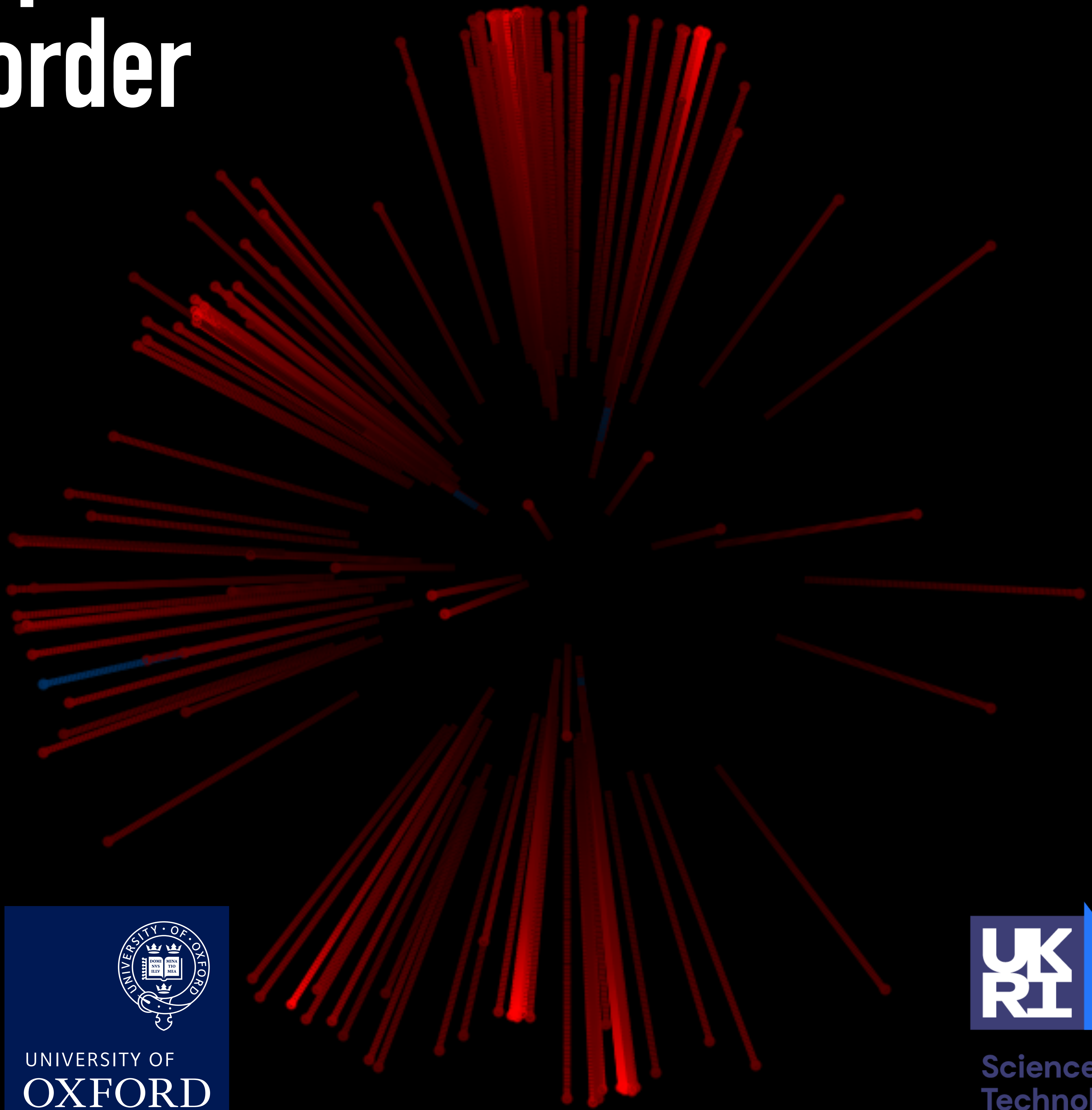
Positivity in QCD predictions beyond leading order

What is particle Physics?
KITP, UC Santa Barbara
March 26, 2025



Gavin Salam
Rudolf Peierls Centre for Theoretical Physics
& All Souls College, Oxford

UNIVERSITY OF OXFORD

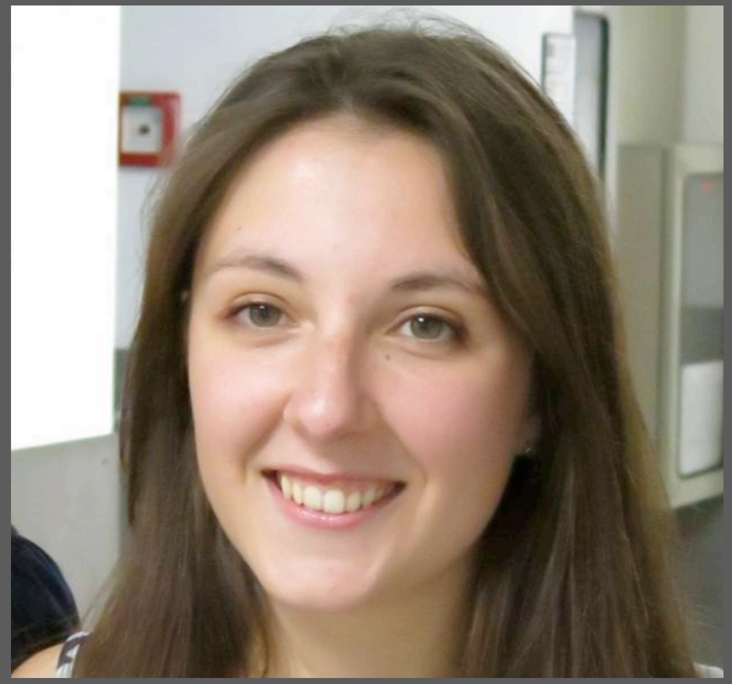
UKRI Science and Technology Facilities Council

THE ROYAL SOCIETY


Melissa van Beekveld
NIKEHF


Mrinal Dasgupta
Manchester


Basem El-Menoufi
Monash


Silvia Ferrario Ravasio
CERN


Keith Hamilton
Univ. Coll. London


Jack Helliwell
Monash


Alexander Karlberg
CERN


Pier Monni
CERN


GPS
Oxford


Nicolas Schalch
Oxford


Ludovic Scyboz
Monash


Alba Soto-Ontoso
Granada


Grégory Soyez
IPhT, Saclay


Silvia Zanoli
Oxford

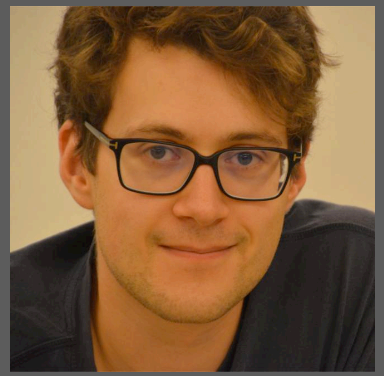
PanScales

A project to bring logarithmic understanding and accuracy to parton showers

European Research Council
Established by the European Commission

ERC funded
2018–2024

former members


Frédéric Dreyer


Rob Verheyen


Rok Medves


Emma Slade

Logarithmically-accurate and positive-definite NLO shower matching

arXiv:2504.nnnnn

Melissa van Beekveld
NIKEHF

Silvia Ferrario Ravasio
CERN

Jack Helliwell
Monash

Alexander Karlberg
CERN

GPS
Oxford

Ludovic Scyboz
Monash

Alba Soto-Ontoso
Granada

Grégory Soyez
IPhT, Saclay

Silvia Zanoli
Oxford

Logarithmically-accurate and positive-definite NLO shower matching

arXiv:2504.nnnnn

Melissa van Beekveld
NIKEHF

Silvia Ferrario Ravasio
CERN

Jack Helliwell
Monash

Alexander Karlberg
CERN

GPS
Oxford

Ludovic Scyboz
Monash

Alba Soto-Ontoso
Granada

Grégory Soyez
IPhT, Saclay

Silvia Zanoli
Oxford

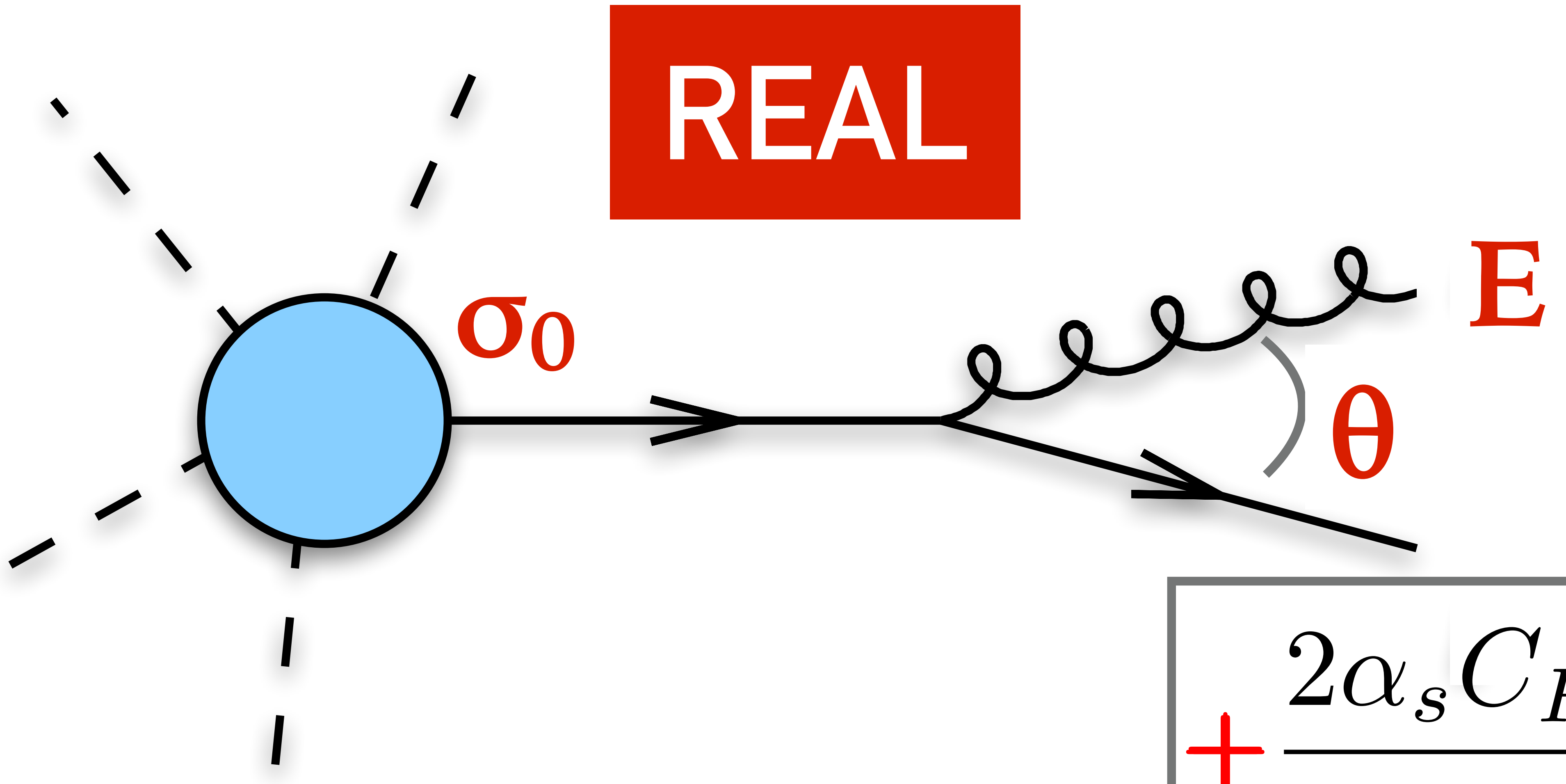
Why not just plain (N)NLO?

incredibly powerful, get scattering cross-sections from first few orders of perturbative expansion in the strong coupling α_s

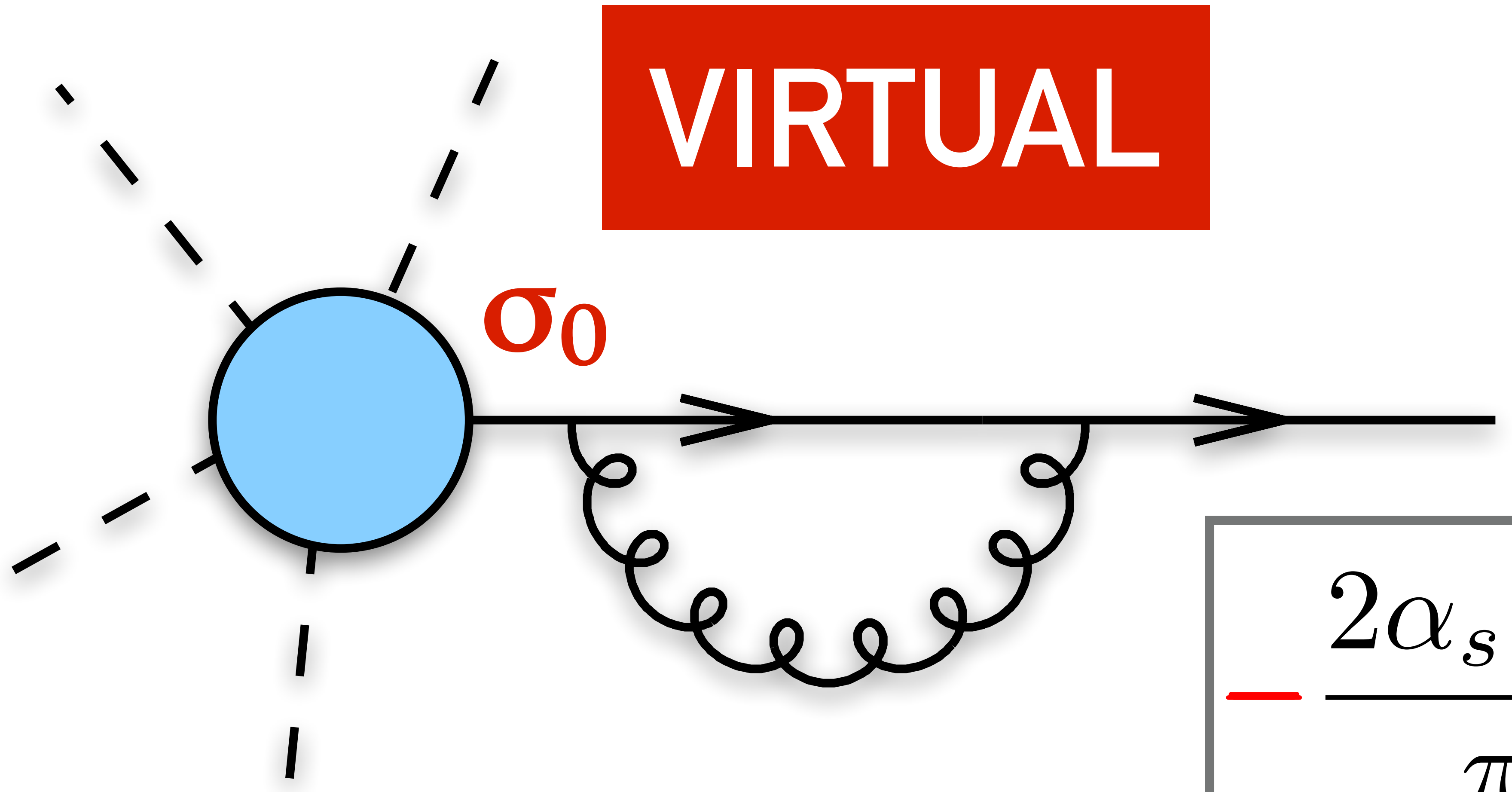
$$\sigma = \sigma_0 + \alpha_s \sigma_1 + \alpha_s^2 \sigma_2 + \cdots$$

LO NLO NNLO

What kind of contributions do we get at NLO?



$$+\frac{2\alpha_s C_F}{\pi}\frac{dE}{E}\frac{d\theta}{\theta}$$



$$-\frac{2\alpha_s C_F}{\pi}\frac{dE}{E}\frac{d\theta}{\theta}$$

Divergences are present in both real and virtual diagrams.

