Computer Physics Communications 320 (2026) 109933

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

Computer Programs in Physics

ggxy: A flexible library to compute gluon-induced cross sections

Joshua Davies [a], Kay Schönwald [b], Matthias Steinhauser [c], Daniel Stremmer [c,*]

[a] Department of Mathematical Sciences, University of Liverpool, Liverpool, L69 3BX, UK
[b] Physik-Institut, Universität Zürich, Winterthurerstrasse 190, 8057, Zürich, Switzerland
[c] Institut für Theoretische Teilchenphysik, Karlsruhe Institute of Technology (KIT), Wolfgang-Gaede Straße 1, 76131, Karlsruhe, Germany

ARTICLE INFO

Editor: Prof. Z. Was

Keywords:
Gluon-induced NLO processes
Partonic and hadronic cross sections
Analytic expansions.

ABSTRACT

We present the library ggxy, written in C++, which can be used to compute partonic and hadronic cross sections for gluon-induced processes with at least one closed heavy quark loop. It is based on analytic ingredients which avoids, to a large extent, expensive numerical integration. This results in significantly shorter run-times than other similar tools. Modifying input parameters, changing the renormalization scheme and varying renormalization and factorization scales is straightforward. In Version 1 of ggxy we implement all routines which are needed to compute partonic and hadronic cross sections for Higgs boson pair production up to next-to-leading order in QCD. We provide flexible interfaces and allow the user to interact with the built-in amplitudes at various levels.

PROGRAM SUMMARY

Program title: ggxy
Developer's repository link: https://gitlab.com/ggxy/ggxy-release
Licensing provisions: GNU General Public License Version 3
Programming language: C++ and Fortran
External routines/libraries used: avhlib, boost, Collier, CuTtools, eigen, LHAPDF, lievaluate, OneLOop, Recola, CRunDec
Nature of problem: The computation of partonic and hadronic cross sections for gluon-induced processes. In Version 1, the Higgs boson pair production process is implemented at next-to-leading order in Quantum Chromodynamics.
Solution method: For the virtual corrections, deep expansions around the forward and high energy limit are used.
Restrictions: The run-times depend crucially on the requested precision. Results at the per-mille level can be obtained in about 30 minutes using a single core on a AMD Ryzen Threadripper PRO 3955WX processor.
References and Links: are provided in the paper

1. Introduction

At the Large Hadron Collider (LHC) at CERN, gluon-induced processes have a comparatively large cross section and are thus important for various phenomenological analyses, in particular in the context of the Higgs boson (see, e.g., Ref. [1]). This is also true for Higgs boson pair production, a process which already receives a lot of attention from the experimental groups. It will be further scrutinised during the high-luminosity phase of the LHC, since it is the only way to access the self-interaction of the Higgs boson directly.

Higgs boson pair production is one of the first gluon-initiated loop-induced $2 \to 2$ processes which has been computed to next-to-leading order (NLO) in QCD without relying on any approximation [2–4]. This was an important milestone since the NLO corrections amount to almost 100 % of the LO and are thus very important. In addition, there is a large theoretical uncertainty due to the dependence on the renormalization scheme of the top quark which comes with a change in the mass value [5,6].

The first calculation with exact m_t dependence at NLO from Refs. [2,3] suffered from large run-times. This has been circumvented by generating an interpolation grid for the virtual amplitude [7]. However, this approach has the disadvantage that the top quark and Higgs boson masses have fixed values. As a consequence, studies using the $\overline{\text{MS}}$ definition of the top quark mass with an energy-dependent renormalization scale are prohibitively expensive. In Ref. [4] such a study has been performed, however no (detailed) timings are provided in the paper.

A possibility to reduce or even eliminate the bottlenecks related to the run-time and the lack of flexibility to modify numerical input values

* Corresponding author.
E-mail addresses: J.O.Davies@liverpool.ac.uk (J. Davies), kay.schoenwald@physik.uzh.ch (K. Schönwald), matthias.steinhauser@kit.edu (M. Steinhauser), daniel.stremmer@kit.edu (D. Stremmer).

https://doi.org/10.1016/j.cpc.2025.109933
Received 13 June 2025; Received in revised form 23 September 2025; Accepted 1 November 2025
Available online 14 November 2025

is to incorporate analytic results from expansions in various phase-space regions. In a first approach in this direction, the analytic results obtained in the high-energy region were combined with the purely numerical results of Ref. [7], which are then only needed in a restricted region of phase space [8]. In a later work the expansion around small transverse Higgs boson momenta has been combined with the high-energy expansion which completely avoids expensive numerical calculations [9,10].

In this work we implement the results of Ref. [10] in a fast and flexible C++ library: ggxy. We supplement these results by our implementation of the real radiation contribution and the possibility to compute differential and total hadronic cross sections. Higgs boson pair production serves as a sample process. The structure of ggxy is such that it can easily be extended to other processes in the future.

The remainder of the paper is structured as follows: In Sections 2 and 3 we comment on the virtual and real corrections to $gg \to HH$, respectively, and put special emphasis on the implementation in ggxy. Section 4 provides details on the installation and the structure of ggxy. We describe the dependencies and provide a manual for using the functions implemented in ggxy. Section 5 contains various examples which demonstrate the various ways to use ggxy. The user is invited to adapt the examples to their own purpose. We conclude in Section 6.

2. Virtual corrections

The virtual corrections to Higgs boson pair production have the same kinematics as the Born process. They are conveniently parameterized in terms of two form factors reflecting the two possible tensor structures for $gg \to HH$. We follow the notation introduced in [10] and refer to this paper for details. For convenience we summarize in the following only those formulae which are relevant for the implementation in ggxy.

We introduce the perturbative expansion of the form factors F_1 and F_2 as

$$F = F^{(0)} + \left(\frac{\alpha_s(\mu_r)}{\pi}\right) F^{(1)} + \left(\frac{\alpha_s(\mu_r)}{\pi}\right)^2 F^{(2)} + \cdots, \tag{1}$$

and decompose them into "triangle" and "box" form factors ($k = 0, 1, \ldots$)

$$\begin{aligned} F_1^{(k)} &= \frac{3m_H^2}{s - m_H^2} F_{\text{tri}}^{(k)} + F_{\text{box1}}^{(k)} + F_{\text{dt1}}^{(k)}, \\ F_2^{(k)} &= F_{\text{box2}}^{(k)} + F_{\text{dt2}}^{(k)}. \end{aligned} \tag{2}$$

$F_{\text{dt1}}^{(k)}$ and $F_{\text{dt2}}^{(k)}$ denote the contribution from one-particle reducible double-triangle diagrams, see e.g. Fig. 1(f) of Ref. [8] (note that $F_{\text{dt1}}^{(0)} = F_{\text{dt2}}^{(0)} = 0$). Analytic results for the leading-order form factors are available from Refs. [11,12]. The two-loop triangle form factors have been computed in Refs. [13–15], but we refrain from implementing them in ggxy since the analytic expansions provide excellent approximations. The exact results for the double-triangle contribution can be found in Ref. [16], and are implemented in ggxy.

In the pre-factor of $F_{\text{tri}}^{(k)}$ it is possible to identify the trilinear self-coupling of the Higgs boson, λ_3, present in the Higgs potential

$$V(H) = \frac{m_H^2}{2} H^2 + \lambda_3 v H^3 + \frac{\lambda_4}{4} H^4, \tag{3}$$

where v is the vacuum expectation value and H is the Higgs boson field. In the Standard Model we have $\lambda_3^{\text{SM}} = \lambda_4^{\text{SM}} = m_H^2/(2v^2)$. Deviations from this value are often parametrized in the so-called κ framework by introducing $\kappa_\lambda = \lambda_3/\lambda_3^{\text{SM}}$. In ggxy it is possible to choose a value for $\kappa_\lambda \neq 1$. The current experimental bounds for κ_λ from ATLAS and CMS are $-1.2 < \kappa_\lambda < 7.2$ [17] and $-1.39 < \kappa_\lambda < 7.02$ [18], respectively.

For the virtual corrections we introduce the Mandelstam variables s, t and u which are defined in the usual way with $s + t + u = 2m_H^2$, and we also need the transverse momentum of the Higgs bosons which is given by $p_T^2 = (tu - m_H^4)/s$.

For the virtual corrections we implement the (semi-)analytic expansions around the forward and the high-energy limits as obtained in Ref. [10], where for the form factors F_1 and F_2 expansions in t up to order t^5 and in m_t up to order m_t^{120} have been computed.[1] In both kinematic limits we expand in m_H up to quartic order. Although this is the limitation in precision of our approach, it is more than sufficient for all practical purposes. For higher values of the partonic centre-of-mass energy $\sqrt{s}$ and of p_T the uncertainty is (far) below the percent level; close to the production threshold $\sqrt{s} \approx 250$ GeV it can be larger (in particular for F_2), however, there the numerical values of the form factors are small. Overall it has been shown [10] that the expansions approximate the (exact) numerical results with high precision.

In ggxy ultraviolet renormalized and infrared (IR) subtracted form factors are implemented. For the renormalization we apply the $\overline{\text{MS}}$ scheme for α_s and renormalize the external gluon field on-shell. For the top quark mass there is the option to choose the on-shell (M_t) or $\overline{\text{MS}}$ scheme ($\overline{m}_t(\mu_t)$), where μ_t is the corresponding renormalization scale. Our final results are expressed in term of $\alpha_s^{(5)}(\mu_r)$.

