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

M. F. Zoller
in collaboration with F. Buccioni and S. Pozzorini
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):

\[
\mathcal{W}_0 = \sum_h \sum_{col} |\mathcal{M}_0(h)|^2 \quad \text{and} \quad \mathcal{W}_1 = \sum_h \sum_{col} 2 \text{Re} \left( \mathcal{M}_0^*(h) \mathcal{M}_1(h) \right) \quad \text{and} \quad \mathcal{W}_{1}^{\text{loop-ind}} = \sum_h \sum_{col} |\mathcal{M}_1(h)|^2
\]

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

\[
\mathcal{M}_0 = \sum_d \mathcal{M}_0^{(d)} \quad \text{and} \quad \mathcal{M}_1 = \sum_d \mathcal{M}_1^{(d)}
\]

• hybrid tree-loop recursion ⇒ high CPU efficiency and numerical stability

• NLO QCD and NLO EW corrections fully implemented

• OpenLoops is interfaced to Sherpa, Powheg, Herwig, Whizard, Geneva, Munich, Matrix
• **OpenLoops 1** publicly available at [openloops.hepforge.org](openloops.hepforge.org) [Cascioli, Lindert, Maierhöfer, Pozzorini]

  – Third party tools for the tensor integral reduction to scalar MILs:
    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

\[
\text{Long-term goal: NNLO automation for } 2 \to 2 \text{ and } 2 \to 3 \text{ processes} \\
\quad - \text{2 loop amplitude construction and reduction needed } \Rightarrow \text{avoid high tensor rank complexity} \\
\quad - \text{Numerical stability at NLO for } 2 \to 4 \text{ is crucial}
\]

• **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 \(\leq 2\) at all times

  – Stability issues addressed in a targeted way
Tree level amplitudes

\[ M_0 = \sum_d M_0^{(d)} \]

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

\[ M_l^{(d)} = C_l^{(d)} A_l^{(d)}. \]

colour stripped \( A_l^{(d)} \) are split into subtrees by cutting an internal line:

⇒ Numerical merging of subtrees performed recursively:

\[
\sigma_a w_a(k_a, h_a) = \sigma_a w_b(k_b, h_b) w_c(k_c, h_c)
\]

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

⇒ Once computed subtrees used in multiple Feynman diagrams at tree and loop level
One-loop amplitude

\[ \mathcal{A}_1^{(d)} = \int d^D q \frac{\text{Tr}[\mathcal{N}(q, h)]}{D_0 D_1 \cdots D_{N-1}} = \]

\[ \begin{array}{c}
\begin{array}{c}
\text{propagators } D_i = (q + p_i)^2 - m_i^2,
\text{helicity configurations of subtree } w_i: h_i
\end{array}
\text{spinor/Lorentz indices } \beta_i \Rightarrow \text{trace: contraction with } \delta_\beta^\beta_0,
\text{helicity configurations of } \mathcal{A}_1^{(d)}: h = h_1 + \ldots + h_N
\end{array} \]

Numerator factorizes into segments:

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

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

3-point segment:

\[ [S_i(q, h_i)]_{\beta_{i-1}}^{\beta_i} = \left[ \begin{array}{c}
D_i
\end{array} \right]_{\beta_i}^{\beta_{i-1}} = \left\{ \begin{array}{c}
Y_{\sigma_i}^{i} \beta_i \left[ Z_{\nu;\sigma_i}^{i} \beta_i \right] q^\nu
\end{array} \right\} w_{i_1}^{\sigma_i}(k_i, h_i) \]

4-point segment:

\[ [S_i(q, h_i)]_{\beta_{i-1}}^{\beta_i} = \left[ \begin{array}{c}
D_i
\end{array} \right]_{\beta_i}^{\beta_{i-1}} = \left[ \begin{array}{c}
Y_{\sigma_1;\sigma_2}^{i} \beta_i
\end{array} \right] \ w_{i_1}^{\sigma_1}(k_{i_1}, h_{i_1}) w_{i_2}^{\sigma_2}(k_{i_2}, h_{i_2}) \] (\( h_i = h_{i_1} + h_{i_2} \)
The OpenLoops dressing step

define partially dressed numerator

\[ N_n(q, \hat{h}_n) = S_1(q, h_1) \cdots S_n(q, h_n) \quad (\hat{h}_n = \sum_{i=1}^{n} h_i) \]

\[ \begin{array}{c}
\w_1 \\
\downarrow k_1 \\
D_1 \\
\w_2 \\
\downarrow k_2 \\
D_2 \\
\vdots \\
\w_n \\
\downarrow k_n \\
D_n \\
\w_{n+1} \\
\downarrow k_{n+1} \\
D_{n+1} \\
\w_N \\
\downarrow k_N \\
D_0 \\
\end{array} \]

\begin{align*}
\text{dressed segments} & \quad \text{undressed segments} \\
\beta_0 & \quad \beta_N
\end{align*}

dressing step

\[ N_n(q, \hat{h}_n) = N_{n-1}(q, \hat{h}_{n-1}) S_n(q, h_n) \]

with initial condition \( N_0 = 1 \)

