Unbinned MVA techniques for EFT analyses

Alfredo Glioti

Institut de Physique Théorique

LHC EFT Working group meeting
24/10/2023

Chen, AG, Panico, Wulzer - **2007.10356**
Chen, AG, Panico, Wulzer - **2308.05704**

Motivations

Huge variety of possible measurements at the LHC
(high PT probes, Higgs couplings and distributions, …)

Huge variety of putative New Physics effects in model-independent (**EFT**) approach

⇩

Effective and **systematic** data analysis techniques needed to **maximize sensitivity**

Motivations

We can parametrize New Physics using the SMEFT Lagrangian

Theory

$$\mathcal{L}_{SMEFT} = \mathcal{L}_{SM} + \theta_i O_i^{d=6} + \dots$$

Wilson coefficients

More common approach

Predictions

$$\sigma(\theta), \quad p(x|\theta) = \frac{1}{\sigma(\theta)} \frac{d\sigma(\theta)}{dx}$$

Measurable observables

Simplified predictions

$$\sigma_i(\theta), \qquad i = 1, \dots N_{bins}$$

Often loses information

Extracting the **full information** would require the likelihood p(x|θ) (as function of both x and θ)

How to access $p(x|\theta)$?

Monte Carlo generators work in the “forward mode”:

1) sample unobservable “partonic” variables z_{part} from a known $p(z_{part}|\theta)$
2) Transforms z_{part} to x, event by event

Unknown distribution of observables x

$$p(x|\theta) \approx \int dz_{part} p(x|z_{part}) p(z_{part}|\theta)$$

z_{part} is normally very far from x (e.g. invisible particles, NLO effects…)

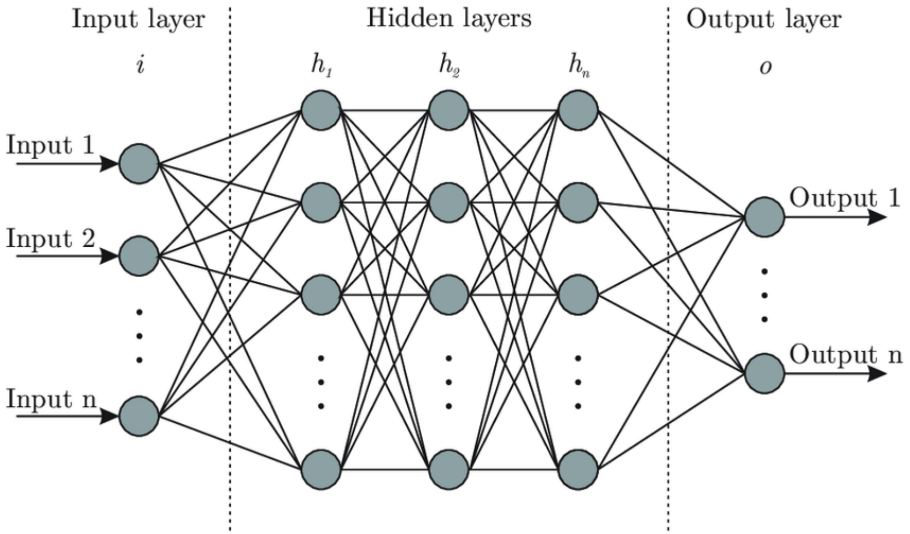
$$x \neq z_{part}$$

Even worse if we include showering, hadronization, detector…

p(x|θ) through Machine Learning

Brehmer & al. 1805.00013

Basic idea: approximate $p(x|\theta)$ with **Neural Networks**: $p(x|\theta) \leftrightarrow nn(x;w)$



The result will be **fully differential** on **all observables**, quick to evaluate and it can be obtained with a relatively small amount of Monte Carlo points.

No transfer functions modeling required.

Universal and systematically improvable

Simple Classifier

Consider **fixed** $\theta = \bar{\theta}$. We can **learn** $p(x|\bar{\theta})$ with respect to the SM ($\theta = 0$).

Training sample $\mathcal{T} = \{(x_i \sim p(x|0), y_i = 0), (x_i \sim p(x|\bar{\theta}), y_i = 1)\}$

$$\boxed{\ell[nn(\cdot)] = \frac{1}{N_T} \sum (nn(x_i; w) - y_i)^2}$$

Loss function.
Minimized by training (wrt w)

Infinite training sample limit

$$\ell \to \int dx \, \left[p(x|0)(nn(x) - 0)^2 + p(x|\bar{\theta})(nn(x) - 1)^2\right]$$

$$\boxed{\frac{\delta \ell}{\delta nn} = 0 \implies nn(x) = \frac{p(x|\bar{\theta})}{p(x|0) + p(x|\bar{\theta})} \implies \tau(x; \bar{\theta}) \equiv \frac{p(x|\bar{\theta})}{p(x|0)} = \frac{nn(x)}{1 - nn(x)}}$$

Reweighted data

Chen, AG, Panico, Wulzer - 2308.05704

The Simple Classifier **loss** can be generalized as a **weighted sum**

$$\ell[f(\cdot)] = \sum_{\mathrm{e}\in\mathrm{S}_0} w_{\mathrm{e}}(\bar{c})\,[f(x_{\mathrm{e}})]^2 + \sum_{\mathrm{e}\in\mathrm{S}_1} w_{\mathrm{e}}(0)\,[f(x_{\mathrm{e}})-1]^2$$

