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

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

Towards numerical two-loop integrand reduction

Giuseppe Bevilacqua
NCSR "Demokritos"

41st Conference on Recent Developments in High Energy Physics and Cosmology (HEP2025)

Athens, 3rd July 2025

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

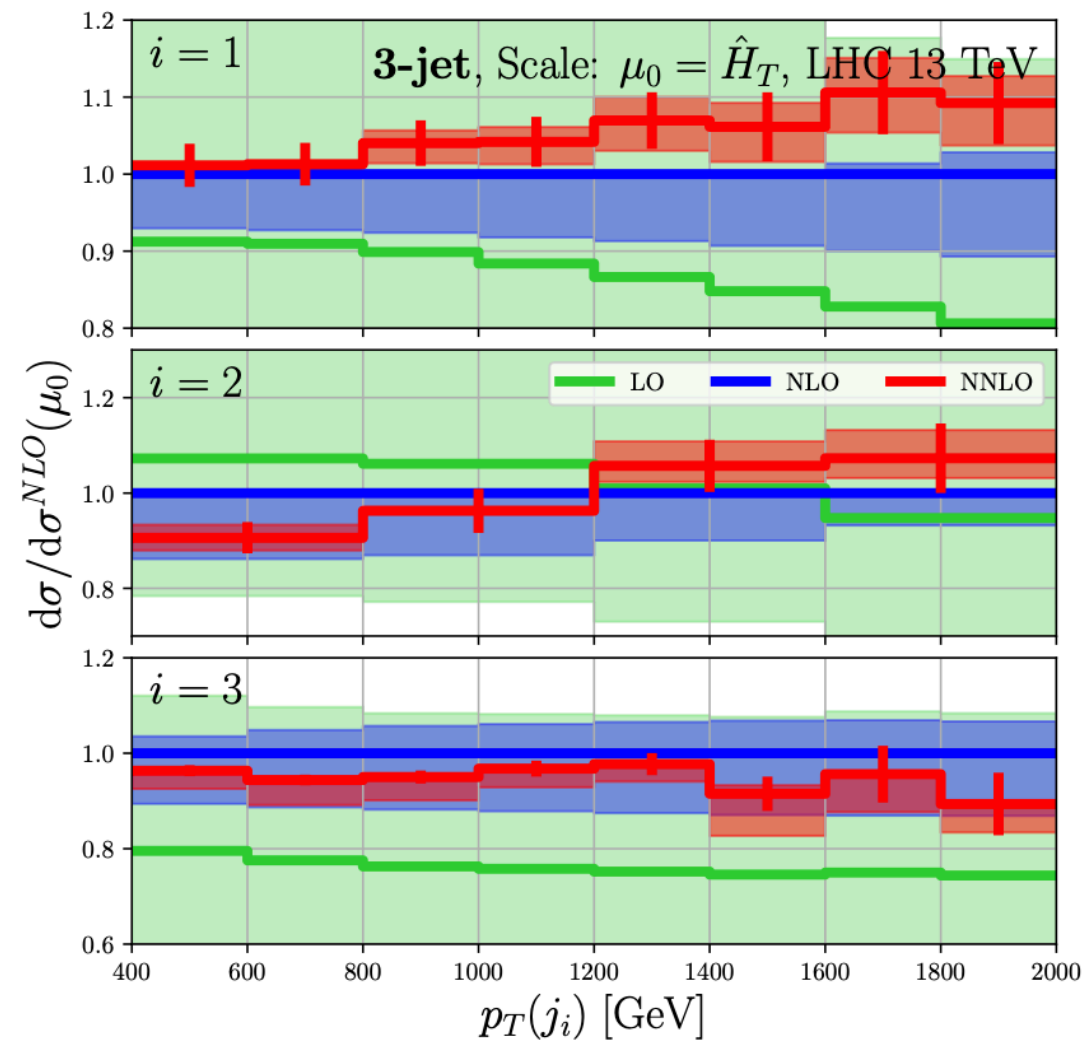
Based on arXiv:2506.07231 [hep-ph] + ongoing work

Introduction

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

$$
\begin{aligned}
\sigma_{NNLO} = & \int_m d\Phi_m \left(2Re(M_m^{(0)*} M_m^{(2)}) + \left|M_m^{(1)}\right|^2\right) J_m(\Phi) && \text{[VV]} \\
& + \int_{m+1} d\Phi_{m+1} \left(2Re\left(M_{m+1}^{(0)*} M_{m+1}^{(1)}\right)\right) J_{m+1}(\Phi) && \text{[RV]} \\
& + \int_{m+2} d\Phi_{m+2} \left|M_{m+2}^{(0)}\right|^2 J_{m+2}(\Phi) && \text{[RR]}
\end{aligned}
$$

{Czakon, Mitov and Poncelet '21}



- The current NNLO frontier is $2 \to 3$ processes

[see also Les Houches 2023 report]

