Inverse Cachazo-He-Yuan integrand construction from factorization channels with exact integer constraint propagation

Simeng Li[*] and Yaobo Zhang[†]
School of Physics, Ningxia University, Yinchuan, 750021, China

(Received 22 January 2026; accepted 23 March 2026; published 12 May 2026)

In the Cachazo-He-Yuan (CHY) representation, many tree-level scattering amplitudes are written as integrals localized by the scattering equations, where the CHY integrand encodes the theory and must reproduce the physical factorization pattern of Mandelstam singularities. The inverse CHY problem therefore asks for a rational integrand whose physical poles match a prescribed set of factorization channels, a task that becomes rapidly combinatorial at higher multiplicity. We present an algorithmic approach that makes the subset structure of pole data explicit by assigning each particle subset A a generalized pole degree $K(A)$, realized as a signed internal-edge count in a colored integrand graph. Additivity under integrand multiplication and an elementary face recursion on the subset lattice imply that all higher-channel $K(A)$ are linear functions of the two-particle integers $\{K(s_{ij})\}$, so the inverse step becomes a mixed-integer linear feasibility problem. The same subset lattice provides a fixed computation graph for exact constraint satisfaction: pole requirements and recursion identities are treated as hard constraints, updates are restricted to those that preserve feasibility, and whenever a proposed change would enter a fixed node its constraint residual is rerouted and redistributed through unfixed degrees of freedom so that all constraints remain satisfied throughout. After a factorial rescaling, every local propagation update is purely integral, keeping the entire procedure exact in integer arithmetic and avoiding *a posteriori* reconstruction from numerical evaluations. Since physical tree-level amplitudes must contain only simple factorization poles as dictated by unitarity and locality, we further organize generalized integrands by an n-regular grading under multiplication, where degree-zero factors act as Möbius-invariant insertions admitting a decomposition into products of four-point cross ratios, which guides efficient higher-order pole reduction. We illustrate the construction at six and eight points, including pole selection and higher-order pole reduction.

DOI: 10.1103/nczg-mn96

I. INTRODUCTION

The Cachazo–He–Yuan (CHY) formalism [1–3] expresses many tree-level scattering amplitudes in massless theories (e.g., Yang–Mills, gravity, and scalar models) as integrals over the moduli space of n marked points on the Riemann sphere. The scattering equations localize these integrals to a finite set of solutions fixed by the external kinematics [1–3].

Given a CHY integrand, evaluating the corresponding CHY integral (the *forward CHY problem*) can be approached by combinatorial integration rules and diagrammatic correspondences for simple poles [4,5], as well as by contour-integral algorithms on $\mathcal{M}_{0,n}$ [6]. Reduction schemes for higher-order poles based on cross-ratio identities provide a further systematic handle [7–9]. The complementary *inverse CHY problem* asks for a rational CHY integrand whose physical singularities match a prescribed set of poles [10]. Existing inverse constructions are mostly limited to special settings (for example, reconstructing Parke-Taylor data from sets of cubic diagrams in bi-adjoint scalar theory [10]), and the inverse direction is discrete and becomes rapidly combinatorial at higher multiplicity and with nontrivial numerators.

In this paper we present an algorithmic approach to the inverse CHY problem that makes the subset structure of pole data explicit. We organize pole information using a generalized pole degree $K(s_A)$, defined as a signed internal-edge count for each subset $A \subset \{1,\ldots,n\}$ in the standard 4-regular graph representation of CHY integrands [4,5,7]. Additivity under integrand multiplication and local face-recursion constraints on the subset lattice imply that all higher-channel $K(s_A)$ are linear functions of the two-particle data $\{K(s_{ij})\}$, so the inverse step can be formulated as a mixed-integer linear feasibility problem [11,12]. The same

[*]Contact author: lisimeng1@stu.nxu.edu.cn
[†]Contact author: yaobozhang@nxu.edu.cn

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by SCOAP[3].

subset lattice also provides a fixed dependency graph for deterministic constraint propagation. After a factorial rescaling, every local update can be implemented using only integer additions, so the propagation remains exact in the rescaled variables. This avoids workflows where exact expressions are recovered *a posteriori* from numerical evaluations [13–18].

A. Background: The CHY formalism and computational motivation

At tree level, the CHY representation writes an n-particle scattering amplitude $\mathcal{A}_n$ as an integral over puncture positions $\{z_i\}$ on the Riemann sphere [3,19]

$$\mathcal{A}_n = \int \frac{\prod_{i=1}^{n} dz_i}{\mathrm{vol}(\mathrm{SL}(2,\mathbb{C}))} \Omega(\mathcal{E}) \mathcal{I}(z). \tag{1}$$

The measure $\Omega(\mathcal{E})$ imposes the scattering equations

$$\mathcal{E}_a \equiv \sum_{b \neq a} \frac{s_{ab}}{z_a - z_b} = 0, \tag{2}$$

which implement momentum conservation and on-shell kinematics in this representation [3]. Different quantum field theories correspond to different choices of the integrand $\mathcal{I}(z)$. The formalism was introduced for tree-level amplitudes in a range of massless theories [3,19], and later extended to polynomial forms [20].

Beyond the original tree-level setting, CHY-inspired structures have been explored in several directions [21–25]. For example, factorization properties of Yang-Mills amplitudes have been studied within the CHY framework [21,22]. Loop-level generalizations and related world sheet formulations have also been explored, including forward-limit constructions [23] and the connection to ambitwistor string theory [24,25]. We will not review these developments here; our focus is the inverse problem at tree level formulated in terms of pole data.

On the computational side, a variety of tools exist for manipulating and evaluating CHY integrals [4–6,13]. The "pick pole" algorithm isolates contributions containing specified physical singularities $1/s_A$ and has been implemented for biadjoint scalar theory [26,27]. Extensions to more general integrands, including labeled tree-graph representations [28] and the appearance of higher-order poles [7], motivate systematic methods that organize pole constraints and help manage the associated combinatorics.

More broadly, the CHY formalism has stimulated interactions with tropical geometry [29], positive geometries [30], and twisted (co)homology/intersection-theoretic formulations of worldsheet integrals on $\mathcal{M}_{0,n}$ [31–34]. Motivated by these connections, we focus on discrete optimization and combinatorial/topological structures as a way to organize the pole hierarchy and to formulate inverse-construction tasks.

Machine-learning methods have been applied in several areas of theoretical and computational physics, including scans of Calabi-Yau data [35–37] and amplitude-adjacent tasks such as phase-space integration and multiloop integral evaluation [38,39], as well as symbolic or diagnostic applications [40–43]. Here we invoke neural-network language in a different, constraint-driven sense. Rather than fitting a flexible function class to sampled data through a loss, we exploit the fact that the inverse CHY problem comes with rigid discrete supervision: the target pole pattern together with exact recursion relations on the subset lattice provide the effective "dataset." The discrete degrees of freedom are the two-particle integers $\{K(s_{ij})\}$, and higher-channel data are then fixed by the lattice recursion.

Exact results in amplitude calculations are often obtained by solving systems of rational constraints, for example in integration-by-parts (IBP) reductions over $\mathbb{Q}$ [44]. In many contemporary workflows, the target expressions are constrained indirectly through numerical evaluations, with exact dependence recovered *a posteriori* via finite-field sampling, functional reconstruction, or high-precision numerics followed by exactification procedures [14–18]. In contrast, in the inverse CHY setting the relevant degrees of freedom are discrete from the outset: the integrand data are integer edge multiplicities, and the pole hierarchy is governed by local recursion relations on the subset lattice. This structure allows the construction to be organized directly on a fixed computation graph for exact constraint satisfaction. Pole requirements and recursion identities are treated as hard constraints, updates are restricted to those that preserve feasibility, and whenever a proposed change would violate a fixed constraint its residual is rerouted and redistributed through unfixed degrees of freedom so that all constraints remain satisfied throughout. After a natural factorial rescaling, all local propagation updates are purely integral, keeping the entire procedure exact in integer arithmetic and avoiding any reliance on numerical reconstruction.

B. From graphs to combinatorial message passing

The pole hierarchy of a CHY integrand is indexed by particle subsets $A \subset \{1,\ldots,n\}$, ordered by inclusion. This inclusion poset can be viewed as the face poset of the full simplex on n labels, and it provides a fixed dependency graph on which local updates can be propagated [45–47]. In our setting, this graph plays the role of a fixed computation graph for exact constraint satisfaction: the target pole pattern together with recursion identities acts as hard constraints, and updates are restricted to those that preserve feasibility. Whenever a proposed change would enter a constrained node, the corresponding constraint residual is redistributed through unfixed degrees of freedom along the same graph so that all constraints remain satisfied throughout. Two structural properties provide local constraints on this poset:

(i) Additivity. Under integrand multiplication, pole data add: $K_{\alpha\cdot\beta}(A) = K_\alpha(A) + K_\beta(A)$.
(ii) Local face recursions. For $|A| \geq 3$,

$$K(A) = \frac{1}{|A|-2} \sum_{\substack{B\subset A \\ |B|=|A|-1}} K(B).$$

Together, these relations define a deterministic upward propagation on the subset lattice. To implement this propagation in a constraint-preserving manner using integer arithmetic, we introduce the factorial rescaling

$$\tilde{K}(A) \coloneqq (|A|-2)!K(A) \qquad (|A| \geq 2),$$

for which the recursion becomes an unnormalized face sum,

$$\tilde{K}(A) = \sum_{\substack{B\subset A \\ |B|=|A|-1}} \tilde{K}(B), \qquad |A| \geq 3.$$

In this representation, every local update is purely integral. With these ingredients, inverse CHY construction and pole selection reduce to a discrete constraint-satisfaction problem on a fixed dependency graph: the unknown degrees of freedom are the integer two-particle data $\{K(s_{ij})\}$, the recursion deterministically fixes all higher-subset values, and pole requirements appear as hard constraints on selected nodes. Feasibility is enforced by mixed-integer solving supplemented with deterministic integer message passing on the same graph. We analyze six- and eight-point examples to illustrate the resulting constraint propagation and MILP-based construction.

