Parton branching at Amplitude Level

Jack Holguin

In collaboration with Jeff Forshaw and Simon Plätzer

This talk

This talk is based on *Parton branching at amplitude level* [arXiv:1905.08686].

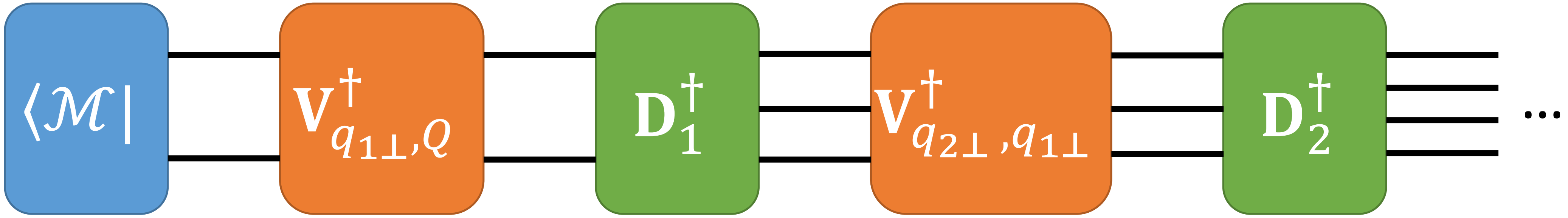
In this 20-minute talk I will present:

- The algorithm in overview.
- A short discussion on handling the soft-collinear region.
- Our observations on collinear factorisation within our algorithm.

In [arXiv:1905.08686] you will also find:

- Alternative formations of the algorithm.
- Examples of analytic resummations calculated using the algorithm.
- Discussions on the spin correlations and colour structures in the algorithm.
- Full definitions of all operators involved, which are often unwieldy.

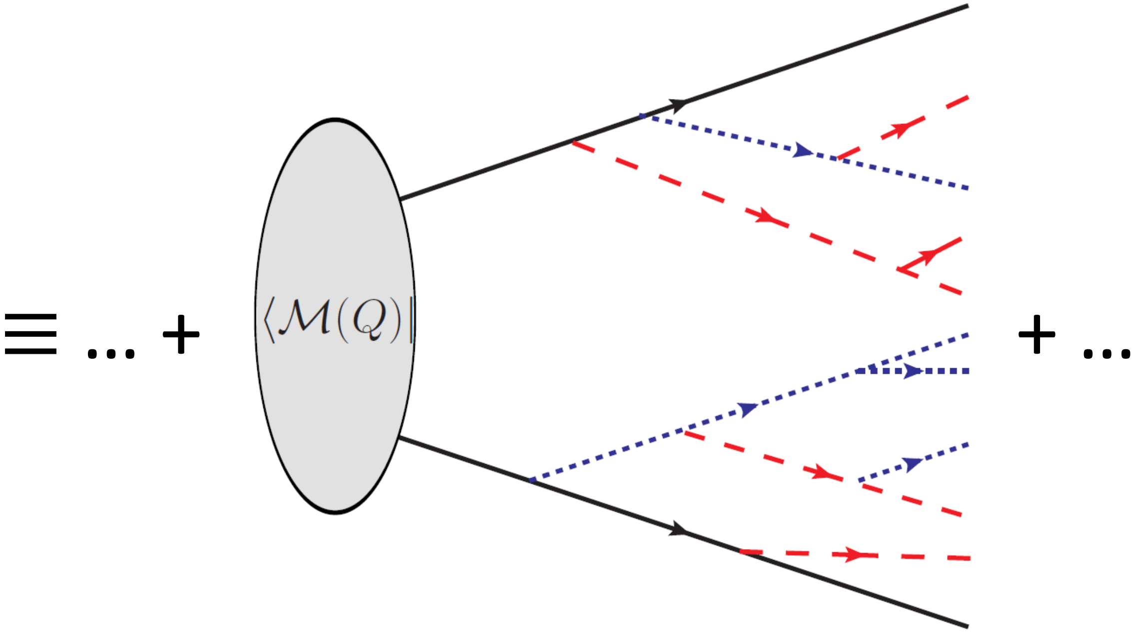
The algorithm



$\langle\mathcal{M}|$ is a conjugate amplitude.

$\mathbf{V}^{\dagger}_{b,a}$ is an amplitude level Sudakov factor (i.e. it has the complete colour structure of exponentiated one loop soft and collinear exchanges).

$\mathbf{D}^{\dagger}_{i}$ is an amplitude level operator that emits a single parton, i.



The algorithm

$$\mathrm{Tr}\left[\mathbf{V}_{\mu,q_{1\perp}}\,\mathbf{D}_1\,\mathbf{V}_{q_{1\perp},Q}\,|\mathcal{M}\rangle\langle\mathcal{M}|\,\mathbf{V}^{\dagger}_{q_{1\perp},Q}\,\mathbf{D}^{\dagger}_1\,\mathbf{V}^{\dagger}_{\mu,q_{1\perp}}\right] = \mathrm{d}\sigma_1$$

$$\begin{aligned}\Sigma(\mu) &= \int \sum_n \mathrm{d}\sigma_n \, u_n(q_1, \dots, q_n), \\ &= \int \sum_n \left(\prod_{i=1}^{n} \mathrm{d}\Pi_i\right) \mathrm{Tr}\,\mathbf{A}_n(\mu; \{p\}_n)\, u_n(q_1, \dots, q_n),\end{aligned}$$

