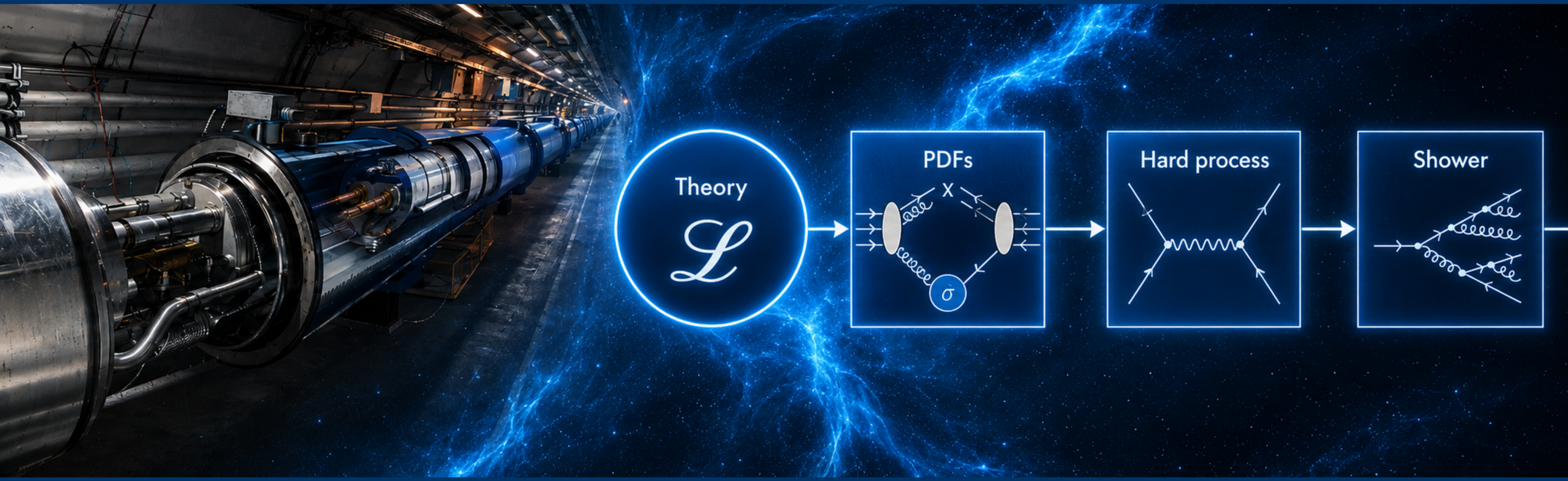
Simulation and Machine Learning Methods in HEP: Automated Computational Tools.



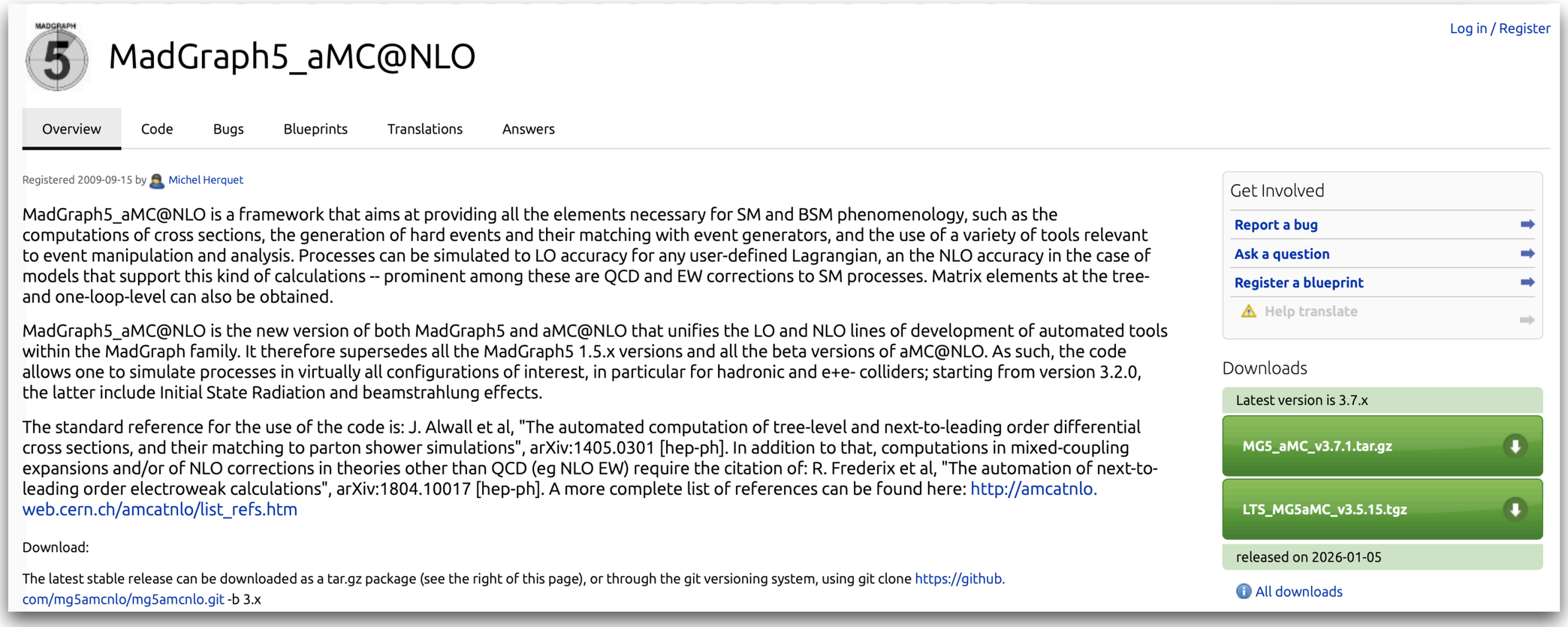
UNIVERSITÀ DEGLI STUDI DI MILANO

INFN MILANO

Ramon Winterhalder

June 2026

Literature and wiki



MadGraph5_aMC@NLO

Log in / Register

Overview | Code | Bugs | Blueprints | Translations | Answers

Registered 2009-09-15 by Michel Herquet

MadGraph5_aMC@NLO is a framework that aims at providing all the elements necessary for SM and BSM phenomenology, such as the computations of cross sections, the generation of hard events and their matching with event generators, and the use of a variety of tools relevant to event manipulation and analysis. Processes can be simulated to LO accuracy for any user-defined Lagrangian, an the NLO accuracy in the case of models that support this kind of calculations -- prominent among these are QCD and EW corrections to SM processes. Matrix elements at the tree- and one-loop-level can also be obtained.

MadGraph5_aMC@NLO is the new version of both MadGraph5 and aMC@NLO that unifies the LO and NLO lines of development of automated tools within the MadGraph family. It therefore supersedes all the MadGraph5 1.5.x versions and all the beta versions of aMC@NLO. As such, the code allows one to simulate processes in virtually all configurations of interest, in particular for hadronic and e+e- colliders; starting from version 3.2.0, the latter include Initial State Radiation and beamstrahlung effects.

The standard reference for the use of the code is: J. Alwall et al, "The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations", arXiv:1405.0301 [hep-ph]. In addition to that, computations in mixed-coupling expansions and/or of NLO corrections in theories other than QCD (eg NLO EW) require the citation of: R. Frederix et al, "The automation of next-to-leading order electroweak calculations", arXiv:1804.10017 [hep-ph]. A more complete list of references can be found here: http://amcatnlo.web.cern.ch/amcatnlo/list_refs.htm

Download:

The latest stable release can be downloaded as a tar.gz package (see the right of this page), or through the git versioning system, using git clone https://github.com/mg5amcnlo/mg5amcnlo.git -b 3.x

Get Involved

- Report a bug
- Ask a question
- Register a blueprint
- Help translate

Downloads

Latest version is 3.7.x

MG5_aMC_v3.7.1.tar.gz

LTS_MG5aMC_v3.5.15.tgz

released on 2026-01-05

All downloads

Literature and wiki

MadGraph5_aMC@NLO

Overview | Code | Bugs | Blueprints | Translations | Answers

Registered 2009-09-15 by Michel Herquet

MadGraph5_aMC@NLO is a framework that aims at providing all the elements necessary for S computations of cross sections, the generation of hard events and their matching with event to event manipulation and analysis. Processes can be simulated to LO accuracy for any user-de models that support this kind of calculations -- prominent among these are QCD and EW corre and one-loop-level can also be obtained.

MadGraph5_aMC@NLO is the new version of both MadGraph5 and aMC@NLO that unifies the within the MadGraph family. It therefore supersedes all the MadGraph5 1.5.x versions and all allows one to simulate processes in virtually all configurations of interest, in particular for had the latter include Initial State Radiation and beamstrahlung effects.

The standard reference for the use of the code is: J. Alwall et al, "The automated computation cross sections, and their matching to parton shower simulations", arXiv:1405.0301 [hep-ph]. In expansions and/or of NLO corrections in theories other than QCD (eg NLO EW) require the cit leading order electroweak calculations", arXiv:1804.10017 [hep-ph]. A more complete list of r web.cern.ch/amcatnlo/list_refs.htm

Download:

The latest stable release can be downloaded as a tar.gz package (see the right of this page), or through the git versio com/mg5amcnlo/mg5amcnlo.git -b 3.x

SciPost Physics Community Reports Submission

The Monte Carlo Ecosystem in High-Energy Physics: A Primer

Melissa van Beekveld[1], Enrico Bothmann[2], Andy Buckley[3], Christian Gütschow[4], Peter Skands[5], Ramon Winterhalder[6]

1 Nikhef, Amsterdam, the Netherlands
2 CERN, Geneva, Switzerland
3 School of Physics & Astronomy, University of Glasgow, UK
4 Centre for Advanced Research Computing, University College London, UK
5 School of Physics and Astronomy, Monash University, Melbourne, Australia
6 TIFLab, Università degli Studi di Milano & INFN Sezione di Milano, Italy

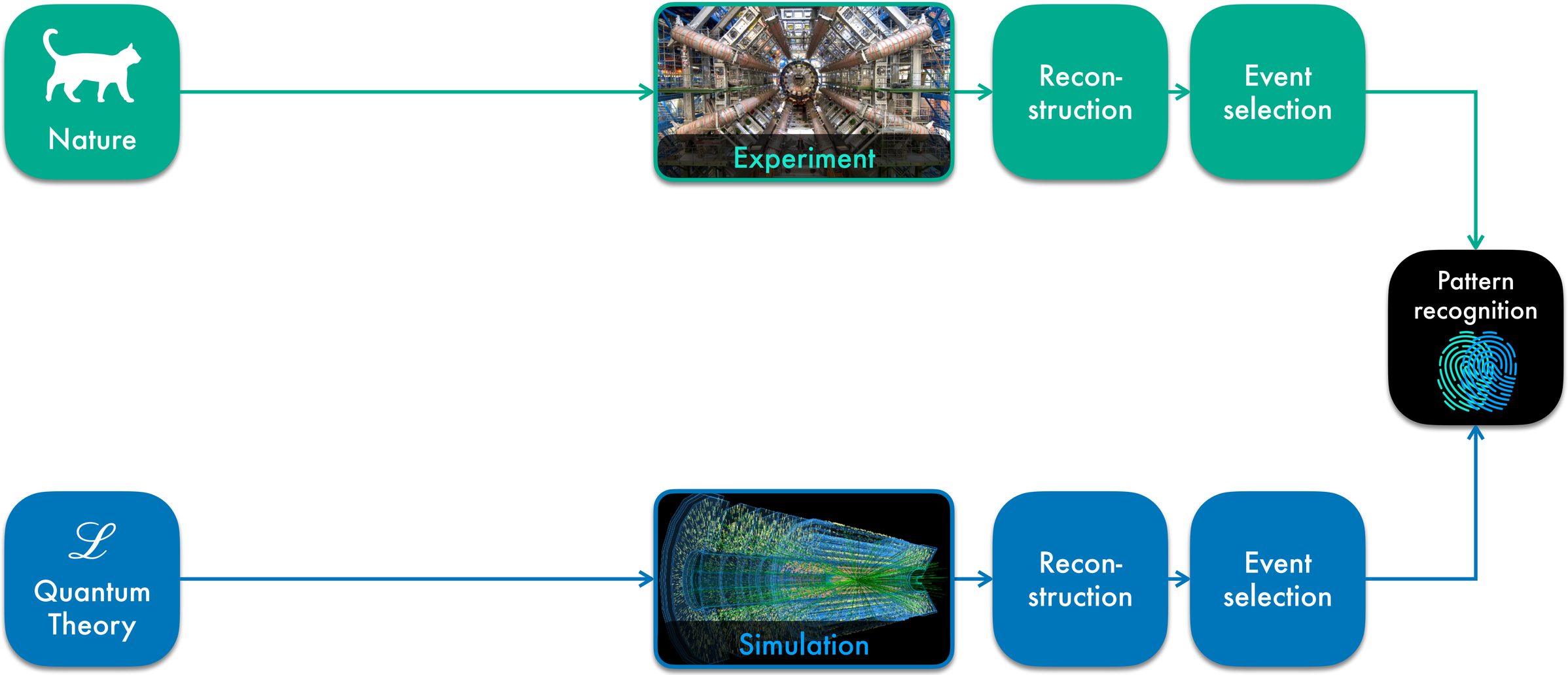
May 18, 2026

Abstract

Monte Carlo event generators are the central interface between theoretical calculations and experimental measurements in collider physics. Over several decades, a comprehensive and highly modular ecosystem of tools has developed around them, encompassing matrix-element calculations, parton showers, hadronisation models, and their integration with detector simulation, event-level analysis and statistical inference. While these tools are ubiquitous in modern research, the conceptual scope and technical structure of the full simulation chain can be challenging to navigate, particularly for researchers entering the field. In this primer, we provide a structured and up-to-date overview of the high-energy physics Monte Carlo ecosystem, focusing primarily on event-generator methodologies and their role within the broader collider workflow. We discuss the conceptual foundations of modern generators, the computational and organisational challenges of large-scale simulations, and the principles that enable interoperability and reproducibility across theory and experiment. We also examine the evolving computing landscape and sustainability considerations that will shape the future development of these tools. Aimed primarily at early-stage doctoral researchers while serving as a reference for the broader community, this article seeks to clarify architecture, methodology, and long-term trajectory of Monte Carlo event generation in collider physics.

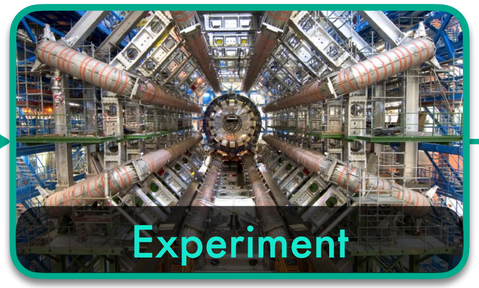
[2605.16036]

Bread and butter of collider physics



Bread and butter of collider physics

Nature



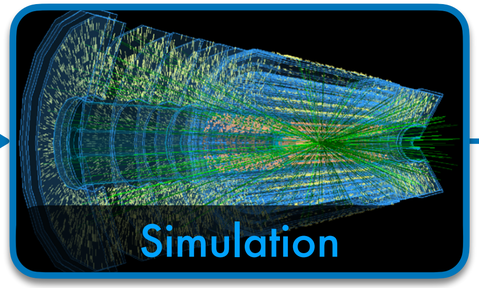
Recon-
struction

Event
selection

Pattern
recognition

New Physics?

$\mathscr{L}$
Quantum
Theory

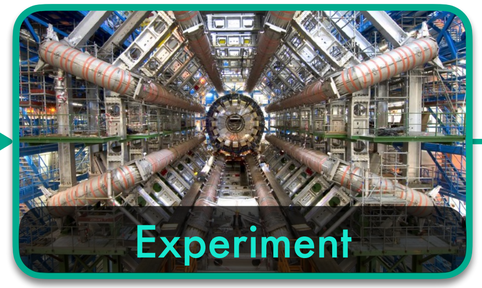


Recon-
struction

Event
selection

Bread and butter of collider physics

Nature



Recon-
struction

Event
selection

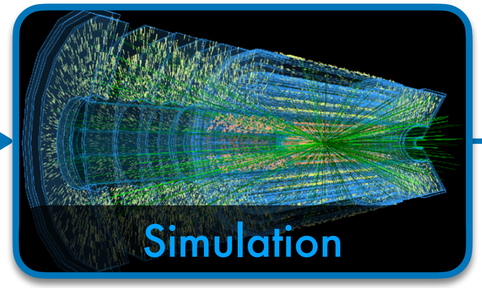
Pattern
recognition

Yes 🥳

New Physics?

$\mathscr{L}$

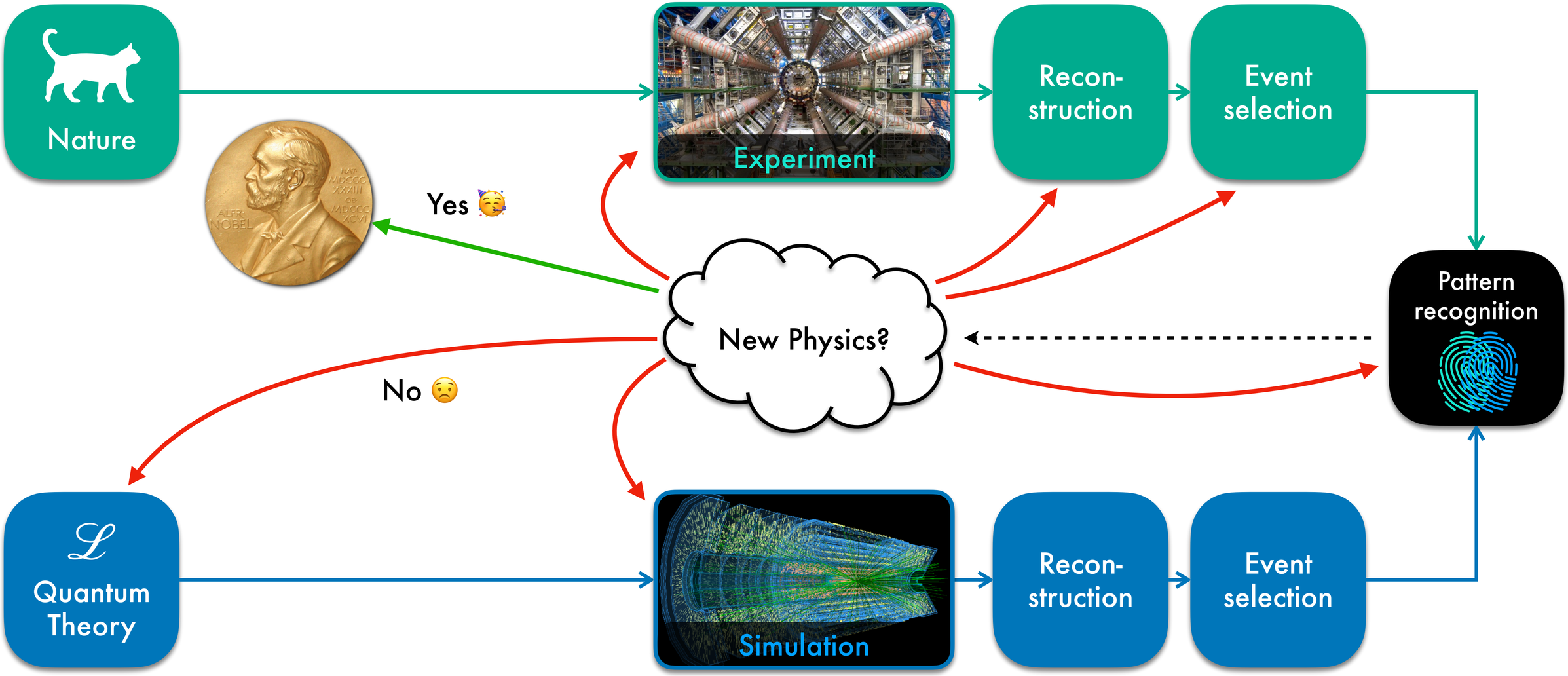
Quantum
Theory



Recon-
struction

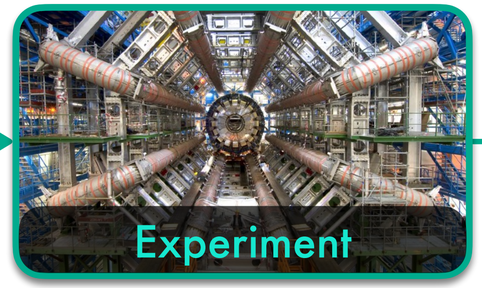
Event
selection

Bread and butter of collider physics



Collider Simulation Tools

Nature



Recon-
struction

Event
selection

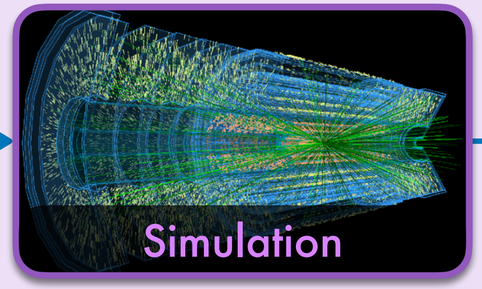
Pattern
recognition

Yes

No

These lectures focus on
Collider Simulation Tools

$\mathscr{L}$
Quantum
Theory



Recon-
struction

Event
selection

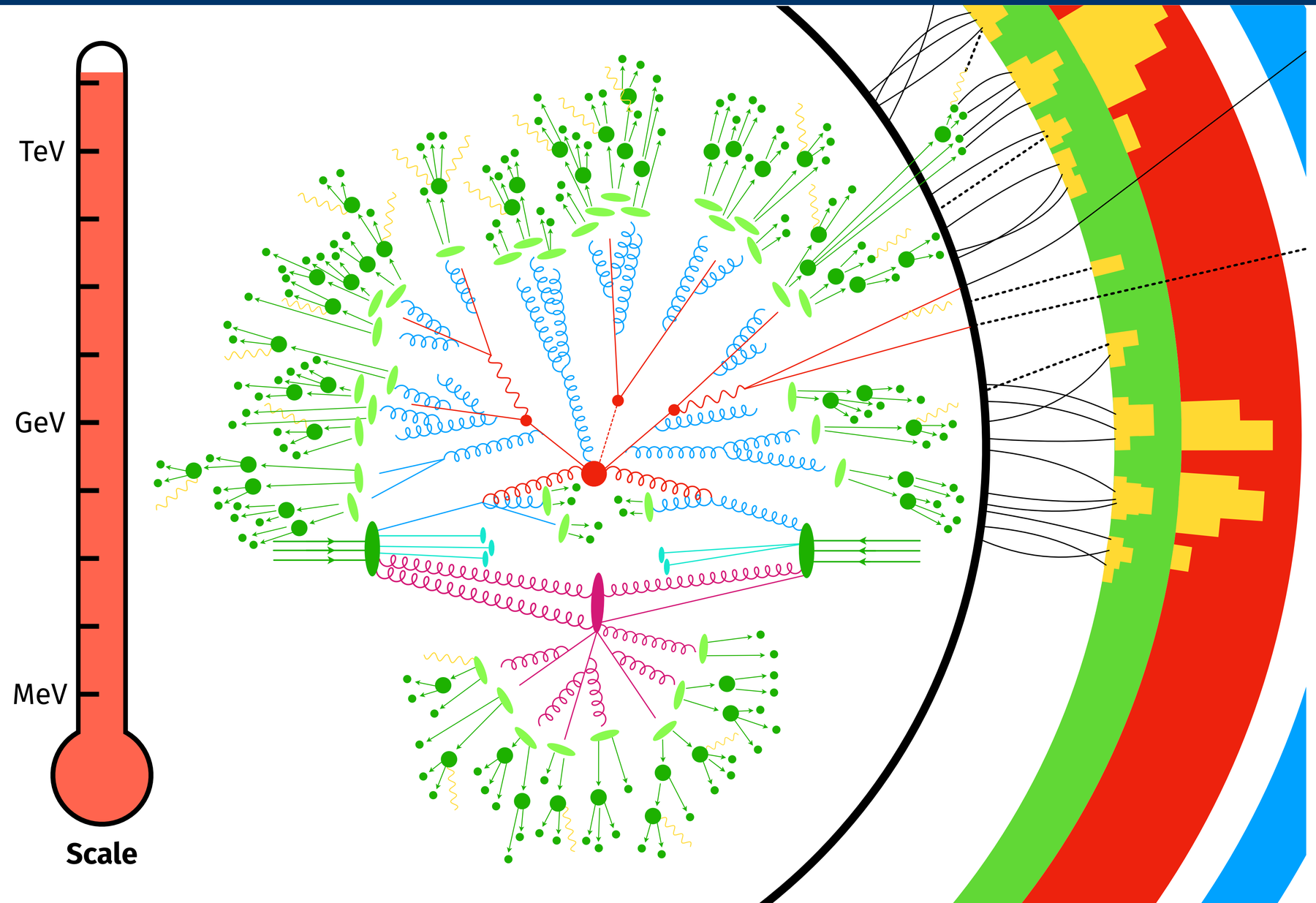
Outline

1. LHC basics
2. Matrix elements
3. Monte Carlo integration
4. Phase-space sampling
5. Event generation
6. Decays
7. Machine learning
8. Parton showers
9. Multijet merging

