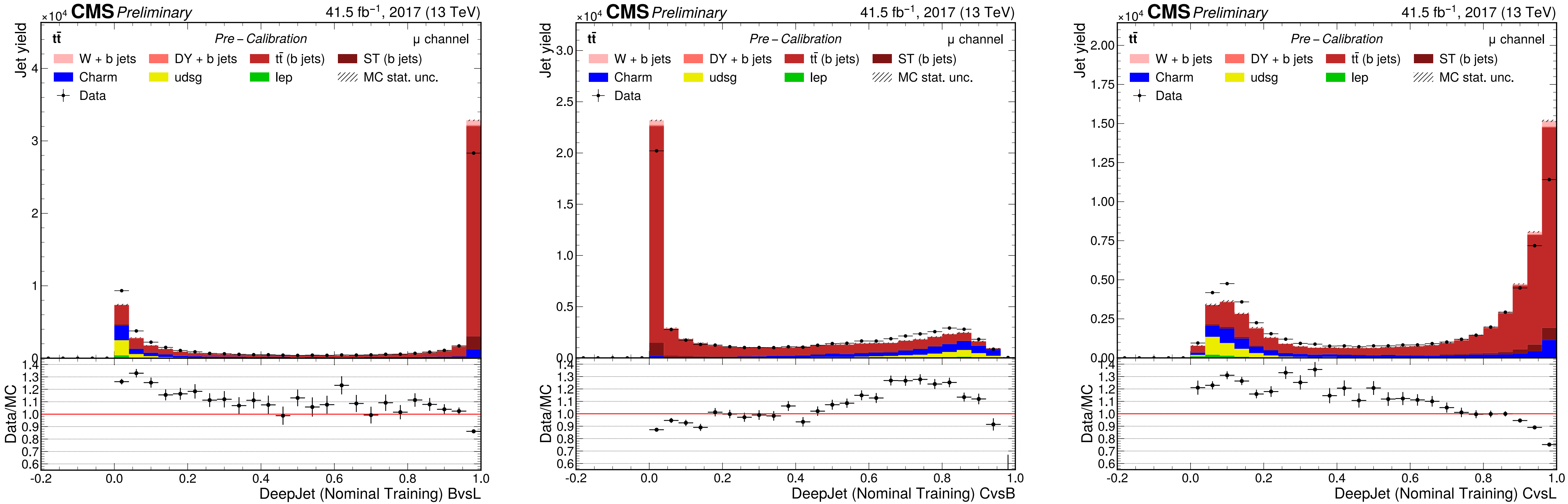


Data/MC agreement of three discriminators BvsL, CvsB, CvsL (left, center, right) for the charm flavour-enriched selection, using the adversarial model. Agreement for the BvsL discriminator is qualitatively similar to nominal training, quantifying slight improvements however is facilitated in a later step using a dedicated metric. For CvsB, more charm jets in data are tagged as charm jets, and therefore, even though the overall agreement between the two domains (later specified with a summary quantity) does not improve for this discriminator, the root cause is now not the worse performance in data, but instead better performance in data than in simulation. A different way to phrase this is to note that adversarial training generalizes better to data than to perform on simulated samples. For CvsL, the situation improves in that sense that instead of a negative slope as seen for nominal training, the ratios now show a comparatively flat behaviour, should one try to fit a straight line. While the current hyperparameter chosen for adversarial training was close to perfect for light jets, this is already at the edge of an over-correction for charm jets.

