Imperial College London

The DLAdvocate: playing the *devil's advocate* with hidden systematic uncertainties

Andrei Golutvin[1], Aleksander Iniukhin[2], Andrea Mauri[1], Patrick Owen[3], Nicola Serra[3], Andrey Ustyuzhanin[4]

based on arXiv:2303.15956

[1] Imperial College London
[2] Yandex School of Data Analysis
[3] Universität Zürich
[4] National University Singapore

ML4jets 2023, Hamburg, 6-10 November 2023

Motivation

Measurement = Central value ± Statistical uncertainty ± Systematic uncertainty ± Hidden systematic uncertainty

- How to alleviate the risk of *hidden systematic uncertainties*
  - independent confirmation from a different experiment

Under which condition one can claim a physics discovery in an experiment which has unique physics sensitivity and therefore no direct competitors?

- *Deep Learning (DL) Advocate* to quantitatively address the unknown unknowns

Motivation

Measurement — Central value ± Statistical uncertainty ± Systematic uncertainty ± Hidden systematic uncertainty

- How to alleviate the risk of *hidden systematic uncertainties*
  - independent confirmation from a different experiment

> Under which condition one can claim a physics discovery in an experiment which has unique physics sensitivity and therefore no direct competitors?

- *Deep Learning (DL) Advocate* to quantitatively address the unknown unknowns

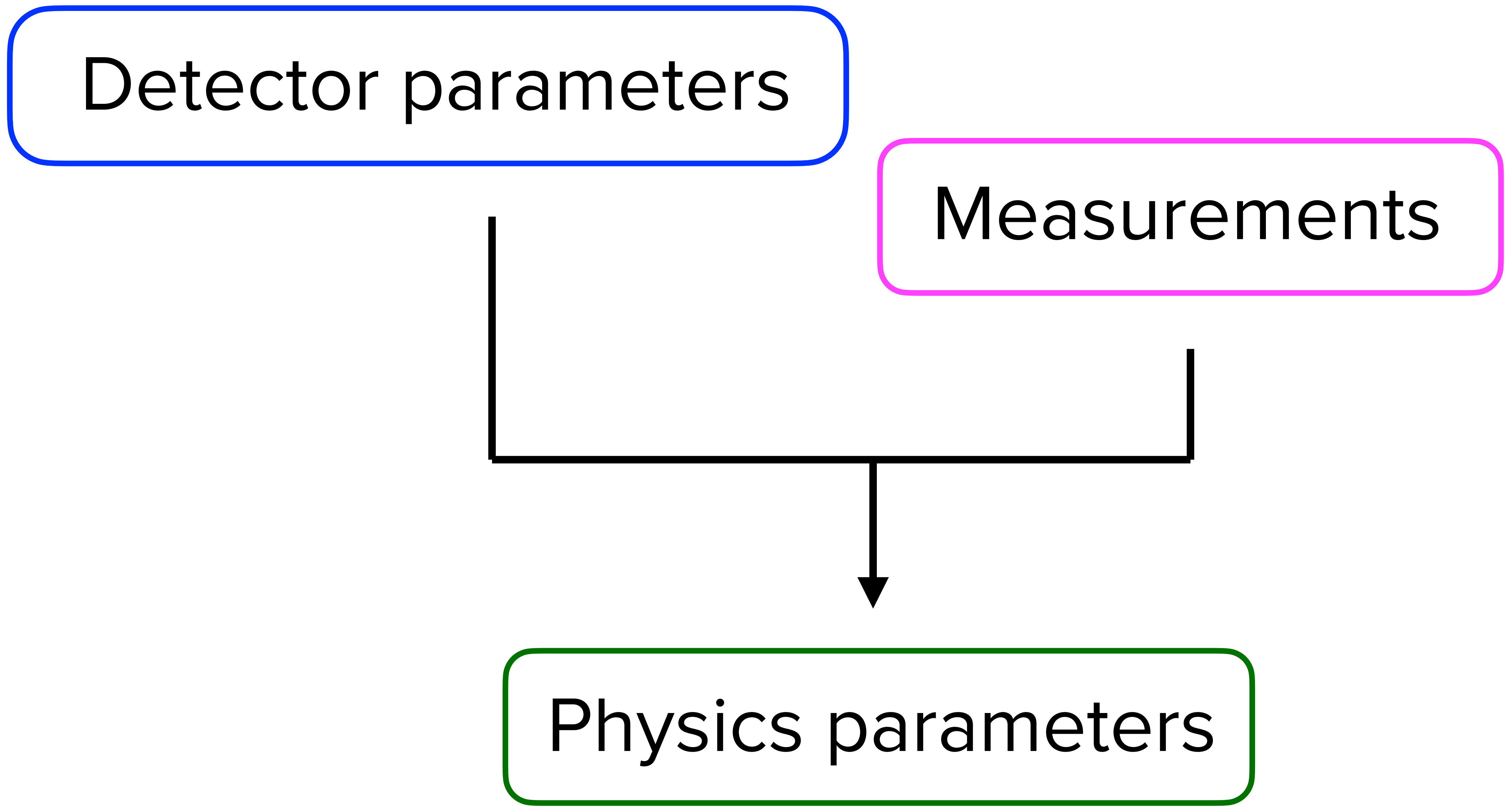
The traditional logic flow of a measurement

Measurements

Detector parameters

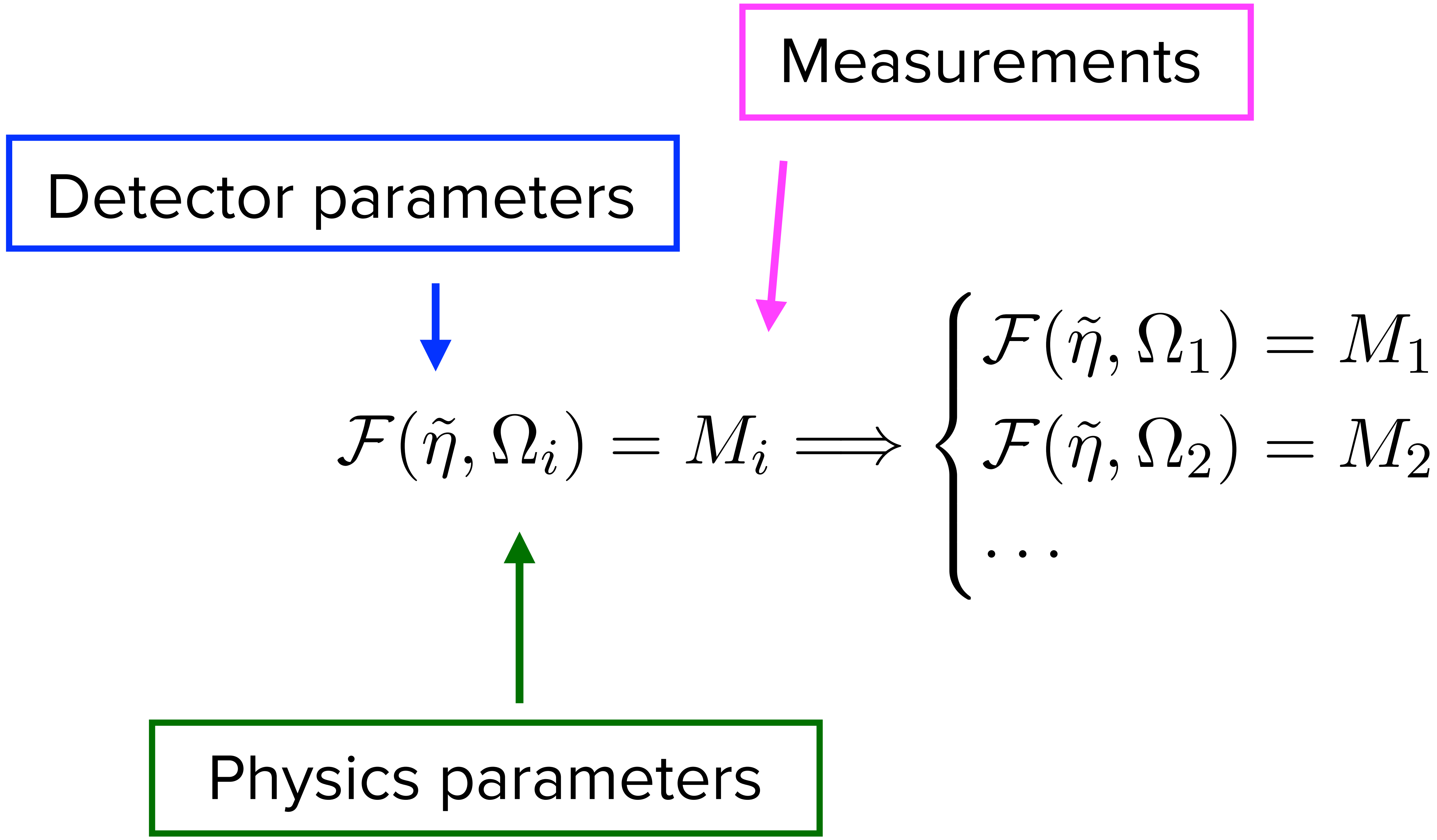
$$\mathcal{F}(\tilde{\eta}, \Omega_i) = M_i \Longrightarrow \begin{cases} \mathcal{F}(\tilde{\eta}, \Omega_1) = M_1 \\ \mathcal{F}(\tilde{\eta}, \Omega_2) = M_2 \\ \dots \end{cases}$$

