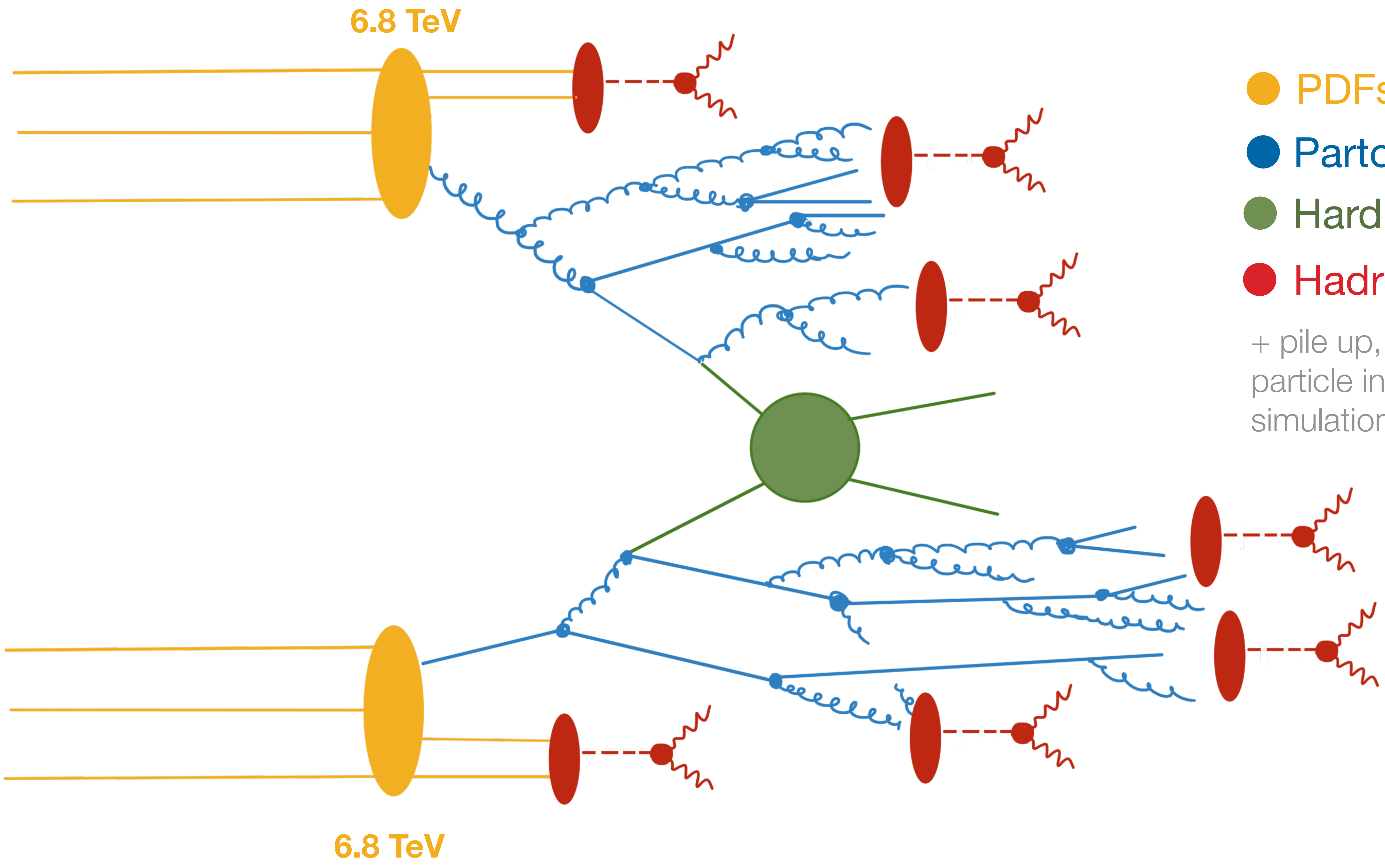
Nikhef topical lectures

Basics of parton showers

Melissa van Beekveld

Nikhef

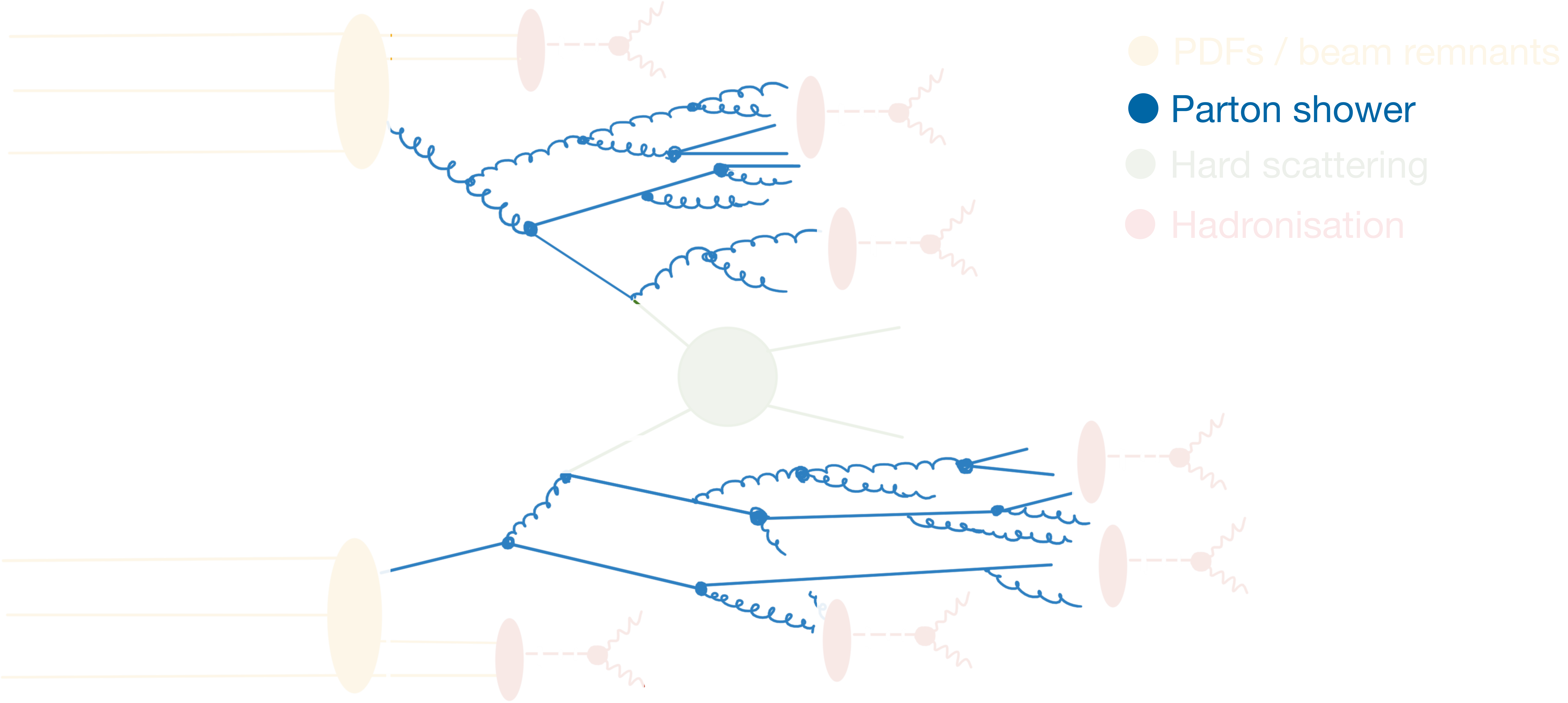
LHC experiments rely on event generators



- PDFs / beam remnants
- Parton shower $\mathcal{O}(1-100)$ GeV
- Hard scattering $\mathcal{O}(0.1-1)$ TeV
- Hadronisation $\mathcal{O}(1)$ GeV

+ pile up, underlying event, multiple-particle interactions (MPI), detector simulation…

LHC experiments rely on event generators



Today's focus

Parton showers: a crucial ingredient

Sherpa

Event generation with SHERPA 1.1 #1

T. Gleisberg (SLAC), Stefan. Hoeche (Zurich U.), F. Krauss (Durham U., IPPP), M. Schonherr (Dresden, Tech. U.), S. Schumann (Edinburgh U.) et al. (Nov, 2008)

Published in: *JHEP* 02 (2009) 007 • e-Print: 0811.4622 [hep-ph]

pdf links DOI cite 3,890 citations

Event Generation with Sherpa 2.2 1,186 citations

Event generation with Sherpa 3 10 citations

Herwig 7

Herwig++ Physics and Manual #1

M. Bahr (Karlsruhe U., ITP), S. Gieseke (Karlsruhe U., ITP), M.A. Gigg (Durham U., IPPP), D. Grellscheid (Durham U., IPPP), K. Hamilton (Louvain U.) et al. (Mar, 2008)

Published in: *Eur.Phys.J.C* 58 (2008) 639-707 • e-Print: 0803.0883 [hep-ph]

pdf links DOI cite 3,229 citations

Herwig 7.0/Herwig++ 3.0 release note 1,621 citations

Pythia 8

An introduction to PYTHIA 8.2 #1

Torbjörn Sjöstrand (Lund U., Dept. Theor. Phys.), Stefan Ask (Cambridge U.), Jesper R. Christiansen (Lund U., Dept. Theor. Phys.), Richard Corke (Lund U., Dept. Theor. Phys.), Nishita Desai (U. Heidelberg, ITP) et al. (Oct 11, 2014)

Published in: *Comput.Phys.Commun.* 191 (2015) 159-177 • e-Print: 1410.3012 [hep-ph]

pdf links DOI cite 6,797 citations

PYTHIA 6.4 Physics and Manual 13,268 citations

A comprehensive guide to the physics and usage of PYTHIA 8.3 796 citations

➡Their strength lies in their **versatility**:

- they will (usually) not give state-of-the-art theory predictions for every single observable
- but will allow you to e.g.
  - implement arbitrary experimental cuts
  - attach hadronisation corrections

Parton showers: a crucial ingredient

Sherpa

Event generation with SHERPA 1.1 #1

T. Gleisberg (SLAC), Stefan. Hoeche (Zurich U.), F. Krauss (Durham U., IPPP), M. Schonherr (Dresden, Tech. U.), S. Schumann (Edinburgh U.) et al. (Nov, 2008)

Published in: *JHEP* 02 (2009) 007 • e-Print: 0811.4622 [hep-ph]

pdf links DOI cite 3,890 citations

Event Generation with Sherpa 2.2 1,186 citations

Event generation with Sherpa 3 10 citations

Herwig 7

Herwig++ Physics and Manual #1

M. Bahr (Karlsruhe U., ITP), S. Gieseke (Karlsruhe U., ITP), M.A. Gigg (Durham U., IPPP), D. Grellscheid (Durham U., IPPP), K. Hamilton (Louvain U.) et al. (Mar, 2008)

Published in: *Eur.Phys.J.C* 58 (2008) 639-707 • e-Print: 0803.0883 [hep-ph]

pdf links DOI cite 3,229 citations

Herwig 7.0/Herwig++ 3.0 release note 1,621 citations

Pythia 8

An introduction to PYTHIA 8.2 #1

Torbjörn Sjöstrand (Lund U., Dept. Theor. Phys.), Stefan Ask (Cambridge U.), Jesper R. Christiansen (Lund U., Dept. Theor. Phys.), Richard Corke (Lund U., Dept. Theor. Phys.), Nishita Desai (U. Heidelberg, ITP) et al. (Oct 11, 2014)

Published in: *Comput.Phys.Commun.* 191 (2015) 159-177 • e-Print: 1410.3012 [hep-ph]

pdf links DOI cite 6,797 citations

PYTHIA 6.4 Physics and Manual 13,268 citations

A comprehensive guide to the physics and usage of PYTHIA 8.3 796 citations

➡Their strength lies in their **versatility**

Improving their accuracy is in general more difficult than for any defined process/observable (and topic of active research!)

