**THE EUROPEAN
PHYSICAL JOURNAL C**

CrossMark

Regular Article - Theoretical Physics

On-the-fly reduction of open loops

Federico Buccioni[a]**, Stefano Pozzorini**[b]**, Max Zoller**[c]

Physik-Institut, Universität Zürich, 8057 Zurich, Switzerland

Abstract Building on the open-loop algorithm we introduce a new method for the automated construction of one-loop amplitudes and their reduction to scalar integrals. The key idea is that the factorisation of one-loop integrands in a product of loop segments makes it possible to perform various operations on-the-fly while constructing the integrand. Reducing the integrand on-the-fly, after each segment multiplication, the construction of loop diagrams and their reduction are unified in a single numerical recursion. In this way we entirely avoid objects with high tensor rank, thereby reducing the complexity of the calculations in a drastic way. Thanks to the on-the-fly approach, which is applied also to helicity summation and for the merging of different diagrams, the speed of the original open-loop algorithm can be further augmented in a very significant way. Moreover, addressing spurious singularities of the employed reduction identities by means of simple expansions in rank-two Gram determinants, we achieve a remarkably high level of numerical stability. These features of the new algorithm, which will be made publicly available in a forthcoming release of the OPENLOOPS program, are particularly attractive for NLO multi-leg and NNLO real–virtual calculations.

Contents

[a] e-mail: buccioni@physik.uzh.ch
[b] e-mail: pozzorin@physik.uzh.ch
[c] e-mail: zoller@physik.uzh.ch

1 Introduction

The continuous improvement of statistics and experimental systematics at the Large Hadron Collider (LHC) permits to challenge the Standard Model of particle physics at steadily increasing levels of energy and precision. In this context, the uncertainty of theoretical predictions starts playing a critical role in many areas of the physics program of the LHC, providing strong motivation for developing new techniques that make it possible to push theoretical calculations towards more complex processes and higher perturbative orders.

In the last decade, the advent of new powerful methods for the calculation of one-loop scattering amplitudes [1–9]

Springer

has opened the door to the automation of next-to-leading order (NLO) calculations. Nowadays, one-loop calculations are supported by a number of highly automated tools [10–20] that provide the key to achieve NLO precision in the context of multi-purpose Monte Carlo generators [21–27]. This recent progress has enabled NLO calculations for a huge number of processes and has extended their reach up to multi-particle final states of unprecedented complexity [28–32]. Nevertheless, in various cases the technical limitations of one-loop generators still represent a serious bottleneck or even a show stopper. These issues can be encountered in processes with many final-state particles and for kinematic configurations with two or more widely separated scales. An important example is given by the real–virtual contributions to next-to-next-to leading order (NNLO) calculations, which require very fast and highly stable one-loop amplitudes in deeply infrared regions of phase space.

Motivated by these considerations, in this paper we introduce a new method that leads to very significant efficiency and stability improvements in the construction of one-loop amplitudes. This new method builds on OPENLOOPS [9,16], a fully automated framework for the automated generation of scattering amplitudes in the Standard Model. The original implementation of the open-loop approach [9,16] supports NLO QCD [31,33–39] as well as NLO EW [40–44] calculations and is interfaced to various multi-purpose Monte Carlo tools. The OPENLOOPS program is also part of MATRIX [45] and has already been applied to several NNLO calculations [46–55]. The essence of the open-loop method [9] consists of a numerical recursion that generates cut-open loop diagrams, called open loops, by multiplying, one after the other, the various building blocks that are connected through loop propagators. More precisely, the construction of N-point loop integrands is organised through the factorisation of N loop *segments*, which consist each of a loop propagator and a corresponding external subtree. Segment multiplications are implemented through process-independent numerical routines that correspond to the Feynman rules of the model at hand. This type of recursion was first proposed in the context of off-shell recurrence relations for colour-ordered gluon-scattering amplitudes [8]. Thanks to a tensorial representation that retains the loop-momentum dependence of all building blocks, this approach can be used in combination with reduction techniques based on tensor integrals [4] or with the OPP reduction method [5], resulting in both cases in very fast computer code [9].

The new method presented in this paper exploits the factorised structure of the open-loop representation in a completely new way. The key idea is that certain operations, which are usually done when all building blocks of Feynman diagrams have been assembled, can be anticipated and performed *on-the-fly* during the construction of the diagrams. Exploiting the factorised structure of the integrands, this on-the-fly approach permits to perform various types of operations at a much lower level of complexity, thereby boosting their efficiency. As we will show, it can be exploited in order to factorise helicity summations as well as the sums over different Feynman diagrams that share the same one-loop topology. Moreover, based on the integrand reduction method by del Aguila and Pittau [2], we will introduce an on-the-fly technique for the reduction of open loops. In this way, we will promote OPENLOOPS to an algorithm that combines the construction and the reduction of loop amplitudes in a unified numerical recursion. A notable feature of this approach is that it permits to avoid high-rank objects at any stage of the calculations. More precisely, tensor integrals are always kept at rank two or lower, thereby reducing the computational complexity in a dramatic way.

The on-the-fly technique leads to very significant improvements of CPU efficiency. For what concerns numerical stability, in order to avoid severe instabilities that result from squared inverse Gram determinants in the reduction identities of [2], we present a method that isolates such instabilities in certain triangle topologies and circumvents them via analytic expansions in the limit of small Gram determinants. In this way we obtain the first integrand-reduction algorithm that is essentially free from Gram-determinant instabilities. The achieved level of stability in double precision is competitive with the most sophisticated tools on the market [19] and with public implementations of OPP reduction in quadruple precision.

The paper is organized as follows. In Sect. 2 we review the original open-loop method. The on-the-fly approach is introduced in Sect. 3 for the case of helicity sums and for the merging of topologically equivalent open loops. In Sect. 4 the on-the-fly approach is generalised to the reduction of open loops. Details on the employed integrand-reduction identities and our treatment of Gram-determinant instabilities are discussed in Sect. 5. The entire algorithm and its implementation are outlined in Sect. 6, where we also present technical studies on the CPU performance and numerical stability. Our conclusions are presented in Sects. 7, and Appendix A deals with low-rank integrals that remain to be solved at the end of the on-the-fly recursion.

2 The open-loop method

In this section we review the original open-loop method [9], which is implemented in the publicly available OPENLOOPS 1 program [16]. At variance with the original publication [9], here we refine various aspects of the notation and we adopt a particular perspective that sets the stage for the new methods introduced in Sects. 3–5. These new techniques are going to become publicly available in the OPENLOOPS 2 release.

2.1 Helicity and colour bookkeeping

The task carried out by the open-loop algorithm is the calculation of the tree-level and one-loop contributions to the scattering probability density,

$$\mathcal{W}_{\text{tree}} = \sum_h \sum_{\text{col}} |\mathcal{M}_0(h)|^2,$$

$$\mathcal{W}_{1-\text{loop}} = \sum_h \sum_{\text{col}} 2\,\text{Re}\Big[\mathcal{M}_0^*(h)\mathcal{M}_1(h)\Big], \qquad (1)$$

or the squared one-loop contribution

$$\mathcal{W}_{1-\text{loop}^2} = \sum_h \sum_{\text{col}} |\mathcal{M}_1(h)|^2, \qquad (2)$$

for loop-induced processes. The polarised matrix elements $\mathcal{M}_0(h)$ and $\mathcal{M}_1(h)$ should be understood as generic tree and one-loop amplitudes, in the sense that the techniques presented in this paper are applicable to any renormalisable theory, including the QCD and electroweak sectors of the Standard Model, as well as BSM theories. The sums in (1)–(2) run over all helicity and colour degrees of freedom of the scattering particles. While colour indices are kept implicit, the helicity dependence is characterised by a single index h, which corresponds to the global helicity configuration of the event, as described below.

Scattering amplitudes are computed as sums of Feynman diagrams,

$$\mathcal{M}_0(h) = \sum_{\mathcal{I} \in \Omega_{\text{tree}}} \mathcal{M}_0(\mathcal{I}, h),$$

$$\mathcal{M}_1(h) = \sum_{\mathcal{I} \in \Omega_{1-\text{loop}}} \mathcal{M}_1(\mathcal{I}, h), \qquad (3)$$

where Ω_{tree} and $\Omega_{1-\text{loop}}$ stand for the sets of tree and one-loop diagrams. Each tree and one-loop diagram can be factored into a colour factor $\mathcal{C}(\mathcal{I})$ and a colour-stripped diagram amplitude,[1]

$$\mathcal{M}_L(\mathcal{I}, h) = \mathcal{C}(\mathcal{I})\,\mathcal{A}_L(\mathcal{I}, h), \qquad (4)$$

for $L = 0, 1$. The colour-stripped amplitudes $\mathcal{A}_L(\mathcal{I}, h)$ are the main source of complexity. In the open-loop approach, their calculation is addressed with numerical recursions as described in Sects. 2.2–2.4. For what concerns colour factors, exploiting the factorisation properties (4), all relevant operations can be reduced to the calculation of colour-summed interference terms of the form

$$\mathcal{K}(\mathcal{I}_a, \mathcal{I}_b) = \sum_{\text{col}} \mathcal{C}(\mathcal{I}_a)^* \mathcal{C}(\mathcal{I}_b), \qquad (5)$$

---

[1] Quartic gluon couplings involve three independent colour structures. Thus each diagram involving k quartic couplings needs to be split into 3^k contributions of type (4) that are effectively handled as separate diagrams.

which appear in the calculation of the scattering probabilities (1)–(2). This task must be addressed only once per process. It is handled by algebraically reducing all $\mathcal{C}(\mathcal{I})$ to a standard basis $\{\mathcal{C}_i\}$ and relating the terms (5) to the interference matrix [9,56]

$$\mathcal{K}_{ij} = \sum_{\text{col}} \mathcal{C}_i^* \mathcal{C}_j. \qquad (6)$$

In OPENLOOPS we use a basis where all colour factors are expressed through products and traces of the SU(3) generators T_{ij}^a in the fundamental representation.

For the bookkeeping of external momenta and helicities in a process with N_p scattering particles we introduce the set of particle indices

$$\mathcal{E} = \{1, 2, \ldots, N_\text{p}\}. \qquad (7)$$

To characterise the helicity configurations s of individual particles we use labels

$$\lambda_i = \begin{cases} 1, 3 & \text{for fermions with } s = -1/2, 1/2 \\ 1, 2, 3 & \text{for gauge bosons with } s = -1, 0, 1 \\ 0 & \text{for scalars with } s = 0 \end{cases} \qquad (8)$$

$\forall\, i \in \mathcal{E}$. The configuration $\lambda_i = 0$ will also be used to characterise unpolarised particles, i.e. fermions or gauge bosons whose helicity is still unassigned or has already been summed over. Since a particle can have up to four different helicity states, it is convenient to adopt a helicity numbering scheme based on the labels

$$\bar{h}_i = \lambda_i\, 4^{i-1}, \qquad (9)$$

which correspond to a quaternary number with $\lambda_i \in \{0, 1, 2, 3\}$ as ith-last digit and all other digits equal to zero. In this way, the helicity configurations $(\lambda_1, \ldots, \lambda_{N_\text{p}})$ of the full event can be uniquely identified with the label

$$h = \bar{h}_1 + \cdots + \bar{h}_{N_\text{p}}, \qquad (10)$$

which corresponds to a quaternary number of N_p digits, each of which describes the helicity of a particular external particle. Let us also introduce the single-particle helicity spaces, $\bar{\mathcal{H}}_i \ni \bar{h}_i$, defined as

$$\bar{\mathcal{H}}_i = \begin{cases} \{\lambda_i\, 4^{i-1} | \lambda_i = 1, 3\} & \text{for fermions or mass-less gauge bosons,} \\ \{\lambda_i\, 4^{i-1} | \lambda_i = 1, 2, 3\} & \text{for massive gauge bosons,} \\ \{0\} & \text{for scalars,} \end{cases} \qquad (11)$$

where we do not include unpolarised states. Finally, the global helicity space for the full set of scattering particles, $\mathcal{H} \ni h$, is defined as

$$\mathcal{H} = \bar{\mathcal{H}}_1 \otimes \cdots \otimes \bar{\mathcal{H}}_{N_\text{p}}, \qquad (12)$$

where the product is understood as $\mathcal{A} \otimes \mathcal{B} = \{a + b | a \in \mathcal{A}, b \in \mathcal{B}\}$.

2.2 Tree amplitudes

At tree level, each colour-stripped Feynman diagram is constructed by contracting two so-called subtrees, which arise by cutting the diagram in two pieces in correspondence of an internal propagator,

$$\mathcal{A}_0(\mathcal{I}, h) = \quad (13)$$

A generic subtree w_a corresponds to the part of a certain Feynman diagram that connects an internal off-shell line with outgoing momentum k_a to a subset of external particles,

$$\mathcal{E}_a = \{\alpha_{a1}, \ldots, \alpha_{an_a}\} \subset \mathcal{E}. \quad (14)$$

In the numbering scheme (9)–(10), the helicity configurations of a subtree are labeled

$$h_a = \bar{h}_{\alpha_{a1}} + \cdots + \bar{h}_{\alpha_{an_a}}, \quad (15)$$

and the corresponding helicity space, $\mathcal{H}_a \ni h_a$, is defined as

$$\mathcal{H}_a = \bar{\mathcal{H}}_{\alpha_{a1}} \otimes \cdots \otimes \bar{\mathcal{H}}_{\alpha_{an_a}}. \quad (16)$$

Subtrees are represented as complex n-tuples, $w_a^{\sigma_a}(k_a, h_a)$, where σ_a is the spinor or Lorentz index of the cut line. With this notation, the contraction (13) takes the form

$$\mathcal{A}_0(\mathcal{I}, h) = w_a^{\sigma_a}(k_a, h_a) \, \delta_{\sigma_a \sigma_b} \widetilde{w}_b^{\sigma_b}(k_b, h_b), \quad (17)$$

where $k_b = -k_a$, $h = h_a + h_b$, and summation over repeated indices is implicitly understood. The propagator associated with the cut line is included only in the subtree w_a and not in $\widetilde{w}_b$.

Subtrees are constructed by means of a numerical recursion that starts from the external wave functions and recursively merges subtrees by attaching their off-shell lines to the vertices that occur in the various tree diagrams. A recursion step for the case of a generic three-particle vertex is depicted[2] in Fig. 1. Its algebraic form reads

$$w_a^{\sigma_a}(k_a, h_a) = \frac{X_{\sigma_b \sigma_c}^{\sigma_a}(k_b, k_c)}{k_a^2 - m_a^2} \, w_b^{\sigma_b}(k_b, h_b) w_c^{\sigma_c}(k_c, h_c), \quad (18)$$

where the tensor $X_{\sigma_b \sigma_c}^{\sigma_a}$ describes the vertex that connects w_b and w_c to w_a, as well as the numerator of the propagator that connects to w_a. The related denominator, $(k_a^2 - m_a^2)$, appears explicitly in (18).

The momentum of the resulting subtree is $k_a = k_b + k_c$ and its helicity is $h_a = h_b + h_c$. Each recursion step must

---

[2] To draw Feynman diagrams we use AXODRAW [57].
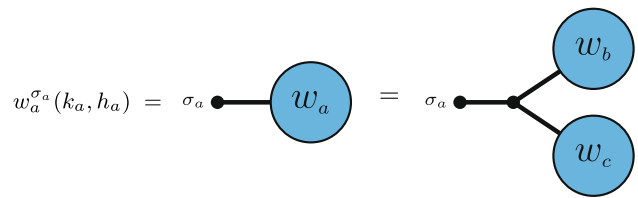
Fig. 1 Diagrammatic representation of a subtree and its numerical construction through the recurrence relation (18). The outgoing momentum k_a and the spin or Lorentz index σ_a are associated with the off-shell internal line, which is shown explicitly, while the on-shell external particles with helicity h_a are implicitly understood

be repeated for all independent helicity configuration $h_a \in \mathcal{H}_a = \mathcal{H}_b \otimes \mathcal{H}_c$. The corresponding recursion step for quartic vertices reads

$$w_a^{\sigma_a}(k_a, h_a) = \frac{X_{\sigma_b \sigma_c \sigma_d}^{\sigma_a}(k_b, k_c, k_d)}{k_a^2 - m_a^2} \, w_b^{\sigma_b}(k_b, h_b)$$
$$\times w_c^{\sigma_c}(k_c, h_c) \, w_d^{\sigma_d}(k_d, h_d). \quad (19)$$

The recursion ends when all off-shell propagators that have been cut in the beginning can be reconnected, such as to obtain the colour-stripped amplitudes (17) for the full set of tree diagrams.

Note that (18)–(19) are analogous to Berends–Giele recurrence relations for off-shell currents [58]. However, while each subtree corresponds to a single topology, off-shell currents incorporate all possible subtrees associated with a certain internal line. The inefficiency due to the usage of individual subtrees is compensated, especially at one-loop level, by the optimisation opportunities that result from the colour-factorisation identities (4) and from the fact that each subtree can occur in multiple Feynman diagrams at tree and loop level.

2.3 One-loop amplitudes

The amplitude of a colour-stripped N-point one-loop diagram, $\mathcal{I}_N$, has the general form

$$\bar{\mathcal{A}}_1(\mathcal{I}_N, h) = \qquad =$$

$$= \int \mathrm{d}^D \bar{q} \, \frac{\mathrm{Tr}\left[\bar{\mathcal{N}}(\mathcal{I}_N, \bar{q}, h)\right]}{\bar{D}_0 \bar{D}_1 \cdots \bar{D}_{N-1}}. \quad (20)$$

Symbols carrying a bar denote quantities in $D = 4 - 2\varepsilon$ dimensions, and for the loop momentum $\bar{q}$ we adopt the decomposition

$$\bar{q} = q + \tilde{q}, \tag{21}$$

where q and $\tilde{q}$ denote its four-dimensional and $(D - 4)$-dimensional parts, respectively. The denominators read

$$\bar{D}_i = (\bar{q} + p_i)^2 - m_i^2, \quad p_i = \sum_{j=1}^{i} k_j, \tag{22}$$

where k_j is the external momentum flowing into the loop at the jth loop vertex. Internal momenta are chosen such that $p_0 = p_N = 0$, i.e. the momentum flowing through the $\bar{D}_0 = \bar{D}_N$ propagator is $\bar{q}$. The one-loop diagram $\mathcal{I}_N$ in (20) can be regarded as a sequence of loop segments,

$$\mathcal{I}_N = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_N\}, \tag{23}$$

where the segment $\mathcal{S}_i$ consists of a subtree w_i that involves a certain set $\mathcal{E}_i$ of external particles and is connected to the ith loop vertex, v_i, and to the adjacent loop propagator associated with $\bar{D}_i$. Segments associated to a quartic vertex involve two subtrees, w_{i_1} and w_{i_2}. The helicity configurations of the whole diagram are related to the ones of individual segments, $h_i \in \mathcal{H}_i$, via

$$h = h_1 + \cdots + h_N. \tag{24}$$

The trace in (20) stands for the full contraction of the spinor and Lorentz indices of propagators and vertices along the loop. In general, the numerator $\bar{\mathcal{N}}(\bar{q})$ consists of a 4-dimensional part $\mathcal{N}(q)$ and an ε-dependent remnant $\widetilde{\mathcal{N}}(\bar{q})$,

$$\text{Tr}\left[\bar{\mathcal{N}}(\bar{q})\right] = \text{Tr}\left[\mathcal{N}(q)\right] + \text{Tr}\left[\widetilde{\mathcal{N}}(\bar{q})\right]. \tag{25}$$

The terms that result from $\widetilde{\mathcal{N}}(\bar{q})$ are known as rational terms of type R_2 and can be reconstructed separately as counterterms using appropriate Feynman rules [59–62]. Thus, the full amplitude can be decomposed as

$$\bar{\mathcal{A}}_1(h) = \mathcal{A}_1(h) + \mathcal{A}_{1,R_2}(h), \tag{26}$$

and in the following we focus on the nontrivial part

$$\mathcal{A}_1(\mathcal{I}_N, h) = \int d^D\bar{q} \, \frac{\text{Tr}\left[\mathcal{N}(\mathcal{I}_N, q, h)\right]}{\bar{D}_0 \bar{D}_1 \cdots \bar{D}_{N-1}}, \tag{27}$$

which stems from the four-dimensional part of the numerator but involves the D-dimensional denominators (22).

In the open-loop approach, loop diagrams are cut-open in correspondence of the $\bar{D}_0$ propagator, in the sense that the loop numerator is constructed as a tensor,

$$\left[\mathcal{N}(\mathcal{I}_N, q, h)\right]^{\beta_N}_{\beta_0} = \quad \text{} \tag{28}$$

where β_0 and β_N are the spinor or Lorentz indices associated with the cut propagator. We use the Feynman gauge, which means that the numerator of the gluon propagator is simply $-ig_\alpha^\beta$. Once $\left[\mathcal{N}(\mathcal{I}_N, q, h)\right]^{\beta_N}_{\beta_0}$ is determined, we take its trace,

$$\text{Tr}\left[\mathcal{N}(\mathcal{I}_N, q, h)\right] = \delta^{\beta_0}_{\beta_N} \left[\mathcal{N}(\mathcal{I}_N, q, h)\right]^{\beta_N}_{\beta_0}, \tag{29}$$

where summation over repeated indices is implicitly understood.

A key feature of the open-loop approach is that, similarly to the product of loop denominators $\bar{D}_0 \cdots \bar{D}_{N-1}$ in (27), the loop numerator (28) is factored into a product of *segments*,

$$\mathcal{N}(\mathcal{I}_N, q, h) = S_1(q, h_1) \cdots S_N(q, h_N). \tag{30}$$

Here and in the following, the matrix structure is implicitly understood, i.e. (30) should be interpreted as
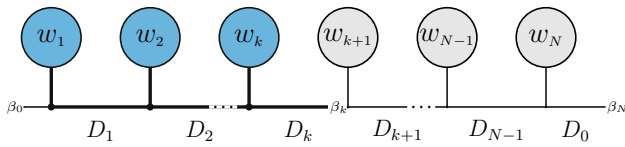
$$\left[\mathcal{N}(\mathcal{I}_N, q, h)\right]^{\beta_N}_{\beta_0} = \left[S_1(q, h_1)\right]^{\beta_1}_{\beta_0} \left[S_2(q, h_2)\right]^{\beta_2}_{\beta_1} \\ \cdots \left[S_N(q, h_N)\right]^{\beta_N}_{\beta_{N-1}}. \tag{31}$$

Segments involving a triple vertex have the generic form

$$\left[S_i(q, h_i)\right]^{\beta_i}_{\beta_{i-1}} = X^{\beta_i}_{\beta_{i-1}\sigma_i}(q + p_{i-1}, k_i) \, w_i^{\sigma_i}(k_i, h_i), \tag{32}$$

where $w_i^{\sigma_i}(k_i, h_i)$ is the corresponding external subtree. The tensor $X^{\beta_i}_{\beta_{i-1}\sigma_i}(q + p_{i-1}, k_i)$ corresponds to the interaction term in (18) and embodies the q-dependent contributions of the loop vertex v_i and of the numerator of the adjacent D_i propagator. In renormalisable theories, each segment $S_i(q, h_i)$ is a q-polynomial of rank $R \leq 1$. In the SM, the structure of three-point vertices is

$$\left[S_i(q, h_i)\right]^{\beta_i}_{\beta_{i-1}} = \quad \text{} = \\ = \left\{\left[Y^i_{\sigma_i}\right]^{\beta_i}_{\beta_{i-1}} + \left[Z^i_{\nu;\sigma_i}\right]^{\beta_i}_{\beta_{i-1}} q^\nu\right\} w_i^{\sigma_i}(k_i, h_i), \tag{33}$$

Fig. 2 Diagrammatic representation of an N-point open loop with k dressed and $N - k$ undressed segments. The segment containing the last subtree, $w_N \equiv w_0$, is associated with the propagator $\bar{D}_N \equiv \bar{D}_0 = \bar{q}^2 - m_0^2$

while four-point vertices have rank zero,

$$
\begin{aligned}
\left[S_i(q, h_i) \right]_{\beta_{i-1}}^{\beta_i} &= \\
&= \left[Y_{\sigma_1 \sigma_2}^i \right]_{\beta_{i-1}}^{\beta_i} w_{i_1}^{\sigma_1}(k_{i_1}, h_{i_1}) \, w_{i_2}^{\sigma_2}(k_{i_2}, h_{i_2}),
\end{aligned} \tag{34}
$$

with $h_i = h_{i_1} + h_{i_2}$ and $k_i = k_{i_1} + k_{i_2}$.

The numerator (30) is built as a sequence of N segment multiplications, and we refer to such a multiplication as the dressing of a segment. In the following, we will represent the state of the numerator after k dressing steps as,

$$
\begin{aligned}
\mathcal{N}(\mathcal{I}_N, q, h) &= \mathcal{N}_N(\mathcal{I}_N, q, h) \\
&= \mathcal{N}_k(\mathcal{I}_N, q, \hat{h}_k) S_{k+1}(q, h_{k+1}) \cdots S_N(q, h_N),
\end{aligned} \tag{35}
$$

where $S_{k+1}, \ldots, S_N$ are the still undressed segments, and

$$
\mathcal{N}_k(\mathcal{I}_N, q, \hat{h}_k) = S_1(q, h_1) \cdots S_k(q, h_k) \tag{36}
$$

is a q-polynomial of rank $R \le k$ that incorporates the k dressed segments. The symbol $\hat{h}_k$ and its counterpart $\check{h}_k = h - \hat{h}_k$ denote, respectively, the helicity configurations of the dressed and undressed parts of a diagram with k dressed segments and $N - k$ undressed ones. They are defined through

$$
\underbrace{h_1 + \cdots + h_k}_{\hat{h}_k} + \underbrace{h_{k+1} + \cdots + h_N}_{\check{h}_k} = h, \tag{37}
$$

where h is the global helicity state.

The corresponding helicity spaces, $\hat{\mathcal{H}}_k$ and $\check{\mathcal{H}}_k$, are defined by

$$
\underbrace{\mathcal{H}_1 \otimes \cdots \otimes \mathcal{H}_k}_{\hat{\mathcal{H}}_k} \otimes \underbrace{\mathcal{H}_{k+1} \otimes \cdots \otimes \mathcal{H}_N}_{\check{\mathcal{H}}_k} = \mathcal{H}. \tag{38}
$$

The q-dependent polynomials (36) are denoted *open loops*, and this notion implicitly includes also the corresponding undressed segments $S_{k+1}, \ldots, S_N$ and loop denominators $\bar{D}_0, \ldots, \bar{D}_N$. A graphical representation of a generic open loop with k dressed segments is depicted in Fig. 2.