Sum on different samples → **Simple** Classifier

Sum on same sample → **Reweighted** Classifier

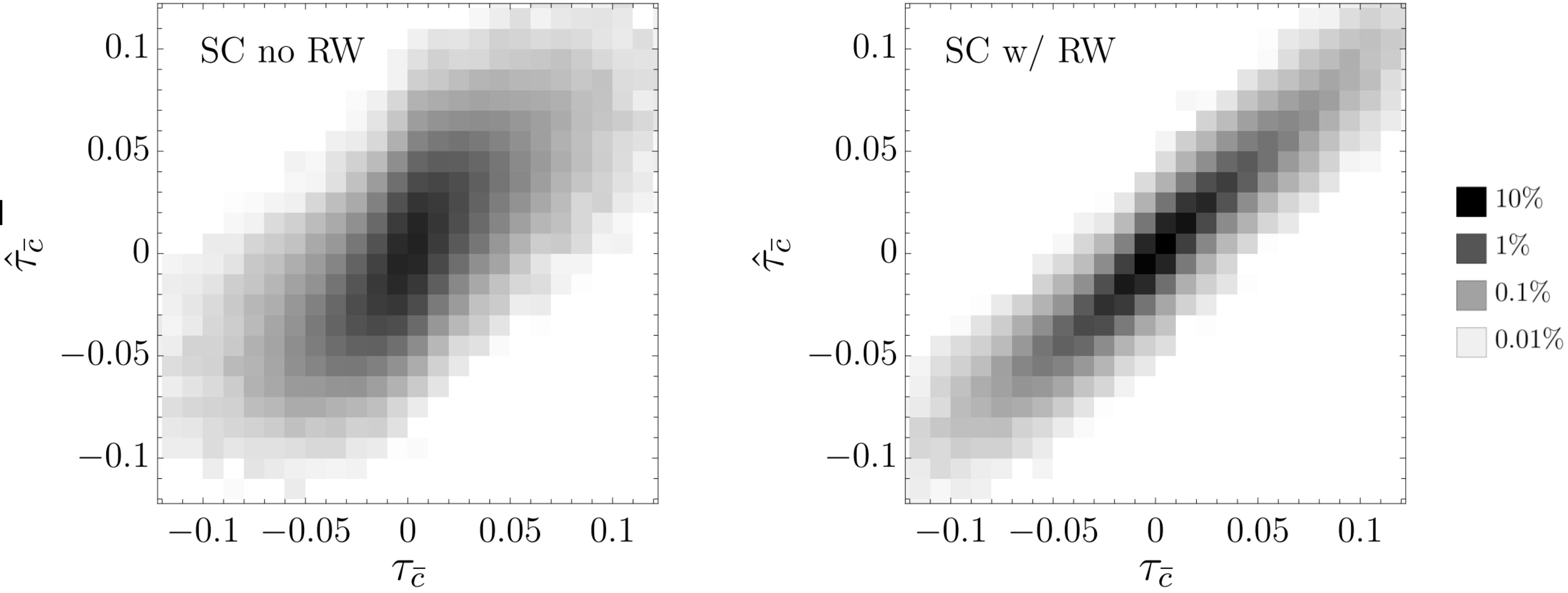
The reason why reweighting helps can be understood by writing $f(x) = 1/2 + \delta f(x)$

No RW ⟶ $\ell[f(\cdot)] = \sum_{\mathrm{e}\in\mathrm{S}_0} w_{\mathrm{e}}(\bar{c})\delta f(x_{\mathrm{e}}) - \sum_{\mathrm{e}\in\mathrm{S}_1} w_{\mathrm{e}}(0)\delta f(x_{\mathrm{e}}) + \sum_{\mathrm{e}\in\mathrm{S}_0} w_{\mathrm{e}}(\bar{c})\delta f(x_{\mathrm{e}})^2 + \sum_{\mathrm{e}\in\mathrm{S}_1} w_{\mathrm{e}}(0)\delta f(x_{\mathrm{e}})^2$

With RW ⟶ $\ell[f(\cdot)] = \sum_{\mathrm{e}\in\mathrm{S}} [w_{\mathrm{e}}(\bar{c}) - w_{\mathrm{e}}(0)]\,\delta f(x_{\mathrm{e}}) + \sum_{\mathrm{e}\in\mathrm{S}} [w_{\mathrm{e}}(\bar{c}) + w_{\mathrm{e}}(0)]\,\delta f(x_{\mathrm{e}})^2$