They arise when an emission has a small energy ($E \ll 1$) or a small angle ($\theta \ll 1$).

In dim-reg, this brings $1/\varepsilon^2$ for each order in α_s.

What a NLO calculation gives you (here, Event2, $e^+e^- \to q\bar{q}$)

LO (2-particle) tree-level event

with weight 1.00000

px, py, pz, E =	-1.32	-1.38	-49.96	50.00
px, py, pz, E =	1.32	1.38	49.96	50.00

What a NLO calculation gives you (here, Event2, $e^+e^- \to q\bar{q}$)

LO (2-particle) tree-level event
with weight 1.00000
px, py, pz, E = -1.32 -1.38 -49.96 50.00
px, py, pz, E = 1.32 1.38 49.96 50.00

LO event ($q\bar{q}$)

What a NLO calculation gives you (here, Event2, $e^+e^- \to q\bar{q}$)

LO (2-particle) tree-level event
with weight 1.00000
px, py, pz, E = -1.32 -1.38 -49.96 50.00
px, py, pz, E = 1.32 1.38 49.96 50.00

NLO (3-particle) tree-level event
with **weight** **893.22103**, multiplying (alphas/2pi)
px, py, pz, E = -1.60 -1.75 -49.87 49.93
px, py, pz, E = 1.31 1.36 49.25 49.29
px, py, pz, E = 0.30 0.39 0.62 0.79

LO event ($q\bar{q}$)

NLO event, with real emission
~ LO event + extra soft gluon
and **large positive weight**

What a NLO calculation gives you (here, Event2, $e^+e^- \to q\bar{q}$)

LO (2-particle) tree-level event
with weight 1.00000
px, py, pz, E = -1.32 -1.38 -49.96 50.00
px, py, pz, E = 1.32 1.38 49.96 50.00

NLO (3-particle) tree-level event
with **weight 893.22103**, multiplying (alphas/2pi)
px, py, pz, E = -1.60 -1.75 -49.87 49.93
px, py, pz, E = 1.31 1.36 49.25 49.29
px, py, pz, E = 0.30 0.39 0.62 0.79

NLO (2-particle) virtual subtraction event
with **weight -84.49299**, multiplying (alphas/2pi)
px, py, pz, E = -1.32 -1.38 -49.96 50.00
px, py, pz, E = 1.32 1.38 49.96 50.00

NLO (2-particle) virtual subtraction event
with **weight -808.58646**, multiplying (alphas/2pi)
px, py, pz, E = -1.61 -1.75 -49.94 50.00
px, py, pz, E = 1.61 1.75 49.94 50.00

NLO (2-particle) virtual finite event
with weight 2.66667, multiplying (alphas/2pi)
px, py, pz, E = -1.32 -1.38 -49.96 50.00
px, py, pz, E = 1.32 1.38 49.96 50.00

LO event ($q\bar{q}$)

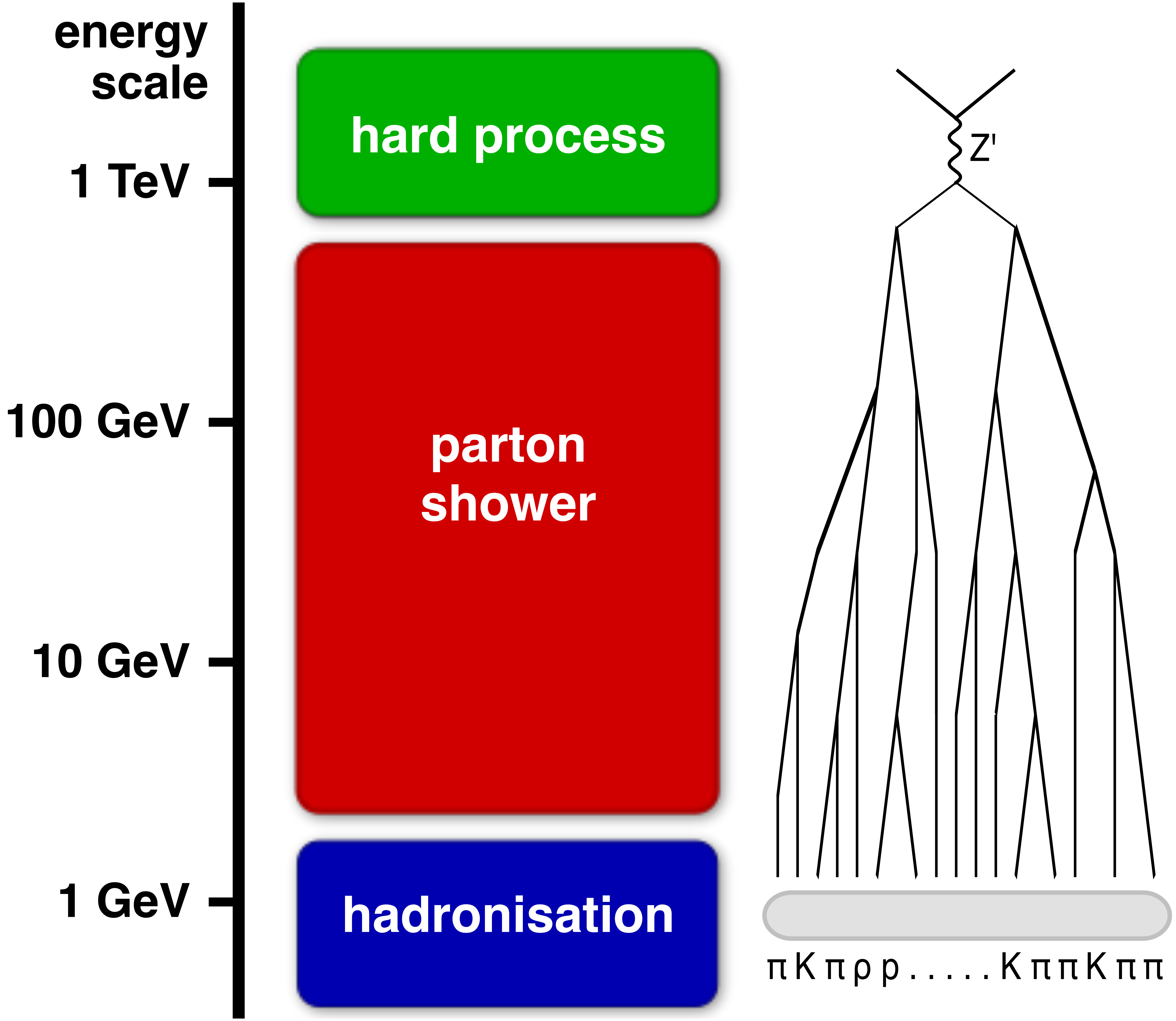
NLO event, with real emission
~ LO event + extra soft gluon
and **large positive weight**

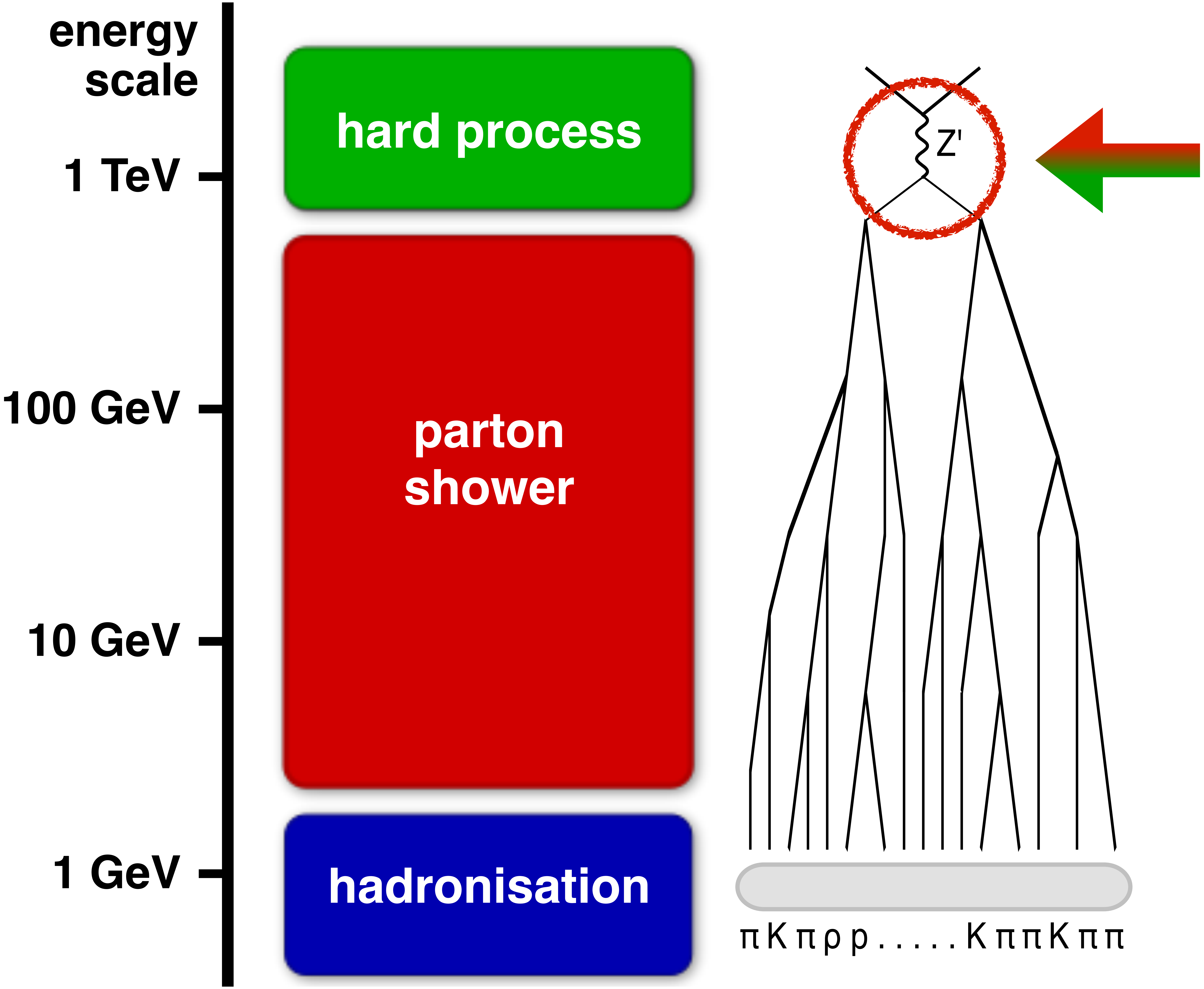
NLO event, “virtual” correction
~ LO event
and **large negative weight**

event weights are ~ probabilities

- real life doesn't have negative probabilities
- real life doesn't have (near-)divergent probabilities
- you can evade these problems in perturbation theory if you ask very limited kinds of questions, i.e. nearly always summing real & virtual divergences (infrared safe observable, single momentum scale)*
- but experiments don't limit themselves to those kinds of questions

* though there can still be nasty surprises, cf. Chen et al 2102.07607, GPS & Slade 2106.08329


energy scale
1 TeV
100 GeV
10 GeV
1 GeV
hard process
parton shower
hadronisation
Z'
π K π ρ p . . . . . K π π K π π



<u>Key innovation of 2002–'04:</u>
correct or replace
first step so that perturbative
expansion of hard process +
parton-shower is equivalent to the
true NLO.

Frixione & Webber: MC@NLO
<u>hep-ph/0204244</u>

Nason: POWHEG
<u>hep-ph/0409146</u>

[>7500 citations; these methods used also in Sherpa, Herwig]

Key features of MC@NLO and POWHEG events

- MC@NLO and POWHEG methods, supplemented with parton showers + hadronisation models, provide NLO-accurate realistic hadron-level events
- they avoid the problem of (near) divergent event weights
- instead the event weights are just ± 1

This is a big advantage over "pure" NLO

But the event sample doesn't quite look like a true physical event sample, because there are still some negative weights

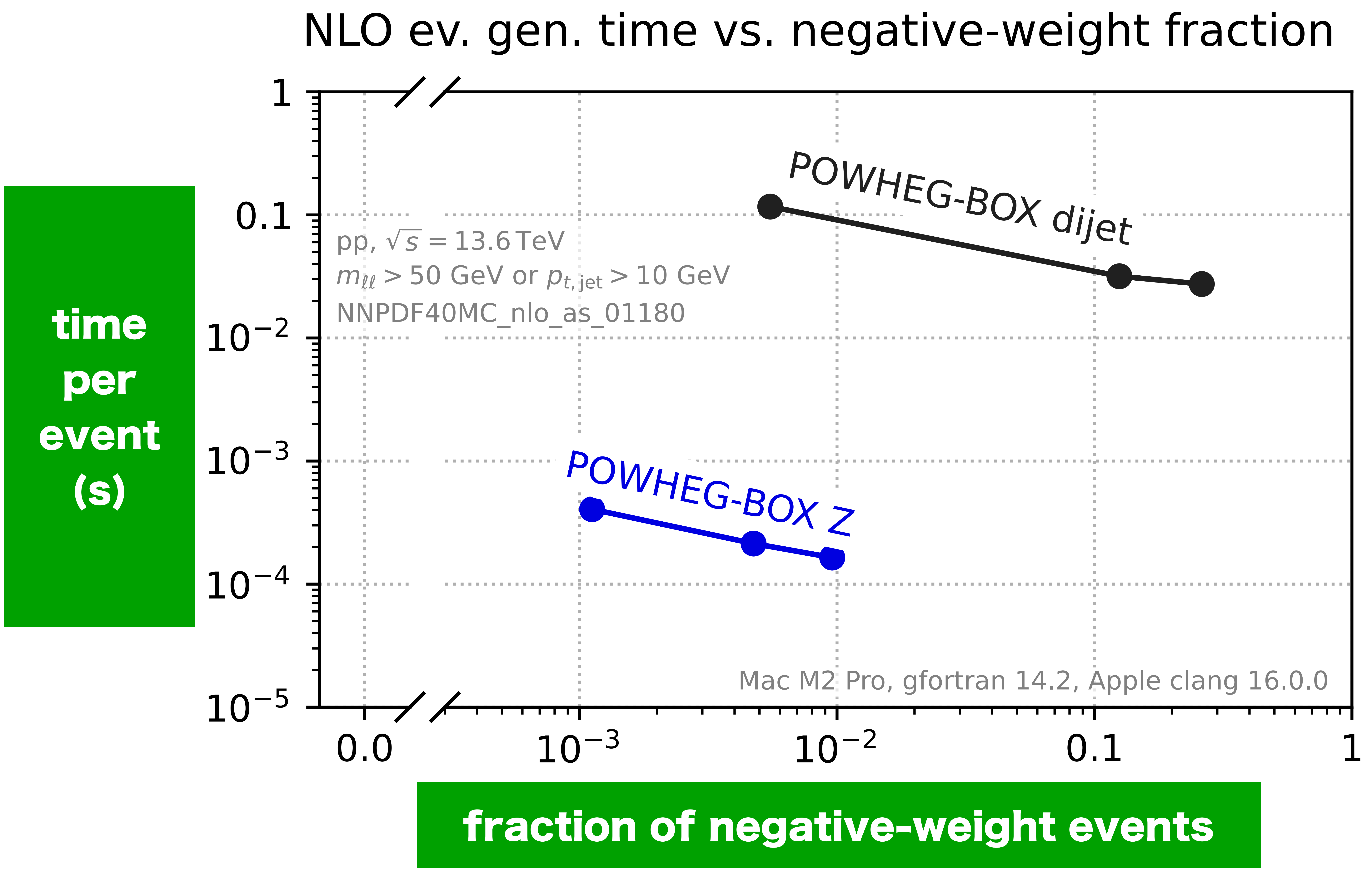
Are negative weights a problem?

Given fraction f of negative-weight events, to reach the same statistical error as for N unit positive-weight events, you need to generate a larger number of events,

$$\frac{N}{(1-2f)^2}$$

E.g. for $f = 15\,\%$ this doubles the required number of events.

