Modern Machine Learning for Monte Carlo Event Generators

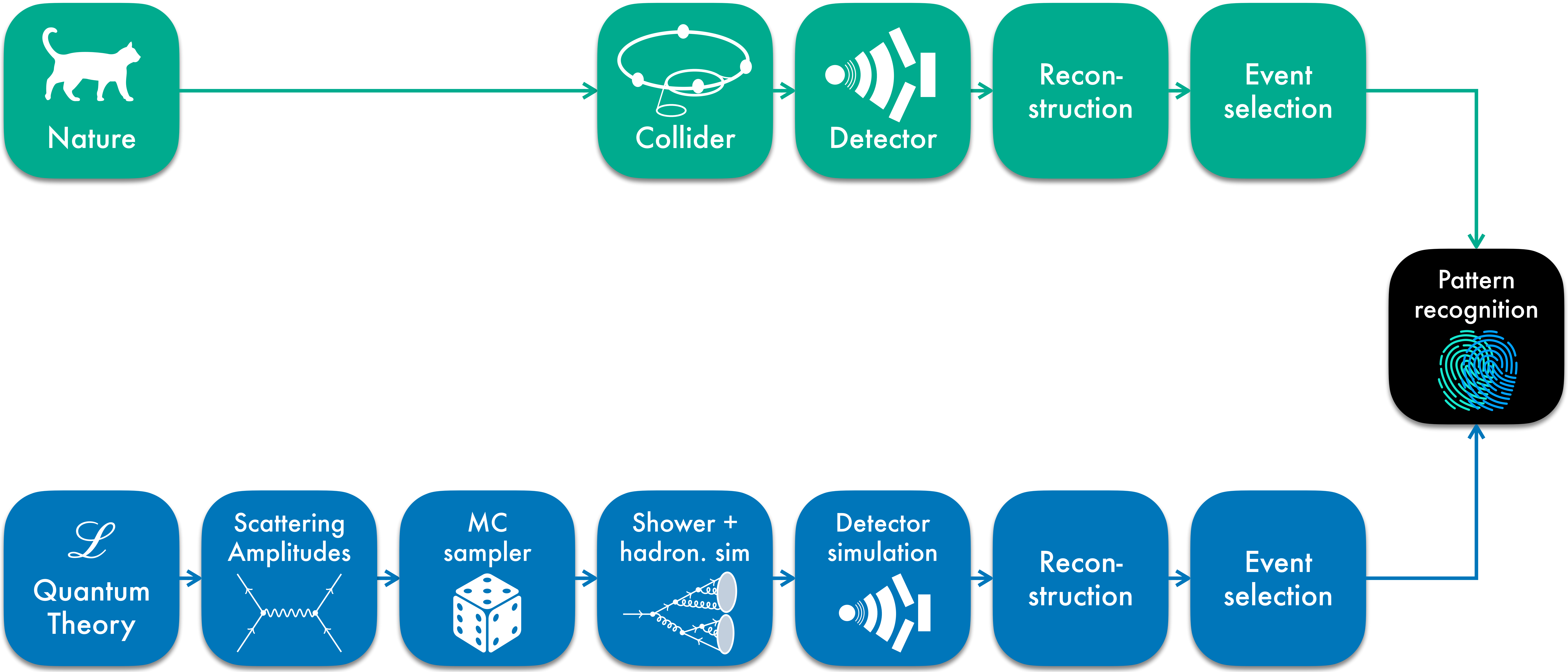
UNIVERSITÀ DEGLI STUDI DI MILANO

INFN MILANO

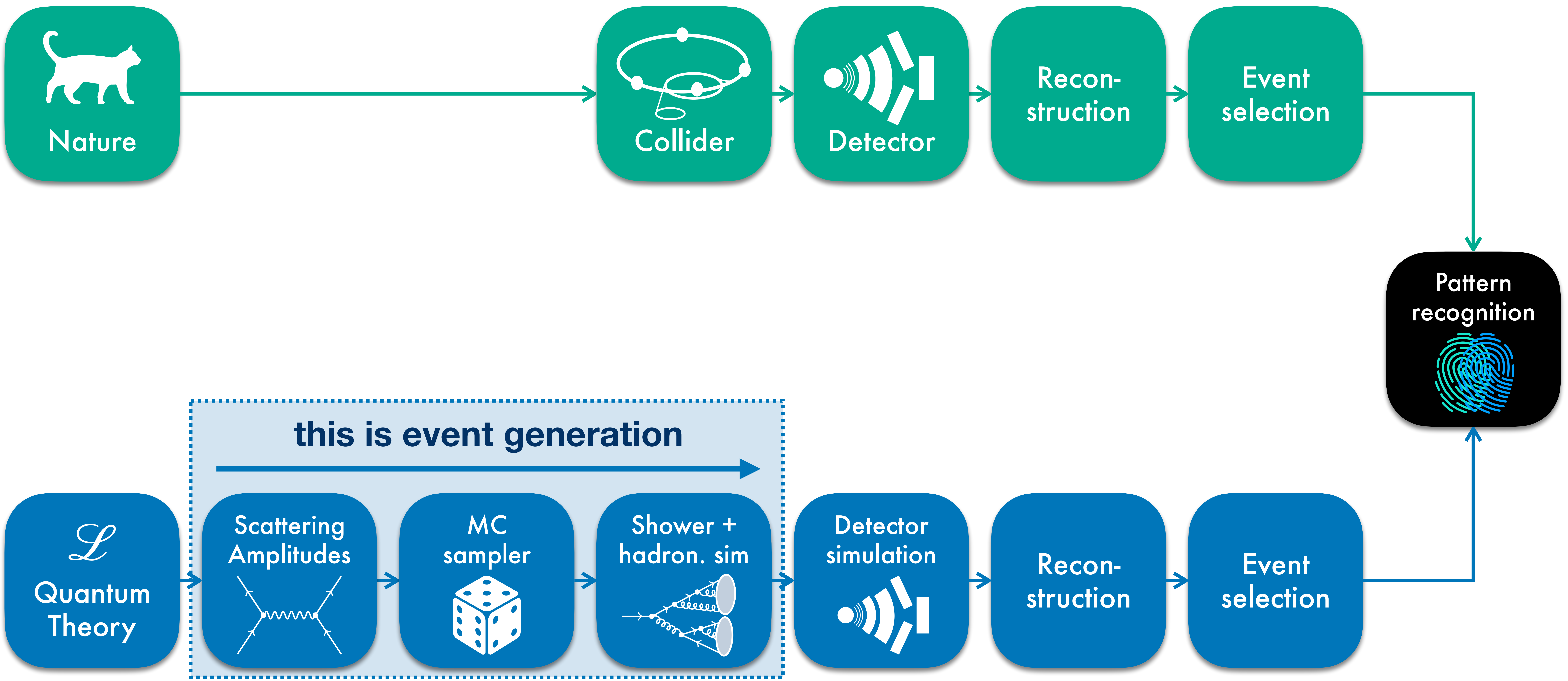
NPTwins 2025 | Messina University

Ramon Winterhalder

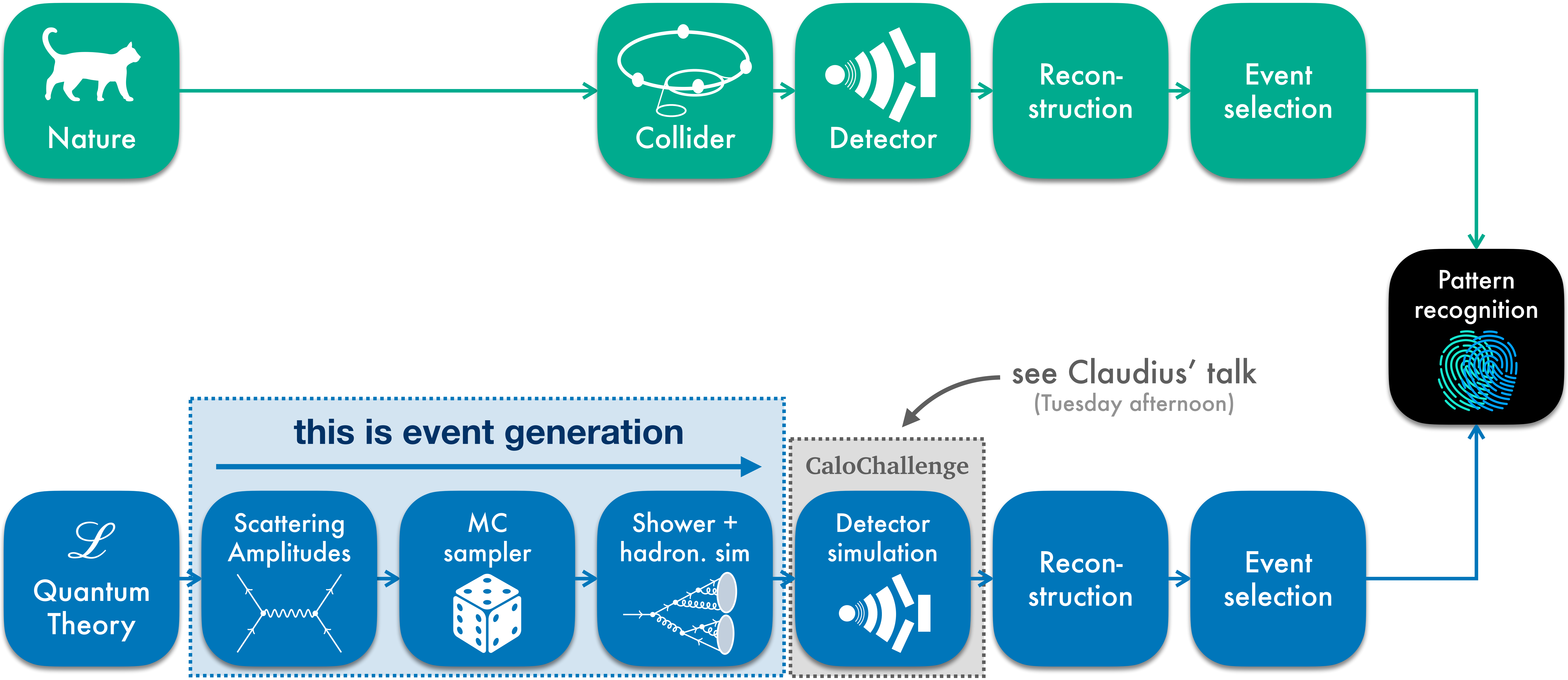
Collider analysis in a nutshell



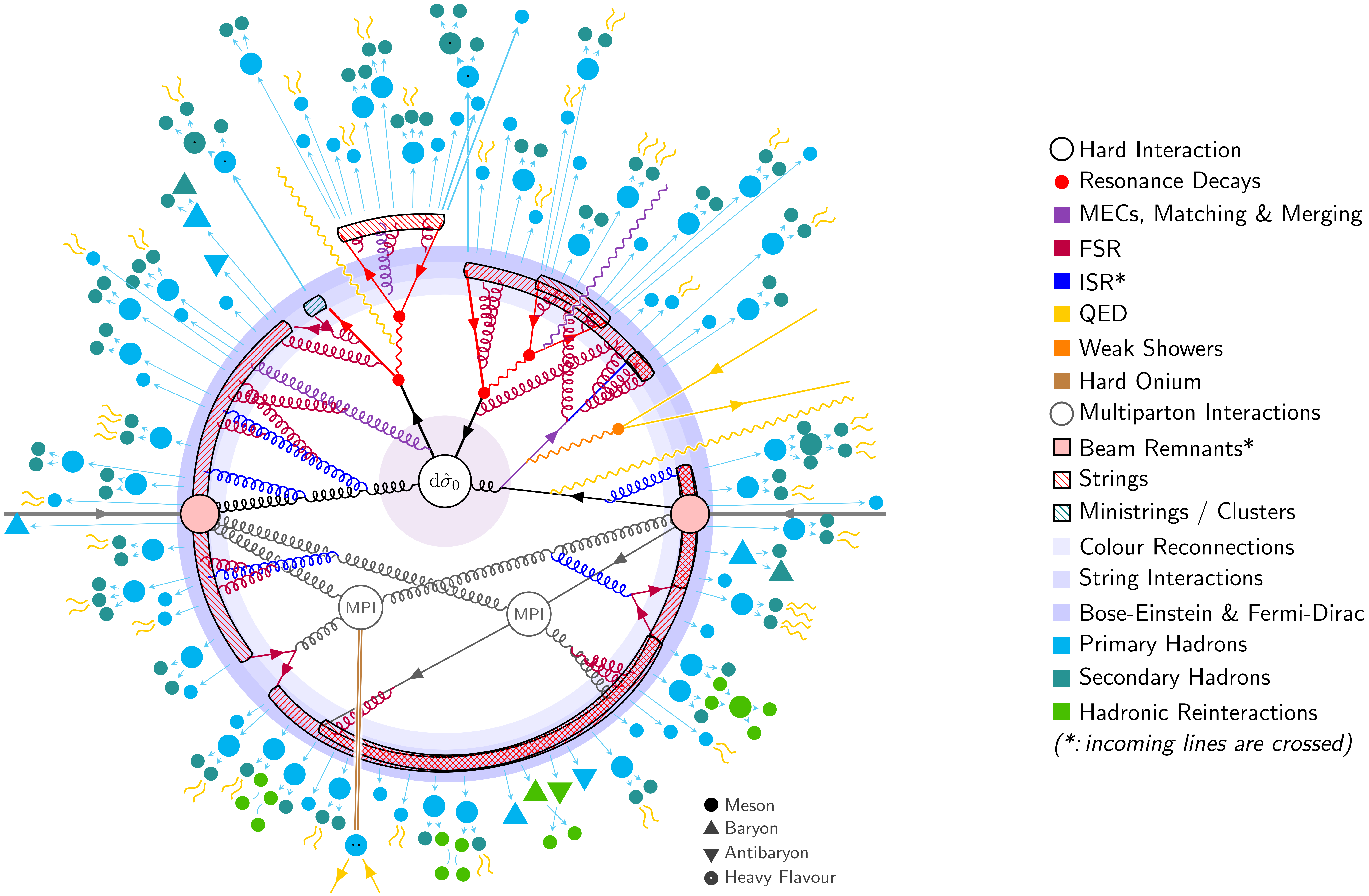
Collider analysis in a nutshell



Collider analysis in a nutshell

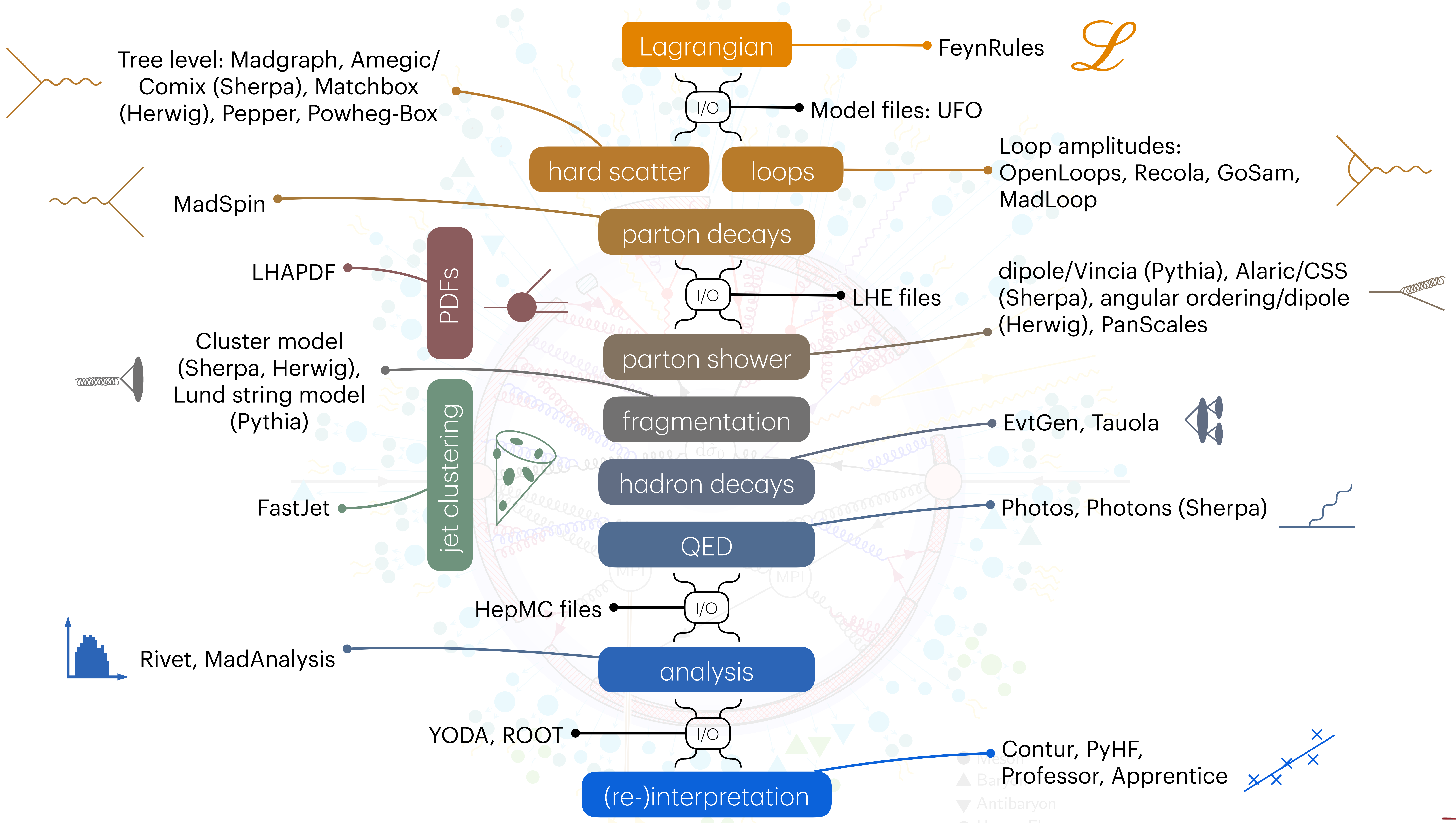


The devil is in the details

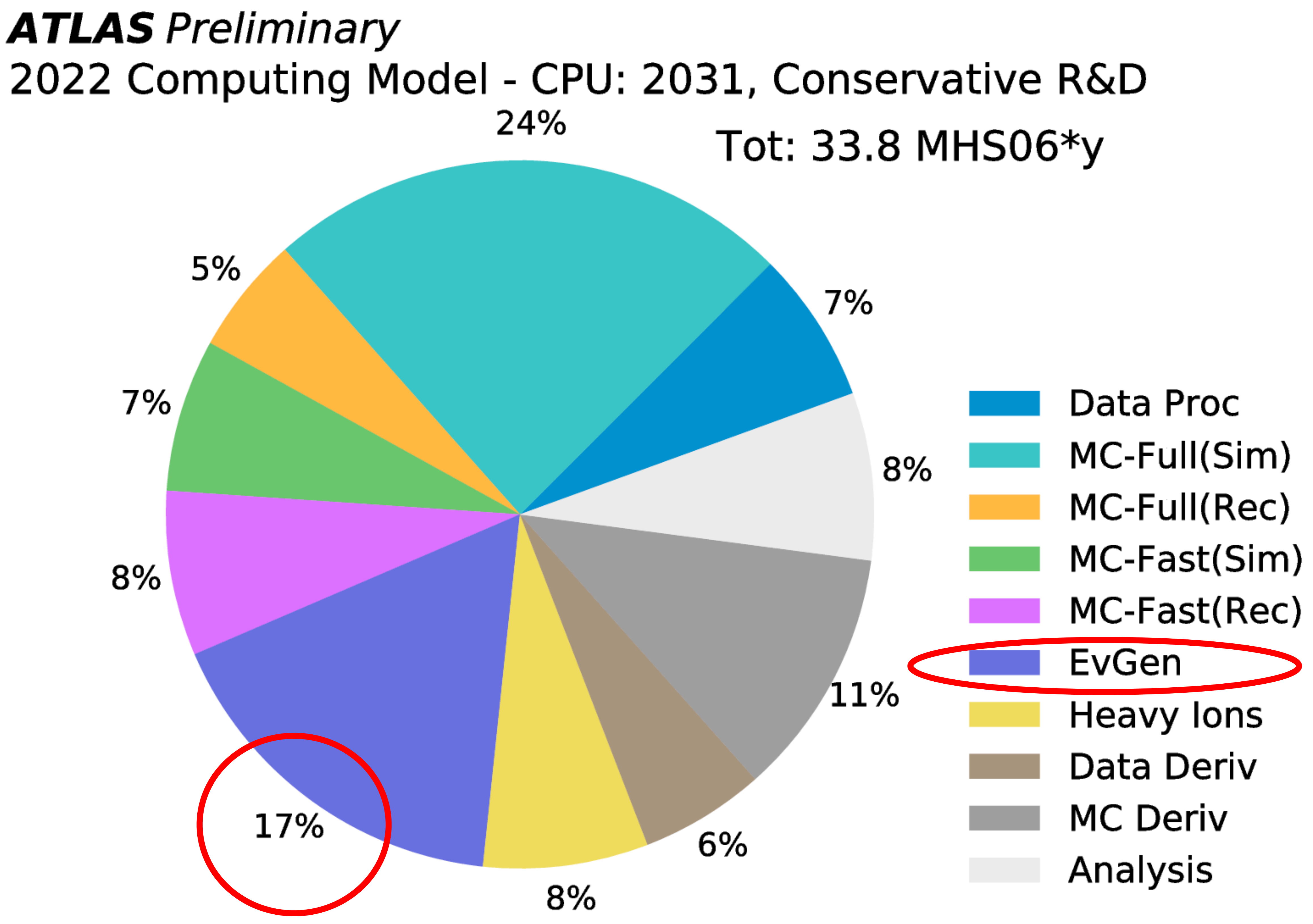


Pythia8 Manual [2203.11601]