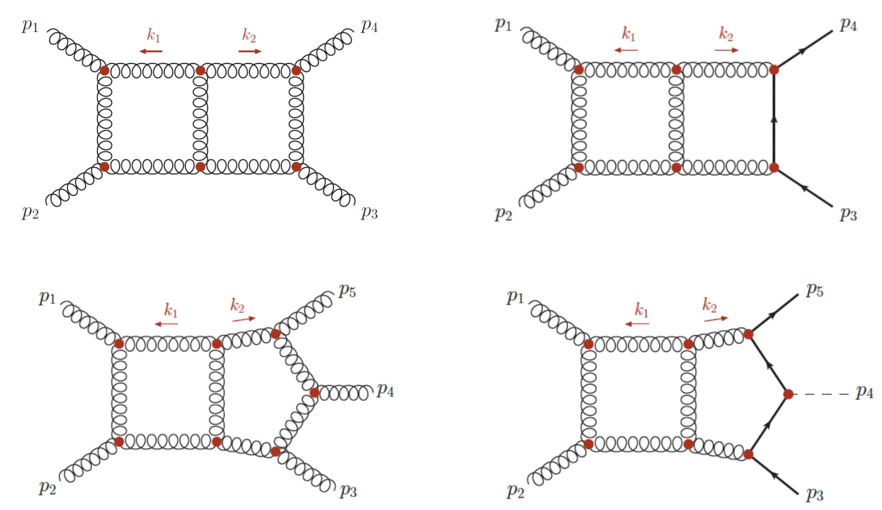
- $pp \to 3$ jets [Czakon, Mitov and Poncelet '21]
- $pp \to W^{\pm} b\bar{b}$ [Hartanto, Poncelet, Popescu and Zoia '22]
- $pp \to t\bar{t}H$ [Catani *et al.* '22; Devoto *et al.* '24]
- $pp \to t\bar{t}W$ [Buonocore *et al.* '23]
- $pp \to \gamma jj$ [Badger *et al.* '23]
- …

[selected examples]

Introduction

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

$$
\begin{aligned}
\sigma_{NNLO} = & \int_m d\Phi_m \left(2Re(M_m^{(0)*} M_m^{(2)}) + \left|M_m^{(1)}\right|^2\right) J_m(\Phi) && \text{[VV]} \\
& + \int_{m+1} d\Phi_{m+1} \left(2Re\left(M_{m+1}^{(0)*} M_{m+1}^{(1)}\right)\right) J_{m+1}(\Phi) && \text{[RV]} \\
& + \int_{m+2} d\Phi_{m+2} \left|M_{m+2}^{(0)}\right|^2 J_{m+2}(\Phi) && \text{[RR]}
\end{aligned}
$$



- 2-loop amplitudes are crucial ingredients (and often bottlenecks) for NNLO calculations
  - *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, …]

Introduction

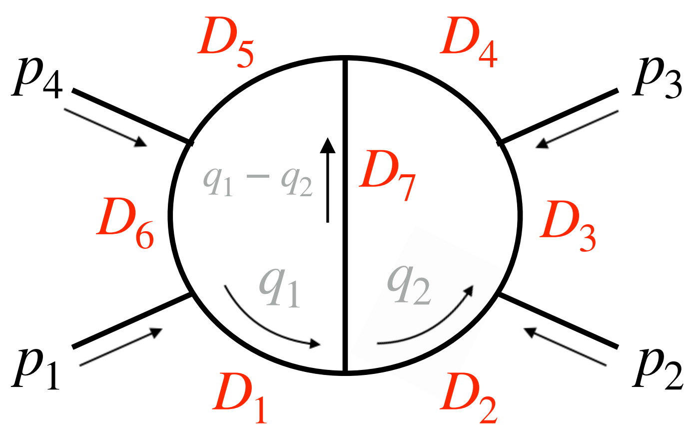
- So far, no complete framework for *automated numerical* 2-loop computations (as compared to the 1-loop case) is available*

  (*) 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*

- Example: 4-point function



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

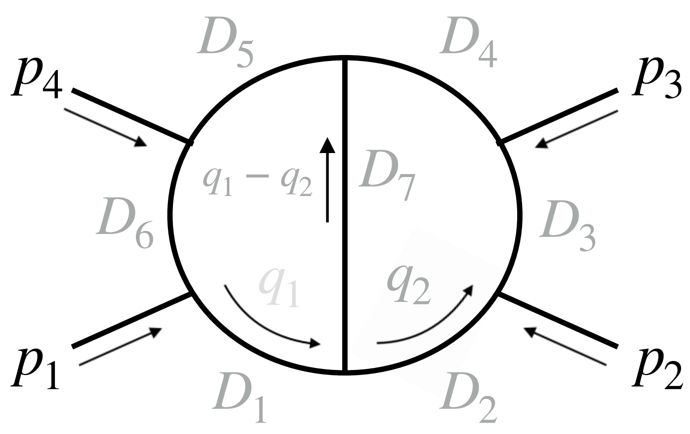
Denominator

“Inverse Propagators”