Physics parameters

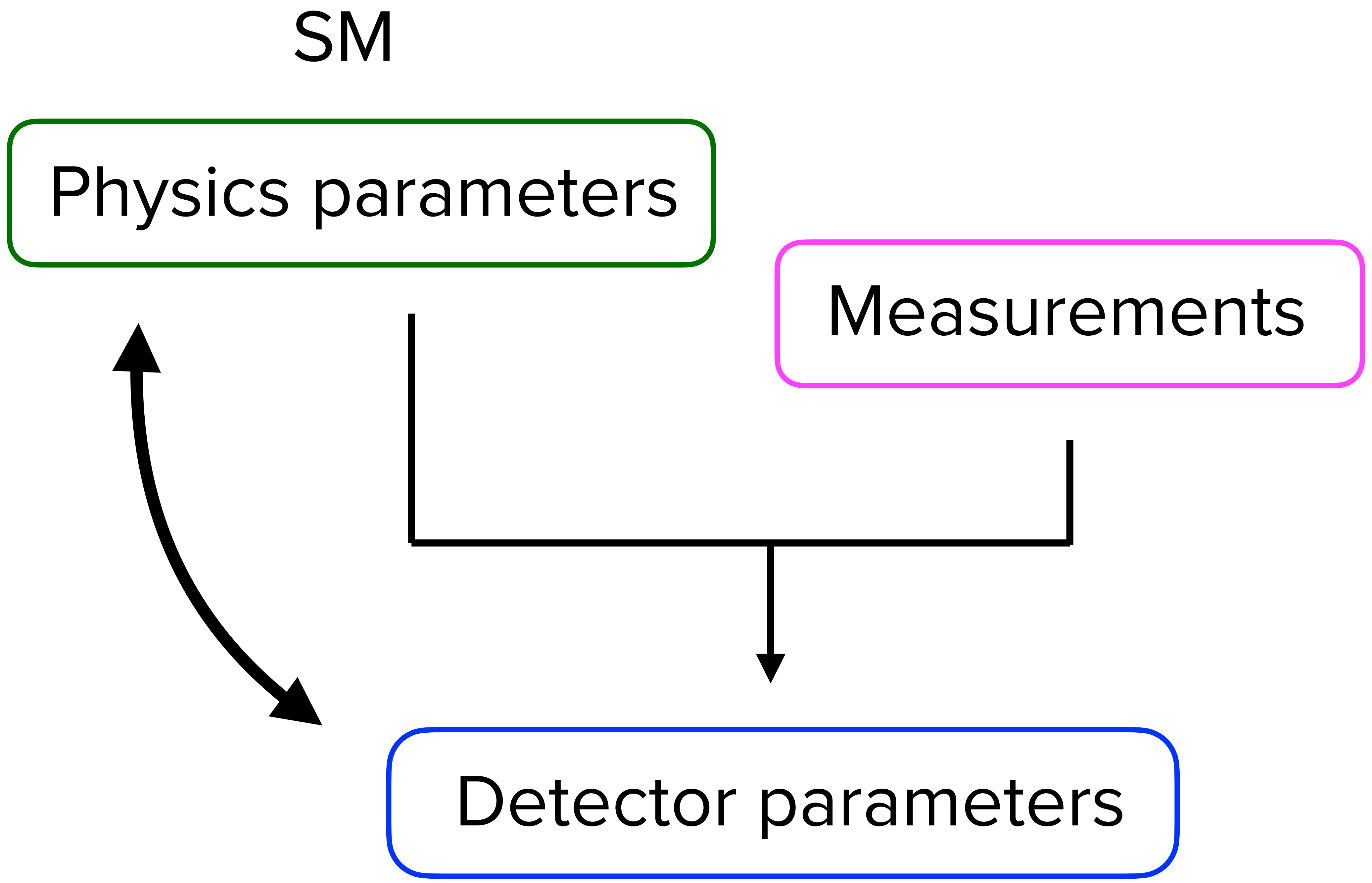


SM or NP?

The DLAdvocate logic flow



$$\mathcal{F}(\tilde{\eta}, \Omega_i) = M_i \Longrightarrow \begin{cases} \mathcal{F}(\tilde{\eta}, \Omega_1) = M_1 \\ \mathcal{F}(\tilde{\eta}, \Omega_2) = M_2 \\ \dots \end{cases}$$



Can I explain an anomaly I see in the data by modifying the detector parameters?

