Diboson production in POWHEG-BOX: theory and tutorial

Emanuele Re

University & INFN Milano-Bicocca

COMETA workshop on vector-boson polarisations
Toulouse, 24 September 2024

Outline & introduction

- (N)NLO+PS with POWHEG: theory
  - EW corrections
  - diboson production: what is available

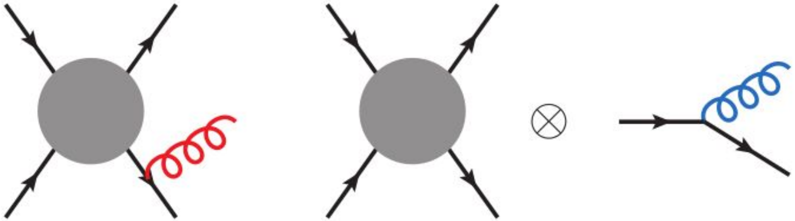
- Tutorial:
  - how the POWHEG BOX code works
  - how to run it
  - live demo

- main theory focus here: EW/QCD corrections matched to PS, no polarized bosons so far
  → 2 papers/codes out: [Hoppe et al. ‘23 / Pelliccioli,Zanderighi ‘23]
- I won’t discuss PS uncertainties and recent work on NLL showers
- this talk and codes: “VV” stands for “4 leptons”

Part 1: theory

The POWHEG method in a nutshell (I)

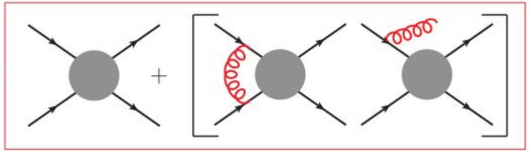
- POWHEG: method to achieve NLO+PS. Match fixed-order computation at NLO in QCD with Parton Showers
- Problem: overlapping regions



- NLO+PS is well understood, general solutions applicable to virtually any process: `MC@NLO` and `POWHEG` [Frixione-Webber '03, Nason '04]
- Other approaches exist, e.g.
  `KrkNLO`, `Vincia` [Jadach et al., Skands et al.]
  `Geneva`, `U(N)NLOPS`, `MAcNLOPS` [Alioli et al., Prestel et al./Plätzer, Nason,Salam]

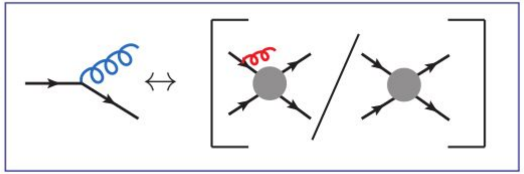
The POWHEG method in a nutshell (II)

$$B(\Phi_n) \Rightarrow \bar{B}(\Phi_n) = B(\Phi_n) + \frac{\alpha_s}{2\pi}\left[V(\Phi_n) + \int R(\Phi_{n+1})\, d\Phi_r\right]$$



$$d\sigma_{\mathrm{POW}} = d\Phi_n \quad \bar{B}(\Phi_n) \quad \left\{\Delta(\Phi_n; k_{\mathrm{T}}^{\mathrm{min}}) + \Delta(\Phi_n; k_{\mathrm{T}})\frac{\alpha_s}{2\pi}\frac{R(\Phi_n, \Phi_r)}{B(\Phi_n)}\, d\Phi_r\right\}$$

[+ p_{T}-*vetoing subsequent emissions*, to avoid double-counting]



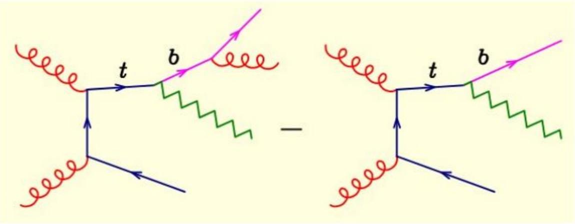
$$\Delta(t_{\mathrm{m}}, t) \Rightarrow \Delta(\Phi_n; k_{\mathrm{T}}) = \exp\left\{-\frac{\alpha_s}{2\pi}\int \frac{R(\Phi_n, \Phi'_r)}{B(\Phi_n)}\theta(k'_{\mathrm{T}} - k_{\mathrm{T}})\, d\Phi'_r\right\}$$

The POWHEG BOX framework

- Main focus: matching of accurate fixed-order predictions with PS for SM processes.
- All publicly available at

  `powhegbox.mib.infn.it`
- Two main releases:
  - POWHEG BOX V2: main release, almost all processes are here
  - POWHEG BOX RES: most recent one, able to deal with processes with resonances

POWHEG BOX RES: resonances



$$d\sigma = d\Phi_{\rm rad}\,\bar{B}(\Phi_B)\,\frac{R(\Phi_B,\Phi_{\rm rad})}{B(\Phi_B)} \times \exp\left[-\int \frac{R(\Phi_B,\Phi_{\rm rad})}{B(\Phi_B)}\,d\Phi_{\rm rad}\right]$$

$\Phi_B \to (\Phi_B, \Phi_{\rm rad})$ mapping does not preserve virtuality

⇒ *R/B* can become large also far from collinear singularity

- NW limit spoiled
- BW shape distorted

“Resonant-aware” NLO+PS methods

[Ježo,Nason ‘15]

- project full MEs onto “resonance histories”: $B = \sum_{f_b} B_{f_b}$, where $B_{f_b} \equiv \frac{P^{f_b}(\Phi_B)}{\sum_{f'_b} P^{f'_b}(\Phi_B)} B(\Phi_B)$

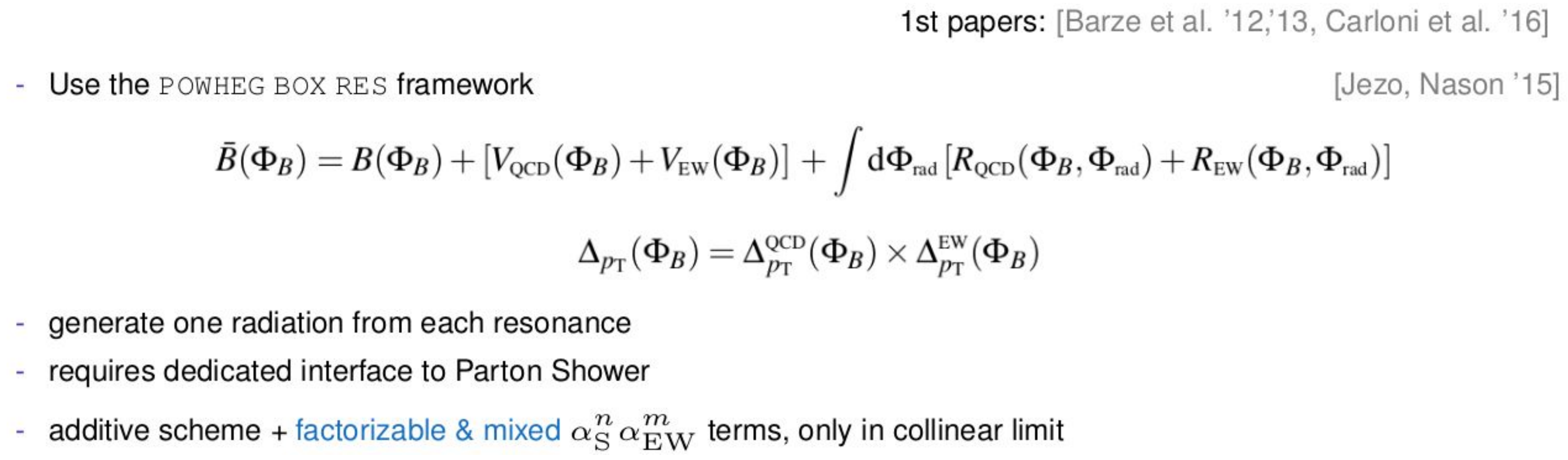
$P^{f_b}(\Phi_B)$ (products of) Breit-Wigner functions ⇔ resonance history f_b