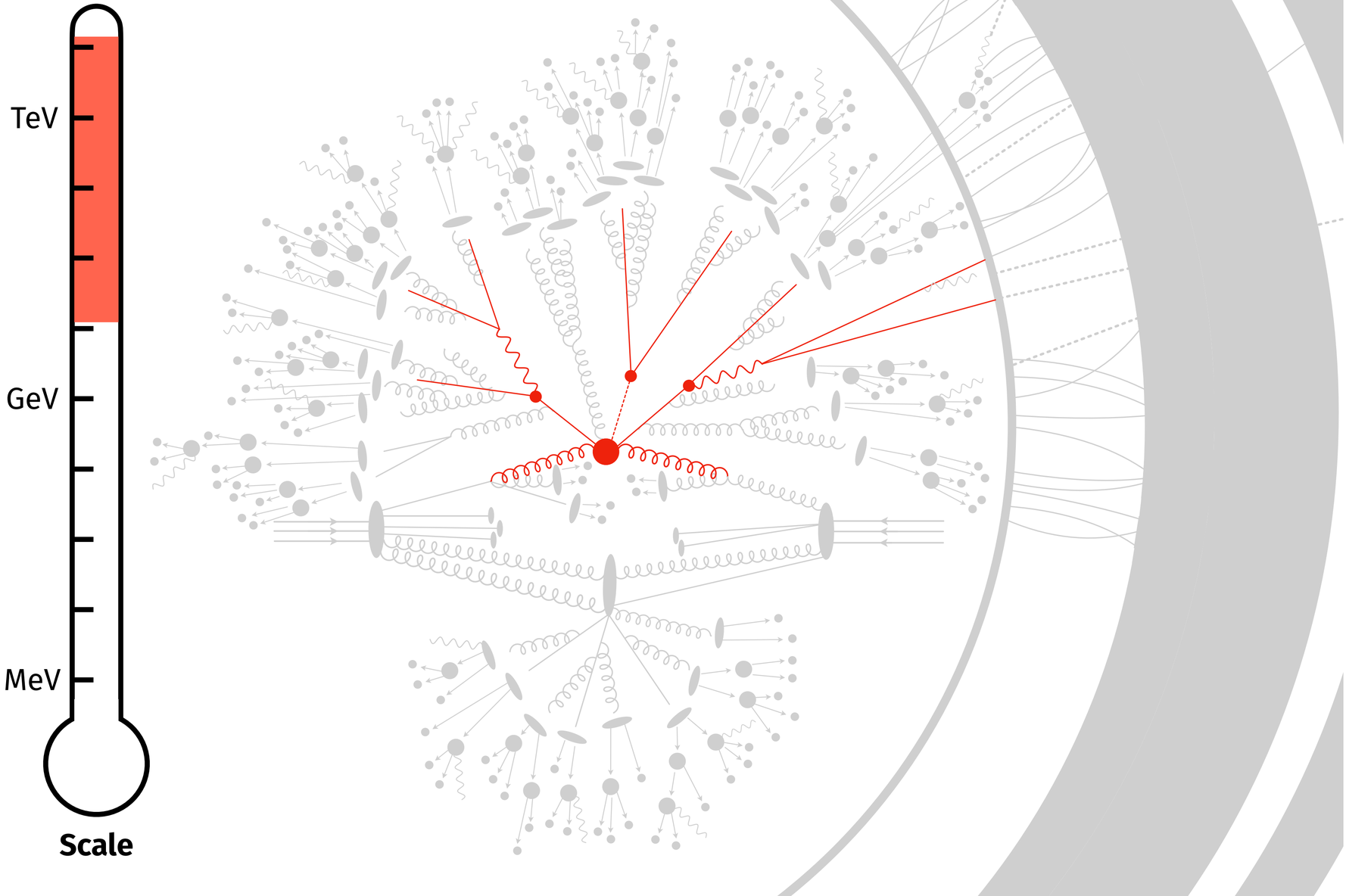
Outline

1. **LHC basics**
2. Matrix elements
3. Monte Carlo integration
4. Phase-space sampling
5. Event generation
6. Decays
7. Machine learning
8. Parton showers
9. Multijet merging

Colliding protons



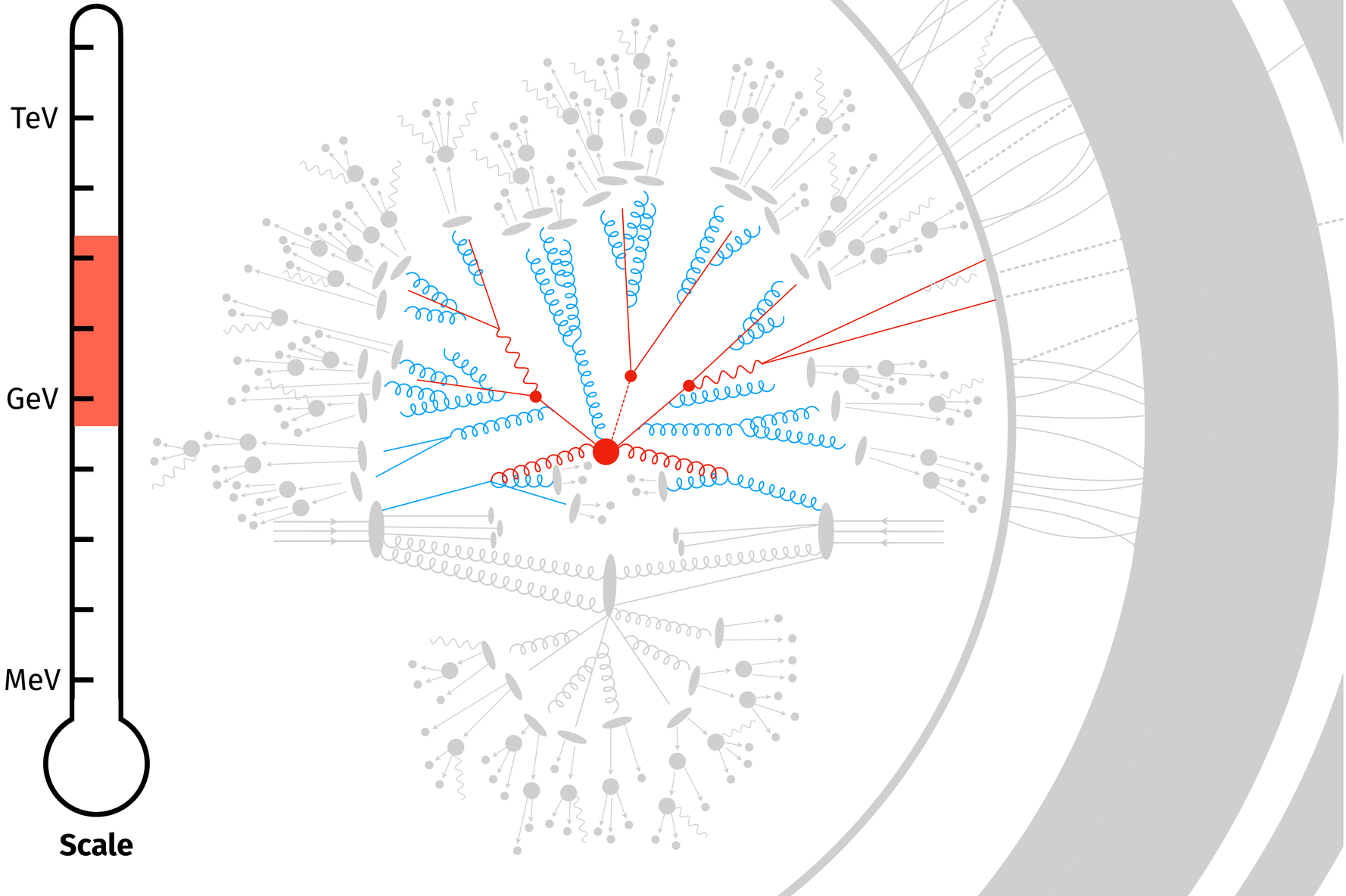
Colliding protons



Hard scattering

- process dependent
- first principles description
- BSM physics happens here

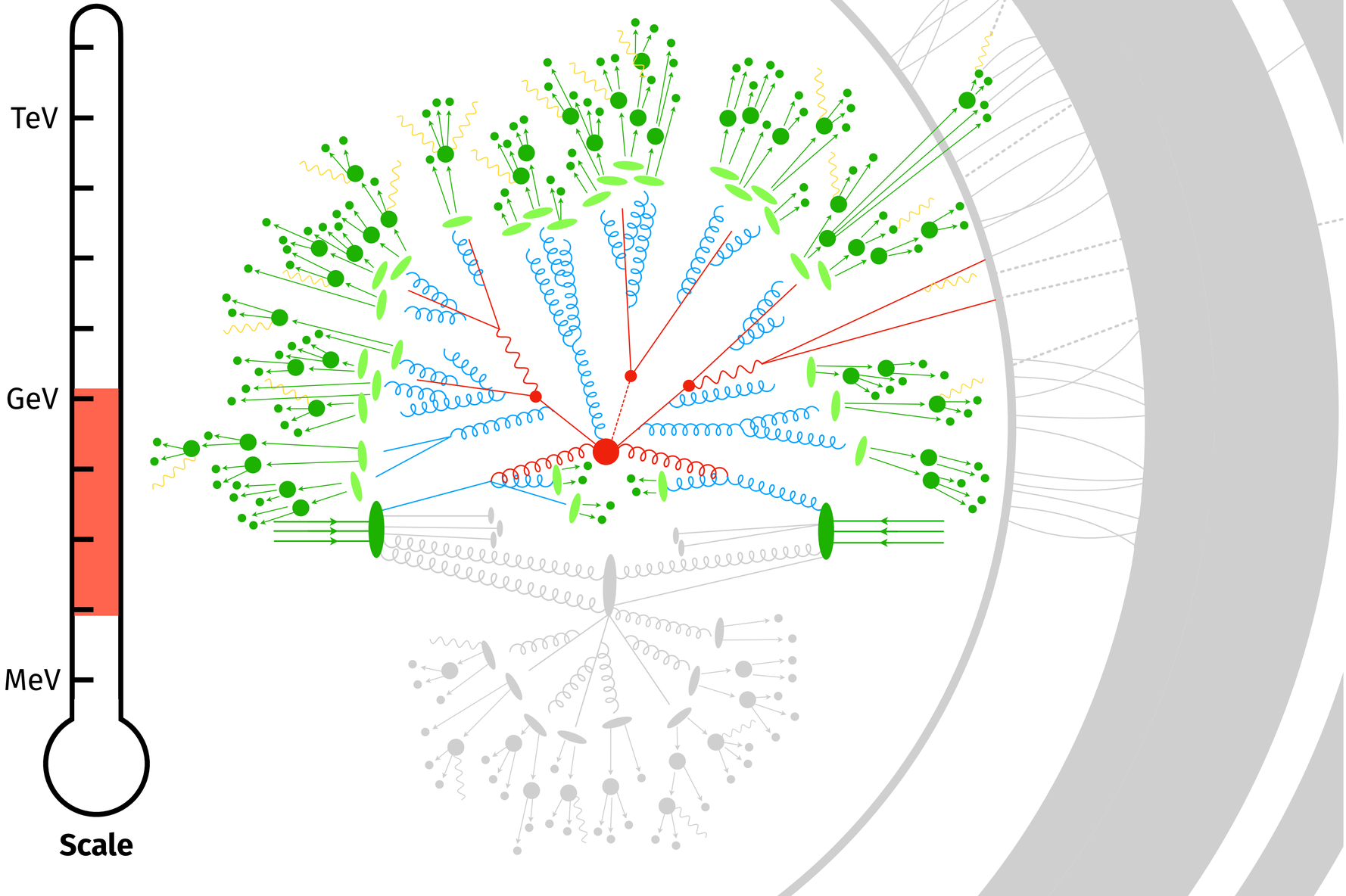
Colliding protons



Hard scattering
- process dependent
- first principles description
- BSM physics happens here

Parton shower
- QCD: “known physics”
- first principles description
- universal

Colliding protons

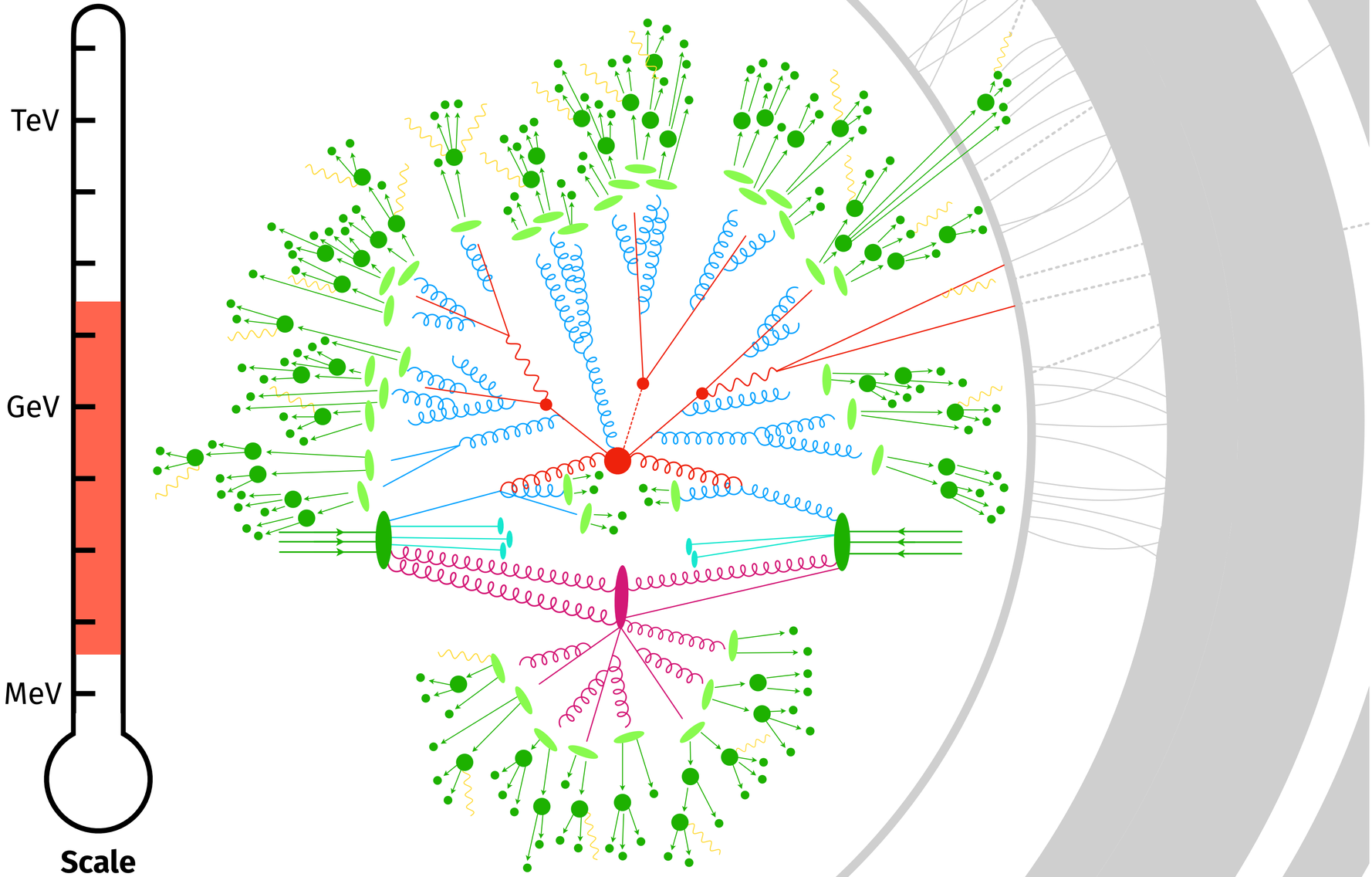


Hard scattering
- process dependent
- first principles description
- BSM physics happens here

Parton shower
- QCD: “known physics”
- first principles description
- universal

Hadronization
- low energy, universal
- phenomenological models

Colliding protons



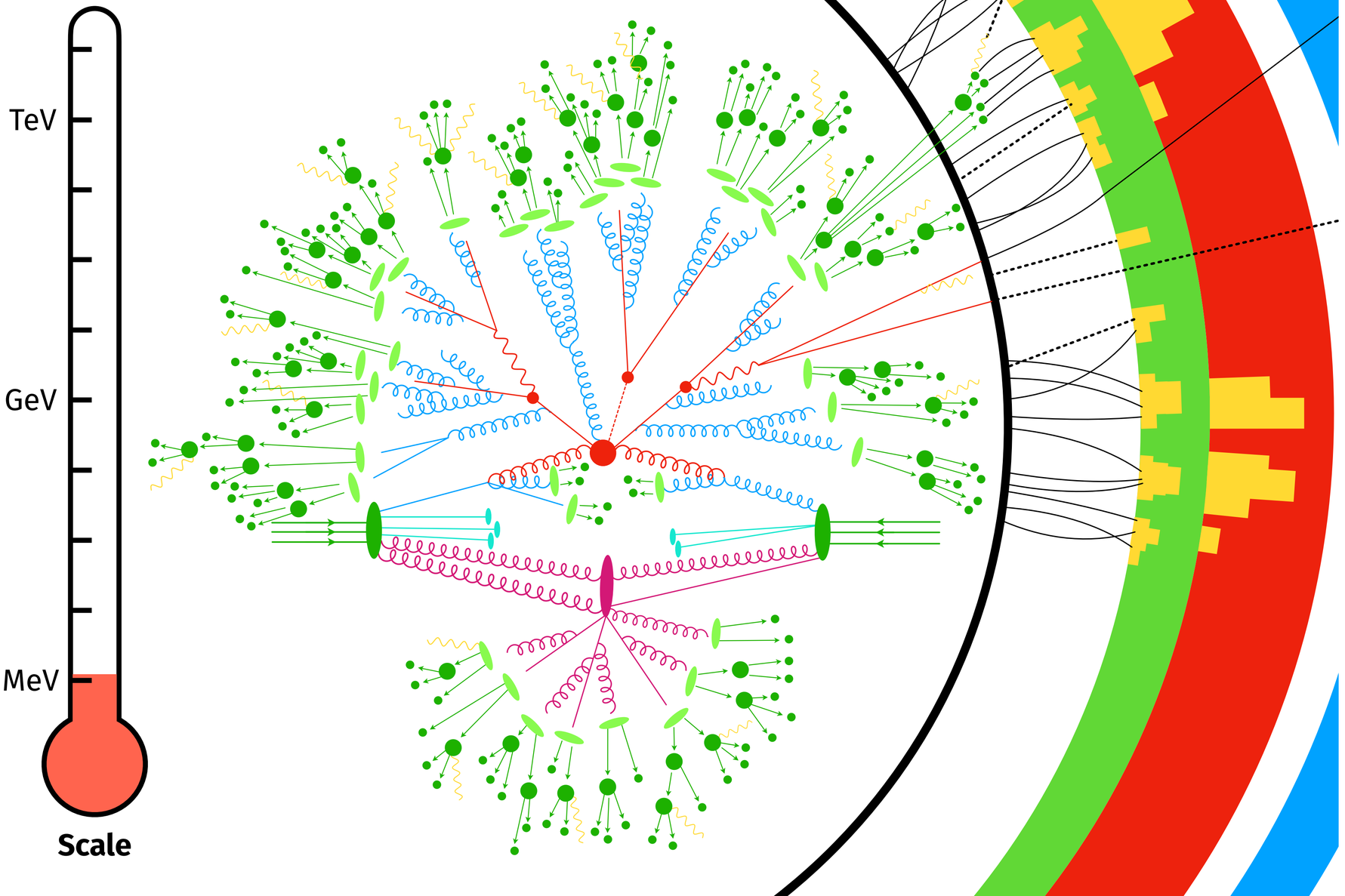
Hard scattering
- process dependent
- first principles description
- BSM physics happens here

Parton shower
- QCD: “known physics”
- first principles description
- universal

Hadronization
- low energy, universal
- phenomenological models

Underlying event
- low energy, process-dep.
- phenomenological models

Colliding protons



Hard scattering
- process dependent
- first principles description
- BSM physics happens here

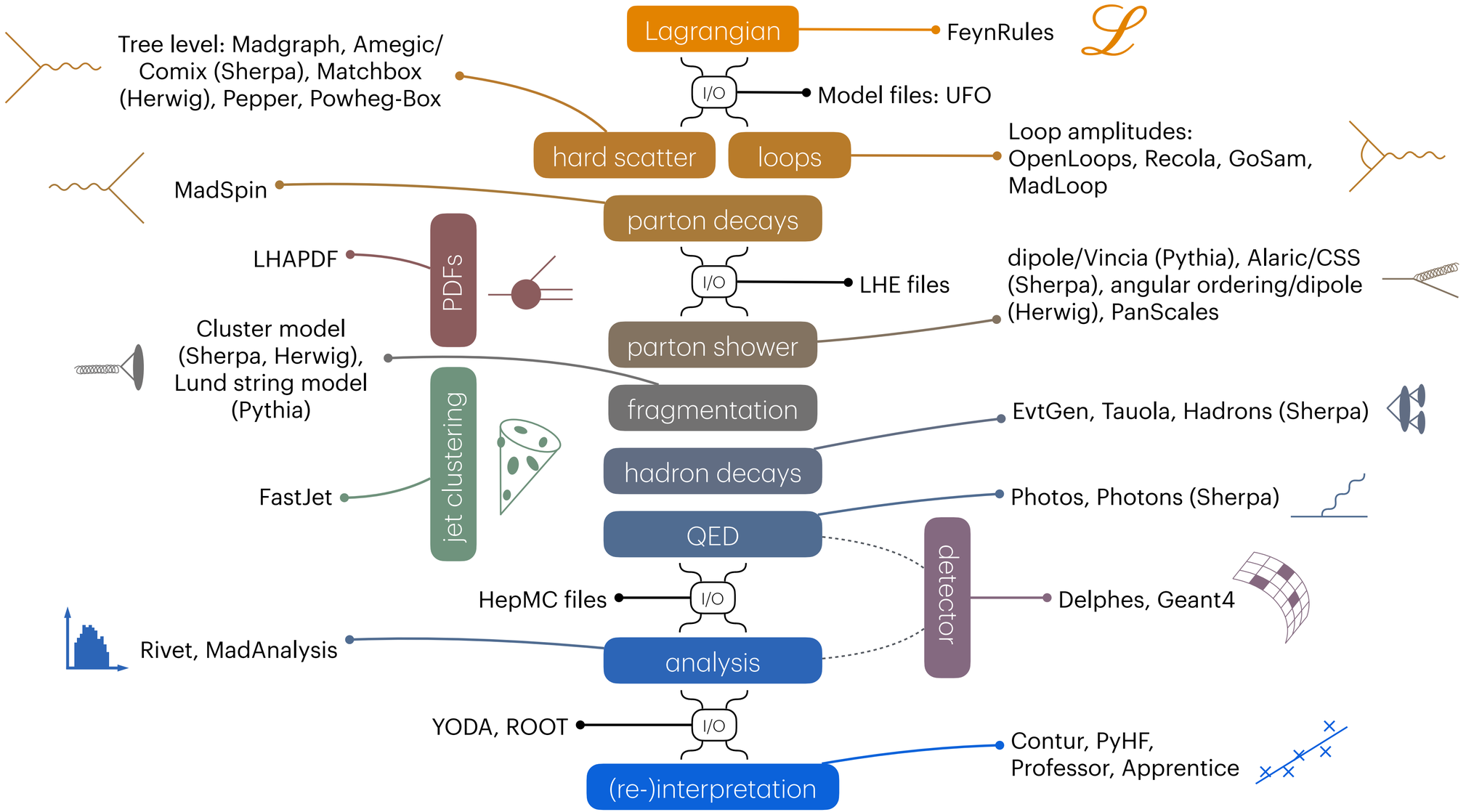
Parton shower
- QCD: “known physics”
- first principles description
- universal

Hadronization
- low energy, universal
- phenomenological models

Underlying event
- low energy, process-dep.
- phenomenological models

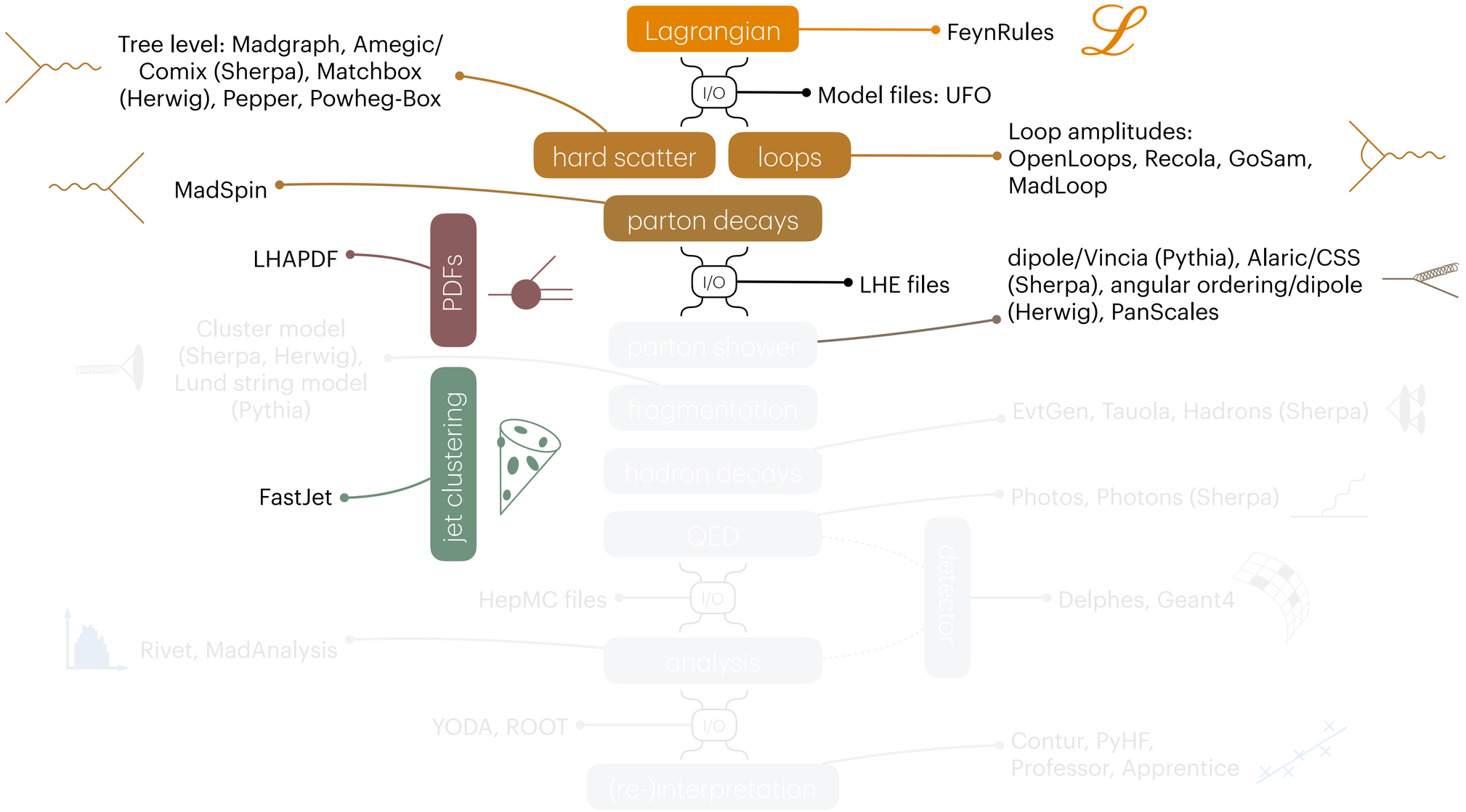
Detector effects

The Monte Carlo toolbox



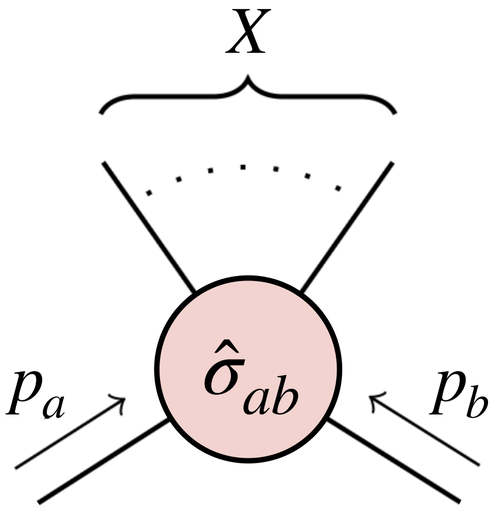
MCnet [2605.16036]

