Prepared for submission to JHEP

NNLOPS accurate Drell Yan production

Alexander Karlberg,[a] **Emanuele Re,**[a] **Giulia Zanderighi**[a,b]

[a] *Rudolf Peierls Centre for Theoretical Physics, University of Oxford*
1 Keble Road, UK

[b] *Theory Division, CERN,*
CH–1211, Geneva 23, Switzerland

E-mail: a.karlber1g@physics.ox.ac.uk, e.re1@physics.ox.ac.uk, g.zanderighi1@physics.ox.ac.uk

Abstract: This document contains instructions for the promotion of `VJ-MiNLO` Les Houches event files from NLO to NNLO accuracy in the description of inclusive Drell-Yan production observables, by a reweighting procedure making use of the `DYNNLO` program [1].

Contents

1 Introduction

These instructions explain how to use the code in the `ZJ/DYNNLOPS` and `WJ/DYNNLOPS` directories, upgraded with the `MiNLO` procedure, in conjunction with the `DYNNLO` program to get NNLO accuracy for inclusive Drell Yan production. As there is no fundamental difference between obtaining `NNLOPS`accurate results for Z and W production from this code, we will here give instructions for the case of V production, where V denotes either Z or W.

In order to obtain the needed source code, the user has to download the `DYNNLOPS` directory from the ZJ or WJ main directory, using the following command

```
svn co [-revision n] -username anonymous -password anonymous \
    svn://powhegbox.mib.infn.it/trunk/User-Processes-V2/DYNNLOPS .
```

The `DYNNLOPS` directory contains five folders

- `aux`: a directory containing auxiliary files (e.g. to combine histograms, make plots, etc.);
- `Docs`: a directory containing the manual;
- `COMMON`: a directory containing all files common to W and Z production;
- `WNNLOPS` and `ZNNLOPS`: directories containing files specific of W or Z production.

2 NNLO input

A fundamental ingredient required to obtain NNLO accurate event samples is a triple differential distribution for the Z boson Born kinematics.[1] We here give instructions on how to obtain such distributions, in a format suitable for upgrading the `VJ-MiNLO` events via the `VJ/DYNNLOPS` reweighting code.

1. Make sure the `LHAPDF` package is installed:
   https://lhapdf.hepforge.org
   In particular, for the installation of `DYNNLO`, the command `lhapdf-config --libdir` should return the location of the installed `Lhapdf` libraries.

2. Download `DYNNLO` from the following URL:
   http://theory.fi.infn.it/grazzini/codes/dynnlo-v1.4.tgz

3. Unpack the tarball in a convenient location
   ```
   $ cp  /Downloads/dynnlo-v1.4.tgz ./
   $ tar -xzvf dynnlo-v1.4.tgz
   $ ls dynnlo-v1.4
   ```
   Under the parent directory `dynnlo-v1.4` one should find subdirectories `bin`, `doc`, `obj`, `src`, and a `makefile`.

4. Enter the `DYNNLO` parent directory
   ```
   $ cd dynnlo-v1.4
   ```

5. Replace (or link) the `DYNNLO` default `makefile` with the one from the `Zj/DYNNLOPS/ZNNLOPS/dynnlo-patches` directory
   ```
   $ cp /path/to/Zj/DYNNLOPS/ZNNLOPS/dynnlo-patches/dynnlo.makefile ./makefile
   ```

6. Link (or copy using the `-L` option) the `DYNNLO` patches directory into the parent directory
   ```
   $ cp -r -L /path/to/Zj/DYNNLOPS/ZNNLOPS/dynnlo-patches ./
   ```

7. Build the code
   ```
   $ make
   ```
   A message `--> DYNNLO compiled with LHAPDF routines <--` indicates success. You might need to set the enviromental variable `LD_LIBRARY_PATH` to the output of `lhapdf-config --libdir`, although the patched `dynnlo.makefile` contains commands that should take care of this automatically.

8. Copy over the template input file
   ```
   $ cd bin
   $ cp /path/to/Zj/DYNNLOPS/ZNNLOPS/dynnlo-patches/dynnlo.infile ./
   ```

[1] In this case the Born kinematics is fully specified in terms of the rapidity of the boson, an (arbitrary) angle describing the final state leptons and the invariant mass of the lepton pair.

9. Edit the input file as desired. The input file is nothing but a standard `DYNNLO` input file and may be run simply by typing

   ```
   $ ./dynnlo < dynnlo.infile >> my.log
   ```

   yielding a `nnlo.top` and `nnlo3D.top` histogram file.