The dressing of open loops is implemented through a numerical recursion

$$
\mathcal{N}_k(\mathcal{I}_N, q, \hat{h}_k) = \mathcal{N}_{k-1}(\mathcal{I}_N, q, \hat{h}_{k-1}) S_k(q, h_k), \tag{39}
$$

where $\hat{h}_k = \hat{h}_{k-1} + h_k$. This operation needs to be performed separately for all relevant helicity configurations $\hat{h}_k \in \hat{\mathcal{H}}_k = \hat{\mathcal{H}}_{k-1} \otimes \mathcal{H}_k$ and iterated for $k = 1, \ldots, N$. The initial condition is

$$
\mathcal{N}_0(\mathcal{I}_N, q, \hat{h}_0) = \mathbb{1}, \tag{40}
$$

where $\hat{h}_0 \in \hat{\mathcal{H}}_0 = \{0\}$, and the identity operator is understood as $[\mathbb{1}]_\beta^{\beta'} = \delta_\beta^{\beta'}$.

In order to capture the full q-dependence of open-loop polynomials we use the tensorial representation

$$
\mathcal{N}_k(\mathcal{I}_N, q, \hat{h}_k) = \sum_{r=0}^{R} \mathcal{N}_{k; \, \mu_1 \ldots \mu_r}(\mathcal{I}_N, \hat{h}_k) \, q^{\mu_1} \cdots q^{\mu_r}, \tag{41}
$$

and numerical operations are always performed at the level of the tensor coefficients $\mathcal{N}_{k; \, \mu_1 \ldots \mu_r}(\mathcal{I}_N, \hat{h}_k)$. In particular, the explicit form of a step of the dressing recursion (39) is

$$
\begin{aligned}
&\left[\mathcal{N}_{k; \, \mu_1 \cdots \mu_r}(\mathcal{I}_N, \hat{h}_k) \right]_{\beta_0}^{\beta_k} \\
&= \left\{ \left[\mathcal{N}_{k-1; \, \mu_1 \cdots \mu_r}(\mathcal{I}_N, \hat{h}_{k-1}) \right]_{\beta_0}^{\beta_{k-1}} \left[Y_{\sigma_k}^k \right]_{\beta_{k-1}}^{\beta_k} \right. \\
&\quad \left. + \left[\mathcal{N}_{k; \, \mu_2 \cdots \mu_r}(\mathcal{I}_N, \hat{h}_{k-1}) \right]_{\beta_0}^{\beta_{k-1}} \left[Z_{\mu_1; \sigma_k}^k \right]_{\beta_{k-1}}^{\beta_k} \right\} \\
&\quad \times w_k^{\sigma_k}(k_k, h_k)
\end{aligned} \tag{42}
$$
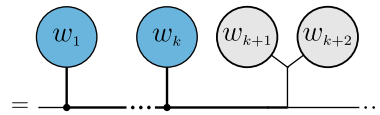
for a three-point vertex as defined in (33). For an efficient implementation the $\mu_1 \cdots \mu_r$ indices shall be symmetrised throughout.

2.4 Parent–child relations and cutting rule

The original open-loop algorithm can be boosted by using parts of pre-computed $(N - 1)$-point diagrams as a starting point for the construction of more involved N-point diagrams. This approach is based on so-called parent–child relations, which connect open loops of type

$$
\mathcal{N}(\mathcal{I}_N, q, h) =
$$

$$
\tag{43}
$$

and

$$\mathcal{N}(\tilde{\mathcal{I}}_{N-1}, q, h) =$$

$$(44)$$

that start with identical segments $\mathcal{S}_1, \ldots, \mathcal{S}_k$. Since open loops are colour-stripped, i.e. they do not depend on the different colour factors of the loop diagrams

$$\mathcal{I}_N = \{\mathcal{S}_1, \ldots \mathcal{S}_k, \mathcal{S}_{k+1}, \ldots, \mathcal{S}_N\} \quad \text{and}$$
$$\tilde{\mathcal{I}}_{N-1} = \{\mathcal{S}_1, \ldots \mathcal{S}_k, \tilde{\mathcal{S}}_{k+1}, \ldots, \tilde{\mathcal{S}}_{N-1}\}, \qquad (45)$$

it is clear that the dressed parts of (43) and (44) remain identical up to step k of the recursion, i.e.

$$\mathcal{N}_k(\mathcal{I}_N, q, \hat{h}_k) = \mathcal{N}_k(\tilde{\mathcal{I}}_{N-1}, q, \hat{h}_k). \qquad (46)$$

This allows one to construct the more involved N-point parent diagram (43) using a building block of the simpler $(N-1)$-point child diagram (44). In general, such relations can be applied for any k with $2 \le k \le N-2$, and the maximum gain in efficiency is obtained when $k = N-2$, so that only the last two segments of the parent diagram remain to be dressed.

The availability of child diagrams of type (44) is an obvious prerequisite for the applicability of the parent–child approach, and in QCD most one-loop diagrams turn out to be the parent of a corresponding child. Moreover, the correspondence between the first k dressed segments in (43) and (44) requires an appropriate *cutting rule*, i.e. a prescription that determines the cut propagator and the dressing direction in a similar way as in (43)–(44).

To this end, for each segment $\mathcal{S}_i$ with external particles $\mathcal{E}_i = \{\alpha_{i1}, \ldots, \alpha_{in_i}\}$ we introduce a binary weight defined as the sum of the weights $2^{\alpha-1}$ for each particle α, i.e.

$$F(\mathcal{S}_i) = \sum_{\alpha \in \mathcal{E}_i} 2^{\alpha-1}. \qquad (47)$$

For example, $F(\mathcal{S}_i) = 2^0 + 2^1 + 2^3 = 11$ for a segment connected to the external legs $\mathcal{E}_i = \{1, 2, 4\}$. For the merging of subtrees $\mathcal{S}_i$ and $\mathcal{S}_j$ into a single segment $\mathcal{S}_i \oplus \mathcal{S}_j$ with external legs $\mathcal{E}_i \cup \mathcal{E}_j$, the weight function obeys the useful distributive property

$$F(\mathcal{S}_i \oplus \mathcal{S}_j) = F(\mathcal{S}_i) + F(\mathcal{S}_j). \qquad (48)$$

This implies that merged segments always outweigh the original segments. Based on this feature, for N-point diagrams we adopt the cutting rule [9]

$$F(\mathcal{S}_k) > F(\mathcal{S}_1) \quad \forall \; k > 1, \qquad \text{(selection rule)} \qquad (49)$$
$$F(\mathcal{S}_N) > F(\mathcal{S}_2). \qquad \text{(direction rule)} \qquad (50)$$

The fact that the first segment is identified as the one with lowest weight guarantees its stability with respect to the merging of $\mathcal{S}_{k+1} \oplus \mathcal{S}_{k+2}$ in (43)–(44), while (50) guarantees the stability of the dressing direction for all configurations with $k \ge 2$. In this way, the parent–child approach permits to recycle the longest possible open loops.

Note that relations of type (46) can be exploited also for diagrams that involve the same number N of loop propagators and identical dressed segments, but different undressed ones.

2.5 Helicity treatment and reduction to scalar integrals

In the following we discuss the operations that are required in order to determine the contribution of a loop diagram $\mathcal{I}_N$ to the scattering probability density (1), starting form the output of the open-loop recursion, i.e. from an open-loop numerator (35) with $k = N$ dressed segments.

Instead of proceeding via a direct construction of the one-loop amplitude $\mathcal{A}_1(\mathcal{I}_N, h)$ defined in (27), we start with the associated colour structure $\mathcal{C}(\mathcal{I}_N)$ defined in (4), and we proceed by building the colour-summed interference with the Born amplitude,

$$\mathcal{U}_0(\mathcal{I}_N, h) = 2 \left(\sum_{\text{col}} \mathcal{M}_0^*(h) \, \mathcal{C}(\mathcal{I}_N) \right) \quad \forall \, h \in \mathcal{H}, \qquad (51)$$

combining it with the trace of the colour-stripped loop numerator,

$$\mathcal{U}(\mathcal{I}_N, q, h) = \mathcal{U}_0(\mathcal{I}_N, h) \text{Tr}\Big[\mathcal{N}(\mathcal{I}_N, q, h)\Big], \qquad (52)$$

and performing helicity sums,

$$\mathcal{U}(\mathcal{I}_N, q, 0) = \sum_h \mathcal{U}(\mathcal{I}_N, q, h). \qquad (53)$$

Here we use $h = 0$ for the configuration where all particles are unpolarised, in the sense that their helicities have been summed over. The above operations are performed at the level of q-coefficients in the tensorial representation (41), i.e. in practice we compute

$$\mathcal{U}_{\mu_1 \cdots \mu_r}(\mathcal{I}_N, 0) = \sum_h \mathcal{U}_{\mu_1 \cdots \mu_r}(\mathcal{I}_N, h)$$
$$= \sum_h \mathcal{U}_0(\mathcal{I}_N, h) \, \text{Tr}\Big[\mathcal{N}_{\mu_1 \cdots \mu_r}(\mathcal{I}_N, h)\Big]. \quad (54)$$

After the summation over colours and helicities it is possible to combine all diagrams with the same one-loop topology, i.e. diagrams of type

$$\mathcal{I}_N^{\alpha_1 \ldots \alpha_N} = \{\mathcal{S}_1^{\alpha_1}, \ldots, \mathcal{S}_N^{\alpha_N}\} \qquad (55)$$

with all possible combinations of segments,

$$\mathcal{S}_i \equiv \big\{\mathcal{S}_i^{\alpha_i} \big| \alpha_i = 1, \ldots, m_i\big\}, \qquad (56)$$

Fig. 3 Schematic representation of the on-the-fly helicity sums in (62). Taking the interference with the Born amplitude makes it possible to sum over the helicities $h_1, \ldots, h_k$ of the first k dressed segments of an open loop, while the remaining segments are still undressed
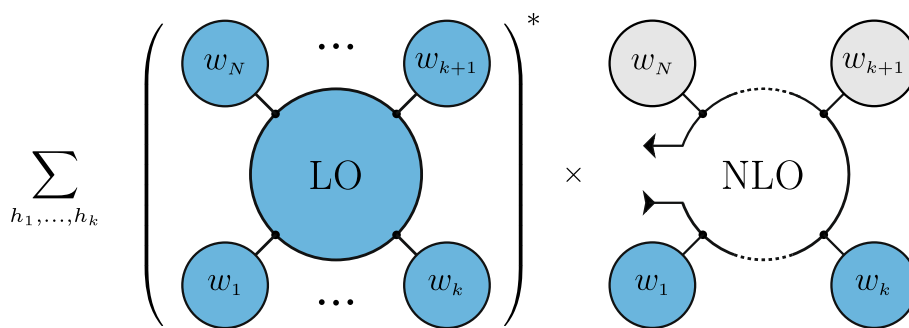
that have the same external legs $\mathcal{E}_i$ and loop propagators $\bar{D}_i$ but different external subtrees $w_i^{\alpha_i}$ and/or loop vertices $v_i^{\alpha_i}$. To filter out combinations of segments that are not allowed by the Feynman rules we introduce the tensor

$$\delta_{\alpha_1 \ldots \alpha_N} = \begin{cases} 1 & \text{if } \mathcal{S}_1^{\alpha_1}, \ldots, \mathcal{S}_N^{\alpha_N} \text{ form a} \\ & \text{valid one-loop diagram} \\ 0 & \text{else}. \end{cases} \quad (57)$$

In this way, the full set of topologically equivalent one-loop diagrams can be defined as

$$\Omega_N = \left\{ \mathcal{I}_N^{\alpha_1 \cdots \alpha_N} \middle| \alpha_i = 1, \ldots, m_i \text{ with } \delta_{\alpha_1 \ldots \alpha_N} \neq 0 \right\}, \quad (58)$$

and their sum yields

$$\mathcal{V}_{\mu_1 \ldots \mu_r}(\Omega_N, 0) \\ = \sum_{\alpha_1 \ldots \alpha_N} \sum_h \delta_{\alpha_1 \ldots \alpha_N} \mathcal{U}_{\mu_1 \ldots \mu_r}(\mathcal{I}_N^{\alpha_1 \ldots \alpha_N}, h). \quad (59)$$

The contribution of the diagrams (58) to the scattering probability (1) reads

$$\mathcal{W}_{1-\text{loop}} = \text{Re}\left\{ \int \mathrm{d}^D \bar{q} \, \frac{\mathcal{V}(\Omega_N, q, 0)}{\bar{D}_0 \cdots \bar{D}_{N-1}} \right\} \\ = \text{Re}\left\{ \sum_{r=0}^{R} \mathcal{V}_{\mu_1 \cdots \mu_r}(\Omega_N, 0) \int \mathrm{d}^D \bar{q} \, \frac{q^{\mu_1} \cdots q^{\mu_r}}{\bar{D}_0 \cdots \bar{D}_{N-1}} \right\}. \quad (60)$$

In OPENLOOPS 1, the calculation of the coefficients (59) is entirely based on the open-loop approach, but the reduction of the loop integrals (60) to scalar integrals, as well as the numerical evaluation of the latter, are performed by means of external libraries.

By default, OPENLOOPS 1 adopts the tensor representation on the rhs of (60) and computes the relevant tensor integrals with the COLLIER library [19], which implements the reduction techniques of [4,63] and the scalar integrals of [64]. One of the powerful features of COLLIER lies in sophisticated analytic expansions [4] that avoid dangerous numerical instabilities in phase space regions with small Gram determinants.

Alternatively, the reduction to scalar integrals is performed with CUTTOOLS [10], and scalar integrals are computed with ONELOOP [65]. The CUTTOOLS program implements the OPP reduction method [5], which is based on double, triple and quadruple cuts of the integrand on the lhs of (1). This requires a large number of evaluations of $\mathcal{V}(\Omega_N, q, 0)$, and the high efficiency of the open-loop representation, $\mathcal{V}(\Omega_N, q, 0) = \sum_{r=0}^{R} \mathcal{V}_{\mu_1 \cdots \mu_r}(\Omega_N, 0) \, q^{\mu_1} \cdots q^{\mu_r}$, results in a dramatic boost of the OPP method.

Another key feature behind the high speed of the open-loop method is the fact that, irrespectively of the reduction method, the time-consuming reduction to scalar integrals is performed after summing over colour and helicity degrees of freedom.

3 Summing helicities and diagrams on-the-fly

In this section we introduce a new technique that makes it possible to sum helicities and to merge different one-loop diagrams on-the-fly, i.e. after each step of the open-loop dressing recursion. Besides boosting the open-loop algorithm in a significant way, this approach is also a key aspect of the on-the-fly reduction technique introduced in the Sect. 4.

3.1 On-the-fly helicity summation

In the original formulation of the open-loop method, helicity sums (53) are performed at the end of the dressing recursion. This implies that the kth dressing step (39) needs to be performed for all helicity configurations of the dressed segments $\mathcal{S}_1, \ldots, \mathcal{S}_k$. This feature, combined with the fact that the number of relevant helicity states and the cost of a single dressing step scale exponentially with k, result in a very rapid growth of the CPU cost of dressing operations in the course of the open-loop recursion. To avoid this negative trend, in this section we introduce a method that exploits the factorisation properties of the open-loop representation (30) in a way that makes it possible to perform helicity sums *on-the-fly*, after the dressing of each new segment.

The idea, sketched in Fig. 3, is that, upon taking the interference of open loops with the Born amplitude, it is possible to sum over the helicities of all dressed segments, irrespec-

tively of the presence of still undressed segments. To introduce the technical aspects of this approach, let us rewrite the interference (53) between the colour-summed Born term $\mathcal{U}_0(\mathcal{I}_N, h)$ and the one-loop numerator as

$$
\begin{aligned}
\mathcal{U}(\mathcal{I}_N, q, 0) &= \sum_h \mathcal{U}_0(\mathcal{I}_N, h) \mathrm{Tr}\big[\mathcal{N}(\mathcal{I}_N, q, h)\big] \\
&= \sum_h \mathcal{U}_0(\mathcal{I}_N, h) \mathrm{Tr}\big[\mathcal{N}_k(\mathcal{I}_N, q, \hat{h}_k) \\
&\quad \times S_{k+1}(q, h_{k+1}) \cdots S_N(q, h_N)\big].
\end{aligned}
\tag{61}
$$

To take advantage of the factorisation of loop segments on the rhs of (61), we then postpone the trace operation and generalise (61) by defining

$$
\begin{aligned}
\mathcal{U}_k(\mathcal{I}_N, q, \check{h}_k) &= \sum_{\hat{h}_k} \mathcal{U}_0(\mathcal{I}_N, h) \mathcal{N}_k(\mathcal{I}_N, q, \hat{h}_k) \\
&= \sum_{h_1, \ldots, h_k} \mathcal{U}_0(\mathcal{I}_N, h) S_1(q, h_1) \cdots S_k(q, h_k),
\end{aligned}
\tag{62}
$$

where the interference with $\mathcal{U}_0(\mathcal{I}_N, h)$ is restricted to the first k dressed segments of the open loop, and the corresponding helicities, $\hat{h}_k = h_1 + \cdots + h_k$, are summed over. As a result, the first k segments become effectively unpolarised, and (62) depends only on the helicities of the remaining $N - k$ undressed segments, $\check{h}_k = h_{k+1} + \cdots + h_N$. Due to this dependence, which is induced by the fact that the Born term (51) depends on the helicities $h = \hat{h}_k + \check{h}_k$ of all external particles (37), the sums over $h_1, \ldots, h_k$ in (62) cannot be entirely factorised. However, they can be cast in the nested form

$$
\begin{aligned}
\mathcal{U}_k(\mathcal{I}_N, q, \check{h}_k) &= \sum_{h_k}\Bigg[\cdots \sum_{h_2}\bigg[\sum_{h_1} \mathcal{U}_0(\mathcal{I}_N, h) S_1(q, h_1)\bigg] \\
&\quad \times S_2(q, h_2) \cdots \Bigg] S_k(q, h_k),
\end{aligned}
\tag{63}
$$

which highlights the fact that each segment becomes effectively unpolarised after its dressing.

In practice, in analogy with the standard open-loop recursion (39), the helicity-summed open loops (62) are constructed with the recurrence relation

$$
\mathcal{U}_k(\mathcal{I}_N, q, \check{h}_k) = \sum_{h_k} \mathcal{U}_{k-1}(\mathcal{I}_N, q, \check{h}_{k-1}) S_k(q, h_k),
\tag{64}
$$

where the helicities $h_k \in \mathcal{H}_k$ of the dressed segment are summed on-the-fly. To this end, the dressing operation needs to be performed for all $\check{h}_{k-1} = \check{h}_k + h_k \in \check{\mathcal{H}}_{k-1}$. The initial condition reads

$$
\begin{aligned}
\mathcal{U}_0(\mathcal{I}_N, q, h) &= \mathcal{U}_0(\mathcal{I}_N, h) \cdot \mathbb{1} \\
&= 2 \sum_{\mathrm{col}} \mathcal{M}_0^*(h)\, \mathcal{C}(\mathcal{I}_N) \cdot \mathbb{1},
\end{aligned}
\tag{65}
$$

i.e. a fully undressed open loop is given by the interference of its colour structure with the Born amplitude, whose helicity states h live in the global helicity space $\mathcal{H}$. At each dressing step, helicity degrees of freedom are reduced by a factor equal to the number of helicity states of the dressed segment, i.e. by factor two for each external fermion or massless vector boson and a factor three for each external massive vector boson in the segment.

At the end of the recursion, when all N segments are dressed, no helicity dependence is left over ($\check{h}_N \equiv 0$), and the unpolarised loop numerator (53) is obtained by taking the trace

$$
\mathcal{U}(\mathcal{I}_N, q, 0) = \mathrm{Tr}\big[\mathcal{U}_N(\mathcal{I}_N, q, \check{h}_N)\big].
\tag{66}
$$

The recursion (64) is understood as matrix multiplication,

$$
\begin{aligned}
&\big[\mathcal{U}_k(\mathcal{I}_N, q, \check{h}_k)\big]_{\beta_0}^{\beta_k} \\
&= \sum_{h_k} \big[\mathcal{U}_{k-1}(\mathcal{I}_N, q, \check{h}_{k-1})\big]_{\beta_0}^{\beta_{k-1}} \big[S_k(q, h_k)\big]_{\beta_{k-1}}^{\beta_k},
\end{aligned}
\tag{67}
$$

in the tensor representation

$$
\mathcal{U}_k(\mathcal{I}_N, q, \check{h}_k) = \sum_{r=0}^R \mathcal{U}_{k;\, \mu_1 \cdots \mu_r}(\mathcal{I}_N, \check{h}_k)\, q^{\mu_1} \cdots q^{\mu_r},
\tag{68}
$$

which leads to the same tensorial recursion as in (42).

In summary, performing helicity sums on-the-fly leads to a decreasing number of helicity degrees of freedom when the number k of dressed segments increases. In this way, the effect of the growing CPU cost of dressing operations at large k can be strongly attenuated. The price to pay is that the parent–child approach (43)–(46) is not applicable anymore, due to the fact that (65) incorporates the colour structure $\mathcal{C}(\mathcal{I}_N)$ of the whole one-loop diagram. However, as we will see in Sect. 3.2, the parent–child relations can be replaced by a similarly efficient method based on the merging of topologically equivalent one-loop diagrams. Finally, let us note that the recursion (64)–(65) is not applicable to squared one-loop amplitudes. For this case we still rely on the original open-loop algorithm.

3.2 On-the-fly merging of topologically equivalent open loops

The key idea behind the recursion (64)–(65) is that, taking the interference between the Born amplitude and the one-loop colour structure $\mathcal{C}(\mathcal{I}_N)$ as initial condition makes it possible to anticipate operations that are usually performed after completion of the construction of a one-loop diagram. In particular, such operations become applicable *on-the-fly* after the dressing of individual loop segments. This technique will be denoted as on-the-fly approach, and its applicability goes well beyond helicity sums.
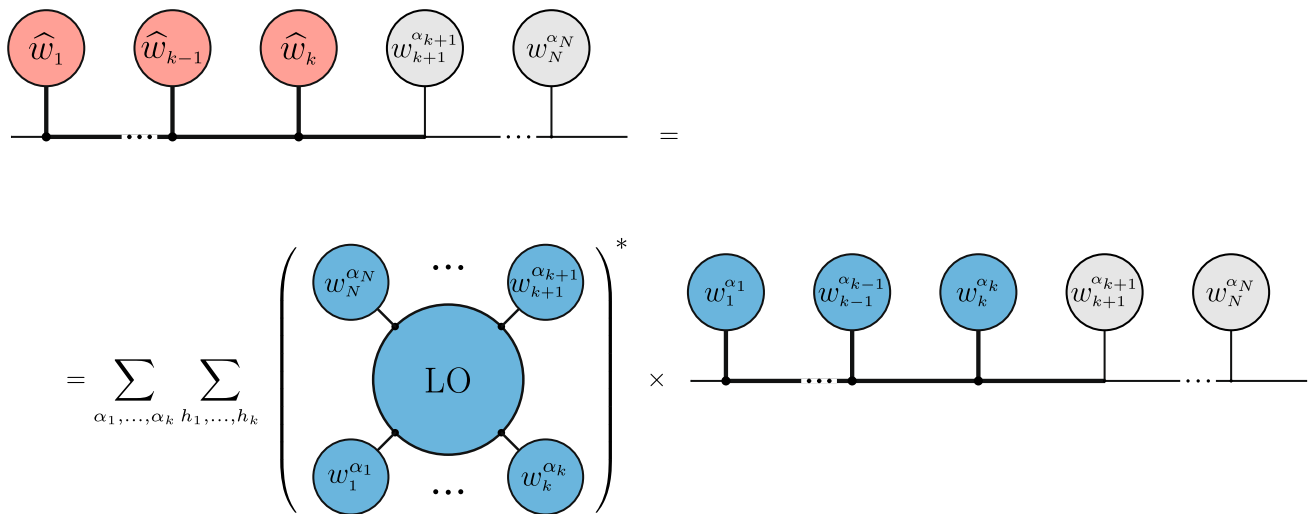
Fig. 4 Diagrammatic representation of helicity sums and diagram merging. Upon taking the interference with the Born amplitude, the helicities of the k dressed segments are summed, and the full set of topologically equivalent diagrams with the same undressed segments $\{\mathcal{S}_{k+1}^{\alpha_{k+1}}, \ldots, \mathcal{S}_N^{\alpha_N}\}$ is merged in a single open loop. See (70) and (72)

As sketched in Fig. 4, the on-the-fly technique can be extended to the double sums over helicity states and topologically equivalent loop diagrams in (59). The idea is that, rather than being constructed one by one, the topologically equivalent diagrams

$$\mathcal{I}_N^{\alpha_1\ldots\alpha_N} = \{\mathcal{S}_1^{\alpha_1}, \ldots, \mathcal{S}_N^{\alpha_N}\} \in \Omega_N, \tag{69}$$

defined in (55)–(58), can be merged in a recursive way by summing over the various subtrees $\mathcal{S}_i^{\alpha_i}$ as soon as they get dressed.