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

Anatomy of 2-loop amplitudes

- Example: 4-point function



Numerator

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

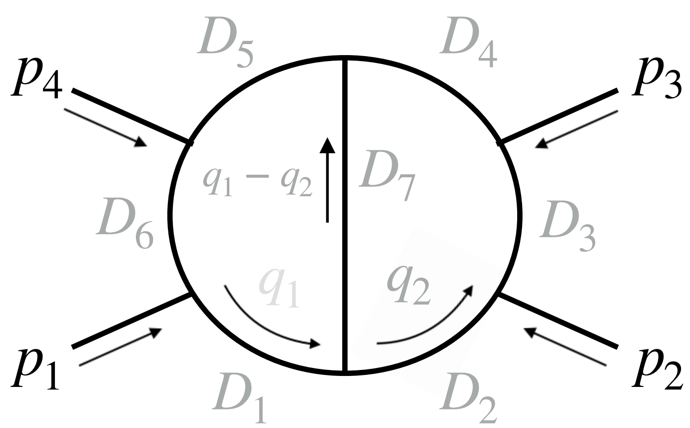
$$\begin{aligned}
D_1 &= \bar{q}_1^{\,2} \\
D_2 &= \bar{q}_2^{\,2} \\
D_3 &= (\bar{q}_2 + p_2)^2 \\
D_4 &= (\bar{q}_2 + p_2 + p_3)^2 \\
D_5 &= (\bar{q}_1 + p_2 + p_3)^2 \\
D_6 &= (\bar{q}_1 + p_2 + p_3 + p_4)^2 \\
D_7 &= (\bar{q}_1 - \bar{q}_2)^2
\end{aligned}$$

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

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

$\longrightarrow (\eta \cdot p_j) \equiv 0$
[transverse vector]

- Example: 4-point function



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

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

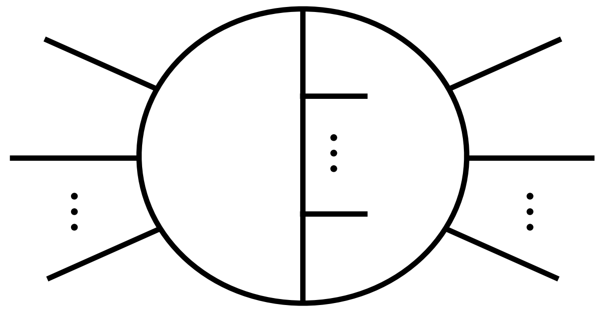
The Numerator is a function of few constituents (Lorentz scalars):

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

- 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)

Aims and claims

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



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

- 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 $\mathscr{A}_{2L}$

- 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:

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

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

Parametrising the Numerator

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

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

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

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

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

Parametrising the Numerator

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

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

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

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

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

- 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:

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

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

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

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

- 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 ? → using **`BasisDet`** [Zhang '12]
    - Maximal power ?

Fitting coefficients

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

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

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

Fitting coefficients

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

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

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

- Method n.1: "global fit" ⟷ Method n.2: "fit by cut"

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

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

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

Method n.1: "global fit"

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

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

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

↪ `LAPACK, Eigen`

Fitting coefficients

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

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 (→ "*maximal-cuts*"). Extract coefficients of $P^{(m)}$;

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

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

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

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

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

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

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

Method n.2: "fit by cut"

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

⋮

- Proceed iteratively till all polynomials are reconstructed

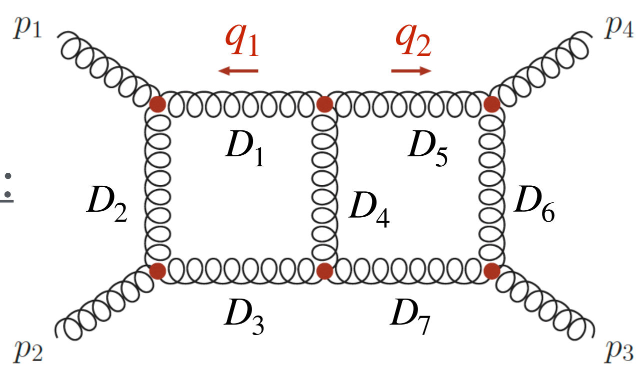
↪ Consistency check: *original* Numerator = *ansatz* Numerator ("$N = N$ test")

Constraints from cut equations

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

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

e.g.:



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

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

Constrains **7** of **11** ISPs

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

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

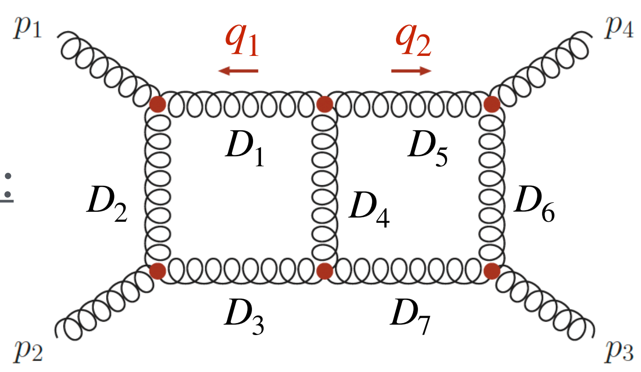
↪ Polynomial $P^{(m)}$ is simpler than $P_i^{(m-1)}$, which in turn is simpler than $P_{ij}^{(m-2)}$ etc …

Projecting over the full family of propagators

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

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

e.g.:



"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: $D_1 = D_2 = D_3 = D_4 = D_5 = D_6 = D_7 = D_8 = D_9 = 0$ $\qquad s \equiv (p_1 + p_2)^2$

Constrains **9** of **11** ISPs

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

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

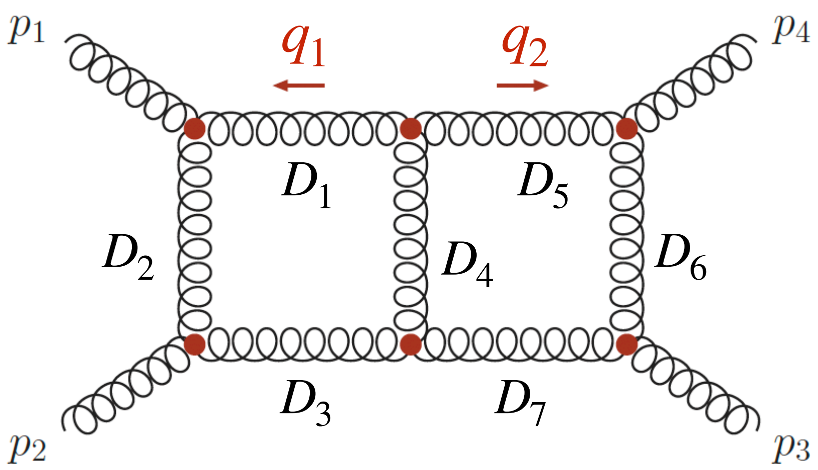
Note: different diagram topologies share the same family

Comparing different cutting approaches

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

"standard cut" approach

Level	Number of cuts	Number of coefficients
7	1	70
6	7	695
5	21	1430
4	35	1017
3	35	225
2	21	9



projecting over family

Level	Number of cuts	Number of coefficients
9	1	13
8	9	227
7	36	963
6	84	1445
5	126	780
4	126	116

- Tradeoff between number of cut equations and coefficients for each polynomial

[G.B, Canko, Spourdalakis and Papadopoulos, 2506.07231 [hep-ph]]

Approaching cut solutions numerically

- The sketched method is independent of dimensionality (works as well in $d = 4$ and in $d = 4 - 2\varepsilon$ dimensions)

- 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 ⟶ 't Hooft-Veltman scheme

- $p_1, \ldots, p_n$ (ext. momenta) in $d = 4$
- $\bar{q}_1, \bar{q}_2$ (loop momenta) in $d = 4 - 2\varepsilon$

↪ $\bar{q}_1, \bar{q}_2$ have $8 + 3 = \boxed{11}$ d.o.f.

$$\begin{cases} \bar{q}_1^2 = q_1^2 + \mu_{11} \\ \bar{q}_2^2 = q_2^2 + \mu_{22} \\ (\bar{q}_1 \cdot \bar{q}_2) = (q_1 \cdot q_2) + \mu_{12} \end{cases}$$

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

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

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

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*:

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

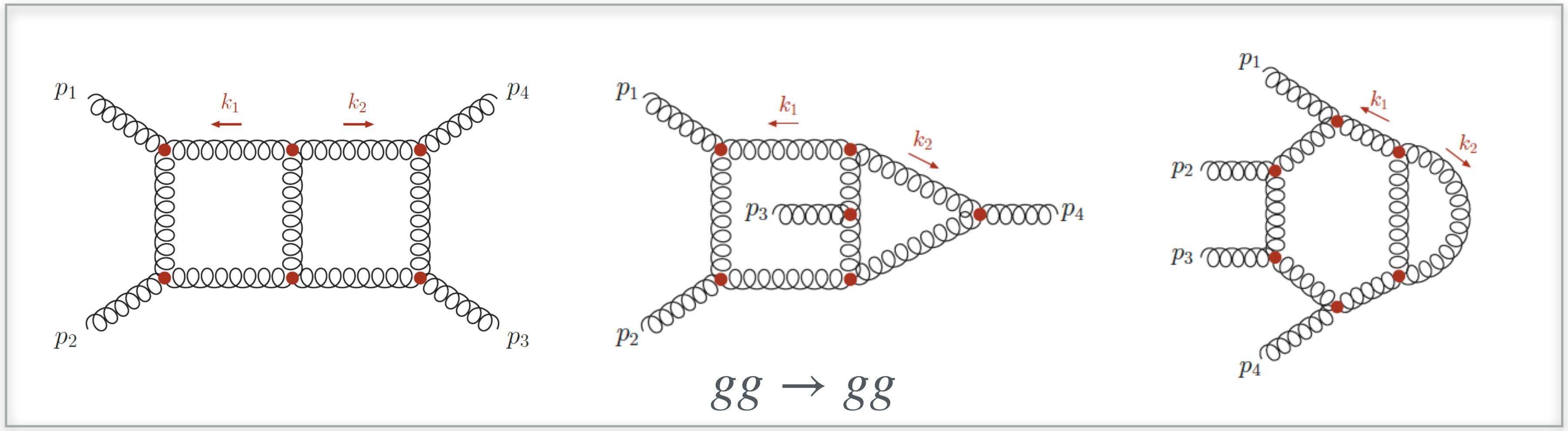
- Solutions from *all* branches need to be considered for fitting coefficients!
  ↪ larger system of equations to solve (more rows)

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

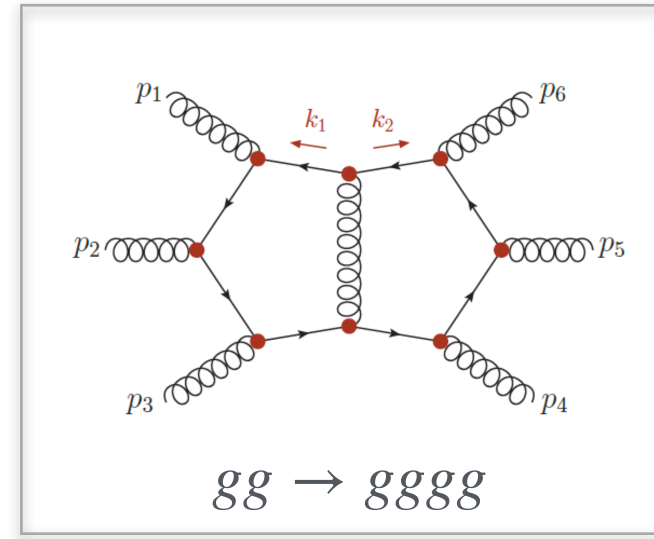
- There are cases where the matrix $\mathbf{M}$ is rank-deficient

Validation of the method

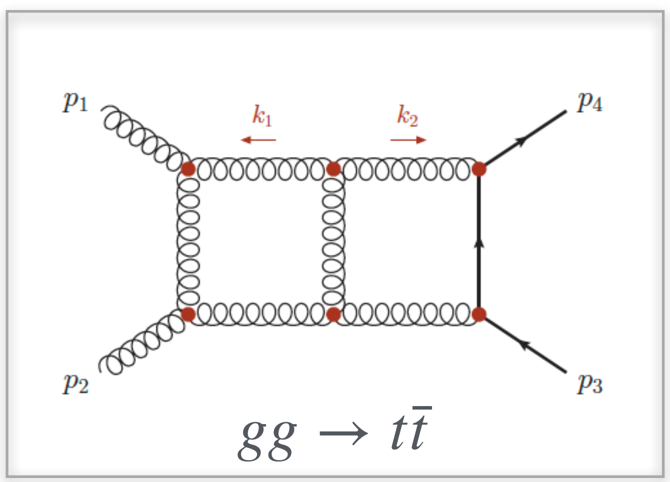
- We have validated our method considering up to 6-point loop functions, with *massless* and *massive* propagators



