Universität Zürich[UZH]

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

On-the-fly reduction of open loops and its applications

M. F. Zoller

in collaboration with F. Buccioni, J.-N. Lang, H. Zhang and S. Pozzorini

LoopFest 2018 - Michigan State University - 17 July 2018

Outline

I. Introduction: Numerical amplitude generation in OpenLoops

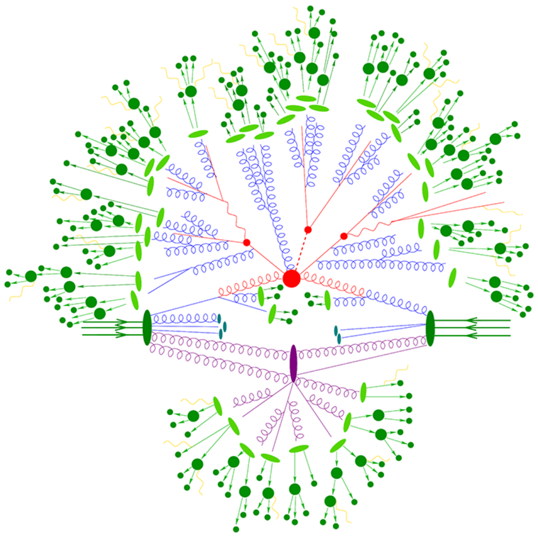
II. The on-the-fly method (see Eur. Phys. J. C **78** (2018) no.1, 70 [arXiv:1710.11452 [hep-ph]])

III. Treatment of numerical instabilities due to small Gram determinants

IV. Performance and numerical stability benchmarks

V. Summary and Outlook

I. Scattering amplitudes in OpenLoops



[Höche]

Monte-Carlo simulations of scattering events

[Sherpa, Powheg, Herwig, Whizard, Geneva, Munich, Matrix] require

- PDFs
- Hard scattering amplitudes $\rightarrow$ **OpenLoops**
- Parton shower, hadronisation model

OpenLoops: Fully automated numerical tool for tree and one-loop scattering probability densities

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

(h = helicity configuration)

- **OpenLoops 1** [Cascioli, Lindert, Maierhöfer, Pozzorini], available at openloops.hepforge.org
- **OpenLoops 2** [Buccioni, Lindert, Maierhöfer, Pozzorini, M.Z.], publication in preparation

▷ NLO QCD and NLO EW corrections fully implemented

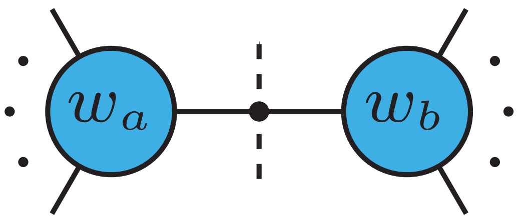
The OpenLoops framework

Amplitudes are sums of diagrams factorising into a colour factor and a colour-stripped amplitude

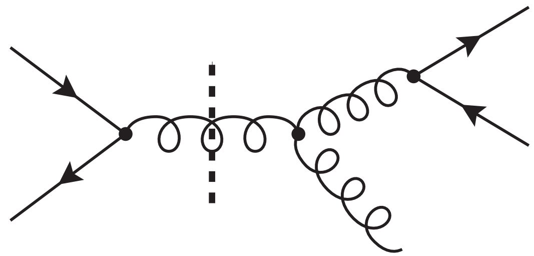
$$\mathcal{M}_l = \sum_d \mathcal{M}_l^{(d)} \qquad (l=0,1) \quad \text{with} \quad \mathcal{M}_l^{(d)} = \mathcal{C}_l^{(d)} \, \mathcal{A}_l^{(d)} \, .$$

Tree level amplitudes split into subtrees

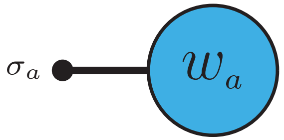
$\mathcal{A}_0^{(d)} =$



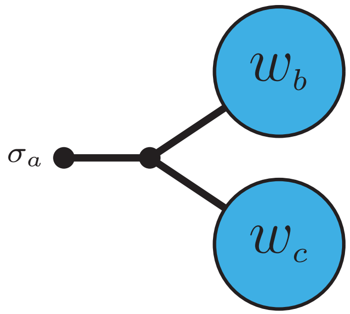
for example



Numerical recursion: $w_a^\alpha =$



$=$



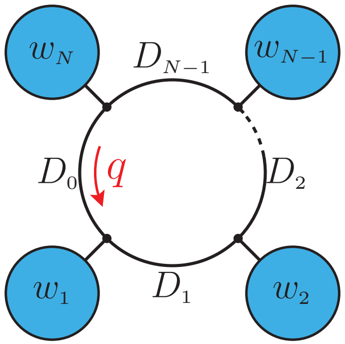
$$= \underbrace{\frac{X^\alpha_{\beta\gamma}}{k_a^2 - m_a^2}}_{\text{universal building block from Feynman rules}} \; w_b^\beta \; w_c^\gamma$$

⇒ Subtrees constructed once for multiple Feynman diagrams at tree and loop level

The OpenLoops framework

One-loop diagram

$$\mathcal{A}_1^{(d)} =$$



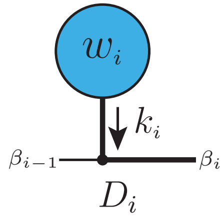
$$= \int \mathrm{d}^D q \, \frac{\mathrm{Tr}\left[\mathcal{N}(q)\right]}{D_0 \cdots D_{N-1}}$$

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

Recursive construction exploiting

factorisation into segments

$$S_i(q) =$$



$$= \left\{ Y_\sigma^i + Z_{\nu;\sigma}^i \, q^\nu \right\} w_i^\sigma$$

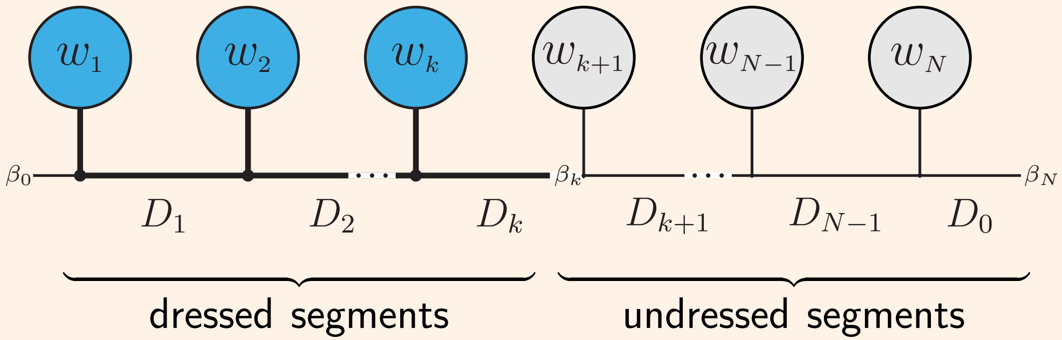
(loop vertex + propagator + subtree(s))

Each segment increases rank in q^μ by 0,1

Open loop at D_0 $\Rightarrow$ $\left[\mathcal{N}(q)\right]_{\beta_0}^{\beta_N} = \left[S_1(q)\right]_{\beta_0}^{\beta_1} \left[S_2(q)\right]_{\beta_1}^{\beta_2} \cdots \left[S_N(q)\right]_{\beta_{N-1}}^{\beta_N}$

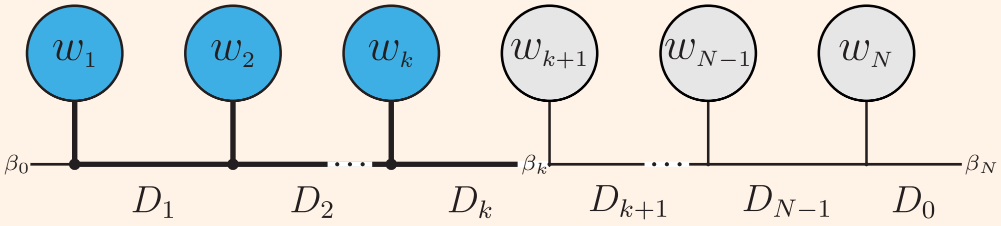
Dress open loop recursively (initial condition $\mathcal{N}_0 = \mathbb{1}$):

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



The OpenLoops dressing recursion

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



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

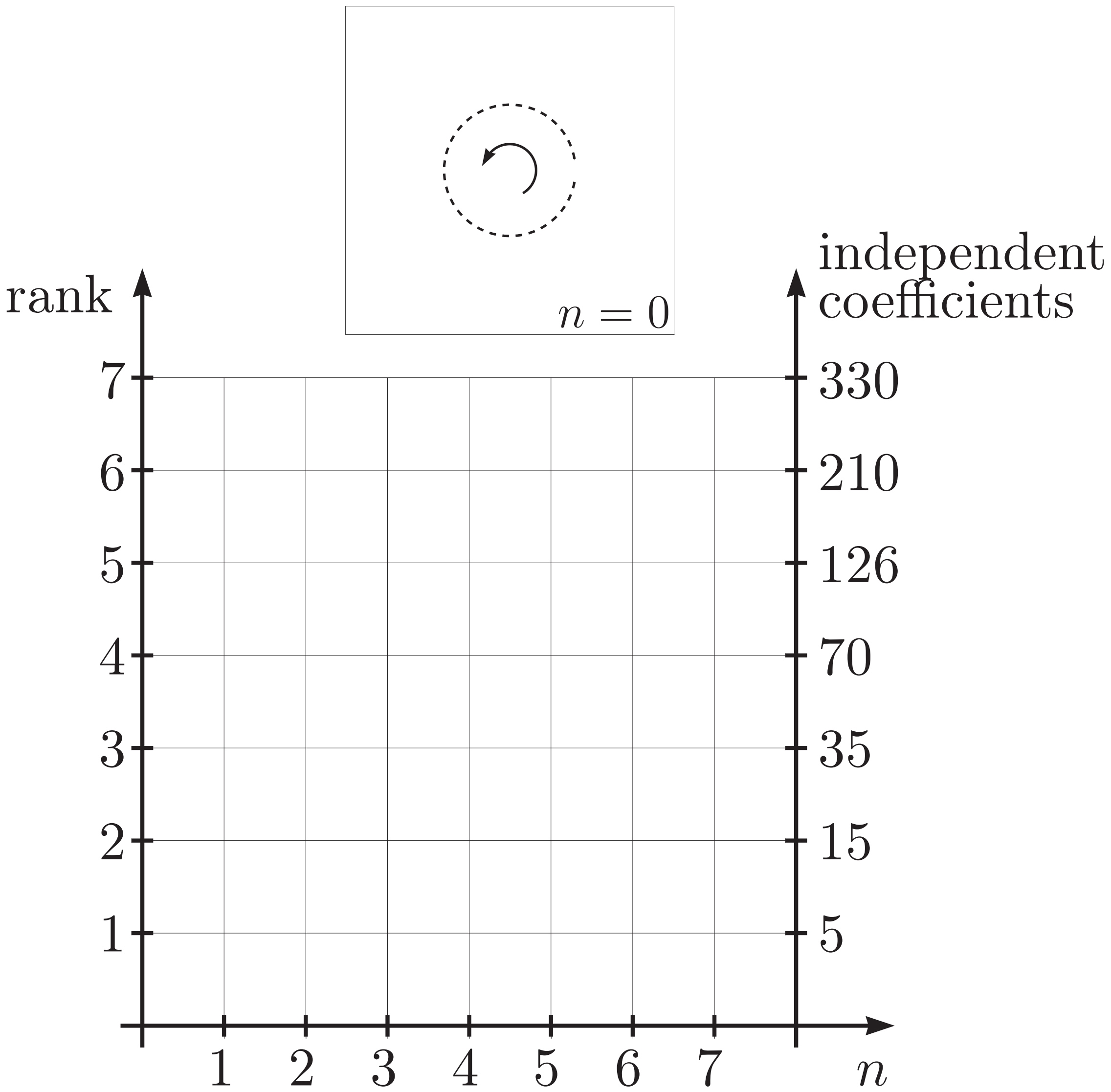
N dressing steps at level of tensor coefficients $\to$ Trace over β_0, β_N $\to$ closed loop

Closed loop treatment in OpenLoops 1:

- For each diagram d and helicity configuration h construct $\mathrm{Tr}\Big[\mathcal{N}_N^{(d)}(q,h)\Big]$
- Interference 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: $\mathcal{V}_N^{(d)}(q,0) = \sum_h \mathcal{V}_N^{(d)}(q,h)$
- Sum same topology diagrams, reduce and evaluate integrals: $\int \mathrm{d}^D q \sum_d \frac{\mathrm{Tr}\Big[\mathcal{V}_N^{(d)}(q,0)\Big]}{D_0, \ldots, D_{N-1}}$

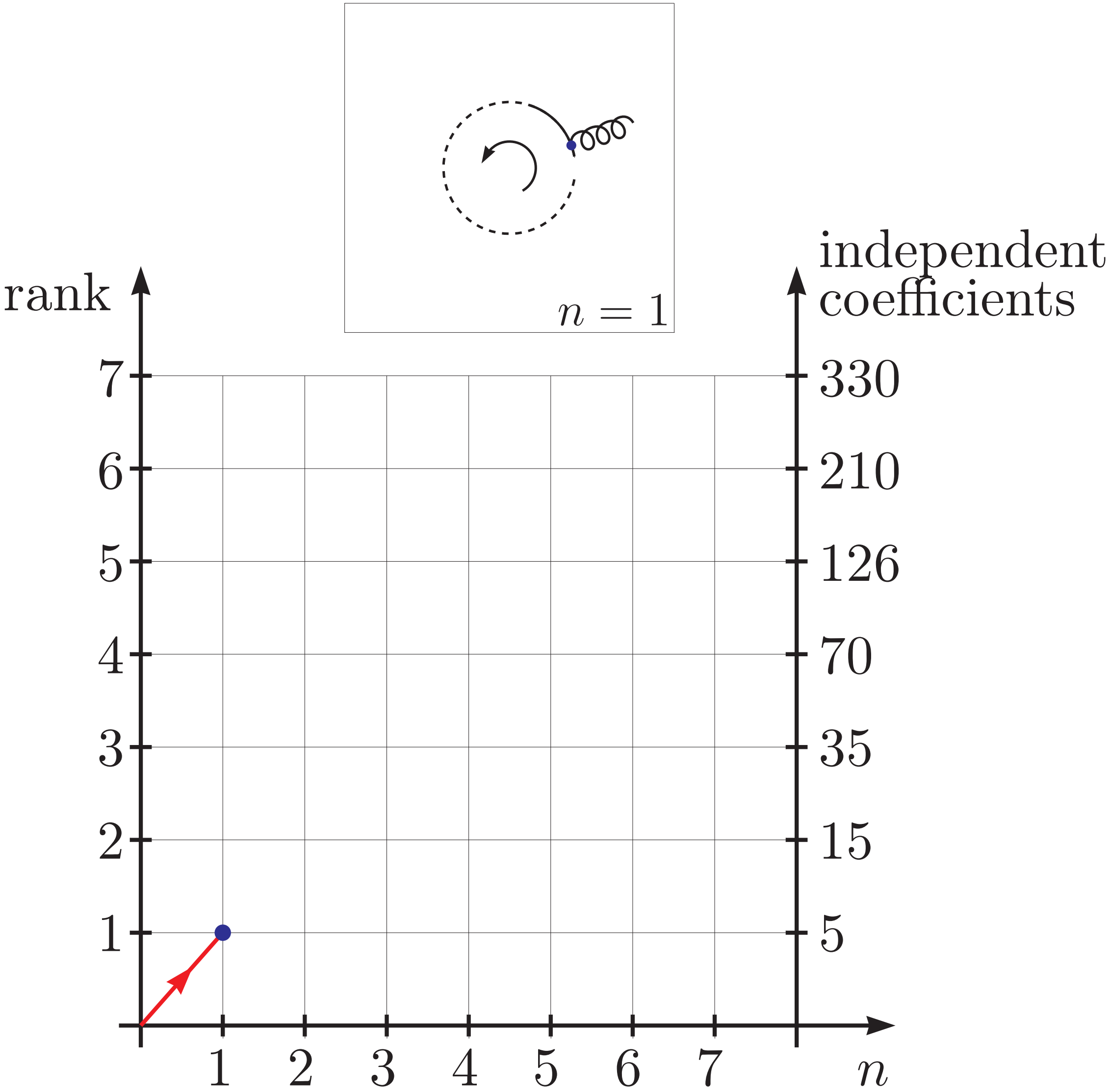
External reduction libraries: `Collier` 1.2 [Denner, Dittmaier, Hofer '16], `Cuttools` 1.9.5 [Ossola, Papadopoulos, Pittau '08] + `OneLoop` 3.6.1 [van Hameren '10]