The subsequent subtraction of the remaining IR poles in ϵ leads to finite form factors. Since there are different schemes concerning the subtraction we provide explicit expressions for our implementation. We obtain the finite form factors via

$$F^{(1),\text{fin}} = F^{(1)} - \frac{1}{2} I_g^{(1)} F^{(0)}, \tag{4}$$

where the quantities on the right-hand side are ultraviolet-renormalized and $I_g^{(1)}$ is given by [22]

$$I_g^{(1)} = -\left(\frac{\mu_r^2}{-s - i\delta}\right)^\epsilon \frac{e^{\epsilon\gamma_E}}{\Gamma(1-\epsilon)} \frac{1}{\epsilon^2}\left[C_A + 2\epsilon\beta_0\right], \tag{5}$$

with

$$\beta_0 = \frac{1}{4}\left(\frac{11}{3} C_A - \frac{4}{3} T n_l\right). \tag{6}$$

In addition to the form factors, in ggxy we also implement the virtual finite NLO corrections in the form (see, e.g., Refs. [23,24])

$$\widetilde{\mathcal{V}}_{\text{fin}} = \frac{\alpha_s^2(\mu_r)}{16\pi^2} \frac{G_F^2 s^2}{64}\left[C + 2\left(\tilde{F}_1^{(0)*}\tilde{F}_1^{(1)} + \tilde{F}_2^{(0)*}\tilde{F}_2^{(1)} + \tilde{F}_1^{(0)}\tilde{F}_1^{(1)*} + \tilde{F}_2^{(0)}\tilde{F}_2^{(1)*}\right)\right], \tag{7}$$

where $\tilde{F}^{(i)} = F^{\text{fin},(i)}(\mu_r^2 = -s)$ and

$$C = \left(\left|\tilde{F}_1^{(0)}\right|^2 + \left|\tilde{F}_2^{(0)}\right|^2\right)\left(C_A \pi^2 - C_A \log^2\frac{\mu_r^2}{s}\right), \tag{8}$$

with μ_r being the renormalization scale which is also present at the hadronic level. Here, α_s corresponds to the five-flavour strong coupling constant. Furthermore, we introduce

$$\mathcal{V}_{\text{fin}} = \frac{\widetilde{\mathcal{V}}_{\text{fin}}}{\alpha_s^2(\mu_r)}. \tag{9}$$

Details on the implementation can be found in Section 5.2.

In the remaining part of this section we comment on the implementation of the analytic expressions. After expansion in m_H^2, the expansion of the form factors in t is a simple Taylor expansion with coefficients depending on s/m_t^2. We express the amplitude in terms of 48 master integrals which we compute with the help of the "expand and match" method [25–27]. This provides results for each ϵ coefficient of each master integral as a power-log expansion around properly chosen values for s/m_t^2. The combination of these expansions provides results with a precision of 10 or more digits over the whole $\sqrt{s}$ range. In practice we parametrize the ϵ expansion of each master integral in terms of coefficients, which depend on s/m_t^2. We insert the generic expansion into the amplitude and convert this expression into C++. For

[1] Results to lower expansion depths in the high-energy expansion have been obtained in Refs. [19,20]. In the forward limit three expansion terms have been computed in Ref. [21] in the context of the expansion in the transverse momentum of the Higgs boson, p_T.

the numerical evaluation we provide routines which implement the results of the "expand and match" results for the coefficients of the master integrals. In order to make the routine more efficient, we do not implement all of the series expansions provided by the method, but approximate the results via Chebychev polynomials, see e.g. Ref. [28]. In practice we use Chebychev approximations with 100 terms in the regions $50 \le s/m_t^2 < 500$, $25 \le s/m_t^2 < 50$, $8 \le s/m_t^2 < 25$, $4.2 \le s/m_t^2 < 8$, $2 \le s/m_t^2 < 3.8$, $0.1 \le s/m_t^2 < 2$. To cover the regions around the singular points $s/m_t^2 = \{0, 4, \infty\}$ we include generalized series expansions with $\{10, 10, 7\}$ expansion terms, respectively. The implementation of the master integrals rather than the approximation of the amplitude itself provides the possibility to re-use the implementation also for other scattering processes in the future. For numerical stability in the limit $s/m_t^2 \to \infty$, however, it was necessary to insert the expansions of the master integrals and implement the expanded amplitudes in C++. We switch to these explicit expansions for $s/m_t^2 > 500$.

In the high-energy limit one encounters a complicated asymptotic expansion for $|s|, |t|, |u| \gg m_t^2$ which is discussed in detail in Refs. [19,20,29]. The efficient use of the differential equations for the master integrals enables us to compute more than 100 expansion terms in the limit $m_t \to 0$. For the numerical evaluation of the form factors based on these expansions, for $p_T \lesssim 500$ GeV, it is necessary to construct Padé approximations [30]. In Ref. [10] it has been shown that a deep expansion in m_t leads to precise results even close to the top–anti-top quark threshold if $p_T \gtrsim 150$ GeV. We implement the Padé approximation following Ref. [28]. To be self-contained, we provide some details here:

In the high-energy expansion, our results are given as series expansions in the variable $x = m_t^2/s$:

$$F(x) = \sum_{k=0}^{N+M} c_k x^k,$$

where the c_k can still depend on $\log(x)$. We wish to find the Padé approximant

$$R(x) = \frac{\sum_{i=0}^{M} a_i x^i}{1 + \sum_{j=1}^{N} b_j x^j},$$

which satisfies

$$\frac{\mathrm{d}^i}{\mathrm{d}x^i} F(x)\bigg|_{x=0} = \frac{\mathrm{d}^i}{\mathrm{d}x^i} R(x)\bigg|_{x=0} \tag{10}$$

with $0 \le i < N + M$. This problem is equivalent to finding the solution to the following system of equations:

$$\sum_{m=1}^{N} b_m C_{N-m+k} = -c_{N+k}, \qquad k = 1, \ldots, N, \tag{11}$$

$$\sum_{m=0}^{k} b_m c_{k-m} = a_k, \qquad k = 1, \ldots, N. \tag{12}$$

In order to solve this equation we use the QR decomposition with the help of Householder transformations implemented in the Eigen library [31]. We follow Ref. [30] and compute several Padé approximations and combine them in a weighted way to obtain a central value and an uncertainty estimate of our procedure. Numerical instabilities can show up when we compute Padé approximations for rather small values of $p_T \lesssim 200$ GeV, where the expansion coefficients become very large. The instabilities can be identified due to a large Padé uncertainty on the approximation in a region where our procedure should still provide accurate results. Whenever we find these large uncertainty estimates we rerun the Padé procedure in quadruple precision. Although we have over 100 expansion coefficients at hand, we implement only the first 48 terms in ggxy; higher expansion terms quickly become numerically unstable in double precision. We observe that this is sufficient to obtain precise numerical values down to $p_T \sim 175$ GeV, where we can switch to the small-t expansion. In practice, we interpolate between the small-t and high-energy expansions in the region $200\,\text{GeV} < p_T < 220$ GeV.

3. Real radiation

The proper treatment of IR divergences at NLO in QCD is very well studied. However, in order to motivate the implementations in ggxy we briefly repeat the main features.

The partonic NLO QCD cross section can be written schematically as

$$\sigma_{ab}^{\text{NLO}} = \int_{n+1} \mathrm{d}\sigma_{ab}^{\text{R}} + \int_n \left[\mathrm{d}\sigma_{ab}^{\text{LO+V}} + \mathrm{d}\sigma_{ab}^{\text{C}}\right], \tag{13}$$

where the subscript on the integrals indicates the dimension of the phase-space integration, σ_{ab}^{R} is the partonic process with an additional parton, $\mathrm{d}\sigma_{ab}^{\text{LO+V}}$ is the combination of the LO cross section and the virtual NLO corrections, and $\mathrm{d}\sigma_{ab}^{\text{C}}$ is the counterterm coming from the redefinition of PDFs due to the absorption of initial-state collinear singularities. While the partonic NLO cross section is IR finite, $\mathrm{d}\sigma_{ab}^{\text{V}}$ and $\mathrm{d}\sigma_{ab}^{\text{C}}$ contain explicit IR ϵ poles and further IR singularities arise in the first term after the phase-space integration of the unresolved parton in $d = 4 - 2\epsilon$ dimensions. In order to make a phase-space integration in $d = 4$ dimensions possible, we use the Catani-Seymour dipole subtraction scheme [32], where the partonic cross section is rewritten as

$$\sigma_{ab}^{\text{NLO}} = \int_{n+1} \left[\sigma_{ab}^{\text{R}} - \sigma_{ab}^{\text{A}}\right] + \int_n \left[\mathrm{d}\sigma_{ab}^{\text{LO+V}} + \mathrm{d}\sigma_{ab}^{\text{C}} + \int_1 \mathrm{d}\sigma_{ab}^{\text{A}}\right], \tag{14}$$

where a new subtraction term $\mathrm{d}\sigma_{ab}^{\text{A}}$ is introduced which mimics σ_{ab}^{R} in all IR limits and makes the integrand of the first term IR finite. The same term is added back to the second integrand and the phase-space integration over the unresolved parton leads to explicit IR ϵ poles which cancel exactly those from $\mathrm{d}\sigma_{ab}^{\text{LO+V}}$ and $\mathrm{d}\sigma_{ab}^{\text{C}}$. In summary, both terms are separately IR finite and can be safely integrated over the phase space in $d = 4$ dimensions.

