FACULTÉ
DES SCIENCES
D'ORSAY

Quarkonium production in MadGraph5_aMC@NLO

Quarkonia as Tools 2025

Alice COLPANI SERRI

C. FLETT, L. SIMON, J.-P. LANSBERG, O. MATTELAER, H.-S. SHAO

January 8, 2025

MadGraph5_aMC@NLO = event / diagram generator (Python)
↪ compute matrix-element (helicity amplitude formalism)

- Total cross section
- Differential cross section
- Un-weighted event

★ Lagrangian
⇓
★Feynman Rules
⇓
★ Matrix Element
⇓
★ Parton Events
⇓
★Hadronise Events
⇓
★Detector Events

Master integral for symmetric collisions in MadGraph5:

$$\sigma(AA \to X) = \sum_{i,j} \int dx_i dx_j d\Phi f_{i/A}(x_i) f_{j/A}(x_j) \hat{\sigma}_{(ab \to X)}(x_i, x_j, \mu_F, \mu_R)$$

How MadGraph5 works:

- Identify partonic processes and calculate partonic cross section
- Use PDFs (LHAPDF package)
- Do phase space integral and convolute with PDFs
- Generate events

NEW implementations:

- Asymmetric collisions [see Carlo's talk]
- **Quarkonium production!** (this talk)

There are many reasons why quarkonia are interesting:

- QCD studies!
- internal structure of nucleons
- gluon investigation (e.g. EIC) → TMDs!!
- ...

Focus on inclusive processes → factorisation:

$$\sigma(\mathrm{pp} \to \mathcal{Q} + X) = \sum_{i,j,n} \int \mathrm{d}x_1 \mathrm{d}x_2 f_{i/\mathrm{p}}(x_1) f_{j/\mathrm{p}}(x_2) \times \hat{\sigma}(ij \to Q\bar{Q}[n] + X) \langle \mathcal{O}_n^{\mathcal{Q}} \rangle$$

$$n = {}^{2s+1}L_J^c$$

Models for quarkonium production: NRQCD, CSM, CEM, ...

Motivation: why automation of quarkonium cross sections?

Many reasons, including:

- global data-theory comparisons
- physics cases for future experimental facilities
- global NRQCD fits

Matrix element/event generators publicly available
($\hookrightarrow$ interfacing of e.g. HERWIG or PYTHIA with e.g. MG5_aMC)

$\downarrow$

facilitates complete computation

✓ versatility and enhanced physics simulation capabilities

✗ integration complexity, computational overhead, code compatibility and increased learning requirements

Why MadGraph5_aMC@NLO?

MadOnia:

- ✓ single quarkonium production phenomenology (only)
- ✗ (deprecated) module within MadGraph4

Helac-Onia:

- ✓ (S-wave/P-wave) multiple-quarkonium production based on tree-level helicity amplitudes
- ✗ limited to LO (not immediately extendable to NLO)

MadGraph5_aMC@NLO:

- ✓ flexibility to support SM, BSM and large number of particle physics models
- ? no quarkonia final states → (technical) complexities arise!

  Goal: **automation of LO quarkonium with NLO in sight**

Objectives and plan of the project

□ Implementation of quarkonia in `MadGraph5_aMC@NLO` at LO
↪ Single and multiple S-wave inclusive quarkonium production

- Colour projectors
- Spin projectors
- Interface
- Phase space adaptation

□ Extensions to states with leading P-wave Fock states
LO implementation ⟶ □ NLO in the easiest way possible!

□ TMD factorisation also to be implemented
↪ for example gg → di-J/ψ (... not only for quarkonia)

Status of the project

- Implementation of quarkonia in `MadGraph5_aMC@NLO` at LO
  ↪ Single and multiple S-wave inclusive quarkonium production
  - Colour projectors → implemented ✓
  - Spin projectors → implemented ✓
  - Interface → implemented ✓
  - Phase space adaptation

- Extensions to states with leading P-wave Fock states
  LO implementation ⟶ NLO in the easiest way possible!

