Automation of 2-loop Amplitude Calculations

Max-Planck-Institut für Physik
(Werner-Heisenberg-Institut)

Stephen Jones

Borowka, Greiner, Heinrich, Kerner, Schlenk, Schubert, Zirke

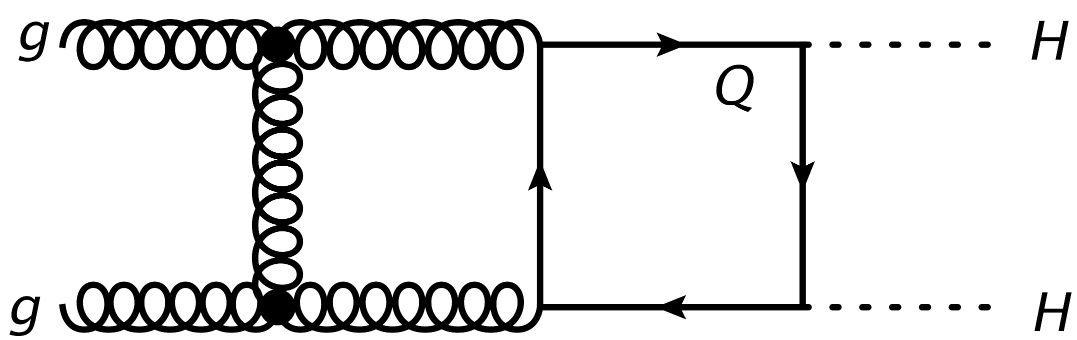
arXiv:1604.06447 [hep-ph]
CPC 196 (2015) 470, arXiv:1502.06595 [hep-ph]

MAX-PLANCK-GESELLSCHAFT

higgstools

Overview

Will discuss method recently used to compute $gg \to HH$ at NLO (2-loop) with full top mass dependence



Our result was presented on Tuesday by M. Kerner

1. What extent can these methods be automated: `GoSam-XLoop`
2. Some details of numerical integration: `SecDec`

Method

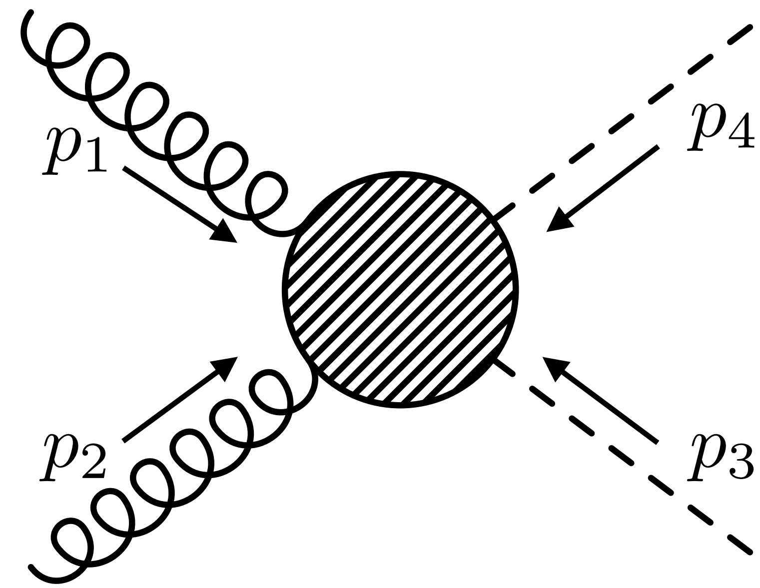
We follow a ***fairly* traditional method**:

1. Decompose into Form Factors & Construct Projectors
2. Generate Feynman Diagrams
3. Apply Projectors & Compute Bare Amplitude
4. Renormalize
5. Integral Reduction
6. Apply Subtraction & Compute Real Radiation
7. ***Numerically*** Compute Master Integrals ← **Not currently known analytically**
8. Generate Events & Compute (Differential) Cross-Section

Note: $gg \to HH$ is a loop induced process, real radiation & subtraction is a solved problem (**Huge simplification**)
Catani, Seymour 96

Form Factor Decomposition

Form Factor decomposition is a standard procedure for loop amplitudes



Expose tensor structure: $\mathcal{M} = \epsilon^1_\mu \epsilon^2_\nu \mathcal{M}^{\mu\nu}$

Decompose:

$$
\begin{aligned}
\mathcal{M}^{\mu\nu} = a_{00} g^{\mu\nu} &+ \cancel{a_{11} p_1^\mu p_1^\nu + a_{12} p_1^\mu p_2^\nu + a_{13} p_1^\mu p_3^\nu} \\
&+ a_{21} p_2^\mu p_1^\nu + a_{22} p_2^\mu p_2^\nu + a_{23} p_2^\mu p_3^\nu \\
&+ a_{31} p_3^\mu p_1^\nu + a_{32} p_3^\mu p_2^\nu + a_{33} p_3^\mu p_3^\nu
\end{aligned}
$$

a_{ij} **functions of Mandelstams +** D

p_i **linearly indep.**

Transversity: $\boxed{g(p_1): \;\; \epsilon^1_\mu p_1^\mu = 0}$ $\boxed{g(p_2): \;\; \epsilon^2_\nu p_2^\nu = 0}$

Ward/Gauge: ${p_1}_\mu \mathcal{M}^{\mu\nu} = 0$, ${p_2}_\nu \mathcal{M}^{\mu\nu} = 0$ gives further identities

Form Factor Decomposition (II)

Form Factors (Contain integrals)

$$\boxed{\mathcal{M}^{\mu\nu} = F_1(\hat{s},\hat{t},m_h^2,m_t^2,D)T_1^{\mu\nu} + F_2(\hat{s},\hat{t},m_h^2,m_t^2,D)T_2^{\mu\nu}}$$

(Tensor) Basis

Construct Projectors (we use CDR $D = 4 - 2\epsilon$):

$$P_1^{\mu\nu} = \frac{1}{4}\frac{D-2}{D-3}T_1^{\mu\nu} - \frac{1}{4}\frac{D-4}{D-3}T_2^{\mu\nu}$$

$$P_2^{\mu\nu} = -\frac{1}{4}\frac{D-4}{D-3}T_1^{\mu\nu} + \frac{1}{4}\frac{D-2}{D-3}T_2^{\mu\nu}$$

Same Basis as amplitude

Compute:

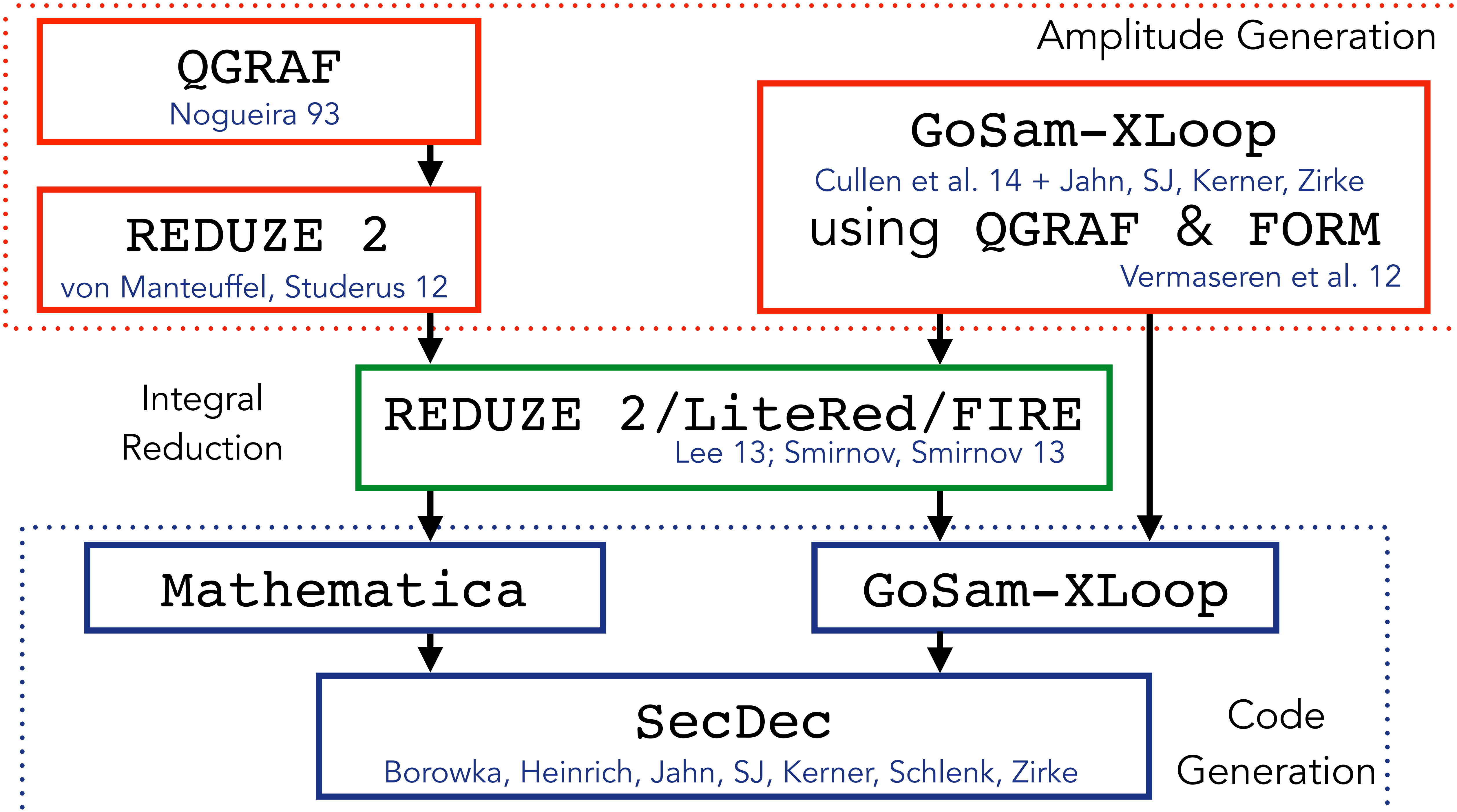
$$P_1^{\mu\nu}\mathcal{M}_{\mu\nu} = F_1(\hat{s},\hat{t},m_h^2,m_t^2,D)$$

$$P_2^{\mu\nu}\mathcal{M}_{\mu\nu} = F_2(\hat{s},\hat{t},m_h^2,m_t^2,D)$$

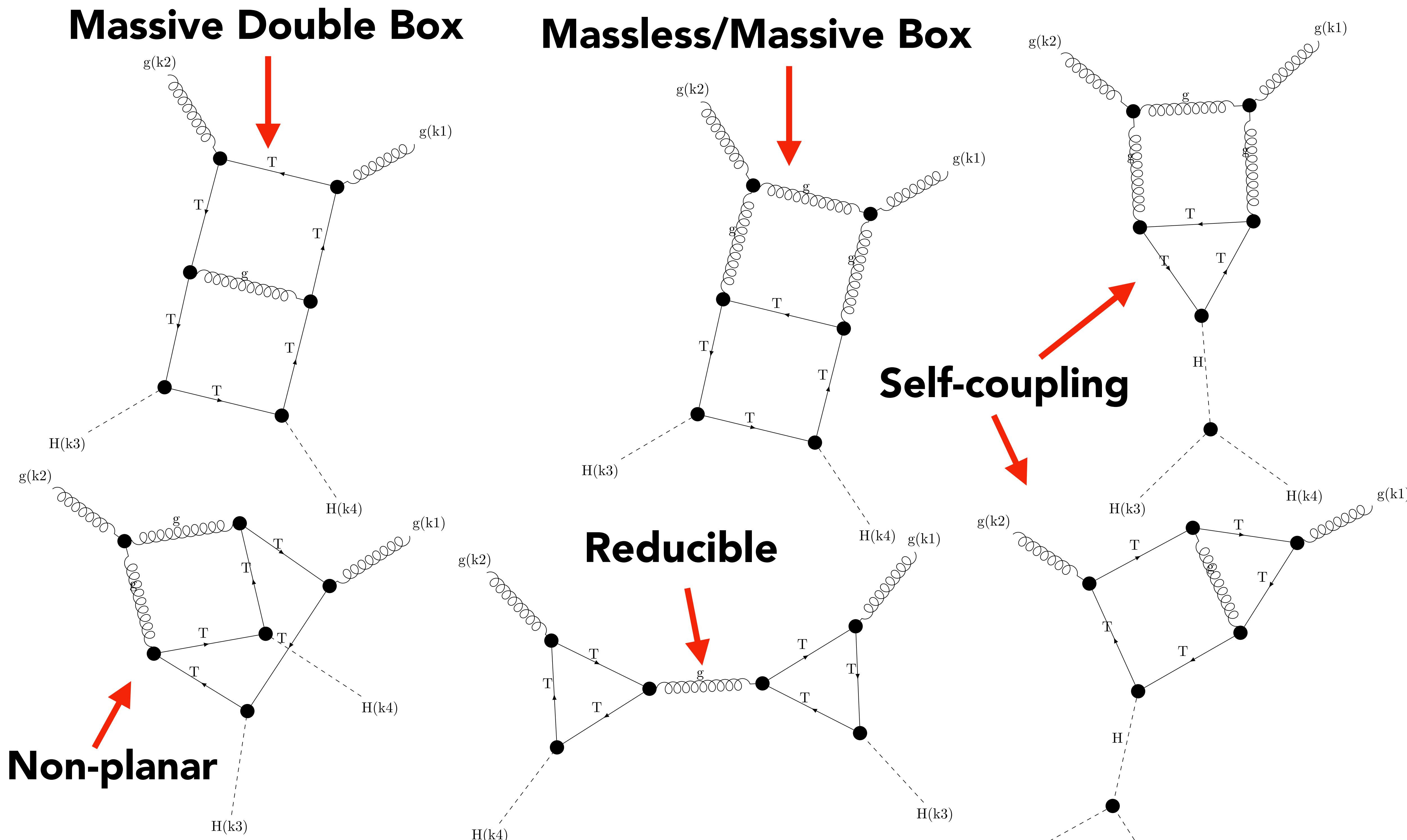
Projectors constructed/ input by hand

Virtual MEs: Tool Chain

Partial cross-check: 2 Implementations



Generate & Compute Diagrams



Total: 122 Diagrams

Integral Reduction

Integral Families input by hand:

Very easy to generate **some** integral family

Seems harder to generate a **good** integral family

Scalar products:

$$S = \frac{l(l+1)}{2} + lm \qquad \begin{array}{ll} l = 2 & \text{\# Loops} \\ m = 3 & \text{\# L.I External momenta} \end{array} \qquad \boxed{S = 9}$$

Choose 8 Integral families with 9 propagators each

Integrals	1-loop	2-loop
Direct	63	9865
+ Symmetries	21	1601
+ IBPs	8	~260-270 (currently 327)

Reduction with `REDUZE 2`

$(r = 8, s = 4)$

from # inverse propagators appearing in problem

Non-planar integrals can be computed **without** reduction

Numerical Master Integrals

To evaluate Master Integrals we use **SecDec** which implements Sector Decomposition

Collaboration: Borowka, Heinrich, Jahn, SJ, Kerner, Schlenk, Zirke

Completely automated procedure

Sector Decomposition

1) Feynman Parametrise integral and compute momentum integrals

$$\boxed{G = (-1)^{N_\nu} \frac{\Gamma(N_\nu - LD/2)}{\prod_{j=1}^{N} \Gamma(\nu_j)} \int_0^\infty \prod_{j=1}^{N} \mathrm{d}x_j \; x_j^{\nu_j - 1} \delta(1 - \sum_{i=1}^{N} x_i) \frac{\mathcal{U}^{N_\nu - (L+1)D/2}(\vec{x})}{\mathcal{F}^{N_\nu - LD/2}(\vec{x}, s_{ij})}}$$