Beyond the familiar 4-regular graph representation that encodes many simple-pole CHY integrands (e.g., products of two Parke–Taylor factors) [4,5], we will work in a slightly more general colored-graph language in which each integrand carries an integer *regularity* (net degree) and integrand multiplication is graded (Sec. II B). In particular, the product of an m-regular and an n-regular graph is $(m+n)$-regular, while the degree-zero sector (0-regular graphs) consists of Möbius-invariant multiplicative factors. This graded multiplication is naturally suggested by the twisted (co)homology viewpoint of world sheet integrals on $\mathcal{M}_{0,n}$, where logarithmic singularities, cross-ratio coordinates, and intersection pairings provide a canonical bookkeeping of divisor data [31,32,34]. Practically, 0-regular factors give a constructive handle to generate families of integrands within a fixed regularity class and to engineer the cross-ratio insertions used to reduce higher-order poles [7–9].

1. Summary of contributions

(i) We formulate the inverse CHY problem as a mixed-integer linear feasibility problem in the two-particle data $\{K(s_{ij})\}$ using additivity and face-recursion constraints. (ii) We implement deterministic constraint propagation on the subset lattice using factorial-rescaled variables $\tilde{K}(A)$, so that updates can be kept integral. (iii) We introduce a regularity grading for generalized colored graphs and identify the 0-regular (degree-zero) sector of Möbius-invariant factors; we use 0-regular factorizations to construct cross-ratio factors to organize pick-pole selection and higher-order pole reduction on explicit low-point examples.

2. Organization

Section II reviews the graph-theoretic representation of CHY integrands and the associated pole data. Section III develops deterministic constraint propagation on the subset lattice and the resulting fixed message-passing architecture. Section IV presents applications to pick-pole selection and higher-order pole expansions. We conclude in Sec. V.

II. POLE CONSTRAINTS FROM COLORED GRAPHS

A. Graph representation and pole index/pole degree

The CHY formalism allows for a direct translation from the algebraic form of an integrand to a graph-theoretic object. For an n-particle process, the punctures z_i are treated as vertices of a graph. For each factor of $(z_i - z_j)^{-k}$ in the denominator of the integrand, k edges are drawn between vertices i and j. For the BAS integrand, $\mathcal{I}_{\rm BAS} = \mathrm{PT}(\alpha)\mathrm{PT}(\beta)$, each PT factor contributes edges between cyclically adjacent vertices in its respective ordering. Since each vertex k is connected to two neighbors in the α ordering and two neighbors in the β ordering, every vertex has exactly four edges connected to it. Such a graph, where every vertex has a degree of 4, is known as a 4-regular (or quartic) graph [4,5,24,48,49].

This graphical representation leads to a set of purely combinatorial integration rules, which provide a practical algorithm for evaluating the CHY integral without explicitly solving the scattering equations [4,5,7,48]. The final amplitude is a sum over terms, each being a product of inverse Mandelstam variables, $\prod 1/s_{A_i}$. The rules determine which sets of Mandelstam variables appear.

(1) Identify candidate factorization channels (pole index). For any nontrivial proper subset $A \subset \{1,\ldots,n\}$ with $2 \leq |A| \leq n-2$ (and identifying $A \sim \bar{A}$ since $s_A = s_{\bar{A}}$ by momentum conservation), define the internal-edge count

$$L[A] \coloneqq \text{number of edges with both endpoints in } A$$

(counted with multiplicity), and the pole index

$$\chi(A) \coloneqq L[A] - 2(|A|-1). \tag{3}$$

In the CHY integration rules for *simple-pole* integrands, a channel s_A can contribute only if $\chi(A) \geq 0$,

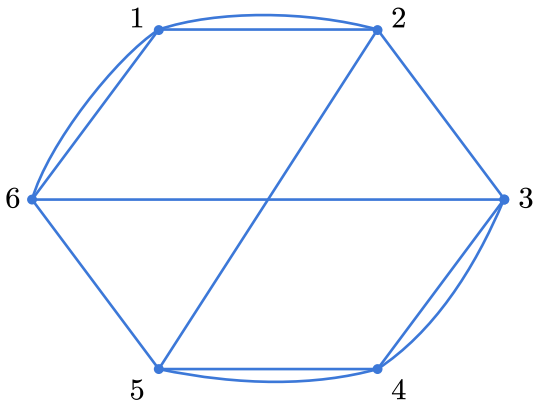


FIG. 1. The 4-regular multigraph for the BAS integrand $\mathcal{I} = \mathrm{PT}(123456)\mathrm{PT}(125436)$. Each Parke-Taylor factor contributes a cycle; shared edges appear with multiplicity two (drawn as double lines).

and $\chi(A) = 0$ corresponds to a *simple* pole $1/s_A$ (while $\chi(A) > 0$ signals potential higher-order poles at the integrand level, which typically require cross-ratio reduction before applying the simple-pole rules). For the biadjoint scalar (BAS) integrand $I = \mathrm{PT}(\alpha)\mathrm{PT}(\beta)$, all poles appearing in the cubic expansion are simple, i.e., $\chi(A) = 0$ for propagator channels [4,5,7].

(2) Find maximal compatible collections. A cubic tree with n external legs has $n-3$ internal propagators, so each term in the amplitude corresponds to a set of $n-3$ channels $\{A_1, \ldots, A_{n-3}\}$ with $\chi(A_i) = 0$ that are *pairwise compatible*. A standard compatibility criterion is: for any two subsets A, B, they are compatible iff

$$A \subset B, \quad \text{or} \quad B \subset A, \quad \text{or} \quad A \cap B = \varnothing,$$

i.e., the corresponding propagators can coexist in a single cubic Feynman diagram (no crossing) [4,5].

(3) Sum contributions (BAS/simple-pole case). For BAS (and more generally for simple-pole CHY integrands in this class), each maximal compatible set contributes

$$\frac{1}{s_{A_1} \cdots s_{A_{n-3}}}$$

to the CHY integral, and the full amplitude is the sum over all such compatible sets [4,5].

For the six-point biadjoint scalar (BAS) integrand

$$\mathcal{I}(z) = \mathrm{PT}(123456)\mathrm{PT}(125436), \tag{4}$$

the associated 4-regular multigraph is shown in Fig. 1. Each Parke-Taylor factor contributes a cyclic edge set

$$(12), (23), (34), (45), (56), (61) \quad \text{and}$$
$$(12), (25), (54), (43), (36), (61),$$

so the union graph has 12 edges counting multiplicity, and every vertex has degree 4. In particular, the edges (12) and (61) [as well as (34) and (45)] occur twice, which are drawn as double lines in Fig. 1.

The pole index is read off from induced subgraphs. As an explicit check, take the subset $A = \{1, 2, 6\}$. From Fig. 1, the induced subgraph on $\{1, 2, 6\}$ contains two edges between 1 and 2 and two edges between 1 and 6 (and no edge between 2 and 6), hence

$$L[A] = 4, \qquad \chi(A) = L[A] - 2(|A| - 1) = 4 - 2 \cdot 2 = 0, \tag{5}$$

so the channel s_{126} contributes a simple pole. (Equivalently, its complement $\bar{A} = \{3, 4, 5\}$ gives the same factorization invariant $s_{126} = s_{345}$ by momentum conservation).

Scanning all nontrivial subsets with $2 \le |A| \le n-2$, one finds that the independent simple-pole channels for this graph are

$$s_{12}, \qquad s_{16}, \qquad s_{34}, \qquad s_{45}, \qquad s_{126}(= s_{345}),$$

together with their complementary representatives. At six points, each cubic tree term contains $n - 3 = 3$ propagators, so we enumerate the *compatible* triples of $\chi = 0$ subsets (pairwise nested or disjoint). In this example, the allowed triples are

$$\{12, 34, 126\}, \qquad \{12, 45, 126\},$$
$$\{16, 34, 126\}, \qquad \{16, 45, 126\},$$

and the CHY integration rules therefore give

$$m(123456|125436) = \frac{1}{s_{12}s_{34}s_{126}} + \frac{1}{s_{12}s_{45}s_{126}} + \frac{1}{s_{16}s_{34}s_{126}} + \frac{1}{s_{16}s_{45}s_{126}}, \tag{6}$$

which matches the standard cubic-diagram expansion of the BAS double-partial amplitude for the two orderings [4,5].

B. Generalized pole degree $K(A)$ and additivity

The graph-theoretic methods outlined for the BAS theory can be extended into a more general framework. This allows for a unified combinatorial analysis of any theory described by a rational CHY integrand.

To begin, we generalize the graph representation itself. To accommodate any rational function of differences $(z_i - z_j)$, we introduce two types of edges:

(i) A factor of $(z_i - z_j)^{-1}$ in the denominator corresponds to a solid edge.

(ii) A factor of $(z_i - z_j)^{+1}$ in the numerator corresponds to a dashed edge.

This establishes a direct map between any rational integrand $I(z)$ and a corresponding graph with two edge types.

With this representation, we can define a local property for each vertex v, its net degree, which counts the net power of factors involving z_v in the integrand

$$\deg_{\text{net}}(v) \coloneqq \deg_{\text{solid}}(v) - \deg_{\text{dashed}}(v). \tag{7}$$

This concept of a local net count can be conceptually extended to a regional property for any subset of vertices $A \subset \{1, \ldots, n\}$. This gives rise to the generalized pole degree, K(A), which measures the net number of internal edges within the region defined by A

$$K(A) \coloneqq L_{\text{solid}}[A] - L_{\text{dashed}}[A] \tag{8}$$

where $L_{\text{solid}}[A]$ and $L_{\text{dashed}}[A]$ are the number of solid and dashed edges, respectively, with both endpoints in the set A. This new degree directly relates to the original pole index $\chi(A)$ found in the literature via the simple formula

$$\chi(A) = K(A) - 2(|A| - 1). \tag{9}$$

This relation elegantly separates the theory-dependent contribution to the pole structure, captured by $K(s_A)$, from the standard geometric contribution of the world sheet, $-2(|A| - 1)$.

