Stefano Frixione

# LHC phenomenology: from postdictions to predictions, or the quest for accuracy

CERN, 4/5/2011

# Postdictive approach:

Calibrate (i.e. tune) your predictions on data (control sample)

## Postdictive approach:

Calibrate (i.e. tune) your predictions on data (control sample)

Apply calibrated predictions to the problem of interest (e.g. a search). Ideally, do so only for backgrounds. Predictions for signal should be used only to define the cuts (i.e., the search strategy)

# Postdictive approach:

Calibrate (i.e. tune) your predictions on data (control sample)

Apply calibrated predictions to the problem of interest (e.g. a search). Ideally, do so only for backgrounds. Predictions for signal should be used only to define the cuts (i.e., the search strategy)

When the control and problem regions do not coincide, a leap of faith (i.e. one trusts one's MC) is required

# Predictive approach:

Accept the risk of being wrong, and reliably so

## Predictive approach:

Accept the risk of being wrong, and reliably so

Or: a reliable estimate of theoretical uncertainties is as important as the central value for a predictive approach to be meaningful

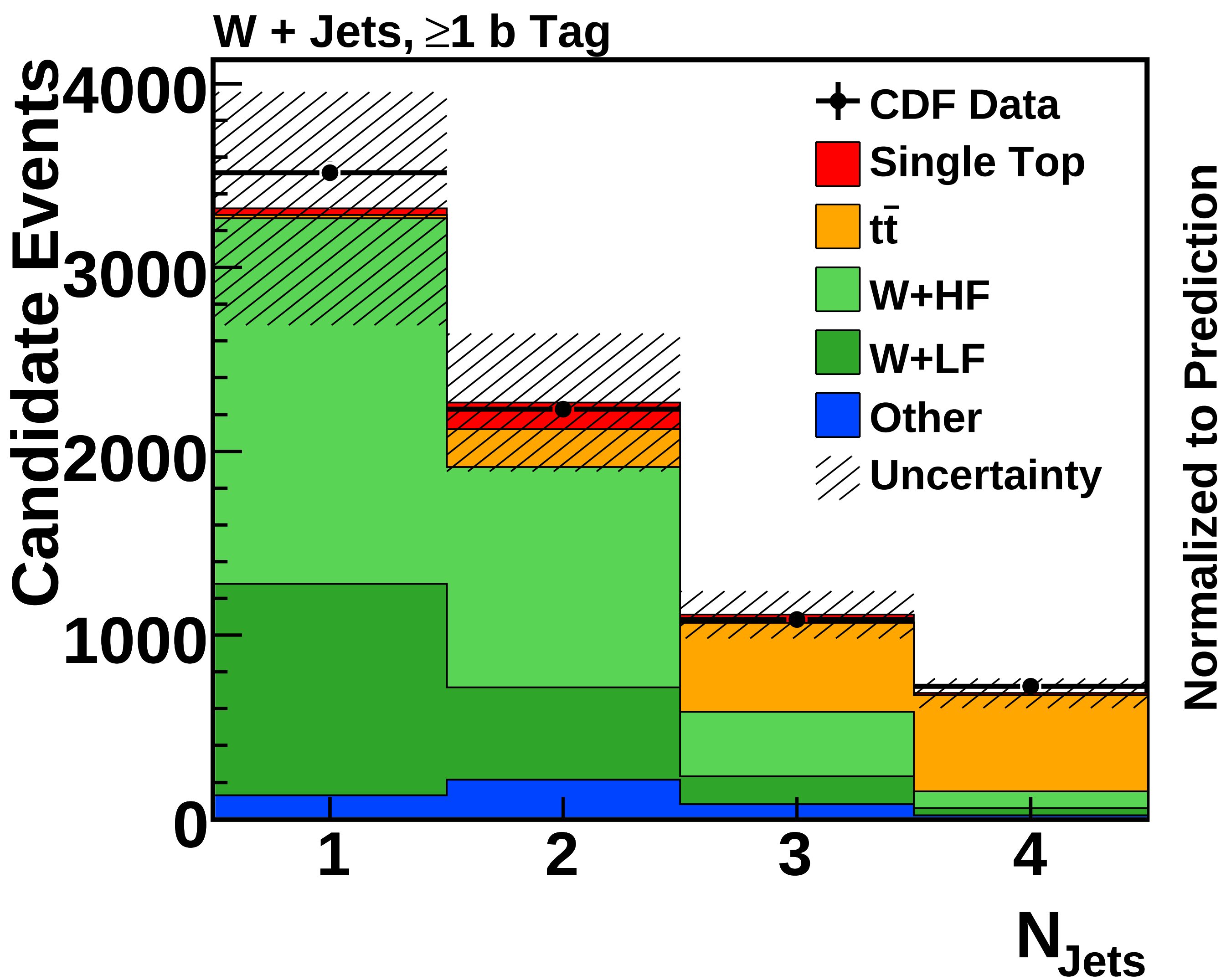
# Predictive approach:

Accept the risk of being wrong, and reliably so

Or: a reliable estimate of theoretical uncertainties is as important as the central value for a predictive approach to be meaningful

Namely: predict and compare to data. If discrepancy is larger than uncertainties, discovery. Else, another SM success

# Single top at CDF (1004.1181)




Blind predictive approach hopeless: uncertainty much larger than signal

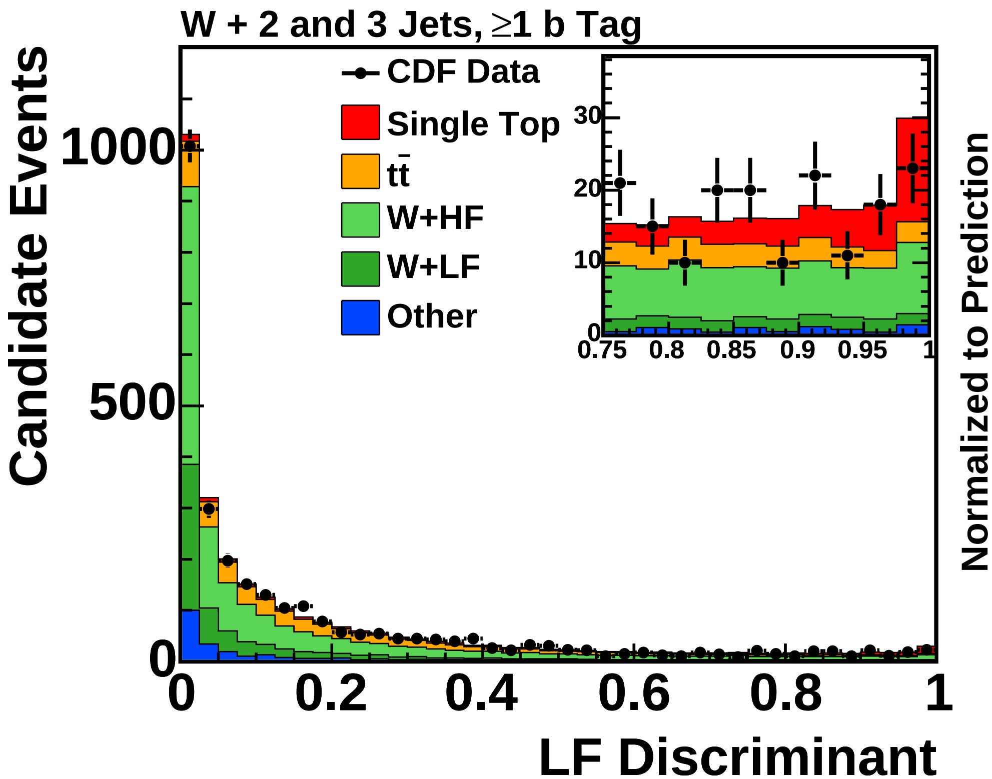
But: real life is more complicated

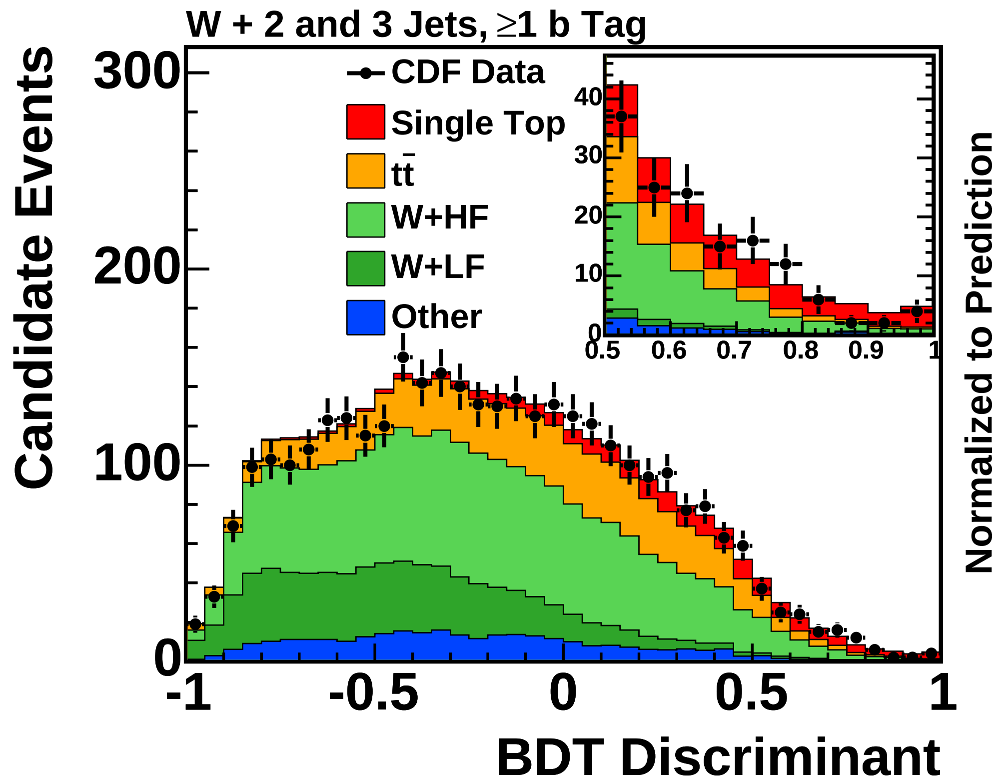
One always uses a mixture of postdictions and predictions

There is nothing wrong in this. The problem is that of knowing when to stop, i.e. understand what is tunable and what is not

That “when” is dictated by the accuracy of theoretical computations

# Discovery




The signal region is at Discriminant $\longrightarrow 1$, as one sees from the templates of single-top production (not shown here)

Single-top discovery at the Tevatron is a paradigm for LHC analyses

- An extremely large number of background processes, which swamp the small-cross-section signal
- The necessity of using fully-exclusive theoretical results, that can go through detector simulation
- Lots of postdictions

So what are the lessons to be learned?

