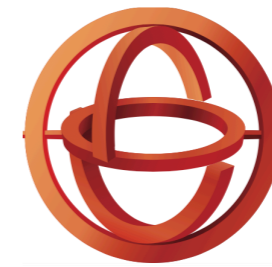




NATIONAL CENTRE FOR
SCIENTIFIC RESEARCH "DEMOKRITOS"
INSTITUTE OF NUCLEAR AND PARTICLE PHYSICS



H.F.R.I.
Hellenic Foundation for
Research & Innovation

Progress towards numerical two-loop integrand reduction

Giuseppe Bevilacqua
NCSR "Demokritos"

2nd APCTP - INPP Demokritos meeting

APCTP (Pohang), 29th August 2025

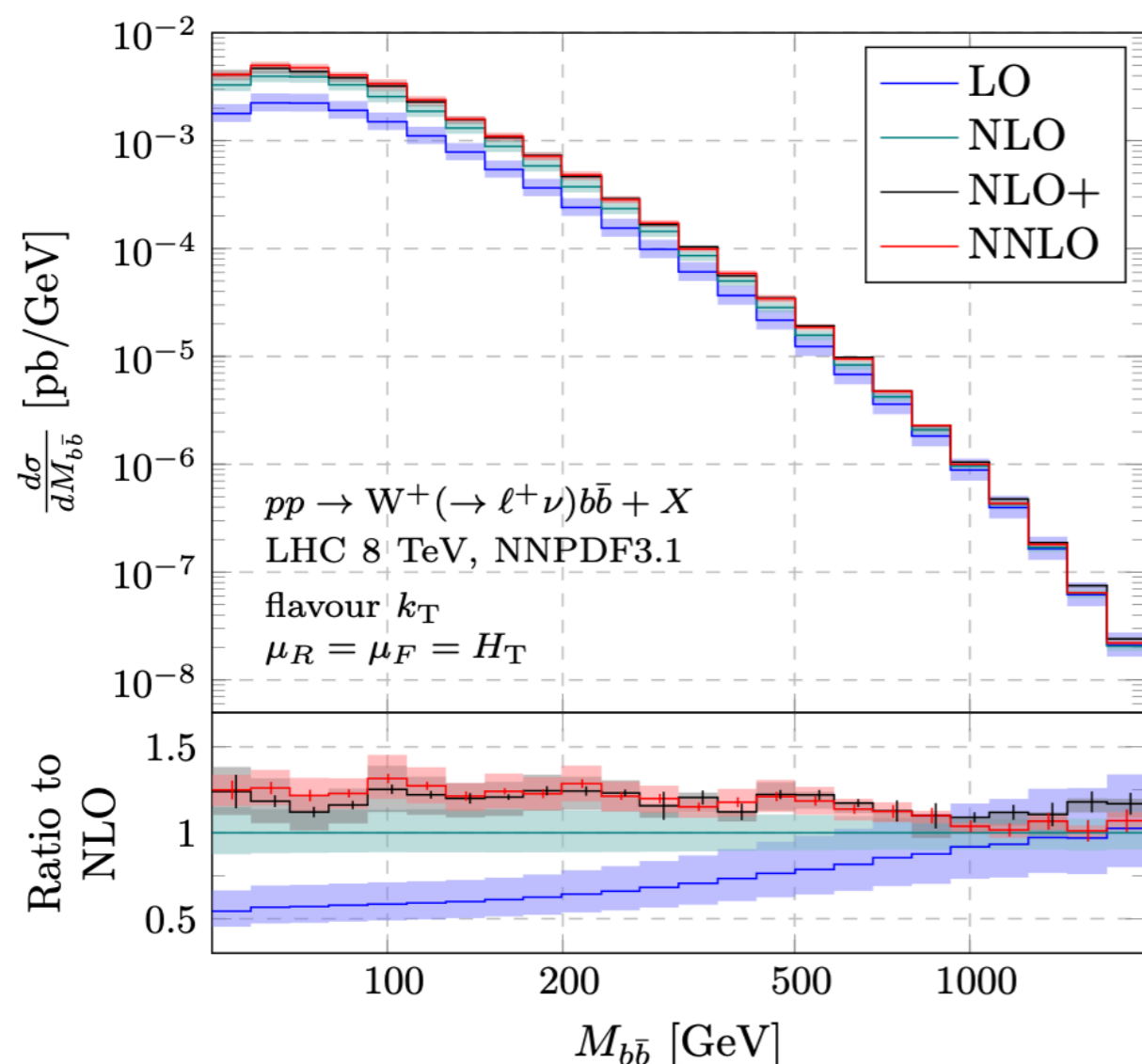
In collaboration with D. Canko, C. Papadopoulos and A. Spourdalakis

Based on [arXiv:2506.07231 \[hep-ph\]](https://arxiv.org/abs/2506.07231) + ongoing work

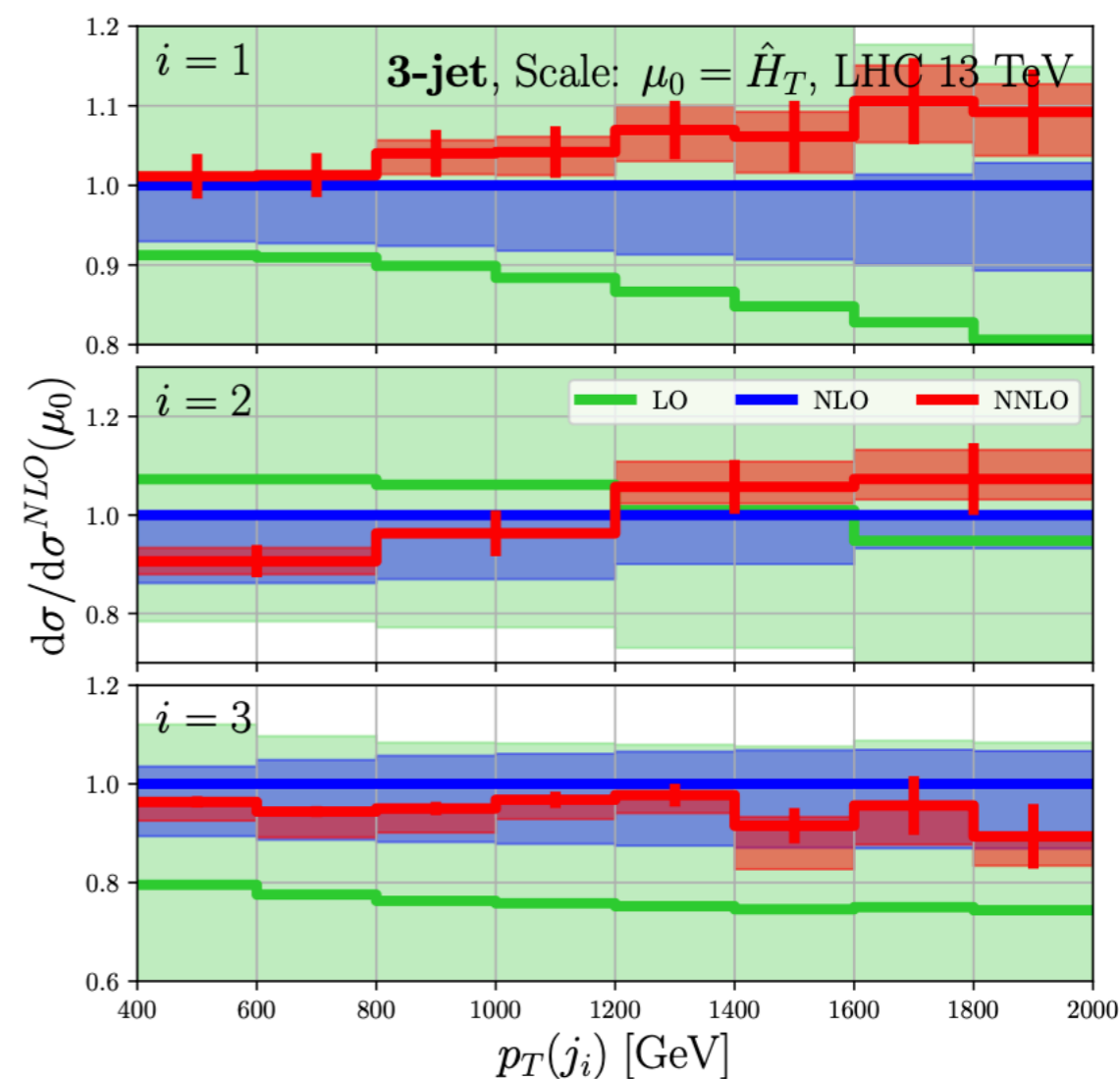
Introduction

- After the “NLO revolution” has set, **NNLO** is becoming the new theory standard for many QCD processes at the LHC

[Hartanto, Poncelet, Popescu and Zoia '22]



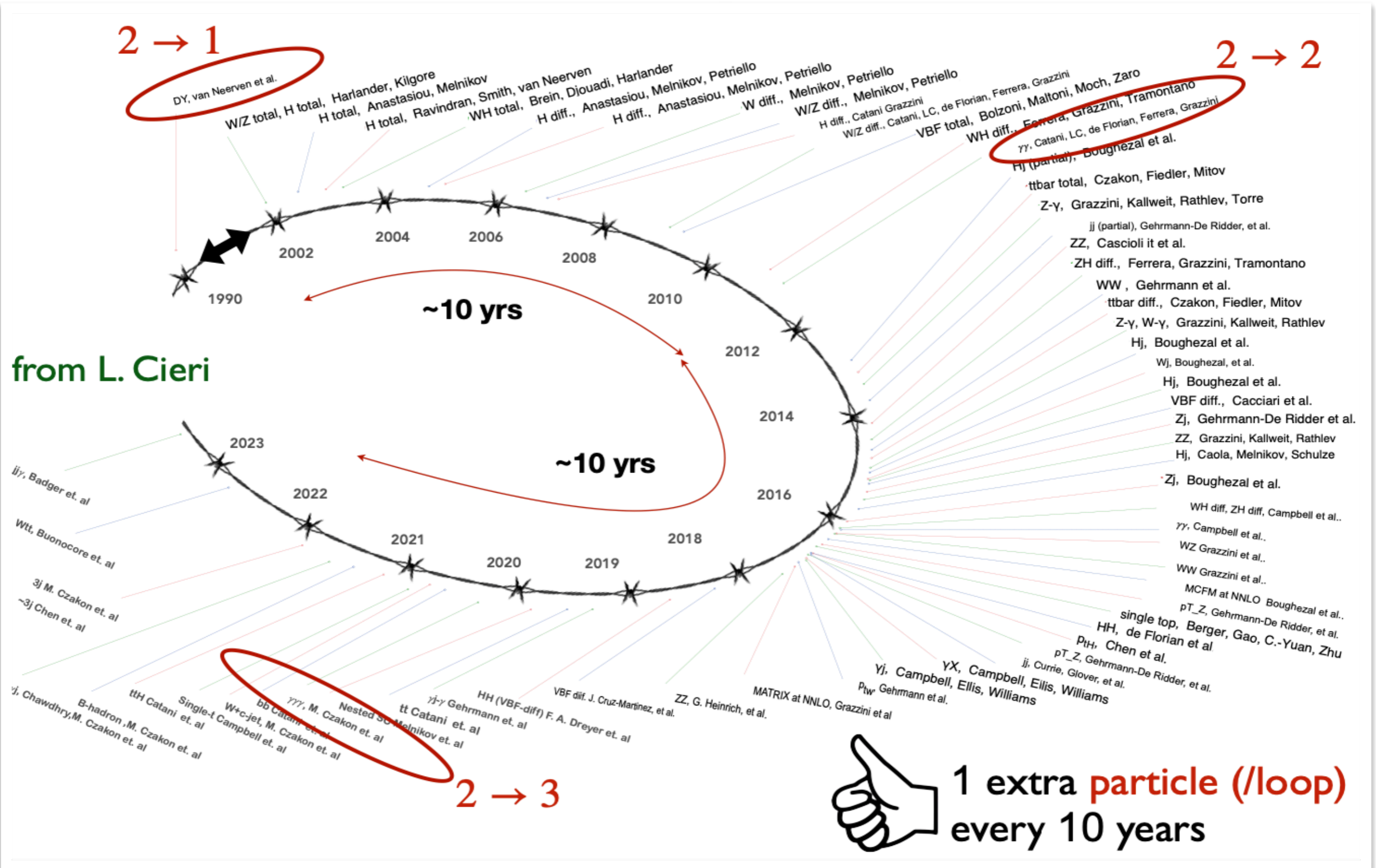
{Czakon, Mitov and Poncelet '21}



NNLO is crucial to achieve *percent* level precision for most LHC processes

Progress in NNLO calculations

[Slide by D. de Florian, EPS-HEP 2023]