In the Catani-Seymour subtraction scheme the last two terms in Eq. (14) are rewritten in terms of the so-called integrated dipole operators as

$$\int_n \left[\mathrm{d}\sigma_{ab}^{\text{C}} + \int_1 \mathrm{d}\sigma_{ab}^{\text{A}}\right] = \int_n \left[\mathbf{I}(\mu_r^2) \otimes \mathrm{d}\sigma_{ab}^{\text{LO}} + \sum_{a',b'} \int_0^1 \mathrm{d}x\, \mathbf{KP}_{ab,a'b'}(x, \mu_f^2) \otimes \mathrm{d}\sigma_{a'b'}^{\text{LO}}(x)\right], \tag{15}$$

where the symbol $\otimes$ denotes colour correlations and the dependence on the IR and factorization scales is made explicit. The explicit definitions of the operators can be found in Ref. [32]. The $\mathbf{KP}_{ab,a'b'}$ operator is further convoluted with the LO cross section and contains non-diagonal terms in flavour space with respect to the initial-state partons. Since at NLO we can have at most one $1 \to 2$ splitting, all terms in the $\mathbf{KP}_{ab,a'b'}$ are proportional to either $\delta_{aa'}$ or $\delta_{bb'}$.

For the $gg \to HH$ process we have implemented the two initial-state dipoles with initial-state spectators corresponding to the $g \to gg$ and $g \to q\bar{q}$ splittings. In addition, we have implemented the phase-space restriction on the subtraction terms [33,34] parametrized by α_{CS}. This parameter can be used to cross-check the calculation since the sum of all parts is independent of this artificial parameter, where the case $\alpha_{\text{CS}} = 1$ corresponds to no phase-space restriction. In ggxy we set $\alpha_{\text{CS}} = 0.1$. Following Ref. [35], the parameter α_{CS} is also used to parametrize a technical cut to avoid numerical instabilities due to large cancellations between the real emission contribution and the subtraction terms. We have cross-checked our implementation of the Catani-Seymour subtraction scheme against the program Helac-Dipoles [35] for single phase-space points and after phase-space integration. The finite part of the real corrections can be safely combined with the finite virtual correction $\widetilde{\mathcal{V}}_{\text{fin}}$, defined in Eq. (7), where the latter contribution has to be multiplied by a factor of $\frac{\alpha_s}{2\pi}$.

For the real corrections to Higgs boson pair production, which have already been calculated in Refs. [2–4,36–38], the following subprocesses have to be taken into account

$$\begin{aligned} gg &\to HHg, & gq/qg &\to HHq, \\ q\bar{q}/\bar{q}q &\to HHg, & g\bar{q}/\bar{q}g &\to HH\bar{q}. \end{aligned} \tag{16}$$

The corresponding one-loop amplitudes, as well as the spin-correlated squared matrix element of $gg \to HH$ required for the subtraction terms, are calculated with `Recola` [39,40] where the one-loop matrix elements are written in terms of tensor coefficients $c^{(t)}_{\mu_1\dots\mu_{r_t}}$ and tensor integrals $T^{\mu_1\dots\mu_{r_t}}_{(t)}$ as

$$\mathcal{M}_{1-\text{loop}} = \sum_t c^{(t)}_{\mu_1\dots\mu_{r_t}} T^{\mu_1\dots\mu_{r_t}}_{(t)}. \tag{17}$$

The tensor integrals are defined as

$$T^{\mu_1\dots\mu_{r_t}}_{(t)} = \frac{(2\pi\mu)^{4-D}}{i\pi^2} \int \mathrm{d}^D q \frac{q^{\mu_1}\dots q^{\mu_{r_t}}}{D^{(t)}_0 \dots D^{(t)}_{k_t}}, \tag{18}$$

where k_t is the number of propagators and r_t the rank of the tensor integral. The denominators are given by

$$D^{(t)}_i = \left(q + p^{(t)}_i\right)^2 - \left(m^{(t)}_i\right)^2, \tag{19}$$

with $p^{(t)}_0 = 0$.

The tensor coefficients are calculated in a recursive approach in `Recola` and the tensor integrals are calculated with `Collier` [41], which performs an uncertainty estimation on the precision of the tensor integrals. If we encounter a phase-space point that leads to tensor integrals that are marked as unstable by `Collier`, we perform an alternative reduction to scalar integrals using the OPP reduction technique [42] with the program `CutTools` [43] by using the interface with `Recola` of Ref. [44]. In this case, we construct the numerator of the one-loop integrals using the tensor coefficients calculated by `Recola` (in double precision) and multiply it with the tensor $q^{\mu_1}\dots q^{\mu_{r_t}}$. The reduction as well as the calculation of the scalar integrals with `OneLOop` [45] is done in quadruple precision.

The rescaling of the Higgs boson self coupling by $\kappa_\lambda = \lambda_3/\lambda_3^{\text{SM}}$ in `Recola` is performed with the already built-in function to rescale a specific tree-level vertex, which is sufficient for our purpose. On the other hand, `Recola` does not support modifications of numerical input parameters after the process initialization, such as the top-quark mass, which would be required in the $\overline{\text{MS}}$ top-quark mass scheme when using a dynamical scale definition for μ_t. A similar problem has already been encountered in Ref. [46] when using `Recola 2` [47], where the authors reinitialized this program for each phase-space point, leading to an increase in computation time by a factor of 5. Instead, we have implemented in `Recola` the possibility to update the top-quark mass even after the initialization phase, following the same approach that is already used in `Recola` to update the UV counterterms. In `ggxy` this option is only enabled if the $\overline{\text{MS}}$ top-quark mass scheme is used and results in only a moderate increase of computation time of about 10 % of the matrix elements. Because of this modification in the `Recola` version included in `ggxy`, it is not straightforward to replace it with a different version.[2]

As an additional cross check we have computed analytic results for the one-loop helicity amplitudes of $gg \to HHg$ in terms of scalar integrals that are evaluated with `CutTools` and `OneLOop`. For the computation of the helicity amplitudes we have generated all Feynman diagrams with `qgraf` [48], which are mapped onto different topologies and converted to `FORM` [49] notation with `tapir` [50] and `exp` [51,52]. The computation of the diagrams is then performed with the in-house code `calc` and the reduction to master integrals is carried out with `Kira` [53,54]. We find good agreement between the calculation with `Recola` and the analytic results. However, in the latter case the coefficients in front of the scalar integrals turn out to be not sufficiently numerically stable over the whole phase space. For this reason we use the approach based on `Recola` as our default option.

[2] Currently, all modifications necessary for this can be tracked by searching the tag `dynamic_params` in the source code.

4. Using `ggxy`

4.1. Installation and structure

`ggxy` can be downloaded or cloned from the repository hosted at https://gitlab.com/ggxy/ggxy-release. It contains the files and directories

```
CMakeLists.txt    README.md    example-
build.sh    examples/
include/          lib_ext/     src/
```

where `CMakeLists.txt` is the `CMake` configuration file of `ggxy`, `README.md` contains useful information, `example-build.sh` is an example script to build `ggxy` using `CMake`. The directory `examples/` contains two subdirectories `gghh-FF/` and `gghh-nlo/` which contain examples for the usage of `ggxy` to calculate the two form factors as well as $\mathcal{V}_{\text{fin}}$, and the calculation of integrated and differential hadronic cross sections at LO and NLO QCD. The examples are discussed in more detail in the next section. The directories `src/` and `include/` contain the source code and the corresponding header files of `ggxy`, respectively. For convenience the source code of the following external libraries is located in the directory `lib_ext`:

- `avhlib` [55,56]: Phase-space generation.
- `Collier` [41]: Numerical evaluation of one-loop functions for real radiation.
- `CRunDec` [57]: Running and decoupling for α_s and the top quark mass.
- `CutTools` [43]: Fall-back option for reduction to scalar integrals.
- `OneLOop` [45]: Fall-back option for numerical evaluation of one-loop functions for real radiation.
- `Recola` [40]: Generation of amplitudes for real corrections.

They are directly compiled together and correctly linked with `ggxy` by `CMake`. In addition, it is required that the following libraries are already installed:

- `boost` [58] and `eigen` [31]: `C++` libraries with convenient containers and functions, in particular in the context of linear algebra.
- `LHAPDF` [59,60]: Provides parton distribution functions.

We note that we have developed and tested `ggxy` using `GCC` 7.5 and 13.3, `boost` 1.66 and 1.83, `eigen` 3.4 and `LHAPDF` 6.2.0. In order to evaluate polylogarithmic functions we have included the code from the ancillary files of Ref. [61] directly into `ggxy`, which can be found in `src/lievaluate/`.

For the installation of `ggxy` it is sufficient to provide only the path to the `LHAPDF` directory which contains the directories `include` and `lib/lib64`, in the variable `LHAPDFPATH` in example build script. In addition, it is possible to compile `ggxy` only for the evaluation of the form factors by setting `onlyFF=On` in `example-build.sh`. In this case the path to `LHAPDF` is ignored and only `CRunDec` from the external libraries is compiled together with `ggxy`. With `onlyFF=On` the examples in `examples/gghh-FF/` can still be compiled whereas the examples in `examples/gghh-nlo/` require `onlyFF=Off`. Further details about the compilation can be found in `README.md`.

By running the installation script all external libraries of `lib_ext/` and `ggxy` are built in the directory `example-build/`. In addition, a second directory is created, `example-install/`, that contains the subdirectory `include/` with all header and module files of the external libraries and `ggxy`. The shared-libraries of all external libraries and `ggxy` are placed in the directory `example-install/lib/`, so that the content of `example-install/` is sufficient to link `ggxy` with other programs.

