

Local Analytic Sector Subtraction with Initial-State Hadrons: Infrared Kernels and their Integration

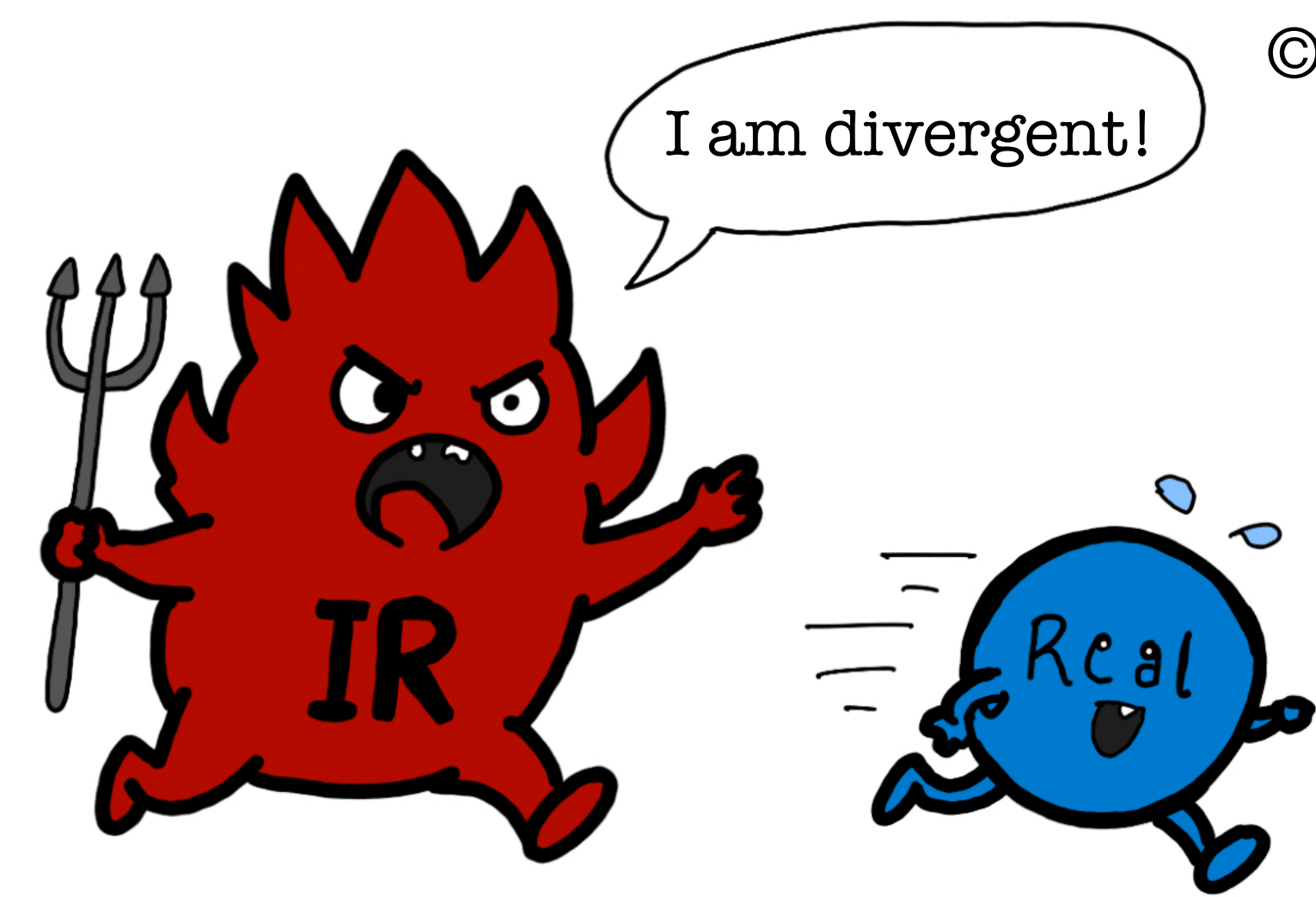
In collaboration with: G. Bertolotti, G. Limatola, L. Magnea, M. Rocco, C. Signorile-Signorile, P. Torrielli, S. Uccirati [\[2010.14493\]](#)[\[26XX.XXXX\]](#)

Motivation

Handling IR singularities at NNLO

- At NNLO,
See Chiara's talk

$$\frac{d\sigma_{NNLO}}{dX} = \int d\Phi_n VV \delta_{X_n} + \int d\Phi_{n+1} RV \delta_{X_{n+1}} + \int d\Phi_{n+2} RR \delta_{X_{n+2}}$$

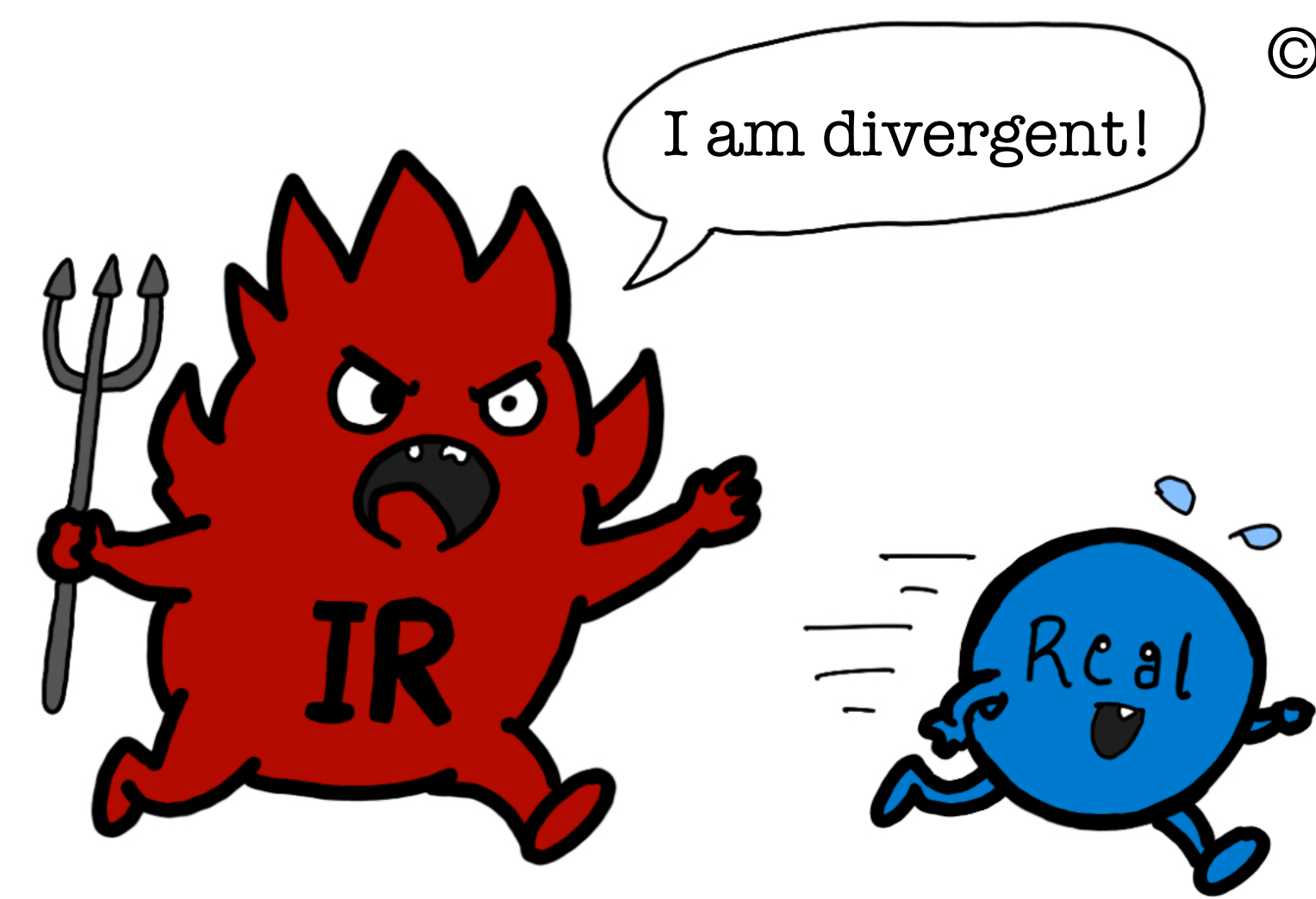
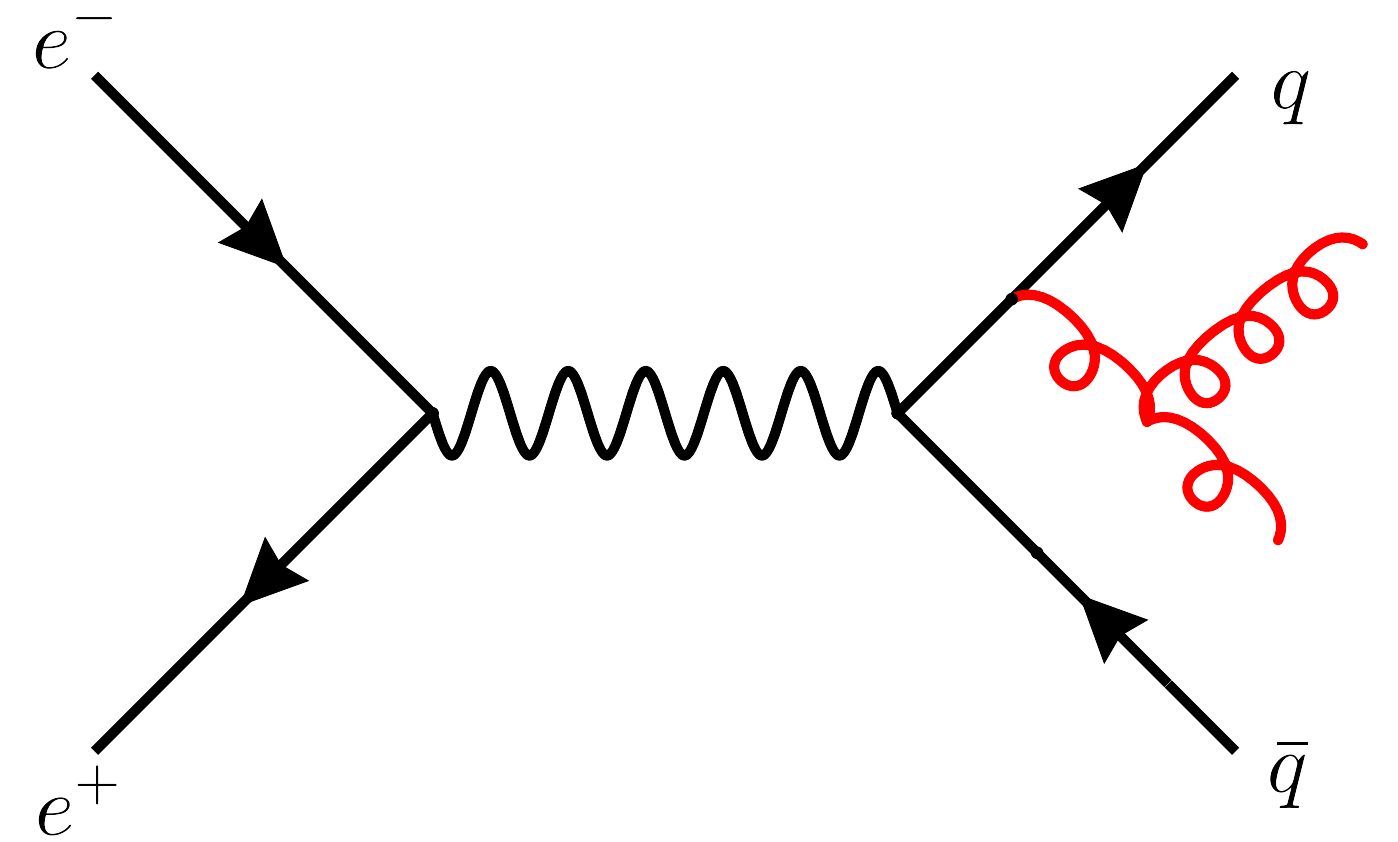
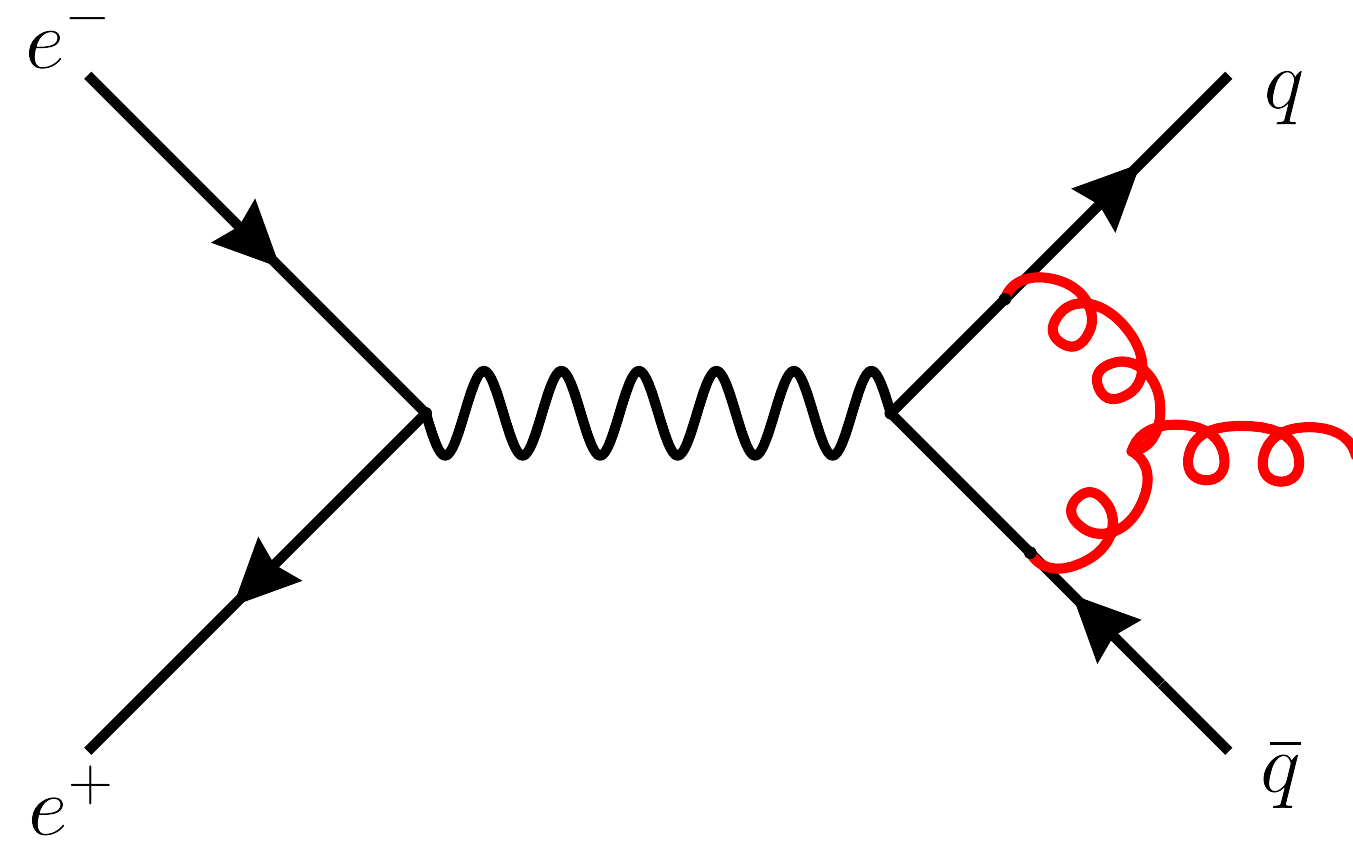
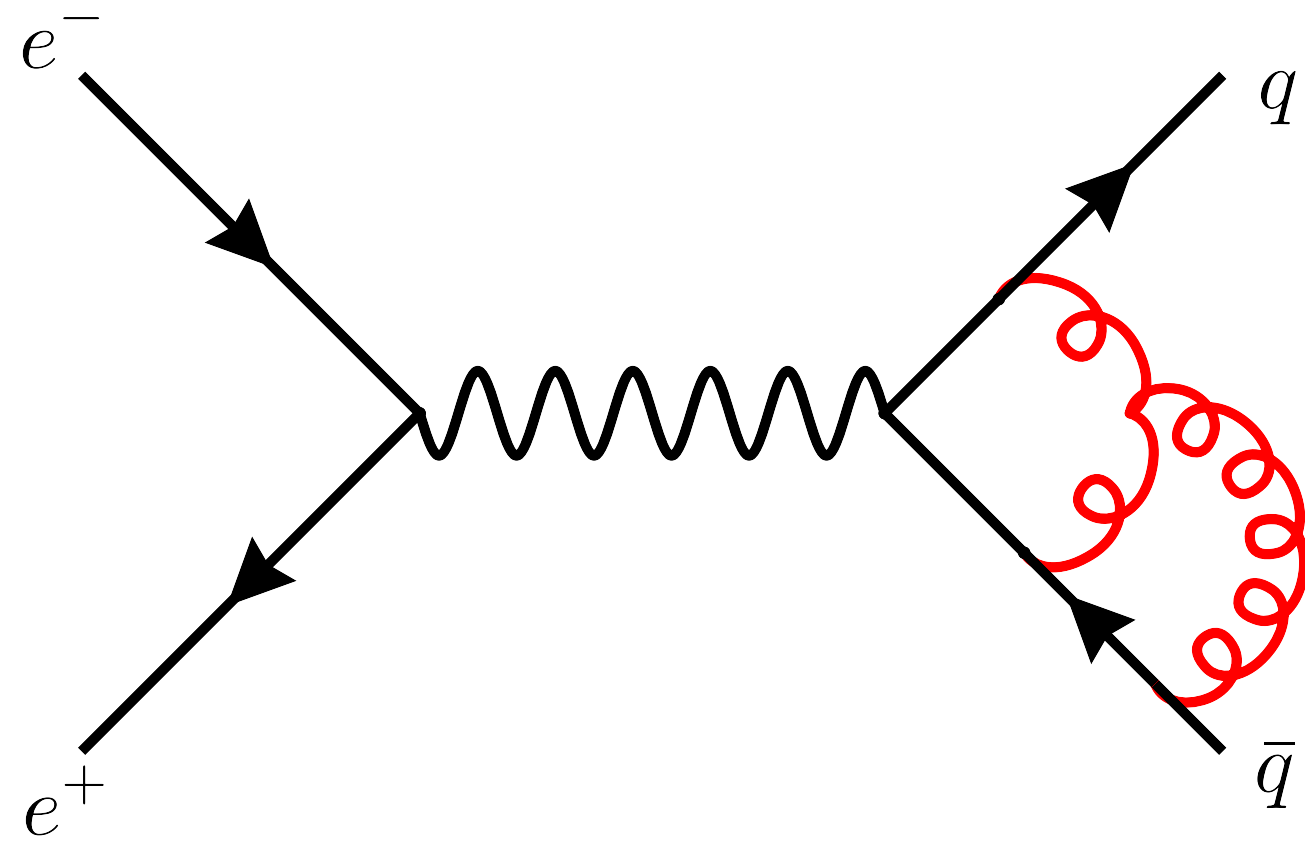


