Emulation of high multiplicity matrix elements for electron-positron and hadron-hadron colliders

Henry Truong

A Thesis presented for the degree of
Doctor of Philosophy

Institute for Particle Physics Phenomenology
Department of Physics
Durham University
United Kingdom

November 2022

Emulation of high multiplicity matrix elements for electron-positron and hadron-hadron colliders

Henry Truong

Submitted for the degree of Doctor of Philosophy

November 2022

Abstract: This is some abstract about this thesis.

Contents

List of Figures

List of Tables

Declaration

The work in this thesis is based on research carried out in the Department of Physics at Durham University. No part of this thesis has been submitted elsewhere for any degree or qualification.

Research presented in this thesis is based on joint work published in the articles:

- Chapter 5 is based on [5]: Daniel Maître and Henry Truong, *A factorisation-aware Matrix element emulator*, Journal of High Energy Physics, 2021 (11). 066.

- Chapter 6 is based on ...

- Chapter 7 is based on ...

Other research projects published during my studies but not included in this thesis:

- [6]: J. Aylett-Bullock, C. Cuesta-Lazaro, A. Quera-Bofarull, M. Icaza-Lizaola, A. Sedgewick, H. Truong, A. Curran, E. Elliott, T. Caulfield, K. Fong, I. Vernon, J. Williams, R. Bower and F. Krauss, *June: open-source individual-based epidemiology simulation*, (2021), R. Soc. open sci.8210506210506.

- [7]: I. Vernon, J. Owen, J. Aylett-Bullock, C. Cuesta-Lazaro, J. Frawley, A. Quera-Bofarull, A. Sedgewick, D. Shi, H. Truong, M. Turner, J. Walker, T. Caulfield, K. Fong and F. Krauss, *Bayesian emulation and history matching of JUNE*, (2022), Phil. Trans. R. Soc. A.3802022003920220039.

Acknowledgements

I would like to thank. . .

This a place for some profound quote which sums up your work, feelings or life.

— from *Some book* by A. Author

Dedicated to

Someone special

and

that other guy

Chapter 1

Introduction

The standard model of particle physics (SM) is our current best working theory that describes the phenomenon we observe, for example, at collider experiments. The large hadron collider (LHC) is a high energy proton-proton collider experiment that allows us to study the fundamental particles laid out in the standard model.

The interaction of these fundamental particles are described by scattering amplitudes. The standard model allows us to calculate these amplitudes, or matrix elements, from first principles. A key challenge in the past few decades has been to compute these matrix elements for increasing number of final state particles and for higher orders in perturbative expansions in fixed order calculations. Although there has been tremendous progress in this sector a complication that naturally arises is that the evaluation of these matrix elements becomes increasingly time-consuming due to their algebraic complexity.

The topic of this thesis is to apply modern machine learning techniques to reduce the time spent evaluating these complex matrix elements. Considering the physical properties of these matrix elements allows us to implement physics knowledge into these machine learning algorithms giving us higher levels of performance than would otherwise be achievable.

The structure of this thesis is as follows: in Chapter 1 I will recap the necessary concepts from the standard model, in particular, on quantum chromodynamics

(QCD), the sector which governs the behaviour of quarks and gluons. Furthermore, the relationship between theory and experiment will be elaborated on.

In Chapter 2 I will discuss the challenges that arise in fixed order perturbative calculations and the current methods that have been adopted in the community to deal with them. These methods also form the basis of our physics knowledge which we embed into the machine learning algorithm. This procedure is described in detail in Chapter 5 for electron-positron annihilation for tree-level matrix element emulation. A similar philosophy is applied in Chapter 6 where next-to-leading order QCD k-factors are emulated. These two chapters discuss in detail the procedure for electron-positron annihilation, however, the most relevant experiment currently is the LHC which is colliding protons. The extension to hadron-hadron colliders is detailed in Chapter 7 where we explore using the emulator in a novel implementation to replace traditional matrix element providers in the event generator SHERPA.

Finally, I will conclude the thesis in Chapter 8.

1.1 The Standard Model of particle physics

The Standard Model of particle physics (SM) is our current best working theory to describe all known elementary particles as well as three of the four fundamental forces. Developed predominantly in the latter half of the 20th century, it is one of the most well tested theories we have in science today. Some highlights include the highly precise predictions of the anomalous magnetic moment of the electron, agreeing with experimental measurements to more than 10 significant figures [8], and the discovery of the Higgs boson in 2012 by the ATLAS [9] and CMS experiments [10] at the Large Hadron Collider (LHC) which was theorised decades prior.

The SM is a gauge quantum field theory (QFT) where particles are described as excitations in quantum fields. It can be specified by the gauge group

$$\mathrm{SU(3)}_c \times \mathrm{SU(2)}_L \times \mathrm{U(1)}_Y \,, \tag{1.1.1}$$

where subscripts denote the charges of the gauge groups. The first gauge group with colour charge c describes the interactions of the strong force within the theory of quantum chromodynamics (QCD), which we will elaborate more on in Section 1.2. The second and third gauge groups represent the electroweak sector of the SM with charges L for left and Y for hypercharge. Under electroweak spontaneous symmetry breaking (EWSB), this product becomes

$$\text{SU(2)}_L \times \text{U(1)}_Y \xrightarrow{\text{EWSB}} \text{U(1)}_{\text{EM}} , \tag{1.1.2}$$

giving rise to the electromagnetic and weak forces we observe. The effect of gravity is considered to be negligible on the scales considered in the SM and so it is not described. Each of the three fundamental forces described by the SM is mediated by the exchange of a gauge boson. The gauge bosons mediating the strong and EM force are the massless gluon g and massless photon γ, respectively. For the weak force the gauge bosons are the $W^\pm$ and Z^0 bosons which attain a mass through the Higgs mechanism [11–13] during EWSB, which elucidates the Higgs boson H.

The matter content of the SM consists of fermions which can be split into quarks and leptons. Quarks are massive and experience the strong, weak, and EM force. Leptons are defined by their lack of colour charge, meaning they do not experience the strong force. Leptons can be separated into charged leptons (electron e, muon μ, tau τ and their antiparticles) which experience the weak and EM force, and neutrinos (electron neutrino ν_e, muon neutrino ν_μ, tau neutrino ν_τ and their antiparticles) which only experience the weak force. Neutrinos are massless in the SM but this is not a requirement of the model. In fact they have been observed to have mass [14,15].

The particle content of the SM is summarised in Figure 1.1 showing the quarks, leptons and bosons along with their masses, charges and spin. The interactions of these fields are governed by the SM Lagrangian [1] which can be written as

$$\mathcal{L}_{\text{SM}} = \mathcal{L}_{\text{gauge}} + \mathcal{L}_{\text{fermion}} + \mathcal{L}_{\text{Higgs}} + \mathcal{L}_{\text{Yukawa}} + \mathcal{L}_{\text{GF}} + \mathcal{L}_{\text{ghost}} , \tag{1.1.3}$$

---

[1]technically Lagrangian but we use the terms Lagrangian and Lagrangian density interchangeably

Standard Model of Elementary Particles

three generations of matter (fermions)			interactions / force carriers (bosons)	
I	II	III		

mass $\simeq$2.2 MeV/c² charge ⅔ spin ½ **u** up	$\simeq$1.28 GeV/c² ⅔ ½ **c** charm	$\simeq$173.1 GeV/c² ⅔ ½ **t** top	0 0 1 **g** gluon	$\simeq$124.97 GeV/c² 0 0 **H** higgs
$\simeq$4.7 MeV/c² −⅓ ½ **d** down	$\simeq$96 MeV/c² −⅓ ½ **s** strange	$\simeq$4.18 GeV/c² −⅓ ½ **b** bottom	0 0 1 **γ** photon	
$\simeq$0.511 MeV/c² −1 ½ **e** electron	$\simeq$105.66 MeV/c² −1 ½ **μ** muon	$\simeq$1.7768 GeV/c² −1 ½ **τ** tau	$\simeq$91.19 GeV/c² 0 1 **Z** Z boson	
<1.0 eV/c² 0 ½ **ν$_e$** electron neutrino	<0.17 MeV/c² 0 ½ **ν$_\mu$** muon neutrino	<18.2 MeV/c² 0 ½ **ν$_\tau$** tau neutrino	$\simeq$80.433 GeV/c² ±1 1 **W** W boson	

QUARKS   **LEPTONS**   **GAUGE BOSONS** VECTOR BOSONS   **SCALAR BOSONS**

Figure 1.1: Particle content of the SM split into quarks, leptons, gauge bosons and scalar bosons. The columns for the fermions depict the three different generations. Masses, electric charge, and spin are given for each particle. The yellow contours indicate the coupling of bosons to fermions, illustrating the forces experienced by the fermions inside the contour. Figure reference [1].

where $\mathcal{L}_{\text{gauge}}$ describes the gauge fields, $\mathcal{L}_{\text{fermion}}$ describes how fermions interact with gauge fields as well as their kinetic terms, $\mathcal{L}_{\text{Higgs}}$ describes the Higgs field, $\mathcal{L}_{\text{Yukawa}}$ describes the interaction between the Higgs field and fermions, $\mathcal{L}_{\text{GF}}$ is a gauge fixing term, and $\mathcal{L}_{\text{ghost}}$ is a ghost term. The last two terms are required to remove unphysical degrees of freedom when gauge fixing the theory. All terms in the SM Lagrangian are invariant under local transformations of the gauge group (1.1.1).

In the following section we will focus on the gauge, fermion, gauge-fixing and ghost Lagrangian terms in the framework of QCD as that will be the most relevant sector for this thesis. The remaining terms are discussed at length in standard reference texts [16–18].

1.2 Introduction to quantum chromodynamics

1.2.1 The QCD Lagrangian

Quantum chromodynamics is the sector of the SM that describes the strong interaction. QCD is a non-Abelian gauge theory with gauge group $\text{SU}(N_c = 3)_c$ where the charge is named colour. The gauge and fermion part of the QCD Lagrangian is

$$\mathcal{L}_{\text{QCD}} = -\frac{1}{4} F_{\mu\nu}^a F^{a,\,\mu\nu} + \sum_f \bar{\psi}_i^f (\text{i}\slashed{D}_{ij} - \delta_{ij} m_f) \psi_j^f \,, \tag{1.2.1}$$

where twice repeated indices are summed over. The fields ψ_i^f ($\bar{\psi}_i^f$) are the fermions (antifermions), representing quarks (antiquarks) with flavours f: up u, down d, strange s, charm c, top t, and bottom b, with masses m_f. They transform under the fundamental representation with indices $i, j \in \{1, 2, 3\}$, named colour indices. The gauge fields A_μ^a, corresponding to gluons, appear in the quark covariant derivative

$$(D_\mu)_{ij} = \delta_{ij} \partial_\mu - i g_s T_{ij}^a A_\mu^a \,, \tag{1.2.2}$$

and the gauge field strength tensor

$$F_{\mu\nu}^a = \partial_\mu A_\nu^a - \partial_\nu A_\mu^a + g_s f^{abc} A_\mu^b A_\nu^c \,. \tag{1.2.3}$$

Gauge fields transform under the adjoint representation, which is an $N_c^2 - 1$ dimensional representation, meaning the adjoint indices $a, b, c \in \{1, ..., 8\}$. T_{ij}^a are the group generators in a fundamental representation. In SU(3) it is common to write the group generators as

$$T_{ij}^a = \frac{1}{2}\lambda_{ij}^a \tag{1.2.4}$$

where λ_{ij}^a are the Gell-Mann matrices [19]. The generators of the group have to obey the Lie algebra

$$[T^a, T^b] = \mathrm{i} f^{abc} T^c \,, \tag{1.2.5}$$

where f^{abc} are the structure constants of SU(3). The generators are normalised by convention to be

$$\mathrm{Tr}(T^a T^b) = \delta^{ab} T_R \,, \quad T_R = \frac{1}{2} \,. \tag{1.2.6}$$

This normalisation sets the values of the Casimirs of the group as

$$T_{ij}^a T_{jk}^a = \delta_{ik} C_F \,, \quad C_F = \frac{N_c^2 - 1}{2N_c} \,,$$
$$f^{abc} f^{abd} = \delta^{cd} C_A \,, \quad C_A = N_c \,, \tag{1.2.7}$$

where $C_F = 4/3$ and $C_A = 3$ for QCD. They are collectively referred to as colour factors.

The gauge coupling of the group g_s is a dimensionless free parameter of the theory. It is common to use the strong coupling constant instead,

$$\alpha_s = \frac{g_s^2}{4\pi} \,. \tag{1.2.8}$$

The strong coupling constant, is in fact not constant, but is dependent on the energy scale of the process. This is due to the process of renormalisation which will be discussed in Section 2.1.2. The implication of this is that at collider experiments where collisions occur at extremely high energy, the strong coupling constant becomes small, a property known as asymptotic freedom. A consequence of this is that predictions made in QCD can be expressed in the form of perturbative expansions in α_s.

To remove unphysical degrees of freedom from the theory we need to gauge-fix the theory and add a ghost Lagrangian. The gauge-fixing term in the R_ξ gauges is written as

$$\mathcal{L}_{\text{GF}} = -\frac{1}{2\xi}(\partial^\mu A_\mu^a)^2 \,, \tag{1.2.9}$$

where $\xi = 1$ corresponds to the Feynman-'t-Hooft gauge. The ghost Lagrangian is most commonly written in the Faddeev-Popov procedure as

$$\mathcal{L}_{\text{ghost}} = (\partial_\mu \bar{c}^a)(\delta^{ac}\partial_\mu + g_s f^{abc} A_\mu^b)c^c \,, \tag{1.2.10}$$

where c^a ($\bar{c}^a$) are Faddeev-Popov ghosts (anti-ghosts). They are unphysical external states but must be included in internal lines for a Lorentz invariant perturbative gauge theory.

1.2.2 QCD Feynman rules

Now that we have written down the QCD Lagrangian, we can examine the terms to write down how the gauge bosons interact with the fermions, and especially important for QCD, the gauge boson self interactions. A convenient way to do this is by deriving the Feynman rules of the theory, which can be pieced together to form Feynman diagrams. We will see that Feynman diagrams are a simple but powerful tool to systematically build up all the possible ways a process can occur.

Expanding (1.2.1) and extracting the terms that mix the gauge and fermions fields, we get an interaction Lagrangian

$$\mathcal{L}_{\text{int}} = g_s A_\mu^a \bar{\psi}_i^f \gamma^\mu T_{ij}^a \psi_j^f - g_s f^{abc}(\partial_\mu A_\nu^a)A^{b,\mu}A^{c,\nu} - \frac{g_s^2}{4}f^{abc}A_\mu^b A_\nu^c f^{ade}A^{d,\mu}A^{e,\nu} \,, \tag{1.2.11}$$

which we can interpret as follows: the first term is an interaction between a gluon, a quark, and an anti-quark, the second term is a triple gluon self-interaction, and the third term is a four gluon self-interaction. These mixing terms can be recast into Feynman rules representing interaction vertices.

To connect vertices we require propagators, which we can read off from the kinetic

Lagrangian where we collect terms from (1.2.1) and (1.2.9),

$$\mathcal{L}_{\text{kin}} = -\frac{1}{4}(\partial_\mu A_\nu^a - \partial_\nu A_\mu^a)^2 - \frac{1}{2\xi}(\partial_\mu A_\mu^a)^2 + \bar{\psi}_i^f(\mathrm{i}\slashed{\partial} - m_f)\psi_i^f \,, \qquad (1.2.12)$$

where the first two terms corresponds to a gluon propagator and the third term corresponds to a quark propagator.

From these Lagrangians, we can read off the QCD Feynman rules which we have collected in Figure 1.2. Note that we have only written down the vertices and propagators for the physical gluons and quarks, ghosts also have associated Feynman rules but we do not collect those here. A full list of Feynman rules in the SM can be found at Ref. [18].

From these Feynman rules, we identify that each quark-antiquark-gluon and triple gluon vertices are associated with one power of α_s (due to squaring g_s), and the four gluon vertex is associated with two powers of α_s. This relationship between vertices and α_s allows us to systematically build up Feynman diagrams that correspond to a fixed order in α_s. This point will be elaborated on in Section 1.4 where we discuss scattering amplitudes and how they are related to Feynman diagrams.

1.3   Factorisation theorem

At the LHC collisions occur between two protons, which have constituent quarks and gluons, collectively named partons. The application of QCD to describe phenomenon in these proton-proton collisions rests on the use of the factorisation theorem which enables the separation of low and high energy scales (so-called soft and hard). Within the proton, there are quarks and gluons that are constantly absorbed and emitted on a timescale inversely proportional to the mass of the proton. The timescale of this fluctuation is much longer than the timescale of the interaction between a parton and a highly energetic probing parton from another proton. This is the scenario that occurs at collisions at the LHC. During the collision, or the hard scattering, the probe is able to interact with a parton that is effectively frozen. The probing parton

$$\mu, a \underset{\xleftarrow{\hspace{0.5cm} p \hspace{0.5cm}}}{\mathrm{\longleftarrow \, gluon \, \longleftarrow}} \nu, b \;=\; -\mathrm{i}\delta_{ab}\left[\frac{g^{\mu\nu}}{p^2+i\epsilon}-(1-\xi)\frac{p^\mu p^\nu}{(p^2)^2}\right]$$

$$\xrightarrow{\hspace{1cm} p \hspace{1cm}} \;=\; \frac{\mathrm{i}(\slashed{p}+m_f)}{p^2-m_f^2+i\epsilon}$$

$$= -\mathrm{i}g_s\gamma^\mu T^a_{ij}$$

$$= -g_s f^{abc}\big[g^{\mu\nu}(p_1-p_2)^\rho$$
$$+g^{\nu\rho}(p_2-p_3)^\mu$$
$$+g^{\rho\mu}(p_3-p_1)^\nu\big]$$

$$= -\mathrm{i}g_s^2\big[f^{eab}f^{ecd}(g^{\mu\rho}g^{\nu\sigma}-g^{\mu\sigma}g^{\nu\rho})$$
$$+f^{eac}f^{edb}(g^{\mu\sigma}g^{\rho\nu}-g^{\mu\nu}g^{\rho\sigma})$$
$$+f^{ead}f^{ebc}(g^{\mu\nu}g^{\rho\sigma}-g^{\mu\rho}g^{\nu\sigma})\big]$$

Figure 1.2: QCD Feynman rules with the Minkowski metric signature $g^{\mu\nu}=\mathrm{diag}(1,-1,-1,-1)$.

knows nothing of the proton being probed, except for the fact that it collided with a parton carrying a fraction of the proton momentum. The distribution of partons within a proton is process independent and a fundamental property of the proton, meaning it can be separated from the actual scattering between the partons.

This heuristic argument motivates the form of the factorisation equation, where the production of particles from a hadronic collision, the hadronic cross-section, can be written as

$$
\sigma_{AB \to n} = \sum_{a,b} \int_0^1 \mathrm{d}x_a \mathrm{d}x_b \; f_{a/A}(x_a, \mu_F) f_{b/B}(x_b, \mu_F) \; \hat{\sigma}_{ab \to n}(Q, \mu_F, \mu_R) + \mathcal{O}\left(\frac{\Lambda_{\mathrm{QCD}}}{Q}\right),
$$
(1.3.1)

where a is the parton from hadron A, and b is the parton from hadron B. The sum over initial-state partons a and b indicates the sum over flavours, which depends on the hadron composition in general. This factorisation is not exact as indicated by the correction term inversely proportional to characteristic hard scale Q. However, for high energy collisions, where $Q \gg \Lambda_{\mathrm{QCD}}$, this term and higher order terms are suppressed. In fact, the factorisation equation has only been proven for a few specific cases [20–23].

$f_{a/h}(x, \mu)$ are parton distribution functions (PDFs) which depend on the momenta fractions x parton a has with respect to its parent hadron h, at a scale μ, usually taken to be the factorisation scale μ_F, the interface between soft and hard physics. The interpretation of PDFs at leading order are as probability distributions. $f_{a/h}(x, \mu)$ is the probability of finding parton a within h carrying a momentum fraction x at the energy scale μ. PDFs are non-perturbative objects that encapsulate the soft effects of the scattering which occur below energy μ_F. They are non-perturbative because they are determined by fitting to experimental data, instead of calculated in perturbation theory.

For a review of PDFs and how they are determined see Refs. [24,25]. In practice, these PDF fits are accessed through the LHAPDF interface [26] which provides PDF sets from multiple working groups [27–30]. The evolution of PDFs between factorisation

scales is possible through the use of the Dokshitzer-Gribov-Lipatov-Altarelli-Parisi (DGLAP) equations [31–34].

$$\mu_F^2 \frac{\partial}{\partial \mu_F^2} \begin{pmatrix} f_{q/h}(x, \mu_F) \\ f_{g/h}(x, \mu_F) \end{pmatrix} = \frac{\alpha_s(\mu_F)}{2\pi} \int_x^1 \frac{\mathrm{d}z}{z} \begin{pmatrix} P_{qq}(\frac{x}{z}) & P_{qg}(\frac{x}{z}) \\ P_{gq}(\frac{x}{z}) & P_{gg}(\frac{x}{z}) \end{pmatrix} \begin{pmatrix} f_{q/h}(x, \mu_F) \\ f_{g/h}(x, \mu_F) \end{pmatrix}, \quad (1.3.2)$$

where $P_{ba}(\frac{x}{z})$ are splitting functions representing parton b emitting a parton a carrying a momentum fraction x. These splitting functions are calculable as a power series in α_s where they have been computed up to three-loops [35, 36]. At leading order [34] they are

$$P_{qq}^{(0)}(z) = C_F \left[\frac{1 + z^2}{(1 - z)_+} + \frac{3}{2}\delta(1 - z) \right] \quad (1.3.3)$$

$$P_{qg}^{(0)}(z) = T_R \left[z^2 + (1 - z)^2 \right] \quad (1.3.4)$$

$$P_{gq}^{(0)}(z) = C_F \left[\frac{1 + (1 - z)^2}{z^2} \right] \quad (1.3.5)$$

$$P_{gg}^{(0)}(z) = 2C_A \left[\frac{z}{(1 - z)_+} + \frac{1 - z}{z} + z(1 - z) \right] + \left(\frac{11C_A - 4T_R n_f}{6} \right) \delta(1 - z) \quad (1.3.6)$$

where the '+'-prescription has been used to regulate divergences at $z = 1$.

$$\int_0^1 \mathrm{d}z \, [g(z)]_+ f(z) = \int_0^1 \mathrm{d}z \, g(z)(f(z) - f(1)), \quad (1.3.7)$$

where the function $f(z)$ cancels the divergence if it is sufficiently smooth at $z = 1$. At leading order, there is a physical interpretation of the splitting kernels as the probability of finding a parton a in parton b carrying a fraction x of the parent parton.

The momenta fractions, x, also link the squared centre-of-mass energies of the hadronic collision, denoted as s, to the partonic equivalent as

$$\hat{s} = x_a x_b s. \quad (1.3.8)$$

$\hat{\sigma}_{ab \to n}(Q, \mu_F, \mu_R)$ is the partonic cross-section which descibes the interaction of partons a and b scattering into n particles, where the scattering occurs at energy scale Q, which is often taken to be $\hat{s}$. Notice that the partonic cross-section depends on

both the factorisation scale, μ_F and the renormalisation scale, μ_R (see Section 2.1.2). This will be discussed in Section 3.4 in the context of theoretical uncertainties.

The discussion so far has been focused hadron-hadron initiated scattering, however, a similar argument can be made about electron-positron annihilation, where the electron-positron annihilates to form a photon (Z boson) which fluctuates into a quark-antiquark pair. For a sufficiently high energy collision, these interactions can be calculated in perturbation theory. Since electrons and positrons are not composite particles, there is no requirement for PDFs, therefore (1.3.1) simplifies to just the partonic cross-section, which will be computed in perturbation theory.

The details of how $\hat{\sigma}_{ab \to n}$ is computed is discussed in the next section.

1.4 Scattering amplitudes and cross-sections

As mentioned in the previous section, at collider experiments we typically collide two beams, consisting of bundles of energetic particles, and analyse the resulting products of these collisions. The predictions we make from our theoretical model are the partonic cross-sections. To arrive at an expression for the partonic cross-section, we need to first consider the hard scattering of particles. To model the hard collisions in a collider experiment, we look at the specific case of two particles colliding into n particles. Scattering amplitudes are used to mathematically describe these scattering processes. For an initial-state $|i\rangle$ and final-state $|f\rangle$ the scattering amplitude can be written as

$$\langle f|S|i\rangle , \tag{1.4.1}$$

where the scattering matrix, or S-matrix can be decomposed into an identity matrix and a transfer matrix $\mathcal{T}$,

$$S = \mathbb{1} + \mathrm{i}\mathcal{T} . \tag{1.4.2}$$

The S-matrix encodes all the information about how the initial-state will evolve into the final-state over time. By writing the S-matrix in this way, all the interactions

are separated into the transfer matrix $\mathcal{T}$, as the identity matrix describes the free theory. By imposing a momentum conserving constraint on the S-matrix, the matrix element, $\mathcal{M}$, is defined in the following expression

$$\langle f|\mathcal{T}|i\rangle = (2\pi)^4 \delta^4 \left(p_a + p_b - \sum_{f=1}^{n} p_f^\mu \right) \mathcal{M} \,, \tag{1.4.3}$$

where we take p_a and p_b to be the momenta of the two colliding particles in the initial-state, and p_f to be the momenta of the n-particle final states. The transition probability of this scattering process is the modulus squared of the scattering amplitude

$$P = |\langle f|S|i\rangle|^2 \propto |\langle f|\mathcal{M}|i\rangle|^2 \tag{1.4.4}$$

where $|\langle f|\mathcal{M}|i\rangle|^2 \equiv |\mathcal{M}|^2$ is the matrix element squared [2].

Since detectors at collider experiments have limited resolution, single scattering events are not measured. Instead we consider the cross-section, σ, as introduced in (1.3.1). The cross-section is a property of the particles being scattered, and is independent of the way the experiment is carried out. Since the hadronic cross-section factorises the soft energy effects into the non-perturbative PDFs, which are determined once and for all processes, the object of interest now becomes the partonic cross-section, $\hat{\sigma}$, which encodes the scattering information of the specific process considered.

Given that cross-sections are measurable quantities, the partonic cross-section directly relates the matrix elements which we calculate in our QFT, to measurable observables at experiments. In practice, it is more useful to consider the differential cross-section, $\mathrm{d}\hat{\sigma}$, where the cross-section can be differential in quantities such as energies and angles. This is because the final-state particles will be produced with a range of, for example, energies and momenta so the cross-section would be vanishing for an exact momentum configuration.

---

[2]from henceforth we refer to the matrix element squared as matrix element

The differential partonic cross-section can be written as

$$\mathrm{d}\hat{\sigma} = \frac{1}{\mathcal{F}}|\mathcal{M}_{2\to n}|^2 \mathrm{d}\Phi_n \,. \tag{1.4.5}$$

$\mathcal{F}$, the flux factor is

$$\mathcal{F} = 4\sqrt{(p_a p_b)^2 - m_a^2 m_b^2} \,, \tag{1.4.6}$$

which in the massless limit reduces to

$$\mathcal{F} = 2(p_a + p_b)^2 = 2\hat{s} \,. \tag{1.4.7}$$

It is common to use the massless limit because the centre-of-mass energy of a collider experiment is much greater than the mass of the colliding particles, and so they can be neglected.

$\mathrm{d}\Phi_n$ is the Lorentz-invariant phase-space which contains all the possible configurations of the n-particle final-state. Absorbing the momentum conserving δ-function from (1.4.9), we can write it as

$$\mathrm{d}\Phi_n = (2\pi)^4 \delta^4\left(p_a + p_b - \sum_{f=1}^{n} p_f^\mu\right) \prod_{f=1}^{n} \frac{\mathrm{d}^4 p_f}{(2\pi)^3} \delta(p_f^2 - m_f^2)\theta(E_f)\,, \tag{1.4.8}$$

where the second δ-function restricts the final-state particles to be on mass-shell (on-shell). The step function selects only the positive energy solution. This gives an intuitive picture of the partonic cross-section as the probability of a $2 \to n$ process occurring, summed over all the possible valid final-state configurations.

From the definition of the transition probability (1.4.4), it becomes clear that the matrix element is the object which relates back to the Lagrangian of the theory as it encodes the interactions of the particles. We saw in Section 1.2.2 that interactions could be codified into Feynman rules which are pieced together to construct Feynman diagrams. In this picture, matrix elements are exactly the sum over all Feynman diagrams [3] for a particular process. However, there are an infinite number of Feynman diagrams you can draw as there is no constraints on how the vertices and propagators

---

[3]Since we are working on the level of $|\mathcal{M}|^2$, it would technically be the sum of all interference of Feynman diagrams.

can be connected, as long as the diagram contains the desired initial and final-states. Fortunately because α_s is small in high energy collisions, we can expand the matrix element as a perturbative series in α_s to write

$$|\mathcal{M}|^2 = \alpha_s^m |\mathcal{M}|^2_{\text{LO}} + \alpha_s^{m+1} |\mathcal{M}|^2_{\text{NLO}} + \alpha_s^{m+2} |\mathcal{M}|^2_{\text{NNLO}} + \mathcal{O}(\alpha_s^{m+3}), \qquad (1.4.9)$$

where m corresponds to the powers of α_s in the simplest Feynman diagrams you can draw for the process of interest. Since each term of this expansion is at a fixed-order in α_s, it is possible to systematically compute the diagrams which contribute at the given order of α_s. The set of diagrams that contribute at the lowest order in α_s are $|\mathcal{M}|^2_{\text{LO}}$, where LO stands for leading order. The next term in the expansion corresponds to the leading order diagrams with an additional loop, or external leg, which contributes an extra factor of α_s. This term is dubbed NLO for next-to-leading order in α_s, and the following term next-to-next-to-leading order has again an additional loop or leg. In general, we have terms $\text{N}^{\text{k}}\text{LO}$ where each additional power of α_s increases complexity of the computations, however, because the expansion α_s is small, each additional term should contribute less and less. This means that the first terms dominate the expansion, and it is sufficient to terminate the series after a few terms to reach an acceptable level of accuracy. A further discussion of this point will be given in Chapter 2 where we break down the computation of each term in (1.4.9).