$\#$ 1: There are certain analyses whose discovery potential is strongly biased by theory, since the capability of simulating the signal well is essential

# 1: There are certain analyses whose discovery potential is strongly biased by theory, since the capability of simulating the signal well is essential

# 2: Lack of predictivity means the possibility of overstretching predictions, which implies that one does overstretch predictions, which means a leap of faith without noticing

# 1: There are certain analyses whose discovery potential is strongly biased by theory, since the capability of simulating the signal well is essential

# 2: Lack of predictivity means the possibility of overstretching predictions, which implies that one does overstretch predictions, which means a leap of faith without noticing

# 3: “Exclusive” parton-level predictions are mandatory. Experimenters will always prefer to have them attached to event generators (parton showers)

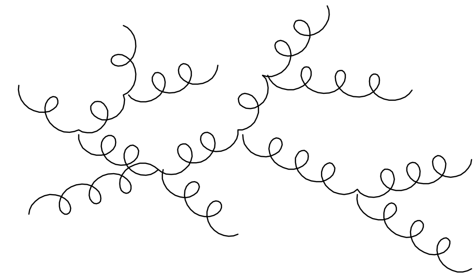
Fine, point taken: accurate, fully-differential, realistic (i.e., do not use standalone Pythia/Herwig for processes other than $2 \to 2$), and hadron-level predictions may play *a very important role*

Does this mean NLO (or beyond) at the perturbative level?

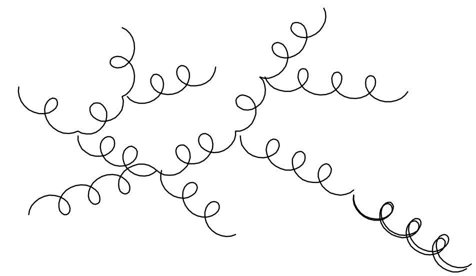
By NLO, one should really mean the complete stuff $\longrightarrow$

# Anatomy of NLO computations

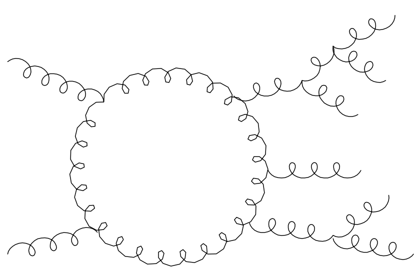
$$\frac{d\sigma}{dO} = \int d\phi_{n+1} \Big( \mathcal{M}^{(r)}(\phi_{n+1})\delta(O - O_{n+1}(\phi_{n+1}))$$

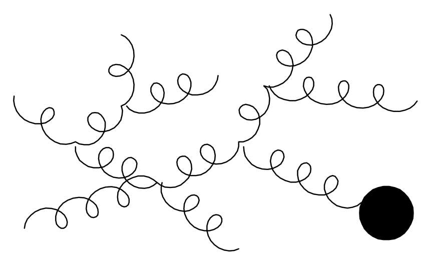


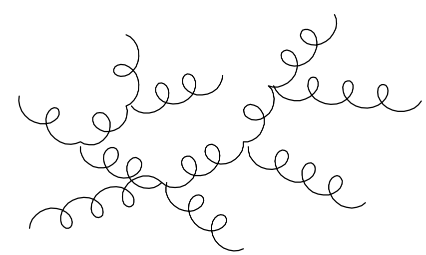
$$-\mathcal{M}^{(c.t.)}(\mathcal{P}\phi_{n+1})\delta(O - O_n(\mathcal{P}\phi_{n+1}))\Big)$$



$$+ \int d\phi_n \Big( \mathcal{M}^{(v)}(\phi_n) + \mathcal{M}^{(rem)}(\phi_n) + \mathcal{M}^{(b)}(\phi_n) \Big) \delta(O - O_n(\phi_n))$$







Fine, point taken: accurate, fully-differential, realistic (i.e., do not use standalone Pythia/Herwig for processes other than $2 \to 2$), and hadron-level predictions may play *a very important role*

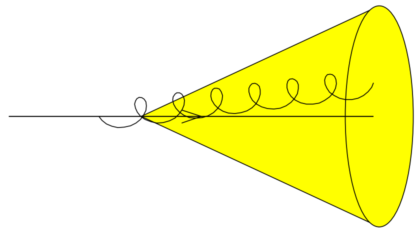
Does this mean NLO (or beyond) at the perturbative level?

Although my final answer is a clear yes, I must first stress that many of the motivations usually given are simply not good enough

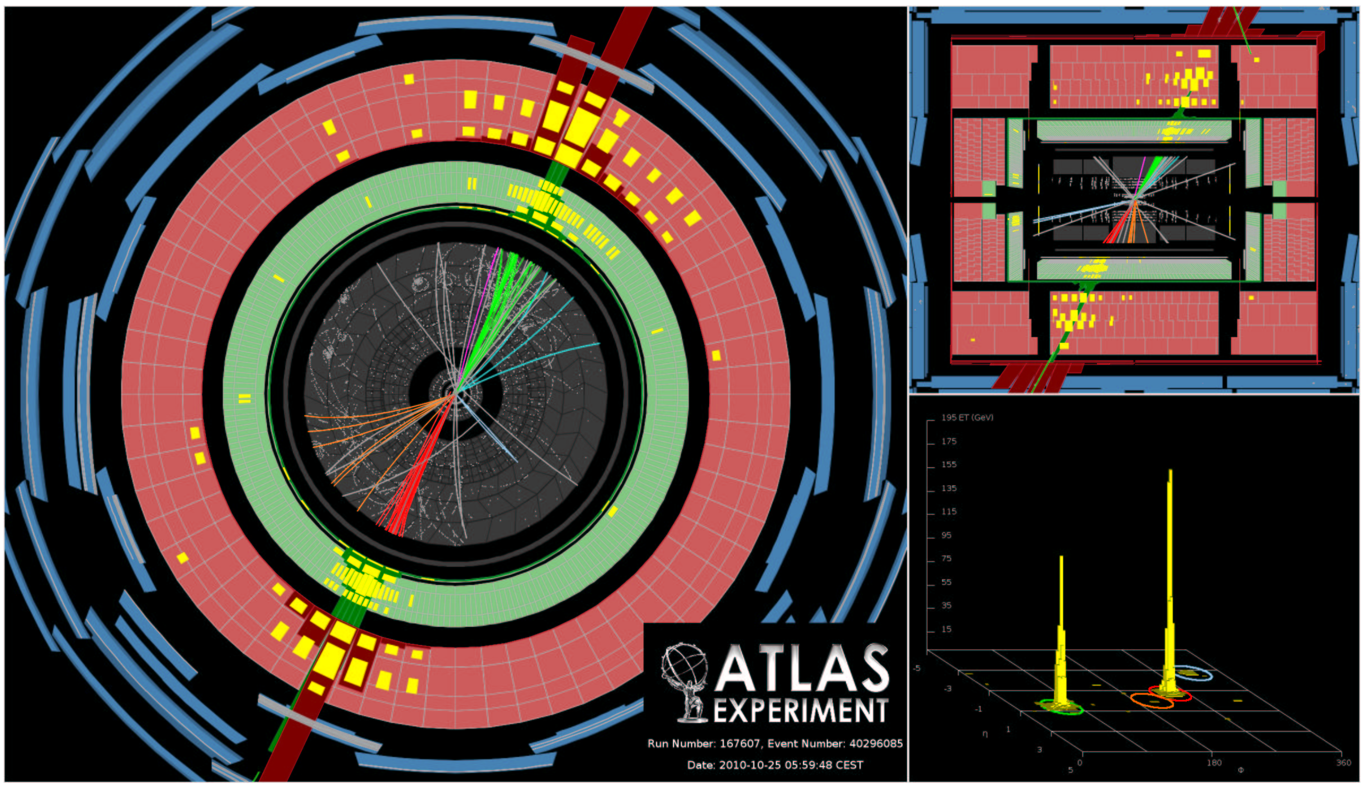
# A: Better description of jet structure

(...in the sense that doing worse is difficult, perhaps?)

