PAUL SCHERRER INSTITUT

FONDS NATIONAL SUISSE
SCHWEIZERISCHER NATIONALFONDS
FONDO NAZIONALE SVIZZERO
SWISS NATIONAL SCIENCE FOUNDATION

Two-loop amplitude generation in OpenLoops

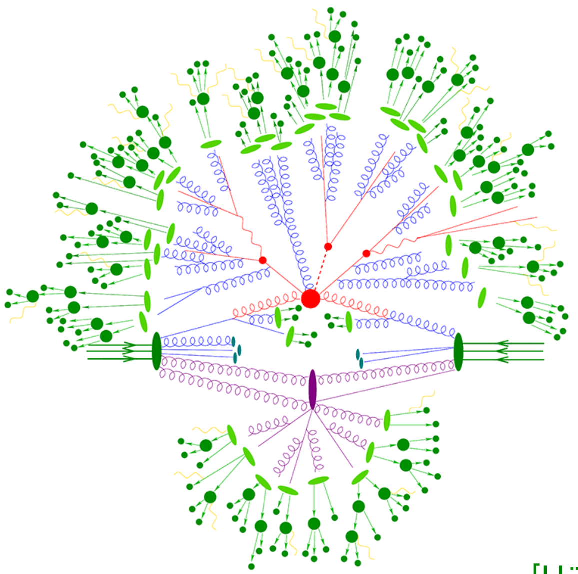
M. F. Zoller

in collaboration with

S. Pozzorini and N. Schär

RADCOR – LoopFest 2021

OpenLoops



[Höche]

OpenLoops is a fully automated numerical tool for the **tree and one-loop** computation of **hard scattering amplitudes** required in **Monte-Carlo simulations**

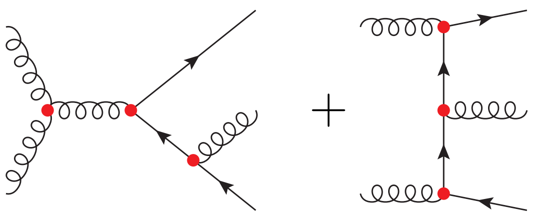
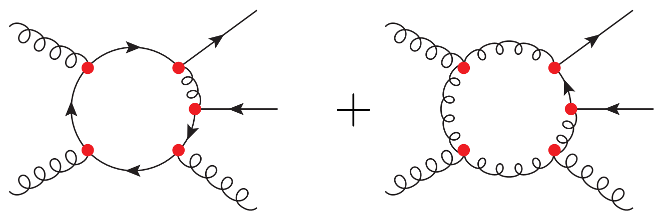
- Full $\mathrm{NLO\ QCD}$ and $\mathrm{NLO\ EW}$ corrections available
- Strong CPU performance and excellent numerical stability

Available from `https://gitlab.com/openloops/OpenLoops.git`

or `https://openloops.hepforge.org`

Scattering probability densities in perturbation theory from sums of L-loop Feynman diagrams (L=0,1):

$$\mathcal{W}_{00} = \sum_{\mathrm{hel}} \sum_{\mathrm{col}} |\mathcal{M}_0|^2, \qquad \mathcal{W}_{01} = \sum_{\mathrm{hel}} \sum_{\mathrm{col}} 2\,\mathrm{Re}\Big[\mathcal{M}_0^* \mathcal{M}_1\Big], \qquad \mathcal{W}_{11} = \sum_{\mathrm{hel}} \sum_{\mathrm{col}} |\mathcal{M}_1|^2$$

$\mathcal{M}_0 =$  $+ \ldots$ $\qquad \mathcal{M}_1 =$  $+ \ldots$

Automation at NNLO highly desirable → Goal: Two-loop OpenLoops

Outline

I. The OpenLoops algorithm at tree level and one loop

II. Requirements for two-loop automation

III. New algorithm to construct two-loop tensor coefficients in OpenLoops

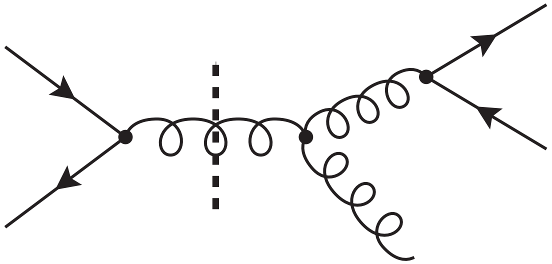
IV. Numerical stability

V. Timings

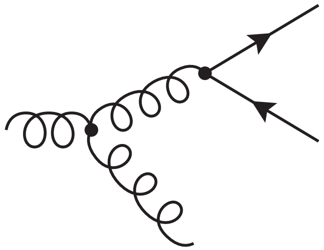
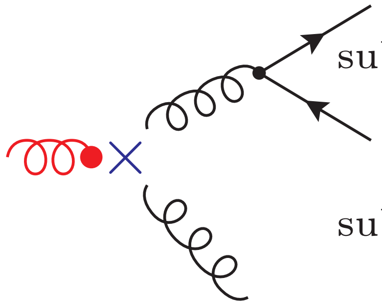
VI. Summary and Outlook

I. The OpenLoops algorithm at tree level

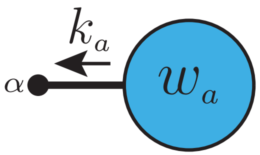
Tree-level amplitudes constructed recursively from sub-trees (starting from external lines)

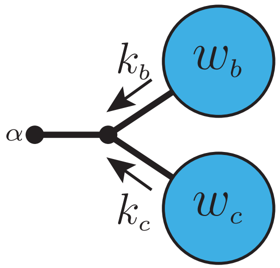
For example $\mathcal{M}_0 =$  $+ \dots$ $\rightarrow$ split into sub-trees

Numerical recursion step:

$w_a^\alpha =$  $=$ 

$$= \underbrace{\frac{X^\alpha_{\beta\gamma}(k_b, k_c)}{k_a^2 - m_a^2}}_{\text{universal building block from Feynman rules}} \; w_b^\beta \; w_c^\gamma$$

Generic depiction: 

$=$ 

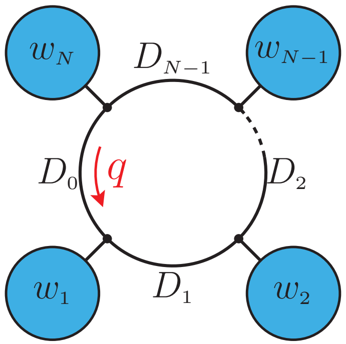
(k_i external momenta)

Highly efficient: Sub-trees constructed only once for multiple tree and loop diagrams

I. The OpenLoops algorithm at one loop

High complexity in loop diagram Γ due to analytical structure in loop momentum q

$$\mathcal{M}_{1,\Gamma} =$$

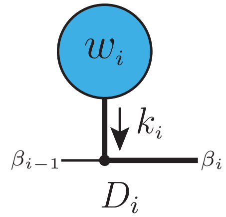


$$= \mathcal{C}_{1,\Gamma} \int \mathrm{d}^D q \, \frac{S_1(q) \cdots S_N(q)}{D_0 \cdots D_{N-1}}$$

Scalar propagators $D_i(q) = (q + p_i)^2 - m_i^2$

Factorisation into colour factor $\mathcal{C}_{1,\Gamma}$ and **loop segments**

$$S_i(q) =$$

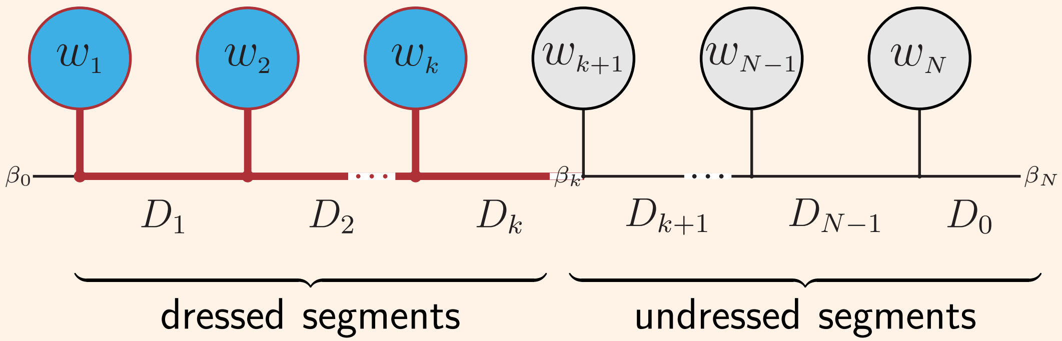


$$= X_i^{\alpha}(k_i, p_i, q) w_i^{\alpha}$$

Universal building block $\times$ sub-tree(s)

Open loop diagram at D_0 $\rightarrow$ **Dress chain of segments (open loop) recursively**:

$$\mathcal{N}_k(q) = \prod_{i=1}^{k} S_i(q) = \mathcal{N}_{k-1}(q) S_k(q) =$$



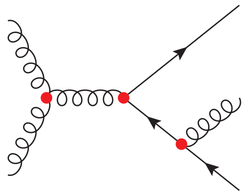
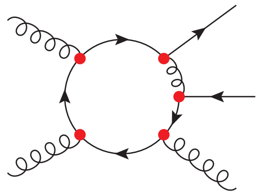
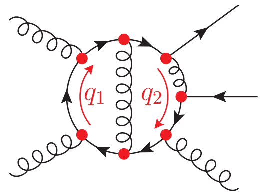
$$= \sum_{r=0}^{k} \mathcal{N}^{(k)}_{\mu_1 \ldots \mu_r} q^{\mu_1} \ldots q^{\mu_r}$$

Completely generic and highly efficient algorithm

Implemented at the level of tensor integral coefficients $\mathcal{N}^{(k)}_{\mu_1 \ldots \mu_r}$

II. Requirements for NNLO automation

NNLO scattering probability density: $\mathcal{W}_{\text{NNLO}}^{\text{virtual}} = \sum_{\text{hel}} \sum_{\text{col}} \left(2\,\mathrm{Re}\left[\mathcal{M}_0^* \mathcal{M}_2\right] + |\mathcal{M}_1|^2 \right)$

$\mathcal{M}_0 =$  $+ \ldots$ $\quad \mathcal{M}_1 =$  $+ \ldots$ $\quad \mathcal{M}_2 =$ 

$+ \ldots$

- $|\mathcal{M}_1|^2$, $\mathcal{W}_{\text{NNLO}}^{\text{real-virtual}}$, $\mathcal{W}_{\text{NNLO}}^{\text{real-real}}$ available in OpenLoops [Buccioni, Lang, Lindert, Maierhöfer, Pozzorini, Zhang, M.Z.]

