Next to Leading Order and aMC@NLO

Olivier Mattelaer
University of Illinois at Urbana Champaign

Aims for this lecture

- Get you acquainted with the concepts and techniques used in event generation
- Give you hands-on experience with matrix element generation, event generation and analysis
- Answer as many of your questions as I can (so please ask questions!)

Contents

- Ingredients to a NLO calculations
  - ➡ A bit more detail on canceling divergences
  - ➡ Computing loops efficiently
- aMC@NLO
- No Shower

Master equation for hadron colliders

$$d\sigma = \sum_{a,b} \int dx_1 dx_2 \;\; f_a(x_1,\mu_F) f_b(x_2,\mu_F)\; d\hat{\sigma}_{ab\to X}(\hat{s},\mu_F,\mu_R)$$

Parton density functions

Parton-level (differential) cross section

- Parton-level cross section from matrix elements: model and process dependent
- Parton density (or distribution) functions: process independent

Perturbative expansion

$d\hat{\sigma}_{ab\to X}(\hat{s}, \mu_F, \mu_R)$ Parton-level cross section

- The parton-level cross section can be computed as a series in perturbation theory, using the coupling constant as an expansion parameter, schematically:

$$\hat{\sigma} = \sigma^{\mathrm{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)} + \ldots\right)$$

- Including higher corrections improves predictions and reduces theoretical uncertainties

Perturbative expansion

$d\hat{\sigma}_{ab\to X}(\hat{s},\mu_F,\mu_R)$ Parton-level cross section

- The parton-level cross section can be computed as a series in perturbation theory, using the coupling constant as an expansion parameter, schematically:

$$\hat{\sigma} = \sigma^{\text{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)} + \dots\right)$$

LO predictions

- Including higher corrections improves predictions and reduces theoretical uncertainties

Perturbative expansion

$d\hat{\sigma}_{ab\to X}(\hat{s}, \mu_F, \mu_R)$ Parton-level cross section

- The parton-level cross section can be computed as a series in perturbation theory, using the coupling constant as an expansion parameter, schematically:

$$\hat{\sigma} = \sigma^{\mathrm{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)} + \dots\right)$$

LO predictions

NLO corrections

- Including higher corrections improves predictions and reduces theoretical uncertainties

Perturbative expansion

$d\hat{\sigma}_{ab\to X}(\hat{s},\mu_F,\mu_R)$ Parton-level cross section

- The parton-level cross section can be computed as a series in perturbation theory, using the coupling constant as an expansion parameter, schematically:

$$\hat{\sigma} = \sigma^{\mathrm{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)} + \ldots\right)$$

LO predictions

NLO corrections

NNLO corrections

- Including higher corrections improves predictions and reduces theoretical uncertainties

Perturbative expansion

$d\hat{\sigma}_{ab\to X}(\hat{s},\mu_F,\mu_R)$ Parton-level cross section

- The parton-level cross section can be computed as a series in perturbation theory, using the coupling constant as an expansion parameter, schematically:

$$\hat{\sigma} = \sigma^{\text{Born}}\left(1+\frac{\alpha_s}{2\pi}\sigma^{(1)}+\left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)}+\left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)}+\ldots\right)$$

LO predictions

NLO corrections

NNLO corrections

N3LO or NNNLO corrections

- Including higher corrections improves predictions and reduces theoretical uncertainties

Perturbative expansion

$d\hat{\sigma}_{ab\to X}(\hat{s},\mu_F,\mu_R)$ Parton-level cross section

- The parton-level cross section can be computed as a series in perturbation theory, using the coupling constant as an expansion parameter, schematically:

$$\hat{\sigma} = \sigma^{\mathrm{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)} + \ldots\right)$$

LO predictions

NLO corrections

NNLO corrections

N3LO or NNNLO corrections

- Including higher corrections improves predictions and reduces theoretical uncertainties

Improved predictions

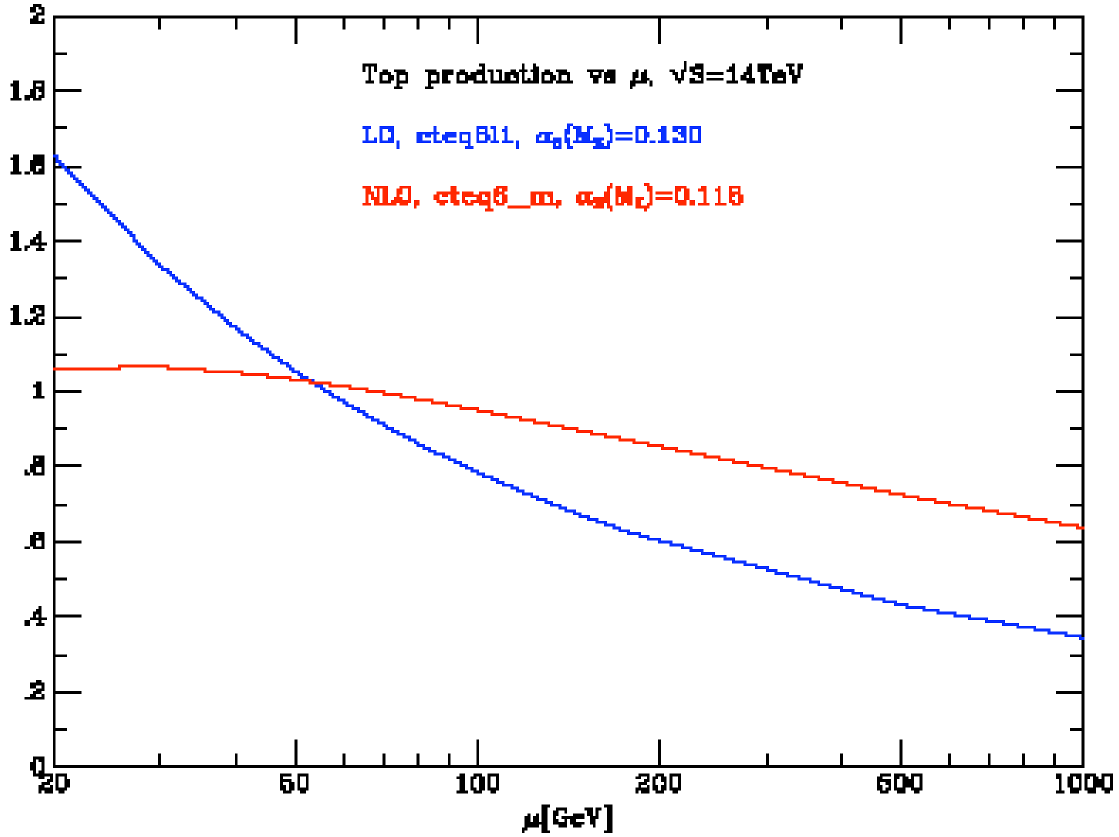
$$d\sigma = \sum_{a,b} \int dx_1 dx_2 \; f_a(x_1, \mu_F) f_b(x_2, \mu_F) \, d\hat{\sigma}_{ab\to X}(\hat{s}, \mu_F, \mu_R)$$

$$\hat{\sigma} = \sigma^{\mathrm{Born}} \left(1 + \frac{\alpha_s}{2\pi} \sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2 \sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3 \sigma^{(3)} + \dots \right)$$

- Leading Order predictions can depend strongly on the renormalization and factorization scales
- Including higher order corrections reduces the dependence on these scales

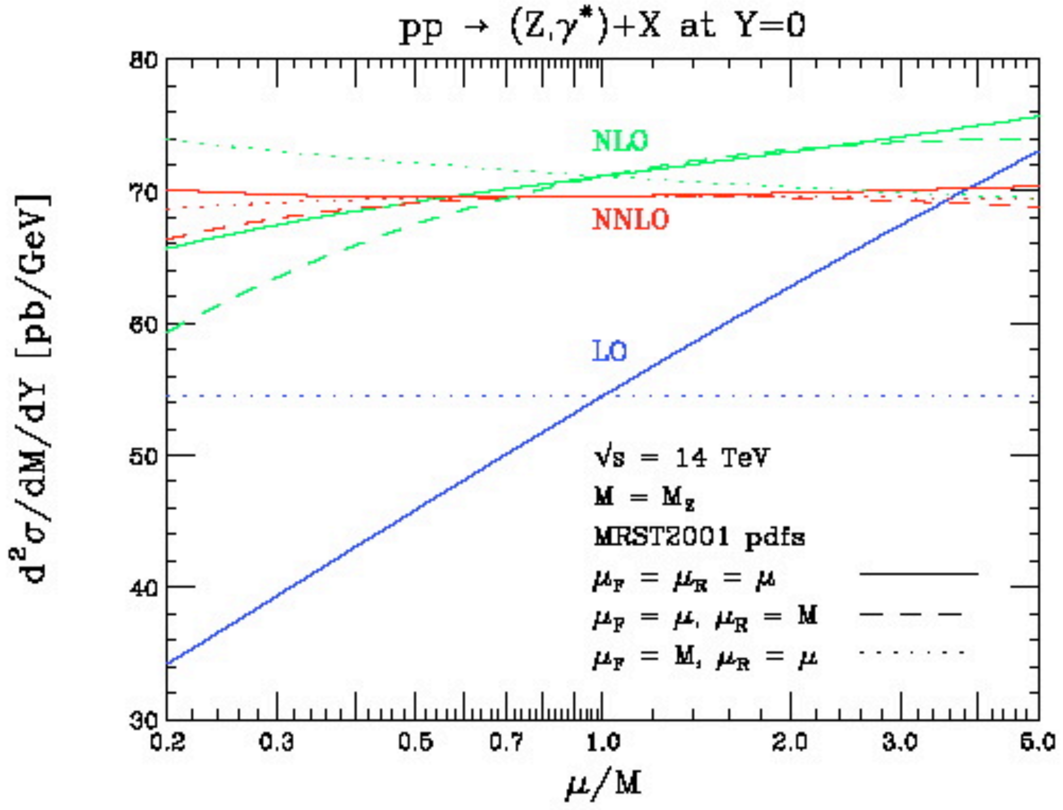
Going NLO

- At NLO the dependence on the renormalization and factorization scales is reduced
  - ➡ First order where scale dependence in the running coupling and the PDFs is compensated for via the loop corrections: **first reliable estimate of the total cross section**
  - ➡ Better description of final state: impact of extra radiation included (e.g. jets can have substructure)
  - ➡ Opening of additional initial state partonic channels

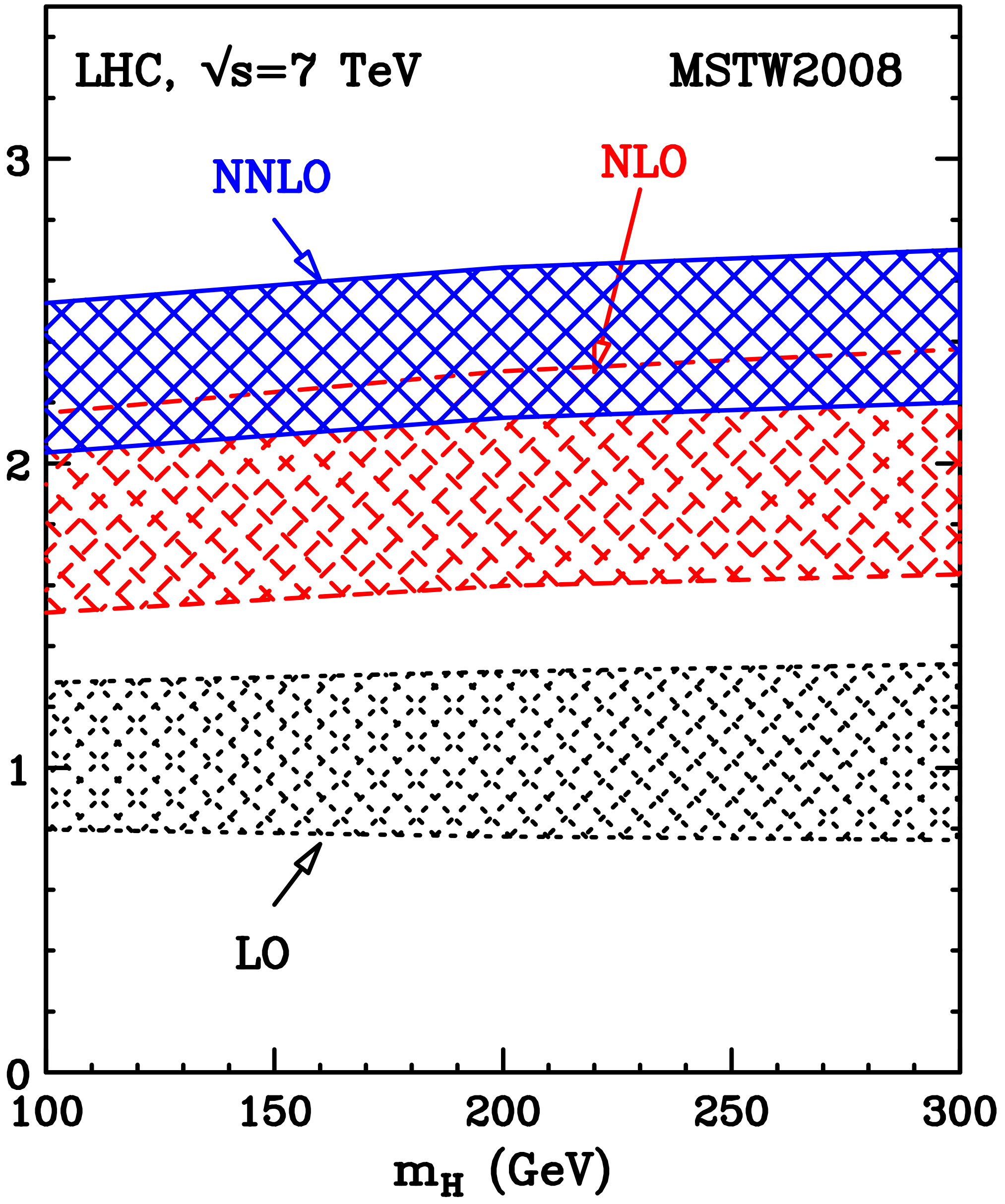


Going NNLO...?

- NNLO is the current state-of-the-art. There are only a few results available: Higgs, Drell-Yan, ttbar (qqbar induced only)
- Why do we need it?
  - ➡ control of the uncertainties in a calculation
  - ➡ It is “mandatory” if NLO corrections are very large to check the behavior of the perturbative series
  - ➡ It is needed for Standard Candles and very precise tests of perturbation theory, exploiting all the available information, e.g. for determining NNLO PDF sets

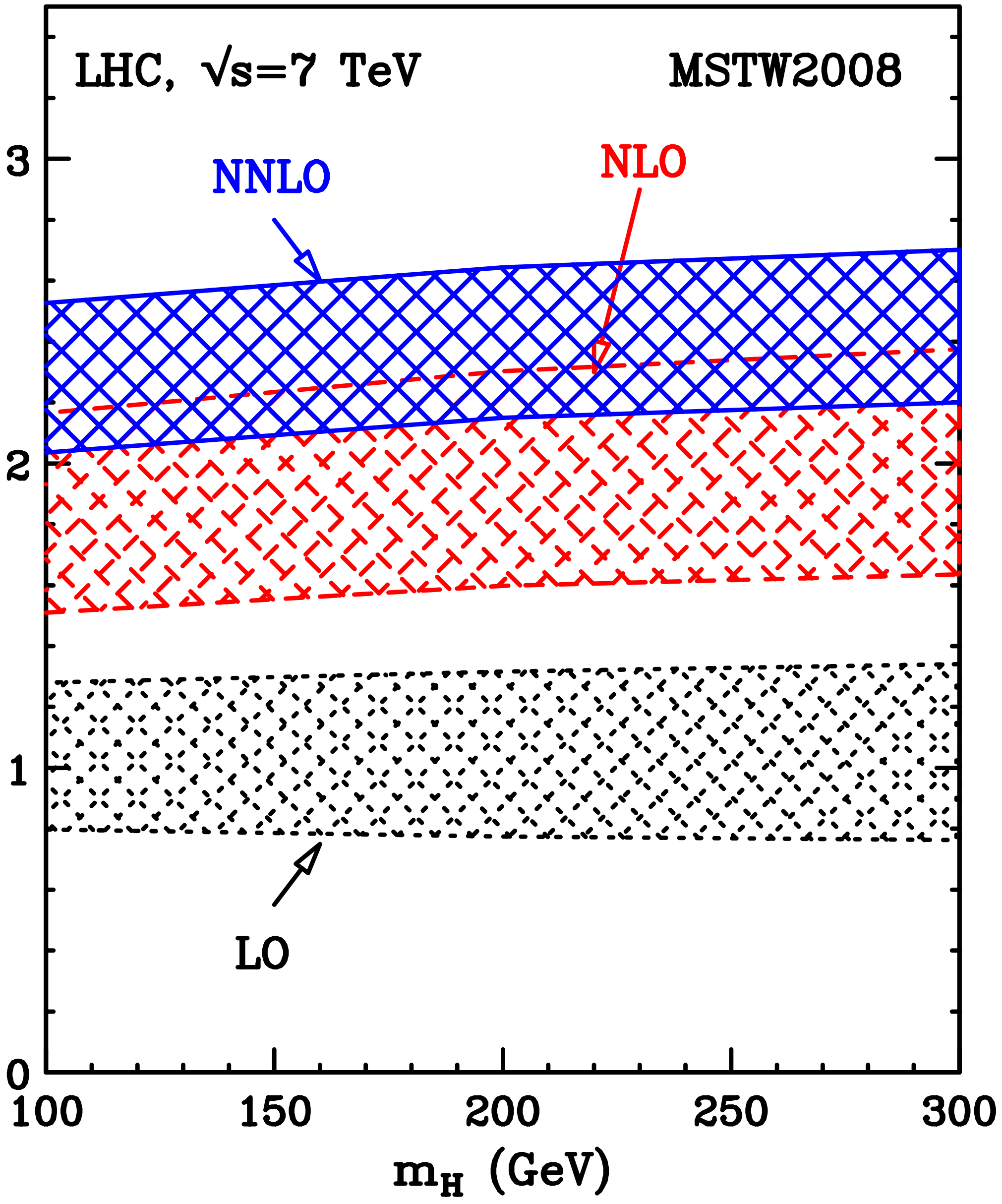


Higgs predictions at NNLO



- LO calculation is not reliable,
- but the perturbative series stabilizes at NNLO
- NLO estimation of the uncertainties (by scale variation) works reasonably well

