

# Transverse Momentum Resummation

Alexander Huss

August 30, 2023

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b><math>q_T</math> resummation</b>	<b>1</b>
<b>3</b>	<b>Implementation</b>	<b>2</b>
3.1	Python . . . . .	2
3.2	Mathematica . . . . .	3
<b>4</b>	<b>Playground</b>	<b>3</b>
4.1	Low- $q_T^2$ behaviour . . . . .	3
4.2	Transverse Momentum Distributioins . . . . .	5

## 1 Introduction

In the lectures we have seen a brief overview of the  $q_T$  resummation formalism for the Drell-Yan process. We will have a closer look at the main results here and highlight some features.

## 2 $q_T$ resummation

In the leading double-logarithmic approximation, we have found the following result in impact parameter space

$$\frac{1}{\sigma_0} \frac{d\sigma}{dq_T^2} = \int_0^\infty db \frac{b}{2} J_0(q_T b) \exp \left[ -\frac{\alpha_s}{2\pi} C_F \ln^2(Q^2 b^2) \right], \quad (1)$$

where we have completely ignored effects from sub-leading logarithms, the running of the strong coupling, and parton distributions functions. Nonetheless, this simple formula already allows us to inspect some important features of  $q_T$  resummation, which we will inspect in the following.

## 3 Implementation

We start with a simple implementation of the above formula.

### 3.1 Python

The integral is a bit nasty because of the oscillating behaviour of the Bessel function  $J_0$  so we need to adjust the `scipy.integrate` settings a little bit to reach a desired accuracy. Despite that, the implementation is straightforward:

---

```
#!/usr/bin/env python
import sys
from math import pi, exp, log, log10, ceil, floor
from scipy.special import jv # Bessel function of the 1st kind
from scipy.integrate import quad
import numpy as np

alpha_s: float = 0.118

def res_integrand(b: float, QT: float, Q: float, CX: float) -> float:
    b0: float = 2. * exp(-0.57721566490153286061)
    # blim: float = 5. # should be > 1/Lambda_QCD ~ 5
    # bs2: float = b**2 * blim**2 / (b**2 + blim**2)
    # return (b / 2.) * jv(0, b * QT) * exp(
    #     -alpha_s / (2. * pi) * CX * log(Q**2 * bs2 / b0**2 + 1.)**2)
    return (b / 2.) * jv(0, b * QT) * exp(
        -alpha_s / (2. * pi) * CX * log(Q**2 * b**2)**2)

if __name__ == "__main__":
    if len(sys.argv) < 3:
        raise RuntimeError("I expect at least two arguments: Q [g|q]")
    Q = float(sys.argv[1]) # the hard scale
    pow_low = -3
    pow_upp = ceil(log10(Q/2.)) # floor(log10(Q))
    if sys.argv[2].lower() == "q":
        CX = 4. / 3.
    elif sys.argv[2].lower() == "g":
        CX = 3.
    else:
        raise RuntimeError("unrecognised parton: {}".format(sys.argv[2]))

    if len(sys.argv) >= 4:
        alpha_s = float(sys.argv[3])

    if len(sys.argv) >= 5:
        nsteps = int(sys.argv[4])
    else:
        nsteps = 51

    # print("# qt dSigQT2_val dSigQT2_err")
    for qt in np.logspace(pow_low, pow_upp, nsteps):
        val, err = quad(res_integrand,
            0.,
            np.inf,
            args=(qt, Q, CX),
            epsabs=0.,
```

---

```

        epsrel=1e-4,
        limit=100000)
print("{} {} {}".format(qt, val, err))

```

---

And we can generate some data files for Drell-Yan and Higgs production where in the latter we simply swap out the colour charge  $C_F \rightarrow C_A$  for the gluon-fusion process.