Here $\mathcal{U}, \mathcal{F}$ are 1st, 2nd Symanzik Polynomials

We have exchanged L momentum integrals for N parameter integrals

Sector Decomposition

2) After integrating out δ we are faced with integrals of the form:

$$G_i = \int_0^1 \left(\prod_{j=1}^{N-1} \mathrm{d}x_j x_j^{\nu_j - 1} \right) \frac{\mathcal{U}_i(\vec{x})^{\mathrm{expo}\mathcal{U}(\epsilon)}}{\mathcal{F}_i(\vec{x}, s_{ij})^{\mathrm{expo}\mathcal{F}(\epsilon)}}$$

Powers depending on ϵ

Polynomials in F.P

Which may contain overlapping singularities which appear when several $x_j \to 0$ simultaneously (corresponding to UV/IR singularities)

Sector decomposition maps each integral into integrals of the form:

$$G_{ik} = \int_0^1 \left(\prod_{j=1}^{N-1} \mathrm{d}x_j x_j^{a_j - b_j \epsilon} \right) \frac{\mathcal{U}_{ik}(\vec{x})^{\mathrm{expo}\mathcal{U}(\epsilon)}}{\mathcal{F}_{ik}(\vec{x}, s_{ij})^{\mathrm{expo}\mathcal{F}(\epsilon)}}$$

Singularity structure can be read off

$$\mathcal{U}_{ik}(\vec{x}) = 1 + u(\vec{x})$$

$$\mathcal{F}_{ik}(\vec{x}) = -s_0 + f(\vec{x})$$

$u(\vec{x}), f(\vec{x})$ **have no constant term**

Hepp 66; Denner, Roth 96; Binoth, Heinrich 00

Sector Decomposition (II)

3) Expand in ϵ (simple case $a = -1$):

$$\int_0^1 \mathrm{d}x^{-1-b\epsilon} g(x) = \frac{g(0)}{-b\epsilon} + \int_0^1 \mathrm{d}x x^{-b\epsilon} \left[\frac{g(x) - g(0)}{x}\right]$$

Finite

Poles

Note: \`subtraction' of $g(0)$

By Definition: $g(0) \neq 0, g(0)$ finite

4) Numerically integrate

`SecDec` supports: numerators, inverse propagators, \`\`dots'', physical kinematics, arbitrary loops & legs (within reason)

Soper 00; Nagy, Soper 06; Borowka 14

Key Point: Sector Decomposed integrals can be expanded in ϵ and numerically integrated

Rank 1 Shifted Lattices

$\mathcal{O}(n^{-1})$ algorithm for numerical integration: Review: Dick, Kuo, Sloan 13

$$I_s[f] \equiv \int_{[0,1]^s} \mathrm{d}^s x\, f(\vec{x})$$

$$f : \mathbb{R}^s \to \mathbb{C}$$

$$I_s[f] \approx \bar{Q}_{s,n,m}[f] \equiv \frac{1}{m} \sum_{k=1}^{m} \frac{1}{n} \sum_{i=0}^{n-1} f\left(\left\{ \frac{i\vec{z}}{n} + \vec{\Delta}_k \right\} \right)$$

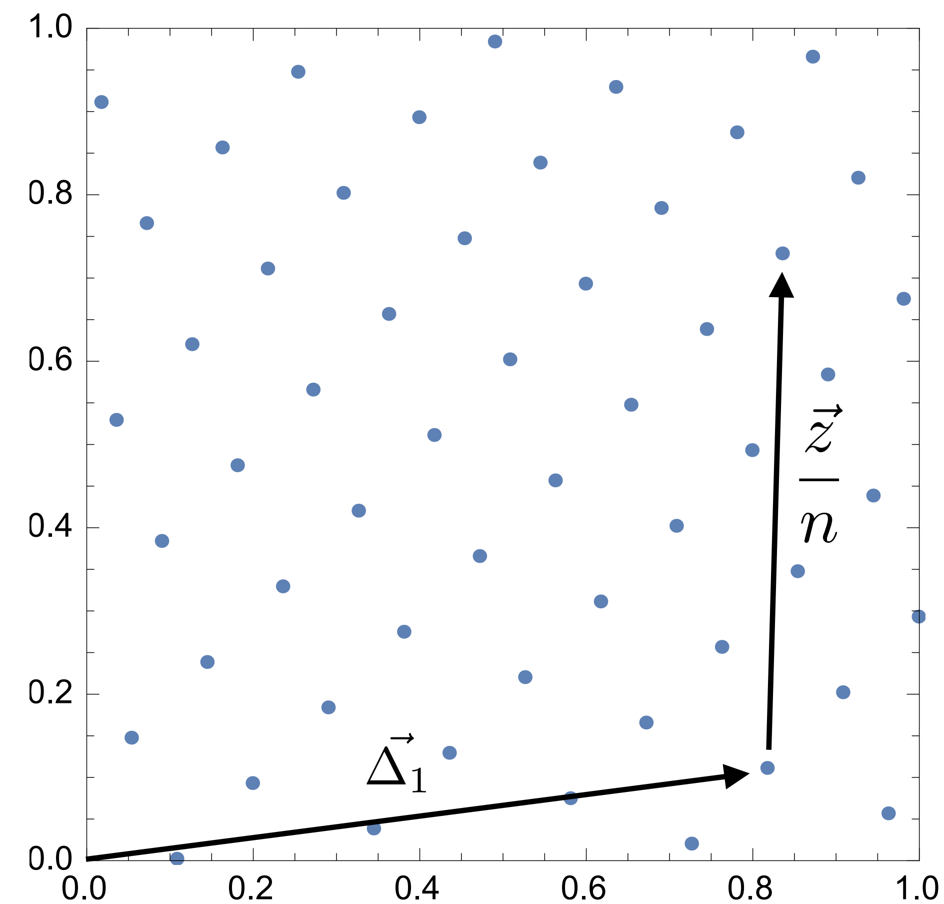
$\vec{z}$ - Generating vec.

$\vec{\Delta}_k$ - Random shift vec.

$\{\}$ - Fractional part

n - # Lattice points

m - # Random shifts

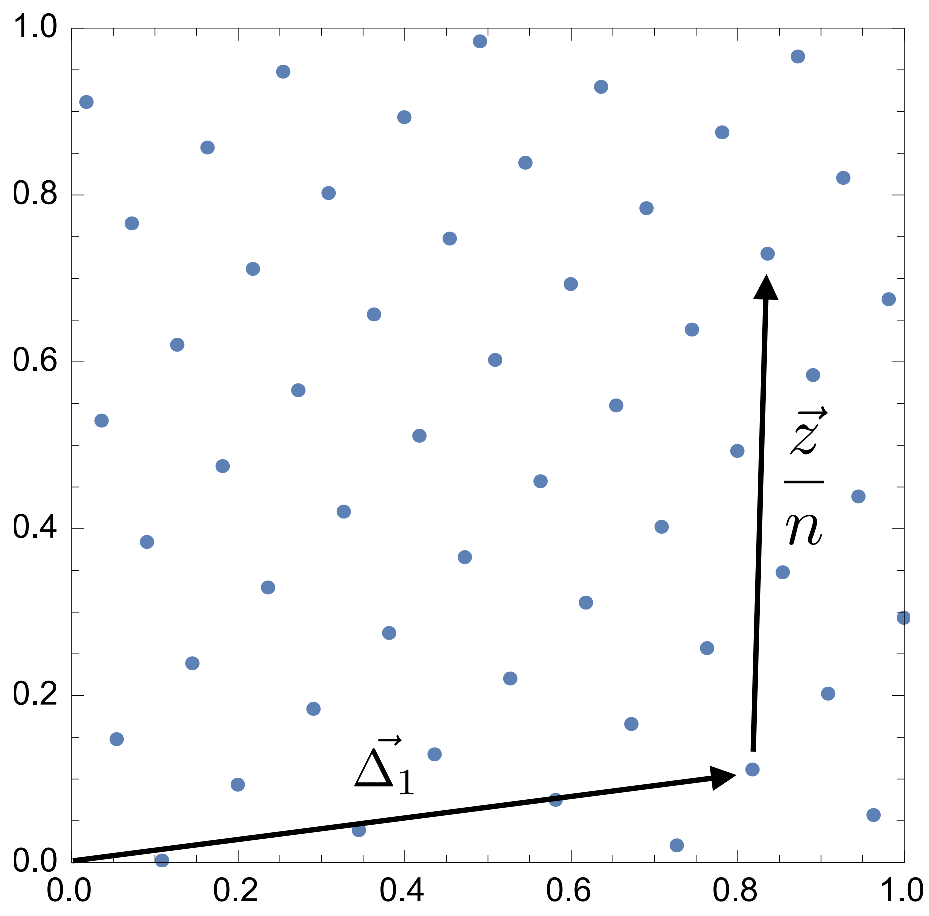


Generating vector $\vec{z}$ precomputed for a **fixed** number of lattice points, chosen to minimise worst-case error Nuyens 07

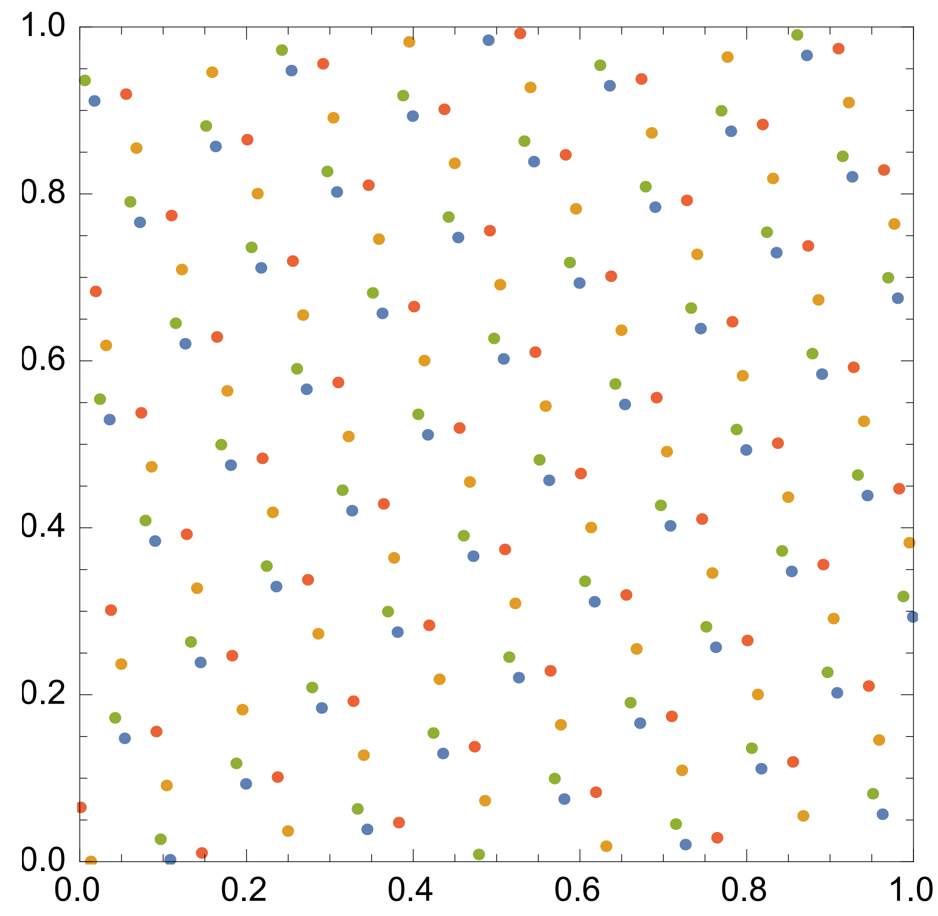
Rank 1 Shifted Lattices

Unbiased error estimate computed from random shifts:

$$\mathrm{Var}[\bar{Q}_{s,n,m}[f]] \approx \frac{1}{m(m-1)} \sum_{k=1}^{m} (Q_{s,n,k} - \bar{Q}_{s,n,m})^2$$



