Flavoured jets with exact anti-k_t kinematics

Ludovic Scyboz

In collaboration with
F. Caola, R. Grabarczyk, M. Hutt, G. Salam and J. Thaler

(paper from this morning)

Royal Society Research Grant (RP/R1/180112)

Les Houches, France
June 14th 2023

Flavour wish-list

Let's find an ambitious (and attainable...) set of requirements to impose on ourselves

We'd like to have an algorithm that:

1. is **IRC-safe** definition of jet flavour at all orders
   - Flavour-k_t attempt

Flavour wish-list

Let's find an ambitious (and attainable...) set of requirements to impose on ourselves

We'd like to have an algorithm that:

1. is **IRC-safe** definition of jet flavour at all orders
   - Flavour-k_t attempt
2. reproduces **exact** anti-k_t kinematics
   - CMP $\sim$ approximate
   - GHS $\equiv$ exact

Flavour wish-list

Let's find an ambitious (and attainable...) set of requirements to impose on ourselves

We'd like to have an algorithm that:

1. is **IRC-safe** definition of jet flavour at all orders
   - Flavour-k_t attempt
2. reproduces **exact** anti-k_t kinematics
   - CMP $\sim$ approximate
   - GHS $\equiv$ exact
3. for a tree-level event (1 parton $\equiv$ 1 jet), retains the flavour of the underlying parton

Flavour wish-list

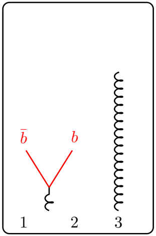
Let's find an ambitious (and attainable...) set of requirements to impose on ourselves

We'd like to have an algorithm that:

1. is **IRC-safe** definition of jet flavour at all orders
   - Flavour-k_t attempt
2. reproduces **exact** anti-k_t kinematics
   - CMP $\sim$ approximate
   - GHS $\equiv$ exact
3. for a tree-level event (1 parton $\equiv$ 1 jet), retains the flavour of the underlying parton
4. may make flavour information accessible for jet substructure (tracking flavour along the cluster sequence)

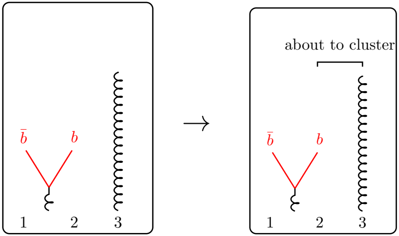
- Cluster particles with a generalised-k_t algorithm (e.g. anti-k_t and C/A),

$$d_{ij} = \min\left(p_{ti}^{2p}, p_{tj}^{2p}\right)\frac{\Delta R_{ij}^2}{R^2} \qquad d_{iB} = p_{ti}^{2p}$$



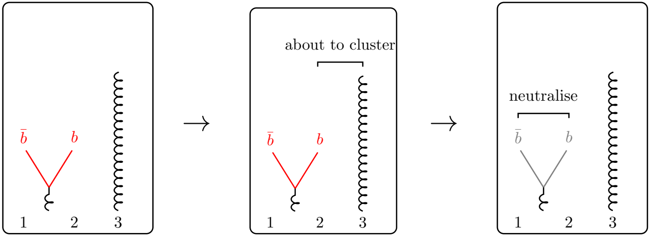
- Cluster particles with a generalised-k_t algorithm (e.g. anti-k_t and C/A),

$$d_{ij} = \min\left(p_{ti}^{2p}, p_{tj}^{2p}\right)\frac{\Delta R_{ij}^2}{R^2} \qquad d_{iB} = p_{ti}^{2p}$$



- Cluster particles with a generalised-k_t algorithm (e.g. anti-k_t and C/A),

$$d_{ij} = \min\left(p_{ti}^{2p}, p_{tj}^{2p}\right)\frac{\Delta R_{ij}^2}{R^2} \qquad d_{iB} = p_{ti}^{2p}$$

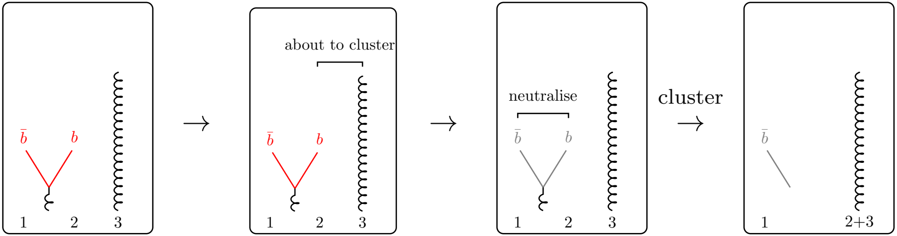


neutralise ≡ remove the (opposite) flavour of both 1 & 2 while maintaining kinematics

Interleaved Flavour Neutralisation (IFN)

- Cluster particles with a generalised-k_t algorithm (e.g. anti-k_t and C/A),

$$d_{ij} = \min\left(p_{ti}^{2p}, p_{tj}^{2p}\right) \frac{\Delta R_{ij}^2}{R^2} \qquad d_{iB} = p_{ti}^{2p}$$



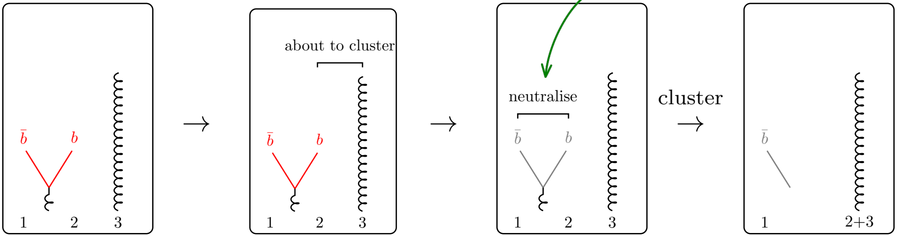
neutralise ≡ remove the (opposite) flavour of both 1 & 2
while maintaining kinematics

Interleaved Flavour Neutralisation (IFN)

- Cluster particles with a generalised-k_t algorithm (e.g. anti-k_t and C/A),

$$d_{ij} = \min\left(p_{ti}^{2p}, p_{tj}^{2p}\right) \frac{\Delta R_{ij}^2}{R^2} \qquad d_{iB} = p_{ti}^{2p}$$

based on a neutralisation distance u_{ik}

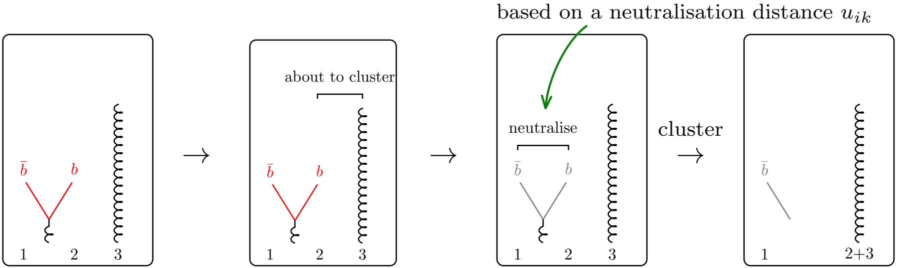


neutralise ≡ remove the (opposite) flavour of both 1 & 2
while maintaining kinematics

Interleaved Flavour Neutralisation (IFN)

- Cluster particles with a generalised-k_t algorithm (e.g. anti-k_t and C/A),

$$d_{ij} = \min\left(p_{ti}^{2p}, p_{tj}^{2p}\right) \frac{\Delta R_{ij}^2}{R^2} \qquad d_{iB} = p_{ti}^{2p}$$

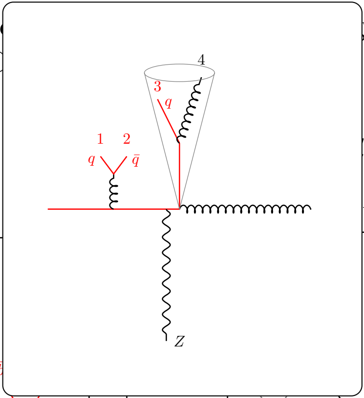


neutralise ≡ remove the (opposite) flavour of both 1 & 2 while maintaining kinematics

need to apply this recursively

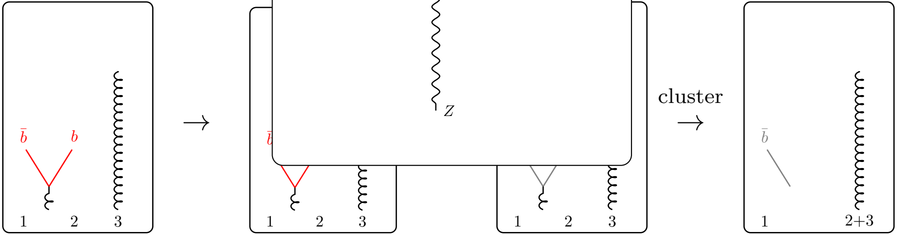
Interleaved Flavour Neutralisation (IFN)