- Amplitude of a two-loop diagram Γ:

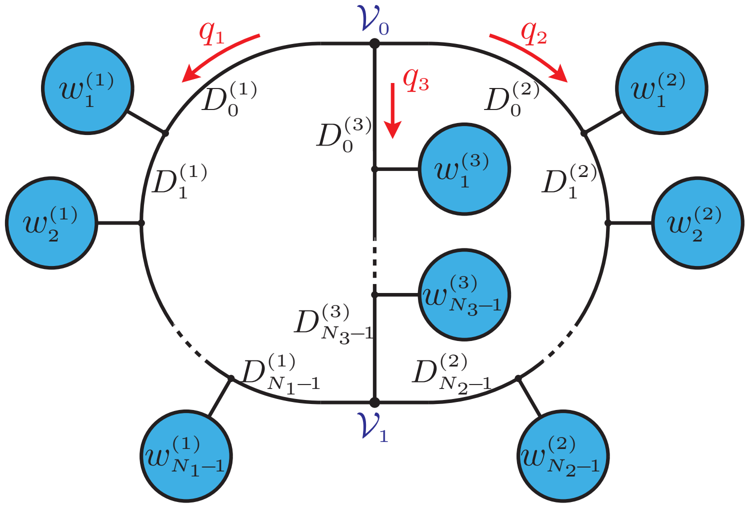
$$\mathcal{M}_{2,\Gamma} = \underbrace{\mathcal{C}_{2,\Gamma}}_{\text{colour}} \sum_{r_1=0}^{R_1} \sum_{r_2=0}^{R_2} \underbrace{\mathcal{N}_{\mu_1\cdots\mu_{r_1}\nu_1\cdots\nu_{r_2}}}_{\text{tensor coefficient}} \underbrace{\int \mathrm{d}^D q_1 \int \mathrm{d}^D q_2 \, \frac{q_1^{\mu_1}\cdots q_1^{\mu_{r_1}}\, q_2^{\nu_1}\cdots q_2^{\nu_{r_2}}}{\mathcal{D}(q_1,q_2)}}_{\text{tensor integral}}$$

  - **Numerical construction of tensor coefficients** in four dimensions $\rightarrow$ this talk
  - Restoration of $(D-4)$-dimensional numerator parts and renormalisation through UV and rational counterterms $\rightarrow$ Hantian Zhang's talk on Friday
  - Remaining tasks: Tensor integrals, treatment of IR divergences

III. New algorithm to construct two-loop tensor coefficients in OpenLoops

Amplitude of irreducible two-loop diagram Γ (1PI on amputation of all external subtrees):

$$\mathcal{M}_{2,\Gamma} \; = $$



$$= \mathcal{C}_{2,\Gamma} \int \mathrm{d}^D q_1 \int \mathrm{d}^D q_2 \frac{\mathcal{N}(q_1,q_2)}{\prod_{i=1}^{3} \mathcal{D}^{(i)}(q_i)}\Bigg|_{q_3 \to -(q_1+q_2)}$$

Exploit factorisation of numerator $\mathcal{N}(q_1,q_2) = \prod_{i=1}^{3} \mathcal{N}^{(i)}(q_i) \prod_{j=0}^{1} \mathcal{V}_j(q_1,q_2)$

- **Three chains**, each depending on a single loop momentum q_i $(i = 1,2,3)$

  with **chain numerators factorising into loop segments** $\mathcal{N}^{(i)}(q_i) = S_0^{(i)}(q_i) \cdots S_{N_i-1}^{(i)}(q_i)$

  $\rightarrow$ **Same structure as one-loop chain**

- **Two connecting vertices** $\mathcal{V}_0, \mathcal{V}_1$
- **Chain denominators** $\mathcal{D}^{(i)}(q_i) = D_0^{(i)}(q_i) \cdots D_{N_i-1}^{(i)}(q_i)$ where $D_a^{(i)}(q_i) = (q_i + p_{ia})^2 - m_{ia}^2$

  (External momenta p_{ia} and masses m_{ia} along i-th chain)

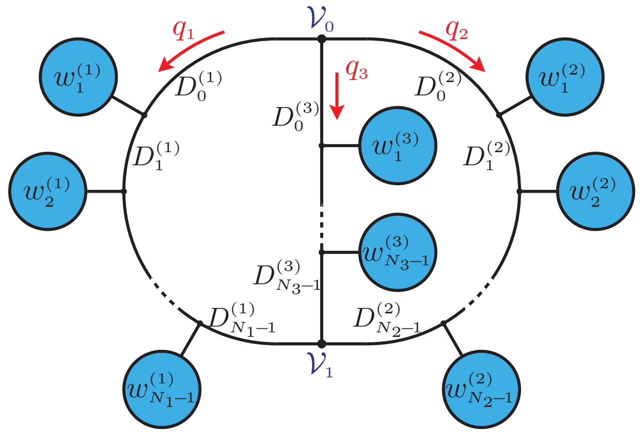
Building blocks of two-loop amplitudes

Numerator $\mathcal{N}(q_1, q_2) = \left\{ \prod_{i=1}^{3} \left[\mathcal{N}^{(i)}(q_i)\right]_{\beta_0^{(i)}}^{\beta_{N_i}^{(i)}} \right\} \left[\mathcal{V}_0(q_1, q_2)\right]^{\beta_0^{(1)}\beta_0^{(2)}\beta_0^{(3)}} \left[\mathcal{V}_1(q_1, q_2)\right]_{\beta_{N_1}^{(1)}\beta_{N_2}^{(2)}\beta_{N_3}^{(3)}}$

can be constructed recursively, multiplying one chain segment or vertex $\mathcal{V}_j$ per recursion step.

Observations:

- Chains have same complexity as one-loop chains
- Higher complexity in steps connecting $\mathcal{V}_j$ due to dependence on q_1, q_2 and three open Lorentz/spinor indices $\beta_k^{(i)}$
- Each chain segment or vertex $\mathcal{V}_j$ increases helicity d.o.f. by those of its external subtree(s) and the rank in a q_i by 0 or 1
- Number of independent tensor coefficients $\mathcal{N}_{\mu_1\cdots\mu_{r_1}\nu_1\cdots\nu_{r_2}}$ increases exponentially with ranks r_1, r_2 in q_1,q_2



Number of tensor components

r_1 \ r_2	0	1	2	3
0	1	5	15	35
1	5	25	75	175
2	15	75	225	525
3	35	175	525	1225
4	70	350	1050	2450
5	126	630	1890	4410

Naive algorithm: Dress all chains first, then connect $\mathcal{V}_{0,1}$, interfer with Born and sum over helicities

→ Would be inefficient due to expensive last steps

Connecting the building blocks of two-loop amplitudes

Final result: Helicity and colour-summed interference with Born $\mathcal{U}(q_1, q_2)$

$$= \sum_h 2\Big(\sum_{\text{col}} \underbrace{\mathcal{M}_0^*(h)}_{\text{Born}} \underbrace{C_{2,\Gamma}}_{\text{colour}}\Big)\Big\{\prod_{i=1}^{3} \underbrace{\Big[\prod_{k=0}^{N_i-1} S_k^{(i)}(q_i, h_k^{(i)})\Big]_{\beta_0^{(i)}}^{\beta_{N_i}^{(i)}}}_{\text{chain } \mathcal{N}^{(i)}(h^{(i)})}\Big\} \underbrace{\Big[\mathcal{V}_0(q_1, q_2)\Big]^{\beta_0^{(1)}\beta_0^{(2)}\beta_0^{(3)}} \Big[\mathcal{V}_1(q_1, q_2)\Big]_{\beta_{N_1}^{(1)}\beta_{N_2}^{(2)}\beta_{N_3}^{(3)}}}_{\text{two-loop vertices}}$$

with segment helicities $h_k^{(i)}$ $\rightarrow$ chain helicities $h^{(i)} = \sum_{k=0}^{N_i-1} h_k^{(i)}$ $\rightarrow$ global helicity $h = \sum_{i=1}^{3} h^{(i)}$

Algorithm will consist of N recursion steps: $\boxed{\mathcal{N}_n = \mathcal{N}_{n-1} \cdot S_n, \quad (n = 1, \ldots, N)}$

with partially dressed numerators $\mathcal{N}_n$ and building blocks $S_n \in \{S_k^{(i)}, \mathcal{V}_j, \mathcal{N}^{(i)}, \mathcal{M}_0^* C_{2,\Gamma}\}$.

CPU cost of step n $\sim$ number of multiplications

$\rightarrow$ dependent on structure of S_n and number of components of $\mathcal{N}_n$

$=$ (number of tensor components in q_i) $\times$ (helicitiy d.o.f.) $\times$ $4^{\text{(number of open Lorentz/spinor indices)}}$

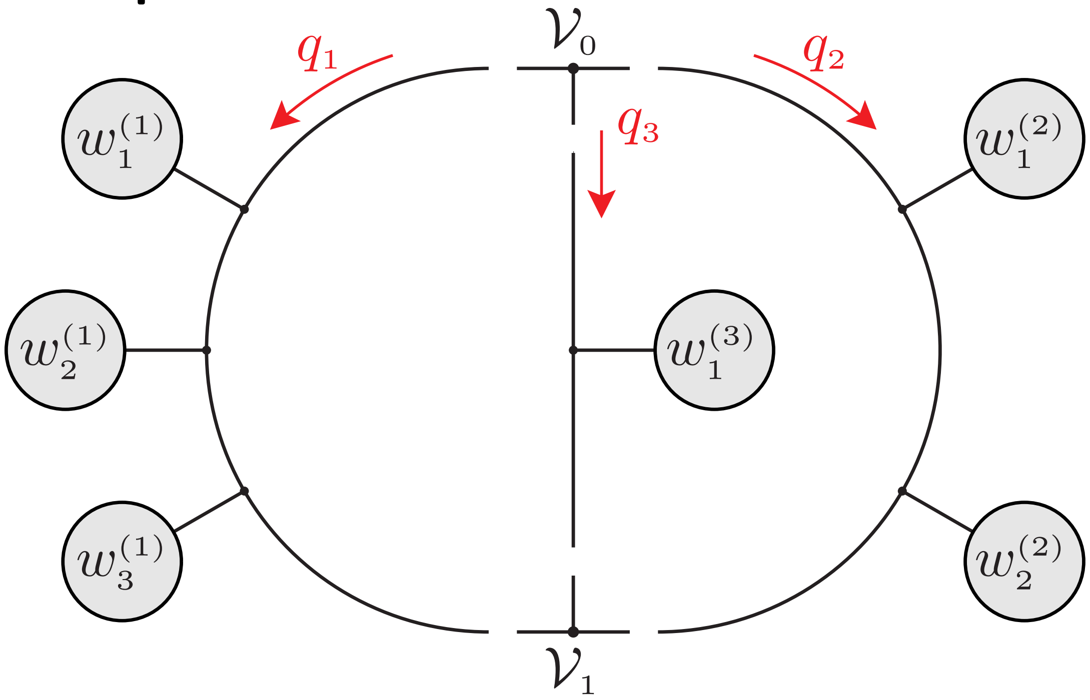
$\Rightarrow$ **Most efficient algorithm found through cost simulation**

of possible candidates for a wide range of QED and QCD Feynman diagrams

New two-loop algorithm

- Sort chains by length: $N_1 \geq N_2 \geq N_3$
  Choose order of $\mathcal{V}_0, \mathcal{V}_1$ by vertex type