The algorithm (extra details)

$$d\sigma_0 = \mathrm{Tr}\left(\mathbf{V}_{\mu,Q}\mathbf{H}(Q;\{p\})\mathbf{V}^\dagger_{\mu,Q}\right) = \mathrm{Tr}\,\mathbf{A}_0(\mu;\{p\}),$$

$$d\sigma_1 = \int \prod_{i=1}^{n_{\mathrm{H}}+1} d^4p_i\,\mathrm{Tr}\left(\mathbf{V}_{\mu,q_{1\perp}}\mathbf{D}_1\mathbf{V}_{q_{1\perp},Q}\mathbf{H}(Q;\{p\})\mathbf{V}^\dagger_{q_{1\perp},Q}\mathbf{D}_1^\dagger\mathbf{V}^\dagger_{\mu,q_{1\perp}}\right) d\Pi_1$$

$$= \mathrm{Tr}\,\mathbf{A}_1(\mu;\{\tilde{p}\}\cup q_1)\,d\Pi_1,$$

$$d\sigma_n = \mathrm{Tr}\,\mathbf{A}_n(\mu;\{p\}_n)\prod_{i=1}^{n} d\Pi_i,$$

$$\ldots\mathbf{D}_i\mathcal{O}\mathbf{D}_i^\dagger\ldots = \ldots\mathbf{S}_i\mathcal{O}\mathbf{S}_i^\dagger\ldots + \ldots\mathbf{C}_i\mathcal{O}\mathbf{C}_i^\dagger\ldots.$$

$$\mathbf{S}_i = \sum_j\left(\frac{q_{i\perp}^{(j\vec{m})}}{2\tilde{p}_j\cdot q_i}\mathbb{T}_j^g\otimes\left(\tilde{p}_j\cdot\epsilon_+^*(q_i)\mathbb{S}^{1_i}+\tilde{p}_j\cdot\epsilon_-^*(q_i)\mathbb{S}^{-1_i}\right)\right)\mathfrak{R}_{ij}^{\mathrm{soft}}(\{p\},\{\tilde{p}\},q_i),$$

$$\mathbf{C}_i = \sum_j\frac{q_{i\perp}^{(j\vec{n})}}{2\sqrt{z_i}}\Delta_{ij}\,\overline{\mathbf{P}}_{ij}\,\mathfrak{R}_{ij}^{\mathrm{coll}}(\{p\},\{\tilde{p}\},q_i),$$

where

$$\mathbf{A}_n(q_\perp;\{\tilde{p}\}_{n-1}\cup q_n) = \int\prod_{i=1}^{n_{\mathrm{H}}+n} d^4p_i\mathbf{V}_{q_\perp,q_{n\perp}}\mathbf{D}_n\mathbf{A}_{n-1}(q_{n\perp};\{p\}_{n-1})\mathbf{D}_n^\dagger\mathbf{V}^\dagger_{q_\perp,q_{n\perp}}\Theta(q_\perp\le q_{n\perp}). \tag{2.2}$$

The algorithm (extra details)

$$\mathbf{P}_{ij} = \delta_{s_j,\frac{1}{2}}\delta_j^{\text{final}}\left(\sqrt{\frac{\mathcal{P}_{qq}}{2\mathcal{C}_{\text{F}}(1+z_i^2)}}\frac{1}{\langle q_i\tilde{p}_j\rangle}(\mathbb{T}_j^g\otimes\mathbb{S}^{+1_i}) + \sqrt{\frac{z_i^2\mathcal{P}_{qq}}{2\mathcal{C}_{\text{F}}(1+z_i^2)}}\frac{1}{[\tilde{p}_j q_i]}(\mathbb{T}_j^g\otimes\mathbb{S}^{-1_i})\right.$$

$$\left. + \sqrt{\frac{\mathcal{P}_{gq}}{2\mathcal{C}_{\text{F}}(2-2z_i+z_i^2)}}\frac{1}{\langle\tilde{p}_j q_i\rangle}\mathbb{W}^{ij}(\mathbb{T}_j^g\otimes\mathbb{S}^{+1_i}) + \sqrt{\frac{(1-z_i)^2\mathcal{P}_{gq}}{2\mathcal{C}_{\text{F}}(2-2z_i+z_i^2)}}\frac{1}{[q_i\tilde{p}_j]}\mathbb{W}^{ij}(\mathbb{T}_j^g\otimes\mathbb{S}^{-1_i})\right)$$

$$+ \delta_{s_j,\frac{1}{2}}\delta_j^{\text{initial}}\sqrt{\frac{1}{z_i}}\left(\sqrt{\frac{\mathcal{P}_{qq}}{\mathcal{C}_{\text{F}}(1+z_i^2)}}\frac{1}{\langle q_i p_j\rangle}(\mathbb{T}_j^g\otimes\mathbb{S}^{+1_i}) + \sqrt{\frac{z_i^2\mathcal{P}_{qq}}{\mathcal{C}_{\text{F}}(1+z_i^2)}}\frac{1}{[p_j q_i]}(\mathbb{T}_j^g\otimes\mathbb{S}^{-1_i})\right.$$

$$+ \sqrt{\frac{(1-z_i)^2\mathcal{P}_{qg}}{n_f\mathcal{C}_F(1-2z_i(1-z_i))}}\frac{1}{[p_j q_i]}\mathbb{W}^{ij}(\mathbb{T}_j^g\otimes\mathbb{S}^{+1_i})$$

$$\left. + \sqrt{\frac{z_i^2\mathcal{P}_{qg}}{n_f\mathcal{C}_F(1-2z_i(1-z_i))}}\frac{1}{\langle q_i p_j\rangle}\mathbb{W}^{ij}(\mathbb{T}_j^g\otimes\mathbb{S}^{-1_i})\right)$$

$$+\cdots$$

These are the terms that generate emissions from a positive helicity electron.

Soft-collinear contributions

$$\langle\mathcal{M}|\mathbf{D}_1^\dagger\mathbf{D}_1|\mathcal{M}\rangle = \langle\mathcal{M}|\mathbf{SoftCol}|\mathcal{M}\rangle + \langle\mathcal{M}|\mathbf{WideSoft}|\mathcal{M}\rangle + \langle\mathcal{M}|\mathbf{HardCol}|\mathcal{M}\rangle + \langle\mathcal{M}|\mathbf{Remainder}|\mathcal{M}\rangle$$

$\langle\mathcal{M}|\mathbf{Eikonal}|\mathcal{M}\rangle = \langle\mathcal{M}|\mathbf{SoftCol}|\mathcal{M}\rangle + \langle\mathcal{M}|\mathbf{WideSoft}|\mathcal{M}\rangle$,

$\langle\mathcal{M}|\mathbf{SplittingFunctions}|\mathcal{M}\rangle = \langle\mathcal{M}|\mathbf{SoftCol}|\mathcal{M}\rangle + \langle\mathcal{M}|\mathbf{HardCol}|\mathcal{M}\rangle$,

we ignore $\langle\mathcal{M}|\mathbf{Remainder}|\mathcal{M}\rangle$ as not it's logarithmically enhanced.

