A new method to generate and reduce one-loop amplitudes in OpenLoops 2

M. F. Zoller

in collaboration with F. Buccioni and S. Pozzorini

PSI, Villigen - Theoretical Particle Physics Seminar - 04/10/2017

Outline

I. Numerical amplitude generation in OpenLoops

II. New colour and helicity treatment

III. On-the-fly Reduction

IV. Numerical stability

V. Summary and Outlook

I. Numerical amplitude generation in OpenLoops

- Fully automated numerical algorithm for tree and one-loop amplitudes (h = helicity configuration):

$$\mathcal{W}_0 = \sum_h \sum_{\mathrm{col}} |\mathcal{M}_0(h)|^2, \qquad \mathcal{W}_1 = \sum_h \sum_{\mathrm{col}} 2\,\mathrm{Re}\Big[\mathcal{M}_0^*(h)\mathcal{M}_1(h)\Big], \qquad \mathcal{W}_1^{\text{loop-ind}} = \sum_h \sum_{\mathrm{col}} |\mathcal{M}_1(h)|^2$$

  Tree level and one-loop amplitudes are sums of Feynman diagrams

$$\mathcal{M}_0 = \sum_d \mathcal{M}_0^{(d)}, \qquad \mathcal{M}_1 = \sum_d \mathcal{M}_1^{(d)}$$

- hybrid tree-loop recursion $\Rightarrow$ high CPU efficiency and numerical stability

- $\mathrm{NLO\ QCD}$ and $\mathrm{NLO\ EW}$ corrections fully implemented

- OpenLoops is interfaced to Sherpa, Powheg, Herwig, Whizard, Geneva, Munich, Matrix

- OpenLoops 1 publicly available at openloops.hepforge.org [Cascioli, Lindert, Maierhöfer, Pozzorini]
  - Third party tools for the tensor integral reduction to scalar MIs: `Cuttools` 1.9.5 [Ossola, Papadopoulos, Pittau '08], `OneLoop` 3.6.1 [van Hameren '10], `Collier` 1.2 [Denner, Dittmaier, Hofer '16]
  - High tensor rank in loop momentum $q \Rightarrow$ high complexity
  - Stability in the IR region is challenging for $2 \to 4$ processes

Long-term goal: NNLO automation for $2 \to 2$ and $2 \to 3$ processes

- 2 loop amplitude construction and reduction needed $\Rightarrow$ avoid high tensor rank complexity
- Numerical stability at NLO for $2 \to 4$ is crucial

- OpenLoops 2 to be published soon [Buccioni, Lindert, Maierhöfer, Pozzorini, M.Z.]
  - Amplitude construction and integrand reduction merged $\Rightarrow$ On-the-fly Reduction

$\Rightarrow$ tensor rank ≤ 2 at all times

  - Stability issues addressed in a targeted way

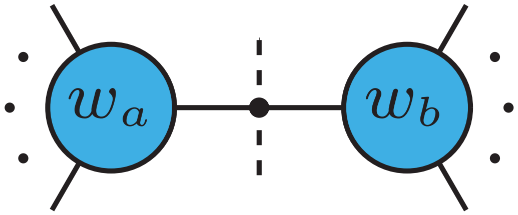
Tree level amplitudes

$$\mathcal{M}_0 = \sum_d \mathcal{M}_0^{(d)}$$

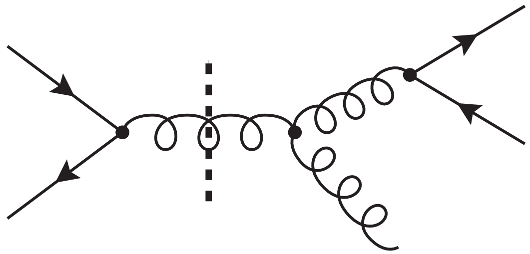
Each diagram factorizes into a colour factor and a colour stripped amplitude

$$\mathcal{M}_l^{(d)} = \mathcal{C}_l^{(d)} \, \mathcal{A}_l^{(d)} \, .$$

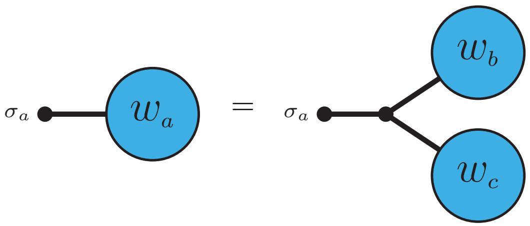
colour stripped $\mathcal{A}_0^{(d)}$ are split into subtrees by cutting an internal line:



for example



$\Rightarrow$ Numerical merging of subtrees performed recursively:



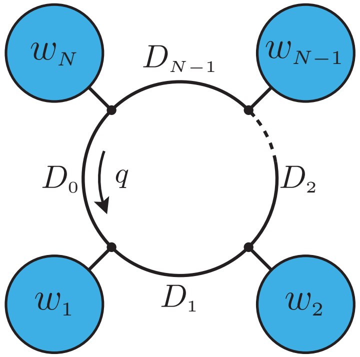
$$w_a^{\alpha}(k_a, h_a) = \frac{X^{\alpha}_{\beta\gamma}(k_b,k_c)}{k_a^2 - m_a^2} \, w_b^{\beta}(k_b, h_b) \, w_c^{\gamma}(k_c, h_c)$$

with momentum $k_a = k_b + k_c$ and for all possible helicity configurations $h_a = h_b + h_c$.

$\Rightarrow$ Once computed subtrees used in multiple Feynman diagrams at tree and loop level

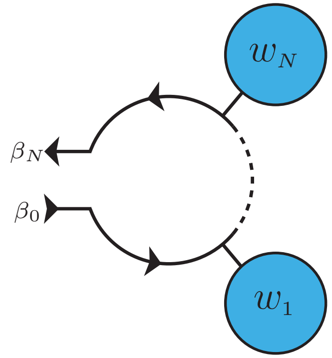
One-loop amplitude

$$\mathcal{A}_1^{(d)} = \int \mathrm{d}^D q \, \frac{\mathrm{Tr}\Big[\mathcal{N}(q,h)\Big]}{\bar{D}_0 \bar{D}_1 \cdots \bar{D}_{N-1}} =$$



cut open at $\bar{D}_0$

$$\Big[\mathcal{N}(q,h)\Big]^{\beta_N}_{\beta_0} =$$



propagators $D_i = (q+p_i)^2 - m_i^2$,
helicity configurations of subtree w_i: h_i

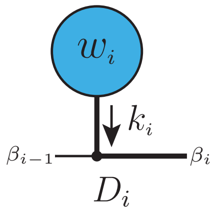
spinor/Lorentz indices $\beta_i \Rightarrow$ trace: contraction with $\delta^{\beta_0}_{\beta_N}$,
helicity configurations of $\mathcal{A}_1^{(d)}$: $h = h_1 + \ldots + h_N$

Numerator **factorizes** into **segments**:

$$\Big[\mathcal{N}(q,h)\Big]^{\beta_N}_{\beta_0} = \Big[\prod_{i=1}^{N} S_i(q,h_i)\Big]^{\beta_N}_{\beta_0} = \Big[S_1(q,h_1)\Big]^{\beta_1}_{\beta_0}\Big[S_2(q,h_2)\Big]^{\beta_2}_{\beta_1}\cdots\Big[S_N(q,h_N)\Big]^{\beta_N}_{\beta_{N-1}}$$

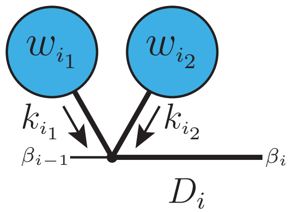
In the SM a segment (external subtree(s) + one loop vertex + propagator) is a q-polynomial of rank $r \le 1$:

3-point segment: $\Big[S_i(q,h_i)\Big]^{\beta_i}_{\beta_{i-1}} =$



$$= \Big\{\Big[Y^i_{\sigma_i}\Big]^{\beta_i}_{\beta_{i-1}} + \Big[Z^i_{\nu;\sigma_i}\Big]^{\beta_i}_{\beta_{i-1}} q^\nu\Big\}\, w_i^{\sigma_i}(k_i,h_i)$$

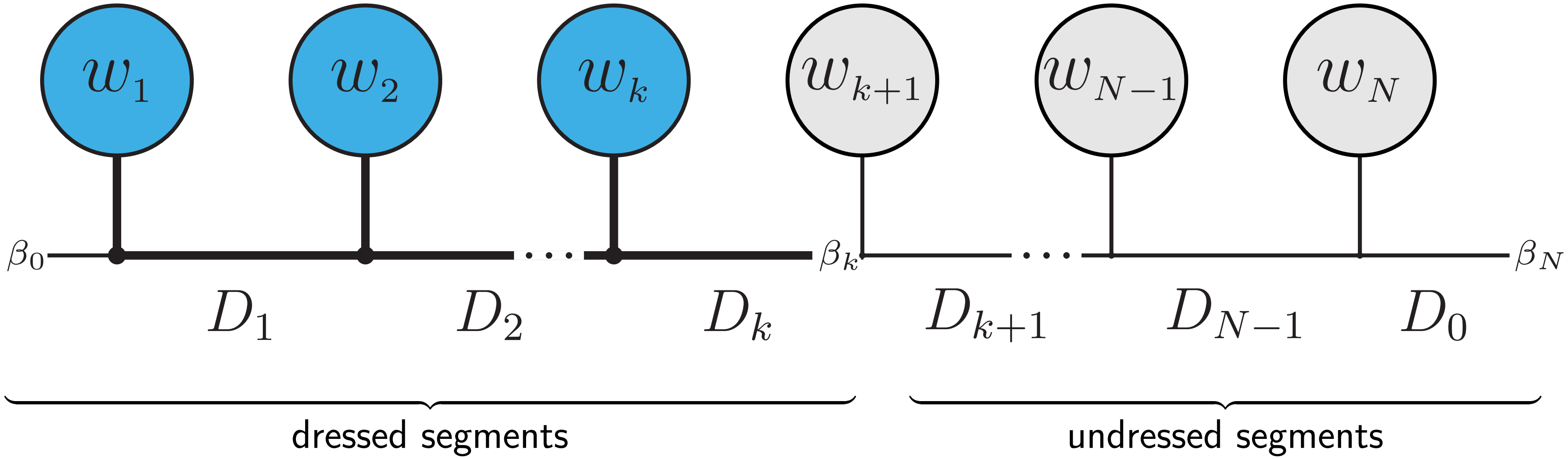
4-point segment: $\Big[S_i(q,h_i)\Big]^{\beta_i}_{\beta_{i-1}} =$



$$= \Big[Y^i_{\sigma_1\sigma_2}\Big]^{\beta_i}_{\beta_{i-1}} w_{i_1}^{\sigma_1}(k_{i_1},h_{i_1})\, w_{i_2}^{\sigma_2}(k_{i_2},h_{i_2}) \qquad (h_i = h_{i_1} + h_{i_2})$$

The OpenLoops dressing step

define partially dressed numerator $\boxed{\mathcal{N}_n(q,\hat{h}_n) = S_1(q,h_1)\cdots S_n(q,h_n)}$ $\quad (\hat{h}_n = \sum_{i=1}^{n} h_i)$



dressing step $\boxed{\mathcal{N}_n(q,\hat{h}_n) = \mathcal{N}_{n-1}(q,\hat{h}_{n-1}) S_n(q,h_n)}$ with initial condition $\mathcal{N}_0 = \mathbb{1}$

(rank $R \leq n$) performed numerically for the tensor coefficients in

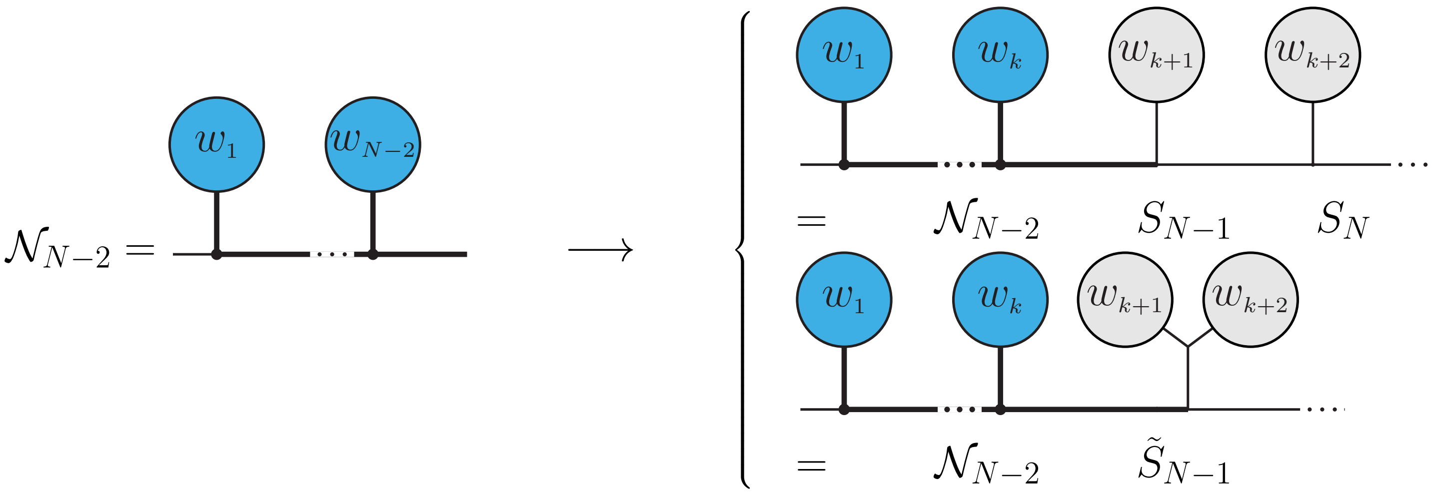
$$\mathcal{N}(q,\hat{h}_n) = \sum_{r=0}^{R} \mathcal{N}_{\mu_1\ldots\mu_r}(\hat{h}_n)\, q^{\mu_1}\cdots q^{\mu_r},$$

$$\left[\mathcal{N}_{\mu_1\ldots\mu_r}(\hat{h}_n)\right]^{\beta_n}_{\beta_0} = \left\{\left[\mathcal{N}_{\mu_1\ldots\mu_r}(\hat{h}_{n-1})\right]^{\beta_{n-1}}_{\beta_0}\left[Y^n_{\sigma_n}\right]^{\beta_n}_{\beta_{n-1}} + \left[\mathcal{N}_{\mu_2\ldots\mu_r}(\hat{h}_{n-1})\right]^{\beta_{n-1}}_{\beta_0}\left[Z^n_{\mu_1;\sigma_n}\right]^{\beta_n}_{\beta_{n-1}}\right\} w_n^{\sigma_n}(k_n,h_n)$$

Colour, helicity and diagram sums in OpenLoops 1

- for each diagram d and global helicity h configuration construct $\mathrm{Tr}\Big[\mathcal{N}_N^{(d)}(q,h)\Big]$
- colour sum with Born: $\mathcal{V}_N^{(d)}(q,h) = 2\left(\sum_{\mathrm{col}} \mathcal{M}_0(h)^* \mathcal{C}^{(d)}\right) \mathrm{Tr}\Big[\mathcal{N}_N^{(d)}(q,h)\Big]$
- helicity sum: $\mathcal{V}_N^{(d)}(q) = \sum_h \mathcal{V}_N^{(d)}(q,h)$
- sum same topology diagrams, reduce and evaluate integrals: $\int \mathrm{d}^D q \sum_d \frac{\mathrm{Tr}\Big[\mathcal{V}_N^{(d)}(q,0)\Big]}{\bar{D}_0,\ldots,\bar{D}_{N-1}}$

$\Rightarrow$ parent-child trick (recycling of colour-stripped partially dressed numerators)



New idea: formulate the OpenLoops recursion *directly* for
the colour-helicity summed interference with the Born amplitude $\mathcal{V}_N^{(d)}(q,0)$.