Reweighted data

Chen, AG, Panico, Wulzer - 2308.05704



Reconstructed likelihood ratio

True likelihood ratio

Parametrized classifier

Chen, AG, Panico, Wulzer - 2007.10356

The **quadratic dependence** on the **Wilson Coefficients** can be learned in training by using a **parametrized** likelihood ratio

$$\ell[\gamma(\cdot)] = \sum_{\mathrm{e}\in\mathrm{S}} \sum_{\bar{c}\in\mathcal{C}} \left\{ w_{\mathrm{e}}(\bar{c}) \left[f(\gamma(x_{\mathrm{e}});\bar{c}) \right]^2 + w_{\mathrm{e}}(0) \left[f(\gamma(x_{\mathrm{e}});\bar{c}) - 1 \right]^2 \right\}$$

$$f(\gamma(x);c) = \frac{1}{1 + \mathcal{P}\left(\gamma(x);c\right)}$$

This will converge to the likelihood ratio during training

Most **generic positive quadratic polynomial** for d Wilson Coefficients

$$\mathcal{P}\left(\lambda(x);c\right) = \sum_{I=1}^{d+1} \left[\sum_{J=1}^{d+1} \lambda_{IJ}(x) c_{J-1} \right]^2$$

λ upper triangular matrix

$$\lambda_{11} = 1$$

While the other components are Neural Networks

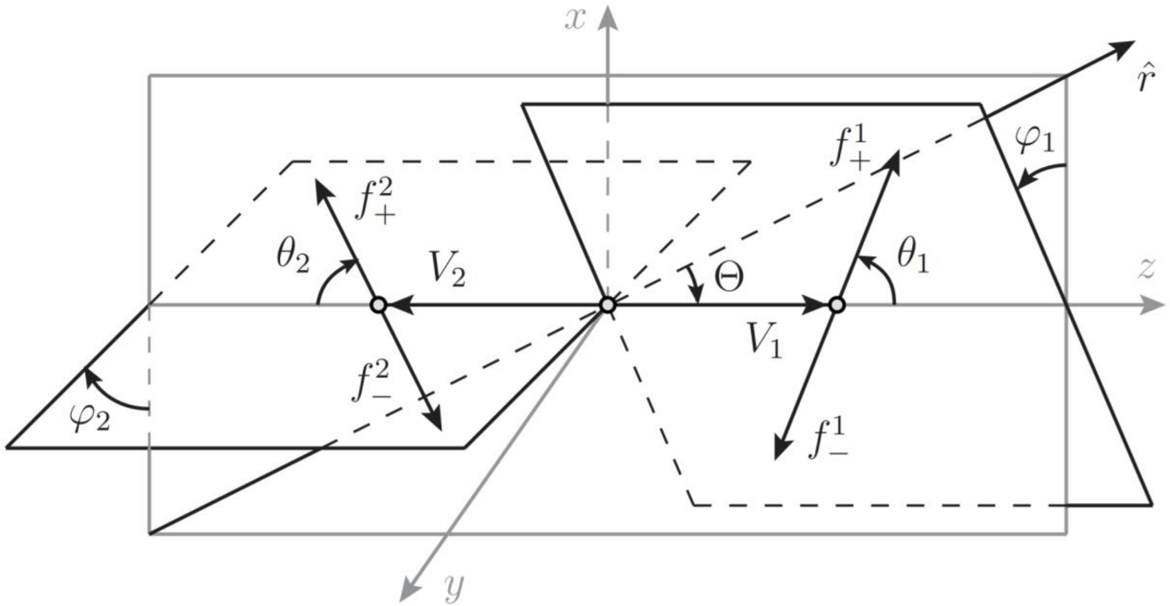
c Wilson Coefficients

$$c_0 = 1$$

Application: WZ production

Franceschini & al. 1708.07823
Panico & al. 1712.01310

$$p\,p \to W^{\pm}\,Z \to (l^{\pm}\,\nu)\,(l^{+}\,l^{-})$$



BSM contribution growing with collision energy from **two operators**

$$\mathcal{O}_{\varphi q}^{(3)} = G_{\varphi q}^{(3)}\left(\overline{Q}_L\sigma^a\gamma^\mu Q_L\right)\left(iH^\dagger \overleftrightarrow{D}_\mu^a H\right)$$

$$\mathcal{O}_W = G_W\varepsilon_{abc}W_\mu^{a\,\nu}W_\nu^{b\,\rho}W_\rho^{c\,\mu}$$

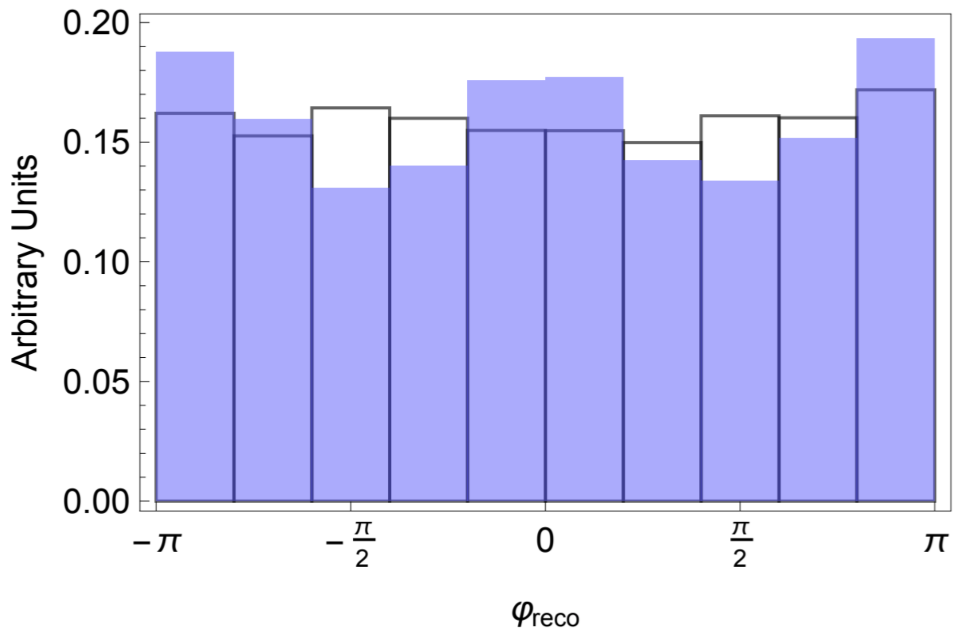
Six independent and discriminating variables: ŝ + 5 angles

Why measure the decay angles?