- each term (Born- and real-like) is attributed to an unique resonance history
  - virtuality-preserving mappings
  - POWHEG radiation can be assigned a resonance → (up to) 1 emission per resonance

POWHEG BOX RES: NLOQCD + NLOEW + PS

- $\mathrm{NLO}_{\mathrm{EW}}$+PS not conceptually solved in full generality
  - bottleneck: processes with “QCD/EW interference” at LO
  - possible for some processes, e.g. DY, dibosons

Exact matching of EW corrections for *n*- and *n+1*-body contributions



1st papers: [Barze et al. ’12,’13, Carloni et al. ’16]

- Use the `POWHEG BOX RES` framework [Jezo, Nason ’15]

$$\bar{B}(\Phi_B) = B(\Phi_B) + [V_{\mathrm{QCD}}(\Phi_B) + V_{\mathrm{EW}}(\Phi_B)] + \int d\Phi_{\mathrm{rad}} \, [R_{\mathrm{QCD}}(\Phi_B, \Phi_{\mathrm{rad}}) + R_{\mathrm{EW}}(\Phi_B, \Phi_{\mathrm{rad}})]$$

$$\Delta_{p_\mathrm{T}}(\Phi_B) = \Delta^{\mathrm{QCD}}_{p_\mathrm{T}}(\Phi_B) \times \Delta^{\mathrm{EW}}_{p_\mathrm{T}}(\Phi_B)$$

- generate one radiation from each resonance
- requires dedicated interface to Parton Shower
- additive scheme + factorizable & mixed $\alpha_{\mathrm{S}}^{n}\,\alpha_{\mathrm{EW}}^{m}$ terms, only in collinear limit

NNLO+PS

- Consider $F+X$ production (F=massive color singlet)
- NNLO accuracy for observables inclusive on radiation. $[d\sigma/dy_F]$
- NLO(LO) accuracy for $F+1(2)$ jet observables (in the hard region). $[d\sigma/dp_{T,j_1}]$
  - appropriate scale choice for each kinematics regime
- Sudakov resummation from the Parton Shower (PS) $[\sigma(p_{T,j} < p_{T,\mathrm{veto}})]$
- preserve the PS accuracy (leading log - LL)
- possibly, no merging scale required.

- methods: reweighted `MiNLO′` ("NNLOPS") [Hamilton,et al. '12,'13,...],
  `UNNLOPS` [Höche,Li,Prestel '14,...],
  `Geneva` [Alioli,Bauer,et al. '13,'15,'16,...],
  MiNNLO$_{\mathrm{PS}}$ [Monni,Nason,ER,Wiesemann,Zanderighi '19,...],
  `Vincia`+sector showers [Campbell et al, '21]

[Notation: From this point, $X = \sum_k \left(\frac{\alpha_S}{2\pi}\right)^k [X]^{(k)}$]

NNLO+PS: MiNNLO PS (I)

- from p_T resummation, differential cross section for F+X production can be written as:

$$\frac{\mathrm{d}\sigma}{\mathrm{d}p_{\mathrm{T}}\mathrm{d}\Phi_{\mathrm{F}}} = \frac{\mathrm{d}}{\mathrm{d}p_{\mathrm{T}}}\left\{\mathcal{L}(\Phi_{\mathrm{F}}, p_{\mathrm{T}})\exp(-\tilde{S}(p_{\mathrm{T}}))\right\} + R_{\mathrm{finite}}(p_{\mathrm{T}})$$

$$\mathcal{L}(\Phi_{\mathrm{F}}, p_{\mathrm{T}}) \ni \{H^{(1)}, H^{(2)}, C^{(1)}, C^{(2)}, (G^{(1)}\cdot G^{(1)})\} \qquad R_{\mathrm{finite}}(p_{\mathrm{T}}) = \frac{\mathrm{d}\sigma_{\mathrm{FJ}}}{\mathrm{d}\Phi_{\mathrm{F}}\mathrm{d}p_{\mathrm{T}}} - \frac{\mathrm{d}\sigma^{\mathrm{sing}}}{\mathrm{d}\Phi_{\mathrm{F}}\mathrm{d}p_{\mathrm{T}}}$$

- recast it, to match the `POWHEG` $\bar{B}^{(\mathrm{FJ})}(\Phi_{\mathrm{FJ}})$

$$\frac{\mathrm{d}\sigma}{\mathrm{d}\Phi_{\mathrm{F}}\mathrm{d}p_{\mathrm{T}}} = \exp[-\tilde{S}(p_{\mathrm{T}})]\left\{D(p_{\mathrm{T}}) + \frac{R_{\mathrm{finite}}(p_{\mathrm{T}})}{\exp[-\tilde{S}(p_{\mathrm{T}})]}\right\}$$

$$D(p_{\mathrm{T}}) \equiv -\frac{\mathrm{d}\tilde{S}(p_{\mathrm{T}})}{\mathrm{d}p_{\mathrm{T}}}\mathcal{L}(p_{\mathrm{T}}) + \frac{\mathrm{d}\mathcal{L}(p_{\mathrm{T}})}{\mathrm{d}p_{\mathrm{T}}} \qquad \tilde{S}(p_{\mathrm{T}}) = \int_{p_{\mathrm{T}}}^{Q}\frac{dq^2}{q^2}\left[A_{\mathrm{f}}(\alpha_{\mathrm{S}}(q))\log\frac{Q^2}{q^2} + B_{\mathrm{f}}(\alpha_{\mathrm{S}}(q))\right]$$

- expand the above integrand in power of $\alpha_{\mathrm{S}}(p_{\mathrm{T}})$, keep the terms that are needed to get NLO$^{(\mathrm{F})}$ & NNLO$^{(\mathrm{F})}$ accuracy, when integrating over p_{T}

NNLO+PS: MiNNLO PS (II)

$$\frac{\mathrm{d}\bar{B}(\Phi_{\mathrm{FJ}})}{\mathrm{d}\Phi_{\mathrm{FJ}}} = \exp[-\tilde{S}(p_{\mathrm{T}})]\Bigg\{ \frac{\alpha_{\mathrm{S}}(p_{\mathrm{T}})}{2\pi}\left[\frac{\mathrm{d}\sigma_{\mathrm{FJ}}}{\mathrm{d}\Phi_{\mathrm{FJ}}}\right]^{(1)}\left(1+\frac{\alpha_{\mathrm{S}}(p_{\mathrm{T}})}{2\pi}[\tilde{S}(p_{\mathrm{T}})]^{(1)}\right) + \left(\frac{\alpha_{\mathrm{S}}(p_{\mathrm{T}})}{2\pi}\right)^2\left[\frac{\mathrm{d}\sigma_{\mathrm{FJ}}}{\mathrm{d}\Phi_{\mathrm{FJ}}}\right]^{(2)} + [D(p_{\mathrm{T}})]^{(\geq 3)}\, F_\ell^{\mathrm{corr}}(\Phi_{\mathrm{FJ}})\Bigg\}$$

- $$[D(p_{\mathrm{T}})]^{(\geq 3)} = \underbrace{-\frac{\mathrm{d}\tilde{S}(p_{\mathrm{T}})}{\mathrm{d}p_{\mathrm{T}}}\mathcal{L}(p_{\mathrm{T}}) + \frac{\mathrm{d}\mathcal{L}(p_{\mathrm{T}})}{\mathrm{d}p_{\mathrm{T}}}}_{D} - \frac{\alpha_{\mathrm{S}}(p_{\mathrm{T}})}{2\pi}[D(p_{\mathrm{T}})]^{(1)} - \left(\frac{\alpha_{\mathrm{S}}(p_{\mathrm{T}})}{2\pi}\right)^2 [D(p_{\mathrm{T}})]^{(2)}$$
- $F_\ell^{\mathrm{corr}}(\Phi_{\mathrm{FJ}})$: projection → recover $[D(p_{\mathrm{T}})]^{(\geq 3)}$ when integrating over Φ_{FJ} at fixed $(\Phi_{\mathrm{F}}, p_{\mathrm{T}})$

- The second radiation is generated by the usual POWHEG mechanism.

