Computational Physics Laboratory:
Tree-Level Gluon Scattering Amplitudes in Mathematica

Jiahao Miao
AY 2024/25

Abstract

This report explores the use of the Mathematica software package to compute tree-level gluon scattering amplitudes in perturbative quantum chromodynamics (QCD), leveraging its powerful algebraic capabilities. The main focus is on utilizing Mathematica's list manipulation functions to represent Feynman diagrams, applying Feynman rules to evaluate the corresponding scattering amplitudes. Emphasis is placed on the computational techniques used to automate and simplify these calculations and resources consumed. Key properties such as Ward identities and symmetry under the exchange of external legs are verified to ensure the robustness of the calculations. While Mathematica proves efficient in handling complex algebraic expressions and streamlining the evaluation of QFT amplitudes, the report also highlights the exponential growth in computational resource requirements as the number of external legs increases.

Contents

1 Introduction

Quantum Chromodynamics (QCD), the fundamental theory of the strong interaction, describes the behavior of quarks and gluons. Understanding high-energy collider phenomena, such as jet production, necessitates the precise calculation of scattering amplitudes involving these fundamental particles.

While the theoretical framework for computing these amplitudes using Feynman diagrams is well-established, the practical execution of these calculations quickly becomes an arduous and error-prone task due to the rapidly increasing number of diagrams, the complexity of algebraic expressions, and the intricate handling of momentum, color, and Lorentz indices.

For gluon scattering in QCD, the number of tree-level Feynman diagrams grows more than factorially with the number of external legs $n+1$, following an asymptotic growth pattern given by (see notebook):

$$\mathcal{O}\left(\left(\frac{9\sqrt{3}+12}{11}\right)^{n}\frac{n!}{n^{3/2}}\right) \tag{1}$$

For instance, the number of tree-level diagrams for the first few $n+1$ gluons can be computed using a recurrence relation (Listing 1) or counted directly after generating the diagrams in Mathematica.

Code 1: Number of tree-level diagrams for $n+1$ gluons

```
a[0] = 0; a[1] = 1; a[2] = 1;
a[n_] := a[n_]:=a[n]=(12(2n-3)a[n-1]+(3n-5)(3n-7)a[n-2])/11;
Table[a[n], {n, 3, 12}];
Output: {4, 25, 220, 2485, 34300, 559405, 10525900, 224449225, 5348843500, 140880765025}
```

This increasing complexity, particularly in perturbative QCD, makes a computational approach essential. For this project, Mathematica [1] was selected as the primary computational environment due to its unparalleled symbolic manipulation capabilities, which are uniquely well-suited for the challenges of calculating Feynman amplitudes. Unlike compiled languages such as C++ or Fortran, which excel in numerical computations and low-level control, Mathematica provides a high-level, interactive environment that natively understands and operates on symbolic expressions.

This report details the computational implementation of gluon amplitude calculations in QCD using Mathematica. We will demonstrate how Mathematica's unique strengths in symbolic computation, automated algebraic simplification, and high-level programming facilitate the systematic construction of Feynman diagrams, the automated assignment of momenta, the application of Feynman rules, and ultimately, the derivation of explicit amplitude expressions and modulus squared.

The subsequent sections will walk through the design of our data structures, the algorithms for momentum assignment and constraint solving, the implementation of Feynman rules as substitution rules, color algebra handling, verification of key properties such as Ward identities, and computation of the modulus squared.

All the code used in this report were run in a laptop with AMD Ryzen 7 5800H CPU and 16 GB RAM.

2 Implementation

2.1 Generating Feynman Diagrams

The first step is to encode the Feynman diagrams in a way that Mathematica can manipulate them (fig. 1). We represent each diagram as a list of pairs, for instance, the third diagram in fig. 1 can be represented as:

$$\{\{1,-1\},\{2,-1\},\{3,-2\},\{4,-2\},\{5,-2\},\{-1,-2\}\};$$

with the convention that the external legs are represented by positive integers and the internal vertices by negative integers, the pairs representing the connections between the vertices always ordered in decreasing order.