4.2. Partonic form factors

The elementary building blocks implemented at the partonic level are functions for the form factors F_1 and F_2 at one- and two-loop order

in QCD. ggxy allows for an easy access to the finite parts of the form factors as defined in Eq. (4). The function prototype looks as follows

```
complex<double> gghhFF(int loops, int ff, double s, double t,
double mhs, double mts,
double murs = gghhFFmursDefault,
double muts = 0.0, unsigned scheme = 0,
double kappa_lam = 1.0, double dTriCoeff = 1.0);
```

where the parameters are defined as follows,

- `loops`: QCD loop order, 1 or 2
- `ff`: choice of form factor, 1 or 2
- `s,t`: Mandelstam variables
- `mhs`: squared Higgs boson mass, m_H^2
- `mts`: squared top quark mass, m_t^2
- `murs`: squared renormalization scale μ_r^2
- `muts`: squared renormalization scale for the $\overline{\text{MS}}$ top quark mass μ_t^2
- `scheme`: choice of renormalization scheme for top quark mass, 0 (OS) or 1 ($\overline{\text{MS}}$).
- `kappa_lam`: corresponds to κ_λ
- `dTriCoeff`: additional coefficient for the double-triangle contribution, see Eq. (2); can be used, for e.g., to switch on and off the double-triangle contribution

The default value of `murs` is set to the pre-processor variable `gghhFFmursDefault` which stands for the choice $\mu_r^2 = -s$. At one-loop order the exact results [11,12] are implemented.[3] At two-loop order, depending on the numerical value of p_T, routines for either the small-t or the high-energy expansions [10] are called.

Using the function `gghhFF()` it is straightforward to construct the quantity $\mathcal{V}_{\text{fin}}$ in Eq. (9) using the on-shell or $\overline{\text{MS}}$ scheme for different choices of μ_t. For convenience ggxy provides the function

```
double gghh2lVfin(double s, double t, double mhs, double
mts, double GF,
double murs = gghhFFmursDefault, double muts = 0.0,
unsigned scheme = 0, double kappa_lam = 1.0);
```

where the meaning of the parameters is as above with the additional parameter `GF` for G_F. In this case, the pre-processor variable `gghhFFmursDefault` stands for $\mu_r^2 = s/4$. Note that this is in contrast to the default value of `murs` $= \mu_r^2 = -s$ used for the form factors. `gghh2lVfin` only accepts positive values for `murs`.

In addition, we provide the function

```
double gghh1l(double s, double t, double mhs, double mts,
double GF, double kappa_lam = 1.0);
```

which can be used to calculate the LO squared matrix element with a factor of α_s^2 is factored out, and the function

```
vector<double> gghh2l(double s, double t, double mhs,
double mts,
double GF, double murs = gghhFFmursDefault,
double muts = 0.0, unsigned scheme = 0,
double kappa_lam = 1.0);
```

which returns the LO squared matrix element as the first vector element and $\mathcal{V}_{\text{fin}}$ as the second.

4.3. Functions for hadronic cross sections

The calculation of hadronic total and differential cross sections is managed by the class `mc_gen` which has the following functionality:

- Set input parameters (m_t, m_H, top-quark mass scheme, renormalization and factorization scales, PDF set, …).
- Integrate all contributions needed for NLO predictions together, or each contribution individually.
- Optimize phase-space integration.
- Perform Monte-Carlo integration.
- Fill results into histograms and write final results to disk.

The Monte-Carlo integration is constructed based on several integration channels for the different subprocesses in the real corrections and of the individual contributions with a Born-like phase space. The individual weights of these channels are optimized during the optimization phase of the program following the approach of Ref. [62]. The phase spaces of Born-like contributions and of the real corrections are generated with `Kaleu` [55,56] as part of the `avhlib` library, which is a multi-channel phase-space generator that performs further optimizations on-the-fly.

In the following we provide a brief summary of the functions that can be called; a detailed example of the class is given in the file `examples/gghh-nlo/nlo-gghh.cpp`. The `mc_gen` class is called as

```
mc_gen gen = mc_gen(int seed, double ss);
```

where the first parameter is the seed for the initialization of the random numbers and the second parameter is the hadronic centre-of-mass energy $\sqrt{s}$ in units of GeV. The structure of this class is process independent. However, currently we offer only a configuration of `mc_gen` for the production of two Higgs bosons at the LHC that is called with

```
configure_mc_gghh(string& mode, mc_gen& gen);
```

The first parameter defines the contribution that should be integrated over the phase space and can be one of the following keywords:

- `lo`: LO cross section
- `nlo`: NLO cross section
- `V`: Virtual corrections given by $\frac{\alpha_s}{2\pi}\widetilde{\mathcal{V}}_{\text{fin}}$
- `I`: **I** operator of Catani-Seymour subtraction scheme
- `KP`: Sum of **K** and **P** operator of Catani-Seymour subtraction scheme
- `RS`: Real subtracted contribution using Catani-Seymour subtraction scheme
- `LOVDIP`: Sum of `lo`, `V`, `I` and `KP`.

Note that `nlo` corresponds to the sum of `LOVDIP` and `RS`.

The parameters of the process should be modified by one of the following class methods

```
void set_mass_top(double mtop);
void set_mass_higgs(double mhiggs);
void set_gfermi(double gfermi);
```

where the default values can be found in `src/tools/params.cpp`. In addition, it is possible to rescale the trilinear self-coupling of the Higgs boson by κ_λ with the function

```
void set_kappa_lam(double kappa_lam);
```

The top-quark mass scheme can be set with the function

```
void set_mtscheme(unsigned mtscheme, int crd_runLoops_as = 5,
int crd_runLoops_mt = 5, int crd_convLoops_mt = 4);
```

where again `mtscheme=0` corresponds to the on-shell and `mtscheme=1` to the $\overline{\text{MS}}$ scheme. The last three parameters of the function `set_mtscheme` control the loop order for the running of the strong coupling constant and top-quark mass, and the loop order for the conversion from the on-shell to the $\overline{\text{MS}}$ top-quark mass. The conversion and the running are performed with `CRunDec` and the default values of the parameters are set to highest orders available. The input values $\alpha_s(m_Z)$ and m_Z for the running are read from the PDF set. The strong coupling constant, which appears explicitly in the matrix element, is always calculated with `LHAPDF`, where the parameters are controlled by the given PDF set. Thus, the last two parameters are always ignored in the on-shell scheme.

The generator is able to produce results for different values of the renormalization and factorization scales in a single run. In particular, it

[3] At the border of the phase space we switch to expansions which provide more stable results.

is possible to perform the usual 7-point scale variation to estimate theoretical uncertainties, where the central values ($\mu_{r,0}$, $\mu_{f,0}$) of the renormalization and factorization scales are varied as follows:

$$\left(\frac{\mu_r}{\mu_{r,0}}, \frac{\mu_f}{\mu_{f,0}}\right) \in \{(1,1),(0.5,0.5),(2,2),(2,1),(0.5,1),(1,2),(1,0.5)\}. \quad (20)$$

The names of the corresponding scale definitions, the information about the variation, and the PDF set is either initialized with

```
void set_scales_pdfs(vector<string>& scale_names,
int gridtype,
string pdf_name);
```

or

```
void set_scales_pdfs(vector<string>& scale_names,
vector<vector<double>> igrid,
string pdf_name);
```

where the first input parameter is a vector with the names of the scales. They are only used for bookkeeping purposes and appear as a label in the histogram files which are generated by ggxy. The actual definition of the scales happens in the function `set_scale` (see also below) where vectors for the renormalization and factorization scales, with the same length as the vector containing the names of the scales, are introduced. The last input parameter, `set_scales_pdfs`, is the name of the PDF set as defined in LHAPDF. The second input parameter is either an integer number to use a predefined grid, where `gridtype=0` corresponds to the case where only the central value of the scale is calculated. With `gridtype=1` the 7-point scale variation as given in Eq. (20) is performed. Alternatively, a matrix that contains the information about the scale variations can be provided. For example, the 7-point scale variation of Eq. (20) can also be calculated with

```
igrid={{1.0,1.0},{0.5,0.5},{2.0,2.0},{2.0,1.0},
       {0.5,1.0},{1.0,2.0},{1.0,0.5}}
```

The renormalization and factorization scales are then defined with

```
void set_scalefunc(scalefunc set_scale);
```

where `scalefunc` is a type definition of a function (or pointer to a function) with the following prototype

```
void set_scale(vector<int>& iproc,
vector<lorentz_vec>& p,
params& pars,
vector<double>& muR, vector<double>& muF,
double& mut);
```

Here `iproc` is a vector containing the identification of the particles of the process following the conventions of the PDG [63] and `p` is a vector containing the corresponding four momenta in the center-of-mass system. The parameter `pars` is a class containing all numerical parameters, so that the top-quark or Higgs boson mass can be obtained by `pars.mt` or `pars.mh`, respectively. The different scale settings for the renormalization and factorization scales are defined in the vectors `muR` and `muF`, where the length of these vectors is equal to the length of the vector `scale_names` which has been used in the function `set_scales_pdfs` to initialize the scale and PDF settings. In the case of the $\overline{\text{MS}}$ top-quark mass scheme the corresponding scale μ_t has to be passed as the parameter `mut`. For convenience the top-quark mass $\overline{m}_t(\overline{m}_t)$ is stored in `pars.mtmt`.