$$\mathrm{d}\sigma = \bar{B}(\Phi_{\mathrm{FJ}})\, \mathrm{d}\Phi_{\mathrm{FJ}} \left\{ \Delta_{\mathrm{pwg}}(\Lambda_{\mathrm{pwg}}) + \mathrm{d}\Phi_{\mathrm{rad}}\Delta_{\mathrm{pwg}}(p_{\mathrm{T,rad}})\frac{R(\Phi_{\mathrm{FJ}}, \Phi_{\mathrm{rad}})}{B(\Phi_{\mathrm{FJ}})} \right\}$$

- if emissions are strongly ordered, same emission probabilities as in k_t-ordered shower → LL shower accuracy preserved

Diboson production in POWHEG-BOX: overview

Dibosons

- NLO$_{\mathrm{QCD}}$ + PS	(WW/WZ/ZZ)	[V2]
- NLO$_{\mathrm{QCD}}$ + PS	(WW/WZ, anomalous couplings)	[V2]
- gg → VV + PS NLO$_{\mathrm{QCD}}$ + PS		[RES]
- NLO$_{\mathrm{QCD}}$ + NLO$_{\mathrm{EW}}$ + PS	(WW/WZ/ZZ)	[RES]
- NNLO$_{\mathrm{QCD}}$ + PS	(WW/WZ/ZZ/Zγ)	[RES]
- with nNNLO	(ZZ)	[RES]
- with NLO$_{\mathrm{EW}}$	(WZ)	[RES]
- **NLO$_{\mathrm{QCD}}$ + PS (polarized bosons)**	(WW/WZ/ZZ)	[RES]

Vector-boson scattering

- NLO$_{\mathrm{QCD}}$ + PS	({WW/ZZ} + 2jets, EW production)	[V2]
- NLO$_{\mathrm{EW}}$ + PS	(same sign WW + 2jets, VBS)	[RES]

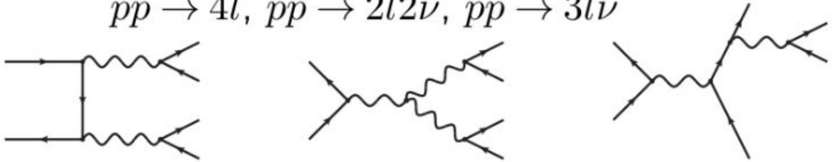
→ Authors and codes: see `powhegbox.mib.infn.it`

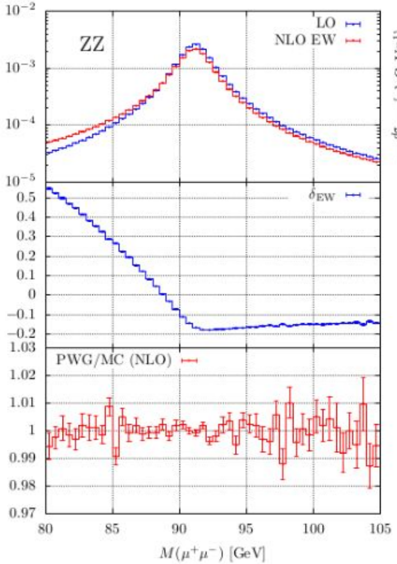
Diboson production in POWHEG-BOX: EW+QCD

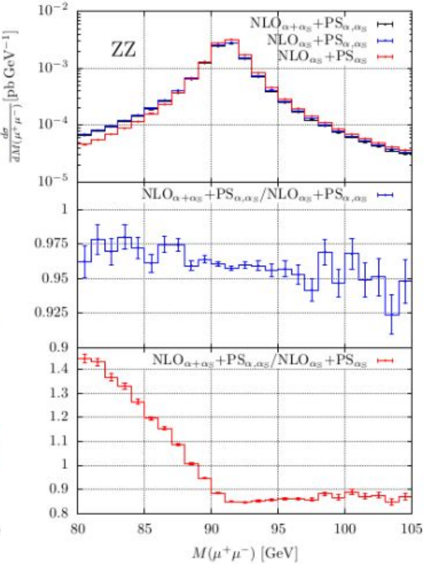
[Chiesa,ER,Oleari '20]

loop amplitudes from `Recola2`

$$pp \to 4l,\ pp \to 2l2\nu,\ pp \to 3l\nu$$







- possible to have control on few percent effects

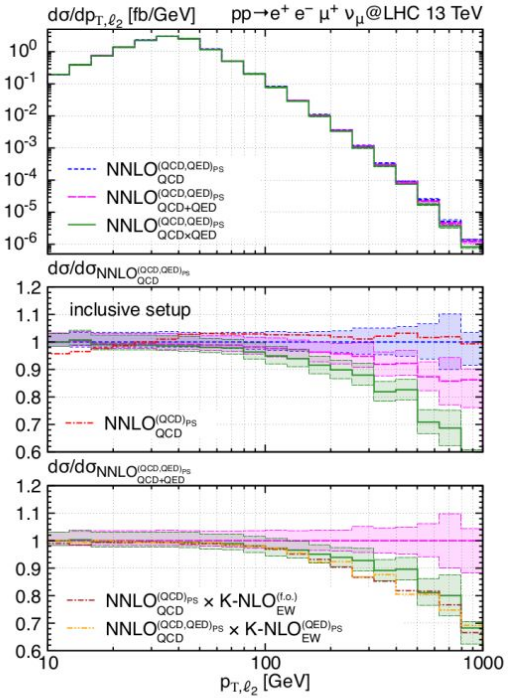
- $\mathrm{NLO}_{\alpha_S+\alpha}$ + $\mathrm{PS}_{\alpha_S,\alpha}$ / NLO_{α_S} + $\mathrm{PS}_{\alpha_S,\alpha}$:

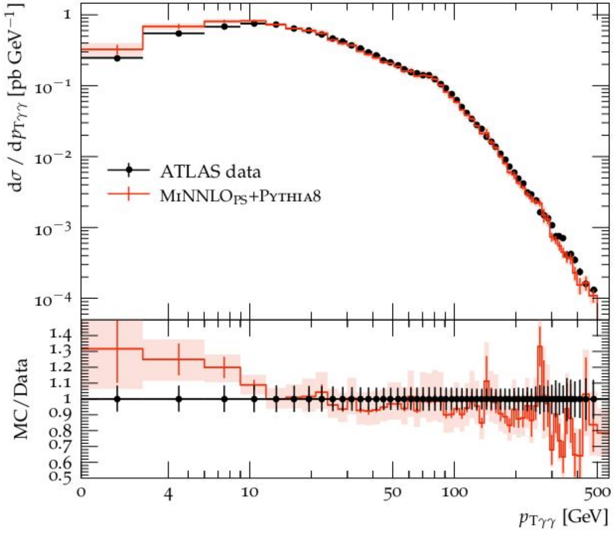
  - NLO weak, non-log QED $\mathcal{O}(\alpha)$, mixed

- $\mathrm{NLO}_{\alpha_S+\alpha}$ + $\mathrm{PS}_{\alpha_S,\alpha}$ / NLO_{α_S} + PS_{α_S}:

  - NLO weak, QED $\mathcal{O}(\alpha)$, leading-log QED $\mathcal{O}(\alpha^n)$ $(n > 2)$, mixed

Diboson production in POWHEG-BOX: NNLO QCD





Zγ [Lombardi et al. ‘20]

WW [Lombardi et al. ‘20]

ZZ [Buonocore et al. ‘21]

WZ [Lindert et al. ‘22]

γγ [Gavardi et al. ‘22]

$$\mathrm{NNLO}_{\mathrm{QCD}}^{(\mathrm{QCD,QED})_{\mathrm{PS}}} \times \text{K-NLO}_{\mathrm{EW}}^{(\mathrm{QCD,QED})_{\mathrm{PS}}}$$

$$\mathrm{NNLO}_{\mathrm{QCD}}^{(\mathrm{QCD,QED})_{\mathrm{PS}}} + \delta\mathrm{NLO}_{\mathrm{EW}}^{(\mathrm{QCD,QED})_{\mathrm{PS}}}$$

