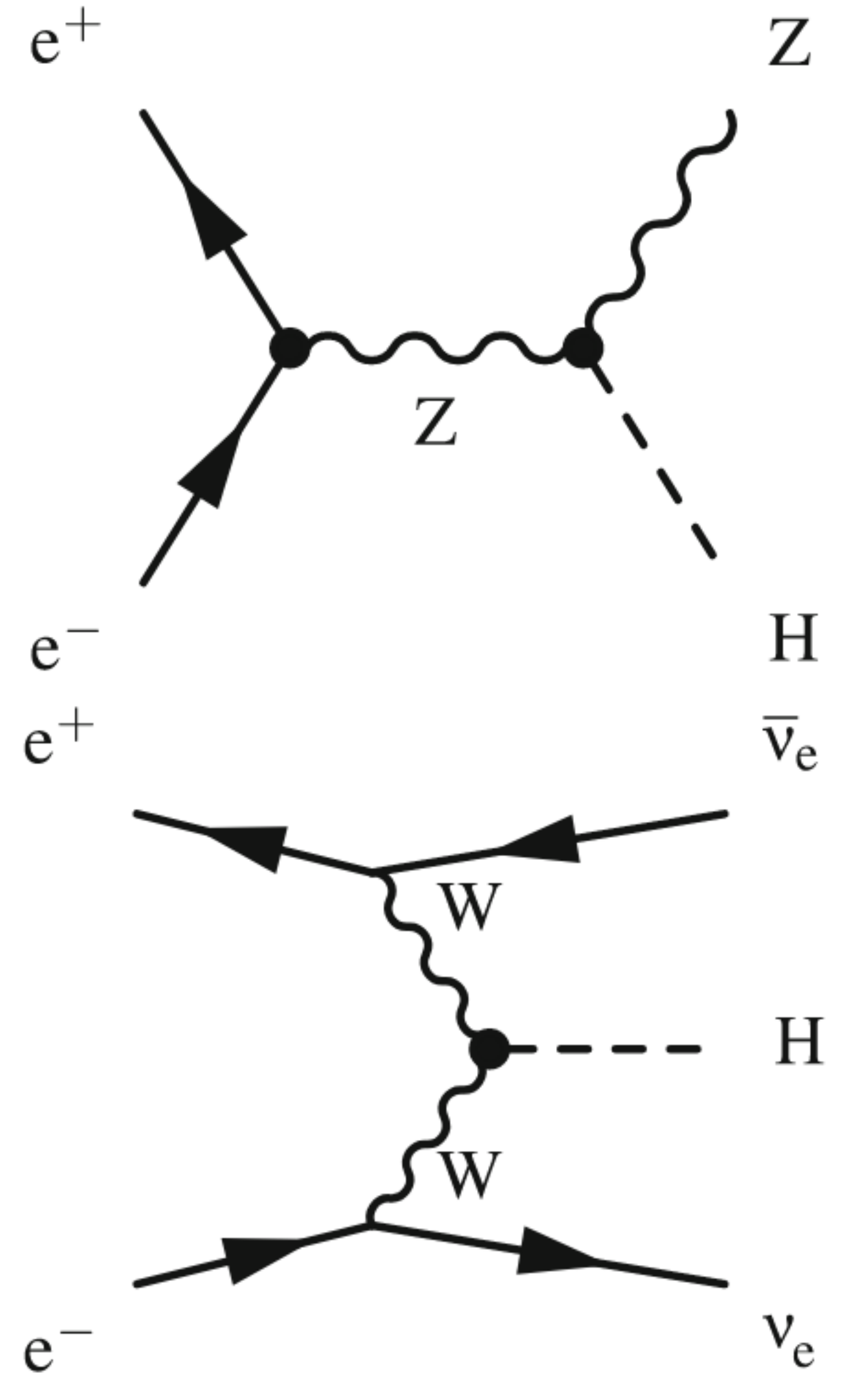
Higgs production and decay at e+e- colliders: theoretical status and challenges

Li Lin Yang
Zhejiang University

Higgs production at e+e- colliders

Higgs-strahlung (ZH)



W-fusion (WWH)

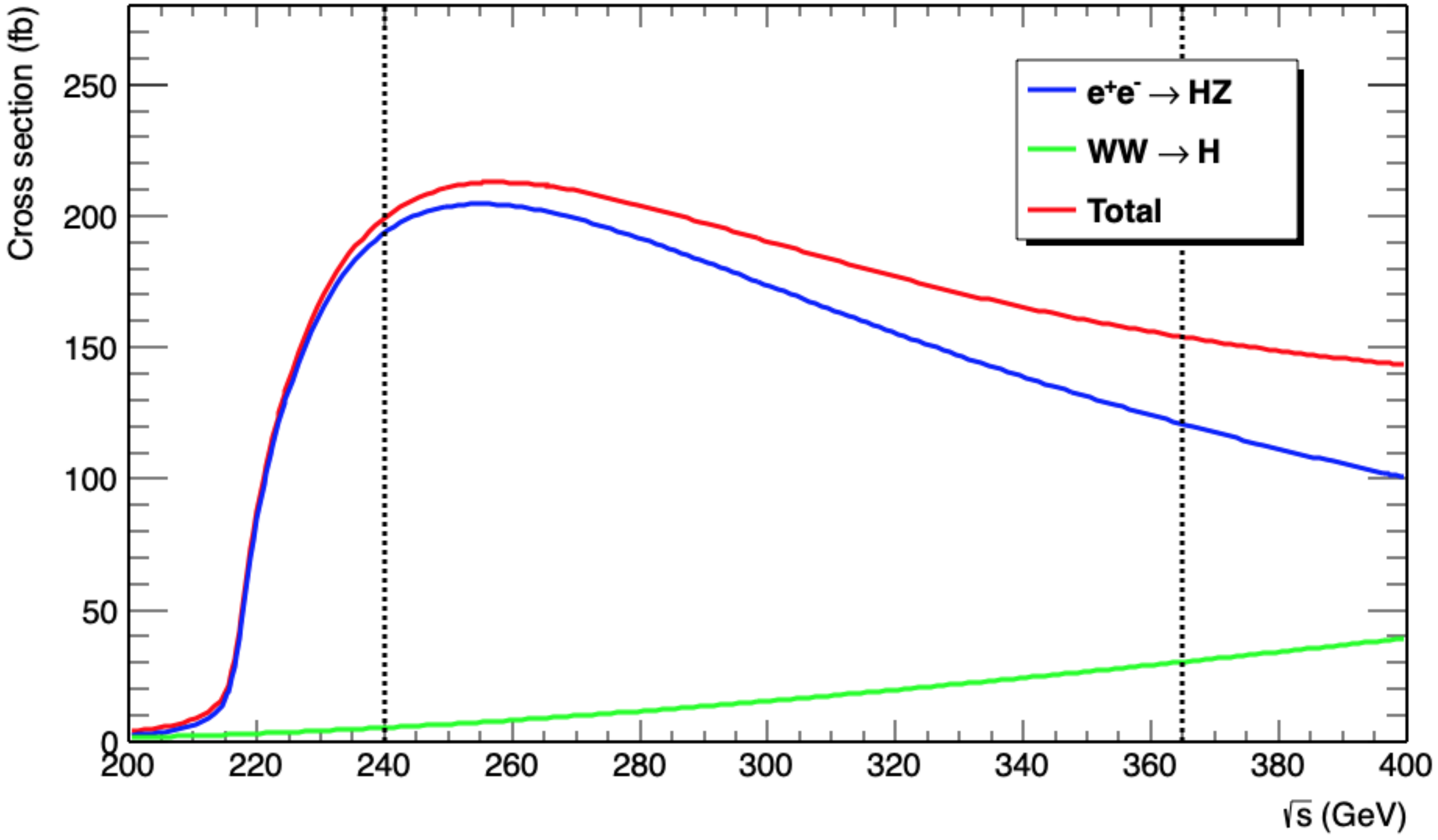


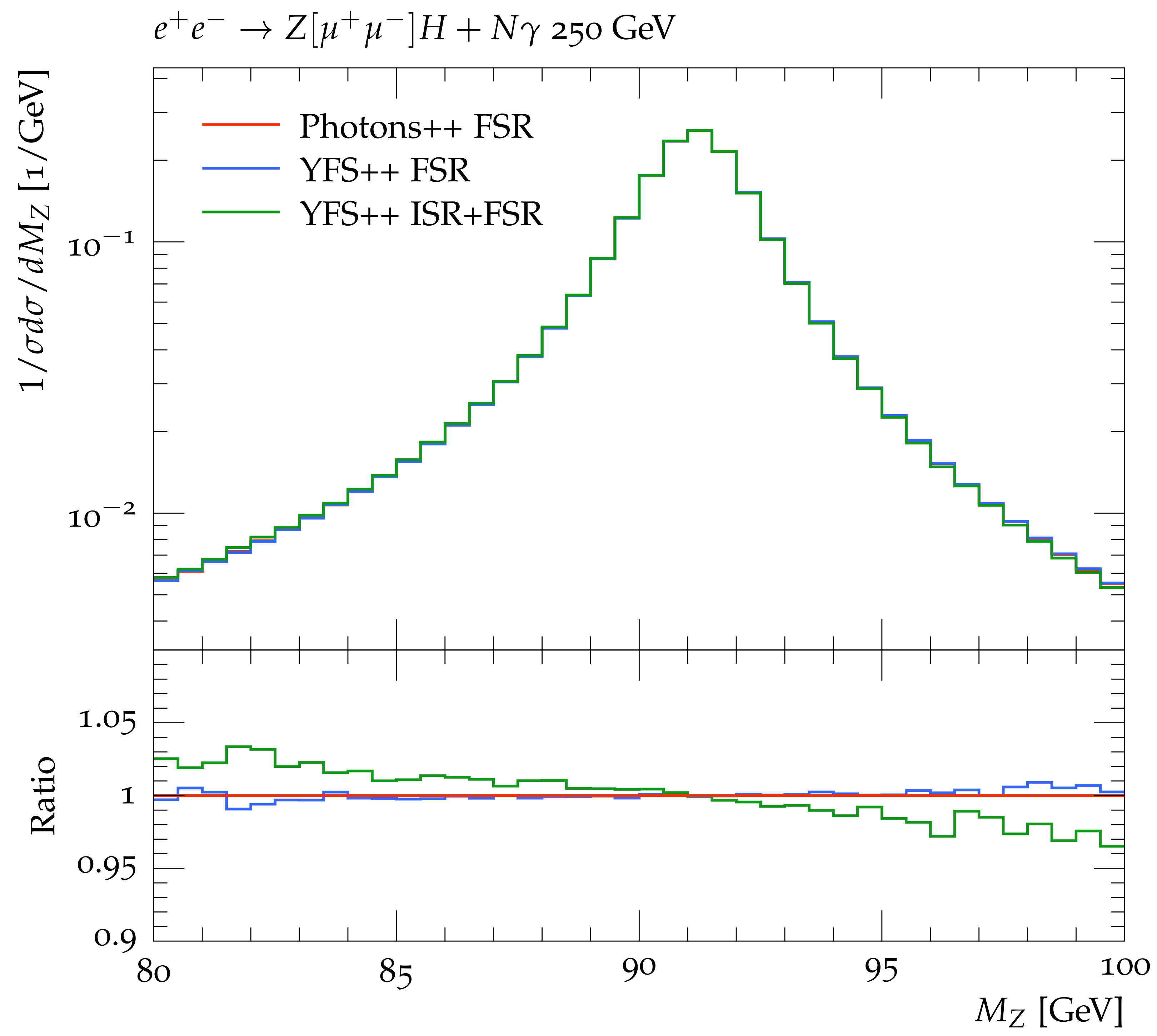
Figure from 2106.15438

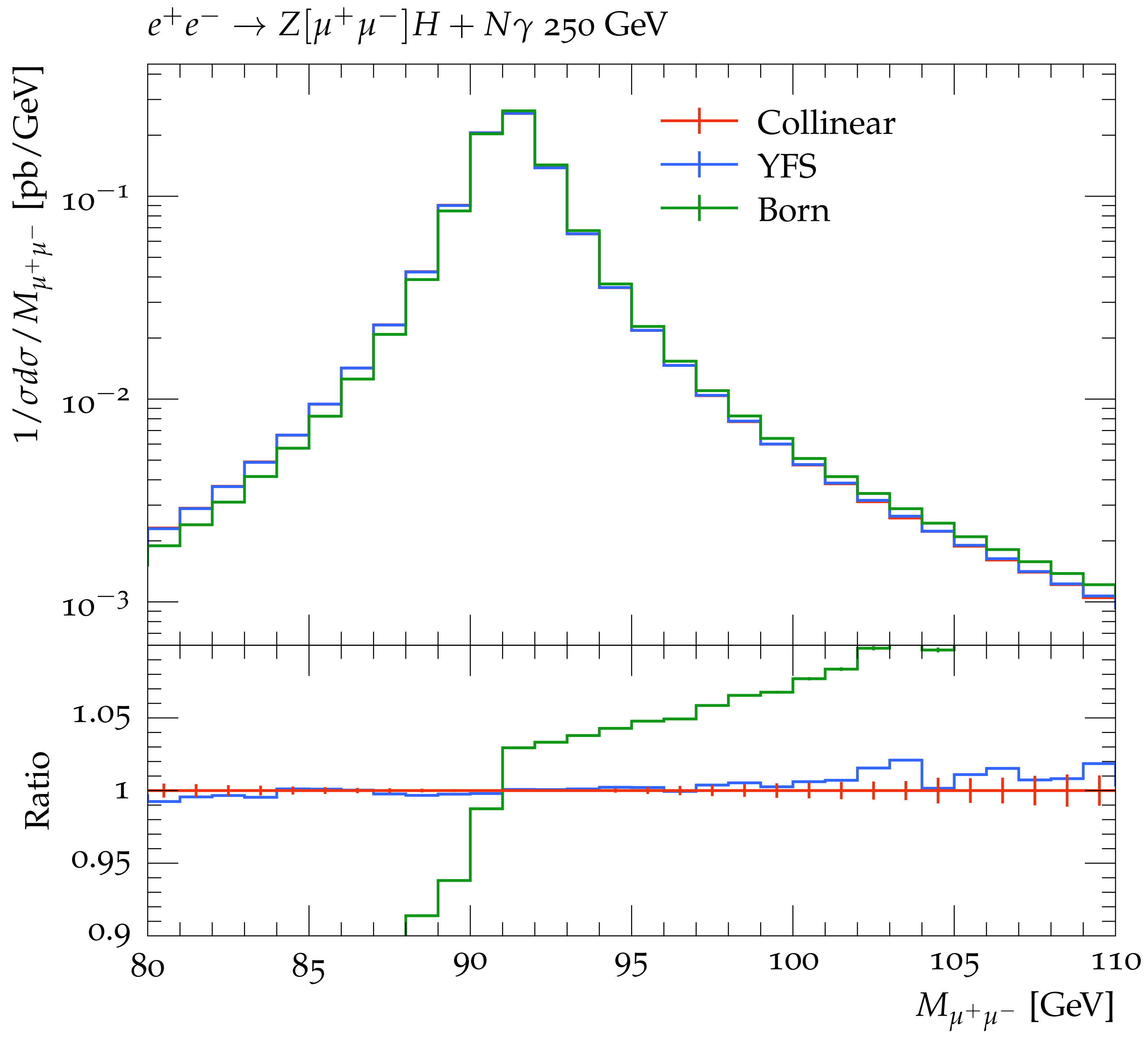
NLO EW + QED radiations built in Monte Carlo event generators

QED ISR and FSR effects

Critically re-examined very recently
(Talk by Stefano Frixione)

Blümlein, Schönwald: 2202.08476
Krauss, Price, Schönherr: 2203.10948
Frixione et al.: 2203.12557
and many more references therein





Figures from 2203.10948

Mixed QCD-EW corrections to ZH

Gong, Li, Xu, **LLY**, Zhao: 1609.03955

the $\alpha(m_Z)$ scheme.