Pre-calibration, nominal training, no FGSM applied anywhere

Nominal training

b jets enriched

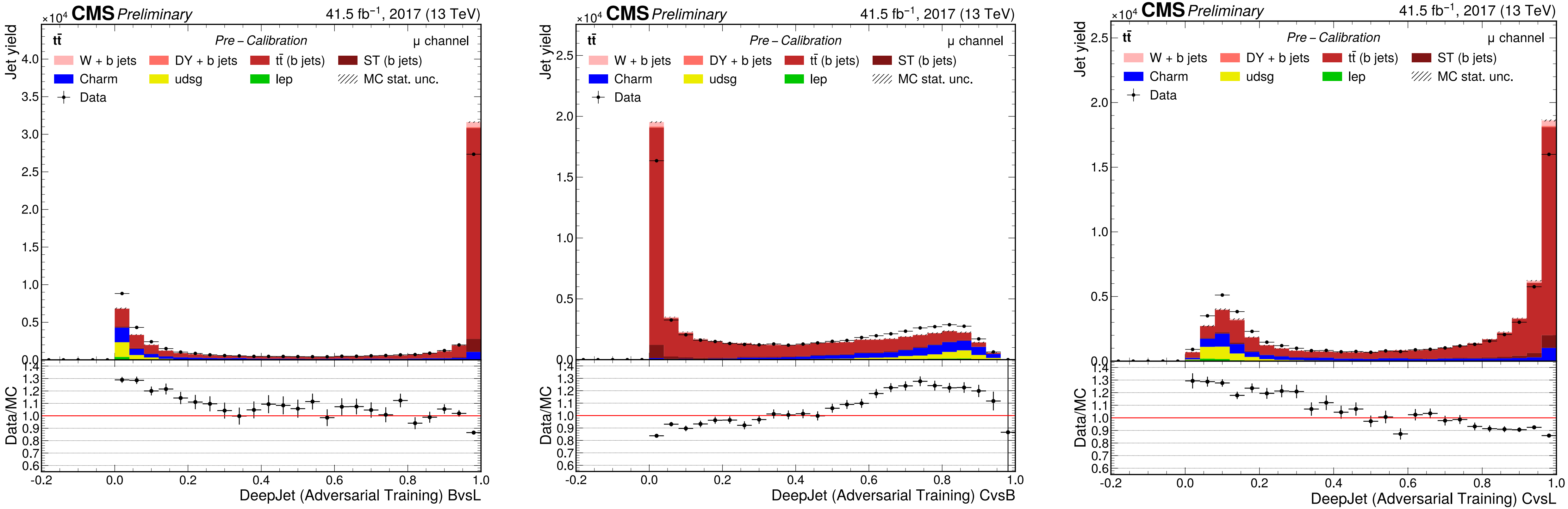


Data/MC agreement of three discriminators BvsL, CvsB, CvsL (left, center, right) for the bottom flavour-enriched selection, using the nominal model. Events are selected by identifying a leptonically decaying W boson, a jet with a soft-muon inside, as well as additional jets (see Ref. [3] for more details on the selection). Ratios between data and MC show oscillations and ranges with non-zero slope.

Pre-calibration, adversarial training, but no FGSM applied at evaluation

Adversarial training

b jets enriched



Data/MC agreement of three discriminators BvsL, CvsB, CvsL (left, center, right) for the bottom flavour-enriched selection, using the adversarial model. Agreement improves slightly compared to nominal training for BvsL and CvsL, but the slopes still exist. Just like for charm jets, a hyperparameter scan may yield better agreement than choosing the same parameter as for light jets, which already improves the situation for two out of three cases.

Interpretation and next steps

☝ Currently, for simplicity we assume that all flavors are „mismodeled“ identically and use the same **hyperparameter** ϵ across flavours

- already **~perfect for udsg**, but for the ***other*** two selections we can ***do better***
- more **commissioning** results will help determining the necessary severity of the attack and deliver better understanding of systematic uncertainties

~~One size fits all?~~

Performance ϵ **Robustness / Generalization**

➔ optimize per flavour

Performance ϵ_{udsg} **Robustness / Generalization udsg jets**

Performance $\epsilon_{\mathrm{charm}}$ **Robustness / Generalization charm jets**

Performance $\epsilon_{\mathrm{bottom}}$ **Robustness / Generalization bottom jets**

⚠ Adversarial attack can **not fully capture the data/MC mismodeling**

- no one tells us the mismodeling is just a linear shift of inputs along the steepest gradient, this is totally **arbitrary** and unlikely
- and yet, in seven out of nine cases, there is a significant improvement for the agreement between the two domains, data and MC, measured with JS divergence