1 Shift

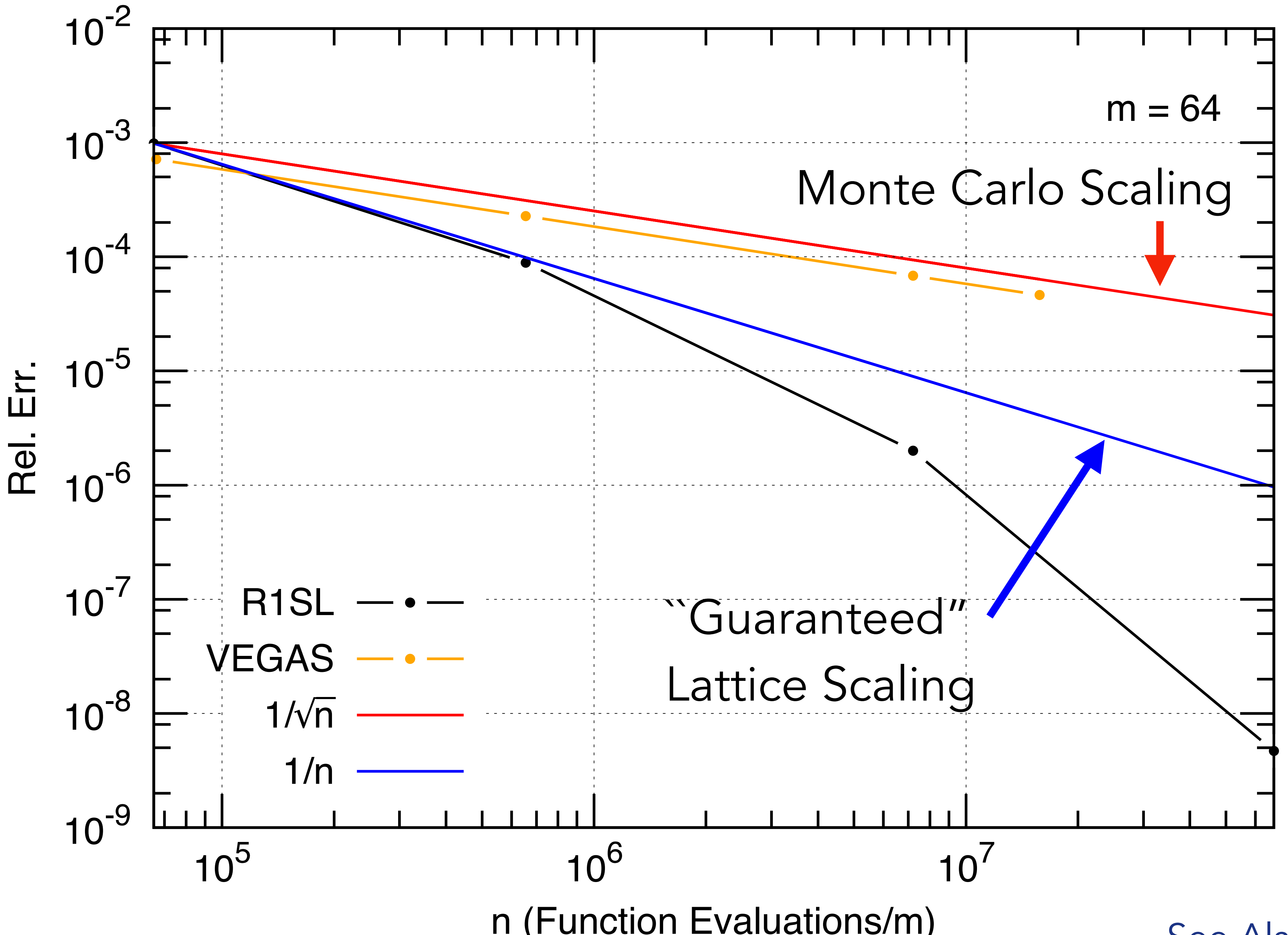


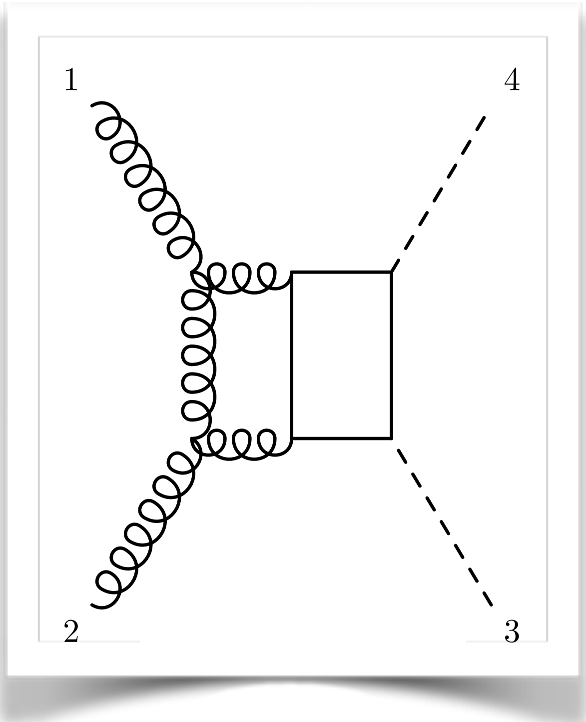
4 Shifts

Typically 10-50 shifts, production run: 20 shifts

R1SL: Algorithm Performance

Example: Rel. Err. of one sector of sector decomposed loop integral





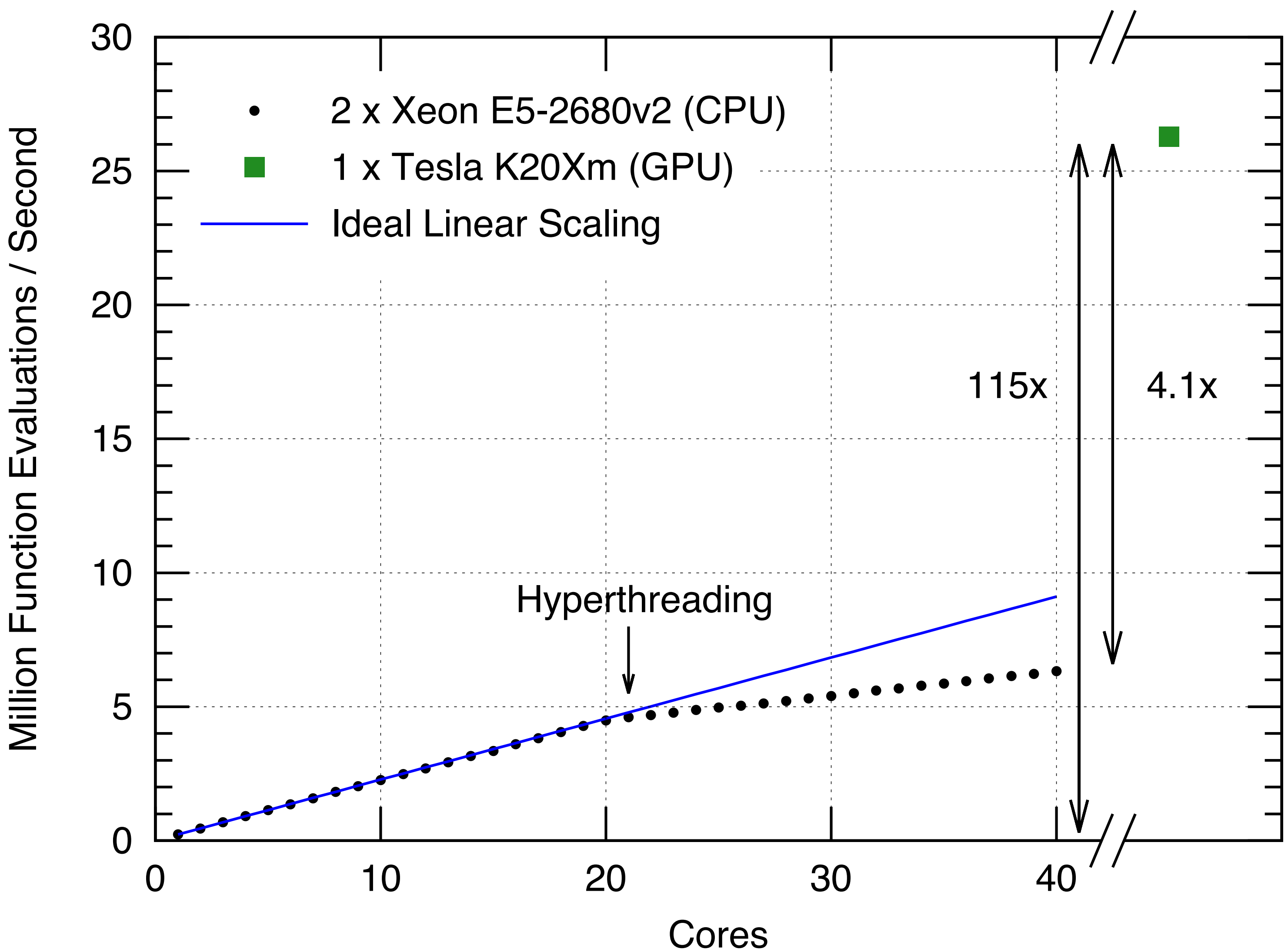
6 dimensional numerical integral

See Also: Li, Wang, Yan, Zhao 15

R1SL: Implementation Performance

Accuracy limited primarily by number of function evaluations

Implemented in OpenCL 1.1 for CPU & GPU, generate points on GPU/CPU core, sum blocks of points (reduce memory usage/transfers)



2 CPUs (20 Cores + HT)

1 GPU

n	CPU (s)	GPU(s)	C/G
655357	6.63	1.60	4.1
7208951	72.3	16.4	4.4
67264993	674.2	152.2	4.4

Integral Reduction & Finite Basis

One **important** extra step: (Partly) transform to (Quasi-) Finite basis, also handled by `REDUZE 2`

Panzer 14; von Manteuffel, Panzer, Schabinger 15

- (elegantly) Extracts subdivergences into coefficient of integrals
- May require integrals in shifted dimension or with propagators raised to higher powers

⟶ **See talk of A. von Manteuffel**

Finite basis is a sensible basis choice for numerical evaluation

- (Quasi-) Finite integrals need no `subtraction' as singularities already resolved
- Practically; can choose basis such that relatively few orders of ϵ are required for most complicated integrals

Amplitude Structure

Writing integrals with r propagators and s inverse propagators as

Arbitrary scale

$$I_{r,s}(\hat{s},\hat{t},m_h^2,m_t^2) = (M^2)^{-L\epsilon}(M^2)^{2L-r+s} I_{r,s}\left(\frac{\hat{s}}{M^2},\frac{\hat{t}}{M^2},\frac{m_h^2}{M^2},\frac{m_t^2}{M^2}\right)$$

Renormalized form factors for $gg \to HH$ with $\overline{\mathrm{MS}}$ scheme strong coupling a and OS scheme gluon field, top-quark mass:

$$\mathrm{F} = a\mathrm{F}^{(1)} + a^2(\delta Z_A + \delta Z_a)\mathrm{F}^{(1)} + a^2 \delta m_t^2 \mathrm{F}^{ct,(1)} + a^2\mathrm{F}^{(2)} + O(a^3)$$

$$\mathrm{F}^{(1)} = \left(\frac{\mu_R^2}{M^2}\right)^{\epsilon}\left[b_0^{(1)} + b_1^{(1)}\epsilon + b_2^{(1)}\epsilon^2 + \mathcal{O}(\epsilon^3)\right]$$ 1-loop

$$\mathrm{F}^{ct,(1)} = \left(\frac{\mu_R^2}{M^2}\right)^{\epsilon}\left[c_0^{(1)} + c_1^{(1)}\epsilon + \mathcal{O}(\epsilon^2)\right]$$ Mass Counter-Terms

$$\mathrm{F}^{(2)} = \left(\frac{\mu_R^2}{M^2}\right)^{2\epsilon}\left[\frac{b_{-2}^{(2)}}{\epsilon^2} + \frac{b_{-1}^{(2)}}{\epsilon} + b_0^{(2)} + \mathcal{O}(\epsilon)\right]$$ 2-loop

Scale variations do not require any re-computation of **red** terms

Amplitude Structure (II)

Form factors are sums of rational functions multiplied by integrals that depend on ratios of the scales s, t, m_h^2, m_t^2 and the arbitrary scale M^2

$$
\begin{aligned}
F^{(L)} &= \sum_i \left[\left(\sum_j C_{i,j}^{(L)} \epsilon^j \right) \cdot \left(\sum_k I_{i,k}^{(L)} \epsilon^k \right) \right] \\
&= \quad \epsilon^{-2} \left[C_{1,-2}^{(L)} \cdot I_{1,0}^{(L)} + C_{1,-1}^{(L)} \cdot I_{1,-1}^{(L)} + \ldots \right] \\
&\quad + \epsilon^{-1} \left[C_{1,-1}^{(L)} \cdot I_{1,0}^{(L)} + \ldots \right] + \ldots
\end{aligned}
$$

compute only once

Additionally, all L-loop form factors are computed simultaneously without re-evaluating common integrals

Note: $gg \to HH$ is a loop induced process, real subtraction and mass factorisation contained in $\mathbf{I}, \mathbf{P}, \mathbf{K}$ operators (not discussed here)

Catani, Seymour 96

SecDec as a Library

Single program to compute **all** coefficients & integrals to obtain **amplitude** to given accuracy

```
Amplitude si(epsrel,devinds,crossings);

// coeffs/coeff1.cpp
si.addTerm(
            string("ReduzeF1L2_230000010ord0"),
            ReduzeF1L2_230000010ord0nfunc(),
            crossing,
            &ReduzeF1L2_230000010ord0Integrand,
            &ReduzeF1L2_230000010ord0findoptlam,
            ReduzeF1L2_230000010ord0ndim(),
            params,
            termCoeff1
            );

// coeffs/coeff204.cpp
si.addTerm(
            string("ReduzeF3L2diminc2_131010100ord1"),
            ReduzeF3L2diminc2_131010100ord1nfunc(),
            crossing,
            &ReduzeF3L2diminc2_131010100ord1Integrand,
            &ReduzeF3L2diminc2_131010100ord1findoptlam,
            ReduzeF3L2diminc2_131010100ord1ndim(),
            params,
            termCoeff2
            );

si.optimizeLambda();
si.integrate();
```

SecDec as a Library

Single program to compute **all** coefficients & integrals to obtain **amplitude** to given accuracy

desired precision

list of GPUs & CPUs

```
Amplitude si(epsrel,devinds,crossings);

// coeffs/coeff1.cpp
si.addTerm(
           string("ReduzeF1L2_230000010ord0"),
           ReduzeF1L2_230000010ord0nfunc(),
           crossing,
           &ReduzeF1L2_230000010ord0Integrand,
           &ReduzeF1L2_230000010ord0findoptlam,
           ReduzeF1L2_230000010ord0ndim(),
           params,
           termCoeff1
           );

// coeffs/coeff204.cpp
si.addTerm(
           string("ReduzeF3L2diminc2_131010100ord1"),
           ReduzeF3L2diminc2_131010100ord1nfunc(),
           crossing,
           &ReduzeF3L2diminc2_131010100ord1Integrand,
           &ReduzeF3L2diminc2_131010100ord1findoptlam,
           ReduzeF3L2diminc2_131010100ord1ndim(),
           params,
           termCoeff2
           );

si.optimizeLambda();
si.integrate();
```

SecDec as a Library

Single program to compute **all** coefficients & integrals to obtain **amplitude** to given accuracy

desired precision

list of GPUs & CPUs

```
Amplitude si(epsrel,devinds,crossings);

// coeffs/coeff1.cpp
si.addTerm(
           string("ReduzeF1L2_230000010ord0"),
           ReduzeF1L2_230000010ord0nfunc(),
           crossing,
           &ReduzeF1L2_230000010ord0Integrand,
           &ReduzeF1L2_230000010ord0findoptlam,
           ReduzeF1L2_230000010ord0ndim(),
           params,
           termCoeff1
           );

// coeffs/coeff204.cpp
si.addTerm(
           string("ReduzeF3L2diminc2_131010100ord1"),
           ReduzeF3L2diminc2_131010100ord1nfunc(),
           crossing,
           &ReduzeF3L2diminc2_131010100ord1Integrand,
           &ReduzeF3L2diminc2_131010100ord1findoptlam,
           ReduzeF3L2diminc2_131010100ord1ndim(),
           params,
           termCoeff2
           );

si.optimizeLambda();
si.integrate();
```

name & reference to integrand to integrate

$(\hat{s}, \hat{t}, m_t^2, m_h^2)$

vector of coefficients $C_{1,-2}, C_{1,-1}, \ldots$ for all Form Factors, evaluated at this phase-space point

SecDec as a Library

Single program to compute **all** coefficients & integrals to obtain **amplitude** to given accuracy

desired precision

list of GPUs & CPUs

```
Amplitude si(epsrel,devinds,crossings);

// coeffs/coeff1.cpp
si.addTerm(
            string("ReduzeF1L2_230000010ord0"),
            ReduzeF1L2_230000010ord0nfunc(),
            crossing,
            &ReduzeF1L2_230000010ord0Integrand,
            &ReduzeF1L2_230000010ord0findoptlam,
            ReduzeF1L2_230000010ord0ndim(),
            params,
            termCoeff1
            );

// coeffs/coeff204.cpp
si.addTerm(
            string("ReduzeF3L2diminc2_131010100ord1"),
            ReduzeF3L2diminc2_131010100ord1nfunc(),
            crossing,
            &ReduzeF3L2diminc2_131010100ord1Integrand,
            &ReduzeF3L2diminc2_131010100ord1findoptlam,
            ReduzeF3L2diminc2_131010100ord1ndim(),
            params,
            termCoeff2
            );

si.optimizeLambda();
si.integrate();
```

name & reference to integrand to integrate

$(\hat{s}, \hat{t}, m_t^2, m_h^2)$

vector of coefficients $C_{1,-2}, C_{1,-1}, \ldots$ for all Form Factors, evaluated at this phase-space point

Find contour deformation (physical region) in parallel for all integrals in amplitude

Computes integrals in parallel on GPUs & CPUs. Dynamically adjusts # points per sector to reduce amplitude error