Example:



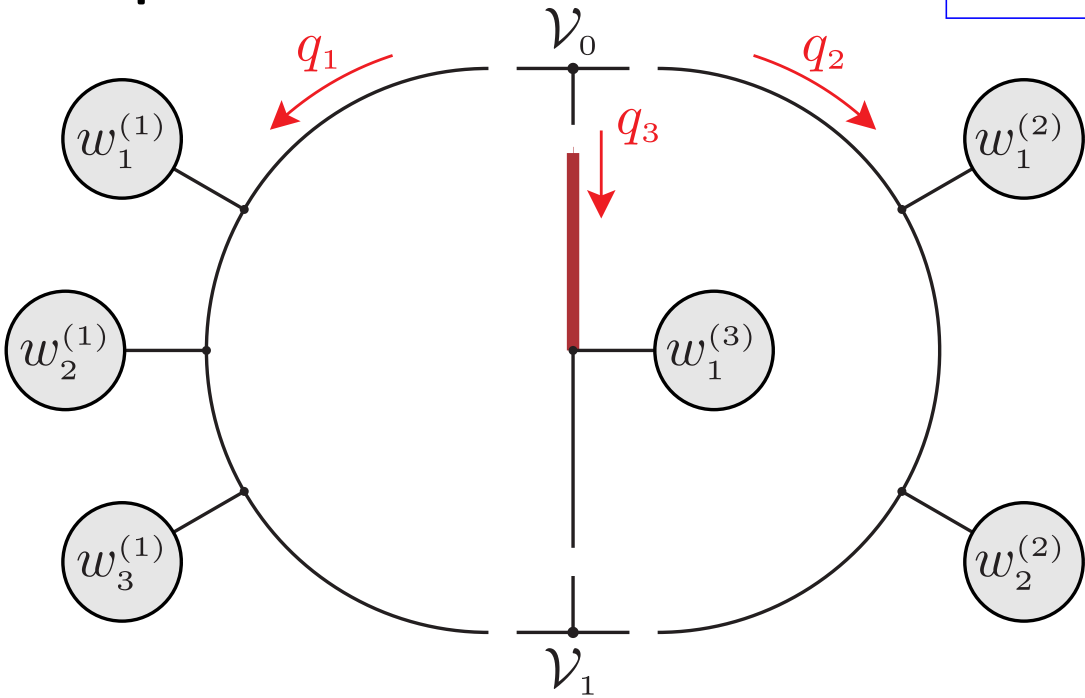
Order of chains and of two-loop vertices $\mathcal{V}_0, \mathcal{V}_1$ has major impact on efficiency

New two-loop algorithm

- Sort chains by length: $N_1 \geq N_2 \geq N_3$
  Choose order of $\mathcal{V}_0, \mathcal{V}_1$ by vertex type
- Dress $\mathcal{N}^{(3)}$ (shortest chain)

Example:

$n = 0$



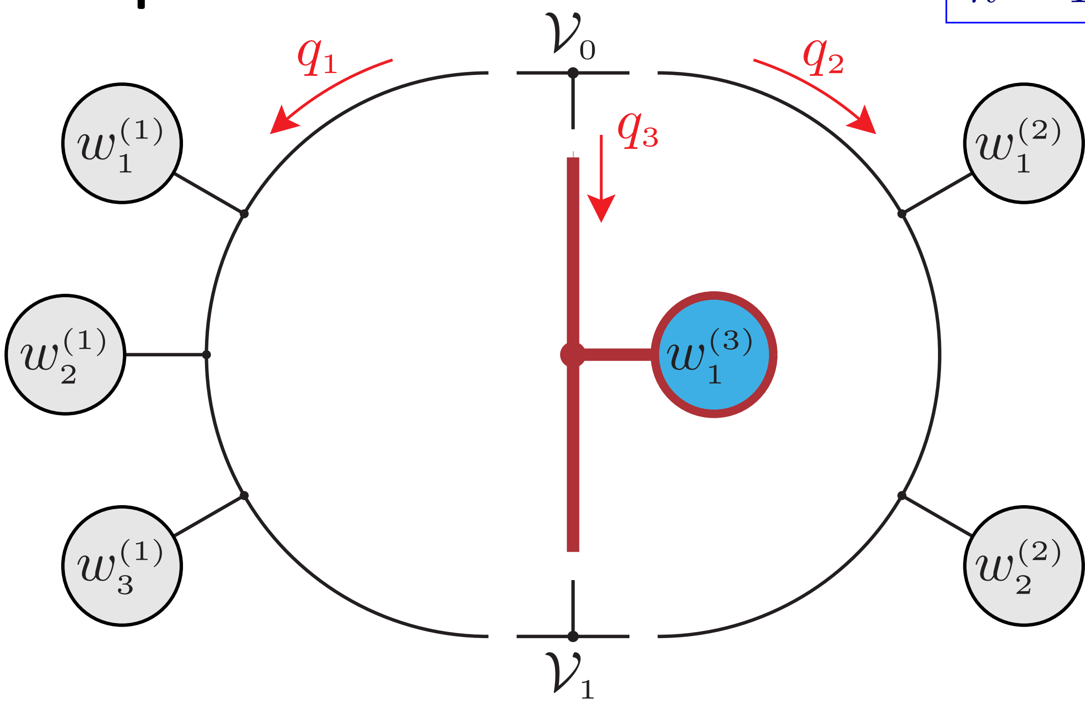
$$\mathcal{N}_n^{(3)}(q_3, \hat{h}_n^{(3)}) = \mathcal{N}_{n-1}^{(3)}(q_3, \hat{h}_{n-1}^{(3)}) \cdot S_n^{(3)}(q_3, h_n^{(3)})$$ with initial condition $\mathcal{N}_{-1}^{(3)} = \mathbb{1}$

New two-loop algorithm

- Sort chains by length: $N_1 \geq N_2 \geq N_3$
  Choose order of $\mathcal{V}_0, \mathcal{V}_1$ by vertex type
- Dress $\mathcal{N}^{(3)}$ (shortest chain)

Example:

$n = 1$



$$\mathcal{N}_n^{(3)}(q_3, \hat{h}_n^{(3)}) = \mathcal{N}_{n-1}^{(3)}(q_3, \hat{h}_{n-1}^{(3)}) \cdot S_n^{(3)}(q_3, h_n^{(3)}) \qquad \text{with initial condition } \mathcal{N}_{-1}^{(3)} = \mathbb{1}$$

- Shortest chain $\Rightarrow$ Low number of helicity d.o.f. $\hat{h}_n^{(3)} = \hat{h}_{n-1}^{(3)} + h_n^{(3)}$ and low rank in q_3
- Partial chains $\mathcal{N}_n^{(3)}$ computed only once for multiple diagrams

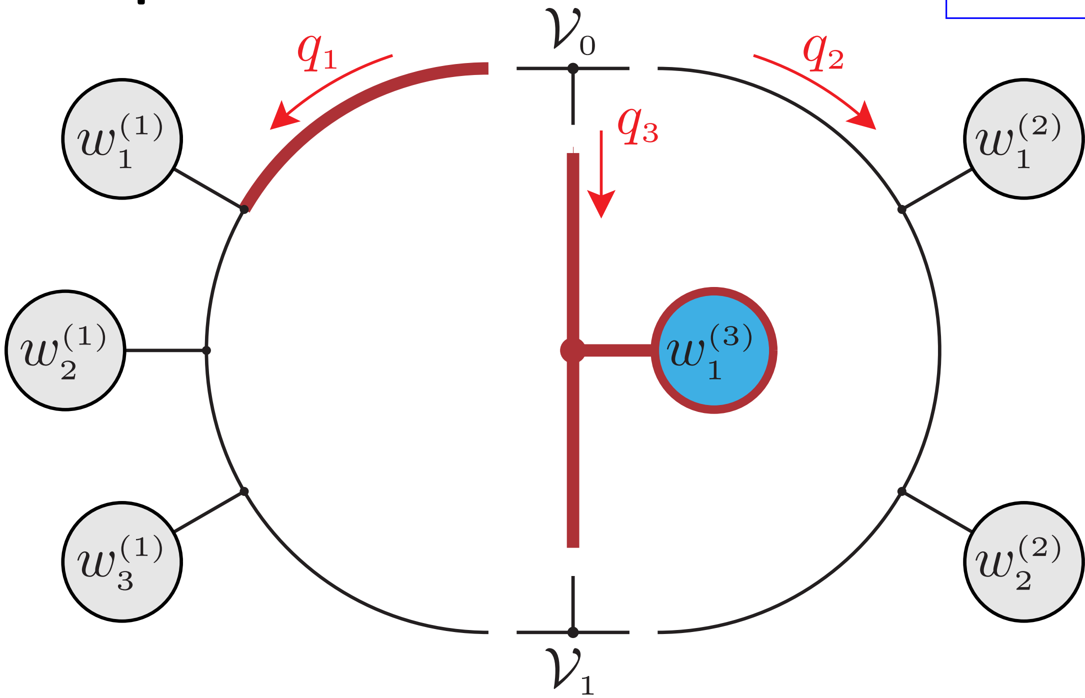
$\Rightarrow$ **Only a small number of low-complexity steps for the full process**

New two-loop algorithm

- Sort chains by length: $N_1 \geq N_2 \geq N_3$
  Choose order of $\mathcal{V}_0, \mathcal{V}_1$ by vertex type
- Dress $\mathcal{N}^{(3)}$ (shortest chain)
- Dress $\mathcal{U}^{(1)} \propto \mathcal{M}_0^* \mathcal{N}^{(1)}$ (longest chain)

Example:

$n = 0$



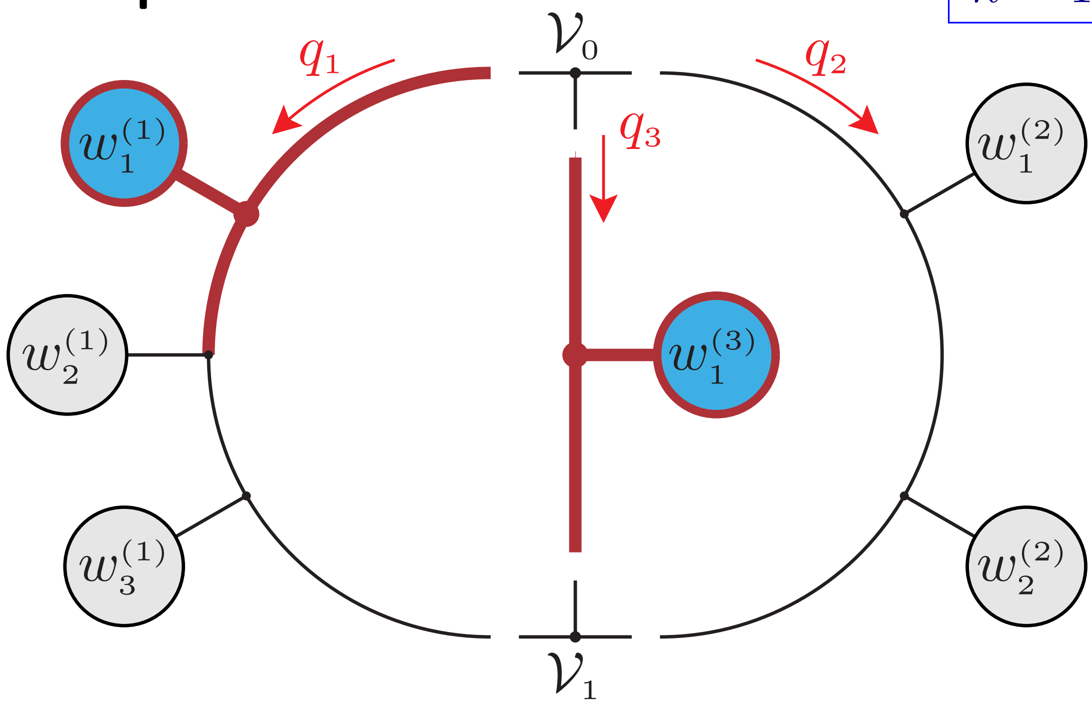
$$\mathcal{U}_n^{(1)}(q_1, \check{h}_n^{(1)}) = \sum_{h_n^{(1)}} \mathcal{U}_{n-1}^{(1)}(q_1, \check{h}_{n-1}^{(1)}) \cdot S_n^{(1)}(q_1, h_n^{(1)}) \qquad \text{with} \quad \mathcal{U}_{-1}^{(1)}(h) = 2\Big(\sum_{\text{col}} \underbrace{\mathcal{M}_0^*(h)}_{\text{Born}} \underbrace{C_{2,\Gamma}}_{\text{colour}} \Big)$$

New two-loop algorithm

- Sort chains by length: $N_1 \geq N_2 \geq N_3$
  Choose order of $\mathcal{V}_0, \mathcal{V}_1$ by vertex type
- Dress $\mathcal{N}^{(3)}$ (shortest chain)
- Dress $\mathcal{U}^{(1)} \propto \mathcal{M}_0^* \mathcal{N}^{(1)}$ (longest chain)

Example:

$n = 1$



$$\mathcal{U}_n^{(1)}(q_1, \check{h}_n^{(1)}) = \sum_{h_n^{(1)}} \mathcal{U}_{n-1}^{(1)}(q_1, \check{h}_{n-1}^{(1)}) \cdot S_n^{(1)}(q_1, h_n^{(1)}) \qquad \text{with} \quad \mathcal{U}_{-1}^{(1)}(h) = 2\Big(\sum_{\text{col}} \underbrace{\mathcal{M}_0^*(h)}_{\text{Born}} \underbrace{C_{2,\Gamma}}_{\text{colour}}\Big)$$

On-the-fly summation of segment helicities $h_n^{(1)}$

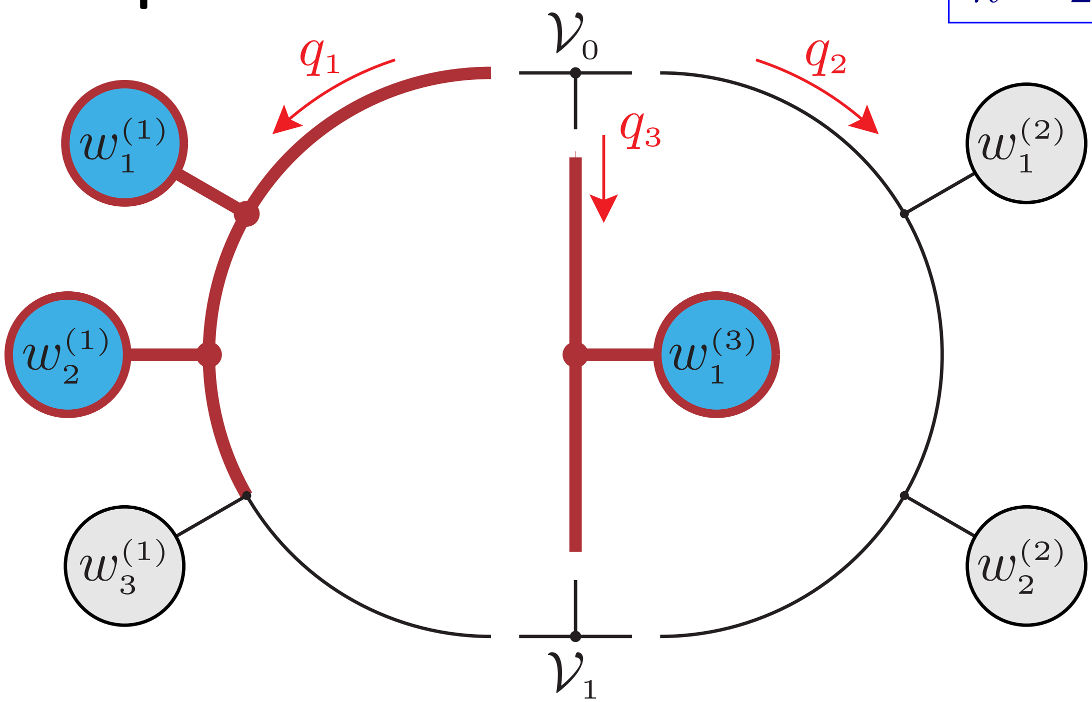
$\Rightarrow$ Partial chains depend on remaining helicities of the diagram $\check{h}_n^{(1)} = h - \sum_{k=1}^{n} h_k^{(1)}$

New two-loop algorithm

- Sort chains by length: $N_1 \geq N_2 \geq N_3$
  Choose order of $\mathcal{V}_0, \mathcal{V}_1$ by vertex type
- Dress $\mathcal{N}^{(3)}$ (shortest chain)
- Dress $\mathcal{U}^{(1)} \propto \mathcal{M}_0^* \mathcal{N}^{(1)}$ (longest chain)

Example:

$n = 2$



$$\mathcal{U}_n^{(1)}(q_1, \check{h}_n^{(1)}) = \sum_{h_n^{(1)}} \mathcal{U}_{n-1}^{(1)}(q_1, \check{h}_{n-1}^{(1)}) \cdot S_n^{(1)}(q_1, h_n^{(1)}) \qquad \text{with} \quad \mathcal{U}_{-1}^{(1)}(h) = 2\Big(\sum_{\text{col}} \underbrace{\mathcal{M}_0^*(h)}_{\text{Born}} \underbrace{C_{2,\Gamma}}_{\text{colour}}\Big)$$

On-the-fly summation of segment helicities $h_n^{(1)}$

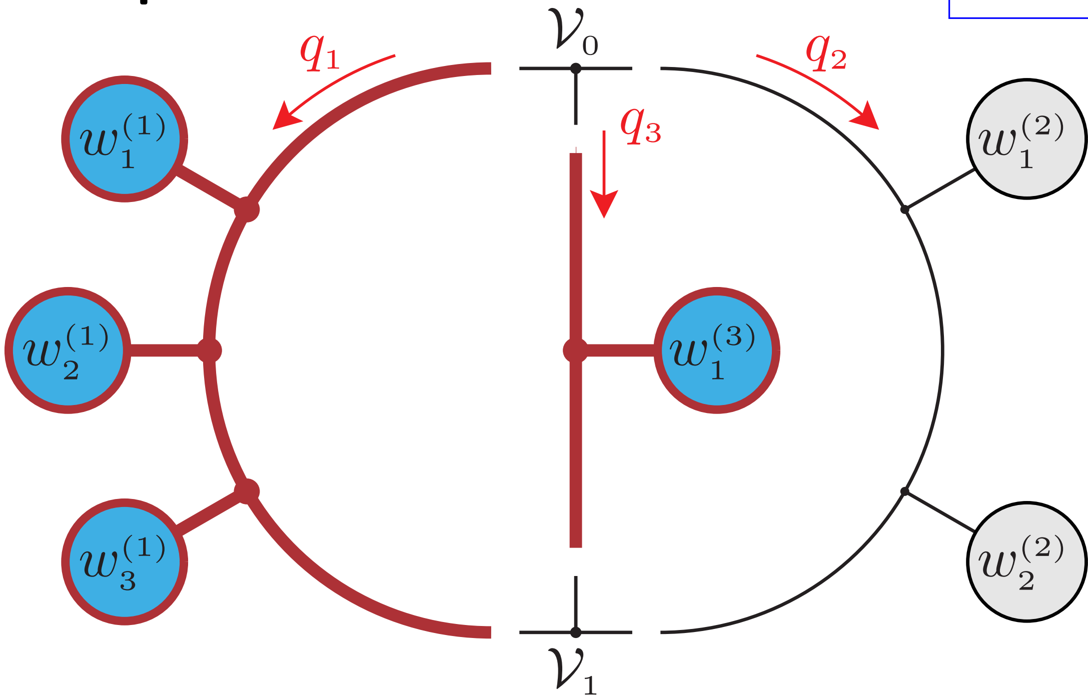
$\Rightarrow$ Partial chains depend on remaining helicities of the diagram $\check{h}_n^{(1)} = h - \sum_{k=1}^{n} h_k^{(1)}$