$\sqrt{s}$ (GeV)	σ_{LO} (fb)	σ_{NLO} (fb)	σ_{NNLO} (fb)
240	252.0	228.6	231.5
250	252.0	227.9	230.8
300	190.0	170.7	172.9
350	135.6	122.5	124.2
500	60.12	54.03	54.42

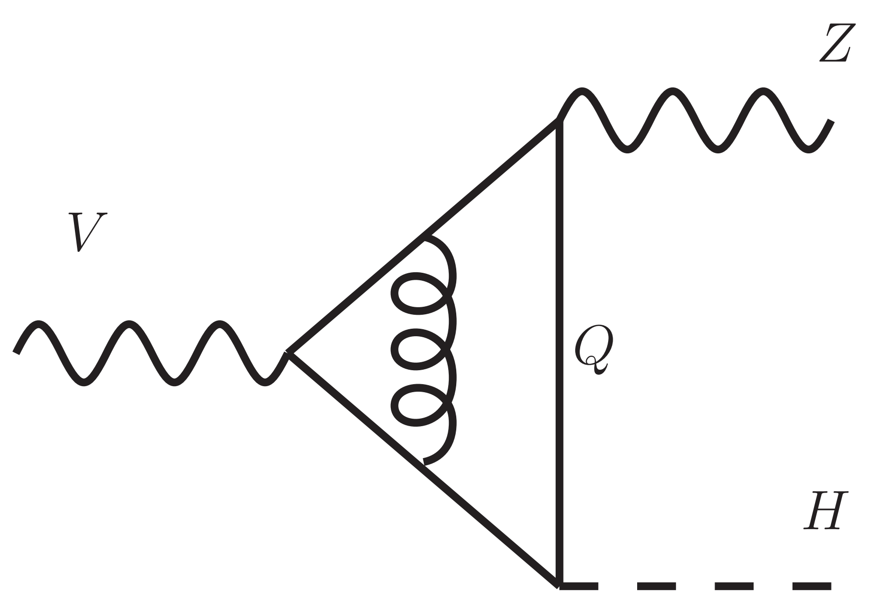
Corrections at the level of ~1%: non-negligible compared to the ~0.3% experimental accuracy

Sun, Feng, Jia, Sang: 1609.03995

$\sqrt{s}$	Schemes	σ_{LO} (fb)	σ_{NLO} (fb)	σ_{NNLO} (fb)
240	$\alpha(0)$	223.14 ± 0.47	229.78 ± 0.77	$232.21^{+0.75+0.10}_{-0.75-0.21}$
	$\alpha(M_Z)$	252.03 ± 0.60	$228.36^{+0.82}_{-0.81}$	$231.28^{+0.80+0.12}_{-0.79-0.25}$
	G_μ	239.64 ± 0.06	$232.46^{+0.07}_{-0.07}$	$233.29^{+0.07+0.03}_{-0.06-0.07}$

Residue dependence on renormalization schemes

Calculation methods back then



Bottleneck was the two-loop triangle integrals

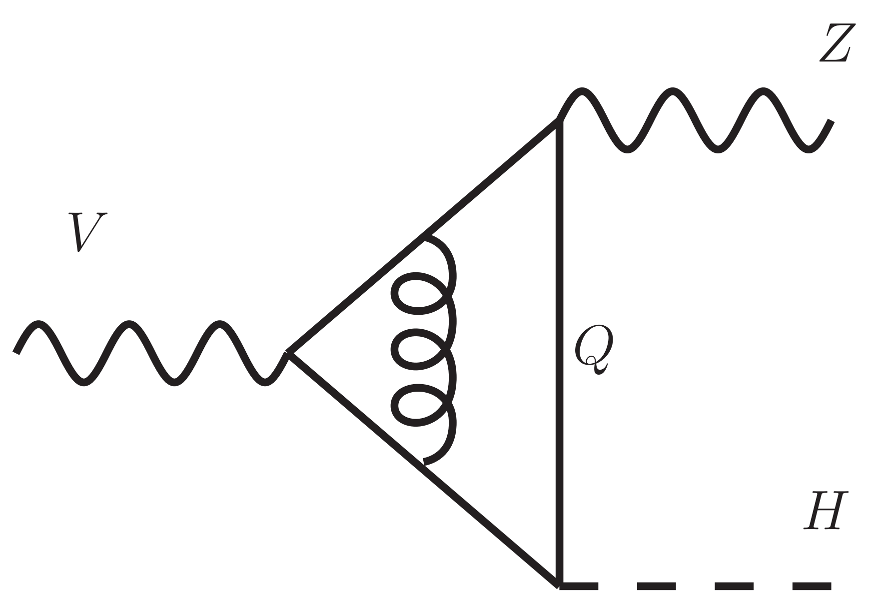
→ Purely numeric evaluation with sector decomposition

Private code of 1508.02512 (employed by 1609.03955)

FIESTA/CubPack (employed by 1609.03995)

Slow; bad convergence around or above $2m_Q$ threshold

Calculation methods back then



Bottleneck was the two-loop triangle integrals

→ Purely numeric evaluation with sector decomposition

Private code of 1508.02512 (employed by 1609.03955)

FIESTA/CubPack (employed by 1609.03995)

Slow; bad convergence around or above $2m_Q$ threshold

Alternative method: $1/m_t$ expansion

Gong, Li, Xu, **LLY**, Zhao: 1609.03955

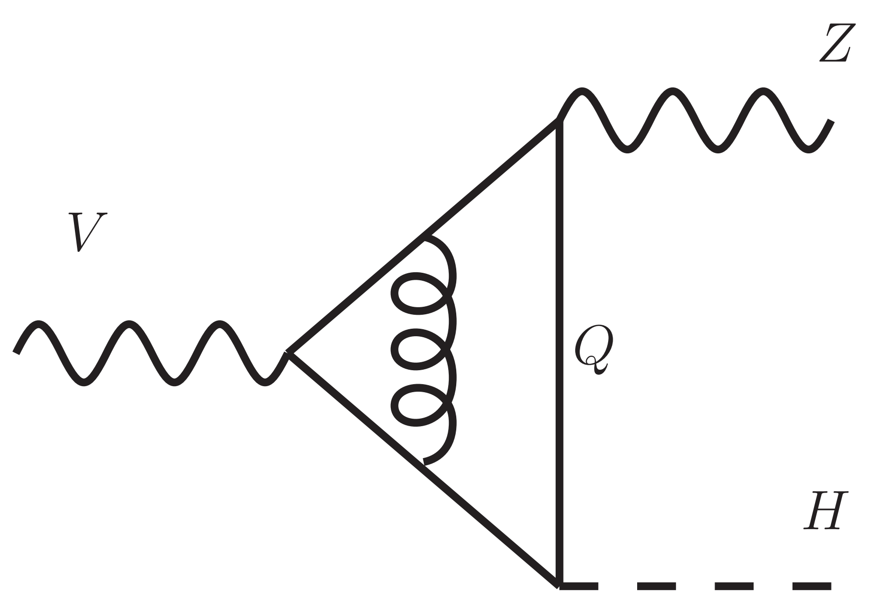
$\sqrt{s}$ (GeV)	$\mathcal{O}(m_t^2)$	$\mathcal{O}(m_t^0)$	$\mathcal{O}(m_t^{-2})$	$\mathcal{O}(m_t^{-4})$
240	81.8%	16.2%	1.4%	0.4%
250	81.7%	16.1%	1.5%	0.5%
300	80.0%	15.2%	2.1%	1.1%
350	69.7%	12.6%	2.7%	2.1%
500	137%	18.6%	17.3%	31.1%

Good approximation for low energies: analytic expressions easy to implement in Monte-Carlo

Not valid for high energies…

A new calculation for the HZV two-loop diagrams

Wang, Xu, **LLY**: 1905.11463



$$x = -\frac{q^2}{4m_Q^2}$$

$$y = -\frac{p_Z^2}{4m_Q^2}$$

$$z = -\frac{p_H^2}{4m_Q^2}$$

Constructed a canonical basis of master integrals

$$d\vec{f}(x,y,z;\epsilon) = \epsilon\, dA(x,y,z)\, \vec{f}(x,y,z;\epsilon)$$

$$= \epsilon \sum_i A_i\, d\log(\alpha_i)\, \vec{f}(x,y,z;\epsilon)$$

Alphabet contains 4 kinds of square roots

$$\sqrt{x(x+1)} \quad \sqrt{y(y+1)} \quad \sqrt{z(z+1)} \quad \sqrt{x^2+y^2+z^2-2xy-2yz-2zx}$$

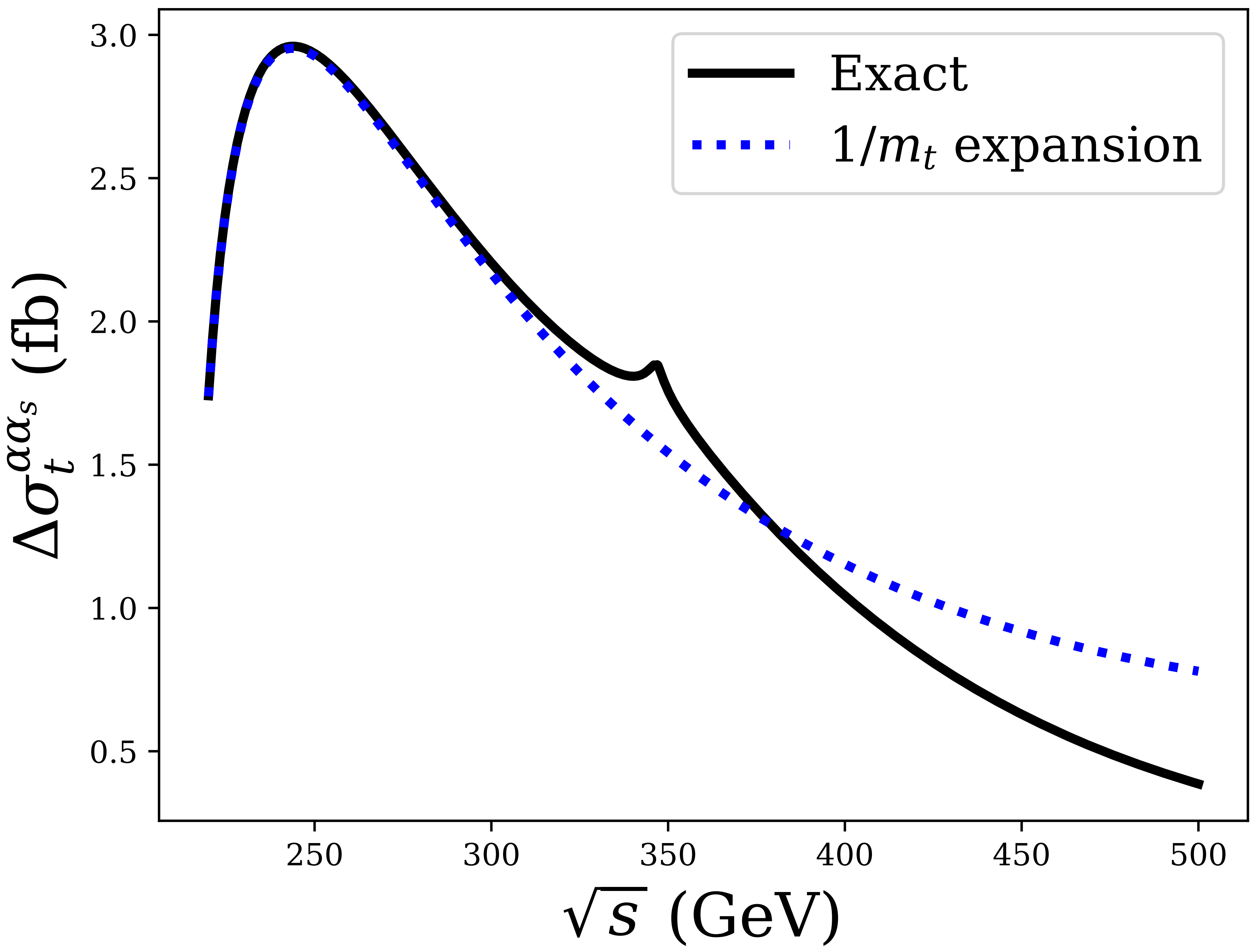
Solutions up to weight-3 written in terms of GPLs

Weight-4 parts expressed as one-fold integrals (not ideal, but usable)