Higgs predictions at NNLO



- LO calculation is not reliable,
- but the perturbative series stabilizes at NNLO
- NLO estimation of the uncertainties (by scale variation) works reasonably well

Let's focus on NLO

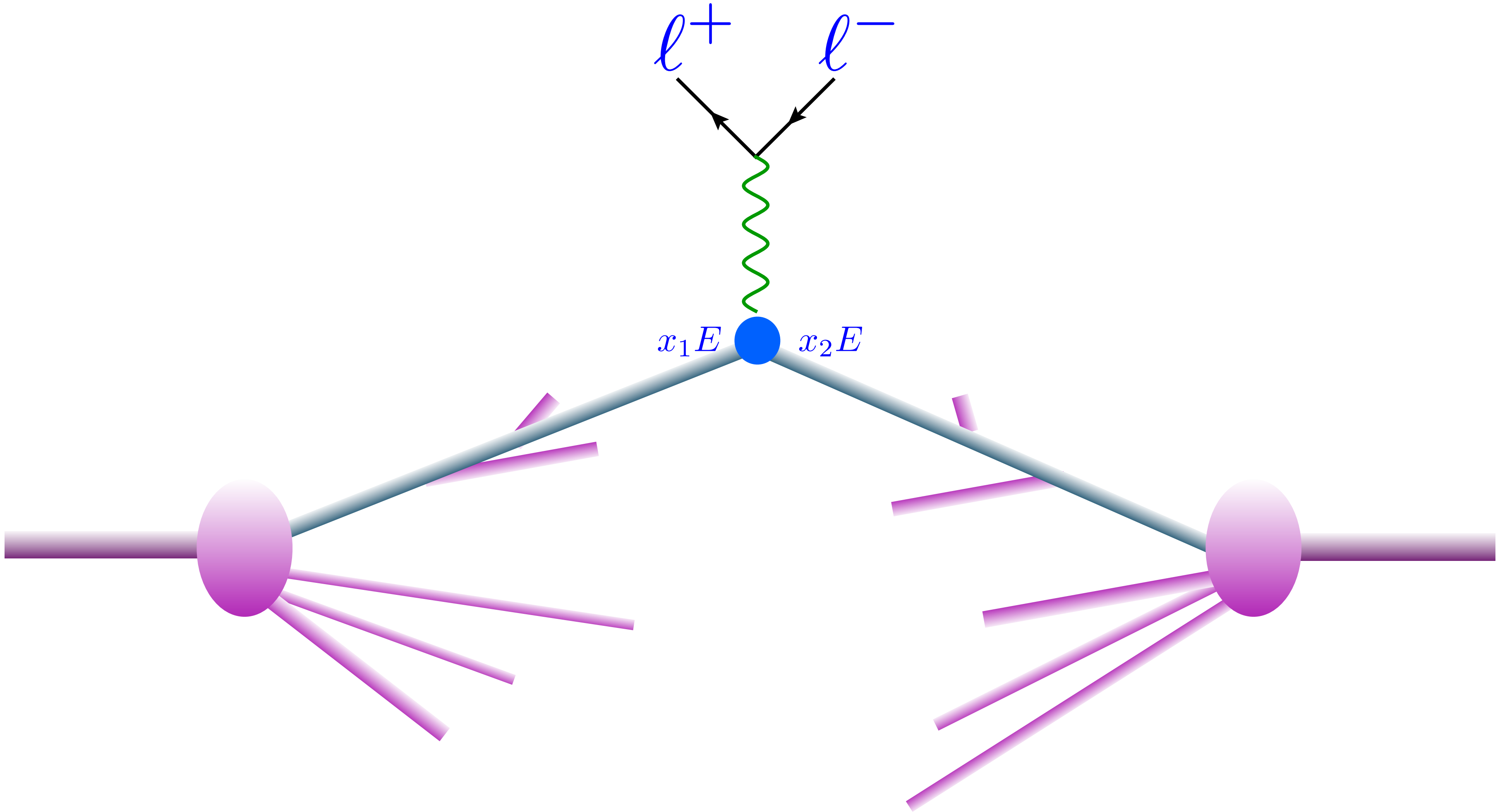
NLO corrections

- NLO corrections have three parts:
  - ➡ The Born contribution, i.e. the Leading order.
  - ➡ Virtual (or Loop) corrections: formed by an amplitude with a closed loop of particles interfered with the Born amplitudes
  - ➡ Real emission corrections: formed by amplitudes with one extra parton compared to the Born process
- Both Virtual and Real emission have one power of α_s extra compared to the Born process

$$\sigma^{\mathrm{NLO}} = \int_m d\sigma^B + \int_m d\sigma^V + \int_{m+1} d\sigma^R$$

NLO predictions

- As an example, consider Drell-Yan Z/γ* production



NLO predictions

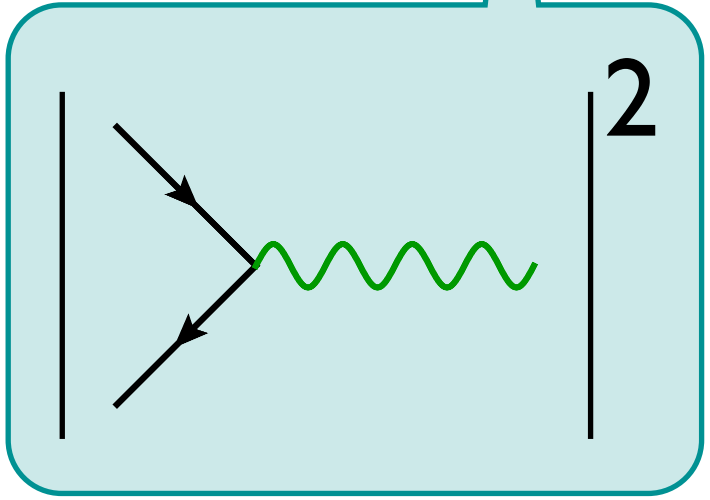
- As an example, consider Drell-Yan Z/γ* production

$$\hat{\sigma} = \sigma^{\mathrm{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \ldots\right)$$

NLO predictions

- As an example, consider Drell-Yan Z/γ* production

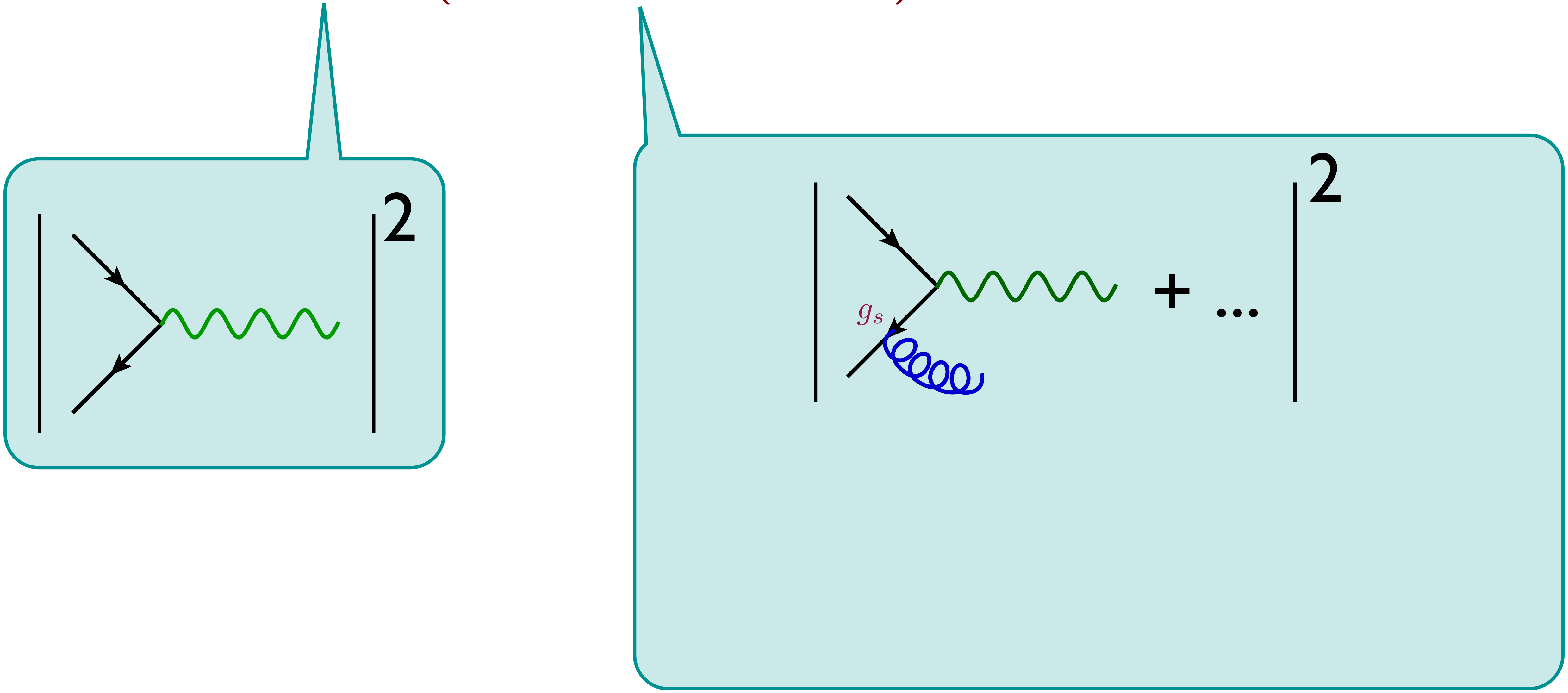
$$\hat{\sigma} = \sigma^{\mathrm{Born}} \left(1 + \frac{\alpha_s}{2\pi} \sigma^{(1)} + \dots \right)$$



NLO predictions

- As an example, consider Drell-Yan Z/γ* production

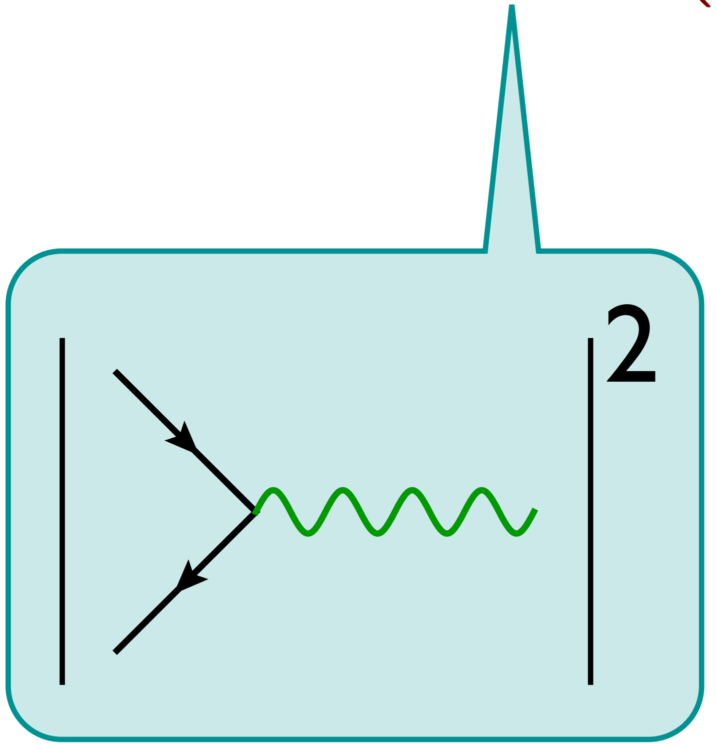
$$\hat{\sigma} = \sigma^{\mathrm{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \ldots\right)$$

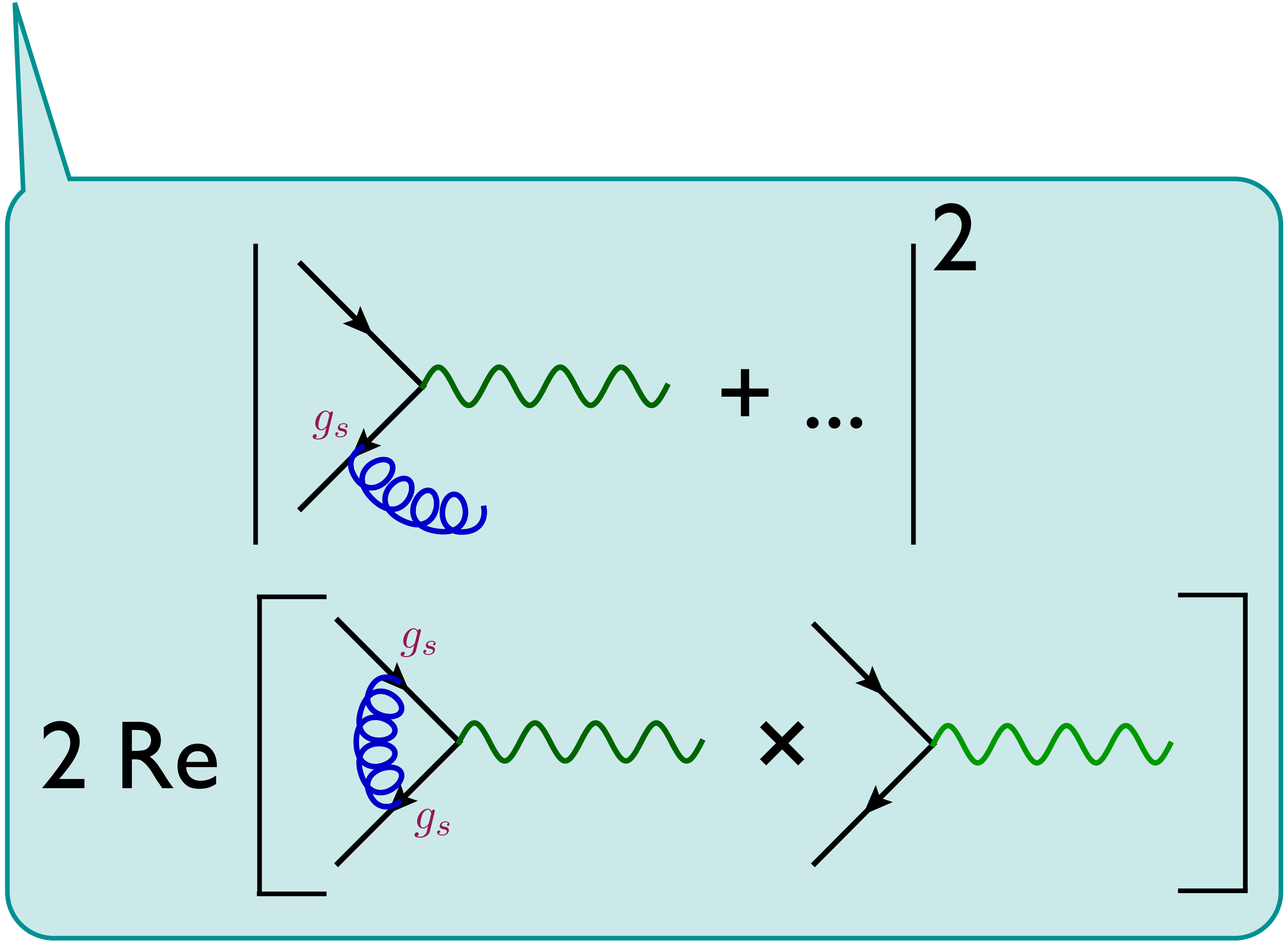


NLO predictions

- As an example, consider Drell-Yan Z/γ* production

$$\hat{\sigma} = \sigma^{\mathrm{Born}} \left(1 + \frac{\alpha_s}{2\pi} \sigma^{(1)} + \ldots \right)$$