To this end, let us define subsets of diagrams, $\Omega_N^k \subset \Omega_N$, that share the same undressed segments, $\{\mathcal{S}_{k+1}^{\alpha_{k+1}}, \ldots, \mathcal{S}_N^{\alpha_N}\}$, after k dressing steps,

$$\Omega_N^k \equiv \Omega_N^k(\alpha_{k+1}, \ldots, \alpha_N)$$
$$= \left\{ \mathcal{I}_N^{\rho_1\ldots\rho_k\alpha_{k+1}\cdots\alpha_N} \,\middle|\, 1 \le \rho_j \le m_j \right.$$
$$\left. \text{with } \delta_{\rho_1\cdots\rho_k\alpha_{k+1}\cdots\alpha_N} \ne 0 \right\}, \tag{70}$$

where the tensor δ, defined in (57), filters out one-loop diagrams that are not allowed by the Feynman rules. By construction, all diagrams in the set (70) must undergo identical future dressing steps, which can be performed only once after merging the first k segments. This operation can be organised in a very similar way as helicity summations in Sect. 3.1. Technically, taking as a starting point the nested helicity sums in (63), it is sufficient to generalise the loop segments and the Born term (65) by replacing

$$S_i(q, h_i) \to S_i^{\alpha_i}(q, h_i),$$
$$\mathcal{U}_0(\mathcal{I}_N, h) \to \delta_{\alpha_1\cdots\alpha_N} \mathcal{U}_0(\mathcal{I}_N^{\alpha_1\ldots\alpha_N}, h), \tag{71}$$

and to extend the summation over the helicities $h_1, \ldots, h_k$ of the dressed segments to the corresponding "diagrammatic" degrees of freedom $\alpha_1, \ldots, \alpha_k$. This leads to the identity

$$\mathcal{V}_k^{\alpha_{k+1}\ldots\alpha_N}(\Omega_N^k, q, \check{h}_k)$$
$$= \sum_{\alpha_k, h_k} \left[\cdots \sum_{\alpha_2, h_2} \left[\sum_{\alpha_1, h_1} \delta_{\alpha_1\cdots\alpha_N} \mathcal{U}_0(\mathcal{I}_N^{\alpha_1\cdots\alpha_N}, h) \right. \right.$$
$$\left. \left. \times S_1^{\alpha_1}(q, h_1) \right] S_2^{\alpha_2}(q, h_2) \right] \cdots \right] S_k^{\alpha_k}(q, h_k), \tag{72}$$

which defines an open-loop object with fixed undressed segments $\{\mathcal{S}_{k+1}^{\alpha_{k+1}}, \ldots \mathcal{S}_N^{\alpha_N}\}$ and helicities $\check{h}_k = h_{k+1} + \cdots + h_N$ that incorporates all possible chains of dressed segments $\{\mathcal{S}_1^{\alpha_1}, \ldots, \mathcal{S}_k^{\alpha_k}\}$ forming a valid Feynman diagram, summed over the corresponding helicities $h_1, \ldots, h_k$.

Note that the dependence of (72) on the helicities $h_{k+1}, \ldots, h_N$ and diagrammatic indices $\alpha_{k+1}, \ldots, \alpha_N$ of the undressed segments is due to the fact that the Born term defined in (65) and (71) retains the full helicity dependence of the Born amplitude as well as the tensor (57) and the colour structure of the whole one-loop diagram.

In analogy with (39) and (64), the open-loop objects (72) can be constructed with the recurrence relation

$$\mathcal{V}_k^{\alpha_{k+1}\ldots\alpha_N}(\Omega_N^k, q, \check{h}_k)$$
$$= \sum_{\alpha_k} \sum_{h_k} \mathcal{V}_{k-1}^{\alpha_k\cdots\alpha_N}(\Omega_N^{k-1}, q, \check{h}_{k-1}) S_k^{\alpha_k}(q, h_k), \tag{73}$$

where helicity sums and diagram merging are performed on-the-fly. An explicit example of an on-the-fly merging step is illustrated in Fig. 5. Similarly as for (64), the recursion (73) is implemented in the form of tensorial relations (42). The
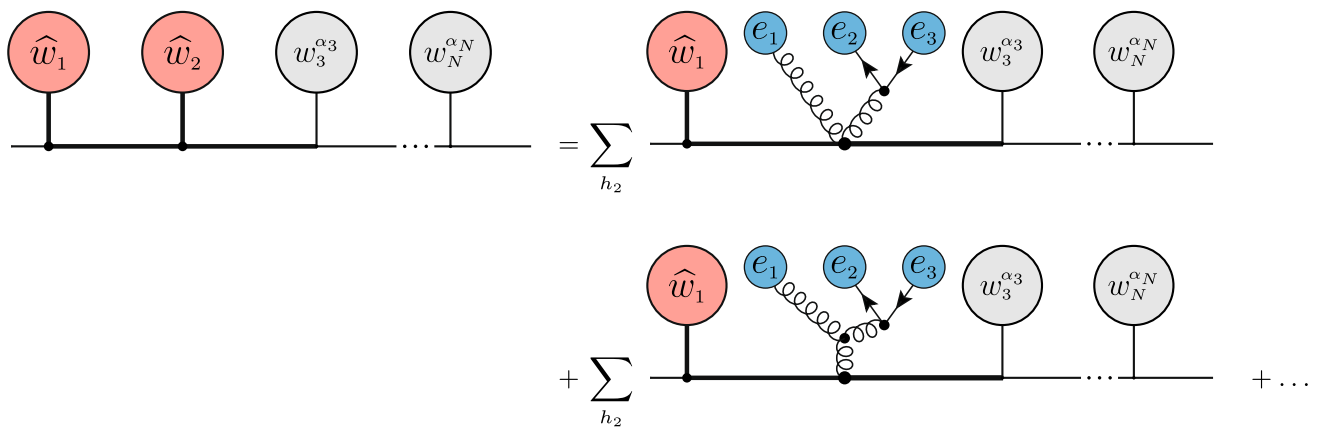
Fig. 5 Example of a step of the diagram merging recursion (73). After dressing the second segment and summing over its helicity configurations h_2, all diagrams with equivalent one-loop topology and identical undressed segments $\mathcal{S}_3^{\alpha_3}, \ldots, \mathcal{S}_N^{\alpha_N}$ are merged into a single open loop

relevant initial conditions at $k = 0$ are

$$
\begin{aligned}
\mathcal{V}_0^{\alpha_1 \cdots \alpha_N}(\Omega_N^0, h) &= \delta_{\alpha_1 \cdots \alpha_N} \, \mathcal{U}_0(\mathcal{I}_N^{\alpha_1 \cdots \alpha_N}, h) \cdot \mathbb{1} \\
&= 2 \, \delta_{\alpha_1 \cdots \alpha_N} \sum_{\mathrm{col}} \mathcal{M}_0(h)^* \mathcal{C}(\mathcal{I}^{\alpha_1 \cdots \alpha_N}) \cdot \mathbb{1},
\end{aligned}
\tag{74}
$$

where each fully undressed contribution corresponds to an individual diagram,

$$
\Omega_N^0 \equiv \Omega_N^0(\alpha_1, \ldots, \alpha_N) = \{ \mathcal{I}_N^{\alpha_1 \cdots \alpha_N} \},
\tag{75}
$$

with helicities h that live in the full helicity space $\mathcal{H}$. Let us point out that, thanks to the absorption of the colour factors $\mathcal{C}(\mathcal{I}_N^{\alpha_1 \cdots \alpha_N})$ in the Born interference term (74), in (73) it is possible to merge parts of diagrams that carry different colour structures in a single object, while respecting the exact colour dependence.

After N recursion steps one obtains a single open-loop object $\mathcal{V}_N(\Omega_N^N, q, \check{h}_N)$, which merges the full set of topologically equivalent diagrams ($\Omega_N^N \equiv \Omega_N$) and is entirely unpolarised ($\check{h}_N \equiv 0$). At this stage, taking the trace that closes the loop one arrives at

$$
\mathcal{V}(\Omega_N, q, 0) = \mathrm{Tr}\left[\mathcal{V}_N(\Omega_N^N, q, \check{h}_N) \right],
\tag{76}
$$

which is equivalent to (59).

As demonstrated in Sect. 6.2, performing helicity sums and merging diagrams on-the-fly yields a very significant efficiency improvement with respect to the original open-loop algorithm. More precisely, if helicity sums are performed at the end of the recursion as in (53), the merging approach and the parent–child relations (46) permit to achieve a similar speed-up factor of the order of two. However, contrary to the parent–child technique, the on-the-the-fly approach is applicable both to diagram merging and helicity sums. This leads to a further speed-up factor that can vary from two to three, depending on the process.

As we will see in Sect. 4, the on-the-fly approach will be a crucial ingredient in order to arrive at a new efficient algorithm that combines the operations of open-loop dressing and tensor reduction at the level of each individual step of the open-loop recursion.

4 On-the-fly reduction of open loops

In the original version of the OPENLOOPS program, the construction of integrand numerators and the reduction to scalar integrals are performed independently of one another using different tools. Open-loop numerators of N-point diagrams are constructed by recursively dressing N segments as described in Sect. 2.3. Each step of the recursion can increase the tensor rank by one, and, upon symmetrisation of all $q^{\mu_1} \cdots q^{\mu_r}$ monomials with $r \leq R$, open-loop polynomials of rank R involve $\binom{R+4}{4}$ independent tensor coefficients. Thus their complexity grows exponentially with the number of recursion steps. For instance, open loops with $R = 6$ and $R = 7$ involve, respectively, 210 and 330 components, while only 5 components are present for $R = 1$. As illustrated in the left plot of Fig. 6, in the original open-loop algorithm tensorial complexity keeps growing until the maximum rank $R \leq N$ is reached at the end of the dressing recursion. At this stage, upon summation of helicity configurations and loop diagrams with equivalent one-loop topology, tensor integrals are reduced to scalar integrals using external libraries, such as COLLIER [66] or CUTTOOLS [10], as described in Sect. 2.5.

Dealing with intermediate results with a large number of tensor components requires a considerable amount of computing power, both for the reduction of high-rank objects and at the level of the tensorial structure of the open-loop recursion (42), which needs to be performed for each relevant helicity configuration and each $[\ldots]_{\beta_0}^{\beta_k}$ component. These operations can be significantly accelerated by means of the

(a) A posteriori reduction   **(b)** On-the-fly reduction
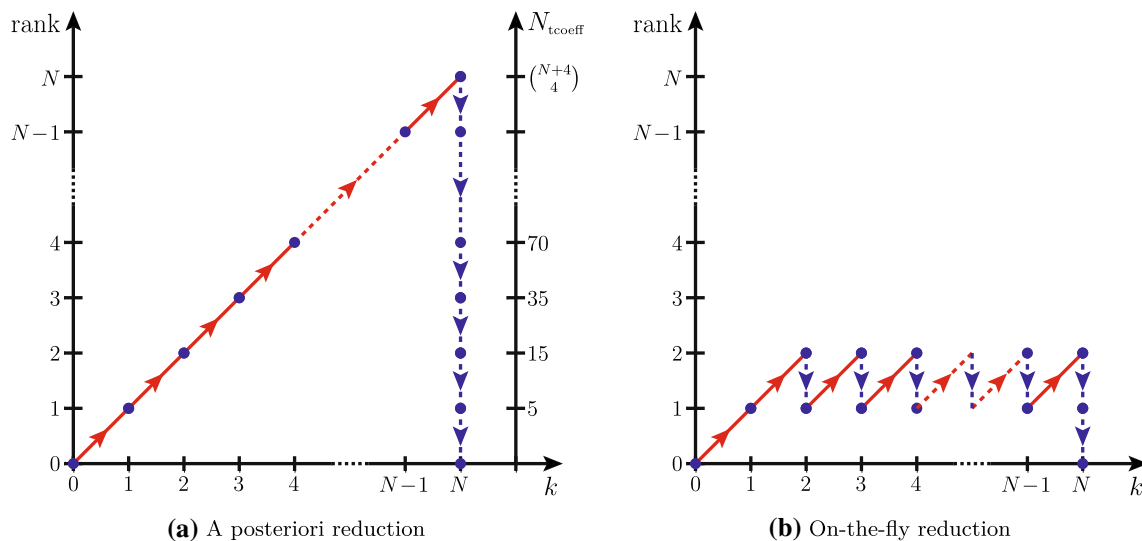
Fig. 6 Evolution of the tensor rank R and the number $N_{\text{tcoeff}}(R) = \binom{R+4}{4}$ of open-loop tensor coefficients (right vertical axis) as a function of the number k of dressed segments during the open-loop recursion. Each dressing step is assumed to increase the rank by one. The origi-nal open-loop algorithm, where tensor reduction is applied a posteriori (left), is compared to the on-the-fly reduction approach (right). The red diagonal lines illustrate the dressing steps and the blue vertical lines the reduction steps

techniques introduced in Sect. 3. Nevertheless, they remain the most CPU intensive aspect of multi-particle one-loop cal-culations in OPENLOOPS.

Motivated by the above considerations, in this section we introduce a new approach that avoids the appearance of high-rank objects at any stage of the calculation. This is achieved by extending the on-the-fly approach introduced in Sect. 3 to the reduction of open loops. In this way, interleaving the operations of open-loop dressing and tensor reduction, we build a single recursive algorithm, where each increase of tensorial rank caused by a dressing step is compensated by an integrand-reduction step.

As illustrated in the right plot of Fig. 6, the on-the-fly reduction approach avoids the appearance of any intermedi-ate object with rank higher than two. Besides the CPU cost needed for the processing of high-rank objects, this alleviates also possible memory issues due to their storage.

4.1 On-the-fly integrand reduction

For the on-the-fly reduction of open-loop polynomials we are going to use the method of [2], which permits to reduce rank-two monomials of the loop momentum through identities of the form

$$q^\mu q^\nu = \left[A_{-1}^{\mu\nu} + A_0^{\mu\nu} D_0\right] + \left[B_{-1,\lambda}^{\mu\nu} + \sum_{j=0}^{3} B_{j,\lambda}^{\mu\nu} D_j\right] q^\lambda. \tag{77}$$

The rank-one polynomial on the rhs is a linear combination of four loop denominators, $D_0, \ldots, D_3$, and the correspond-ing tensor coefficients, $A_j^{\mu\nu}$ and $B_{j,\lambda}^{\mu\nu}$, depend only on the three external momenta p_1, p_2, p_3. The coefficients of loop denominators are labeled with indices $j = 0, \ldots, 3$, while $j = -1$ is used for the constant parts. Their explicit expres-sions are presented in Sect. 5.2.

The identity (77) provides an exact reconstruction of $q^\mu q^\nu$ in terms of four-dimensional loop denominators, but can be easily generalised to D-dimensional denominators by replac-ing

$$D_j \rightarrow \bar{D}_j - \tilde{q}^2 \quad \text{for} \quad j = 0, 1, 2, 3. \tag{78}$$

Note that $\tilde{q}^2$ contributions resulting from the terms $B_{j,\lambda}^{\mu\nu} D_j$ with $j = 0, 1, 2, 3$ must cancel among each other in (77) since they generate rank-three terms of type $q^\lambda \tilde{q}^2$ that are not consistent with the rank-two structure on the lhs. Thus the substitutions (78) generate only an extra term $-\tilde{q}^2 A_0^{\mu\nu}$ on the rhs of (77).

The integrand reduction (77) holds at the integrand level, irrespectively of the presence of extra loop denominators $D_4, \ldots, D_{N-1}$ or additional q-dependent factors that may multiply the $q^\mu q^\nu$ monomial. These properties, in combi-nation with the factorisation of open loops into segments, make it possible to apply the reduction (77) at any interme-diate stage of the recursion (73). This on-the-fly reduction approach is illustrated in Fig. 7, and the corresponding reduc-tion identities for N-point integrands at step k of the dressing recursion have the form
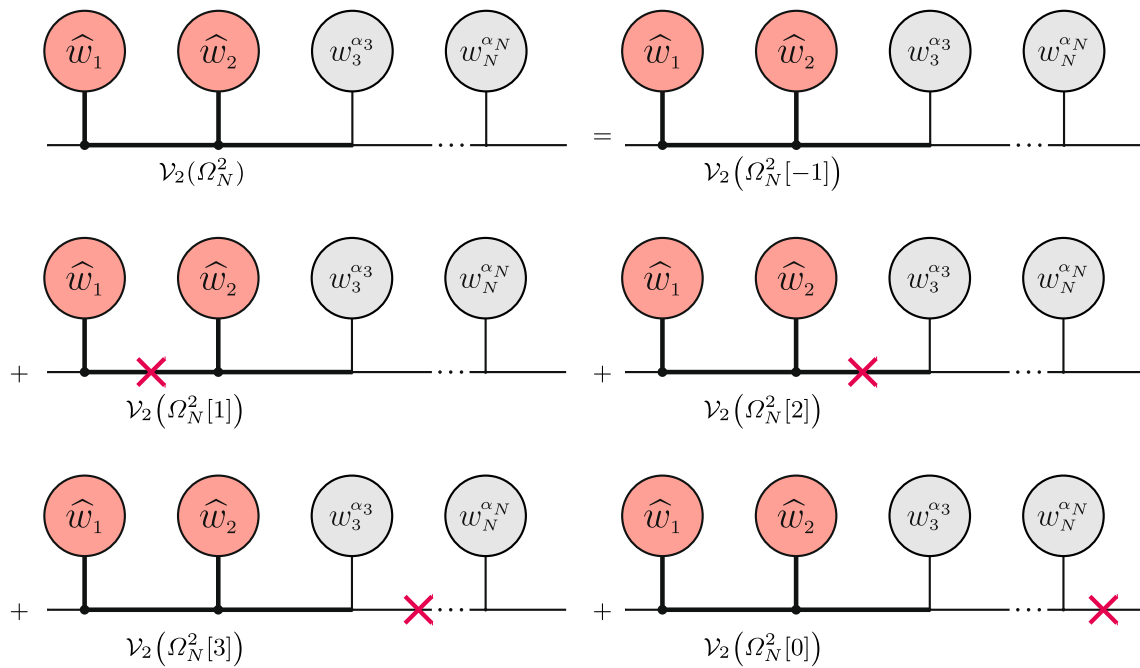
Fig. 7 Diagrammatic representation of the on-the-fly reduction step (79) for an N-point open loop at step $k = 2$ of the dressing recursion. The symbols $\mathcal{V}_k(\Omega_N^k)$ and $\mathcal{V}_k(\Omega_N^k[j])$ correspond, respectively, to the rank-two polynomial on the lhs of (79) and its reduced rank-one counterparts on the rhs. The red crosses indicate the pinching of the $\bar{D}_j$ propagators in the $\mathcal{V}_k(\Omega_N^k[j])$ terms with $j = 0, \ldots, 3$. Since $\bar{D}_0 = \bar{D}_N$, in our graphical representation the $\bar{D}_0$ denominator is located on the Nth segment

$$\frac{\mathcal{V}_k(\Omega_N^k, \bar{q}) \, S_{k+1}(q) \cdots S_N(q)}{\bar{D}_0 \cdots \bar{D}_3 \cdots \bar{D}_{N-1}}$$
$$= \sum_{j=-1}^{3} \frac{\mathcal{V}_k(\Omega_N^k[j], \bar{q}) \, S_{k+1}(q) \cdots S_N(q)}{\bar{D}_0 \cdots \bar{D}_j \cdots \bar{D}_3 \cdots \bar{D}_{N-1}}, \tag{79}$$

where $\Omega_N^k[j]$ for $0 \leq j \leq 3$ denote the $(N-1)$-point subtopologies that arise from Ω_N^k by pinching the $\bar{D}_j$ propagator, while terms with $j = -1$ on the rhs correspond to the original topology, $\Omega_N^k[-1] = \Omega_N^k$. Note that the denominators $\bar{D}_j$ can be pinched irrespectively of whether the related $S_j(q)$ segments are already dressed or not. In (79) we adopt the approach of Sect. 3.2, where open-loop polynomials incorporate the colour-summed interference with the Born amplitude as well as helicity sums and merging of all dressed segments. However, for simplicity, the bookkeeping of helicities, merged diagrams, and $[\ldots]_{\beta_0}^{\beta_k}$ indices is kept implicit.

The partially dressed open loops on the lhs and rhs of (79) have the general form

$$\mathcal{V}_k(\Omega, \bar{q}) = \sum_{s=0}^{S} \sum_{r=0}^{R} \mathcal{V}_{k;\mu_1 \cdots \mu_r}^s(\Omega) \, q^{\mu_1} \cdots q^{\mu_r} \, \tilde{q}^{2s}, \tag{80}$$

where four-dimensional loop-momentum components are accompanied by $\tilde{q}^2$ terms that arise from (77) to (78). As discussed in Sect. 4.4, only a small fraction of the $\tilde{q}^2$-dependent

terms can lead to non-vanishing contributions at the end of the recursion. Thus, in order to avoid the proliferation of tensor coefficients, all $\tilde{q}^2$ terms that are expected to vanish are identified and discarded in advance at each dressing and reduction step.

In general, the relation (79) allows one to reduce any polynomial $\mathcal{V}_k(\Omega_N^k, \bar{q})$ of rank $R \geq 2$ to rank $R-1$ polynomials $\mathcal{V}_k(\Omega_N^k[j], \bar{q})$. But, in practice, the reduction (79) can be interleaved with the open-loop dressing recursion in a way that the tensor rank never exceeds two. For $R = 2$ the coefficients of the rank-one open-loop polynomials that arise from the reduction (79) read

$$\mathcal{V}_{k;}^s(\Omega_N^k[j]) = \mathcal{V}_{k;\nu_1\nu_2}^s(\Omega_N^k) A_j^{\nu_1\nu_2} + \delta_{-1j} \Big[\mathcal{V}_{k;}^s(\Omega_N^k)$$
$$- \mathcal{V}_{k;\nu_1\nu_2}^{s-1}(\Omega_N^k) A_0^{\nu_1\nu_2} \Big],$$
$$\mathcal{V}_{k;\mu_1}^s(\Omega_N^k[j]) = \mathcal{V}_{k;\nu_1\nu_2}^s(\Omega_N^k) B_{j,\mu_1}^{\nu_1\nu_2} + \delta_{-1j} \mathcal{V}_{k;\mu_1}^s(\Omega_N^k). \tag{81}$$

The transformations (79)–(81) can be used to reduce any rank-two open loop with $N \geq 4$ propagators to a rank-one N-point object and four $(N-1)$-point pinched objects of rank one.

Rank-two open loops with only $N = 3$ loop propagators can be reduced to rank one in a very similar way [2]. The relevant identities (see Sect. 5.3) have the same structure as

(77) but involve only three reconstructed propagators, $\bar{D}_0$, $\bar{D}_1$ and $\bar{D}_2$. Moreover, they do not hold at the integrand level, but only upon integration over the loop momentum. The tensors $A_j^{\mu\nu}$ and $B_{j,\lambda}^{\mu\nu}$ for the case $N = 3$ depend only on p_1 and p_2. They are obtained from the ones for $N = 4$ by simply setting to zero the terms involving p_3 (see Sect. 5.3).

The on-the-fly reduction formula for $N = 3$ has the form

$$\int d^D \bar{q} \, \frac{\mathcal{V}_k(\Omega_3^k, \bar{q}) \, S_{\text{rem}}(q)}{\bar{D}_0 \cdots \bar{D}_2}$$
$$= \sum_{j=-1}^{2} \int d^D \bar{q} \, \frac{\mathcal{V}_k(\Omega_3^k[j], \bar{q}) \, S_{\text{rem}}(q)}{\bar{D}_0 \cdots \not{\bar{D}}_j \cdots \bar{D}_2}, \qquad (82)$$

where $\mathcal{V}_k(\Omega_3^k, \bar{q})$ is an open loop of rank $R = 2$ that results from a certain number $k \geq 2$ of dressing steps and $k-2$ or less reduction steps. Possible undressed segments are denoted as $S_{\text{rem}}(q)$, and (82) is valid only for terms $S_{\text{rem}}(q)$ of rank zero or one. In general this allows for only $N_{\text{rem}} \leq 1$ extra segments, i.e.

$$S_{\text{rem}}(q) = \begin{cases} S_{k+1}(q) & \text{for } N_{\text{rem}} = 1 \\ 1 & \text{for } N_{\text{rem}} = 0, \end{cases} \qquad (83)$$

and these two cases are sufficient to cover all relevant $N = 3$ topologies and pinched subtopologies in renormalisable theories.[3] The relations between the rank-two polynomial $\mathcal{V}_k(\Omega_3^k, \bar{q})$ and its reduced counterparts $\mathcal{V}_k(\Omega_3^k[j], \bar{q})$ have the same form as in (81).

Although the above three-point reduction holds only at the integral level, the fact that the term $S_{\text{rem}}(q)$ can be factorised makes it possible to apply (82) as soon as the dressed open loop $\mathcal{V}_k(\Omega_3^k, \bar{q})$ has reached rank two, independently of the remaining part of the numerator.

In summary, exploiting the fact that open loops are factorised into segments, the identities (77) and (82) can be applied on-the-fly during the dressing recursion, while an arbitrary number of segments is still undressed.[4] Thus, dressing and reduction steps can be interleaved in a way that the increase of tensorial rank resulting from dressing is promptly compensated through an on-the-fly reduction. More precisely, on-the-fly reduction steps are applied to diagrams and pinched sub-diagrams with $N \geq 3$ at those stages of the recursion where the next dressing step would generate

a rank-three object.[5] The reduced rank-one objects with N and $N - 1$ loop propagators are further dressed and possibly reduced until one arrives at fully dressed open loops of rank $R \leq 2$ for all two- and higher-point contributions. An this stage, the open-loop matrix structure can be eliminated by taking the trace (76), and all rank-two objects with $N \geq 3$ can be reduced to rank one with a final reduction step of type (77) or (82).

After the above dressing and on-the-fly reduction steps, the following types of integrals remain to be reduced:

(i)   integrals with $N \geq 5$ loop propagators
      and rank $R = 1, 0$;
(ii) integrals with $N = 4, 3$ loop propagators
      and rank $R = 1$;   (84)
(iii) integrals with $N = 2$ loop propagators
      and rank $R = 2, 1$.

For their reduction to scalar integrals with $N \leq 4$ we use a combination of integral reduction and OPP reduction identities as described in Appendix A.

4.2 Merging pinched topologies

Since it allows one to keep the tensor rank low at any stage of the calculations, the on-the-fly reduction approach has the potential to accelerate one-loop calculations in a significant way. However, some aspects of the on-the-fly reduction approach could lead to a dramatic increase of the computational cost. First, the fact that the reduction is performed when the loop is still open, i.e. before taking the trace (76), implies that the entries of the $[\ldots]_{\beta_0}^{\beta_k}$ matrix have to be processed as 16 independent objects. Second, the reduction has to be performed for all not yet summed helicity configurations of the undressed segments. Third, each reduction step (79) generates four pinched topologies that need to be processed as independent contributions in subsequent dressing and reduction steps.