A new calculation for the HZV two-loop diagrams

Wang, Xu, **LLY**: 1905.11463

The new result works well for all kinematic configurations

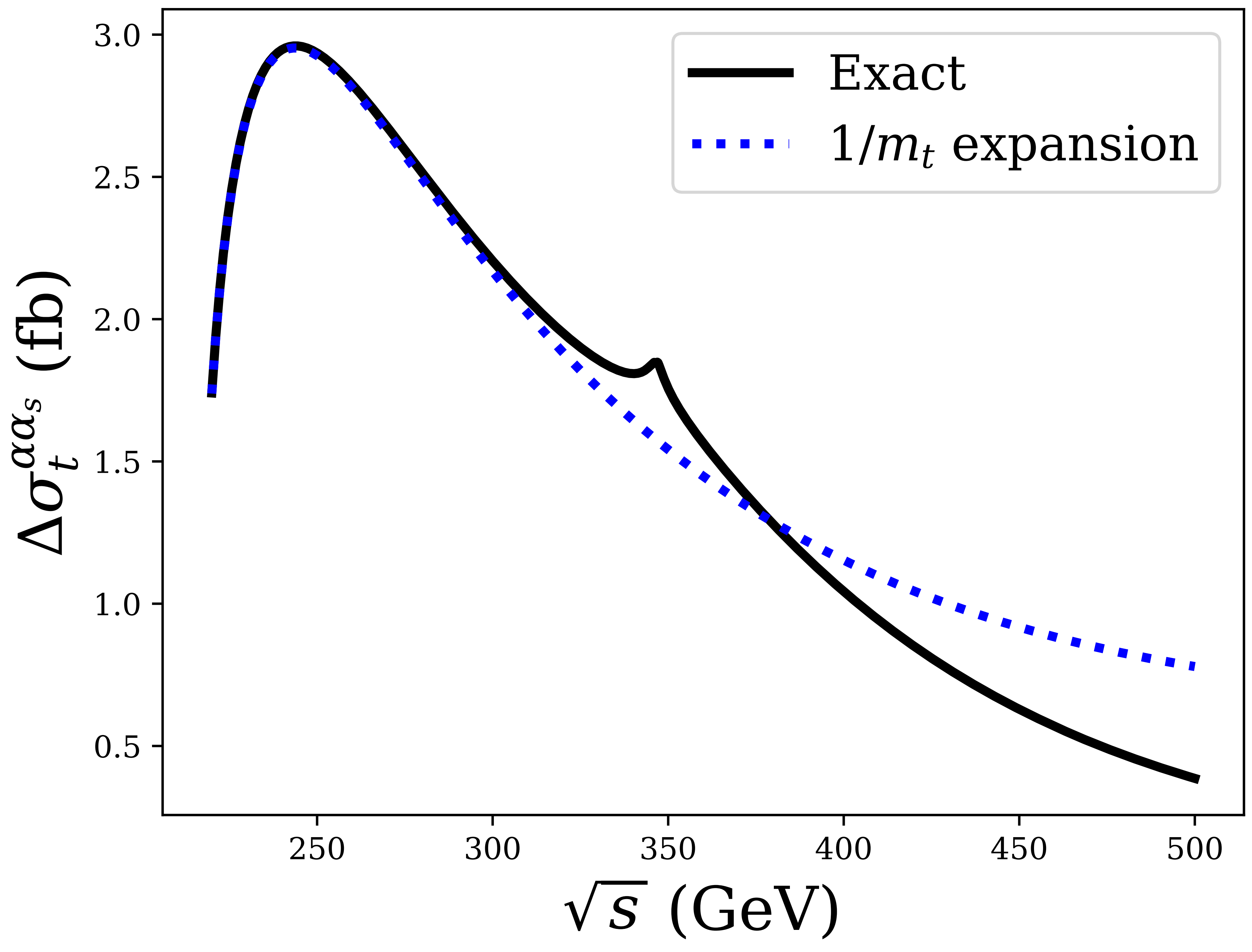


NNLO $\mathcal{O}(\alpha\alpha_s)$ corrections
to ZH cross section

A new calculation for the HZV two-loop diagrams

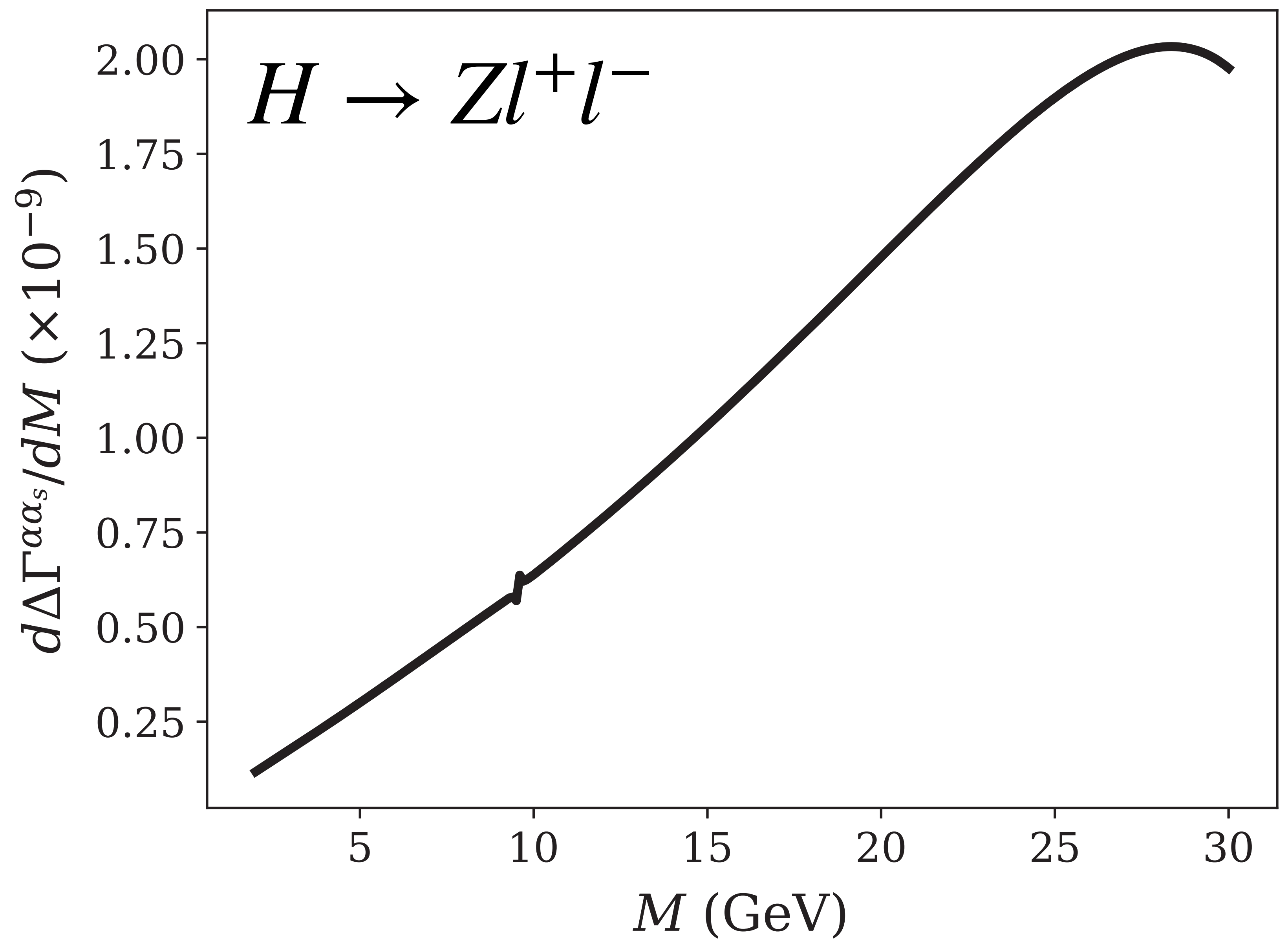
Wang, Xu, **LLY**: 1905.11463

The new result works well for all kinematic configurations



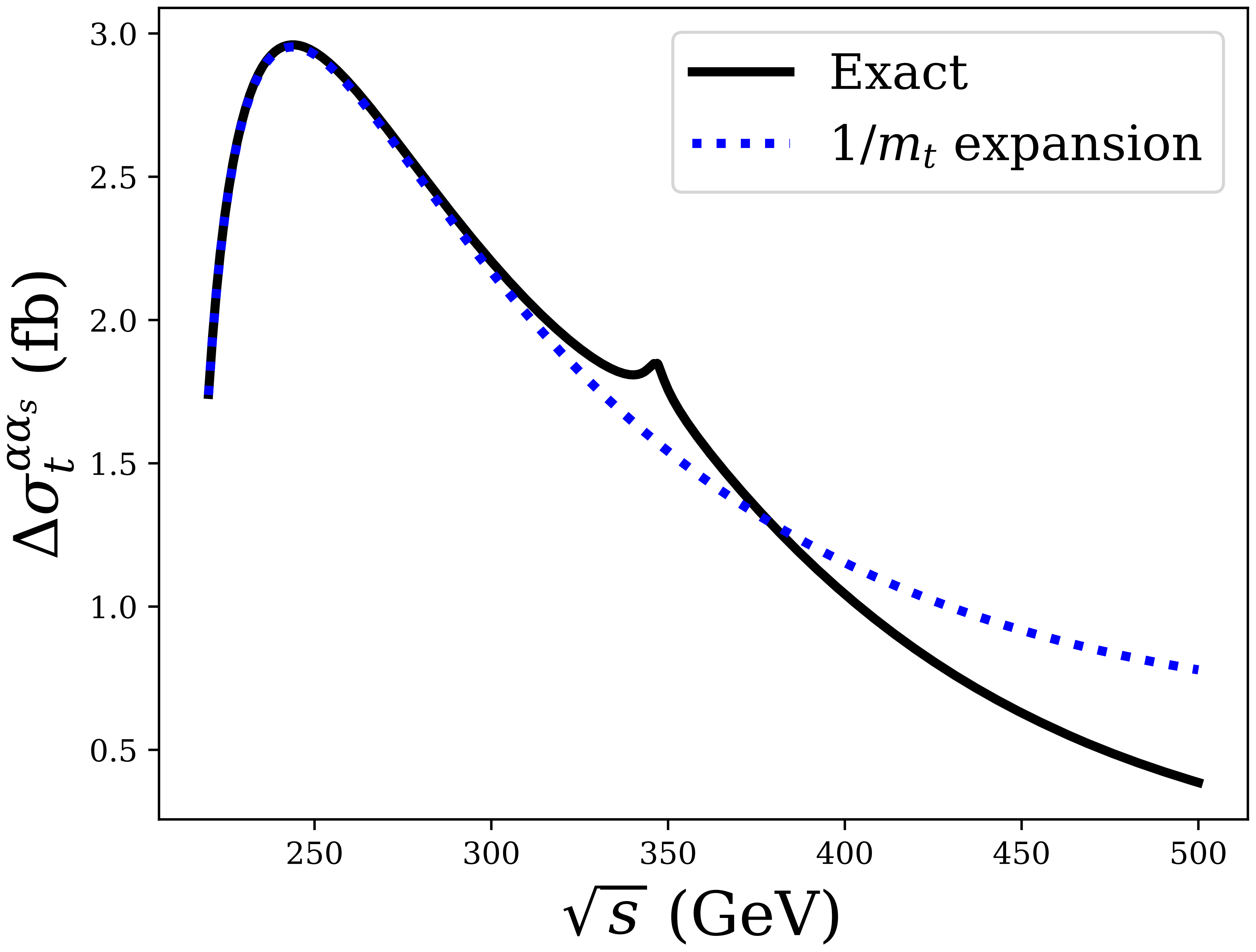
NNLO $\mathcal{O}(\alpha\alpha_s)$ corrections to ZH cross section

Also for bottom quark loops



Bottom contribution to the M_{ll} distribution

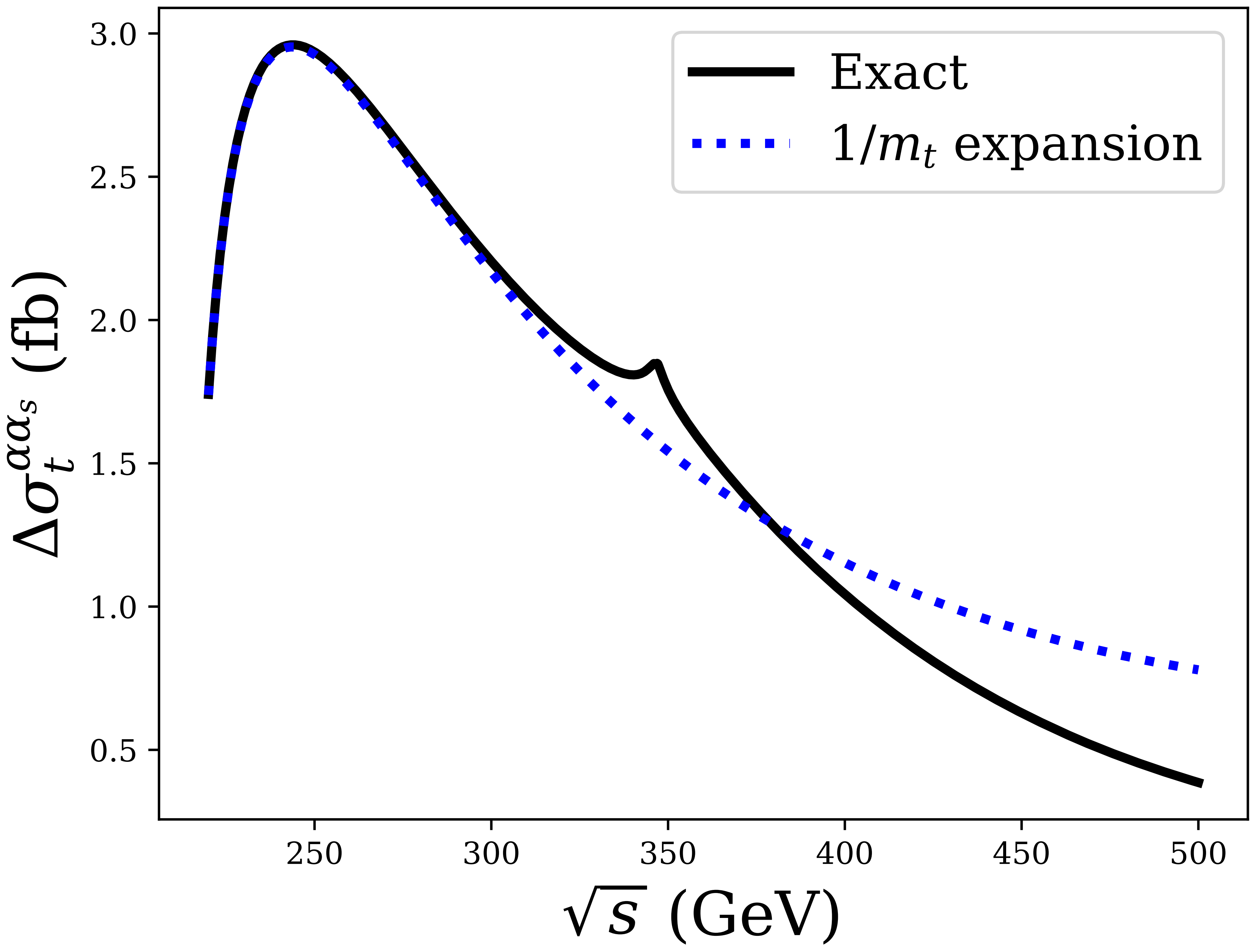
A different expansion?



While the $1/m_t$ expansion does not work at high energies, we have found that an expansion in the limit $m_H^2, m_Z^2 \ll s, m_t^2$ works pretty well!

Easier (than the exact result) to implement in Monte Carlo (available but unpublished)

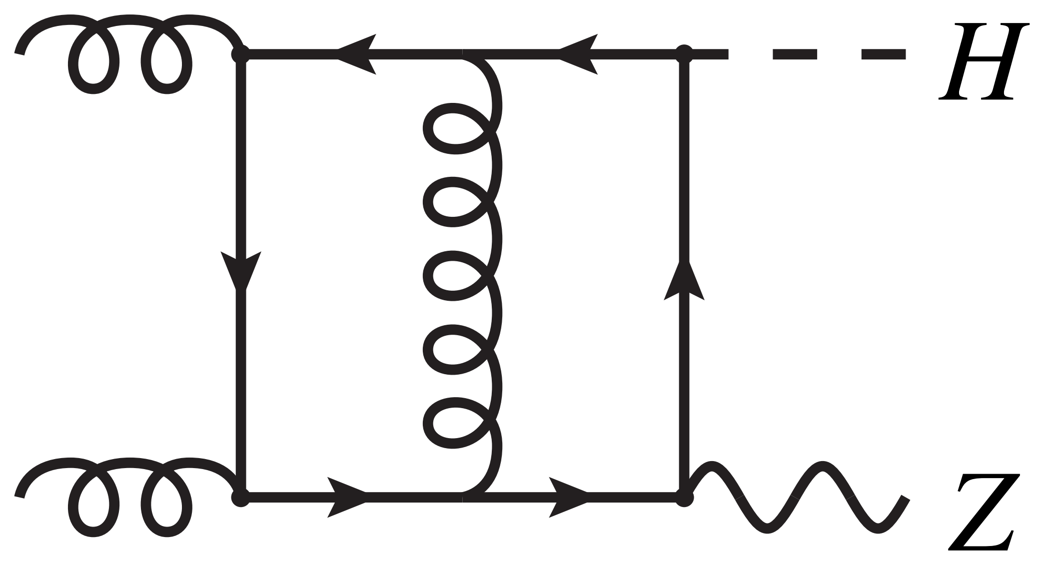
A different expansion?



While the $1/m_t$ expansion does not work at high energies, we have found that an expansion in the limit $m_H^2, m_Z^2 \ll s, m_t^2$ works pretty well!

