UCLouvain
Institut de recherche en mathématique et physique
Centre de Cosmologie, Physique des Particules et Phénoménologie
MCnet

How to accelerate MG5aMC?

Olivier Mattelaer

This Talk

Will not cover

- SIMD/GPU port (-> Andrea's talk)
- Machine Learning effort (-> Ramon's talk)

Will cover

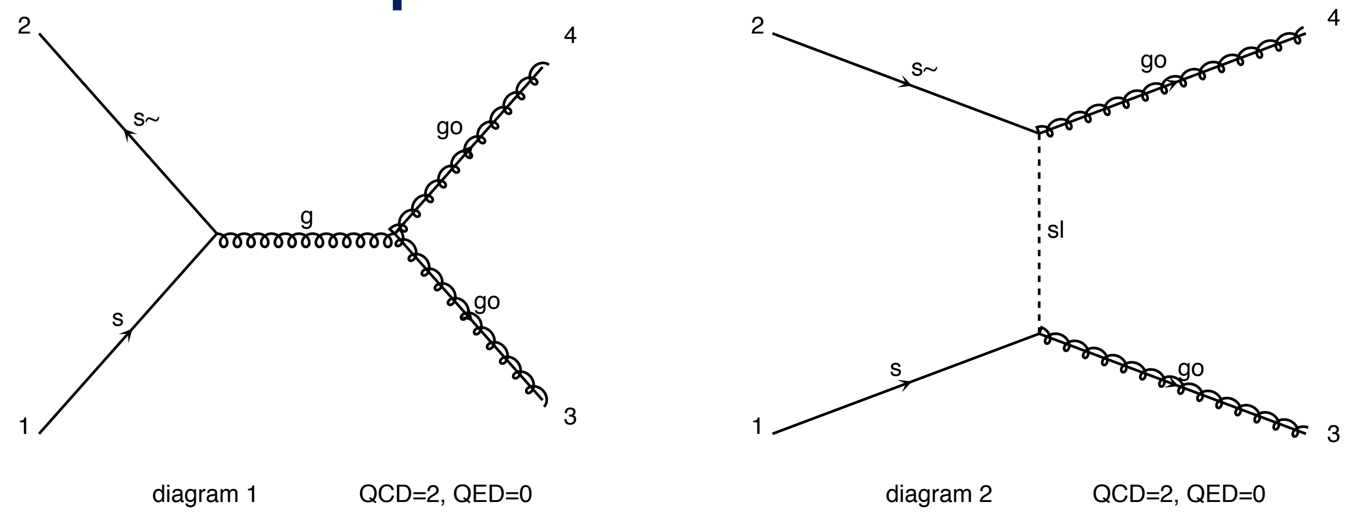
- How computation is done
- Older optimisation:
  - Kiran Ostrelenk, OM: 2102.00773
  - Andrew Lifson, OM: 2210.07267

Thanks to Jenny for some plot and to the full MG5aMC teams for support

Computation step

Calculate a given process (e.g. gluino pair)

- Determine the production mechanism



- Evaluate the matrix-element

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

- Phase-Space Integration

$$\int d\Phi f_1(x_1) f_2(x_2) \, |M|^2$$

- Un-weighting

How to compute the matrix-element

Idea

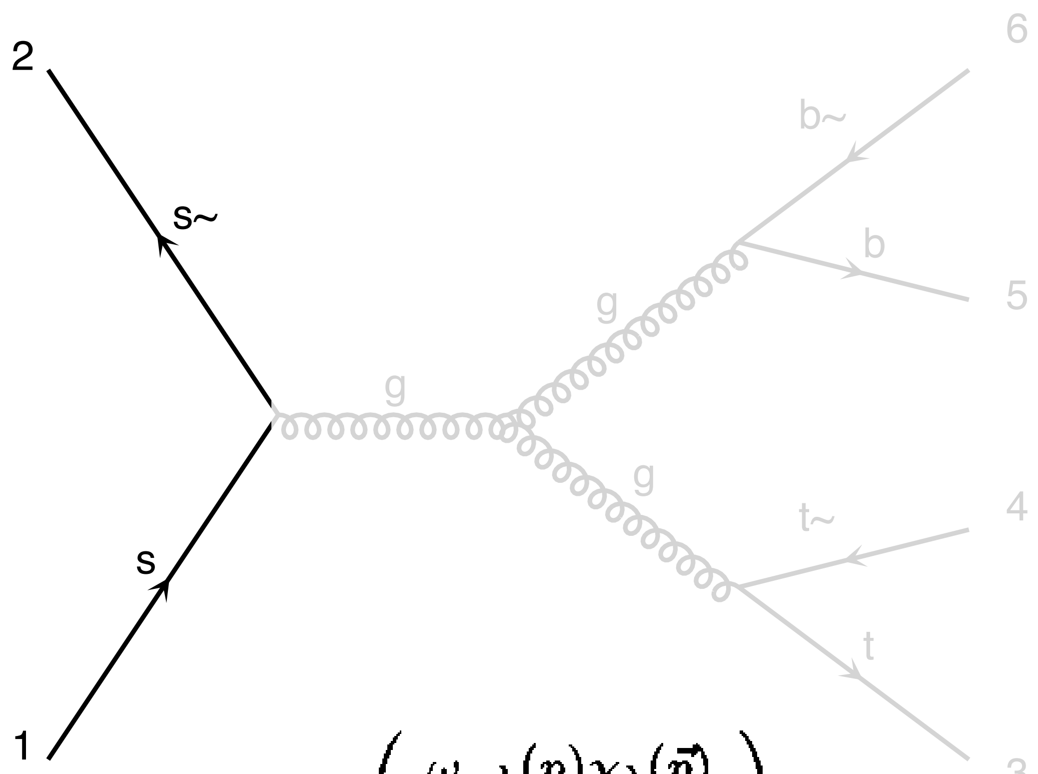
- Evaluate $\mathcal{M}$ for fixed helicity of external particles
  - ➡Multiply $\mathcal{M}$ with $\mathcal{M}^*$ -> $|\mathcal{M}|$^2
  - ➡Loop on Helicity and average the results

How to compute the matrix-element

Idea

- Evaluate $\mathcal{M}$ for fixed helicity of external particles
  - ➡Multiply $\mathcal{M}$ with $\mathcal{M}^*$ -> $|\mathcal{M}|$^2
  - ➡Loop on Helicity and average the results

For one helicity



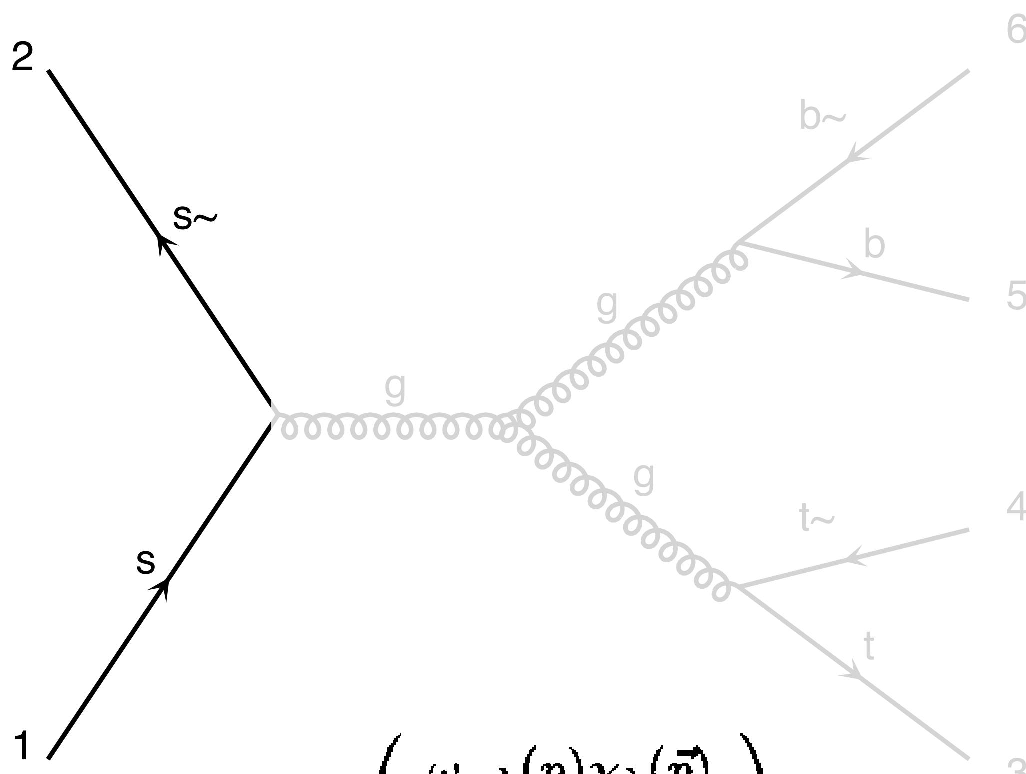
$$u(p) = \begin{pmatrix} \omega_{-\lambda}(p)\chi_\lambda(\vec{p}) \\ \omega_\lambda(p)\chi_\lambda(\vec{p}) \end{pmatrix}$$

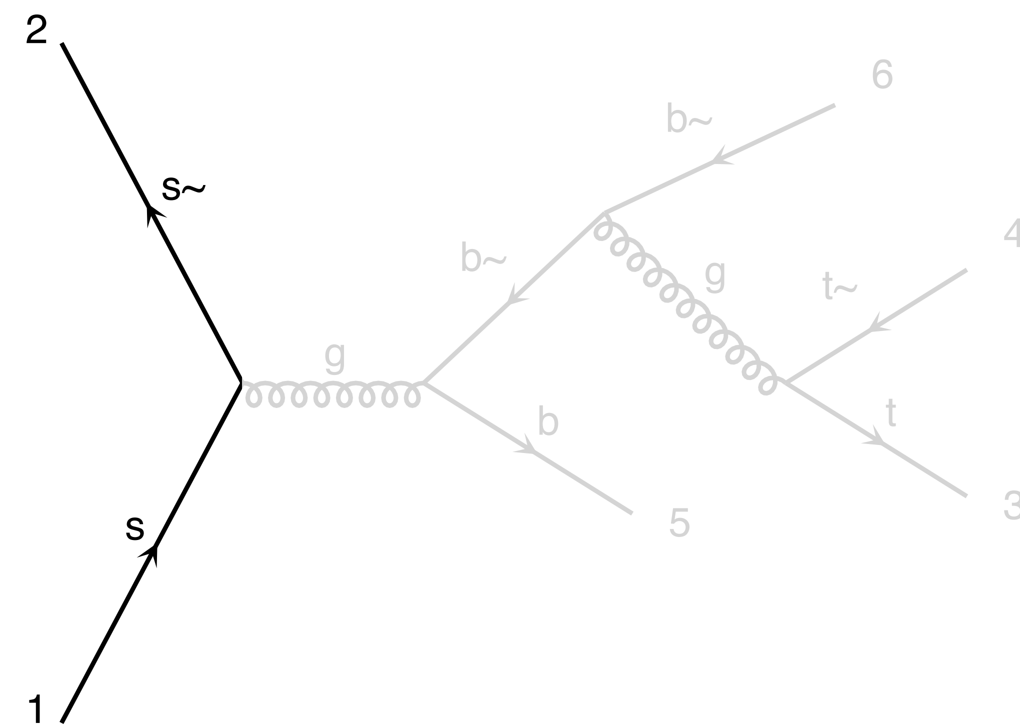
$$\omega_\pm(p) \equiv \sqrt{E \pm |\vec{p}|}.$$

How to compute the matrix-element

Idea
- Evaluate $\mathcal{M}$ for fixed helicity of external particles
  - ➡Multiply $\mathcal{M}$ with $\mathcal{M}^*$ -> $|\mathcal{M}|$^2
  - ➡Loop on Helicity and average the results

For one helicity





$$u(p) = \begin{pmatrix} \omega_{-\lambda}(p)\chi_\lambda(\vec{p}) \\ \omega_\lambda(p)\chi_\lambda(\vec{p}) \end{pmatrix}$$