   The first file contains one-dimensional distributions which can be used for phenomenological studies. The second file is the one of interest here, as it contains a set of three-dimensional distributions which can be used for reweighting `ZJ-MiNLO` events. As each histogram consists of $25^3 = 15625$ bins, very high statistics are required to properly populate the tails of the distributions. For the analysis presented in [2] we ran 100 instances of `DYNNLO` with the following parameters:

   ```
   15 10000000 !  itmx1, ncall1
   30 10000000 !  itmx2, ncall2
   ```

   Changing the random seed `rseed` for each run. To combine the runs we provide a fortran program, merge3ddata.f, in the `aux` folder. As it can use big amounts of memory, we recommend compiling with the following flags

   ```
   $ gfortran -mcmodel=medium -o merge3ddata merge3ddata.f
   ```

   To then merge the nnlo3D.top files into a file called dynnlo.top do the following

   ```
   $ ./merge3ddata 1 file_1.top ...  file_N.top
   $ mv fort.12 dynnlo.top
   ```

   We would like to note here, that due to the way `DYNNLO` computes the real contribution of the cross-section, the errors of the nnlo3D.top files (and nnlo.top) are not reliable. In order to asses the precision on the three dimensional distributions, we provide a script, 3dto1d.f, which will integrate the three-dimensional distribution into three one-dimensional distributions. After compilation it is run by

   ```
   $ ./3dto1d dynnlo.top
   ```

   which will produce a file fort.12 with one-dimensional distributions.

   In the case where a user might not be interested in high rapidity events, it is presumably reasonable to run with fewer points than suggested here.

10. Finally, make sure that all input parameters (width, mass, couplings, pdfs) should be the same in `DYNNLO` and `ZJ-MiNLO`. (The patched `DYNNLO` program outputs all relevant physical parameters).

3 `Zj-MiNLO` events

The other fundamental ingredient needed for the NNLO reweighting procedure are `ZJ-MiNLO` Les Houches events. A few modifications of the standard code are needed to obtain three-dimensional distributions and to maintain consistency of the physical parameters in `DYNNLO` and `ZJ-MiNLO`. In order to include them, it is sufficient to replace the default `POWHEG` file with the patched one:

```
$ cp DYNNLOPS/ZNNLOPS/powheg-patches/powheg.makefile ./makefile
```

In `DYNNLOPS/ZNNLOPS/powheg-patches/powheg.input` a template input file can also be found. If the code is run with the patches out of the box and the above input file, consistency between `DYNNLO` and `ZJ-MiNLO` should be achieved. However, we stress again that it is up to the user to make sure this is the case.

Few comments are helpful:

1. A large number of physical parameters used by the `ZJ-MiNLO` code, such as the Fermi constant, are assigned by the subroutine `init_couplings` defined in file `Zj/init_couplings.f`. As in the case of the `DYNNLO` `mdata.f` file, some of the parameters in this file are irrelevant. Nevertheless, in `Zj/DYNNLOPS/ZNNLOPS/powheg.input` all these parameters can be changed. The version we provide is consistent with our version of mdata.f

2. For detailed instructions on setting up the `ZJ-MiNLO` program to perform numerous runs in parallel see section 4.1 in `/path/to/W2jet/Docs/manual-BOX-WZ2jet.pdf`

3. We add that although it is not strictly necessary to generate the `ZJ-MiNLO` events using a NNLO PDF, in the limited studies that we have carried out to date we used the same (NNLO) PDF set in `ZJ-MiNLO` and `DYNNLO`.

4. As was the case when obtaining NNLO distributions, a large number of events is needed to populate the tails of the three-dimensional distributions. For the study carried out in [2] we used 20 million events to compute denominators.