Parton showers: a crucial ingredient

Sherpa

Event generation with SHERPA 1.1 #1

T. Gleisberg (SLAC), Stefan. Hoeche (Zurich U.), F. Krauss (Durham U., IPPP), M. Schonherr (Dresden, Tech. U.), S. Schumann (Edinburgh U.) et al. (Nov, 2008)

Published in: *JHEP* 02 (2009) 007 • e-Print: 0811.4622 [hep-ph]

pdf links DOI cite 3,890 citations

Event Generation with Sherpa 2.2 1,186 citations

Event generation with Sherpa 3 10 citations

Herwig 7

Herwig++ Physics and Manual #1

M. Bahr (Karlsruhe U., ITP), S. Gieseke (Karlsruhe U., ITP), M.A. Gigg (Durham U., IPPP), D. Grellscheid (Durham U., IPPP), K. Hamilton (Louvain U.) et al. (Mar, 2008)

Published in: *Eur.Phys.J.C* 58 (2008) 639-707 • e-Print: 0803.0883 [hep-ph]

pdf links DOI cite 3,229 citations

Herwig 7.0/Herwig++ 3.0 release note 1,621 citations

Pythia 8

An introduction to PYTHIA 8.2 #1

Torbjörn Sjöstrand (Lund U., Dept. Theor. Phys.), Stefan Ask (Cambridge U.), Jesper R. Christiansen (Lund U., Dept. Theor. Phys.), Richard Corke (Lund U., Dept. Theor. Phys.), Nishita Desai (U. Heidelberg, ITP) et al. (Oct 11, 2014)

Published in: *Comput.Phys.Commun.* 191 (2015) 159-177 • e-Print: 1410.3012 [hep-ph]

pdf links DOI cite 6,797 citations

PYTHIA 6.4 Physics and Manual 13,268 citations

A comprehensive guide to the physics and usage of PYTHIA 8.3 796 citations

The field of shower development is rich, covering all would need multiple sets of topical lectures!

you are welcome to ask about specific topics ;)

Goal of today:

- Explain the basic ingredients of a parton shower
- Show the (sometimes arbitrary) choices that **have to** be made along the way that lead to finite and important differences in their predictions

Basics of a parton shower

QCD

- Described by the $SU(N_c = 3)$ group
- Quarks are in the fundamental representation (N_c colours)
- Anti-quarks in the anti-fundamental representation
- Gluons in the adjoint ($N_c^2 - 1$ colours)

Basics of a parton shower

QCD

- Described by the $SU(N_c = 3)$ group
- Quarks are in the fundamental representation (N_c colours)
- Anti-quarks in the anti-fundamental representation
- Gluons in the adjoint ($N_c^2 - 1$ colours)

Special type of shower: the dipole shower

- Take the $N_c \to \infty$ limit
- (Anti-)quarks carry (anti-)colour
- Gluons carry one colour and one anti-colour charge
- Assign a colour connection between all colour charges

Basics of a parton shower

QCD

- Described by the $SU(N_c = 3)$ group
- Quarks are in the fundamental representation (N_c colours)
- Anti-quarks in the anti-fundamental representation
- Gluons in the adjoint ($N_c^2 - 1$ colours)

Special type of shower: the dipole shower

- Take the $N_c \to \infty$ limit
- (Anti-)quarks carry (anti-)colour
- Gluons carry one colour and one anti-colour charge
- Assign a colour connection between all colour charges

This is a simple way to build in ‘colour coherence’ (another is angular ordering)

What are dipoles?

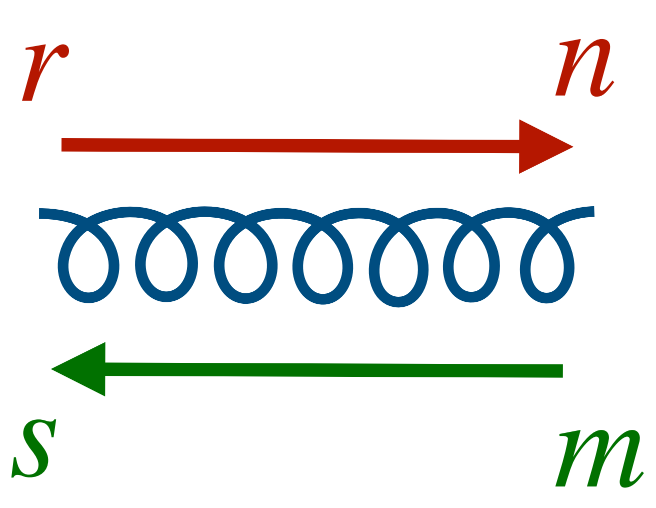
Basically, one connects every colour charge in the process

How is this justified?

We write the colour conservation of the gluon propagator as

$$\delta^{ab} = 2\, t^a_{nm} t^b_{mn} = 2\, t^a_{nm} \delta_{rn} \times \delta_{ms} t^b_{sr}$$

colour flow



What are dipoles?

Basically, one connects every colour charge in the process

How is this justified?

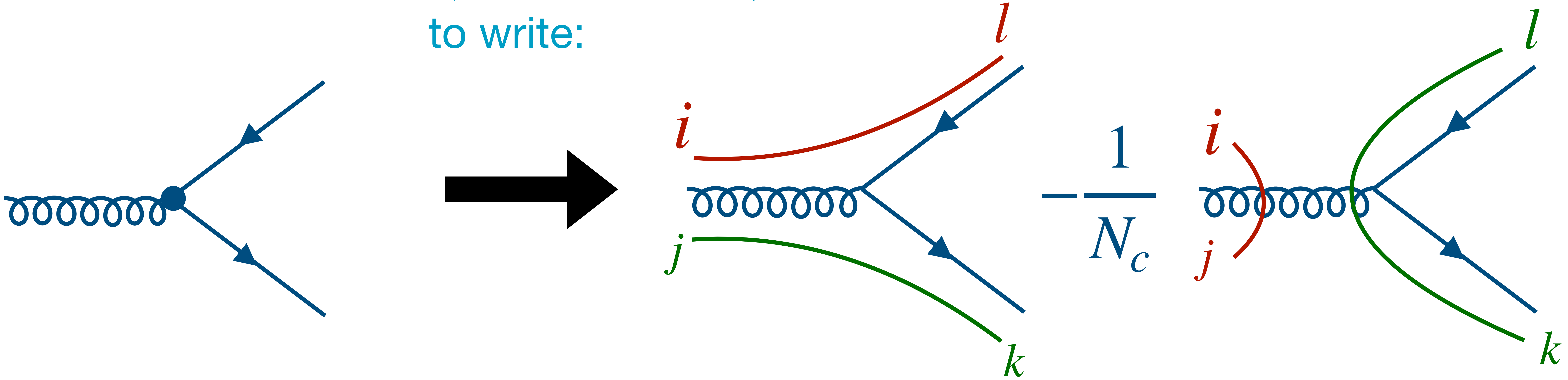
We write the colour conservation of the gluon propagator as

$$\delta^{ab} = 2\, t^a_{nm} t^b_{mn} = 2\, t^a_{nm} \delta_{rn} \times \delta_{ms} t^b_{sr}$$

and use colour algebra, such as

$$t^a_{ij} t^a_{kl} = \frac{1}{2}\left(\delta_{il}\delta_{kj} - \frac{1}{N_c}\delta_{ij}\delta_{kl}\right)$$

to write:



What are dipoles?

Basically, one connects every colour charge in the process

How is this justified?

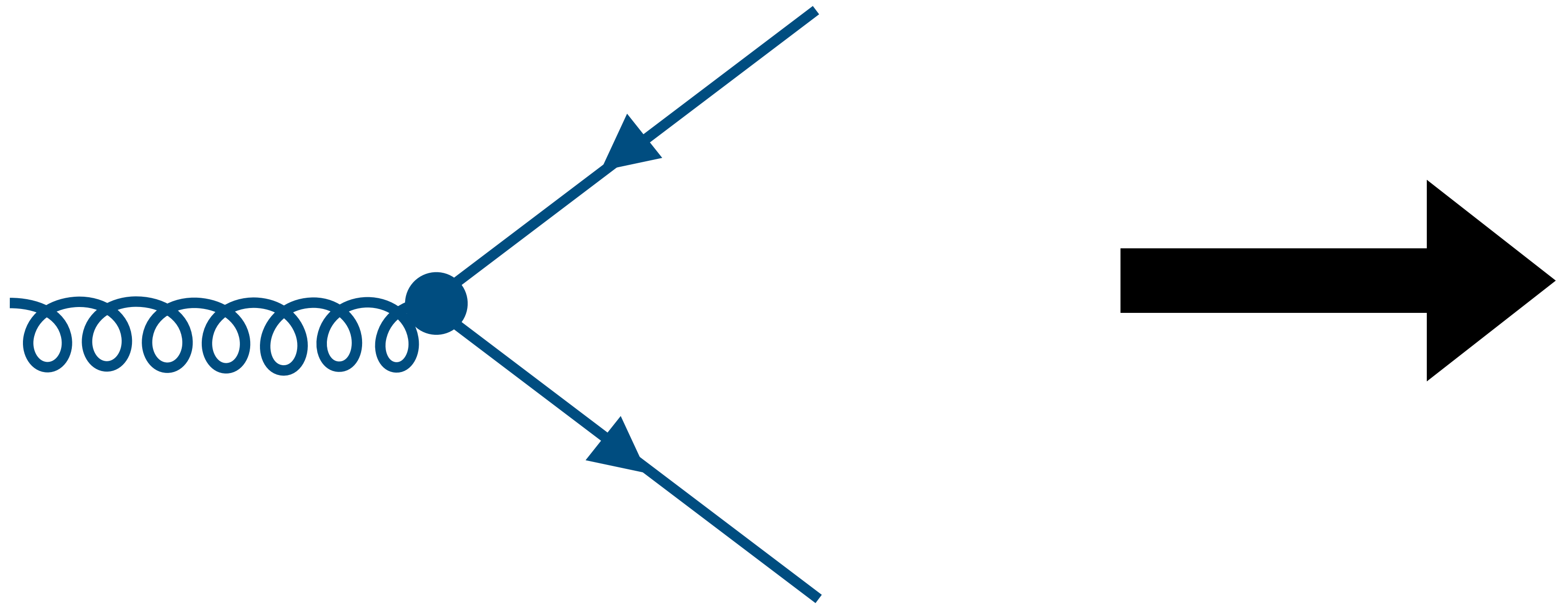
We write the colour conservation of the gluon propagator as

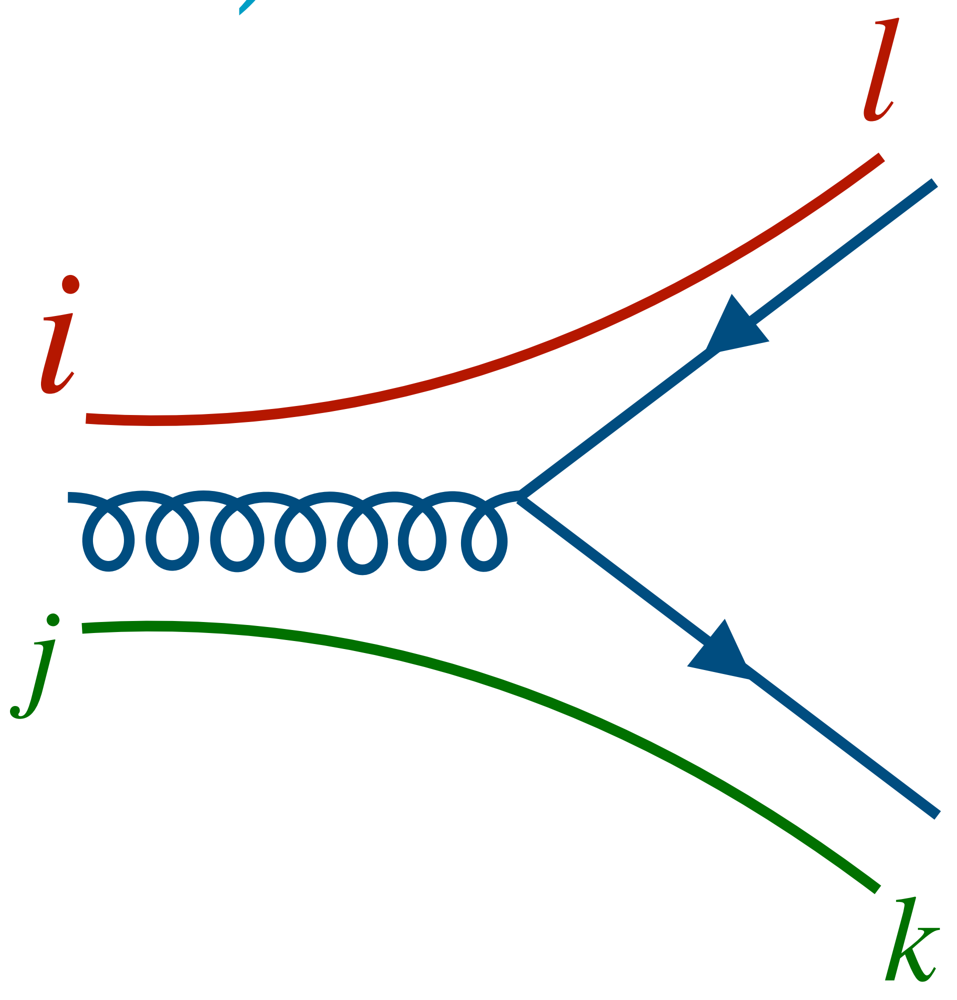
$$\delta^{ab} = 2\, t^a_{nm} t^b_{mn} = 2\, t^a_{nm} \delta_{rn} \times \delta_{ms} t^b_{sr}$$

and use colour algebra, such as

$$t^a_{ij} t^a_{kl} = \frac{1}{2}\left(\delta_{il}\delta_{kj} - \frac{1}{N_c}\delta_{ij}\delta_{kl}\right)$$

to write:





$+\mathcal{O}(1/N_c)$

leading-colour approximation

What are dipoles?

Basically, one connects every colour charge in the process

How is this justified?