This framework becomes useful when we consider the multiplication of integrands, as in the "double copy" construction. The multiplication of two graphs, G_α and G_β, corresponding to integrands $I_\alpha(z)$ and $I_\beta(z)$, is defined as a new graph $G_\gamma = G_\alpha * G_\beta$ corresponding to the product integrand $I_\alpha(z) I_\beta(z)$. The edge set of the product graph is the union of the edge sets of the original graphs. The index $K(s_A)$ behaves very simply under this operation.

Theorem. The pole degree $K(s_A)$ is additive under graph multiplication.

Proof. For a product graph $G_\gamma = G_\alpha * G_\beta$, the number of internal edges in a subset A is the sum of internal edges from the constituent graphs. Thus,

$$\begin{aligned} K_\gamma(A) &\coloneqq L_{\text{solid},\gamma}[A] - L_{\text{dashed},\gamma}[A] \\ &= (L_{\text{solid},\alpha}[A] + L_{\text{solid},\beta}[A]) \\ &\quad - (L_{\text{dashed},\alpha}[A] + L_{\text{dashed},\beta}[A]) \\ &= (L_{\text{solid},\alpha}[A] - L_{\text{dashed},\alpha}[A]) \\ &\quad + (L_{\text{solid},\beta}[A] - L_{\text{dashed},\beta}[A]) \\ &= K_\alpha(A) + K_\beta(A). \end{aligned}$$

■

This additivity provides a simple algebraic tool to compute the pole structure of complicated theories by summing the known indices of their simpler constituents.

Finally, we return to the local property of net degree to classify the global structure of graphs. A graph is called "n-regular" if all its vertices have a net degree of n. The additivity of the net degree under multiplication implies that the product of an m-regular and an n-regular graph is an $(m + n)$-regular graph. This implies that the set of "0-regular graphs" is closed under multiplication. A representative example is shown in Fig. 2. Furthermore, one can show that this set forms a commutative group, with the empty graph ($I(z) = 1$) as the identity and graph inversion (swapping solid and dashed edges) as the inverse operation. This "0-regular group" acts as a set of transformations on other graphs; multiplying an n-regular graph by any 0-regular graph yields another n-regular graph, preserving the graph's regularity class while modifying its detailed pole structure, as tracked by $K(s_A)$. Representative examples are shown in Figs. 3 and 4.

C. Subset-lattice recursion and integer rescaling

The generalized pole degree $K(A)$ obeys a simple recursion on the subset lattice. For any $A \subseteq \{1, \ldots, n\}$ with $|A| \geq 3$,

$$K(A) = \frac{1}{|A| - 2} \sum_{\substack{B \subset A \\ |B| = |A| - 1}} K(B). \tag{10}$$

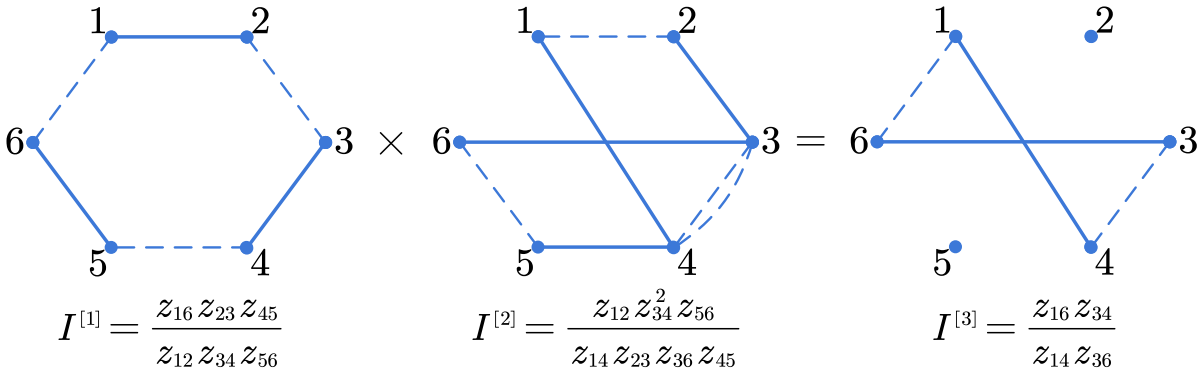


FIG. 2. The set of 0-regular graphs is closed under multiplication.

	$I^{[1]}$	$I^{[2]}$	$I^{[3]}$		$I^{[1]}$	$I^{[2]}$	$I^{[3]}$
$deg(s_1)$	0	0	0	$K(s_{35})$	1	−1	0
$deg(s_2)$	0	0	0	$K(s_{36})$	0	0	0
$deg(s_3)$	0	0	0	$K(s_{45})$	0	1	1
$deg(s_4)$	0	0	0	$K(s_{46})$	0	0	0
$deg(s_5)$	0	0	0	$K(s_{56})$	−1	0	−1
$deg(s_6)$	0	0	0	$K(s_{123})$	0	0	0
$K(s_{12})$	1	−1	0	$K(s_{124})$	1	0	1
$K(s_{13})$	0	0	0	$K(s_{125})$	1	−1	0
$K(s_{14})$	0	1	1	$K(s_{126})$	0	−1	−1
$K(s_{15})$	0	0	0	$K(s_{134})$	1	−1	0
$K(s_{16})$	−1	0	−1	$K(s_{135})$	0	0	0
$K(s_{23})$	−1	1	0	$K(s_{136})$	−1	1	0
$K(s_{24})$	0	0	0	$K(s_{145})$	−1	2	1
$K(s_{25})$	0	0	0	$K(s_{146})$	−1	1	0
$K(s_{26})$	0	0	0	$K(s_{156})$	0	−1	−1
$K(s_{34})$	1	−2	−1				

FIG. 3. The $K_{I^{[3]}}(s_A) = K_{I^{[1]}}(s_A) + K_{I^{[2]}}(s_A)$ for all poles.

We refer to (10) as the *elementary face recursion*. It is an edge-counting identity: any internal edge (i, j) with $i, j \in A$ is counted in $K(B)$ for exactly $|A| - 2$ of the $(|A| - 1)$-subsets $B = A\backslash\{k\}$ (namely all $k \neq i, j$). We record (10) because it makes all higher-channel values $K(A)$ *linear functions* of the two-particle data $\{K(ij)\}$, which is the key input for the mixed-integer feasibility formulation in Sec. II D and Appendix B.

Illustrative cases include $K(123) = K(12) + K(13) + K(23)$ and $K(1234) = \frac{1}{2}(K(123) + K(124) + K(134) + K(234))$.

More generally, averaging over codimension-d faces gives, for $1 \leq d \leq |A| - 2$,

$$K(A) = \frac{1}{\binom{|A|-2}{d}} \sum_{\substack{B\subset A \\ |B|=|A|-d}} K(B). \tag{11}$$

Different values of d provide redundant evaluations of the same $K(A)$ and serve as useful consistency checks in implementations.

D. Bootstrap constraints and Möbius/regularity conditions

The relationship between the generalized pole degree $K(s_A)$ and the pole index $\chi(A)$ established in Eq. (9) provides a useful framework for solving the inverse CHY problem. This inverse problem asks: given a desired set of poles in a scattering amplitude, can we construct a CHY integrand that produces exactly those poles?

From Eq. (9), we know that the pole index is determined by (9). A basic CHY integration rule tells us that only subsets A with $\chi(A) = 0$ contribute simple poles $1/s_A$ to the final amplitude [4,5]. This immediately gives us a classification scheme for any subset $A \subset \{1, \ldots, n\}$:

(i) Physical poles: $\chi(A) = 0 \rightarrow$ simple poles $1/s_A$ appear in the amplitude.
(ii) Absent poles: $\chi(A) < 0 \rightarrow$ no poles appear for these channels.
(iii) Higher-order poles: $\chi(A) > 0 \rightarrow$ would correspond to multiple poles, but these typically do not appear in tree-level amplitudes for generic theories.

This classification leads to a key insight: for any desired amplitude with a specific pole structure, the pattern of which poles appear and which are absent constrains the possible values of $K(s_A)$. Consider a target amplitude that has simple poles for a set $\mathcal{P} = \{A_1, A_2, \ldots, A_k\}$ and no poles for the complementary set $\bar{\mathcal{P}} := \mathcal{S}_n\backslash\mathcal{P}$, where $\mathcal{S}_n$ denotes the set of all subsets of $\{1, \ldots, n\}$. The bootstrap equations are

$$\text{For } A \in \mathcal{P}\text{: } K(A) = 2(|A| - 1) \tag{12}$$

$$\text{For } A \in \bar{\mathcal{P}}\text{: } K(A) < 2(|A| - 1). \tag{13}$$

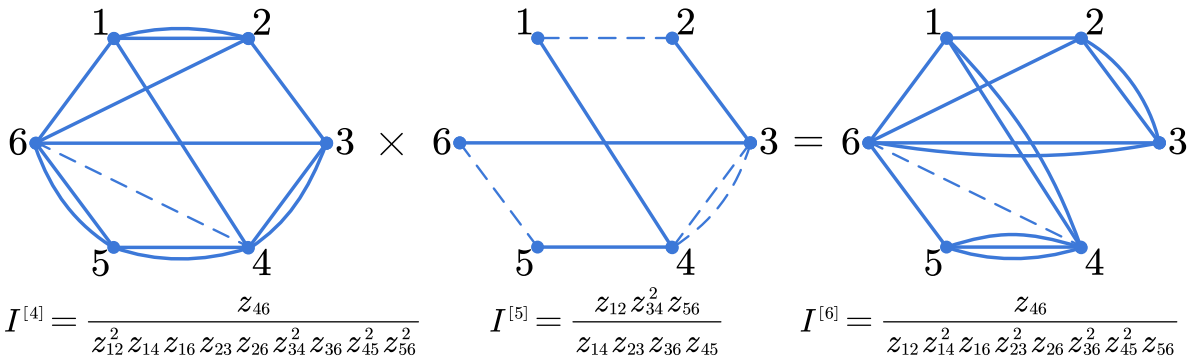


FIG. 4. The "0-regular group" acts as a set of transformations on other graphs.