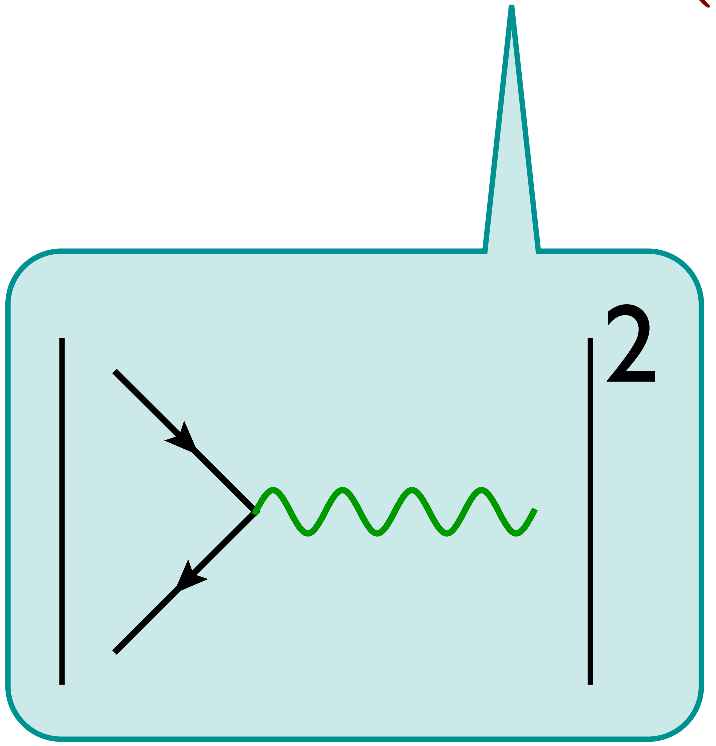


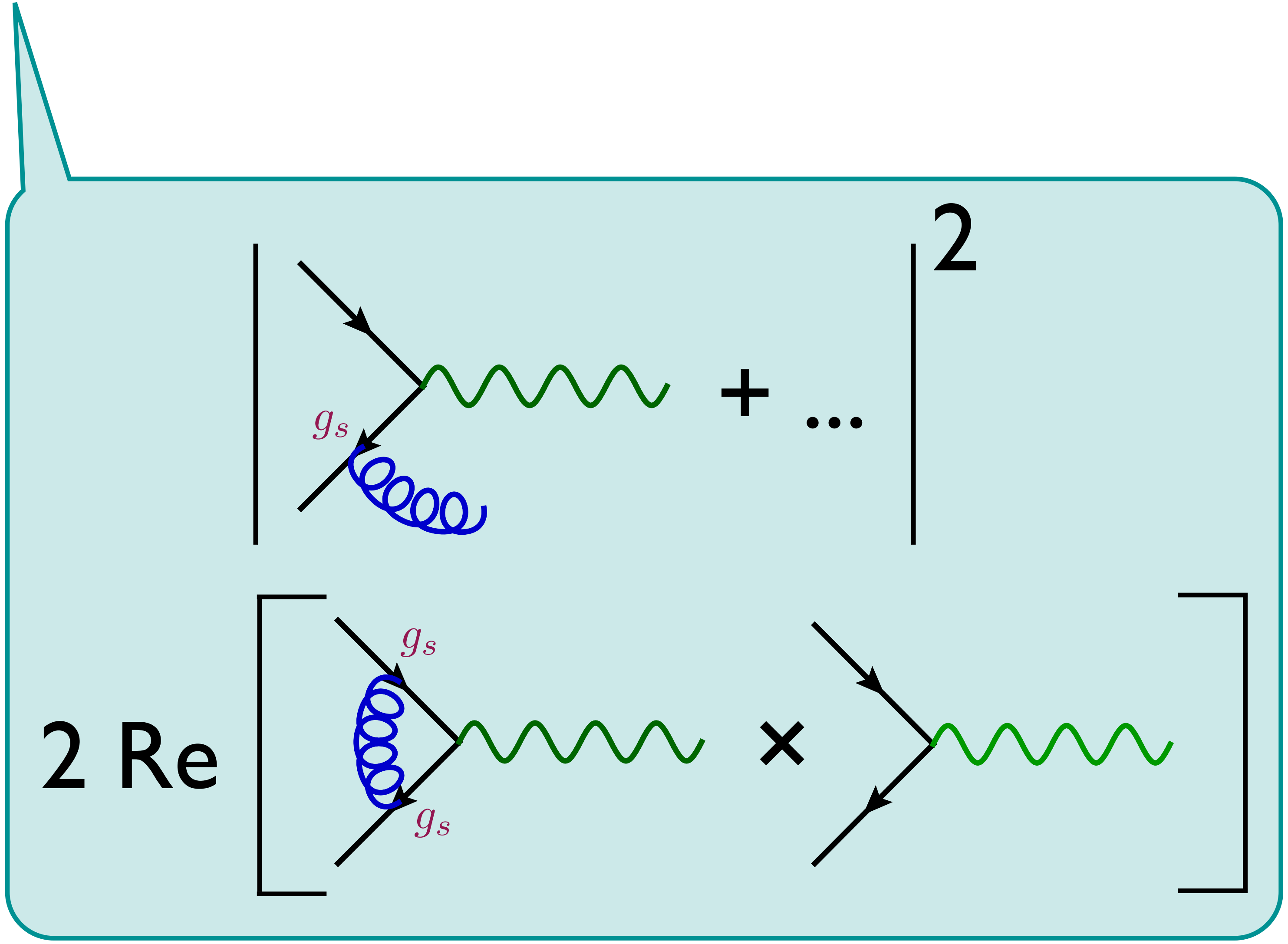


NLO predictions

- As an example, consider Drell-Yan Z/γ* production

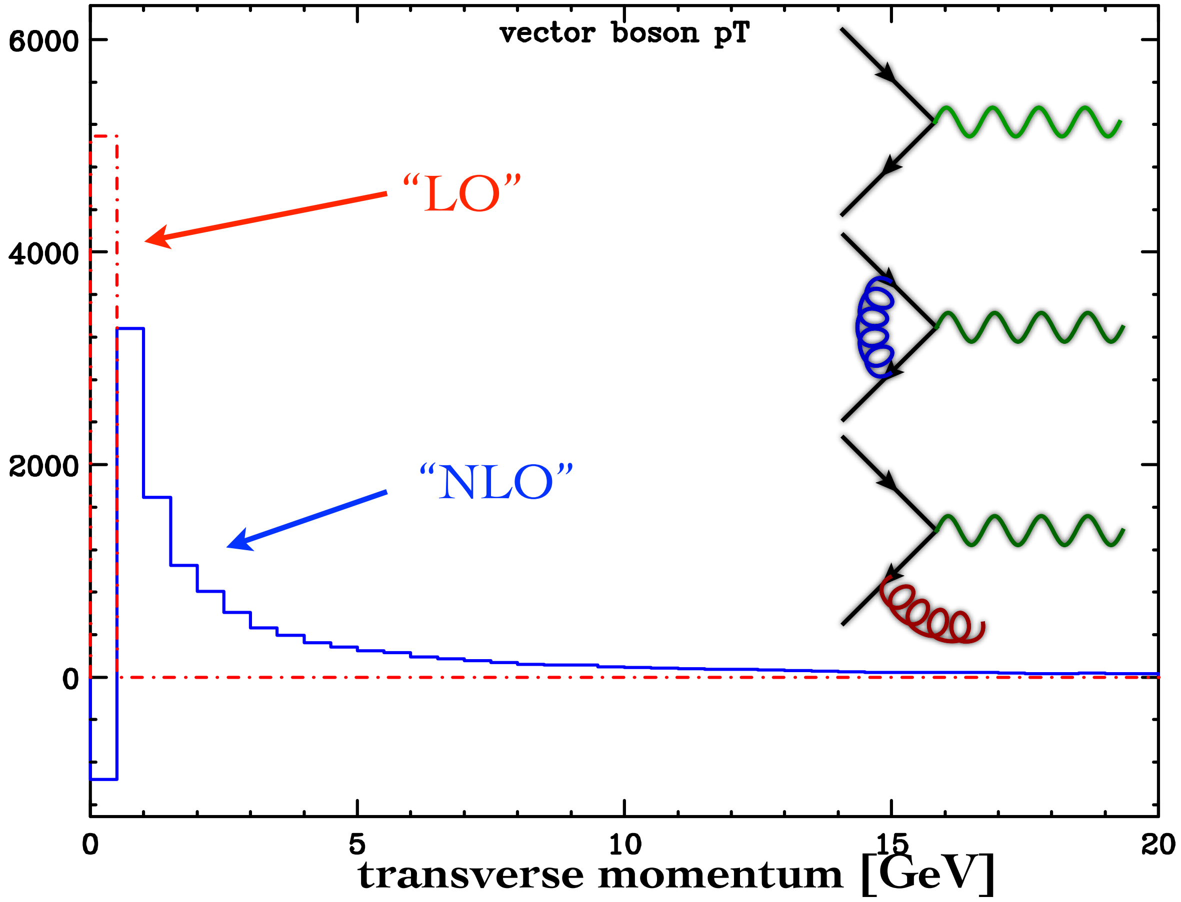
$$\hat{\sigma} = \sigma^{\text{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \dots\right)$$



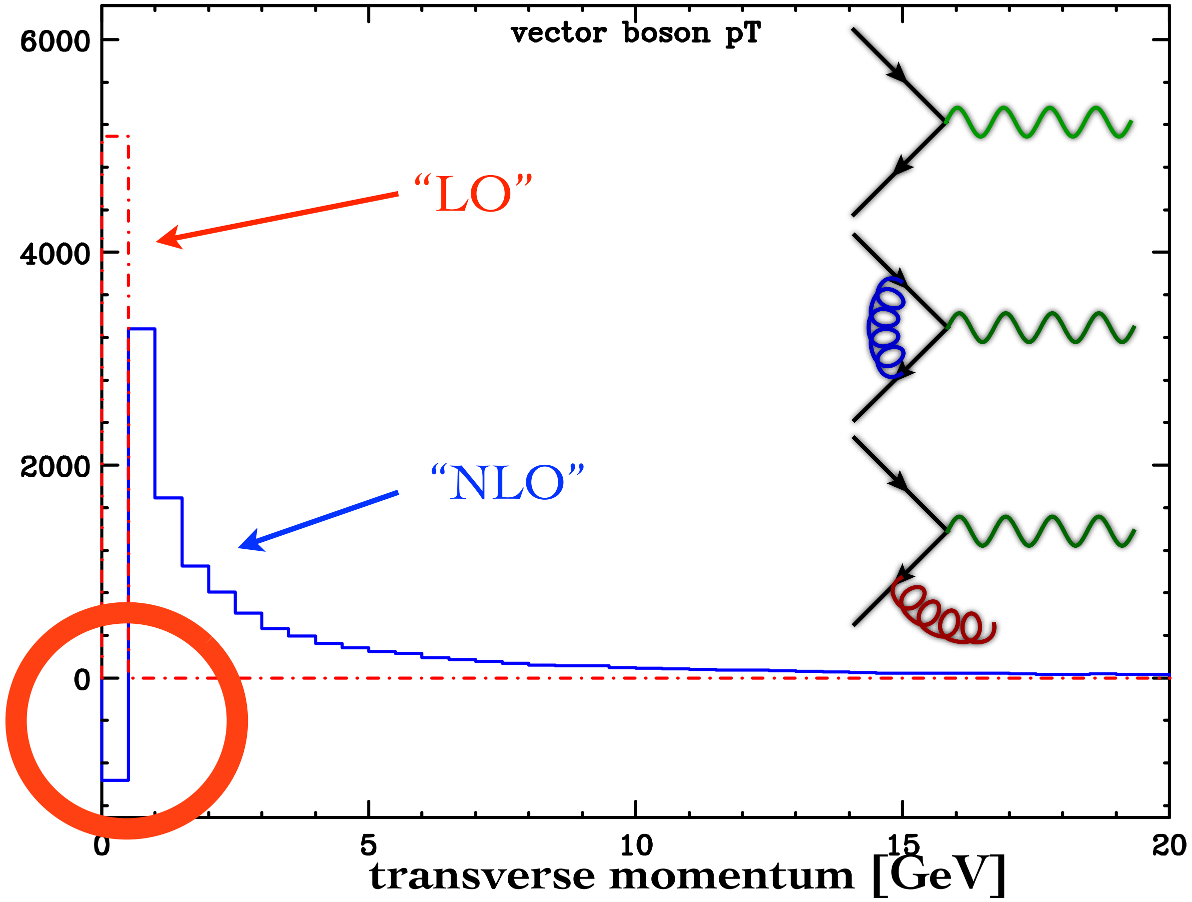


Not definite positive

Limitations of Fixed Order calculations

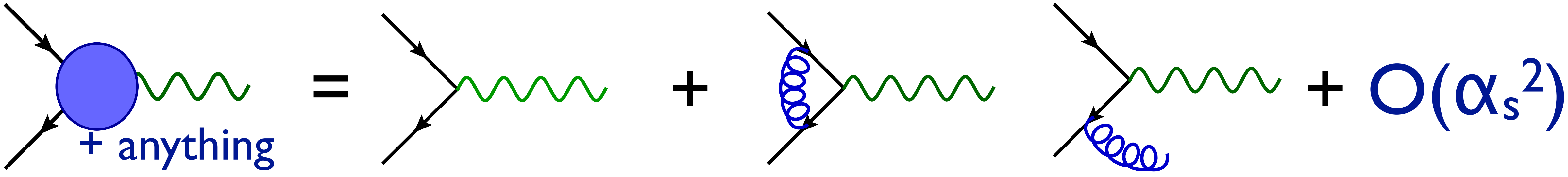


Limitations of Fixed Order calculations



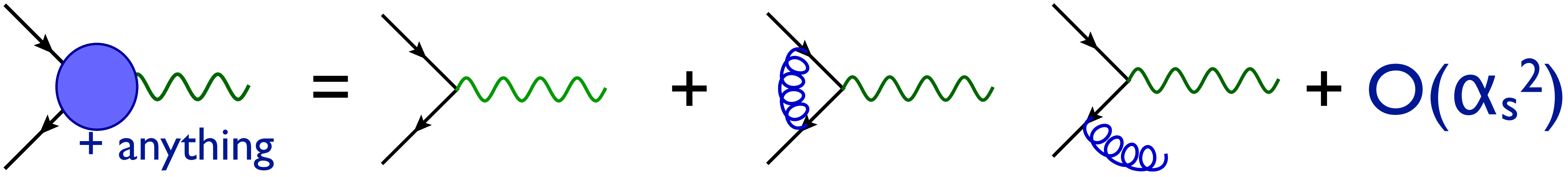
Negative contribution of the 0-bin

Difficulties



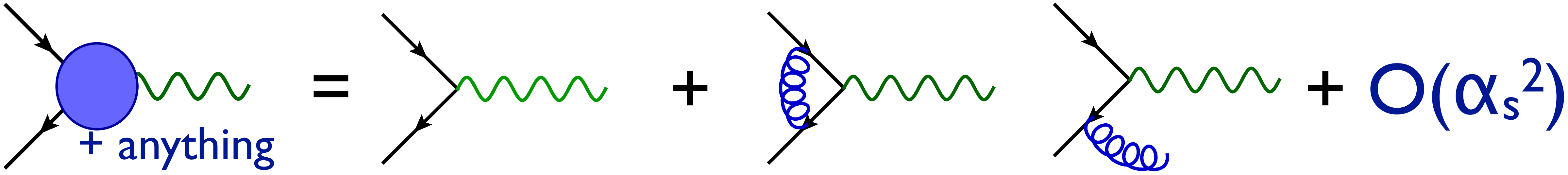
- Multiple steps
  - ➡ Fix divergencies
  - ➡ Virtual amplitudes: how to compute the loops automatically in a reasonable amount of time
  - ➡ How to deal with infra-red divergences: virtual corrections and real-emission corrections are separately divergent and only their sum is finite (for IR-safe observables) according to the KLN theorem
  - ➡ How to match these processes to a parton shower without double counting

Difficulties



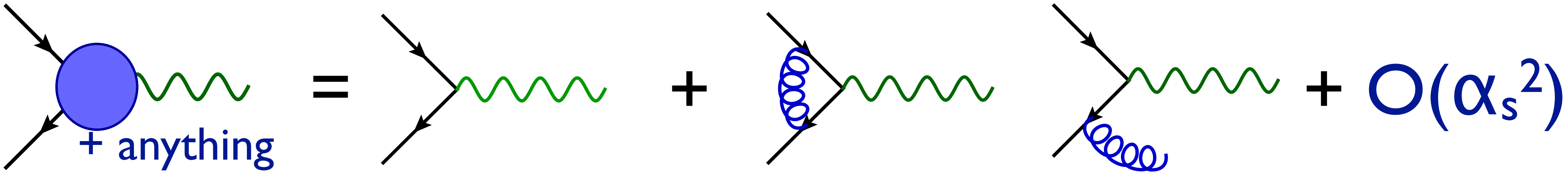
- Multiple steps
  - ➡ Fix divergencies — Next Section
  - ➡ Virtual amplitudes: how to compute the loops automatically in a reasonable amount of time
  - ➡ How to deal with infra-red divergences: virtual corrections and real-emission corrections are separately divergent and only their sum is finite (for IR-safe observables) according to the KLN theorem
  - ➡ How to match these processes to a parton shower without double counting

Difficulties



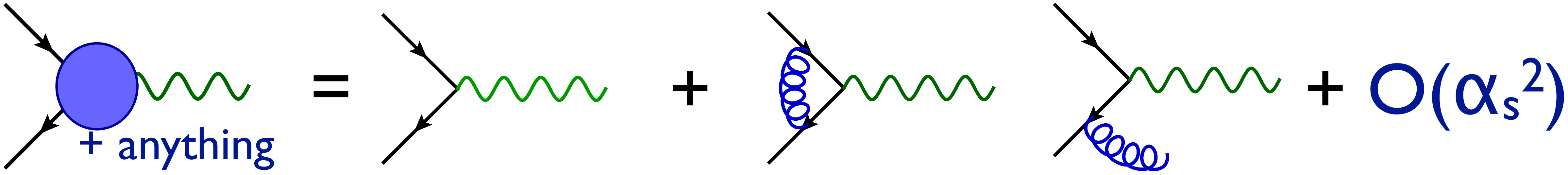
- Multiple steps
  - ➡ Fix divergencies — Next Section
  - ➡ Virtual amplitudes: how to compute the loops automatically in a reasonable amount of time — The section after
  - ➡ How to deal with infra-red divergences: virtual corrections and real-emission corrections are separately divergent and only their sum is finite (for IR-safe observables) according to the KLN theorem
  - ➡ How to match these processes to a parton shower without double counting

Difficulties



- Multiple steps
  - ➡ Fix divergencies **Next Section**
  - ➡ Virtual amplitudes: how to compute the loops automatically in a reasonable amount of time **The section after**
  - ➡ How to deal with infra-red divergences: virtual corrections and real-emission corrections are separately divergent and only their sum is finite (for IR-safe observables) according to the KLN theorem **The section after (bis)**
  - ➡ How to match these processes to a parton shower without double counting

Difficulties



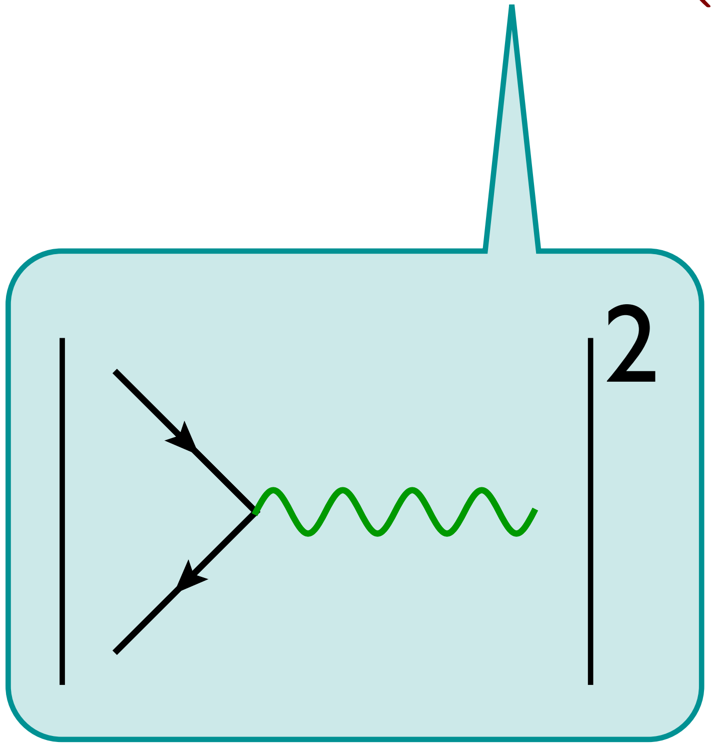
- Multiple steps
  - ➡ Fix divergencies **Next Section**
  - ➡ Virtual amplitudes: how to compute the loops automatically in a reasonable amount of time **The section after**
  - ➡ How to deal with infra-red divergences: virtual corrections and real-emission corrections are separately divergent and only their sum is finite (for IR-safe observables) according to the KLN theorem **The section after (bis)**
  - ➡ How to match these processes to a parton shower without double counting **The lecture before**