Histograms can be defined and filled with the functions

```
void set_hinit(histo_init hinit);
```

and

```
void set_hfill(histo_fill hfill);
```

where `histo_init` and `histo_fill` are type definitions to functions with the following prototypes

```
void hinit(histo& hlist);
```

and

```
void hfill(histo& hlist,
vector<int>& iproc,
vector<lorentz_vec>& plab,
params& pars);
```

The histograms can then be initialized in the function `hinit` with the following class method of `hlist`

```
void add(string& name, int bn,
double start, double end);
```

where the first input parameter is the observable name, `bn` is the number of bins and the variables `start` and `end` define the range of the histogram. Alternatively, it is possible to add histograms with

```
void add(string& name, int bn,
vector<double>& bins);
```

where in this case the vector `bins` should contain the edges of the histograms so that this function can be used to create histograms with unequal bin sizes. The observables should then be constructed in the function `hfill` where the variables `iproc` and `pars` are the same as in the function `set_scale` and `plab` are the particle momenta in the laboratory frame. The histograms should then be filled in this function with

```
void fill(int ih, double val);
```

which is a class method of `hlist`. The variable `ih` is used to identify a histogram defined in `hfill`, where the first histogram defined with `add` has `ih`= 0, and the variable `val` is the value of the observable for this event. Finally, it is possible to define a function for possible phase-space cuts

```
bool event_cut(vector<int>& iproc,
vector<lorentz_vec>& plab,
params& pars);
```

where the input parameters are identical to `hfill`. This function should return `false` if the event should be rejected and otherwise `true`. The function can then be given to the generator by using the class method

```
void set_event_cut(cutfunc event_cut);
```

of `mc_gen`.

The initialization of the generator is considered complete when the class method

```
void finish_init();
```

is called, after which none of the functions above should be called and the optimization phase is activated. In this phase the weights of all integration channels and the phase-space generator `Kaleu` are optimized. The generation of phase-space points is then achieved with the class method

```
vector<double> integrate(int npT, int istep=10000);
```

where the variable `npT` is the number of phase-space points that should be generated and `istep` defines the number of phase-space points after which log information is printed. The return vector contains the cross section of the first scale definition and the corresponding MC uncertainty. The optimization phase is deactivated by calling the class method

```
void set_phase_optim(bool phase_optim);
```

with `phase_optim=false`, after which the histogram will be filled in the next call of `integrate`. All MC integration weights including the histograms, can be reset with the class method

```
void reset_weights();
```

of `mc_gen`. Finally, the histograms can be saved to disk with

```
void output(string outfile);
```

where `outfile` is the output file.

5. Example results

The example files to compute partonic and hadronic quantities are in the subdirectories `examples/gghh-FF/` and `examples/gghh-nlo/`, respectively. They are compiled by executing the script `build.sh` in the corresponding directory.

5.1. One- and two-loop form factors

Using the functions `gghh<n>lFF<i>` it is straightforward to reproduce numerical results for the form factors present in the literature. The example file `examples/gghh-FF/ff.cpp` shows how the data points for the (exact) one- and two-loop curves in Fig. 3 of Ref. [64] can be generated. In addition the example shows how the corresponding data points for a $\overline{\text{MS}}$ top quark mass can be generated.

5.2. Virtual NLO corrections

The ultraviolet renormalized and IR subtracted virtual corrections $\mathcal{V}_{\text{fin}}$ from Eq. (9) are implemented in the function `gghh2lVfin()`, described above. The example given in the file `examples/gghh-FF/checkhhgrid.cpp` calls this function for all 6320 data points contained in the `hhgrid` [7] interpolation grid and evaluates them in less than 10 seconds; this performance allows the use of `ggxy` for Monte-Carlo studies. In principle the use of the grid for interpolation might be even faster, however it lacks the flexibility to change the parameters such as the masses of the top quark or the Higgs boson. Extending the grid, for example for a running top quark mass with an m_{HH}-dependent scale μ_t would require computational resources many orders of magnitude greater than our implementation. All results from Ref. [7] are validated within the uncertainties.

5.3. Hadronic cross sections

The file `examples/gghh-nlo/nlo-gghh.cpp` illustrates how hadronic cross sections can be computed. It contains auxiliary functions to define the renormalization and factorization scales and to initialize and fill the histograms. In the main part of the program one first selects which part of the NLO corrections to compute and initializes the Monte-Carlo generator. Afterwards one specifies the input parameters using, e.g.

```
gen.set_mass_top(mt);
```

The renormalization scheme for the top quark mass is selected via

```
gen.set_mtscheme(mtscheme);
```

After initializing the histograms and specifying cuts[4] it is possible to initiate the Monte-Carlo integration. Finally the generated data are stored to disk. They are used to obtain total and differential cross sections which are discussed in the following subsections.

5.3.1. Total hadronic cross section

In the sample file `examples/gghh-nlo/nlo-gghh.cpp` the total hadronic cross section is computed using the input values from Ref. [3]. In particular, we use the PDF set `PDF4LHC15_nlo_100_pdfas` which is obtained via the interface of LHAPDF [59]. The renormalization and factorization scales are set to a common scale $\mu_r = \mu_f = m_{HH}/2$. We reproduce the central values and uncertainties based on the on-shell top quark mass given in Ref. [3] both for $\sqrt{s} = 14$ TeV and $\sqrt{s} = 100$ TeV. The comparison is shown in Table 1. The results from `ggxy` are produced by averaging the results from five different seeds. The relative uncertainty for each seed is about 0.2 % with a run time of about 30 minutes on a single core on a AMD Ryzen Threadripper PRO 3955WX processor. An example script `examples/gghh-nlo/run.sh` is provided to illustrate the usage of the sample file to compute cross sections with different seeds in parallel followed by a combination of the results using python scripts.

Table 1
Comparison with results of Ref. [3] for $\sqrt{s} = 14$ TeV and $\sqrt{s} = 100$ TeV.

$\sqrt{s}$		ggxy	Ref. [3]
14 TeV	σ^{LO} [fb]	$19.848(4)^{+27.6\,\%}_{-20.5\,\%}$	$19.85^{+27.6\,\%}_{-20.5\,\%}$
	σ^{NLO} [fb]	$32.92(2)^{+13.6\,\%}_{-12.6\,\%}$	$32.91^{+13.6\,\%}_{-12.6\,\%}$
100 TeV	σ^{LO} [fb]	$731.2(2)^{+20.9\,\%}_{-15.9\,\%}$	$731.3^{+20.9\,\%}_{-15.9\,\%}$
	σ^{NLO} [fb]	$1150(1)^{+10.8\,\%}_{-10.0\,\%}$	$1149^{+10.8\,\%}_{-10.0\,\%}$

Table 2
Comparison with results of Ref. [6].

$\sqrt{s}$	κ_λ	$\sigma^{\text{NLO}}_{\text{ggxy}}$ [fb]	σ^{NLO}[6] [fb]
13 TeV	−10.0	1424.1(9)	1438(1)
13 TeV	−5.0	509.4(3)	512.8(3)
13 TeV	−1.0	113.53(7)	113.66(7)
13 TeV	0.0	61.36(4)	61.22(6)
13 TeV	1.0	27.72(2)	27.73(7)
13 TeV	2.0	12.777(9)	13.2(1)
13 TeV	2.4	12.035(8)	12.7(1)
13 TeV	3.0	16.50(1)	17.6(1)
13 TeV	5.0	80.00(5)	83.2(3)
13 TeV	10.0	564.7(4)	579(1)
14 TeV	1.0	32.79(2)	32.81(7)

For illustration the total cross section using the $\overline{\text{MS}}$ scheme can also be calculated with the example file by only setting `mtscheme=1`, where we use $\mu_t = \overline{m}_t(\overline{m}_t)$. We obtain

$$\sigma_{\text{tot}}(gg \to HH) = 31.63(2)\ \text{fb}\,. \tag{21}$$

The cross section is calculated again by averaging over five seeds which all have a relative uncertainty of less than 0.2 %. The runtime increased slightly to about 35 minutes.

5.3.2. Differential distributions

The data files generated in the runs for the total cross section in the above example also contain the data points for the differential cross section distributions. As an example in Fig. 1 the invariant mass distribution of the di-Higgs system and the average of both transverse momenta of the two Higgs bosons are presented at LO and NLO QCD. These plots are generated by the `python` scripts provided in `examples/gghh-nlo/`. By default, the example script considers five seeds. However, depending on the observable and the kinematic ranges, more seeds might be required to obtain smooth distributions.

5.4. Variation of λ

With the command

```
gen.set_kappa_lam(kappa_lam);
```

the quantity κ_λ can be modified.

In Table 2 we compare the results for the total cross section to Refs. [6] (see also Ref. [5]) for different choices of κ_λ. We observe good agreement within the uncertainties for small values of κ_λ (i.e., for $|\kappa_\lambda| \lesssim 1$). However for larger values of $|\kappa_\lambda|$ significant differences are observed. Similar deviations are also observed in Ref. [37], see the discussion in Section 3.1 of that paper.

Finally, in Table 3 we compare to the results provided in Ref. [37] for different values of κ_λ. After adopting their input parameters i.e., $\sqrt{s} = 13.6$ TeV, $M_t = 172.5$ GeV, $m_H = 125$ GeV and the PDF set `NNPDF31_nlo_as_0118` we observe good agreement, however in Ref. [37] no Monte Carlo uncertainties are given with which we could compare. We also agree with the results given in the revised version of Ref. [65], see also Ref. [66].

[4] In our example no cuts are applied.