So $\langle\mathcal{M}|\mathbf{D}_1^\dagger\mathbf{D}_1|\mathcal{M}\rangle = \langle\mathcal{M}|\mathbf{Eikonal}|\mathcal{M}\rangle + \langle\mathcal{M}|\mathbf{SplittingFunctions}|\mathcal{M}\rangle - \langle\mathcal{M}|\mathbf{SoftCol}|\mathcal{M}\rangle$

i.e. we can't just interleave soft and collinear emissions, else we double count the soft-collinear region.

Soft-collinear contributions

Hence two versions of the algorithm, A and B. In A the soft-collinear region is removed from **C**. In B it is removed from **S**.

$$\mathbf{S}_i^{\mathrm{B}}\mathcal{O}\mathbf{S}_i^{\mathrm{B}\,\dagger}\,\mathrm{d}\Pi_i + \mathbf{C}_i^{\mathrm{B}}\mathcal{O}\mathbf{C}_i^{\mathrm{B}\,\dagger}\,\mathrm{d}\Pi_i = \underbrace{\mathbf{S}_i^{\mathrm{A}}\mathcal{O}\mathbf{S}_i^{\mathrm{A}\,\dagger}\,\mathrm{d}\Pi_i - \mathbf{s}_i\mathcal{O}\mathbf{s}_i^\dagger}_{\equiv \mathbf{S}_i^{\mathrm{B}}\mathcal{O}\mathbf{S}_i^{\mathrm{B}\,\dagger}\,\mathrm{d}\Pi_i} + \underbrace{\mathbf{C}_i^{\mathrm{A}\,\dagger}\mathcal{O}\mathbf{C}_i^{\mathrm{A}}\,\mathrm{d}\Pi_i + \mathbf{s}_i\mathcal{O}\mathbf{s}_i^\dagger}_{\equiv \mathbf{C}_i^{\mathrm{B}}\mathcal{O}\mathbf{C}_i^{\mathrm{B}\,\dagger}\,\mathrm{d}\Pi_i},$$

To make this subtraction we need to be able to accurately compute $\langle\mathcal{M}|\mathbf{SoftCol}|\mathcal{M}\rangle$. This can be done reasonably simply from the collinear side, just isolate the terms with soft poles.

$$\frac{1+z^2}{1-z} = \frac{2}{1-z} - (1+z)$$

Note that in the final state soft poles are also found at $z = 0$. In the initial state the poles disappear as they are screened by PDFs.

Soft-collinear contributions

Hence two versions of the algorithm, A and B. In A the soft-collinear region is removed from **C**. In B it is removed from **S**.

$$\mathbf{S}_i^{\mathrm{B}}\mathcal{O}\mathbf{S}_i^{\mathrm{B}\,\dagger}\,\mathrm{d}\Pi_i+\mathbf{C}_i^{\mathrm{B}}\mathcal{O}\mathbf{C}_i^{\mathrm{B}\,\dagger}\,\mathrm{d}\Pi_i=\underbrace{\mathbf{S}_i^{\mathrm{A}}\mathcal{O}\mathbf{S}_i^{\mathrm{A}\,\dagger}\,\mathrm{d}\Pi_i-\mathbf{s}_i\mathcal{O}\mathbf{s}_i^{\dagger}}_{\equiv\mathbf{S}_i^{\mathrm{B}}\mathcal{O}\mathbf{S}_i^{\mathrm{B}\,\dagger}\,\mathrm{d}\Pi_i}+\underbrace{\mathbf{C}_i^{\mathrm{A}\,\dagger}\mathcal{O}\mathbf{C}_i^{\mathrm{A}}\,\mathrm{d}\Pi_i+\mathbf{s}_i\mathcal{O}\mathbf{s}_i^{\dagger}}_{\equiv\mathbf{C}_i^{\mathrm{B}}\mathcal{O}\mathbf{C}_i^{\mathrm{B}\,\dagger}\,\mathrm{d}\Pi_i},$$

We can also compute $\langle\mathcal{M}|\mathbf{SoftCol}|\mathcal{M}\rangle$ from analysing pole terms in the eikonal currents. Doing so allows us to pull the wide angle soft contribution out from the Eikonal currents. However the process is more lengthy, Matthew will discuss this more in his talk. The important take away is that both approaches give the same result.

The algorithm (extra details), Variant A

$$d\sigma_0 = \mathrm{Tr}\left(\mathbf{V}_{\mu,Q}\mathbf{H}(Q;\{p\})\mathbf{V}^\dagger_{\mu,Q}\right) = \mathrm{Tr}\,\mathbf{A}_0(\mu;\{p\}),$$

$$d\sigma_1 = \int\prod_{i=1}^{n_{\mathrm{H}}+1} d^4p_i\,\mathrm{Tr}\left(\mathbf{V}_{\mu,q_{1\perp}}\mathbf{D}_1\mathbf{V}_{q_{1\perp},Q}\mathbf{H}(Q;\{p\})\mathbf{V}^\dagger_{q_{1\perp},Q}\mathbf{D}_1^\dagger\mathbf{V}^\dagger_{\mu,q_{1\perp}}\right)d\Pi_1$$