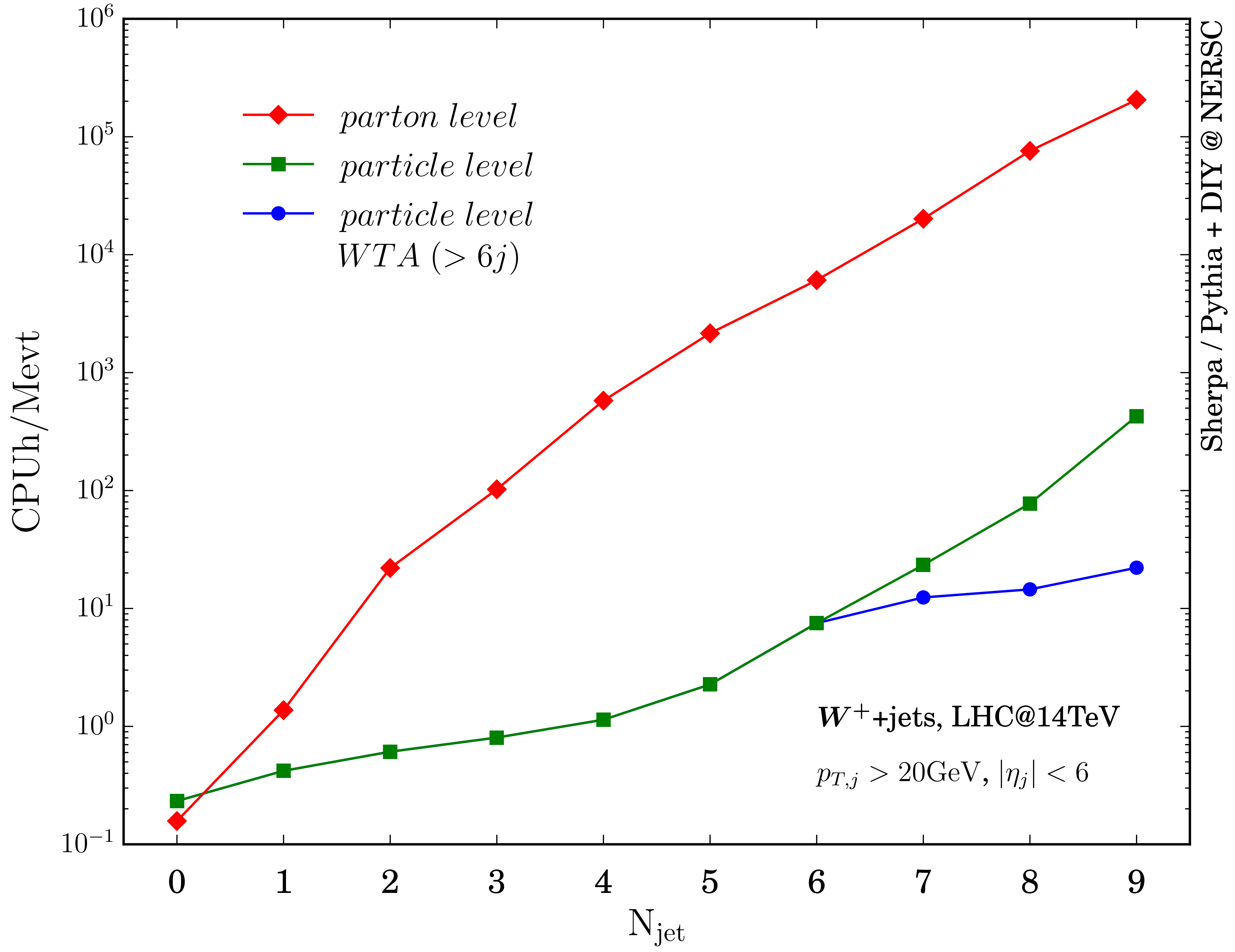
The devil is in the details



Computing Budget

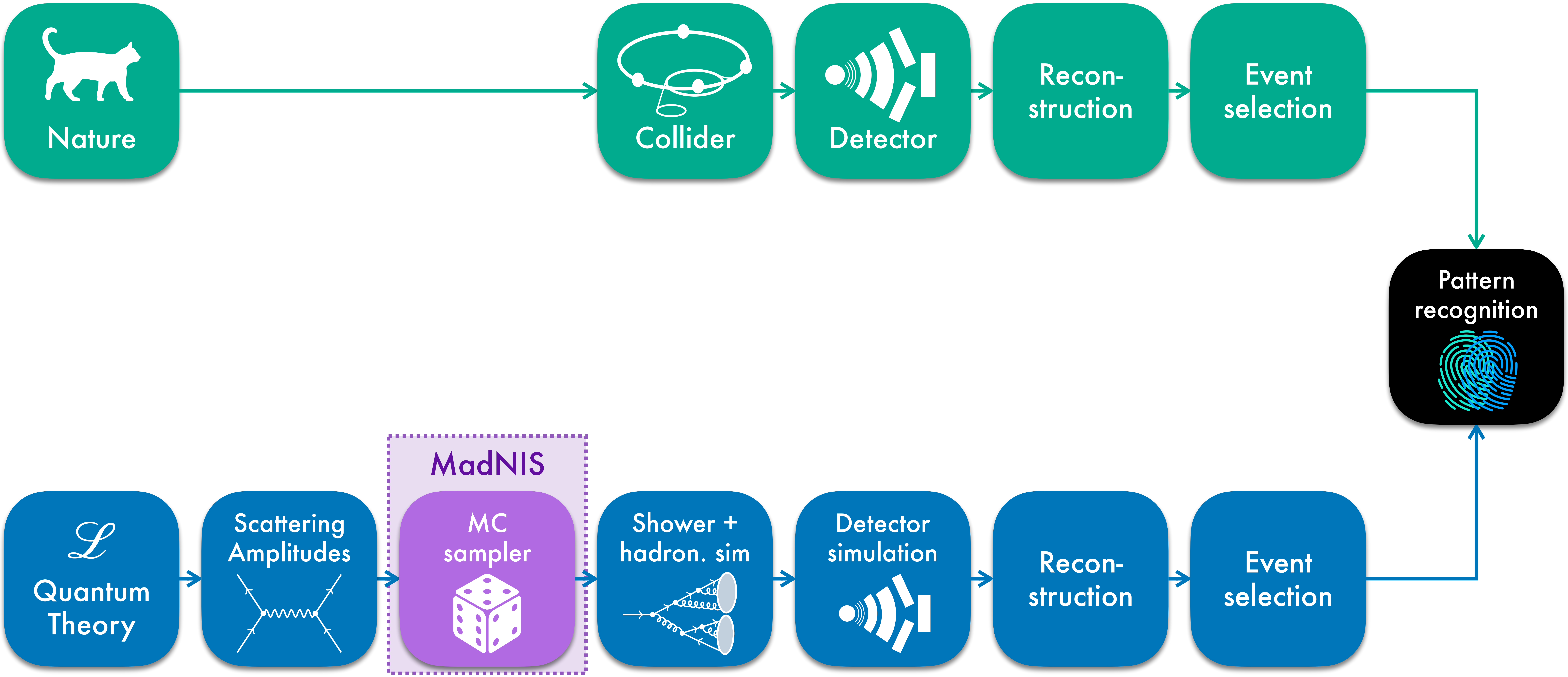


[CERN-LHCC-2022-005]



[Höche *et al.*, 1905.05120]

MadNIS — Neural Importance Sampling

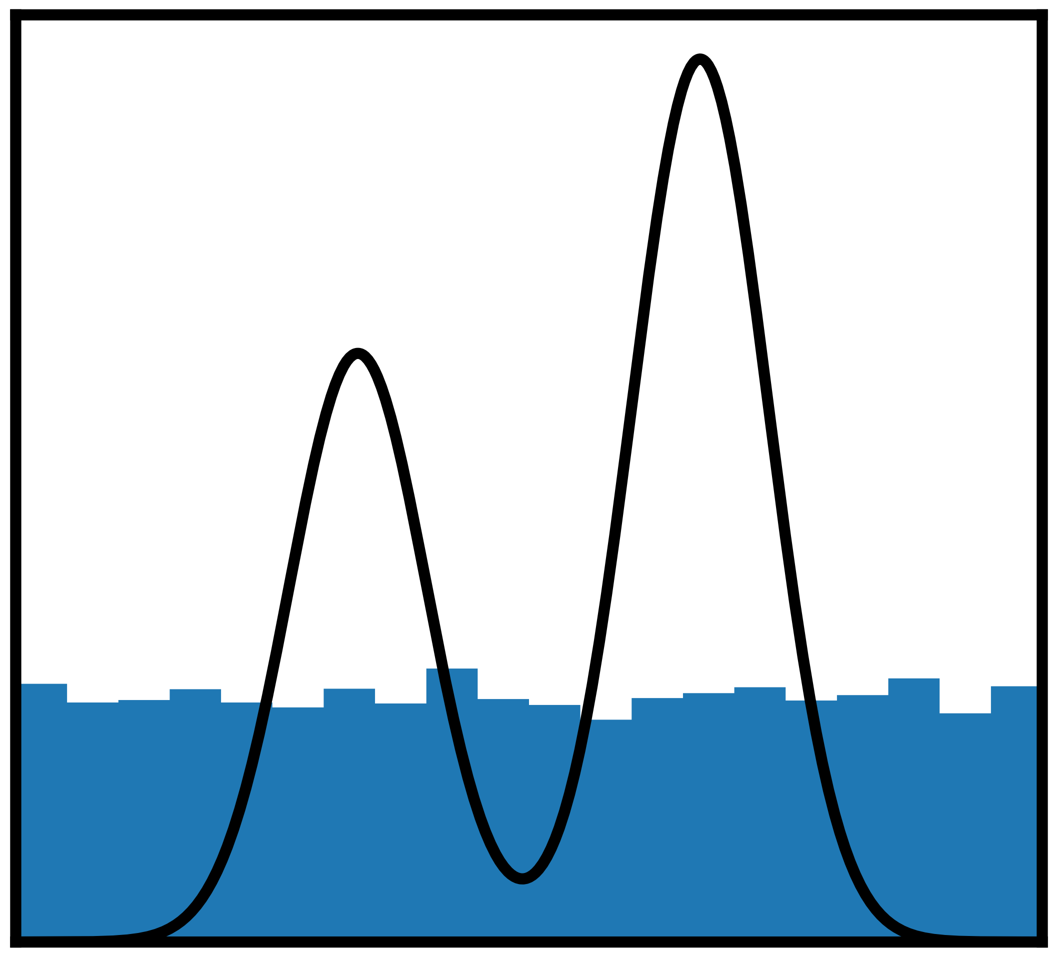


MadNIS [2212.06172, 2311.01548, 2408.01486]

Monte Carlo integration

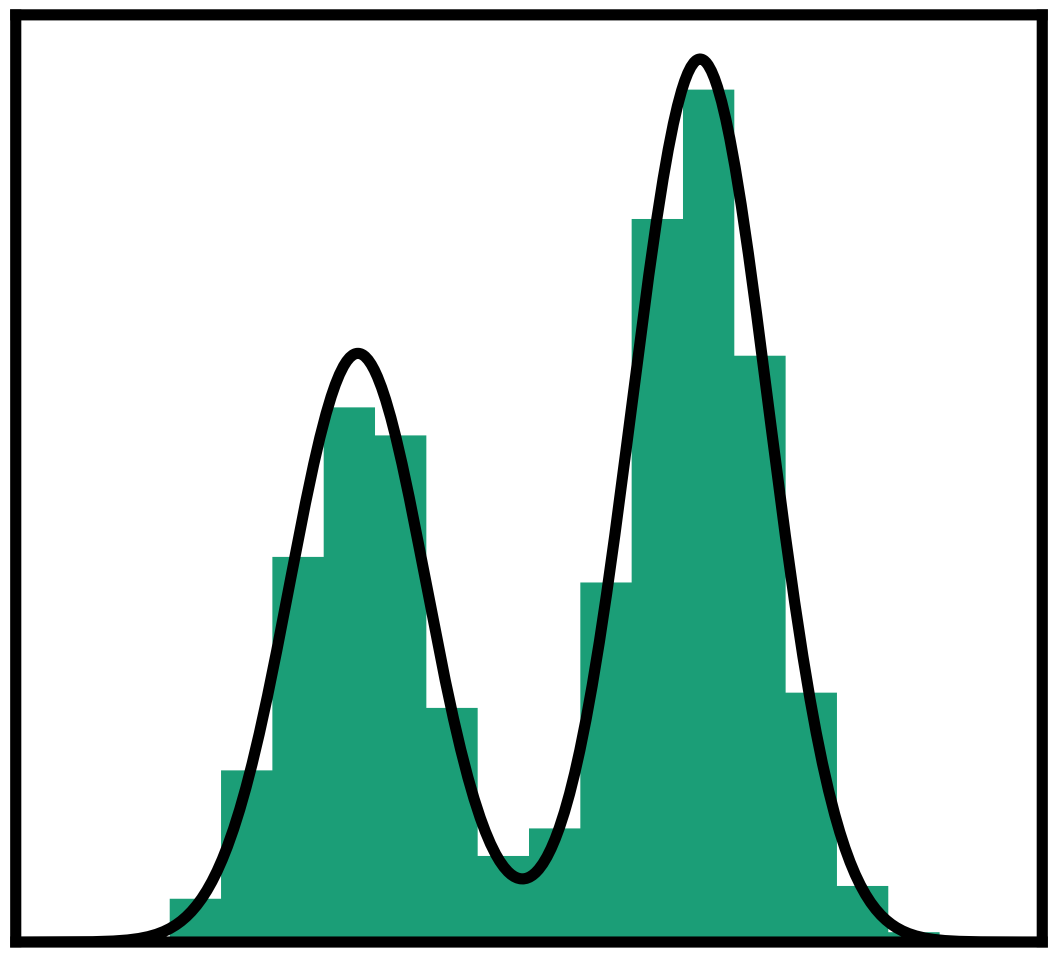
Calculate (differential) cross sections

$$d\sigma = \frac{1}{\text{flux}} dx_a dx_b \, f(x_a) f(x_b) \, d\Phi_n \left\langle \, |M_{\lambda,c,\ldots}(p_a, p_b \,|\, p_1, \ldots, p_n)|^2 \, \right\rangle$$