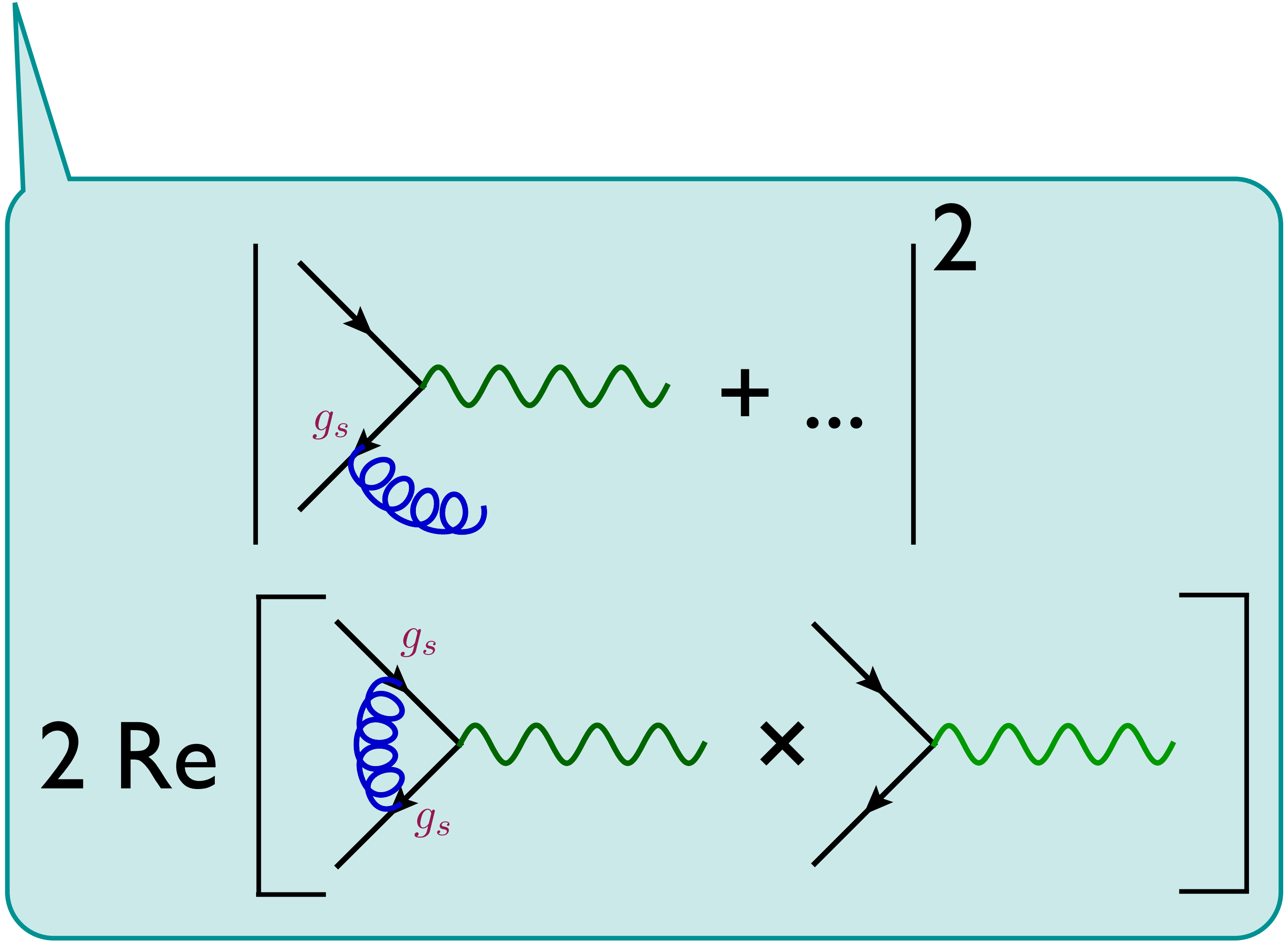
Canceling infrared divergences:

NLO predictions

- As an example, consider Drell-Yan production

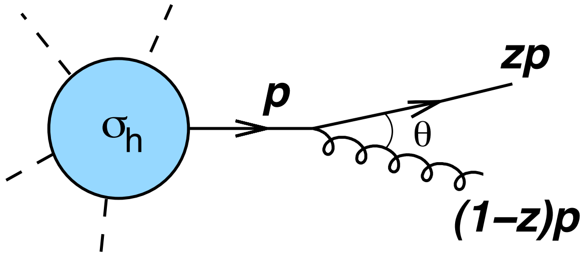
$$\hat{\sigma} = \sigma^{\text{Born}} \left(1 + \frac{\alpha_s}{2\pi} \sigma^{(1)} + \ldots \right)$$





Branching

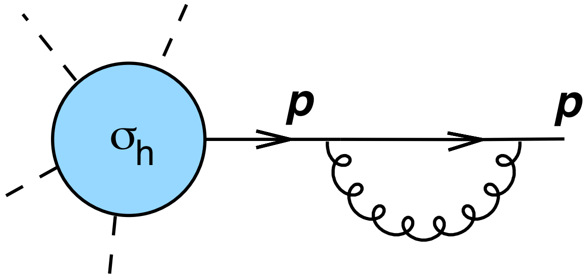
- In the soft and collinear region, the branching of a gluon from a quark can be written as



$$\sigma_{h+g} \simeq \sigma_h \frac{\alpha_s C_F}{\pi} \frac{dz}{1-z} \frac{dk_t^2}{k_t^2}$$

where k_t is the transverse momentum of the gluon, k_t=E sinθ.

- The singularities cancel against the singularities in the virtual corrections, which result from the integral over the loop momentum of the function



$$\sigma_{h+V} \simeq -\sigma_h \frac{\alpha_s C_F}{\pi} \frac{dz}{1-z} \frac{dk_t^2}{k_t^2}$$

Infrared cancellation

$$\sigma^{\mathrm{NLO}} \sim \int d^4\Phi_m \, B(\Phi_m) + \int d^4\Phi_m \int_{\mathrm{loop}} d^d l \, V(\Phi_m) + \int d^d\Phi_{m+1} \, R(\Phi_{m+1})$$

- The KLN theorem tells us that divergences from virtual and real-emission corrections cancel in the sum for observables insensitive to soft and collinear radiation (“IR-safe observables”)
- When doing an analytic calculation in dimensional regularization this can be explicitly seen in the cancellation of the $1/\epsilon$ and $1/\epsilon^2$ terms (with ϵ the regulator, $\epsilon \rightarrow 0$)
- In the real emission corrections, the explicit poles enter after the phase-space integration (in d dimensions)

Infrared safe observables

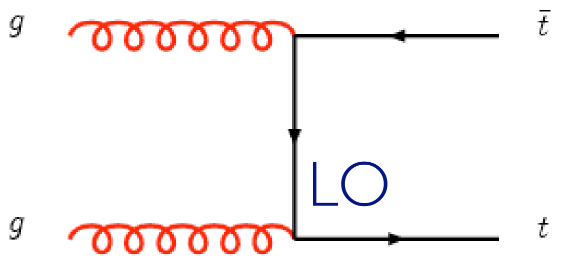
- For an observable to be calculable in fixed-order perturbation theory, the observable should be infrared safe, i.e., it should be insensitive to the emission of soft or collinear partons.
- In particular, if p_i is a momentum occurring in the definition of an observable, it most be invariant under the branching

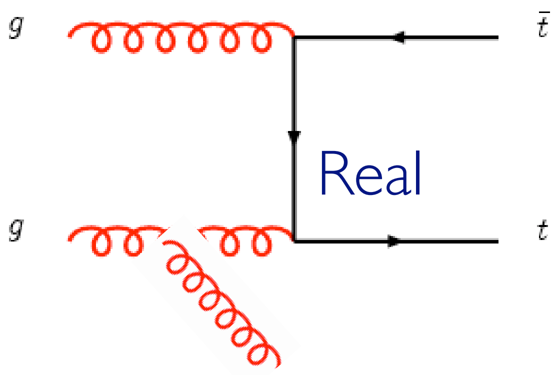
  $$p_i \longrightarrow p_j + p_k,$$

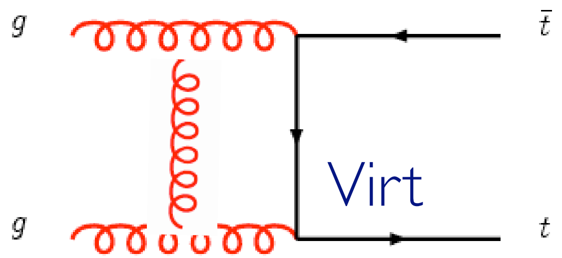
  whenever p_j and p_k are collinear or one of them is soft.
- Examples
  - ➡ "The number of gluons" produced in a collision is not an infrared safe observable

NLO...?

- Are all (IR-safe) observables that we can compute using a NLO code correctly described at NLO? Suppose we have a NLO code for pp ⟶ ttbar



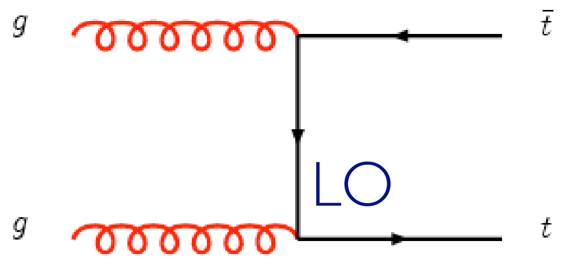


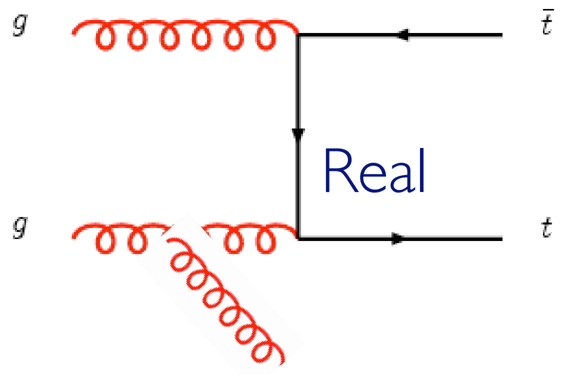


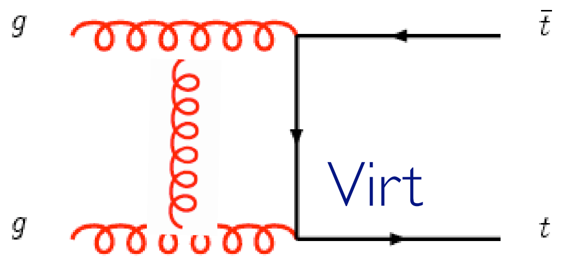
  - ➡ Total cross section
  - ➡ Transverse momentum of the top quark
  - ➡ Transverse momentum of the top-antitop pair
  - ➡ Transverse momentum of the jet
  - ➡ Top-antitop invariant mass
  - ➡ Azimuthal distance between the top and anti-top

NLO...?

- Are all (IR-safe) observables that we can compute using a NLO code correctly described at NLO? Suppose we have a NLO code for pp ⟶ ttbar





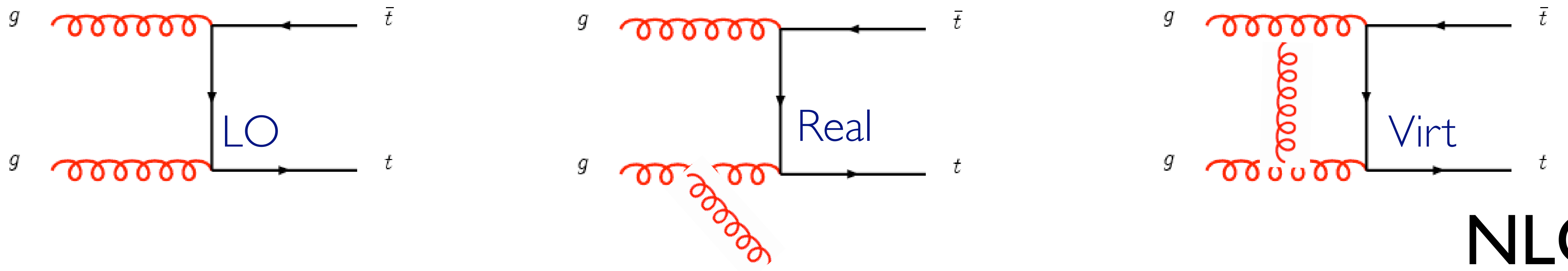


NLO?

  - ➡ Total cross section
  - ➡ Transverse momentum of the top quark
  - ➡ Transverse momentum of the top-antitop pair
  - ➡ Transverse momentum of the jet
  - ➡ Top-antitop invariant mass
  - ➡ Azimuthal distance between the top and anti-top

NLO...?

- Are all (IR-safe) observables that we can compute using a NLO code correctly described at NLO? Suppose we have a NLO code for pp ⟶ ttbar

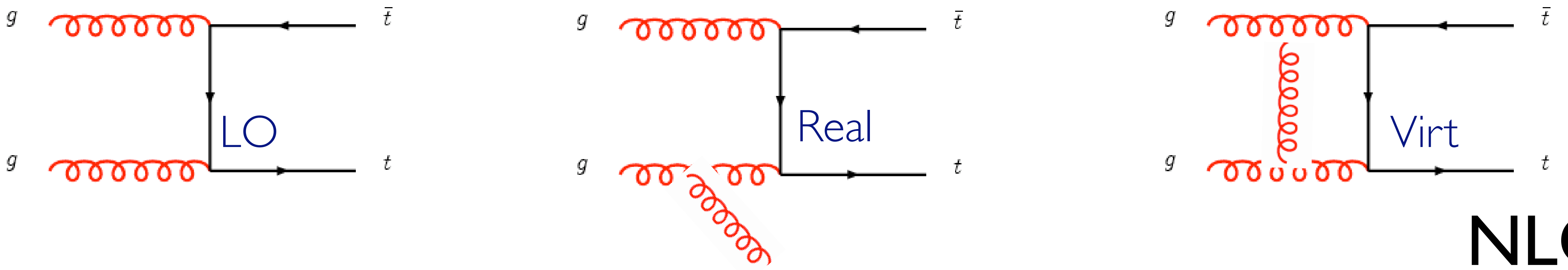


NLO?

- ➡ Total cross section ✔
- ➡ Transverse momentum of the top quark
- ➡ Transverse momentum of the top-antitop pair
- ➡ Transverse momentum of the jet
- ➡ Top-antitop invariant mass
- ➡ Azimuthal distance between the top and anti-top

NLO...?

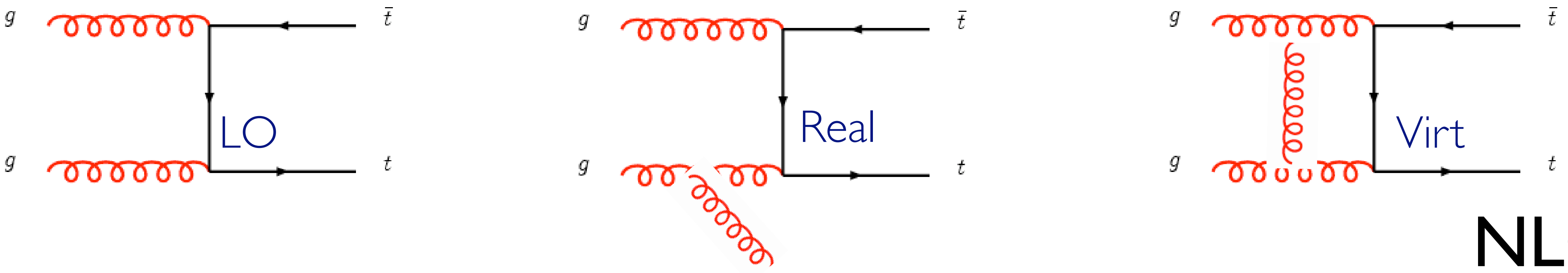
- Are all (IR-safe) observables that we can compute using a NLO code correctly described at NLO? Suppose we have a NLO code for pp ⟶ ttbar



	NLO?
➡ Total cross section	✔
➡ Transverse momentum of the top quark	✔
➡ Transverse momentum of the top-antitop pair	
➡ Transverse momentum of the jet	
➡ Top-antitop invariant mass	
➡ Azimuthal distance between the top and anti-top	

NLO...?

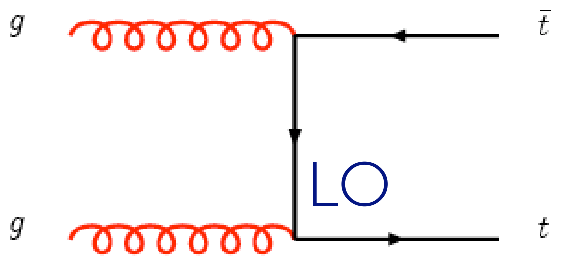
- Are all (IR-safe) observables that we can compute using a NLO code correctly described at NLO? Suppose we have a NLO code for pp ⟶ ttbar

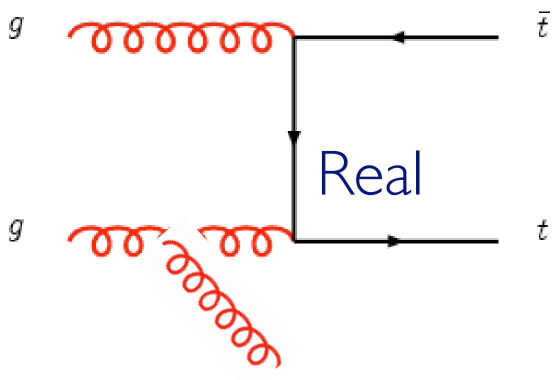


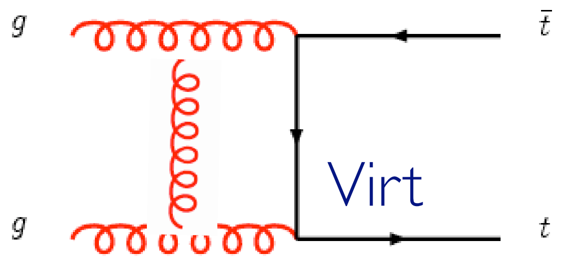
	NLO?
➡ Total cross section	✔
➡ Transverse momentum of the top quark	✔
➡ Transverse momentum of the top-antitop pair	✘
➡ Transverse momentum of the jet	
➡ Top-antitop invariant mass	
➡ Azimuthal distance between the top and anti-top	

NLO...?