Easier (than the exact result) to implement in Monte Carlo (available but unpublished)

Similar ideas have been successfully applied to $gg \to HH$ and $gg \to ZH$

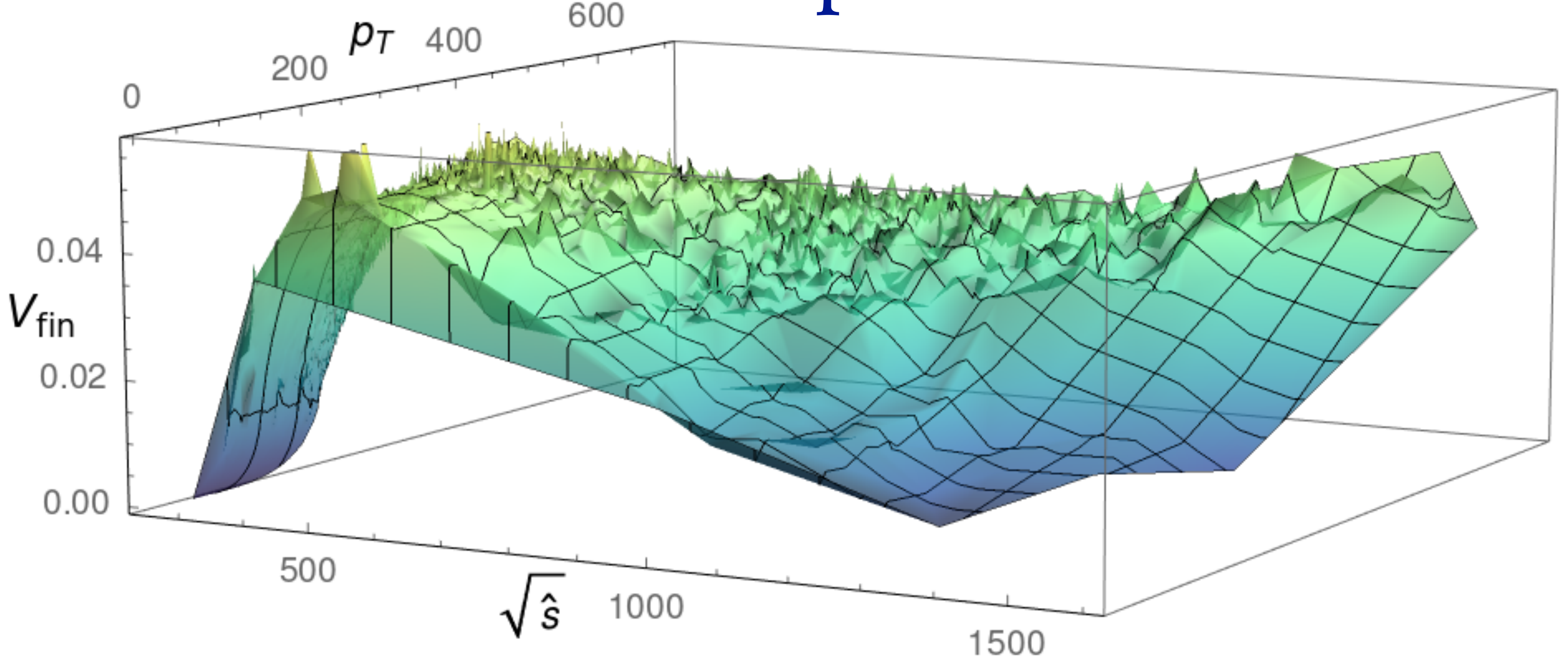


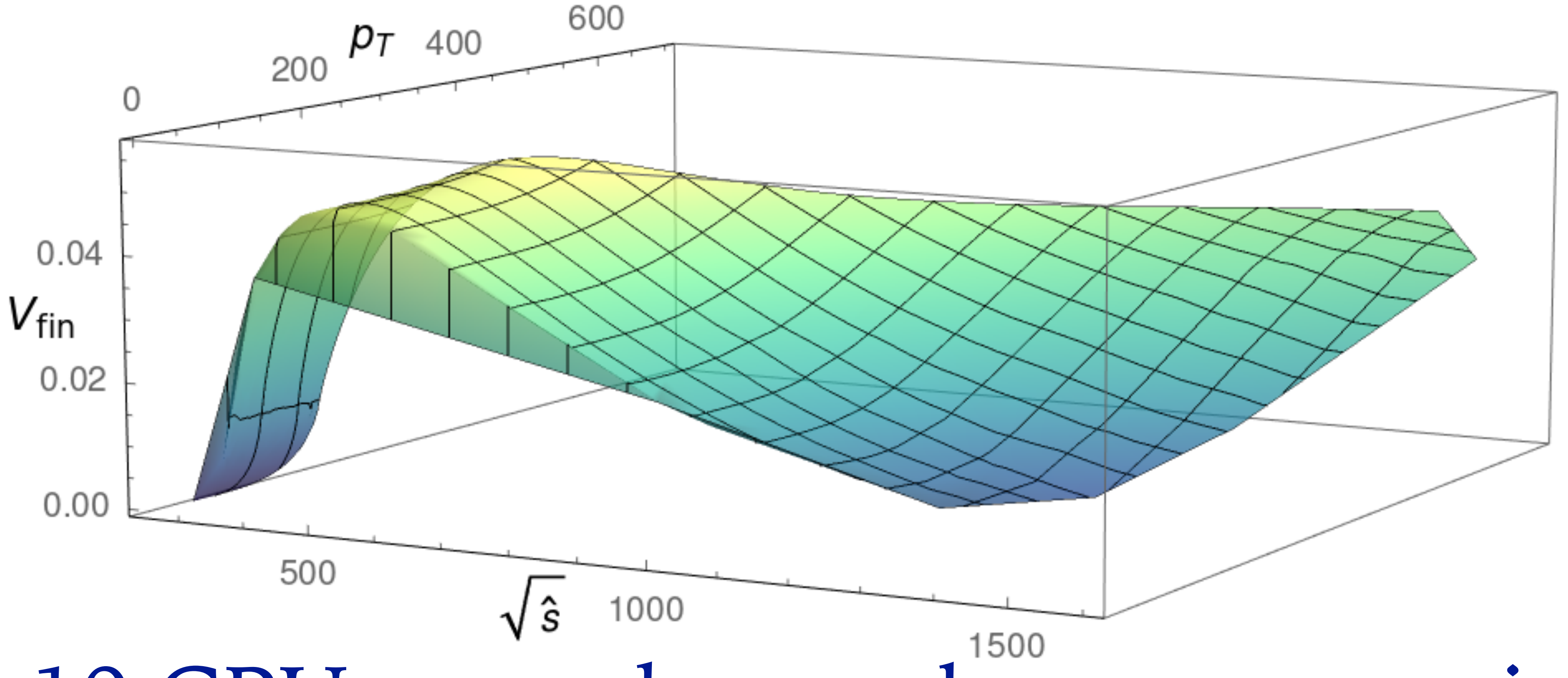
Xu, **LLY**: 1810.12002
Wang, Wang, Xu, Xu, **LLY**: 2010.15649
Wang, Xu, Xu, **LLY**: 2107.08206

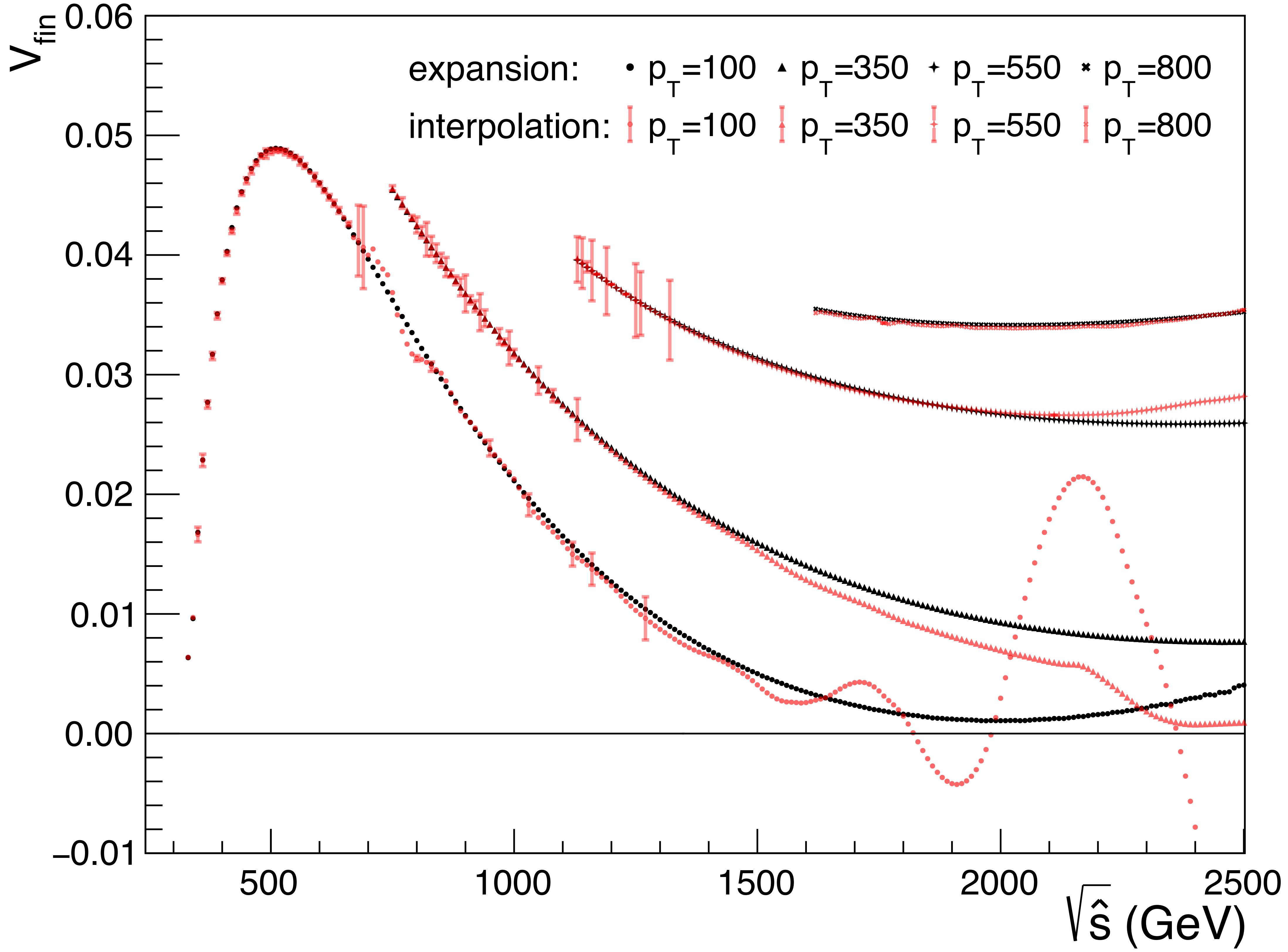
Off-topic: gg→HH

Wang, Wang, Xu, Xu, **LLY**: 2010.15649

7 GPGPU hours per phase space point with sector decomposition







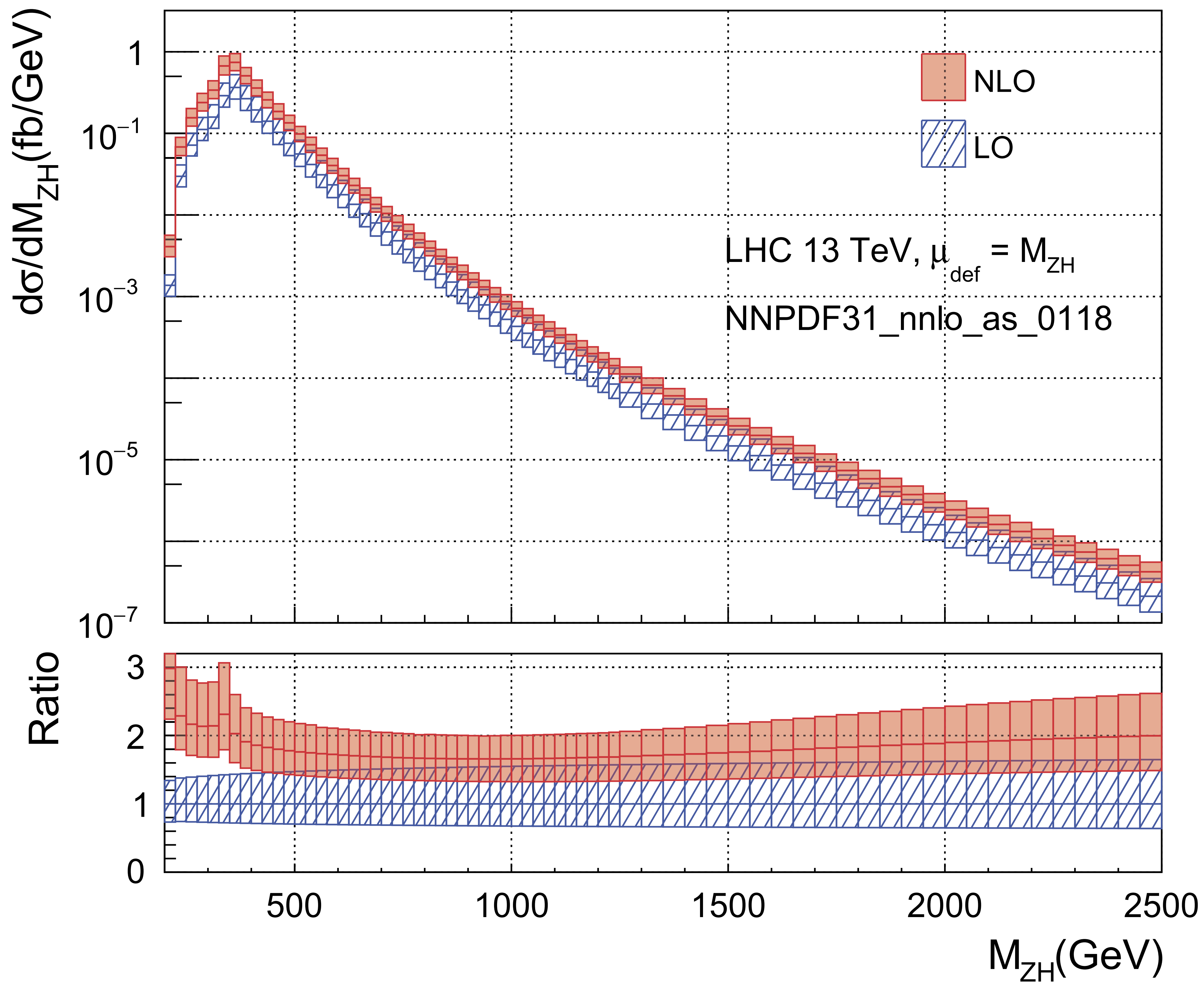
10 CPU seconds per phase space point with small-mass expansion