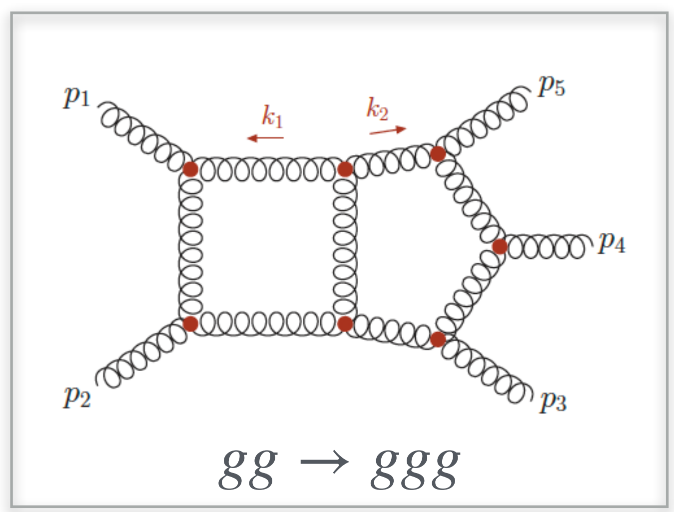
$gg \to gg$



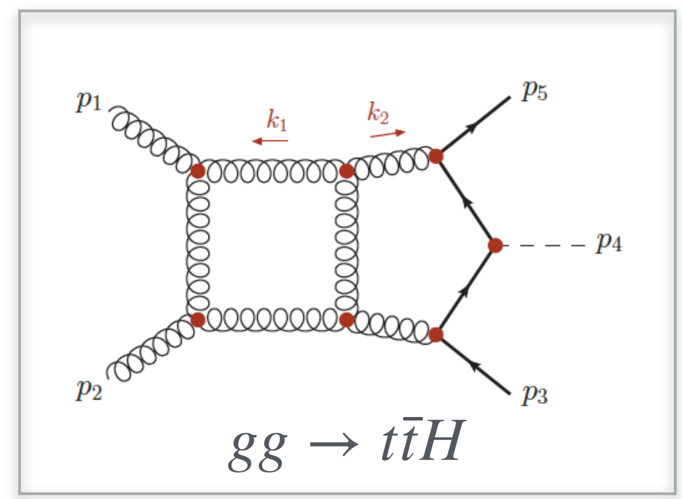
$gg \to gggg$



$gg \to t\bar{t}$



$gg \to ggg$



$gg \to t\bar{t}H$

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

[G.B, Canko, Spourdalakis and Papadopoulos, 2506.07231 [hep-ph]]

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

Outlook (I)

Towards numerical computations of dimensionally regularised Numerators

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

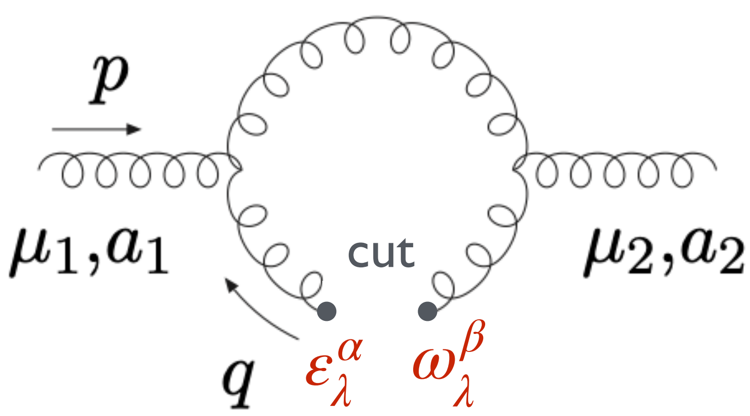
Simple example: QCD @ 1-loop (only gluons)

- μ and ε terms arise upon contracting few Lorentz structures: $\tilde{q}^{\alpha}$, $\tilde{\varepsilon}_{\lambda}{}^{\alpha}$, $\tilde{\omega}_{\lambda}{}^{\alpha}$, $(\tilde{q}\cdot\tilde{\varepsilon}_{\lambda})\,\tilde{q}^{\alpha}$

$$q^2 X \to \mu X$$

$$\sum_{\lambda} (\varepsilon_{\lambda}\cdot\omega_{\lambda})\, X \equiv g^{\mu\nu} g_{\mu\nu}\, X \to (d-4)\, X$$

$$\sum_{\lambda} (\varepsilon_{\lambda}\cdot q)(\omega_{\lambda}\cdot q)\, X \equiv q^2 X \to \mu X$$



$$\hookrightarrow \sum_{\lambda} \varepsilon_{\lambda}^{\alpha}\omega_{\lambda}^{\beta} \equiv g^{\alpha\beta}$$

- Track the *coefficients* of such structures appearing in the current J at each step of recursion:

$$J^{(N)\,\alpha} \to J^{(n)\,\alpha} + \boxed{C_q^{(N)}}\,\tilde{q}^{\alpha} + \boxed{C_{\varepsilon}^{(N)}}\,\tilde{\varepsilon}_{\lambda}{}^{\alpha} + \boxed{J_{q\varepsilon}^{(N)\,\alpha}}(\tilde{q}\cdot\tilde{\varepsilon}_{\lambda}) + \boxed{C_{q\varepsilon q}^{(N)}}(\tilde{q}\cdot\tilde{\varepsilon}_{\lambda})\,\tilde{q}^{\alpha}$$

- Derive recursion relations for the coefficients;
- At each step of the recursion, compute and add μ and ε terms to the current.