The Monte Carlo toolbox



MCnet [2605.16036]

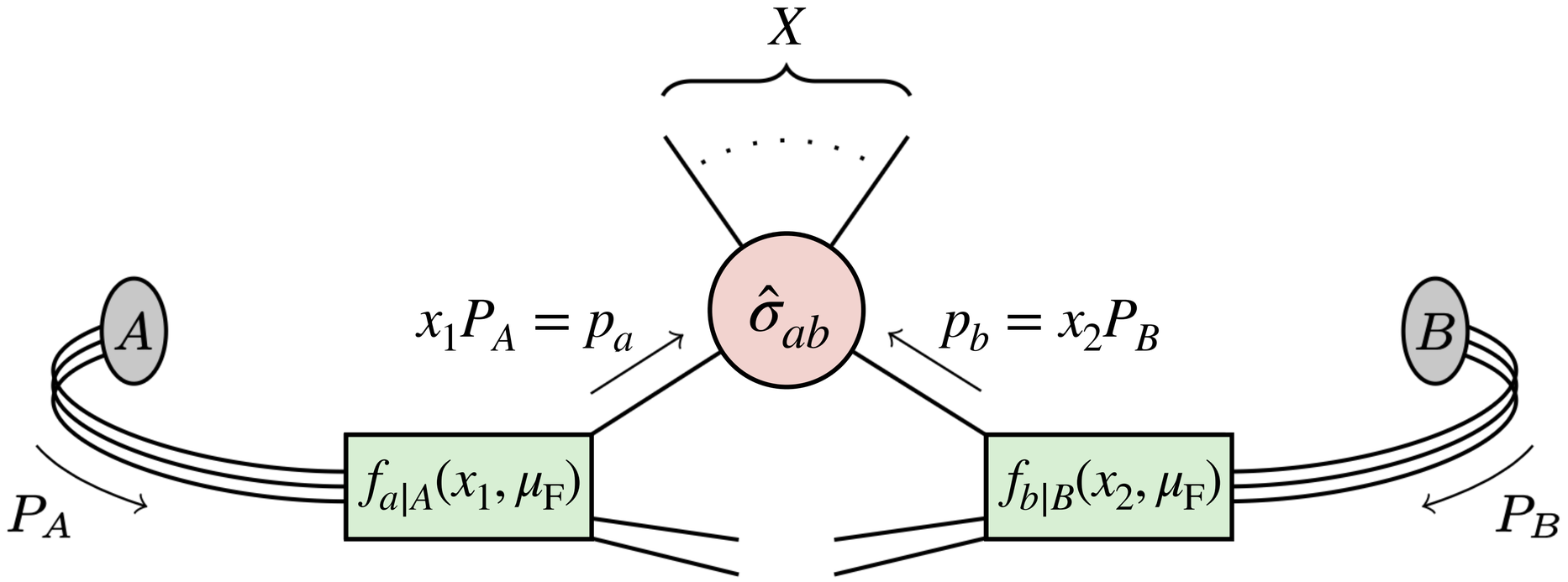
Master formula for the LHC



$$\hat{\sigma}_{ab\to X}(\mu_{\mathrm{R}}, \mu_{\mathrm{F}})$$

Hard process

Master formula for the LHC

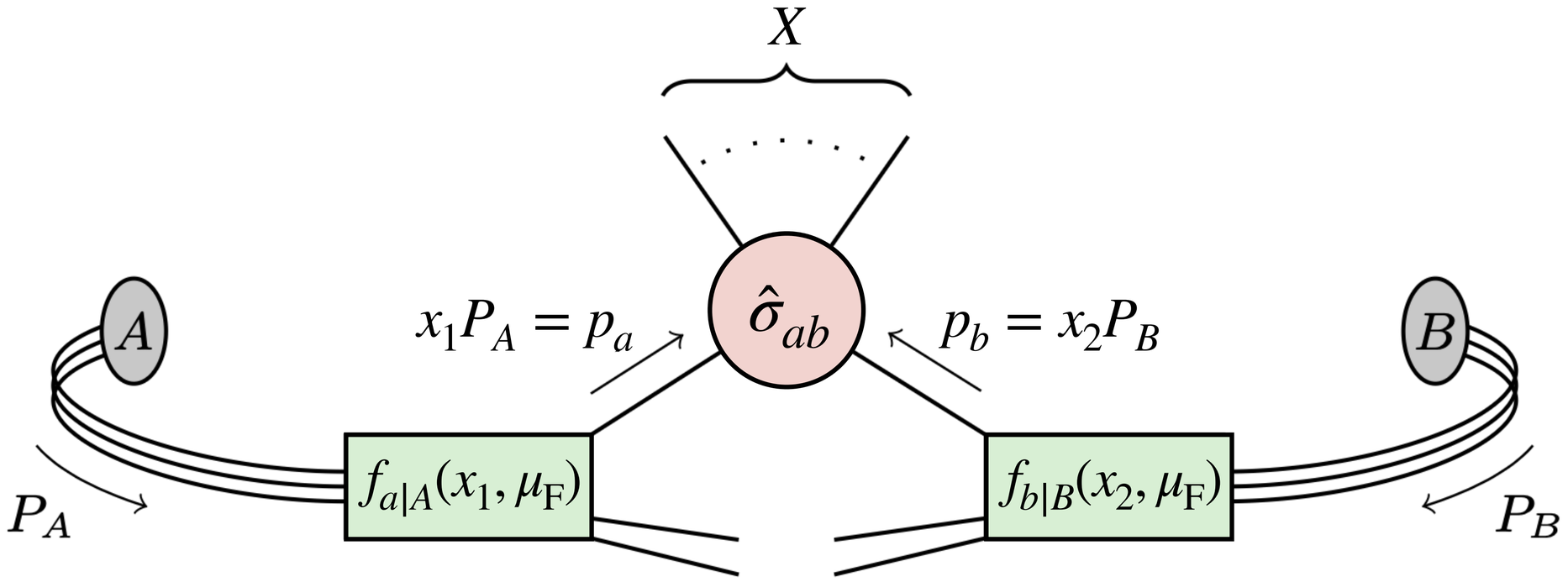


$$f_{a|A}(x_1, \mu_F)\, f_{b|B}(x_2, \mu_F)\, \hat{\sigma}_{ab\to X}(\mu_R, \mu_F)$$

Parton distribution functions

Hard process

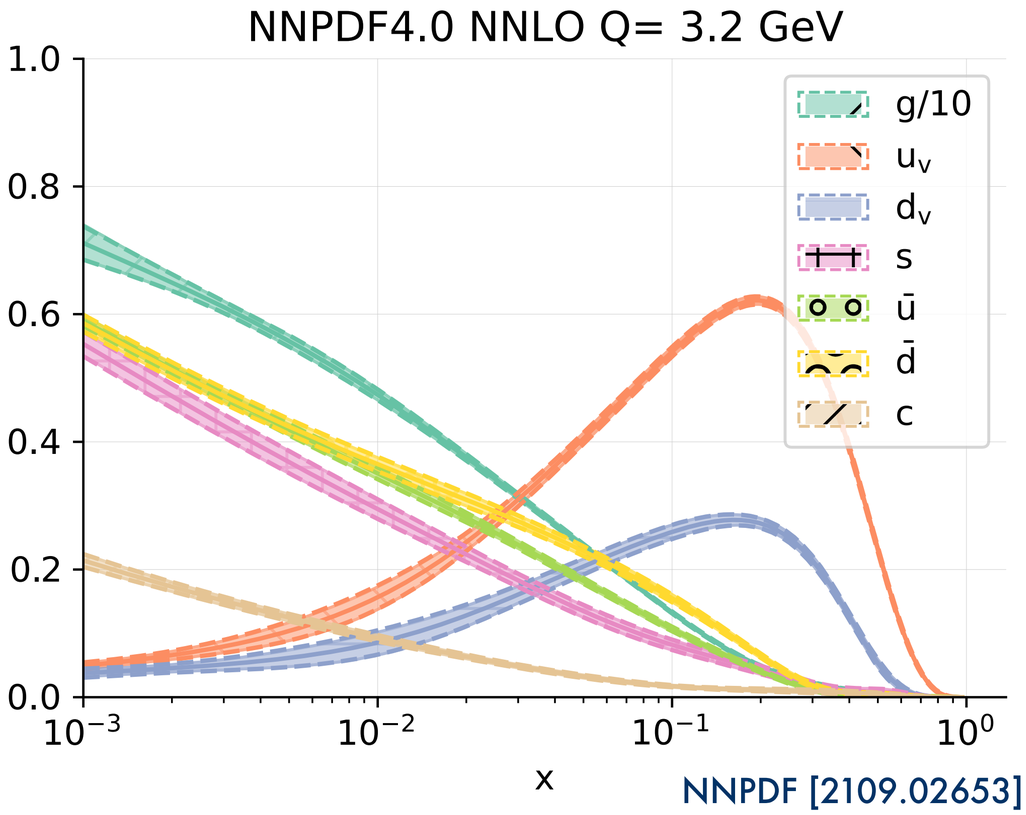
Master formula for the LHC



$$\sum_{a,b} \int \mathrm{d}x_1 \, \mathrm{d}x_2 \; f_{a|A}(x_1, \mu_\mathrm{F}) \, f_{b|B}(x_2, \mu_\mathrm{F}) \; \hat{\sigma}_{ab \to X}(\mu_\mathrm{R}, \mu_\mathrm{F})$$

Convolution — Parton distribution functions — Hard process

Parton distribution functions (PDFs)



Function of

- momentum fraction x
- factorization scale $Q^2 = \mu_{\mathrm{F}}^2$

Small x: gluons dominate

Large x: valence quarks dominate

Perturbative expansion

Parton-level cross section computed as a series in perturbation theory
→ strong coupling constant as expansion parameter

$$\hat{\sigma} = \sigma^{\text{Born}} \left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2 \sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3 \sigma^{(3)} + \dots \right)$$

LO **NLO** **NNLO** **NNNLO**

fast, simple

improved predictions,
reduces theory uncertainties

Renormalization and factorization scale

- PDFs and running coupling depend on energy scales
  → renormalization scale μ_{R}
  → factorization scale μ_{F}

- options to determine these scales in MadGraph:
  → clustering to 2→2 system
  → transverse mass
  → center-of-mass energy
  → constant (eg. $\mu_{\mathrm{F}} = M_{\mathrm{Z}}$)

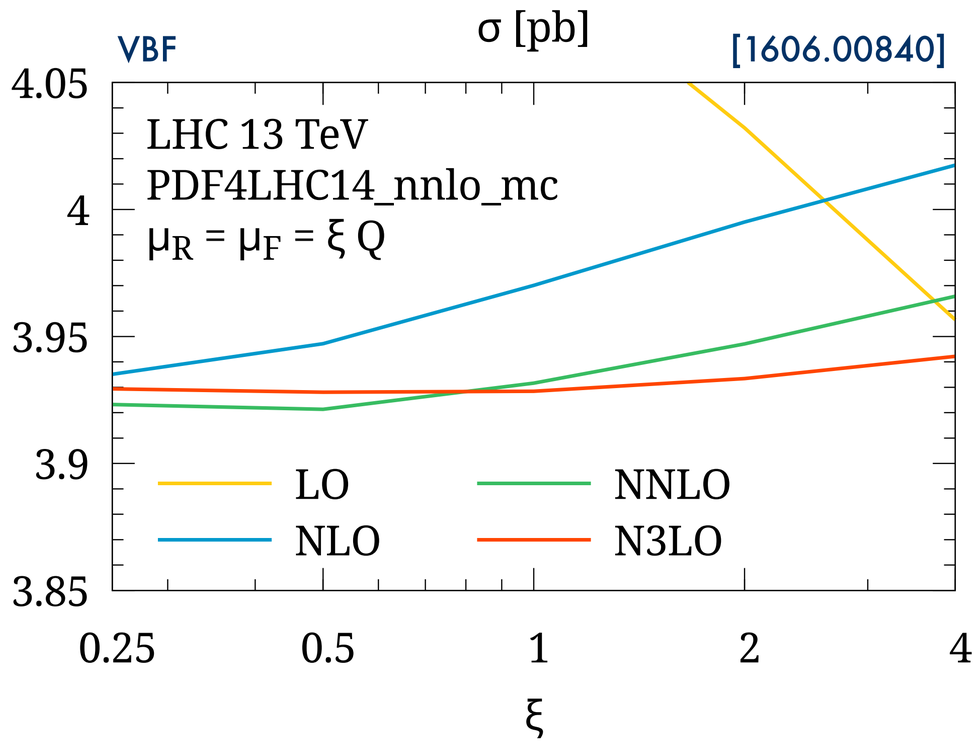
- not an exact science → large scale uncertainties

LO vs NLO

$$\sum_{a,b} \int \mathrm{d}x_1\, \mathrm{d}x_2\; f_{a|A}(x_1, \mu_\mathrm{F})\, f_{b|B}(x_2, \mu_\mathrm{F})\; \hat{\sigma}_{ab\to X}(\mu_\mathrm{R}, \mu_\mathrm{F})$$

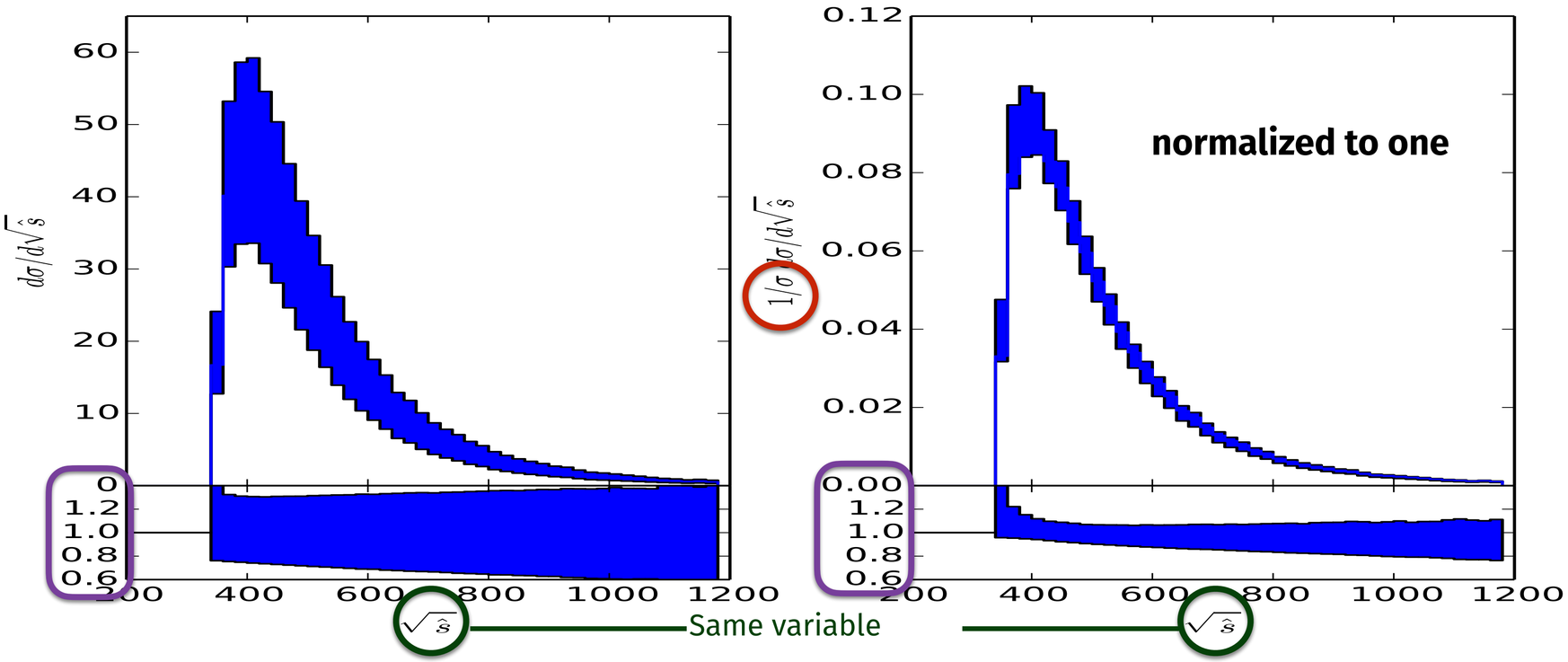
$$\hat{\sigma} = \sigma^{\mathrm{Born}} \left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2 \sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3 \sigma^{(3)} + \ldots \right)$$

- LO predictions depend strongly on the renormalization and factorization scales
- Including higher order corrections reduces the dependence on these scales → vanishes for ***all*** orders



Scale uncertainties at LO

Example LO computation (top quark pair)



Large scale uncertainty, but mostly in the normalization
→ LO is good for shape

Summary

$$\sum_{a,b} \int \mathrm{d}x_1 \, \mathrm{d}x_2 \; f_{a|A}(x_1, \mu_\mathrm{F}) \, f_{b|B}(x_2, \mu_\mathrm{F}) \; \hat{\sigma}_{ab \to X}(\mu_\mathrm{R}, \mu_\mathrm{F})$$

Convolution — Parton distribution functions — Hard process

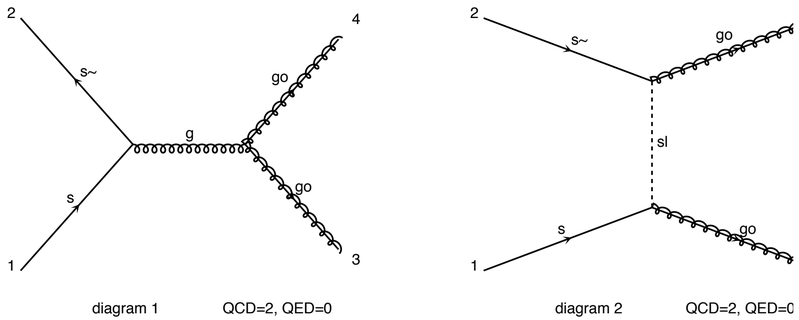
- PDF: content of the proton
  → define the physics/processes that will dominate on your accelerator
- LO: often good for shape
- NLO/NNLO: reduce scale uncertainty
- Computations inclusive (+ any jet) due to renormalization/factorization scale

Outline

1. LHC basics
2. **Matrix elements**
3. Monte Carlo integration
4. Phase-space sampling
5. Event generation
6. Decays
7. Machine learning
8. Parton showers
9. Multijet merging

From process to cross section

1. Determine production mechanism → find all diagrams



2. Evaluate matrix element → use Feynman rules

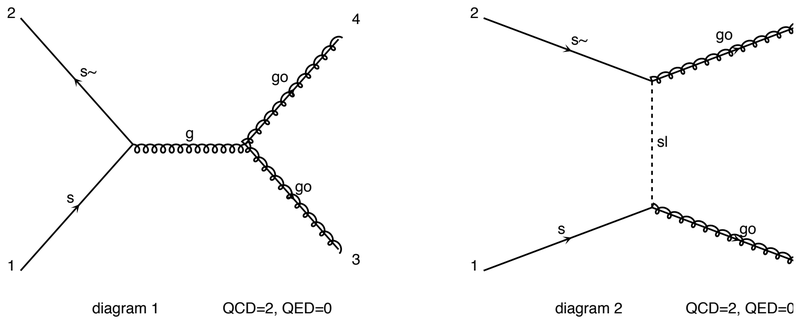
$$|\mathcal{M}|^2$$

3. Phase-space integration

$$\hat{\sigma} = \frac{1}{2\hat{s}} \int \mathrm{d}\Phi_n \, |\mathcal{M}|^2$$

From process to cross section

1. Determine production mechanism → find all diagrams



Easy enough

2. Evaluate matrix element → use Feynman rules

$$|\mathcal{M}|^2$$

Hard

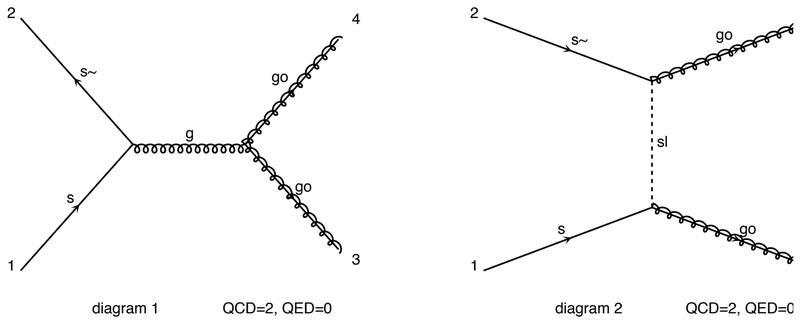
3. Phase-space integration

$$\hat{\sigma} = \frac{1}{2\hat{s}} \int \mathrm{d}\Phi_n \, |\mathcal{M}|^2$$

Very hard

From process to cross section

1. Determine production mechanism → find all diagrams



Easy enough

2. Evaluate matrix element → use Feynman rules

$$|\mathcal{M}|^2$$

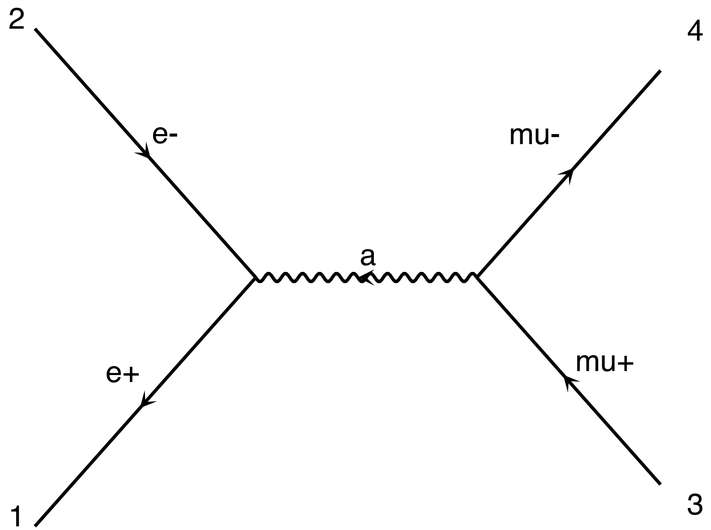
Hard

3. Phase-space integration

$$\hat{\sigma} = \frac{1}{2\hat{s}} \int \mathrm{d}\Phi_n \, |\mathcal{M}|^2$$

Very hard

Text book recipe



1. Write down amplitude

$$\mathcal{M} = e^2 \left[\bar{v}(p_2)\gamma^\mu u(p_1)\right] \frac{g_{\mu\nu}}{q^2} \left[\bar{u}(p_3)\gamma^\nu v(p_4)\right]$$

2. Compute squared matrix element

$$\frac{1}{4}\sum_{\text{hel}} |\mathcal{M}|^2 = \frac{1}{4}\sum_{\text{hel}} \mathcal{M}^*\mathcal{M}$$

3. Use sum rules/trace identities/... to get function of momenta

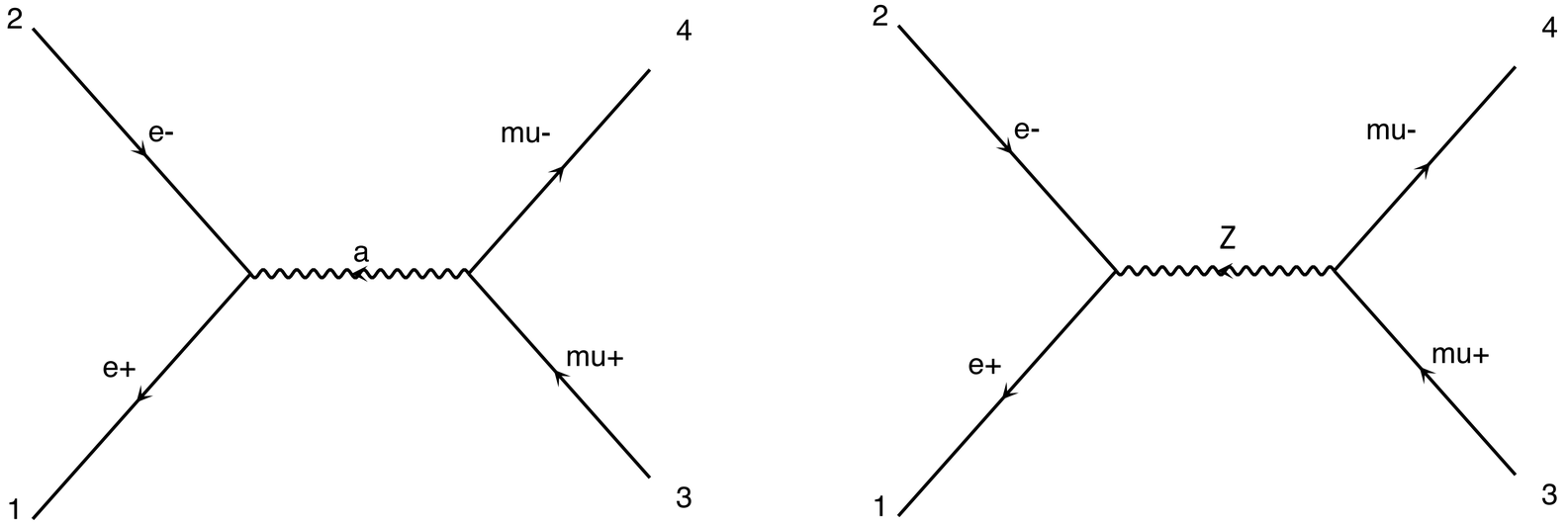
$$\sum_{\text{hel}} u\bar{u} = \not{p} + m$$

$$\frac{e^4}{4q^4}\text{Tr}[\not{p}_1\gamma^\mu \not{p}_2\gamma^\nu]\text{Tr}[\not{p}_3\gamma_\mu \not{p}_4\gamma_\nu]$$

$$\frac{8e^4}{q^4}\left[(p_1\cdot p_3)(p_2\cdot p_4) + (p_1\cdot p_4)(p_2\cdot p_3)\right]$$