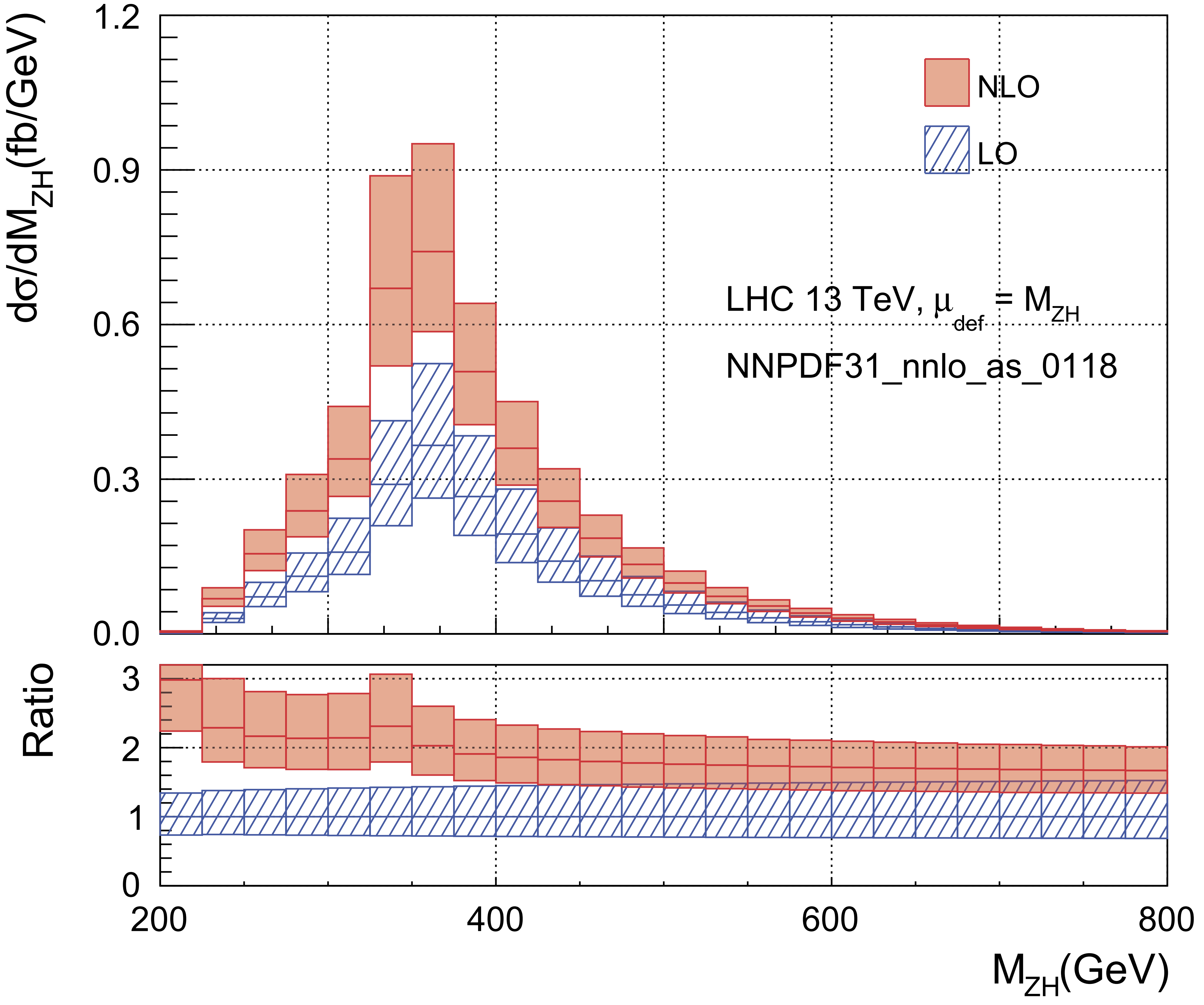
Off-topic: gg→ZH

Wang, Xu, Xu, **LLY**: 2107.08206

NLO predictions for both total and differential cross sections including top quark mass dependence

$$\sigma_{pp\to ZH} = 882.9^{+3.5\%}_{-2.5\%}\ \text{fb}$$



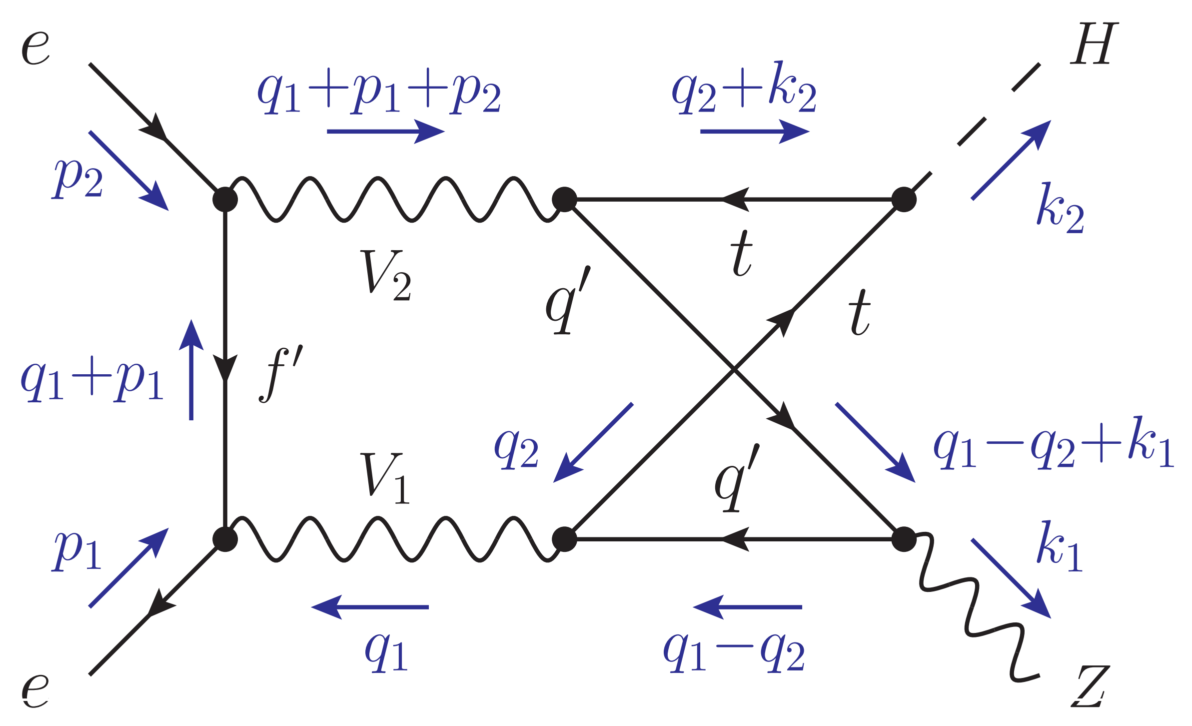


Towards two-loop EW corrections to ZH

A must to match the ~0.3% experimental accuracy

A rather challenging task: ~20000 diagrams, a lot of physical scales   Li, Wang, Wu: 2012.12513

Evaluation of a class of double boxes with a top quark loop   Song, Freitas: 2101.00308

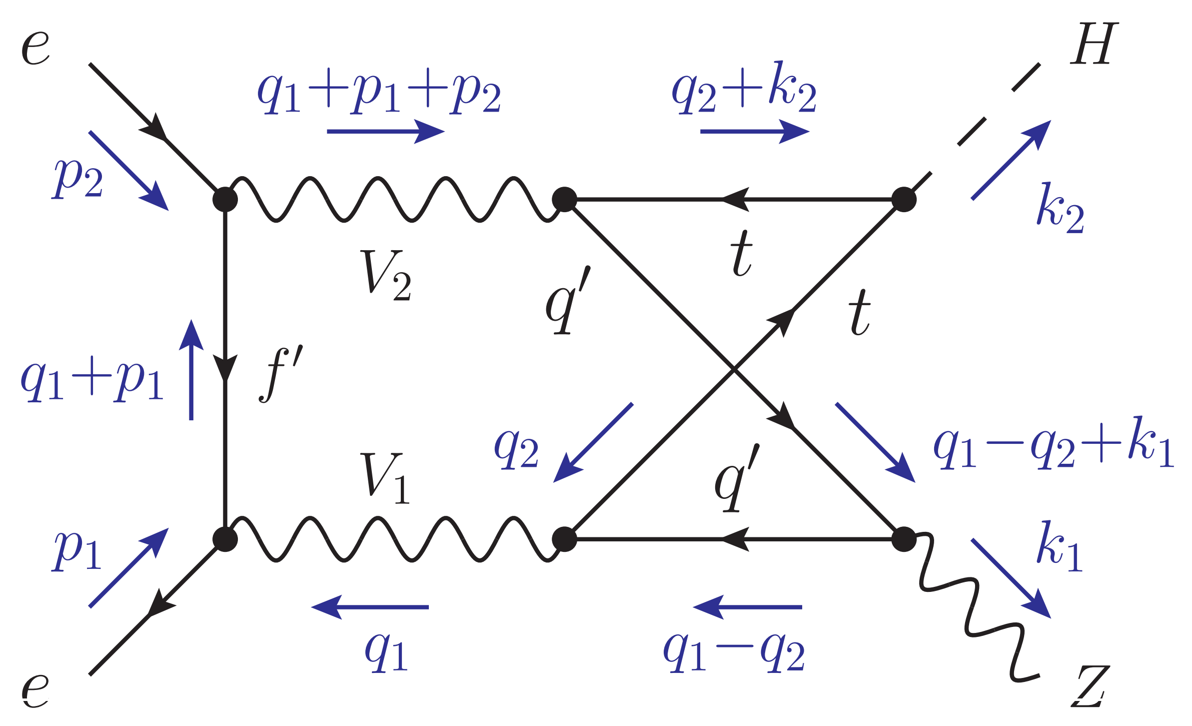


Towards two-loop EW corrections to ZH

A must to match the ~0.3% experimental accuracy

A rather challenging task: ~20000 diagrams, a lot of physical scales Li, Wang, Wu: 2012.12513

Evaluation of a class of double boxes with a top quark loop Song, Freitas: 2101.00308



Further development of computational techniques required!

→ Talks covering both analytic and numeric methods

e.g.: Canonical differential equations in both GPL sectors and elliptic sectors

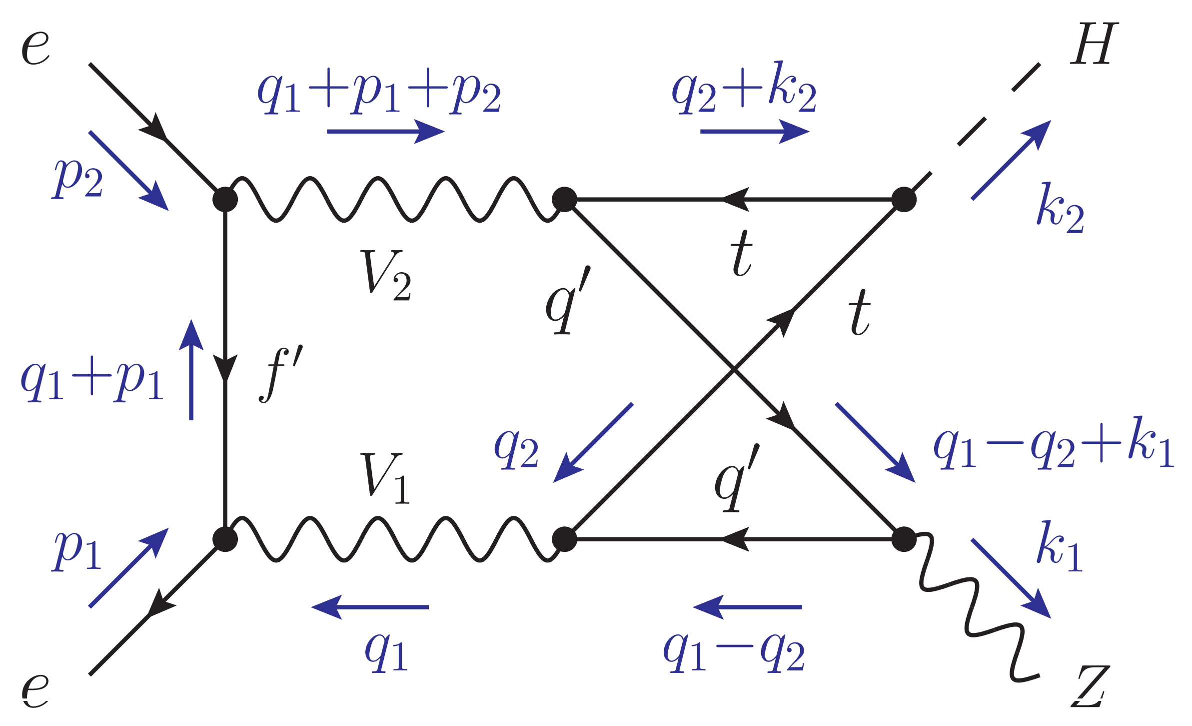
Numeric solutions (pySecDec, DiffExp, AMFlow, …)

Towards two-loop EW corrections to ZH

A must to match the ~0.3% experimental accuracy

A rather challenging task: ~20000 diagrams, a lot of physical scales Li, Wang, Wu: 2012.12513

Evaluation of a class of double boxes with a top quark loop Song, Freitas: 2101.00308



Further development of computational techniques required!

→ Talks covering both analytic and numeric methods

e.g.: Canonical differential equations in both GPL sectors and elliptic sectors

Numeric solutions (pySecDec, DiffExp, AMFlow, …)

Perhaps some kind of approximate result is good enough

→ Vague thought: asymptotic expansion in the limit $m^2_{\text{everything}} \ll s, m_t^2$?

Construction of canonical bases

As an addition to existing methods (probably covered by Andreas von Manteuffel), we proposed a novel approach to construct canonical Feynman integrals

Chen, Jiang, Xu, **LLY**: 2008.03045
Chen, Jiang, Ma, Xu, **LLY**: 2202.08127

d-log integrals in the generalized loop-by-loop Baikov representation (a larger vector space than the space of Feynman integrals)

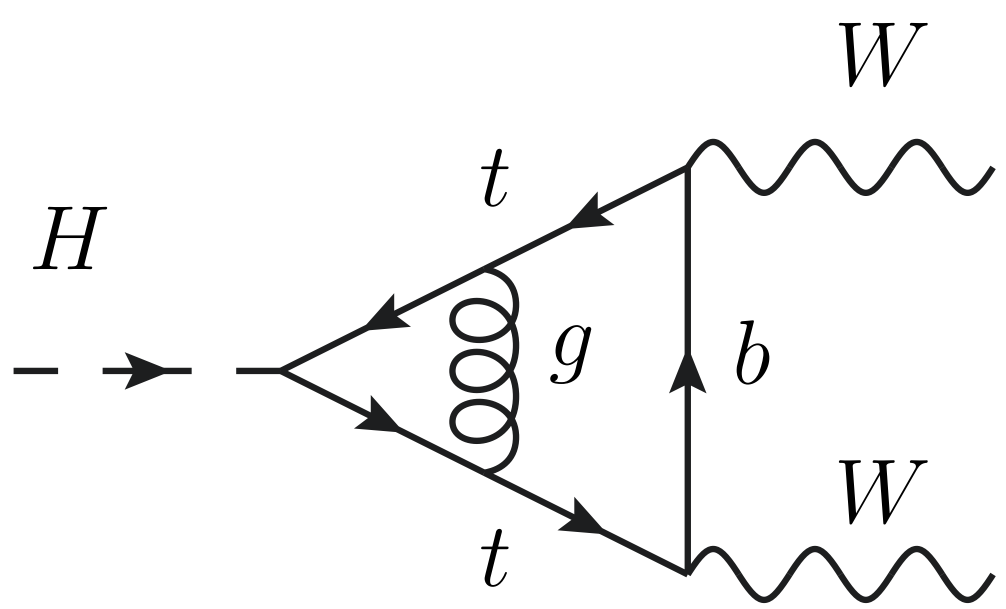
$$\int_{\mathscr{C}} \left[G(z)\right]^{\epsilon} \bigwedge_{j=1}^{n} d\log f_j(z) \longrightarrow \int \left[\prod_{i=1}^{L} \frac{d^d k_i}{i\pi^{d/2}}\right] \frac{1}{z_1^{a_1} z_2^{a_2} \cdots z_N^{a_N}}$$

Project into the space of Feynman integrals using intersection theory

$$\langle \varphi | = \sum_{i=1}^{\nu} c_i \langle e_i |$$

Mixed QCD-EW corrections to WWH

Di Vita, Mastrolia, Primo, Schubert: 1702.07331
Ma, Wang, Xu, **LLY**, Zhou: 2105.06316
Wang, **LLY**, Zhou: 2112.04122