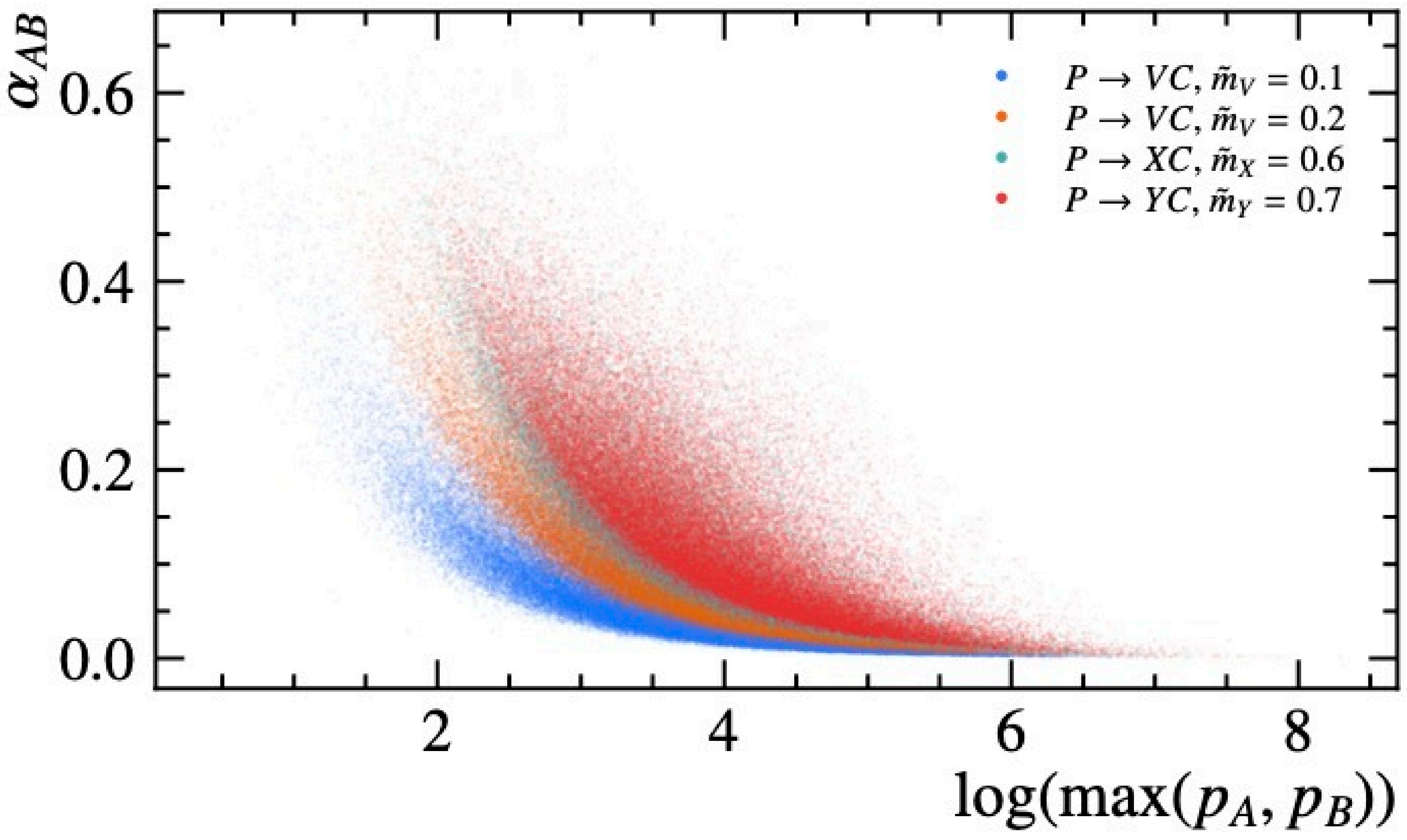
Playing the DL advocate: employ Deep Learning to systematically check all[*] possible effects

[*] For the moment we will focus on the detector efficiency

A simple example: a BR measurement

- Signal mode:
  - $P \to V(\to AB)C$ with mass m_V
- Control channel(s):
  - $P \to X(\to AB)C$
  - $P \to Y(\to AB)C$

  with known masses $m_{X(Y)}$



Different masses → different kinematic!

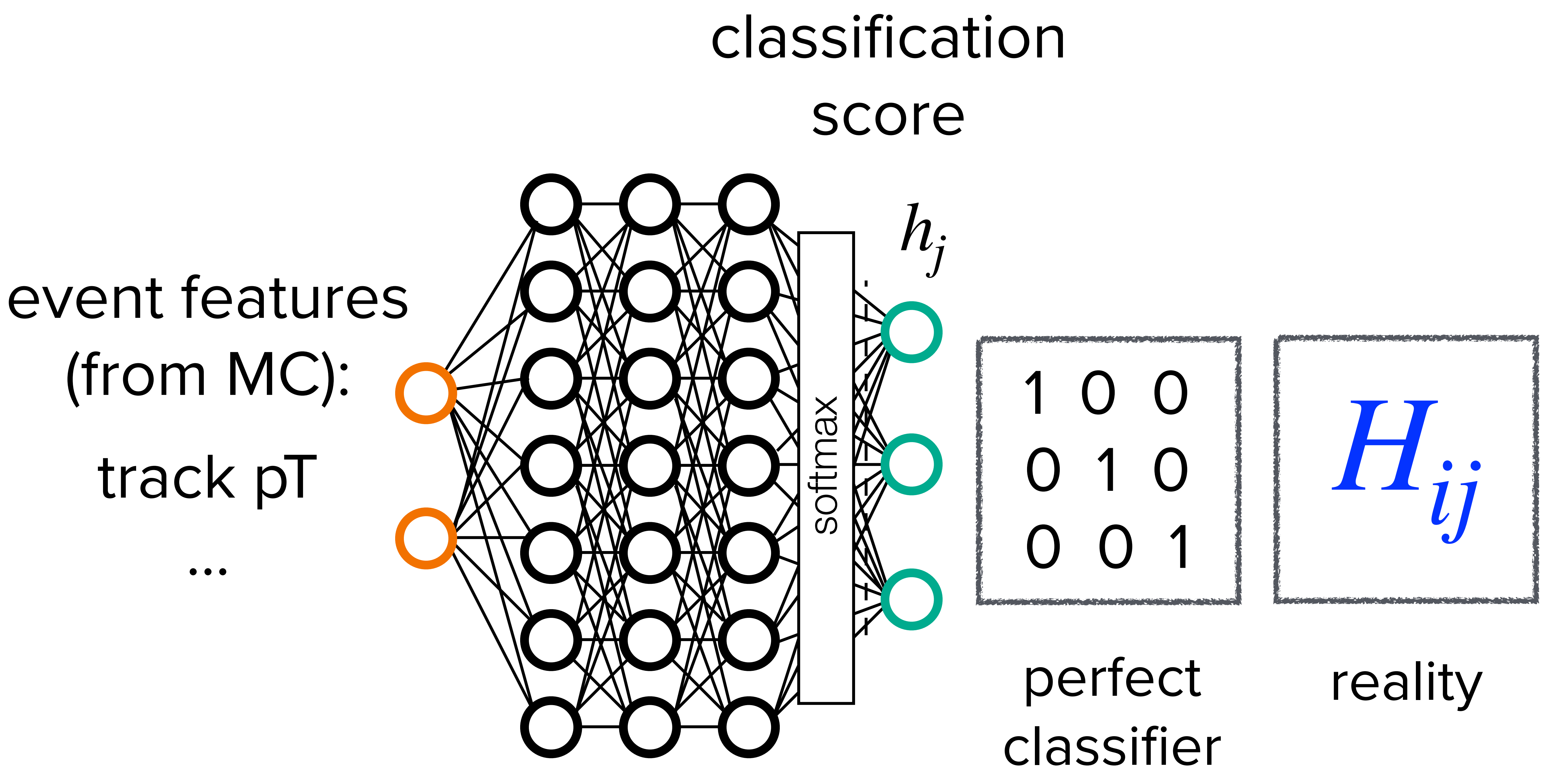
- Detector efficiency typically depends on kinematics (e.g. pT)
- A mismodelling of the efficiency will affect differently signal and control channels
- How a mismodelling of the efficiency can bias the signal given the constraints provided by the control channels

Key idea - step 1

- Train a classifier to distinguish the different channels

- The “perfect” classifier would be able to completely separate the phasespace of the different channels
    - control channels impose no constraints on the signal
    - I can arbitrarily modify the efficiency to bias the signal without touching the control channels