II. New colour and helicity treatment

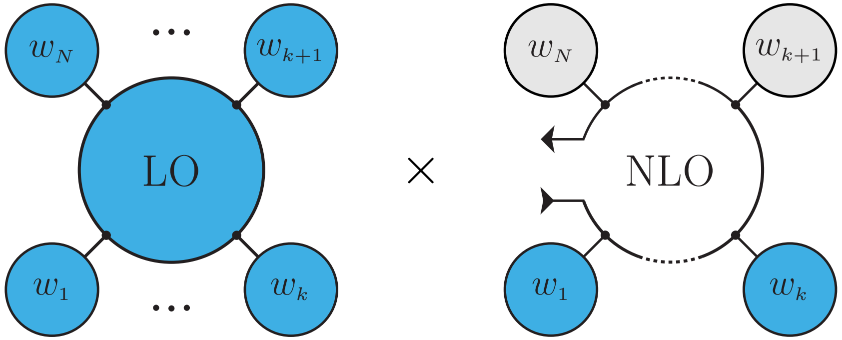
consider color-helicity summed numerator

$$\mathcal{V}_N(q,0) = \sum_h 2\left(\sum_{\text{col}} \mathcal{M}_0(h)^* \mathcal{C}\right) \mathcal{N}_N(q,h) = \sum_{h_1 \ldots h_N} \underbrace{2\left(\sum_{\text{col}} \mathcal{M}_0(h)^* \mathcal{C}\right)}_{=\mathcal{V}_0(h)} S_1(q,h_1) \cdots S_N(q,h_N)$$

and formulate recursion for partially dressed numerator with nested helicity sums

$$\mathcal{V}_n(q,\check{h}_n) \;=\; \sum_{h_n}\left[\ldots \sum_{h_2}\left[\sum_{h_1} \mathcal{V}_0(h) S_1(q,h_1)\right] S_2(q,h_2) \cdots \right] S_n(q,h_n) \quad \forall\; \check{h}_n = h_{n+1} + \cdots + h_N$$

$$= \sum_{h_1 \ldots h_n} \sum_{\text{col}}$$



and a dressing step as $\boxed{\mathcal{V}_n(q,\check{h}_n) = \sum_{h_n} \mathcal{V}_{n-1}(q,\check{h}_{n-1}) S_n(q,h_n)}$

$\Rightarrow$ Remaining helicity dof are those of the undressed segments!
Parent-child trick not possible (different colour factors) $\Rightarrow$ OpenLoops Merging instead

The OpenLoops Merging

Sum partially dressed open loops

$$\mathcal{V}_n(q,\check{h}_n) = \sum_{\alpha} \mathcal{V}_n^{(\alpha)}(q,\check{h}_n)$$

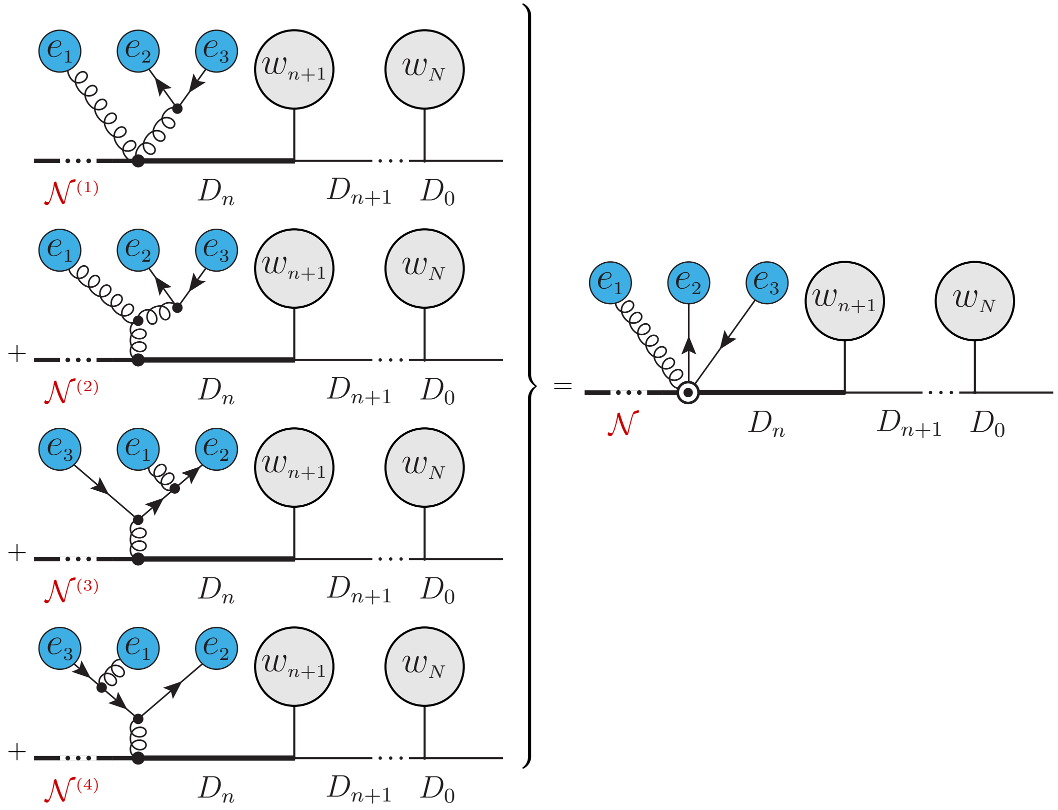
with

- the same topology $\bar{D}_0, \ldots, \bar{D}_{N-1}$
- the same undressed segments $S_{n+1}, \ldots, S_N$

since

$$\sum_{\alpha} \frac{\mathcal{V}_n^{(\alpha)} S_{n+1} \cdots S_{N-1}}{\bar{D}_0 \bar{D}_1 \cdots \bar{D}_{N-1}} = \frac{\mathcal{V}_n S_{n+1} \cdots S_{N-1}}{\bar{D}_0 \bar{D}_1 \cdots \bar{D}_{N-1}}$$

Example:

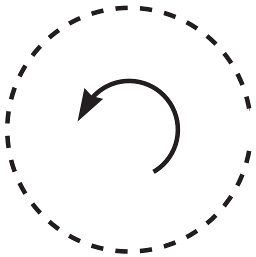


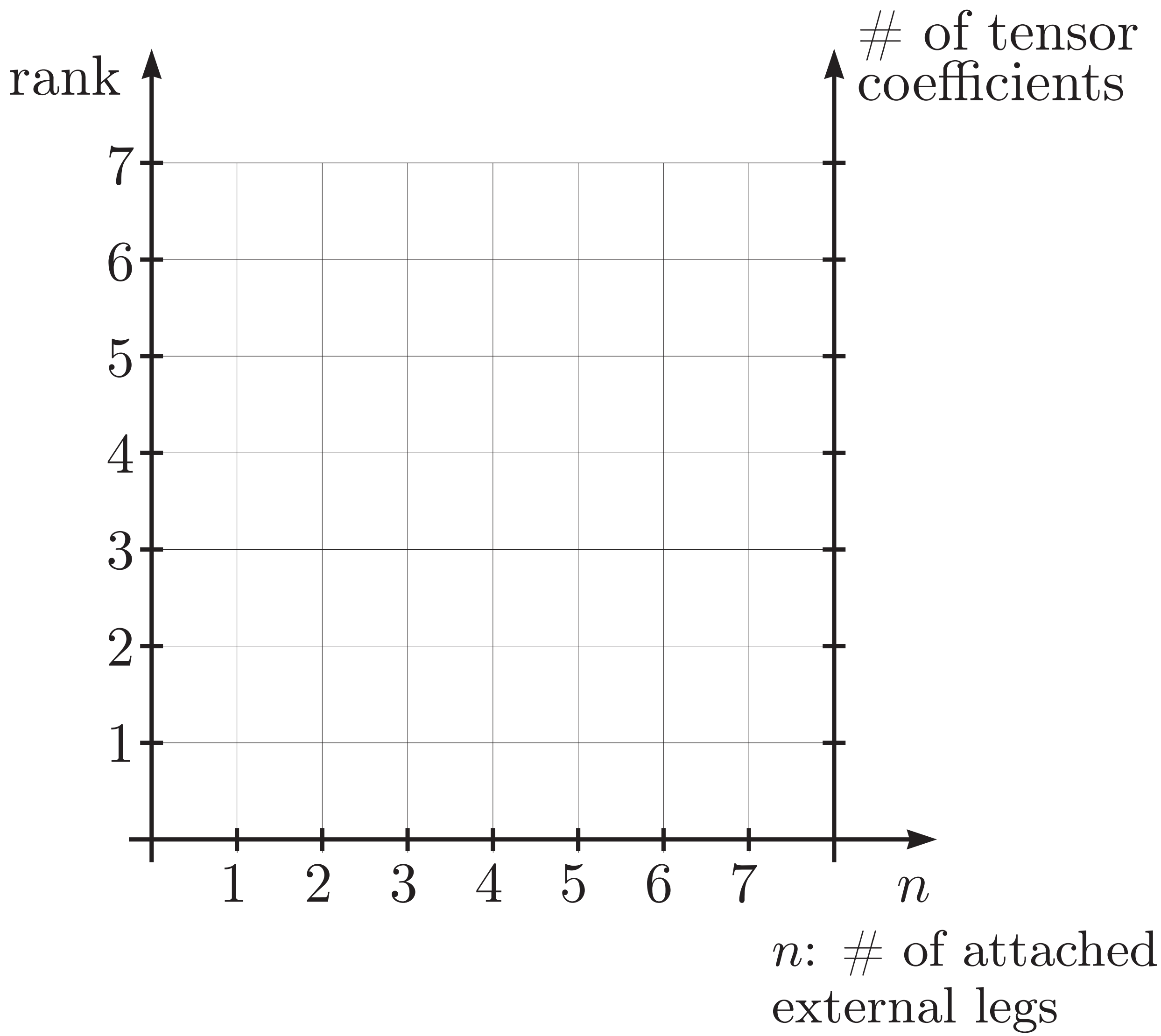
▷ dressing steps for $S_{n+1}, \ldots, S_N$ performed only once for the merged object

▷ crucial for combination with on-the-fly integrand reduction (see later)

Amplitude generation and tensor reduction in OpenLoops 1

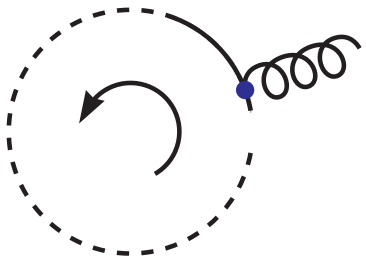
Example:

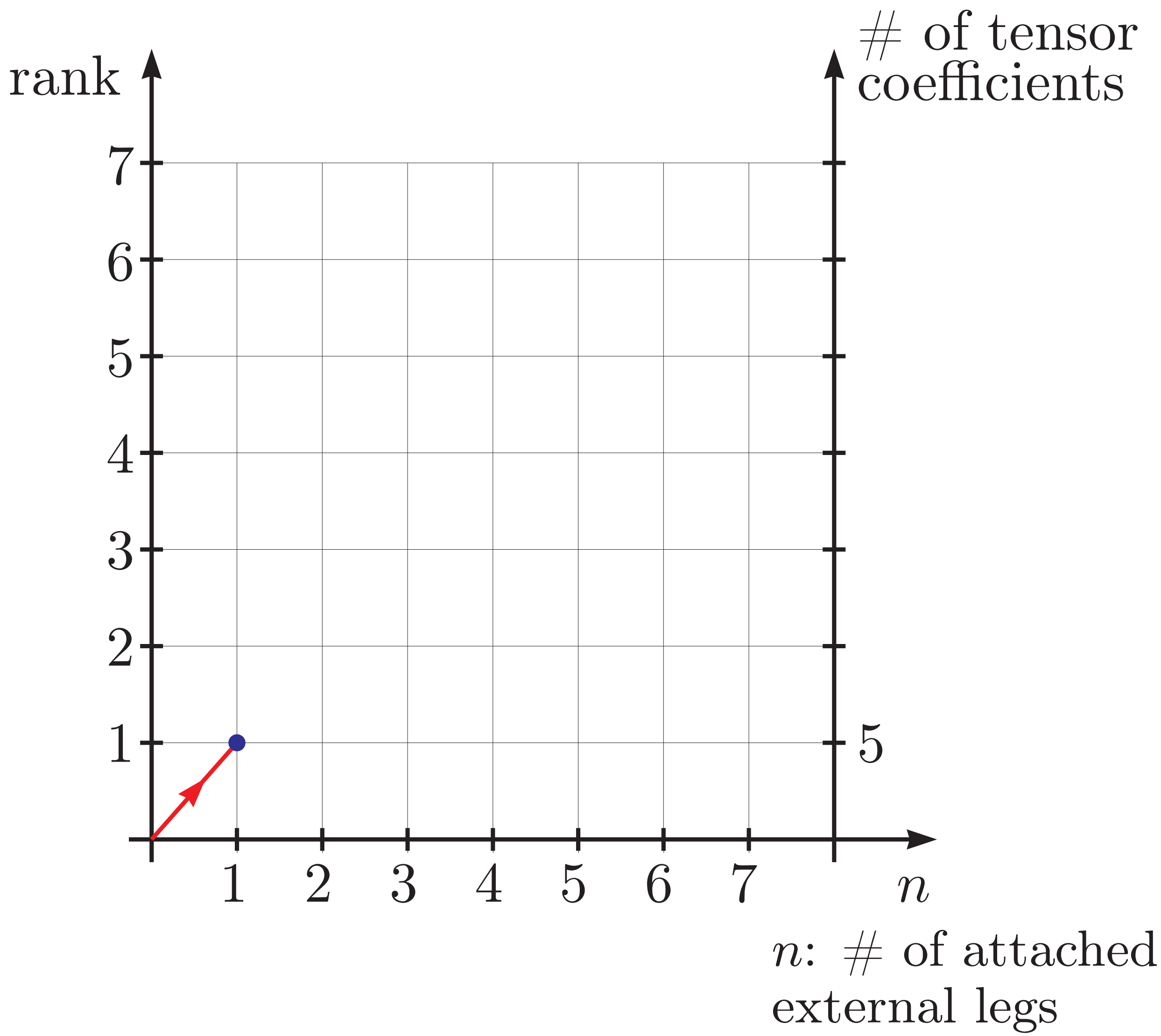




Amplitude generation and tensor reduction in OpenLoops 1

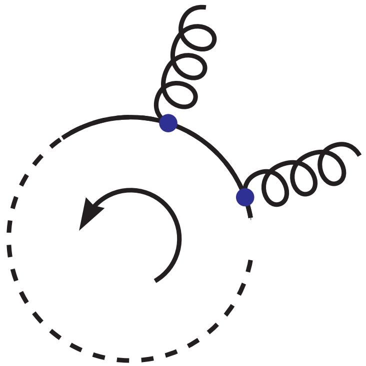
Example:

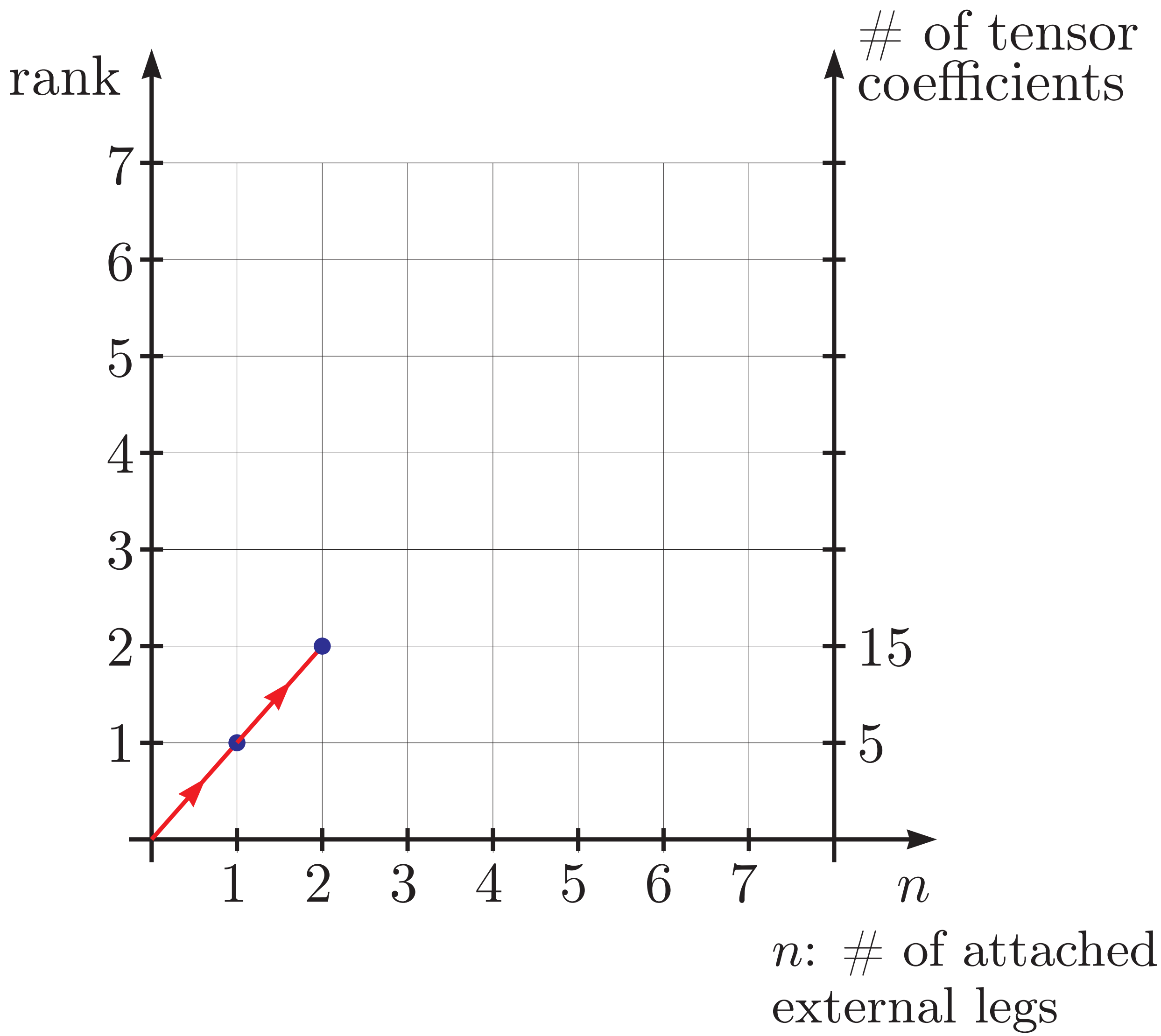




Amplitude generation and tensor reduction in OpenLoops 1

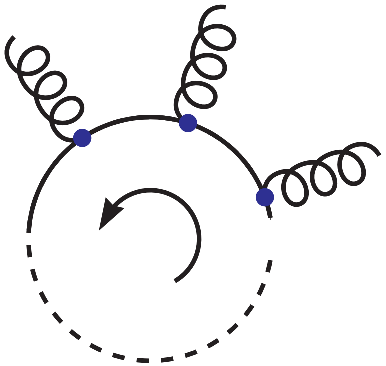
Example:

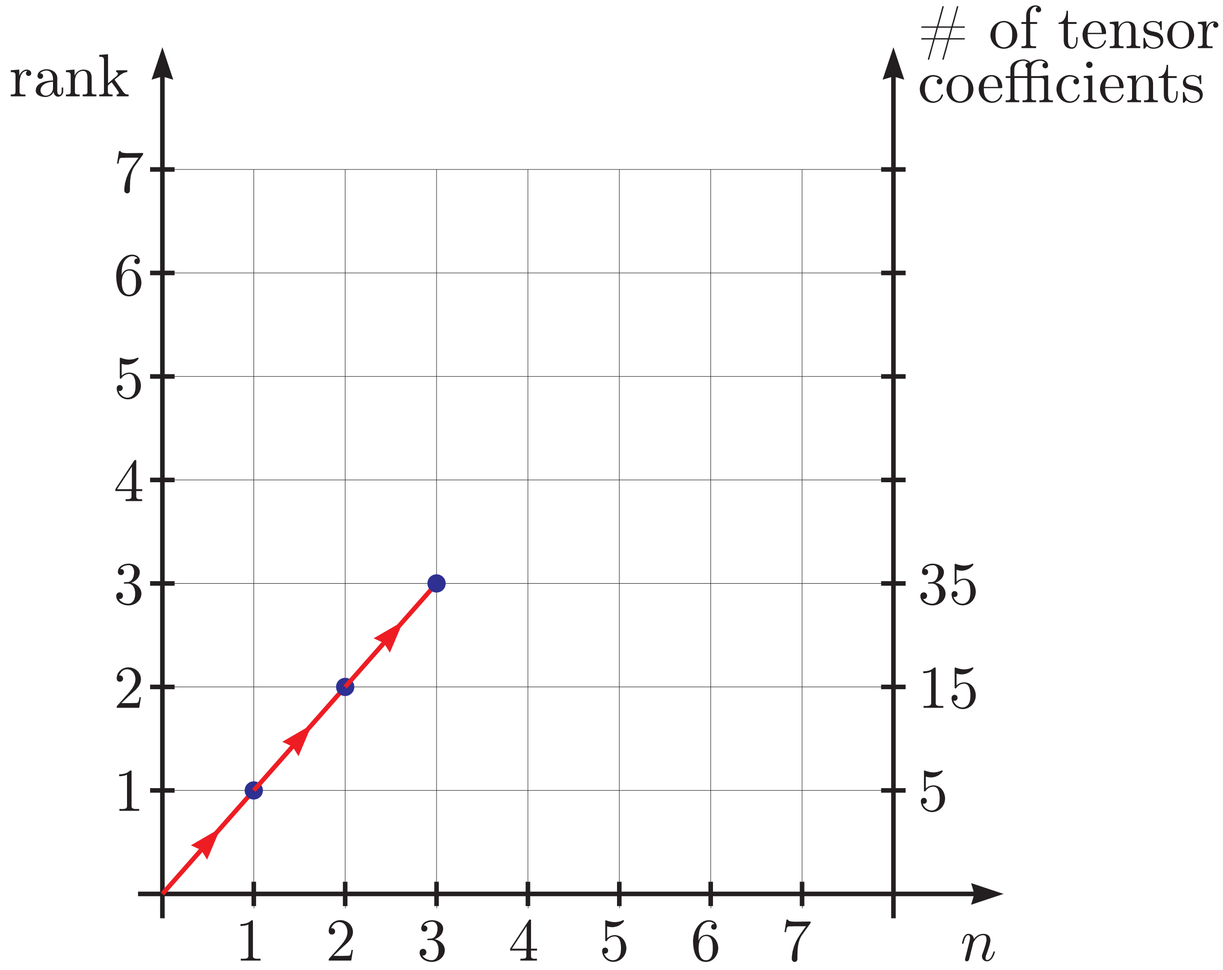




Amplitude generation and tensor reduction in OpenLoops 1

Example:

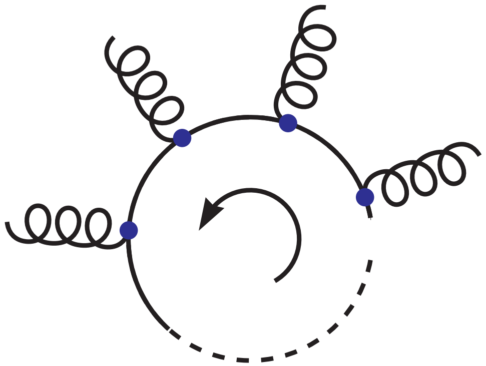


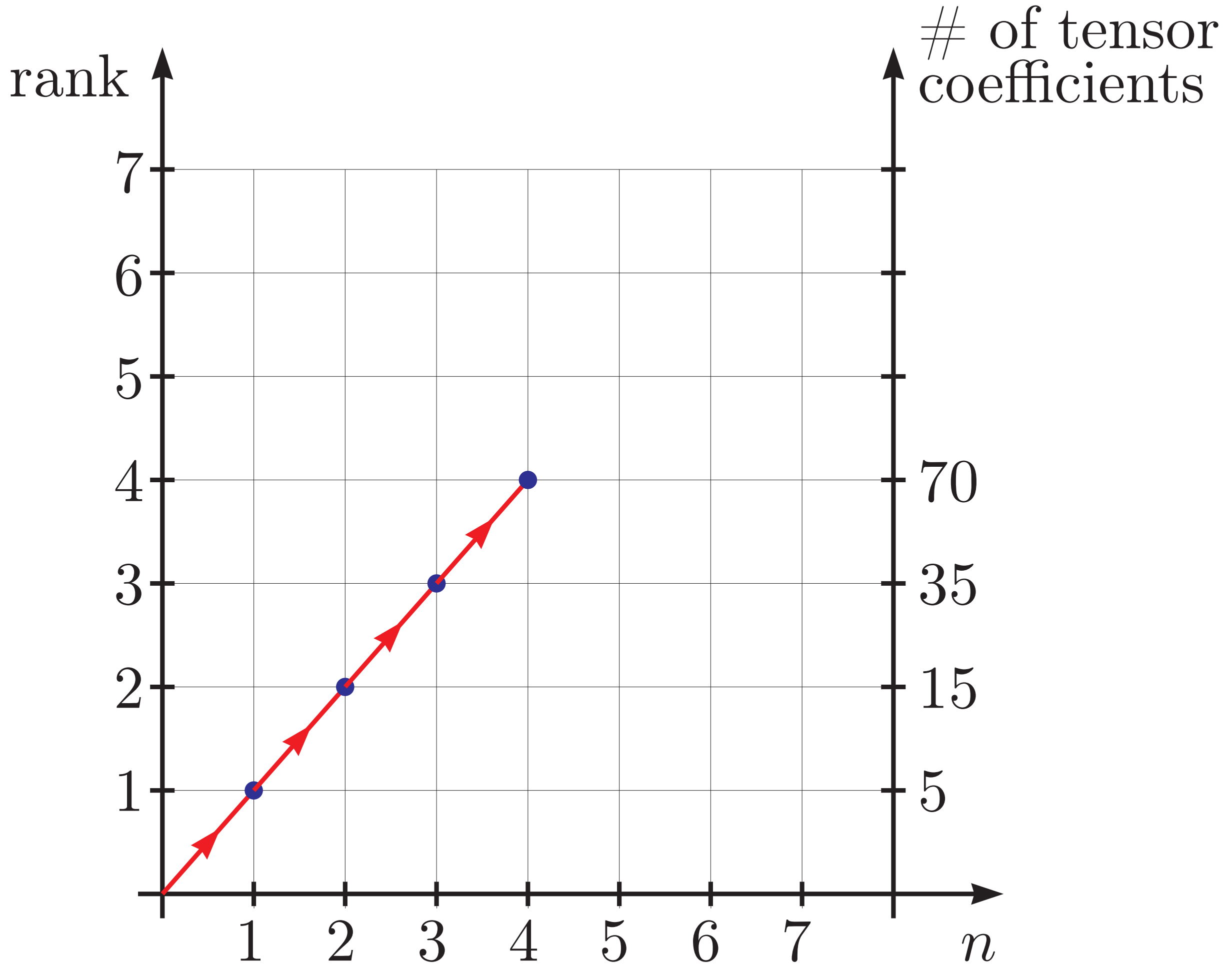


n: # of attached external legs

Amplitude generation and tensor reduction in OpenLoops 1

Example:

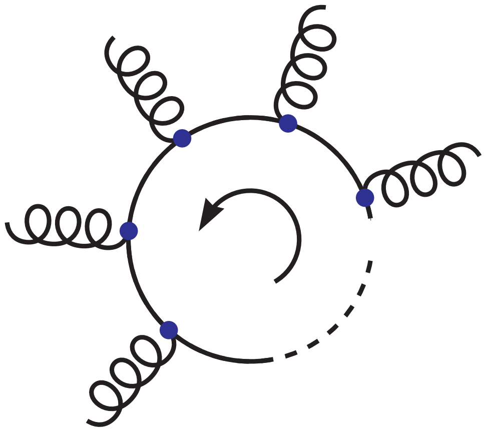


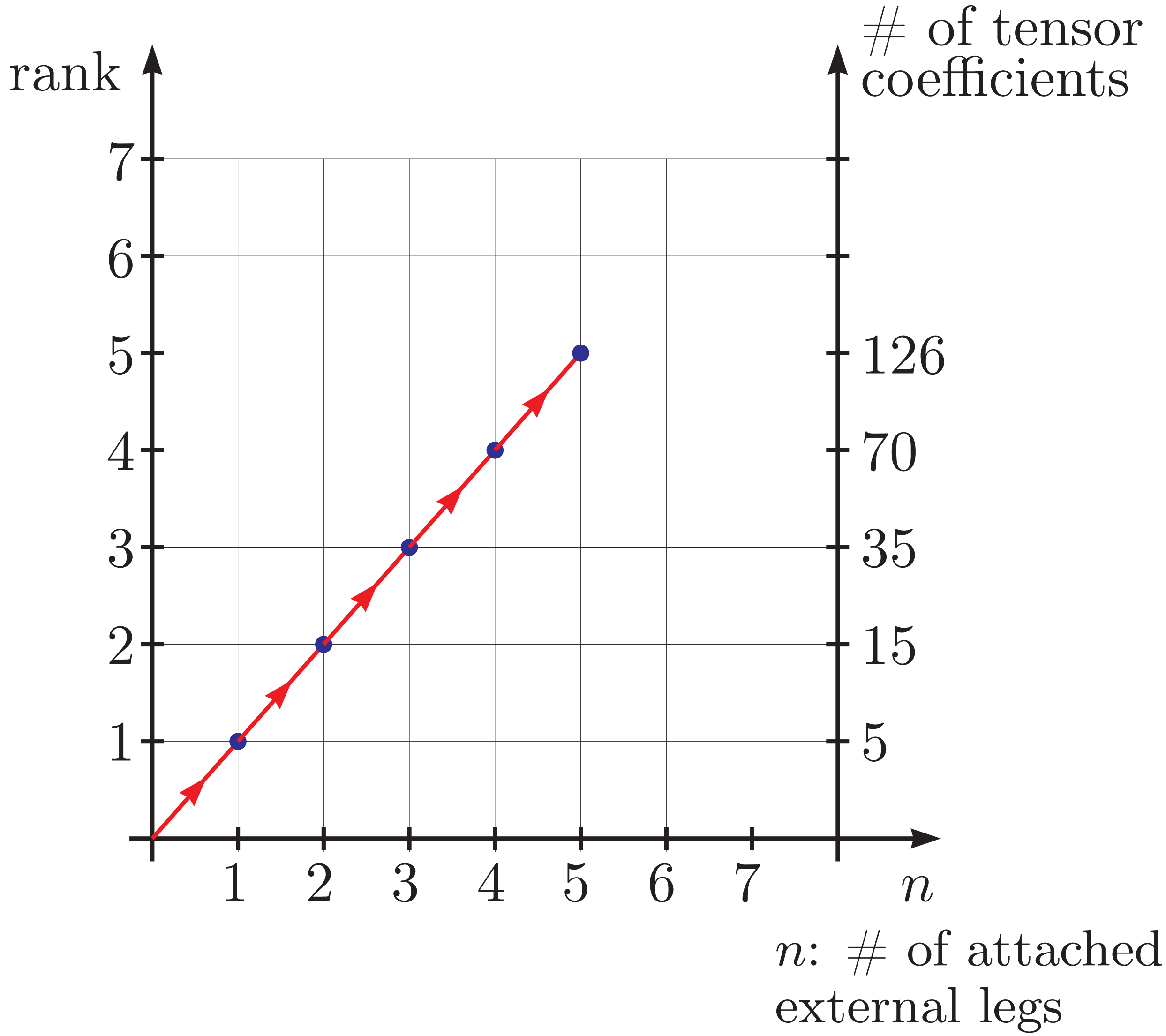


n: # of attached external legs

Amplitude generation and tensor reduction in OpenLoops 1

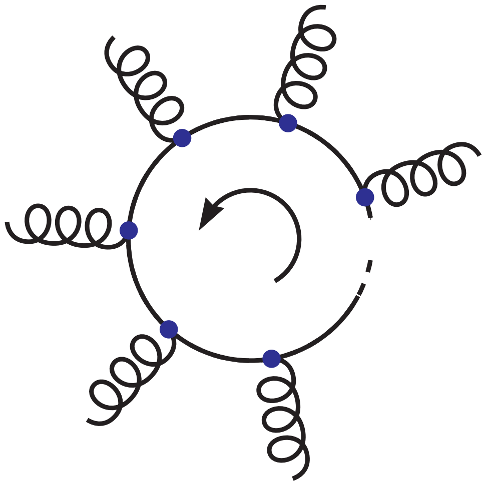
Example:

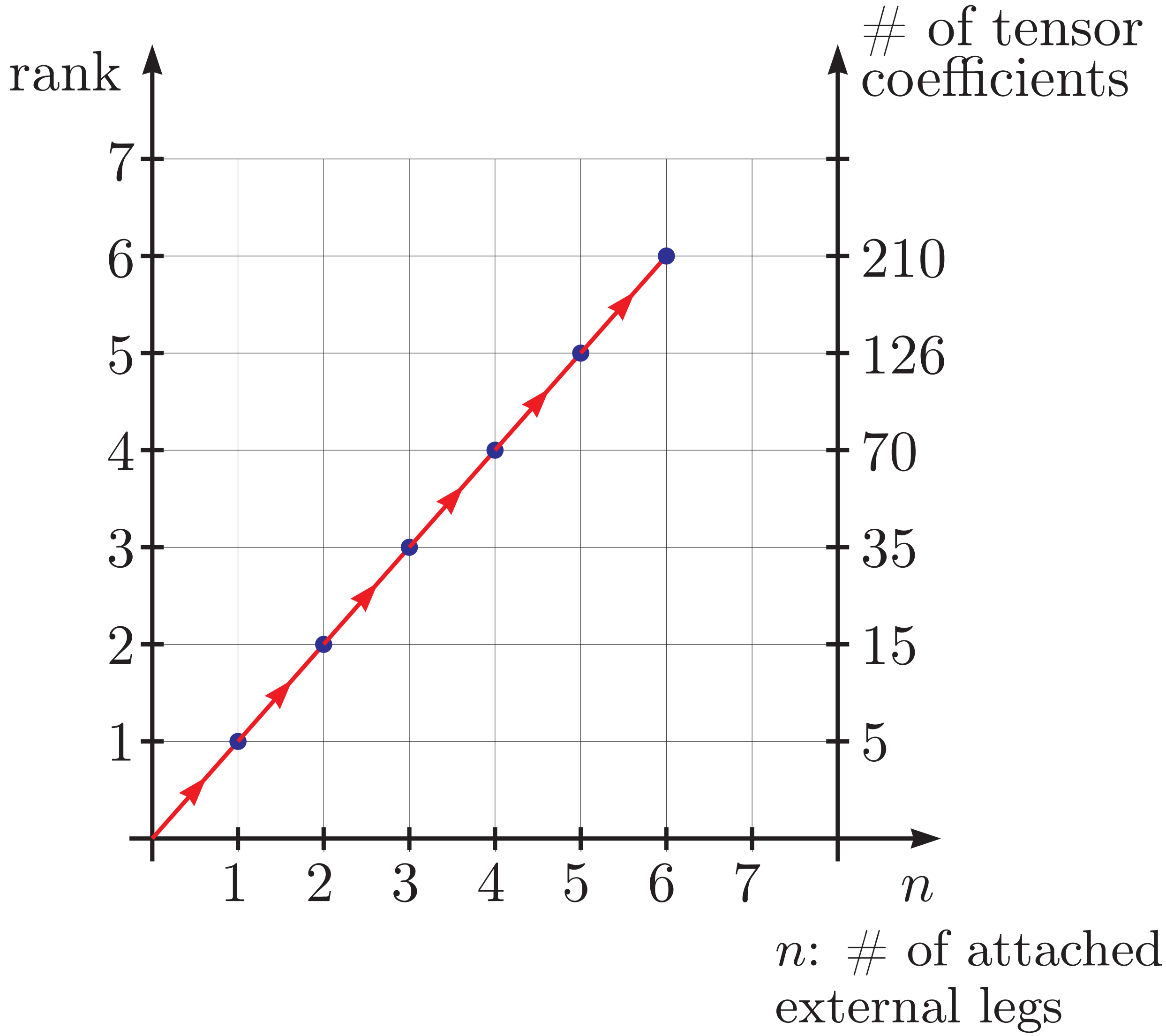




Amplitude generation and tensor reduction in OpenLoops 1

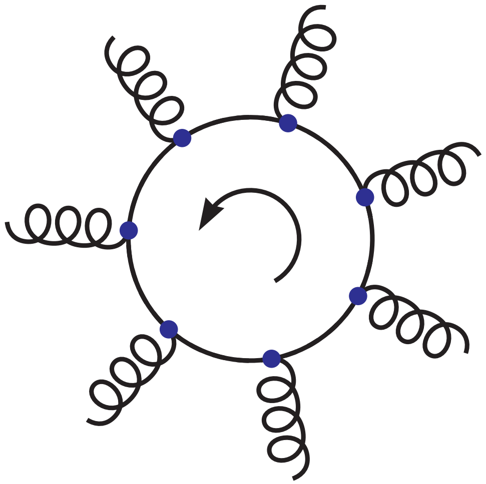
Example:



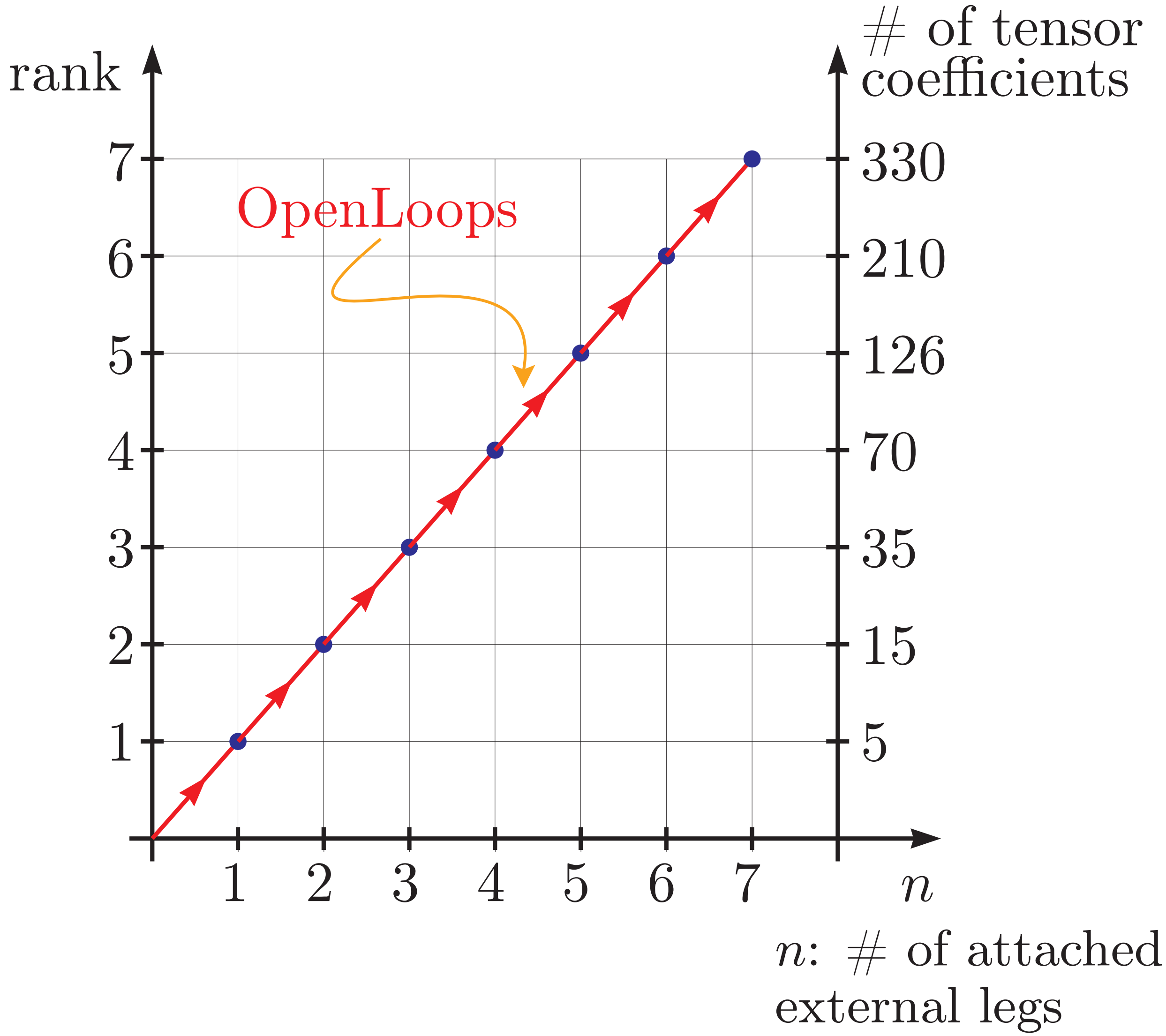


Amplitude generation and tensor reduction in OpenLoops 1

Example:

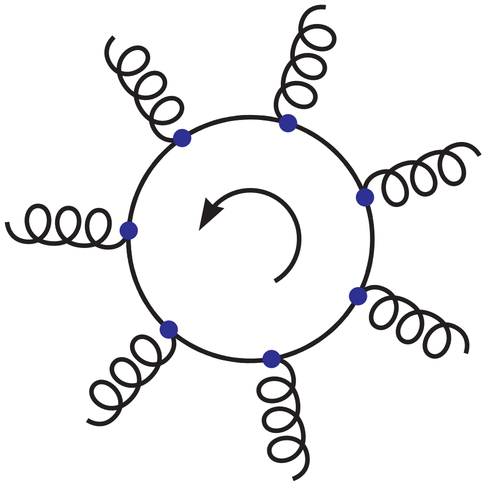


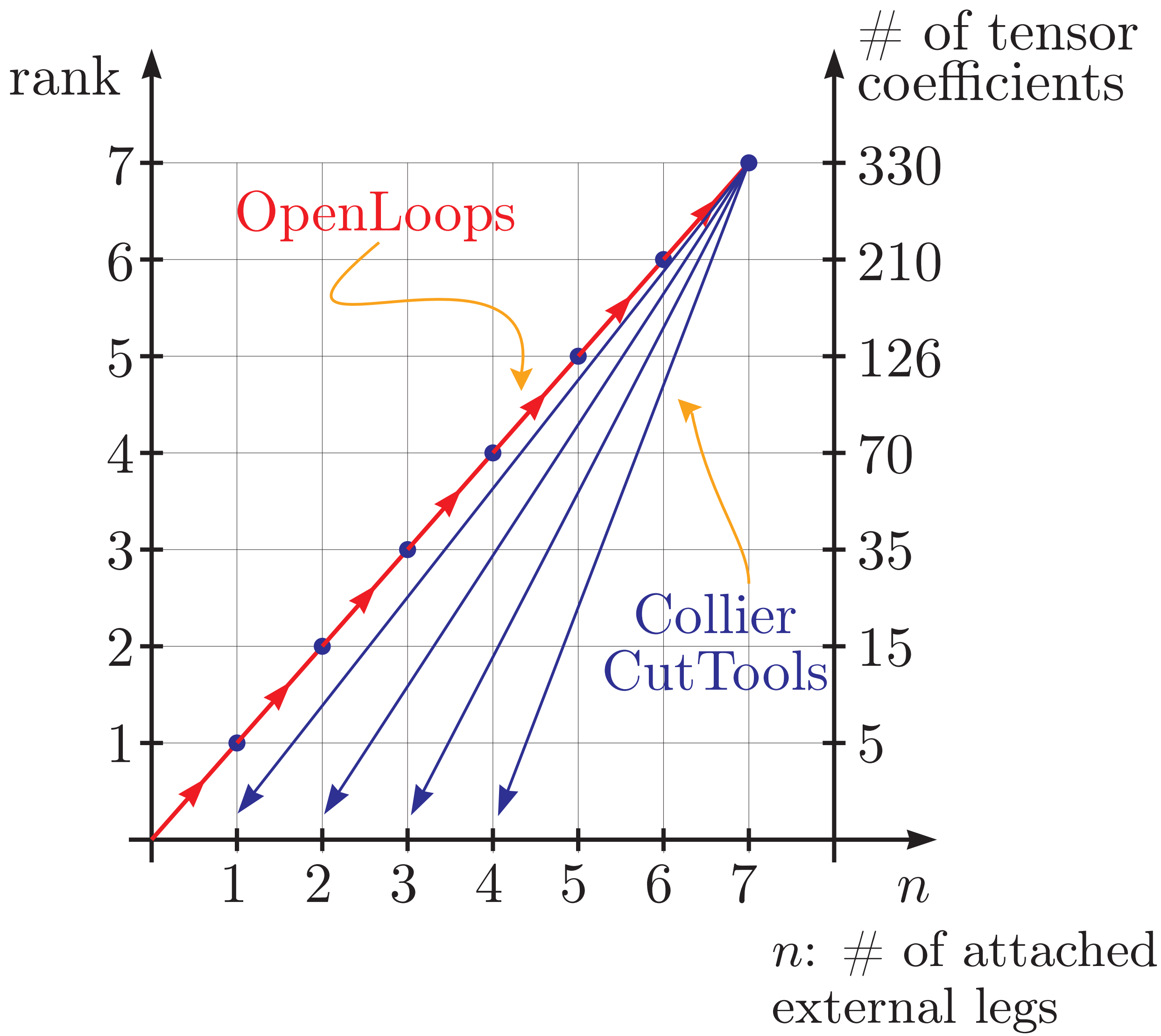
complexity grows exponentially with tensor rank



Amplitude generation and tensor reduction in OpenLoops 1

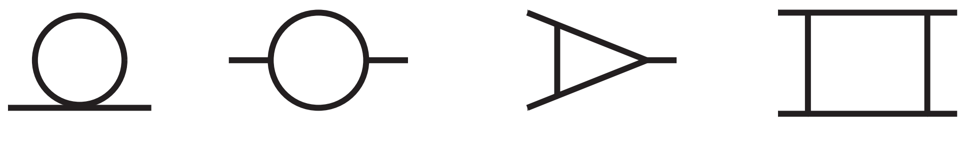
Example:





complexity grows exponentially with tensor rank

Numerical tensor integral reduction to scalar MI



III. On-the-fly Reduction

Use reduction identities valid at integrand level [del Aguila, Pittau '05]

$$\begin{aligned} q^\mu q^\nu &= A^{\mu\nu} + B^{\mu\nu}_{\lambda} q^\lambda \\ &= \left[A^{\mu\nu}_{-1} + A^{\mu\nu}_{0} D_0\right] + \left[B^{\mu\nu}_{-1,\lambda} + \sum_{i=0}^{3} B^{\mu\nu}_{i,\lambda} D_i\right] q^\lambda, \qquad D_i = (q+p_i)^2 - m_i^2 \end{aligned}$$

in order to reduce the factorized open loop integrand:

$$\frac{\mathcal{V}_N(q)}{D_0 \cdots D_N} = \frac{S_1(q) S_2(q) \cdots S_n(q) \cdots S_N(q)}{D_0 D_1 D_2 D_3 \cdots D_{N-1}}$$

III. On-the-fly Reduction

Use reduction identities valid at integrand level [del Aguila, Pittau '05]

$$
\begin{aligned}
q^\mu q^\nu &= A^{\mu\nu} + B^{\mu\nu}_\lambda q^\lambda \\
&= \left[A^{\mu\nu}_{-1} + A^{\mu\nu}_0 D_0\right] + \left[B^{\mu\nu}_{-1,\lambda} + \sum_{i=0}^{3} B^{\mu\nu}_{i,\lambda} D_i\right] q^\lambda, \qquad D_i = (q+p_i)^2 - m_i^2
\end{aligned}
$$

in order to reduce the factorized open loop integrand:

$$
\frac{\mathcal{V}_N(q)}{D_0 \cdots D_N} = \frac{S_1(q) S_2(q) \cdots S_n(q) \cdots S_N(q)}{D_0 D_1 D_2 D_3 \cdots D_{N-1}}
$$

integrand reduction applicable after n steps $\forall n \geq 2$ (independently of future steps!)

