PAUL SCHERRER INSTITUT PSI

FNSNF
Fonds national suisse
Schweizerischer Nationalfonds
Fondo nazionale svizzero
Swiss National Science Foundation

OpenLoops @ 2 loops

M. F. Zoller

based on

JHEP 05 (2022) 161 [arXiv:2201.11615]

in collaboration with

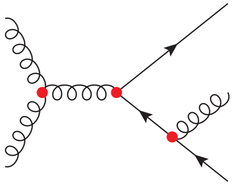
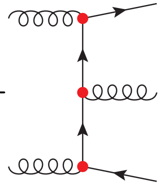
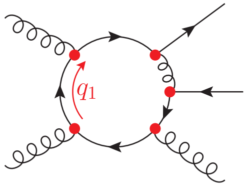
S. Pozzorini and N. Schär

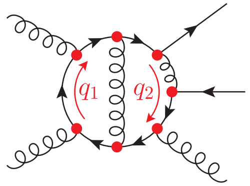
CERN workshop "Precision calculations for future e^+e^- colliders: targets and tools", 13th May 2022

Scattering amplitudes in perturbation theory

Hard scattering amplitudes for Monte Carlo simulations are computed in perturbation theory from matrix elements

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

with $\bar{\mathcal{M}}_0 =$  $+$  $+\ldots,\quad \bar{\mathcal{M}}_1 =$ 

$+\ldots,\quad \bar{\mathcal{M}}_2 =$ 

$+\ldots$

Partonic cross sections

$$\hat{\sigma} \;=\; \underbrace{\int \mathrm{d}\Phi_N}_{\text{N-particle phase space integration, flux factor}} \;\mathcal{W} \quad + \sum_X \;\underbrace{\int \mathrm{d}\Phi_{N+X}\; \mathcal{W}^{(X)}}_{\text{contribution with X extra unresolved particles}}$$

computed from colour- and helicity-summed **scattering probability density**

$$\mathcal{W} \;=\; \underbrace{\bar{\sum_{h,\mathrm{col}}}}_{\text{colour and helicity sum with average and symmetry factor}} \left|\mathbf{R}\bar{\mathcal{M}}\right|^2 \;=\; \bar{\sum_{h,\mathrm{col}}}\left\{\underbrace{|\bar{\mathcal{M}}_0|^2}_{\textbf{LO}} + \underbrace{2\,\mathrm{Re}\Big[\bar{\mathcal{M}}_0^*\,\mathbf{R}\bar{\mathcal{M}}_1\Big]}_{\textbf{NLO virtual}} + \underbrace{|\mathbf{R}\bar{\mathcal{M}}_1|^2 + 2\,\mathrm{Re}\Big[\bar{\mathcal{M}}_0^*\,\mathbf{R}\bar{\mathcal{M}}_2\Big]}_{\textbf{NNLO virtual-virtual}} + \;\ldots\right\}$$

with UV divergences subtracted by the renormalisation procedure $\mathbf{R}\,\bar{\mathcal{M}} = \bar{\mathcal{M}}_0 + \mathbf{R}\,\bar{\mathcal{M}}_1 + \mathbf{R}\,\bar{\mathcal{M}}_2 + \ldots$

OpenLoops

OpenLoops [Buccioni, Lang, Lindert, Maierhöfer, Pozzorini, Zhang, M.Z.] is a fully automated numerical tool for the computation of **scattering probability densities** from tree and one-loop amplitudes

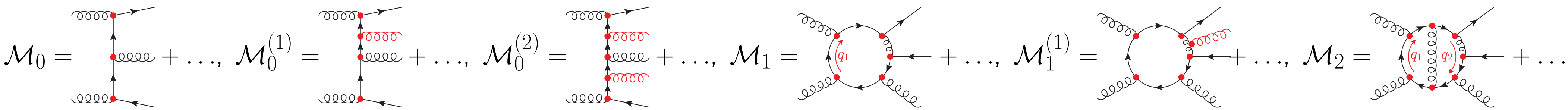
$$\mathcal{W}_{00} = \bar{\sum_{h,\mathrm{col}}} |\bar{\mathcal{M}}_0|^2, \qquad \mathcal{W}_{01} = \bar{\sum_{h,\mathrm{col}}} 2\,\mathrm{Re}\Big[\bar{\mathcal{M}}_0^* \mathbf{R} \bar{\mathcal{M}}_1\Big], \qquad \mathcal{W}_{11} = \bar{\sum_{h,\mathrm{col}}} |\mathbf{R}\bar{\mathcal{M}}_1|^2$$

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

- Full NLO QCD and NLO EW corrections available
- Excellent CPU performance and numerical stability ← Crucial for real-virtual contributions

Real-emission contributions up to NNLO available in OpenLoops

$$\mathcal{W}^{(1)}_{00} = \bar{\sum_{h,\mathrm{col}}} |\bar{\mathcal{M}}^{(1)}_0|^2, \qquad \mathcal{W}^{(1)}_{01} = \bar{\sum_{h,\mathrm{col}}} 2\,\mathrm{Re}\Big[\bar{\mathcal{M}}_0^{(1)*} \mathbf{R} \bar{\mathcal{M}}^{(1)}_1\Big], \qquad \mathcal{W}^{(2)}_{00} = \bar{\sum_{h,\mathrm{col}}} |\bar{\mathcal{M}}^{(2)}_0|^2$$



$$\mathcal{W}_{02} = \bar{\sum_{h,\mathrm{col}}} 2\,\mathrm{Re}\Big[\bar{\mathcal{M}}_0^* \mathbf{R} \bar{\mathcal{M}}_2\Big]$$

required for NNLO, but no fully automated tool available

⇒ OpenLoops **for two-loop amplitudes highly desirable**

Outline

I. One-loop amplitudes

$\rightarrow$ OpenLoops algorithm for tree and one-loop amplitudes

II. Two-loop amplitudes

$\rightarrow$ New algorithm for two-loop integrands

$\rightarrow$ Numerical stability and CPU efficiency

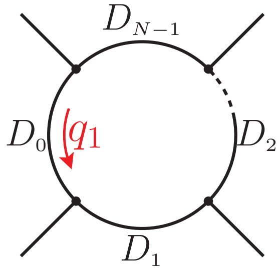
III. OpenLoops features @ 1 loop and 2 loop

V. Summary and Outlook

I. One-loop amplitudes

One-loop diagram Γ in $D = 4 - 2\varepsilon$ dimensions

$$\bar{\mathcal{M}}_{1,\Gamma} = \underbrace{C_{1,\Gamma}}_{\text{colour factor}} \int \mathrm{d}\bar{q}_1 \frac{\bar{\mathcal{N}}(\bar{q}_1)}{\mathcal{D}(\bar{q}_1)} =$$



$$\mathcal{D}(\bar{q}_1) = \prod_{i=0}^{N-1} D_k(\bar{q}_1),$$
$$D_k(\bar{q}_1) = (\bar{q}_1 + p_k)^2 - m_k^2,$$
$$\int \mathrm{d}\bar{q}_1 = \mu^{2\varepsilon} \int \frac{\mathrm{d}^D \bar{q}_1}{(2\pi)^D}$$

Numerical tools, such as OpenLoops [Buccioni et al], Recola [Actis et al], MadLoop [Hirschi et al], construct the numerator in 4 dimensions (D-dim quantities with bar, 4-dim without)

$$\underbrace{\mathcal{N}(q_1)}_{4-\text{dim}} = \underbrace{\bar{\mathcal{N}}(\bar{q}_1)}_{D-\text{dim}} \Bigg|_{\substack{\bar{q}_i \to q_i, \\ \bar{\gamma}^{\bar{\mu}} \to \gamma^{\mu}, \\ \bar{g}^{\bar{\mu}\bar{\nu}} \to g^{\mu\nu}}} \Rightarrow$$

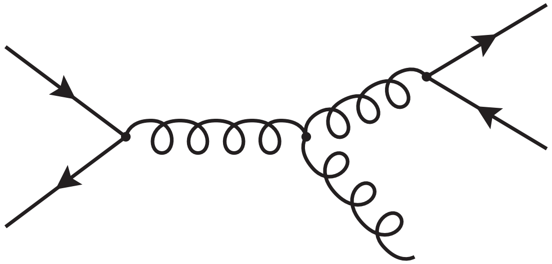
$$\mathcal{M}_{1,\Gamma} = C_{1,\Gamma} \sum_{r=0}^{R_1} \underbrace{\mathcal{N}_{\mu_1 \cdots \mu_r}}_{\text{tensor coefficient}} \underbrace{\int \mathrm{d}\bar{q}_1 \frac{q_1^{\mu_1} \cdots q_1^{\mu_{r_1}}}{\mathcal{D}(\bar{q}_1)}}_{\text{tensor integral}}$$

Steps of the calculation

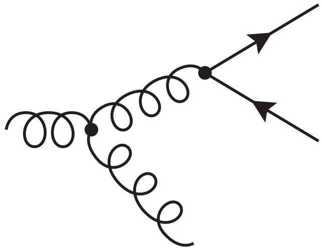
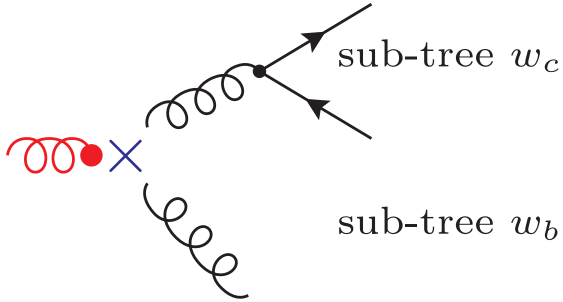
- Construction of tensor coefficients ← OpenLoops algorithm [van Hameren; Cascioli, Maierhöfer, Pozzorini; Buccioni, Lang, Lindert, Pozzorini, Zhang, M.Z.]
- Reduction of tensor integrals and evaluation of master integrals ← On-the-fly reduction [Buccioni, Pozzorini, M.Z.] and Collier [Denner, Dittmaier, Hofer], OneLoop [van Hameren]
- Restoration of ε-dim numerator parts $\tilde{\mathcal{N}}(\bar{q}_1) = \bar{\mathcal{N}}(\bar{q}_1) - \mathcal{N}(q_1)$ ← Rational counterterms [Ossola, Papadopoulos, Pittau]

The OpenLoops algorithm at tree level

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

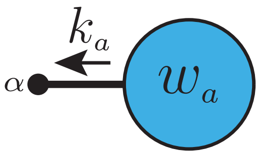
Example: $\mathcal{M}_0 =$  $+ \ldots$

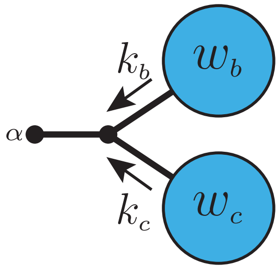
Numerical recursion step:

$w_a^\alpha =$  $=$ 

$$= \underbrace{\frac{X^\alpha_{\beta\gamma}(k_b, k_c)}{k_a^2 - m_a^2}} \; w_b^\beta \; w_c^\gamma$$

universal building block from Feynman rules

Generic depiction: 

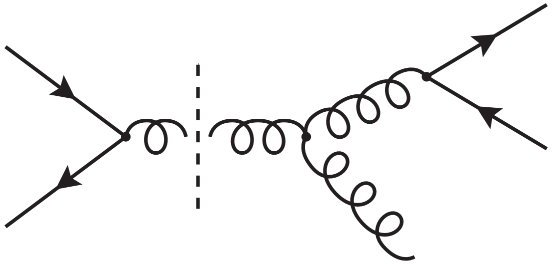
$=$ 

(k_i external momenta)

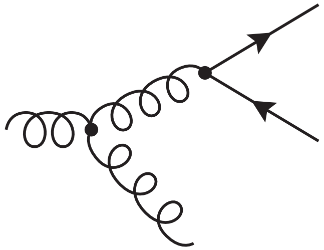
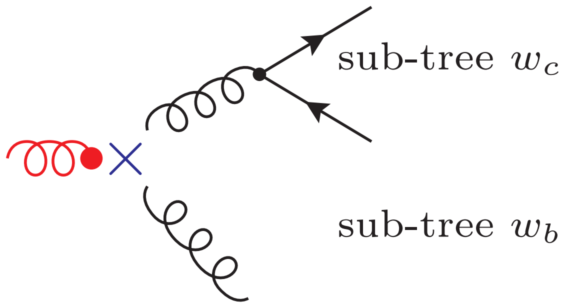
Highly efficient: Subtrees constructed only once for multiple tree and loop diagrams

The OpenLoops algorithm at tree level

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

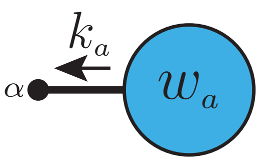
Example: $\mathcal{M}_0 =$  $+ \ldots$ → split into subtrees

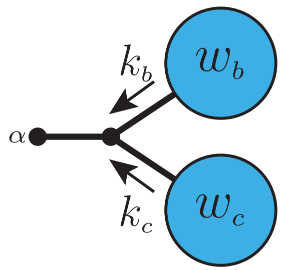
Numerical recursion step:

$w_a^\alpha =$  $=$ 

$= \dfrac{X^\alpha_{\beta\gamma}(k_b, k_c)}{k_a^2 - m_a^2} \; w_b^\beta \; w_c^\gamma$

universal building block from Feynman rules

Generic depiction: 

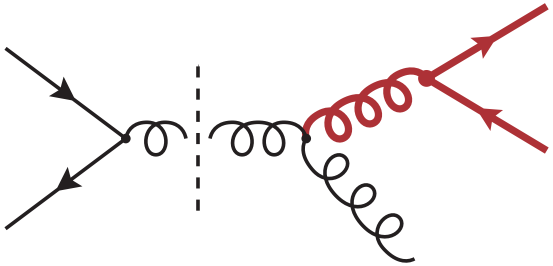
$=$ 

(k_i external momenta)

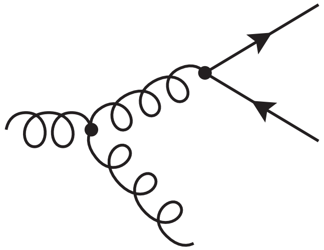
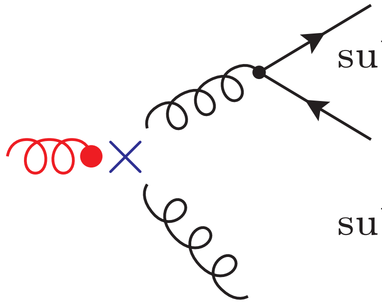
Highly efficient: Subtrees constructed only once for multiple tree and loop diagrams

The OpenLoops algorithm at tree level

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

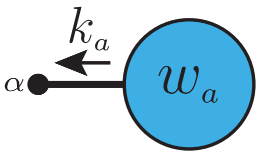
Example: $\mathcal{M}_0 =$  $+ \ldots$ → connect subtrees

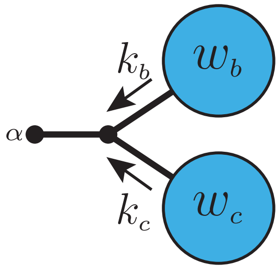
Numerical recursion step:

$w_a^\alpha =$  $=$ 

$= \underbrace{\dfrac{X^\alpha_{\beta\gamma}(k_b, k_c)}{k_a^2 - m_a^2}}$ $w_b^\beta \; w_c^\gamma$

universal building block from Feynman rules

Generic depiction: 

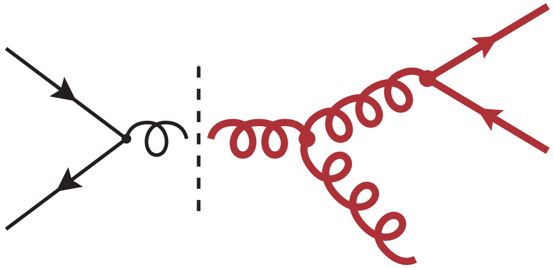
$=$ 

(k_i external momenta)

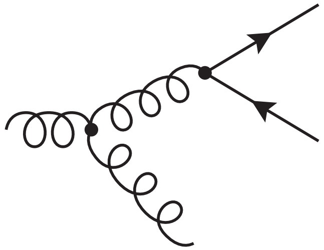
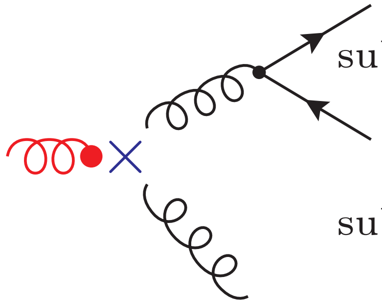
Highly efficient: Subtrees constructed only once for multiple tree and loop diagrams

The OpenLoops algorithm at tree level

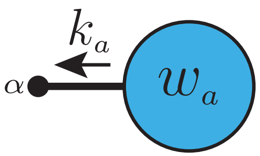
Tree-level amplitudes constructed recursively from subtrees (starting from external lines)

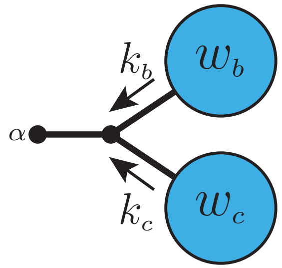
Example: $\mathcal{M}_0 =$  $+ \ldots$ $\rightarrow$ connect subtrees

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: 

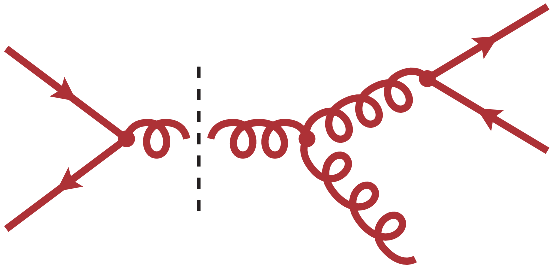
$=$ 

(k_i external momenta)

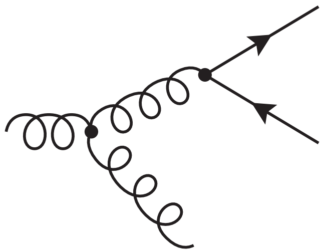
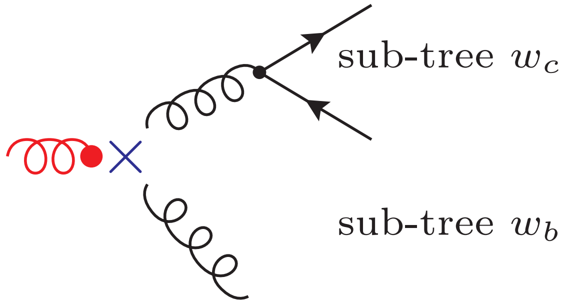
Highly efficient: Subtrees constructed only once for multiple tree and loop diagrams

The OpenLoops algorithm at tree level

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

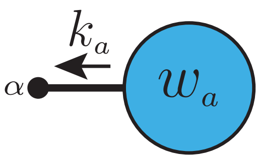
Example: $\mathcal{M}_0 =$  $+ \ldots$ $\rightarrow$ connect subtrees

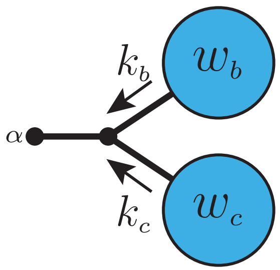
Numerical recursion step:

$w_a^\alpha =$  $=$ 

$$= \underbrace{\frac{X^\alpha_{\beta\gamma}(k_b, k_c)}{k_a^2 - m_a^2}} \; w_b^\beta \; w_c^\gamma$$

universal building block from Feynman rules

Generic depiction: 

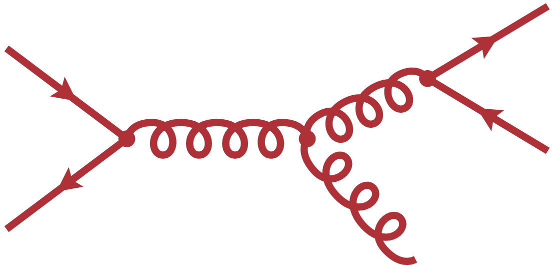
$=$ 

(k_i external momenta)

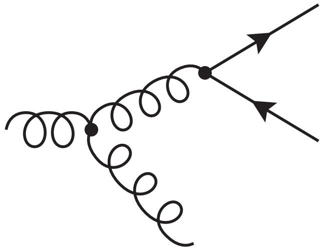
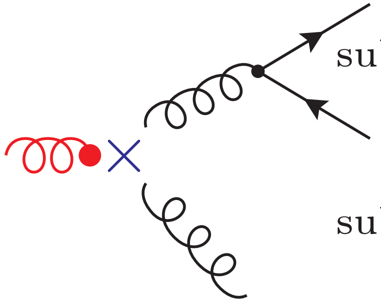
Highly efficient: Subtrees constructed only once for multiple tree and loop diagrams

The OpenLoops algorithm at tree level

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

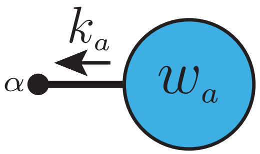
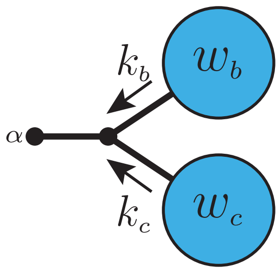
Example: $\mathcal{M}_0 =$  $+ \ldots$ $\rightarrow$ connect diagram

Numerical recursion step:

$w_a^\alpha =$  $=$ 

$$= \underbrace{\frac{X^\alpha_{\beta\gamma}(k_b, k_c)}{k_a^2 - m_a^2}} \; w_b^\beta \; w_c^\gamma$$

universal building block from Feynman rules

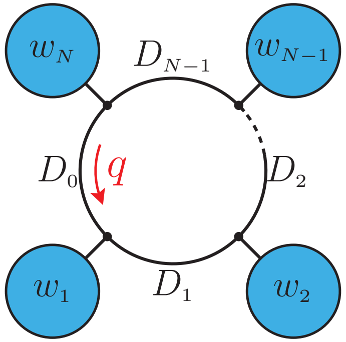
Generic depiction:  $=$  (k_i external momenta)

Highly efficient: Subtrees constructed only once for multiple tree and loop diagrams

The OpenLoops algorithm at one loop

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

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

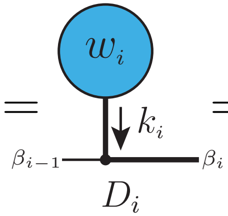


$$= \mathcal{C}_{1,\Gamma} \int \mathrm{d}^D q \, \frac{S_1(q)\cdots S_N(q)}{D_0\cdots D_{N-1}}$$

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

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

$S_i(q) =$



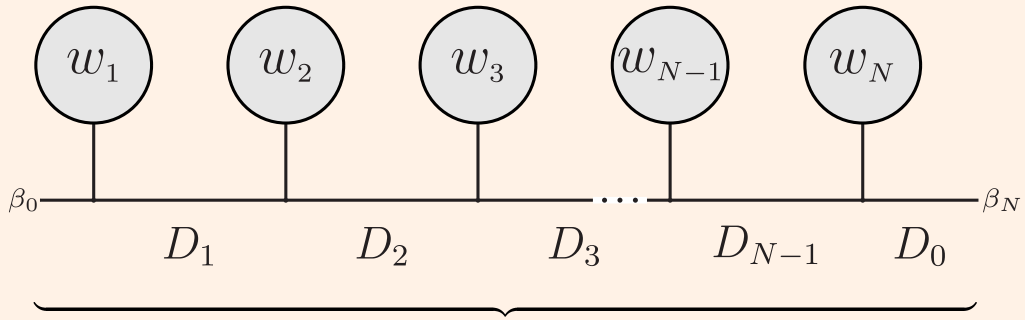
$$= \{Y^i_\sigma(k_i, p_i) + Z^i_{\nu;\sigma}\, q^\nu\}\, w_i^\sigma$$

Universal building block $\times$ subtree(s)

Cut-open loop at D_0

$\mathcal{N}_0(q) = \mathbb{1}$

$=$

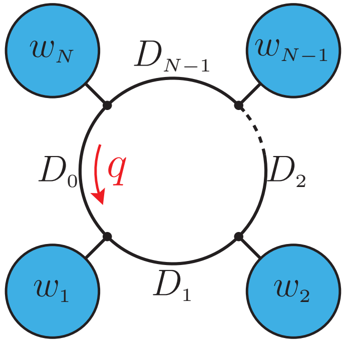


Open loop is a matrix with two Lorentz/spinor indices β_0, β_N

The OpenLoops algorithm at one loop

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

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

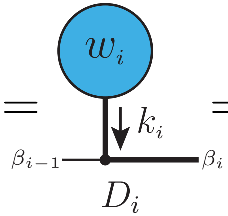


$$= \mathcal{C}_{1,\Gamma} \int \mathrm{d}^D q \, \frac{S_1(q) \cdots S_N(q)}{D_0 \cdots D_{N-1}}$$

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

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

$$S_i(q) =$$

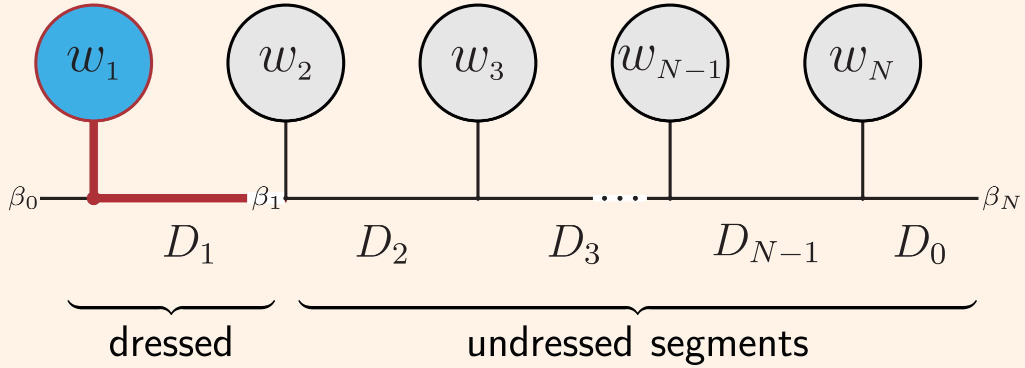


$$= \{Y^i_\sigma(k_i, p_i) + Z^i_{\nu;\sigma} \, q^\nu\} \, w_i^\sigma$$

Universal building block $\times$ subtree(s)

Dress chain of segments (open loop) recursively

$$\mathcal{N}_1(q) \;=\; \mathcal{N}_0(q) S_1(q) \;=\; S_1(q) \;=$$



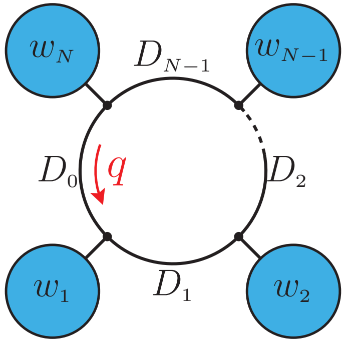
$$= \mathcal{N}^{(1)} + \mathcal{N}^{(1)}_{\mu_1} q^{\mu_1}$$

Recursion steps can increase the rank in q by 1.

The OpenLoops algorithm at one loop

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

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

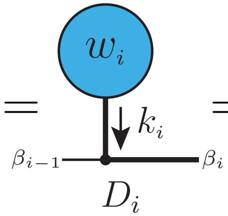


$$= \mathcal{C}_{1,\Gamma} \int \mathrm{d}^D q \, \frac{S_1(q)\cdots S_N(q)}{D_0\cdots D_{N-1}}$$

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

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

$$S_i(q) = $$

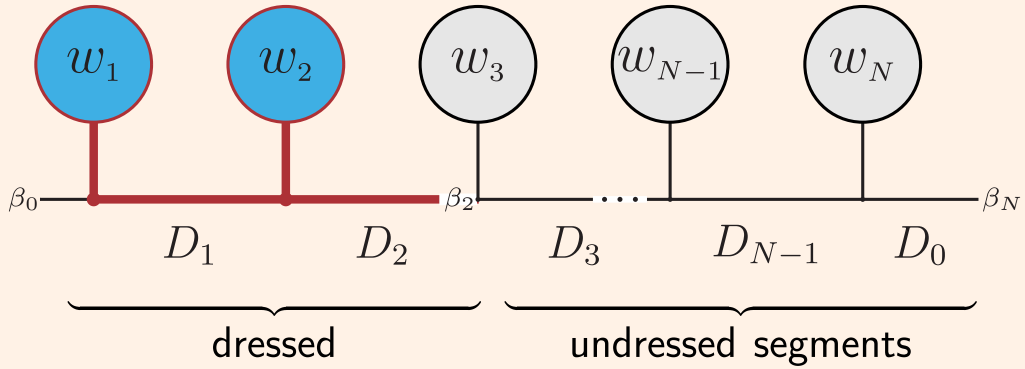


$$= \{Y^i_\sigma(k_i,p_i) + Z^i_{\nu;\sigma}\, q^\nu\}\, w_i^\sigma$$

Universal building block $\times$ subtree(s)

Dress chain of segments (open loop) recursively

$$\mathcal{N}_2(q) = \mathcal{N}_1(q) S_2(q) = \prod_{i=1}^{2} S_i(q) =$$



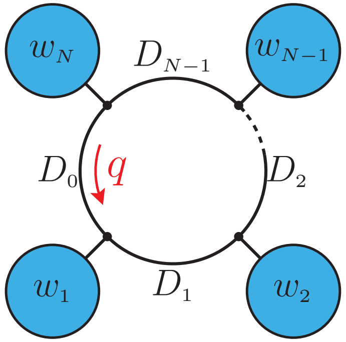
$$= \mathcal{N}^{(2)} + \mathcal{N}^{(2)}_{\mu_1} q^{\mu_1} + \mathcal{N}^{(2)}_{\mu_1\mu_2} q^{\mu_1} q^{\mu_2}$$