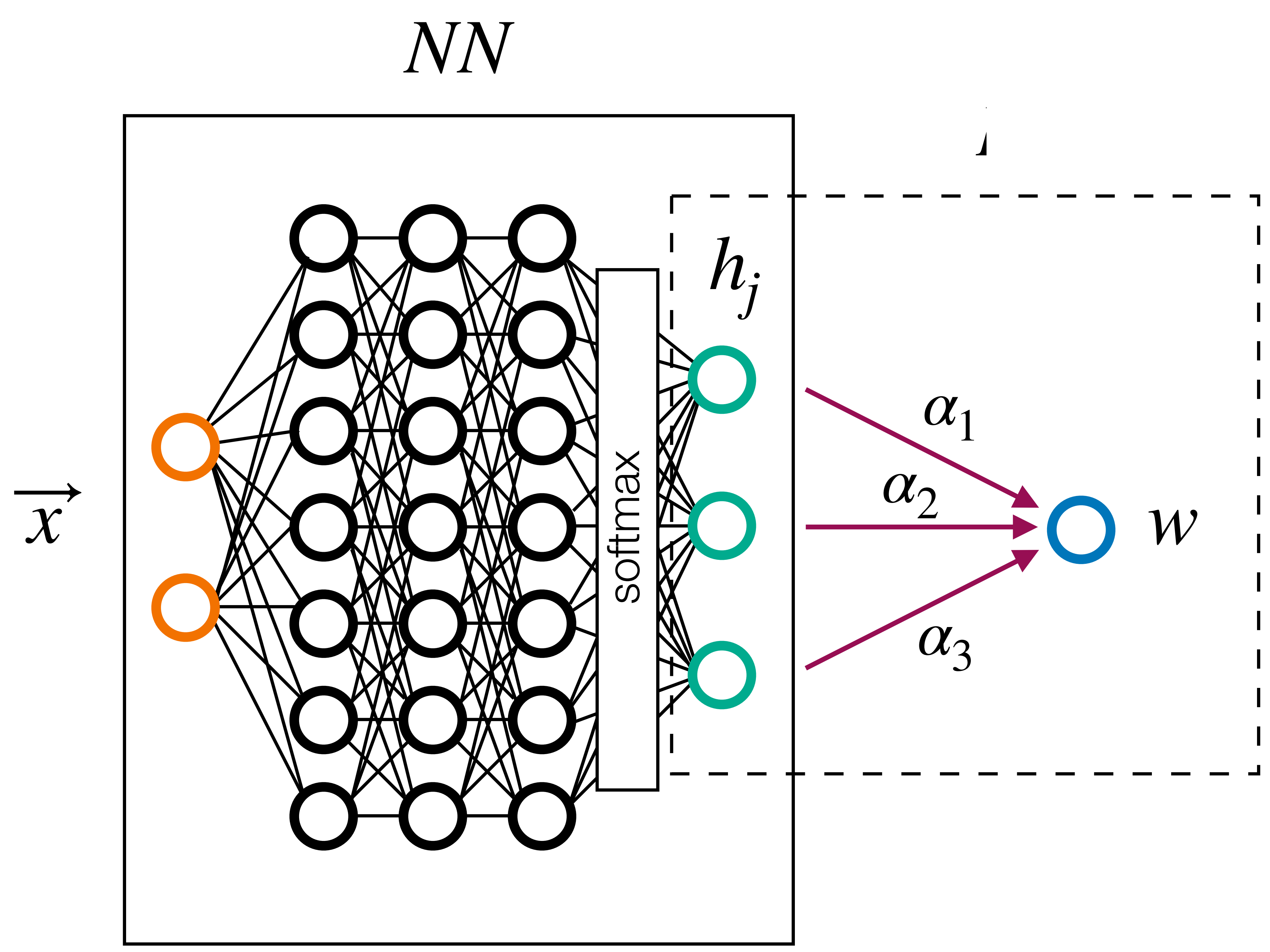
- Overlapping response will give the level of constraints provided by the different channels



Key idea - step 2

- Linear combination of NN output nodes to determine mismodelling weight as function of the input detector features

$$
w(x_i) \begin{cases} = 1 & \text{perfect efficiency} \\ < 1 & \text{efficiency over-estimated} \\ > 1 & \text{efficiency under-estimated} \end{cases}
$$



Channel efficiency

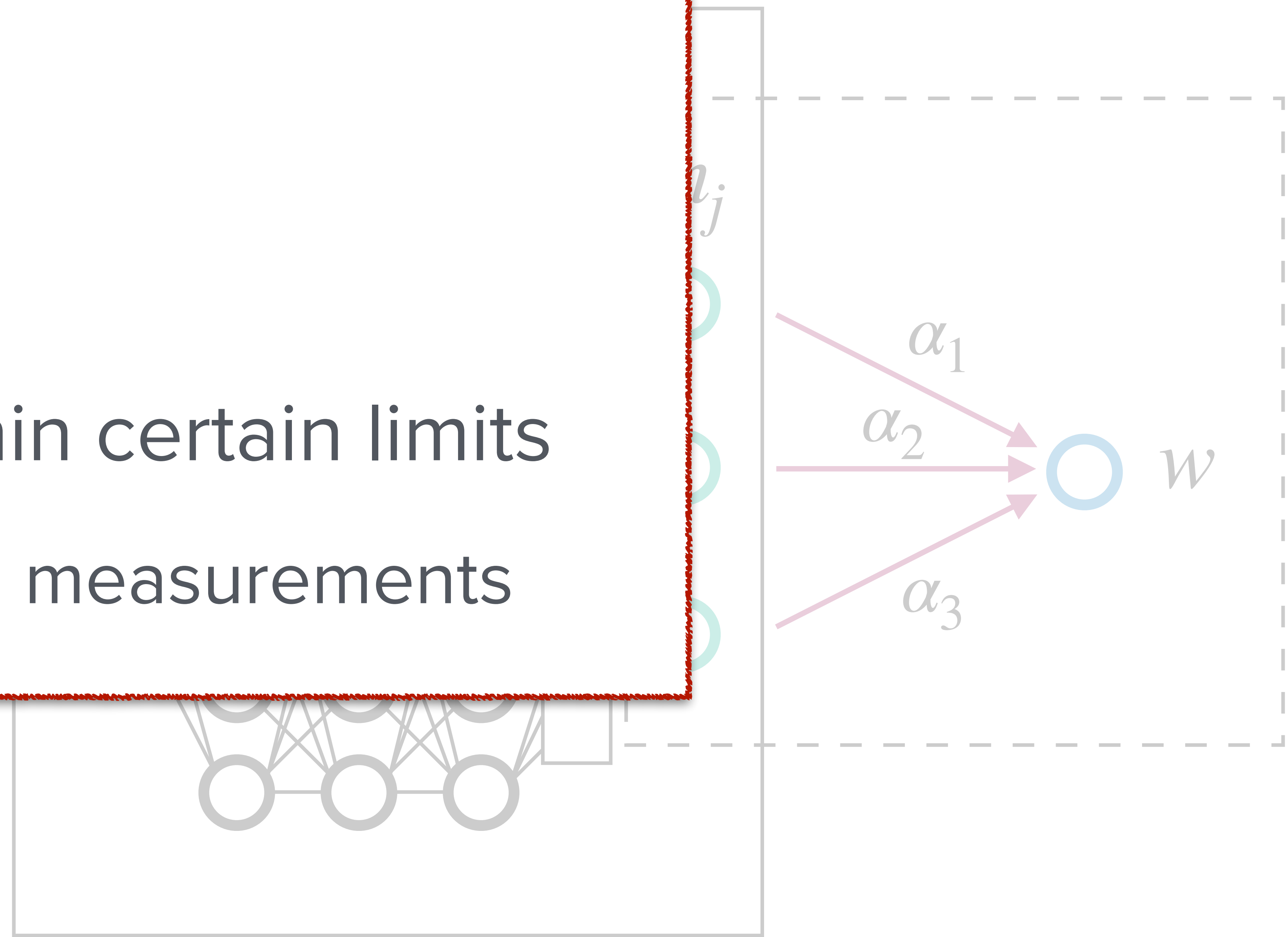
$$
e_i = \frac{1}{n_i} \sum_k \vec{\alpha} \cdot \vec{h}(x_{k,i})
$$