BSM and SM contribute to **different helicities** of W and Z

$$SM \to 0/0,\ \pm 1/\mp 1$$
$$\mathcal{O}_{\varphi q}^{(3)} \to 0/0$$
$$\mathcal{O}_W \to \pm 1/\pm 1$$

Integrating over the decay angles makes us **lose** this **discriminating information**.
For **Ow** integrating kills **interference with the SM**



Toy case: implementation

To check performances, we implemented a simple Monte Carlo for this process for which we know the **true likelihood analytically**

$$\{Q,\, \underbrace{\chi}_{\text{Helicity of Z decay products}},\, \underbrace{\bar{s},\, \bar{\Theta},\, \bar{\theta}_W,\, \bar{\varphi}_W,\, \bar{\theta}_Z,\, \bar{\varphi}_Z\,,\, y}_{\text{Variables if the neutrino was measured}}\}$$

Kinematics in the **latent space**

The observables given to the networks are

$$\left\{\log[s/\mathrm{GeV}^2],\, \Theta,\, \theta_Z,\, \theta_W,\, \log[p_T/\mathrm{GeV}],\, Q,\, \sin\varphi_Z,\, \sin\varphi_W,\, \cos\varphi_Z,\, \cos\varphi_W\right\}$$

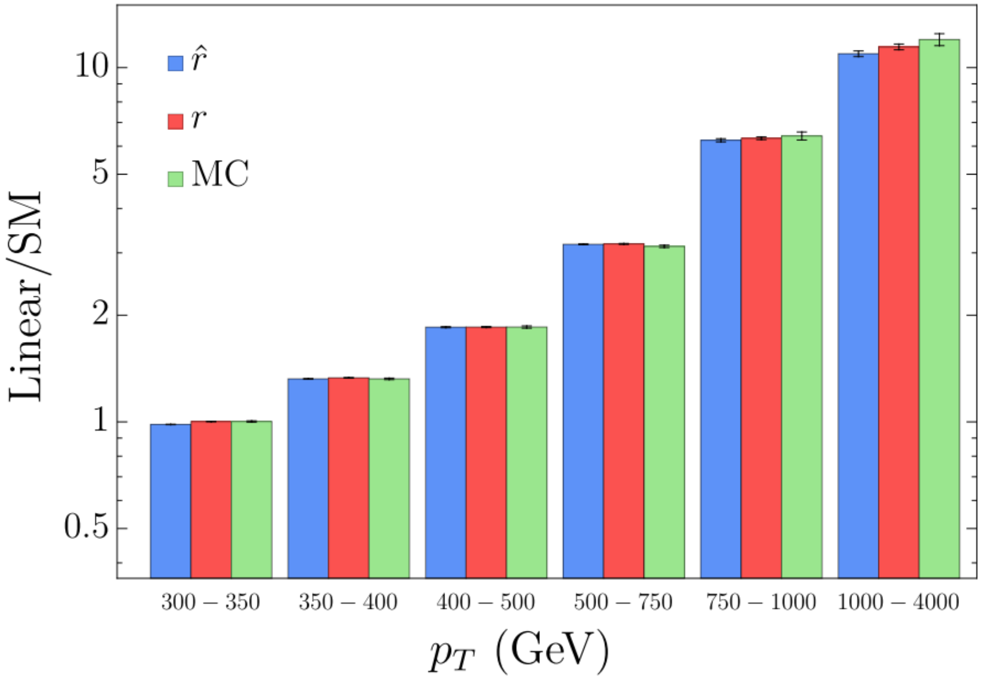
Kinematics in the **physical space**

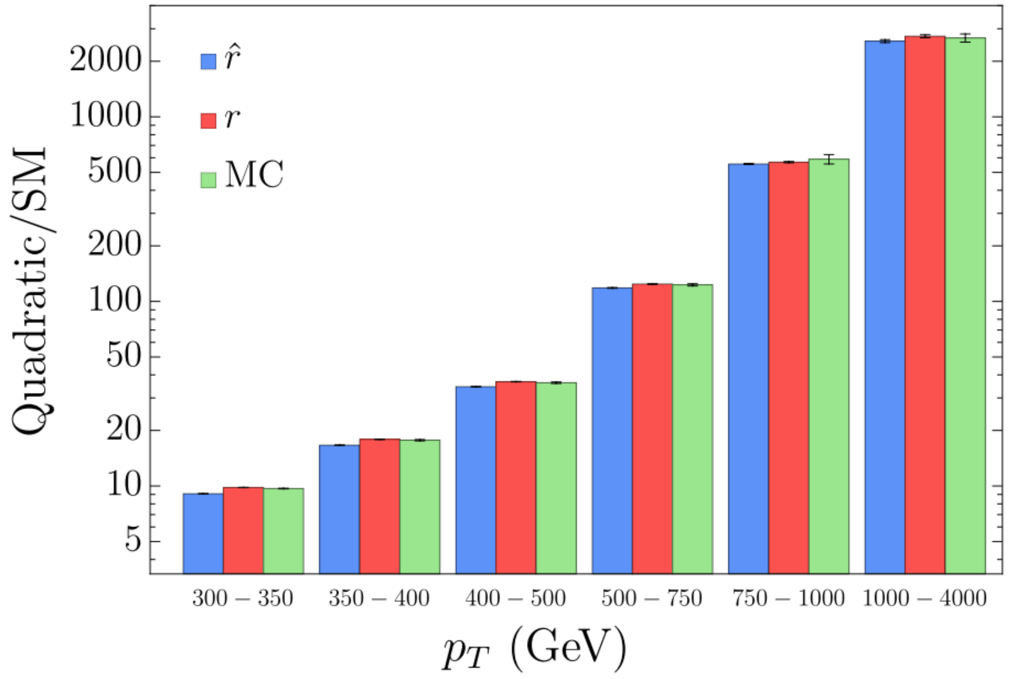
Variables after reconstructing the neutrino

Redundancy helps a little bit, sin and cos of angles are useful to impose periodicity exactly

Toy case: validation

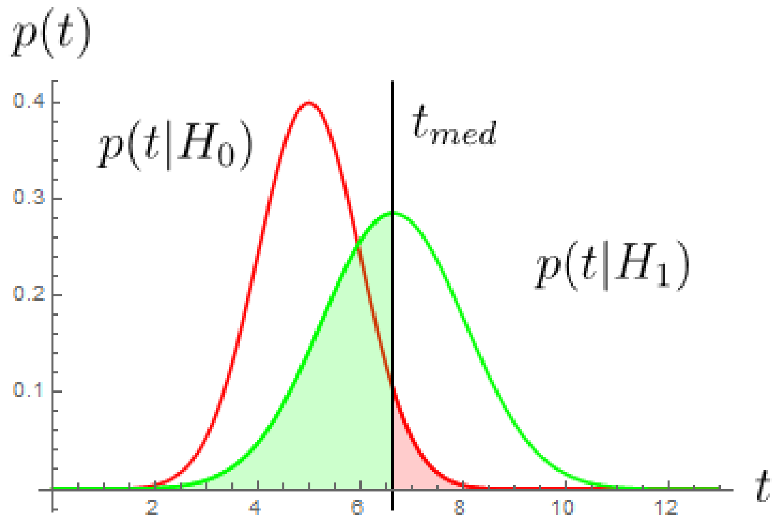
We can check how well the network **reconstructs** the **linear** and **quadratic** terms of the differential cross-section





Toy case: validation

For a more quantitative check of performance, we use the **Neyman-Pearson p-value**

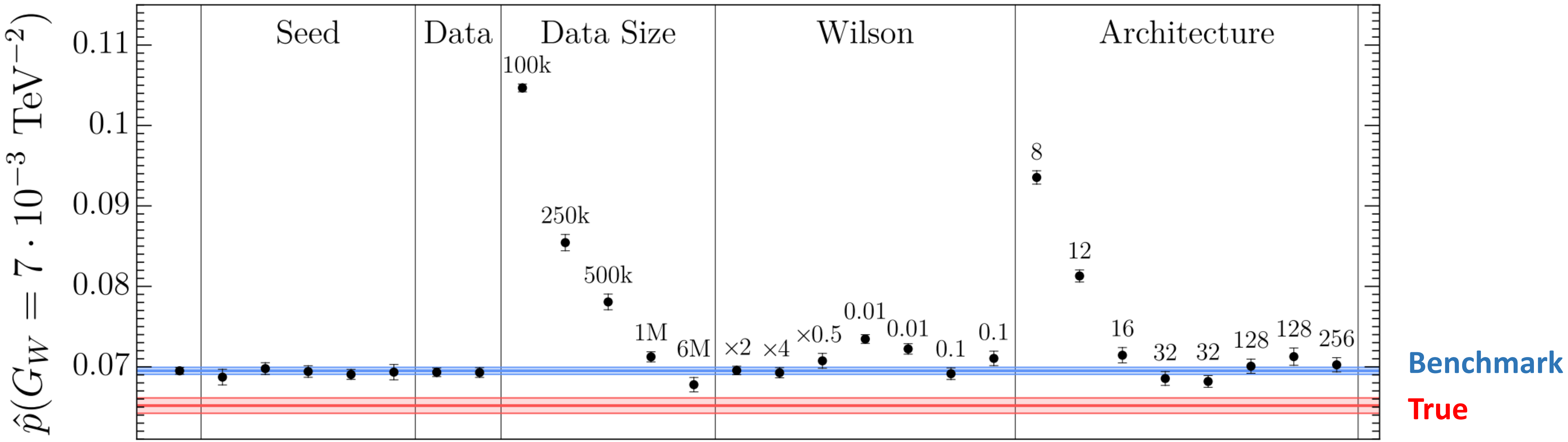


$$t(\mathcal{D};c) = -2\left(N(c) - N(0) + \sum_{x\in\mathcal{D}} \tau_c(\mathcal{D})\right)$$