The two-loop amplitude can be written in a fully-analytic form (involving a lot of weight-4 GPLs)

$H \to Wl\nu$

$\alpha(m_Z)$	LO	NLO EW	NNLO QCD-EW
$\Gamma\ (10^{-5}\ \text{GeV})$	4.597	4.474	4.518

G_μ	LO	NLO EW	NNLO QCD-EW
$\Gamma\ (10^{-5}\ \text{GeV})$	4.374	4.524	4.531

$e^+e^- \to \nu\bar{\nu}H$

$\sqrt{s}$ (GeV)	σ_{LO} (fb)	$\delta\sigma_{\text{NNLO}}$ (fb)
250	7.88	0.010
350	30.6	0.040
500	74.8	0.101

Rather small corrections

Note: did not consider mixing with $Z(\to \nu\bar{\nu}) + H$

In the future: two-loop EW? Perhaps only some approximations…

The numeric evaluation of GPLs

In all the above calculations one needs numeric evaluations of a large amount of GPLs

The algorithm has been well-known for many years

Gehrmann, Remiddi: hep-ph/0111255
Vollinga, Weinzierl: hep-ph/0410259
Ablinger, Blümlein, Schneider: 1302.0378

The numeric evaluation of GPLs

In all the above calculations one needs numeric evaluations of a large amount of GPLs

The algorithm has been well-known for many years

Gehrmann, Remiddi: hep-ph/0111255
Vollinga, Weinzierl: hep-ph/0410259
Ablinger, Blümlein, Schneider: 1302.0378

Program implementations:

GiNaC: works with arbitrary-precision numbers (slow), not optimized for double-precision floating point numbers   Vollinga, Weinzierl: hep-ph/0410259

The numeric evaluation of GPLs

In all the above calculations one needs numeric evaluations of a large amount of GPLs

The algorithm has been well-known for many years

Gehrmann, Remiddi: hep-ph/0111255
Vollinga, Weinzierl: hep-ph/0410259
Ablinger, Blümlein, Schneider: 1302.0378

Program implementations:

GiNaC: works with arbitrary-precision numbers (slow), not optimized for double-precision floating point numbers   Vollinga, Weinzierl: hep-ph/0410259

For Monte-Carlo, one may generate a large grid and interpolate from it, but for high precision applications, the grid has to be dense enough (slow to generate)

The numeric evaluation of GPLs

In all the above calculations one needs numeric evaluations of a large amount of GPLs

The algorithm has been well-known for many years

Gehrmann, Remiddi: hep-ph/0111255
Vollinga, Weinzierl: hep-ph/0410259
Ablinger, Blümlein, Schneider: 1302.0378

Program implementations:

GiNaC: works with arbitrary-precision numbers (slow), not optimized for double-precision floating point numbers Vollinga, Weinzierl: hep-ph/0410259

For Monte-Carlo, one may generate a large grid and interpolate from it, but for high precision applications, the grid has to be dense enough (slow to generate)

handyG: newer implementation using double-precision or quad-precision numbers, aimed for usage in Monte-Carlo Naterop, Signer, Ulrich: 1909.01656

The numeric evaluation of GPLs

The algorithm is recursive: one transforms the target GPL to a sum of so-called “convergent” GPLs, which can be evaluated by series expansion

A problem of **numerically** recursive implementations: to evaluate a single GPL, sometimes a transformed GPL needs to be computed for many many times!

- Greatly slows down the computation speed
- May lose accuracy due to repeated floating-point cancellations

The numeric evaluation of GPLs

The algorithm is recursive: one transforms the target GPL to a sum of so-called "convergent" GPLs, which can be evaluated by series expansion

A problem of **numerically** recursive implementations: to evaluate a single GPL, sometimes a transformed GPL needs to be computed for many many times!

- Greatly slows down the computation speed
- May lose accuracy due to repeated floating-point cancellations

We have encountered such situations in the calculation of $e^+e^- \to \nu\bar{\nu}H$: in general handyG can evaluate a weight-4 GPL in far less than a second, but sometimes it takes several seconds

$$\text{e.g.:} \quad \mathtt{G}(1.0025, 0.989, 0.45, 0.89 + 0.24\mathtt{i}; 1)$$

The problem becomes much worse at higher weights: at three-loops one needs weight-6

FastGPL

https://github.com/llyang/FastGPL

A re-implementation of the algorithm: hybrid analytic/numeric

The reduction to convergent GPLs are (mostly) done in a Mathematica package (to be released)

```
<< reduceGPL`
map[{1, 0, 1, 1}, 3]
{a, 0, b, b}
There is no any artificial divergence!
{a, 0, b, c}
There is artificial divergence when c=x!
We need to rescale indices and argument of GPLs!
complex<double> G4_a0bc_b(complex<double> a, complex<double> b, complex<double> c, int sa, int sb, int sc, double x) {
a=a/x;
b=b/x;
c=c/x;
x=1.;
if(b==c) {
const vector<complex<double>> sy = {G({a, b}, {sa, sb}, x), G({a}, {sa}, x), G({0, a/b, 1}, 1), G({b}, {sb}, x)};
complex<double> res= sy[1]*sy[2] - sy[2]*sy[3] + sy[0]*G({0, 1}, 1) - sy[1]*G({0, 0, 1}, 1) + sy[3]*G({0, a/b, x/b}, 1) + G({0, a/b,
  0, 1}, 1) - G({0, a/b, x/b, 1}, 1) + G({0, a/b}, 1)*(-sy[0] + G({0, b}, {1, sb}, x)) + G({a, 0, 0, b}, {sa, 1, 1, sb}, x);
return res;
 }
else {
const vector<complex<double>> sy = {Log(b, sb), G({a}, {sa}, x), G({c/b}, 1), G({a,
  c}, {sa, sc}, x), G({0, a}, {1, sa}, x), G({0, a/b, c/b}, 1), G({0, c/b, a/b}, 1), G({c/b, 0, a/b}, 1)};
complex<double> res= sy[0]*(sy[2]*sy[4] - sy[5] - sy[6] - sy[7]) + sy[3]*G({0, c/b}, 1) + 2.*G({0, 0, a/b, c/b}, 1) + 2.*G({0, 0, c/b, a/b},
  1) + G({0, a/b, 0, c/b}, 1) + G({0, a/b, c/b, x/b}, 1) + 2.*G({0, c/b, 0, a/b}, 1) + G({0, c/b, a/b, x/b}, 1) + 2.*G({c/b, 0, 0,
  a/b}, 1) + G({c/b, 0, a/b, x/b}, 1) + G({0, a/b}, 1)*(-sy[3] + G({0, c}, {1, sc}, x)) + sy[2]*(G({0, 0, a}, {1, 1, sa}, x) - G({a,
  0, c}, {sa, 1, sc}, x)) + G({a, 0, 0, c}, {sa, 1, 1, sc}, x) + (-(sy[0]*sy[1]*sy[2]) - sy[2]*sy[4] + sy[5] + sy[6] + sy[7])*Log(-x,
  sb) + (sy[1]*sy[2]*pow(sy[0], 2.))/2. + sy[1]*(-sy[6] - sy[7] - G({0, 0, c/b}, 1) + sy[2]*(-G({0, 0}, {1, 1}, x) - 2.*Zeta(2)));
if(c!=x) res += (-sy[5] + G({0, a/b, x/b}, 1))*G({c}, {sc}, x);
return res;
 }
}
```

Generate numeric codes automatically

↓

The FastGPL library (up to weight-4 well-tested, up to weight-6 implemented)

Aiming at fast evaluations using double-precision numbers

Comparison of speed

Wang, **LLY**, Zhou: 2112.04122

	t_f (s)	t_h (s)	t_h/t_f
$\texttt{G}(1.0025, 0.989, 0.45, 0.89 + 0.24\texttt{i}; 1)$	0.006	2.2	~ 400
$\texttt{G}(0.998, 1.0545 + 0.127\texttt{i}, 0.91 + 0.25\texttt{i}, -0.226; 1)$	0.004	1.5	~ 400
$\texttt{G}(-1.04, -0.97, 0.25, -0.84 + 0.45\texttt{i}; 1)$	0.004	1.1	~ 300

Table 2: Average evaluation times of several GPLs which require many iterations.

	$0aBC$	$0abC$	$0abc$	$00aB$	$00ab$
t_f (ms)	0.22	0.25	0.20	0.08	0.05
t_h (ms)	3.1	5.8	4.5	1.3	0.80
t_h/t_f	~ 14	~ 23	~ 23	~ 17	~ 16

	$ABCD$	$abCD$	$abcD$	$abcd$
t_f (ms)	0.22	0.47	0.50	0.42
t_h (ms)	1.7	7.4	11.0	9.1
t_h/t_f	~ 7.5	~ 16	~ 22	~ 22

Table 3: Average evaluation times of a few categories of weight-4 GPLs.

$e^+e^- \to \nu\bar{\nu}H$

$\sqrt{s}$ (GeV)	σ_{LO} (fb)	$\delta\sigma_{\mathrm{NNLO}}$ (fb)	t_f (h)	t_h (h)	t_h/t_f
250	7.88	0.010	0.45	8.60	~ 19
350	30.6	0.040	0.51	9.02	~ 18
500	74.8	0.101	0.52	9.24	~ 18

10000 sample phase-space points

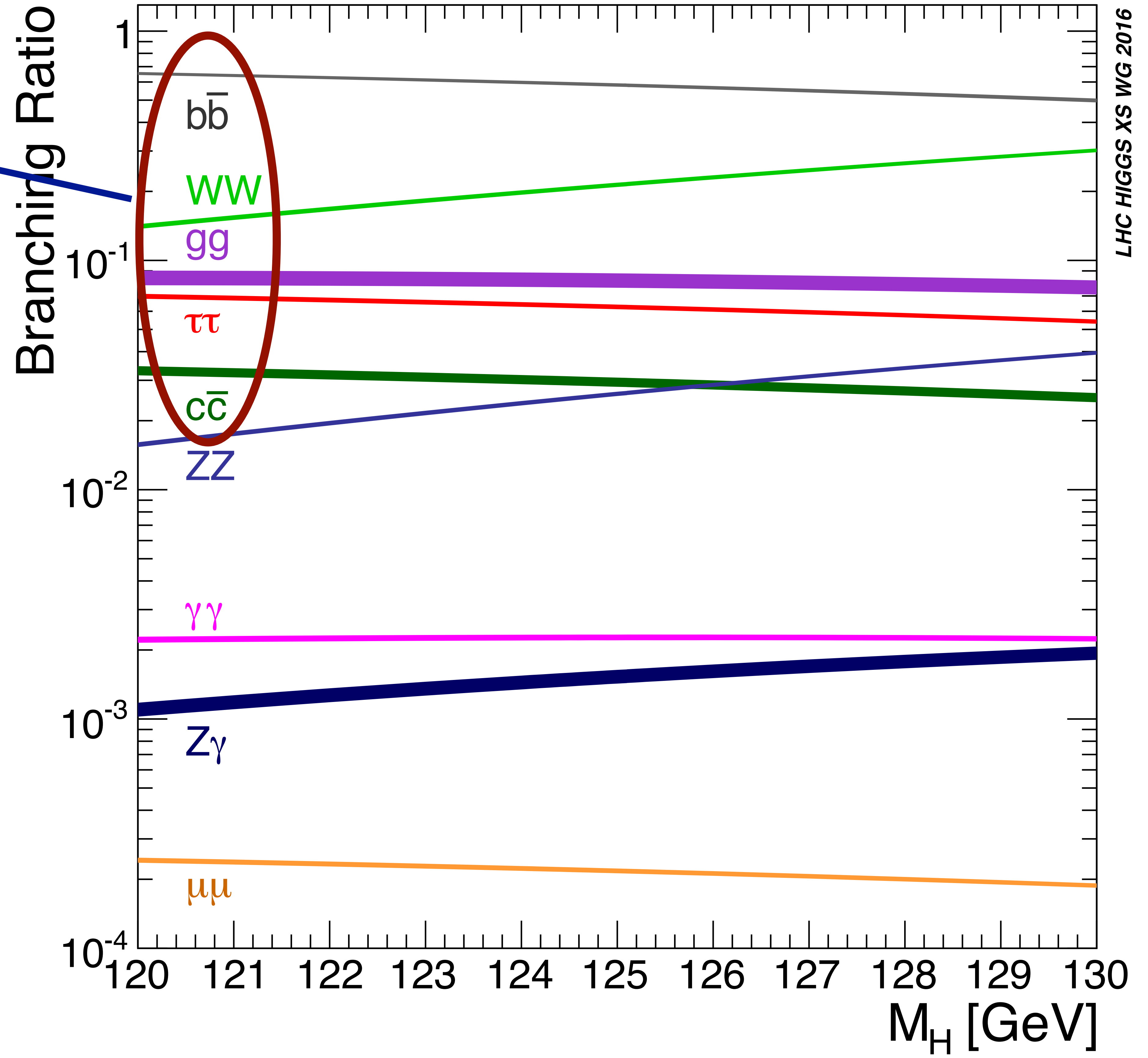
several thousand GPLs per point

FastGPL is faster in general, and is much faster for special cases

Preliminary tests show that the speed-boost is much larger at weight-6

Higgs decay

Weakness of the LHC



The hadronic channels

$H \to b\bar{b}$	Important for *HZZ* and $Hb\bar{b}$ couplings
$H \to gg$	Probes new particles running in the loop
$H \to c\bar{c}$	Unique window to charm Yukawa

Partial widths

Freitas (2021) and references therein

- $H \to q\bar{q}$
  - $\mathcal{O}(\alpha_s^4)$ in the limit of massless quarks
  - $\mathcal{O}(\alpha)$, $\mathcal{O}(\alpha\alpha_s)$ and partial $\mathcal{O}(\alpha^2)$
- $H \to gg$