Evaluated on MC sample

Key idea - step 2

- Line…
to de…
func…

w…



<u>Goal of the algorithm</u>:

Check how biased can be the signal efficiency

$$\boxed{e_s \to \min}$$

while keeping the control channel efficiency within certain limits

$$e_i \in [V_i^{low}; V_i^{high}]$$ ← from measurements

Channel efficiency

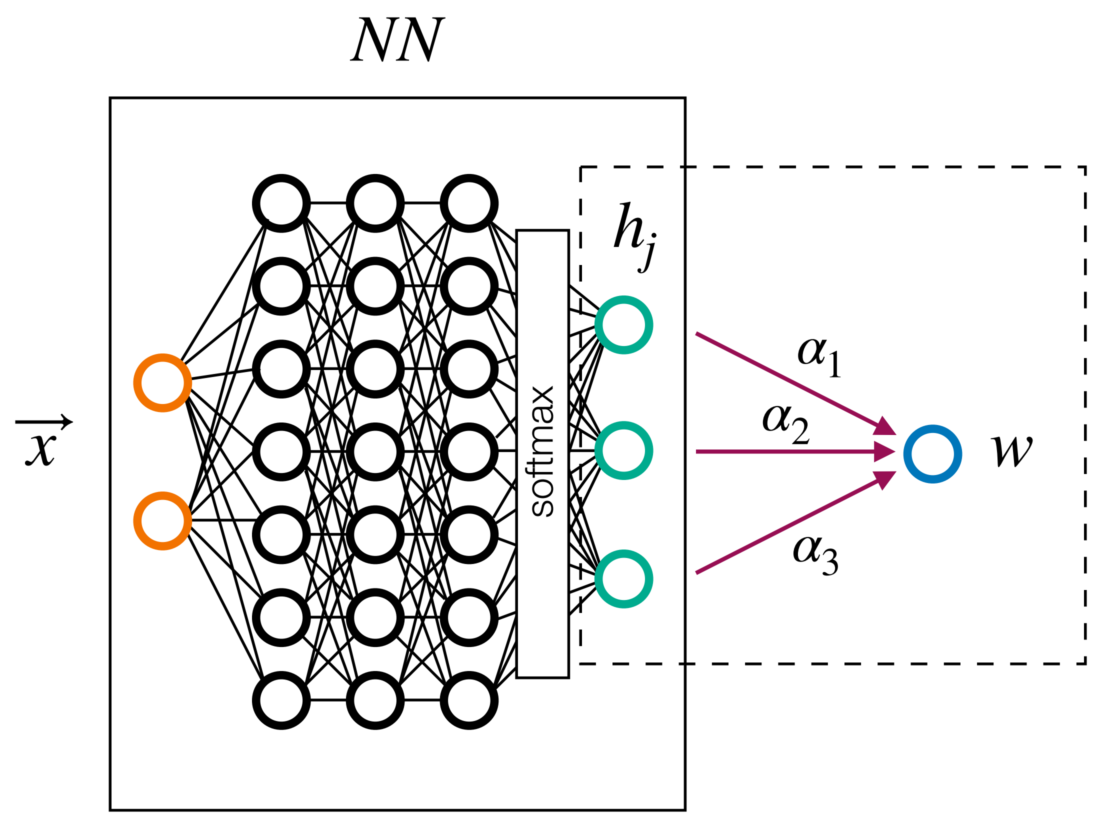
$$e_i = \frac{1}{n_i} \sum_k \vec{\alpha} \cdot \vec{h}(x_{k,i})$$