Very simple final expression!
→ fast to evaluate

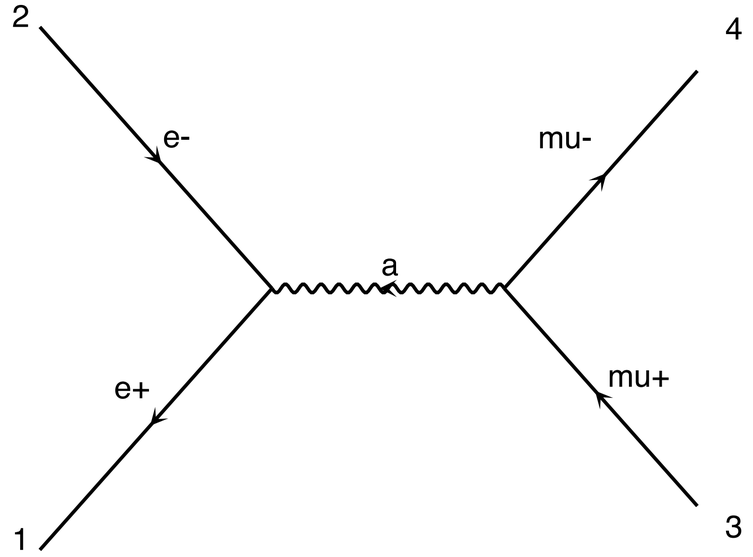
Matrix elements and interferences



- Need to compute $|\mathcal{M}_\gamma|^2, |\mathcal{M}_Z|^2, \text{and } 2\text{Re}(\mathcal{M}_\gamma^* \mathcal{M}_Z)$
- For M Feynman diagrams: need to compute M^2 different term
- The number of diagrams ***scales factorially*** with the number of particles
- In practice possible up to 2→4

Helicity amplitudes

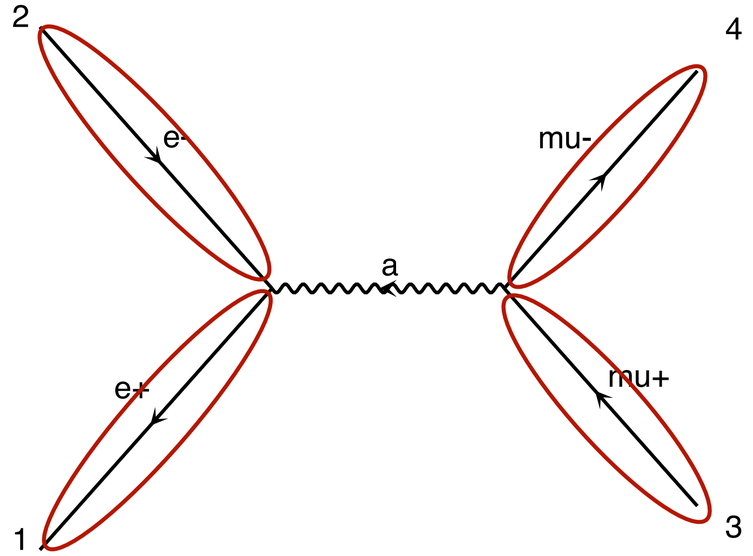
- Idea: ***evaluate first, square later***
- Compute $\overline{|\mathcal{M}|^2} = \frac{1}{N_{\text{hel}}^{\text{in}}} \sum_{\lambda_1,\ldots,\lambda_n} |\mathcal{M}(\lambda_1,\ldots,\lambda_n)|^2$
- Evaluate ***single helicity amplitude*** $\mathcal{M}(\lambda_1,\ldots,\lambda_n)$
- Example: $\mathcal{M}_{\lambda_1,\lambda_2,\lambda_3,\lambda_4} = \left[\left(\bar{v}_{\lambda_1}\, e\gamma^\mu\, u_{\lambda_2}\, \frac{g_{\mu\nu}}{q^2}\right) \bar{u}_{\lambda_3}\, e\gamma^\nu\, v_{\lambda_4}\right]$



HELAS 92' [KEK], Dittmaier 98' [hep-ph/9805445]

Helicity amplitudes

- Idea: ***evaluate first, square later***
- Compute $\overline{|\mathcal{M}|^2} = \frac{1}{N_{\text{hel}}^{\text{in}}} \sum_{\lambda_1,\ldots,\lambda_n} |\mathcal{M}(\lambda_1,\ldots,\lambda_n)|^2$
- Evaluate ***single helicity amplitude*** $\mathcal{M}(\lambda_1,\ldots,\lambda_n)$
- Example: $\mathcal{M}_{\lambda_1,\lambda_2,\lambda_3,\lambda_4} = \left[\left(\bar{v}_{\lambda_1}\, e\gamma^\mu\, u_{\lambda_2} \frac{g_{\mu\nu}}{q^2} \right) \bar{u}_{\lambda_3}\, e\gamma^\nu\, v_{\lambda_4} \right]$



Pseudocode:

$$\bar{v}_1 = \text{fct}(\bar{p}_1, m_1, \lambda_1)$$
$$u_2 = \text{fct}(\bar{p}_2, m_2, \lambda_2)$$
$$\bar{u}_3 = \text{fct}(\bar{p}_3, m_3, \lambda_3)$$
$$v_4 = \text{fct}(\bar{p}_4, m_4, \lambda_4)$$

Construct external wave functions

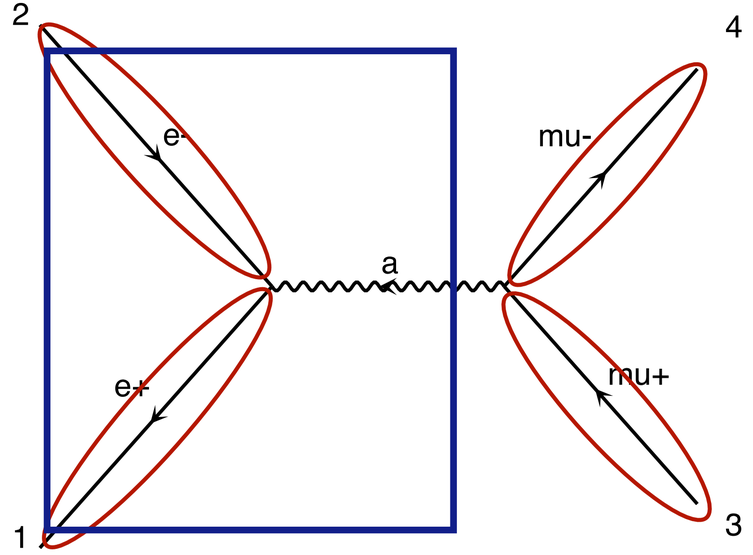
$$u_\lambda(p) = \begin{pmatrix} \omega_{-\lambda}(p)\,\chi_\lambda(\vec{p}) \\ \omega_\lambda(p)\,\chi_\lambda(\vec{p}) \end{pmatrix} \qquad \chi_+(\vec{p}) = \frac{1}{\sqrt{2|\vec{p}|(|\vec{p}|+p_z)}} \begin{pmatrix} |\vec{p}|+p_z \\ p_x + ip_y \end{pmatrix}$$

$$\omega_\pm(p) = \sqrt{E \pm |\vec{p}|} \qquad \chi_-(\vec{p}) = \frac{1}{\sqrt{2|\vec{p}|(|\vec{p}|+p_z)}} \begin{pmatrix} -p_x + ip_y \\ |\vec{p}|+p_z \end{pmatrix}$$

HELAS 92' [KEK], Dittmaier 98' [hep-ph/9805445]

Helicity amplitudes

- Idea: ***evaluate first, square later***
- Compute $\overline{|\mathcal{M}|^2} = \frac{1}{N_{\text{hel}}^{\text{in}}} \sum_{\lambda_1,\ldots,\lambda_n} |\mathcal{M}(\lambda_1,\ldots,\lambda_n)|^2$
- Evaluate ***single helicity amplitude*** $\mathcal{M}(\lambda_1,\ldots,\lambda_n)$
- Example: $\mathcal{M}_{\lambda_1,\lambda_2,\lambda_3,\lambda_4} = \left[\left(\bar{v}_{\lambda_1}\, e\gamma^\mu\, u_{\lambda_2}\frac{g_{\mu\nu}}{q^2}\right)\bar{u}_{\lambda_3}\, e\gamma^\nu\, v_{\lambda_4}\right]$



Pseudocode:

$$\bar{v}_1 = \text{fct}(\bar{p}_1, m_1, \lambda_1)$$
$$u_2 = \text{fct}(\bar{p}_2, m_2, \lambda_2)$$
$$\bar{u}_3 = \text{fct}(\bar{p}_3, m_3, \lambda_3)$$
$$v_4 = \text{fct}(\bar{p}_4, m_4, \lambda_4)$$

$$V_a^\mu = \text{fct}(\bar{v}_1, u_2, m_a, \Gamma_a) = e\,\bar{v}_1\gamma^\nu u_2 \frac{g_\nu{}^\mu}{q^2 - m_a^2 + i m_a \Gamma_a}$$

Construct external wave functions

$$u_\lambda(p) = \begin{pmatrix} \omega_{-\lambda}(p)\,\chi_\lambda(\vec{p}) \\ \omega_\lambda(p)\,\chi_\lambda(\vec{p}) \end{pmatrix} \qquad \chi_+(\vec{p}) = \frac{1}{\sqrt{2|\vec{p}|(|\vec{p}| + p_z)}} \begin{pmatrix} |\vec{p}| + p_z \\ p_x + i p_y \end{pmatrix}$$

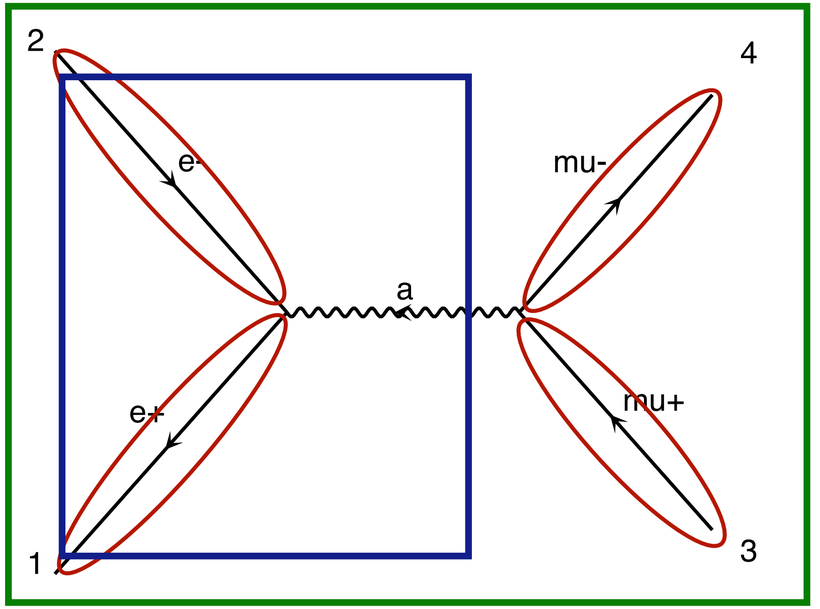
$$\omega_\pm(p) = \sqrt{E \pm |\vec{p}|} \qquad \chi_-(\vec{p}) = \frac{1}{\sqrt{2|\vec{p}|(|\vec{p}| + p_z)}} \begin{pmatrix} -p_x + i p_y \\ |\vec{p}| + p_z \end{pmatrix}$$

Calculate internal propagator wave functions

HELAS 92' [KEK], Dittmaier 98' [hep-ph/9805445]

Helicity amplitudes

- Idea: ***evaluate first, square later***
- Compute $\overline{|\mathcal{M}|^2} = \frac{1}{N_{\text{hel}}^{\text{in}}} \sum_{\lambda_1,\ldots,\lambda_n} |\mathcal{M}(\lambda_1,\ldots,\lambda_n)|^2$
- Evaluate ***single helicity amplitude*** $\mathcal{M}(\lambda_1,\ldots,\lambda_n)$
- Example: $\mathcal{M}_{\lambda_1,\lambda_2,\lambda_3,\lambda_4} = \left[\left(\bar{v}_{\lambda_1}\, e\gamma^\mu\, u_{\lambda_2} \frac{g_{\mu\nu}}{q^2} \right) \bar{u}_{\lambda_3}\, e\gamma^\nu\, v_{\lambda_4} \right]$



Pseudocode:

$$\bar{v}_1 = \text{fct}(\bar{p}_1, m_1, \lambda_1)$$
$$u_2 = \text{fct}(\bar{p}_2, m_2, \lambda_2)$$
$$\bar{u}_3 = \text{fct}(\bar{p}_3, m_3, \lambda_3)$$
$$v_4 = \text{fct}(\bar{p}_4, m_4, \lambda_4)$$

$$V_a^\mu = \text{fct}(\bar{v}_1, u_2, m_a, \Gamma_a) = e\, \bar{v}_1 \gamma^\nu u_2 \frac{g_\nu{}^\mu}{q^2 - m_a^2 + i m_a \Gamma_a}$$

$$\mathcal{M} = \text{fct}(\bar{u}_3, v_4, V_a) = e\, \bar{u}_3 \gamma_\mu v_4\, V_a^\mu$$

Construct external wave functions

$$u_\lambda(p) = \begin{pmatrix} \omega_{-\lambda}(p)\, \chi_\lambda(\vec{p}) \\ \omega_\lambda(p)\, \chi_\lambda(\vec{p}) \end{pmatrix} \qquad \chi_+(\vec{p}) = \frac{1}{\sqrt{2|\vec{p}|(|\vec{p}| + p_z)}} \begin{pmatrix} |\vec{p}| + p_z \\ p_x + i p_y \end{pmatrix}$$

$$\omega_\pm(p) = \sqrt{E \pm |\vec{p}|} \qquad \chi_-(\vec{p}) = \frac{1}{\sqrt{2|\vec{p}|(|\vec{p}| + p_z)}} \begin{pmatrix} -p_x + i p_y \\ |\vec{p}| + p_z \end{pmatrix}$$

Calculate internal propagator wave functions

Contact wave functions to obtain the amplitude

HELAS 92' [KEK], Dittmaier 98' [hep-ph/9805445]

Helicity amplitudes

- Idea: ***evaluate first, square later***
- Compute $\overline{|\mathcal{M}|^2} = \frac{1}{N_{\text{hel}}^{\text{in}}} \sum_{\lambda_1,\ldots,\lambda_n} |\mathcal{M}(\lambda_1,\ldots,\lambda_n)|^2$
- Evaluate ***single helicity amplitude*** $\mathcal{M}(\lambda_1,\ldots$
- Example: $\mathcal{M}_{\lambda_1,\lambda_2,\lambda_3,\lambda_4} = \left[\left(\bar{v}_{\lambda_1}\, e\gamma^\mu\, u_{\lambda_2}\, \frac{g_{\mu\nu}}{q^2}\right) \bar{u}_{\lambda_3}\, e\gamma^\nu$

Pseudocode:

$$\bar{v}_1 = \text{fct}(\bar{p}_1, m_1, \lambda_1)$$
$$u_2 = \text{fct}(\bar{p}_2, m_2, \lambda_2)$$
$$\bar{u}_3 = \text{fct}(\bar{p}_3, m_3, \lambda_3)$$
$$v_4 = \text{fct}(\bar{p}_4, m_4, \lambda_4)$$

$$V_a^\mu = \text{fct}(\bar{v}_1, u_2, m_a, \Gamma_a) = e\,\bar{v}_1\gamma^\nu u_2\, \frac{g_\nu{}^\mu}{q^2 - m_a^2 + i m_a \Gamma_a}$$

$$\mathcal{M} = \text{fct}(\bar{u}_3, v_4, V_a) = e\,\bar{u}_3\gamma_\mu v_4\, V_a^\mu$$

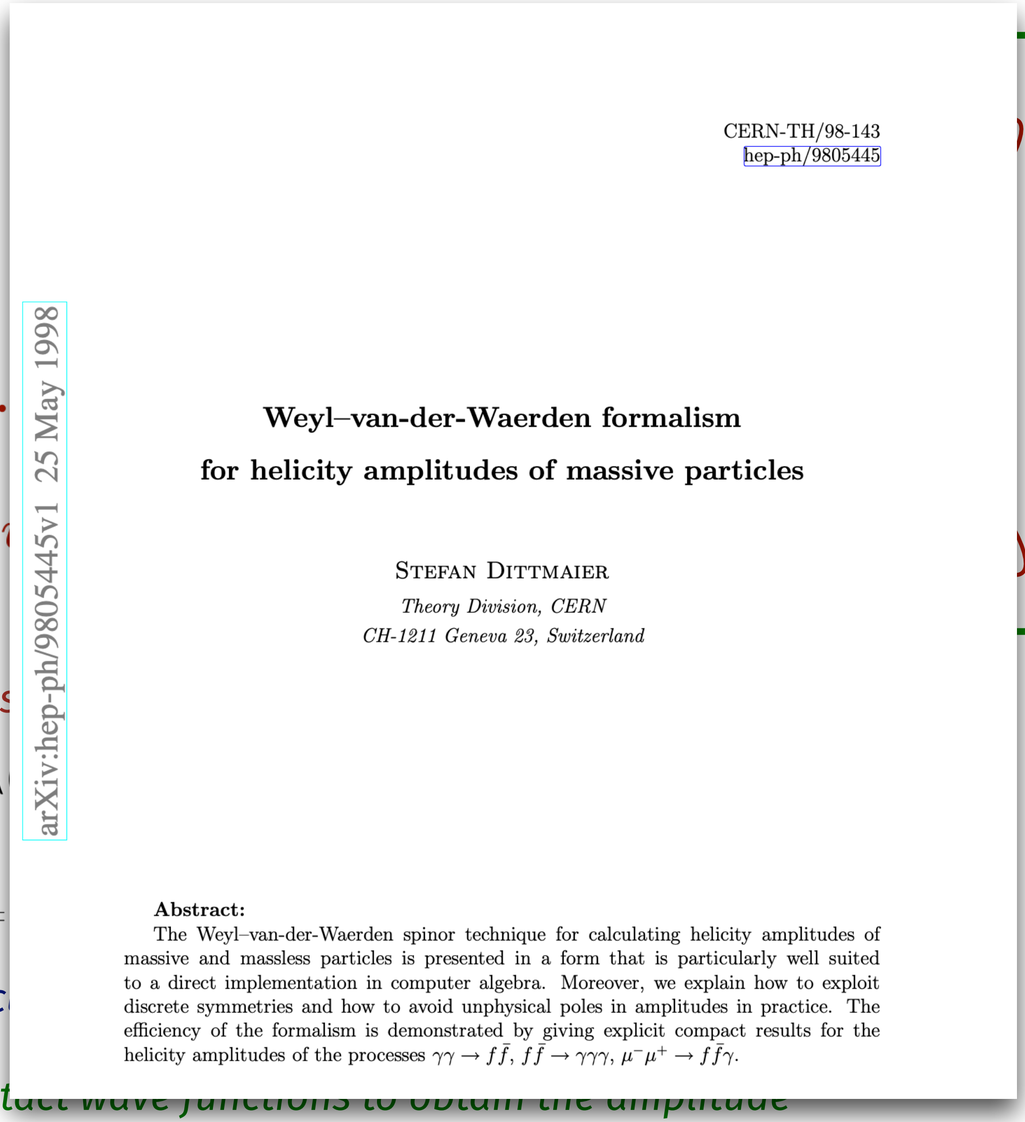
Cons

u_λ

$\omega_\pm$

Calc

Cont



CERN-TH/98-143
hep-ph/9805445

arXiv:hep-ph/9805445v1 25 May 1998

Weyl–van-der-Waerden formalism

for helicity amplitudes of massive particles

STEFAN DITTMAIER
Theory Division, CERN
CH-1211 Geneva 23, Switzerland

Abstract:
The Weyl–van-der-Waerden spinor technique for calculating helicity amplitudes of massive and massless particles is presented in a form that is particularly well suited to a direct implementation in computer algebra. Moreover, we explain how to exploit discrete symmetries and how to avoid unphysical poles in amplitudes in practice. The efficiency of the formalism is demonstrated by giving explicit compact results for the helicity amplitudes of the processes $\gamma\gamma \to f\bar{f}$, $f\bar{f} \to \gamma\gamma\gamma$, $\mu^-\mu^+ \to f\bar{f}\gamma$.

4

3

HELAS 92' [KEK], Dittmaier 98' [hep-ph/9805445]

MadGraph implementation

You can check how MadGraph builds the matrix element
in the file `SubProcesses/<processname>/matrix1_orig.f`

Incoming/outgoing particles

Vertex & photon propagator

Vertex & first amplitude

Vertex & Z propagator

Vertex & second amplitude

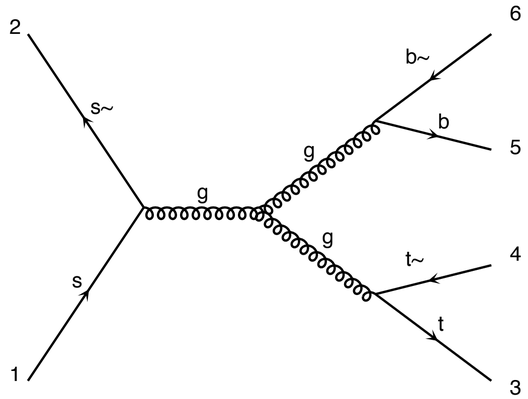
```
      CALL OXXXXX(P(0,1),ZERO,NHEL(1),-1*IC(1),W(1,1))
      CALL IXXXXX(P(0,2),ZERO,NHEL(2),+1*IC(2),W(1,2))
      CALL IXXXXX(P(0,3),ZERO,NHEL(3),-1*IC(3),W(1,3))
      CALL OXXXXX(P(0,4),ZERO,NHEL(4),+1*IC(4),W(1,4))
      CALL FFV1P0_3(W(1,2),W(1,1),GC_3,ZERO, FK_ZERO,W(1,5))
C     Amplitude(s) for diagram number 1
      CALL FFV1_0(W(1,3),W(1,4),W(1,5),GC_3,AMP(1))
      CALL FFV2_4_3(W(1,2),W(1,1),GC_50,GC_59,MDL_MZ, FK_MDL_WZ,W(1,5))
C     Amplitude(s) for diagram number 2
      CALL FFV2_4_0(W(1,3),W(1,4),W(1,5),GC_50,GC_59,AMP(2))
```

Comparison

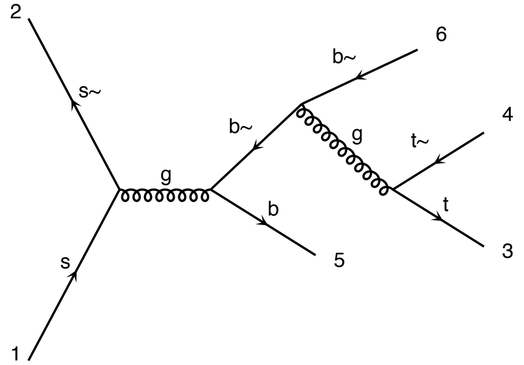
	M diagrams	N particles
Analytical	M^2	$(N!)^2$
Helicity	M	$(N!)\,2^N$

Share work between diagrams

Known



$\mathcal{M}_1$



$\mathcal{M}_2$

Number of routines: 0

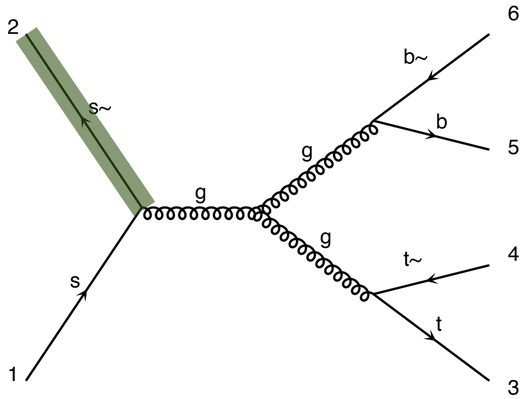
Number of routines: 0

Number of routines for both: 0

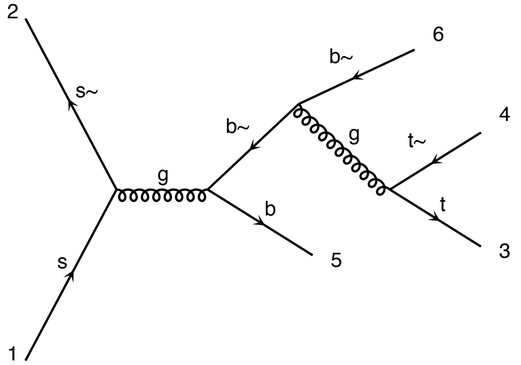
$$\mathcal{M} = \mathcal{M}_1 + \mathcal{M}_2$$

Share work between diagrams

Known



$\mathcal{M}_1$



$\mathcal{M}_2$

Number of routines: 1

Number of routines: 0

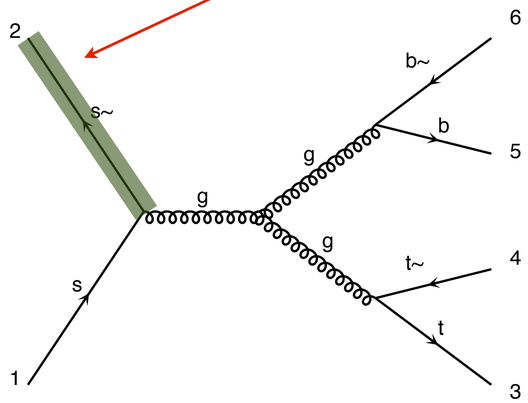
Number of routines for both: 1

$$\mathcal{M} = \mathcal{M}_1 + \mathcal{M}_2$$

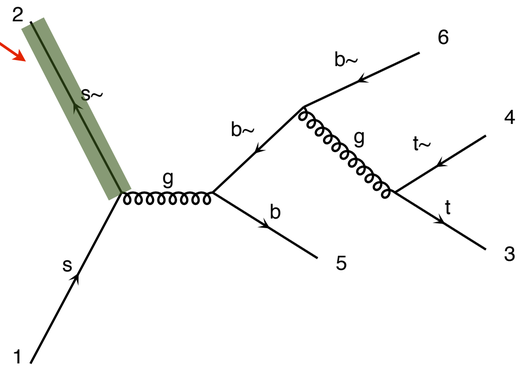
Share work between diagrams

Identical

Known



$\mathcal{M}_1$



$\mathcal{M}_2$

Number of routines: 1

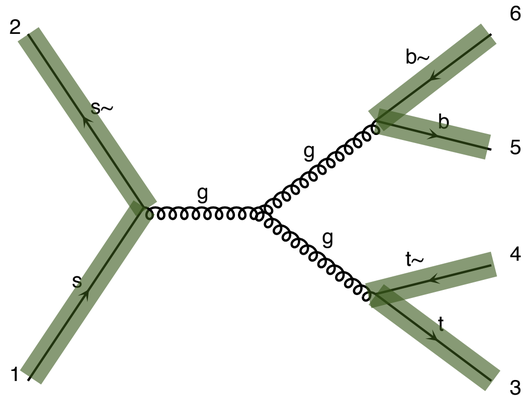
Number of routines: 1

Number of routines for both: 1

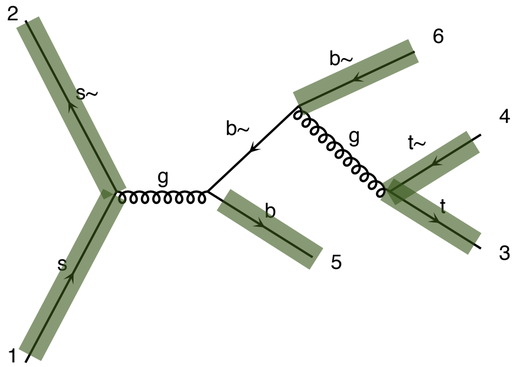
$$\mathcal{M} = \mathcal{M}_1 + \mathcal{M}_2$$