$$= \mathrm{Tr}\,\mathbf{A}_1(\mu;\{\tilde{p}\}\cup q_1)\,d\Pi_1,$$

$$d\sigma_n = \mathrm{Tr}\,\mathbf{A}_n(\mu;\{p\}_n)\prod_{i=1}^{n} d\Pi_i,$$

$$\mathbf{S}_i = \sum_j \left(\frac{q_{i\perp}^{(j\vec{m})}}{2\tilde{p}_j\cdot q_i}\mathbb{T}_j^g \otimes (\tilde{p}_j\cdot\epsilon_+^*(q_i)\mathbb{S}^{1_i} + \tilde{p}_j\cdot\epsilon_-^*(q_i)\mathbb{S}^{-1_i})\right)\mathfrak{R}_{ij}^{\mathrm{soft}}(\{p\},\{\tilde{p}\},q_i),$$

$$\mathbf{C}_i = \sum_j \frac{q_{i\perp}^{(j\vec{n})}}{2\sqrt{z_i}}\Delta_{ij}\,\overline{\mathbf{P}}_{ij}\,\mathfrak{R}_{ij}^{\mathrm{coll}}(\{p\},\{\tilde{p}\},q_i),$$

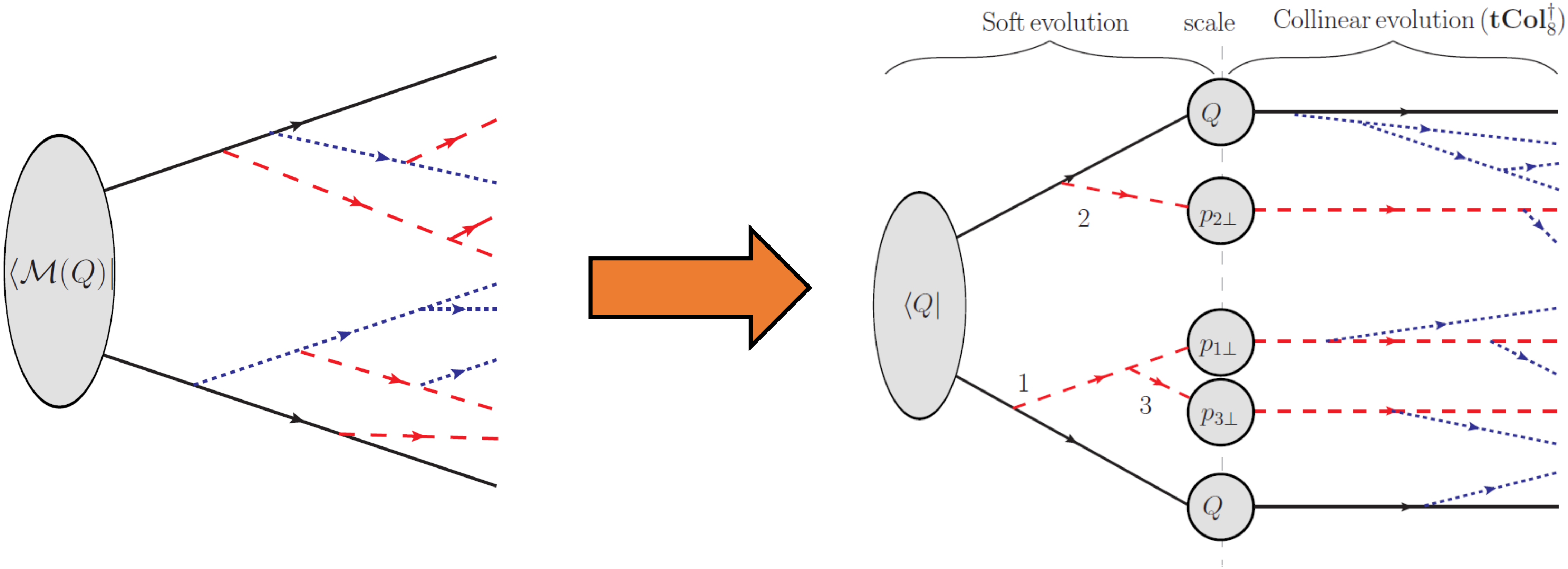
$$\ldots\mathbf{D}_i\mathcal{O}\mathbf{D}_i^\dagger\ldots = \ldots\mathbf{S}_i\mathcal{O}\mathbf{S}_i^\dagger\ldots + \ldots\mathbf{C}_i\mathcal{O}\mathbf{C}_i^\dagger\ldots.$$

where

$$\mathbf{A}_n(q_\perp;\{\tilde{p}\}_{n-1}\cup q_n) = \int\prod_{i=1}^{n_{\mathrm{H}}+n} d^4p_i\mathbf{V}_{q_\perp,q_{n\perp}}\mathbf{D}_n\mathbf{A}_{n-1}(q_{n\perp};\{p\}_{n-1})\mathbf{D}_n^\dagger\mathbf{V}^\dagger_{q_\perp,q_{n\perp}}\Theta(q_\perp\le q_{n\perp}). \tag{2.2}$$

Hard-collinear operator (what remains after the subtraction, i.e. $-(1+z)$ and other such pieces).