- Cluster parti… …gorithm (e.g. anti-k_t and C…



d_{ij} … $_{iB} = p_{ti}^{2p}$

…utralisation distance u_{ik}



neutralise ≡ remove the (opposite) flavour of both 1 & 2 while maintaining kinematics

→ need to apply this recursively

- Generic form (with parameters α and ω):

$$u_{ik} = \max\left(p_{ti}, p_{tk}\right)^{\alpha} \min\left(p_{ti}, p_{tk}\right)^{2-\alpha} \cdot \Omega_{ik}^2$$

$$\Omega_{ik}^2 = 2\left[\frac{1}{\omega^2}\left(\cosh(\omega\Delta y_{ik}) - 1\right) - \left(\cos\Delta\phi_{ik} - 1\right)\right]$$

The choice of the neutralisation distance u_{ik}

- Generic form (with parameters α and ω):

$$u_{ik} = \max\left(p_{ti}, p_{tk}\right)^{\alpha} \min\left(p_{ti}, p_{tk}\right)^{2-\alpha} \cdot \Omega_{ik}^2$$

$$\Omega_{ik}^2 = 2\left[\frac{1}{\omega^2}\left(\cosh(\omega \Delta y_{ik}) - 1\right) - \left(\cos \Delta\phi_{ik} - 1\right)\right]$$

(similar to alternative proposal for ΔR^2 by [Catani et al. '93]!)

- Identical to flavour-k_t distance, except for angular part:
  - $\to \Delta R_{ik}^2$ for any ω when $\Delta R_{ik} \to 0$
  - $\to \exp\left(\omega \Delta y_{ik}\right)$ for $\Delta y_{ik} \gg 1$

eliminates divergence from interplay between ISR collinear and soft, large-angle flavour

- Generic form (with parameters α and ω):

$$u_{ik} = \max\left(p_{ti}, p_{tk}\right)^{\alpha} \min\left(p_{ti}, p_{tk}\right)^{2-\alpha} \cdot \Omega_{ik}^2$$

$$\Omega_{ik}^2 = 2\left[\frac{1}{\omega^2}\left(\cosh(\omega\Delta y_{ik}) - 1\right) - \left(\cos\Delta\phi_{ik} - 1\right)\right]$$

(similar to alternative proposal for ΔR^2 by [Catani et al. '93]!)

- Identical to flavour-k_t distance, except for angular part:
  - $\rightarrow \Delta R_{ik}^2$ for any ω when $\Delta R_{ik} \rightarrow 0$
  - $\rightarrow \exp\left(\omega\Delta y_{ik}\right)$ for $\Delta y_{ik} \gg 1$

- In the following:
  - $\alpha = 1,\ \omega = 2$
  - $\alpha = 2,\ \omega = 1$

eliminates divergence from interplay between ISR collinear and soft, large-angle flavour

The choice of the neutralisation distance u_{ik}

- Generic form (with parameters α and ω):

$$u_{ik} = \max\left(p_{ti}, p_{tk}\right)^{\alpha} \min\left(p_{ti}, p_{tk}\right)^{2-\alpha} \cdot \Omega_{ik}^2$$

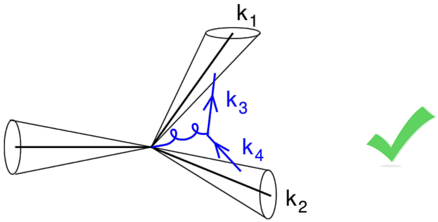
$$\Omega_{ik}^2 = 2\left[\frac{1}{\omega^2}\left(\cosh(\omega \Delta y_{ik}) - 1\right) - \left(\cos \Delta\phi_{ik} - 1\right)\right]$$

(similar to alternative proposal for ΔR^2 by [Catani et al. '93]!)

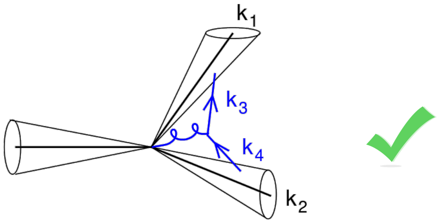
- Identical to flavour-k_t distance, except for angular part:
  - $\rightarrow \Delta R_{ik}^2$ for any ω when $\Delta R_{ik} \rightarrow 0$
  - $\rightarrow \exp\left(\omega \Delta y_{ik}\right)$ for $\Delta y_{ik} \gg 1$

- In the following:
  - $\alpha = 1,\ \omega = 2$
  - $\alpha = 2,\ \omega = 1$

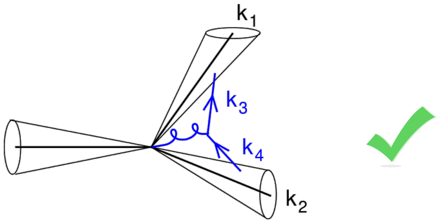
Need $\alpha + \omega > 2$ from IRC safety too



- All of the above algorithms take care of the “original” issue with soft, large-angle gluon splittings at NNLO

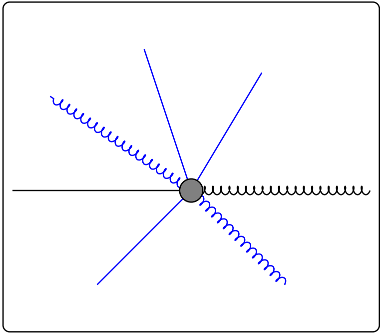


- All of the above algorithms take care of the “original” issue with soft, large-angle gluon splittings at NNLO
- Safety **at any order**, for generic configurations?



- All of the above algorithms take care of the "original" issue with soft, large-angle gluon splittings at NNLO
- Safety **at any order**, for generic configurations?
- Complicated (nested) structure of soft/collinear divergences $\rightarrow$ need a systematic framework to check for bad behaviour

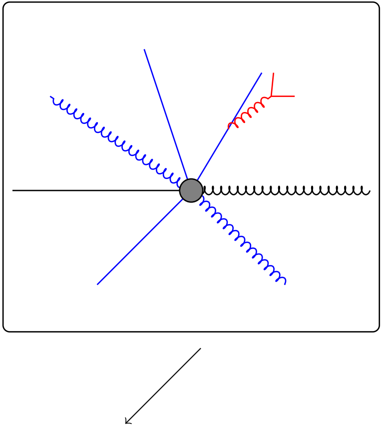
- Implemented such a fixed-order framework:



Cluster "hard" event

Set of hard jets
$\mathcal{J}_{\text{hard}} = \{(p_1, f_1), ...\}$

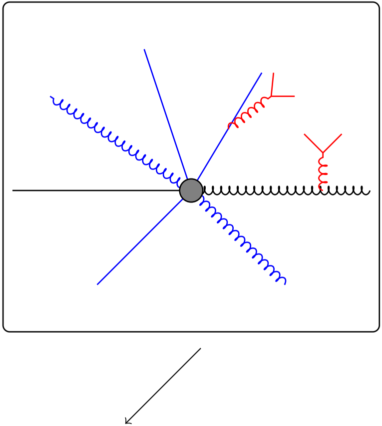
- Implemented such a fixed-order framework:



FDS = FS double-soft

Set of hard jets
$\mathcal{J}_{\text{hard}} = \{(p_1, f_1), ...\}$

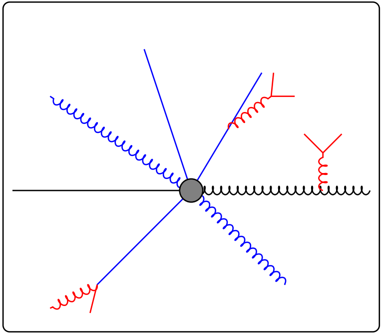
- Implemented such a fixed-order framework:



FDS = FS double-soft
IDS = IS double-soft

Set of hard jets
$\mathcal{J}_{\text{hard}} = \{(p_1, f_1), ...\}$

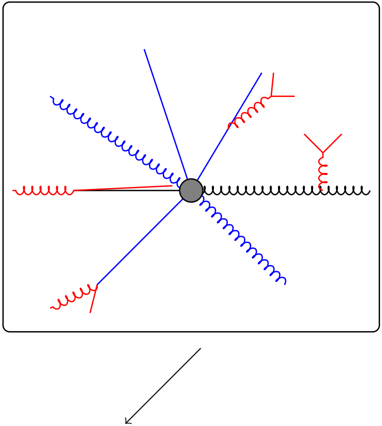
- Implemented such a fixed-order framework:



FDS = FS double-soft
IDS = IS double-soft
FC = final hard-collinear

Set of hard jets
$\mathcal{J}_{\text{hard}} = \{(p_1, f_1), ...\}$

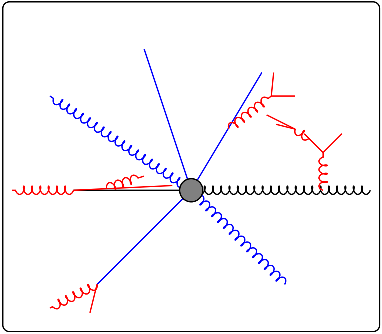
- Implemented such a fixed-order framework:



FDS = FS double-soft
IDS = IS double-soft
FC = final hard-collinear
IC = IS hard-collinear

Set of hard jets
$\mathcal{J}_{\text{hard}} = \{(p_1, f_1), ...\}$

- Implemented such a fixed-order framework:



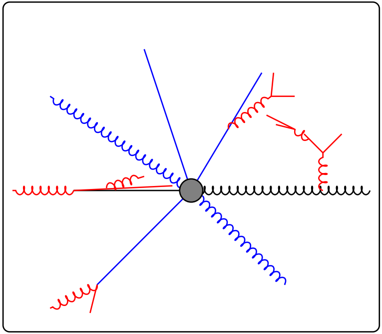
FDS = FS double-soft
IDS = IS double-soft
FC = final hard-collinear
IC = IS hard-collinear

possibly nested

Set of hard jets
$\mathcal{J}_{\text{hard}} = \{(p_1, f_1), ...\}$

Systematic IRC-safety tests

- Implemented such a fixed-order framework:

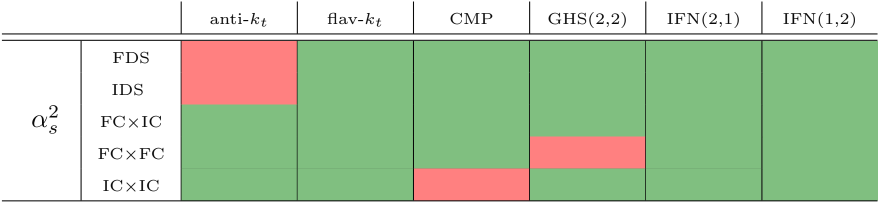


Set of hard jets
$\mathcal{J}_{\text{hard}} = \{(p_1, f_1), ...\}$

$\stackrel{!}{=}$

Set of hard+IRC jets
$\mathcal{J}_{\text{hard+IRC}} = \{(\tilde{p}_1, \tilde{f}_1), ...\}$

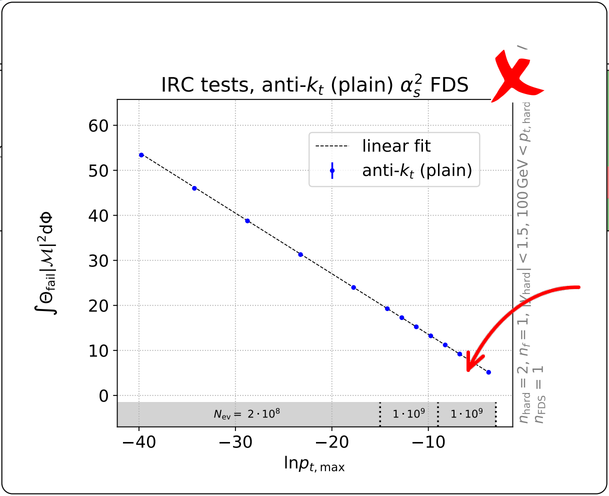
IRC-safety tests: results



- Systematic tests of all configurations at given order
  - DS = double-soft
  - FC = final-state collinear
  - IC = initial-state collinear

IRC-safety tests: results

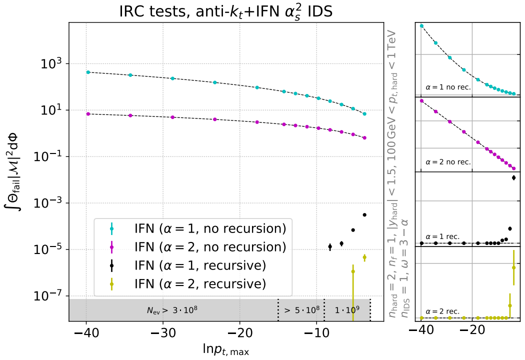
IFN(2,1)	IFN(1,2)



$\mathcal{O}(\alpha_s^2 L)$ divergence

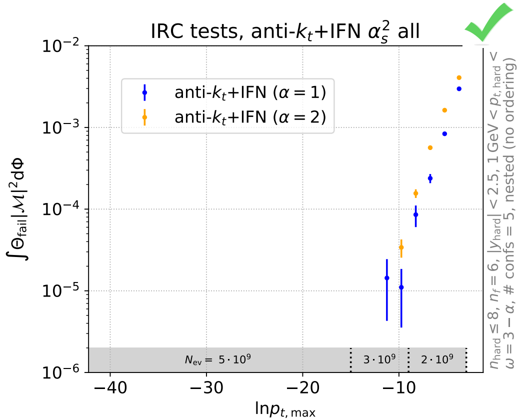
IRC-safety tests: results

IFN(2,1)	IFN(1,2)



divergent without recursion!

IRC-safety tests: results



IFN(2,1)	IFN(1,2)

IRC-safety tests: results

		anti-k_t	flav-k_t	CMP	GHS(2,2)	IFN(2,1)	IFN(1,2)
α_s^2	FDS						
	IDS						
	FC×IC						
	FC×FC						
	IC×IC						
α_s^3			marginal	+IC×FDS			
α_s^4					+IDS×FDS		

Fix by replacing angular factor in CMP (for flavoured pairs)

$$S_{ij} \to \bar{S}_{ij} = S_{ij} \frac{\Omega_{ij}^2}{\Delta R_{ij}^2} \quad \text{with } \omega > 1$$

IRC-safety tests: results

		anti-k_t	flav-k_t	CMP	GHS(2,2)	IFN(2,1)	IFN(1,2)
α_s^2	FDS						
	IDS						
	FC×IC						
	FC×FC						
	IC×IC						
α_s^3			marginal	+IC×FDS			
α_s^4					+IDS×FDS		
α_s^5							
α_s^6							

(TBC) Fix $\alpha\beta < 2$, and modification of the algorithm

IRC-safety tests: results

		anti-k_t	flav-k_t	CMP	GHS(2,2)	IFN(2,1)	IFN(1,2)
α_s^2	FDS						
	IDS						
	FC×IC						
	FC×FC						
	IC×IC						
α_s^3			marginal	+IC×FDS			
α_s^4					+IDS×FDS		
α_s^5							
α_s^6							

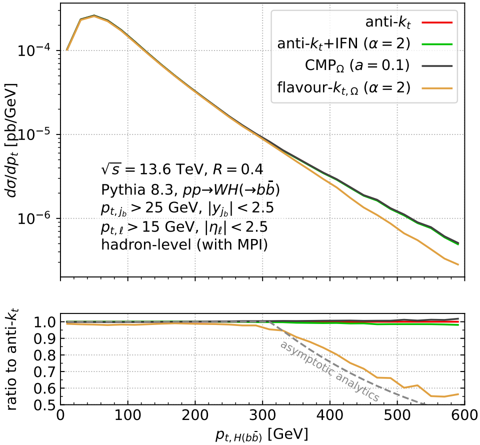
IRC-safety tests: results

		anti-k_t	flav-k_t	CMP	GHS(2,2)	IFN(2,1)	IFN(1,2)
α_s^2	FDS						
	IDS						
	FC×IC						
	FC×FC						
	IC×IC						
α_s^3			marginal	+IC×FDS			
α_s^4					+IDS×FDS		
α_s^5							
α_s^6							

In contact with both sets of authors (CMP & GHS)

$$|\eta_\ell| < 2.5\,, \quad p_{t\ell} > 15\,\text{GeV}$$
$$|\eta_{j_b}| < 2.5\,, \quad p_{tj_b} > 25\,\text{GeV}$$