Algorithm 1. CHY integrand bootstrap construction.

Require: Target pole structure $\mathcal{P}$ (set of desired simple poles)
Ensure: CHY integrand $I(z)$ producing amplitude with pole structure $\mathcal{P}$
Input: Specify desired pole structure $\mathcal{P} \subset \mathcal{S}_n$ where $\mathcal{S}_n$ is the set of all possible subsets
Constraint setup: For each $A \in \mathcal{P}$, impose $K(A) = 2(|A| - 1)$
Constraint setup: For each $A \in \bar{\mathcal{P}}$, impose $K(A) < 2(|A| - 1)$
Recursion matrix: Construct matrices $\mathbf{M}_{\text{eq}}$, $\mathbf{M}_{\text{ineq}}$ encoding relations between $K(A)$ and $K(s_{ij})$
Mixed integer system: Formulate $\mathbf{M}_{\text{eq}}\vec{K}_2 = \vec{b}_{\text{eq}}$ and $\mathbf{M}_{\text{ineq}}\vec{K}_2 < \vec{b}_{\text{ineq}}$
Integer solution: Solve for integer vector $\vec{K}_2$ satisfying the mixed system
if no integer solution exists **then**
  return "No CHY integrand exists for this pole structure"
end if
Decomposition: Use additivity $K_{\alpha\star\beta} = K_\alpha + K_\beta$ to factor solution
Construction: Build integrand $I(z) = \prod_i I_i(z)$ from elementary factors
return CHY integrand $I(z)$

The equality constraints ensure that the desired poles appear with the correct order, while the inequality constraints ensure that unwanted poles are absent.

However, the values $K(A)$ cannot be chosen arbitrarily—they must satisfy the recursion relations derived in Sec. II C. This creates a highly constrained system where the unknowns are the two-particle pole degrees $K(s_{ij})$ for all pairs $\{i, j\}$. Using the recursion relation (10), any higher-order pole degree can be expressed in terms of lower-order ones. For example

$$K(s_{ijk}) = K(s_{ij}) + K(s_{ik}) + K(s_{jk}) \tag{14}$$

$$K(s_{ijkl}) = \frac{K(s_{ijk}) + K(s_{ijl}) + K(s_{ikl}) + K(s_{jkl})}{2}. \tag{15}$$

Continuing this process, we can express all pole degrees $K(A)$ with $|A| \geq 3$ as linear combinations of the two-particle pole degrees $K(s_{ij})$. This reduces our bootstrap system to a set of linear constraints on the $\binom{n}{2}$ variables $\{K(s_{ij})\}_{1\leq i<j\leq n}$.

A crucial observation is that for rational CHY integrands constructed from products of Parke-Taylor factors and their generalizations considered here, all pole degrees $K(s_A)$ are integers. This follows from the fact that each factor $(z_i - z_j)^{\pm 1}$ contributes ± 1 to the edge count. Therefore, the bootstrap equations become a mixed system of linear Diophantine equations and inequalities

$$\mathbf{M}_{\text{eq}}\vec{K}_2 = \vec{b}_{\text{eq}} \tag{16}$$

$$\mathbf{M}_{\text{ineq}}\vec{K}_2 < \vec{b}_{\text{ineq}}, \tag{17}$$

where $\vec{K}_2 = (K(s_{12}), K(s_{13}), \ldots, K(s_{n-1,n}))^T$ is the vector of two-particle pole degrees, $\mathbf{M}_{\text{eq}}$ and $\mathbf{M}_{\text{ineq}}$ encode the recursion relations for equality and inequality constraints respectively, while $\vec{b}_{\text{eq}}$ and $\vec{b}_{\text{ineq}}$ encode the corresponding target pole structures.

The bootstrap approach yields a constructive algorithm for the inverse CHY problem. The basic idea is to transform the desired pole structure into a system of linear constraints on the two-particle variables $K(s_{ij})$, solve this constrained system, and then use the additivity property under graph multiplication to construct the corresponding integrand. The algorithm proceeds in the following steps: A high-level summary of the bootstrap construction is given in Algorithm 1.

For illustration, consider a six-particle amplitude where we want simple poles for s_{12}, s_{34}, and s_{56}, but no other two-particle poles. The bootstrap equations become $K(s_{12}) = K(s_{34}) = K(s_{56}) = 2$ and $K(s_{ij}) < 2$ for all other pairs $\{i, j\}$. Combined with the recursion relations for higher-order poles, this system constrains the possible CHY

	$I^{[4]}$	$I^{[5]}$	$I^{[6]}$		$I^{[4]}$	$I^{[5]}$	$I^{[6]}$
$deg(s_1)$	4	0	4	$K(s_{35})$	0	−1	−1
$deg(s_2)$	4	0	4	$K(s_{36})$	1	0	1
$deg(s_3)$	4	0	4	$K(s_{45})$	2	1	3
$deg(s_4)$	4	0	4	$K(s_{46})$	−1	0	−1
$deg(s_5)$	4	0	4	$K(s_{56})$	2	0	2
$deg(s_6)$	4	0	4	$K(s_{123})$	3	0	3
$K(s_{12})$	2	−1	1	$K(s_{124})$	3	0	3
$K(s_{13})$	0	0	0	$K(s_{125})$	2	−1	1
$K(s_{14})$	1	1	2	$K(s_{126})$	4	−1	3
$K(s_{15})$	0	0	0	$K(s_{134})$	3	−1	2
$K(s_{16})$	1	0	1	$K(s_{135})$	0	0	0
$K(s_{23})$	1	1	2	$K(s_{136})$	2	1	3
$K(s_{24})$	0	0	0	$K(s_{145})$	3	2	5
$K(s_{25})$	0	0	0	$K(s_{146})$	1	1	2
$K(s_{26})$	1	0	1	$K(s_{156})$	3	−1	2
$K(s_{34})$	2	−2	0				

FIG. 5. The $K_{I^{[6]}}(s_A) = K_{I^{[4]}}(s_A) + K_{I^{[5]}}(s_A)$ for all poles.

integrands. One solution corresponds to the integrand $I = \frac{1}{z_{12}^2 z_{14} z_{15} z_{23} z_{26} z_{34}^2 z_{36} z_{45} z_{56}^2}$, which indeed produces poles only in the appropriate channels.

This bootstrap framework thus provides a systematic method to reverse-engineer CHY integrands from desired amplitude structures, supporting amplitude construction and the exploration of additional theories within the CHY formalism. The method is particularly useful because it leverages the additive structure of the generalized pole degree $K(s_A)$, allowing complex integrands to be built systematically from simpler components. A representative additivity example is shown in Fig. 5.

E. MILP formulation (high-level) and a 6-pt example

To illustrate the utility of the bootstrap framework developed in Sec. II D, we present concrete examples demonstrating how to reverse-engineer CHY integrands from desired pole structures. These examples showcase both the computational implementation of the bootstrap algorithm and the interplay between the algebraic constraints and the resulting graph structures.

Consider a six-particle amplitude where we desire simple poles for the channels s_{12}, s_{23}, s_{34}, s_{56}, s_{123}, s_{156}, s_{234}, and s_{456}. This represents a highly constrained scenario with eight simple poles, approaching the maximum possible for a six-particle amplitude.

Following the bootstrap algorithm, we begin by translating these requirements into constraints on the K-values. The equality constraints (12) yield

$$K(s_{12}) = K(s_{23}) = K(s_{34}) = K(s_{56}) = 2 \tag{18}$$

$$K(s_{123}) = K(s_{156}) = K(s_{234}) = K(s_{456}) = 4. \tag{19}$$

For all other channels, the inequality constraints (13) require $K(s_A) < 2(|A| - 1)$. Additionally, the single-particle K-values must satisfy the regularity condition

$$K_i = \sum_{j \neq i} K(s_{ij}) = 4 \quad \text{for all } i \in \{1, \ldots, 6\}. \tag{20}$$

This yields a mixed integer linear programming problem with 15 two-particle variables $K(s_{ij})$, subject to 14 equality constraints (8 from desired poles, 6 from single-particle conditions) and 27 inequality constraints from unwanted channels. The constraint matrix $\mathbf{M}_{\text{eq}}$ encodes how each higher-order K-value depends on the two-particle degrees through the recursion relations. For instance, the constraint for $K(s_{123}) = 4$ translates to

$$K(s_{12}) + K(s_{13}) + K(s_{23}) = 4. \tag{21}$$

The complete system of inequalities includes 27 inequality constraints (from unwanted two-particle and three-particle channels)

$$\begin{aligned}
&K_{1,3} < 2, \quad K_{1,4} < 2, \quad K_{1,5} < 2, \quad K_{1,6} < 2, \quad K_{2,4} < 2, \quad K_{2,5} < 2, \quad K_{2,6} < 2,\\
&K_{3,5} < 2, \quad K_{3,6} < 2, \quad K_{4,5} < 2, \quad K_{4,6} < 2, \quad K_{1,2} + K_{1,4} + K_{2,4} < 4,\\
&K_{1,2} + K_{1,5} + K_{2,5} < 4, \quad K_{1,2} + K_{1,6} + K_{2,6} < 4, \quad K_{1,3} + K_{1,4} + K_{3,4} < 4,\\
&K_{1,3} + K_{1,5} + K_{3,5} < 4, \quad K_{1,3} + K_{1,6} + K_{3,6} < 4, \quad K_{1,4} + K_{1,5} + K_{4,5} < 4,\\
&K_{1,4} + K_{1,6} + K_{4,6} < 4, \quad K_{2,3} + K_{2,5} + K_{3,5} < 4, \quad K_{2,3} + K_{2,6} + K_{3,6} < 4,\\
&K_{2,4} + K_{2,5} + K_{4,5} < 4, \quad K_{2,4} + K_{2,6} + K_{4,6} < 4, \quad K_{2,5} + K_{2,6} + K_{5,6} < 4,\\
&K_{3,4} + K_{3,5} + K_{4,5} < 4, \quad K_{3,4} + K_{3,6} + K_{4,6} < 4, \quad K_{3,5} + K_{3,6} + K_{5,6} < 4,
\end{aligned}$$