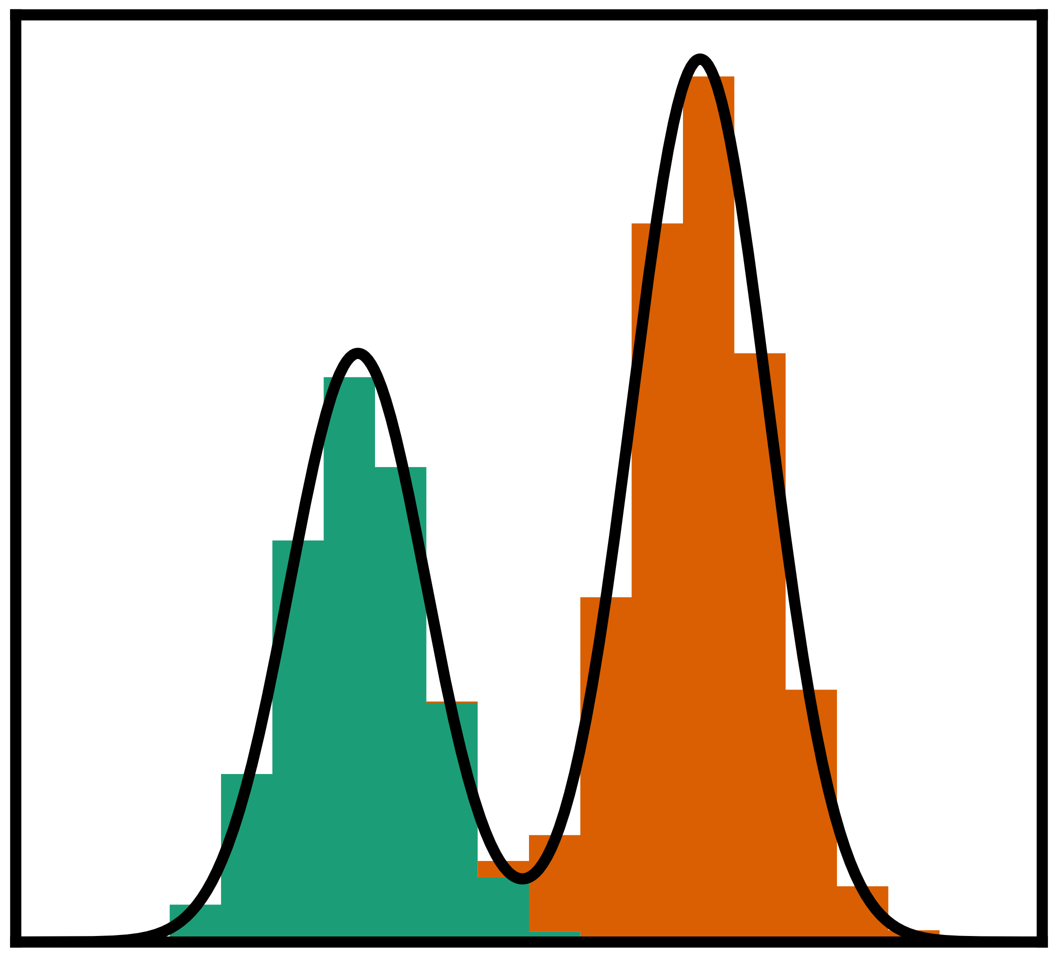
Flat sampling:
inefficient

$$I = \langle f(x) \rangle_{x \sim \text{unif}}$$



Importance sampling:
find p close to f

$$I = \left\langle \frac{f(x)}{p(x)} \right\rangle_{x \sim p(x)}$$



Multi-channel:
one map for each channel

$$I = \sum_i \left\langle \alpha_i(x) \frac{f(x)}{p_i(x)} \right\rangle_{x \sim p_i(x)}$$

Event generation in MadGraph

Calculate (differential) cross sections

$$d\sigma = \frac{1}{\text{flux}} dx_a dx_b\, f(x_a) f(x_b)\, d\Phi_n \left\langle\, |M_{\lambda,c,\ldots}(p_a, p_b \mid p_1, \ldots, p_n)|^2 \,\right\rangle$$

$$I = \sum_i \left\langle \alpha_i(x)\, \frac{f(x)}{p_i(x)} \right\rangle_{x \sim p_i(x)}$$

Sum over channels

MadGraph: build channels from Feynman diagrams

Integrand

MadGraph: $d\sigma/dx$

Channel weights

MadGraph: $\alpha_i^{\mathrm{MG}}(x) \sim |M_i|^2$

Channel mappings

MadGraph: use amplitude structure, …

Analytic mappings + refine with **VEGAS** (factorized, histogram based importance sampling)

Event generation in MadNIS

MadNIS

Calculate (differential) cross sections

$$d\sigma = \frac{1}{\text{flux}} dx_a dx_b\, f(x_a) f(x_b)\, d\Phi_n \left\langle\, |M_{\lambda,c,\dots}(p_a, p_b \,|\, p_1, \dots, p_n)|^2 \,\right\rangle$$

Sum over channels

MadGraph: build channels from Feynman diagrams

Integrand

MadGraph: $d\sigma/dx$

$$I = \sum_i \left\langle \alpha_i(x) \frac{f(x)}{p_i^\theta(x)} \right\rangle_{x \sim p_i^\theta(x)}$$

Channel weights

MadGraph: $\alpha_i^{\text{MG}}(x) \sim |M_i|^2$

Learned channel mappings

MadGraph: use amplitude structure, …

Analytic mappings + ~~refine with VEGAS~~

refine with **NF**

Event generation in MadNIS

Calculate (differential) cross sections

$$d\sigma = \frac{1}{\text{flux}} dx_a dx_b\, f(x_a) f(x_b)\, d\Phi_n \left\langle\, |M_{\lambda,c,\ldots}(p_a, p_b \,|\, p_1, \ldots, p_n)|^2 \,\right\rangle$$

MadNIS

Sum over channels

MadGraph: build channels
from Feynman diagrams

Integrand

MadGraph: $d\sigma/dx$

$$I = \sum_i \left\langle \alpha_i^{\xi}(x) \frac{f(x)}{p_i^{\theta}(x)} \right\rangle_{x \sim p_i^{\theta}(x)}$$

Learned channel weights

MadGraph: $\alpha_i^{\text{MG}}(x) \sim |M_i|^2$

$$\alpha_i(x) \rightarrow \alpha_i^{\xi}(x) = \alpha_i^{\text{MG}}(x) \cdot K_i^{\xi}(x)$$

parametrize with **NN**

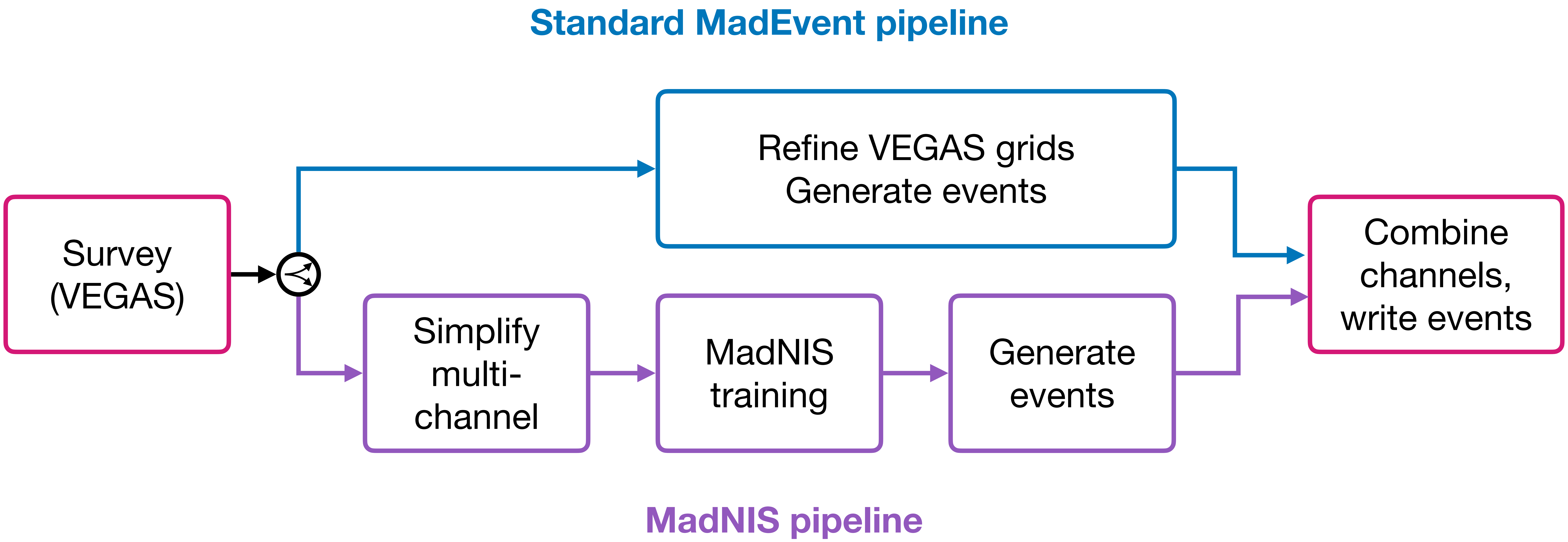
Learned channel mappings

MadGraph: use amplitude structure, …

Analytic mappings + ~~refine with Vegas~~

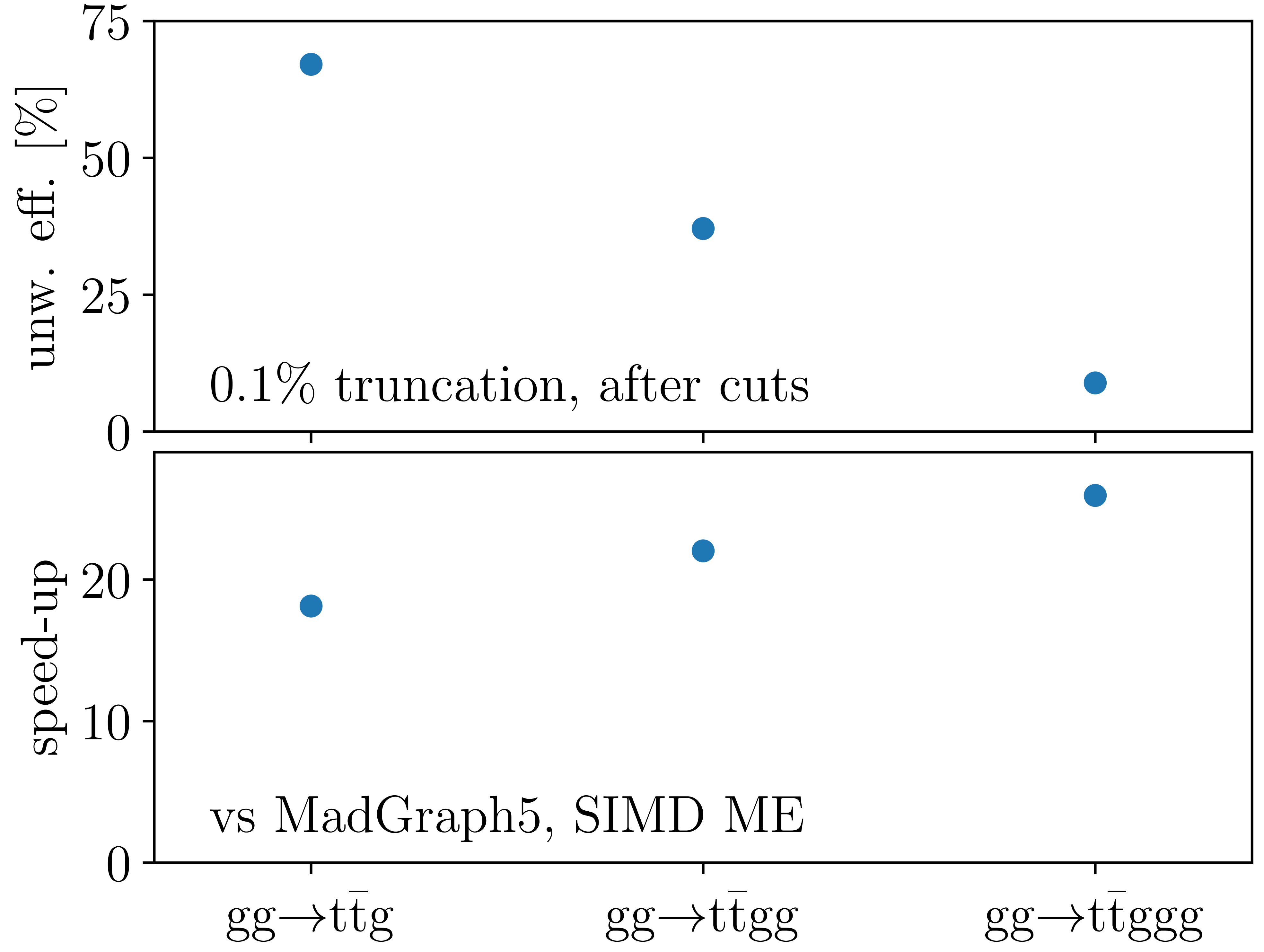
refine with **NF**

Building MadNIS into MadGraph



Unweighting performance

Unweighting efficiency



Improved unweighting efficiency

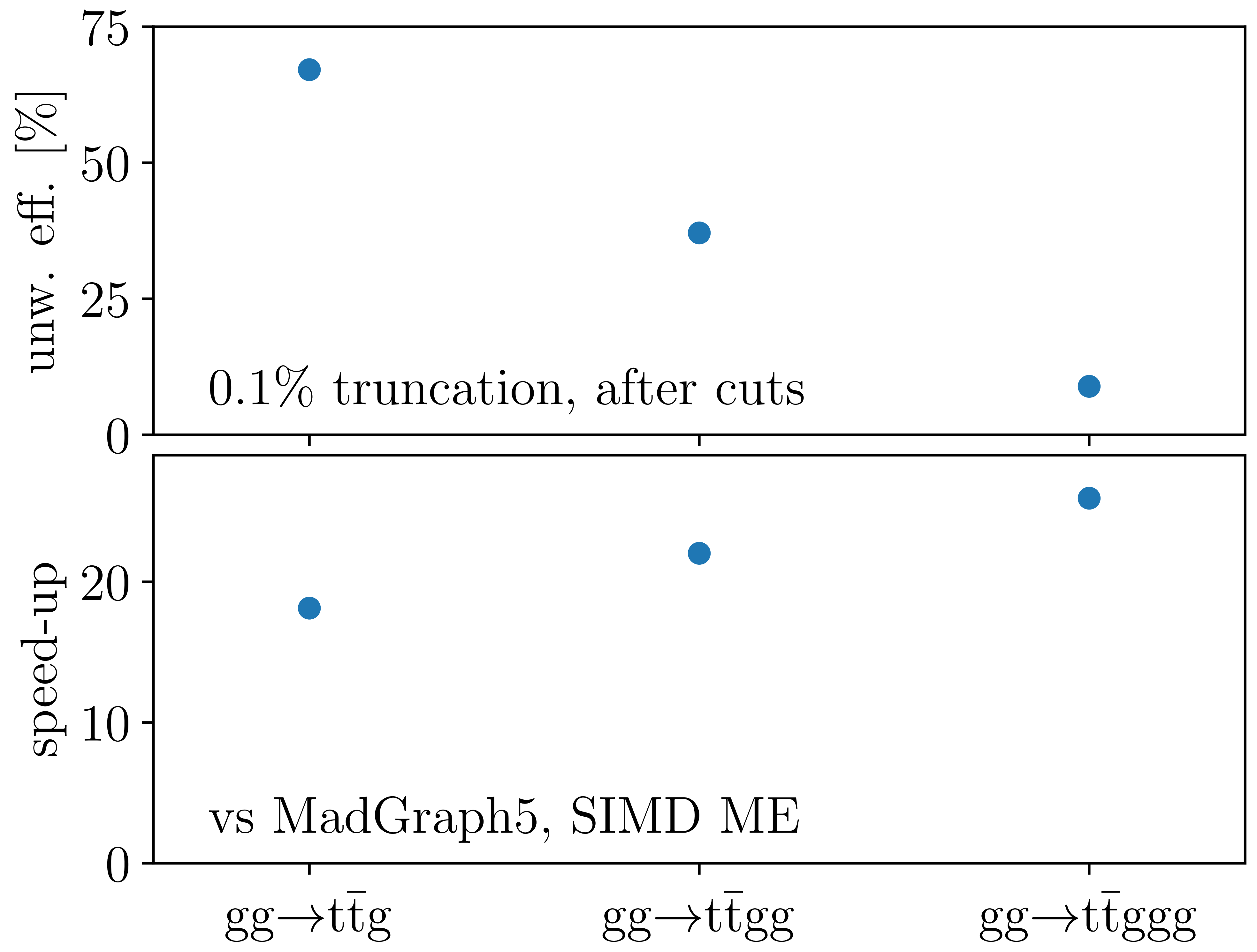
$$\epsilon^{\mathrm{MadNIS}} \Big/ \epsilon^{\mathrm{MG5}} \approx 25$$

@ $gg \to t\bar{t}\,ggg$

(preliminary)