Due to the proliferation of pinched subtopologies, the naive iteration of on-the-fly reduction steps would lead to a dramatic increase of the CPU cost. Fortunately, this can be avoided by means of the merging technique introduced in Sect. 3.2, which makes it possible to absorb pinched N-point open loops into unpinched $(N - 1)$-point open loops, in such a way that they do not need to be processed as separate objects. As explained in the following, the merging of pinched subtopologies requires a different implementation depending on whether the pinch belongs to the dressed part

---

[3] The fact that $N_{\text{rem}} \leq 1$ follows from the inequality $N_{\text{rem}} + R - N_{\text{unpinched}} \leq 0$, where $R = 2$ is the rank of $\mathcal{V}_k(\Omega_3^k, \bar{q})$, and $N_{\text{unpinched}} = 3$ is the number of unpinched propagators in (82). The above inequality is obviously fulfilled at the beginning of the open-loop recursion, where $k = R = N_{\text{rem}} = N_{\text{unpinched}} = 0$, and its validity is guaranteed by the fact that, in renormalisable theories, dressing and reduction steps cannot increase $N_{\text{rem}} + R - N_{\text{unpinched}}$.

[4] Note that the fact that the technique of [2] can be applied on-the-fly is not a general feature of integrand reduction methods. For instance, the OPP method [5] is not applicable on-the-fly since it is based on the multiple poles of the entire integrand.

[5] This means that, before performing an on-the-fly reduction, rank-two open loops are first dressed with all possible adjacent segments of rank zero. Delaying the reduction step in this way reduces the additional CPU cost that results from the appearance of new pinched objects.

of the open loop or not, as well as for the special case of a $\bar{D}_0$ pinch.

4.2.1 Pinching a dressed propagator

Let us consider the on-the-fly reduction of an N-point open loop with k dressed segments, focusing on contributions that result from the pinching of a $\bar{D}_j$ denominator with $j < k$,

$$
= \frac{\mathcal{V}_k^{\alpha_{k+1}\cdots\alpha_N}(\Omega_N^k[j],\bar{q})\, S_{k+1}^{\alpha_{k+1}}(q)\cdots S_N^{\alpha_N}(q)}{\bar{D}_0\cdots\bar{\slashed{D}}_j\cdots\bar{D}_3\cdots\bar{D}_{N-1}}.
$$
(85)

Here, consistently with the notation of Sect. 3.2, we have restored the indices α_i of the various undressed segments, while helicity indices are kept implicit. The pinched propagator in (85) is entirely dressed and merged, in the sense that both adjacent segments, S_j and S_{j+1}, are dressed and merged. Thus, for what concerns all future dressing and reduction steps, apart from the disappearance of the $\bar{D}_j$ denominator, the pinch has no effect. Consequently, the above contribution can be *absorbed* into unpinched $(N-1)$-point open loops that involve the same loop denominators, $\bar{D}_0\cdots\bar{D}_{j-1}\bar{D}_{j+1}\cdots\bar{D}_{N-1}$, and the same undressed segments, $S_{k+1}^{\alpha_{k+1}},\ldots,S_{N-1}^{\alpha_{N-1}}$. To this end, it is sufficient to bring the pinched open loop (85) in the standard form

which corresponds to an unpinched $(N-1)$-point open loop with $k-1$ dressed segments. The crossed vertex on the lhs of (86) indicates that the two original segments that are connected by the pinched propagator should be regarded as a single effective segment. This symbolic contraction of segments does not change anything in the numerics of the open-loop numerator, and the transformation (86) is nothing but a trivial relabeling of the denominators and of the undressed segments that lie on the right side of the pinch,

$$
\begin{aligned}
&\widetilde{D}_i(q) = \bar{D}_i(q) && \text{for } i < j, \\
&\widetilde{D}_i(q) = \bar{D}_{i+1}(q) \text{ and } \widetilde{S}_i^{\alpha_i}(q) = S_{i+1}^{\alpha_{i+1}}(q) && \text{for } i \geq j.
\end{aligned}
$$
(87)

The pinched open loop (86) can be merged with corresponding unpinched open loops to form a single $(N-1)$-point object with $k-1$ dressed segments. The corresponding formula reads

$$
\begin{aligned}
&\frac{\widetilde{\mathcal{V}}_{k-1}^{\alpha_k\cdots\alpha_{N-1}}(\widetilde{\Omega}_{N-1}^{k-1},\bar{q})\,\widetilde{S}_k^{\alpha_k}(q)\cdots\widetilde{S}_{N-1}^{\alpha_{N-1}}(q)}{\widetilde{D}_0\cdots\widetilde{D}_3\cdots\widetilde{D}_{N-2}} \\
&= \frac{\mathcal{V}_{k-1}^{\alpha_k\cdots\alpha_{N-1}}(\Omega_{N-1}^{k-1},\bar{q})\,\widetilde{S}_k^{\alpha_k}(q)\cdots\widetilde{S}_{N-1}^{\alpha_{N-1}}(q)}{\widetilde{D}_0\cdots\widetilde{D}_3\cdots\widetilde{D}_{N-2}} \\
&\quad + \sum_{\Omega_N^k[j]}\frac{\widetilde{\mathcal{V}}_k^{\alpha_k\cdots\alpha_{N-1}}(\Omega_N^k[j],\bar{q})\,S_{k+1}^{\alpha_k}(q)\cdots S_N^{\alpha_{N-1}}(q)}{\bar{D}_0\cdots\bar{\slashed{D}}_j\cdots\bar{D}_3\cdots\bar{D}_{N-1}}.
\end{aligned}
$$
(88)

A diagrammatic representation of this identity is given in Fig. 8. The resulting object, for which we introduce the symbol $\widetilde{\mathcal{V}}_{k-1}^{\alpha_k\cdots\alpha_{N-1}}$, is a combination of unpinched and pinched open loops, which enter, respectively, through the first and second term on the rhs of (88). By construction, the corresponding set of diagrams ($\widetilde{\Omega}_{N-1}^{k-1}$) includes all unpinched (Ω_{N-1}^{k-1}) and pinched ($\Omega_N^k[j]$) diagrams with loop propagators $\widetilde{D}_0\cdots\widetilde{D}_{N-2}$ and undressed segments $S_{k+1}^{\alpha_{k+1}},\ldots,S_{N-1}^{\alpha_{N-1}}$. The set of pinched diagrams $\Omega_N^k[j]$ corresponds to an N-point topology that results from Ω_{k-1}^{N-1} by undoing a $\bar{D}_j$ pinch, and (88) involves all possible $\Omega_N^k[j]$ contributions with $1 \leq j \leq k$.

As a necessary condition for the merging operation (88) to be applicable, the different open loops on the rhs of (88) have to feature the same undressed segments. This implies that they must be at the same stage of the dressing recursion, and, most importantly, that the starting position and the directions of the respective dressing recursions should be equivalent to each other. With other words, $\widetilde{D}_0\cdots\widetilde{D}_{N-2}$ and $\bar{D}_0\cdots\bar{\slashed{D}}_j\cdots\bar{D}_{N-1}$ should be two identical *ordered* sets of propagators. In particular they should start from the same cut propagator, $\widetilde{D}_0 = \bar{D}_0$. As discussed in Sect. 4.3 this can be guaranteed, to some extent, by means of an appropriate cutting rule.

Another obvious prerequisite for the absorption of pinched N-point open loops is the existence of corresponding unpinched $(N-1)$-point Feynman diagrams. With other words, the crossed vertex in (86) should have a physical counterpart consisting of a triple or quartic vertex, which can directly connect the $\bar{D}_{j-1}$ and $\bar{D}_{j+1}$ propagators to subtrees involving the external legs attached to $\hat{w}_j$ and $\hat{w}_{j+1}$ (see Fig. 8). In QCD, this turns out to be the case for most pinched configurations.

Moreover, pinched N-point configurations of the form (86) can also be merged with other pinched higher-point diagrams that get the relevant pinches in past or future reduction steps. Thus, pinched objects of type (86) will always be denoted as
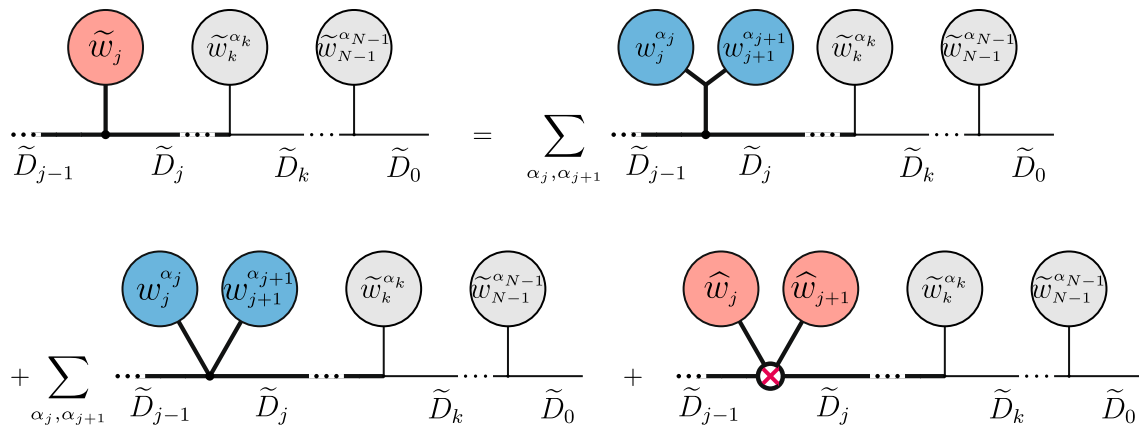
Fig. 8 Diagrammatic representation of formula (88) for the merging of pinched open loops. The first two unpinched diagrams on the rhs, where the generic subtrees $w_j^{\alpha_j}$ and $w_{j+1}^{\alpha_{j+1}}$ are directly connected to the loop propagators $\widetilde{D}_{j-1}$ and $\widetilde{D}_j$, correspond to the first term on the rhs of (88). The corresponding triple or quartic vertices of the Feynman

rules play an analogous role as the crossed vertex in the pinched open loop (last diagram on the rhs). Besides merging all relevant unpinched combinations α_j, α_{j+1}, also possible unpinched topologies where some of the external legs in w_j and w_{j+1} are interchanged (not shown) should be included

absorbable, irrespective of whether corresponding unpinched $(N-1)$-point Feynman diagrams exist or not.

The merging procedure (88) needs to be applied after any on-the-fly reduction step that creates new pinched objects. Thus merging operations have to be interleaved with iterated dressing and reduction steps. Since pinched N-point objects are absorbed into lower-point objects, the algorithm should start with the dressing and on-the-fly reduction of open loops with $N = N_{max}$, and continue towards lower N. The merging operation (88) starts at stage $N - 1 = N_{max} - 1$ and is applied after every dressing step, together with the merging of unpinched open loops (see Sect. 3.2). Note that, due to the iterative nature of the algorithm, the term $\widetilde{\mathcal{V}}_k^{\alpha_k \ldots \alpha_{N-1}}$ on the rhs of (88) can be the result of multiple pinching and merging steps.

4.2.2 Pinching an undressed propagator

The possibility to absorb pinched open loops as in Fig. 8 is based on the fact that all future dressing and reduction operations can be performed only once at the level of a merged object. Thus, the undressed segments of pinched and unpinched open loops should be identical.

However, the segments connected to the $\widetilde{D}_{j-1}$ and $\widetilde{D}_j$ propagators are *different* for pinched and unpinched terms. In the pinched case there are two separate segments, which involve w_j and w_{j+1} and require two subsequent dressing steps. In contrast, unpinched open loops require a single dressing step, since w_j and w_{j+1} are combined in a single segment.

It is thus clear that pinched open loops can be absorbed only after the dressing of the segments that lie on the two sides of the pinch. If this is not the case, i.e. when a $\bar{D}_j$ pinch

is applied to an open loop with $k \leq j$ dressed segments, its absorption becomes possible only after $(k - j + 1)$ additional dressing steps, which result in

$$
\begin{aligned}
&\widetilde{\mathcal{V}}_{j+1}^{\alpha_{j+1} \ldots \alpha_{N-1}} (\Omega_N^{j+1}[j], \bar{q}) \\
&= \sum_{\alpha'_{k+1} \ldots \alpha'_{j+1}} \widetilde{\mathcal{V}}_k^{\alpha'_{k+1} \ldots \alpha'_{j+1} \alpha_{j+1} \ldots \alpha_{N-1}} (\Omega_N^k[j], \bar{q}) \\
&= \sum_{\alpha'_{k+1} \ldots \alpha'_{j+1}} \times S_{k+1}^{\alpha'_{k+1}}(q) \cdots S_{j+1}^{\alpha'_{j+1}}(q),
\end{aligned}
\tag{89}
$$

where $S_{k+1} \cdots S_{j+1}$ are dressed. Note that in (89) we also sum over all possible α'_i, and use indices $\alpha_{j+1}, \ldots, \alpha_{N-1}$ with shifted labels for the undressed segments $S_{j+2}, \ldots, S_N$ on the right side of the pinch.

As illustrated in Fig. 9, the dressing operation (89) combined with the relabeling (87) brings the pinched open loops in a configuration that can be absorbed with the merging formula (88). However, the absorption of undressed pinched propagators is more involved than the simplified picture of Fig. 9. Pinched open loops can require more than one dressing step to become absorbable, in which case, in general, also new reduction steps are needed in order to keep the tensor rank below three. Such new reductions generate additional pinches, and their iteration can lead to a proliferation of multi-pinched configurations.

Thus, to avoid dramatic inefficiencies, it is crucial to minimise the number of required dressing steps by keeping pinchable propagators as close as possible to the dressed part of the numerator. This is why we choose to perform the on-the-fly reduction using the denominators $\bar{D}_0, \ldots, \bar{D}_3$.[6]

---

[6] The on-the-fly reduction (79) can be performed using any set of four propagators, $\bar{D}_{i_0}, \ldots, \bar{D}_{i_3}$.
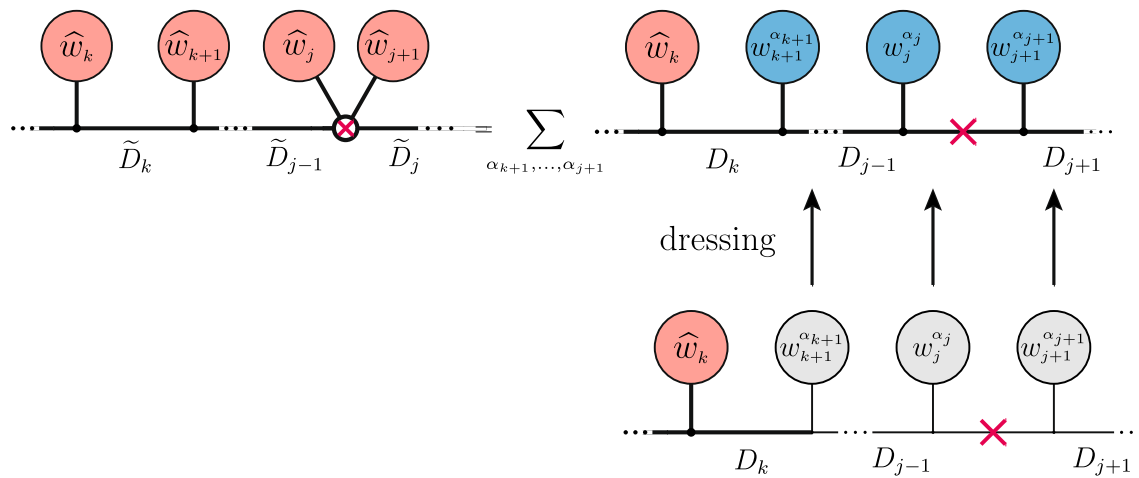
Fig. 9 Required dressing (89) and relabeling (87) operations before merging open loops with an undressed pinched propagator $\bar{D}_j$

In this way, when the first two segments are dressed and the first reduction step is applied (see Fig. 7), the various pinched propagators are located at most one left-dressing step ($\bar{D}_0$) and two right-dressing steps ($\bar{D}_3$) away from the dressed part.

In order to identify pinches that cannot be directly absorbed and to anticipate how they propagate through the recursion, let us consider generic open-loop configurations before the creation of a new pinch through a reduction step. At this stage the rank R must be equal to two. We first consider the very first reduction step, which can occur after $k \geq R = 2$ dressing steps, and we focus on a $\bar{D}_2$ pinch in the case where only $k = 2$ segments are dressed. In this case, an on-the-fly reduction step and a subsequent dressing step yield


$$(90)$$
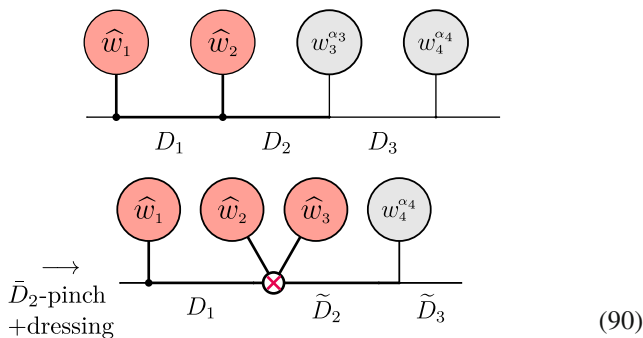
i.e. the $\bar{D}_2$ pinch can be brought in the standard form (86) and can thus be absorbed into unpinched contributions. The same considerations apply also to $\bar{D}_1$ and $\bar{D}_0$ pinches (see Sect. 4.2.3) and can be generalised to any step of the recursion, since the structure of the dressed parts on the lhs and rhs of (90) is the same.

Also $\bar{D}_3$ pinches can be absorbed in a similar way in case there are at least three dressed segments before the reduction step. Otherwise, when only two segments are dressed, the combination of an on-the-fly reduction step with a subsequent

dressing step leads to

$$(91)$$
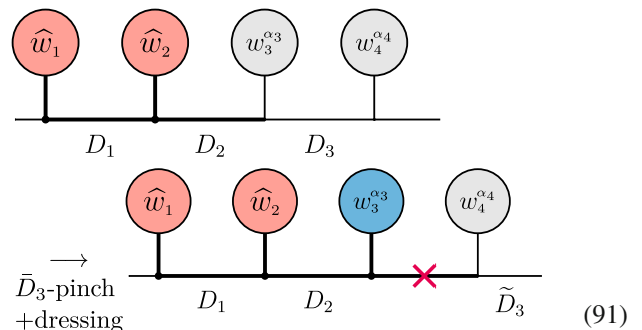
where the pinch is applied on the last dressed segment. Unless another dressing step can be applied before reaching rank three, this kind of pinch cannot be absorbed without a further reduction and dressing step. Dressing one more segment allows one to absorb the original D_3 pinch as well as new $\bar{D}_0$, $\bar{D}_1$, and $\bar{D}_2$ pinches that arise from the new reduction step. However, the new reduction leads again to a configuration with a $\widetilde{D}_3$ pinch on the last dressed segment,

$$(92)$$
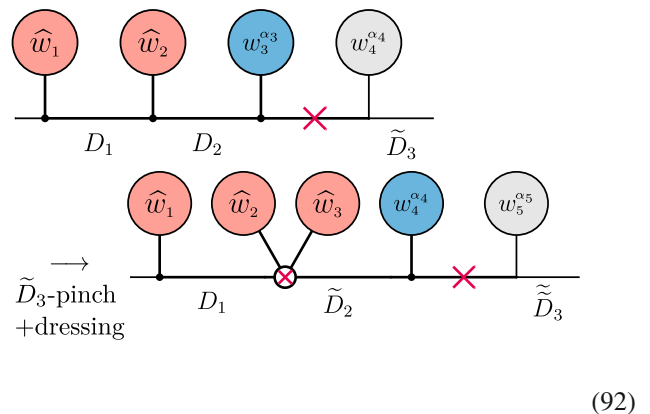
Again, the dressed parts on the lhs and rhs of (92) have the same structure, which implies that such $\bar{D}_3$-pinched config-
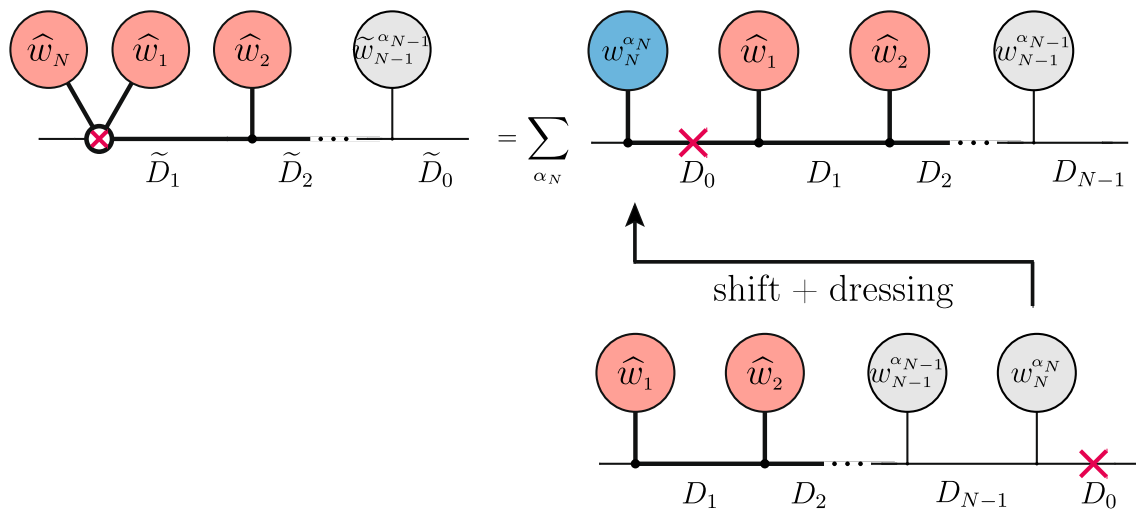
Fig. 10 Required dressing, loop-momentum shift and relabeling operations (94)–(96) before merging an open loop with a pinched $\bar{D}_0$ propagator

urations are stable with respect to further reduction steps. Thus, open loops with multiple non-absorbable pinches do not occur, and the only type of configuration with a single non-absorbable pinch is the one in (92).

4.2.3 Pinching the $\bar{D}_0$ propagator

Finally, we consider open loops with k dressed segments and a pinched $\bar{D}_0$ denominator,

$$
= \frac{\mathcal{V}_k^{\alpha_{k+1}\cdots\alpha_N}(\Omega_N^k[0],\bar{q})\, S_{k+1}^{\alpha_{k+1}}(q)\cdots S_N^{\alpha_N}(q)}{\bar{D}_1\cdots\bar{D}_3\cdots\bar{D}_{N-1}\bar{\mathcal{D}}_0}.
\tag{93}
$$

For convenience, here and in the following we put $\bar{D}_0$ at the end of the chain of denominators. Similarly as for the cases discussed in Sect. 4.2.2, the above pinched configuration becomes absorbable only when the segments connected to the $\bar{D}_0$ propagator, i.e. $S_N(q)$ and $S_1(q)$, are dressed. However, this happens only at the end of the standard dressing recursion.

In order to anticipate the absorption of the most advanced pinch one could replace the denominators $\bar{D}_0, \ldots, \bar{D}_3$ used for the on-the-fly reduction by $\bar{D}_1, \ldots, \bar{D}_4$, which lie all directly on the right side of the cut. However, the absorption of each $\bar{D}_4$ pinch would require up to three extra dressing steps and two related reduction steps, resulting in the creation of multiple new pinches. This problem can be circumvented by observing that the $\bar{D}_0$ propagator lies only one step away

from the dressed part of the open loop, if one reverts the dressing direction. Therefore, as illustrated in Fig. 10, the pinched $\bar{D}_0$ propagator can be entirely dressed by means of a single left-dressing step. This operation results in

$$
\widetilde{\mathcal{V}}_{k+1}^{\alpha_{k+1}\cdots\alpha_{N-1}}(\Omega_N^{k+1}[0],\bar{q})
$$
$$
= \sum_{\alpha_N} S_N^{\alpha_N}(q)\, \widetilde{\mathcal{V}}_k^{\alpha_{k+1}\cdots\alpha_{N-1}\alpha_N}(\Omega_N^k[0],\bar{q}),
\tag{94}
$$

where the left multiplication of the Nth segment should be understood as

$$
\left[S_N(q)\,\widetilde{\mathcal{V}}_k(\Omega,\bar{q}) \right]_{\beta_0}^{\beta_{k+1}} = \sum_{\beta_N} \left[S_N(q) \right]_{\beta_0}^{\beta_N} \times \left[\widetilde{\mathcal{V}}_k(\Omega,\bar{q}) \right]_{\beta_N}^{\beta_{k+1}}
$$
$$
= \left[\left(\left[\widetilde{\mathcal{V}}_k(\Omega,\bar{q}) \right]^{\text{T}} \left[S_N(q) \right]^{\text{T}} \right)^{\text{T}} \right]_{\beta_0}^{\beta_{k+1}}.
\tag{95}
$$

Technically, as indicated on the rhs, this operation can be easily implemented through a standard right-dressing step upon transposition of the input matrices and back-transposition of the result.[7]

As usual, before merging with unpinched open loops, the propagators and undressed segments that lie on the right side of the pinch need to be brought back in standard form. In case of a $\bar{D}_0$ pinch, all remaining $N-1$ denominators and segments preserve their relative position along the open loop. Thus, only $\bar{D}_{N-1}$ needs to be relabeled, since it assumes the role of the new $\widetilde{D}_0$. Moreover, the standard form $\widetilde{D}_0 = q^2 - \widetilde{m}_0^2$ requires a loop momentum shift $q \to q - p_{N-1}$ for

---

[7] Note that in a three-gluon vertex this transposition leads to a flip of the colour indices of the two gluons in the loop and hence a minus sign in the colour factor. Since the colour factor has been fixed in the beginning this minus sign has to be taken into account in the colour-stripped open loop.

the entire open loop. The corresponding reparametrisations for the various denominators and segments read

$$
\begin{aligned}
\widetilde{D}_0(q) &= \bar{D}_{N-1}(q - p_{N-1}), \\
\widetilde{D}_i(q) &= \bar{D}_i(q - p_{N-1}) \quad \text{for } 1 \le i \le N-2, \\
\widetilde{S}_i^{\alpha_i}(q) &= S_i^{\alpha_i}(q - p_{N-1}) \quad \text{for } k+1 \le i \le N-1.
\end{aligned}
\tag{96}
$$

In terms of masses and momenta this corresponds to $\widetilde{m}_0 = m_{N-1}, \widetilde{m}_i = m_i$ and $\widetilde{p}_i = p_i + p_{N-1}$ for $1 \le i \le N-2$. With these transformations the left-dressed $\bar{D}_0$-pinched open loop (94) can be merged with related unpinched objects according to

$$
\begin{aligned}
&\frac{\widetilde{\mathcal{V}}_k^{\alpha_{k+1}\dots\alpha_{N-1}}(\widetilde{\Omega}_{N-1}^k, \bar{q}) \widetilde{S}_{k+1}^{\alpha_{k+1}}(q) \cdots \widetilde{S}_{N-1}^{\alpha_{N-1}}(q)}{\bar{D}_1 \cdots \widetilde{D}_3 \cdots \widetilde{D}_{N-2} \widetilde{D}_0} \\
&\equiv \frac{\mathcal{V}_k^{\alpha_{k+1}\dots\alpha_{N-1}}(\Omega_{N-1}^k, \bar{q}) \widetilde{S}_{k+1}^{\alpha_{k+1}}(q) \cdots \widetilde{S}_{N-1}^{\alpha_{N-1}}(q)}{\bar{D}_1 \cdots \widetilde{D}_3 \cdots \widetilde{D}_{N-2} \widetilde{D}_0} \\
&\quad + \sum_{\Omega_N^k[0]} \frac{\widetilde{\mathcal{V}}_{k+1}^{\alpha_{k+1}\dots\alpha_{N-1}}(\Omega_N^{k+1}[0], \bar{q})}{\bar{D}_1 \cdots \bar{D}_3 \cdots \bar{D}_{N-1} \bar{D}_0} \\
&\quad \times S_{k+1}^{\alpha_{k+1}}(q) \cdots S_{N-1}^{\alpha_{N-1}}(q) \Bigg|_{q \to q - p_{N-1}}.
\end{aligned}
\tag{97}
$$

Apart from the loop momentum shift, $q \to q - p_{N-1}$, this formula is entirely analogous to (88). Let us note that, since the shift is applied to a single term, the identity (97) holds only upon loop momentum integration. Nevertheless, as far as the correctness of final results at integral level is concerned, it can be safely applied at the integrand level.