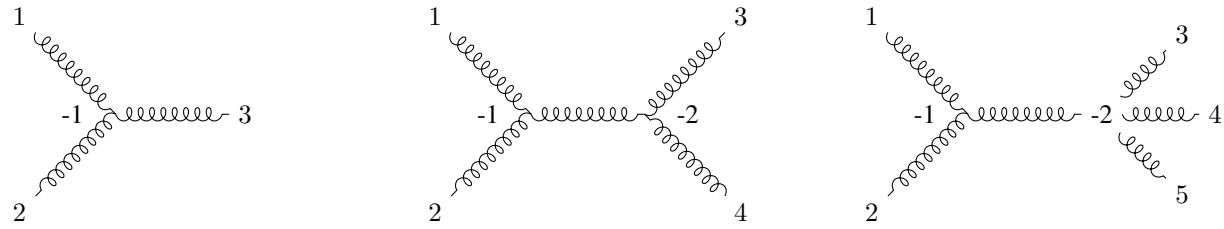

Figure 1: Example of a Feynman diagram for gluon scattering with 3, 4, and 5 external legs. On the left, the diagram with 3 external legs is the skeleton from which the other diagrams are built.

To generate the Feynman diagrams, we use a recursive function that starts with an initial diagram and progressively adds vertices. First, 3-point vertices are introduced at each internal line, and then 4-point vertices are added at each 3-point vertex. The number of 3-point vertices is determined by tallying how many times a negative number appears in the list, with each negative number appearing at most 4 times.

The next step is to assign momenta to the external and internal lines. For a generic diagram with N external lines, P internal lines , L loops and V_3, V_4 the number of 3 and 4 point vertices respectively, the number of loops is equal to the number of independent momenta:

$$L = P - V_3 - V_4 + 1 \tag{2}$$

in tree-level diagrams $L = 0$, so that we have $V_3 + V_4 = P + 1$. Imposing momentum conservation at each vertex provides $P + 1$ equations. However, there are only P unknowns, meaning the system is underdetermined. To solve this, we must include the constraint of global momentum conservation, which removes one of the equations, since momentum conservation in each vertex imply the global momentum consevation. All the previous steps are implemented in the function `GenerateDiagrams` in listing 2.

Code 2: Generating Feynman diagrams

```
1 Diagrams[graph_] (* Generates diagrams from a graph, calculating internal momenta and Lorentz and color indices for the vertices and
      propagators *)
2 generatediagrams[n_] (* Generates diagrams with n external points and only 3 and 4-point vertices *)
3 generatediagrams3[n_] (* Generates diagrams with n external points and only 3-point vertices *)
```

To complete the Feynman diagrams, we need to include propagators and vertices, each with momenta, color, and Lorentz indices fig. 2. We can achieve this by using Mathematica's list manipulation and pattern matching capabilities, constructing vertices and propagators as functions that take momentum, Lorentz indices, and color indices as arguments.

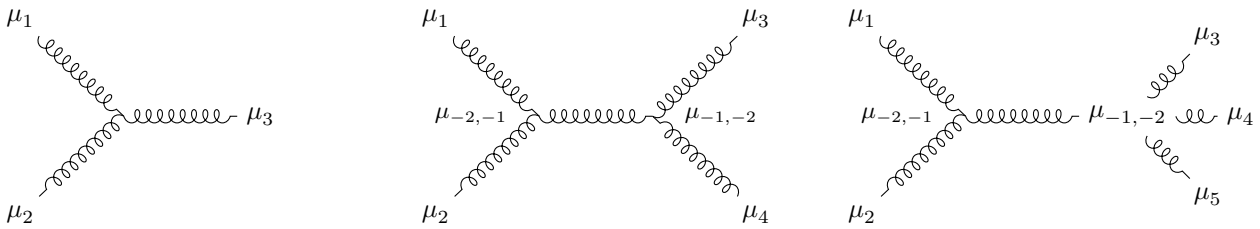

Figure 2: Each external leg is assigned a Lorentz index μ_i and each vertex j a Lorentz indices $\mu_{i,j}$. The same logic is applied for the color indices.

.