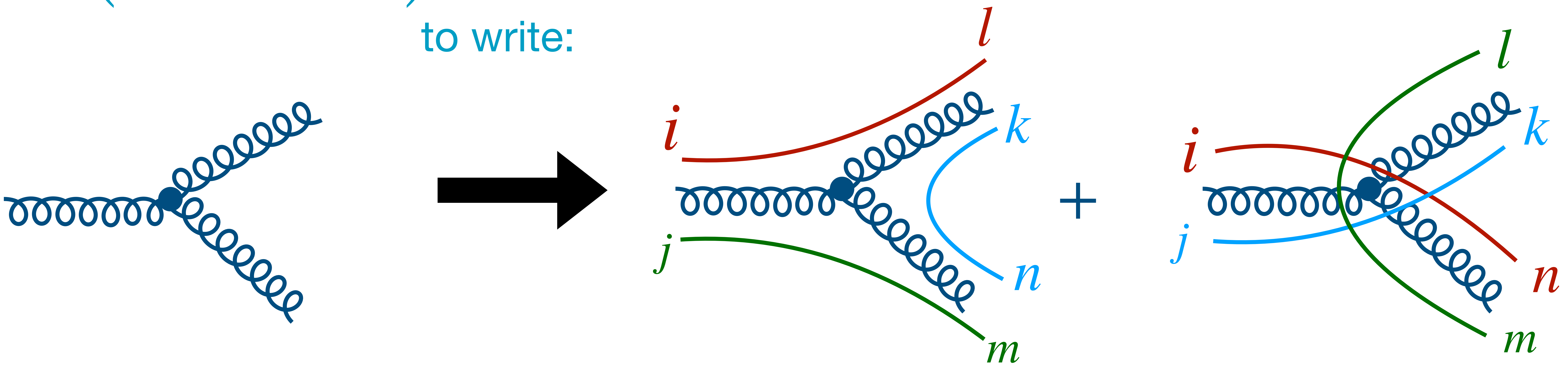
We write the colour conservation of the gluon propagator as

$$\delta^{ab} = 2\, t^a_{nm} t^b_{mn} = 2\, t^a_{nm} \delta_{rn} \times \delta_{ms} t^b_{sr}$$

and use colour algebra, such as

$$t^a_{ij} t^a_{kl} = \frac{1}{2}\left(\delta_{il}\delta_{kj} - \frac{1}{N_c}\delta_{ij}\delta_{kl}\right) \quad \text{and} \quad f^{abc} t^a_{ij} t^b_{kl} t^c_{mn} = \delta_{il}\delta_{kn}\delta_{mj} - \delta_{in}\delta_{ml}\delta_{kj}$$

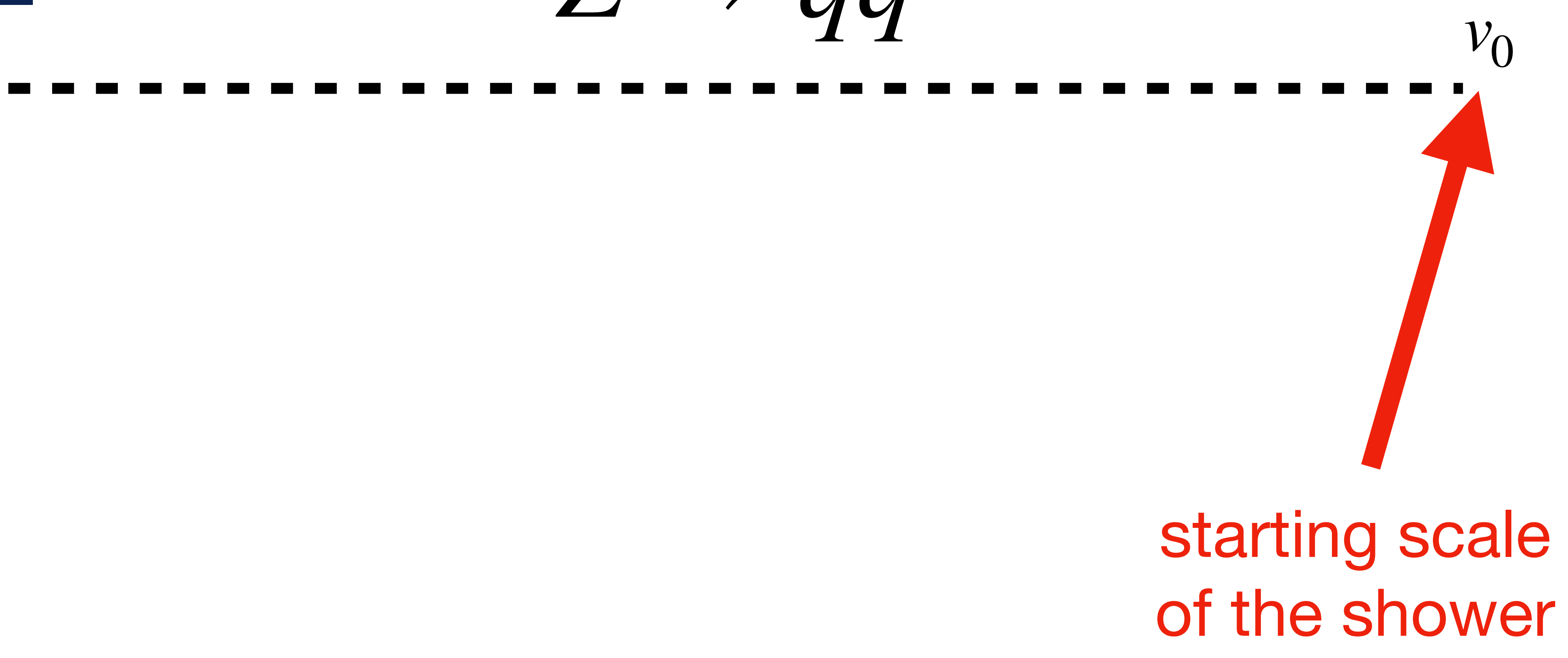
to write:



Evolution of a parton shower

Illustrated with a dipole shower for final-state emissions

$Z \to q\bar{q}$



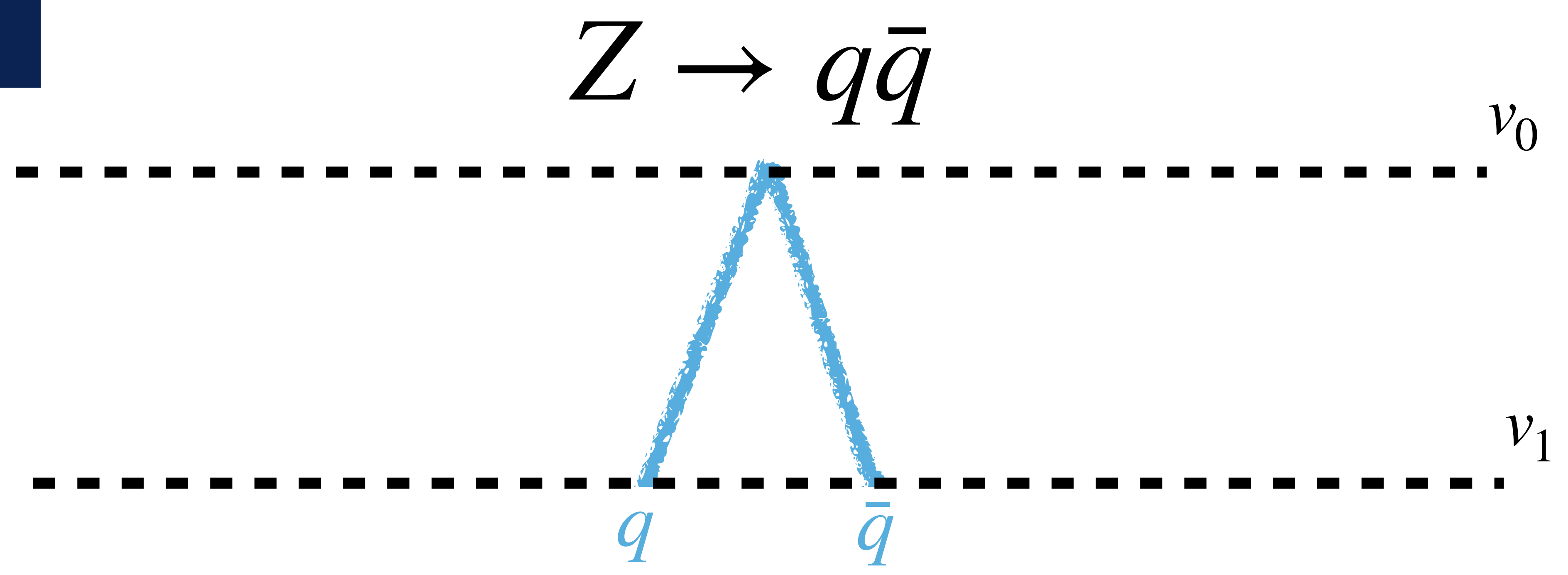
Start with some partonic state
This spans an initial ‘colour dipole’

Evolution of a parton shower

Illustrated with a dipole shower for final-state emissions

Start with some partonic state
This spans an initial 'colour dipole'

Throw a random number to determine the scale v_1 until which 'nothing happens'



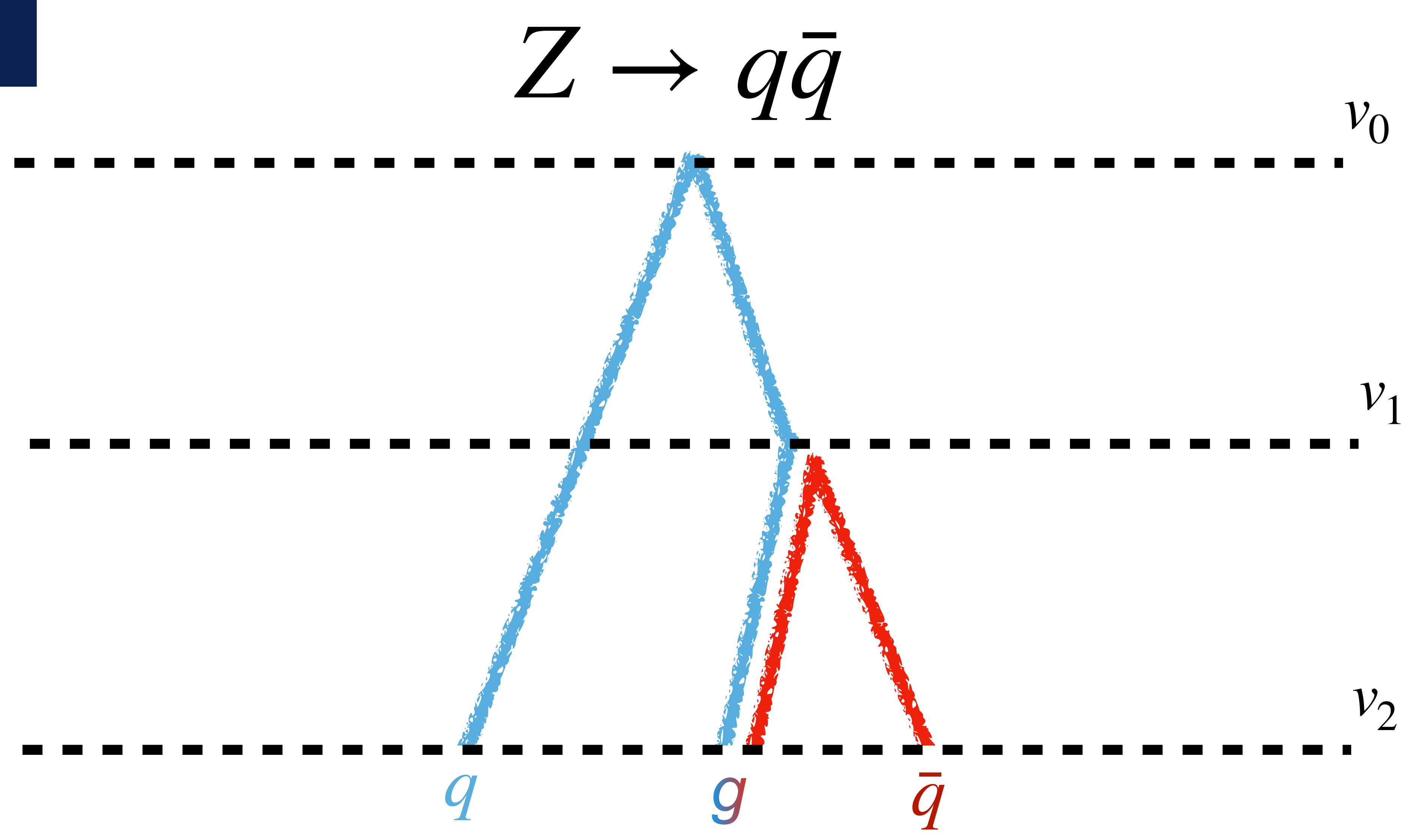
Evolution of a parton shower

Illustrated with a dipole shower for final-state emissions

Start with some partonic state
This spans an initial 'colour dipole'

Throw a random number to determine the scale v_1 until which 'nothing happens'