As demonstrated in Sect. 6.2, using the on-the fly reduction with pinch absorption in combination with the on-the-fly techniques of Sect. 3 results in a very fast and numerically stable one-loop algorithm. In particular, as compared to the original version of OPENLOOPS, we find very significant improvements, both in terms of speed and numerical stability. Actually, using only the new techniques of Sect. 3 without on-the-fly reduction yields even higher CPU efficiency. However, as we will see, the moderate extra CPU cost that results from the on-the-fly reduction approach is counterbalanced by a very significant gain in numerical stability, which implies a reduced usage of quadruple precision for exceptional phase space points.

4.3 Cutting rule

As pointed out in Sect. 4.2.1, the possibility to merge pinched N-point open loops and corresponding unpinched $(N-1)$-point open loops depends on the way they are cut. In order to identify the relevant requirements, let us consider the cut-open topology defined by the following ordered set of loop segments,

$$
\mathcal{I}_N = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_N\}.
\tag{98}
$$

Here we have applied our standard labeling scheme, where the cut is located between $\mathcal{S}_N$ and $\mathcal{S}_1$, i.e. on the $\bar{D}_0$ propagator, while the dressing recursion starts with $\mathcal{S}_1$ and is directed towards $\mathcal{S}_2$. This configuration will be referred to as a $\mathcal{S}_N/\mathcal{S}_1$ ordered cut. Since the labeling scheme is a consequence of the position of the cut, and not vice versa, we have to define a cutting rule that selects $\mathcal{S}_N/\mathcal{S}_1$ out of all possible $\mathcal{S}_i/\mathcal{S}_j$ cuts.

The cutting rule should enable the merging of pinched subtopologies that arise from (98) by pinching certain propagators $\bar{D}_j$, i.e. by combining $\mathcal{S}_j$ and $\mathcal{S}_{j+1}$ in a single segment $\mathcal{S}_j \oplus \mathcal{S}_{j+1}$. To this end, unless the cut propagator $\bar{D}_0$ is pinched, the cutting rule should guarantee that the position of the cut and its direction remain unchanged after a pinch. More explicitly, the desired cut configurations after a $\bar{D}_j$ pinch with $j > 0$ are

$$
\mathcal{I}_N[1] = \{\mathcal{S}_1 \oplus \mathcal{S}_2, \mathcal{S}_3, \dots, \mathcal{S}_N\},
\tag{99}
$$
$$
\mathcal{I}_N[j] = \{\mathcal{S}_1, \dots, \mathcal{S}_{j-1}, \mathcal{S}_j \oplus \mathcal{S}_{j+1}, \mathcal{S}_{j+2}, \dots, \mathcal{S}_N\}
$$
$$
\text{for } 2 \le j \le N-2,
\tag{100}
$$
$$
\mathcal{I}_N[N-1] = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_{N-1} \oplus \mathcal{S}_N\}.
\tag{101}
$$

In this way, as required for the merging operations described in Sects. 4.2.1–4.2.2, the dressing of pinched and unpinched objects always starts and ends with segments that contain the external legs attached to $\mathcal{S}_1$ and $\mathcal{S}_N$, respectively. In the case of a $\bar{D}_0$ pinch, where the original cut propagator disappears, in order to enable the merging of left-dressed pinched subtopologies described in Sect. 4.2.3, the cut should be moved to the left of $\mathcal{S}_N \oplus \mathcal{S}_1$, so that

$$
\mathcal{I}_N[0] = \{\mathcal{S}_N \oplus \mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_{N-1}\}.
\tag{102}
$$

In order to ensure, at least in part, that pinched topologies are cut as in (99)–(102), we replace the original cutting rule (49)–(50) by the new prescriptions

$$
F(\mathcal{S}_1) > F(\mathcal{S}_k) \ \forall \ k > 1, \qquad \text{(selection rule)} \tag{103}
$$
$$
F(\mathcal{S}_N) > F(\mathcal{S}_2), \qquad \text{(direction rule)} \tag{104}
$$

where the weights $F(\mathcal{S}_a)$ are defined in (47). The key property of the above cutting rule is the pinch-invariance of the selection rule, which determines the first segment $\mathcal{S}_1$. In fact, if this condition is realised for (98), then it is guaranteed to hold also for all pinched configurations (99)–(102). For $j = 0, 1$ this is an obvious consequence of the fact that $F(\mathcal{S}_1 \oplus \mathcal{S}_a) > F(\mathcal{S}_1)$ for any $a \ne 1$. In the other cases, the fact that $\mathcal{S}_1$ remains the first subtree in spite of the appearance of a new pinched subtree $\mathcal{S}_j \oplus \mathcal{S}_{j+1}$ with weight $F(\mathcal{S}_j) + F(\mathcal{S}_{j+1})$, is guaranteed by

$$
F(\mathcal{S}_1) > \sum_{i=2}^{N} F(\mathcal{S}_i).
\tag{105}
$$

This inequality is an automatic consequence of (103) and of the binary nature of the weights (47). This is easily understood by observing that, due to $2^{N_p-1} = 1 + \sum_{\alpha=1}^{N_p-1} 2^{\alpha-1}$, the last external particle ($\alpha = N_p$) outweighs the ensemble of all other particles. Therefore the last external particle must belong to the leading-weight subtree $\mathcal{S}_1$, which implies that $F(\mathcal{S}_1) \geq 2^{N_p-1}$ and leads to (105).

Unfortunately, the direction rule (104) is not sufficient in order to preserve the direction of the cut. For instance, in case of a $\bar{D}_1$ pinch, the desired cut (99) requires $F(\mathcal{S}_N) > F(\mathcal{S}_3)$, which does not automatically follow from $F(\mathcal{S}_N) > F(\mathcal{S}_2)$. More generally, apart from the case of a D_3 pinch, where the second and last subtree do not change, there is no guarantee that the condition (104) preserves the direction of the cut.

Thus the above cutting rule does not allow one to absorb all pinched open loops. Nevertheless, as demonstrated in Sect. 6.2, it is sufficient to obtain a very fast on-the-fly reduction algorithm. Moreover, it should be possible to further increase CPU efficiency, either by means of an improved cutting rule or by inverting the dressing direction after certain pinches.

4.4 Rational terms of type R_1

As discussed in Sect. 4.1, each reduction step of type (79) and (82) generates terms $\bar{D}_i - D_i = \tilde{q}^2$ that account for the mismatch between the D-dimensional and four-dimensional parts of loop denominators. The resulting tensor integrals with $(D-4)$-dimensional $\tilde{q}^2$ terms in the numerator can give rise to finite terms. As is well known [2,59], these so-called rational terms of type R_1 can arise only in the presence of $1/(D-4)$ poles of ultraviolet type. Thus, vanishing integrals of type R_1 can be easily identified by means of the simple power counting criterion

$$\int d^D \bar{q} \, \frac{q^{\mu_1} q^{\mu_2} \cdots q^{\mu_r} \tilde{q}^{2s}}{\bar{D}_0 \bar{D}_1 \cdots \bar{D}_{N-1}} = \mathcal{O}(D-4)$$
$$\text{if} \quad s \geq 1 \quad \text{and} \quad r + 2s + 4 < 2N. \tag{106}$$

In a renormalisable theory, where each loop segment increases the rank at most by one, $r + 2s \leq N$ and all integrals with $N \geq 5$ and $s \geq 1$ vanish. Thus, the only non-vanishing integrals of type R_1 that remain at the end of the on-the-fly reductions of Sect. 4.1 are [2]

$$\int d^D \bar{q} \, \frac{\tilde{q}^2}{\bar{D}_0 \bar{D}_1} = \frac{-i\pi^2}{2} \left(m_0^2 + m_1^2 - \frac{p_1^2}{3} \right)$$
$$+ \mathcal{O}(D-4), \tag{107}$$

$$\int d^D \bar{q} \, \frac{q^\mu \tilde{q}^2}{\bar{D}_0 \bar{D}_1 \bar{D}_2} = \frac{i\pi^2}{6} (p_1 + p_2)^\mu$$
$$+ \mathcal{O}(D-4), \tag{108}$$

$$\int d^D \bar{q} \, \frac{\tilde{q}^2}{\bar{D}_0 \bar{D}_1 \bar{D}_2} = \frac{-i\pi^2}{2} + \mathcal{O}(D-4), \tag{109}$$

$$\int d^D \bar{q} \, \frac{\tilde{q}^4}{\bar{D}_0 \bar{D}_1 \bar{D}_2 \bar{D}_3} = \frac{-i\pi^2}{6} + \mathcal{O}(D-4). \tag{110}$$

The power counting criterion (106) can be exploited in a way that makes it possible to discard irrelevant terms of type R_1 at any intermediate step of the open-loop recursion. To this end, for each number k of dressing steps we anticipate the maximum rank $R_k^{\max}$ of the segments $S_{k+1}(q), \ldots, S_N(q)$ that remain to be dressed. Given this information, it is clear that monomials of type $q^{\mu_1} q^{\mu_2} \ldots q^{\mu_r} \tilde{q}^{2s}$ in the dressed open loop cannot give rise to terms of D-dimensional rank higher than $r + 2s + R_k^{\max}$ at the end of the recursion. Thus, we can anticipate that,

$$\int d^D \bar{q} \, \frac{q^{\mu_1} \cdots q^{\mu_r} \tilde{q}^{2s} S_{k+1}(q) \cdots S_N(q)}{\bar{D}_0 \bar{D}_1 \cdots \bar{D}_{N-1}} = \mathcal{O}(D-4)$$
$$\text{if} \quad s \geq 1 \quad \text{and} \quad r + 2s + 4 + R_k^{\max} < 2N. \tag{111}$$

The systematic application of this condition allows one to filter out a very large number of $\tilde{q}^2$ terms, thereby improving the efficiency of the algorithm.

Note also that the unpinched contributions $\mathcal{V}_k(\Omega_N^k[-1])$ in the reduction identities (81) involve terms that reduce $r + 2s - 2N$, and thus the degree of ultraviolet divergence, by one and two. Depending on the values of N and $R_k^{\max}$, this can result in vanishing R_1 contributions that can also be immediately discarded. For instance, in the reduction of $\int d^D q \, \frac{q^{\mu_1} q^{\mu_2} \tilde{q}^2}{\bar{D}_0 \bar{D}_1 \bar{D}_2 \bar{D}_3}$ all unpinched contributions apart from those of type (110) can be neglected.

5 Reduction identities and numerical stability

This section deals with the reduction method of [2], which provides the basis of the on-the-fly reduction approach of Sect. 4.1. In Sects. 5.1–5.3 we outline the derivation of the tensor coefficients $A_j^{\mu\nu}$ and $B_{j,\lambda}^{\mu\nu}$ in the reduction identities (77) and (82). In doing so we set the stage for Sect. 5.4, where we discuss numerical instability problems and present a systematic approach for their solution.

5.1 The reduction basis

The reduction identities of [2] are based on a decomposition of the four-dimensional loop momentum,

$$q^\mu = \sum_{i=1}^{4} c_i l_i^\mu, \tag{112}$$

in a basis $l_1, \ldots, l_4$, formed by massless momenta in two orthogonal planes,

$$l_i^2 = 0, \quad l_{1,2} \cdot l_{3,4} = 0. \tag{113}$$

This *reduction basis* is defined in terms of the external momenta p_1, p_2, which enter the propagators D_1, D_2. The basis momenta $l_{1,2}$ are chosen in the plane spanned by p_1 and p_2,

$$l_1^\mu = p_1^\mu - \alpha_1 p_2^\mu, \quad l_2^\mu = p_2^\mu - \alpha_2 p_1^\mu, \tag{114}$$

while $l_{3,4}$ lie in the plane orthogonal to p_1, p_2 and are defined as

$$l_3^\mu = \bar{v}(l_1)\gamma^\mu \left(\frac{1-\gamma^5}{2}\right) u(l_2),$$

$$l_4^\mu = \bar{v}(l_2)\gamma^\mu \left(\frac{1-\gamma^5}{2}\right) u(l_1), \tag{115}$$

where u and $\bar{v}$ are massless spinors. This definition of $l_{3,4}$ implies $l_3^* = e^{i\chi} l_4$, where χ is twice the phase difference between the u and v spinors. The normalization of the basis is chosen such that

$$\gamma = 2(l_1 \cdot l_2) = -\frac{1}{2}(l_3 \cdot l_4), \tag{116}$$

and the $\alpha_{1,2}$ coefficients in (114) read[8]

$$\alpha_i = \frac{p_i^2}{(p_1 \cdot p_2) \pm \sqrt{\Delta}}, \tag{117}$$

where Δ is related to the rank-two Gram determinant $\Delta_{12} = \det(p_i \cdot p_j)$ via

$$\Delta = -\Delta_{12} = (p_1 \cdot p_2)^2 - p_1^2 p_2^2. \tag{118}$$

The Gram determinant is related to the normalisation factor γ via

$$\gamma = \frac{4\Delta}{(p_1 \cdot p_2) \pm \sqrt{\Delta}}, \tag{119}$$

and these two parameters play a critical role for the stability of the reduction. In fact, in the limit of vanishing Gram determinant, (116) and (119) imply that $(l_1 \cdot l_2) \propto (l_3 \cdot l_4) \propto \gamma \propto \Delta$. Thus

$$\lim_{\Delta \to 0} (l_i \cdot l_j) = 0 \quad \forall \, i, j, \tag{120}$$

which implies that all light-like basis momenta l_i become parallel to each other[9] leading to severe numerical instabilities in the decomposition (112).

Note that in [2] the basis momenta l_i contain an additional normalisation factor[10]

$$\beta = \frac{1}{1 - \alpha_1\alpha_2} = \pm \frac{(p_1 \cdot p_2) \pm \sqrt{\Delta}}{2\sqrt{\Delta}}, \tag{121}$$

which diverges like $1/\sqrt{\Delta}$ when $\Delta \to 0$. As a consequence, in [2] numerical instabilities are in part visible as factors β in the reduction formulas and in part hidden in the definition of the basis vectors. Instead, the basis momenta defined in (114)–(115) are stable in the $\Delta \to 0$ limit. Thus, in the reduction formulas presented in Sects. 5.2 and 5.3, instabilities related to the Gram determinant (118) are fully manifest in the form of inverse powers of the parameter $\gamma \propto \Delta$. More precisely, for $p_1^2 = 0$ and $p_2^2 \neq 0$, Gram-determinant instabilities arise also from $\alpha_2 = \pm p_2^2/(2\sqrt{\Delta})$. However, the parametrisation adopted in Sect. 5.3 ensures that α_2 is always regular.

5.2 On-the-fly box reduction

In the following we discuss the reduction identity (77), which can be rewritten in a slightly more compact form as

$$q^\mu q^\nu = \sum_{j=-1}^{3} \left[A_j^{\mu\nu} + B_{j,\lambda}^{\mu\nu} q^\lambda \right] D_j, \tag{122}$$

with $D_{-1} = 1$. Since $q^\mu q^\nu$ is reconstructed in terms of D_0, D_1, D_2, D_3, we denote (122) as box reduction identity, although it is applicable to any integrand with $N \geq 4$ loop propagators. The starting point for its derivation is given by the decomposition (112). Since the basis momenta $l_{1,2}$ and $l_{3,4}$ lie in mutually orthogonal planes, it is natural to split the loop momentum into corresponding components,

$$q^\mu = q_\parallel^\mu + q_\perp^\mu, \tag{123}$$

with $q_\parallel^\mu = c_1 l_1^\mu + c_2 l_2^\mu$ and $q_\perp^\mu = c_3 l_3^\mu + c_4 l_4^\mu$. The respective c_i coefficients can be easily related to scalar products $(q \cdot l_i)$ using (113) and (116). This leads to,[11]

$$q_\parallel^\mu = \frac{2}{\gamma}\left[(q \cdot l_1) l_2^\mu + (q \cdot l_2) l_1^\mu \right] \quad \text{and}$$

$$q_\perp^\mu = -\frac{1}{2\gamma}\left[(q\cdot l_3)l_4^\mu + (q\cdot l_4)l_3^\mu\right]. \tag{124}$$

The $q_\parallel^\mu$ component can be directly reduced to rank zero by reconstructing the scalar products $(q\cdot l_{1,2})$ in terms of D_0, D_1, D_2 using

$$p_i\cdot q = \frac{1}{2}\left(D_i - D_0 + f_{i0}\right), \quad f_{i0} = m_i^2 - m_0^2 - p_i^2. \tag{125}$$

This yields

$$q_\parallel^\mu = \frac{1}{\gamma}\left[f_{10}r_2^\mu + f_{20}r_1^\mu + D_1 r_2^\mu + D_2 r_1^\mu - D_0\left(r_1^\mu + r_2^\mu\right)\right], \tag{126}$$

with

$$r_1^\mu = l_1^\mu - \alpha_1 l_2^\mu \quad \text{and} \quad r_2^\mu = l_2^\mu - \alpha_2 l_1^\mu. \tag{127}$$

In order to obtain an identity that reduces also $q_\perp$ to a linear combination of $D_0, \ldots, D_3$ one has to move to rank two by squaring (123) in a way that does not generate $q_\parallel^\mu q_\parallel^\nu$ terms. To this end one can write

$$q^\mu q^\nu = \frac{1}{2}\left[\left(q^\mu - q_\perp^\mu\right)\left(q^\nu + q_\perp^\nu\right) + q_\perp^\mu q_\perp^\nu\right] + (\mu \leftrightarrow \nu)$$
$$= \frac{1}{2}\left[q_\parallel^\mu\left(q^\nu + q_\perp^\nu\right) + q_\perp^\mu q_\perp^\nu\right] + (\mu \leftrightarrow \nu). \tag{128}$$

Applying (126) to the rhs of (128) reduces $q_\parallel^\mu q^\nu$ and $q_\parallel^\mu q_\perp^\nu$ to rank one, such that only

$$q_\perp^\mu q_\perp^\nu = \frac{1}{4\gamma^2}\sum_{i,j=3}^4 (q\cdot\hat{l}_i)(q\cdot\hat{l}_j)\, l_i^\mu l_j^\nu, \tag{129}$$

with $\hat{l}_{3,4} = l_{4,3}$, remains to be reduced. This is achieved by means of the relations [2]

$$(q\cdot l_3)(q\cdot l_4) = \gamma\left(q_\parallel^\lambda q_\lambda - D_0 + m_0^2\right), \tag{130}$$

$$(q\cdot l_{3(4)})^2 = \frac{\gamma}{(p_3\cdot l_{4(3)})}$$
$$\times\left[\left(D_0 + m_0^2 - q_\parallel^\lambda q_\lambda\right)(p_3\cdot l_{3(4)})\right.$$
$$\left. - \left(D_3 - D_0 + f_{30} - 2p_{3,\alpha}q_\parallel^\alpha\right)(l_{3(4),\lambda}q^\lambda)\right], \tag{131}$$

where the quadratic terms $(q\cdot l_i)(q\cdot l_j)$ with $i, j = 3, 4$ are reconstructed in terms of D_0 and D_3 using also the external momentum p_3.

Combining (126)–(131) leads to the reduction identity (122) with

$$A_{-1}^{\mu\nu} = m_0^2 A_0^{\mu\nu}, \quad A_{1,2,3}^{\mu\nu} = 0,$$
$$A_0^{\mu\nu} = \frac{1}{4\gamma}\left(\alpha L_{33}^{\mu\nu} + \frac{1}{\alpha}L_{44}^{\mu\nu} - L_{34}^{\mu\nu}\right),$$

$$B_{-1,\lambda}^{\mu\nu} = \sum_{i=1}^3 f_{i0}B_{i,\lambda}^{\mu\nu}, \quad B_{0,\lambda}^{\mu\nu} = -\sum_{i=1}^3 B_{i,\lambda}^{\mu\nu},$$

$$B_{1,\lambda}^{\mu\nu} = \frac{1}{4\gamma^2}\left[\frac{2(p_3\cdot r_2)}{(p_3\cdot l_3)}\left(L_{33}^{\mu\nu}l_{4,\lambda} + \frac{1}{\alpha}L_{44}^{\mu\nu}l_{3,\lambda}\right)\right.$$
$$\left. - \left(r_2^\mu L_{34,\lambda}^\nu + r_2^\nu L_{34,\lambda}^\mu\right)\right] + \frac{1}{\gamma}\left(r_2^\mu\delta_\lambda^\nu - A_0^{\mu\nu}r_{2,\lambda}\right),$$

$$B_{2,\lambda}^{\mu\nu} = B_{1,\lambda}^{\mu\nu}\big|_{r_1\leftrightarrow r_2},$$

$$B_{3,\lambda}^{\mu\nu} = -\frac{1}{4\gamma(p_3\cdot l_3)}\left(L_{33}^{\mu\nu}l_{4,\lambda} + \frac{1}{\alpha}L_{44}^{\mu\nu}l_{3,\lambda}\right), \tag{132}$$

where we have introduced

$$L_{33}^{\mu\nu} = l_3^\mu l_3^\nu, \quad L_{44}^{\mu\nu} = l_4^\mu l_4^\nu,$$
$$L_{34}^{\mu\nu} = l_3^\mu l_4^\nu + l_4^\mu l_3^\nu, \quad \alpha = \frac{p_3\cdot l_4}{p_3\cdot l_3}. \tag{133}$$

The relations between the $A_j^{\mu\nu}$ and $B_{j,\lambda}^{\mu\nu}$ tensors in the first two lines of (132) follow from the requirement that terms of rank different from two vanish on the rhs of (122). Note also that the tensor $L_{34}^{\mu\nu}$ can be rewritten in terms of l_1, l_2 and $g^{\mu\nu}$ as

$$L_{34}^{\mu\nu} = 4\left(l_1^\mu l_2^\nu + l_2^\mu l_1^\nu - \frac{\gamma}{2}g^{\mu\nu}\right). \tag{134}$$

5.3 On-the-fly triangle reduction

The identity (82), which reconstructs $q^\mu q^\nu$ in terms of D_0, D_1, D_2 at the integral level, will be denoted as triangle reduction. Its derivation is based on the observation that the only terms that involve D_3 and p_3 in Sect. 5.2, i.e. the squared scalar products $(q\cdot l_3)^2$ and $(q\cdot l_4)^2$ in (131), do not contribute in three-point integrals of rank $R \le 3$. More precisely [2], for $i = 3, 4$,

$$\int d^D\bar{q}\,\frac{(q\cdot l_i)^2}{\bar{D}_0\bar{D}_1\bar{D}_2} = \int d^D\bar{q}\,\frac{(q\cdot l_i)^2 q^\rho}{\bar{D}_0\bar{D}_1\bar{D}_2} = 0. \tag{135}$$
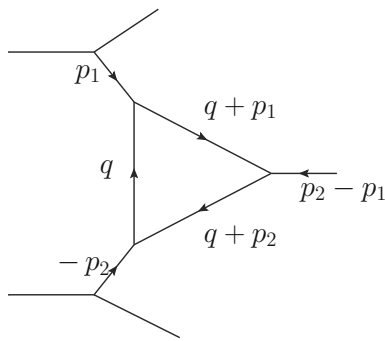
As a consequence, the derivations of Sect. 5.2 are also applicable to three-point functions at the integral level upon replacing (129) by

$$q_\perp^\mu q_\perp^\nu \to \frac{1}{4\gamma^2}(q\cdot l_3)(q\cdot l_4)\,L_{34}^{\mu\nu}. \tag{136}$$

In this way one arrives at the reduction identities

$$\int d^D\bar{q}\,\frac{q^\mu q^\nu S(q)}{\bar{D}_0\cdots\bar{D}_2}$$
$$= \sum_{j=-1}^2 \int d^D\bar{q}\,\frac{\left(A_j^{\mu\nu} + B_{j,\lambda}^{\mu\nu}q^\lambda\right)S(q)}{\bar{D}_0\cdots\bar{D}_j\cdots\bar{D}_2}, \tag{137}$$

where $S(q) = S + S_\rho q^\rho$ is an arbitrary rank-one polynomial, and the tensors $A_j^{\mu\nu}$ and $B_{j,\lambda}^{\mu\nu}$ are obtained from (132) through

Fig. 11 Triangle t-channel (sub)topology that gives rise to $\Delta_{12} \to 0$ numerical instabilities when $(p_2 - p_1)^2 = 0$ and $p_1^2 \to p_2^2$

the trivial replacements

$$L_{33}^{\mu\nu} \to 0, \quad L_{44}^{\mu\nu} \to 0. \tag{138}$$

5.4 Treatment of Gram-determinant instabilities

As pointed out in Sect. 5.1, when the rank-two Gram determinant Δ_{12} tends to zero the reduction basis (114)–(115) becomes degenerate. This leads to spurious singularities that manifest themselves as factors $\gamma^{-k} \propto \Delta_{12}^{-k}$ in the reduction identities. In practice, the residues of Δ_{12}^{-k} poles are suppressed at $\mathcal{O}(\Delta_{12}^k)$ as a result of subtle numerical cancellations between various contributions. Thus, for $\Delta_{12} \to 0$ the results of the reduction are finite but suffer from severe numerical instabilities. As can be seen from (132), spurious singularities reach the maximum power $k = 2$, i.e. each reduction step results in a numerical instability that scales quadratically in the inverse Gram determinant Δ_{12}.