and the complete system of 14 equality constraints (8 from desired poles, 6 from single-particle conditions)

$$\begin{aligned}
&K_{1,2} = 2, \quad K_{2,3} = 2, \quad K_{3,4} = 2, \quad K_{5,6} = 2, \quad K_{1,2} + K_{1,3} + K_{2,3} = 4,\\
&K_{1,5} + K_{1,6} + K_{5,6} = 4, \quad K_{4,5} + K_{4,6} + K_{5,6} = 4, \quad K_{2,3} + K_{2,4} + K_{3,4} = 4,\\
&K_1 = K_{1,2} + K_{1,3} + K_{1,4} + K_{1,5} + K_{1,6} = 4,\\
&K_2 = K_{1,2} + K_{2,3} + K_{2,4} + K_{2,5} + K_{2,6} = 4,\\
&K_3 = K_{1,3} + K_{2,3} + K_{3,4} + K_{3,5} + K_{3,6} = 4,\\
&K_4 = K_{1,4} + K_{2,4} + K_{3,4} + K_{4,5} + K_{4,6} = 4,\\
&K_5 = K_{1,5} + K_{2,5} + K_{3,5} + K_{4,5} + K_{5,6} = 4,\\
&K_6 = K_{1,6} + K_{2,6} + K_{3,6} + K_{4,6} + K_{5,6} = 4.
\end{aligned}$$

Solving this system yields the unique integer solution

$$\begin{aligned} &K_{1,2}\to 2, \quad K_{1,3}\to 0, \quad K_{1,4}\to 0,\\ &K_{2,3}\to 2, \quad K_{2,4}\to 0, \quad K_{2,5}\to 0,\\ &K_{2,6}\to 0, \quad K_{3,4}\to 2, \quad K_{3,5}\to 0,\\ &K_{3,6}\to 0, \quad K_{4,5}\to 1, \quad K_{4,6}\to 1,\\ &K_{5,6}\to 2, \quad K_{1,5}\to 1, \quad K_{1,6}\to 1. \end{aligned} \tag{22}$$

The resulting CHY integrand

$$I_6=\frac{1}{z_{12}^2 z_{15} z_{16} z_{23}^2 z_{34}^2 z_{45} z_{46} z_{56}^2} \tag{23}$$

shows that the bootstrap construction produces an integrand with the desired pole structure in this example. The resulting integrand I_6 generates an amplitude whose only simple poles are s_{12}, s_{23}, s_{34}, s_{56}, s_{123}, s_{156}, s_{234}, and s_{456}, exactly as specified in our requirements. The K-values directly translate to the powers in the integrand: each $K(s_{ij})=2$ yields a factor z_{ij}^{-2} in the denominator, while $K(s_{ij})=1$ gives z_{ij}^{-1}, and $K(s_{ij})=0$ means the corresponding factor is absent.

For this particular example, the mixed integer linear programming problem yields a unique solution, making it an ideal illustration of the bootstrap method. The constraint system can be solved using standard integer programming packages available in *Mathematica* (through functions like `LinearProgramming` with integer constraints) or Python (using libraries such as `cvxpy`, `PuLP`, or `scipy.optimize.milp`). These computational tools efficiently handle the combinatorial nature of finding integer solutions to our linear constraint system. More complex examples with nonunique solutions and detailed computational methods for solving the bootstrap equations are presented in Appendix B.

1. Summary

Representing a rational CHY integrand as a graph with solid/dashed edges turns the generalized pole degree $K(A)$ into a signed internal-edge count on any subset A. Two structural properties organize the inverse construction: (i) additivity under integrand multiplication, and (ii) the elementary face recursion (10), which expresses every higher-channel $K(A)$ as a linear combination of lower-layer values. As a result, all pole constraints reduce to linear equalities/inequalities in the two-particle variables $\{K(ij)\}$, so the inverse step can be posed as a mixed-integer feasibility problem (Appendix B). In Sec. III B we repackage the same subset lattice as a face poset/Hasse diagram to organize deterministic constraint propagation used by the pick-pole and pole-reduction algorithms.

III. DETERMINISTIC CONSTRAINT PROPAGATION ON THE SUBSET LATTICE

A. Integer propagation rules

The pole data of a CHY integrand are indexed by particle subsets $A\subset\{1,\ldots,n\}$, partially ordered by inclusion. For each A we attach the generalized pole degree $K(A)$ and the pole index $\chi(A)=K(A)-2(|A|-1)$. The elementary face recursion (10) expresses $K(A)$ for $|A|\geq 3$ as an average over its codimension-one subsets, so the hierarchy is determined by the two-particle layer $\{K(ij)\}$.

It is convenient to identify each subset A with a simplex σ_A of dimension $|A|-1$ in the full $(n-1)$-simplex. Then $B\subset A$ is the face relation $\sigma_B\prec\sigma_A$, and (10) can be written as a local averaging rule on the face poset (Hasse diagram)

$$K(\sigma_A)=\frac{1}{|A|-2}\sum_{\substack{B\subset A\\ |B|=|A|-1}} K(\sigma_B), \qquad |A|\geq 3. \tag{24}$$

We use this simplicial viewpoint mainly as a bookkeeping device for the subset lattice and recursion constraints. The channels with $\chi(A)=0$ are treated as a marked subset of faces (together with inequality constraints for $\chi(A)<0$ and higher-order poles), rather than as a subcomplex closed under taking subfaces. What we will use in practice is the induced fixed dependency graph (the inclusion lattice with marked/fixed nodes), which supports deterministic message passing and constraint propagation in Sec. III B. We do not rely on homological invariants in what follows.

B. Constraint-preserving residual redistribution

1. Scope and terminology

We borrow the language of combinatorial neural networks (CoNNs) to describe message passing on the fixed face lattice induced by the recursion relations. The analogy is architectural rather than statistical: a forward sweep deterministically evaluates higher-layer pole data from lower layers via the subset-lattice recursion, while a backward sweep propagates and redistributes discrete constraint residuals when an update attempts to enter a fixed node. In this constraint-driven setting, the target pole pattern together with the recursion identities provides the effective "dataset" (hard supervision), and the residual of a violated constraint plays the role of a discrete "loss." The discrete degrees of freedom are the two-particle integers $\{K(s_{ij})\}$, so what one might call "training" is replaced by a feasibility procedure combining mixed-integer linear solving for $\{K(s_{ij})\}$ with deterministic integer message passing on the same dependency graph. Consistency across higher channels is guaranteed by the recursion itself—a form of structural generalization that does not rely on statistical inference.

2. Integer message passing via factorial rescaling

Although the recursion for $K(A)$ contains the normalization factor $1/(|A|-2)$, the propagation can be implemented using strictly integer updates after a layer-dependent rescaling. Define

$$\tilde{K}(A) \coloneqq (|A|-2)!K(A) \qquad (|A| \geq 2).$$

Then

$$\tilde{K}(A) = \sum_{\substack{B \subset A \\ |B|=|A|-1}} \tilde{K}(B), \qquad |A| \geq 3,$$

so every local update along a face relation carries integer weight. This is exact within the integer-rescaled recursion variables: we only perform integer additions/subtractions on a fixed dependency graph.

Mainstream autodiff frameworks target differentiable real-valued programs; our integer, constraint-preserving propagation is outside their typical design scope, so we implement it explicitly.

3. Local influence weights (discrete sensitivities)

For adjacent faces $B \subset A$ with $|A| = |B| + 1$, a unit change in $\tilde{K}(B)$ contributes a unit change to $\tilde{K}(A)$. We use derivative notation as a compact bookkeeping device for these local influence weights

$$\frac{\partial \tilde{K}(A)}{\partial \tilde{K}(B)} = 1, \qquad B \subset A, \qquad |A| = |B| + 1. \quad (25)$$

Multistep influence is given by counting directed paths in the Hasse diagram. In the original K-normalization the corresponding coefficients are rational due to the $1/(|A|-2)$ factors. Equivalently, any update can be written as a linear combination of bottom-layer updates,

$$\Delta K(s_A) = \sum_{\substack{i,j \in A \\ i<j}} c_{A,ij} \Delta K(s_{ij}), \qquad c_{A,ij} \in \mathbb{Q},$$
$$\Delta \tilde{K}(s_A) = \sum_{\substack{i,j \in A \\ i<j}} \tilde{c}_{A,ij} \Delta \tilde{K}(s_{ij}), \qquad \tilde{c}_{A,ij} \in \mathbb{Z}. \quad (26)$$

4. Deterministic constraint propagation (CoNN-style message passing)

The recursion constraints define a fixed computation graph on faces. CoNN-style message passing on simplicial complexes is commonly written as updates that aggregate information from boundary and coboundary relations; for example,

$$h_\sigma^{(t+1)} = \phi\left(h_\sigma^{(t)}, \bigoplus_{\tau \in \mathcal{B}(\sigma)} m_{\tau \to \sigma}^{(t)}, \bigoplus_{\tau \in \mathcal{C}(\sigma)} m_{\tau \to \sigma}^{(t)} \right), \quad (27)$$

where $\mathcal{B}(\sigma)$ and $\mathcal{C}(\sigma)$ denote boundary and coboundary neighborhoods. In our setting, the "messages" are integer updates $\Delta\tilde{K}$ propagated on the inclusion lattice, with fixed nodes encoding physical constraints.