A jet in an NLO computation:



Actual jets:

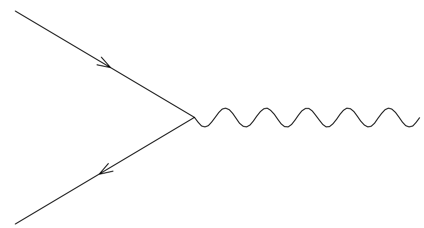



- One or two partons vs $\mathcal{O}(50)$ hadrons

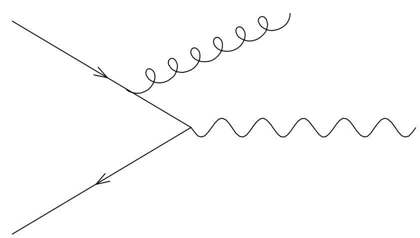
# B: More combinations of initial-state partons

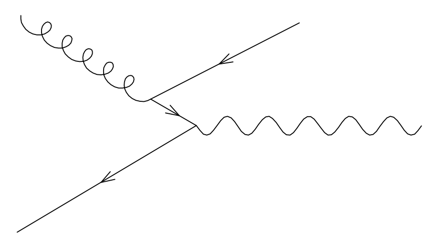
True, but misses the point. Consider $Z$ production:

Born:

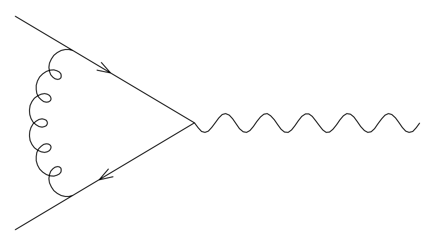


Real corrections:





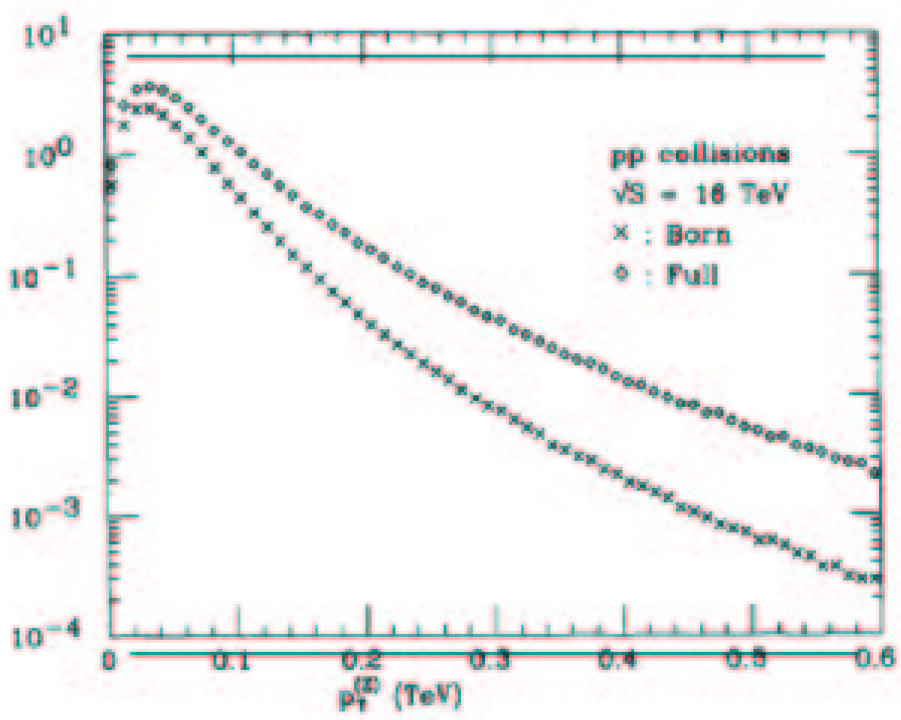
Virtual corrections:



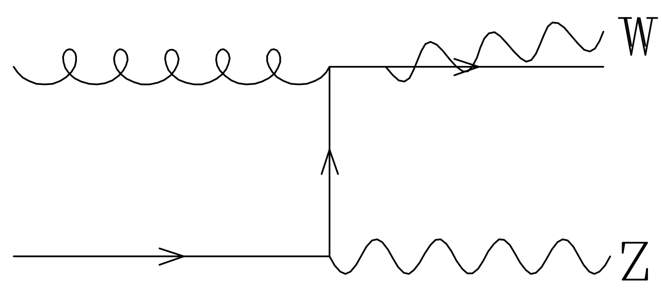
- This is a feature of tree-level corrections

# C: NLO effects on distributions

Again, misses the point. Consider $p_T^{(Z)}$ in $WZ$ production:




A $K$ factor of about 6 at $p_T^{(Z)} = 600$ GeV. But:




is what dominates (double Sudakov log)

- Again, a feature of tree-level corrections

It's tempting to conclude that all we need are trees and more trees

- Incredibly easier to compute than full NLO
- Easier to match to event generators
- Combinations of processes with different multiplicities (e.g. $W + 0$, $1$, $\ldots n$ jets) established
- Understood and routinely used by experiments

It's tempting to conclude that all we need are trees and more trees

- Incredibly easier to compute than full NLO
- Easier to match to event generators
- Combinations of processes with different multiplicities (e.g. $W + 0$, $1$, $\ldots n$ jets) established
- Understood and routinely used by experiments

But:

- They diverge! Hence, extra uncertainties due to cutoffs
- Very large scale dependence
- Predictions for rates unreliable

Given that trees beyond LO do contribute to NLO, it is clear that the true motivations for performing NLO computations are those that tell NLO and tree-level-only results apart:

- An effective way to reduce impact of postdictions
- Absence of control samples – necessity of trusting MC's
- To build confidence on predictive capabilities
- Predictions for rates are credible
- Scale dependence is a meaningful assessment of uncertainties
- Extracting parameters $(\alpha_S,\ m_H, \ldots)$ from measurements

Given that trees beyond LO do contribute to NLO, it is clear that the true motivations for performing NLO computations are those that tell NLO and tree-level-only results apart:

- ...
- ...

$\}$ Different things, having in common the reduction of theory uncertainties. This may be crucial, irrelevant, or anything in between, for a given type of physics

The advantages of precise results must over-compensate the steep technical difficulties one faces when performing NLO computations. Whether this is the case is debatable, but there is one tautological way out:

Difficulties are irrelevant, since NLO calculations will be carried out by a computer

Difficulties are irrelevant, since NLO calculations will be carried out by a computer

If one is strict about this, it means: FULL AUTOMATION – no action other than giving inputs is required. The choice of process is just one of the inputs

[An added (and very significant) benefit of automation. New results, being constructed with the same "basis" of a small number of simple components, are correct almost by definition]

For QCD corrections to SM processes, this has now been achieved

Two (independent) ingredients:

- One-loop matrix elements

  Automated in 1103.0621 (MadLoop)

  Hirschi, Frederix, Frixione, Garzelli, Maltoni, Pittau

  Based on OPP procedure/CutTools

Two (independent) ingredients:

- One-loop matrix elements

  Automated in 1103.0621 (MadLoop)

  Hirschi, Frederix, Frixione, Garzelli, Maltoni, Pittau

  Based on OPP procedure/CutTools

- All the rest (subtraction, integration,...)

  Automated in 0908.4272 (MadFKS)

  Frederix, Frixione, Maltoni, Stelzer

  Based on FKS subtraction

Two (independent) ingredients:

- One-loop matrix elements

  Automated in 1103.0621 (MadLoop)

  Hirschi, Frederix, Frixione, Garzelli, Maltoni, Pittau

  Based on OPP procedure/CutTools

- All the rest (subtraction, integration,...)

  Automated in 0908.4272 (MadFKS)

  Frederix, Frixione, Maltoni, Stelzer

  Based on FKS subtraction

The whole structure is embedded into MadGraph, which is used to compute the underlying trees and gives several other components

# MadFKS + MadLoop (LHC 7 TeV)

From 1103.0621

Key features:

- Diversity of processes
- Reasonably smooth distributions with the same statistics

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

Two weeks of running on $\sim 150$ cores

Automated approaches have attracted a lot of attention lately

The subtraction of real-emission matrix elements has been achieved independently by several groups:

- HELAC (Czakon, Papadopoulos, Worek 0905.0883)
- MadDipole (Frederix, Gehrmann, Greiner 1004.2905, 0808.2128)
- SHERPA (massless only) (Gleisberg, Krauss 0709.2881)
- Other less systematic attempts (Seymour, Tevlin 0803.2231; Hasegawa, Moch and Uwer 0911.4371)

All these use the dipole subtraction method, and are thus significantly different from MadFKS

The extent to which they are automated differ. So far, no systematic comparison exists among the various codes

Automated approaches have attracted a lot of attention lately

For the calculation of one-loop matrix elements, several methods are now established:

- Generalized Unitarity (Bern, Dixon, Dunbar, Kosower hep-ph/9403226 + ...; Ellis, Giele, Kunszt 0708.2398, +Melnikov 0806.3467)
- Integrand Reduction (Ossola, Papadopoulos, Pittau hep-ph/0609007; del Aguila, Pittau hep-ph/0404120; Mastrolia, Ossola, Reiter, Tramontano 1006.0710)
- Tensor Reduction (Passarino, Veltman 1979; Denner, Dittmaier hep-ph/0509141; Binoth, Guillet, Heinrich, Pilon, Reiter 0810.0992 (GOLEM))

...and have been put to use $\longrightarrow$

- GU $\longleftarrow$ BlackHat (Berger, Bern, Dixon, Febres Cordero, Forde, Gleisberg, Ita, Kosower, Maitre 1009.2338 + ...)
- GU $\longleftarrow$ Rocket (Ellis, Giele, Kunszt, Melnikov, Zanderighi 0810.2762 + ...)
- IR $\longleftarrow$ OPP/HELAC-1Loop (Bevilacqua, Czakon, Papadopoulos, Pittau, Worek, 0907.4723 + ...)