Share work between diagrams

Known



$\mathcal{M}_1$



$\mathcal{M}_2$

Number of routines: 6

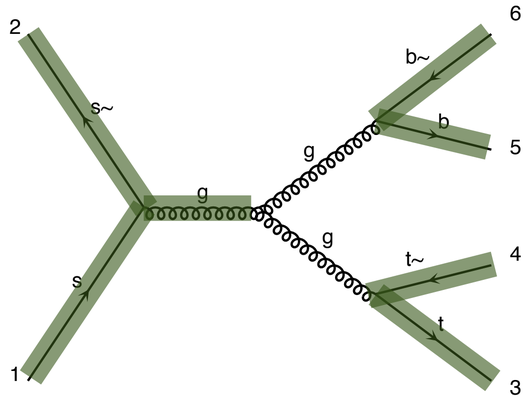
Number of routines: 6

Number of routines for both: 6

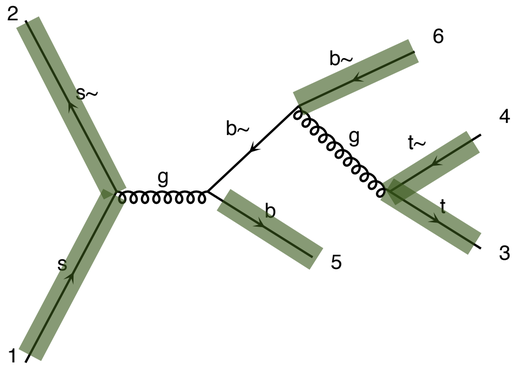
$$\mathcal{M} = \mathcal{M}_1 + \mathcal{M}_2$$

Share work between diagrams

Known



$\mathcal{M}_1$



$\mathcal{M}_2$

Number of routines: 7

Number of routines: 6

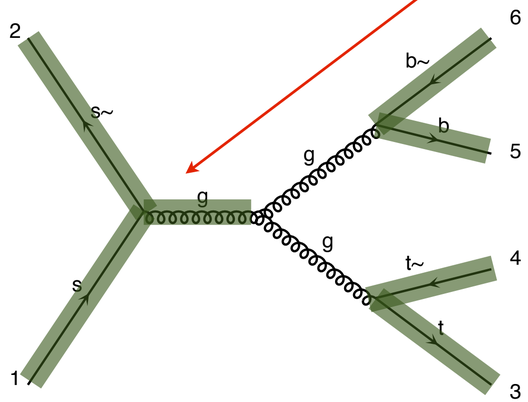
Number of routines for both: 7

$$\mathcal{M} = \mathcal{M}_1 + \mathcal{M}_2$$

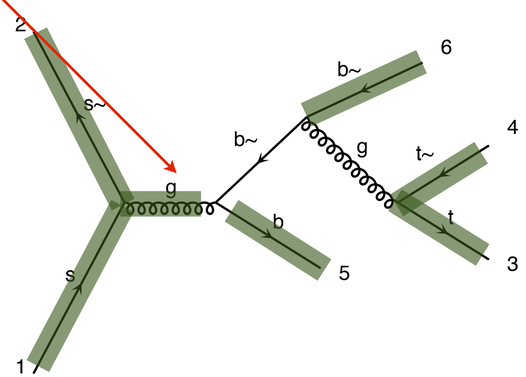
Share work between diagrams

Known

Identical



$\mathcal{M}_1$



$\mathcal{M}_2$

Number of routines: 7

Number of routines: 7

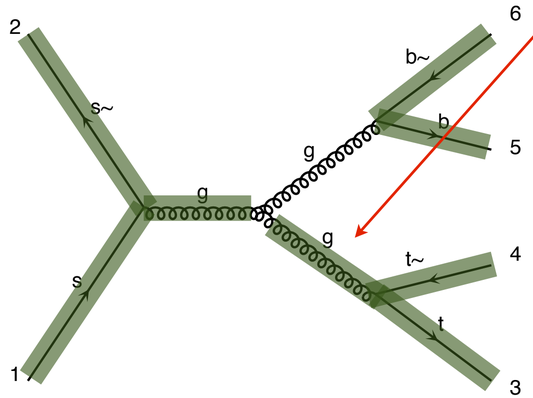
Number of routines for both: 7

$$\mathcal{M} = \mathcal{M}_1 + \mathcal{M}_2$$

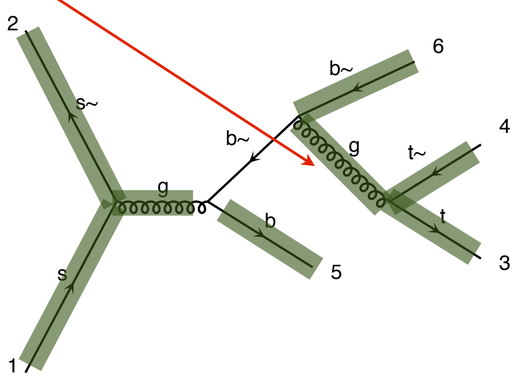
Share work between diagrams

Known

Identical



$\mathcal{M}_1$



$\mathcal{M}_2$

Number of routines: 8

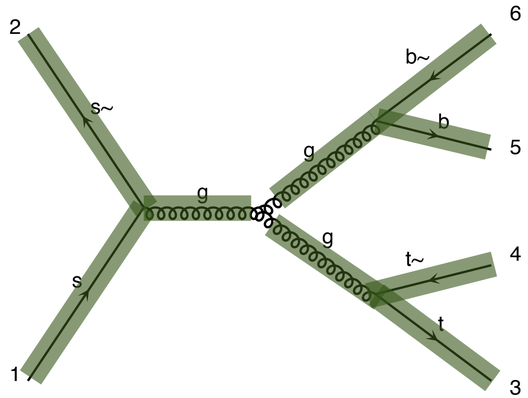
Number of routines: 8

Number of routines for both: 8

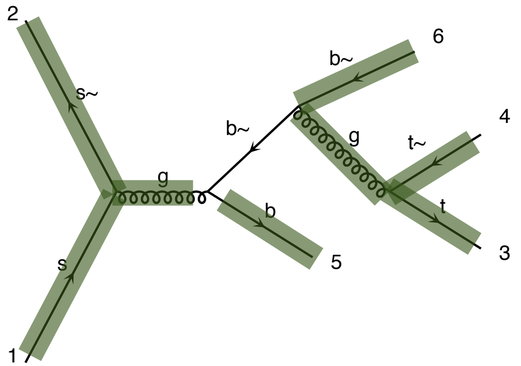
$$\mathcal{M} = \mathcal{M}_1 + \mathcal{M}_2$$

Share work between diagrams

Known



$\mathcal{M}_1$



$\mathcal{M}_2$

Number of routines: 9

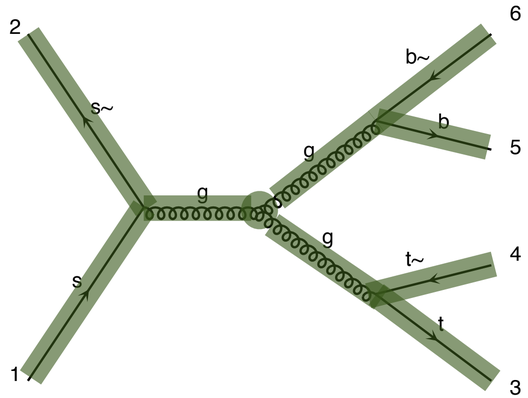
Number of routines: 8

Number of routines for both: 9

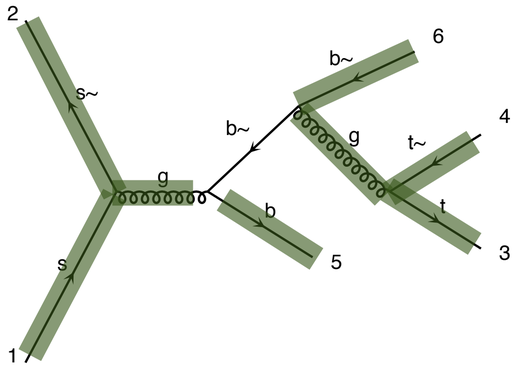
$$\mathcal{M} = \mathcal{M}_1 + \mathcal{M}_2$$

Share work between diagrams

Known



$\mathcal{M}_1$



$\mathcal{M}_2$

Number of routines: 10

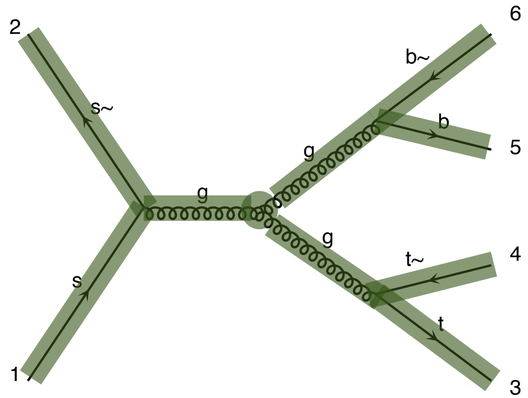
Number of routines: 8

Number of routines for both: 10

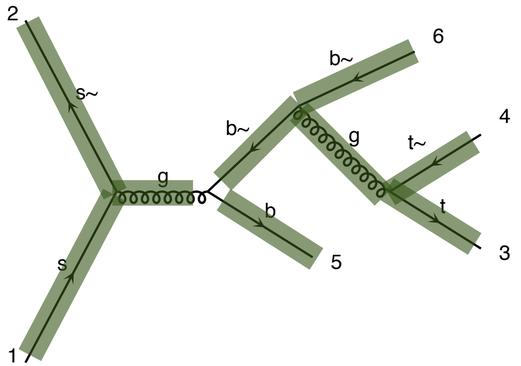
$$\mathcal{M} = \mathcal{M}_1 + \mathcal{M}_2$$

Share work between diagrams

Known



$\mathcal{M}_1$



$\mathcal{M}_2$

Number of routines: 10

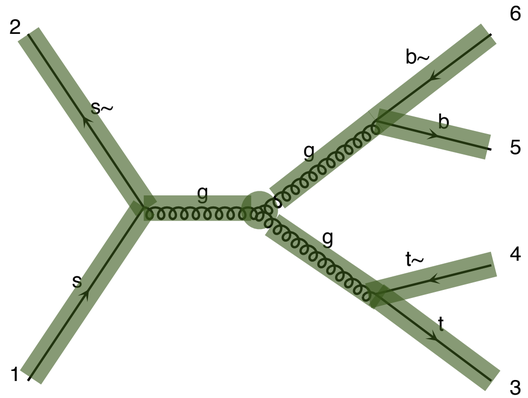
Number of routines: 9

Number of routines for both: 11

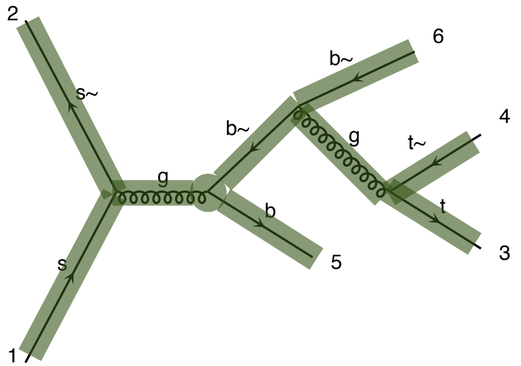
$$\mathcal{M} = \mathcal{M}_1 + \mathcal{M}_2$$

Share work between diagrams

Known



$\mathcal{M}_1$



$\mathcal{M}_2$

Number of routines: 10

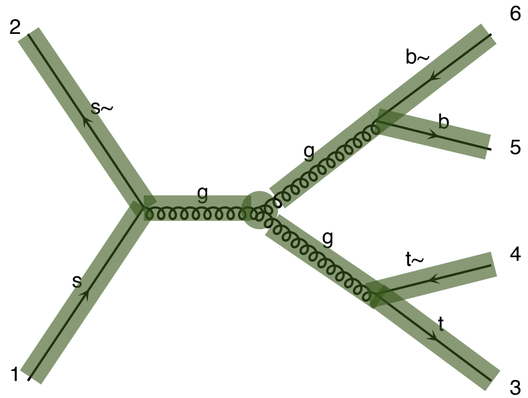
Number of routines: 10

Number of routines for both: 12

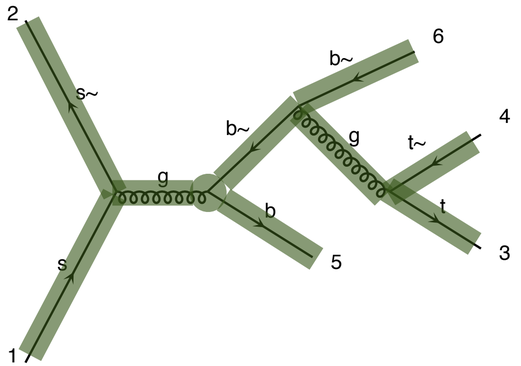
$$\mathcal{M} = \mathcal{M}_1 + \mathcal{M}_2$$

Share work between diagrams

Known



$\mathcal{M}_1$



$\mathcal{M}_2$

Number of routines: 10

2(N+1)

Number of routines: 10

2(N+1)

Number of routines for both: 12

$$\mathcal{M} = \mathcal{M}_1 + \mathcal{M}_2$$

Comparison

	M diagrams	N particles
Analytical	M^2	$(N!)^2$
Helicity	M	$(N!)\,2^N$
Recycling	M	$(N-1)!\,2^{(N-1)}$

Can we do even better?

- Recursion relation (used in Sherpa) [WIP]
- New in MG5aMC: Helicity Recycling [2102.00773]
- 5 Dimensional helicity wave function [2203.10440]
- Only leading-color computation [2601.19483]

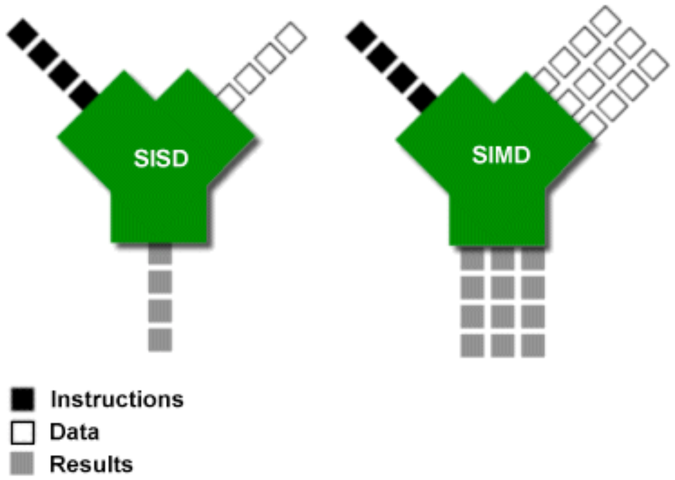
	M diagrams	N particles
Analytical	M^2	$(N!)^2$
Helicity	M	$(N!)\,2^N$
Recycling	M	$(N-1)!\,2^{(N-1)}$
Helicity Recycling	M	$\approx (N-1)!2^{N/2}$

Hardware acceleration

GPU

- Was first done a while ago: [0908.4403, 1305.0708]
- Recently released as fully-functional plugin for MG5: https://github.com/mg5amcnlo/mg5amcnlo_cudacpp
- Large speed-ups, less energy used per event [2507.21039]

SIMD



- Modern CPUs can act as a baby GPU: “Single Instruction, Multiple Data”
- Perform N identical operation as fast as one

Summary

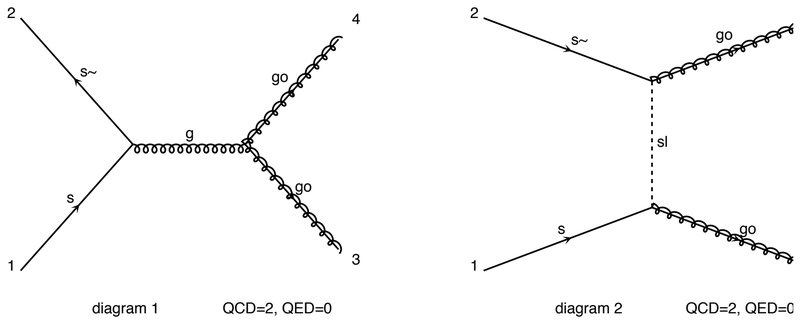
- ***Numerical computation faster than analytic*** one
  → Helicity amplitude method
- Lots of (ongoing) work on improving matrix element computation
  → reuse parts of computation
  → hardware acceleration
  → recursion rules
- ME computation automated in MG for
  → large number of final states
  → any BSM theory
  → loop computations (next lecture)

Outline

1. LHC basics
2. Matrix elements
3. **Monte Carlo integration**
4. Phase-space sampling
5. Event generation
6. Decays
7. Machine learning
8. Parton showers
9. Multijet merging

From process to cross section

1. Determine production mechanism → find all diagrams



Easy enough

2. Evaluate matrix element → use Feynman rules

$$|\mathcal{M}|^2$$

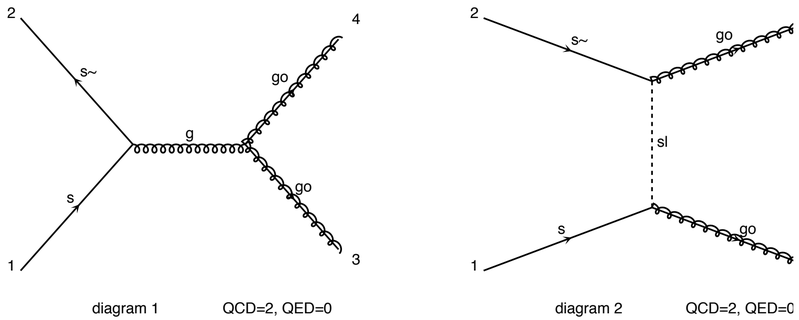
Hard

3. Phase-space integration

$$\hat{\sigma} = \frac{1}{2\hat{s}} \int \mathrm{d}\Phi_n \, |\mathcal{M}|^2$$

From process to cross section

1. Determine production mechanism → find all diagrams



Easy enough

2. Evaluate matrix element → use Feynman rules

$$|\mathcal{M}|^2$$

Hard

3. Phase-space integration

$$\hat{\sigma} = \frac{1}{2\hat{s}} \int \mathrm{d}\Phi_n \, |\mathcal{M}|^2$$

Very hard

Requirements

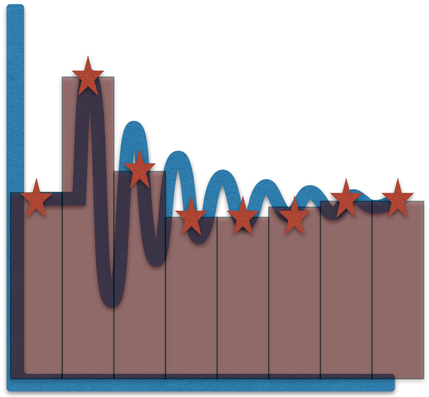
- cross section calculation requires high-dimensional integral

$$\hat{\sigma} = \frac{1}{2\hat{s}} \int \mathrm{d}\Phi_n \, |\mathcal{M}|^2 \qquad \dim[\Phi_n] = 3n - 4$$

- functions can be very peaked (resonances)
- must be general and flexible
- not only integration, but also event generation
  → necessarily involves randomness

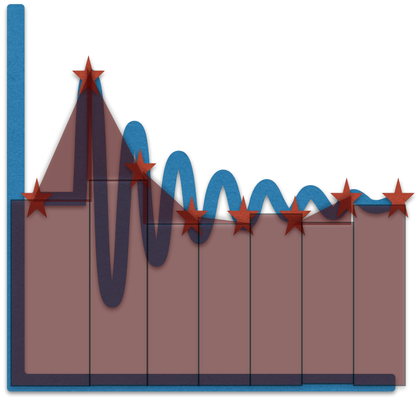
Numerical integration

Rectangles



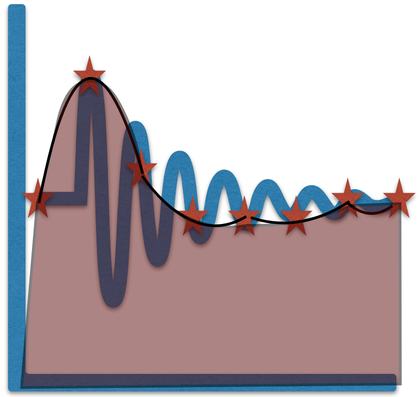
- Fixed grid
- Approximate by rectangles

Trapezium



- Fixed grid
- Linear interpolation

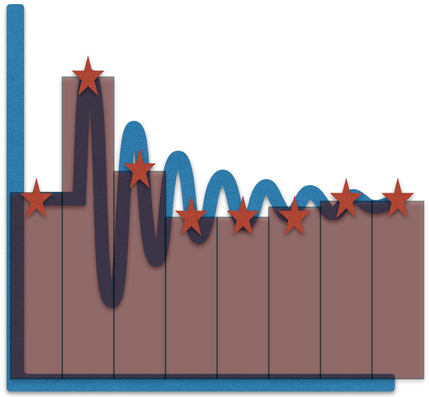
Simpson



- Fixed grid
- Quadratic interpolation

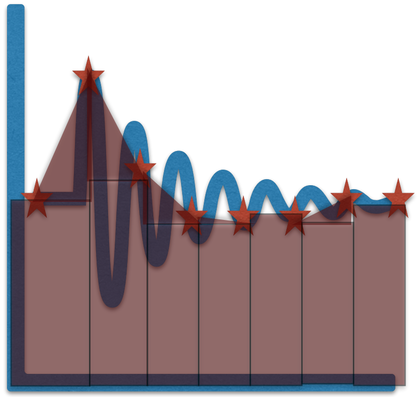
Numerical integration

Rectangles



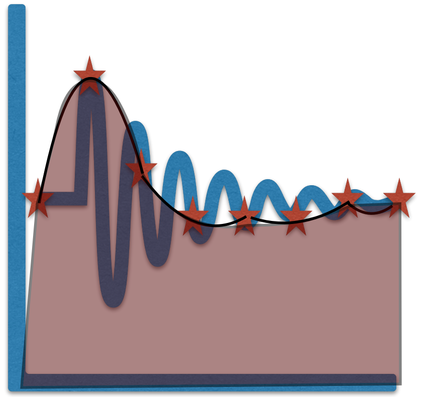
- Fixed grid
- Approximate by rectangles

Trapezium



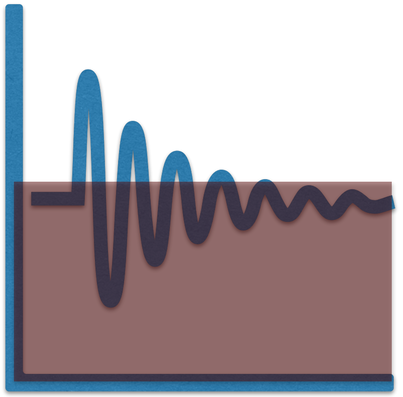
- Fixed grid
- Linear interpolation

Simpson



- Fixed grid
- Quadratic interpolation

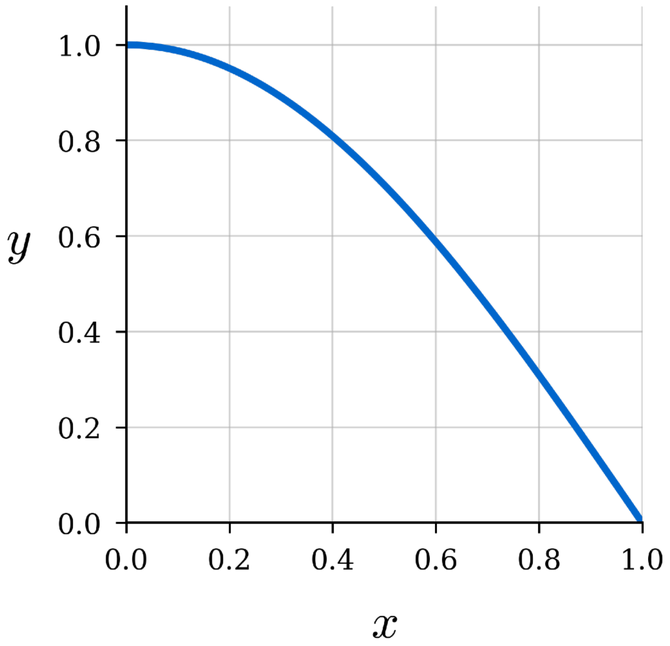
Monte Carlo



- Sample points randomly
- Compute average

Scaling of integration error

$$I = \int_0^1 \mathrm{d}x \cos \frac{\pi}{2} x$$

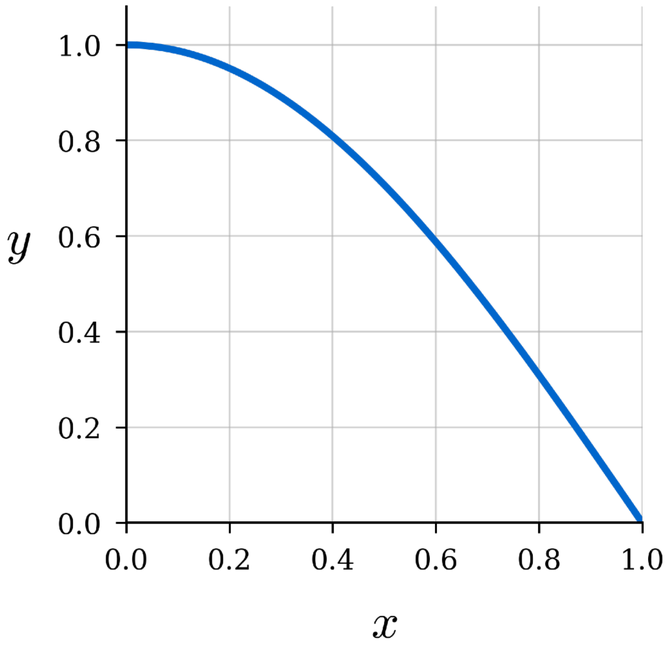


	Simpson	MC
3	0,638	0,3
5	0,6367	0,8
20	0,63662	0,6
100	0,636619	0,65
1000	0,636619	0,636

MC integration:
slow convergence in 1D

Scaling of integration error

$$I = \int_0^1 \mathrm{d}x \cos \frac{\pi}{2} x$$



	Simpson	MC
3	0,638	0,3
5	0,6367	0,8
20	0,63662	0,6
100	0,636619	0,65
1000	0,636619	0,636

MC integration:
slow convergence in 1D

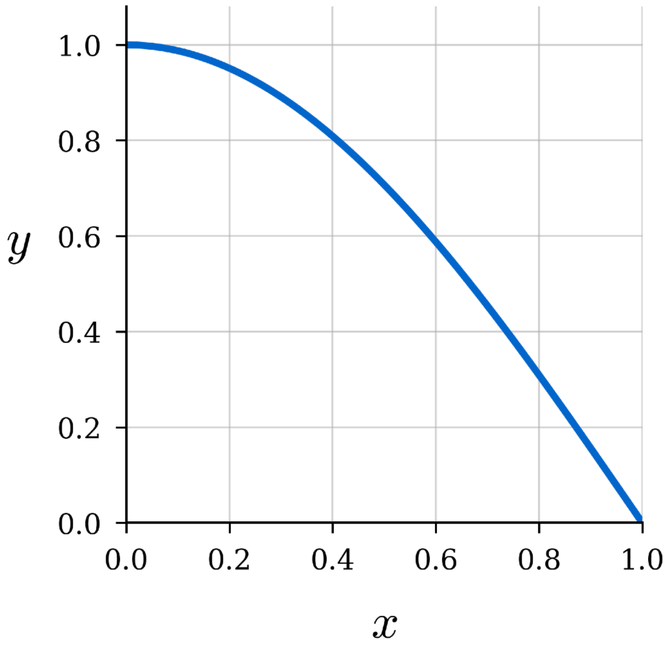
$1/\sqrt{N}$

$1/N^2$

$1/N^4$

Scaling of integration error

$$I = \int_0^1 \mathrm{d}x \cos \frac{\pi}{2} x$$



	Simpson	MC
3	0,638	0,3
5	0,6367	0,8
20	0,63662	0,6
100	0,636619	0,65
1000	0,636619	0,636

MC integration:
slow convergence in 1D

$1/\sqrt{N}$

$1/N^2$

$1/N^4$

d dimensions

- Monte Carlo $1/\sqrt{N}$
- Trapezium $1/N^{2/d}$
- Simpson $1/N^{4/d}$

Monte Carlo integration

Integral

$$I = \int_0^1 \mathrm{d}x\, f(x) \qquad I_N = \frac{1}{N}\sum_{i=0}^{N} f(x_i)$$

Variance

$$V = \int_0^1 \mathrm{d}x\, f(x)^2 - I^2 \qquad V_N = \frac{1}{N}\sum_{i=0}^{N} f(x_i)^2 - I_N^2$$

MC integration error:

$$I = I_N \pm \sqrt{\frac{V_N}{N}}$$

Monte Carlo integration

Integral

$$I = \int_0^1 \mathrm{d}x\, f(x) \qquad I_N = \frac{1}{N}\sum_{i=0}^{N} f(x_i)$$

Variance

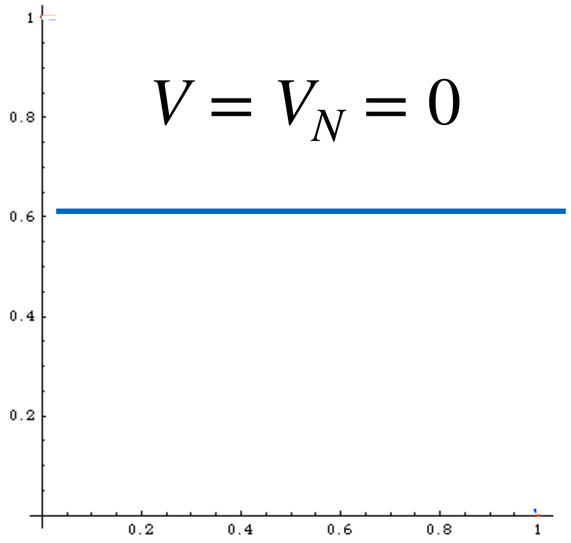
$$V = \int_0^1 \mathrm{d}x\, f(x)^2 - I^2 \qquad V_N = \frac{1}{N}\sum_{i=0}^{N} f(x_i)^2 - I_N^2$$

MC integration error:

$$I = I_N \pm \sqrt{\frac{V_N}{N}}$$

Can be reduced!