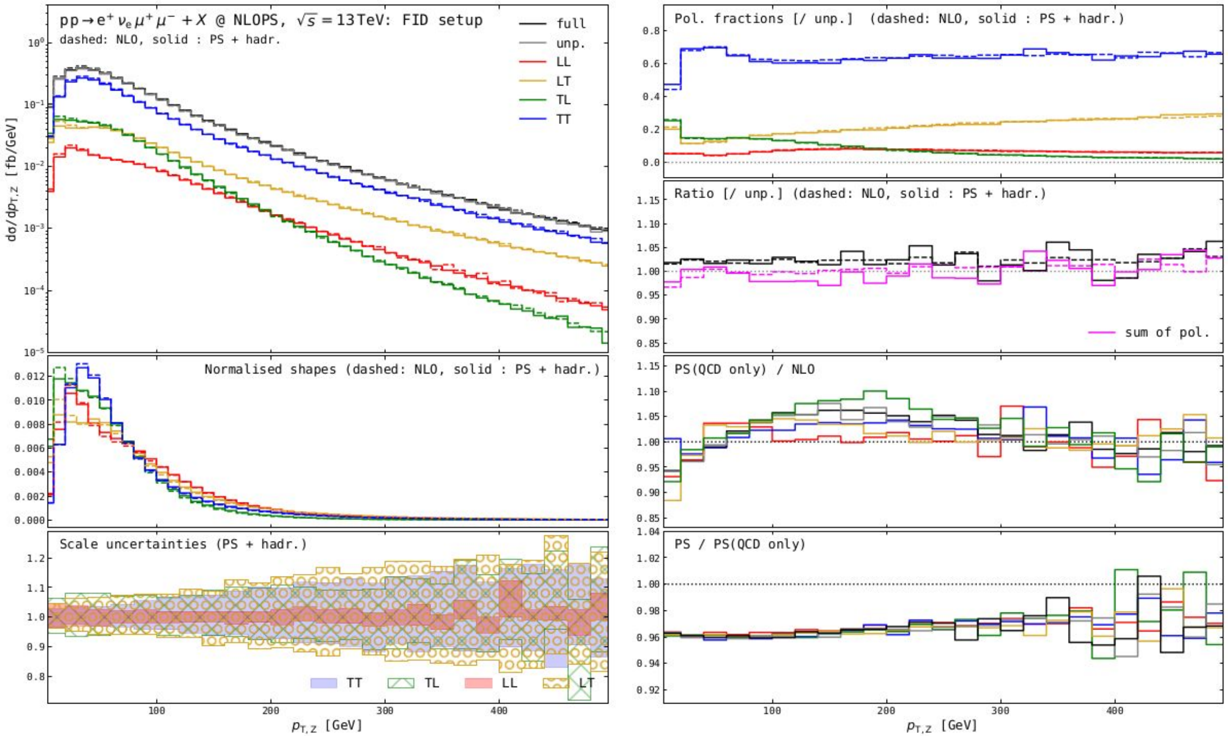
- left: $W^{\pm}Z$. It includes also NLO$_{\mathrm{EW}}$+PS corrections in various approximations
- right: $\gamma\gamma$. It required also some minor modification to the MiNNLO$_{\mathrm{PS}}$ master formula

Polarized diboson

$$\Phi_{4\ell} = \{x_1, x_2; k_{1\ldots4}\} \xrightarrow{\text{FKS}} (\bar{\Phi}_{4\ell}, \Phi_{\text{rad}}) = \{\bar{x}_1, \bar{x}_2; \bar{k}_{1\ldots4}, k_{\text{rad}}\}$$

$$\xrightarrow{\text{DPA}} (\bar{\tilde{\bar{\Phi}}}_{4\ell}, \Phi_{\text{rad}}) = \{\bar{x}_1, \bar{x}_2; \bar{\tilde{\bar{k}}}_{1\ldots4}, k_{\text{rad}}\}$$

Z-boson p_T, fiducial setup [ATLAS 2211.09435].



[Pelliccioli,Zanderighi ‘23]

+ talk J.Linder (yesterday)

Conclusions

- Diboson production in POWHEG: different generators, different accuracy
- Up to NNLO_{QCD} / NLO_{EW} available
- Polarized bosons: NLO_{QCD}

- Accuracy of parton showers not discussed at all
  - LL → NLL accuracy of parton showers [talk by M. Van Beekveld @ MBI]

Part 2: tutorial

Technical details: general remarks (I)

- In general, process "**X**" contains one or more directories with tailored input cards. You should always start from these inputs, and, in normal circumstances, change/adapt only the "integration" parameters, or those related to couplings, PDFs, etc...
  - The purpose of this presentation and the next one is to explain (or recall) how and when to do this
- Copying flags from input cards of process **X** to input cards of process **Y** should be avoided
  - especially for complex processes (or in modern applications, *e.g.* treatment of resonances, QCD+EW corrections, `MiNLO/MiNNLOPS`), the authors of process **X** might have introduced specific flags that make sense only in that case
- "how to run / how to check": most of what I'll discuss is documented in `POWHEG-BOX-V2/Docs/V2-paper.pdf`

Technical details: general remarks (II)

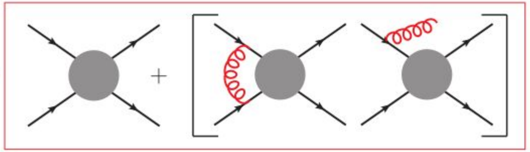
- green flag: for the users, ok to change it, although, in general, the values you find in the public input cards are already optimized (especially for CPU-intensive processes)
- orange flag: it is useful if you know what the flag does, but you should not change it, unless you really know what you are doing

  (if the flag is there and has a given value, typically there is a reason)
- red flag: you should not change it

  (I mention it because, sometimes, these flags are explicitly present in the input cards)
- (flag in bracket): optional flag

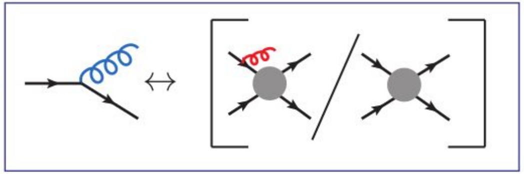
`POWHEG-BOX-V2` / `POWHEG-BOX-RES`: very similar flags

$$B(\Phi_n) \Rightarrow \bar{B}(\Phi_n) = B(\Phi_n) + \frac{\alpha_s}{2\pi}\left[V(\Phi_n) + \int R(\Phi_{n+1})\, d\Phi_r\right]$$



$$d\sigma_{\rm POW} = d\Phi_n \quad \bar{B}(\Phi_n) \quad \left\{\Delta(\Phi_n; k_{\rm T}^{\rm min}) + \Delta(\Phi_n; k_{\rm T})\frac{\alpha_s}{2\pi}\frac{R(\Phi_n, \Phi_r)}{B(\Phi_n)}\, d\Phi_r\right\}$$

[+ $p_{\rm T}$ *-vetoing subsequent emissions*, to avoid double-counting]



$$\Delta(t_{\rm m}, t) \Rightarrow \Delta(\Phi_n; k_{\rm T}) = \exp\left\{-\frac{\alpha_s}{2\pi}\int \frac{R(\Phi_n, \Phi'_r)}{B(\Phi_n)}\theta(k'_{\rm T} - k_{\rm T})\, d\Phi'_r\right\}$$

Technical details (I - workflow)

- stage 1: build and store the importance sampling grid for the integration of the inclusive cross section.

  $[\tilde{B}(\Phi), \Phi \leftrightarrow X_i]$

  - **`ncall1`** and **`itmx1`**: total number of calls per run is `ncall1*itmx1`

- stage 2: using the grids from stage 1, integrate inclusive cross section and find the *upper bounding envelope of the inclusive cross section*.

  $[\tilde{B}(\Phi) < \prod_i f_i(X_i)]$

  - **`ncall2`** and **`itmx2`**: total number of calls per run is `ncall2*itmx2`
  - (**`testplots`**): `0/1`. If activated, “NLO” plots are produced while the x-section is integrated.