OpenLoops 1: Example of a high-rank process



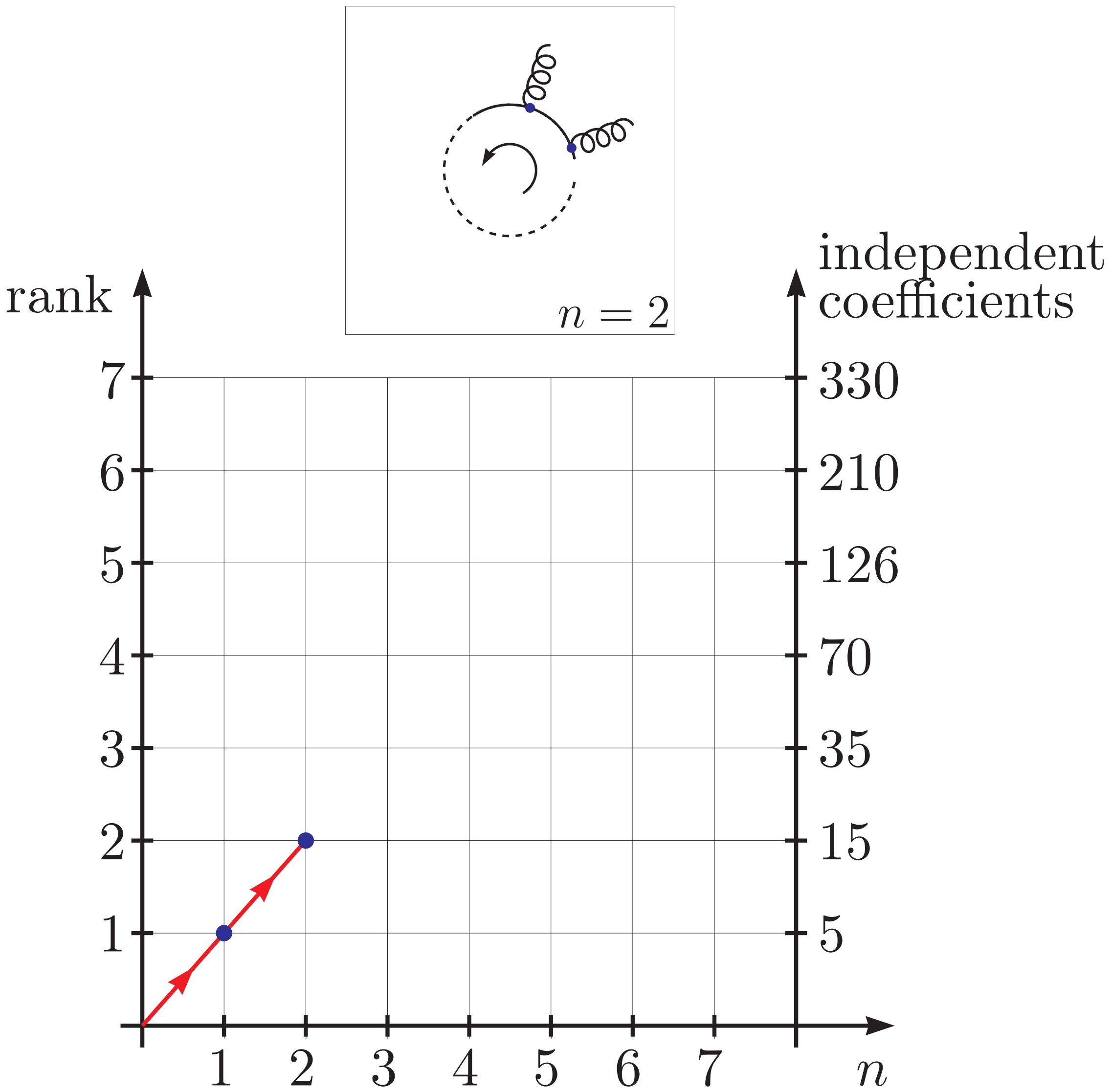
$$\mathcal{N}_0 = \mathbb{1}$$

OpenLoops 1: Example of a high-rank process



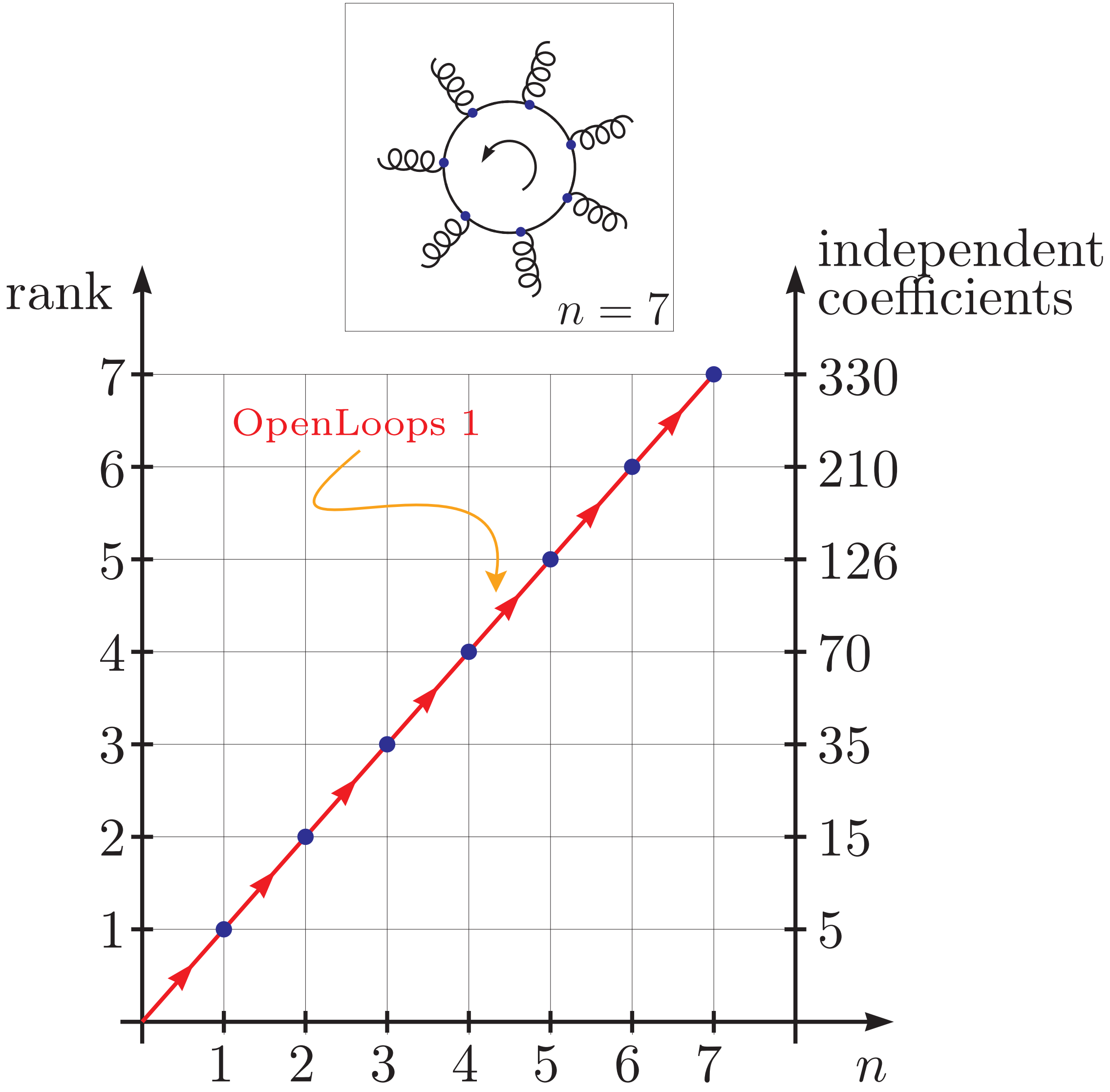
$$\mathcal{N}_1 = \mathcal{N}^{(1)}_{\mu_1}\, q^{\mu_1} + \mathcal{N}^{(1)}$$

OpenLoops 1: Example of a high-rank process



$$\mathcal{N}_2 = \mathcal{N}^{(2)}_{\mu_1\mu_2}\, q^{\mu_1} q^{\mu_2} + \ldots$$

OpenLoops 1: Example of a high-rank process



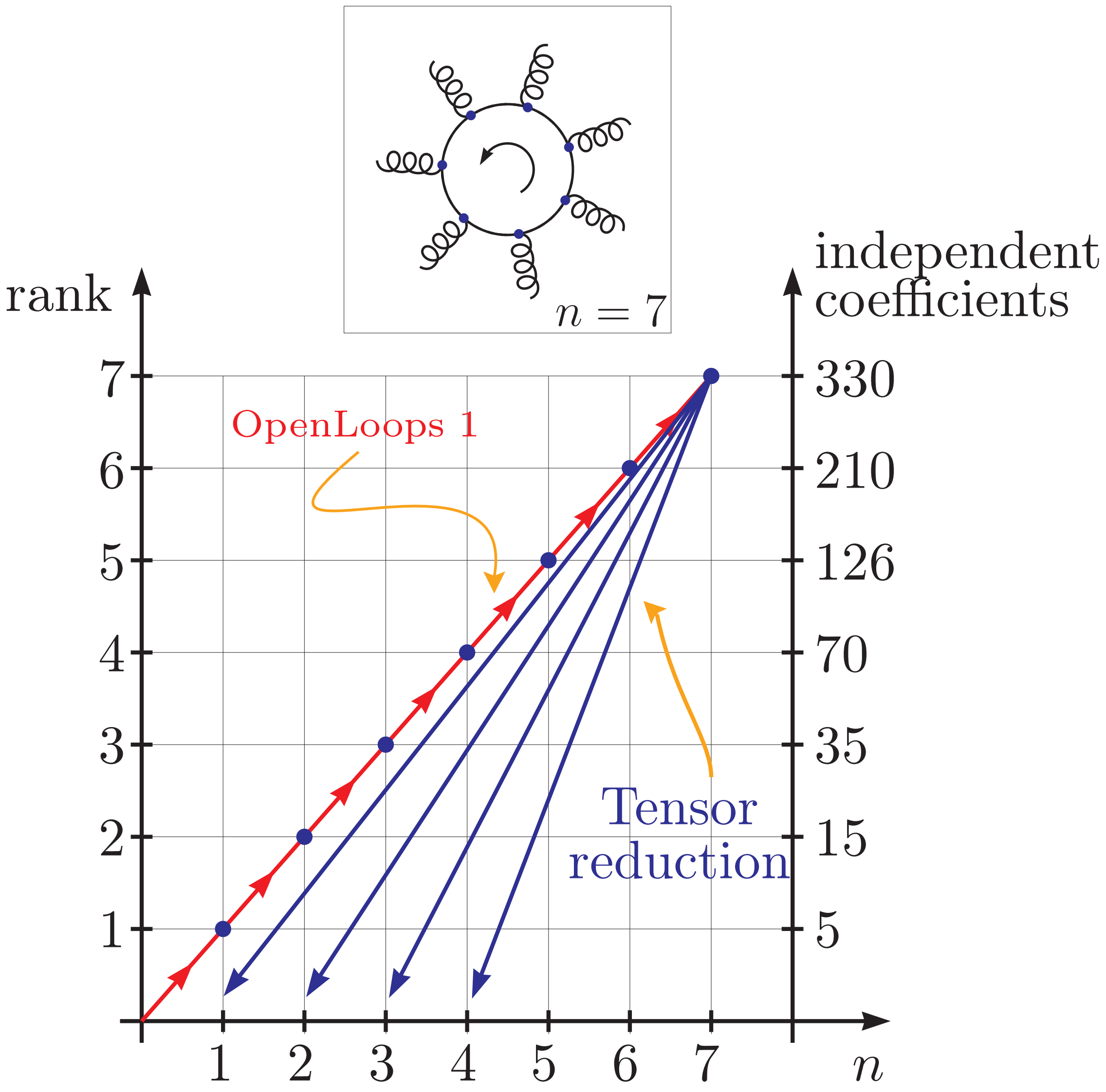
$$\mathcal{N}_7 = \mathcal{N}^{(7)}_{\mu_1\mu_2\cdots\mu_7}\, q^{\mu_1} q^{\mu_2} \cdots q^{\mu_7} + \ldots$$

Problems:

- High complexity in loop diagram
- Stability in IR region challenging for $2 \to 4$

▷ Crucial for $2 \to 3$ NNLO calculations

OpenLoops 1: Example of a high-rank process



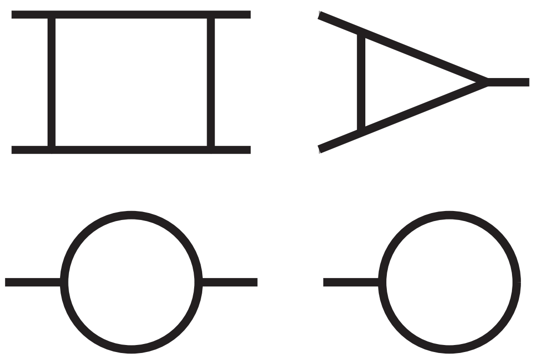
$\mathcal{N}_7 = \mathcal{N}^{(7)}_{\mu_1\mu_2\cdots\mu_7}\, q^{\mu_1} q^{\mu_2} \cdots q^{\mu_7} + \ldots$

Problems:

- High complexity in loop diagram
- Stability in IR region challenging for $2 \to 4$

▷ Crucial for $2 \to 3$ NNLO calculations

Reduction to scalar Master integrals

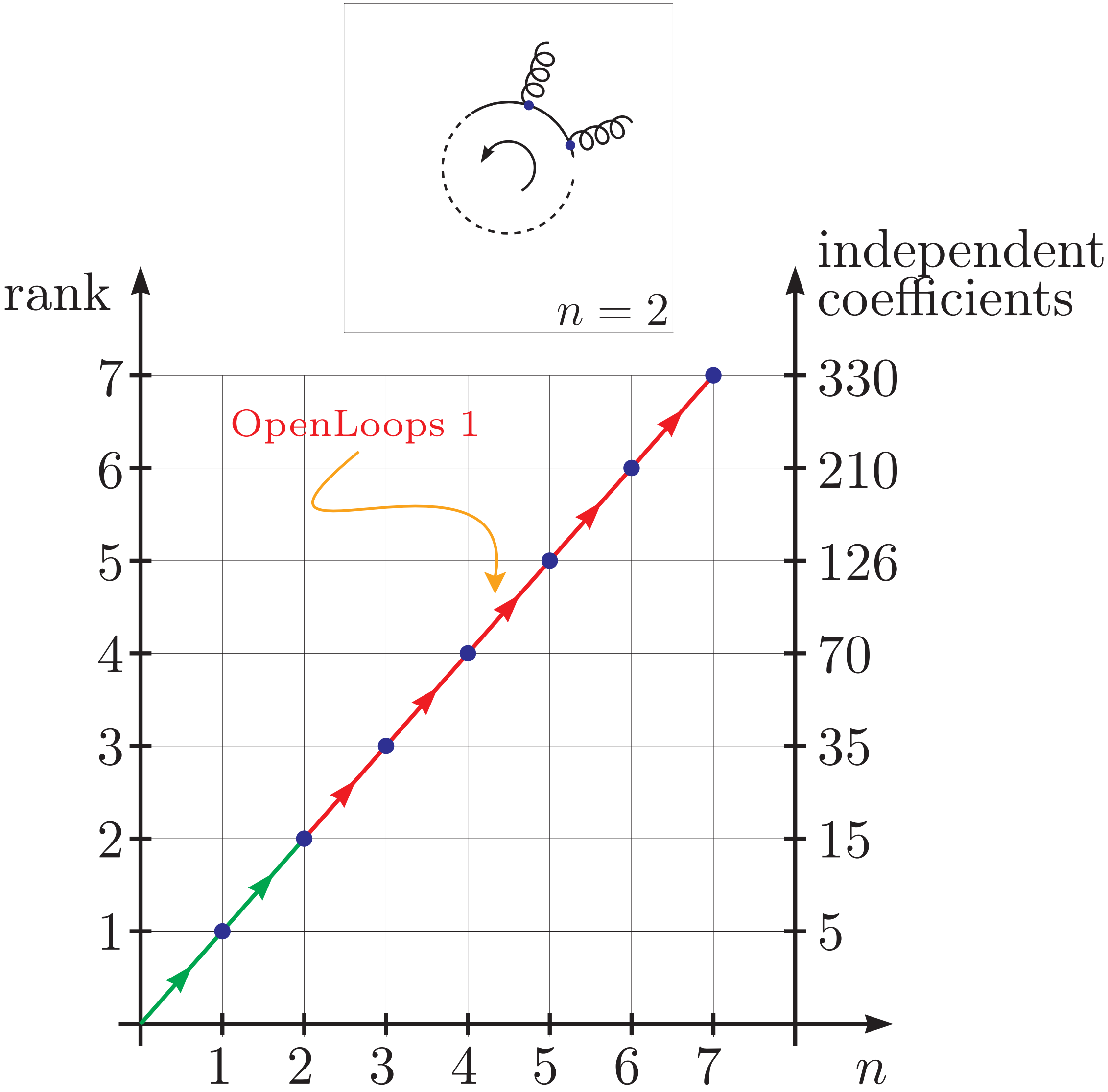


with `Collier` 1.2 [Denner, Dittmaier, Hofer '16],

`Cuttools` 1.9.5 [Ossola, Papadopoulos, Pittau '08]+

`OneLoop` 3.6.1 [van Hameren '10]