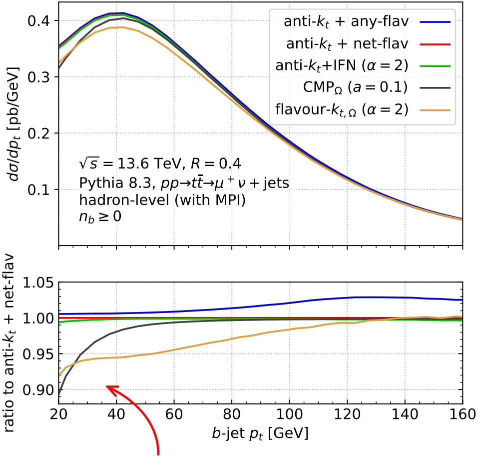
- Standard anti-k_t: one needs $m_b > 0$ to regulate $g \to b\bar{b}$ divergence
- Large differences between anti-k_t and flavour-k_t in e.g. the tail of $p_{t,H}$ (at NNLO: [Behring et al. '20])



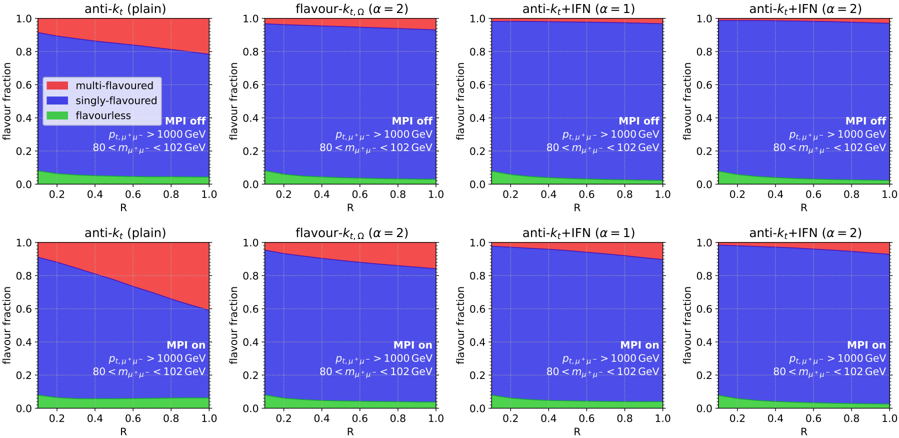
Semi-leptonic $t\bar{t}$: b-jet p_t distribution

$$p_{t\mu} > 30\,\text{GeV}\,, \quad |\eta_\mu| < 2.4$$
$$p_{t,j_b} > 20\,\text{GeV}\,, \quad p_{t,\text{miss}} > 30\,\text{GeV}$$

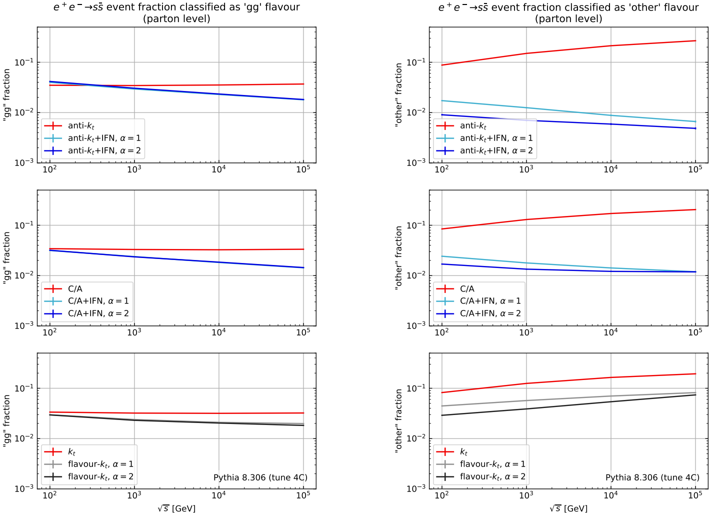
- CMP (with $a = 0.1$, corrected) can still differ from exact anti-k_t kinematics
- FN reproduces anti-k_t kinematics instead (as does GHS by construction)



Flavour-k_t, CMP: $\mathcal{O}(10\%)$ disagreement with anti-k_t


anti-k_t (plain)
flavour-$k_{t,\Omega}$ ($\alpha = 2$)
anti-k_t+IFN ($\alpha = 1$)
anti-k_t+IFN ($\alpha = 2$)
flavour fraction
R
multi-flavoured
singly-flavoured
flavourless
MPI off
MPI on
$p_{t,\mu^+\mu^-} > 1000$ GeV
$80 < m_{\mu^+\mu^-} < 102$ GeV

$e^+e^- \to q\bar{q}$



e+e−→ss̄ event fraction classified as 'gg' flavour (parton level)
e+e−→ss̄ event fraction classified as 'other' flavour (parton level)
"gg" fraction
"other" fraction
anti-kt
anti-kt+IFN, α = 1
anti-kt+IFN, α = 2
C/A
C/A+IFN, α = 1
C/A+IFN, α = 2
kt
flavour-kt, α = 1
flavour-kt, α = 2
Pythia 8.306 (tune 4C)
√s [GeV]

Conclusions

- Jet flavour is a difficult problem!
- Multiple recent attempts to define a IRC-safe flavour algorithm with kinematics identical (or close) to anti-k_t

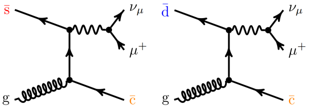
- Our proposal: definition of flavour *interleaved* with the kinematic clustering
  - Keep kinematics unchanged & neutralise flavour based on the u_{ik} distances

- As our project evolved, it became clear that a set of systematic tests for divergences was needed

- Release of `IFNPlugin` as a FastJet Plugin

Backup

Heavy-flavoured jets

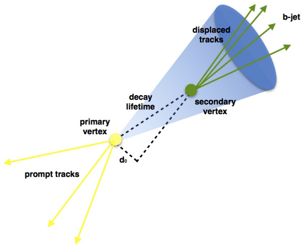


[see e.g. Czakon et al, 2212.00467]

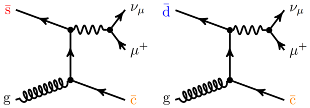
- Heavy-flavour tagging useful for many pheno applications:
  - Extract information on the partonic content of the proton (e.g. $W + c$)
  - Identify signal events ($t\bar{t}$, $WH(H \to b\bar{b})$, ...)

Experimental definition of a flavoured (b)-jet j

$\approx$ at least one (ghost-associated) B-hadron ($\Delta R_{Bj} < R$, $p_{t,B} > p_{t,\mathrm{cut}}$)



Heavy-flavoured jets

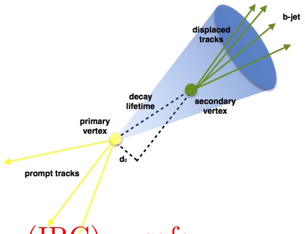


[see e.g. Czakon et al, 2212.00467]

- Heavy-flavour tagging useful for many pheno applications:
  - Extract information on the partonic content of the proton (e.g. $W + c$)
  - Identify signal events ($t\bar{t}$, $WH(H \to b\bar{b})$, ...)

Experimental definition of a flavoured (b)-jet j

$\approx$ at least one (ghost-associated) B-hadron ($\Delta R_{Bj} < R$, $p_{t,B} > p_{t,\text{cut}}$)

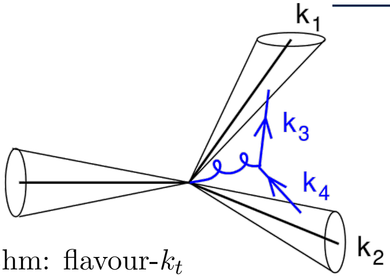


this definition is infrared and collinear (IRC) unsafe (divergent for $m_b = 0$, or logarithms $\ln \frac{m_b}{p_t}$)

The problem of jet flavour & IRC safety

First appearance at order $\mathcal{O}(\alpha_s^2)$

- Soft, large-angle gluon splitting to $q\bar{q}$
  → flavour *pollution*

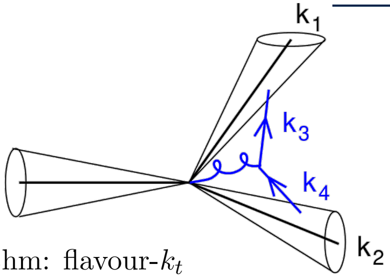


- Proposal for IRC-safe algorithm: flavour-k_t
  [Banfi,Salam,Zanderighi '06]
- More recently: multiple alternative proposals
  - "Practical jet flavour through NNLO"
    [Caletti,Larkoski,Marzani,Reichelt '22]
  - "Infrared-safe flavoured anti-k_t jets"
    [Czakon,Mitov,Poncelet '22]
  - "A dress of flavour to suit any jet"
    [Gauld,Huss,Stagnitto '22]