Motivation

Handling IR singularities at NNLO

- At NNLO,
See Chiara's talk

$$\frac{d\sigma_{NNLO}}{dX} = \int d\Phi_n VV \delta_{X_n} + \int d\Phi_{n+1} RV \delta_{X_{n+1}} + \int d\Phi_{n+2} RR \delta_{X_{n+2}}$$



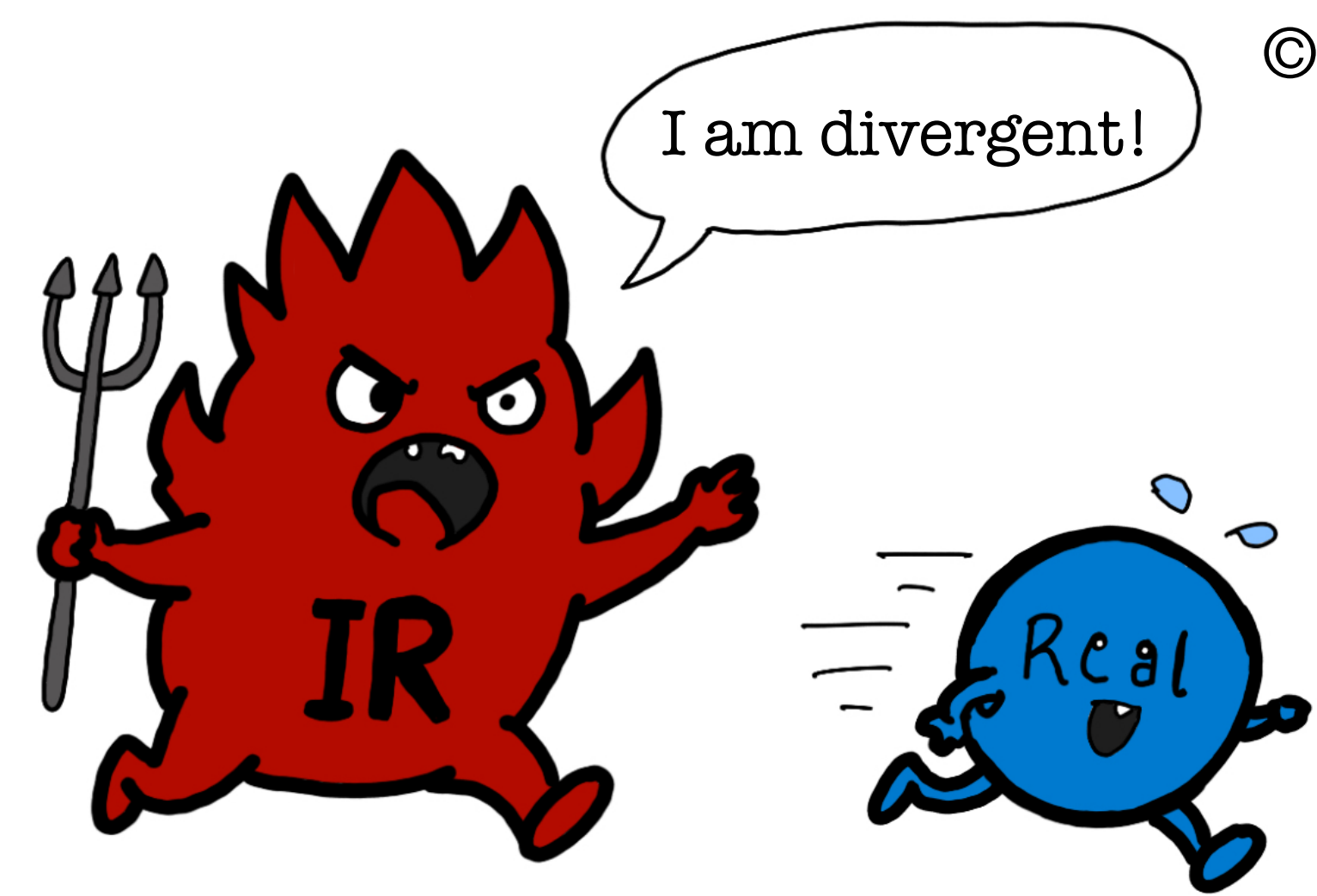
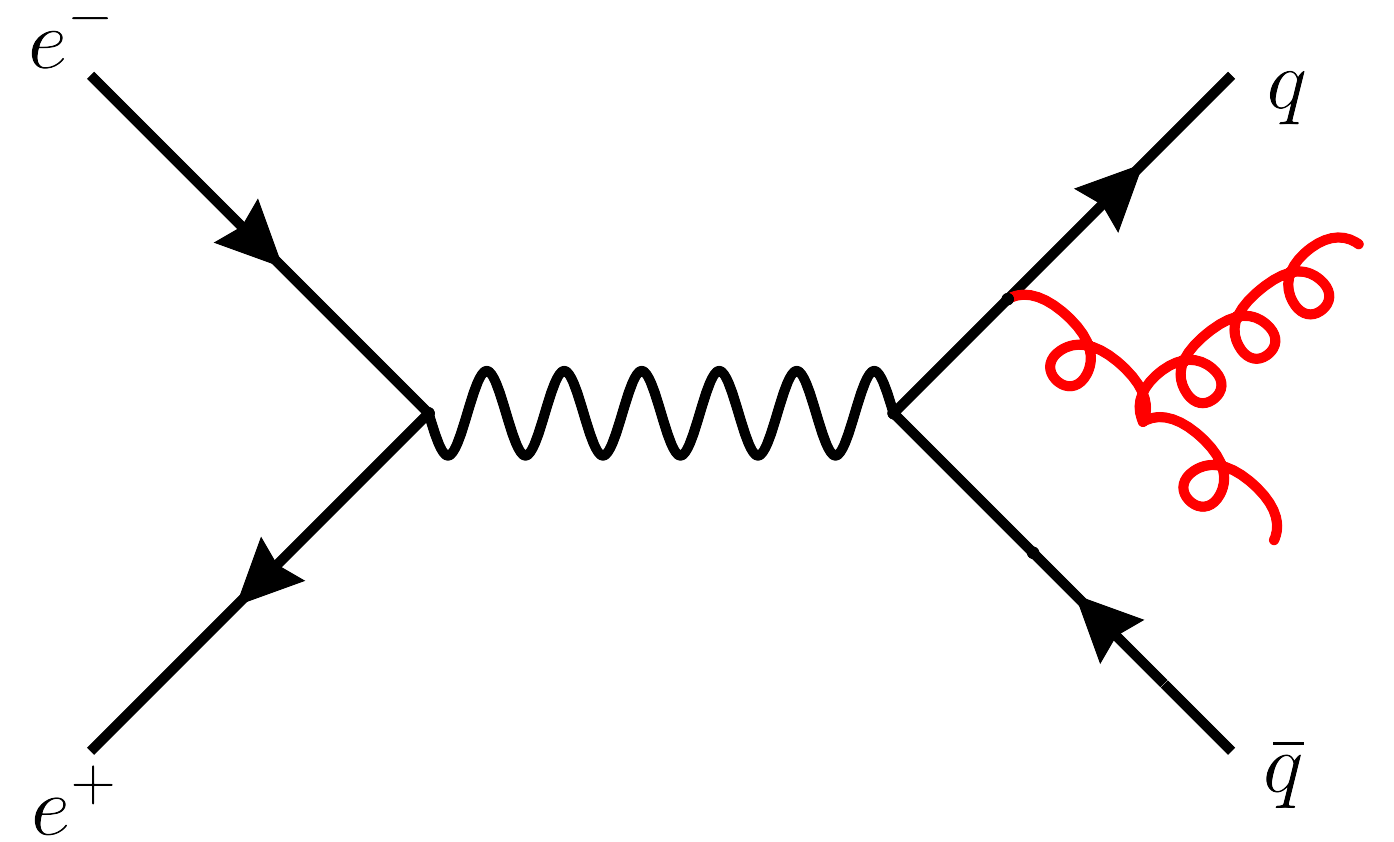
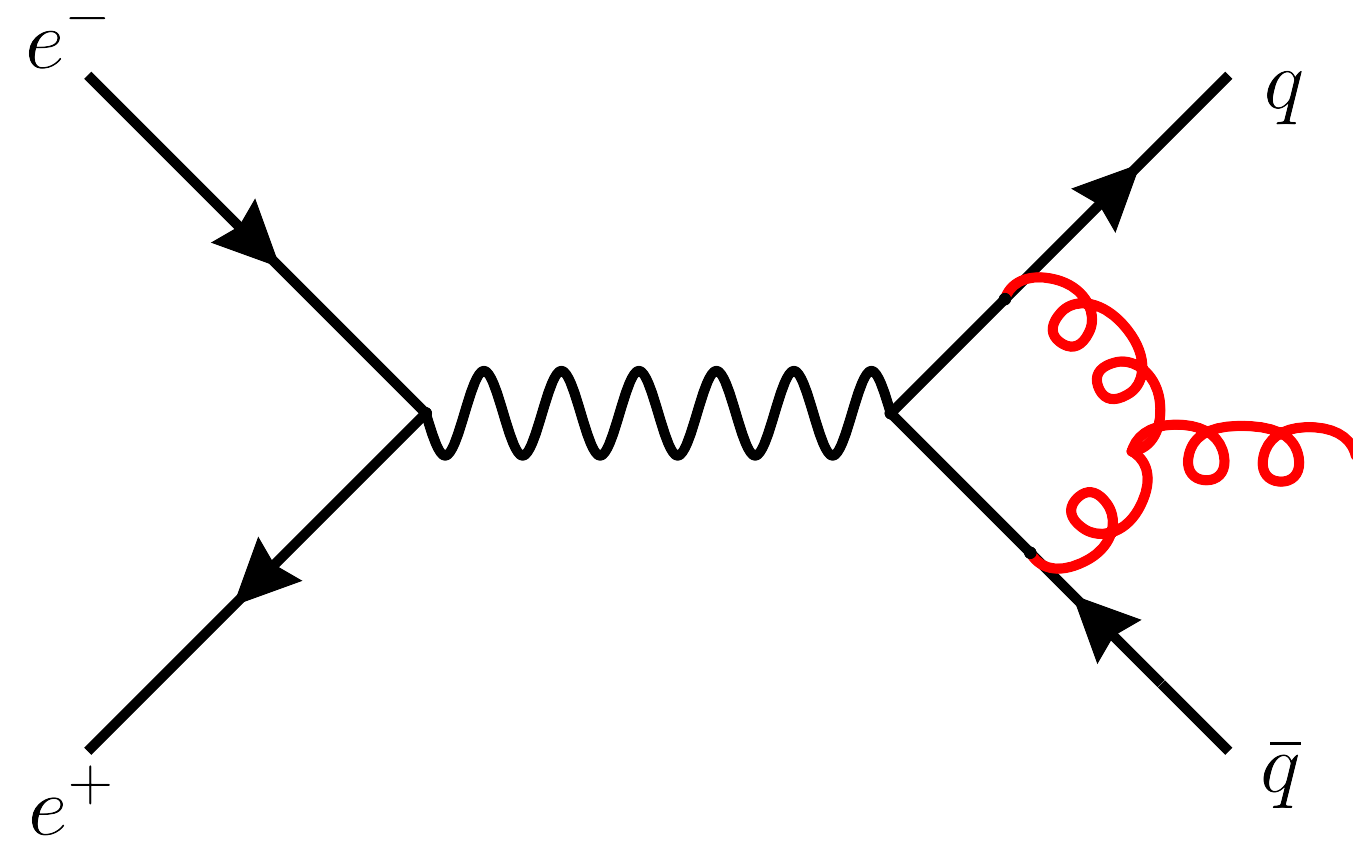
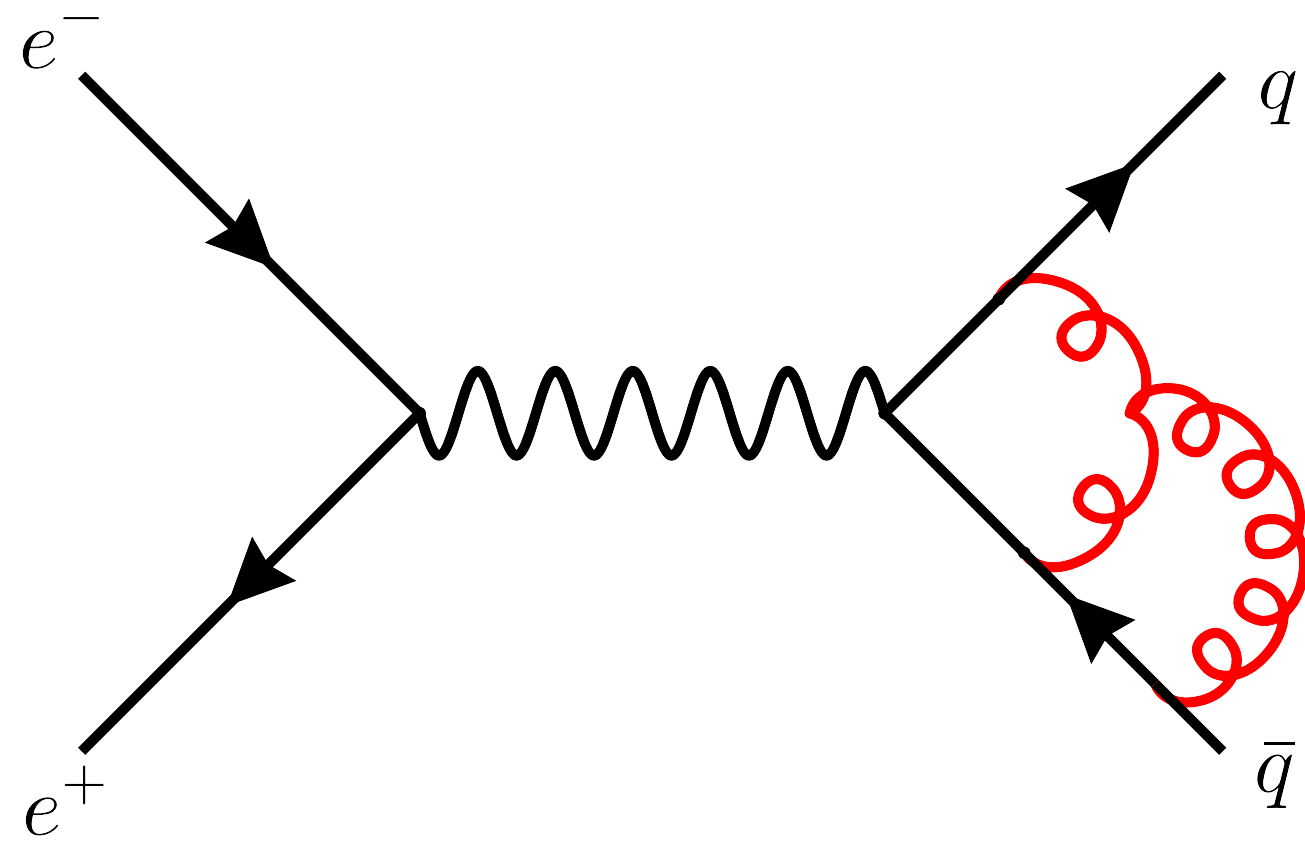
Motivation

Handling IR singularities at NNLO

- At NNLO,
See Chiara's talk

$$\frac{d\sigma_{NNLO}}{dX} = \int d\Phi_n \text{VV} \delta_{X_n} + \int d\Phi_{n+1} \text{RV} \delta_{X_{n+1}} + \int d\Phi_{n+2} \text{RR} \delta_{X_{n+2}}$$