2.2 Feynman Rules

The QCD Feynman rules are implemented in Mathematica as follows:

Code 3: Feynman rules for QCD

```
gluon = prop[p_, {mu1_, mu2_}, {c1_,c2_}] :> -I myDelta[c1, c2] gprop[p, mu1, mu2];
propagator[p_, mu1_, mu2_] := SP[{mu1}, {mu2}]/SP[p, p]
V3QCD = V[i_][{p1_, p2_, p3_}, {mu1_, mu2_, mu3_}, {c1_, c2_, c3_}] :> -g f[c1, c2, c3] V3Lorentz[{p1, p2, p3}, {mu1, mu2,
    mu3}]
V3p[{p1_, p2_, p3_}, {mu1_, mu2_, mu3_}] := SP[p1 - p2, {mu3}] SP[{mu1}, {mu2}] + SP[p2 - p3, {mu1}] SP[{mu2}, {mu3}] +
    SP[p3 - p1, {mu2}] SP[{mu3}, {mu1}]
V4QCD = V[i_][{p1_, p2_, p3_, p4_}, {mu1_, mu2_, mu3_, mu4_}, {c1_, c2_, c3_, c4_}] :> -I g^2
      (f[c1, c2, c[x, i]] f[c3, c4, c[x, i]] (SP[{mu1}, {mu3}] SP[{mu2}, {mu4}] - SP[{mu1}, {mu4}] SP[{mu2}, {mu3}])
      - f[c1, c3, c[x, i]] f[c2, c4, c[x, i]] (SP[{mu1}, {mu4}] SP[{mu2}, {mu3}] - SP[{mu1}, {mu2}] SP[{mu3}, {mu4}])
      + f[c1, c4, c[x, i]] f[c2, c3, c[x, i]] (SP[{mu1}, {mu2}] SP[{mu3}, {mu4}] - SP[{mu1}, {mu3}] SP[{mu2}, {mu4}]))
SetAttributes[myDelta, Orderless]
ColorDelta = # //. {myDelta[a_, b_] myDelta[a_, b_] :> (Nc^2 - 1) ,
    myDelta[a_, a_] :> (Nc^2 - 1),
    myDelta[a_, b_]*expr_ :> (expr /. b -> a),
    c[i_, j_] :> a[i,j]} &
FeynmanRules = (# /. gluon /. V3QCD /. V4QCD /.gprop -> propagator /. V3Lorentz -> V3p // ColorDelta // antisymf // Expand)
    &
```

where Lorentz structure and color structure are factorized when possible so we can manipulate them separately. This property is especially useful for generating all the distinct products of structure constants f^{abc} for a given number of external legs. The Feynman rules are applied to the Feynman diagrams using the `Replace` function, which substitutes the propagators and vertices with their corresponding expressions.

2.3 Color Algebra and Lorentz

The color algebra is handled using the following functions:

- **antisymf**: This functions applies the antisymmetry property of the structure constants f^{abc}, and brings the color indices in decreasing alfanumeric order.
- **frule**: This functions acts on products of structure constants, it counts the number of repeated indices and substitute them with $a[i]$. For $n \geq 5$ there are more than one repeated indices and the degeneracy associated with the permutation of the dummy indices must be taken into account.
- **Color**: This function acts on fully contracted products of structure constants and it calculates the results in terms of N_c, the dimensions of the fundamental representation of the $SU(3)$ group. This function converts the products of structure constants into its fundamental representation, using the following identity:

$$f^{abc} = -2iTr\left([T^a, T^b]T^c\right) \tag{3}$$

where T^a are the generators of the $SU(3)$ group in the fundamental representation. The trace is expressed as a product of matrices with cyclic indices, and the the Fierz identity is utilized to convert the product of T^a into a product of Kronecker deltas:

$$T^a_{i,j}T^a_{k,l} = \frac{1}{2}\left(\delta_{i,l}\delta_{j,k} - \frac{1}{N_c}\delta_{i,j}\delta_{k,l}\right) \tag{4}$$

This conversion allows us to express the color algebra in terms of Kronecker deltas, which can be manipulated in Mathematica.