Factorisation

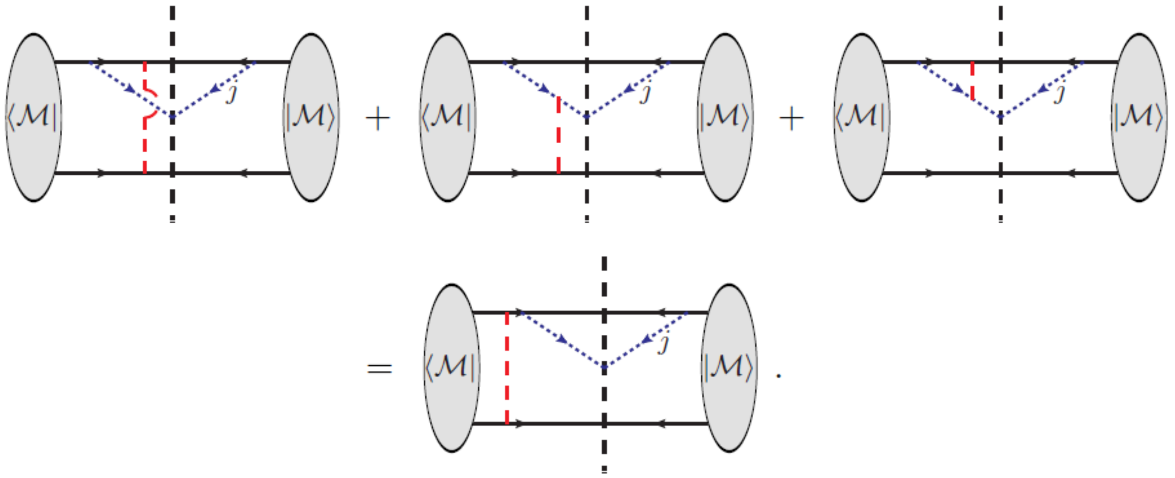


Factorisation

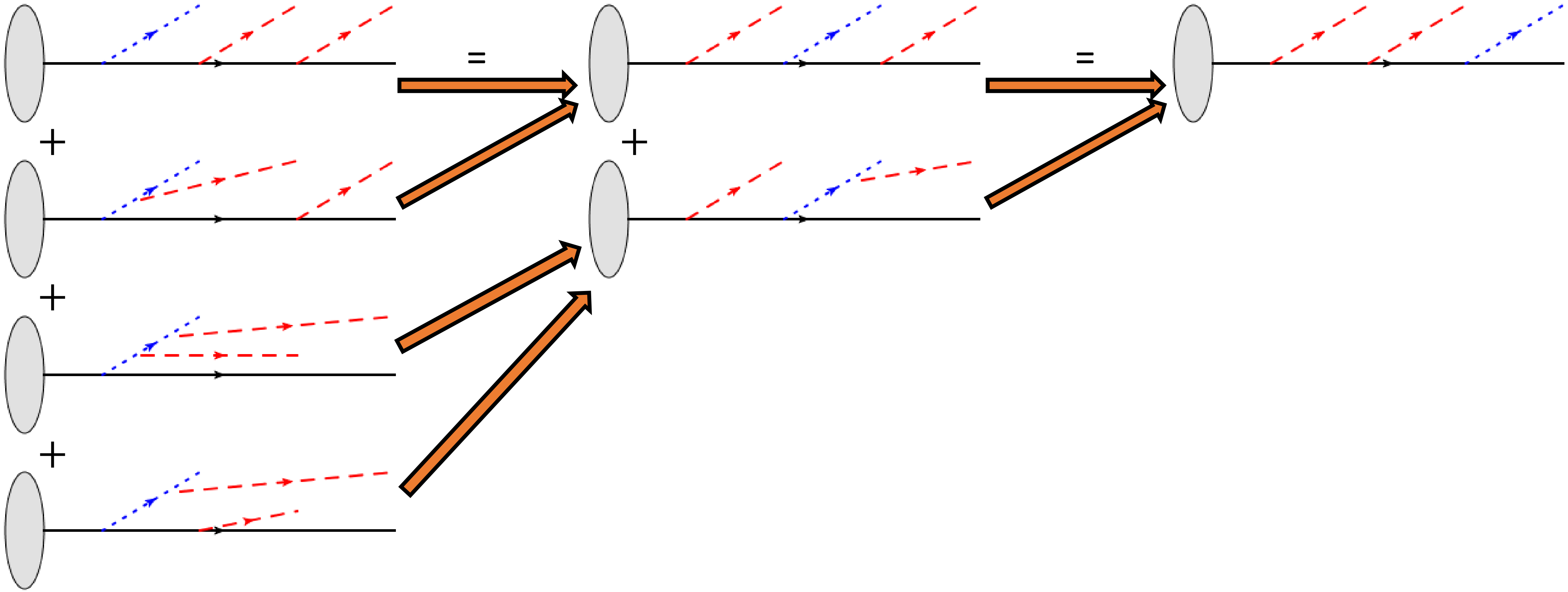
$$\left[\mathbf{D}_i - \overline{\mathbf{C}}_i, \overline{\mathbf{C}}_j\right] \simeq 0, \qquad \left[\mathbf{V}_{a,b}(\mathbf{V}^{\mathrm{col}}_{a,b})^{-1}, \overline{\mathbf{C}}_j\right] \simeq 0,$$

$$\left[\mathbf{V}_{a,b}(\mathbf{V}^{\mathrm{col}}_{a,b})^{-1}, \mathbf{V}^{\mathrm{col}}_{c,d}\right] \simeq 0, \qquad \left[\mathbf{D}_i - \overline{\mathbf{C}}_i, \mathbf{V}^{\mathrm{col}}_{a,b}\right] \simeq 0.$$

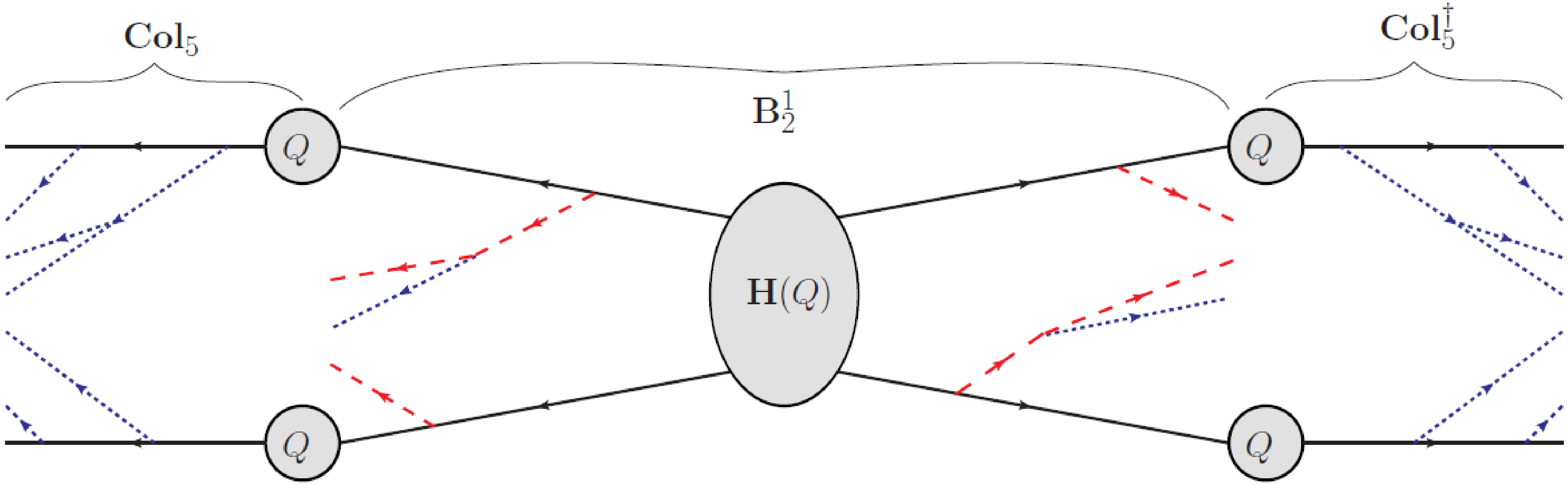
The equality only holds when considering only the real part of these diagrams. The soft loop also generates imaginary parts; Coulomb/Glauber exchanges.