Towards numerical computations of dimensionally regularised Numerators

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

[see also Fazio, Mastrolia, Mirabella and Torres Bobadilla '14]

Simple exam

- μ and ε t ... $_\lambda)\,\tilde{q}^\alpha$

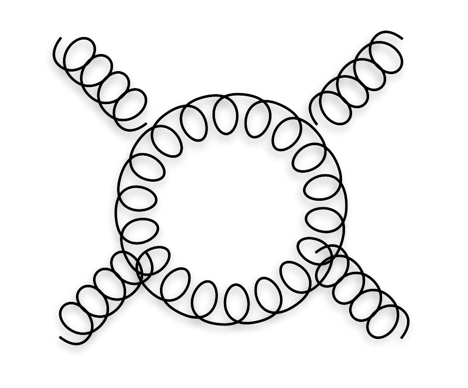
a_2

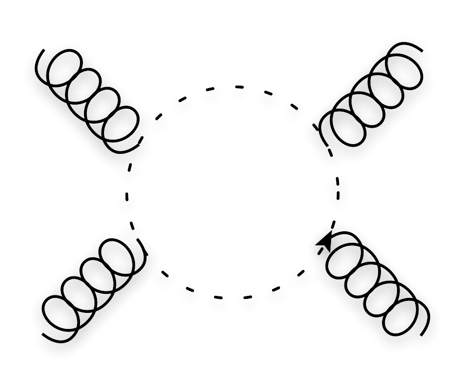
$\omega_\lambda^\beta \equiv g^{\alpha\beta}$

- Track th ... recursion:

$J^{(N}$ ... $_\lambda)\,\tilde{q}^\alpha$

I. Completed 1-loop numerical validation $(gg \to gg)$





- Cut Constructible (CC) + Rational Terms computed at *integrand* level

vs

- *Standard approach*: CC+R_1 at integrand level, R_2 via effective Feynman rules

- Derive recursion relations for the coefficients;
- At each step of the recursion, compute and add μ and ε terms to the current.

Towards numerical computations of dimensionally regularised Numerators

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

Simple examp

- μ and ε t ... $_\lambda)\,\tilde{q}^\alpha$

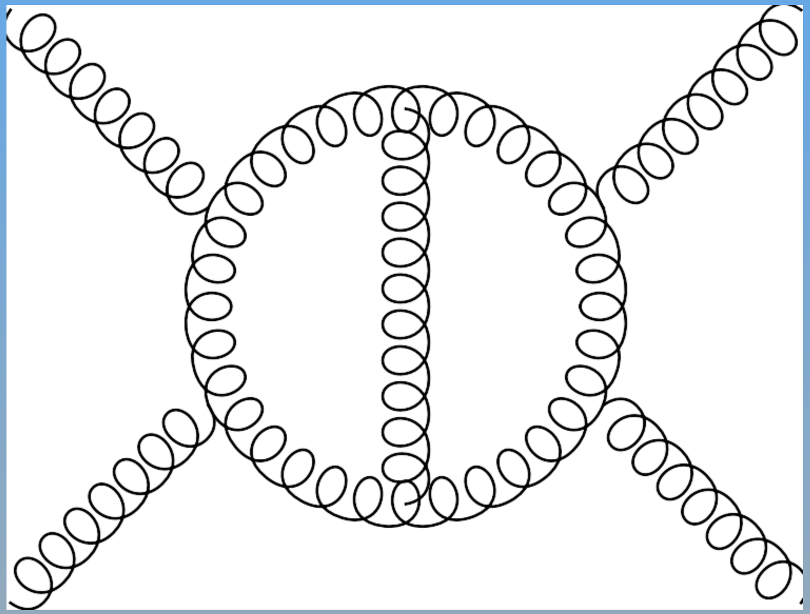
a_2

$\omega_\lambda^\beta \equiv g^{\alpha\beta}$

- Track th ... recursion:

$J^{(N}$ ... $_\lambda)\,\tilde{q}^\alpha$

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



- all μ_{ij} and ε terms reproduced successfully

- Derive recursion relations for the coefficients;
- At each step of the recursion, compute and add μ and ε terms to the current.