Different kinematics than anti-k_t

First appearance at order $\mathcal{O}(\alpha_s^2)$

- Soft, large-angle gluon splitting to $q\bar{q}$
  $\rightarrow$ flavour *pollution*



- Proposal for IRC-safe algorithm: flavour-k_t
  [Banfi,Salam,Zanderighi '06]
- More recently: multiple alternative proposals
  - "Practical jet flavour through NNLO"
    [Caletti,Larkoski,Marzani,Reichelt '22]
  - "Infrared-safe flavoured anti-k_t jets"
    [Czakon,Mitov,Poncelet '22] "CMP"
  - "A dress of flavour to suit any jet"
    [Gauld,Huss,Stagnitto '22] "GHS"

Different kinematics than anti-k_t

Flavour-k_t

New distances

$$d_{ij} = \max(p_{ti}, p_{tj})^{\alpha} \cdot \min(p_{ti}, p_{tj})^{2-\alpha} \frac{\Delta R_{ij}^2}{R^2}$$

if the softer of i and j is flavoured

... and some new beam distances d_{iB}

CMP

New distances

$$d_{ij} = d_{ij}^{\mathrm{anti}-k_t} \times \mathcal{S}_{ij}$$

$$\mathcal{S}_{ij} = 1 - \Theta(1-\kappa)\cos\left(\frac{\pi}{2}\kappa\right)$$

$$\kappa = \frac{1}{a}\frac{p_{ti}^2 + p_{tj}^2}{2p_{t,\max}^2}$$

if i and j are oppositely flavoured

GHS

1. Cluster with anti-k_t
2. "Accumulation" step with C/A + SoftDrop
3. Flavour "dressing" with flav-k_t distances

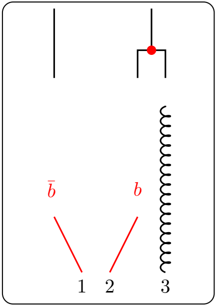
   $d_{\hat{f}_i \hat{f}_j}$, $d_{\hat{f}_i j_k}$, $d_{\hat{f}_i B}$

... assigns flavour $\hat{f}_i$ to jet j_k if $d_{\hat{f}_i j_k}$ is smaller

$i = 2, j = 3$

When pseudojets i and j recombine (where $p_{ti} < p_{tj}$):

1. If i is flavourless, combine i and j and apply the usual flavour summation.

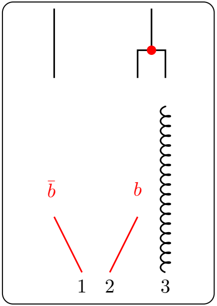


Flavour neutralisation: the “simple” algorithm

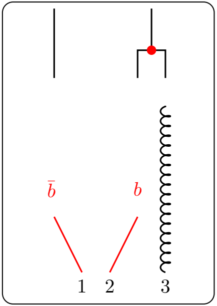
$i = 2, j = 3$
$K = \{1\}$

When pseudojets i and j recombine (where $p_{ti} < p_{tj}$):

1. If i is flavourless, combine i and j and apply the usual flavour summation.
2. If i is flavoured, identify all pseudojets carrying flavour at this stage (including jets that were declared as *beam jets* earlier). → list K.



$i = 2, j = 3$
$K = \{1\}$
$E = \{2, 3\}$

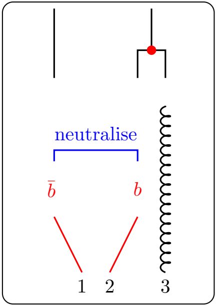


When pseudojets i and j recombine (where $p_{ti} < p_{tj}$):

1. If i is flavourless, combine i and j and apply the usual flavour summation.
2. If i is flavoured, identify all pseudojets carrying flavour at this stage (including jets that were declared as *beam jets* earlier). $\rightarrow$ list K.
   Initialise a set of particles to be excluded from consideration, $E = \{i, j\}$.
3. **Neutralisation step**: $\forall k \in K \setminus E$, in order of decreasing $u_{ik} < u_{ij}$, neutralise as much flavour in i from k as possible.

When pseudojets i and j recombine (where $p_{ti} < p_{tj}$):

$i = 2, j = 3$
$K = \{1\}$
$E = \{2, 3\}$



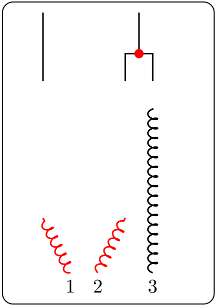
1. If i is flavourless, combine i and j and apply the usual flavour summation.
2. If i is flavoured, identify all pseudojets carrying flavour at this stage (including jets that were declared as *beam jets* earlier). → list K.
   Initialise a set of particles to be excluded from consideration, $E = \{i, j\}$.
3. **Neutralisation step**: $\forall k \in K \setminus E$, in order of decreasing $u_{ik} < u_{ij}$, neutralise as much flavour in i from k as possible.

Flavour neutralisation: the "simple" algorithm

When pseudojets i and j recombine (where $p_{ti} < p_{tj}$):

1. If i is flavourless, combine i and j and apply the usual flavour summation.
2. If i is flavoured, identify all pseudojets carrying flavour at this stage (including jets that were declared as *beam jets* earlier). $\rightarrow$ list K.
   Initialise a set of particles to be excluded from consideration, $E = \{i, j\}$.
3. **Neutralisation step**: $\forall k \in K \setminus E$, in order of decreasing $u_{ik} < u_{ij}$, neutralise as much flavour in i from k as possible.

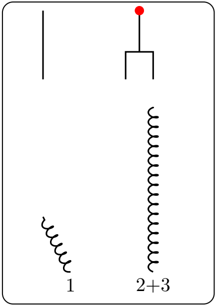
$i = 2, j = 3$
$K = \{1\}$
$E = \{2, 3\}$



$i = 2, j = 3$
$K = \{1\}$
$E = \{2, 3\}$

When pseudojets i and j recombine (where $p_{ti} < p_{tj}$):

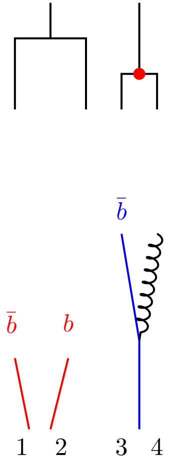
1. If i is flavourless, combine i and j and apply the usual flavour summation.
2. If i is flavoured, identify all pseudojets carrying flavour at this stage (including jets that were declared as *beam jets* earlier). $\rightarrow$ list K.
   Initialise a set of particles to be excluded from consideration, $E = \{i, j\}$.



3. **Neutralisation step**: $\forall k \in K \setminus E$, in order of decreasing $u_{ik} < u_{ij}$, neutralise as much flavour in i from k as possible.
4. Move on to the next kinematic clustering step.

Flavour neutralisation: the "recursive" algorithm

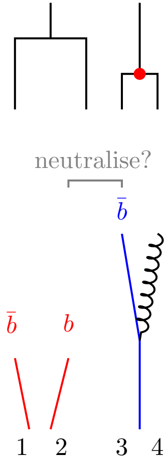
- Recursive version (needed for IRC safety!)



- Recursive version (needed for IRC safety!)

$N(K, E, i, u_{max})$:
$\forall k \in K \setminus E$, in order of decreasing $u_{ik} < u_{\max}$,

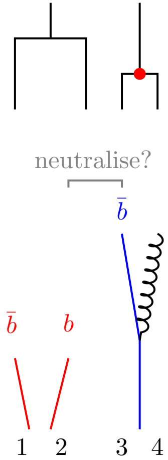
1. If k has no flavour that can neutralise i, continue
2. If it does, call $N(K, E \cup \{k\}, k, u_{ik})$
3. Neutralise as much flavour in i as one can with k.
4. If i is now flavourless, stop.
5. Otherwise continue to the next k, or exit if there aren't any left.



- Recursive version (needed for IRC safety!)

$N(K, E, i, u_{max})$:
$\forall k \in K \setminus E$, in order of decreasing $u_{ik} < u_{\max}$,

1. If k has no flavour that can neutralise i, continue
2. If it does, call $N(K, E \cup \{k\}, k, u_{ik})$
3. Neutralise as much flavour in i as one can with k.
4. If i is now flavourless, stop.
5. Otherwise continue to the next k, or exit if there aren't any left.