Progress in NNLO calculations

[Slide by D. de Florian, EPS-HEP 2023]

2 → 1

2 → 2

- Fostering *automation* is key to accelerate progress in NNLO phenomenology

↪ see “NLO revolution”

2 → 3



1 extra **particle (/loop)**
every 10 years

from L. C.

ij, Badger et. al
Wt, Buonocore et. al
3j M. Czakon et. al
~3j Chen et. al
B-hadron, M. Czakon et. al
ttH Catani et. al
Single-t Catani et. al
W+c-jet, M. Czakon et. al
bb Catani et. al
γγ, M. Czakon et. al
Nested SO, Melnikov et. al
tt Catani et. al
γ-γ Gehrman et. al
HH (VBF-diff) F. A. Dreyer et. al
VBF diff. J. Cruz-Martinez, et al.
ZZ, G. Heinrich, et al.
MATRIX at NNLO, Grazzini et al
Yj, Campbell, Ellis, Williams
YX, Campbell, Ellis, Williams
jj, Currie, Glover, et al.
pT_Z, Gehrman-De Ridder, et al.
single top, Berger, Gao, C.-Yuan, Zhu
HH, de Florian et al
PtH, Chen et al.
pT_Z, Gehrman-De Ridder, et al.

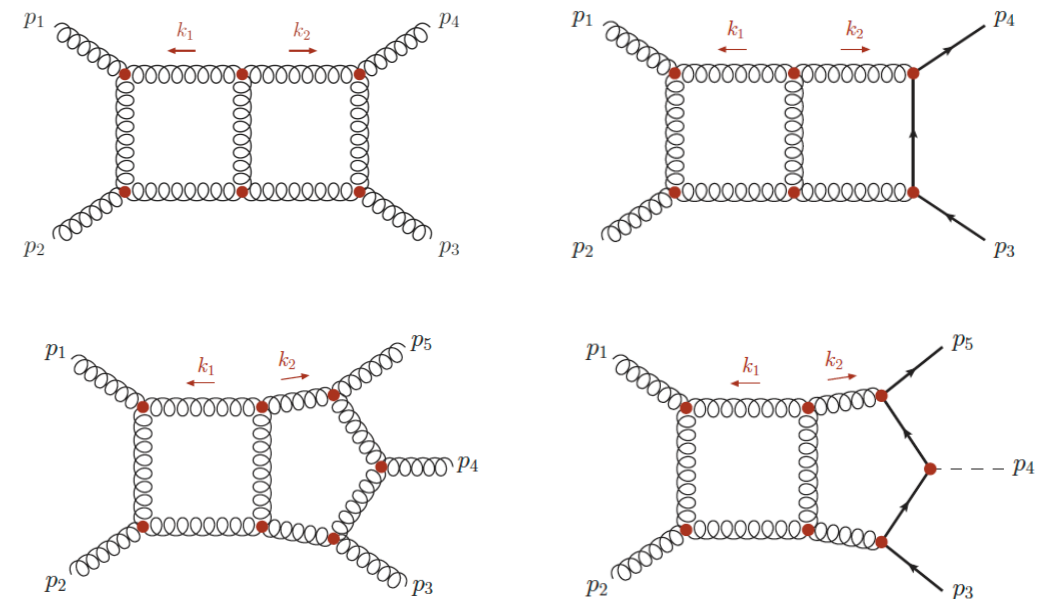
DY, van Neerven et al.
W/Z total, H total, Harlander, Kilgore
H total, Anastasiou, Melnikov
H total, Ravindran, Smith, van Neerven
WH total, Brein, Djouadi, Harlander
H diff., Anastasiou, Melnikov, Petriello
H diff., Anastasiou, Melnikov, Petriello
W diff., Melnikov, Petriello
W/Z diff., Melnikov, Petriello
H diff., Catani, Grazzini
W/Z diff., Catani, LC, de Florian, Ferrara, Grazzini
VBF total, Bolzoni, Maltoni, Moch, Zaro
WH diff., Ferrara, Grazzini, Tramontano
γγ, Catani, LC, de Florian, Ferrara, Grazzini
Hj (partial), Boughezal et al.
ttbar total, Czakon, Fiedler, Mitov
Z-γ, Grazzini, Kallweit, Rathlev, Torre
jj (partial), Gehrmann-De Ridder, et al.
ZZ, Cascioli et al.
ZH diff., Ferrara, Grazzini, Tramontano
WW, Gehrmann et al.
ttbar diff., Czakon, Fiedler, Mitov
Z-γ, W-γ, Grazzini, Kallweit, Rathlev
Hj, Boughezal et al.
Wj, Boughezal, et al.
Hj, Boughezal et al.
VBF diff., Cacciari et al.
Zj, Gehrmann-De Ridder et al.
ZZ, Grazzini, Kallweit, Rathlev
Hj, Caola, Melnikov, Schulze
Zj, Boughezal et al.
WH diff, ZH diff, Campbell et al..
γγ, Campbell et al..
WZ Grazzini et al..
WW Grazzini et al..
MCFM at NNLO Boughezal et al..
pT_Z, Gehrmann-De Ridder, et al.
PtH, Chen et al.
pT_Z, Gehrman-De Ridder, et al.

Ingredients to NNLO calculations

$$\sigma_{NNLO} = \int_m d\Phi_m \left(2\text{Re}(M_m^{(0)*} M_m^{(2)}) + |M_m^{(1)}|^2 \right) J_m(\Phi) \quad [\text{VV}]$$

$$+ \int_{m+1} d\Phi_{m+1} \left(2\text{Re} \left(M_{m+1}^{(0)*} M_{m+1}^{(1)} \right) \right) J_{m+1}(\Phi) \quad [\text{RV}]$$

$$+ \int_{m+2} d\Phi_{m+2} |M_{m+2}^{(0)}|^2 J_{m+2}(\Phi) \quad [\text{RR}]$$



- **2-loop amplitudes** are crucial ingredients (and often bottlenecks) for NNLO

- *Construction of 2-loop integrands*

Feynman diagrams | Dyson-Schwinger recursion

[Nogueira, Hahn, Ita, Pozzorini, Schar, Zoller, Papadopoulos, ...]

- *Reduction of 2-loop integrals*

Integrand reduction | IBP reduction + Finite Fields

[Ita, Pozzorini, Schar, Zoller, Maierhofer, Lange, Usovitsch, Smirnov, von Manteuffel, Studerus, Peraro, Tancredi, Anastasiou, Mastrolia, Zhang, Badger, Chetyrkin, Tkachov, Laporta, ...]

- *Calculation of MI's*

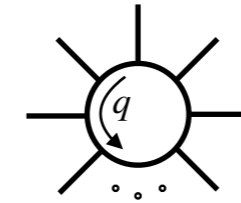
Analytic | Numerical | Semi-numerical

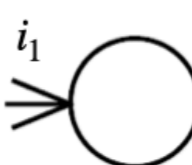
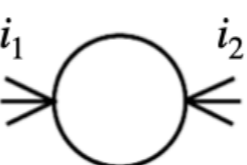
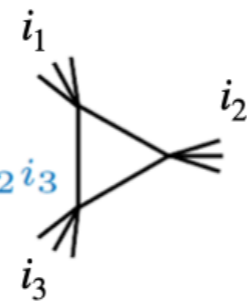
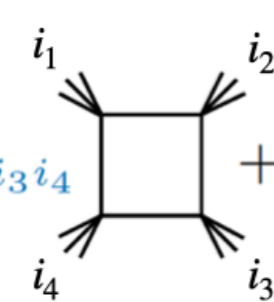
[Binoth, Heinrich, Gehrmann, Remiddi, Henn, Huang, ...]

1-loop legacy: integrand-level reduction

OPP reduction method in a nutshell

{Ossola, Papadopoulos and Pittau, [Nucl.Phys.B 763 \(2007\) 147-169](#)}

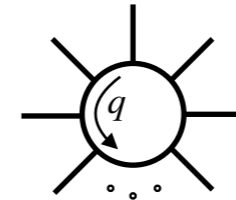

$$\rightarrow \mathcal{A}_m = \int \frac{d^d q}{(2\pi)^d} \frac{N(q)}{D_0 D_1 \dots D_{m-1}}$$

$$\mathcal{A}_m = \sum d_{i_1 i_2 i_3 i_4} \text{[square diagram]} + \sum c_{i_1 i_2 i_3} \text{[triangle diagram]} + \sum b_{i_1 i_2} \text{[circle with 2 legs]} + \sum a_{i_1} \text{[circle with 1 leg]}$$


1-loop legacy: integrand-level reduction

OPP reduction method in a nutshell

{Ossola, Papadopoulos and Pittau, [Nucl.Phys.B 763 \(2007\) 147-169](#)}



$$\rightarrow \mathcal{A}_m = \int \frac{d^d q}{(2\pi)^d} \frac{N(q)}{D_0 D_1 \dots D_{m-1}}$$

$$\mathcal{A}_m = \sum d_{i_1 i_2 i_3 i_4} \text{[square diagram]} + \sum c_{i_1 i_2 i_3} \text{[triangle diagram]} + \sum b_{i_1 i_2} \text{[circle diagram]} + \sum a_{i_1} \text{[circle diagram]}$$

$$\begin{aligned} N(q) = & \sum_{i_0 < i_1 < i_2 < i_3}^{m-1} [d(i_0, i_1, i_2, i_3) + \tilde{d}(q; i_0, i_1, i_2, i_3)] \prod_{i \neq i_0, i_1, i_2, i_3}^{m-1} D_i \\ & + \sum_{i_0 < i_1 < i_2}^{m-1} [c(i_0, i_1, i_2) + \tilde{c}(q; i_0, i_1, i_2)] \prod_{i \neq i_0, i_1, i_2}^{m-1} D_i \\ & + \sum_{i_0 < i_1}^{m-1} [b(i_0, i_1) + \tilde{b}(q; i_0, i_1)] \prod_{i \neq i_0, i_1}^{m-1} D_i \\ & + \sum_{i_0}^{m-1} [a(i_0) + \tilde{a}(q; i_0)] \prod_{i \neq i_0}^{m-1} D_i \\ & + \tilde{P}(q) \prod_i^{m-1} D_i \end{aligned}$$

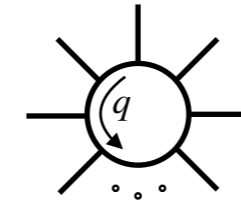
Sample $N(q)$ numerically using different values of q to extract coefficients

purely algebraic problem

1-loop legacy: integrand-level reduction

OPP reduction method in a nutshell

{Ossola, Papadopoulos and Pittau, [Nucl.Phys.B 763 \(2007\) 147-169](#)}



$$\rightarrow \mathcal{A}_m = \int \frac{d^d q}{(2\pi)^d} \frac{\bar{N}(q)}{\bar{D}_0 \bar{D}_1 \dots \bar{D}_{m-1}}$$

$$\mathcal{A}_m = \sum d_{i_1 i_2 i_3 i_4} \text{ (square) } + \sum c_{i_1 i_2 i_3} \text{ (triangle) } + \sum b_{i_1 i_2} \text{ (circle) } + \sum a_{i_1} \text{ (circle) }$$