Jenson-Shannon divergence measure

Def.: *Jenson-Shannon divergence* between two distributions P and Q

$$\text{JSD}(P||Q) = \frac{1}{2}D(P||M) + \frac{1}{2}D(Q||M)$$

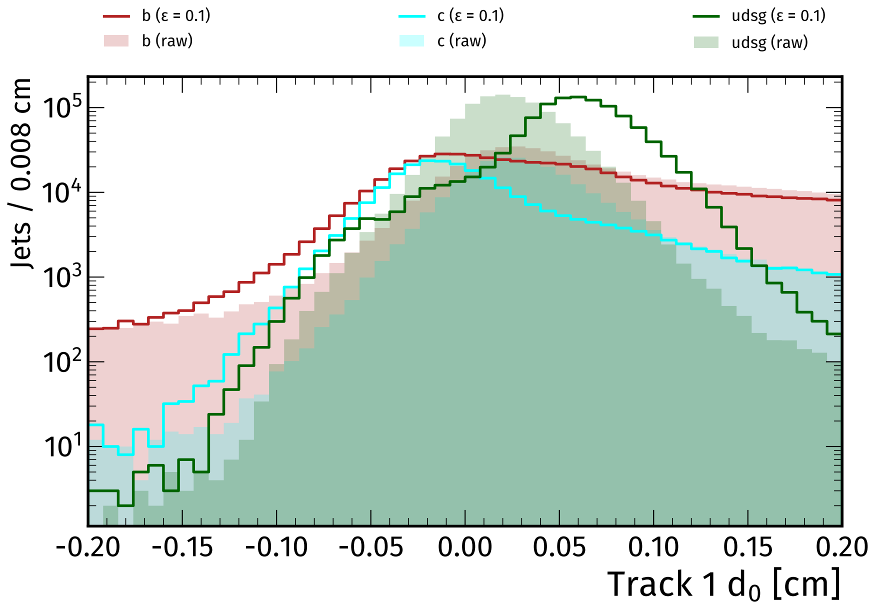
where $M = \frac{1}{2}(P+Q)$ and

$$D(X||Y) = \sum_{\text{all bins } k_i} X(k_i) \log \frac{X(k_i)}{Y(k_i)} \text{ (Kullback-Leibler divergence)}$$

- **Lower** is better (0 perfect)
- **Symmetrized** KL divergence
- Exclude 0s and negative values (OS-SS subtraction)

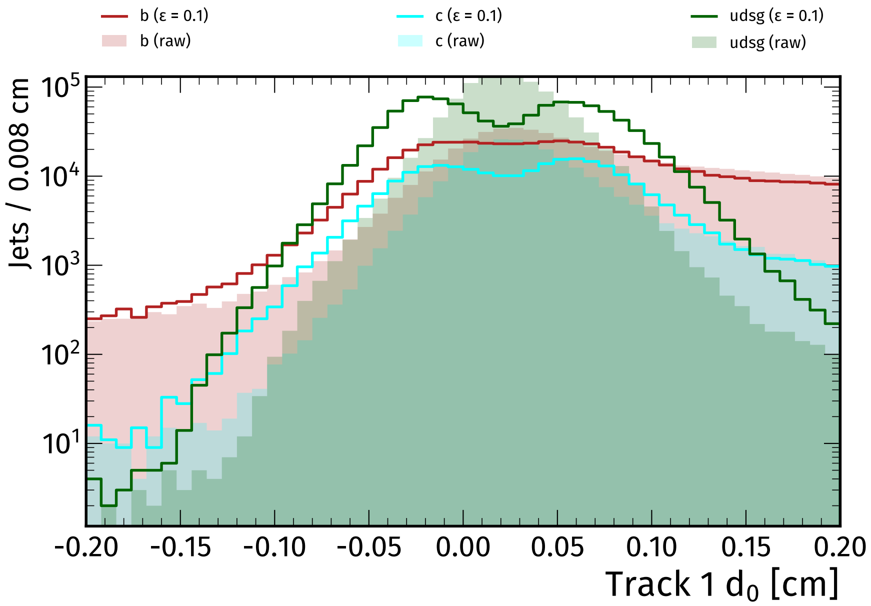
Why do the nominal and adversarial model react differently? — Inputs