II. The On-the-fly method [Buccioni, Pozzorini, M.Z. '18]

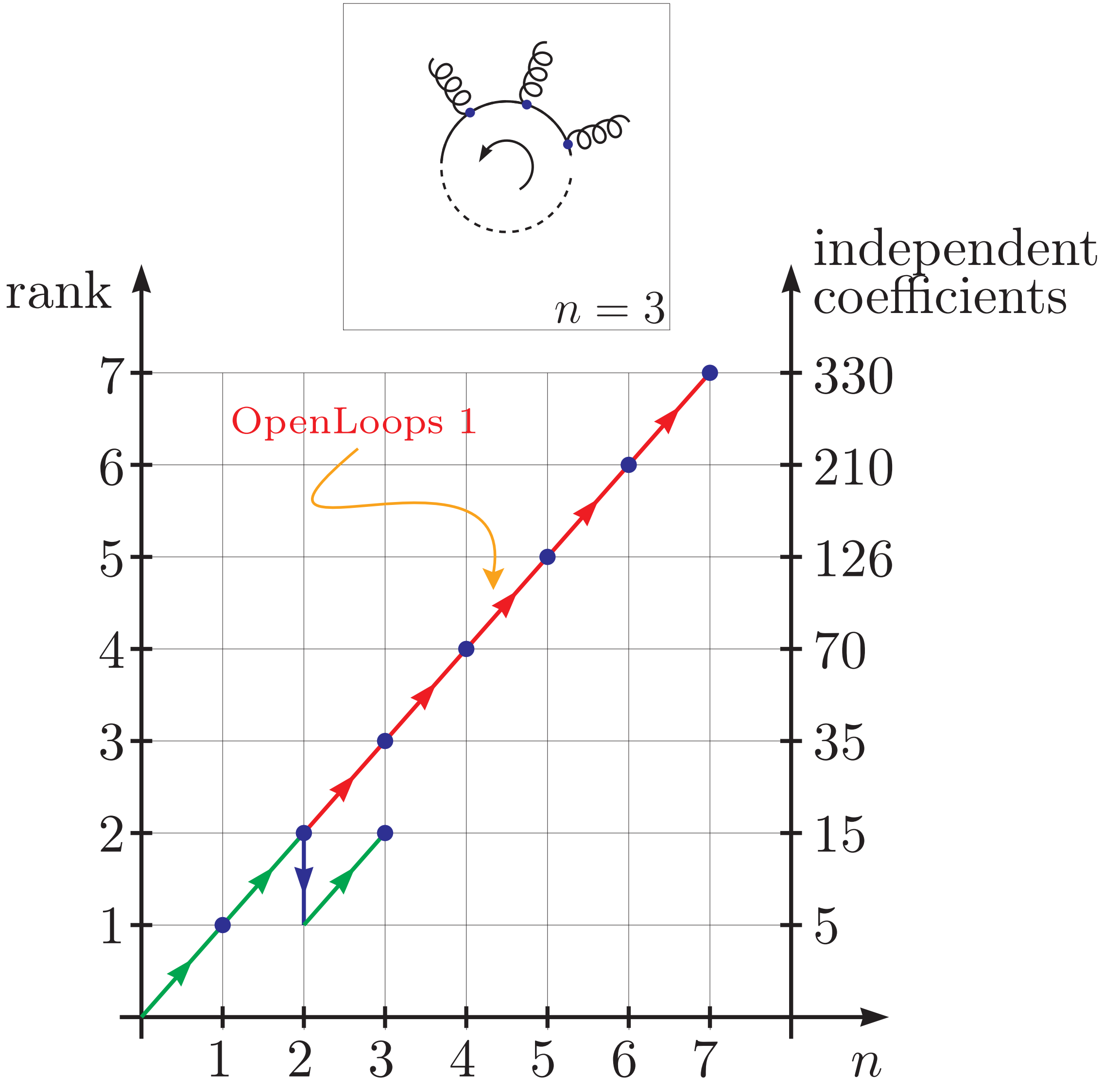


$$\mathcal{N}_2 = \mathcal{N}^{(2)}_{\mu_1\mu_2}\, q^{\mu_1} q^{\mu_2} + \ldots$$

On-the-fly reduction of tensor integrand

$$q_\mu q_\nu \quad = \quad A_{\mu\nu} + B^{\lambda}_{\mu\nu}\, q_\lambda$$

II. The On-the-fly method [Buccioni, Pozzorini, M.Z. '18]

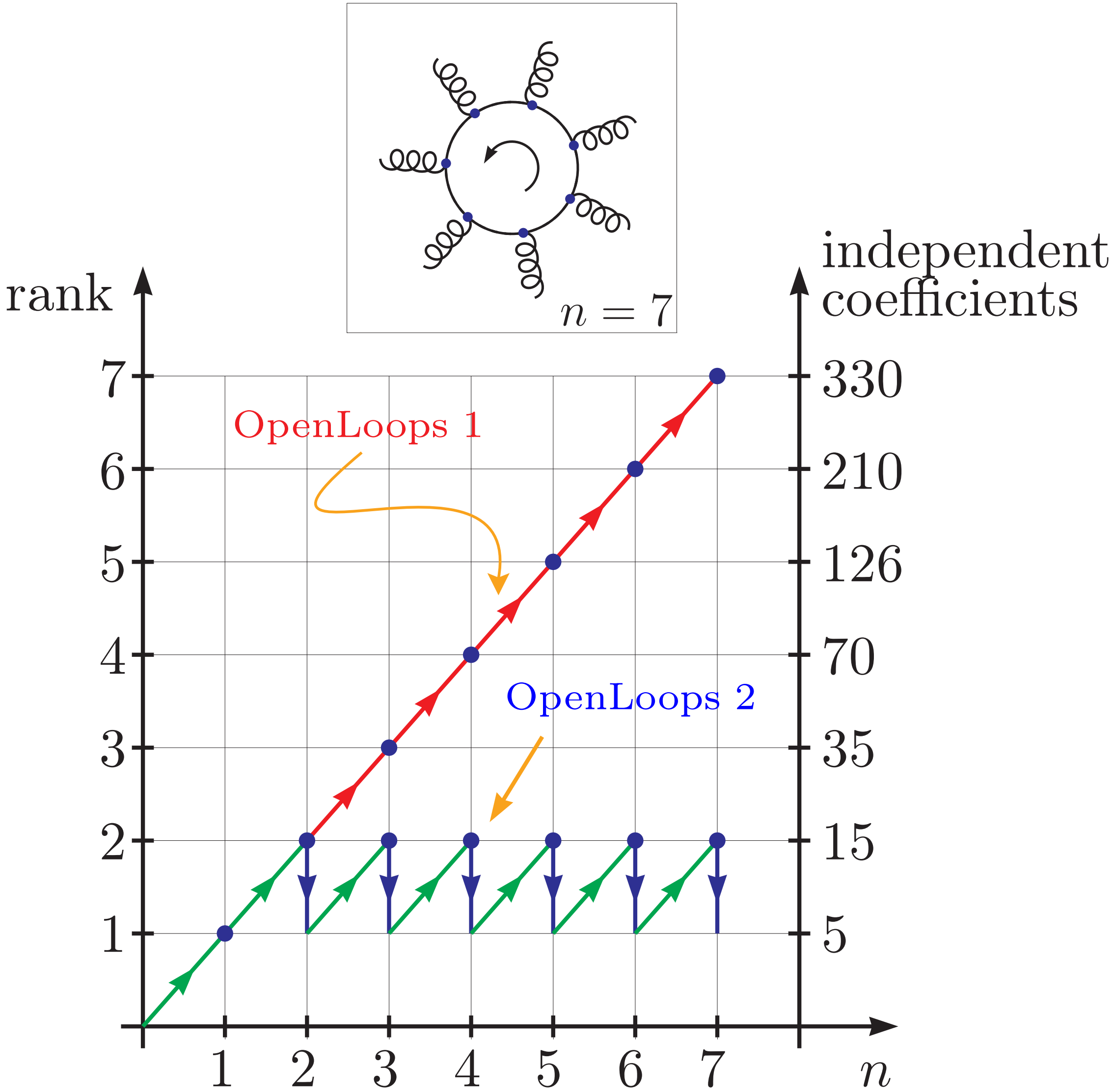


$$\mathcal{N}_3 = \mathcal{N}^{(3)}_{\mu_1\mu_2}\, q^{\mu_1} q^{\mu_2} + \ldots$$

On-the-fly reduction of tensor integrand

$$q_\mu q_\nu \quad = \quad A_{\mu\nu} + B^{\lambda}_{\mu\nu}\, q_\lambda$$

II. The On-the-fly method [Buccioni, Pozzorini, M.Z. '18]



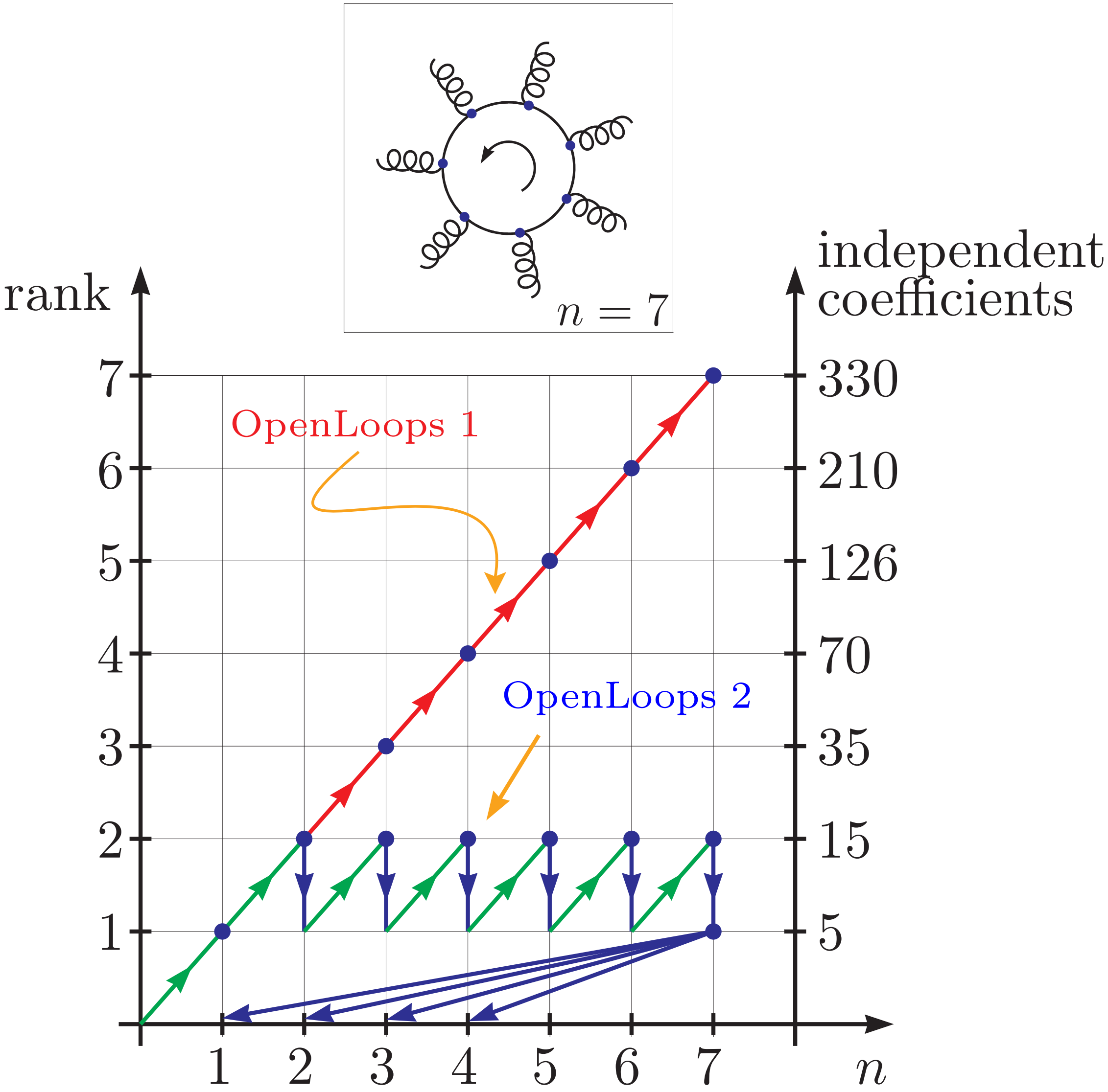
$$\mathcal{N}_7 = \mathcal{N}^{(7)}_{\mu_1}\, q^{\mu_1} + \ldots$$

On-the-fly reduction of tensor integrand

$$q_\mu q_\nu \quad = \quad A_{\mu\nu} + B^{\lambda}_{\mu\nu}\, q_\lambda$$

- Numerical instabilities identified and cured in single reduction steps

II. The On-the-fly method [Buccioni, Pozzorini, M.Z. '18]



$$\mathcal{N}_7 = \mathcal{N}^{(7)}_{\mu_1}\, q^{\mu_1} + \ldots$$

On-the-fly reduction of tensor integrand

$$q_\mu q_\nu \quad = \quad A_{\mu\nu} + B^{\lambda}_{\mu\nu}\, q_\lambda$$

- Numerical instabilities identified and cured in single reduction steps
- Rank 1 and 0 integral reduction to scalar

  with simple OPP for $N \geq 5$ propagators [Ossola, Papadopoulos, Pittau '07] and integral identities for $N \leq 4$ [del Aguila, Pittau '05]

- Evaluate scalar integrals ($N \leq 4$) with `Collier` 1.2 or `OneLoop` 3.6.1

On-the-fly Reduction

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

$$q^\mu q^\nu = \left[A_{-1}^{\mu\nu} + A_0^{\mu\nu} D_0(q)\right] + \left[B_{-1,\lambda}^{\mu\nu} + \sum_{i=0}^{N_{\text{pinch}}-1} B_{i,\lambda}^{\mu\nu} D_i(q)\right] q^\lambda, \qquad D_i(q) = (q+p_i)^2 - m_i^2$$

with $N_{\text{pinch}} = \begin{cases} 4 & \text{for } N \geq 4 \text{ propagators} \\ 3 & \text{for triangles} \end{cases}$ reconstructed denominators $\Rightarrow$ cancel D_i in denominator

Coefficients $A_i^{\mu\nu}, B_{i,\lambda}^{\mu\nu}$ depend on external momenta p_1, p_2 (and p_3 for $N \geq 4$).