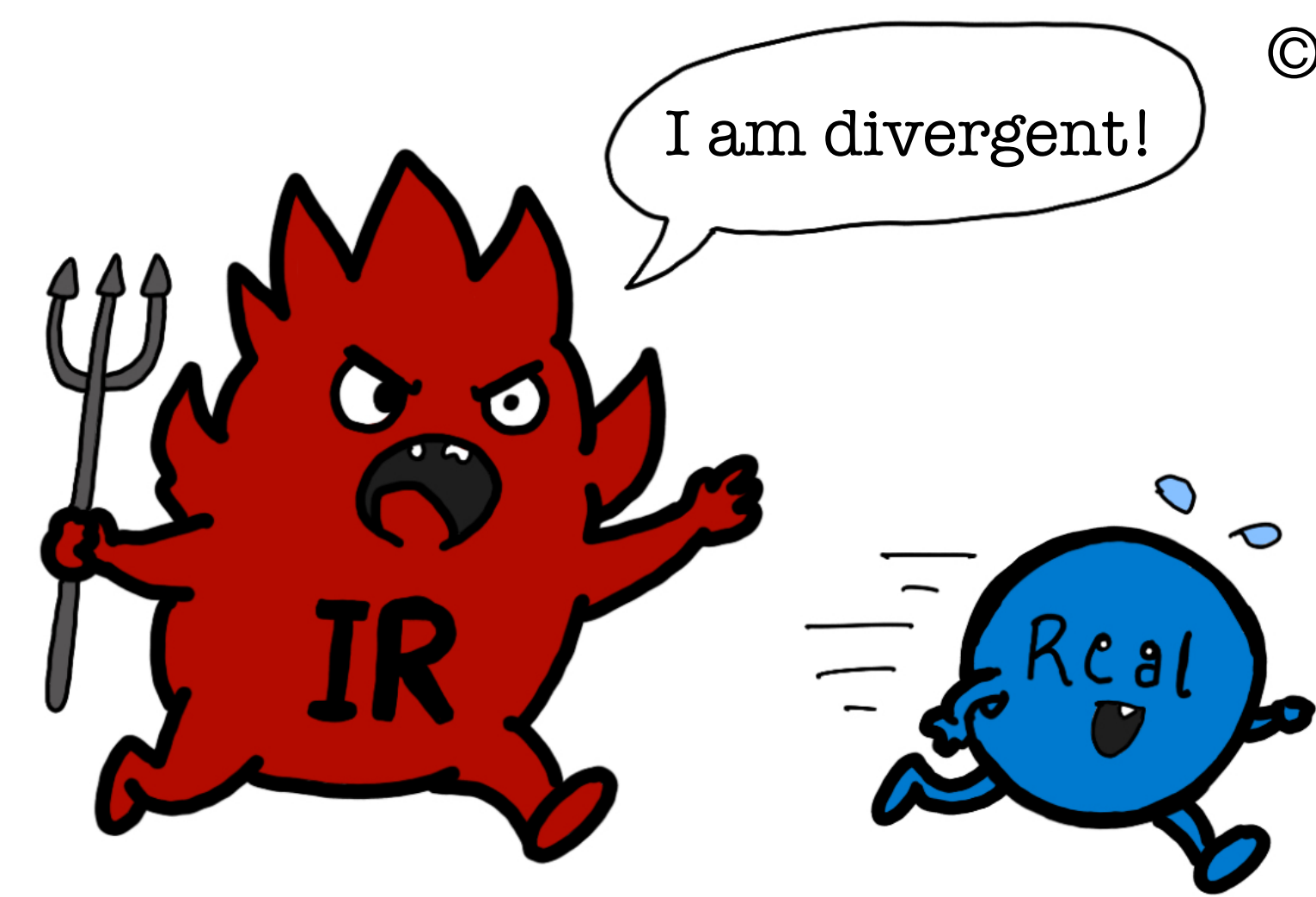
Poles in ϵ



Motivation

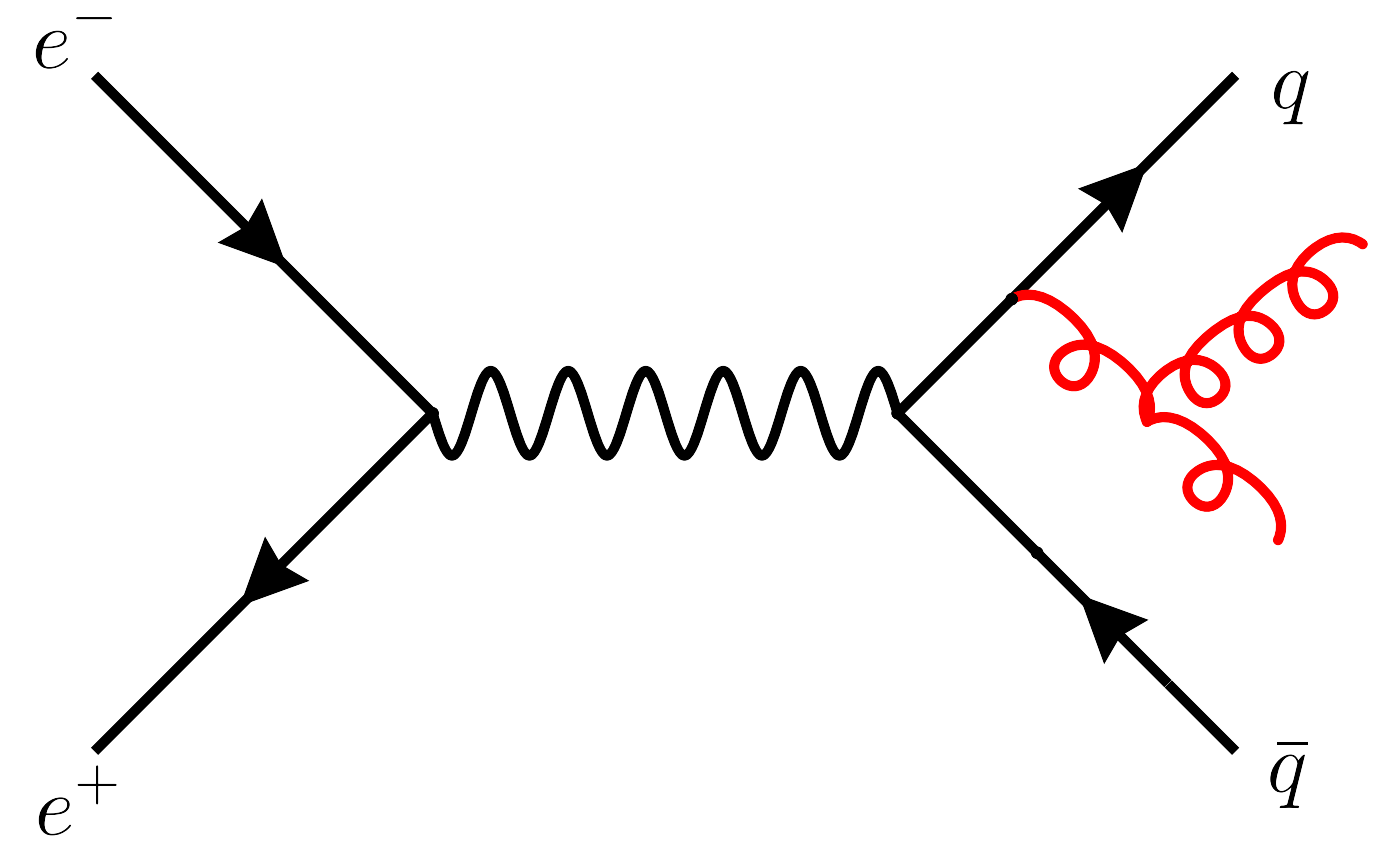
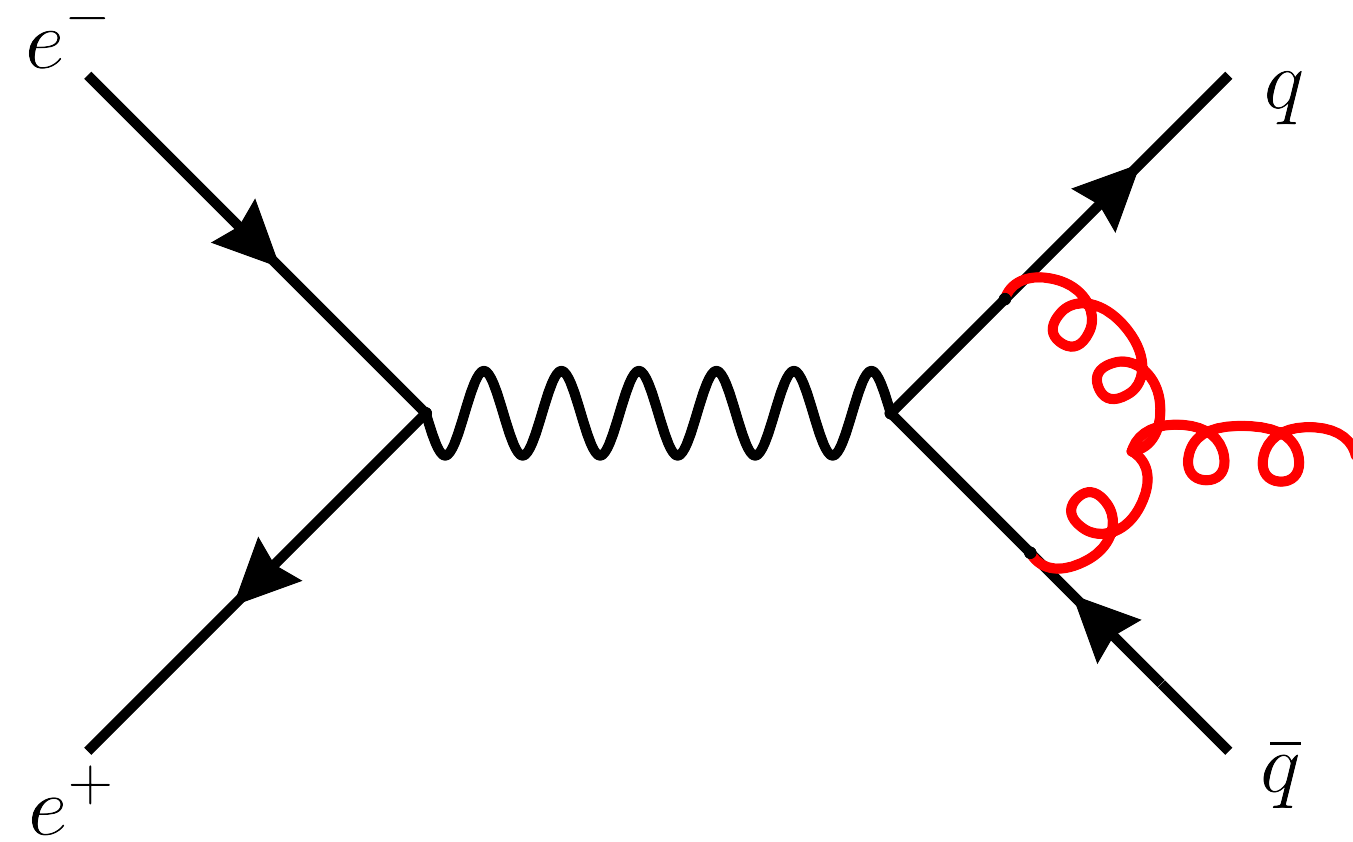
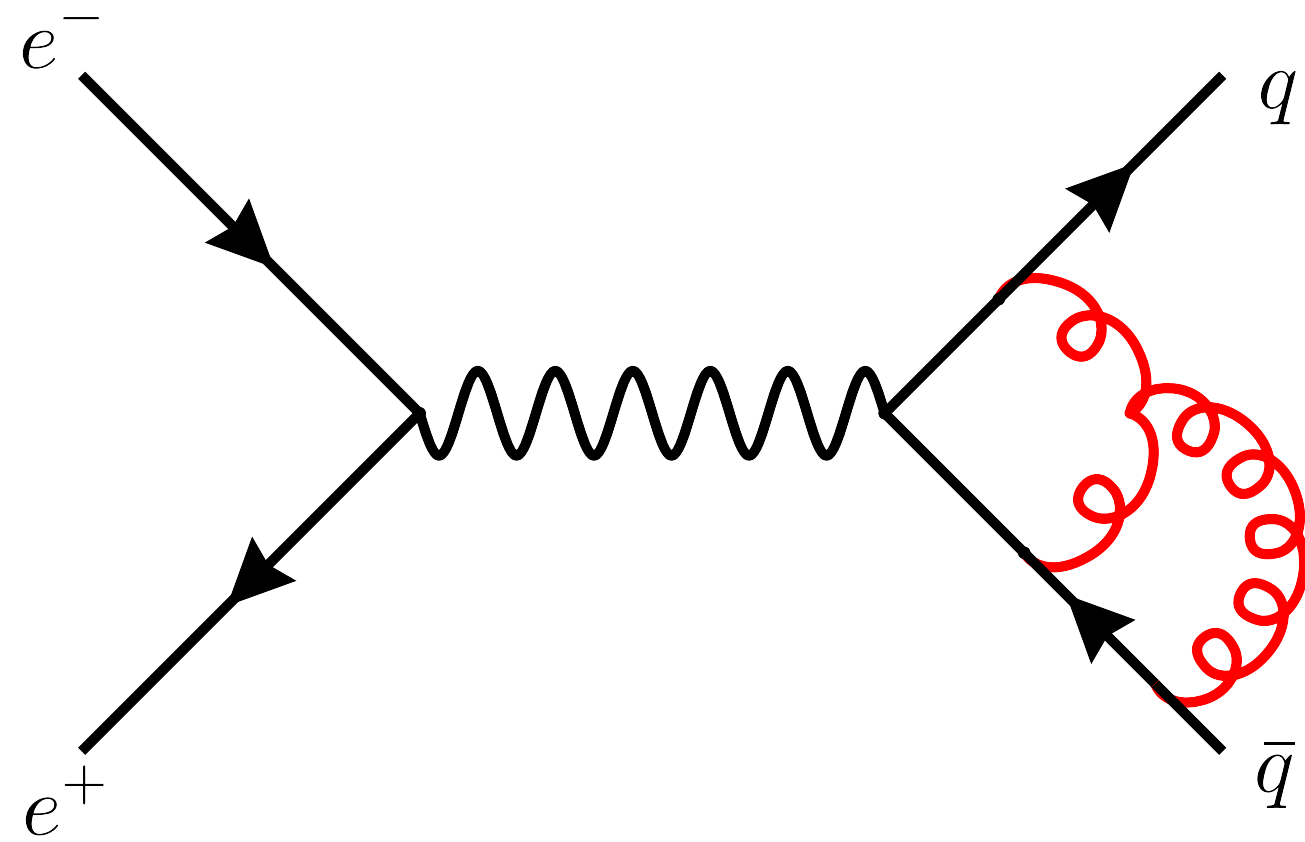
Handling IR singularities at NNLO

- At NNLO,
See Chiara's talk



$$\frac{d\sigma_{NNLO}}{dX} = \int d\Phi_n \text{VV} \delta_{X_n} + \int d\Phi_{n+1} \text{RV} \delta_{X_{n+1}} + \int d\Phi_{n+2} \text{RR} \delta_{X_{n+2}}$$

Poles in ϵ Diverge in phase-space



Motivation

Handling IR singularities at NNLO

Local Subtraction

Antenna [[hep-ph/0505111](#)]
[Gehrmann-De Ridder, Gehrmann,
Glover]

ColourfulNNLO [[hep-ph/0609042](#)]
[Del Duca, Trócsányi, Somogyi][**See
talk by G. Somogyi**]

STRIPPER [[hep-ph/1005.0274](#)]
[Czakon]

Nested soft-collinear [[1702.01352](#)]
[Caola, Melnikov, Röntschi][**See talk by
M. Tresoldi**]

Geometric IR subtraction
[[1804.07949](#)][Herzog]

Local Analytic Sector Subtraction
→ **this talk!**

Slicing

q_T slicing [[hep-ph/0703012](#)]
[Catani, Grazzini]

N-jettiness [[1505.04794](#)]
[Gaunt, Stahlhofen,
Tackmann, Walsh]

Projection-to-Born
[[1506.02660](#)][Cacciari,
Dreyer, Karlberg, Salam,
Zanderighi]

Integrand Level

Unsubtraction
[[1604.06699](#)][Sborlini,
Driencourt-Mangin,
Hernández-Pinto, Rodrigo]

FDR [[1208.5457](#)][Pittau]

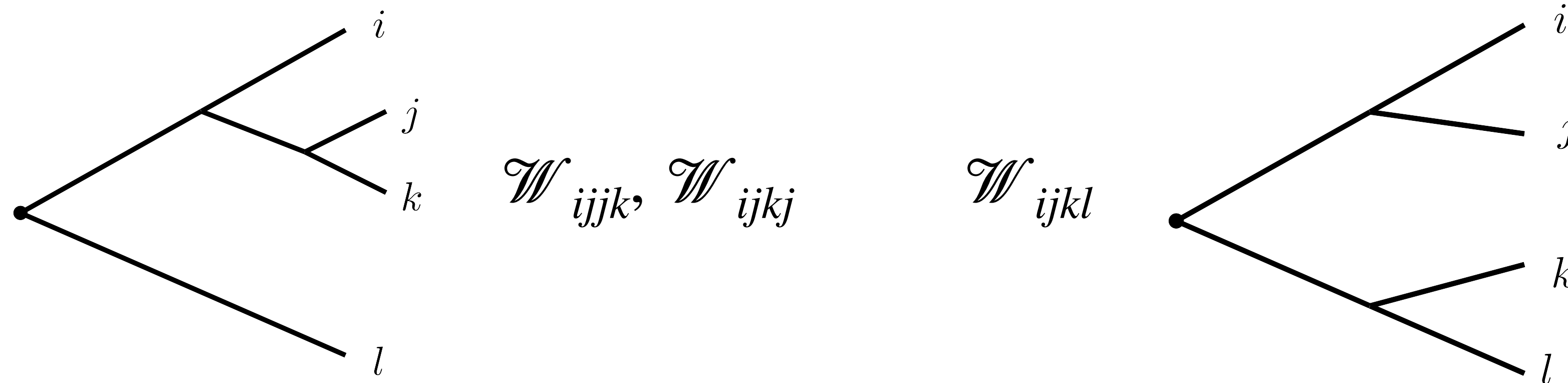
Universal factorisation
[[1812.03753](#)][Anastasiou,
Sterman]

Local Analytic Sector Subtraction

Introduction [[2212.11190](#)] [G. Bertolotti, L. Magnea,
G. Pelliccioli, A. Ratti, C. Signorile-Signorile, P. Torrielli, S. Uccirati]



- **Local** counterterms from the universal soft and collinear kernels
- **Analytic** integration of the counterterms once and for all
- **Sector** functions to disentangle singularities