Recursion steps are matrix multiplications: $\Big[\mathcal{N}_n(q)\Big]^{\beta_n}_{\beta_0} = \Big[\mathcal{N}_{n-1}(q)\Big]^{\beta_{n-1}}_{\beta_0} \Big[S_n(q)\Big]^{\beta_n}_{\beta_{n-1}}$

The OpenLoops algorithm at one loop

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

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

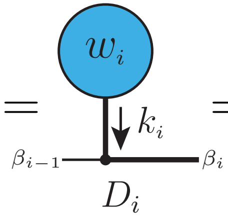


$$= \mathcal{C}_{1,\Gamma} \int \mathrm{d}^D q \, \frac{S_1(q) \cdots S_N(q)}{D_0 \cdots D_{N-1}}$$

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

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

$S_i(q) =$

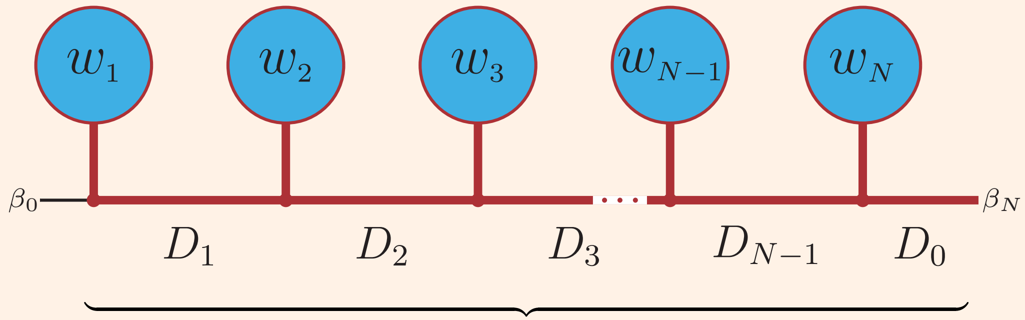


$$= \{ Y^i_\sigma(k_i, p_i) + Z^i_{\nu;\sigma} \, q^\nu \} \, w_i^\sigma$$

Universal building block $\times$ subtree(s)

Dress chain of segments recursively $\rightarrow$ Close loop by contracting β_0 and β_N

$$\mathcal{N}_N(q) = \mathcal{N}_{N-1}(q) S_N(q) = \prod_{i=1}^{N} S_i(q) =$$



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

Recursion steps $\mathcal{N}_n(q) = \mathcal{N}_{n-1}(q) S_n(q)$ at the level of tensor coefficients $\mathcal{N}^{(n)}_{\mu_1 \dots \mu_r}$

Completely general and highly efficient algorithm

Born-loop interference

Scattering probability density from interference of one-loop diagrams Γ with full Born

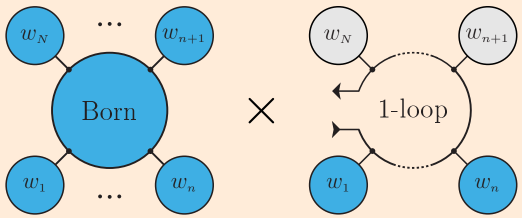
$$\mathcal{W}_{01,\Gamma} = \bar{\sum_{h,\mathrm{col}}} 2\,\mathrm{Re}\left[\bar{\mathcal{M}}_0^* \mathbf{R} \bar{\mathcal{M}}_{1,\Gamma}\right] \quad \Rightarrow \quad \mathcal{W}_{01} = \sum_{\Gamma} \mathcal{W}_{01,\Gamma}$$

Consider colour-helicity summed numerator $\Rightarrow$ nested sums of helicities h_i of individual segments

$$\mathcal{U}(q,0) = \sum_h \underbrace{2\left(\sum_{\mathrm{col}} \mathcal{M}_0^*(h) C_{1,\Gamma}\right)}_{=\mathcal{U}_0(h)} \mathcal{N}(q,h) = \sum_{h_N}\left[\dots\sum_{h_2}\left[\sum_{h_1}\mathcal{U}_0(h)S_1(q,h_1)\right]S_2(q,h_2)\cdots\right]S_n(q,h_N)$$

On-the-fly helicity summation [Buccioni, Pozzorini, M.Z.]

$$\mathcal{U}_n(q,\check{h}_n) = \sum_{h_n}\mathcal{U}_{n-1}(q,\check{h}_{n-1})S_n(q,h_n) = \sum_{h_1\dots h_n}\sum_{\mathrm{col}}$$



$$\mathcal{U}_n(q,\check{h}_n) = \sum_{r=0}^{n}\mathcal{U}^{(n)}_{\mu_1\dots\mu_r}\,q^{\mu_1}\dots q^{\mu_r}$$

depends on helicity $\check{h}_n = \sum_{k=n+1}^{N} h_k$ of undressed segments

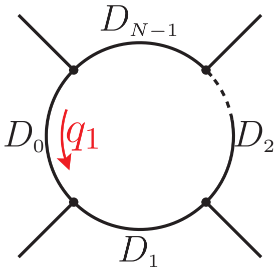
Implemented at the level of tensor integral coefficients $\mathcal{U}^{(n)}_{\mu_1\dots\mu_r}$

Huge gain in CPU efficiency, especially for high-multiplicity processes

One-loop rational terms

Amputated one-loop diagram γ (1PI)

$$\bar{\mathcal{M}}_{1,\gamma} = \underbrace{C_{1,\gamma}}_{\text{colour factor}} \int \mathrm{d}\bar{q}_1 \frac{\mathcal{N}(q_1) + \tilde{\mathcal{N}}(\bar{q}_1)}{\mathcal{D}(\bar{q}_1)} =$$



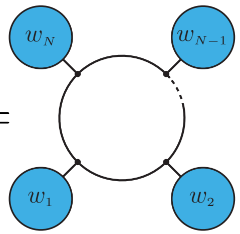
$$\Rightarrow \quad \delta\mathcal{R}_{1,\gamma} = C_{1,\gamma} \int \mathrm{d}\bar{q}_1 \frac{\tilde{\mathcal{N}}(\bar{q}_1)}{\mathcal{D}(\bar{q}_1)}$$

The ε-dim numerator parts $\tilde{\mathcal{N}}(\bar{q}_1) = \bar{\mathcal{N}}(\bar{q}_1) - \mathcal{N}(q_1)$ contribute only via interaction with $\frac{1}{\varepsilon}$ UV poles
$\Rightarrow$ Can be restored through **rational counterterm** $\delta\mathcal{R}_{1,\gamma}$ [Ossola, Papadopoulos, Pittau]

$$\Rightarrow \quad \underbrace{\mathbf{R}\,\bar{\mathcal{M}}_{1,\gamma}}_{D-\text{dim, renormalised}} = \underbrace{\mathcal{M}_{1,\gamma}}_{4-\text{dim numerator}} + \underbrace{\delta Z_{1,\gamma} + \delta\mathcal{R}_{1,\gamma}}_{\text{UV and rational counterterm}}$$

Generic one-loop diagram Γ factorises into 1PI subdiagram γ and external subtrees w_i (4-dim):

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



$$= \left[\bar{\mathcal{M}}_{1,\gamma}\right]^{\sigma_1\ldots\sigma_N} \prod_{i=1}^{N} \left[w_i\right]_{\sigma_i} \quad \Rightarrow$$

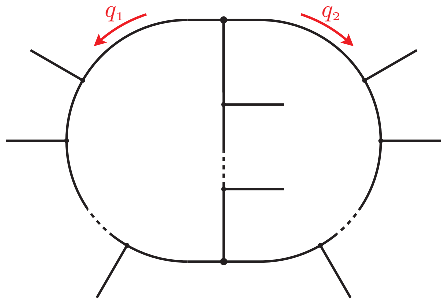
$$\mathbf{R}\,\bar{\mathcal{M}}_{1,\Gamma} = \mathcal{M}_{1,\Gamma} + \left(\delta Z_{1,\gamma} + \delta\mathcal{R}_{1,\gamma}\right) \underbrace{\prod_{i=1}^{N} w_i}_{\text{tree diagram}}$$

Finite set of process-independent rational terms in renormalisable models
computed from UV divergent vertex functions

II. Two-loop amplitudes

Two-loop diagram Γ in $D = 4 - 2\varepsilon$ dimensions

$$\bar{\mathcal{M}}_{2,\Gamma} = \underbrace{C_{2,\Gamma}}_{\text{colour factor}} \int \mathrm{d}\bar{q}_1 \int \mathrm{d}\bar{q}_2 \frac{\bar{\mathcal{N}}(\bar{q}_1,\bar{q}_2)}{\mathcal{D}(\bar{q}_1,\bar{q}_2)} =$$



with the D-dim denominator

$$\mathcal{D}(\bar{q}_1,\bar{q}_2) = \prod_i \prod_k D_k^{(i)}(\bar{q}_i)$$

$$D_k^{(i)}(\bar{q}_i) = (\bar{q}_i + p_{ik})^2 - m_{ik}^2$$

Numerical construction requires $\underbrace{\mathcal{N}(q_1,q_2)}_{4-\text{dim}} = \underbrace{\bar{\mathcal{N}}(\bar{q}_1,\bar{q}_2)}_{D-\text{dim}}\Big|_{\bar{q}_i \to q_i,\ \bar{\gamma}^{\bar{\mu}} \to \gamma^{\mu},\ \bar{g}^{\bar{\mu}\bar{\nu}} \to g^{\mu\nu}}$

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

Steps of the calculation

- Construction of tensor coefficients $\leftarrow$ **Now fully implemented**
- Reduction and evaluation of tensor integrals $\leftarrow$ Not yet automated
- Restoration of $\tilde{\mathcal{N}}(\bar{q}_1,\bar{q}_2) = \bar{\mathcal{N}}(\bar{q}_1,\bar{q}_2) - \mathcal{N}(q_1,q_2)$ $\leftarrow$ **Two-loop rational terms** [Lang, Pozzorini, Zhang, M.Z.]

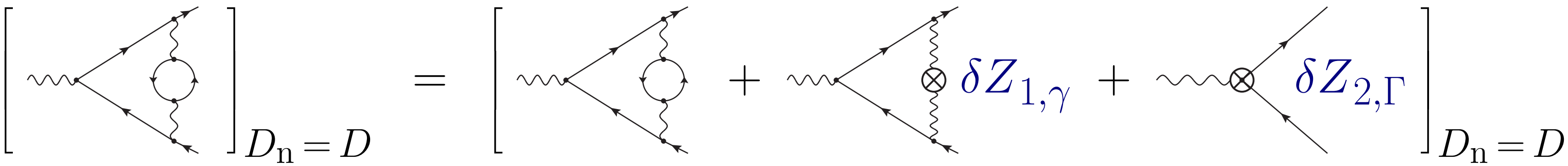
Two-loop rational terms

Start from renormalisation procedure for (1PI) diagram Γ in D-dim

$$\mathbf{R}\,\bar{\mathcal{M}}_{2,\Gamma} \;=\; \bar{\mathcal{M}}_{2,\Gamma} \;+\; \sum_{\gamma} \underbrace{\delta Z_{1,\gamma}\cdot\bar{\mathcal{M}}_{1,\Gamma/\gamma}}_{\text{subtract subdivergences}} \;+\; \underbrace{\delta Z_{2,\Gamma}}_{\text{subtract remaining } \textbf{local} \text{ divergence}}$$

Sum over all subdiagrams γ of Γ. Numerator dimension $D_{\mathrm{n}} = D$.

Example: **R**



Extension from single diagrams to full vertex functions Γ due to linearity of **R**

Goal: Computation from amplitudes with numerator dimension $D_{\mathrm{n}} = 4$

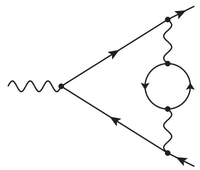
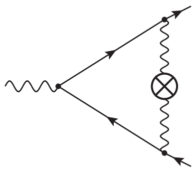
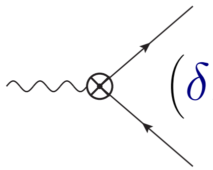
- Split numerator $\bar{\mathcal{N}}(\bar{q}_1,\bar{q}_2) = \mathcal{N}(q_1,q_2) + \tilde{\mathcal{N}}(\bar{q}_1,\bar{q}_2)$
- Compute amplitudes on lhs with $\mathcal{N}(q_1,q_2) = \bar{\mathcal{N}}(\bar{q}_1,\bar{q}_2)\Big|_{\bar{g}^{\bar{\mu}\bar{\nu}}\to g^{\mu\nu},\ \bar{\gamma}^{\bar{\mu}}\to\gamma^{\mu},\ \bar{q}_i\to q_i}$
- Restore $\tilde{\mathcal{N}}$-terms (from subdiagrams and a remaining global one) through additional counterterms

Two-loop rational terms

Renormalised D-dim amplitudes from amplitudes with 4-dim numerator [Pozzorini, Zhang, M.Z.]

$$\mathbf{R}\,\bar{\mathcal{M}}_{2,\Gamma} = \mathcal{M}_{2,\Gamma} + \sum_{\gamma}\Big(\underbrace{\delta Z_{1,\gamma} + \delta\tilde{Z}_{1,\gamma}}_{\text{subtract subdivergences}} + \underbrace{\delta\mathcal{R}_{1,\gamma}}_{\text{restore } \tilde{\mathcal{N}}\text{-terms from subdiagrams}}\Big)\cdot\mathcal{M}_{1,\Gamma/\gamma} + \Big(\underbrace{\delta Z_{2,\Gamma}}_{\text{subtract remaining local divergence}} + \underbrace{\delta\mathcal{R}_{2,\Gamma}}_{\text{restore remaining } \tilde{\mathcal{N}}\text{-term}}\Big)$$

Example:

$\mathbf{R}\,\bar{\mathcal{M}}_{2,\Gamma} = \Big[$  $+$  $(\delta Z_{1,\gamma} + \delta\tilde{Z}_{1,\gamma} + \delta\mathcal{R}_{1,\gamma})$ $+$  $(\delta Z_{2,\Gamma} + \delta\mathcal{R}_{2,\Gamma})\Big]_{\text{4-dim numerators}}$

- Divergences from subdiagrams γ and remaining global one subtracted by usual UV counterterms $\delta Z_{1,\gamma}, \delta Z_{2,\Gamma}$. Additional UV counterterm $\delta\tilde{Z}_{1,\gamma} \propto \frac{\tilde{q}_1^{\,2}}{\varepsilon}$ for subdiagrams with mass dimension 2.
- $\delta\mathcal{R}_{2,\Gamma}$ is a **two-loop rational term** stemming from the interplay of $\tilde{N}$ with UV poles
- External subtrees factorise and do not generate rational terms (see one-loop case)
- Extension from single diagrams to full vertex functions due to linearity of **R**

$\Rightarrow$ **Finite set of process-independent rational terms for UV divergent vertex functions**

Two-loop rational terms

Renormalised D-dim amplitudes can be computed from amplitudes with 4-dim numerators and a **finite set of universal UV and rational counterterms** inserted lower-loop amplitudes

$$\mathbf{R}\,\bar{\mathcal{M}}_{2,\Gamma} \;=\; \mathcal{M}_{2,\Gamma} + \sum_{\gamma}\left(\delta Z_{1,\gamma} + \delta\tilde{Z}_{1,\gamma} + \delta\mathcal{R}_{1,\gamma}\right)\cdot\mathcal{M}_{1,\Gamma/\gamma} + \left(\delta Z_{2,\Gamma} + \delta\mathcal{R}_{2,\Gamma}\right)$$