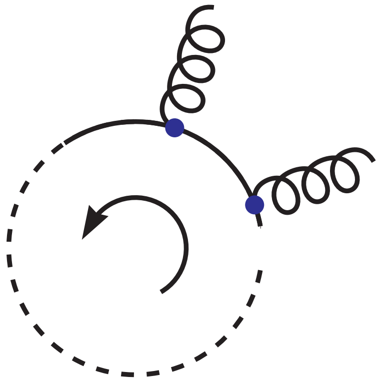
$$
\Rightarrow \quad \frac{\mathcal{V}^{\mu\nu} q_\mu q_\nu}{\bar{D}_0 \cdots \bar{D}_{N-1}} = \frac{\mathcal{V}^\mu_{-1} q_\mu + \mathcal{V}_{-1}}{\bar{D}_0 \cdots \bar{D}_{N-1}} + \sum_{i=0}^{3} \frac{\mathcal{V}^\mu_i q_\mu + \mathcal{V}_i}{\bar{D}_0 \cdots \bar{D}_{i-1} \bar{D}_{i+1} \cdots \bar{D}_{N-1}}
$$

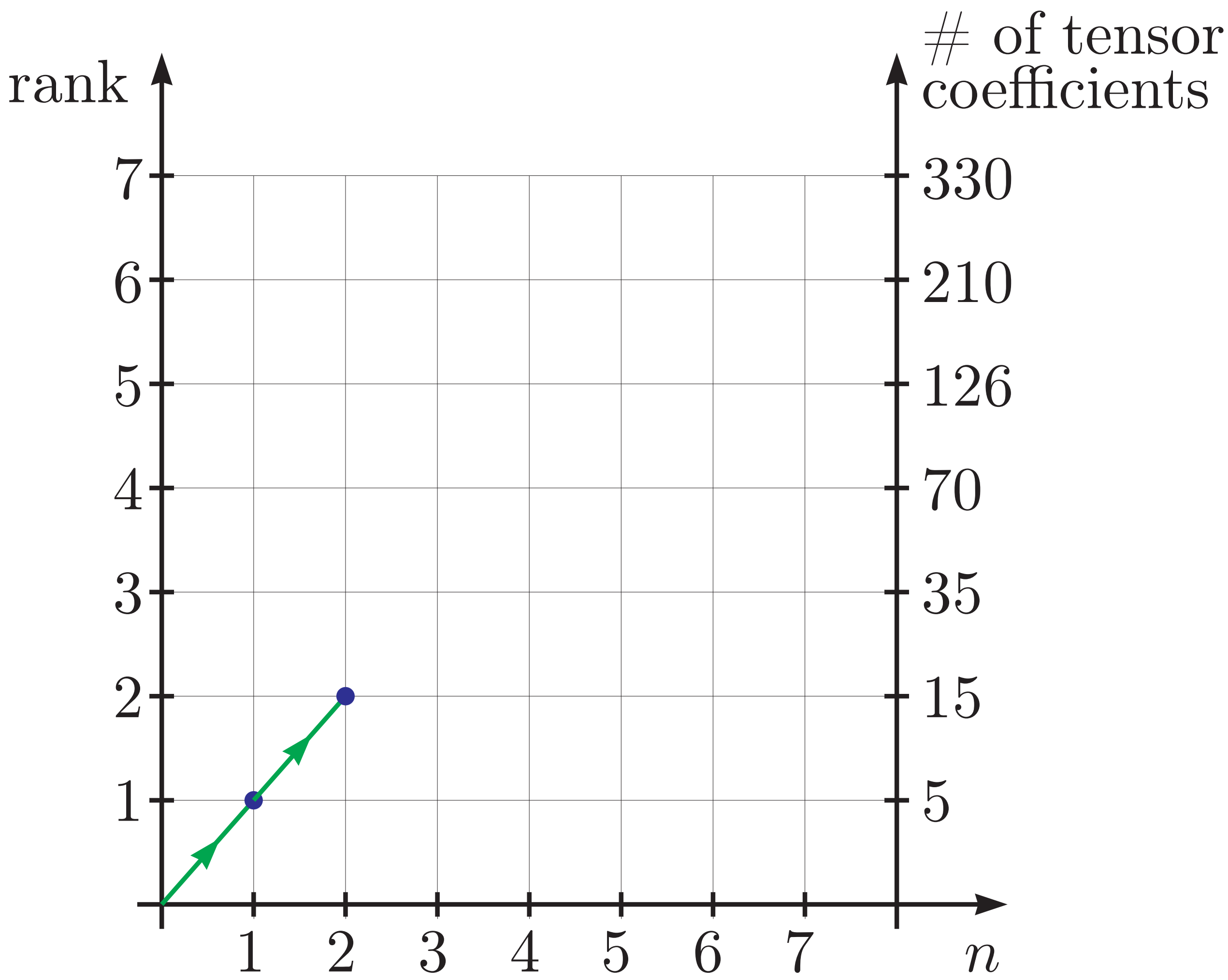
- q-dependence reconstructed in terms of 4 propagators $\Rightarrow$ new topologies with pinched propagators
- $A^{\mu\nu}, B^{\mu\nu}_\lambda$ depend on external momenta p_1, p_2, p_3

$\Rightarrow$ Compute with momentum space basis $l_1^\mu = p_1^\mu - \alpha_1 p_2^\mu, \quad l_2^\mu = p_2^\mu - \alpha_2 p_1^\mu, \quad l_3, l_4 \perp l_1, l_2, \quad l_i^2 = 0$

Amplitude generation and tensor reduction in OpenLoops 2

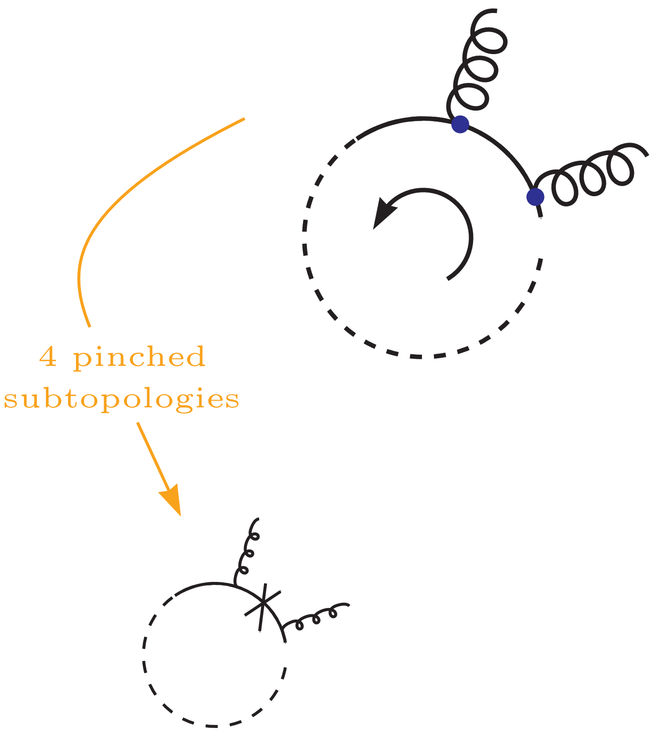
Example:

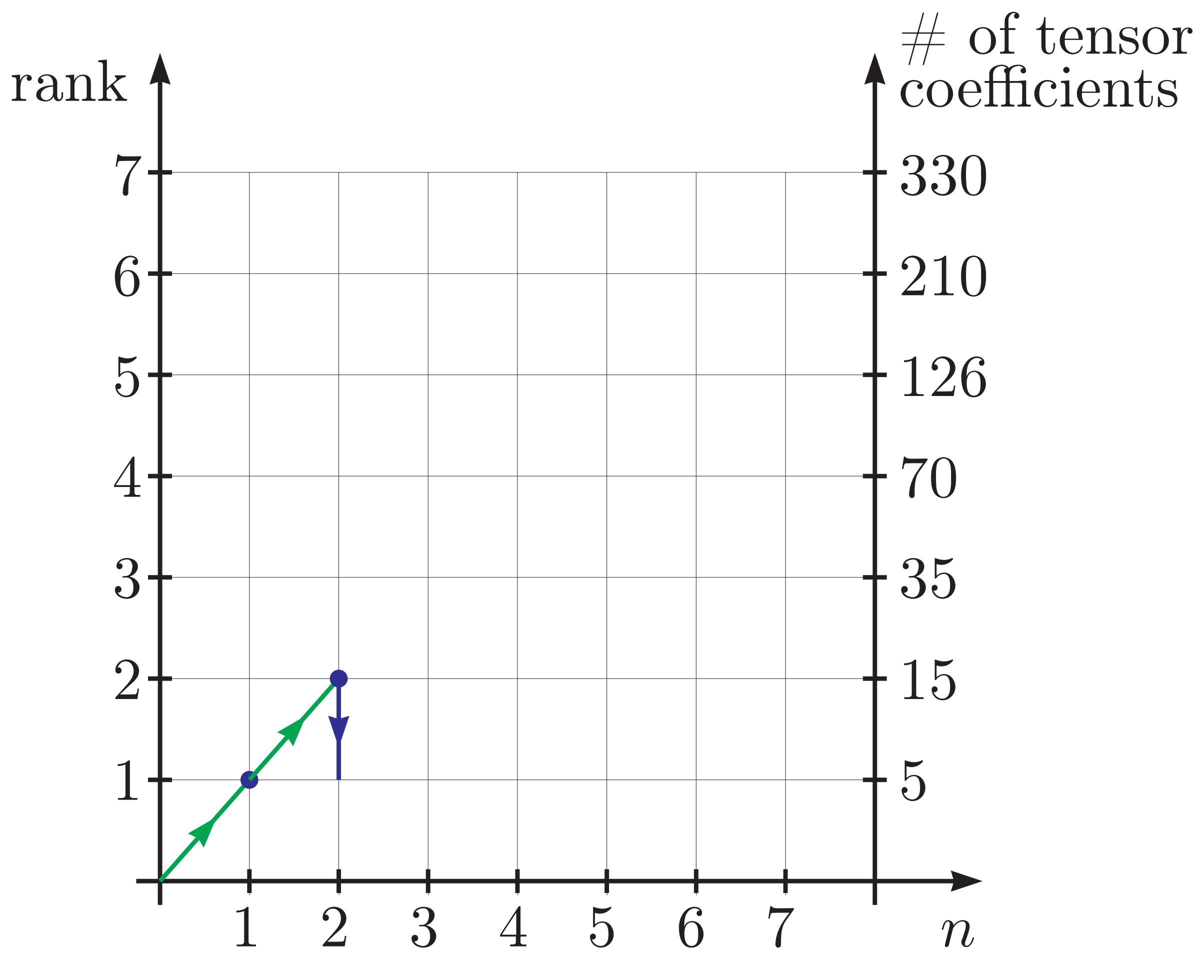




Amplitude generation and tensor reduction in OpenLoops 2

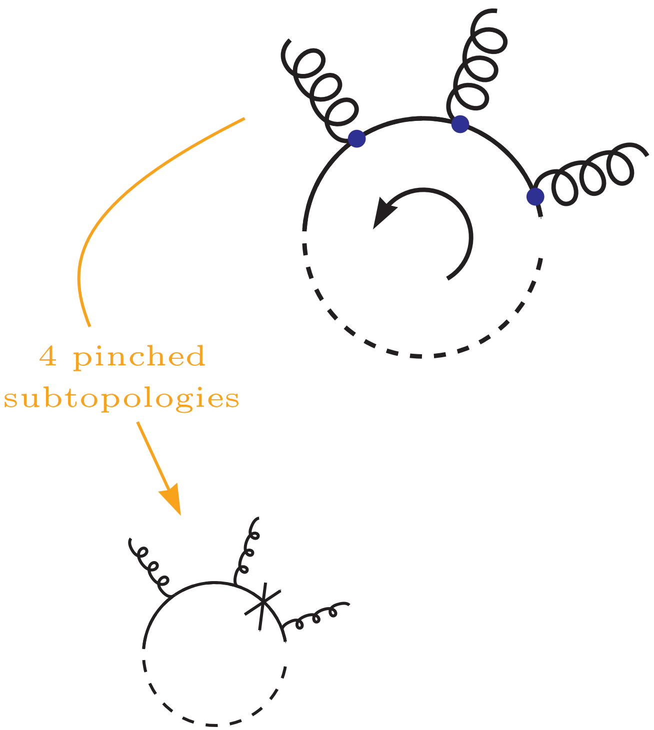
Example:

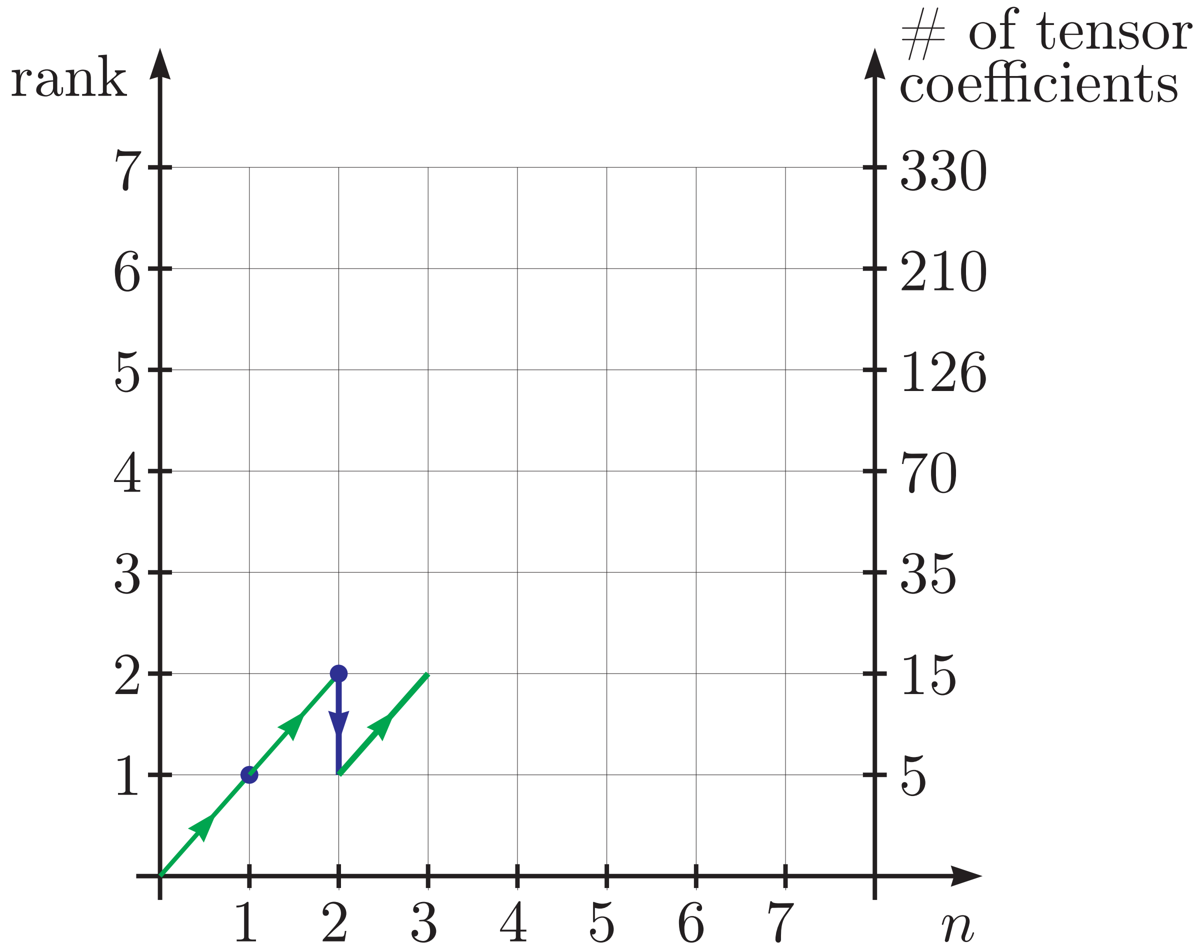




Amplitude generation and tensor reduction in OpenLoops 2

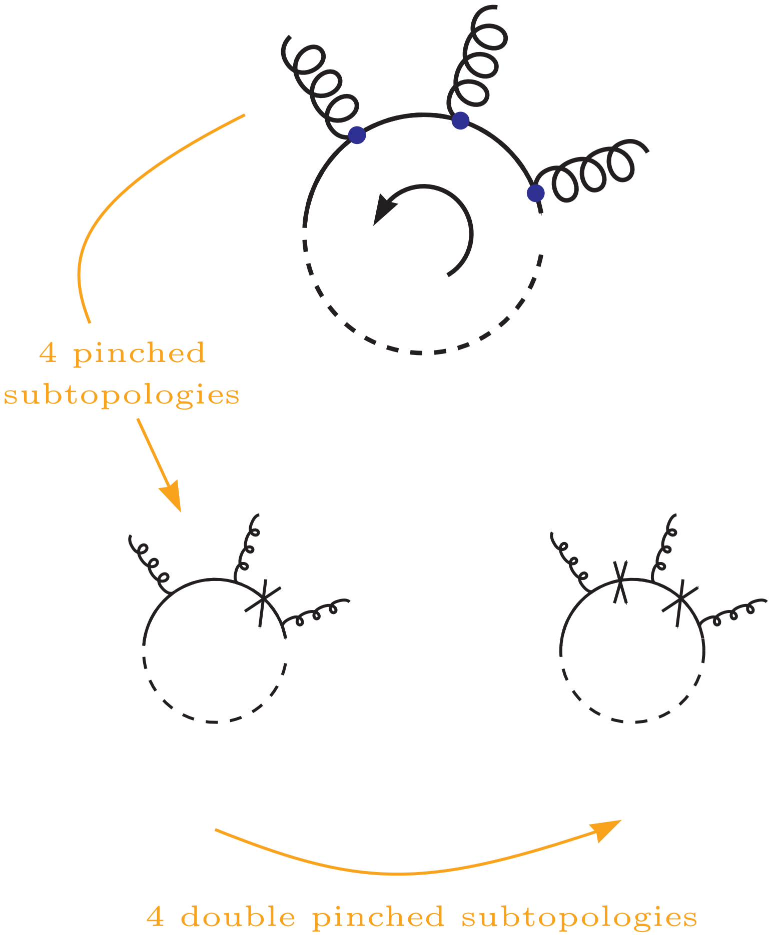
Example:

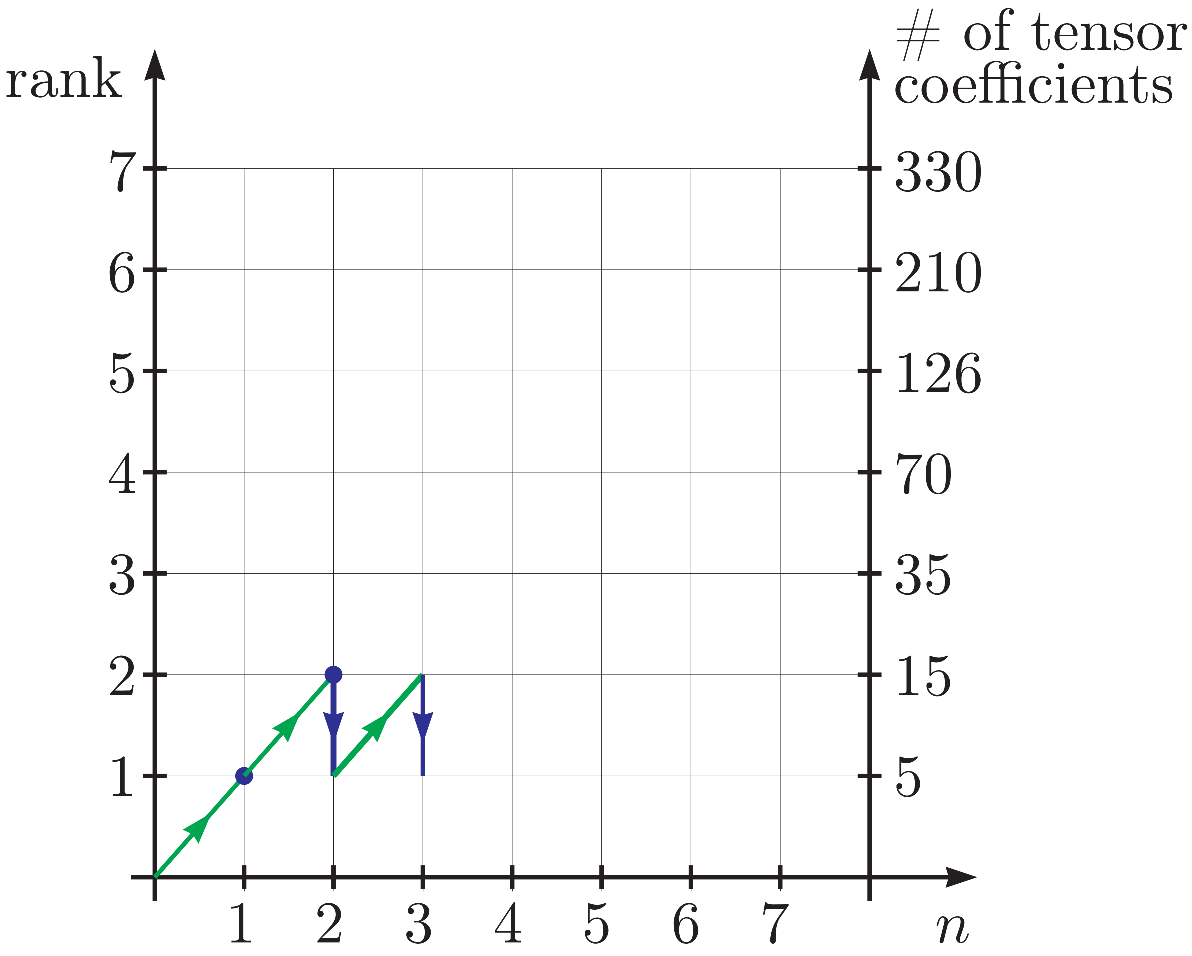




Amplitude generation and tensor reduction in OpenLoops 2

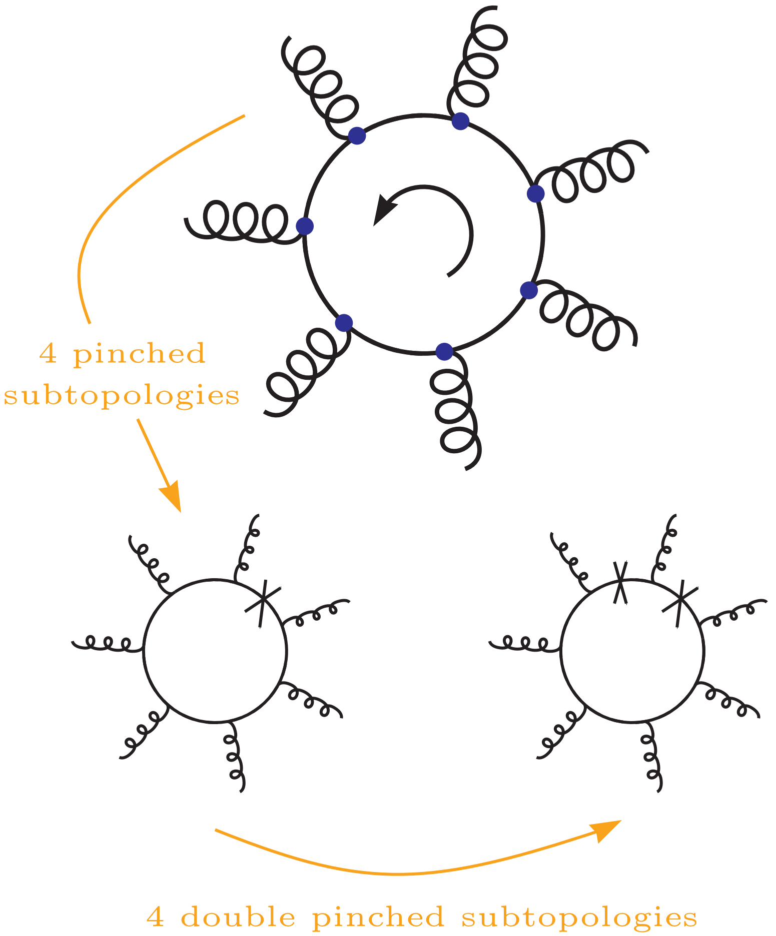
Example:

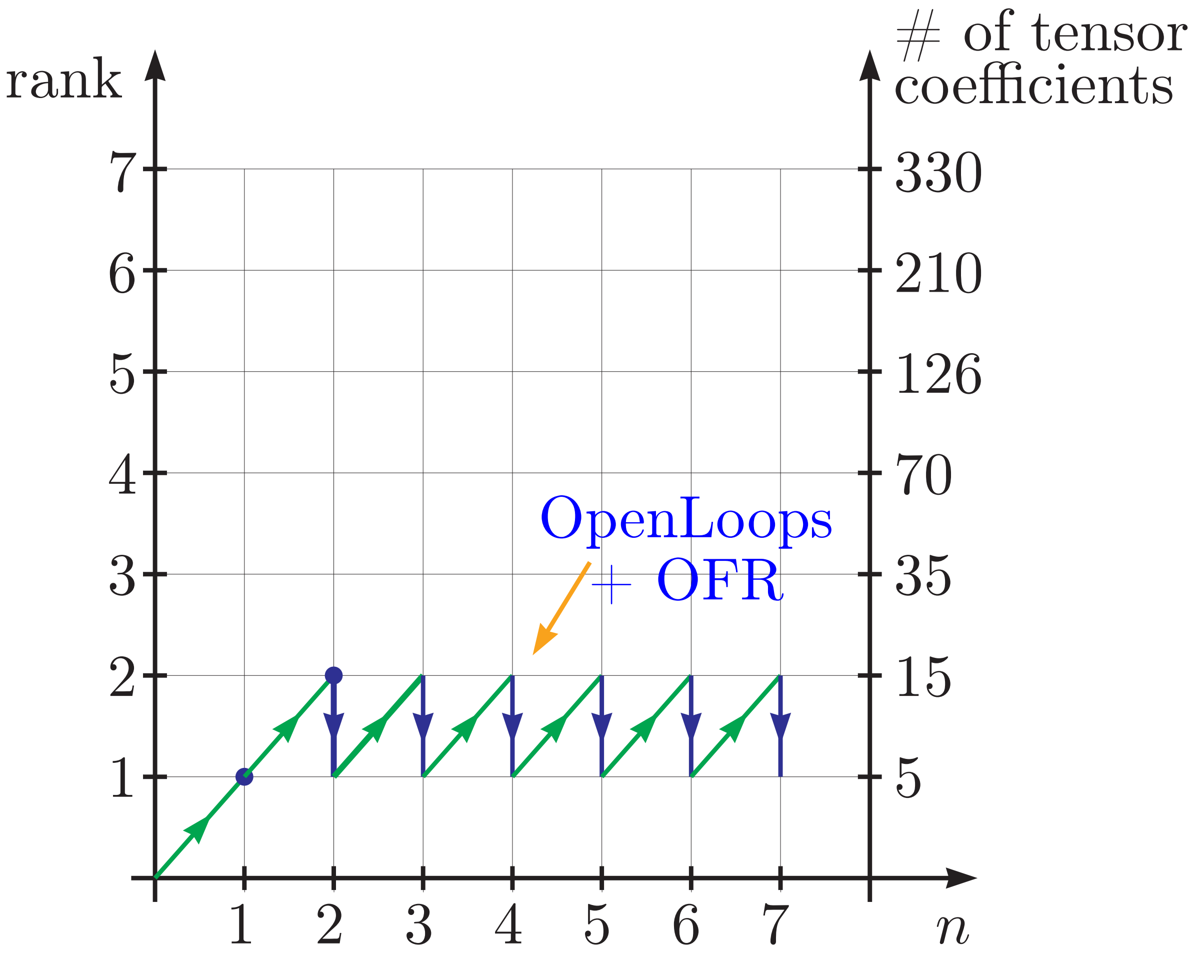




Amplitude generation and tensor reduction in OpenLoops 2

Example:

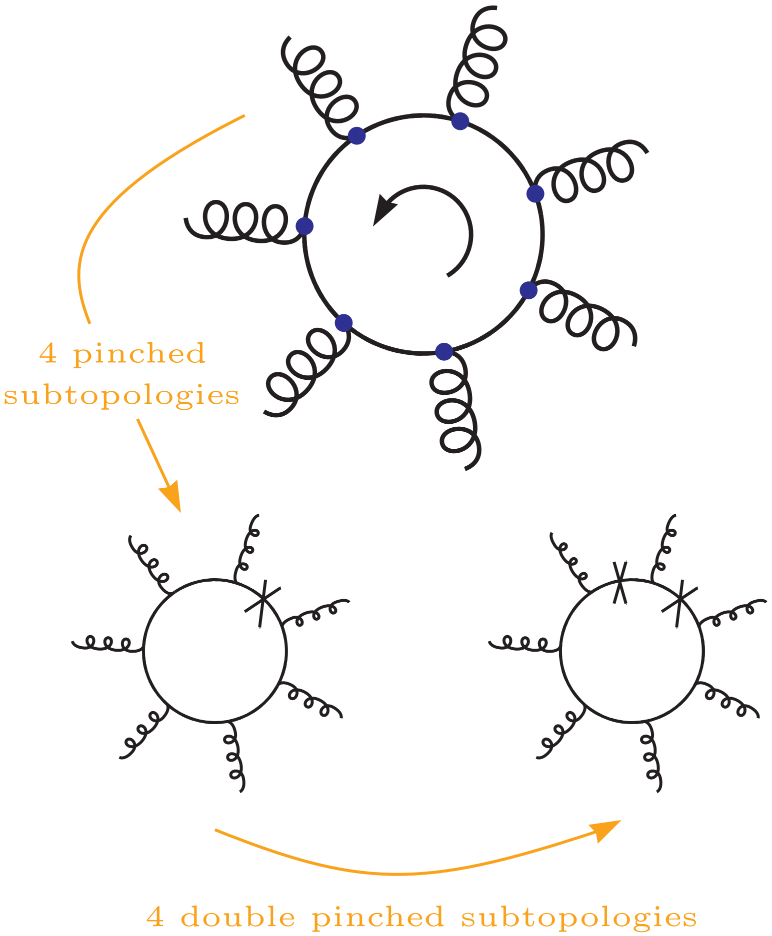


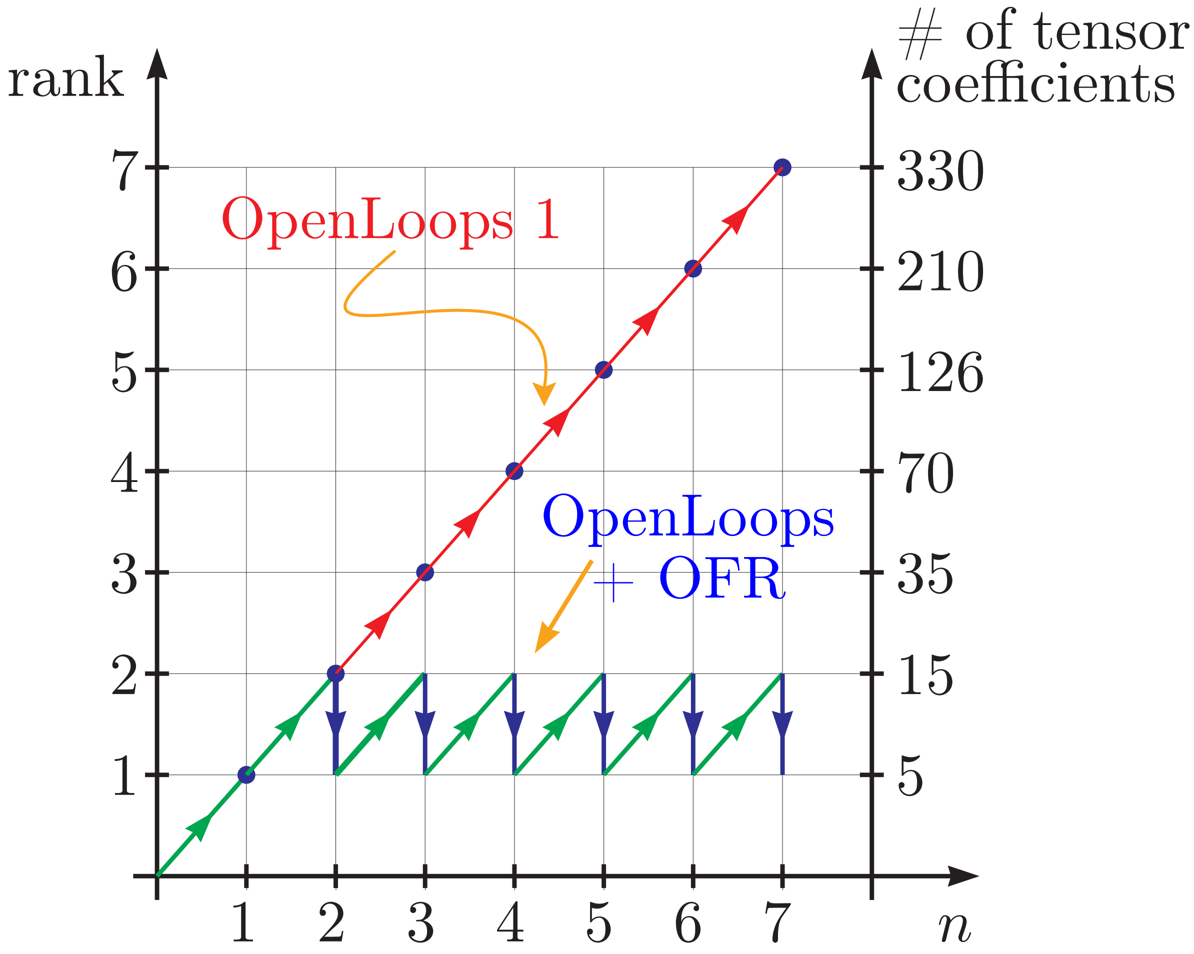


complexity associated with tensor rank remains small!