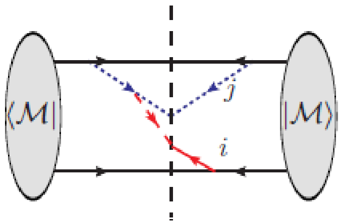
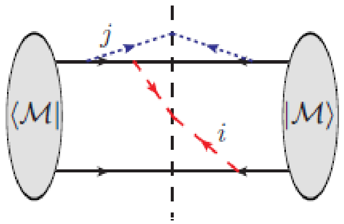
Factorisation

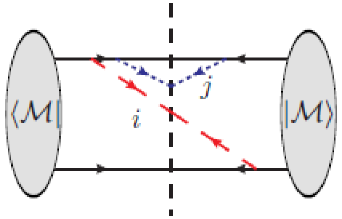
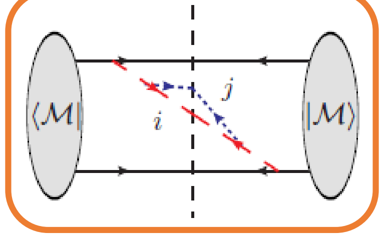


Factorisation

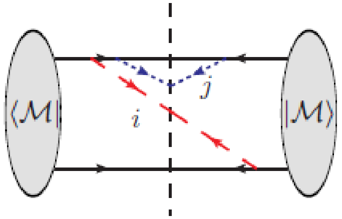


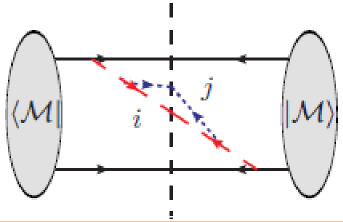
Factorisation

$\Big[$  $+$  $\Big]\ \Theta(q_{j\perp} - q_{i\perp})\Theta(q_{i\perp} - \mu)\Theta(Q - q_{j\perp})$

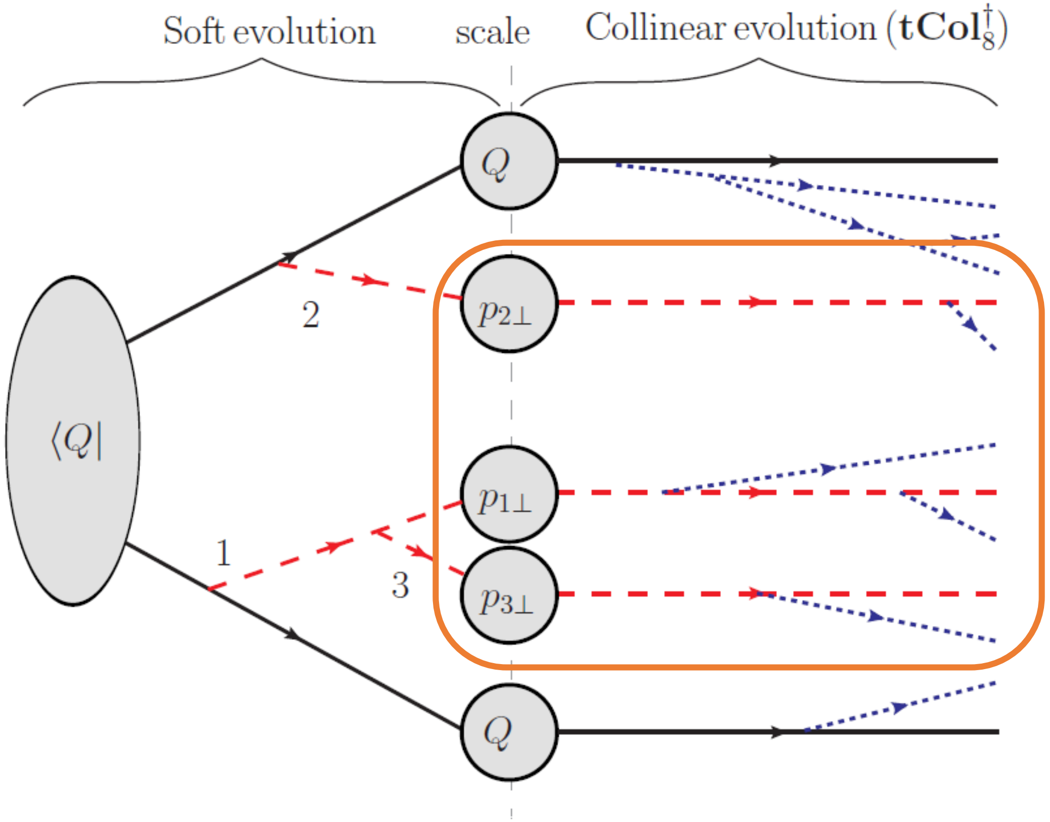
$+\Big[$  $+$  $\Big]\ \Theta(q_{i\perp} - q_{j\perp})\Theta(q_{j\perp} - \mu)\Theta(Q - q_{i\perp})$