---

```

python main.py 91 q > data_dy.dat
python main.py 125 g > data_h.dat

```

---

## 3.2 Mathematica

To cross-check the numerics, we can use a simple Mathematica implementation

---

```

dSigQT2[qt_] := Module[{cf = 4/3, as = 0.118, q = 91},
  NIntegrate[b/2 BesselJ[0, b*qt] Exp[-(as/(2 Pi)) cf Log[q^2 b^2]^2], {b, 0, Infinity}]
]
datQT2 = Table[{qt, dSigQT2[qt]}, {qt, 10^Range[-4, 2, 0.1]}]
Export["mma_dy.dat", datQT2, "Table", "FieldSeparators" -> " "]

```

---

## 4 Playground

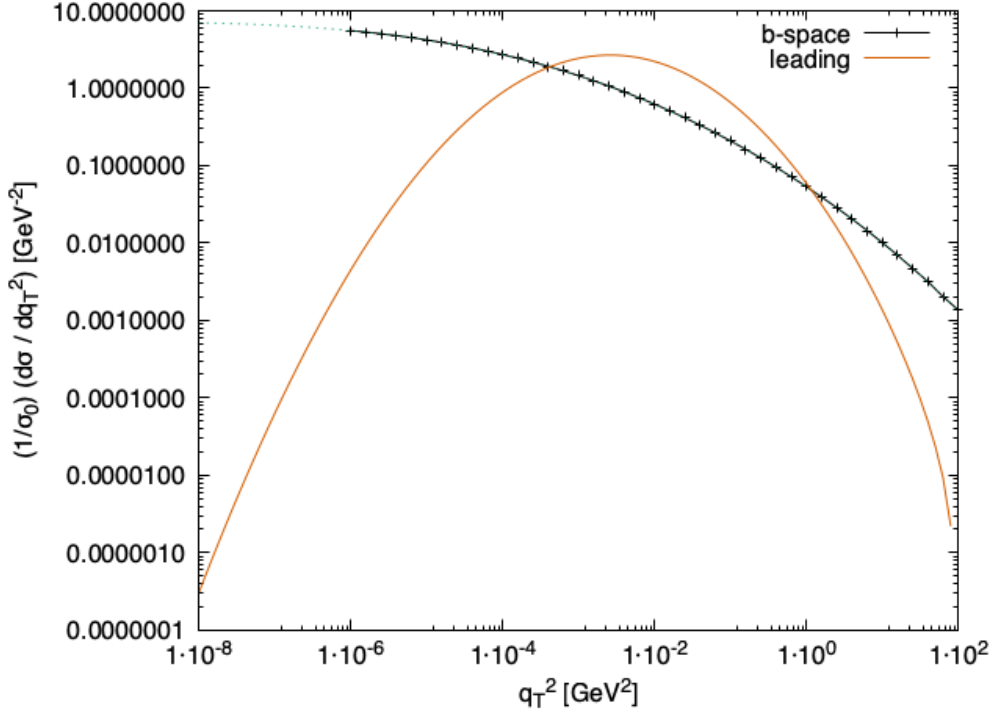
### 4.1 Low- $q_T^2$ behaviour

Let us have a look at the analytic expression for  $d\sigma/dq_T^2$  in the leading double-logarithmic approximation in momentum space.

$$\frac{1}{\sigma_0} \frac{d\sigma}{dq_T^2} = \frac{\alpha_s}{\pi} C_F \frac{\ln(Q^2 q_T^2)}{q_T^2} \exp \left[ -\frac{\alpha_s}{2\pi} C_F \ln^2(Q^2 q_T^2) \right]. \quad (2)$$

This expression can be obtained from Eq. (1) by systematically expanding the Fourier transform or alternatively by naively resumming the emissions *without* the transverse momentum conservation constraint.

We can compare this expression with the numerically evaluated b-space formula from above:



We notice that the leading expression shows a strikingly different behaviour in the small  $q_T^2$  limit compared to the b-space formula. The physical interpretation is quite clear: the leading term corresponds to restricting *all* gluon emissions to have  $k_T$  below the gauge-boson transverse momentum  $q_T$ . This gives a suppression at low  $q_T$  that is stronger than any power and as a consequence, the sub-leading effect suddenly becomes the leading one. In this situation, the small- $q_T$  region is not restricted to only soft gluon emissions but instead by multiple gluon emissions that can individually have  $k_T > q_T$  but they *balance out* in the azimuthal plane. By formulating the resummation in impact parameter space, this feature is automatically incorporated in the prediction.

