MadGraph 5

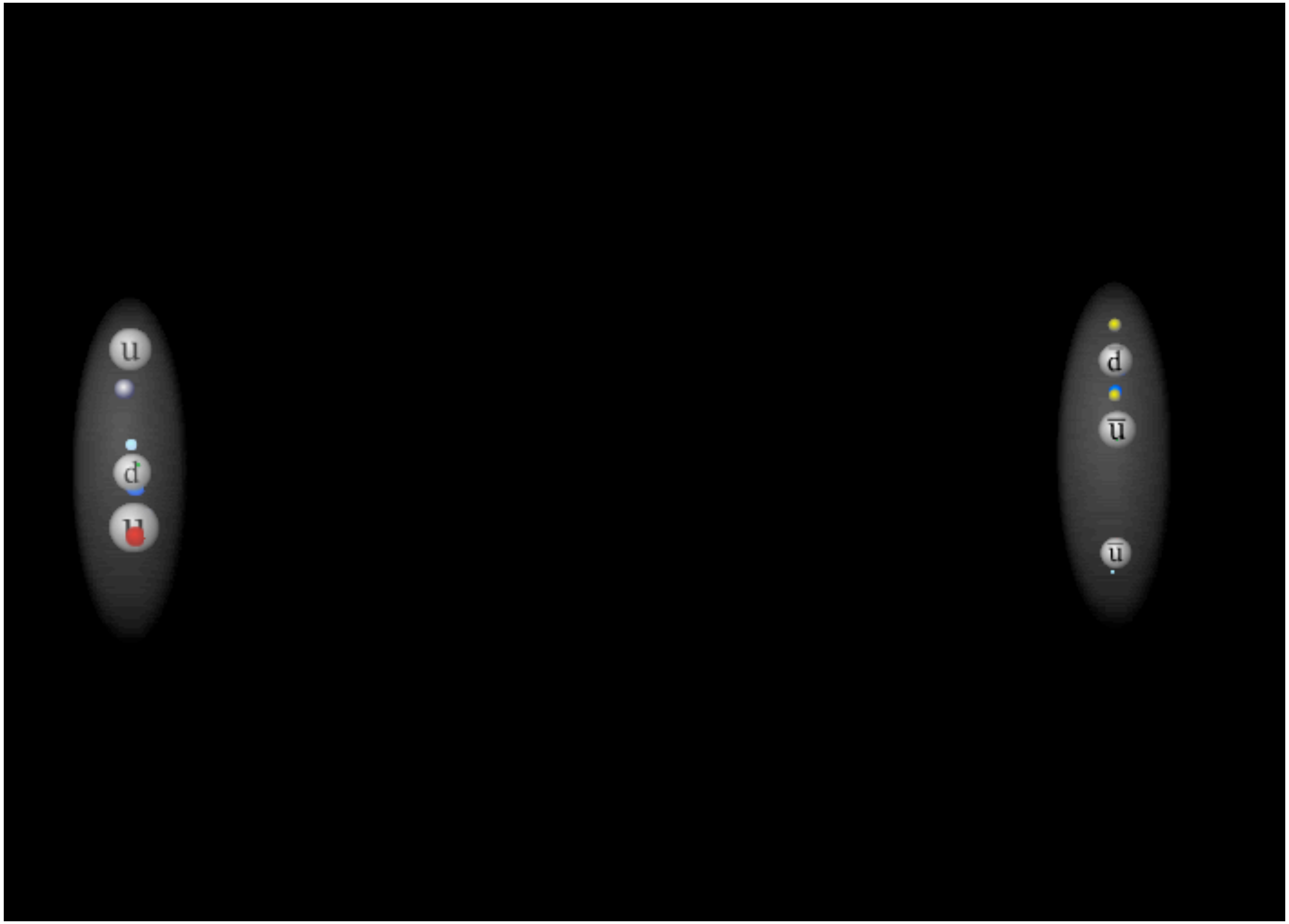
Olivier Mattelaer
University of Illinois at Urbana Champaign

Special thanks to Fabio Maltoni / Johan Alwall from whom I have shamelessly borrowed many of the following slides

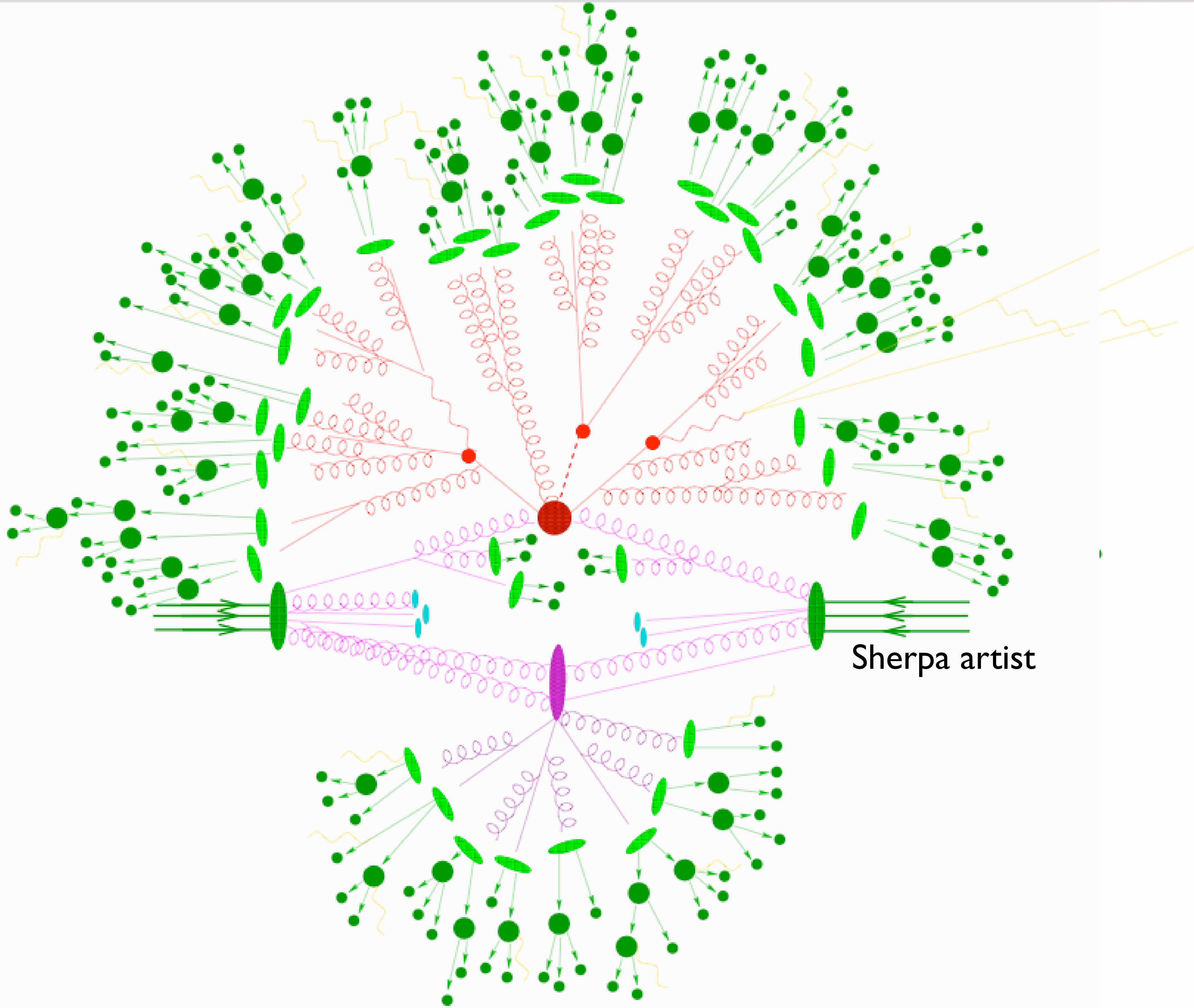
Aims for these lectures

- Get you acquainted with the concepts and techniques used in event generation
- Give you hands-on experience with matrix element generation, event generation and analysis
- Answer as many of your questions as I can (so please ask questions!)

- PROTON COLLISIONS



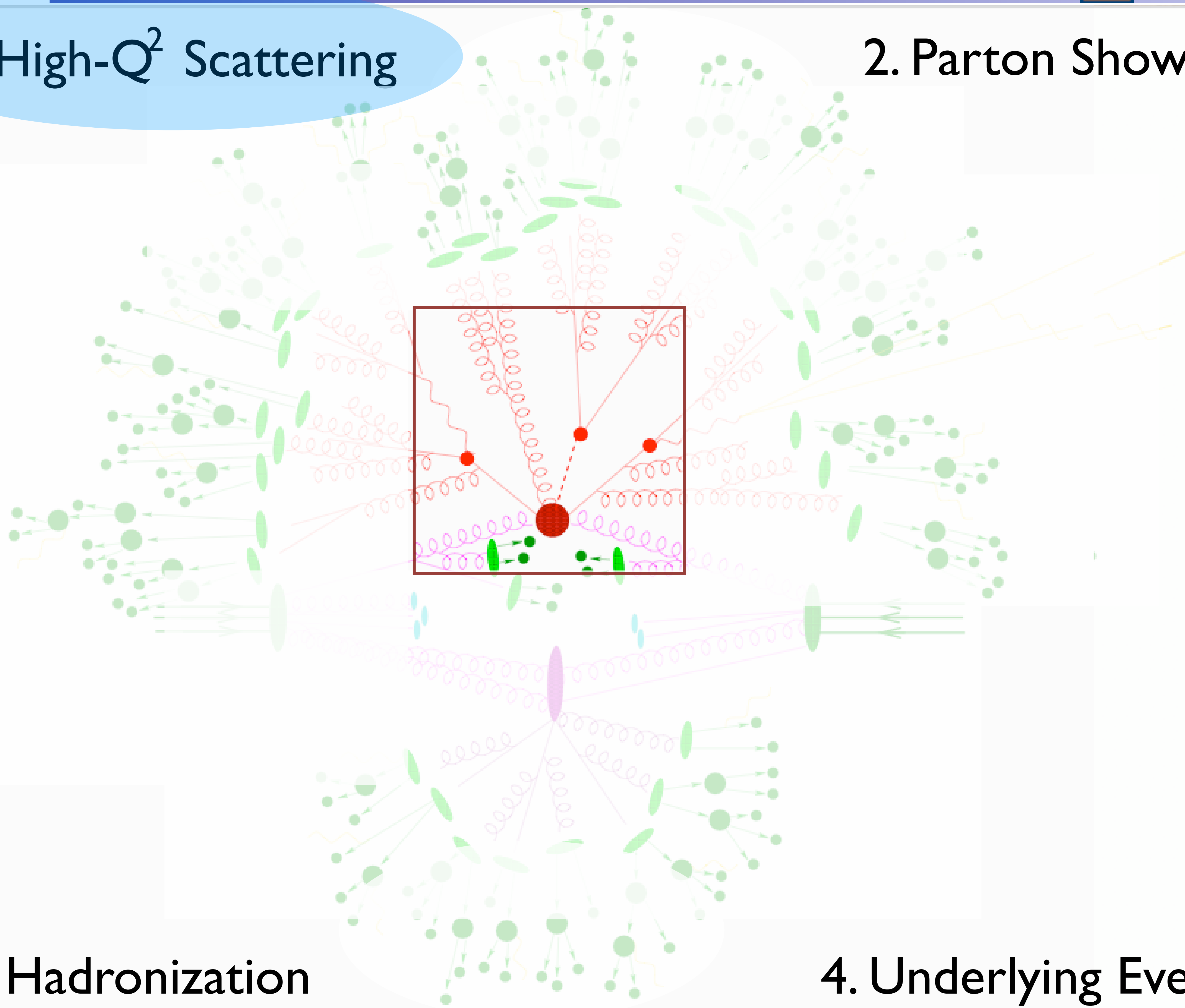
M. scott



Sherpa artist

1. High-Q^2 Scattering

2. Parton Shower



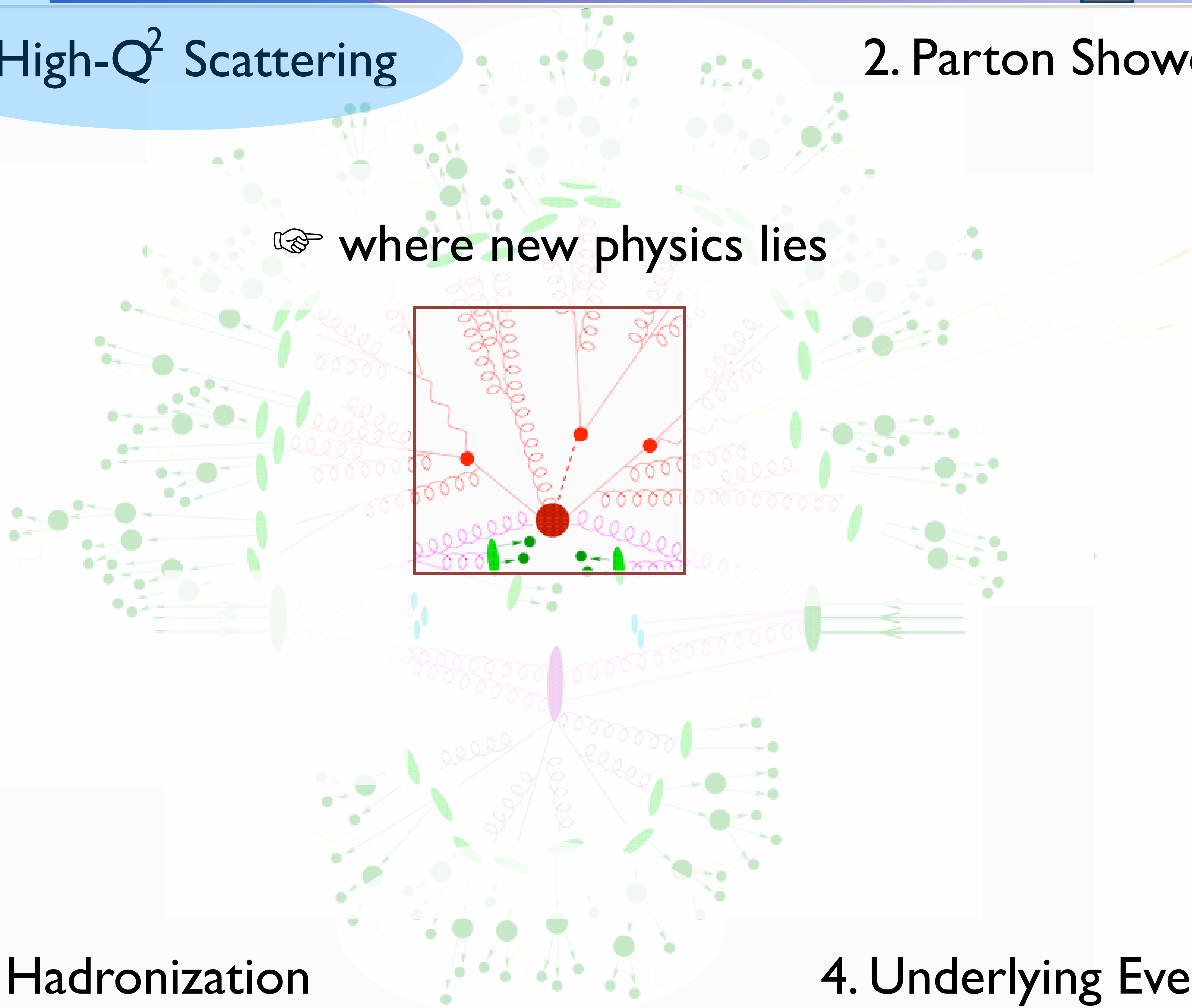
3. Hadronization

4. Underlying Event

1. High-Q^2 Scattering

2. Parton Shower

☞ where new physics lies



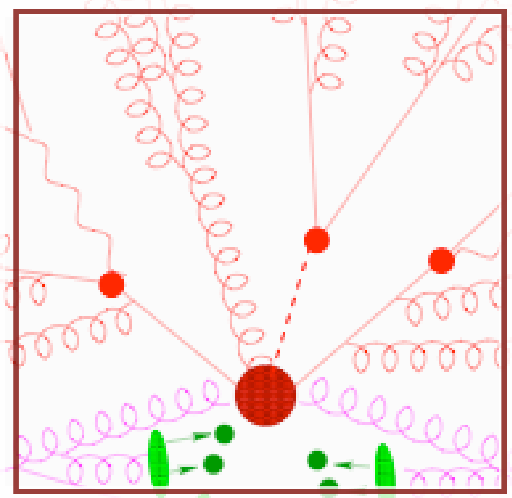
3. Hadronization

4. Underlying Event

1. High-Q^2 Scattering

2. Parton Shower

☞ where new physics lies



☞ process dependent

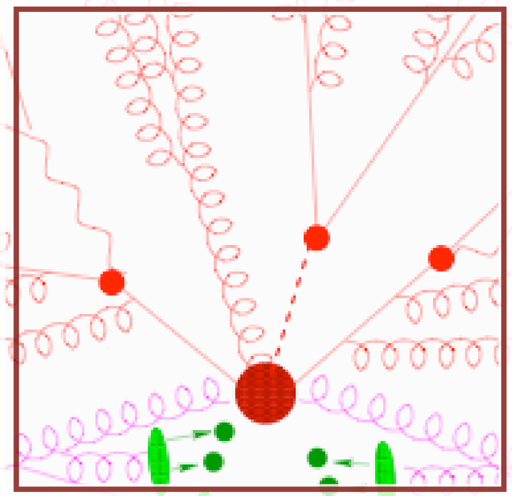
3. Hadronization

4. Underlying Event

1. High-Q^2 Scattering

2. Parton Shower

- ☞ where new physics lies



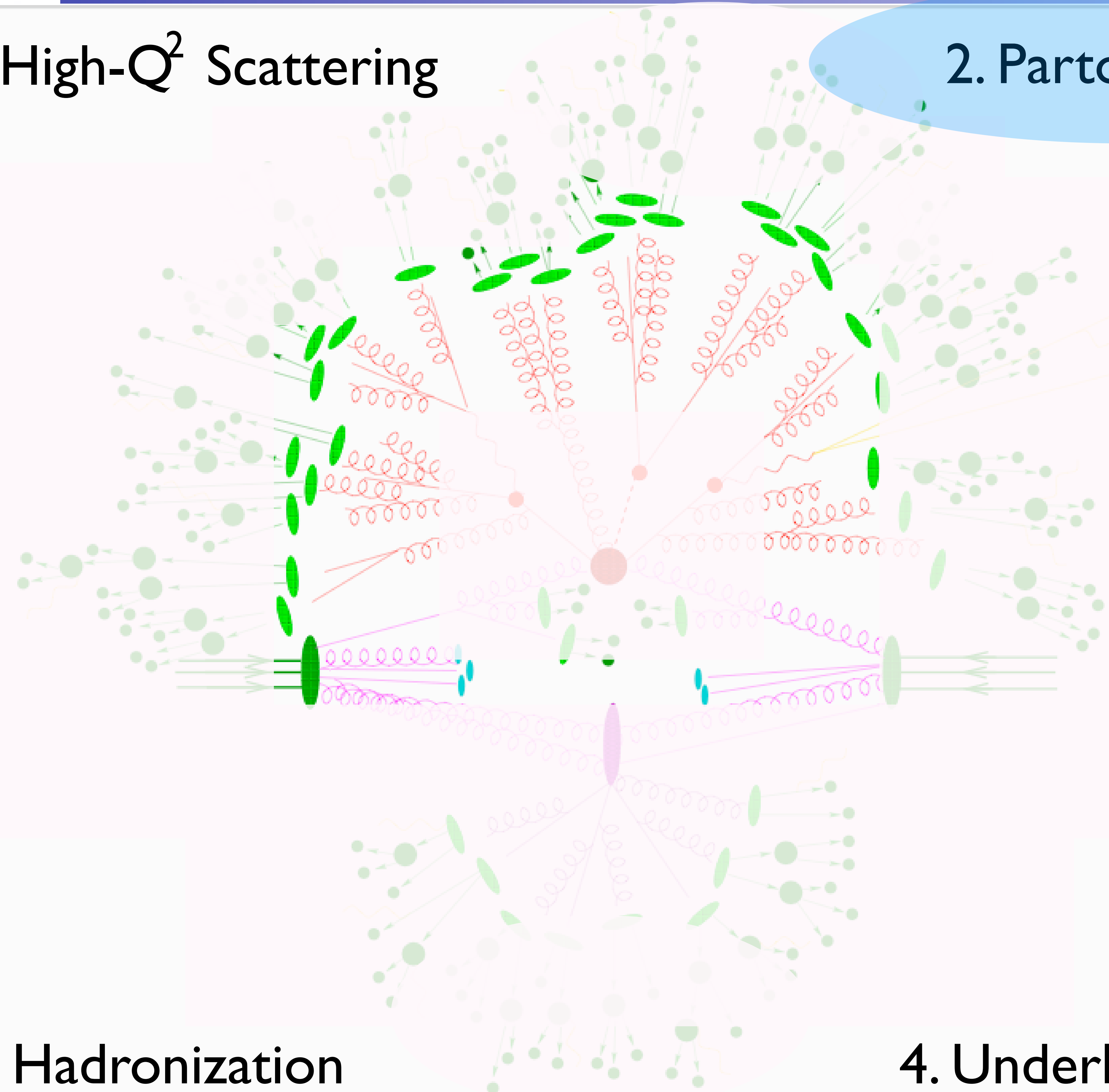
- ☞ process dependent
- ☞ first principles description

3. Hadronization

4. Underlying Event

1. High-Q^2 Scattering

2. Parton Shower

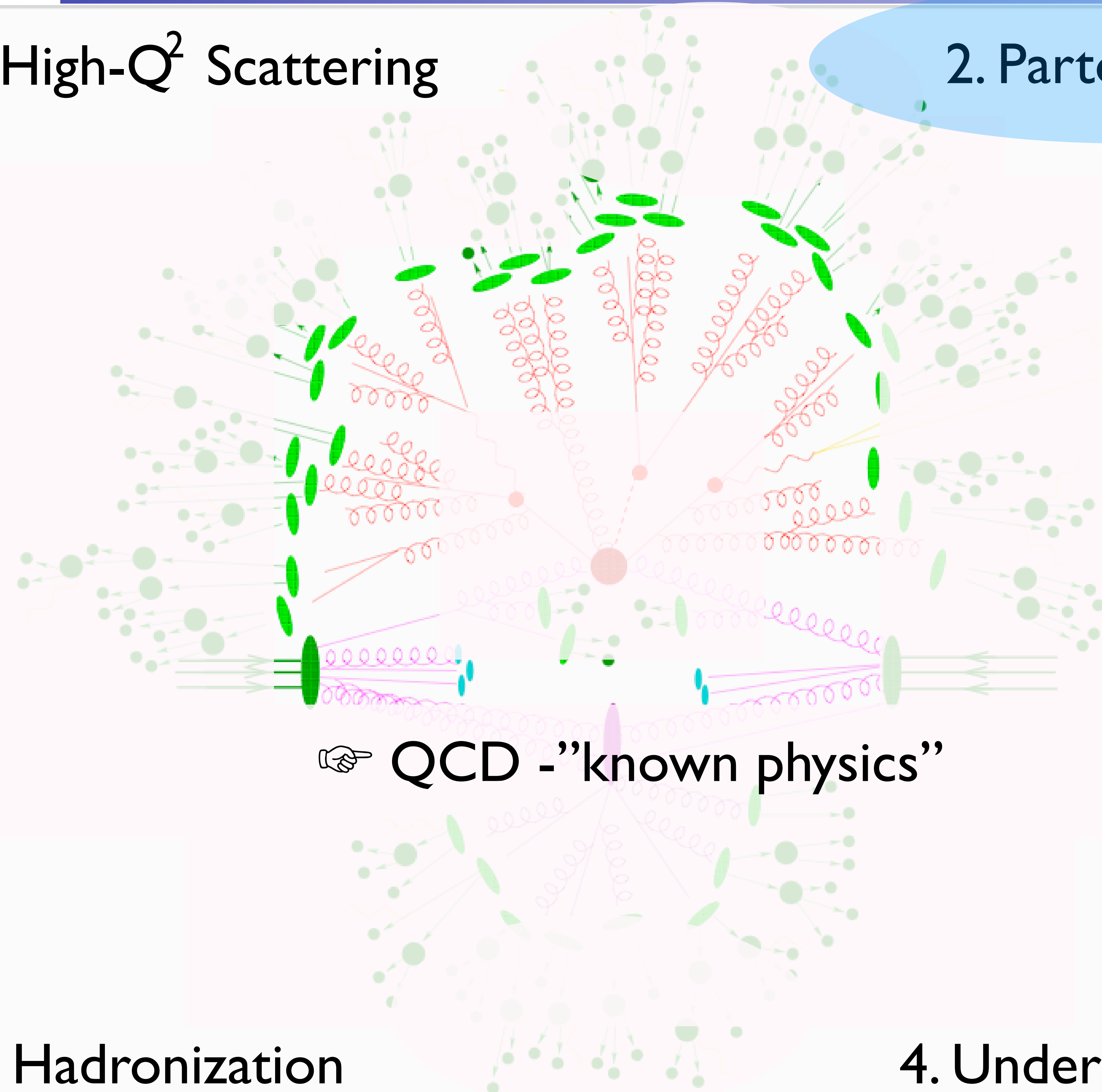


3. Hadronization

4. Underlying Event

1. High-Q^2 Scattering

2. Parton Shower



☞ QCD -"known physics"

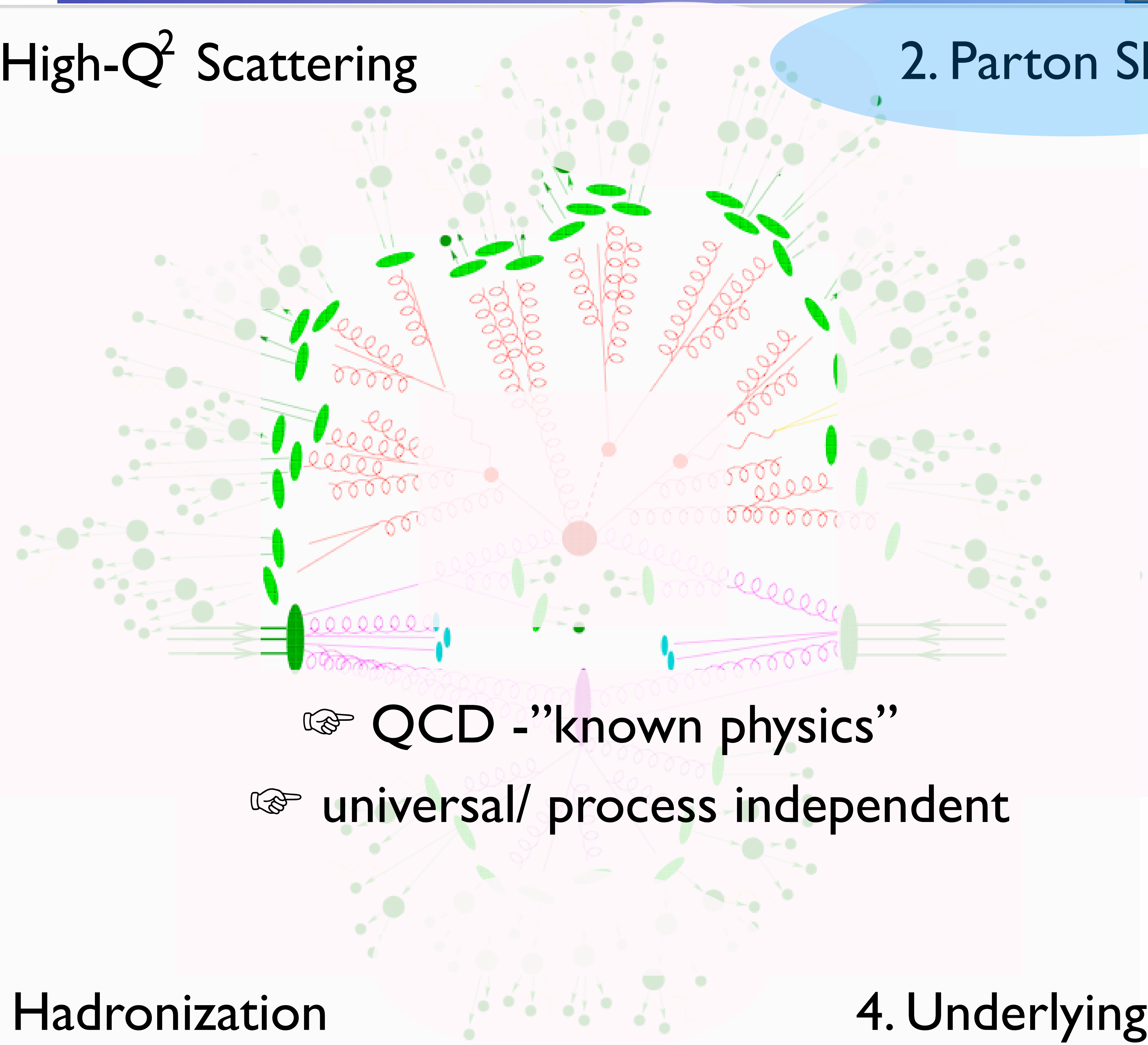
3. Hadronization

4. Underlying Event

Wednesday, October 3, 12

1. High-Q^2 Scattering

2. Parton Shower



☞ QCD -"known physics"

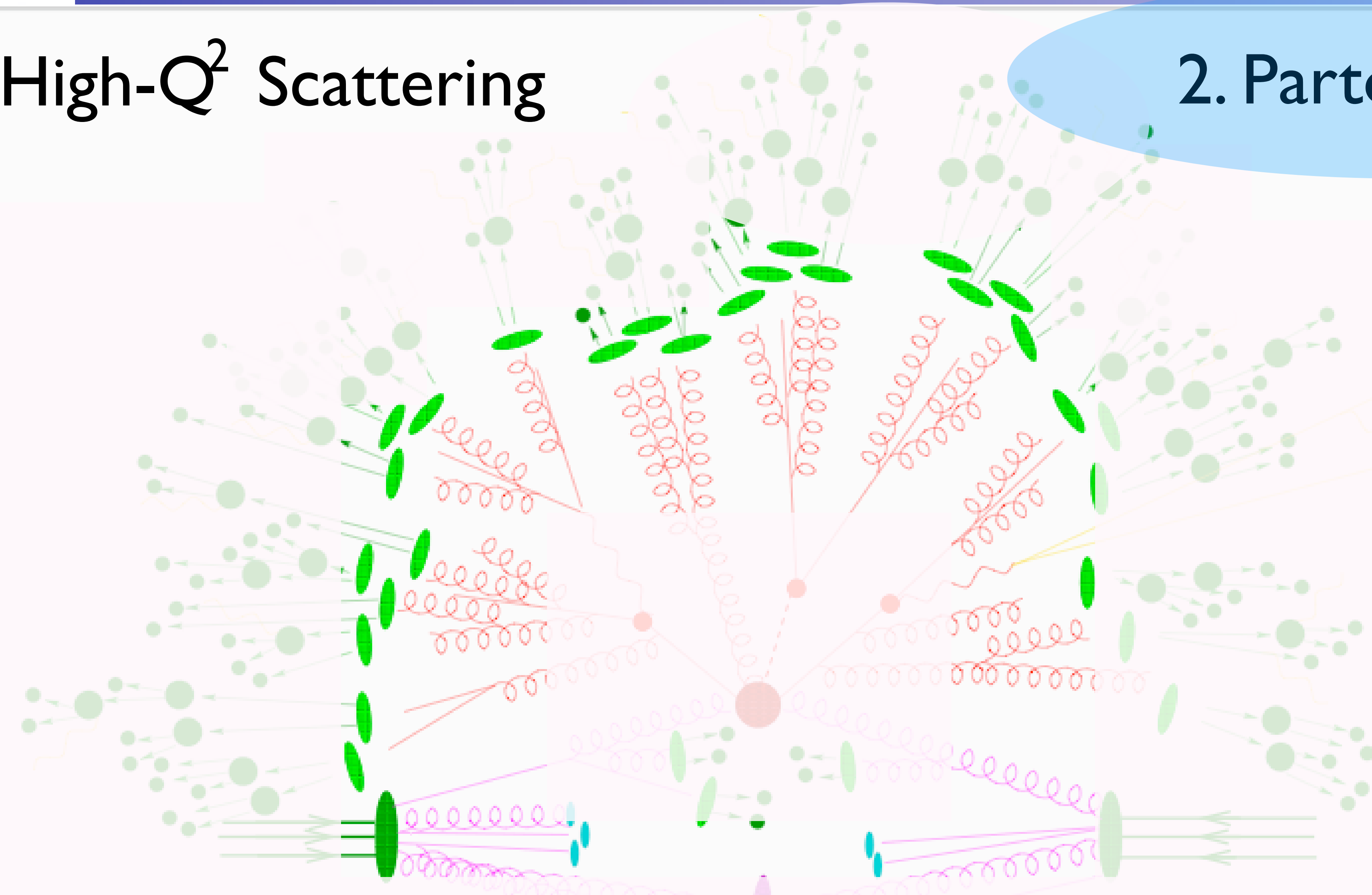
☞ universal/ process independent

3. Hadronization

4. Underlying Event

1. High-Q^2 Scattering

2. Parton Shower



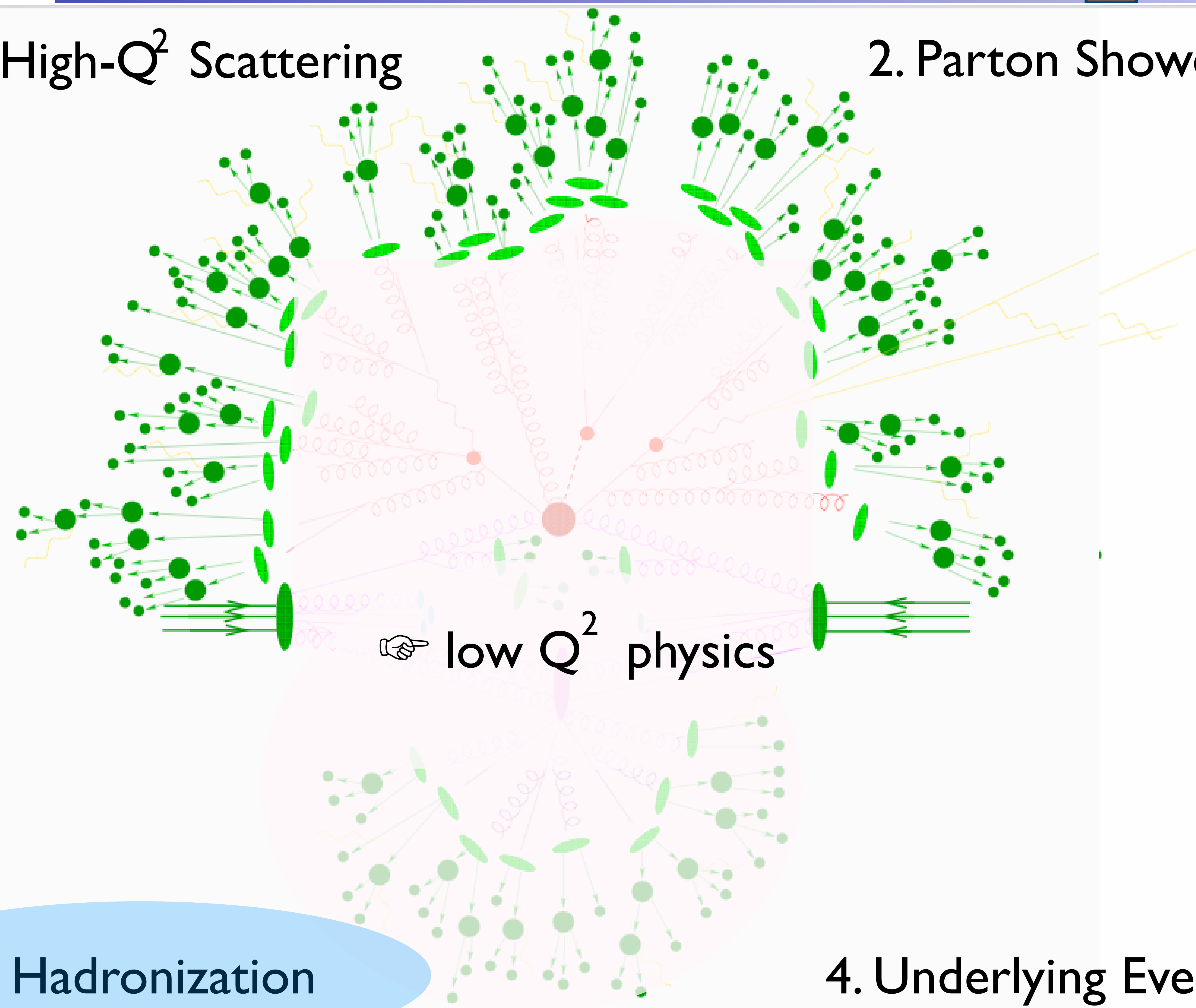
- ☞ QCD -"known physics"
- ☞ universal/ process independent
- ☞ first principles description