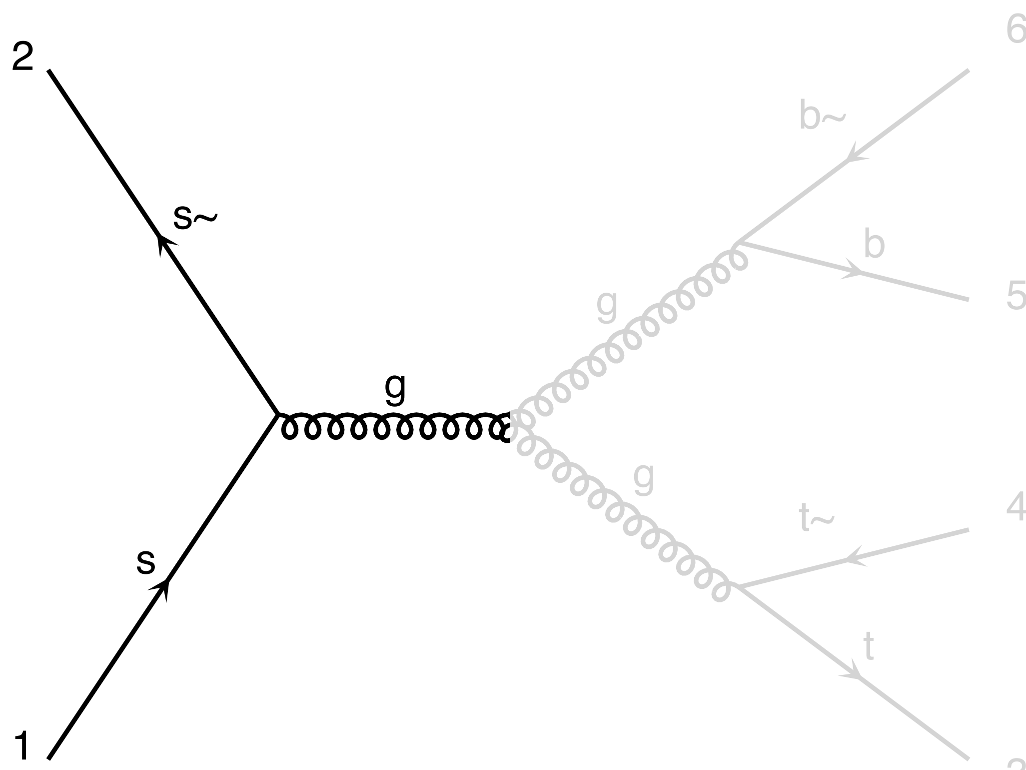
$$\omega_\pm(p) \equiv \sqrt{E \pm |\vec{p}|}.$$

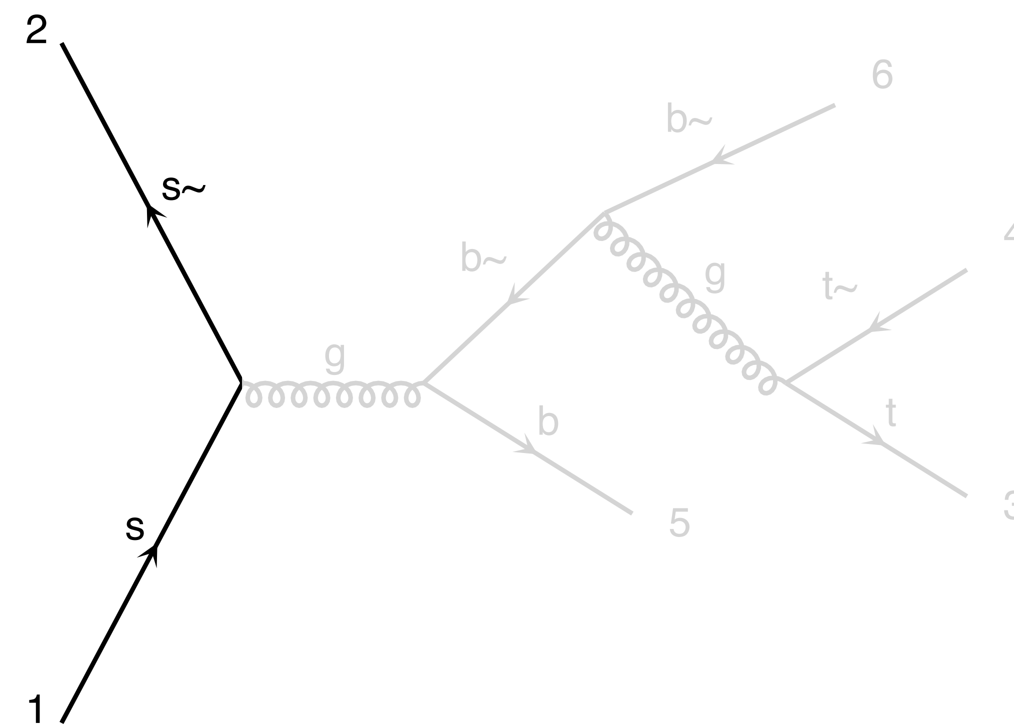
How to compute the matrix-element

Idea

- Evaluate $\mathcal{M}$ for fixed helicity of external particles
  - ➡Multiply $\mathcal{M}$ with $\mathcal{M}^*$ -> $|\mathcal{M}|$^2
  - ➡Loop on Helicity and average the results

For one helicity





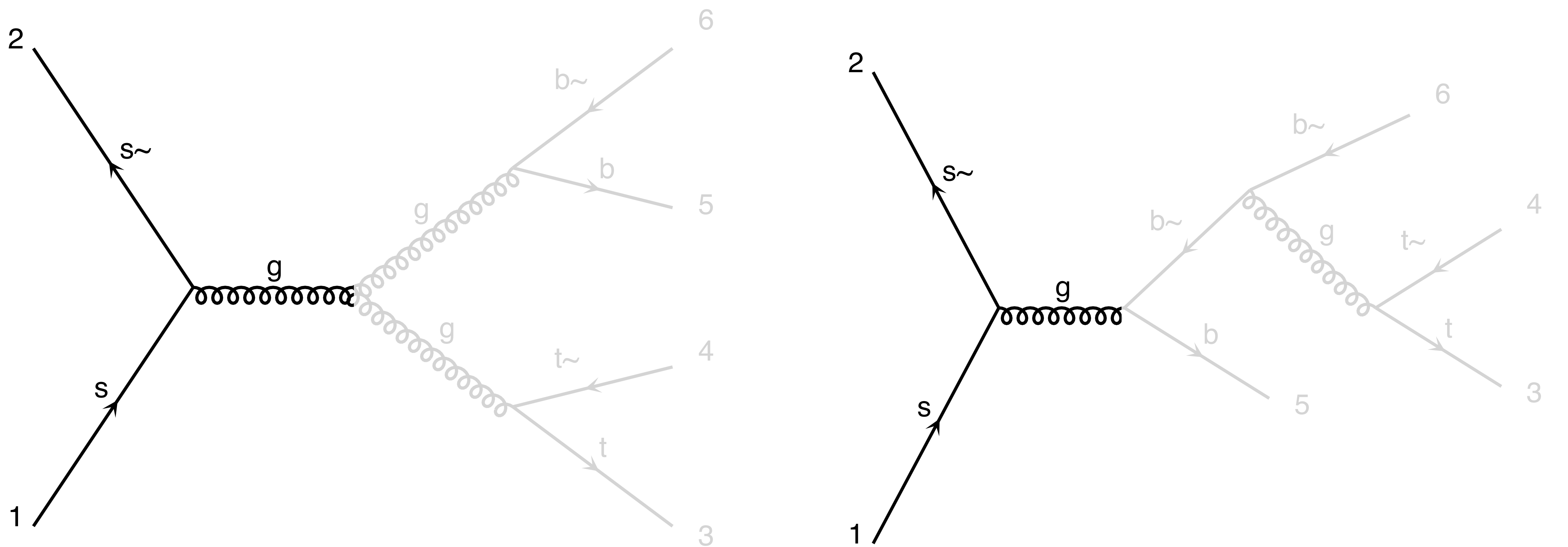
$$W_g = fct(\bar{v}_1, u_2, g_s, m_b, \Gamma_b) = g_s \bar{v}_1 \gamma^\mu u_2 \frac{g_{\mu\nu}}{q^2 - m_g^2 + i m_g \Gamma_g}$$

How to compute the matrix-element

Idea

- Evaluate $\mathcal{M}$ for fixed helicity of external particles
  - ➡Multiply $\mathcal{M}$ with $\mathcal{M}^*$ -> $|\mathcal{M}|$^2
  - ➡Loop on Helicity and average the results

For one helicity

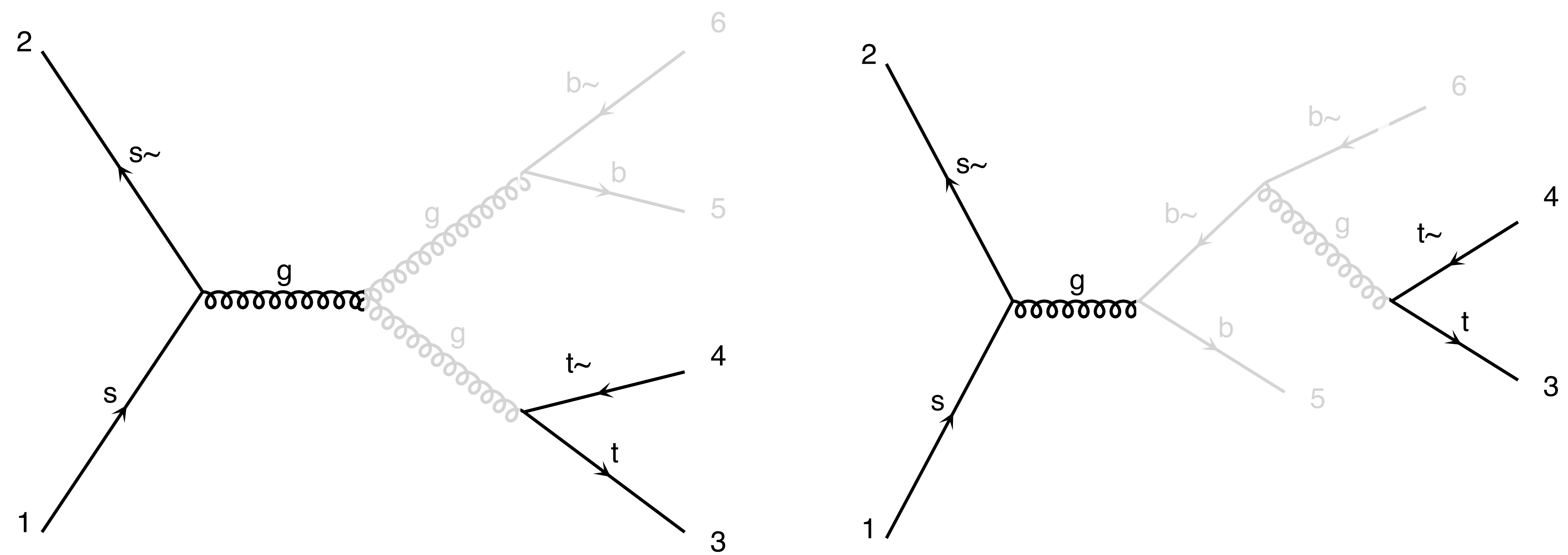


$$W_g = fct(\bar{v}_1, u_2, g_s, m_g, \Gamma_g) = g_s \bar{v}_1 \gamma^\mu u_2 \frac{g_{\mu\nu}}{q^2 - m_g^2 + i m_g \Gamma_g}$$

How to compute the matrix-element

Idea
- Evaluate $\mathcal{M}$ for fixed helicity of external particles
  - ➡Multiply $\mathcal{M}$ with $\mathcal{M}^*$ -> $|\mathcal{M}|$^2
  - ➡Loop on Helicity and average the results