This non-vanishing intercept in  $d\sigma/dq_T^2$  for  $q_T \rightarrow 0$  is a very important feature of transverse momentum resummation. In fact, we can compute what this intercept is

$$\frac{1}{\sigma_0} \left. \frac{d\sigma}{dq_T^2} \right|_{q_T=0} = \frac{\pi}{2Q^2} \frac{e^{\frac{\pi}{2\alpha_s C_F}}}{\sqrt{2\alpha_s C_F}}. \quad (3)$$

We have also superimposed a dotted line obtained from the Mathematica implementation, which is in good agreement so numerics appear to be under good control in the relevant regions. At high  $q_T$ , the oscillations become very severe rendering the pPython predictions less reliable (with larger integration errors).

## 4.2 Transverse Momentum Distributions

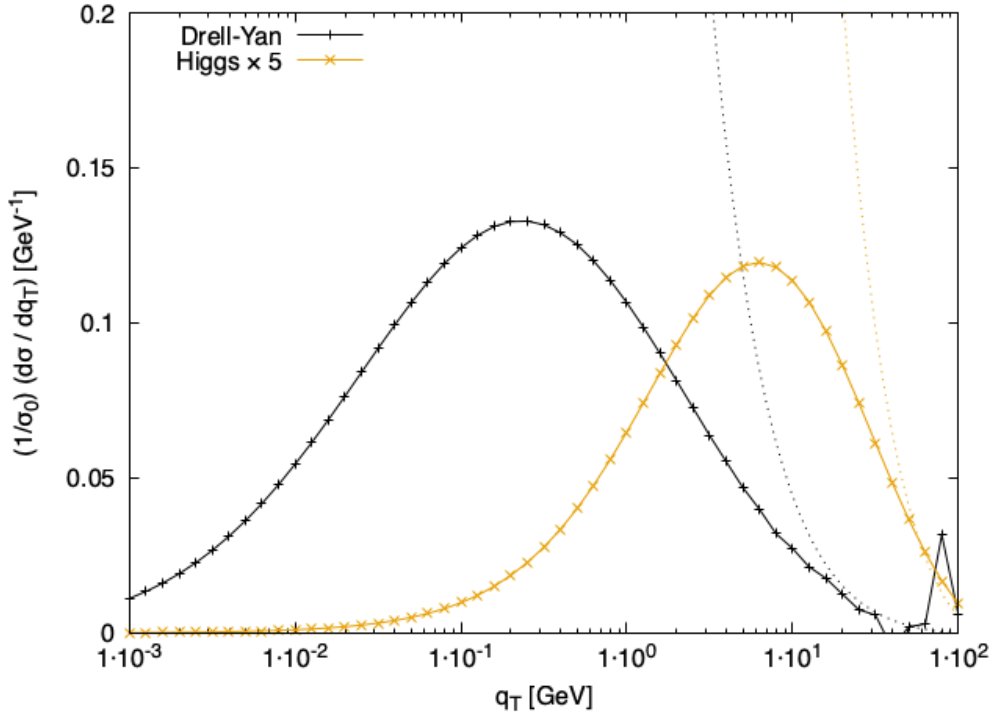
We now look at the transverse momentum distribution, which we simply get by

$$\frac{d\sigma}{dq_T} = 2q_T \frac{d\sigma}{dq_T^2}, \quad (4)$$

from the data we generated. We can contrast it to the divergent behaviour

$$\frac{1}{\sigma_0} \frac{d\sigma}{dq_T^2} = \frac{\alpha_s}{\pi} C_F \frac{\ln(Q^2 q_T^2)}{q_T^2} \quad (5)$$

of an NLO fixed-order prediction (dashed lines).



Note that  $d\sigma/dq_T$  vanishes for  $q_T \rightarrow 0$ , however, the  $q_T^2$  behaviour we discussed in the previous section makes it a power-like suppression rather than an exponential one we would get from the naive leading behaviour. The resummation tames the divergent fixed-order behaviour and the turn-around point is also often called the “Sudakov peak”. We see that due to the larger colour charge in the gluon-induced Higgs process ( $C_A = 3 > \frac{4}{3} = C_F$ ), the location of the Sudakov peak is further to the right in that case.