Unweighting performance

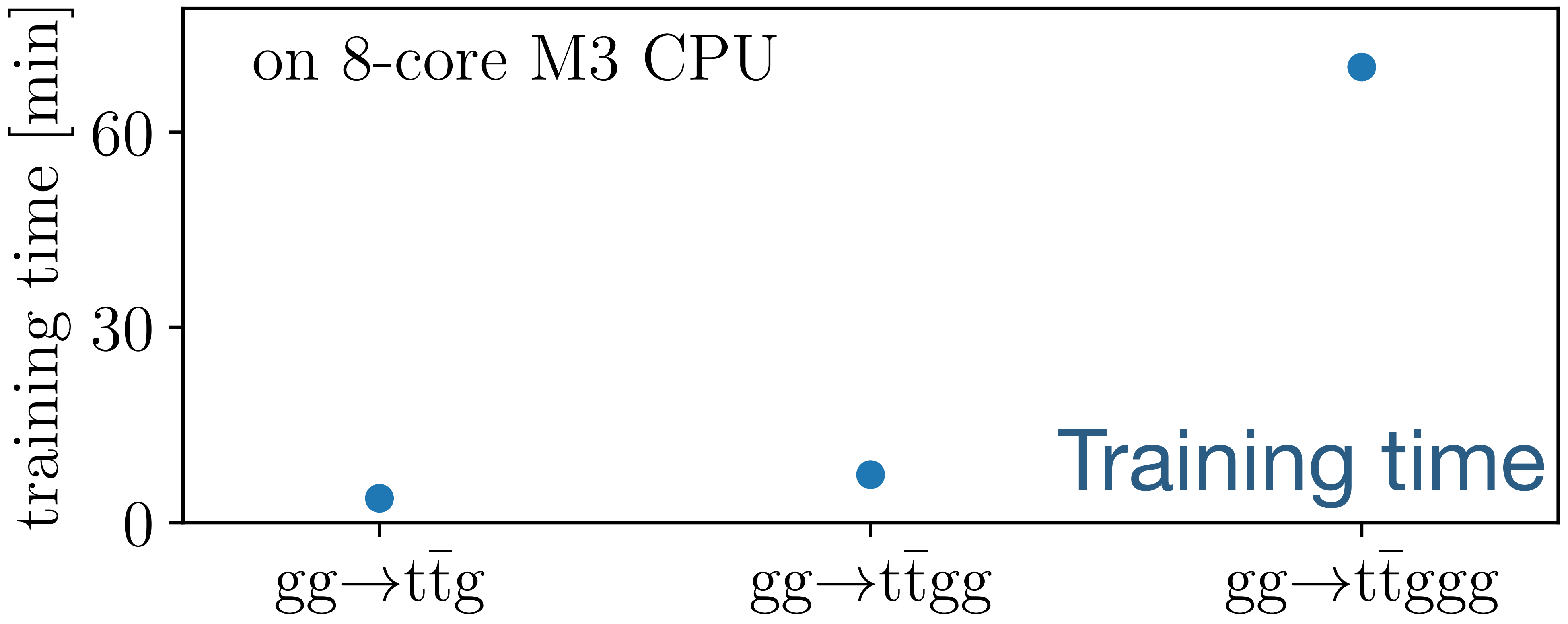
Unweighting efficiency



Improved unweighting efficiency

$$\epsilon^{\mathrm{MadNIS}} \Big/ \epsilon^{\mathrm{MG5}} \approx 25$$

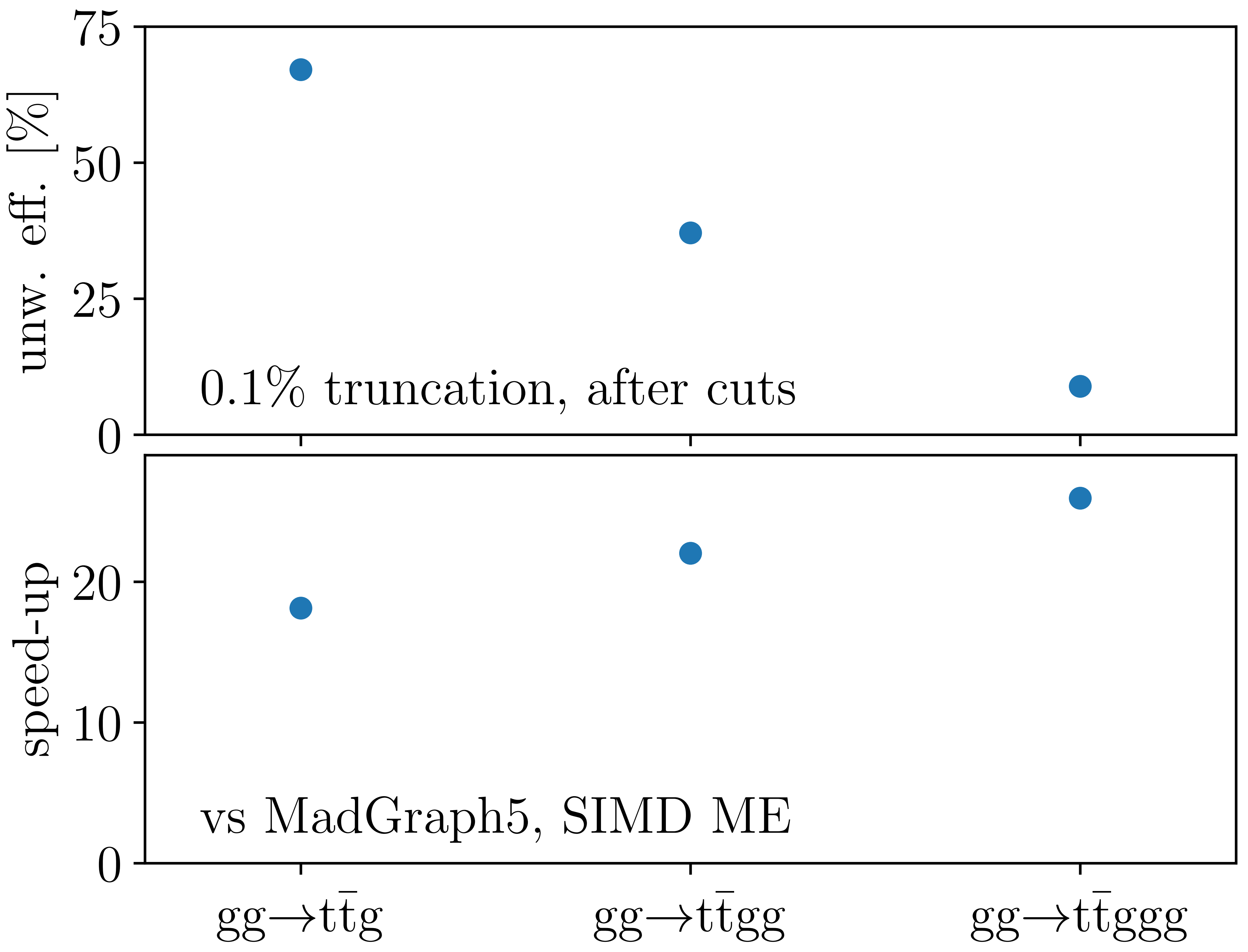
@ $gg \to t\bar{t}\, ggg$



(preliminary)

Unweighting performance

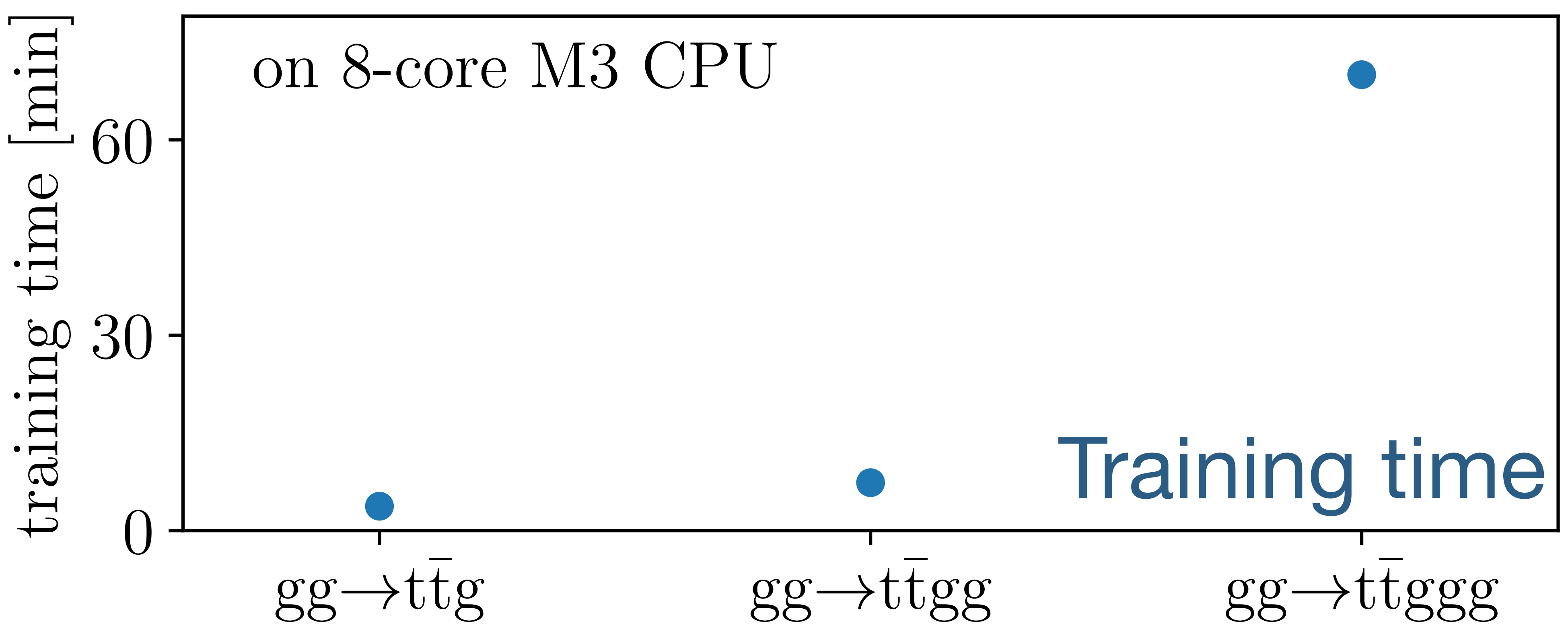
Unweighting efficiency



Improved unweighting efficiency

$$\epsilon^{\text{MadNIS}} \big/ \epsilon^{\text{MG5}} \approx 25$$

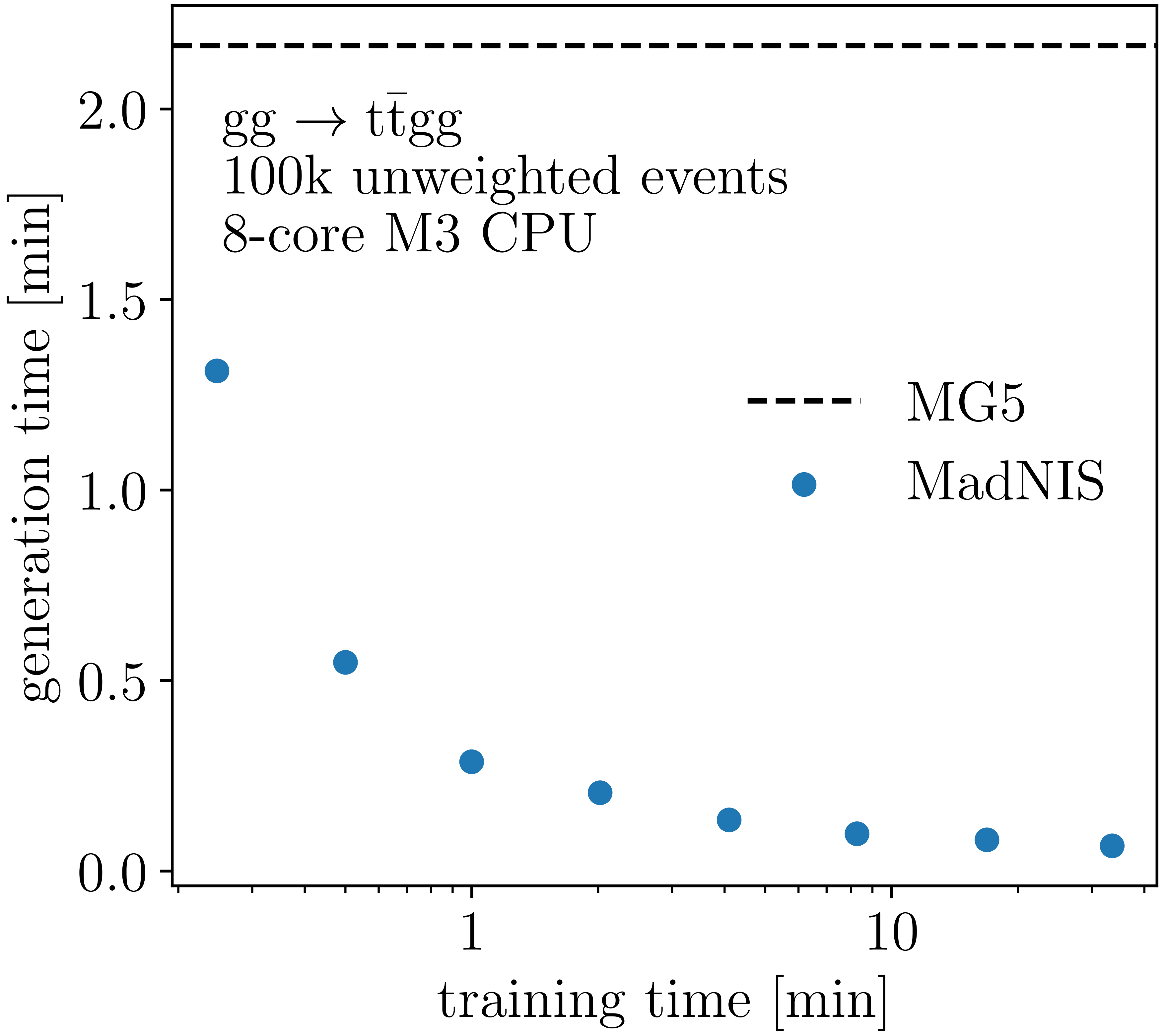
@ $gg \to t\bar{t}\,ggg$

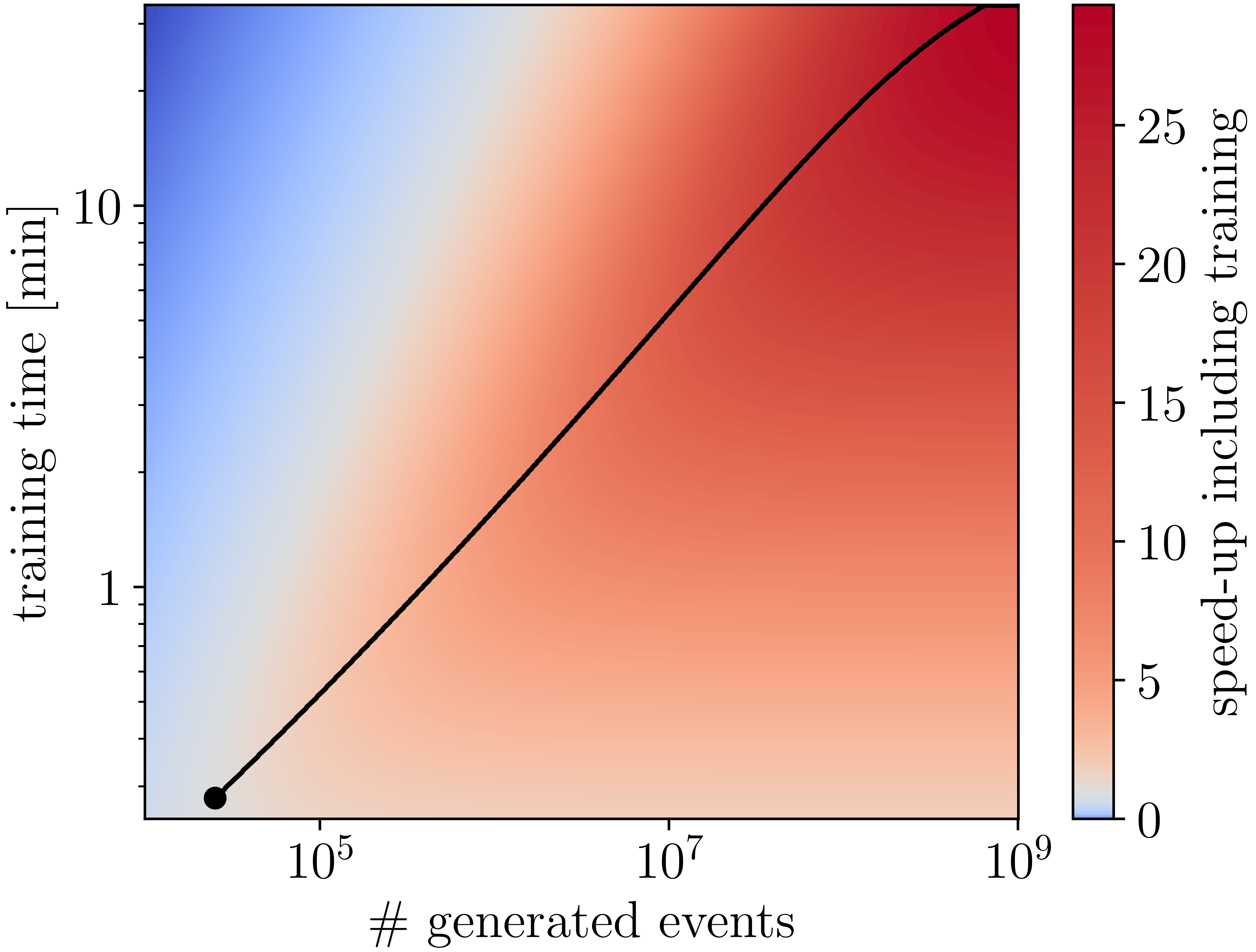


→ Does it still pay off?