However this algorithm is not highly efficient. Before all the Kronecker deltas are contracted, the number of terms grows exponentially. For n gluons, there are $2(n-2)$ fully contracted structure constants, converting to traces gives $2^{2(n-2)}$ products of $2(n-2)$ traces each, writing the traces as products of matrices we get $3*2(n-2)$ matrices, and using Fierz it becomes $2^{2(n-2)*3}$ terms, in total we have $2^{2(n-2)*3+2(n-2)} = 2^{8(n-2)}$ Kronecker deltas to contract, making it a very inefficient algorithm see table 1. It is possible to reduce the number of terms by using known identities before applying the

Table 1: Testing Color for various cases

n	product	Result	Time (s)
4	$f^{c_3c_1a_1}f^{c_3c_2b_1}f^{c_4c_1b_1}f^{c_4c_2a_1}$	$\frac{Nc^2}{2}(N_c^2-1)$	0.0625
5	$f^{c_1b_2b_1}f^{c_3c_2a_2}f^{c_3c_2b_1}f^{c_4a_2a_1}f^{c_5c_1a_1}f^{c_5c_4b_2}$	$-\frac{Nc^3}{2}(N_c^2-1)$	0.5868
6	$f^{c_1a_3a_1}f^{c_1b_3b_1}f^{c_3c_2a_1}f^{c_3c_2b_1}f^{c_4b_3b_2}f^{c_5a_3a_2}f^{c_6c_4a_2}f^{c_6c_5b_2}$	$\frac{Nc^4}{2}(N_c^2-1)$	25.6701

Fierz identity, for example: $Tr(T^aT^a) = Nc^2/2$. Further speed up can be obtained by dropping the $-\frac{1}{Nc}$ term in the Fierz identity, since it cancels out in the final result for gluon scattering amplitudes.

Lorentz algebra is implemented using the `SP` function, which is a wrapper for the `Dot` function. This function is used to contract Lorentz indices and it is defined as follows:

Code 4: Lorentz algebra

```
SetAttributes[SP, Orderless]
Contract = (# //. {
  SP[a_, {mu_}] SP[b_, {mu_}] :> SP[a, b], (*Contraction*) SP[p_, {mu_}] SP[p_, {mu_}] :> SP[p, p],(*Square*)
  SP[a_Integer *p_, {mu_}] :> a SP[p, {mu}],(*Homogeneity*) SP[a_Integer *p_, q_] :> a SP[p, q], (*Homogeneity*)
  SP[p_ + q_, {mu_}] :> SP[p, {mu}] + SP[q, {mu}],(*Linearity*) SP[a_, b_ + c_] :> (SP[a, b] + SP[a, c]),(*Distributive property*)
  SP[{mu_}, {mu_}] :> 4 (*Trace*), SP[n, n] :> 0 (*Light Cone Gauge*), SP[p[i_], p[i_]] :> 0 (*On Shell Condition*)
  }) &;
```

3 Benchmarks and Calculations

Armed with the framework for constructing Feynman diagrams and applying Feynman rules, we can now proceed to compute the scattering amplitudes for gluon interactions in (QCD) and verify their properties.

Before diving into the calculations, it's important to understand the expected complexities arising from the generated amplitudes.

3.1 Number of Feynman Diagrams and Color Structures

The total number of tree-level Feynman diagrams for n gluons with 3 and 4 point vertices can be computed using a recurrence relation[2] and shown in Listing 1. Among these, the number of diagrams featuring only 3-point vertices is given by $(2n-5)!!$ (double factorial) [3].Each of these diagrams corresponds to a unique color substructure. The remaining diagrams, which include at least one 4-point vertex, are generated by adding a 4-point vertex to existing diagrams with only 3-point vertices and they do not introduce any new color substructures.

In general, for n gluons, there are $(2n-5)!!$ color substructures.These takes the form of contractions of $n-2$ structure constants f^{abc} with $n-3$ dummy(summed over) indices a_i and n distinct indices c_i, e.g $f^{c_2c_1a_1}f^{c_4c_3a_1}$ for $n=4$, $f^{c_1a_2a_1}f^{c_3c_3a_1}f^{c_5c_4a_2}$ for $n=5$, and so on.