Best case: flat integrand

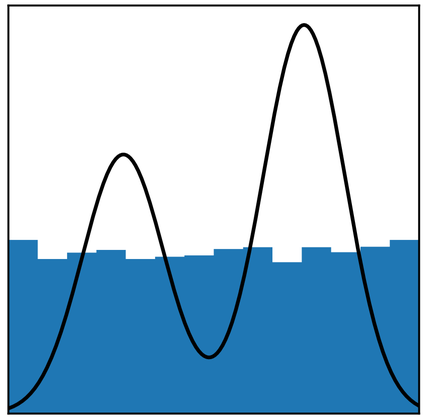


Importance sampling

$$I = \int \mathrm{d}x \, f(x)$$

Importance sampling

$$I = \int \mathrm{d}x\, f(x)$$

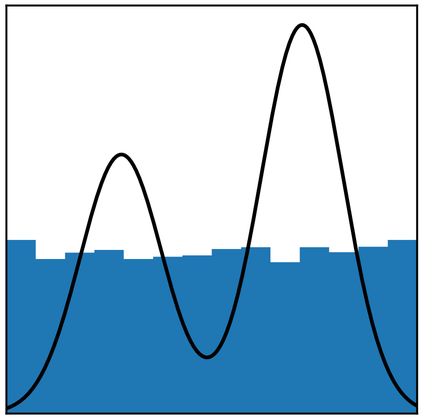


Flat sampling
inefficient

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

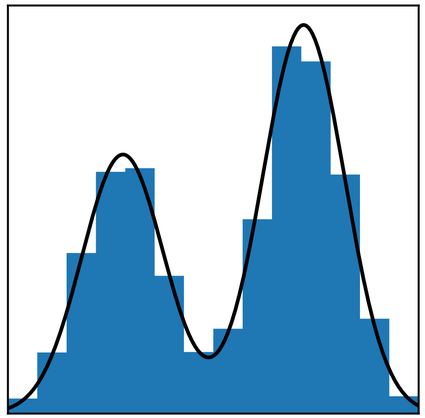
Importance sampling

$$I = \int \mathrm{d}x\, f(x)$$



Flat sampling
inefficient

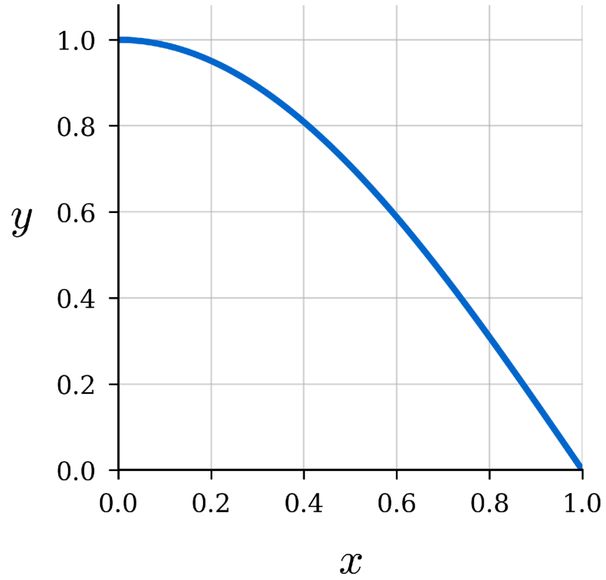
$$I = \langle f(x) \rangle_{x \sim p(x)}$$



Importance sampling
Find mapping close to integrand

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

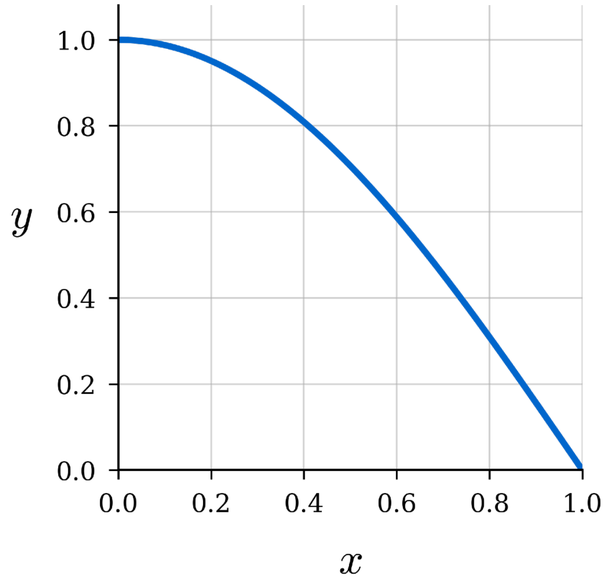
1D example

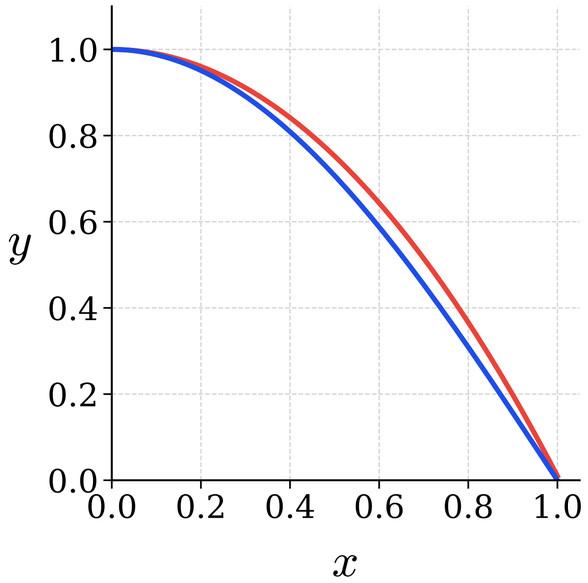


$$I = \int_0^1 \mathrm{d}x \cos \frac{\pi}{2} x$$

$$I_N = 0.637 \pm 0.307/\sqrt{N}$$

1D example

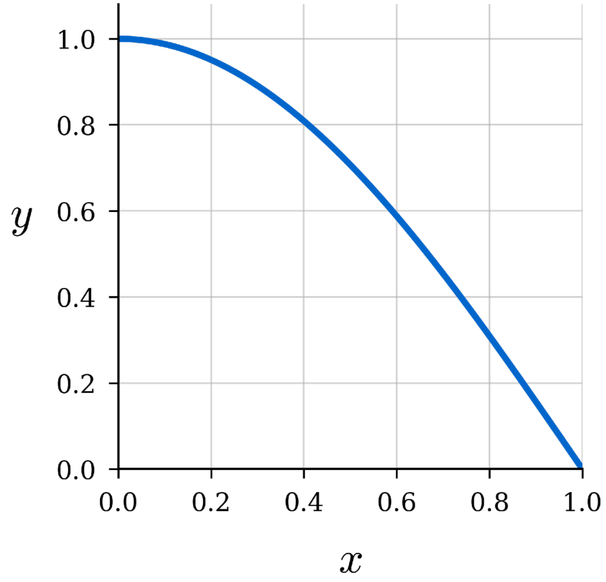


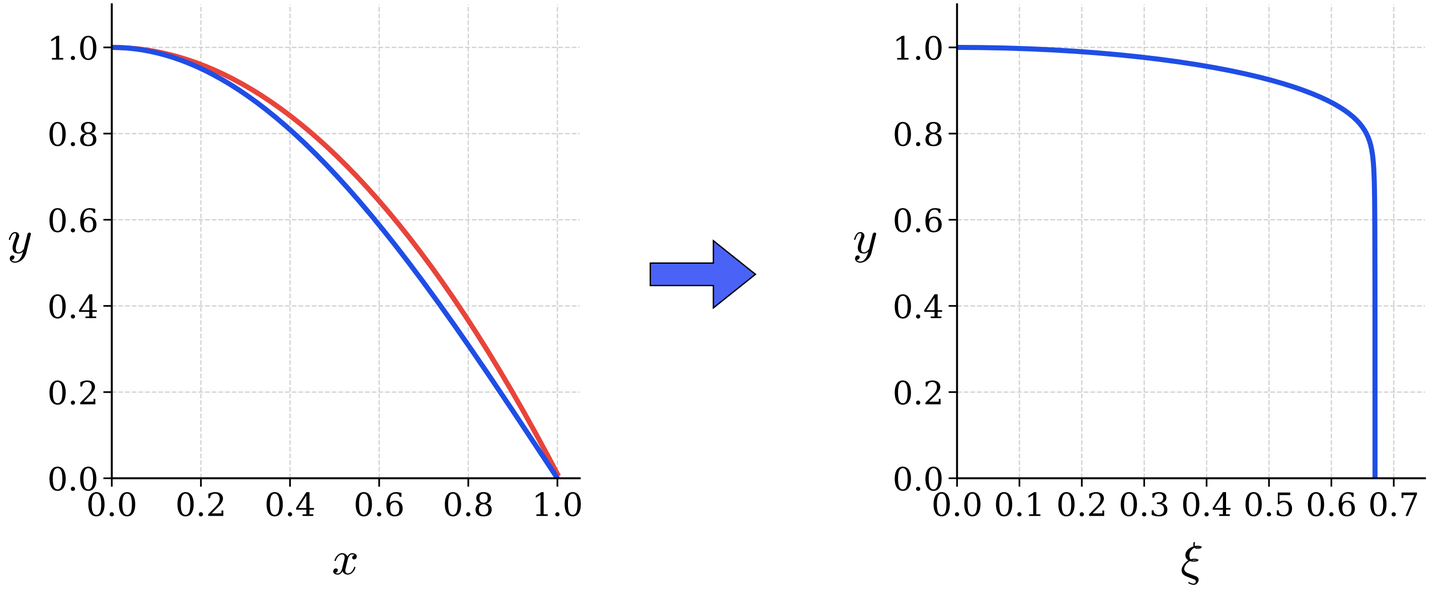


$$I = \int_0^1 \mathrm{d}x \cos\frac{\pi}{2}x \qquad I = \int_0^1 \mathrm{d}x(1-cx^2)\frac{\cos\frac{\pi}{2}x}{(1-cx^2)}$$

$$I_N = 0.637 \pm 0.307/\sqrt{N}$$

1D example

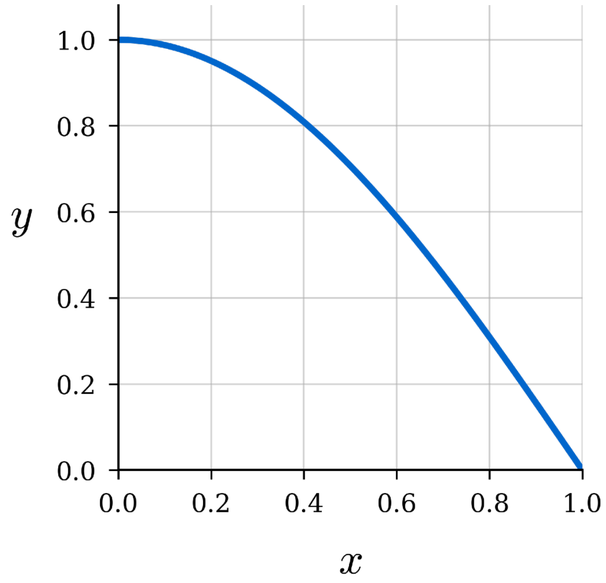


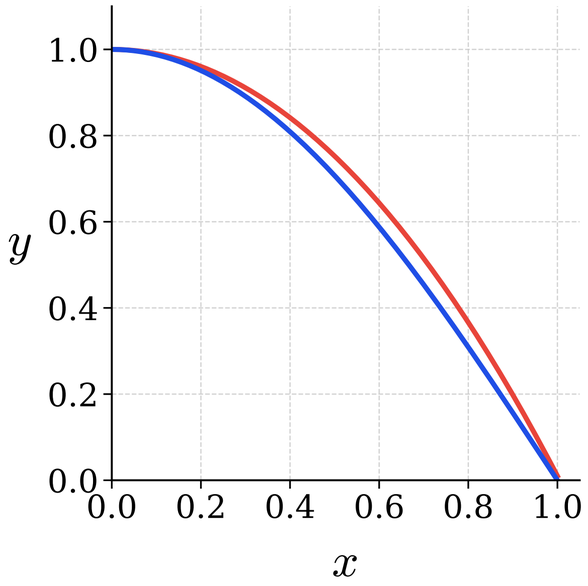


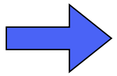
$$I = \int_0^1 \mathrm{d}x \cos\frac{\pi}{2}x \qquad I = \int_0^1 \mathrm{d}x(1-cx^2)\frac{\cos\frac{\pi}{2}x}{(1-cx^2)} = \int_{\xi_1}^{\xi_2} \mathrm{d}\xi \frac{\cos\frac{\pi}{2}x[\xi]}{1-cx[\xi]^2}$$

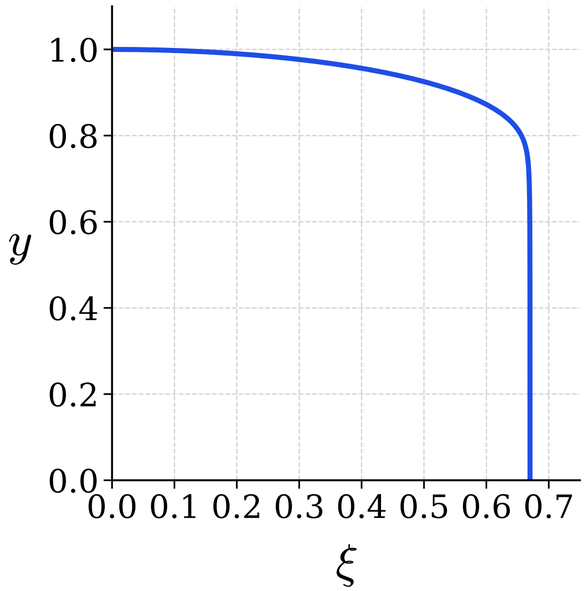
$$I_N = 0.637 \pm 0.307/\sqrt{N}$$

1D example









$$I = \int_0^1 \mathrm{d}x \cos\frac{\pi}{2}x$$

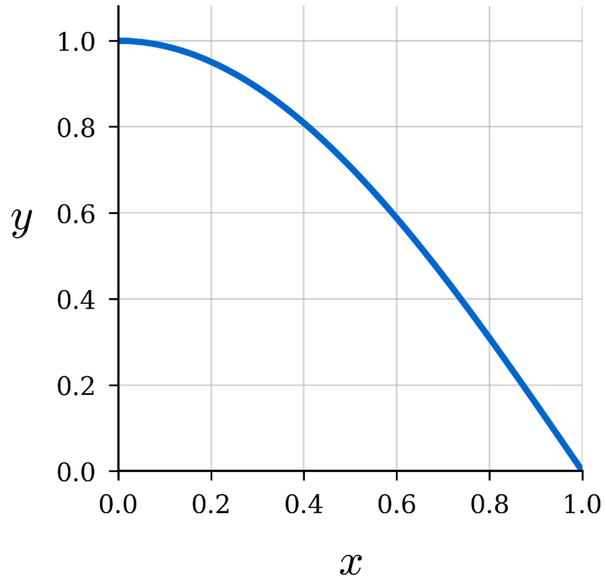
$$I = \int_0^1 \mathrm{d}x (1 - cx^2) \frac{\cos\frac{\pi}{2}x}{(1 - cx^2)} = \int_{\xi_1}^{\xi_2} \mathrm{d}\xi \frac{\cos\frac{\pi}{2}x[\xi]}{1 - cx[\xi]^2}$$

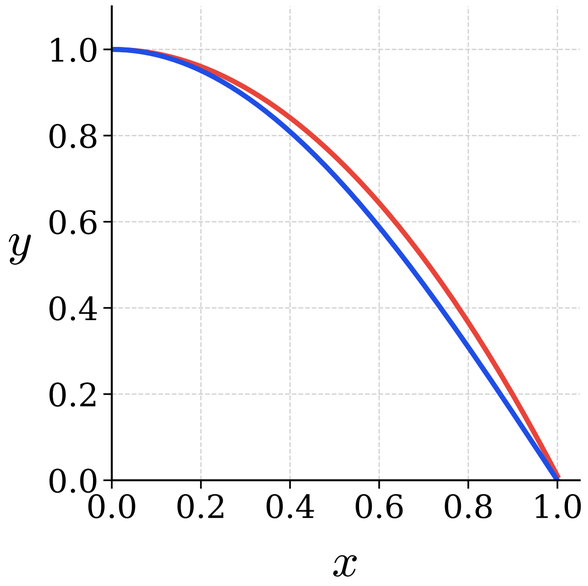
$$I_N = 0.637 \pm 0.307/\sqrt{N}$$

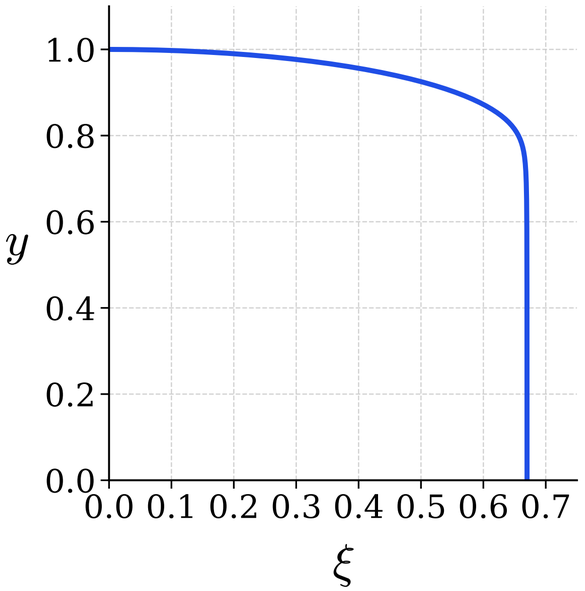
$$I_N = 0.637 \pm 0.031/\sqrt{N}$$

approx. flat!

1D example







$$I = \int_0^1 \mathrm{d}x \cos\frac{\pi}{2}x$$

$$I = \int_0^1 \mathrm{d}x (1 - cx^2) \frac{\cos\frac{\pi}{2}x}{(1 - cx^2)} = \int_{\xi_1}^{\xi_2} \mathrm{d}\xi \frac{\cos\frac{\pi}{2}x[\xi]}{1 - cx[\xi]^2}$$

$$I_N = 0.637 \pm 0.307/\sqrt{N}$$

$$I_N = 0.637 \pm 0.031/\sqrt{N}$$

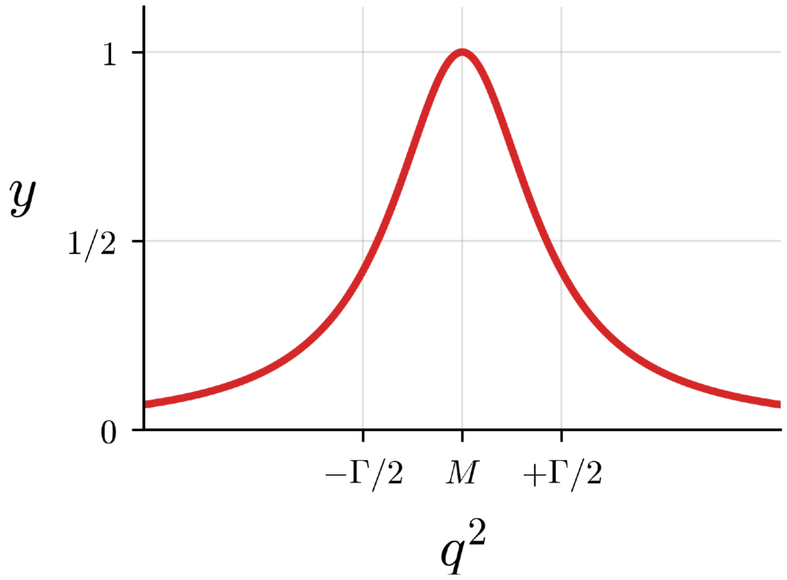
approx. flat!

Choice of parameterization important for efficient computation!

Flattening Breit-Wigner

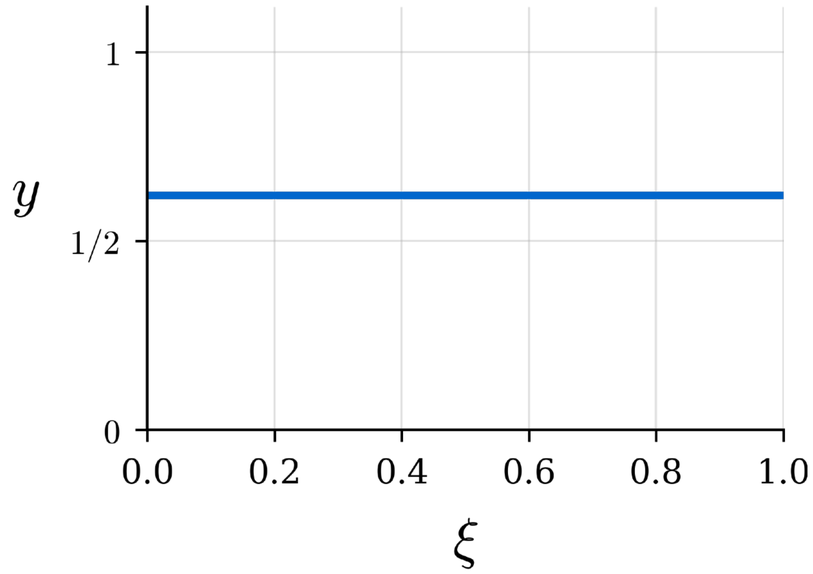
Important example: Resonant propagators

$$\int \frac{\mathrm{d}q^2}{(q^2 - M^2 + iM\Gamma)^2}$$



Change of variables

$$\xi = \arctan\left(\frac{q^2 - M^2}{M\,\Gamma}\right)$$



Simple and effective!
Widths and masses are known parameters

Summary

- Evaluate integrand at random points and take the average
- Monte Carlo integration ***performs well in high dimensions***

  → integration error scales with $\sqrt{N}$
- ***Importance sampling***: find smart proposal distribution

  → change of variables that ***flattens the integrand***

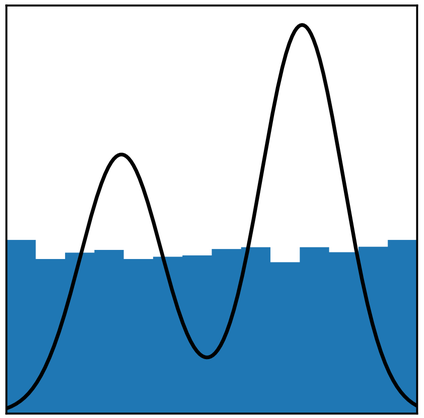
  → reduces variance of integral estimate
- Very flexible: able to deal with complicated cuts, etc.