(preliminary)

Training time and amortization





(preliminary)

MadNIS is faster starting at 10k events!

Standalone Python module

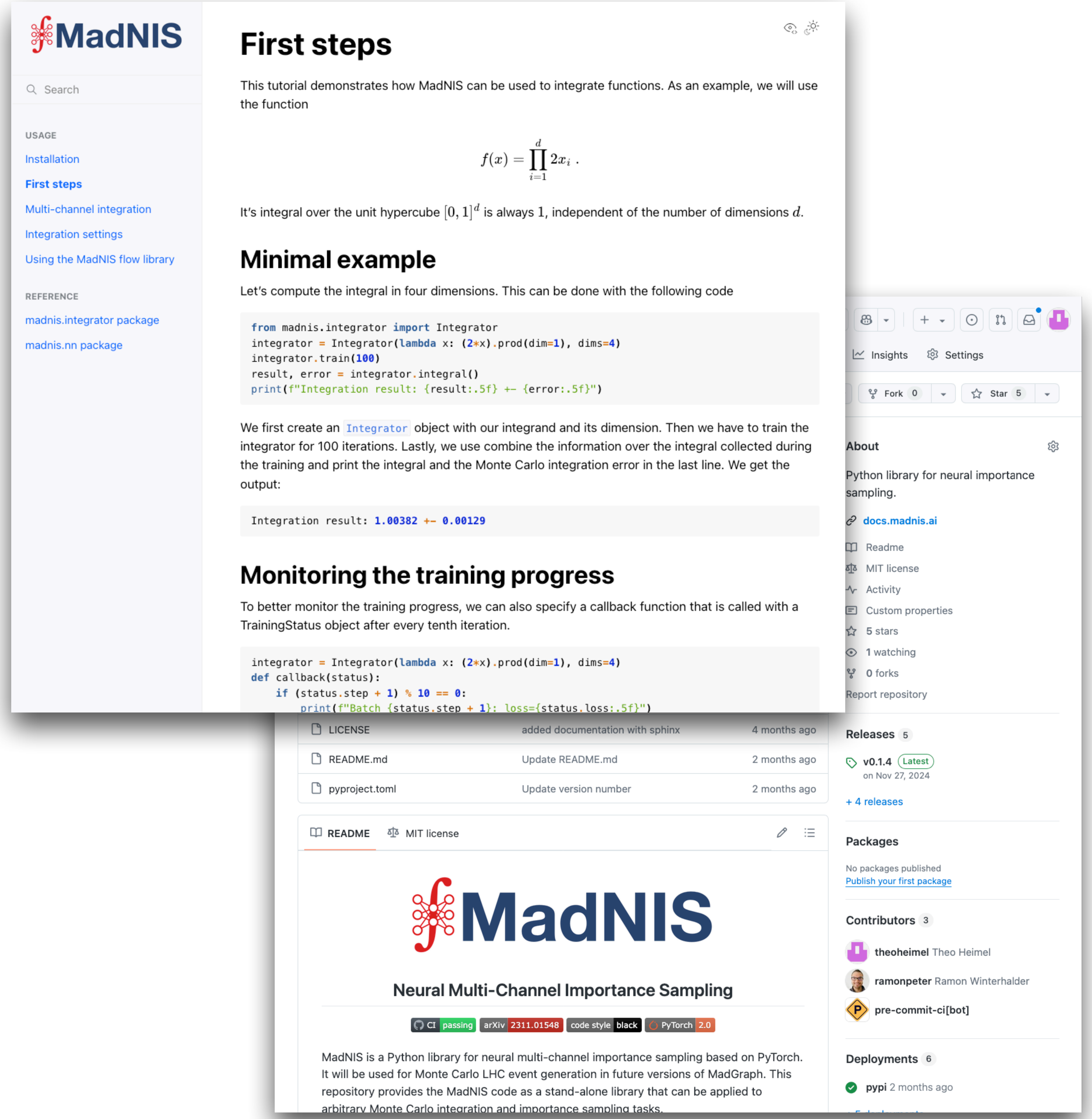
- **MadNIS** as a Python package
  → apply to your own integration tasks

- From simple single-channel integrals to complex multi-channel setups

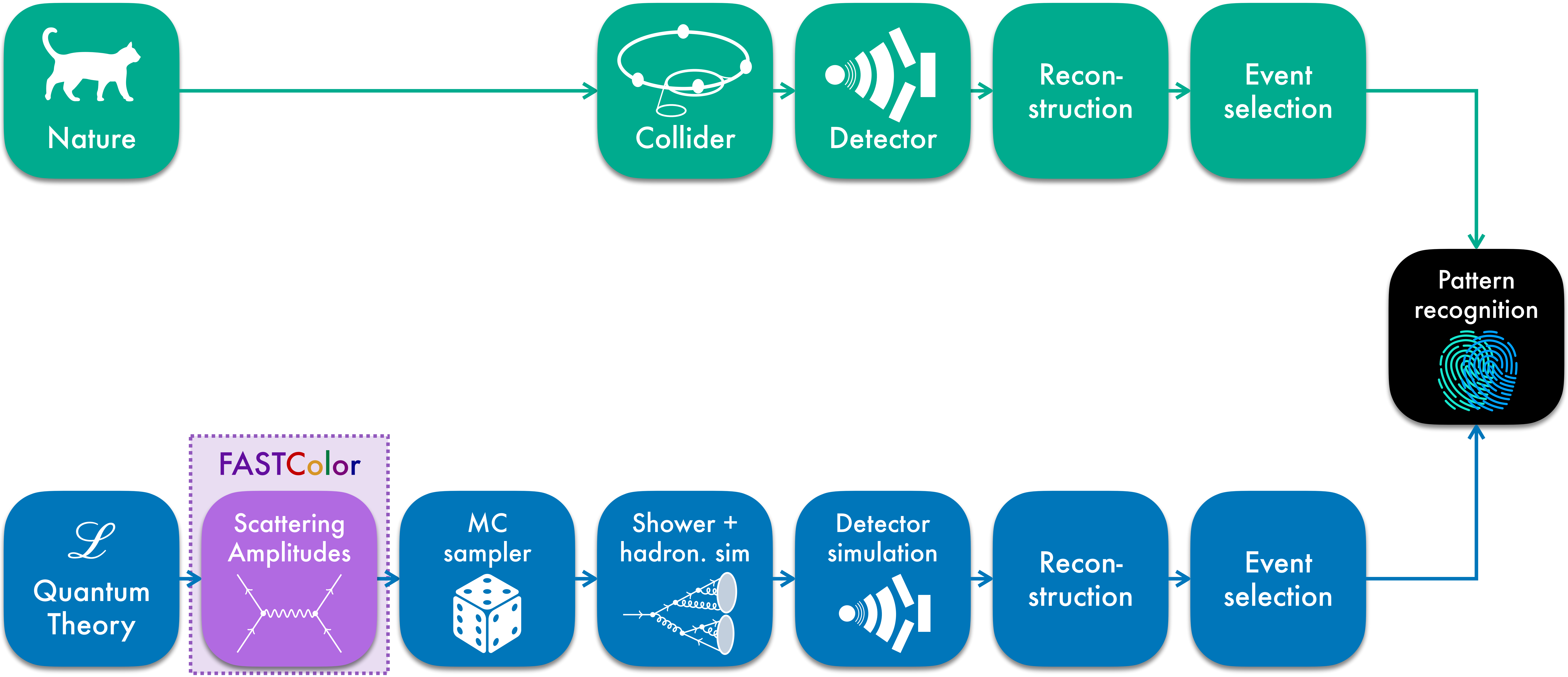
- Easy-to-use implementation of normalizing flows

https://madnis.ai/

`pip install madnis`



FASTColor — Full-color surrogates



FASTColor [2509.07068]

Setting the stage

Full-color **A**mplitude **S**urrogate **T**oolkit for QCD

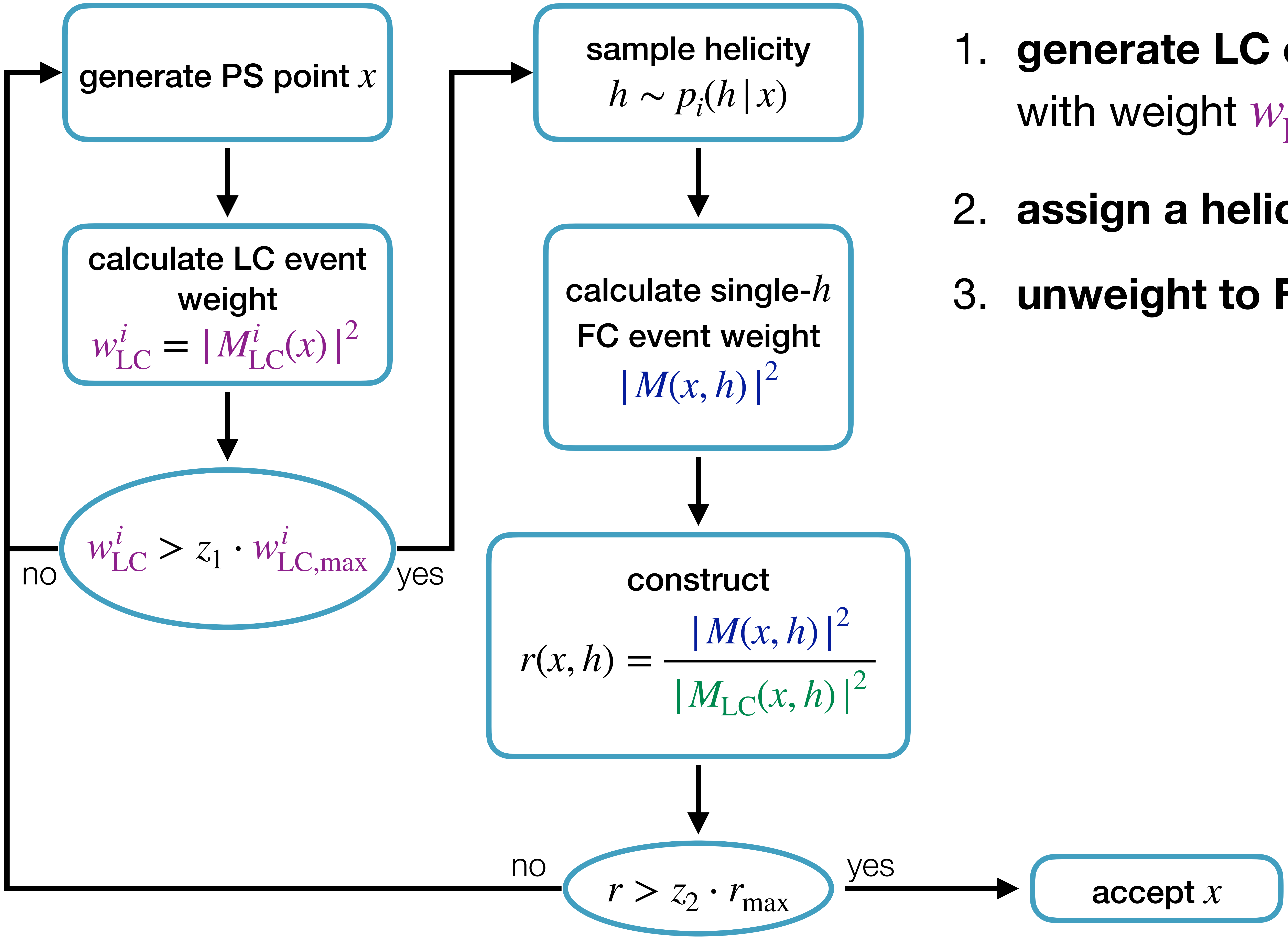
What we do

Use ML to **accelerate event generation at Full-Color (FC) accuracy** in QCD

How we do it

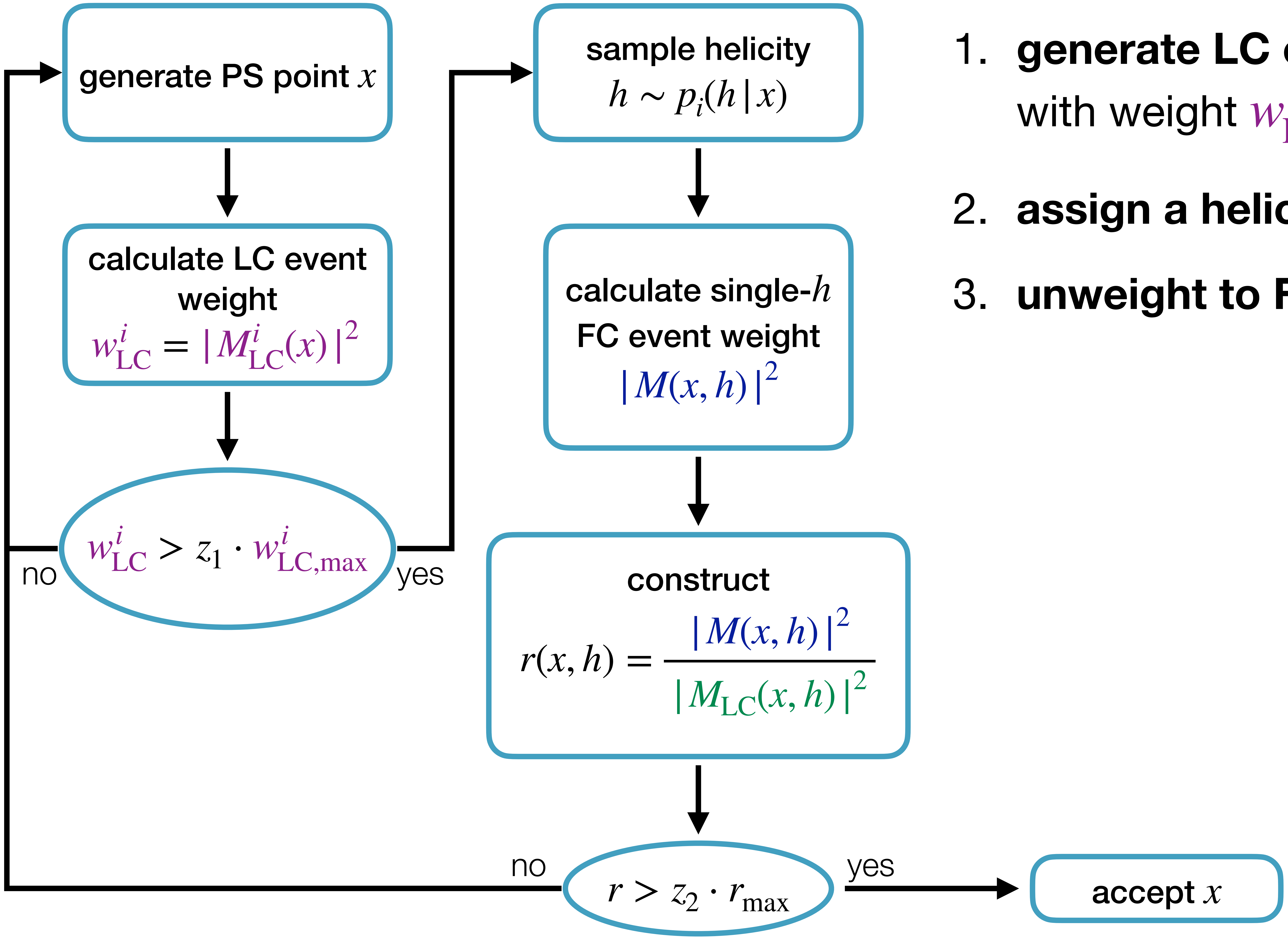
We **build on a Leading-Color (LC) based, two-step unweighting** approach

LC-based unweighting



1. **generate LC events** for a single color flow i with weight w^i_{LC}
2. **assign a helicity h** with probability $p_i(h\,|\,x)$
3. **unweight to FC**

LC-based unweighting



1. **generate LC events** for a single color flow i with weight w^i_{LC}
2. **assign a helicity h** with probability $p_i(h|x)$
3. **unweight to FC**

Unbiased result

Generated events follow FC density

$$\sigma_{\mathrm{FC}} = \sum_i \int \mathrm{d}\Phi \; |M^i_{\mathrm{LC}}(x)|^2 \langle r(x,h) \rangle_{h \sim p_i(h|x)}$$

