aMC@NLO

Olivier Mattelaer
IPPP/Durham

Plan

- NLO Introduction
- Loop Computation
- Dealing with Singularities
- Matching@NLO
- Live Demo

Plan

- NLO Introduction
- Loop Computation
- Dealing with Singularities
- Matching@NLO
- Live Demo

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:

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

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

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

NLO corrections

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

NLO corrections

NNLO corrections

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

NNLO corrections

N3LO or NNNLO corrections

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

NLO corrections

NNLO corrections

N3LO or NNNLO corrections

- Including higher corrections improves predictions and reduces theoretical uncertainties

Improved predictions

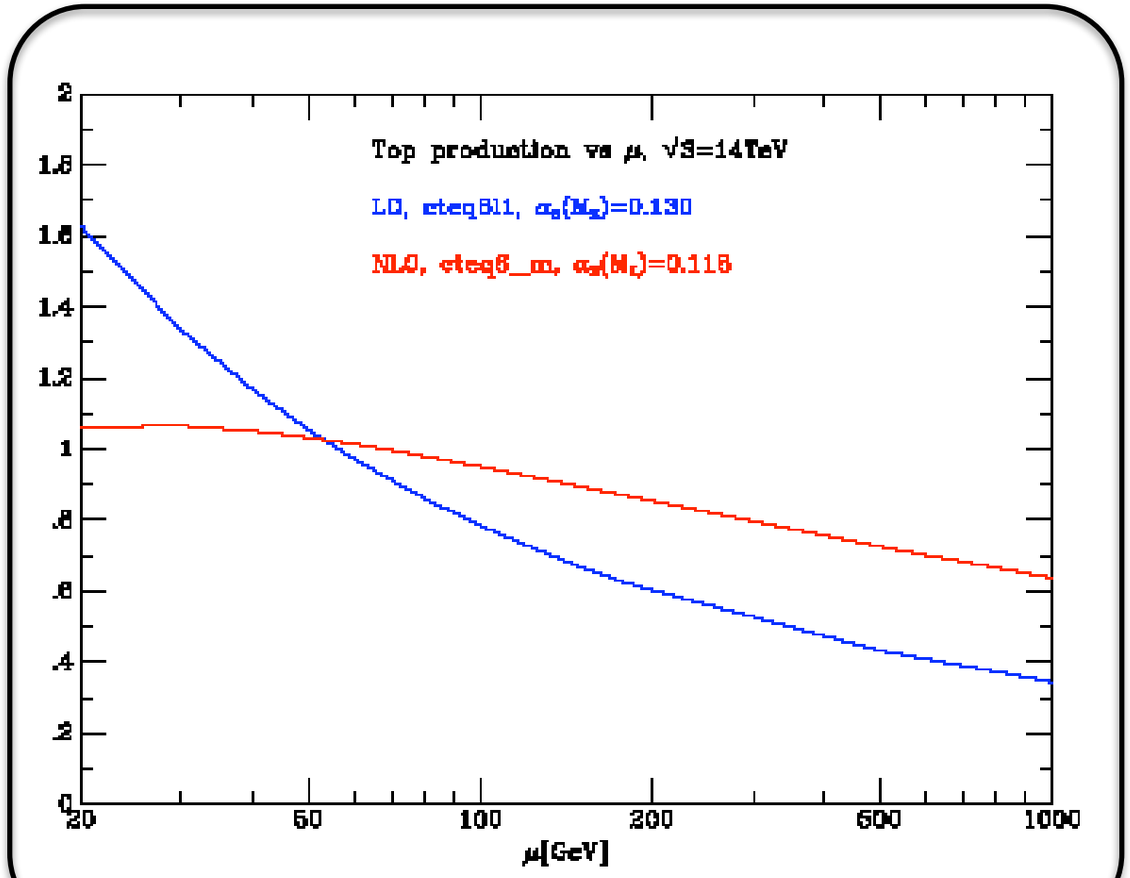
$$d\sigma = \sum_{a,b} \int dx_1 dx_2 \; f_a(x_1, \mu_F) f_b(x_2, \mu_F) \, d\hat{\sigma}_{ab \to X}(\hat{s}, \mu_F, \mu_R)$$

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

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

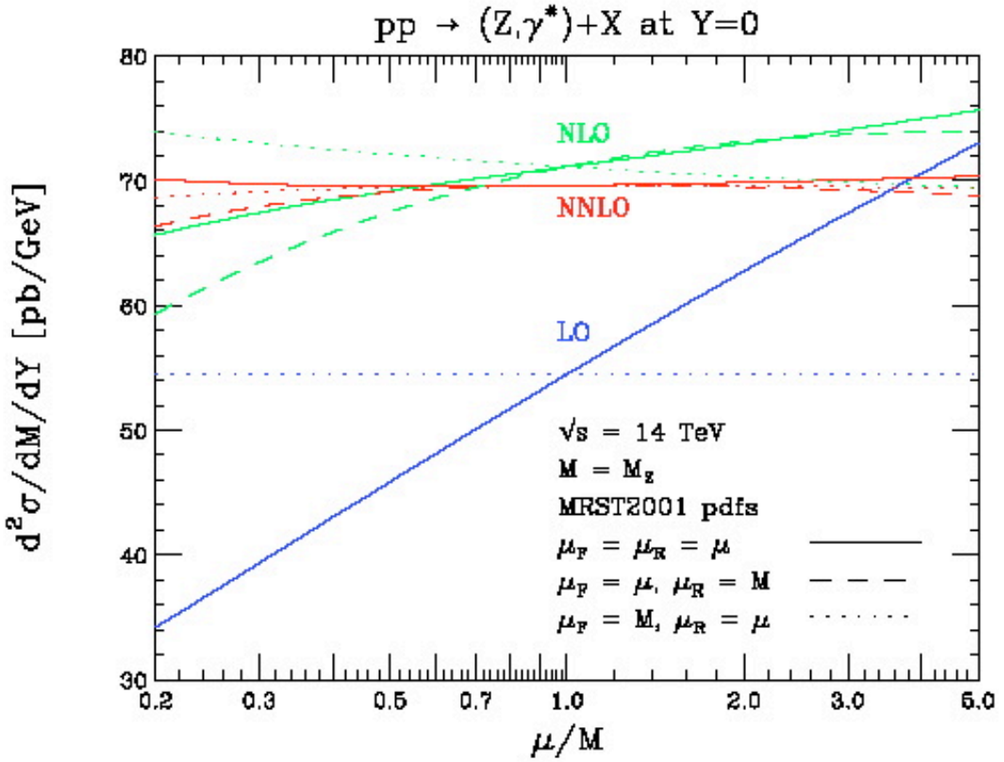
Going NLO

- 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

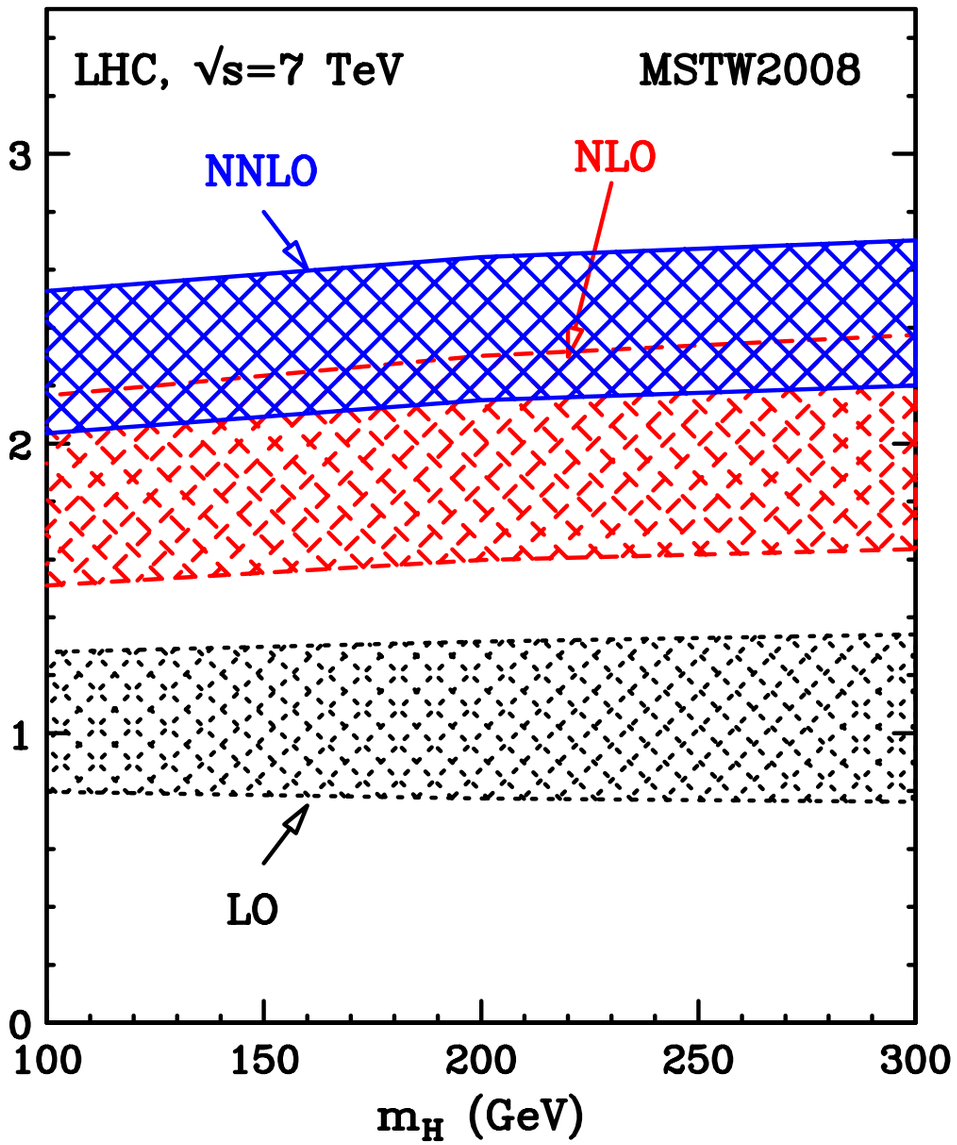


Going NNLO…?

- NNLO is the current state-of-the-art. There are only a few results available: Higgs, Drell-Yan, ttbar
- 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