(3.32)

$=$  $\Theta(q_{i\perp} - \mu)\Theta(Q - q_{i\perp})\Theta(q_{j\perp} - \mu)\Theta(Q - q_{j\perp})$

$+$  $\Theta(q_{i\perp} - q_{j\perp})\Theta(q_{j\perp} - \mu)\Theta(Q - q_{i\perp}).$

Factorisation



Factorisation

$$\mathbf{V}_{a,b} = \mathrm{P}\ \exp\Bigg[-\frac{\alpha_s}{\pi}\sum_{i<j}\int_a^b \frac{\mathrm{d}k_\perp^{(ij)}}{k_\perp^{(ij)}}(-\mathbb{T}_i^g\cdot\mathbb{T}_j^g)\left\{\int\frac{\mathrm{d}y\,\mathrm{d}\phi}{4\pi}(k_\perp^{(ij)})^2\frac{\tilde{p}_i\cdot\tilde{p}_j}{(\tilde{p}_i\cdot k)(\tilde{p}_j\cdot k)}\theta_{ij}(k) - i\pi\,\tilde{\delta}_{ij}\right\}$$

$$\times\mathcal{R}_{ij}^{\mathrm{soft}}(k,\{\tilde{p}\}) - \frac{\alpha_s}{\pi}\sum_i\int_a^b\frac{\mathrm{d}k_\perp^{(i\bar{n})}}{k_\perp^{(i\bar{n})}}\sum_{v\in\{q,g\}}\mathbb{T}_i^{\bar{v}\,2}\int\frac{\mathrm{d}z\,\mathrm{d}\phi}{8\pi}\overline{\mathcal{P}}^{\circ}_{vv_i}(z)\,\theta_i(k)\,\mathcal{R}_i^{\mathrm{coll}}(k,\{\tilde{p}\})\Bigg],$$

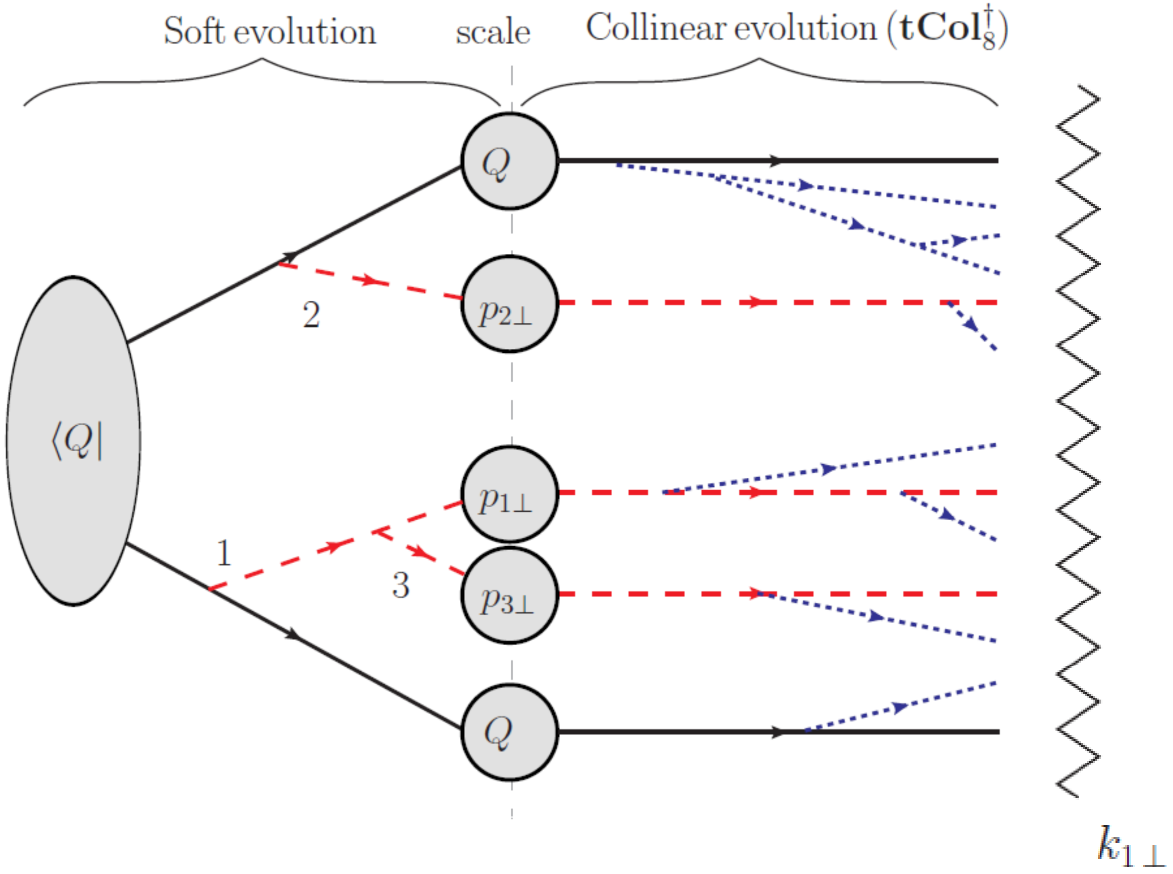
$$\mathbf{V}_{a,b} = \hat{\mathbf{V}}_{a,b} - \frac{\alpha_s}{\pi}\sum_{i_1<j_1}\int_a^b\frac{\mathrm{d}k_{1\perp}^{(i_1j_1)}}{k_{1\perp}^{(i_1j_1)}}\hat{\mathbf{V}}_{a,k_{1\perp}}(\mathbb{T}_{i_1}^g\cdot\mathbb{T}_{j_1}^g)\,i\pi\,\tilde{\delta}_{i_1j_1}\hat{\mathbf{V}}_{k_{1\perp},b}$$