NLO state of the art [POWHEG]



More complex processes tend to have higher fractions of negative weights.

Mitigation options (e.g. folding, Nason 0709.2085) often trade off negative weight fraction v. generation time.

Problem usually worse for NNLO event generation

NLO state of the art [MG5_aMC — MC@NLO method]

	step-0 (s) (grid setup)	step-1 (s) (integration)	step-2 (s) (generation)	negative $\mathbb{S}$ weights
$pp \to e^+e^-$				
default	1	14	147	7.1%
$2 \times 2 \times 1$ folding	1	33	258	2.1%
$4 \times 4 \times 1$ folding	1	114	781	1.8%
Born spreading	113	30	189	2.0%
$pp \to W^+j$				
default	10	604	2013	24.2%
$2 \times 2 \times 1$ folding	10	1265	5160	13.2%
$4 \times 4 \times 1$ folding	7	2803	16020	9.0%
Born spreading	355	645	2226	18.8%

Frederix & Torrielli
2310.04160

wall times for 1 million events on a 12-core i7-8700K @ 3.7 GHz desktop machine

Some LHC experiments' statements on negative weights and machine learning

- ATLAS 2211.01136: "To avoid the use of negative weights present in the nominal NLO sample in the training of the multivariate discriminant used to separate SM $t\bar{t}t\bar{t}$ events from background [...], **a sample was produced with similar generator settings, but at LO.**"

- CMS 2411.03023: "However, the binary cross-entropy given by Eq. (2), can become negatively unbounded for negative event weights, **making the classification task potentially impossible**"

- ATLAS 2412.15123: "Since XGBoost [ML framework] cannot handle negative-weight events, **the absolute value of each event weight is used**."

other work trying to reduce negative weight fractions (+ further refs below)

K. Danziger, S. Höche and F. Siegert, *Reducing negative weights in Monte Carlo event generation with Sherpa*, 2110.15211.

J. R. Andersen and A. Maier, *Unbiased elimination of negative weights in Monte Carlo samples*, *Eur. Phys. J. C* **82** (2022) 433, [2109.07851].

J. R. Andersen, A. Maier and D. Maître, *Efficient negative-weight elimination in large high-multiplicity Monte Carlo event samples*, *Eur. Phys. J. C* **83** (2023) 835, [2303.15246].

J. R. Andersen, A. Cueto, S. P. Jones and A. Maier, *A Cell Resampler study of Negative Weights in Multi-jet Merged Samples*, 2411.11651.

B. Nachman and J. Thaler, *Neural resampler for Monte Carlo reweighting with preserved uncertainties*, *Phys. Rev. D* **102** (2020) 076004, [2007.11586].

E.g. “We have demonstrated that the fraction of negative event weights in existing large high-multiplicity samples can be reduced by more than an order of magnitude, whilst preserving predictions for observables within statistical uncertainties.” [2303.15246]

are we doing our (perturbative QFT) job properly if we can't deliver guaranteed positive predictions?

3 stages of NLO event generation

1. Generate “Born” event, e.g. $q\bar{q} \to Z$, with an overall NLO-correct normalisation
2. Generate real radiation, e.g. extra gluon, with correct real matrix element
3. Let a parton shower generate all remaining perturbative emission

3 stages of NLO event generation

1. Generate "Born" event, e.g. $q\bar{q} \to Z$, with an overall NLO-correct normalisation
2. Generate real radiation, e.g. extra gluon, with correct real matrix element
3. Let a parton shower generate all remaining perturbative emission

	$1 + \mathcal{O}(\alpha_s)$	$\mathcal{O}(\alpha_s)$
	NLO event normalisation	Generation of first emission
MC@NLO	can be negative	can be negative
POWHEG*	can be negative	always positive

** and also the KrkNLO [Jadach et al 1503.06849] and MAcNLOPS [Nason & GPS, 2111.03553] methods*

3 stages of NLO event generation

1. Generate "Born" event, e.g. $q\bar{q} \to Z$, with an overall NLO-correct normalisation
2. Generate real radiation, e.g. extra gluon, with correct real matrix element
3. Let a parton shower generate all remaining perturbative emission

	$1 + \mathcal{O}(\alpha_s)$	$\mathcal{O}(\alpha_s)$
	NLO event normalisation	Generation of first emission
MC@NLO	can be negative	can be negative
POWHEG*	can be negative	always positive

* *and also the KrkNLO [Jadach et al 1503.06849] and MAcNLOPS [Nason & GPS, 2111.03553] methods*

NLO Born normalisation, inclusive over subsequent branching

Born +NLO norm. = *Born* + *1-loop virtual* + *real*
(LO + NLO) *(LO)* *(NLO)* *(NLO)*

3 dim. of phase space for one extra emission

$$\bar{B}(\Phi_{\rm B}) = B_0(\Phi_{\rm B}) + \underbrace{V(\Phi_{\rm B}) + \int R(\Phi_{\rm B}, \Phi_{\rm rad})\, d\Phi_{\rm rad}}_{\text{relative order } \alpha_s}\,,$$

NLO Born normalisation, inclusive over subsequent branching

Born +NLO norm. = *Born* + *1-loop virtual* + *real*
(LO + NLO) *(LO)* *(NLO)* *(NLO)*

3 dim. of phase space for one extra emission

$$\bar{B}(\Phi_B) = B_0(\Phi_B) + \underbrace{V(\Phi_B) + \int R(\Phi_B, \Phi_{rad})\, d\Phi_{rad}}_{\text{relative order } \alpha_s},$$

Made explicitly finite with counterterms

counterterm (integrated)

counterterm (differential)

$$\bar{B}(\Phi_B) = B_0(\Phi_B) + \underbrace{V(\Phi_B) + C_{int}(\Phi_B) + \int [R(\Phi) - C(\Phi)]\, d\Phi_{rad}}_{\text{relative order } \alpha_s},$$

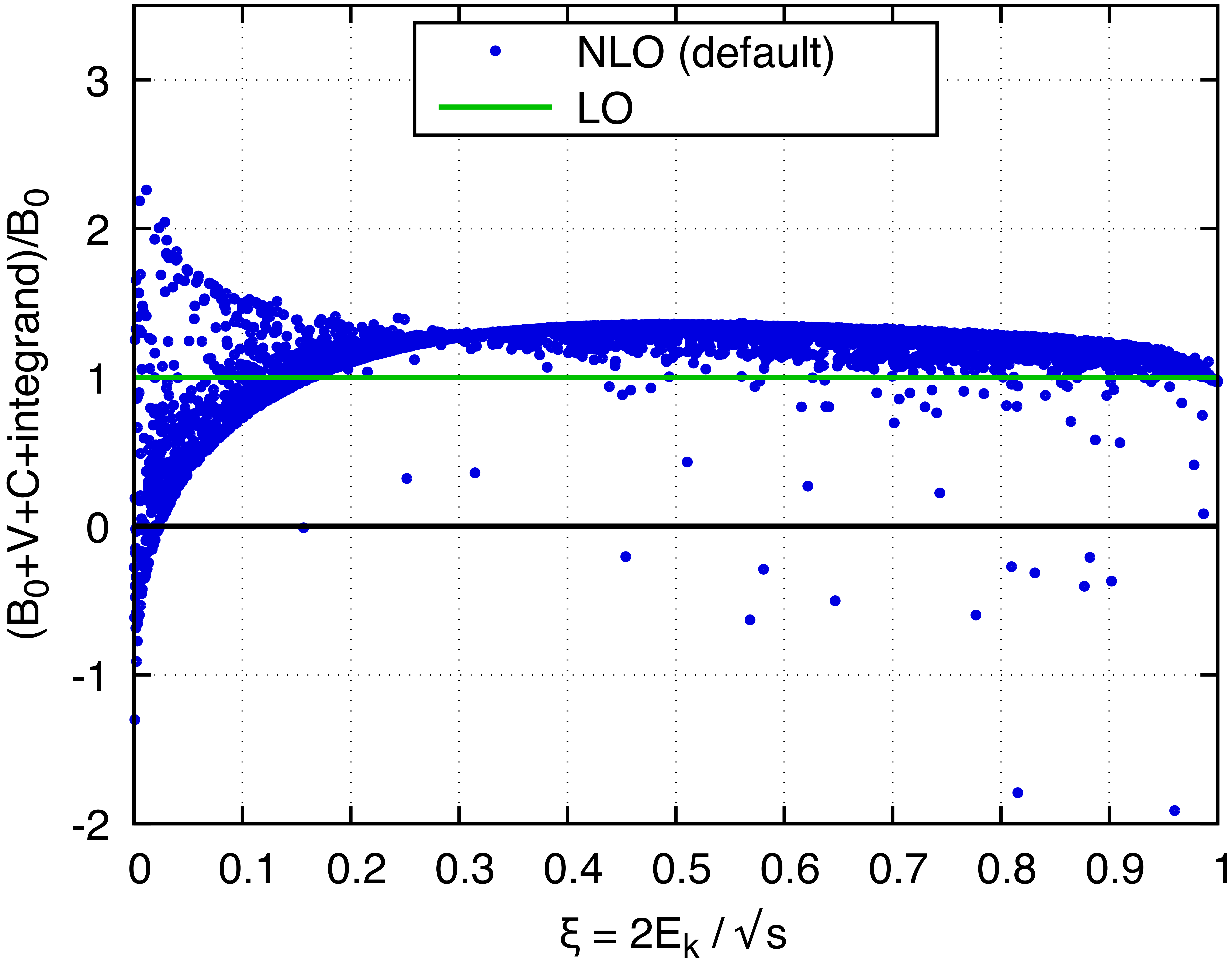
How does it work in practice?

- Choose a Born phase space point Φ_B randomly
- Instead of accurately evaluating the $d\Phi_{\text{rad}}$ integral, choose a random real phase space point Φ_{rad} and use that to get a “single-point Monte Carlo” estimate for the integral

$$\bar{B}(\Phi_{\rm B}) = B_0(\Phi_{\rm B}) + \underbrace{V(\Phi_{\rm B}) + C_{\rm int}(\Phi_{\rm B}) + \int \left[R(\Phi) - C(\Phi)\right] d\Phi_{\rm rad}}_{\text{relative order } \alpha_s},$$

- accept with probability $|\bar{B}|/\text{max}$, event weight is sign of $\bar{B}$

$$\bar{B}(\Phi_B) = B_0(\Phi_B) + V(\Phi_B) + C_{\rm int}(\Phi_B) + \int [R(\Phi) - C(\Phi)]\, d\Phi_{\rm rad},$$



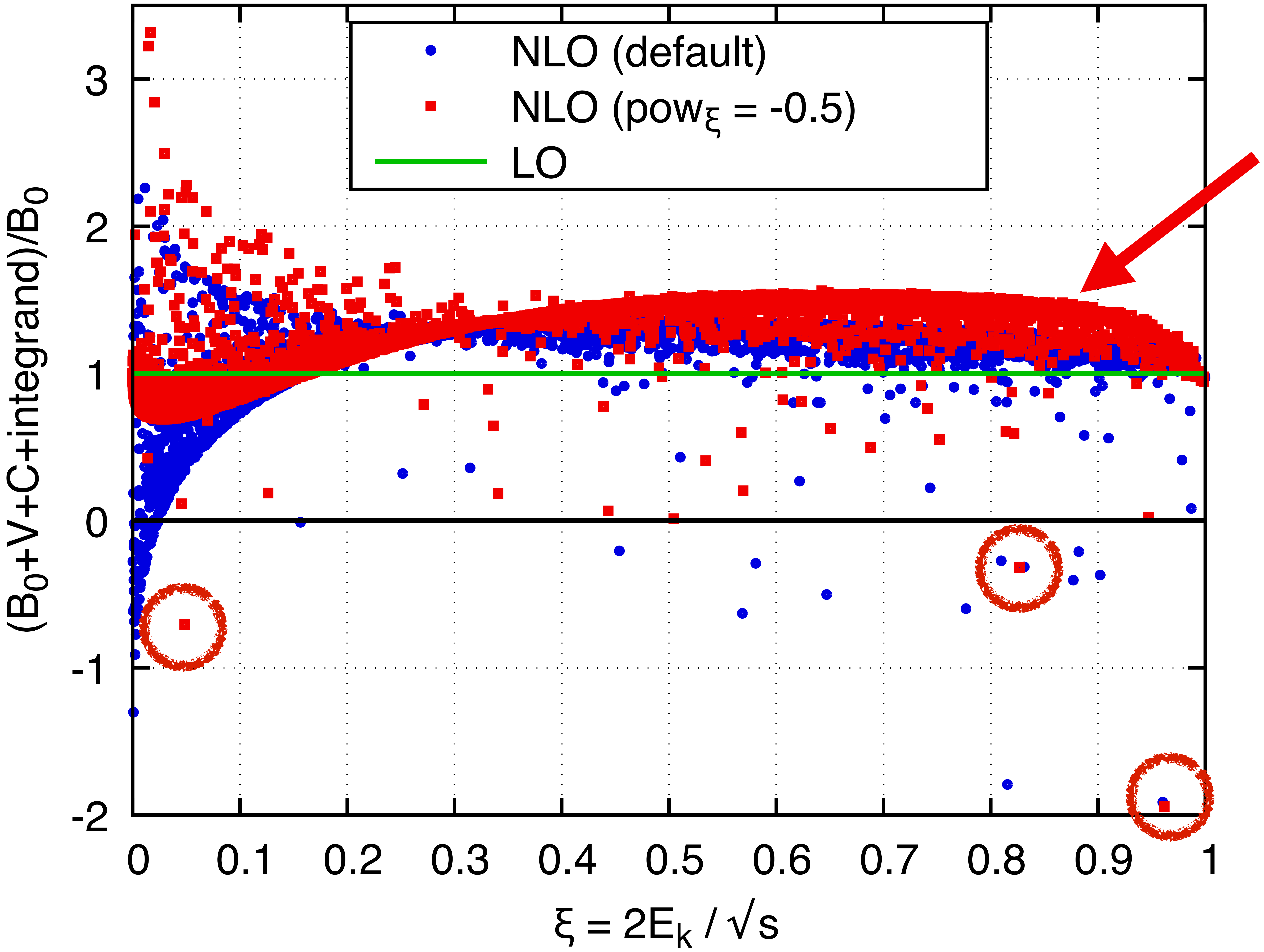
each point ≡ one Born phase space choice and one radiation phase space choice

Plotted as a function of one of the 3 real radiation phase space variables

Result should be $1 + \mathcal{O}(\alpha_s)$

But sometimes coefficient in front of α_s is large and result is negative

$$\bar{B}(\Phi_\mathrm{B}) = B_0(\Phi_\mathrm{B}) + V(\Phi_\mathrm{B}) + C_\mathrm{int}(\Phi_\mathrm{B}) + \int [R(\Phi) - C(\Phi)]\, d\Phi_\mathrm{rad},$$



reweighting of integration variables can help*

But still some negative-weight events and reweighting not always easy or successful