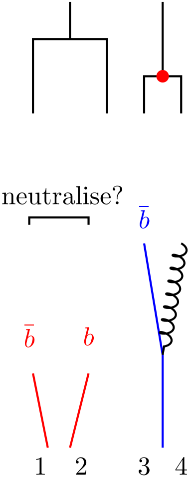


Flavour neutralisation: the “recursive” algorithm

- Recursive version (needed for IRC safety!)

$N(K, E, i, u_{max})$:
$\forall k \in K \setminus E$, in order of decreasing $u_{ik} < u_{\max}$,

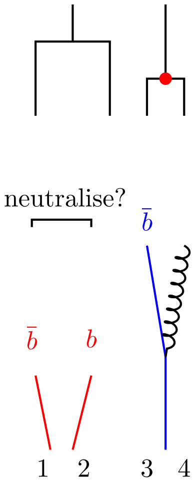
1. If k has no flavour that can neutralise i, continue
2. If it does, call $N(K, E \cup \{k\}, k, u_{ik})$
3. Neutralise as much flavour in i as one can with k.
4. If i is now flavourless, stop.
5. Otherwise continue to the next k, or exit if there aren't any left.



- Recursive version (needed for IRC safety!)

$N(K, E, i, u_{max})$:
$\forall k \in K \setminus E$, in order of decreasing $u_{ik} < u_{\max}$,

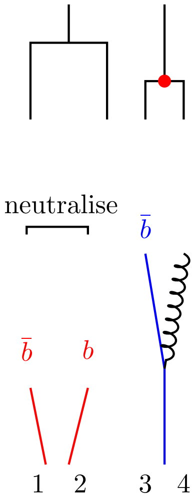
1. If k has no flavour that can neutralise i, continue
2. If it does, call $N(K, E \cup \{k\}, k, u_{ik})$
3. Neutralise as much flavour in i as one can with k.
4. If i is now flavourless, stop.
5. Otherwise continue to the next k, or exit if there aren't any left.



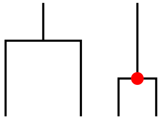
- Recursive version (needed for IRC safety!)

$N(K, E, i, u_{max})$:
$\forall k \in K \setminus E$, in order of decreasing $u_{ik} < u_{\max}$,

1. If k has no flavour that can neutralise i, continue
2. If it does, call $N(K, E \cup \{k\}, k, u_{ik})$
3. Neutralise as much flavour in i as one can with k.
4. If i is now flavourless, stop.
5. Otherwise continue to the next k, or exit if there aren't any left.



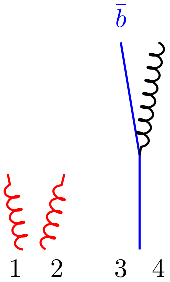
Flavour neutralisation: the “recursive” algorithm

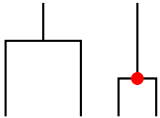


- Recursive version (needed for IRC safety!)

$N(K, E, i, u_{max})$:
$\forall k \in K \setminus E$, in order of decreasing $u_{ik} < u_{\max}$,

1. If k has no flavour that can neutralise i, continue
2. If it does, call $N(K, E \cup \{k\}, k, u_{ik})$
3. Neutralise as much flavour in i as one can with k.
4. If i is now flavourless, stop.
5. Otherwise continue to the next k, or exit if there aren't any left.



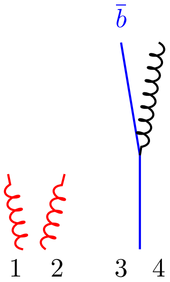


- Recursive version (needed for IRC safety!)

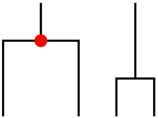
$N(K, E, i, u_{max})$:

$\forall k \in K \setminus E$, in order of decreasing $u_{ik} < u_{\max}$,

1. If k has no flavour that can neutralise i, continue
2. If it does, call $N(K, E \cup \{k\}, k, u_{ik})$
3. Neutralise as much flavour in i as one can with k.
4. If i is now flavourless, stop.
5. Otherwise continue to the next k, or exit if there aren't any left.



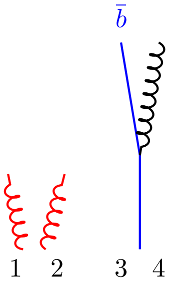
Flavour neutralisation: the “recursive” algorithm



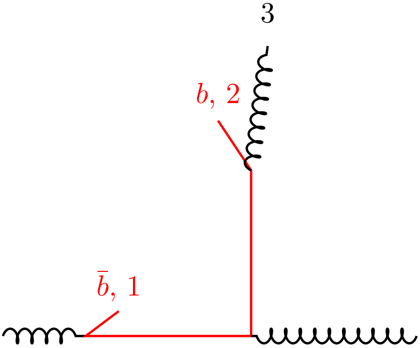
- Recursive version (needed for IRC safety!)

$N(K, E, i, u_{max})$:
$\forall k \in K \setminus E$, in order of decreasing $u_{ik} < u_{\max}$,

1. If k has no flavour that can neutralise i, continue
2. If it does, call $N(K, E \cup \{k\}, k, u_{ik})$
3. Neutralise as much flavour in i as one can with k.
4. If i is now flavourless, stop.
5. Otherwise continue to the next k, or exit if there aren’t any left.



IFN: angular parameter $\omega + \alpha > 2$



$$e^{y_1} \sim p_{t,1} \ll p_{t,2} \lesssim p_{t,3} \sim 1$$

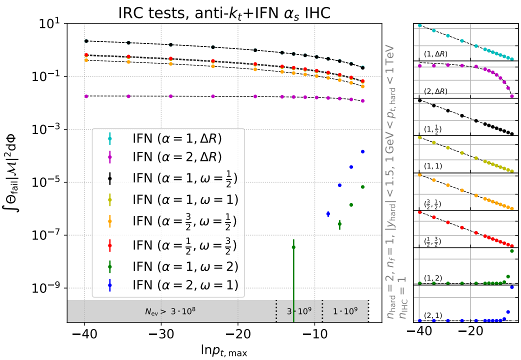
- We need to ensure $u_{12} > u_{23}$

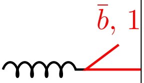
$$p_{t2}^{\alpha} p_{t1}^{2-\alpha} \left(\frac{p_{t3}}{p_{t1}}\right)^{\omega} > p_{t3}^{\alpha} p_{t2}^{2-\alpha} \Delta R_{23}^2$$

$$\leftrightarrow p_{t3}^{\alpha+\omega} p_{t1}^{2-\alpha-\omega} > p_{t3}^{\alpha} p_{t2}^{2-\alpha} \Delta R_{23}^2$$

$$\leftrightarrow \alpha + \omega > 2$$

IFN: angular parameter $\omega + \alpha > 2$





$u_{12} > u_{23}$

$p_{t2}^{2-\alpha} \Delta R_{23}^2$

$p_{t2}^{2-\alpha} \Delta R_{23}^2$

$e^{y_1} \sim p_t,$

Examples of pitfalls / I

- Czakon, Mitov & Poncelet (CMP): modification of anti-k_t distance for flavoured pairs

$$d_{ij} = \min\left(p_{ti}^{-2}, p_{tj}^{-2}\right) \frac{\Delta R_{ij}^2}{R^2} \times \mathcal{S}_{ij} \qquad d_{iB} = p_{ti}^{-2}$$

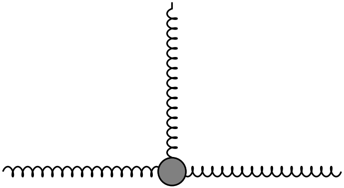
$$\mathcal{S}_{ij} = 1 - \Theta(1-\kappa)\cos\left(\frac{\pi}{2}\kappa\right) \qquad \kappa = \frac{1}{2a}\frac{p_{ti}^2 + p_{tj}^2}{p_{t,\max}^2}$$