For one helicity

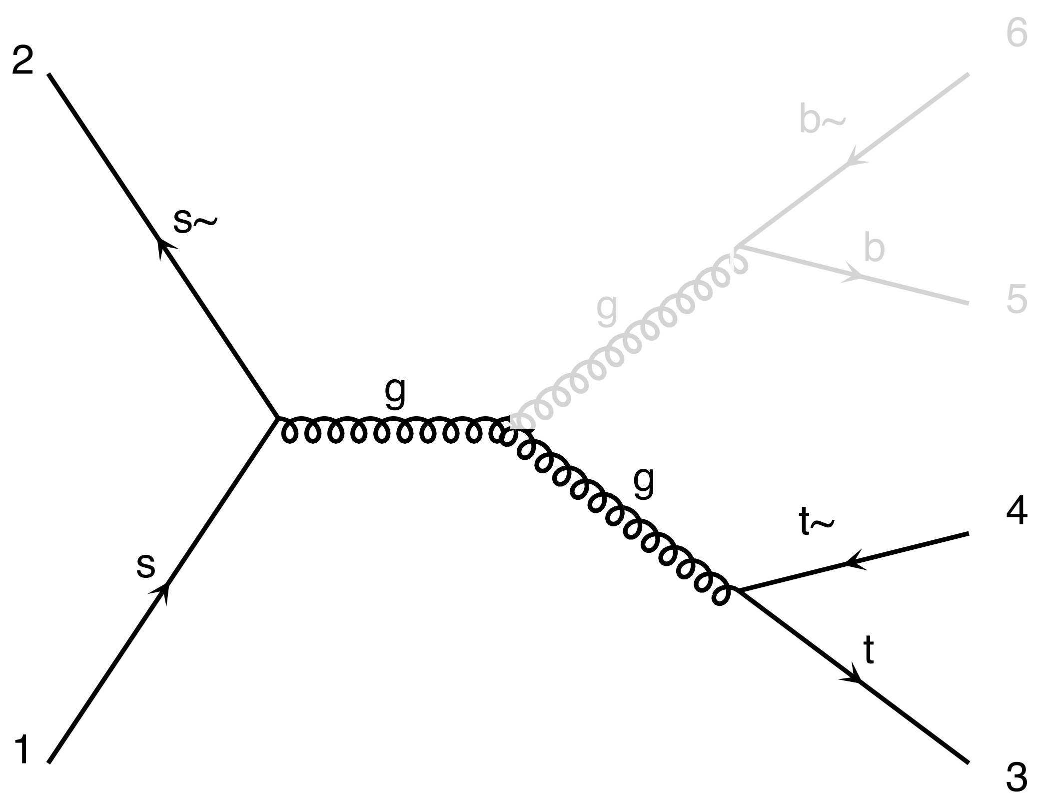


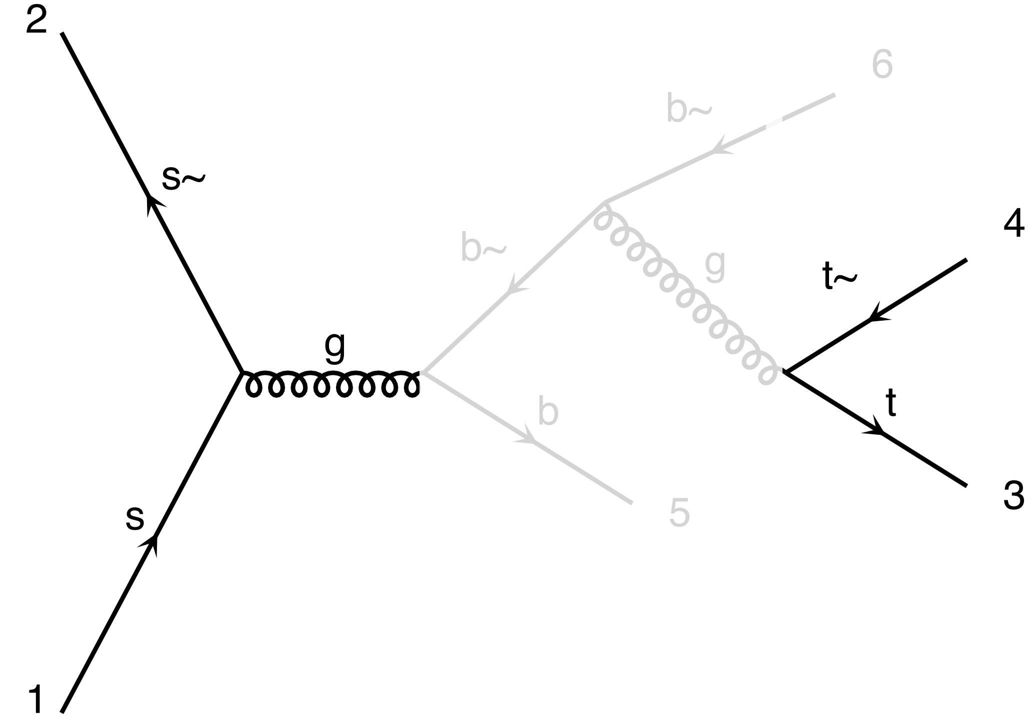
How to compute the matrix-element

Idea

- Evaluate $\mathcal{M}$ for fixed helicity of external particles
  - ➡Multiply $\mathcal{M}$ with $\mathcal{M}^*$ -> $|\mathcal{M}|$^2
  - ➡Loop on Helicity and average the results

For one helicity

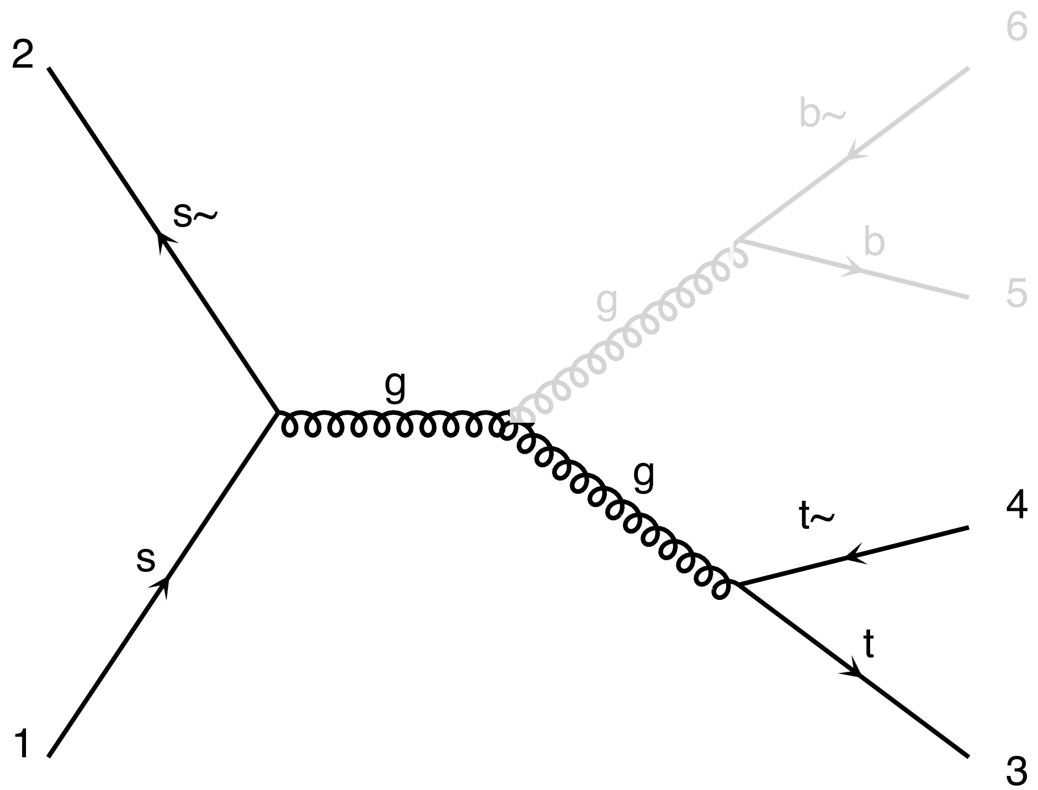


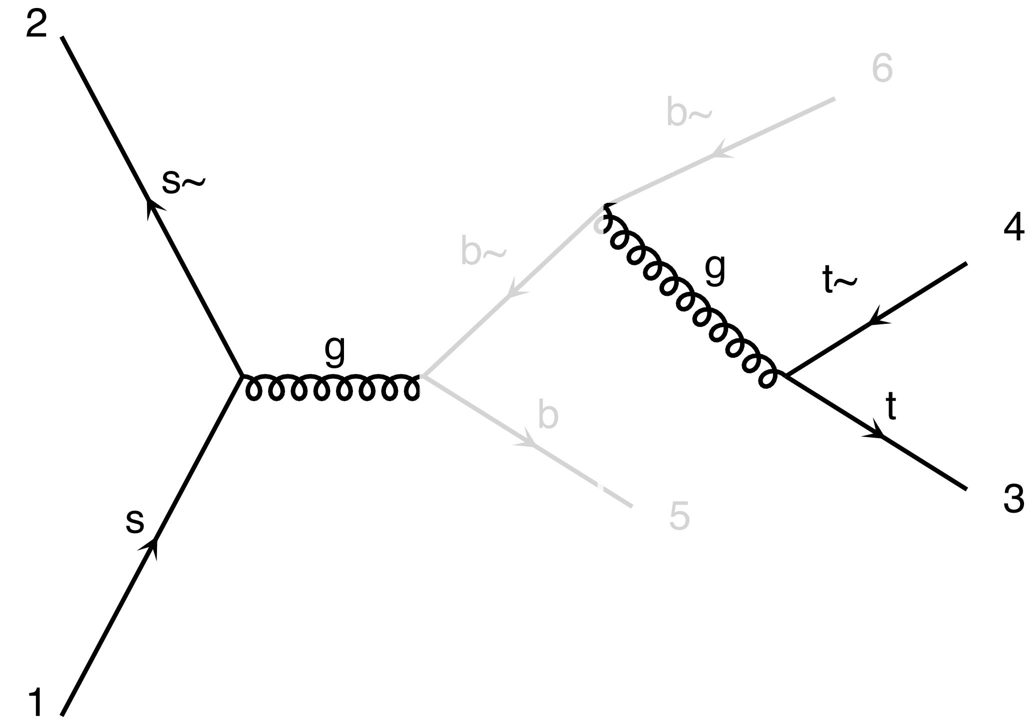


How to compute the matrix-element

Idea
- Evaluate $\mathcal{M}$ for fixed helicity of external particles
  - ➡Multiply $\mathcal{M}$ with $\mathcal{M}^*$ -> $|\mathcal{M}|$^2
  - ➡Loop on Helicity and average the results