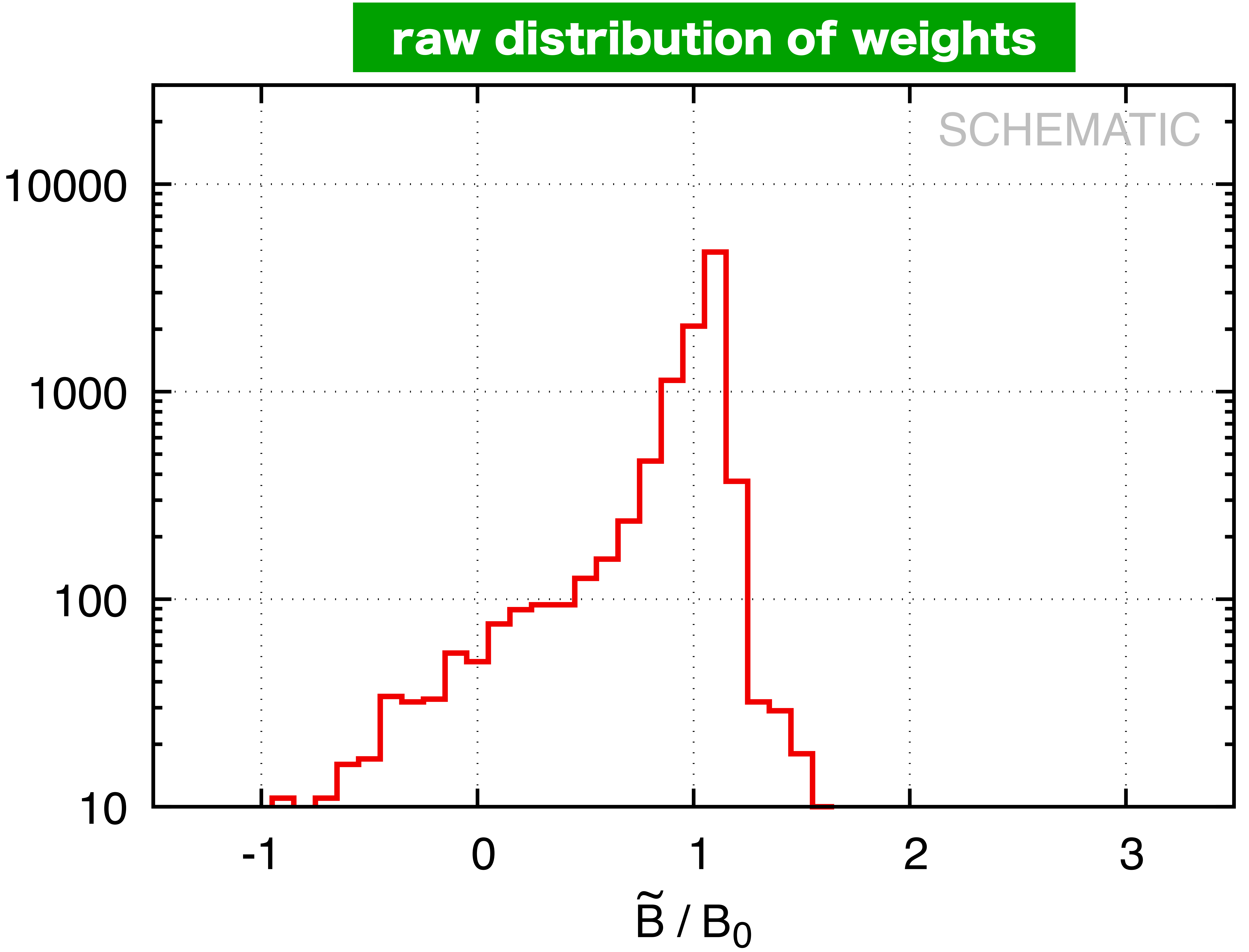
* recent proposals can be viewed as doing something similar: "Born spreading" [Frederix & Torrielli, 2310.04160] and "ARCANE" [Shyamsunda, 2502.08052, 2502.08053]

Key question

- Can you evaluate the following integral fast & reliably, and be both positive and NLO accurate

$$\bar{B}(\Phi_{\rm B}) = B_0(\Phi_{\rm B}) + \int \left[R(\Phi) - C(\Phi)\right] \, d\Phi_{\rm rad} \, ,$$

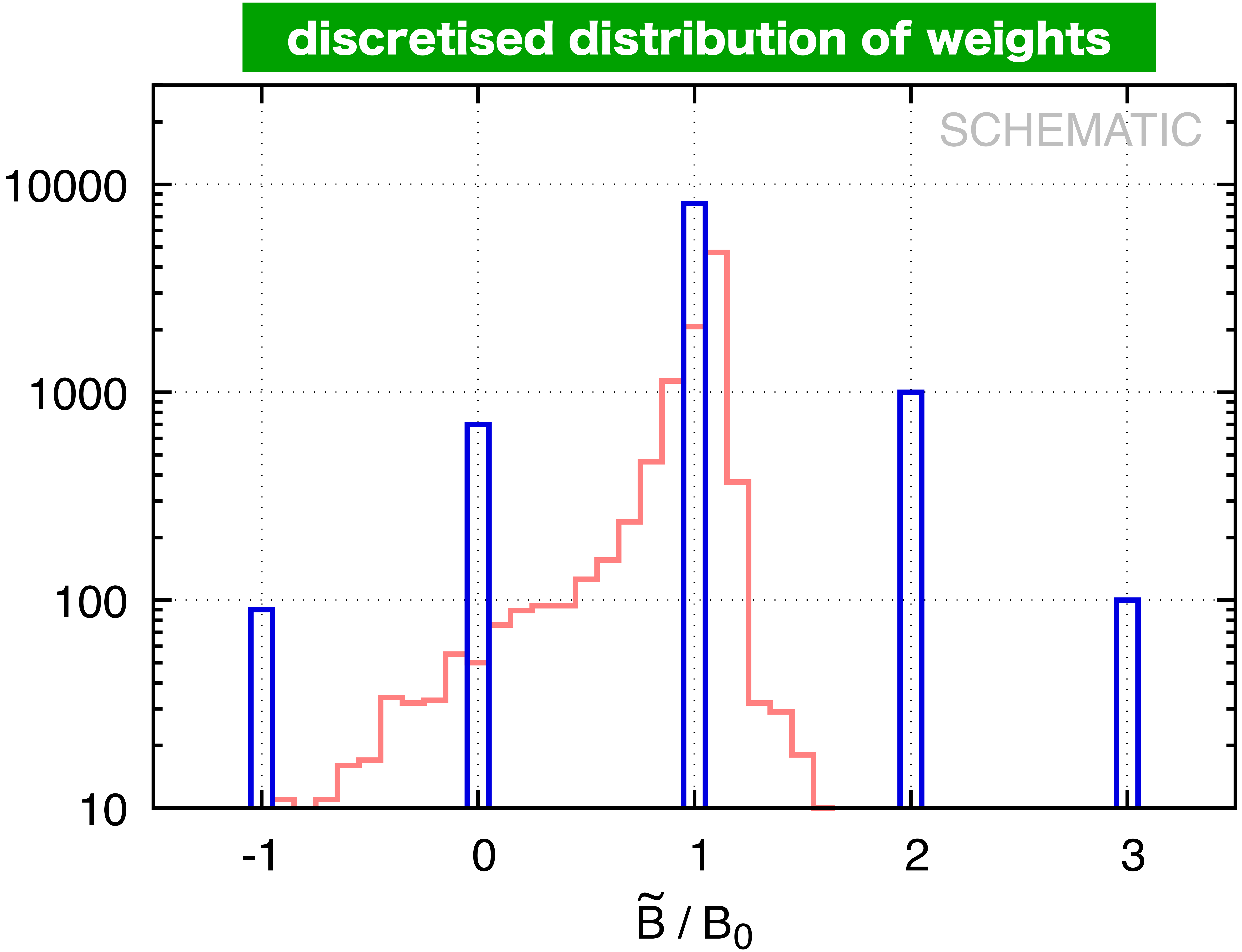
Could we just discard the negative events?



How do you guarantee that what you are discarding is genuinely beyond the NLO order you're trying to control?

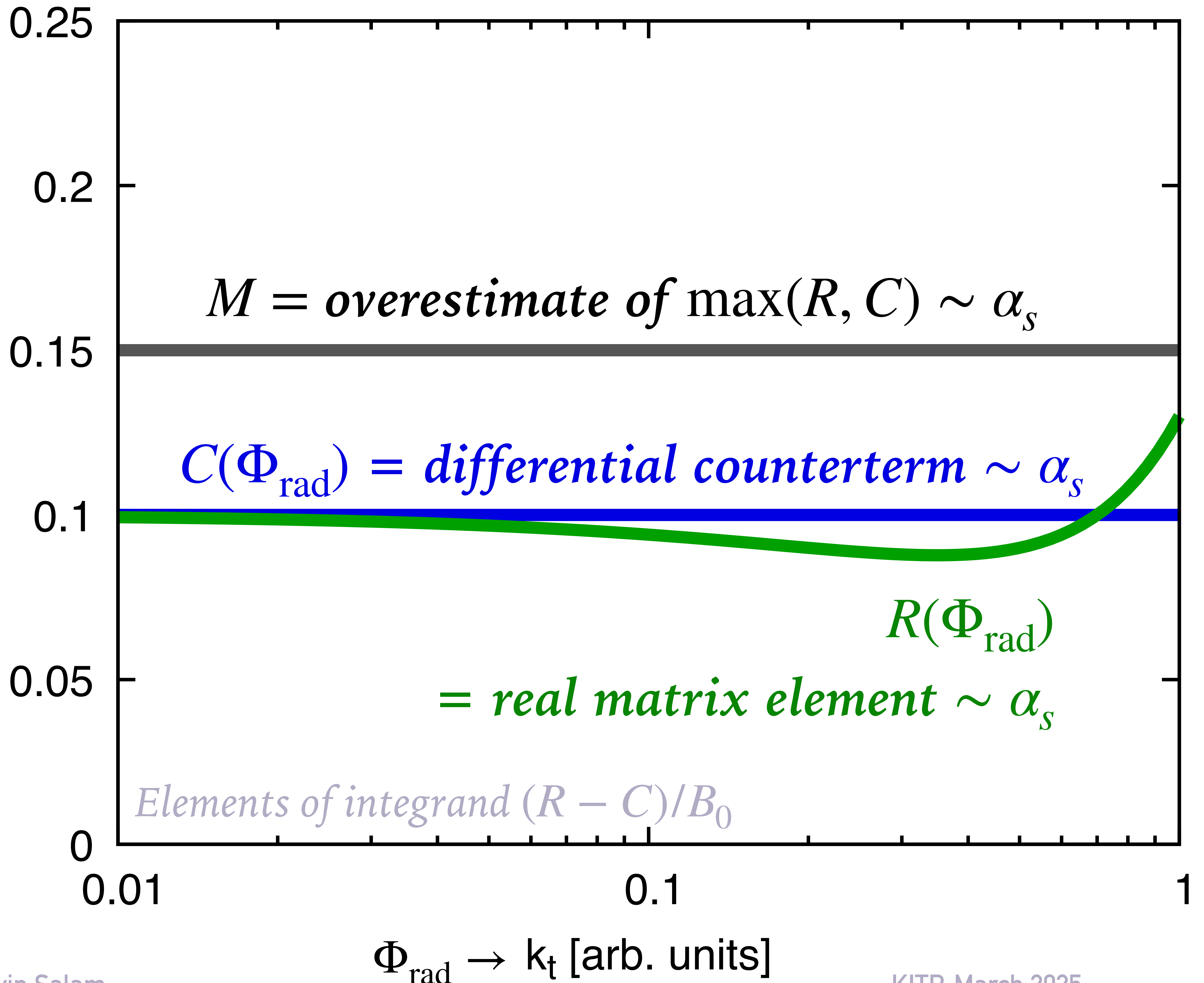
E.g. in one toy example discarding negative-weight events would give a spurious $\alpha_s^{3/2}$ contribution

Core idea: map the integral to an event-by event integer



this can be done in a way that the sum over the distribution gives the exact original answer

illustrate algorithm for one dimension of real phase space (k_t)



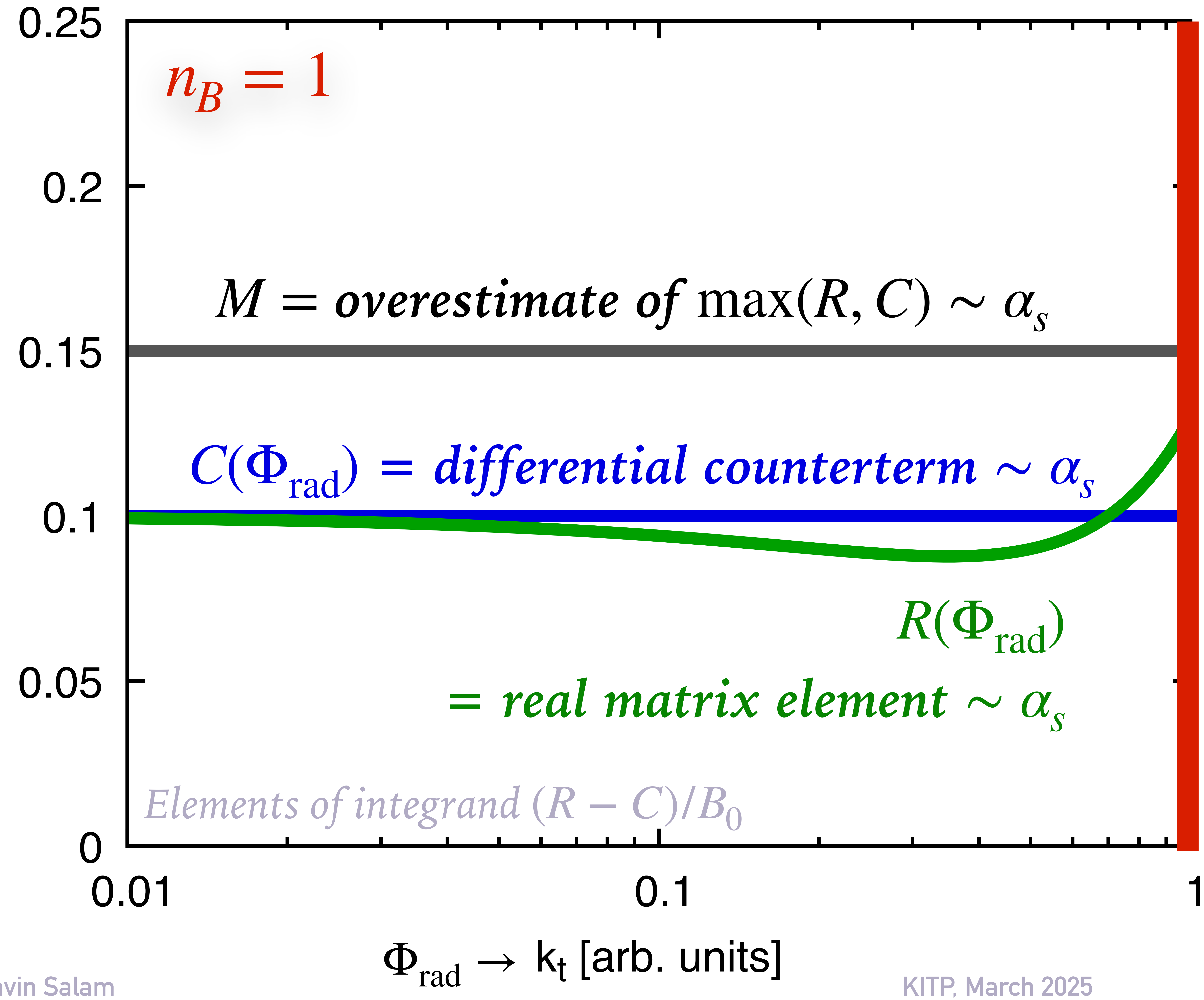
Start with

- normalisation $n_B = 1$
- k_t at max allowed value

Run the following loop:

- From $\ln k_t$ subtract a random amount sampled from $e^{-M/B_0 \ln 1/k_t}$
- With probability $r < \dfrac{|R - C|}{M}$
  - if $R > C$: $n_B \rightarrow n_B + 1$,
  - if $R < C$: $n_B \rightarrow n_B - 1$

illustrate algorithm for one dimension of real phase space (k_t)