Examples of pitfalls / I

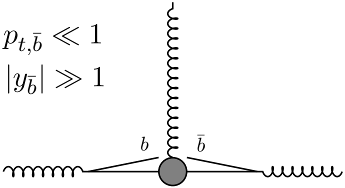
- Czakon, Mitov & Poncelet (CMP): modification of anti-k_t distance for flavoured pairs

$$d_{ij} = \min\left(p_{ti}^{-2}, p_{tj}^{-2}\right) \frac{\Delta R_{ij}^2}{R^2} \times \mathcal{S}_{ij} \qquad d_{iB} = p_{ti}^{-2}$$

$$\mathcal{S}_{ij} = 1 - \Theta(1-\kappa)\cos\left(\frac{\pi}{2}\kappa\right) \qquad \kappa = \frac{1}{2a}\frac{p_{ti}^2 + p_{tj}^2}{p_{t,\max}^2}$$



$$p_{t,b},\, p_{t,\bar{b}} \ll 1$$
$$|y_b|,\, |y_{\bar{b}}| \gg 1$$

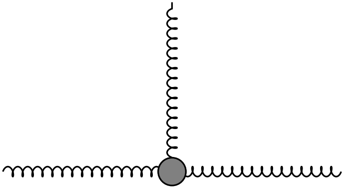


Examples of pitfalls / I

- Czakon, Mitov & Poncelet (CMP): modification of anti-k_t distance for flavoured pairs

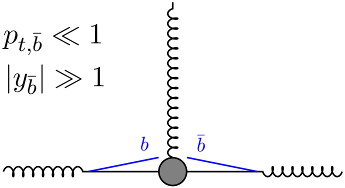
$$d_{ij} = \min\left(p_{ti}^{-2}, p_{tj}^{-2}\right)\frac{\Delta R_{ij}^2}{R^2} \times \mathcal{S}_{ij} \qquad d_{iB} = p_{ti}^{-2}$$

$$\mathcal{S}_{ij} = 1 - \Theta(1-\kappa)\cos\left(\frac{\pi}{2}\kappa\right) \qquad \kappa = \frac{1}{2a}\frac{p_{ti}^2 + p_{tj}^2}{p_{t,\max}^2}$$



$p_{t,b},\, p_{t,\bar{b}} \ll 1$

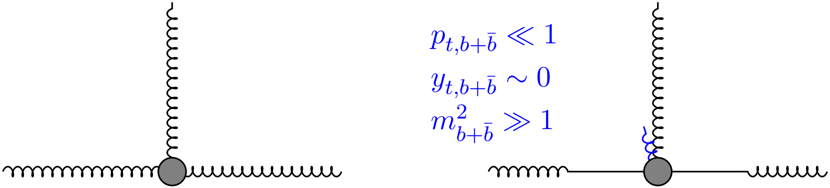
$|y_b|,\, |y_{\bar{b}}| \gg 1$



- Czakon, Mitov & Poncelet (CMP): modification of anti-k_t distance for flavoured pairs

$$d_{ij} = \min\left(p_{ti}^{-2}, p_{tj}^{-2}\right) \frac{\Delta R_{ij}^2}{R^2} \times \mathcal{S}_{ij} \qquad d_{iB} = p_{ti}^{-2}$$

$$\mathcal{S}_{ij} = 1 - \Theta(1-\kappa)\cos\left(\frac{\pi}{2}\kappa\right) \qquad \kappa = \frac{1}{2a}\frac{p_{ti}^2 + p_{tj}^2}{p_{t,\max}^2}$$

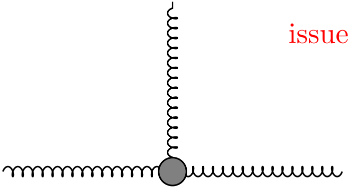


Examples of pitfalls / I

- Czakon, Mitov & Poncelet (CMP): modification of anti-k_t distance for flavoured pairs

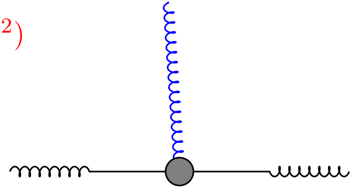
$$d_{ij} = \min\left(p_{ti}^{-2}, p_{tj}^{-2}\right) \frac{\Delta R_{ij}^2}{R^2} \times \mathcal{S}_{ij} \qquad d_{iB} = p_{ti}^{-2}$$

$$\mathcal{S}_{ij} = 1 - \Theta(1-\kappa)\cos\left(\frac{\pi}{2}\kappa\right) \qquad \kappa = \frac{1}{2a}\frac{p_{ti}^2 + p_{tj}^2}{p_{t,\max}^2}$$



issue at $\mathcal{O}(\alpha_s^2 L^2)$

$\neq$

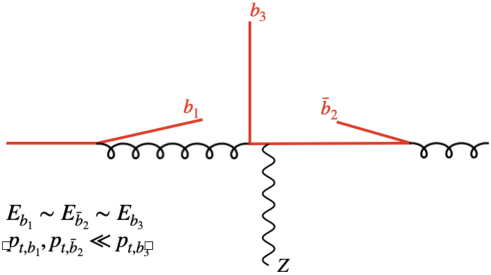


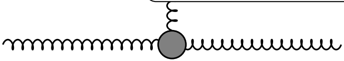
- Czakon ... ti-k_t distanc ...

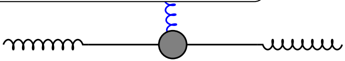
d_{ij} ...

$\mathcal{S}_{ij}$ ...

Could also be $Z + b$ at NNLO:



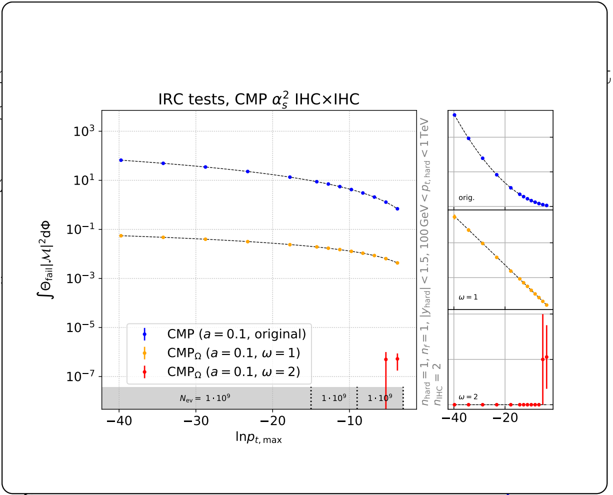


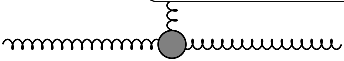


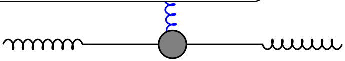
- Czakon … ti-k_t distanc…

d_{ij}

$\mathcal{S}_{ij}$







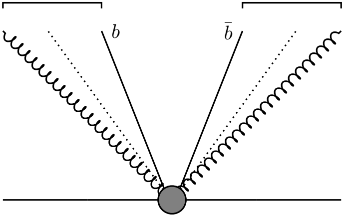
Examples of pitfalls / II

- Gauld, Huss, Stagnitto (GHS): set of jet-cluster and cluster-cluster distances

$$d_{\hat{f}_i,\hat{f}_j}, \quad d_{\hat{f}_i,j_k}, \quad d_{\hat{f}_i,B_\pm}$$

- Gauld, Huss, Stagnitto (GHS): set of jet-cluster and cluster-cluster distances

$$d_{\hat{f}_i,\hat{f}_j}, \quad d_{\hat{f}_i,j_k}, \quad d_{\hat{f}_i,B_\pm}$$



Hard event: 2 flavoured jets

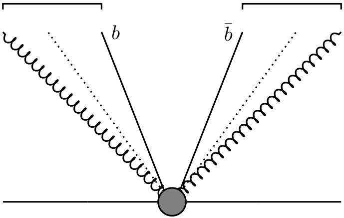
Examples of pitfalls / II

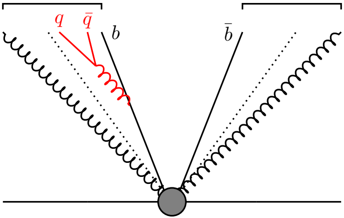
- Gauld, Huss, Stagnitto (GHS): set of jet-cluster and cluster-cluster distances

$$d_{\hat{f}_i,\hat{f}_j}, \quad d_{\hat{f}_i,j_k}, \quad d_{\hat{f}_i,B_\pm}$$

$d_{\bar{b},j_{\bar{b}}} < d_{b,\bar{b}}$

$p_{t,g} \sim z p_{t,b},\ z \to 1$



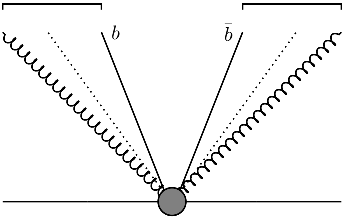


Examples of pitfalls / II

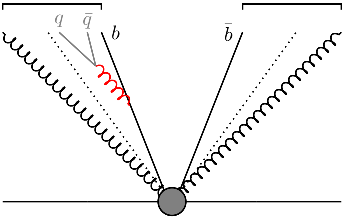
- Gauld, Huss, Stagnitto (GHS): set of jet-cluster and cluster-cluster distances