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

On-the-fly Reduction

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

$$q^\mu q^\nu \;=\; \left[A^{\mu\nu}_{-1} + A^{\mu\nu}_{0} D_0(q)\right] + \left[B^{\mu\nu}_{-1,\lambda} + \sum_{i=0}^{N_{\text{pinch}}-1} B^{\mu\nu}_{i,\lambda} D_i(q)\right] q^\lambda, \qquad D_i(q) = (q+p_i)^2 - m_i^2$$

with $N_{\text{pinch}} = \begin{cases} 4 & \text{for } N \geq 4 \text{ propagators} \\ 3 & \text{for triangles} \end{cases}$ reconstructed denominators $\Rightarrow$ cancel D_i in denominator

Coefficients $A^{\mu\nu}_i, B^{\mu\nu}_{i,\lambda}$ depend on external momenta p_1, p_2 (and p_3 for $N \geq 4$).

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

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

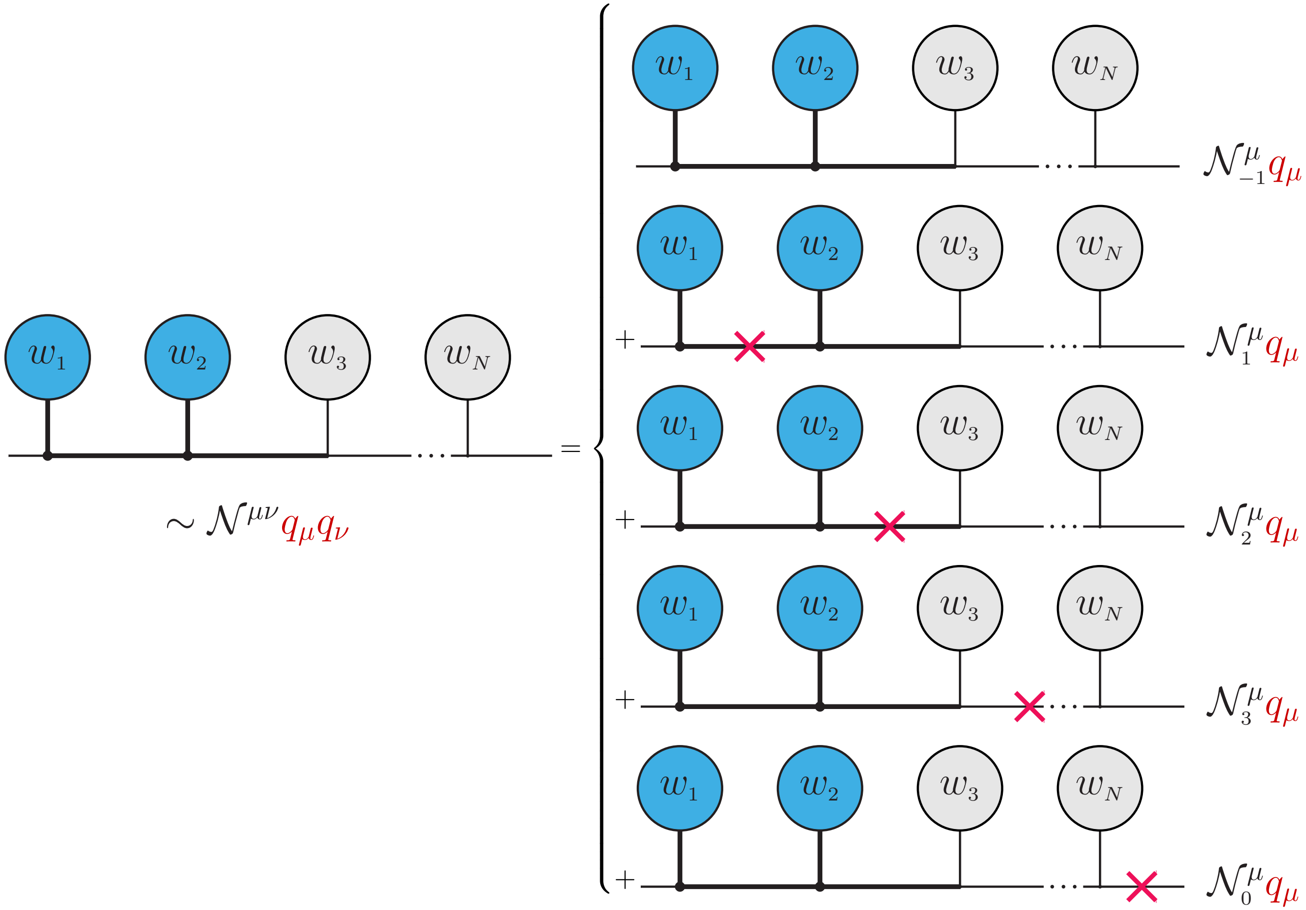
$\Rightarrow$ N_{pinch} new topologies with pinched propagators in each reduction step:

$$\frac{\mathcal{N}^{\mu\nu}\, q_\mu q_\nu}{D_0 \cdots D_{N-1}} = \frac{\mathcal{N}^\mu_{-1} q_\mu + \mathcal{N}_{-1}}{D_0 \cdots D_{N-1}} + \sum_{i=0}^{3} \frac{\mathcal{N}^\mu_i q_\mu + \mathcal{N}_i}{D_0 \cdots D_{i-1} D_{i+1} \cdots D_{N-1}}$$

On-the-fly Reduction

Advantage: Low tensor rank complexity (keep rank ≤ 2 at all times)

Problem: Huge proliferation of topologies due to pinching of propagators:



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

$\Rightarrow$ **Solution: On-the-fly merging**

On-the-fly merging

Sum partially dressed open loops

$$\mathcal{N}_n(q) = \sum_{\alpha} \mathcal{N}_n^{(\alpha)}(q)$$

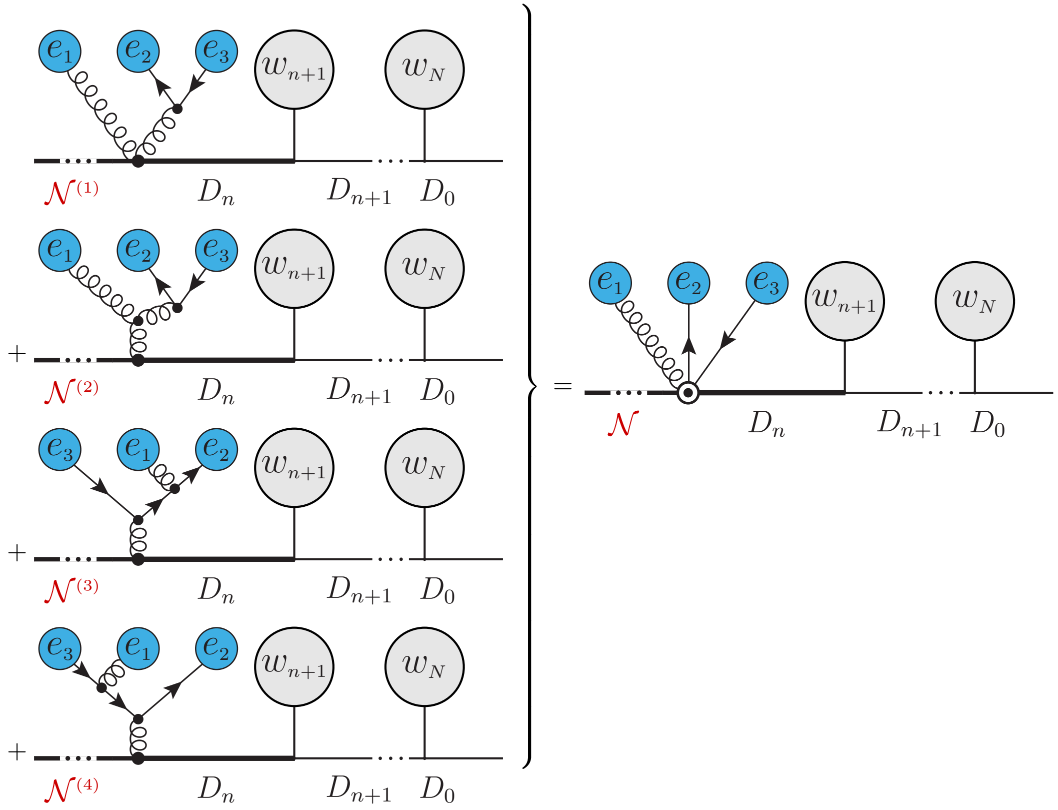
with

- the same topology $D_0, \ldots, D_{N-1}$
- the same undressed segments $S_{n+1}, \ldots, S_N$

since

$$\sum_{\alpha} \frac{\mathcal{N}_n^{(\alpha)}\, S_{n+1} \cdots S_{N-1}}{D_0 D_1 \cdots D_{N-1}} = \frac{\mathcal{N}_n\, S_{n+1} \cdots S_{N-1}}{D_0 D_1 \cdots D_{N-1}}$$

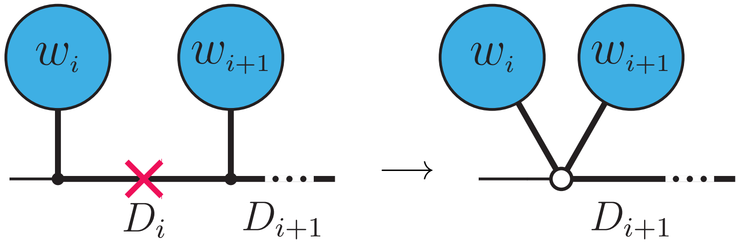
Example:



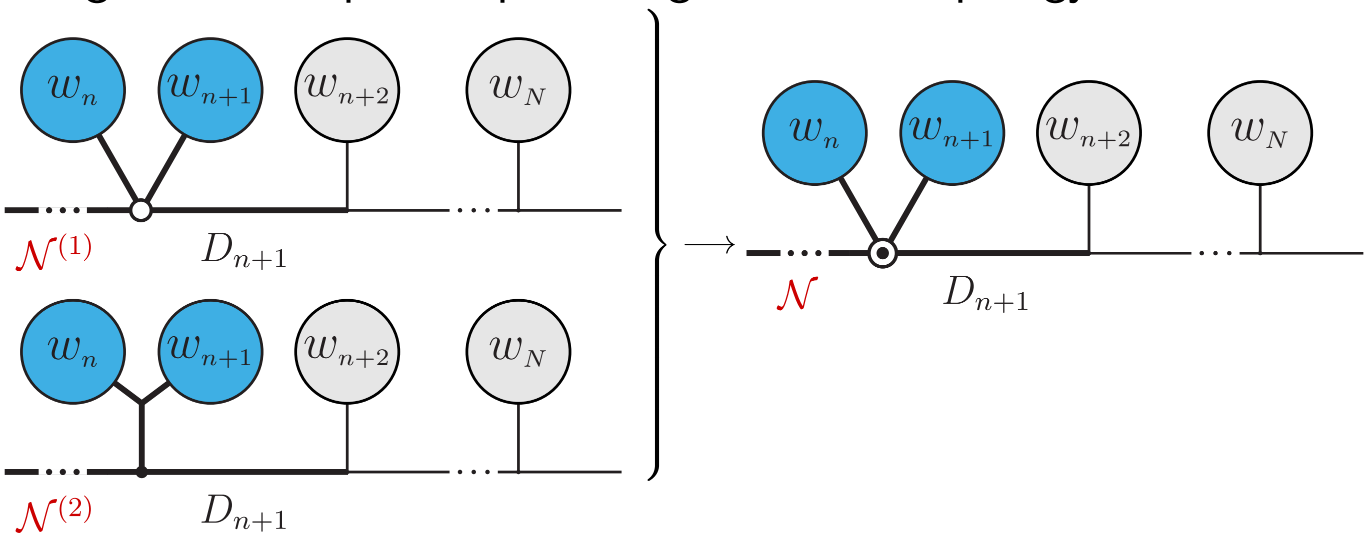
▷ dressing steps for $S_{n+1}, \ldots, S_N$ performed only once for the merged object

On-the-fly merging of pinched-propagator topologies

- Treat two dressed segments with pinched propagator as one effective segment:



- Merge with all open loops having the same topology and same undressed segments



$\Rightarrow$ No extra cost for pinched topologies after merging

OpenLoops 2 recursion step: dress one segment $\rightarrow$ reduce if necessary $\rightarrow$ merge

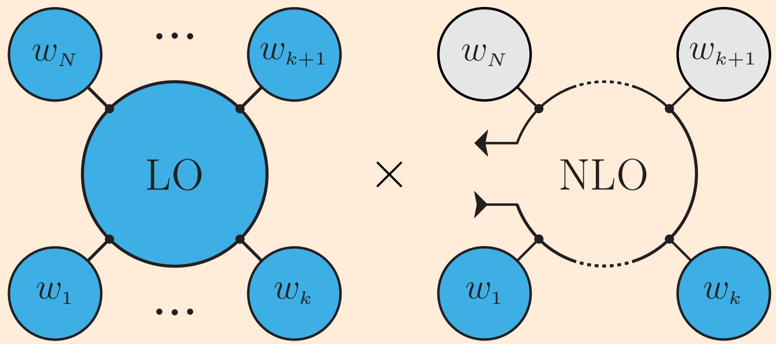
On-the-fly helicity summation

Consider colour-helicity summed numerator ⇒ nested sums of helicities h_i of individual segments

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

- Interfere with colour factor and Born before dressing ⇒ initial open loop $\mathcal{V}_0(h)$
- Sum helicity dof of segment n during n-th dressing step

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



⇒ Open loop only depends on helicity $\check{h}_n = h_{n+1} + \cdots + h_N$ of undressed segments

⇒ **Huge gain in CPU efficiency, especially for high-multiplicity processes**
→ see Federico Buccioni's talk

III. Treatment of numerical instabilities due to small Gram determinants

$$q^\mu q^\nu \;=\; \left[A^{\mu\nu}_{-1} + A^{\mu\nu}_0 D_0\right] + \left[B^{\mu\nu}_{-1,\lambda} + \sum_{i=0}^{3} B^{\mu\nu}_{i,\lambda} D_i\right] q^\lambda, \qquad D_i(q) = (q+p_i)^2 - m_i^2, \quad p_0 = 0$$

$A^{\mu\nu}_i, B^{\mu\nu}_{i,\lambda}$ involve inverse of Gram determinant $\Delta = (p_1 p_2)^2 - p_1^2 p_2^2 = -\Delta_{12}$
(p_3 affects numerical stability much less)

$$A^{\mu\nu}_i \;=\; \frac{1}{\Delta} a^{\mu\nu}_i,$$
$$B^{\mu\nu}_{i,\lambda} \;=\; \frac{1}{\Delta^2}\left[b^{(1)}_{i,\lambda}\right]^{\mu\nu} + \frac{1}{\Delta}\left[b^{(2)}_{i,\lambda}\right]^{\mu\nu}$$

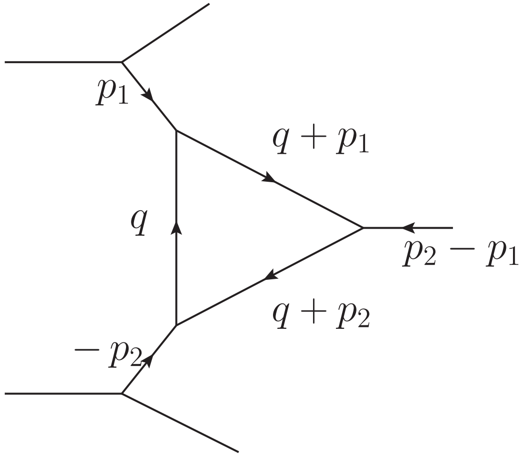
Severe numerical instabilities for $\Delta \to 0$

- For $N \geq 4$: Re-order at runtime: $\{D_1, D_2, D_3\} \longrightarrow \{D_{i_1}, D_{i_2}, D_{i_3}\}$ such that $|\Delta_{i_1 i_2}|/Q^4_{i_1 i_2}$ is maximal ($Q^2_{ij} = \max\{|p_i \cdot p_j|, |p_i^2|, |p_j^2|\}$)
  $\Rightarrow$ **avoid small Gram determinants until triangle reduction!**

- For $N = 3$: Identify problematic kinematic configurations and use analytical expansions.