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

\[ N(q, \hat{h}_n) = \sum_{r=0}^{R} N_{\mu_1 \cdots \mu_r}(\hat{h}_n) q^{\mu_1} \cdots q^{\mu_r}, \]

\[ \left[ N_{\mu_1 \cdots \mu_r}(\hat{h}_n) \right]_{\beta_0}^{\beta_n} = \left[ N_{\mu_1 \cdots \mu_r}(\hat{h}_{n-1}) \right]_{\beta_0}^{\beta_{n-1}} \left[ Y_{\sigma_{n-1}}^{n} \right]_{\beta_{n-1}}^{\beta_n} + \left[ N_{\mu_2 \cdots \mu_r}(\hat{h}_{n-1}) \right]_{\beta_0}^{\beta_{n-1}} \left[ Z_{\mu_1, \sigma_{n-1}}^{n} \right]_{\beta_{n-1}}^{\beta_n} \right] w_{n}^{\sigma_{n}}(k_n, h_n) \]
Colour, helicity and diagram sums in OpenLoops 1

- for each diagram \( d \) and global helicity \( h \) configuration construct \( \text{Tr} \left[ \mathcal{N}^{(d)}_{N}(q, h) \right] \)

- colour sum with Born: \( \mathcal{V}^{(d)}_{N}(q, h) = 2 \left( \sum_{\text{col}} \mathcal{M}_{0}(h) \mathcal{C}^{(d)} \right) \text{Tr} \left[ \mathcal{N}^{(d)}_{N}(q, h) \right] \)

- helicity sum: \( \mathcal{V}^{(d)}_{N}(q) = \sum_{h} \mathcal{V}^{(d)}_{N}(q, h) \)

- sum same topology diagrams, reduce and evaluate integrals: \( \int d^{D}q \sum_{d} \frac{\text{Tr} \left[ \mathcal{V}^{(d)}_{N}(q, 0) \right]}{D_{0}, \ldots, D_{N-1}} \)

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

\[ w_{1} \quad w_{N-2} \quad \cdots \quad \rightarrow \quad \begin{cases} \begin{array}{c} \mathcal{N}_{N-2} = \quad \cdots \quad = \quad \mathcal{N}_{N-2} \\ w_{1} \quad w_{N-2} \quad w_{N-1} \quad w_{N} \end{array} \end{cases} \]

\[ = \quad \mathcal{N}_{N-2} \quad S_{N-1} \quad S_{N} \]

\[ w_{1} \quad w_{N-2} \quad w_{N-1} \quad w_{N} \]

\[ = \quad \mathcal{N}_{N-2} \quad \hat{S}_{N-1} \]

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

consider color-helicity summed numerator

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

and formulate recursion for partially dressed numerator with nested helicity sums

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

and a dressing step as

\[
\mathcal{V}_n(q, \hat{h}_n) = \sum_{h_n} \mathcal{V}_{n-1}(q, \hat{h}_{n-1}) S_n(q, h_n)
\]

⇒ Remaining helicity dof are those of the undressed segments!

Parent-child trick not possible (different colour factors) ⇒ OpenLoops Merging instead
The OpenLoops Merging

Sum partially dressed open loops

\[ \mathcal{V}_n(q, \tilde{h}_n) = \sum_{\alpha} \mathcal{V}^{(\alpha)}_n(q, \tilde{h}_n) \]

with

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

since

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

▷ 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:

- $n$: # of attached external legs
- Rank
- # of tensor coefficients

$n$: # of attached external legs

Diagram:

- Vertical axis: rank
- Horizontal axis: # of tensor coefficients
- Dashed circle with arrow
Amplitude generation and tensor reduction in OpenLoops 1

Example:

\[
\begin{array}{c|c|c|c|c|c|c|c}
 n & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\hline
\text{rank} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\end{array}
\]

$n$: # of attached external legs

$\#$ of tensor coefficients

$n$: # of attached external legs
Example:

<table>
<thead>
<tr>
<th>n</th>
<th># of attached external legs</th>
<th># of tensor coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>
Amplitude generation and tensor reduction in OpenLoops 1

Example:

<table>
<thead>
<tr>
<th>$n$</th>
<th># of tensor coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>35</td>
</tr>
</tbody>
</table>

$n$: # of attached external legs

Diagram:

- Diagram shows a graph with axes labeled $n$ and # of tensor coefficients.
- The graph plots the relationship between the number of attached external legs ($n$) and the number of tensor coefficients.
- The graph illustrates an upward trend, indicating an increase in tensor coefficients as $n$ increases.