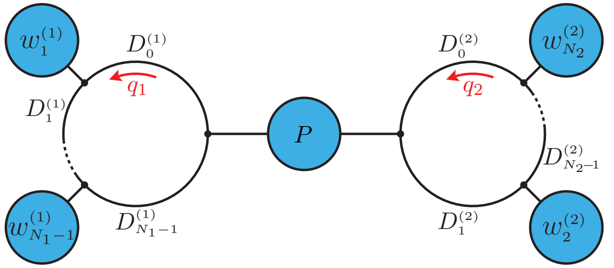
Status of two-loop rational terms

- **General method for the computation** of rational counterterms of UV origin from simple tadpole integrals in any renormalisable model [Pozzorini, Zhang, M.Z.,2020]
- **Complete renormalisation scheme dependence** [Lang, Pozzorini, Zhang, M.Z.,2020]
- **Rational Terms for Spontaneously Broken Theories** [Lang, Pozzorini, Zhang, M.Z.,2021]
- **Full set of two-loop rational terms** for **QED and QCD corrections to the SM** [Pozzorini, Zhang, M.Z.,2020] [Lang, Pozzorini, Zhang, M.Z.,2020] [Lang, Pozzorini, Zhang, M.Z.,2021]
- **Rational terms of IR origin** currently under investigation

Reducible two-loop diagrams

Reducible diagram Γ factorises into one-loop diagrams and a tree-like bridge P (or quartic vertex)

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

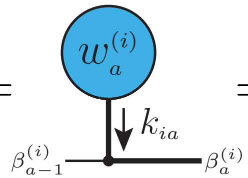


$$= C_{2,\Gamma}\, P_{\alpha_1\alpha_2} \prod_{i=1}^{2} \int \mathrm{d}\bar{q}_i \, \frac{\left[\mathcal{N}^{(i)}(q_i)\right]^{\alpha_i}}{\mathcal{D}^{(i)}(\bar{q}_i)}$$

with $\mathcal{D}^{(i)}(\bar{q}_i) = D_0^{(i)}(\bar{q}_i) \cdots D_{N_i-1}^{(i)}(\bar{q}_i)\,, \qquad D_a^{(i)}(\bar{q}_i) \;=\; (\bar{q}_i + p_{ia})^2 - m_{ia}^2$

Loop numerators factorise into segments

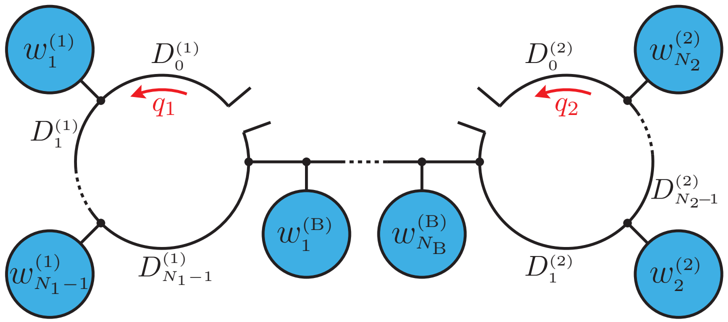
$$S_a^{(i)}(q_i, h_a^{(i)}) = $$



$$= \underbrace{\left\{ Y_\sigma^a(k_{ia}, p_{ia}) + Z_{\nu;\sigma}^i \, q_i^\nu \right\}}_{\text{Feynman rule of loop vertex and propagator}} \quad \underbrace{\left[w_a^{(i)}(h_a^{(i)}) \right]^\sigma}_{\text{external subtree with helicity configuration } h_a^{(i)}}$$

- Cut-open both loops and dress first one
- Close and integrate first loop, attach bridge
- Use first loop + bridge as “subtree” for second loop

$\Rightarrow$ **Extension of the tree and one-loop algorithm**

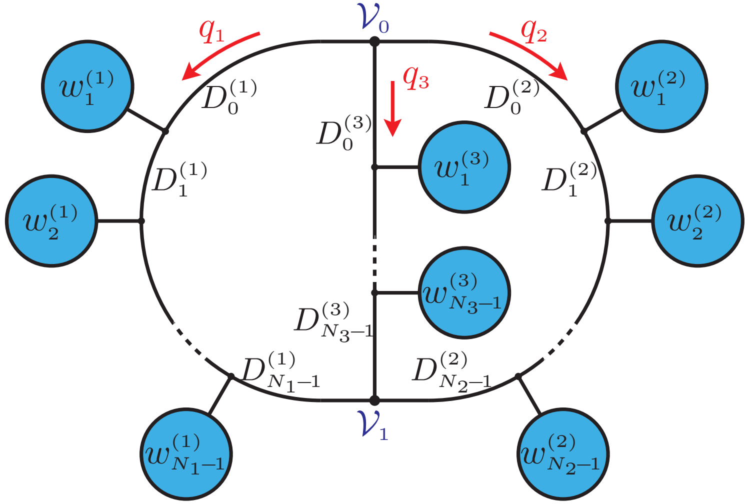


Fully implemented for QED and QCD corrections to the SM

New algorithm to construct two-loop tensor coefficients

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

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



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

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)

General structure of a recursive two-loop algorithm

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

$$= \sum_h \mathcal{U}_0(h) \Big\{ \prod_{i=1}^{3} \underbrace{\Big[\prod_{k=0}^{N_i-1} S_k^{(i)}(q_i, h_k^{(i)}) \Big]_{\beta_0^{(i)}}^{\beta_{N_i}^{(i)}}}_{\text{chain } \mathcal{N}^{(i)}} \Big\} \underbrace{\Big[\mathcal{V}_0(q_1, q_2, h_0^{(V)}) \Big]^{\beta_0^{(1)}\beta_0^{(2)}\beta_0^{(3)}} \Big[\mathcal{V}_1(q_1, q_2, h_1^{(V)}) \Big]_{\beta_{N_1}^{(1)}\beta_{N_2}^{(2)}\beta_{N_3}^{(3)}}}_{\text{connecting vertices (quartic vertices with external subtrees } w_a^{(V)})}$$

with Born–colour factor $\mathcal{U}_0(h) = 2\Big(\sum_{\text{col}} \mathcal{M}_0^*(h)\, C_{2,\Gamma} \Big)$

Algorithm with recursion steps $\boxed{\hat{\mathcal{U}}_n = \hat{\mathcal{U}}_{n-1} \cdot \mathcal{K}_n}$ $= \sum_{r=0}^{R_1} \sum_{s=0}^{R_2} \hat{\mathcal{U}}^{(n)}_{\mu_1\cdots\mu_r\nu_1\cdots\nu_s} q_1^{\mu_1} \ldots q_1^{\mu_r} q_2^{\nu_1} \ldots q_2^{\nu_s}$

with partially dressed numerators $\hat{\mathcal{U}}_n$ and building blocks $\mathcal{K}_n \in \Big\{ \mathcal{U}_0, S_k^{(i)}, \mathcal{V}_j, \mathcal{N}^{(i)} \Big\}$.

- Each step increases the rank in a q_i by 0 or 1
- Segment $S_k^{(i)}$, $\mathcal{V}_j$ depend on helicities of external subtrees

  $\Rightarrow$ global helicity $\boxed{h = \sum_{i=1}^{3} \sum_{k=0}^{N_i-1} h_k^{(i)} + h_0^{(V)} + h_1^{(V)}}$
- High complexity in steps connecting $\mathcal{V}_j$ due to dependence on q_1, q_2 and three open Lorentz/spinor indices $\beta_k^{(i)}$
- Number of tensor coefficients grows exponentially with ranks R_1, R_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

General structure of a recursive two-loop algorithm

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

$$= \sum_h \mathcal{U}_0(h) \Big\{ \underbrace{\prod_{i=1}^{3} \Big[\prod_{k=0}^{N_i-1} S_k^{(i)}(q_i, h_k^{(i)}) \Big]_{\beta_0^{(i)}}^{\beta_{N_i}^{(i)}}}_{\text{chain } \mathcal{N}^{(i)}} \Big\} \underbrace{\Big[\mathcal{V}_0(q_1, q_2, h_0^{(V)}) \Big]^{\beta_0^{(1)}\beta_0^{(2)}\beta_0^{(3)}} \Big[\mathcal{V}_1(q_1, q_2, h_1^{(V)}) \Big]_{\beta_{N_1}^{(1)}\beta_{N_2}^{(2)}\beta_{N_3}^{(3)}}}_{\text{connecting vertices (quartic vertices with external subtrees } w_a^{(V)})}$$

with Born–colour factor $\mathcal{U}_0(h) = 2\Big(\sum_{\text{col}} \mathcal{M}_0^*(h)\, C_{2,\Gamma} \Big)$

Algorithm with recursion steps $\boxed{\hat{\mathcal{U}}_n = \hat{\mathcal{U}}_{n-1} \cdot \mathcal{K}_n}$ $= \sum_{r=0}^{R_1} \sum_{s=0}^{R_2} \hat{\mathcal{U}}^{(n)}_{\mu_1 \cdots \mu_r \nu_1 \cdots \nu_s} q_1^{\mu_1} \dots q_1^{\mu_r} q_2^{\nu_1} \dots q_2^{\nu_s}$

with partially dressed numerators $\hat{\mathcal{U}}_n$ and building blocks $\mathcal{K}_n \in \Big\{ \mathcal{U}_0, S_k^{(i)}, \mathcal{V}_j, \mathcal{N}^{(i)} \Big\}$.

CPU cost of n-th step $\sim$ number of (#) multiplications $\rightarrow$ depends on type of $\mathcal{K}_n$ and

\# components of $\hat{\mathcal{U}}_n$ = (# tensor components in q_1, q_2) $\times$ (# active helicities) $\times\ 4^{(\#\text{ open indices } \beta_a^{(i)})}$

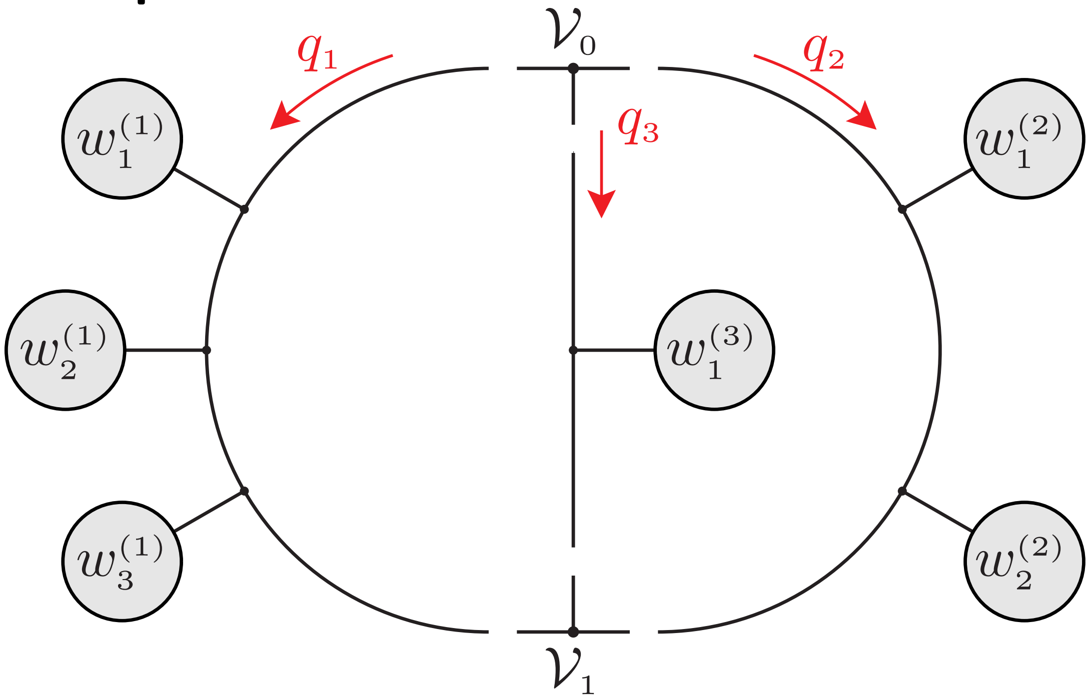
$\Rightarrow$ **Most efficient algorithm found through cost simulation**

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

Two-loop algorithm for irreducible diagrams

- 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:



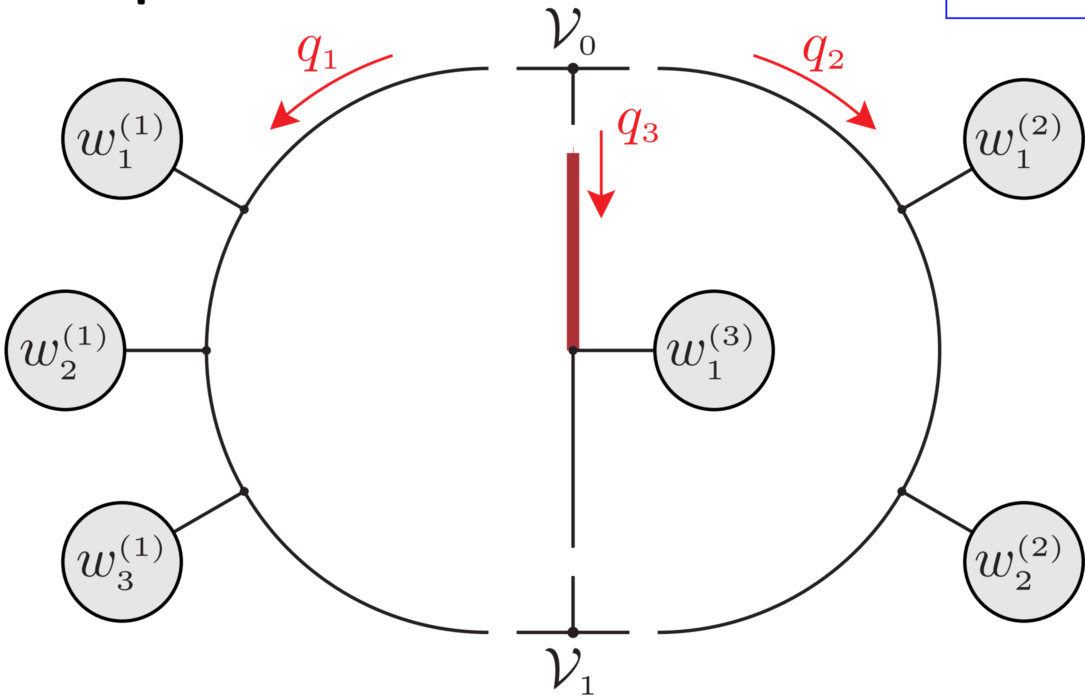
Order of chains and of two-loop vertices $\mathcal{V}_0, \mathcal{V}_1$ has major impact on efficiency

Two-loop algorithm for irreducible diagrams

- 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:

$n = 0$



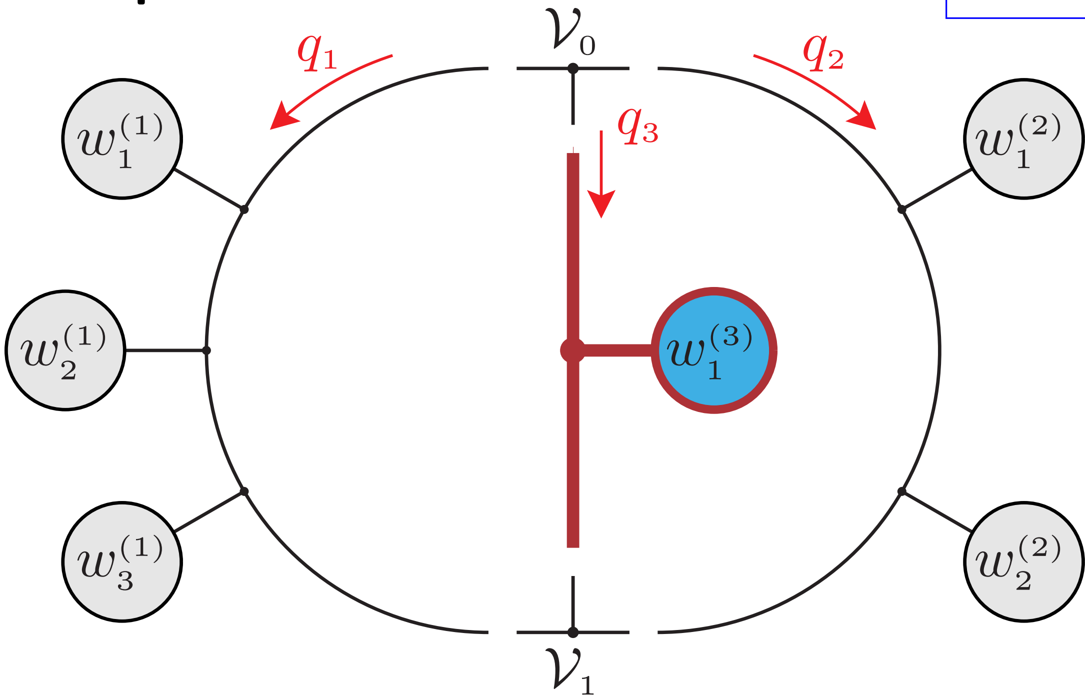
$$\mathcal{N}_n^{(3)}(q_3, \hat{h}_n^{(3)}) = \mathcal{N}_{n-1}^{(3)}(q_3, \hat{h}_{n-1}^{(3)}) \cdot S_n^{(3)}(q_3, h_n^{(3)})$$ with initial condition $\mathcal{N}_{-1}^{(3)} = \mathbb{1}$

Two-loop algorithm for irreducible diagrams

- 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 ⇒ 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

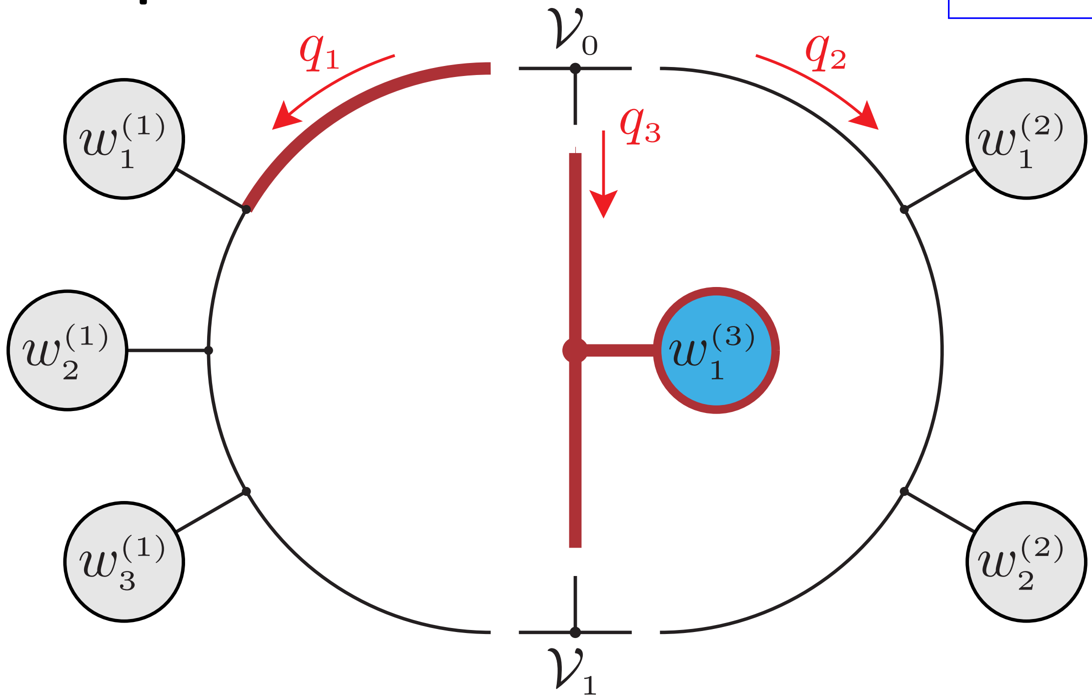
⇒ Only a small number of low-complexity steps for the full process

Two-loop algorithm for irreducible diagrams

- 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:

$n = 0$



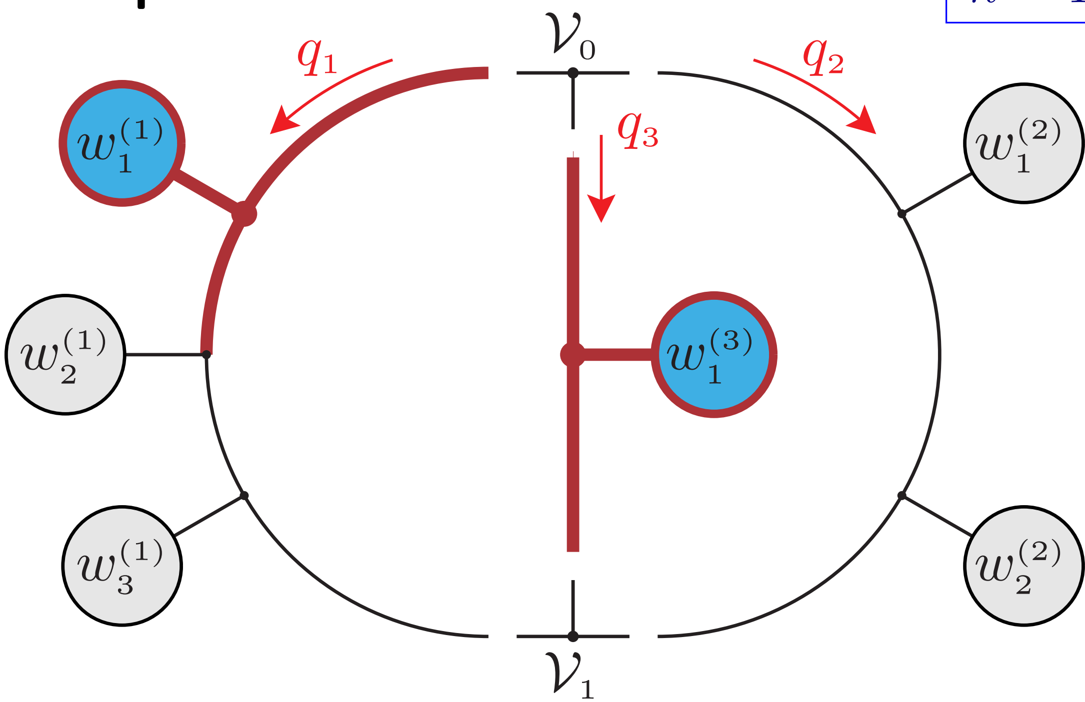
$$\mathcal{U}_n^{(1)}(q_1, \check{h}_n^{(1)}) = \sum_{h_n^{(1)}} \mathcal{U}_{n-1}^{(1)}(q_1, \check{h}_{n-1}^{(1)}) \cdot S_n^{(1)}(q_1, h_n^{(1)}) \qquad \text{with} \quad \mathcal{U}_{-1}^{(1)}(h) = 2\Big(\sum_{\text{col}} \underbrace{\mathcal{M}_0^*(h)}_{\text{Born}} \underbrace{C_{2,\Gamma}}_{\text{colour}}\Big)$$

Two-loop algorithm for irreducible diagrams

- 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 $h_n^{(1)}$

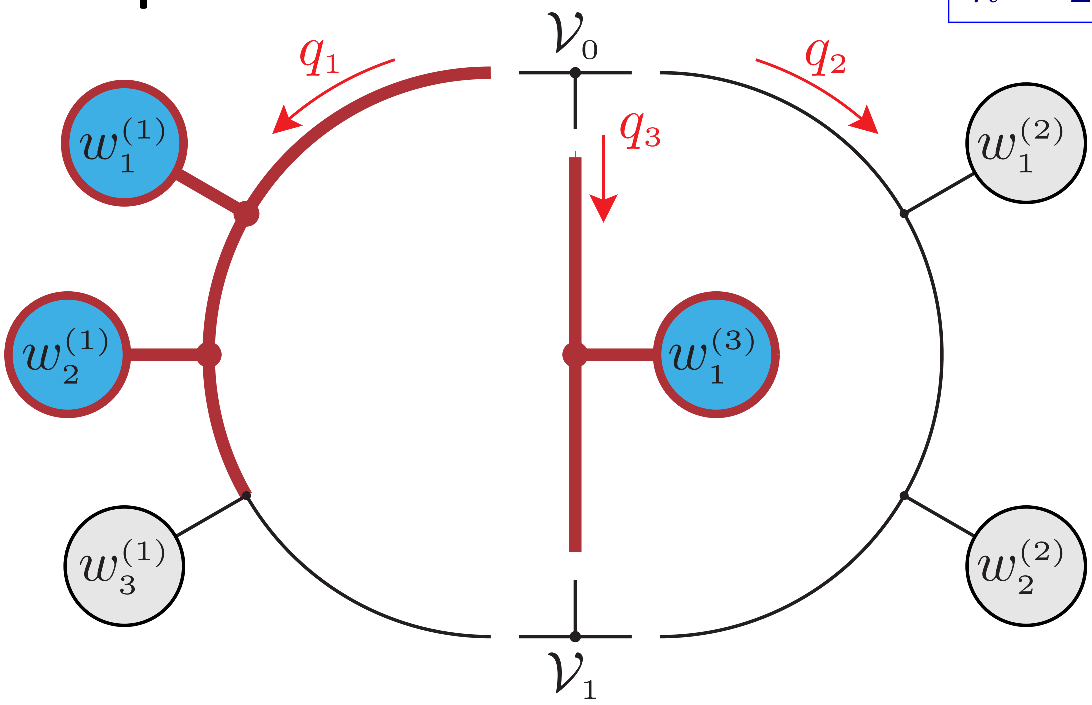
$\Rightarrow$ Partial chains depend on remaining helicities of the diagram $\check{h}_n^{(1)} = h - \sum_{k=1}^{n} h_k^{(1)}$

Two-loop algorithm for irreducible diagrams

- 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:

$n = 2$



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

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

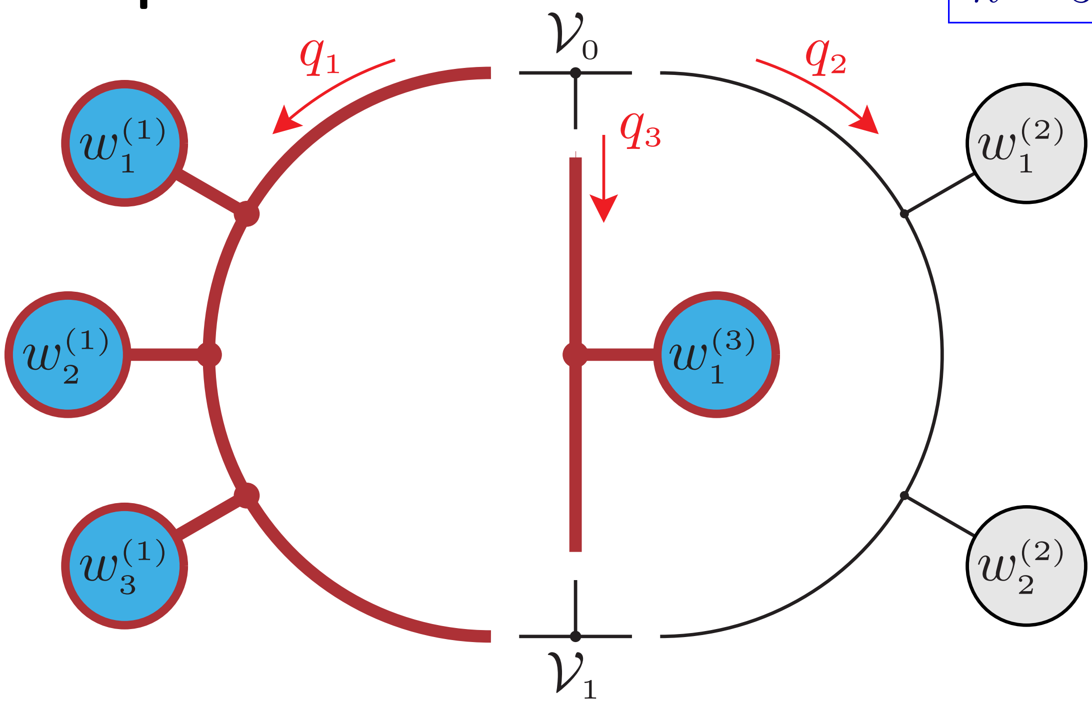
$\Rightarrow$ Partial chains depend on remaining helicities of the diagram $\check{h}_n^{(1)} = h - \sum_{k=1}^{n} h_k^{(1)}$