Triangle reduction

For hard kinematics only one case with small Gram determinant: t-channel with



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

- Expand reduction formula in δ, e.g. massless rank-1 topology:

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

- Expand master integrals as well $\Rightarrow$ $\frac{1}{\delta}$-poles cancel (also for massive cases and higher rank):

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

with $C_0(p_1^2, p_2^2) \sim \int \mathrm{d}^D q \, \frac{1}{D_0 D_1 D_2}$ and $B_0(p_1^2) \sim \int \mathrm{d}^D q \, \frac{1}{D_0 D_1}$

Any-order expansions [in collaboration with J.-N. Lang, H. Zhang]

Expand B_0, C_0 in δ and cancel all poles, e.g.

$$\frac{1}{\delta^n} B_0(-p^2(1+\delta)) = \underbrace{\left(\frac{1}{\delta^n} B_0(-p^2) + \ldots + \frac{1}{\delta} B_0^{(n)}(-p^2)\right)}_{\text{poles} \to \text{cancel}} + \underbrace{B_{0,n}(-p^2, \delta)}_{\text{regular in } \delta}$$

with

$$\begin{aligned} B_{0,n}(-p^2, \delta) &= \sum_{m=n}^{\infty} \delta^{m-n} \left[\frac{1}{m!} \left(\frac{\partial}{\partial \delta} \right)^m B_0 \left(-p^2(1+\delta) \right) \right]_{\delta=0} \\ C_{0,n}(-p^2, \delta) &= \sum_{m=n}^{\infty} \delta^{m-n} \left[\frac{1}{m!} \left(\frac{\partial}{\partial \delta} \right)^m C_0 \left(-p^2, -p^2(1+\delta) \right) \right]_{\delta=0} \end{aligned}$$

Example:

$$C^\mu = (p_1 - p_2)^\mu \left[\frac{B_{0,1} + 2\, B_{0,2}}{p^2} - C_{0,1} \right] + p_1^\mu \left[\frac{B_{0,1}}{p^2} - C_0 \right]$$

Compact formulas derived and implemented for $\left(\frac{\partial}{\partial \delta}\right)^m B_0$ and $\left(\frac{\partial}{\partial \delta}\right)^m C_0$ (all QCD mass configurations).

$\Rightarrow$ $B_{0,n}$ and $C_{0,n}$ computed to any order m_{max} in order to reach any given target precision!
Uncertainty due to truncation of series avoided entirely.

Extremely fast implementation: Complexity of $B_{0,n}$ and $C_{0,n}$ scales like (number of computed terms)^2.

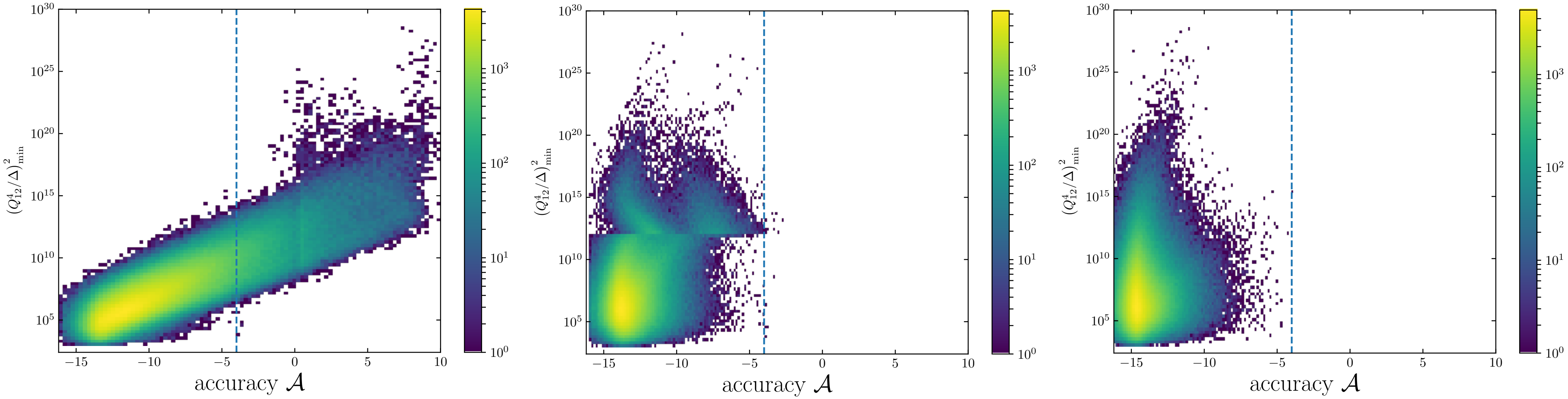
Accuracy improvements and stability system

Correlation between accuracy $\mathcal{A}$ and the largest $(Q^4/\Delta)^2$ in the event

from any rank-2 Gram determinant $\Delta = \Delta(p_i, p_j)$ with corresponding $Q^2 = \max\{|p_i \cdot p_j|, |p_i^2|, |p_j^2|\}$

$gg \to t\bar{t}gg$ with 10^6 events (OpenLoops 2 in double precision)

On-the-fly reduction with no stability improvements | (D_1, D_2, D_3)-permutation + expansions to $\mathcal{O}(\delta^2)$ | (D_1, D_2, D_3)-permutation + any-order expansions



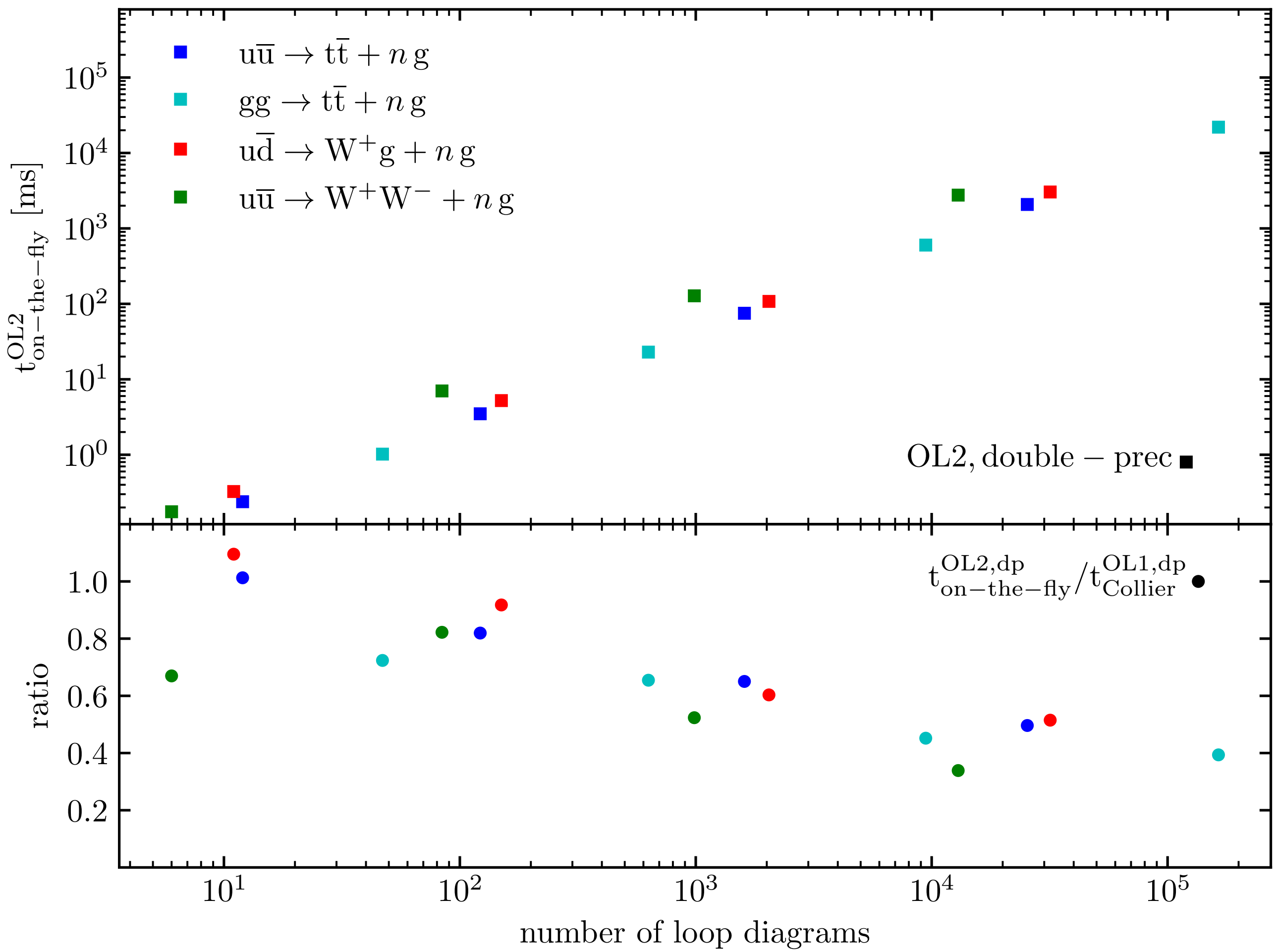
All features implemented in double and quadruple precision. No truncation error in expansions.

$\Rightarrow$ **Stability rescue system**: Use rescaling test for calculations in double precision and re-compute in quadruple precision if result is below target accuracy.

IV. Performance and numerical stability benchmarks

Runtime per phase space point – OpenLoops 1 with Collier vs OpenLoops 2:

one-loop scattering probabilities for processes with $n = 0, 1, 2, 3$ gluons (up to $2 \to 5$ with $\sim 10^5$ diagrams)

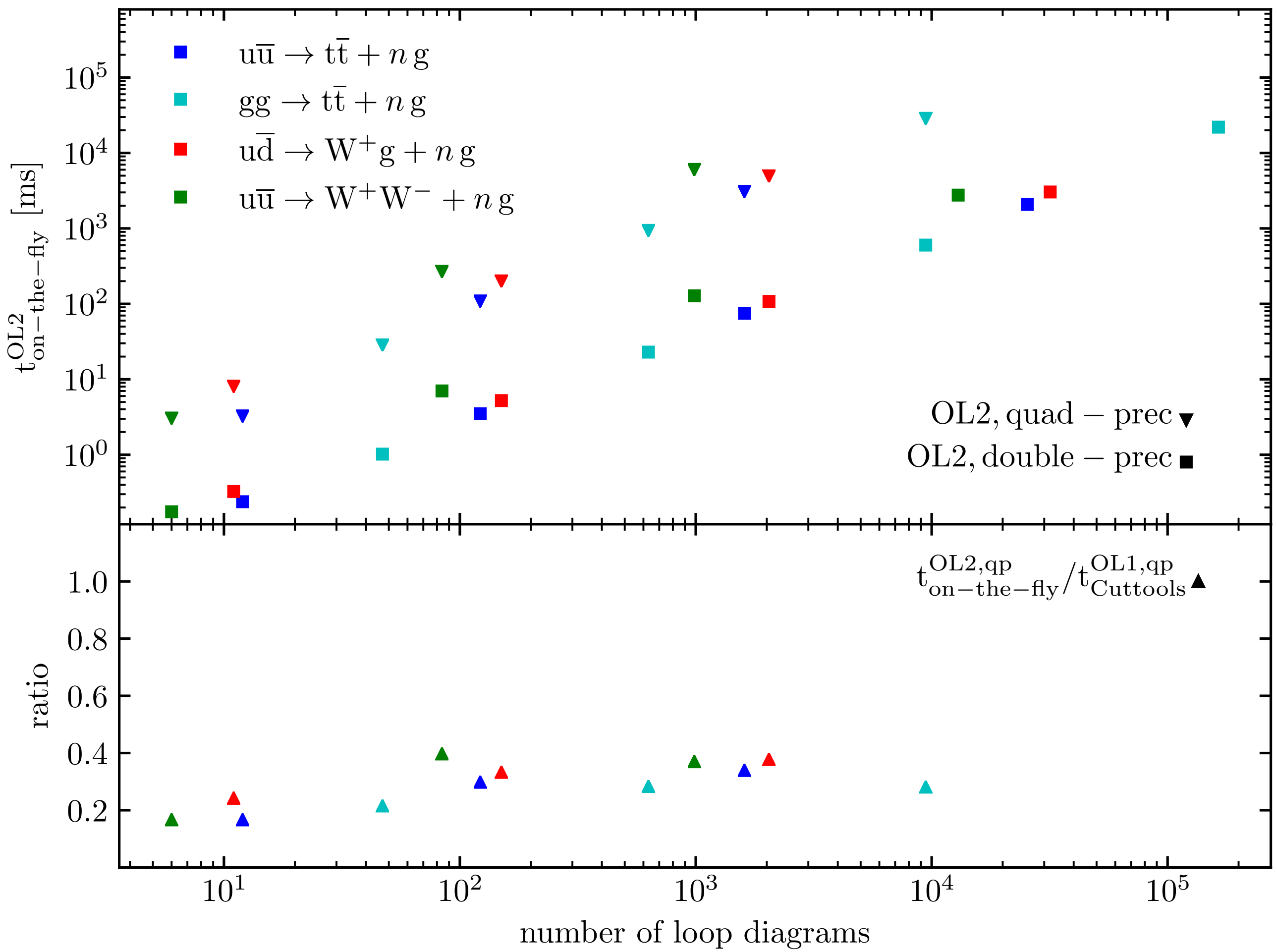


Factor $\sim (2-4)$ speedup for complicated processes in double precision (single Intel i7-4790K core, gfortran-4.8.5)

IV. Performance and numerical stability benchmarks

Runtime per phase space point – OpenLoops 1 with Collier vs OpenLoops 2:

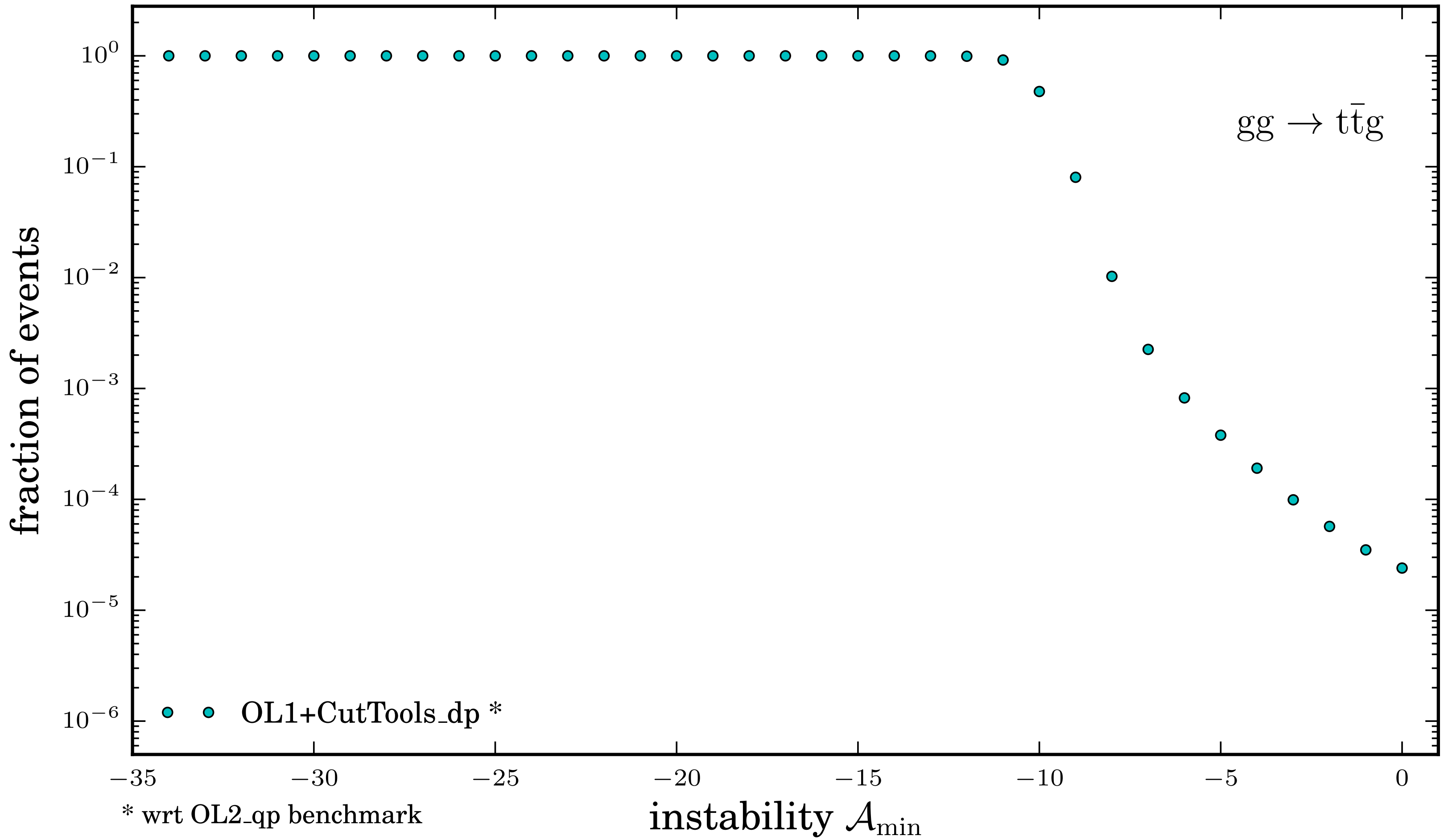
one-loop scattering probabilities for processes with $n = 0, 1, 2, 3$ gluons (up to $2 \to 5$ with $\sim 10^5$ diagrams)