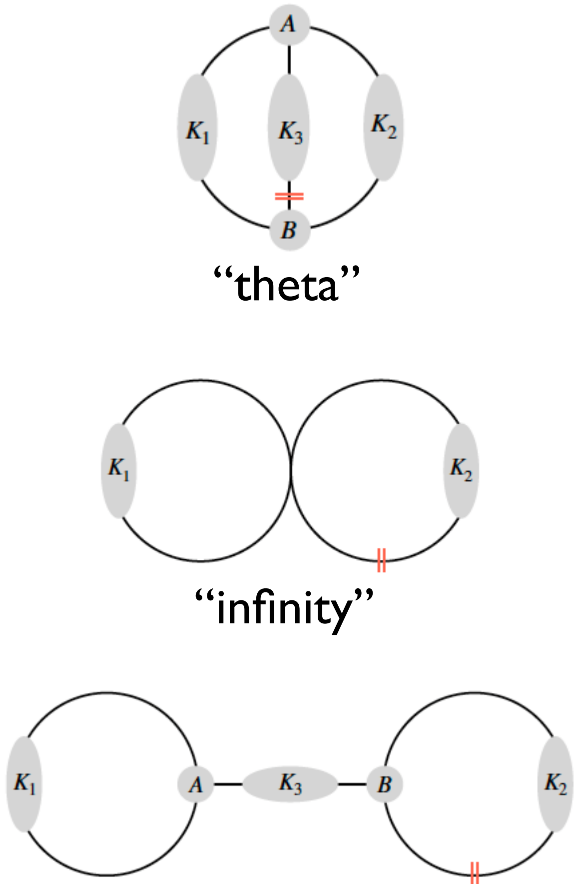
Constructing HELAC-2LOOP

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

- General algorithm

{G.B, Canko and Papadopoulos, PoS RADCOR2023 (2024) 081]

Master 2-loop topologies

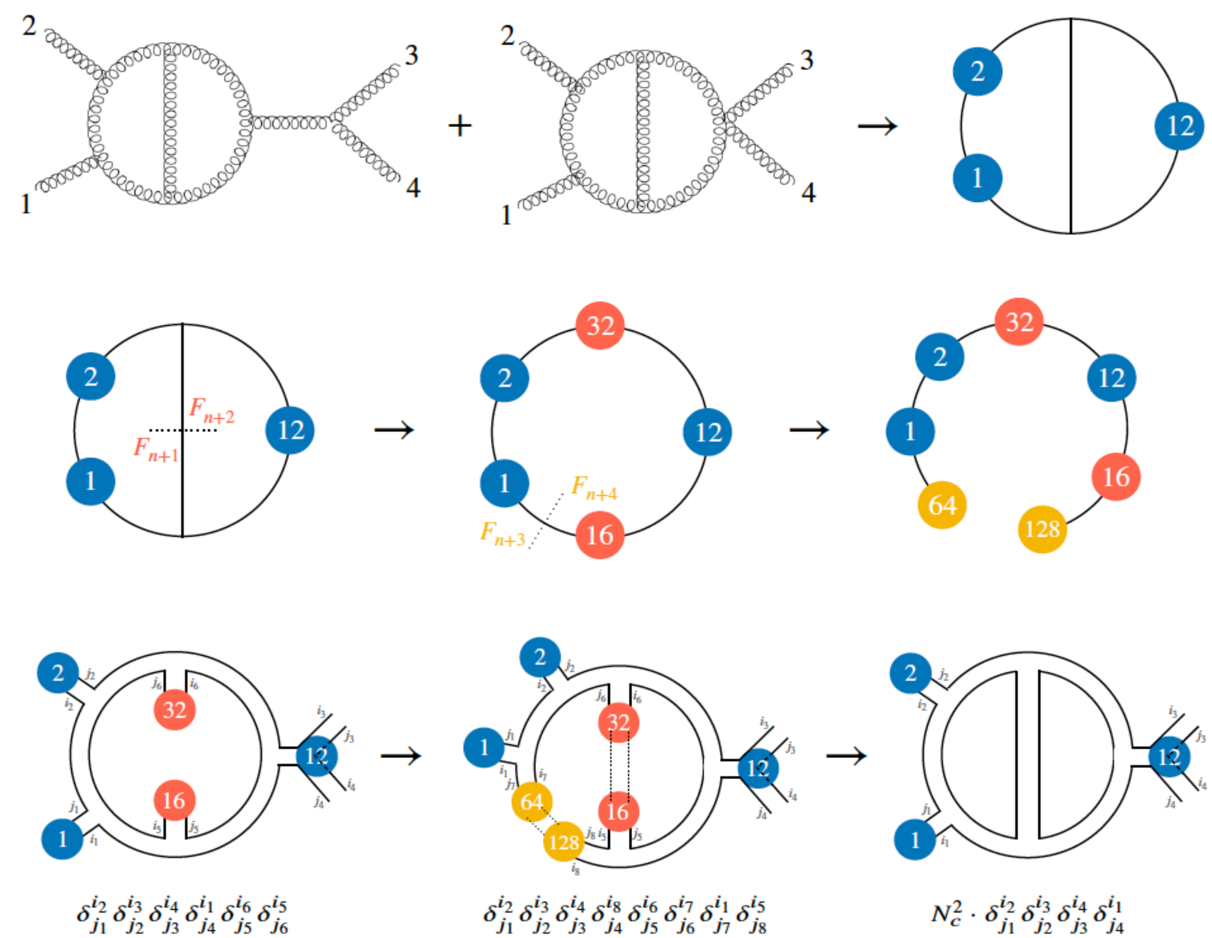


"theta"

"infinity"

"dumbbell"

2-loop numerator computation strategy



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

Thank you for your attention

Ευχαριστώ για τη προσοχή σας

Backup slides

2-loop reduction: a toy example

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

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

2-loop reduction: a toy example

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

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

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

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

- 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

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

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

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

- 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

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

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

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

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

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

- 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