- In dimensional regularisation, the loop integrand is defined in $d = 4 - 2\epsilon$ dimensions

Dimensional regularisation ('t Hooft-Veltman scheme)

$$d = 4 - 2\epsilon$$

$$\bar{q}^2 = q^2 + \tilde{q}^2$$

|||
 μ

$$\bar{\gamma}^\mu = \gamma^\mu + \tilde{\gamma}^\mu$$

$$\hookrightarrow \bar{\gamma}^\mu \bar{\gamma}_\mu = d = 4 - 2\epsilon$$

$$\bar{g}^{\mu\nu} = g^{\mu\nu} + \tilde{g}^{\mu\nu}$$

$$\hookrightarrow \bar{g}^{\mu\nu} \bar{g}_{\mu\nu} = d = 4 - 2\epsilon$$

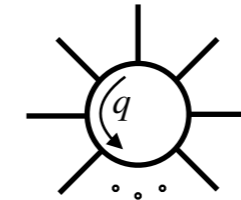
$$\Rightarrow \bar{D}_i^2 = D_i^2 + \mu$$

$$\bar{N}(\bar{q}) = N(q) + \tilde{N}(\tilde{q})$$

1-loop legacy: integrand-level reduction

OPP reduction method in a nutshell

{Ossola, Papadopoulos and Pittau, [Nucl.Phys.B 763 \(2007\) 147-169](#)}



$$\rightarrow \mathcal{A}_m = \int \frac{d^d q}{(2\pi)^d} \frac{\bar{N}(q)}{\bar{D}_0 \bar{D}_1 \dots \bar{D}_{m-1}}$$

$$\mathcal{A}_m = \underbrace{\sum d_{i_1 i_2 i_3 i_4} \text{ (square)} + \sum c_{i_1 i_2 i_3} \text{ (triangle)} + \sum b_{i_1 i_2} \text{ (circle)} + \sum a_{i_1} \text{ (circle)}}_{\text{“Cut-constructible” part}} + \underbrace{R_1 + R_2}_{\text{“Rational Terms”}}$$

- In dimensional regularisation, the loop integrand is defined in $d = 4 - 2\epsilon$ dimensions

Dimensional regularisation ('t Hooft-Veltman scheme)

$d = 4 - 2\epsilon$

$\bar{q}^2 = q^2 + \tilde{q}^2$ $\quad \quad \quad \parallel$ $\quad \quad \quad \mu$	$\bar{\gamma}^\mu = \gamma^\mu + \tilde{\gamma}^\mu$ $\hookrightarrow \bar{\gamma}^\mu \bar{\gamma}_\mu = d = 4 - 2\epsilon$	$\bar{g}^{\mu\nu} = g^{\mu\nu} + \tilde{g}^{\mu\nu}$ $\hookrightarrow \bar{g}^{\mu\nu} \bar{g}_{\mu\nu} = d = 4 - 2\epsilon$
$\Rightarrow \bar{D}_i^2 = D_i^2 + \mu$	$\bar{N}(\bar{q}) = N(q) + \tilde{N}(\tilde{q})$	

- Typically, *numerical* ME generators compute numerators in $d = 4$
- If so, the mismatch must be compensated by adding **Rational Terms** (R_1, R_2)

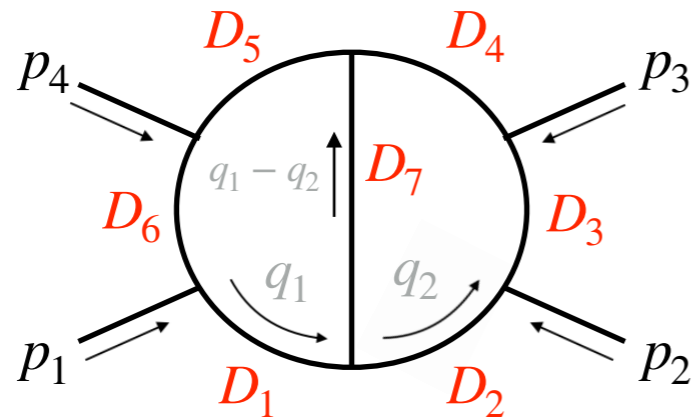
Roadmap to 2-loop reduction

- So far, no complete framework for *automated numerical 2-loop* computations (as compared to the 1-loop case) is available*
 - (*) ongoing efforts by HELAC and OpenLoops collaborations
- The path to 2-loop automation requires coordinated efforts in:
 - Extending *integrand-level reduction* to 2-loop calculations
 - Developing a numerical framework for the reduction method → “CutTools2”
 - Building 2-loop amplitude *integrands* → HELAC-2LOOP, OpenLoops
 - Evaluating Feynman integrals numerically (using e.g. IBP...)

In this talk we will discuss ongoing efforts in these directions, with emphasis on *integrand-level reduction*

Anatomy of 2-loop amplitudes

- Example: 4-point function



$$\rightarrow A_{2L} = \int d^d \bar{q}_1 d^d \bar{q}_2 \frac{N(\bar{q}_1, \bar{q}_2, \{p\})}{\prod_{i=1}^7 D_i(\bar{q}_1, \bar{q}_2, \{p\})}$$

$$\begin{aligned} D_1 &= \bar{q}_1^2 \\ D_2 &= \bar{q}_2^2 \\ D_3 &= (\bar{q}_2 + p_2)^2 \\ D_4 &= (\bar{q}_2 + p_2 + p_3)^2 \\ D_5 &= (\bar{q}_1 + p_2 + p_3)^2 \\ D_6 &= (\bar{q}_1 + p_2 + p_3 + p_4)^2 \\ D_7 &= (\bar{q}_1 - \bar{q}_2)^2 \end{aligned}$$

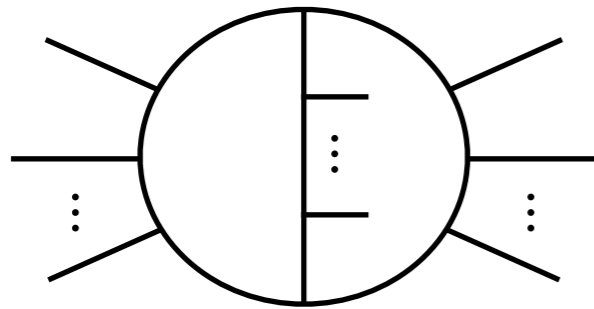
The Numerator is a function of few constituents (Lorentz scalars):

$$(\bar{q}_i \cdot \bar{q}_j), (\bar{q}_i \cdot p_j), (p_i \cdot p_j), (\bar{q}_i \cdot \eta)$$

- Some of the scalars above can be decomposed in terms of D_i 's \rightarrow *Reducible Scalar Products (RSP)*
- Other scalars do *not* allow such decomposition in terms of D_i 's \rightarrow *Irreducible Scalar Products (ISP)*

Take home message

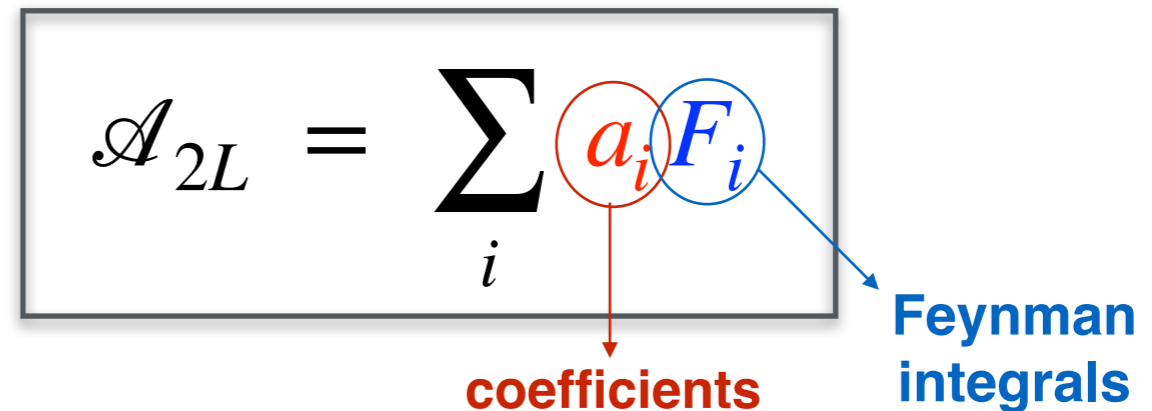
- A generic 2-loop amplitude can be *reduced* into a set of simpler Feynman integrals:



$$\mathcal{A}_{2L} = \sum_i a_i F_i$$

coefficients

Feynman integrals



- If we know how to:

- determine the **coefficients**, → [Task 1]
- decompose all **Feynman integrals** into *Master Integrals* (using IBP relations), → [Task 2]

then we know how to compute \mathcal{A}_{2L}

- Goal: establish a *numerical* method which allows to perform the above tasks in *automated* form and for *arbitrary* processes (similarly to e.g. OPP method at 1-loop)

Focus of this talk → extracting **coefficients** at *integrand* level

Parametrising the Numerator

- The first step is to build a general parametrisation of the Numerator:

$$N(\bar{q}_1, \bar{q}_2, \{p\}) = P^{(m)} + \sum_i P_i^{(m-1)} D_i + \sum_{i<j} P_{ij}^{(m-2)} D_i D_j + \sum_{i<j<k} P_{ij}^{(m-3)} D_i D_j D_k + \dots$$

$$A_{2L} = \int d^d \bar{q}_1 d^d \bar{q}_2 \frac{N(\bar{q}_1, \bar{q}_2, \{p\})}{\prod_{i=1}^7 D_i(\bar{q}_1, \bar{q}_2, \{p\})}$$

Parametrising the Numerator

- The first step is to build a general parametrisation of the Numerator:

$$N(\bar{q}_1, \bar{q}_2, \{p\}) = P^{(m)} + \sum_i P_i^{(m-1)} D_i + \sum_{i<j} P_{ij}^{(m-2)} D_i D_j + \sum_{i<j<k} P_{ij}^{(m-3)} D_i D_j D_k + \dots$$

- The $P^{(x)}$'s are *polynomials* built out of the available ISP's :

$$P^{(x)} = \sum_k c_k M_k(\bar{q}_1, \bar{q}_2, \{p\}) \quad \rightarrow \quad \text{e.g.: } M_k = (\bar{q}_1 \cdot p_2) (\bar{q}_2 \cdot p_4)^3$$

$$A_{2L} = \int d^d \bar{q}_1 d^d \bar{q}_2 \frac{N(\bar{q}_1, \bar{q}_2, \{p\})}{\prod_{i=1}^7 D_i(\bar{q}_1, \bar{q}_2, \{p\})}$$

Parametrising the Numerator

- The first step is to build a general parametrisation of the Numerator:

$$N(\bar{q}_1, \bar{q}_2, \{p\}) = P^{(m)} + \sum_i P_i^{(m-1)} D_i + \sum_{i<j} P_{ij}^{(m-2)} D_i D_j + \sum_{i<j<k} P_{ij}^{(m-3)} D_i D_j D_k + \dots$$

- The $P^{(x)}$'s are *polynomials* built out of the available ISP's :

$$A_{2L} = \int d^d \bar{q}_1 d^d \bar{q}_2 \frac{N(\bar{q}_1, \bar{q}_2, \{p\})}{\prod_{i=1}^7 D_i(\bar{q}_1, \bar{q}_2, \{p\})}$$

$$P^{(x)} = \sum_k c_k M_k(\bar{q}_1, \bar{q}_2, \{p\}) \quad \rightarrow \quad \text{e.g.: } M_k = (\bar{q}_1 \cdot p_2) (\bar{q}_2 \cdot p_4)^3$$

- The coefficients c_k , that we want to extract, are functions of external momenta:

$$c_k = c_k(\{p\})$$

Parametrising the Numerator

- The first step is to build a general parametrisation of the Numerator:

$$N(\bar{q}_1, \bar{q}_2, \{p\}) = P^{(m)} + \sum_i P_i^{(m-1)} D_i + \sum_{i<j} P_{ij}^{(m-2)} D_i D_j + \sum_{i<j<k} P_{ij}^{(m-3)} D_i D_j D_k + \dots$$

- The $P^{(x)}$'s are *polynomials* built out of the available ISP's :

$$A_{2L} = \int d^d \bar{q}_1 d^d \bar{q}_2 \frac{N(\bar{q}_1, \bar{q}_2, \{p\})}{\prod_{i=1}^7 D_i(\bar{q}_1, \bar{q}_2, \{p\})}$$

$$P^{(x)} = \sum_k c_k M_k(\bar{q}_1, \bar{q}_2, \{p\}) \quad \rightarrow \quad \text{e.g.: } M_k = (\bar{q}_1 \cdot p_2) (\bar{q}_2 \cdot p_4)^3$$

- The coefficients c_k , that we want to extract, are functions of external momenta:

$$c_k = c_k(\{p\})$$

- We need an *ansatz* to characterise the polynomials:

- How many terms ?
- Analytic structure of each term ? \rightarrow using **BasisDet** [Zhang '12]
- Maximal power ?