In summary, the partonic cross-section of a collision reduces to the computation of the matrix elements up to a fixed-order in the coupling parameter, which is then integrated over the valid phase-space of final-state particle configurations. For hadronic collisions the partonic cross-section also needs to be convoluted with the PDFs to obtain the hadronic cross-sections.

With the introduction of matrix elements and cross-sections complete, we will move the discussion onto more practical aspects of computations within the QCD framework. More specifically, we will discuss the divergent structure of matrix elements and the machinery developed to tackle these unphysical singularities.

Chapter 2

Quantum chromodynamics in practice

In the previous chapter we introduced the basic concepts of QCD and setup the framework of calculating cross-sections. Cross-sections are the most relevant quantities to compute as they can be measured experimentally, and are a property of the particles being collided, rather than depending on the specifics of the experimental procedure. This provides a bridge to compare theoretical predictions and experimental measurements. The relevant quantity calculated in perturbation theory is the partonic cross-section which is the matrix element integrated over the relevant final-state phase-space, normalised by a flux factor. The matrix elements are calculated order-by-order in the strong coupling α_s (c.f. 1.4.9), where each order corresponds to a set of Feynman diagrams that have the appropriate number of vertices, as each QCD vertex brings along a factor of α_s.

In this chapter we will discuss the problems that arise when evaluating these Feynman diagrams and outline some solutions that have been widely adopted to circumvent these issues in order to provide real-world applicable predictions. This chapter forms the theoretical foundations upon which this thesis is built on.

We will consider the divergent nature of matrix elements and introduce the most widely used techniques to tame these singularities.

2.1   Divergent structures

The computation of matrix elements essentially reduces down to evaluating Feynman diagrams which are analytical expressions built up from the Feynman rules of a theory (c.f. QCD Feynman rules 1.2). In evaluating certain topologies of Feynman diagrams, integral expressions containing unconstrained momenta will give rise to singularities. The divergences associated with high energy modes are ultraviolet (UV) divergences and on the opposite end of the energy spectrum, low energy modes give rise to infrared (IR) divergences.

The de-facto methods to alleviate these divergences are through regularisation for UV divergences, and through subtraction schemes for IR divergences. Both of these methods will be discussed in this section.

2.1.1   Ultraviolet divergences

During intermediate steps of calculations, such as the compututation of loop diagrams seen in Figure 2.1, we have to evaluate integrals of the form

$$I_{\mathrm{UV}} = \int_0^\Lambda \frac{\mathrm{d}^4\ell}{(2\pi)^4} \frac{1}{\ell^2(\ell+p)^2} \sim \log\Lambda\,, \tag{2.1.1}$$

where a cutoff scale Λ has been introduced to capture the divergence as the loop momenta $\ell \to \infty$. It is clear that the integral diverges in this high energy limit, hence the name ultraviolet divergence. This UV divergence cannot influence the experimentally observable quantities at experiments, which are finite, because the final observable should not depend on the intermediate steps taken to get there. Infinities associated with UV divergences are removed via the process of renormalisation which introduces counterterms in the Lagrangian to exactly cancel these divergences.

Before we introduce these counterterms, we first consider the procedure of regularisation which makes these infinities explicitly manifest. The most widely adopted method of regularisation is dimensional regularisation (DR) [37] which is based on
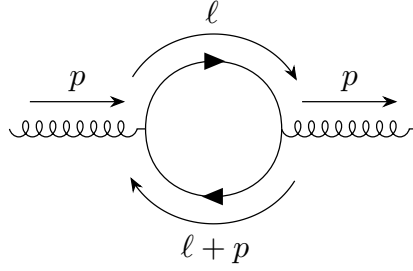
Figure 2.1: Massless bubble diagram which is an example of where unconstrained loop momenta can lead to UV divergences.

the observation that the integral (2.1.1), which is carried out in $d = 4$ space-time dimensions, would be finite if $d < 4$.

In DR we define $d = 4 - 2\epsilon$ such that after Feynman parametrisation (2.1.1) becomes

$$I_{\mathrm{UV}} = \int \frac{\mathrm{d}^d \ell}{(2\pi)^d} \frac{1}{\ell^2 (\ell + p)^2} = \frac{\mathrm{i}}{(4\pi)^2} \left(\frac{-p^2}{4\pi} \right)^{-\epsilon} \frac{\Gamma(1-\epsilon)^2}{\Gamma(2-2\epsilon)} \Gamma(\epsilon) \,, \qquad (2.1.2)$$

where the divergence for $d = 4$, or equivalently $\epsilon \to 0$, is now captured in the Gamma function, $\Gamma(\epsilon)$. This becomes clear once we expand $\Gamma(\epsilon)$ around small ϵ

$$\Gamma(\epsilon) = \frac{1}{\epsilon} - \gamma_E + \frac{1}{2} \left(\gamma_E^2 + \frac{\pi^2}{6} \right) \epsilon + \mathcal{O}(\epsilon^2) \,, \qquad (2.1.3)$$

where $\gamma_E \approx 0.577$ is the Euler-Mascheroni constant, meaning the divergence is now regularised as a pole in ϵ.

2.1.2 Renormalisation

With the UV divergence regularised by dimensional regularisation, it is now possible to construct the counterterms order-by-order in α_s to explicitly cancel the divergences.

A subtlety with adjusting the dimension of the theory from $d = 4$ to arbitrary dimensions is that the mass dimension of the Lagrangian has to change accordingly to retain a dimensionless action in natural units. We can account for this by introducing an arbitrary energy scale μ_R, the renormalisation scale, into the gauge coupling

constant to modify the mass dimension of the Lagrangian. For QCD this is done via the modification

$$g_s \to g_s \mu_R^{\epsilon} \,. \tag{2.1.4}$$

where it is understood that the original gauge coupling is dimensionless. The order of μ_R is determined by examining the dimensions of the gauge and fermion fields in the Lagrangian (1.2.1). We see that this introduces a scale dependence on the coupling constant α_s, a point we will return to in Section 2.1.3.

The inclusion of counterterms into the theory can be thought of as renormalisations of the bare fields and couplings in the QCD Lagrangian (1.2.1) to restore predictive power to the theory. The renormalised fields and couplings can be given as (see for instance Ref. [38])

$$\psi_{\text{bare}} = \sqrt{Z_2} \psi_R \,,$$
$$A_{\text{bare}}^{\mu} = \sqrt{Z_3} A_R^{\mu} \,, \tag{2.1.5}$$
$$g_{s,\text{bare}} = Z_g \mu_R^{\epsilon} g_{s,R} \,,$$

where it conventional to define $Z_1 = Z_g Z_2 \sqrt{Z_3}$ such that we can set $Z_n = 1 + \delta_n$ for $n \in \{1, 2, 3\}$. In this way we can write the Lagrangian as

$$\mathcal{L}_R = \mathcal{L}_{\text{bare}} + \mathcal{L}_{\text{c.t.}} \,, \tag{2.1.6}$$

where the counterterms appearing in $\mathcal{L}_{\text{c.t.}}$ are determined by calculating δ_n. There is freedom in the choice of δ_n which gives rise to different regularisation schemes. The minimal subtraction (MS) scheme subtracts only the epsilon pole appearing in loop integrals. However, the most commonly used regularisation scheme is the modified minimal subtraction ($\overline{\text{MS}}$) scheme which subtracts the epsilon pole plus a universal constant appearing in all loop integrals.

In an all orders calculation of an observable, there would be no dependence on the renormalisation scale as it is a remnant of the regularisation prescription However, because in perturbative QCD observables are calculated order-by-order, there will be

a residual dependence on the renormalisation scale stemming from the missing higher order terms. The conventional way that the community estimates the uncertainty associated with this residual dependence is by carrying out a scale variation (see Section **??**).

2.1.3   Running of the coupling constant

The renormalisation scale introduced during regularisation is a mathematical artifact and should not impact any measurable quantities. Therefore, the bare coupling should not depend on the renormalisation scale,

$$\frac{\mathrm{d}g_{s,\mathrm{bare}}}{\mathrm{d}\mu_R} = 0\,, \tag{2.1.7}$$

or said another way, the renormalisation process is independent of the actual value of the renormalisation scale. The consequence of this is that the renormalised coupling, $\alpha_s(\mu_R)$, has to depend on the renormalisation scale instead.

The renormalisation scale dependence of α_s is governed by the Callan-Symanzik [39, 40] β-function

$$\mu_R^2 \frac{\partial \alpha_s(\mu_R^2)}{\partial \mu_R^2} = \beta(\alpha_s)\,, \tag{2.1.8}$$

where the β-function can be written as a perturbative expansion in α_s

$$-\beta(\alpha_s) = \alpha_s \sum_{n=0}^{\infty} \left(\frac{\alpha_s}{4\pi}\right)^{n+1} \beta_n \tag{2.1.9}$$

where the coefficients β_n have been computed up to β_4 [41, 42]. The solution to (2.1.8) to first order is

$$\alpha_s(\mu_R^2) = \frac{1}{\frac{\beta_0}{4\pi} \log\left(\frac{\mu_R^2}{\Lambda_{\mathrm{QCD}}^2}\right)}\,, \tag{2.1.10}$$

where Λ_{QCD} is the QCD scale, the scale at which α_s becomes large enough that perturbation theory breaks down, and $\beta_0 = (11C_A - 4T_R n_f)/3$. In QCD, $C_A = 3$ and $T_R = \frac{1}{2}$, meaning for $n_f < \frac{33}{2}$ the sign of β_0 is positive. In nature we have observed six flavours of quarks, so $\beta_0 > 0$. In fact all β-coefficients computed to date are positive, meaning that α_s decreases with increasing energy due to the minus sign
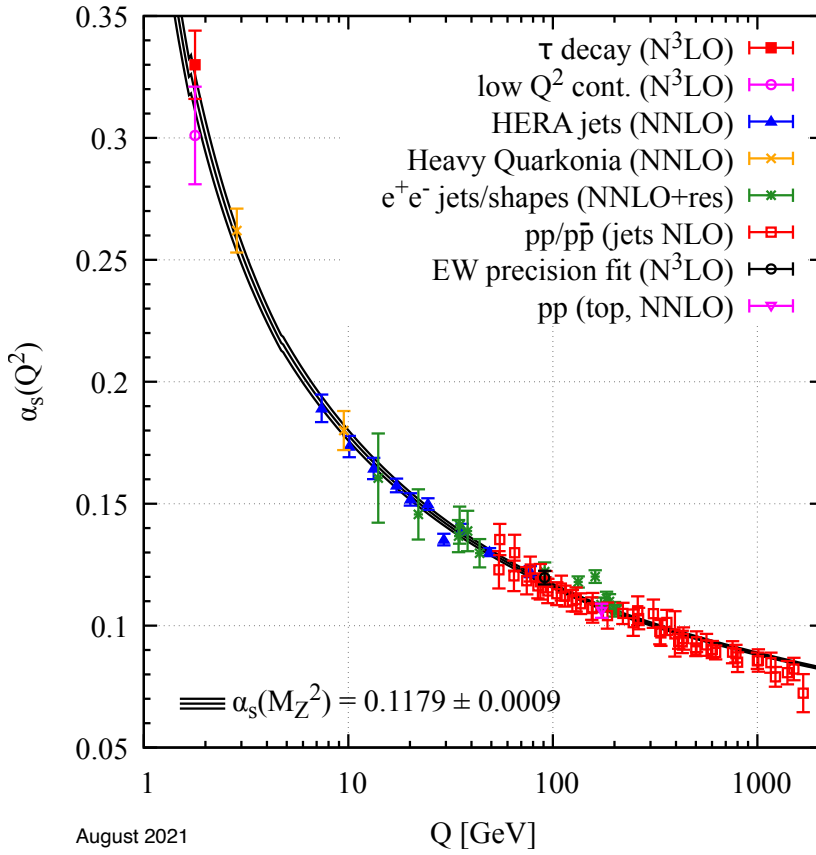
Figure 2.2: The running of the strong coupling constant α_s, as determined by experiments, with QCD theory prediction in black. Figure reference [2].

in (2.1.9). This property is known as asymptotic freedom [43, 44], and is justification for treating QCD perturbatively when the energy scale is high, such as at collider experiments. The running of the coupling constant has been observed experimentally as illustrated in Figure 2.2.

Another consequence of the running coupling is colour confinement, in which quarks and gluons cannot exist as free particles. Due to the increasing strength of coupling at low energies, quarks and gluons are forced to form composite, colourless particles. This means that at low energies, perturbative QCD is not an accurate description of nature, and the behaviour of hadrons cannot be predicted from perturbation theory. Instead phenomenological models have to be used to describe these processes.

2.1.4   Infrared divergences

In Section 2.1.1 we saw how divergences stemming from high energy behaviour would occur in loop diagrams. On the other end of the energy spectrum we also have divergences from low energy modes. Since these divergences emerge from low energy behaviour they are called infrared (IR) divergences. There are two cases in which these divergences arise:

- virtual divergences in loop integrals,

- real emission divergences when an emission of an extra particle has vanishing energy (soft), or becomes parallel to an external leg (collinear).

Virtual IR divergences can be understood by looking at an integral of the form

$$I_V = \int \frac{\mathrm{d}^4 \ell}{(2\pi)^4} \frac{1}{\ell^2 (\ell + p_1)^2 (\ell - p_2)^2} \,, \tag{2.1.11}$$

which is encountered when calculating the one-loop virtual correction to the gluon-quark-antiquark vertex as shown in Figure 2.3. It is clear that the denominator vanishes when $\ell \to 0$ or when either $(\ell + p_1)^2$ or $(\ell - p_2)^2 \to 0$. These situations correspond to the gluon in the loop propagator going soft, or collinear to the external quark or antiquark. These divergences are regulated in DR to give

$$I_V = \frac{\mathrm{i}}{(4\pi)^2 Q^2} \left(\frac{-Q^2}{4\pi} \right)^{-\epsilon} \frac{\Gamma(1 + \epsilon)\Gamma(1 - \epsilon)^2}{\Gamma(1 - 2\epsilon)} \left[\frac{1}{\epsilon^2} \right] \,, \tag{2.1.12}$$

where $Q = (p_1 + p_2)$. There is a double ϵ pole manifest, corresponding to the soft and collinear divergence.

Real IR divergences are manifest when integrating matrix elements over the phase-space of external state momenta. Matrix elements can be written as functions of Mandelstam variables

$$s_{ij} = (p_i + p_j)^2 \xrightarrow[\text{limit}]{\text{massless}} 2 E_i E_j (1 - \cos \theta_{ij}) \,, \tag{2.1.13}$$

where p_i and p_j are the 4-momenta of partons i and j in the hard scattering. When either of the partons beecome soft, $E_{i,j} \to 0$, or when they go collinear to each other,
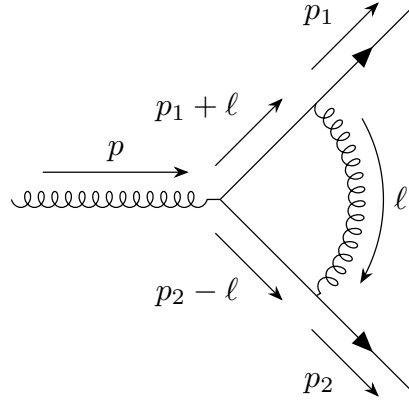
Figure 2.3: One-loop vertex correction for the gluon-quark-antiquark vertex which diverges for $\ell \to 0$.

$\theta_{ij} \to 0$, the matrix element will diverge if s_{ij} appears in the demoninator. These scenarios correspond to an external particle becoming unresolved.

In Section 2.1.2, we removed UV divergences through the use of counterterms to renormalise the theory, making predictions from the theory physical. IR divergences, on the other hand, arise even after the theory has been renormalised. The resolution to IR divergences is given by the Block-Nordsieck [45,46] and Kinoshita-Lee-Nauenberg (KLN) theorems [47,48] which states that at each order of perturbation theory, the virtual and real contributions exactly cancel out, leaving only finite corrections. This cancellation occurs because the infrared pole structure is identical in the real and virtual corrections, but with an opposite sign.

At a detector of a collider experiment, there is a finite energy resolution of the calorimeter. This means that it is not physically possible to observe arbitrarily soft partons. It is also not possible to distinguish two partons at arbitrarily small angles from one parton with the combined momenta. These limitations along with the KLN theorem means that any physical observable that we wish to predict with fixed-order perturbation theory must be insensitive to the emissions of soft or collinear partons. Any observable obeying this criteria is an infrared and collinear (IRC) safe observable. This leads to the concept of jets [49] which are collimated particles combined according to some jet algorithm.

In DR, the singularities from soft and collinear divergences are manifest as ϵ poles which makes the cancellation simple between the real and virtual parts. However, in practice, the phase-space integrals of the matrix elements are rarely carried out analytically due to the high dimensionality of the integral, rendering them intractable. Instead they are done numerically through the use of Monte Carlo methods (see Chapter 3), which requires the integral to be in integer dimensions. Therefore, numerical techniques have been developed to deal with these IR divergences as well. The most common technique, subtraction, will be discussed in more detail in Section 2.3.

2.2 Factorisation of matrix elements

In the previous section we examined the IR divergences that can arise in matrix elements, either explicitly through loop diagrams, or implicitly when carrying out the phase-space integral over the external state momenta. In this section we will discuss how these IR singularities are universal and how the divergences associated with real emissions can be factorised out of the matrix element. This property is exploited extensively in Chapters 5, 6, and 7 where the research of this thesis is presented.

The regions of phase-space in which real emission matrix elements diverge are when the emission is soft and/or collinear. It can be shown that in these regions of phase-space, the matrix element factorises into a process-independent singular factor, multiplied by a matrix element with the unresolved parton being absorbed by an emitting leg – we call this the reduced matrix element. This factorisation, however, is not exact in QCD where there are spin and colour correlations.

Schematically, the $(n + 1)$-body matrix element factorises as

$$|\mathcal{M}_{n+1}|^2 \rightarrow \boldsymbol{S}_{ijk} \otimes |\mathcal{M}_n|^2 , \qquad (2.2.1)$$

where $\boldsymbol{S}_{ijk}$ is a universal singular factor capturing the IR divergent behaviour and

$|\mathcal{M}_n|^2$ is the reduced matrix element. The $\otimes$ represents the colour and spin correlations existing between the singular function and the reduced matrix element. This singular factor is not unique and can be represented by different approximations as long as they reproduce the correct IR behaviour. In this thesis we discuss two approximations: Catani-Seymour dipoles [50] in Section 2.4 and antenna functions [51] in Section 2.5. The indices of $\boldsymbol{S}_{ijk}$ already hints at the fact that the singular functions only depends on three partons, and not all of the external states.

We will inspect the soft and collinear limits separately to see how the matrix element factorises in these respective limits. The following section will follow the conventions of Ref. [50], namely that the colour and helicity summed n-body matrix element written as

$$|\mathcal{M}_n|^2 = {}_n\langle 1, \ldots, n | 1, \ldots, n \rangle_n \tag{2.2.2}$$

where $|1, \ldots, n\rangle_n$ is a vector in colour + helicity space (see Appendix A for a more thorough explanation on notation).

2.2.1 Soft limits

Consider a tree-level matrix element $|\mathcal{M}_{n+1}|^2$ with a final-state gluon j. Note that reduced matrix elements associated with taking a quark soft have to vanish due to violation of quark number, therefore only gluons are considered. The limit of the soft gluon with momentum p_j can be parametrised by

$$p_j^\mu = \lambda q^\mu, \quad \lambda \to 0, \tag{2.2.3}$$

where q^μ is an arbitrary four-vector and λ is a scale parameter. In this limit the matrix element can be written as

$$|\mathcal{M}_{n+1}|^2 \to -\frac{1}{\lambda^2} 8\pi\mu^{2\epsilon}\alpha_s \sum_i \frac{1}{p_i q} \sum_{k \neq i} \frac{p_k p_i}{(p_i + p_k)q} \, {}_n\langle 1, \ldots, n+1 | \boldsymbol{T}_k \cdot \boldsymbol{T}_i | 1, \ldots, n+1 \rangle_n,$$
$$\tag{2.2.4}$$

where terms less singular than $1/\lambda^2$ have been neglected. The reduced matrix element represented by the colour + helicity vector inner product is obtained by removing

the soft gluon from the $(n + 1)$-body matrix element. The indices i, j, and k label partons involved in the factorsation process: i is the emitter parton, j is the emitted parton, and k is a parton accounting for the colour correlations. The scale μ can be identified as the renormalisation scale, and $\boldsymbol{T_i}$ is the colour-charge operator. It acts on the colour vectors in the reduced matrix element to give factors of C_A and C_F, hence this factorisation is not exact.

The matrix elements in (2.2.4) are unambiguously defined only when momentum conservation is fulfilled. This is only true in the strict $\lambda = 0$ limit. Away from the limit, care has to be taken to conserve momentum conservation and keep the relevant partons on-shell. This can be done through the use of momentum mappings [50, 52].

2.2.2 Collinear limits

For the collinear limit, consider partons i and j in the matrix element $|\mathcal{M}_{n+1}|^2$. Their momenta can be decomposed as

$$
\begin{aligned}
p_i^\mu &= z p^\mu + k_\perp^\mu - \frac{k_\perp^2}{z} \frac{n^\mu}{2p \cdot n} \\
p_j^\mu &= (1 - z)p^\mu - k_\perp^\mu - \frac{k_\perp^2}{1 - z} \frac{n^\mu}{2p \cdot n} ,
\end{aligned}
\tag{2.2.5}
$$

where p^μ denotes the collinear direction of the two partons, $k_\perp^\mu$ specifies the transverse direction perpendicular to the collinear direciton $(p \cdot k_\perp = 0)$, and n^μ is an auxiliary vector satisfying the conditions $n^2 = 0$ and $k_\perp \cdot n = 0$. z is the fraction of momenta carried away from the collinear momentum by parton i.

The collinear limit can be then be defined as

$$
2p_i p_j = s_{ij} = -\frac{k_\perp^2}{z(1 - z)} , \quad k_\perp \to 0 .
\tag{2.2.6}
$$

In this limit the matrix element can be written as

$$
|\mathcal{M}_{n+1}|^2 \to \frac{1}{p_i p_j} 4\pi \mu^{2\epsilon} \alpha_s \ _n\langle 1, \ldots, n + 1 | \hat{P}_{ij}(z, k_\perp) | 1, \ldots, n + 1 \rangle_n ,
\tag{2.2.7}
$$

where terms less singular than $k_\perp^2$ have been neglected. This reduced matrix element

is obtained by replacing the partons i and j with a single parton ij which carries the momentum p^μ and suitable quantum numbers depending on the partons i and j. For example, if $i =$ quark and $j =$ gluon, then $ij =$ quark, or if $i =$ quark and $j =$ antiquark, then $ij =$ gluon. $\hat{P}_{ij}$ are the d-dimensional Altarelli-Parisi splitting functions that depend on the momentum fraction z for the splitting $ij \to i + j$ and the transverse momentum $k_\perp$. Each splitting function is a matrix acting on the spin indices of ij. Due to these spin correlations the reduced matrix element does not factorise from the splitting functions exactly.

$\hat{P}_{ij}$ become the more recognisable Altarelli-Parisi splitting functions (1.3.3) once spin-averaged and the $\epsilon \to 0$ limit is taken.

In general, it is possible for a final-state parton i to become collinear with an initial-state parton a. This is described by the splitting process $a \to ai + i$. In this case their momenta can be decomposed as

$$
\begin{aligned}
p_i^\mu &= (1 - x)p_a^\mu + k_\perp^\mu - \frac{k_\perp^2}{1 - x}\frac{n^\mu}{2p_a \cdot n}\,, \\
p_a^\mu &= xp_a^\mu\,,
\end{aligned}
\tag{2.2.8}
$$

with the collinear limit defined as

$$
2p_i p_a = s_{ia} = -\frac{k_\perp^2}{1 - x}\,, \quad k_\perp \to 0\,.
\tag{2.2.9}
$$

The analogous expression of (2.2.7) for initial-state splitting is

$$
|\mathcal{M}_{n+1}|^2 \to \frac{1}{x}\frac{1}{p_i p_a}4\pi\mu^{2\epsilon}\alpha_s\,_n\langle 1, \ldots, n + 1; ai, \ldots |\hat{P}_{ai}(x, k_\perp)|1, \ldots, n + 1; ai, \ldots \rangle_n\,,
\tag{2.2.10}
$$

where the replacement of partons a and i have been made explicit by the presence of the parton ai in the colour + helicity vector. The specific type of parton ai depends upon the types of a and i.

Similar to (2.2.4) where the factorisation was only true in the strict soft limit, (2.2.7) and (2.2.10) are only true in the strict collinear limit. Away from these limits care has to be taken to conserve momenta via the use of momenta mappings.

2.2.3   Factorisation of colour-ordered amplitudes

The factorisation formulae expressed in (2.2.4), (2.2.7), and (2.2.10) were not exact because of colour and spin correlations. It can be shown that the factorisation of matrix elements in the soft and collinear limits becomes exact once the colour structure of the gauge group is separated from the kinematics.

In general any QCD amplitude can be colour decomposed, that is the colour structure is separated from the kinematics. Consider the process $e^+e^- \to q\bar{q} + n\ g$ [1], the amplitude can be written as a product of hadronic and leptonic currents [53]

$$\mathcal{M}(q_1, \bar{q}_2; 1, \ldots, n) = \hat{\mathcal{S}}_\mu^{n+2}(q_1; 1, \ldots, n; \bar{q}_2)V^\mu, \qquad (2.2.11)$$

where V^μ is the leptonic current and the hadronic current is

$$\hat{\mathcal{S}}_\mu^{n+2}(q_1; 1, \ldots, n; \bar{q}_2) = \mathrm{i}eg_s^n \sum_{P(1,\ldots,n)} (T^{a_1} \ldots T^{a_n})_{c_1 c_2} S_\mu(q_1; 1, \ldots, n; \bar{q}_2). \qquad (2.2.12)$$

In this expression we have the electromagnetic gauge coupling e and the strong gauge coupling g_s appearing. The colour structure has been factored into a product of fundamental group generators where the indices $a_i \in \{1, \ldots, N_c^2 - 1\}$ and $c_i \in \{1, \ldots, N_c\}$. This leaves the colour-ordered partial amplitude $S_\mu(q_1; 1, \ldots, n; \bar{q}_2)$ depending only on kinematical variables. In this partial amplitude, the gluons are emitted in an ordered fashion from the quarks, meaning the quarks have a fixed position in the partial amplitude. The sum over $P(1, \ldots, n)$ represents the sum over all permutations of gluon emissions which accounts for all Feynman diagrams and colour structures.

Upon squaring the amplitude (2.2.11), we get

$$\left| \hat{\mathcal{S}}_\mu^{n+2} V^\mu \right|^2 = e^2 \left(\frac{g_s^2 N_c}{2} \right)^n \left(\frac{N_c^2 - 1}{N_c} \right) \sum_{P(1,\ldots,n)} \left(\left| S_\mu(q_1; 1, \ldots, n; \bar{q}_2)V^\mu \right|^2 + \mathcal{O}\left(\frac{1}{N_c^2} \right) \right), \qquad (2.2.13)$$

where the subleading colour terms proportional to $1/N_c^2$ have been omitted. The

---

[1]For the scope of this thesis it is sufficient to consider colour decomposition of electron-positron annihilation into a single quark pair plus gluons, and not the more general case of multiple quark flavours in the final-state.

term proportional to $1/N_c$ is the leading colour term and is the dominant term in the colour expansion.

With the amplitude written in terms of the colour-ordered partial amplitudes, it is now possible to factorise (2.2.13) exactly in the soft and collinear limits. Consider the limit where a final-state gluon j is soft, we have

$$\left| S_\mu(q_1; 1, \ldots, i, j, k, \ldots, n; \bar{q}_2) V^\mu \right|^2 \to S_{ijk} \left| S_\mu(q_1; 1, \ldots, i, k, \ldots, n; \bar{q}_2) V^\mu \right|^2 , \quad (2.2.14)$$

where the factor

$$S_{ijk} = 4 \frac{s_{ik}}{s_{ij} s_{jk}} , \quad (2.2.15)$$

is the well-known eikonal factor. We see that the colour-ordered amplitude on the RHS of (2.2.14) has gluon j removed but the ordering of all hard partons remain unchanged.

In the collinear limit where partons i and j become collinear to form parton k, the matrix element factorises as

$$\left| S_\mu(q_1; 1, \ldots, i, j, \ldots, n; \bar{q}_2) V^\mu \right|^2 \to \frac{2}{s_{ij}} P_{ij}(z) \left| S_\mu(q_1; 1, \ldots, k, \ldots, n; \bar{q}_2) V^\mu \right|^2 ,$$
$$(2.2.16)$$

where $P_{ij}(z)$ are the spin-averaged Altarelli-Parisi splitting functions (1.3.3) with the colour factors removed. In this expression it is understood that the indices i, j, and k have to be self-consistent for the different partonic splittings. For partons which are not colour connected (are not neighbouring partons in the colour-ordered amplitude), there will be no singular behaviour as $s_{ij} \to 0$.

In (2.2.14) and (2.2.16) the colour-ordered amplitude factorises exactly into a colour-ordered amplitude with one parton removed, multiplied by a universal singular factor. This singular factor depends on the unresolved parton and the two neighbouring hard particles. Interpreting this as the two hard particles forming an antenna which radiates the unresolved parton gives rise to the antenna functions which will be discussed further in Section 2.5.

2.2.4 One-loop matrix element factorisation

The discussion of matrix element factorisation so far has been focused on tree-level matrix elements. It has been shown that one-loop colour-ordered amplitudes also factorise in the soft and collinear limits [54–56]. At the one-loop level, there are new universal singular functions and the factorisation formulae are modified.

In the soft limit, a one-loop colour-ordered amplitude factorises as

$$M_{n+1}^{(1)}(\ldots, i, j, k, \ldots) \to S_{ijk}^{(0)} M_n^1(\ldots, i, k, \ldots) + S_{ijk}^{(1)}(\epsilon) M_n^0(\ldots, i, k, \ldots), \quad (2.2.17)$$

where $S_{ijk}^{(0)}$ is the eikonal factor (2.2.15), and $S_{ijk}^{(1)}(\epsilon)$ is the one-loop soft radiation function [57].