The state splits…
The new gluon is part of two (independent) dipoles



Evolution of a parton shower

Illustrated with a dipole shower for final-state emissions

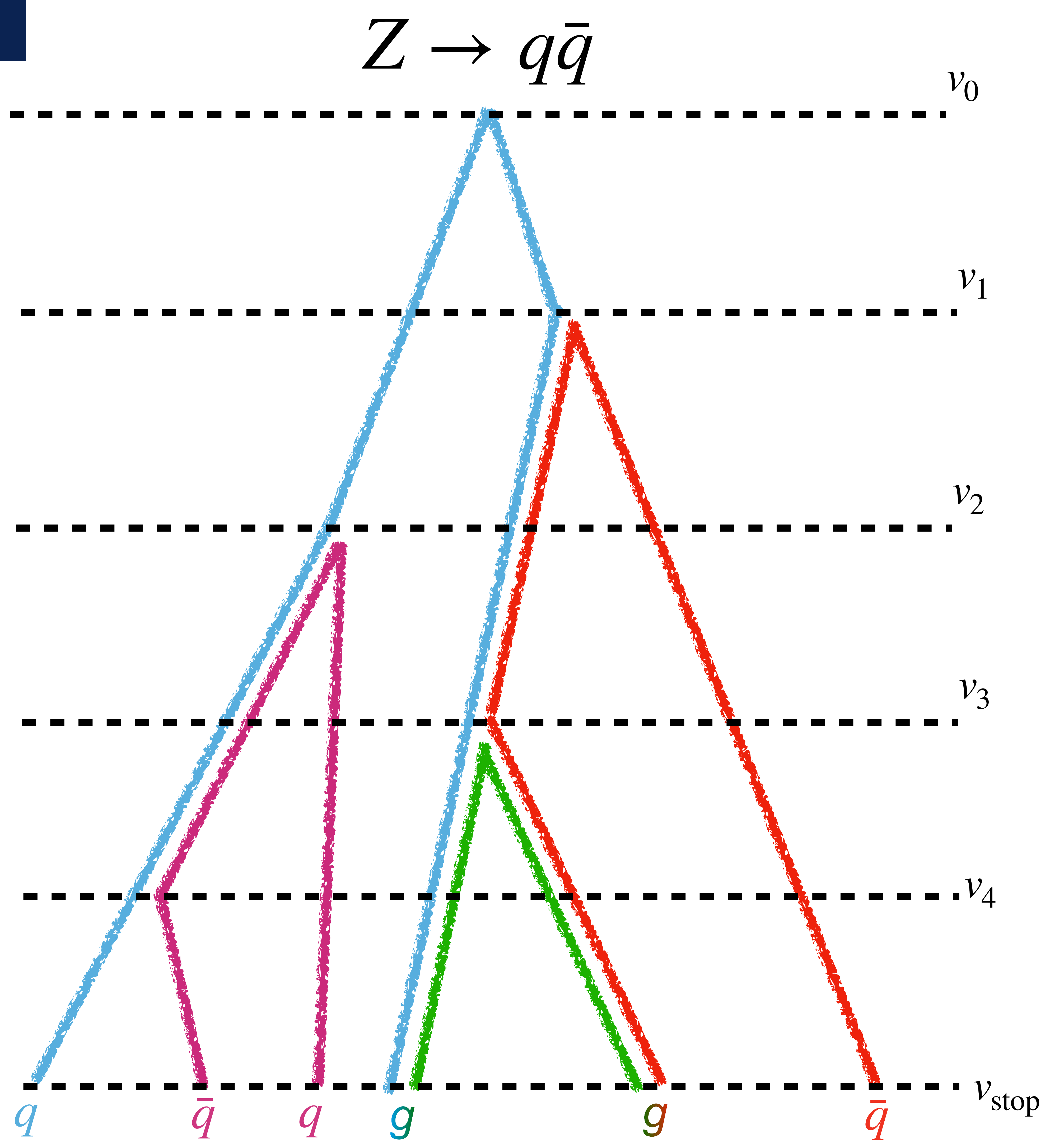
Start with some partonic state
This spans an initial ‘colour dipole’

Throw a random number to determine the scale v_1 until which ‘nothing happens’

The state splits…
The new gluon is part of two (independent) dipoles

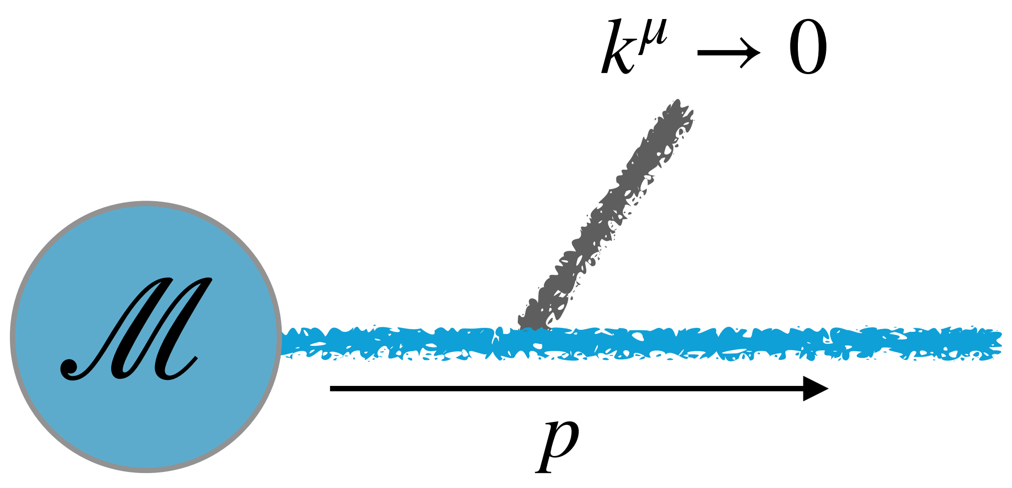
Process continues until it reaches a non-perturbative cut-off scale

End result: set of particles and their four momenta, from which any (well-defined) observable may be reconstructed



The splitting probability

Emission of a soft gluon: the eikonal Feynman rule

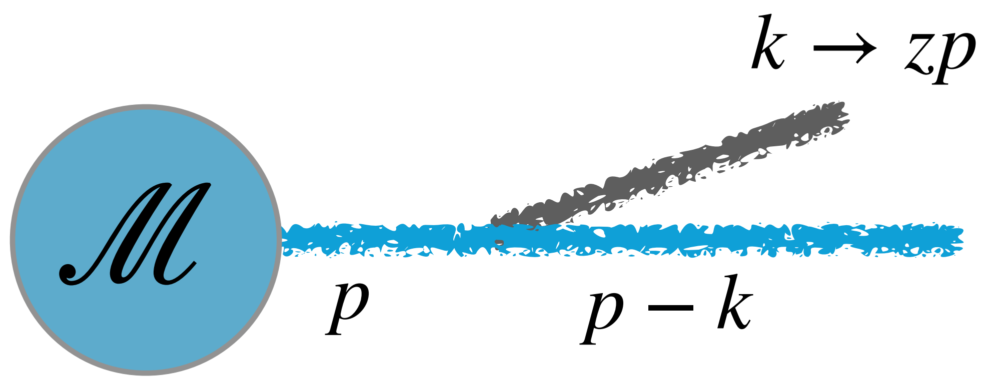


$$\propto g_s \frac{p^\mu}{p \cdot k} \mathbf{T} \otimes \mathscr{M}$$

$\mathbf{T}$ is a colour-generator

- Spin dependence is factorised
- Colour dependence is not

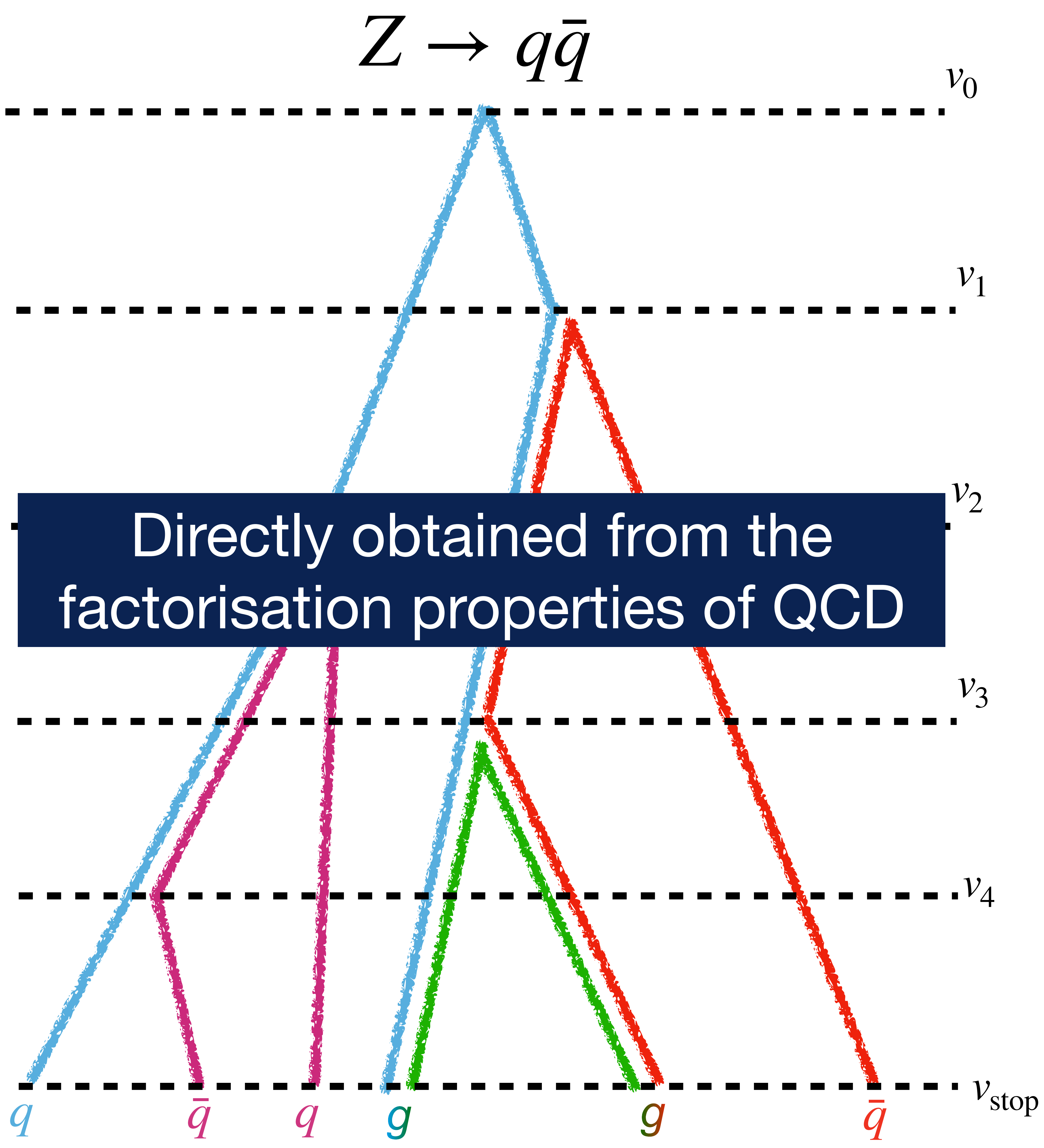
Emission of a collinear particle: Splitting functions $P_{(ij)a}$



$$\propto g_s \frac{1}{p \cdot k} P_{(ij),a}(z) \otimes \mathscr{M}_a$$

a is a spin index

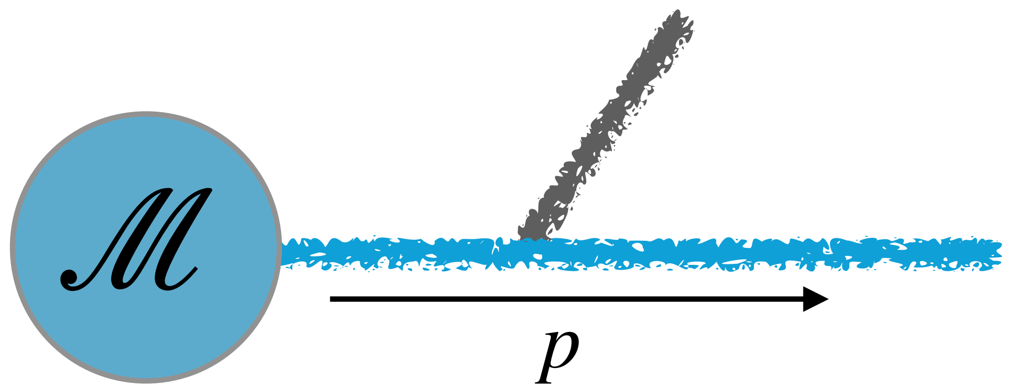
- Colour dependence is factorised
- Spin dependence is not