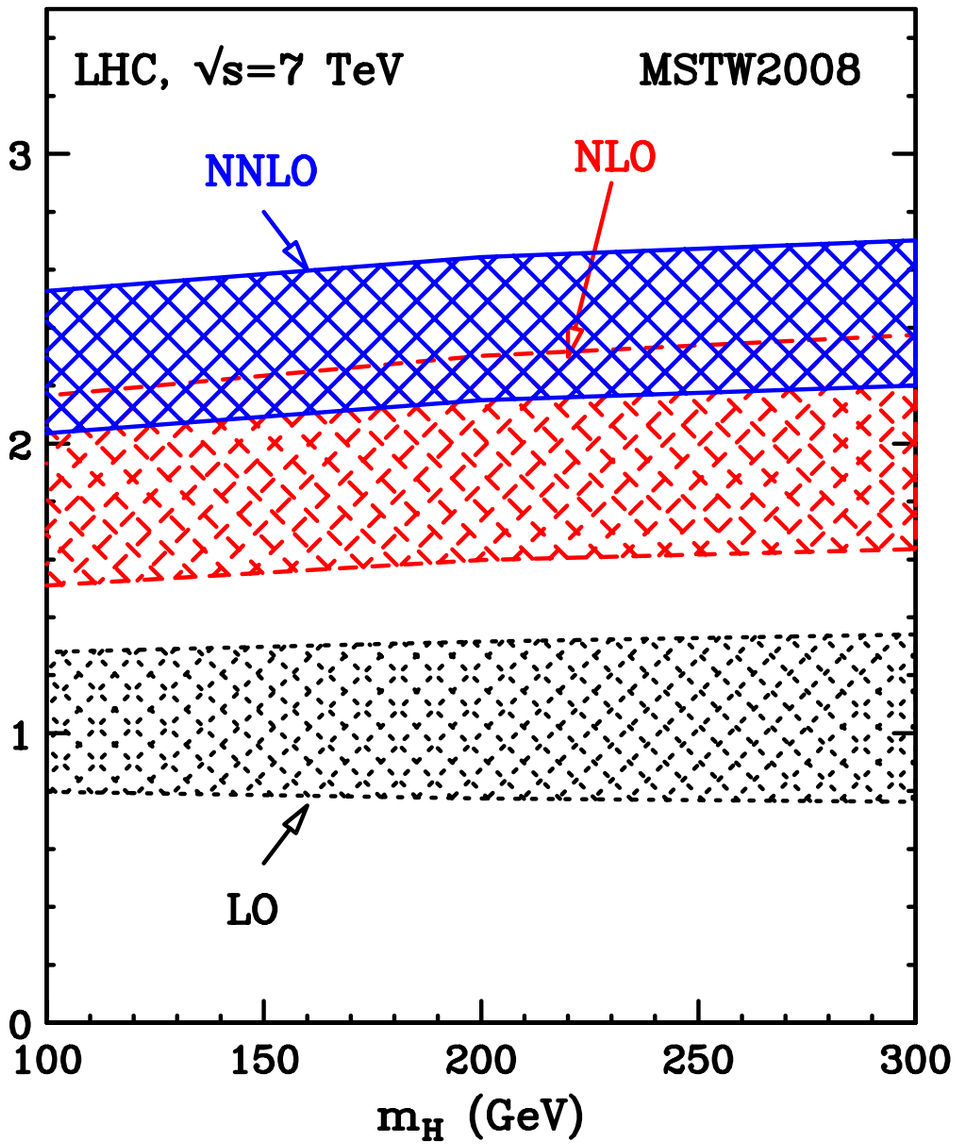


Higgs at NNLO



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

Higgs 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

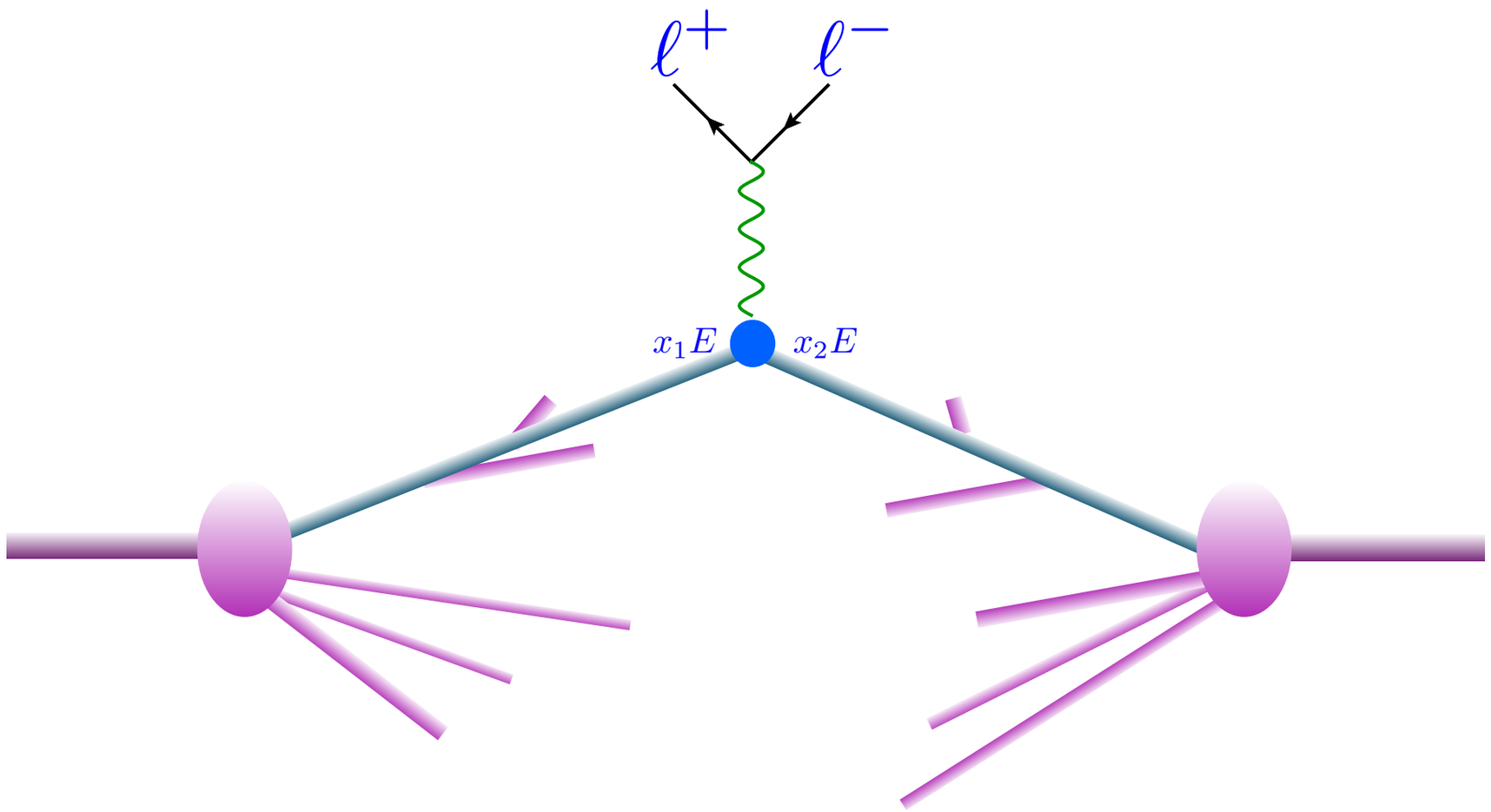
NLO corrections

- **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

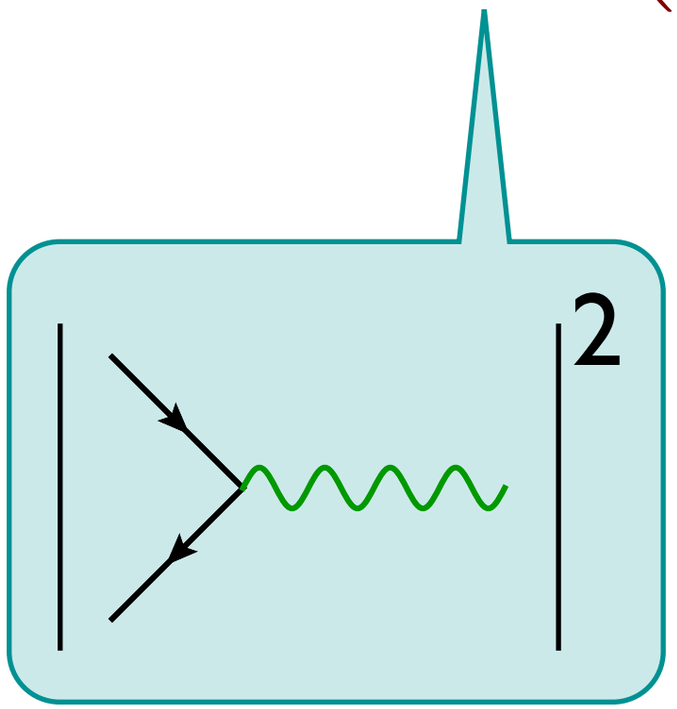
- As an example, consider Drell-Yan Z/γ* production

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

NLO predictions

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

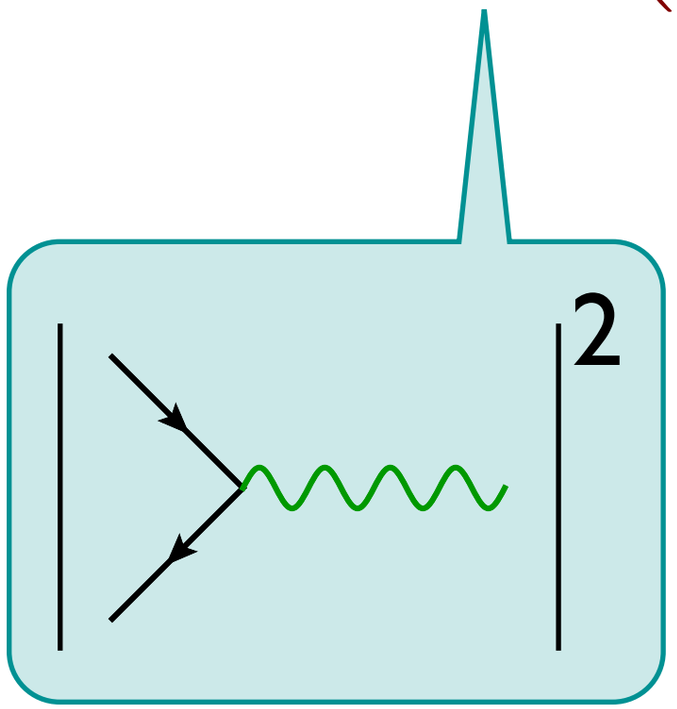
$$\hat{\sigma} = \sigma^{\text{Born}} \left(1 + \frac{\alpha_s}{2\pi} \sigma^{(1)} + \dots \right)$$