New two-loop algorithm

- Sort chains by length: $N_1 \geq N_2 \geq N_3$
  Choose order of $\mathcal{V}_0, \mathcal{V}_1$ by vertex type
- Dress $\mathcal{N}^{(3)}$ (shortest chain)
- Dress $\mathcal{U}^{(1)} \propto \mathcal{M}_0^* \mathcal{N}^{(1)}$ (longest chain)

Example:

$n = 3$



$$\mathcal{U}_n^{(1)}(q_1, \check{h}_n^{(1)}) = \sum_{h_n^{(1)}} \mathcal{U}_{n-1}^{(1)}(q_1, \check{h}_{n-1}^{(1)}) \cdot S_n^{(1)}(q_1, h_n^{(1)}) \qquad \text{with} \quad \mathcal{U}_{-1}^{(1)}(h) = 2\Big(\sum_{\text{col}} \underbrace{\mathcal{M}_0^*(h)}_{\text{Born}} \underbrace{C_{2,\Gamma}}_{\text{colour}} \Big)$$

On-the-fly summation of segment helicities $h_n^{(1)}$

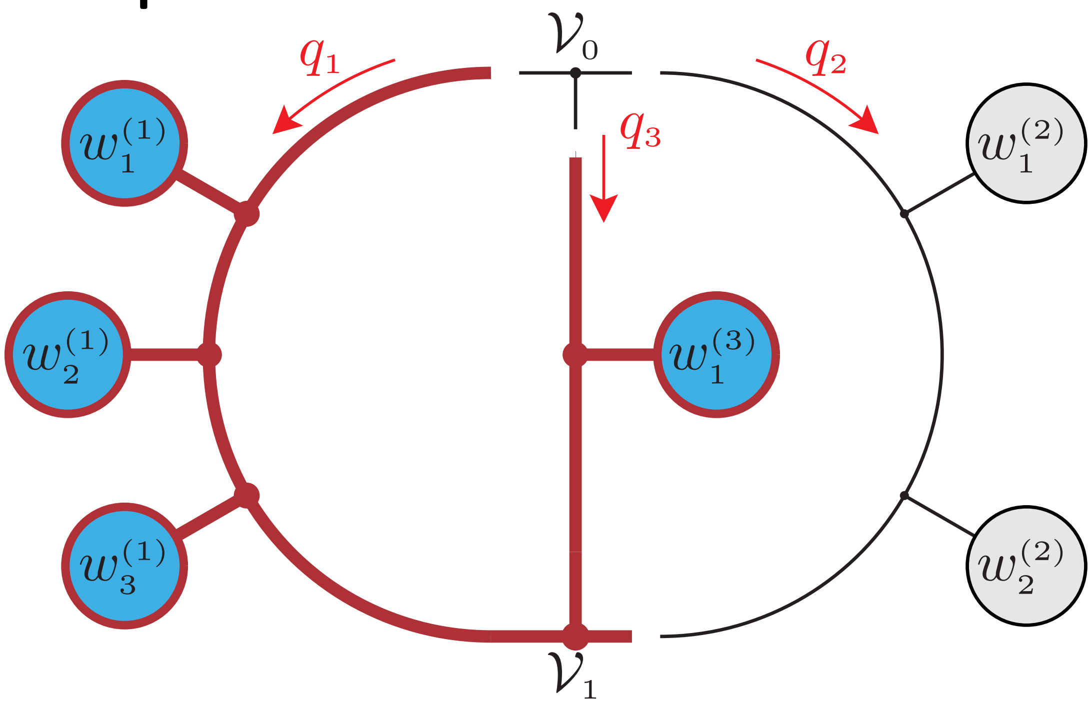
$\Rightarrow$ Partial chains depend on remaining helicities of the diagram $\check{h}_n^{(1)} = h - \sum_{k=1}^{n} h_k^{(1)}$

$\Rightarrow$ Large portion of helicity d.o.f already summed over during dressing of longest chain

New two-loop algorithm

- Sort chains by length: $N_1 \geq N_2 \geq N_3$
  Choose order of $\mathcal{V}_0, \mathcal{V}_1$ by vertex type
- Dress $\mathcal{N}^{(3)}$ (shortest chain)
- Dress $\mathcal{U}^{(1)} \propto \mathcal{M}_0^* \mathcal{N}^{(1)}$ (longest chain)
- Connect $\mathcal{V}_1$ with $\mathcal{U}^{(1)}$ and $\mathcal{N}^{(3)}$

Example:



$$\mathcal{Y}(q_1, q_3, h^{(2)}) = \sum_{h^{(3)}} \mathcal{U}^{(1)}(q_1, h - h^{(1)})\, \mathcal{N}^{(3)}(q_3, h^{(3)})\, \mathcal{V}_1(q_1, q_3)$$

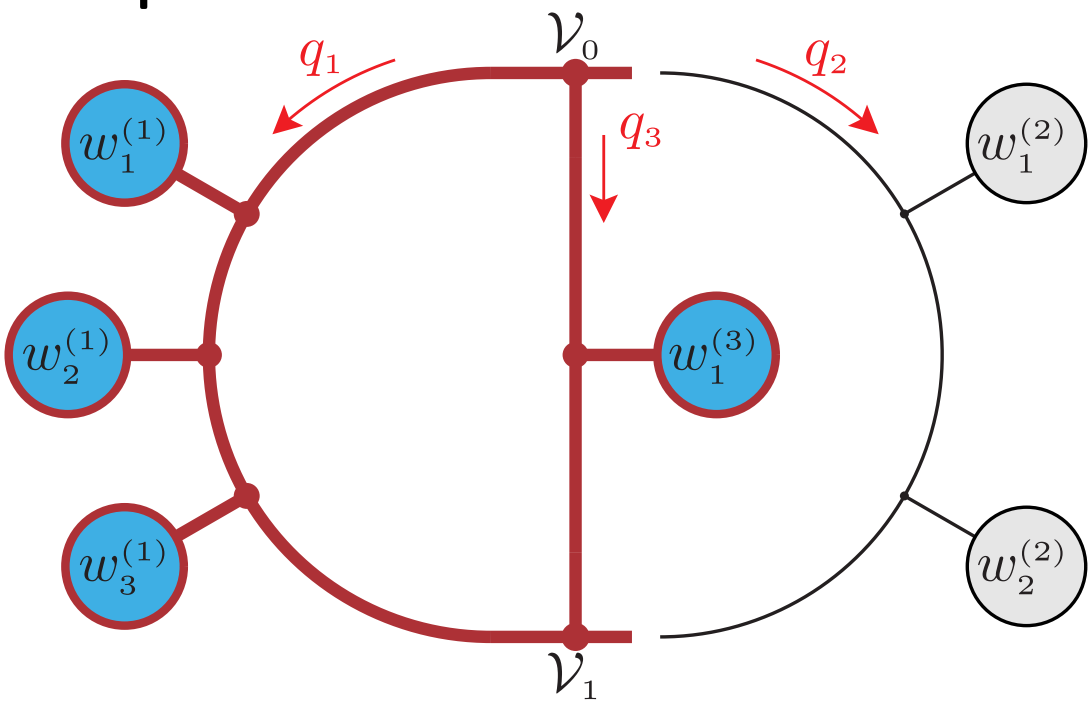
On-the-fly summation of chain helicity $h^{(3)}$ (and potential subtree helicity at $\mathcal{V}_1$)
$\Rightarrow$ Partial diagram depends on undressed chain helicity $h^{(2)}$

$\Rightarrow$ Intermediate object depends on three open indices and two loop momenta

New two-loop algorithm

- Sort chains by length: $N_1 \geq N_2 \geq N_3$
  Choose order of $\mathcal{V}_0, \mathcal{V}_1$ by vertex type
- Dress $\mathcal{N}^{(3)}$ (shortest chain)
- Dress $\mathcal{U}^{(1)} \propto \mathcal{M}_0^* \mathcal{N}^{(1)}$ (longest chain)
- Connect $\mathcal{V}_1$ with $\mathcal{U}^{(1)}$ and $\mathcal{N}^{(3)}$
- Connect $\mathcal{V}_0$ and map $q_3 \to -(q_1 + q_2)$

Example:



$$\mathcal{U}_{-1}^{(2)}(q_1, q_2, h^{(2)}) = \mathcal{Y}(q_1, q_3, h^{(2)})\ \mathcal{V}_0(q_1, q_1)\Big|_{q_3 \to -(q_1+q_2)}$$

- ◦ Partial diagram depends on undressed chain helicity $h^{(2)}$ and two open indices
- ◦ Exploit analytical q_i-structure, e.g. dependence of maximal rank R_2 in q_2 on rank $r_1 \leq R_1$ in q_1
  Example: $R_2(r_1 \leq 3) = 1$ and $R_2(r_1 = 4) = 0 \Rightarrow$ No simple $(R_1 = 4, R_2 = 1)$ array

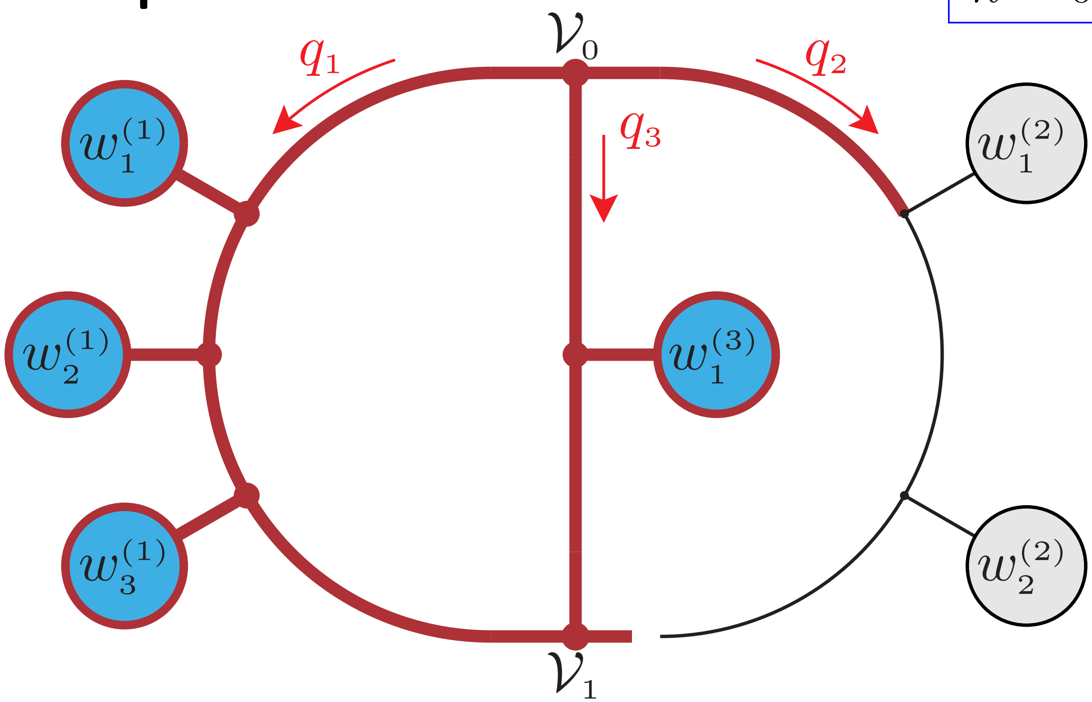
$\Rightarrow$ Use this partial diagram as initial object for the last chain dressing