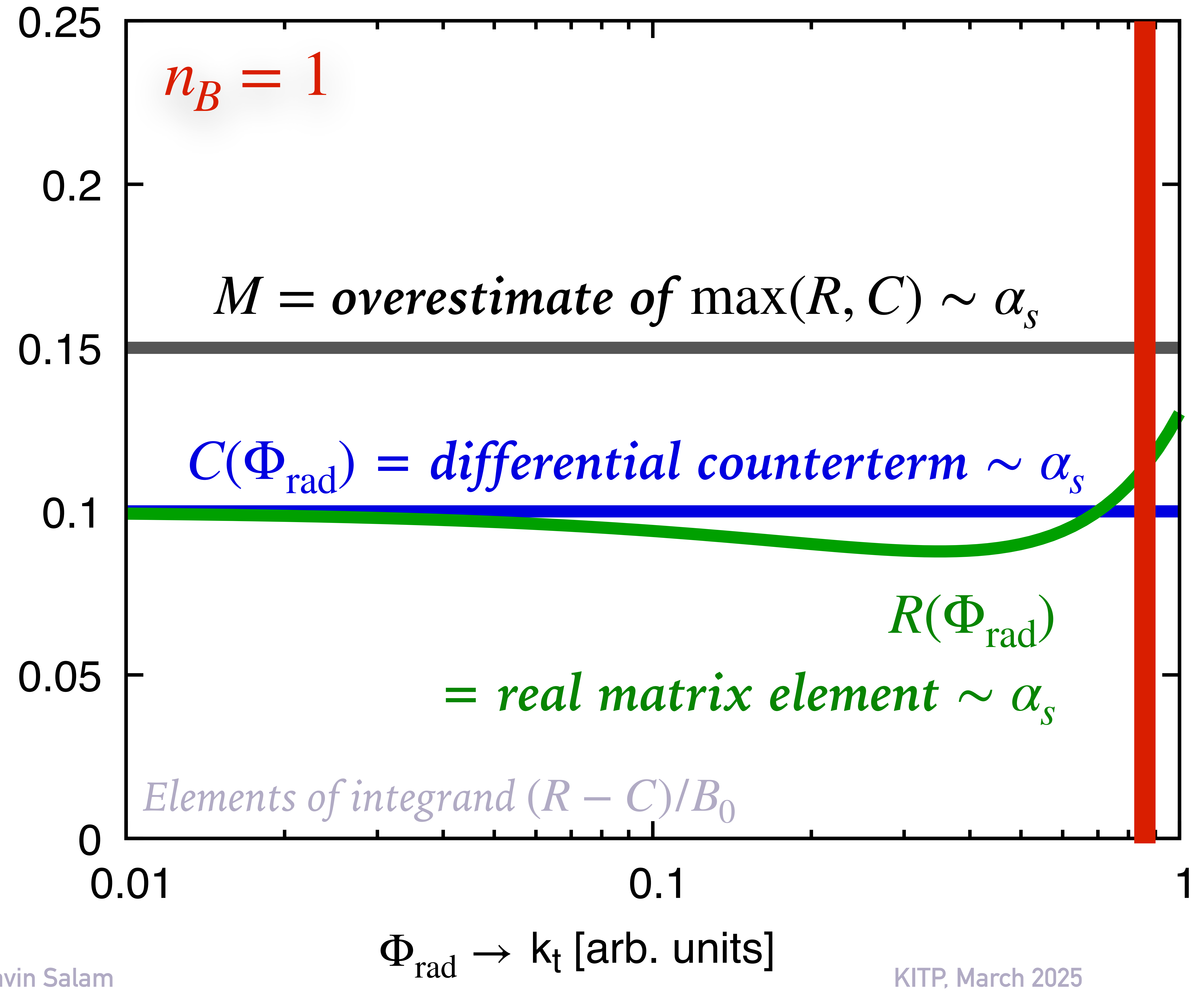
<u>Start with</u>

- normalisation $n_B = 1$
- k_t at max allowed value

<u>Run the following loop:</u>

- From $\ln k_t$ subtract a random amount sampled from $e^{-M/B_0 \ln 1/k_t}$
- With probability $r < \dfrac{|R - C|}{M}$
  - if $R > C$: $n_B \rightarrow n_B + 1$,
  - if $R < C$: $n_B \rightarrow n_B - 1$

illustrate algorithm for one dimension of real phase space (k_t)



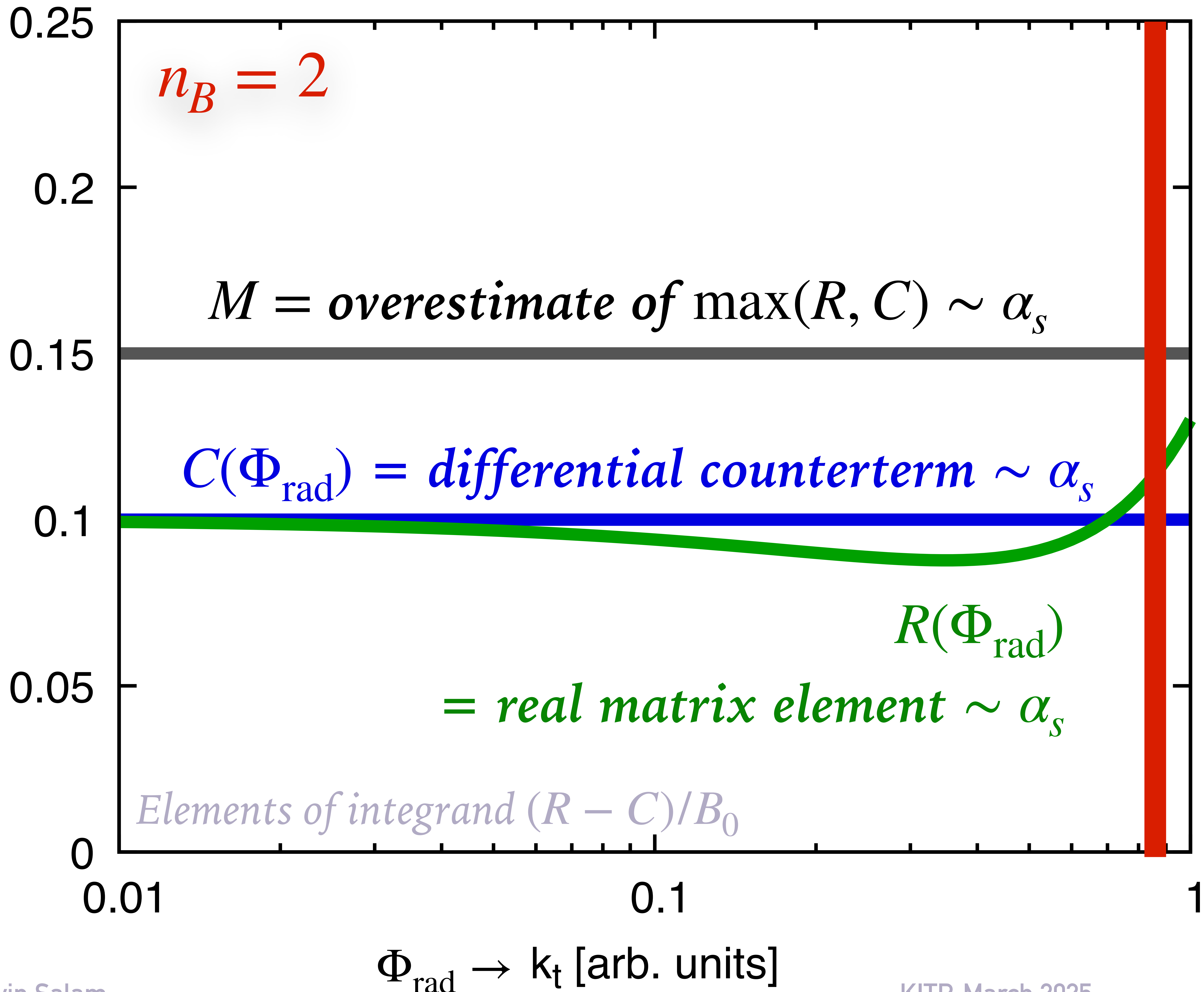
Start with

- normalisation $n_B = 1$
- k_t at max allowed value

Run the following loop:

- From $\ln k_t$ subtract a random amount sampled from $e^{-M/B_0 \ln 1/k_t}$
- With probability $r < \dfrac{|R - C|}{M}$
  - if $R > C$: $n_B \to n_B + 1$,
  - if $R < C$: $n_B \to n_B - 1$

illustrate algorithm for one dimension of real phase space (k_t)



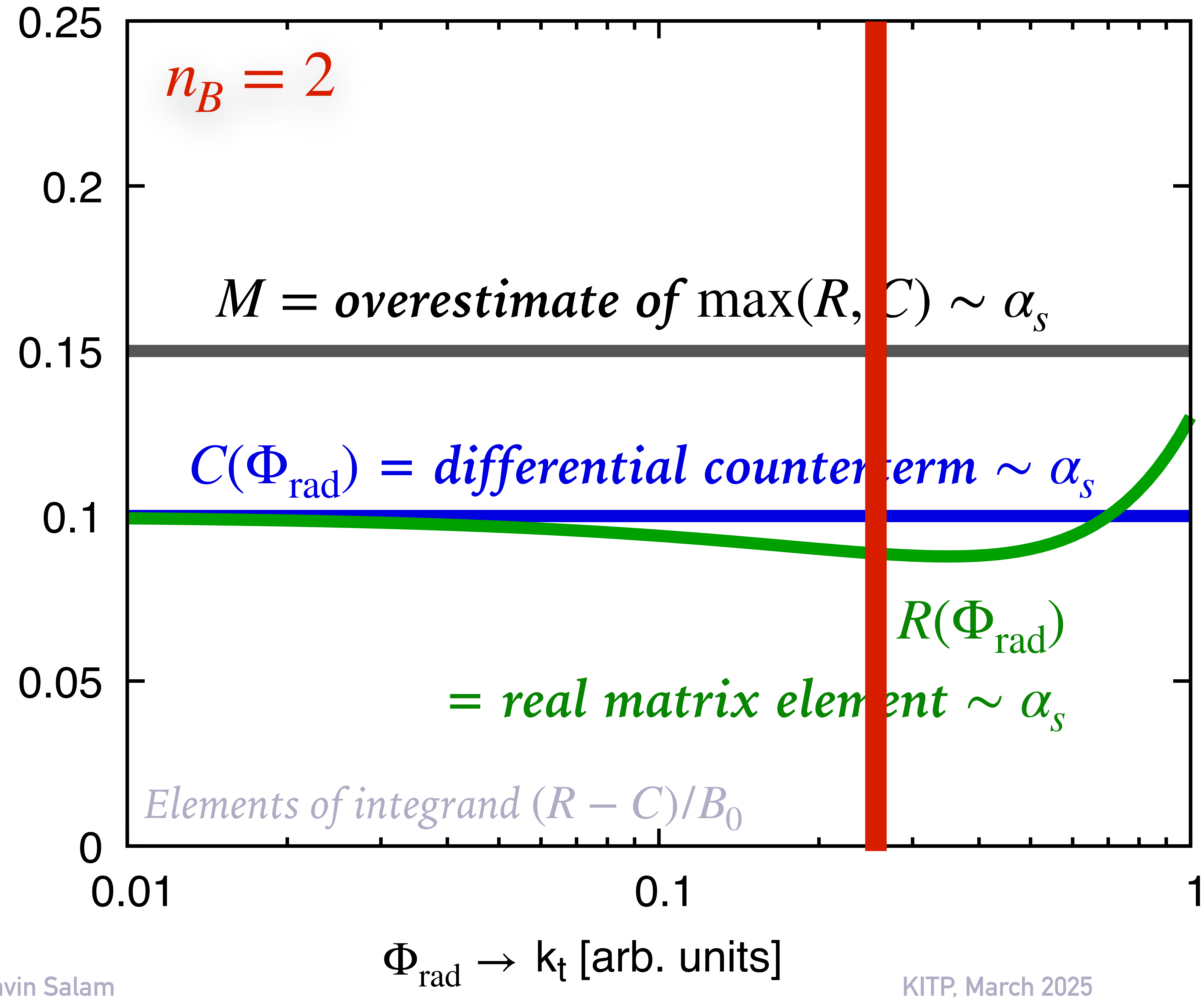
Start with

- normalisation $n_B = 1$
- k_t at max allowed value

Run the following loop:

- From $\ln k_t$ subtract a random amount sampled from $e^{-M/B_0 \ln 1/k_t}$
- With probability $r < \frac{|R - C|}{M}$
  - if $R > C$: $n_B \to n_B + 1$,
  - if $R < C$: $n_B \to n_B - 1$

illustrate algorithm for one dimension of real phase space (k_t)



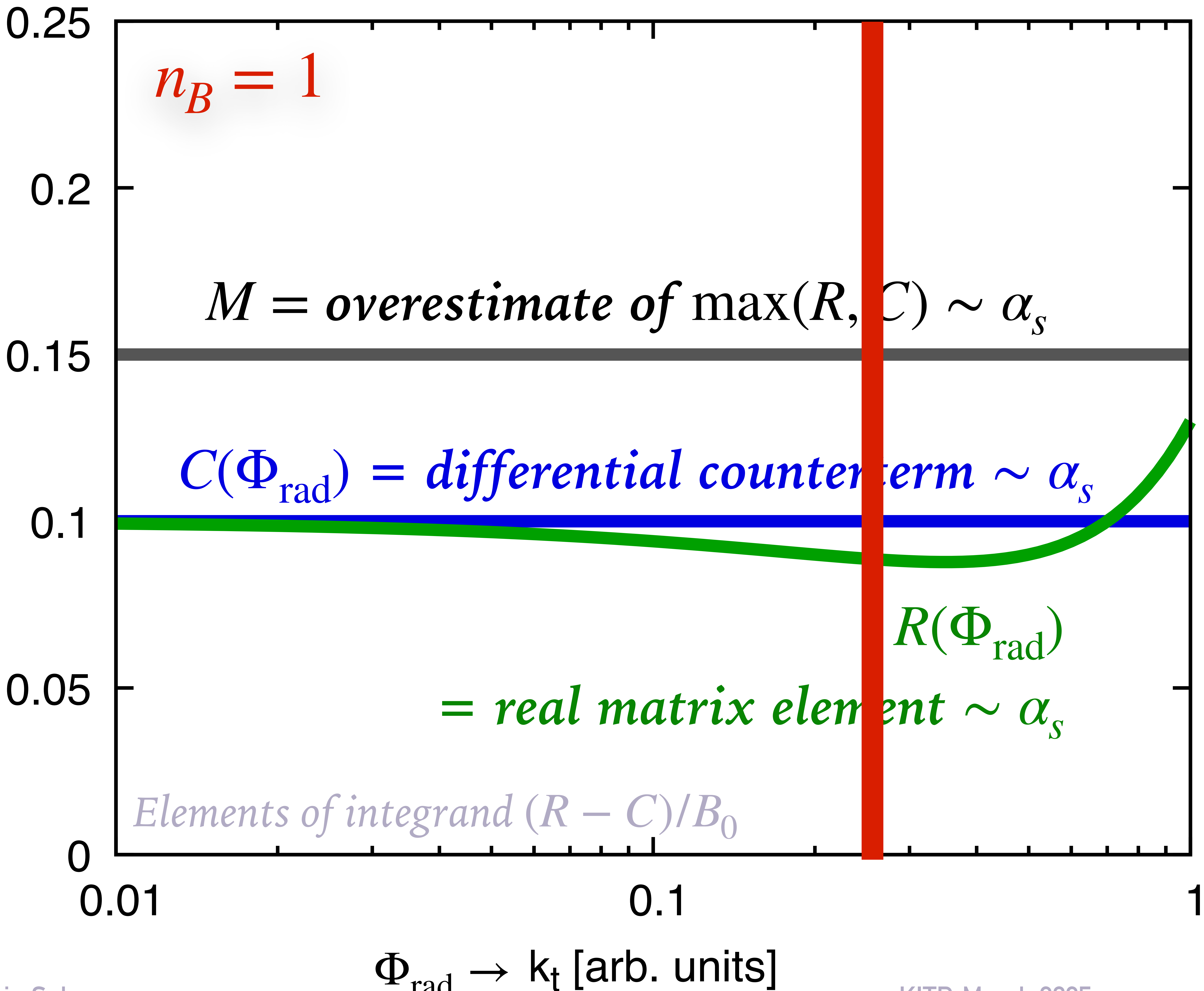
Start with

- normalisation $n_B = 1$
- k_t at max allowed value

Run the following loop:

- From $\ln k_t$ subtract a random amount sampled from $e^{-M/B_0 \ln 1/k_t}$
- With probability $r < \dfrac{|R - C|}{M}$
  - if $R > C$: $n_B \to n_B + 1$,
  - if $R < C$: $n_B \to n_B - 1$

illustrate algorithm for one dimension of real phase space (k_t)



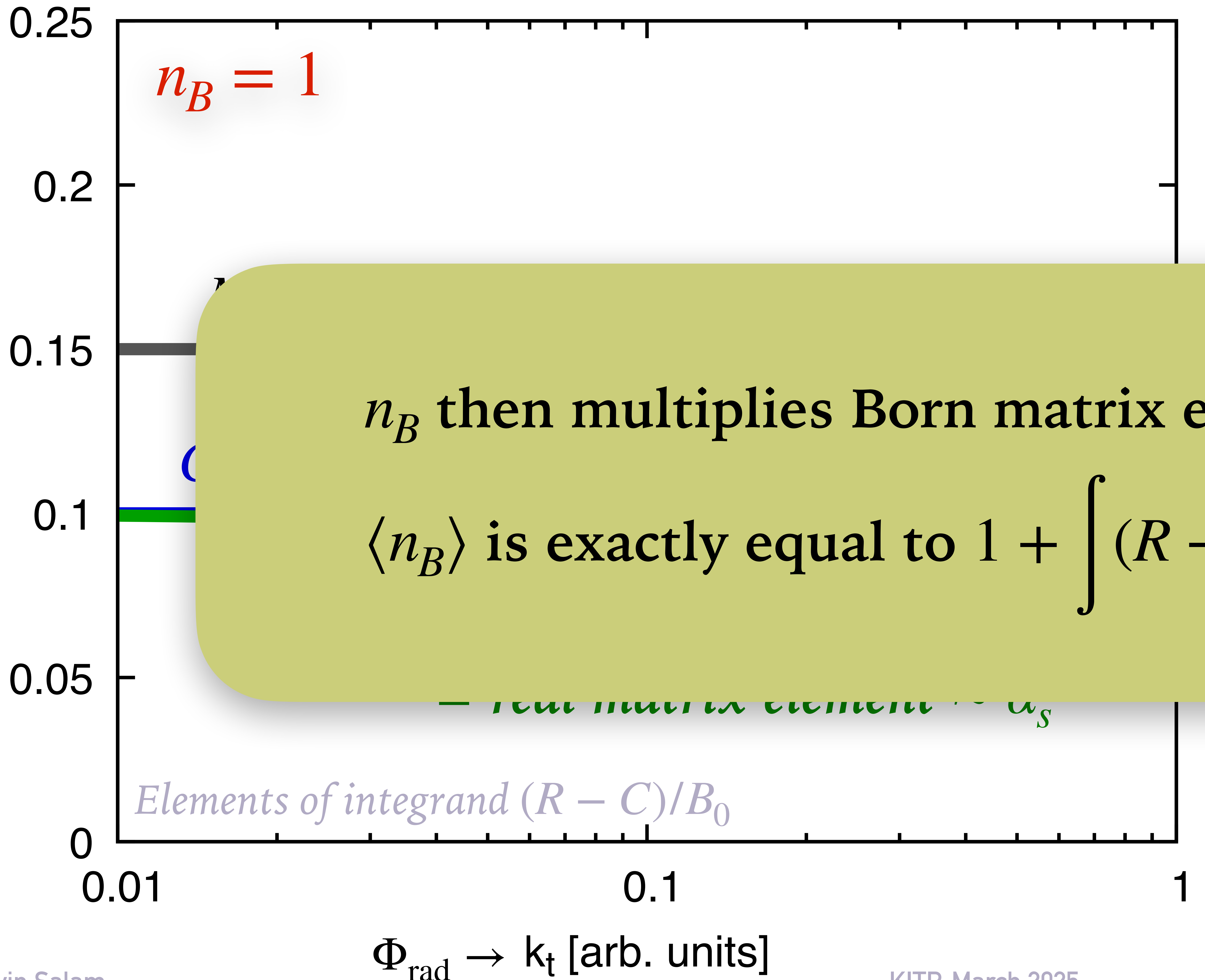
<u>Start with</u>

- normalisation $n_B = 1$
- k_t at max allowed value

<u>Run the following loop:</u>

- From $\ln k_t$ subtract a random amount sampled from $e^{-M/B_0 \ln 1/k_t}$
- With probability $r < \dfrac{|R - C|}{M}$
  - if $R > C$: $n_B \to n_B + 1$,
  - if $R < C$: $n_B \to n_B - 1$

illustrate algorithm for one dimension of real phase space (k_t)



n_B then multiplies Born matrix element $B_0(\Phi_B)$

$\langle n_B \rangle$ is exactly equal to $1 + \int (R - C)/B_0 \, d\Phi_{\rm rad}$

Start with

- normalisation $n_B = 1$
- k_t at max allowed value

p:

a random

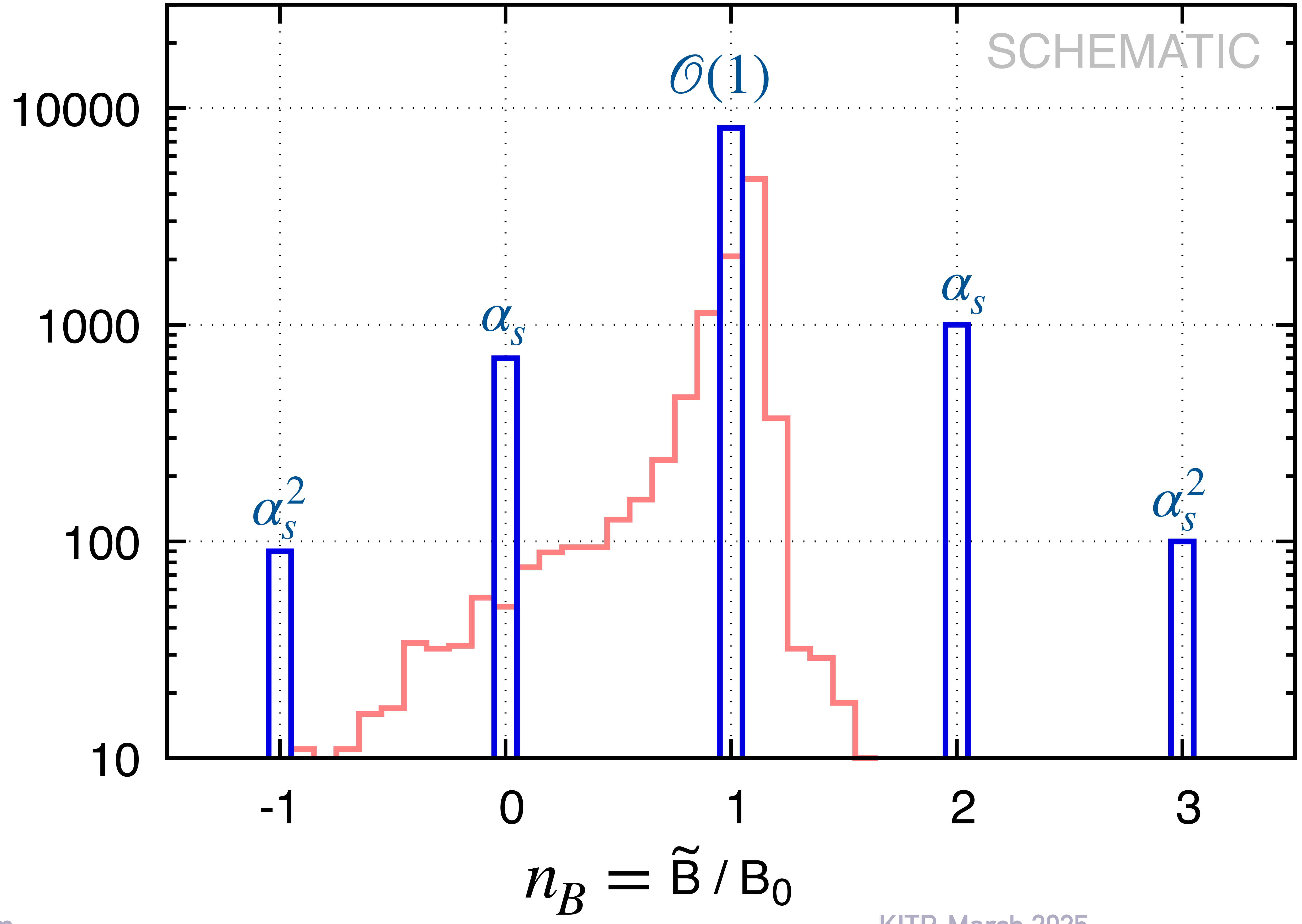
om $e^{-M/B_0 \ln 1/k_t}$

$< \frac{|R - C|}{M}$

- if $R > C$: $n_B \to n_B + 1$,
- if $R < C$: $n_B \to n_B - 1$

Robust positivity with NLO accuracy (spurious terms from NNLO)

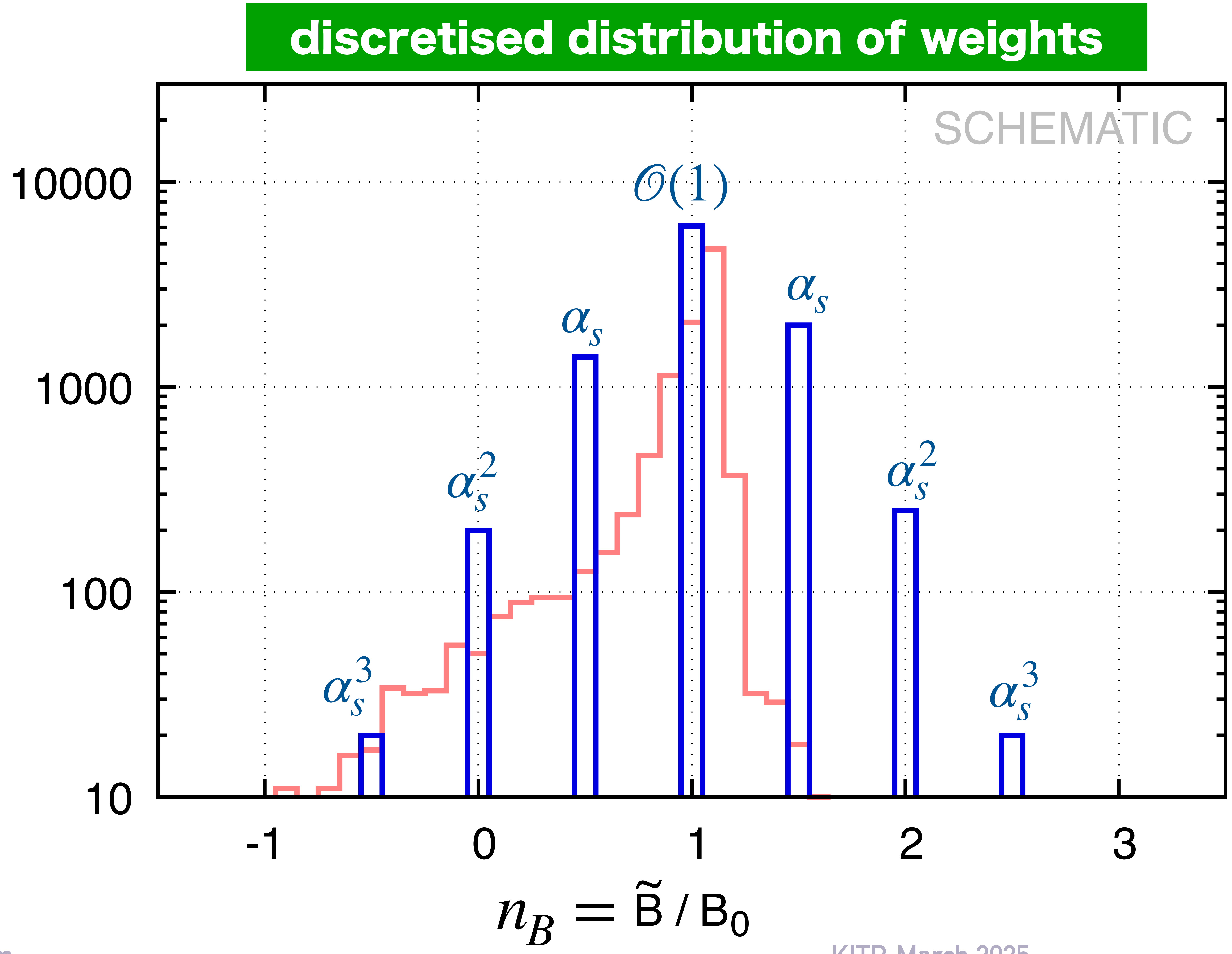
discretised distribution of weights



Now if you discard negative-weight events you have a guarantee that you only change the result at NNLO or beyond

Because each decrement of n_B costs a power of α_s

Robust positivity with NLO accuracy (spurious terms from N3LO)



Multiply M, R, C in the algorithm by factor p (here $p = 2$)

Increment n_B by $\pm\frac{1}{p}$

Algorithm gives exactly the same $\langle n_B \rangle$

Keeping only positive-weight events changes integral by just α_s^{p+1}

This is a "foundation" algorithm — can be adapted in many ways

- **ESME** ≡ Exponentiated Subtraction for Matching Events
- Our implementation actually uses a variant that handles real emissions and NLO normalisation simultaneously (more efficient, only a bit more complex)

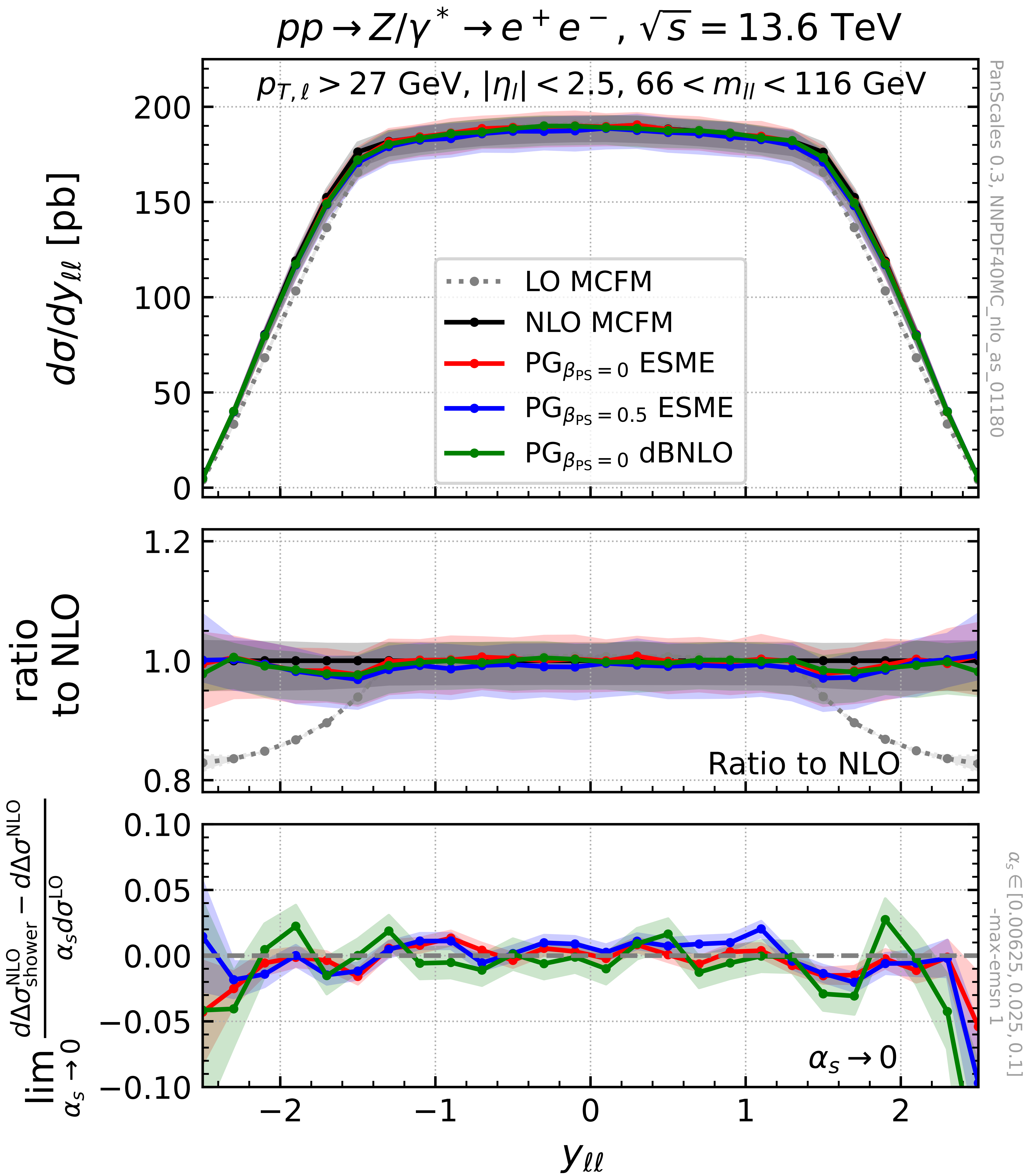
Algorithm Stream 1 (ESME) Born + NLO rejection