Fitting coefficients

$$N(\bar{q}_1, \bar{q}_2, \{p\}) = P^{(m)} + \sum_i P_i^{(m-1)} D_i + \sum_{i<j} P_{ij}^{(m-2)} D_i D_j + \sum_{i<j<k} P_{ij}^{(m-3)} D_i D_j D_k + \dots$$

Once the ansatz is established, fit the coefficients (c_k)

$$P^{(x)} = \sum_k c_k M_k(\bar{q}_1, \bar{q}_2, \{p\})$$



Fitting coefficients

$$N(\bar{q}_1, \bar{q}_2, \{p\}) = P^{(m)} + \sum_i P_i^{(m-1)} D_i + \sum_{i<j} P_{ij}^{(m-2)} D_i D_j + \sum_{i<j<k} P_{ij}^{(m-3)} D_i D_j D_k + \dots$$

Once the ansatz is established, fit the coefficients (c_k)

$$P^{(x)} = \sum_k c_k M_k(\bar{q}_1, \bar{q}_2, \{p\})$$

• Method n.1: “global fit”



• Method n.2: “fit by cut”

(aka “recursive fit”)

Fitting coefficients

$$N(\bar{q}_1, \bar{q}_2, \{p\}) = P^{(m)} + \sum_i P_i^{(m-1)} D_i + \sum_{i<j} P_{ij}^{(m-2)} D_i D_j + \sum_{i<j<k} P_{ij}^{(m-3)} D_i D_j D_k + \dots$$

Once the ansatz is established, fit the coefficients (c_k)

$$P^{(x)} = \sum_k c_k M_k(\bar{q}_1, \bar{q}_2, \{p\})$$

- **Method n.l: “global fit”**

- given the external kinematics ($\{p\}$), sample $N(\bar{q}_1, \bar{q}_2, \{p\})$ with *random* values of $\{\bar{q}_1, \bar{q}_2\}$

$$\begin{pmatrix} N_1 \\ \vdots \\ N_m \end{pmatrix} = \begin{pmatrix} M_{11} & \cdots & M_{1m} \\ & \ddots & \\ M_{m1} & \cdots & M_{mm} \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_m \end{pmatrix}$$

Linear system

- coefficients of all polynomials are extracted at the same time
- *huge* system of equations! Not the most efficient method, but feasible numerically

↪ using e.g. **LAPACK** or **Eigen**

Fitting coefficients

$$N(\bar{q}_1, \bar{q}_2, \{p\}) = P^{(m)} + \sum_i P_i^{(m-1)} D_i + \sum_{i<j} P_{ij}^{(m-2)} D_i D_j + \sum_{i<j<k} P_{ij}^{(m-3)} D_i D_j D_k + \dots$$

Once the ansatz is established, fit the coefficients (c_k)

$$P^{(x)} = \sum_k c_k M_k(\bar{q}_1, \bar{q}_2, \{p\})$$

- Method n.2: “fit by cut”

- Sample N using sets of $\{\bar{q}_1, \bar{q}_2\}$ such that all D_k 's vanish (\rightarrow “*maximal-cuts*”). Extract coefficients of $P^{(m)}$;

Fitting coefficients

$$N(\bar{q}_1, \bar{q}_2, \{p\}) - P^{(m)} = \sum_i P_i^{(m-1)} D_i + \sum_{i<j} P_{ij}^{(m-2)} D_i D_j + \sum_{i<j<k} P_{ij}^{(m-3)} D_i D_j D_k + \dots$$

Once the ansatz is established, fit the coefficients (c_k)

$$P^{(x)} = \sum_k c_k M_k(\bar{q}_1, \bar{q}_2, \{p\})$$

• Method n.2: “fit by cut”

- Sample N using sets of $\{\bar{q}_1, \bar{q}_2\}$ such that all D_k 's vanish (\rightarrow “maximal-cuts”). Extract coefficients of $P^{(m)}$;
- Sample $N - P^{(m)}$ using sets of $\{\bar{q}_1, \bar{q}_2\}$ such that all D_k 's except D_i vanish (\rightarrow “next-to-maximal cuts”). Extract coefficients of $P_i^{(m-1)}$;

Fitting coefficients

$$N(\bar{q}_1, \bar{q}_2, \{p\}) - P^{(m)} - \sum_i P_i^{(m-1)} D_i = \sum_{i<j} P_{ij}^{(m-2)} D_i D_j + \sum_{i<j<k} P_{ij}^{(m-3)} D_i D_j D_k + \dots$$

Once the ansatz is established, fit the coefficients (c_k)

$$P^{(x)} = \sum_k c_k M_k(\bar{q}_1, \bar{q}_2, \{p\})$$

• Method n.2: “fit by cut”

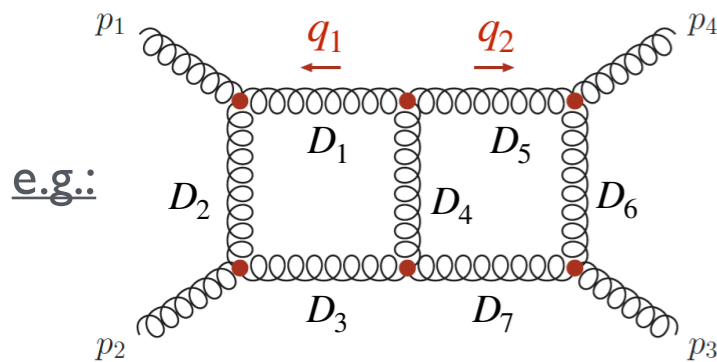
- Sample N using sets of $\{\bar{q}_1, \bar{q}_2\}$ such that all D_k 's vanish (\rightarrow “maximal-cuts”). Extract coefficients of $P^{(m)}$;
- Sample $N - P^{(m)}$ using sets of $\{\bar{q}_1, \bar{q}_2\}$ such that all D_k 's except D_i vanish (\rightarrow “next-to-maximal cuts”). Extract coefficients of $P_i^{(m-1)}$;
- \vdots
- Proceed iteratively till all polynomials are reconstructed

\hookrightarrow Consistency check: *original Numerator* = *ansatz Numerator* (“ $N = N$ test”)

Constraints from cut equations

$$N(\bar{q}_1, \bar{q}_2, \{p\}) = P^{(m)} + \sum_{i=1}^7 P_i^{(m-1)} D_i + \sum_{i<j}^7 P_{ij}^{(m-2)} D_i D_j + \sum_{i<j<k}^7 P_{ij}^{(m-3)} D_i D_j D_k + \dots$$

- Cut equations constrain the ISP's appearing in the polynomials



$$D_1 = q_1^2, \quad D_2 = (q_1 + p_1)^2, \quad D_3 = (q_1 + p_{12})^2, \quad D_4 = (q_1 + q_2)^2, \quad D_5 = q_2^2, \\ D_6 = (q_2 - p_{123})^2, \quad D_7 = (q_2 - p_{12})^2$$

- Maximal cut equation: $D_1 = D_2 = D_3 = D_4 = D_5 = D_6 = D_7 = 0$ $s \equiv (p_1 + p_2)^2$

Constrains
7 of 11
scal. products

$$(q_1 \cdot q_1) = 0, \quad (q_1 \cdot q_2) = 0, \quad (q_1 \cdot p_1) = 0, \quad (q_1 \cdot p_2) = -\frac{s}{2}, \quad (q_2 \cdot q_2) = 0, \\ (q_2 \cdot p_2) = \frac{s}{2} - (q_2 \cdot p_1), \quad (q_2 \cdot p_3) = -\frac{s}{2}$$

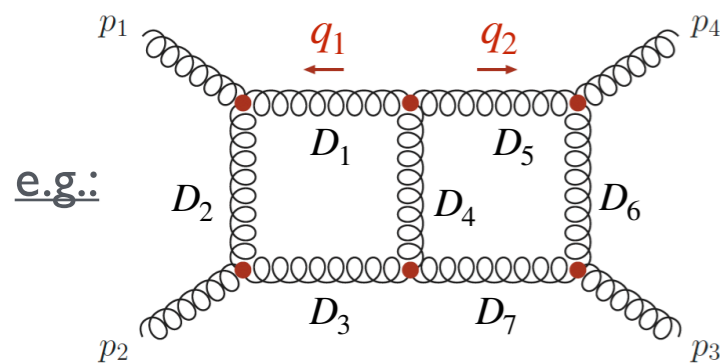
- The more D_i 's are set to zero, the more ISP's are constrained

↪ Polynomial $P^{(m)}$ is simpler than $P_i^{(m-1)}$, which in turn is simpler than $P_{ij}^{(m-2)}$ etc ...

Projecting over the full family of propagators

$$N(\bar{q}_1, \bar{q}_2, \{p\}) = P^{(m)} + \sum_{i=1}^9 P_i^{(m-1)} D_i + \sum_{i<j}^9 P_{ij}^{(m-2)} D_i D_j + \sum_{i<j<k}^9 P_{ij}^{(m-3)} D_i D_j D_k + \dots$$

- **Definition of family:** enlarged set of D_i 's such that *all scalar products* (including ISP's) can be expressed as combinations of them.



“Family” of propagators (9 in total)

$$D_1 = q_1^2, \quad D_2 = (q_1 + p_1)^2, \quad D_3 = (q_1 + p_{12})^2, \quad D_4 = (q_1 + q_2)^2, \quad D_5 = q_2^2, \\ D_6 = (q_2 - p_{123})^2, \quad D_7 = (q_2 - p_{12})^2, \quad D_8 = (q_2 - p_1)^2, \quad D_9 = (q_1 - p_{123})^2$$

- **Maximal cut equation:** $D_1 = D_2 = D_3 = D_4 = D_5 = D_6 = D_7 = D_8 = D_9 = 0$ $s \equiv (p_1 + p_2)^2$

Constrains
9 of **11**
scal. products

$$(q_1 \cdot q_1) = 0, \quad (q_1 \cdot q_2) = 0, \quad (q_1 \cdot p_1) = 0, \quad (q_1 \cdot p_2) = -\frac{s}{2}, \quad (q_2 \cdot q_2) = 0, \\ (q_2 \cdot p_2) = \frac{s}{2}, \quad (q_2 \cdot p_1) = 0, \quad (q_2 \cdot p_3) = -\frac{s}{2}, \quad (q_1 \cdot p_3) = \frac{s}{2}$$

- Projecting over *family* helps reducing the complexity of *individual* polynomials (at the cost of more cut equations)