For one helicity



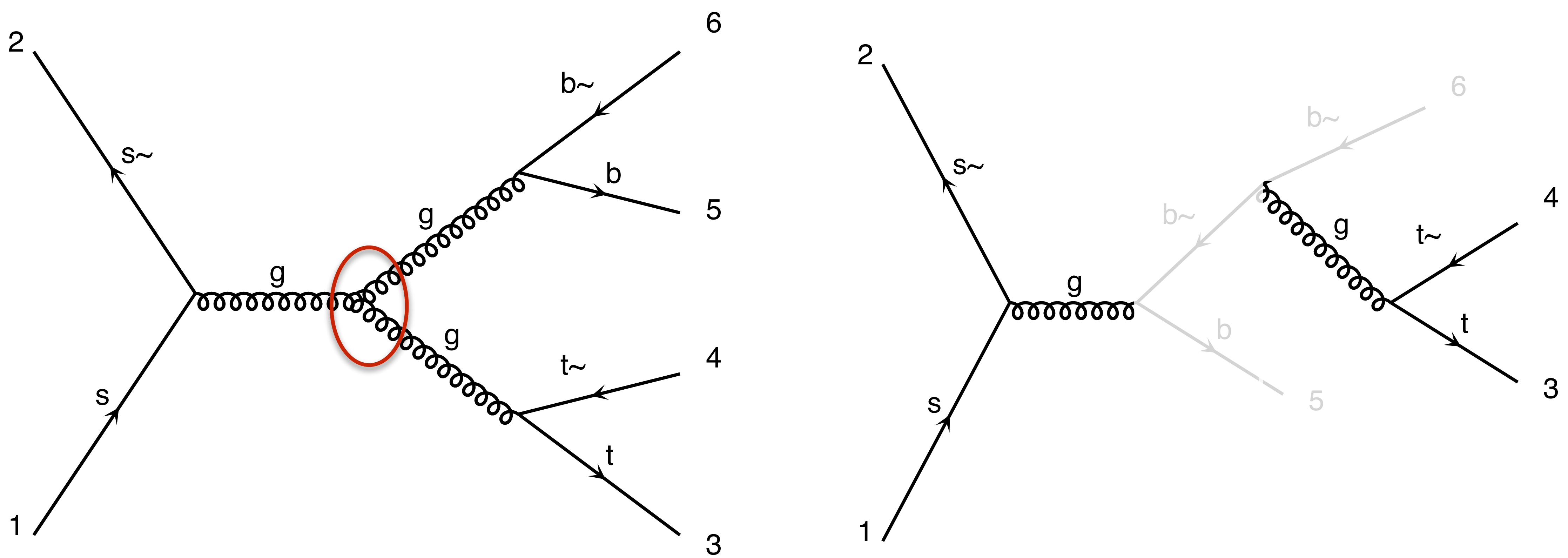


How to compute the matrix-element

Idea

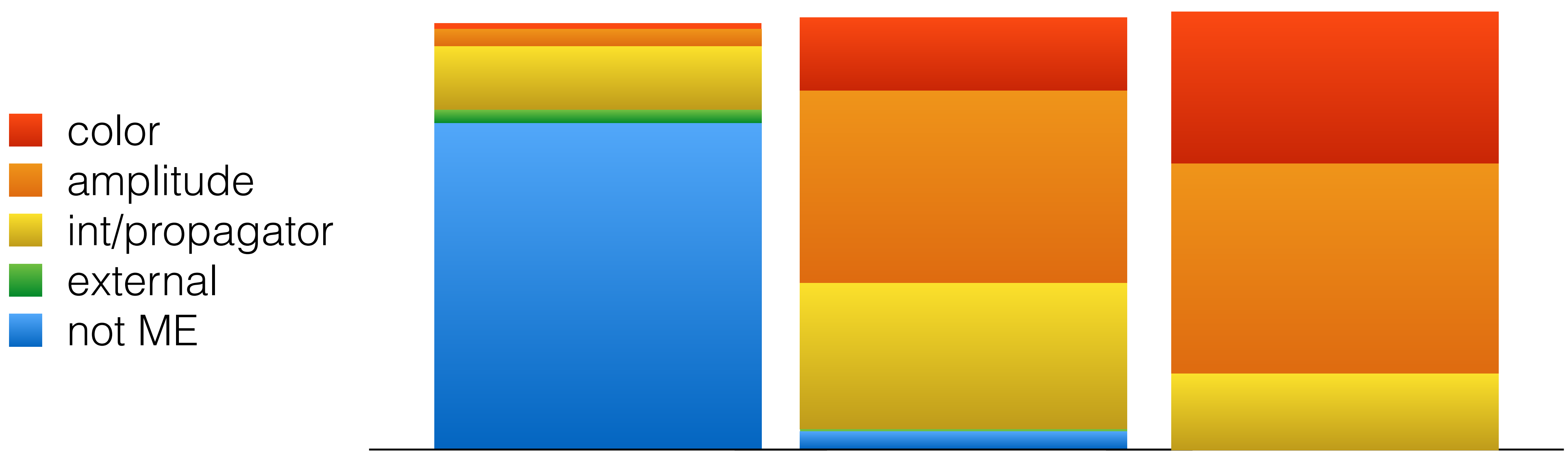
- Evaluate $\mathcal{M}$ for fixed helicity of external particles
  - ➡Multiply $\mathcal{M}$ with $\mathcal{M}^*$ -> $|\mathcal{M}|$^2
  - ➡Loop on Helicity and average the results

For one helicity



Speed status

	$gg \to t\bar{t}$	$gg \to t\bar{t}gg$	$gg \to t\bar{t}ggg$
`madevent`	13G	470G	11T
`matrix1`	3.1G (23%)	450G (96%)	11T (>99%)
↳ ext	450M (3.4%)	3.3G (<1%)	7.3G (<1%)
↳ int	1.9G (14%)	160G (35%)	2T (19%)
↳ amp	530M (4.0%)	210G (44%)	5.5T (51%)



Helicity Amplitude

Idea

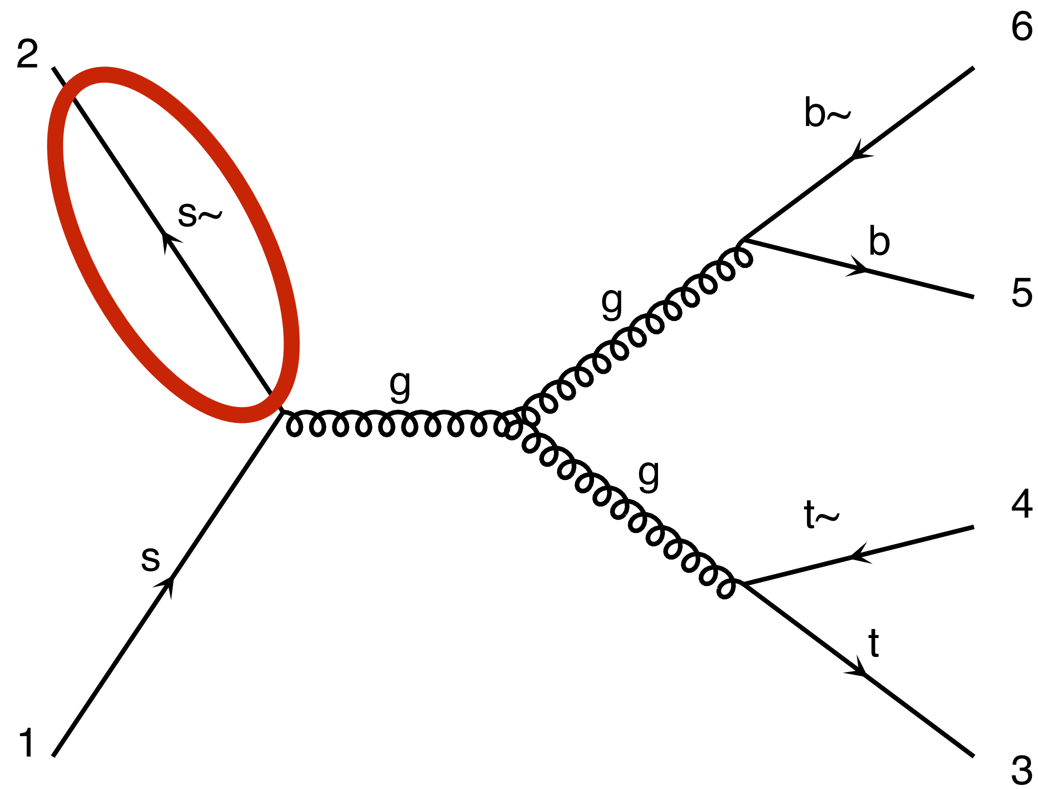
- Evaluate $\mathcal{M}$ for fixed helicity of external particles
  - ➡Multiply $\mathcal{M}$ with $\mathcal{M}^*$ -> $|\mathcal{M}|$^2
  - ➡Loop on Helicity and average the results

Doing the loop

$$\sum_{h=1}^{2^N} |M_h|^2$$

- Do we recompute the same quantity over an over?

Helicity Recycling

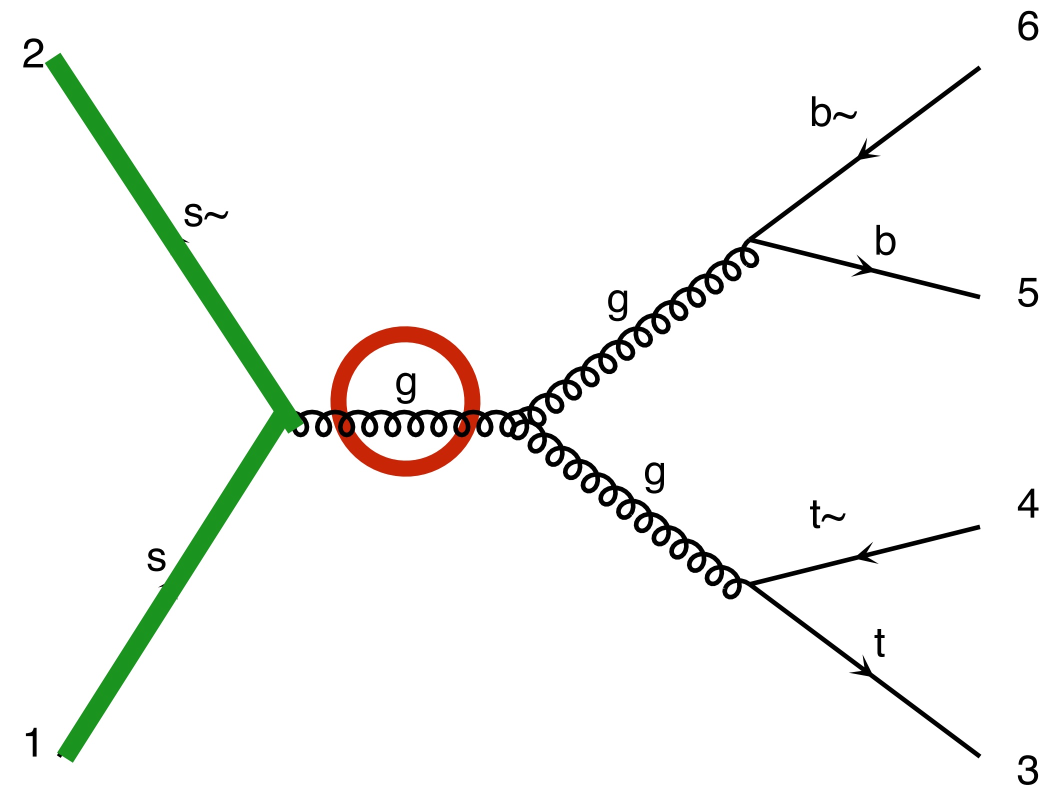


Number of helicity for particle 1: 2

Number of helicity for the full event: ~~64~~ 16

Wasted computation ratio: ~~32~~ 8

Helicity Recycling

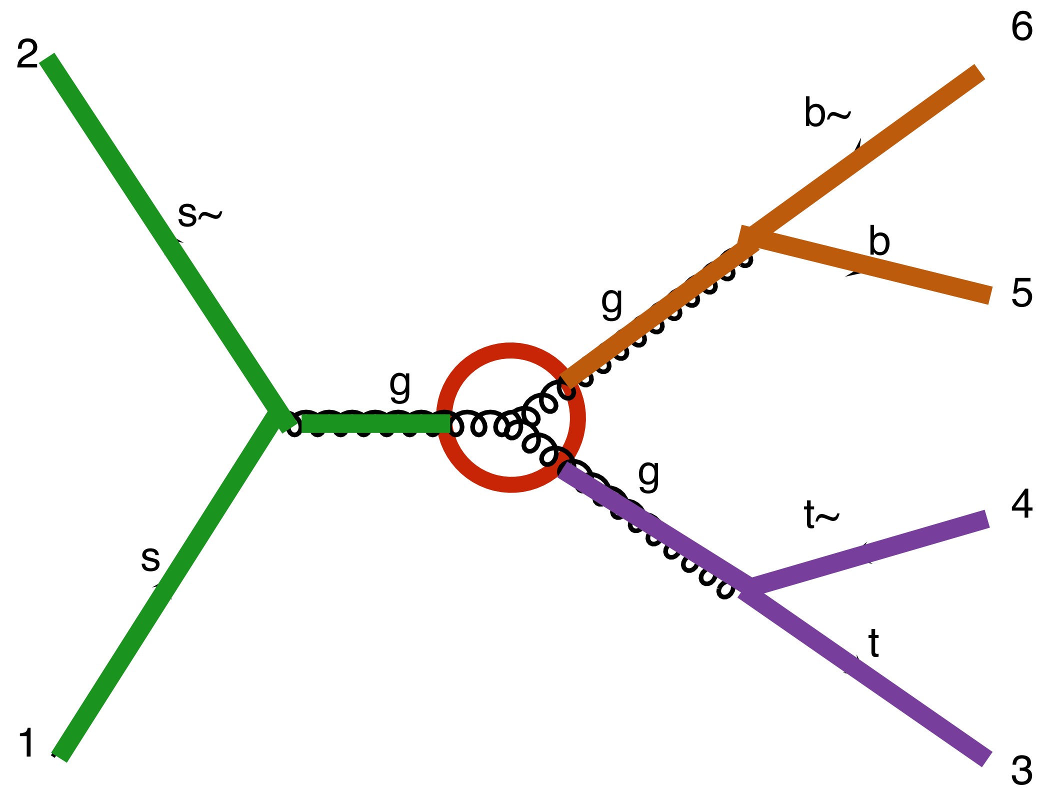


Number of helicity combination for the gluon: ~~4~~ 2

Number of helicity for the full event: ~~64~~ 16

Wasted computation ratio: ~~16~~ 8

Helicity Recycling



Number of helicity combination for the final computation: ~~64~~ 16

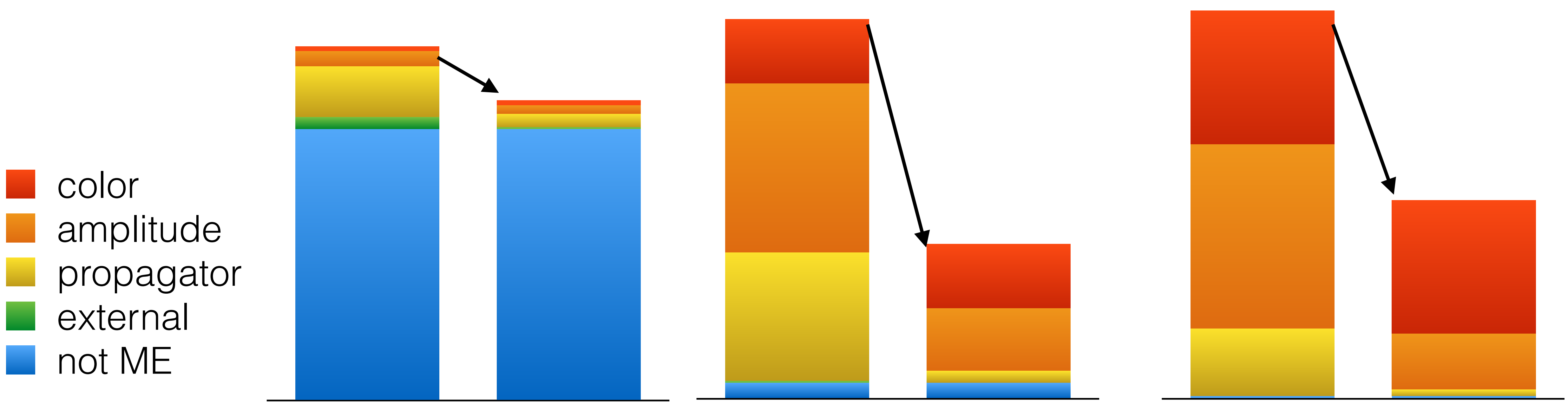
Number of helicity for the full event: ~~64~~ 16