- Are all (IR-safe) observables that we can compute using a NLO code correctly described at NLO? Suppose we have a NLO code for pp ⟶ ttbar





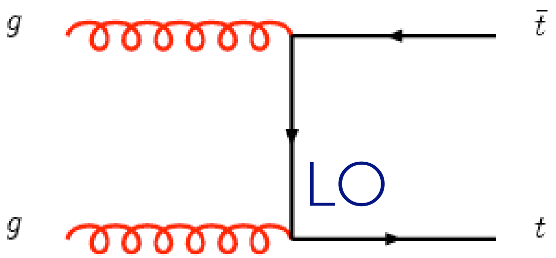


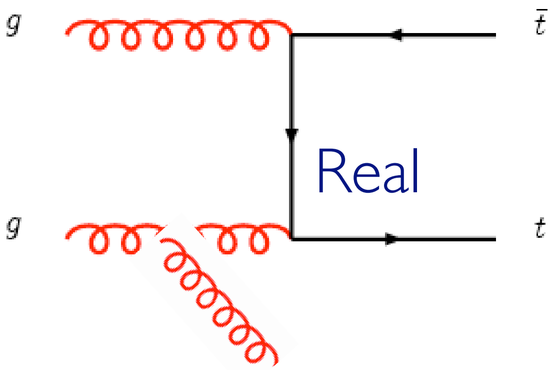
NLO?

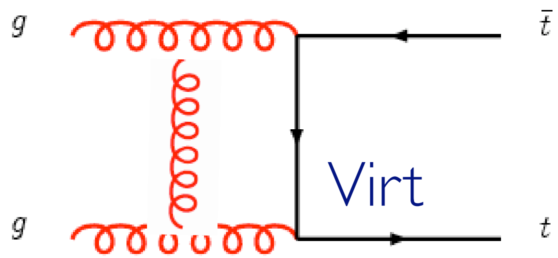
- ➡ Total cross section ✓
- ➡ Transverse momentum of the top quark ✓
- ➡ Transverse momentum of the top-antitop pair ✗
- ➡ Transverse momentum of the jet ✗
- ➡ Top-antitop invariant mass
- ➡ Azimuthal distance between the top and anti-top

NLO...?

- Are all (IR-safe) observables that we can compute using a NLO code correctly described at NLO? Suppose we have a NLO code for pp ⟶ ttbar



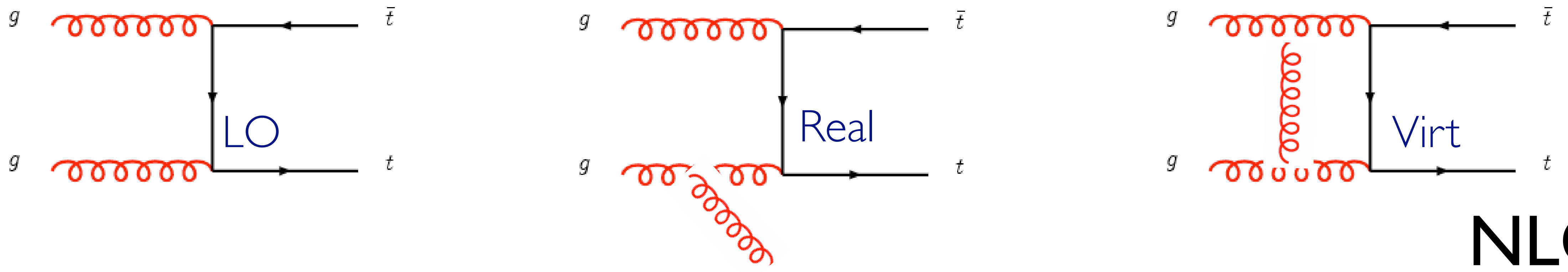




	NLO?
➡ Total cross section	✔
➡ Transverse momentum of the top quark	✔
➡ Transverse momentum of the top-antitop pair	✘
➡ Transverse momentum of the jet	✘
➡ Top-antitop invariant mass	✔
➡ Azimuthal distance between the top and anti-top	

NLO...?

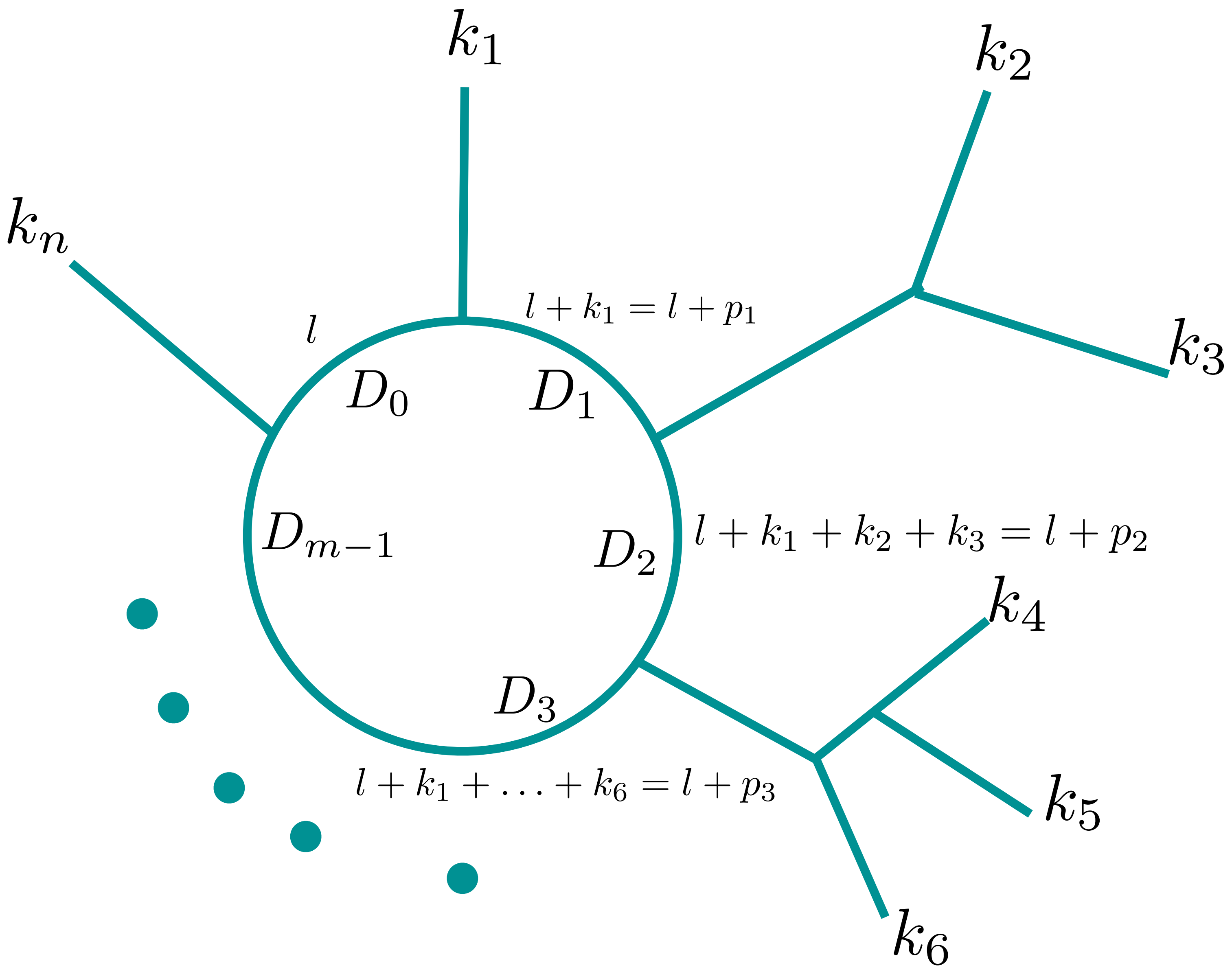
- Are all (IR-safe) observables that we can compute using a NLO code correctly described at NLO? Suppose we have a NLO code for pp ⟶ ttbar



	NLO?
➡ Total cross section	✔
➡ Transverse momentum of the top quark	✔
➡ Transverse momentum of the top-antitop pair	✘
➡ Transverse momentum of the jet	✘
➡ Top-antitop invariant mass	✔
➡ Azimuthal distance between the top and anti-top	✘

Loop Computation

one-loop integral



- Consider this *m*-point loop diagram with *n* external momenta
- The integral to compute is

$$\int d^d l \, \frac{N(l)}{D_0 D_1 D_2 \cdots D_{m-1}}$$

$$D_i = (l+p_i)^2 - m_i^2$$

loop techniques

- Passarino-Veltman Reduction
- The "loop revolution": new techniques for computing one-loop matrix elements are now established:
  - ➡ Generalized unitarity (e.g. BlackHat, Rocket, ...)

    [Bern, Dixon, Dunbar, Kosower, 1994...; Ellis Giele Kunst 2007 + Melnikov 2008;...]
  - ➡ Integrand reduction (e.g. CutTools, GoSam)

    [Ossola, Papadopoulos, Pittau 2006; del Aguila, Pittau 2004; Mastrolia, Ossola, Reiter, Tramontano 2010;...]
  - ➡ Tensor reduction (e.g. Golem)

    [Passarino, Veltman 1979; Denner, Dittmaier 2005; Binoth Guillet, Heinrich, Pilon, Reiter 2008]

loop techniques

- Passarino-Veltman Reduction
- The "loop revolution": new techniques for computing one-loop matrix elements are now established:
  - ➡ Generalized unitarity (e.g. BlackHat, Rocket, ...)

    [Bern, Dixon, Dunbar, Kosower, 1994...; Ellis Giele Kunst 2007 + Melnikov 2008;...]
  - ➡ Integrand reduction (e.g. CutTools, GoSam)

    [Ossola, Papadopoulos, Pittau 2006; del Aguila, Pittau 2004; Mastrolia, Ossola, Reiter, Tramontano 2010;...]
  - ➡ Tensor reduction (e.g. Golem)

    [Passarino, Veltman 1979; Denner, Dittmaier 2005; Binoth Guillet, Heinrich, Pilon, Reiter 2008]

Integrand reduction

- Any one-loop integral can be decomposed in scalar integrals
- The task is to find these coefficients efficiently (analytically or numerically)

Basis of scalar integrals

$$\mathcal{M}^{\text{1-loop}} = \sum_{i_0<i_1<i_2<i_3} d_{i_0 i_1 i_2 i_3} \mathrm{Box}_{i_0 i_1 i_2 i_3} + \sum_{i_0<i_1<i_2} c_{i_0 i_1 i_2} \mathrm{Triangle}_{i_0 i_1 i_2} + \sum_{i_0<i_1} b_{i_0 i_1} \mathrm{Bubble}_{i_0 i_1} + \sum_{i_0} a_{i_0} \mathrm{Tadpole}_{i_0} + R + \mathcal{O}(\epsilon)$$

- The a, b, c, d and R coefficients depend only on external parameters and momenta

$$D_i = (l + p_i)^2 - m_i^2$$

$$\mathrm{Tadpole}_{i_0} = \int d^d l \frac{1}{D_{i_0}}$$

$$\mathrm{Bubble}_{i_0 i_1} = \int d^d l \frac{1}{D_{i_0} D_{i_1}}$$

$$\mathrm{Triangle}_{i_0 i_1 i_2} = \int d^d l \frac{1}{D_{i_0} D_{i_1} D_{i_2}}$$

$$\mathrm{Box}_{i_0 i_1 i_2 i_3} = \int d^d l \frac{1}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}}$$

- All these scalar integrals are known and available in computer libraries (FF [v. Oldenborgh], QCDLoop [Ellis, Zanderighi], OneLOop [v. Hameren])

Divergences

$$\mathcal{M}^{\text{1-loop}} = \sum_{i_0<i_1<i_2<i_3} d_{i_0 i_1 i_2 i_3} \text{Box}_{i_0 i_1 i_2 i_3} + \sum_{i_0<i_1<i_2} c_{i_0 i_1 i_2} \text{Triangle}_{i_0 i_1 i_2} + \sum_{i_0<i_1} b_{i_0 i_1} \text{Bubble}_{i_0 i_1} + \sum_{i_0} a_{i_0} \text{Tadpole}_{i_0} + R + \mathcal{O}(\epsilon)$$

$$D_i = (l+p_i)^2 - m_i^2$$

$$\text{Tadpole}_{i_0} = \int d^d l \frac{1}{D_{i_0}}$$

$$\text{Bubble}_{i_0 i_1} = \int d^d l \frac{1}{D_{i_0} D_{i_1}}$$

$$\text{Triangle}_{i_0 i_1 i_2} = \int d^d l \frac{1}{D_{i_0} D_{i_1} D_{i_2}}$$

$$\text{Box}_{i_0 i_1 i_2 i_3} = \int d^d l \frac{1}{D_{i_0} D_{i_1} D_{i_2} D_{i_3}}$$

➡The coefficients d, c, b and a are finite and do not contain poles in 1/ϵ

➡The 1/ϵ dependence is in the scalar integrals (and the UV renormalization)

➡When we have solved this system (and included the UV renormalization) we have the full dependence on the soft/collinear divergences in terms of coefficients in front of the poles. These divergences should cancel against divergences in the real emission corrections (according to KLN theorem)

$$\text{Virtual} \sim v_0 + \frac{v_1}{\epsilon} + \frac{v_2}{\epsilon^2}$$

OPP Reduction

- The decomposition to scalar integrals presented before works at the level of the integrals

$$\begin{aligned}\mathcal{M}^{\text{1-loop}} = & \sum_{i_0<i_1<i_2<i_3} d_{i_0 i_1 i_2 i_3} \text{Box}_{i_0 i_1 i_2 i_3} \\ & + \sum_{i_0<i_1<i_2} c_{i_0 i_1 i_2} \text{Triangle}_{i_0 i_1 i_2} \\ & + \sum_{i_0<i_1} b_{i_0 i_1} \text{Bubble}_{i_0 i_1} \\ & + \sum_{i_0} a_{i_0} \text{Tadpole}_{i_0} \\ & + R + \mathcal{O}(\epsilon)\end{aligned}$$

OPP Reduction

- The decomposition to scalar integrals presented before works at the level of the integrals

- If we would know a similar relation at the **integrand** level, we would be able to manipulate the integrands and extract the coefficients without doing the integrals

$$\begin{aligned}\mathcal{M}^{\text{1-loop}} = &\sum_{i_0<i_1<i_2<i_3} d_{i_0 i_1 i_2 i_3} \text{Box}_{i_0 i_1 i_2 i_3} \\ &+ \sum_{i_0<i_1<i_2} c_{i_0 i_1 i_2} \text{Triangle}_{i_0 i_1 i_2} \\ &+ \sum_{i_0<i_1} b_{i_0 i_1} \text{Bubble}_{i_0 i_1} \\ &+ \sum_{i_0} a_{i_0} \text{Tadpole}_{i_0} \\ &+ R + \mathcal{O}(\epsilon)\end{aligned}$$

OPP Reduction

- The decomposition to scalar integrals presented before works at the level of the integrals

- If we would know a similar relation at the **integrand** level, we would be able to manipulate the integrands and extract the coefficients without doing the integrals

$$\mathcal{M}^{\text{1-loop}} = \sum_{i_0<i_1<i_2<i_3} d_{i_0i_1i_2i_3} \text{Box}_{i_0i_1i_2i_3} + \sum_{i_0<i_1<i_2} c_{i_0i_1i_2} \text{Triangle}_{i_0i_1i_2} + \sum_{i_0<i_1} b_{i_0i_1} \text{Bubble}_{i_0i_1} + \sum_{i_0} a_{i_0} \text{Tadpole}_{i_0} + R + \mathcal{O}(\epsilon)$$

$$N(l) = \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0i_1i_2i_3} + \tilde{d}_{i_0i_1i_2i_3}(l)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i + \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0i_1i_2} + \tilde{c}_{i_0i_1i_2}(l)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i + \sum_{i_0<i_1}^{m-1} \left[b_{i_0i_1} + \tilde{b}_{i_0i_1}(l)\right] \prod_{i\neq i_0,i_1}^{m-1} D_i + \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l)\right] \prod_{i\neq i_0}^{m-1} D_i + \tilde{P}(l) \prod_{i}^{m-1} D_i$$

OPP Reduction

- The decomposition to scalar integrals presented before works at the level of the integrals

$$\mathcal{M}^{\text{1-loop}} = \sum_{i_0<i_1<i_2<i_3} d_{i_0 i_1 i_2 i_3} \text{Box}_{i_0 i_1 i_2 i_3} + \sum_{i_0<i_1<i_2} c_{i_0 i_1 i_2} \text{Triangle}_{i_0 i_1 i_2} + \sum_{i_0<i_1} b_{i_0 i_1} \text{Bubble}_{i_0 i_1} + \sum_{i_0} a_{i_0} \text{Tadpole}_{i_0} + R + \mathcal{O}(\epsilon)$$