The reduction (132) involves also spurious singularities related to the rank-three Gram determinant, Δ_{123}, which enter through the terms $(l_3 \cdot p_3)^{-1} \propto |\Delta_{123}|^{-1/2}$ [2]. However, as compared to the $\Delta_{123} \to 0$ case, $\Delta_{12} \to 0$ instabilities are statistically more likely and are also more enhanced due to their Δ_{12}^{-2} scaling behaviour. In fact, studying high statistics samples for various representative processes, we have found that the numerical instabilities of the on-the-fly reductions of Sects. 5.2–5.3 are very strongly correlated to the parameter $\gamma \propto \Delta_{12}$. Therefore, as we will see in the following, avoiding $\Delta_{12} \to 0$ spurious singularities in a systematic way makes it possible to reach excellent numerical stability.

5.4.1 Box reduction

In the case of the on-the-fly box reduction (122), poles in Δ_{12} arise only through the factors $1/\gamma^2$ and $1/\gamma$ in (132), which are a direct consequence of the choice of the momenta p_1, p_2 for the construction of the reduction basis (114)–(115). Since rank-two Gram-determinant instabilities depend on only two

of the three available momenta, they can be easily avoided by constructing the basis with p_1, p_3 or p_2, p_3 instead of p_1, p_2, depending on the values of the respective Gram determinants Δ_{13}, Δ_{23}, Δ_{12}.

In practice, in order to avoid small rank-two Gram determinants we perform the box reduction upon a permutation,

$$\{D_1, D_2, D_3\} \longrightarrow \{D_{i_1}, D_{i_2}, D_{i_3}\}, \tag{139}$$

which orders loop denominators with related momenta and masses in such a way that

$$\frac{|\Delta_{i_1 i_2}|}{Q_{i_1 i_2}^4} = \max \left\{ \frac{|\Delta_{12}|}{Q_{12}^4}, \frac{|\Delta_{13}|}{Q_{13}^4}, \frac{|\Delta_{23}|}{Q_{23}^4} \right\}. \tag{140}$$

The scales Q_{ij}^2, which render the above ratios dimensionless, are defined as the largest element of the respective Gram matrices, i.e.

$$Q_{ij}^2 = \max\{|p_i \cdot p_i|, |p_i^2|, |p_j^2|\}. \tag{141}$$

Note that the permutation (139) can be applied without changing the order of the corresponding segments $S_i(q)$, i.e. without any modification of the open-loop dressing recursion. Moreover, the choice of the optimal permutation (139)–(140) can be done in a fully flexible way at runtime and locally in individual reduction steps, depending on the kinematics of the actual phase space point. In practice (139) is applied only to compute the reduction basis and the coefficients (132), which are then converted back to the original ordering.

Avoiding a spurious $\Delta_{12} \to 0$ singularity with (139)–(140) does not guarantee its disappearance in future reduction steps. In fact, all reduced contributions where D_1 and D_2 remain unpinched will still involve the same small Gram determinant. However, the permutation trick (139) can be iterated as long as $N \geq 4$ loop denominators are available. In this way, rank-two Gram-determinant instabilities can be isolated in triangle contributions, which arise only at later steps of the open-loop recursion for loop diagrams with $N > 3$.

5.4.2 Triangle reduction

For the case of triangle topologies one can show that, excluding regions where the external particles become soft or collinear, vanishing Δ_{12} Gram determinants can arise only from the t-channel topology depicted in Fig. 11, where the triangle exhibits two space-like external momenta, p_1 and p_2, and a time-like external momentum, $p_2 - p_1$. Since the Gram determinant vanishes when $p_1^2 \to p_2^2$, we adopt the parametrisation

$$p_1^2 = -p^2 < 0, \quad p_2^2 = -p^2(1 + \delta), \quad (p_2 - p_1)^2 = 0, \tag{142}$$

where p_1 and p_2 can be ordered such that $\delta > 0$. The parameters Δ and γ are related to δ via

$$\sqrt{\Delta} = \frac{p^2}{2}\delta \quad \text{and} \quad \gamma = -p^2\delta^2, \tag{143}$$

i.e. the $\Delta \to 0$ limit corresponds to $\delta \to 0$.

In kinematic regions with small δ, the numerical reduction of rank-r triangles via iterated on-the-fly reductions (132) and subsequent rank-one reductions (A.12)–(A.14) can lead to spurious singularities up to order $1/\delta^{4r-2}$. In order to avoid numerical instabilities, we first perform a full *algebraic* reduction of 3-point tensor integrals,

$$C^{\mu_1\cdots\mu_r} = \frac{(2\pi\mu)^{2\varepsilon}}{i\pi^2}\int d^D\bar{q}\, \frac{q_1^\mu \cdots q^{\mu_r}}{\bar{D}_0\bar{D}_1\bar{D}_2}, \tag{144}$$

to scalar integrals. This leads to a cancellation of the leading spurious singularites, and the resulting analytic expressions for rank-r triangles involve only poles up to order $1/\delta^{r+1}$. For instance, for the case of triangles with massless internal lines, $m_0 = m_1 = m_2 = 0$, reducing tensor integrals of rank $r = 1, 2, 3$ to scalar integrals we obtain

$$C^\mu = \frac{2}{\delta^2 p^2}\left\{B_0(-p^2, 0, 0)\left[-p_1^\mu\left(1 + \frac{\delta}{2}\right) + p_2^\mu\right]\right.$$
$$\left. + B_0\left(-p^2(1+\delta), 0, 0\right)\left[p_1^\mu(1+\delta) - p_2^\mu\left(1 + \frac{\delta}{2}\right)\right]\right\}$$
$$+ \frac{1}{\delta}C_0\left(-p^2, -p^2(1+\delta), 0, 0, 0\right)\left[-p_1^\mu(1+\delta) + p_2^\mu\right]$$
$$+ \frac{1}{\delta p^2}B_0(0, 0, 0)\left[p_2^\mu - p_1^\mu\right], \tag{145}$$

$$C^{\mu\nu} = B_0(-p^2, 0, 0)\left[-\frac{g^{\mu\nu}}{4\delta} + \frac{p_{11}^{\mu\nu}}{p^2}\left(\frac{3}{\delta^3} + \frac{5}{\delta^2} + \frac{3}{2\delta}\right)\right.$$
$$\left. - \frac{p_{12}^{\mu\nu}}{p^2}\left(\frac{3}{\delta^3} + \frac{5}{2\delta^2}\right) + \frac{p_{22}^{\mu\nu}}{p^2}\frac{3}{\delta^3}\right] + B_0\left(-p^2(1+\delta), 0, 0\right)$$
$$\times \left[\frac{g^{\mu\nu}}{4\delta} + \frac{1}{4}g^{\mu\nu} - \frac{p_{11}^{\mu\nu}}{p^2}\left(\frac{3}{\delta^3} + \frac{6}{\delta^2} + \frac{3}{\delta}\right) + \frac{p_{12}^{\mu\nu}}{p^2}\left(\frac{3}{\delta^3}\right.\right.$$
$$\left.\left. + \frac{7}{2\delta^2} + \frac{1}{2\delta}\right) + \frac{p_{22}^{\mu\nu}}{p^2}\left(-\frac{3}{\delta^3} - \frac{1}{\delta^2} + \frac{1}{2\delta}\right)\right] + B_0(0, 0, 0)$$
$$\times \left[\frac{p_{11}^{\mu\nu}}{p^2}\left(\frac{1}{\delta^2} + \frac{3}{2\delta}\right) - \frac{p_{12}^{\mu\nu}}{p^2}\left(\frac{1}{\delta^2} + \frac{1}{2\delta}\right) + \frac{p_{22}^{\mu\nu}}{p^2}\left(\frac{1}{\delta^2}\right.\right.$$
$$\left.\left. - \frac{1}{2\delta}\right)\right] + C_0\left(-p^2, -p^2(1+\delta), 0, 0, 0\right)\left[p_{11}^{\mu\nu}\left(\frac{1}{\delta^2}\right.\right.$$
$$\left.\left. + \frac{2}{\delta} + 1\right) - p_{12}^{\mu\nu}\left(\frac{1}{\delta^2} + \frac{1}{\delta}\right) + \frac{p_{22}^{\mu\nu}}{\delta^2}\right] - \frac{1}{2}\left[-\frac{1}{2}g^{\mu\nu}\right.$$
$$\left. + \frac{p_{11}^{\mu\nu}}{p^2}\left(\frac{2}{\delta^2} + \frac{2}{\delta}\right) - \frac{p_{12}^{\mu\nu}}{p^2}\left(\frac{2}{\delta^2} + \frac{1}{\delta}\right) + \frac{2p_{22}^{\mu\nu}}{\delta^2 p^2}\right], \tag{146}$$

$$C^{\mu\nu\rho} = B_0(-p^2, 0, 0)\left[p_{111}^{\mu\nu\rho}\left(-\frac{11}{3\delta^4 p^2} - \frac{10}{\delta^3 p^2}\right.\right.$$
$$\left. - \frac{17}{2\delta^2 p^2} - \frac{11}{6\delta p^2}\right) + p_{112}^{\mu\nu\rho}\left(\frac{11}{3\delta^4 p^2} + \frac{20}{3\delta^3 p^2} + \frac{17}{6\delta^2 p^2}\right)$$
$$\left. + p_{122}^{\mu\nu\rho}\left(-\frac{11}{3\delta^4 p^2} - \frac{10}{3\delta^3 p^2}\right) + p_{222}^{\mu\nu\rho}\frac{11}{3\delta^4 p^2} + p_1^{\{\mu}g^{\nu\rho\}}\right.$$

$$\times \left(\frac{1}{12\delta^2} + \frac{1}{6\delta}\right) - \frac{p_2^{\{\mu}g^{\nu\rho\}}}{12\delta^2}\right] + B_0\left(-p^2(1+\delta), 0, 0\right)$$
$$\times \left[p_{111}^{\mu\nu\rho}\left(\frac{11}{3\delta^4 p^2} + \frac{11}{\delta^3 p^2} + \frac{11}{\delta^2 p^2} + \frac{11}{3\delta p^2}\right) + p_{112}^{\mu\nu\rho}\right.$$
$$\times \left(-\frac{11}{3\delta^4 p^2} - \frac{23}{3\delta^3 p^2} - \frac{13}{3\delta^2 p^2} - \frac{1}{3\delta p^2}\right) + p_{122}^{\mu\nu\rho}\left(\frac{11}{3\delta^4 p^2}\right.$$
$$\left. + \frac{13}{3\delta^3 p^2} + \frac{1}{2\delta^2 p^2} - \frac{1}{6\delta p^2}\right) + p_{222}^{\mu\nu\rho}\left(-\frac{11}{3\delta^4 p^2} - \frac{1}{\delta^3 p^2}\right.$$
$$\left. + \frac{1}{2\delta^2 p^2} - \frac{1}{3\delta p^2}\right) + p_1^{\{\mu}g^{\nu\rho\}}\left(-\frac{1}{12\delta^2} - \frac{1}{6\delta} - \frac{1}{12}\right)$$
$$\left. + p_2^{\{\mu}g^{\nu\rho\}}\left(\frac{1}{12\delta^2} - \frac{1}{12}\right)\right] + B_0(0, 0, 0)\left[p_{111}^{\mu\nu\rho}\left(-\frac{1}{\delta^3 p^2}\right.\right.$$
$$\left. - \frac{5}{2\delta^2 p^2} - \frac{11}{6\delta p^2}\right) + p_{112}^{\mu\nu\rho}\left(\frac{1}{\delta^3 p^2} + \frac{3}{2\delta^2 p^2} + \frac{1}{3\delta p^2}\right)$$
$$\left. + p_{122}^{\mu\nu\rho}\left(-\frac{1}{\delta^3 p^2} - \frac{1}{2\delta^2 p^2} + \frac{1}{6\delta p^2}\right) + p_{222}^{\mu\nu\rho}\left(\frac{1}{\delta^3 p^2}\right.\right.$$
$$\left.\left. - \frac{1}{2\delta^2 p^2} + \frac{1}{3\delta p^2}\right)\right] + C_0\left(-p^2, -p^2(1+\delta), 0, 0, 0\right)$$
$$\times \left[p_{111}^{\mu\nu\rho}\left(-\frac{1}{\delta^3} - \frac{3}{\delta^2} - \frac{3}{\delta} - 1\right) + p_{112}^{\mu\nu\rho}\left(\frac{1}{\delta^3} + \frac{2}{\delta^2} + \frac{1}{\delta}\right)\right.$$
$$\left. + p_{122}^{\mu\nu\rho}\left(-\frac{1}{\delta^3} - \frac{1}{\delta^2}\right) + \frac{p_{222}^{\mu\nu\rho}}{\delta^3}\right] - \frac{1}{2}\left[p_{111}^{\mu\nu\rho}\left(-\frac{10}{3\delta^3 p^2}\right.\right.$$
$$\left. - \frac{22}{3\delta^2 p^2} - \frac{37}{9\delta p^2}\right) + p_{112}^{\mu\nu\rho}\left(\frac{10}{3\delta^3 p^2} + \frac{14}{3\delta^2 p^2} + \frac{10}{9\delta p^2}\right)$$
$$\left. + p_{122}^{\mu\nu\rho}\left(-\frac{10}{3\delta^3 p^2} - \frac{2}{\delta^2 p^2} + \frac{2}{9\delta p^2}\right) + p_{222}^{\mu\nu\rho}\left(\frac{10}{3\delta^3 p^2}\right.\right.$$
$$\left. - \frac{2}{3\delta^2 p^2} + \frac{1}{9\delta p^2}\right) + p_1^{\{\mu}g^{\nu\rho\}}\left(\frac{1}{6\delta} + \frac{5}{18}\right) + p_2^{\{\mu}g^{\nu\rho\}}$$
$$\times \left(\frac{1}{9} - \frac{1}{6\delta}\right)\right], \tag{147}$$

with the tensors

$$p_{ij}^{\mu\nu} = \sum_{\pi(i,j)} p_i^\mu p_j^\nu, \quad p_{ijk}^{\mu\nu\rho} = \sum_{\pi(i,j,k)} p_i^\mu p_j^\nu p_k^\rho,$$
$$p_i^{\{\mu}g^{\nu\rho\}} = p_i^\mu g^{\nu\rho} + p_i^\nu g^{\mu\rho} + p_i^\rho g^{\mu\nu}, \tag{148}$$

where $i, j, k = 1, 2$, and the sums are restricted to inequivalent permutations, e.g. $p_{11}^{\mu\nu} = p_1^\mu p_1^\nu$, $p_{112}^{\mu\nu\rho} = p_1^\mu p_1^\nu p_2^\rho + p_1^\mu p_2^\nu p_1^\rho + p_2^\mu p_1^\nu p_1^\rho$, etc.

Analytic expression of type (145)–(147) guarantee a reduced sensitivity to Gram-determinant instabilities. Thus they are used as default for the reduction of triangles configurations of type (142) with $\delta > \delta_{\text{thr}}$. The freely adjustable threshold parameter δ_{thr} is set to $\delta_{\text{thr}} = 10^{-3}$. To avoid numerical instabilities in regions with $\delta < \delta_{\text{thr}}$ we perform systematic expansions in δ. In particular, for a complete cancellation of the $1/\delta$ poles also the δ-dependent C_0 and B_0 scalar integrals have to be expanded in δ. To this end we use LITERED [67], and expanding the residues of δ^{-k} poles up to order δ^{k+m} we obtain regular Taylor series including terms up to order δ^m.

For the case $m_1 = m_2 = m_3 = 0$, expanding up to order δ^2 yields

$$C^\mu = \frac{p_1^\mu + p_2^\mu}{2p^2}(1 - \Delta B_0) + \delta\,\frac{p_1^\mu + 2p_2^\mu}{6p^2}\Delta B_0$$
$$- \delta^2\,\frac{p_1^\mu + 3p_2^\mu}{12p^2}\left(\Delta B_0 + \frac{1}{2}\right) + \mathcal{O}(\delta^3), \qquad (149)$$

$$C^{\mu\nu} = \frac{2p_{11}^{\mu\nu} + p_{12}^{\mu\nu} + 2p_{22}^{\mu\nu}}{6p^2}\left(\Delta B_0 - \frac{1}{2}\right) + \frac{1}{4}g^{\mu\nu}B_0$$
$$- \delta\left[\frac{p_{11}^{\mu\nu} + p_{12}^{\mu\nu} + 3p_{22}^{\mu\nu}}{12p^2}\left(\Delta B_0 + \frac{1}{2}\right) - \frac{1}{8}g^{\mu\nu}\right]$$
$$+ \delta^2\left[\frac{2p_{11}^{\mu\nu} + 3p_{12}^{\mu\nu} + 12p_{22}^{\mu\nu}}{60p^2}(\Delta B_0 + 1)\right.$$
$$\left. + \frac{1}{24}g^{\mu\nu}\right] + \mathcal{O}(\delta^3), \qquad (150)$$

$$C^{\mu\nu\rho} = \frac{3p_{111}^{\mu\nu\rho} + p_{112}^{\mu\nu\rho} + p_{122}^{\mu\nu\rho} + 3p_{222}^{\mu\nu\rho}}{12p^2}\left(\frac{1}{6} - \Delta B_0\right)$$
$$- \frac{p_1^{\{\mu}g^{\nu\rho\}} + p^2 p_2^{\{\mu}g^{\nu\rho\}}}{72}(6B_0 + 1)$$
$$+ \delta\left[\frac{3p_{111}^{\mu\nu\rho} + 2p_{112}^{\mu\nu\rho} + 3p_{122}^{\mu\nu\rho} + 12p_{222}^{\mu\nu\rho}}{60p^2}\right.$$
$$\times \left(\Delta B_0 + \frac{5}{6}\right) + \frac{1}{36}p_1^{\{\mu}g^{\nu\rho\}} + \left.\frac{1}{18}p_2^{\{\mu}g^{\nu\rho\}}\right]$$
$$- \delta^2\left[\frac{p_{111}^{\mu\nu\rho} + p_{112}^{\mu\nu\rho} + 2p_{122}^{\mu\nu\rho} + 10p_{222}^{\mu\nu\rho}}{60p^2}\right.$$
$$\times \left(\Delta B_0 + \frac{4}{3}\right) - \frac{1}{144}p_1^{\{\mu}g^{\nu\rho\}} - \left.\frac{1}{48}p_2^{\{\mu}g^{\nu\rho\}}\right]$$
$$+ \mathcal{O}(\delta^3), \qquad (151)$$

where

$$B_0 = B_0(-p^2, 0, 0),$$
$$\Delta B_0 = B_0(-p^2, 0, 0) - B_0(0, 0, 0). \qquad (152)$$

Similar results have been obtained for the case of massive internal propagators. More precisely, we have implemented all needed mass configurations for NLO QCD calculations, i.e. $(m_0, m_1, m_2) = (0, 0, 0)$, (m, m, m), $(0, m, m)$ and $(m, 0, 0)$, with $m > 0$, including terms up to order δ^2 in the expansions. The extra cases needed for NLO EW calculations, i.e. $(0, m_1, m_2)$, (m_1, m_2, m_2) and (m_1, m_2, m_3), with $m_i > 0$, will be implemented soon.

Since the analytic expressions (145)–(147) and their expansions (149)–(151) reduce triangles of rank r to rank zero in a single step, in order to be able to apply them in a fully flexible way, the on-the-fly reduction of triangles (82) has to be postponed at the end of the open-loop recursion, where triangle contributions have reached the maximum rank. This means that, in addition to the contributions listed in (84), we also generate $N = 3$ terms with $R = 3$. After taking the trace (76), depending on the actual value of δ, the reduction
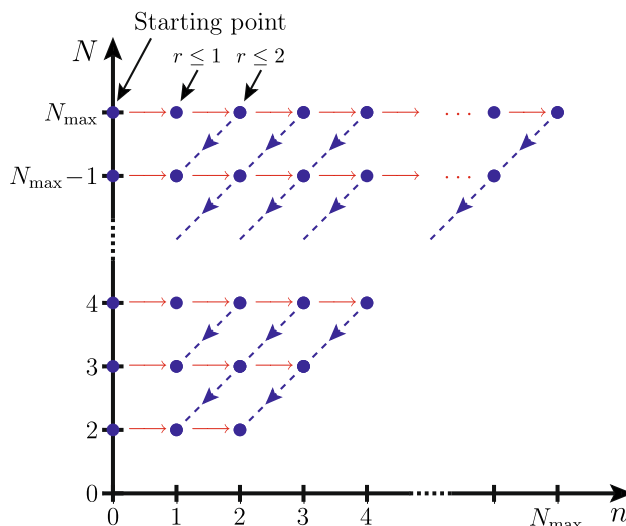
Fig. 12 Evolution of the total number N of loop segments and the number n of dressed segments during the open-loop recursion. Horizontal and diagonal arrows describe, respectively, dressing steps and the generation of pinched subtopologies in the on-the-fly reduction. Corresponding unpinched contributions, where the rank is reduced but (N, n) remains unchanged are not shown. The algorithm starts at $(N_{max}, 0)$ and first proceeds towards the highest possible n before moving one step lower in N

of triangles is done either using the expansion formulas or the on-the-fly reduction (82) followed by Passarino–Veltman reduction steps (A.12)–(A.14).

6 Implementation and performance

This section summarises the key structure of the new algorithm, outlines some aspects of its implementation, and presents technical performance studies.

6.1 Structure and implementation of the new algorithm

Similarly as for the original open-loop method, given a certain scattering process the algorithm starts with the generation of all tree and one-loop diagrams. This is done in symbolic form, including only topological information and particle content. One-loop diagrams are colour stripped and cut-open, and the interference of their colour structure with the Born amplitude is taken as initial condition for the open-loop recursion.

The recursion is organised by grouping open loops according to the total number N of segments and the number n of dressed segments. Dressing steps increase n and reduction steps reduce N or keep it constant. Thus, as illustrated in Fig. 12, the (N, n) groups are dressed through a series of iterations with $N = N_{max}, N_{max} - 1, N_{max} - 2, \ldots$, where N is kept fixed while all segments $n = 0, 1, 2, \ldots, N$ are

processed. A step of the open-loop recursion, to be applied to all objects in an (N, n)-group, consists of the following operations:

1. Merge all open loops with the same one-loop topology, the same cut and the same undressed segments into a single object. Note that one can merge open loops of different rank.

2. For each open loop with $n < N$ and rank R determine the rank R_{next} that would be reached by performing the subsequent dressing step.

2(a) If $R_{\text{next}} = 3$ or if $n = N$ and $R = 2$ avoid the dressing step and perform an on-the-fly reduction, which generates unpinched and pinched terms with $R = 1$. The unpinched terms remain in the (N, n) group. As for the pinched terms, if the adjacent segments of a pinched propagator are dressed they can be reduced to a single effective segment, thereby turning $(N, n) \rightarrow (N - 1, n - 1)$. Otherwise, contributions with an undressed pinch stay in the (N, n) group.

2(b) If $n < N$ and $R_{\text{next}} \leq 2$ perform a dressing step, which turns $(N, n) \rightarrow (N, n + 1)$. If this step dresses a pinched propagator that was previously undressed, the corresponding segments can be reduced to a single effective segment turning $(N, n + 1) \rightarrow (N - 1, n)$.

3. Sort all open loops into the proper group and repeat steps 1–3 for all open loops in the (N, n)-group until the group is empty or $n = N$ and all open loops in the group have rank $R \leq R_{\text{last}}$ (see below).

Topologies with $N = 2$ are dressed without reduction, and, in order to enable Gram-determinant expansions (see Sect. 5.4.2), also $N = 3$ contributions are dressed without on-the-fly reduction. Thus $R_{\text{last}} = N$ for $N = 2, 3$, while open loops with $N \geq 4$ are reduced on-the-fly down to $R_{\text{last}} = 1$. The algorithm starts at $(N, n) = (N_{\text{max}}, 0)$ and terminates with the dressing of two-point contributions at $(N, n) = (2, 2)$. At this point all open loops are closed with the trace operation (76), and the last reduction steps described in Appendix A are applied.

The above on-the-fly algorithm has been implemented in the framework of the original OPENLOOPS program [16], which consists of a computer-algebraic code generator written in MATHEMATICA and a numerical part written in FORTRAN 90. Given an arbitrary Standard Model process, the MATHEMATICA generator simulates the full chain of recursion steps in symbolic form and translates it into FORTRAN 90 code for the calculation of the actual scattering amplitude. The only external tools that need to be interfaced to the new OPENLOOPS program are FEYNARTS [68], for the generation of tree and one-loop diagrams, and COLLIER [19], for the calculation of scalar integrals. All other aspects of the open-loop method are directly implemented as process-

independent MATHEMATICA and FORTRAN routines. This includes the management of colour algebra, the kernels of the dressing recursion at tree and one-loop level, the on-the-fly and integral reductions, the helicity bookkeeping system, R_2 rational terms, UV counterterms, and several other aspects.

The entire program is fully automated, the new on-the-fly methods are implemented and widely tested at NLO QCD and they will soon be extended to NLO EW. These methods will be made publicly available with the upcoming release of OPENLOOPS 2. Similarly as for OPENLOOPS 1, numerical routines generated with the new on-the-fly techniques will be accessible through an automated download and installation system and the standard OPENLOOPS interfaces to a variety of public Monte Carlo programs. In addition to the on-the-fly approach, OPENLOOPS 2 will support also the original open-loop method, which requires additional third-party tools such as COLLIER or CUTTOOLS [10] and ONELOOP [65] for the reduction to scalar integrals.