Generate Born event according to $\bar{B}_C$ distribution and set $v = v_{\max}$
while $v > v_{\min}$ **do**
    generate next v and Φ_2 according to Sudakov with density $\rho(v) d\ln v$, Eq. (3.6)
    generate random number $0 < r < 1$
    if $C(\Phi) > R(\Phi)$ **then**
        if $r > C(\Phi)/M(\Phi)$: veto emission
        else if $r > R(\Phi)/M(\Phi)$: **return** *reject event*
        else: accept emission and **return** *continue shower, accept event*
    else
        if $r > C(\Phi)/M(\Phi)$: veto emission
        else: accept emission and **return** *continue shower, accept event*
return *accept event*

Algorithm Stream 2 (ESME) NLO addition

Generate Born event according to $\bar{B}_C$ (or B_0) distribution and set $v = v_{\max}$
while $v > v_{\min}$ **do**
    generate next v and Φ_2 according to Sudakov with density $\rho(v) d\ln v$, Eq. (3.6)
    generate random number $0 < r < 1$
    if $C(\Phi) > R(\Phi)$ **then**
        if $r > R(\Phi)/M(\Phi)$: veto emission
        else: **return** *reject event*
    else
        if $r > R(\Phi)/M(\Phi)$: veto emission
        else if $r > C(\Phi)/M(\Phi)$: accept emsn, **return** *continue shower, accept event*
        else: **return** *reject event*
return *reject event*

Does ESME give the correct answer? → Yes

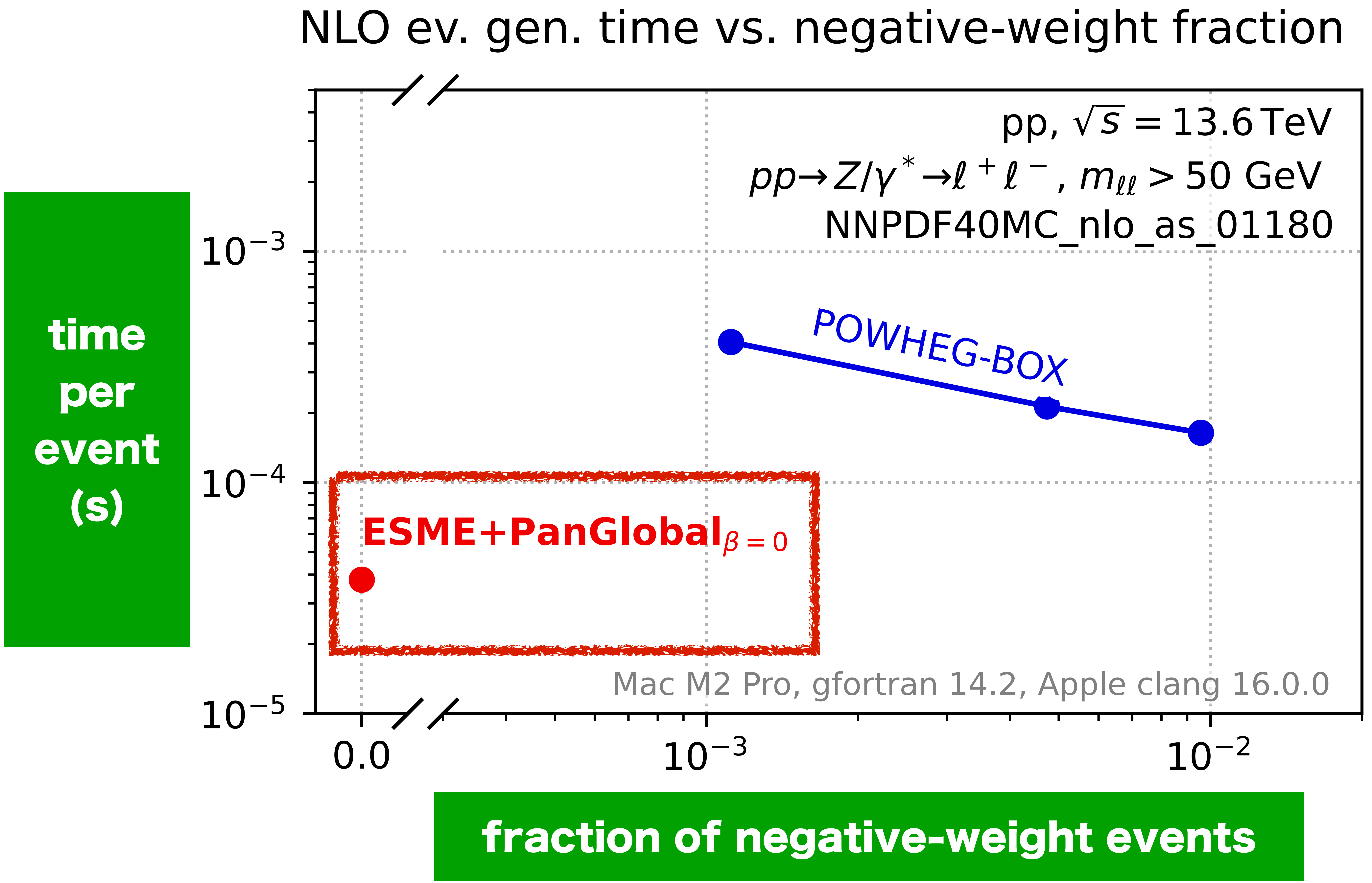


Comparison to exact NLO
(and to more standard method, here labelled dBNLO)

Ratio to exact NLO

Check of NLO coefficient

Is ESME fast enough?



- positive definite by construction
- turns out to be ~4× faster than fastest of public NLO tools (POWHEG-Boxv2)*
- NB: this is a simple process (Drell-Yan) — timing more critical for more complex processes

** with some effort having gone into optimising it*

Conclusions

- Positivity ***can*** be built into NLO calculations
- Price to pay is a change of higher terms beyond NLO
  - these higher order terms appear to be numerically modest
  - higher order terms anyway present for most observables in other NLO matching methods
  - underlying algorithm can be adapted to push them to arbitrary high order, e.g. for NNLO matching
- Underlying algorithms are simple, should be possible for other groups to try them out, also for more complex processes than the ones we studied
- Key step on path to making pQFT predictions simultaneously accurate and physical

Backup

Algorithm 1 General algorithm to convert NLO subtraction integral to integer

Set $n_\mathrm{B} = 1$ and $v = v_\mathrm{max}$
while $v > v_\mathrm{min}$ **do**
  generate next v and Φ_2 according to Sudakov with density $\rho(v) d\ln v$, Eq. (3.6)
  generate random number $0 < r < 1$
  if $r < |R(\Phi) - C(\Phi)|/M(\Phi)$ **then**
    if $R(\Phi) > C(\Phi)$: $n_\mathrm{B} \to n_\mathrm{B} + 1$
    else: $n_\mathrm{B} \to n_\mathrm{B} - 1$
return n_B

$$d\Phi_\mathrm{rad} \to \frac{dv}{v} d\Phi_2$$

$$\Delta(v) = \exp\left[-\int_v^{v_\mathrm{max}} \frac{dv'}{v'} \rho(v')\right]$$

$$\rho(v) = \int d\Phi_2 \, J \, \frac{M(\Phi)}{B_0(\Phi_\mathrm{B})}$$

$$M(\Phi) \geq \max[R(\Phi), C(\Phi)]$$

n_B then multiplies Born matrix element $B_0(\Phi_B)$

$\langle n_B \rangle$ is exactly equal to $1 + \underbrace{\int \frac{R - C}{B_0} d\Phi_\mathrm{rad}}_{\text{order } \alpha_s}$

Algorithm 1 General algorithm to convert NLO subtraction integral to integer

```
Set n_B = 1 and v = v_max
while v > v_min do
    generate next v and Φ_2 according to Sudakov with density ρ(v) d ln v, Eq. (3.6)
    generate random number 0 < r < 1
    if r < |R(Φ) − C(Φ)|/M(Φ) then
        if R(Φ) > C(Φ):  n_B → n_B + 1
        else:            n_B → n_B − 1
return n_B
```

$\mathcal{O}(\alpha_s)$

$$d\Phi_{\rm rad} \to \frac{dv}{v} d\Phi_2$$

$$\Delta(v) = \exp\left[-\int_v^{v_{\max}} \frac{dv'}{v'}\rho(v')\right]$$

$$\rho(v) = \int d\Phi_2\, J\, \frac{M(\Phi)}{B_0(\Phi_{\rm B})}$$

$$M(\Phi) \geq \max[R(\Phi), C(\Phi)]$$

n_B then multiplies Born matrix element $B_0(\Phi_B)$

$\langle n_B \rangle$ is exactly equal to $\displaystyle 1 + \underbrace{\int \frac{R-C}{B_0} d\Phi_{\rm rad}}_{\text{order } \alpha_s}$

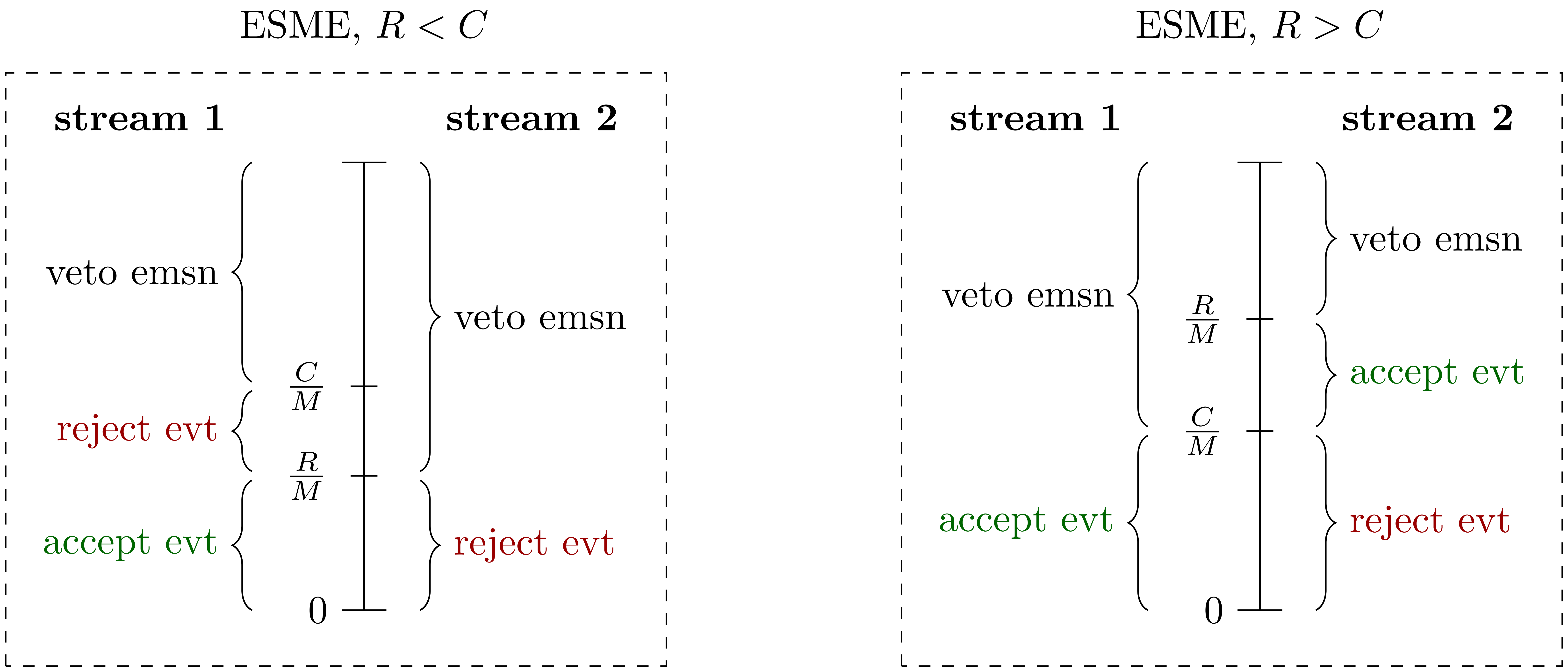


Figure 1: Simple illustration of the different possible actions in the two streams of the ESME algorithm with joint reals and subtractions. The actions are shown separately for the cases $R(\Phi) < C(\Phi)$ (left) and $R(\Phi) > C(\Phi)$ (right). In each case, when summing the two streams, one sees that the "accept evt" action occurs with total weight R/M. One can also verify that the contribution to the total event rate change relative to the $\bar{B}_C$ normalisation is $(R - C)/M$. Recall that the default action in stream 1 (2) is to accept (reject) the event if the shower scale reaches $v_{\min}$ — only when the action is different from the stream's default is the total event rate affected.