A similar factorisation formula for the collinear limit is

$$M_{n+1}^{(1)}(\ldots, i, j, \ldots) \to \frac{1}{s_{ij}} \left[P_{ij}^{(0)}(z) M_n^{(1)}(\ldots, k, \ldots) + P_{ij}^{(1)}(z, \epsilon) M_n^{(0)}(\ldots, k, \ldots) \right],$$
$$(2.2.18)$$

where $P_{ij}^{(0)}(z)$ are the tree-level splitting functions (1.3.3) and $P_{ij}^{(1)}(z, \epsilon)$ are the one-loop splitting functions [57].

In both these formulae the structure is of the form: tree-level splitting function multipled by a one-loop amplitude, plus a one-loop splitting function multiplied by a tree-level amplitude.

In Section 2.5 we will give a brief overview of antenna functions where these one-loop factorisation formulae were used to obtain universal singular functions at the one-loop level for squared matrix elements. These antenna functions are then applied in the context of NLO k-factor emulation in Chapter 6.

2.3 Subtraction

In Section 2.1.4 we discussed the structure of IR divergences and how the KLN theorem necessitated the cancellation of IR divergences arising from the virtual

corrections and real-emission matrix elements once integrated over soft and collinear regions of phase-space, at each order in perturbation theory.

Over the past few decades there has been a vast amount of research into devising methods to systematically isolate these singularities such that it is possible to make finite predictions of physical quantities numerically. To tackle this problem, three main mathods have been proposed: phase-space slicing [58–60], sector decomposition [61, 62], and subtraction [63].

By now the method of choice at NLO QCD is subtraction, and is the method we will focus on in this section. The main idea behind subtraction methods is to define local counterterms that exactly replicate the IR divergent behaviour of matrix elements. While there is not one single subtraction scheme that is universally used, the general form of a subtraction term must fulfill the requirements of replicating the matrix element behaviour in all IR singular limits, and be analytically integrable over the regions of phase-space corresponding to these IR limits.

To illustrate the idea behind the subtraction method, first consider the calculation of a LO partonic cross-section

$$\sigma^{\mathrm{LO}} = \int \mathrm{d}\Phi_n \, \mathcal{B}_n \, , \tag{2.3.1}$$

where $\mathcal{B}_n$ is the Born (tree-level) matrix element and Φ_n is the n-body phase-space. At the next order in perturbation theory, we have the NLO cross-section which recieves contributions from the real and virtual corrections

$$\sigma^{\mathrm{NLO}} = \int \mathrm{d}\Phi_n \, [\mathcal{B}_n + \mathcal{V}_n] + \int \mathrm{d}\Phi_{n+1} \, \mathcal{R}_{n+1} \, , \tag{2.3.2}$$

where $\mathcal{V}_n$ is the virtual matrix element (renormalised to remove UV divergences as described in Section 2.1.1) which lives in the same n-body phase-space as the Born matrix element. The real-emission matrix element, $\mathcal{R}_{n+1}$, lives in the $(n+1)$-body phase-space, Φ_{n+1}, due to the emission of an additional external particle. The integrals over Φ_n and Φ_{n+1} are separately divergent but their sum is finite. To carry out a numerical calculation, it is therefore necessary to regulate these divergences to

make them explicit. Using dimensional regularisation these divergences are mapped to poles in ϵ.

The motivation behind the subtraction method is that the divergences in (2.3.2) can be cancelled upon the insertion of a counterterm evaluated in Φ_{n+1}, $\mathcal{C}_{n+1}$, and an integrated counterterm evaluated in Φ_n, $\mathcal{I}_n$, such that the condition

$$\int \mathrm{d}\Phi_n \, \mathcal{I}_n - \int \mathrm{d}\Phi_{n+1} \, \mathcal{C}_{n+1} = 0 \,, \tag{2.3.3}$$

holds. Here phase-space factorisation is utilised: $\Phi_{n+1} \to \Phi_n \Phi_1$ when using an appropriate $3 \to 2$ momentum mapping such that

$$\mathcal{I}_n = \int \Phi_1 \, \mathcal{C}_{n+1} \,, \tag{2.3.4}$$

where Φ_1 is the one-parton phase-space leading to ϵ poles once $\mathcal{C}_{n+1}$ is integrated over. The counterterm $\mathcal{C}_{n+1}$ should be a proper pointwise approximation of $\mathcal{R}_{n+1}$ to cancel all IR divergences such that $\mathcal{R}_{n+1} - \mathcal{C}_{n+1}$ is finite. Additionally, since the pole structure in $\mathcal{V}_n$ is identical to $\mathcal{R}_{n+1}$ but with an opposite sign, $\mathcal{V}_n + \mathcal{I}_n$ will be finite by construction. Inserting the subtraction terms into (2.3.2) we get

$$\sigma^{\mathrm{NLO}} = \int \mathrm{d}\Phi_n \left[\mathcal{B}_n + \mathcal{V}_n + \mathcal{I}_n \right] + \int \mathrm{d}\Phi_{n+1} \left[\mathcal{R}_{n+1} - \mathcal{C}_{n+1} \right] \,, \tag{2.3.5}$$

where each integral can now be carried out numerically in integer dimensions by taking the $\epsilon \to 0$ limit. This is possible as the integrals are all separately finite now. The counterterm $\mathcal{C}_{n+1}$ has been kept general but specific exmples of automated subtractions schemes include Catani-Seymour (CS) [50, 64], Frixione-Kunszt-Signer (FKS) [65, 66], and antenna subtraction [52, 53, 67].

Beyond NLO QCD, the algorithms available have not reached the maturity of the automated methods widely use at NLO. However, this is an active area of research. See Reference [68] for a review of methods that have been applied to NNLO QCD.

In the next sections we will describe in detail two sets of functions that are used to build subtraction terms: Catani-Seymour dipoles, and antenna functions. However, we will not construct the counterterms explicitly. In other words, we are interested

in the approximations of the matrix elements in the soft and collinear limits as these are universal and can be applied to any process, and not the subtraction terms themselves. These functions will become instrumental during our construction of matrix element emulators in Chapters 5, 6, and 7.

2.4 Catani-Seymour dipoles

Catani-Seymour dipoles introduced in Ref. [50] are process independent functions that reproduce the IR singular behaviour of matrix elements. They depend on the momenta and quantum numbers of three partons in the real-emission phase-space. These three partons are identified by indices i, j, and k where i is the emitting parton, j is the unresolved emitted parton, and k is a spectator parton. Since the dipoles only depend on three partons, to map out all the singular limits in a process, it is necessary to construct all the permutations of dipole functions.

Dipole functions can be separated into four categories depending on whether the emitter and spectator are in the initial-state or the final-state, as illustrated in Figure 2.4. Namely, there are final-final (FF) dipoles, final-initial dipoles (FI), initial-final (IF) dipoles and initial-initial (II) dipoles where the nomenclature is emitter-spectator dipole. For electron-positron annihilations only FF dipoles are required, but for any hadronic collision the inclusion of the remaining dipoles are required to capture all IR singular behaviour.

In the following we will describe in detail the FF dipoles and the associated matrix element factorisation formula, but only give a brief description of the remaining dipoles as the structure of the terms and factorisation formulae are essentially identical.

(a) FF dipole

(b) FI dipole

(c) IF dipole

(d) II dipole
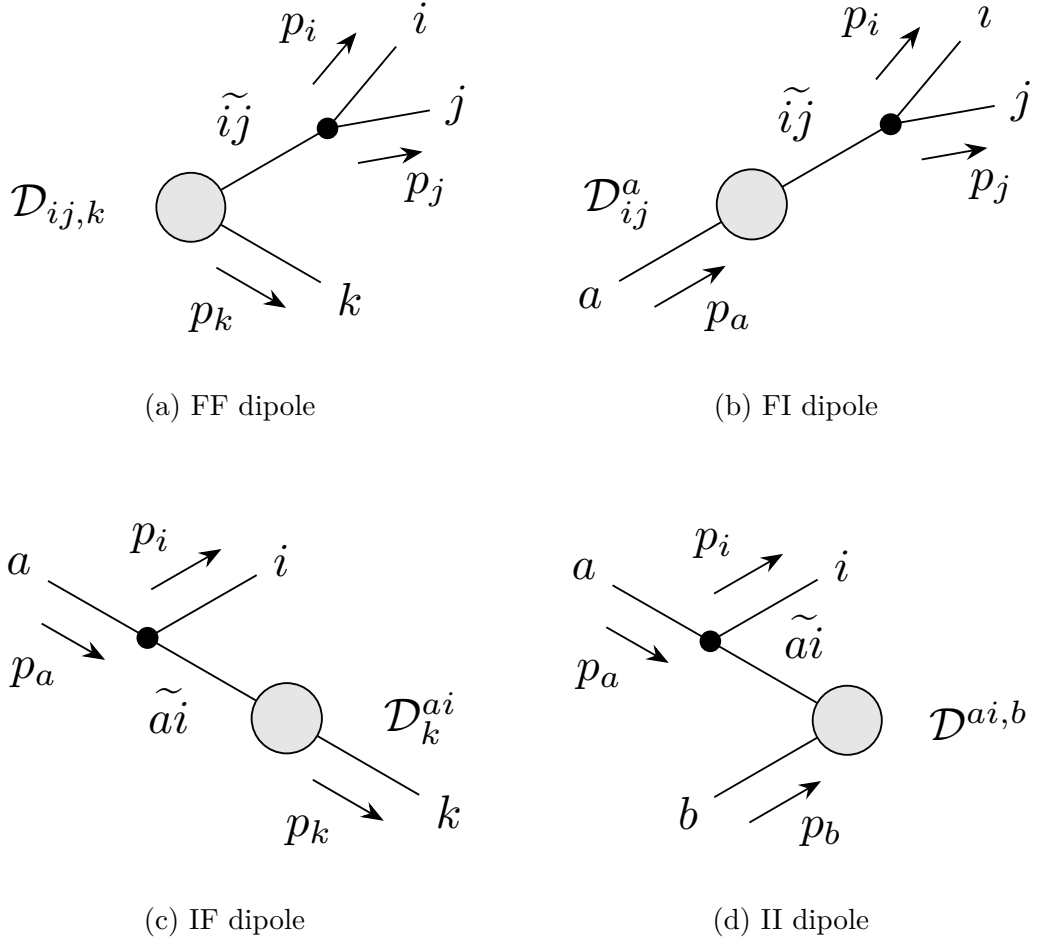
Figure 2.4: Schematic diagrams of the four classes of Catani-Seymour dipoles, $\mathcal{D}$. The dipoles are named according to whether the emitter and spectator are in the initial (upper indices) or final-state (lower indices). Each dipole consists of a composite particle (denoted by tilde) that decays into two partons, and a spectator that recoils to conserve momentum. The grey blob represents the hard scattering process, with incoming and outgoing lines representing initial and final-state partons, respectively. The black circle represents the splitting function within the dipole function which contains the divergent behaviour.

2.4.1   Final-final dipoles

The utilisation of dipoles is encapsulated in the dipole factorisation formula where matrix elements in the limit $p_i p_j \to 0$ can be written as

$$_{n+1}\langle 1, \dots, n+1 || 1, \dots, n+1 \rangle_{n+1} = \sum_{i,j} \sum_{k \neq i,j} \mathcal{D}_{ij,k}(p_1, \dots, p_{n+1}) + \dots, \quad (2.4.1)$$

where terms not singular in the limit $p_i p_j \to 0$ are omitted, and the dipole is

$$\mathcal{D}_{ij,k}(p_1, \dots, p_{n+1}) = -\frac{1}{2 p_i p_j}$$
$$_{n}\langle 1, \dots, \widetilde{ij}, \dots, \tilde{k}, \dots, n+1 | \frac{\boldsymbol{T}_k \cdot \boldsymbol{T}_{ij}}{\boldsymbol{T}_{ij}^2} \boldsymbol{V}_{ij,k} | 1, \dots, \widetilde{ij}, \dots, \tilde{k}, \dots, n+1 \rangle_n,$$

$$(2.4.2)$$

where $\boldsymbol{T}_i$ are the colour-charge operators and $\boldsymbol{V}_{ij,k}$ is a matrix in the helicity space of the emitter embedding the IR divergent behaviour. The sum in (2.4.1) can be understood as summing over all possible 3 leg permutations to capture all the soft and collinear limits. The reduced matrix element on the RHS of (2.4.2) is obtained by replacing the partons i and j with a single parton $\widetilde{ij}$, and replacing parton k with a parton $\tilde{k}$. $\tilde{k}$ has all the same quantum numbers as k, where as the partonic nature of $\widetilde{ij}$ depends on the specific splitting process (c.f. Section 2.2.2). The momenta of these particles are modified in the following way

$$\tilde{p}_{ij}^\mu = p_i^\mu + p_j^\mu - \frac{y_{ij,k}}{1 - y_{ij,k}} p_k^\mu, \quad \tilde{p}_k^\mu = \frac{1}{1 - y_{ij,k}} p_k^\mu, \quad (2.4.3)$$

where $y_{ij,k}$, the recoil parameter, is a dimensionless variable given as

$$y_{ij,k} = \frac{p_i p_j}{p_i p_j + p_i p_k + p_j p_k}. \quad (2.4.4)$$

Equations (2.4.3) and (2.4.4) are a $3 \to 2$ momenta mapping that maps $p_i + p_j + p_k \to \tilde{p}_{ij} + \tilde{p}_k$. The mapping ensures momentum conservation is maintained across all of phase-space and all particles are kept on-shell:

$$p_i^\mu + p_j^\mu + p_k^\mu = \tilde{p}_{ij}^\mu + \tilde{p}_k^\mu,$$
$$\tilde{p}_{ij}^2 = \tilde{p}_k^2 = 0. \quad (2.4.5)$$

The matrices $\boldsymbol{V}_{ij,k}$ are functions of $y_{ij,k}$ and the splitting variables $\tilde{z}$

$$\tilde{z}_i = \frac{p_i p_k}{p_i p_k + p_j p_k}\,, \quad \tilde{z}_j = \frac{p_j p_k}{p_i p_k + p_j p_k} = 1 - \tilde{z}_i\,. \tag{2.4.6}$$

They are analogous to the z from Altarelli-Parisi splitting functions. $\boldsymbol{V}_{ij,k}$ acts on the spin indices of the composite particle $\widetilde{ij}$, but is independent of the type of the spectator. Making the spin-dependence on parton $\widetilde{ij}$ explicit (s and s' for $\widetilde{ij} = $ fermion, and μ and ν for $\widetilde{ij} = $ gluon) we list all the kernels here

$$\langle s|\boldsymbol{V}_{q_i g_j,k}|s'\rangle = 8\pi\mu^{2\epsilon}\alpha_s C_F \left[\frac{2}{1 - \tilde{z}_i(1 - y_{ij,k})} - (1 + \tilde{z}_i) - \epsilon\tilde{z}_j\right]\delta_{ss'}\,,$$

$$\langle \mu|\boldsymbol{V}_{q_i,\bar{q}_j,k}|\nu\rangle = 8\pi\mu^{2\epsilon}\alpha_s T_R \left[-g^{\mu\nu} - \frac{2}{p_i p_j}(\tilde{z}_i p_i^\mu - \tilde{z}_j p_j^\mu)(\tilde{z}_i p_i^\nu - \tilde{z}_j p_j^\nu)\right]\,,$$

$$\langle \mu|\boldsymbol{V}_{g_i g_j,k}|\nu\rangle = 16\pi\mu^{2\epsilon}\alpha_s C_A \left[-g^{\mu\nu}\left(\frac{1}{1 - \tilde{z}_i(1 - y_{ij,k})} + \frac{1}{1 - \tilde{z}_j(1 - y_{ij,k})} - 2\right)\right.$$
$$\left. + \frac{(1 - \epsilon)}{p_i p_j}(\tilde{z}_i p_i^\mu - \tilde{z}_j p_j^\mu)(\tilde{z}_i p_i^\nu - \tilde{z}_j p_j^\nu)\right]\,.$$

$$\tag{2.4.7}$$

A key feature of these kernels are that they smoothly interpolate between the soft and collinear limits, meaning they do not double count any limit. Additionally, they only contain divergences in the $p_i p_j \to 0$ limit and not for any other pair of momenta. In the soft and collinear limits, the dipole function $\mathcal{D}_{ij,k}$ correctly reproduces the matrix element factorisation behaviour seen in (2.2.4) and (2.2.7), repsectively. In particular, $\boldsymbol{V}_{ij,k}$ becomes proportional to the eikonal factor and Altarelli-Parisi splitting functions in these respective limits.

In this thesis we will be focussing on the emulation of colour and spin-averaged matrix elements. Therefore the spin-indices are not explicitly available. It is possible to average over the spin indices in (2.4.7) to obtain the spin-averaged splitting functions

$$\langle\boldsymbol{V}_{q_i g_j,k}\rangle = 8\pi\mu^{2\epsilon}\alpha_s C_F \left[\frac{2}{1 - \tilde{z}_i(1 - y_{ij,k})} - (1 + \tilde{z}_i) - \epsilon\tilde{z}_j\right]\,,$$

$$\langle\boldsymbol{V}_{q_i \bar{q}_j,k}\rangle = 8\pi\mu^{2\epsilon}\alpha_s T_R \left[1 - \frac{2\tilde{z}_i\tilde{z}_j}{1 - \epsilon}\right]\,,$$

$$\langle\boldsymbol{V}_{g_i g_j,k}\rangle = 16\pi\mu^{2\epsilon}\alpha_s C_A \left[\frac{1}{1 - \tilde{z}_i(1 - y_{ij,k})} + \frac{1}{1 - \tilde{z}_j(1 - y_{ij,k})} - 2 + \tilde{z}_i\tilde{z}_j\right]\,. \tag{2.4.8}$$

2.4.2　Final-initial dipoles

For the case of final-state emitter and initial-state spectator we have the dipole factorisation formula

$$
{}_{n+1}\langle 1, \ldots, n+1; a, \ldots \| 1, \ldots, n+1; a, \ldots \rangle_{n+1} =
$$
$$
\sum_{i,j} \sum_{k \neq i,j} \mathcal{D}_{ij,k}(p_1, \ldots, p_{n+1}; p_a, \ldots) \tag{2.4.9}
$$
$$
\sum_{i,j} \sum_{a} \mathcal{D}_{ij}^a(p_1, \ldots, p_{n+1}; p_a, \ldots) + \ldots ,
$$

where the new dipole contribution appears on the second line. The initial-state parton is represented as a. The dipole term $\mathcal{D}_{ij}^a$ is given by

$$
\mathcal{D}_{ij}^a(p_1, \ldots, p_{n+1}; p_a, \ldots) = -\frac{1}{2p_i p_j} \frac{1}{x_{ij,a}}
$$
$$
{}_{n}\langle 1, \ldots, \widetilde{ij}, \ldots, n+1; \tilde{a}, \ldots | \frac{\boldsymbol{T}_a \cdot \boldsymbol{T}_{ij}}{\boldsymbol{T}_{ij}^2} \boldsymbol{V}_{ij}^a | 1, \ldots, \widetilde{ij}, \ldots, n+1; \tilde{a}, \ldots \rangle_n .
$$
$$
\tag{2.4.10}
$$

where $\widetilde{ij}$ is the composite emitter in the final-state, and the spectator $\tilde{a}$ is in the initial-state. The momentum mapping for $p_i + p_j + p_a \to \tilde{p}_{ij} + \tilde{p}_a$ is given as

$$
\tilde{p}_{ij}^\mu = p_i^\mu + p_j^\mu - (1 - x_{ij,a})p_a^\mu , \quad \tilde{p}_a^\mu = x_{ij,a}\, p_a^\mu , \tag{2.4.11}
$$

such that the following momentum conservation and on-shell conditions are met

$$
p_i^\mu + p_j^\mu - p_a^\mu = \tilde{p}_{ij}^\mu - \tilde{p}_a^\mu ,
$$
$$
\tilde{p}_{ij}^2 = \tilde{p}_a^2 = 0 . \tag{2.4.12}
$$

The recoil parameter $x_{ij,a}$ and splitting variables $\tilde{z}$ are given by

$$
x_{ij,a} = \frac{p_i p_a + p_j p_a - p_i p_j}{(p_i p_a + p_j p_a)} , \tag{2.4.13}
$$
$$
\tilde{z}_i = \frac{p_i p_a}{p_i p_a + p_j p_a} , \quad \tilde{z}_j = \frac{p_j p_a}{p_i p_a + p_j p_a} = 1 - \tilde{z}_i . \tag{2.4.14}
$$

The spin-averaged splitting functions are given as

$$
\langle \boldsymbol{V}_{q_i g_j, a} \rangle = 8\pi\mu^{2\epsilon}\alpha_s C_F \left[\frac{2}{1 - \tilde{z}_i + (1 - x_{ij,a})} - (1 + \tilde{z}_i) - \epsilon\tilde{z}_j \right] ,
$$

$$\langle \boldsymbol{V}_{q_i\bar{q}_j,a} \rangle = 8\pi\mu^{2\epsilon}\alpha_s T_R \left[1 - \frac{2\tilde{z}_i\tilde{z}_j}{1-\epsilon} \right],$$

$$\langle \boldsymbol{V}_{g_ig_j,a} \rangle = 16\pi\mu^{2\epsilon}\alpha_s C_A \left[\frac{1}{\tilde{z}_j + (1 - x_{ij,a})} + \frac{1}{\tilde{z}_i + (1 - x_{ij,a})} - 2 + \tilde{z}_i\tilde{z}_j \right]. \quad (2.4.15)$$

2.4.3 Initial-final dipoles

For the case of initial-state emitter and final-state spectator, the dipole factorisation formula is given as

$$_{n+1}\langle 1, \ldots, n+1; a || 1, \ldots, n+1; a \rangle_{n+1} = \sum_{a,i} \sum_{k\neq i} \mathcal{D}_k^{ai}(p_1, \ldots, p_{n+1}; p_a) + \ldots, \quad (2.4.16)$$

where there is only one initial-state parton. The dipole is given by

$$\mathcal{D}_k^{ai}(p_1, \ldots, p_{n+1}; p_a) = -\frac{1}{2p_ap_i}\frac{1}{x_{ik,a}}$$
$$_n\langle 1, \ldots, \tilde{k}, \ldots, n+1; \widetilde{ai}| \frac{\boldsymbol{T}_k \cdot \boldsymbol{T}_{ai}}{\boldsymbol{T}_{ai}^2} \boldsymbol{V}_k^{ai} |1, \ldots, \tilde{k}, \ldots, n+1; \widetilde{ai}\rangle_n. \quad (2.4.17)$$

where the emitter is the initial-state parton $\widetilde{ai}$ and the spectator is the final-state parton $\tilde{k}$. The momenta mapping for $p_a + p_i + p_k \rightarrow \tilde{p}_{ai} + \tilde{p}_k$ is

$$\tilde{p}_{ai}^\mu = x_{ik,a}\,p_a^\mu, \quad \tilde{p}_k^\mu = p_k^\mu + p_i^\mu - (1 - x_{ik,a})p_a^\mu \quad (2.4.18)$$

such that

$$p_i^\mu + p_k^\mu - p_a^\mu = \tilde{p}_k^\mu - \tilde{p}_{ai}^\mu,$$
$$\tilde{p}_{ai}^2 = \tilde{p}_k^2 = 0. \quad (2.4.19)$$

Note that the momentum $\tilde{p}_{ai}$ is parallel to p_a, therefore the modified spectator momentum $\tilde{p}_k$ has to recoil the transverse momentum. This leads to $x_{ik,a}$ acting as a splitting variable.

With $x_{ik,a}$ taking the role of $\tilde{z}$, there is another parameter, u_i, appearing in the splitting functions. These are given as

$$x_{ik,a} = \frac{p_kp_a + p_ip_a - p_ip_k}{p_ip_a + p_kp_a}, \quad u_i = \frac{p_ip_a}{p_ip_a + p_kp_a}. \quad (2.4.20)$$

The spin-averaged splitting functions are

$$\langle \boldsymbol{V}_k^{q_a g_i} \rangle = \frac{n_s(q)}{n_s(\tilde{q})}\, 8\pi\mu^{2\epsilon}\alpha_s C_F \left[\frac{2}{1 - x_{ik,a} + u_i} + (1 + x_{ik,a}) - \epsilon(1 - x_{ik,a}) \right] ,$$

$$\langle \boldsymbol{V}_k^{g_a \bar{q}_i} \rangle = \frac{n_s(g)}{n_s(\tilde{q})}\, 8\pi\mu^{2\epsilon}\alpha_s T_R \left[1 - \frac{2 x_{ik,a}(1 - x_{ik,a})}{1 - \epsilon} \right] ,$$

$$\langle \boldsymbol{V}_k^{g_a g_i} \rangle = \frac{n_s(g)}{n_s(\tilde{g})}\, 16\pi\mu^{2\epsilon}\alpha_s C_A \left[\frac{1}{1 - x_{ik,a} + u_i} + \frac{1 - x_{ik,a}}{x_{ik,a}} - 1 + x_{ik,a}(1 - x_{ik,a}) \right] ,$$

$$\langle \boldsymbol{V}_k^{q_a q_i} \rangle = \frac{n_s(q)}{n_s(\tilde{g})}\, 8\pi\mu^{2\epsilon}\alpha_s C_F \left[(1 - \epsilon)x_{ik,a} + 2\frac{1 - x_{ik,a}}{x_{ik,a}} \right] , \qquad (2.4.21)$$

where $n_s(a)$ is the number of polarisations of particle a. For fermions $n_s(q) = n_s(\bar{q}) = 2$ and for gluons $n_s(g) = d - 2$. In the limit $\epsilon \to 0$ all ratios of n_s cancel to give unity.

2.4.4   Initial-initial dipoles

In the case of two initial-state partons, a and b, there is an additional dipole contribution. The dipole factorisation in this case is given by

$$_{n+1}\langle 1, \ldots, n + 1; a, b || 1, \ldots, n + 1; a, b \rangle_{n+1} =$$

$$\sum_{a,i} \sum_{k \neq i} \mathcal{D}_k^{ai}(p_1, \ldots, p_{n+1}; p_a, p_b) \qquad (2.4.22)$$

$$\sum_{a,i} \sum_{b \neq a} \mathcal{D}^{ai,b}(p_1, \ldots, p_{n+1}; p_a, p_b) + \ldots ,$$

where the new contribution is the dipole $\mathcal{D}^{ai,b}$ given by

$$\mathcal{D}^{ai,b}(p_1, \ldots, p_{n+1}; p_a, p_b) = -\frac{1}{2 p_a p_i}\frac{1}{x_{i,ab}}$$

$$_n\langle \widetilde{1}, \ldots, \widetilde{n+1}; \widetilde{ai}, b | \frac{\boldsymbol{T}_b \cdot \boldsymbol{T}_{ai}}{\boldsymbol{T}_{ai}^2} \boldsymbol{V}^{ai,b} | \widetilde{1}, \ldots, \widetilde{n+1}; \widetilde{ai}, b \rangle_n .$$

$$(2.4.23)$$

The initial-state parton $\widetilde{ai}$ is the emitter and the other initial-state parton b is the spectator. Notice that even the momenta not involved in the dipole term have been modified with only parton b remaining unchanged. The momentum mapping for this case has $\tilde{p}_{ai}$ parallel with p_a and also modifies the momenta for all other final-state

momenta (even non-QCD particles) k_j:

$$\tilde{p}^\mu_{ai} = x_{i,ab}\, p^\mu_a\,,$$

$$\tilde{k}^\mu_j = k^\mu_j - \frac{2k_j \cdot (K + \widetilde{K})}{(K + \widetilde{K})^2}(K + \widetilde{K})^\mu + \frac{2k_j \cdot K}{K^2}\widetilde{K}^\mu\,, \qquad (2.4.24)$$

$$x_{i,ab} = \frac{p_a p_b - p_i p_a - p_i p_b}{p_a p_b}\,.$$

where K and $\widetilde{K}$ are the total momenta of the dipole before and after the mapping. They are given by

$$K^\mu = p^\mu_a + p^\mu_b - p^\mu_i\,,$$

$$\widetilde{K}^\mu = \tilde{p}^\mu_{ai} + p^\mu_b\,. \qquad (2.4.25)$$

This mapping conserves momentum and keeps mapped momenta on-shell

$$p^\mu_a + p^\mu_b - p^\mu_i - \sum_j k^\mu_j = \tilde{p}^\mu_{ai} + p^\mu_b - \sum_j \tilde{k}^\mu_j = 0\,,$$

$$\tilde{p}^2_{ai} = \tilde{k}^2_j = 0\,. \qquad (2.4.26)$$

The spin-averaged splitting functions are proportional to the Altarelli-Parisi splitting functions which we quote here for completeness

$$\langle \boldsymbol{V}^{q_a q_i, b}\rangle = \frac{n_s(q)}{n_s(\tilde{g})}\, 8\pi\mu^{2\epsilon}\alpha_s C_F \left[\frac{1 + (1 - x_{i,ab})^2}{x_{i,ab}}\right]\,,$$

$$\langle \boldsymbol{V}^{q_a g_i, b}\rangle = \frac{n_s(q)}{n_s(\tilde{q})}\, 8\pi\mu^{2\epsilon}\alpha_s C_F \left[\frac{1 + x^2_{i,ab}}{1 - x_{i,ab}}\right]\,,$$

$$\langle \boldsymbol{V}^{g_a \bar{q}_i, b}\rangle = \frac{n_s(g)}{n_s(\tilde{q})}\, 8\pi\mu^{2\epsilon}\alpha_s T_R \left[x^2_{i,ab} + (1 - x_{i,ab})^2\right]\,, \qquad (2.4.27)$$

$$\langle \boldsymbol{V}^{g_a g_i, b}\rangle = \frac{n_s(g)}{n_s(\tilde{g})}\, 16\pi\mu^{2\epsilon}\alpha_s C_A \left[\frac{x_{i,ab}}{1 - x_{i,ab}} + \frac{1 - x_{i,ab}}{x_{i,ab}} + x_{i,ab}(1 - x_{i,ab})\right]\,.$$

2.4.5 Master factorisation formula

Now that we have described each of the dipole contributions, we can collect them all into a master dipole factorisation formula

$$
{}_{n+1}\langle 1, \ldots, n+1 || 1, \ldots, n+1 \rangle_{n+1} =
$$
$$
\sum_{i,j} \sum_{k \neq i,j} \mathcal{D}_{ij,k}(p_1, \ldots, p_{n+1}; p_a, p_b) + \sum_{i,j} \sum_{a} \mathcal{D}_{ij}^{a}(p_1, \ldots, p_{n+1}; p_a, p_b)
$$
$$
\sum_{a,i} \sum_{k \neq i} \mathcal{D}_{k}^{ai}(p_1, \ldots, p_{n+1}; p_a, p_b) + \sum_{a,i} \sum_{b \neq a} \mathcal{D}^{ai,b}(p_1, \ldots, p_{n+1}; p_a, p_b) + \ldots,
$$

$$(2.4.28)$$

where it is understood that for a given partonic process, the matrix element in the different soft and collinear limits will be reproduced by summing over all the possible dipoles that are relevant and hence will form the set of basis functions for approximating the matrix elements in Chapters 5 and 7. The omitted terms are non-divergent in all of phase-space.