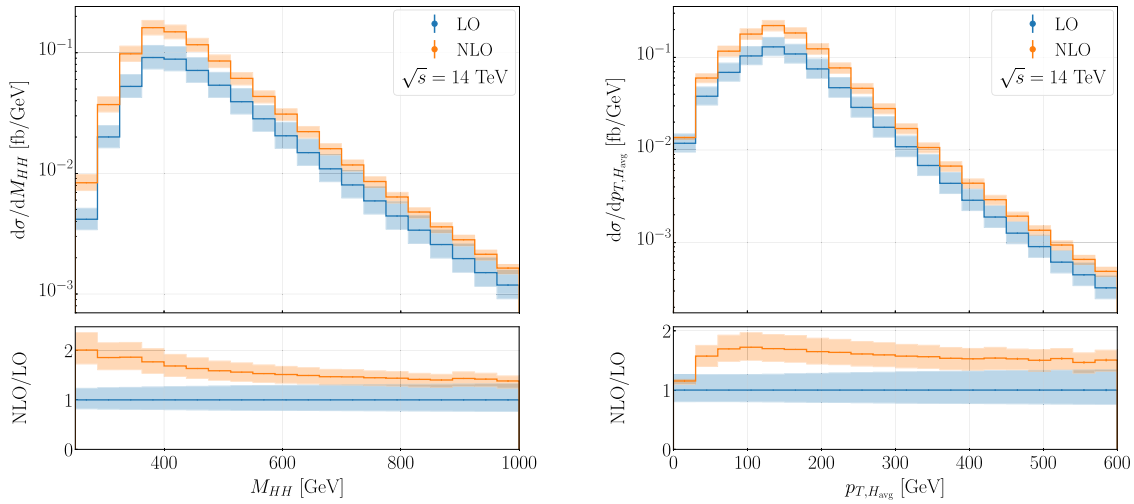

Fig. 1. LO and NLO QCD predictions for the invariant mass distribution of the di-Higgs system and the average of both transverse momenta of the two Higgs bosons. Theoretical uncertainties from scale variation are shown as bands and the statistical uncertainties from Monte Carlo integration are shown as vertical lines. Note that the latter are small and are only visible for larger values of M_{HH} and $p_{T,H_{avg}}$.

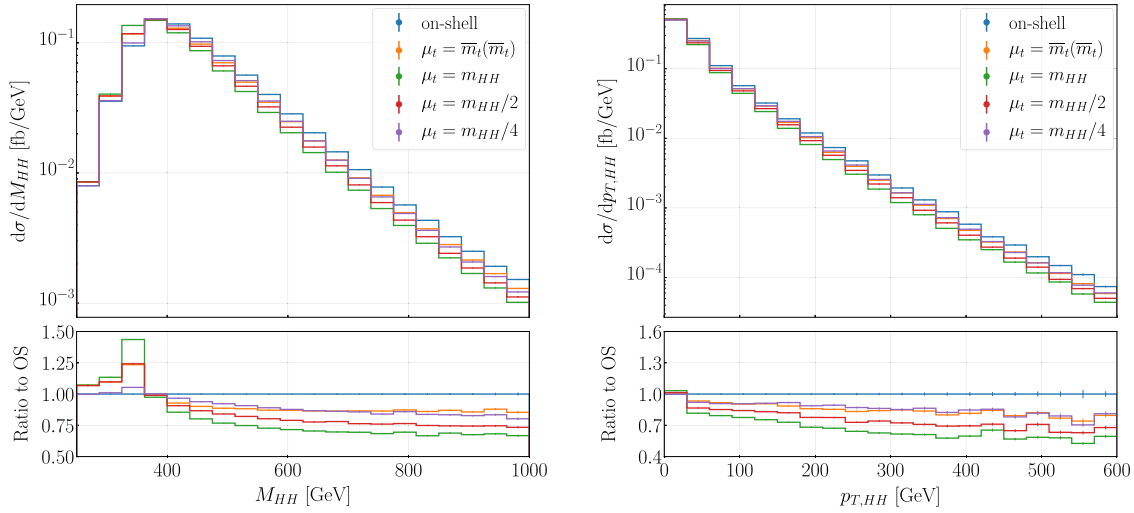

Fig. 2. NLO predictions for m_{HH} and $p_{T,HH}$ distributions using the on-shell mass scheme as well as the $\overline{\text{MS}}$ scheme for different settings of μ_t. Input parameters are adapted from Ref. [37] with $\sqrt{s} = 13.6$ TeV. The lower panels display the ratio to the on-shell predictions. Statistical uncertainties from Monte Carlo integration are shown as vertical lines.

Table 3
Comparison with results of Ref. [37].

κ_λ	Top-mass scheme	$\sigma^{\text{NLO}}_{\text{ggxy}}$ [fb]	σ^{NLO}[37] [fb]
−0.6	on-shell	$100.34(6)^{+15.7\,\%}_{-13.6\,\%}$	$100.77^{+15.8\,\%}_{-13.7\,\%}$
0.0	on-shell	$68.08(4)^{+15.1\,\%}_{-13.4\,\%}$	$68.38^{+15.1\,\%}_{-13.4\,\%}$
1.0	on-shell	$30.83(2)^{+13.8\,\%}_{-12.7\,\%}$	$30.93^{+13.7\,\%}_{-12.7\,\%}$
1.0	$\overline{\text{MS}}$, $\mu_t = \overline{m}_t(\overline{m}_t)$	$29.78(2)^{+14.3\,\%}_{-13.0\,\%}$	$29.78^{+14.3\,\%}_{-13.0\,\%}$
1.0	$\overline{\text{MS}}$, $\mu_t = M_{HH}/2$	$28.79(2)^{+15.3\,\%}_{-13.5\,\%}$	$28.90^{+15.2\,\%}_{-13.5\,\%}$
2.4	on-shell	$13.369(9)^{+14.7\,\%}_{-13.2\,\%}$	$13.41^{+14.8\,\%}_{-13.1\,\%}$
6.6	on-shell	$203.4(1)^{+19.0\,\%}_{-15.1\,\%}$	$203.91^{+19.0\,\%}_{-15.2\,\%}$

5.5. Top quark mass renormalization scheme dependence

The sample file `examples/nlo-gghh.cpp` can also be used to study the dependence on the top quark mass renormalization scheme. We perform separate runs for the on-shell (`mtscheme=0`) and the $\overline{\text{MS}}$ scheme (`mtscheme=1`). The scale μ_t can be modified in the function `set_scale`. For example, $\mu_t = \overline{m}_t(\overline{m}_t)$ corresponds to `mut = pars.mtmt` and $\mu_t = m_{HH}/2$ is obtained with `mut = phh.mass()/2.0`. In Fig. 2 we show the differential distributions for the observables m_{HH} and $p_{T,HH}$ in the on-shell and the $\overline{\text{MS}}$ scheme for different choices of μ_t. The lower panels display the ratio to the on-shell predictions. We reproduce the results from Refs. [5,37].

6. Conclusions and outlook

In this paper we present the fast and flexible library ggxy which can be used to compute partonic and hadronic quantities related to the loop-induced gluon-initiated processes. In Version 1 we implement the functionality which allows the computation of NLO QCD corrections to Higgs boson pair production. Example files are provided which demonstrate how to compute LO and NLO corrections to the form factors, NLO virtual corrections, total cross sections and distributions. All results can be obtained for on-shell or $\overline{\text{MS}}$ top quark masses. Furthermore, it is possible to modify the self-coupling of the Higgs boson λ. The typical runtime for partonic quantities is a few milliseconds and for hadronic quantities minutes to hours. The high degree of flexibility and the low runtime makes ggxy attractive for use as an amplitude library for parton shower programs such as, e.g., `POWHEG` [67].

It is straightforward to extend ggxy in various directions. NLO QCD corrections to processes such as top-quark mediated $gg \to ZZ$ or $gg \to ZH$ can be implemented in complete analogy to $gg \to HH$ and will be made available in a future version. The implementation of the virtual corrections for other gluon-induced processes is quite straightforward, once sufficiently deep small-t and high-energy expansions are available since they have the same structure as for the $gg \to HH$ process. In particular, we can re-use the implementation of the master integrals for the small-t expansion and the routines for the construction of the Padé approximants. For some processes (e.g. for $gg \to ZZ$) there are also contributions which do not involve the top quark. If these corrections are available in the literature they can be implemented in ggxy to provide the full amplitude. As for the real-radiation contributions, the corresponding processes can be provided in a straightforward way using Recola. In future versions we additionally plan to implement NNLO QCD and NLO electroweak corrections for these processes.

Note added:

A user process for $gg \to HH$ based on ggxy has been implemented in the POWHEG-BOX framework, and can be obtained from https://gitlab.com/POWHEG-BOX/V2/User-Processes/ggxy_ggHH.

CRediT authorship contribution statement

Joshua Davies: Writing – original draft, Software; **Kay Schönwald:** Writing – original draft, Software; **Matthias Steinhauser:** Writing – original draft, Software; **Daniel Stremmer :** Writing – original draft, Software.

Data availability

Data will be made available on request.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grant 396021762 — TRR 257 “Particle Physics Phenomenology after the Higgs Discovery”. The work of K. S. was supported by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme grant agreement 101,019,620 (ERC Advanced Grant TOPUP) and the UZH Postdoc Grant, grant no. [FK-24-115]. The work of J. D. was supported by STFC Consolidated Grant ST/X000699/1. We thank Emanuele Bagnaschi for communications concerning Ref. [37] and Gudrun Heinrich for comments on the manuscript.

Appendix A. Low-level functions

Beyond the “high-level” interface to the form factors described in Section 4.2, it is also possible to call the “low-level” functions which provide results for the exact (at one loop) and high-energy and small-t expansions (at one and two loops) as well as the exact two-loop double-triangle contribution. In these functions, the triangle contribution of F_1 is separated from the box contribution; these pieces are called `FF0` and `FF1` in the function names, and F_2 is called `FF2`.