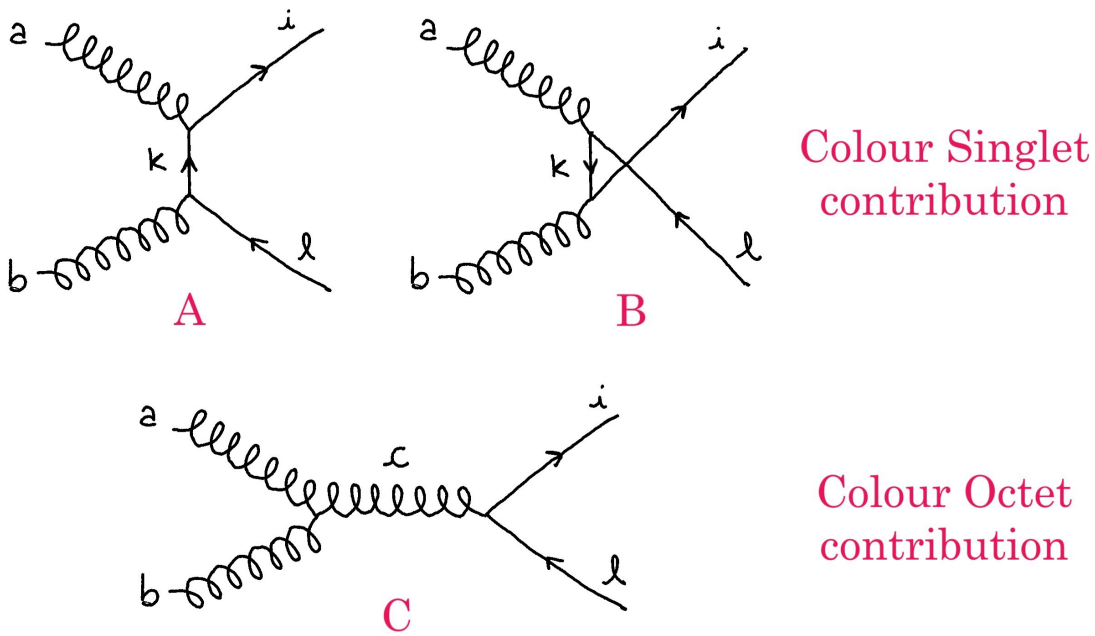
- TMD factorisation also to be implemented
  ↪ for example gg → di-J/ψ

Colour projectors $\mathbb{P}_C$ (1)

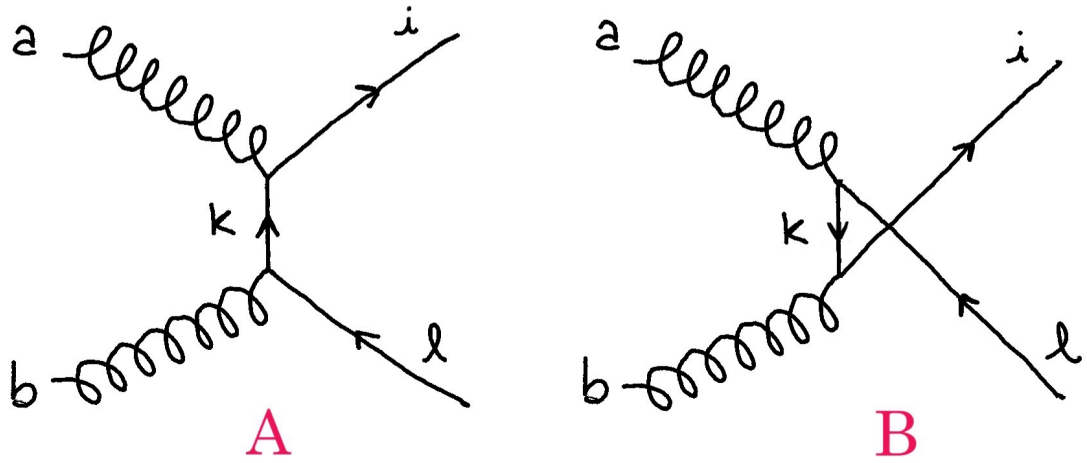
Quarkonium in the quantum state $n \longrightarrow$ colour singlet or octet?

$$\mathbb{P}_1 = \delta_{ij}/\sqrt{N_c}$$
$$\mathbb{P}_8 = \sqrt{2}t^c_{ij}$$

- Example: $gg \longrightarrow c\bar{c}$ **3 diagrams** at LO



Colour projectors $\mathbb{P}_C$ (2)

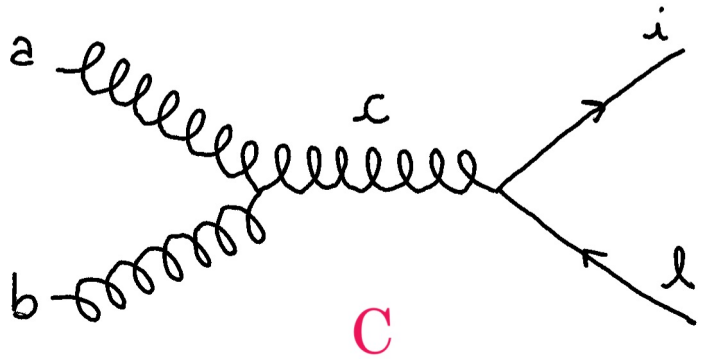


$$(\mathrm{A}) \sim (t^a_{ik} t^b_{kl}) = (t^a t^b)_{il} \quad \rightarrow \text{we set } (t^a t^b)_{il} = c1$$

$$(\mathrm{B}) \sim (t^b_{ik} t^a_{kl}) = (t^b t^a)_{il} \quad \rightarrow \text{we set } (t^b t^a)_{il} = c2$$

$$\hookrightarrow \text{open } c\bar{c} \text{ colour basis of dim} = 2$$

Colour projectors $\mathbb{P}_C$ (3)