4 Reweighting

The NLO-to-NNLO weight factor assigned to the `VJ-MiNLO` events is differential in the vector boson phase space (Φ_B) and the transverse momentum of the leading jet (p_{T}), at the NNLO level in the `DYNNLO` case, and at the level of the hardest emission cross section (Les Houches events) in `VJ-MiNLO`. It is given by

$$\begin{aligned} \mathcal{W}\left(\Phi_B,\, p_{\mathrm{T}}\right) = h\left(p_{\mathrm{T}}\right) & \frac{\int d\sigma^{\mathrm{NNLO}}\;\; \delta\left(\Phi_B - \Phi_B\left(\Phi\right)\right) - \int d\sigma_B^{\mathrm{MINLO}}\, \delta\left(\Phi_B - \Phi_B\left(\Phi\right)\right)}{\int d\sigma_A^{\mathrm{MINLO}}\, \delta\left(\Phi_B - \Phi_B\left(\Phi\right)\right)} \\ & + \left(1 - h\left(p_{\mathrm{T}}\right)\right)\,, \end{aligned} \tag{4.1}$$

where

$$d\sigma_A = d\sigma\, h\left(p_{\mathrm{T}}\right)\,, \tag{4.2}$$

$$d\sigma_B = d\sigma\, \left(1 - h\left(p_{\mathrm{T}}\right)\right)\,, \tag{4.3}$$

with h a monotonic profile function

$$h(p_{\mathrm{T}}) = \frac{(\beta\, m_{\mathrm{V}})^2}{(\beta\, m_{\mathrm{V}})^2 + p_{\mathrm{T}}^2}, \tag{4.4}$$

and β a constant parameter. On convoluting $\mathcal{W}(\Phi_B, p_{\rm T})$ with the ZJ-MiNLO differential cross section and integrating over $p_{\rm T}$ one finds, exactly,

$$\left(\frac{d\sigma}{d\Phi_B}\right)^{\rm NNLOPS} = \left(\frac{d\sigma}{d\Phi_B}\right)^{\rm NNLO} . \tag{4.5}$$

For a proof of why such a reweighting procedure leads to NNLO accuracy in general, we refer the reader to [2, 3].

The role of the profile function h and, in particular, the β parameter within it, is, roughly speaking, to determine how to spread out the NLO-to-NNLO corrections along the $p_{\rm T}$ axis. For $\beta = \infty$ the corrections are spread uniformly in $p_{\rm T}$ (see e.g. fig. 3 of ref. [3]), while for $\beta = 1$ they are concentrated in the region $p_{\rm T} < \beta\, m_Z$. In the latter respect the β parameter plays a similar role to the resummation scale in dedicated resummation calculations, and as such we favour that β be set consistently with the preferred values in those. Thus we recommend $\beta = 1$ in carrying out the reweighting. Indeed for $\beta = 1$ we find good agreement with dedicated NNLL+NNLO calculations of the Z boson transverse momentum and the 0-jet veto efficiency where the resummation scale was set to m_Z.

We should emphasise that, while our NNLOPS simulation is formally NNLO accurate for inclusive quantities, the accuracy of its resummation of all-orders large logarithms is, formally, categorically not at the NNLL+NNLO level. We recommend that the predictions of such calculations be used to 'tune' the NNLOPS simulation (in particular the β parameter) to approximate the yet higher order, large logarithmic, terms which it does not take into account. Setting $\beta = 1$ appears to do a very good job in this respect, thus it is the default, recommended, value in the following. If the reweighting is performed using eq. (4.1), the value of β can be changed by just modifying in a straightforward way the file `get_zdamp.f`.

4.1 A simple run

The code to be used in order to produce an NNLO-reweighted event file can be found in the `DYNNLOPS/ZNNLOPS/Reweighter` directory. After compiling it (`FastJet` needs to be linked), the program has to be run with the following command line:

```
$ ./minnlo <ZjMiNLO-eventfile> <nr-DYNNLOfiles> <DYNNLO-file1>
   [ <DYNNLO-file2>...]  [<ZjMiNLO-file1> <ZjMiNLO-file2> ....  ]
```

where `ZjMiNLO-eventfile` is a LH file (suffix `.lhe`) containing events produced with `ZJ-MiNLO`, obtained as described in sec. 3, `nr-DYNNLOfiles` denotes the number of DYNNLO files containing the 3D distributions (more then one is needed for instance when commuting uncertainty bands using scale variation), `DYNNLO-file*` are output files of `DYNNLO`, computed as described in sec 2. The number of files given in the input line must correspond to `nr-DYNNLOfiles`. Finally, `ZjMiNLO-file*` contain three-dimensional histograms computed using `ZJ-MiNLO` events. These last arguments are optional. If these files are not present in the command line, the reweighing program computes them and stops. We call this stage one. When the `ZjMiNLO-file*` are computed by the reweighing program using the `ZjMiNLO-eventfile`, the first weight present in the LHE file will produce a file called `MINLO-W1-denom.top`, the second weight a file called `MINLO-W2-denom.top` and so on.