6.2 Technical performance

In this section we study the technical performance of the new algorithm. Similarly as in [9], we present speed and stability benchmarks for the one-loop QCD corrections to four families of partonic processes,

$$
\begin{align}
&\text{(a) } u\bar{u} \rightarrow t\bar{t} + n\,g, \\
&\text{(b) } gg \rightarrow t\bar{t} + n\,g, \\
&\text{(c) } u\bar{d} \rightarrow W^+ g + n\,g, \\
&\text{(d) } u\bar{u} \rightarrow W^+ W^- + n\,g,
\end{align}
\tag{153}
$$

with $n = 0, 1, 2, 3$ additional gluons, i.e. including processes with up to 7 scattering particles. Top quarks and W bosons are not decayed, and sums over the colour and helicity degrees of freedom of the external particles are included throughout.

Benchmarks obtained with the new open-loop algorithm, denoted as OPENLOOPS 2, are based on COLLIER for the calculation of scalar integrals. In order to highlight the effect of the new on-the-fly methods of Sects. 3–4, we also consider variants of the OPENLOOPS program where these new methods are not used. Specifically, as detailed in Table 1, we restrict the on-the-fly approach to helicity sums and diagram merging, using COLLIER for tensor reduction. This approach is denoted as OPENLOOPS 2+COLLIER. Alternatively, we apply the original open-loop method in combination with tensor integrals (denoted as OPENLOOPS 1+COLLIER) or OPP reduction (denoted as OPENLOOPS 1+CUTTOOLS). The OPENLOOPS 1+CUTTOOLS mode relies on ONELOOP for the scalar integrals and is used also to generate benchmarks in quadruple precision.

By default, OPENLOOPS calculations are monitored through a built-in stability system that estimates the level of instability of one-loop results and automatically triggers re-evaluations in double or quadruple precision for critical

Table 1 List of the different variants of the OPENLOOPS program that are compared in Sects. 6.2.1–6.2.2. As third-party tools we use COLLIER1.2 [19], CUTTOOLS1.9.5 [10] and ONELOOP3.6.1 [65]

Method	Tensor reduction	Scalar integrals	Diagram merging	Helicity summation
OPENLOOPS 2	On-the-fly	COLLIER	On-the-fly	On-the-fly
OPENLOOPS 2+COLLIER	COLLIER	COLLIER	On-the-fly	On-the-fly
OPENLOOPS 1+COLLIER	COLLIER	COLLIER	Parent–child	Standard
OPENLOOPS 1+CUTTOOLS	CUTTOOLS	ONELOOP	Parent–child	Standard
Quad Precision	CUTTOOLS	ONELOOP	Parent–child	Standard

Fig. 13 Runtimes per phase space point for the calculation of the one-loop scattering probability (1) on a single Intel i7-4790K core with gfortran-4.8.5. Results for the processes in (153) are plotted versus the number of one-loop diagrams. Timings of OPENLOOPS 2 with on-the-fly reduction ($t^{\mathrm{OL2}}_{\mathrm{on\text{-}the\text{-}fly}}$) are shown in the upper frame. The lower frame presents a comparison to OPENLOOPS 2+COLLIER and OPENLOOPS 1+COLLIER (see Table 1)
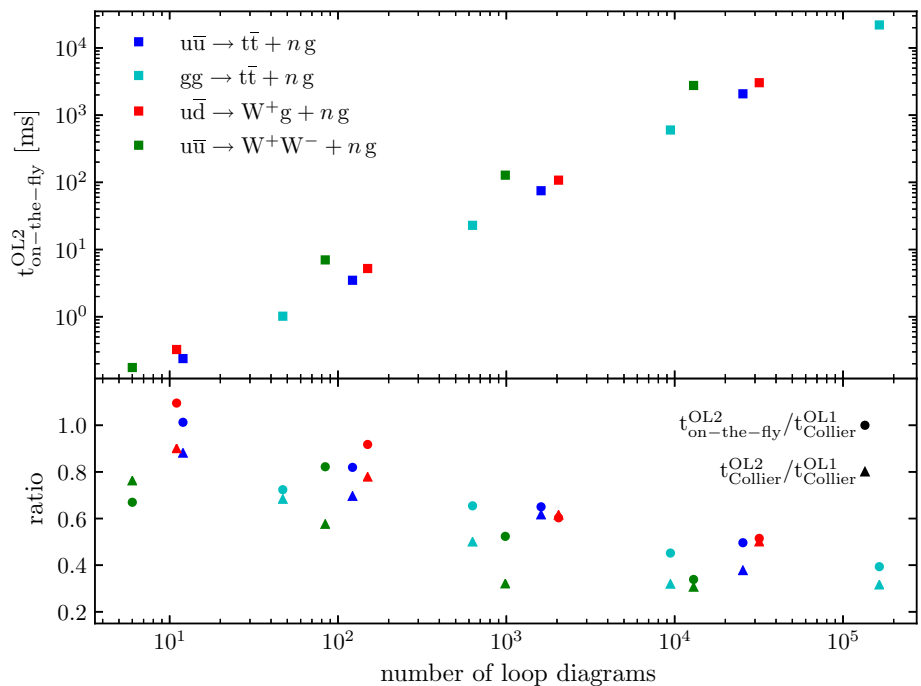
phase space points. In the following, in order to avoid any bias in the comparisons, the stability system is switched off. In this way, one-loop amplitudes are computed only once per phase space point in double precision. Unless stated otherwise, Gram-determinant expansions are always kept active, both in the on-the-fly reduction of OPENLOOPS2 and in COLLIER.

6.2.1 Speed benchmarks

To illustrate the speed of the new algorithm, in Fig. 13 we plot runtimes per phase space point for the calculation of the one-loop scattering probability (1). The processes listed in (153) involve a number of one-loop Feynman diagrams that ranges from the order of 1 to 10^5. The corresponding runtimes, measured on a single Intel i7-4790K core with gfortran-4.8.5, vary from the order of 10^{-1} to 10^4 ms. In this range, we confirm that runtimes tend to grow linearly with the number of one-loop diagrams up to $2 \to 4$ processes [9], and we find that this scaling behaviour persists up to $2 \to 5$

processes. As compared to OPENLOOPS 1+COLLIER, the new algorithm with on-the-fly reduction is up to a factor 2–3 faster for multi-particle processes. Depending on the process, using OPENLOOPS 2+COLLIER, i.e. restricting the on-the-fly approach to helicity sums plus diagram merging and reducing tensor integrals with COLLIER, can result in a further significant speed-up. However, the moderate slowdown caused by the on-the-fly reduction can be counterbalanced by the improved numerical stability, which implies a reduced need of re-evaluations in quadruple precision (see Sect. 6.2.2).

6.2.2 Stability benchmarks

In this section we study the numerical stability of the new open-loop algorithm. To this end, one-loop scattering probability densities computed in double precision ($\mathcal{W}^{\mathrm{DP}}_{1-\mathrm{loop}}$) are compared against benchmarks in quadruple precision ($\mathcal{W}^{\mathrm{QP}}_{1-\mathrm{loop}}$). More precisely, defining the relative difference between two results as
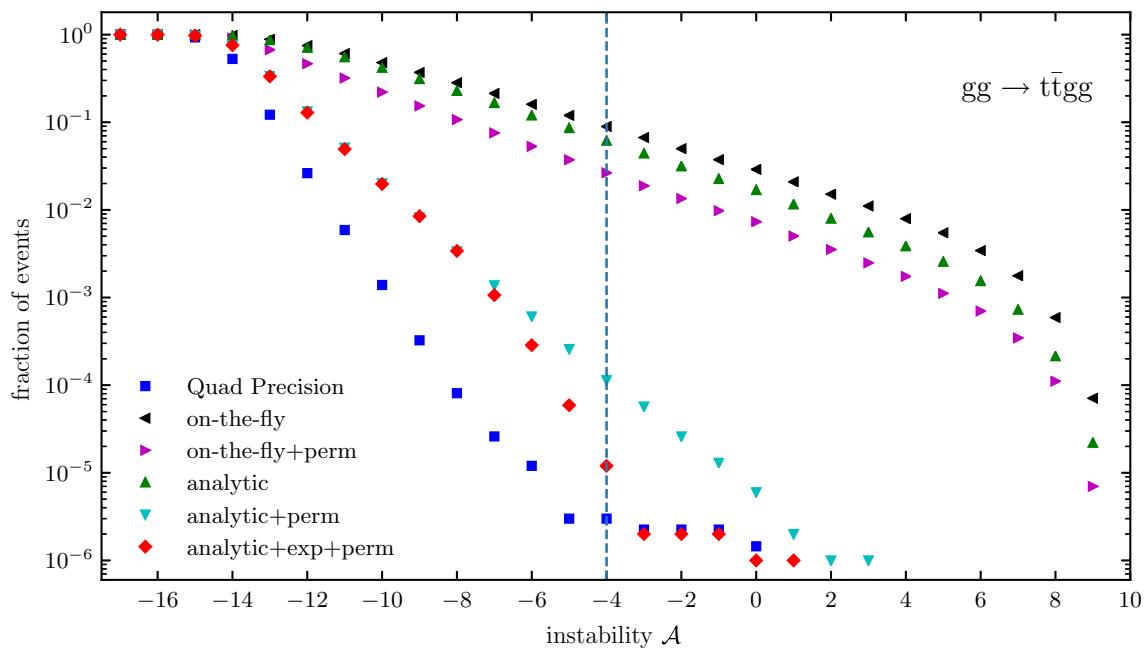
Fig. 14 Probability of finding events with instability $\mathcal{A} > \mathcal{A}_{\min}$ as a function of $\mathcal{A}_{\min}$ in a sample of 10^6 gg $\rightarrow$ t$\bar{\text{t}}$gg events. The stability of quad-precision benchmarks (blue) is compared to different variants of the OPENLOOPS 2 algorithm in double precision. Unstable results without special treatment of Gram determinants ("on-the-fly") are stabilised using the permutation trick (139) for boxes ("perm"), exact analytic formulas for triangles ("analytic") and Gram-determinant expansions for $\delta < \delta_{\text{thr}}$ ("exp"). See Sect. 5.4.2

$$\mathcal{A}(\mathcal{W}_a, \mathcal{W}_b) = \log_{10} \left| \frac{\mathcal{W}_a - \mathcal{W}_b}{\mathcal{W}_{\min}} \right| \quad \text{with}$$
$$\mathcal{W}_{\min} = \min \{|\mathcal{W}_a|, |\mathcal{W}_b|\}, \tag{154}$$

we estimate the instability of double-precision results as[12]

$$\mathcal{A}_{\text{DP}} = \mathcal{A}\left(\mathcal{W}_{1-\text{loop}}^{\text{DP}}, \mathcal{W}_{1-\text{loop}}^{\text{QP,R}}\right). \tag{155}$$

This quantity can be regarded, up to a minus sign, as the number of correct digits of the double-precision evaluation.

To estimate the intrinsic accuracy of quad-precision benchmarks, computed using OPENLOOPS 1+CUTTOOLS and ONELOOP, we use a so-called rescaling test [9,69], where scattering amplitudes are computed with rescaled masses and momenta and scaled back according to their mass dimensionality. Thus for a given phase space point the accuracy of the quadruple precision benchmarks is assessed as

$$\mathcal{A}_{\text{QP}} = \mathcal{A}\left(\mathcal{W}_{1-\text{loop}}^{\text{QP}}, \mathcal{W}_{1-\text{loop}}^{\text{QP,R}}\right), \tag{156}$$

where $\mathcal{W}_{1-\text{loop}}^{\text{QP}}$ and $\mathcal{W}_{1-\text{loop}}^{\text{QP,R}}$ are the original and rescaled quad-precision evaluations. This quantity represents the finite resolution of the instability estimate (155). As we will

see, quad-precision benchmarks can become more unstable than double-precision results obtained with OPENLOOPS 2. In this case, the instability estimate (155) yields $\mathcal{A}_{\text{DP}} \sim \mathcal{A}_{\text{QP}}$ but should be interpreted as $\mathcal{A}_{\text{DP}} < \mathcal{A}_{\text{QP}}$.

To assess the stability of OPENLOOPS 2, for each process in (153) we have studied a sample of 10^6 homogeneously distributed phase space points at $\sqrt{s} = 1$ TeV. To exclude soft and collinear regions we have required $p_{i,\text{T}} > 50$ GeV and $\Delta R_{ij} > 0.5$ for all massless final-state QCD partons.

Figure 14 illustrates the effect of Gram-determinant instabilities and the goodness of the solutions introduced in Sect. 5.4 in the case of gg $\rightarrow$ t$\bar{\text{t}}$gg. For this challenging multi-particle process, using the OPENLOOPS 2 on-the-fly reductions without any special treatment of Gram determinants we observe an extremely high level of numerical instability in double precision. The probabilities to obtain one-loop results with less than four or zero correct digits are around 10^{-1} and 10^{-2}, respectively, and the tail of the stability distribution extends up to a level of instability of ten orders of magnitude and more. Applying the permutation trick (139) ("perm") and using analytic expressions for three-point integrals ("analytic") result in a dramatic stability improvement for the box and triangle reductions, respectively. Combining these two improvements ("analytic+perm") reduces the probability of finding points with only few correct digits by three orders of magnitude, and yields a maximum level of instability around 10^2. Finally, switching on the Gram-determinant

---

[12] Note that, in order to avoid possible sources of bias, the quad-precision benchmark $\mathcal{W}_{1-\text{loop}}^{\text{QP,R}}$ is computed with rescaled kinematics as detailed below.
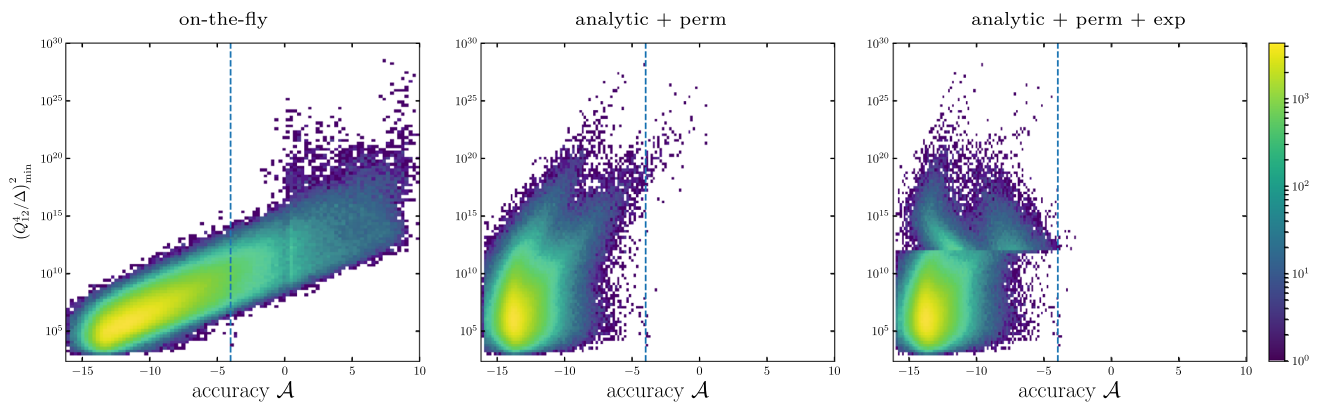
Fig. 15 Correlation between the instability $\mathcal{A}$ of OpenLoops 2 in double precision and the largest $(Q^4/\Delta)^2$ in the event, where Δ is any rank-two Gram determinant and Q^2 is the maximum scale in the corresponding Gram matrix. See (141). Probability densities correspond to 10^6 events. Unstable results without special treatment of Gram determinants (left) are stabilised using the permutation trick (139) for box reduction and analytic expressions for triangle reduction (middle) plus Gram-determinant expansions for $\delta < \delta_{thr}$ (right)

expansions for $\delta < \delta_{thr}$ leads to a further very drastic reduction of the probability of finding results with less than 3–4 correct digits. In this range, we observe an overlap with the tail of the quad-precision distribution. As discussed above, this indicates that OpenLoops 2 in double precision is more stable than the quad-precision benchmarks, and its estimated instability represents only an upper bound. Most likely, the tail of the true OpenLoops 2 stability distribution ends at 10^{-3}.

To gain more insights into the origin of numerical instabilities in the on-the-fly reduction of OpenLoops 2, let us investigate the correlation between the instability (155) and rank-two Gram determinants Δ in the gg $\rightarrow$ t$\bar{\text{t}}$gg sample of Fig. 14. More precisely, in Fig. 15 we consider the minimal value of the dimensionless parameter Δ_{ij}/Q^4_{ij} in the event, where Q^2_{ij} is the largest $|p_i \cdot p_j|$ in the corresponding Gram matrix (see Sect. 5.4.1). As demonstrated by the left plot in Fig. 15, the instability of the entire scattering amplitude features a remarkably strong correlation with rank-two Gram determinants over twenty orders of magnitude. Moreover we observe a quadratic or faster scaling in Q^4/Δ, consistent with the form of the $\gamma^{-2} \sim \Delta^{-2}$ poles in (132). The middle plot shows the combined effect of the permutation trick (139), which avoids the smallest Gram determinant of the event in all reductions with $N \geq 4$ loop denominators, and the attenuation of spurious singularites through analytic expressions for three-point configurations of type (142). In this way the probability of having less than four correct digits is reduced to 0.1 permil. Finally, in the right plot we see that points with less than 3–4 correct digits disappear completely when Gram-determinant expansions are switched on.[13] As one can clearly recognise in the right plot, the threshold for

the activation of Gram-determinant expansions corresponds to $(Q^4/\Delta)^2 \sim \delta_{thr}^{-4} = 10^{12}$.

Finally, in Figs. 16 and 17 we compare the stability of OpenLoops 2 against OpenLoops 1+Collier and OpenLoops 1+CutTools for the 2 $\rightarrow$ 3 and 2 $\rightarrow$ 4 processes in (153). In OpenLoops 2 the stability improvements of Sect. 5.4 are applied throughout. The results of OpenLoops 1+CutTools feature the highest instability tails for all considered processes. The probability of finding less than four correct digits can exceed 10^{-3} in 2 $\rightarrow$ 3 and 10^{-2} in 2 $\rightarrow$ 4 processes, while the fraction of fully unstable points with $\mathcal{A} \geq 0$ can reach 10^{-3} in 2 $\rightarrow$ 4 processes. Switching to OpenLoops 1+Collier we find that, depending on the process, the probability of finding only a few correct digits goes down by one to three orders of magnitude, while in eight samples of 10^6 points we do not find a single result with $\mathcal{A} > 0$.[14]

Using OpenLoops 2 can lead to a further significant stability improvement. This is especially evident for 2 $\rightarrow$ 3 processes, where the stability of the on-the-fly reduction in OpenLoops 2 is remarkably close to the quad-precision benchmarks and even superior than quad precision for the case of t$\bar{\text{t}}$g production. When quad precision is sufficiently accurate to resolve the instabilities of OpenLoops 2 we observe improvements of one–two orders of magnitude with respect to OpenLoops 1+Collier. In the case of 2 $\rightarrow$ 4 processes, depending on the process and the considered number of digits, OpenLoops 2 can perform somewhat better or slightly worse than OpenLoops 1+Collier, like in the case of u$\bar{\text{d}}$ $\rightarrow$ W$^+$ggg or u$\bar{\text{u}}$ $\rightarrow$ W$^+$W$^-$g, respectively. How-

---

[13] Points with unreliable quad-precision benchmarks ($\mathcal{A}_{QP} \sim \mathcal{A}_{DP}$) are not considered in Fig. 15.

[14] As discussed above, due to the insufficient quality of quad-precision benchmarks, instability estimates in the tail of gg $\rightarrow$ t$\bar{\text{t}}$gg are not significant.

Fig. 16 Stability distributions for 2 → 3 processes defined as in Fig. 14. The stability of OPENLOOPS 2 with on-the-fly reduction is compared to OPENLOOPS 1 with COLLIER or CUTTOOLS. The instability of the employed quad precision benchmarks is also shown
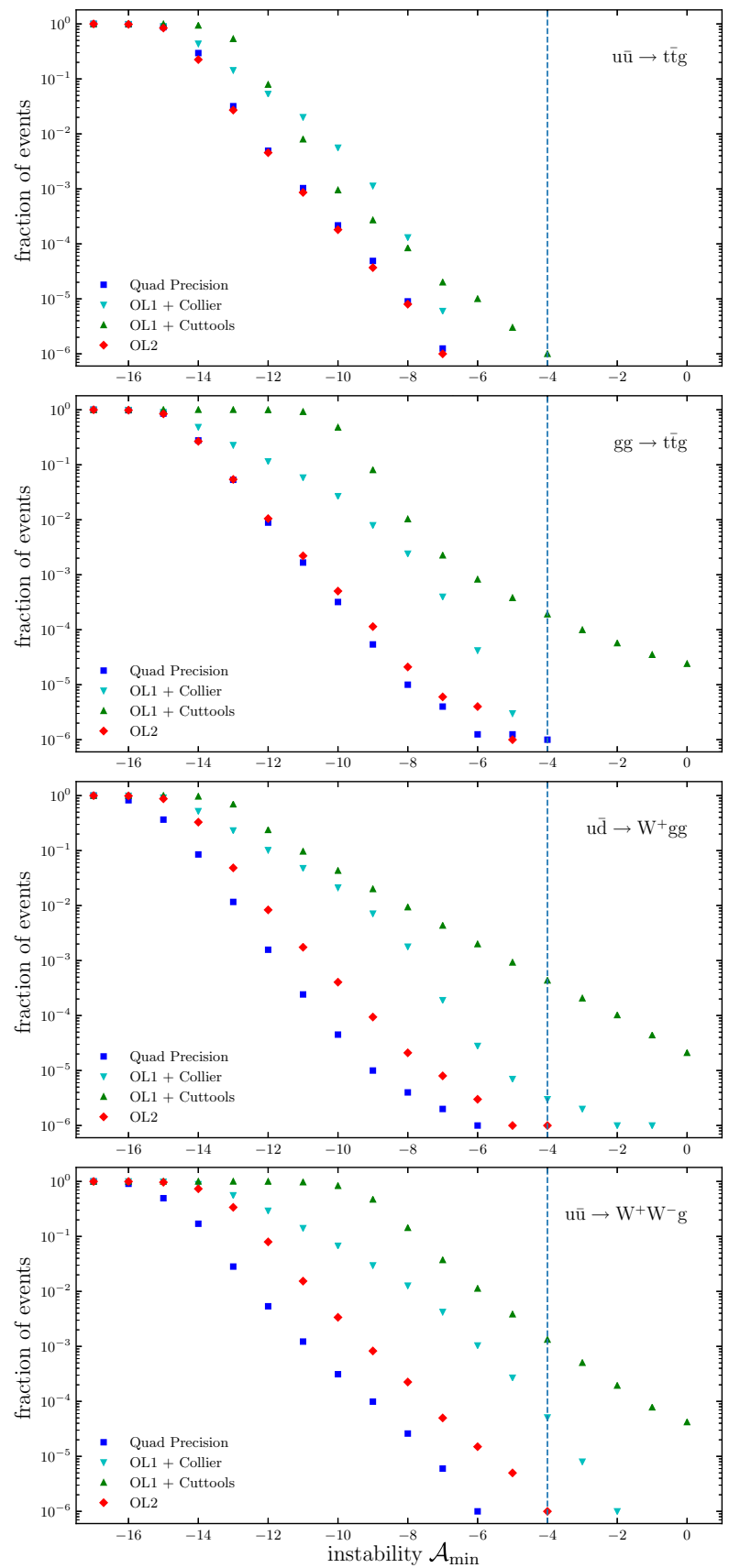
Fig. 17 Stability distributions for 2 → 4 processes defined as in Fig. 16
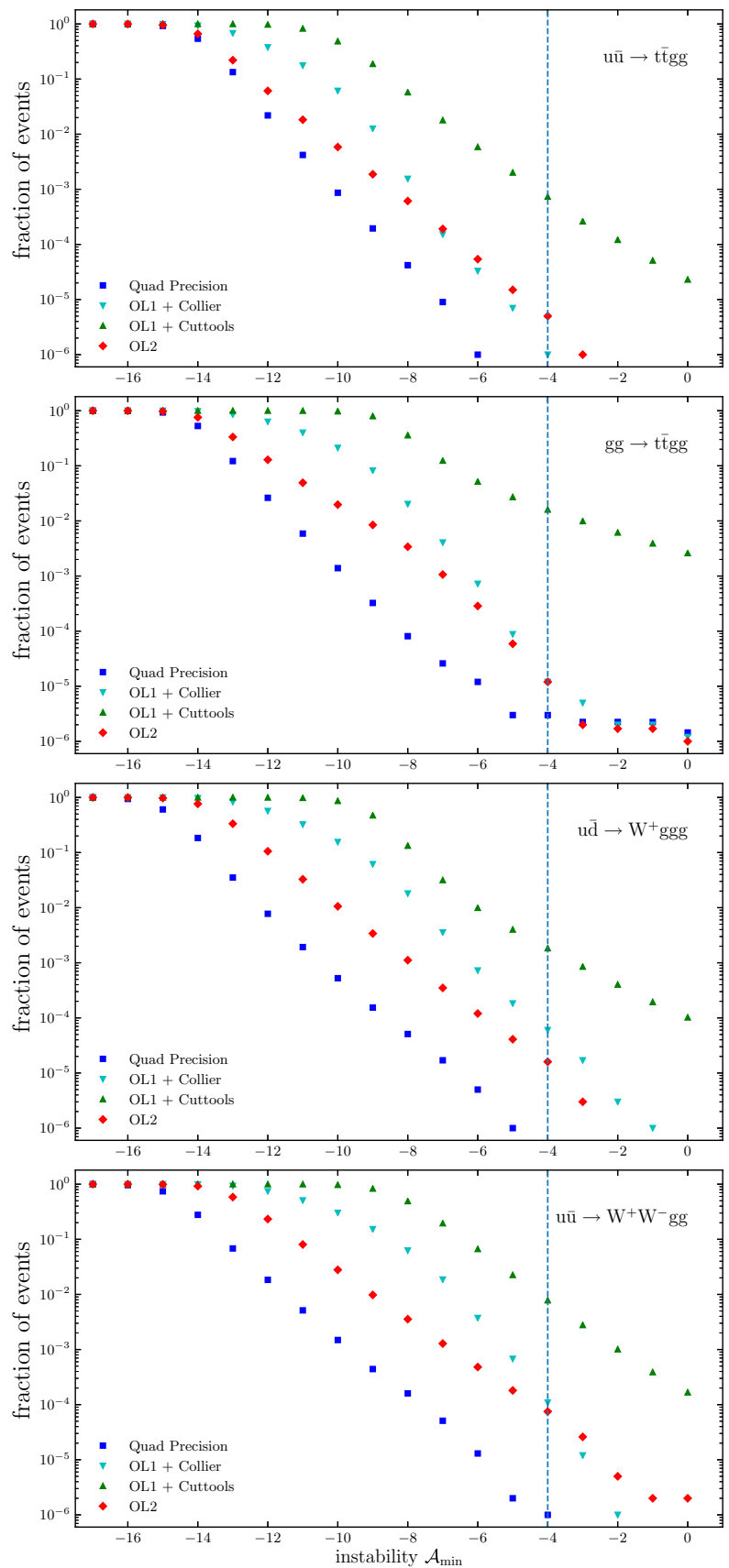
ever, both approaches guarantee excellent numerical stability.