    [Code slightly slower, mainly intended for developers, useful to inspect if unstable st2 results even with large stat.]

- stage 3: compute the *upper bounds for the generation of radiation*.

  $[R/B < N f(\Phi_{\mathrm{rad}})]$

  - **`nubound`**: number of points in $(\Phi_B, \Phi_{\mathrm{rad}})$ used to probe R/B

- stage 4: generation of the events with hardest radiation.

  $[d\sigma = \bar{B}(\Phi_{\mathrm{B}}) d\Phi_{\mathrm{B}} \{\Delta(p_{\mathrm{T}})(R/B) d\Phi_{\mathrm{rad}}\}]$

  - **`numevts`**: number of events

☞ There are 2 different “*upper bounds*”, computed at stage 2 and 3.
If not determined precisely (too low stat, too many unstable runs,...), the final results will not be reliable!

Technical details (II - other settings)

- scale choice:
  Central scale choice ($M_{\ell\ell}$, $p_{\rm T}$, H_T,...) is process dependent, typically documented in the published paper, coded in `set_fac_ren_scales`. When different choices are available, typically you'll find a flag with an obvious name and a comment, *e.g.* **`runningscales`**
- (**`renscfact`** , **`facscfact`**): `0.5/1.0/2.0`
  Pilot μ_R and μ_F scale variation. Results are obtained through reweighting of the events.
- (**`hdamp`**): `X [GeV]`
  Parameter to split real matrix elements into a singular and finite part. The latter enter the "remnant" contribution
  $$R_s = R\frac{\mathtt{X}^2}{\mathtt{X}^2 + p_{\rm T}^2}, \quad R_f = R - R_s$$
- (**`bornzerodamp`**): `0/1`
  Further separation of real matrix elements, according to their "distance" from the singular limits.
  - Originally introduced to deal with zeros in the Born matrix elements, by now used in (most) processes.
  - It helps the efficiency of event generation, and also helps in having LO-like uncertainty band for observables where this is expected.
- (**`withdamp`** , **`hfact`**): superseded and deprecated
- (**`xupbound`**): `X`
  Overall multiplication of the upper bounds of R/B: $Nf(\Phi_{\rm rad}) \to \mathtt{X} Nf(\Phi_{\rm rad})$
  - Can help in reducing *failures in generation of radiation* without re-executing stage 3 (which remains, however, the preferred choice, as I'll discuss).

Technical details (III - other settings)

- (**`alphas_from_pdf`**): `0/1`
  Running of α_S from PDF provider. Flag introduced recently, it must be activated in all `MiNNLOPS` simulations, but can be used also elsewhere.
- (**`alphas_from_lhapdf`**): superseded
- (**`ptsqmin`**): `X` $[\mathrm{GeV}^2]$
  IR cutoff on the generation of the hardest radiation.
  This parameter should not be changed, unless you have a very strong reason.
  If you need to do so, first you are strongly encouraged to discuss with us.

  Exception: **`ptsqmin 1`**
  This is needed if `Herwig 7` used to shower
- (**`withnegweights`**): `0/1`
  Keep/discard events with negative weights.
  In all modern applications, negative weights are present. If you have too many, they can be reduced through the “folding” technique (see below).
  Removing them leads to WRONG results.
- (**`foldcsi`** , **`foldy`** , **`foldphi`**): `1/2/3/...`
  Multiple sampling of emission phase space (Φ_{rad}) at fixed underlying-Born (Φ_{B}).
  Slower code, but less negative weights. If in doubt, please ask us.

Technical details (parallelization)

- Even when running in parallel on a cluster, the executable always reads `powheg.input`. It needs to be changed when going from step n to step $n+1$.
- Template scripts to run on a multicore machine are often provided in the process directory. General (but detailed) explanations can be found in `POWHEG-BOX-V2/Docs`

Relevant flags

- **`manyseeds`**: `0/1`, activates parallel runs
  - a `pwgseeds.dat` file is needed: list of integer seeds for the random number sequence, one per line
- **`parallelstage`**: `1/2/3/4`, selects the stage
- **`xgriditeration`**: `1/2/...`, iteration of the calculation of the importance sampling grid at stage 1.
  - If set to $n+1$, all results obtained during iteration n are loaded, and another iteration of stage 1 is launched.
  - In practice, for parallel runs, `itmx1` is superseded by this mechanism.
- (**`maxseeds`**): `N`, default is 200, increase this when using more integer seeds
- (**`check_bad_st1`** , **`check_bad_st2`**): `0/1`.
  - When assembling grids produced at previous stage, check if some grids have big numerical instabilities. If this is the case, they are discarded.
  - If too many occurrences are found, the integration is probably not converging. The program stops and complains!

Now the real tutorial…

1) run interactively the HJ code (fast)

2) look at WW: LO/$\mathrm{NLO}_{\mathrm{QCD}}$/$\mathrm{NLO}_{\mathrm{QCD+}}\mathrm{NLO}_{\mathrm{EW}}$:

⟶ $\mathrm{NLO}_{\mathrm{QCD}}$ with polarized bosons works the same way, publis soon

Technical details (final checks during/after parallel runs)

- This is the “final” check, you should always do it
- Inspect `pwgcounters-st4-*.dat`

```
# grep failure pwgcounters-st4-*.dat
```

- Guideline:
  - `upper bound failure in inclusive cross section`

  → If $> \mathcal{O}(1)\%$ of N_{events}
    - issue: NLO accuracy at risk, some phase space regions wrongly populated
    - how to fix it: increase `ncall2, itmx2` (possibly also `ncall1`)
    - requires to re-start from st2, or even before
  - `upper bound failures in generation of radiation`

  → If $> \mathcal{O}(1)\%$ of N_{events}
    - issue: pattern of hardest radiation compromised
    - how to fix it: `nubound` needs to be increased (increasing `xupbound` can also help)
    - requires to re-start from st3 (st4)

Technical details (final checks during/after parallel runs)

Take home messages:

- cross sections reported in `pwg-st2-*-stat.dat` should be stable.
- at the beginning of stage 3, the results of stage 2 are combined: the cross sections reported in `pwg-st3-*-stat.dat` should be stable and with a small error.
- the information in `pwgcounters-st4-*.dat` should always be checked: this is the place where even subtle problems are more likely to be found
- the same comments apply to results obtained with the POWHEG-BOX-RES generator (just minor differences in the syntax, but the content of these files is the same)

Now the real tutorial…

1) run interactively the HJ code (fast)

2) look at WW: LO/NLO$_{QCD}$/NLO$_{QCD+}$NLO$_{EW}$:

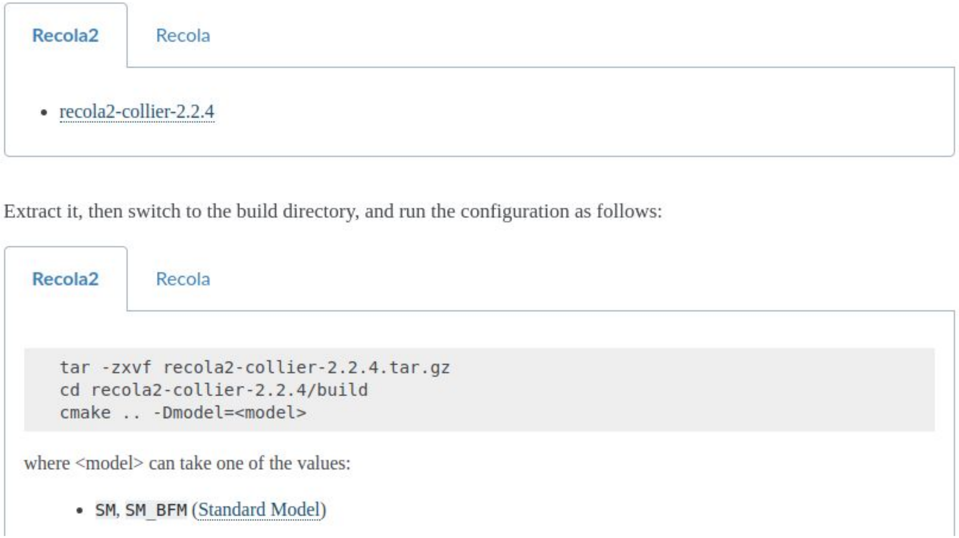
→ **NLO$_{QCD}$ with polarized bosons works the same way**, public soon