Implementation notes. (1) We implement Algorithm 2 using $\tilde{K}(A) = (|A|-2)!K(A)$, so the upward update along every face relation is integer and requires no division. In particular, for $B \subset C$ with $|C| = |B| + 1$ we use $\Delta\tilde{K}(C) \leftarrow \Delta\tilde{K}(B)$, which is equivalent to $\Delta K(C) \leftarrow \Delta K(B)/(|B|-1)$ in the original normalization. (2) Selection policy (deterministic). When multiple admissible subsets $s_D \subset s_B$ are available in the downward step, we choose the one that maximizes the remaining slack to inequality constraints; ties are broken lexicographically. This is for reproducibility and does not imply optimality.

5. Two phases

Operationally, the update alternates between: (i) upward propagation, which pushes $\Delta\tilde{K}$ to immediate supersets to maintain the recursion, and (ii) residual redistribution (backpropagation), which is triggered by fixed nodes and reroutes the update through unfixed alternatives. At convergence, the two-particle data $\{K(s_{ij})\}$ determine the integrand via

$$I(z) = \prod_{1 \leq i < j \leq n} (z_i - z_j)^{-K(s_{ij})}. \quad (28)$$

C. Pole selection algorithm

The computational framework developed in Sec. III B, combining combinatorial neural networks with discrete automatic differentiation, provides the essential infrastructure for implementing the pole selection algorithm. We now turn to the central computational challenge in the CHY formalism: isolating specific physical singularities from the complete amplitude structure.

The pick-pole algorithm addresses a basic practical task in applying the CHY formalism [26,27]. Given a CHY integrand $I(z)$ that produces an amplitude with multiple poles, we often need to isolate contributions containing a specific pole $1/s_A$ where $s_A = (\sum_{i \in A} k_i)^2$. This capability is useful for studying factorization limits and related analytic structures, and it also interfaces naturally with forward-limit constructions of loop-level objects [23].

For the biadjoint scalar theory, where the integrand takes the form $I[\pi|\rho] = \mathrm{PT}(\pi) \times \mathrm{PT}(\rho)$, the algorithm is elegant and provably correct [4,10]. The key insight is that a pole s_A appears if and only if the set A forms a contiguous block in both orderings π and ρ. One can then multiply by an appropriate cross-ratio factor to eliminate unwanted contributions:

$$\mathcal{P}_{bd}^{ac} = \frac{z_{ac} z_{bd}}{z_{ad} z_{cb}}, \quad (29)$$

Algorithm 2. PropagateUpdate: Discrete constraint propagation.

Require: Graph G, modified pole s_A, change $\Delta\tilde{K}$, fixed poles $\mathcal{F}$
Ensure: Updated $\tilde{K}$-values satisfying all constraints
Queue $Q \leftarrow \{s_A\}$, **Visited** $V \leftarrow \emptyset$
while $Q \neq \varnothing$ **do**
  $s_B \leftarrow \text{DEQUEUE}(Q)$
  if $s_B \in V$ **then**
    continue
  end if
  $V \leftarrow V \cup \{s_B\}$
  {Upward propagation to supersets (integer update in $\tilde{K}$)}
  for each superset $s_C \supset s_B$ with $|s_C| = |s_B| + 1$ **do**
    if $s_C \notin \mathcal{F}$ **then**
      $\Delta\tilde{K}_C \leftarrow \Delta\tilde{K}$
      $\tilde{K}^*(s_C) \leftarrow \tilde{K}^*(s_C) + \Delta\tilde{K}_C$
      $\text{ENQUEUE}(Q, s_C)$
    else
      {Fixed node encountered: trigger residual redistribution (backpropagation)}
      $\text{BACKPROPAGATE}(G, s_B, s_C, \Delta\tilde{K}, \mathcal{F})$
    end if
  end for
  {Downward step (only if needed when updating from higher layer)}
  if $|s_B| > 2$ **then**
    Select $s_D \subset s_B$ with $|s_D| = |s_B| - 1$ using a deterministic policy (see note)
    if $s_D \notin \mathcal{F}$ **then**
      $\tilde{K}^*(s_D) \leftarrow \tilde{K}^*(s_D) + \Delta\tilde{K}$
      $\text{ENQUEUE}(Q, s_D)$
    end if
  end if
end while

where a, b are the endpoints of the contiguous set A and c, d are their nearest neighbors in the complement $\bar{A}$ [50].

For more general CHY integrands, particularly those associated with labeled tree graphs [28], the notion of "contiguity" loses meaning, and the algorithm becomes noticeably more complex. As noted in [50], the current approach is heuristic, relying on a generate-and-test search through combinations of cross-ratio factors. This heuristic method, while sufficient to construct a broadly applicable one-loop CHY integrand for BAS theory [27], suffers from exponential computational complexity and lacks a rigorous proof of correctness.

Our approach reformulates this challenge within the graph-theoretic and topological framework developed in previous sections. The key insight is that pole selection corresponds to a systematic adjustment of K-values while maintaining all recursion relations and fixed-point constraints. Rather than searching through combinations of cross-ratio factors, we directly manipulate the underlying graph structure through controlled modifications of the generalized pole degrees. The complete pick-pole workflow is summarized in Algorithm 3.

The algorithm's correctness relies on several key properties established in previous sections. The additivity of K-values (Theorem in Sec. II B) ensures that modifications propagate correctly through the graph structure. The recursion relations (10) and (11) maintain consistency across different levels of the pole hierarchy. Most critically, the discrete automatic differentiation framework embodied in the PropagateUpdate subroutine (Algorithm 2) ensures that all constraints are satisfied throughout the modification process.

The connection to the topological perspective is particularly illuminating. Pole selection can be viewed as a projection operation on the simplicial complex, where we retain only those simplices (poles) compatible with our target. This geometric interpretation provides both intuition and a correctness criterion: the resulting integrand must correspond to a valid marked subset of the pole structure. The discrete nature of our approach—working with integer-valued integrand data and exact constraint satisfaction (integral in $\tilde{K}$)—distinguishes it from continuous optimization methods and is designed to yield a valid CHY integrand within this framework.

The following sections illustrate the algorithm through explicit examples, showing how it isolates desired pole structures in both six-point and eight-point amplitudes. These examples serve not only as validation of the method but also as pedagogical illustrations of how the discrete automatic differentiation operates on the hierarchical pole structure of CHY integrands.

Algorithm 3. Pick-pole algorithm for CHY integrands.

Require: CHY integrand $I(z)$ with pole structure, target pole s_T to retain
Ensure: Modified integrand $I^*(z)$ containing only terms with pole s_T
1: **Initialize:** Construct pole hierarchy graph G with K-values from $I(z)$
2: **Identify:** Mark poles to retain $\mathcal{P}_{\text{retain}} = \{s_T\} \cup \text{CompatiblePoles}(s_T)$
3: **Identify:** Mark poles to remove $\mathcal{P}_{\text{remove}} = \text{AllPoles}(I) \backslash \mathcal{P}_{\text{retain}}$
4: **for** each pole $s_A \in \mathcal{P}_{\text{remove}}$ **do**
5: $K^*(s_A) \leftarrow K(s_A) - 1$ {Decrement K-value}
6: **call** PROPAGATEUPDATE$(G, s_A, -1, \mathcal{P}_{\text{retain}})$
7: **end for**
8: **return** Integrand $I^*(z)$ constructed from modified K-values

D. Short example (6-pt)

We demonstrate the neural-inspired pick-pole algorithm through a concrete 6-point example. The initial hierarchy and the successive propagation/update steps are shown in Figs. 6–18. The algorithm operates on discrete structures where the bottom-layer $K(s_{ij})$ values are integers and propagation is implemented integrally in $\tilde{K}$; the pole hierarchy forms a two-layer network structure that enables systematic pole selection through discrete automatic differentiation.

For any subset $A \subset \{1, 2, \ldots, n\}$, we define the K-value as:

$$K(s_A) \coloneqq L[A].$$

In this example there are no dashed edges, so $K(s_A) = L[A]$.

Consider a six-point CHY integrand:

$$I_6 = \frac{1}{z_{12}^2 z_{23}^2 z_{34}^2 z_{56}^2 z_{16} z_{45} z_{46} z_{15}}$$

which produces five cubic Feynman diagrams:

$$\frac{1}{s_{12}s_{34}s_{56}} + \frac{1}{s_{123}s_{12}s_{56}} + \frac{1}{s_{123}s_{23}s_{56}} + \frac{1}{s_{156}s_{23}s_{56}} + \frac{1}{s_{156}s_{34}s_{56}}.$$

To pick pole s_{12}, we need to retain poles s_{12}, s_{34}, s_{56}, and s_{123} while removing poles s_{23} and s_{156}.

The recursion relations defining the pole hierarchy are

$$K(s_{abc}) = K(s_{ab}) + K(s_{ac}) + K(s_{bc})$$
$$deg(s_a) = K(s_{ab}) + K(s_{ac}) + K(s_{ad}) + K(s_{ae}) + K(s_{af}).$$

We fix the K-values of desired poles

$$K(s_{12}) = 2, \qquad K(s_{34}) = 2,$$
$$K(s_{56}) = 2, \qquad K(s_{123}) = 4.$$

Step 1: Remove pole s_{23}

We decrement the K-value of the unwanted pole s_{23}:

$$K^*(s_{23}) = K(s_{23}) - 1 = 1.$$

The upward propagation encounters fixed points s_{123}, s_2, and s_3, triggering backward propagation.

The algorithm compensates by selecting appropriate poles

$$K^*(s_{13}) = K(s_{13}) + 1 = 1,$$
$$K^*(s_{24}) = K(s_{24}) + 1 = 1.$$

The upward propagation again encounters fixed points s_1 and s_4.

The algorithm terminates when no fixed points are encountered, completing the removal of pole s_{23}.

Step 2: Remove pole s_{156}

We now repeat the procedure to remove pole s_{156}.

We decrement the K-value of pole s_{156}

$$K^*(s_{156}) = K(s_{156}) - 1 = 3.$$

The downward propagation selects subset s_{15} for modification.