- If we would know a similar relation at the **integrand** level, we would be able to manipulate the integrands and extract the coefficients without doing the integrals

$$\begin{aligned} N(l) = & \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \boxed{\tilde{d}_{i_0 i_1 i_2 i_3}(l)}\right] \prod_{i \neq i_0, i_1, i_2, i_3}^{m-1} D_i \\ & + \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \boxed{\tilde{c}_{i_0 i_1 i_2}(l)}\right] \prod_{i \neq i_0, i_1, i_2}^{m-1} D_i \\ & + \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \boxed{\tilde{b}_{i_0 i_1}(l)}\right] \prod_{i \neq i_0, i_1}^{m-1} D_i \\ & + \sum_{i_0}^{m-1} \left[a_{i_0} + \boxed{\tilde{a}_{i_0}(l)}\right] \prod_{i \neq i_0}^{m-1} D_i \\ & + \tilde{P}(l) \prod_{i}^{m-1} D_i \end{aligned}$$

Spurious term

Functional form of the spurious terms

- The functional form of the spurious terms is known (it depends on the rank of the integral and the number of propagators in the loop) [del Aguila, Pittau 2004]
  - ➡ for example, a box coefficient from a rank 1 numerator is

$$\tilde{d}_{i_0 i_1 i_2 i_3}(l) = \tilde{d}_{i_0 i_1 i_2 i_3}\, \epsilon^{\mu\nu\rho\sigma}\, l^\mu p_1^\nu p_2^\rho p_3^\sigma$$

(remember that p_i is the sum of the momentum that has entered the loop so far, so we always have $p_0 = 0$)

  - ➡ The integral is zero

$$\int d^d l \frac{\tilde{d}_{i_0 i_1 i_2 i_3}(l)}{D_0 D_1 D_2 D_3} = \tilde{d}_{i_0 i_1 i_2 i_3} \int d^d l \frac{\epsilon^{\mu\nu\rho\sigma}\, l^\mu p_1^\nu p_2^\rho p_3^\sigma}{D_0 D_1 D_2 D_3} = 0$$

Numerical evaluation

- By choosing specific values for the loop momentum *l*, we end up with a system of linear equations
    - ➡ In a renormalizable theory, the rank of the integrand is always smaller (or equal) to the number of particles in the loop (with a conveniently chosen gauge)
    - ➡ We can straight-forwardly set the it up by sampling the numerator numerically for various values of the loop momentum *l*
    - ➡ By choosing *l* smartly, the system greatly reduces
        - ✦ In particular when we chose *l* to be a complex 4-vector

How it works...

$$
\begin{aligned}
N(l) = &\sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i \\
+ &\sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i \\
+ &\sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l)\right] \prod_{i\neq i_0,i_1}^{m-1} D_i \\
+ &\sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l)\right] \prod_{i\neq i_0}^{m-1} D_i \\
+ &\tilde{P}(l) \prod_{i}^{m-1} D_i
\end{aligned}
$$

To solve the OPP reduction, choosing special values for the loop momenta helps a lot

For example, choosing *l* such that

$$D_0(l^\pm) = D_1(l^\pm) = D_2(l^\pm) = D_3(l^\pm) = 0$$

sets all the terms in this equation to zero except the first line

There are two (complex) solutions to this equation due to the quadratic nature of the propagators

How it works...

$$
\begin{aligned}
N(l) = & \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i \\
& + \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i \\
& + \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l)\right] \prod_{i\neq i_0,i_1}^{m-1} D_i \\
& + \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l)\right] \prod_{i\neq i_0}^{m-1} D_i \\
& + \tilde{P}(l) \prod_{i}^{m-1} D_i
\end{aligned}
$$

To solve the OPP reduction, choosing special values for the loop momenta helps a lot

For example, choosing *l* such that

$$D_0(l^{\pm}) = D_1(l^{\pm}) = D_2(l^{\pm}) = D_3(l^{\pm}) = 0$$

sets all the terms in this equation to zero except the first line

There are two (complex) solutions to this equation due to the quadratic nature of the propagators

How it works...

$$N(l) = \boxed{\sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0i_1i_2i_3} + \tilde{d}_{i_0i_1i_2i_3}(l)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i}$$
$$+ \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0i_1i_2} + \tilde{c}_{i_0i_1i_2}(l)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i$$
$$+ \sum_{i_0<i_1}^{m-1} \left[b_{i_0i_1} + \tilde{b}_{i_0i_1}(l)\right] \prod_{i\neq i_0,i_1}^{m-1} D_i$$
$$+ \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l)\right] \prod_{i\neq i_0}^{m-1} D_i$$
$$+\tilde{P}(l) \prod_{i}^{m-1} D_i$$

To solve the OPP reduction, choosing special values for the loop momenta helps a lot

For example, choosing *l* such that

$$D_0(l^i) = D_1(l^i) = D_2(l^i) = 0$$

sets all the terms in this equation to zero except the first and second line

How it works...

$$\begin{aligned} N(l) = & \boxed{\sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)\right] \prod_{i \neq i_0, i_1, i_2, i_3}^{m-1} D_i} \\ & + \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l)\right] \prod_{i \neq i_0, i_1, i_2}^{m-1} D_i \\ & + \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l)\right] \prod_{i \neq i_0, i_1}^{m-1} D_i \\ & + \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l)\right] \prod_{i \neq i_0}^{m-1} D_i \\ & + \tilde{P}(l) \prod_{i}^{m-1} D_i \end{aligned}$$

To solve the OPP reduction, choosing special values for the loop momenta helps a lot

For example, choosing *l* such that

$$D_0(l^i) = D_1(l^i) = D_2(l^i) = 0$$

sets all the terms in this equation to zero except the first and second line

How it works...

$$
\begin{aligned}
N(l) = & \boxed{\sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0i_1i_2i_3} + \tilde{d}_{i_0i_1i_2i_3}(l)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i} \\
+ & \boxed{\sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0i_1i_2} + \tilde{c}_{i_0i_1i_2}(l)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i} \\
+ & \sum_{i_0<i_1}^{m-1} \left[b_{i_0i_1} + \tilde{b}_{i_0i_1}(l)\right] \prod_{i\neq i_0,i_1}^{m-1} D_i \\
+ & \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l)\right] \prod_{i\neq i_0}^{m-1} D_i \\
+ & \tilde{P}(l) \prod_{i}^{m-1} D_i
\end{aligned}
$$

To solve the OPP reduction, choosing special values for the loop momenta helps a lot

For example, choosing *l* such that

$$D_0(l^i) = D_1(l^i) = 0$$

sets all the terms in this equation to zero except the first, second and third line

How it works...

$$
\begin{aligned}
N(l) =& \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)\right] \prod_{i \neq i_0, i_1, i_2, i_3}^{m-1} D_i \\
&+ \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l)\right] \prod_{i \neq i_0, i_1, i_2}^{m-1} D_i \\
&+ \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l)\right] \prod_{i \neq i_0, i_1}^{m-1} D_i \\
&+ \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l)\right] \prod_{i \neq i_0}^{m-1} D_i \\
&+ \tilde{P}(l) \prod_{i}^{m-1} D_i
\end{aligned}
$$

To solve the OPP reduction, choosing special values for the loop momenta helps a lot

For example, choosing *l* such that

$$D_0(l^i) = D_1(l^i) = 0$$

sets all the terms in this equation to zero except the first, second and third line

How it works...

- For each phase-space point we have to solve the system of equations
- Due to the fact that the system reduces when picking special values for the loop momentum, the system greatly reduces
- We can decompose the system at the level of the squared matrix element, amplitude, diagram or anywhere in between. As long as we provide the corresponding numerator function. Since each reduction with CutTools is computationally heavy, we directly reduce the squared element with MadGraph.
- For a given phase-space point, we have to compute the numerator function several times (~50 or so for a box loop)

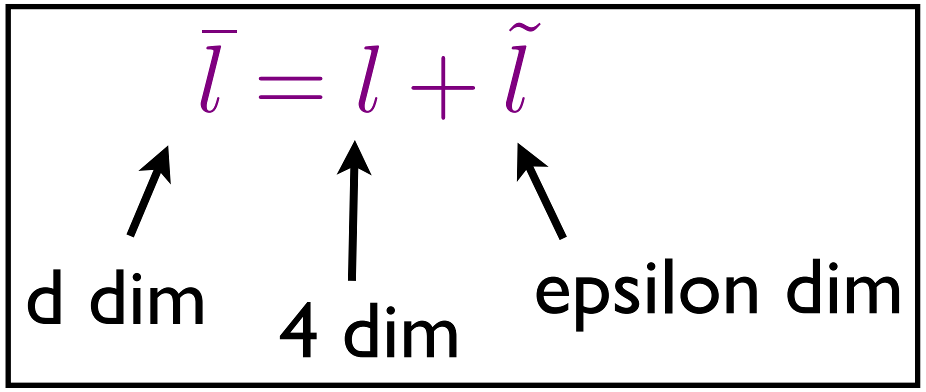
Integrand reduction

- Any one-loop integral can be decomposed in scalar integrals
- The task is to find these coefficients efficiently (analytically or numerically)
- The integrand (or OPP [Ossola, Papadopoulos, Pittau 2006]) reduction method is a method that has been automated in the **CutTools** program to find these coefficients in an automated way
- The integrand reduction technique is what we have adopted to use in MadGraph to compute the loop diagrams

Complications in d dimensions

- In the previous consideration I was very sloppy in considering if we are working in 4 or d dimensions
- In general, external momenta and polarization vectors are in 4 dimensions; only the loop momentum is in d dimensions
- To be more correct, we compute the integral

$$\int d^d l \, \frac{N(l,\tilde{l})}{\bar{D}_0 \bar{D}_1 \bar{D}_2 \cdots \bar{D}_{m-1}}$$



$$\bar{l} = l + \tilde{l}$$

d dim

4 dim

epsilon dim

$$\bar{D}_i = (\bar{l} + p_i)^2 - m_i^2 = (l + p_i)^2 - m_i^2 + \tilde{l}^2 = D_i + \tilde{l}^2$$

$$l \cdot \tilde{l} = 0 \qquad \bar{l} \cdot p_i = l \cdot p_i \qquad \bar{l} \cdot \bar{l} = l \cdot l + \tilde{l} \cdot \tilde{l}$$

Implications

$$
\begin{aligned}
&\sum_{0\le i_0<i_1<i_2<i_3}^{m-1} d(i_0 i_1 i_2 i_3) \int d^d\bar{\ell}\, \frac{1}{\bar{D}_{i_0}\bar{D}_{i_1}\bar{D}_{i_2}\bar{D}_{i_3}} \\
&+ \sum_{0\le i_0<i_1<i_2}^{m-1} c(i_0 i_1 i_2) \int d^d\bar{\ell}\, \frac{1}{\bar{D}_{i_0}\bar{D}_{i_1}\bar{D}_{i_2}} \\
&+ \sum_{0\le i_0<i_1}^{m-1} b(i_0 i_1) \int d^d\bar{\ell}\, \frac{1}{\bar{D}_{i_0}\bar{D}_{i_1}} \\
&+ \sum_{i_0=0}^{m-1} a(i_0) \int d^d\bar{\ell}\, \frac{1}{\bar{D}_{i_0}} \\
&+ R\,.
\end{aligned}
$$

- The decomposition in terms of scalar integrals has to be done in d dimensions
- This is why the rational part *R* is needed

Rational terms

$$R = R_1 + R_2$$

Rational terms

- In the OPP method, they are split into two contributions, generally called

$$R = R_1 + R_2$$

- Both have their origin in the UV part of the model, but only R_1 can be directly computed in the OPP reduction and is given by the CutTools program

Rational terms

- In the OPP method, they are split into two contributions, generally called

$$R = \bigcirc\!\!\!\!\!R_1 + R_2$$

- Both have their origin in the UV part of the model, but only R_1 can be directly computed in the OPP reduction and is given by the CutTools program
  - ➡ R1: originates from the propagator (calculate by CutTools)

Rational terms

- In the OPP method, they are split into two contributions, generally called

$$R = R_1 + \boxed{R_2}$$

- Both have their origin in the UV part of the model, but only R_1 can be directly computed in the OPP reduction and is given by the CutTools program
  - ➡ R1: originates from the propagator (calculate by CutTools)
  - ➡ R2: originates from the numerator (need in the model)

Rational terms

- In the OPP method, they are split into two contributions, generally called

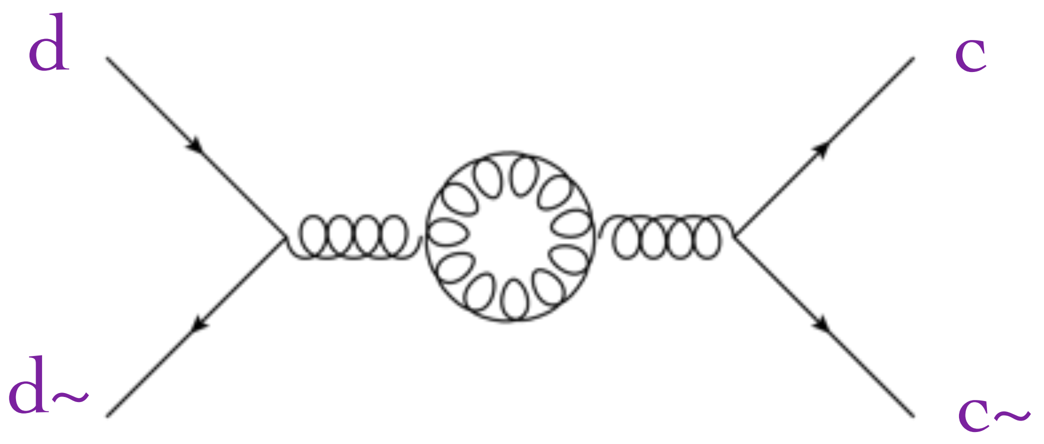
$$R = R_1 + \boxed{R_2}$$

- Both have their origin in the UV part of the model, but only R_1 can be directly computed in the OPP reduction and is given by the CutTools program
    - ➡ R1: originates from the propagator (calculate by CutTools)
    - ➡ R2: originates from the numerator (need in the model)

The MadGraph Solution: MadLoop

- Need to upgrade MadGraph so to generate loop diagrams and numerical code for the integrand N(q)

Remember:



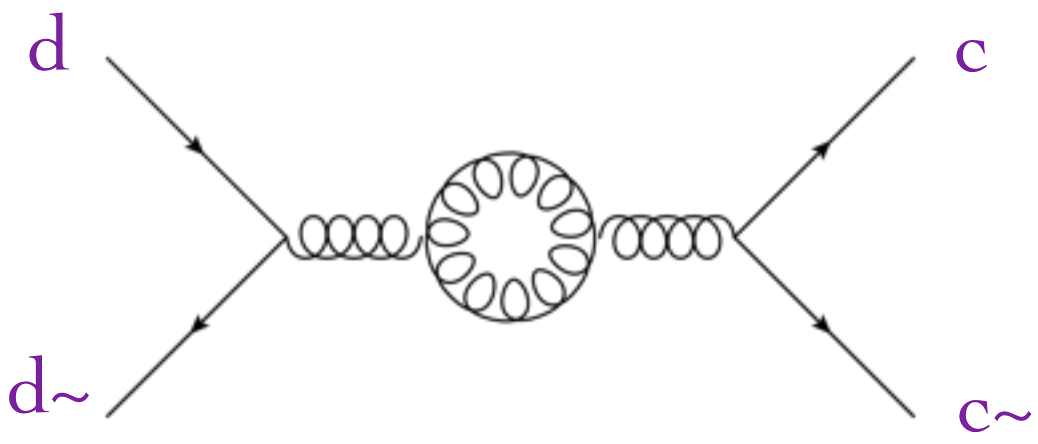
$$\mathcal{N}(l^\mu)$$

The MadGraph Solution: MadLoop

- Need to upgrade MadGraph so to generate loop diagrams and numerical code for the integrand N(q)

Remember:

➡ For a given phase-space point, we have to compute the numerator function several times (~50 or so for a box loop) per external phase-space point.



$$\mathcal{N}(l^\mu)$$

The MadGraph Solution: MadLoop

- Need to upgrade MadGraph so to generate loop diagrams and numerical code for the integrand N(q)

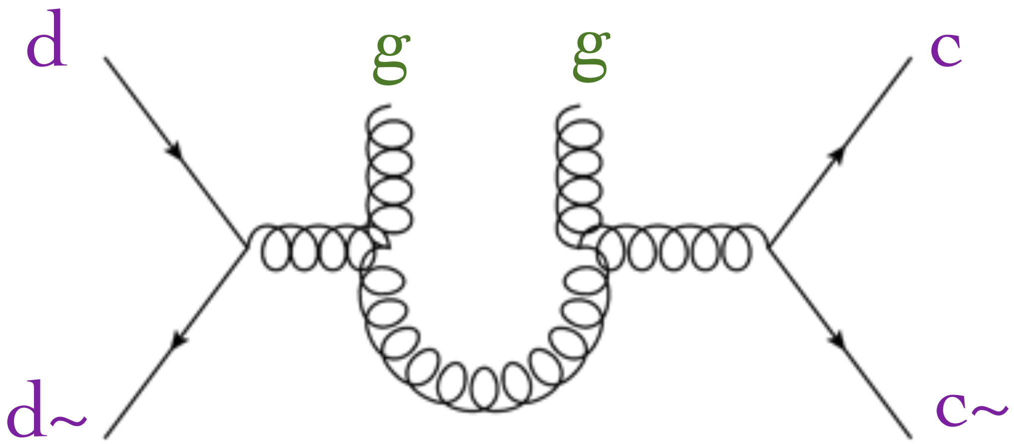
Remember:

➡ For a given phase-space point, we have to compute the numerator function several times (~50 or so for a box loop) per external phase-space point.