Outline

1. LHC basics
2. Matrix elements
3. Monte Carlo integration
4. **Phase-space sampling**
5. Event generation
6. Decays
7. Machine learning
8. Parton showers
9. Multijet merging

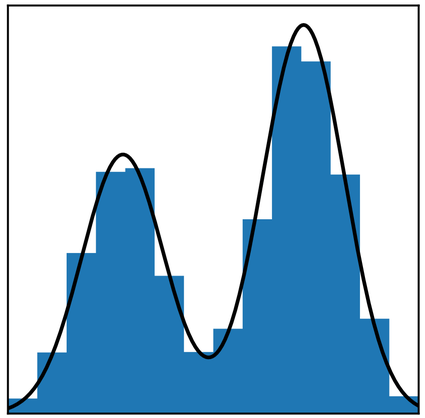
Monte Carlo Integration

$$I = \int \mathrm{d}x\, f(x)$$



Flat sampling
inefficient

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

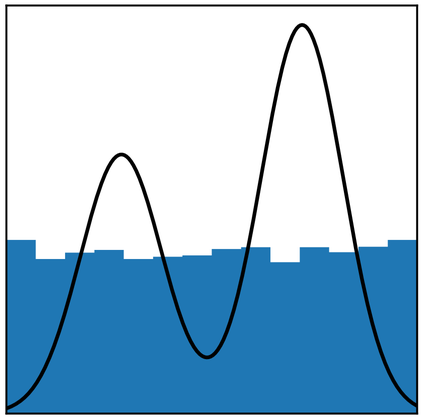


Importance sampling
Find mapping close to integrand

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

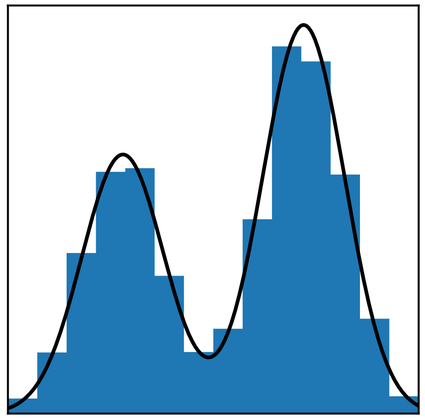
Monte Carlo Integration

$$I = \int \mathrm{d}x\, f(x)$$



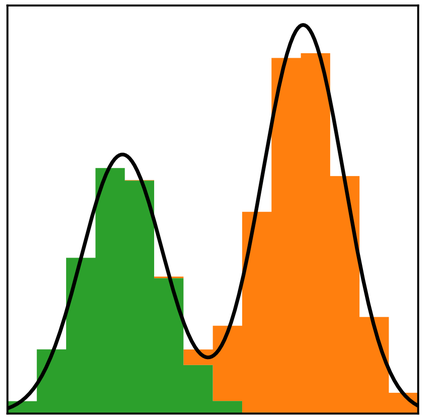
Flat sampling
inefficient

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



Importance sampling
Find mapping close to integrand

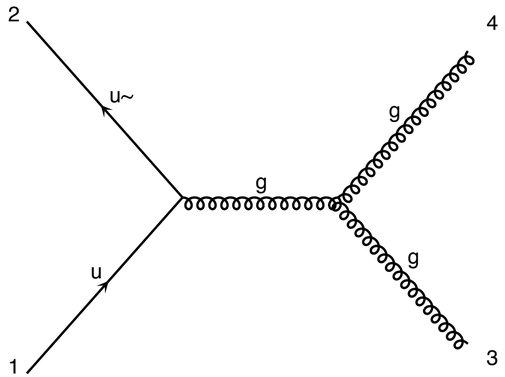
$$I = \left\langle \frac{f(x)}{g(x)} \right\rangle_{x \sim g(x)}$$



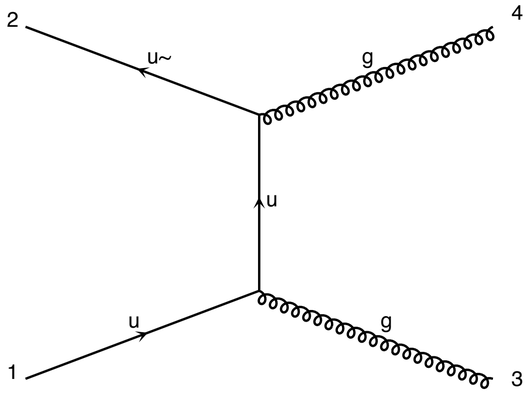
Multi-channeling
one mapping for each channel

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

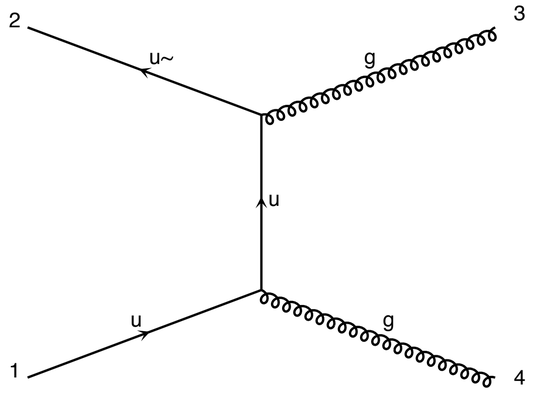
Single diagram enhancement



$$\propto \frac{1}{\hat{s}} = \frac{1}{(p_1 + p_2)^2}$$



$$\propto \frac{1}{\hat{t}} = \frac{1}{(p_1 - p_3)^2}$$



$$\propto \frac{1}{\hat{u}} = \frac{1}{(p_1 - p_4)^2}$$

Three very different pole structures
contribute to the same matrix element!

Single diagram enhancement

Trick in MadEvent: Split the complexity

$$\int |\mathcal{M}_{\text{tot}}|^2 = \int \frac{\sum_i |\mathcal{M}_i|^2}{\sum_j |\mathcal{M}_j|^2} |\mathcal{M}_{\text{tot}}|^2 = \sum_i \int \underbrace{\frac{|\mathcal{M}_i|^2}{\sum_j |\mathcal{M}_j|^2}}_{\approx 1} |\mathcal{M}_{\text{tot}}|^2$$

- Build integration channel ***for each Feynman diagram***
- Single diagrams are “easier” to integrate
  → pole structure and integration variables known from propagators
- Computationally cheap:
  → channel weights already computed during $|\mathcal{M}|^2$ calculation
  → channels independent: easy to parallelize
- Suboptimal when ***interferences are large***

Single diagram enhancement

$$\int |\mathcal{M}_{\text{tot}}|^2 = \int \frac{\sum_i |\mathcal{M}_i|^2}{\sum_j |\mathcal{M}_j|^2} |\mathcal{M}_{\text{tot}}|^2 = \sum_i \int \frac{|\mathcal{M}_i|^2}{\sum_j |\mathcal{M}_j|^2} |\mathcal{M}_{\text{tot}}|^2$$

P1_qq_wpwm

s= 725.73 ± 2.07 (pb)

Graph	Cross-Section ↓	Error	Events (K)	Unwgt	Luminosity
G2.2	377.6	1.67	142.285	7941.0	21
G3	239	1.16	220.04	10856.0	45.5
G1	109.1	0.378	70.88	3793.0	34.8

P1_gg_wpwm

s= 20.714 ± 0.332 (pb)

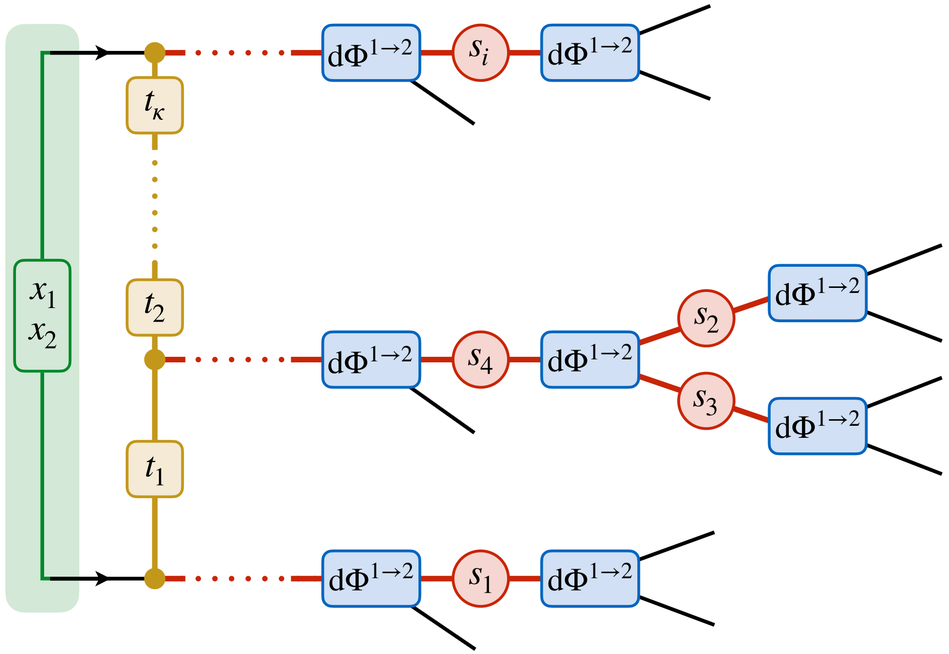
Graph	Cross-Section ↓	Error	Events (K)	Unwgt	Luminosity
G1.2	20.71	0.332	7.01	373.0	18

You can look at the individual channels in the MadGraph web interface

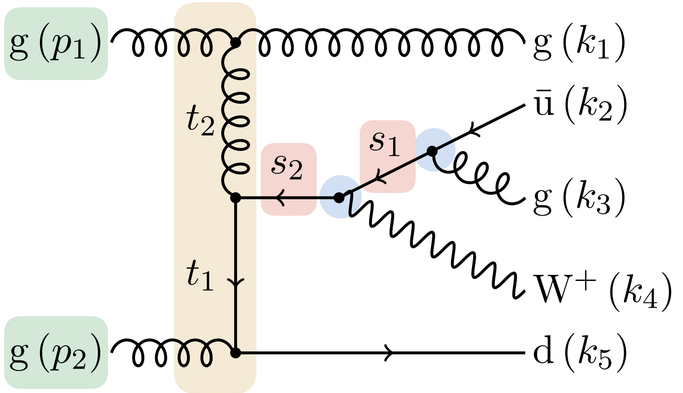
Attention:

Channel integrals are not physical!
(typically not gauge invariant)

Phase-space mappings



- Identify topology of Feynman diagram



Example:
W+jets

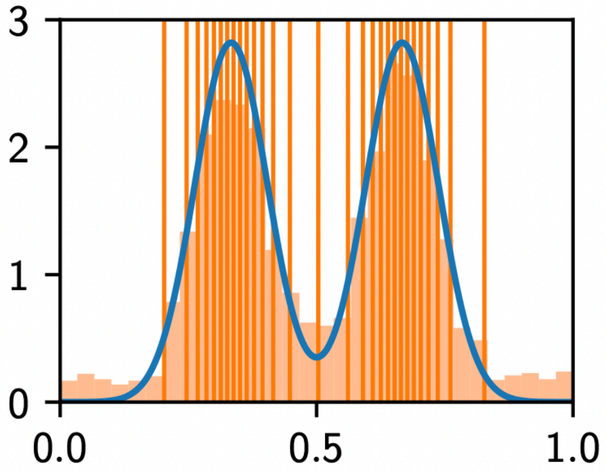
- Then build change of variables from standard components
  → Luminosity (PDF conv.) mapping
  → Invariants: ~1/s or Breit-Wigner
  → Two-particle decays
  → Two-particle scattering

VEGAS algorithm

Factorize probability

$$g(x) = g(x_1) \cdots g(x_n)$$

Fit bins with equal probability and varying width



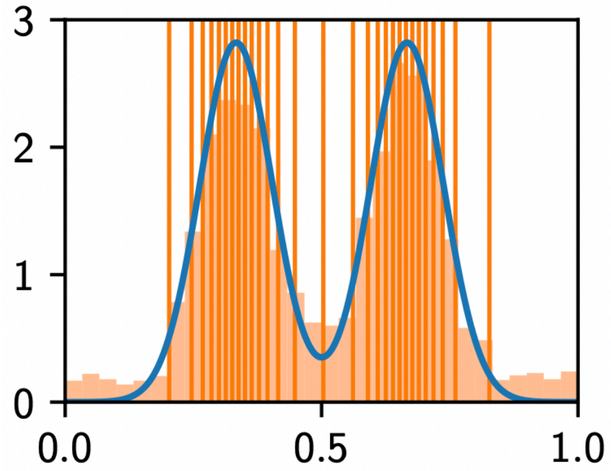
[G. P. Lepage, 1978]

VEGAS algorithm

Factorize probability

$$g(x) = g(x_1) \cdots g(x_n)$$

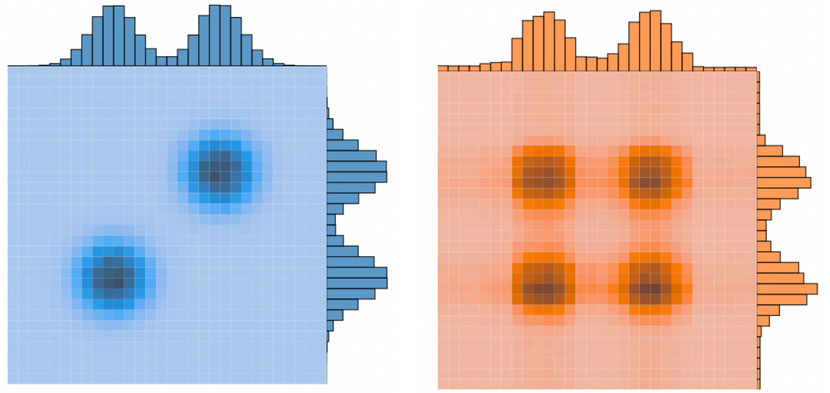
Fit bins with equal probability and varying width



⊕ Computationally cheap

⊖ High-dim and rich peaking functions
→ slow convergence

⊖ Peaks not aligned with grid axes
→ phantom peaks



[G. P. Lepage, 1978]

Summary

- Finding a good proposal distribution is hard

  → **simplify by splitting integral** into channels
- Single diagram enhancement

  → use Feynman diagrams, **weight by squared matrix element**
- Phase-space mappings for single diagrams

  → simple, reusable building blocks
- Further refinement with **VEGAS** algorithm

  → **factorize distribution**, fit bins with equal probability

Outline

1. LHC basics
2. Matrix elements
3. Monte Carlo integration
4. Phase-space sampling
5. **Event generation**
6. Decays
7. Machine learning
8. Parton showers
9. Multijet merging

Unweighting

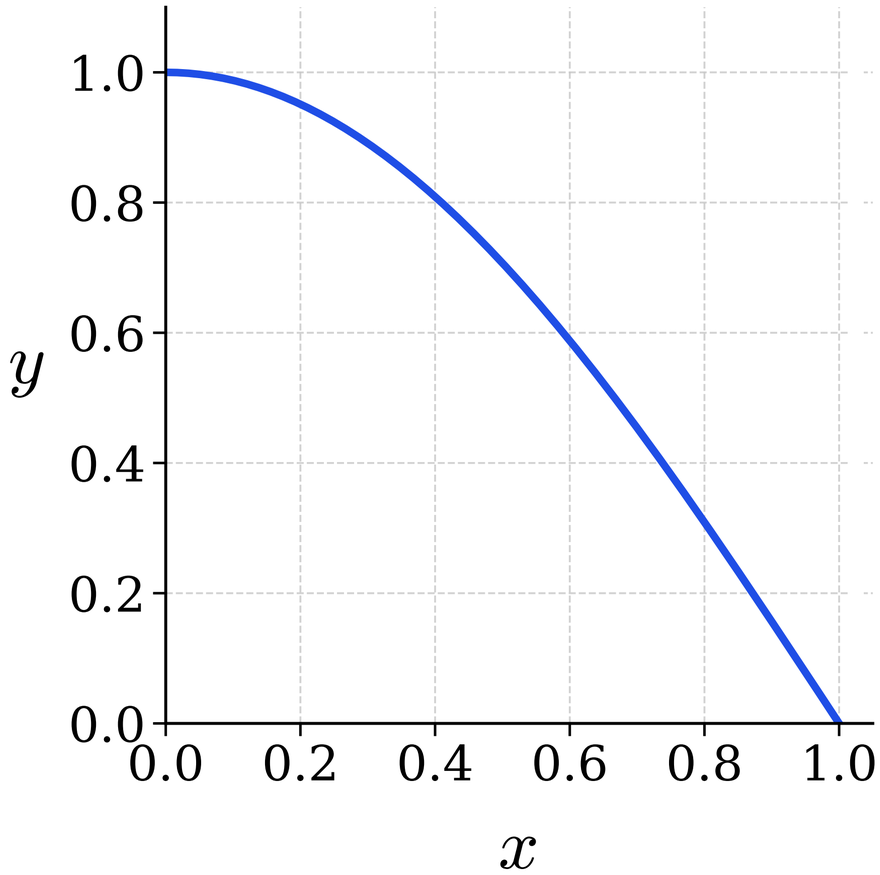
- We want to use our integrator also for event generation
  → **MC integration** already gives us **weighted samples**

$$w_i = \frac{f(x_i)}{g(x_i)}$$

- Problems with many low-weight events
  → need lots of memory to store
  → costly following simulation steps

- Idea: **unweighted events**
  → every event should have a weight of 1
  → **distribution** of events **as observed at collider**

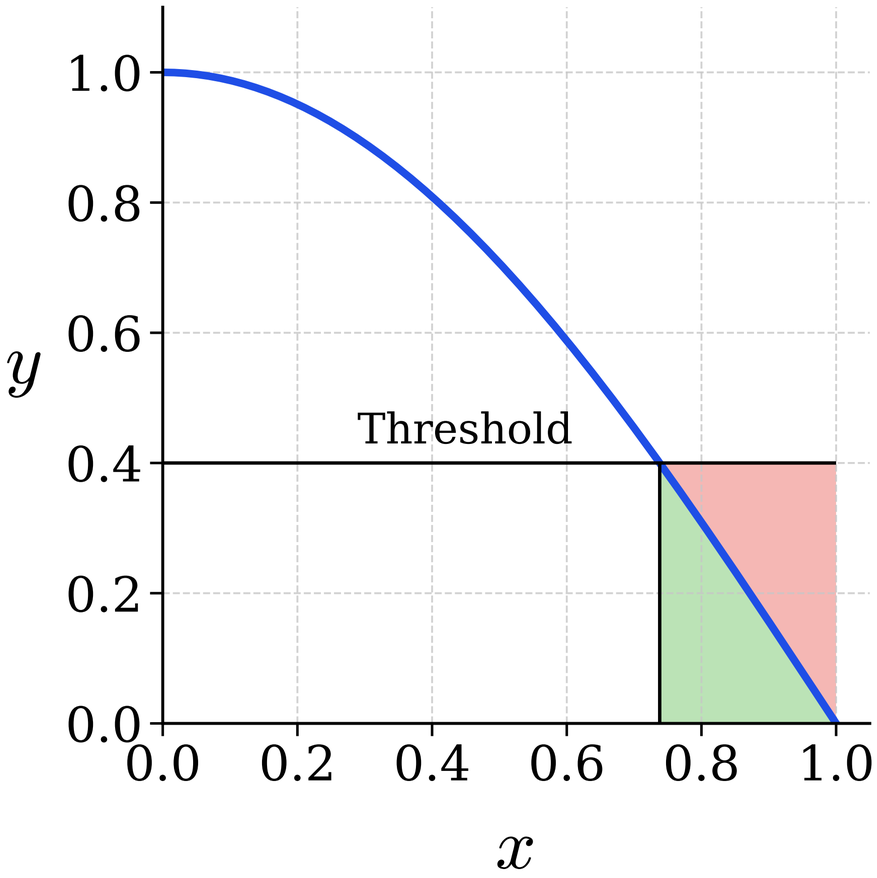
Unweighting



Requirements:

- want to reduce number of events
- (weighted) distribution should stay the same

Unweighting



Requirements:

- want to reduce number of events
- (weighted) distribution should stay the same

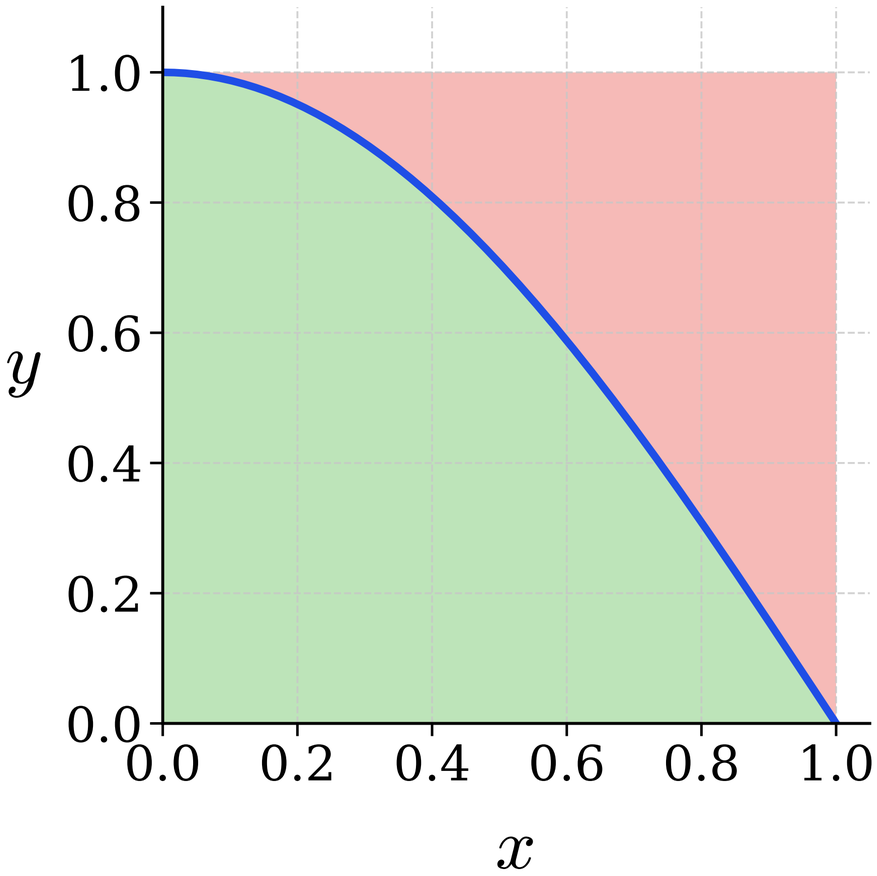
Idea: set threshold

- accept all events above threshold
- below threshold:

  keep with probability w_i / w_{thres}
- if accepted: need to compensate by

  multiplying weight with w_{thres} / w_i
- new weight is w_{thres}

Unweighting



Requirements:

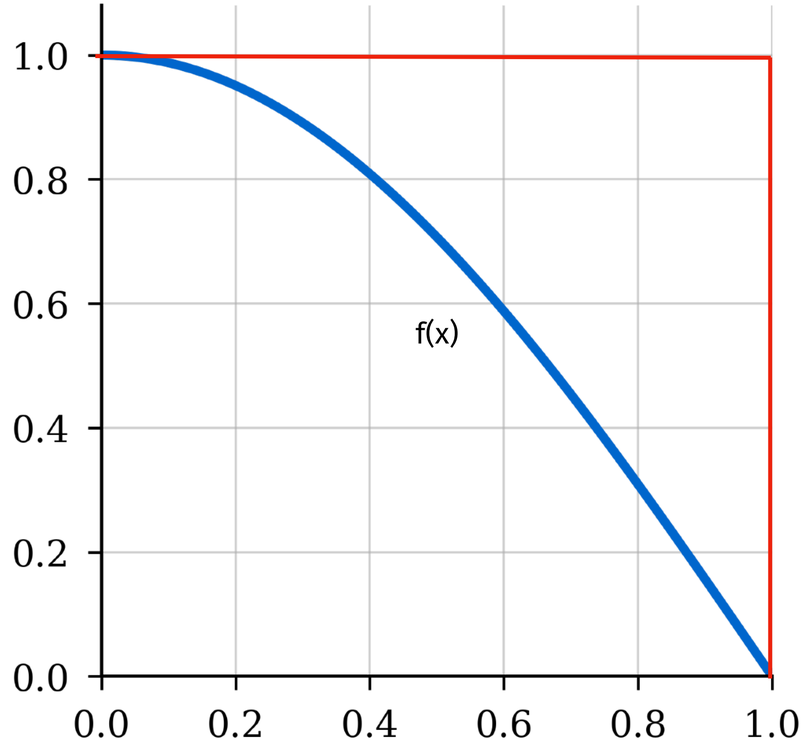
- want to reduce number of events
- (weighted) distribution should stay the same

Unweighted sample:

- set threshold to $w_{\max}$
- keep with probability $w_i / w_{\max}$
- all weights now have same weight $w_{\max}$
- maximal compression

Rejection sampling

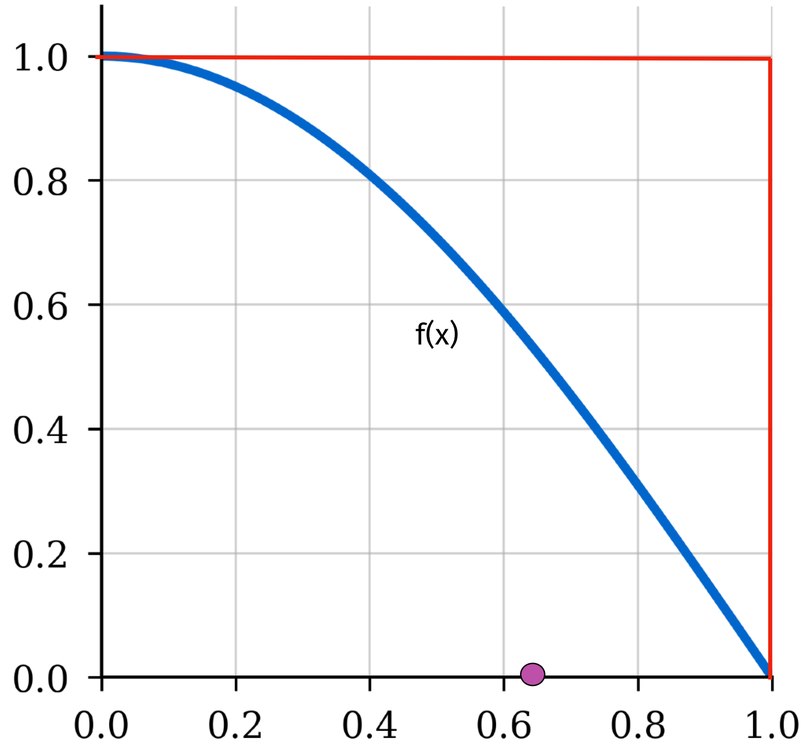
$$\int \mathrm{d}x\, f(x) \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i) = \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{w_{\max}} w_{\max} \equiv \frac{1}{N}\sum_{i=1}^{N} \frac{w_i}{w_{\max}} w_{\max}$$