Phase-Space Sampling

Phase-space implemented by hand

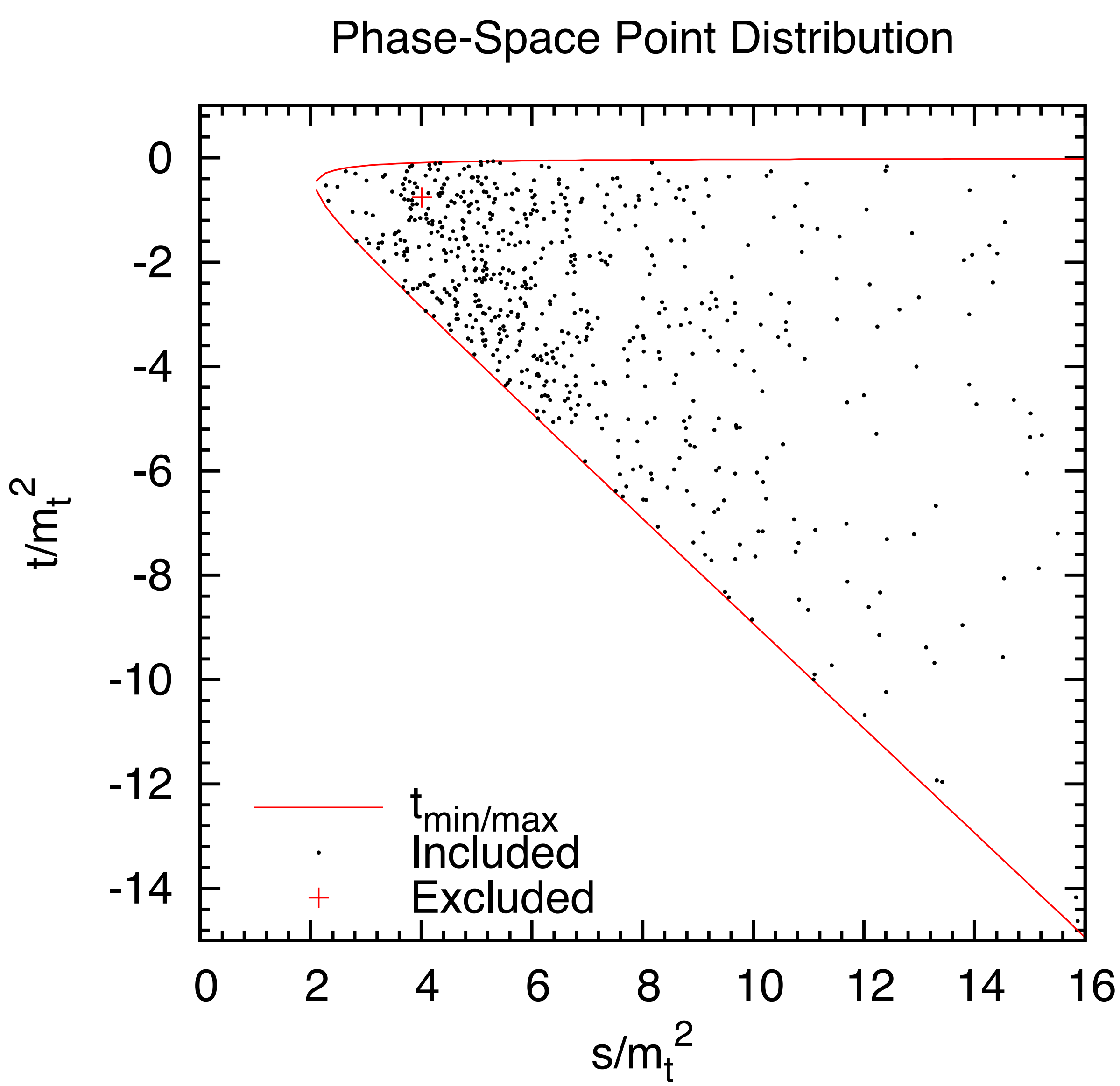
limited to 2-3 w/ 2 massive particles

Events for virtual:

1) VEGAS algorithm applied to LO matrix element $\mathcal{O}(100k)$ events computed

2) Using LO events unweighted events generated using accept/reject method $\mathcal{O}(30k)$ events remain

3) Randomly select 666 Events (woops), compute at NLO, exclude 1



Note: No grids used either for integrals or phase-space

Timings

Bottleneck: Integral reduction, tried `Fire`, `Litered`, `Reduze 2`

Smirnov, Smirnov 13; Lee 13; von Manteuffel, Studerus 12

Note: Not a criticism, we are not using these tools smartly or on ideal hardware, this problem is **HARD** for these tools

Hardware (numerics): ~16 Dual Nvidia Tesla K20X GPGPU Nodes

Thanks: MPCDF

Median GPU time per PS point: 2 hours

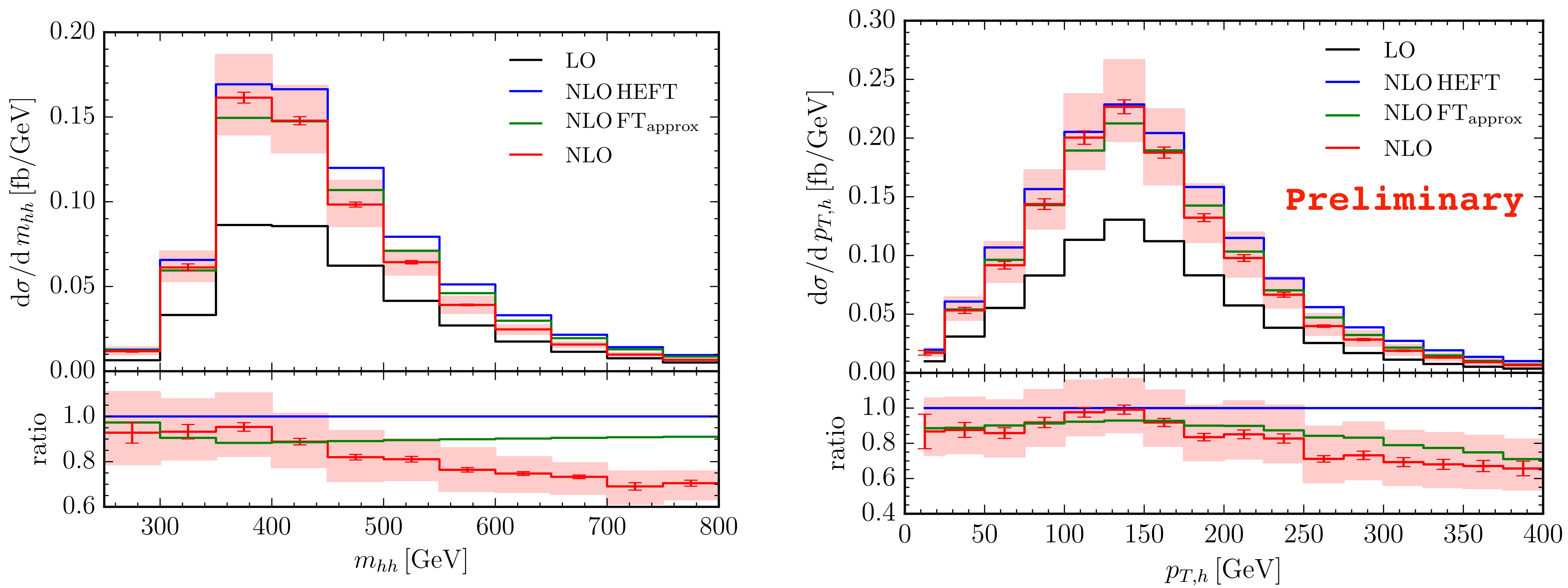
Total compute time used: 4680 GPU Hours

Wall time: **6 days**

Key Point: Even after the advances discussed here numerical integration is slow but our setup can scale to use the available compute resources

Results

$gg \to HH$ @ NLO



$$\sigma^{NLO} = 32.80^{+13\%}_{-12\%}\,\text{fb} \pm 0.4\%\,(\text{stat.}) \pm 0.1\%\,(\text{int.})$$

Born-improved HEFT: $\sigma^{NLO}_{HEFT} = 38.32^{+18\%}_{-15\%}\,\text{fb}$

LO: $\sigma^{LO} = 19.85^{+28\%}_{-21\%}\,\text{fb}$

Result presented on Tuesday by M. Kerner

Conclusion

`GoSam-XLoop`

- Aims to be a flexible framework for generating amplitudes (mostly) relevant for massive/multi-scale problems
- Tightly coupled to integral reduction programs
- In lieu of a general `master' integral library beyond one-loop, interfaced to `SecDec` for automatic numeric evaluation of **amplitudes**

Ongoing/Future

- More details of our method and our calculation will be discussed in an upcoming paper
- Implement obvious code improvements
- (Future) Fully exploit analytically known master integrals, needs searchable database: Loopedia (?)

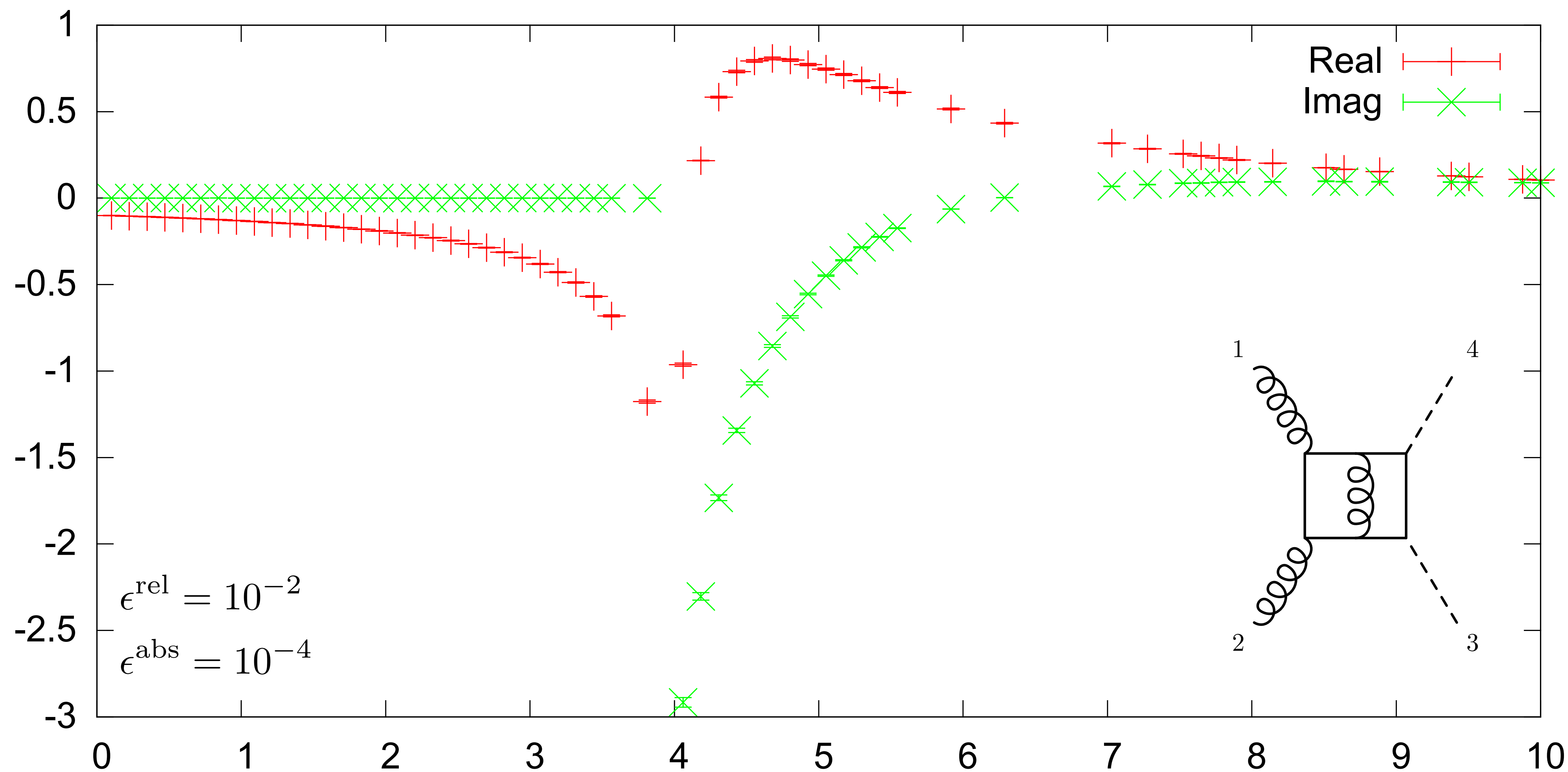
Thank you for listening

Backup

Master Integrals (Numerical)

SecDec (https://secdec.hepforge.org)

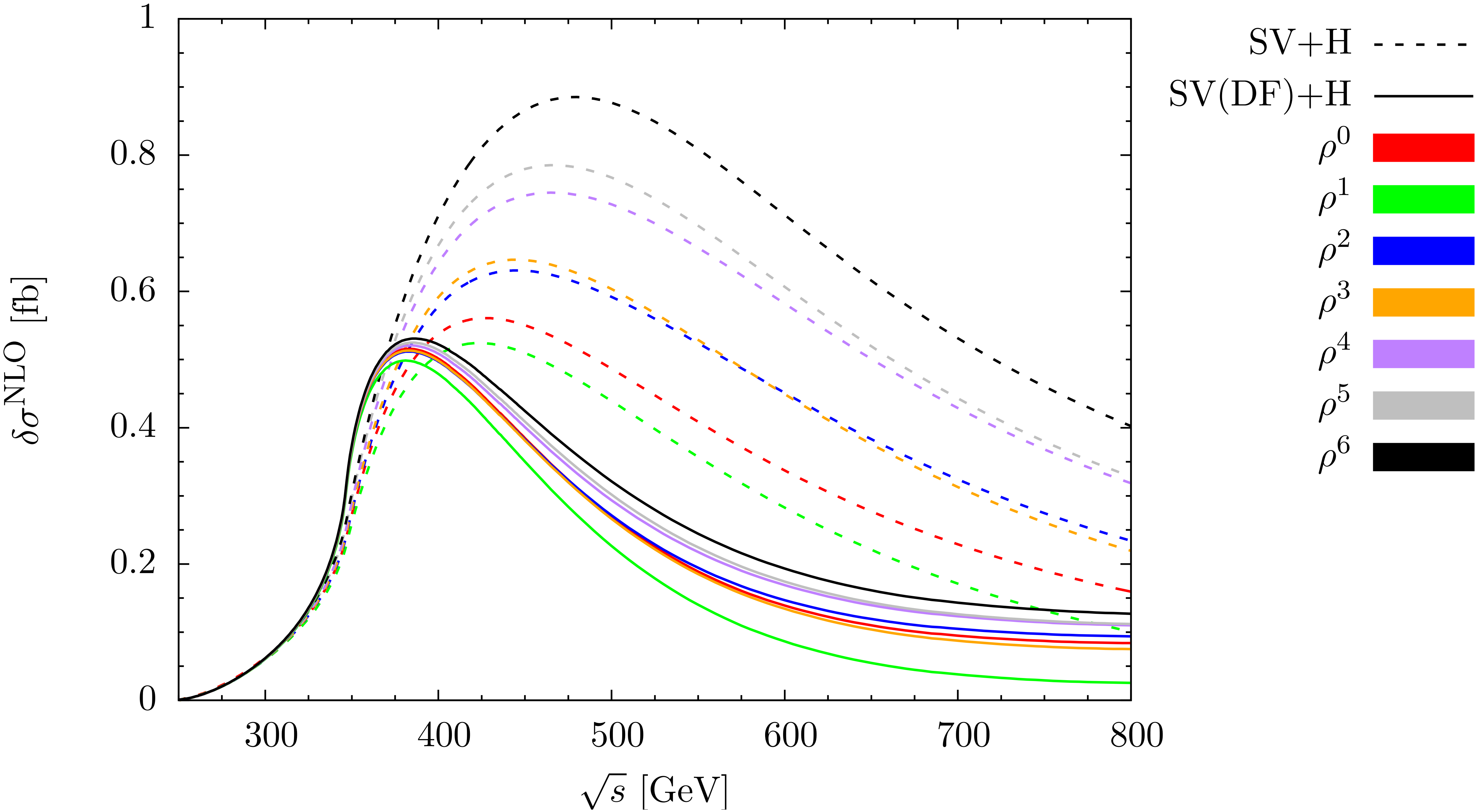
Evaluate Dimensionally regulated parameter integrals numerically



Top Threshold s$_{12}$ (s/m_T^2)