3. Hadronization

4. Underlying Event

1. High-Q^2 Scattering

2. Parton Shower



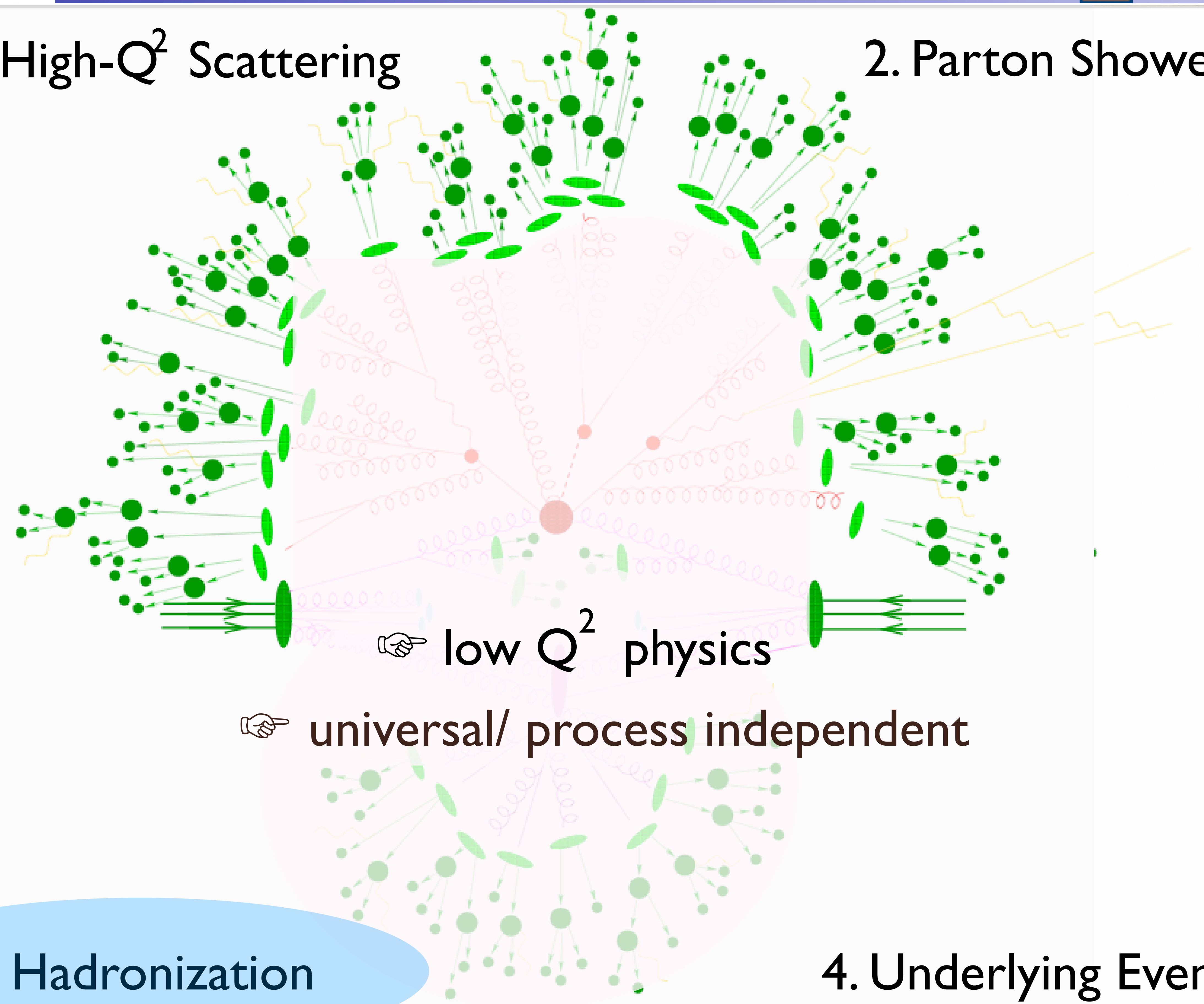
☞ low Q^2 physics

3. Hadronization

4. Underlying Event

1. High-Q^2 Scattering

2. Parton Shower



☞ low Q^2 physics

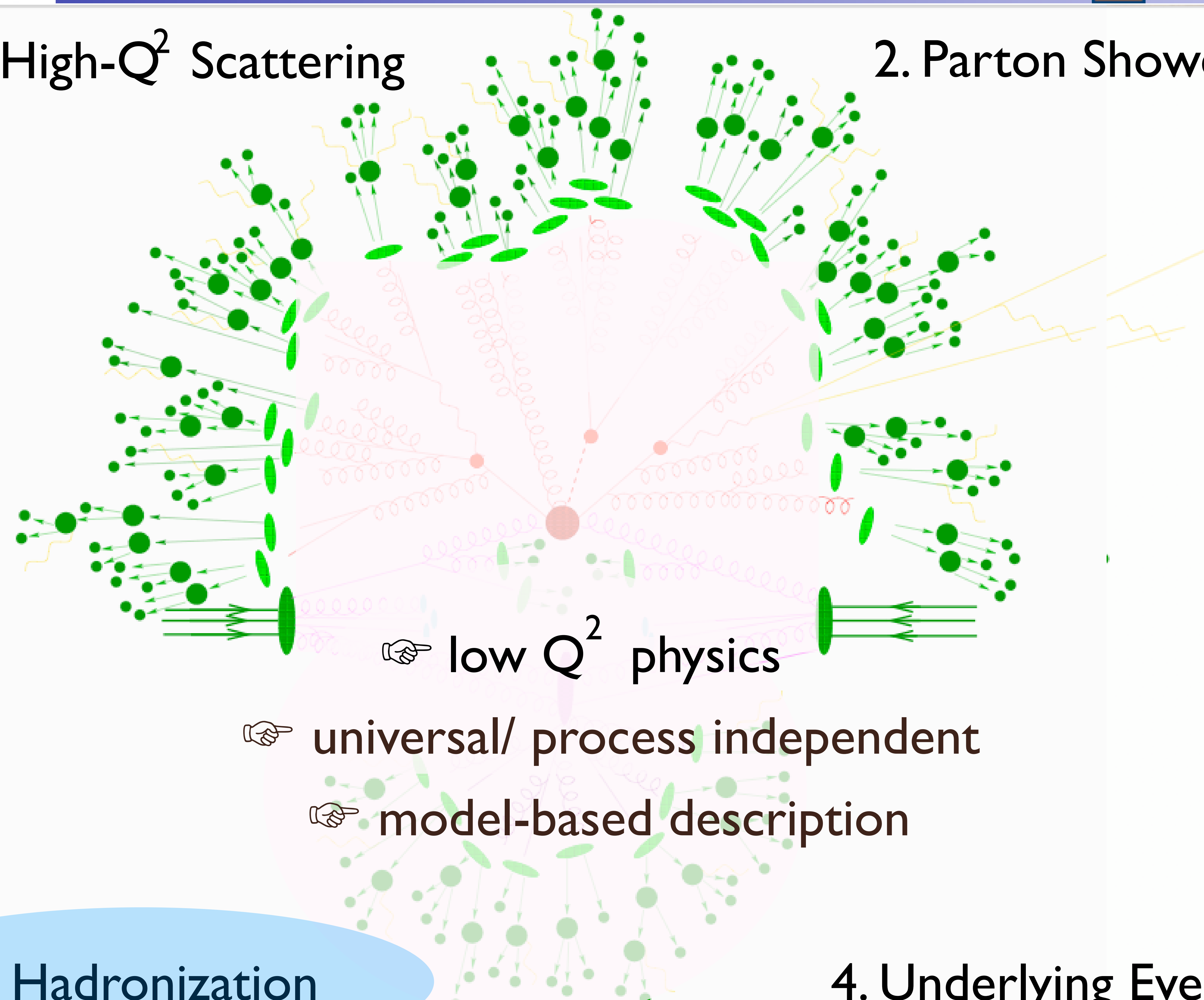
☞ universal/ process independent

3. Hadronization

4. Underlying Event

1. High-Q^2 Scattering

2. Parton Shower



☞ low Q^2 physics

☞ universal/ process independent

☞ model-based description

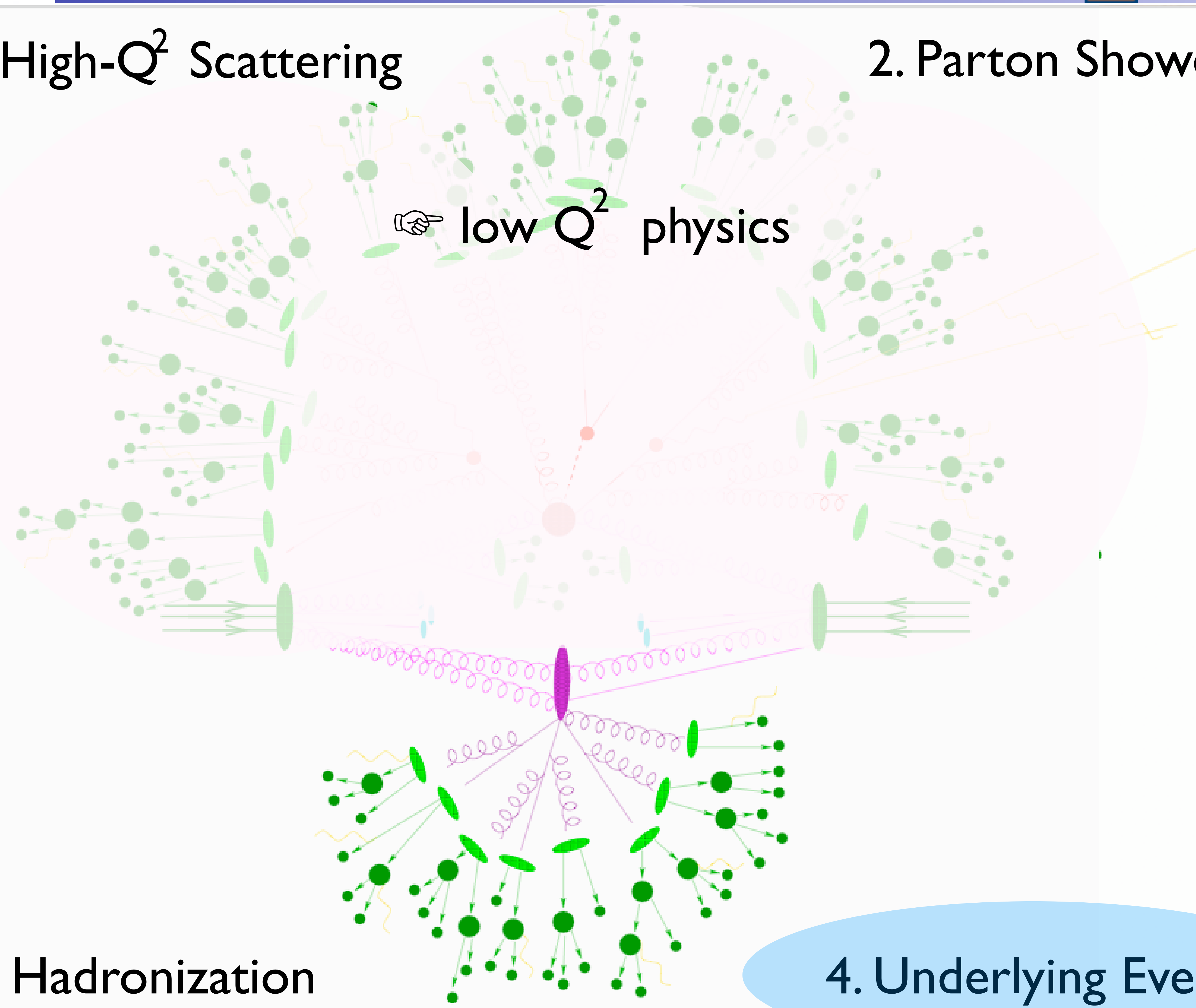
3. Hadronization

4. Underlying Event

1. High-Q^2 Scattering

2. Parton Shower

☞ low Q^2 physics



3. Hadronization

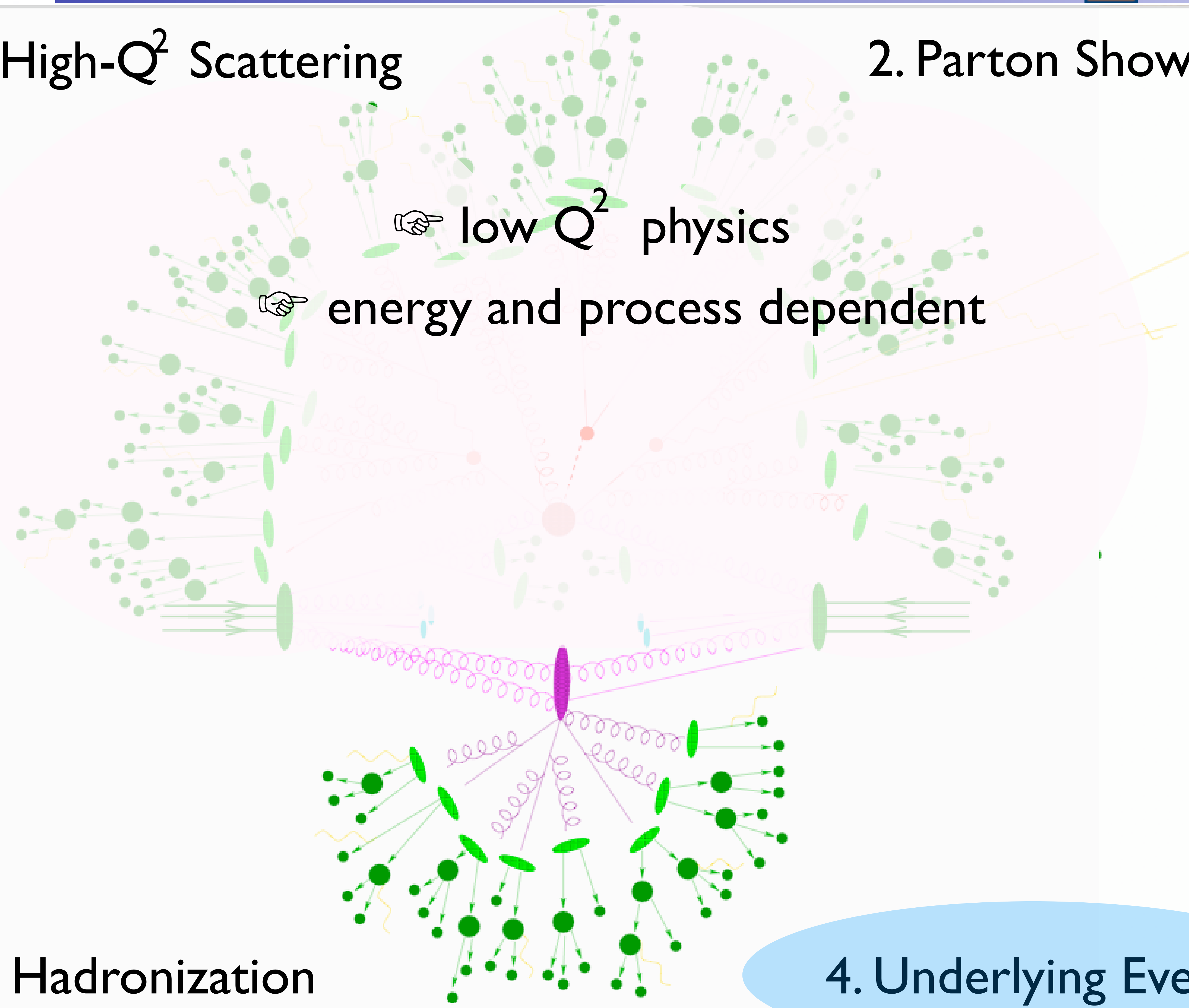
4. Underlying Event

1. High-Q^2 Scattering

2. Parton Shower

☞ low Q^2 physics

☞ energy and process dependent

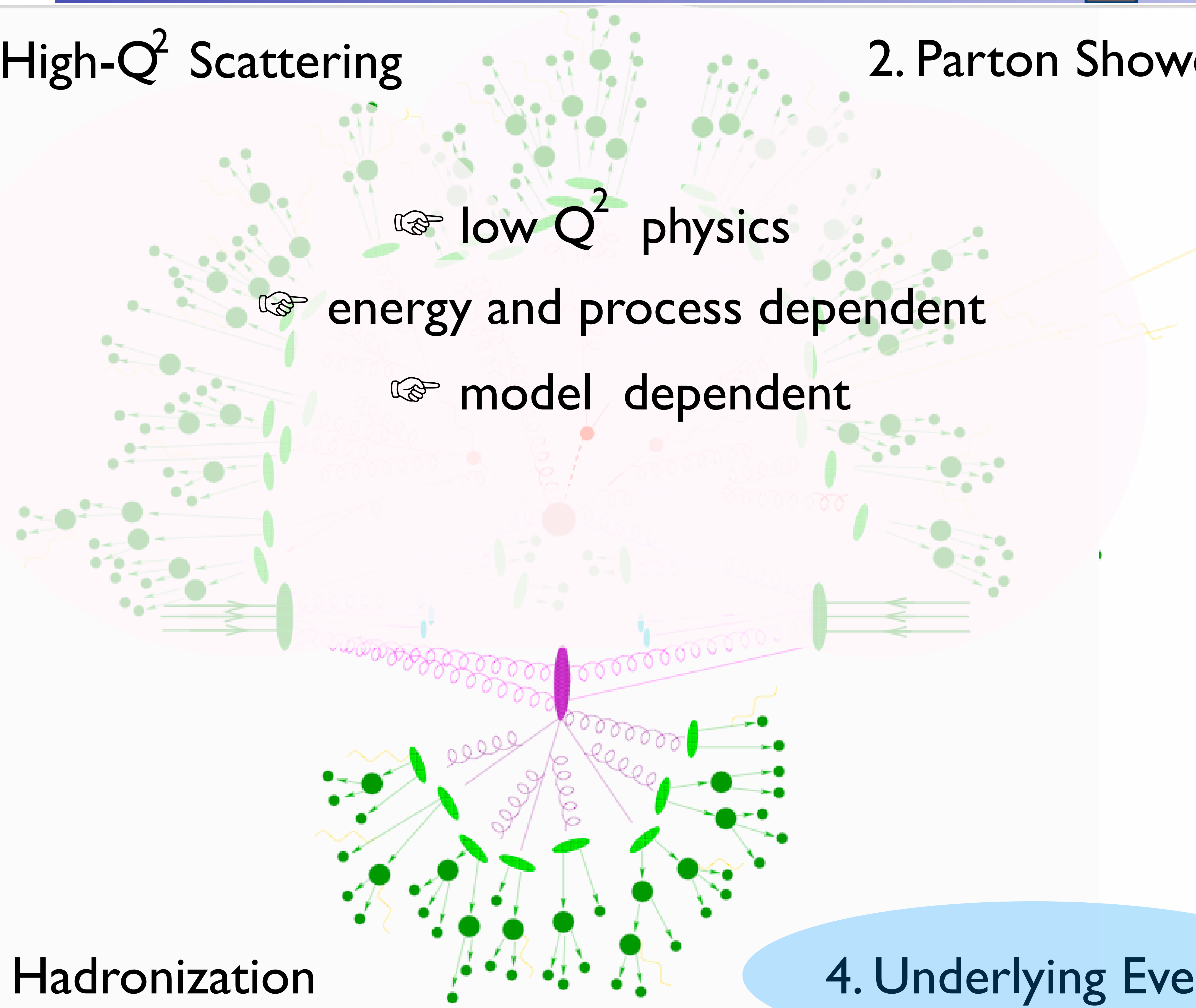


3. Hadronization

4. Underlying Event

1. High-Q^2 Scattering

2. Parton Shower



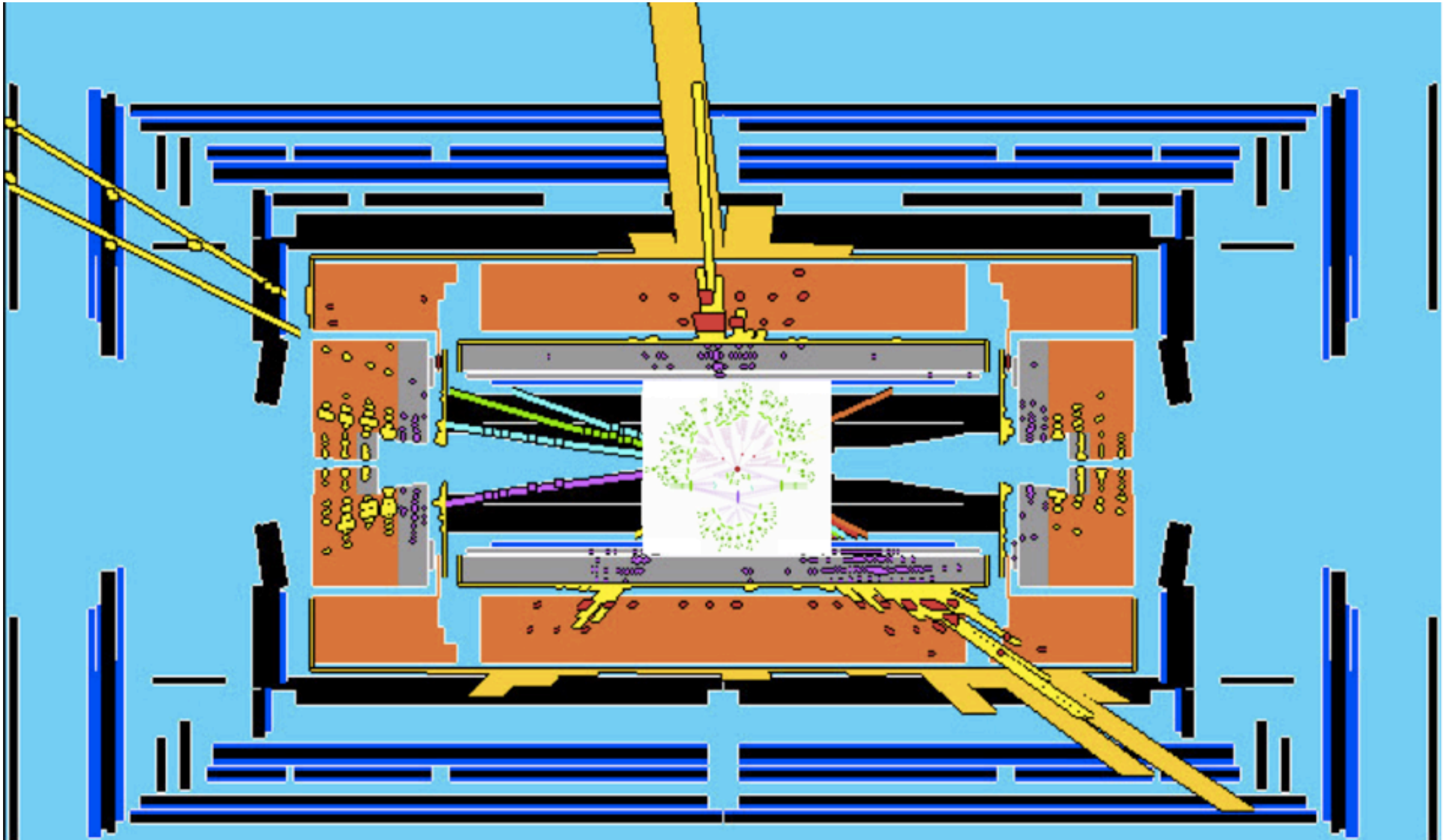
☞ low Q^2 physics

☞ energy and process dependent

☞ model dependent

3. Hadronization

4. Underlying Event



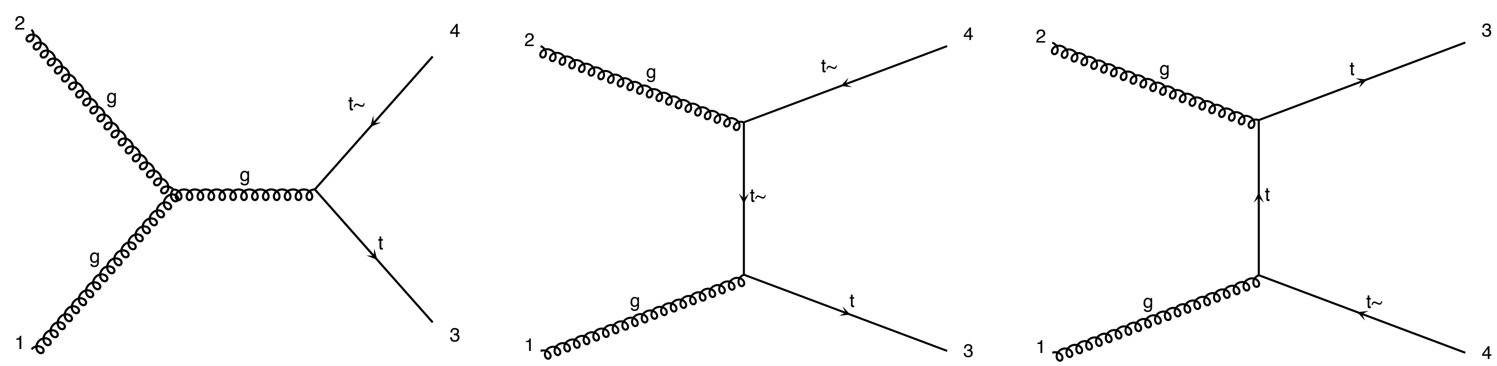
5. Detector simulation

Matrix Element calculation at Hadron Colliders

To calculate a given process (e.g., $p\,p \rightarrow t\,\bar{t}$)

1. Determine contributing subprocesses
   $g\,g \rightarrow t\,\bar{t}$, $q\,\bar{q} \rightarrow t\,\bar{t}$, $\bar{q}\,q \rightarrow t\,\bar{t}$ with $\overset{(-)}{q} = \overset{(-)}{d}, \overset{(-)}{u}, \overset{(-)}{s}, \overset{(-)}{c}, (\overset{(-)}{b})$

2. Determine matrix element for each subprocess



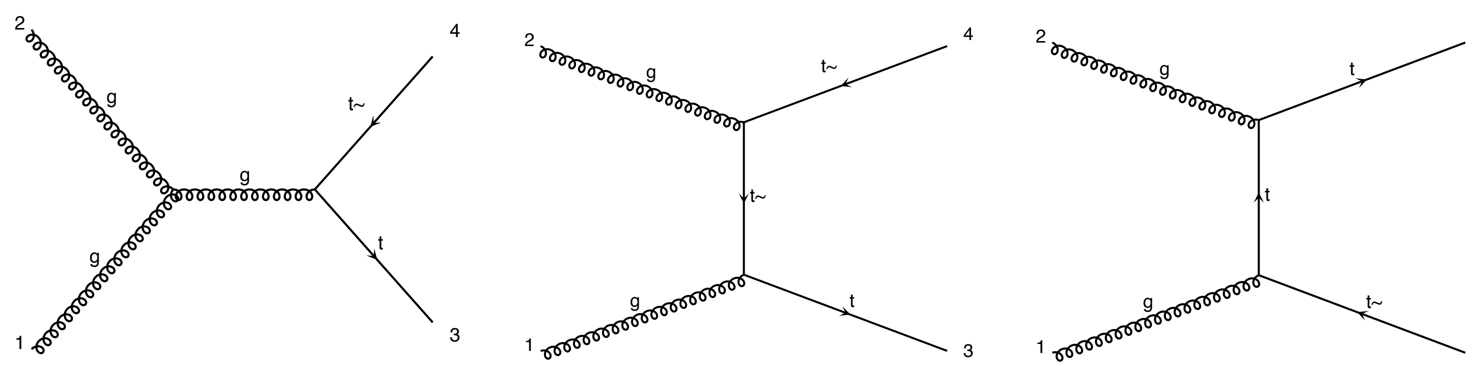
3. Perform phase space integration for each subprocess

$$\sigma = \frac{1}{2s} \int |\mathcal{M}|^2 d\Phi(n)$$

Matrix Element calculation at Hadron Colliders

To calculate a given process (e.g., $p\,p \to t\,\bar{t}$)

1. Determine contributing subprocesses
   $g\,g \to t\,\bar{t}$, $q\,\bar{q} \to t\,\bar{t}$, $\bar{q}\,q \to t\,\bar{t}$ with $\overset{(-)}{q} = \overset{(-)}{d}, \overset{(-)}{u}, \overset{(-)}{s}, \overset{(-)}{c}, (\overset{(-)}{b})$ ⇦ Easy enough
2. Determine matrix element for each subprocess ⇦ Hard



3. Perform phase space integration for each subprocess

$$\sigma = \frac{1}{2s} \int |\mathcal{M}|^2 d\Phi(n)$$

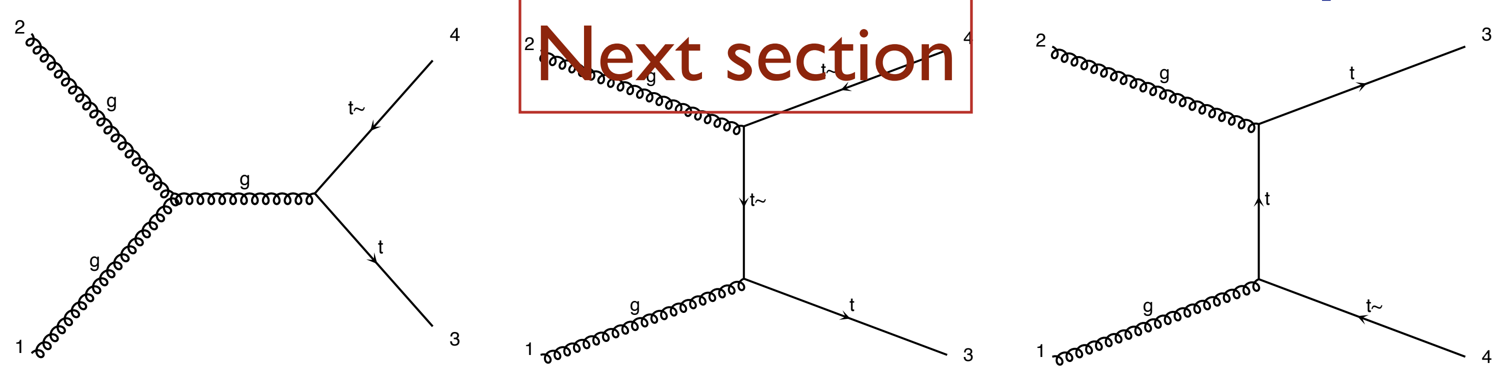
⇦ Very Hard (in general)

Matrix Element calculation at Hadron Colliders

To calculate a given process (e.g., $p\,p \rightarrow t\,\bar{t}$)

1. Determine contributing subprocesses
   $g\,g \rightarrow t\,\bar{t}$, $q\,\bar{q} \rightarrow t\,\bar{t}$, $\bar{q}\,q \rightarrow t\,\bar{t}$ with $\overset{(-)}{q} = \overset{(-)}{d}, \overset{(-)}{u}, \overset{(-)}{s}, \overset{(-)}{c}, (\overset{(-)}{b})$ ⇐ Easy enough
2. Determine matrix element for each subprocess ⇐ Hard