WW@{LO+PS}

- Download [Recola2 needed + lhapdf (+fastjet)]

POWHEG: **https://powhegbox.mib.infn.it/**
\# svn checkout --username anonymous --password anonymous svn://powhegbox.mib.infn.it/trunk/POWHEG-BOX-RES
\# svn checkout --username anonymous --password anonymous svn://powhegbox.mib.infn.it/trunk/User-Processes-RES/**VV_dec_ew**

Recola: **https://recola.gitlab.io/recola2/installation.html**



- recola2-collier-2.2.4

Extract it, then switch to the build directory, and run the configuration as follows:

```
tar -zxvf recola2-collier-2.2.4.tar.gz
cd recola2-collier-2.2.4/build
cmake .. -Dmodel=<model>
```

where <model> can take one of the values:

- `SM`, `SM_BFM` (Standard Model)

WW@{LO+PS}

- Install [Recola2 needed + lhapdf (+fastjet)]:
  in the **VV_dec_ew** directory, edit the **Makefile**:

```
##############################################
# >>> tutorial
# Recola
RECOLALOCATION=/home/ere/path_to_recola2
RECOLAMODULEDIR = $(RECOLALOCATION)/include
RECOLALIB += -Wl,-rpath,$(RECOLALOCATION)/lib -L$(RECOLALOCATION)/lib -lrecola
RECFLAGS=-I$(RECOLAMODULEDIR)
###############################################
…..
####################################
# >>> tutorial
HEPMCLOCATION=/home/ere/path_to_HEPMC3/hepmc3-install
LIBSHEPMC += -L$(HEPMCLOCATION)/lib -lHepMC
#####################################
……
###############################################
# >>> tutorial
PYTHIA8LOCATION=/home/ere/path_to_pythia
FJCXXFLAGS+=-I$(PYTHIA8LOCATION)/include
LIBPYTHIA8=-L$(PYTHIA8LOCATION)/lib -lpythia8 -lstdc++ -ldl
################################################
```

WW@{LO+PS}

- Input card:

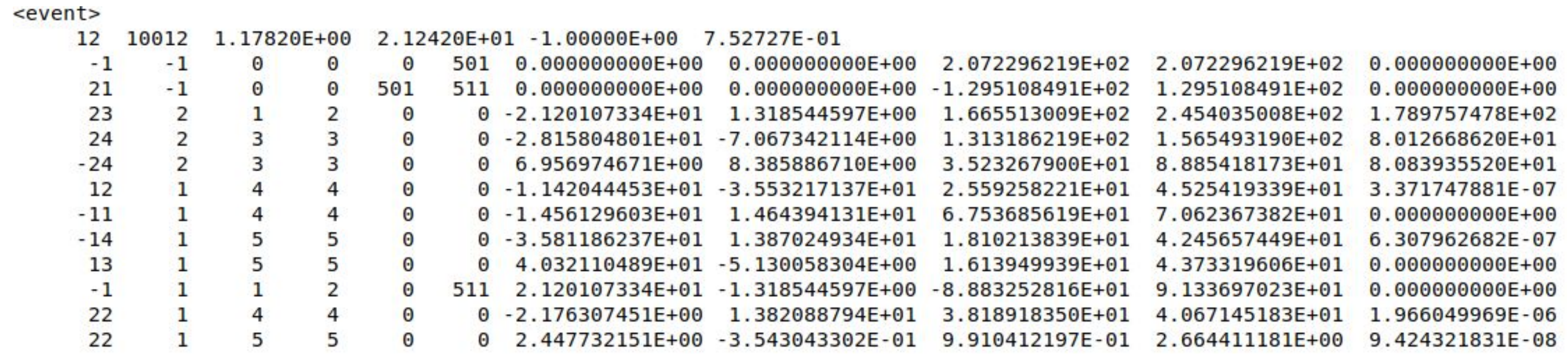
```
bornonly 1 ! for LO+PS

!Single vector boson production parameters

procVV 1         ! 1=WW,2=ZZ,3=ZW,4=ZW
idvecbos 24      !W+=+24, W-=-24. Used only for procVV=3,4
decayV1 1        ! 1=electrons/nu e, 2=muons/nu mu, 3=taus/nu tau.
decayV2 2        ! 1=electrons/nu e, 2=muons/nu mu, 3=taus/nu tau.

ncall1 20000     ! number of calls for initializing the integration grid
itmx1   1        ! number of iterations for initializing the integration grid
ncall2 20000     ! number of calls for computing the integral and finding upper
bound
itmx2    1       ! number of iterations for computing the integral and…
nubound 10000    ! number of bbarra calls to setup norm of upper bounding…
numevts 10000    ! number of events (for each random number seed)
```

WW@{LO+PS}

- run (small cluster, here (LO+PS) using simply 128 CPU)
- will generate 1.3M *unweighted* partonic events (LHE files)
- will go step-by-step, but automatic scripts exist

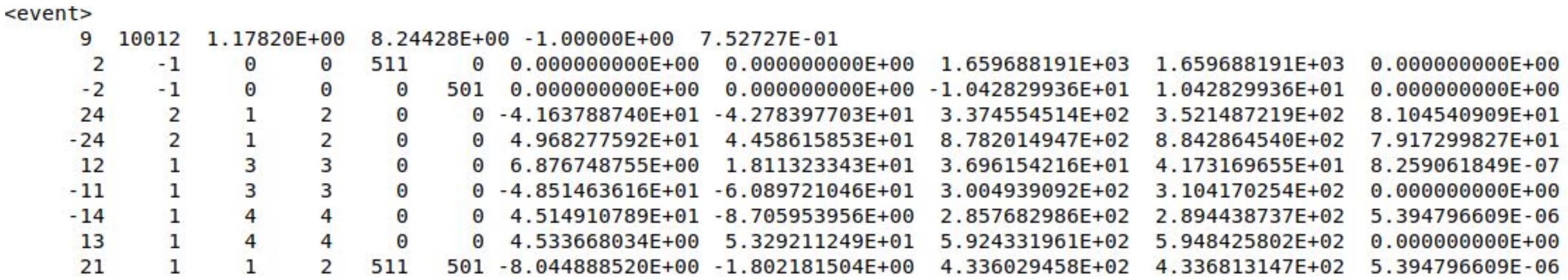
WW@{NLO QCD+NLO EW+PS}: hard events

- stage 4: “partonic” (LHE) event files: `pwgevents-*.lhe`
  - → can contain multiple radiation `[allrad` flag`]`
- for polarized bosons: only QCD radiation



```
<event>
     12  10012  1.17820E+00  2.12420E+01 -1.00000E+00  7.52727E-01
      -1     -1      0      0      0    501  0.000000000E+00  0.000000000E+00  2.072296219E+02  2.072296219E+02  0.000000000E+00
      21     -1      0      0    501    511  0.000000000E+00  0.000000000E+00 -1.295108491E+02  1.295108491E+02  0.000000000E+00
      23      2      1      2      0      0 -2.120107334E+01  1.318544597E+00  1.665513009E+02  2.454035008E+02  1.789757478E+02
      24      2      3      3      0      0 -2.815804801E+01 -7.067342114E+00  1.313186219E+02  1.565493190E+02  8.012668620E+01
     -24      2      3      3      0      0  6.956974671E+00  8.385886710E+00  3.523267900E+01  8.885418173E+01  8.083935520E+01
      12      1      4      4      0      0 -1.142044453E+01 -3.553217137E+01  2.559258221E+01  4.525419339E+01  3.371747881E-07
     -11      1      4      4      0      0 -1.456129603E+01  1.464394131E+01  6.753685619E+01  7.062367382E+01  0.000000000E+00
     -14      1      5      5      0      0 -3.581186237E+01  1.387024934E+01  1.810213839E+01  4.245657449E+01  6.307962682E-07
      13      1      5      5      0      0  4.032110489E+01 -5.130058304E+00  1.613949939E+01  4.373319606E+01  0.000000000E+00
      -1      1      1      2      0    511  2.120107334E+01 -1.318544597E+00 -8.883252816E+01  9.133697023E+01  0.000000000E+00
      22      1      4      4      0      0 -2.176307451E+00  1.382088794E+01  3.818918350E+01  4.067145183E+01  1.966049969E-06
      22      1      5      5      0      0  2.447732151E+00 -3.543043302E-01  9.910412197E-01  2.664411181E+00  9.424321831E-08
```