2.4.6 Treatment of massive partons

The dipoles described in this section so far have treated all partons to be massless. Since quarks are massive particles it is not possible in general to treat them as massless. This is especially true for the top quark which has a mass of approximately 175 GeV, the most massive SM particle. In the case of partonic processes involving top quarks, it is therefore necessary to include mass effects into the dipole functions.

The universality of infrared divergent structure extends to massive quarks [69]. In Reference [64] the authors derive dipole functions for massive partons which are utilised in Chapter 7, but will not be written here as the philosophy of constructing the massive dipole functions is similar to massless ones.

The extension to massive quarks necessitates the straightforward modification of the master dipole factorisation formula (2.4.28) to replace any massless dipole with its massive counterpart whenever there is a massive parton in the dipole.

2.5   Antenna functions

Another set of functions that can be used to construct subtractions terms in NLO [52, 53] and NNLO QCD [51, 70] are the antenna functions. They are derived from physical matrix elements and so by construction contain the correct IR behaviour in the soft and collinear limits. More specifically, they are derived from colour-ordered matrix elements so they follow the colour-ordered factorisation properties outlined in Section 2.2.3. In the factorisation formulae, the colour-ordered matrix elements factorised exactly into a colour-ordered matrix element with one parton removed multiplied by a singular factor. Since this factorisation is exact, and the IR behaviour is universal, it is possible to extract these singular factors once and for all.

In contrast to the dipole functions, where there is an identified emitter, antenna functions can have the unresolved parton radiate from either of the two hard partons in the antenna. In this sense, an antenna is a linear combination of two dipoles.

The discussion in this section will be limited to the situation of a massive colour-neutral boson decaying to massless QCD partons, suitable for multijet production in electron-positron annihilation. Additionally, we assume that there will only ever be at most one unresolved parton in the final-state. The antenna formalism has been extended to more general cases than this [71–75].

Antenna functions are constructed from a ratio of colour-ordered matrix elements. For the case of one unresolved parton, the three-parton tree-level antenna function is given by

$$X_3^0(i, j, k) = S_{ijk,IK} \frac{|M_{n+1}^{(0)}(i, j, k)|^2}{|M_n^{(0)}(I, K)|^2} \tag{2.5.1}$$

where $|M_n^{(\ell)}|^2$ is the n-body colour-ordered matrix element at loop-level ℓ. $S_{ijk,IK}$ is a symmetry factor accounting for identical particles in the final-state and for the presence of multiple antennae in the two parton process. I and K are hard partons forming a colour connected antenna that radiates particle j. The identities of i/I and k/K defines the specific class of the antenna function. There are three classes corresponding to the three underlying two-parton processes: quark-antiquark,

quark-gluon and gluon-gluon. Therefore, there will be an antenna for every radiative correction to these two-parton processes. The physical matrix elements used to the derive the antenna functions for each class are as follows:

- Quark-antiquark: $\gamma^* \to q\bar{q}$ + (partons), the decay of a virtual photon into a quark-antiquark pair with QCD radiation from the quark pair [70].

- Quark-gluon: $\tilde{\chi} \to \tilde{g}g$ + (partons), the decay of a heavy neutralino into a gluino and gluon with QCD radiation from the gluon-gluino pair [76].

- Gluon-gluon: $H \to gg$ + (partons), the decay of a Higgs boson into a pair of gluons with QCD radiation from the gluon pair [77].

Along with the tree-level antenna, there are also the one-loop antennae. For the three-parton case these are given by

$$X_3^1(i,j,k) = S_{ijk,IK} \frac{|M_{n+1}^{(1)}(i,j,k)|^2}{|M_n^{(0)}(I,K)|^2} - X_3^0(i,j,k) \frac{|M_n^{(1)}(I,K)|^2}{|M_n^{(0)}(I,K)|^2} , \qquad (2.5.2)$$

where X_3^1 are defined such that they are proportional to the one-loop singular functions. Since the one-loop factorisation formulae (2.2.17) and (2.2.18) are of the form $|M_{n+1}^{(1)}|^2 \to S^{(0)}|M_n^{(1)}|^2 + S^{(1)}|M_n^{(0)}|^2$, the term proportional to X_3^0 has to be removed to fulfill this requirement. The antenna functions are summarised in Table 2.1, where they are categorised by their class and radiative processes.

Another requirement in constructing a subtraction term is that the functions have to be analytically integrable over the unresolved phase-space. For one unresolved parton, this requirement is met with a suitable $3 \to 2$ momentum mapping that allows the $(n+1)$-body phase-space to factorise into a product of the mapped n-body phase-space and the antenna phase-space. The antenna phase-space is independent of the n-body phase-space and depends only on the momenta of partons i, j, k appearing in the antenna. In the case of antenna functions, the mapping of choice

| Class | Radiation | Antenna functions | |
		Tree-level	One-loop
Quark-antiquark	$q\bar{q} \to qg\bar{q}$	A_3^0	$A_3^1,\ \tilde{A}_3^1,\ \hat{A}_3^1$
Quark-gluon	$qg \to qgg$	D_3^0	$D_3^1,\ \hat{D}_3^1$
	$qg \to qQ\bar{Q}$	E_3^0	$E_3^1,\ \tilde{E}_3^1,\ \hat{E}_3^1$
Gluon-gluon	$gg \to ggg$	F_3^0	$F_3^1,\ \hat{F}_3^1$
	$gg \to gq\bar{q}$	G_3^0	$G_3^1,\ \tilde{G}_3^1,\ \hat{G}_3^1$

Table 2.1: Three-parton antenna functions at tree-level, X_3^0, and one-loop level, X_3^1. The different antenna functions at one-loop level correspond to the leading colour (X_3^1), subleading colour $(\tilde{X}_3^1)$, and closed quark loop $(\hat{X}_3^1)$ contributions.

is the Kosower mapping [78] which maps $p_i + p_j + p_k \to p_I + p_K$. It reads

$$p_I = xp_i + rp_j + yp_k$$
$$p_K = (1-x)p_i + (1-r)p_j + (1-y)p_k\,, \tag{2.5.3}$$

where the mapping variables are given as

$$s_{ijk} = s_{ij} + s_{ik} + s_{jk}\,,$$
$$r = \frac{s_{jk}}{s_{ij}s_{jk}}\,,$$
$$\rho = \sqrt{1 + 4r(1-r)\frac{s_{ij}s_{jk}}{s_{ijk}s_{ik}}}\,,$$
$$x = \frac{(1+\rho)s_{ijk} - 2rs_{jk}}{2(s_{ij} + s_{ik})}\,,$$
$$y = \frac{(1-\rho)s_{ijk} - 2rs_{ij}}{2(s_{jk} + s_{ik})}\,. \tag{2.5.4}$$

This momentum mapping maintains momentum conservation and keeps the mapped partons on-shell

$$p_i + p_j + p_k = p_I + p_K\,,$$
$$p_I^2 = p_K^2 = 0\,. \tag{2.5.5}$$

Below we list the antenna functions relevant for multijet production in electron-positron annihiliation with a single quark flavour, namely the partonic channel

(a) Quark-antiquark antenna, A_3 (b) Quark-gluon antenna, D_3

(c) Gluon-gluon antenna, F_3
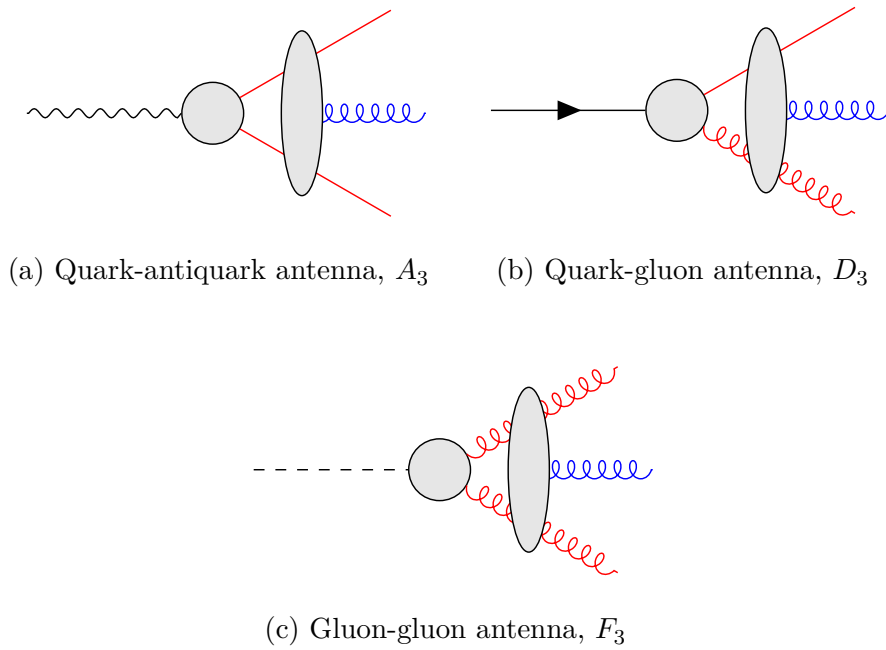
Figure 2.5: Three-parton antenna functions where the loop order is implicitly illustrated by the grey circle, and the grey ellipse represents all diagrams that give rise to the given external states. The hard radiators are depicted in red, where as the unresolved parton is coloured in blue. External fermion lines have been drawn without an arrow to depict the equivalence of quark and antiquark antenna functions.

$e^- e^+ \rightarrow q\bar{q} + n\,g$. The emulation of the NLO QCD K-factors of this partonic process is examined in Chapter 6. For this process we require the A, D, and F antenna functions, which are illustrated in Figure 2.5. The i, j, k indices will be fixed such that $q = 1$, $\bar{q} = 2$, and $g \in \{3, 4, 5\}$ in view of using the antenna functions where the momenta of partons will have fixed indices in a phase-space point represented as an array.

It will be useful to define the $\mathcal{P}oles$ and $\mathcal{F}inite$ operators to extract the singular and finite contributions from an antenna function. However, there are still finite terms from $\mathcal{P}oles(X)$ coming from the ϵ-expansion of the IR singularity operators given below. Therefore, care has to be taken to extract all the finite remainders from the antenna functions when trying to use them as basis functions for emulating the finite part of the one-loop matrix element.

For the one-loop antenna functions, there are three different functions corresponding to the: leading colour, subleading colour, and closed quark look contributions. It is sufficient to consider the leading colour structure in this thesis as the remaining contributions are subleading. Ignoring subleading terms in colour is equivalent to taking the large N_c limit. For QCD where $N_c = 3$, neglecting subleading colour terms amounts to an approximately 10% effect.

The antenna functions have some functions in common, namely the singularity operators

$$
\begin{aligned}
\mathbf{I}_{q\bar{q}}^{(1)}(\epsilon, s_{q\bar{q}}) &= -\frac{e^{\epsilon\gamma}}{2\Gamma(1-\epsilon)}\left[\frac{1}{\epsilon^2} + \frac{3}{2\epsilon}\right]\mathrm{Re}(-s_{q\bar{q}})^{-\epsilon}\,, \\
\mathbf{I}_{qg}^{(1)}(\epsilon, s_{qg}) &= -\frac{e^{\epsilon\gamma}}{2\Gamma(1-\epsilon)}\left[\frac{1}{\epsilon^2} + \frac{5}{3\epsilon}\right]\mathrm{Re}(-s_{qg})^{-\epsilon}\,, \\
\mathbf{I}_{gg}^{(1)}(\epsilon, s_{gg}) &= -\frac{e^{\epsilon\gamma}}{2\Gamma(1-\epsilon)}\left[\frac{1}{\epsilon^2} + \frac{11}{6\epsilon}\right]\mathrm{Re}(-s_{gg})^{-\epsilon}\,, \\
\mathbf{I}_{g\bar{q}}^{(1)}(\epsilon, s_{g\bar{q}}) &= \mathbf{I}_{qg}^{(1)}(\epsilon, s_{g\bar{q}})\,,
\end{aligned}
\tag{2.5.6}
$$

and dilogarithms which are embedded in the function

$$R(y, z) = \log(y)\log(z) - \log(y)\log(1-y) - \log(z)\log(1-z)$$
$$+ \frac{\pi^2}{6} - \text{Li}_2(y) - \text{Li}_2(z).$$

(2.5.7)

It will also be useful to introduce the variables $y_{ij} = s_{ij}/s_{ijk}$.

2.5.1   Quark-antiquark antenna functions

The quark-antiquark antenna functions are derived by normalising the colour-ordered QCD radiative corrections to $\gamma^* \to q\bar{q}$ at NNLO.

Three-parton tree-level antenna function

$$A_3^0(1_q, 3_g, 2_{\bar{q}}) = \frac{1}{s_{123}}\left(\frac{s_{13}}{s_{23}} + \frac{s_{23}}{s_{13}} + 2\frac{s_{12}s_{123}}{s_{13}s_{23}}\right) + \mathcal{O}(\epsilon)$$

(2.5.8)

Three-parton one-loop antenna function

$$\mathcal{P}oles\left(A_3^1(1_q, 3_g, 2_{\bar{q}})\right) = 2\left(\mathbf{I}_{qg}^{(1)}(\epsilon, s_{13}) + \mathbf{I}_{qg}^{(1)}(\epsilon, s_{23})\right.$$
$$\left. - \mathbf{I}_{q\bar{q}}^{(1)}(\epsilon, s_{123})\right) A_3^0(1_q, 3_g, 2_{\bar{q}}),$$

(2.5.9)

$$\mathcal{F}inite\left(A_3^1(1_q, 3_g, 2_{\bar{q}})\right) = -\left(R(y_{13}, y_{23}) + \frac{5}{3}\log y_{13} + \frac{5}{3}\log y_{23}\right) A_3^0(1_q, 3_g, 2_{\bar{q}})$$
$$+ \frac{1}{s_{123}} + \frac{s_{12} + s_{23}}{2s_{123}s_{13}} + \frac{s_{12} + s_{13}}{2s_{123}s_{23}}$$
$$- \frac{s_{13}}{2s_{123}(s_{12} + s_{13})} - \frac{s_{23}}{2s_{123}(s_{12} + s_{23})}$$
$$+ \frac{\log y_{13}}{s_{123}}\left(2 - \frac{1}{2}\frac{s_{13}s_{23}}{(s_{12} + s_{23})^2} + 2\frac{s_{13} - s_{23}}{s_{12} + s_{23}}\right)$$
$$+ \frac{\log y_{23}}{s_{123}}\left(2 - \frac{1}{2}\frac{s_{13}s_{23}}{(s_{12} + s_{13})^2} + 2\frac{s_{23} - s_{13}}{s_{12} + s_{13}}\right).$$

(2.5.10)

2.5.2   Quark-gluon antenna functions

The quark-gluon antenna functions are obtained by considering the QCD real radiation corrections in the process $\tilde{\chi} \to \tilde{g}g$.

Three-parton tree-level antenna function

$$D_3^0(1_q, 3_g, 4_g) = \frac{1}{s_{134}^2} \left(\frac{2s_{134}^2 s_{14}}{s_{13}s_{34}} + \frac{2s_{134}^2 s_{13}}{s_{14}s_{34}} + \frac{s_{14}s_{34} + s_{34}^2}{s_{13}} \right.$$
$$\left. + \frac{s_{13}s_{34} + s_{34}^2}{s_{14}} + \frac{2s_{13}s_{14}}{s_{34}} + 5s_{134} + s_{34} \right) + \mathcal{O}(\epsilon).$$

$$(2.5.11)$$

Three-parton one-loop antenna functions

$$\mathcal{P}oles\left(D_3^1(1_q, 3_g, 4_g)\right) = 2\left(\mathbf{I}_{qg}^{(1)}(\epsilon, s_{13}) + \mathbf{I}_{qg}^{(1)}(\epsilon, s_{14}) + \mathbf{I}_{gg}^{(1)}(\epsilon, s_{34})\right.$$
$$\left. - 2\mathbf{I}_{qg}^{(1)}(\epsilon, s_{134})\right) D_3^0(1_q, 3_g, 4_g),$$

$$(2.5.12)$$

$$\mathcal{F}inite\left(D_3^1(1_q, 3_g, 4_g)\right) = -\left(R(y_{13}, y_{34}) + R(y_{14}, y_{34}) + R(y_{13}, y_{14}) + \frac{5}{3}\log y_{13}\right.$$
$$\left. + \frac{5}{3}\log y_{14} + \frac{11}{6}\log y_{34}\right) D_3^0(1_q, 3_g, 4_g) + \frac{1}{3s_{34}}.$$

$$(2.5.13)$$

2.5.3   Gluon-gluon antenna functions

The gluon-gluon antenna functions are obtained from the QCD real radiation corrections in the process $H \to gg$.

Three-parton tree-level antenna function

$$F_3^0(3_g, 4_g, 5_g) = \frac{2}{s_{345}^2} \left(\frac{s_{345}^2 s_{34}}{s_{35}s_{45}} + \frac{s_{345}^2 s_{35}}{s_{34}s_{45}} + \frac{s_{345}^2 s_{45}}{s_{34}s_{35}} \right.$$
$$\left. + \frac{s_{34}s_{35}}{s_{45}} + \frac{s_{34}s_{45}}{s_{35}} + \frac{s_{35}s_{45}}{s_{34}} + 4s_{345} + \mathcal{O}(\epsilon) \right).$$

$$(2.5.14)$$

Three-parton one-loop antenna functions

$$\mathcal{P}oles\left(F_3^1(3_g, 4_g, 5_g)\right) = 2\left(\mathbf{I}_{gg}^{(1)}(\epsilon, s_{34}) + \mathbf{I}_{gg}^{(1)}(\epsilon, s_{35}) + \mathbf{I}_{gg}^{(1)}(\epsilon, s_{45})\right.$$
$$\left. - 2\mathbf{I}_{gg}^{(1)}(\epsilon, s_{345})\right) F_3^0(3_g, 4_g, 5_g),$$

$$(2.5.15)$$

$$\mathcal{F}inite\left(F_3^1(3_g, 4_g, 5_g)\right) = -\left(R(y_{34}, y_{35}) + R(y_{35}, y_{45}) + R(y_{34}, y_{45})\right.$$
$$\left. + \frac{11}{6}\log y_{34} + \frac{11}{6}\log y_{35} + \frac{11}{6}\log y_{45}\right) F_3^0(3_g, 4_g, 5_g)$$

$$+ \frac{1}{3s_{34}} + \frac{1}{3s_{35}} + \frac{1}{3s_{45}} + \frac{1}{3s_{345}} \, . \tag{2.5.16}$$

2.5.4   Limiting behaviour of antenna functions

Examining (2.5.8) more closely, it becomes clear that the correct IR behaviour is exhibited by the antenna function. The first two terms correspond to the gluon going collinear with the quark and antiquark. The final term encapsulates singularity in the limit of the gluon going soft.

The other antenna functions have similar limiting behaviour, collapsing to the single collinear and soft splitting functions in the relevant regions of phase-space.

Utilising this property of the antenna functions, and the fact that they naturally interpolate between the soft and collinear limits due to being derived from physical matrix elements, they can be used as a set of basis functions for approximating matrix elements. This will be explored in Chapter 6.

Chapter 3

Monte Carlo Event Generators

At particle collider experiments, the collisions between incoming particles can lead to highly complex final-states with a large number of particles. In a hadron collider, such as the LHC, the particles that we observe are not the constituents of the proton, the quarks and gluons. Instead we observe hadrons which are the bound states of these partons due to the property of confinement at low energy scales.

In order to describe the entire chain of events from the initial energetic collision to the production of the lower energy hadrons, it is customary to split up the process into stages characterised by their kinematic regime. The standard tool used to simulate these stages of an event are general-purpose Monte Carlo event generators (event generators), see for example Reference [79] for a review. They are indispensable tools that help bridge the gap between theoretical predictions and experimental measurements with nearly all analyses made at the ATLAS and CMS experiments involve some form of event generator usage. General-purpose event generators available such as Herwig [80, 81], Pythia [82, 83], and Sherpa [84, 85] all broadly share the same features at a high-level and it is common to utilise many different event generators in one analysis to get an estimate of the uncertainty due to the different approaches from each collaboration.

In the following we will motivate the usage of Monte Carlo methods and give a brief overview of the main stages involved in simulating an event. We will give

particular attention to the concept of event unweighting which is required to compare probabilistically generated events to experimental results.

3.1   Monte Carlo integration

As seen in Section 1.4, the task of computing partonic cross-sections boils down to integrating the matrix elements over the final-state phase-space. Whilst it is possible to carry out these integrals analytically for the simplest cases, the integral quickly becomes intractable when the final-state multiplicity becomes large. For an n-body final-state the dimensionality of the integral scales as $d = 3n - 4$, where momentum conservation and on-shell conditions have already been applied to reduce the dimensionality. Therefore, instead of carrying out the integrals analytically, numerical methods become more attractive. The current prevailing method is Monte Carlo integration which has no scaling with the dimensionality of the integral, so can be used for arbitrarily high multiplicity final-states. The purpose of this chapter is to give a brief overview of event generators, a more indepth treatment of the specifics of Monte Carlo methods can be found in, for example, Reference [86].

The basic idea of Monte Carlo integration is to approximate the integral of a function $f(\vec{x})$ by sampling it at d-dimensional parameter space points $\vec{x}$

$$I = \int_V d^D x \, f(\vec{x}) \approx \langle I(f) \rangle_x = V \frac{1}{N} \sum_{i=1}^{N} f(\vec{x}_i) = V \langle f \rangle_x \,, \qquad (3.1.1)$$

where the points $\vec{x}$ are randomly (hence Monte Carlo), uniformly sampled in the volume V. $\langle f \rangle_x$ denotes the average of f over these uniformly sampled points. The law of large numbers ensures that the estimator $\langle I(f) \rangle_x$ will converge to the true value I for $N \to \infty$. As a result of the central limit theorem, the function evaluations $f(\vec{x}_i)$ will follow a Gaussian distribution with mean $\langle I(f) \rangle_x$, meaning an error esimate on the estimator is given by the standard error of the mean

$$\sigma_I = V \sqrt{\frac{\langle f^2 \rangle_x - \langle f \rangle_x^2}{N}} \qquad (3.1.2)$$

which falls as $1/\sqrt{N}$, with no dependence on d.

An algorithm that samples points $\vec{x}$ is called an integrator. To make use of the random numbers sampled, they need to be mapped to quantities that are useful in a simulation setting. The most commonly used mapping is to map the random numbers to four-momenta. An example of a simple integrator is the RAMBO algorithm [87,88] which samples four-momenta space uniformly. Since the integrator now effectively samples in momentum space, we refer to the points generated as phase-space points and so the terms integrator and phase-space generator are used interchangeably.

Whilst Monte Carlo integration is independent of the dimensionality of the integral, meaning convergence is guaranteed for large enough N, there are variance reduction methods to more rapidly reduce the error on the estimator.

One such method is importance sampling which aims to reduce the variance via a remapping of the uniformly sampled variables $\vec{x}$ to a more suitable non-uniform distribution. In the context of particle physics, this corresponds to using a mapping of random numbers to four-momenta that captures the distribution of the matrix elements and their divergences, i.e. samples more frequently in regions where the matrix element is large in magnitude. Integrators inspired by the pole structure of multi-parton QCD processes have been studied in References [89,90].

Another common variance reduction technique is stratified sampling which divides the integration volume into sub-volumes, or bins. The overall integral and variances are the sums of the partial results in each bin. The overall variance is minimised when the contributions from each bin are equal. This is achieved by sampling more from bins where the integral is rapidly fluctuating (and not necesarily large), and less from bins where the integral has less fluctuations.

In practice, state-of-the-art event generators combine the concepts of importance sampling and stratified sampling with adaptive sampling [91,92] and multi-channelling methods [93,94] to achieve faster rates of convergence.

3.2   Event unweighting

Once a suitable configuration of momenta has been found by the integrator, this point forms the basis of an event. The matrix element of interest can be evaluated at this phase-space point at the desired order in perturbation theory (see Section 3.3). Additionally, there will be an associated Jacobian from the mapping from random variables to the four-momenta. This Jacobian is referred to as the phase-space weight, which together with the matrix element forms the event weight[1]

$$w_i = |\mathcal{M}|_i^2 J_i \,. \tag{3.2.1}$$

The event weight can be interpreted as the probability of this particular event occurring at an experiment [95]. By generating a number of events in this manner, and histogramming them, a sample of weighted events is created.

At experiments, measured events are either recorded or they are not, i.e. events come with unit-weight, not a variable weight in weighted events. In order to generate events that compare to reality, it has become customary to translate weighted events into unit-weight events. This process is known as unweighting and is typically carried out using a rejection sampling algorithm outlined in Algorithm 1. The rejection sampling algorithm makes use of w_{max} which is the maximal event weight in the integration volume.

Algorithm 1: Rejection sampling for unweighting events

    while *unweighting* **do**
        generate random phase-space point $\vec{x}$;
        evaluate event weight $w \leftarrow w(\vec{x})$;
        generate uniform random number $r \leftarrow \mathrm{Random}(0,\ 1)$;
        if $w/w_{\mathrm{max}} > r$ **then**
            return $\vec{x}$ and $\tilde{w} \leftarrow \max(1,\ w/w_{\mathrm{max}})$;
        end
    end

From this description, it is clear that the number of accepted events will generally

---
[1]For the more sophisticated sampling methods employed in event generators there will be additional weights involved.

be lower than the number of total events generated. This is encapsulated in the unweighting efficiency

$$\varepsilon = \frac{N_{\mathrm{accepted}}}{N_{\mathrm{total}}} \, , \qquad (3.2.2)$$

whose inverse is the average number of trial events required to obtain a single unweighted event. Unweighting efficiency for high-multiplicity processes are often well below 1% [96, 97] and development of methods to improve the efficiency have been explored [98, 99]. These low efficiencies lead to most of the yearly event generator CPU budgets of LHC experiments to be spent on unweighting events [100].

An alternative to improving the unweighting efficiency itself is to accelerate the process of unweighting itself. This was explored in Reference [101] where a novel two-stage unweighting procedure employing a surrogate model of the exact event weight was used to decrease the amount of time spent unweighting events. This idea will be elaborated on in Chapter 7 where we build upon Reference [101] by increasing the fidelity of the surrogate model.

3.3 Anatomy of an event

The simulation of an event can be split up into three main stages: the hard scattering process, followed by the parton shower, and finally hadronisation. This is motivated by the factorisation theorem which separates the high energy regimes from the low energy regimes.

Hard scattering

The first stage of an event simulation begins with the computation of the matrix element. The partonic channel of interest sets the multiplicity of the process, therefore informing the integrator on how to sample the phase-space accordingly.

As seen in (1.4.9), the exact matrix element is approximated by a perturbative series in a coupling constant. By now it is standard to have at least the LO and

NLO contributions in this expansion. The calculations of tree-level [102–107] and one-loop [108–121] matrix elements are now largely fully automated in NLO QCD and electroweak corrections. Many of the modern matrix element generators are supplied with a common generic interface to event generators, the Binoth Les Houches Accord [122, 123]. For more information on the machinery developed to compute nearly arbitrary matrix elements at tree-level and one-loop level, see for instance Reference [124]. NNLO and N³LO QCD corrections have been computed for select processes but have yet to be reach the level of automation of NLO. For a review of recent advancements see References [125, 126].

Matrix element calculations make up a large portion of the total event generation time [127]. Reducing the time spent in this stage of the event generation therefore represents a sizeable increase in the number of events generated given a fixed computing budget [100]. This thesis presents a novel technique to accelerate matrix element calculations by building accurate emulation models from modern machine learning techniques.

Parton shower

Fixed-order calculations of matrix elements are limited to relatively few final-state particles due to the complexity of analytical expressions and their phase-space integrals. They are also only accurate in the high energy limit. To evolve from the fixed-order calculation to the non-perturbative regime where hadrons are formed, parton showers are employed in event generators.

Parton showers are implemented as algorithms that iteratively emit radiation (gluons and quarks in QCD parton showers) described by splitting functions. By repeating this procedure of emissions, parton showers produce a cascade of partons that naturally evolves the energy scale from the hard process to an infrared scale where non-perturbative effects begin to set in. In doing so parton showers account for potentially large logarithms that arise in these kinematic regimes at all orders in the

coupling parameter, this is known as resummation [79].

In event generators, fixed-order matrix elements and parton showers have to be utilised together to describe the many different observables at experiments. In order to combine the two stages of event generation in a consistent manner there are matching [128–130] and merging [131, 132] schemes.

Hadronisation

The formation of hadrons in the low energy regime of an event generator follow non-perturbative models. The two main classes of models used to simulate hadronisation are the cluster models [133, 134] and string models [135]. These models share a similar concept of grouping together quarks and antiquarks to ensure only hadrons remain in the final-state. For more discussion on hadronisation see for example Reference [136].

3.4 A note on theoretical uncertainties

The partonic cross-section, $\hat{\sigma}(\mu_F, \mu_R)$ has a dependence on both the factorisation scale and the renormalisation scale. Although these scales are unphysical, and so any observable should not depend on them, there is a residual dependence on them in fixed-order calculations due to missing higher order terms.

The most widely adopted approach to estimate the theoretical uncertainties associated with these missing higher order terms is to carry out a seven-point scale variation. Typically in a calculation we take $\mu = \mu_F = \mu_R$ where μ depends on the specifics of the process of interest. Taking the value of the partonic cross-section at this scale as the central value, the seven variations correspond to multiplying (μ_F, μ_R) by the factors

$$S \in \{(1, 1), (1/2, 1/2), (2, 2), (1/2, 1), (1, 1/2), (2, 1), (1, 2)\} , \qquad (3.4.1)$$

and re-evaluating the cross-section at these scales. The estimate of the uncertainty is then taken to be half the width of the maximum interval.