Note that the phase-space factorises as well

The splitting probability

Emission of a soft gluon: the eikonal Feynman rule

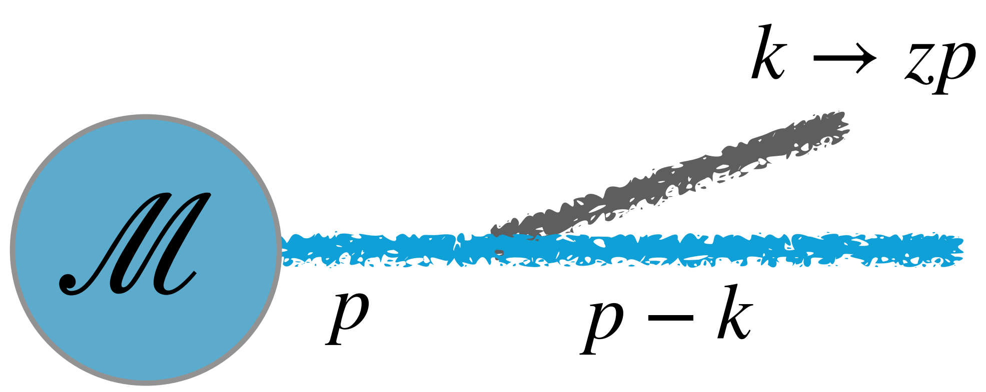


$$\propto g_s \frac{p^\mu}{p \cdot k} \mathbf{T} \otimes \mathscr{M}$$

$\mathbf{T}$ is a colour-generator

- Spin dependence is factorised
- Colour dependence is not

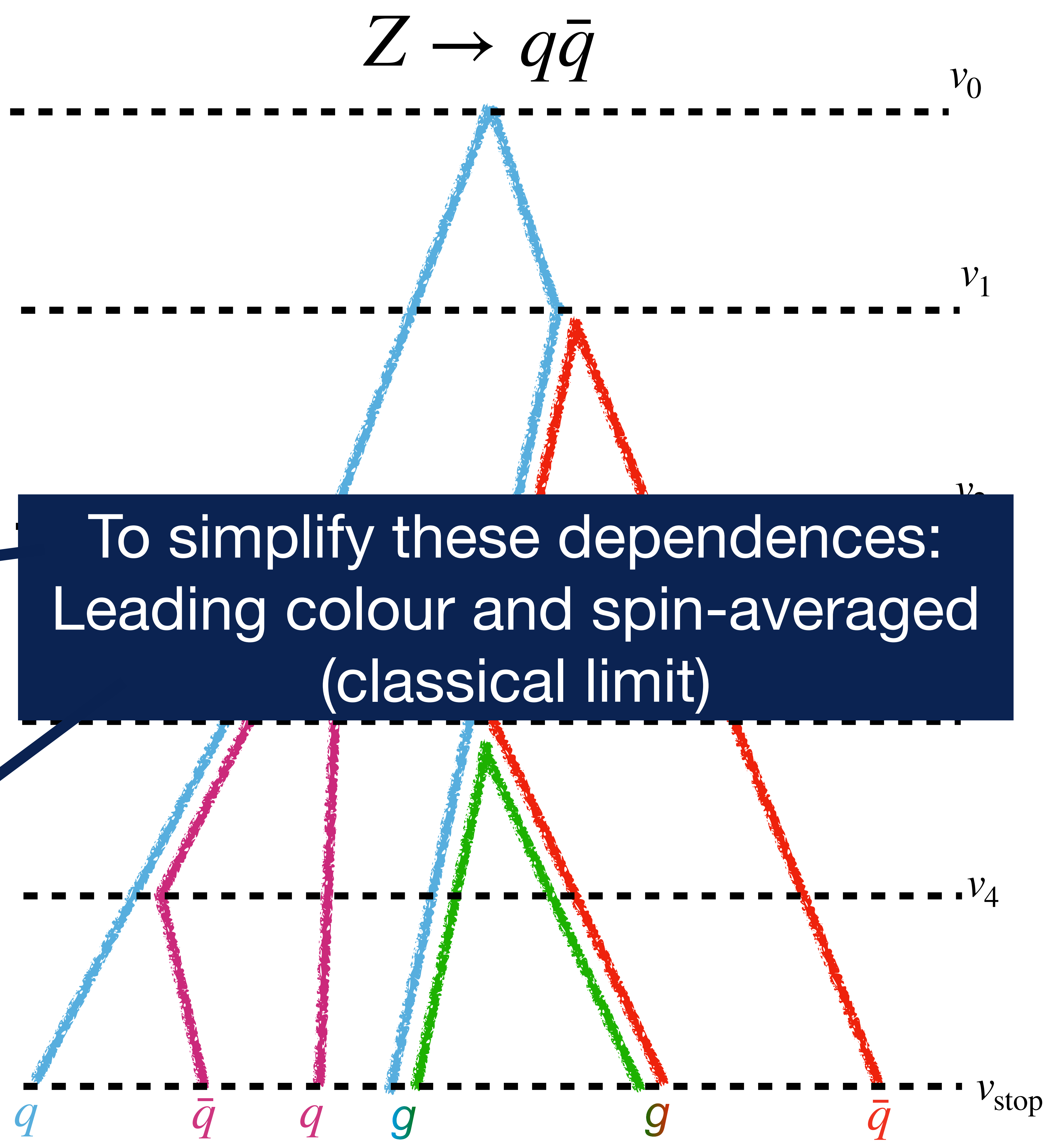
Emission of a collinear particle: Splitting functions $P_{(ij)a}$



$$\propto g_s \frac{1}{p \cdot k} P_{(ij),a}(z) \otimes \mathscr{M}_c$$

a is a spin index

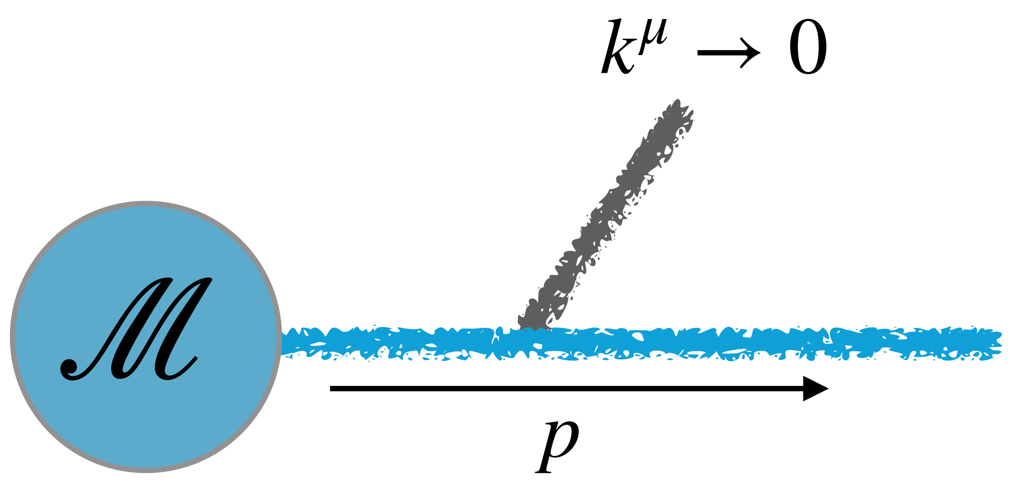
- Colour dependence is factorised
- Spin dependence is not



To simplify these dependences: Leading colour and spin-averaged (classical limit)

The splitting probability

Emission of a soft gluon: the eikonal Feynman rule

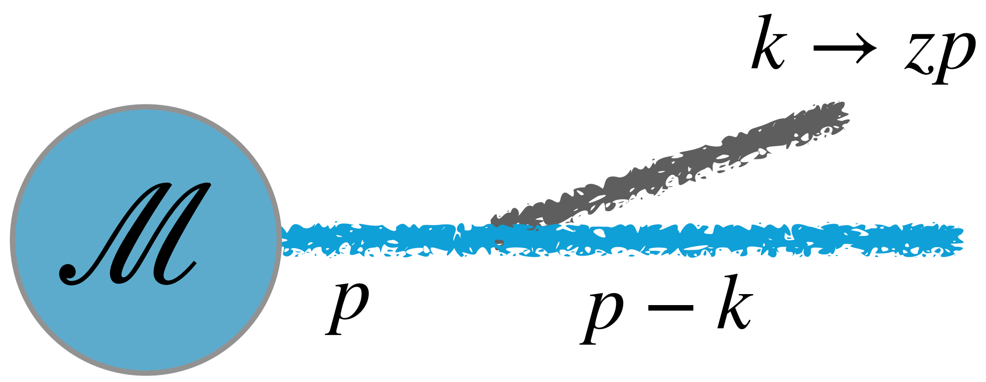


$$\propto g_s \frac{p^\mu}{p \cdot k} \mathbf{T} \otimes \mathscr{M}$$

$\mathbf{T}$ is a colour-generator

- Spin dependence is factorised
- Colour dependence is not

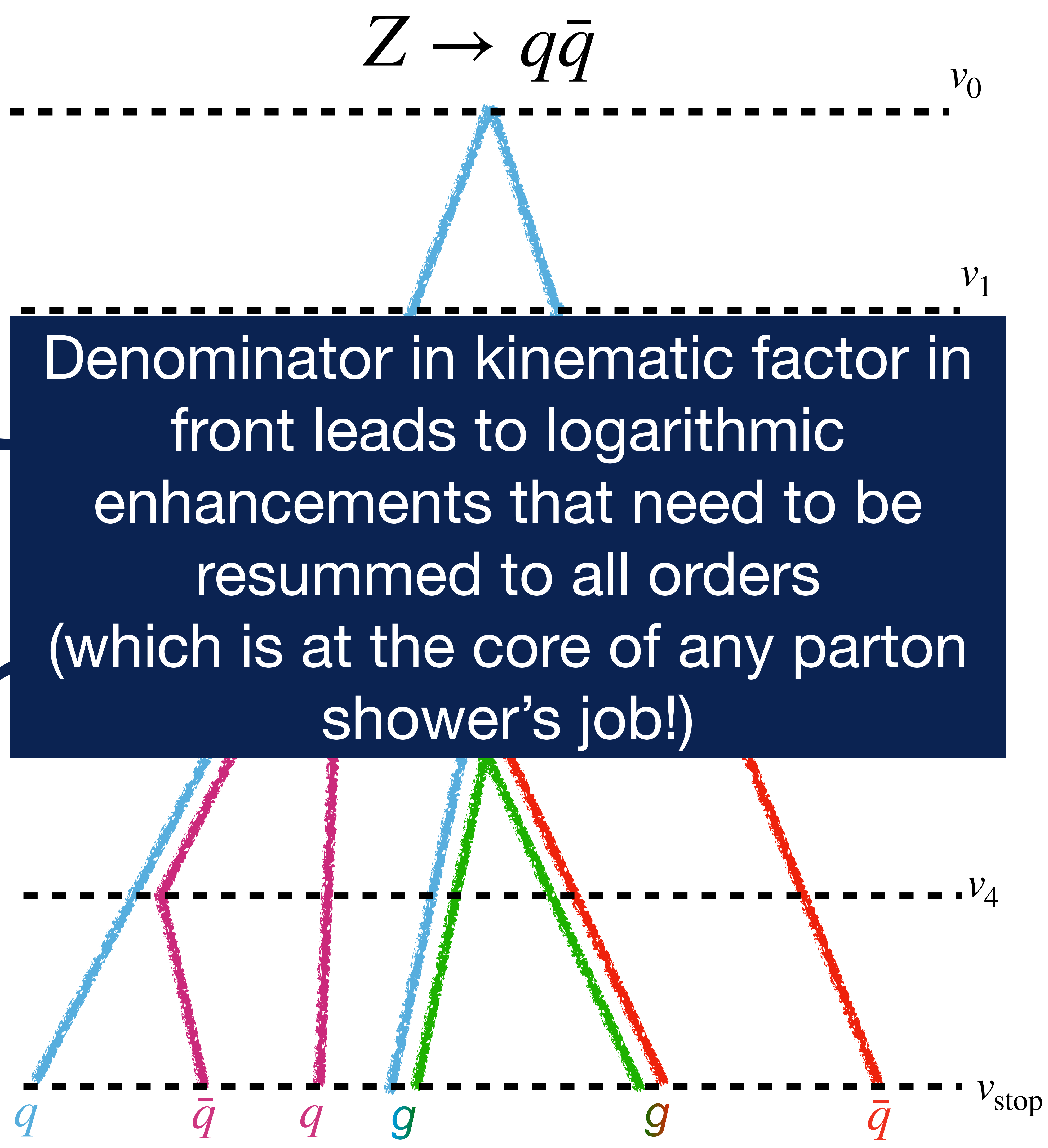
Emission of a collinear particle: Splitting functions $P_{(ij)a}$



$$\propto g_s \frac{1}{p \cdot k} P_{(ij),a}(z) \otimes \mathscr{M}_a$$

a is a spin index

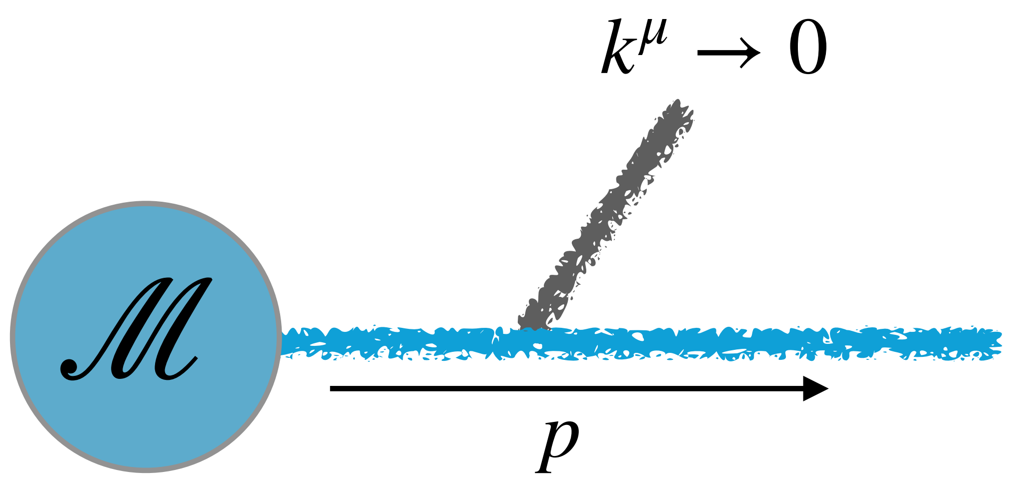
- Colour dependence is factorised
- Spin dependence is not



Denominator in kinematic factor in front leads to logarithmic enhancements that need to be resummed to all orders (which is at the core of any parton shower's job!)