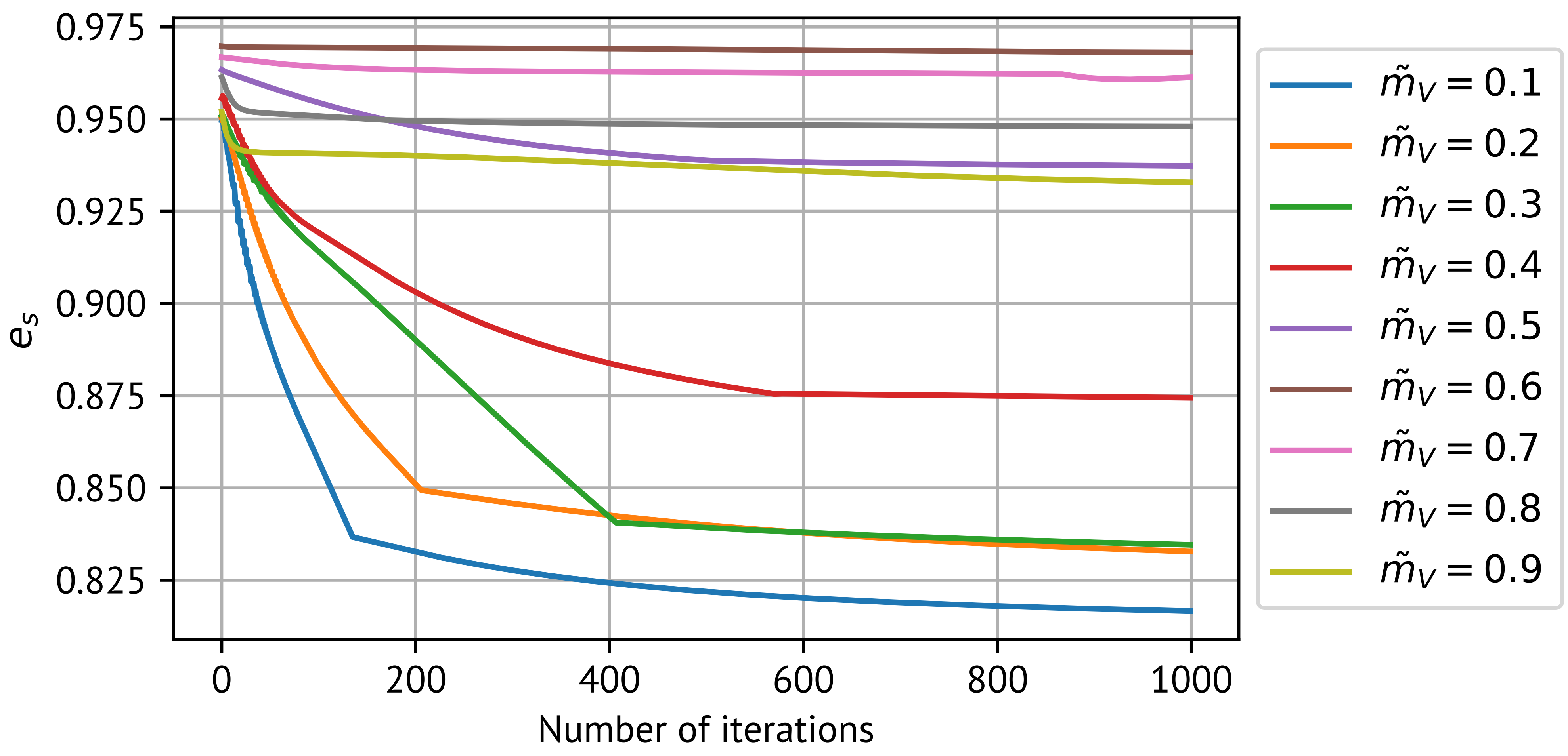
Evaluated on MC sample

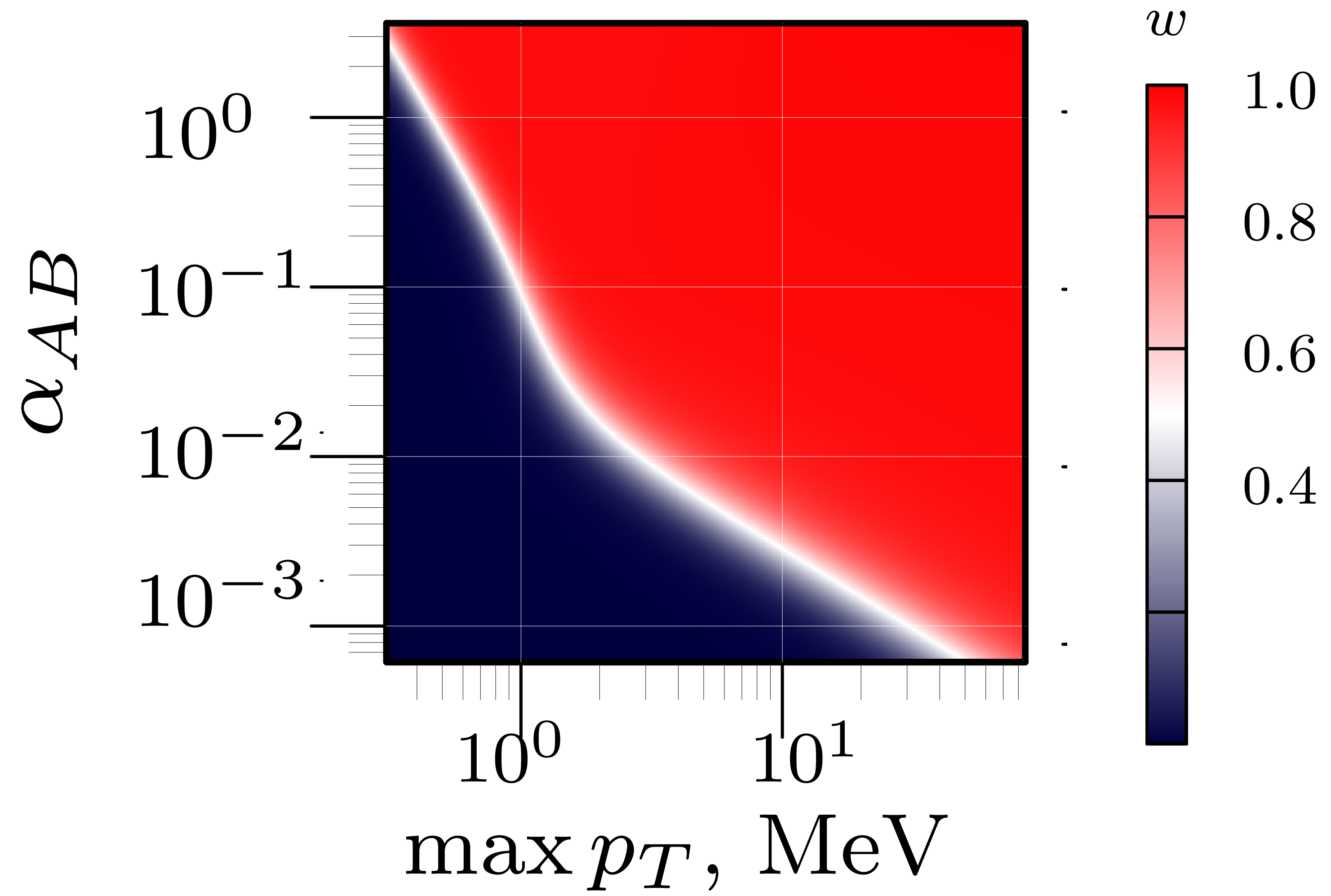
Training

- Iterative procedure:
  0. NN pretrained as a pure classifier
  1. update $\vec{\alpha}$
     - simple minimization with constraints
  2. update NN parameters
     - $\ell(\theta) = e_s - \log \left|\det(H)\right|$