Chapter 4

Modern machine learning algorithms and their uses

4.1 Rise of machine learning in HEP

In high energy physics where analyses depend on making use of large computing clusters and international grid efforts [137], advancements in technology and more efficient methods are required to keep up with the needs of the experimental and theoretical community. However, the computational resources needed are at risk of outpacing the growth in these research areas [127], especially with HL-LHC over the horizon. Therefore new techniques and algorithms from the machine learning community have been adopted to augment the ongoing efforts in the high energy physics community. Machine learning is a category of artificial intelligence concerned with the design of algorithms that automates learning from data. By now machine learning is utilised throughout every stage of particle physics analysis, from collection of data from experiments all the way to novel applications on the theory side.

Around the turn of the last decade the culmination in advancement in hardware and techniques led to an explosion in image classification research in the machine learning community, specifically in the utilisation of very large neural networks. The

accuracy of these deep neural networks greatly outperformed the previous state-of-the-art [138,139] and lead to the widespread use of neural networks for a wide array of tasks [140]. This led to the rise of programming frameworks such as scikit-learn [141], TensorFlow [142], PyTorch [143], and XGBoost [144] to name a few which implement neural networks, alongside other mainstay algorithms such as decisions trees. These algorithms are now extremely commonplace in many domains, including in particle physics.

Collisions at the LHC can produce hundreds of particles with complex final-state configurations leading to the design of the ATLAS and CMS experiments to contain of the order 100 million detection elements in an attempt to disentangle these events. With these large arrays of detection elements there is an enormous amount of data harvested from each collision, terabytes per second, which presents a challenge in the form of data collection. Already at this stage, boosted decision trees have been used to enhance triggers [145] to accept or reject data from being saved to disk. At the LHCb experiment 70% of all data retained have had some involvement with machine learning algorithms [146].

On the theory side of things, machine learning has been applied to many stages of the event generation process. To name a few examples of active research areas: phase-space sampling, matrix element modelling, hadronisation modelling, and anomaly searches for new physics. The research carried out in these areas have shown promising results but a challenge that remains is to interface these novel machine learning methods to existing event generators. A living review that is actively being updated archiving advancements in the field is available at [147]. For more traditional reviews, see for instance [148–150].

In this chapter we give an overview of neural networks: how they are constructed, how they are trained, and how their parameters are optimised. We will then proceed to discuss their application to modelling matrix elements with a review of the current state-of-the-art methods.
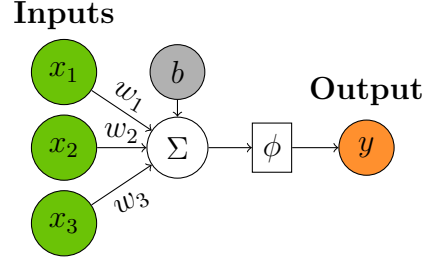
Figure 4.1: A schematic diagram of a neuron. The model inputs $\mathbf{x}$ and weights $\mathbf{w}$ are vectors, where as the bias b is a scalar. The weighted inputs are summed with the bias before passing through an activation function to give the scalar output y.

4.2 Neural networks

Neural networks are a machine learning algorithm inspired vaguely by the structure of the human brain. That is to say they are an interconnected network of neurons used for analysing data. Here we will introduce the densely-connected neural network which will be the main workhorse algortihm used in this thesis, and frame the discussion around a regression task. Namely, how to use sampled training data from a target function $g(\mathbf{x})$ to create a surrogate model with the neural network algorithm.

4.2.1 Model of a neuron

In this context a neuron is modelled by

$$y = \phi(z) = \phi(\mathbf{w}^T\mathbf{x} + b) \tag{4.2.1}$$

where the model inputs $\mathbf{x}$ are multiplied by the model weights $\mathbf{w}$ before being summed with a bias term b. Collectively the model weights and biases are the model parameters θ. This combination of terms is then modified by an activation function ϕ which is carefully chosen to perform a non-linear transformation. A neuron is illustrated in Figure 4.1.

Some of the more well-known activation functions are the sigmoid function, hy-

Activation	Functional form	Gradient
Linear	$f(x) = x$	$f'(x) = 1$
Sigmoid	$f(x) = \dfrac{1}{1 + e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
Tanh	$f(x) = \tanh(x)$	$f'(x) = 1 - f(x)^2$
ReLU	$f(x) = \begin{cases} 0, & \text{for} \quad x \leq 0 \\ x, & \text{for} \quad x > 0 \end{cases}$	$f'(x) = \begin{cases} 0, & \text{for} \quad x \leq 0 \\ 1, & \text{for} \quad x > 0 \end{cases}$ [1]
swish	$f(x) = \dfrac{x}{1 + e^{-x}}$	$f'(x) = \dfrac{1}{x} f(x) \left[(1 + x) - f(x) \right]$

Table 4.1: Functional forms of common activation functions and their gradients.

perbolic tangent, rectified linear unit (ReLU). A less familiar class of activation functions are the sigmoid linear units (SiLU) of which swish is an example, which have been shown to outperform ReLU is some cases [151]. The functional forms of these functions are summarised in Table 4.1. There is no single activation function that is the best for any given task, and it is common practice to choose activation functions on a case-by-case basis.

4.2.2   Densely-connected neural networks

In order to build a neural network, the neurons have to be connected in some fashion. Perhaps the most straightforward method is to create layers of neurons, and then connect every neuron in a layer with every neuron in neighbouring layers. This is illustrated in Figure 4.2. Such a configuration is called a fully-connected, or densely-connected neural network. The constituents of this network structure are the input layer where model inputs enter, hidden layers where the bulk of model parameters live, and the output layer which outputs the model prediction(s). The training data only provides concrete desired outputs for the overall model, and not for the layers before the output layer. This is why mediating layers between inputs and outputs are

---

[1] The gradient of ReLU is not defined at $x = 0$ but for a numerical implementation, defining it to be 0 at this point is sufficient.

Input Layer $\in \mathbb{R}^3$ Hidden Layer $\in \mathbb{R}^5$ Hidden Layer $\in \mathbb{R}^5$ Output Layer $\in \mathbb{R}^1$
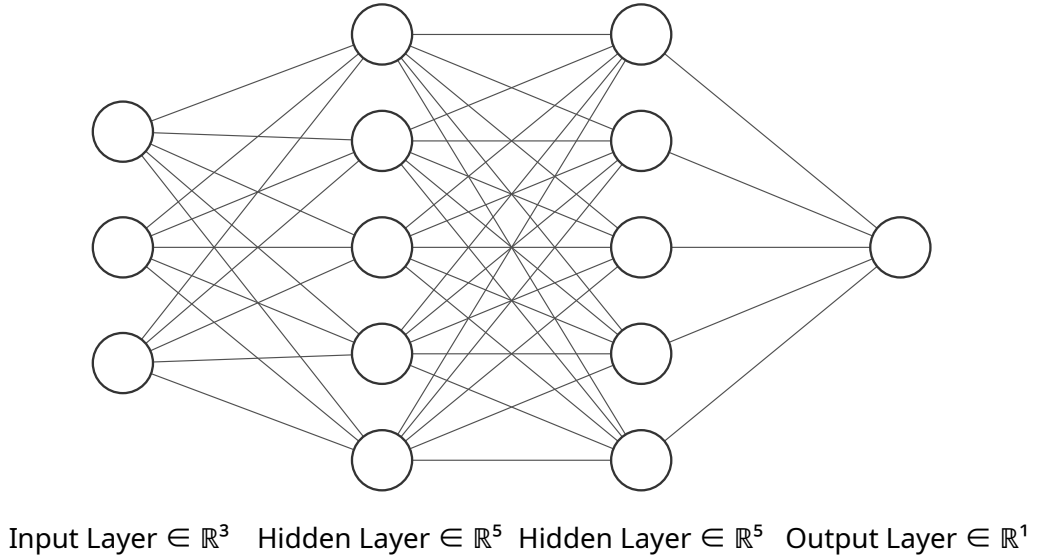
Figure 4.2: An example of a densely-connected neural network with two hidden layers. Each node on this image represents a neuron and the connections between then represents the inputs/outputs to neighbouring neurons.

so-called hidden. The neural network is free to change the parameters in these layers to best approximate the target function $g(\mathbf{x})$. Output layers are functionally the same as hidden layers, except that their outputs are taken as the model prediction and so can be compared with the truth value.

The depth of a neural network is generally denoted by the number of hidden layers, or alternatively, the number of activation functions between the input and output layers. Similarly, the width of a neural network is generally referring to the number of nodes in the hidden layers.

The output of a densely-connected neural network can be formulated by repeatedly applying (4.2.1) to give

$$f(\mathbf{x};\theta) = \boldsymbol{\phi}^{(n)}(\ldots \boldsymbol{\phi}^{(2)}(\, \boldsymbol{\phi}^{(1)}(\mathbf{x};\theta^{(1)})\, ;\theta^{(2)})\, ;\theta^{(n)})\, , \qquad (4.2.2)$$

where $\boldsymbol{\phi}^{(n)}$ and $\theta^{(n)}$ denotes the outputs and parameters of layer n, where the output layer is included in this notation. By chaining together activation functions, and especially non-linear activation functions, neural networks become good function

approximators

$$f(\mathbf{x}; \theta) \approx g(\mathbf{x}) \,. \tag{4.2.3}$$

The universal approximation theorem states that neural networks are able to represent any continuous function $g(\mathbf{x})$ with one hidden layer and a finite number of neurons [152]. However it is very difficult to achieve this due to practical constraints such as limited data and network size. It is much more common to link together a larger number of hidden layers, leading to deep neural networks since better learning algorithms have been found for this case [153].

4.2.3   Loss functions and optimisation of parameters

To summarise so far, neural networks are a vast network of connected nodes with a large number of parameters that can be tuned for the problem at hand. The task of optimising the parameters is the main challenge in training a neural network and is a key area of research.

In order to quantify the performance of a network for a given task it is useful to define a loss function. For regression tasks, such as approximating a function, a commonly used loss function is the mean squared error

$$L_{\mathrm{MSE}} = \frac{1}{N} \sum_{i=0}^{N} (g(\mathbf{x}_i) - y(\mathbf{x}_i; \theta))^2 \,, \tag{4.2.4}$$

where $g(\mathbf{x}_i)$ and $f(\mathbf{x}_i; \theta)$ are samples from the target distribution, and model predictions for inputs $\mathbf{x}_i$ belonging to a data set with N samples. The loss function encodes the discrepency between the truth value and the model prediction, therefore, the task of finding an acceptable set of parameters can be reframed as an optimisation of the loss function. It should be noted that minimisation of the loss function is simply a proxy for maximising neural network predictive power. Due to the training and testing datasets being finite, the generalisation of the neural network is not guaranteed for the true underlying distribution from which the datasets are sampled from. This problem is generally referred to as overfitting. We will refer back to this

problem in the context of fitting matrix elements in Section 4.3.

Since neural networks are generally aimed at learning non-linear functions, the loss surface corresponding to (4.2.4) is a function of many variables, and is likely highly non-convex with many local minima. The methods of choice for traversing these loss surfaces are all iterative gradient-based methods with modifications to improve convergence. Broadly speaking, the algorithms iteratively update the neural network parameters based on the gradient of the loss with respect to these parameters with some step size (or learning rate) η. That is

$$\theta = \theta - \eta \cdot \nabla_\theta L(\mathbf{x}_{\text{batch}}; \theta) , \qquad (4.2.5)$$

where $\nabla_\theta L(\mathbf{x}_{\text{batch}}; \theta)$ is the gradient of the loss with respect to the parameters of the model, averaged over the batch of inputs $\mathbf{x}_{\text{batch}}$. The initial state of the parameters are usually distributed according to normal or uniform distributions [154, 155]. This variation of gradient descent is referred to as mini-batch gradient descent because the gradient $\nabla_\theta L$ is averaged over a mini-batch of samples $\mathbf{x}_{\text{batch}}$. Mini-batch gradient descent strikes a balance between stochastic gradient descent (estimating gradient with one sample at a time), and batch gradient descent (calculating gradient with respect to entire training set) by having an efficient estimation of the gradient that is accurate enough for practical applications. In modern machine learning libraries, the updating of weights in (4.2.5) is carried out via matrix multiplications which are highly efficient on graphics-processing units (GPUs), and the gradients are computed numerically with the aid of automatic differentiation tools. For these reasons, deep neural networks have become much more widespread due to the ease of constructing and training them, as well as their good predictive performance for general problems.

The basic gradient descent algorithm described above is the basis upon which many of the most widely adopted optimisers [156–158] are based on. For an overview of these optimisers, see for instance Reference [159].

4.2.4   Optimisation of hyperparameters

Another set of parameters that need to be optimised are the model hyperparameters. Hyperparameters are parameters that are not explicitly parameters of the model, instead they control the speed and quality of the training process. For instance, the learning rate in (4.2.5) is an important hyperparameter whose initial value has to be chosen carefully to observe a good rate of convergence. Some other examples would be the number of hidden layers, the choice of activation function, or the mini-batch size.

The choice of hyperparameters is an important factor during the training process as a suboptimal set of hyperparameters can perform significantly worse than a more carefully chosen set. The process of tuning hyperparameters is often computationally expensive for two reason. Firstly the dimensionality of parameter space to choose from is high, and so the curse of dimensionality kicks in rapidly. Secondly there is no exact a priori way to evaluate the final performance of the neural network given a set of hyperparameters meaning training has to take place for some steps before the final model performance can be estimated.

Some of the simplest methods of hyperparameter optimisation are grid search and random search. Grid search is simply an exhaustive scan of parameters systematically sampled in a grid-like fashion, where as random search replaces this systematic sampling with random sampling. Whilst these methods are quick to implement, the number of trials to land on a good set of hyperparameters could potentially be very large. Methods which aim to reduce the number of trials have been proposed in the literature, see for example Reference [160] for a review. Below we discuss two paradigms of hyperparameter optimisation that aim to reduce number of trials before landing on a set of decent hyperparameters.

Bayesian optimisation methods have been explored to carry out hyperparameter optimisation [161, 162]. The method begins with the construction of a probabilistic surrogate model for the performance of the neural network. This surrogate model

collects evidence through successive trainings of the neural network. By selecting promising hyperparameter sets based on the updated model, the exploration of hyperparameter space is much more efficient.

Another group of methods are based on the principle that the first few epochs of training are indicative of the final model performance. This in combination with a fixed time budget gives rise to multi-armed bandit strategies [163, 164] that allocate more budget to better performing models whilst simultaneously dropping poorly performing models. By iterating this procedure only the best performing models remain. Recently, there have been proposals of combining Bayesian optimsation and the bandit-based approaches [165].

In practice the area of hyperparameter optimisation is still extremely computationally expensive even with the more advanced methods. Given a sensible set of hyperparameters, one has to balance the additional time spent tuning with the relatively incremental improvements in performance. For novel machine learning experiments in high energy physics, the hyperparameter tuning is not of paramount importance, and generally more focus is spent on the model building itself.

4.3 Neural networks as matrix element surrogate models

Motivated by the ability of neural networks to model non-linear functions, as well as the rise in GPU usage in high energy physics, it seems natural to leverage neural networks to build surrogate models for matrix elements. There are multiple advantages to this: the time taken to evaluate matrix elements remains a bottleneck in event generation, and so building a fast surrogate model would accelerate this process; once a fast and accurate surrogate model has been built it can supplement the traditional matrix element generator to increase statistical power of the sample by predicting on many more phase-space points. This point requires the accumulated error in the

surrogate model predictions to be lower than the statistical error in the integration of matrix elements, which is a key motivation for building an accurate emulator.

The task of building a surrogate model for matrix elements is in principle straightforward. The dataset usually consists of phase-space kinematics in the form of four-momenta and the targets are the corresponding matrix elements for the process of interest, at the desired order in perturbation theory. As already discussed in Sections 3.1 and 3.3, phase-space samplers and matrix element generators are now widely available so the acquisition of data is only limited by time. Even then, this is a bottleneck only for the most expensive of processes as generation of data is trivial to parallelise. In the ideal case, the surrogate model would be faster than the matrix element generators, and sufficiently accurate. Neural network predictions are fast by virtue of them being simple matrix multiplications, and because they are predicted in batches, vectorisation is automatic. Therefore the difficulty in building a good surrogate model is in controlling the accuracy.

The complication in accurately emulating matrix elements on a per-point basis arises due to the infrared divergent behaviour as discussed in Section 2.1.4 which occurs in corners of the phase-space. (UV divergences have been removed via renormalisation). Outside of these soft and collinear limits, the matrix element is a well-behaved function that varies smoothly with phase-space. However, once these infrared regions are approached, small changes in the kinematics can lead to very large changes in the matrix element. This rapid response in the target function makes it difficult to accurately model the non-divergent regions simultaneously with the divergent regions due to the disparity in scales. This problem becomes more apparent for higher multiplicity final-states as there are more regions in phase-space for which the matrix element diverges.

Related to the problem of modelling matrix elements, in recent years there has been work in applying machine learning methods to learn: event weights [101], cross-sections [166, 167], analytic expressions of squared amplitudes [168], contour deformations for multi-loop integrals [169], and simplifying polylogarithms [170].

Emulation of matrix elements directly have been carried out in recent years in various projects. Bishara and Montull showed [171] that it was possible to build a boosted decision tree model to reproduce the matrix elements of the loop induced process $gg \to ZZ$ with good accuracy – below 0.1% errors for fully differential distributions. However, for this $2 \to 2$ process it was already demonstrated that training a single model on the entire phase-space sampled was suboptimal compared to training a number of regressors on subdomains of the phase-space.

This trend was observed in a future work by Aylett-Bullock and Badger [4] where the authors emulated tree-level matrix elements and NLO QCD K-factors for $e^+e^- \to q\bar{q}$ + up to 3 gluons. With a higher final-state multiplicity, controlling the infrared divergences became a crucial challenge. Their approach was to first partition phase-space into divergent and non-divergent regions, then subsequently partition the divergent regions further according to partition functions based on FKS subtraction. In each partition the authors trained a separate neural network, meaning that their model was an ensemble of neural networks. The performance of this method was shown to be an improvement over a single neural network trained on the unpartitioned phase-space, with good agreement in histogrammed distributions. However, the per-point agreement was lacking.

This FKS partitioning method was applied in Reference [172] for diphoton plus jet production, where the authors presented a use case of the emulator with a novel interface to the Sherpa event generator. This work was revisited in [173], where particular attention was paid to quantifying the uncertainty associated with the neural network prediction. This uncertainty was modelled with a Bayesian neural network which enabled the boosted training on regions of phase-space that were lacking in accuracy. Additionally, the authors showed that accuracy was improved with a more careful preprocessing of the training data compared to the previous treatment. However, both of these works showed that increasing multiplicity from $2 \to 3$ to $2 \to 4$ represented a large decrease in accuracy.

To summarise, the current state of matrix element emulation has to trade-off higher

multiplicity processes with accuracy. The research in this thesis attempts to tackle both of these problems by exploiting the factorisation property of matrix elements (Sections 2.2, 2.2.3, and 2.2.4), to build into the emulator the universal singular functions that describe the matrix elements in the soft and collinear limits (Sections 2.4 and 2.5). Since this method relies on the factorisation of matrix elements, we collectively refer to these family of models as factorisation-aware models. In Chapter 5, we show that by using Catani-Seymour dipoles as a set of basis functions, tree-level matrix elements for $e^+e^- \rightarrow q\bar{q} +$ up to 3 gluons can be emulated accurately to well below 1% accuracy, even for the highest multiplicity. Extending this study to the one-loop level is done in Chapter 6 where antenna functions replace Catani-Seymour dipoles as the basis functions.

The question of uncertainty of the neural network prediction is important as it contributes to the total theoretical uncertainty. An alternative to quantifying the uncertainty of the neural network prediction, is to bypass the usage of the neural network prediction itself in production. Instead it can be used as a proxy for the exact matrix element in event unweighting. This was explored in Reference [101], where it was shown that with a surrogate model of the event weight it was possible to use a double unweighting algorithm to increase the rate of event unweighting. In Chapter 7 we examine replacing the original emulator in said work with a factorisation-aware model to study the potential gain factors with a more accuracte model. As previously mentioned, a consequence of this two-stage unweighting algorithm is that the exact event weight has to be evaluated at some point, meaning the uncertainty of the surrogate model is practically irrelevant.

Chapter 5

Emulation of tree-level e^+e^- matrix elements

5.1 Introduction

Neural networks (NN) first appeared in the field of particle physics as a tool for event discrimination in the analysis of collider data. Since then neural networks and other machine learning techniques have proved useful in many other areas of the field. On the theory prediction side they have been used to improve the efficiency of Monte Carlo sampling [174–178], to accelerate the simulation of radiation within a jet [179–181], to streamline the processes of generation and unweighting of simulated event samples [97, 166, 171, 182–188]. Closer to the experimental measurements they have also been used to emulate detector simulation [189–192], they can be used to perform unfolding [193] or correcting for detector effects [194], and perform pileup subtraction [193].

The capacity of neural networks to approximate intricate functions have already been used to provide fast calculations of production cross-sections [166, 167]. In this article we investigate whether NNs can approximate production cross-sections more differentially by replacing computationally expensive matrix element calculations.

The challenge with this endeavour is that matrix elements are plagued with numerous divergences that arise from infrared divergences. In previous works [4, 172] a combination of individual neural networks were used to approximate matrix elements. In order to deal with the complex structure of the matrix elements the authors of these studies divided the phase-space into sectors according to the infrared singularities and trained networks on these sectors, thereby limiting the complexity of the fit by isolating a single divergence per sector. All sectors were then combined to make the final prediction. While the authors of the initial study for electron-positron annihilation showed good agreement between the total cross-section and histogrammed distributions both at LO and NLO, we note that the accuracy of the interpolation at the level of individual points was a lot worse than when averaged in the histograms. In addition, the performance of the extrapolation outside of its training phase-space (i.e. more singular configurations than those considered to fit the model) is problematic.

In this article we present a different approach to the emulation of matrix elements that incorporates the factorisation properties of the matrix elements in the interpolation model and therefore is able to safely extrapolate the matrix element in regions more singular than that covered by the data used for the training of the emulator. We find that our interpolator also displays a much improved pointwise accuracy.

The article is organised as follows. Section 5.2 introduces our factorisation-aware deep neural network model, Section 5.3 showcases our results with a comparison with the model from Ref. [4] in Section 5.3.1, and an analysis of the performance of our model in Section 5.3.2. To show that the neural network is both interpolating and extrapolating well we show its behaviour on random trajectories in phase-space in Section 5.3.3. Our conclusions are presented in Section 5.4 and some discussion of specific details are collected in the appendices so as not to distract from the main discussion.

We provide code to reproduce the methodology detailed in this article [195].

5.2 Fitting framework

For this work we consider the $e^+ e^- \to Z/\gamma^* \to q\bar{q} + ng$ matrix elements for n up to and including 3, which corresponds to events with up to 5 jets. We denote the number of jets in the final state as n_j.

We formulate the problem of emulating matrix elements as a supervised regression task with a set of phase-space points' kinematic information as input and the values of the matrix element for each of these phase-space points as the targets. Section 5.2.3 describes how these matrix elements were obtained.

A neural network can be seen as a function $f(\vec{x}; \theta) = \vec{y}$, where $f : \mathbb{R}^d \to \mathbb{R}$ maps a d-dimensional vector $\vec{x}$ of inputs onto the vector of outputs $\vec{y}$, and where θ are the parameters of the neural network which we aim to optimise such that the outputs $\vec{y}$ of the neural network match the target as well as possible. The simplest implementation of an emulator would be to take the input as the kinematic information of the phase-space point and the output to be the full matrix element. Our approach modifies both the input of the NN and its target, as described in the following sections.

5.2.1 Infrared divergences and dipole factorisation formula

It is well known that in soft and collinear limits the matrix element in $n + 1$-body phase-space factorises into a singular factor and a reduced matrix element in n-body phase-space [34, 196]. This factorisation was used by Catani and Seymour [50] to construct subtraction terms for the real radiation part of a NLO calculation. They introduced a factorisation formula with universal dipoles that smoothly interpolates between the soft and collinear limits to capture the singular structure in these regions of phase-space. The dipole factorisation formula can be written schematically as

$$|\mathcal{M}_{n+1}|^2 \to |\mathcal{M}_n|^2 \otimes \mathbf{V}_{ij,k}, \tag{5.2.1}$$

where $\mathbf{V}_{ij,k}$ is a process independent, singular factor. It depends on the momenta and quantum numbers (colour and spin) of partons i, j, k, where i is the emitter

parton, j is the emitted parton, and k is the spectator parton. For singly unresolved limits, this factorisation isolates all the divergent behaviour in $\mathbf{V}_{ij,k}$ and the factor $|\mathcal{M}_n|^2$ is free of divergences, which makes it more amenable to emulation through a neural network. The dipole factorisation formula forms the basis of our fitting ansatz which we present in detail in Section 5.2.2.

5.2.2 Fitting coefficients of Catani-Seymour dipoles

Instead of using a neural network to fit the matrix element directly, we use the dipole factorisation formula to build an ansatz of the colour and helicity summed $n + 1$-body matrix element,

$$\langle |\mathcal{M}_{n+1}|^2 \rangle = \sum_{\{ijk\}} C_{ijk} D_{ij,k} \,, \tag{5.2.2}$$

where $D_{ij,k} = \langle V_{ij,k} \rangle / s_{ij}$ are the spin-averaged Catani-Seymour dipoles divided by the corresponding Mandelstam invariant and C_{ijk} are the coefficients we train the neural network to fit. C_{ijk} can be interpreted as the reduced matrix element in n-body phase-space. Since the input for the C_{ijk} function is the full $n + 1$ phase-phase information, the neural network will also model the phase-space mappings usually introduced in the factorisation formula. A schematic diagram illustrating our ansatz is given in Figure 5.1. The sum over $\{ijk\}$ denotes the sum over relevant permutations of the external outgoing legs. More detail on this is given in Section 5.2.4. The representation (5.2.2) is not unique but through appropriate training, the neural network takes advantage of the right ingredients to model the divergent soft and collinear behaviour of the matrix elements.

This form of the ansatz allows the neural network to avoid fitting a rapidly varying function over the phase-space, leaving the Catani-Seymour dipoles to reproduce the correct singular behaviour, meaning a single neural network can interpolate a now relatively smooth function over the phase-space.
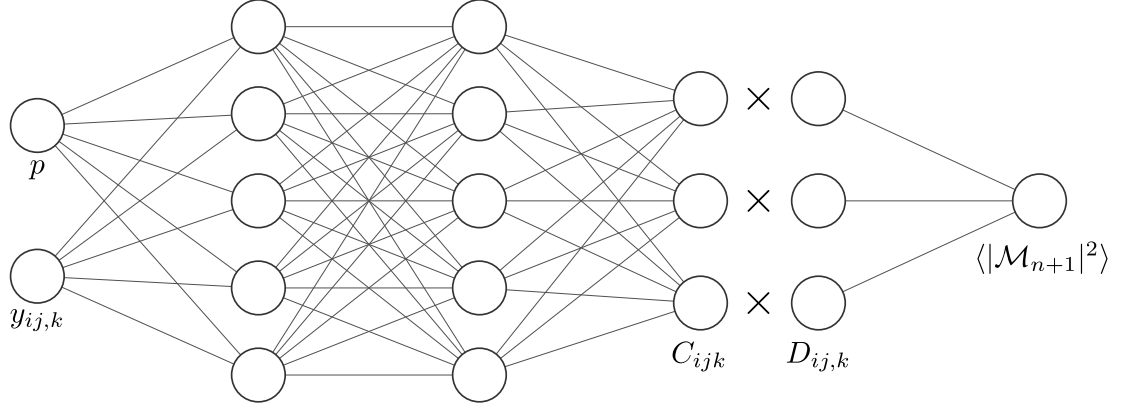
Figure 5.1: Schematic diagram of our neural network architecture. We have a densely-connected neural network with inputs phase-space points, p, and recoil factors, $y_{ij,k}$, propagated through hidden layers to the output layer which outputs C_{ijk}. These coefficients are combined with their corresponding spin-averaged dipoles as in (5.2.2) to produce an approximation of the matrix element. Diagram of neural network generated with the aid of [3].

5.2.3   Data generation

For all multiplicities, phase-space is sampled uniformly using the RAMBO algorithm [87] with a centre-of-mass energy $\sqrt{s_{\mathrm{com}}} = 1000$ GeV. Phase-space points are subsequently clustered using `FastJet` [197,198] with the e^+e^- k_t algorithm [199]. Global phase-space cuts are applied according to the criterion $y_{\mathrm{cut}} \leq y_{ij}$ where y_{ij} are the Mandelstam invariants normalised by s_{com}. Jets are clustered exclusively where d_{cut} was supplied to `FastJet`. We took $d_{\mathrm{cut}} = \max(2 \times y_{\mathrm{cut}}, 0.01 \times s_{\mathrm{com}})$. We explore three different values of the global phase-space cut parameter, $y_{\mathrm{cut}} = [0.01, 0.001, 0.0001]$, to demonstrate the ability of the factorisation-aware neural network to effectively interpolate in more and more singular regions of phase-space.

The generated phase-space points are fed to the NJET package [117] to calculate colour and helicity summed tree-level matrix elements. All external legs have been considered to be massless. The strong coupling constant has been set to $\alpha_s = 0.118$, and the electromagnetic coupling constant has been set to $\alpha_e = 1.0/132.5070$. The mass of the Z-boson is taken to be $m_Z = 91.188$ GeV. These parameters

and those not listed, are consistent with those in the Standard Model mode of
MADGRAPH5_AMC@NLO [113].

The phase-space points generated form the basis of our inputs to the neural network
with the matrix elements as our fitting targets. As with most machine learning
applications, we need to demonstrate that our neural network emulator has managed
to generalise outside of the training dataset. We do this by firstly testing on an
independent testing dataset that is never exposed to the network during training, and
secondly by predicting on random trajectories in phase-space. Generation of phase-
space trajectories is described in Appendix B.3. We believe that accurate predictions
on random phase-space trajectories demonstrates the ability of the neural network
to extrapolate to never before seen data that is of a different nature to both the
training and testing datasets.

5.2.4   Neural network emulator

We construct our emulator using a densely-connected neural network built using the
`Keras` API [200] with the `TensorFlow` back-end [142] with GPU support. A simple
model architecture such as a densely-connected neural network allows for quicker
training and inference compared to more complicated setups.