The splitting probability

Emission of a soft gluon: the eikonal Feynman rule

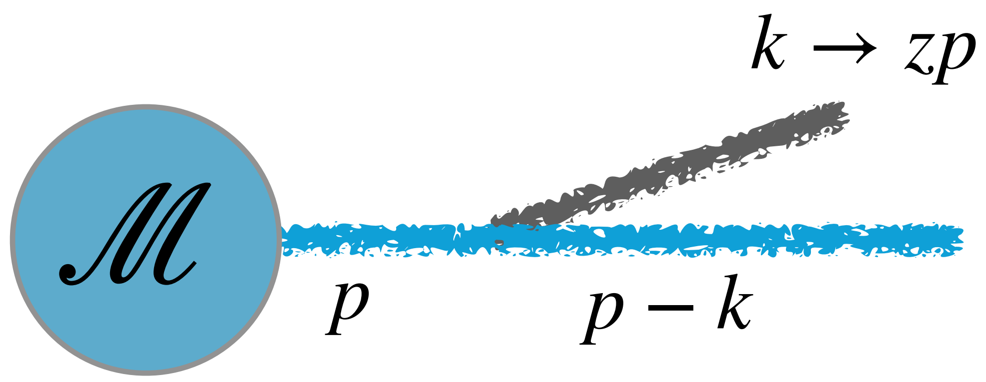


$$\propto g_s \frac{p^\mu}{p \cdot k} \mathbf{T} \otimes \mathcal{M}$$

$\mathbf{T}$ is a colour-generator

- Spin dependence is factorised
- Colour dependence is not

Emission of a collinear particle: Splitting functions $P_{(ij)a}$



$$\propto g_s \frac{1}{p \cdot k} P_{(ij),a}(z) \otimes \mathcal{M}_a$$

a is a spin index

- Colour dependence is factorised
- Spin dependence is not

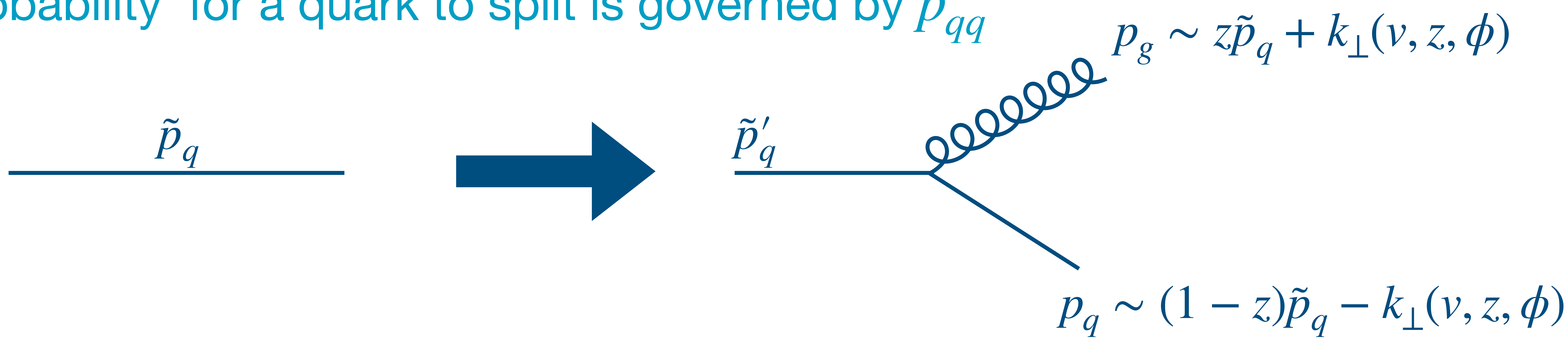
Together with:

1. A physical meaning for v
2. A 'kinematic map'

this defines a shower algorithm

Probability to generate an emission

- The 'probability' for a quark to split is governed by p_{qq}



Probability to generate an emission

- The 'probability' for a quark to split is governed by p_{qq}

Total probability: $p_{1\to 2} = \int_{z_-}^{z_+} dz \frac{\alpha_s}{2\pi} p_{qq}(z)$

- The probability to split in a small evolution interval is $p_{1\to 2}\delta v$
- The probability not to split in a small evolution interval is $p_{\text{nosplit}} = 1 - p_{1\to 2}\delta v$

Probability to generate an emission

- The ‘probability' for a quark to split is governed by p_{qq}

Total probability: $p_{1\to 2} = \int_{z_-}^{z_+} dz \frac{\alpha_s}{2\pi} p_{qq}(z)$

- The probability to split in a small evolution interval is $p_{1\to 2}\delta v$
- The probability not to split in a small evolution interval is $p_{\text{nosplit}} = 1 - p_{1\to 2}\delta v$

→ integrating over a larger interval Δv gives

$$p_{\text{nosplit}}(v_0, v) = \lim_{n\to\infty} p_{\text{nosplit}}(v_0, v_0 + \delta v) \cdot p_{\text{nosplit}}(v_0 + \delta v, v_0 + 2\delta v) \cdot \ldots \cdot p_{\text{nosplit}}(v_0 + n\delta v, v)$$

Probability to generate an emission

- The 'probability' for a quark to split is governed by p_{qq}

  Total probability: $p_{1\to 2} = \int_{z_-}^{z_+} dz \frac{\alpha_s}{2\pi} p_{qq}(z)$

- The probability to split in a small evolution interval is $p_{1\to 2}\delta v$
- The probability not to split in a small evolution interval is $p_{\text{nosplit}} = 1 - p_{1\to 2}\delta v$

  $\to$ integrating over a larger interval Δv gives

$$p_{\text{nosplit}}(v_0, v) = \lim_{n\to\infty} p_{\text{nosplit}}(v_0, v_0 + \delta v) \cdot p_{\text{nosplit}}(v_0 + \delta v, v_0 + 2\delta v) \cdot \ldots \cdot p_{\text{nosplit}}(v_0 + n\delta v, v)$$

$$= \lim_{n\to\infty} \left(1 + \frac{1}{n} \int_{v_0}^{v} dv' p_{1\to 2}\right)^n$$

Probability to generate an emission

- The 'probability' for a quark to split is governed by p_{qq}

  Total probability: $p_{1\to 2} = \int_{z_-}^{z_+} dz \frac{\alpha_s}{2\pi} p_{qq}(z)$

- The probability to split in a small evolution interval is $p_{1\to 2}\delta v$
- The probability not to split in a small evolution interval is $p_{\text{nosplit}} = 1 - p_{1\to 2}\delta v$

  $\rightarrow$ integrating over a larger interval Δv gives

$$p_{\text{nosplit}}(v_0, v) = \lim_{n\to\infty} p_{\text{nosplit}}(v_0, v_0 + \delta v) \cdot p_{\text{nosplit}}(v_0 + \delta v, v_0 + 2\delta v) \cdot \ldots \cdot p_{\text{nosplit}}(v_0 + n\delta v, v)$$

$$= \lim_{n\to\infty} \left(1 + \frac{1}{n}\int_{v_0}^{v} dv' p_{1\to 2}\right)^n = \exp\left[-\int_{v_0}^{v} dv' p_{1\to 2}\right] = \Delta(v_0, v)$$

this is called the Sudakov, or no-branching probability

Probability to generate an emission

- The probability of branching at a scale v is given by the probability of branching times the probability that the quark did not branch before

$$p_{\text{branch}}(v) = p_{1\to 2}(v)\Delta(v_0, v)$$

Probability to generate an emission

- The probability of branching at a scale v is given by the probability of branching times the probability that the quark did not branch before

$$p_{\mathrm{branch}}(v) = p_{1\to 2}(v)\Delta(v_0, v)$$

- v is an evolution variable that orders the sequence of emissions

Probability to generate an emission

- The probability of branching at a scale v is given by the probability of branching times the probability that the quark did not branch before

$$p_{\mathrm{branch}}(v) = p_{1\to 2}(v)\Delta(v_0, v)$$

- v is an evolution variable that orders the sequence of emissions

Often taken to be related to the transverse momentum of the emission in the collinear limit (Pythia, Sherpa)…

Probability to generate an emission

- The probability of branching at a scale v is given by the probability of branching times the probability that the quark did not branch before

$$p_{\rm branch}(v) = p_{1\to 2}(v)\Delta(v_0, v)$$

- v is an evolution variable that orders the sequence of emissions

Often taken to be related to the transverse momentum of the emission in the collinear limit (Pythia, Sherpa)…

…but also angular ordering is a popular choice (Herwig)

Probability to generate an emission

- The probability of branching at a scale v is given by the probability of branching times the probability that the quark did not branch before

$$p_{\mathrm{branch}}(v) = p_{1\to2}(v)\Delta(v_0, v)$$

- v is an evolution variable that orders the sequence of emissions

Often taken to be related to the transverse momentum of the emission in the collinear limit (Pythia, Sherpa)…

…but also angular ordering is a popular choice (Herwig)

This is the 'master equation' of the shower algorithm…

… but how to solve it?

The Sudakov veto algorithm

- Define $p_{1\to 2}(v) \equiv f(v)$ and $\Delta(0,v) \equiv \exp\left[-\int_0^v dv' f(v')\right]$
- Assume we know $F(v) = \int_0^v dv' f(v')$ and its inverse

The Sudakov veto algorithm

- Define $p_{1\to 2}(v) \equiv f(v)$ and $\Delta(0,v) \equiv \exp\left[-\int_0^v dv' f(v')\right]$
- Assume we know $F(v) = \int_0^v dv' f(v')$ and its inverse

To then select v values according to

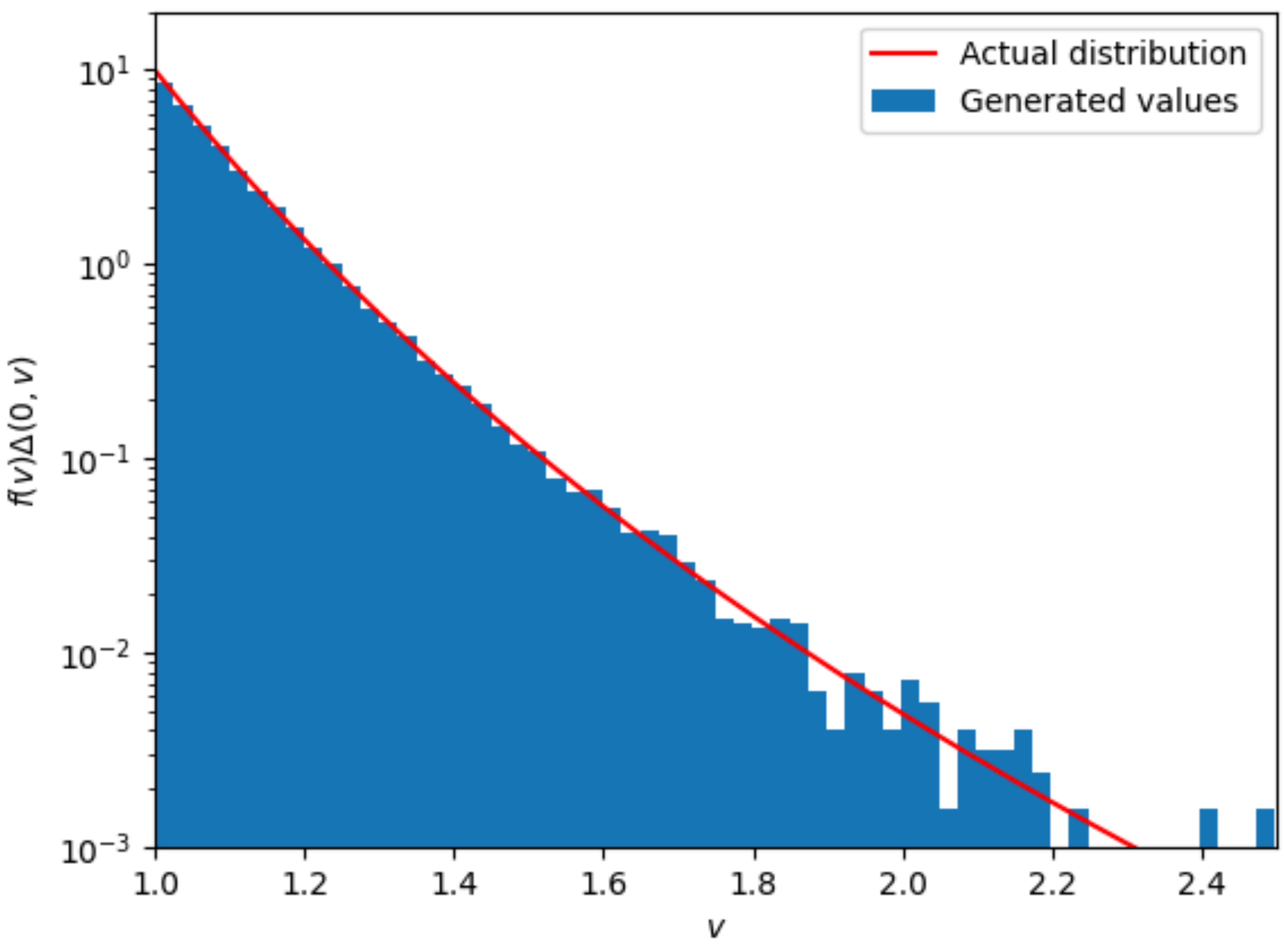
$$p_{\text{branch}}(v) = p_{1\to 2}(v)\Delta(0,v)$$

we can use 'standard' Monte-Carlo techniques:

$$\int_0^v dv' f(v')\Delta(v') = \Delta(0) - \Delta(0,v) = 1 - \exp\left[-F(v)\right] \equiv 1 - r$$