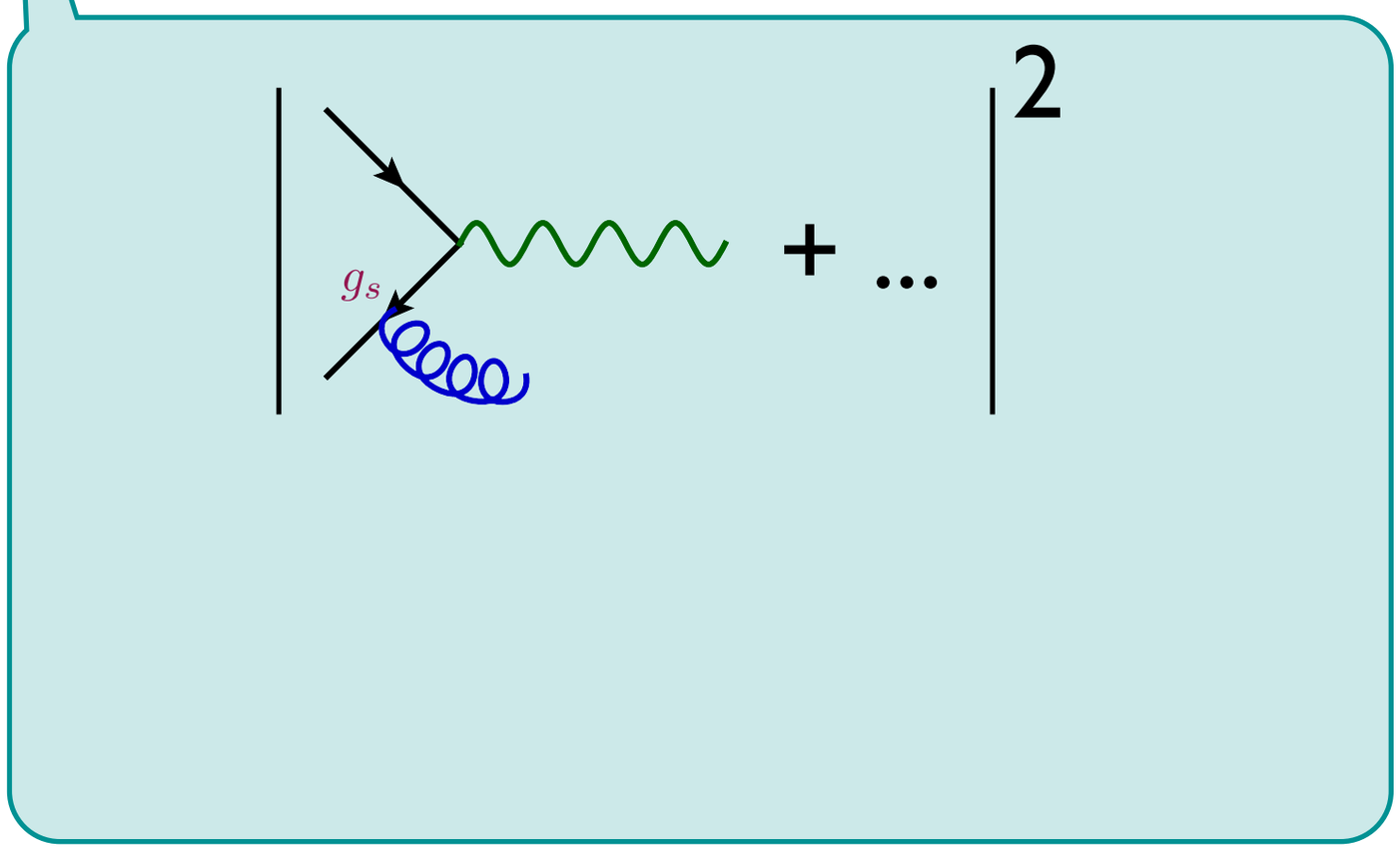


NLO predictions

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

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

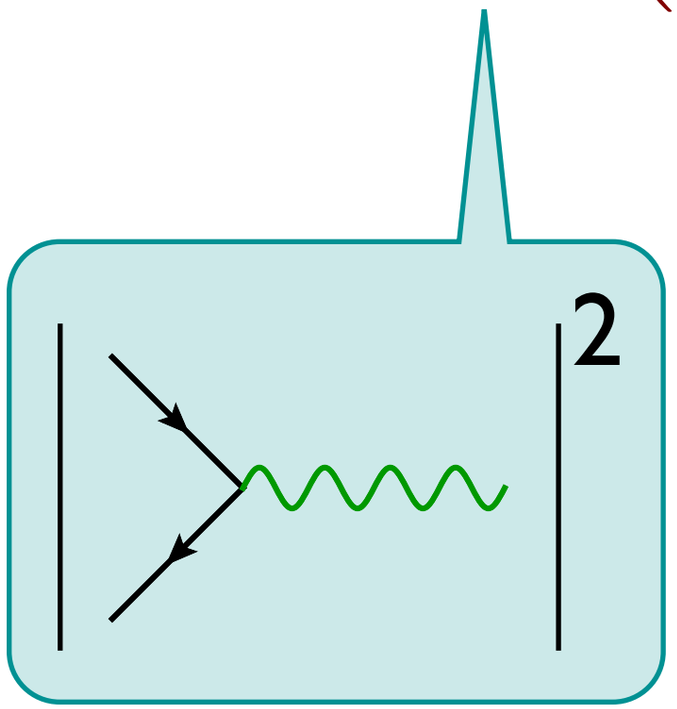


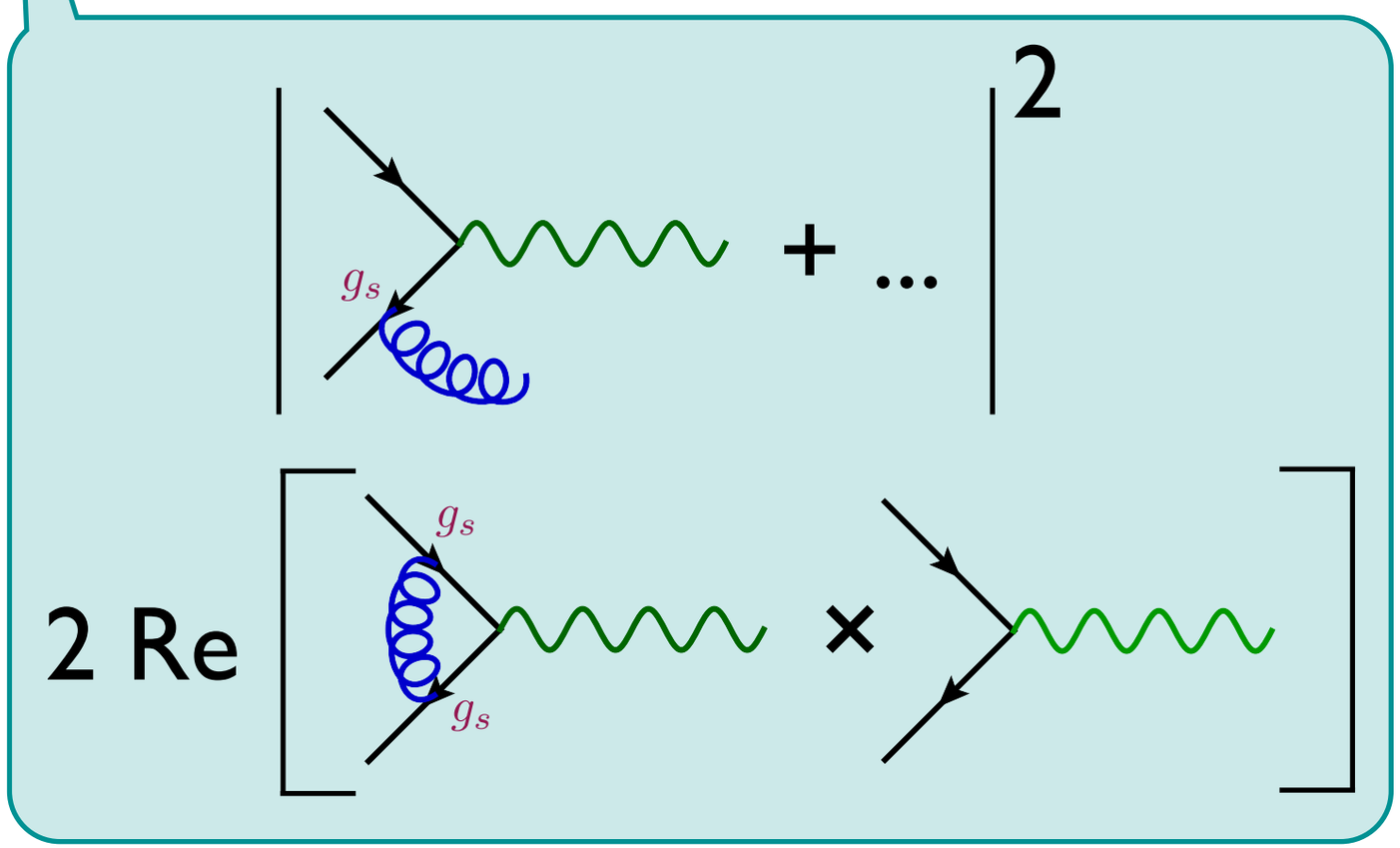


NLO predictions

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

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

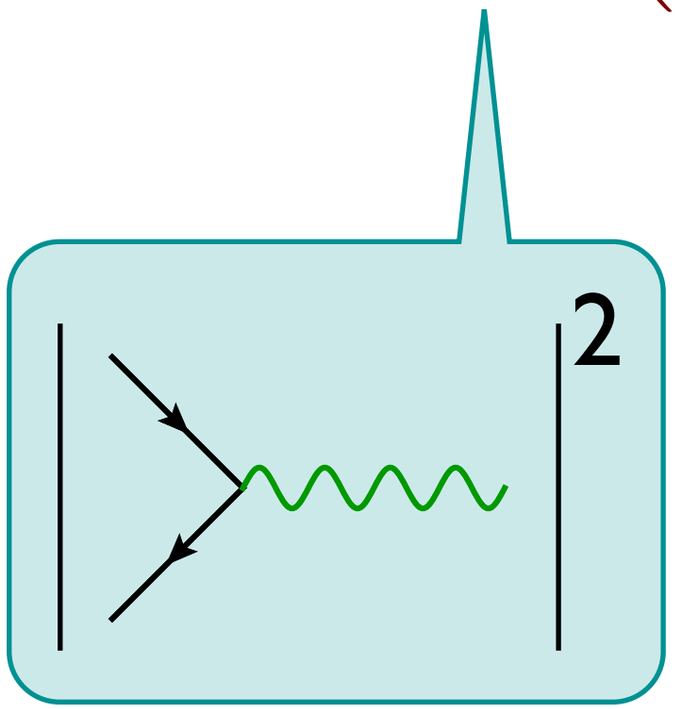


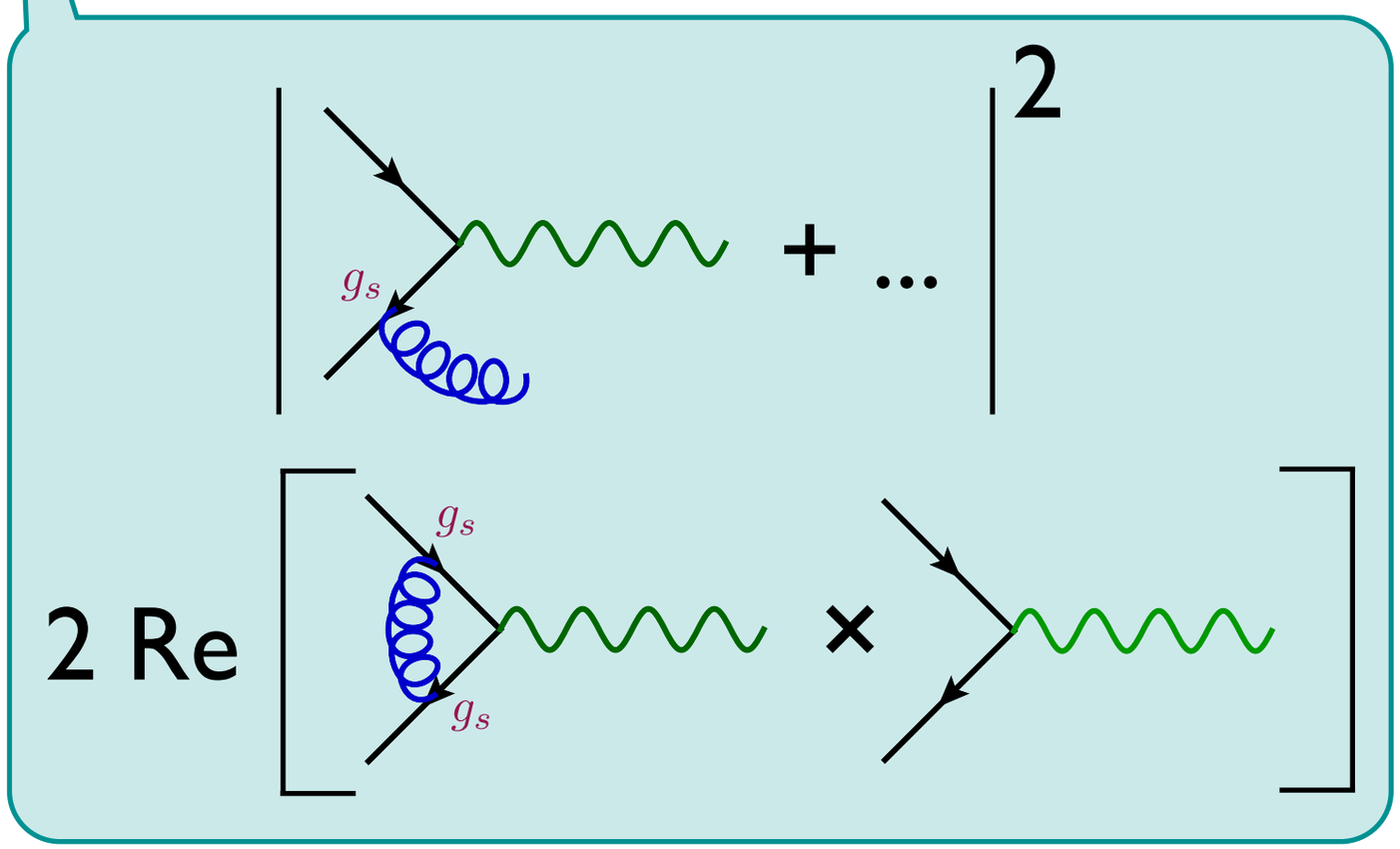


NLO predictions

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

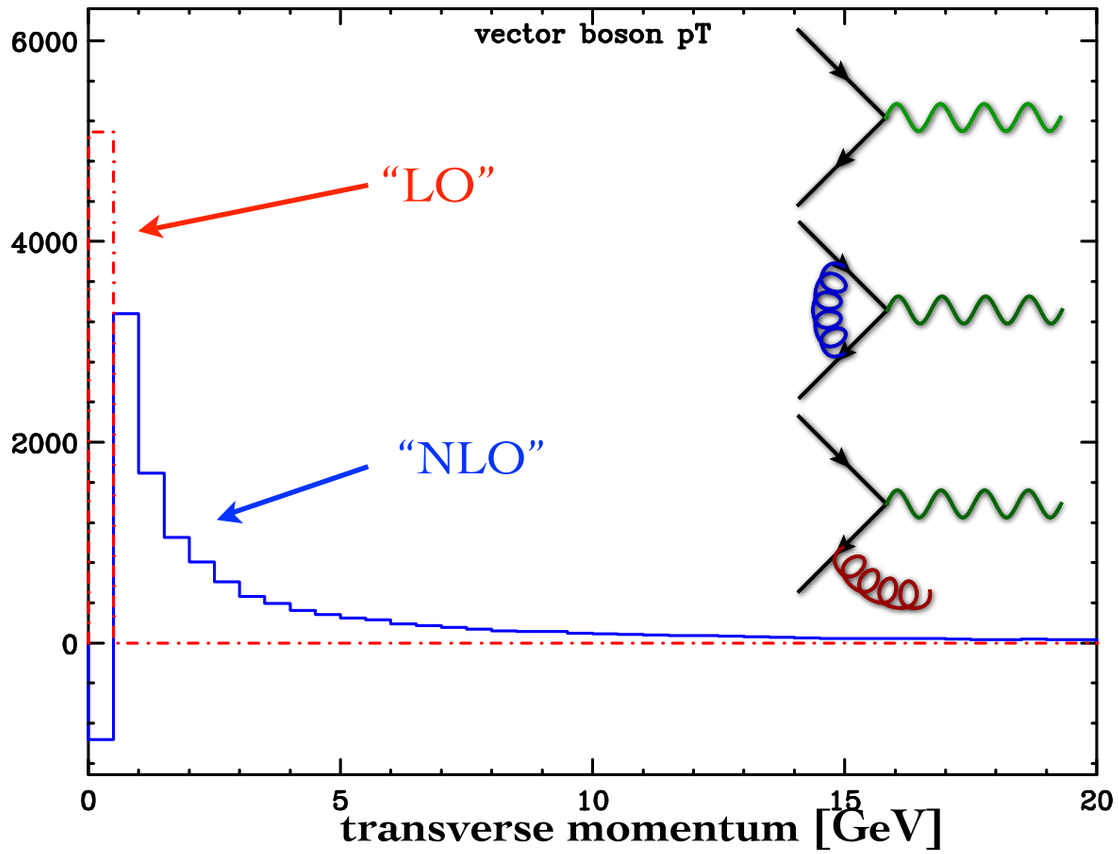
$$\hat{\sigma} = \sigma^{\text{Born}} \left(1 + \frac{\alpha_s}{2\pi} \sigma^{(1)} + \dots \right)$$



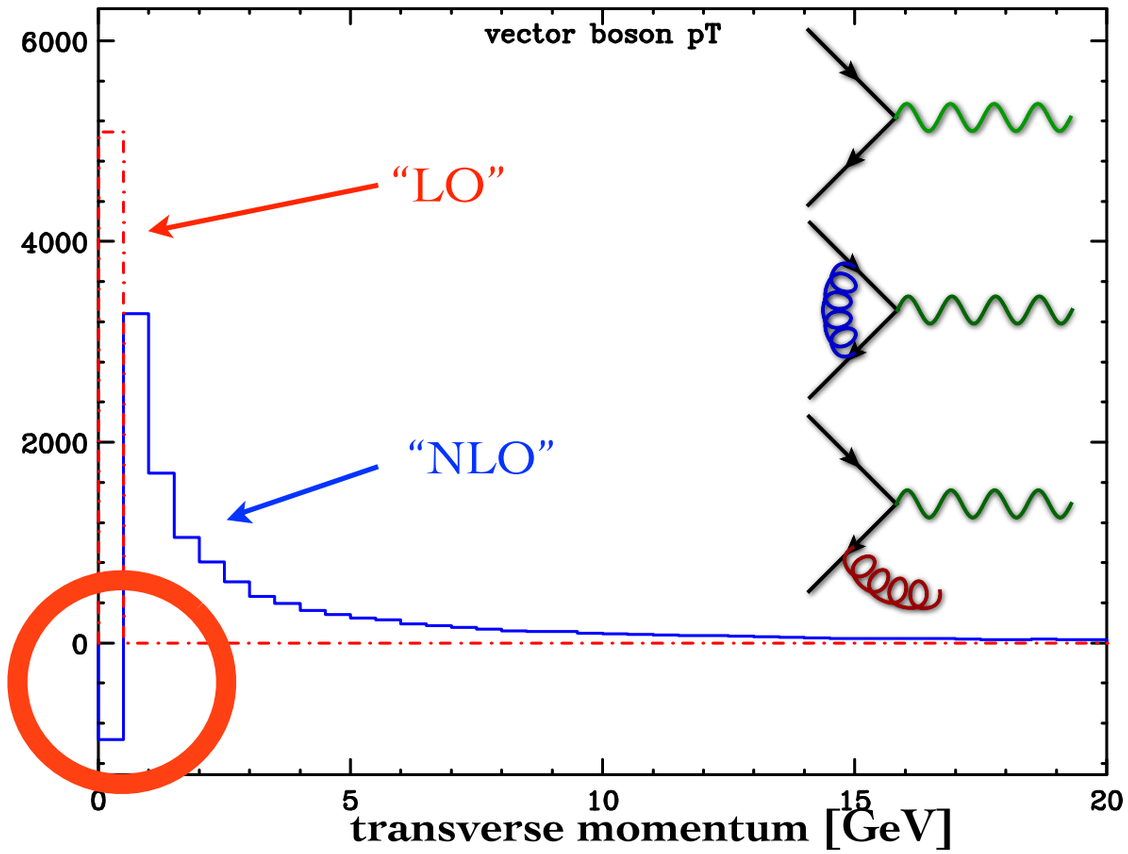


Not definite positive

Fixed Order calculations

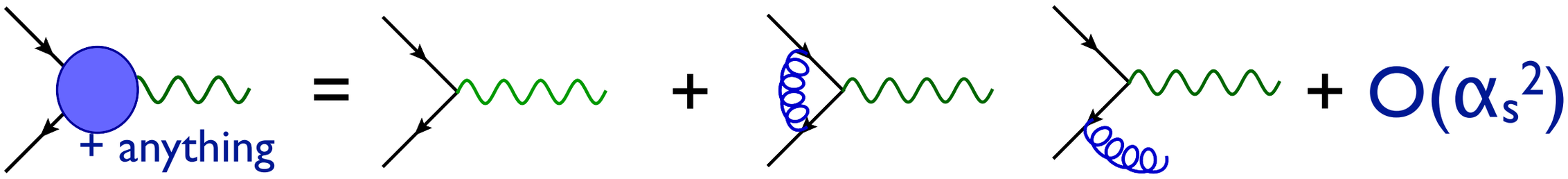


Fixed Order calculations



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 behavior for phase-space integration
  - How to match these processes to a parton shower without double counting

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

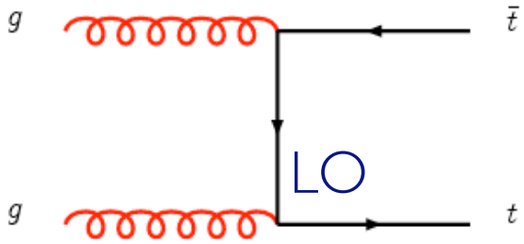
  $p_i \longrightarrow p_j + p_k$,

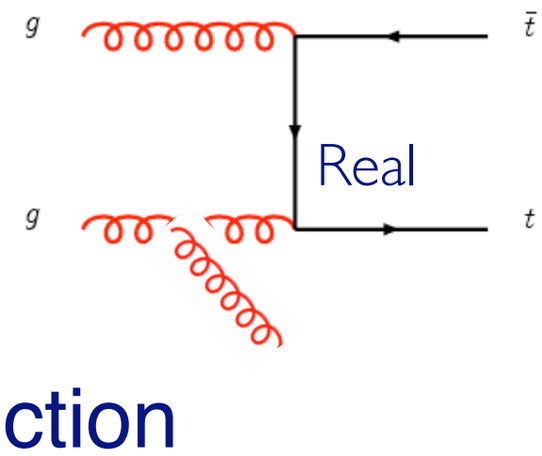
  whenever p_j and p_k are collinear or one of them is soft.

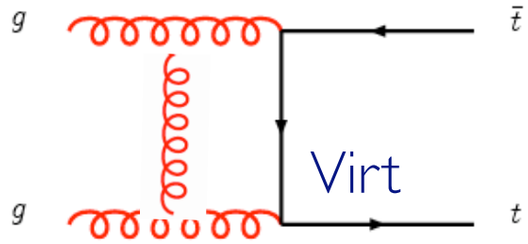
- Examples
  - "The number of gluons" produced in a collision is not an infrared safe observable
  - "The number of hard jets defined using the k_T algorithm with a transverse momentum above 40 GeV," produced in a collision is an infrared safe observable

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



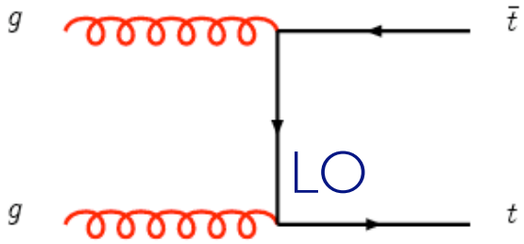


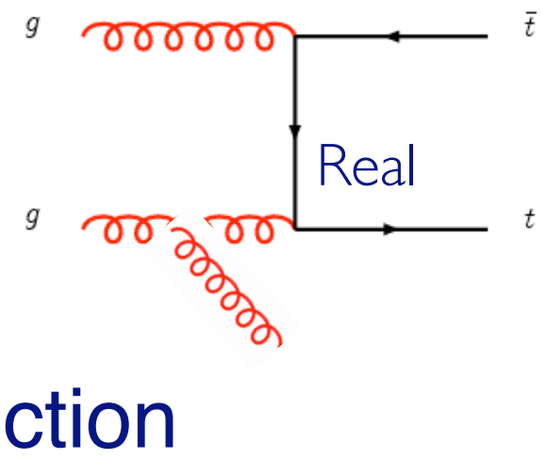


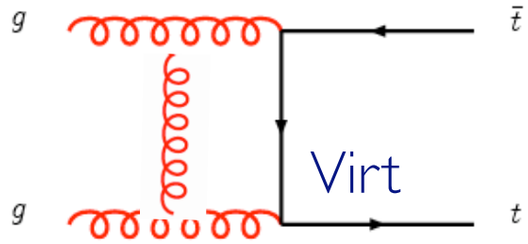
- Total cross section
- Transverse momentum of the top quark
- Transverse momentum of the top-antitop pair
- Transverse momentum of the jet
- Top-antitop invariant mass
- Azimuthal distance between the top and anti-top

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





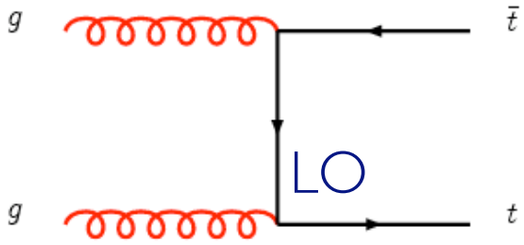


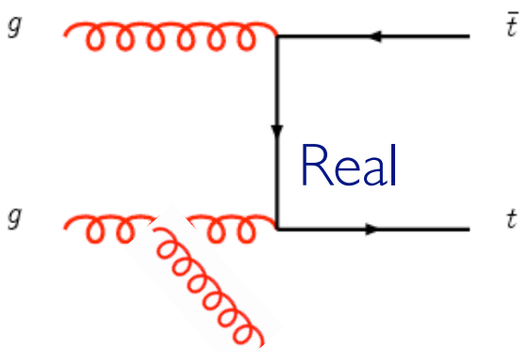
NLO?