Tensor Reduction is not really suited to automation

So far, GU applied mostly to large-multiplicity, massless final states (e.g. W+4 jets by BlackHat), IR to lower-multiplicity, massive final states (e.g. HELAC $t\bar{t}b\bar{b}$)

These are truly impressive results! Some compromises with full automation, which was not the authors' first priority

# Subtraction

Consider a $2 \to n$ pure-gluon process. There are

- $(n^2 + 3n)/2$ collinear singularities (two-body correlations)
- $n$ soft singularities (three-body correlations)

Their systematic subtractions for any $n$ in a process-independent manner is a solved problem

- Residue (Frixione, Kunszt, Signer, hep-ph/9512328 + ...)
- Dipole (Catani, Seymour, hep-ph/9605323 + ...)
- Antenna (Kosower, hep-ph/9720213)

Dipole has been the most popular for NLO computations based on traditional ("by hand") techniques. Perhaps as a consequence of this, it has been adopted by all groups pursuing automation, as seen before

Except us. Why?

# FKS #1: Simplify the problem

Find parton pairs $(i, j)$ that can give collinear singularities. Then:

$$\mathcal{M}^{(r)} = \sum_{ij} \mathcal{M}_{ij}^{(r)} \qquad \mathcal{M}_{ij}^{(r)} = \mathcal{S}_{ij} \mathcal{M}^{(r)}$$

with

$$\begin{aligned}
&\sum_{ij} \mathcal{S}_{ij} = 1 && \\
&\sum_{j} \mathcal{S}_{ij} \longrightarrow 1 && k_i \longrightarrow 0 \\
&\mathcal{S}_{ij} \longrightarrow 1 && k_i \parallel k_j \\
&\mathcal{S}_{ij} \longrightarrow 0 && \text{all other singularities}
\end{aligned}$$

- $\mathcal{M}_{ij}^{(r)}$ has one soft and one collinear singularity at most
- Soft singularities are "split" into underlying collinear structures
- The $\mathcal{M}_{ij}^{(r)}$'s are independent from each other

# FKS #2: Subtract

For a given $\mathcal{M}_{ij}^{(r)}$, the choice of variables associated with singularities is natural and essentially unique

$$\mathcal{M}_{ij}^{(r)} d\phi_{n+1} \longrightarrow \left(\frac{1}{E_i}\right)_+ \left(\frac{1}{1-\cos\theta_{ij}}\right)_+ E_i^2 \left(1-\cos\theta_{ij}\right) \mathcal{M}_{ij}^{(r)} \frac{d\phi_{n+1}}{E_i}$$

- Plus distributions understand the projections $\mathcal{P}\phi_{n+1}$. FKS defines subtraction terms *exactly* on shell, and thus eliminates the problem of recoil altogether (NO approximation is required)
- Soft and collinear counterterms can be defined so as they have the *same* kinematics $\Longrightarrow$ the subtraction term is unique

By-product: important sampling (thanks to $E_i$ and $\theta_{ij}$) is straightforward

# FKS $\#3$: Exploit symmetries

Collinear structure and definition of projections imply that the total number of subtraction terms scales as $n^2$. However for any observable $O$

$$O(\mathcal{M}_{ij}^{(r)}) = O(\mathcal{M}_{kl}^{(r)})$$

when flavour$_i$=flavour$_k$ and flavour$_j$=flavour$_l$, the key being the independence of $\mathcal{M}_{ij}^{(r)}$ and $\mathcal{M}_{kl}^{(r)}$

$\Longrightarrow$ The majority of contributions can be taken into account simply with an overall symmetry factor

Thus: for a $2 \to n$ gluon process, the total number of subtractions is 3 ($\forall n$). The presence of quarks complicates the counting, but the subtractions are never more than a few

# CS dipoles

- Use soft singularities to organize subtractions $\Longrightarrow$ three-body kernels
- Define subtractions off-shell. The recoil is distributed using a mapping defined by the three partons of each kernel. Hence, each subtraction term has a different kinematics
- Subtract all subtraction terms from $\mathcal{M}^{(r)}$

# CS dipoles

- Use soft singularities to organize subtractions $\Longrightarrow$ three-body kernels
- Define subtractions off-shell. The recoil is distributed using a mapping defined by the three partons of each kernel. Hence, each subtraction term has a different kinematics
- Subtract all subtraction terms from $\mathcal{M}^{(r)}$

## Implications:

- Number of subtraction terms scales as $n^3$. I am not aware of any systematic method to reduce it by exploiting symmetries
- Numerics become intractable without the use of $\alpha$-dependent subtractions (Nagy). One is forced to tune $\alpha$ for large $n$
- Importance sampling must be done dynamically

## Moreover:

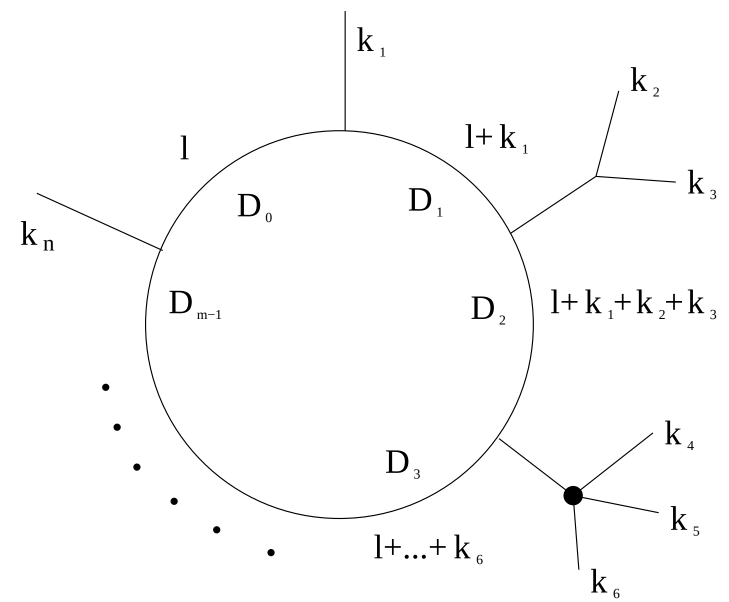
- FKS has a “collinear” structure. It is therefore the method of choice for NLO-parton shower matching formalisms (MC@NLO and POWHEG)
- FKS-derived approach (Czakon, 1005.0274) to NNLO double-real subtraction appears to be general and applicable to hadronic collisions. Other attempts based on antenna. Dipole is not pursued (too many subtractions, plus spurious singularities)

Scaling properties, numerical stability, and matching to showers are sufficiently good reasons for us to prefer FKS to CS dipoles

# One-loop computations

# OPP #1: Remember Passarino-Veltman

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




$$\bar{\ell} = \ell + \tilde{\ell}\,, \qquad \ell\cdot\tilde{\ell} = 0$$

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

# OPP #2: Move to the integra*nd* level

$$\mathcal{C} = \int d^d\bar{\ell}\,\bar{C}(\bar{\ell})\,, \qquad \bar{C}(\bar{\ell}) = \frac{\bar{N}(\bar{\ell})}{\prod_{i=0}^{m-1}\bar{D}_i}$$

Separate 4- and $\epsilon$-dimensional part in the numerator

$$\bar{N}(\bar{\ell}) = N(\ell) + \widetilde{N}(\ell,\tilde{\ell}) \qquad \Longrightarrow \qquad \bar{C}(\bar{\ell}) = \frac{N(\ell)}{\prod_{i=0}^{m-1}\bar{D}_i} + \frac{\widetilde{N}(\ell,\tilde{\ell})}{\prod_{i=0}^{m-1}\bar{D}_i}$$

This defines a cut-constructible-plus-rational part, and a pure-rational part:

$$\begin{aligned}
\mathcal{C} &= \mathcal{C}_{cc+R_1} + R_2 \\
\mathcal{C}_{cc+R_1} &= \int d^d\bar{\ell}\,\frac{N(\ell)}{\prod_{i=0}^{m-1}\bar{D}_i} \\
R_2 &= \int d^d\bar{\ell}\,\frac{\widetilde{N}(\ell,\tilde{\ell})}{\prod_{i=0}^{m-1}\bar{D}_i}
\end{aligned}$$

# OPP #3: Expand the numerator

$$
\begin{aligned}
N(\ell) &= \sum_{0\le i_0<i_1<i_2<i_3}^{m-1} \left[d(i_0i_1i_2i_3) + \hat{d}(\ell; i_0i_1i_2i_3)\right] \prod_{\substack{i=0\\ i\notin\{i_0,i_1,i_2,i_3\}}}^{m-1} D_i \\
&+ \sum_{0\le i_0<i_1<i_2}^{m-1} \left[c(i_0i_1i_2) + \hat{c}(\ell; i_0i_1i_2)\right] \prod_{\substack{i=0\\ i\notin\{i_0,i_1,i_2\}}}^{m-1} D_i \\
&+ \sum_{0\le i_0<i_1}^{m-1} \left[b(i_0i_1) + \hat{b}(\ell; i_0i_1)\right] \prod_{\substack{i=0\\ i\notin\{i_0,i_1\}}}^{m-1} D_i \\
&+ \sum_{0\le i_0}^{m-1} \left[a(i_0) + \hat{a}(\ell; i_0)\right] \prod_{\substack{i=0\\ i\neq i_0}}^{m-1} D_i
\end{aligned}
$$

- $d \ldots a$ are the *same as at the integral* level
- Spurious terms $\hat{d} \ldots \hat{a}$ vanish upon integration
- Mismatch between $D_i$ and $1/\bar{D}_i$ is the origin of $R_1$