For the **true likelihood** this test gives the **best possible bound** (Neyman-Pearson lemma)

For the **network** this gives an objective **measure** of how well the **true likelihood is approximated**

Toy case: hyperparameters



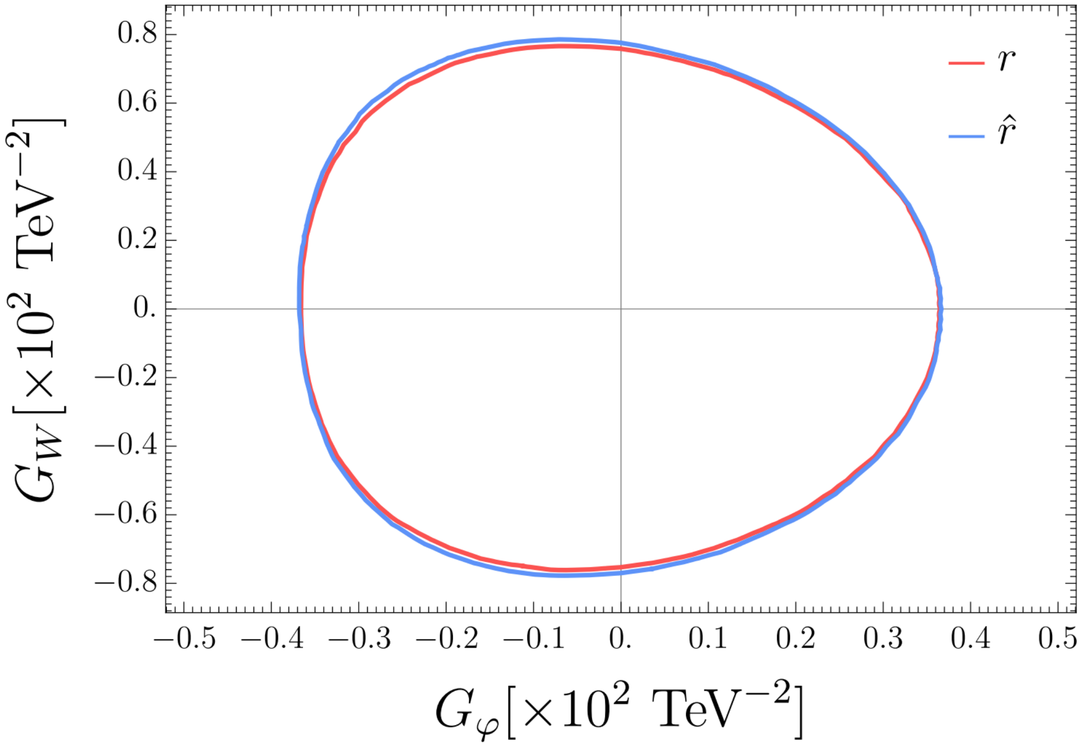
No dependence on initial configuration or training data

More data = better

Choice of training coefficients marginally relevant

Architecture must be selected to avoid underfit or overfit

Toy case: results



Red: **optimal** exclusion bound
Blue: **Neural Network** result

- 5 Neural Networks {10, 24, 24, 1}
- Sigmoid activations
- Adam optimizer
- 3 Million reweighted training points
- 1000 epochs/minute on a GPU
- ~ 200k total epochs

NLO case: implementation

For a more realistic example we studied the same process generated with **MadGraph at NLO QCD**, with reweighting on New Physics

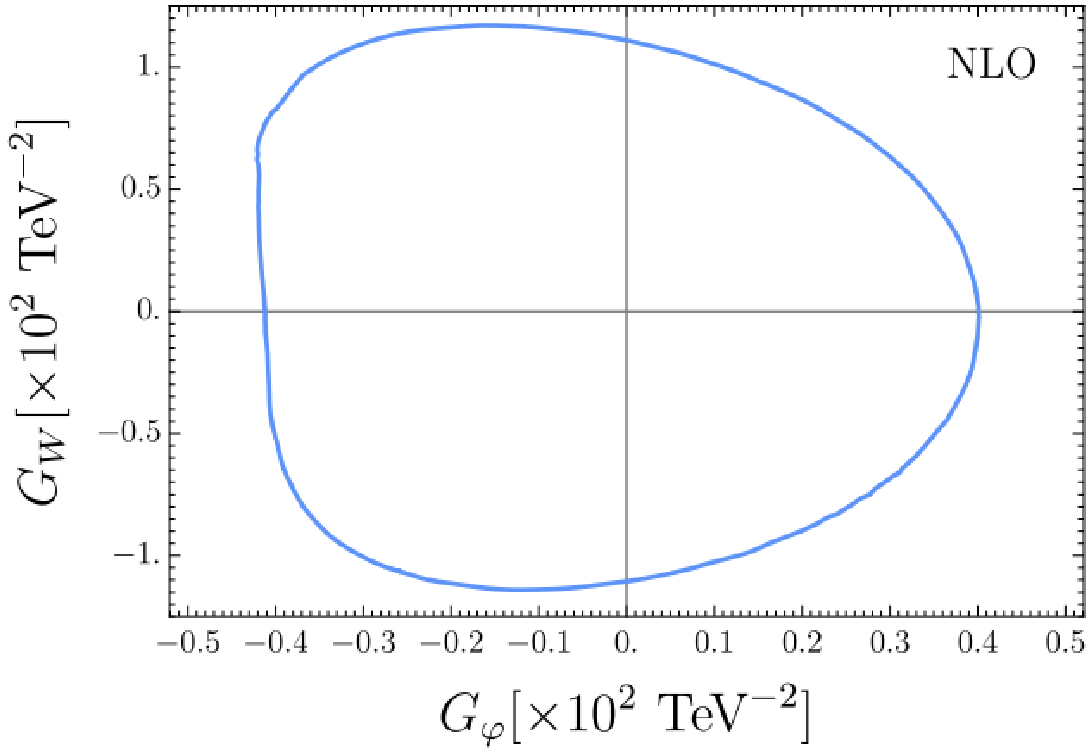
In this case the network is trained on **13** features