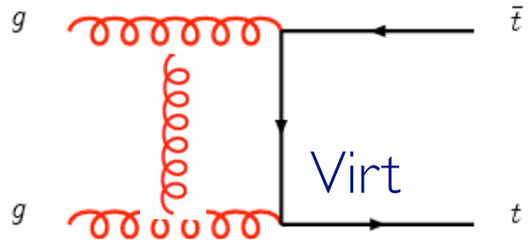
- Total cross section
- Transverse momentum of the top quark
- Transverse momentum of the top-antitop pair
- Transverse momentum of the jet
- Top-antitop invariant mass
- Azimuthal distance between the top and anti-top

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





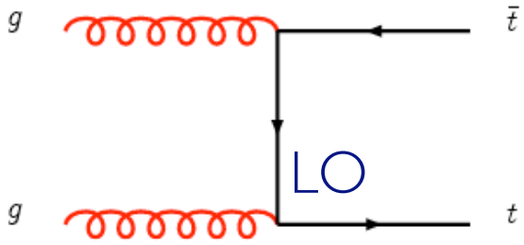


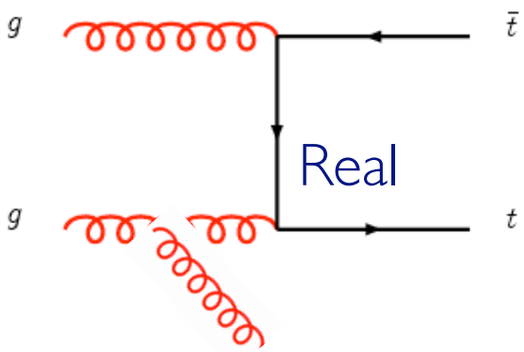
NLO? ✔

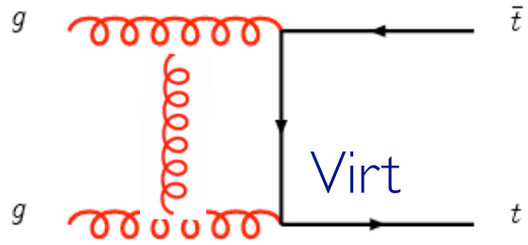
- Total cross section
- Transverse momentum of the top quark
- Transverse momentum of the top-antitop pair
- Transverse momentum of the jet
- Top-antitop invariant mass
- Azimuthal distance between the top and anti-top

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





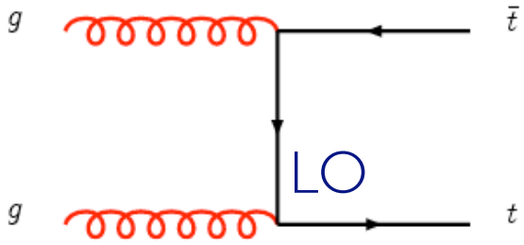


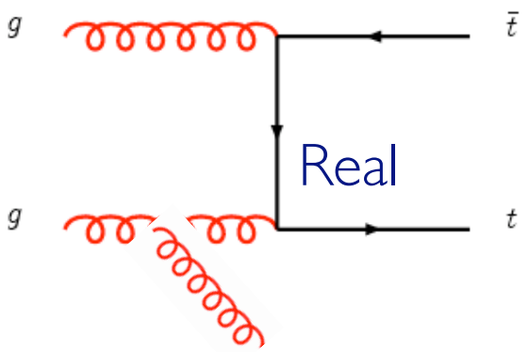
NLO?

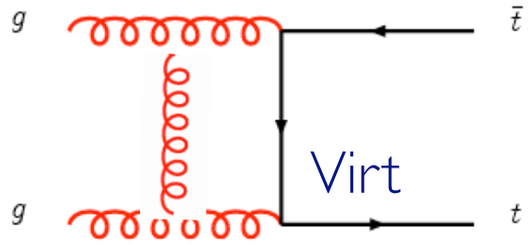
- Total cross section ✔
- Transverse momentum of the top quark ✔
- Transverse momentum of the top-antitop pair
- Transverse momentum of the jet
- Top-antitop invariant mass
- Azimuthal distance between the top and anti-top

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





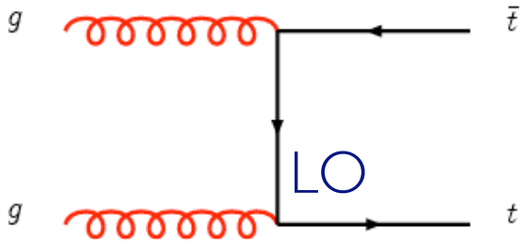


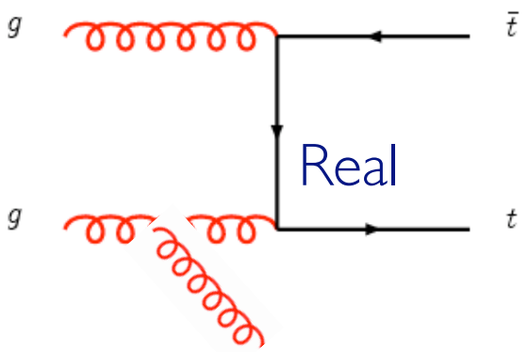
NLO?

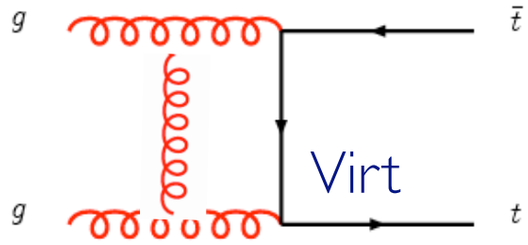
- Total cross section ✔
- Transverse momentum of the top quark ✔
- Transverse momentum of the top-antitop pair ✘
- Transverse momentum of the jet
- Top-antitop invariant mass
- Azimuthal distance between the top and anti-top

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





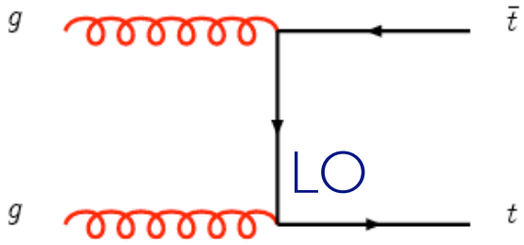


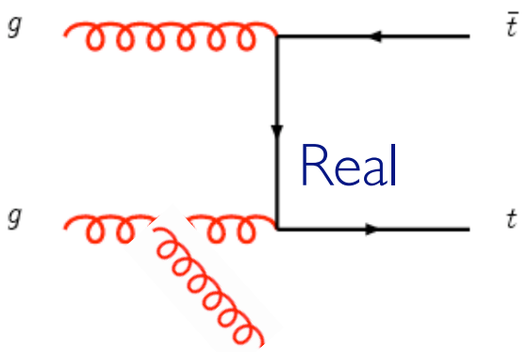
NLO?

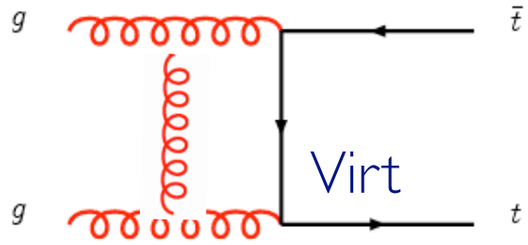
- Total cross section ✔
- Transverse momentum of the top quark ✔
- Transverse momentum of the top-antitop pair ✘
- Transverse momentum of the jet ✘
- Top-antitop invariant mass
- Azimuthal distance between the top and anti-top

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





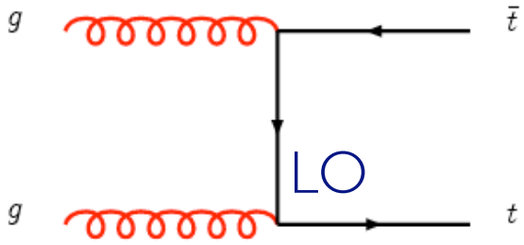


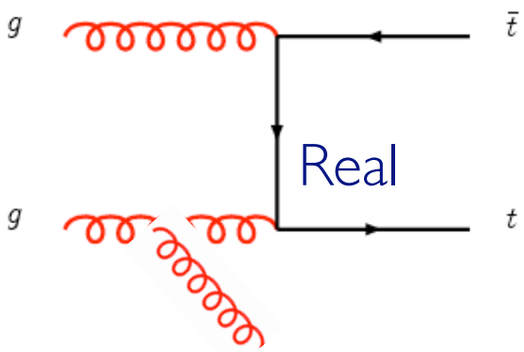
NLO?

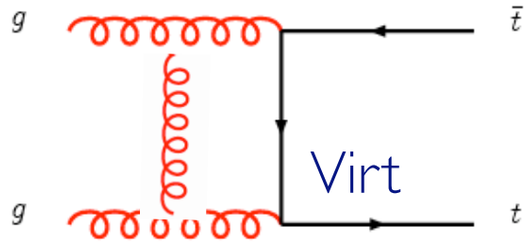
- Total cross section ✔
- Transverse momentum of the top quark ✔
- Transverse momentum of the top-antitop pair ✘
- Transverse momentum of the jet ✘
- Top-antitop invariant mass ✔
- Azimuthal distance between the top and anti-top

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?

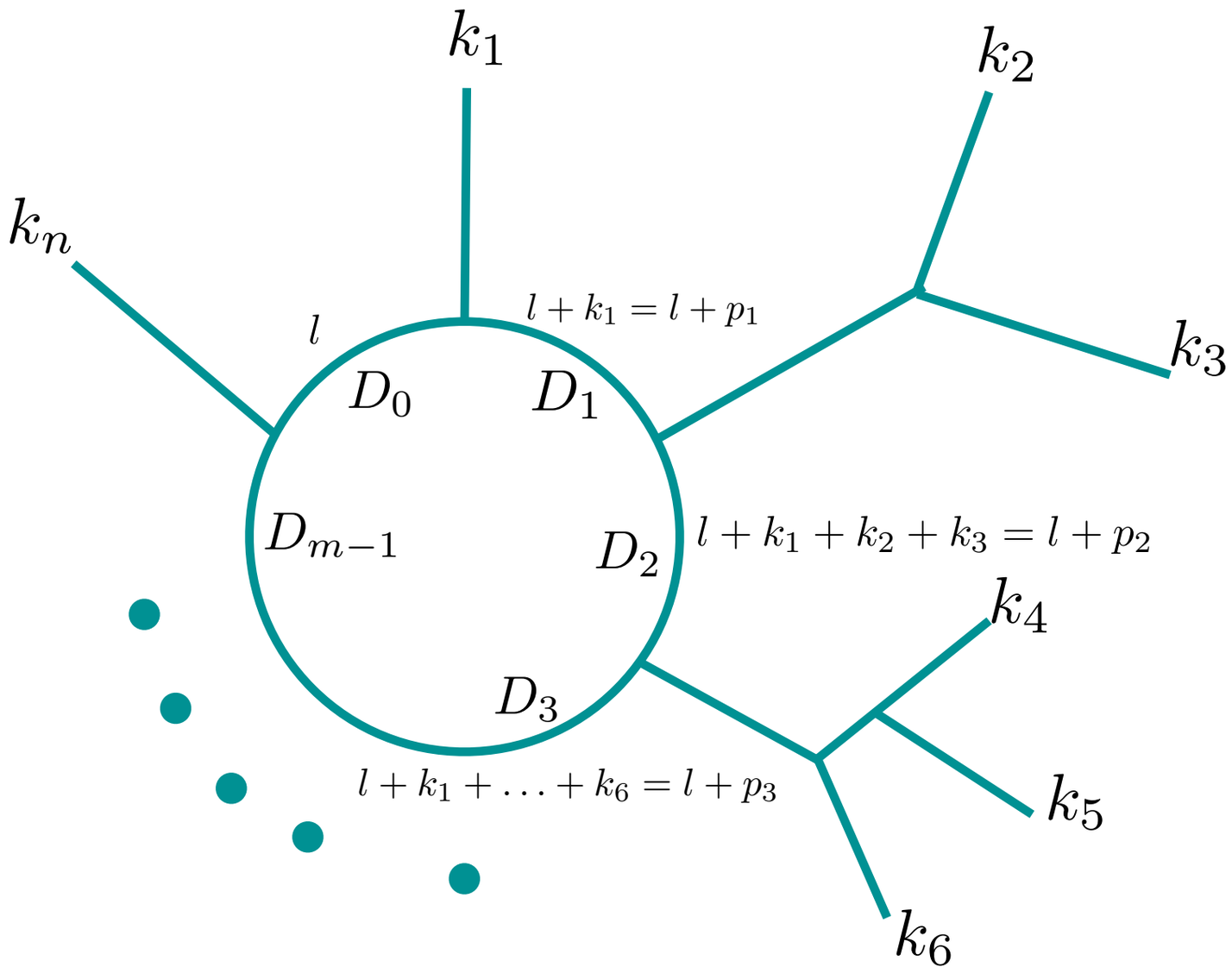
- Total cross section ✔
- Transverse momentum of the top quark ✔
- Transverse momentum of the top-antitop pair ✘
- Transverse momentum of the jet ✘
- Top-antitop invariant mass ✔
- Azimuthal distance between the top and anti-top ✘

Plan

- NLO Introduction
- Loop Computation
- Dealing with Singularities
- Matching@NLO
- Live Demo