New two-loop algorithm

- Sort chains by length: $N_1 \geq N_2 \geq N_3$
  Choose order of $\mathcal{V}_0, \mathcal{V}_1$ by vertex type
- Dress $\mathcal{N}^{(3)}$ (shortest chain)
- Dress $\mathcal{U}^{(1)} \propto \mathcal{M}_0^* \mathcal{N}^{(1)}$ (longest chain)
- Connect $\mathcal{V}_1$ with $\mathcal{U}^{(1)}$ and $\mathcal{N}^{(3)}$
- Connect $\mathcal{V}_0$ and map $q_3 \to -(q_1 + q_2)$
- Connect segments of $\mathcal{N}^{(2)}$

Example:

$n = 0$



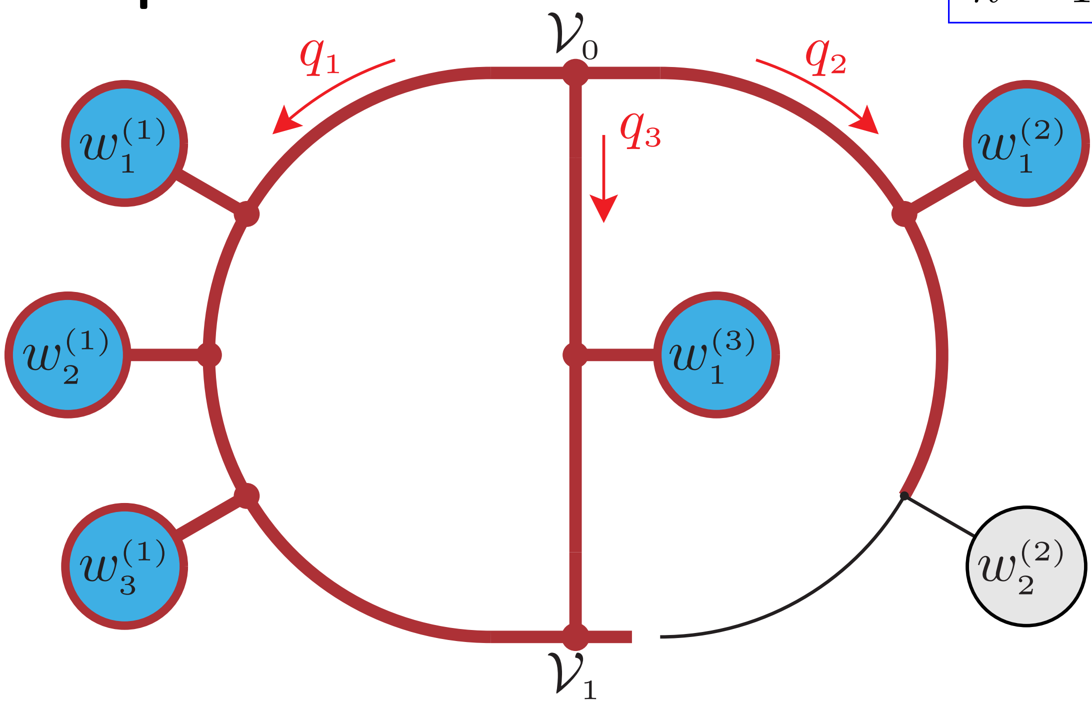
$$\mathcal{U}_n^{(2)}(q_1, q_2, \tilde{h}_n^{(2)}) = \sum_{h_n^{(2)}} \mathcal{U}_{n-1}^{(2)}(q_1, q_2, \tilde{h}_{n-1}^{(2)})\; S_n^{(2)}(q_2, h_n^{(2)})$$

New two-loop algorithm

- Sort chains by length: $N_1 \geq N_2 \geq N_3$
  Choose order of $\mathcal{V}_0, \mathcal{V}_1$ by vertex type
- Dress $\mathcal{N}^{(3)}$ (shortest chain)
- Dress $\mathcal{U}^{(1)} \propto \mathcal{M}_0^* \mathcal{N}^{(1)}$ (longest chain)
- Connect $\mathcal{V}_1$ with $\mathcal{U}^{(1)}$ and $\mathcal{N}^{(3)}$
- Connect $\mathcal{V}_0$ and map $q_3 \to -(q_1 + q_2)$
- Connect segments of $\mathcal{N}^{(2)}$

Example:

$n = 1$



$$\mathcal{U}_n^{(2)}(q_1, q_2, \tilde{h}_n^{(2)}) = \sum_{h_n^{(2)}} \mathcal{U}_{n-1}^{(2)}(q_1, q_2, \tilde{h}_{n-1}^{(2)})\ S_n^{(2)}(q_2, h_n^{(2)})$$

On-the-fly summation of segment helicities $\tilde{h}_n^{(2)} = \sum_{k=n+1}^{N_2-1} h_k^{(2)}$

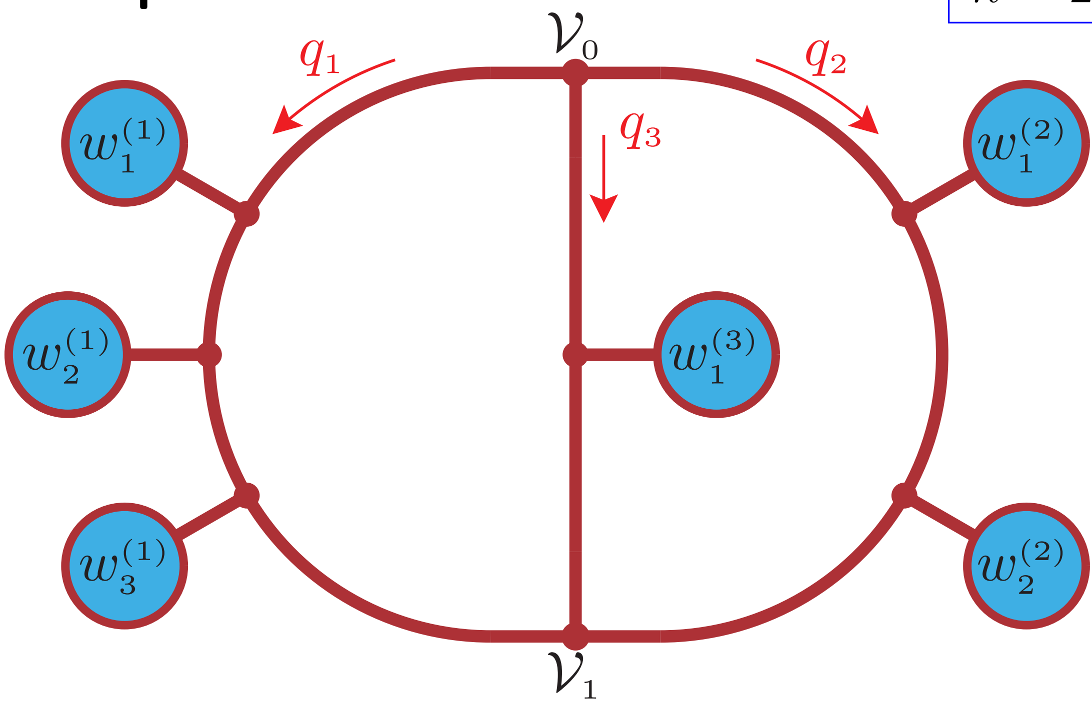
$\Rightarrow$ Partial diagram depends only on helicities of remaining undressed segments

New two-loop algorithm

- Sort chains by length: $N_1 \geq N_2 \geq N_3$
  Choose order of $\mathcal{V}_0, \mathcal{V}_1$ by vertex type
- Dress $\mathcal{N}^{(3)}$ (shortest chain)
- Dress $\mathcal{U}^{(1)} \propto \mathcal{M}_0^* \mathcal{N}^{(1)}$ (longest chain)
- Connect $\mathcal{V}_1$ with $\mathcal{U}^{(1)}$ and $\mathcal{N}^{(3)}$
- Connect $\mathcal{V}_0$ and map $q_3 \to -(q_1 + q_2)$
- Connect segments of $\mathcal{N}^{(2)}$

Example:

$n = 2$



$$\mathcal{U}_n^{(2)}(q_1, q_2, \tilde{h}_n^{(2)}) = \sum_{h_n^{(2)}} \mathcal{U}_{n-1}^{(2)}(q_1, q_2, \tilde{h}_{n-1}^{(2)}) \; S_n^{(2)}(q_2, h_n^{(2)})$$

On-the-fly summation of segment helicities $\tilde{h}_n^{(2)} = \sum_{k=n+1}^{N_2-1} h_k^{(2)}$