C

$$(\mathrm{C}) \sim f^{abc}(t^c_{il}) = (t^a t^b)_{il} - (t^b t^a)_{il}$$

$$= \quad c1 \quad - \quad c2$$

$$c1c1^\dagger = (t^a t^b)_{il}(t^b t^a)_{il} = \mathrm{Tr}(t^a t^b t^b t^a) = \frac{16}{3}$$

$$\vdots$$

$$\longrightarrow \text{colour matrix:} \begin{pmatrix} c1c1^\dagger & c1c2^\dagger \\ c2c1^\dagger & c2c2^\dagger \end{pmatrix}$$

Colour projectors $\mathbb{P}_C$ (4)

Apply colour projectors: $\mathbb{P}_1 = \delta^{il}$ and $\mathbb{P}_8 = t^c_{il}$

- Colour singlet:

$$(t^a t^b)_{il}\delta^{il} = \mathrm{Tr}(t^a t^b) \qquad (\mathrm{A})$$
$$(t^b t^a)_{il}\delta^{il} = \mathrm{Tr}(t^b t^a) \qquad (\mathrm{B})$$

- Colour octet:

$$\left.\begin{aligned}(t^a t^b)_{il}\, t^c_{il} &= \mathrm{Tr}(t^a t^b t^c) = \tfrac{1}{4}(d^{abc} + i f^{abc}) \quad (1)\\ (t^b t^a)_{il}\, t^c_{il} &= \mathrm{Tr}(t^b t^a t^c) = \tfrac{1}{4}(d^{bac} + i f^{bac}) \quad (2)\end{aligned}\right\} \ (1) - (2): \quad (\mathrm{C})$$

The amplitude will be given by the sum of the three contributions

$$\mathcal{A} = A(\mathrm{A}) + A(\mathrm{B}) + A(\mathrm{C})$$

Code: implementation of Colour Projectors

Colour projectors for m colour singlet and colour octet quarkonia production and associated production implemented ✓

Metacode: quarkonium formalism implemented via extension of python files which produce fortran code → numerical manipulations

In `/mg5amcnlo/madgraph/core`

- `color_algebra.py`
- `color_amp.py`
- `helas_objects.py`

Spin projectors $\mathbb{P}_S$ implementation

Quarkonium in the quantum state $n \longrightarrow$ spin singlet or triplet?

- Considering $a(k_1)b(k_2) \longrightarrow Q(k_3)\bar{Q}(k_4)$: JHEP 07 (2024), 050

$$\mathcal{A}(r) = \bar{u}_{\lambda_Q}(k_3)\Gamma(r)v_{\lambda_{\bar{Q}}}(k_4)$$

- Apply colour projector: $\mathcal{A}_{\{C\}}(r) = \sum_{c_3,c_4} \mathbb{P}_C \mathcal{A}(r)$
- Apply spin projector: $\mathcal{A}_{\{C,S\}}(r) = \sum_{\lambda_Q,\lambda_{\bar{Q}}} \mathbb{P}_S \mathcal{A}_{\{C\}}(r)$

$$\mathbb{P}_S = \frac{\bar{v}_{\lambda_{\bar{Q}}}(k_4)\Gamma_S u_{\lambda_Q}(k_3)}{2\sqrt{2m_Q m_{\bar{Q}}}} \qquad S = 0, \Gamma_S = \gamma_5; 1, \Gamma_S = \not{\epsilon}(K)$$

Declaration of new effective spinors in:

- `/mg5amcnlo/aloha/template_files/aloha_functions.f`
- `/mg5amcnlo/madgraph/core/helas_objects.py`

Python: calls `template_files` $\longrightarrow$ `matrix_i.f` files

Amplitudes organised into colour basis **JAMPs**

$$\mathcal{A} = \sum_i A_i \overset{\text{example}}{=} A(\mathrm{A}) + A(\mathrm{B}) + A(\mathrm{C})$$

$$A(\mathrm{A}) = c1A_1 \qquad A(\mathrm{B}) = c2A_2 \qquad A(\mathrm{C}) = c1A_{31} - c2A_{32}$$

$$\underline{\text{JAMP decomposition}} \quad \begin{cases} \text{JAMP}_1 & = A_1 + A_{31} \propto c1 \\ \text{JAMP}_2 & = A_2 - A_{32} \propto c2 \end{cases}$$

$$|\mathcal{A}|^2 = \sum_{i,j=1,2} \text{JAMP}_i^* \langle c_i | c_j \rangle \text{JAMP}_j$$

(Depends on spin projectors - constructed from helas routines)
Efficiency: large number of Feynman diagrams possible...
...but colour basis much smaller!