Amplitude generation and tensor reduction in OpenLoops 2

Example:

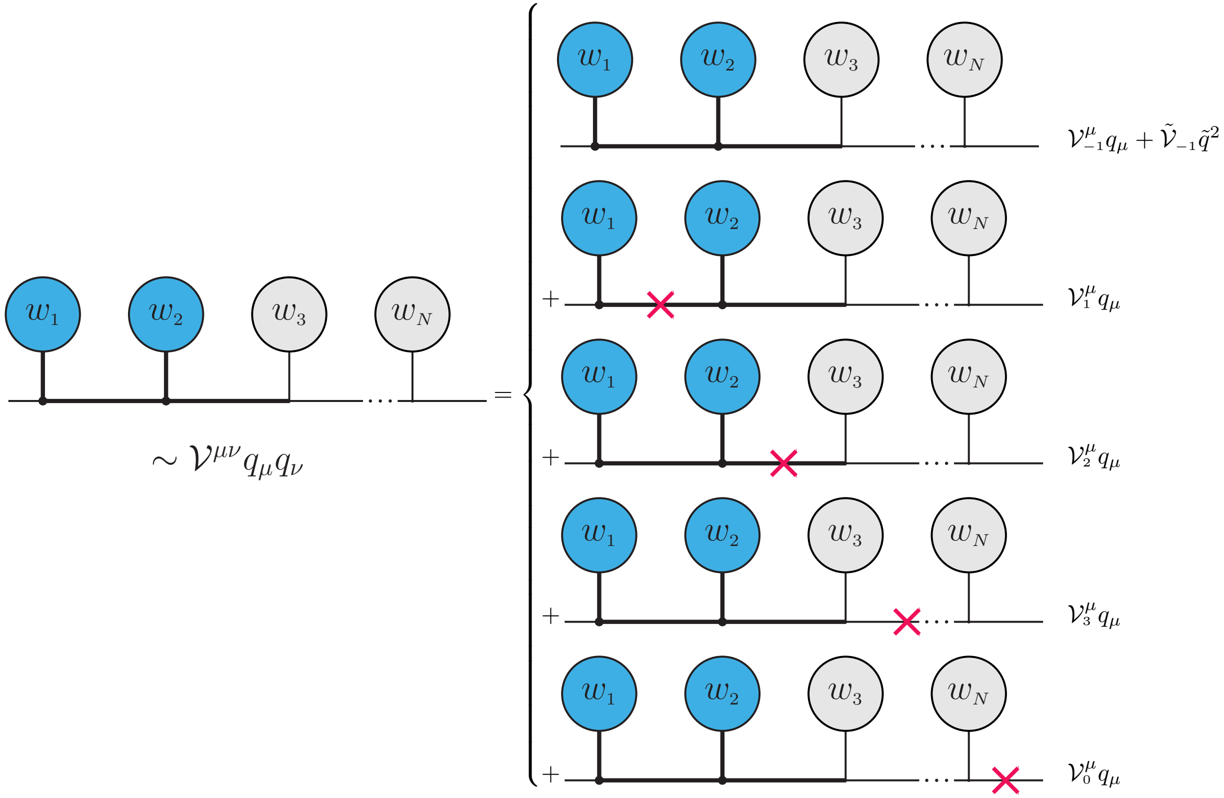




complexity associated with tensor
rank remains small!

Problem: huge proliferation of topologies due to pinching of propagators

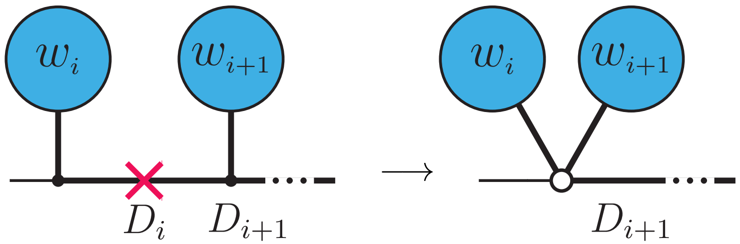
$$\Rightarrow \frac{\mathcal{V}_{\mu\nu}\, q^\mu q^\nu}{\bar{D}_0 \cdots \bar{D}_{N-1}} = \left[\underbrace{\left(\mathcal{V}^\mu_{-1} + \sum_{i=0}^{3} \mathcal{V}^\mu_i \bar{D}_i \right) q_\mu}_{\text{rank 1}} + \underbrace{\mathcal{V}_{-1} + \mathcal{V}_0 \bar{D}_0}_{\text{rank 0}} + \underbrace{\tilde{\mathcal{V}}_{-1} \tilde{q}^2}_{\text{rational term}} \right] \frac{1}{\bar{D}_0 \cdots \bar{D}_{N-1}}$$



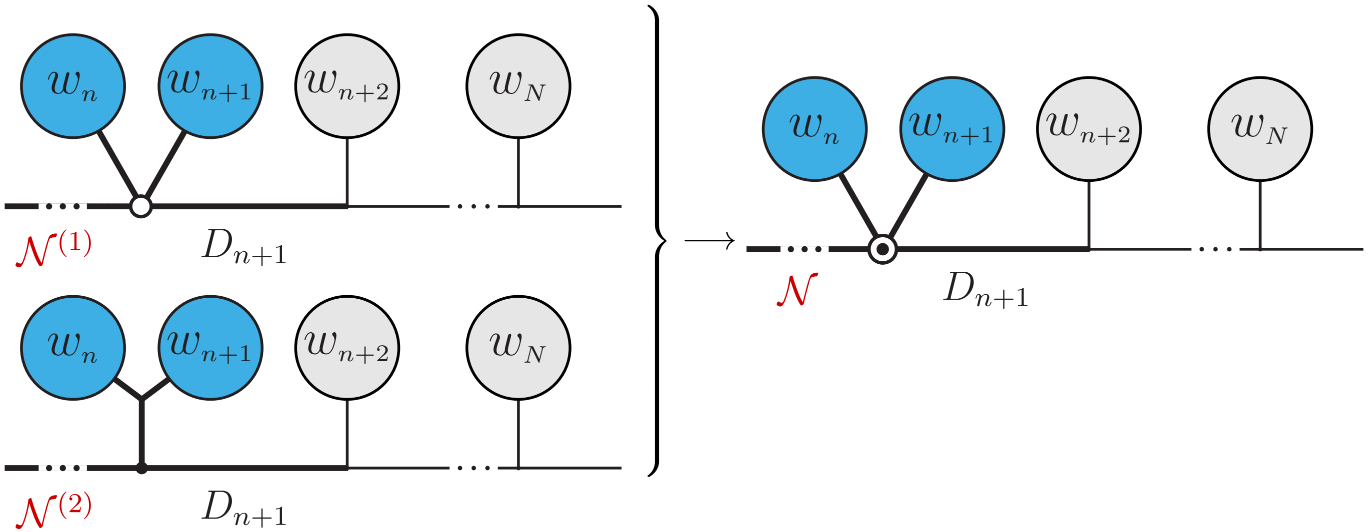
$\Rightarrow$ factor ~ 5 higher complexity after each reduction step!

Solution: OpenLoops Merging

- Contract pinched propagator between dressed segments



- Merge with all (pinched and unpinched) diagrams with same topology and undressed segments

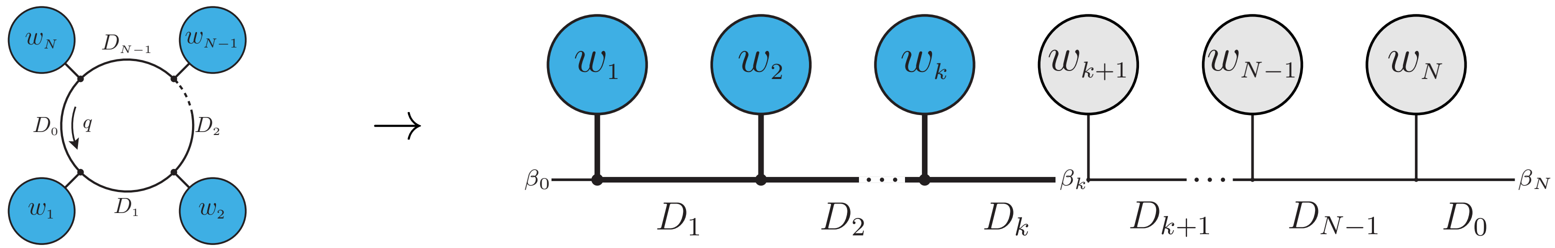


- No extra cost for pinched topologies after merging

- Algorithm:
  - Start with highest point diagrams $\rightarrow$ merging with lower point diagrams
  - OpenLoops 2 recursion step: dress one segment $\rightarrow$ reduce if necessary $\rightarrow$ merge

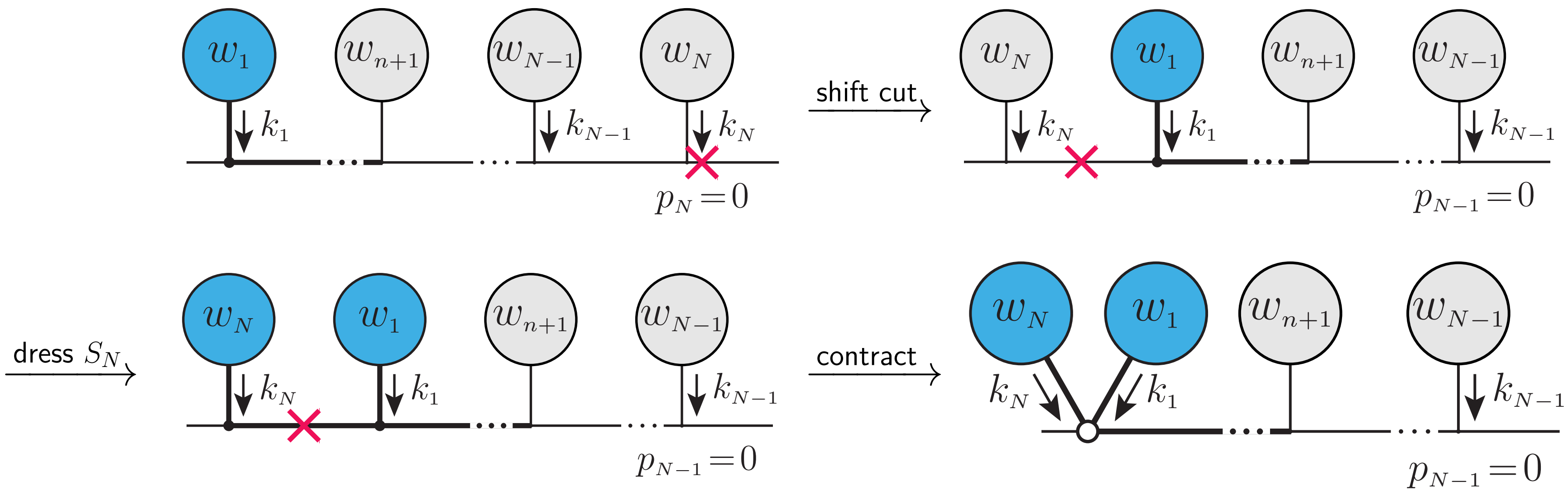
Technicalities

- Important: Cutting rule , i.e. choice of $\bar{D}_0$.



$\Rightarrow$ One specific external particle always in w_1.
$\Rightarrow$ Unique rule for dressing direction based on external particles in w_2 and w_N.

- Treatment of pinches of $\bar{D}_0 = (q^2 - m_0^2)$ $\qquad$ $(p_0 = p_N = 0)$



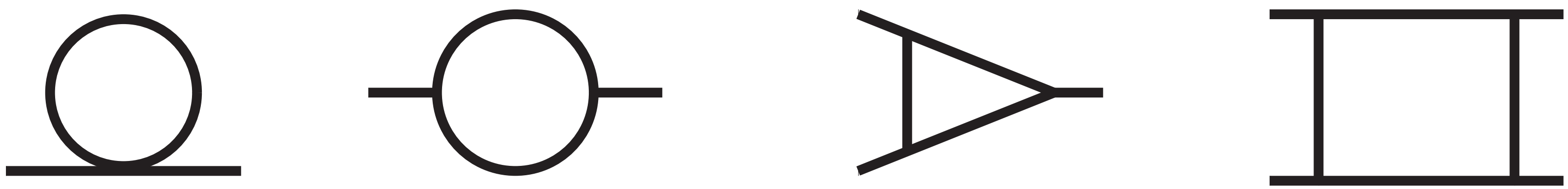
Final integral reduction

- reduce bubbles, rank-1 triangles and boxes with integral level identities [del Aguila, Pittau '05]

- reduce rank-1 and rank-0 integrals with $N \geq 5$ propagators to scalar boxes via simple OPP relations [Ossola, Papadopoulos, Pittau '07]

$$\frac{\mathcal{V} + \mathcal{V}_\mu q^\mu}{\bar{D}_0 \bar{D}_1 \cdots \bar{D}_{N-1}} = \sum_{i_0 < i_1 < i_2 < i_3}^{N-1} \frac{d(i_0 i_1 i_2 i_3)}{\bar{D}_{i_0} \bar{D}_{i_1} \bar{D}_{i_2} \bar{D}_{i_3}}$$

- use Collier 1.2 [Denner, Dittmaier, Hofer '16] for scalar boxes, triangles, bubbles, tadpoles



IV. Numerical Stability

$$q^\mu q^\nu \;=\; \left[A^{\mu\nu}_{-1} + A^{\mu\nu}_0 D_0\right] + \left[B^{\mu\nu}_{-1,\lambda} + \sum_{i=0}^{3} B^{\mu\nu}_{i,\lambda} D_i\right] q^\lambda$$

$A^{\mu\nu}_i, B^{\mu\nu}_{i,\lambda}$ computed from reduction basis $l_i(p_1, p_2)$ with $i = 1, 2, 3, 4$ and third momentum p_3

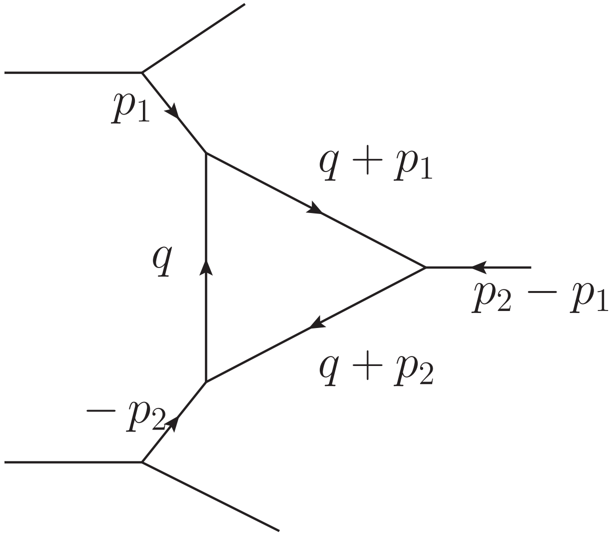
$$\begin{aligned} A^{\mu\nu}_i &= \frac{1}{\gamma} a^{\mu\nu}_i, \\ B^{\mu\nu}_{i,\lambda} &= \frac{1}{\gamma^2}\left[b^{(1)}_{i,\lambda}\right]^{\mu\nu} + \frac{1}{\gamma}\left[b^{(2)}_{i,\lambda}\right]^{\mu\nu} \end{aligned}$$

Severe numerical instabilities for $\gamma \propto \Delta(p_1, p_2) \to 0$

$\gamma = \gamma(p_1, p_2) = 4\frac{\Delta(p_1,p_2)}{p_1 p_2 \pm \sqrt{\Delta(p_1,p_2)}}$ with $\Delta = (p_1 p_2)^2 - p_1^2 p_2^2$

- Freedom to choose two momenta from p_1, p_2, p_3
  $\Rightarrow$ maximize γ in on-the-fly reduction with $N \geq 4$ propagators.
  $\Rightarrow$ avoid small Gram determinants until triangle reduction
- For $N = 3$: identify problematic kinematic configurations and use targeted expansions.