Next section



3. Perform phase space integration for each subprocess

$$\sigma = \frac{1}{2s} \int |\mathcal{M}|^2 d\Phi(n)$$

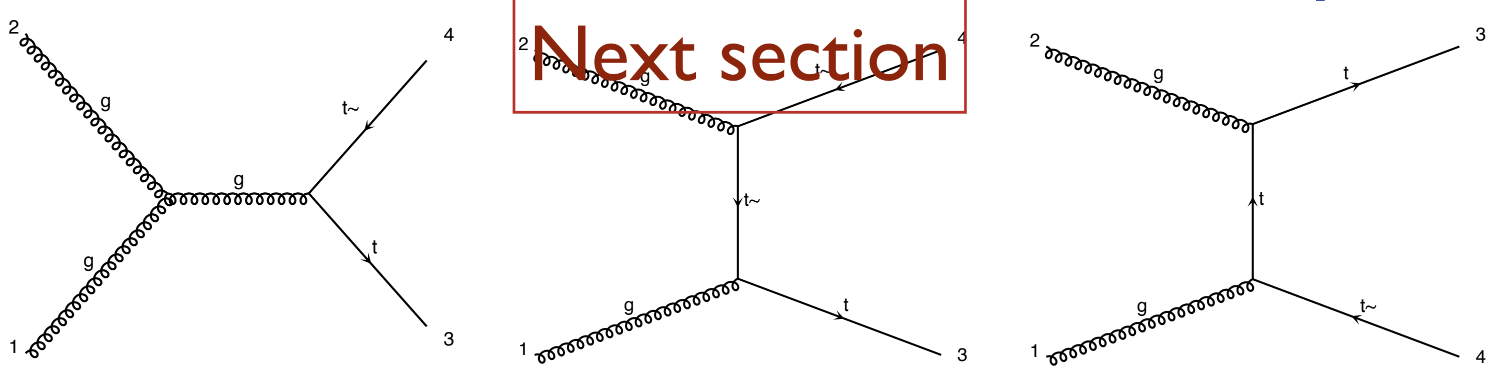
⇐ Very Hard (in general)

Matrix Element calculation at Hadron Colliders

To calculate a given process (e.g., $p\,p \rightarrow t\,\bar{t}$)

1. Determine contributing subprocesses
   $g\,g \rightarrow t\,\bar{t}$, $q\,\bar{q} \rightarrow t\,\bar{t}$, $\bar{q}\,q \rightarrow t\,\bar{t}$ with $\overset{(-)}{q} = \overset{(-)}{d}, \overset{(-)}{u}, \overset{(-)}{s}, \overset{(-)}{c}, (\overset{(-)}{b})$ ⇐ Easy enough
2. Determine matrix element for each subprocess ⇐ Hard

Next section

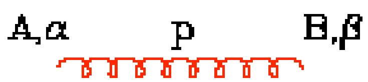


3. Perform phase space integration for each subprocess

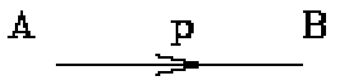
$$\sigma = \frac{1}{2s}\int |\mathcal{M}|^2 d\Phi(n)$$

Not This times ⇐ Very Hard (in general)

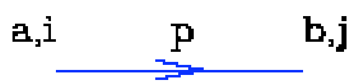
The Matrix Element



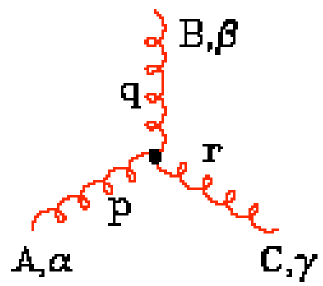
$$\delta^{AB}\left[-g^{\alpha\beta}+(1-\lambda)\frac{p^\alpha p^\beta}{p^2+i\epsilon}\right]\frac{i}{p^2+i\epsilon}$$



$$\delta^{AB}\frac{i}{(p^2+i\epsilon)}$$

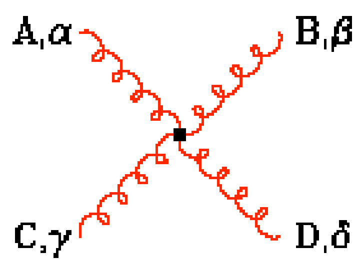


$$\delta^{ab}\frac{i}{(\not{p}-m+i\epsilon)_{ji}}$$

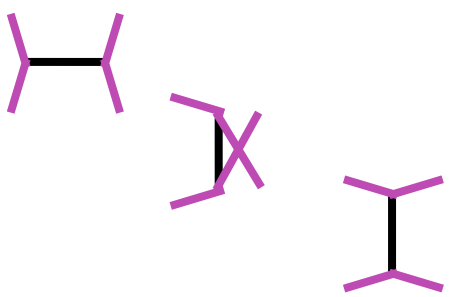


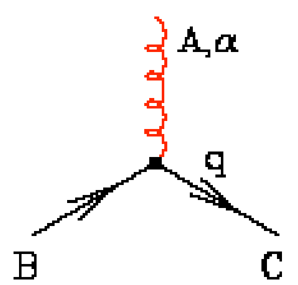
$$-g\ f^{ABC}[(p-q)^\gamma g^{\alpha\beta}+(q-r)^\alpha g^{\beta\gamma}+(r-p)^\beta g^{\gamma\alpha}]$$

(all momenta incoming)

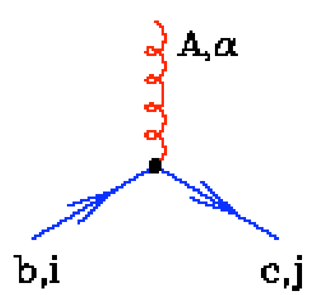


$$-ig^2\ f^{XAC}f^{XBD}\ [g^{\alpha\beta}g^{\gamma\delta}-g^{\alpha\delta}g^{\beta\gamma}]$$
$$-ig^2\ f^{XAD}f^{XBC}\ [g^{\alpha\beta}g^{\gamma\delta}-g^{\alpha\gamma}g^{\beta\delta}]$$
$$-ig^2\ f^{XAB}f^{XCD}\ [g^{\alpha\gamma}g^{\beta\delta}-g^{\alpha\delta}g^{\beta\gamma}]$$



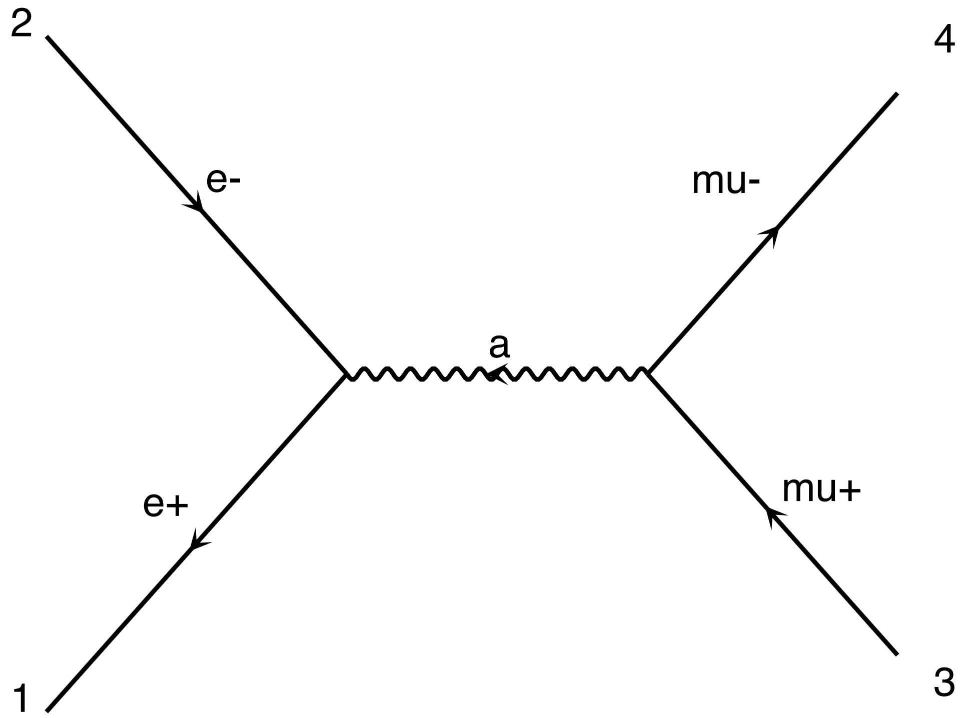


$$g\ f^{ABC}q^\alpha$$

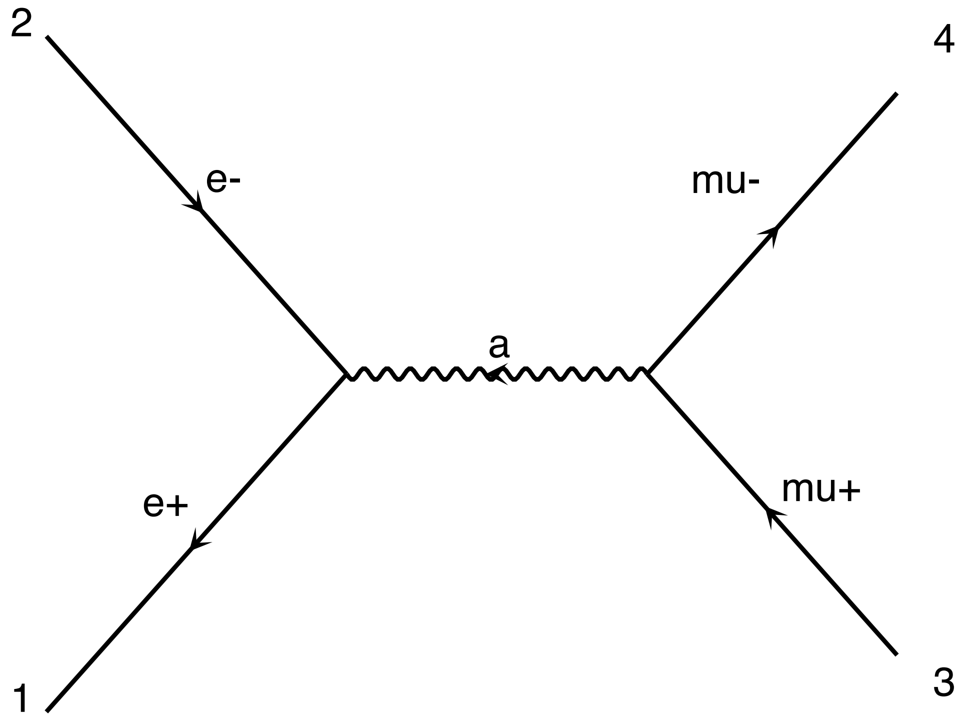


$$-ig\ (t^A)_{cb}\ (\gamma^\alpha)_{ji}$$

Evaluate a square Matrix Element

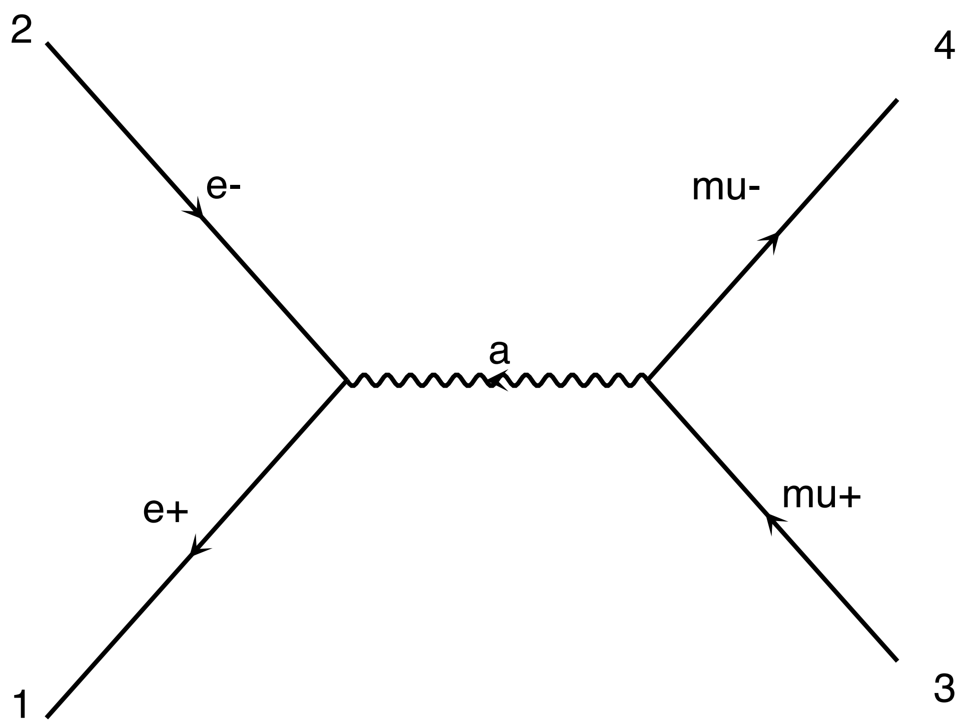


Evaluate a square Matrix Element



$$\mathcal{M} = e^2 (\bar{u}\gamma^\mu v) \frac{g_{\mu\nu}}{q^2} (\bar{u}\gamma^\nu v)$$

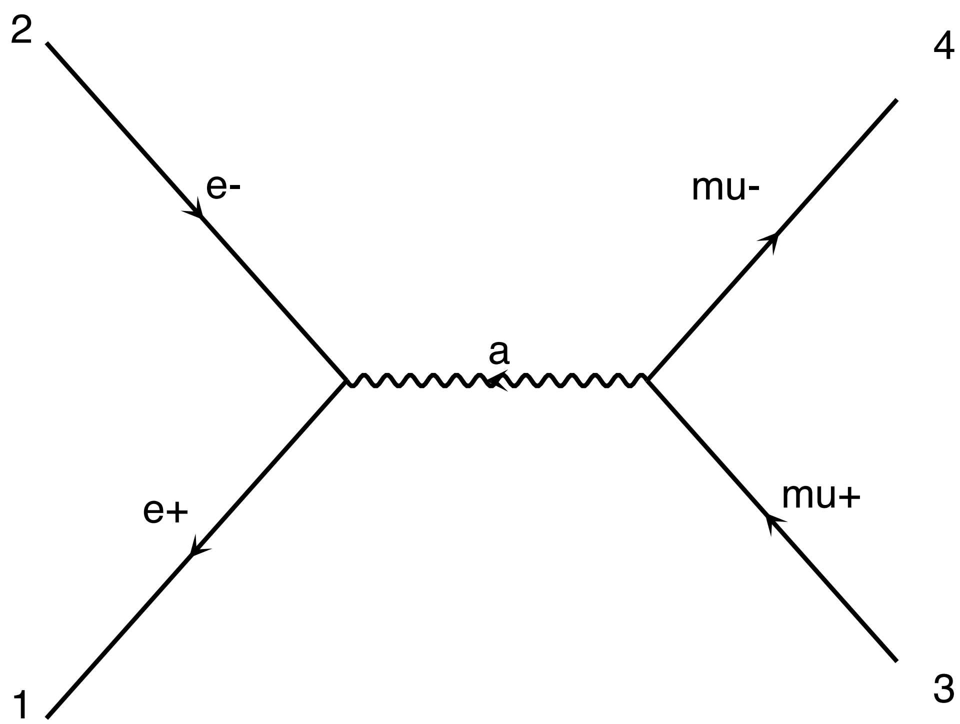
Evaluate a square Matrix Element



$$\mathcal{M} = e^2 (\bar{u}\gamma^\mu v) \frac{g_{\mu\nu}}{q^2} (\bar{u}\gamma^\nu v)$$

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

Evaluate a square Matrix Element

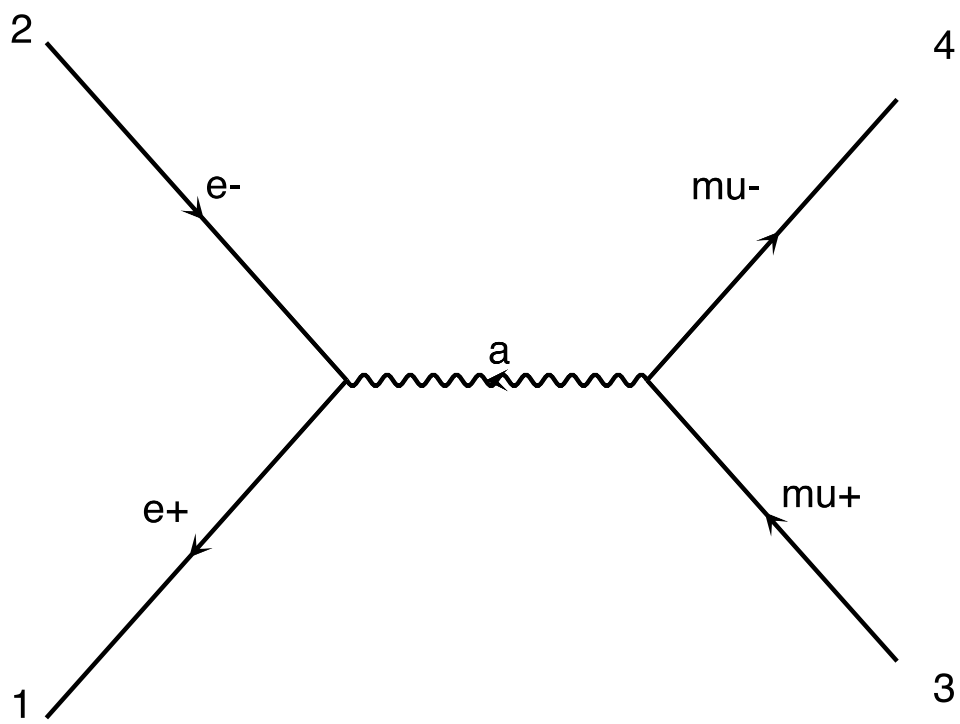


$$\mathcal{M} = e^2(\bar{u}\gamma^\mu v)\,\frac{g_{\mu\nu}}{q^2}\,(\bar{u}\gamma^\nu v)$$

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

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

Evaluate a square Matrix Element



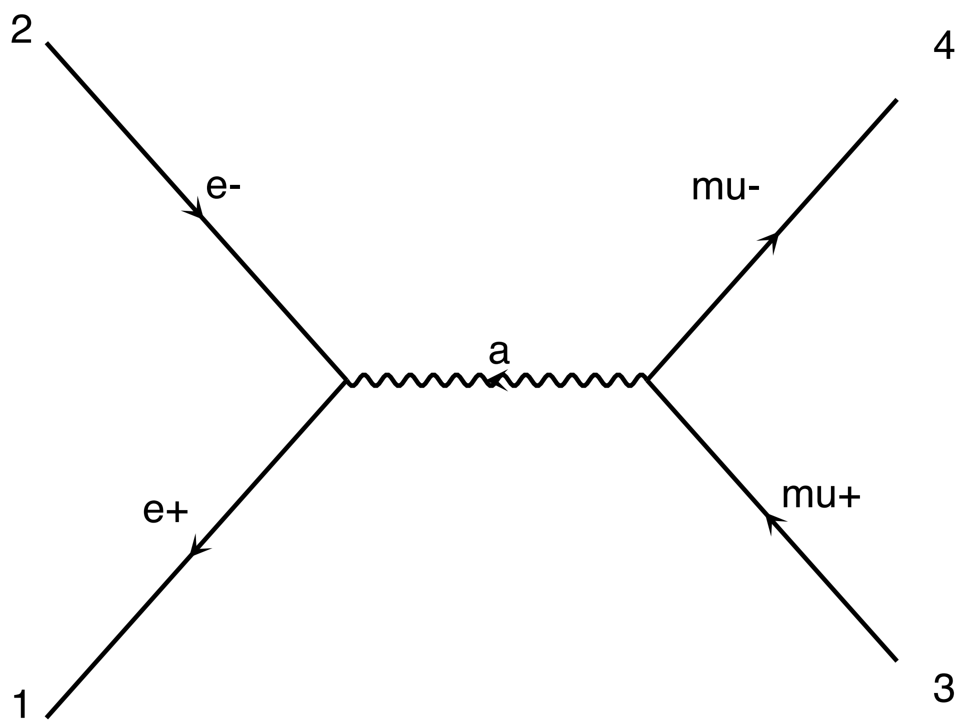
$$\mathcal{M} = e^2(\bar{u}\gamma^\mu v)\,\frac{g_{\mu\nu}}{q^2}\,(\bar{u}\gamma^\nu v)$$

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

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

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

Evaluate a square Matrix Element



$$\mathcal{M} = e^2 (\bar{u}\gamma^\mu v) \frac{g_{\mu\nu}}{q^2} (\bar{u}\gamma^\nu v)$$

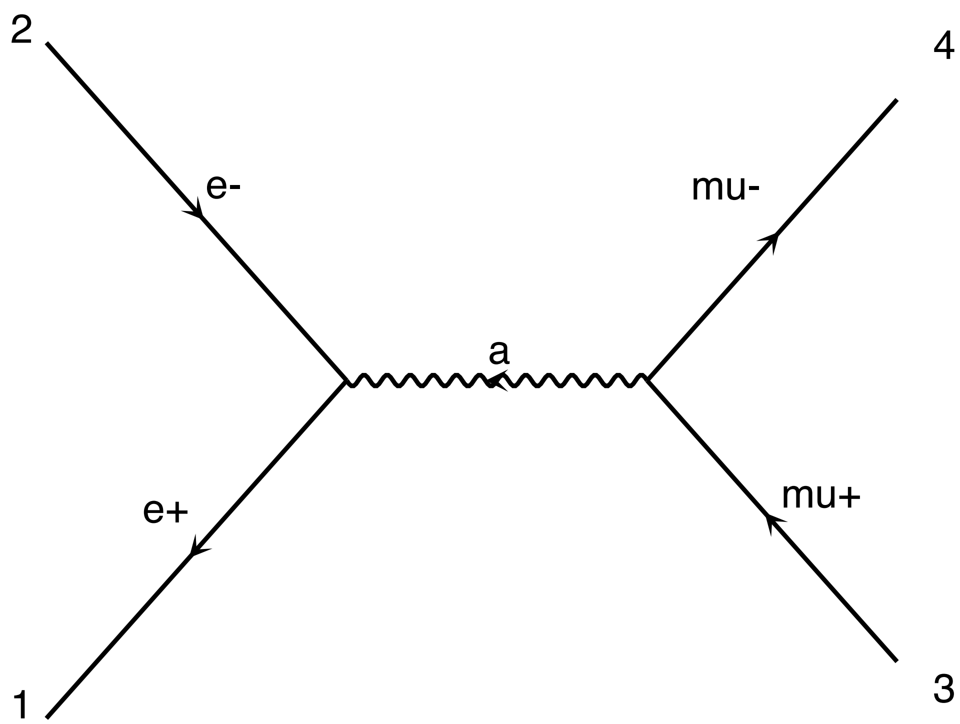
$$\frac{1}{4}\sum_{pol} |\mathcal{M}|^2 = \frac{1}{4}\sum_{pol} \mathcal{M}^* \mathcal{M}$$

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

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

Very Efficient !!!

Evaluate a square Matrix Element



$$\mathcal{M} = e^2(\bar{u}\gamma^\mu v)\,\frac{g_{\mu\nu}}{q^2}\,(\bar{u}\gamma^\nu v)$$

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

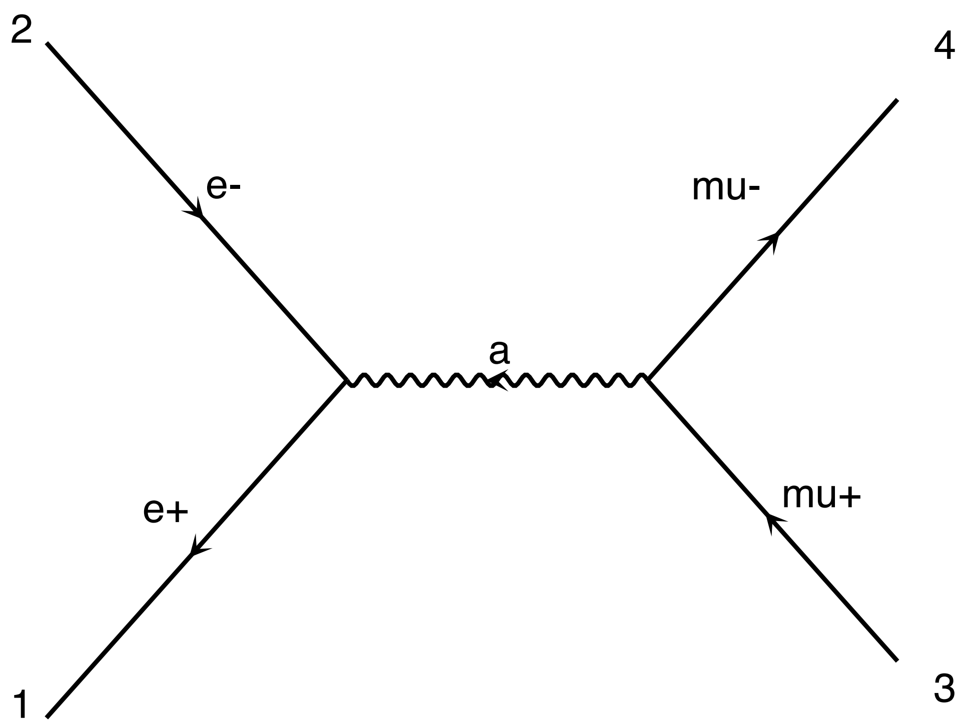
$$\sum_{pol}\bar{u}u = \not{p} + m$$

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

Very Efficient !!!

But The number of term raises as N^2

Evaluate a square Matrix Element



$$\mathcal{M} = e^2(\bar{u}\gamma^\mu v)\,\frac{g_{\mu\nu}}{q^2}\,(\bar{u}\gamma^\nu v)$$

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

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

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

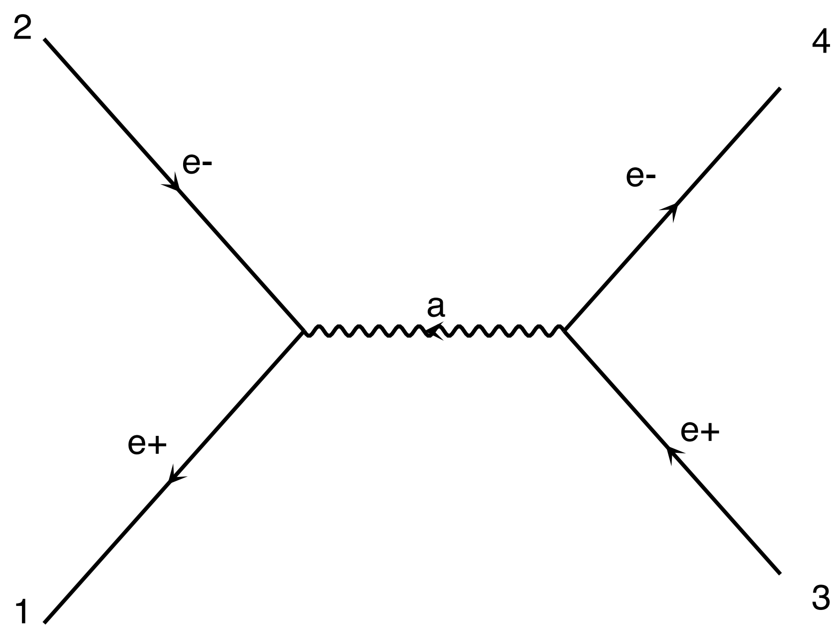
Very Efficient !!!

But The number of term raises as N^2

Only for 2>2 and 2>3

Basics: Helicity amplitudes

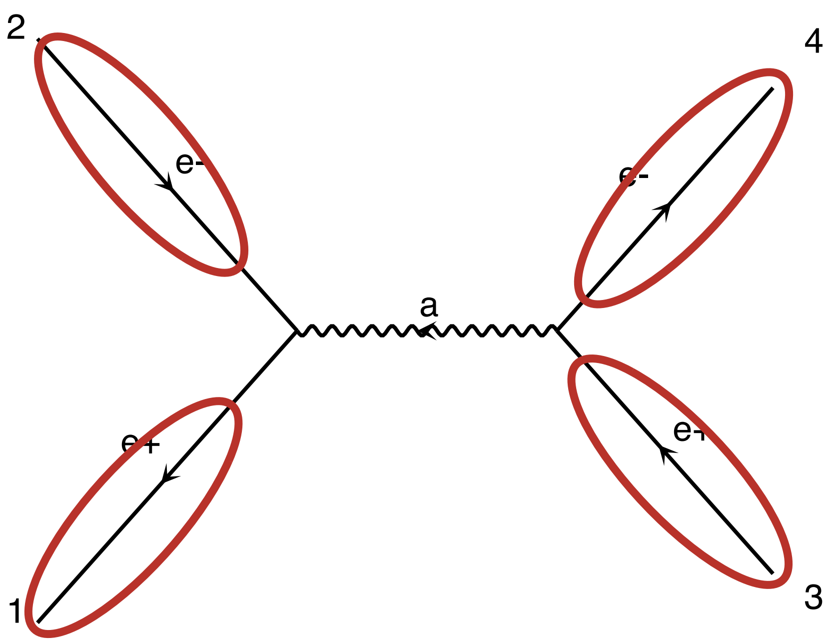
Idea: Evaluate $\mathscr{M}$ for fixed helicity of external particles



$$\mathcal{M} = \bar{u}\gamma^{\mu} v P_{\mu\nu} \bar{u} \gamma^{\nu} v$$

Basics: Helicity amplitudes

Idea: Evaluate $\mathscr{M}$ for fixed helicity of external particles

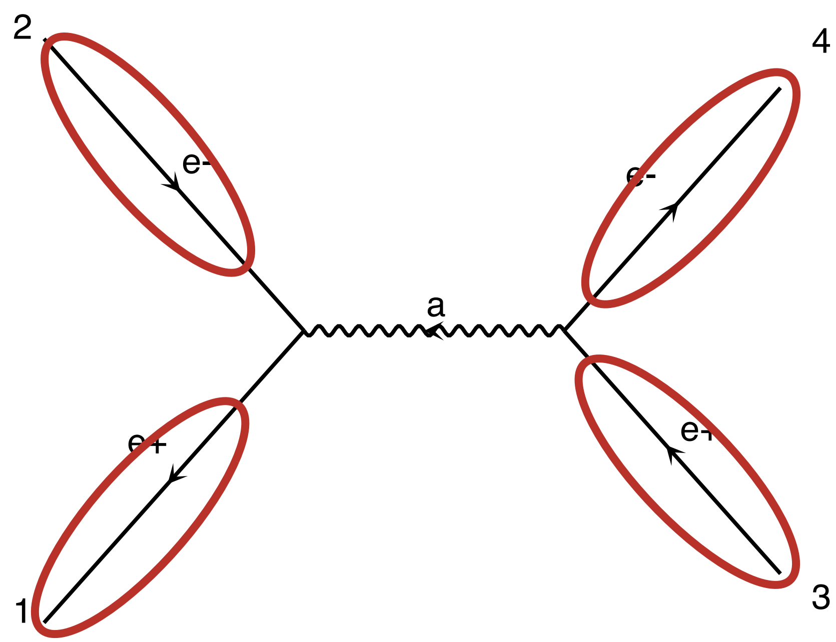


$$\mathcal{M} = \bar{u}\gamma^{\mu} v P_{\mu\nu} \bar{u}\gamma^{\nu} v$$

Numbers for given helicity and momenta

Basics: Helicity amplitudes

Idea: Evaluate $\mathscr{M}$ for fixed helicity of external particles



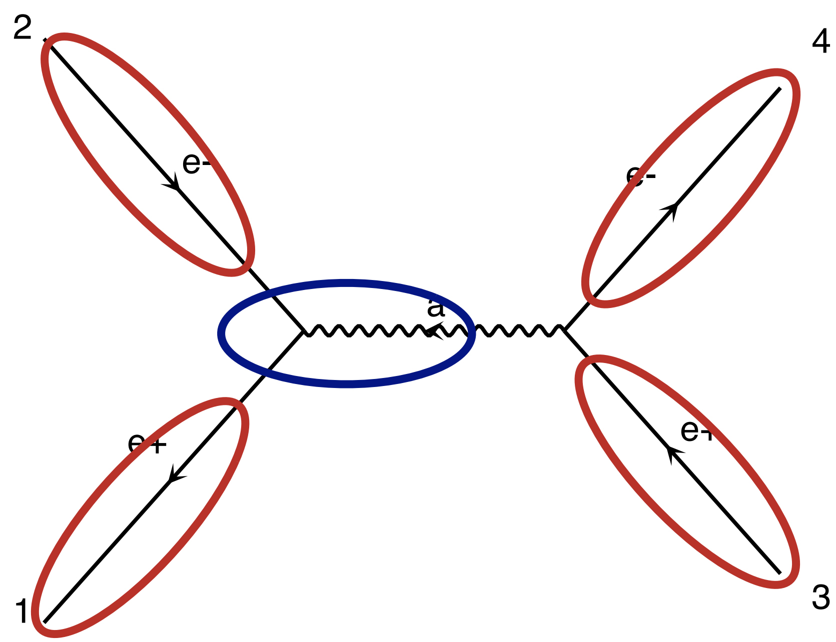
$$\mathcal{M} = \bar{u}\gamma^\mu v P_{\mu\nu} \bar{u}\gamma^\nu v$$

Numbers for given helicity and momenta

```
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))
```

Basics: Helicity amplitudes

Idea: Evaluate $\mathscr{M}$ for fixed helicity of external particles



$$\mathcal{M} = \bar{u}\gamma^\mu v P_{\mu\nu} \bar{u}\gamma^\nu v$$

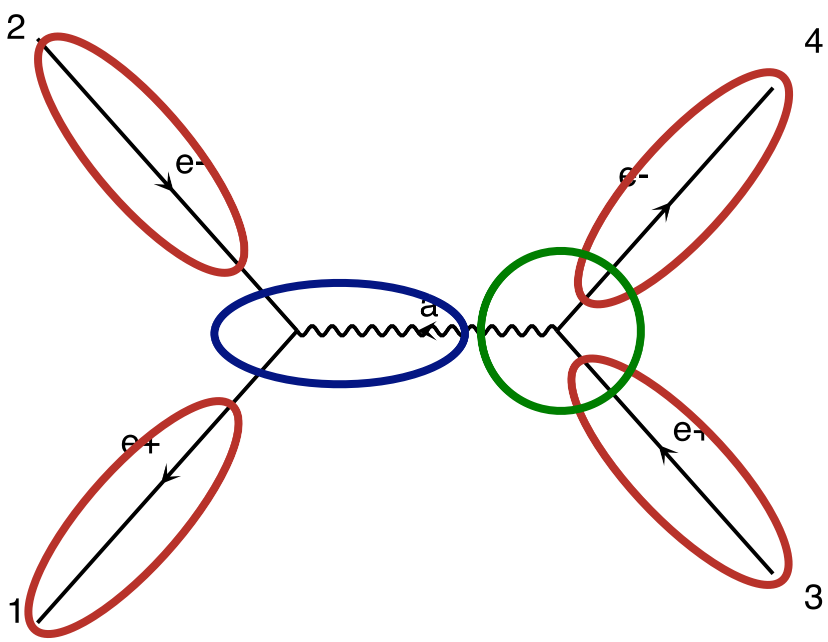
Numbers for given helicity and momenta

Calculate propagator wavefunctions

```
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 JIOXXX(W(1,2),W(1,1),GAL,ZERO,ZERO,W(1,5))
```