Interface implementation

Generate process: single, associated and multiple production
From the mg5amcnlo folder: type `./bin/mg5_aMC`

```
MG_aMC >
```

- ⋆ Example: $pp \longrightarrow J/\psi + c\bar{c}g$

```
MG_aMC > generate p p > c.c~(1|3S11) c c~ g
MG_aMC > generate p p > c.c~(1|3S18) c c~ g
```

dot notation for onia with spectroscopic notation in brackets

or

```
MG_aMC > generate p p > jpsi c c~ g
```

Particle name directly in the process generation
New files containing onium information in the `input` folder:

- `boundstates_default.txt`
- `fockstates_default.txt`

Interface implementation made in `/madgraph/interface/`

boundstates_default.txt

```
#
# Quarkonium Boundstates (S-waves only)
#
# Syntax: label = Fock states (separated by spaces)
#
# Charmonium
etac = c.c~(1|1S01) c.c~(1|1S08)
etac(2s) = c.c~(2|1S01) c.c~(2|1S08)
jpsi = c.c~(1|3S11) c.c~(1|3S18)
psi(2s) = c.c~(2|3S11) c.c~(2|3S18)
# Charmed  B mesons
bc+ = c.b~(1|1S01) c.b~(1|1S08)
b*c+ = c.b~(1|3S11) c.b~(1|3S18)
bc- = b.c~(1|1S01) b.c~(1|1S08)
b*c- = b.c~(1|3S11) b.c~(1|3S18)
# Bottomonium
etab = b.b~(1|1S01) b.b~(1|1S08)
etab(2s) = b.b~(2|1S01) b.b~(2|1S08)
upsilon = b.b~(1|3S11) b.b~(1|3S18)
upsilon(2s) = b.b~(2|3S11) b.b~(2|3S18)
# Toponium
etat = t.t~(1|1S01) t.t~(1|1S08)
etat(2s) = t.t~(2|1S01) t.t~(2|1S08)
```

fockstates_default.txt

```
# N: principal quantum number
# S: spin type
# L: orbital angular momentum quantum number
# J: total angular momentum quantum number
# C: color type

pid             pythia  name            notation        particle        anti-particle   N       S       L       J       C       charge

####################
# Charmonia        #
####################

# etac
4411001         441     etac(1|1S01)    c.c~(1|1S01)    c               c~              1       1       0       0       1       0
4411008         9940001 etac(1|1S08)    c.c~(1|1S08)    c               c~              1       1       0       0       8       0
4421001         100441  etac(2|1S01)    c.c~(2|1S01)    c               c~              2       1       0       0       1       0
4421008         9999999 etac(2|1S08)    c.c~(2|1S08)    c               c~              2       1       0       0       8       0

# J/psi
4413011         443     jpsi(1|3S11)    c.c~(1|3S11)    c               c~              1       3       0       1       1       0
4413018         9940003 jpsi(1|3S18)    c.c~(1|3S18)    c               c~              1       3       0       1       8       0
4423011         100443  psi(2|3S11)     c.c~(2|3S11)    c               c~              2       3       0       1       1       0
4423018         9940103 psi(2|3S18)     c.c~(2|3S18)    c               c~              2       3       0       1       8       0

####################
# Charmed B mesons #
####################

# Meson
5411001         541     bc+(1|1S01)     c.b~(1|1S01)    c               b~              1       1       0       0       1       1
5411008         9999999 bc+(1|1S08)     c.b~(1|1S08)    c               b~              1       1       0       0       8       1

5413011         543     b*c+(1|3S11)    c.b~(1|3S11)    c               b~              1       3       0       1       1       1
5413018         9999999 b*c+(1|3S18)    c.b~(1|3S18)    c               b~              1       3       0       1       8       1
```

display command

```
MG5_aMC>display boundstates
Bound state lables:
etac = c.c~(1|1S01) c.c~(1|1S08)
etac(2s) = c.c~(2|1S01) c.c~(2|1S08)
jpsi = c.c~(1|3S11) c.c~(1|3S18)
psi(2s) = c.c~(2|3S11) c.c~(2|3S18)
bc+ = c.b~(1|1S01) c.b~(1|1S08)
b*c+ = c.b~(1|3S11) c.b~(1|3S18)
bc- = b.c~(1|1S01) b.c~(1|1S08)
b*c- = b.c~(1|3S11) b.c~(1|3S18)
etab = b.b~(1|1S01) b.b~(1|1S08)
etab(2s) = b.b~(2|1S01) b.b~(2|1S08)
upsilon = b.b~(1|3S11) b.b~(1|3S18)
upsilon(2s) = b.b~(2|3S11) b.b~(2|3S18)
etat = t.t~(1|1S01) t.t~(1|1S08)
etat(2s) = t.t~(2|1S01) t.t~(2|1S08)
```