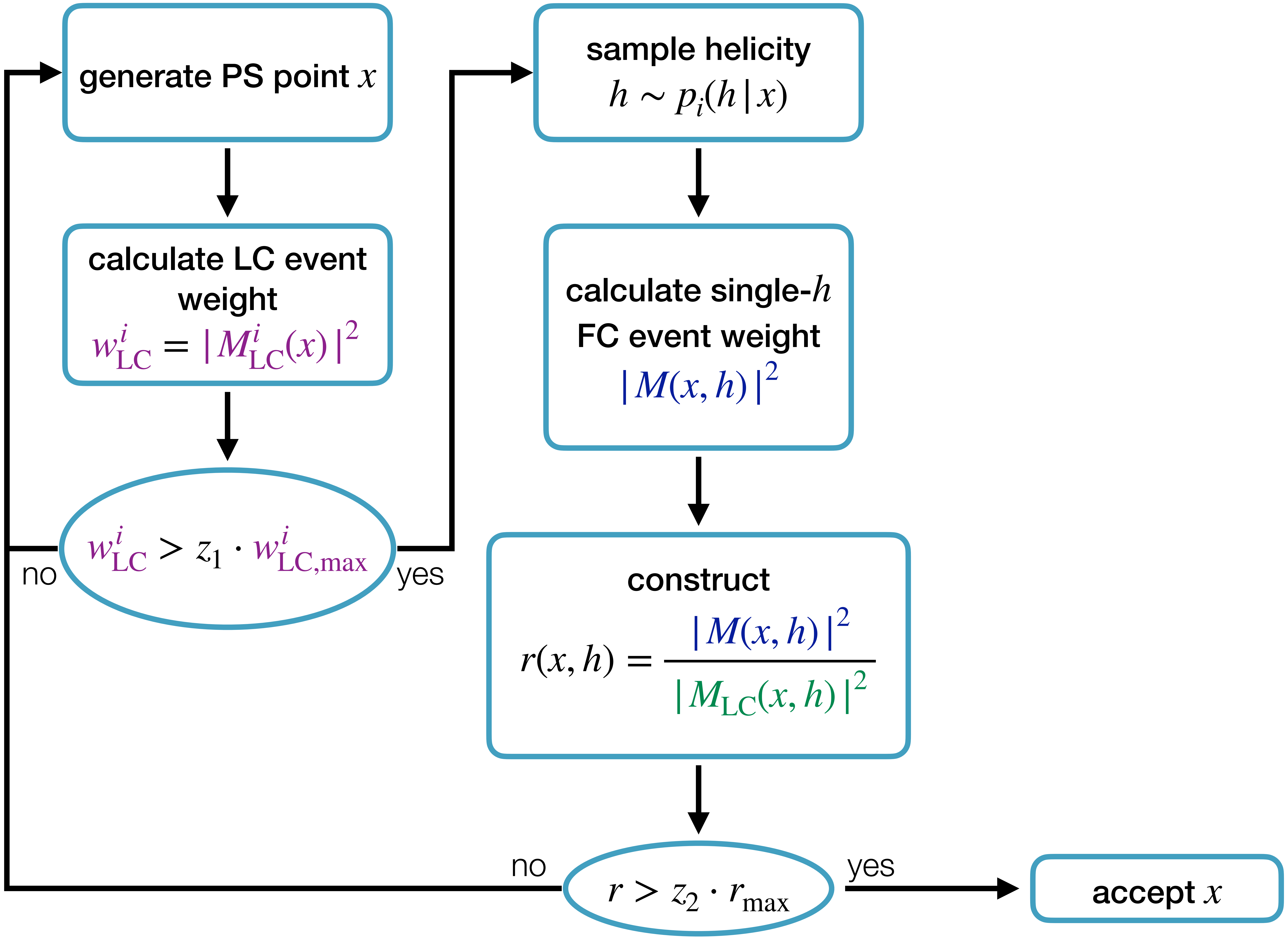
Surrogate-based unweighting

Introduce **additional unweighting step** against ML surrogate

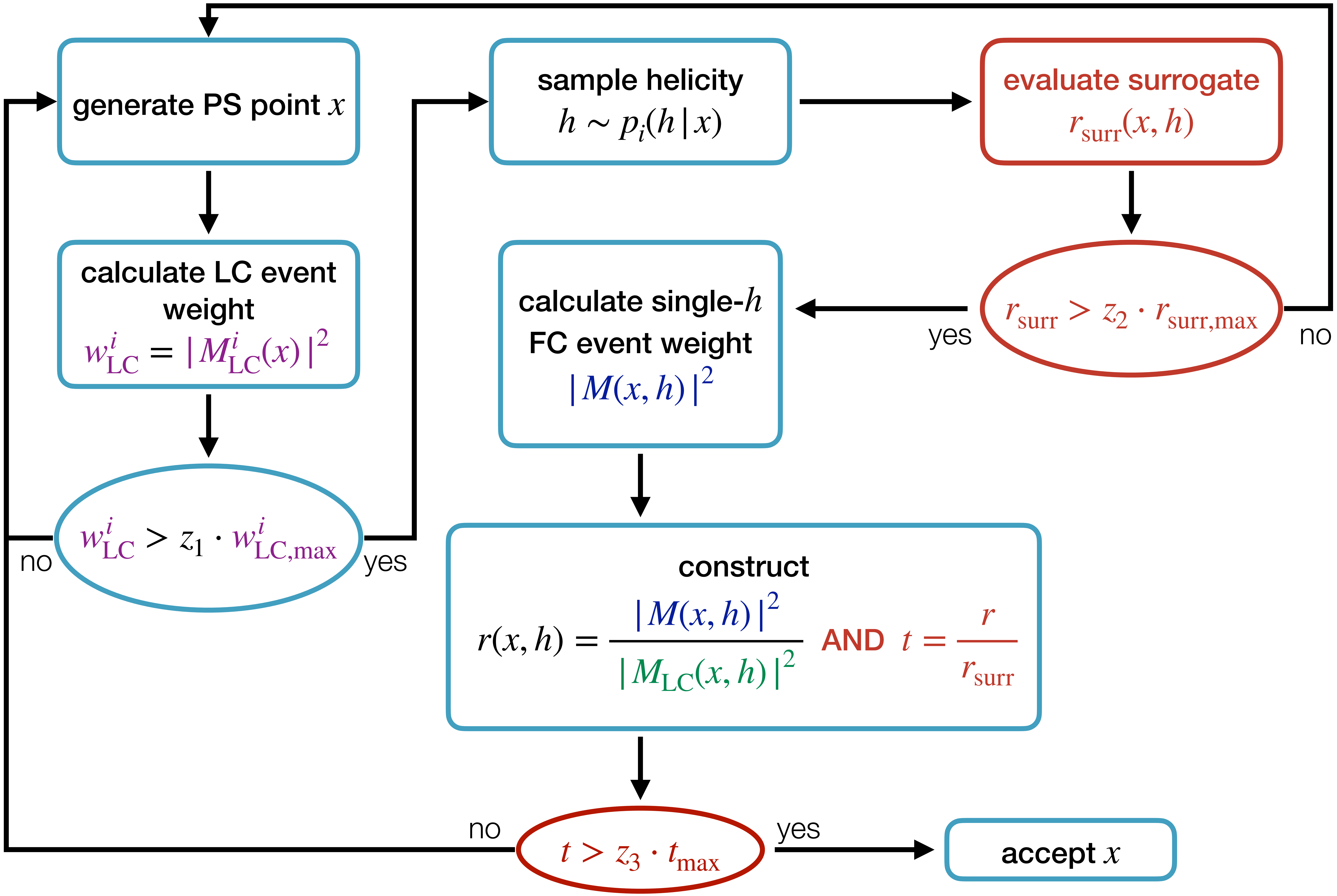
$$r_{\text{surr}}(x, h) \approx r(x, h)$$

- ***Faster to evaluate***: e.g. $\mathcal{O}_{\text{FC}}(1)$ vs $\mathcal{O}_{\text{surr}}(10^{-5})$ s/event
- Much ***better scaling*** with particle multiplicity than approximating $|M_{\text{surr}}(x, h)|^2 \approx |M(x, h)|^2$

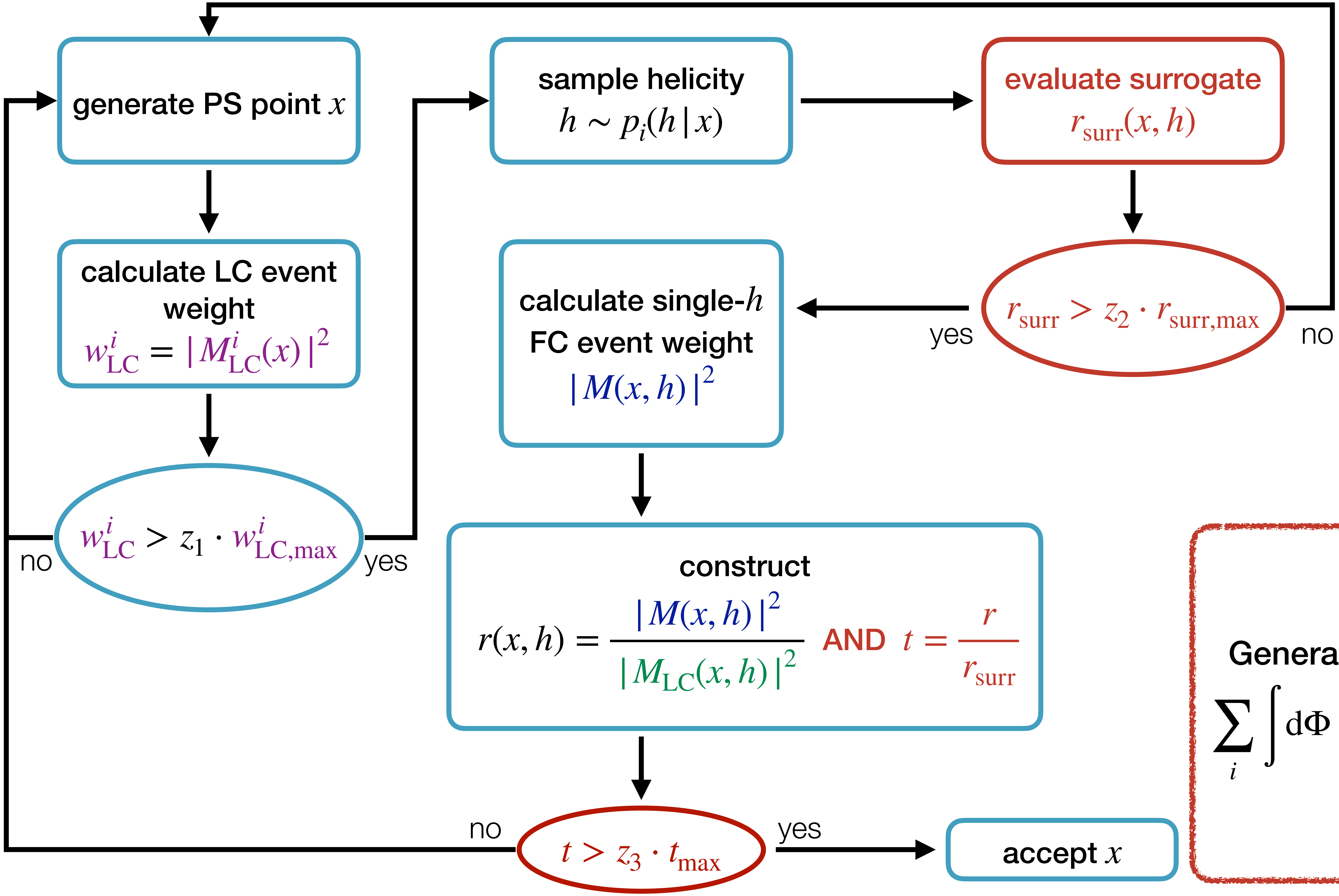
Surrogate-based unweighting



Surrogate-based unweighting



Surrogate-based unweighting



Also unbiased!

Generated events follow FC density

$$\sum_i \int \mathrm{d}\Phi \; |M^i_{\text{LC}}(x)|^2 \left\langle r_{\text{surr}}(x, h) \cdot t \right\rangle_{h \sim p_i(h|x)}$$

$$= \sigma_{\text{FC}}$$

Surrogate-based unweighting

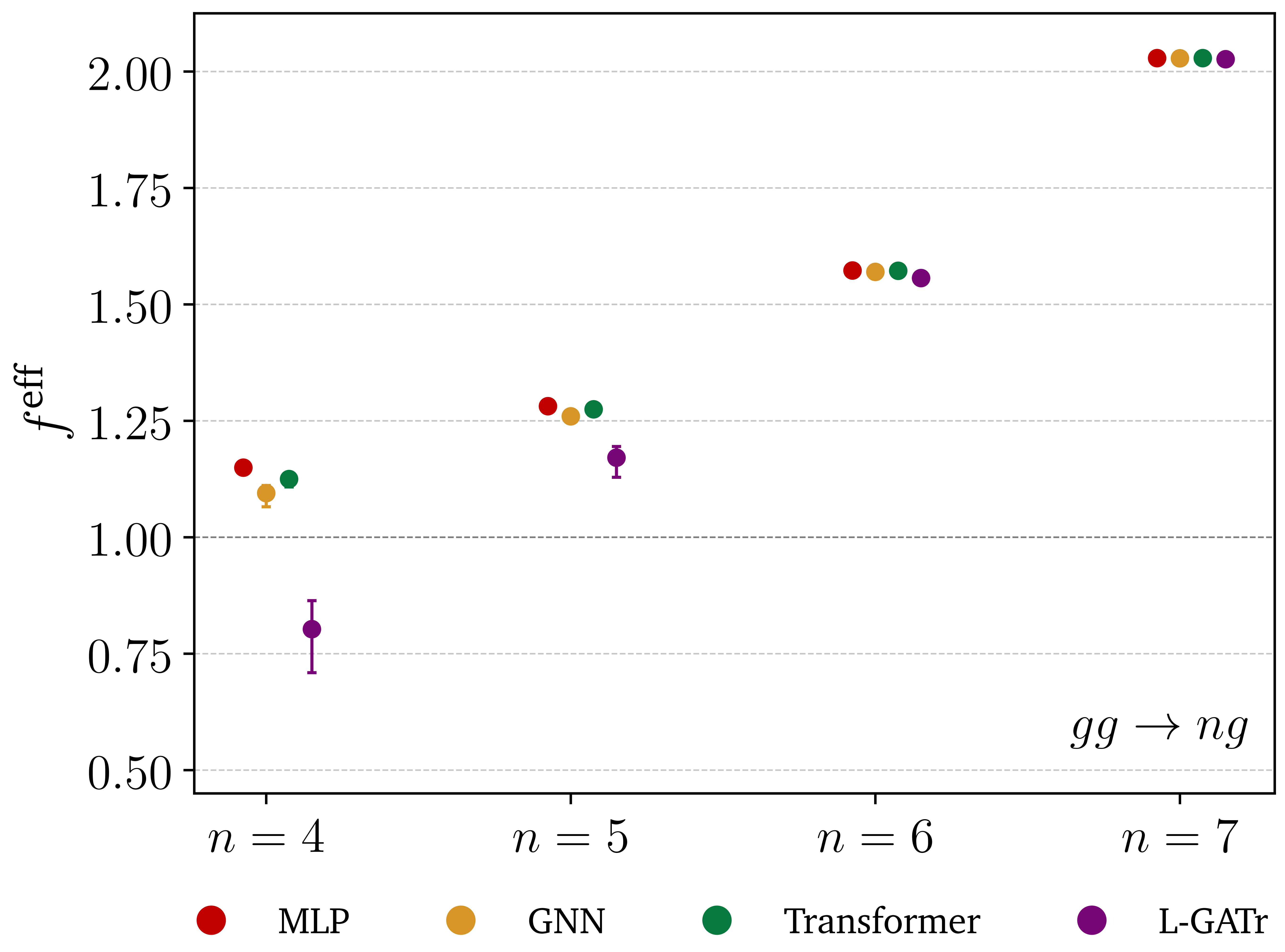
How to quantify gains in performance?

Effective gain factor:

$$f^{\mathrm{eff}} \equiv \frac{T_{\mathrm{LC}}}{T_{\mathrm{surr}}}$$

- T takes into account evaluation time, efficiency, and statistical power of the generated sample

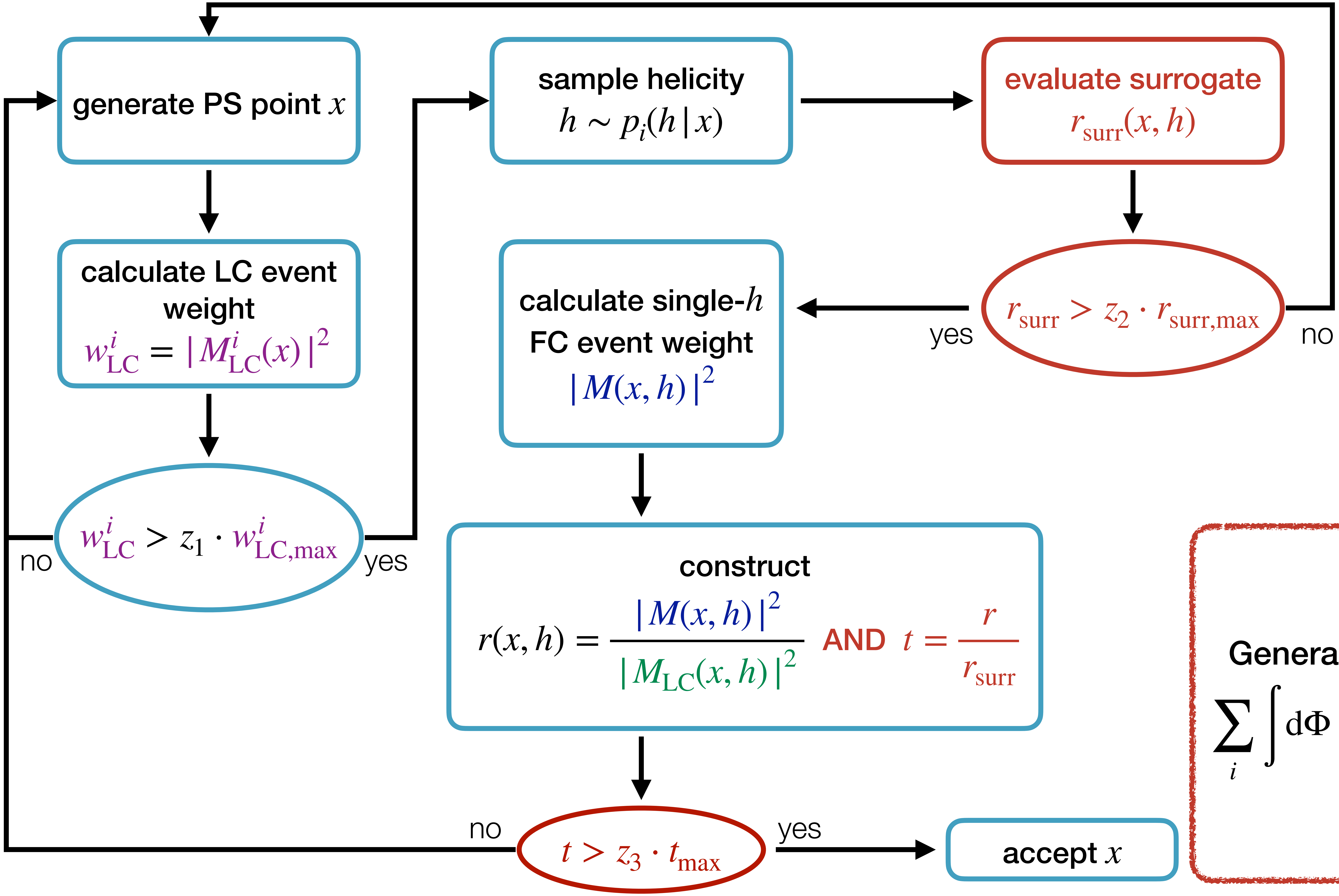
Results - Gain factors



Gains across all processes, from x1.1 to x2 in the all-gluons $n = 7$ case

What if we ignore the FC reweighting?