       keeps matrix invertible







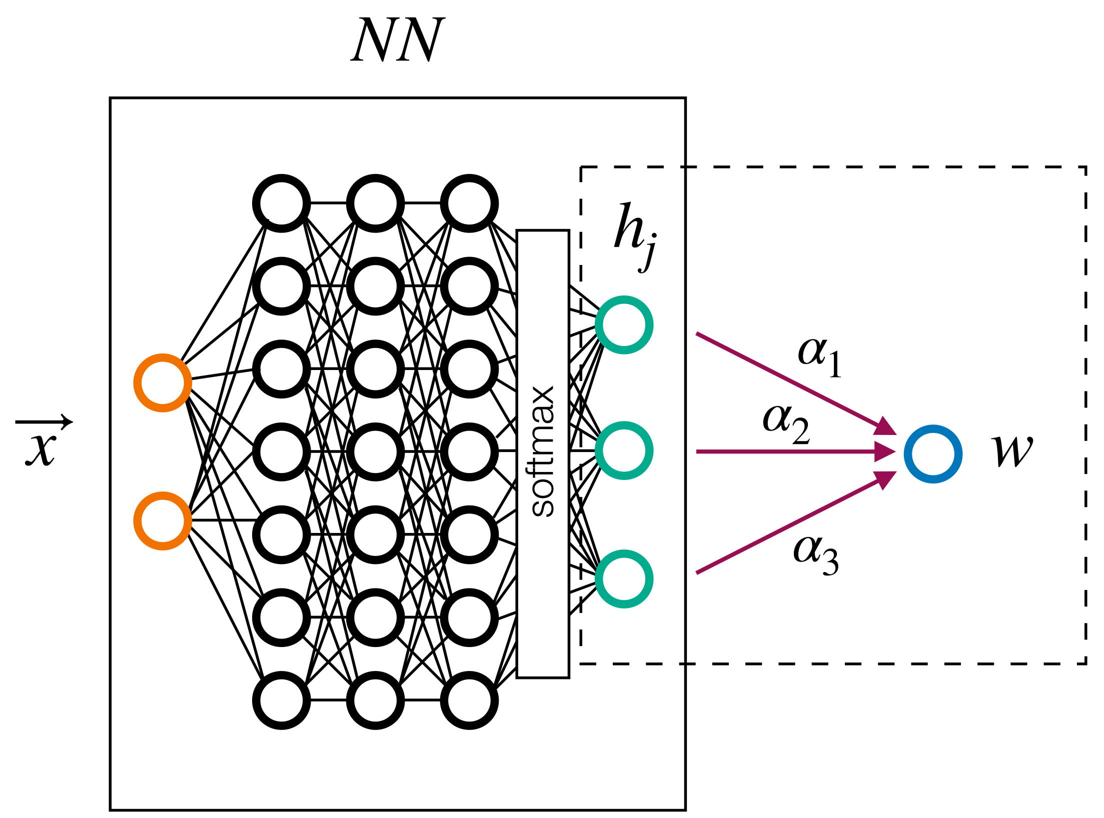
Training

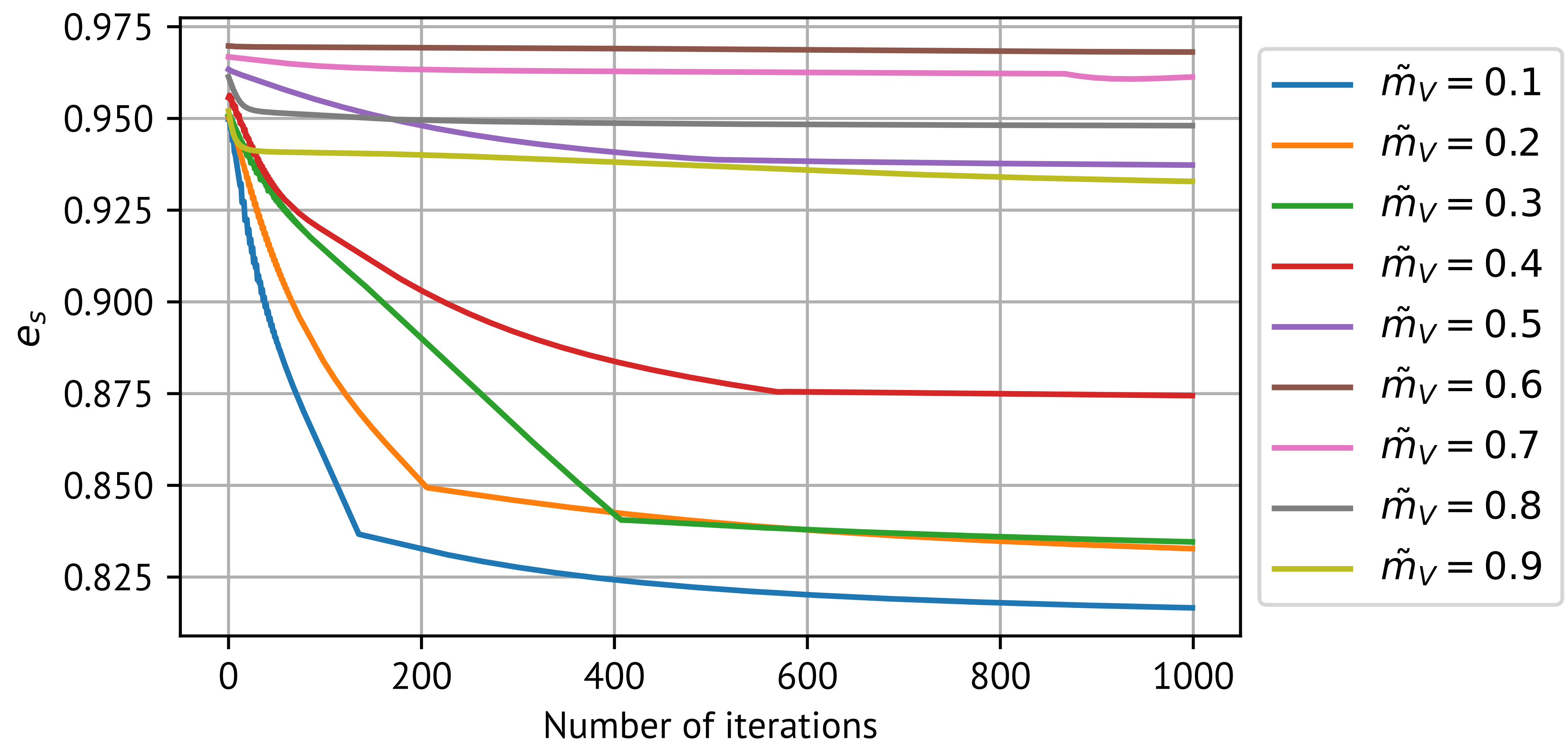
- Iterative procedure:
  0. NN pretrained as a pure classifier
  1. update $\overrightarrow{\alpha}$
     - simple minimization with constraints
  2. update NN parameters
     - $\ell(\theta) = e_s - \log\left|\det(H)\right| + \ell_g$

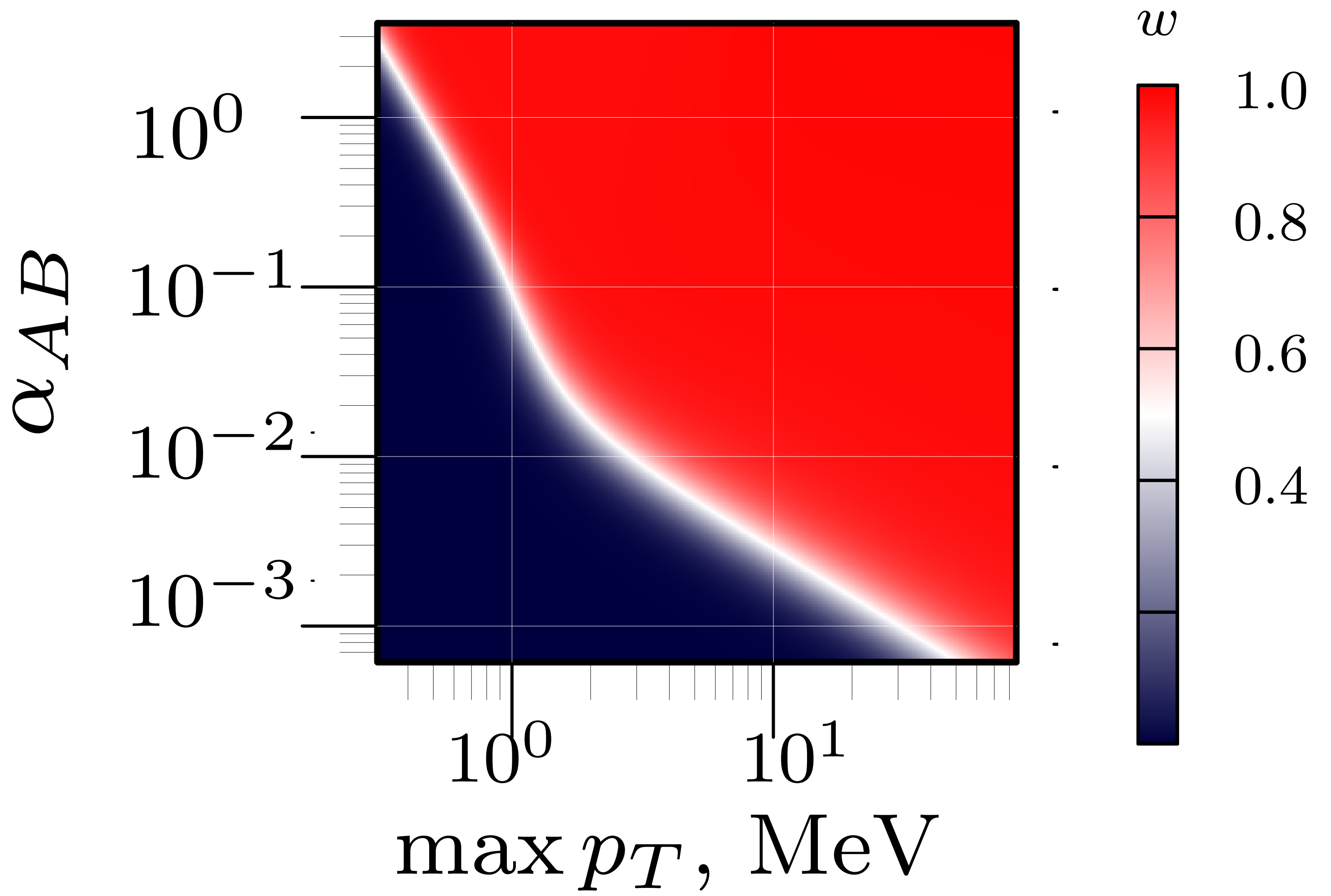
keeps matrix invertible

regulariser

$$\ell_g(\theta) = \sum\left[\max\left(\frac{||\nabla \overrightarrow{h}(x_k,\theta)||}{p} - 1{,}0\right)\right]^2$$