Factor $\sim (3-5)$ speedup in quadruple precision (single Intel i7-4790K core, gfortran-4.8.5)

Stability of OpenLoops (OL1 and OL2) for a $2 \to 3$ process at $\sqrt{\hat{s}} = 1$ TeV (10^6 events)

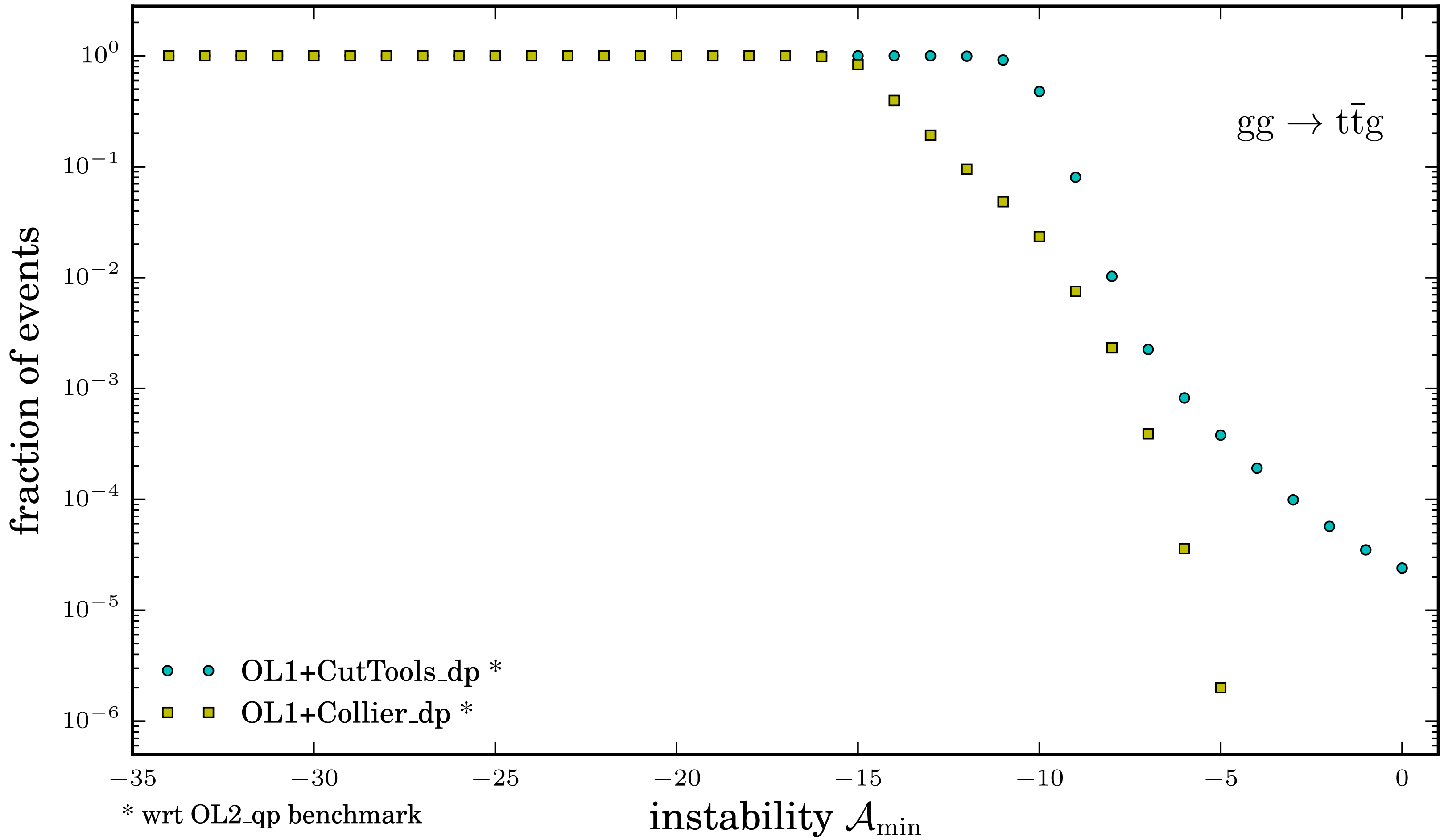
Probability of relative accuracy $\mathcal{A} \le \mathcal{A}_{min}$ in **OL1+Cuttools in double precision (dp)** wrt quad precision benchmark



Hard cuts: $p_T > 50$ GeV and $\Delta R_{ij} => 0.5$ for final state QCD partons.

Stability of OpenLoops (OL1 and OL2) for a $2 \to 3$ process at $\sqrt{\hat{s}} = 1$ TeV (10^6 events)

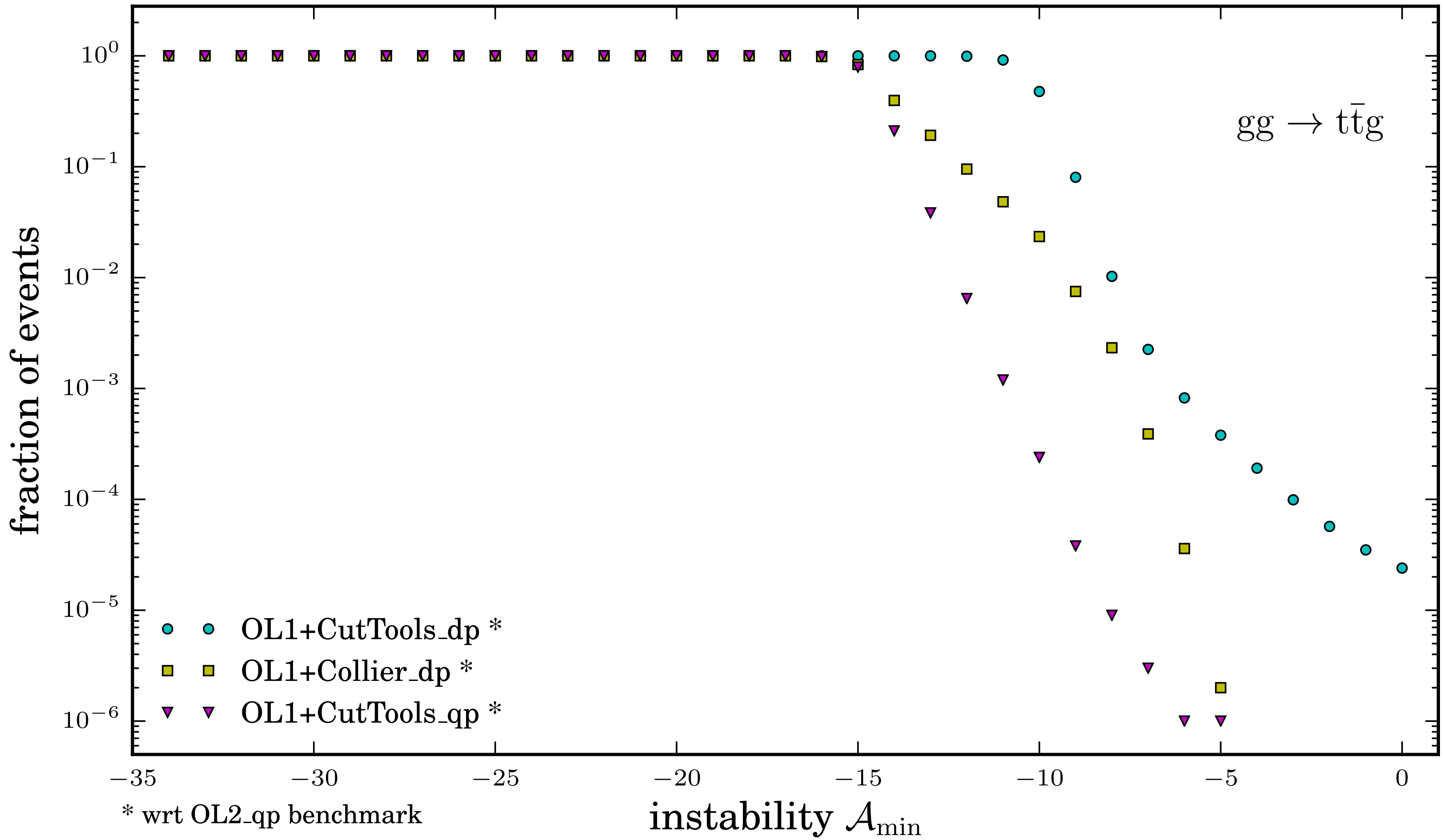
Probability of relative accuracy $\mathcal{A} \leq \mathcal{A}_{min}$ in **OL1+Collier in double precision (dp)** wrt quad precision benchmark



Hard cuts: $p_T > 50$ GeV and $\Delta R_{ij} => 0.5$ for final state QCD partons.

Stability of OpenLoops (OL1 and OL2) for a $2 \to 3$ process at $\sqrt{\hat{s}} = 1$ TeV (10^6 events)

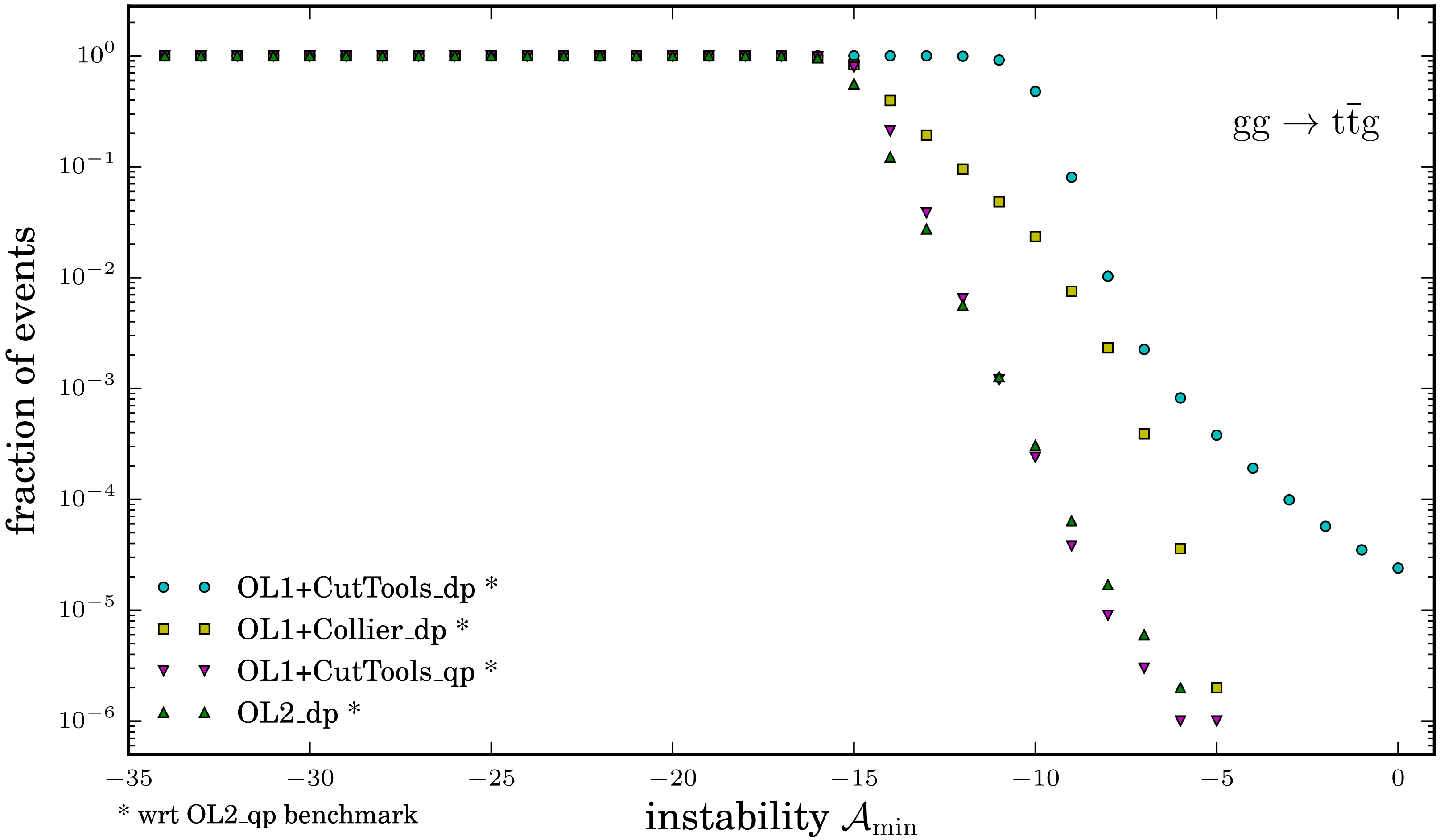
Probability of relative accuracy $\mathcal{A} \leq \mathcal{A}_{min}$ in **OL1+Cuttools in quad precision (qp)** wrt quad precision benchmark



Hard cuts: $p_T > 50$ GeV and $\Delta R_{ij} => 0.5$ for final state QCD partons.