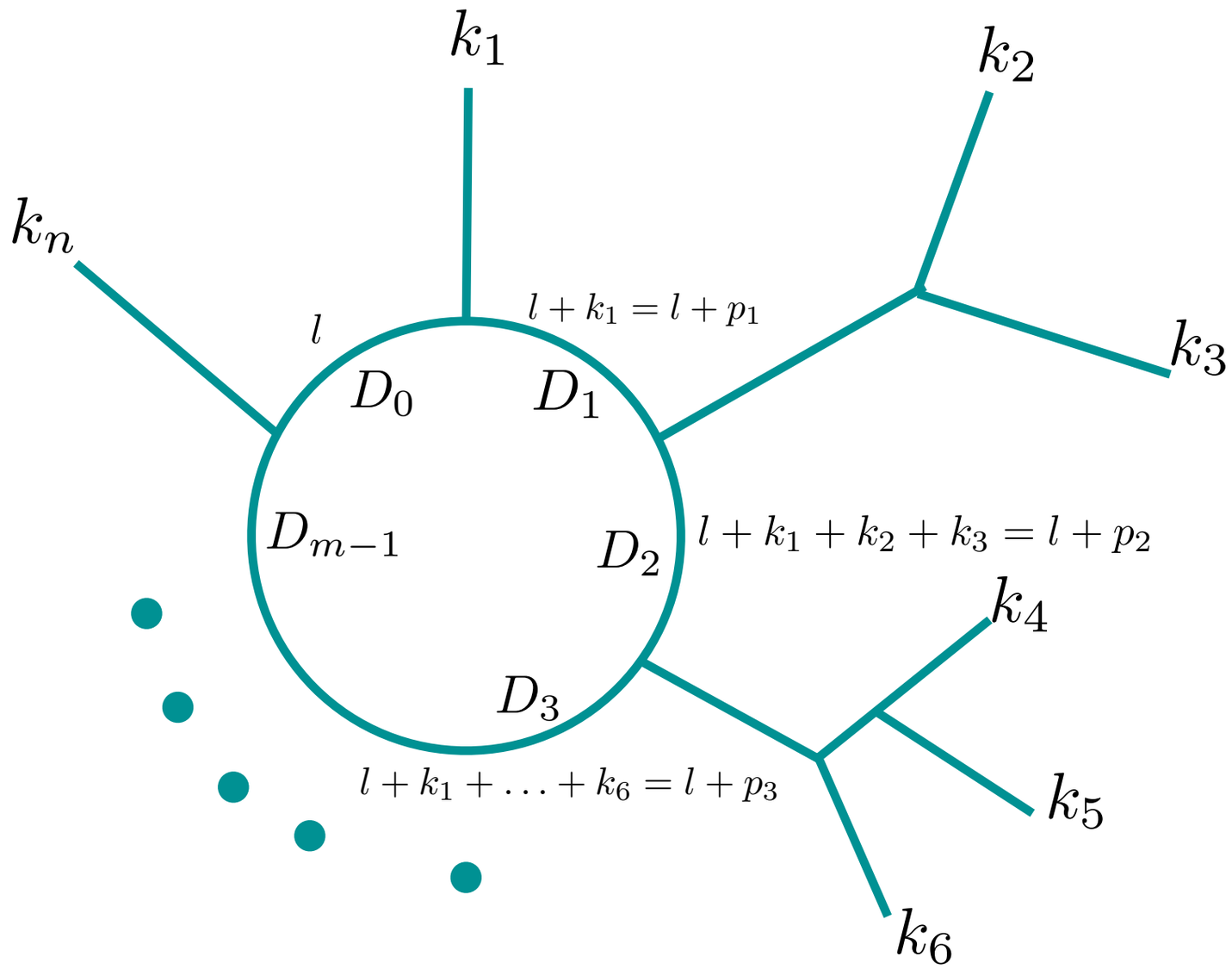
One Loop

- Consider this *m*-point loop diagram with *n* external momenta



One Loop

- Consider this *m*-point loop diagram with *n* external momenta



- The integral to compute is

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

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

Integrand reduction

Key Point

- 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

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

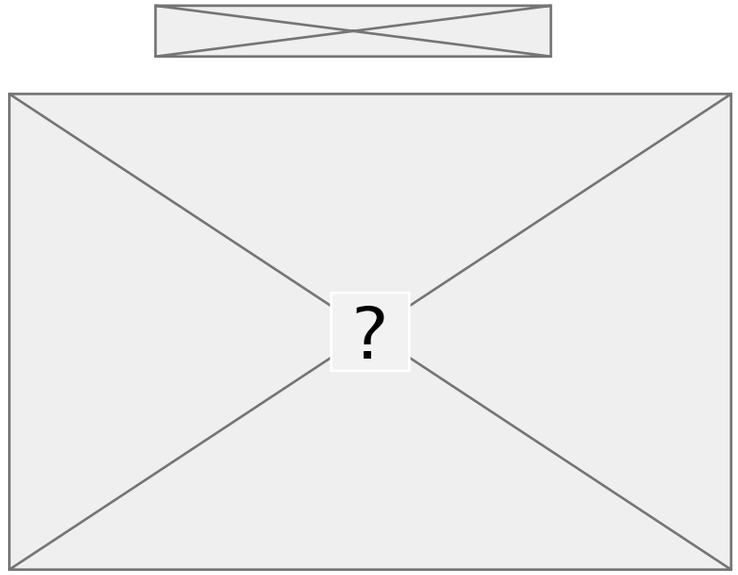
$$\mathcal{M}^{\text{1-loop}} = \sum_{i_0<i_1<i_2<i_3} d_{i_0 i_1 i_2 i_3} \text{Box}_{i_0 i_1 i_2 i_3}$$
$$+ \sum_{i_0<i_1<i_2} c_{i_0 i_1 i_2} \text{Triangle}_{i_0 i_1 i_2}$$
$$+ \sum_{i_0<i_1} b_{i_0 i_1} \text{Bubble}_{i_0 i_1}$$
$$+ \sum_{i_0} a_{i_0} \text{Tadpole}_{i_0}$$
$$+ R + \mathcal{O}(\epsilon)$$

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

Divergences

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

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



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

➡The $1/\epsilon$ 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)

Integrand reduction

Key Point

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

Two methods

- Passarino-Veltman
- OPP

Integrand reduction

Key Point

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

Two methods

- Passarino-Veltman
- OPP

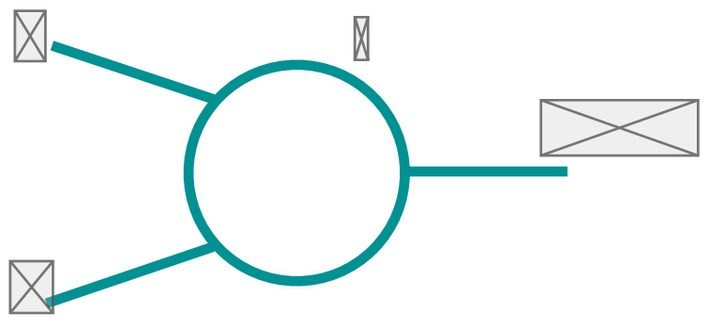
Standard Approach

- Passarino-Veltman reduction:
  - Reduce a general integral to "scalar integrals" by "completing the square"

Standard Approach

- Passarino-Veltman reduction:
  - Reduce a general integral to "scalar integrals" by "completing the square"

- Let's do an example:
  Suppose we want to calculate this triangle integral



Passarino-Veltman

Main Idea

- The only independent four vectors are p^μ and q^μ .Therefore, the integral must be proportional to those. We can set-up a system of linear equations.

Passarino-Veltman

Main Idea

- The only independent four vectors are p^μ and q^μ .Therefore, the integral must be proportional to those. We can set-up a system of linear equations.

Passarino-Veltman

Main Idea

- The only independent four vectors are p^μ and q^μ .Therefore, the integral must be proportional to those. We can set-up a system of linear equations.

Resolution (dropping the mass)

- contracting with 2**p* and 2**q*

Passarino-Veltman

Main Idea

- The only independent four vectors are p^μ and q^μ .Therefore, the integral must be proportional to those. We can set-up a system of linear equations.

Resolution (dropping the mass)

- contracting with 2**p* and 2**q*

Passarino-Veltman

Main Idea

- The only independent four vectors are p^μ and q^μ .Therefore, the integral must be proportional to those. We can set-up a system of linear equations.

Resolution (dropping the mass)

- contracting with 2**p* and 2**q*

Passarino-Veltman

Main Idea

- The only independent four vectors are p^μ and q^μ .Therefore, the integral must be proportional to those. We can set-up a system of linear equations.

Resolution (dropping the mass)

- contracting with 2**p* and 2**q*

Passarino-Veltman

Main Idea

- The only independent four vectors are p^μ and q^μ .Therefore, the integral must be proportional to those. We can set-up a system of linear equations.

Resolution (dropping the mass)

- contracting with 2**p* and 2**q*

Passarino-Veltman

Main Idea

- The only independent four vectors are p^μ and q^μ .Therefore, the integral must be proportional to those. We can set-up a system of linear equations.

Resolution (dropping the mass)

- contracting with 2**p* and 2**q*

Passarino-Veltman

Main Idea

- The only independent four vectors are p^μ and q^μ .Therefore, the integral must be proportional to those. We can set-up a system of linear equations.

Resolution (dropping the mass)

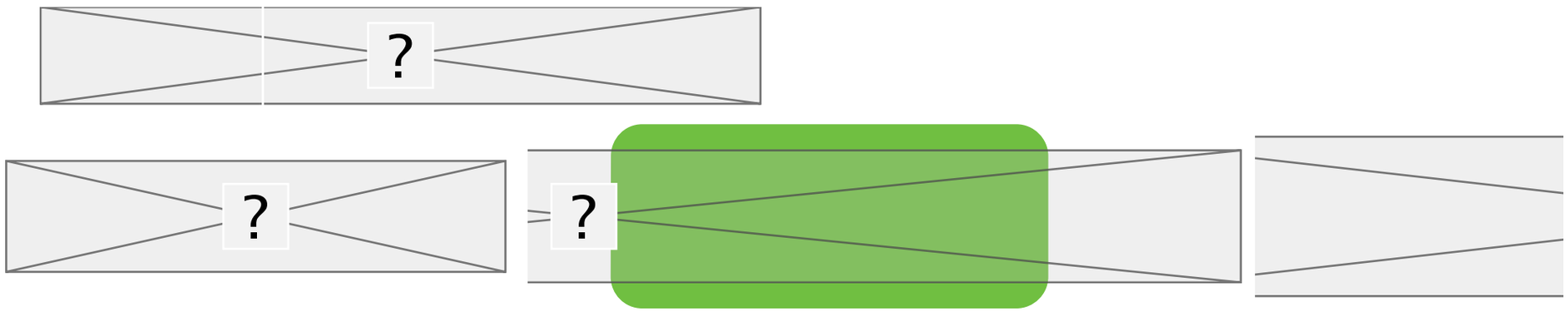
- contracting with 2**p* and 2**q*

Gram Determinant: G

Passarino-Veltman

Resolution (dropping the mass)

- contracting with $2*p$ and $2*q$

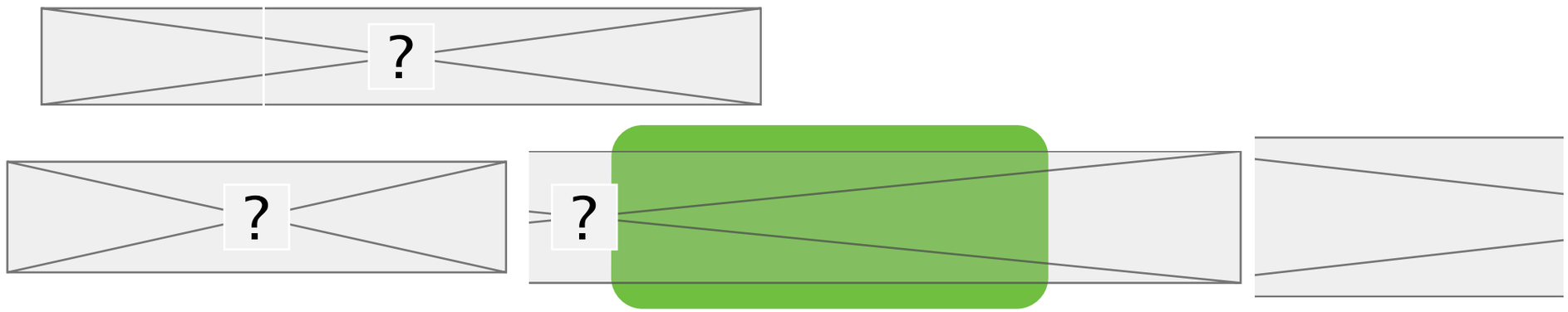


Gram Determinant: G

Passarino-Veltman

Resolution (dropping the mass)

- contracting with $2*p$ and $2*q$



Gram Determinant: G

Resolution (dropping the mass)

- express the integral as simpler integral

Passarino-Veltman

Resolution (dropping the mass)

- contracting with $2*p$ and $2*q$

Gram Determinant: G

Resolution (dropping the mass)

- express the integral as simpler integral

Passarino-Veltman

Resolution (dropping the mass)

- contracting with 2**p* and 2**q*

Gram Determinant: G

Resolution (dropping the mass)

- express the integral as simpler integral

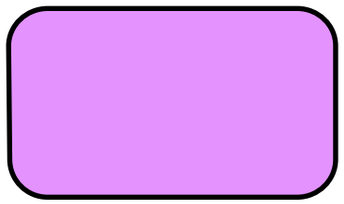
Scalar Integral: Know analytically

Passarino-Veltman

Resolution (dropping the mass)

- contracting with $2*p$ and $2*q$

Gram Determinant: G

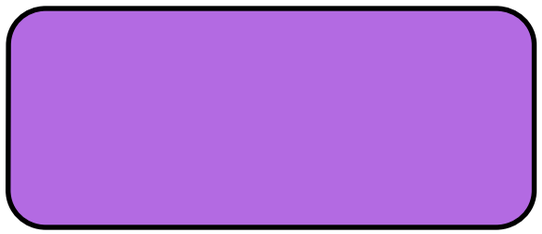


Already computed

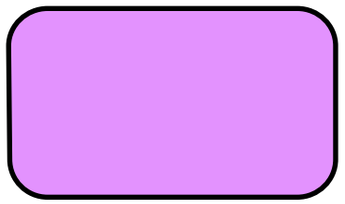
Passarino-Veltman

Resolution (dropping the mass)

- contracting with $2*p$ and $2*q$



Gram Determinant: G

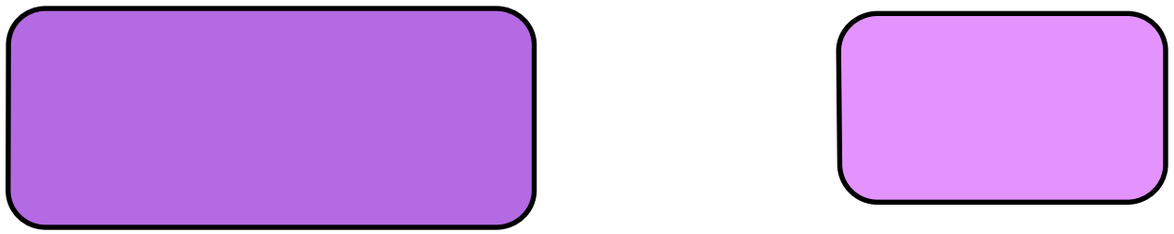


Already computed

Passarino-Veltman

Resolution (dropping the mass)

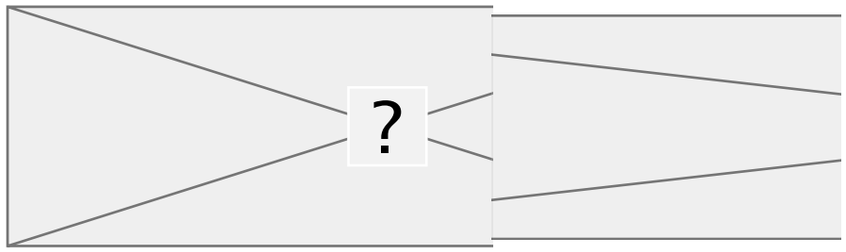
- contracting with 2**p* and 2**q*



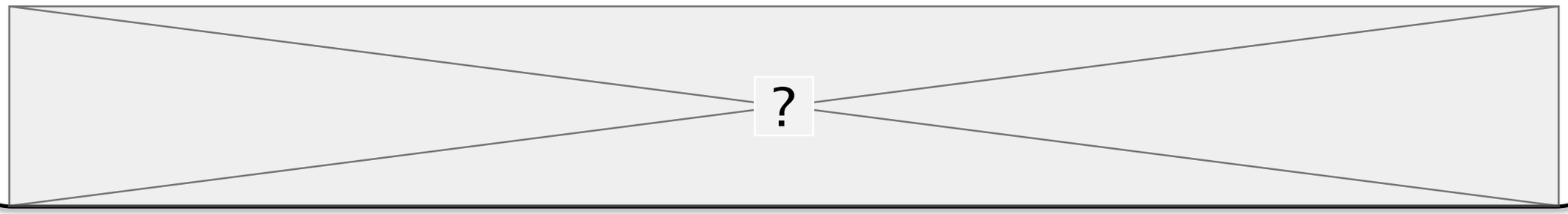
Gram Determinant: G

Final Step

- Inverting the Gram Determinant



- We have an expression in term of scalar integral



Already computed

Integrand reduction

Key Point

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

Two methods

- Passarino-Veltman
- OPP

Integrand reduction

Key Point

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

Two methods

- Passarino-Veltman
- OPP

OPP Reduction

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

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

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

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

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

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

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

Spurious term

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

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

How it works...