Rather then producing just one huge event file, it is also possible to compute the `MINLO-W*-denom.top` files on each event sample and then combine the resulting distributions using the program merge3ddata (in `aux`).[2] With the generated `MINLO-W*-denom.top` files one can then go on and perform the actual NNLO reweighing including in the command line these files. We call this stage two. The files *have* to be given in the same order as the corresponding weights in the LH file. Otherwise the reweighting will be incorrect.

Here follows an example on how to go through both stages, assuming that the user has a `ZJ-MiNLO` event file called `pwgevents.lhe` containing 7 weights and 3 `DYNNLO` files called `dynnlo-mur0.5-muf0.5.top`, `dynnlo-mur1.0-muf1.0.top` and `dynnlo-mur2.0-muf2.0.top`, the program should be called as

```
$./minnlo pwgevents.lhe 3 dynnlo-mur0.5-muf0.5.top \
    dynnlo-mur1.0-muf1.0.top dynnlo-mur2.0-muf2.0.top
```

This will result in an output of `MINLO-W1-denom.top`, `MINLO-W2-denom.top`, `MINLO-W3-denom.top`, `MINLO-W4-denom.top`, `MINLO-W5-denom.top`, `MINLO-W6-denom.top`, `MINLO-W7-denom.top`.

To perform stage two the user would run

```
$./minnlo pwgevents2.lhe 3 dynnlo-mur0.5-muf0.5.top \
    dynnlo-mur1.0-muf1.0.top dynnlo-mur2.0-muf2.0.top MINLO-W*-denom.top
```

which will produce a file called `pwgevents.lhe-nnlo`. This LH file is the final output of the `MiNNLO` NLO-to-NNLO reweighter program, and can now be read and showered by `Pythia` or `Herwig`, as it is usually done with LH event files generated by the `POWHEG BOX`. It is important to notice that the new NNLO weight attached to each event is written *after* the event record, in the line comprised between the partonic momenta and the end-of-event tag (`</event>`). Similarly to the format of a `POWHEG BOX` output when the reweighting machinery is used, this line starts with the text

```
#new weight,renfact,facfact,pdf1,pdf2
```

followed by a weight and other information. In a LH file obtained with the `MiNNLO` reweighter, the names of the `DYNNLO-output` files are appended at the end of this line, and the NNLO weight to be used is the first number appearing, which will obviously be different from the original weight contained in `ZJMiNLO-eventfile`. This different weight, together with the appended identifier at the end of each line starting with `#new weight`, should be the only differences between the input and output LH files. Notice that the NNLO-reweighted output file will always contain the `#new weight` line(s), even if the original LH file `ZJMiNLO-eventfile` didn't.

The program takes few hours to reweight a LH file produced by `ZJ-MiNLO` containing 20 million events. The output printed on the terminal at run time is self explanatory. We

[2] In this case, care has to be take to produce files in different directories.

have also included a template script (`runminnlo_template.sh`) to help the user.

4.2 Estimating uncertainties

The conservative ansatz we recommend in estimating errors (the one employed in ref. [2, 3]) is that the μ_{R} and μ_{F} scales in the NNLO and NLO inputs should be varied in a fully independent way. In doing so we regard the uncertainties in the normalisations of distributions, as being independent of the respective uncertainties in the shapes — at least in the region covered by the profile function, $h(p_{\mathrm{T}})$, i.e. that which includes the low p_{T} domain. The former are determined by the `DYNNLO` program, while the latter are due to `ZJ-MiNLO`. Outside of the low p_{T} region, the uncertainty is given by the standard `ZJ-MiNLO` computation (which uses in that region $\mu_{\mathrm{R}} = \mu_{\mathrm{F}} = p_{\mathrm{T}}$ for the central scale choice).

In order to compute an uncertainty band, one first needs to have obtained multiple outputs from `DYNNLO` and `ZJ-MiNLO` with different μ_{R} and μ_{F}. For the sake of simplicity, we will now describe a case where μ_{R} and μ_{F} are kept equal both when running `DYNNLO` and `ZJ-MiNLO`. We call this situation a "3x3 pts" scale variation study: for each event we will obtain 9 NNLOPS weights, associated to each of the 9 possible combinations among 3 results from `DYNNLO` and 3 from `ZJ-MiNLO`. This procedure is general enough to be straightforwardly adapted to the more general case of a "7x7 pts" scale variation, or variation thereof.