Stability of OpenLoops (OL1 and OL2) for a $2 \to 3$ process at $\sqrt{\hat{s}} = 1$ TeV (10^6 events)

Probability of relative accuracy $\mathcal{A} \leq \mathcal{A}_{min}$ in **OL2 in double precision (dp)** wrt quad precision benchmark

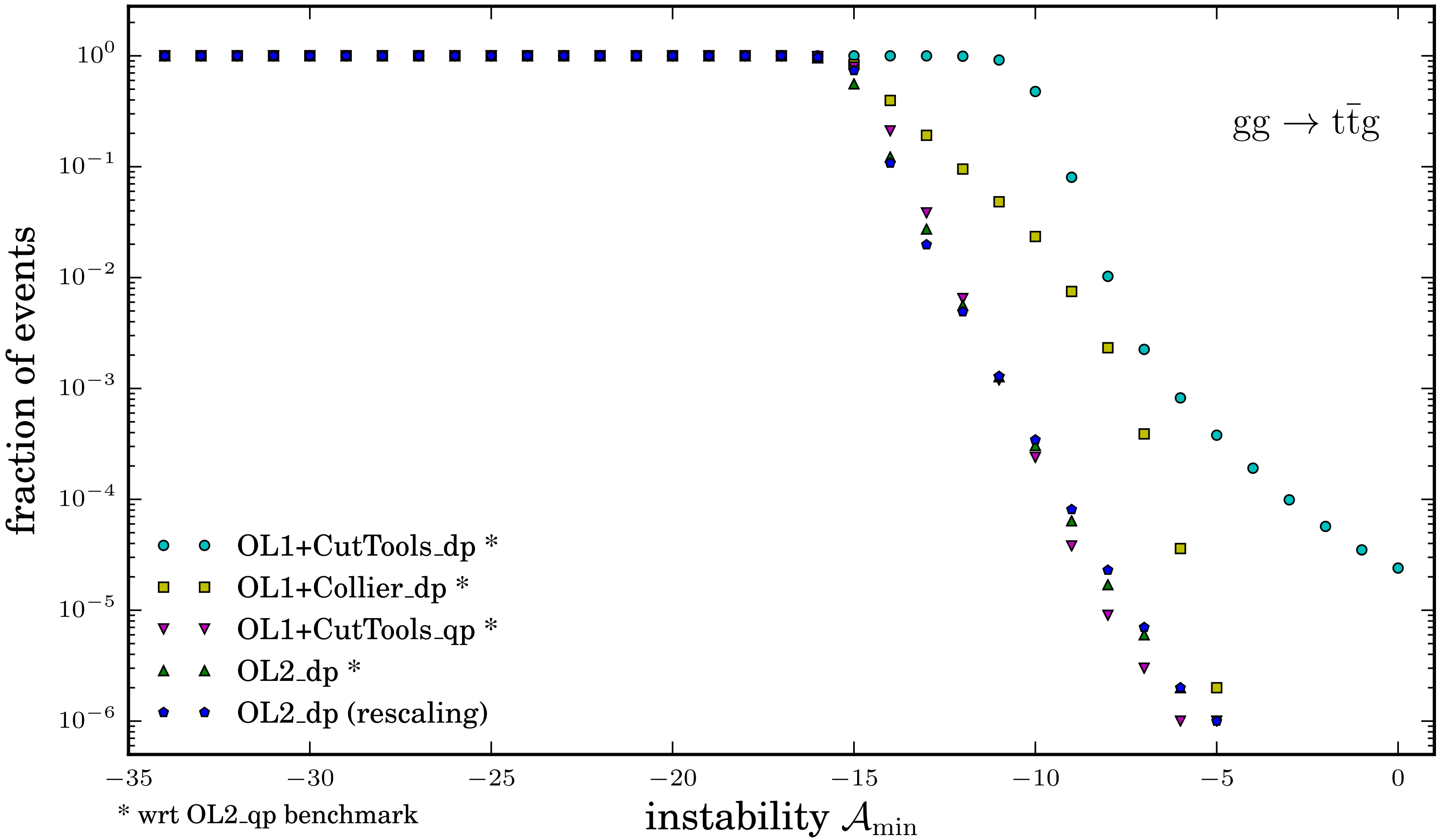


Hard cuts: $p_T > 50$ GeV and $\Delta R_{ij} => 0.5$ for final state QCD partons. **Scalar ($N \leq 4$)-integrals:** Collier

Excellent stability thanks to on-the-fly reduction and dedicated any-order expansions

Stability of OpenLoops (OL1 and OL2) for a $2 \to 3$ process at $\sqrt{\hat{s}} = 1$ TeV (10^6 events)

Probability of relative accuracy $\mathcal{A} \leq \mathcal{A}_{min}$ in **OL2 in double precision (dp)** from rescaling test

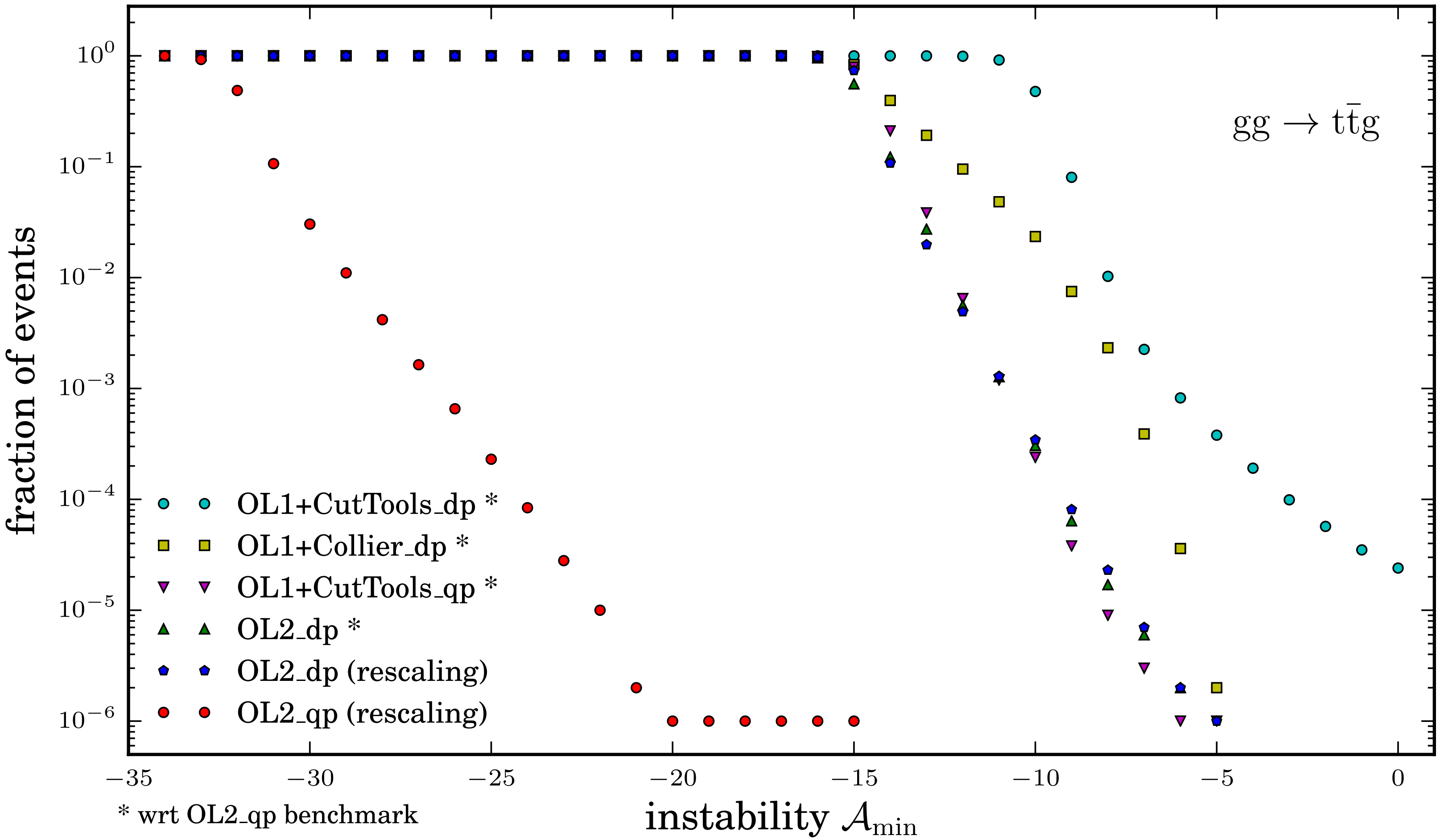


Hard cuts: $p_T > 50$ GeV and $\Delta R_{ij} => 0.5$ for final state QCD partons. **Scalar ($N \leq 4$)-integrals:** Collier

No error from truncation of expansions $\Rightarrow$ Reliable rescaling test

Stability of OpenLoops (OL1 and OL2) for a $2 \to 3$ process at $\sqrt{\hat{s}} = 1$ TeV (10^6 events)

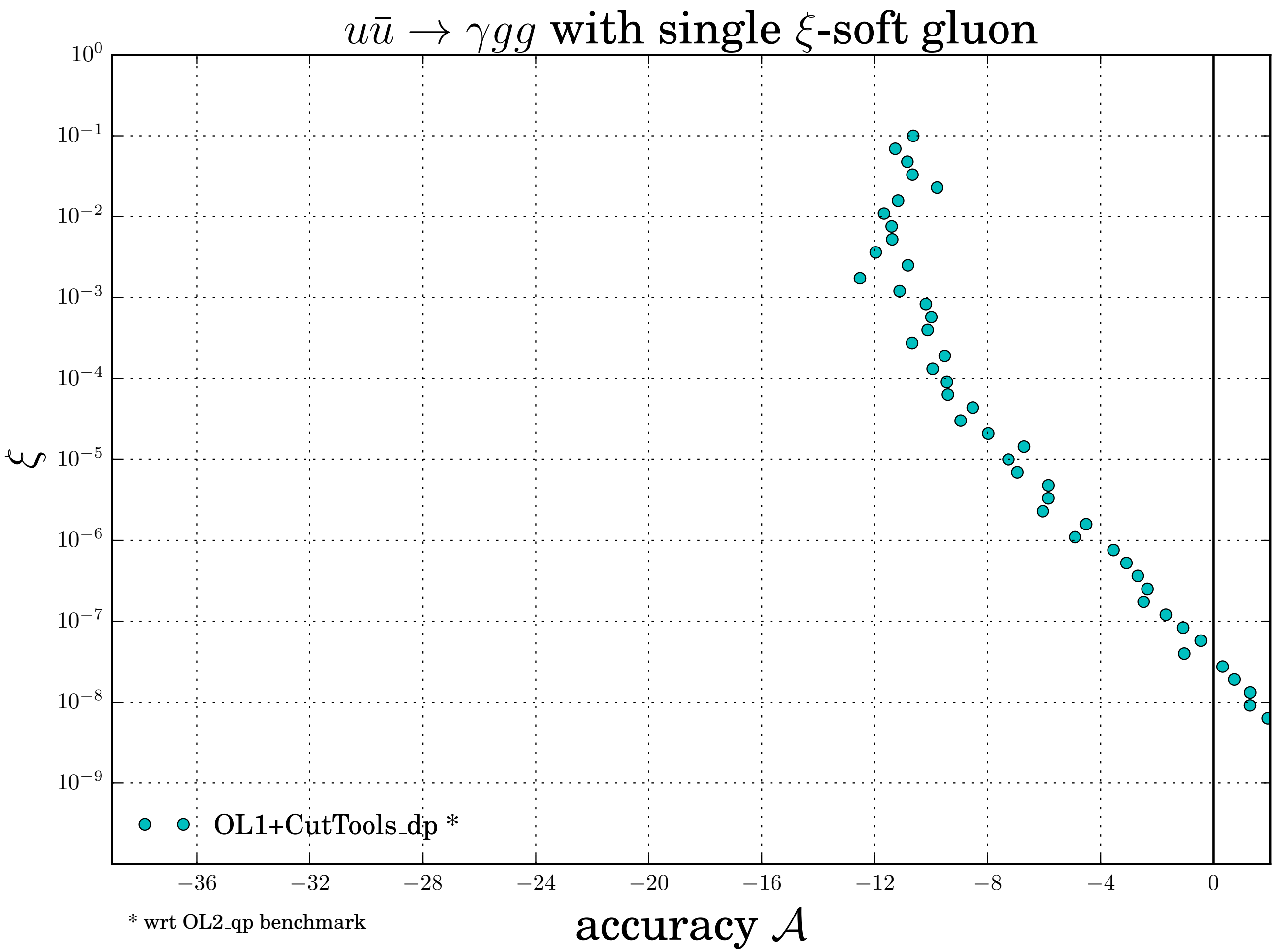
Probability of relative accuracy $\mathcal{A} \le \mathcal{A}_{min}$ in **OL2 in quad precision (qp)** from rescaling test wrt quad precision benchmark



Hard cuts: $p_T > 50$ GeV and $\Delta R_{ij} => 0.5$ for final state QCD partons. **Scalar ($N \le 4$)-integrals:** OneLoop

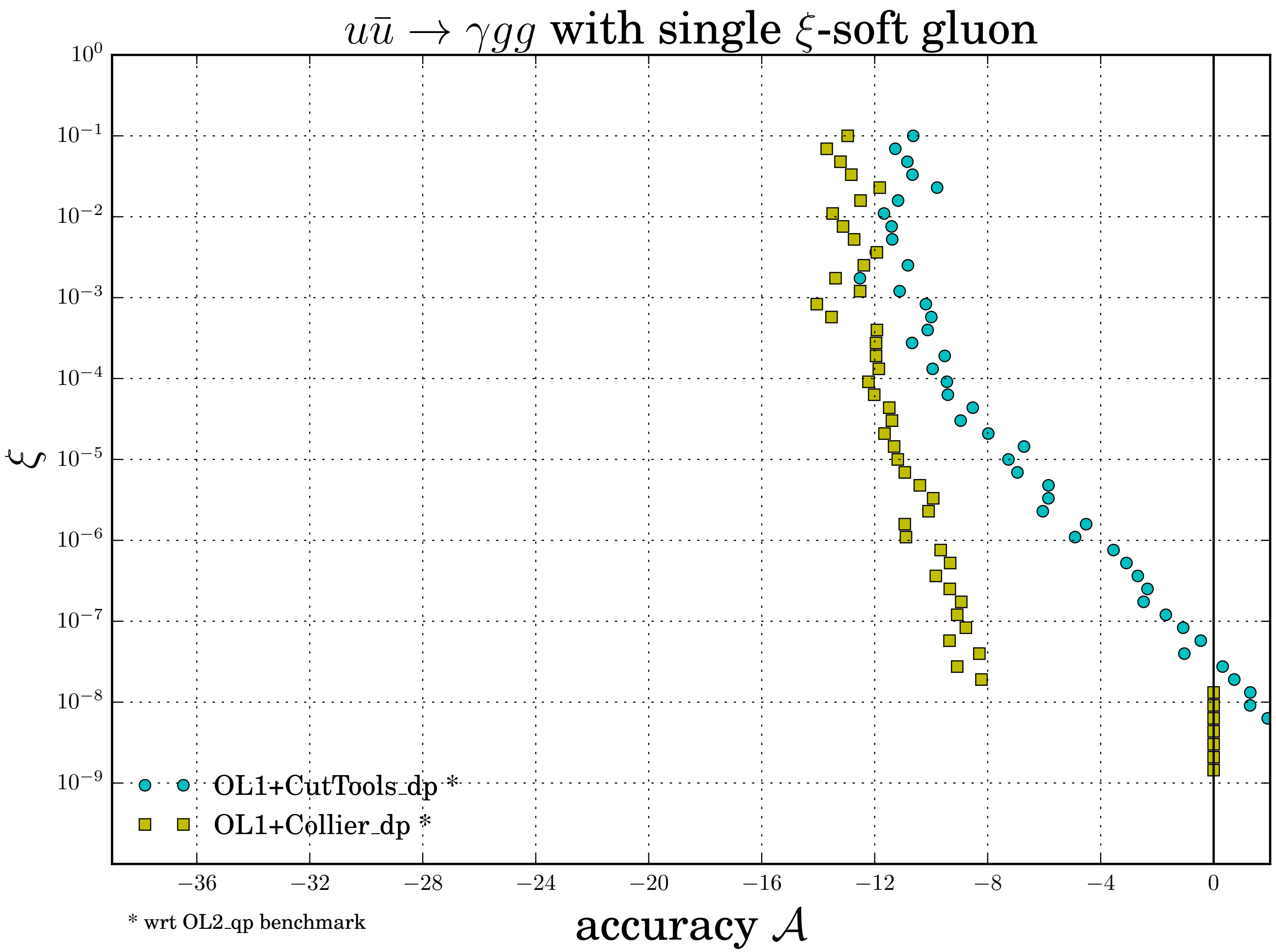
Up to 32 digits thanks to on-the-fly reduction and any-order expansions **(no truncation error)**

Stability in the soft region: $2 \to 3$ process at $\sqrt{\hat{s}} = 1$ TeV **OpenLoops 1+Cuttools (dp)**



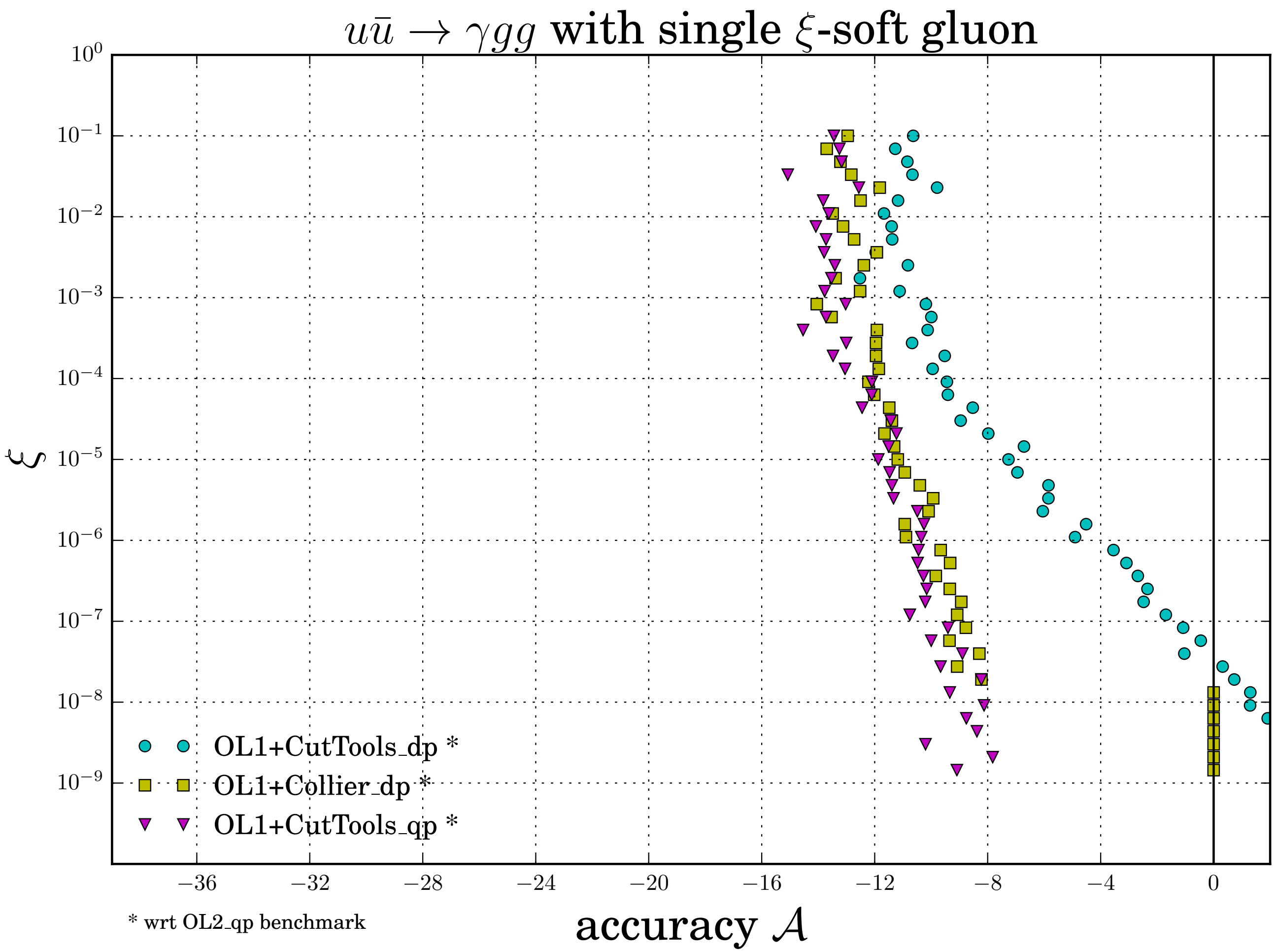
Single soft gluon with energy $E_{soft} = \xi\sqrt{\hat{s}}$. All other kinematic parameters fixed.