Inputs and outputs

Inputs to neural network As mentioned in Section 5.2.3, phase-space points
form the basis of the inputs to our neural network. We input the 4-momenta of
each outgoing parton as an $(n_j \times 4)$ array with each component of its 4-momenta
standardised to zero mean and unit variance across the training dataset. Although
it would be feasible to omit the energy component of the momenta or leave one
outgoing parton out of the training due to our datasets being generated with a fixed
centre-of-mass energy, we find that keeping all momenta information to improve

the network's performance. With the acceleration of training of neural networks on GPUs, the slowdown in keeping all outgoing momenta components is negligible.

Along with the 4-momenta, we also include the recoil factors

$$y_{ij,k} = \frac{p_i p_j}{p_i p_j + p_j p_k + p_i p_k} \,, \qquad (5.2.3)$$

as input for relevant permutations of $\{ijk\}$. We take the natural logarithm of $y_{ij,k}$ before standardising them to a zero mean and unit variance across the training dataset. We find that the addition of these recoil factors significantly improves performance of our neural network emulator during training and testing. This can be attributed to the fact that the coefficients C_{ijk} rely heavily on the mapped momenta in n-body phase-space, where $y_{ij,k}$ is usually required when performing the momentum mapping, see e.g. [50].

Following our ansatz (5.2.2), we must provide spin-averaged Catani-Seymour dipoles to the neural network. These dipoles are computed for all phase-space points before training begins, but they are not passed directly to the neural network as input features. Instead, they are only included in our custom loss function which will be explained in the next subsection.

Accounting for spin-correlation in $g \rightarrow gg$ In addition to the Catani-Seymour dipoles, we include other functions to account for the spin-correlations of $g \rightarrow gg$ and $g \rightarrow q\bar{q}$ splittings which are present in the factorisation formula but averaged out in the spin-averaged dipoles. This effect becomes relevant when there are two or more gluons in the final state. We seek to capture this behaviour by introducing a pair of terms of the form

$$S_{ij} \sin\left(2\phi_{ij}\right) + C_{ij} \cos\left(2\phi_{ij}\right) \qquad (5.2.4)$$

in the fitting ansatz for each gluon pair. The coefficients S_{ij} and C_{ij} are fitted by the neural network along with the dipole coefficients. The angle ϕ_{ij} is the azimuthal angle of the decay particles in the plane perpendicular to the parent particle momentum.

The procedure to obtain this angle is described in Appendix B.1.

Outputs of neural network   Denoting the raw output of our neural networks as c_{ijk}, they are transformed to C_{ijk} according to

$$C_{ijk} = S_{\text{coef}} \times \sinh\left(c_{ijk}\right) \tag{5.2.5}$$

where $S_{\text{coef}} = S_{\text{pred}}/S_{\text{dipole}}$. S_{pred} is the prediction scale, taken to be the minimum of the matrix elements in the training set and S_{dipole} is the dipole scale, taken to be the mean of all dipoles in the training set. These scaled coefficients are then multiplied with their corresponding Catani-Seymour dipole $D_{ij,k}$, and then summed to produce an estimation of the matrix element. The targets are the matrix elements corresponding to the phase-space point inputs in the training dataset. We transform the targets according to

$$y = \operatorname{arcsinh}\left(\frac{\langle|\mathcal{M}_{n+1}|^2\rangle}{S_{\text{pred}}}\right), \tag{5.2.6}$$

to reduce the orders of magnitude the matrix elements span. This performs a similar transformation to taking the natural logarithm except that it remains a valid transformation for negative arguments. We require this transformation to allow negative values as arguments because the coefficients C_{ijk} will not be restricted to only positive values, meaning that during training there is a possibility for the outputs of the network to go negative. By reducing the span of the targets, the neural network is able to more effectively pick out patterns across the entire training dataset rather than a smaller region, helping it to generalise. This technique is employed in other studies, see for example [201, 202]. We would like to stress that we do not expect, and have not observed, negative predictions for the matrix element from the neural network, as this would be unphysical. y is finally standardised to a zero mean and unit variance.

Training, validation, and testing datasets In Ref. [4], the authors used 500k training samples to train their models. For a fair comparison of our respective methods, we follow their methodology closely by constructing our models to be as close as possible to theirs and generate training and testing data by using code from their project repository [203][1]. Our model architectures will not be identical due to the difference in our methods, details of our model architecture are given in Section 5.2.4. Results of this comparison are presented in Section 5.3.1.

While training on 500k samples gives acceptable performance for the total cross-section, per-point accuracy is lacking. We find that increasing the size of the training dataset drastically improves the per-point prediction accuracy. Neural networks have been shown to scale well with large datasets [204] and given that they only need to be trained one time, it is useful to provide neural networks which have been pre-trained with maximum accuracy in mind. In addition to improving the per-point accuracy, we aim to overcome the problem of extrapolating to more singular regions. It is well known that neural networks in general do not extrapolate well [205, 206], but with our factorisation-aware model we show that letting the models learn about the infrared structure of QCD alleviates this problem.

To demonstrate the scaling performance of our model, we present our main results with models trained on more training samples where details of data generation are given in Section 5.2.3. For each multiplicity and global phase-space cut we generate a dataset consisting of 60 million phase-space points and their corresponding recoil factors, dipoles, and matrix elements. We then split this dataset into training, validation and testing datasets in a ratio of 4:1:1, meaning we have 40 million training samples, 10 million validation, and 10 million testing samples. The validation dataset is used to monitor model performance after each epoch of training, whereas the testing dataset is used as an out-of-sample check of the model's performance after training is complete.

---

[1]for our main results we generate all data using our own code

Number of final state jets	Input nodes	Output nodes
3	$(3 \times 4) + 2 = 14$	$2 + (2 \times 0) = 2$
4	$(4 \times 4) + 10 = 26$	$10 + (2 \times 1) = 12$
5	$(5 \times 4) + 27 = 47$	$27 + (2 \times 3) = 33$

Table 5.1: List of the number of input and output nodes for every process we consider.

Architecture

We have a fixed base to our neural network architecture that is used for all processes considered in this work, with variations in the number of nodes in the input and output layers due to the change in number of outgoing partons. Our base neural network consists of one input layer, eight hidden layers consisting of (64, 128, 256, 512, 768, 386, 128, 64) nodes, and one output layer.

The number of nodes in the input and output layers scales with the number of jets in the final state. The input layer has $(n_j \times 4) + n_{\text{rel}}$ nodes, and the output layer has $n_{\text{rel}} + (2 \times n_\phi)$ nodes, where n_{rel} is the number of relevant permutations and n_ϕ denotes the number of ϕ_{ij} angles. Relevant permutations, $\{ijk\}$, are the set of permutations for which their corresponding dipole could have the possibility to have a meaningful contribution to the matrix element for a given process. They are a subset of all the possible permutations for a given multiplicity, $P(n_j, 3)$. Relevant permutations exclude any permutation where a quark or anti-quark are emitted (i.e. $j = q$ or $\bar{q}$), as low energy quarks do not give rise to singularities in our processes of interest. Furthermore, we can remove degenerate permutations where swapping $i = g$ and $j = g$ has no effect as they have identical Catani-Seymour dipoles, e.g. $D_{34,1} = D_{43,1}$, so we only keep $D_{34,1}$. The omission of these redundant permutations speeds up training of the neural networks as we have fewer inputs, fewer dot products to compute in the loss function, as well as speeding up inference due to there being fewer dipoles to compute. For reference, we list the number of input and output nodes for each multiplicity we consider in Table 5.1.

The neural network weights are initialised according to the 'Glorot uniform' dis-

tribution as described in [154]. We use the tanh activation function for all nodes in the hidden layers, and have a linear activation function for nodes in the output layer. Initial learning rate is set to 0.001 and training mini-batch size is set to 4096. During training, we reduce the learning rate by a factor of 0.7 whenever there is no improvement in validation loss for 20 epochs. We use the `Keras` callback `ReduceLROnPlateau` to achieve this. Model training is terminated after the validation loss does not improve after 40 epochs of training using the `EarlyStopping` callback in `Keras`. We find that reducing the learning rate during training helps the model to converge to more optimal parameter sets. Since there are periods during training where the validation loss stagnates, reducing the learning rate helps to reach minima which otherwise wouldn't be accessible due to a too large learning rate. There is a possibility to reduce the learning rate too rapidly causing the model to have suboptimal optimisation, but this is countered by the high patience we set for the `ReduceLROnPlateau` callback.

These choices of hyperparameters were not results of extensive scanning of parameter space and were chosen heuristically. Hyperparameters can be tuned more optimally using more sophisticated methods, but they all rely on training a large number of models which is computationally expensive and time-consuming.

Custom loss function   To assess the model's performance when training we need to compare the network predictions with the targets. Our metric for the model's regression performance is the mean squared error (MSE)

$$L_{\mathrm{MSE}} = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - p(\vec{x}_i; \theta) \right)^2 , \tag{5.2.7}$$

where y_i is the target for the i-th sample out of N samples, and $p(\vec{x}_i; \theta)$ is the corresponding prediction obtained by combining the dipole factors and the azimuthal dependency terms multiplied by their NN-learned coefficients. In order to correct for the scale difference, we need to apply the transformation (5.2.6) on the neural network prediction first, with the same S_{pred} that was used for scaling the targets.

We choose the mean squared error to measure the network's performance because it is sensitive to outliers in the target distribution. This is useful because even though we have taken measures to reduce the span of the targets, the distribution still contains a tail towards larger values which correspond to soft and collinear configurations. These points have large contributions to the cross-section when integrating over phase-space, meaning it is important that we accurately predict these points. It is also convenient that the mean squared error tends to learn the mean of the target distribution which, in our case, corresponds to the cross-section.

In addition to the MSE, we introduce a regularisation term to penalise non-sparse representations of the matrix element. We know that in soft and collinear limits there will be dominant dipoles that have large contributions to the matrix element and there will be other dipoles with minimal contribution. We try to suppress the coefficients associated with these minimally contributing dipoles with the penalty term

$$L_{\mathrm{pen}} = J \sum_i \frac{D_i^{-2}}{\sum_j D_j^{-2}} |C_i D_i| \tag{5.2.8}$$

where the sum over i replaces the sum over $\{ijk\}$ for brevity, and $\sum_j D_j^{-2}$ is the sum over all dipoles for a phase-space point, acting as a normalisation factor. J is a tunable parameter that scales the importance of L_{pen} versus L_{MSE}. We found that models perform the best when $L_{\mathrm{pen}} < L_{\mathrm{MSE}}$, so J is tuned accordingly. It is possible for the product $C_i D_i$ to be large due to D_i alone, so to penalise this we regularise the product rather than just the coefficient, since it is the product that contributes to the matrix element.

The form of L_{pen} is reminiscent of the usual L1 regularisation which promotes sparse models. Regularisation is usually included to prevent overfitting by making the model make decisions on the most important features, reducing other features to zero. In our case we would like the neural network to learn about the universal factorisation property in QCD by making it choose a minimal amount of dipoles to represent the matrix element in singular regions. In addition to preventing overfitting,

([5.2.8]) helps the neural network to extrapolate to more soft and collinear regions as it has learnt to choose which dipoles are relevant in specific configurations. For non-singular configurations, the neural network is free to interpolate as there is not a clear set of dipoles that dominate, meaning L_{pen} is small.

Combining the regression loss term and the regularisation term, our expression for the total loss is

$$L = L_{\mathrm{MSE}} + L_{\mathrm{pen}}, \tag{5.2.9}$$

which is minimised through mini-batch gradient descent [207] with the Adam optimiser [158] to find optimal parameters θ for the neural network.

Ensemble of models Due to the random initialisation of weights in the neural network, and the fact that the optimisation procedure is carried out on mini-batches of the full training dataset, every neural network trained will be similar but non-identical, even with identical model architecture. This is partly because the loss surface is unlikely to be a completely smooth surface with a single global minimum, instead it is likely to contain multiple local minima, meaning it would be optimistic to believe that a single neural network is able to find the most optimal set of parameters. Given that we don't expect a single network to perform optimally, we train a number of models and aggregate their predictions to create an ensemble of models. Ensembling models is a well-known technique within machine learning, for a review see [208].

Each model in the ensemble is initialised with different weights according to the 'Glorot uniform' distribution, and trained on the same but randomly shuffled dataset, resulting in models that have been exposed to different distributions of the training data. After sufficient training, each model will have a distinct set of parameters that have similar predictive power. The prediction for the matrix element is then the mean of the outputs of all models in the ensemble[2]. Taking the mean will give

---

[2]if not explicitly stated, all references to neural network predictions from henceforth refers to

a more accurate and robust prediction as averaging over the different models will reduce variance due to over/underfitting in the training phase. We choose to have 20 models in our ensembles because we begin to see diminishing returns in the accuracy of per-point predictions after this. Another advantage of training an ensemble of models is that we have a measure of uncertainty, due to the neural networks, on the model predictions by calculating the standard error of the mean. That is we take the standard deviation of predictions across the ensemble and divide by the square root of number of models in the ensemble. This would not be possible with just a single model.

By choosing to build an ensemble, there is a performance impact because we have to spend more resources on training, and inference is slower due to having to predict on all models in the ensemble. However, we believe that having a more robust prediction with a measure of uncertainty outweigh these negatives. Additionally, each model only needs to be trained once, and slowdown during inference is alleviated with GPUs.

5.3   Results

In this section we present results for our matrix element emulator for e^+e^- annihilation into up to 5-jets. We first compare results obtained with our method to the tree-level results of Ref. [4], then proceed to exhibit results from larger NNs that have been exposed to larger training datasets to demonstrate the full scaling performance of our model. We then further demonstrate our method's capability to generalise to unseen regions of phase-space by assessing the prediction accuracy on random phase-space trajectories that venture well outside of the phase-space region used for the training.

---

the prediction from the ensemble of networks

5.3.1 Comparison with previous work

We compare our method to methods for tree-level matrix elements emulation from Ref. [4]. The authors presented two methods: 'single' and 'ensemble' models. A 'single' model indicates that there is one neural network trained across the entire phase-space, and an 'ensemble' model indicates that there is a group of neural networks trained together with weighting functions that focus individual networks on a specific divergent region of phase-space. In order to conduct a fair comparison, we have made attempts to follow their training methodology closely by constructing the neural networks at the centre of our model with a similar structure (e.g. same hidden layer structure, same activation functions, same random initial weight distribution), training methodology (same `EarlyStopping` criteria, same initial learning rate), and have generated training and testing datasets using code available on the authors' project repository with the relevant cuts.

We compare the distribution of errors for the matrix elements predictions on 3 million phase-space points. The distribution of errors is crucial because it informs us of the performance of our emulators at the matrix element level. It is well known that a neural network optimised with the mean squared error loss function has tendencies to learn the mean of the target distribution [209], meaning the quality of the cross-section prediction can potentially belie the point-to-point accuracy of emulators. In Figure 5.2, we plot the distribution of errors for matrix element predictions where our method is labelled 'Dipole NN'. We compare against the 'single' and 'ensemble' methods by training and testing using the corresponding datasets. For more detailed information on the differences please refer to [4]. Note that the height of the peak for the dipole histograms are not illustrated in the figure as it would not fit on the current axes but that is not important for this comparison. We can see that the prediction-to-truth ratio distribution for our method is much narrower and consistently peaked around the ideal accuracy, indicating our model performs better on a per-point basis for all multiplicities. Even with this reduced NN size we can see that incorporating

the known divergent structure explicitly in the model gives better results, as it uses the NN representation to learn a function that is more suitably approximated by a NN. For example, even though the three jet matrix element has a fairly trivial analytical structure, a standard fitting approach using a NN typically struggles to reproduce divergences. In our approach the NN only needs to emulate a non-singular modulation on top of the main divergent behaviour and is therefore more suited to the task.

5.3.2 Main results

Here we present our main results which are obtained using the larger NNs described in Section 5.2.4 along with larger training datasets described in Section 5.2.3. In Figure 5.3, we show the error distributions on 10 million matrix element predictions for each multiplicity and global phase-space cut.

Predicting on a large number of phase-space points allows us to explore singular regions with higher statistics. The ratios (right) clearly highlight the symmetry of the errors with Gaussian-like distributions tightly centred around 0. We have also included the absolute percentage difference distribution for more easily interpretable errors where the bulk of predictions are below the 0.1% error level. With increasing multiplicity, the fitting gets more challenging due to the rise in the number of singular regions in phase-space and the dimensionality of the phase-space, which can be seen in the decrease in accuracy as we increase multiplicity. Although the errors do increase, practically all matrix element predictions are below the 1% error level even for the most challenging scenario. Relaxing the global phase-space cut for the training and testing set is also expected to decrease model performance as allowing more singular regions of phase-space stretches the span of the target distribution making it difficult to fit with a single neural network. Our method manages to retain good performance while global phase-space cuts have been relaxed by a factor of 100 with only a small decrease in accuracy as illustrated in the left column of Figure 5.3.
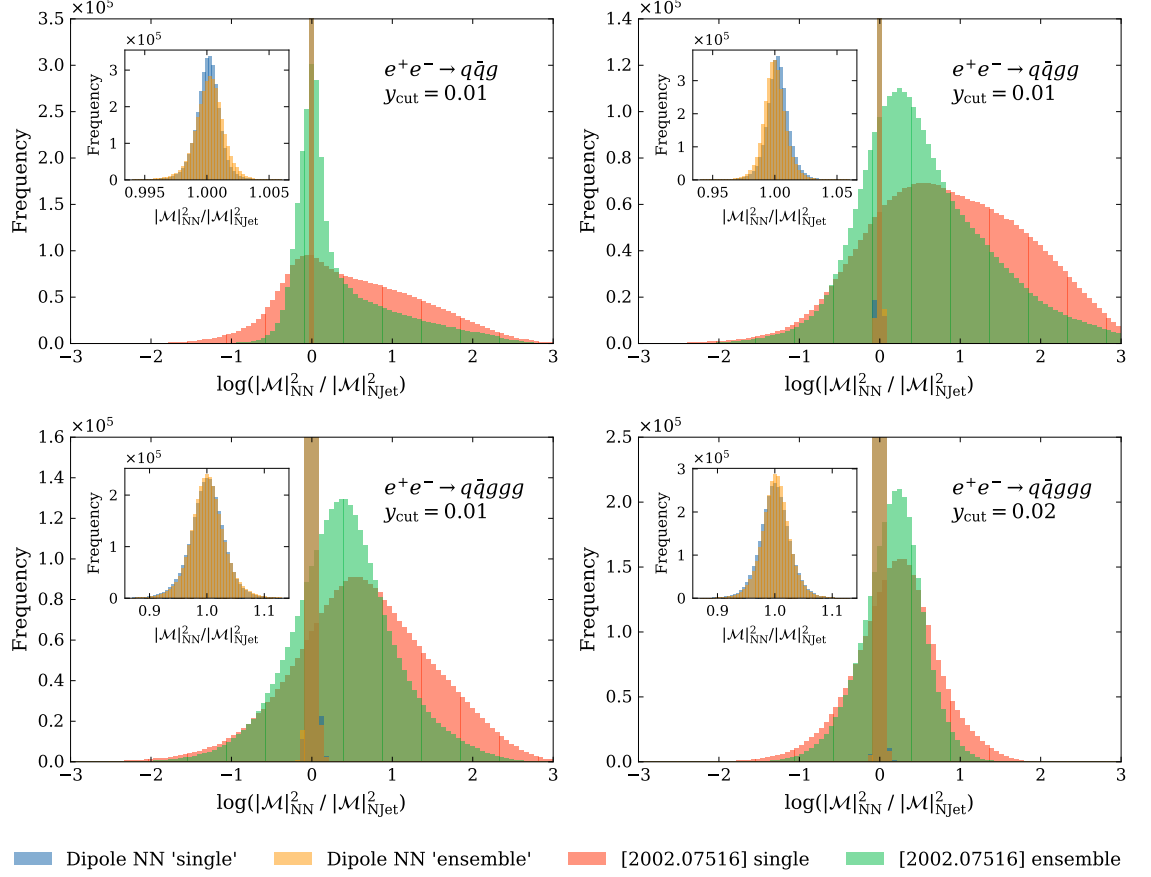
Figure 5.2: Error distribution compared to Figure 3 in Ref. [4], where data to reproduce the histograms were provided by the authors. We plot the log ratio of the matrix element as predicted by the neural network ensemble and the value from NJET on the main axes for comparison. The blue and orange dipole histograms representing our method are cut off at the top on the main axes, but the most important feature is the narrowness of the peak centred around the ideal value. The insets show the detailed distribution of our result on a linear scale.
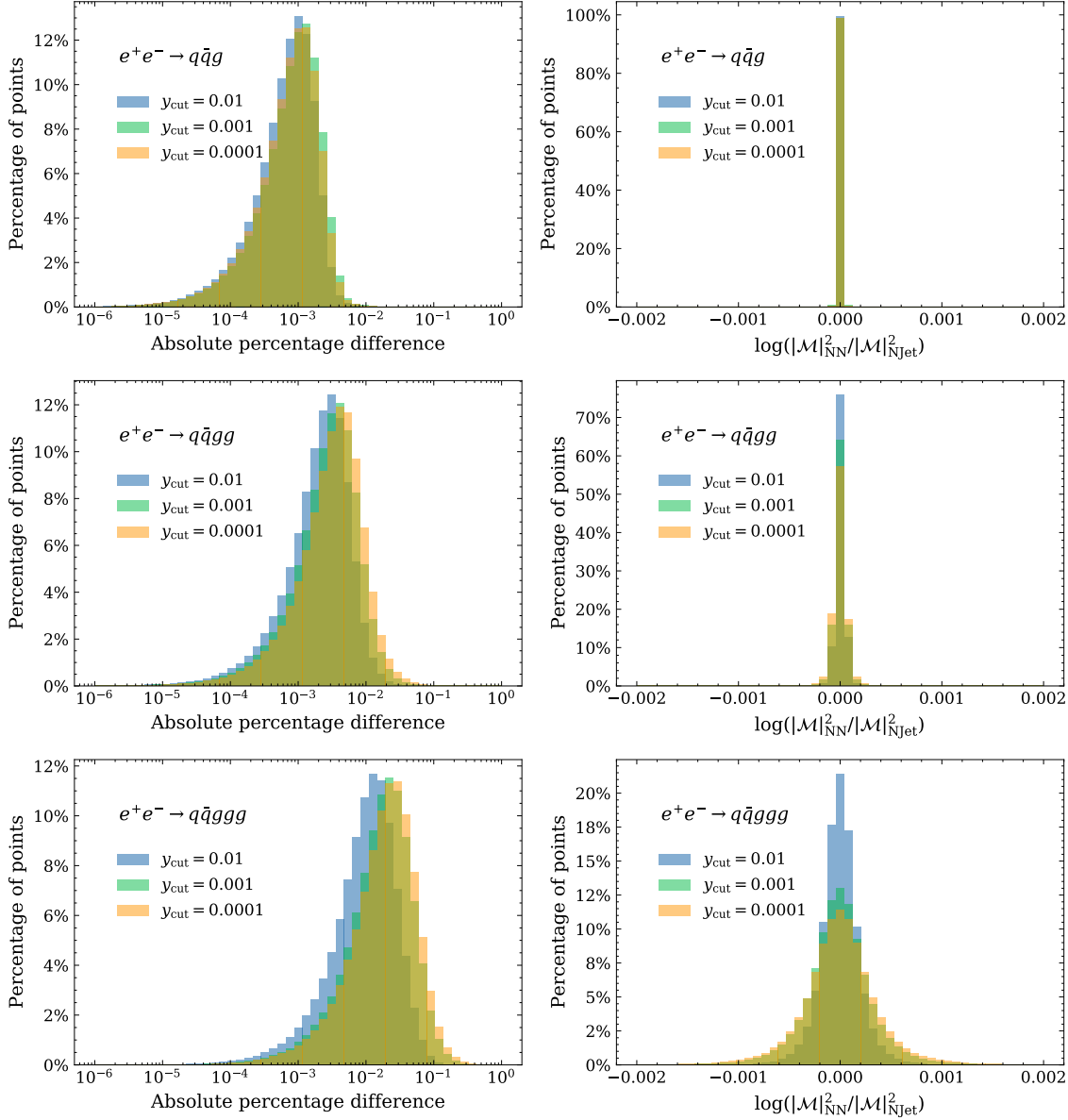
Figure 5.3: Error distributions for all three multiplicities (rows) and global phase-space cuts. Left: absolute percentage difference between NN prediction and NJET. Right: ratio of matrix elements from NN and NJET. Axis scales have been fixed for each column of subplots for ease of comparison between multiplicities.

This is because most of the span is accounted for by the dipole factors, while the coefficients themselves vary less. In the 3-jet case there is negligible difference between the different phase-space cuts while the 4 and 5 jet cases see less than a factor of 10 difference in the peaks of the absolute percentage difference distributions going from $y_{\mathrm{cut}} = 0.01$ to $y_{\mathrm{cut}} = 0.0001$.

By increasing the size of the training datasets we aim to expose the neural network to more samples of the phase-space, thereby increasing accuracy on predictions on as much of the phase-space as possible. Along with increasing the number of training samples, the neural network architecture has been expanded to include more hidden nodes and hidden layers. The extra hidden nodes and layers introduces more parameters into the model allowing the neural network to utilise the additional data. We found that to get good performance, we had to balance the size of the training dataset used and the size of the network. i.e. a small network is not expected to capture all the variations in a large dataset as easily as a larger network would, due to the smaller number of parameters available to the network. Of course, we also had to consider more physical constraints such as the time spent on training the neural networks which limits both the size of training datasets and architecture. In Figure 5.4, we show the improvements in accuracy of our main NNs compared to the smaller NNs from Section 5.3.1. Although the training and testing data for the smaller NNs are not identical due to differences in code used to generate the sets[3], for this comparison it suffices to show that the larger NNs are orders of magnitude more accurate than the smaller NNs.

Improvements in per-point accuracy translate to improved total cross-section predictions. In Figure 5.5 we show the percentage differences of the NN cross-section predicted compared to those from NJET. There is a similar trend of errors increasing with increasing multiplicity and more inclusive phase-space cuts. All total cross-section predictions are well below 0.1% error. There is a small systematic offset of

---

[3]although RAMBO is used for phase-space generation in both works, the selection criteria is different (JADE algorithm vs k_t algorithm)
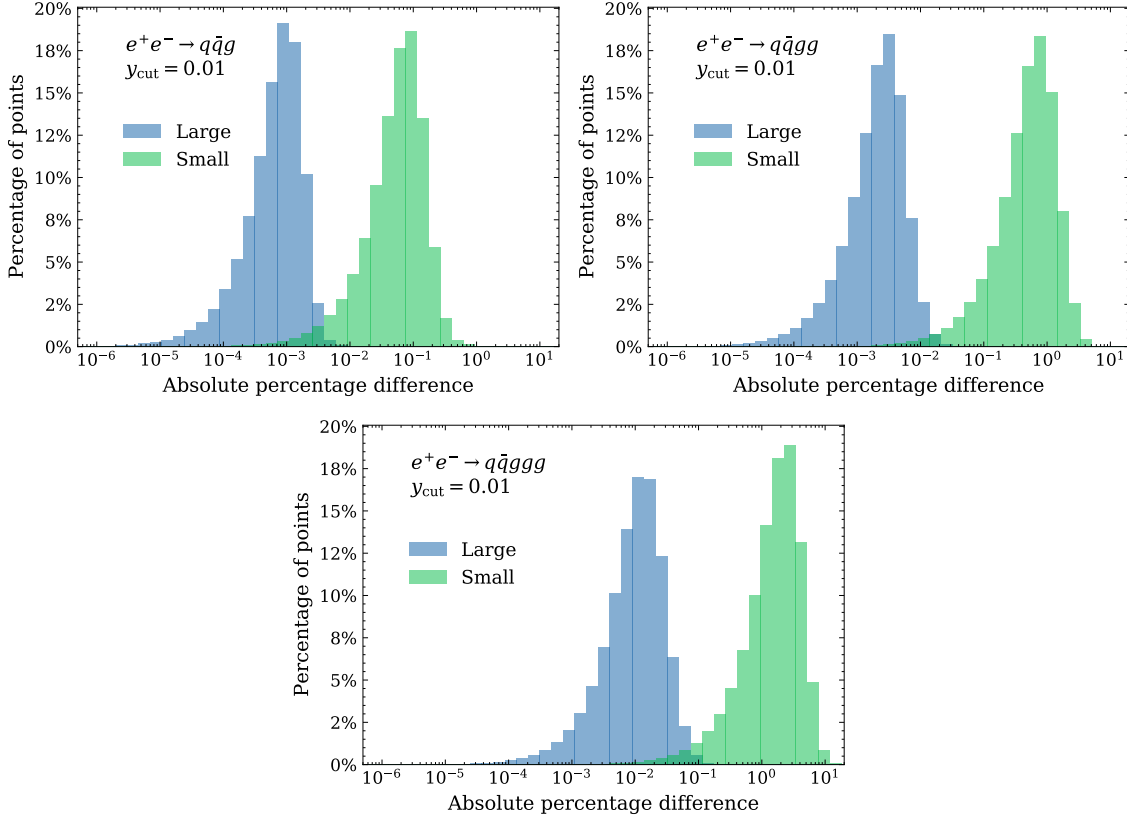
Figure 5.4: Comparison of error distributions of 10 million matrix element predictions between NNs used in Section 5.3.1 labelled as 'Small' and our larger NNs labelled as 'Large'. Note that 'small' and 'large' NNs were trained on 500k and 40m training samples, respectively. The testing data is identical to those shown in Figure 5.3, for the relevant global phase-space cuts.

the neural network cross-section compared to the NJET cross-section that becomes apparent under closer inspection. This was discussed in Ref. [4] and we provide an additional explanation in Appendix B.2.

We also show in Figure 5.5 the estimated statistical Monte Carlo integration relative error for comparison. The fact that the accumulated error on the matrix element is much lower than the MC error on the cross-section opens up the possibility to use the knowledge the network has gathered on the matrix element to augment the dataset to reduce the statistical error. Using such an augmentation technique would introduce a new systematic error on the prediction related to the accumulated network interpolation/extrapolation error, which would have to be balanced with the reduction in the MC integration error. Figure 5.5 suggests that the dataset could be augmented in such a way by a large factor before reaching a minimal overall uncertainty. This opportunity might not seem very useful for this particular example of leading order matrix elements where evaluations are relatively cheap computationally, but if a similar degree of accuracy in the emulation can be obtained for higher order matrix elements, this procedure could reduce the resource cost of matrix element calculation significantly. We defer the study of this augmentation method to future work.