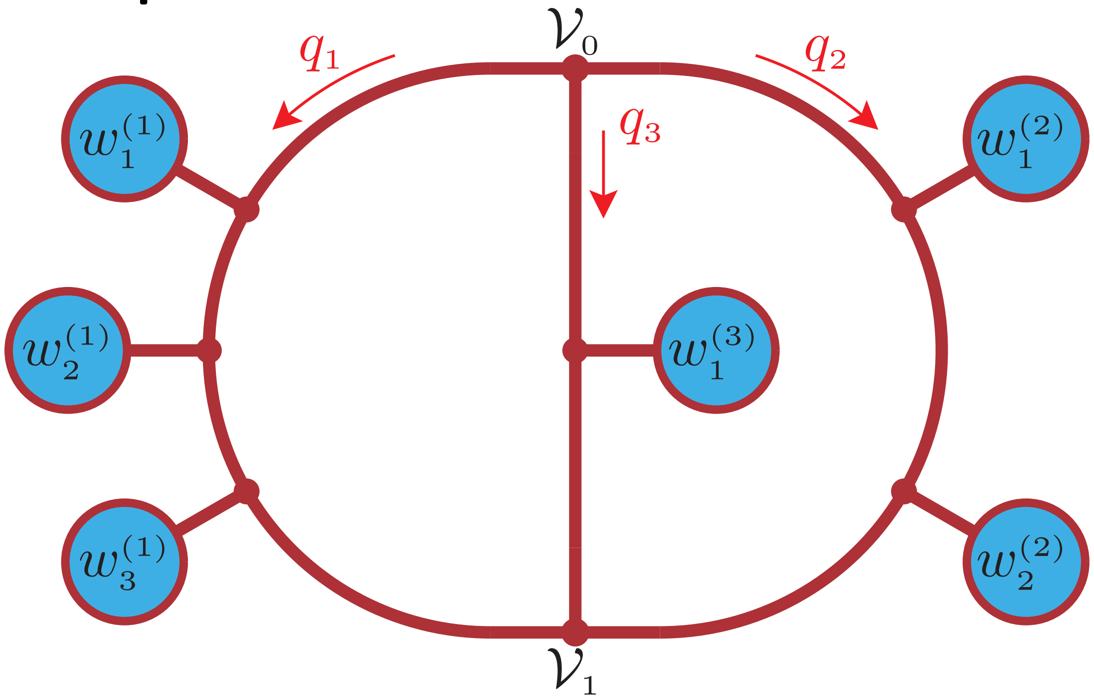
$\Rightarrow$ Partial diagram depends only on helicities of remaining undressed segments

$\Rightarrow$ Lowest complexity in helicities for steps with highest rank in loop momenta

New two-loop algorithm

- Sort chains by length: $N_1 \geq N_2 \geq N_3$
  Choose order of $\mathcal{V}_0, \mathcal{V}_1$ by vertex type
- Dress $\mathcal{N}^{(3)}$ (shortest chain)
- Dress $\mathcal{U}^{(1)} \propto \mathcal{M}_0^* \mathcal{N}^{(1)}$ (longest chain)
- Connect $\mathcal{V}_1$ with $\mathcal{U}^{(1)}$ and $\mathcal{N}^{(3)}$
- Connect $\mathcal{V}_0$ and map $q_3 \to -(q_1 + q_2)$
- Connect segments of $\mathcal{N}^{(2)}$

Example:



Exploit diagram factorisation for full process:

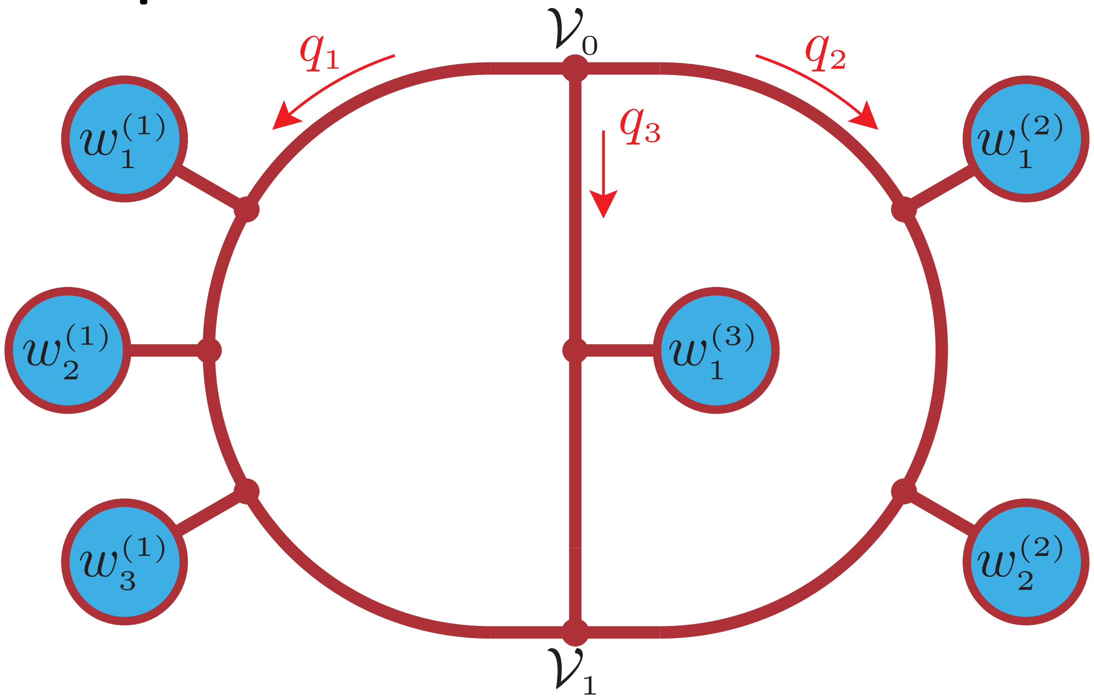
$$\mathcal{U}_A + \mathcal{U}_B = \big(\mathcal{U}_{A,n} \cdot S_{n+1} \cdots S_N\big) + \big(\mathcal{U}_{B,n} \cdot S_{n+1} \cdots S_N\big) = \big(\mathcal{U}_{A,n} + \mathcal{U}_{B,n}\big) \cdot S_{n+1} \cdots S_N$$

Merge partially dressed diagrams with same topology and subsequent recursion steps

New two-loop algorithm

- Sort chains by length: $N_1 \geq N_2 \geq N_3$
  Choose order of $\mathcal{V}_0, \mathcal{V}_1$ by vertex type
- Dress $\mathcal{N}^{(3)}$ (shortest chain)
- Dress $\mathcal{U}^{(1)} \propto \mathcal{M}_0^* \mathcal{N}^{(1)}$ (longest chain)
- Connect $\mathcal{V}_1$ with $\mathcal{U}^{(1)}$ and $\mathcal{N}^{(3)}$
- Connect $\mathcal{V}_0$ and map $q_3 \to -(q_1 + q_2)$
- Connect segments of $\mathcal{N}^{(2)}$

Example:



Exploit diagram factorisation for full process:

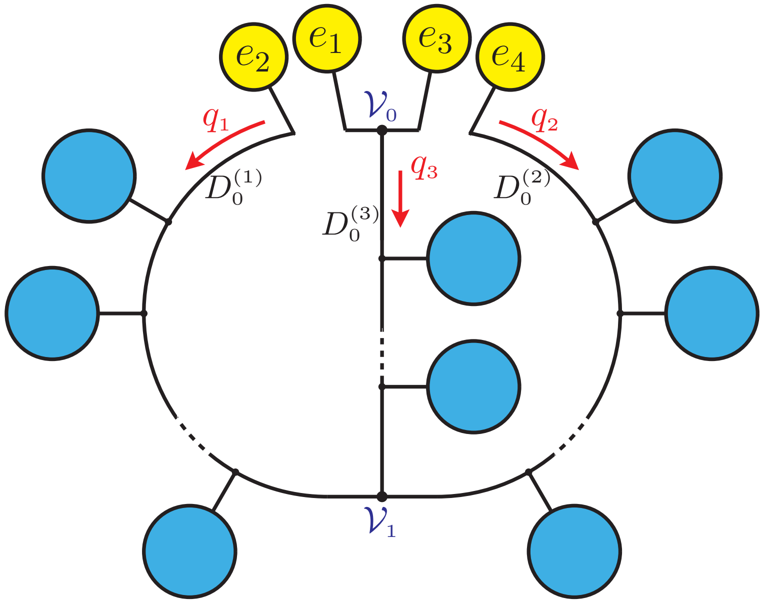
$$\mathcal{U}_A + \mathcal{U}_B = \left(\mathcal{U}_{A,n} \cdot S_{n+1} \cdots S_N\right) + \left(\mathcal{U}_{B,n} \cdot S_{n+1} \cdots S_N\right) = \left(\mathcal{U}_{A,n} + \mathcal{U}_{B,n}\right) \cdot S_{n+1} \cdots S_N$$

Merge partially dressed diagrams with same topology and subsequent recursion steps

Highly efficient and completely generic algorithm for two-loop tensor coefficients

Fully implemented for QED and QCD corrections to the SM

IV. Numerical stability



Pseudo-tree test

- Cut diagram at two propagators
- Saturate indices with random wavefunctions $e_1, \ldots, e_4$
- Fixed values for loop momenta q_1, q_2

$\Rightarrow$ Compute with well-tested tree-level algorithm and with new two-loop algorithm

$\rightarrow$ contract coefficients $\mathcal{N}_{\mu_1 \cdots \mu_{r_1} \nu_1 \cdots \nu_{r_2}}$ with fixed-value tensor integrand $\frac{q_1^{\mu_1} \cdots q_1^{\mu_{r_1}} \, q_2^{\nu_1} \cdots q_2^{\nu_{r_2}}}{\mathcal{D}(q_1, q_2)}$

Test several processes in double and quadruple precision for 10^5 uniform random phase space points

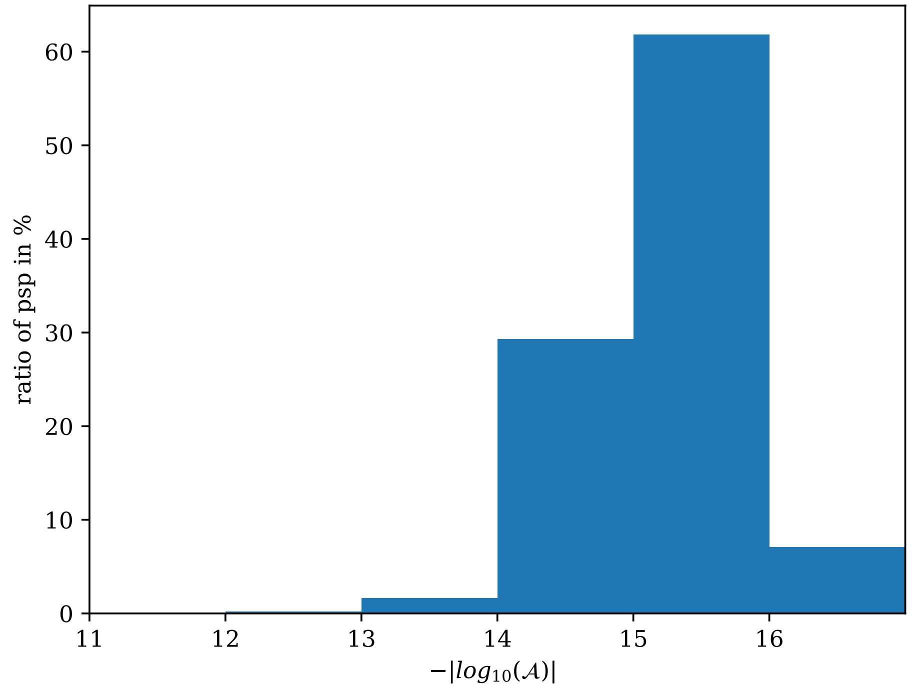
Bulk of points has $14 - 16$ digits agreement (all points 12 or more digits) in double precision
$\Rightarrow$ **Implementation validated** without computing two-loop tensor integrals

