Improving the likelihood learning

Alfredo Glioti

Institut de Physique Théorique

The LHC precision program
05/10/2023

Two possible improvements

Two ways of **improving** the likelihood trick and go **beyond** the simple classifier

Learning from reweighted data

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

- Reduces statistical fluctuations from Monte Carlo
- Less data needed for training

Learning EFT quadratic dependence

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

- Analytic dependence on the Wilson Coefficients
- Improves performances by learning on regions where BSM and SM are very different

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 $f(x) = 1/2 + \delta f(x)$

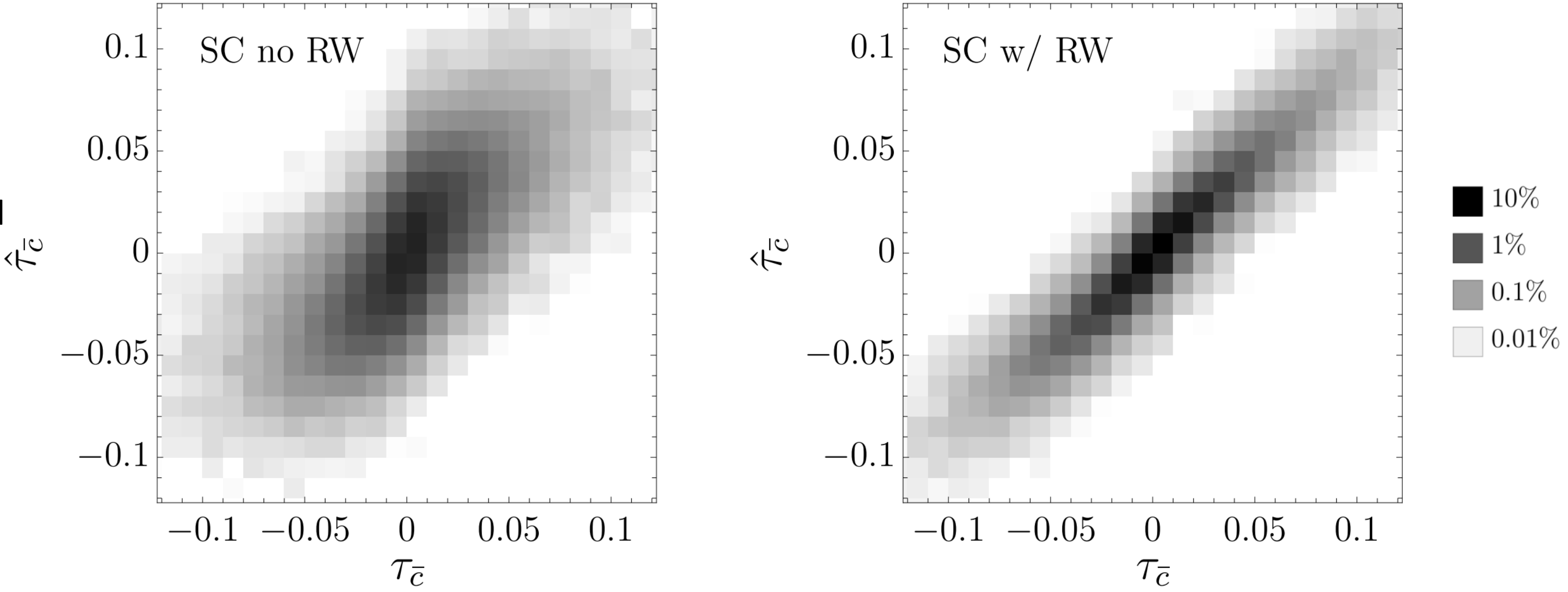
No RW → $$\ell[f(\cdot)] = \sum_{\mathrm{e}\in\mathrm{S}_0} w_{\mathrm{e}}(\bar{c})\delta f(x_{\mathrm{e}}) - \sum_{\mathrm{e}\in\mathrm{S}_1} w_{\mathrm{e}}(0)\delta f(x_{\mathrm{e}}) + \sum_{\mathrm{e}\in\mathrm{S}_0} w_{\mathrm{e}}(\bar{c})\delta f(x_{\mathrm{e}})^2 + \sum_{\mathrm{e}\in\mathrm{S}_1}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