The algorithm terminates successfully, yielding the modified CHY integrand

$$I_6^* = \frac{1}{z_{12}^2 z_{13} z_{16} z_{23} z_{25} z_{34}^2 z_{45} z_{46} z_{56}^2}.$$

This produces the amplitudes

$$\frac{1}{s_{12}s_{34}s_{56}} + \frac{1}{s_{123}s_{12}s_{56}}$$

which successfully picks up all terms containing the desired pole s_{12}.

1. Summary (6-point)

This example illustrates the pick-pole pipeline as deterministic constraint propagation on a fixed (two-layer) dependency graph. Two-particle channels form the base layer, while the single-particle and three-particle nodes form the next layer.

Starting from an integer modification of an unwanted pole (implemented integrally in $\tilde{K}$), the procedure: (i) propagates the induced update upward along face relations to maintain the recursion constraints; and (ii) upon encountering a fixed

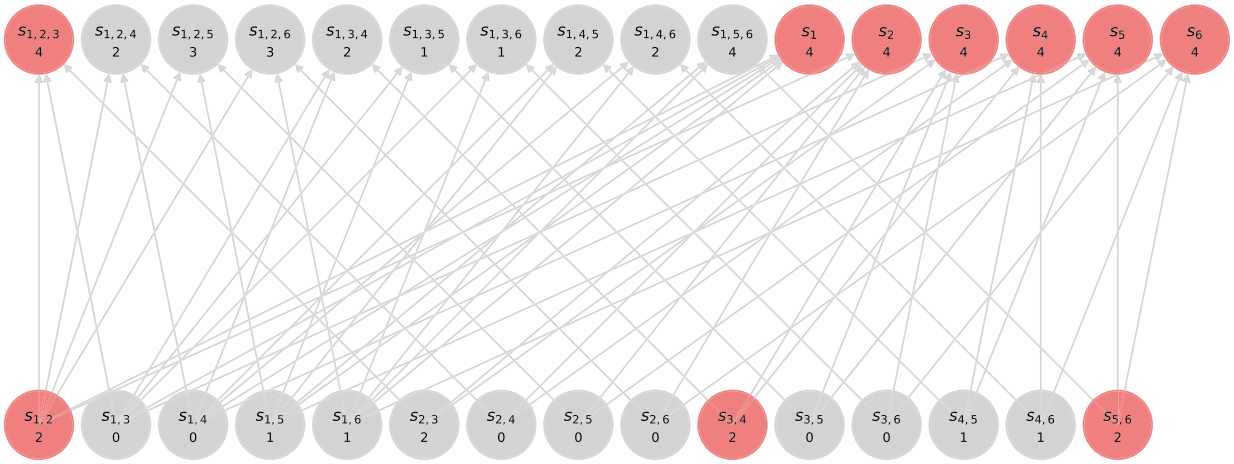


FIG. 6. Initial pole hierarchy for a 6-point CHY integrand. The two-layer network has two-particle poles in the initial layer (bottom) and single-particle and three-particle poles in the second layer (top). Gray edges represent dependency relationships from the recursion relations $K(s_{abc}) = K(s_{ab}) + K(s_{ac}) + K(s_{bc})$ and $deg(s_a) = K(s_{ab}) + K(s_{ac}) + K(s_{ad}) + K(s_{ae}) + K(s_{af})$. Red nodes indicate fixed poles that must be preserved during the algorithm.

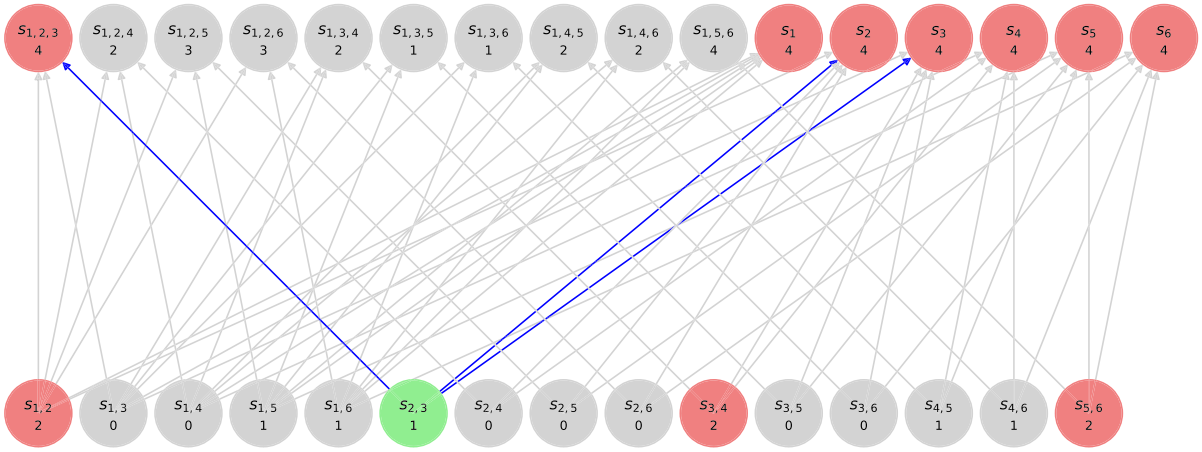


FIG. 7. Initial upward propagation: The modification $\Delta K = -1$ at pole s_{23} (green node) propagates upward through the network hierarchy. Blue arrows show the upward propagation paths to single-particle subsets and three-particle supersets. The algorithm encounters fixed constraints at s_{123}, s_2, and s_3 (red nodes), triggering the backward propagation mechanism.

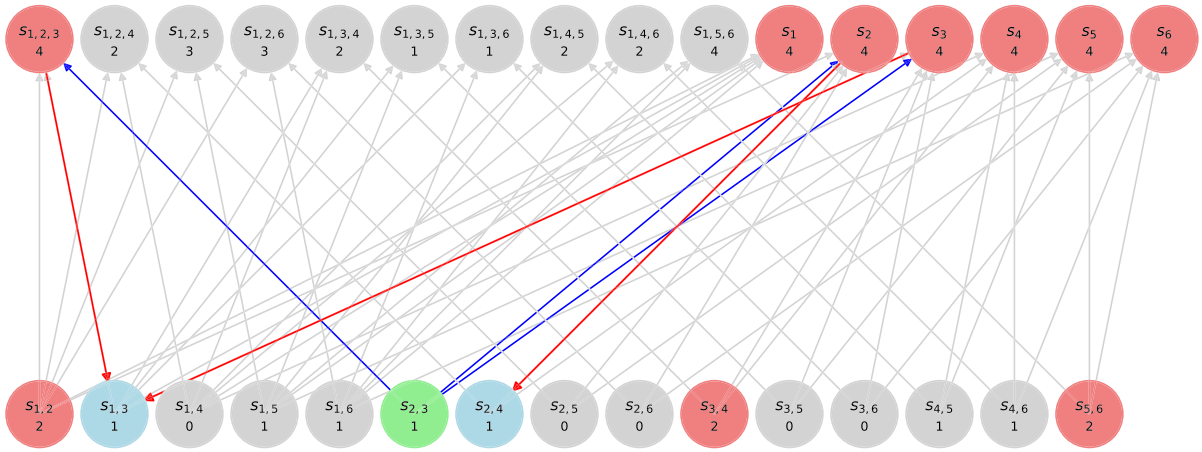


FIG. 8. Initial backward propagation: When upward propagation hits fixed constraints, the discrete automatic differentiation algorithm redistributes the changes through backward propagation (red arrows). The algorithm selects intersection s_{13} and superset s_{24}, incrementing their K-values by $+1$ to maintain all recursion relations while respecting fixed constraints.

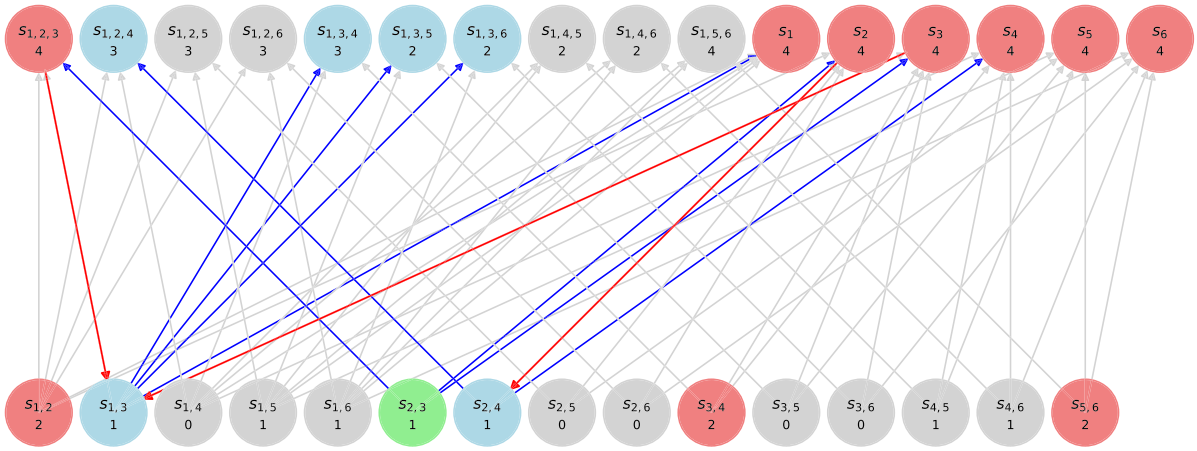


FIG. 9. Second upward propagation: The changes at s_{13} and s_{24} propagate upward (blue arrows), updating their three-particle supersets s_{134}, s_{135}, s_{136}, and s_{124} by $+1$. The propagation again encounters fixed points s_1 and s_4, necessitating another round of backward propagation.