Since we retain good performance by relaxing the global phase-space cut, we carry out a simple test of generalisability by using the 5-jet $y_{\text{cut}} = 0.0001$ model to infer on the two datasets with harsher cuts. This is shown in Figure 5.6 where we see that accuracy is comparable to the reference (blue) in both cases. In the case of $y_{\text{cut}} = 0.01$ (left), the model trained with more of the phase-space reduces errors in the right-hand tail of the distribution. This proves that enlarging the training phase-space can be done without having a large detrimental effect on the overall accuracy, and can significantly reduce the number of large prediction errors.
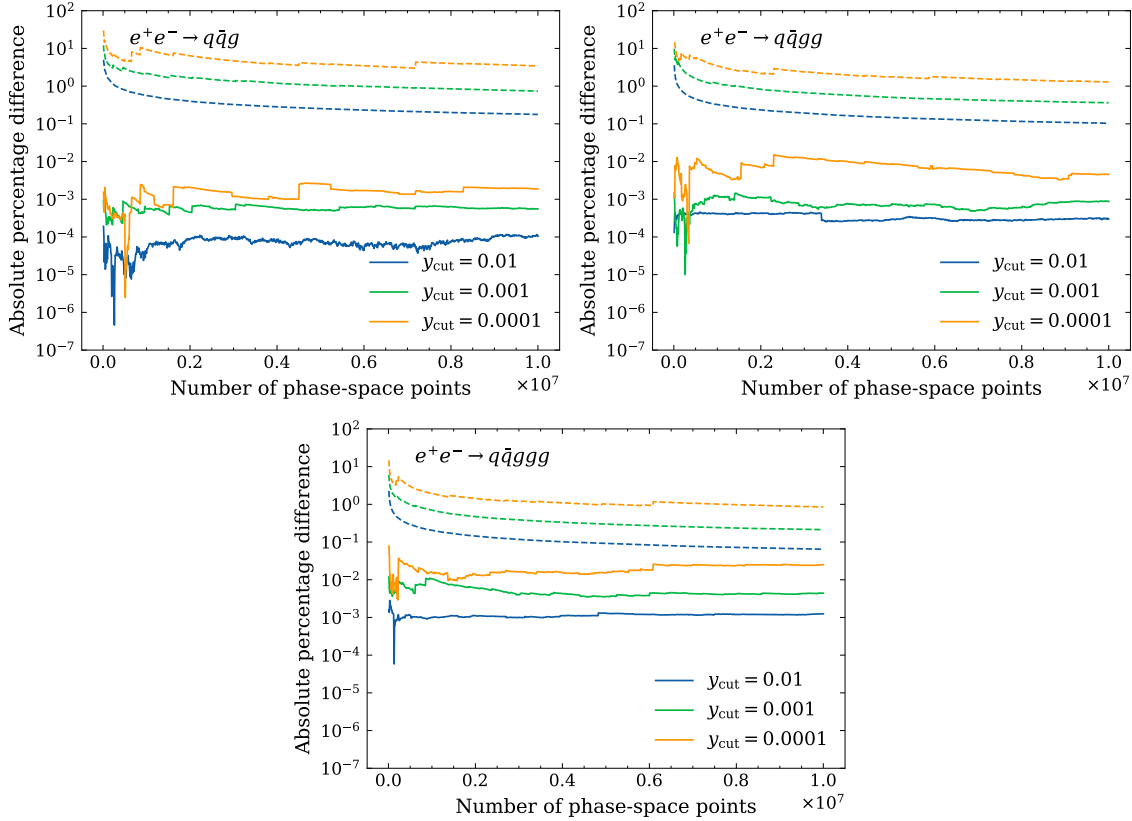
Figure 5.5: Absolute percentage differences between NN and NJet cross-section predictions in solid lines. Dashed lines represent the Monte Carlo error expressed as a percentage error relative to the NJet cross-section calculated with the corresponding number of phase-space points. Cross-sections have been calculated at intervals of 100k points up to the full 10 million phase-space points in the test set. Axis scales have been fixed to highlight the differences between multiplicities.
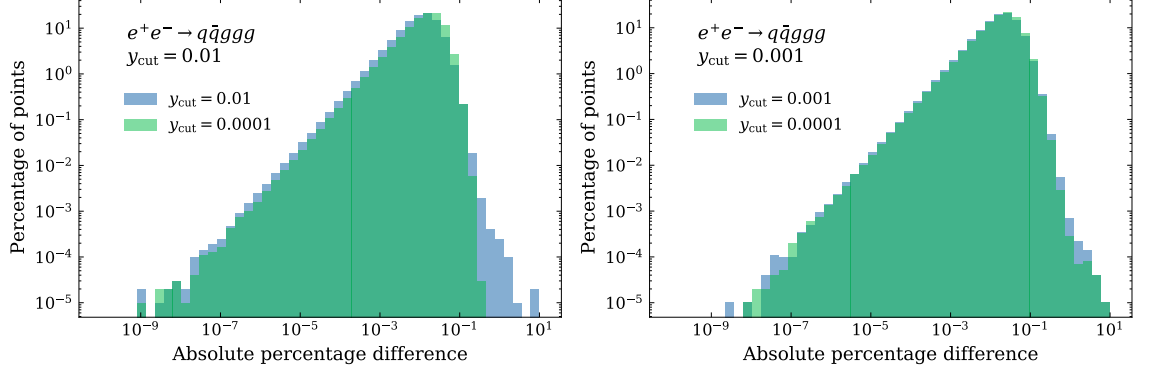
Figure 5.6: Left: error distribution on the $y_{\text{cut}} = 0.01$ testing dataset as predicted by $y_{\text{cut}} = 0.01$ and $y_{\text{cut}} = 0.0001$ models. Right: error distribution on the $y_{\text{cut}} = 0.001$ testing dataset as predicted by $y_{\text{cut}} = 0.001$ and $y_{\text{cut}} = 0.0001$ models. We use a logarithmic vertical axis to highlight the right-hand tail of the error distributions.

5.3.3 Random trajectories

As another test of our emulator we assess the accuracy of our predictions on random phase-space trajectories. These random phase-space trajectories are generated by connecting two random points in phase-space continuously without excluding any region of phase-space. This presents an interesting and challenging test of the interpolation and extrapolation abilities of the NNs as some parts of the trajectories may lie outside of the phase-space region of the training datasets. We show the results for 5-jet trajectories as this is the highest multiplicity we considered, predictions on lower multiplicities are better-behaved. We investigated 50 different random trajectories[4]. For the discussion in this article we chose one that contains many interesting features, namely the matrix elements span many orders of magnitudes and there are distinct peaks in the trajectory. In Figure 5.7 we show the predictions by the three 5-jet models trained on data with different global phase-space cuts for this trajectory. Left column shows the actual matrix element prediction and right column shows the ratios of the prediction to NJET. We analysed the predictions for the 50 random trajectories and measured the fraction of their length where the accuracy falls within given intervals. Table 5.2 shows the result for the regions of

---

[4]the corresponding plots are available in a document attached to this submission

phase-space where training data was available, and those falling beyond the data available to the model.

NN predictions are depicted as coloured scatter plots where the colour indicates the value of the minimum s_{ij} between any pair of final-state particles at that phase-space point. To more easily visualise the extrapolation performance of the NNs we highlight the regions where the minimum s_{ij} goes below the global phase-space cut applied to the training set the models were trained on, for each cut made. The regions of the plots where this occurs have been coloured in red. With these trajectories being completely randomly selected in phase-space there is a possibility for there to be doubly singular points or worse. To check for this we used `FastJet` to cluster the phase-space points in the same way we did for data generation, see Section 5.2.3. The pink regions indicate points which have two separate single unresolved limits, we label this configuration 'Single+single'. The purple regions indicate points which have a double unresolved limit (i.e. three particles in one jet), we label this configuration 'Double'. The blue regions indicate points which have both a double unresolved limit and a separate single unresolved limit, we label this configuration 'Double+single'. We do not include the quark-anti-quark invariant in defining these regions as there is no associated infrared divergence. The pink, purple, and blue bands indicates regions of points which would have been discarded for our training and testing datasets.

Accuracy is high when the minimum s_{ij} of the trajectory is not below any y_{cut}, i.e. when the NN prediction curve is blue. This is demonstrating that the NNs are interpolating well. Performance generally declines in the red, pink, purple, and blue regions, which is not unexpected as the NNs are extrapolating. Given that this trajectory has regions which go more collinear than any points the networks have been exposed to before, we would expect the networks which have been trained with the smallest y_{cut} parameter to perform best. We see that this is the case as accuracy is acceptable in the $y_{\text{cut}} = 0.0001$ models, including in the regions where the NN is extrapolating.

In summary, we have shown that the neural networks show acceptable performance
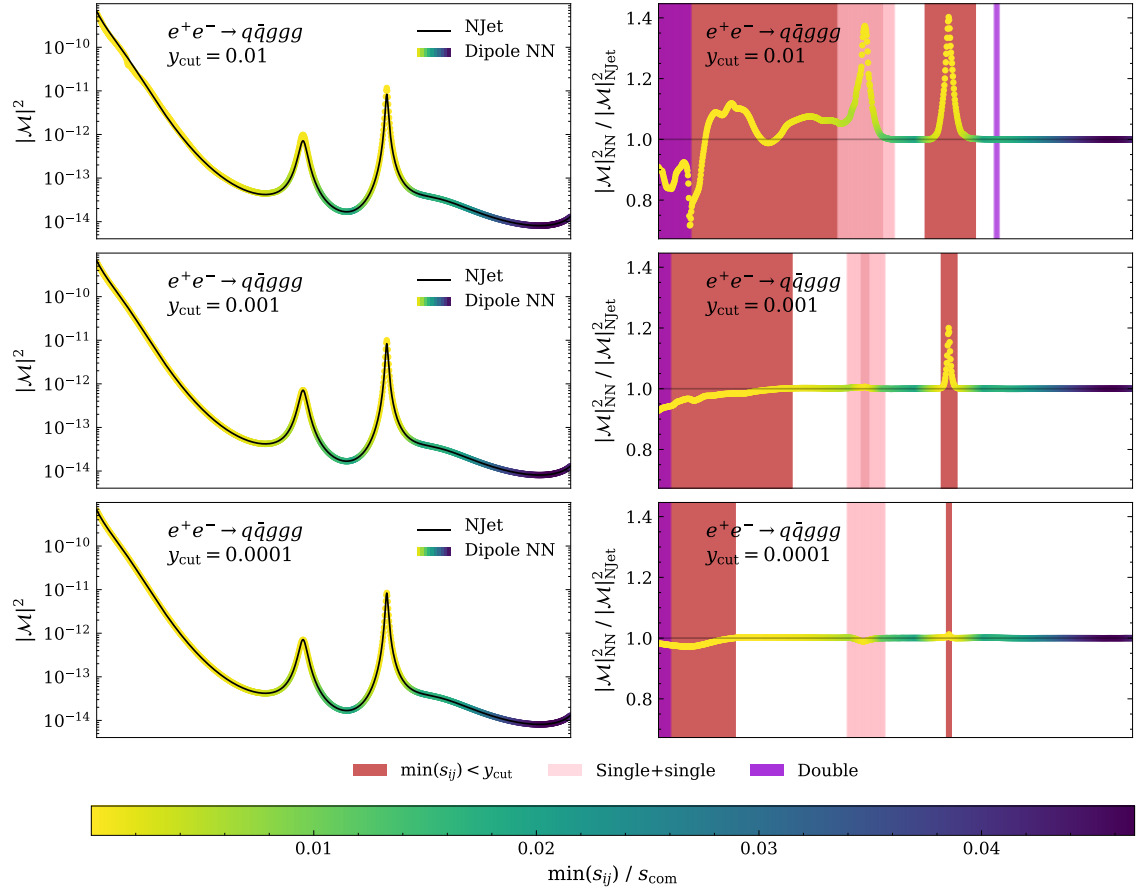
Figure 5.7: Left: matrix element prediction of random phase-space trajectory. Right: ratio of NN and NJET. NN predictions are coloured by minimum s_{ij} pair. Red bands indicate $\min(s_{ij})$ is smaller than y_{cut}. Pink bands indicate where there are two separate single unresolved limits. Purple bands indicate double unresolved limits, i.e. three particles in one jet. Blue bands indicate where there is both a double and a single unresolved limit. The pink, purple, and blue bands represent regions of phase-space which would have been excluded by `FastJet`.

Region	y_{cut}	Frac. of pts	Pts outside 0.1%	Pts outside 1%	Pts outside 5%
White	0.01	36.5%	1.9%	0.0%	0.0%
	0.001	74.8%	0.98%	0.0%	0.0%
	0.0001	78.4%	1.9%	0.0%	0.0%
Pink	0.01	27.6%	58.2%	25.2%	12.9%
	0.001	13.5%	31.7%	6.5%	0.53%
	0.0001	13.5%	38.9%	9.3%	2.8%
Purple	0.01	7.2%	69.1%	32.1%	20.3%
	0.001	3.6%	31.3%	7.9%	4.5%
	0.0001	3.6%	30.5%	1.6%	0.0%
Blue	0.01	1.5%	79.3%	41.6%	0.77%
	0.001	0.4%	29.2%	0.0%	0.0%
	0.0001	0.4%	76.4%	0.0%	0.0%
Red	0.01	52.5%	76.2%	38.6%	20.5%
	0.001	7.0%	69.9%	25.2%	9.5%
	0.0001	1.1%	90.8%	33.8%	1.5%

Table 5.2: The performance of trajectory predictions separated for white, pink, purple, blue, and red regions. We present the percentage of points that lie outside 0.1%/1.0%/5.0% errors. Fraction of points indicates the percentage of points that lie in the region of interest, out of all phase-space points from the 50 trajectories we examine.

on random phase-space trajectories which are of different nature to the datasets used to train and test the networks. Given that the general performance of the NNs of all three phase-space cuts are similar, it would make sense to use the models trained with the most inclusive phase-space cuts as it has been exposed to more of the complete phase-space.

5.4   Conclusion

In this article we presented a new strategy to emulate matrix elements using a neural network. By leveraging the knowledge of the factorisation properties of the matrix elements our model is able to extrapolate well outside of its training range. We showed that using this method we obtain significantly improved per-point accuracy than obtained in previous works. We also showed that the per-point accuracy of the model is not significantly affected by the generation cut for the training, which means that it would be possible to train our emulator on very inclusive cuts, allowing

them to be applied in a multitude of settings.

The accuracy of the emulation could allow users to augment the training dataset to reduce the MC error of cross-sections or distributions while using fewer computing resources compared to the original calculation. We leave the investigation of this aspect to further work.

Our method was demonstrated in this article using a tree-level process, but it could be generalised to higher order matrix elements by adapting the set of ingredients made available to the network for the interpolation. Specifically, we can include extra terms into our ansatz to help capture additional divergences at higher orders. We look forward to applying this method to such cases.

Chapter 6

Emulation of e^+e^- NLO k-factors

Chapter 7

Emulation of hadron-hadron initiated matrix elements

Chapter 8

Conclusion

Appendix A

Catani-Seymour notation and dipole functions

A.1 Notation

The notation for matrix elements given in Ref. [50] is repeated here for convenience.

Consider a tree-level matrix element with n partons in the final-state

$$\mathcal{M}_n^{c_1,\ldots,c_n;s_1,\ldots,s_n}(p_1,\ldots,p_n)\,, \qquad (A.1.1)$$

where $\{c_1,\ldots,c_n\}$ are colour indices, $\{s_1,\ldots,s_n\}$ are spin indices, and $\{p_1,\ldots,p_n\}$ are momenta.

Introducing the basis $\{|c_1,\ldots,c_n\rangle \otimes |s_1,\ldots,s_n\rangle\}$ in colour + helicity space allows us to write

$$\mathcal{M}_n^{c_1,\ldots,c_n;s_1,\ldots,s_n}(p_1,\ldots,p_n) \equiv (\langle c_1,\ldots,c_n| \otimes \langle s_1,\ldots,s_n|)\,|1,\ldots,n\rangle_n\,, \qquad (A.1.2)$$

where $|1,\ldots,n\rangle_n$ is a vector in colour + helicity space. The indices inside the vector denote the spin and colour indices, while the subscript denotes the multiplicity of the vector. The matrix element squared, summed over final-state colours and spins,

can therefore be written as

$$|\mathcal{M}_n|^2 = {}_n\langle 1, \ldots, n | 1, \ldots, n \rangle_n \, . \tag{A.1.3}$$

In the case of partons in the initial-state the colour + helicity vector $|1, \ldots, n\rangle$ has to be normalised by a factor $\sqrt{n_c}$ for each initial state parton. For hadron-hadron collisions this amounts to

$$|1, \ldots, n\rangle_n \to \frac{1}{\sqrt{n_c(a)n_c(b)}} |1, \ldots, n\rangle_n \, , \tag{A.1.4}$$

where $n_c(a)$ denotes the number of colour states of parton a. Namely $n_c(q) = n_c(\bar{q}) = N_c$ and $n_c(g) = N_c^2 - 1$.

Appendix B

Appendices for Chapter 5

B.1 Azimuthal angle ϕ_{ij} calculation

To calculate the azimuthal angle ϕ_{ij} for a pair of particle momenta p_i and p_j we first consider the plane perpendicular to the momentum

$$\vec{p}_{ij} = \vec{p}_i + \vec{p}_j \,. \tag{B.1.1}$$

We project the unit vector in the z direction and the momentum of particle i onto this plane[1]:

$$\vec{r}_z = \vec{e}_z - \left(\frac{\vec{p}_{ij} \cdot \vec{e}_z}{\vec{p}_{ij}^2} \right) \vec{p}_{ij} \,, \tag{B.1.2}$$

$$\vec{r}_i = \vec{p}_i - \left(\frac{\vec{p}_{ij} \cdot \vec{p}_i}{\vec{p}_{ij}^2} \right) \vec{p}_{ij} \,. \tag{B.1.3}$$

The angle ϕ_{ij} is the angle between these two projected vectors.

$$\sin \phi_{ij} = \hat{r}_{ij} \cdot (\hat{r}_i \times \hat{r}_z) \,, \quad \cos \phi_{ij} = \hat{r}_i \cdot \hat{r}_z \,, \tag{B.1.4}$$

where we have normalised all vectors to be unit vectors:

$$\hat{r}_z = \frac{\vec{r}_z}{|r_z|} \,, \qquad \hat{r}_i = \frac{\vec{r}_i}{|r_i|} \,, \qquad \hat{r}_{ij} = \frac{\vec{p}_{ij}}{|p_{ij}|} \,. \tag{B.1.5}$$

---

[1]using particle j instead results in a shift of ϕ_{ij} by π which makes no difference for $\sin 2\phi$ or $\cos 2\phi$.

B.2   Jensen's Inequality

Jensen's inequality states that for concave functions

$$f(\mathbb{E}[Y]) \geq \mathbb{E}[f(Y)]\,, \tag{B.2.1}$$

where $\mathbb{E}$ is the expectation value, the function f in our case is arcsinh that behaves similarly to the natural logarithm due to the scale of the problem, and Y is a random variable representing our target distribution. This inequality can be rewritten as

$$f(\mathbb{E}[Y]) - \mathbb{E}[f(Y)] \geq 0\,, \tag{B.2.2}$$

which is known as Jensen's gap. With a concave function, the mean of the transformed target distribution will always be underestimating the actual mean, i.e.

$$\mathbb{E}[Y] \geq \sinh\left(\mathbb{E}[\mathrm{arcsinh}(y)]\right), \tag{B.2.3}$$

where the LHS is the actual expectation value (cross-section) of the random variable Y and the RHS is the shifted expectation value that the neural network learns. In our scenario the neural network reduces the variance of the residual distribution, meaning the gap in reality is small but there will always be an offset in the mean value learnt.

B.3   Phase-space trajectories

We can see the RAMBO algorithm as a map from the unit hypercube in some high dimension into the n-particle phase-space. A flat distribution in the hypercube maps to a flat distribution in the multi-particle phase-space. To generate our phase-space trajectories we pick two points at random in the unit hypercube and map the line between them using the RAMBO mapping. As a result, each point on the resulting trajectory has equal probability density in phase-space. Any other phase-space generator that smoothly maps the unit hypercube to a multi-particle
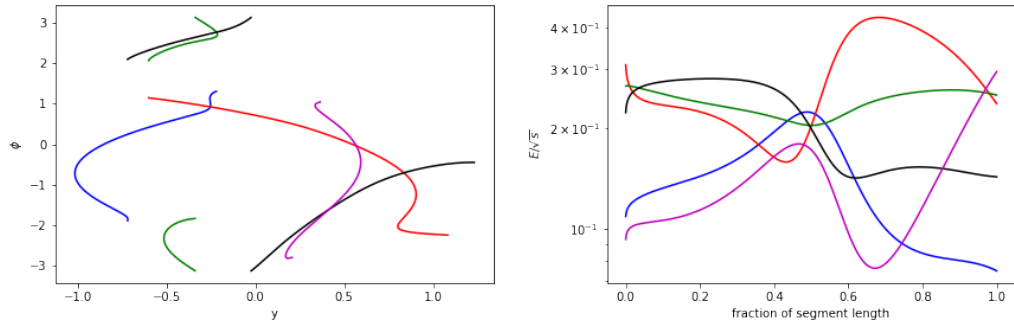
Figure B.1: Left: rapidity and azimuthal angle trajectories for the five final state particles in the trajectories used in Section 5.3.3. Right: Evolution of the same particle energies as a function of the position along the segment between the two random points in the unit hypercube.

phase-space could replace RAMBO in this procedure and could lead to trajectories with very different characteristics. One could for example imagine a sophisticated algorithm that only maps points close to the boundary of the hypercube to soft or collinear configurations. Using such an algorithm would have trajectories that avoid configurations with many particles more collinear or soft than the end-points of the trajectories. We find that with RAMBO the trajectories tend not to avoid difficult phase-space configurations and are therefore a good test of the extrapolation properties of our method.

Figure B.1 shows the trajectory we chose in Section 5.3.3.

Bibliography

[1] "File:Standard Model of Elementary Particles.svg - Wikimedia Commons." https://commons.wikimedia.org/wiki/File:Standard_Model_of_Elementary_Particles.svg.

[2] PARTICLE DATA GROUP collaboration, R. L. Workman, *Review of Particle Physics*, *PTEP* **2022** (2022) 083C01.

[3] A. LeNail, *Nn-svg: Publication-ready neural network architecture schematics*, *Journal of Open Source Software* **4** (2019) 747.

[4] S. Badger and J. Bullock, *Using neural networks for efficient evaluation of high multiplicity scattering amplitudes*, *JHEP* **06** (2020) 114, [2002.07516].

[5] D. Maître and H. Truong, *A factorisation-aware Matrix element emulator*, *JHEP* **11** (2021) 066, [2107.06625].

[6] J. Aylett-Bullock, C. Cuesta-Lazaro, A. Quera-Bofarull, M. Icaza-Lizaola, A. Sedgewick, H. Truong et al., *June: open-source individual-based epidemiology simulation*, *Royal Society Open Science* **8** (2021) 210506.

[7] I. Vernon, J. Owen, J. Aylett-Bullock, C. Cuesta-Lazaro, J. Frawley, A. Quera-Bofarull et al., *Bayesian emulation and history matching of june*, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **380** (2022) 20220039.

[8] T. Aoyama, T. Kinoshita and M. Nio, *Revised and Improved Value of the QED Tenth-Order Electron Anomalous Magnetic Moment*, *Phys. Rev. D* **97** (2018) 036001, [`1712.06060`].

[9] ATLAS collaboration, G. Aad et al., *Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC*, *Phys. Lett. B* **716** (2012) 1–29, [`1207.7214`].

[10] CMS collaboration, S. Chatrchyan et al., *Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC*, *Phys. Lett. B* **716** (2012) 30–61, [`1207.7235`].

[11] F. Englert and R. Brout, *Broken Symmetry and the Mass of Gauge Vector Mesons*, *Phys. Rev. Lett.* **13** (1964) 321–323.

[12] P. W. Higgs, *Broken Symmetries and the Masses of Gauge Bosons*, *Phys. Rev. Lett.* **13** (1964) 508–509.

[13] G. S. Guralnik, C. R. Hagen and T. W. B. Kibble, *Global Conservation Laws and Massless Particles*, *Phys. Rev. Lett.* **13** (1964) 585–587.

[14] Super-Kamiokande collaboration, Y. Fukuda et al., *Evidence for oscillation of atmospheric neutrinos*, *Phys. Rev. Lett.* **81** (1998) 1562–1567, [`hep-ex/9807003`].

[15] SNO collaboration, Q. R. Ahmad et al., *Direct evidence for neutrino flavor transformation from neutral current interactions in the Sudbury Neutrino Observatory*, *Phys. Rev. Lett.* **89** (2002) 011301, [`nucl-ex/0204008`].

[16] M. E. Peskin and D. V. Schroeder, *An Introduction to quantum field theory*. Addison-Wesley, Reading, USA, 1995.

[17] M. D. Schwartz, *Quantum Field Theory and the Standard Model*. Cambridge University Press, 3, 2014.

[18] J. C. Romao and J. P. Silva, *A resource for signs and Feynman diagrams of the Standard Model*, *Int. J. Mod. Phys. A* **27** (2012) 1230025, [1209.6213].

[19] M. Gell-Mann, *Symmetries of baryons and mesons*, *Phys. Rev.* **125** (1962) 1067–1084.

[20] J. Collins, *Foundations of perturbative QCD*, vol. 32. Cambridge University Press, 11, 2013.

[21] J. C. Collins and D. E. Soper, *The Theorems of Perturbative QCD*, *Ann. Rev. Nucl. Part. Sci.* **37** (1987) 383–409.

[22] J. C. Collins, D. E. Soper and G. F. Sterman, *Factorization of Hard Processes in QCD*, *Adv. Ser. Direct. High Energy Phys.* **5** (1989) 1–91, [hep-ph/0409313].

[23] S. Amoroso et al., *Snowmass 2021 whitepaper: Proton structure at the precision frontier*, 2203.13923.

[24] J. J. Ethier and E. R. Nocera, *Parton Distributions in Nucleons and Nuclei*, *Ann. Rev. Nucl. Part. Sci.* **70** (2020) 43–76, [2001.07722].

[25] P. Jimenez-Delgado, W. Melnitchouk and J. F. Owens, *Parton momentum and helicity distributions in the nucleon*, *J. Phys. G* **40** (2013) 093102, [1306.6515].

[26] A. Buckley, J. Ferrando, S. Lloyd, K. Nordström, B. Page, M. Rüfenacht et al., *LHAPDF6: parton density access in the LHC precision era*, *Eur. Phys. J. C* **75** (2015) 132, [1412.7420].

[27] PDF4LHC WORKING GROUP collaboration, R. D. Ball et al., *The PDF4LHC21 combination of global PDF fits for the LHC Run III*, *J. Phys. G* **49** (2022) 080501, [2203.05506].

[28] T.-J. Hou et al., *New CTEQ global analysis of quantum chromodynamics with high-precision data from the LHC*, *Phys. Rev. D* **103** (2021) 014013, [1912.10053].

[29] S. Bailey, T. Cridge, L. A. Harland-Lang, A. D. Martin and R. S. Thorne, *Parton distributions from LHC, HERA, Tevatron and fixed target data: MSHT20 PDFs*, *Eur. Phys. J. C* **81** (2021) 341, [2012.04684].

[30] NNPDF collaboration, R. D. Ball et al., *The path to proton structure at 1% accuracy*, *Eur. Phys. J. C* **82** (2022) 428, [2109.02653].

[31] Y. L. Dokshitzer, *Calculation of the Structure Functions for Deep Inelastic Scattering and e+ e- Annihilation by Perturbation Theory in Quantum Chromodynamics.*, *Sov. Phys. JETP* **46** (1977) 641–653.

[32] V. N. Gribov and L. N. Lipatov, *Deep inelastic e p scattering in perturbation theory*, *Sov. J. Nucl. Phys.* **15** (1972) 438–450.

[33] L. N. Lipatov, *The parton model and perturbation theory*, *Yad. Fiz.* **20** (1974) 181–198.

[34] G. Altarelli and G. Parisi, *Asymptotic Freedom in Parton Language*, *Nucl. Phys. B* **126** (1977) 298–318.

[35] S. Moch, J. A. M. Vermaseren and A. Vogt, *The Three loop splitting functions in QCD: The Nonsinglet case*, *Nucl. Phys. B* **688** (2004) 101–134, [hep-ph/0403192].

[36] A. Vogt, S. Moch and J. A. M. Vermaseren, *The Three-loop splitting functions in QCD: The Singlet case*, *Nucl. Phys. B* **691** (2004) 129–181, [hep-ph/0404111].

[37] G. 't Hooft and M. J. G. Veltman, *Regularization and Renormalization of Gauge Fields*, *Nucl. Phys. B* **44** (1972) 189–213.

[38] M. L. Mangano, *Introduction to qcd*, 1999.

[39] C. G. Callan, Jr., *Broken scale invariance in scalar field theory*, *Phys. Rev. D* **2** (1970) 1541–1547.

[40] K. Symanzik, *Small distance behavior in field theory and power counting*, *Commun. Math. Phys.* **18** (1970) 227–246.

[41] P. A. Baikov, K. G. Chetyrkin and J. H. Kühn, *Five-Loop Running of the QCD coupling constant*, *Phys. Rev. Lett.* **118** (2017) 082002, [1606.08659].

[42] T. Luthe, A. Maier, P. Marquard and Y. Schroder, *The five-loop Beta function for a general gauge group and anomalous dimensions beyond Feynman gauge*, *JHEP* **10** (2017) 166, [1709.07718].

[43] D. J. Gross and F. Wilczek, *Ultraviolet Behavior of Nonabelian Gauge Theories*, *Phys. Rev. Lett.* **30** (1973) 1343–1346.

[44] H. D. Politzer, *Reliable Perturbative Results for Strong Interactions?*, *Phys. Rev. Lett.* **30** (1973) 1346–1349.

[45] F. Bloch and A. Nordsieck, *Note on the Radiation Field of the electron*, *Phys. Rev.* **52** (1937) 54–59.

[46] D. R. Yennie, S. C. Frautschi and H. Suura, *The infrared divergence phenomena and high-energy processes*, *Annals Phys.* **13** (1961) 379–452.