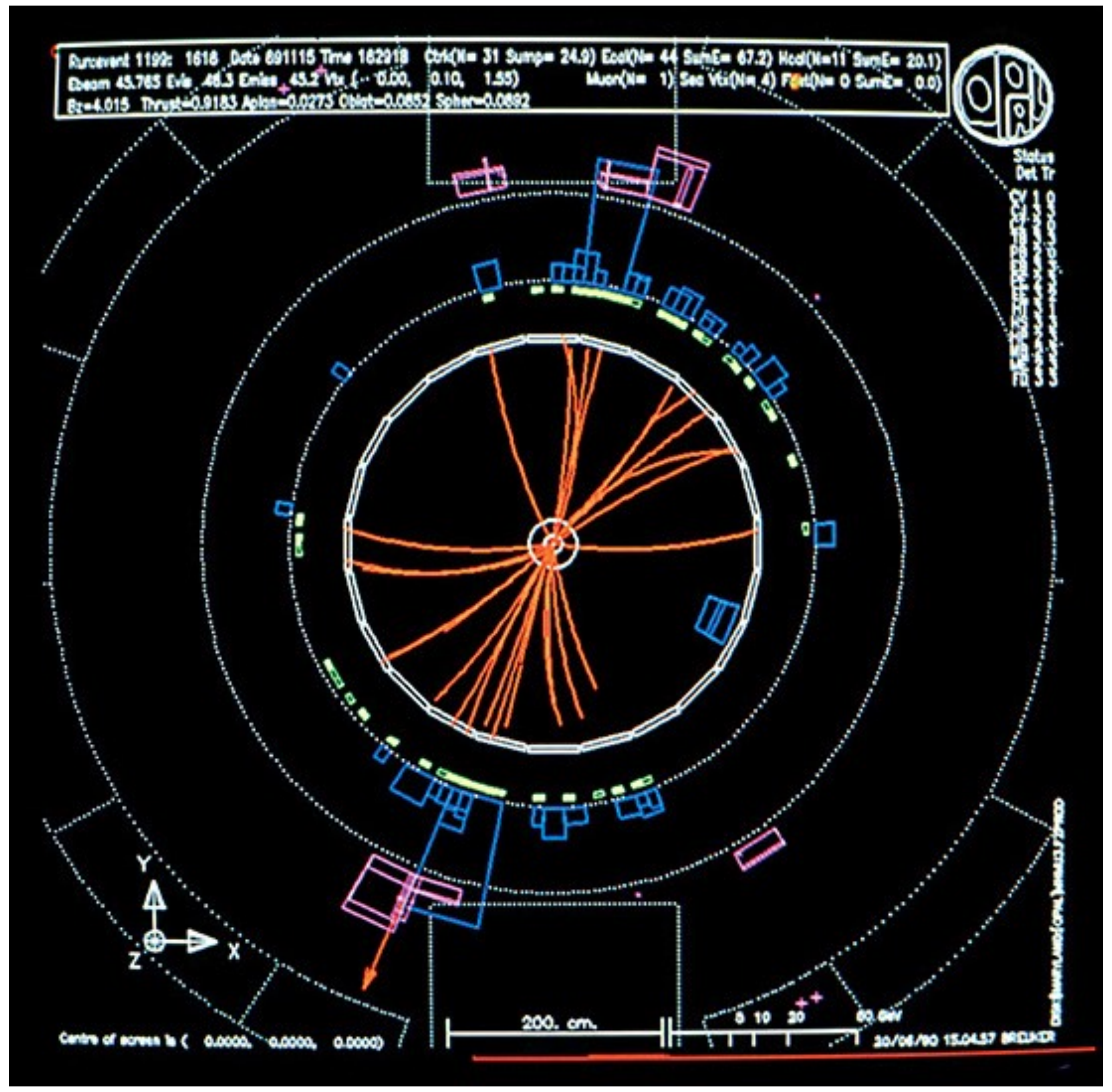
  Herzog et al.: 1707.01044
  - $\mathcal{O}(\alpha_s^4)$ with infinite m_t $\quad \Gamma_{\mathrm{N^4LO}}(H \to gg) = \Gamma_0 \left(1.844 \pm 0.011_{\,\mathrm{series}} \pm 0.045_{\,\alpha_s(M_Z),1\%}\right)$
  - $\mathcal{O}(\alpha_s^2)$ with $1/m_t$ expansion
  - $\mathcal{O}(\alpha_s^2)$ three-loop form factor with full m_t dependence (hence also bottom loop)

    Czakon, Niggetiedt: 2001.03008
  - $\mathcal{O}(\alpha)$ EW corrections

Event shapes

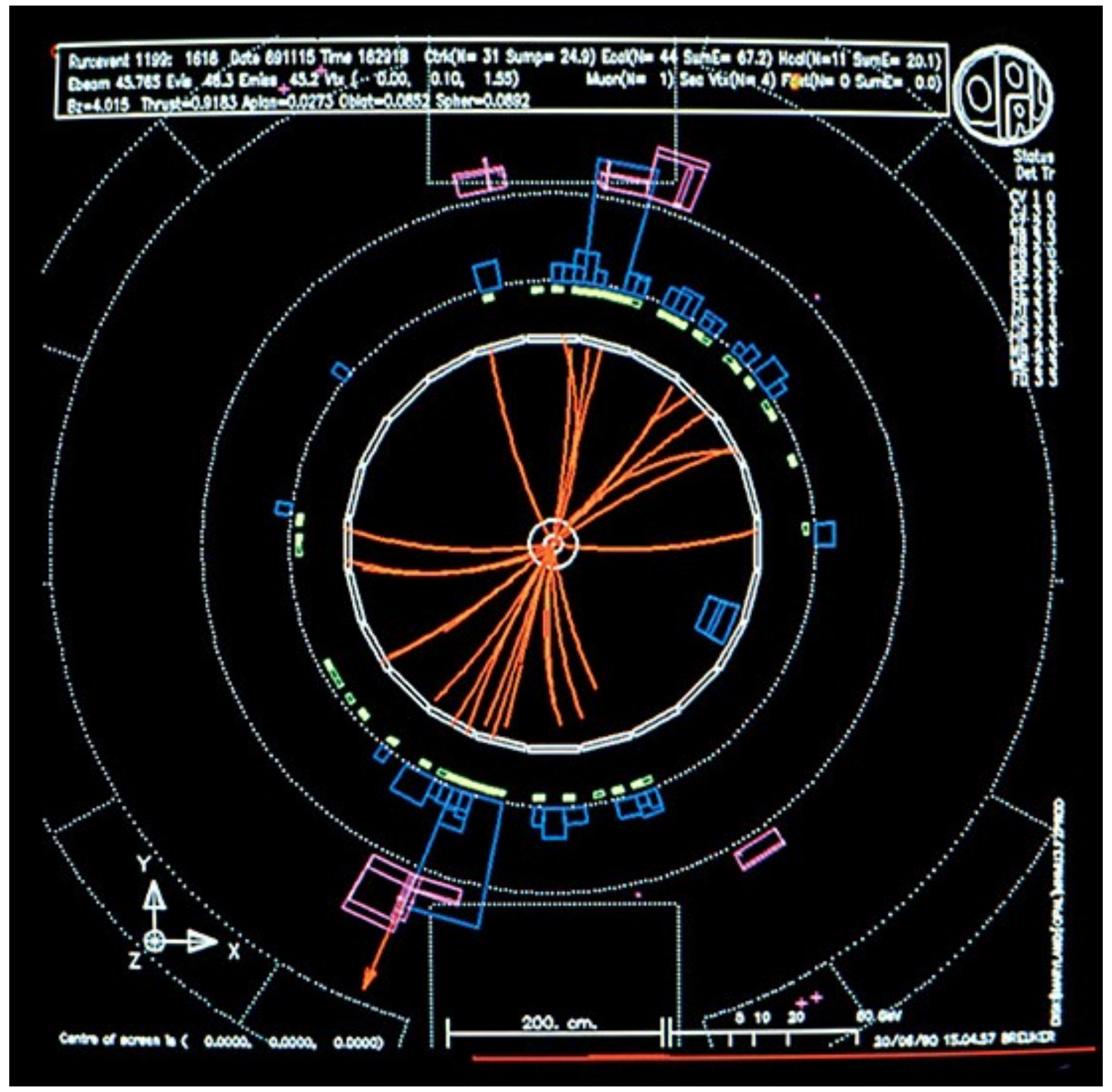
Event shapes provide more information than the total rates

- ➤ Discrimination between quark and gluon final states
- ➤ Probing kinematic dependence of the Hgg vertex
- ➤ New-physics enhanced light-quark Yukawa couplings?



Event shapes

Event shapes provide more information than the total rates



- ➤ Discrimination between quark and gluon final states
- ➤ Probing kinematic dependence of the Hgg vertex
- ➤ New-physics enhanced light-quark Yukawa couplings?

I'll focus on one particular variable: thrust

$$T = \max_{\vec{n}} \frac{\sum_i |\vec{n} \cdot \vec{p}_i|}{\sum_i |\vec{p}_i|}$$

$$\tau = 1 - T$$

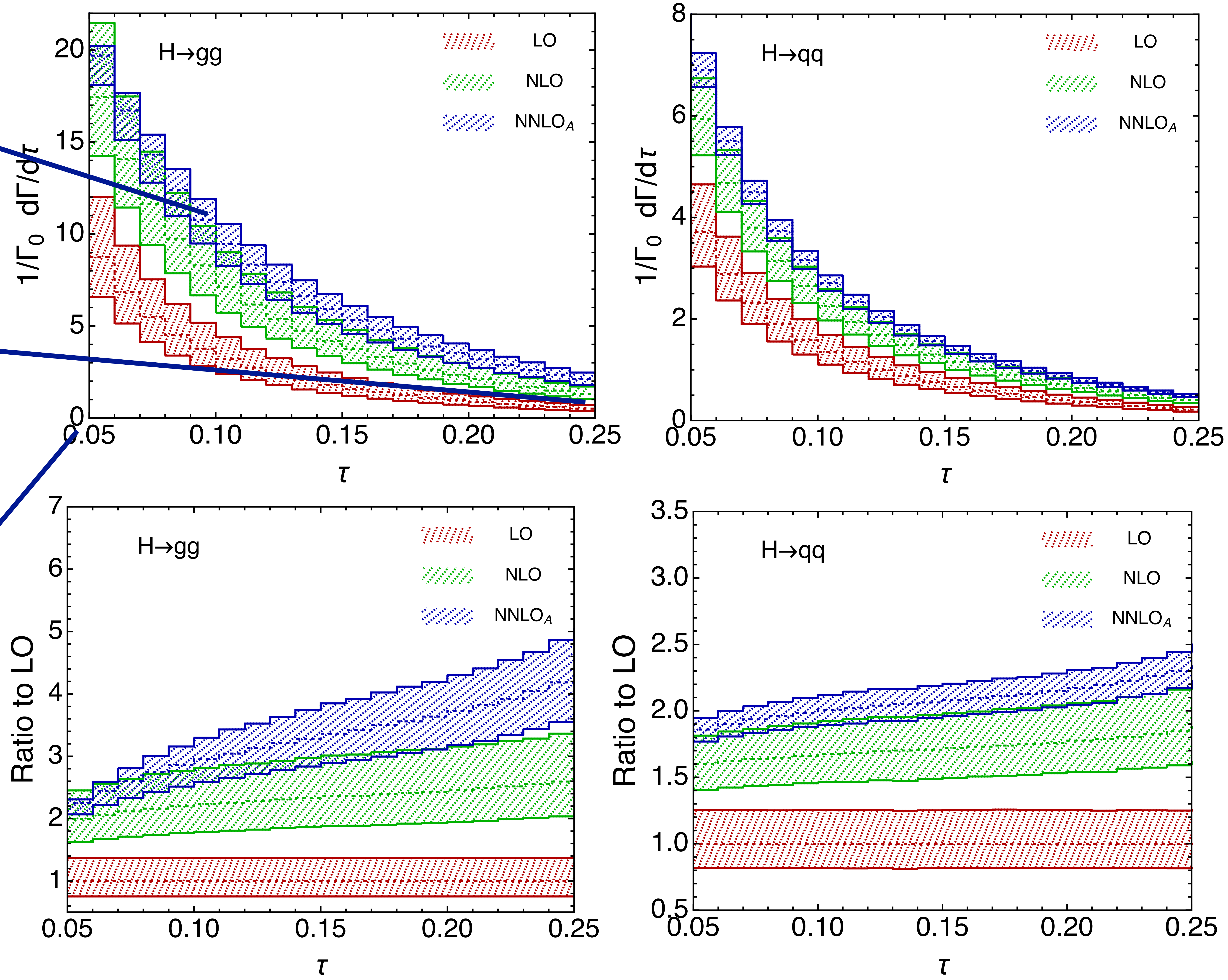
Fixed-order predictions for thrust distribution

Gao, Gong, Ju, **LLY**: 1901.02253

Large corrections, especially in the gluon channel; N^3LO needed?

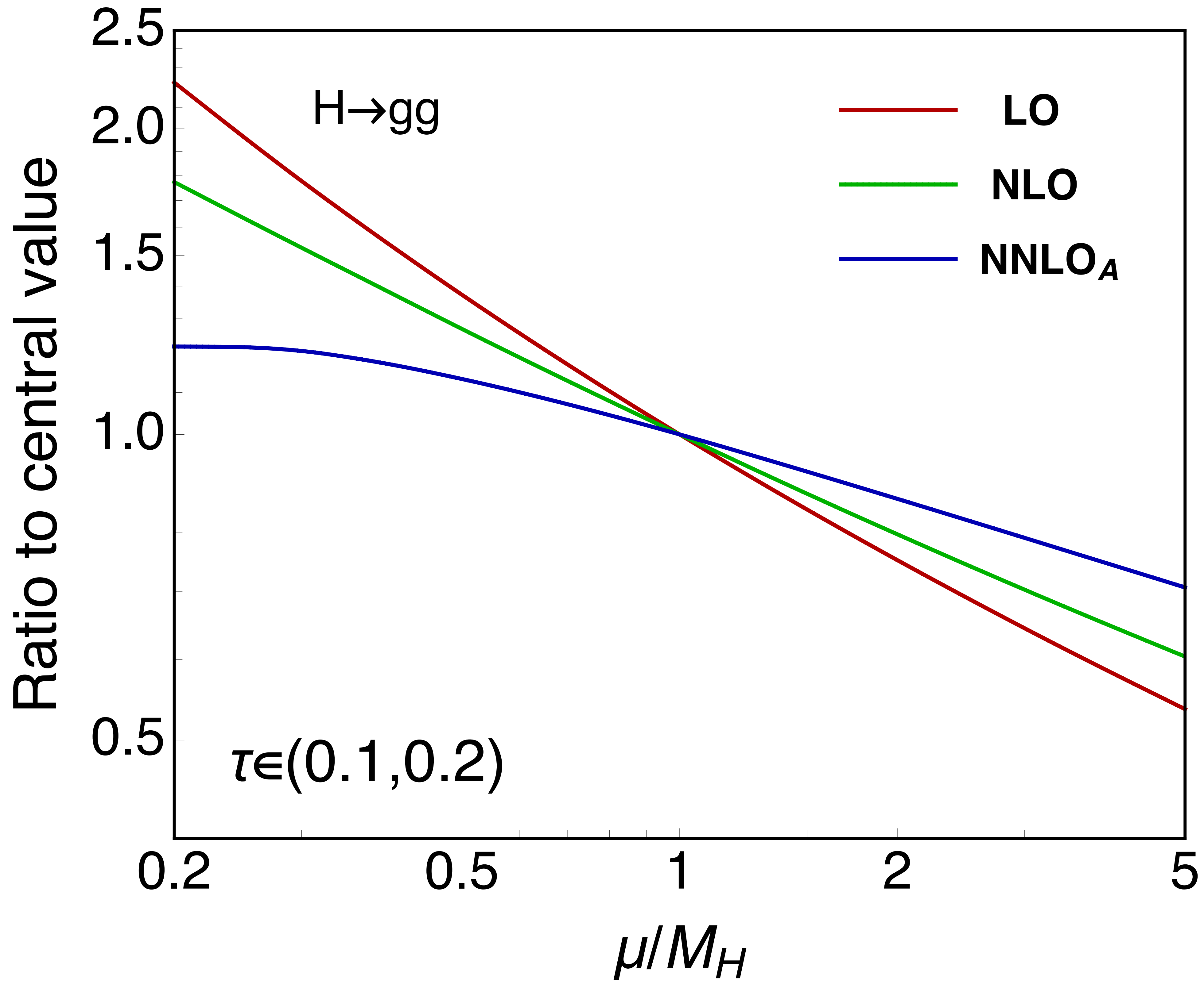
Soft-collinear approximation not valid for larger τ; a full NNLO calculation required!