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

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

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

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

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

How it works...

$$N(l) = \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i$$

$$+ \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i$$

$$+ \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l)\right] \prod_{i\neq i_0,i_1}^{m-1} D_i$$

$$+ \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l)\right] \prod_{i\neq i_0}^{m-1} D_i$$

$$+ \tilde{P}(l) \prod_{i}^{m-1} D_i$$

$$= 0$$

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

How it works...

$$N(l) = \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i$$

$$+ \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i$$

$$+ \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l)\right] \prod_{i\neq i_0,i_1}^{m-1} D_i$$

$$+ \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l)\right] \prod_{i\neq i_0}^{m-1} D_i$$

$$+ \tilde{P}(l) \prod_{i}^{m-1} D_i$$

$$= 0$$

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...

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

Coefficient computed in a previous step

How it works...

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

Now we choose 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

Coefficient computed in a previous step

How it works...

$$N(l) = \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i$$

$$+ \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i$$

$$+ \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l)\right] \prod_{i\neq i_0,i_1}^{m-1} D_i$$

$$+ \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l)\right] \prod_{i\neq i_0}^{m-1} D_i$$

$$+ \tilde{P}(l) \prod_{i}^{m-1} D_i$$

$$= 0$$

Now we choose 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

Coefficient computed in a previous step

How it works...

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

Coefficient computed in a previous step

How it works...

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

Now, 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

Coefficient computed in a previous step

How it works...

$$
\begin{aligned}
N(l) = & \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l) \right] \prod_{i \neq i_0,i_1,i_2,i_3}^{m-1} D_i \\
& + \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l) \right] \prod_{i \neq i_0,i_1,i_2}^{m-1} D_i \\
& + \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l) \right] \prod_{i \neq i_0,i_1}^{m-1} D_i \\
& + \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l) \right] \prod_{i \neq i_0}^{m-1} D_i \\
& + \tilde{P}(l) \prod_{i}^{m-1} D_i \\
= & \ 0
\end{aligned}
$$

Now, 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

Coefficient computed in a previous step

How it works...

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

Now, choosing *l* such that

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

sets the last line to zero

Coefficient computed in a previous step

How it works...

$$N(l) = \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l)\right] \prod_{i\neq i_0,i_1,i_2,i_3}^{m-1} D_i$$
$$+ \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l)\right] \prod_{i\neq i_0,i_1,i_2}^{m-1} D_i$$
$$+ \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l)\right] \prod_{i\neq i_0,i_1}^{m-1} D_i$$
$$+ \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l)\right] \prod_{i\neq i_0}^{m-1} D_i$$
$$+ \tilde{P}(l) \prod_{i}^{m-1} D_i$$
$$= 0$$

Now, choosing *l* such that

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

sets the last line to zero

Coefficient computed in a previous step

How it works...

$$N(l) = \sum_{i_0<i_1<i_2<i_3}^{m-1} \left[d_{i_0 i_1 i_2 i_3} + \tilde{d}_{i_0 i_1 i_2 i_3}(l) \right] \prod_{i \neq i_0, i_1, i_2, i_3}^{m-1} D_i$$

$$+ \sum_{i_0<i_1<i_2}^{m-1} \left[c_{i_0 i_1 i_2} + \tilde{c}_{i_0 i_1 i_2}(l) \right] \prod_{i \neq i_0, i_1, i_2}^{m-1} D_i$$

$$+ \sum_{i_0<i_1}^{m-1} \left[b_{i_0 i_1} + \tilde{b}_{i_0 i_1}(l) \right] \prod_{i \neq i_0, i_1}^{m-1} D_i$$

$$+ \sum_{i_0}^{m-1} \left[a_{i_0} + \tilde{a}_{i_0}(l) \right] \prod_{i \neq i_0}^{m-1} D_i$$

$$+ \tilde{P}(l) \prod_{i}^{m-1} D_i$$

Now, choosing arbitrary *l*

Coefficient computed in a previous step

How it works...

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

We have our Numerator!

Coefficient computed in a previous step

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)

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

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

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

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

$$R = R_1 + R_2$$

- 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

$$R = R_1 + R_2$$

- 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

$$R = R_1 + R_2$$

- 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)

Integrand reduction

Key Point

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

Two methods

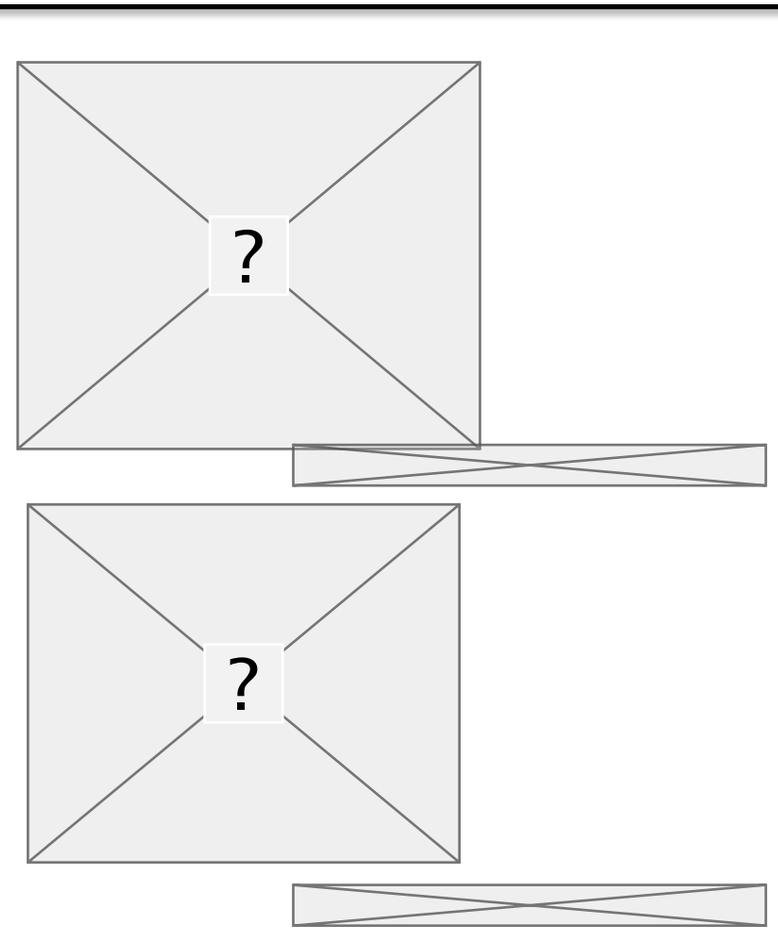
- Passarino-Veltman
- OPP

One Tool

- MadLoop

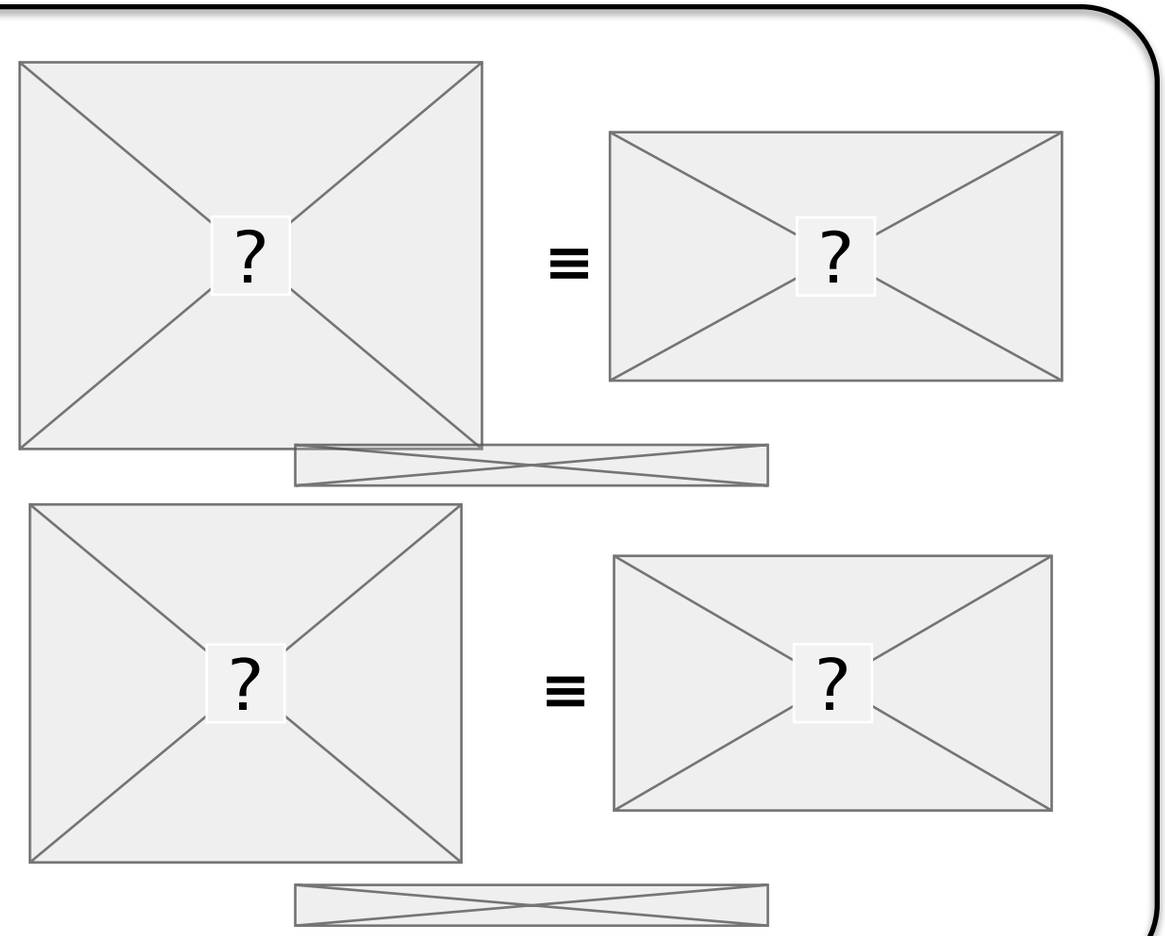
Generating loop diagrams

- Instead of using an external tool for loop diagram generation, we recycle MadGraph5 power for tree level diagram generation.
- A loop diagrams with the loop cut open has to extra external particles. Consider $e^+e^- \rightarrow u\ u\sim\ u\ u\sim$ (loop particles are in green). MadGraph will generate 8 L-cut diagrams. Here are two of them:



Generating loop diagrams

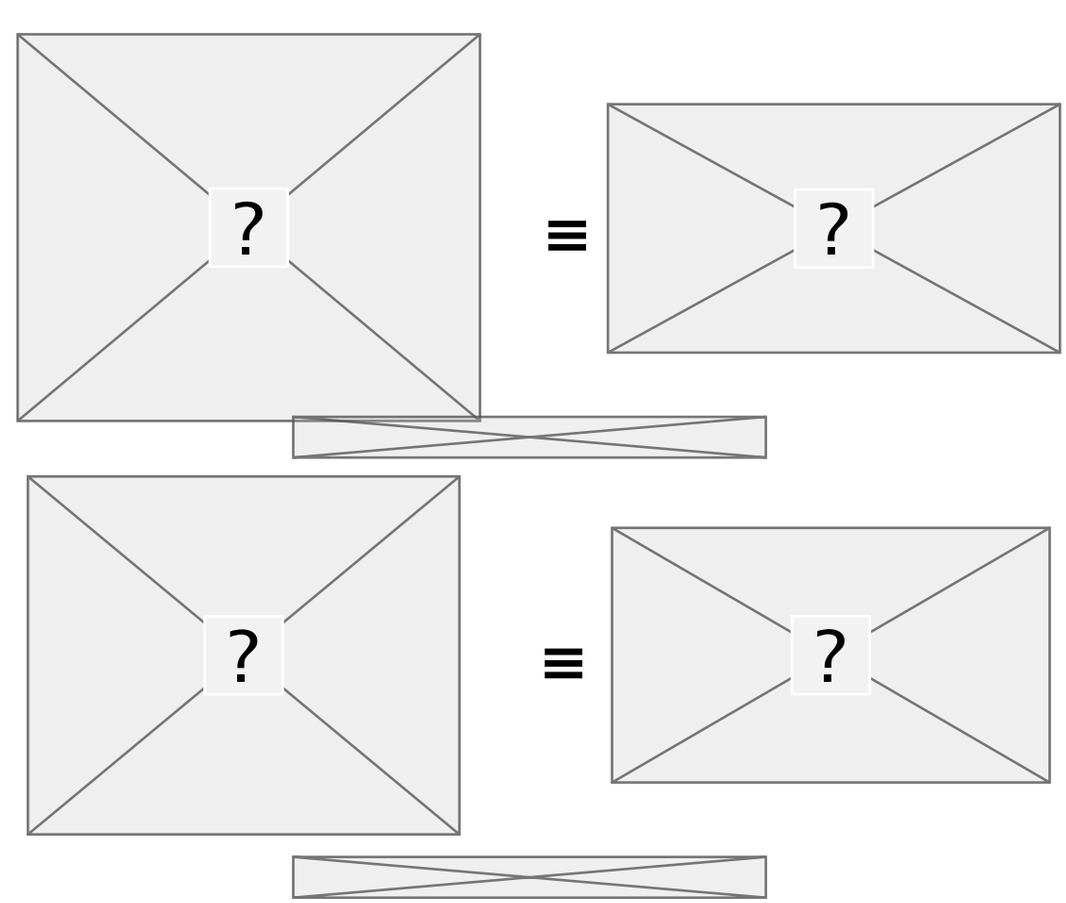
- Instead of using an external tool for loop diagram generation, we recycle MadGraph5 power for tree level diagram generation.
- A loop diagrams with the loop cut open has to extra external particles. Consider $e^+e^- \rightarrow u\ u\sim\ u\ u\sim$ (loop particles are in green). MadGraph will generate 8 L-cut diagrams. Here are two of them:



Generating loop diagrams

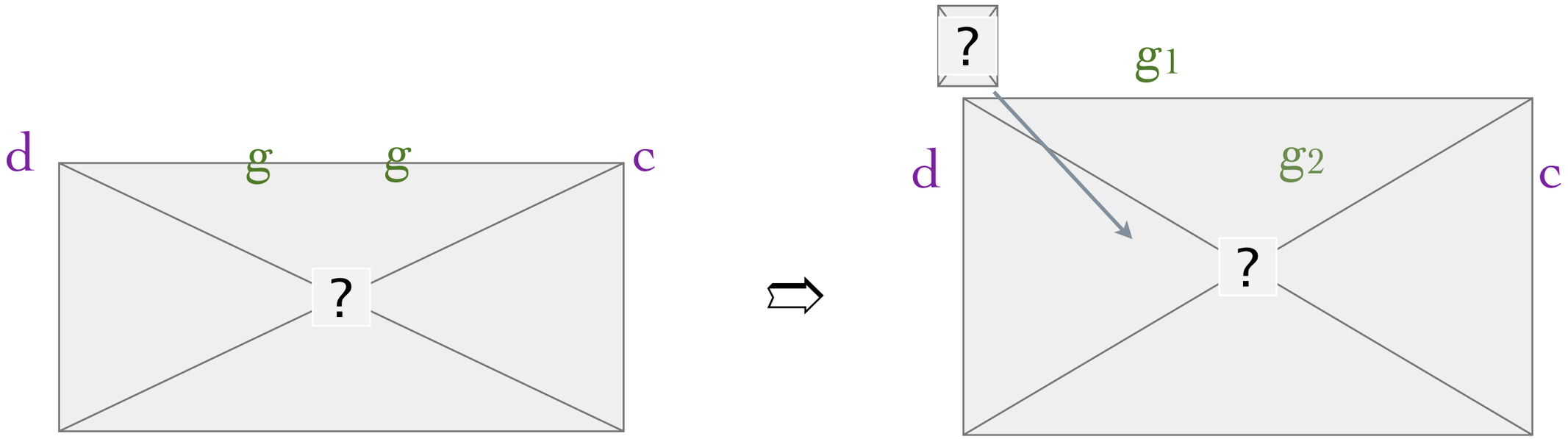
- Instead of using an external tool for loop diagram generation, we recycle MadGraph5 power for tree level diagram generation.
- A loop diagrams with the loop cut open has to extra external particles. Consider $e^+e^- \rightarrow u\ u\sim\ u\ u\sim$ (loop particles are in green). MadGraph will generate 8 L-cut diagrams. Here are two of them:

- All diagrams with two extra particles are generated and the ones that are needed are filtered out
- Additional filter to eliminate tadpoles and bubbles attached to external lines

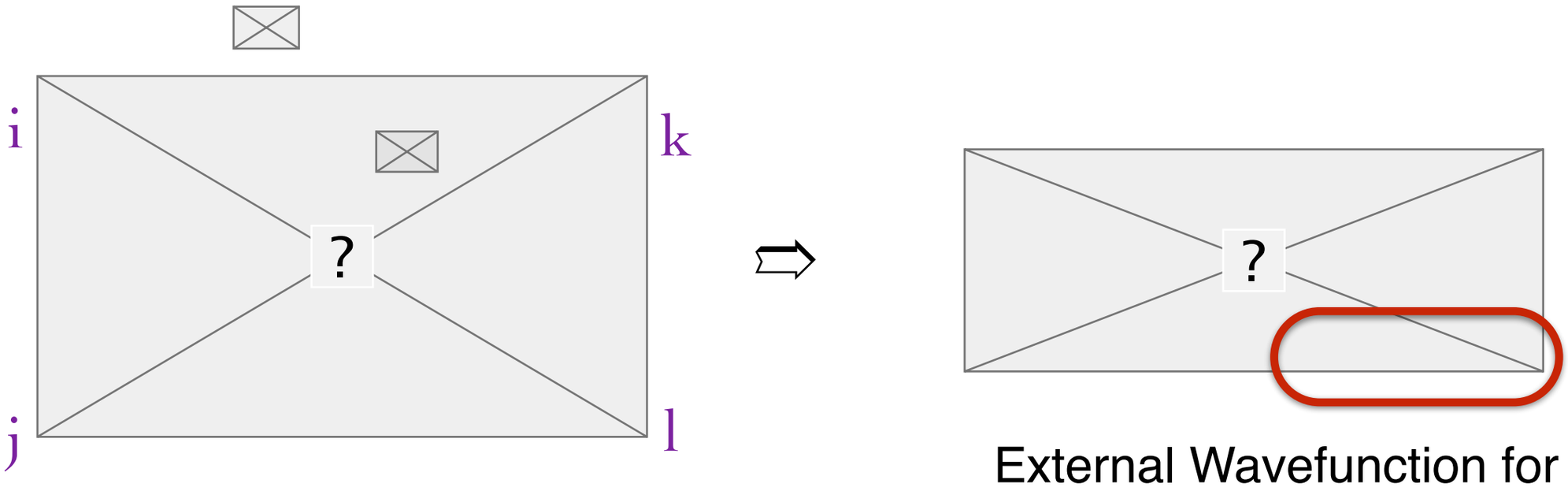


Numerator

- We want to use (modified) HELAS method



- Closing the lorentz trace :



External Wavefunction for HELAS

MadLoop

- Other modifications :
  - ➥ Allow for the loop momentum to be complex
  - ➥ Remove the denominator of the loop propagators
  - ➥ Close the color trace

ALOHA

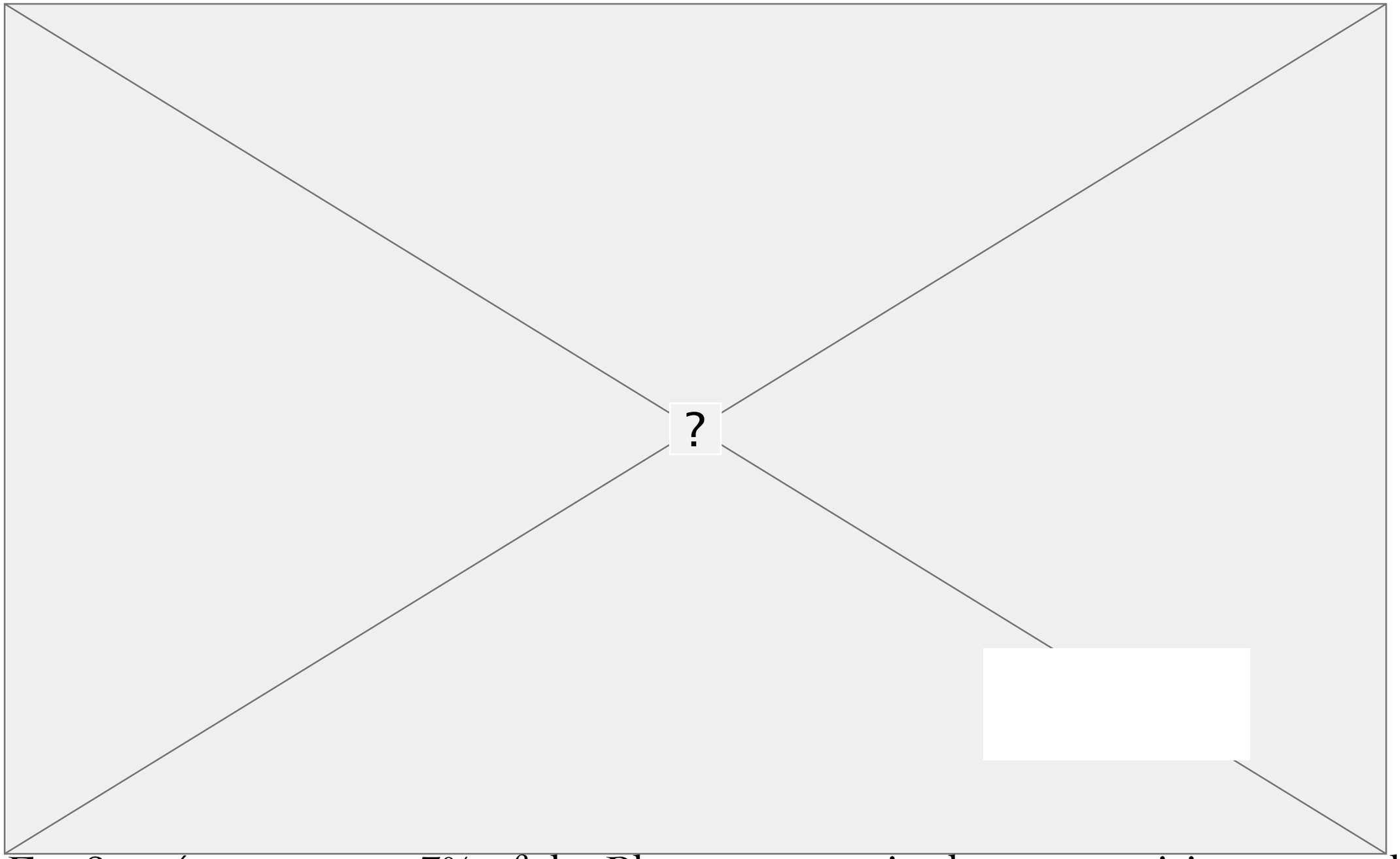
- Ok, now this gives you $\mathcal{N}(l^\mu)$, the integrand numerator to be fed to CT!

- But this is **SLOW**!!
- We have to compute this numerator ~ 50 times for each phase-space point!
- Idea instead of computing the numerator compute the polynomial form

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

[S. Pozzorini & al. hep-ph/1111.5206]

Numerical Stability



- For 2 to 4 processes, ~7% of the Phase-space point have a precision worse than 1e-3

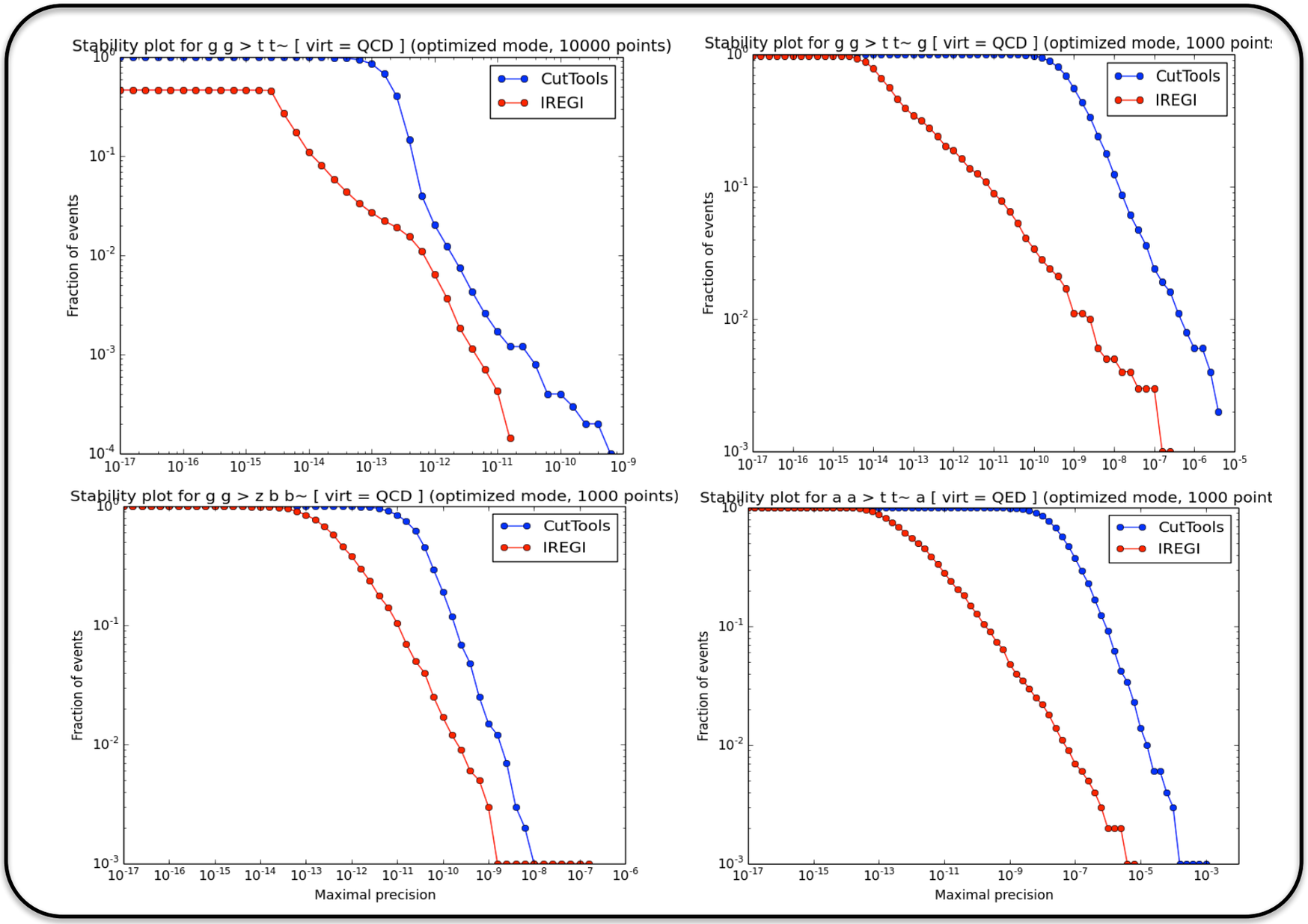
  ➡ Previous solution pass to quadruple precision (extremelly slow)

IREGI

- New Solution use IREGI: a TIR program

  ➡ Slower than previous method but faster than quadruple precision

  ➡Usually less uncertainty (and not for the same PS point)



[H.-shao]

Plan

- NLO Introduction
- Loop Computation
- Dealing with Singularities
- Matching@NLO
- Live Demo

phase-space integration

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

phase-space integration

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

- This sum is Finite (KLN theory)

phase-space integration

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

- This sum is Finite (KLN theory)
- Each piece is divergent

phase-space integration

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

- This sum is Finite (KLN theory)
- Each piece is divergent
  - Cannot use a finite value for the dimensional regulator and take the limit to zero in a numerical code

phase-space integration

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

- This sum is Finite (KLN theory)
- Each piece is divergent
  - Cannot use a finite value for the dimensional regulator and take the limit to zero in a numerical code
- We have to cancel the divergences explicitly

phase-space integration

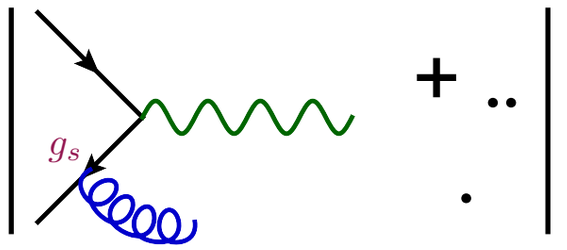
$$\sigma^{\mathrm{NLO}} \sim \int d^4\Phi_m \, B(\Phi_m) + \int d^4\Phi_m \int_{\mathrm{loop}} d^d l \, V(\Phi_m) + \int d^d\Phi_{m+1} \, R(\Phi_{m+1})$$

- This sum is Finite (KLN theory)
- Each piece is divergent
  - Cannot use a finite value for the dimensional regulator and take the limit to zero in a numerical code
- We have to cancel the divergences explicitly
- Use a subtraction method to explicitly factor out the divergences from the phase-space integrals

Example

$$\int_0^1 dx\, f(x) \qquad f(x) = \frac{g(x)}{x} \qquad g(x) \text{ Finite everywhere}$$

- Type of Divergencies of the real



- 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

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

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

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

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

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

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

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

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

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

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

- We have factored out the $1/\epsilon$ divergence and are left with a finite integral

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 $1/\epsilon$ divergence and are left with a finite integral
- According to the KLN theorem the divergence cancels against the virtual corrections

Limitations

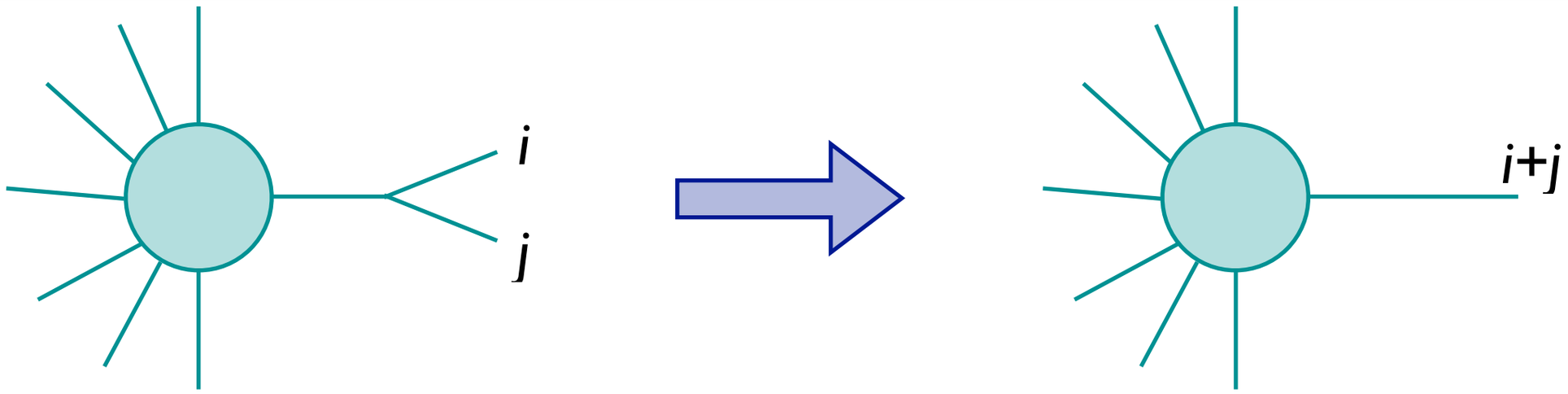
Subtraction:

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

"Plus distribution"