- **Subtraction**, a universal formula – same structure for any process

Structure

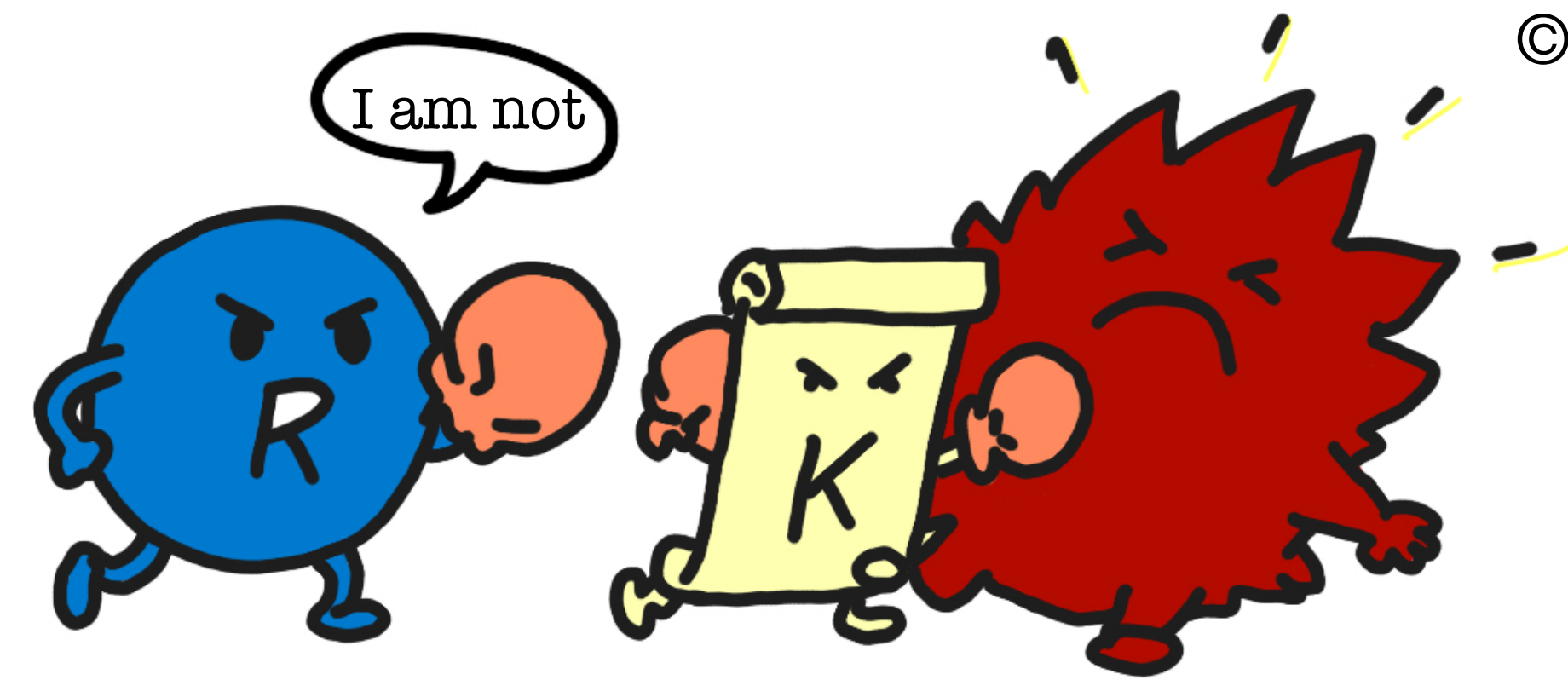
Towards NNLO cross-sections

- How is LASS doing it?

$$\frac{d\sigma_{NNLO}}{dX} = \int d\Phi_n VV_{\text{sub}}(X) + \int d\Phi_{n+1} RV_{\text{sub}}(X) + \int d\Phi_{n+2} RR_{\text{sub}}(X)$$

Structure

Towards NNLO cross-sections



- How is LASS doing it?

$$\frac{d\sigma_{NNLO}}{dX} = \int d\Phi_n \text{VV}_{\text{sub}}(X) + \int d\Phi_{n+1} \text{RV}_{\text{sub}}(X) + \int d\Phi_{n+2} \text{RR}_{\text{sub}}(X)$$

↓

$$\left(VV + I^{(2)} + I^{(RV)} \right) \delta_n(X)$$

↓

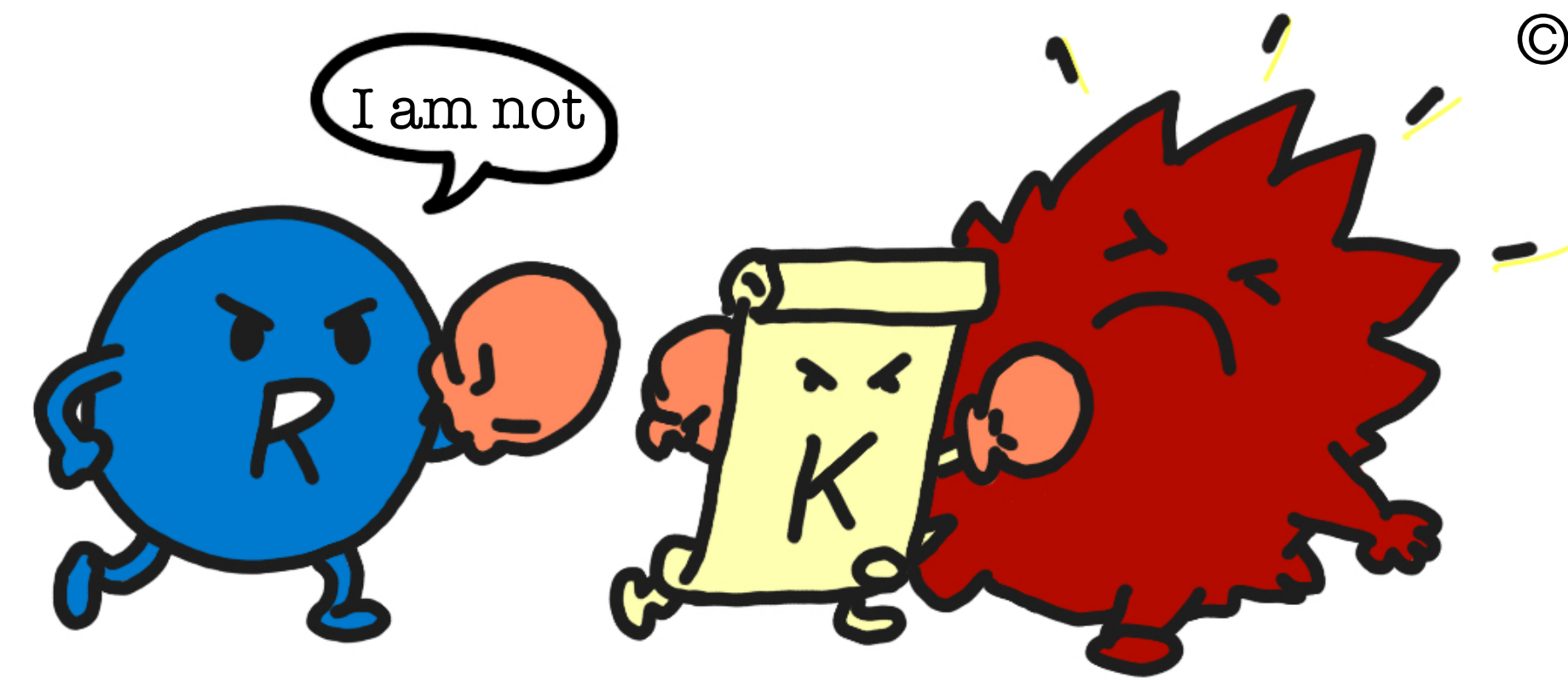
$$\left(RV + I^{(1)} \right) \delta_{n+1}(X) - \left(K^{(RV)} + I^{(12)} \right) \delta_n(X)$$

↓

$$RR\delta_{n+2}(X) - K^{(1)}\delta_{n+1}(X) - \left(K^{(2)} - K^{(12)} \right) \delta_n(X)$$

Structure

Towards NNLO cross-sections



- How is LASS doing it?

$$\frac{d\sigma_{NNLO}}{dX} = \int d\Phi_n VV_{\text{sub}}(X) + \int d\Phi_{n+1} RV_{\text{sub}}(X) + \int d\Phi_{n+2} RR_{\text{sub}}(X)$$

$$(VV + I^{(2)} + I^{(RV)}) \delta_n(X)$$

Integrated CT

$$(RV + I^{(1)}) \delta_{n+1}(X) - (K^{(RV)} + I^{(12)}) \delta_n(X)$$

$$RR\delta_{n+2}(X) - K^{(1)}\delta_{n+1}(X) - (K^{(2)} - K^{(12)}) \delta_n(X)$$

Counterterm (CT)

From Kernels to Counterterms

Both ingredients of I

$$I = \int d\Phi_{\text{rad}} K$$

Phase space

Kernels

Why Mappings?

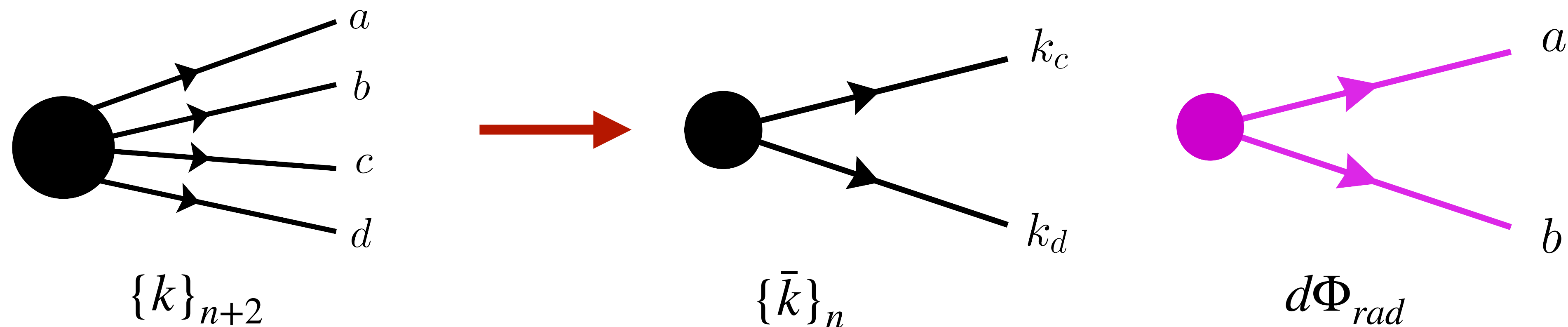
To integrate analytically

- **Why?** → Factorised Born: momentum not conserved [soft] and not on-shell [collinear]
- **What is should do?** → Mapping restores physical momenta: on-shell, momentum conserving

Why Mappings?

To integrate analytically

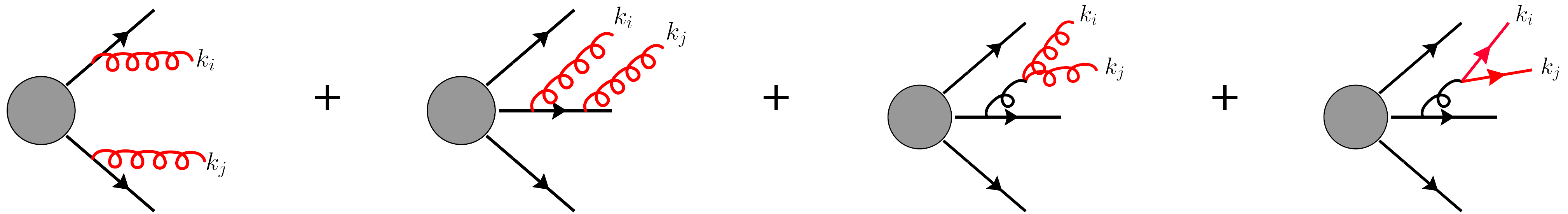
- **Why?** → Factorised Born: momentum not conserved [soft] and not on-shell [collinear]
- **What is should do?** → Mapping restores physical momenta: on-shell, momentum conserving
- **How?** → Exact factorisation, $d\Phi_{n+2} = d\Phi_n d\Phi_{rad}$



Sample **Kernels**

A concrete example

- Double soft limit [Catani, Grazzini]



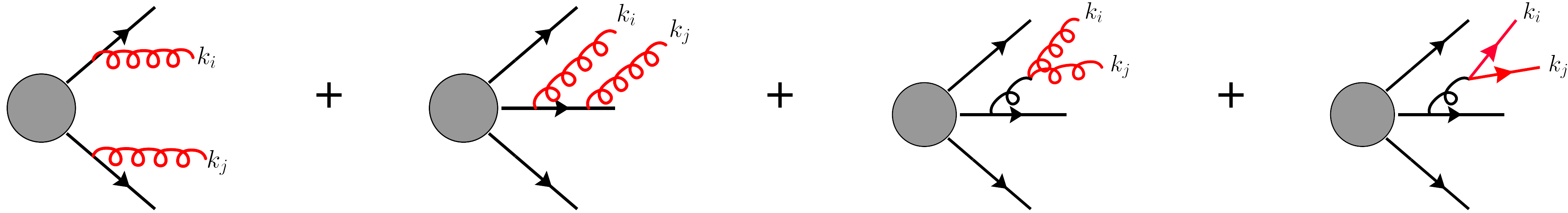
$$\lim_{k_a, k_b \rightarrow 0} RR_{n+2}(\{k\}_n, k_a, k_b) \sim \text{Eik}(\{k\}_n, k_a, k_b) \otimes B_n(\{k\}_n)$$

$$s_{ab} = 2k_a \cdot k_b$$

Sample Kernels

A concrete example

- Double soft limit [Catani, Grazzini]



$$\lim_{k_a, k_b \rightarrow 0} RR_{n+2}(\{k\}_n, k_a, k_b) \sim \text{Eik}(\{k\}_n, k_a, k_b) \otimes B_n(\{k\}_n) \rightarrow \sum_{c,d \neq a,b} \left[\sum_{e,f \neq a,b} I_{cd}^{(a)} I_{ef}^{(b)} B_{cdef} + I_{cd}^{(ab)} B_{cd} \right]$$

$$I_{cd}^{(a)} = \frac{s_{cd}}{s_{ac}s_{ad}}$$

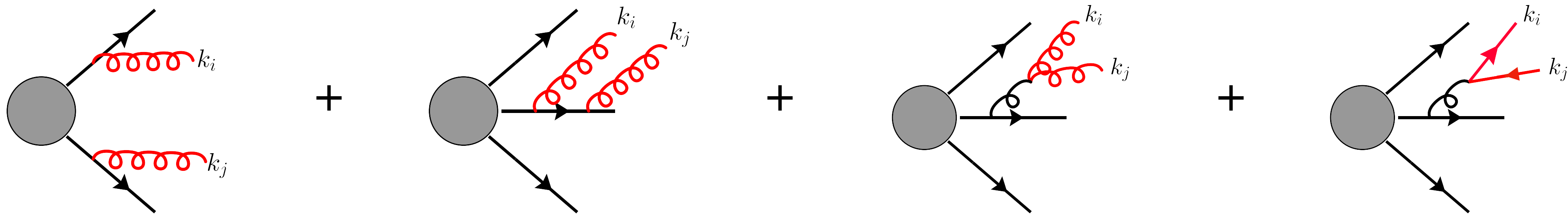
$$I_{cd}^{(ab)} = 2T_R I_{cd}^{(q\bar{q})(ab)} - 2C_A I_{cd}^{(gg)(ab)}$$

$$s_{ab} = 2k_a \cdot k_b$$

Sample **Kernels**

A concrete example

- Double soft limit [Catani, Grazzini]

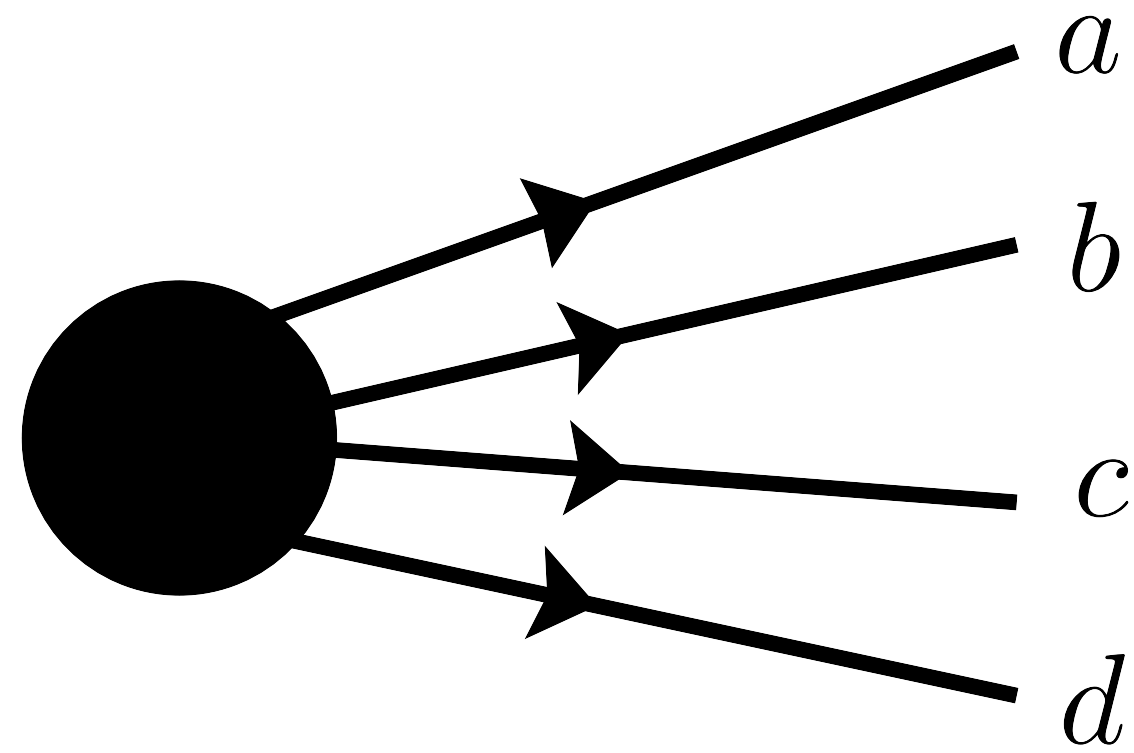


$$\lim_{k_a, k_b \rightarrow 0} RR_{n+2}(\{k\}_n, k_a, k_b) \sim \text{Eik}(\{k\}_n, k_a, k_b) \otimes B_n(\{k\}_n)$$