Table 2: Number of Tree-level Feynman Diagrams for n gluons. The first column indicates the number of external gluons, the second column shows the total number of diagrams, the third column counts the diagrams with only 3-point vertices, and the fourth column counts those with at least one 4-point vertex.

n	Number of Diagrams	3-point Vertices only	with 4-point Vertices
4	4	3	1
5	25	15	10
6	220	105	115
7	2485	945	1540
8	34300	10395	23905

3.2 Amplitude Generation The process of generating the scattering amplitude is straightforward. We generate the Feynman diagrams using the 'generatediagrams' function (listing 2), sum the contributions from each diagram, and then apply the Feynman rules to obtain the amplitude.

The computational resources required for generating the diagrams only, specifically the time and memory, are summarized in Table 3, which shows that time and memory usage grows exponentially with the number of gluons involved in the scattering process. Applying the Feynman rules further increases the computational cost, as shown in the same table.

Table 3: Computational Resources for Feynman Diagram Generation and Amplitude. The Amplitude is generated by applying the operations: Total, FeynmanRules and Expand. The result for 8 gluons is not available (TBD = To Be Determined) due to the high computational cost.

Number of Gluons		**Generated Diagrams**		**Amplitude**	
n	**Number of Diagrams**	**Time (s)**	**Memory (MB)**	**Time (s)**	**Memory (MB)**
4	4	0.0011	0.01	0.004	0.096
5	25	0.0100	0.12	0.051	2.87
6	220	0.1153	1.51	6.503	104.52
7	2485	1.5998	22.99	1112.738	4430.25
8	34300	27.5766	404.70	TBD	TBD

After generating the amplitude, there are several important properties that we can verify to ensure the correctness of our calculations.

3.3 Symmetry under Exchange of External Legs The scattering amplitude for gluon interactions should be symmetric under the exchange of external legs.

To veriry this property, the following function are defined:

- **swapTwoParticles[amp_,i_,j_]**: This function takes an amplitude and swaps the i-th and j-th external legs, by replacing the corresponding Lorentz, color and momentum labels in the amplitude expression.
- **pairmap[list1_, list2_]** : This function takes two lists and returns a list of pairs, where each pair consists of an element from the first list and the corresponding element from the second list.

In the two to two scattering case ($n = 4$), there are only 3 color substructures and after exchangin external legs, these color substructures are permuted among themselves, their Lorentz coefficients also changes accordingly, but the overall amplitude remains unchanged.

For $n \geq 5$, the permutation are more complex and the function **pairmap** is needed to keep track of which term goes where. Then verify that the amplitude remains unchanged by subtracting the original color substructure Lorentz coefficients from the swapped ones and checking if the result is zero.

From this we can conclude that the amplitude is symmetric under the exchange of external legs and that all the color substructures are permuted among themselves, so only a single color substructure is needed and rest can be generated by substitution rules, which is significatly more efficient than all the diagrams, substituting Feynman rules, summing over all diagrams and then collecting the color substructures.

It is possible to generate the Lorentz coefficient for a single color substructure by generating only the diagrams contributing to that color substructures. This has been implemented in the function `generatemasterdiagram[n_]`, which generates diagrams which contributes to the color substructure $f^{c_1 c_2 a_1} f^{c_3 a_1 a_2} \cdots f^{c_n c_{n-1} a_1}$.

The bottleneck given by the generation of Feynman diagrams has thus been reduced to a symmetry problem, given this color substructure one has to find the correct permutation to obtain all the other color substructures.

3.4 Ward Identities The Ward identities are a set of relations that must be satisfied by the scattering amplitudes in gauge theories. In the case of gluon scattering, the Ward identity states that the amplitude must vanish when any external on-shell gluon polarization is substituted with its momentum.

While the overall scattering amplitude for gluon interactions satisfies the Ward identity, the presence of distinct color substructures within the amplitude poses a unique challenge. Since these color substructures prevent a direct summation of the various Lorentz structures, each individual color substructure must inherently be gauge invariant for the full amplitude to satisfy the identity.