Basics: Helicity amplitudes

Idea: Evaluate $\mathscr{M}$ for fixed helicity of external particles



$$\mathcal{M} = \bar{u}\gamma^{\mu} v P_{\mu\nu} \bar{u}\gamma^{\nu} v$$

Numbers for given helicity and momenta

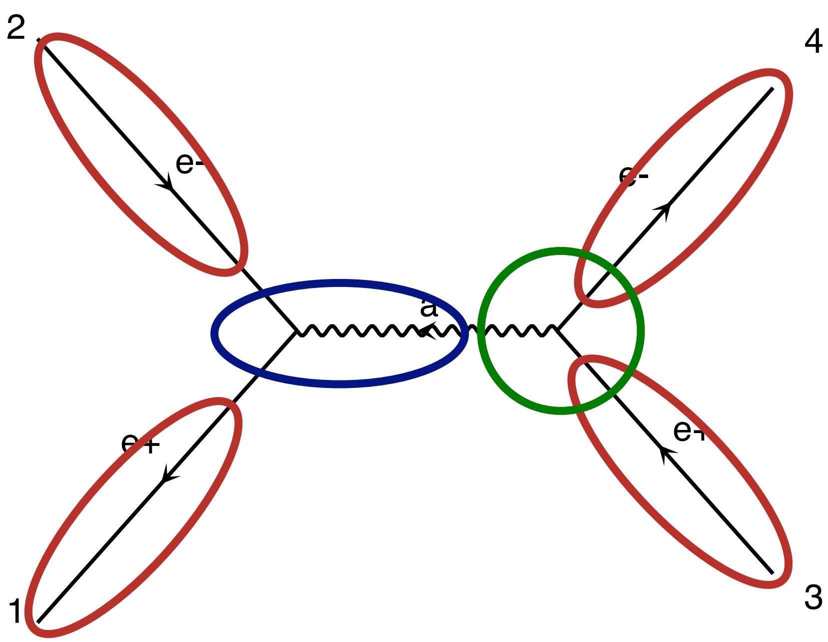
Calculate propagator wavefunctions

Finally evaluate amplitude (c-number)

```
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 JIOXXX(W(1,2),W(1,1),GAL,ZERO,ZERO,W(1,5))
CALL IOVXXX(W(1,3),W(1,4),W(1,5),GAL,AMP(1))
```

Basics: Helicity amplitudes

Idea: Evaluate $\mathscr{M}$ for fixed helicity of external particles



$$\mathcal{M} = \bar{u}\gamma^{\mu} v P_{\mu\nu} \bar{u}\gamma^{\nu} v$$

Numbers for given helicity and momenta

Calculate propagator wavefunctions

Finally evaluate amplitude (c-number)

Helicity amplitude calls written by MadGraph

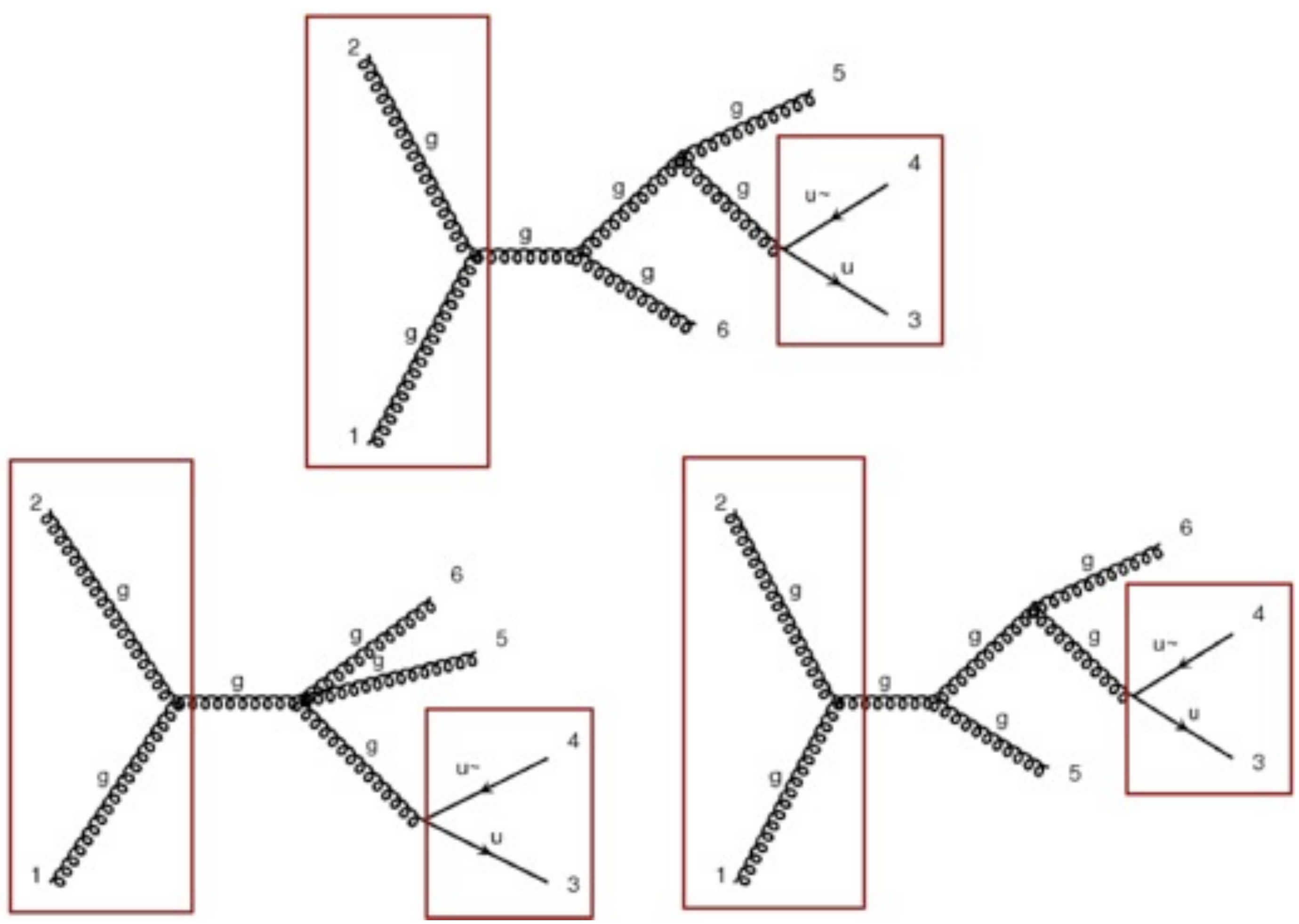
```
CALL OXXXXX(P(0,1),ZERO,NHEL(1),-1*IC(1),W(1,1))
CALL IXXXXX(P(0,2),ZERO,NHEL(2),+1*IC(2),W(1,2))
CALL IXXXXX(P(0,3),ZERO,NHEL(3),-1*IC(3),W(1,3))
CALL OXXXXX(P(0,4),ZERO,NHEL(4),+1*IC(4),W(1,4))
CALL JIOXXX(W(1,2),W(1,1),GAL,ZERO,ZERO,W(1,5))
CALL IOVXXX(W(1,3),W(1,4),W(1,5),GAL,AMP(1))
```

HELAS

- Original HELicity Amplitude Subroutine library
  Murayama, Watanabe, Hagiwara (1991)
- All helicity amplitude routines needed for the Standard Model, MSSM and certain other applications in hand-written library
- Any new Lorentz structures or other refinements need addition by hand
- Introduced a severe restriction on types of models that could be implemented in MadGraph

Basics: Helicity amplitudes

- Allows for fast calculation of any tree-level matrix element through efficient reuse of previously calculated wavefunctions across diagrams



Basics: Helicity amplitudes

- Thanks to new diagram generation algorithm, wf recycling much more efficient in MG5 than MG4

Process	Amplitudes	Wavefunctions		Run time	
		MG 4	MG 5	MG 4	MG 5
$u\bar{u} \to e^+e^-$	2	6	6	$< 6\mu s$	$< 6\mu s$
$u\bar{u} \to e^+e^-e^+e^-$	48	62	32	0.22 ms	0.14 ms
$u\bar{u} \to e^+e^-e^+e^-e^+e^-$	3474	3194	301	46.5 ms	19.0 ms
$u\bar{u} \to d\bar{d}$	1	5	5	$< 4\mu s$	$< 4\mu s$
$u\bar{u} \to d\bar{d}g$	5	11	11	27 μs	27 μs
$u\bar{u} \to d\bar{d}gg$	38	47	29	0.42 ms	0.31 ms
$u\bar{u} \to d\bar{d}ggg$	393	355	122	10.8 ms	6.75 ms
$u\bar{u} \to u\bar{u}gg$	76	84	40	1.24 ms	0.80 ms
$u\bar{u} \to u\bar{u}ggg$	786	682	174	35.7 ms	17.2 ms
$u\bar{u} \to d\bar{d}d\bar{d}$	14	28	19	84 μs	83 μs
$u\bar{u} \to d\bar{d}d\bar{d}g$	132	178	65	1.88 ms	1.15 ms
$u\bar{u} \to d\bar{d}d\bar{d}gg$	1590	1782	286	141 ms	34.4 ms
$u\bar{u} \to d\bar{d}d\bar{d}d\bar{d}$	612	758	141	42.5 ms	6.6 ms

Time for matrix element evaluation on a Sony Vaio TZ laptop

MadGraph 5

MadGraph 5

Alwall, Herquet, Maltoni, OM, Stelzer, arXiv:1106.0522

- MadGraph 5 is the successor of the well-known MadGraph matrix element generator
- Original MadGraph by Tim Stelzer was written in Fortran, first version from 1994 hep-ph/9401258
- Event generation by MadEvent using the single diagram enhanced multichannel integration technique in 2002 (Stelzer, Maltoni) hep-ph/0208156
- Full support for BSM (and many other improvements) in MG/ME 4 (2006-2008)

arXiv:0706.2334, arXiv:0809.2410

Master formula

Master formula

$$\hat{\sigma}_{ab\to X}(\hat{s},\ldots)$$

Parton level
cross section

- Parton level cross section from matrix element

Master formula

$$\hat{\sigma}_{ab\to X}(\hat{s},\ldots)\, f_a(x_1) f_b(x_2)$$

Parton level cross section

Parton density functions

- Parton level cross section from matrix element
- Parton density (or distribution) functions:
  Process independent, determined by particle type

Master formula

$$\int \hat{\sigma}_{ab\to X}(\hat{s},\ldots)\, f_a(x_1) f_b(x_2)\, dx_1 dx_2 d\Phi_{FS}$$

Parton level cross section | Parton density functions | Phase space integral

- Parton level cross section from matrix element
- Parton density (or distribution) functions:
  Process independent, determined by particle type

Demo

Using MadGraph on the Web!

To generate matrix elements using MadGraph:

- Go to http://madgraph.hep.uiuc.edu/
  (or google for MadGraph)
- Register
- Write your process
- Press Submit
- Download the tar file or
  generate events directly online on our clusters!

Process results

file:///Users/omatt/Documents/eclipse/madgraph5/PROC_sm_0/HTML/run_01/results.html

s= 0.00053102 ± 2.91e-07 (pb)

Graph	Cross-Section ↓	Error	Events (K)	Unwgt	Luminosity
P0 sum	0.000531025				
P0_ll_ll	0.000531	2.91e-07	358013.0	19088.0	0

P0_ll_ll

s= 0.00053102 ± 2.91e-07 (pb)

Graph	Cross-Section ↓	Error	Events (K)	Unwgt	Luminosity
G1	0.0004673	2.62e-07	330011.0	17468.0	3.74e+07
G2	6.368e-05	1.27e-07	28002.0	1620.0	2.55e+07

Not the single diagram contribution but:

$$f_i = \frac{|A_i|^2}{\sum_i |A_i|^2} |A_{\text{tot}}|^2$$

Using MadGraph on your computer!

To generate matrix elements and events:

- Download MadGraph 5 from https://launchpad.net/madgraph5
- Untar and run bin/mg5
- Write "generate *process*"
- Write "output"
- Write "launch"

Sounds easy? It is! Let me show you!

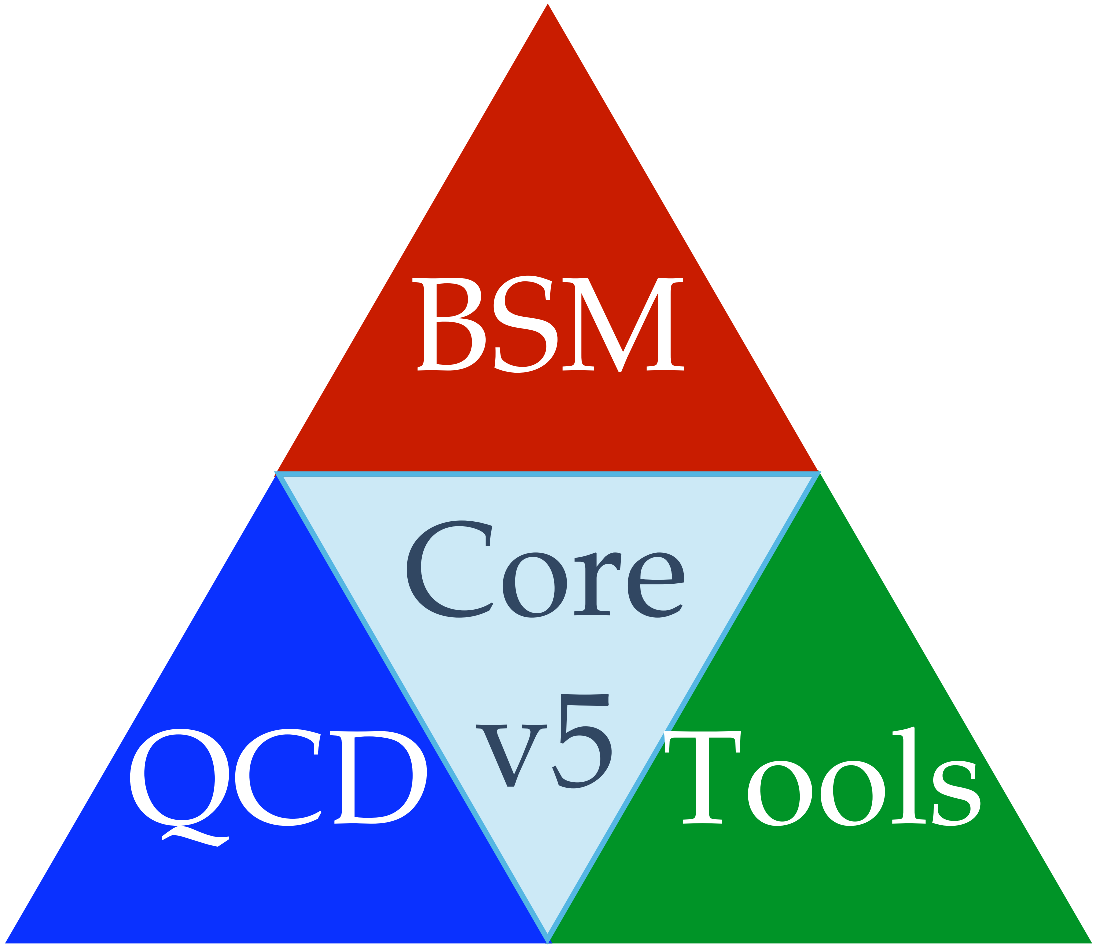
Examples shown

- p p > t t~
  This gives only (the dominant) QCD vertices, and ignores (the negligible) QED vertices.
- p p > t t~ QED=2
  This gives both QED and QCD vertices.
- p p > w+ j j, w+ > l+ vl
  More complicated example.

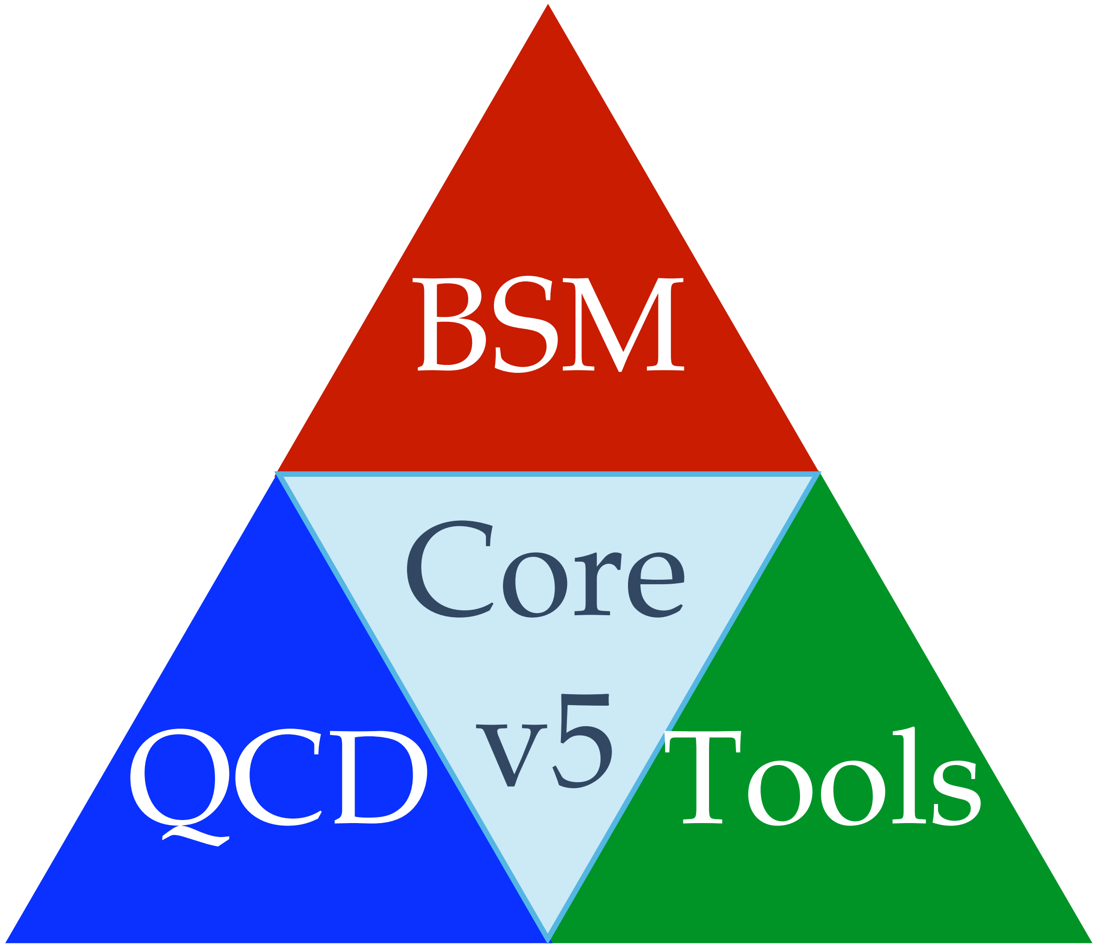
More syntax examples

- p p > t t~ j QED=2: Generate all combinations of processes for particles defined in multiparticle labels p / j, including up to two QED vertices (and unlimited QCD vertices)
- p p > t t~, (t > b w+, w+ > l+ vl), t~ > b~ j j :
  - Only diagrams compatible with given decay
  - Only t / t~ and W+ close to mass shell in event generation
- p p > w+ w- / h : Exclude any diagrams with h
- p p > w+ w- $ h : Exclude on-shell h in event generation (but retain interference effects)

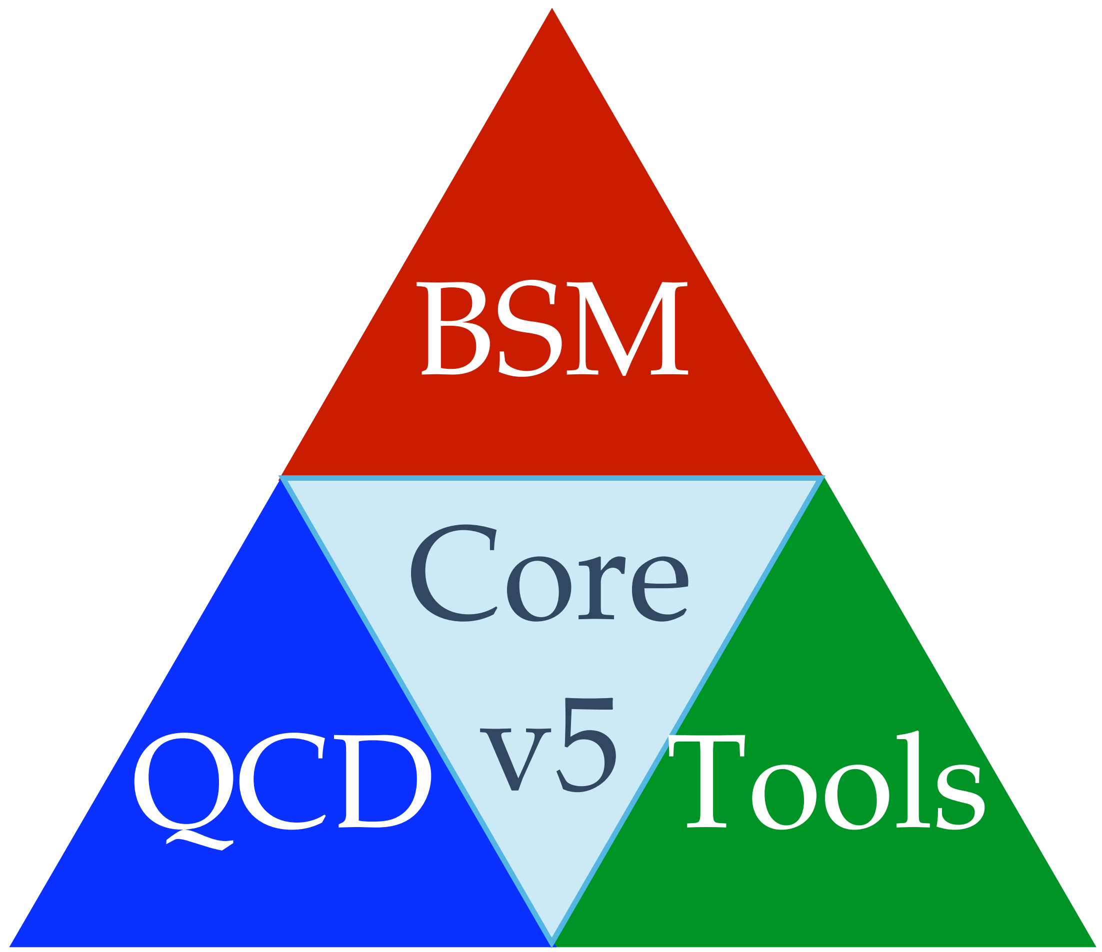
MadGraph 5 feature


BSM
Core
v5
QCD
Tools

UFO / ALOHA

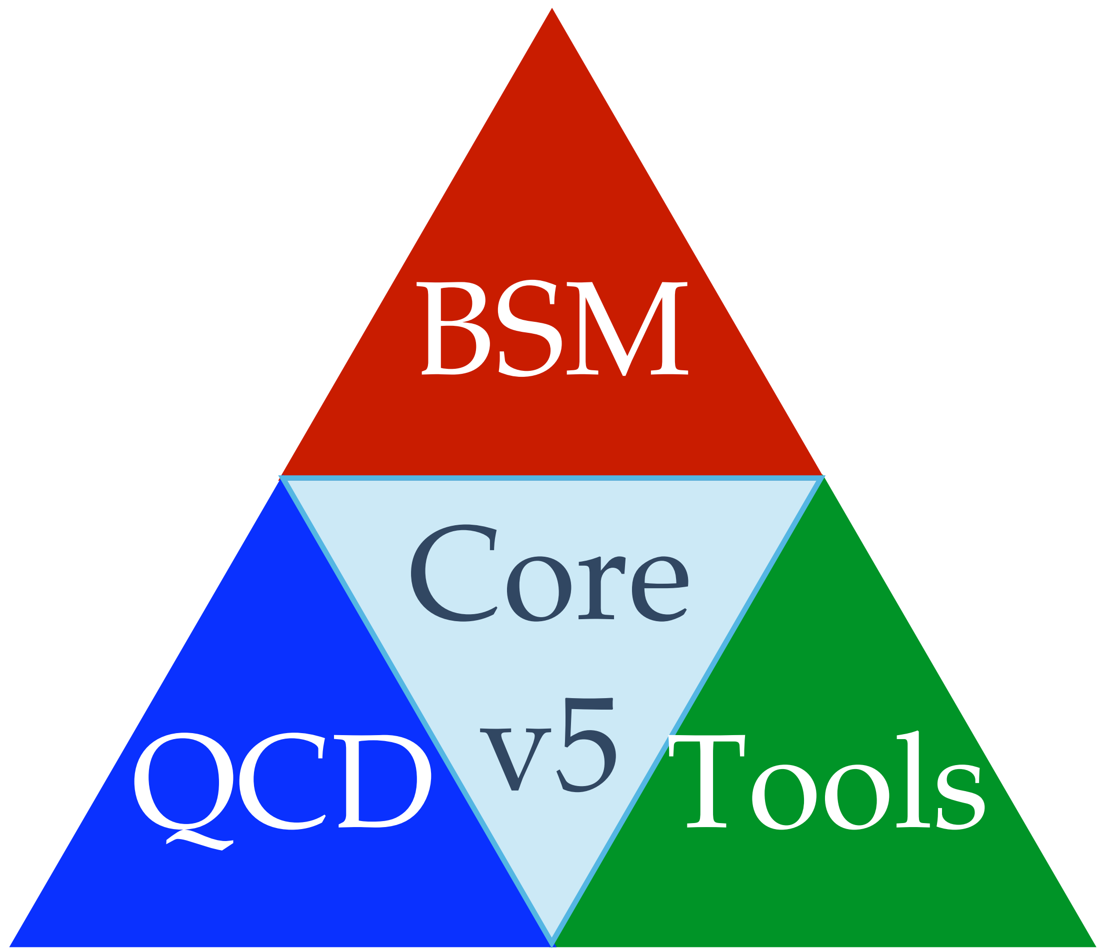


UFO / ALOHA



DECAY
MADWEIGHT
MadAnalysis5

UFO / ALOHA



MADLOOP
MADFKS
MADGOLEM

DECAY
MADWEIGHT
MadAnalysis5

Core
v5

- Diagram Generation
- Cross-section computation
- Generation of Events

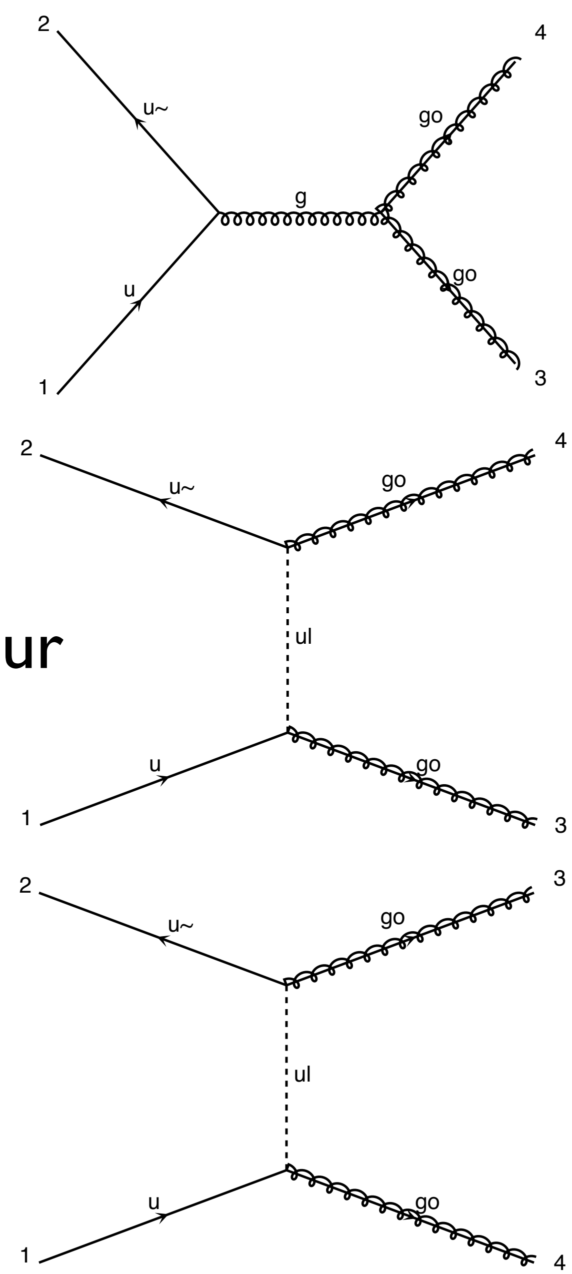
Output Format

- StandAlone (Fortran / C++)
- Event Generation (Fortran)
- Pythia 8 (C++)

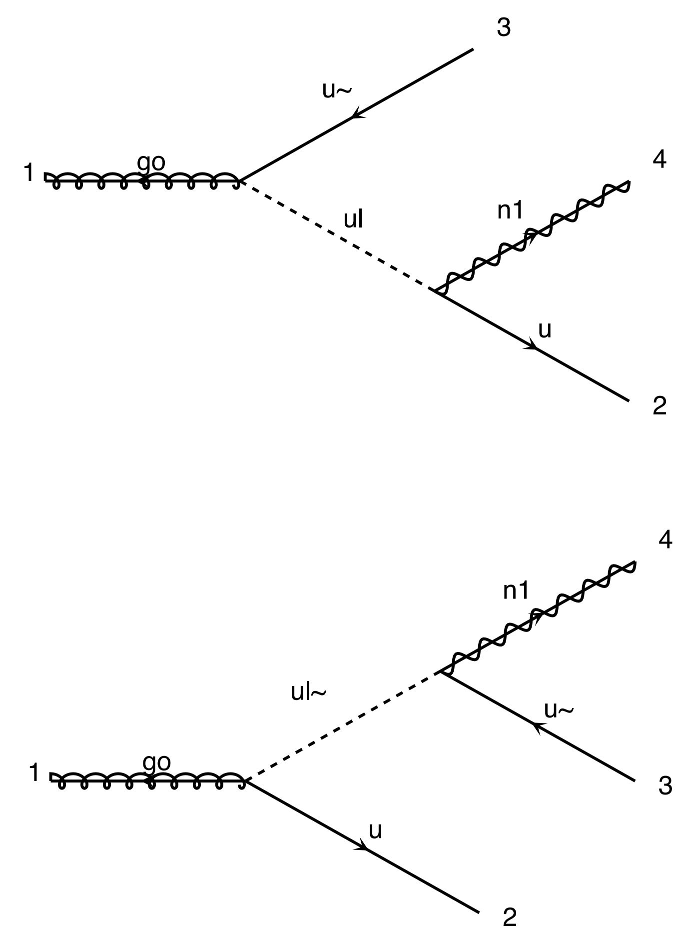
Decay chains

- If multiple diagrams in decays, need to multiply together core process and decay diagrams:

u u~ > go go / ur



x



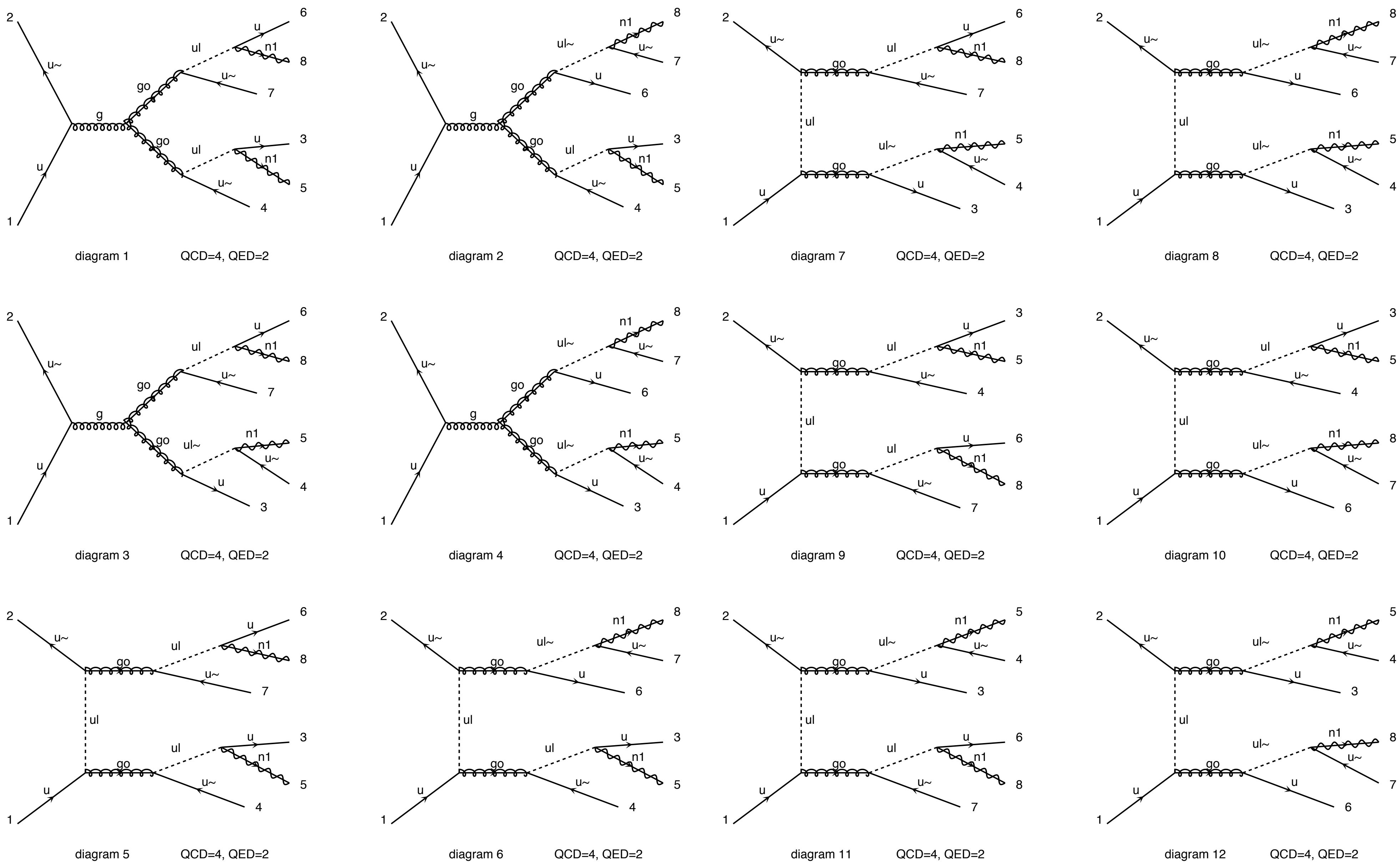
go > u u~ n1 / ur

(to the second power since both gluinos decay)

Decay chains

- If multiple diagrams in decays, need to multiply together core process and decay diagrams:

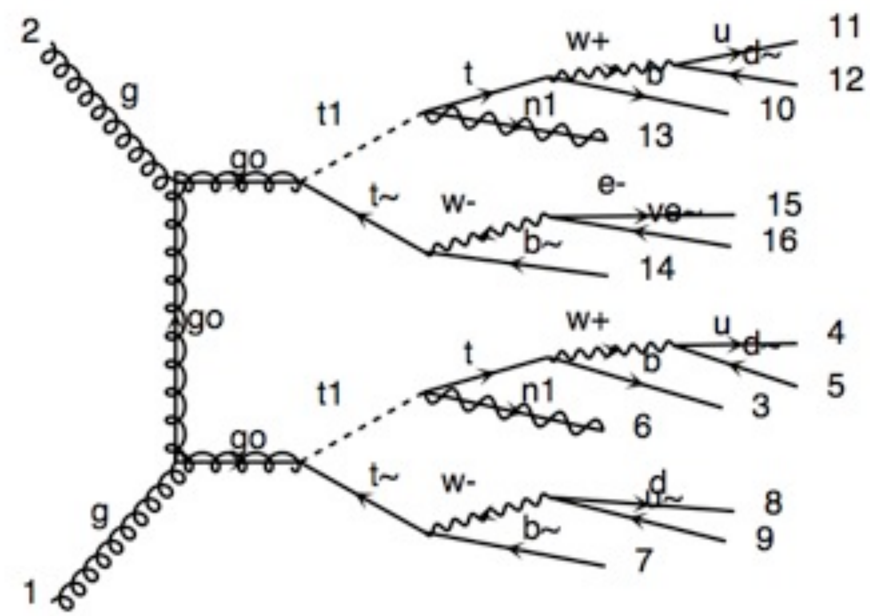
u u~ > go go / ur, go > u u~ n1 / ur

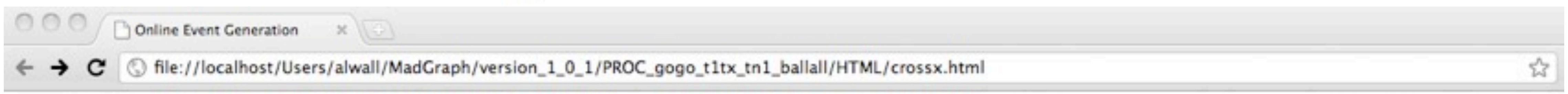


Decay chains

- Decay chains retain **full matrix element** for the diagrams compatible with the decay
- Full spin correlations (within and between decays)
- Full width effects
- However, no interference with non-resonant diagrams
  - ➡ Description only valid close to pole mass
  - ➡ Cutoff at $|m \pm n\Gamma|$ where n is set in run_card.