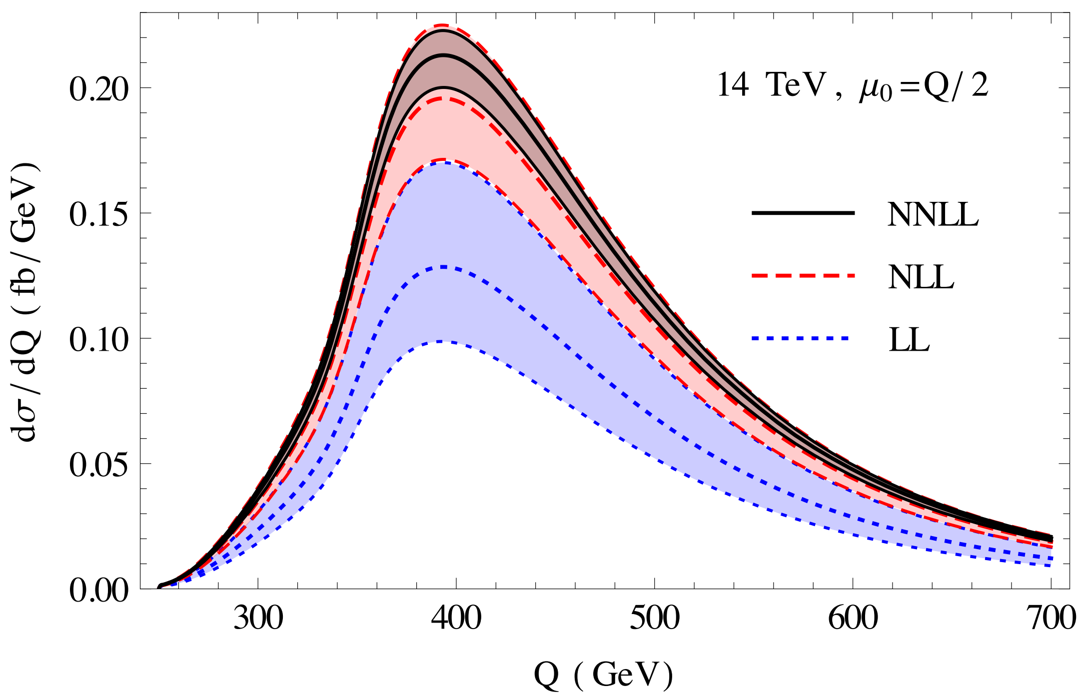
Thanks: Anton Stoyanov

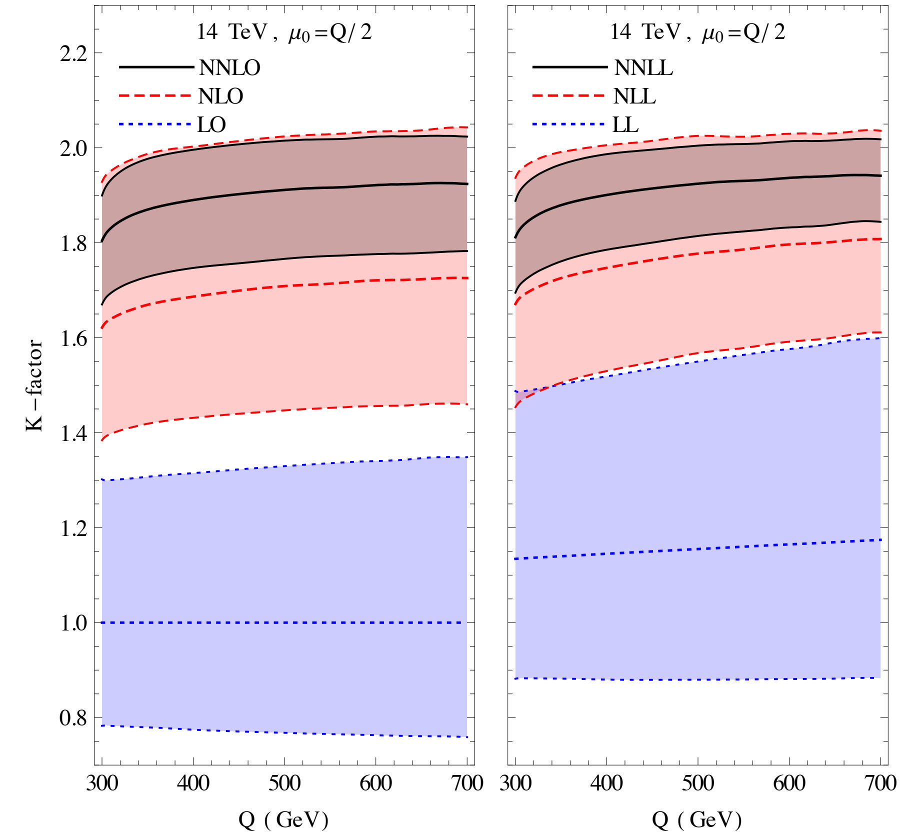
G.H.S Top Mass Expansion



Grigo, Hoff, Steinhauser 15

HEFT NNLO + NNLL

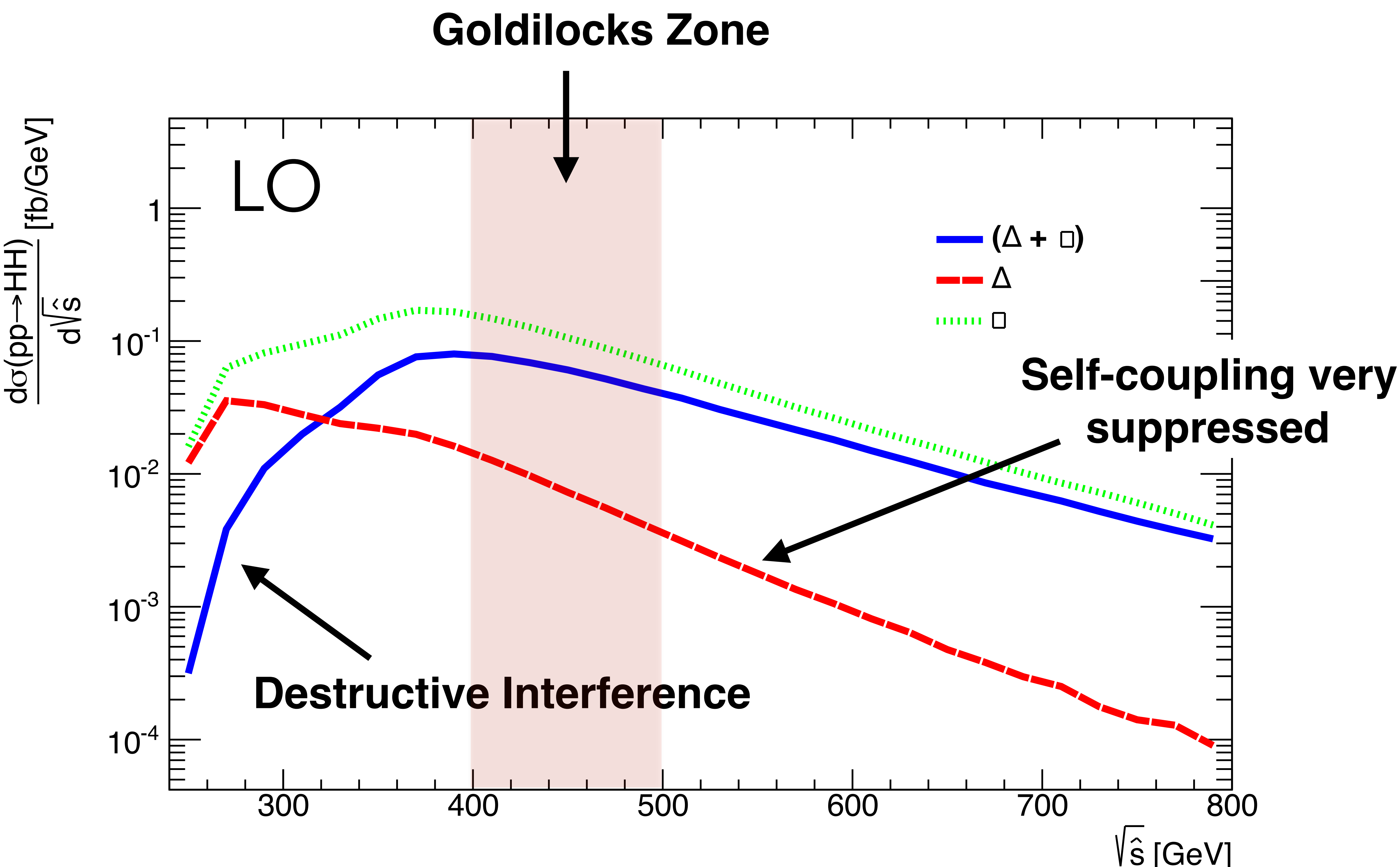




de Florian, Mazzitelli 15

$\mu_0 = Q$	NNLO (fb)	scale unc. (%)	NNLL (fb)	scale unc. (%)	PDF unc. (%)	PDF+α_S unc. (%)
8 TeV	9.92	$+9.3 - 10$	10.8	$+5.4 - 5.9$	$+5.6 - 6.0$	$+9.3 - 9.2$
13 TeV	34.3	$+8.3 - 8.9$	36.8	$+5.1 - 6.0$	$+4.0 - 4.3$	$+7.7 - 7.5$
14 TeV	40.9	$+8.2 - 8.8$	43.7	$+5.1 - 6.0$	$+3.8 - 4.0$	$+7.5 - 7.3$
33 TeV	247	$+7.1 - 7.4$	259	$+5.0 - 6.1$	$+2.2 - 2.8$	$+6.1 - 6.1$
100 TeV	1660	$+6.8 - 7.1$	1723	$+5.2 - 6.1$	$+2.1 - 3.0$	$+5.7 - 5.8$
$\mu_0 = Q/2$	NNLO (fb)	scale unc. (%)	NNLL (fb)	scale unc. (%)	PDF unc. (%)	PDF+α_S unc. (%)
8 TeV	10.8	$+5.7 - 8.5$	11.0	$+4.0 - 5.6$	$+5.8 - 6.1$	$+9.6 - 9.3$
13 TeV	37.2	$+5.5 - 7.6$	37.4	$+4.2 - 5.8$	$+4.1 - 4.3$	$+7.8 - 7.6$
14 TeV	44.2	$+5.5 - 7.6$	44.5	$+4.2 - 5.9$	$+3.9 - 4.1$	$+7.6 - 7.4$
33 TeV	264	$+5.3 - 6.6$	265	$+4.6 - 6.1$	$+2.4 - 2.7$	$+6.3 - 6.1$
100 TeV	1760	$+5.3 - 6.7$	1762	$+4.9 - 6.4$	$+2.2 - 3.1$	$+6.2 - 7.0$

Self-Coupling Sensitivity



Slawinska, van den Wollenberg, Eijk, Bentvelsen 14

Uncertainties

Total Cross Section:

Born Improved NLO HEFT

Scale $\mu_0 = \mu_R = \mu_F = M_{HH}$, Variation: $[\frac{\mu_0}{2}, 2\mu_0]$

Some arguments for switching to $\mu_0 = M_{HH}/2$ **(account for NNLL?)**

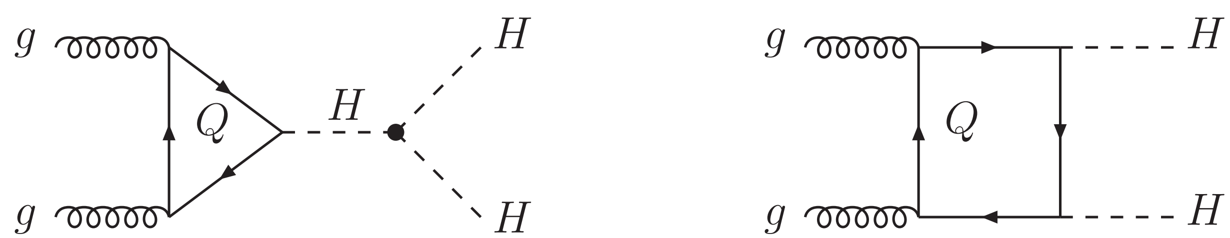
Scale	15-20%
PDF + α_s	6-7%
EFT (NLO)	~10%
Total	30-40%

See: Eg… Baglio, Djouadi et al. 12

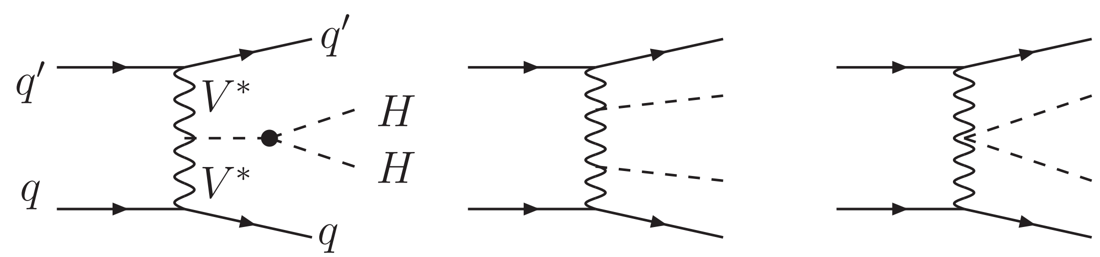
Production Channels

$\sigma(pp \to HH + X)$ @ 13TeV

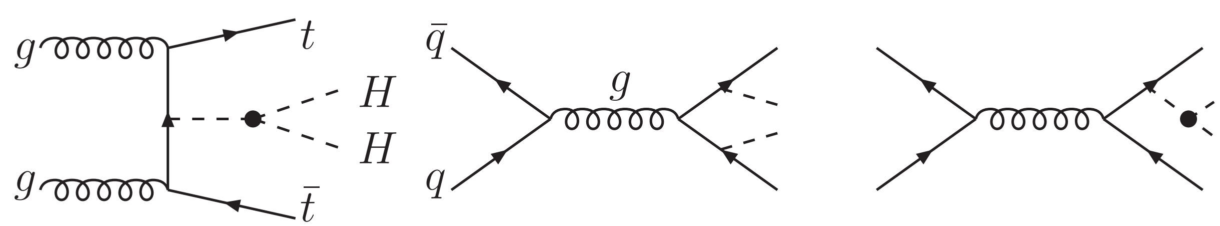
Gluon Fusion



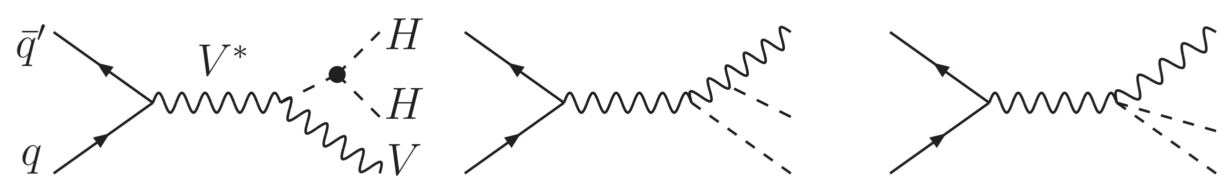
Vector Boson Fusion (VBF)