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

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

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

This will converge to the likelihood ratio during training

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

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

λ upper triangular matrix

$$\lambda_{11} = 1$$

While the other components are Neural Networks

c Wilson Coefficients

$$c_0 = 1$$

Parametrized classifier

For example, a possible parametrization of λ for two Wilson Coefficients is

$$\lambda(x) = \begin{pmatrix} 1 & \rho_1(x)\sin\theta_{11}(x) & \rho_2(x)\sin\theta_{22}(x) \\ 0 & \rho_1(x)\cos\theta_{11}(x) & \rho_2(x)\cos\theta_{22}(x)\sin\theta_{12}(x) \\ 0 & 0 & \rho_2(x)\cos\theta_{22}(x)\cos\theta_{12}(x) \end{pmatrix}$$

Where ρ and θ are all neural networks

This parametrization is also useful to train any number of coefficients

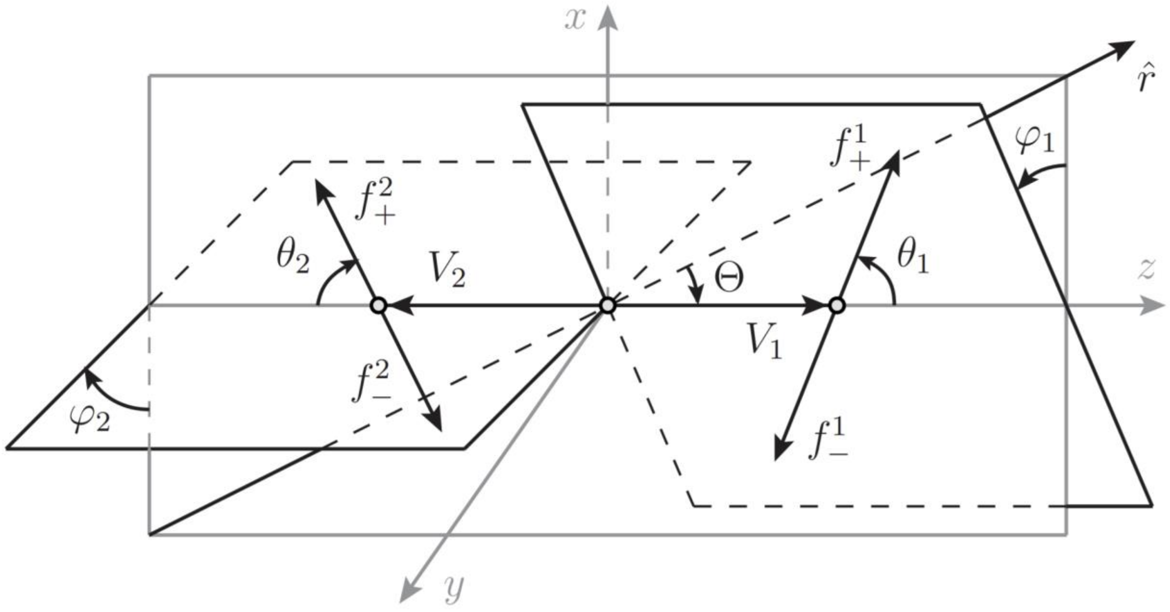
1) Train in all single-operator directions

2) Fix these networks and learn the mixed terms one by one by training on all possible pairs of Wilson Coefficients

Application: WZ at (HL-)LHC

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**

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

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

Six independent and discriminating variables: ŝ + 5 angles

Example of **interference resurrection**

Toy case: implementation

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

$$\{Q,\, \chi,\, \underline{\bar{s},\, \bar{\Theta},\, \bar{\theta}_W,\, \bar{\varphi}_W,\, \bar{\theta}_Z,\, \bar{\varphi}_Z\,,\, y}\}$$