Wasted computation ratio: ~~1~~ 1

But we can be smarter here -> around 2 (simple process) to 4 times faster

Solution Helicity Recycling

	$gg \to t\bar{t}$		$gg \to t\bar{t}gg$		$gg \to t\bar{t}ggg$	
	Instructions	Reduction	Instructions	Reduction	Instructions	Reduction
`madevent`	11G	15%	180G	62%	5T	55%
`matrix1`	1G (9.3%)	68%	160G (90%)	64%	4.9T (98%)	55%
↳ ext	76M (<1%)	83%	100M (<1%)	97%	110M (<1%)	98%
↳ int	540M (4.8%)	72%	16G (8.9%)	90%	180G (3.6%)	91%
↳ amp	280M (2.6%)	47%	77G (42%)	63%	1.7T (33%)	69%



Not doing the sum

- One can replace a sum by an integral

$$\sum_{h=1}^{2^N} |M_h|^2 \longrightarrow \int_0^{2^N} dh\, |M_{Round(h)}|^2$$

- Increase the dimension of the integral by one

$$\int d\Phi\, f_1 f_2 \sum_{h=1}^{2^N} |M_h|^2 \longrightarrow \int d\Phi \int_0^{2^N} dh\, f_1 f_2\, |M_{Round(h)}|^2$$

  - Reduce complexity of the function to compute
    - Higher impact of the PDF/...
  - Does not change the scaling of the convergence
  - But increase the variance of the function

Comparison

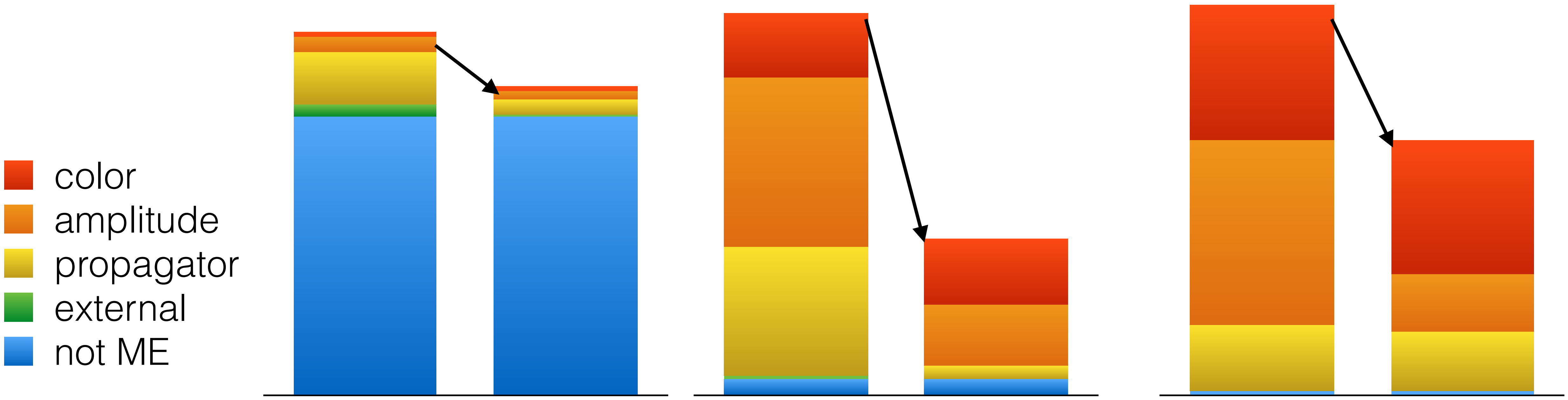
Preliminary

	Type	Survey	Refine	nb_events
gg>tt	$\sum_{h=1}^{2^N} \lvert M_h \rvert^2$	<1s	4m57s	500k
	$\int_0^{2^N} dh \lvert M_{Round(h)} \rvert^2$	<1s	4m53s	
gg>ttgg	$\sum_{h=1}^{2^N} \lvert M_h \rvert^2$	2m48s	1h22	100k
	$\int_0^{2^N} dh \lvert M_{Round(h)} \rvert^2$	2m24s	1h05	
gg>ttggg	$\sum_{h=1}^{2^N} \lvert M_h \rvert^2$	10h	25h	10k
	$\int_0^{2^N} dh \lvert M_{Round(h)} \rvert^2$	1h50	4h20	**28**

Comparison

- Monte-Carlo over ....
  - simplify the function to integrate (a lot)
  - Forbid some optimisation
  - Increase the number of required evaluation
- For helicity case:
  - No super clear winner
- Possible to combine both method (?)
  - Monte-Carlo over subset of helicity

Computation status



Colour becomes a computation bottleneck!!

depends on two (large) matrix (but constant) matrix

$$J = B_{n!,n_{diag}} * M$$

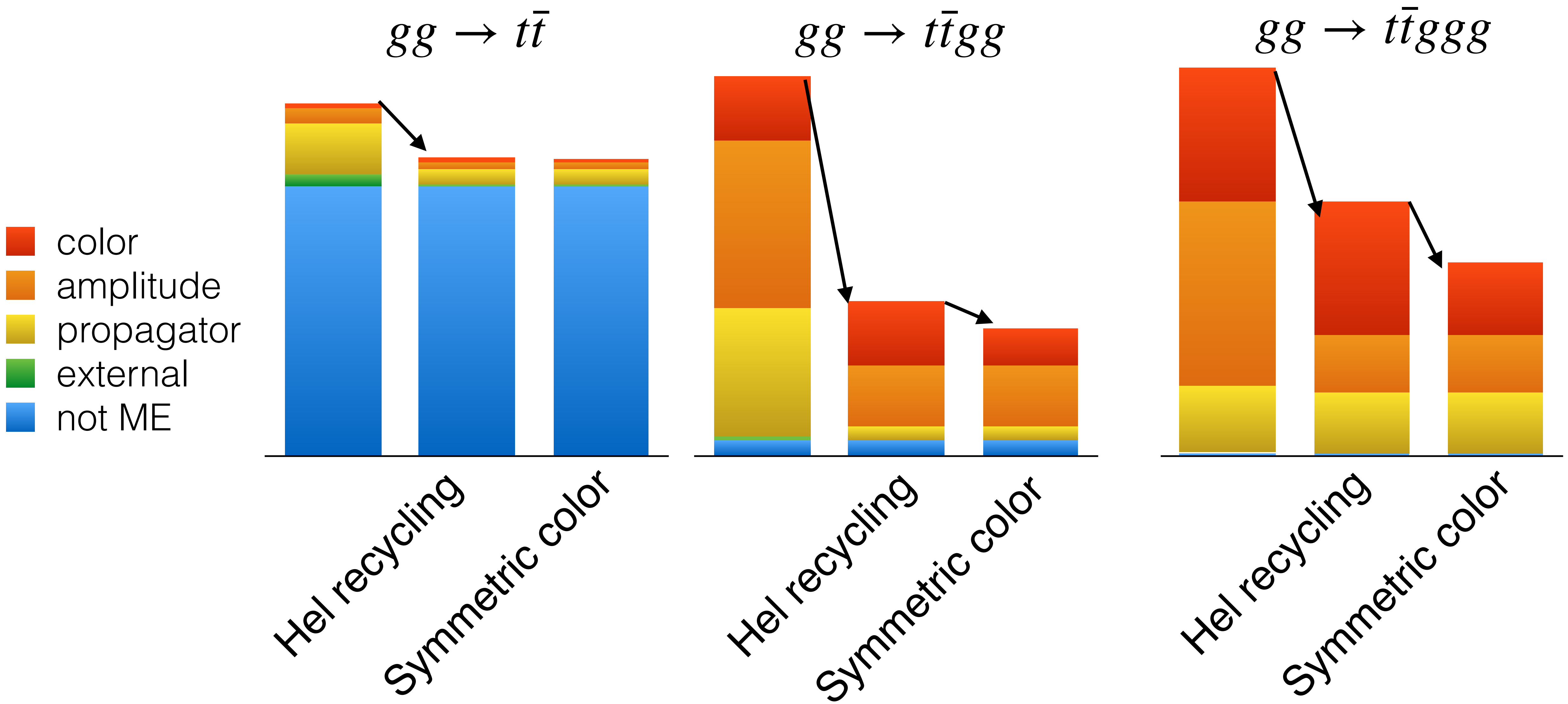
B is a very sparse matrix. (Not the main bottleneck)

$$|M_h|^2 = J^\dagger C_{n!,n!} J$$

C is a real symmetric matrix

Color

- Trivial update: use the symmetry
  - Only identify in 2022!

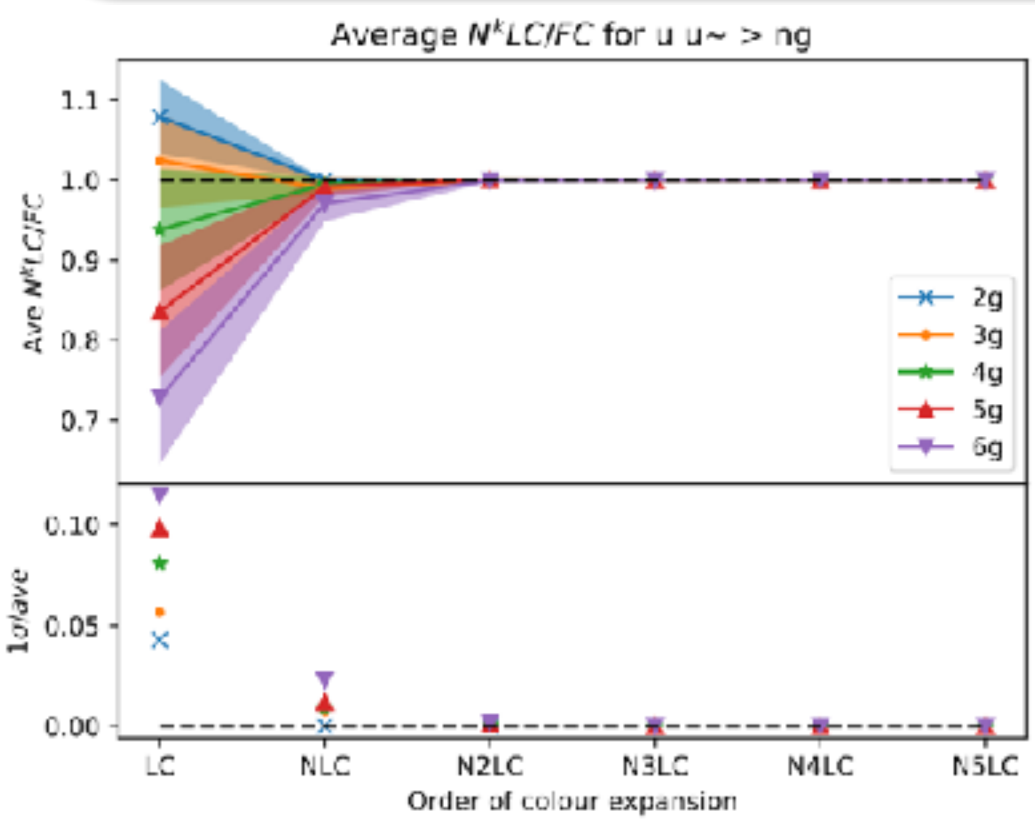


Color

$$C_{i,j} = N_c^X \left(a_0 + a_1 \frac{1}{N_c^2} + \mathcal{O}(\frac{1}{N_c^4}) \right) \qquad N_c = 3$$