The exact one-loop form factors can be evaluated using the functions defined in the header file `ff/gghh/EXgghh1lFF.h`,

```
complex<double> EXgghh1lFF{0,1,2}(double s, double t,
double mhs, double mts);
```

and the exact two-loop double-triangle contribution using the functions defined in the header file `ff/gghh/DTgghh2lFF.h`,

```
complex<double> DTgghh2lFF{1,2}(double s, double t,
double mhs, double mts);
```

where although the triangle contribution is t-independent, each function has the same signature. Here and below, the notation `EXgghh1lFF{0,1,2}` implies that each of `EXgghh1lFF0`, `EXgghh1lFF1` and `EXgghh1lFF2` is defined in the library.

At one and two loops, the high-energy and small-t expansion results can be evaluated using the functions defined in the header files `ff/gghh/HEgghh{1,2}lFF.h` and `ff/gghh/t0gghh{1,2}lFF.h`,

```
complex<double> HEgghh{1,2}lFF{0,1,2}(double s, double t,
double mhs, double mts);
```

where the Padé approximation procedure has been already used, and

```
complex<double> t0gghh{1,2}lFF{0,1,2}(double s, double t,
double mhs, double mts);
```

which returns the sum of the small-t expansion terms. At this level, the functions should return numerically stable results within each expansion’s region of validity, but return nonsensical results beyond these regions.

At the “lowest” level, vectors of the expansion coefficients can be obtained. At this level, no attempt is made to return sensible results in numerically unstable regions or regions beyond the validity of the expansions. These functions are defined in the header files `ff/gghh/HEgghh{1,2}lFF-lowlevel.h` and `ff/gghh/t0gghh{1,2}lFF-lowlevel.h`. For the high-energy expansion, the expansion coefficients for the real and imaginary parts of each of the m_H^2 expansion terms are returned by

```
vector<double> HEgghh{1,2}lFF{0,1,2}mhs{0,1,2}{re,im}(
double s, double t,
double mhs, double mts,
unsigned mtsExpDepth = 24);
```

where the final parameter controls the depth at which the expansion is evaluated (at most 24). The small-t expansion coefficients are given by

```
vector<complex<double>> t0gghh{1,2}lFF{0,1,2}mhs{0,1,2}(
double s, double t,
double mhs, double mts,
unsigned tExpDepth = 6);
```

where again the final parameter controls the expansion depth (at most 6). At one loop, higher-order m_H^2 terms are available: `mhs{3,4}`.

References

[1] D. de Florian, et al., LHC Higgs Cross Section Working Group, Handbook of LHC higgs cross sections: 4. deciphering the nature of the higgs sector 2/2017 (2016). 1610.07922. https://doi.org/10.23731/CYRM-2017-002.

[2] S. Borowka, N. Greiner, G. Heinrich, S.P. Jones, M. Kerner, J. Schlenk, U. Schubert, T. Zirke, Higgs boson pair production in gluon fusion at next-to-Leading order with full top-Quark mass dependence, Phys. Rev. Lett. 117 (1) (2016) 012001. [Erratum: Phys. Rev. Lett. 117, 079901 (2016)]. 1604.06447. https://doi.org/10.1103/PhysRevLett.117.079901

[3] S. Borowka, N. Greiner, G. Heinrich, S.P. Jones, M. Kerner, J. Schlenk, T. Zirke, Full top quark mass dependence in higgs boson pair production at NLO, JHEP 10 (2016) 107. 1608.04798. https://doi.org/10.1007/JHEP10(2016)107

[4] J. Baglio, F. Campanario, S. Glaus, M. Mühlleitner, M. Spira, J. Streicher, Gluon fusion into higgs pairs at NLO QCD and the top mass scheme, Eur. Phys. J. C 79 (6) (2019) 459. 1811.05692. https://doi.org/10.1140/epjc/s10052-019-6973-3

[5] J. Baglio, F. Campanario, S. Glaus, M. Mühlleitner, J. Ronca, M. Spira, J. Streicher, Higgs-Pair production via gluon fusion at hadron colliders: NLO QCD corrections, JHEP 04 (2020) 181. 2003.03227. https://doi.org/10.1007/JHEP04(2020)181

[6] J. Baglio, F. Campanario, S. Glaus, M. Mühlleitner, J. Ronca, M. Spira, $gg \to HH$: Combined uncertainties, Phys. Rev. D 103 (5) (2021) 056002. 2008.11626. https://doi.org/10.1103/PhysRevD.103.056002

[7] https://github.com/mppmu/hhgrid.

[8] J. Davies, G. Heinrich, S.P. Jones, M. Kerner, G. Mishima, M. Steinhauser, D. Wellmann, Double higgs boson production at NLO: combining the exact numerical result and high-energy expansion, JHEP 11 (2019) 024. 1907.06408. https://doi.org/10.1007/JHEP11(2019)024

[9] L. Bellafronte, G. Degrassi, P.P. Giardino, R. Gröber, M. Vitti, Gluon fusion production at NLO: merging the transverse momentum and the high-energy expansions, JHEP 07 (2022) 069. 2202.12157. https://doi.org/10.1007/JHEP07(2022)069

[10] J. Davies, G. Mishima, K. Schönwald, M. Steinhauser, Analytic approximations of 2 → 2 processes with massive internal particles, JHEP 06 (2023) 063. 2302.01356. https://doi.org/10.1007/JHEP06(2023)063

[11] E.W.N. Glover, J.J. van der Bij, HIGGS BOSON PAIR PRODUCTION VIA GLUON FUSION, Nucl. Phys. B 309 (1988) 282–294. https://doi.org/10.1016/0550-3213(88)90083-1

[12] T. Plehn, M. Spira, P.M. Zerwas, Pair production of neutral higgs particles in gluon-gluon collisions, Nucl. Phys. B 479 (1996) 46–64. [Erratum: Nucl.Phys.B 531, 655–655 (1998)]. hep-ph/9603205. https://doi.org/10.1016/0550-3213(96)00418-X

[13] R. Harlander, P. Kant, Higgs production and decay: analytic results at next-to-leading order QCD, JHEP 12 (2005) 015. hep-ph/0509189. https://doi.org/10.1088/1126-6708/2005/12/015

[14] C. Anastasiou, S. Beerli, S. Bucherer, A. Daleo, Z. Kunszt, Two-loop amplitudes and master integrals for the production of a higgs boson via a massive quark and a scalar-quark loop, JHEP 01 (2007) 082. hep-ph/0611236. https://doi.org/10.1088/1126-6708/2007/01/082

[15] U. Aglietti, R. Bonciani, G. Degrassi, A. Vicini, Analytic results for virtual QCD corrections to higgs production and decay, JHEP 01 (2007) 021. hep-ph/0611266. https://doi.org/10.1088/1126-6708/2007/01/021

[16] G. Degrassi, P.P. Giardino, R. Gröber, On the two-loop virtual QCD corrections to higgs boson pair production in the standard model, Eur. Phys. J. C 76 (7) (2016) 411. 1603.00385. https://doi.org/10.1140/epjc/s10052-016-4256-9

[17] G. Aad, et al., ATLAS, Combination of searches for higgs boson pair production in pp collisions at $s=13$ TeV with the ATLAS detector, Phys. Rev. Lett. 133 (10) (2024) 101801. 2406.09971. https://doi.org/10.1103/PhysRevLett.133.101801

[18] CMS collaboration, Combination of searches for nonresonant higgs boson pair production in proton-proton collisions at sqrt(s) = 13 TeV (2024).

[19] J. Davies, G. Mishima, M. Steinhauser, D. Wellmann, Double-Higgs boson production in the high-energy limit: planar master integrals, JHEP 03 (2018) 048. 1801.09696. https://doi.org/10.1007/JHEP03(2018)048

[20] J. Davies, G. Mishima, M. Steinhauser, D. Wellmann, Double higgs boson production at NLO in the high-energy limit: complete analytic results, JHEP 01 (2019) 176. 1811.05489. https://doi.org/10.1007/JHEP01(2019)176

[21] R. Bonciani, G. Degrassi, P.P. Giardino, R. Gröber, Analytical method for next-to-Leading-Order QCD corrections to double-Higgs production, Phys. Rev. Lett. 121 (16) (2018) 162003. 1806.11564. https://doi.org/10.1103/PhysRevLett.121.162003

[22] S. Catani, The singular behavior of QCD amplitudes at two loop order, Phys. Lett. B 427 (1998) 161–171. hep-ph/9802439. https://doi.org/10.1016/S0370-2693(98)00332-3

[23] G. Heinrich, S.P. Jones, M. Kerner, G. Luisoni, E. Vryonidou, NLO Predictions for higgs boson pair production with full top quark mass dependence matched to parton showers, JHEP 08 (2017) 088. 1703.09252. https://doi.org/10.1007/JHEP08(2017)088

[24] R. Gröber, A. Maier, T. Rauh, Reconstruction of top-quark mass effects in higgs pair production and other gluon-fusion processes, JHEP 03 (2018) 020. 1709.07799. https://doi.org/10.1007/JHEP03(2018)020

[25] M. Fael, F. Lange, K. Schönwald, M. Steinhauser, A semi-analytic method to compute feynman integrals applied to four-loop corrections to the $\overline{\text{MS}}$-pole quark mass relation, JHEP 09 (2021) 152. 2106.05296. https://doi.org/10.1007/JHEP09(2021)152

[26] M. Fael, F. Lange, K. Schönwald, M. Steinhauser, A semi-numerical method for one-scale problems applied to the $\overline{\text{MS}}$-on-shell relation, SciPost Phys. Proc. 7 (2022) 041. 2110.03699. https://doi.org/10.21468/SciPostPhysProc.7.041

[27] M. Fael, F. Lange, K. Schönwald, M. Steinhauser, Singlet and nonsinglet three-loop massive form factors, Phys. Rev. D 106 (3) (2022) 034029. 2207.00027. https://doi.org/10.1103/PhysRevD.106.034029

[28] W.H. Press, S.A. Teukolsky, W.T. Vetterling, B.P. Flannery, Numerical Recipes: The Art of Scientific Computing (Third Edition), Cambridge University Press, 2007.