However, the $(2n-5)!!$ color substructures are not all linearly independent, as they are related by the Jacobi identity as shown in Equation eq. (5) for the structure constants f^{abc}, so they are not gauge invariant. These non gauge invariant dependent substructures still vanish when all external polarization vectors are simultaneously substituted with their respective momentum vectors.

$$f^{aeb}f^{ecd} + f^{ade}f^{ecb} - f^{ace}f^{edb} = 0 \tag{5}$$

For n gluons, each of these color substructures has $n-3$ dummy indices, so each term can generate $n-3$ jacobi identities, though these may not all be unique. These color substructures can be mapped to variables using the Mathematica function **MapIndexed** to variables $v[i]$ so that Mathematica can work with them. The Jacobi identities can then be transformed into a system of equations and solved using **Solve**.

In conclusion, the initial set of $(2n-5)!!$ color substructures reduces to $(n-2)!$ independent color substructures [4]. These independent color substructures are each accompanied by their respective Lorentz structures, which are inherently gauge invariant and collectively satisfy the Ward identity.

Thanks to the symmetry under exchange, it is sufficient to verify the Ward identity for just one of these independent color substructures.

Necessary conditions for the Ward identity are momentum conservation and the transversality of the polarization vectors.

All the steps above are implemented in the module `VerifyWard[ngluons]`, in Table 4 we summarize the performance of the verification of the Ward identity for $n = 4, 5, 6$ gluons. Notice that times and memory usage scales exponentially with the number of gluons and simplification times depends on the size of the expressions,

3.5 Modulus Squared of the Amplitude Calculating the modulus squared of the amplitude is typically the most computationally intensive step in scattering amplitude computations. For gluon scattering, various strategies can be employed to manage this complexity, primarily by organizing the amplitude and its color substructures differently.

1. **Direct Squaring of Feynman Diagrams:** This is the most straightforward method, involving the direct squaring of the sum of all Feynman diagrams.

2. **Collecting Non-Independent Color Substructures:** A more efficient approach involves collecting all $(2n-5)!!$ non-independent color substructures before squaring their sum. This strategy significantly improves efficiency by allowing for early cancellations, thereby substantially reducing the size of intermediate expressions and overall memory footprint.

3. **Utilizing Jacobi Identities for Independent Color Substructures:** This method further reduces the number of color substructures to $(n-2)!$ by employing Jacobi identities to derive a basis of independent color structures. While this approach minimizes the number of terms that need to be computed, it can lead to coefficients with larger individual expressions. This presents a trade-off between the number of calculations performed and the symbolic size of the resulting terms.

To illustrate the computational characteristics of these approaches, Table 5 presents performance metrics, including memory usage and computation time, for $n = 4$ and $n = 5$ gluon scattering processes. The memory usage is detailed for various stages of the calculation: the size of a single selected term, its size after contraction with polarization sums (using specific choices for

Table 4: Performance Metrics for Ward Identity Verification in Gluon Scattering

Contracted with	Contraction Time (s)	Memory (MB)	Simplification Time (s)
$n = 4$ Gluons			
4p 0ϵ	0.0023	0.00002	0.00008
3p 1ϵ	0.0038	0.00350	0.00008
2p 2ϵ	0.0069	0.02103	0.00016
1p 3ϵ	0.0099	0.04923	0.00020
$n = 5$ Gluons			
5p 0ϵ	0.093	0.077	0.00034
4p 1ϵ	0.134	0.162	0.00062
3p 2ϵ	0.203	0.515	0.00160
2p 3ϵ	0.281	0.952	0.00327
1p 4ϵ	0.376	1.561	0.07672
$n = 6$ Gluons			
6p 0ϵ	4.04	2.67	0.23
5p 1ϵ	5.08	4.52	0.79
4p 2ϵ	7.03	11.48	9.23
3p 3ϵ	9.64	22.11	30.04
2p 4ϵ	12.59	37.03	151.34
1p 5ϵ	15.70	54.53	183.86

* The "Memory" column indicates the memory (in Megabytes) required to store the expression just before the simplification step. The "Contraction Time" includes applying momentum conservation, the transverse condition, and contractions. The "Simplification Time" reflects the duration of the `Simplify` operation.

polarization vectors n^μ aligned with external momenta for simplification), the size after expanding against a conjugate term, the size after subsequent contraction, after renaming Lorentz invariants, and finally, after full simplification. Lastly, the total time taken to compute the modulus squared is also recorded.