$a_0 \neq 0$: Leading Color $\qquad a_0 = 0, a_1 \neq 0$: Next Leading Color

$$C(\sigma_k, \sigma_l) = \begin{pmatrix} \mathrm{LC} & 0 & 0 & 0 & 0 & \mathrm{NLC} \\ 0 & \mathrm{LC} & 0 & \mathrm{NLC} & 0 & 0 \\ 0 & 0 & \mathrm{LC} & 0 & 0 & 0 \\ 0 & \mathrm{NLC} & 0 & \mathrm{LC} & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathrm{LC} & 0 \\ \mathrm{NLC} & 0 & 0 & 0 & 0 & \mathrm{NLC} \end{pmatrix}$$



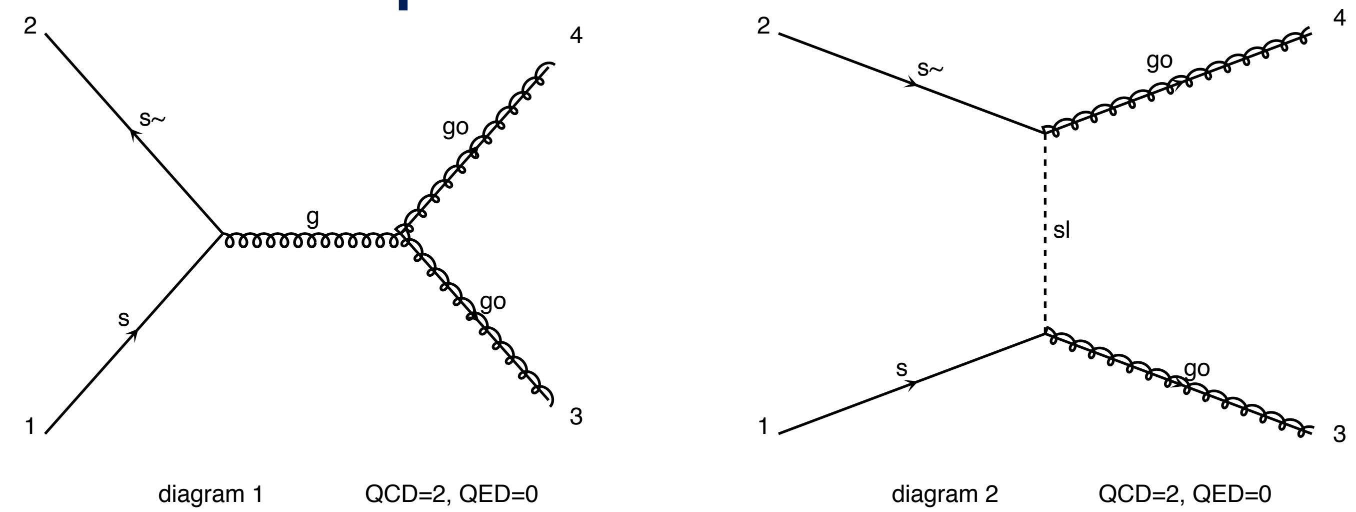
Take away

- Not the full matrix is needed
  - Development needed
- LC is dominant
  - No numerical issue

Computation step

Calculate a given process (e.g. gluino pair)

- Determine the production mechanism



- Evaluate the matrix-element

$|\mathcal{M}|^2$ ➡Need Feynman Rules!

- Phase-Space Integration

$$\int d\Phi f_1(x_1) f_2(x_2) \, |M|^2$$

- Un-weighting

Phase-Space

$$\int |M_{tot}|^2 = \int \frac{\sum_i |M_i|^2}{\sum_j |M_j|^2} |M_{tot}|^2 = \sum_i \int \frac{|M_i|^2}{\boxed{\sum_j |M_j|^2} } \boxed{|M_{tot}|^2}$$

$$\approx 1$$

Key Idea

- Any single diagram is “easy” to integrate (pole structures/ suitable integration variables known from the propagators)

N Integral

- Errors add in quadrature so no extra cost
- “Weight” functions already calculated during $|M|^2$ calculation
- Parallel in nature: **embarrassingly parralel**

- What if interference are large -> change strategy

$$\int |M_{tot}|^2 = \sum_i \int \frac{\alpha_i(x)}{\sum_j \alpha_j(x)} |M_{tot}|^2 \qquad \alpha_i = \prod \frac{1}{|(q^2 - m^2 + im\Gamma|^2}$$

process	old strategy		new strategy		speed-up	default
VBF-like processes	survey	refine	survey	refine		
$pp \to W^+W^+jj\ [g_S = 0]$	13s	**2h12m**/1290	16s	8m1s	16×	new
$pp \to W^+W^-jj, W \to l\nu l[g_S = 0{,}13\,\text{TeV}]$	19m0s	9m6s	10m0s	1m43s	2.4×	new
$pp \to W^+W^-jj, W \to l\nu l[g_S = 0{,}100\,\text{TeV}]$	10m0s	24m8s	7m0s	18m10s	1.4×	new
$u\bar{d} \to W_L^+W_L^-u\bar{d}[g_S = 0]$	23s	**27h56m**/203	14s	1m53s	792×	new
$u\bar{d} \to W_L^+W_L^-u\bar{d}, W^+ \to d\bar{u}, W^- \to \tau^+\nu_\tau[g_S = 0]$	2m0s	**15h52m**/793	1m0s	5m42s	142×	new
$u\bar{d} \to W_T^+W_T^-u\bar{d}, W^+ \to d\bar{u}, W^- \to \tau^+\nu_\tau[g_S = 0]$	36s	2m54s	37s	2m28s	1.1×	new
$\mu^+\mu^- \to hhh\bar{\nu}_\mu\nu_e$ [14 TeV]	3s	**8h50m**/641	1s	11s	2653×	new
$\mu^+\mu^- \to t\bar{t}\mu^+\mu^-$ [13 TeV]	20s	**3h6m**/948	6s	25s	362×	new
$\mu^+\mu^- \to W^+W^-\mu^+\mu^-$ [4 TeV]	1m0s	33m26s	16s	15s	66×	new
other processes	survey	refine	survey	refine		
$pp \to W^+[0-4]j$	20m0s	5s	20m0s	4s	1.0×	old
$pp \to t\bar{t}[0-2]j$	38s	32s	38s	19s	1.2×	old
$pp \to 4j$	1m0s	**1h21m**/7003	1m0s	21m5s	3.7×	new
$pp \to t\bar{t}3j$	1h0m	1h36m	2h0m	1h37m	0.71×	old
$pp \to W^+Z$	1s	3s	1s	2s	1.3×	new
$pp \to t\bar{t}h$	<1s	2s	<1s	3s	0.67×	old
$pp \to t\bar{t}hj$	2s	4s	3s	10s	0.45×	old
$pp \to t\bar{t}Z$	1s	4s	1s	4s	1.0×	old
$pp \to W^+W^-jj$ [QCD only]	11s	36s	11s	37s	1.0×	old

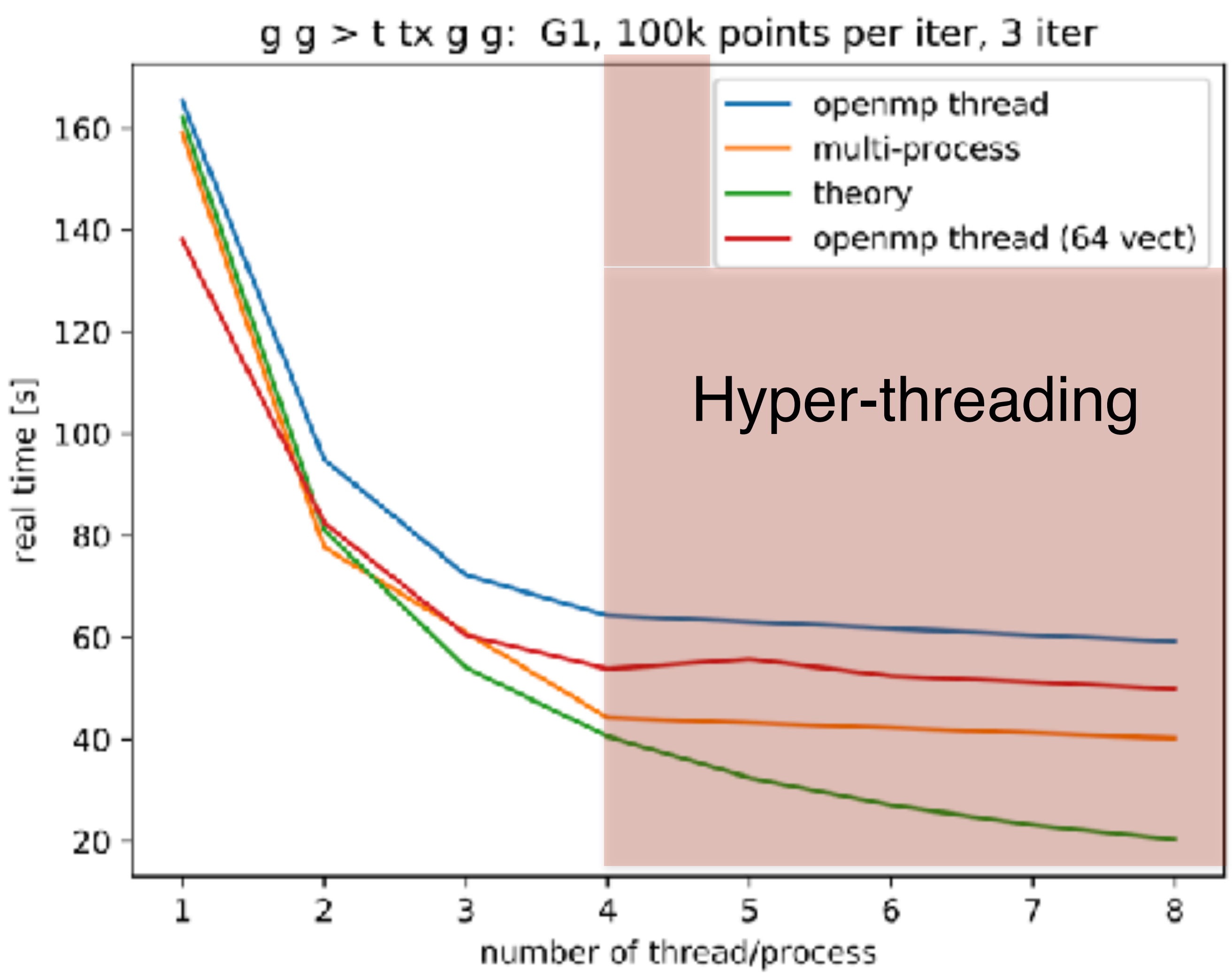
More on this -> MadNIS

Other idea: local gauge cancellation: 2203.10440

Phase-Space

Refactoring (LO) phase-space for data parallelism
See Andrea's talk

Required for SIMD and GPU port of the code
Also allow OpenMP (for running gridpack?)

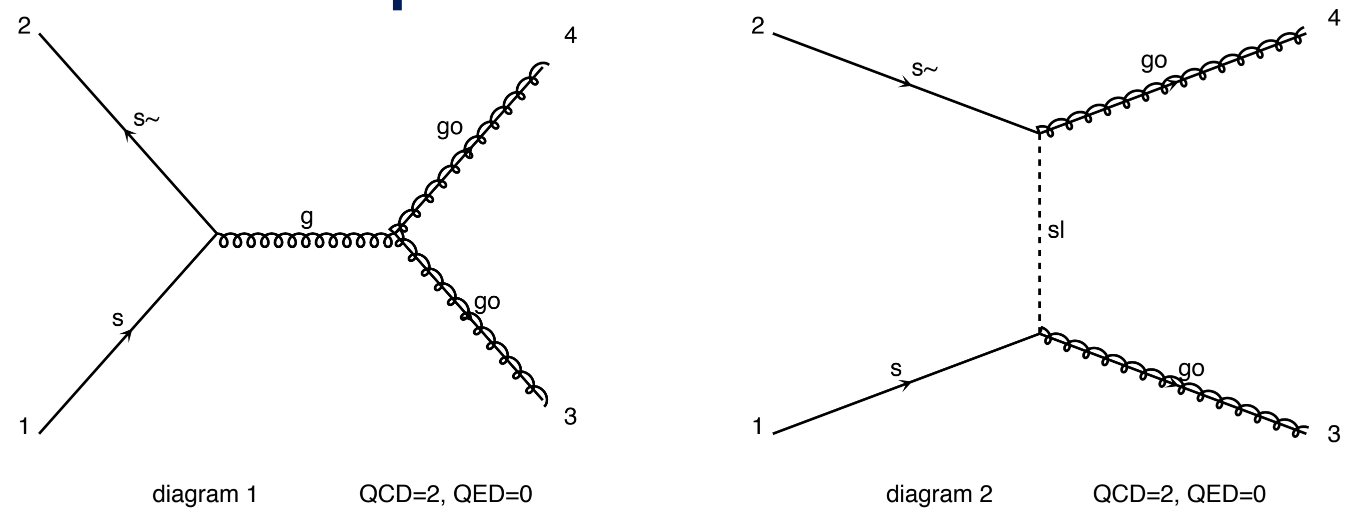


- Multi-process is more efficient
- only if you release the core/thread for other task
- Low impact of hyper-threading

Computation step

Calculate a given process (e.g. gluino pair)

- Determine the production mechanism



- Evaluate the matrix-element

$|\mathcal{M}|^2$ ➡Need Feynman Rules!