# OPP #4: Enter CutTools

The system

$$N(\ell) = f(\ell; d, c, b, a)$$

is solved for $d \ldots a$ by the recursive applications of unitarity-cut-like conditions (4, 3, 2, or 1 denominators are imposed to vanish):

$$D_{i_0}(\ell^{\pm}) = D_{i_1}(\ell^{\pm}) = D_{i_2}(\ell^{\pm}) = D_{i_3}(\ell^{\pm}) = 0$$

# OPP #4: Enter CutTools

The system

$$N(\ell) = f(\ell; d, c, b, a)$$

is solved for $d \dots a$ by the recursive applications of unitarity-cut-like conditions (4, 3, 2, or 1 denominators are imposed to vanish):

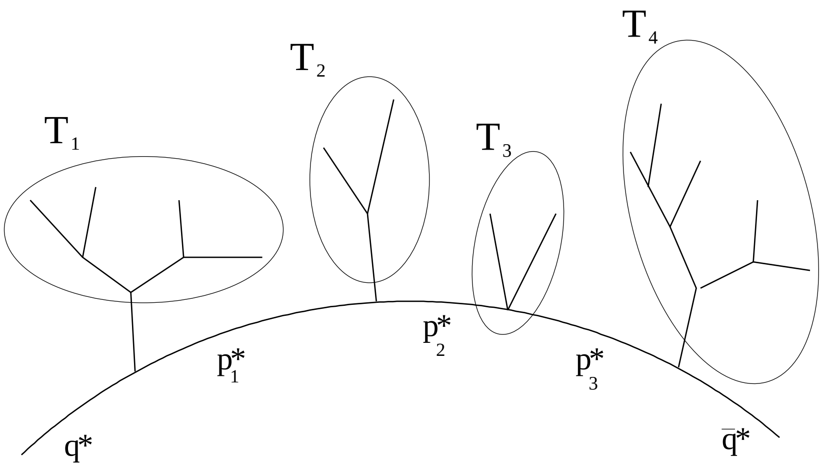
$$D_{i_0}(\ell^{\pm}) = D_{i_1}(\ell^{\pm}) = D_{i_2}(\ell^{\pm}) = D_{i_3}(\ell^{\pm}) = 0$$

Bottom line:

- CutTools solves a system of linear equations. One can see this (owing to the values of $\ell$) as unitarity-cutting, but it is not essential
- Given the function $N(\ell)$, momenta, and masses, CutTools returns the cut-constructible part and $R_1$, using pre-tabulated results for scalar one-loop integrals
- $R_2$ must be computed independently

# MadLoop

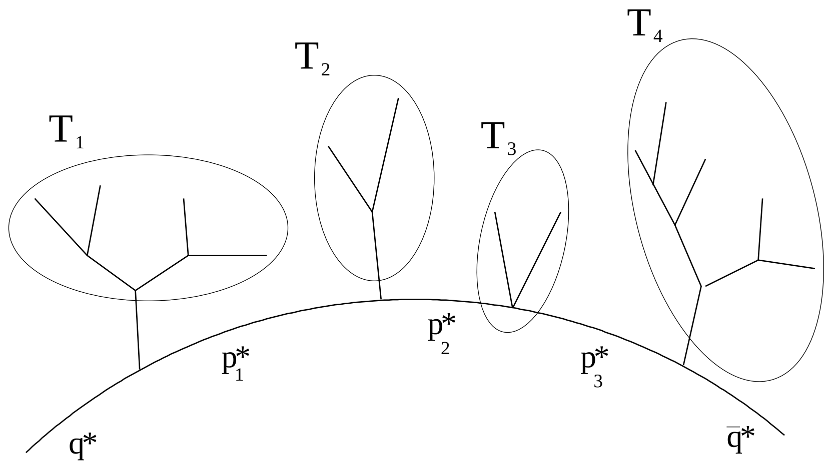
The basic idea: a loop diagram becomes a tree diagram as soon as one cuts one of the loop propagators




Nothing to do with unitarity cuts. We call them L-cuts

# MadLoop

The basic idea: a loop diagram becomes a tree diagram as soon as one cuts one of the loop propagators




Nothing to do with unitarity cuts. We call them L-cuts

- Exploit MadGraph capabilities with trees to compute L-cut diagrams
- Just need to "tag" particles flowing in the loop
- L-cut diagrams have different multiplicities than loop ones. Redundancy can be avoided basically by a topology analysis.
  We call this operation filtering

# MadLoop

Process and kinematic configurations are given in input (e.g. by MadFKS)

# MadLoop

Process and kinematic configurations are given in input (e.g. by MadFKS)

- Generates L-cut diagrams and filter them

# MadLoop

Process and kinematic configurations are given in input (e.g. by MadFKS)

- Generates L-cut diagrams and filter them
- Computes the one-loop integrands, and gives the function $N(\ell)$ to CutTools for each of them

# MadLoop

Process and kinematic configurations are given in input (e.g. by MadFKS)

- Generates L-cut diagrams and filter them
- Computes the one-loop integrands, and gives the function $N(\ell)$ to CutTools for each of them
- Performs the computation of the $R_2$ part. This corresponds to tree-level diagrams, in which one of the vertices is an $\mathcal{O}(\alpha_S)$ $R_2$ one, which is theory-dependent but process-independent

# MadLoop

Process and kinematic configurations are given in input (e.g. by MadFKS)

- Generates L-cut diagrams and filter them
- Computes the one-loop integrands, and gives the function $N(\ell)$ to CutTools for each of them
- Performs the computation of the $R_2$ part. This corresponds to tree-level diagrams, in which one of the vertices is an $\mathcal{O}(\alpha_S)$ $R_2$ one, which is theory-dependent but process-independent
- Performs UV renormalization

# MadLoop

Process and kinematic configurations are given in input (e.g. by MadFKS)

- Generates L-cut diagrams and filter them
- Computes the one-loop integrands, and gives the function $N(\ell)$ to CutTools for each of them
- Performs the computation of the $R_2$ part. This corresponds to tree-level diagrams, in which one of the vertices is an $\mathcal{O}(\alpha_S)$ $R_2$ one, which is theory-dependent but process-independent
- Performs UV renormalization

$R_2$ is due to a set of finite counterterms. Hence, in any theory the automation of UV renormalization gives one the automation of $R_2$ for free

Advantage of $R_2$: it's four-dimensional. Disadvantage: it's gauge dependent

The computation of the rational part is never straightforward.

Alternative approaches: $d$-dimensional unitarity, bootstrap

# Matching to showers

# MC@NLO

The generating functional is:

$$\mathcal{F}_{\text{MC@NLO}} = \mathcal{F}^{(2\to n+1)}\, d\sigma^{(\mathbb{H})}_{\text{MC@NLO}} + \mathcal{F}^{(2\to n)}\, d\sigma^{(\mathbb{S})}_{\text{MC@NLO}}$$

with the two *finite* short-distance cross sections

$$d\sigma^{(\mathbb{H})}_{\text{MC@NLO}} = d\phi_{n+1}\left(\mathcal{M}^{(r)}(\phi_{n+1}) - \mathcal{M}^{(\text{MC})}(\phi_{n+1})\right)$$

$$d\sigma^{(\mathbb{S})}_{\text{MC@NLO}} = \int_{+1} d\phi_{n+1}\left(\mathcal{M}^{(b+v+rem)}(\phi_n) - \mathcal{M}^{(c.t.)}(\phi_{n+1}) + \mathcal{M}^{(\text{MC})}(\phi_{n+1})\right)$$

- Black terms: pure NLO, same as before
- Red terms: MC subtraction terms, with a factorized form

$$\mathcal{M}^{(\text{MC})} = \mathcal{K}^{(\text{MC})}\mathcal{M}^{(b)}$$

Automation of MC@NLO $\equiv$ MadFKS+MadLoop+automation of $\mathcal{K}^{(\text{MC})}$

$\equiv$ aMC@NLO (Frederix, Frixione, Torrielli)

# MC@NLO $\longrightarrow$ aMC@NLO

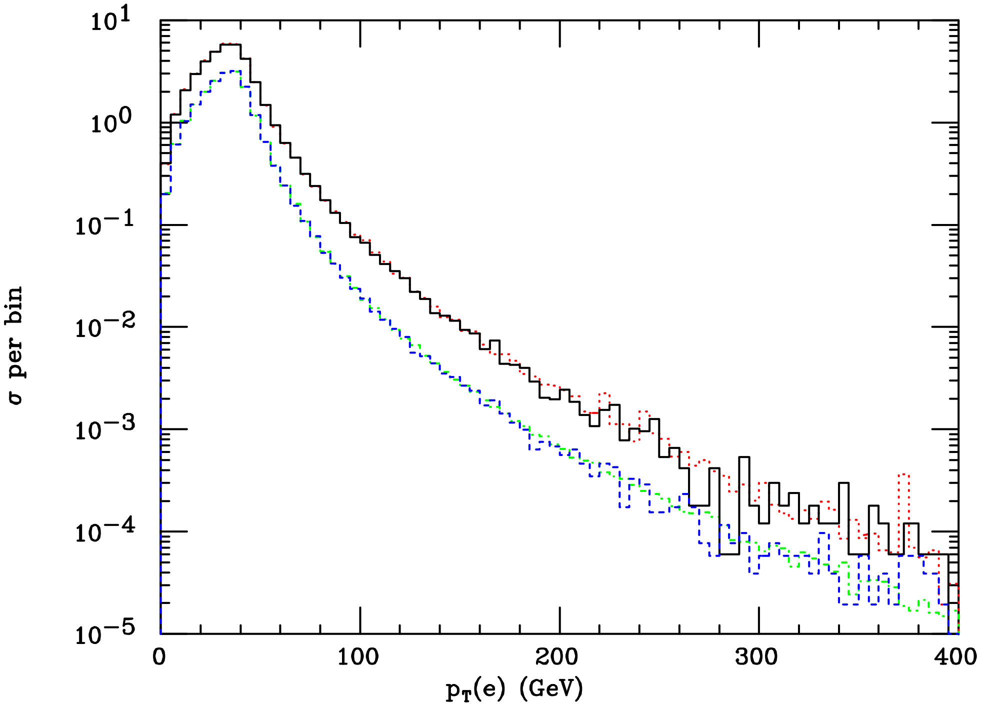
Each Monte Carlo corresponds to a finite set of $\mathcal{K}^{(\mathrm{MC})}$, which need be computed analytically