The correctness of the modulus squared calculation has been verified by comparing with the results in [5], [6].

4 Conclusions

This report highlights Mathematica's capability in efficiently computing tree-level gluon scattering amplitudes in QCD. We developed a framework that automates the generation of Feynman diagrams and the application of Feynman rules, significantly streamlining the derivation of amplitudes. However, the growth of the number of diagrams is super-exponential, demonstrating that while computational tools are essential, they alone cannot manage the rapidly increasing algebraic complexity.

We validated the framework's robustness by confirming its symmetry under external leg exchange and its consistency with Ward identities, both of which exhibited exponential resource scaling. Ultimately, while Mathematica proves to be a powerful tool for addressing this problem, the exponential complexity inherent in gluon scattering amplitudes necessitates a more sophisticated approach that leverages symmetry to manage the growth effectively

References

[1] W. R. Inc., *Mathematica, Version 14.2*, Champaign, IL, 2024. [Online]. Available: https://www.wolfram.com/mathematica.

[2] N. J. A. Sloane, *A268163: Number of non-isomorphic 3-regular graphs on n nodes*, https://oeis.org/A268163, The On-Line Encyclopedia of Integer Sequences, 2015.

Table 5: Performance Metrics for Modulus Squared Calculation in Gluon Scattering

Operation Stage	**Feynman Diagrams**	**Non-Independent Basis**	**Independent Basis**
$n = 4$ Gluons			
Total Terms	10	6	3
Single Term (MB)	0.028	0.025	0.050
Times Polarization (MB)	0.945	0.693	1.512
Times Conjugate (MB)	15.449	4.186	32.592
Contract (MB)	1.364	0.358	2.220
Mandelstam (MB)	0.389	0.116	0.622
Simplify (MB)	0.005	0.003	0.006
All Contributions (MB)	0.026	0.023	0.018
Time Taken (s)	3.318	1.106	3.184
$n = 5$ Gluons			
Total Terms	325	120	21
Single Term (MB)	0.134	0.133	0.633
Times Polarization (MB)	10.455	8.970	56.295
Times Conjugate (MB)	4893.640	3695.500	TBD
Contract (MB)	6.325	1.553	TBD
Mandelstam (MB)	7.760	1.681	TBD
Simplify (MB)	0.072	0.059	TBD
All Contributions (MB)	TBD	6.867	TBD
Time Taken (min)	TBD	$\sim$50	TBD

Note: the most memory-intensive operation is the expansion against a conjugate term, which is necessary for Contraction to act on the expression.

[3] N. J. A. Sloane, *A001147: Number of permutations of length n with no 1-cycles*, https://oeis.org/A001147, The On-Line Encyclopedia of Integer Sequences, 2015.

[4] V. Del Duca, L. Dixon, and F. Maltoni, "New color decompositions for gauge amplitudes at tree and loop level," *Nuclear Physics B*, vol. 571, no. 1, pp. 51–70, 2000, ISSN: 0550-3213. DOI: https://doi.org/10.1016/S0550-3213(99)00809-3. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0550321399008093.

[5] R. K. Ellis, W. J. Stirling, and B. R. Webber, *QCD and collider physics*. Cambridge University Press, Feb. 2011, vol. 8, ISBN: 978-0-511-82328-2, 978-0-521-54589-1. DOI: 10.1017/CBO9780511628788.

[6] F. Berends, R. Kleiss, P. De Causmaecker, R. Gastmans, and T. T. Wu, "Single bremsstrahlung processes in gauge theories," *Physics Letters B*, vol. 103, no. 2, pp. 124–128, 1981, ISSN: 0370-2693. DOI: https://doi.org/10.1016/0370-2693(81)90685-7. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0370269381906857.