7 Conclusions and outlook

We have presented a new approach for the automated calculation of scattering amplitudes at one loop. This new technique is based on the open-loop approach, where cut-open loop integrands are factored into a product of loop-momentum dependent segments that are combined through recursive tensorial multiplications.

The key idea behind the new method is that various operations, which are typically done at the level of full Feynman diagrams or amplitudes, can be performed *on-the-fly* during the open-loop recursion, i.e. after the multiplication of each loop segment. Since it exploits the factorised structure of open loops in a systematic way, this on-the-fly approach can reduce the complexity of certain operations in a very significant way.

We have first applied the on-the-fly method to helicity summations and to the merging of topologically equivalent open loops, finding speed-up factors of up to two or three as compared to the original open-loop algorithm. Moreover, using the integrand reduction method by del Aguila and Pittau, we have introduced an on-the-fly technique for the reduction of open loops. With this approach, the construction of loop amplitudes and their reduction are interleaved step by step within a single numerical recursion. In this way, objects with tensor rank higher than two are avoided throughout, and the complexity of the calculations is reduced in a very drastic way. The proliferation of pinched subtopologies that emerge from the reduction is avoided by absorbing them on-the-fly into topologically equivalent open loops.

The employed integrand reduction method suffers from severe numerical instabilities that are dominated by kinematic regions with small rank-two Gram determinants Δ and scale like $1/\Delta^2$. In the reduction of N-point objects with $N \geq 4$, we have shown that Δ-instabilities can easily be avoided through appropriate permutations of the loop denominators. In this way we were able to isolate Δ-instabilities in triangle topologies with a particular kinematic configuration and to cure them by means of analytic expansions in Δ. This approach is the first example of an integrand reduction algorithm that is essentially free from Gram-determinant instabilities. The level of stability that is achieved in double precision is competitive with public implementations of OPP reduction in quadruple precision.

The new algorithm is fully automated and validated at NLO QCD and *can* be extended to electroweak interactions. It will become publicly available in the upcoming release of OPENLOOPS 2. Its technical features can be especially beneficial in NLO calculations for challenging multi-particle pro-

cesses. Moreover, in view of its excellent numerical stability, the new algorithm is very attractive for the calculation of real–virtual contributions at NNLO. Finally, the idea of simplifying the construction of loop amplitudes through the factorisation of loop integrands and their on-the-fly reduction may open new interesting perspectives for the automation of two-loop calculations.

Appendix A: Reduction of the remaining tensor integrals

After the on-the-fly reduction described in Sect. 4.1 we are left with a few types of low-rank integrals (84) that need to be reduced to the standard basis of one-loop scalar integrals. The relevant reductions are performed upon taking the trace (76) as described in the following. In this appendix we restrict ourselves to tensor integrals with four-dimensional q^μ terms in the numerator, while contributions with additional $\tilde{q}^2$ terms in the numerator are described in Sect. 4.4.

A.1: Five- and higher-point integrals with $R = 0, 1$

Integrals with $N \geq 5$ loop propagators and a numerator of rank $R \leq 1$,

$$\mathcal{N}(q) = \mathcal{N} + \mathcal{N}_\mu q^\mu, \tag{A.1}$$

can be reduced to a linear combination of scalar boxes,

$$\int d^D \bar{q} \, \frac{\mathcal{N}(q)}{\bar{D}_0 \bar{D}_1 \cdots \bar{D}_{N-1}} = \sum_{i_0 < i_1 < i_2 < i_3}^{N-1} \int d^D \bar{q} \, \frac{d_{i_0 i_1 i_2 i_3}}{\bar{D}_{i_0} \bar{D}_{i_1} \bar{D}_{i_2} \bar{D}_{i_3}}. \tag{A.2}$$

To determine the box coefficients we use the OPP reduction formula [5]

$$d_{i_0 i_1 i_2 i_3} = \frac{1}{2} \left[R_{i_0 i_1 i_2 i_3}(q_0^+) + R_{i_0 i_1 i_2 i_3}(q_0^-) \right], \tag{A.3}$$

where $q_0^\pm$ are the solutions of the quadruple-cut conditions $\bar{D}_{i_0} = \bar{D}_{i_1} = \bar{D}_{i_2} = \bar{D}_{i_3} = 0$, and

$$R_{i_0 i_1 i_2 i_3}(q) = \frac{\mathcal{N}(q)}{\prod\limits_{i \neq i_0, i_1, i_2, i_3}^{N-1} \bar{D}_i} \tag{A.4}$$

are the corresponding residues.

For the special case d_{0123}, where $p_{i_0} = p_0 = 0$, the explicit solutions of the quadruple cut equations read

$$q_0^\pm = x_1 l_1 + x_2 l_2 + x_3^\pm l_4 + x_4^\pm l_4, \tag{A.5}$$

where l_i are the basis momenta defined in (114)–(115). The $x_{1,2}$ coefficients are given by

$$x_{1,2} = \frac{1}{\gamma} \left(d_{2,1} - d_0 \left(1 - \alpha_{2,1} \right) - d_{1,2} \alpha_{2,1} \right), \tag{A.6}$$

with $d_i = m_i^2 - p_i^2$. The terms $\alpha_{1,2}$ and γ are defined in Sect. 5, and the remaining coefficients in (A.5) are given by

$$x_4^\pm = ax_3^\pm = -\frac{(b \pm c_0)}{2}, \tag{A.7}$$

with $a = (p_3 \cdot l_3)/(p_3 \cdot l_4)$ and

$$b = \frac{1}{p_3 \cdot l_4} \left(\frac{d_0 - d_3}{2} + x_1 (p_3 \cdot l_1) + x_2 (p_3 \cdot l_2) \right),$$

$$c_0 = \sqrt{b^2 - a \left(x_1 x_2 - \frac{d_0}{\gamma} \right)}. \tag{A.8}$$

The other coefficients $d_{i_0 i_1 i_2 i_3}$ can be obtained from the above formulas through an obvious reparametrisation of masses and momenta ($i_0 i_1 i_2 i_3 \to 0123$) and a subsequent momentum shift, $p_i^\mu \to p_i^\mu - p_0^\mu$ for $i = 0, 1, 2, 3$, which has to be applied throughout in (A.4)–(A.8).

A.2: Four-point functions with $R = 1$

The reduction of rank-one boxes,

$$\int \mathrm{d}^D \bar{q} \, \frac{q^\mu}{\bar{D}_0 \cdots \bar{D}_3} = \sum_{j=-1}^{3} \int \mathrm{d}^D \bar{q} \, \frac{A_j^\mu}{\bar{D}_0 \cdots \bar{\not{D}}_j \cdots \bar{D}_3}, \tag{A.9}$$

results into a scalar box ($j = -1$) and four scalar triangles ($j = 0, 1, 2, 3$). For the corresponding coefficients we use the reduction formulas [2],

$$A_{1,2}^\mu = \frac{1}{2\gamma} \left[r_{2,1}^\mu - \frac{p_3 \cdot r_{2,1}}{p_3 \cdot l_3} \left(l_3^\mu + \frac{1}{\alpha} l_4^\mu \right) \right],$$

$$A_3^\mu = \frac{1}{4} \frac{1}{p_3 \cdot l_3} \left(l_3^\mu + \frac{1}{\alpha} l_4^\mu \right), \tag{A.10}$$

and

$$A_{-1}^\mu = \sum_{i=1}^{3} f_{i0} A_i^\mu, \quad A_0^\mu = -\sum_{i=1}^{3} A_i^\mu. \tag{A.11}$$

The relevant ingredients, $\gamma, \alpha, l_i^\mu, r_i^\mu, f_{i0}$, are defined in Sect. 5.

A.3: Three-point integrals with $R = 1$

For rank-one triangles we use the covariant decomposition

$$C^\mu(p_1, p_2, m_0, m_1, m_2) = C_\varepsilon \int \mathrm{d}^D \bar{q} \, \frac{q^\mu}{\bar{D}_0 \bar{D}_1 \bar{D}_2}$$

$$= \sum_{i=1}^{2} p_i^\mu C_i(p_1^2, p_2^2, m_0, m_1, m_2), \tag{A.12}$$

where $C_\varepsilon = (2\pi\mu)^{2\varepsilon}/(i\pi^2)$, and we compute the C_i coefficients via Passarino–Veltman reduction [70,71].

A.4: Two-point integrals with $R = 1, 2$

Also for two-point integrals with rank $R = 1, 2$ we perform a Passarino–Veltman reduction [70,71] based on the covariant decomposition

$$B^\mu(p_1, m_0, m_1) = C_\varepsilon \int \mathrm{d}^D \bar{q} \, \frac{q^\mu}{\bar{D}_0 \bar{D}_1}$$

$$= p_1^\mu B_1(p_1^2, m_0, m_1), \tag{A.13}$$

$$B^{\mu\nu}(p_1, m_0, m_1) = C_\varepsilon \int \mathrm{d}^D \bar{q} \, \frac{q^\mu q^\nu}{\bar{D}_0 \bar{D}_1}$$

$$= p_1^\mu p_1^\nu B_{11}(p_1^2, m_0, m_1)$$

$$+ g^{\mu\nu} B_{00}(p_1^2, m_0, m_1). \tag{A.14}$$

Besides the standard reduction formulas for B_1, B_{11}, and B_{00}, for $p^2 = 0$ we have implemented the special cases

$$B_1(0, m_0, m_0) = -\frac{1}{2} B_0(0, m_0, m_0), \tag{A.15}$$

$$B_1(0, m_0, m_1) = \frac{1}{4(m_0^2 - m_1^2)} \left(2A_0(m_1) \right.$$

$$\left. - 2m_0^2 B_0(0, m_0, m_1) - 1 \right), \tag{A.16}$$

$$B_{11}(0, 0, 0) = \frac{1}{3} B_0(0, 0, 0) = 0, \tag{A.17}$$

$$B_{11}(0, m_0, m_0) = -\frac{1}{3} + \frac{A_0(m_0)}{3m_0^2}, \tag{A.18}$$

$$B_{11}(0, m_0, m_1) = \frac{1}{18(m_0 - m_1)^3 (m_0 + m_1)^3}$$

$$\times \left(5m_0^6 - 9m_0^2 m_1^4 + 4m_1^6 + 6m_0^4 A_0(m_0) \right.$$

$$\left. - 6(3m_0^4 - 3m_0^2 m_1^2 + m_1^4) A_0(m_1) \right). \tag{A.19}$$

with the scalar master integrals

$$B_0(p_1^2, m_0, m_1) = C_\varepsilon \int \mathrm{d}^D \bar{q} \, \frac{1}{\bar{D}_0 \bar{D}_1},$$

$$A_0(m_0^2) = C_\varepsilon \int \mathrm{d}^D\bar{q} \, \frac{1}{\bar{D}_0}. \tag{A.20}$$

References

1. R. Britto, F. Cachazo, B. Feng, Generalized unitarity and one-loop amplitudes in $N = 4$ super-Yang-Mills. Nucl. Phys. B **725**, 275–305 (2005). arXiv:hep-th/0412103
2. F. del Aguila, R. Pittau, Recursive numerical calculus of one-loop tensor integrals. JHEP **07**, 017 (2004). arXiv:hep-ph/0404120
3. Z. Bern, L.J. Dixon, D.A. Kosower, Bootstrapping multi-parton loop amplitudes in QCD. Phys. Rev. D **73**, 065013 (2006). arXiv:hep-ph/0507005
4. A. Denner, S. Dittmaier, Reduction schemes for one-loop tensor integrals. Nucl. Phys. B **734**, 62–115 (2006). arXiv:hep-ph/0509141
5. G. Ossola, C.G. Papadopoulos, R. Pittau, Reducing full one-loop amplitudes to scalar integrals at the integrand level. Nucl. Phys. B **763**, 147–169 (2007). arXiv:hep-ph/0609007
6. D. Forde, Direct extraction of one-loop integral coefficients. Phys. Rev. D **75**, 125019 (2007). arXiv:0704.1835
7. W.T. Giele, Z. Kunszt, K. Melnikov, Full one-loop amplitudes from tree amplitudes. JHEP **04**, 049 (2008). arXiv:0801.2237
8. A. van Hameren, Multi-gluon one-loop amplitudes using tensor integrals. JHEP **07**, 088 (2009). arXiv:0905.1005
9. F. Cascioli, P. Maierhöfer, S. Pozzorini, Scattering amplitudes with open loops. Phys. Rev. Lett. **108**, 111601 (2012). arXiv:1111.5206
10. G. Ossola, C.G. Papadopoulos, R. Pittau, CutTools: a program implementing the OPP reduction method to compute one-loop amplitudes. JHEP **03**, 042 (2008). arXiv:0711.3596
11. C.F. Berger, Z. Bern, L.J. Dixon, F. Febres Cordero, D. Forde, H. Ita et al., An automated implementation of on-shell methods for one-loop amplitudes. Phys. Rev. D **78**, 036003 (2008). arXiv:0803.4180
12. A. van Hameren, C.G. Papadopoulos, R. Pittau, Automated one-loop calculations: a proof of concept. JHEP **09**, 106 (2009). arXiv:0903.4665
13. V. Hirschi, R. Frederix, S. Frixione, M.V. Garzelli, F. Maltoni, R. Pittau, Automation of one-loop QCD corrections. JHEP **05**, 044 (2011). arXiv:1103.0621
14. P. Mastrolia, G. Ossola, T. Reiter, F. Tramontano, Scattering amplitudes from unitarity-based reduction algorithm at the integrand-level. JHEP **08**, 080 (2010). arXiv:1006.0710
15. S. Badger, B. Biedermann, P. Uwer, V. Yundin, Numerical evaluation of virtual corrections to multi-jet production in massless QCD. Comput. Phys. Commun. **184**, 1981–1998 (2013). arXiv:1209.0100
16. F. Cascioli, J.M. Lindert, P. Maierhöfer, S. Pozzorini, The OPEN-LOOPS one-loop generator. http://openloops.hepforge.org
17. G. Cullen, GOSAM-2.0: a tool for automated one-loop calculations within the Standard Model and beyond. Eur. Phys. J. C **74**, 3001 (2014). arXiv:1404.7096
18. T. Peraro, Ninja: automated integrand reduction via laurent expansion for one-loop amplitudes. Comput. Phys. Commun. **185**, 2771–2797 (2014). arXiv:1403.1229
19. A. Denner, S. Dittmaier, L. Hofer, Collier: a fortran-based complex one-loop library in extended regularizations. Comput. Phys. Commun. **212**, 220–238 (2017). arXiv:1604.06792
20. S. Actis, A. Denner, L. Hofer, J.-N. Lang, A. Scharf, S. Uccirati, RECOLA: recursive computation of one-loop amplitudes. Comput. Phys. Commun. **214**, 140–173 (2017). arXiv:1605.01090
21. T. Gleisberg, S. Hoeche, F. Krauss, M. Schonherr, S. Schumann, F. Siegert, Event generation with SHERPA 1.1. JHEP **02**, 007 (2009). arXiv:0811.4622
22. S. Alioli, P. Nason, C. Oleari, E. Re, A general framework for implementing NLO calculations in shower Monte Carlo programs: the POWHEG BOX. JHEP **06**, 043 (2010). arXiv:1002.2581
23. G. Bevilacqua, M. Czakon, M.V. Garzelli, A. van Hameren, A. Kardos, C.G. Papadopoulos et al., HELAC-NLO. Comput. Phys. Commun. **184**, 986–997 (2013). arXiv:1110.1499
24. J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer et al., The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations. JHEP **07**, 079 (2014). arXiv:1405.0301
25. T. Sjöstrand, S. Ask, J.R. Christiansen, R. Corke, N. Desai, P. Ilten et al., An introduction to PYTHIA 8.2. Comput. Phys. Commun. **191**, 159–177 (2015). arXiv:1410.3012
26. C. Weiss, B. Chokoufe Nejad, W. Kilian, J. Reuter, Automated NLO QCD Corrections with WHIZARD, PoS **EPS-HEP2015** (2015) 466, arXiv:1510.02666
27. J. Bellm, Herwig 7.0/Herwig++ 3.0 release note. Eur. Phys. J. C **76**, 196 (2016). arXiv:1512.01178
28. Z. Bern, L.J. Dixon, F. Febres Cordero, S. Höche, H. Ita, D.A. Kosower et al., Next-to-leading order $W + 5$-jet production at the LHC. Phys. Rev. D **88**, 014025 (2013). arXiv:1304.1253
29. S. Badger, B. Biedermann, P. Uwer, V. Yundin, Next-to-leading order QCD corrections to five jet production at the LHC. Phys. Rev. D **89**, 034019 (2014). arXiv:1309.6585
30. G. Bevilacqua, H.B. Hartanto, M. Kraus, M. Worek, Top quark pair production in association with a jet with next-to-leading-order QCD off-shell effects at the large hadron collider. Phys. Rev. Lett. **116**, 052003 (2016). arXiv:1509.09242
31. S. Höche, P. Maierhöfer, N. Moretti, S. Pozzorini, F. Siegert, Next-to-leading order QCD predictions for top-quark pair production with up to three jets. Eur. Phys. J. C **77**, 145 (2017). arXiv:1607.06934
32. A. Denner, J.-N. Lang, M. Pellen, S. Uccirati, Higgs production in association with off-shell top-antitop pairs at NLO EW and QCD at the LHC. JHEP **02**, 053 (2017). arXiv:1612.07138
33. F. Cascioli, S. Höche, F. Krauss, P. Maierhöfer, S. Pozzorini et al., Precise Higgs-background predictions: merging NLO QCD and squared quark-loop corrections to four-lepton + 0,1 jet production. JHEP **1401**, 046 (2014). arXiv:1309.0500
34. F. Cascioli, P. Maierhöfer, N. Moretti, S. Pozzorini, F. Siegert, NLO matching for $t\bar{t}b\bar{b}$ production with massive b-quarks. Phys. Lett. B **734**, 210–214 (2014). arXiv:1309.5912
35. F. Cascioli, S. Kallweit, P. Maierhöfer, S. Pozzorini, A unified NLO description of top-pair and associated Wt production. Eur. Phys. J. C **74**, 2783 (2014). arXiv:1312.0546
36. S. Höche, F. Krauss, P. Maierhöfer, S. Pozzorini, M. Schönherr et al., Next-to-leading order QCD predictions for top-quark pair production with up to two jets merged with a parton shower. arXiv:1402.6293
37. S. Höche, F. Krauss, S. Pozzorini, M. Schönherr, J. Thompson et al., Triple vector boson production through Higgs-Strahlung with NLO multijet merging. Phys. Rev. D **89**, 093015 (2014). arXiv:1403.7516
38. T. Ježo, J.M. Lindert, P. Nason, C. Oleari, S. Pozzorini, An NLO+PS generator for $t\bar{t}$ and Wt production and decay including non-resonant and interference effects. Eur. Phys. J. C **76**, 691 (2016). arXiv:1607.04538
39. B. Chokoufé Nejad, W. Kilian, J.M. Lindert, S. Pozzorini, J. Reuter, C. Weiss, NLO QCD predictions for off-shell $t\bar{t}$ and $t\bar{t}H$ production and decay at a linear collider. JHEP **12**, 075 (2016). arXiv:1609.03390
40. S. Kallweit, J.M. Lindert, P. Maierhöfer, S. Pozzorini, M. Schönherr, NLO electroweak automation and precise predictions for W+multijet production at the LHC. JHEP **04**, 012 (2015). arXiv:1412.5157

41. S. Kallweit, J.M. Lindert, P. Maierhofer, S. Pozzorini, M. Schönherr, NLO QCD+EW predictions for V + jets including off-shell vector-boson decays and multijet merging. JHEP **04**, 021 (2016). arXiv:1511.08692

42. S. Kallweit, J.M. Lindert, S. Pozzorini, M. Schonherr, NLO QCD+EW predictions for $2\ell2\nu$ diboson signatures at the LHC. arXiv:1705.00598

43. J.M. Lindert et al., Precise predictions for V+jets dark matter backgrounds. arXiv:1705.04664

44. F. Granata, J.M. Lindert, C. Oleari, S. Pozzorini, NLO QCD+EW predictions for HV and HV +jet production including parton-shower effects. JHEP **09**, 012 (2017). arXiv:1706.03522

45. M. Wiesemann, M. Grazzini, S. Kallweit, D. Rathlev, Fully-differential NNLO predictions for vector-boson pair production with MATRIX. PoS **LL2016**, 072 (2016)

46. M. Grazzini, S. Kallweit, D. Rathlev, A. Torre, $Z\gamma$ production at hadron colliders in NNLO QCD. Phys. Lett. B **731**, 204–207 (2014). arXiv:1309.7000

47. F. Cascioli, T. Gehrmann, M. Grazzini, S. Kallweit, P. Maierhöfer et al., ZZ production at hadron colliders in NNLO QCD. Phys. Lett. B **735**, 311–313 (2014). arXiv:1405.2219

48. T. Gehrmann, M. Grazzini, S. Kallweit, P. Maierhöfer, A. von Manteuffel et al., W^+W^- production at hadron colliders in next to next to leading order QCD. Phys. Rev. Lett. **113**, 212001 (2014). arXiv:1408.5243

49. M. Grazzini, S. Kallweit, D. Rathlev, $W\gamma$ and $Z\gamma$ production at the LHC in NNLO QCD. JHEP **07**, 085 (2015). arXiv:1504.01330

50. M. Grazzini, S. Kallweit, D. Rathlev, M. Wiesemann, Transverse-momentum resummation for vector-boson pair production at NNLL+NNLO. JHEP **08**, 154 (2015). arXiv:1507.02565

51. M. Grazzini, S. Kallweit, D. Rathlev, ZZ production at the LHC: fiducial cross sections and distributions in NNLO QCD. Phys. Lett. B **750**, 407–410 (2015). arXiv:1507.06257

52. M. Grazzini, S. Kallweit, D. Rathlev, M. Wiesemann, $W^\pm Z$ production at hadron colliders in NNLO QCD. arXiv:1604.08576

53. M. Grazzini, S. Kallweit, S. Pozzorini, D. Rathlev, M. Wiesemann, W^+W^- production at the LHC: fiducial cross sections and distributions in NNLO QCD. arXiv:1605.02716

54. D. de Florian, M. Grazzini, C. Hanga, S. Kallweit, J.M. Lindert, P. Maierhöfer et al., Differential Higgs boson pair production at next-to-next-to-leading order in QCD. JHEP **09**, 151 (2016). arXiv:1606.09519

55. M. Grazzini, S. Kallweit, D. Rathlev, M. Wiesemann, $W^\pm Z$ production at the LHC: fiducial cross sections and distributions in NNLO QCD. arXiv:1703.09065

56. A. Bredenstein, A. Denner, S. Dittmaier, S. Pozzorini, NLO QCD corrections to top anti-top bottom anti-bottom production at the LHC: 2. full hadronic results. JHEP **03**, 021 (2010). arXiv:1001.4006

57. J.A.M. Vermaseren, Axodraw. Comput. Phys. Commun. **83**, 45–58 (1994)

58. F.A. Berends, W.T. Giele, Recursive calculations for processes with n gluons. Nucl. Phys. B **306**, 759–808 (1988)

59. G. Ossola, C.G. Papadopoulos, R. Pittau, On the rational terms of the one-loop amplitudes. JHEP **05**, 004 (2008). arXiv:0802.1876

60. P. Draggiotis, M.V. Garzelli, C.G. Papadopoulos, R. Pittau, Feynman rules for the rational part of the QCD 1-loop amplitudes. JHEP **04**, 072 (2009). arXiv:0903.0356

61. M.V. Garzelli, I. Malamos, R. Pittau, Feynman rules for the rational part of the electroweak 1-loop amplitudes. JHEP **01**, 040 (2010). arXiv:0910.3130

62. R. Pittau, Primary Feynman rules to calculate the epsilon-dimensional integrand of any 1-loop amplitude. JHEP **02**, 029 (2012). arXiv:1111.4965

63. A. Denner, S. Dittmaier, Reduction of one-loop tensor 5-point integrals. Nucl. Phys. B **658**, 175–202 (2003)

64. A. Denner, S. Dittmaier, Scalar one-loop 4-point integrals. Nucl. Phys. B **844**, 199–242 (2011)

65. A. van Hameren, OneLOop: for the evaluation of one-loop scalar functions. Comput. Phys. Commun. **182**, 2427–2438 (2011). arXiv:1007.4716

66. A. Denner, S. Dittmaier, L. Hofer, COLLIER—a fortran-library for one-loop integrals. arXiv:1407.0087

67. R.N. Lee, LiteRed 1.4: a powerful tool for reduction of multiloop integrals. J. Phys. Conf. Ser. **523**, 012059 (2014). arXiv:1310.1145

68. T. Hahn, Generating Feynman diagrams and amplitudes with FeynArts 3. Comput. Phys. Commun. **140**, 418–431 (2001). hep-ph/0012260

69. S. Badger, B. Biedermann, P. Uwer, NGluon: a package to calculate one-loop multi-gluon amplitudes. Comput. Phys. Commun. **182**, 1674–1692 (2011). arXiv:1011.2900

70. G. Passarino, M.J.G. Veltman, One loop corrections for e+ e- annihilation into mu+ mu- in the Weinberg model. Nucl. Phys. B **160**, 151–207 (1979)

71. A. Denner, Techniques for calculation of electroweak radiative corrections at the one loop level and results for W physics at LEP-200. Fortsch. Phys. **41**, 307–420 (1993). arXiv:0709.1075