Associated top pair



Double Higgs-strahlung

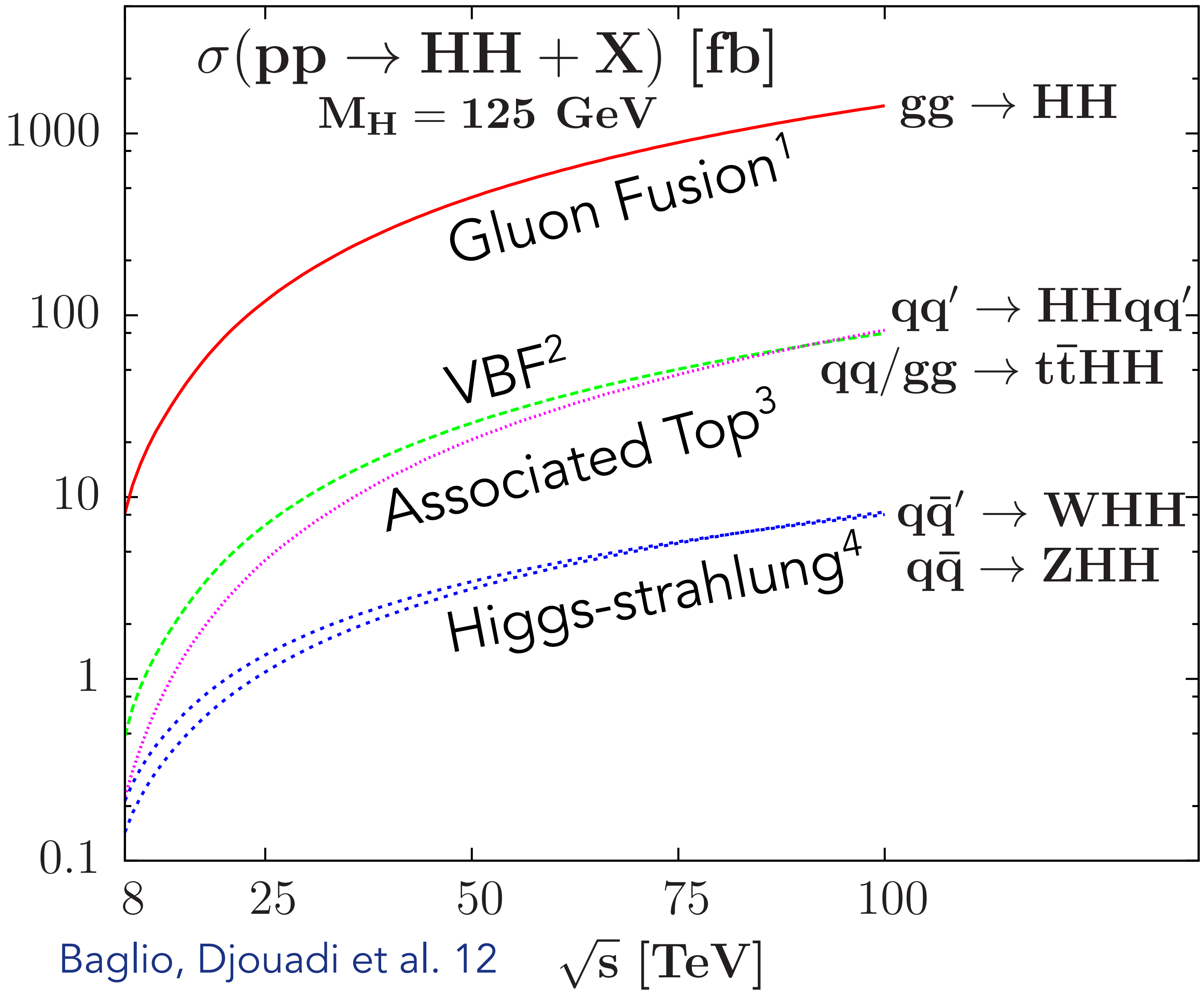


…

Baglio, Djouadi et al. 12

Production Channels

$$\sigma(pp \to HH + X) \sim \frac{1}{1000}\sigma(pp \to H + X)$$



Baglio, Djouadi et al. 12

[1] NLO QCD HEFT, `HPAIR`
Plehn, Spira, Zerwas 96, 98; Dawson et al. 98

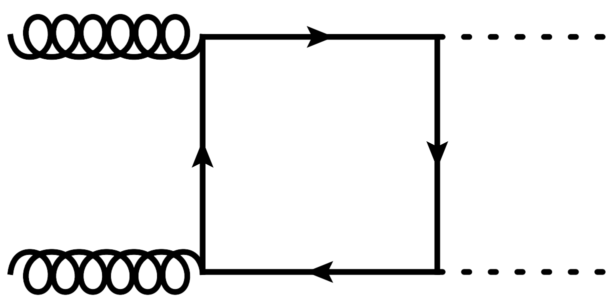
[2] NLO QCD, `VBFNLO`
Baglio, Djouadi et al. 12

[3] LO QCD (NLO, `aMC@NLO`)
Frederix, Frixione et al. 14

[4] NNLO QCD
Baglio, Djouadi et al. 12

Gluon Fusion

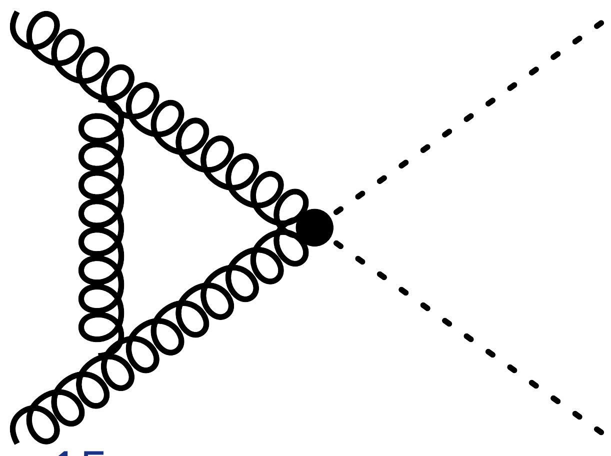
1. LO (1-loop), Dominated by top (bottom contributes ~1%)
   Glover, van der Bij 88



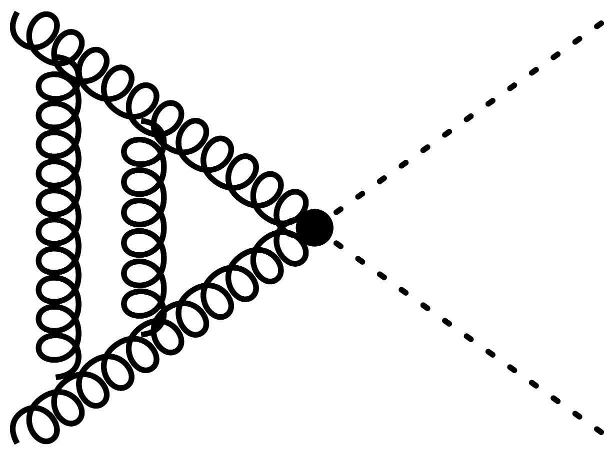
2. Born Improved NLO H(iggs)EFT $m_T \to \infty$ **K≈ 2**
   Plehn, Spira, Zerwas 96, 98; Dawson, Dittmaier, Spira 98

A. Including m_T in Real radiation **-10%**
   Maltoni, Vryonidou, Zaro 14

B. Including $\mathcal{O}(1/m_T^{12})$ terms in Virtual MEs **±10%**
   Grigo, Hoff, Melnikov, Steinhauser 13; Grigo, Hoff 14; Grigo, Hoff, Steinhauser 15



3. Born Improved NNLO HEFT +20%
   De Florian, Mazzitelli 13
   Including matching coefficients
   Grigo, Melnikov, Steinhauser 14
   Including terms $\mathcal{O}(1/m_T^{12})$ in Virtual MEs
   Grigo, Hoff, Steinhauser 15
   NNLL + NNLO Matching +9%
   de Florian, Mazzitelli 15



Gluon Fusion (NLO HEFT)

HEFT Valid for: $\sqrt{s} \ll 2m_T$ HH Production for: $2m_H < \sqrt{s}$

Born Improved NLO QCD HEFT

$$\mathrm{d}\sigma^V_{\mathrm{NLO}}(m_T) \approx \mathrm{d}\bar{\sigma}^V_{\mathrm{NLO}}(m_T) \equiv \frac{\mathrm{d}\sigma^V_{\mathrm{NLO}}(m_T \to \infty)}{\mathrm{d}\sigma^V_{\mathrm{LO}}(m_T \to \infty)} \mathrm{d}\sigma^V_{\mathrm{LO}}(m_T)$$

$$\mathrm{d}\sigma^R(m_T \to \infty)$$

$$K \approx 2$$

A. Maltoni et al.14

$$\mathrm{d}\overline{\sigma}^V(m_T)$$

$$\mathrm{d}\sigma^R(m_T)$$

$$-10\%$$

B. Grigo, Hoff, Steinhauser 15

$$\mathrm{d}\hat{\sigma}(m_T) \equiv \mathrm{d}\sigma_0 + \mathrm{d}\sigma_1 \frac{m_H^2}{m_T^2} + \ldots + \mathrm{d}\sigma_6 \frac{m_H^{12}}{m_T^{12}}$$

$$\mathrm{d}\bar{\sigma}^{SV}_{\mathrm{NLO}}(m_T) \equiv \mathrm{d}\hat{\sigma}^{SV}_{\mathrm{NLO}}(m_T) \frac{\mathrm{d}\sigma^V_{\mathrm{LO}}(m_T)}{\mathrm{d}\hat{\sigma}^V_{\mathrm{LO}}(m_T)}$$

$$\mathrm{d}\bar{\sigma}^{H}_{\mathrm{NLO}}(m_T) \equiv \mathrm{d}\hat{\sigma}^{H}_{\mathrm{NLO}}(m_T) \frac{\sigma^V_{\mathrm{LO}}(m_T)}{\hat{\sigma}^V_{\mathrm{LO}}(m_T)}$$

$$\pm 10\%$$

Shopping List

Virtual MEs (HH):

Channels:

$gg \to HH$ $\quad\quad$ $\cancel{q\bar{q} \to HH}$

Contributes at NNLO

	Diagrams
Tree	0
1-loop	8
2-loop	122

Real Radiation (HH + j...):

Huge simplification!

1-j Channels:

$gg \to HH + g$

$gq \to HH + q$ $\quad\quad$ $g\bar{q} \to HH + \bar{q}$

$q\bar{q} \to HH + g$

	Diagrams
Tree ⊗ Double	0
1-loop ⊗ Single	54+8+8+8

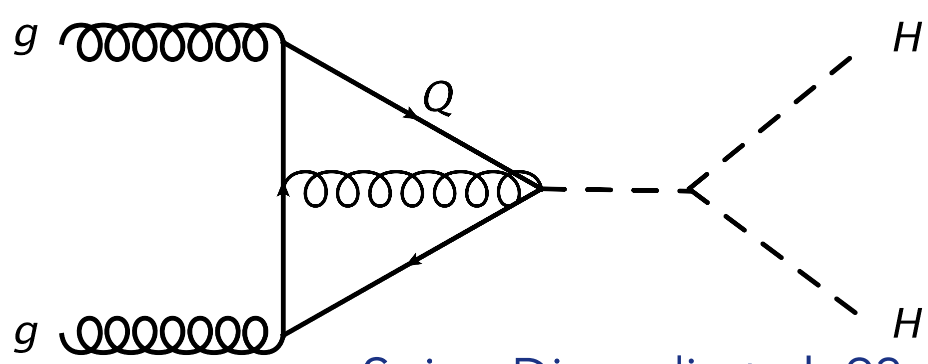
`GoSam` for MEs + Catani-Seymour Dipole Subtraction

Catani, Seymour 96

Backup: Integrals

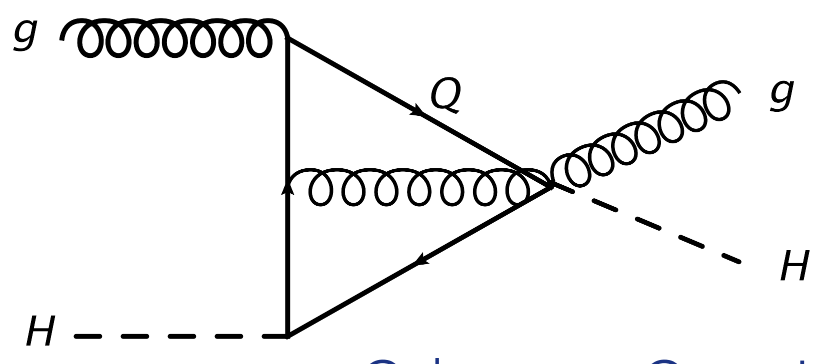
Master Integrals

Known Analytically:



Spira, Djouadi et al. 93, 95;
Bonciani, P. Mastrolia 03,04;
Anastasiou, Beerli et al. 06;

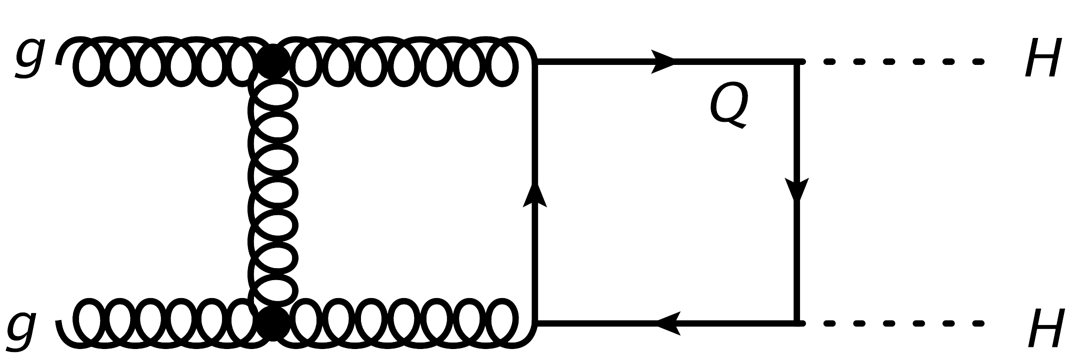
3-point, 1 off-shell leg
HPLs



Gehrmann, Guns, Kara 15

3-point, 2 off-shell legs
Generalized HPLs, 12 Letters

Numeric Evaluation:

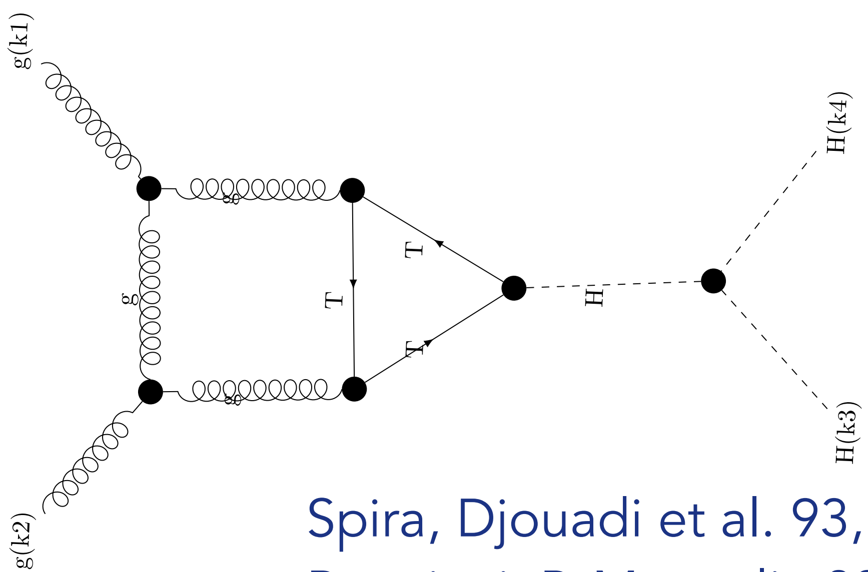


Up to 4-point,
4 scales s, t, m_T^2, m_H^2
`SecDec`

Master Integrals

Double Higgs Production Master Integrals are tough!