In the "3x3 pts" case, the needed inputs are 3 histogram files from `DYNNLO`, which we will call `DYNNLO-outputfile-11.top`, `DYNNLO-outputfile-HH.top` and `DYNNLO-outputfile-22.top`, for values of $\mu_{\mathrm{R}} = \mu_{\mathrm{F}} = \{m_Z/2, m_Z, 2m_Z\}$ respectively. Similarly, the user needs to have obtained a LH file from `ZJ-MiNLO` where 3 weights are associated to each event. This file has to be obtained with the `POWHEG BOX` reweighting machinery: the 3 lines in between the last line of each event record and the `</event>` tag should have the format

```
#new weight,renfact,facfact,pdf1,pdf2 <weight> <ren.scale factor>
 <fact.scale fact> <pdf number for hadron 1>
 <pdf number for hadron 2> <'mlm' or 'lha'>
```

The NLO-to-NNLO reweighter program should now be invoked in a first steep as:

```
$ ./minnlo ZJMiNLO-events.lhe 3 DYNNLO-outputfile-HH.top \
 DYNNLO-outputfile-11.top  DYNNLO-outputfile-22.top
```

This will produce the three `MiNLO` files with three-dimensional distributions.

In a second step, one calls

```
$ ./minnlo ZJMiNLO-events.lhe 3 DYNNLO-outputfile-HH.top \
 DYNNLO-outputfile-11.top  DYNNLO-outputfile-22.top \
 MINLO-W1-denom.top MINLO-W2-denom.top MINLO-W3-denom.top
```

At the end of the run, 9 lines will be present after each event record, each one containing the NNLO weight associated to the `ZJ-MiNLO` result labelled by the values of the pair (`<ren.scale factor>`, `<fact.scale fact>`) and the `DYNNLO` result identified by the name of the file appended to each line.

5 Example analysis

The analysis used for the phenomenological study in ref. [2] can be found in `pwhg_analysis-minlo.f`. The analysis is compiled in the `Zj` directory in the usual way by

```
make lhef_analysis
```

It can also be used to analyse events after the showering stage. We provide drivers for both Pythia6 (version 6.4.28) and Pythia8 (version 8.185) which can be compiled by

```
make main-PYTHIA-lhef
make main-PYTHIA8-lhef
```

Note that `POWHEG BOX` ships with Pythia6.4.25 as standard. It is straightforward to use any other version, by downloading it, and putting it in the main `POWHEG BOX` directory. For Pythia8 the user has to compile it himself and then set the `PYTHIA8LOCATION` appropriately.

6 Suggested citations

If you use this code please cite the inclusive Drell Yan production `NNLOPS` paper [2] and the `ZJ-MiNLO` paper which precedes and lays much of the foundations for it [4]. The `NNLOPS` simulation fundamentally relies on the NNLO Drell Yan boson production calculation of refs. [1], and so these works should also be cited.

A Code description

In this section we briefly record the various additions and modifications to existing `DYNNLO` and `POWHEG BOX` code used to produce `NNLOPS` events.

A.1 DYNNLO

- `makefile` (`HNNLO-makefile`)
  - The `DYNNLO` `makefile` is modified by prepending `$(DYNNLOHOME)/dynnlo-patches` to the `$DIRS` variable, introducing a variable `$PATCHES` equal to the concatenation of the following object files in this list, plus the removal of those elements from the other `Makefile` variables (avoiding duplication). In this way the modified `DYNNLO` files below are compiled and linked from `dynnlo-patches` instead of the default versions in the default locations.
- `mbook.f`
  - The `mtop` subroutine, which outputs the `DYNNLO` histograms to a text file, has undergone a minor modification so as to have the same format as the `ZJ-MiNLO` histogram output, to ease comparisons and for use in the reweighting program.
- `mdata.f`

– This file contains the values of numerous physical constants in `DYNNLO` e.g. Fermi's constant G_F, etc. We have edited all parameters in this file to have agreement with the corresponding `ZJ-MiNLO` default settings.