Ignoring the FC reweighting



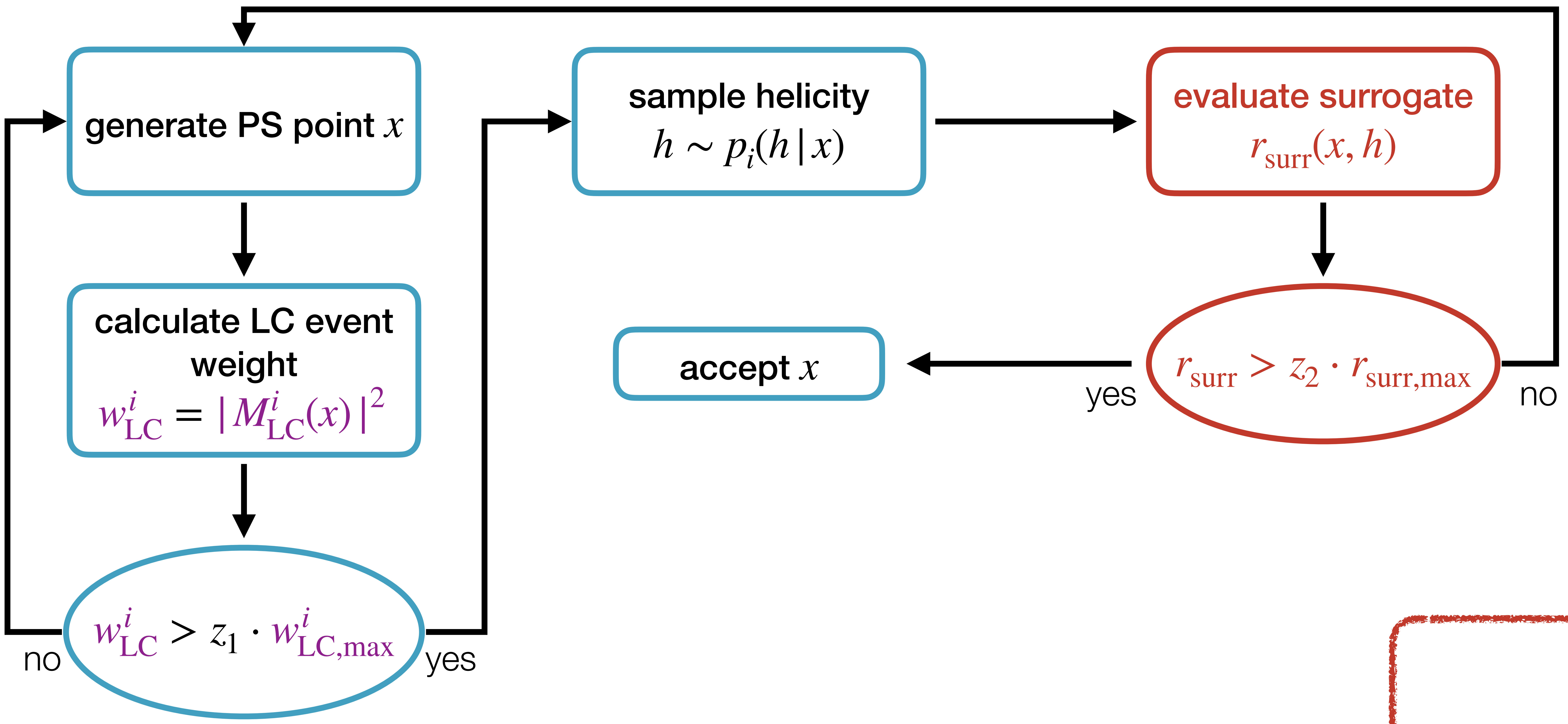
Also unbiased!

Generated events follow FC density

$$\sum_i \int \mathrm{d}\Phi \; |M^i_{\text{LC}}(x)|^2 \left\langle r_{\text{surr}}(x, h) \cdot t \right\rangle_{h \sim p_i(h|x)}$$

$$= \sigma_{\text{FC}}$$

Ignoring the FC reweighting



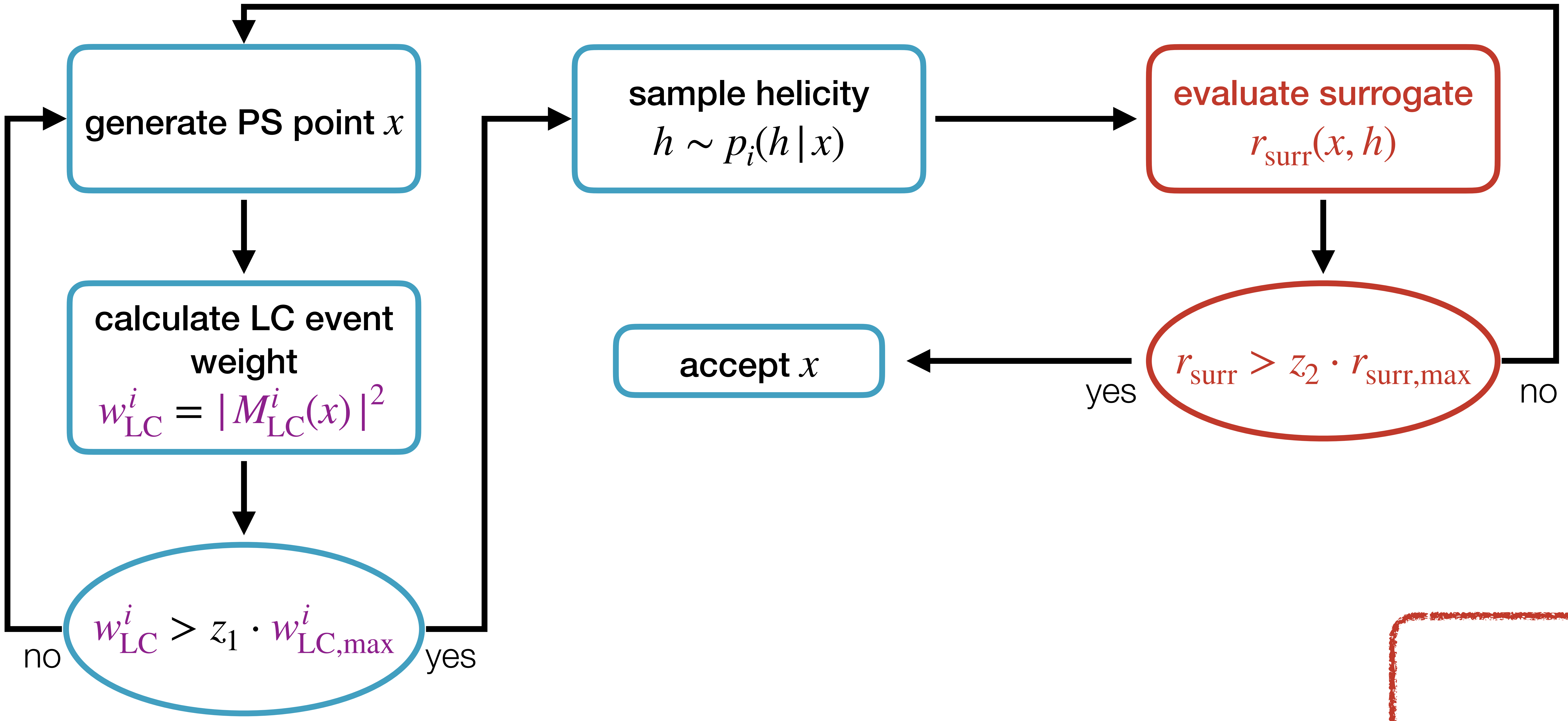
Also unbiased!

Generated events follow FC density

$$\sum_i \int \mathrm{d}\Phi \, |M^i_{\mathrm{LC}}(x)|^2 \left\langle r_{\mathrm{surr}}(x, h) \cdot t \right\rangle_{h \sim p_i(h|x)}$$

$$= \sigma_{\mathrm{FC}}$$

Ignoring the FC reweighting

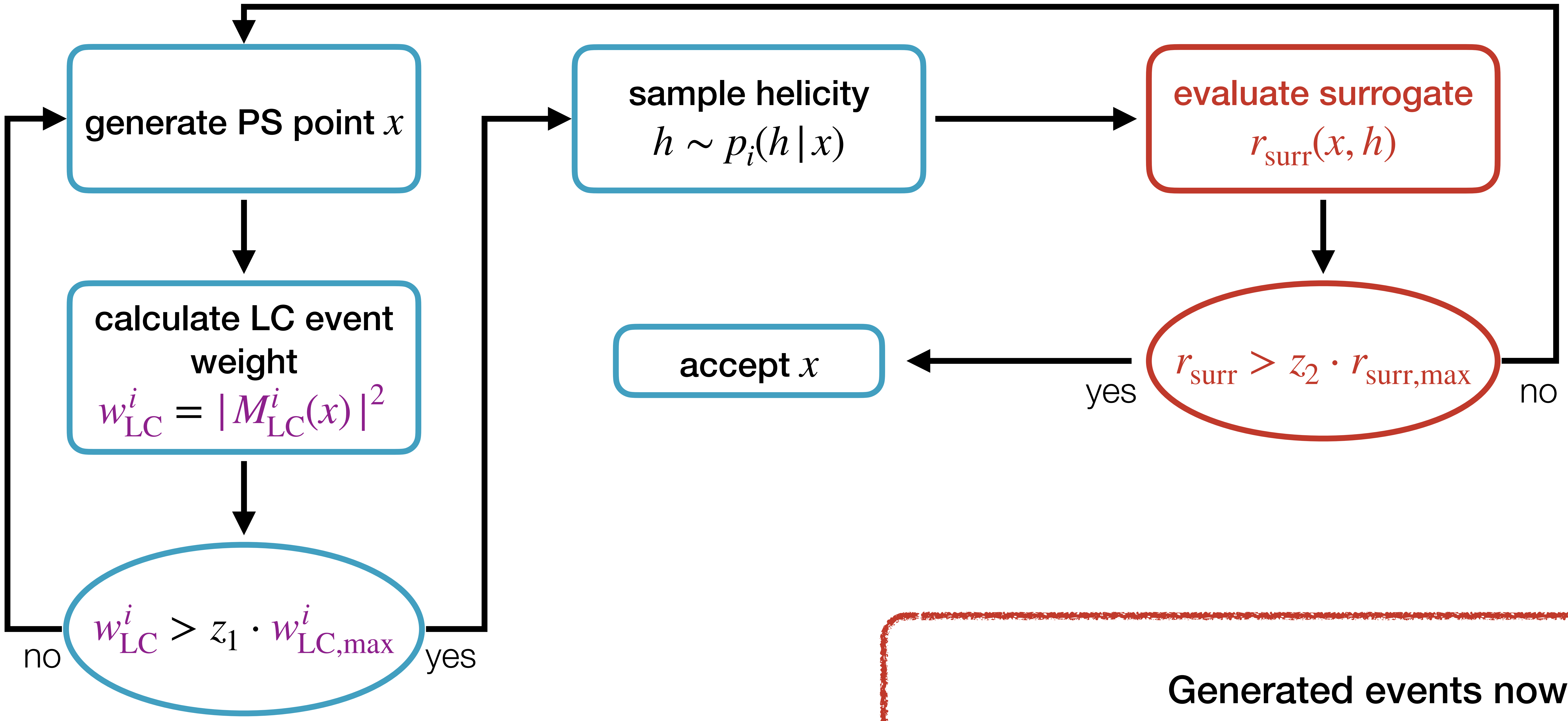


~~**Also unbiased!**~~

~~Generated events follow FC density~~

$$\cancel{\sum_i \int \mathrm{d}\Phi\, |M^i_{\mathrm{LC}}(x)|^2 \left\langle r_{\mathrm{surr}}(x, h) \cdot t \right\rangle_{h \sim p_i(h|x)} = \sigma_{\mathrm{FC}}}$$

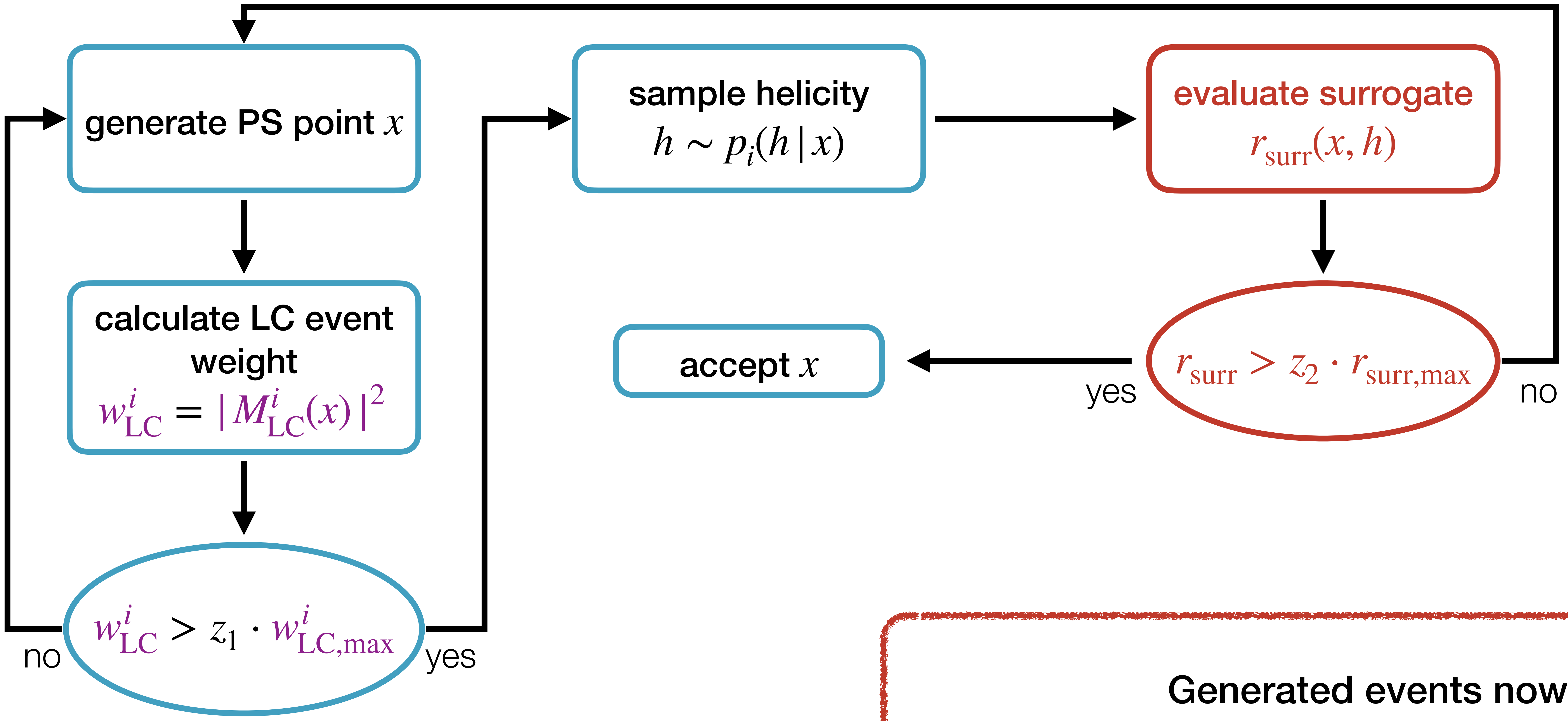
Ignoring the FC reweighting



Generated events now follow density

$$\sum_i \int \mathrm{d}\Phi \, |M^i_{\mathrm{LC}}(x)|^2 \left\langle r_{\mathrm{surr}}(x,h) \right\rangle_{h\sim p_i(h|x)} \equiv \sigma_{\mathrm{surr,FC}} \approx \sigma_{\mathrm{FC}}$$