generate command

```
MG5_aMC>generate g g > jpsi g
INFO: Trying to generate process g g > c.c~(1|3S11) g
INFO: Checking for minimal orders which gives processes.
INFO: Please specify coupling orders to bypass this step.
INFO: Trying coupling order WEIGHTED<=3: WEIGTHED IS QCD+2*QED
INFO: Trying process: g g > c c~ g WEIGHTED<=3 @1
INFO: Process has 16 diagrams
1 processes with 16 diagrams generated in 0.029 s
Total: 1 processes with 16 diagrams
INFO: Trying to generate process g g > c.c~(1|3S18) g
INFO: Checking for minimal orders which gives processes.
INFO: Please specify coupling orders to bypass this step.
INFO: Trying coupling order WEIGHTED<=3: WEIGTHED IS QCD+2*QED
INFO: Trying process: g g > c c~ g WEIGHTED<=3 @1
INFO: Process has 16 diagrams
1 processes with 16 diagrams generated in 0.029 s
Total: 2 processes with 32 diagrams
```

Results: some checked processes

Partial list of processes that have been checked (49 up to now):

- `g g > c.c~(1|1S01)`
- `g g > b.b~(1|1S08)` `single`
- `g g > b.b~(1|1S01) g` `single + elem.part.`
- `g g > b.b~(1|1S01) b.b~(1|1S01)`
- `g g > b.b~(1|1S01) c.c~(1|1S01)` `multiple`

Benchmarked our matrix elements squared against `Helac-Onia`:

Results: MadGraph5 vs Helac-Onia

- `g g > b.b~(1|1S01) g`

Phase-space point			
E	px	py	pz
0.5000000E+03	0.0000000E+00	0.0000000E+00	0.5000000E+03
0.5000000E+03	0.0000000E+00	0.0000000E+00	-0.5000000E+03
0.5000048E+03	0.1109232E+03	0.4448265E+03	-0.1995510E+03
0.4999952E+03	-0.1109232E+03	-0.4448265E+03	0.1995510E+03

Matrix element	1.4532913707599472E-005 GeV^0
Helac-Onia	1.45329122E-05 GeV^0
Ratio	1.0000001005670054

Some results: MadGraph5 vs Helac-Onia

- `g g > b.b~(1|3S11) g`

Matrix element	4.8065942918292797E-010 GeV^0
Helac-Onia	4.80663009E-10 GeV^0
Ratio	0.9999925515141896 4

- `g g > c.c~(1|1S01) g`

Matrix element	4.4080653118388797E-005 GeV^0
Helac-Onia	4.40806543E-05 GeV^0
Ratio	0.9999997242730454

- `g g > b.b~(1|1S01) b.b~(1|1S01)`

Matrix element	5.2953309332877083E-013 GeV^0
Helac-Onia	5.29533448E-13 GeV^0
Ratio	0.9999932957700877

Phase-space integration - work in progress

Phase space intagration in MadGraph5: multi-channelling

↪ efficiency: parallel computation

Phase-space adaptation: onia in final state single particle, not two!

- Process like $gg \longrightarrow c\bar{c}$ always interpreted as $2 \longrightarrow 2$ process

---

How to perform phase-space integration for onia in MG5:

- VEGAS algorithm (already in MG5)
- Single-diagram-enhanced multi-channel integration
- Application to quarkonium production

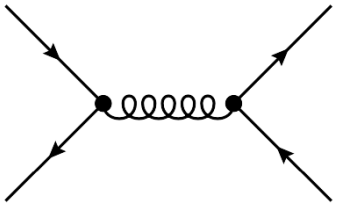
↪ new parameterisation required!

(ex. two t-channel peaks: issue with integration...)