[29] G. Mishima, High-Energy expansion of two-Loop massive four-Point diagrams, JHEP 02 (2019) 080. 1812.04373. https://doi.org/10.1007/JHEP02(2019)080

[30] J. Davies, G. Mishima, M. Steinhauser, D. Wellmann, $gg \to ZZ$: Analytic two-loop results for the low- and high-energy regions, JHEP 04 (2020) 024. 2002.05558. https://doi.org/10.1007/JHEP04(2020)024

[31] https://libeigen.gitlab.io/docs/.

[32] S. Catani, M.H. Seymour, A general algorithm for calculating jet cross-sections in NLO QCD, Nucl. Phys. B 485 (1997) 291–419. [Erratum: Nucl.Phys.B 510, 503–504 (1998)]. hep-ph/9605323. https://doi.org/10.1016/S0550-3213(96)00589-5

[33] Z. Nagy, Z. Trocsanyi, Next-to-leading order calculation of four jet observables in electron positron annihilation, Phys. Rev. D 59 (1999) 014020. [Erratum: Phys.Rev.D 62, 099902 (2000)]. hep-ph/9806317. https://doi.org/10.1103/PhysRevD.62.099902

[34] Z. Nagy, Next-to-leading order calculation of three jet observables in hadron hadron collision, Phys. Rev. D 68 (2003) 094002. hep-ph/0307268. https://doi.org/10.1103/PhysRevD.68.094002

[35] M. Czakon, C.G. Papadopoulos, M. Worek, Polarizing the dipoles, JHEP 08 (2009) 085. 0905.0883. https://doi.org/10.1088/1126-6708/2009/08/085

[36] F. Maltoni, E. Vryonidou, M. Zaro, Top-quark mass effects in double and triple higgs production in gluon-gluon fusion at NLO, JHEP 11 (2014) 079. 1408.6542. https://doi.org/10.1007/JHEP11(2014)079

[37] E. Bagnaschi, G. Degrassi, R. Gröber, Higgs boson pair production at NLO in the POWHEG approach and the top quark mass uncertainties, Eur. Phys. J. C 83 (11) (2023) 1054. 2309.10525. https://doi.org/10.1140/epjc/s10052-023-12238-8

[38] J.M. Campbell, G. De Laurentis, R.K. Ellis, Analytic amplitudes for a pair of higgs bosons in association with three partons, JHEP 10 (2024) 230. 2408.12686. https://doi.org/10.1007/JHEP10(2024)230

[39] S. Actis, A. Denner, L. Hofer, A. Scharf, S. Uccirati, Recursive generation of one-loop amplitudes in the standard model, JHEP 04 (2013) 037. 1211.6316. https://doi.org/10.1007/JHEP04(2013)037

[40] S. Actis, A. Denner, L. Hofer, J.-N. Lang, A. Scharf, S. Uccirati, RECOLA: REcursive Computation of one-Loop amplitudes, Comput. Phys. Commun. 214 (2017) 140–173. 1605.01090. https://doi.org/10.1016/j.cpc.2017.01.004

[41] A. Denner, S. Dittmaier, L. Hofer, Collier: a fortran-based complex one-Loop LIbrary in extended regularizations, Comput. Phys. Commun. 212 (2017) 220–238. 1604.06792. https://doi.org/10.1016/j.cpc.2016.10.013

[42] G. Ossola, C.G. Papadopoulos, R. Pittau, Reducing full one-loop amplitudes to scalar integrals at the integrand level, Nucl. Phys. B 763 (2007) 147–169. hep-ph/0609007. https://doi.org/10.1016/j.nuclphysb.2006.11.012

[43] G. Ossola, C.G. Papadopoulos, R. Pittau, Cuttools: a program implementing the OPP reduction method to compute one-loop amplitudes, JHEP 03 (2008) 042. 0711.3596. https://doi.org/10.1088/1126-6708/2008/03/042

[44] D. Stremmer, M. Worek, Complete NLO corrections to top-quark pair production with isolated photons, JHEP 07 (2024) 091. 2403.03796. https://doi.org/10.1007/JHEP07(2024)091

[45] A. van Hameren, OneLOop: for the evaluation of one-loop scalar functions, Comput. Phys. Commun. 182 (2011) 2427–2438. 1007.4716. https://doi.org/10.1016/j.cpc.2011.06.011

[46] G. Degrassi, R. Gröber, M. Vitti, X. Zhao, On the NLO QCD corrections to gluon-initiated ZH production, JHEP 08 (2022) 009. 2205.02769. https://doi.org/10.1007/JHEP08(2022)009

[47] A. Denner, J.-N. Lang, S. Uccirati, Recola2: REcursive computation of one-Loop amplitudes 2, Comput. Phys. Commun. 224 (2018) 346–361. 1711.07388. https://doi.org/10.1016/j.cpc.2017.11.013

[48] P. Nogueira, Automatic feynman graph generation, J. Comput. Phys. 105 (1993) 279–289. https://doi.org/10.1006/jcph.1993.1074

[49] B. Ruijl, T. Ueda, J. Vermaseren, FORM Version 4.2 (2017). 1707.06453.

[50] M. Gerlach, F. Herren, M. Lang, Tapir: a tool for topologies, amplitudes, partial fraction decomposition and input for reductions, Comput. Phys. Commun. 282 (2023) 108544. 2201.05618. https://doi.org/10.1016/j.cpc.2022.108544

[51] R. Harlander, T. Seidensticker, M. Steinhauser, Complete corrections of order alpha alpha-s to the decay of the z boson into bottom quarks, Phys. Lett. B 426 (1998) 125–132. hep-ph/9712228. https://doi.org/10.1016/S0370-2693(98)00220-2

[52] T. Seidensticker, Automatic application of successive asymptotic expansions of feynman diagrams, in: 6Th International Workshop on New Computing Techniques in Physics Research: Software Engineering, Artificial Intelligence Neural Nets, Genetic Algorithms, Symbolic Algebra, Automatic Calculation, 1999. hep-ph/9905298.

[53] P. Maierhöfer, J. Usovitsch, P. Uwer, Kira—a feynman integral reduction program, Comput. Phys. Commun. 230 (2018) 99–112. 1705.05610. https://doi.org/10.1016/j.cpc.2018.04.012

[54] J. Klappert, F. Lange, P. Maierhöfer, J. Usovitsch, Integral reduction with kira 2.0 and finite field methods, Comput. Phys. Commun. 266 (2021) 108024. 2008.06494. https://doi.org/10.1016/j.cpc.2021.108024

[55] A. van Hameren, PARNI For importance sampling and density estimation, Acta Phys. Polon. B 40 (2009) 259–272. 0710.2448.

[56] A. van Hameren, Kaleu: a general-Purpose parton-Level phase space generator (2010). 1003.4953.

[57] F. Herren, M. Steinhauser, Version 3 of rundec and CRundec, Comput. Phys. Commun. 224 (2018) 333–345. 1703.03751. https://doi.org/10.1016/j.cpc.2017.11.014

[58] https://www.boost.org.

[59] A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page, M. Rüfenacht, M. Schönherr, G. Watt, LHAPDF6: Parton density access in the LHC precision era, Eur. Phys. J. C 75 (2015) 132. 1412.7420. https://doi.org/10.1140/epjc/s10052-015-3318-8

[60] https://lhapdf.hepforge.org/pdfsets.HTML.

[61] H. Frellesvig, D. Tommasini, C. Wever, On the reduction of generalized polylogarithms to Li_n and $\text{Li}_{2,2}$ and on the evaluation thereof, JHEP 03 (2016) 189. 1601.02649. https://doi.org/10.1007/JHEP03(2016)189

[62] R. Kleiss, R. Pittau, Weight optimization in multichannel monte carlo, Comput. Phys. Commun. 83 (1994) 141–146. hep-ph/9405257. https://doi.org/10.1016/0010-4655(94)90043-4

[63] R.L. Workman, et al., Particle Data Group, Review of particle physics, PTEP 2022 (2022) 083C01. https://doi.org/10.1093/ptep/ptac097

[64] J. Davies, K. Schönwald, M. Steinhauser, Three-loop large-N_c virtual corrections to $gg \to HH$ in the forward limit (2025). 2503.17449.

[65] G. Buchalla, M. Capozi, A. Celis, G. Heinrich, L. Scyboz, Higgs boson pair production in non-linear effective field theory with full m_t-dependence at NLO QCD, JHEP 09 (2018) 057. 1806.05162. https://doi.org/10.1007/JHEP09(2018)057

[66] G. Heinrich, S.P. Jones, M. Kerner, G. Luisoni, L. Scyboz, Probing the trilinear higgs boson coupling in di-Higgs production at NLO QCD including parton shower effects, JHEP 06 (2019) 066. 1903.08137. https://doi.org/10.1007/JHEP06(2019)066

[67] 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 (2010) 043. 1002.2581. https://doi.org/10.1007/JHEP06(2010)043