Ignoring the FC reweighting

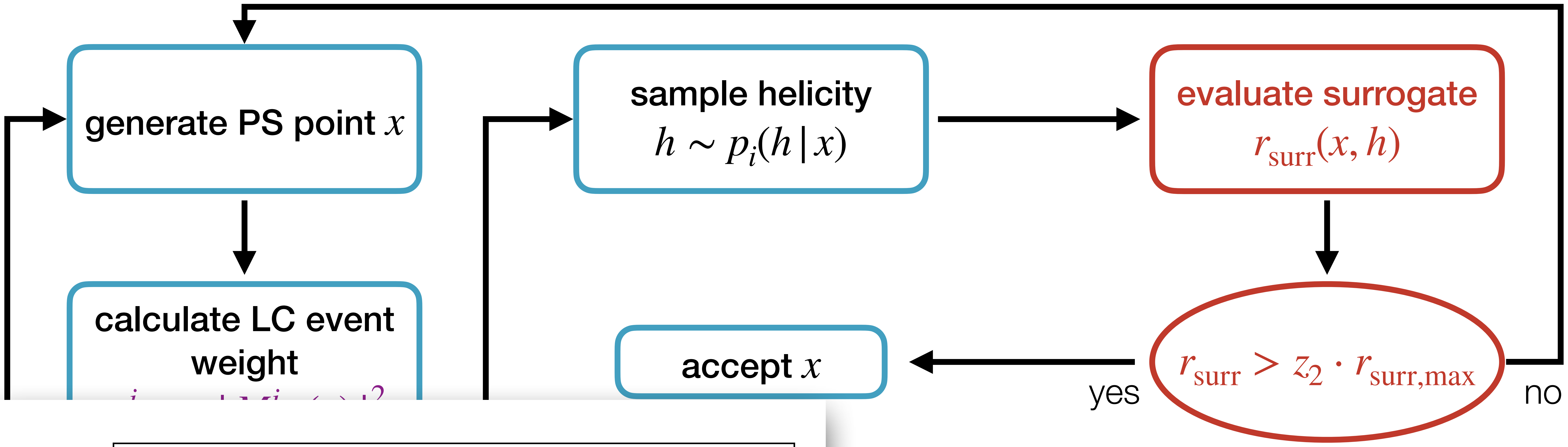


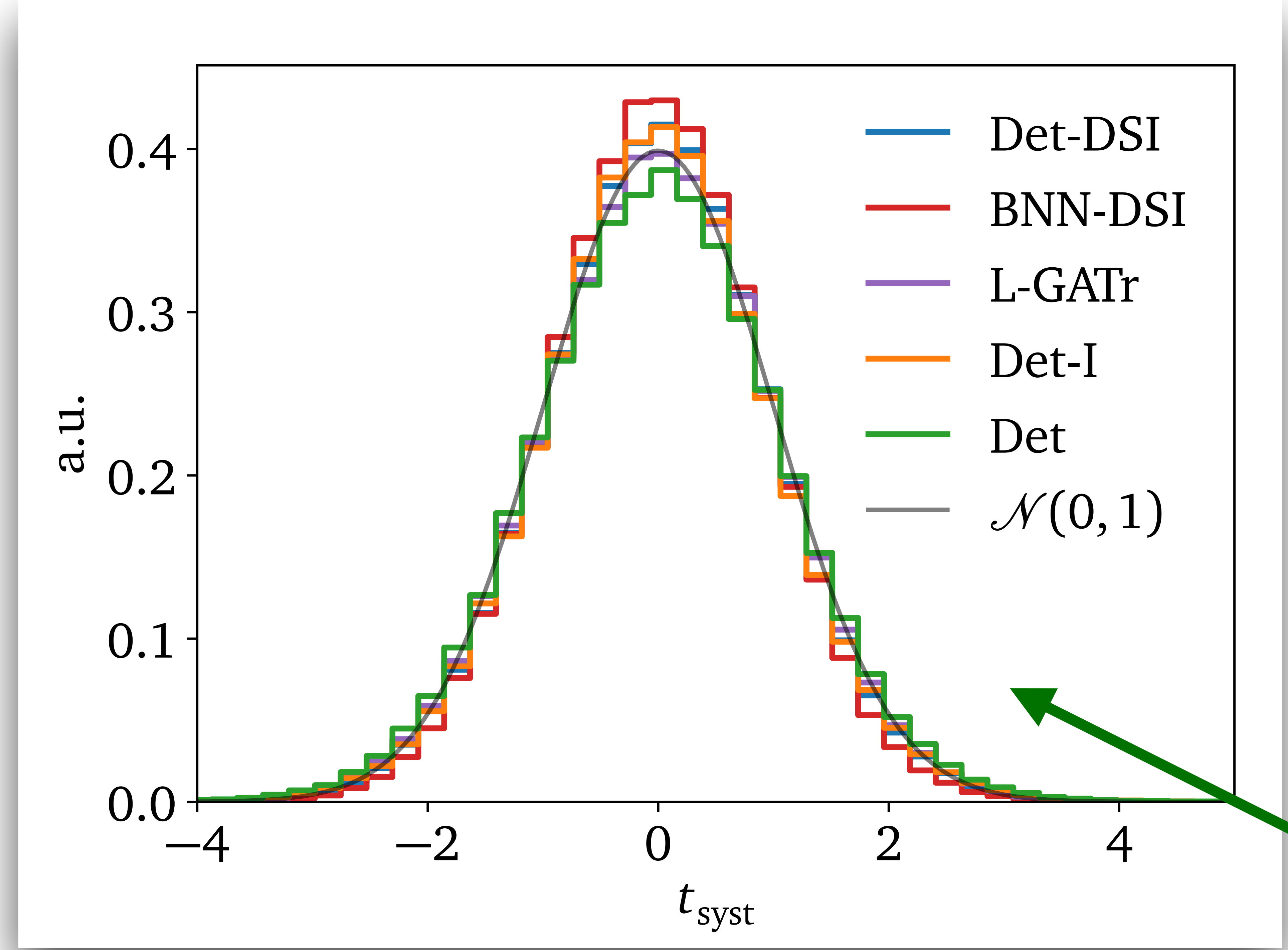
Generated events now follow density

$$\sum_i \int \mathrm{d}\Phi\, |M^i_{\mathrm{LC}}(x)|^2 \left\langle r_{\mathrm{surr}}(x,h) \right\rangle_{h \sim p_i(h|x)} \equiv \sigma_{\mathrm{surr,FC}} \approx \sigma_{\mathrm{FC}}$$

- Requires ***control over surrogate uncertainties***
- But ***very feasible!***

Ignoring the FC reweighting



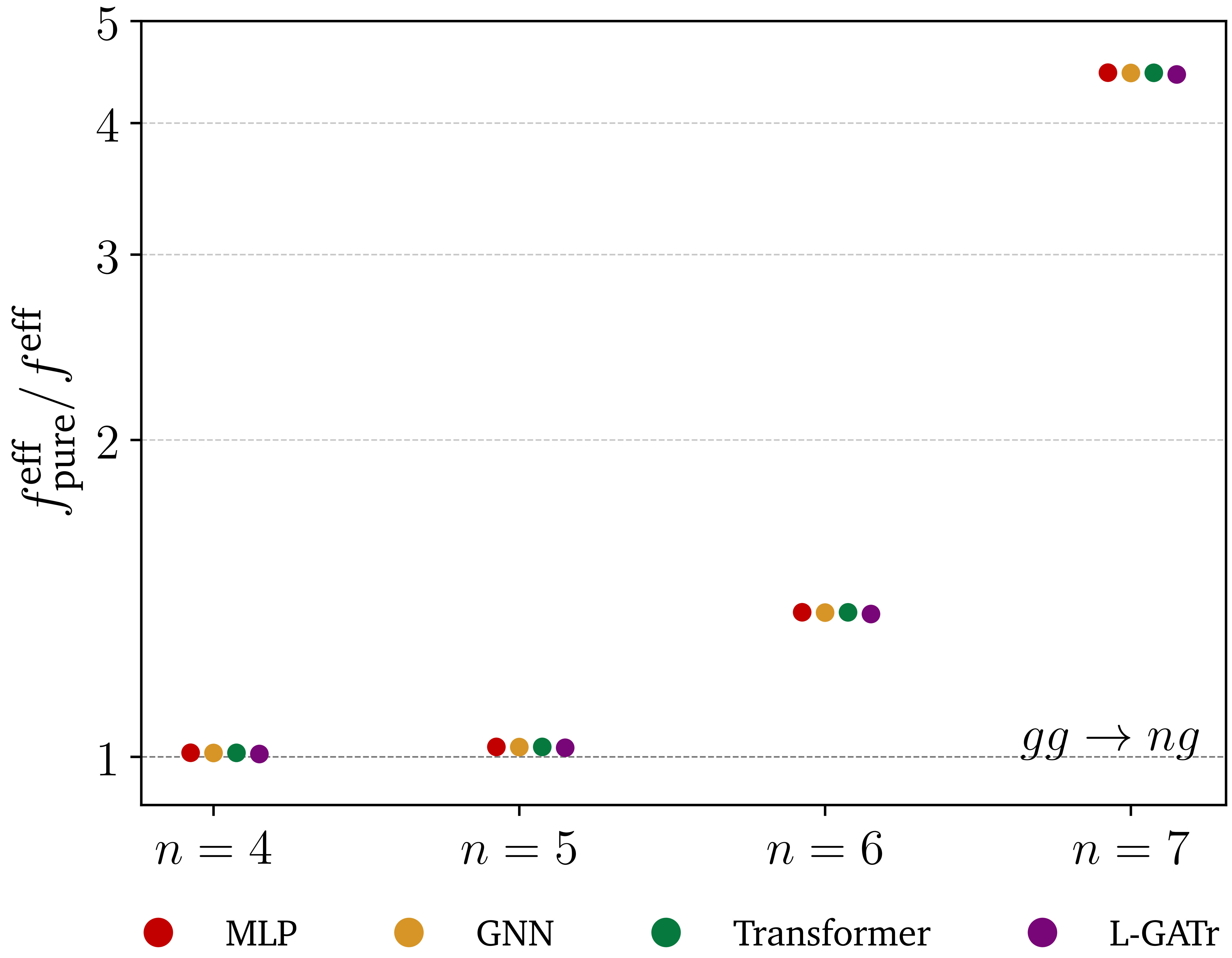


Generated events now follow density

$$\sum_i \int d\Phi \, |M^i_{\text{LC}}(x)|^2 \left\langle r_{\text{surr}}(x,h) \right\rangle_{h\sim p_i(h|x)} \equiv \sigma_{\text{surr,FC}} \approx \sigma_{\text{FC}}$$

- Requires ***control over surrogate uncertainties***
- But ***very feasible!*** **Bahl, Elmer, RW, *et al.* [2412.12069, 2509.00155]**

Results - Gain factors w/o FC rew.



Further improvements compared three-step method

$$f^{\text{eff}}_{\text{pure}} \Big/ f^{\text{eff}} \approx 4.5$$

@ $gg \to 7g$

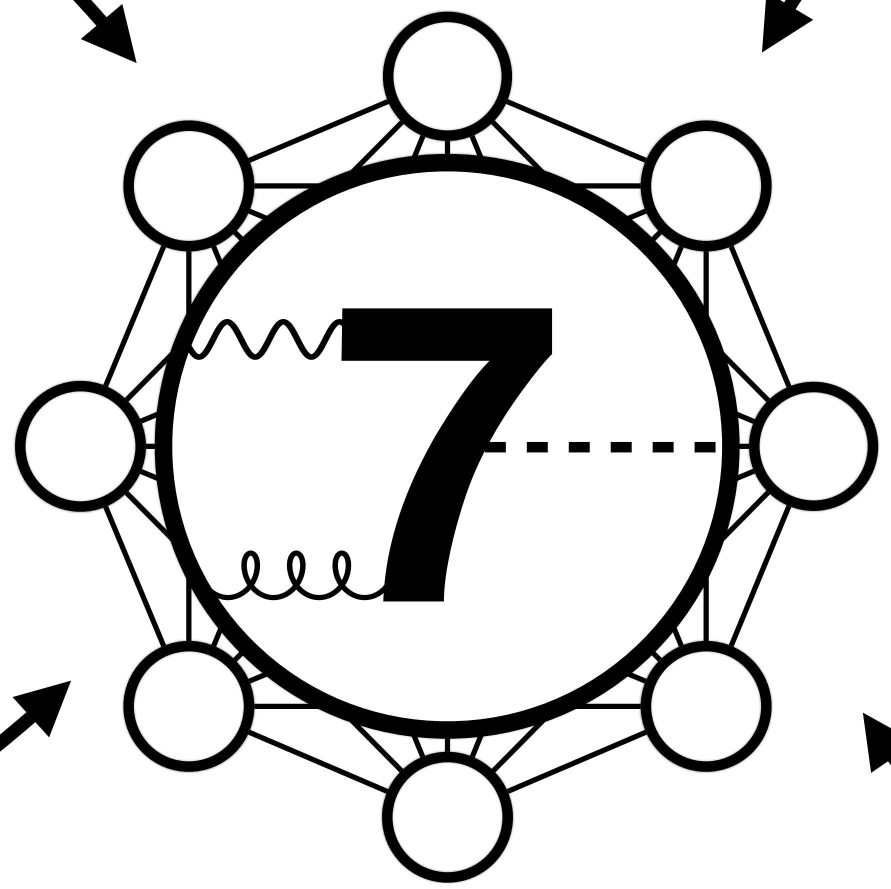
Upcoming MadGraph7

Matrix element on GPU
- huge speed-up from GPUs
- improved CPU performance from SIMD vectorization

MG4GPU [2303.18244, 2312.02898]

MadNIS@LO
- neural importance sampling for better uw. efficiency
- as easy as VEGAS
- Default for many processes

Beccatini, Heimel, Mattelaer, RW [2601.XXXX]



Faster multi-jet events
- Recursion relations for higher multiplicities
- Improved efficiency through 2(3)-stage unweighting

Frederix, Vitos [2409.12128]

Release planned for **early 2026**!

MadEvent7
- new flexible and modular phase space generator
- GPU- and ML-enabled
- usable beyond MadGraph

Heimel, Mattelaer, RW [2512.XXXX]

Towards MadGraph7@NLO

Matrix element on GPU
- huge speed-up from GPUs
- improved CPU performance from SIMD vectorization

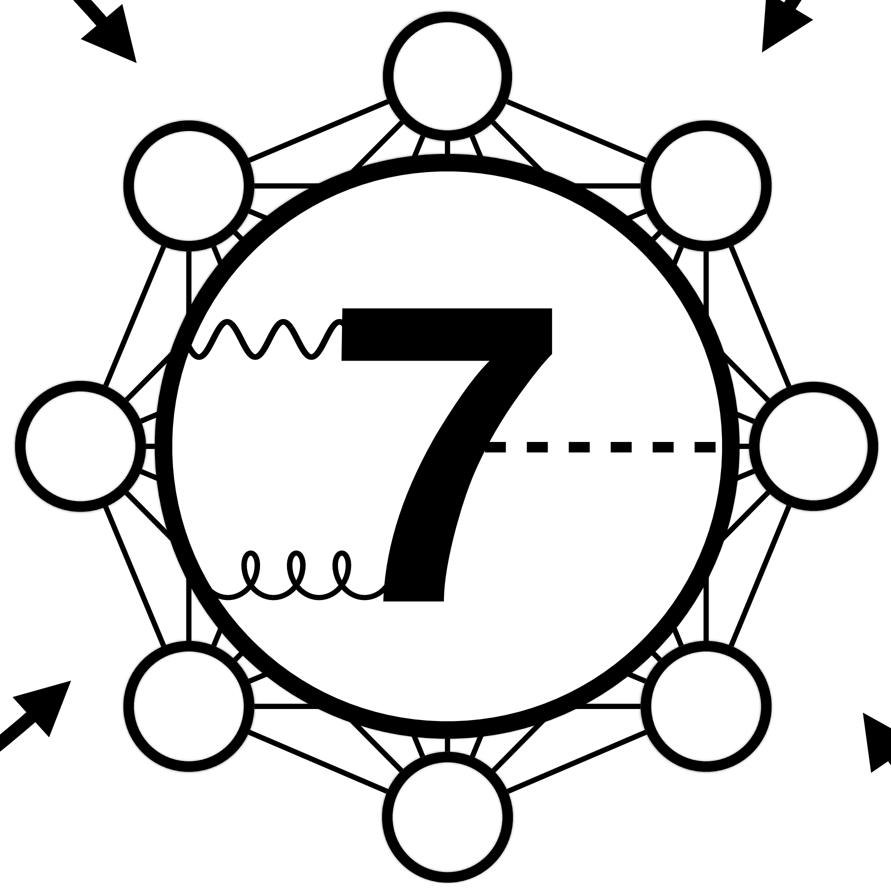
MG4GPU [2303.18244, 2312.02898]

ML-improved subtractions

MadNIS@NLO
- neural importance sampling for better uw. efficiency
- as easy as VEGAS
- Default for many processes

Beccatini, Heimel, Mattelaer, RW [2601.XXXX]

MG4GPU@NLO
- port *real-emission* ME onto GPU [2503.07439]



Fast surrogates@NLO
- ML surrogates for *virtual* MEs

Faster multi-jet events
- Recursion relations for higher multiplicities
- Improved efficiency through 2(3)-stage unweighting

Frederix, Vitos [2409.12128]

NLO-phase
~2026++

MadEvent7@NLO
- new flexible and modular phase space generator
- GPU- and ML-enabled
- usable beyond MadGraph

Heimel, Mattelaer, RW [2512.XXXX]

Open Discussion

The Leading-Color approximation

1. Introduce the ***h*elicity-dependent** squared matrix elements at **FC accuracy** and **LC accuracy**

$$|M(x,h)|^2 = \sum_{i,j} A_{i,h}(x)\, C_{ij}\, A^{*}_{j,h}(x) \qquad\qquad |M_{\mathrm{LC}}(x,h)|^2 = \sum_{i} C_{ii}\, |A_{i,h}(x)|^2$$

2. We also define the **color-dependent** (h-summed) LC matrix element and its relative h-contributions

$$|M^{i}_{\mathrm{LC}}(x)|^2 = \sum_{h} C_{ii}\, |A_{i,h}(x)|^2 \qquad\qquad p_i(h\,|\,x) = \frac{C_{ii}\,|A_{i,h}(x)|^2}{|M^{i}_{\mathrm{LC}}(x)|^2}$$

3. The FC cross section can thus be calculated as:

$$\sigma_{\mathrm{FC}} = \sum_{i} \int \mathrm{d}\Phi\, |M^{i}_{\mathrm{LC}}(x)|^2 \left\langle \boxed{\frac{|M(x,h)|^2}{|M_{\mathrm{LC}}(x,h)|^2}} \right\rangle_{h\sim p_i(h|x)}$$

$$r_{\mathrm{LC}\to\mathrm{FC}}(x,h)$$