Kinematics in the **latent space**

Helicity of Z decay products

Variables if the neutrino was measured

The observables given to the networks are

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

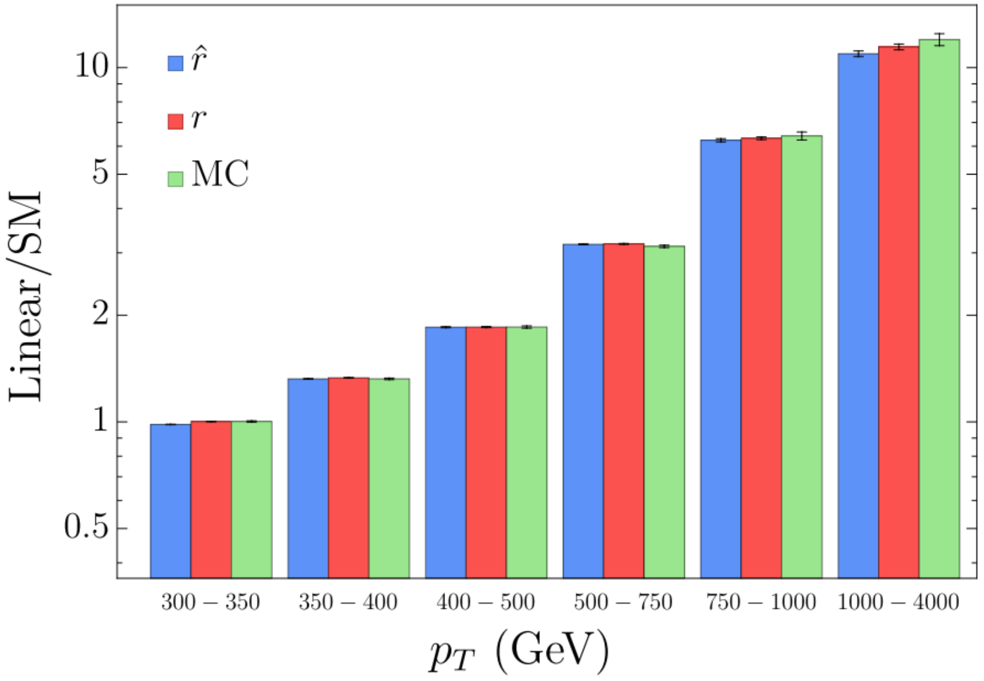
Kinematics in the **physical space**

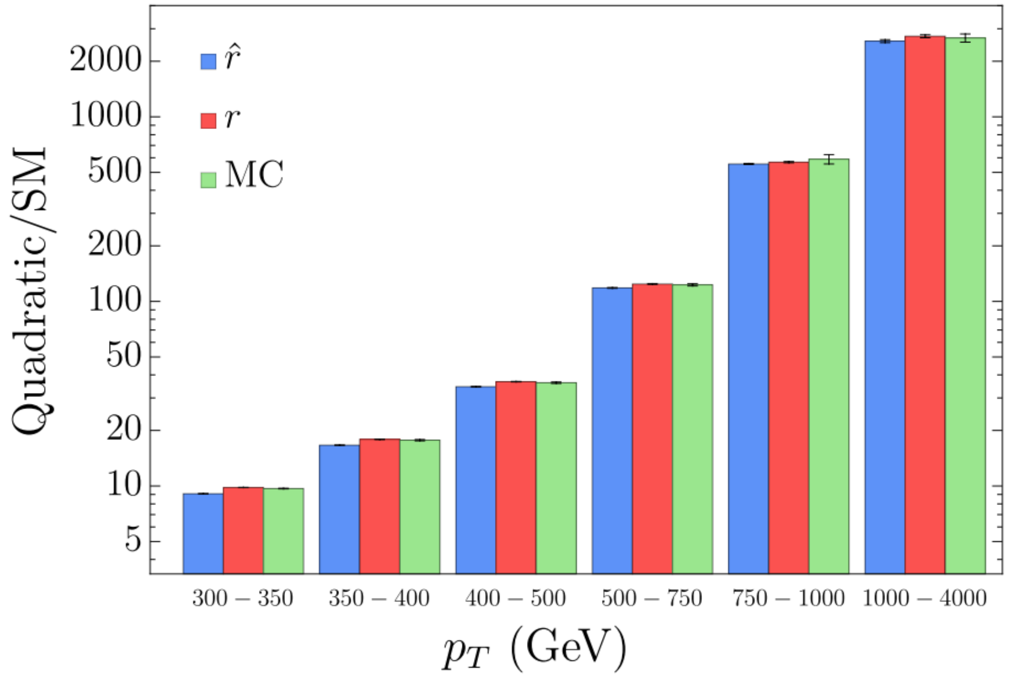
Variables after reconstructing the neutrino