➡ Cut the Loop and use HELAS (with no denominator)

➡ Generic

➡ recycling



$\mathcal{N}(l^{\mu})$

The MadGraph Solution: MadLoop

- Need to upgrade MadGraph so to generate loop diagrams and numerical code for the integrand N(q)

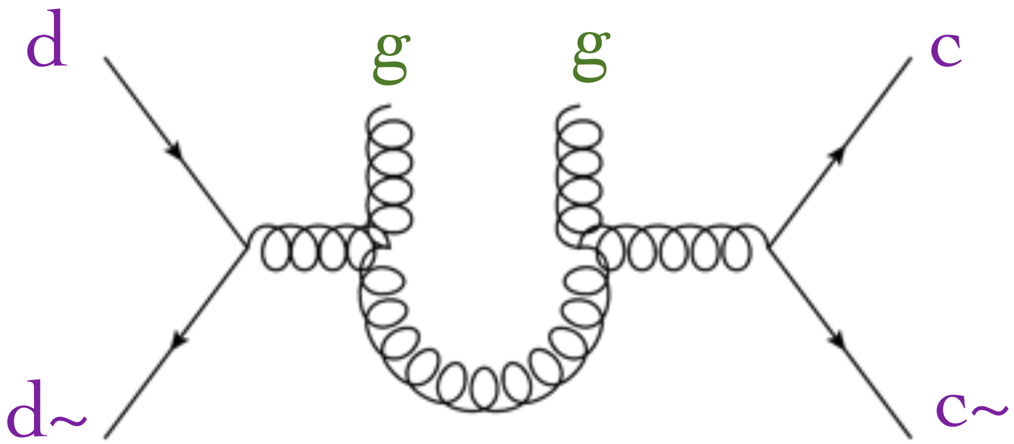
Remember:

➡ For a given phase-space point, we have to compute the numerator function several times (~50 or so for a box loop) per external phase-space point.

➡ Cut the Loop and use HELAS (with no denominator)

➡ Generic

➡ recycling



OpenLoop: [S. Pozzorini & al.(2011)]

$$\mathcal{N}(l^\mu) = \sum_{r=0}^{r_{max}} C^{(r)}_{\mu_0\mu_1\cdots\mu_r} l^{\mu_0} l^{\mu_1} \dots l^{\mu_r}$$

coefficient computed iteratively by ALOHA

The MadGraph Solution: MadLoop

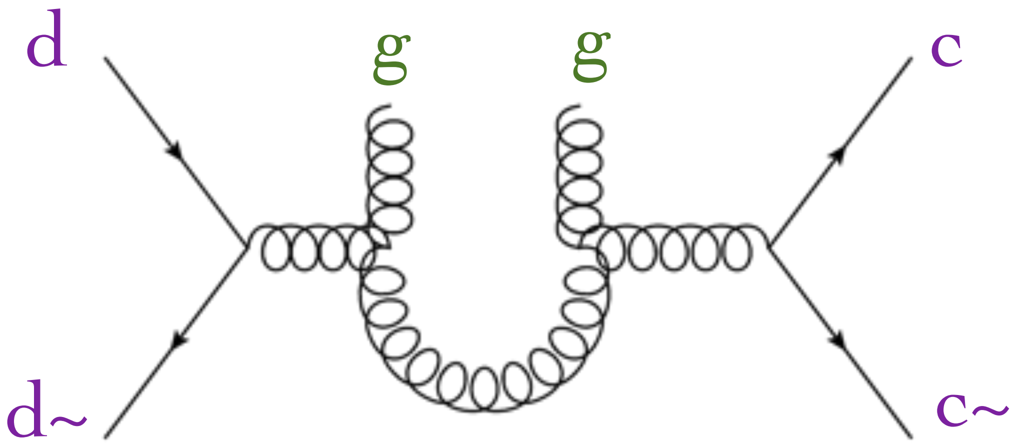
- Need to upgrade MadGraph so to generate loop diagrams and numerical code for the integrand N(q)

Remember:

➡ For a given phase-space point, we have to compute the numerator function several times (~50 or so for a box loop) per external phase-space point.

➡ Cut the Loop and use HELAS (with no denominator)

- ➡ Generic
- ➡ recycling



OpenLoop: [S. Pozzorini & al.(2011)]

$$\mathcal{N}(l^\mu) = \sum_{r=0}^{r_{max}} C^{(r)}_{\mu_0\mu_1\cdots\mu_r} l^{\mu_0} l^{\mu_1} \ldots l^{\mu_r}$$

5-10 times faster

coefficient computed iteratively by ALOHA

FKS

phase-space integration

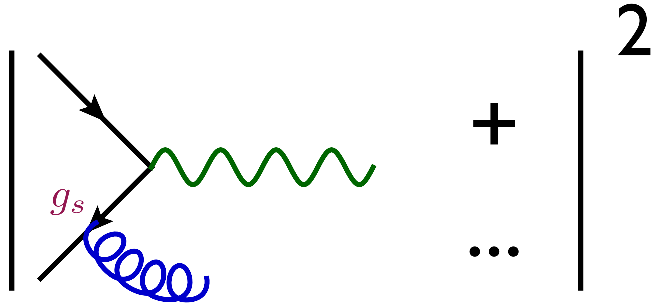
$$\sigma^{\mathrm{NLO}} \sim \int d^4\Phi_m\, B(\Phi_m) + \int d^4\Phi_m \int_{\mathrm{loop}} d^d l\, V(\Phi_m) + \int d^d\Phi_{m+1}\, R(\Phi_{m+1})$$

- For complicated processes we have to result to numerical phase-space integration techniques (“Monte Carlo integration”), which can only be performed in an integer number of dimensions
  - ➡ Cannot use a finite value for the dimensional regulator and take the limit to zero in a numerical code
- But we still have to cancel the divergences explicitly
- Use a subtraction method to explicitly factor out the divergences from the phase-space integrals

Example

- Suppose we want to compute the integral ("real emission radiation", where the 1-particle phase-space is referred to as the 1-dimensional *x*)

$$\int_0^1 dx\, f(x)$$



where $f(x) = \frac{g(x)}{x}$ and $g(x)$ is finite everywhere

- Let's introduce a regulator

$$\lim_{\epsilon \to 0} \int_0^1 dx\, \frac{g(x)}{x^{1+\epsilon}} = \lim_{\epsilon \to 0} \int_0^1 dx\, x^{-\epsilon} f(x)$$

for any non-integer non-zero value for ϵ this integral is finite

- We would like to factor out the explicit poles in ϵ so that they can be canceled explicitly against the virtual corrections

Subtraction method

$$\lim_{\epsilon\to 0}\int_0^1 dx\, x^{-\epsilon} f(x) \qquad\qquad f(x) = \frac{g(x)}{x}$$

- Add and subtract the same term

$$\lim_{\epsilon\to 0}\int_0^1 dx\, x^{-\epsilon} f(x) = \lim_{\epsilon\to 0}\int_0^1 dx\, x^{-\epsilon}\left[\frac{g(0)}{x} + f(x) - \frac{g(0)}{x}\right]$$

$$= \lim_{\epsilon\to 0}\int_0^1 dx\left[g(0)\frac{x^{-\epsilon}}{x} + \frac{g(x)-g(0)}{x^{1+\epsilon}}\right]$$

$$= \lim_{\epsilon\to 0}\frac{-1}{\epsilon}g(0) + \int_0^1 dx\,\frac{g(x)-g(0)}{x}$$

- We have factored out the 1/ϵ divergence and are left with a finite integral
- According to the KLN theorem the divergence cancels against the virtual corrections

Limitations

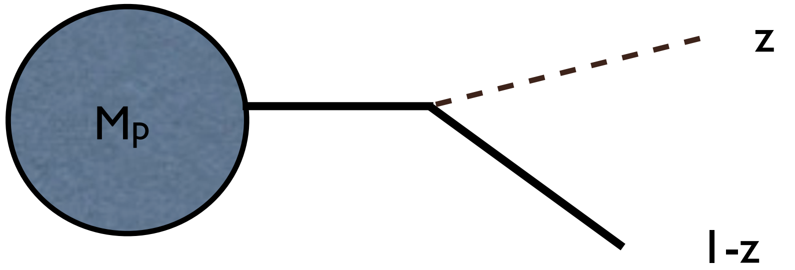
“Plus distribution”

Subtraction: $\int_0^1 dx \, \frac{g(x) - g(0)}{x}$

- Even though the divergence is factored, there are cancellations between large numbers: if for an observable O, if $\lim_{x \to 0} O(x) \neq O(0)$ or we choose the bin-size too small, instabilities render the computation useless
  - ➡ We already knew that! KLN is sufficient; one must have infra-red safe observables and cannot ask for infinite resolution (need a finite bin-size)
- Subtraction method is very flexible -> method of choice in automation

FKS

$$\frac{1}{(p_q+p_g)^2} \simeq \frac{1}{2E_qE_g(1-\cos\theta)}$$



- Split the Phase space into pieces with at most one collinear and one soft divergencies

$$M^{n+1} = \sum_{i,j} S_{ij} M^{n+1} \equiv \sum_{i,j} M_{ij} \qquad \sum_{i,j} S_{ij} = 1$$

- Identify divergent part:

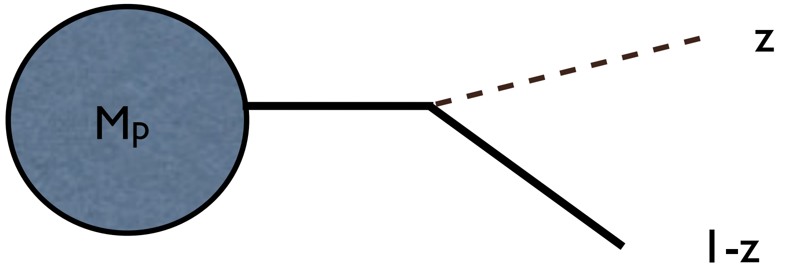
$$M_{ij} = \frac{E_i(1-cos\theta_{ij})}{E_i(1-cos\theta_{ij})} M_{ij}$$

- Remove divergencies:

$$\tilde{M}_{ij} = \left(\frac{1}{E_i}\right)_+ \left(\frac{1}{1-\cos\theta_{ij}}\right)_+ (E_i(1-cos\theta_{ij}))M_{ij}$$

FKS

$$\frac{1}{(p_q+p_g)^2} \simeq \frac{1}{2E_qE_g(1-\cos\theta)}$$



- Split the Phase space into pieces with at most one collinear and one soft divergencies

$$M^{n+1} = \sum_{i,j} S_{ij} M^{n+1} \equiv \sum_{i,j} M_{ij} \qquad\qquad \sum_{i,j} S_{ij} = 1$$

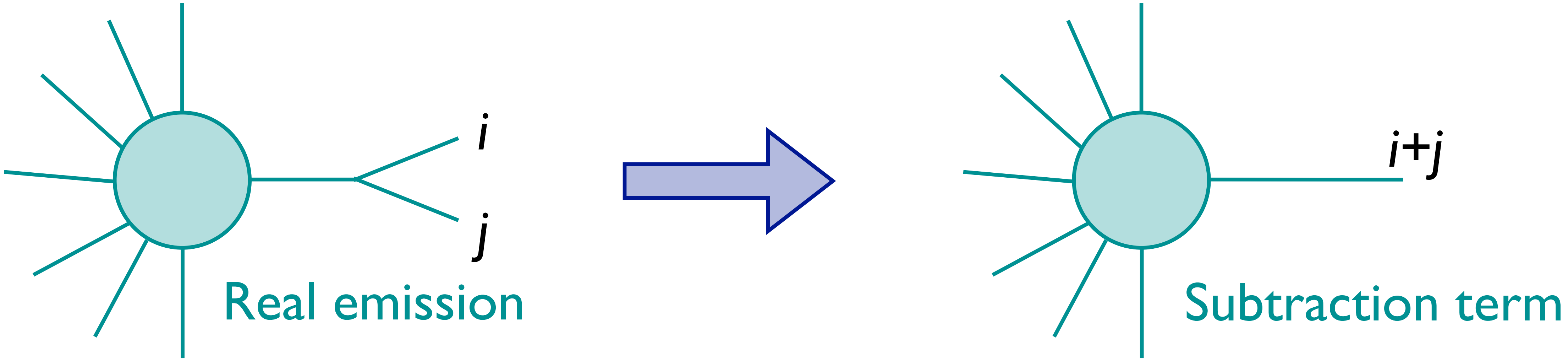
- Identify divergent part:

$$M_{ij} = \frac{E_i(1-cos\theta_{ij})}{E_i(1-cos\theta_{ij})} M_{ij} \quad g(x)$$

- Remove divergencies:

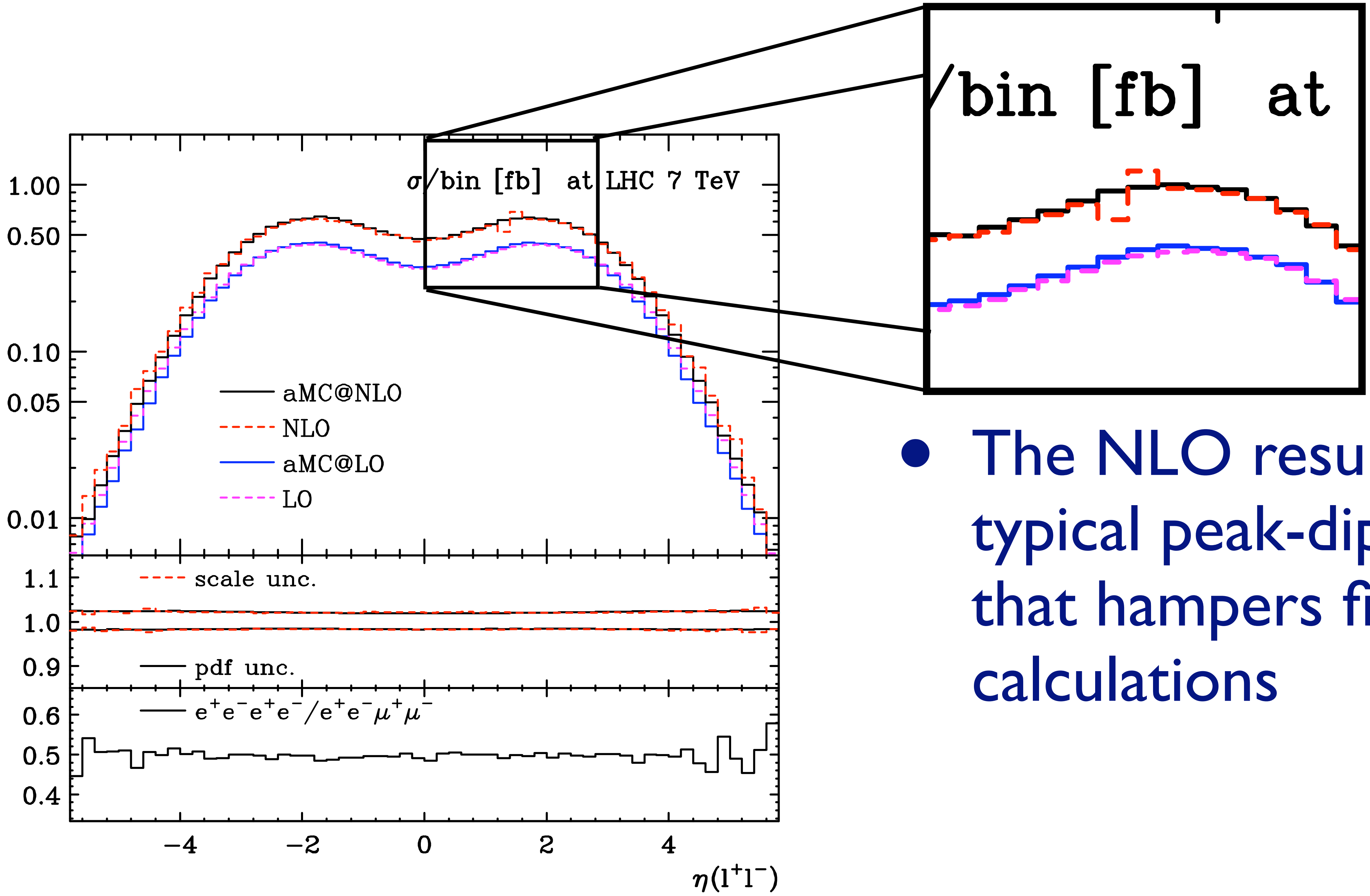
$$\tilde{M}_{ij} = \left(\frac{1}{E_i}\right)_+ \left(\frac{1}{1-\cos\theta_{ij}}\right)_+ (E_i(1-cos\theta_{ij}))M_{ij}$$

Kinematics of counter events



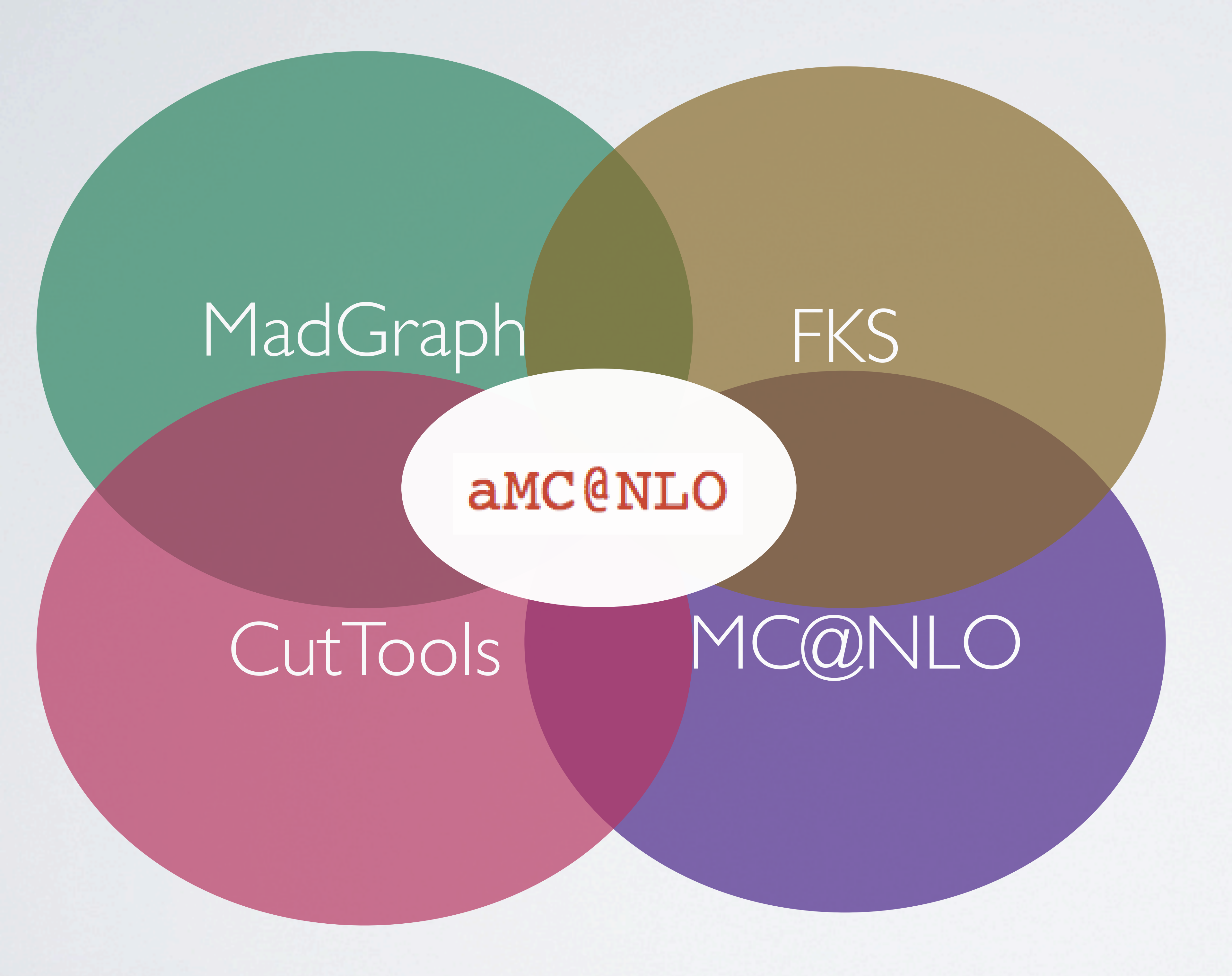
- If *i* and *j* are two on-shell particles that are present in a splitting that leads to an singularity, for the counter events we need to combine their momenta to a new on-shell parton that's the sum of *i+j*
- This is not possible without changing any of the other momenta in the process
- When applying cuts or making plots, events and counter events might end-up in different bins