- Massive propagators
- Off-shell legs

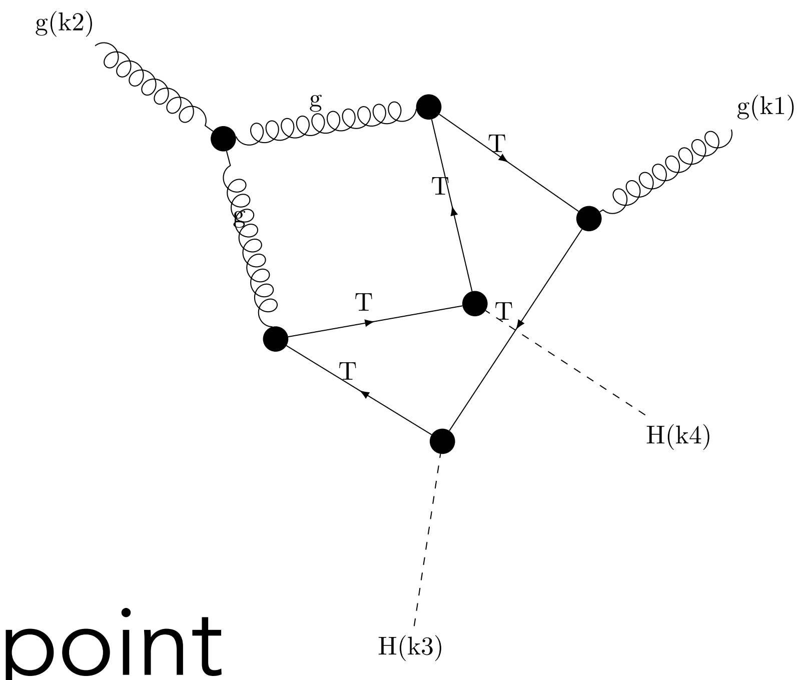


Spira, Djouadi et al. 93, 95;
Bonciani, P. Mastrolia 03,04;
Anastasiou, Beerli et al. 06;
...

3-point

2 scales s, m_T^2

Known analytically



4-point

4 scales s, t, m_T^2, m_H^2

Numerical (`SecDec`)

Backup: Graph Polynomials

Graph Polynomials

Properties:

- Homogenous polynomials in the Feynman Parameters

  $\mathcal{U}(\vec{x})$ is degree L

  $\mathcal{F}(\vec{x}, s_{ij})$ is degree $L+1$

  $\mathcal{F}(\vec{x}, s_{ij}) = \mathcal{F}_0(\vec{x}, s_{ij}) + \mathcal{U}(\vec{x}) \sum_{i=1}^{N} x_i m_i^2$ ← **Internal masses**
- $\mathcal{U}(\vec{x})$ and $\mathcal{F}_0(\vec{x}, s_{ij})$ are linear in each Feynman Parameter

$\mathcal{F}_0(\vec{x}, s_{ij})$ **and** $\mathcal{U}(\vec{x})$ **can be constructed graphically**

We will follow: Bogner, Weinzierl 10

Divergences

From the master formula, 3 possibilities for poles in ϵ to arise:

1. Overall $\Gamma(N_\nu - LD/2)$ diverges (single UV pole)
2. $\mathcal{U}(\vec{x})$ vanishes for some $x = 0$ and has negative exponent (UV sub-divergences)
3. $\mathcal{F}(\vec{x}, s_{ij})$ vanishes on the boundary and has negative exponent (IR divergences)

Outside the Euclidean region ($\forall s_{ij} < 0$) there is a further possibility:

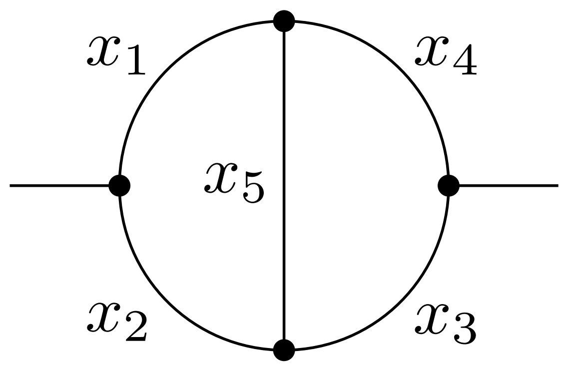
4. $\mathcal{F}(\vec{x}, s_{ij})$ vanishes inside the integration region (May give: Landau singularity which is either a normal or anomalous threshold)

Can be handled by `SecDec`: contourdef=True)

See: Soper 00; Borowka 14

Aside: If only condition 1 leads to a divergence the integral is **Quasi-finite**

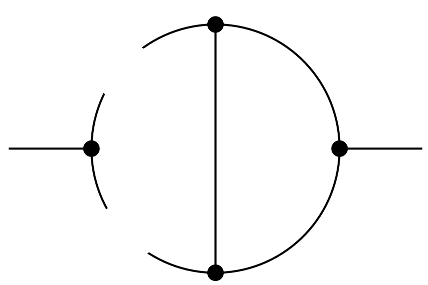
Constructing U



Draw graph, label edges with Feynman Parameters

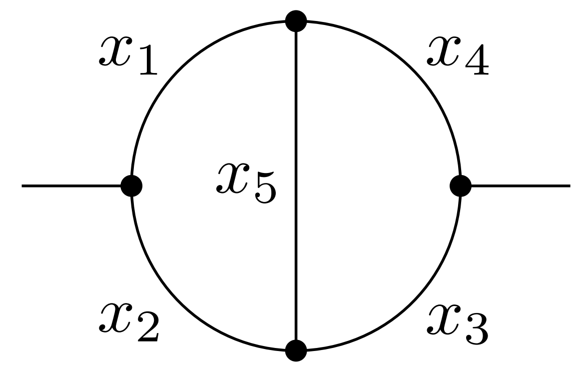
Rules for $\mathcal{U}(\vec{x})$**:**

1. Delete L edges all possible ways
2. Throw away disconnected graphs or graphs with $L \neq 0$
3. Sum monomials of Feynman parameters of deleted edges



$\mathcal{U}(\vec{x}) =$

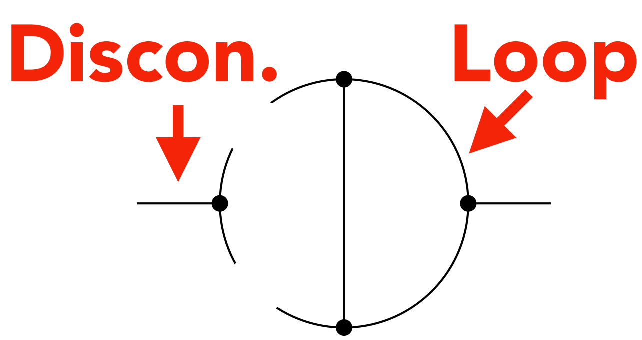
Constructing U



Draw graph, label edges with Feynman Parameters

Rules for $\mathcal{U}(\vec{x})$**:**

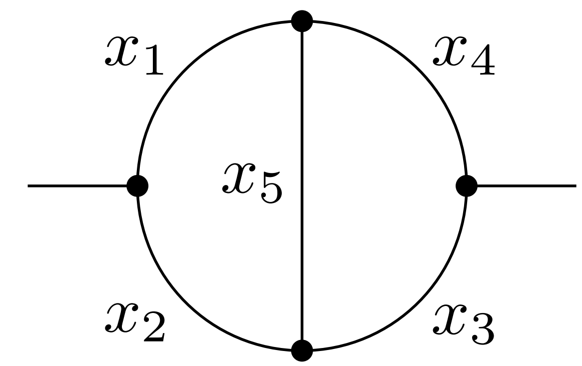
1. Delete L edges all possible ways
2. Throw away disconnected graphs or graphs with $L \neq 0$
3. Sum monomials of Feynman parameters of deleted edges



$\mathcal{U}(\vec{x}) =$

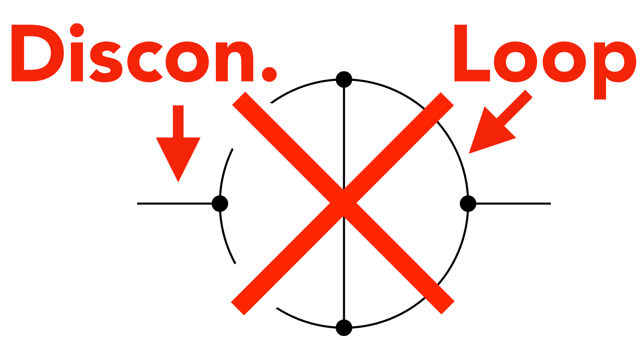
Constructing U

Draw graph, label edges with Feynman Parameters



Rules for $\mathcal{U}(\vec{x})$:

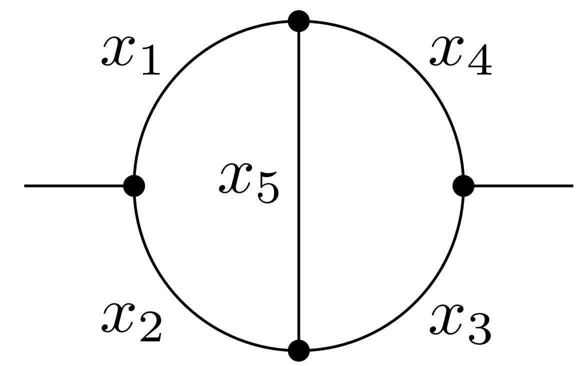
1. Delete L edges all possible ways
2. Throw away disconnected graphs or graphs with $L \neq 0$
3. Sum monomials of Feynman parameters of deleted edges



$\mathcal{U}(\vec{x}) =$

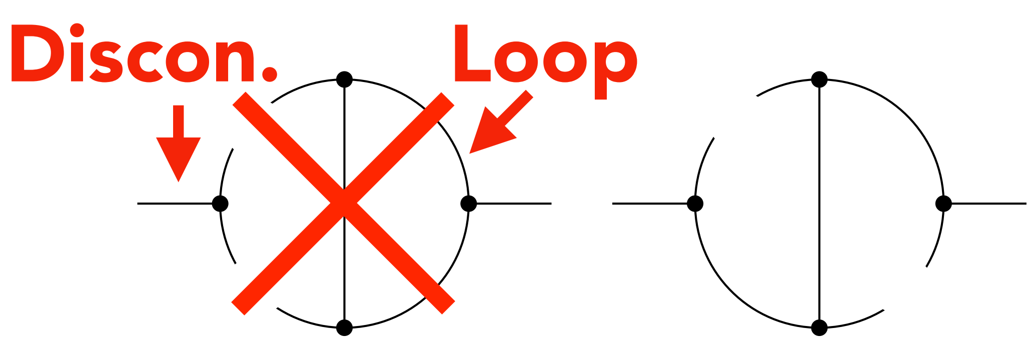
Constructing U

Draw graph, label edges with Feynman Parameters



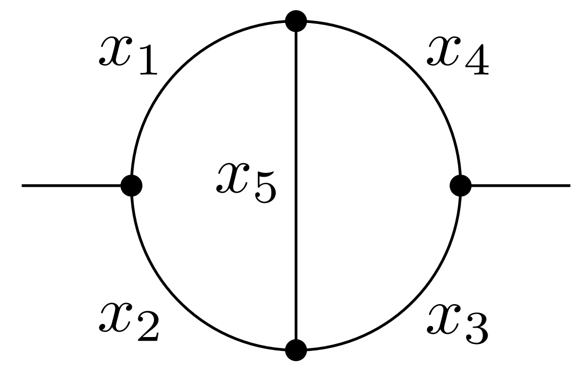
Rules for $\mathcal{U}(\vec{x})$:

1. Delete L edges all possible ways
2. Throw away disconnected graphs or graphs with $L \neq 0$
3. Sum monomials of Feynman parameters of deleted edges



$\mathcal{U}(\vec{x}) =$ $\quad +x_1 x_3$

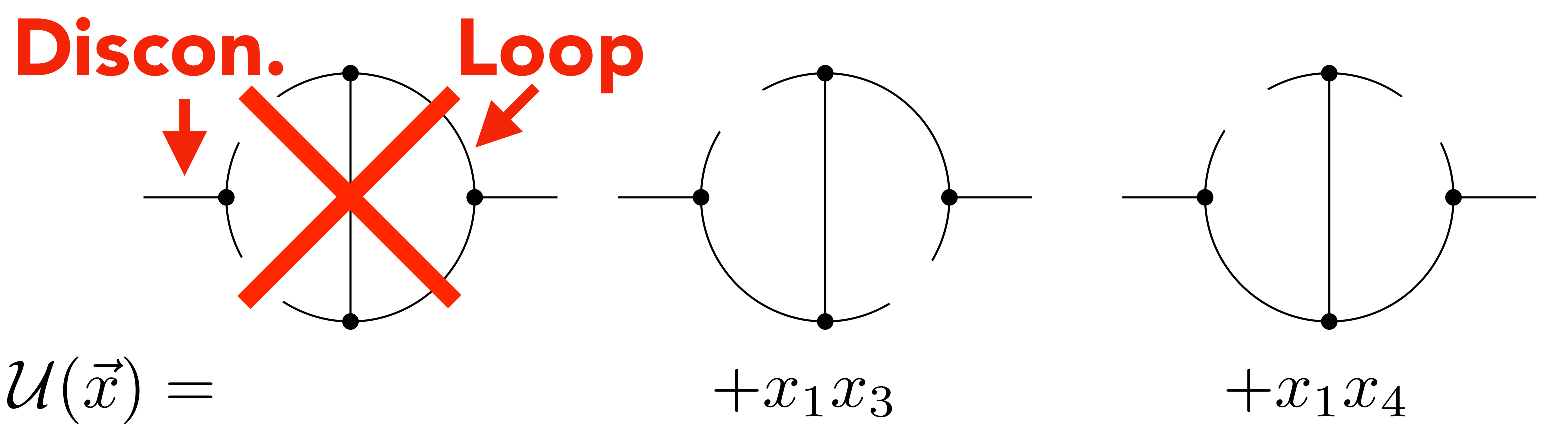
Constructing U



Draw graph, label edges with Feynman Parameters

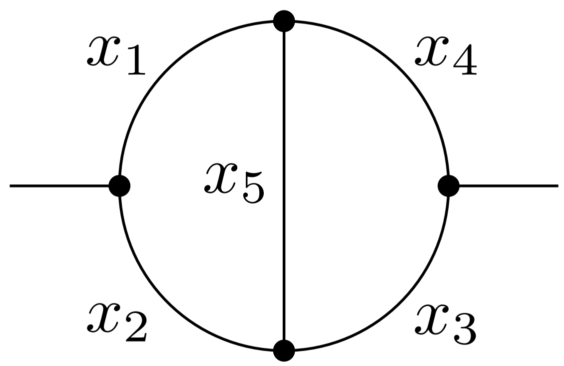
Rules for $\mathcal{U}(\vec{x})$**:**

1. Delete L edges all possible ways
2. Throw away disconnected graphs or graphs with $L \neq 0$
3. Sum monomials of Feynman parameters of deleted edges