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

Toy case: validation

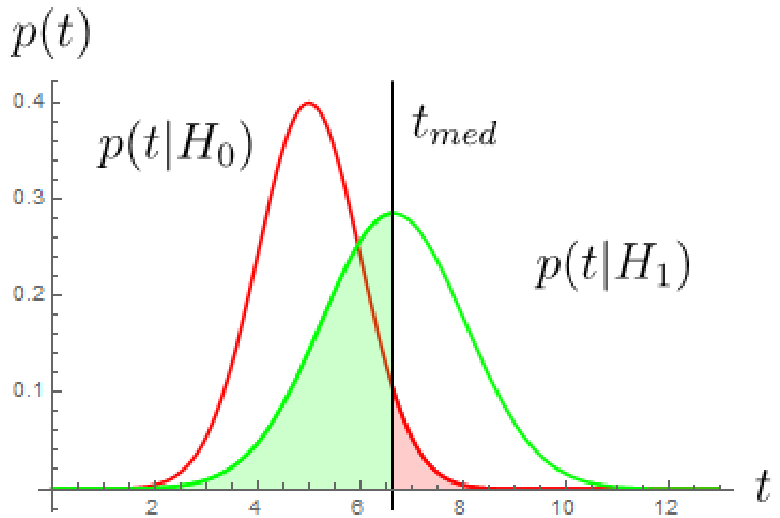
We can check how well the network **reconstructs** the **linear** and **quadratic** terms of the differential cross-section





Toy case: validation

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

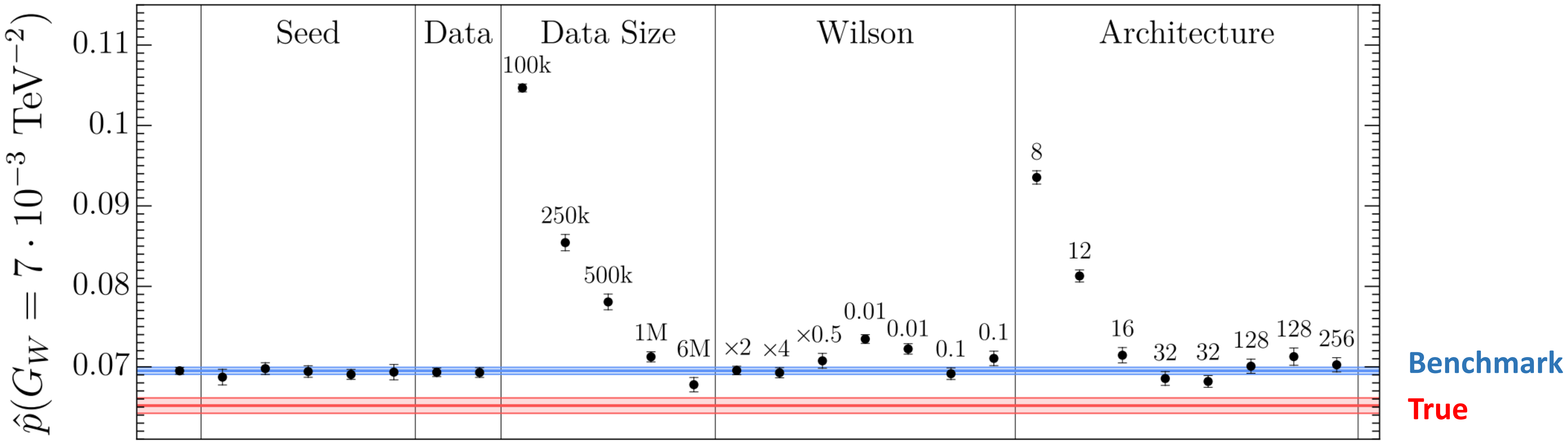


$$t(\mathcal{D};c) = -2\left(N(c) - N(0) + \sum_{x\in\mathcal{D}} \tau_c(\mathcal{D})\right)$$