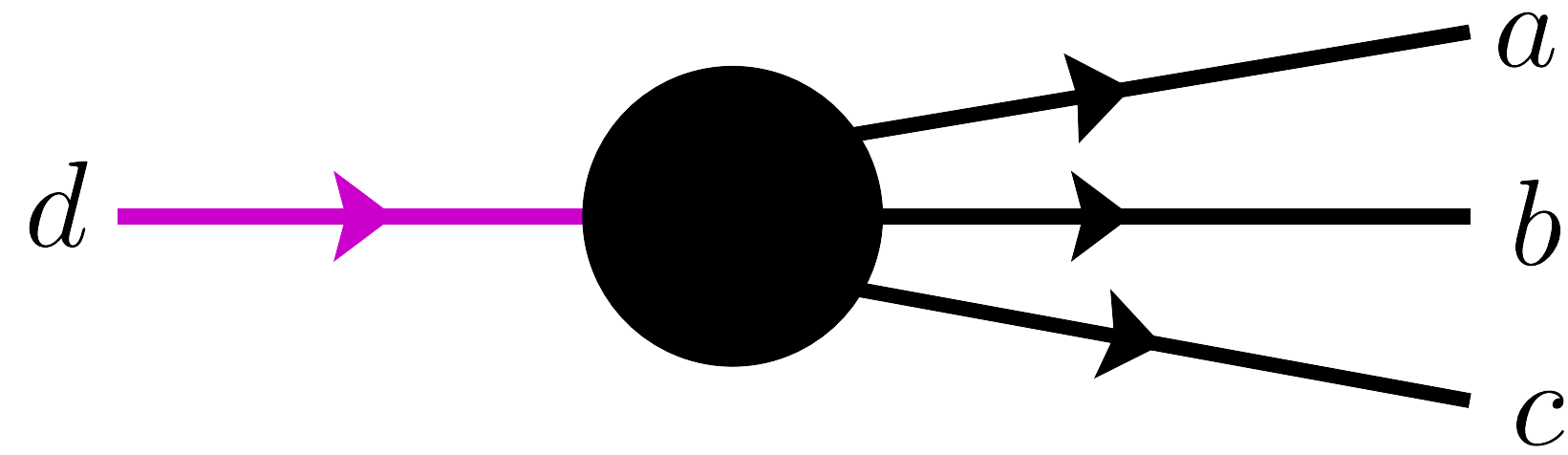
$$I_{cd}^{(gg)(ab)} = \frac{(1 - \epsilon)(s_{ac}s_{bd} + s_{ad}s_{bc}) - 2s_{ab}s_{cd}}{s_{ab}^2(s_{ac} + s_{bc})(s_{ad} + s_{bd})} + s_{cd} \frac{s_{ac}s_{bd} + s_{ad}s_{bc} - s_{ab}s_{cd}}{s_{ab}s_{ac}s_{bd}s_{ad}s_{bc}} \left[1 - \frac{1}{2} \frac{s_{ac}s_{bd} + s_{ad}s_{bc}}{(s_{ac} + s_{bc})(s_{ad} + s_{bd})} \right]$$

Four-Particle Phase-Space

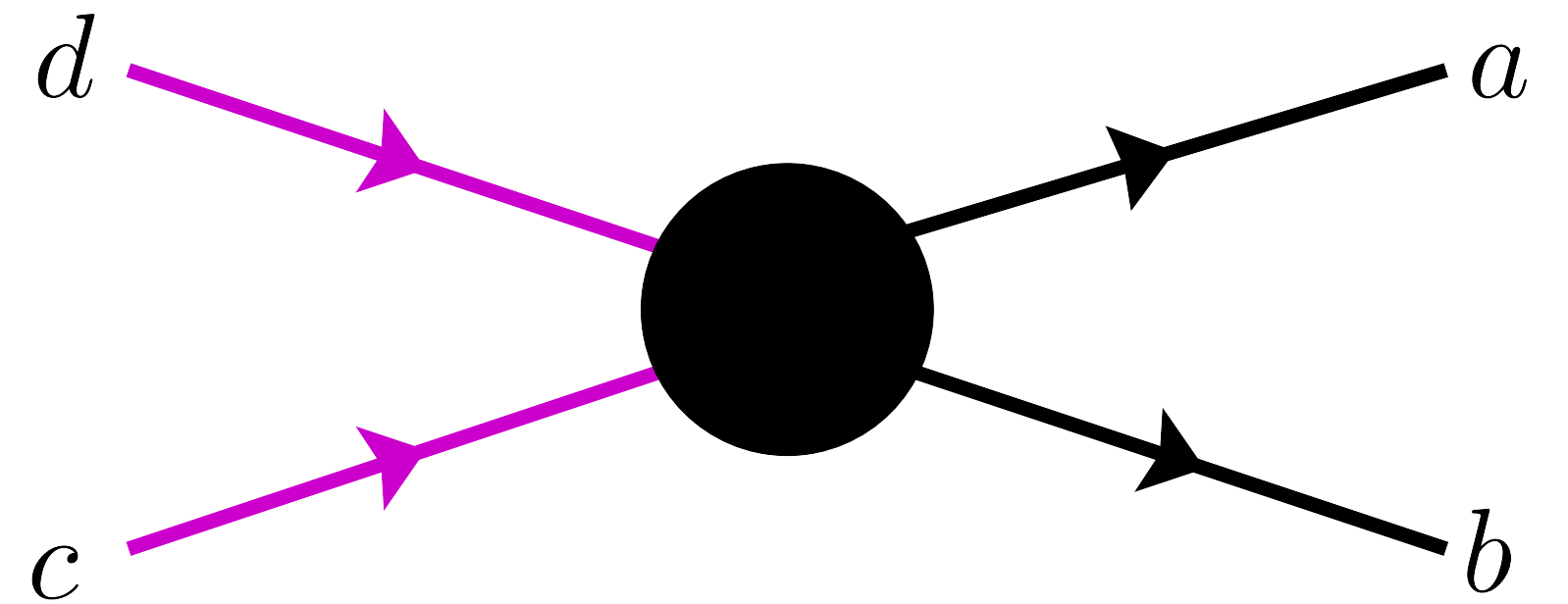
One construction, three configurations



(abcd)



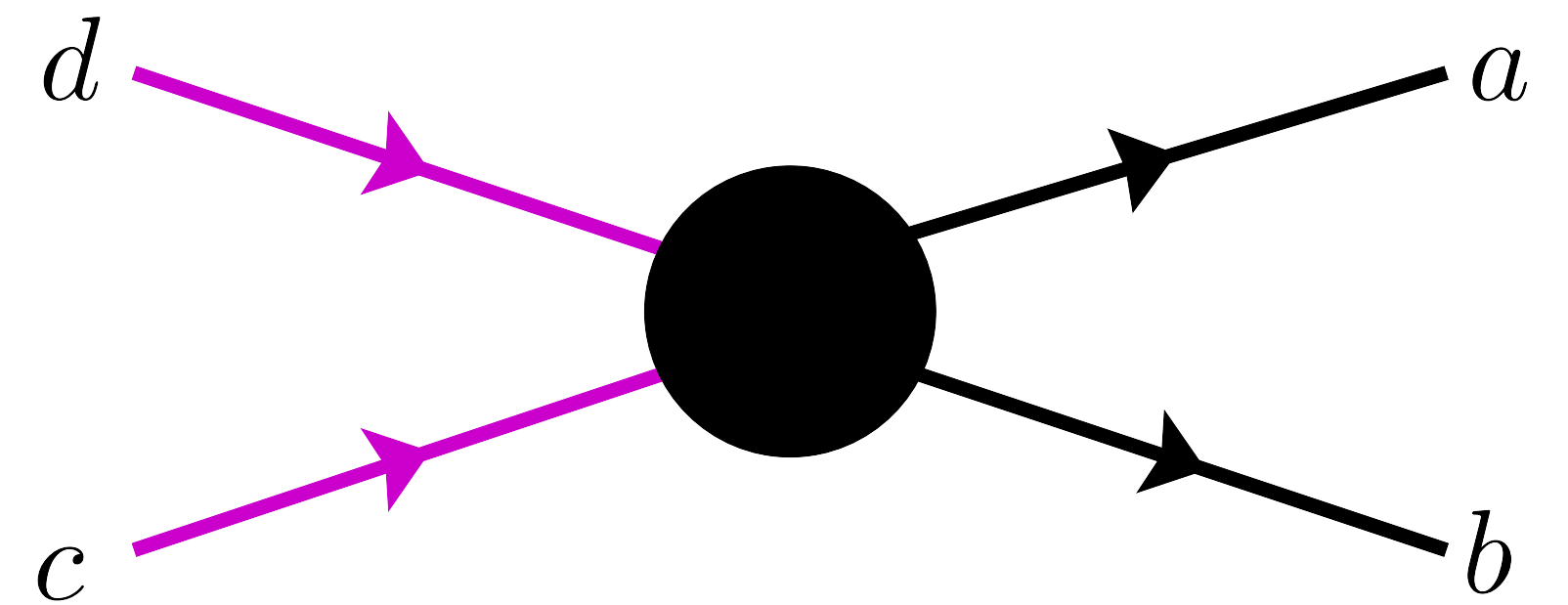
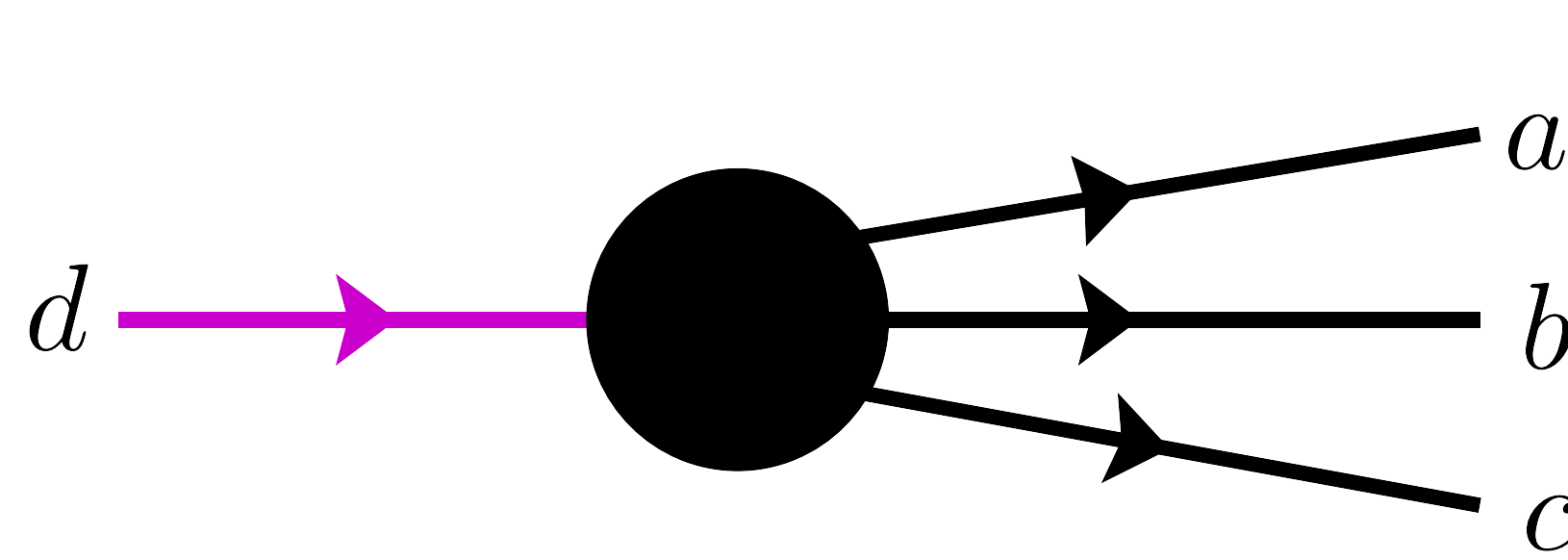
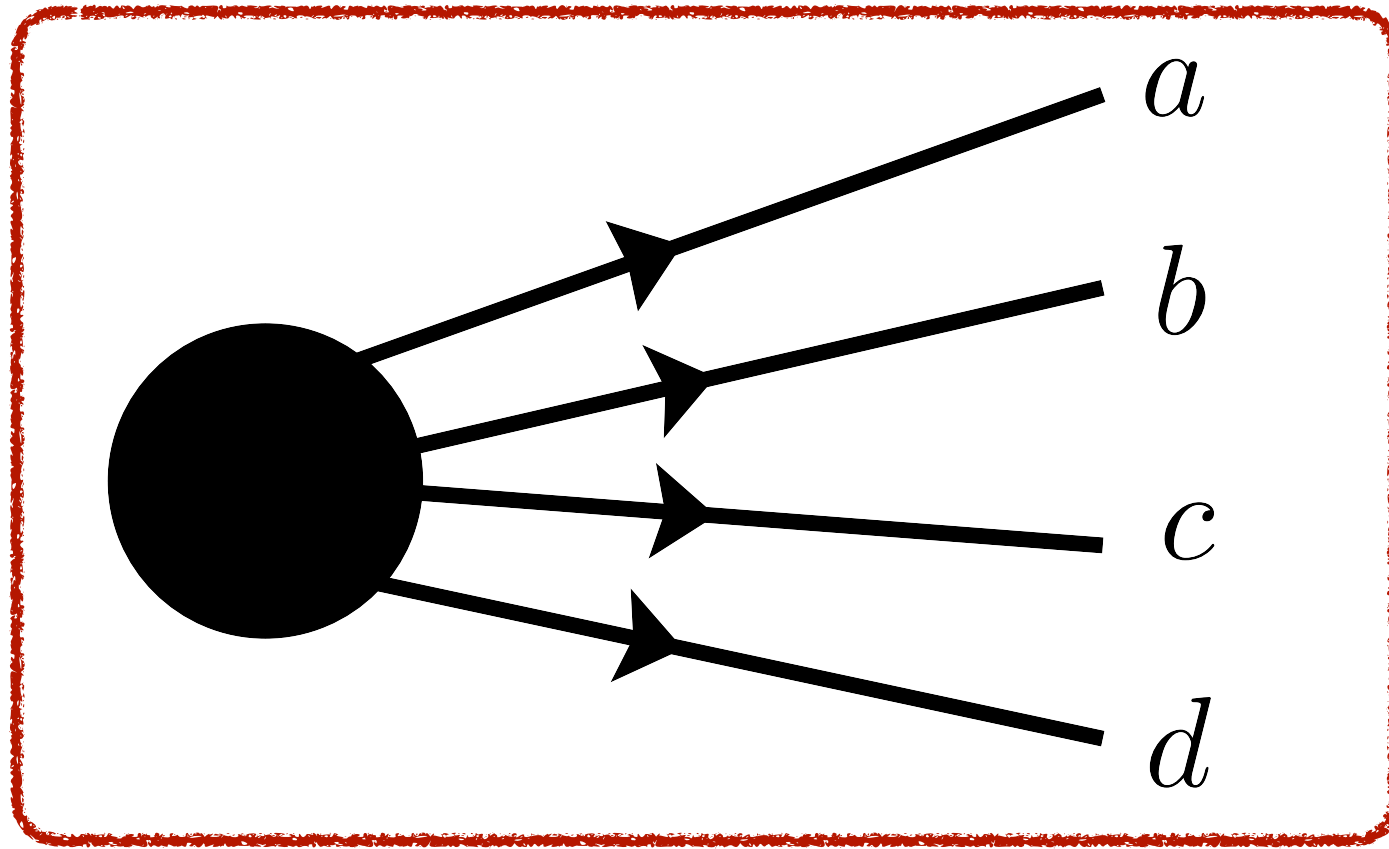
(abc,d)



(ab,cd)

Four-Particle Phase-Space

Final-Final Case



- All legs are outgoing: all four momenta are integrated over
- Phase space is symmetric under permutations of k_a, k_b, k_c, k_d and
 $S_{ab} \leftrightarrow S_{cd}, S_{ac} \leftrightarrow S_{bd}, S_{ad} \leftrightarrow S_{bc}$

$$s_{ab} = 2k_a \cdot k_b$$

Four-Particle Phase-Space

Final-Final Case

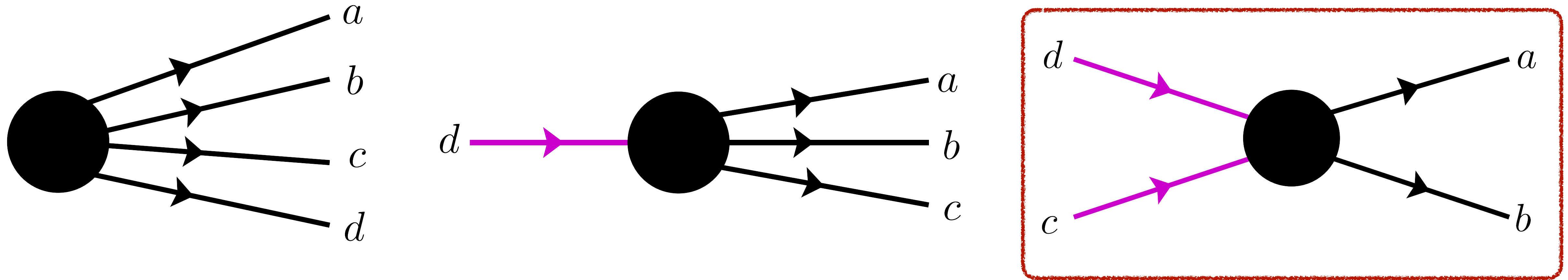
- All legs are outgoing: all four momenta are integrated over
- Phase space is symmetric under permutations of k_a, k_b, k_c, k_d and $s_{ab} \leftrightarrow s_{cd}, s_{ac} \leftrightarrow s_{bd}, s_{ad} \leftrightarrow s_{bc}$
- Exchanging momenta is a substitution in y, y', z, z', w'

$$y' = \frac{s_{ab}}{s_{abc}} \quad z' = \frac{s_{ac}}{s_{ac} + s_{bc}} \quad y = \frac{\bar{s}_{bc}^{(abc)}}{\bar{s}_{bcd}^{(abc)}} \quad z = \frac{\bar{s}_{bd}^{(abc)}}{\bar{s}_{bd}^{(abc)} + \bar{s}_{cd}^{(abc)}}$$

$$w' = \frac{1}{2} - \frac{2k_a \cdot \bar{k}_b^{(abc)} 2k_d \cdot \bar{k}_c^{(abc)} + 2k_a \cdot \bar{k}_c^{(abc)} 2k_d \cdot \bar{k}_b^{(abc)} - s_{abc} 2k_a \cdot k_d}{4 \left[2k_a \cdot \bar{k}_b^{(abc)} 2k_a \cdot \bar{k}_c^{(abc)} \left(2\bar{k}_b^{(abc)} \cdot k_d 2\bar{k}_c^{(abc)} \cdot k_d - s_{abc} k_d^2 \right) \right]^{1/2}}$$