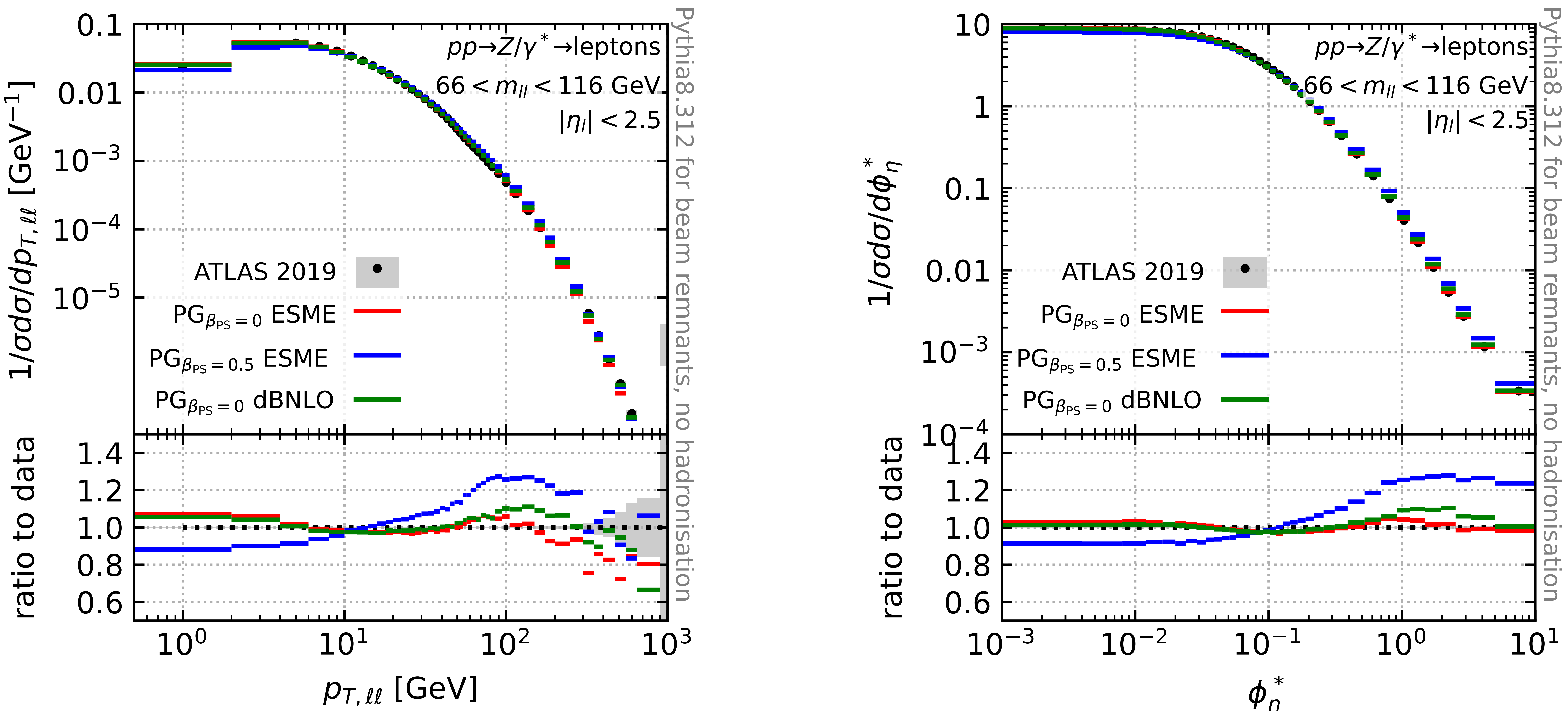
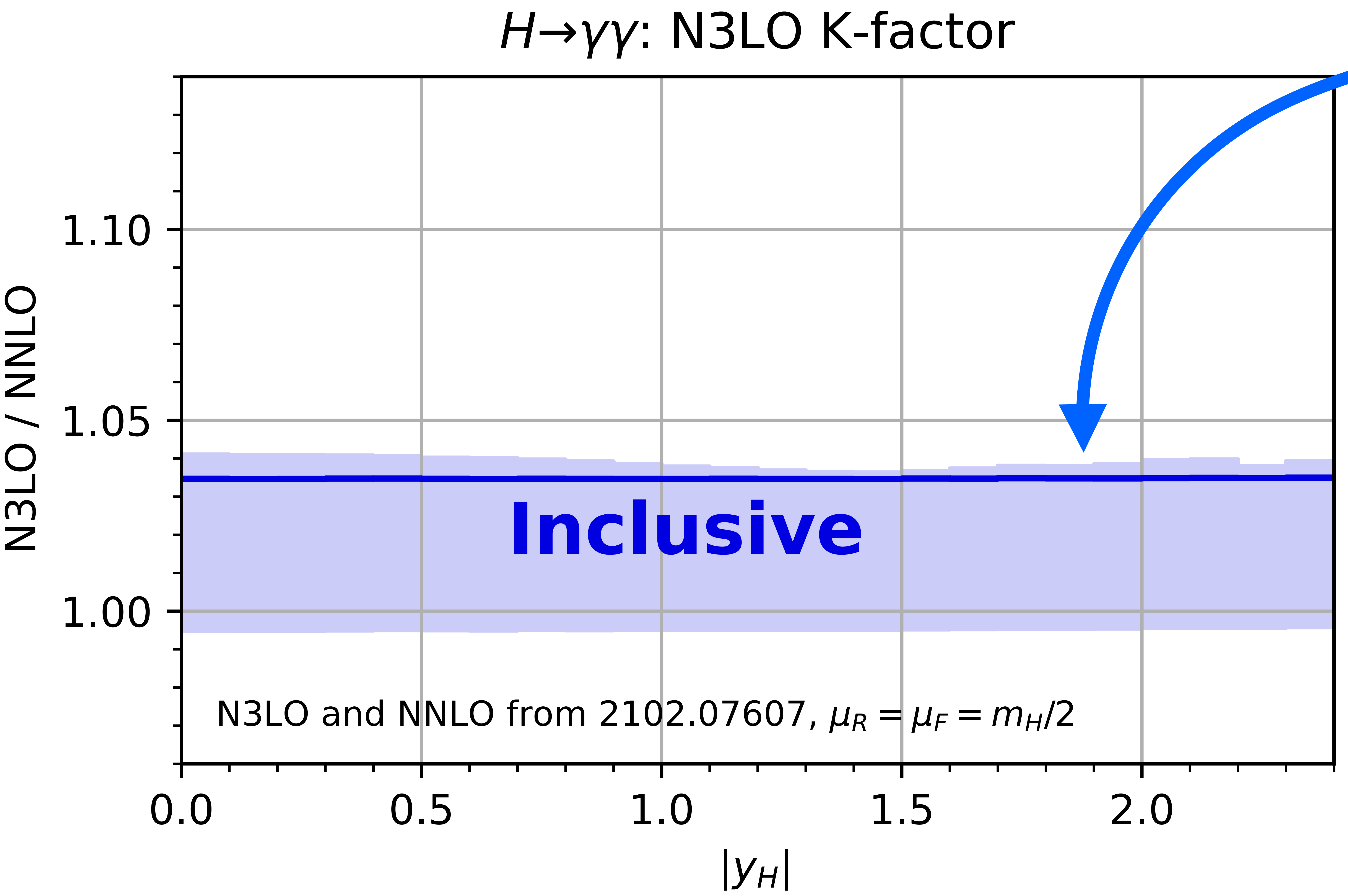


Figure 11: PanScales NLL+NLO matched showers, interfaced with `Pythia` [103], as compared to 13 TeV QED-Born di-lepton data from the ATLAS collaboration [104]. The left-hand plot is for the di-lepton transverse momentum distribution, while the right-hand plot is for the ϕ^*_η variable [105], cf. Eq. (5.1). In the Pythia interface, we include Pythia's primordial transverse momentum but not hadronisation, QED effects or multi-parton interactions.

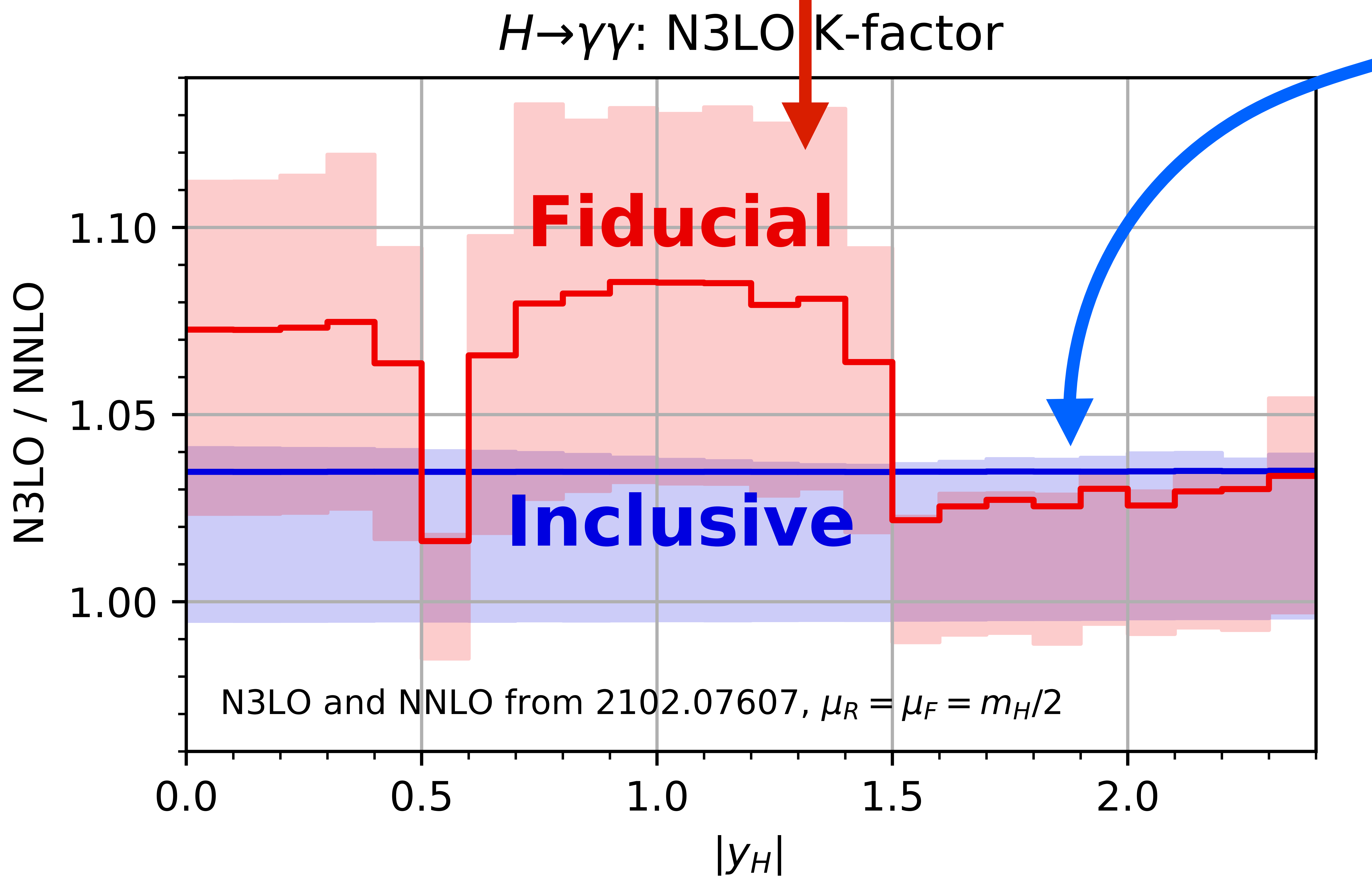
Recent surprise: H→γγ

inclusive N3LO σ uncertaities



Chen, Gehrmann, Glover, Huss, Mistlberger & Pelloni, 2102.07607

Recent surprise: H→γγ fiducial N3LO σ uncertainties ~2× greater than inclusive N3LO σ uncertaities



Chen, Gehrmann, Glover, Huss, Mistlberger & Pelloni, 2102.07607

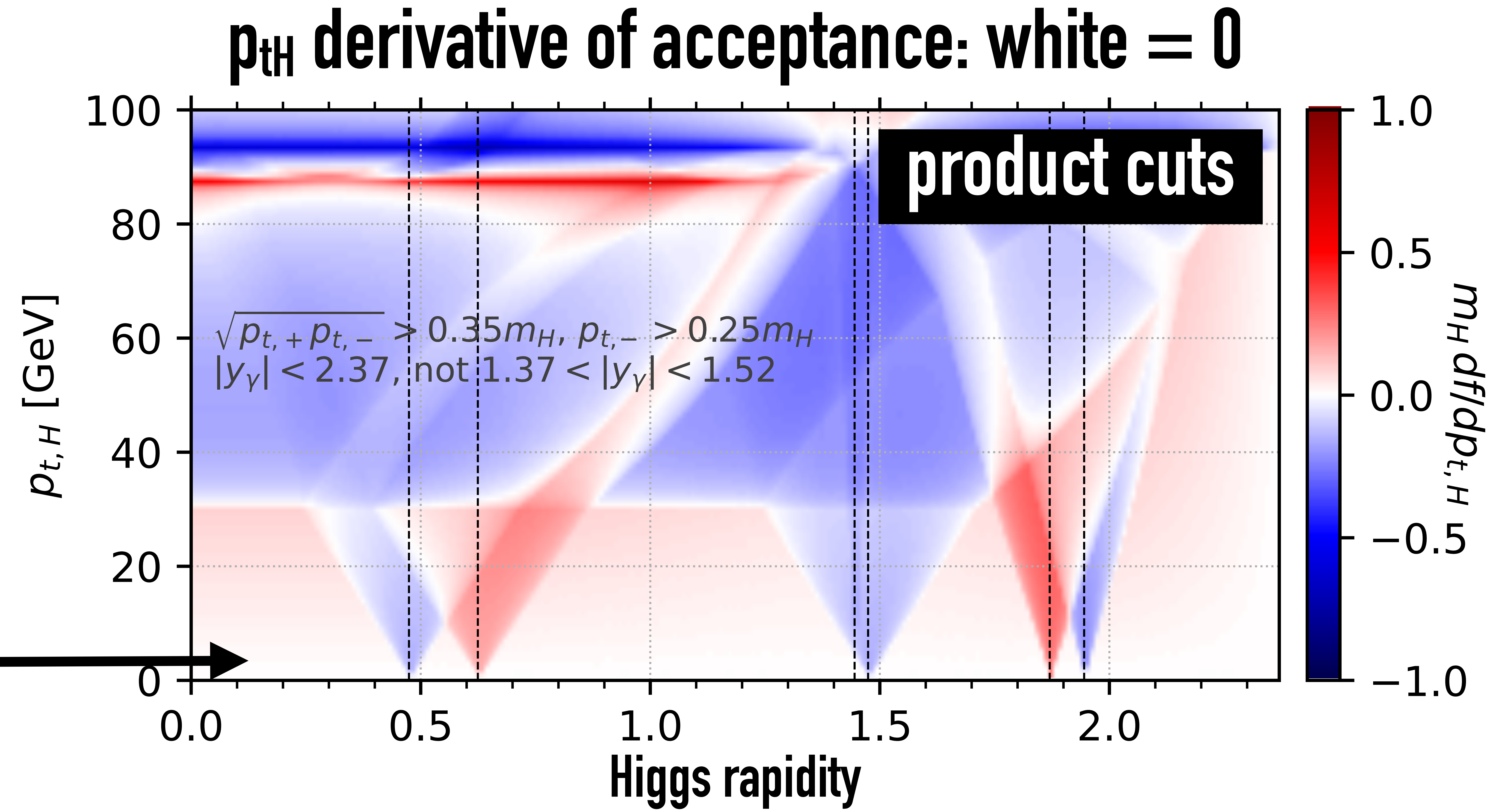
"Gold standard" fiducial cross section gives much worse prediction

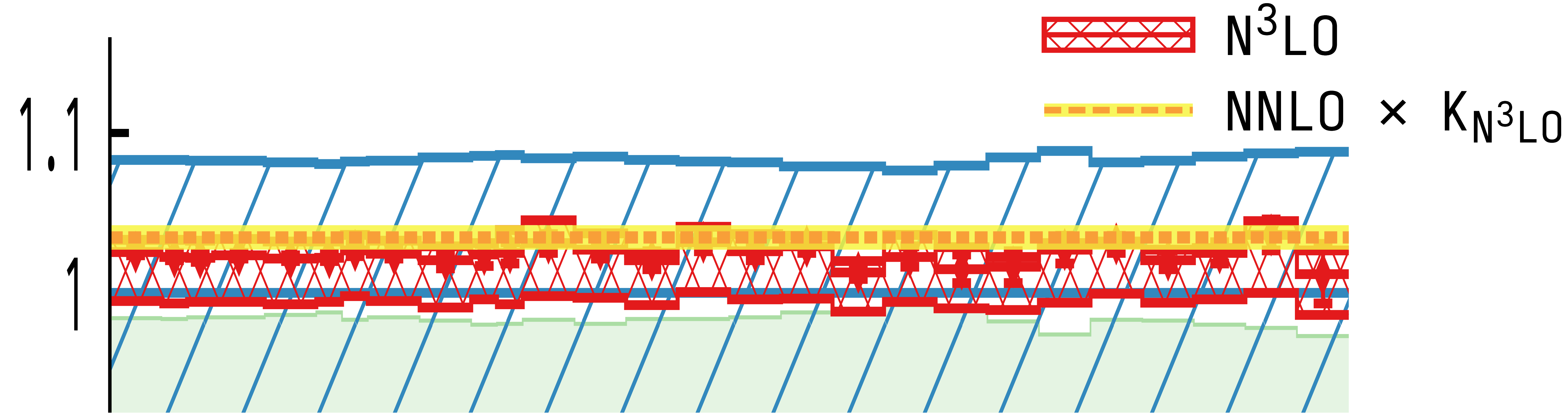
Why?
And can this be solved?

interplay with η_γ cuts

$f(p_{t,\mathrm{H}}, y_\mathrm{H})$ has **zero** linear $p_{t,\mathrm{H}}$ derivative at $p_{t,\mathrm{H}} = 0$

fixed-order perturbation theory will be fine





Huss et al preliminary @ Higgs 2021