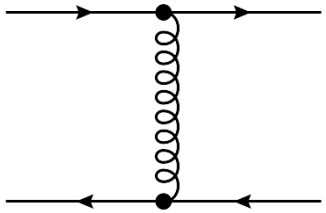
Single-diagram-enhanced multi-channel integration

slide courtesy of L. Simon

Example: $q\bar{q} \to q\bar{q}$



peak in s



peak in t

Single-diagram-enhanced multi-channel integration[Maltoni, Stelzer (2003)]:

$$\int \mathrm{d}\Pi_2 \left|\mathcal{A}_1 + \mathcal{A}_2\right|^2 = \int \mathrm{d}\Pi_2 \left(\underbrace{\left|\mathcal{A}_1\right|^2 \frac{\left|\mathcal{A}_1 + \mathcal{A}_2\right|^2}{\left|\mathcal{A}_1\right|^2 + \left|\mathcal{A}_2\right|^2}}_{\text{peak in } s} + \underbrace{\left|\mathcal{A}_2\right|^2 \frac{\left|\mathcal{A}_1 + \mathcal{A}_2\right|^2}{\left|\mathcal{A}_1\right|^2 + \left|\mathcal{A}_2\right|^2}}_{\text{peak in } t} \right)$$

Phase-space parametrisation[Byckling, Kajantie (1971)]:

$$\int \mathrm{d}\Pi_n \sim \int \mathrm{d}M_{n-1} \int \mathrm{d}\Omega_n \cdots \int \mathrm{d}M_2 \int \mathrm{d}\Omega_3 \int \mathrm{d}\Omega_2$$

$$\int \mathrm{d}\Pi_n \sim \int \mathrm{d}M_{n-1} \int \mathrm{d}\phi_n \int \mathrm{d}t_{n-1} \cdots \int \mathrm{d}M_2 \int \mathrm{d}\phi_3 \int \mathrm{d}t_2 \int \mathrm{d}\phi_2 \int \mathrm{d}t_1$$

New parts in the code: search
ONIA

GitHub: release **onia** branch of `MG5` version 3.x

[see Nanako's talk]

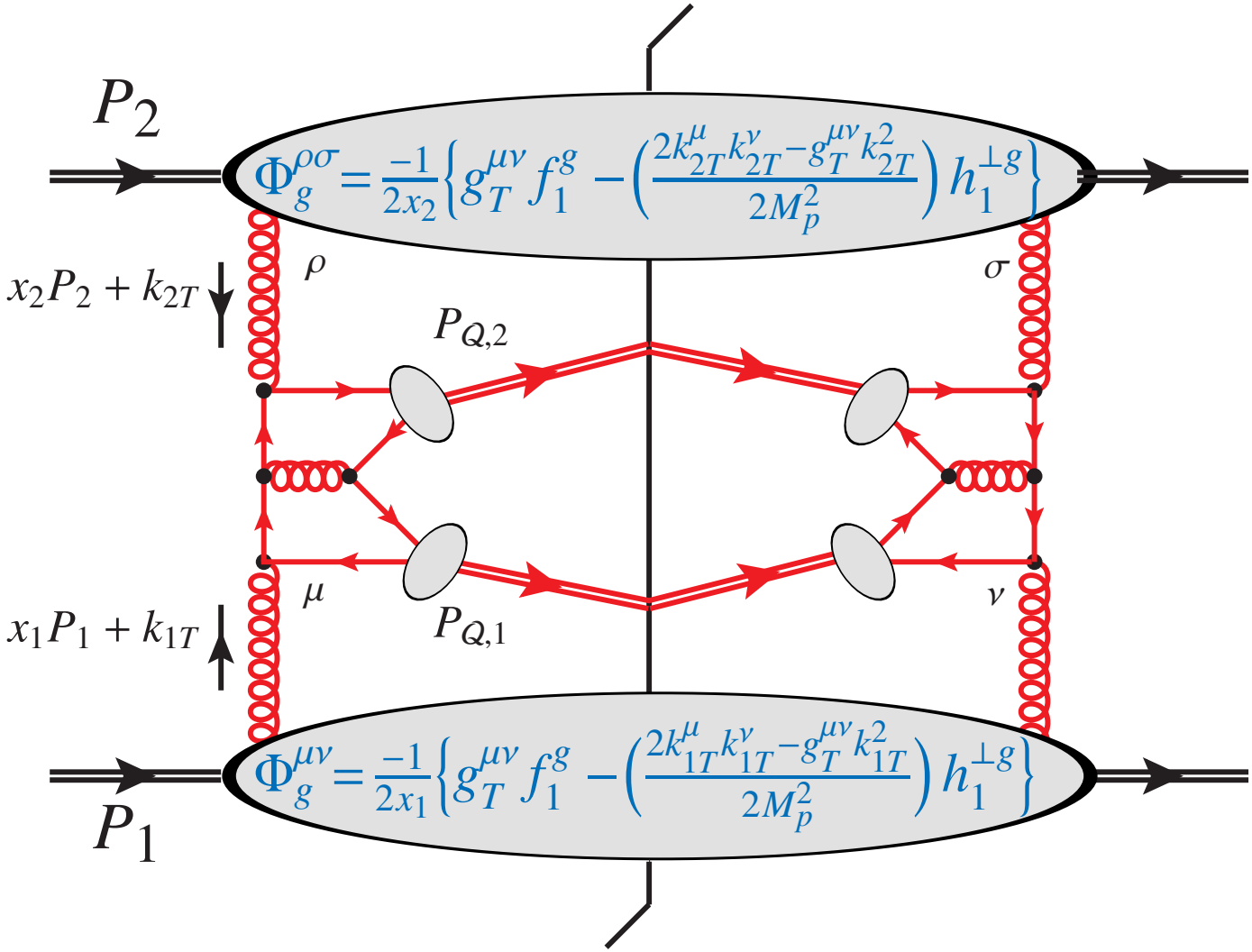
Understanding the internal structure of the nucleon → TMDs
Correlations between k_T and the polarisation of the nucleon/parton
2 components ▸ collinear (x)
▸ transversal ($\vec{k}_\perp$) → generate q_T (final-state)