Four-Particle Phase-Space

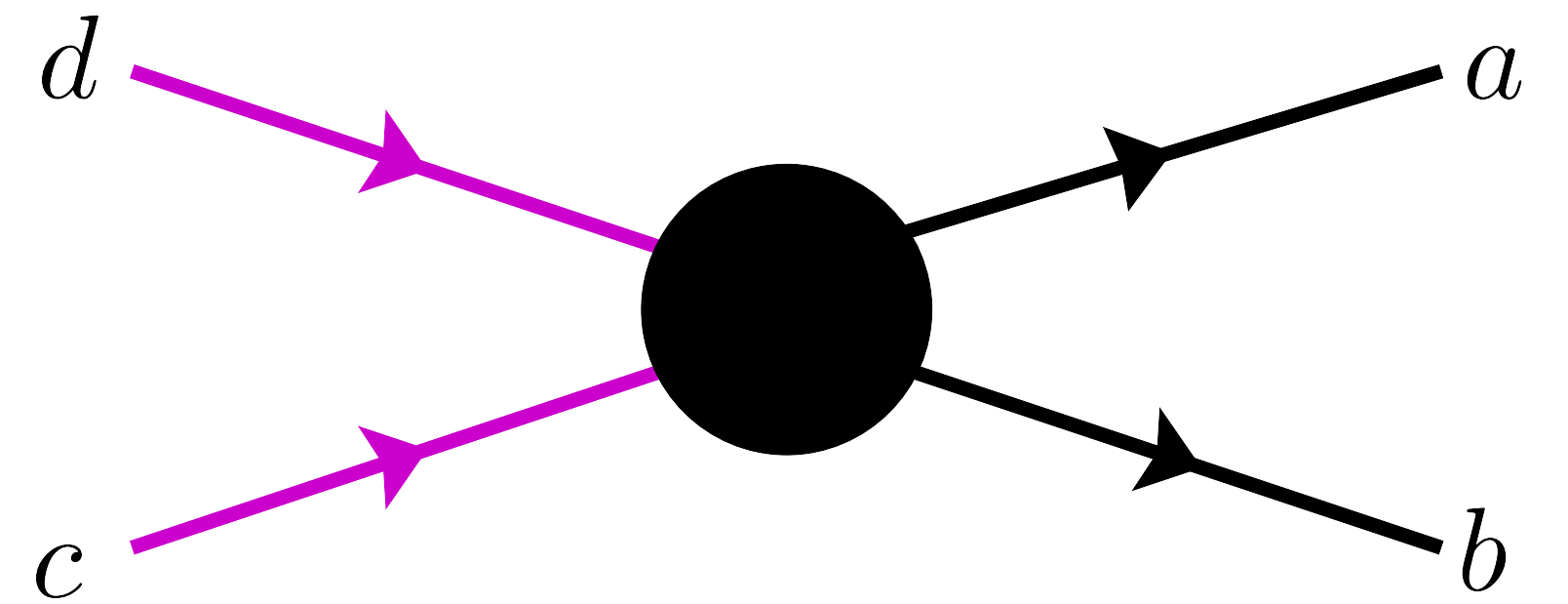
Initial-Initial Case



- Momentum carries a fraction x and recoil is $\bar{k}_d = xk_d$
- Now, for example, $s_{ab} \leftrightarrow s_{cd}$ is not a symmetry of phase space

Initial-Initial Mapping

Phase space (ab, cd)



- Problem: the symmetry is broken under the exchange of variables
- Solution: start from the Catani-Seymour mapping and derive a variable transformation
- Takeaway: apply the transformation

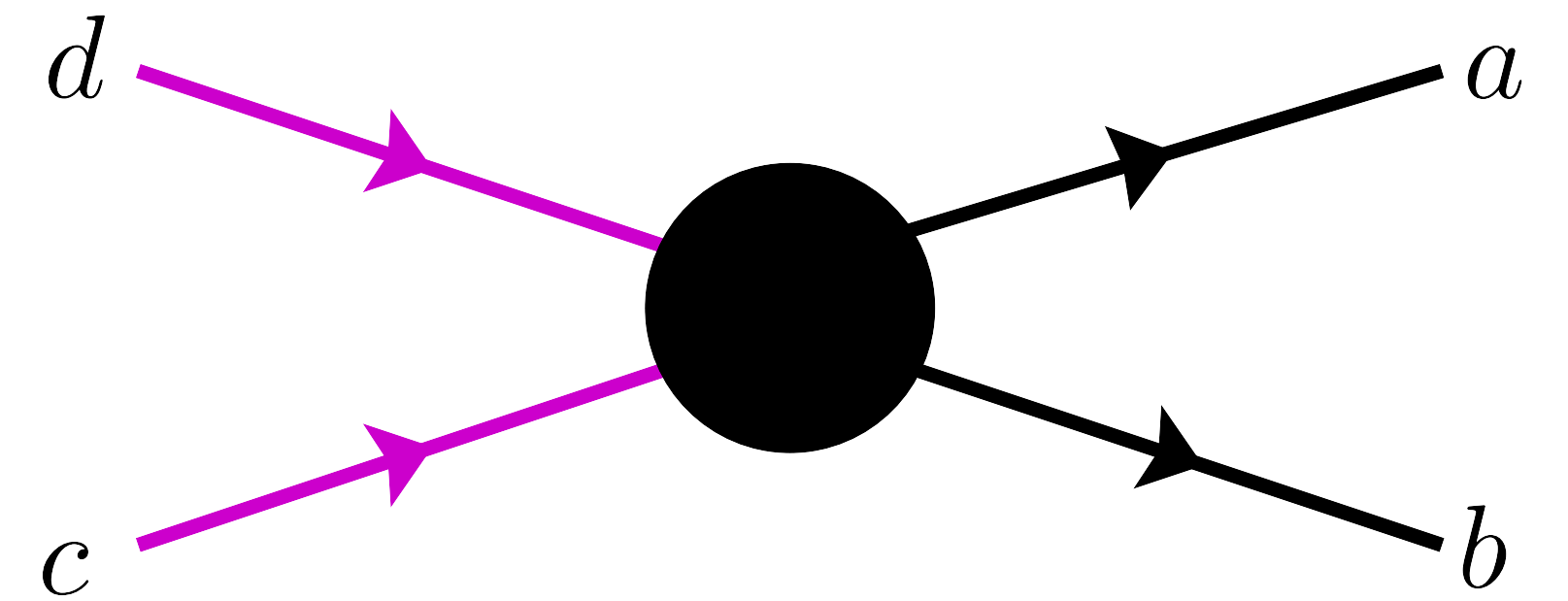
→ $y \leftrightarrow 1 - y$

$$y' \leftrightarrow (1 - y')(1 - z)$$

$$(1 - y')z' \leftrightarrow y'z'(1 - z) + (1 - z')z + 2(1 - 2w')\sqrt{y'z'(1 - z')z(1 - z)}$$

Initial-Initial Mapping

(ab, cd)



$$s_{ab} = (1 - y)(1 - y')z s^{(ab, cd)}$$

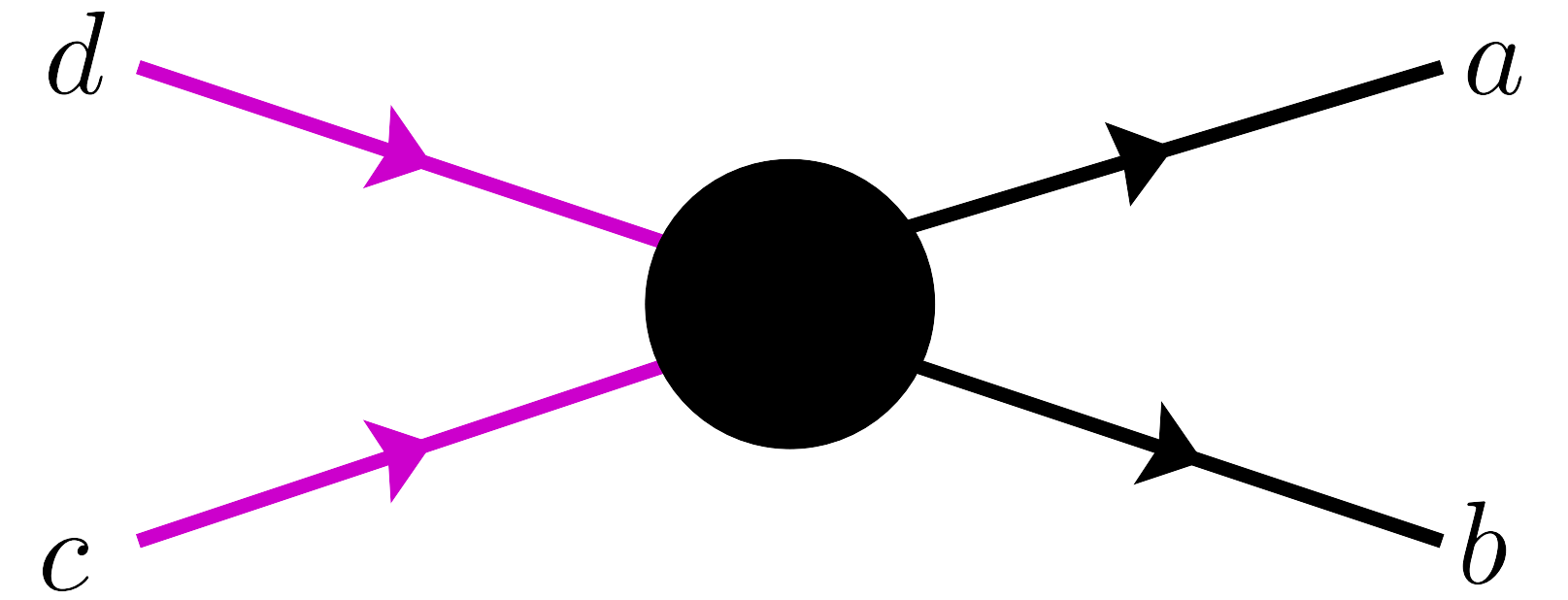
$$s_{ac} = (1 - y) \left[y'(1 - z')(1 - z) + z'z - 2(1 - 2w')\sqrt{y'z'(1 - z')z(1 - z)} \right] s^{(ab, cd)}$$

$$s_{bc} = (1 - y) \left[y'z'(1 - z) + z(1 - z') + 2(1 - 2w')\sqrt{y'z'(1 - z')z(1 - z)} \right] s^{(ab, cd)}$$

$$s_{ad} = (1 - y')(1 - z')s^{(ab, cd)} , \quad s_{bd} = (1 - y')z's^{(ab, cd)} , \quad s_{cd} = s^{(ab, cd)}$$

Initial-Initial Mapping

(ab, cd)



$$s_{ab} = (1 - y)(1 - y')zs^{(ab,cd)}$$

$$s_{ac} = (1 - y) \left[y'(1 - z')(1 - z) + z'z - 2(1 - 2w')\sqrt{y'z'(1 - z')z(1 - z)} \right] s^{(ab,cd)}$$

$$s_{bc} = (1 - y) \left[y'z'(1 - z) + z(1 - z') + 2(1 - 2w')\sqrt{y'z'(1 - z')z(1 - z)} \right] s^{(ab,cd)}$$

$$s_{ad} = (1 - y')(1 - z')s^{(ab,cd)}, \quad s_{bd} = (1 - y')z's^{(ab,cd)}, \quad s_{cd} = s^{(ab,cd)}$$

Out[95]=

$$\frac{2 (ss - 2 ss wp - 2 ss zp + 4 ss wp zp + yp zp - z zp - yp z zp - yp zp^2 + z zp^2 + yp z zp^2) (2 ss - 4 ss wp - 2 yp - z + 2 yp z - 4 ss zp + 8 ss wp zp + 2 yp zp - 2 z zp - 2 yp z zp - 2 yp zp^2 + 2 z zp^2 + 2 yp z zp^2)}{s^2 (-1 + y)^2 (-1 + yp)^2 z (-yp - z + yp z)^2 (-1 + zp) zp (-2 ss + 4 ss wp + yp - yp z - yp zp + z zp + yp z zp)} -$$

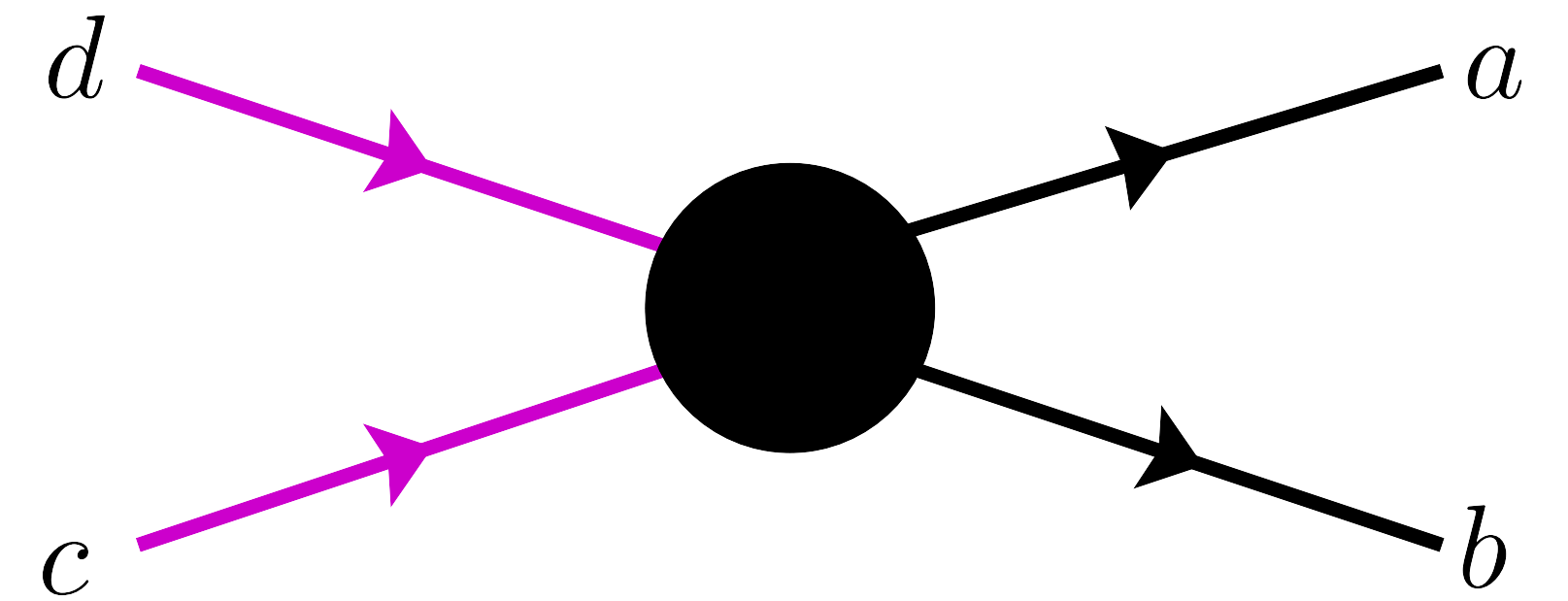
$$\frac{(-1 + zp) zp (-1 + \epsilon)}{s^2 (-1 + y)^2 (-1 + yp)^2 z^2} - \frac{(-2 ss + 4 ss wp - z - yp zp + z zp + yp z zp) (-2 ss + 4 ss wp + yp - yp z - yp zp + z zp + yp z zp) (-1 + \epsilon)}{s^2 (-1 + y)^2 (-1 + yp)^2 z^2 (-yp - z + yp z)^2} +$$

$$\frac{1}{s^2 (-1 + y)^2 (-1 + yp)^2 z^2 (-yp - z + yp z)} (-2 ss + 4 ss wp + z + 4 ss zp - 8 ss wp zp - 2 yp zp + 2 z zp + 2 yp z zp + 2 yp zp^2 -$$

$$2 z zp^2 - 2 yp z zp^2 + 2 ss \epsilon - 4 ss wp \epsilon + z \epsilon - 4 ss zp \epsilon + 8 ss wp zp \epsilon + 2 yp zp \epsilon - 2 z zp \epsilon - 2 yp z zp \epsilon - 2 yp zp^2 \epsilon + 2 z zp^2 \epsilon + 2 yp z zp^2 \epsilon)$$

Initial-Initial Mapping

(ab, cd)



$$s_{ab} = (1 - y)(1 - y')zs^{(ab,cd)}$$

$$s_{ac} = (1 - y) \left[y'(1 - z')(1 - z) + z'z - 2(1 - 2w')\sqrt{y'z'(1 - z')z(1 - z)} \right] s^{(ab,cd)}$$

$$s_{bc} = (1 - y) \left[y'z'(1 - z) + z(1 - z') + 2(1 - 2w')\sqrt{y'z'(1 - z')z(1 - z)} \right] s^{(ab,cd)}$$