Decay chains





Results for g g > go go , (go > t1 t~ , t~ > b~ all all / h+ , (t1 > t n1 , t > b all all / h+)) in the mssm

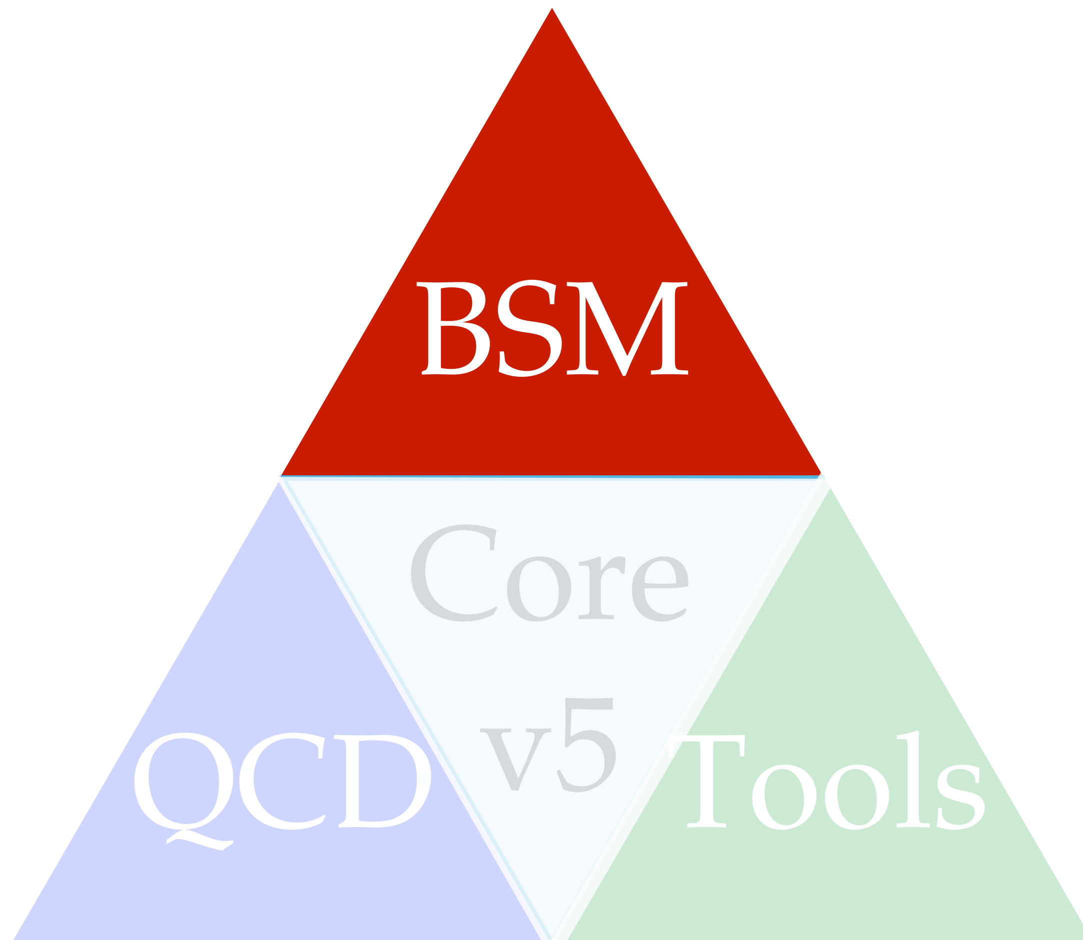
Available Results

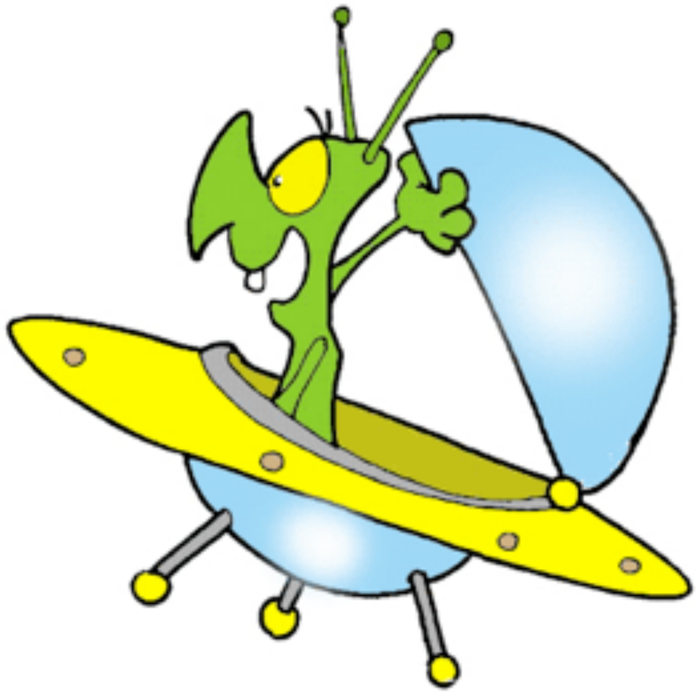
Links	Events	Tag	Run	Collider	Cross section (pb)	Events
results banner	Parton-level LHE	fermi	test	p p 7000 x 7000 GeV	.33857E-03	10000

Main Page

Thanks to developments in MadEvent, also (very) long decay chains possible to simulate directly in MadGraph!

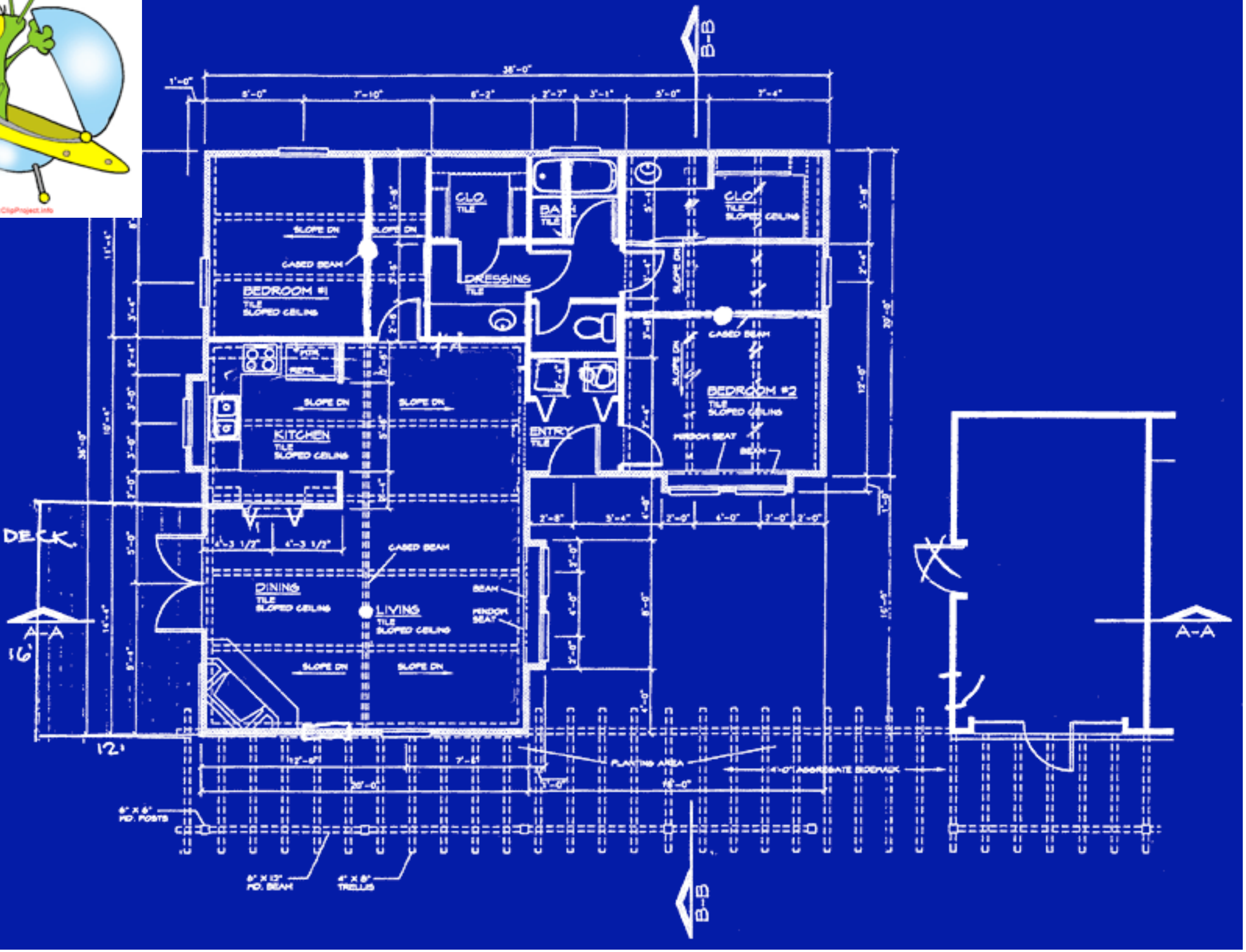
UFO / ALOHA





- New Model Format
- Gosam/ Herwig++/ MG5
- Fully generic color/Lorentz/...

- Automatic Creation of HELAS routine for ANY BSM theory
- Fortran / C++/Python


B-B
36'-0"
BEDROOM #1
TILE
SLOPED CEILING
CLO
TILE
DRESSING
TILE
CASED BEAM
SLOPE DN
KITCHEN
TILE
SLOPED CEILING
ENTRY
TILE
BEDROOM #2
TILE
SLOPED CEILING
WINDOW SEAT
DECK
A-A
DINING
TILE
SLOPED CEILING
LIVING
TILE
SLOPED CEILING
WINDOW SEAT
BEAM
PLANTING AREA
6" x 6" RD. POSTS
6" x 12" RD. BEAM
4" x 8" TRELLIS

Universal FeynRules Output (UFO)

vertices.py:

```
V_2 = Vertex(name = 'V_2',
         particles = [ P.G, P.G, P.G ],
         color = [ 'f(1,2,3)' ],
         lorentz = [ L.VVV1 ],
         couplings = {(0,0):C.GC_4})
```

Universal FeynRules Output (UFO)

particles.py:

```
G = Particle(pdg_code = 21,
         name = 'G',
         antiname = 'G',
         spin = 3,
         color = 8,
         mass = 'ZERO',
         width = 'ZERO',
         texname = 'G',
         antitexname = 'G',
         line = 'curly',
         charge = 0,
         LeptonNumber = 0,
         GhostNumber = 0)
```

vertices.py:

```
V_2 = Vertex(name = 'V_2',
         particles = [ P.G, P.G, P.G ],
         color = [ 'f(1,2,3)' ],
         lorentz = [ L.VVV1 ],
         couplings = {(0,0):C.GC_4})
```

Universal FeynRules Output (UFO)

particles.py:

```
G = Particle(pdg_code = 21,
        name = 'G',
        antiname = 'G',
        spin = 3,
        color = 8,
        mass = 'ZERO',
        width = 'ZERO',
        texname = 'G',
        antitexname = 'G',
        line = 'curly',
        charge = 0,
        LeptonNumber = 0,
        GhostNumber = 0)
```

vertices.py:

```
V_2 = Vertex(name = 'V_2',
        particles = [ P.G, P.G, P.G ],
        color = [ 'f(1,2,3)' ],
        lorentz = [ L.VVV1 ],
        couplings = {(0,0):C.GC_4})
```

Universal FeynRules Output (UFO)

particles.py:

```
G = Particle(pdg_code = 21,
        name = 'G',
        antiname = 'G',
        spin = 3,
        color = 8,
        mass = 'ZERO',
        width = 'ZERO',
        texname = 'G',
        antitexname = 'G',
        line = 'curly',
        charge = 0,
        LeptonNumber = 0,
        GhostNumber = 0)
```

lorentz.py:

```
VVV1 = Lorentz(name = 'VVV1',
        spins = [ 3, 3, 3 ],
        Structure =
            'P(3,1)*Metric(1,2) -
            P(3,2)*Metric(1,2) -
            P(2,1)*Metric(1,3) +
            P(2,3)*Metric(1,3) +
            P(1,2)*Metric(2,3) -
            P(1,3)*Metric(2,3)')
```

vertices.py:

```
V_2 = Vertex(name = 'V_2',
        particles = [ P.G, P.G, P.G ],
        color = [ 'f(1,2,3)' ],
        lorentz = [ L.VVV1 ],
        couplings = {(0,0):C.GC_4})
```

Universal FeynRules Output (UFO)

particles.py:
```
G = Particle(pdg_code = 21,
        name = 'G',
        antiname = 'G',
        spin = 3,
        color = 8,
        mass = 'ZERO',
        width = 'ZERO',
        texname = 'G',
        antitexname = 'G',
        line = 'curly',
        charge = 0,
        LeptonNumber = 0,
        GhostNumber = 0)
```

lorentz.py:
```
VVV1 = Lorentz(name = 'VVV1',
        spins = [ 3, 3, 3 ],
        Structure =
            'P(3,1)*Metric(1,2) -
            P(3,2)*Metric(1,2) -
            P(2,1)*Metric(1,3) +
            P(2,3)*Metric(1,3) +
            P(1,2)*Metric(2,3) -
            P(1,3)*Metric(2,3)')
```

couplings.py:
```
GC_4 = Coupling(name = 'GC_4',
        value = '-G',
        order = {'QCD':1})
```

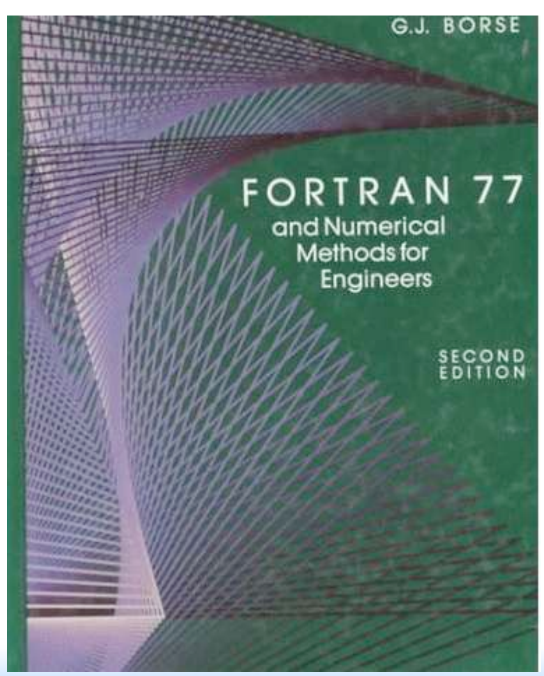
vertices.py:
```
V_2 = Vertex(name = 'V_2',
        particles = [ P.G, P.G, P.G ],
        color = [ 'f(1,2,3)' ],
        lorentz = [ L.VVV1 ],
        couplings = {(0,0):C.GC_4})
```

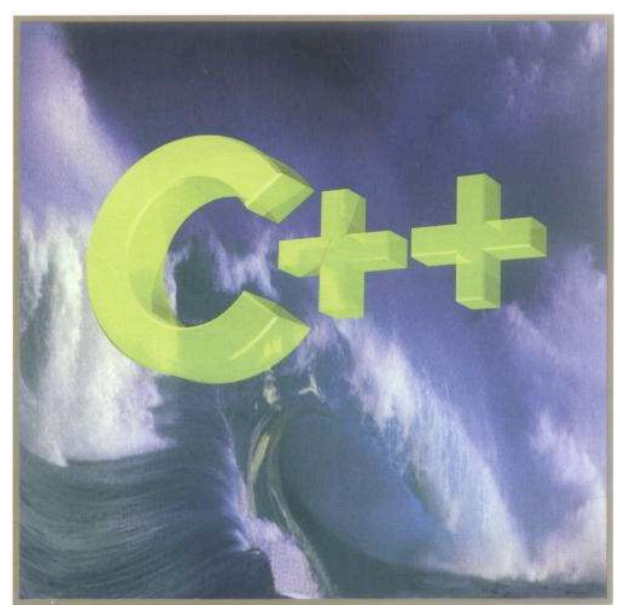
ALOHA

ALOHA ~~Google~~ translate

From: UFO ▾ To: Helicity [Translate]

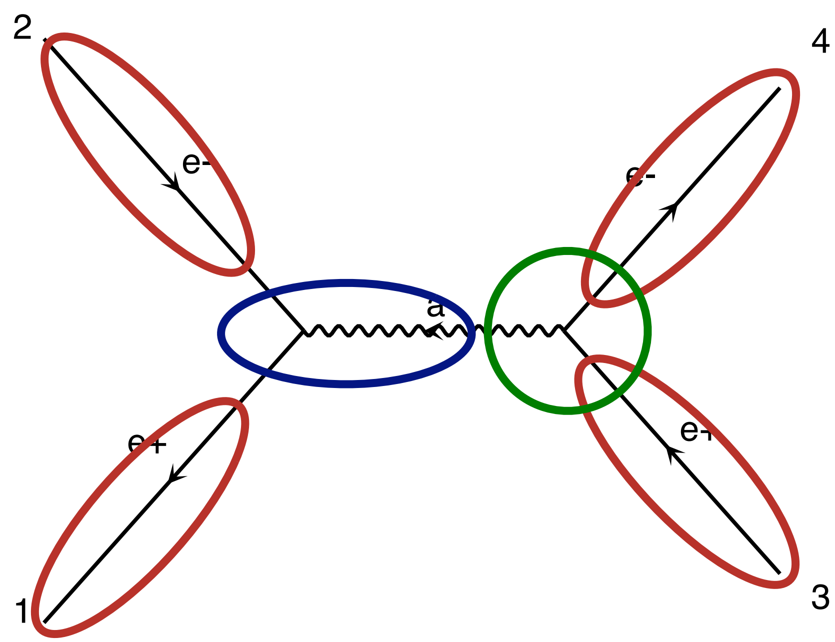
Type text or a website address or translate a document.





Basics: Helicity amplitudes

Idea: Evaluate $\mathscr{M}$ for fixed helicity of external particles



$$\mathcal{M} = \bar{u}\gamma^{\mu} v P_{\mu\nu} \bar{u}\gamma^{\nu} v$$

Numbers for given helicity and momenta

Calculate propagator wavefunctions

Finally evaluate amplitude (c-number)

Helicity amplitude calls written by MadGraph

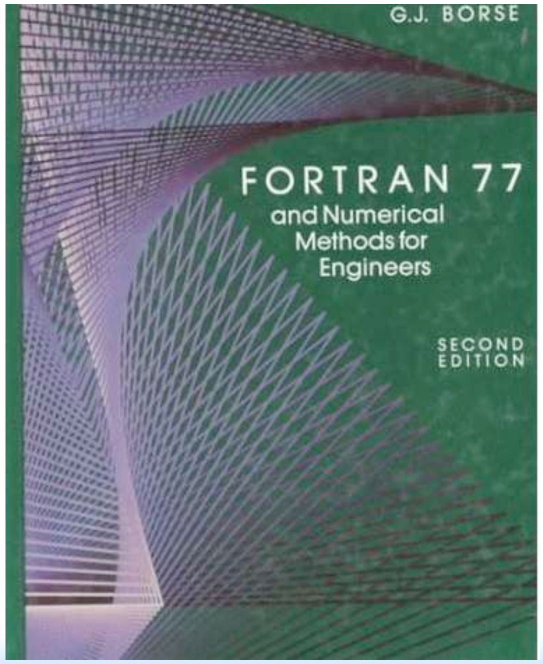
```
CALL OXXXXX(P(0,1),ZERO,NHEL(1),-1*IC(1),W(1,1))
CALL IXXXXX(P(0,2),ZERO,NHEL(2),+1*IC(2),W(1,2))
CALL IXXXXX(P(0,3),ZERO,NHEL(3),-1*IC(3),W(1,3))
CALL OXXXXX(P(0,4),ZERO,NHEL(4),+1*IC(4),W(1,4))
CALL JIOXXX(W(1,2),W(1,1),GAL,ZERO,ZERO,W(1,5))
CALL IOVXXX(W(1,3),W(1,4),W(1,5),GAL,AMP(1))
```

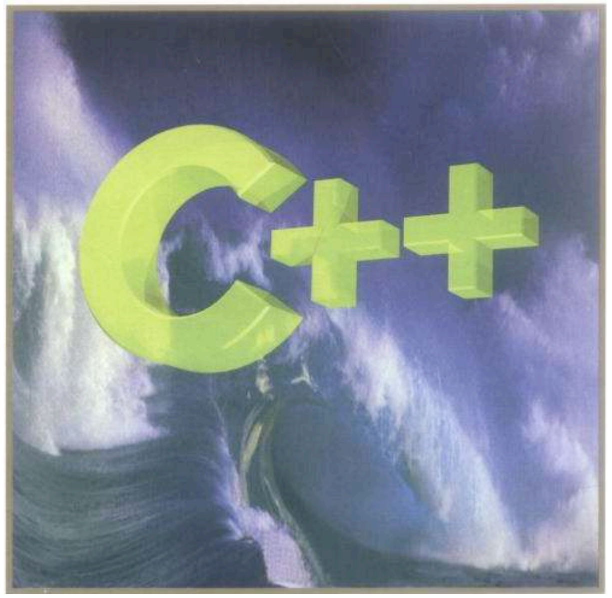
ALOHA

ALOHA
~~Google~~ translate

From: UFO ⇆ To: Helicity [Translate]

Type text or a website address or translate a document.





ALOHA

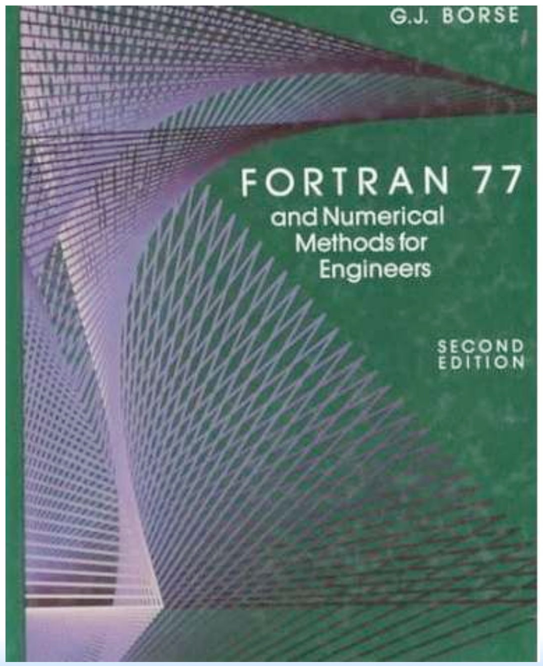
ALOHA

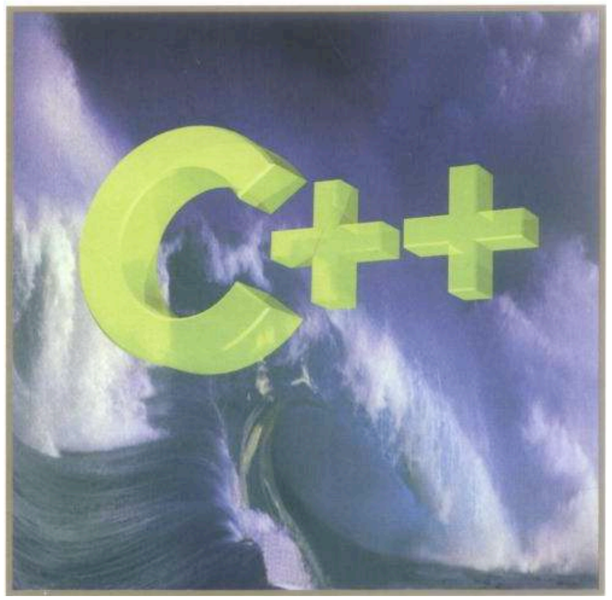
~~Google~~ translate

From: UFO To: Helicity Translate

Options: Standard (HELAS)
Feynman gauge
Complex-mass scheme
Loop

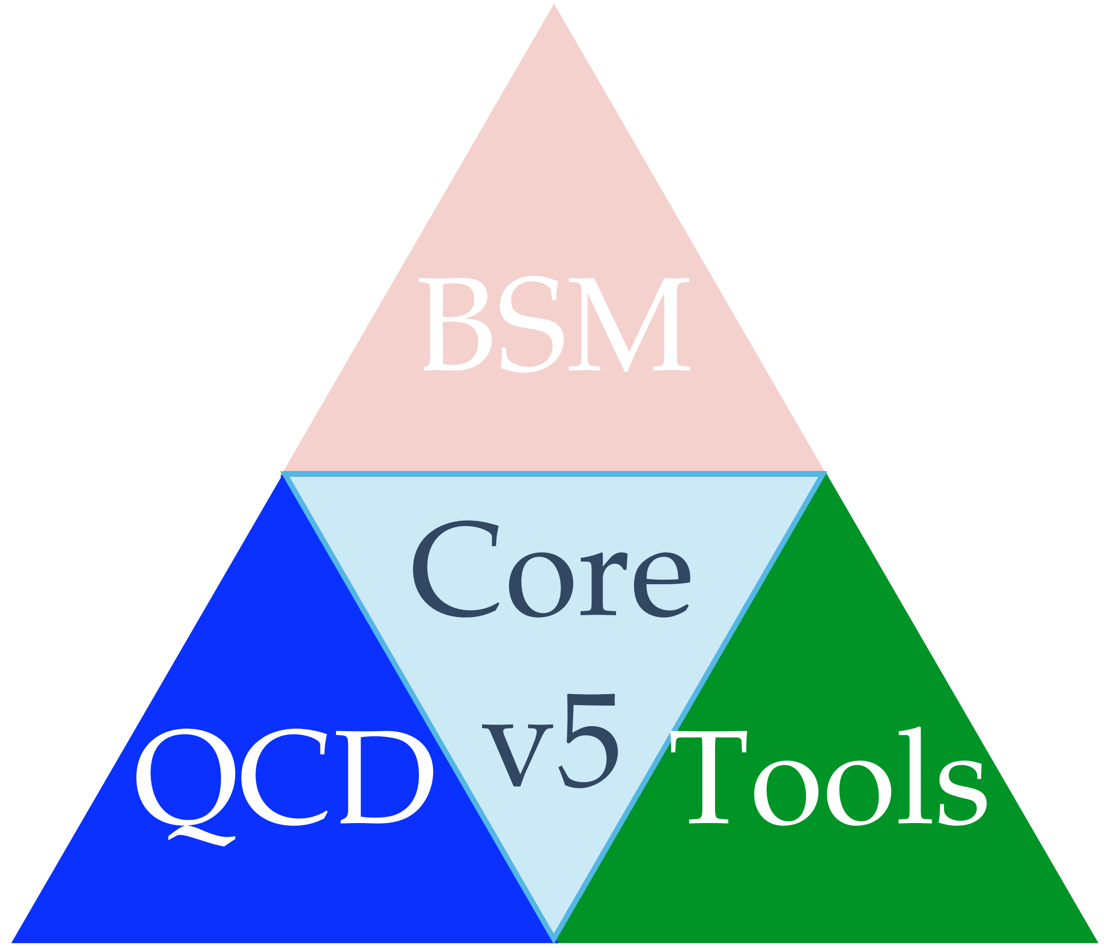
Type text or a website address or translate a document.





UFO + ALOHA + MG5 =

Any BSM should be possible in a fully automatic and efficient way!



MADLOOP
MADFKS
MADGOLEM

DECAY
MADWEIGHT
MadAnalysis5

See the others Lectures !!

It’s your turn to play with it

The Cross Section

Master formula

Master formula

$$\hat{\sigma}_{ab\to X}(\hat{s},\ldots)$$

Parton level
cross section

- Parton level cross section from matrix element

Master formula

$$\hat{\sigma}_{ab\to X}(\hat{s},\ldots)\, f_a(x_1) f_b(x_2)$$

Parton level cross section

Parton density functions

- Parton level cross section from matrix element
- Parton density (or distribution) functions:
  Process independent, determined by particle type

Master formula

$$\int \hat{\sigma}_{ab\to X}(\hat{s},\ldots)\, f_a(x_1) f_b(x_2)\, dx_1 dx_2 d\Phi_{FS}$$

Parton level cross section | Parton density functions | Phase space integral

- Parton level cross section from matrix element
- Parton density (or distribution) functions: Process independent, determined by particle type

Master formula

$$\int \hat{\sigma}_{ab\to X}(\hat{s},\ldots)\, f_a(x_1) f_b(x_2)\, dx_1 dx_2 d\Phi_{FS}$$

Parton level cross section | Parton density functions | Phase space integral

- Parton level cross section from matrix element
- Parton density (or distribution) functions:
  Process independent, determined by particle type
- $\hat{s} = x_1 x_2 s$ (s = collision energy of the collider)

Integrals as averages

Integrals as averages

$$I = \int_{x_1}^{x_2} f(x)dx \quad \Rightarrow \quad I_N = (x_2 - x_1)\frac{1}{N}\sum_{i=1}^{N} f(x)$$

$$V = (x_2 - x_1)\int_{x_1}^{x_2} [f(x)]^2 dx - I^2 \quad \Rightarrow \quad V_N = (x_2 - x_1)^2 \frac{1}{N}\sum_{i=1}^{N}[f(x)]^2 - I_N^2$$

Integrals as averages

$$I = \int_{x_1}^{x_2} f(x)dx \quad \Rightarrow \quad I_N = (x_2 - x_1)\frac{1}{N}\sum_{i=1}^{N} f(x)$$

$$V = (x_2 - x_1)\int_{x_1}^{x_2} [f(x)]^2 dx - I^2 \quad \Rightarrow \quad V_N = (x_2 - x_1)^2 \frac{1}{N}\sum_{i=1}^{N}[f(x)]^2 - I_N^2$$

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

Integrals as averages

$$I = \int_{x_1}^{x_2} f(x)dx \quad \Rightarrow \quad I_N = (x_2 - x_1)\frac{1}{N}\sum_{i=1}^{N} f(x)$$

$$V = (x_2 - x_1)\int_{x_1}^{x_2} [f(x)]^2 dx - I^2 \quad \Rightarrow \quad V_N = (x_2 - x_1)^2 \frac{1}{N}\sum_{i=1}^{N}[f(x)]^2 - I_N^2$$

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

☞ Convergence is slow but it can be easily estimated

Integrals as averages

$$I = \int_{x_1}^{x_2} f(x)dx \quad \Rightarrow \quad I_N = (x_2 - x_1)\frac{1}{N}\sum_{i=1}^{N} f(x)$$

$$V = (x_2 - x_1)\int_{x_1}^{x_2} [f(x)]^2 dx - I^2 \quad \Rightarrow \quad V_N = (x_2 - x_1)^2 \frac{1}{N}\sum_{i=1}^{N}[f(x)]^2 - I_N^2$$

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

☞ Convergence is slow but it can be easily estimated

☞ Improvement by minimizing V_N.