Two-loop algorithm for irreducible diagrams

- 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 $h_n^{(1)}$

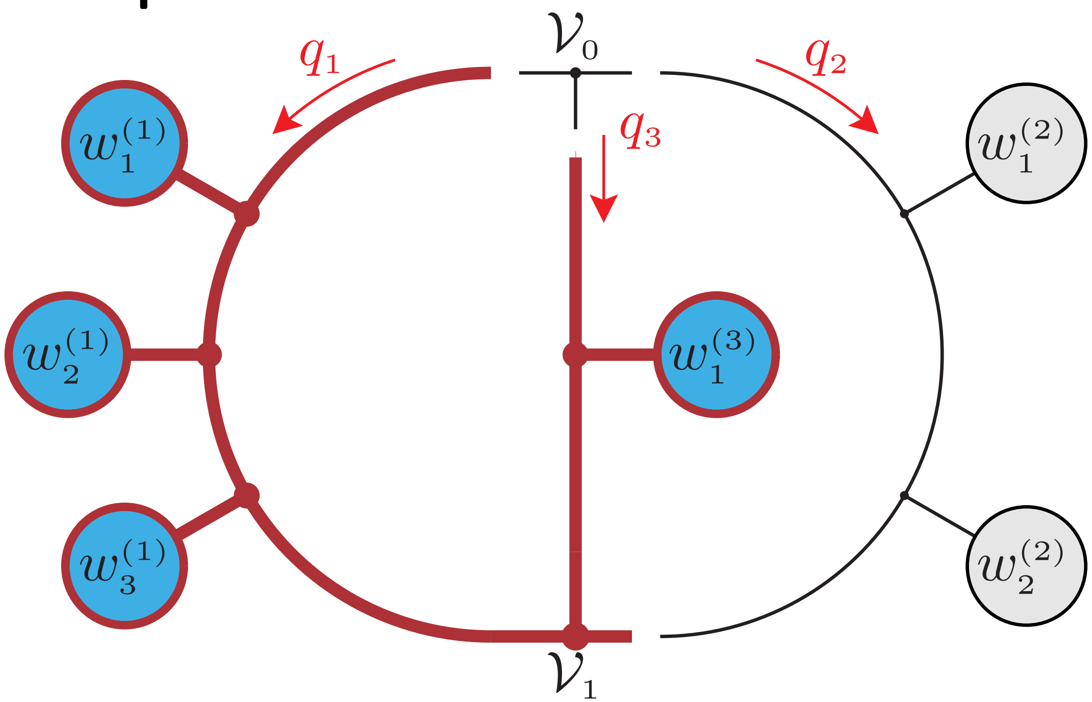
$\Rightarrow$ Partial chains depend on remaining helicities of the diagram $\check{h}_n^{(1)} = h - \sum_{k=1}^{n} h_k^{(1)}$

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

Two-loop algorithm for irreducible diagrams

- 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:



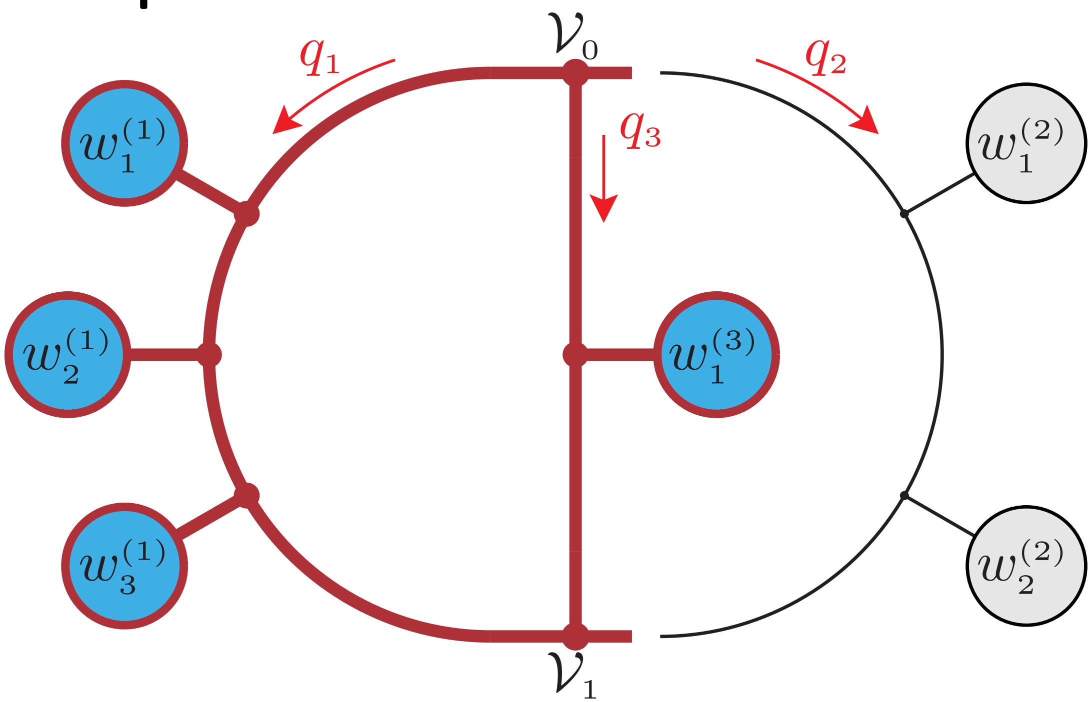
$$\mathcal{U}_1^{(13)}(q_1, q_3, h^{(2)} + h_0^{(V)}) = \sum_{h^{(3)}} \sum_{h_1^{(V)}} \mathcal{U}^{(1)}(q_1, \check{h}_{N_1-1}^{(1)})\, \mathcal{N}^{(3)}(q_3, h^{(3)})\, \mathcal{V}_1(q_1, q_3, h_1^{(V)})$$

◦ On-the-fly summation of chain helicity $h^{(3)}$ (and potential subtree helicity at $\mathcal{V}_1$)

Two-loop algorithm for irreducible diagrams

- 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:



$$\mathcal{U}_{-1}^{(123)}(q_1, q_2, h^{(2)}) = \sum_{h_0^{(V)}} \mathcal{U}_1^{(13)}(q_1, q_3, h^{(2)} + h_0^{(V)})\ \mathcal{V}_0(q_1, q_1, h_0^{(V)})\Big|_{q_3 \to -(q_1+q_2)}$$

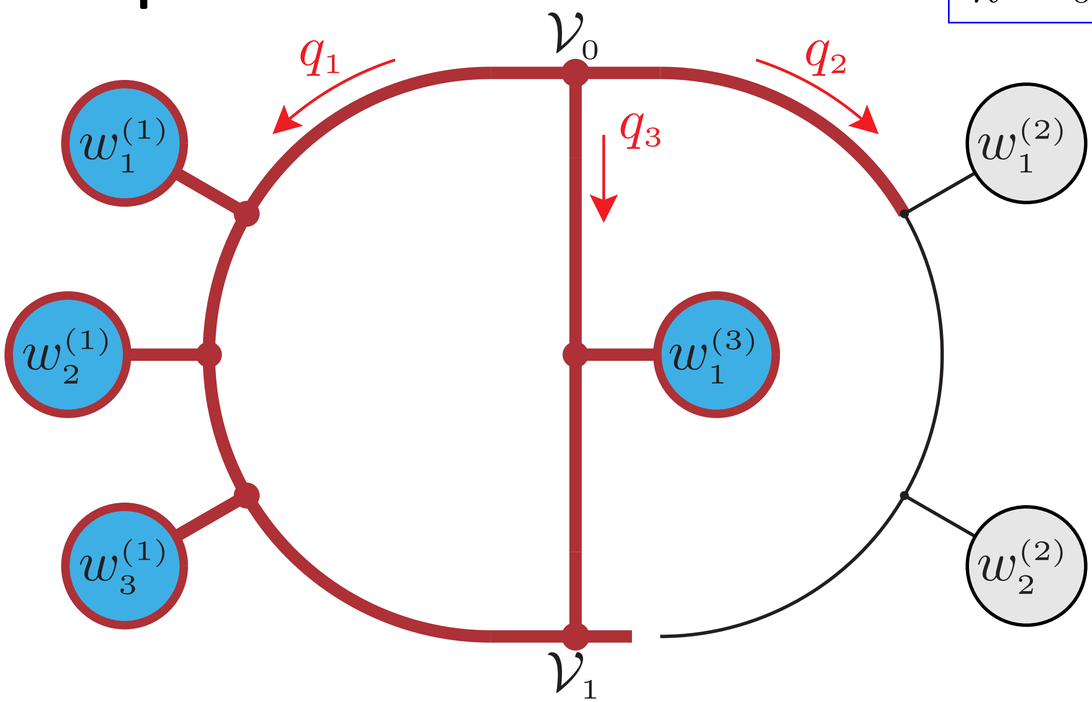
◦ Partial diagram depends on undressed chain helicity $h^{(2)}$ and two open indices

Two-loop algorithm for irreducible diagrams

- 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:

$n = 0$



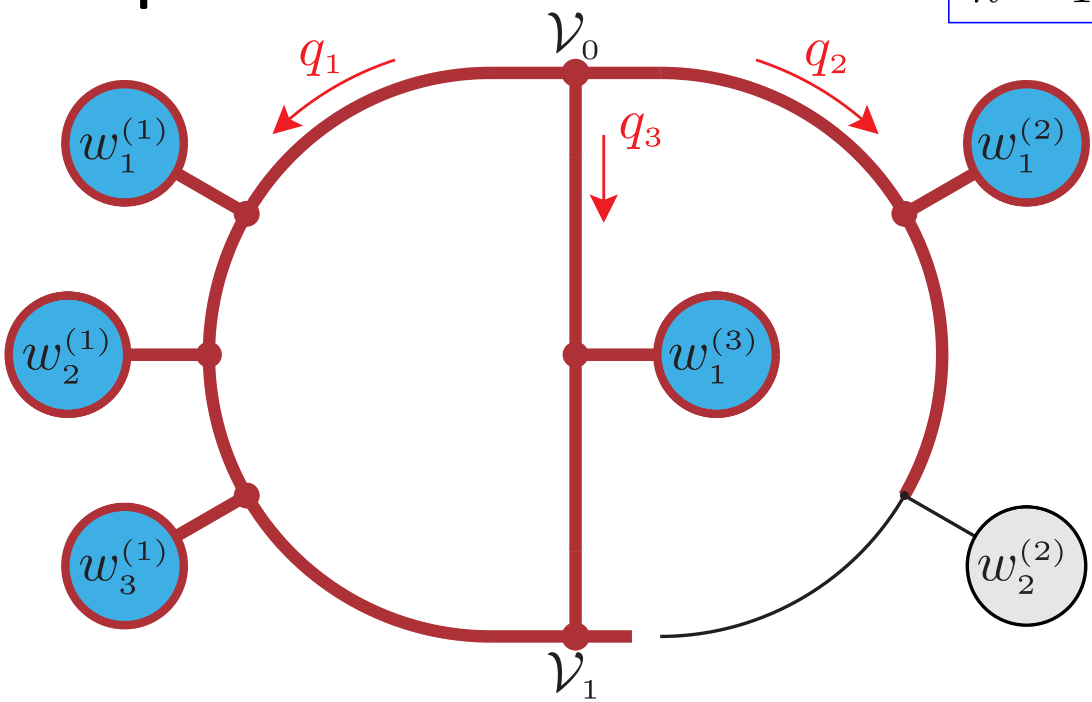
$$\mathcal{U}_n^{(123)}(q_1, q_2, \tilde{h}_n^{(2)}) = \sum_{h_n^{(2)}} \mathcal{U}_{n-1}^{(123)}(q_1, q_2, \tilde{h}_{n-1}^{(2)}) \; S_n^{(2)}(q_2, h_n^{(2)})$$

Two-loop algorithm for irreducible diagrams

- 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:

$n = 1$



$$\mathcal{U}_n^{(123)}(q_1, q_2, \tilde{h}_n^{(2)}) = \sum_{h_n^{(2)}} \mathcal{U}_{n-1}^{(123)}(q_1, q_2, \tilde{h}_{n-1}^{(2)})\; S_n^{(2)}(q_2, h_n^{(2)})$$

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

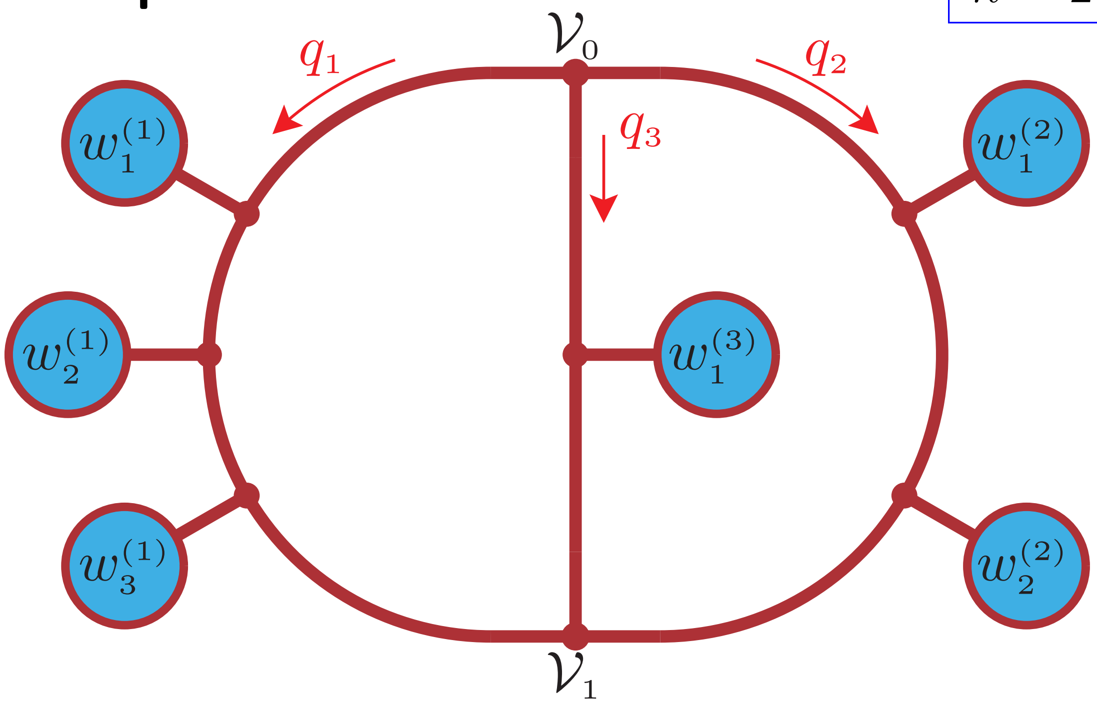
$\Rightarrow$ Partial diagram depends only on helicities of remaining undressed segments

Two-loop algorithm for irreducible diagrams

- 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:

$n = 2$



$$\mathcal{U}_n^{(123)}(q_1, q_2, \tilde{h}_n^{(2)}) = \sum_{h_n^{(2)}} \mathcal{U}_{n-1}^{(123)}(q_1, q_2, \tilde{h}_{n-1}^{(2)}) \; S_n^{(2)}(q_2, h_n^{(2)})$$