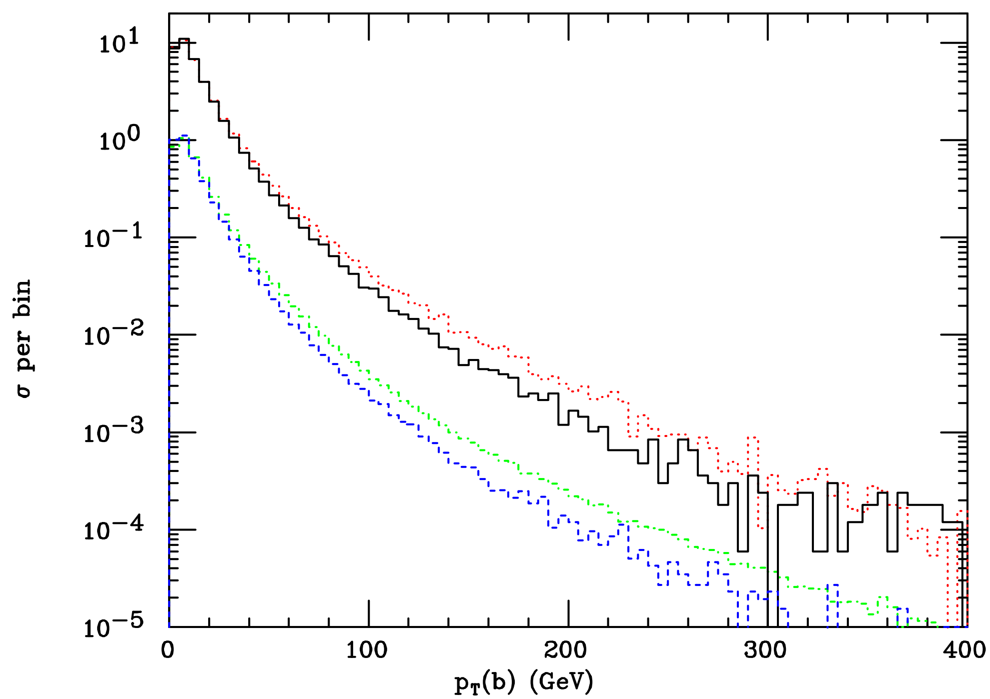
- Done for Herwig6 (Frixione, Webber, hep-ph/0204244; Frixione, Laenen, Motylinski, Webber, hep-ph/0512250), Pythia6 (Torrielli, Frixione, 1002.4293), Herwig++ (Frixione, Stoeckli, Torrielli, Webber, 1010.0568)
- The automatic implementation of $\mathcal{K}^{(\mathrm{MC})}$ also relies on finding pairs associated with collinear singularities – inherited from MadFKS
- Need to figure out colour connections

We have quickly obtained results for processes never matched to showers before, and in general more complicated than those dealt with "by hand"

$\longrightarrow$

# $(W \to) e\nu b\bar{b}$ with aMC@NLO

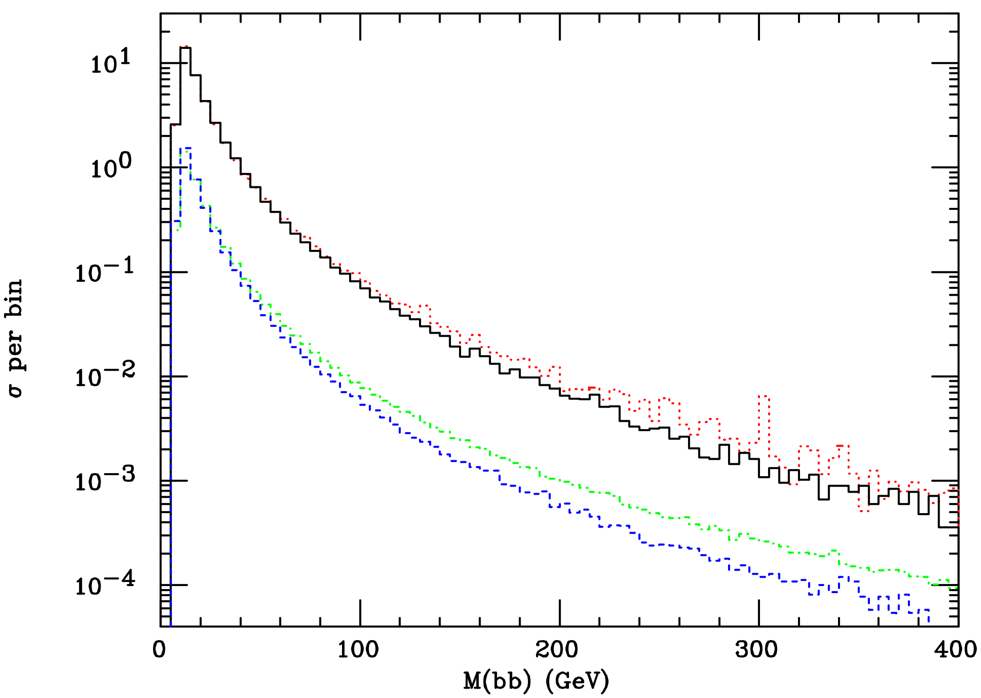


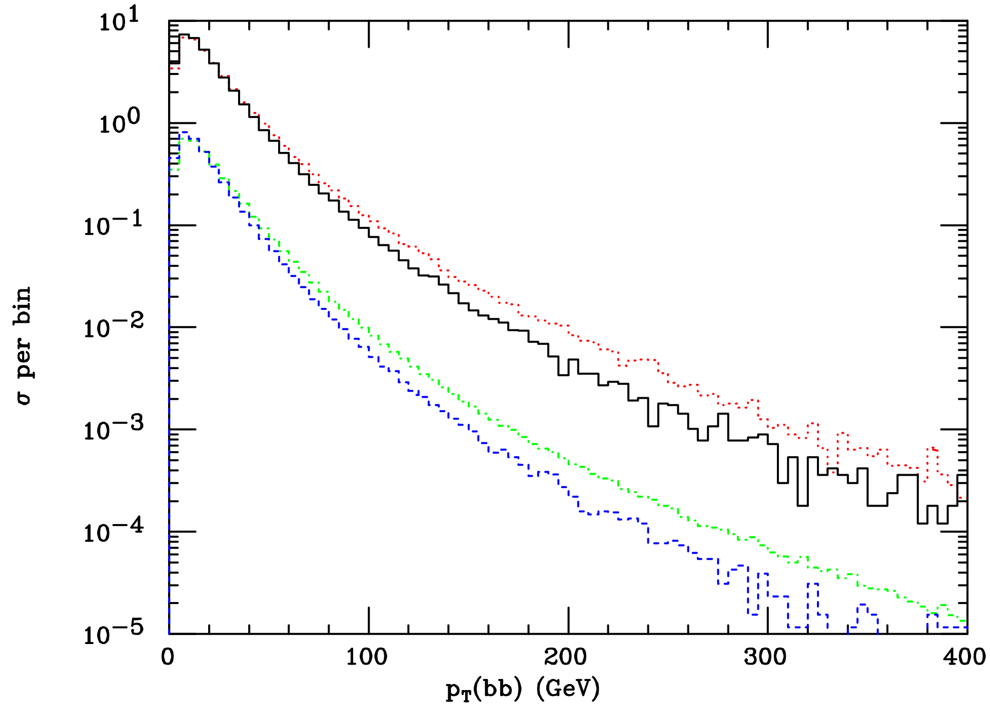




Solid: aMC@NLO. Dashed: aMC@LO/5 Dotted: NLO. Dotdashed: LO/5

Left: electron $p_T$. Right: $b$ quark $p_T$ (LO rescaled)