- Phase-Space Integration

$$\int d\Phi f_1(x_1) f_2(x_2) |M|^2$$

- Un-weighting

Re-weighting

Re-Weighting

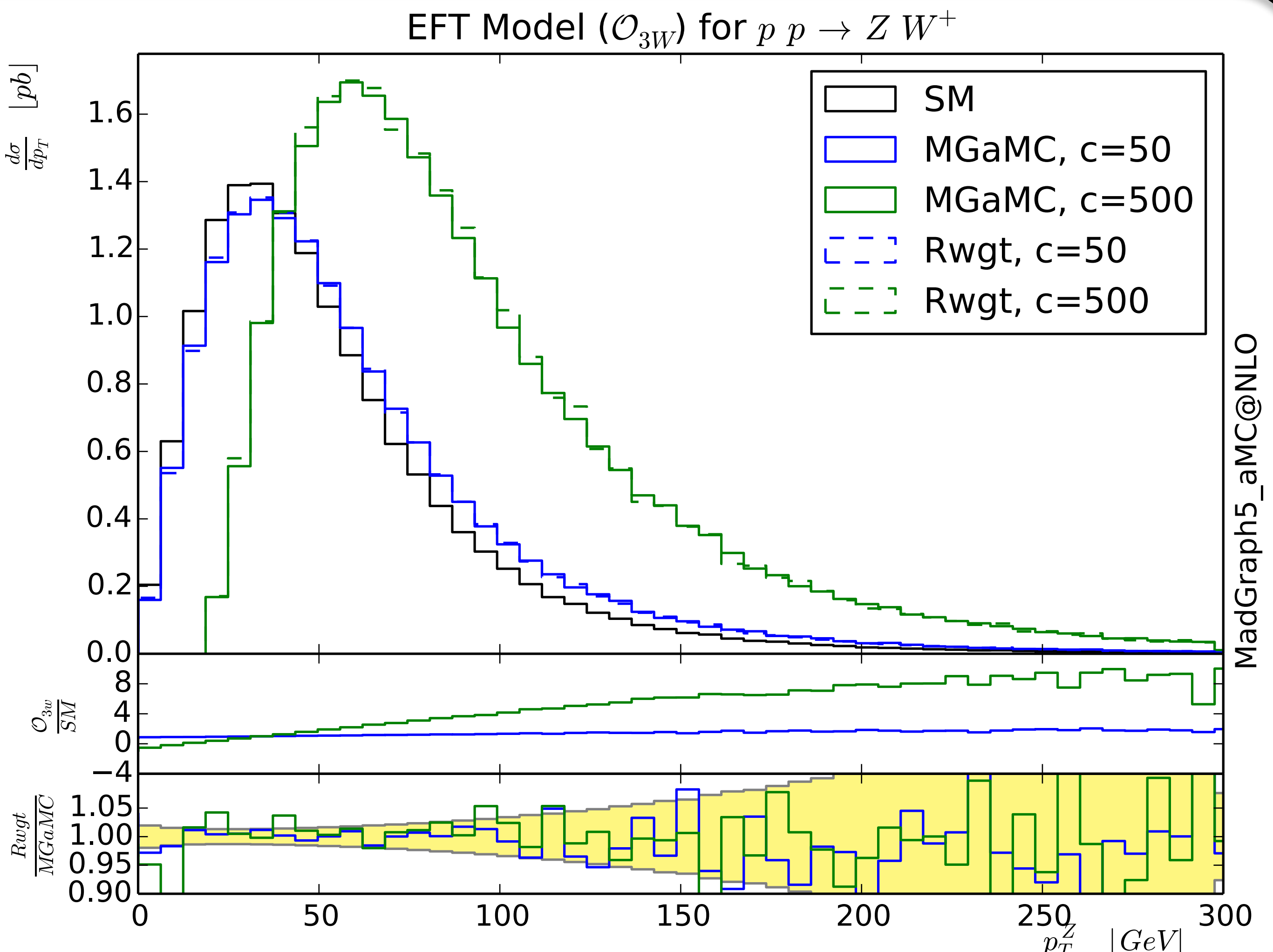
1404.7129
1607.00763

- Change the weight of the events

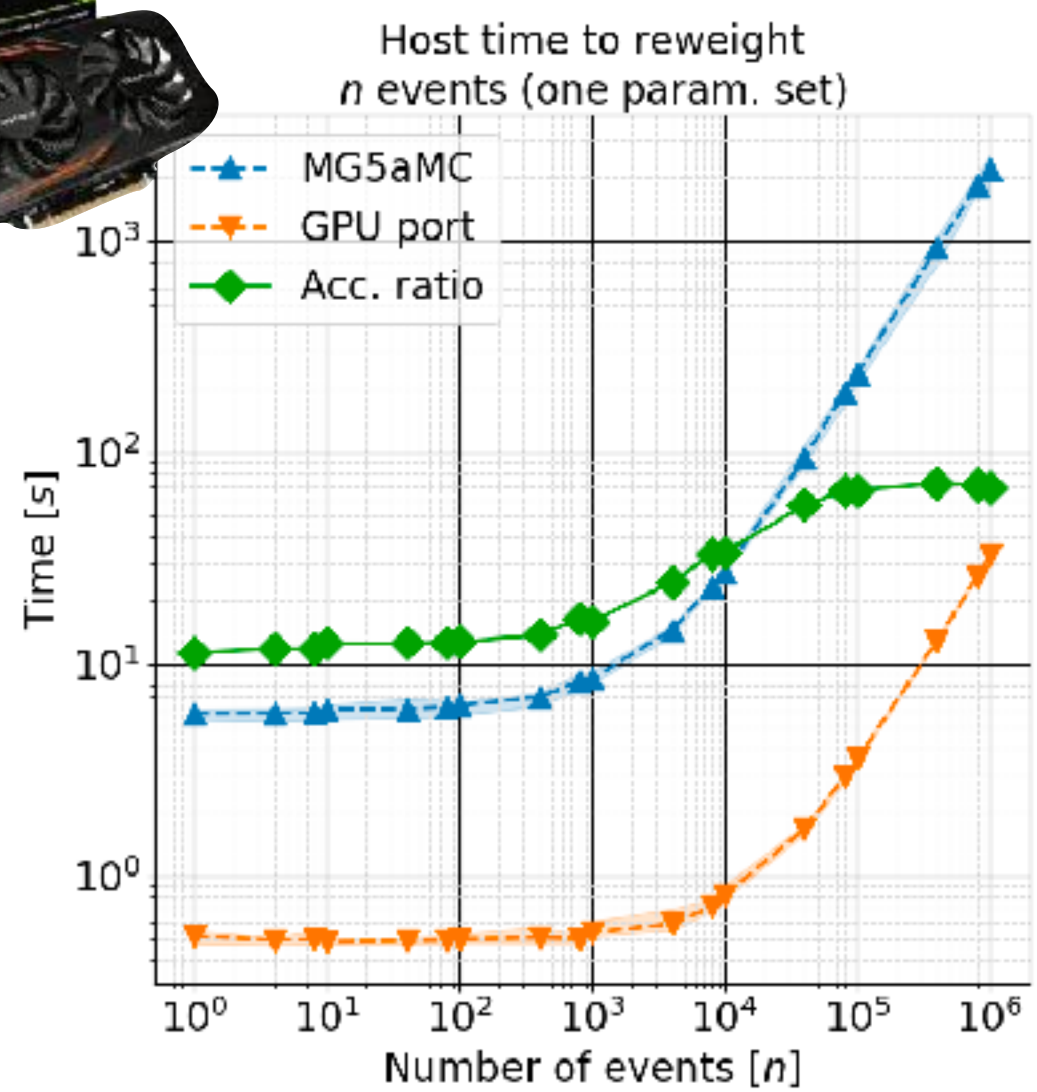
$$W_{new} = \frac{|M_{new}|^2}{|M_{old}|^2} * W_{old}$$

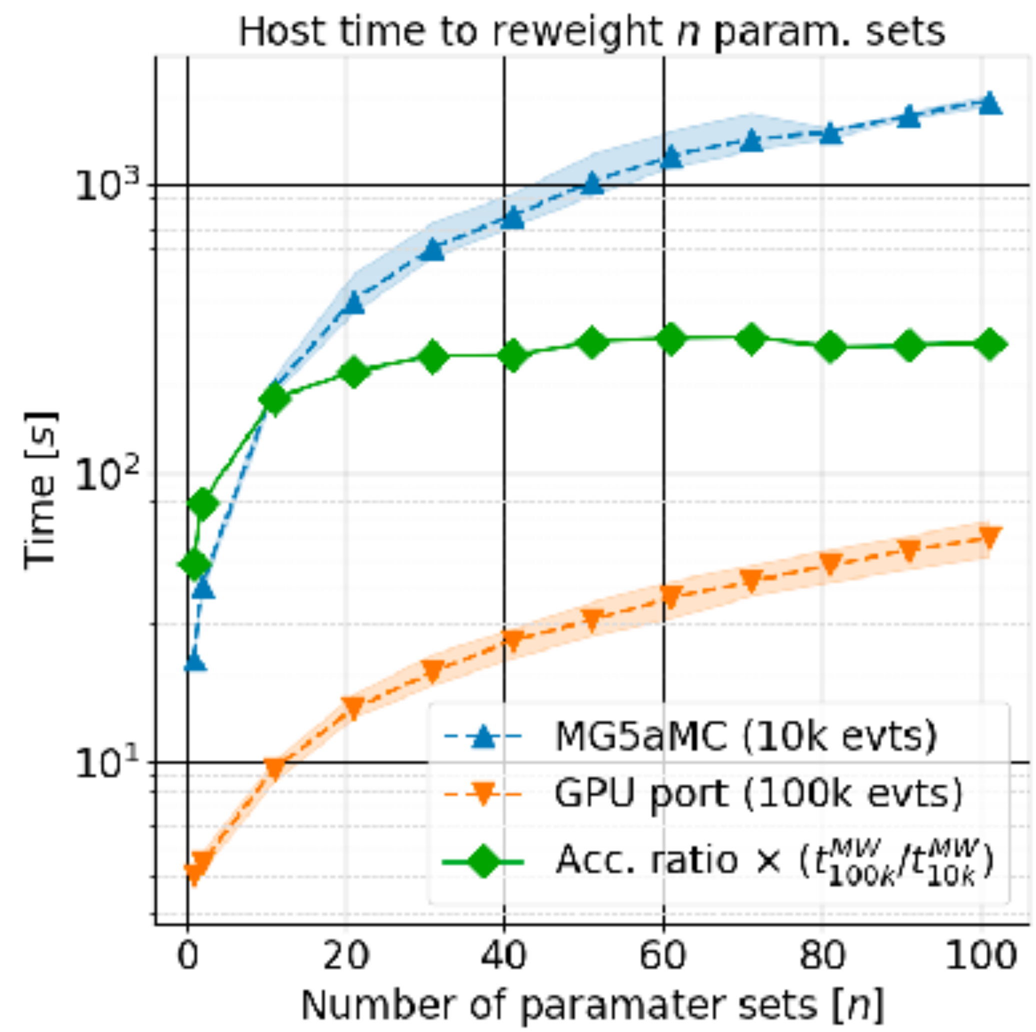
EFT Case

$$\mathcal{O}_{3W} = Tr\left[W_{\mu\nu}W^{\nu\rho}W_{\rho}{}^{\mu}\right]$$



Re-weighting on GPU





Huge speed-up (~100x)

Need Huge sample (to be able to fill the GPU for each flavour)