Note: The diagram includes a physical diagram on the left, which is not described in the text but appears to be related to the example provided.
Amplitude generation and tensor reduction in OpenLoops 1

Example:

<table>
<thead>
<tr>
<th>$n$</th>
<th>Rank</th>
<th># of tensor coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>70</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>150</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>315</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>620</td>
</tr>
</tbody>
</table>

$n$: # of attached external legs

$n$: # of attached external legs

rank

# of tensor coefficients
Amplitude generation and tensor reduction in OpenLoops 1

Example:

![Diagram of Feynman diagram](image)

<table>
<thead>
<tr>
<th>n</th>
<th># of tensor coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>3</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>70</td>
</tr>
<tr>
<td>5</td>
<td>126</td>
</tr>
</tbody>
</table>

$n$: # of attached external legs

(rank vs. # of tensor coefficients graph)

$n$: # of attached external legs
Amplitude generation and tensor reduction in OpenLoops 1

Example:

- $n$: number of attached external legs
- $n$: rank
- # of tensor coefficients

<table>
<thead>
<tr>
<th>$n$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>rank</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td># of tensor coefficients</td>
<td>1</td>
<td>5</td>
<td>15</td>
<td>35</td>
<td>70</td>
<td>126</td>
<td>210</td>
</tr>
</tbody>
</table>

$n$: number of attached external legs
Amplitude generation and tensor reduction in OpenLoops 1

Example:

- Complexity grows exponentially with tensor rank

```
<table>
<thead>
<tr>
<th>n</th>
<th># of attached external legs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

```

```
<table>
<thead>
<tr>
<th>rank</th>
<th># of tensor coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>
```

Graph: OpenLoops 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]

\[
q^{\mu} q^{\nu} = A^{\mu\nu} + B^{\mu\nu} q^{\lambda}
\]

\[
= \left[ A_{-1}^{\mu\nu} + A_{0}^{\mu\nu} D_{0} \right] + \left[ B_{-1,\lambda}^{\mu\nu} + \sum_{i=0}^{3} B_{i,\lambda}^{\mu\nu} D_{i} \right] q^{\lambda}, \quad D_{i} = (q + p_{i})^{2} - m_{i}^{2}
\]

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]

\[ q^\mu q^\nu = A^{\mu\nu} + B^{\mu\nu}_\lambda q^\lambda \]

\[ = [A_{-1}^{\mu\nu} + A_0^{\mu\nu} D_0] + [B^{\mu\nu}_{-1,\lambda} + \sum_{i=0}^3 B^{\mu\nu}_{i,\lambda} D_i] q^\lambda, \quad D_i = (q + p_i)^2 - m_i^2 \]

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!)

\[
\Rightarrow \frac{\mathcal{V}^{\mu\nu} q_\mu q_\nu}{D_0 \cdots D_{N-1}} = \frac{\mathcal{V}^{\mu}_{-1} q_\mu + \mathcal{V}_{-1}}{D_0 \cdots D_{N-1}} + \sum_{i=0}^3 \frac{\mathcal{V}^{\mu}_{i} q_\mu + \mathcal{V}_{i}}{D_0 \cdots D_{i-1} D_{i+1} \cdots D_{N-1}}
\]

- \( q \)-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:

<table>
<thead>
<tr>
<th>n</th>
<th># of tensor coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

Diagram showing the relationship between rank and the number of tensor coefficients for different values of n.
Amplitude generation and tensor reduction in OpenLoops 2

Example:

<table>
<thead>
<tr>
<th>n</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>5</td>
<td>15</td>
<td>35</td>
<td>70</td>
<td>126</td>
<td>210</td>
<td>330</td>
</tr>
</tbody>
</table>

4 pinched subtopologies
Amplitude generation and tensor reduction in OpenLoops 2

Example:

<table>
<thead>
<tr>
<th># of tensor coefficients</th>
<th>rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>35</td>
<td>3</td>
</tr>
<tr>
<td>70</td>
<td>4</td>
</tr>
<tr>
<td>126</td>
<td>5</td>
</tr>
<tr>
<td>210</td>
<td>6</td>
</tr>
<tr>
<td>330</td>
<td>7</td>
</tr>
</tbody>
</table>

4 pinched subtopologies
Amplitude generation and tensor reduction in OpenLoops 2

Example:

```
<table>
<thead>
<tr>
<th>n</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td># of tensor coefficients</td>
<td>5</td>
<td>15</td>
<td>35</td>
<td>70</td>
<td>126</td>
<td>210</td>
<td>330</td>
</tr>
<tr>
<td>Rank</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>
```