# $(W \to) e\nu b\bar{b}$ with aMC@NLO

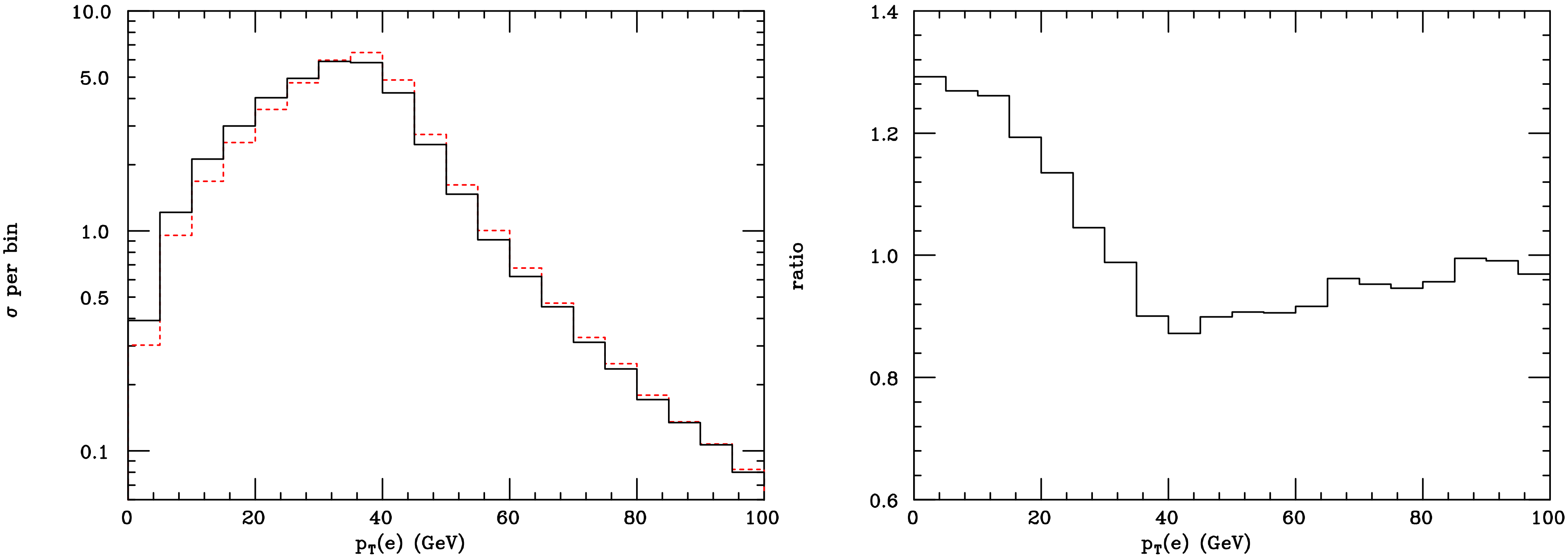



Solid: aMC@NLO. Dashed: aMC@LO$/5$ Dotted: NLO. Dotdashed: LO$/5$

Left: $b\bar{b}$ invariant mass (LO rescaled). Right: $b\bar{b}$ $p_T$ (LO rescaled)

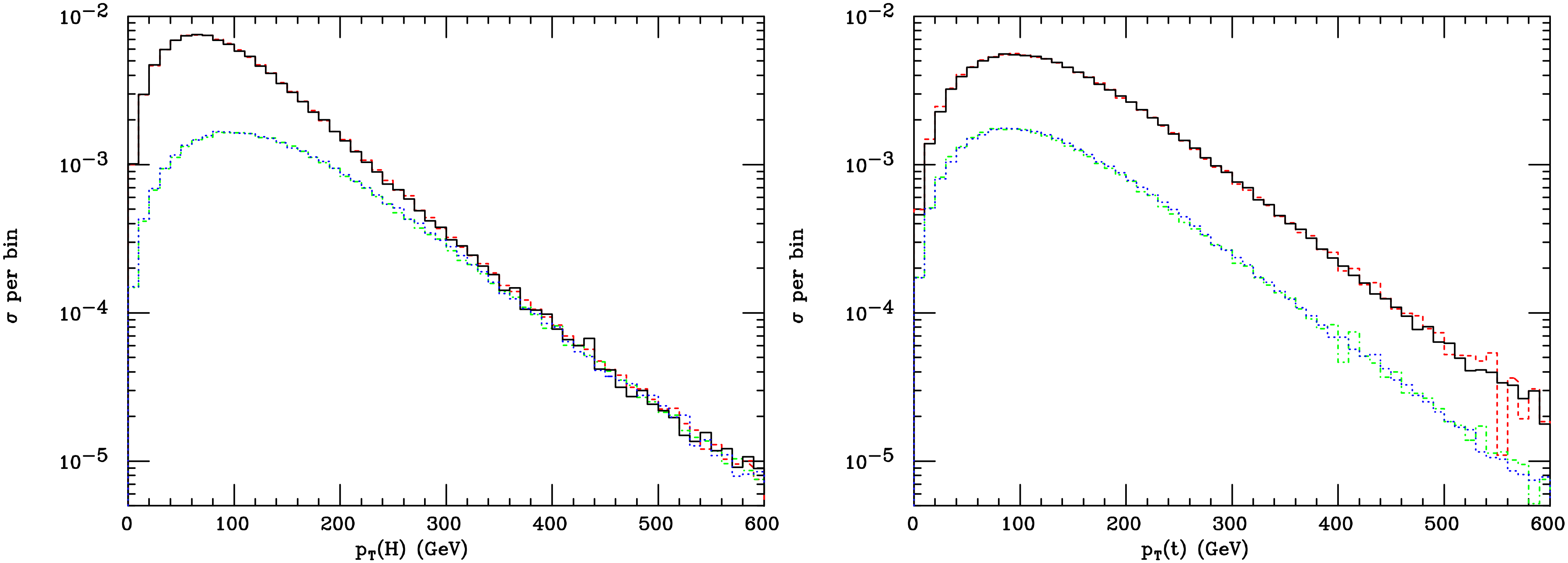
# $(W \to) e\nu b\bar{b}$ with aMC@NLO




Solid: with spin correlations. Dashed: without spin correlations

Left: electron $p_T$. Right: ratio black/red

# $Ht\bar{t}$ and $At\bar{t}$ with aMC@NLO



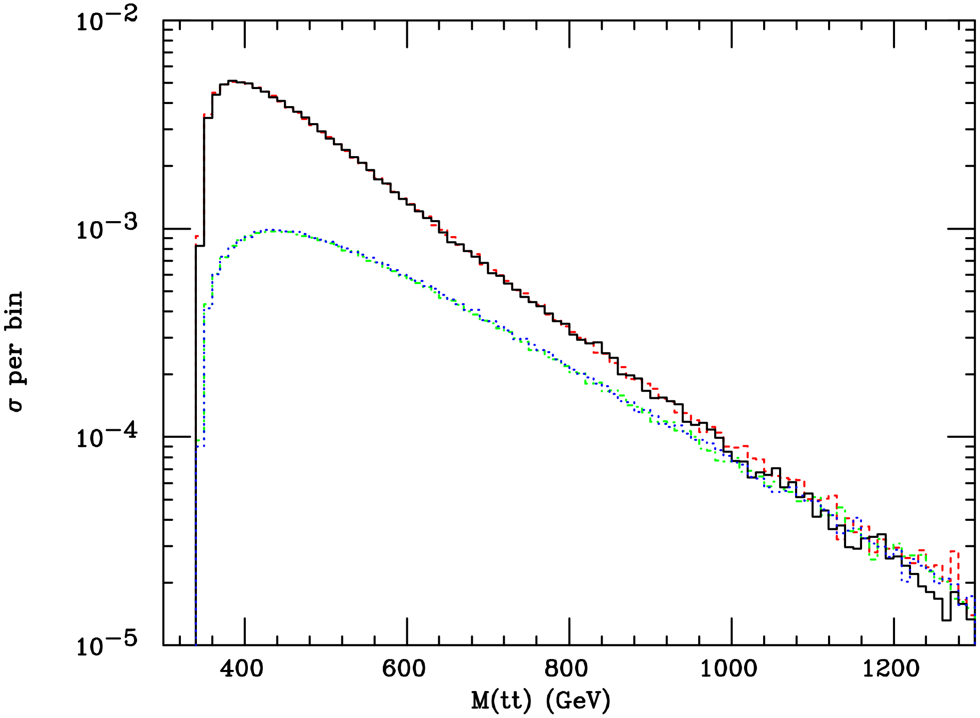

Solid: aMC@NLO scalar. Dashed: aMC@NLO pseudoscalar

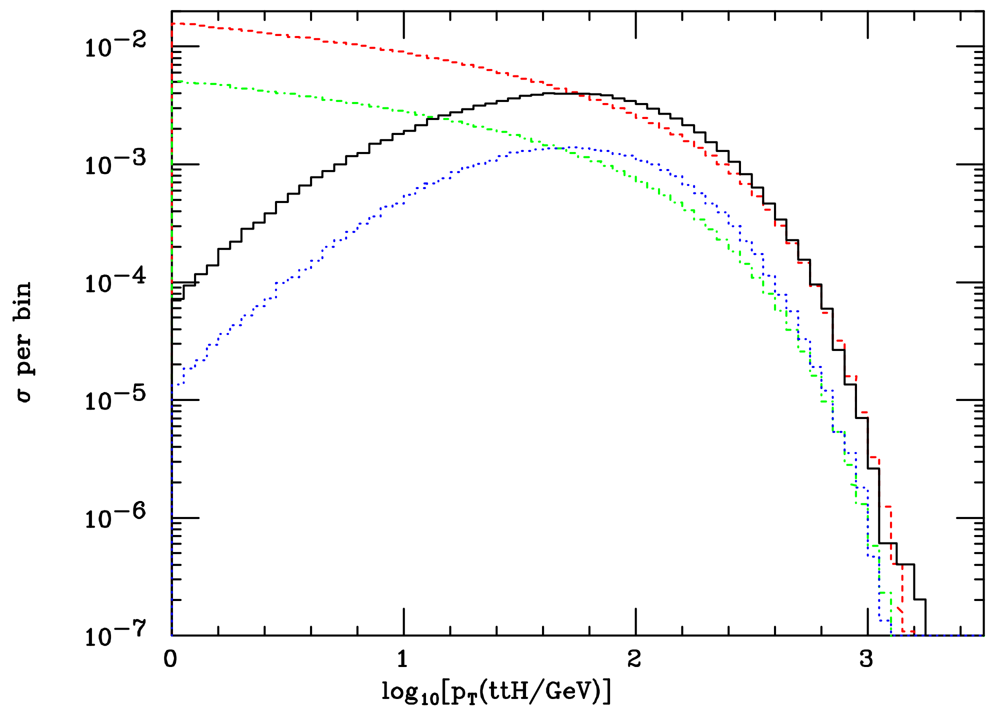
Dotted: NLO scalar. Dotdashed: NLO pseudoscalar