- 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 if 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

counter events



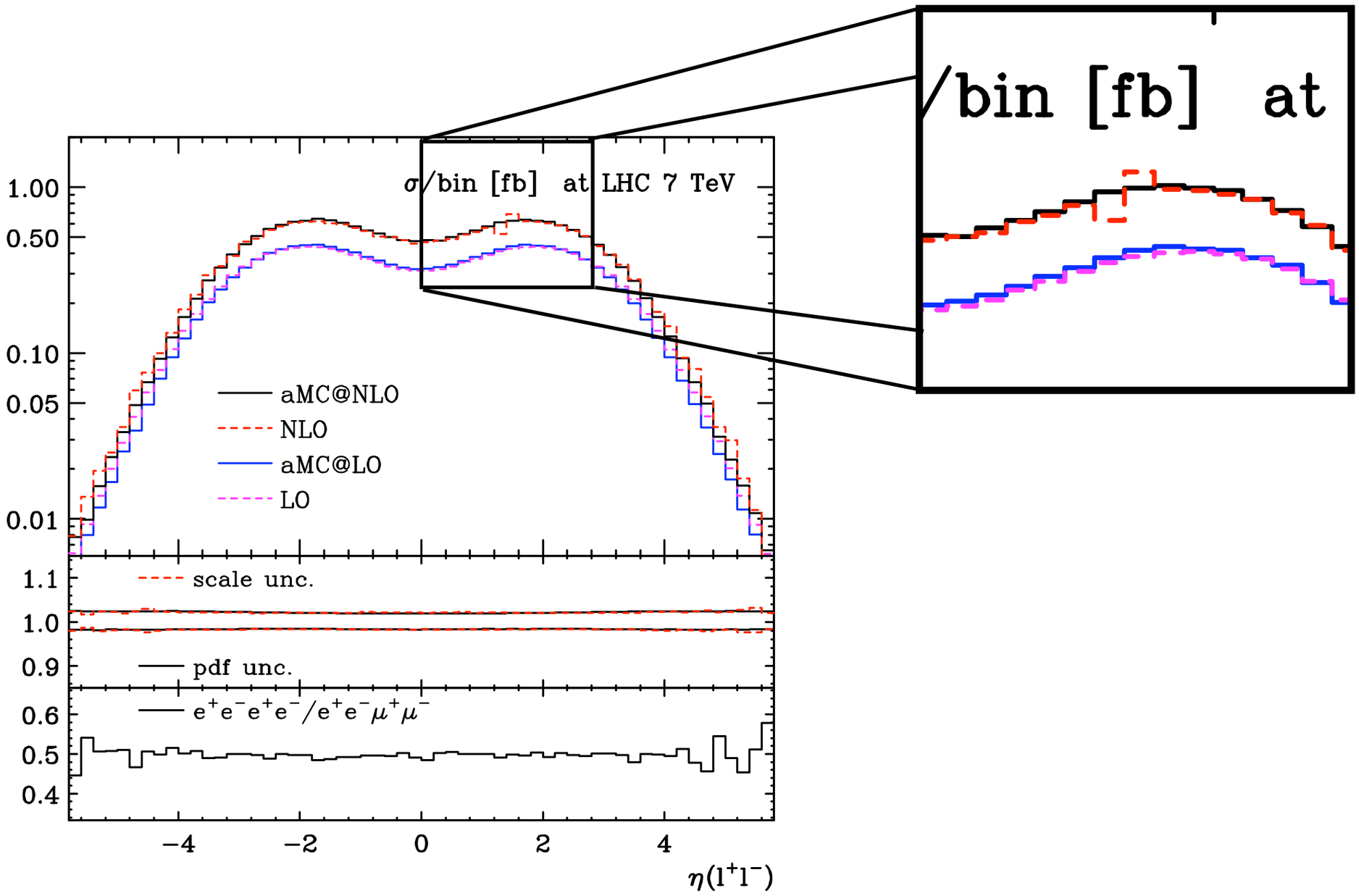
Real emission

Subtraction term

- 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
  - Use IR-safe observables and don't ask for infinite resolution! (KLN theorem)

4 charged lepton

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



Plan

- NLO Introduction
- Loop Computation
- Dealing with Singularities
- Matching@NLO
- Live Demo

NLO+PS matching

ME

1. Fixed order calculation
2. Computationally expensive
3. Limited number of particles
4. Valid when partons are hard and well separated
5. Quantum interference correct
6. Needed for multi-jet description

Shower MC

1. Resums logs to all orders
2. Computationally cheap
3. No limit on particle multiplicity
4. Valid when partons are collinear and/or soft
5. Partial interference through angular ordering
6. Needed for hadronization

Approaches are complementary: merge them!

Difficulty: avoid double counting, ensure smooth distributions

NLO+PS matching

ME

1. Fixed order calculation
2. Computationally expensive
3. Limited number of particles
4. Valid when partons are hard and well separated
5. Quantum interference correct
6. Needed for multi-jet description

Shower MC

1. Resums logs to all orders
2. Computationally cheap
3. No limit on particle multiplicity
4. Valid when partons are collinear and/or soft
5. Partial interference through angular ordering
6. Needed for hadronization

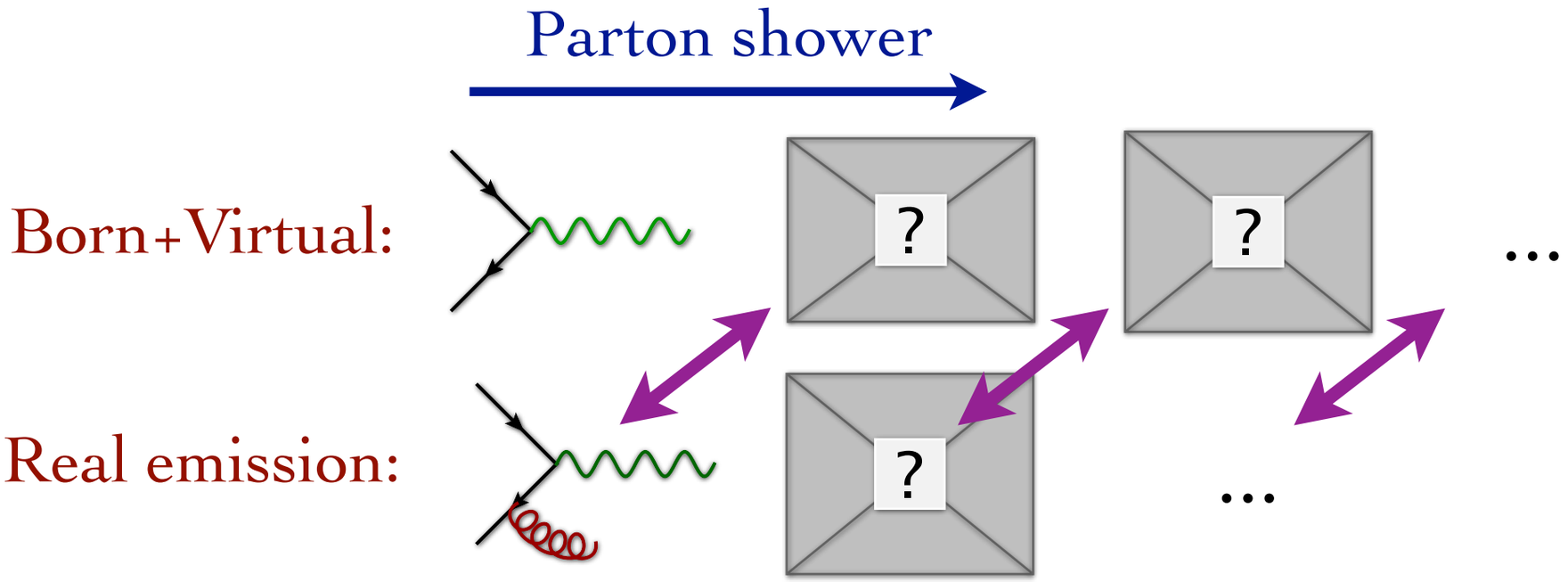
No longer true at NLO!

Approaches are complementary: merge them!

Difficulty: avoid double counting, ensure smooth distributions

Matching NLO

- At NLO one faces even more severe double-counting issues:



- And also part of the virtual contribution is double counted through the definition of the Sudakov factor Δ

Double counting

- Since $\Delta = 1 - P$, Δ contains contributions from the virtual corrections implicitly
- Because at NLO the virtual corrections are already included via explicit matrix elements, Δ is double counting with the virtual corrections
- In fact, because the shower is unitary, what we are double counting in the real emission corrections is exactly equal to what we are double counting in the virtual corrections (but with opposite sign)!

MC@NLO procedure

[Frixione & Webber (2002)]

- To remove the double counting, we can add and subtract the same term to the *m* and *m*+1 body configurations

MC@NLO procedure

[Frixione & Webber (2002)]

- To remove the double counting, we can add and subtract the same term to the *m* and *m*+1 body configurations

MC@NLO procedure

[Frixione & Webber (2002)]

- To remove the double counting, we can add and subtract the same term to the *m* and *m*+1 body configurations

- Where the *MC* are defined to be the contribution of the parton shower to get from the *m* body Born final state to the *m*+1 body real emission final state

MC@NLO procedure

[Frixione & Webber (2002)]

- To remove the double counting, we can add and subtract the same term to the *m* and *m*+1 body configurations

- Where the *MC* are defined to be the contribution of the parton shower to get from the *m* body Born final state to the *m*+1 body real emission final state

MC@NLO procedure

[Frixione & Webber (2002)]

- To remove the double counting, we can add and subtract the same term to the *m* and *m*+1 body configurations

- Where the *MC* are defined to be the contribution of the parton shower to get from the *m* body Born final state to the *m*+1 body real emission final state

MC@NLO properties

- Good features of including the subtraction counter terms

  1. **Double counting avoided**: The rate expanded at NLO coincides with the total NLO cross section

  2. **Smooth matching**: MC@NLO coincides (in shape) with the parton shower in the soft/collinear region, while it agrees with the NLO in the hard region

  3. **Stability:** weights associated to different multiplicities are separately finite. The *MC* term has the same infrared behavior as the real emission (there is a subtlety for the soft divergence)

Double counting avoided

- Expanded at NLO

Smooth matching

- Smooth matching:
  - Soft/collinear region: $R \simeq MC \Rightarrow d\sigma_{\mathrm{MC@NLO}} \sim I^{(m)}_{\mathrm{MC}}(O)dO$
  - Hard region, shower effects suppressed, ie.

$$MC \simeq 0 \quad I^{(m)}_{\mathrm{MC}}(O) \simeq 0 \quad I^{(m+1)}_{\mathrm{MC}}(O) \simeq 1$$

$$\Rightarrow d\sigma_{\mathrm{MC@NLO}} \sim d\Phi_{m+1} R$$

Stability & unweighting

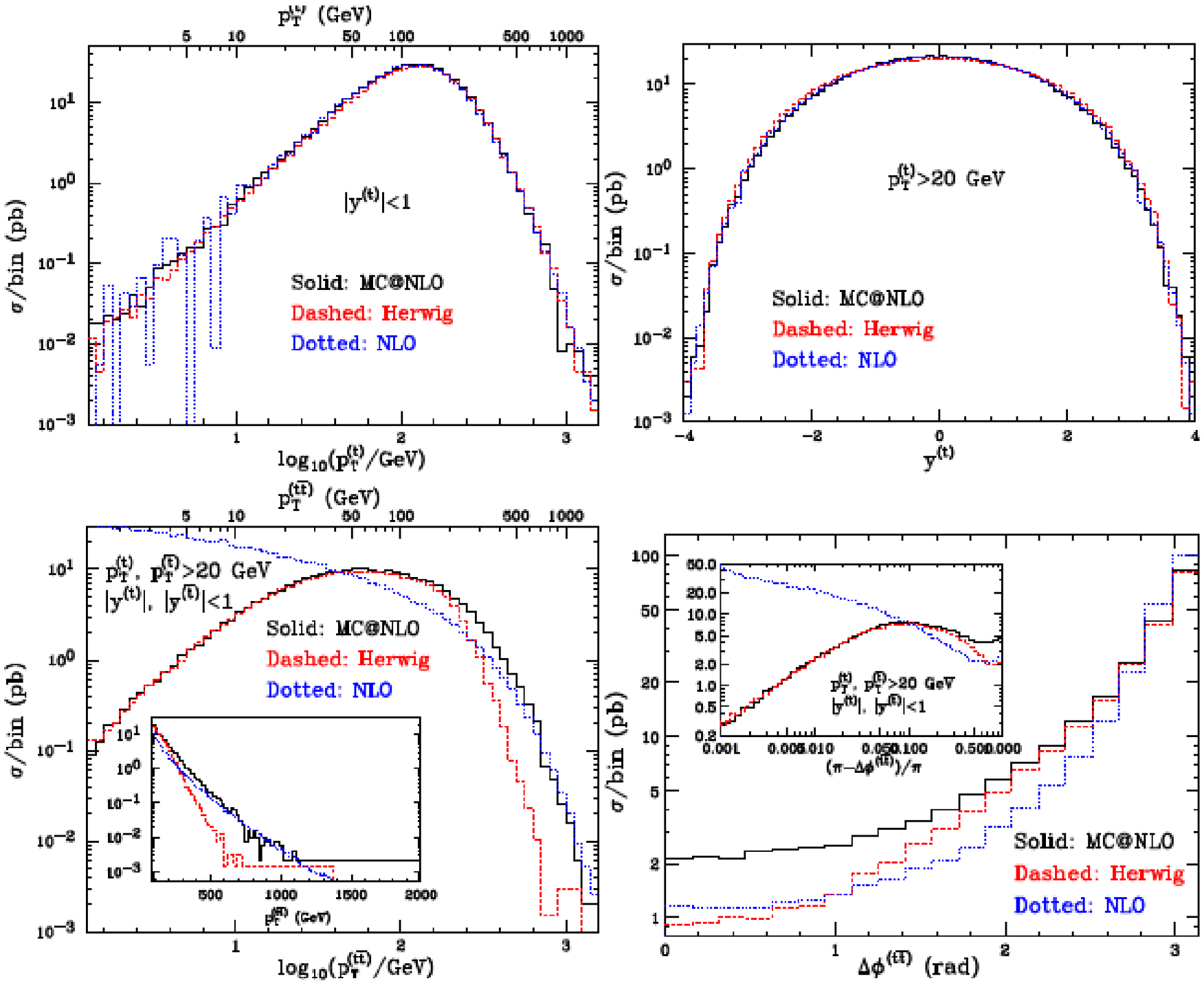
- The MC subtraction terms are defined to be what the shower does to get from the m to the m+1 body matrix elements. Therefore the cancellation of singularities is exact in the (R - MC) term

- The integral is bounded all over phase-space; we can therefore generate unweighted events!
  - "S-events" (which have m body kinematics)
  - "H-events" (which have m+1 body kinematics)

Negative weights

- We generate events for the two terms between the square brackets (S- and H-events) separately
- There is no guarantee that these contributions are separately positive (even though predictions for infra-red safe observables should always be positive!)
- Therefore, when we do event unweighting we can only unweight the events **up to a sign**. These signs should be taken into account when doing a physics analysis (i.e. making plots etc.)
- The events are only physical when they are showered.

Example : ttbar production



POWHEG

Nason (2004)

- One could try to get NLO accuracy by replacing B with the NLO rate (integrated over the extra phase-space)

- In order to avoid double counting, one should replace the definition of the Sudakov form factor

- Its structure is identical an ordinary shower, with normalization rescaled by a global K-factor and a different Sudakov for the first emission: no negative weights are involved.

Plan

- NLO Introduction
- Loop Computation
- Dealing with Singularities
- Matching@NLO
- Live Demo

What we have learned

- That all observable of an NLO computation are not NLO accurate.
- How to evaluate the loop
- NLO computation done with counter-events
  - ➡ can be a bin miss-match
- NLO+PS generation allow event generation
  - ➡ Events Physical only after the Parton-Shower.
  - ➡ The Events should be generated for a given shower (in MC@NLO)