$$\left\{ \log[s/\mathrm{GeV}^2],\, \Theta,\, \theta_Z,\, \theta_W,\, \log[p_T/\mathrm{GeV}],\, \log[1{+}p_{T,\mathrm{ZW}}/\mathrm{GeV}],\, Q,\, \ell_Z,\, \ell_W\ \sin\varphi_Z,\, \sin\varphi_W,\, \cos\varphi_Z,\, \cos\varphi_W \right\}$$

Total WZ transverse momentum,
Non trivial due to additional jet

Flavor or Z and W decays
Distribution changes due to QED showering

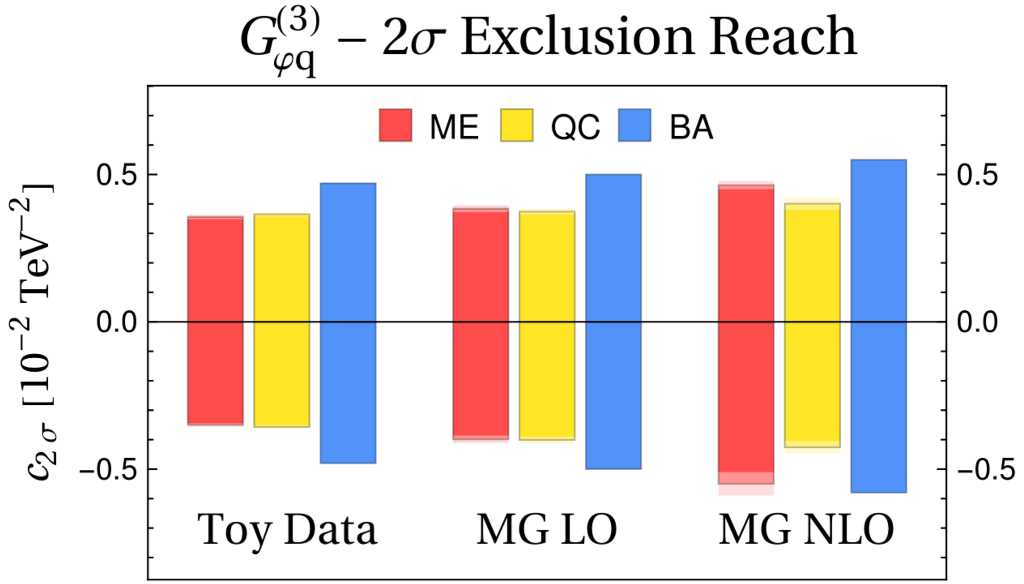
NLO case: results

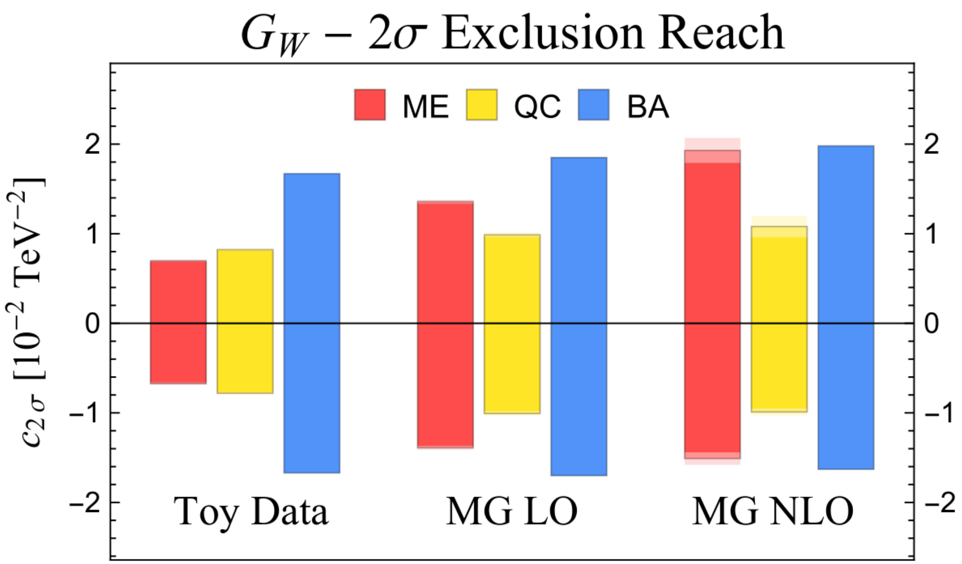


- 5 Neural Networks {13,32, 32, 1}
- Sigmoid activation
- Adam optimizer
- 3 Million reweighted training points
- 1000 epochs/minute
- ~ 200k total epochs