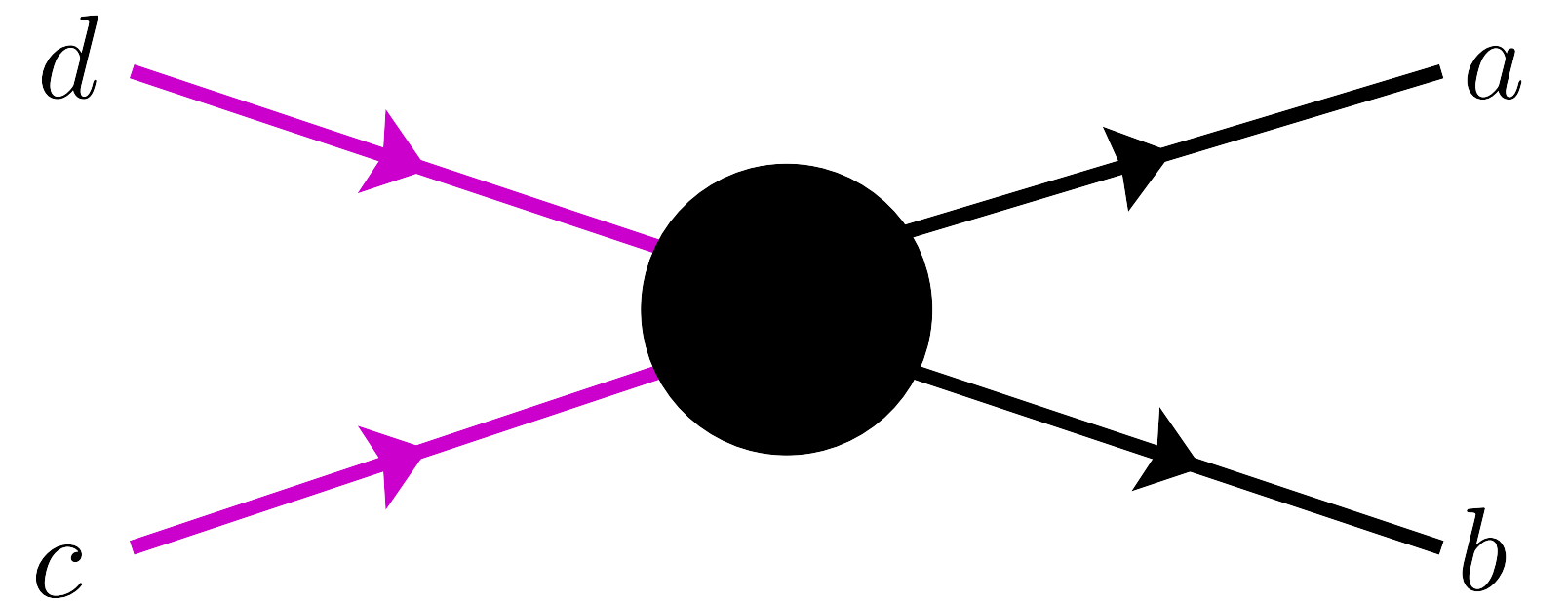
$$s_{ad} = (1 - y')(1 - z')s^{(ab,cd)}, \quad s_{bd} = (1 - y')z's^{(ab,cd)}, \quad s_{cd} = s^{(ab,cd)}$$

• Denominators, $s_{ab}, s_{ac}, s_{bc}, s_{ac} + s_{bc} - s_{ab}$

$s_{ad}, s_{bd}, s_{cd}, s_{cd} - s_{ad} - s_{bd}, s_{ad} + s_{bd}, s_{cd} - s_{ad}, s_{cd} - s_{bd}$

Initial-Initial Mapping

(ab, cd)



$$s_{ab} = (1 - y)(1 - y')zs^{(ab,cd)}$$

$$s_{ac} = (1 - y) \left[y'(1 - z')(1 - z) + z'z - 2(1 - 2w')\sqrt{y'z'(1 - z')z(1 - z)} \right] s^{(ab,cd)}$$

$$s_{bc} = (1 - y) \left[y'z'(1 - z) + z(1 - z') + 2(1 - 2w')\sqrt{y'z'(1 - z')z(1 - z)} \right] s^{(ab,cd)}$$

$$s_{ad} = (1 - y')(1 - z')s^{(ab,cd)}, \quad s_{bd} = (1 - y')z's^{(ab,cd)}, \quad s_{cd} = s^{(ab,cd)}$$

- Denominators,

$$s_{ab}, s_{ac}, s_{bc}, s_{ac} + s_{bc} - s_{ab}$$

$$s_{ad}, s_{bd}, s_{cd}, s_{cd} - s_{ad} - s_{bd}, s_{ad} + s_{bd}, s_{cd} - s_{ad}, s_{cd} - s_{bd}$$

Not so simple ${}_2F_1$ integrals!

Simple factors

Simple ${}_2F_1$ integrals

Integration Method

The pipeline

1 reduce the denominators

- apply variable transformations
- the measure

$$d\Phi_{rad,2} = s_{map}^{2-2\epsilon} d\Phi_{rad,2}^{inv} (s/s_{map})^{1-2\epsilon}$$

Integration Method

The pipeline

1 reduce the denominators

- apply variable transformations
- the measure

$$d\Phi_{rad,2} = s_{map}^{2-2\epsilon} d\Phi_{rad,2}^{inv} (s/s_{map})^{1-2\epsilon}$$



$$\frac{\pi^{2\epsilon-1}(4\pi)^{-4}}{\Gamma(1-2\epsilon)} \int_0^1 dy \int_0^1 dz \int_0^1 dw' \int_0^1 dy' \int_0^1 dz' (w'(1-w'))^{-\epsilon-\frac{1}{2}} \\ \times (y'(1-y')^2 z'(1-z') y^2 (1-y)^2 z(1-z))^{-\epsilon} (1-y') y (1-y)$$

Integration Method

The pipeline

1 reduce the denominators

- apply variable transformations
- the measure

$$d\Phi_{rad,2} = s_{map}^{2-2\epsilon} d\Phi_{rad,2}^{inv} (s/s_{map})^{1-2\epsilon}$$

$f(y, y', z, z', w')$

2 integrate y and w'

- $y \rightarrow$ factorised $\rightarrow$ Beta function
- only s_{ac}, s_{bc} carry w'
- $I_{w'} \rightarrow {}_2F_1(1, 1 + \epsilon; 1 - \epsilon, .) + \Theta - \text{split}$

Integration Method

The pipeline

1 reduce the denominators

- apply variable transformations
- the measure

$$d\Phi_{rad,2} = s_{map}^{2-2\epsilon} d\Phi_{rad,2}^{inv} (s/s_{map})^{1-2\epsilon}$$

$$f(y, y', z, z', w')$$

2 integrate y and w'

- $y \rightarrow$ factorised $\rightarrow$ Beta function
- only s_{ac}, s_{bc} carry w'
- $I_{w'} \rightarrow {}_2F_1(1, 1 + \epsilon; 1 - \epsilon, .) + \Theta$ – split

3 integrate z and z'

- rational str. $\rightarrow$ Beta $\times {}_2F_1$
- $I \sim (1 - z)^n$, $J \sim z^n$
- $p, q, n \geq -1, m = 0, 1$
- partial fraction until $p + q \geq m$
- $z \leftrightarrow (1 - z') \rightarrow$ four families

$$I_{w'z'}^{(n)}, J_{w'z'}^{(n)}, I_{w'z''}^{(n)}, J_{w'z'}^{(n)}$$

- $n \geq 0$, finite sum
- $n = -1$, integral rep. $\rightarrow t$
- everything exact in ϵ

Integration Method

The pipeline

1 reduce the denominators

- apply variable transformations
- the measure

$$d\Phi_{rad,2} = s_{map}^{2-2\epsilon} d\Phi_{rad,2}^{inv} (s/s_{map})^{1-2\epsilon}$$

$$f(y, y', z, z', w')$$

2 integrate y and w'

- $y \rightarrow$ factorised $\rightarrow$ Beta function
- only s_{ac}, s_{bc} carry w'
- $I_{w'} \rightarrow {}_2F_1(1, 1 + \epsilon; 1 - \epsilon, .) + \Theta -$ split

3 integrate z and z'

- rational str. $\rightarrow$ Beta $\times {}_2F_1$
- $I_{w'} \rightarrow$ four families

$$I_{w'z}^{(n)}, J_{w'z}^{(n)}, I_{w'z'}^{(n)}, J_{w'z'}^{(n)}$$

4 integrate y', t , expand in ϵ

- ${}_2F_1(-\epsilon, -2\epsilon; 1 - 2\epsilon; 1 - \omega), {}_2F_1(1, -2\epsilon; 1 - 2\epsilon; 1 - \omega)$
- expand $\rightarrow S_{n,p'}, Li_n$
- extract poles with plus distributions

Numerical Checks

Using PySecDec [\[1703.09692\]](#) [S. Borowka, G. Heinrich, S. Jahn, S.P. Jones, M.Kerner, J. Schlenk, T. Zirke]



Isolate



Integrate



Recombine



Compare

Numerical Checks

Using PySecDec [\[1703.09692\]](#) [S. Borowka, G. Heinrich, S. Jahn, S.P. Jones, M.Kerner, J. Schlenk, T. Zirke]



- An example: double-soft gluon kernel

Order	Analytic	Residual (σ)
ϵ^{-4}	$1/2$	0
ϵ^{-3}	$11/12$	$5\text{e-}15$ (0.95)
ϵ^{-2}	$67/36 - 2\pi^2/3$	$5\text{e-}11$ (0.34)
ϵ^{-1}	$101/27 - 77/72\pi^2 - 77/6\zeta_3$	$3\text{e-}8$ (0.54)
ϵ^0	$607/81 - 469/216\pi^2 - 341/18\zeta(3) - 1/720\pi^4$	$1\text{e-}6$ (0.71)

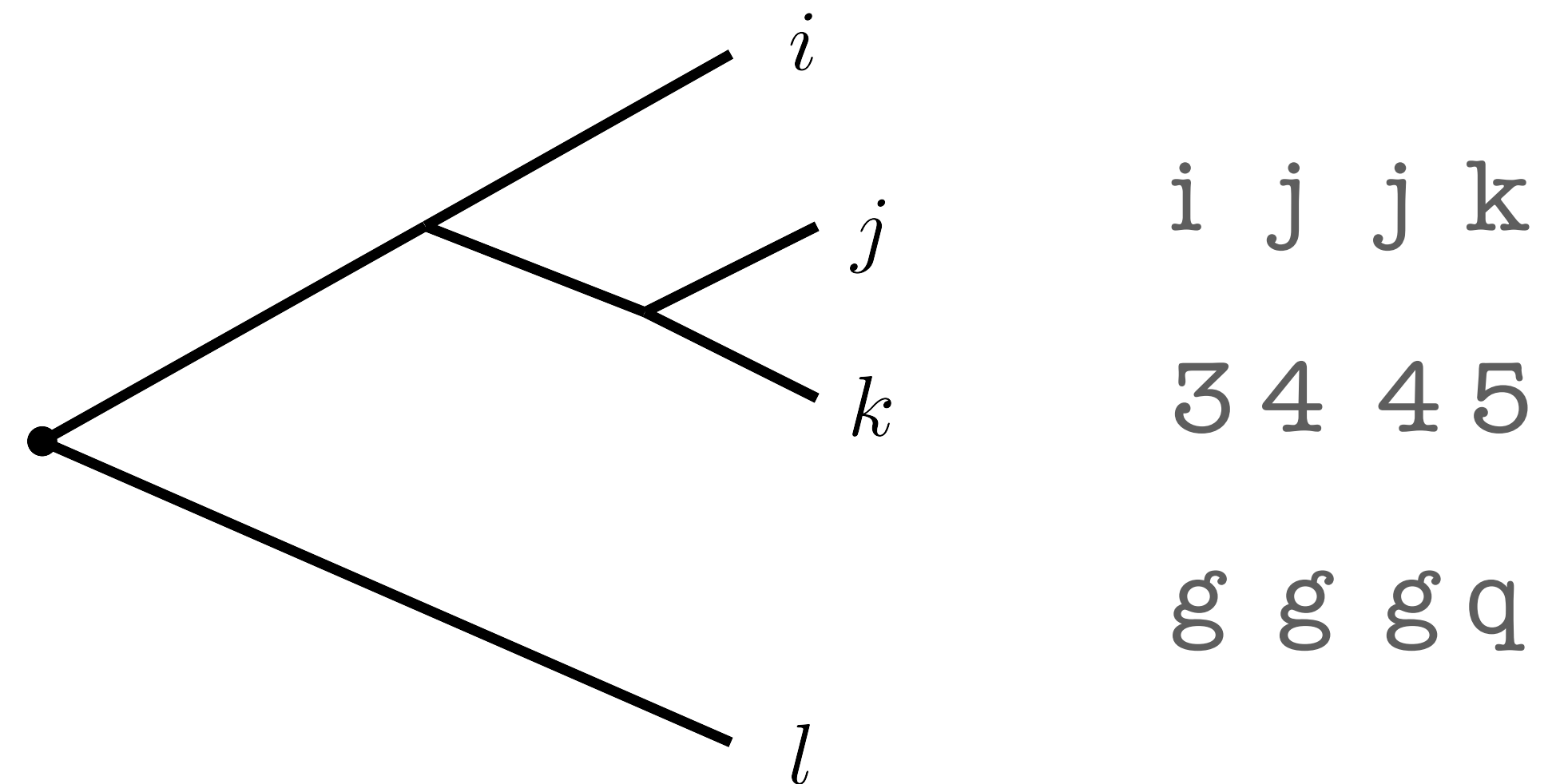
MadNkLO

The Numerical Code

- Generation with MadNkLO [inspired from MadGraph]
- Massive@NLO [[2503.14629](#)]
- Massless FSR@NNLO [[work in progress](#)]
- Implemented python-to-fortran coding structure
- Manages contributions, flavour channels, sectors and IR structure
- Designed for generality and automation

```
1 C      Collection of relevant limits for sector [3,4,4,5]
2 C      K1  ['S_g', 'HC_gg']
3 C      K2  ['SS_gg', 'SC_ggq', 'SS_gg_SC_ggq', 'CC_ggq',
4 C          'SS_gg_CC_ggq', 'SC_ggq_CC_ggq', 'SS_gg_SC_ggq_CC_ggq']
5 C      K12 ['S_SS_gg', 'S_SC_ggq', 'S_SS_gg_SC_ggq', 'S_HCC_ggq',
6 C          'S_SS_gg_HCC_ggq', 'HC_SS_gg', 'HC_CC_ggq', 'HC_SS_gg_CC_ggq']
```

Generator output for RR: the slots for $K^{(1)}$, $K^{(2)}$, and $K^{(12)}$ are filled by the flavours present in the sector



MadNkLO

IR cancellation

- $e^+e^- \rightarrow 2j \Rightarrow e^+e^- \rightarrow gg\bar{c}\bar{c}$
[for double-real emission]

- λ is a scaling parameter

- Two radiations
parametrised by (z, y) and
 (z', y')

- Pick a limit [we see $K^{(2)}$]

- As $\lambda \rightarrow 0$, both diverge, but
converge to each other

LIM = Sij	lambda	RR	LIM	RR-LIM / LIM
1.0000000000000000	1.4592925630444020E-004	-9.8848818099663466E-005	2.4762873154164402	
0.10000000000000001	0.32465746049526678	0.30088103438125813	7.9022681382704282E-002	
1.0000000000000000E-002	361.10501561462451	359.42597654386873	4.6714460843952027E-003	
1.0000000000000000E-003	365025.74726419029	364865.63048469671	4.3883765999245335E-004	
1.0000000000000000E-004	365420709.67716473	365404776.64389300	4.3603790344693719E-005	
1.0000000000000001E-005	365460234891.64777	365458642416.22424	4.3574709657823991E-006	
9.999999999999995E-007	365464187529198.25	365464028395036.88	4.3543043640669585E-007	
9.999999999999995E-008	3.6546458301770170E+017	3.6546456824211277E+017	4.0429607168406749E-008	
1.0000000000000000E-008	3.6546466218963757E+020	3.6546463651061393E+020	7.0264044920073194E-008	

LIM = SCijk				
	lambda	RR	LIM	RR-LIM / LIM
	1.0000000000000000	1.4592925630444020E-004	-9.8848818099663466E-005	2.4762873154164402
	0.10000000000000001	0.36847981958192427	0.41273303736477679	0.10721995521705999
	1.0000000000000000E-002	437.13587302226119	498.40434511879573	0.12292924950710667
	1.0000000000000000E-003	436502.08642417629	458555.46587680804	4.8093155776615340E-002
	1.0000000000000000E-004	433664969.80049700	440828237.23546636	1.6249565771675238E-002
	1.0000000000000001E-005	432510709943.67523	434792412519.50726	5.2477975929022475E-003
	9.999999999999995E-007	432120415551050.50	432843524315758.75	1.6706008617117330E-003
	9.999999999999995E-008	4.3199447688662534E+017	4.3222330025337075E+017	5.2941006792385091E-004
	1.0000000000000000E-008	4.3195443482968968E+020	4.3202677056472993E+020	1.6743345544465005E-004
	1.0000000000000001E-009	4.3194133074778713E+023	4.3196394732755040E+023	5.2357563410541569E-005
	1.0000000000000000E-010	4.3193815927285353E+026	4.3193975883633021E+026	3.7032096350348138E-006