[47] T. Kinoshita, *Mass singularities of Feynman amplitudes*, *J. Math. Phys.* **3** (1962) 650–677.

[48] T. D. Lee and M. Nauenberg, *Degenerate Systems and Mass Singularities*, *Phys. Rev.* **133** (1964) B1549–B1562.

[49] G. P. Salam, *Towards Jetography*, *Eur. Phys. J. C* **67** (2010) 637–686, [0906.1833].

[50] S. Catani and M. H. Seymour, *A General algorithm for calculating jet cross-sections in NLO QCD*, *Nucl. Phys. B* **485** (1997) 291–419, [`hep-ph/9605323`].

[51] A. Gehrmann-De Ridder, T. Gehrmann and E. W. N. Glover, *Antenna subtraction at NNLO*, *JHEP* **09** (2005) 056, [`hep-ph/0505111`].

[52] D. A. Kosower, *Antenna factorization of gauge theory amplitudes*, *Phys. Rev. D* **57** (1998) 5410–5416, [`hep-ph/9710213`].

[53] J. M. Campbell, M. A. Cullen and E. W. N. Glover, *Four jet event shapes in electron - positron annihilation*, *Eur. Phys. J. C* **9** (1999) 245–265, [`hep-ph/9809429`].

[54] Z. Bern, L. J. Dixon, D. C. Dunbar and D. A. Kosower, *One loop n point gauge theory amplitudes, unitarity and collinear limits*, *Nucl. Phys. B* **425** (1994) 217–260, [`hep-ph/9403226`].

[55] Z. Bern, V. Del Duca and C. R. Schmidt, *The Infrared behavior of one loop gluon amplitudes at next-to-next-to-leading order*, *Phys. Lett. B* **445** (1998) 168–177, [`hep-ph/9810409`].

[56] D. A. Kosower, *All order collinear behavior in gauge theories*, *Nucl. Phys. B* **552** (1999) 319–336, [`hep-ph/9901201`].

[57] Z. Bern, V. Del Duca, W. B. Kilgore and C. R. Schmidt, *The infrared behavior of one loop QCD amplitudes at next-to-next-to leading order*, *Phys. Rev. D* **60** (1999) 116001, [`hep-ph/9903516`].

[58] K. Fabricius, I. Schmitt, G. Kramer and G. Schierholz, *Higher Order Perturbative QCD Calculation of Jet Cross-Sections in e+ e- Annihilation*, *Z. Phys. C* **11** (1981) 315.

[59] G. Kramer and B. Lampe, *Jet Cross-Sections in e+ e- Annihilation*, *Fortsch. Phys.* **37** (1989) 161.

[60] W. T. Giele and E. W. N. Glover, *Higher order corrections to jet cross-sections in e+ e- annihilation*, *Phys. Rev. D* **46** (1992) 1980–2010.

[61] T. Binoth and G. Heinrich, *An automatized algorithm to compute infrared divergent multiloop integrals*, *Nucl. Phys. B* **585** (2000) 741–759, [hep-ph/0004013].

[62] T. Binoth and G. Heinrich, *Numerical evaluation of multiloop integrals by sector decomposition*, *Nucl. Phys. B* **680** (2004) 375–388, [hep-ph/0305234].

[63] R. K. Ellis, D. A. Ross and A. E. Terrano, *Calculation of Event Shape Parameters in e+ e- Annihilation*, *Phys. Rev. Lett.* **45** (1980) 1226–1229.

[64] S. Catani, S. Dittmaier, M. H. Seymour and Z. Trocsanyi, *The Dipole formalism for next-to-leading order QCD calculations with massive partons*, *Nucl. Phys. B* **627** (2002) 189–265, [hep-ph/0201036].

[65] S. Frixione, Z. Kunszt and A. Signer, *Three jet cross-sections to next-to-leading order*, *Nucl. Phys. B* **467** (1996) 399–442, [hep-ph/9512328].

[66] S. Frixione, *A General approach to jet cross-sections in QCD*, *Nucl. Phys. B* **507** (1997) 295–314, [hep-ph/9706545].

[67] D. A. Kosower, *Antenna factorization in strongly ordered limits*, *Phys. Rev. D* **71** (2005) 045016, [hep-ph/0311272].

[68] W. J. Torres Bobadilla et al., *May the four be with you: Novel IR-subtraction methods to tackle NNLO calculations*, *Eur. Phys. J. C* **81** (2021) 250, [2012.02567].

[69] S. Catani, S. Dittmaier and Z. Trocsanyi, *One loop singular behavior of QCD and SUSY QCD amplitudes with massive partons*, *Phys. Lett. B* **500** (2001) 149–160, [hep-ph/0011222].

[70] A. Gehrmann-De Ridder, T. Gehrmann and E. W. N. Glover, *Infrared structure of e+ e- —> 2 jets at NNLO*, *Nucl. Phys. B* **691** (2004) 195–222, [hep-ph/0403057].

[71] A. Daleo, T. Gehrmann and D. Maitre, *Antenna subtraction with hadronic initial states*, *JHEP* **04** (2007) 016, [hep-ph/0612257].

[72] A. Daleo, A. Gehrmann-De Ridder, T. Gehrmann and G. Luisoni, *Antenna subtraction at NNLO with hadronic initial states: initial-final configurations*, *JHEP* **01** (2010) 118, [0912.0374].

[73] R. Boughezal, A. Gehrmann-De Ridder and M. Ritzmann, *Antenna subtraction at NNLO with hadronic initial states: double real radiation for initial-initial configurations with two quark flavours*, *JHEP* **02** (2011) 098, [1011.6631].

[74] T. Gehrmann and P. F. Monni, *Antenna subtraction at NNLO with hadronic initial states: real-virtual initial-initial configurations*, *JHEP* **12** (2011) 049, [1107.4037].

[75] A. Gehrmann-De Ridder, T. Gehrmann and M. Ritzmann, *Antenna subtraction at NNLO with hadronic initial states: double real initial-initial configurations*, *JHEP* **10** (2012) 047, [1207.5779].

[76] A. Gehrmann-De Ridder, T. Gehrmann and E. W. N. Glover, *Quark-gluon antenna functions from neutralino decay*, *Phys. Lett. B* **612** (2005) 36–48, [hep-ph/0501291].

[77] A. Gehrmann-De Ridder, T. Gehrmann and E. W. N. Glover, *Gluon-gluon antenna functions from Higgs boson decay*, *Phys. Lett. B* **612** (2005) 49–60, [hep-ph/0502110].

[78] D. A. Kosower, *Multiple singular emission in gauge theories*, *Phys. Rev. D* **67** (2003) 116003, [hep-ph/0212097].

[79] A. Buckley et al., *General-purpose event generators for LHC physics*, *Phys. Rept.* **504** (2011) 145–233, [1101.2599].

[80] M. Bahr et al., *Herwig++ Physics and Manual*, *Eur. Phys. J. C* **58** (2008) 639–707, [0803.0883].

[81] J. Bellm et al., *Herwig 7.0/Herwig++ 3.0 release note*, *Eur. Phys. J. C* **76** (2016) 196, [1512.01178].

[82] T. Sjostrand, S. Mrenna and P. Z. Skands, *PYTHIA 6.4 Physics and Manual*, *JHEP* **05** (2006) 026, [hep-ph/0603175].

[83] C. Bierlich et al., *A comprehensive guide to the physics and usage of PYTHIA 8.3*, 2203.11601.

[84] T. Gleisberg, S. Hoeche, F. Krauss, M. Schonherr, S. Schumann, F. Siegert et al., *Event generation with SHERPA 1.1*, *JHEP* **02** (2009) 007, [0811.4622].

[85] SHERPA collaboration, E. Bothmann et al., *Event Generation with Sherpa 2.2*, *SciPost Phys.* **7** (2019) 034, [1905.09127].

[86] F. James, *Monte Carlo Theory and Practice*, *Rept. Prog. Phys.* **43** (1980) 1145.

[87] R. Kleiss, W. J. Stirling and S. D. Ellis, *A New Monte Carlo Treatment of Multiparticle Phase Space at High-energies*, *Comput. Phys. Commun.* **40** (1986) 359.

[88] S. Plätzer, *RAMBO on diet*, 1308.2922.

[89] P. D. Draggiotis, A. van Hameren and R. Kleiss, *SARGE: An Algorithm for generating QCD antennas*, *Phys. Lett. B* **483** (2000) 124–130, [hep-ph/0004047].

[90] A. van Hameren and C. G. Papadopoulos, *A Hierarchical phase space generator for QCD antenna structures*, *Eur. Phys. J. C* **25** (2002) 563–574, [hep-ph/0204055].

[91] G. P. Lepage, *A New Algorithm for Adaptive Multidimensional Integration*, *J. Comput. Phys.* **27** (1978) 192.

[92] G. P. Lepage, *Adaptive multidimensional integration: VEGAS enhanced*, *J. Comput. Phys.* **439** (2021) 110386, [2009.05112].

[93] R. Kleiss and R. Pittau, *Weight optimization in multichannel Monte Carlo*, *Comput. Phys. Commun.* **83** (1994) 141–146, [hep-ph/9405257].

[94] T. Ohl, *Vegas revisited: Adaptive Monte Carlo integration beyond factorization*, *Comput. Phys. Commun.* **120** (1999) 13–19, [hep-ph/9806432].

[95] T. Plehn, *Lectures on LHC Physics*, *Lect. Notes Phys.* **844** (2012) 1–193, [0910.4182].

[96] S. Höche, S. Prestel and H. Schulz, *Simulation of Vector Boson Plus Many Jet Final States at the High Luminosity LHC*, *Phys. Rev. D* **100** (2019) 014024, [1905.05120].

[97] C. Gao, S. Höche, J. Isaacson, C. Krause and H. Schulz, *Event Generation with Normalizing Flows*, *Phys. Rev. D* **101** (2020) 076002, [2001.10028].

[98] S. Jadach, *Foam: Multidimensional general purpose Monte Carlo generator with selfadapting symplectic grid*, *Comput. Phys. Commun.* **130** (2000) 244–259, [physics/9910004].

[99] S. Jadach, *Foam: A General purpose cellular Monte Carlo event generator*, *Comput. Phys. Commun.* **152** (2003) 55–100, [physics/0203033].

[100] HSF PHYSICS EVENT GENERATOR WG collaboration, S. Amoroso et al., *Challenges in Monte Carlo Event Generator Software for High-Luminosity LHC*, *Comput. Softw. Big Sci.* **5** (2021) 12, [2004.13687].

[101] K. Danziger, T. Janßen, S. Schumann and F. Siegert, *Accelerating Monte Carlo event generation – rejection sampling using neural network event-weight estimates*, *SciPost Phys.* **12** (2022) 164, [2109.11964].

[102] M. L. Mangano, M. Moretti, F. Piccinini, R. Pittau and A. D. Polosa, *ALPGEN, a generator for hard multiparton processes in hadronic collisions*, *JHEP* **07** (2003) 001, [hep-ph/0206293].

[103] F. Krauss, R. Kuhn and G. Soff, *AMEGIC++ 1.0: A Matrix element generator in C++*, *JHEP* **02** (2002) 044, [hep-ph/0109036].

[104] T. Gleisberg and S. Hoeche, *Comix, a new matrix element generator*, *JHEP* **12** (2008) 039, [0808.3674].

[105] A. Cafarella, C. G. Papadopoulos and M. Worek, *Helac-Phegas: A Generator for all parton level processes*, *Comput. Phys. Commun.* **180** (2009) 1941–1955, [0710.2427].

[106] J. Alwall, M. Herquet, F. Maltoni, O. Mattelaer and T. Stelzer, *MadGraph 5 : Going Beyond*, *JHEP* **06** (2011) 128, [1106.0522].

[107] W. Kilian, T. Ohl and J. Reuter, *WHIZARD: Simulating Multi-Particle Processes at LHC and ILC*, *Eur. Phys. J. C* **71** (2011) 1742, [0708.4233].

[108] C. F. Berger, Z. Bern, L. J. Dixon, F. Febres Cordero, D. Forde, H. Ita et al., *An Automated Implementation of On-Shell Methods for One-Loop Amplitudes*, *Phys. Rev. D* **78** (2008) 036003, [0803.4180].

[109] G. Cullen, N. Greiner, G. Heinrich, G. Luisoni, P. Mastrolia, G. Ossola et al., *Automated One-Loop Calculations with GoSam*, *Eur. Phys. J. C* **72** (2012) 1889, [1111.2034].

[110] G. Cullen et al., *GOSAM-2.0: a tool for automated one-loop calculations within the Standard Model and beyond*, *Eur. Phys. J. C* **74** (2014) 3001, [1404.7096].

[111] G. Bevilacqua, M. Czakon, M. V. Garzelli, A. van Hameren, A. Kardos, C. G. Papadopoulos et al., *HELAC-NLO*, *Comput. Phys. Commun.* **184** (2013) 986–997, [1110.1499].

[112] V. Hirschi, R. Frederix, S. Frixione, M. V. Garzelli, F. Maltoni and R. Pittau, *Automation of one-loop QCD corrections*, *JHEP* **05** (2011) 044, [1103.0621].

[113] J. Alwall, R. Frederix, S. Frixione, V. Hirschi, F. Maltoni, O. Mattelaer et al., *The automated computation of tree-level and next-to-leading order differential cross sections, and their matching to parton shower simulations*, *JHEP* **07** (2014) 079, [1405.0301].

[114] R. Frederix, S. Frixione, V. Hirschi, D. Pagani, H. S. Shao and M. Zaro, *The automation of next-to-leading order electroweak calculations*, *JHEP* **07** (2018) 185, [1804.10017].

[115] J. M. Campbell and R. K. Ellis, *An Update on vector boson pair production at hadron colliders*, *Phys. Rev. D* **60** (1999) 113006, [hep-ph/9905386].

[116] J. M. Campbell, S. Höche and C. T. Preuss, *Accelerating LHC phenomenology with analytic one-loop amplitudes: A C++ interface to MCFM*, *Eur. Phys. J. C* **81** (2021) 1117, [2107.04472].

[117] S. Badger, B. Biedermann, P. Uwer and V. Yundin, *Numerical evaluation of virtual corrections to multi-jet production in massless QCD*, *Comput. Phys. Commun.* **184** (2013) 1981–1998, [1209.0100].

[118] F. Cascioli, P. Maierhofer and S. Pozzorini, *Scattering Amplitudes with Open Loops*, *Phys. Rev. Lett.* **108** (2012) 111601, [1111.5206].

[119] F. Buccioni, J.-N. Lang, J. M. Lindert, P. Maierhöfer, S. Pozzorini, H. Zhang et al., *OpenLoops 2*, *Eur. Phys. J. C* **79** (2019) 866, [1907.13071].

[120] S. Actis, A. Denner, L. Hofer, J.-N. Lang, A. Scharf and S. Uccirati, *RECOLA: REcursive Computation of One-Loop Amplitudes*, *Comput. Phys. Commun.* **214** (2017) 140–173, [1605.01090].

[121] A. Denner, J.-N. Lang and S. Uccirati, *Recola2: REcursive Computation of One-Loop Amplitudes 2*, *Comput. Phys. Commun.* **224** (2018) 346–361, [1711.07388].

[122] T. Binoth et al., *A Proposal for a Standard Interface between Monte Carlo Tools and One-Loop Programs*, *Comput. Phys. Commun.* **181** (2010) 1612–1622, [1001.1307].

[123] S. Alioli et al., *Update of the Binoth Les Houches Accord for a standard interface between Monte Carlo tools and one-loop programs*, *Comput. Phys. Commun.* **185** (2014) 560–571, [1308.3462].

[124] J. Campbell, J. Huston and F. Krauss, *The Black Book of Quantum Chromodynamics: A Primer for the LHC Era*. Oxford University Press, 12, 2017.

[125] F. Caola, W. Chen, C. Duhr, X. Liu, B. Mistlberger, F. Petriello et al., *The Path forward to N^3LO*, in *2022 Snowmass Summer Study*, 3, 2022, 2203.06730.

[126] J. M. Campbell et al., *Event Generators for High-Energy Physics Experiments*, in *2022 Snowmass Summer Study*, 3, 2022, 2203.11110.

[127] E. Bothmann, A. Buckley, I. A. Christidi, C. Gütschow, S. Höche, M. Knobbe et al., *Accelerating LHC event generation with simplified pilot runs and fast PDFs*, 2209.00843.

[128] S. Frixione and B. R. Webber, *Matching NLO QCD computations and parton shower simulations*, *JHEP* **06** (2002) 029, [hep-ph/0204244].

[129] S. Frixione, P. Nason and C. Oleari, *Matching NLO QCD computations with Parton Shower simulations: the POWHEG method*, *JHEP* **11** (2007) 070, [0709.2092].

[130] S. Jadach, W. Płaczek, S. Sapeta, A. Siódmok and M. Skrzypek, *Matching NLO QCD with parton shower in Monte Carlo scheme — the KrkNLO method*, *JHEP* **10** (2015) 052, [1503.06849].

[131] S. Catani, F. Krauss, R. Kuhn and B. R. Webber, *QCD matrix elements + parton showers*, *JHEP* **11** (2001) 063, [hep-ph/0109231].

[132] L. Lonnblad and S. Prestel, *Matching Tree-Level Matrix Elements with Interleaved Showers*, *JHEP* **03** (2012) 019, [1109.4829].

[133] B. R. Webber, *A QCD Model for Jet Fragmentation Including Soft Gluon Interference*, *Nucl. Phys. B* **238** (1984) 492–528.

[134] J.-C. Winter, F. Krauss and G. Soff, *A Modified cluster hadronization model*, *Eur. Phys. J. C* **36** (2004) 381–395, [hep-ph/0311085].

[135] B. Andersson, G. Gustafson, G. Ingelman and T. Sjostrand, *Parton Fragmentation and String Dynamics*, *Phys. Rept.* **97** (1983) 31–145.

[136] B. R. Webber, *Fragmentation and hadronization*, *Int. J. Mod. Phys. A* **15S1** (2000) 577–606, [hep-ph/9912292].

[137] "Worldwide lhc computing grid." https://wlcg.web.cern.ch/.

[138] A. Krizhevsky, I. Sutskever and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, in *Advances in Neural Information Processing Systems* (F. Pereira, C. Burges, L. Bottou and K. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012, https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[139] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma et al., *Imagenet large scale visual recognition challenge*, *International journal of computer vision* **115** (2015) 211–252.

[140] J. Schmidhuber, *Deep learning in neural networks: An overview*, *Neural Networks* **61** (jan, 2015) 85–117.

[141] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel et al., *Scikit-learn: Machine learning in Python*, *Journal of Machine Learning Research* **12** (2011) 2825–2830.

[142] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro et al., *TensorFlow: Large-scale machine learning on heterogeneous systems*, 2015.

[143] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan et al., *Pytorch: An imperative style, high-performance deep learning library*, *Advances in neural information processing systems* **32** (2019) .

[144] T. Chen and C. Guestrin, *XGBoost: A scalable tree boosting system*, in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), pp. 785–794, ACM, 2016, DOI.

[145] CMS collaboration, A. M. Sirunyan et al., *Performance of the CMS Level-1 trigger in proton-proton collisions at $\sqrt{s} = 13$ TeV*, *JINST* **15** (2020) P10017, [2006.10165].

[146] LHCв collaboration, R. Aaij et al., *LHCb Detector Performance*, *Int. J. Mod. Phys. A* **30** (2015) 1530022, [1412.6352].

[147] M. Feickert and B. Nachman, *A Living Review of Machine Learning for Particle Physics*, 2102.02770.

[148] D. Guest, K. Cranmer and D. Whiteson, *Deep Learning and its Application to LHC Physics*, *Ann. Rev. Nucl. Part. Sci.* **68** (2018) 161–181, [1806.11484].

[149] A. Radovic, M. Williams, D. Rousseau, M. Kagan, D. Bonacorsi, A. Himmel et al., *Machine learning at the energy and intensity frontiers of particle physics*, *Nature* **560** (2018) 41–48.

[150] S. Badger et al., *Machine Learning and LHC Event Generation*, 2203.07460.

[151] P. Ramachandran, B. Zoph and Q. V. Le, *Searching for activation functions*, *CoRR* **abs/1710.05941** (2017) , [1710.05941].

[152] K. Hornik, *Approximation capabilities of multilayer feedforward networks*, *Neural Networks* **4** (1991) 251–257.

[153] Z. Lu, H. Pu, F. Wang, Z. Hu and L. Wang, *The expressive power of neural networks: A view from the width*, *Advances in neural information processing systems* **30** (2017) , [1709.02540].

[154] X. Glorot and Y. Bengio, *Understanding the difficulty of training deep feedforward neural networks*, in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, vol. 9 of *Proceedings of Machine Learning Research*, pp. 249–256, PMLR, 2010, https://proceedings.mlr.press/v9/glorot10a.html.

[155] K. He, X. Zhang, S. Ren and J. Sun, *Delving deep into rectifiers: Surpassing human-level performance on imagenet classification*, *IEEE International Conference on Computer Vision (ICCV 2015)* **1502** (02, 2015) .

[156] J. Kiefer and J. Wolfowitz, *Stochastic estimation of the maximum of a regression function*, *The Annals of Mathematical Statistics* **23** (09, 1952) .

[157] G. Hinton, N. Srivastava and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent." http://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf, 2012.

[158] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 12, 2014, 1412.6980.

[159] S. Ruder, *An overview of gradient descent optimization algorithms*, 1609.04747.

[160] T. Yu and H. Zhu, *Hyper-parameter optimization: A review of algorithms and applications*, *ArXiv* (2020) , [2003.05689].

[161] J. Bergstra, D. Yamins and D. Cox, *Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures*, in *International conference on machine learning*, pp. 115–123, PMLR, 2013, https://proceedings.mlr.press/v28/bergstra13.html.

[162] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins and D. Cox, *Hyperopt: A python library for model selection and hyperparameter optimization*, *Computational Science & Discovery* **8** (07, 2015) 014008.

[163] K. Jamieson and A. Talwalkar, *Non-stochastic best arm identification and hyperparameter optimization*, in *Artificial intelligence and statistics*, pp. 240–248, PMLR, 2016, https://proceedings.mlr.press/v51/jun16.html.

[164] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh and A. Talwalkar, *Hyperband: A novel bandit-based approach to hyperparameter optimization*, *Journal of Machine Learning Research* **18** (04, 2018) 1–52.

[165] S. Falkner, A. Klein and F. Hutter, *Bohb: Robust and efficient hyperparameter optimization at scale*, in *International Conference on Machine Learning*, pp. 1437–1446, PMLR, 2018, 1807.01774.

[166] S. Otten, K. Rolbiecki, S. Caron, J.-S. Kim, R. Ruiz De Austri and J. Tattersall, *DeepXS: Fast approximation of MSSM electroweak cross sections at NLO*, *Eur. Phys. J. C* **80** (2020) 12, [1810.08312].

[167] A. Buckley, A. Kvellestad, A. Raklev, P. Scott, J. V. Sparre, J. Van Den Abeele et al., *Xsec: the cross-section evaluation code*, *Eur. Phys. J. C* **80** (2020) 1106, [2006.16273].

[168] A. Alnuqaydan, S. Gleyzer and H. Prosper, *SYMBA: Symbolic Computation of Squared Amplitudes in High Energy Physics with Machine Learning*, 2206.08901.

[169] R. Winterhalder, V. Magerya, E. Villa, S. P. Jones, M. Kerner, A. Butter et al., *Targeting multi-loop integrals with neural networks*, *SciPost Phys.* **12** (2022) 129, [2112.09145].

[170] A. Dersy, M. D. Schwartz and X. Zhang, *Simplifying Polylogarithms with Machine Learning*, 2206.04115.

[171] F. Bishara and M. Montull, *(Machine) Learning amplitudes for faster event generation*, 1912.11055.

[172] J. Aylett-Bullock, S. Badger and R. Moodie, *Optimising simulations for diphoton production at hadron colliders using amplitude neural networks*, *JHEP* **08** (2021) 066, [2106.09474].

[173] S. Badger, A. Butter, M. Luchmann, S. Pitz and T. Plehn, *Loop Amplitudes from Precision Networks*, 2206.14831.

[174] J. Bendavid, *Efficient Monte Carlo Integration Using Boosted Decision Trees and Generative Deep Neural Networks*, 1707.00028.

[175] M. D. Klimek and M. Perelstein, *Neural Network-Based Approach to Phase Space Integration*, *SciPost Phys.* **9** (2020) 053, [1810.11509].

[176] E. Bothmann, T. Janßen, M. Knobbe, T. Schmale and S. Schumann, *Exploring phase space with Neural Importance Sampling*, *SciPost Phys.* **8** (2020) 069, [2001.05478].

[177] B. Stienen and R. Verheyen, *Phase Space Sampling and Inference from Weighted Events with Autoregressive Flows*, *SciPost Phys.* **10** (2021) 038, [2011.13445].

[178] I.-K. Chen, M. Klimek and M. Perelstein, *Improved neural network monte carlo simulation*, *SciPost Physics* **10** (Jan, 2021) .

[179] S. Carrazza and F. A. Dreyer, *Lund jet images from generative and cycle-consistent adversarial networks*, *Eur. Phys. J. C* **79** (2019) 979, [1909.01359].

[180] E. Bothmann and L. Debbio, *Reweighting a parton shower using a neural network: the final-state case*, *JHEP* **01** (2019) 033, [1808.07802].

[181] K. Dohi, *Variational Autoencoders for Jet Simulation*, 2009.04842.

[182] B. Hashemi, N. Amin, K. Datta, D. Olivito and M. Pierini, *LHC analysis-specific datasets with Generative Adversarial Networks*, 1901.05282.

[183] R. Di Sipio, M. Faucci Giannelli, S. Ketabchi Haghighat and S. Palazzo, *DijetGAN: A Generative-Adversarial Network Approach for the Simulation of QCD Dijet Events at the LHC*, *JHEP* **08** (2019) 110, [1903.02433].

[184] A. Butter, T. Plehn and R. Winterhalder, *How to GAN LHC Events*, *SciPost Phys.* **7** (2019) 075, [1907.03764].

[185] M. Backes, A. Butter, T. Plehn and R. Winterhalder, *How to GAN Event Unweighting*, 2012.07873.

[186] A. Butter, S. Diefenbacher, G. Kasieczka, B. Nachman and T. Plehn, *GANplifying Event Samples*, 2008.06545.

[187] Y. Alanazi et al., *Simulation of electron-proton scattering events by a Feature-Augmented and Transformed Generative Adversarial Network (FAT-GAN)*, 2001.11103.

[188] B. Nachman and J. Thaler, *Neural resampler for Monte Carlo reweighting with preserved uncertainties*, *Phys. Rev. D* **102** (2020) 076004, [2007.11586].

[189] M. Paganini, L. de Oliveira and B. Nachman, *CaloGAN : Simulating 3D high energy particle showers in multilayer electromagnetic calorimeters with generative adversarial networks*, *Phys. Rev. D* **97** (2018) 014021, [1712.10321].

[190] SHiP collaboration, C. Ahdida et al., *Fast simulation of muons produced at the SHiP experiment using Generative Adversarial Networks*, *JINST* **14** (2019) P11028, [1909.04451].

[191] D. Derkach, N. Kazeev, F. Ratnikov, A. Ustyuzhanin and A. Volokhova, *Cherenkov Detectors Fast Simulation Using Neural Networks*, *Nucl. Instrum. Meth. A* **952** (2020) 161804, [1903.11788].

[192] Y. Alanazi et al., *AI-based Monte Carlo event generator for electron-proton scattering*, 2008.03151.

[193] A. Andreassen, P. T. Komiske, E. M. Metodiev, B. Nachman and J. Thaler, *OmniFold: A Method to Simultaneously Unfold All Observables*, *Phys. Rev. Lett.* **124** (2020) 182001, [1911.09107].

[194] M. Bellagente, A. Butter, G. Kasieczka, T. Plehn and R. Winterhalder, *How to GAN away Detector Effects*, *SciPost Phys.* **8** (2020) 070, [1912.00477].

[195] H. Truong, "Fame." https://github.com/htruong0/fame, 2021.

[196] A. Bassetto, M. Ciafaloni and G. Marchesini, *Jet structure and infrared sensitive quantities in perturbative qcd*, *Physics Reports* **100** (1983) 201–272.

[197] M. Cacciari, G. P. Salam and G. Soyez, *FastJet User Manual*, *Eur. Phys. J. C* **72** (2012) 1896, [1111.6097].

[198] N. Dawe, E. Rodrigues, H. Schreiner, B. Ostdiek, D. Kalinkin, M. R. et al., *scikit-hep/pyjet: Version 1.8.2*, Jan., 2021. 10.5281/zenodo.4446849.

[199] S. Catani, Y. L. Dokshitzer, M. Olsson, G. Turnock and B. R. Webber, *New clustering algorithm for multi - jet cross-sections in e+ e- annihilation*, *Phys. Lett. B* **269** (1991) 432–438.

[200] F. Chollet et al., "Keras." https://keras.io, 2015.

[201] A. Coccaro, M. Pierini, L. Silvestrini and R. Torre, *The dnnlikelihood: enhancing likelihood distribution with deep learning*, *The European Physical Journal C* **80** (Jul, 2020) .

[202] F. Bury and C. Delaere, *Matrix element regression with deep neural networks — breaking the cpu barrier*, *Journal of High Energy Physics* **2021** (Apr, 2021) .

[203] J. Aylett-Bullock, "n3jet." https://github.com/JosephPB/n3jet, 2020.

[204] J. Hestness, S. Narang, N. Ardalani, G. Diamos, H. Jun, H. Kianinejad et al., *Deep learning scaling is predictable, empirically*, 2017.

[205] K. Xu, M. Zhang, J. Li, S. S. Du, K. ichi Kawarabayashi and S. Jegelka, *How neural networks extrapolate: From feedforward to graph neural networks*, 2021.

[206] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child et al., *Scaling laws for neural language models*, 2020.

[207] L. Bottou, F. E. Curtis and J. Nocedal, *Optimization methods for large-scale machine learning*, 2018.

[208] Z.-H. Zhou, J. Wu and W. Tang, *Ensembling neural networks: Many could be better than all*, *Artificial Intelligence* **137** (2002) 239–263.

[209] B. Nachman, *A guide for deploying deep learning in lhc searches: How to achieve optimality and account for uncertainty*, *SciPost Physics* **8** (Jun, 2020) .