4 pinched subtopologies

4 double pinched subtopologies

Graph:

- Rank vs. # of tensor coefficients
- Points at rank 1, 2, 3 connected with arrows
- Labels at the top and right axes for rank and # of tensor coefficients respectively
Amplitude generation and tensor reduction in OpenLoops 2

Example:

- 4 pinched subtopologies
- 4 double pinched subtopologies

The complexity associated with tensor rank remains small!
Amplitude generation and tensor reduction in OpenLoops 2

Example:

<table>
<thead>
<tr>
<th>n</th>
<th># of tensor coefficients</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>4</td>
<td>35</td>
</tr>
<tr>
<td>5</td>
<td>70</td>
</tr>
<tr>
<td>6</td>
<td>126</td>
</tr>
<tr>
<td>7</td>
<td>210</td>
</tr>
</tbody>
</table>

- **OpenLoops 1**: 4 pinched subtopologies
- **OpenLoops + OFR**: 4 double pinched subtopologies

Complexity associated with tensor rank remains small!
Problem: huge proliferation of topologies due to pinching of propagators

\[ \Rightarrow \frac{\mathcal{V}_{\mu\nu} q^\mu q^\nu}{D_0 \cdots D_{N-1}} = \left( \mathcal{V}_{-1} + \sum_{i=0}^{3} \mathcal{V}_i D_i \right) q^\mu + \mathcal{V}_{-1} + \mathcal{V}_0 D_0 + \mathcal{V}_{-1} \tilde{q}^2 \right) \left( \frac{1}{D_0 \cdots D_{N-1}} \right) \]

\sim \mathcal{V}_{\mu\nu} q^\mu q^\nu

\Rightarrow \text{factor } \sim 5 \text{ 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
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]

$$\frac{\mathcal{V} + \mathcal{V}_\mu q^\mu}{D_0 D_1 \cdots D_{N-1}} = \sum_{i_0<i_1<i_2<i_3}^{N-1} \frac{d(i_0i_1i_2i_3)}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}}$$

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

\[ q^\mu q^\nu = [A_{-1}^{\mu\nu} + A_0^{\mu\nu} D_0] + \left[ B_{-1,\lambda}^{\mu\nu} + \sum_{i=0}^{3} B_{i,\lambda}^{\mu\nu} D_i \right] q^\lambda \]

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

\[ A_{i}^{\mu\nu} = \frac{1}{\gamma} a_{i}^{\mu\nu}, \]
\[ B_{i,\lambda}^{\mu\nu} = \frac{1}{\gamma^2} \left[ b_{i,\lambda}^{(1)} \right]^{\mu\nu} + \frac{1}{\gamma} \left[ b_{i,\lambda}^{(2)} \right]^{\mu\nu} \]

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 \( \gamma \) 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

\[ \begin{align*}
    p_1^2 &= -p^2 < 0, \\
    p_2^2 &= -p^2(1 + \delta), \quad 0 \leq \delta \ll 1, \\
    (p_2 - p_1)^2 &= 0,
\end{align*} \]

\[ \Rightarrow \sqrt{\Delta} = \frac{p^2}{2} \delta \]

\[ \Rightarrow \gamma = -p^2 \delta^2 \]

\[ \Rightarrow \text{expand basis momenta } l_i, \text{ reduction formula and scalar integrals in } \delta, \text{ e.g. massless rank 1:} \]

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

with \( C_0(p_1, p_2, m_0, m_1, m_2) \sim \int d^D q \frac{1}{D_0 D_1 D_2} \) and \( B_0(p_1, m_0, m_1) \sim \int d^D q \frac{1}{D_0 D_1} \)

Implemented: direct expansions for the full reduction of rank \( \leq 3 \) 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

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

Factor $\sim 2$ speedup wrt OpenLoops 1 for nontrivial processes!
Stability of OpenLoops 1 and 2 in double precision: $2 \rightarrow 3$ processes (at $\sqrt{s} = 1$ TeV)

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

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

$$\Delta R_{ij} = \sqrt{(\eta_i - \eta_j)^2 + (\phi_i - \phi_j)^2}$$

- 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 $\Delta$-expansions

Soft region under investigation $\Rightarrow$ important for real-virtual part of NNLO
Stability of OpenLoops 1 and 2 in double precision: $2 \rightarrow 4$ processes (at $\sqrt{s} = 1$ 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 $2 \rightarrow 3$
- 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 $\Delta$-expansions
V. Summary and Outlook

• New algorithm for construction and reduction of 1-loop amplitudes in a single recursion

• Drastic reduction of complexity at all stages of the calculation (rank $\leq 2$)

• 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