← QCD
← QED
← QED



```
<event>
      9  10012  1.17820E+00  8.24428E+00 -1.00000E+00  7.52727E-01
       2     -1      0      0    511      0  0.000000000E+00  0.000000000E+00  1.659688191E+03  1.659688191E+03  0.000000000E+00
      -2     -1      0      0      0    501  0.000000000E+00  0.000000000E+00 -1.042829936E+01  1.042829936E+01  0.000000000E+00
      24      2      1      2      0      0 -4.163788740E+01 -4.278397703E+01  3.374554514E+02  3.521487219E+02  8.104540909E+01
     -24      2      1      2      0      0  4.968277592E+01  4.458615853E+01  8.782014947E+02  8.842864540E+02  7.917299827E+01
      12      1      3      3      0      0  6.876748755E+00  1.811323343E+01  3.696154216E+01  4.173169655E+01  8.259061849E-07
     -11      1      3      3      0      0 -4.851463616E+01 -6.089721046E+01  3.004939092E+02  3.104170254E+02  0.000000000E+00
     -14      1      4      4      0      0  4.514910789E+01 -8.705953956E+00  2.857682986E+02  2.894438737E+02  5.394796609E-06
      13      1      4      4      0      0  4.533668034E+00  5.329211249E+01  5.924331961E+02  5.948425802E+02  0.000000000E+00
      21      1      1      2    511    501 -8.044888520E+00 -1.802181504E+00  4.336029458E+02  4.336813147E+02  5.394796609E-06
```

← QCD

running the parton shower

- stage 4: “partonic” (LHE) event files: `pwgevents-*.lhe`
  → `can contain multiple radiation`
- Loop over events in `pwgevents-*.lhe`
- Store partonic information (momenta, scales, …)
- Feed partonic event to `Pythia`
- `Pythia`: generate parton shower, vetoing radiation

- one QCD emission [standard]:
  run p_T-ordered parton shower, vetoing emissions harder than the 1st one
- QCD+QED emissions:
  veto shower for each resonance (requires dedicated interface + use of `Pythia` facilities)

- Interface shipped in process directory: **`main-PYTHIA*-lhef`**
- For NLO QCD, interface is very standard

WW QCD+EW: run time [*unweighted events*]

LO: quick

QCD

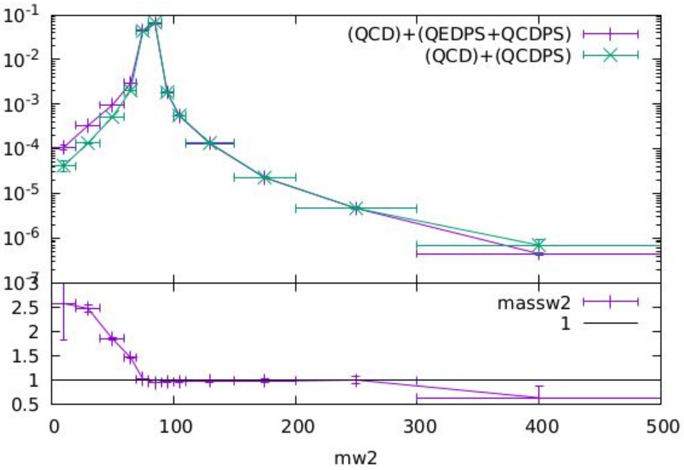
- 2 nodes x 128 CPU
- st1-st2-st3 → 3h
- st4: 10K events → 20min
- 5M events in ½ day
- VV polarized @ NLO+PS: this accuracy
- NLO scale uncertainty through reweighting

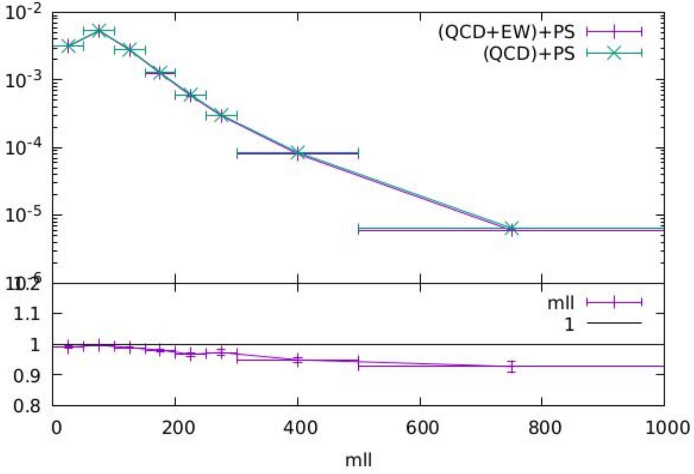
QCD+EW

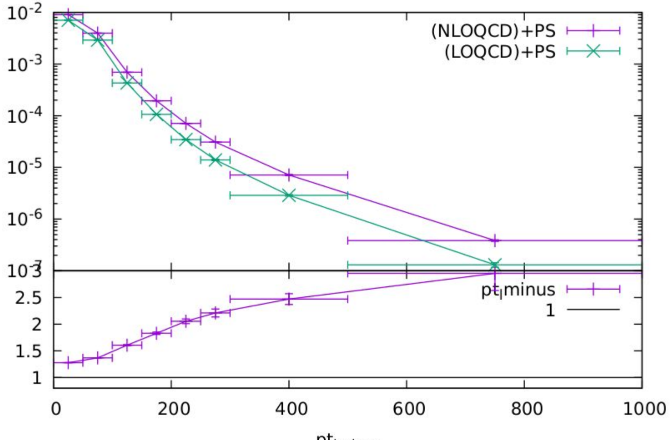
- 3-4 nodes x 128 CPU
- st1-st2-st3 → 1/2 day
- st4: 2K events → ~8h
- 1.5M events in 1.5 day
- 1-loop correction “slow”
- Possible to make it faster through “reweighting” (let us know if needed)

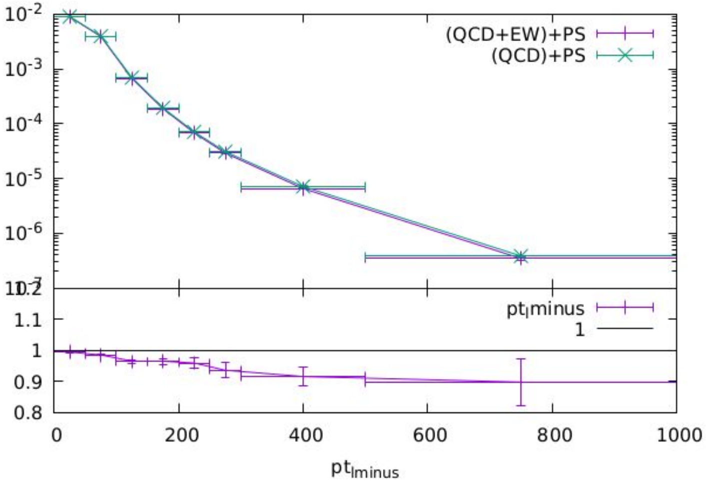
LHE → PS: quick [MPI off]

WW QCD+EW: plots







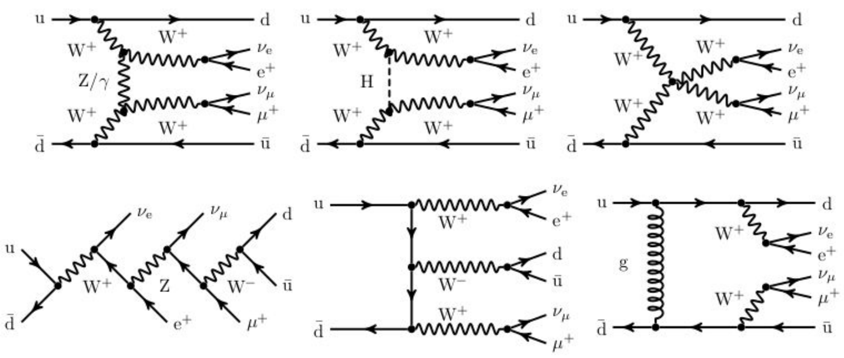


Thank you for your attention!