- `plotter.f`
  – This file contains an example analysis for `DYNNLO` by default. We have modified this analysis to produce three-dimensional histograms. Also a number of distributions relevant for phenomenology is being filled. The three-dimension histogram range has been set to $-5 < y < 5$, $-\pi/2 < a_{mll} < \pi/2$ and $0 < \theta_l < \pi$. The distributions have 25 bins in each direction. These values may be altered by the user as desired, by editing the relevant booking subroutine call. However, in this case one must take care to modify the relevant `POWHEG BOX` analysis file (`pwhg_analysis-release.f`) used by the reweighting code, under the `Zj/DYNNLOPS/ZNNLOPS/Reweighter` directory.
- `setup.f`
  – This file was modified to use the k_T-algorithm instead of the anti-k_T.
- `writeinfo.f`
  – Originally the `writeinfo` subroutine in this file copied the contents of the input file used in running the program, plus the cross section, as a series of comments to the top of the histogram output file. In order to have a simple uniform format for the `ZJ-MiNLO` and `DYNNLO` we removed these comments (the bulk of which was simply a copy of the input file used to run the program).
- `histofinLH.f`
  – This file takes care of finalising the histograms. As we are using the `POWHEG BOX` routines to fill our histograms (and a modified version to fill three-dimensional histograms) this file had to be modified to fill the correct histograms and finalise them.
- `coupling.f`
  – We added a printout to the screen of various parameters, to make it easier to compare with the output of `ZJ-MiNLO`.
- `boost.f`
  – Added a routine to go from the lab frame to the CM frame.
- `auxiliary.f`,`pwhg_bookhist-multi.f`,`pwhg_bookhist-new.f`,`pwhg_book-multi.h`, `pwhg_bookhist-new.h`,`pwhg_math.h`
  – These files contains the routines to fill histograms, to have a consistent output between `DYNNLO` and `ZJ-MiNLO`.

A.2 `ZJ-MiNLO`

- `lhapdfif.f`
  - When using NNLO PDFs Λ_5 is now computed at NLL and not NNLL.
- `setlocalscales2.f`
  - This file is used by the Makefile instead of the default `setlocalscales.f`. We have changed the routine which computes α_S to be the default `POWHEG BOX` one, instead of a locally defined prescription. The two prescriptions agree far away from Λ_5 but significant differences show up close to the Landau pole.
- `powheg.input` / `powheg.input-save`
  - Here all values should be set consistently with the values used by `DYNNLO`.

The rest of the files found in this folder are either a driver for Pythia (as used in [2] or files already described in the section above)

A.3 NNLO reweighting

- `Makefile`
  - The default value for the `ANALYSIS` variable should be `release`. `FastJet` must be linked properly too, and as usual it is recommended to let the `Makefile` call the `fastjet-config` command.
- `minnlo.f`
  - This file contains the main program to perform the NLO-to-NNLO reweighting. Some parameters useful for debugging purposes can be found here, as described in the commented section at the beginning of the file. However, the user is recommended not to change them.
- `opencount.f`, `auxilliary.f`, `lhef_readwrite.f`, `get_zdamp.f`, `genclust_kt.f`, `swapjet.f`, `miscclust.f`, `ptyrap.f`, `r.f`
  - These files contain several functions and routines needed by `minnlo.f` and/or by the analysis subroutine used to process the `ZJ-MiNLO` LH file and compute $d\sigma_{A/B}^{\mathrm{MINLO}}\,\delta\left(\Phi_B - \Phi_B\left(\Phi\right)\right)$.
    [`jetlabel.f`, `jetcuts.f`, `mxpart.f` and `npart.f` contain common blocks used in the source files.]
- `pwhg_analysis-release.f`, `jet_finder-release.f`
  - These files contain the minimal analysis needed to extract $d\sigma_{A/B}^{\mathrm{MINLO}}\,\delta\left(\Phi_B - \Phi_B\left(\Phi\right)\right)$ from the LH input file. They are compiled if `ANALYSIS=release` is set, which is the default option.

References

[1] S. Catani, L. Cieri, G. Ferrera, D. de Florian, and M. Grazzini, *Vector boson production at hadron colliders: A Fully exclusive QCD calculation at NNLO*, *Phys.Rev.Lett.* **103** (2009) 082001, [arXiv:0903.2120].

[2] A. Karlberg, E. Re, and G. Zanderighi, *NNLOPS accurate Drell-Yan production*, *JHEP* **1409** (2014) 134, [arXiv:1407.2940].

[3] K. Hamilton, P. Nason, E. Re, and G. Zanderighi, *NNLOPS simulation of Higgs boson production*, *JHEP* **1310** (2013) 222, [arXiv:1309.0017].

[4] K. Hamilton, P. Nason, C. Oleari, and G. Zanderighi, *Merging H/W/Z + 0 and 1 jet at NLO with no merging scale: a path to parton shower + NNLO matching*, *JHEP* **1305** (2013) 082, [arXiv:1212.4504].