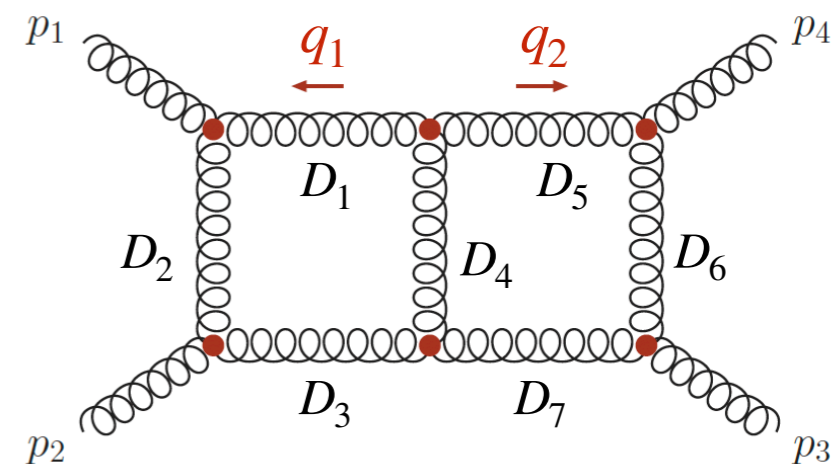
Note: different diagram topologies share the same family

Comparing different cutting approaches

What's the impact on the size of polynomials? Check out *double-box* topology:

“standard cut” approach

Cut level	$N_{\text{polynomials}}$	$\langle N_{\text{terms}} \rangle$
7	1	70.0
6	7	99.3
5	21	46.1
4	35	29.1
3	35	6.4
2	21	0.4



projecting over family

Cut level	$N_{\text{polynomials}}$	$\langle N_{\text{terms}} \rangle$
9	1	13.0
8	9	25.2
7	36	26.8
6	84	17.2
5	126	6.2
4	126	0.9

Tradeoff between number of cut equations and coefficients for each polynomial

[G.B, Canko, Spourdalakis and Papadopoulos, [2506.07231 \[hep-ph\]](https://arxiv.org/abs/2506.07231)]

Approaching cut solutions numerically

- We have devised methods to extract coefficients for the $d = 4$ and $d = 4 - 2\varepsilon$ cases
- If the Numerator is computed *numerically*, we need to build **explicit solutions** for \bar{q}_1, \bar{q}_2

$$N = N(\bar{q}_1, \bar{q}_2, \{p\})$$

Dimensional regularisation \longrightarrow 't Hooft-Veltman scheme

<ul style="list-style-type: none"> • p_1, \dots, p_n (ext. momenta) in $d = 4$ • \bar{q}_1, \bar{q}_2 (loop momenta) in $d = 4 - 2\varepsilon$ <p>$\hookrightarrow \bar{q}_1, \bar{q}_2$ have $8 + 3 = \boxed{11}$ d.o.f.</p>	$\left[\begin{array}{l} \bar{q}_1^2 = q_1^2 + \mu_{11} \\ \bar{q}_2^2 = q_2^2 + \mu_{22} \\ (\bar{q}_1 \cdot \bar{q}_2) = (q_1 \cdot q_2) + \mu_{12} \end{array} \right.$
---	--

- Working in $d = 4 - 2\varepsilon$, there is sufficient freedom to ensure that cut equations, at any level, admit a *unique* parametric solution for \bar{q}_1, \bar{q}_2 :

$$D_1 = D_2 = \dots = D_i = 0 \quad \Rightarrow \quad q_1 = q_1(\{p\}, \vec{x}), \quad q_2 = q_2(\{p\}, \vec{y}), \quad \mu_{11}, \mu_{12}, \mu_{22}$$


$[\vec{x}, \vec{y}, \mu_{ij} = \text{free parameters}]$

Approaching cut solutions numerically

What if we work in $d = 4$? (*) (*) R_2 rational terms must to be provided at some later stage
 [Pozzorini, Lang, Zhang, Zoller '20, '22]

- In $d = 4$, cut solutions do *not* admit a unique parametric solution in general, and are structured in different *branches*:

$$D_1 = D_2 = \dots D_i = 0 \quad \rightarrow \quad \left[\begin{array}{l} \left\{ q_1^{(1)}(\{p\}, \vec{x}), q_2^{(1)}(\{p\}, \vec{y}) \right\} \\ \vdots \\ \left\{ q_1^{(r)}(\{p\}, \vec{x}), q_2^{(r)}(\{p\}, \vec{y}) \right\} \end{array} \right. \begin{array}{l} \longrightarrow 1^{\text{st}} \text{ branch} \\ \vdots \\ \longrightarrow r^{\text{th}} \text{ branch} \end{array}$$

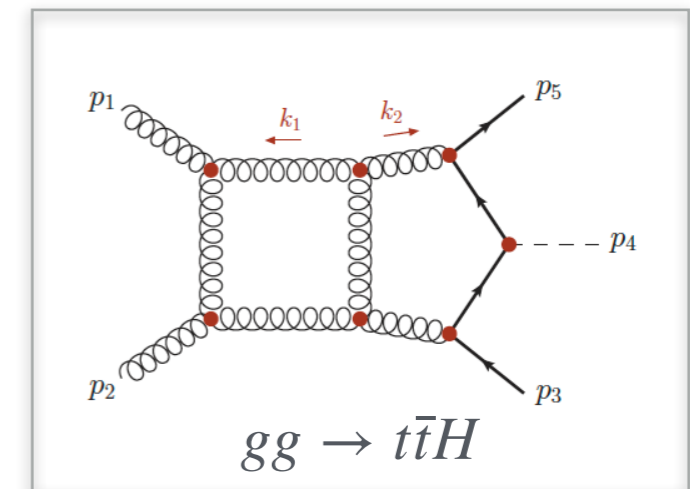
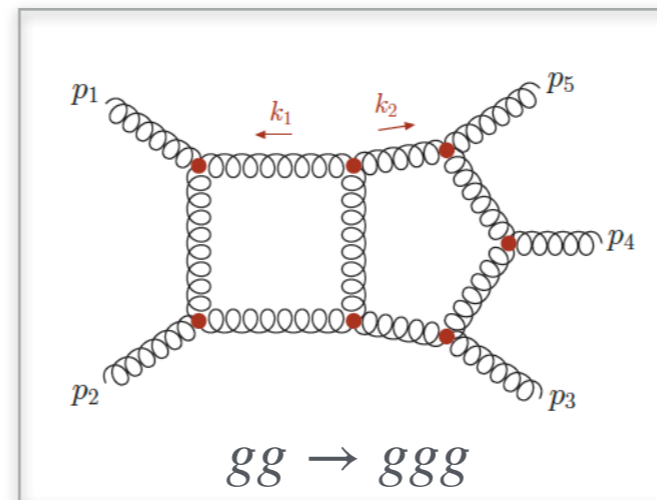
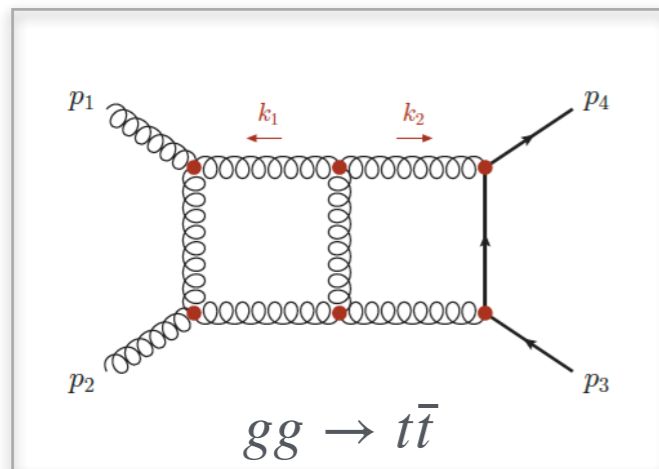
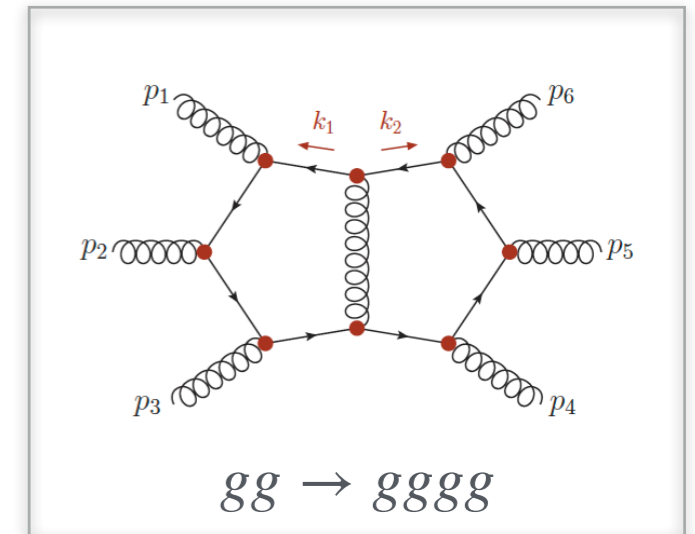
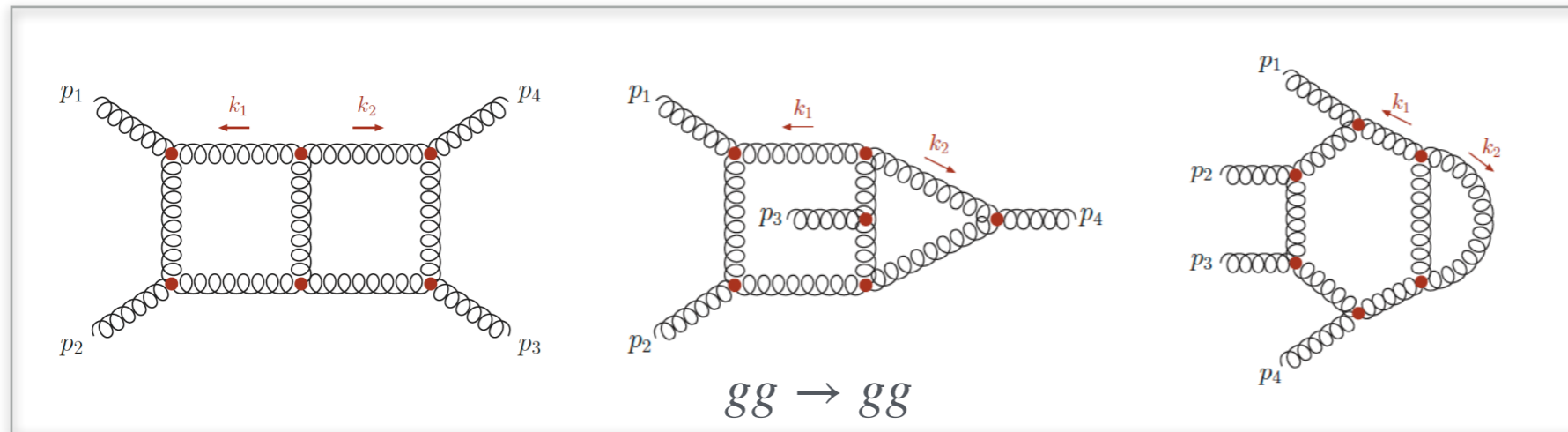
- Solutions from *all* branches need to be considered for fitting coefficients! 
- ↪ larger system of equations to solve (more rows)

$$\begin{pmatrix} N_1^{(1)} \\ \vdots \\ N_m^{(r)} \end{pmatrix} = \begin{pmatrix} M_{1,1} & \dots & M_{1,m} \\ & \ddots & \\ M_{m \cdot r, 1} & \dots & M_{m \cdot r, m} \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_m \end{pmatrix} \quad \leftrightarrow \quad \vec{N} = \mathbf{M} \cdot \vec{c}$$

- There are cases where the matrix \mathbf{M} is rank-deficient

Validation of the method

- We have validated our method considering up to 6-point loop functions, with *massless* and *massive* propagators