Work and plot by Zenny Wettersten

Origin of the negative events

Rate of negative events

$pp \to e^+e^-$	6.9%	(1.3)
$pp \to e^+\nu_e$	7.2%	(1.4)
$pp \to H$	10.4%	(1.6)
$pp \to Hb\bar{b}$	40.3%	(27)
$pp \to W^+j$	21.7%	(3.1)
$pp \to W^+t\bar{t}$	16.2%	(2.2)
$pp \to t\bar{t}$	23.0%	(3.4)

Cost In sample size

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

MC@NLO

$$d\sigma^{(\mathbb{H})} = d\sigma^{(\mathrm{NLO},E)} - d\sigma^{(\mathrm{MC})},$$

$$d\sigma^{(\mathbb{S})} = d\sigma^{(\mathrm{MC})} + \sum_{\alpha=S,C,SC} d\sigma^{(\mathrm{NLO},\alpha)}.$$

Origin

- H: over-estimate of the MC counter term term
- S: related to the fact that we hit multiple times the same born configuration

Result

MC@NLO-Δ /Folding (2002.12716)

	MC@NLO			MC@NLO-Δ		
	111	221	441	Δ-111	Δ-221	Δ-441
$pp \to e^- e^-$	6.9% (1.3)	3.5% (1.2)	3.2% (1.1)	5.7% (1.3)	2.4% (1.1)	2.0% (1.1)
$pp \to e^- \nu_e$	7.2% (1.4)	3.8% (1.2)	3.4% (1.2)	5.9% (1.3)	2.5% (1.1)	2.3% (1.1)
$pp \to H$	10.4% (1.6)	4.9% (1.2)	3.4% (1.2)	7.5% (1.4)	2.0% (1.1)	0.5% (1.0)
$pp \to Hbb$	40.3% (27)	38.4% (19)	38.0% (17)	36.6% (14)	32.6% (8.2)	31.3% (7.2)
$pp \to W^+ j$	21.7% (3.1)	16.5% (2.2)	15.7% (2.1)	14.2% (2.0)	7.9% (1.4)	7.4% (1.4)
$pp \to W^+ t\bar{t}$	16.2% (2.2)	15.2% (2.1)	15.1% (2.1)	13.2% (1.8)	11.9% (1.7)	11.5% (1.7)
$pp \to tt$	23.0% (3.4)	20.2% (2.8)	19.6% (2.7)	13.6% (1.9)	9.3% (1.5)	7.7% (1.4)

Born Spreading (2310.04160)

process	negative S event	
	no born smearing	after born smearing
$pp \to e^+ e^-$	7.1%	2.0%
$pp \to H$	10.6%	1.1%
$pp \to t\bar{t}$	8.6%	2.1%
$pp \to W^+ t\bar{t}$	4.2%	2.6%
$pp \to W^+ j$	24.2%	18.8%
$pp \to Hb\bar{b}$	27.3%	24.7%

Conclusion (1/2)

How to speed-up the computation?

- Faster matrix-element
- Better Hardware support (-> Andreas's talk)
- Better integrator (-> Ramon's talk)
- Better method (re-weighitng, avoid negative events)

A new era is coming

- Machine Learning need large sample to starts with
- GPU needs massive sample
  - The matrix-element will be for free
    - Many new opportunity

Conclusion (2/2)

We need more collaboration with IT

- MC tools are handle by theorist
- career path related to new feature/prediction
  - NOT on efficiency

We need:

- Strengthen the synergy with ML group
- Move GPU/ML production towards NLO
- Move colour optimisation towards GPU
- More efficiency study on Monte-Carlo/…
  - Re-optimise for GPU/SIMD

We need:

- PDF on GPU (CUDA?)

Backup slide

Amplitude solution

- Situation can be improved by splitting the amplitude computation into two steps

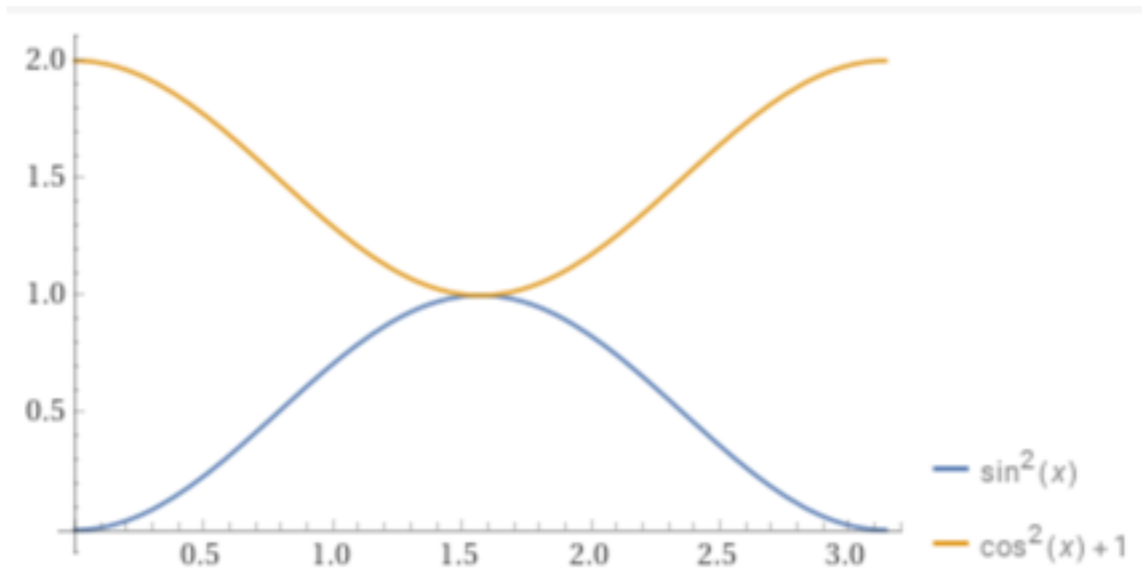
$$\mathscr{M}_{h_1h_2h_3} = \bar{\psi}_1^{h_1}\gamma_\mu\psi_2^{h_2}\phi_{h_3}^{\mu} \longrightarrow J_\mu^{h_1h_2} = \bar{\psi}_1^{h_1}\gamma_\mu\psi_2^{h_2}$$

$$\mathscr{M}_{h_1h_2h_3} = J_\mu^{h_1h_2}\phi_{h_3}^{\mu}$$

New recycling possible for $J_\mu^{h_1h_2}$

- Expected gain (on the amplitude):
  - ~2x for small multiplicity
  - ~4x for high multiplicity

Worst case scenario



$$\int d\Phi f_1 f_2 \sum_{h=1}^{2^N} |M_h|^2 \longrightarrow \int d\Phi \int_0^{2^N} dh\, f_1 f_2 |M_{Round(h)}|^2$$

Constant function
Easy to integrate

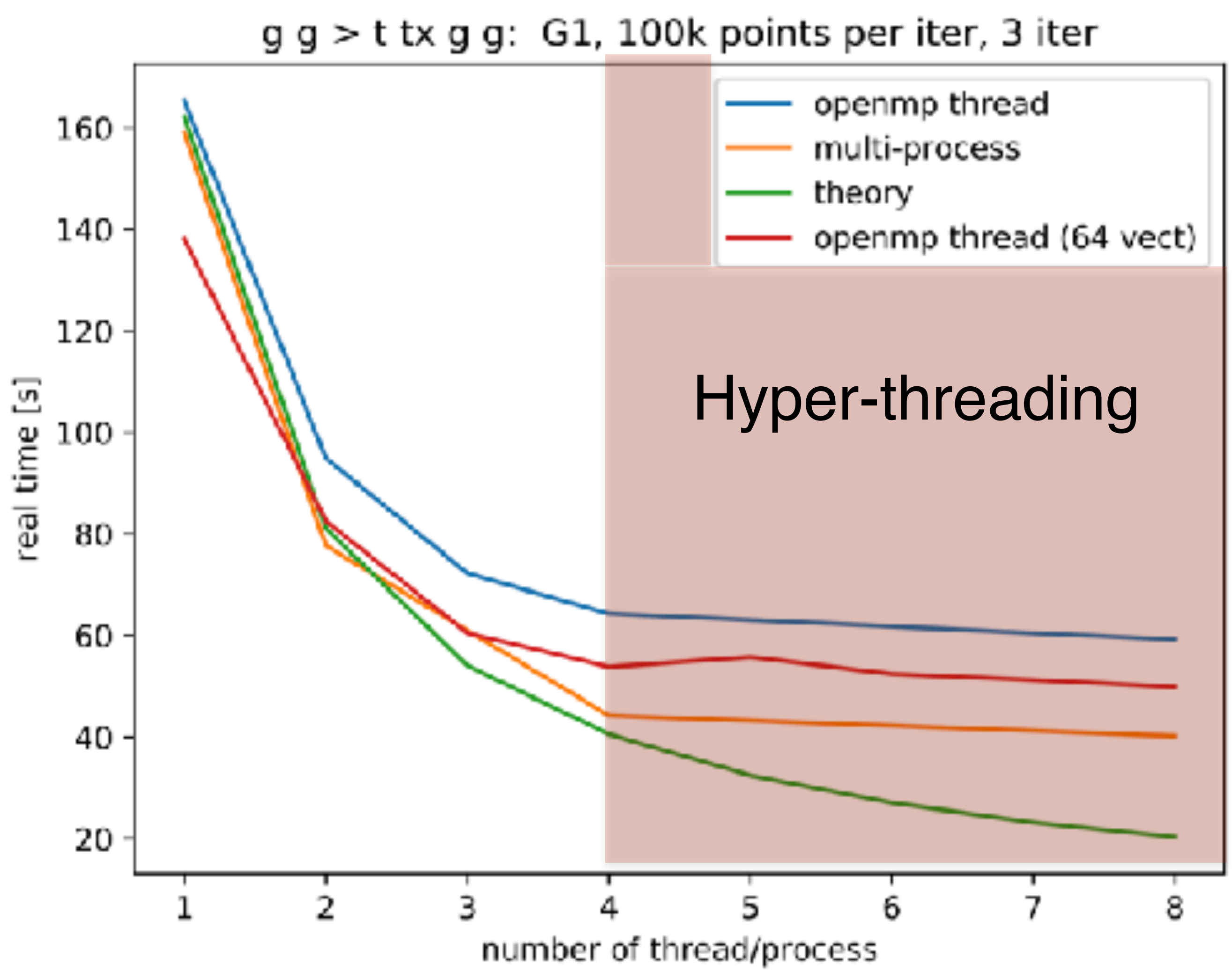
$\longrightarrow$

Fluctuation of the integrand
correlation between degrees of freedom
Not catched by Phase-Space integrator

Phase-Space

Refactoring (LO) phase-space for data parallelism

Required for SIMD and GPU port of the code
Also allow OpenMP (for running gridpack?)



- Multi-process is more efficient
- only if you release the core/thread for other task
- Low impact of hyper-threading

Negative events proposal

Modified MC@NLO (2002.12716)

$$d\sigma^{(\Delta,\mathbb{H})} = \left(d\sigma^{(\mathrm{NLO},E)} - d\sigma^{(\mathrm{MC})}\right)\Delta\,,$$

$$d\sigma^{(\Delta,\mathbb{S})} = d\sigma^{(\mathrm{MC})}\Delta + \sum_{\alpha=S,C,SC} d\sigma^{(\mathrm{NLO},\alpha)} + d\sigma^{(\mathrm{NLO},E)}\left(1-\Delta\right).$$

$\Delta \longrightarrow 0$ soft and collinear limits.

$\Delta \longrightarrow 1$ hard regions.

$$\Delta = 1 + \mathcal{O}(\alpha_S)\,.$$

Born Spreading (2310.04160)

$$\mathcal{F}^{(\mathbb{S})} = \int\left[\frac{B(\Phi_B)\,F(\Phi_r)}{\int F(\Phi_r)\,d\Phi_r} + \frac{V(\Phi_B)}{\int d\Phi_r} + K_{\mathrm{MC}}(\Phi_B,\Phi_r)\right]d\Phi_r \times \mathcal{F}^{(B)}_{\mathrm{MC}}$$

Folding

$$\mathcal{F}_{\mathrm{MC}}\left(\mathcal{K}^{(\mathbb{S})}\right)\int_{\chi_r} d\sigma^{(\mathbb{S})} \simeq \mathcal{F}_{\mathrm{MC}}\left(\mathcal{K}^{(\mathbb{S})}\right)\sum_{i_\xi=1}^{n_\xi}\sum_{i_y=1}^{n_y}\sum_{i_\varphi=1}^{n_\varphi}\frac{w_{i_\xi i_y i_\varphi}}{n_\xi n_y n_\varphi}\,d\sigma^{(\mathbb{S})}\left(\mathcal{K}^{(\mathbb{S})},\xi_{i_\xi},y_{i_y},\varphi_{i_\varphi}\right).$$