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



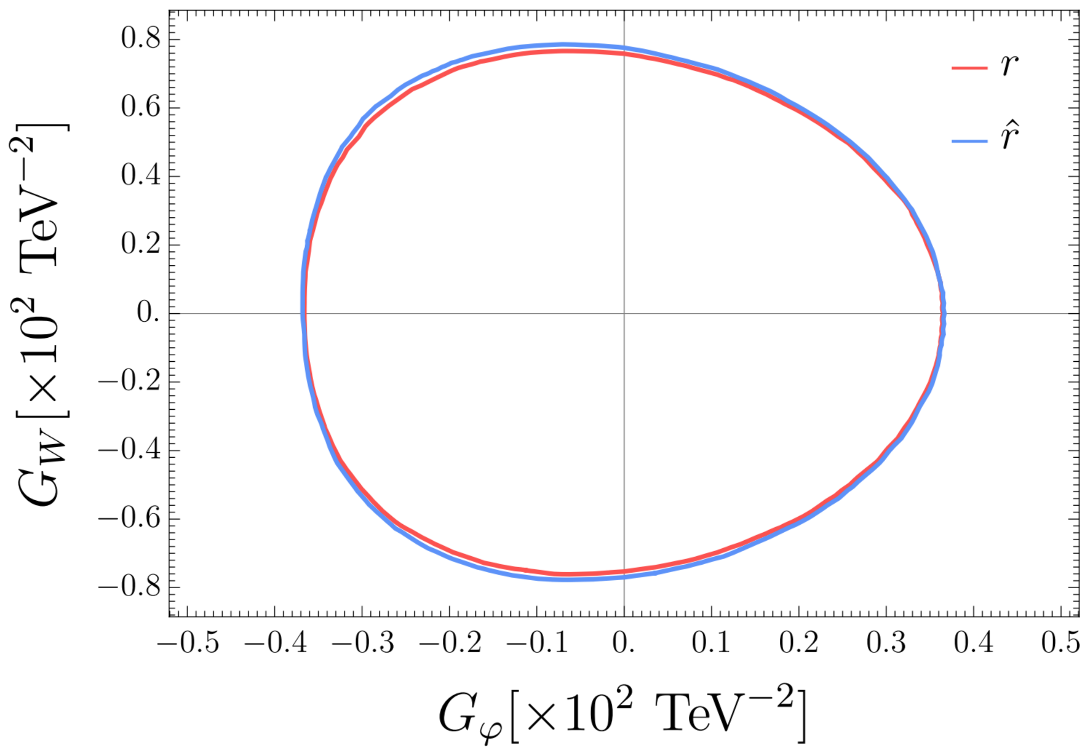
No dependence on initial configuration or training data