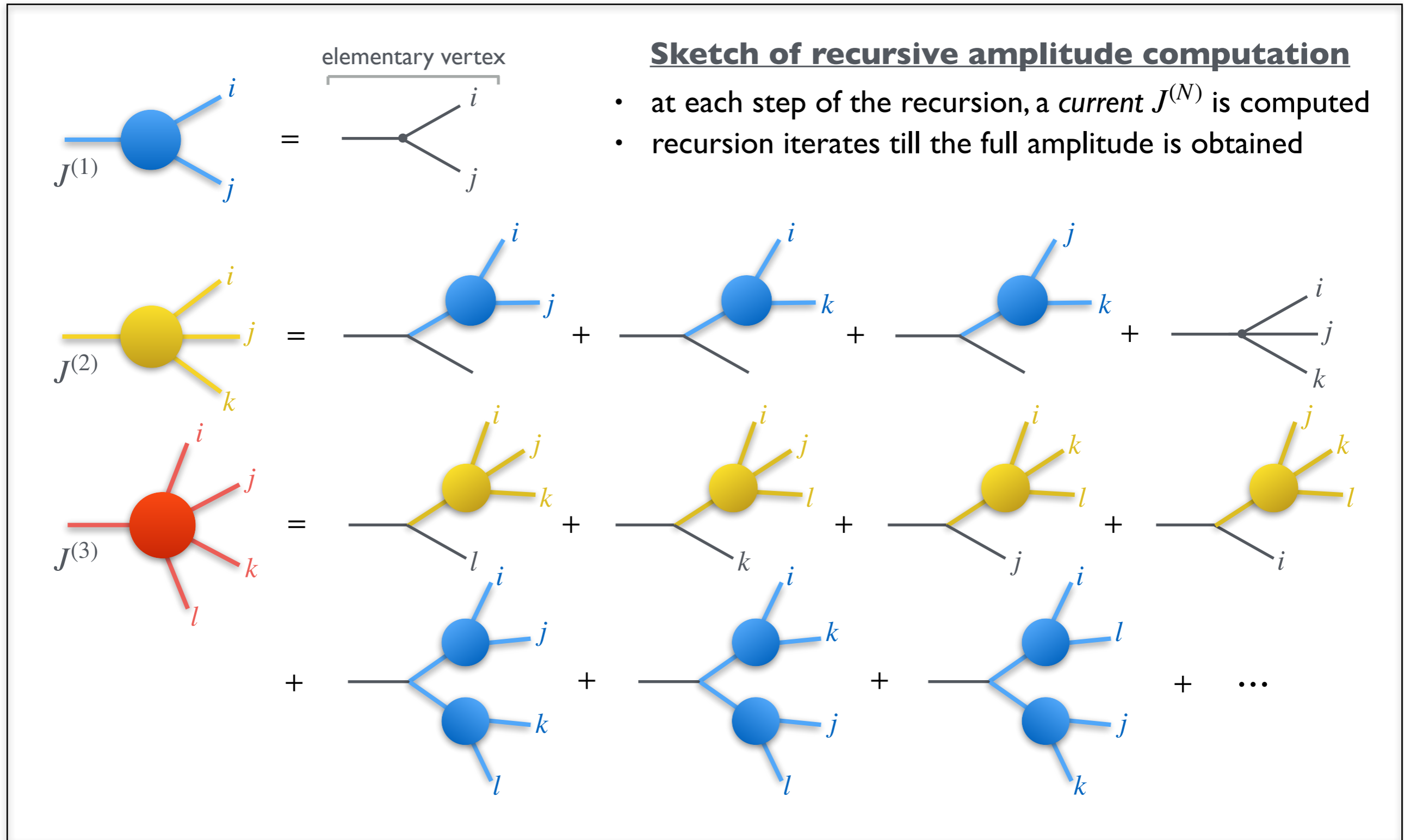


- $N = N$ checks performed in both $d = 4$ and $d = 4 - 2\epsilon$ dimensions

[G.B, Canko, Spourdalakis and Papadopoulos, [2506.07231 \[hep-ph\]](https://arxiv.org/abs/2506.07231)]

Outlook I: dimensionally regularised amplitude computation

- Goal: compute d -dim. $\bar{N}(q)$ within a *numerical, recursive* framework (built on $d = 4$)



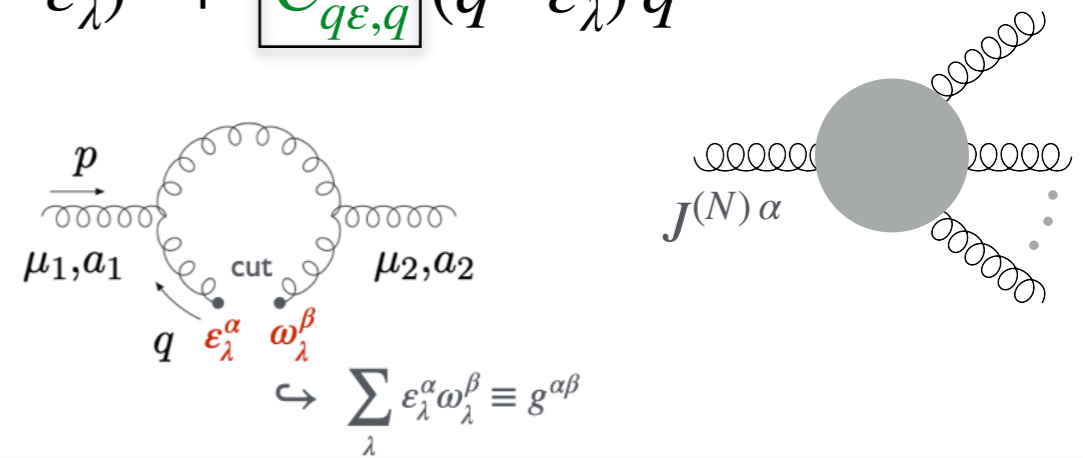
Outlook I: dimensionally regularised amplitude computation

- Introduce formal operator \mathcal{E} which generates all *evanescent contributions* to a given $d = 4$ current $J^{(N)}$

Example: pure Yang Mills QCD @ 1-loop

$$\mathcal{E}[J^{(N)\alpha}] = \boxed{C_q^{(N)}} \tilde{q}^\alpha + \boxed{C_\varepsilon^{(N)}} \tilde{\varepsilon}_\lambda^\alpha + \boxed{J_{q\varepsilon}^{(N)\alpha}} (\tilde{q} \cdot \tilde{\varepsilon}_\lambda) + \boxed{C_{q\varepsilon,q}^{(N)}} (\tilde{q} \cdot \tilde{\varepsilon}_\lambda) \tilde{q}^\alpha$$

$$+ \boxed{J_\mu^{(N)\alpha}} \mu + \boxed{J_{d-4}^{(N)\alpha}} (d-4)$$



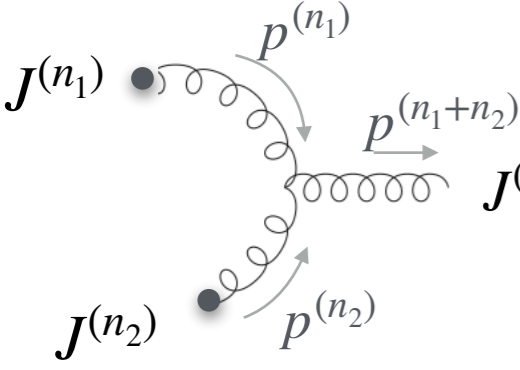
- All coefficients in the formula above satisfy *recursion relations* in complete analogy with the current $J^{(N)\alpha}$ itself
- The recursion relations can be extracted from the structure of the *vertices* of the model under consideration

G.B, Canko, Papadopoulos and Spourdalakis, in progress
[see also Fazio, Mastrolia, Mirabella and Torres Bobadilla '14]

Outlook I: dimensionally regularised amplitude computation

[G.B, Canko, Papadopoulos and Spourdalakis, in progress]

- Example:** recursion relations for 3-gluon vertex

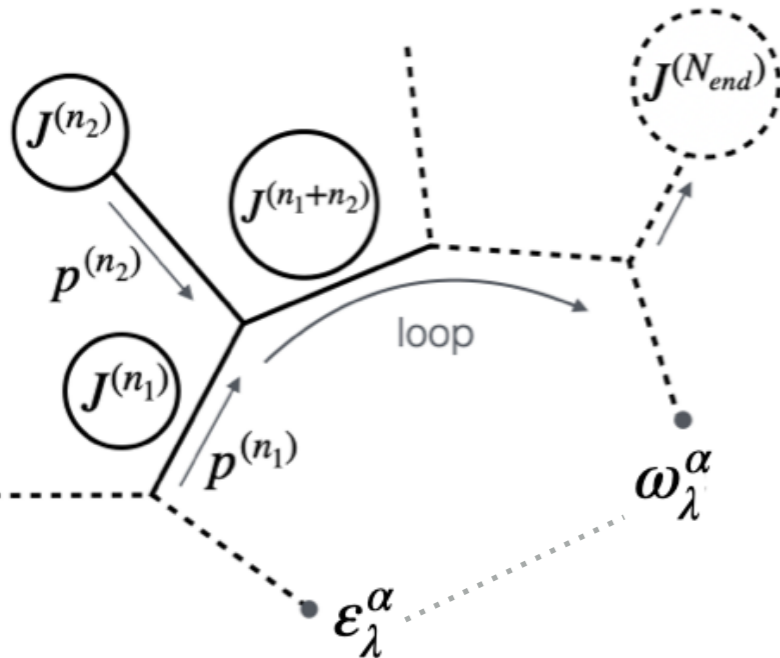


$$\begin{aligned}
 & \equiv V_{\text{ggg}}^\alpha(J^{(n_1)}, J^{(n_2)}) \\
 & \equiv (J^{(n_2)} \cdot (2p^{(n_1)} + p^{(n_2)})) J^{(n_1)\alpha} \\
 & \quad - (J^{(n_1)} \cdot (p^{(n_1)} + 2p^{(n_2)})) J^{(n_2)\alpha} \\
 & \quad + (J^{(n_1)} \cdot J^{(n_2)}) (p^{(n_2)} - p^{(n_1)})^\alpha
 \end{aligned}$$

Initial conditions:

$$\begin{aligned}
 C_q^{(1)} &= 0 & C_\varepsilon^{(1)} &= 1 \\
 C_{\varepsilon q}^{(1)} &= 0 & J_{\varepsilon q}^{(1)\alpha} &= 0
 \end{aligned}$$

[loop endpoint]



$$C_q^{(n_1+n_2)} = C_q^{(n_1)} [J^{(n_2)} \cdot (2p^{(n_1)} + p^{(n_2)})] - (1 + \delta_{(n_1+n_2)N_{\text{end}}}) (J^{(n_1)} \cdot J^{(n_2)})$$

$$C_\varepsilon^{(n_1+n_2)} = C_\varepsilon^{(n_1)} [J^{(n_2)} \cdot (2p^{(n_1)} + p^{(n_2)})]$$

$$J_{\varepsilon q}^{(n_1+n_2)\alpha} = V_{\text{ggg}}^\alpha(J_{\varepsilon q}^{(n_1)}, J^{(n_2)}) - (C_\varepsilon^{(n_1)} + \mu C_{\varepsilon q}^{(n_1)}) J^{(n_2)\alpha}$$

$$C_{\varepsilon q}^{(n_1+n_2)} = C_{\varepsilon q}^{(n_1)} [J^{(n_2)} \cdot (2p^{(n_1)} + p^{(n_2)})] - (1 + \delta_{(n_1+n_2)N_{\text{end}}}) (J_{q\varepsilon}^{(n_1)} \cdot J^{(n_2)})$$