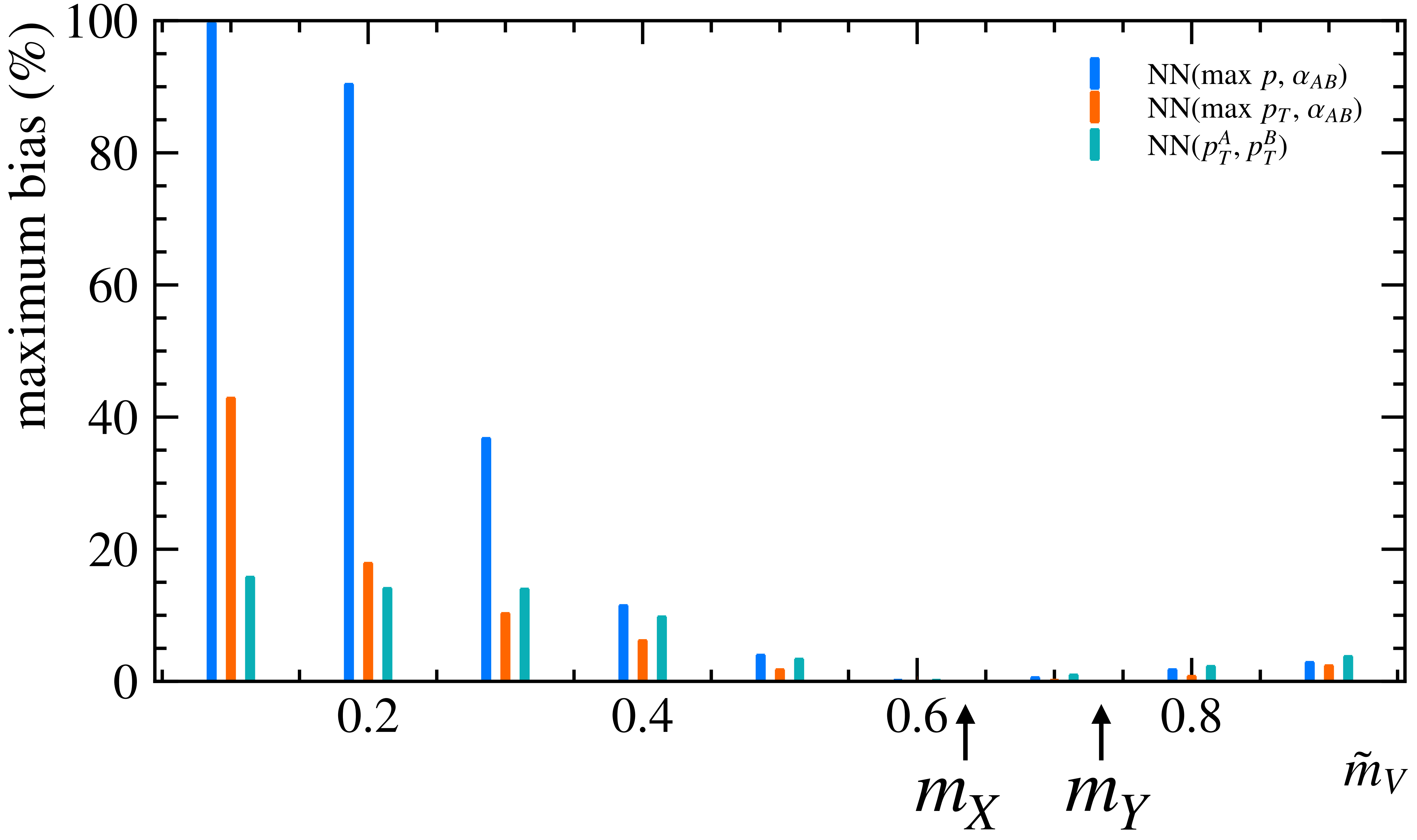




A simple example: results

- Target measurement of $\mathcal{B}(P \to VC)$ as function of m_V
- Control channels:

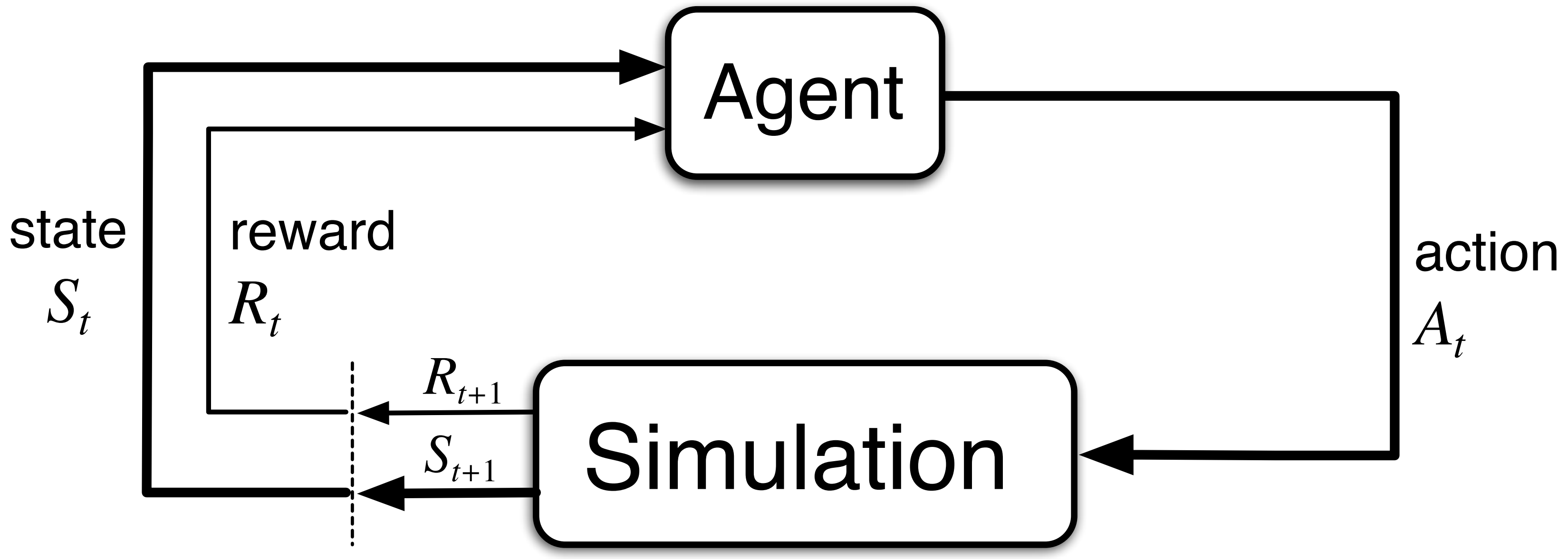
$$\mathcal{B}(P \to XC) \propto e_{P\to XC} \in [-3\%, 3\%]\,,$$
$$\frac{\mathcal{B}(P \to YC)}{\mathcal{B}(P \to XC)} \propto \frac{e_{B\to YC}}{e_{P\to XC}} \in [-1\%, 1\%]$$



- As expected, maximum allowed bias depends on the mass (kinematic overlap) between signal and control channels → But quantifiable now!

Going low level…

- So far, only considered reconstructed quantities (high levels)
- However, everything that happens in the detector happens at low level
  - Hits, energy deposit, material interaction, etc.
- MC simulation cannot be described in a parametric way
  - Requires a different formulation of the problem
  - Interactive tuning of the simulation → RL ?
    - Tested (with high level quantities) on an other example of flavour physics (angular analysis of rare *B* decay)



Conclusions

- Presented method to systematically investigate potentially hidden systematics
- Tested on a simple example
- Fully general: can be extended to any measurement that relies on simulation!

Thank you!