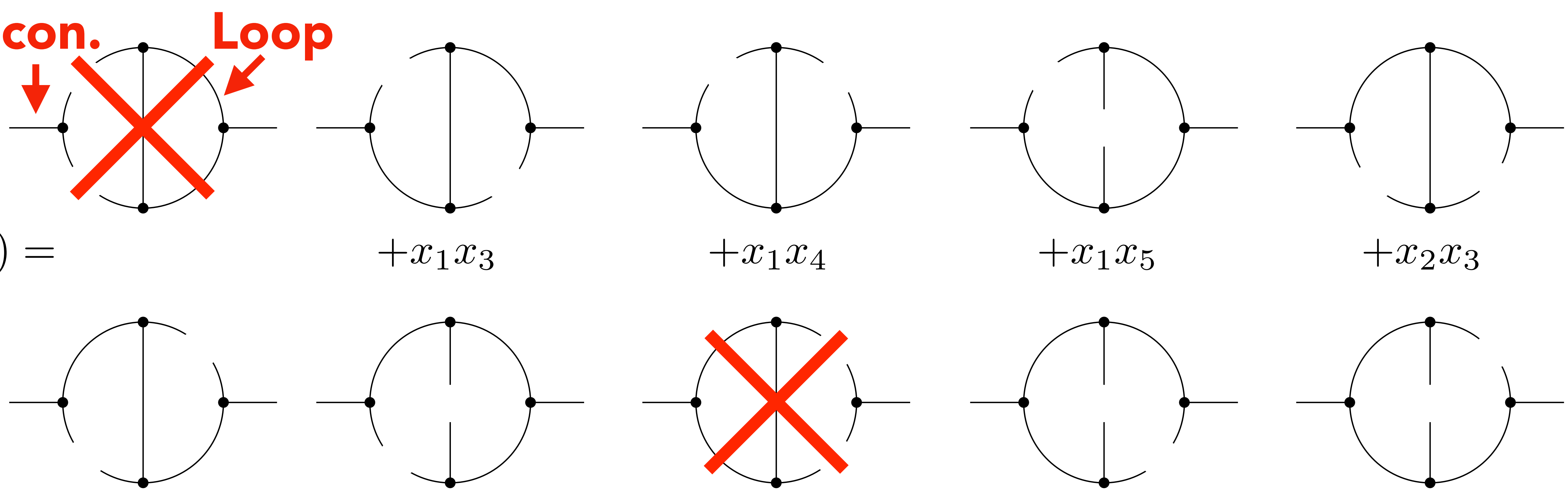
Constructing U

Draw graph, label edges with Feynman Parameters



Rules for $\mathcal{U}(\vec{x})$:

1. Delete L edges all possible ways
2. Throw away disconnected graphs or graphs with $L \neq 0$
3. Sum monomials of Feynman parameters of deleted edges



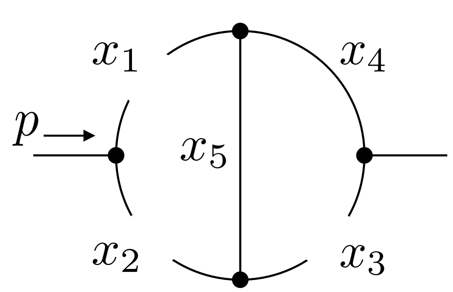
$$\mathcal{U}(\vec{x}) = +x_1x_3 +x_1x_4 +x_1x_5 +x_2x_3 +x_2x_4 +x_2x_5 +x_3x_5 +x_4x_5$$

Constructing F

Rules for $\mathcal{F}_0(\vec{x}, s_{ij})$ **:**

1. Delete $L+1$ edges all possible ways
2. Take only graphs with 2 connected components (T1, T2) and $L=0$
3. Sum F.P. monomials multiplied by: $-s_{ij} = -(\sum_k q_k)^2$
4. (For $\mathcal{F}(\vec{x}, s_{ij})$ add the internal mass terms)

Momenta flowing through cut lines from T1 → T2



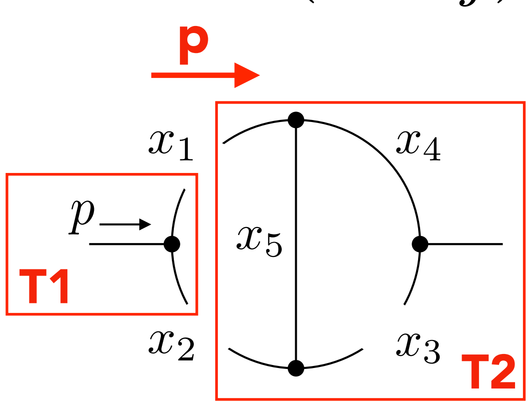
$$\mathcal{F}_0 = \ -p^2 x_1 x_2 x_3$$

Constructing F

Rules for $\mathcal{F}_0(\vec{x}, s_{ij})$:

1. Delete $L+1$ edges all possible ways
2. Take only graphs with 2 connected components (T1, T2) and $L=0$
3. Sum F.P. monomials multiplied by: $-s_{ij} = -(\sum_k q_k)^2$
4. (For $\mathcal{F}(\vec{x}, s_{ij})$ add the internal mass terms)

Momenta flowing through cut lines from T1 → T2



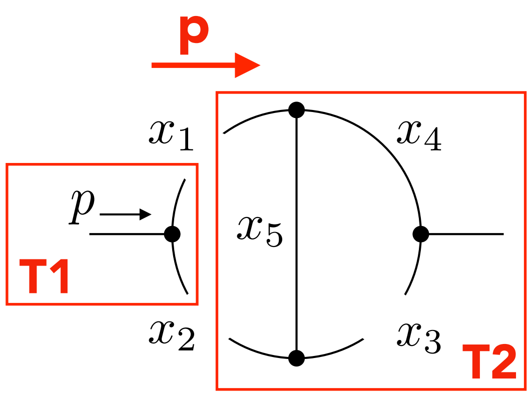
$$\mathcal{F}_0 = -p^2 x_1 x_2 x_3$$

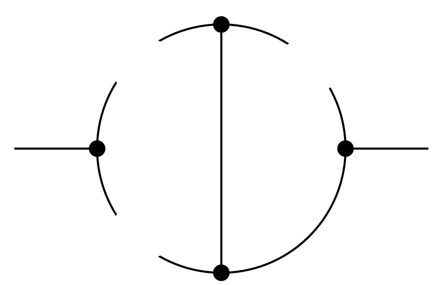
Constructing F

Rules for $\mathcal{F}_0(\vec{x}, s_{ij})$:

1. Delete $L+1$ edges all possible ways
2. Take only graphs with 2 connected components (T1, T2) and $L=0$
3. Sum F.P. monomials multiplied by: $-s_{ij} = -(\sum_k q_k)^2$
4. (For $\mathcal{F}(\vec{x}, s_{ij})$ add the internal mass terms)

Momenta flowing through cut lines from T1 → T2





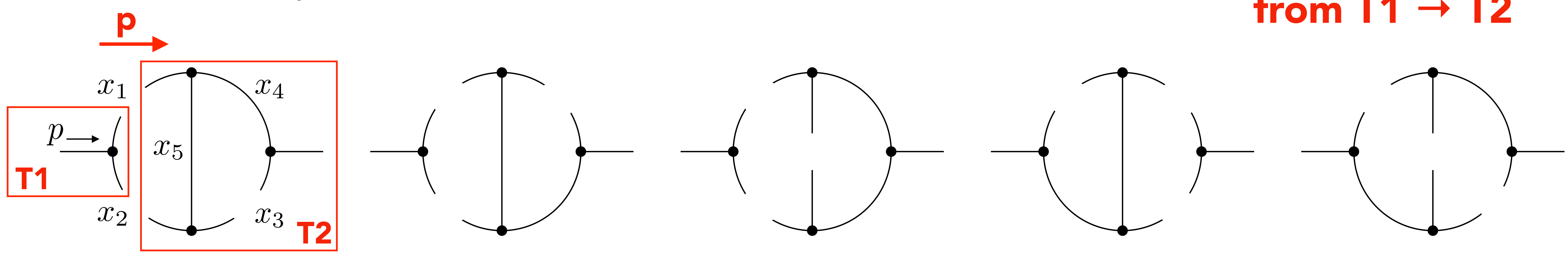
$\mathcal{F}_0 = \quad -p^2 x_1 x_2 x_3 \qquad -p^2 x_1 x_2 x_4$

Constructing F

Rules for $\mathcal{F}_0(\vec{x}, s_{ij})$:

1. Delete $L+1$ edges all possible ways
2. Take only graphs with 2 connected components (T1, T2) and $L=0$
3. Sum F.P. monomials multiplied by: $-s_{ij} = -(\sum_k q_k)^2$
4. (For $\mathcal{F}(\vec{x}, s_{ij})$ add the internal mass terms)

Momenta flowing through cut lines from T1 → T2



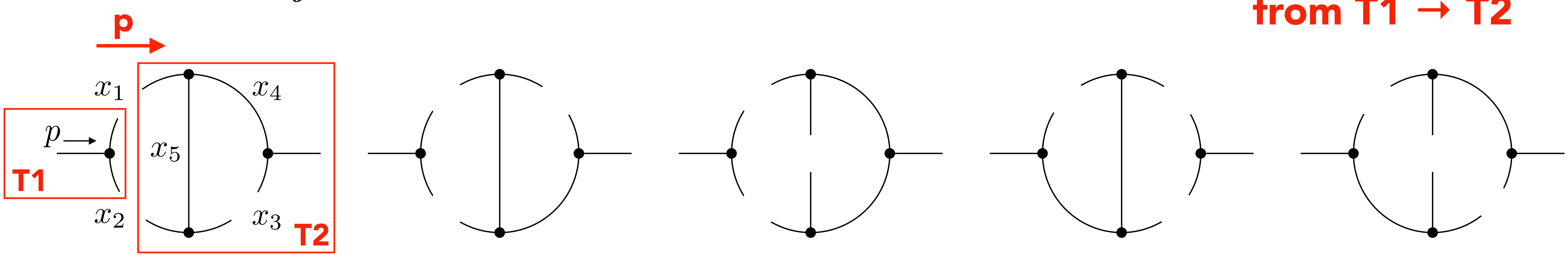
$$\mathcal{F}_0 = \; -p^2 x_1 x_2 x_3 \quad -p^2 x_1 x_2 x_4 \quad -p^2 x_1 x_2 x_5 \quad -p^2 x_1 x_3 x_4 \quad -p^2 x_1 x_3 x_5$$

Constructing F

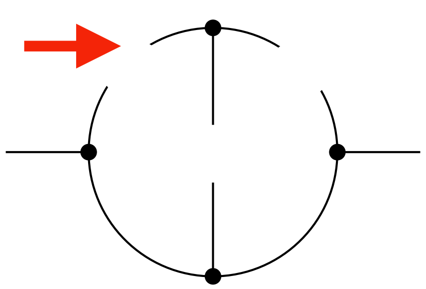
Rules for $\mathcal{F}_0(\vec{x}, s_{ij})$:

1. Delete $L+1$ edges all possible ways
2. Take only graphs with 2 connected components (T1, T2) and $L=0$
3. Sum F.P. monomials multiplied by: $-s_{ij} = -(\sum_k q_k)^2$
4. (For $\mathcal{F}(\vec{x}, s_{ij})$ add the internal mass terms)

Momenta flowing through cut lines from T1 → T2



$$\mathcal{F}_0 = \ -p^2 x_1 x_2 x_3 \quad -p^2 x_1 x_2 x_4 \quad -p^2 x_1 x_2 x_5 \quad -p^2 x_1 x_3 x_4 \quad -p^2 x_1 x_3 x_5$$



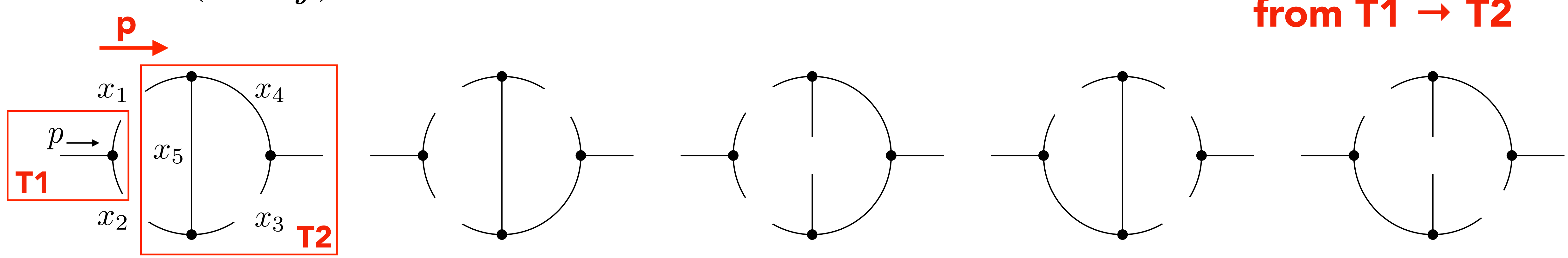
$$-0^2 x_1 x_4 x_5$$

Constructing F

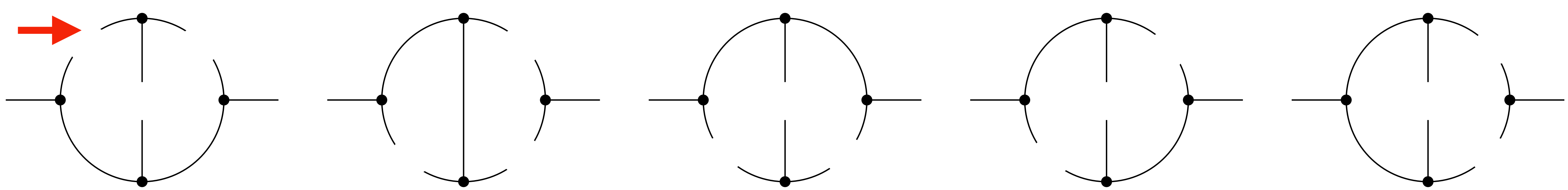
Rules for $\mathcal{F}_0(\vec{x}, s_{ij})$:

1. Delete $L+1$ edges all possible ways
2. Take only graphs with 2 connected components (T1, T2) and $L=0$
3. Sum F.P. monomials multiplied by: $-s_{ij} = -(\sum_k q_k)^2$
4. (For $\mathcal{F}(\vec{x}, s_{ij})$ add the internal mass terms)

Momenta flowing through cut lines from T1 → T2



$\mathcal{F}_0 = \; -p^2 x_1 x_2 x_3 \quad -p^2 x_1 x_2 x_4 \quad -p^2 x_1 x_2 x_5 \quad -p^2 x_1 x_3 x_4 \quad -p^2 x_1 x_3 x_5$



$-0^2 x_1 x_4 x_5 \quad -p^2 x_2 x_3 x_4 \quad -0^2 x_2 x_3 x_5 \quad -p^2 x_2 x_4 x_5 \quad -p^2 x_3 x_4 x_5$

Sector Decomposition

One technique **Iterated Sector Decomposition** repeat:

Binoth, Heinrich 00

$$\int_0^1 \mathrm{d}x_1 \int_0^1 \mathrm{d}x_2 \frac{1}{(x_1+x_2)^{2+\epsilon}}$$ ← **Overlapping singularity for** $x_1, x_2 \to 0$

$$= \int_0^1 \mathrm{d}x_1 \int_0^1 \mathrm{d}x_2 \frac{1}{(x_1+x_2)^{2+\epsilon}} \left(\theta(x_1-x_2)+\theta(x_2-x_1)\right)$$

$$= \int_0^1 \mathrm{d}x_1 \int_0^{x_1} \mathrm{d}x_2 \frac{1}{(x_1+x_2)^{2+\epsilon}} + \int_0^1 \mathrm{d}x_2 \int_0^{x_2} \mathrm{d}x_1 \frac{1}{(x_1+x_2)^{2+\epsilon}}$$

$$= \int_0^1 \mathrm{d}x_1 \int_0^{1} \mathrm{d}t_2 \frac{x_1}{(x_1+x_1t_2)^{2+\epsilon}} + \int_0^1 \mathrm{d}x_2 \int_0^{1} \mathrm{d}t_1 \frac{x_2}{(x_2t_1+x_2)^{2+\epsilon}}$$

$$= \int_0^1 \mathrm{d}x_1 \int_0^{1} \mathrm{d}t_2 \frac{x_1^{-1-\epsilon}}{(1+t_2)^{2+\epsilon}} + \int_0^1 \mathrm{d}x_2 \int_0^{1} \mathrm{d}t_1 \frac{x_2^{-1-\epsilon}}{(t_1+1)^{2+\epsilon}}$$ ← **Singularities factorised**

If this procedure terminates depends on order of decomposition steps

An alternative strategy **Geometric Sector Decomposition** always terminates; both strategies are implemented in `SecDec`.

Kaneko, Ueda 10; See also: Bogner, Weinzierl 08; Smirnov, Tentyukov 09