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

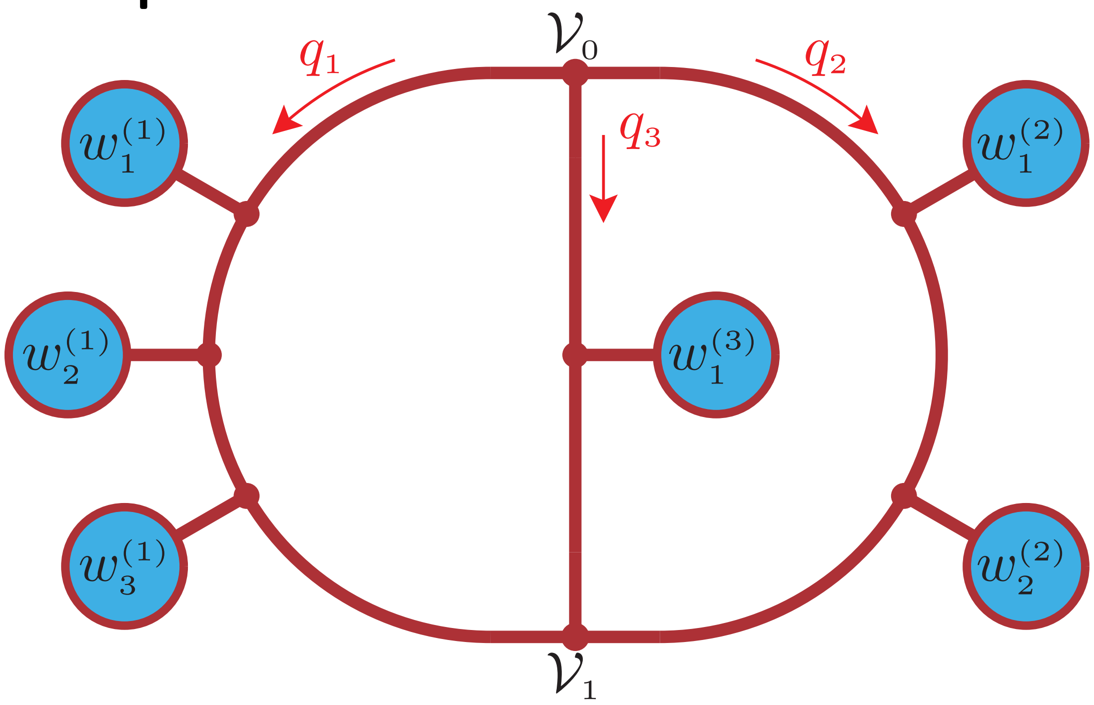
$\Rightarrow$ Partial diagram depends only on helicities of remaining undressed segments

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

Two-loop algorithm for irreducible diagrams

- 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:

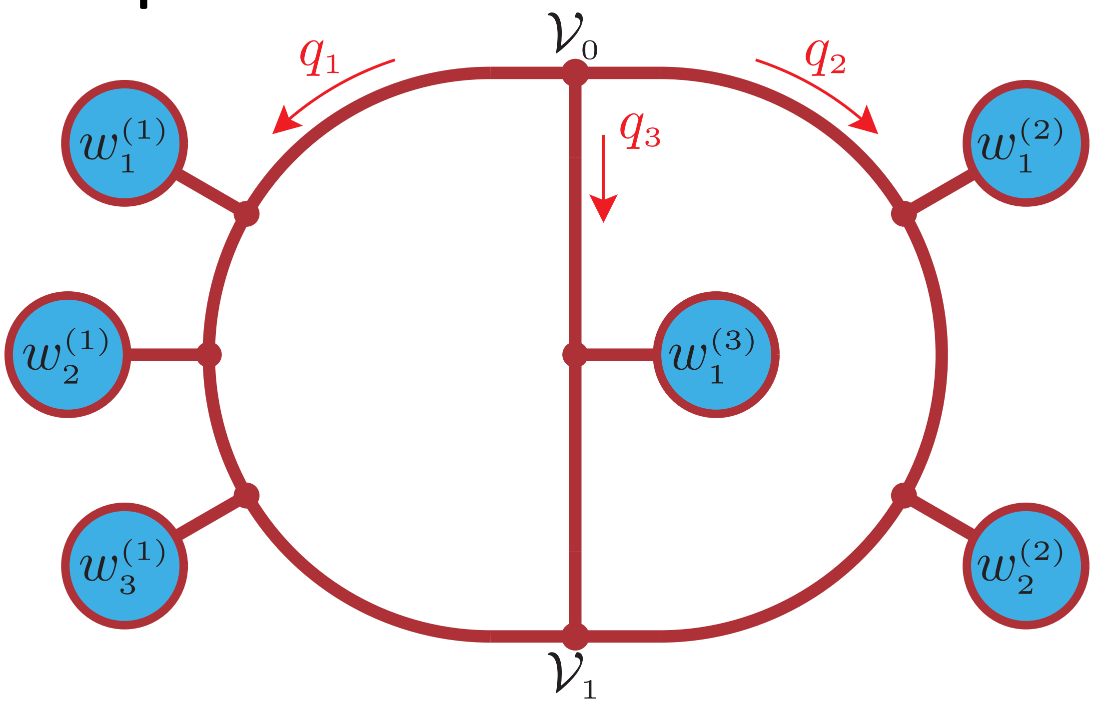
$$\mathcal{U}_A + \mathcal{U}_B = \left(\mathcal{U}_{A,n} \cdot S_{n+1} \cdots S_N\right) + \left(\mathcal{U}_{B,n} \cdot S_{n+1} \cdots S_N\right) = \left(\mathcal{U}_{A,n} + \mathcal{U}_{B,n}\right) \cdot S_{n+1} \cdots S_N$$

Merge partially dressed diagrams with same topology and subsequent recursion steps

Two-loop algorithm for irreducible diagrams

- 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:

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

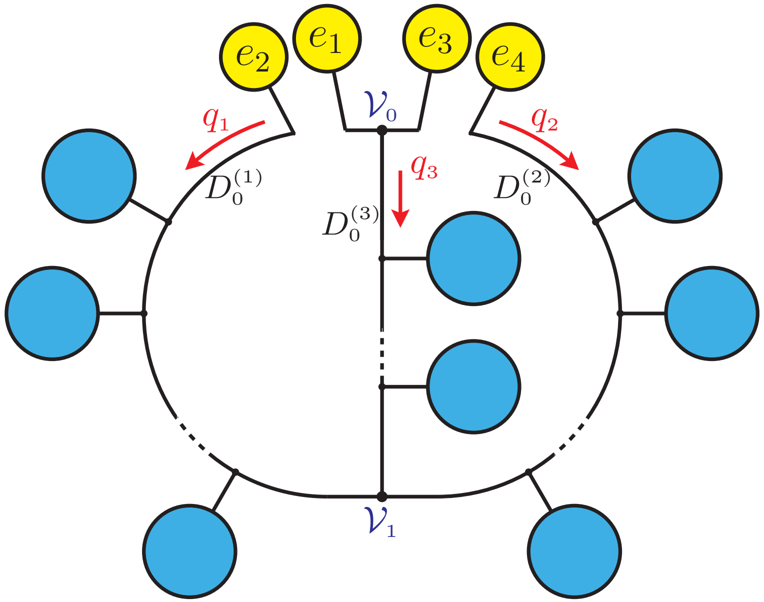
Merge partially dressed diagrams with same topology and subsequent recursion steps

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

Fully implemented for QED and QCD corrections to the SM

Numerical stability

Pseudo-tree test



- Cut-open diagram at two propagators
- Saturate indices with random wavefunctions $e_1, \ldots, e_4$
- Evaluate integrand constructed with new two-loop algorithm at fixed values for q_1, q_2
  $\Rightarrow \widehat{\mathcal{W}}^{(2L)}_{02,\Gamma} = \frac{U(q_1,q_2)}{\mathcal{D}(q_1,q_2)} \quad \Rightarrow \quad \widehat{\mathcal{W}}^{(2L)}_{02} = \sum_\Gamma \widehat{\mathcal{W}}^{(2L)}_{02,\Gamma}$
- Compute the same object with the OpenLoops tree-level algorithm for fixed $q_1, q_2 \Rightarrow \widehat{\mathcal{W}}^{(t)}_{02}$
  Compute relative numerical uncertainty in double (DP) and quadruple (QP) precision

$$\mathcal{A}^{(t)} := \log_{10}\left(\frac{|\widehat{\mathcal{W}}^{(t)}_{02} - \widehat{\mathcal{W}}^{(2L)}_{02}|}{\text{Min}(|\widehat{\mathcal{W}}^{(t)}_{02}|, |\widehat{\mathcal{W}}^{(2L)}_{02}|)}\right)$$

$\Rightarrow$ **Implementation validated** for wide range of processes (10^5 uniform random points)

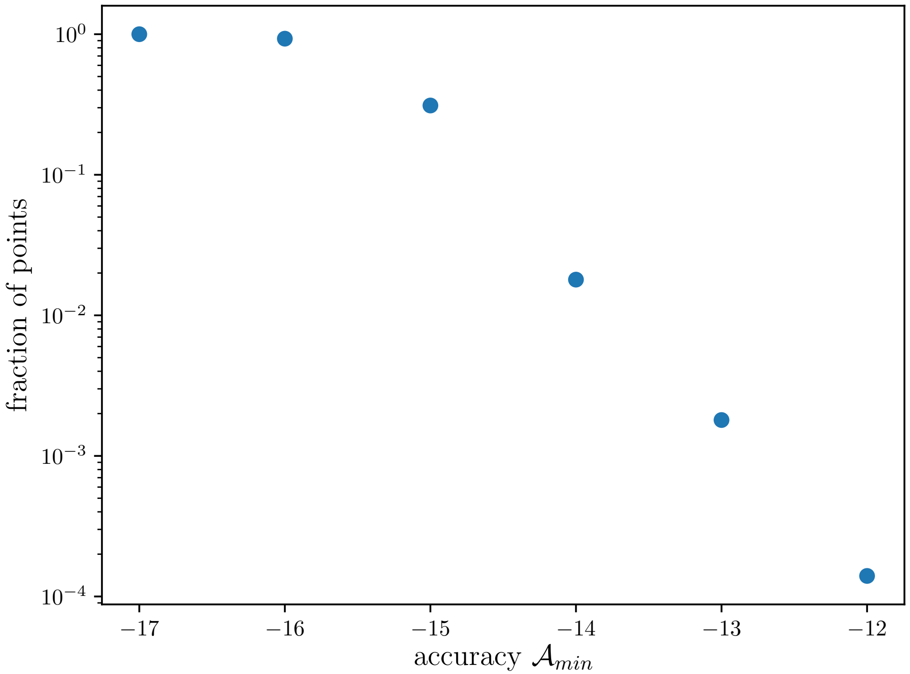
Typical accuracy around 10^{-15} in DP and 10^{-30} in QP, and always much better than 10^{-17} in QP
$\Rightarrow$ **QP calculation as benchmark for numerical accuracy of DP calculation**

Numerical stability

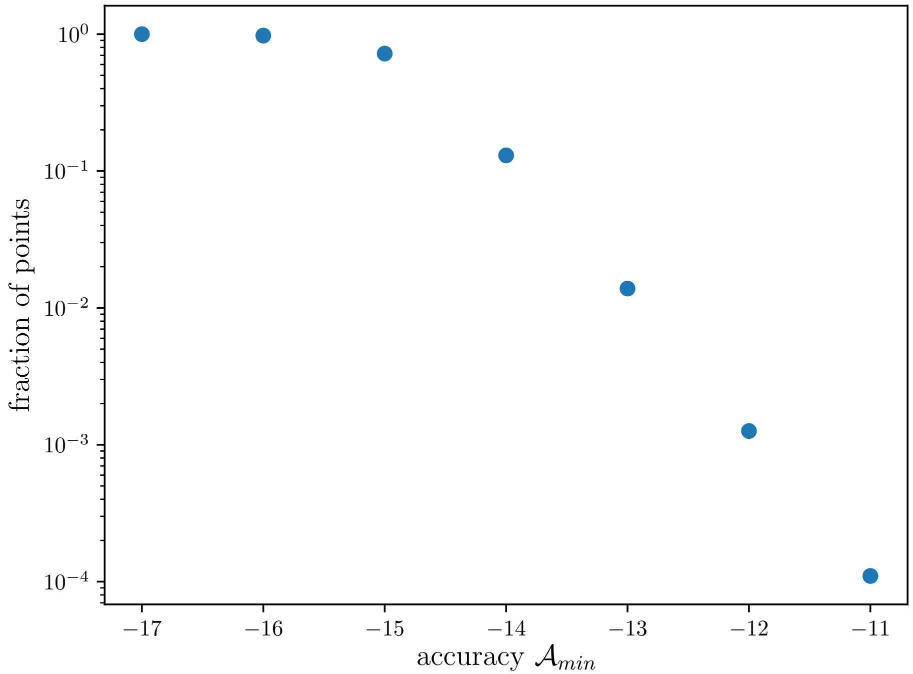
Numerical instability of double (DP) wrt quad precision (QP) calculation:

$$\mathcal{A}_{\mathrm{DP}} = \log_{10}\left(\frac{|\widehat{\mathcal{W}}_{02}^{(2\mathrm{L},\mathrm{DP})} - \widehat{\mathcal{W}}_{02}^{(2\mathrm{L},\mathrm{QP})}|}{\mathrm{Min}(|\widehat{\mathcal{W}}_{02}^{(2\mathrm{L},\mathrm{DP})}|, |\widehat{\mathcal{W}}_{02}^{(2\mathrm{L},\mathrm{QP})}|)}\right)$$

Fraction of points with $\mathcal{A}_{\mathrm{DP}} > A_{\min}$ as a function of $A_{\min}$ for 10^5 uniform random points



$gg \to \bar{t}t$



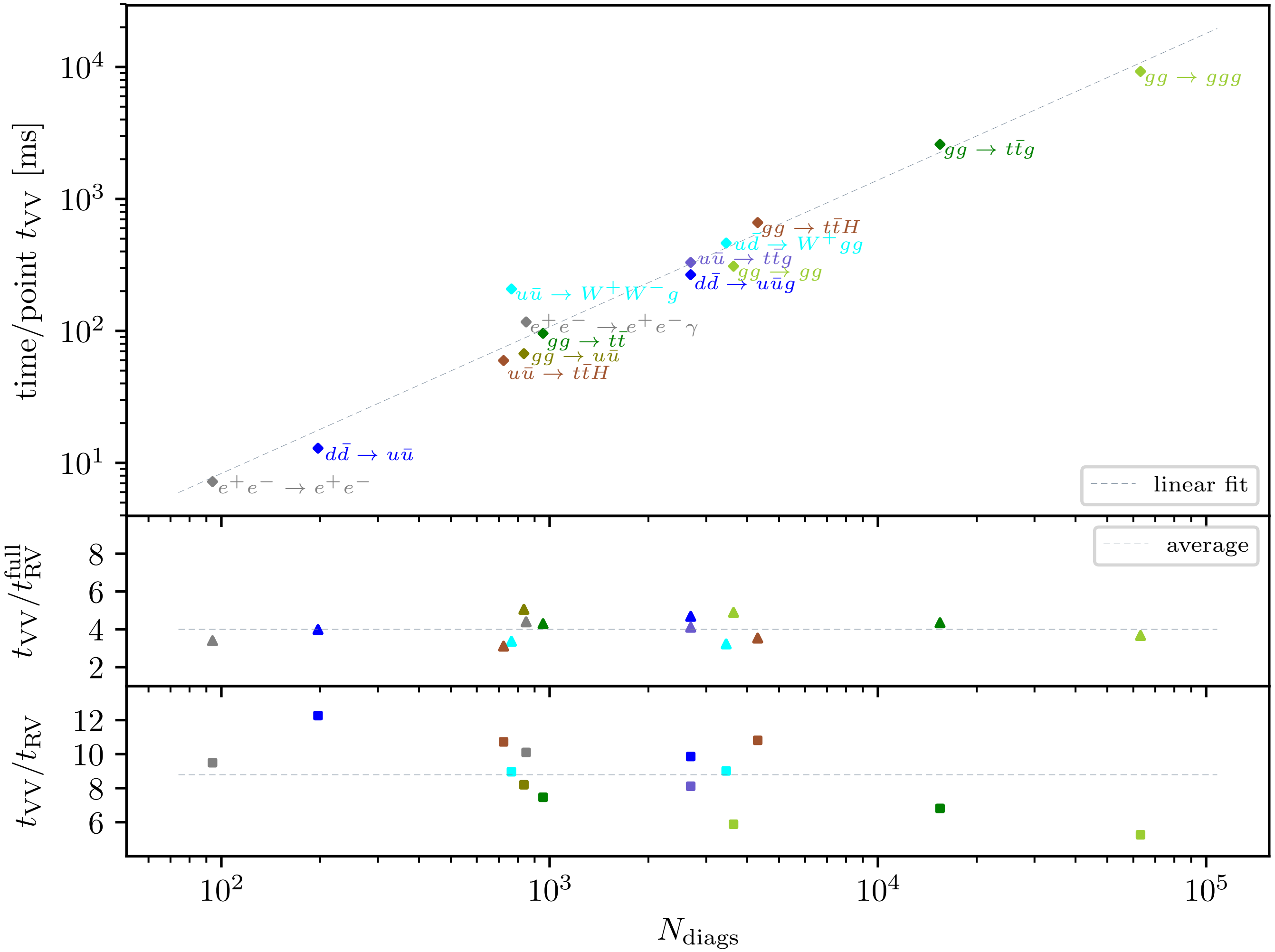
$d\bar{d} \to u\bar{u}g$

Excellent numerical stability

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

Timings for two-loop tensor coefficients

QED, QCD and SM (NNLO QCD) processes (single Intel i7-6600U @ 2.6 GHz, 16GB RAM, 10^3 points)