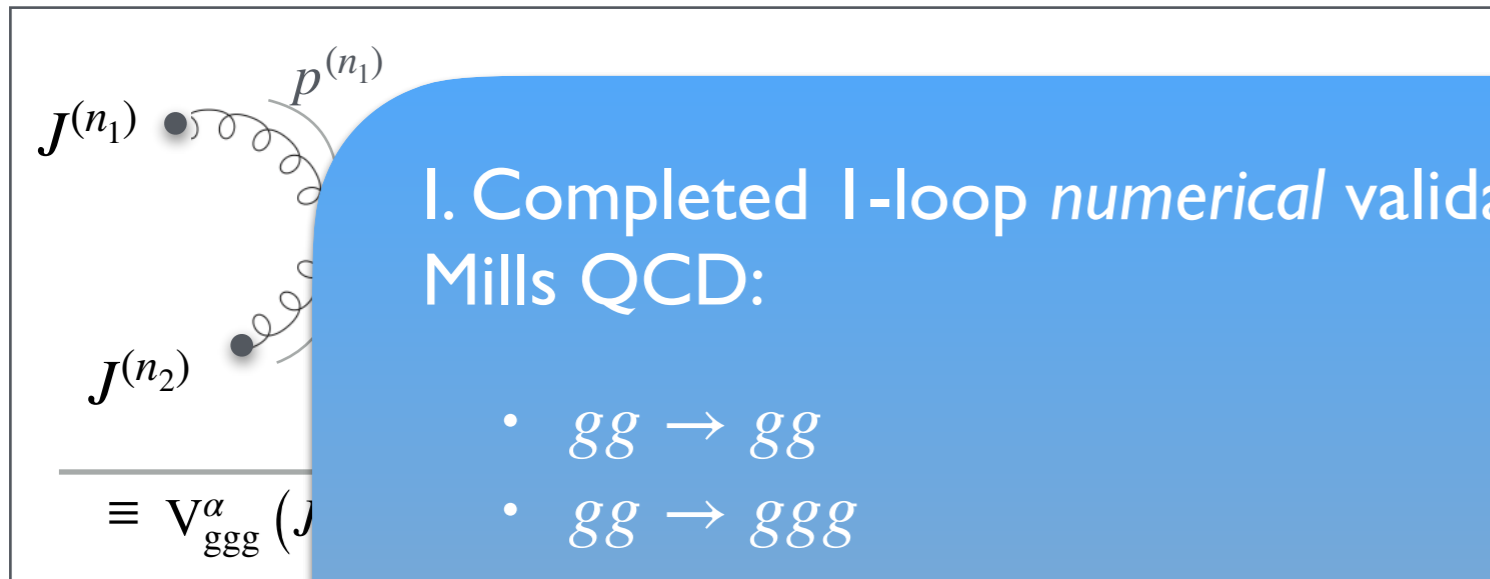
$$J_\mu^{(n_1+n_2)\alpha} = (-1 + 2\delta_{(n_1+n_2)N_{\text{end}}} C_q^{(n_1)} J^{(n_2)\alpha}) + \delta_{(n_1+n_2)N_{\text{end}}} (J_{\varepsilon q}^{(n_1)\alpha} + C_{\varepsilon q}^{(n_1)} (p^{(n_2)} - p^{(n_1)})^\alpha)$$

$$J_{d-4}^{(n_1+n_2)\alpha} = \delta_{(n_1+n_2)N_{\text{end}}} C_\varepsilon^{(n_1)} (p^{(n_2)} - p^{(n_1)})^\alpha$$

Outlook I: dimensionally regularised amplitude computation

[G.B, Canko, Papadopoulos and Spourdalakis, in progress]

- **Example:** recursion relations for 3-gluon vertex



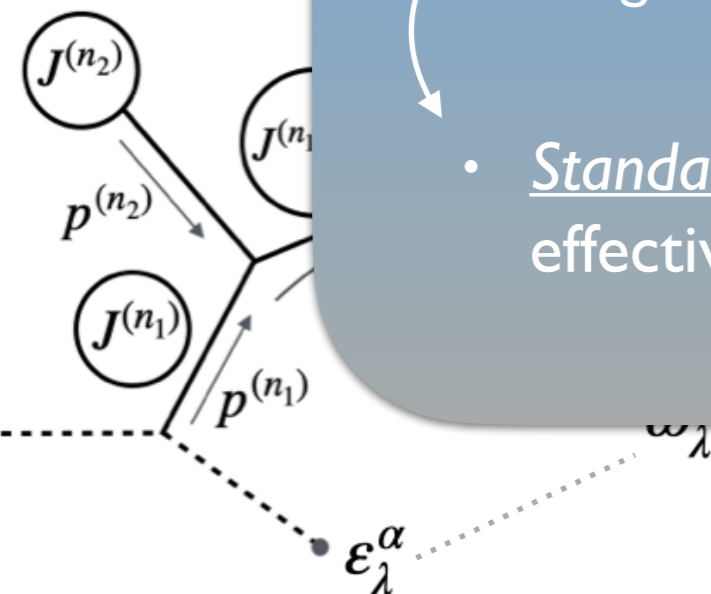
I. Completed 1-loop *numerical* validation for pure Yang-Mills QCD:

- $gg \rightarrow gg$
 - $gg \rightarrow ggg$
- vs
- Cut Constructible (CC) + Rational Terms computed at *integrand* level
 - Standard approach: CC+ R_1 at integrand level, R_2 via effective Feynman rules

conditions:

$$C_\varepsilon^{(1)} = 1$$

$$J_{\varepsilon q}^{(1)\alpha} = 0$$



$$J_\mu^{(n_1+n_2)\alpha} = \left(-1 + 2\delta_{(n_1+n_2)N_{\text{end}}} C_q^{(n_1)} J^{(n_2)\alpha} \right) + \delta_{(n_1+n_2)N_{\text{end}}} \left(J_{\varepsilon q}^{(n_1)\alpha} + C_{\varepsilon q, q}^{(n_1)} (p^{(n_2)} - p^{(n_1)})^\alpha \right)$$

$$J_{d-4}^{(n_1+n_2)\alpha} = \delta_{(n_1+n_2)N_{\text{end}}} C_\varepsilon^{(n_1)} (p^{(n_2)} - p^{(n_1)})^\alpha$$

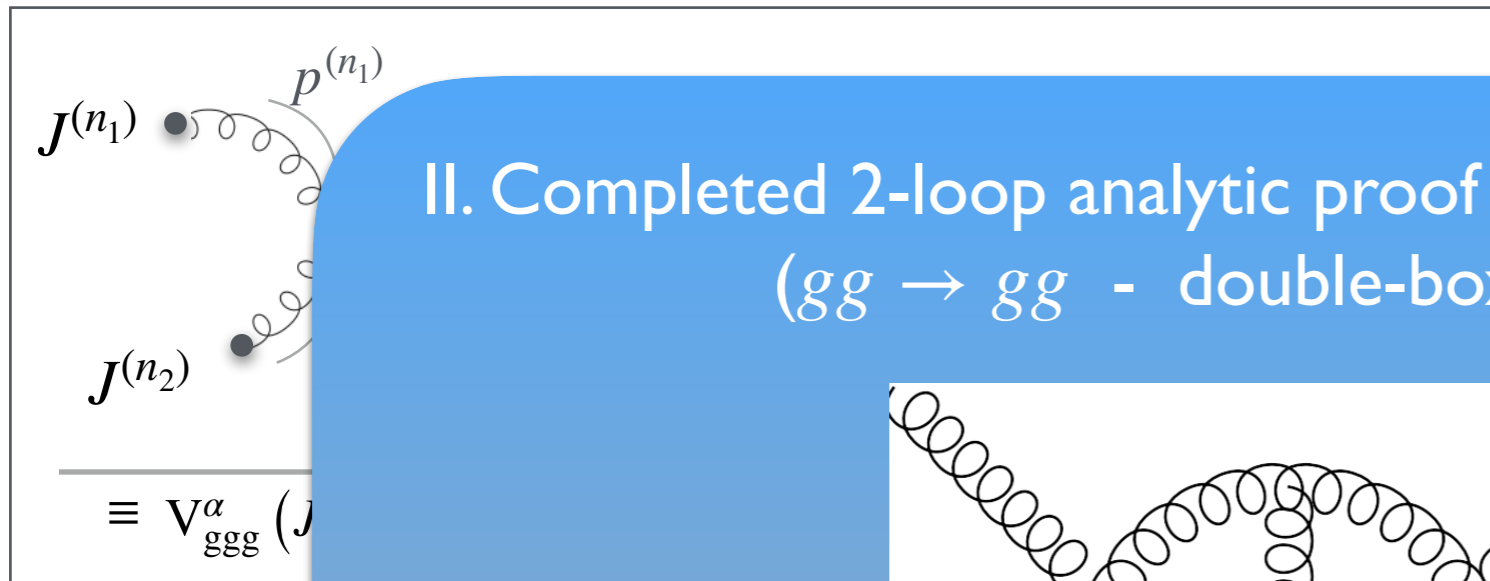
$J^{(n_2)}$

$J^{(n_2)}$

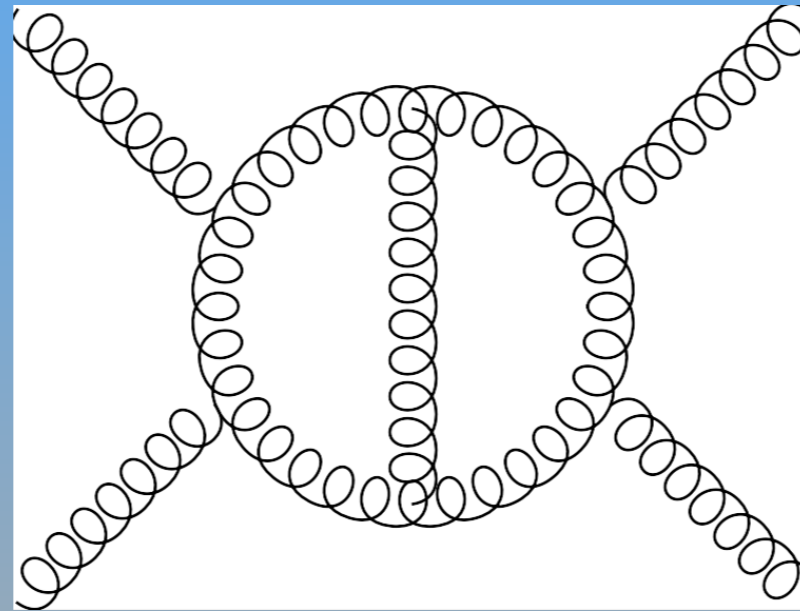
Outlook I: dimensionally regularised amplitude computation

[G.B, Canko, Papadopoulos and Spourdalakis, in progress]

- **Example:** recursion relations for 3-gluon vertex



II. Completed 2-loop analytic proof of concept ($gg \rightarrow gg$ - double-box topology)



conditions:

$$C_\varepsilon^{(1)} = 1$$

$$J_{\varepsilon q}^{(1)\alpha} = 0$$

$J^{(n_2)}$

$J^{(n_2)}$

- all μ_{ij} and ε terms reproduced successfully

$$J_\mu^{(n_1+n_2)\alpha} = \left(-1 + 2\delta_{(n_1+n_2)N_{\text{end}}} C_q^{(n_1)} J^{(n_2)\alpha}\right) + \delta_{(n_1+n_2)N_{\text{end}}} \left(J_{\varepsilon q}^{(n_1)\alpha} + C_{\varepsilon q, q}^{(n_1)} (p^{(n_2)} - p^{(n_1)})^\alpha\right)$$

$$J_{d-4}^{(n_1+n_2)\alpha} = \delta_{(n_1+n_2)N_{\text{end}}} C_\varepsilon^{(n_1)} (p^{(n_2)} - p^{(n_1)})^\alpha$$