$$+\left(\frac{\alpha_s}{\pi}\right)^2\sum_{i_2<j_2}\int_a^b\frac{\mathrm{d}k_{1\perp}^{(i_1j_1)}}{k_{1\perp}^{(i_1j_1)}}\sum_{i_1<j_1}\int_a^{k_{1\perp}^{(i_1j_1)}}\frac{\mathrm{d}k_{2\perp}^{(i_2j_2)}}{k_{2\perp}^{(i_2j_2)}}\hat{\mathbf{V}}_{a,k_{2\perp}}(\mathbb{T}_{i_2}^g\cdot\mathbb{T}_{j_2}^g)\,i\pi\,\tilde{\delta}_{i_2j_2}$$

$$\times\hat{\mathbf{V}}_{k_{2\perp},k_{1\perp}}(\mathbb{T}_{i_1}^g\cdot\mathbb{T}_{j_1}^g)\,i\pi\tilde{\delta}_{i_1j_1}\hat{\mathbf{V}}_{k_{1\perp},b} - \ldots,$$

To interleave Coulomb terms we use a path ordered expansion around the “$i\pi$” terms. Following this we can carefully interleave them into a factorised evolution.

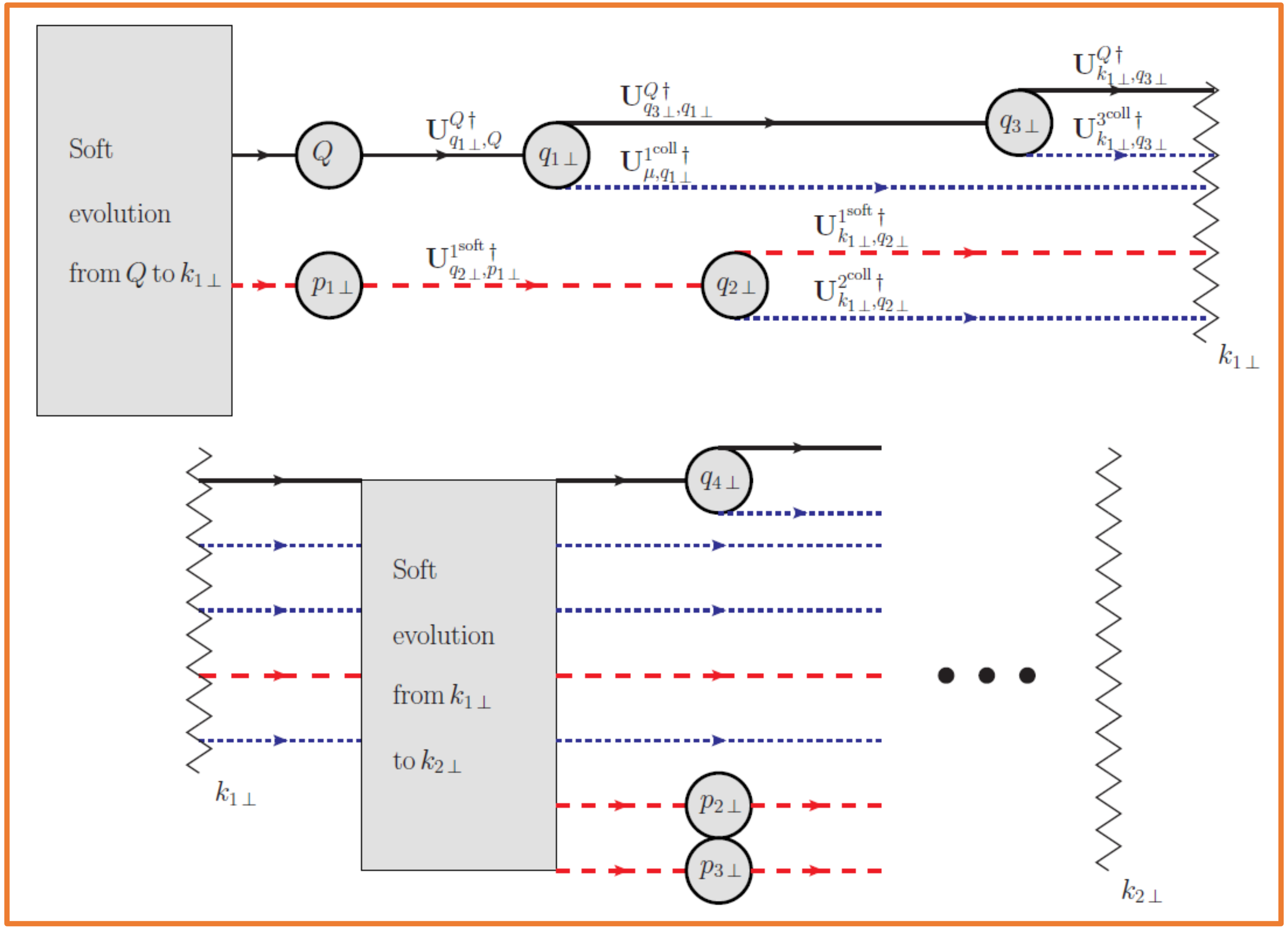
Factorisation



In a practical calculation, this means we can include Coulomb terms by using the factorised algorithm and terminating the evolution at the coulomb scale. After this you then perform a second evolution, using the output of the first as the hard process (initial condition). This second evolution runs from the first Coulomb scale and terminates on a second. Etc.

Finally we must integrate the Coulomb scales over the full ranged allowed by the ordering.

Factorisation



Conclusions

- We've explored the theoretical basis for an algorithm for parton evolution at amplitude level.
- This work will be used to inform future work on the CVolver code. Matthew will discuss this.
- Independent of CVolver, development of this algorithm has opened a number of future avenues for research: i.e.
  - Rederiving current algorithms for parton showers to try and evaluate their accuracy.
  - Re-formulating our algorithm as evolution equations might allow us to make a direct link to SCET.