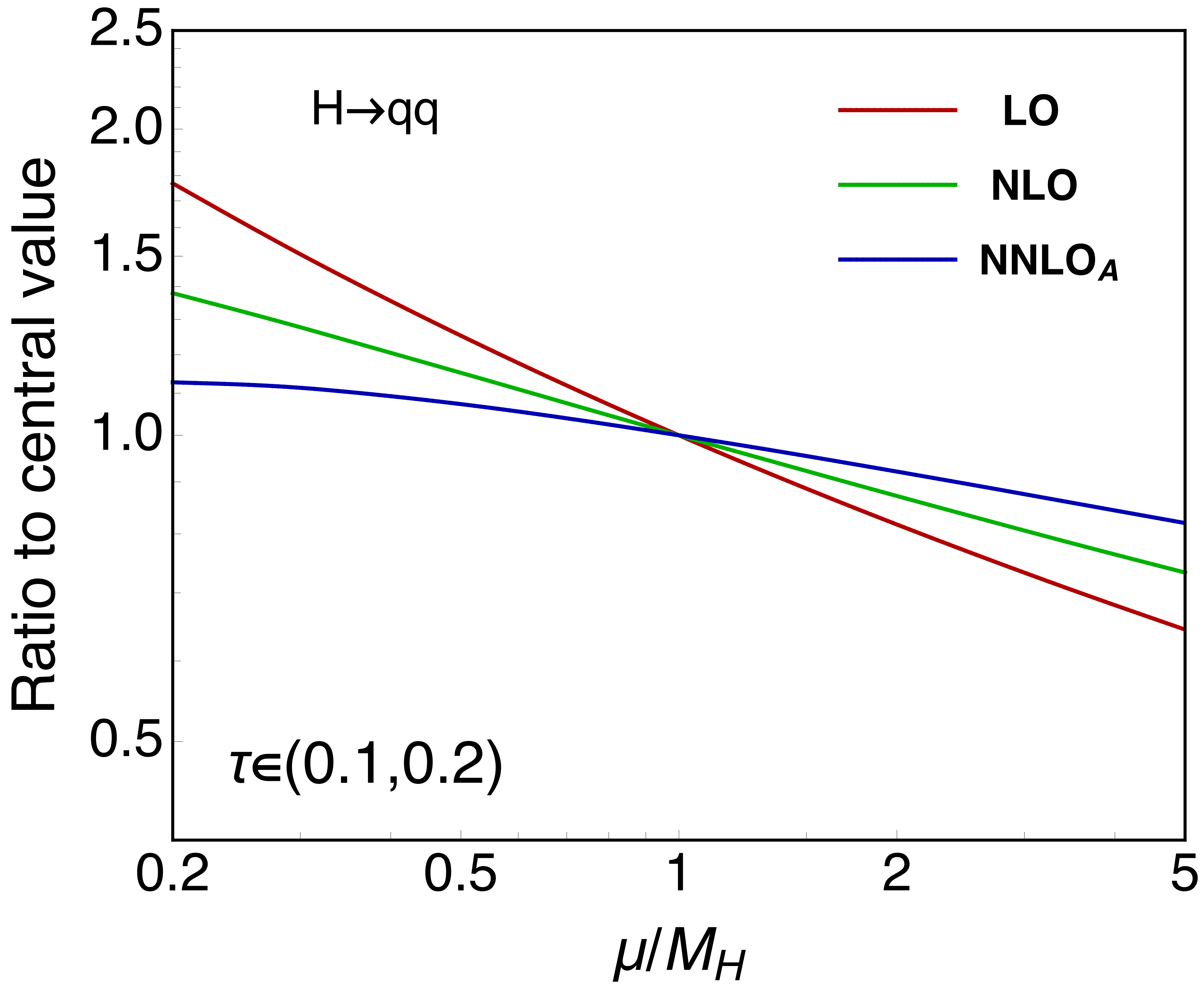
Parton shower and/or resummation needed for smaller τ



Scale dependence

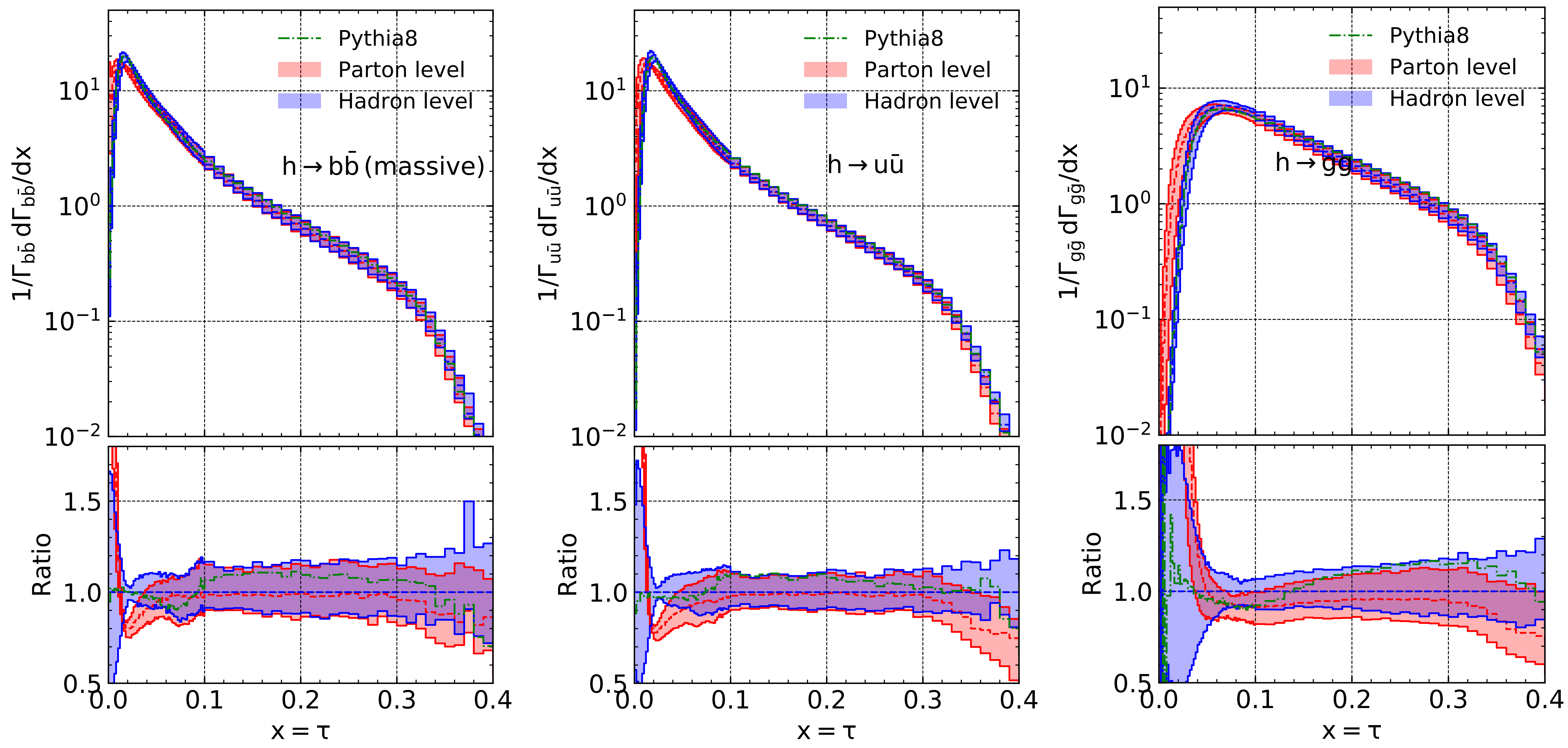
Gao, Gong, Ju, **LLY**: 1901.02253





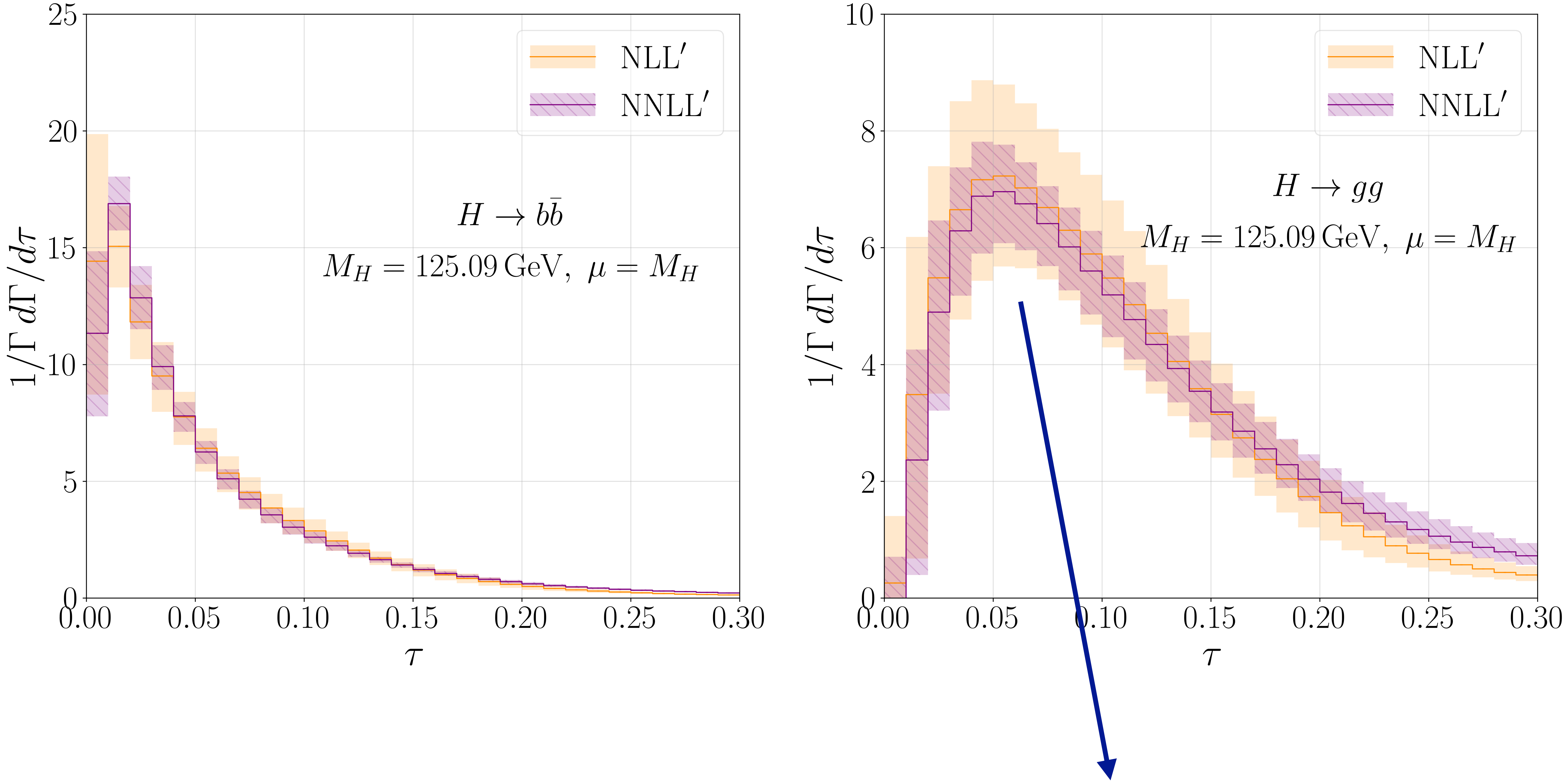
Matched with parton shower

Hu, Sun, Shen, Gao: 2101.08916



Resummed predictions

Alioli et al.: 2009.13533



Large uncertainties in the gluon channel; N^3LL or N^3LL' needed?

Towards N^3LL' thrust resummation

	hard, jet, soft functions	hard, jet, soft anomalous dimensions	cusp anomalous dimension, beta function
NNLL'	2-loop	2-loop	3-loop
N3LL	2-loop	3-loop	4-loop
N3LL'	3-loop	3-loop	4-loop
		available	available

available except the non-logarithmic term of the 3-loop soft function

The 3-loop soft function

The non-logarithmic term of the 3-loop soft function for quarks was extracted from the numeric result of EERAD3

$$c_3^S = 2s_3 + 691 = -19988 \pm 1440\,(\text{stat.}) \pm 4000\,(\text{syst.})$$

Brüser, Liu, Stahlhofen: 1804.09722

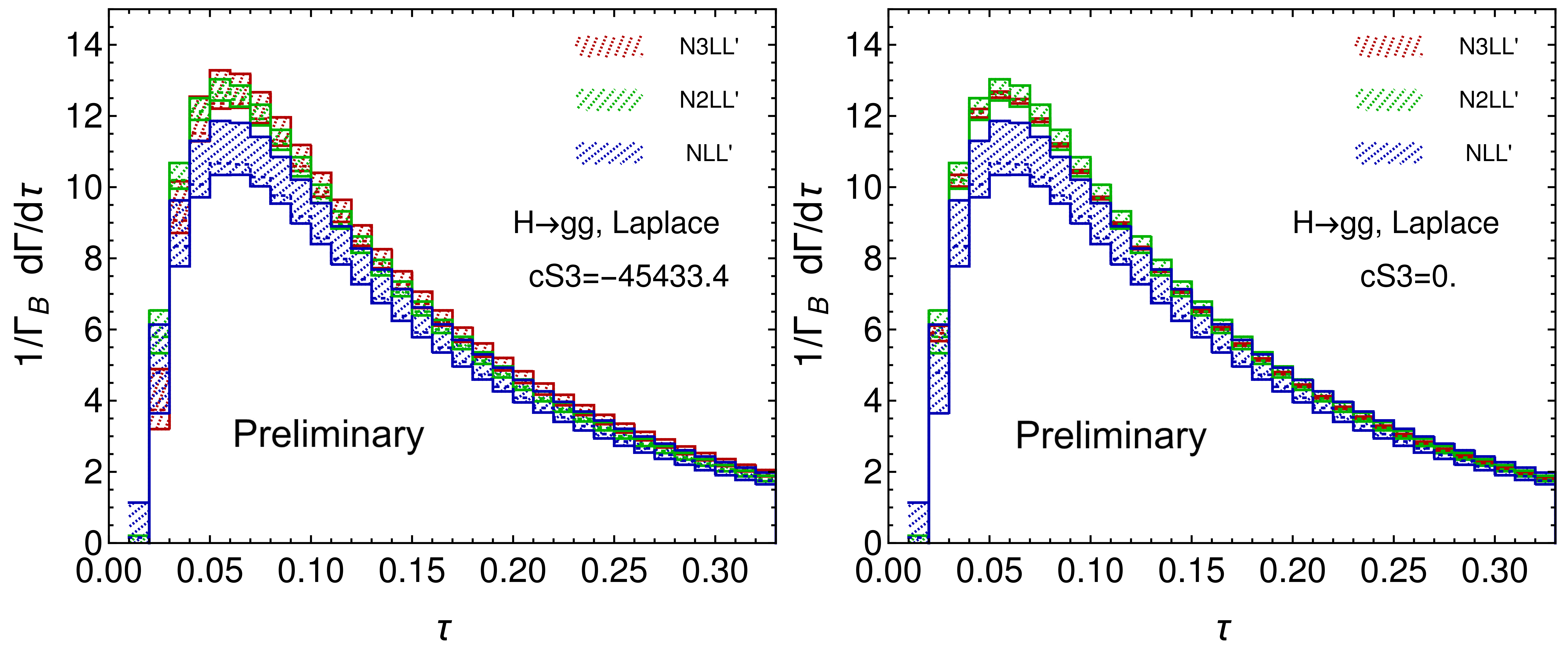
With a Casimir scaling , the corresponding term for gluons

$$c_3^S \sim -45000$$

A rather large constant term, one might worry about convergence!

Especially it multiplies $\alpha_s(\mu_s)$ at the low scale $\mu_s \sim \tau m_H$

Towards N³LL' thrust resummation



Preliminary result shows that the non-logarithmic term of the 3-loop soft function has a large impact!

We may want to know its precise value…

Thank you!