Nominal training ⊗ FGSM ➔ asymmetric shapes



- Shifts light jets into heavy-flavor dominated region and vice-versa ➔ FGSM „*inverts*“ physics

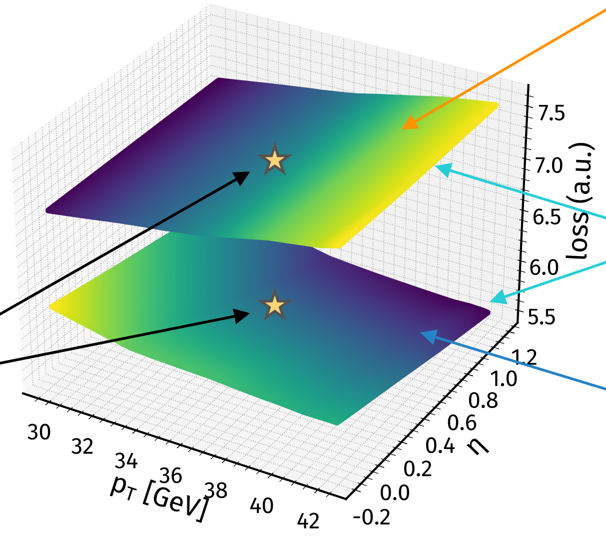
Adversarial training ⊗ FGSM ➔ symmetric shapes

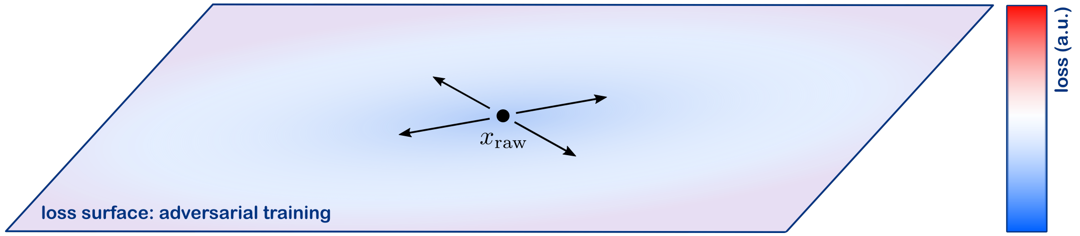


- Crafting adversarial inputs for adversarially trained model is almost like „*coin-flipping*“

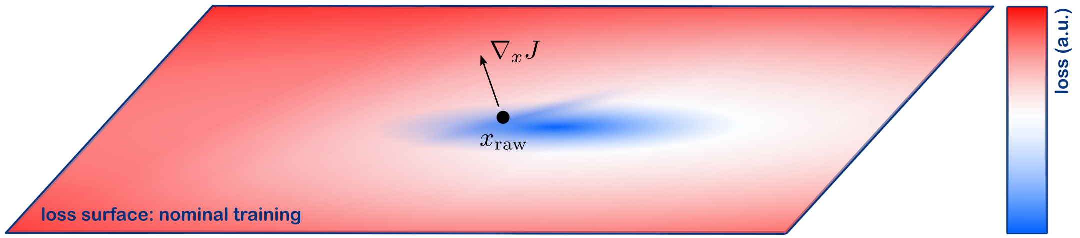
Why do the nominal and adversarial model react differently? — Loss

Conjecture: the **loss surfaces** are **different**!





- **Flat** ⇸ no preferred direction



- Clearly **preferred** direction for first-order adversarial attacks

„Improving robustness of jet tagging algorithms with adversarial training"
A. Stein, S. Mondal, ACAT 2022, Poster presentation (indico)