Integrals as averages

$$I = \int_{x_1}^{x_2} f(x)dx \quad \Rightarrow \quad I_N = (x_2 - x_1)\frac{1}{N}\sum_{i=1}^{N} f(x)$$

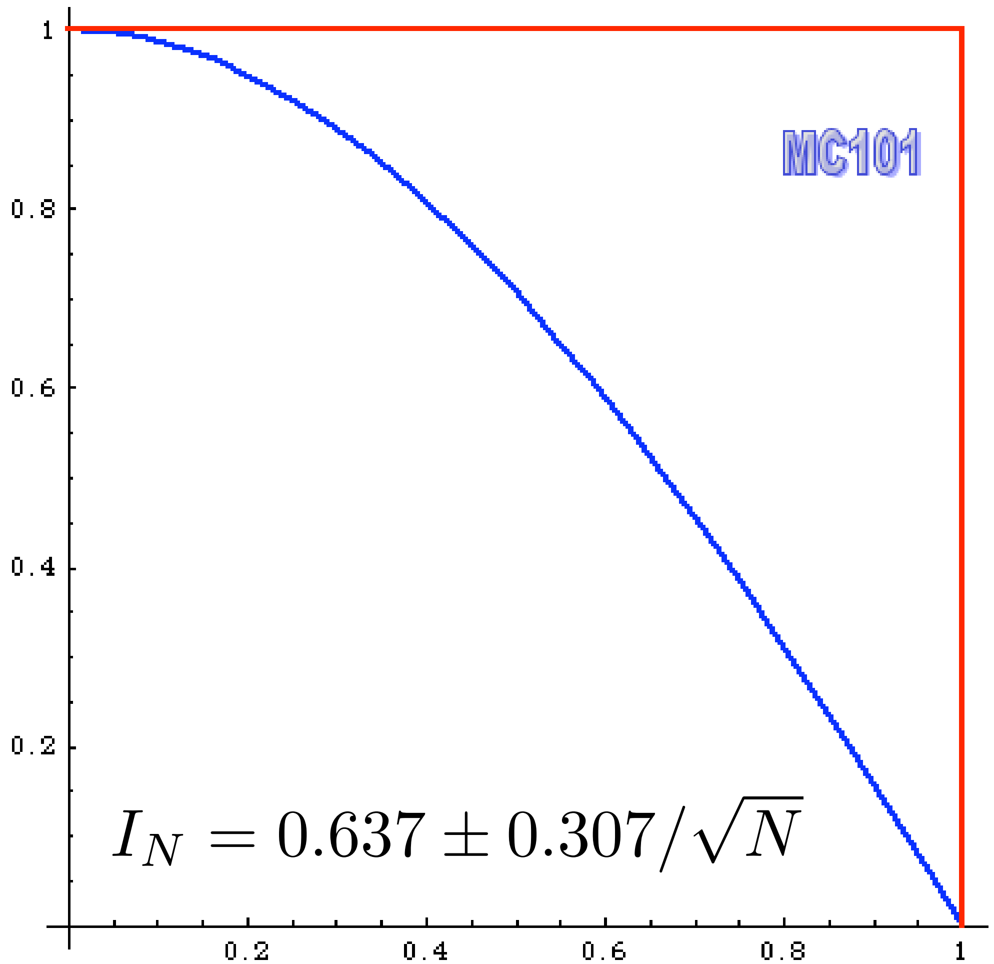
$$V = (x_2 - x_1)\int_{x_1}^{x_2} [f(x)]^2 dx - I^2 \quad \Rightarrow \quad V_N = (x_2 - x_1)^2 \frac{1}{N}\sum_{i=1}^{N}[f(x)]^2 - I_N^2$$

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

- ☞ Convergence is slow but it can be easily estimated
- ☞ Improvement by minimizing V_N.
- ☞ Optimal/Ideal case: $f(x)=C \Rightarrow V_N=0$

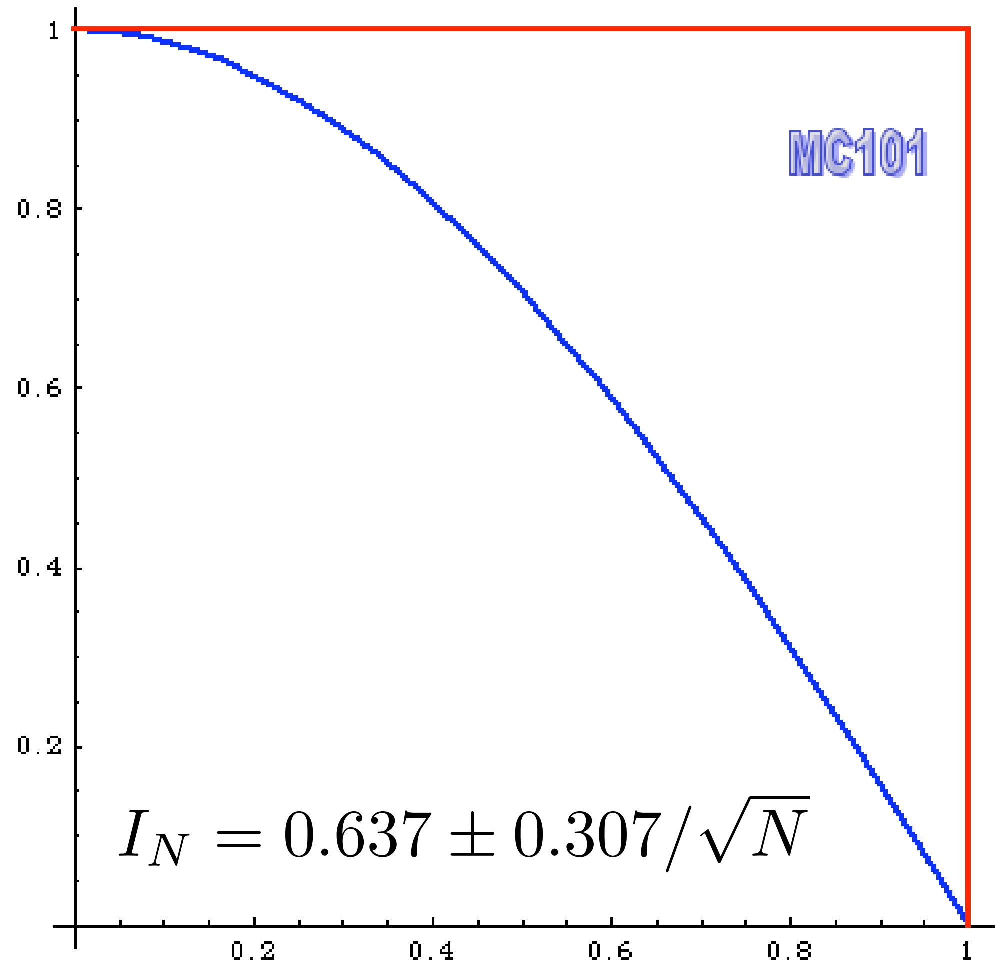
Importance Sampling

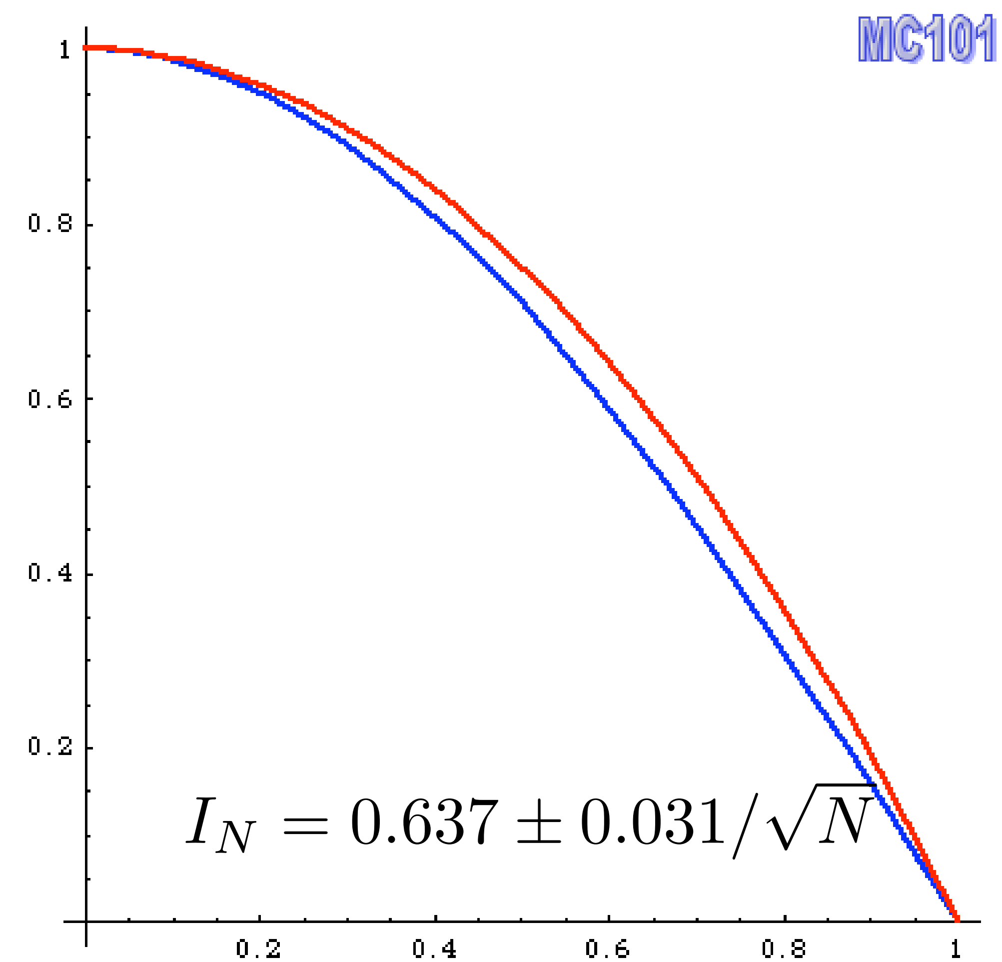
Importance Sampling



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

Importance Sampling

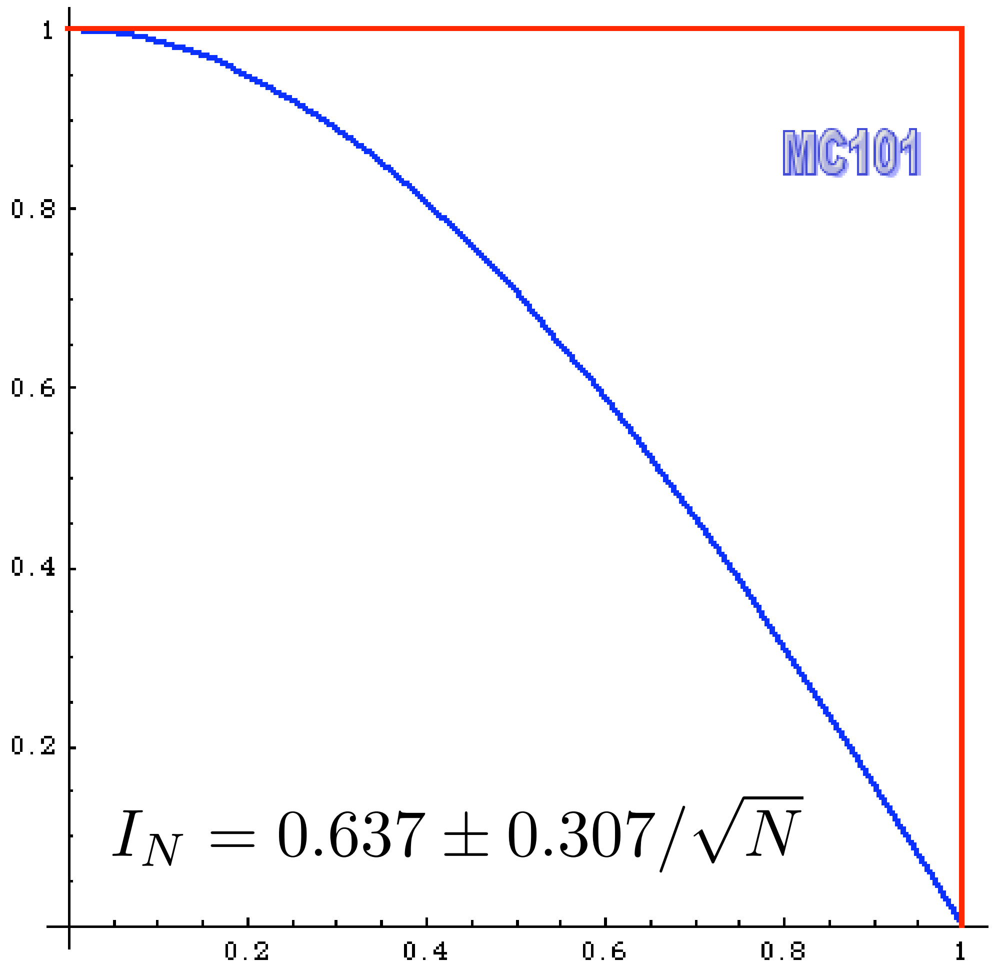


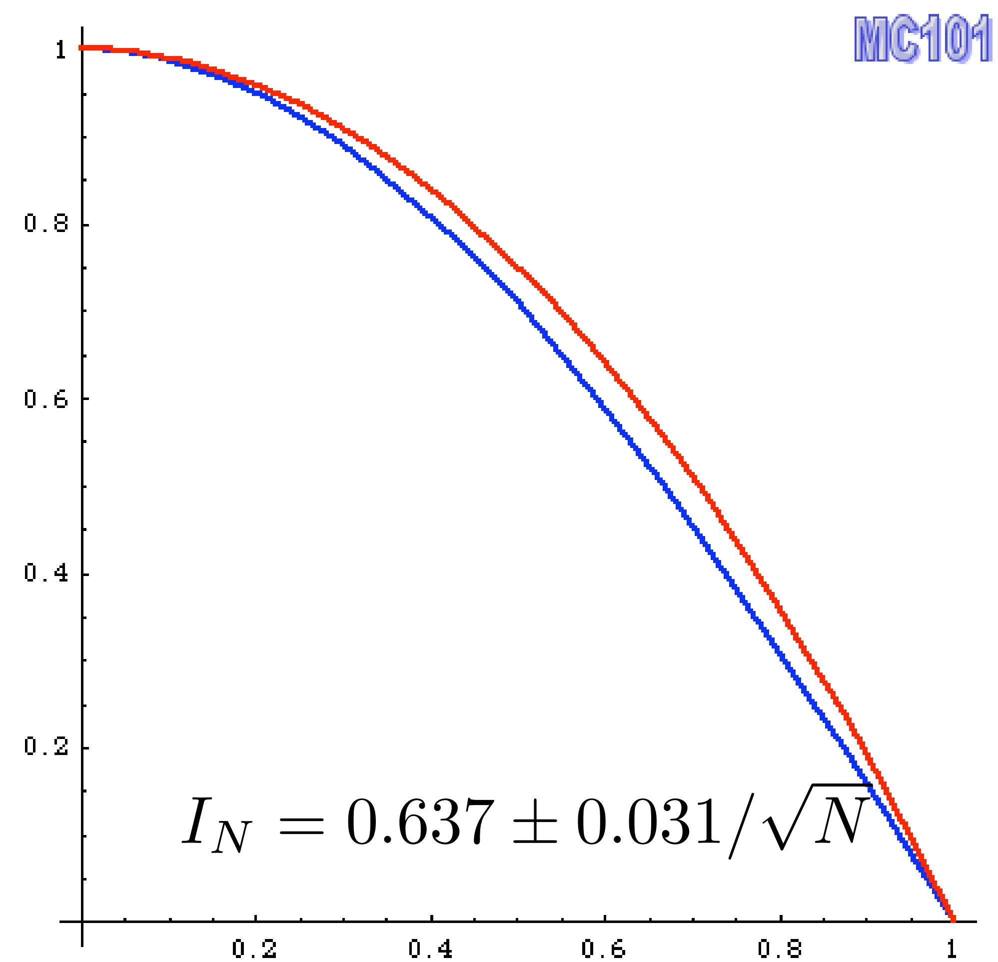


$$I = \int_0^1 dx \cos\frac{\pi}{2}x$$

$$I = \int_0^1 dx (1 - x^2) \frac{\cos\frac{\pi}{2}x}{1 - x^2}$$

Importance Sampling



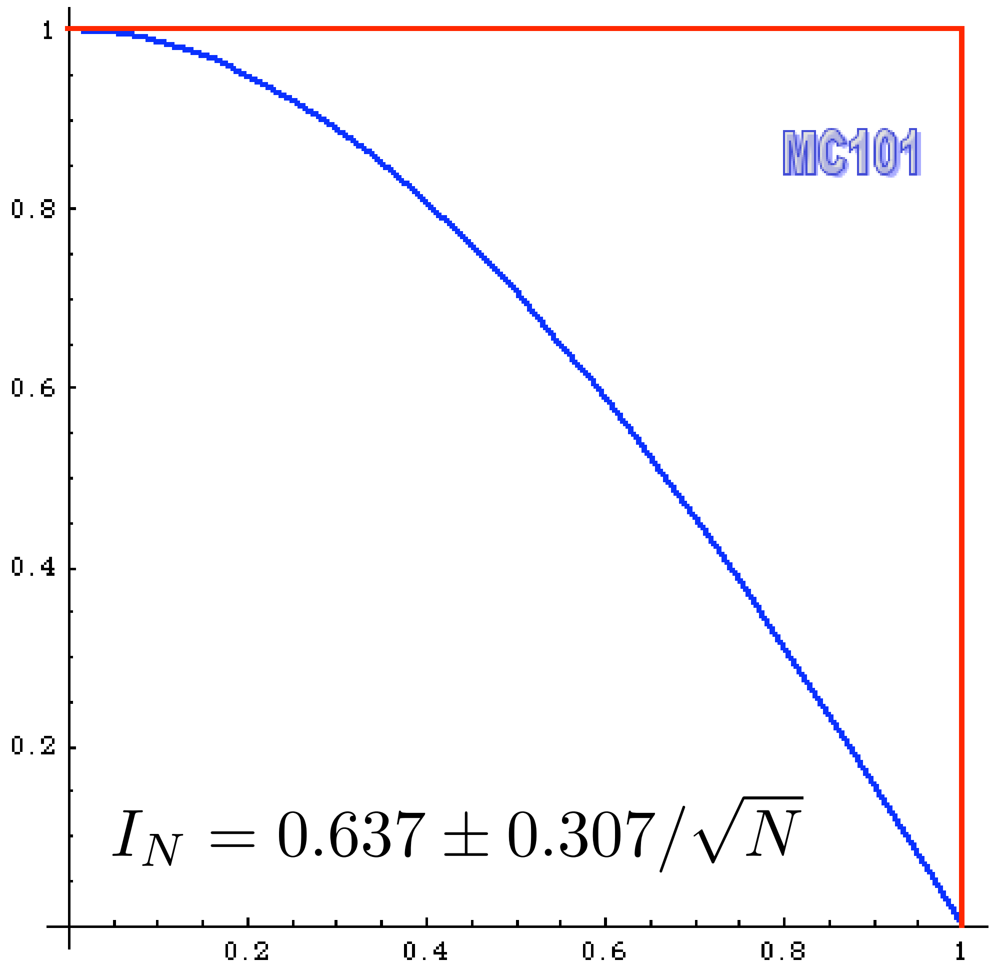


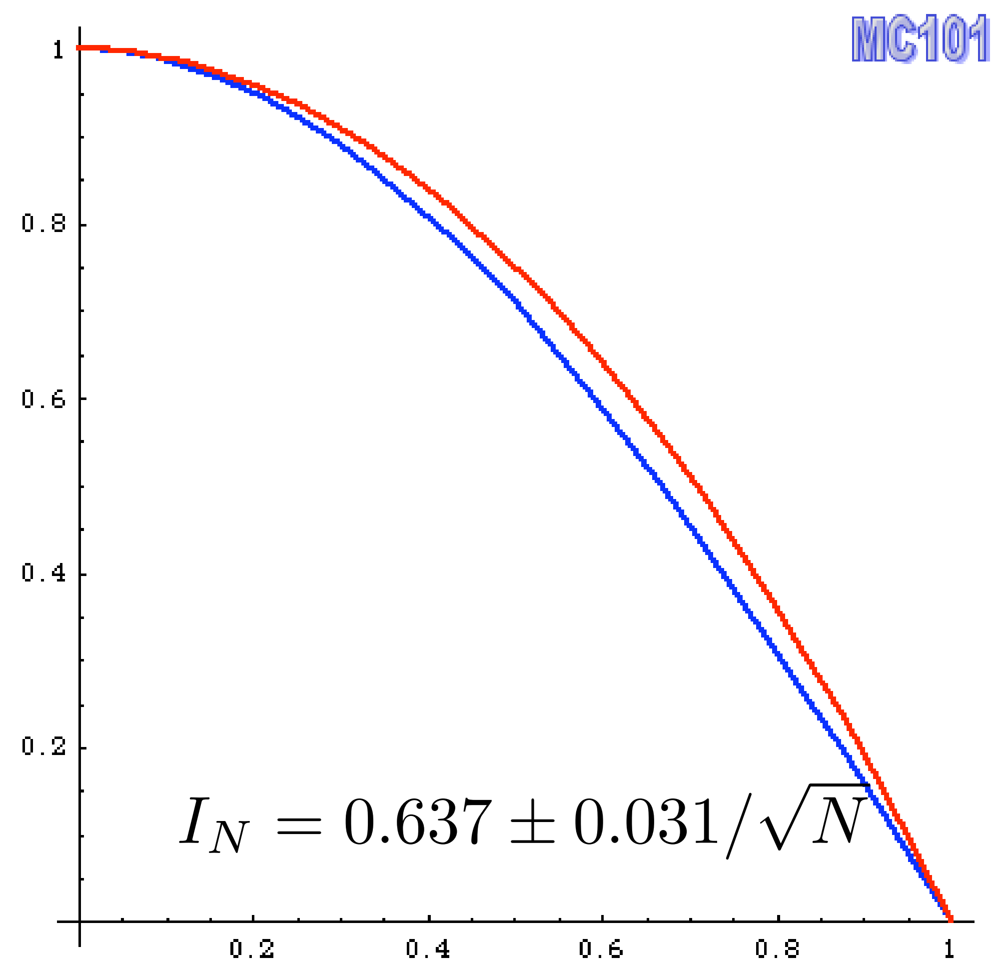
$$I = \int_0^1 dx \cos\frac{\pi}{2}x$$

$$I = \int_0^1 dx (1 - x^2) \frac{\cos\frac{\pi}{2}x}{1 - x^2}$$

$$= \int_{\xi_1}^{\xi_2} d\xi \frac{\cos\frac{\pi}{2}x[\xi]}{1 - x[\xi]^2}$$

Importance Sampling



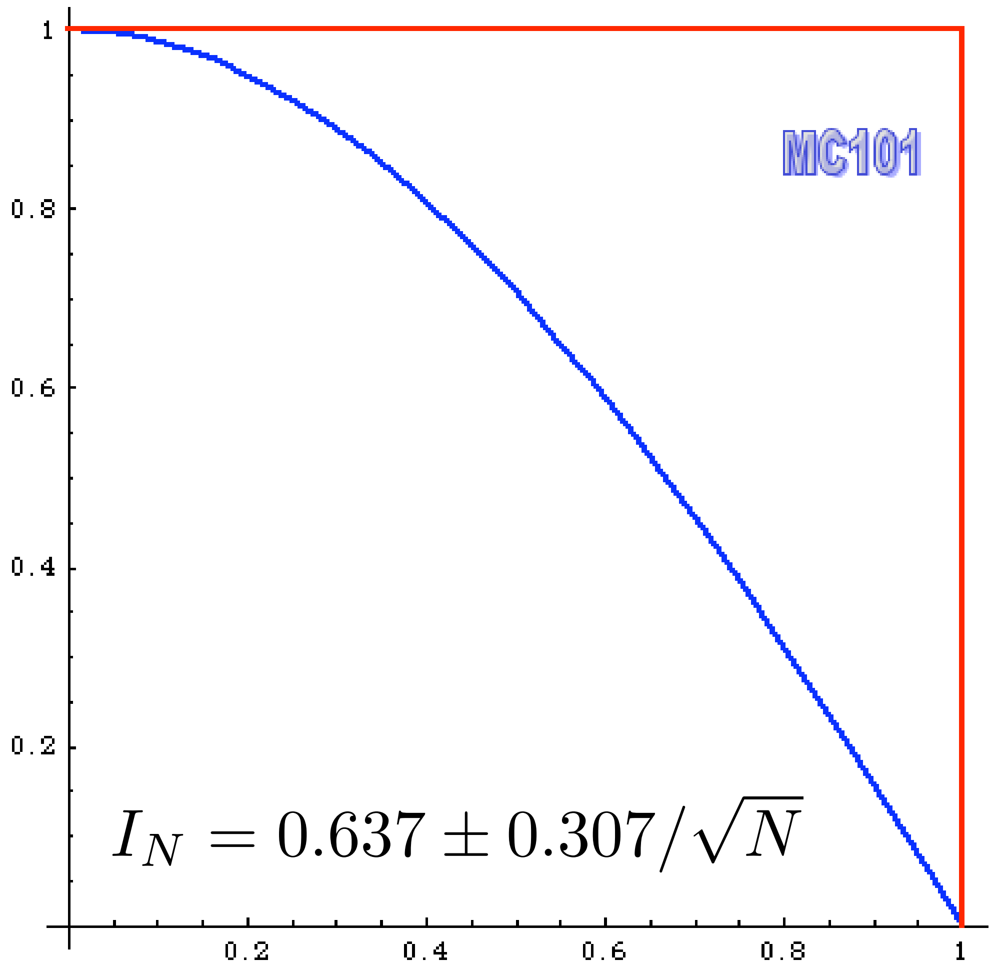


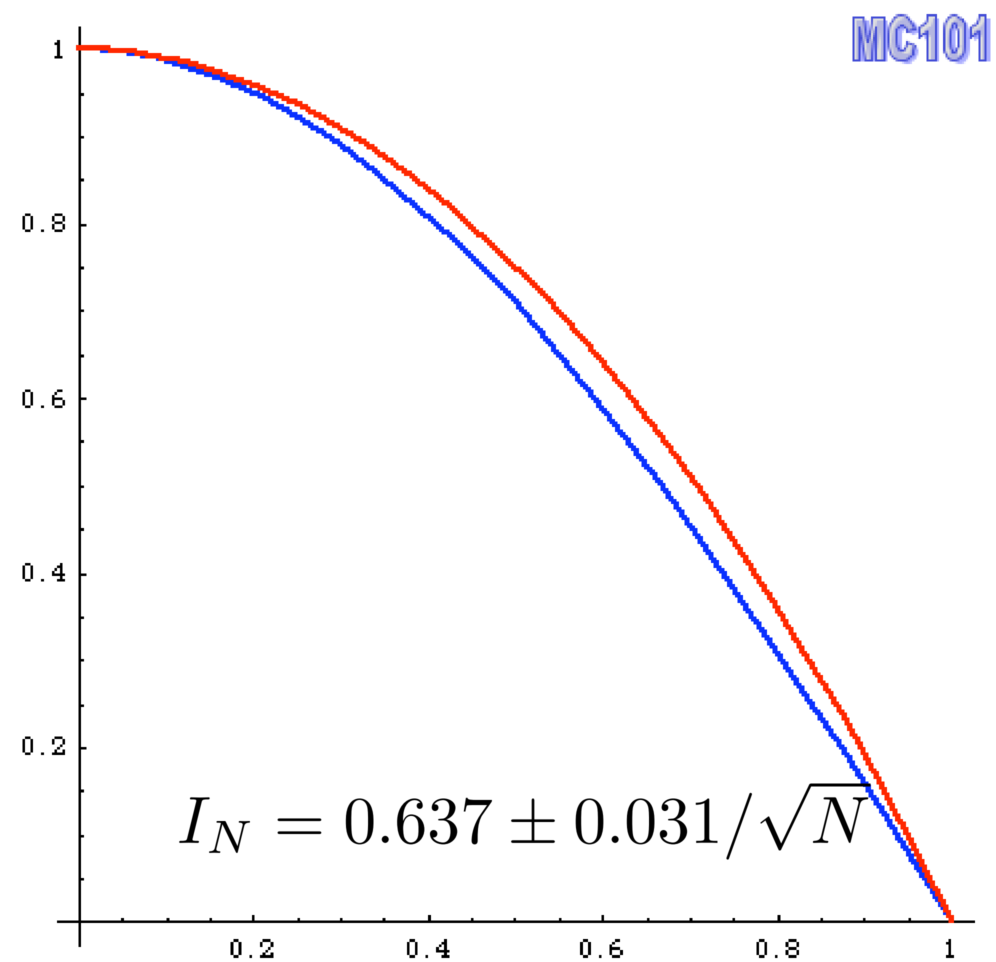
$$I = \int_0^1 dx \cos \frac{\pi}{2} x$$

$$I = \int_0^1 dx (1 - x^2) \frac{\cos \frac{\pi}{2} x}{1 - x^2}$$

$$= \int_{\xi_1}^{\xi_2} d\xi \frac{\cos \frac{\pi}{2} x[\xi]}{1 - x[\xi]^2}$$

Importance Sampling



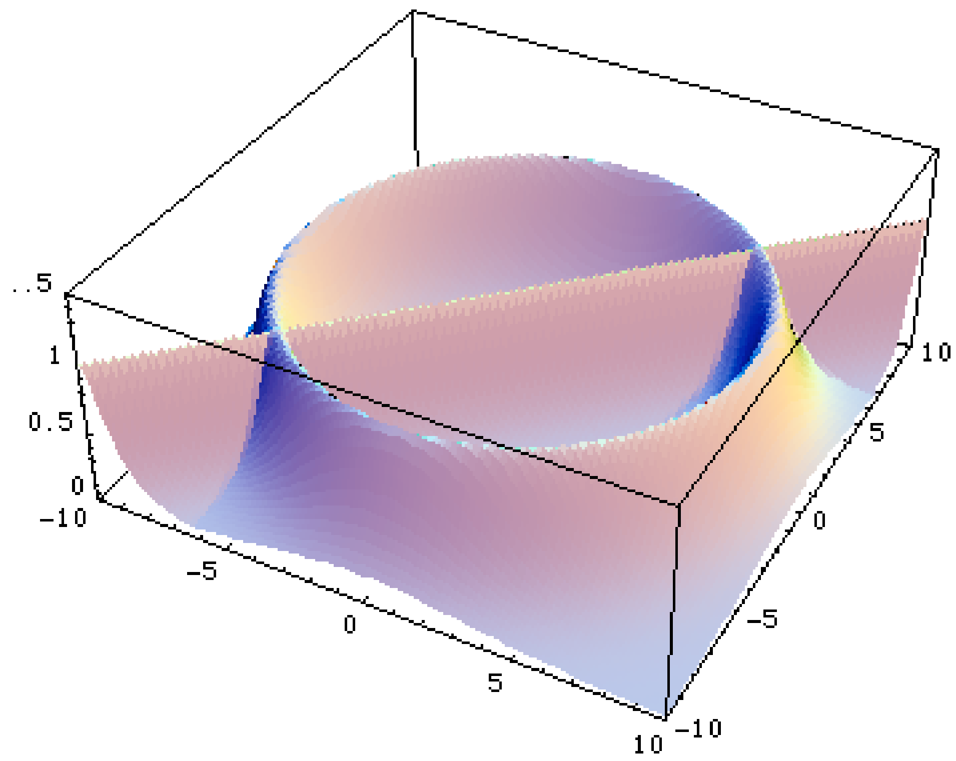


$$I = \int_0^1 dx \cos\frac{\pi}{2}x$$

$$I = \int_0^1 dx (1 - x^2)\frac{\cos\frac{\pi}{2}x}{1 - x^2}$$

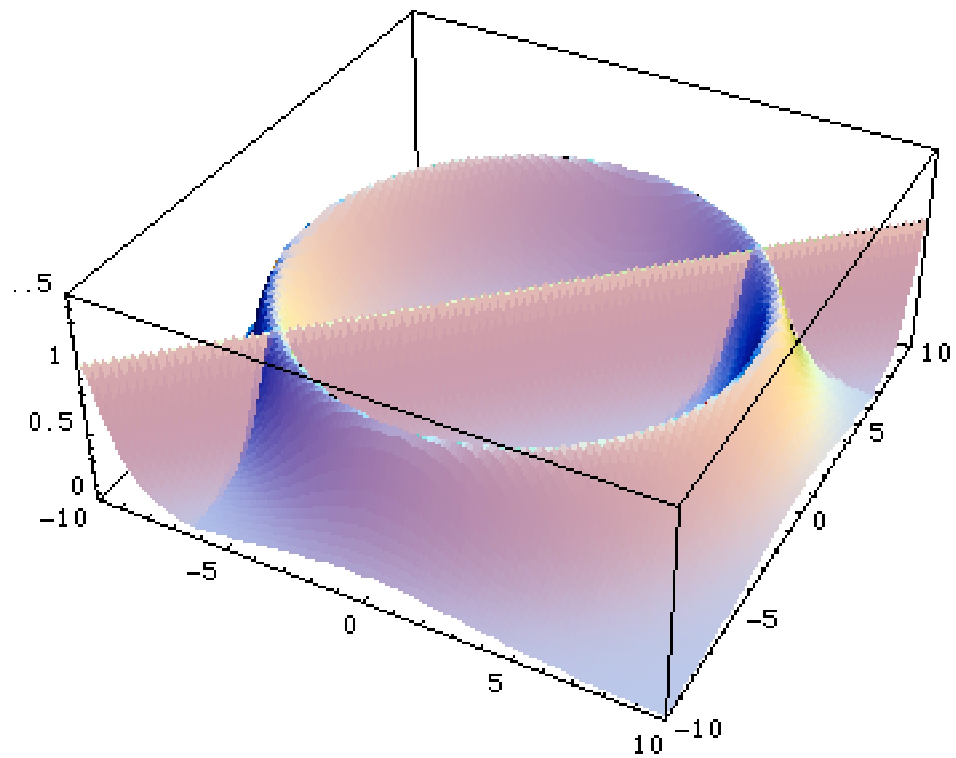
$$= \int_{\xi_1}^{\xi_2} d\xi \frac{\cos\frac{\pi}{2}x[\xi]}{1 - x[\xi]^2} \Rightarrow \simeq 1$$