$2 \to 2$ process: $10 - 300$ ms/psp

$2 \to 3$ process: $65 - 9200$ ms/psp (on a laptop)

Runtime $\propto$ number of diagrams

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

Constant ratios between virtual–virtual (VV) and real-virtual (RV) with and without 1-loop integrals

- tensor coefficients: $\frac{t_{\mathrm{VV}}}{t_{\mathrm{RV}}} \sim 9$
- full RV: $\frac{t_{\mathrm{VV}}}{t_{\mathrm{RV}}^{\mathrm{full}}} \sim 4$

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

III. OpenLoops features at 1 and 2 loops

General algorithm: Any model can be implemented and any process can be generated automatically (provided the tensor/master integrals are available)

Program structure is the same at 1 and 2 loops:

- Process generator (Mathematica) $\rightarrow$ process libraries (Fortran)
- OpenLoops program (Fortran) with process-independent routines and user interfaces
  $\rightarrow$ Simple extension of interfaces and same input parameters as at 1 loop

$\Rightarrow$ **Many OpenLoops features transfered from 1 to 2 loops**

- Wide selection of diagram filters
- Polarisation selection for external particles
- Fully automated and flexible power counting in any number of coupling constants
  $\rightarrow$ Selection of all contributions of order $\alpha^n \alpha_s^m$ to $\mathcal{W} \sim |\mathcal{M}|^2$ in a fully automated way
  $\rightarrow$ Selection of specific powers in charges, e.g. QED corrections to $e^+e^- \rightarrow \mu^+\mu^-$ split into electronic, muonic and mixed corrections (power counting in Q_e,$Q_\mu \rightarrow 1$)
  $\rightarrow$ better control over numerical stability

New OpenLoops features at 1 and 2 loops

- **Massive QED and separation of EW corrections:**
  - Process libraries with any configuration of active lepton generations
  - massive e, μ, τ
  - pure QED (1 and 2 loops), pure weak (1 loop) and full EW (1 loop) corrections available

  Recently applied e.g. to Møller and Bhabha scattering at NNLO QED [Banerjee, Engel, Schalch, Signer, Ulrich] with OpenLoops and McMule [Banerjee, Engel, Signer, Ulrich]

- Fully automated efficient generation of **scattering processes factorising** into a hard process and any number of factorised subtrees, e.g.
  - QCD corrections to

$$e^+e^- \to \bar{q}q + X \quad \text{factorised into} \quad \underbrace{e^+e^- \to V^*}_{\text{subtree}} \text{ and } \underbrace{V^* \to \bar{q}q + X}_{\text{hard process}} \quad \text{with } V = \gamma, Z$$

  - Factorisation of $W \to l\nu_l$ and $Z \to l^+l^-$ decays

Input schemes, parameters and renormalisation

- Three **EW schemes** implemented:

	`ew_scheme`	input parameters	value of $1/\alpha$
$\alpha(0)$-scheme	0	$\alpha(0)$, M_W, M_Z, M_H + fermion masses	≈ 137
G_μ-scheme	1 (default)	G_μ, M_W, M_Z, M_H + fermion masses	≈ 132
$\alpha(M_Z)$-scheme	2	$\alpha(M_Z)$, M_W, M_Z, M_H + fermion masses	≈ 128

- Consistent treatment of resonances with **complex mass scheme** at 1-loop [Denner, Dittmaier]
  $\rightarrow$ complex mass $\mu_p^2 = M_p^2 - i\, M_p \Gamma_p$ from real physical mass M_p and width Γ_p as input

- **Different Renormalisation schemes** implemented, e.g. on-shell or $\overline{MS}$ for quark masses; different flavour schemes for α_S

- **External photons** in process $A \rightarrow B + n \underbrace{\gamma}_{\text{on-shell}} + n_* \underbrace{\gamma^*}_{\text{off-shell}} \quad (+ \underbrace{\gamma}_{\text{real emission}})$

  $\Rightarrow$ rescale with ratios of input α and $\alpha_{\text{on}} = \alpha(0)$, $\alpha_{\text{off}} = \begin{cases} \alpha|_{G_\mu} & \text{if } \alpha = \alpha(0), \\ \alpha & \text{if } \alpha = \alpha|_{G_\mu} \text{ or } \alpha = \alpha(M_Z) \end{cases}$

  $$\Rightarrow \quad \mathcal{W} \rightarrow \left[\frac{\alpha_{\text{on}}}{\alpha}\right]^n \left[\frac{\alpha_{\text{off}}}{\alpha}\right]^{n_*} \mathcal{W}$$
  (No rescaling for real emission)

  Optimal scale choice for external on-shell, off-shell and real-emission photons

IV. Summary and Outlook

One and two-loop calculations can be split into construction of tensor coefficients, reduction and evaluation of tensor integrals, and restoration of $(D-4)$-dim numerator parts

Status of the OpenLoops framework @ 2 loops:

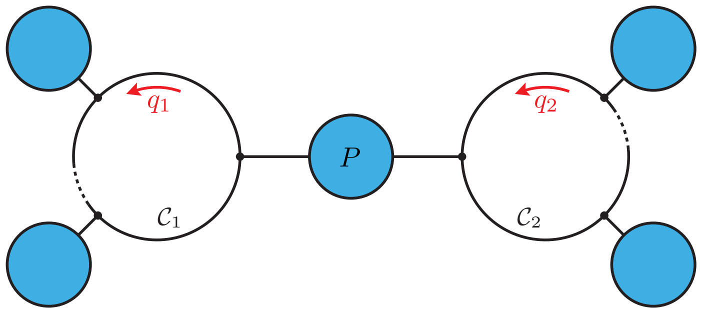
- Numerical calculation of **two-loop tensor coefficients**
  - **Completely general recursive algorithm** exploiting the factorisation of diagrams
  - **Fully implemented for NNLO QCD and NNLO QED** corrections in the SM
  - **Strong numerical precision and CPU performance** comparable to RV contributions
- Method to restore $(D-4)$-dim numerator parts through universal **Rational counterterms of UV origin** @ 2 loops $\rightarrow$ **Full set computed for QED and QCD corrections to the SM** $\rightarrow$ currently being implemented in the OpenLoops framework
- Many OpenLoops features transferable to 2 loops, such as power counting, renormalisation schemes, polarisation selection, input schemes, user interfaces

Short-term and mid-term projects:

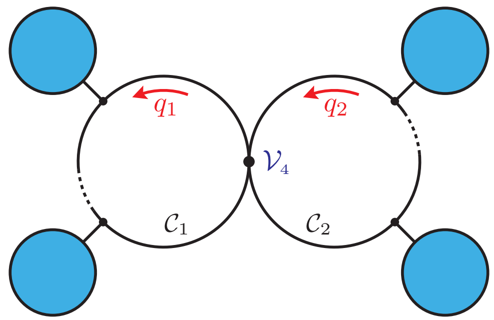
- **Rational terms of IR origin** $\rightarrow$ currently under investigation
- **Tensor integral reduction and evaluation** (analytical or numerical, in-house framework or external tool $\rightarrow$ possible mixture thereof)

Backup

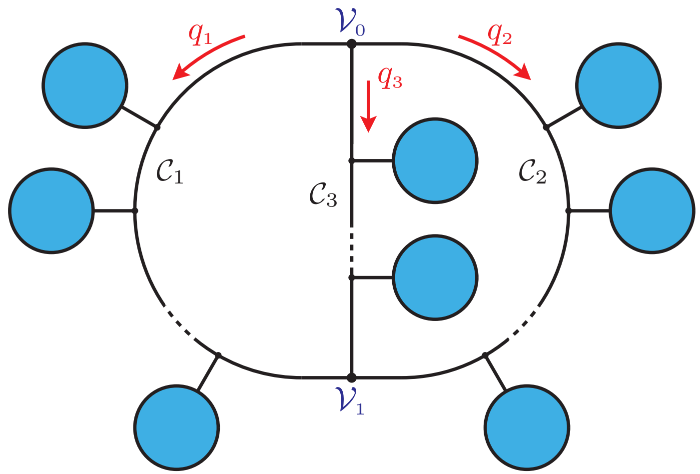
Two-loop diagrams



(Red2)



(Red1)



(Irreducible)

Two-loop diagrams consist of loop chains $\mathcal{C}_i$, each depending on a single loop momentum q_i.

Types of diagrams:

- **Reducible diagrams:** Two factorised loop integrals
  - **Red2:** Two loop chains $\mathcal{C}_1, \mathcal{C}_2$ connected by a tree-like bridge P.
  - **Red1:** Two loop chains $\mathcal{C}_1, \mathcal{C}_2$ connected by a single quartic vertex $\mathcal{V}_4$
- **Irreducible diagrams:** Three loop chains $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ with loop momenta $q_1, q_2, q_3 = -(q_1 + q_2)$ and two connecting vertices $\mathcal{V}_0, \mathcal{V}_1$

Processes considered in performance tests

corrections	process type	massless fermions	massive fermions	process
QED	$2 \to 2$	e	–	$e^+e^- \to e^+e^-$
	$2 \to 3$	e	–	$e^+e^- \to e^+e^-\gamma$
QCD	$2 \to 2$	u	–	$gg \to u\bar{u}$
		u, d	–	$d\bar{d} \to u\bar{u}$
		u	–	$gg \to gg$
		u	t	$u\bar{u} \to t\bar{t}g$
		u	t	$gg \to t\bar{t}$
		u	t	$gg \to t\bar{t}g$
	$2 \to 3$	u, d	–	$d\bar{d} \to u\bar{u}g$
		u	–	$gg \to ggg$
		u, d	–	$u\bar{d} \to W^+gg$
		u, d	–	$u\bar{u} \to W^+W^-g$
		u	t	$u\bar{u} \to t\bar{t}H$
		u	t	$gg \to t\bar{t}H$

Memory usage of the two-loop algorithm

	virtual–virtual memory [MB]		real–virtual [MB]	
hard process	segment-by-segment	diagram-by-diagram	coefficients	full
$e^+e^- \to e^+e^-$	18	8	6	23
$e^+e^- \to e^+e^-\gamma$	154	25	22	54
$gg \to u\bar{u}$	75	31	10	26
$gg \to t\bar{t}$	94	35	15	34
$gg \to t\bar{t}g$	2000	441	152	213
$u\bar{d} \to W^+gg$	563	143	54	90
$u\bar{u} \to W^+W^-g$	264	67	36	67
$u\bar{u} \to t\bar{t}H$	82	28	14	40
$gg \to t\bar{t}H$	604	145	50	90
$u\bar{u} \to t\bar{t}g$	323	83	41	74
$gg \to gg$	271	94	41	55
$d\bar{d} \to u\bar{u}$	18	10	9	20
$d\bar{d} \to u\bar{u}g$	288	85	39	68
$gg \to ggg$	6299	1597	623	683

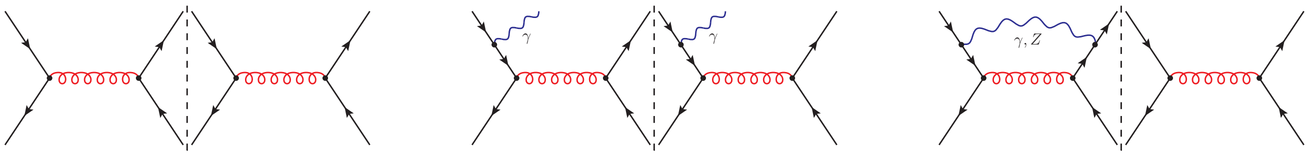
Power counting: Nontrivial QCD-EW interplay at 1 loop

Simple example: $q\bar{q} \to q\bar{q}$ cross section at Born level: $\underbrace{\mathcal{O}(\alpha_S^2)}_{\text{QCD}} + \underbrace{\mathcal{O}(\alpha_S^1\alpha^1)}_{\text{EW}-\text{QCD interf.}} + \underbrace{\mathcal{O}(\alpha^2)}_{\text{EW}}$

In general (e.g. $pp \to X +$ jets): $\mathcal{O}(\alpha_S^n\alpha^m) + \mathcal{O}(\alpha_S^{n-1}\alpha^{m+1}) + \ldots + \mathcal{O}(\alpha_S^{n-k}\alpha^{m+k})$

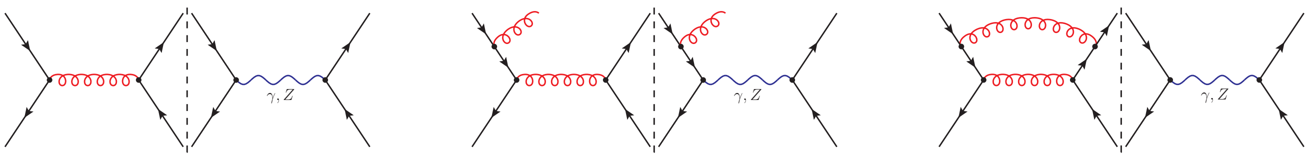
NLO EW corrections of $\mathcal{O}(\alpha_S^2\alpha^1)$ for $q\bar{q} \to q\bar{q}$:

- EW corrections to QCD Born



→ only full $\mathcal{O}(\alpha_S^2\alpha^1)$ IR finite

- QCD corrections to EW–QCD interference



→ $\mathcal{O}(\alpha)$ corrections can involve emissions of γ and $g, q, \bar{q}$

⇒ Mixed $\alpha\,\alpha_S$ power counting with non-trivial interference contributions

⇒ OpenLoops **provides any desired order $\mathcal{O}(\alpha_S^n\alpha^m)$ in a fully automated way**