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

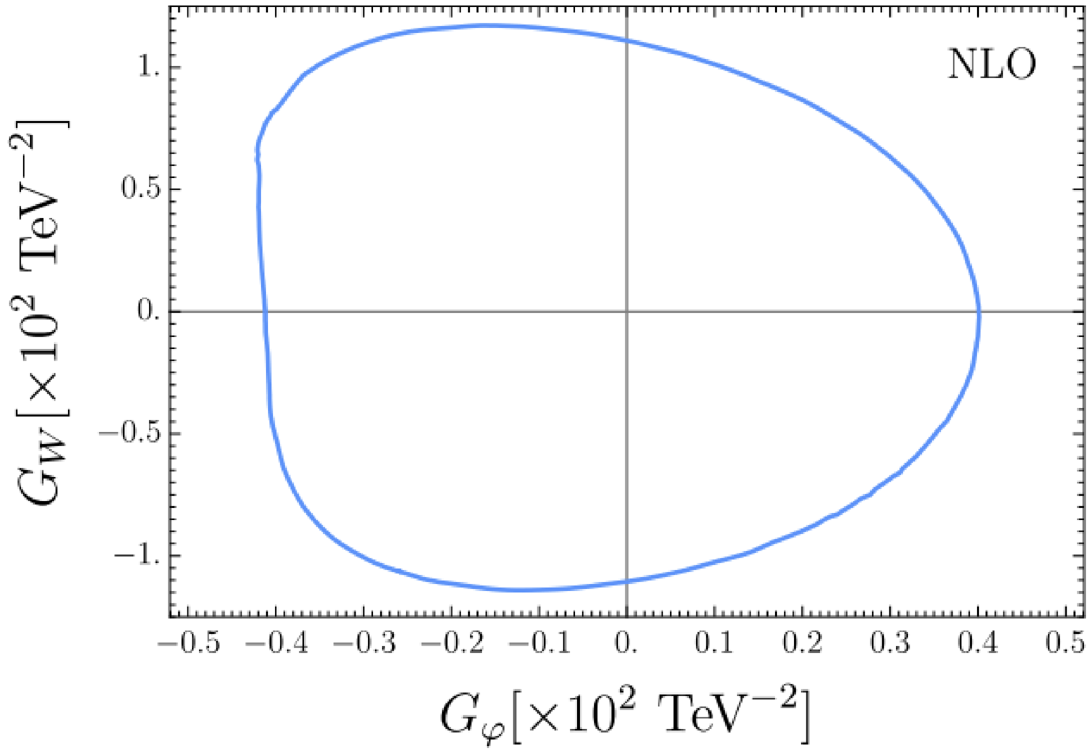
In this case the network is trained on **13** features