Solve for v → $v = F^{-1}\left(F(0) - \ln r\right)$

```python
# Constants
N = 50000
v0 = 1
c = 10.

# define the integrand
def f(v):
    return c/v

# define the integrated function
def integrated_f(v):
    return c*np.log(v)

# define the inverse of the integrated function
def inverse_integrated_f(v):
    return np.exp(v/c)

# generate values for v
def generate_v(v0, n_samples):
    v = np.zeros(n_samples)
    for i in range(n_samples):
        r = random.uniform(0,1)
        v[i] = inverse_integrated_f(integrated_f(v0) - np.log(r))
    return v

# generate the values
v = generate_v(v0, N)
# histogram of the generated values
plt.hist(v, bins = 100, density=True)

# compare to the target distribution
v = np.linspace(1,20,1000)
plt.plot(v, f(v)*np.exp(-integrated_f(v0)-integrated_f(v)), 'r')

# label the plot
plt.xlabel(r'$v$')
plt.ylabel(r'$f(v)\Delta(0,v)$')
plt.legend(['Generated values','Actual distribution'])
# set xrange
plt.xlim([1,2.5])
plt.ylim([1E-3,20])
# set y scale logarithmic
plt.yscale('log')
plt.show()
```



The Sudakov veto algorithm

- Define $p_{1\to2}(v) \equiv f(v)$ and $\Delta(0,v) \equiv \exp\left[-\int_0^v dv' f(v')\right]$
- Assume we know $F(v) = \int_0^v dv' f(v')$ and its inverse ➡ This is very often not possible!

To then select v values according to

$$p_{\text{branch}}(v) = p_{1\to2}(v)\Delta(0,v)$$

we can use 'standard' Monte-Carlo techniques:

$$\int_0^v dv' f(v')\Delta(v') = \Delta(0) - \Delta(0,v) = 1 - \exp\left[-F(v)\right] \equiv 1 - r$$

Solve for $v \to \quad v = F^{-1}\left(F(0) - \ln r\right)$

The Sudakov veto algorithm

The Sudakov veto algorithm:

1. Start with $i = 0$ and $v_0 = 0$

The Sudakov veto algorithm

The Sudakov veto algorithm:

1. Start with $i = 0$ and $v_0 = 0$
2. $i += 1$; select $v_i = G^{-1}(G(v_i) - \ln(r_1))$ with $g(v) \geq f(v)$

The Sudakov veto algorithm

The Sudakov veto algorithm:

1. Start with $i = 0$ and $v_0 = 0$
2. $i += 1$; select $v_i = G^{-1}(G(v_i) - \ln(r_1))$ with $g(v) \geq f(v)$
3. Compare r_2 to $f(v_i)/g(v_i)$

   if $f(v_i)/g(v_i) \leq r_2$: return to 2

   else: accept v_i

The Sudakov veto algorithm

The Sudakov veto algorithm:

1. Start with $i = 0$ and $v_0 = 0$
2. $i += 1$; select $v_i = G^{-1}(G(v_i) - \ln(r_1))$ with $g(v) \geq f(v)$
3. Compare r_2 to $f(v_i)/g(v_i)$

   if $f(v_i)/g(v_i) \leq r_2$: return to 2

   else: accept v_i
4. Continue until v_i reaches v_{cutoff}

The Sudakov veto algorithm

The Sudakov veto algorithm:

1. Start with $i = 0$ and $v_0 = 0$

2. $i += 1$; select $v_i = G^{-1}(G(v_i) - \ln(r_1))$ with $g(v) \geq f(v)$

3. Compare r_2 to $f(v_i)/g(v_i)$

   if $f(v_i)/g(v_i) \leq r_2$: return to 2

   else: accept v_i

4. Continue until v_i reaches v_{cutoff}

One can analytically prove that this gives the right answer by considering all the different ways of selecting a new v_i

The Sudakov veto algorithm

The Sudakov veto algorithm:

1. Start with $i = 0$ and $v_0 = 0$
2. $i += 1$; select $v_i = G^{-1}(G(v_i) - \ln(r_1))$ with $g(v) \geq f(v)$
3. Compare r_2 to $f(v_i)/g(v_i)$

   if $f(v_i)/g(v_i) \leq r_2$: return to 2

   else: accept v_i
4. Continue until v_i reaches v_{cutoff}

The other splitting variables (z and ϕ) can then be generated using standard Monte-Carlo techniques

Selecting z

Aim: select z such that probability to find z in an interval dz is $f(z)dz$

Selecting z

Aim: select z such that probability to find z in an interval dz is $f(z)dz$

- A fraction r of total area under $f(z)$ should be to the left of z

$$\int_{z_{\min}}^{z} dz' f(z') = r \int_{z_{\min}}^{z_{\max}} dz' f(z')$$

Solve for $z \rightarrow$ $z = F^{-1}(F(z_{\min}) + r(F(z_{\max}) - F(z_{\min}))$

Selecting z

Aim: select z such that probability to find z in an interval dz is $f(z)dz$

- A fraction r of total area under $f(z)$ should be to the left of z

$$\int_{z_{\min}}^{z} dz' f(z') = r \int_{z_{\min}}^{z_{\max}} dz' f(z')$$

Solve for z → $z = F^{-1}(F(z_{\min}) + r(F(z_{\max}) - F(z_{\min}))$

- This assumes everything is known analytically → if not we can use the same 'overestimate' trick as before

Also e.g. multiple splitting channels ($g \to gg$, $g \to q\bar{q}$) and running α_s effects can be accounted for this way

Momentum mapping

Basic problem: after every splitting, we need to decide how to go from n momenta to $n+1$ momenta

Momentum mapping

Basic problem: after every splitting, we need to decide how to go from n momenta to $n+1$ momenta

- A few 'rules':
  - We cannot violate total momentum conservation
  - We have to make sure that all particles are on-shell at every step, as the shower can terminate at any moment
  - In the collinear limit we need to identify a (IR safe) momentum fraction z

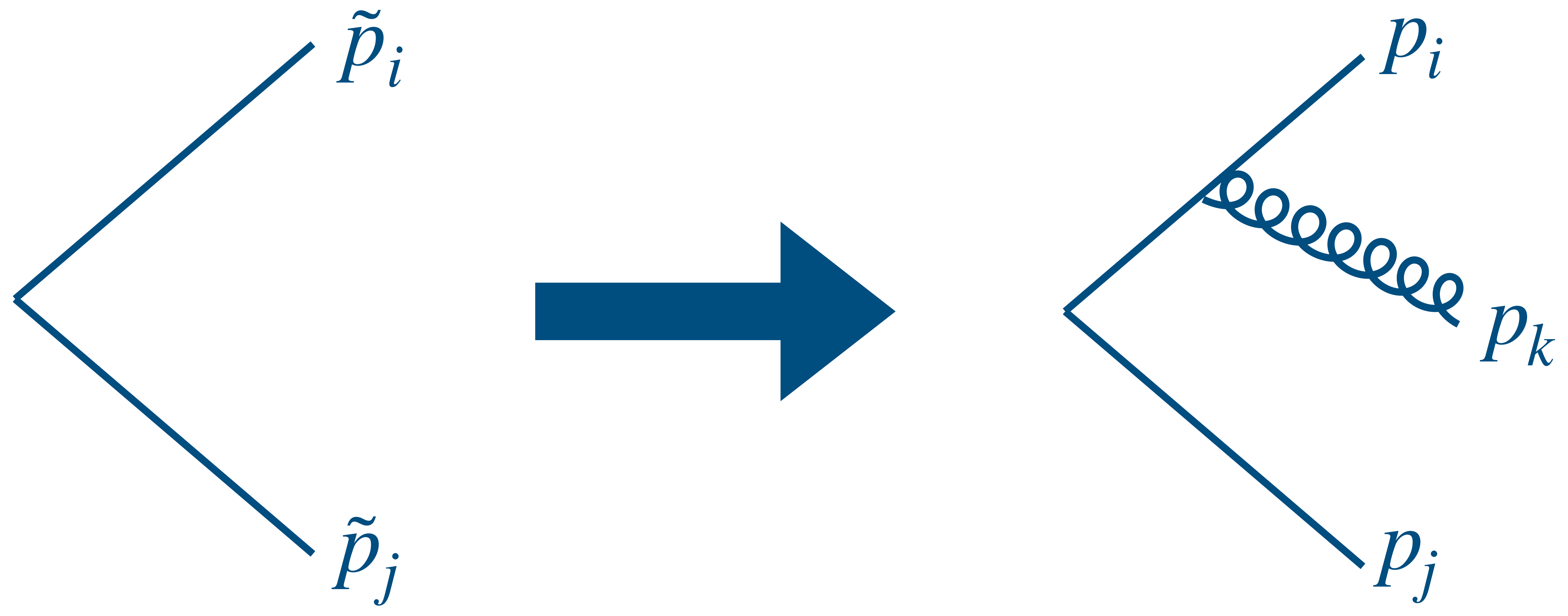
Momentum mapping

Basic problem: after every splitting, we need to decide how to go from n momenta to $n+1$ momenta

- A few 'rules':
  - We cannot violate total momentum conservation
  - We have to make sure that all particles are on-shell at every step, as the shower can terminate at any moment
  - In the collinear limit we need to identify a (IR safe) momentum fraction z
- Other than that: complete freedom
  - Reshuffle momenta 'dipole local' or event-wide (global)?
  - What does the map do beyond the strict unresolved limit?
  - How to relate the three randomly generated kinematic variables to the exact mapping?