Comparing to Binned Analysis

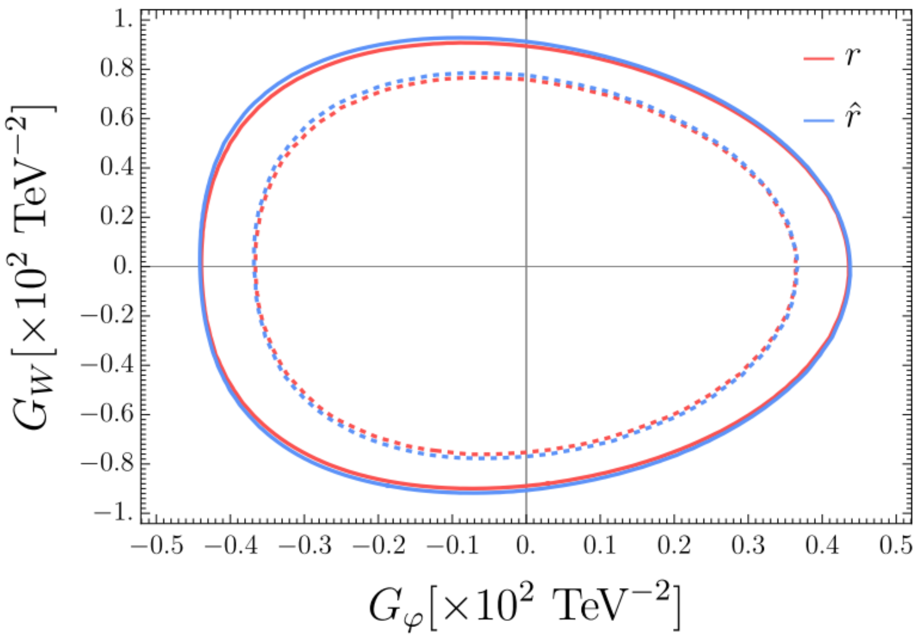
ME = Toy Matrix Element; **QC** = Quadratic Classifier; **BA** = Binned Analysis

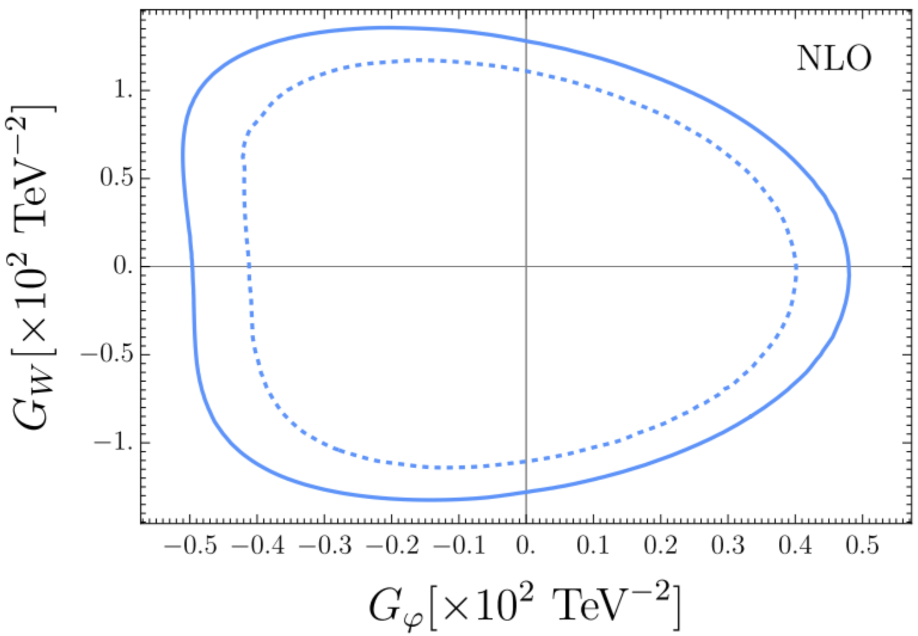




Profile Likelihood

A standard **profile likelihood test** can be used and is nearly optimal





Conclusions

Multivariate analysis can greatly **increase the sensitivity** on BSM parameters, especially when new physics enters in multiple observables with a complex interference pattern.

Machine learning methods can overcome these difficulties by **learning the fully differential distribution** accurately from a Monte Carlo sample. Making the approximation more reliable is just a matter of increasing the training sample size and network architecture.

Two strategies to improve the learning of Likelihood from simulations

- Training on **reweighted samples** reduces number of training points needed and leads to a higher accuracy
- **Linear** and **quadratic** EFT terms can be learned separately in order to fit the likelihood also as a function of the Wilson Coefficients

In progress: including PDF uncertainties, quantifying the impact of detector (Delphes) effects, …