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

Total WZ transverse momentum,
Non trivial due to additional jet

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

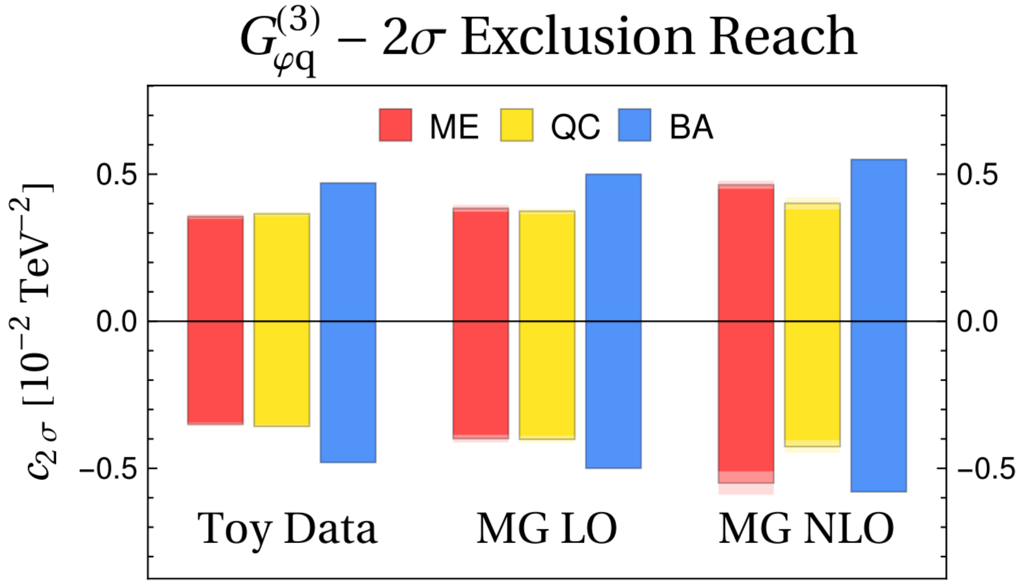
NLO case: results

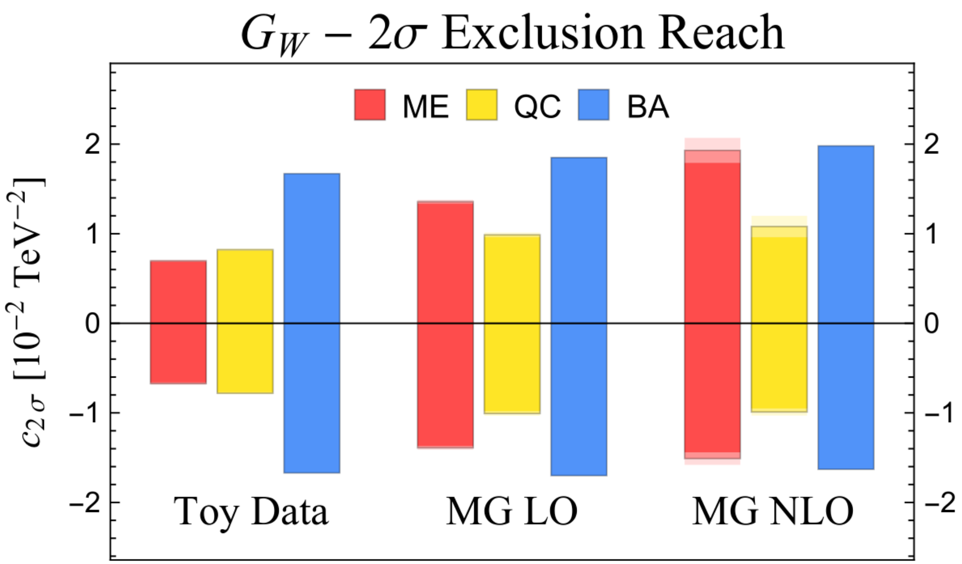


- 5 Neural Networks {13,32, 32, 1}
- Sigmoid activation
- Adam optimized
- 3 Million reweighted training points
- 1000 epochs/minute
- ~ 200k total epochs

Comparing to Binned Analysis

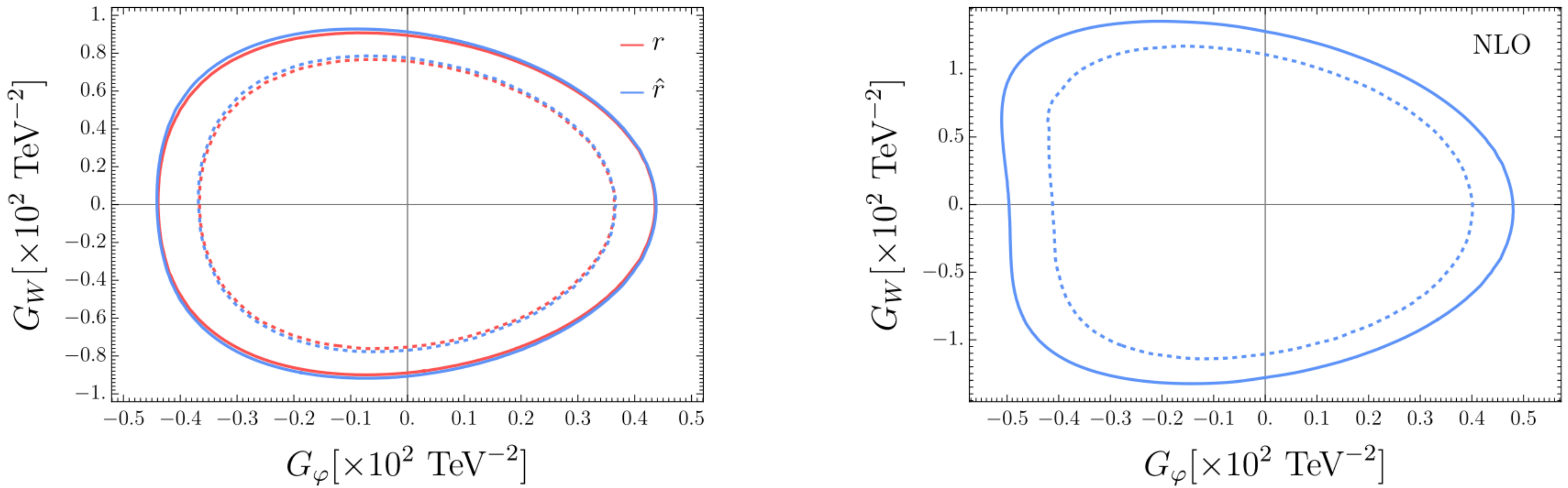
ME = Toy Matrix Element; **QC** = Quadratic Classifier; **BA** = Binned Analysis





Profile Likelihood

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



Conclusions

- **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

- The network performances are extremely close to **optimality**

- The same analysis strategy can be used for any process at any level of **complexity**