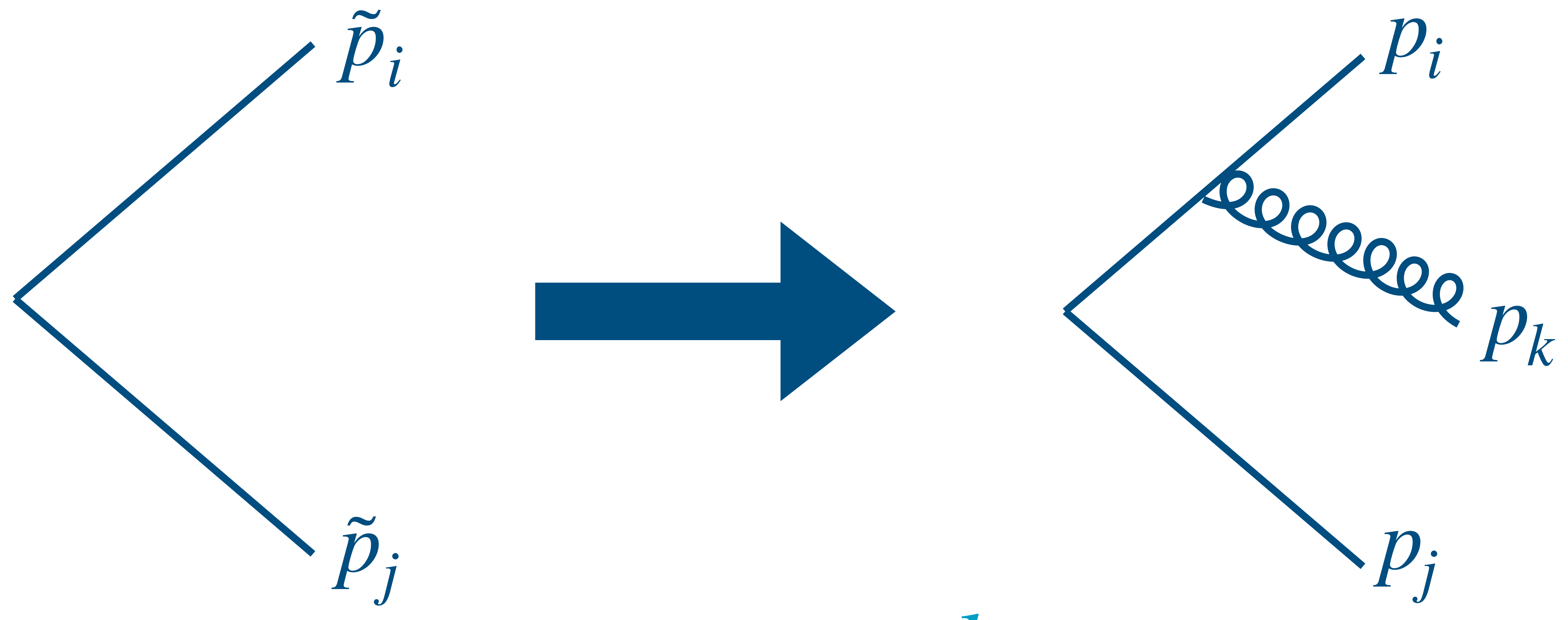
Momentum mapping

- Example: Pythia final-state dipole-local map



Momentum mapping

- Example: Pythia final-state dipole-local map



Emitter: $p_i = a_i \tilde{p}_i + b_i \tilde{p}_j + k_\perp$

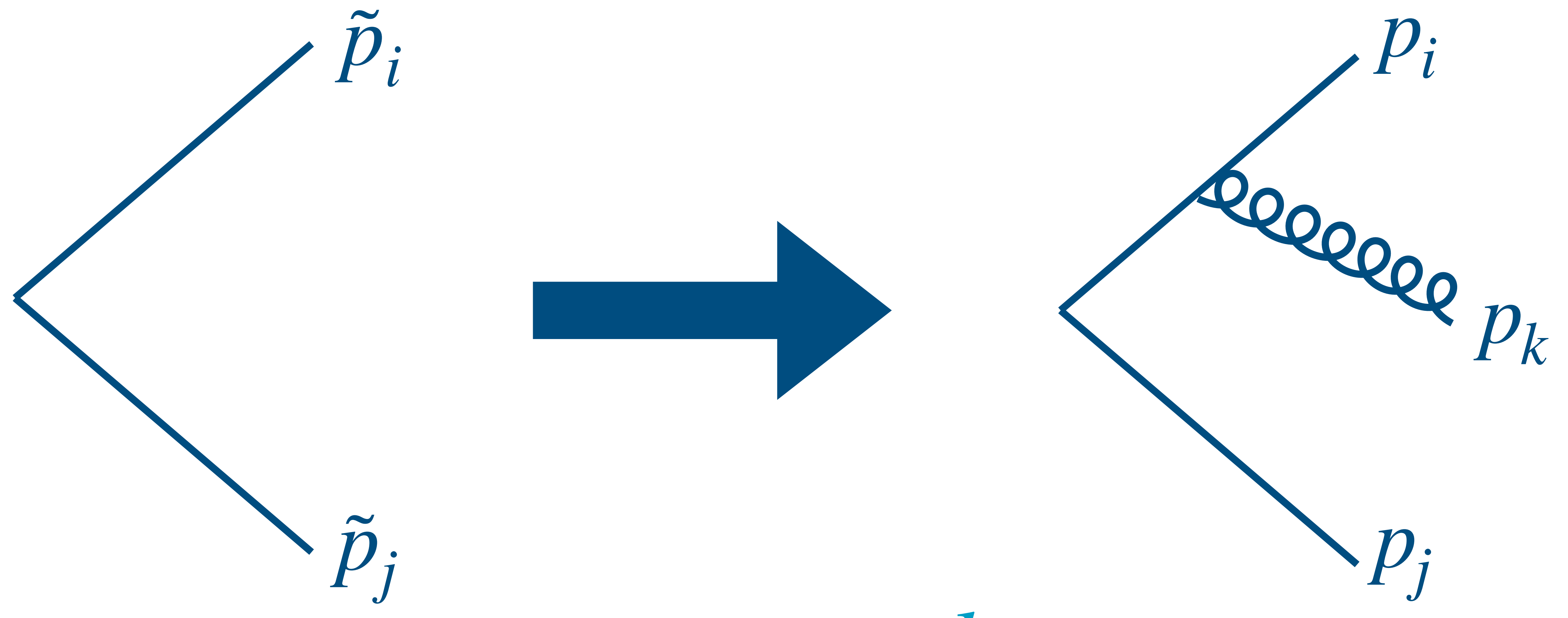
Emission (new): $p_k = a_k \tilde{p}_i + b_k \tilde{p}_j - k_\perp$

Spectator: $p_j = (1 - b_j) \tilde{p}_j$

- Transverse momentum $k_\perp$ is orthogonal to pre-splitting dipole momenta $\tilde{p}_i$, $\tilde{p}_j$ (and depends on ϕ)

Momentum mapping

- Example: Pythia final-state dipole-local map



Emitter: $p_i = a_i\tilde{p}_i + b_i\tilde{p}_j + k_\perp$

Emission (new): $p_k = a_k\tilde{p}_i + b_k\tilde{p}_j - k_\perp$

Spectator: $p_j = (1 - b_j)\tilde{p}_j$

- Transverse momentum $k_\perp$ is orthogonal to pre-splitting dipole momenta $\tilde{p}_i$, $\tilde{p}_j$ (and depends on ϕ)
- Constraints:
  - Momenta are assumed to be massless: $p_i^2 = p_j^2 = p_k^2 = 0$
  - Sum should be the same before and after: $p_i + p_j + p_k = \tilde{p}_i + \tilde{p}_j$

Momentum mapping

- Example: Pythia final-state dipole-local map

Emitter: $p_i = a_i\tilde{p}_i + b_i\tilde{p}_j + k_\perp$

Emission (new): $p_k = a_k\tilde{p}_i + b_k\tilde{p}_j - k_\perp$

Spectator: $p_j = (1 - b_j)\tilde{p}_j$

- Choose to define

$$\kappa_\perp^2 \equiv \frac{v^2}{2\tilde{p}_i \cdot \tilde{p}_j} \qquad y \equiv \frac{\kappa_\perp^2}{z(1-z)} \qquad \tilde{z} \equiv \frac{z(1+y) - y}{1 - y}$$

Momentum mapping

- Example: Pythia final-state dipole-local map

Emitter: $p_i = a_i \tilde{p}_i + b_i \tilde{p}_j + k_\perp$

Emission (new): $p_k = a_k \tilde{p}_i + b_k \tilde{p}_j - k_\perp$

Spectator: $p_j = (1 - b_j)\tilde{p}_j$

- Choose to define

$$\kappa_\perp^2 \equiv \frac{v^2}{2\tilde{p}_i \cdot \tilde{p}_j} \qquad y \equiv \frac{\kappa_\perp^2}{z(1-z)} \qquad \tilde{z} \equiv \frac{z(1+y) - y}{1-y}$$

- And then set $a_k = 1 - \tilde{z}$ and $b_k = y\tilde{z}$
  - for collinear emissions: $y \to 0$ and $\tilde{z} \simeq z$
  - for soft emissions: $\tilde{z} \to 1$ with $y \to 0$

Momentum mapping

- Example: Pythia final-state dipole-local map

Emitter: $p_i = a_i \tilde{p}_i + b_i \tilde{p}_j + k_\perp$

Emission (new): $p_k = a_k \tilde{p}_i + b_k \tilde{p}_j - k_\perp$

Spectator: $p_j = (1 - b_j)\tilde{p}_j$

- Choose to define

$$\kappa_\perp^2 \equiv \frac{v^2}{2\tilde{p}_i \cdot \tilde{p}_j} \qquad y \equiv \frac{\kappa_\perp^2}{z(1-z)} \qquad \tilde{z} \equiv \frac{z(1+y) - y}{1-y}$$

- And then set $a_k = 1 - \tilde{z}$ and $b_k = y\tilde{z}$
  - for collinear emissions: $y \to 0$ and $\tilde{z} \simeq z$
  - for soft emissions: $\tilde{z} \to 1$ with $y \to 0$

- From the constraints one can derive the other coefficients and $k_\perp^2$

Momentum mapping

- Example: Pythia final-state dipole-local map

Emitter: $p_i = a_i \tilde{p}_i + b_i \tilde{p}_j + k_\perp$

Emission (new): $p_k = a_k \tilde{p}_i + b_k \tilde{p}_j - k_\perp$

Spectator: $p_j = (1 - b_j)\tilde{p}_j$

- Choose to define

$$\kappa_\perp^2 \equiv \frac{v^2}{2\tilde{p}_i \cdot \tilde{p}_j} \qquad y \equiv \frac{\kappa_\perp^2}{z(1-z)} \qquad \tilde{z} \equiv \frac{z(1+y) - y}{1-y}$$

- And then set $a_k = 1 - \tilde{z}$ and $b_k = y\tilde{z}$
  - for collinear emissions: $y \to 0$ and $\tilde{z} \simeq z$
  - for soft emissions: $\tilde{z} \to 1$ with $y \to 0$

But this is just one set of choices!

Connection to resummation

What all showers do:

- Decide on a physical meaning of v
- Implement some form of momentum mapping
- Distribute probabilities proportional to the splitting functions

Where showers differ:

- What happens after one single emission (i.e. a correlated pair of emissions)?
- What happens outside the strict collinear and soft limits?

Connection to resummation

What all showers do:

- Decide on a physical meaning of v
- Implement some form of momentum mapping
- Distribute probabilities proportional to the splitting functions

Where showers differ:

important for resummation

- What happens after one single emission (i.e. a correlated pair of emissions)?
- What happens outside the strict collinear and soft limits?

important for higher-order matching

Connection to resummation

What happens after one emission?

That emission can itself become an emitter or spectator

Connection to resummation

What happens after one emission?

That emission can itself become an emitter or spectator

Old emission: $p_k = a_i \tilde{p}_k + b_i \tilde{p}_j + k_\perp$ → its momentum can be modified

New emission: $p_{k'} = a_k \tilde{p}_k + b_k \tilde{p}_j - k_\perp$

Spectator: $p_j = (1 - b_j) \tilde{p}_j$

Connection to resummation

What happens after one emission?

That emission can itself become an emitter or spectator

Old emission: $p_k = a_i \tilde{p}_k + b_i \tilde{p}_j + k_\perp$ → its momentum can be modified

New emission: $p_{k'} = a_k \tilde{p}_k + b_k \tilde{p}_j - k_\perp$

Spectator: $p_j = (1 - b_j)\tilde{p}_j$

This is OK in principle, but not if p_k and $p_{k'}$ are not close in either angle or transverse-momentum

Connection to resummation

What happens after one emission?

That emission can itself become an emitter or spectator

Old emission: $p_k = a_i \tilde{p}_k + b_i \tilde{p}_j + k_\perp$ → its momentum can be modified

New emission: $p_{k'} = a_k \tilde{p}_k + b_k \tilde{p}_j - k_\perp$

Spectator: $p_j = (1 - b_j)\tilde{p}_j$

This is OK in principle, but not if p_k and $p_{k'}$ are not close in either angle or transverse-momentum

in such regions collinear/soft factorisation holds

→ QCD tells us that emissions should be independent there

Connection to resummation

What happens after one emission?

That emission can itself become an emitter or spectator

Old emission: $p_k = a_i \tilde{p}_k + b_i \tilde{p}_j + k_\perp$ → its momentum can be modified

New emission: $p_{k'} = a_k \tilde{p}_k + b_k \tilde{p}_j - k_\perp$

Spectator: $p_j = (1 - b_j) \tilde{p}_j$

This is OK in principle, but not if p_k and $p_{k'}$ are not close in either angle or transverse-momentum

in such regions collinear/soft factorisation holds

→ QCD tells us that emissions should be independent there

Standard dipole showers (Pythia, Sherpa, …) choose their emitter in the dipole CM frame, which violates this basic QCD principle!

Enough theory - time to get coding!

- In the tutorial, you will learn how to set up a parton shower yourself
- We will use python (even though modern showers are coded up in c++)
- A few pre-written python files will be provided for you, so that you can fully focus on writing your own shower!
- If you have Rivet installed, great! If not, the easiest solution is to connect to Stoomboot and run your code there
- By the end of the tutorial, we will see what needs to change to standard shower algorithms to make them next-to-leading logarithmic accurate

References if you want to know more

1. R. K. Ellis, W. J. Stirling, B. R. Webber
   QCD and Collider Physics

2. T. Sjöstrand, S. Mrenna and P. Z. Skands
   PYTHIA 6.4 Physics and Manual

On NLL showers:

3. M. Dasgupta, F. Dreyer, K. Hamilton, P. Monni, G. Salam
   Logarithmic accuracy of parton showers: a fixed-order study

0 minutes