TMD factorisation ($q_T \ll Q$ hard scale)
General factorised cross section
↪ partonic scattering amplitude (*perturbative*)
↪ k_T-dependent correlators (*non-perturbative*)

$$d\sigma = \int dx_1 dx_2 d^2\vec{k}_{T1} d^2\vec{k}_{T2} \delta^{(2)}(\vec{k}_{T1} + \vec{k}_{T2} - \vec{q}_T)$$
$$\times \Phi_g^{\mu\nu}(x_1, \vec{k}_{T1}) \Phi_g^{\rho\sigma}(x_2, \vec{k}_{T2}) \left[\hat{\mathcal{M}}_{\mu\rho} \hat{\mathcal{M}}^*_{\nu\sigma} \right]_{\substack{k_1 = x_1 P_1 \\ k_2 = x_2 P_2}} + \mathcal{O}\left(\frac{q_T^2}{Q^2}\right)$$

LO Feynman diagram for $p(P_1) + p(P_2) \to \mathcal{Q}(P_{Q,1}) + \mathcal{Q}(P_{Q,1}) + X$

F. Scarpa, D. Boer, M.G. Echevarria, J.-P. Lansberg, M. Schlegel, EPJC (2020) 80:87



[see Nanako's talk]

J.-P. Lansberg, C. Pisano, F. Scarpa, M. Schlegel, PLB 784(2018) 217

The general formula for the cross section of gluon fusion is:

$$
\begin{aligned}
d\sigma^{gg} \propto & F_1 \times \mathcal{C}[f_1^g f_1^g] \\
& + F_2 \times \mathcal{C}[w_2 h_1^{\perp g} h_1^{\perp g}] \\
& + (F_3 \times \mathcal{C}[w_3 f_1^g h_1^{\perp g}] + F_3' \times \mathcal{C}[w_3' h_1^{\perp g} f_1^g]) \cos(2\Phi_{CS}) \\
& + (F_4 \times \mathcal{C}[w_4 h_1^{\perp g} h_1^{\perp g}]) \cos(4\Phi_{CS})
\end{aligned}
$$

- F_i: hard scattering coefficients
- w_i: transverse weights
- f_1^g and $h_1^{\perp g}$: unpolarised and linearly polarised gluon TMDs
- Φ_{CS}: Collins-Soper azimuthal angle

Work in terms of helicities!

J.-P. Lansberg, C. Pisano, M. Schlegel, Nucl.Phys.B 920 (2017) 192-210

$$\rightarrow d\sigma = \frac{(2\pi)^4}{S^2}\, dPS_n \sum_{\lambda_a,\bar{\lambda}_a,\lambda_b,\bar{\lambda}_b=\pm 1} \frac{1}{(N_c^2-1)^2} \times$$

$$\sum_{a,b;I} \mathcal{A}^{ab}_{\lambda_a,\lambda_b;I}(\bar{k}_a, \bar{k}_b; \{P_i\}) \mathcal{A}^{ab*}_{\bar{\lambda}_a,\bar{\lambda}_b;I}(\bar{k}_a, \bar{k}_b; \{P_i\}) \times$$

$$\int d^2k_{aT} \int d^2k_{bT} \delta^{(2)}(d^2k_{aT} + d^2k_{bT} - d^2q_T) \times$$

$$\Phi_{\bar{\lambda}_a,\lambda_a}(x_a, d^2k_{aT}, \zeta_a, \mu) \Phi_{\bar{\lambda}_b,\lambda_b}(x_b, d^2k_{bT}, \zeta_b, \mu) + \mathcal{O}(q_T/Q)$$

$$\Phi_{\lambda_a,\bar{\lambda}_a} = \frac{1}{2x_a} \left(\delta_{\lambda_a,\bar{\lambda}_a}\ f_1^g + \frac{k_{ax}^2 - k_{ay}^2 - 2i\lambda_a k_{ax} k_{ay}}{2M^2} \delta_{\lambda_a,-\bar{\lambda}_a} h_1^{\perp g} \right)$$

Summary

Implementation of quarkonia in MG5: in progress!