Multi-channel

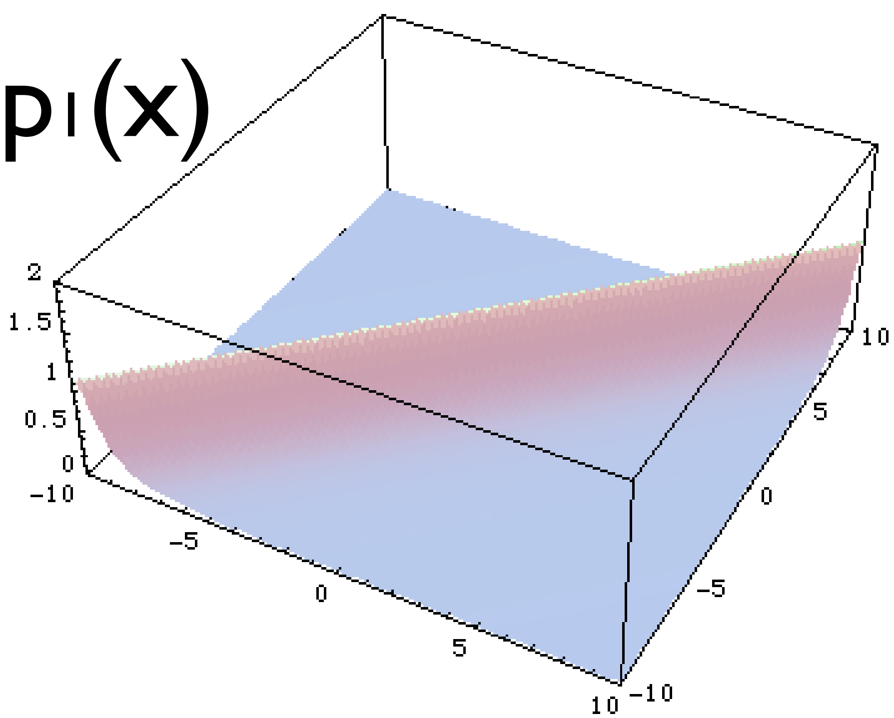


In this case there is no unique tranformation

Multi-channel

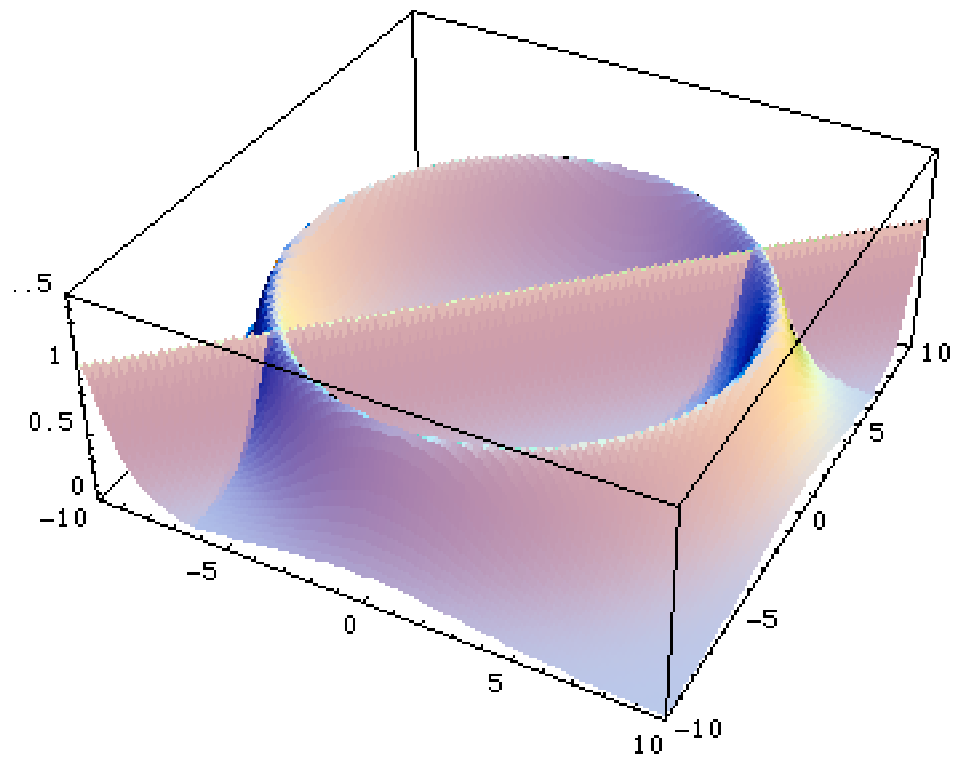


$p_1(x)$



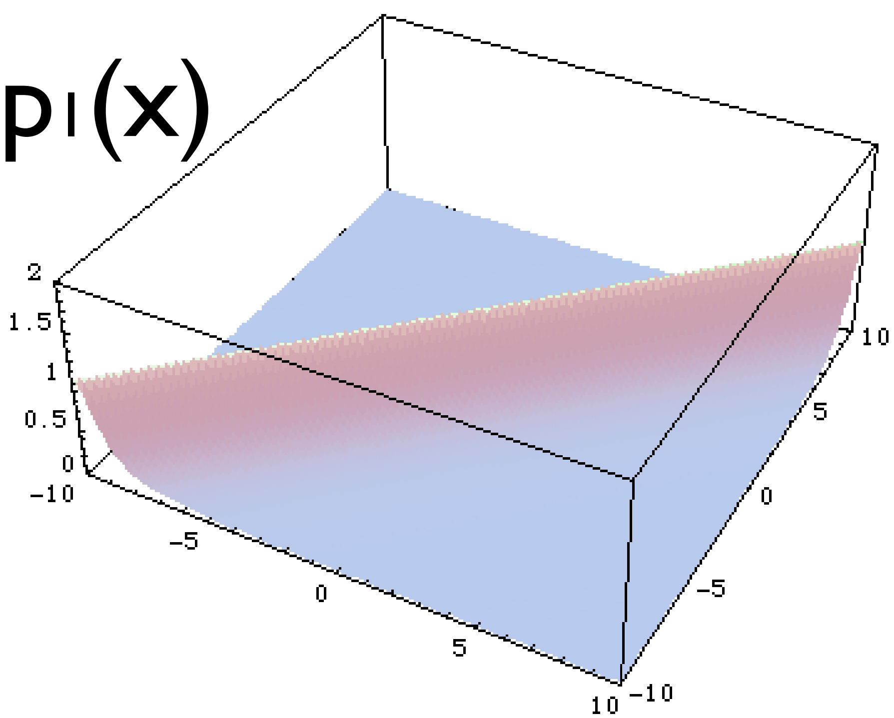
In this case there is no unique tranformation

Multi-channel

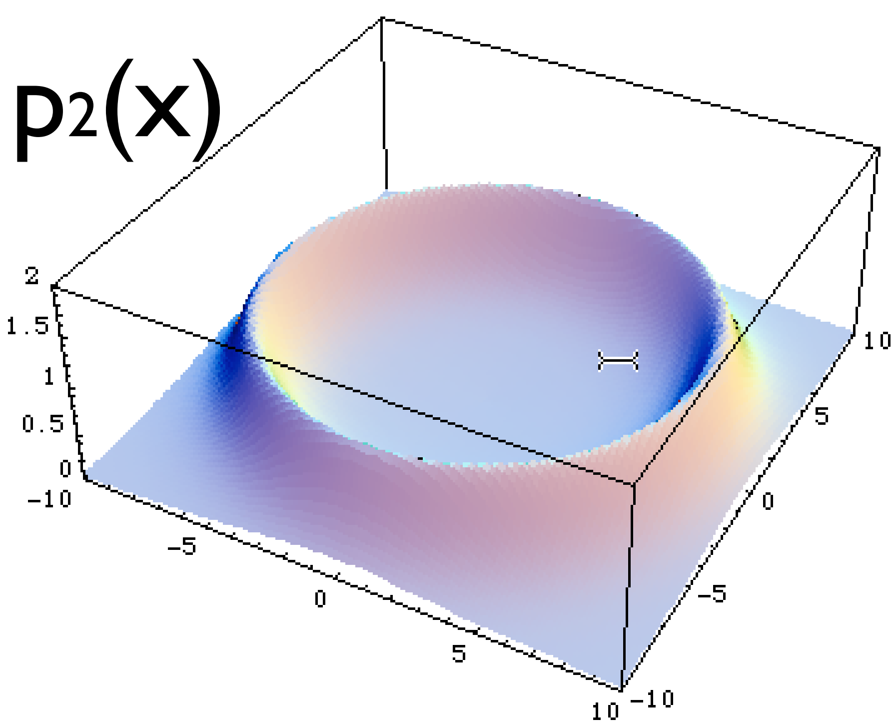


In this case there is no unique tranformation

$p_1(x)$



$p_2(x)$



Multi-channel based on single diagrams*

$$f_i = \frac{|A_i|^2}{\sum_i |A_i|^2} |A_{\text{tot}}|^2$$

- Key Idea
  - Any single diagram is "easy" to integrate (pole structures known based on propagators)
  - Divide integration into pieces, based on diagrams
- Get N independent integrals
  - Errors add in quadrature so no extra cost
  - No need to calculate "weight" function from other channels.
  - Can optimize # of points for each one independently
  - Parallel in nature

Multi-channel based on single diagrams*

$$f_i = \frac{|A_i|^2}{\sum_i |A_i|^2} |A_{\text{tot}}|^2$$

- Key Idea
  - Any single diagram is "easy" to integrate (pole structures known based on propagators)
  - Divide integration into pieces, based on diagrams
- Get N independent integrals
  - Errors add in quadrature so no extra cost
  - No need to calculate "weight" function from other channels.
  - Can optimize # of points for each one independently
  - Parallel in nature

*Method used in MadGraph

Multi-channel based on single diagrams*

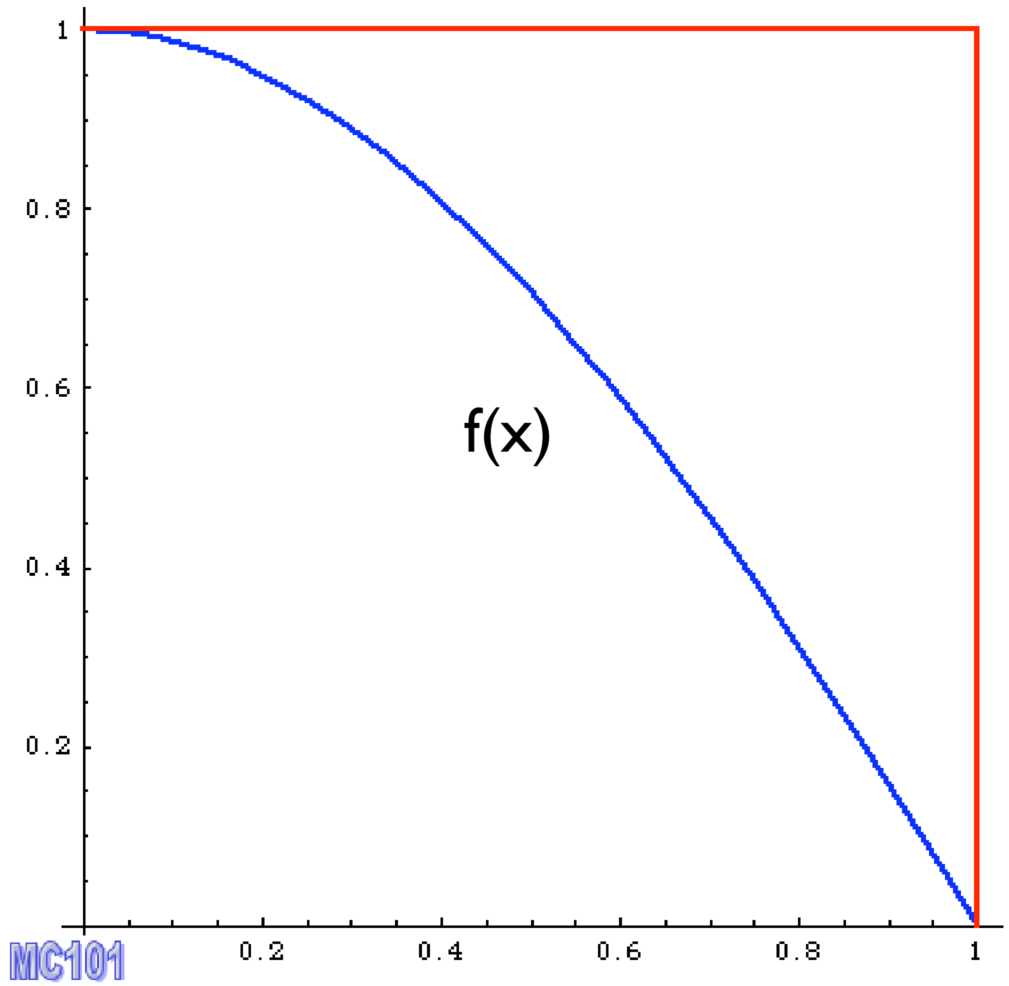
$$f_i = \frac{|A_i|^2}{\sum_i |A_i|^2} \quad \frac{|A_{tot}|^2}{} \simeq 1$$

- Key Idea
  - Any single diagram is "easy" to integrate (pole structures known based on propagators)
  - Divide integration into pieces, based on diagrams
- Get N independent integrals
  - Errors add in quadrature so no extra cost
  - No need to calculate "weight" function from other channels.
  - Can optimize # of points for each one independently
  - Parallel in nature

*Method used in MadGraph

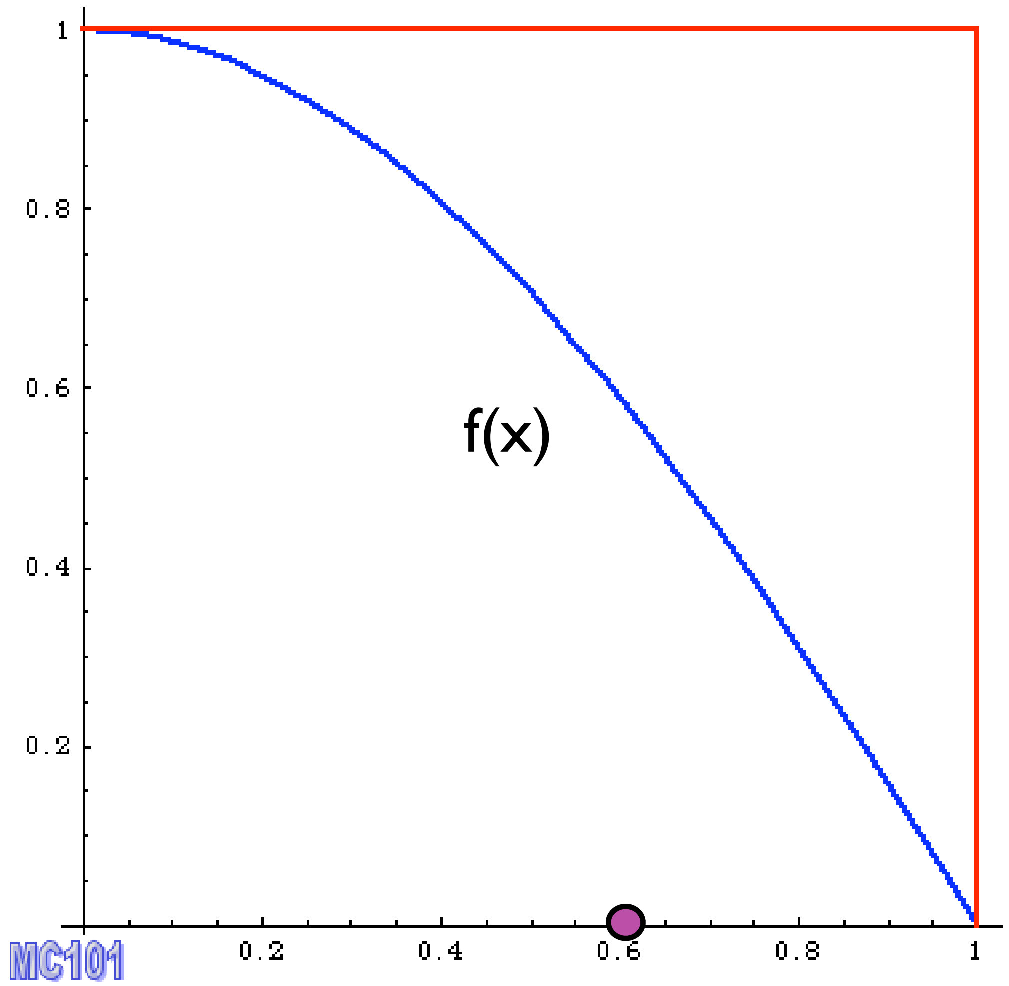
Event Generation

Event generation



Alternative way

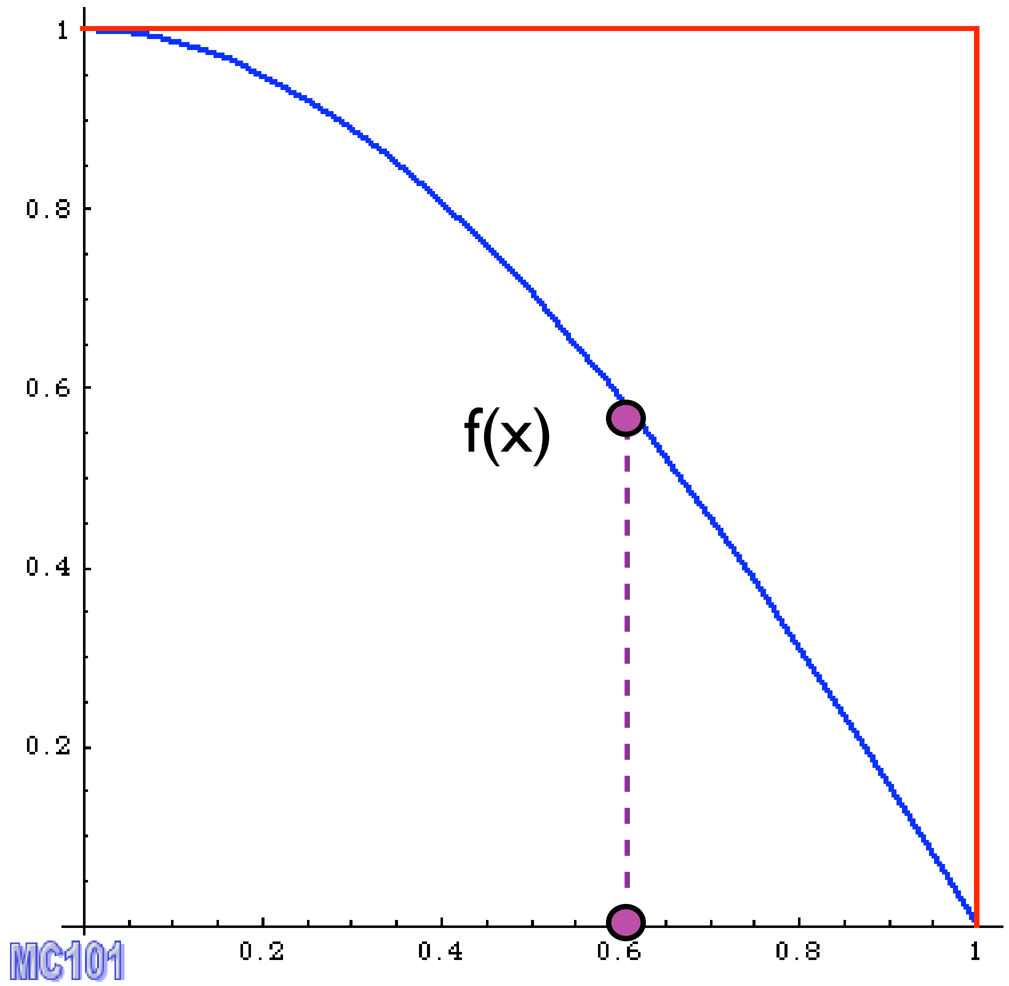
Event generation



Alternative way

1. pick x

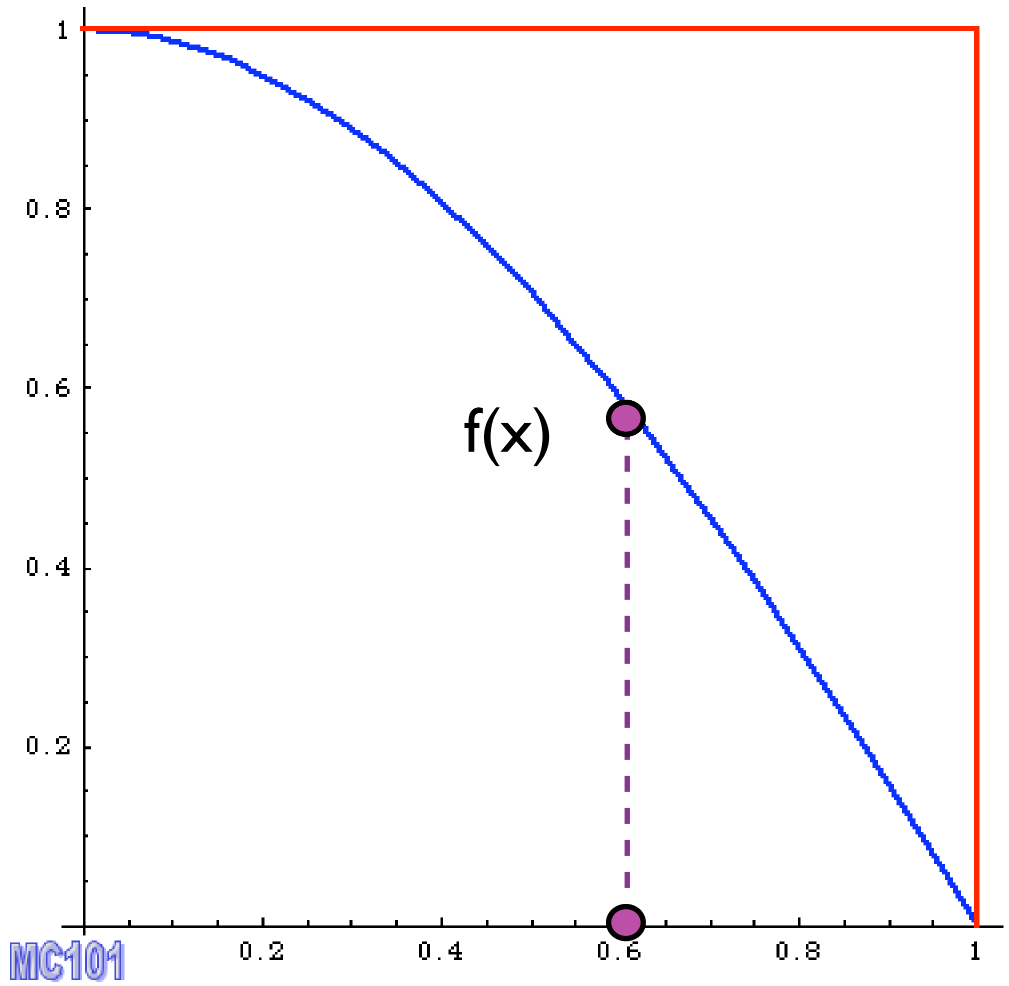
Event generation



Alternative way

1. pick x
2. calculate f(x)

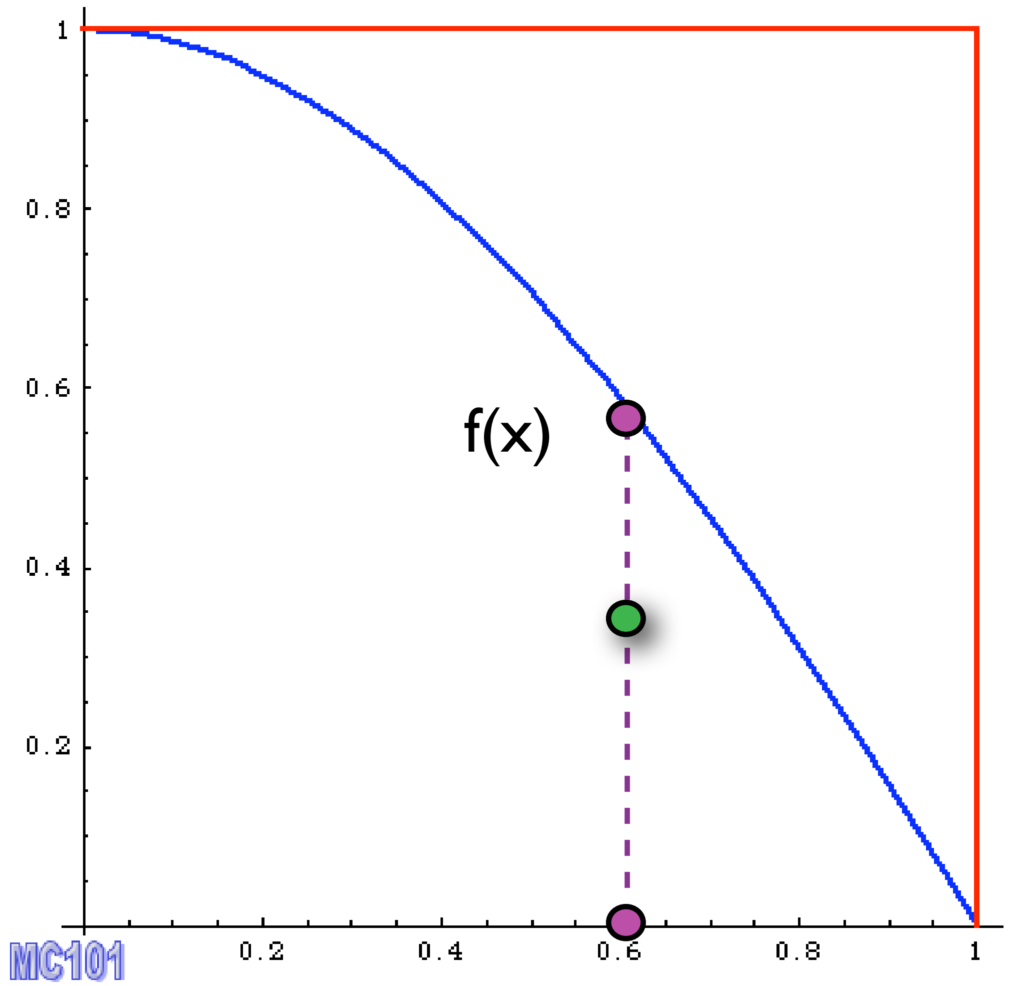
Event generation



Alternative way

1. pick x
2. calculate f(x)
3. pick 0<y<fmax

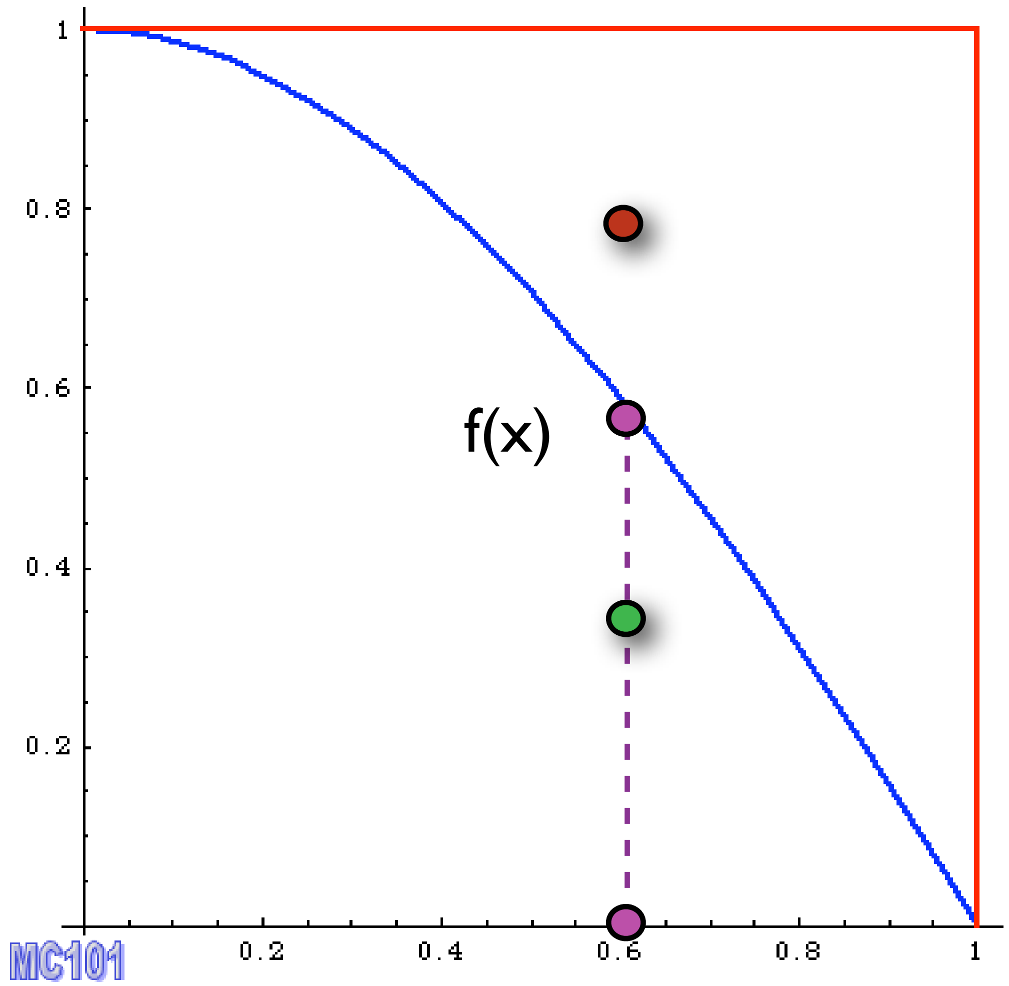
Event generation



Alternative way

1. pick x
2. calculate f(x)
3. pick 0<y<fmax
4. Compare:
   if f(x)>y accept event,

Event generation



Alternative way

1. pick x

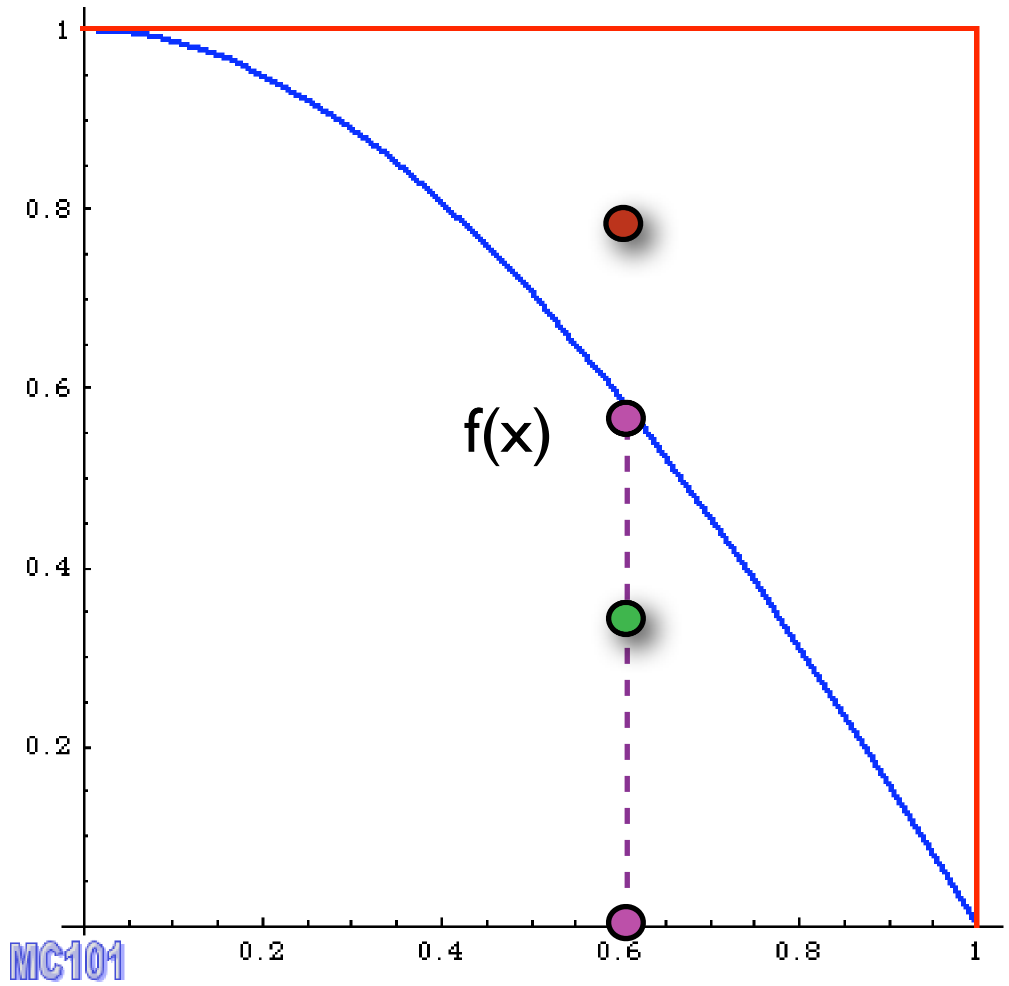
2. calculate f(x)

3. pick 0<y<fmax

4. Compare:
   if f(x)>y accept event,

   else reject it.