$$d_{\hat{f}_i,\hat{f}_j}, \quad d_{\hat{f}_i,j_k}, \quad d_{\hat{f}_i,B_\pm}$$

$d_{\bar{b},j_{\bar{b}}} < d_{b,\bar{b}}$



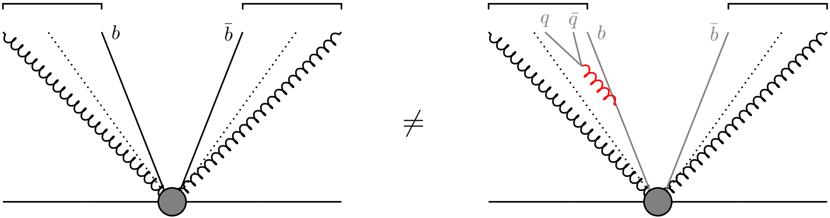
$d_{b\bar{b}} \downarrow$



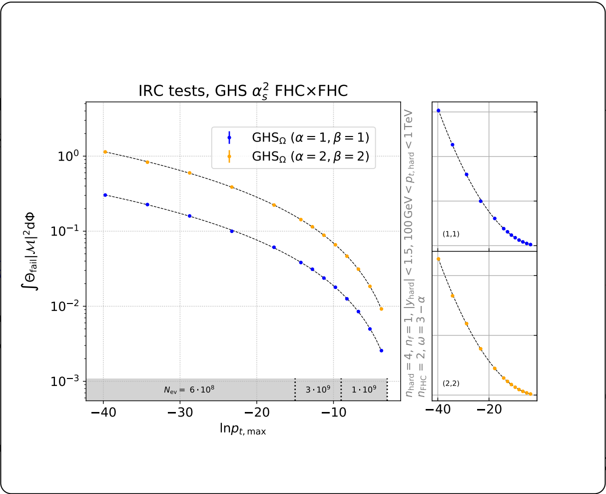
- Gauld, Huss, Stagnitto (GHS): set of jet-cluster and cluster-cluster distances

$$d_{\hat{f}_i,\hat{f}_j}, \quad d_{\hat{f}_i,j_k}, \quad d_{\hat{f}_i,B_\pm}$$

$d_{\bar{b},j_{\bar{b}}} < d_{b,\bar{b}}$

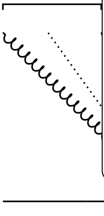


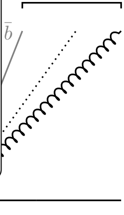
- Ga… d
  clus…



$d_{\bar b, j_{\bar b}} < d_{b,\bar b}$ …

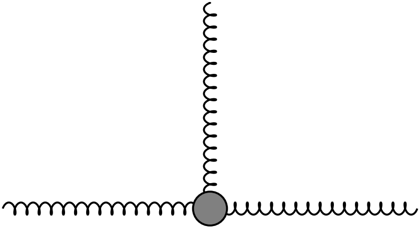
…avourless jets



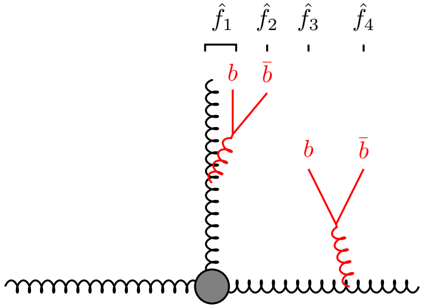


(Could also be $t\bar t$ boosted, or $t\bar t + 1$ jet)

Examples of pitfalls / III



Hard event:
→ 1 flavourless jet



Hard event:
→ 1 flavourless jet

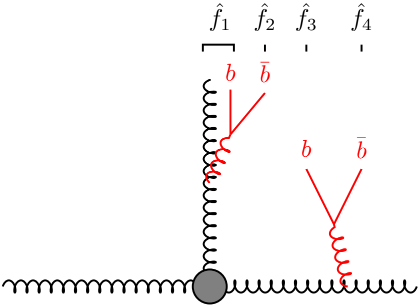
hard+IRC event:

1(b) accumulated into hard g, but not 2($\bar{b}$)

$\hat{f}_2$ and $\hat{f}_3$ annihilate, but $\hat{f}_1$ and $\hat{f}_4$ do not

→ 1 b-jet (+ 1 $\bar{b}$ beam jet)

Examples of pitfalls / III



Hard event:
→ 1 flavourless jet

hard+IRC event:

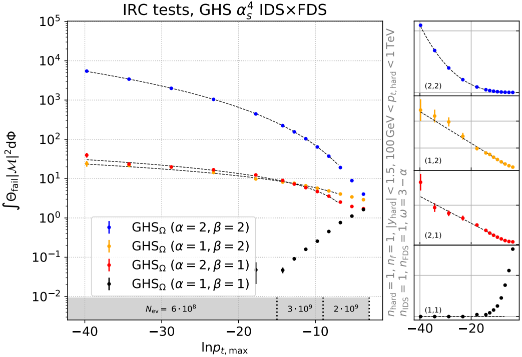
1(b) accumulated into hard g, but not 2($\bar{b}$)

$\hat{f}_2$ and $\hat{f}_3$ annihilate, but $\hat{f}_1$ and $\hat{f}_4$ do not

→ 1 b-jet (+ 1 $\bar{b}$ beam jet)

- Some analytic/numerical understanding of the complicated interplay between distances (as a function of α and β)

  → suggests $\alpha \cdot \beta < 2$ is fine for this configuration



into hard g,

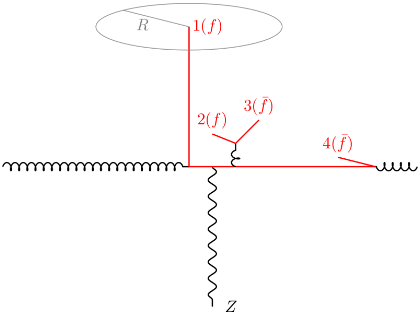
te,

not

am jet)

- Some … cated
  interpl …)

$\rightarrow$ suggests $\alpha \cdot \beta < 2$ is fine for this configuration



Hard event:

$\rightarrow$ 1 flavoured (f) jet

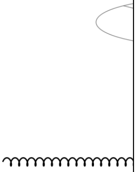
Hard+(1 IDS pair) event:

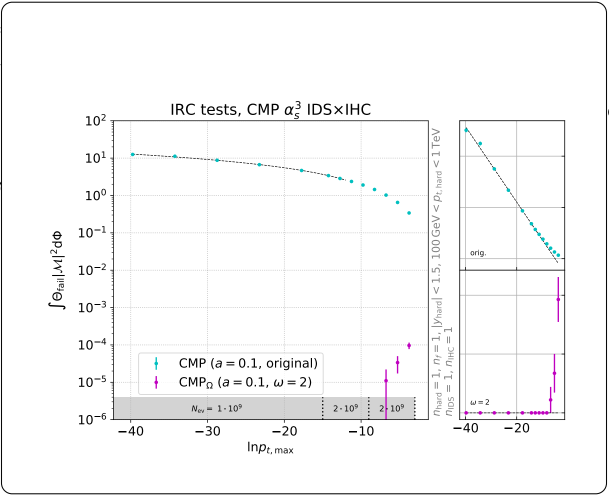
$$p_{t,2} = z p_{t,g}$$
$$E_4 \sim E_1,\ y_4 \sim \log \frac{E_4}{p_{t,4}}$$

$$d_{23} \sim \frac{p_{t,2}^2}{z^2},\ d_{24} \sim p_{t,2}^2 y_4^2$$

CMP failure rate:
$$\mathcal{N} \sim \alpha_s^3 \int_0^{p_{t,1}} \frac{dp_{t,2}}{p_{t,2}} \int_0^{p_{t,1}} \frac{dp_{t,4}}{p_{t,4}} \int_0^1 dz \Theta(d_{24} < d_{23})$$
$$\sim \alpha_s^3 \int_0^L d\ell_{14} \int_0^{l_{14}} d\ell_{24} \int_0^{1/\ell_{14}} dz$$
$$\sim \alpha_s^3 L$$

Examples of pitfalls / IV





et

event:

$\frac{E_4}{p_{t,4}}$