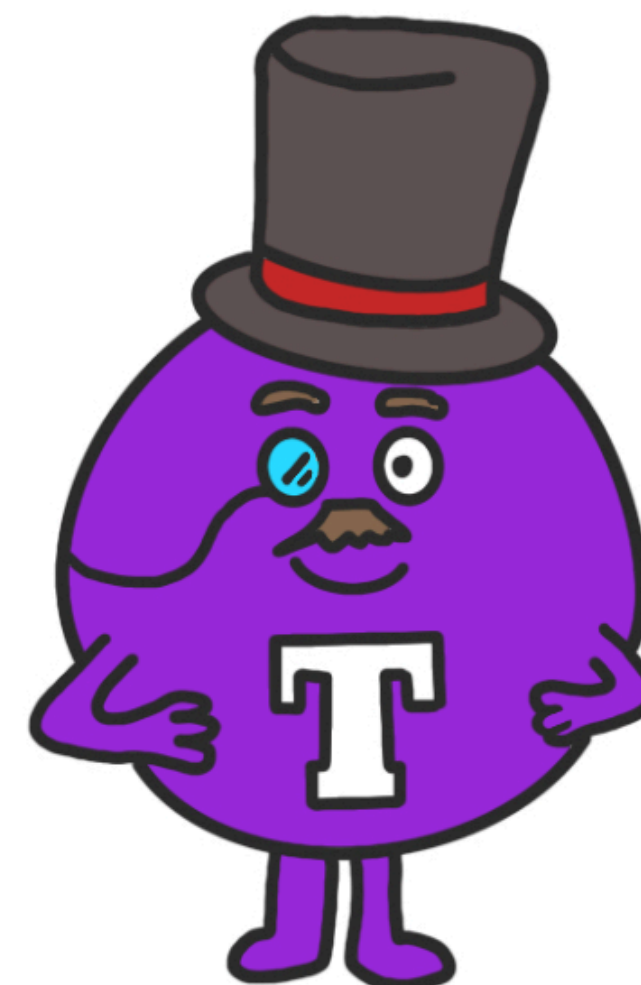
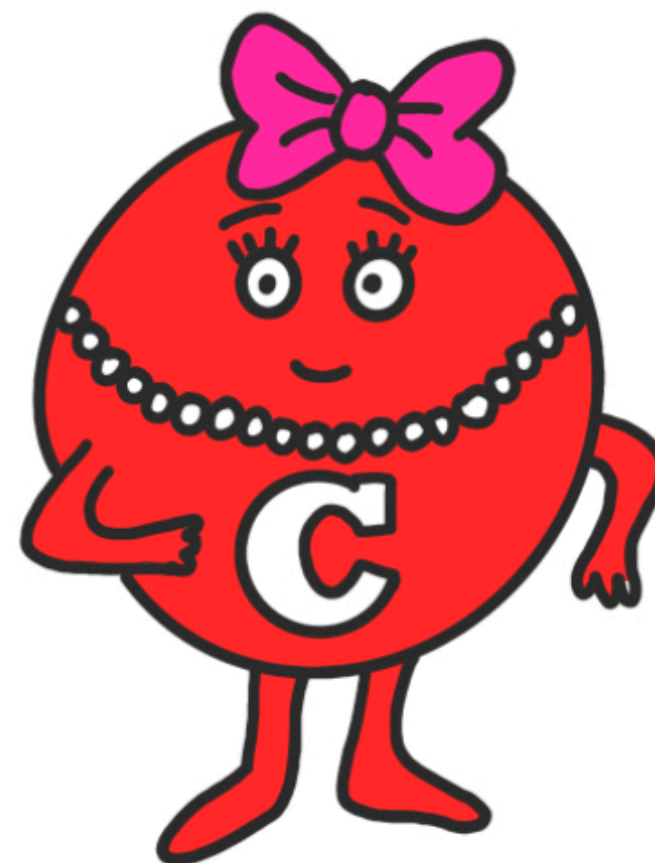
LIM = Cijk	lambda	RR	LIM	RR-LIM / LIM
1.0000000000000000	1.4592925630444020E-004	-9.8848818099663466E-005	2.4762873154164402	
0.10000000000000001	0.21978921509261024	0.19559008074044637	0.12372373006112108	
1.0000000000000000E-002	38.069150407992019	36.735216095804844	3.6312140064952828E-002	
1.0000000000000000E-003	4158.0880747565880	4110.0619184765637	1.1685020136588503E-002	
1.0000000000000000E-004	423568.50460217136	421955.40589063324	3.8229127747120939E-003	
1.0000000000000001E-005	42562797.044339187	42510688.097717173	1.2257845956817693E-003	
9.999999999999995E-007	4262379613.4950447	4260720284.5321884	3.8944799283825341E-004	
9.999999999999995E-008	426426671903.73401	426374079722.91736	1.2334750942371507E-004	
1.0000000000000000E-008	42648592471370.836	42646923086602.227	3.9144319162706995E-005	
1.0000000000000001E-009	4265042557128337.0	4264991366401472.0	1.2002539388067149E-005	

Status

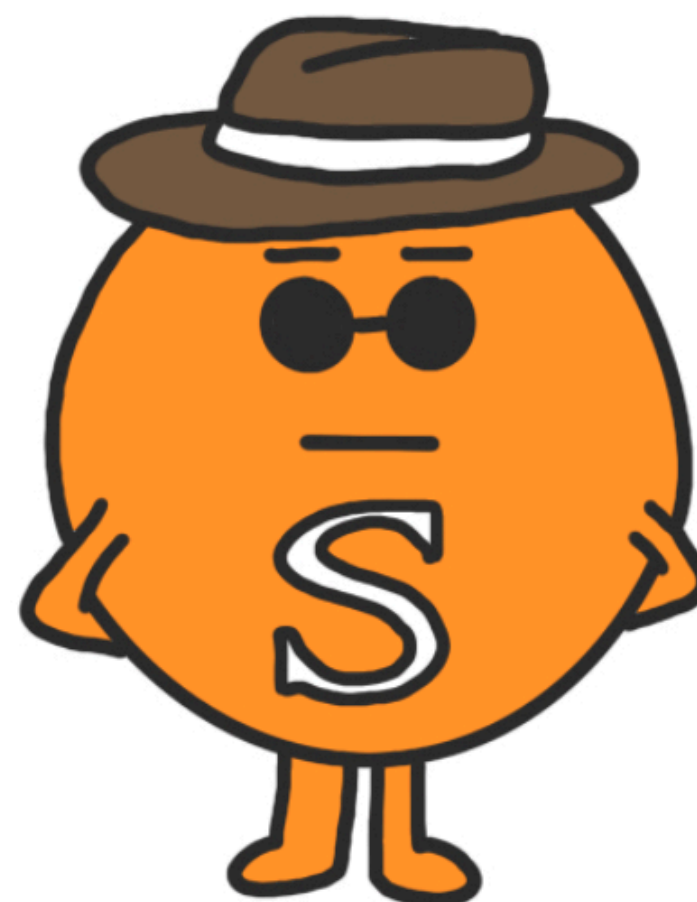
- Complete set of integrated counterterms for massless QCD
- New classes of mappings
- Standard methods to integrate
- Expressed in terms of same set of special functions

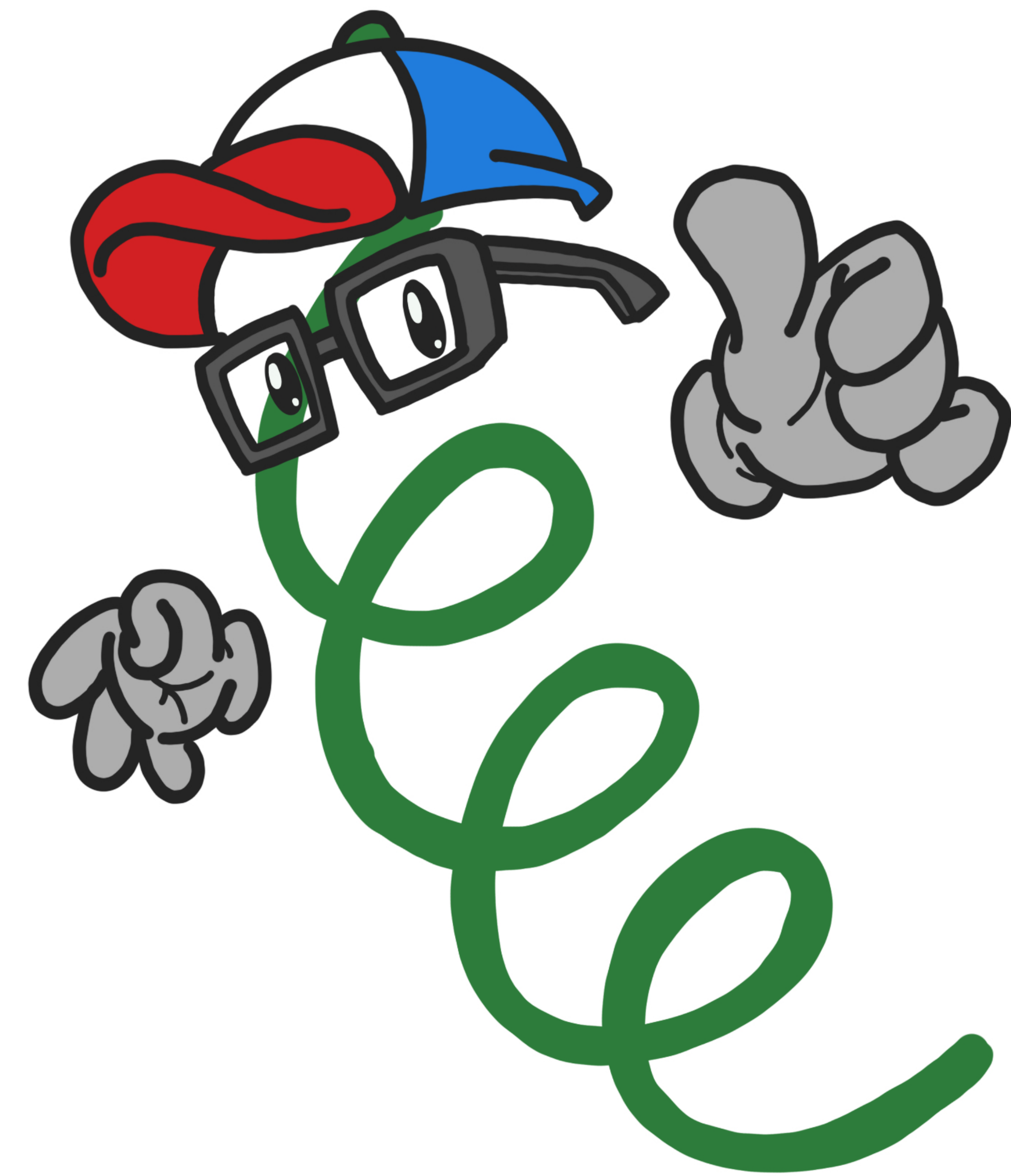
Outlook

- Complete the implementation in madNkLO
- Inclusion of massive colour particle @NNLO



Thank you!

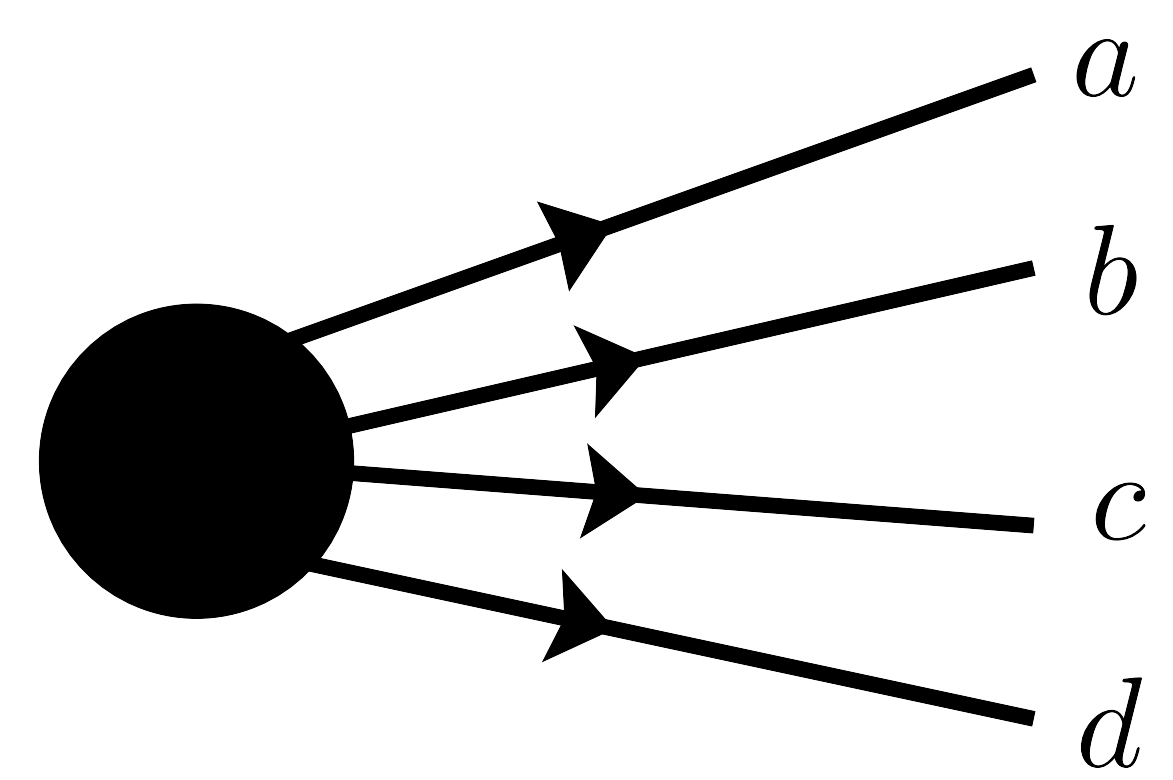




Backup

Final-Final Mapping

$(abcd)$



$$s_{cd} = (1 - y)(1 - y')(1 - z)s^{(abcd)}$$

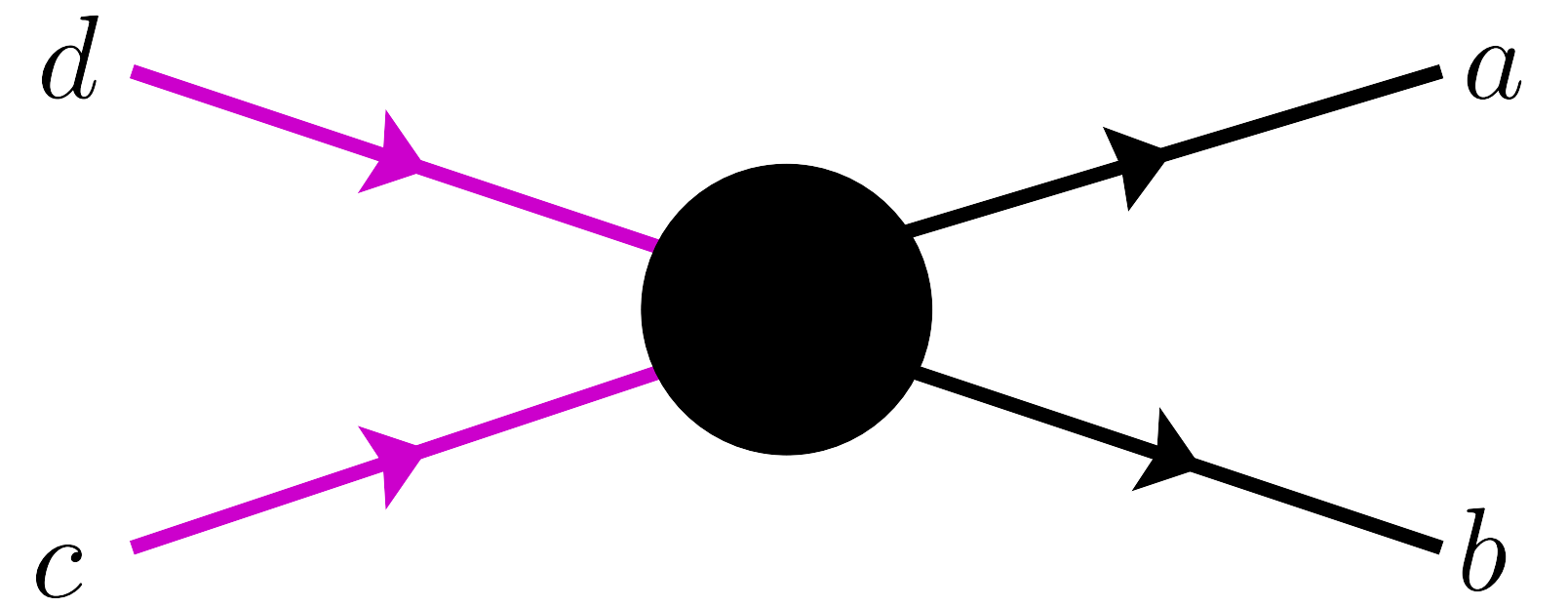
$$s_{ad} = (1 - y) \left[y'(1 - z')(1 - z) + z'z - 2(1 - 2w')\sqrt{y'z'(1 - z')z(1 - z)} \right] s^{(abcd)}$$

$$s_{bd} = (1 - y) \left[y'z'(1 - z) + z(1 - z) + 2(1 - 2w')\sqrt{y'z'(1 - z')z(1 - z)} \right] s^{(abcd)}$$

$$s_{bc} = (1 - y')(1 - z')s^{(abcd)} , \quad s_{ac} = (1 - y')z'y s^{(abcd)} , \quad s_{ab} = yy' s^{(abcd)}$$

Initial-Initial Mapping

The phase space



$$d\Phi_{n+2}(\{k\}) = \frac{\zeta_{n+2}}{\zeta_n} \iint d\Phi_n(xk_c, k_d; \{\bar{k}\}^{(ab,c;b,cd)}) d\Phi_{rad,2}^{(ab,c;b,cd)}$$

$$x = \frac{s_{ab,cd}}{s}$$

$$\int d\Phi_{rad,2}^{\text{inv}} \left(\frac{s_{cd} - s_{ad} - s_{bd}}{s_{ab,cd}} \right)^{1-2\epsilon} \frac{s_{cd} - s_{ad} - s_{bd}}{s}$$