Problematic kinematic configuration: t-channel diagrams with



$$
\begin{aligned}
p_1^2 &= -p^2 < 0, \\
p_2^2 &= -p^2(1+\delta), \qquad 0 \le \delta \ll 1, \\
(p_2-p_1)^2 &= 0, \\
\Rightarrow \sqrt{\Delta} &= \frac{p^2}{2}\delta \\
\Rightarrow \gamma &= -p^2\delta^2
\end{aligned}
$$

$\Rightarrow$ expand basis momenta l_i, reduction formula and scalar integrals in δ, e.g. massless rank 1:

$$
\begin{aligned}
C^\mu &= \frac{2}{\delta^2 p^2}\left\{B_0(-p^2,0,0)\left[-p_1^\mu(1+\delta)+p_2^\mu\right] + B_0\left(-p^2(1+\delta),0,0\right)\left[(p_1^\mu-p_2^\mu)(1+\delta)\right]\right\} \\
&\quad + \frac{1}{\delta}C_0\left(-p^2,-p^2(1+\delta),0,0,0\right)\left[-p_1^\mu(1+\delta)+p_2^\mu\right] \\
&= \frac{p_1^\mu+p_2^\mu}{2p^2}\left[-B_0(-p^2,0,0)+1\right] + \delta\,\frac{p_1^\mu+2p_2^\mu}{6p^2}\left[B_0(-p^2,0,0)\right] + \mathcal{O}(\delta^2)
\end{aligned}
$$

with $C_0(p_1,p_2,m_0,m_1,m_2) \sim \int \mathrm{d}^D q\,\frac{1}{\bar{D}_0\bar{D}_1\bar{D}_2}$ and $B_0(p_1,m_0,m_1) \sim \int \mathrm{d}^D q\,\frac{1}{\bar{D}_0\bar{D}_1}$

Implemented: direct expansions for the full reduction of rank ≤ 3 triangles to scalars for all relevant mass configurations up to and including $\mathcal{O}(\delta^2)$ [soon $\mathcal{O}(\delta^4)$].

CPU performance: OpenLoops 1 + Collier/Cuttools vs OpenLoops 2

Runtimes ($10^{-3}s$) per phase-space point

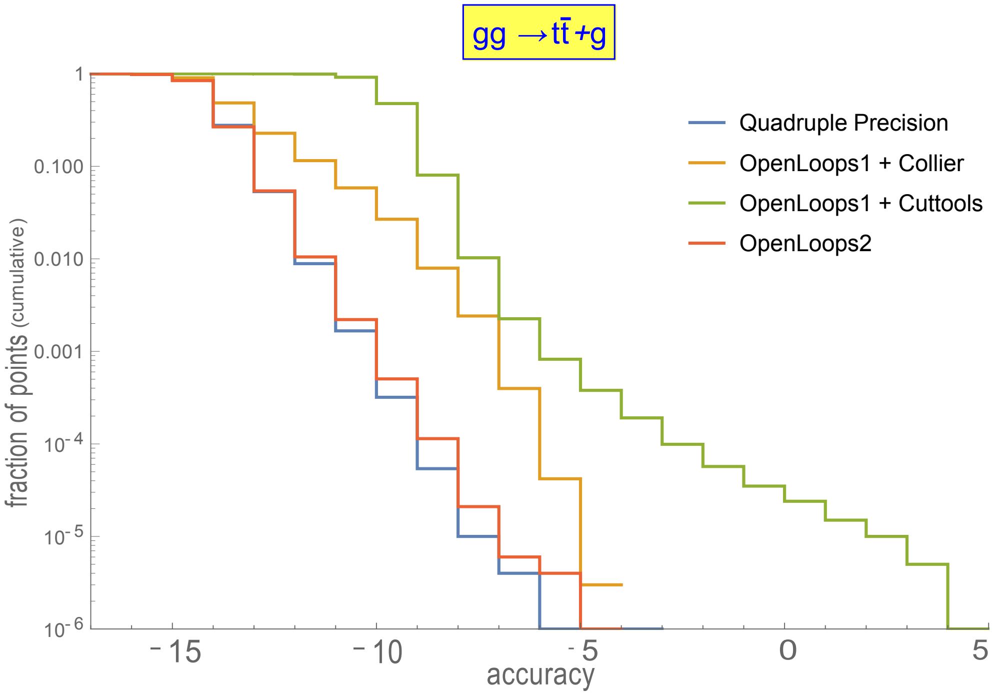
Last column: timing ratio between the fastest OL1+reduction library and OL2

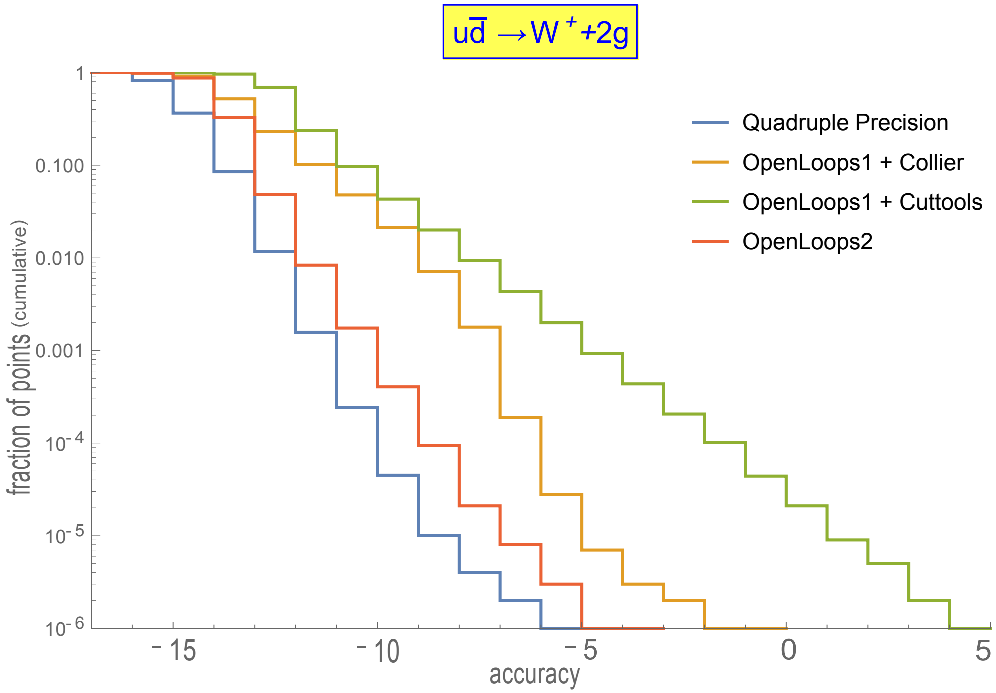
	OL1 (Collier)	OL1 (Cuttools)	OL2	OL1/OL2
$u\bar{u} \to t\,\bar{t}$	0.2355	0.4034	0.2385	0.99
$u\bar{u} \to t\,\bar{t}\,g$	4.259	7.066	3.490	1.2
$u\bar{u} \to t\,\bar{t}\,g\,g$	$1.154 \cdot 10^2$	$1.612 \cdot 10^2$	$0.7505 \cdot 10^2$	1.5
$gg \to t\,\bar{t}$	1.408	2.486	1.019	1.4
$gg \to t\,\bar{t}\,g$	35.03	50.23	22.93	1.5
$gg \to t\,\bar{t}\,g\,g$	$1.330 \cdot 10^3$	$1.519 \cdot 10^3$	$0.6010 \cdot 10^3$	2.2
$u\bar{d} \to W^+\,g$	0.2972	0.6274	0.3255	0.91
$u\bar{d} \to W^+\,g\,g$	5.690	11.30	5.222	1.1
$u\bar{d} \to W^+\,g\,g\,g$	$1.787 \cdot 10^2$	$2.380 \cdot 10^2$	$1.078 \cdot 10^2$	1.7
$u\bar{u} \to W^+\,W^-$	0.2622	0.4140	0.1756	1.5
$u\bar{u} \to W^+\,W^-\,g$	8.528	12.04	7.011	1.2
$u\bar{u} \to W^+\,W^-\,g\,g$	$2.441 \cdot 10^2$	$2.817 \cdot 10^2$	$1.278 \cdot 10^2$	1.9

Factor ~ 2 speedup wrt OpenLoops 1 for nontrivial processes!

Stability of OpenLoops 1 and 2 in double precision: $2 \to 3$ processes (at $\sqrt{\hat{s}} = 1$ TeV)

Probability of relative accuracy $\mathcal{A}$ or less (wrt OL1 + Cuttools in quad precision, 10^6 uniform random points)





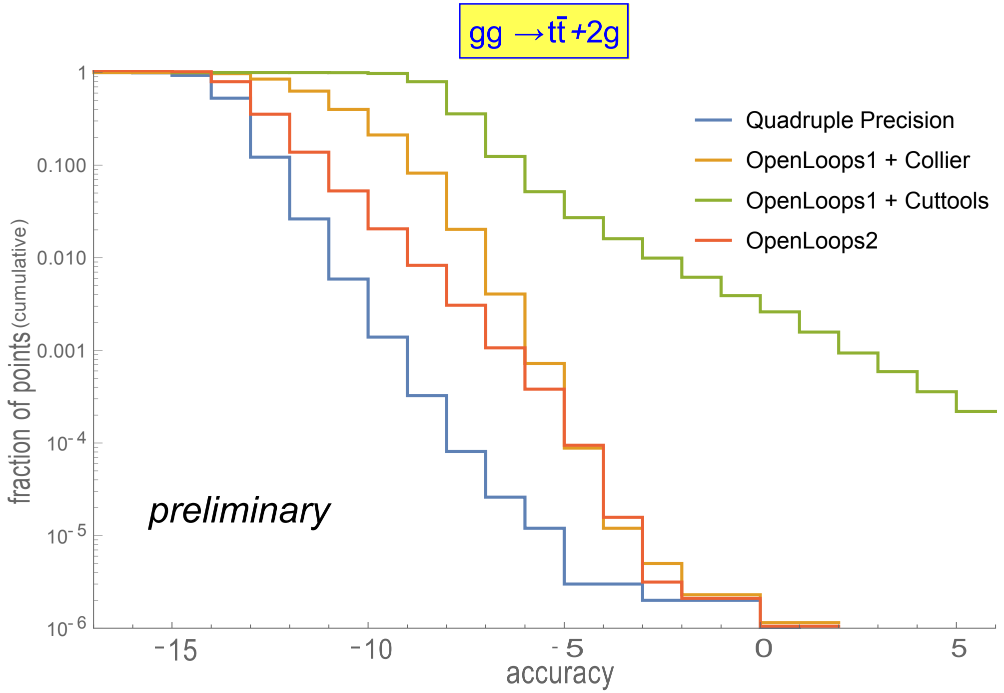
- Hard cuts: $p_T > 50\,\mathrm{GeV}$ and $\Delta R_{ij} => 0.5$ for final state QCD partons
  ($\Delta R_{ij} = \sqrt{(\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2}$, ϕ_i azimuthal angle, η_i rapidity)
- Behaviour in the tails crucial for real-life applications
- 1 to 3 orders of magnitude improvement wrt OL1 + Cuttols and Collier in DP

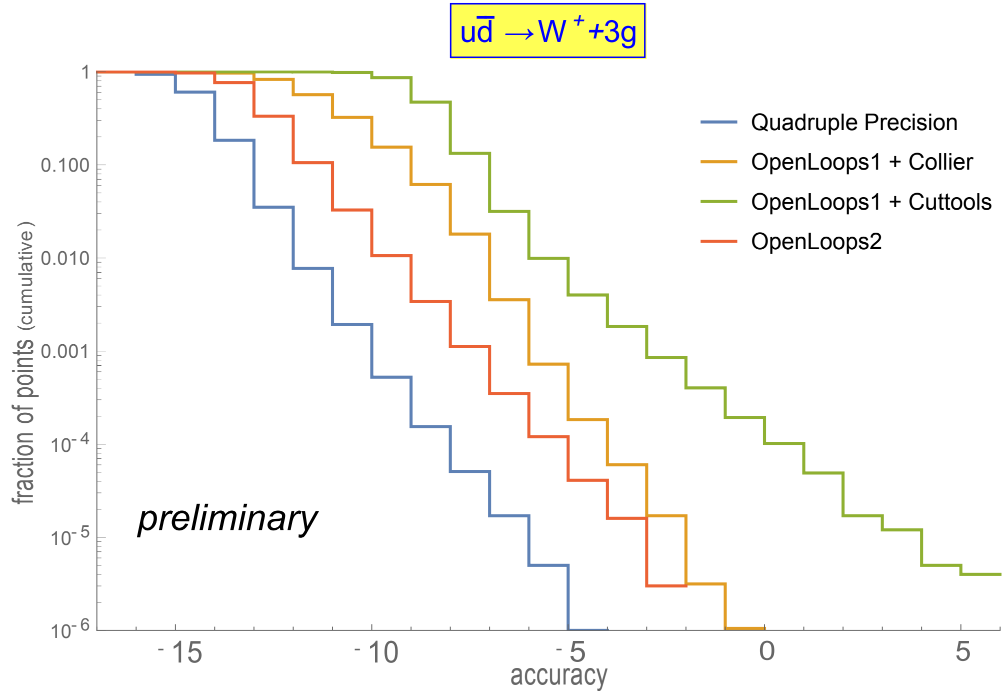
Excellent stability thanks to on-the fly reduction and minimal Δ-expansions

Soft region under investigation $\Rightarrow$ important for real-virtual part of NNLO

Stability of OpenLoops 1 and 2 in double precision: $2 \to 4$ processes (at $\sqrt{\hat{s}} = 1$ TeV)

Probability of relative accuracy $\mathcal{A}$ or less (wrt OL1 + Cuttools in quad precision, 10^6 uniform random points)





- Same hard cuts as for $2 \to 3$
- Orders of magnitude improvement wrt Cuttools and similar or better stability wrt Collier
- Further improvements in the tail under investigation

Very good stability thanks to on-the fly reduction and minimal Δ-expansions

V. Summary and Outlook

- New algorithm for construction and reduction of 1-loop ampitudes in a single recursion
- Drastic reduction of complexity at all stages of the calculation (rank ≤ 2)
- New colour and helicity treatment $+$ OpenLoops merging $\Rightarrow$ significant gain in CPU efficiency
- Same level of automation and same interface as OpenLoops 1
- Dedicated stability analysis possible in a single dressing and reduction tool
  $\Rightarrow$ Simple targeted expansions provide excellent numerical stability in the hard regions
- future projects:
  - improvement of stability in real-virtual NNLO contributions (soft region)
  - extension to 2 loops