All points have more than 16 digits agreement in quad precision
$\Rightarrow$ **Quad precision calculation as benchmark**

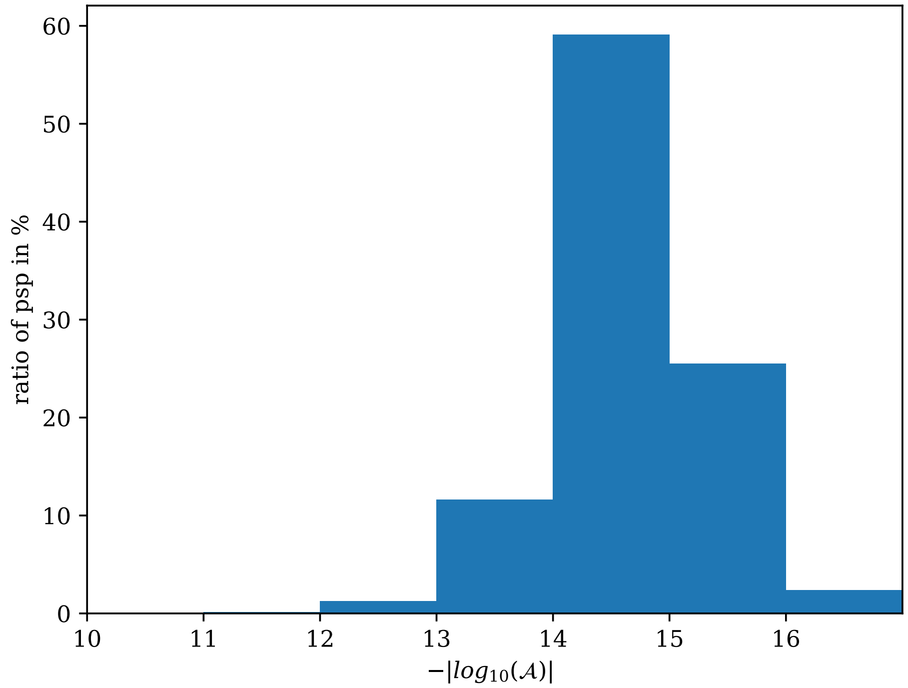
IV. Numerical stability

Two-loop algorithm with fixed q_1, q_2 and pseudo wavefunctions $e_1, \ldots, e_4$ for 10^5 uniform random phase space points (psp). Numerical instability of double (dp) wrt quad precision (qp) calculation:

$$\mathcal{A} := \frac{|\mathcal{W}_{02}^{(dp)} - \mathcal{W}_{02}^{(qp)}|}{\text{Min}(|\mathcal{W}_{02}^{(dp)}|, |\mathcal{W}_{02}^{(qp)}|)}$$



$gg \to \bar{t}t$



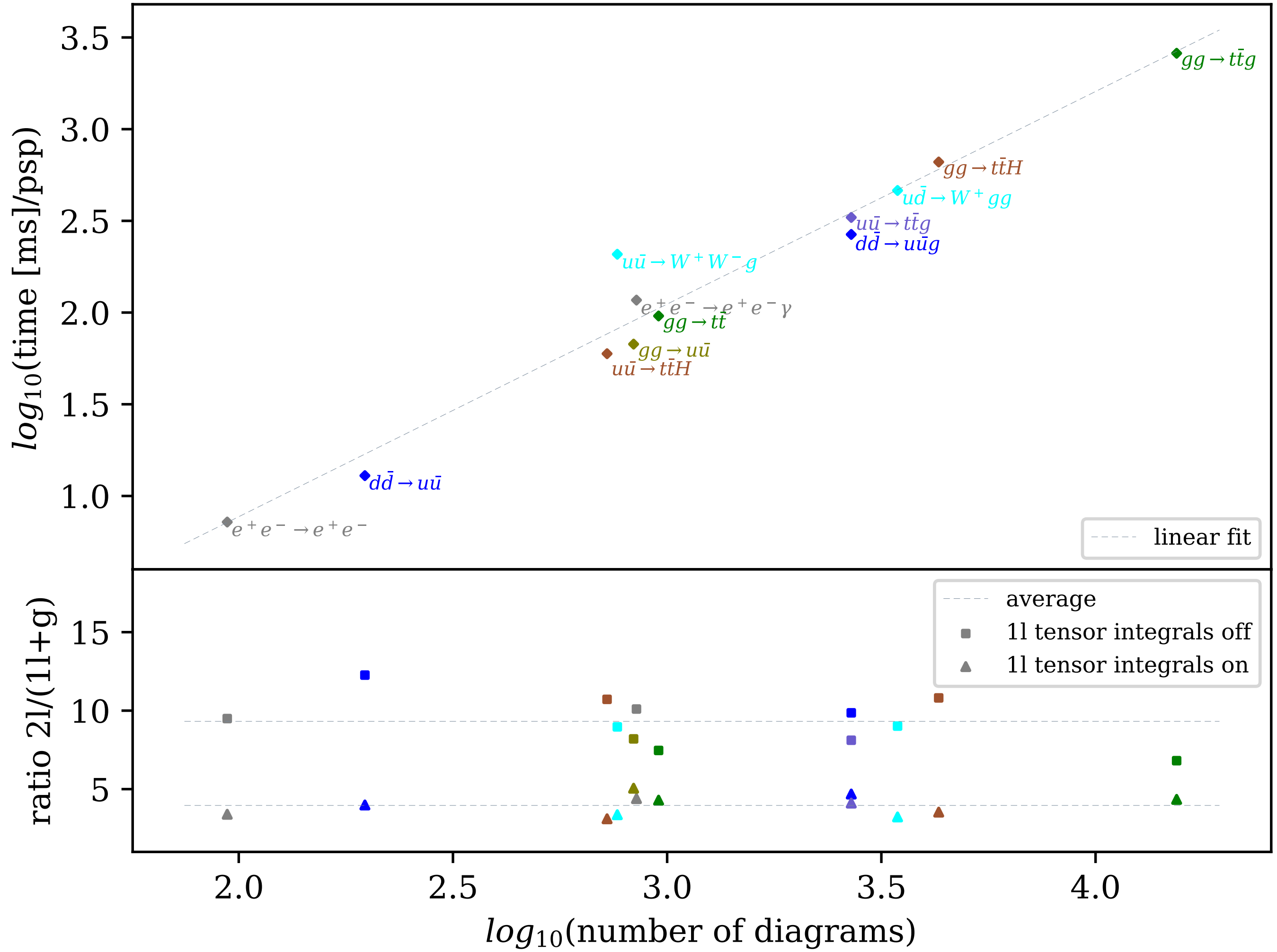
$d\bar{d} \to u\bar{u}g$

Excellent numerical stability

$\Rightarrow$ **Important for full calculation** (tensor integral reduction will be main source of instabilities)

V. Timings for two-loop tensor coefficients

QED, QCD and SM (NNLO QCD) processes (single Intel i7-6600U @ 2.6 GHz, 16GB RAM, 1000 psp)



$2 \to 2$ process: $6 - 100$ ms/psp

$2 \to 3$ process: $60 - 2500$ ms/psp (on a laptop)

Runtime $\propto$ number of diagrams

time/psp/diagram $\sim 150\mu s$

Constant ratios between NNLO virtual (2l) and real-virtual (1l+g):

$$\frac{\text{2l (tensor coefficients)}}{\text{1l+g (tensor coefficients)}} \sim 9$$

$$\frac{\text{2l (tensor coefficients)}}{\text{1l+g (full calculation)}} \sim 4$$

Strong CPU performance, comparable to real-virtual corrections in OpenLoops

VI. Summary and Outlook

Numerical calculation of **two-loop tensor coefficients** in the OpenLoops framework

- **Exploit factorisation of diagrams**
  $\rightarrow$ **Highly efficient and completely generic recursive algorithm**
- **Fully implemented for NNLO QCD and NNLO QED** corrections in the SM (irreducible and reducible two-loop diagrams)
- **Excellent numerical precision**
- **Strong CPU performance** ($\sim 150\mu s$ per diagram and psp) due to
  - Efficient order of building blocks
  - Exploitation of analytical structure in loop momenta
  - On-the-fly helicity summation and diagram merging

Short-term and mid-term projects:

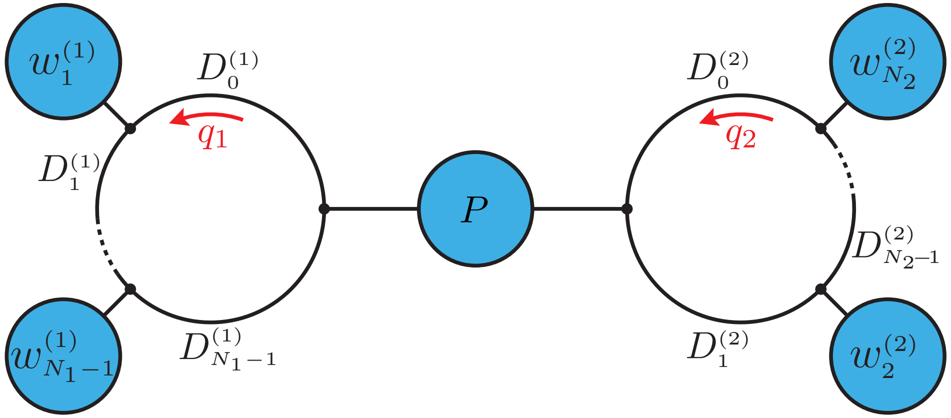
- Implementation of two-loop UV and rational counterterms
- Automation of all one-loop and two-loop ingredients in a single interface
- Tensor integral reduction and evaluation (in-house framework or external tool or mixture thereof)

Backup

Reducible two-loop diagrams

Amplitude of reducible diagram Γ_{red} (1-particle-reducible after amputation of external subtrees):

$$\mathcal{M}_{2,\Gamma_{red}} =$$



$$= C_{2,\Gamma_{red}} P_{\alpha_1\alpha_2} \prod_{i=1}^{2} \int d^D q_i \frac{\left[\mathcal{N}^{(i)}(q_i)\right]^{\alpha_i}}{\mathcal{D}^{(i)}(q_i)}$$

Two factorised one-loop diagrams connected by a tree-like bridge P

$\Rightarrow$ **Fully implemented**