Backup slides

NLOEW+PS: bottlenecks

[slide from M. Chiesa]



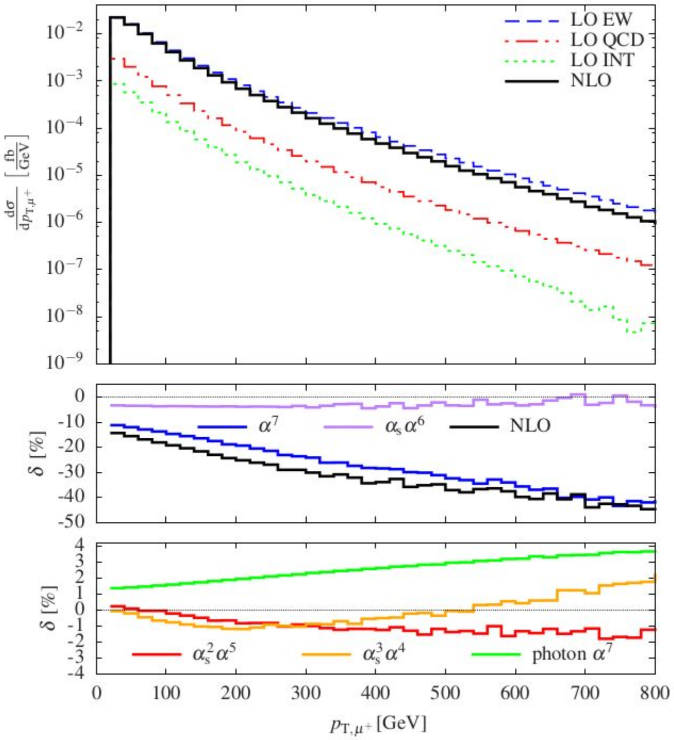
$$\mathcal{M} \simeq \quad \mathcal{O}(\alpha^3) \qquad \mathcal{O}(\alpha^3) \qquad \mathcal{O}(\alpha_S\alpha^2)$$

$$\mathrm{LO} \simeq \quad \mathcal{O}(\alpha^6) \qquad \mathcal{O}(\alpha^5\alpha_S) \qquad \mathcal{O}(\alpha_S^2\alpha^4)$$

$$\begin{tikzcd}
\textbf{LO} & & \mathcal{O}(\alpha^6) \arrow[dl, "\mathrm{EW}"'] \arrow[dr, "\mathrm{QCD}"] & & \mathcal{O}(\alpha_s\alpha^5) \arrow[dl, "\mathrm{EW}"'] \arrow[dr, "\mathrm{QCD}"] & & \mathcal{O}(\alpha_s^2\alpha^4) \arrow[dl, "\mathrm{EW}"'] \arrow[dr, "\mathrm{QCD}"] & \\
\textbf{NLO} & \mathcal{O}(\alpha^7) & & \mathcal{O}(\alpha_s\alpha^6) & & \mathcal{O}(\alpha_s^2\alpha^5) & & \mathcal{O}(\alpha_s^3\alpha^4)
\end{tikzcd}$$

NLOEW+PS: bottlenecks

[slide from M. Chiesa]



Limitations of NLO-EW corrections in `POWHEG`

Strategy:

- consider only LO $\mathcal{O}(\alpha^6)$
- consider only corrections $\mathcal{O}(\alpha^7)$
- $\mathcal{O}(\alpha_S\alpha^6)$ in PS approximation or via combination with NLO-QCD+QCD PS results

Resonant-aware NLO+PS: details I

1. complete matrix elements for $W^+W^-b\bar{b}$: need to project each partonic subprocess onto all possible “resonance histories”:
    - each contribution should be dominated by a single resonance history:

$$B = \sum_{f_b} B_{f_b}, \quad \text{where} \quad B_{f_b} \equiv \frac{P^{f_b}(\Phi_B)}{\sum_{f'_b} P^{f'_b}(\Phi_B)} B(\Phi_B)$$

$P^{f_b}(\Phi_B)$ (products of) Breit-Wigner functions ⇔ resonance history f_b

    - for real contributions, split also according to compatible FKS regions

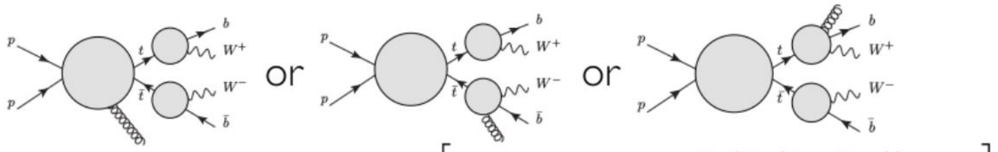
      ⇒ a term R_{α_r} is dominant if the collinear partons of region α_r have the smallest k_T, and the corresponding resonance history is the closest to its mass shell.

2. each term (Born-like and real) is attributed to an unique resonance history
    - virtuality-preserving mappings between Φ_B and $(\Phi_B, \Phi_{\text{rad}})$ can be used
    - `POWHEG` radiation(s) can now be assigned to a resonance
    - (& other technical but crucial subtleties...)

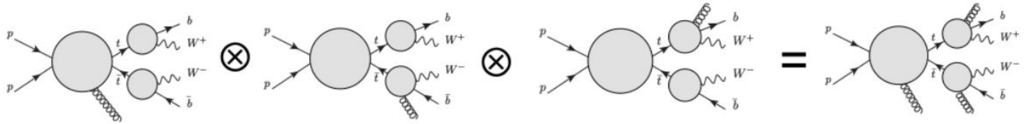
Resonant-aware NLO+PS: details II

“multiplicative `POWHEG`”: keep multiple emissions before showering [☜]

- by default `POWHEG` is additive: keeps only the hardest emission
- keep hard radiation and the emissions from all decaying resonances, then merge them into a single radiation phase space with several radiated partons, up to one for each resonance



$$\Longleftrightarrow \qquad \mathrm{d}\sigma = \bar{B}(\Phi_{\mathrm{B}})\,\mathrm{d}\Phi_{\mathrm{B}}\left[\Delta(q_{\mathrm{cut}}) + \sum_{\alpha}\Delta(k_T^{\alpha})\frac{R_{\alpha}(\Phi_{\alpha}(\Phi_{\mathrm{B}},\Phi_{\mathrm{rad}}))}{B(\Phi_{\mathrm{B}})}\,\mathrm{d}\Phi_{\mathrm{rad}}\right]$$



$$\Longleftrightarrow \qquad \mathrm{d}\sigma = \bar{B}(\Phi_{\mathrm{B}})\,\mathrm{d}\Phi_{\mathrm{B}}\prod_{\alpha=\alpha_b,\alpha_{\bar{b}},\alpha_{\mathrm{ISR}}}\left[\Delta_{\alpha}(q_{\mathrm{cut}}) + \Delta_{\alpha}(k_T^{\alpha})\frac{R_{\alpha}(\Phi_{\alpha}(\Phi_{\mathrm{B}},\Phi_{\mathrm{rad}}^{\alpha}))}{B(\Phi_{\mathrm{B}})}\,\mathrm{d}\Phi_{\mathrm{rad}}^{\alpha}\right]$$

- in the above case, the interface to parton shower becomes more complicated.