Event generation



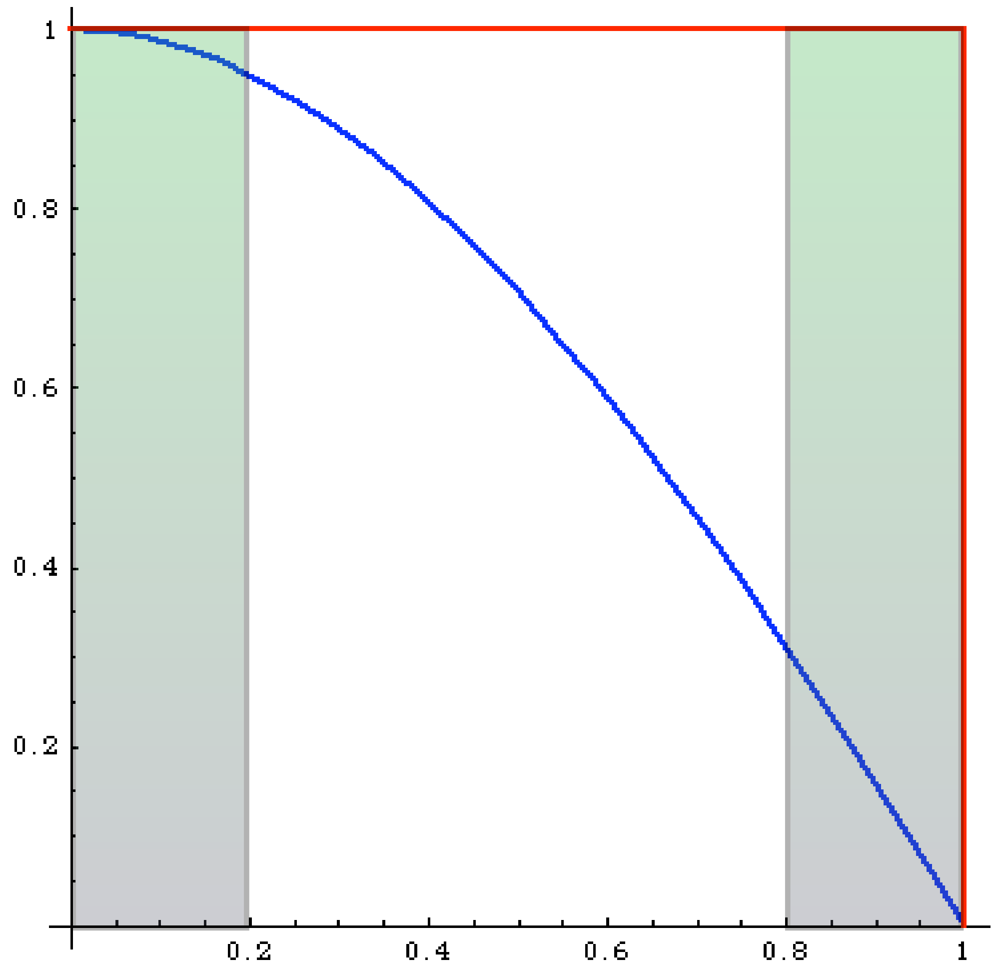
Alternative way

1. pick x
2. calculate f(x)
3. pick 0<y<fmax
4. Compare:
   if f(x)>y accept event,

   else reject it.

$$I = \frac{\text{accepted}}{\text{total tries}} = \text{efficiency}$$

Event generation

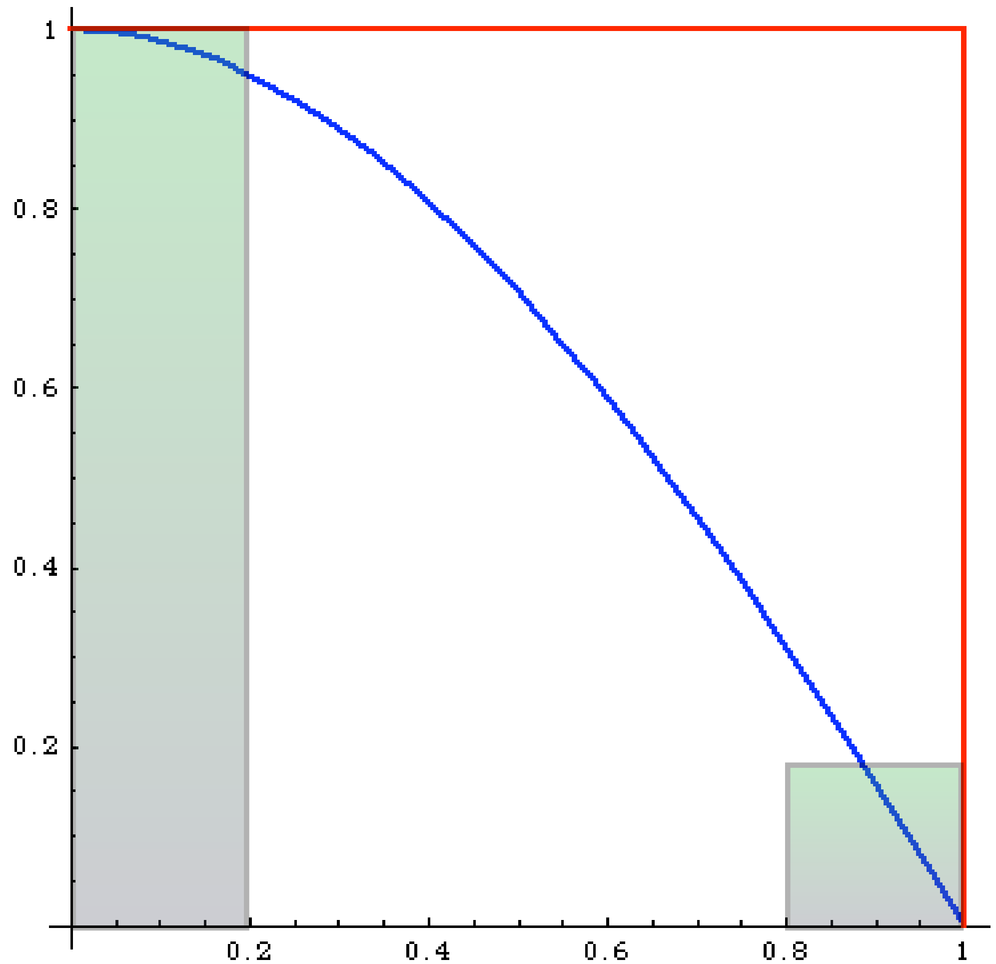


What's the difference?

before:

same # of events in areas of phase space with very different probabilities: events must have different weights

Event generation



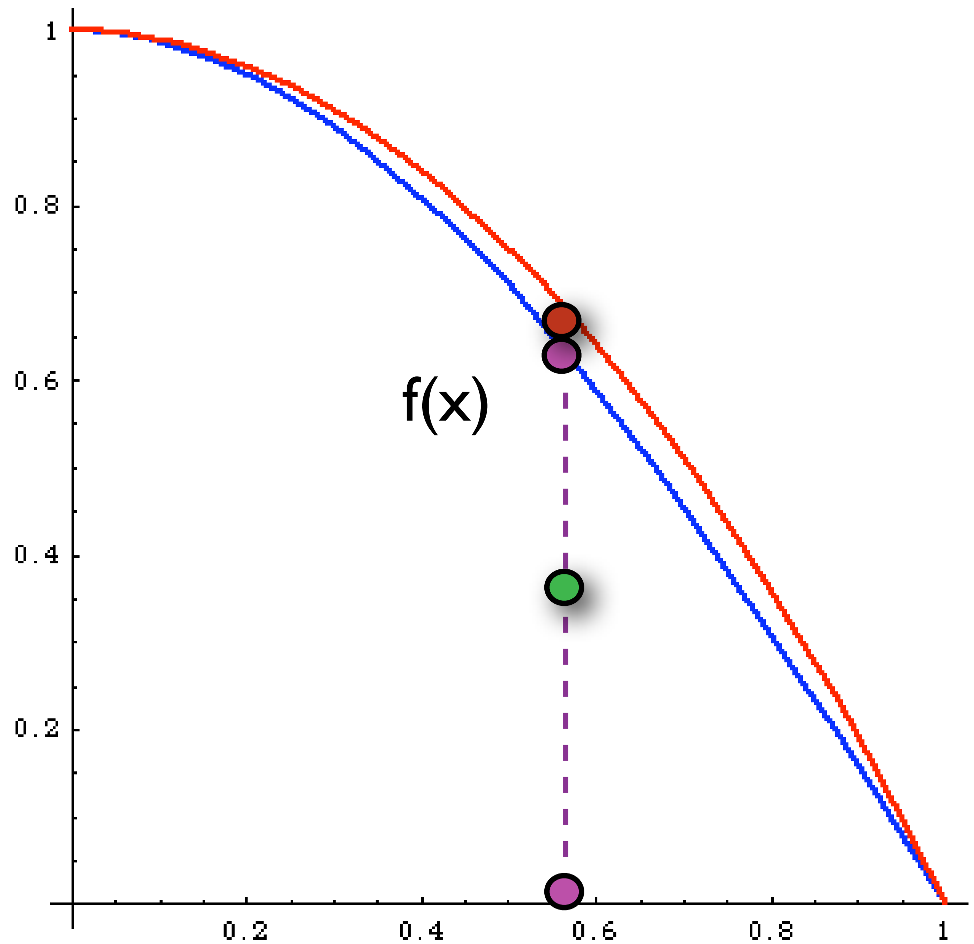
What's the difference?

after:

events is proportional to the probability of areas of phase space:
events have all the same weight ("unweighted")

Events distributed as in Nature

Event generation



Improved

1. pick x distributed as p(x)
2. calculate f(x) and p(x)
3. pick 0<y<1
4. Compare:
   if f(x)>y p(x) accept event,

else reject it.

much better efficiency!!!

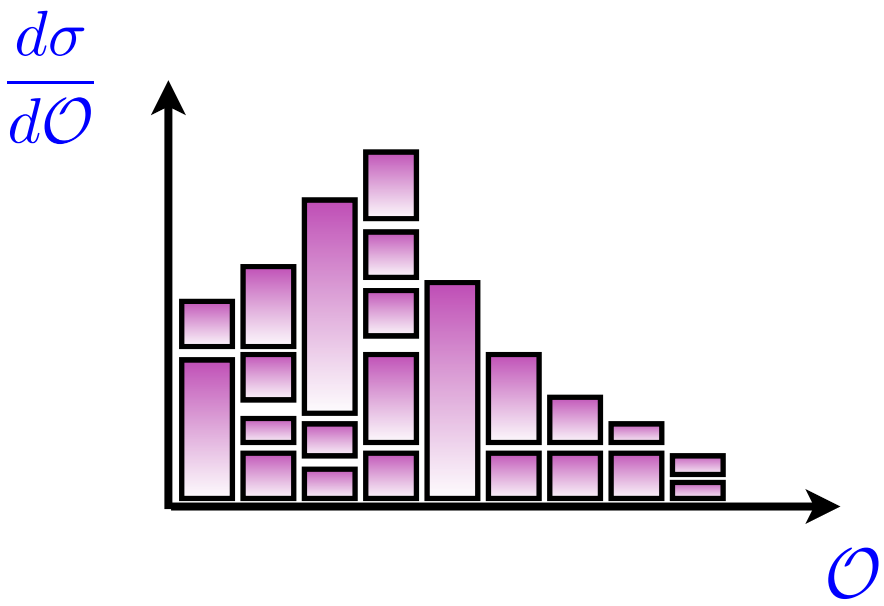
Event generation

Event generation

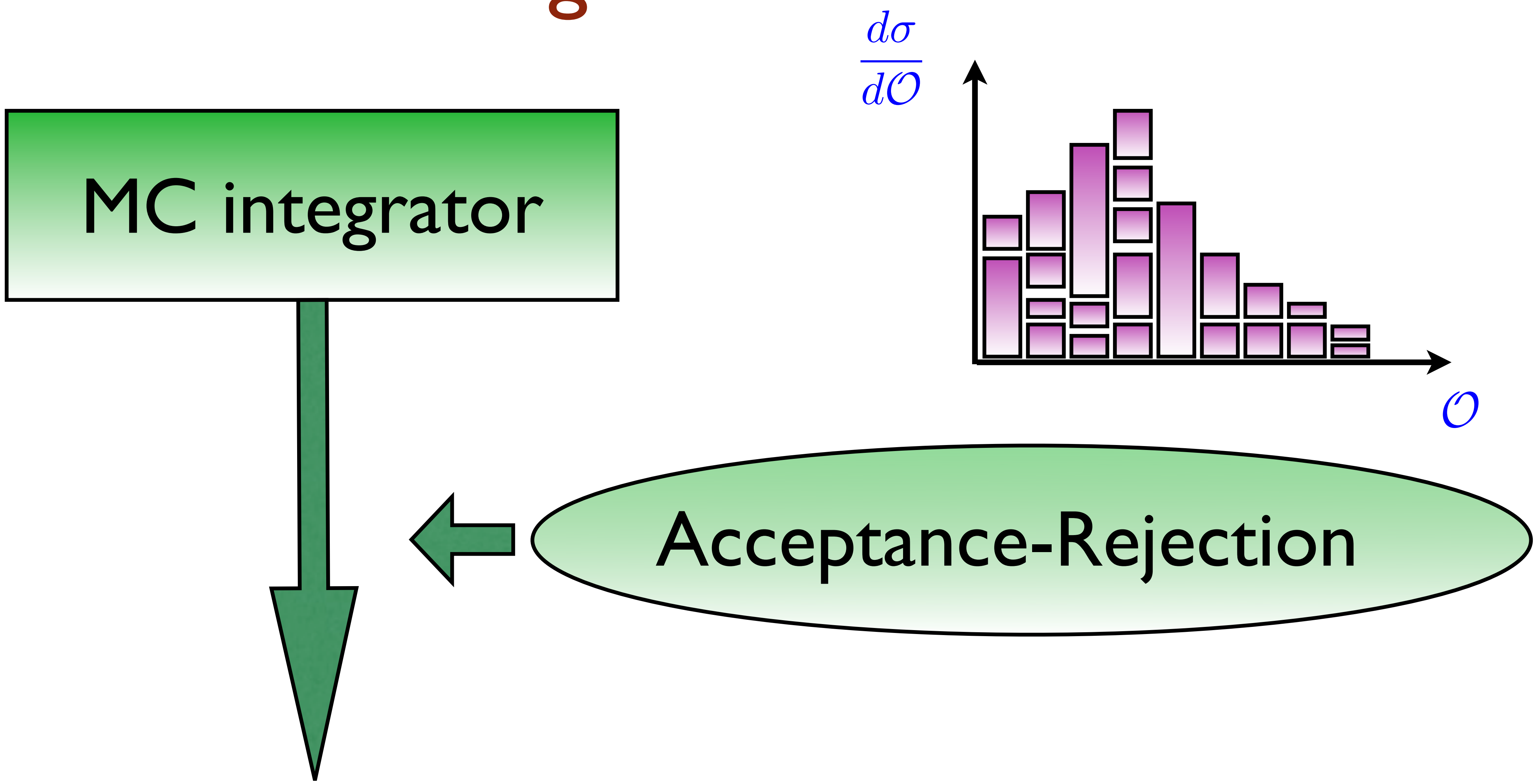
MC integrator

Event generation

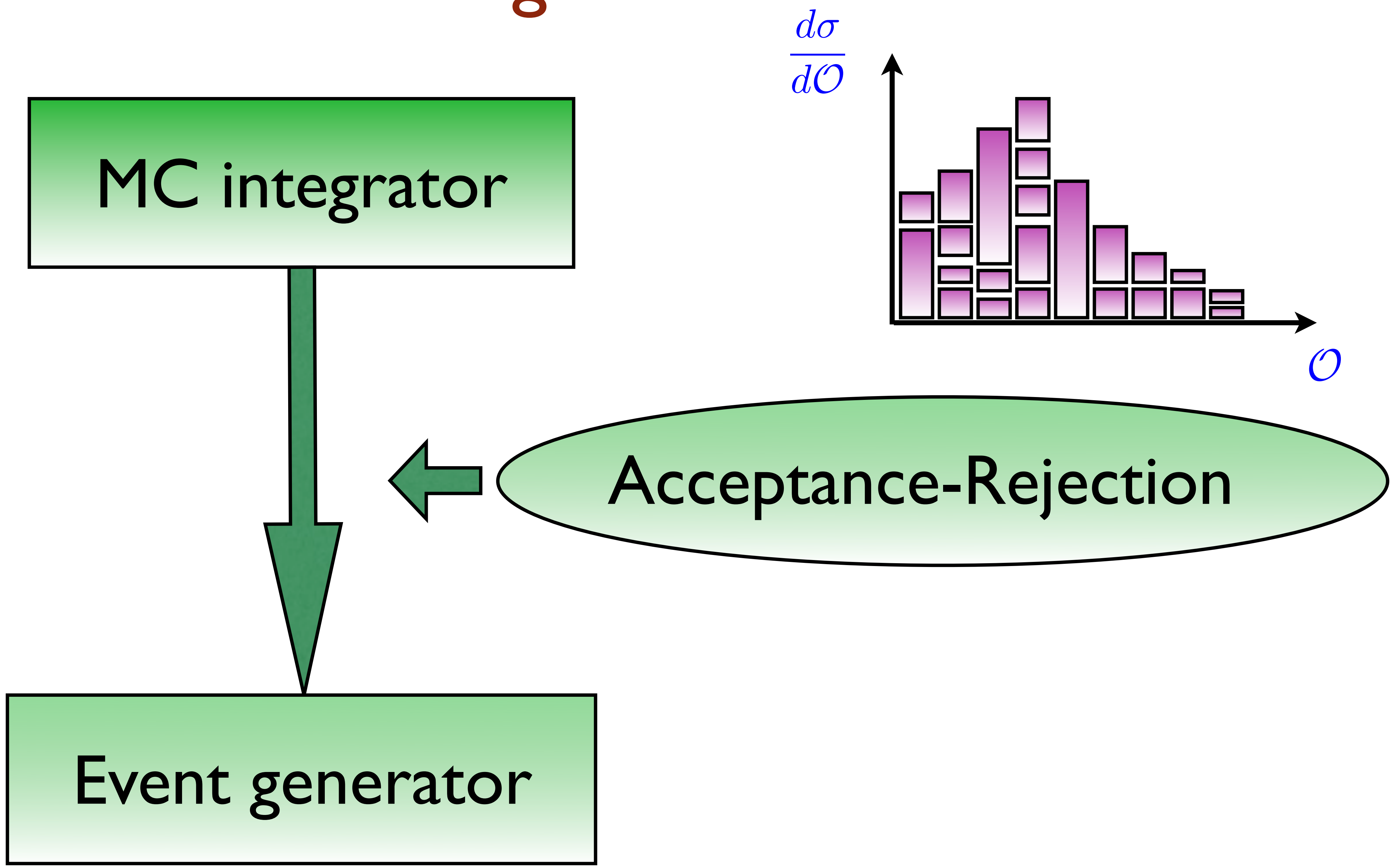
MC integrator



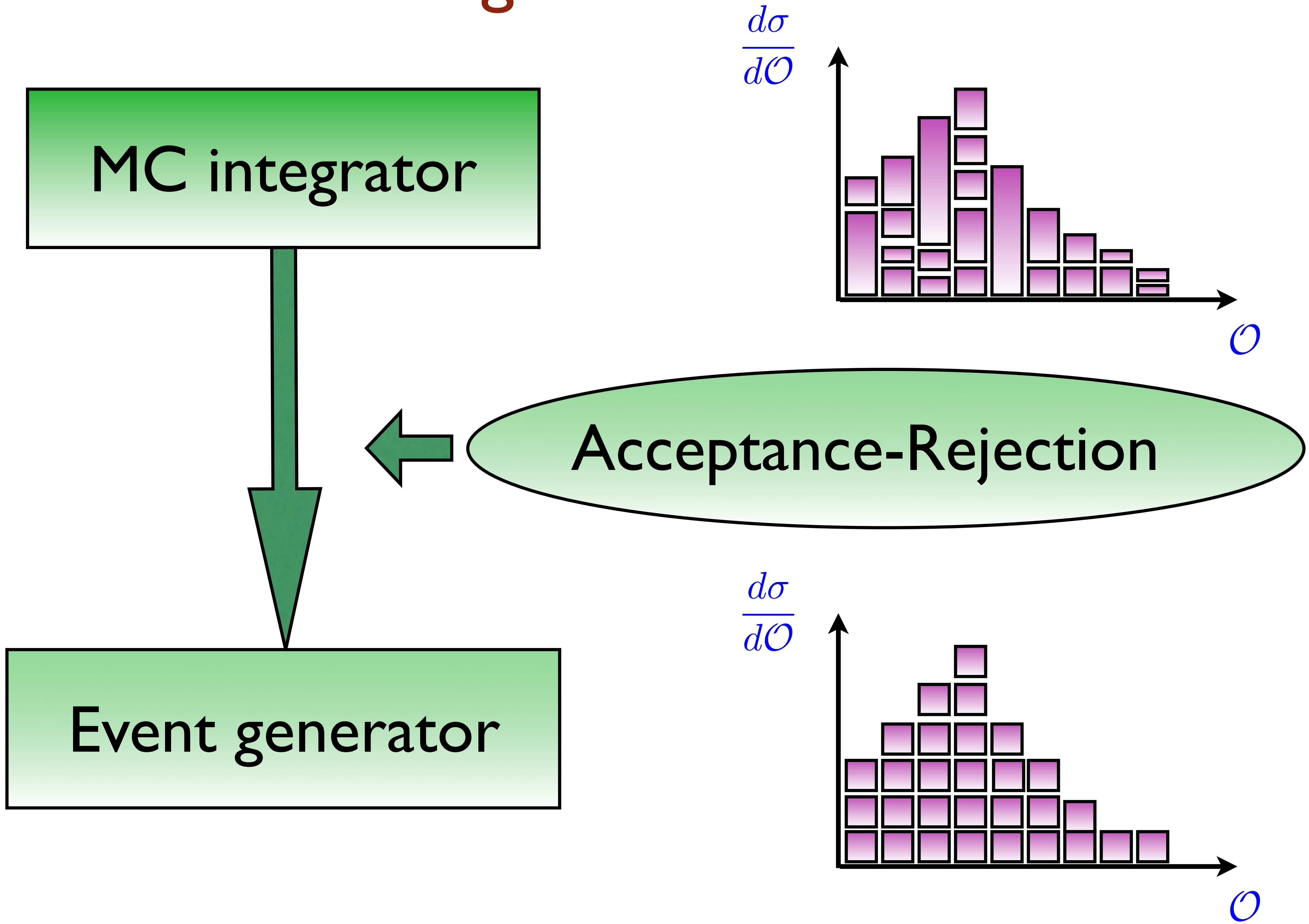
Event generation



Event generation

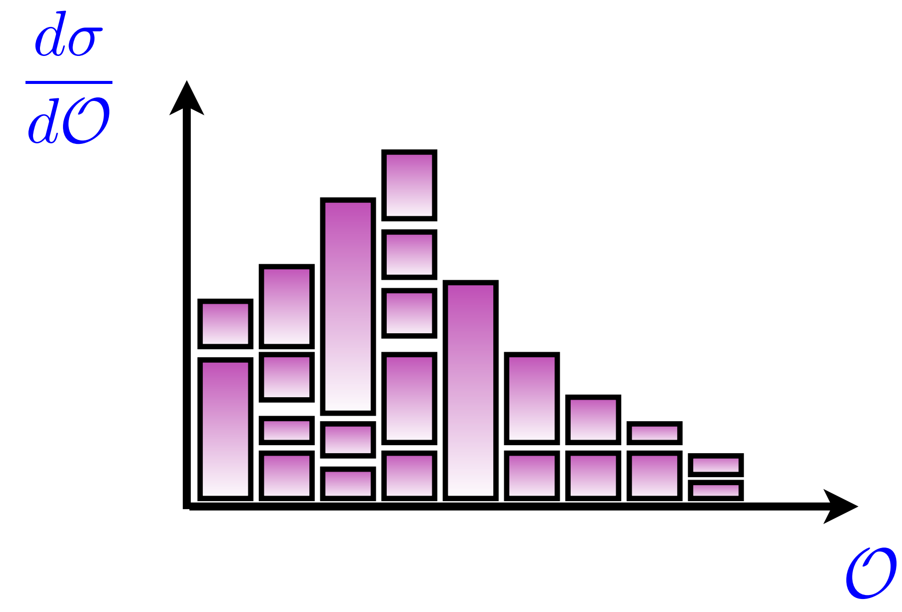


Event generation



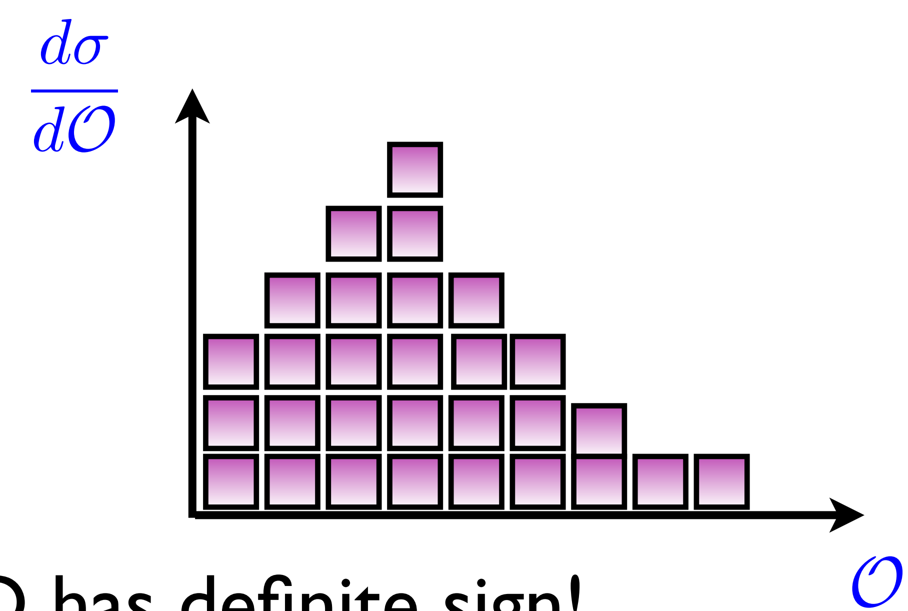
Event generation

MC integrator



Acceptance-Rejection

Event generator



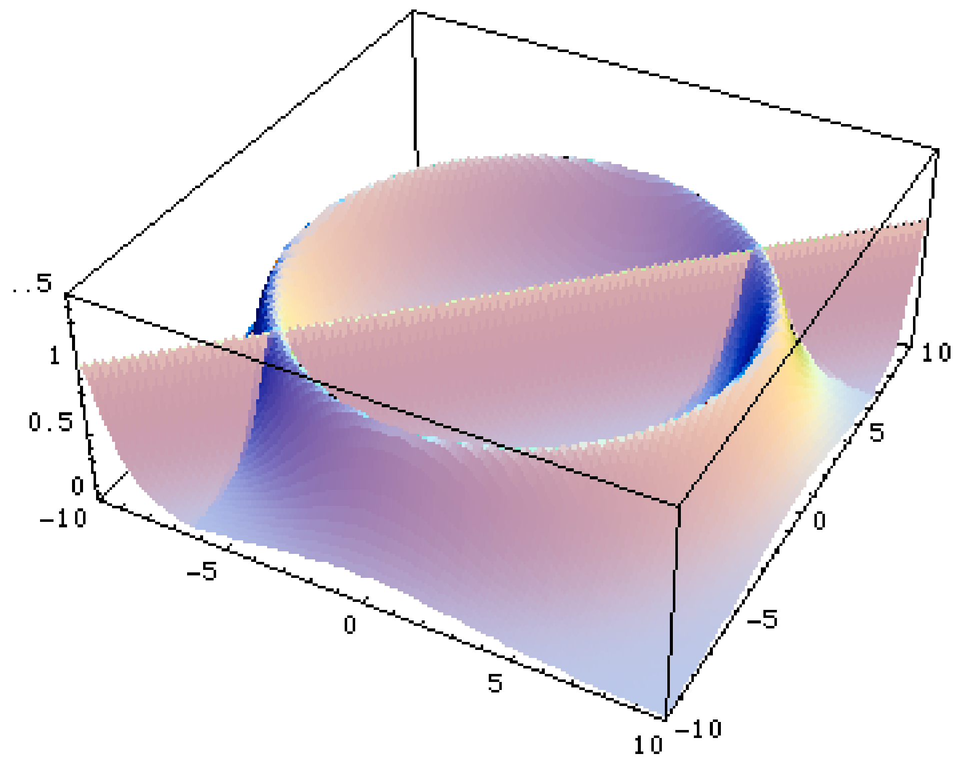
☞ This is possible only if $f(x)<\infty$ AND has definite sign!

Monte Carlo Event Generator: Definiton

At the most basic level a Monte Carlo event generator is a program which produces particle physics events with the same probability as they occur in nature (virtual collider).

In practice it performs a large number of (sometimes very difficult) integrals and then unweights to give the four momenta of the particles that interact with the detector (simulation).

Multi-channel



In this case there is no unique tranformation: Vegas is bound to fail!

If you know where the peaks are (=in which variables) we can use different transformations = channels:

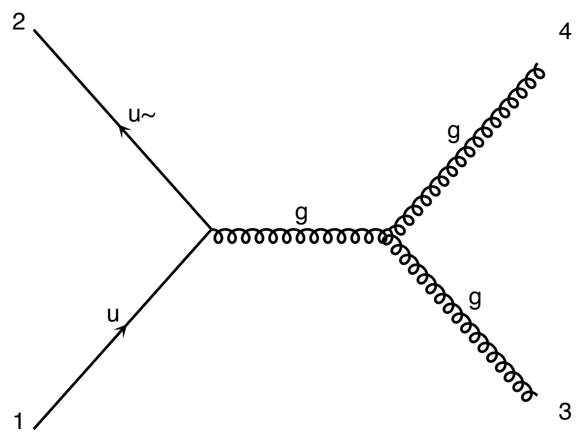
$$p(x) = \sum_{i=1}^{n} \alpha_i p_i(x) \qquad \text{with} \qquad \sum_{i=1}^{n} \alpha_i = 1$$

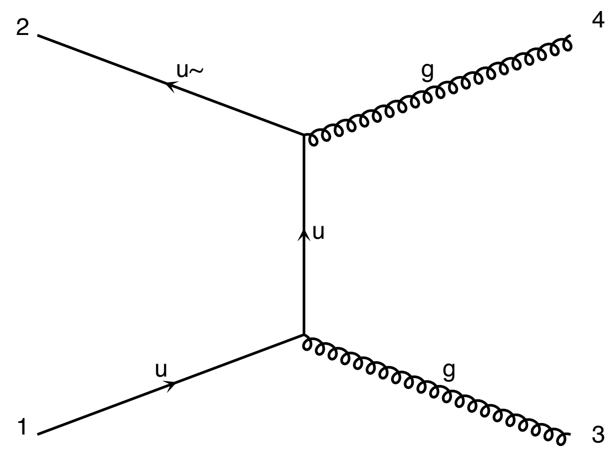
$$I = \int f(x)dx = \sum_{i=1}^{n} \alpha_i \int \frac{f(x)}{p(x)} p_i(x) dx$$

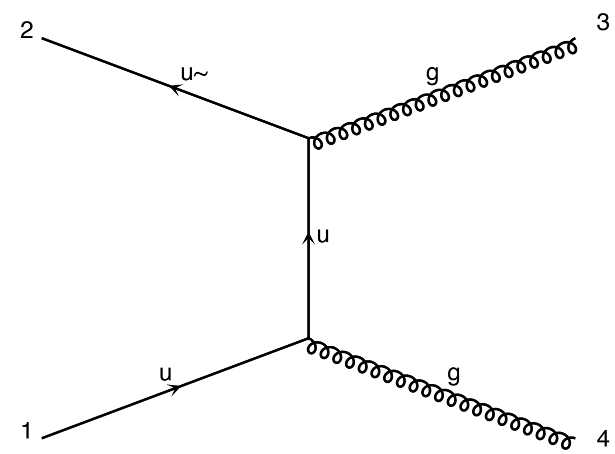
Multi-channel

- Advantages
  - The integral does not depend on the α_i but the variance does and can be minimized by a careful choice
- Drawbacks
  - Need to calculate all g_i values for each point
  - Each phase space channel must be invertible
  - N coupled equations for α_i so it might only work for small number of channels

Example: QCD 2 → 2 production







$$\propto \frac{1}{\hat{s}} = \frac{1}{(p_1 + p_2)^2} \qquad \propto \frac{1}{\hat{t}} = \frac{1}{(p_1 - p_3)^2} \qquad \propto \frac{1}{\hat{u}} = \frac{1}{(p_1 - p_4)^2}$$

Three very different pole structures contributing to the same matrix element.

Multi-channel based on single diagrams

Consider the integration of an amplitude |M|^2 at treel level which many contributing diagrams. If there were a basis of functions,

$$f = \sum_{i=1}^{n} f_i \quad \text{with} \quad f_i \geq 0\,, \quad \forall\ i\,,$$

such that:

1. we know how to integrate each one of them,
2. they describe all possible peaks,

then the problem would be solved:

$$I = \int d\vec{\Phi} f(\vec{\Phi}) = \sum_{i=1}^{n} \int d\vec{\Phi}\, g_i(\vec{\Phi})\, \frac{f_i(\vec{\Phi})}{g_i(\vec{\Phi})} = \sum_{i=1}^{n} I_i\,,$$

Multi-channel based on single diagrams

Consider the integration of an amplitude |M|^2 at treel level which many contributing diagrams. If there were a basis of functions,

$$f = \sum_{i=1}^{n} f_i \quad \text{with} \quad f_i \geq 0\,, \quad \forall \;\, i\,,$$

such that:

1. we know how to integrate each one of them,
2. they describe all possible peaks,

then the problem would be solved:

$$I = \int d\vec{\Phi} f(\vec{\Phi}) = \sum_{i=1}^{n} \int d\vec{\Phi}\, g_i(\vec{\Phi})\, \frac{f_i(\vec{\Phi})}{g_i(\vec{\Phi})} = \sum_{i=1}^{n} I_i\,,$$

Does such a basis exist?

Multi-channel based on single diagrams*

- Key Idea
  - Any single diagram is "easy" to integrate (pole structures known based on propagators)
  - Divide integration into pieces, based on diagrams
- Get N independent integrals
  - Errors add in quadrature so no extra cost
  - No need to calculate "weight" function from other channels.
  - Can optimize # of points for each one independently
  - Parallel in nature
- What about interference?
  - Never creates "new" peaks, so we're OK!

Multi-channel based on single diagrams*

- Key Idea
  - Any single diagram is "easy" to integrate (pole structures known based on propagators)
  - Divide integration into pieces, based on diagrams
- Get N independent integrals
  - Errors add in quadrature so no extra cost
  - No need to calculate "weight" function from other channels.
  - Can optimize # of points for each one independently
  - Parallel in nature
- What about interference?
  - Never creates "new" peaks, so we're OK!

*Method used in MadGraph

Multi-channel based on single diagrams*

YES! $$f_i = \frac{|A_i|^2}{\sum_i |A_i|^2} |A_{\text{tot}}|^2$$

- Key Idea
  - Any single diagram is "easy" to integrate (pole structures known based on propagators)
  - Divide integration into pieces, based on diagrams
- Get N independent integrals
  - Errors add in quadrature so no extra cost
  - No need to calculate "weight" function from other channels.
  - Can optimize # of points for each one independently
  - Parallel in nature
- What about interference?
  - Never creates "new" peaks, so we're OK!

*Method used in MadGraph