Stability in the soft region: $2 \to 3$ process at $\sqrt{\hat{s}} = 1$ TeV **OpenLoops 1+Collier (dp)**



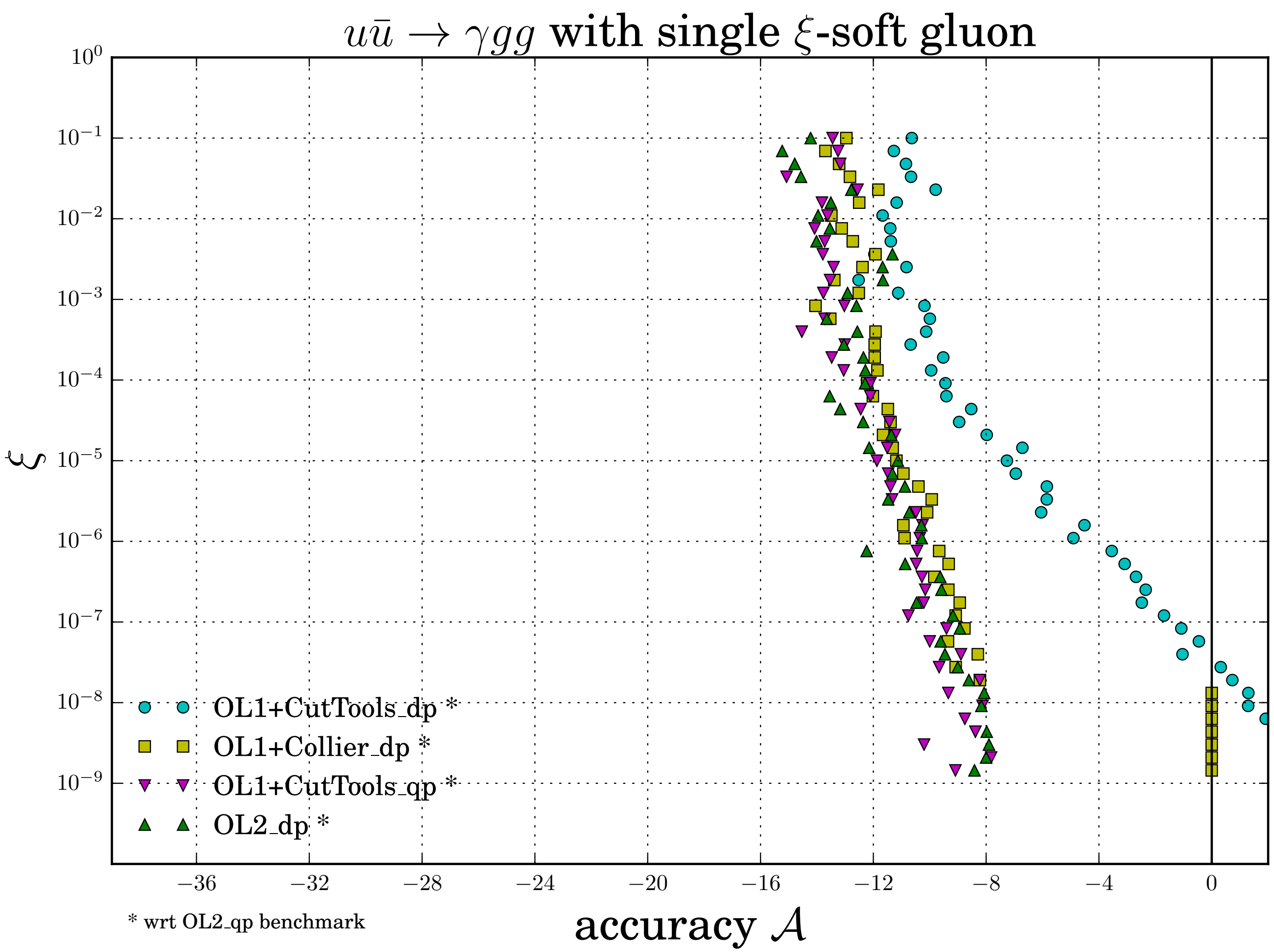
Single soft gluon with energy $E_{soft} = \xi\sqrt{\hat{s}}$. All other kinematic parameters fixed.

Stability in the soft region: $2 \to 3$ process at $\sqrt{\hat{s}} = 1$ TeV **OpenLoops 1+Cuttools (qp)**



Single soft gluon with energy $E_{soft} = \xi\sqrt{\hat{s}}$. All other kinematic parameters fixed.

Stability in the soft region: $2 \to 3$ process at $\sqrt{\hat{s}} = 1$ TeV **OpenLoops 2 (dp)**

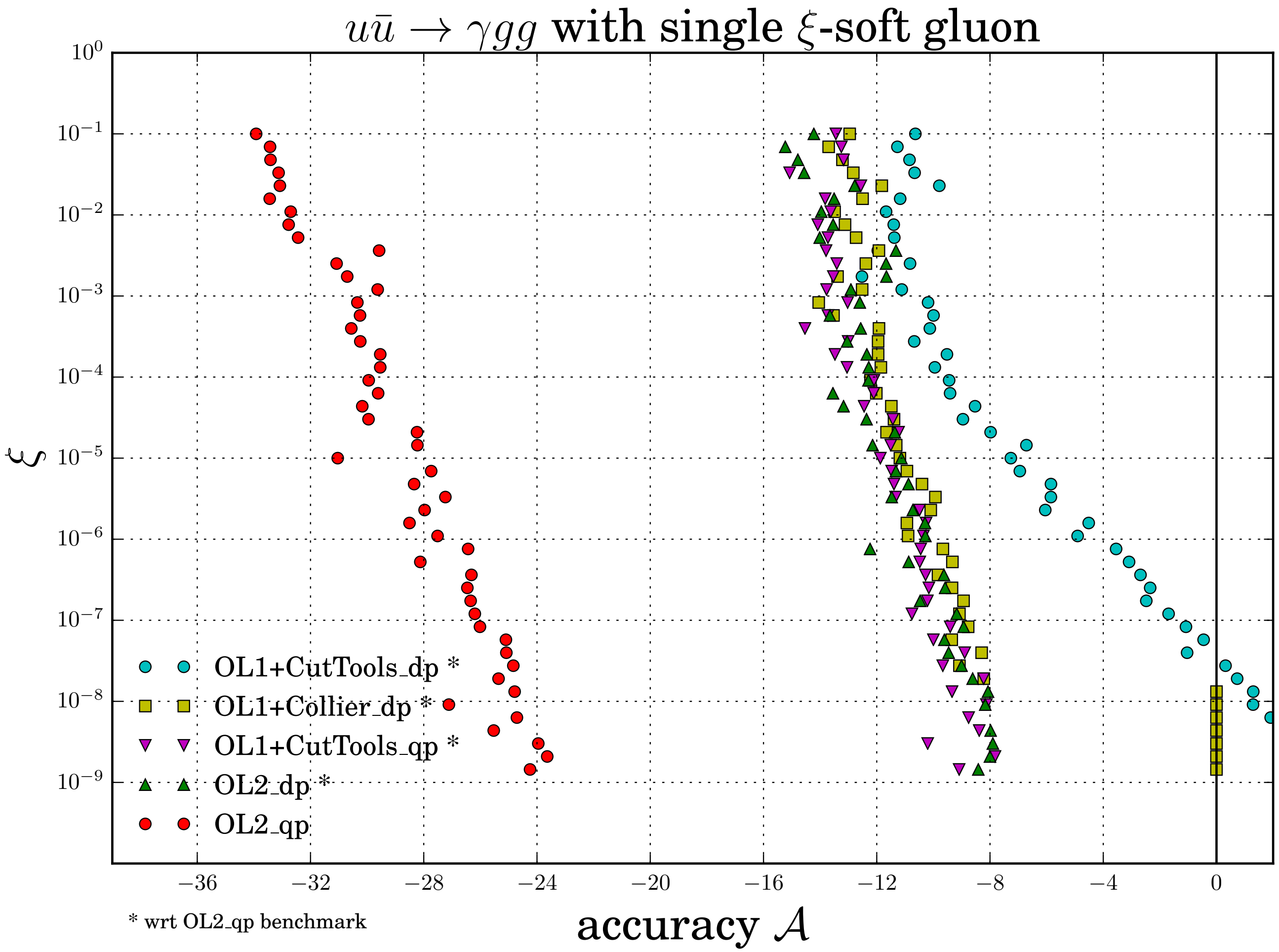


Single soft gluon with energy $E_{soft} = \xi\sqrt{\hat{s}}$. All other kinematic parameters fixed. **MI:** Collier

OpenLoops 2 double precision similarly stable as OpenLoops 1+Cuttools quad precision

Further systematic improvements for soft/collinear regions under investigation

Stability in the soft region: $2 \to 3$ process at $\sqrt{\hat{s}} = 1$ TeV **OpenLoops 2 (qp)**



Single soft gluon with energy $E_{soft} = \xi\sqrt{\hat{s}}$. All other kinematic parameters fixed. **MI:** OneLoop

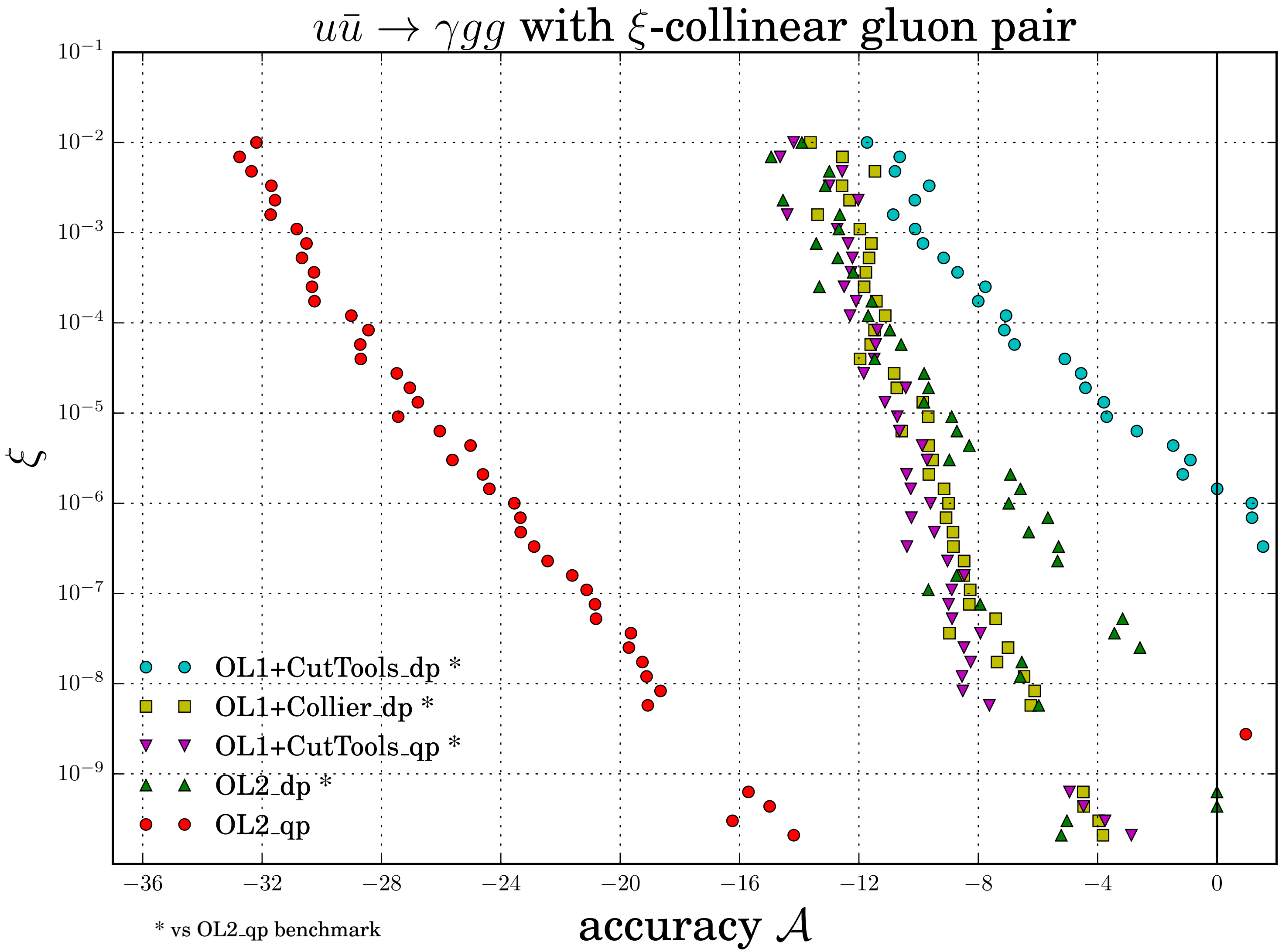
OpenLoops 2 quadruple precision yields > 20 digits in deep IR region

Further systematic improvements for soft/collinear regions under investigation

V. Summary and Outlook

- New on-the-fly algorithm: Construction and reduction of one-loop amplitudes in single recursion
  $\Rightarrow$ No external tensor reduction tools needed
- Drastic reduction of complexity at all stages of the calculation (rank ≤ 2)
- On-the-fly helicity treatment and merging $\Rightarrow$ huge gain in CPU efficiency
- Efficient treatment of numerical instability issues, e.g. with targeted any-order expansions
  $\Rightarrow$ Excellent numerical stability in the hard regions
- True quad precision benchmarks possible in this framework
- Algorithm public soon in **OpenLoops 2** (fully automated, same interface as OpenLoops 1)
- Ongoing/future projects:
  - Improvement of stability in soft and collinear regions at one loop, especially for $2 \to 4$
  - Further strong speed-up of quad precision calculations
  - Extension to two loops

Backup: Stability in the collinear region: $2 \to 3$ process at $\sqrt{\hat{s}} = 1$ TeV



Collinear gluon pair with $\xi = \theta^2$ (angle between gluon pair). All other kinematic parameters fixed.