- ⋆ **Colour projectors** — implemented
- ⋆ **Spin-projectors** — implemented
- ⋆ **Interface** — implemented
- ⋆ Benchmarked our matrix elements squared against Helac-Onia
- **Phase-space adaptation** in progress!
- Application: $J/\psi + H$ hadroproduction arXiv:2109.02025

outlook

- **P-wave** extension
- From LO to **NLO**! arXiv:2402.19221
- **TMD factorisation**

Backup slides

(1) Phase-space integration: VEGAS algorithm

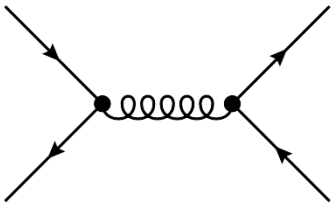
slide courtesy of L. Simon

Approximation of probability distribution [edit]

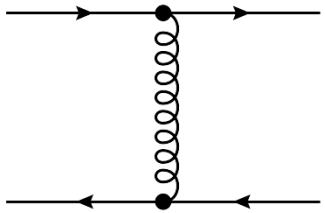
The VEGAS algorithm approximates the exact distribution by making a number of passes over the integration region while histogramming the function f. Each histogram is used to define a sampling distribution for the next pass. Asymptotically this procedure converges to the desired distribution. In order to avoid the number of histogram bins growing like K^d with dimension *d* the probability distribution is approximated by a separable function: $g(x_1, x_2, \ldots) = g_1(x_1)g_2(x_2)\cdots$ so that the number of bins required is only *Kd*. This is equivalent to locating the peaks of the function from the projections of the integrand onto the coordinate axes. The efficiency of VEGAS depends on the validity of this assumption. It is most efficient when the peaks of the integrand are well-localized. If an integrand can be rewritten in a form which is approximately separable this will increase the efficiency of integration with VEGAS.

[wikipedia]

Example: $q\bar{q} \to q\bar{q}$



peak in s



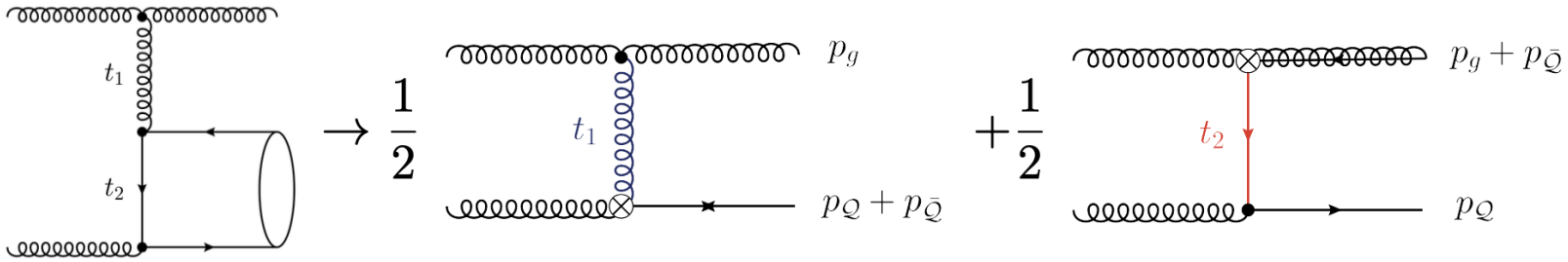
peak in t

[Diagrams drawn with FeynGame]

(2) Phase-space integration: quarkonium production

slide courtesy of L. Simon

Example: $gg \to \mathcal{Q}\bar{\mathcal{Q}}[n]g$



Issue: Two t-channel peaks, but only one can be integration parameter

$$\text{either} \quad \int \mathrm{d}\Pi_2 \sim \int \mathrm{d}t_1 \quad \text{or} \quad \int \mathrm{d}\Pi_2 \sim \int \mathrm{d}t_2$$

Solution:

$$\int \mathrm{d}\Pi_2 = \frac{1}{2} \int \underbrace{\mathrm{d}\Pi_2}_{\sim \mathrm{d}t_1} + \frac{1}{2} \int \underbrace{\mathrm{d}\Pi_2}_{\sim \mathrm{d}t_2}$$

- Momentum $p_{\mathcal{Q}} + p_{\bar{\mathcal{Q}}}$ of bound state is integration parameter
- Momentum $p_{\mathcal{Q}}$ of bound state's parton is integration parameter
  - $\to$ new parametrisation is required