Algorithm

Rejection sampling

$$\int \mathrm{d}x\, f(x) \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i) = \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{w_{\max}} w_{\max} \equiv \frac{1}{N}\sum_{i=1}^{N} \frac{w_i}{w_{\max}} w_{\max}$$

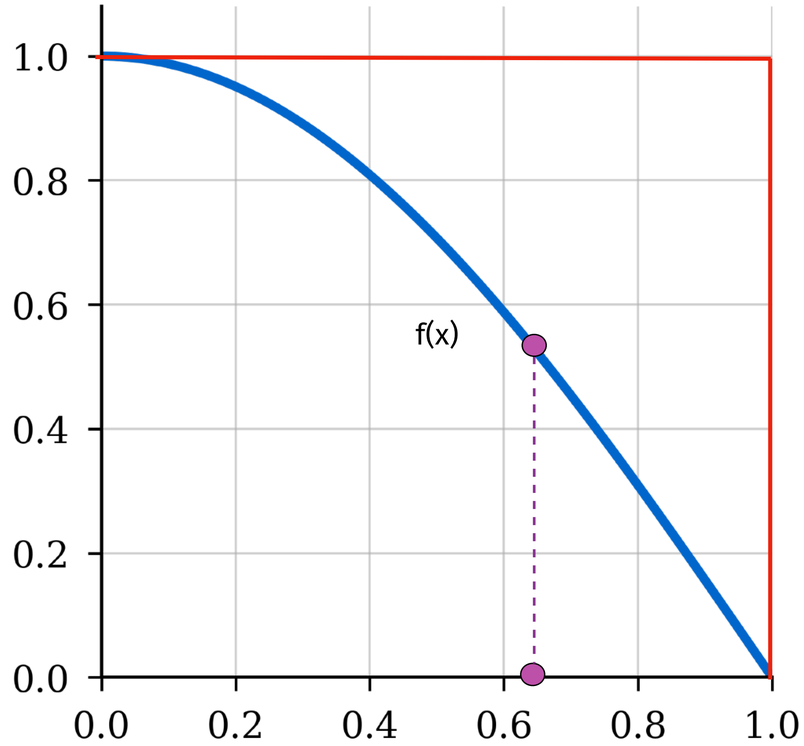


Algorithm

1. pick x_i

Rejection sampling

$$\int \mathrm{d}x\, f(x) \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i) = \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{w_{\max}} w_{\max} \equiv \frac{1}{N}\sum_{i=1}^{N} \frac{w_i}{w_{\max}} w_{\max}$$

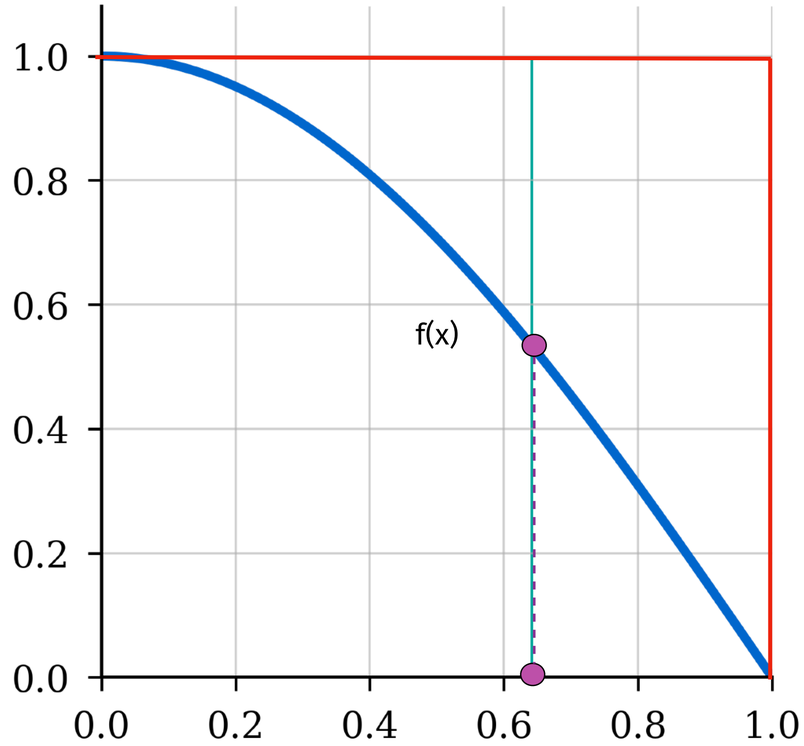


Algorithm

1. pick x_i
2. calculate $f(x_i) \equiv w_i$

Rejection sampling

$$\int \mathrm{d}x\, f(x) \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i) = \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{w_{\max}} w_{\max} \equiv \frac{1}{N}\sum_{i=1}^{N} \frac{w_i}{w_{\max}} w_{\max}$$

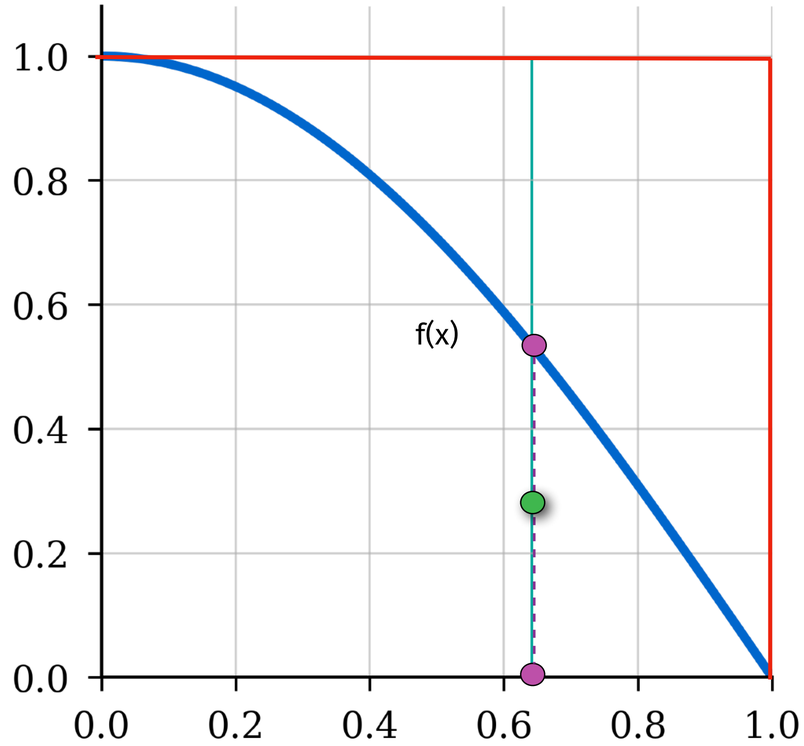


Algorithm

1. pick x_i
2. calculate $f(x_i) \equiv w_i$
3. pick $y \in [0, \max(f) \equiv w_{\max}]$

Rejection sampling

$$\int \mathrm{d}x\, f(x) \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i) = \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{w_{\max}} w_{\max} \equiv \frac{1}{N}\sum_{i=1}^{N} \frac{w_i}{w_{\max}} w_{\max}$$

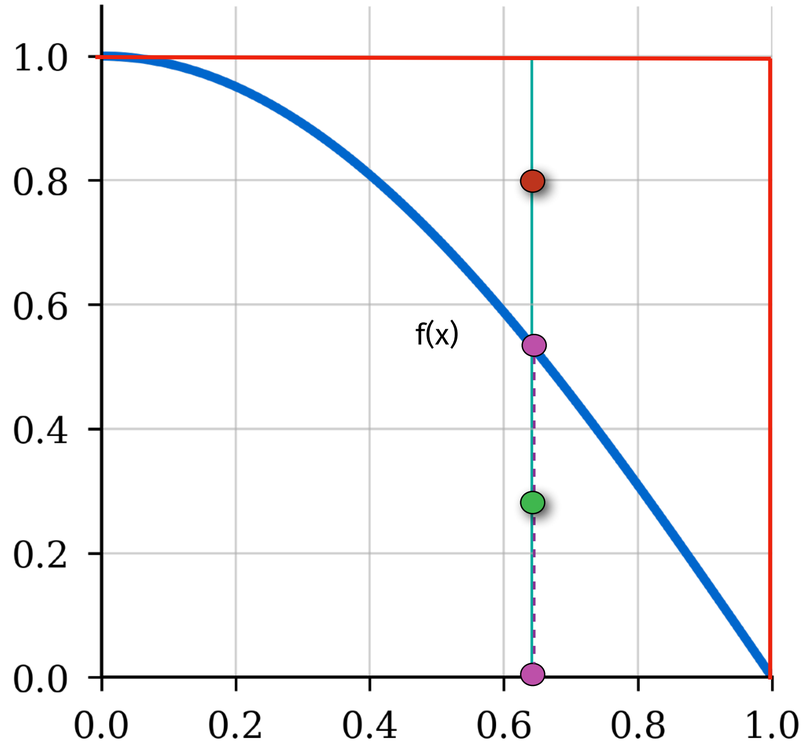


Algorithm

1. pick x_i
2. calculate $f(x_i) \equiv w_i$
3. pick $y \in [0, \max(f) \equiv w_{\max}]$
4. Compare:
   if $y < w_i$ accept event,

Rejection sampling

$$\int \mathrm{d}x\, f(x) \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i) = \frac{1}{N}\sum_{i=1}^{N} \frac{f(x_i)}{w_{\max}} w_{\max} \equiv \frac{1}{N}\sum_{i=1}^{N} \frac{w_i}{w_{\max}} w_{\max}$$



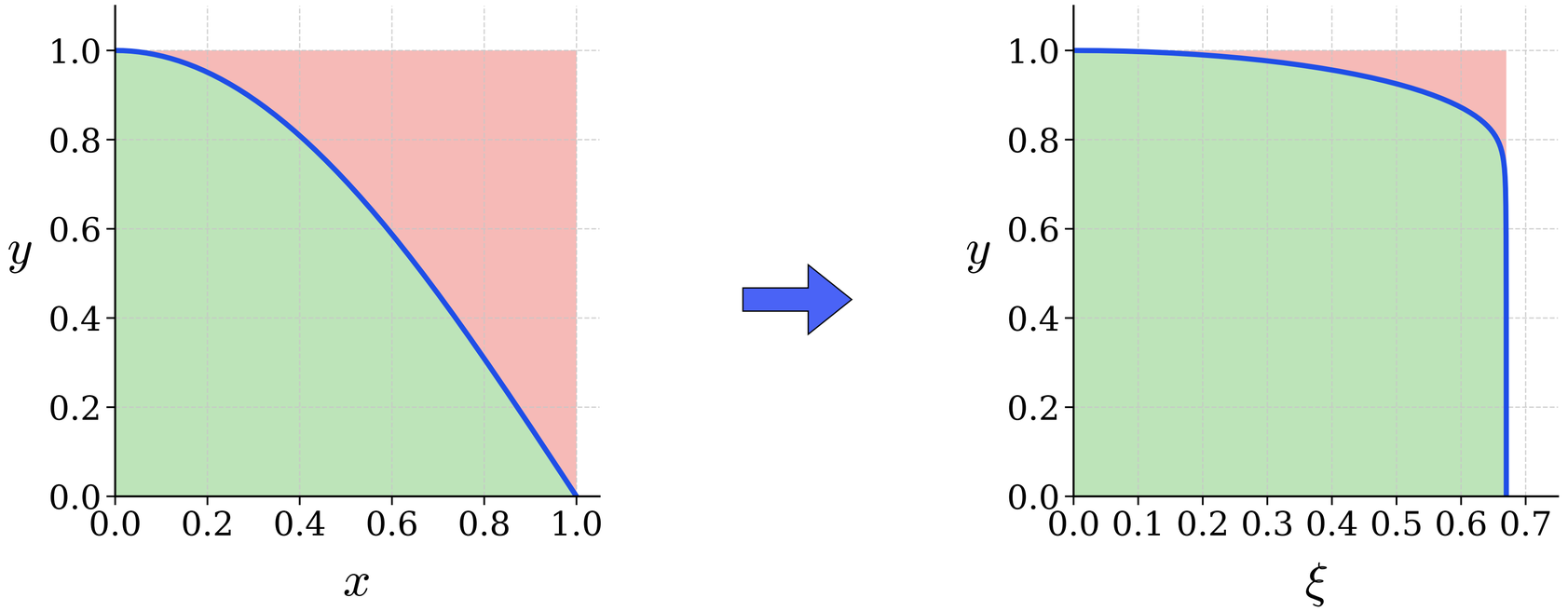
Algorithm

1. pick x_i
2. calculate $f(x_i) \equiv w_i$
3. pick $y \in [0, \max(f) \equiv w_{\max}]$
4. Compare:
   if $y < w_i$ accept event,

   else reject it.

Unweighting efficiency

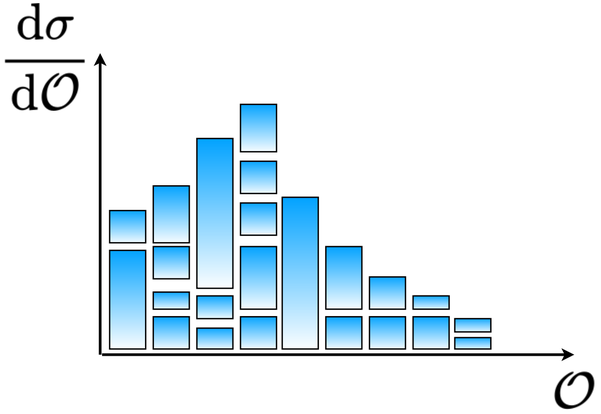
$$\int \mathrm{d}x\, f(x) = \int \mathrm{d}y\, \frac{f(y)}{g(y)} \approx \frac{1}{N}\sum_{i=1}^{N} \frac{f(y_i)}{g(y_i)} = \frac{1}{N}\sum_{i=1}^{N} \frac{f(y_i)}{g(y_i)\ w_{\max}} w_{\max} = \frac{1}{N}\sum_{i=1}^{N} \frac{w_i}{w_{\max}} w_{\max}$$



Smaller variance → $w/w_{\max}$ closer to 1
→ less rejected events, faster generation

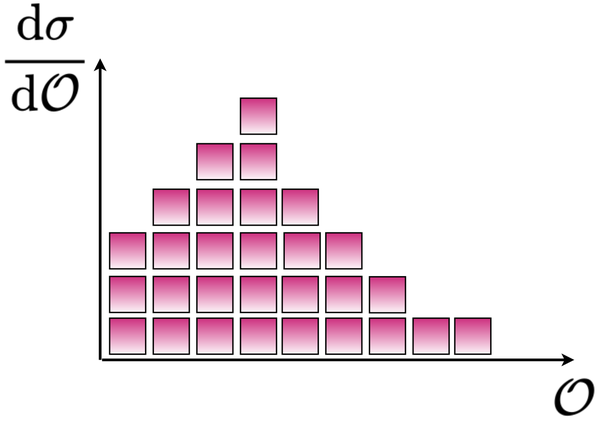
Integration vs event generation

MC integrator



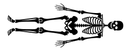
Accept/reject step

Event generator

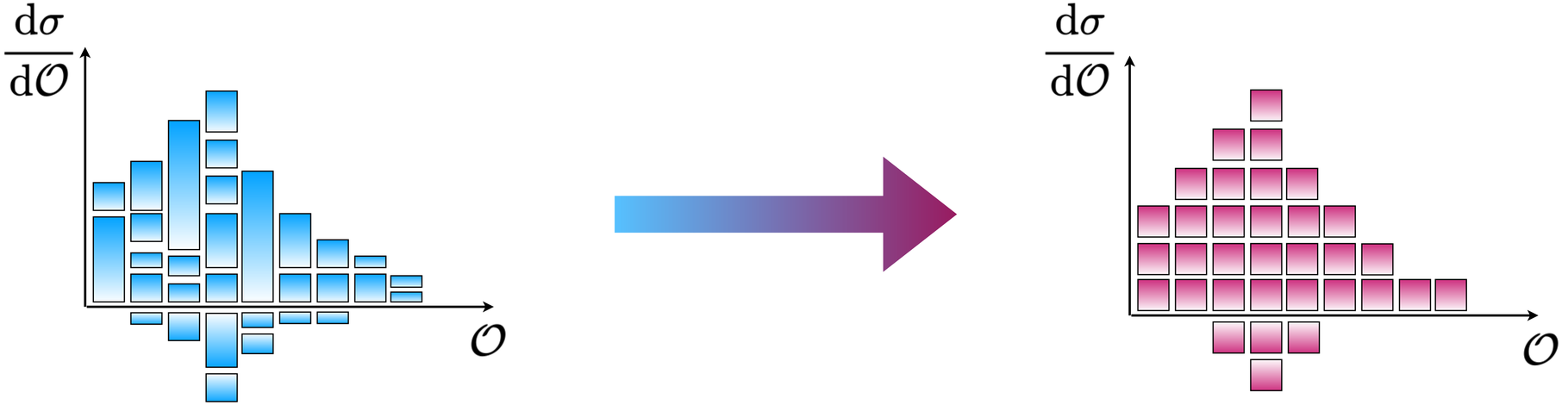


Important: integration weights must be ***bounded*** from above!

Skeletons in the closet

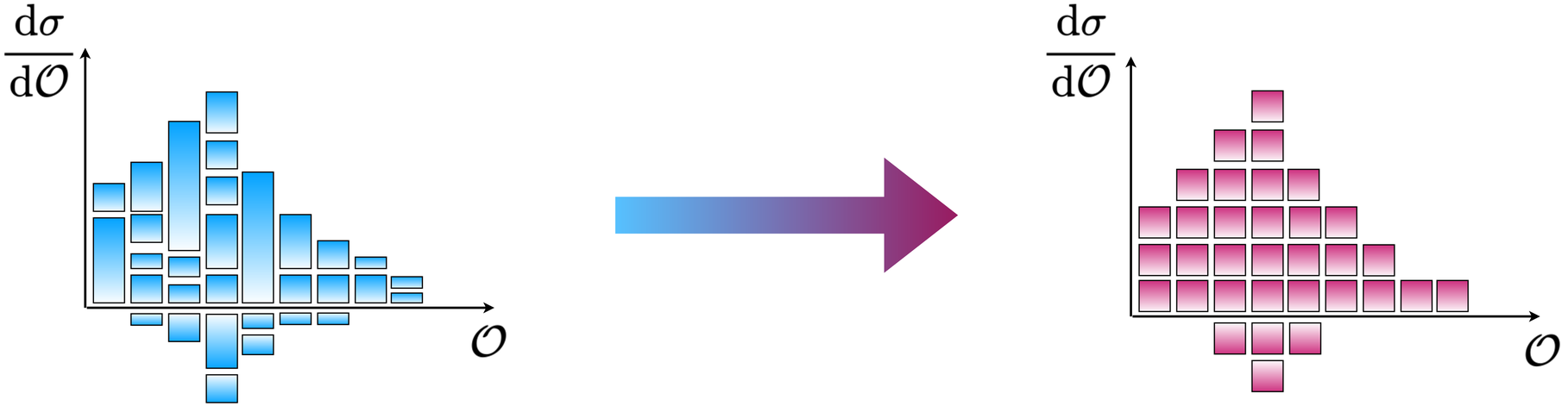


In some cases, we have negative weights
→ especially from subtraction at NLO (see next lecture)
→ have to keep them during unweighting!



Skeletons in the closet

- In some cases, we have negative weights
  → especially from subtraction at NLO (see next lecture)
  → have to keep them during unweighting!



- Often have a few events with large weights
  → can make unweighting extremely inefficient
  → solution: accept fraction of “overweight events”, for example 0.1%

Summary

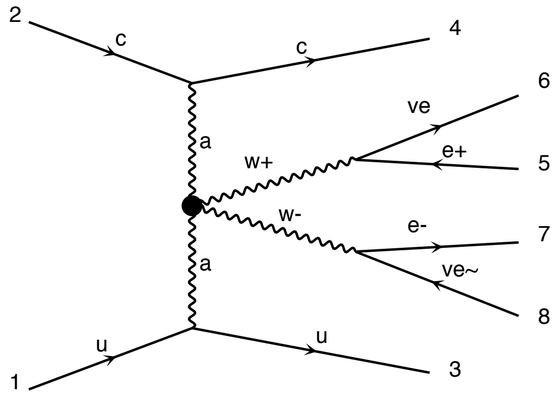
- Generate ***unweighted events***
  → compressed dataset
  → save time on following steps of simulation chain
- Accept events below $w_{\max}$ with probability $w/w_{\max}$
  → all events have weight $w_{\max}$ afterwards
- ***better proposal distribution*** leads to ***better unweighting efficiency***
  → reduces expensive matrix element evaluations

Outline

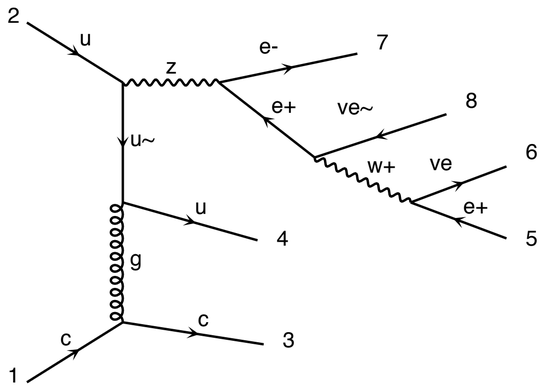
1. LHC basics
2. Matrix elements
3. Monte Carlo integration
4. Phase-space sampling
5. Event generation
6. **Decays**
7. Machine learning
8. Parton showers
9. Multijet merging

Decays

Resonant diagram



Non-resonant diagram



Problem:

Process too complicated too include all diagrams

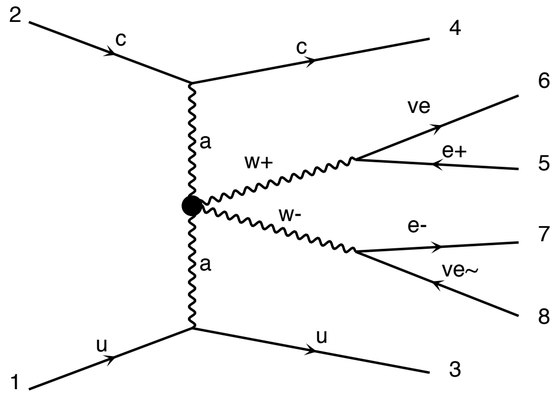
Observation:

Many contributions are off-shell

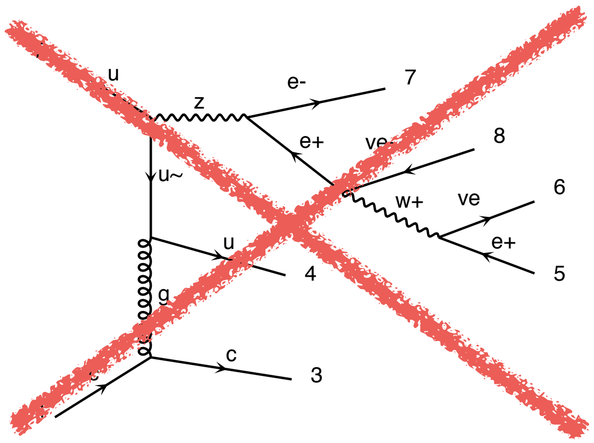
→ small impact on total cross section

Decays

Resonant diagram



Non-resonant diagram



Problem:

Process too complicated too include all diagrams

Observation:

Many contributions are off-shell

→ small impact on total cross section

Solution:

Only keep on-shell contributions

Narrow-width approximation

$$\frac{1}{(q^2 - M^2)^2 + M^2\Gamma^2} \xrightarrow{\Gamma/M \ll 1} \frac{\pi}{M\Gamma}\,\delta(q^2 - M^2)$$

$$\sigma(AB \to R \to f) \simeq \sigma(AB \to R)\,\mathrm{BR}(R \to f)\left[1 + \mathcal{O}\left(\frac{\Gamma}{M}\right)\right]$$

- Valid for narrow resonances: $\Gamma \ll M$
- The resonance ***virtuality is fixed*** by $\delta(q^2 - M^2)$
- Production and decay ***factorize***.
- Neglects off-shell effects, non-resonant diagrams, and interference
- Corrections are typically of relative size Γ/M, but can be enhanced near cuts, thresholds, or resonance tails.

Usage in MadGraph

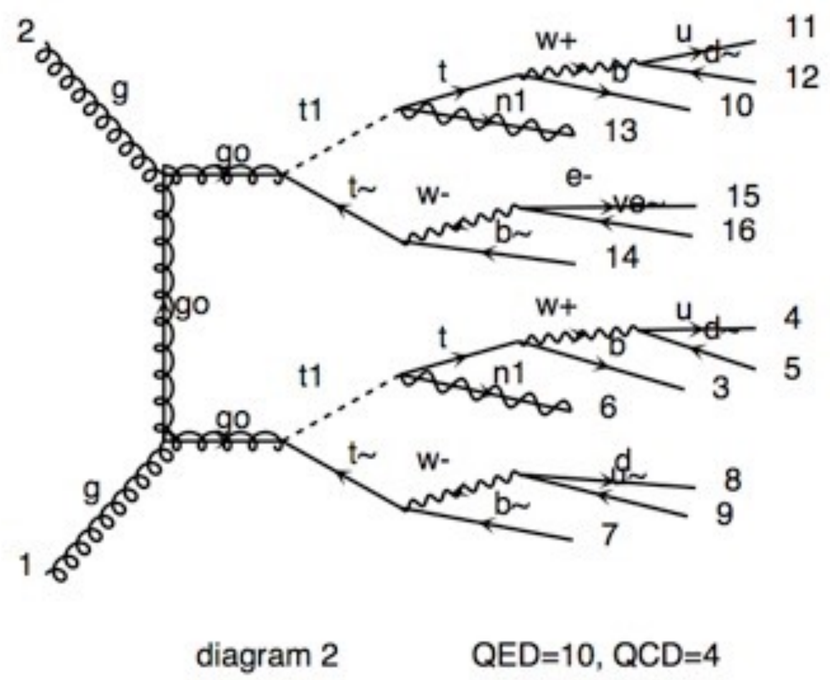


diagram 2 QED=10, QCD=4

- very long decay chains possible to simulate directly in MadGraph
- syntax:

```
p p > t t~ w+, (t > w+ b, w+ > l+ vl), \
              (t~ > w- b~, w- > j j), \
              w+ > l+ vl
```

  (invariant mass cut associated to t, t~, W)
- other tools exist for NWA (like MadSpin)

Summary

- Full matrix element becomes too expensive for large number of final state particles
- Dominant contribution from ***on-shell propagators***
  → go to ***narrow-width approximation***
- makes very long decay chains possible directly in MadGraph
- Decay width is ***not a free parameter***
  → critical in narrow-width approximation