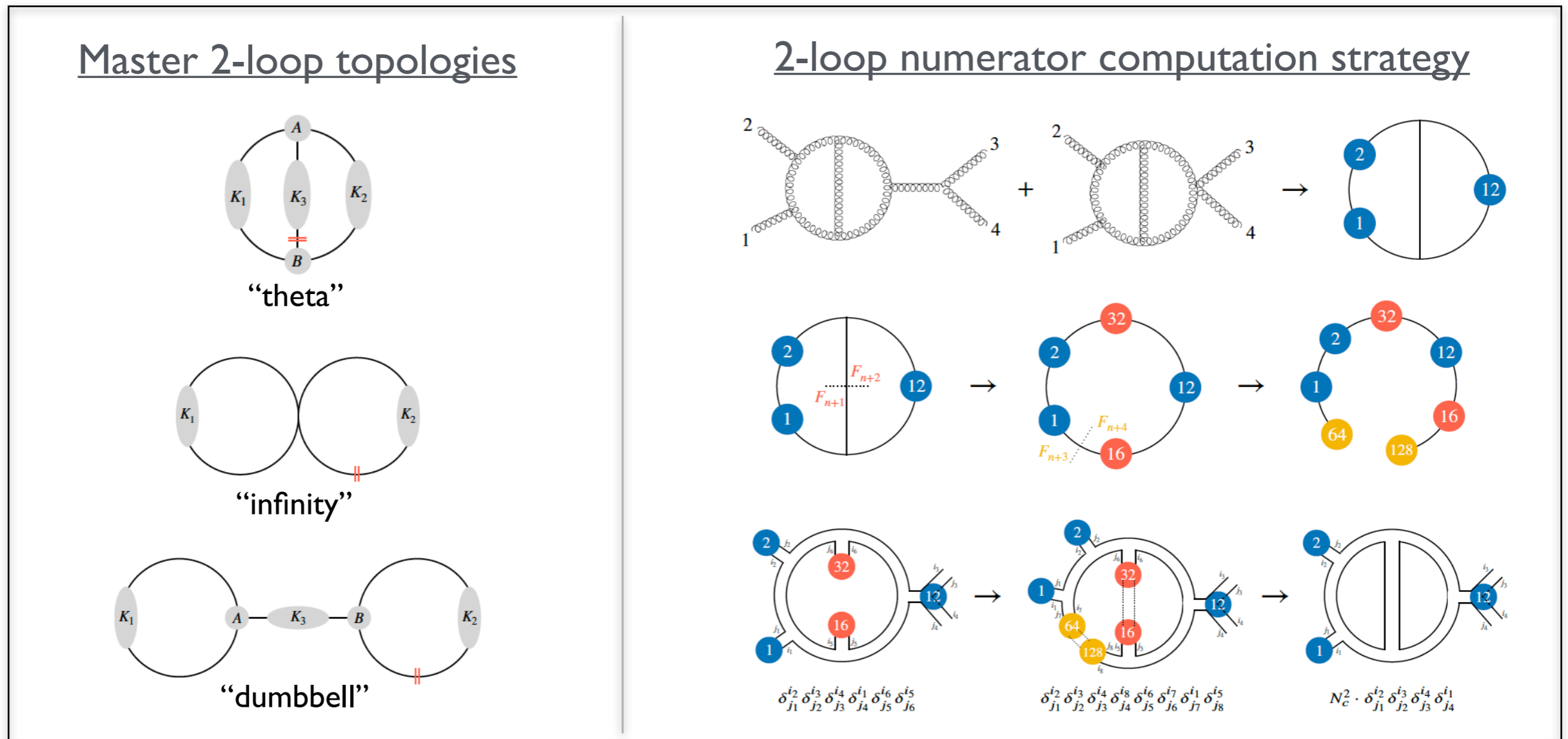
Outlook (II): numerator provider

Constructing HELAC-2LOOP

G.B, Canko, Papadopoulos and Spourdalakis, in progress

- General algorithm:

{G.B, Canko and Papadopoulos, [PoS RADCOR2023 \(2024\) 081](#)}



- Proof of concept ($d = 4 - 2\epsilon$) & code implementing the reduction under construction

Summary and conclusions

- We have established a *method* to fit coefficients of the reduction at **integrand level**
 - the method works in either $d = 4$ or $d = 4 - 2\varepsilon$ dimensions
 - addressing IBP reduction of Feynman integrals is the next big step
- Performing the reduction in $d = 4 - 2\varepsilon$ dimensions potentially pays off
 - no need to separately compute R_2 rational terms
 - simpler structure of cut solutions (\bar{q}_1, \bar{q}_2)
- How feasible are **numerical** computations of loop numerators in $d = 4 - 2\varepsilon$?

Requisites:

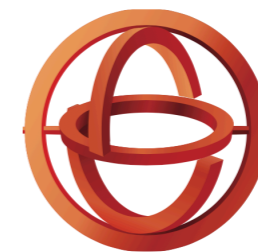
 - a procedure to supplement terms $\propto \mu_{ij}, \varepsilon$ to numerical computations of numerators;
 - a “numerator provider” working in $d = 4 - 2\varepsilon$ dimensions (possibly general-purpose);
 - a ready-to-use implementation of the reduction algorithm (such as e.g. CutTools).

↪ these steps are essential to assess the numerical performance of the method

Thank you for your attention!

감사합니다!

This research is supported by the Hellenic foundation for Research and Innovation (H.F.R.I.) under the “*2nd Call for H.F.R.I. Research Projects to support Faculty Members and Researchers*” (Project Number: 02674 - HOCTools-II)



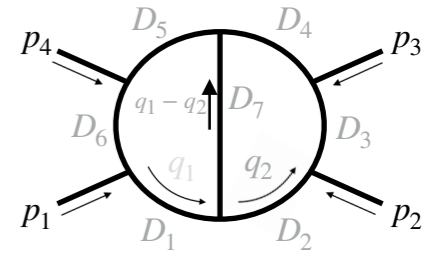
H.F.R.I.
Hellenic Foundation for
Research & Innovation

Backup slides

2-loop reduction: a toy example

$$N(\bar{q}_1, \bar{q}_2, \{p\}) \equiv (\bar{q}_1 \cdot \bar{q}_2)(p_2 \cdot p_3) + (\bar{q}_2 \cdot p_4)(p_1 \cdot p_2)$$

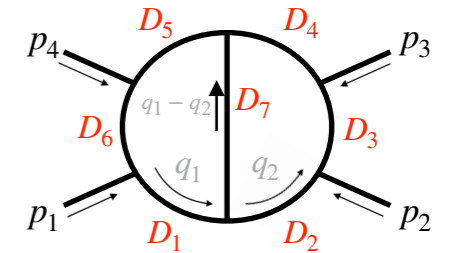
$$A_{2L} = \int d^d \bar{q}_1 d^d \bar{q}_2 \frac{N(\bar{q}_1, \bar{q}_2, \{p\})}{\prod_{i=1}^7 D_i(\bar{q}_1, \bar{q}_2, \{p\})}$$



2-loop reduction: a toy example

$$\begin{aligned}
 N(\bar{q}_1, \bar{q}_2, \{p\}) &\equiv (\bar{q}_1 \cdot \bar{q}_2)(p_2 \cdot p_3) + (\bar{q}_2 \cdot p_4)(p_1 \cdot p_2) \\
 &= -\frac{1}{2}(D_7 - D_1 - D_2)(p_2 \cdot p_3) + (\bar{q}_2 \cdot p_4)(p_1 \cdot p_2)
 \end{aligned}$$

$$A_{2L} = \int d^d \bar{q}_1 d^d \bar{q}_2 \frac{N(\bar{q}_1, \bar{q}_2, \{p\})}{\prod_{i=1}^7 D_i(\bar{q}_1, \bar{q}_2, \{p\})}$$



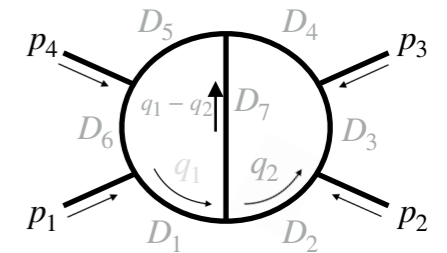
$$\begin{aligned}
 D_1 &= \bar{q}_1^2 \\
 D_2 &= \bar{q}_2^2 \\
 D_3 &= (\bar{q}_2 + p_2)^2 \\
 D_4 &= (\bar{q}_2 + p_2 + p_3)^2 \\
 D_5 &= (\bar{q}_1 + p_2 + p_3)^2 \\
 D_6 &= (\bar{q}_1 + p_2 + p_3 + p_4)^2 \\
 D_7 &= (\bar{q}_1 - \bar{q}_2)^2
 \end{aligned}$$

- Part of the above scalars can be *reduced* in terms of the D_i 's appearing in the Denominator:
Reducible Scalar Products (RSP)

2-loop reduction: a toy example

$$\begin{aligned}
 N(\bar{q}_1, \bar{q}_2, \{p\}) &\equiv (\bar{q}_1 \cdot \bar{q}_2)(p_2 \cdot p_3) + (\bar{q}_2 \cdot p_4)(p_1 \cdot p_2) \\
 &= -\frac{1}{2}(D_7 - D_1 - D_2)(p_2 \cdot p_3) + (\bar{q}_2 \cdot p_4)(p_1 \cdot p_2)
 \end{aligned}$$

$$A_{2L} = \int d^d \bar{q}_1 d^d \bar{q}_2 \frac{N(\bar{q}_1, \bar{q}_2, \{p\})}{\prod_{i=1}^7 D_i(\bar{q}_1, \bar{q}_2, \{p\})}$$



$$\begin{aligned}
 D_1 &= \bar{q}_1^2 \\
 D_2 &= \bar{q}_2^2 \\
 D_3 &= (\bar{q}_2 + p_2)^2 \\
 D_4 &= (\bar{q}_2 + p_2 + p_3)^2 \\
 D_5 &= (\bar{q}_1 + p_2 + p_3)^2 \\
 D_6 &= (\bar{q}_1 + p_2 + p_3 + p_4)^2 \\
 D_7 &= (\bar{q}_1 - \bar{q}_2)^2
 \end{aligned}$$

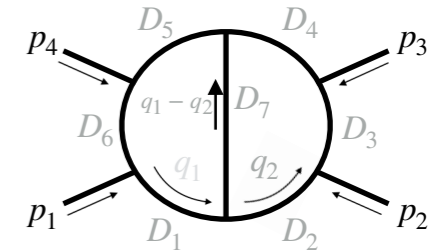
- Part of the above scalars can be *reduced* in terms of the D_i 's appearing in the Denominator:
Reducible Scalar Products (RSP)
- The subset of scalars that cannot be decomposed this way are *Irreducible Scalar Products (ISP)*

2-loop reduction: a toy example

$$N(\bar{q}_1, \bar{q}_2, \{p\}) \equiv (\bar{q}_1 \cdot \bar{q}_2)(p_2 \cdot p_3) + (\bar{q}_2 \cdot p_4)(p_1 \cdot p_2)$$

$$= -\frac{1}{2}(D_7 - D_1 - D_2)(p_2 \cdot p_3) + (\bar{q}_2 \cdot p_4)(p_1 \cdot p_2)$$

$$A_{2L} = \int d^d \bar{q}_1 d^d \bar{q}_2 \frac{N(\bar{q}_1, \bar{q}_2, \{p\})}{\prod_{i=1}^7 D_i(\bar{q}_1, \bar{q}_2, \{p\})}$$



$$D_1 = \bar{q}_1^2$$

$$D_2 = \bar{q}_2^2$$

$$D_3 = (\bar{q}_2 + p_2)^2$$

$$D_4 = (\bar{q}_2 + p_2 + p_3)^2$$

$$D_5 = (\bar{q}_1 + p_2 + p_3)^2$$

$$D_6 = (\bar{q}_1 + p_2 + p_3 + p_4)^2$$

$$D_7 = (\bar{q}_1 - \bar{q}_2)^2$$

$$A_{2L} = \int d\bar{q}_1 d\bar{q}_2 \left\{ \frac{-(p_2 \cdot p_3)/2}{D_1 D_2 D_3 D_4 D_5 D_6} + \frac{(p_2 \cdot p_3)/2}{D_2 D_3 D_4 D_5 D_6 D_7} \right. \\ \left. + \frac{(p_2 \cdot p_3)/2}{D_1 D_3 D_4 D_5 D_6 D_7} + \frac{(\bar{q}_2 \cdot p_4)(p_1 \cdot p_2)}{D_1 D_2 D_3 D_4 D_5 D_6 D_7} \right\}$$

- Part of the above scalars can be *reduced* in terms of the D_i 's appearing in the Denominator: *Reducible Scalar Products (RSP)*
- The subset of scalars that cannot be decomposed this way are *Irreducible Scalar Products (ISP)*
- **After reduction, the 2-loop amplitude is a combination of simpler Feynman integrals**