Left: Higgs $p_T$. Right: top $p_T$. $m_H = m_A = 120$ GeV

Frederix, Frixione, Hirschi, Maltoni, Pittau, Torrielli, 1104.5613

# $Ht\bar{t}$ and $At\bar{t}$ with aMC@NLO



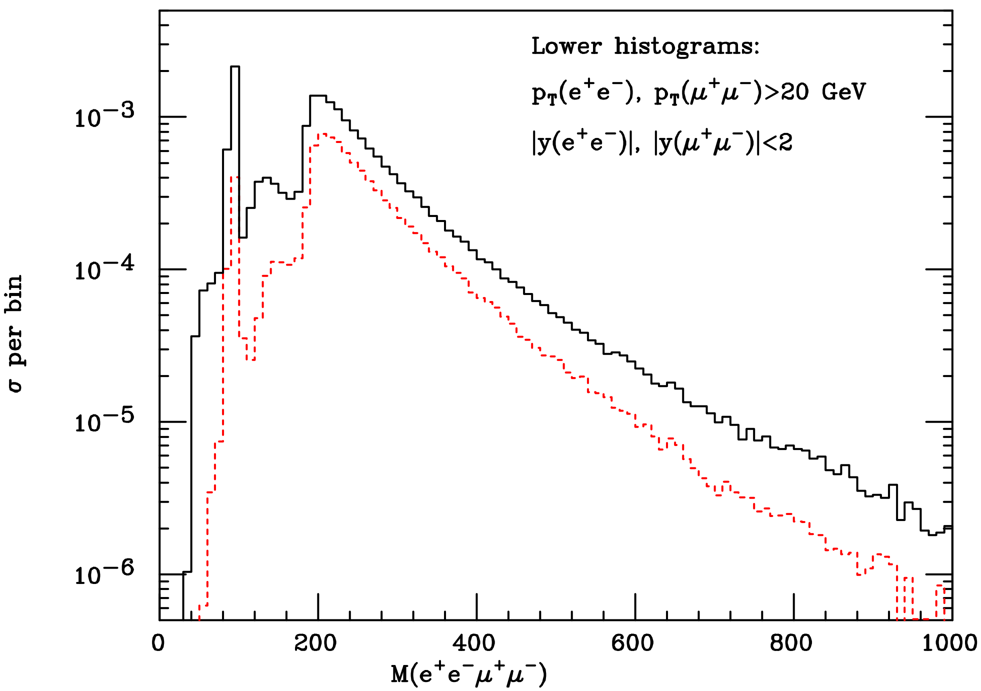

Solid: aMC@NLO scalar. Dashed: aMC@NLO pseudoscalar

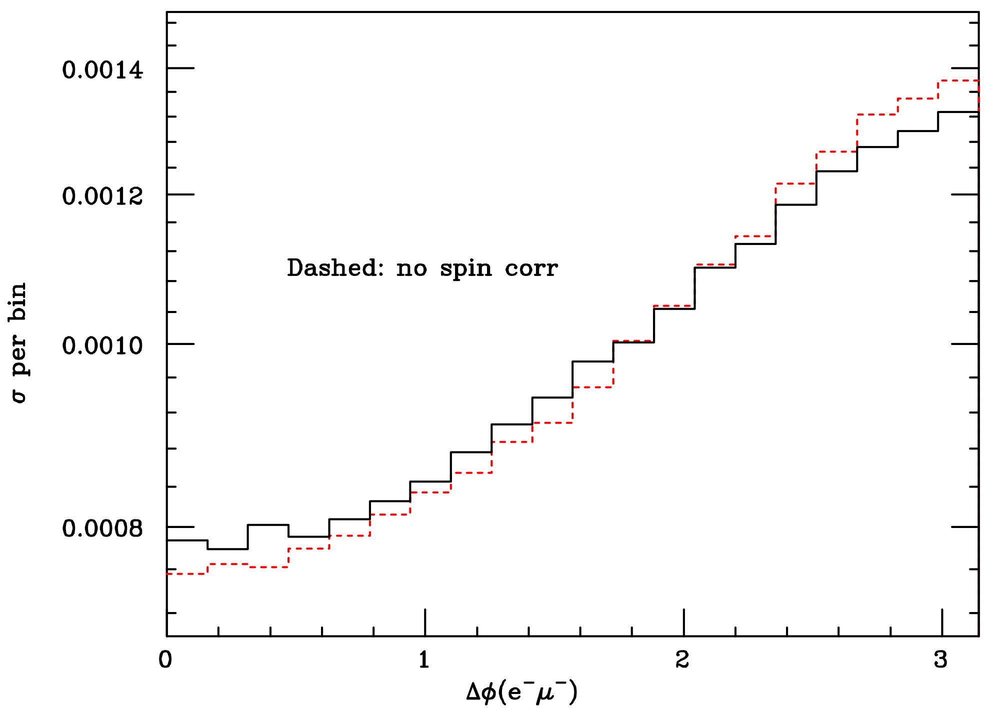
Dotted: NLO scalar. Dotdashed: NLO pseudoscalar

Left: $t\bar{t}$ invariant mass. Right: $t\bar{t}H$ $p_T$. $m_H = m_A = 120$ GeV

Frederix, Frixione, Hirschi, Maltoni, Pittau, Torrielli, 1104.5613

# $(Z/\gamma^\star)(Z/\gamma^\star) \to e^+e^-\mu^+\mu^-$ with aMC@NLO




Left: $e^+e^-\mu^+\mu^-$ invariant mass. Right: $e^-\mu^-$ $\Delta\varphi$ distance

Include singly-resonant diagrams

## Automation: approaches with POWHEG

- HELAC, $t\bar{t} + 1j$ (Kardos, Papadopoulos, Trocsanyi 1101.2672); exploits HELAC-1Loop and POWHEG-Box (Alioli, Nason, Oleari, Re 1002.2581)
- SHERPA (Hoeche, Krauss, Schoenerr, Siegert 1008.5399); embedded in a unique framework, but lacks one-loop corrections. Only very simple processes so far

Again, no systematic comparison and assessment of automation capabilities

The situation is the same as in early 2000 with tree-level calculations: comparisons and cross-checks will be done in due time

# Current situation

- The project MadFKS+MadLoop+aMC@NLO has started in 2009. It took less to arrive at automatic matching to showers than it would have taken if we had computed in the usual way *a few of* the most complicated processes we have considered *at the parton level*

- Computing, matching, and validating $e\nu b\bar{b}$, or $(H/A)t\bar{t}$, or $e^+e^-\mu^+\mu^-$ production would be $\mathcal{O}(1)$ year of work of $\mathcal{O}(1)$ skilled person(s). It now takes one week

- We believe the codes are very robust. For example, MadLoop helped spot mistakes in published loop computations $(Zjj,\ W^+W^+jj)$

- As far as running time is concerned, we are presently limited by MadLoop

This is NLO QCD for SM processes. What's next?

# Outlook

A significant decrease of running time is expected after the implemention of a few, very simple optimizations in MadLoop (essentially, caching information – no caching is done at present)

# Outlook

A significant decrease of running time is expected after the implemention of a few, very simple optimizations in MadLoop (essentially, caching information – no caching is done at present)

- Move to MadGraph v5 – further decrease of running and especially of code-generation time

# Outlook

A significant decrease of running time is expected after the implemention of a few, very simple optimizations in MadLoop (essentially, caching information – no caching is done at present)

- Move to MadGraph v5 – further decrease of running and especially of code-generation time
- Use faster trees (recursion). Obviously doable in FKS. Less obvious with OPP, but true (the key information are topologies and colour structures)

# Outlook

A significant decrease of running time is expected after the implemention of a few, very simple optimizations in MadLoop (essentially, caching information – no caching is done at present)

- Move to MadGraph v5 – further decrease of running and especially of code-generation time
- Use faster trees (recursion). Obviously doable in FKS. Less obvious with OPP, but true (the key information are topologies and colour structures)
- EW loop corrections and QCD/EW double perturbative expansion (one-loop is hard work; subtraction already done)

# Outlook

A significant decrease of running time is expected after the implemention of a few, very simple optimizations in MadLoop (essentially, caching information – no caching is done at present)

- Move to MadGraph v5 – further decrease of running and especially of code-generation time
- Use faster trees (recursion). Obviously doable in FKS. Less obvious with OPP, but true (the key information are topologies and colour structures)
- EW loop corrections and QCD/EW double perturbative expansion (one-loop is hard work; subtraction already done)
- MG v5 can generate Feynman rules from the Lagrangian. Can we generate UV and $R_2$ counterterms as well?

# Outlook

A significant decrease of running time is expected after the implemention of a few, very simple optimizations in MadLoop (essentially, caching information – no caching is done at present)

- Move to MadGraph v5 – further decrease of running and especially of code-generation time
- Use faster trees (recursion). Obviously doable in FKS. Less obvious with OPP, but true (the key information are topologies and colour structures)
- EW loop corrections and QCD/EW double perturbative expansion (one-loop is hard work; subtraction already done)
- MG v5 can generate Feynman rules from the Lagrangian. Can we generate UV and $R_2$ counterterms as well?
- The previous item would pave the way to NLO corrections to arbitrary, user-defined theories (e.g. MSSM), thus strictly putting LO and NLO in MadGraph on the same footing

# Conclusions

The significant progress made in the past few years by several groups has not only led to remarkable physics results, but also to two (unintended) sociological consequences:

- NLO computations will not require any expertise
- Hiring PhD's or young postdocs as (highly-skilled) human computers is not justified any longer

It is high time for this. The emphasis is now on doing exciting phenomenology, and on helping experimenters understand and use these new tools – an experimental analysis fully at NLO done without theory support is not science fiction