Example in 4 charged lepton production



- The NLO results shows a typical peak-dip structure that hampers fixed order calculations

aMC@NLO


MadGraph
FKS
aMC@NLO
CutTools
MC@NLO

aMC@NLO

- Why automation?
    - ➡ Time: Less tools, means more time for physics
    - ➡ Robust: Easier to test, to trust
    - ➡ Easy: One framework/tool to learn

aMC@NLO

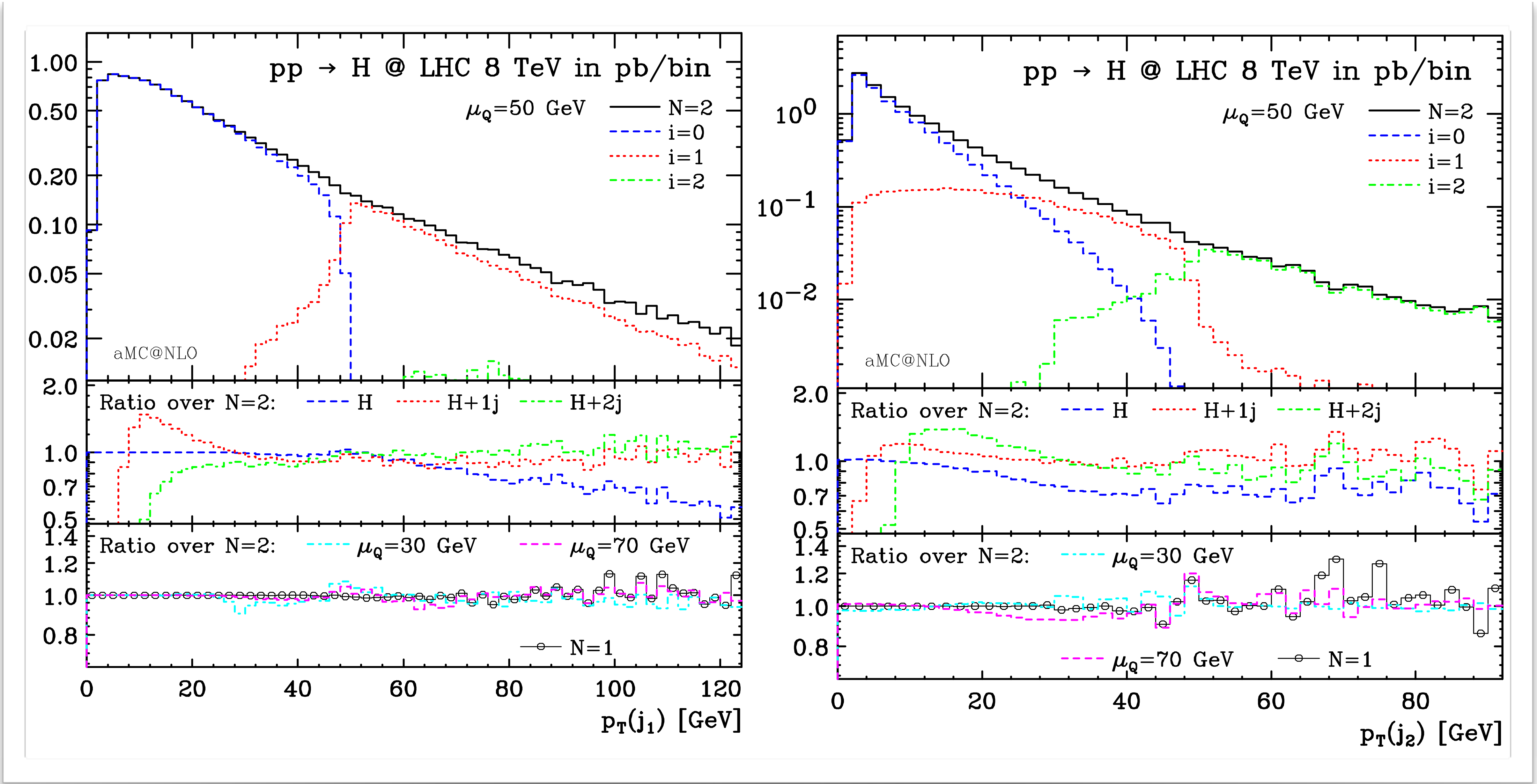
- Why automation?
  - ➡ Time: Less tools, means more time for physics
  - ➡ Robust: Easier to test, to trust
  - ➡ Easy: One framework/tool to learn
- Why matched to the PS?
  - ➡ Parton are not an detector observables
  - ➡ Matching cure some fix-order ill behaved observables

aMC@NLO

- Why automation?
  - ➡ Time: Less tools, means more time for physics
  - ➡ Robust: Easier to test, to trust
  - ➡ Easy: One framework/tool to learn
- Why matched to the PS?
  - ➡ Parton are not an detector observables
  - ➡ Matching cure some fix-order ill behaved observables
- Why NOT merging?
  - ➡ works in progress

RF & Frixione, 2012

ME+PS merging at NLO

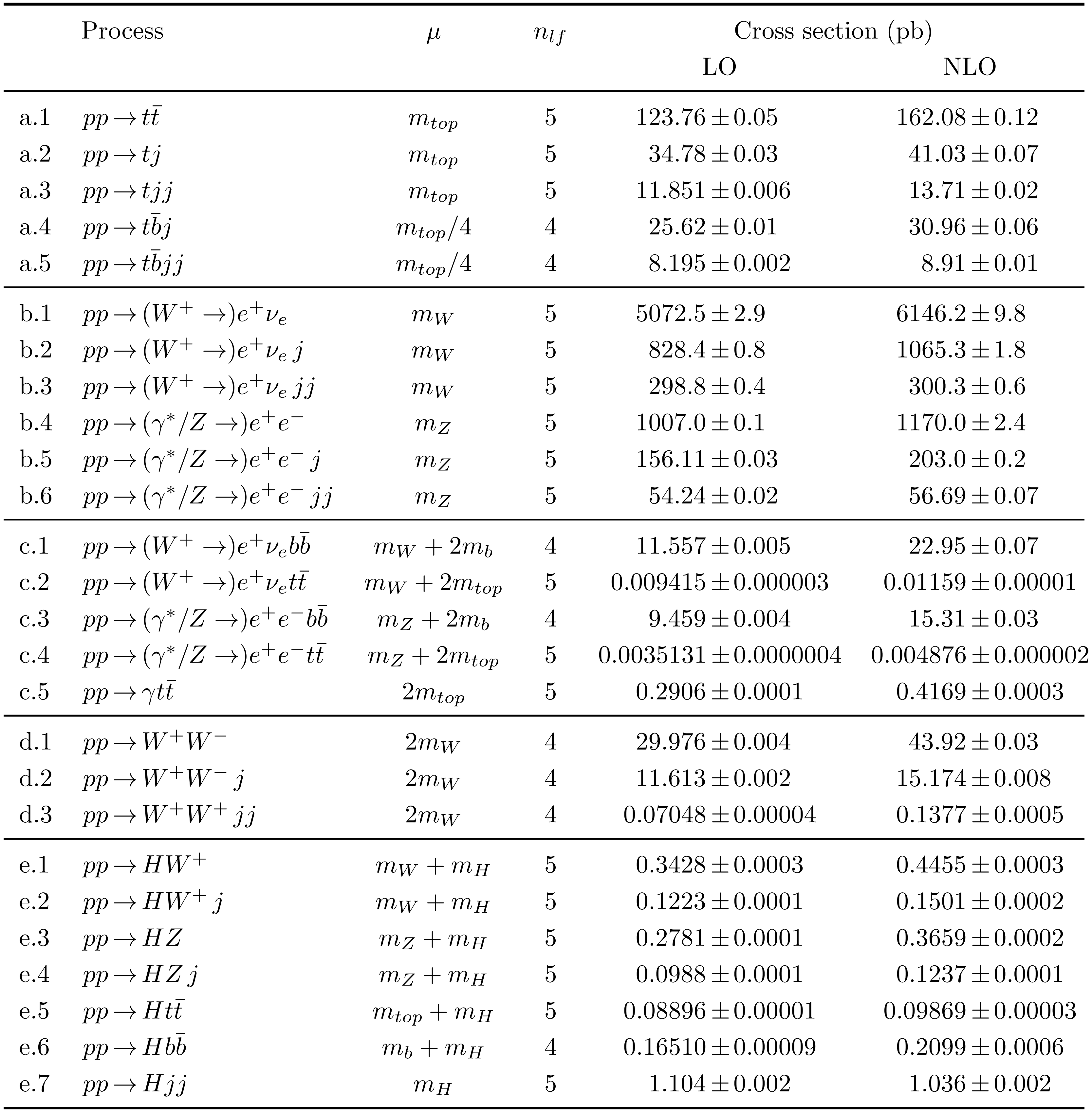


- Hardest and 2nd hardest jets in Higgs production by gluon fusion
- Merged sample agrees with NLO in the regions of phase-space where it should; smooth in between; and nearly no dependence on the matching scale
- Not yet automated... work in progress

- Generation
  - ➡ add [QCD]

generate p p > w+ j [QCD]

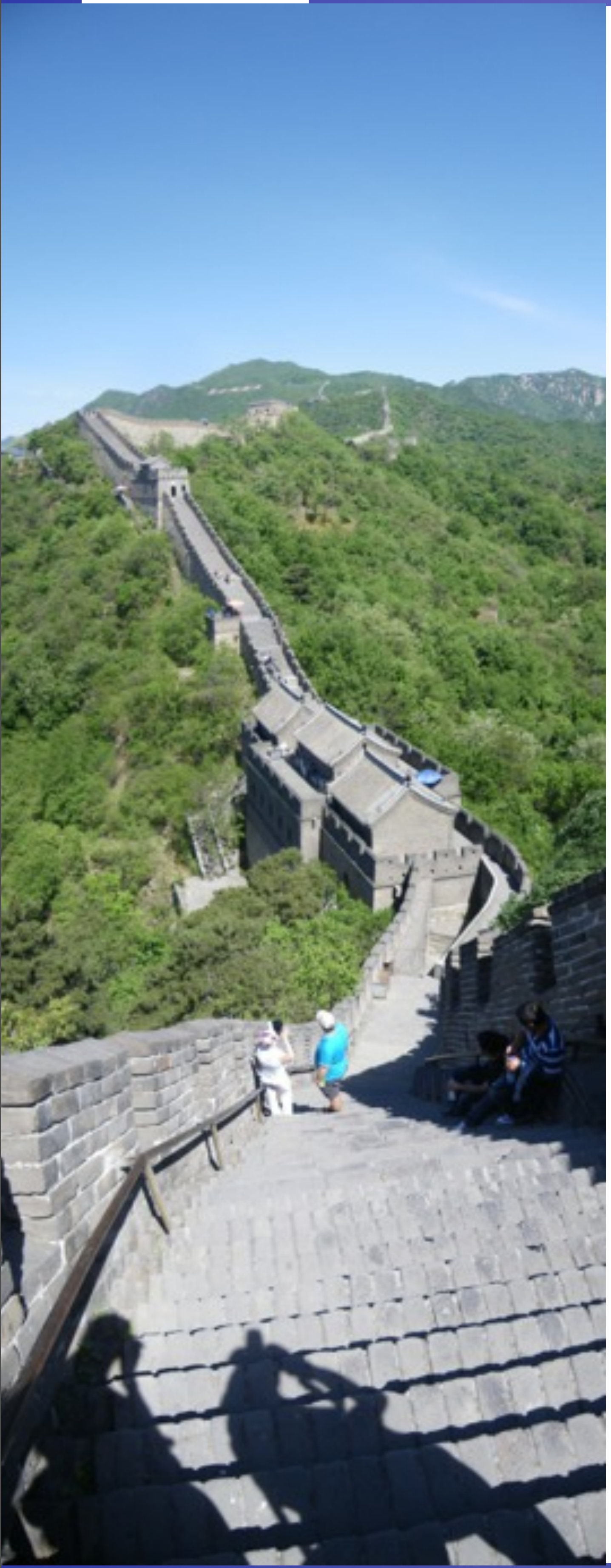
- list run in 2 weeks in a 150 node cluster



	Process	μ	n_{lf}	Cross section (pb)	
				LO	NLO
a.1	$pp \to t\bar{t}$	m_{top}	5	123.76 ± 0.05	162.08 ± 0.12
a.2	$pp \to tj$	m_{top}	5	34.78 ± 0.03	41.03 ± 0.07
a.3	$pp \to tjj$	m_{top}	5	11.851 ± 0.006	13.71 ± 0.02
a.4	$pp \to t\bar{b}j$	$m_{top}/4$	4	25.62 ± 0.01	30.96 ± 0.06
a.5	$pp \to t\bar{b}jj$	$m_{top}/4$	4	8.195 ± 0.002	8.91 ± 0.01
b.1	$pp \to (W^+ \to) e^+ \nu_e$	m_W	5	5072.5 ± 2.9	6146.2 ± 9.8
b.2	$pp \to (W^+ \to) e^+ \nu_e\, j$	m_W	5	828.4 ± 0.8	1065.3 ± 1.8
b.3	$pp \to (W^+ \to) e^+ \nu_e\, jj$	m_W	5	298.8 ± 0.4	300.3 ± 0.6
b.4	$pp \to (\gamma^*/Z \to) e^+ e^-$	m_Z	5	1007.0 ± 0.1	1170.0 ± 2.4
b.5	$pp \to (\gamma^*/Z \to) e^+ e^-\, j$	m_Z	5	156.11 ± 0.03	203.0 ± 0.2
b.6	$pp \to (\gamma^*/Z \to) e^+ e^-\, jj$	m_Z	5	54.24 ± 0.02	56.69 ± 0.07
c.1	$pp \to (W^+ \to) e^+ \nu_e b\bar{b}$	$m_W + 2m_b$	4	11.557 ± 0.005	22.95 ± 0.07
c.2	$pp \to (W^+ \to) e^+ \nu_e t\bar{t}$	$m_W + 2m_{top}$	5	0.009415 ± 0.000003	0.01159 ± 0.00001
c.3	$pp \to (\gamma^*/Z \to) e^+ e^- b\bar{b}$	$m_Z + 2m_b$	4	9.459 ± 0.004	15.31 ± 0.03
c.4	$pp \to (\gamma^*/Z \to) e^+ e^- t\bar{t}$	$m_Z + 2m_{top}$	5	0.0035131 ± 0.0000004	0.004876 ± 0.000002
c.5	$pp \to \gamma t\bar{t}$	$2m_{top}$	5	0.2906 ± 0.0001	0.4169 ± 0.0003
d.1	$pp \to W^+ W^-$	$2m_W$	4	29.976 ± 0.004	43.92 ± 0.03
d.2	$pp \to W^+ W^-\, j$	$2m_W$	4	11.613 ± 0.002	15.174 ± 0.008
d.3	$pp \to W^+ W^+\, jj$	$2m_W$	4	0.07048 ± 0.00004	0.1377 ± 0.0005
e.1	$pp \to HW^+$	$m_W + m_H$	5	0.3428 ± 0.0003	0.4455 ± 0.0003
e.2	$pp \to HW^+\, j$	$m_W + m_H$	5	0.1223 ± 0.0001	0.1501 ± 0.0002
e.3	$pp \to HZ$	$m_Z + m_H$	5	0.2781 ± 0.0001	0.3659 ± 0.0002
e.4	$pp \to HZ\, j$	$m_Z + m_H$	5	0.0988 ± 0.0001	0.1237 ± 0.0001
e.5	$pp \to Ht\bar{t}$	$m_{top} + m_H$	5	0.08896 ± 0.00001	0.09869 ± 0.00003
e.6	$pp \to Hb\bar{b}$	$m_b + m_H$	4	0.16510 ± 0.00009	0.2099 ± 0.0006
e.7	$pp \to Hjj$	m_H	5	1.104 ± 0.002	1.036 ± 0.002

Conclusion

- We are now in the Automated loop computations area
- We expect improvement in ALL directions
  - ➡ Speed
  - ➡ Merging
  - ➡ Tools
- aMC@NLO
  - ➡ MadLoop: OPP+OpenLoops
  - ➡ Real: FKS subtraction
  - ➡ Matched to the shower (MC@NLO)



Conclusion

- We are now in the Automated loop computations area
- We expect improvement in ALL directions
    - ➡ Speed
    - ➡ Merging
    - ➡ Tools
- aMC@NLO
    - ➡ MadLoop: OPP+OpenLoops
    - ➡ Real: FKS subtraction
    - ➡ Matched to the shower (MC@NLO)