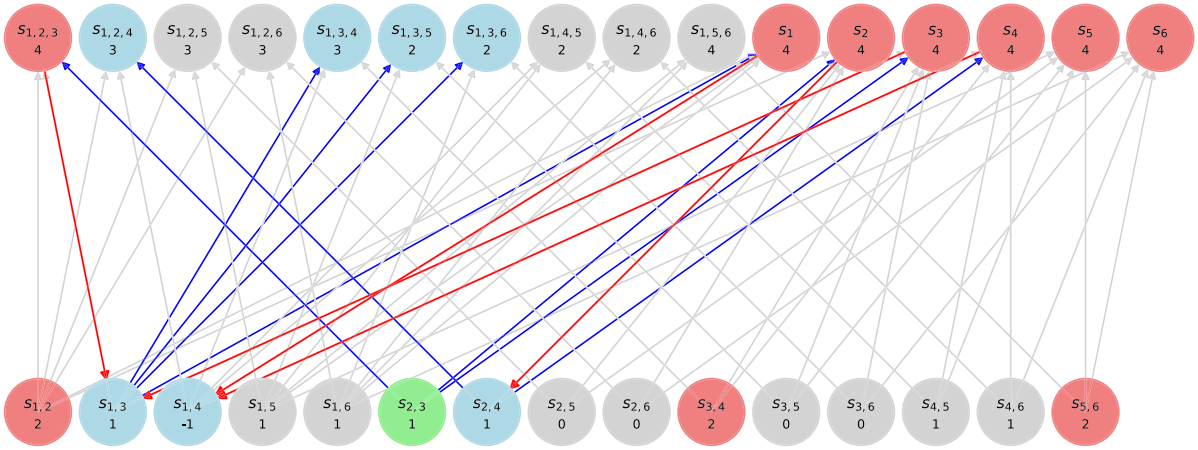


FIG. 10. Second backward propagation: To maintain consistency with fixed single-particle constraints, the algorithm performs backward propagation from the second layer to the initial layer (red arrows). The intersection s_{14} of dependent supersets from fixed nodes s_1 and s_4 is decremented by -1, ensuring all recursion relations remain satisfied.

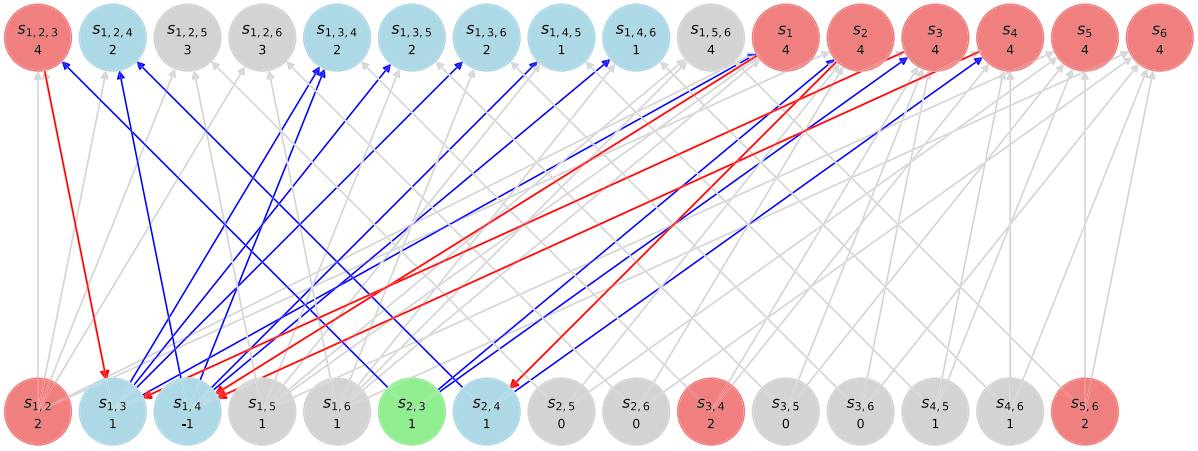


FIG. 11. Final upward propagation for s_{23} removal: The third upward propagation completes without encountering additional fixed points, indicating successful convergence of the discrete automatic differentiation algorithm. The removal of pole s_{23} is now complete with all constraints satisfied.

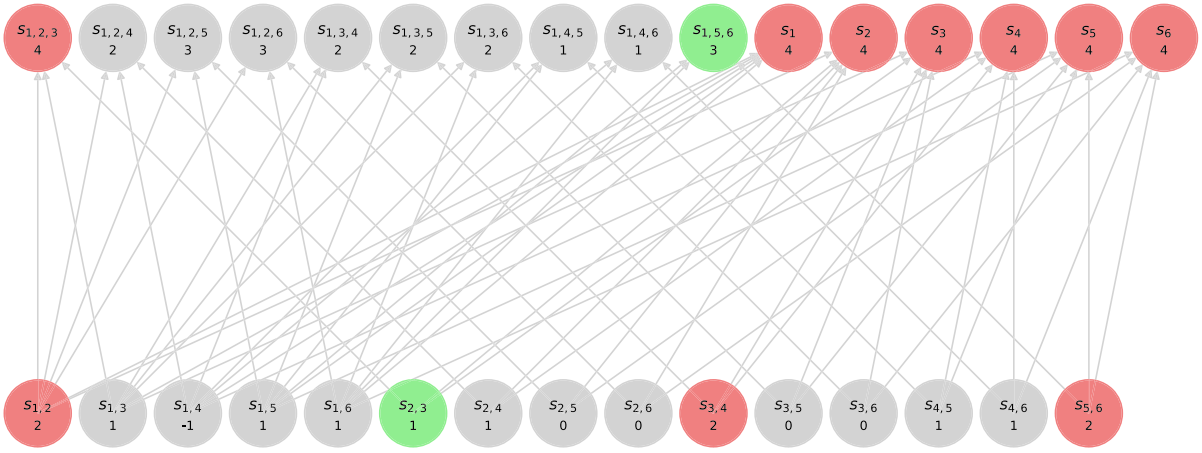


FIG. 12. Updated pole hierarchy for s_{156} removal: The network structure after removing s_{23}, showing updated K-values throughout the hierarchy. Fixed points (red) and the target pole s_{156} for removal (green) are clearly marked. The algorithm now proceeds to remove the second unwanted pole while maintaining all established constraints.

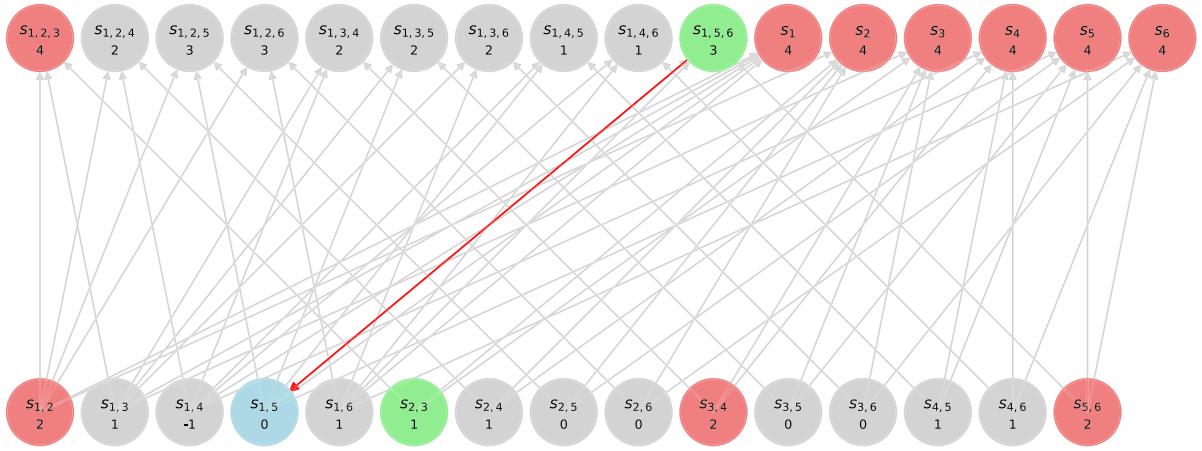


FIG. 13. Downward propagation: Unlike the previous case, the modification at three-particle pole s_{156} requires downward propagation to the initial layer (red arrows). The algorithm selects subset s_{15} and decrements its K-value by -1, demonstrating the bidirectional nature of the discrete automatic differentiation framework.

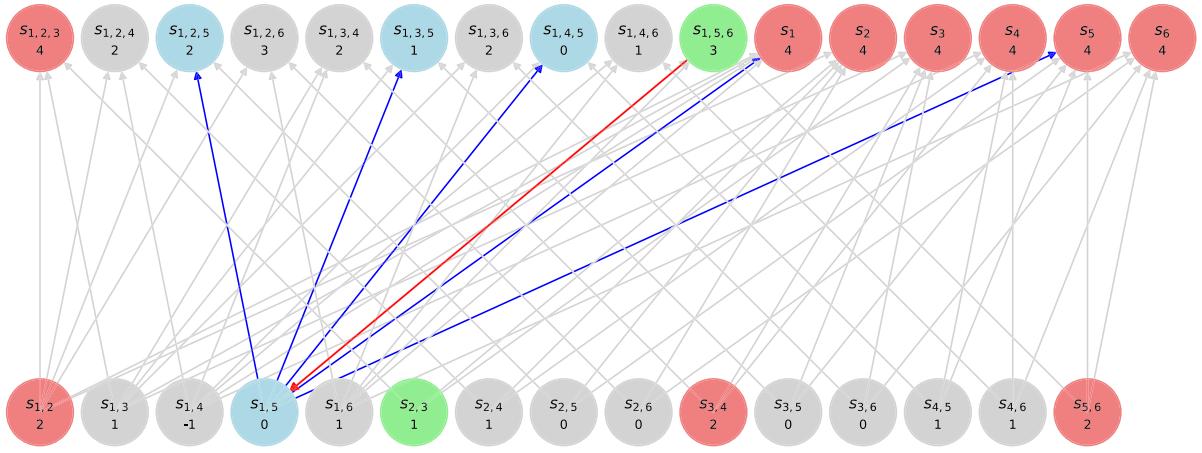


FIG. 14. Upward propagation from s_{15}: The change at s_{15} propagates upward (blue arrows) to its three-particle supersets s_{125}, s_{135}, and s_{145}, each decremented by -1. Fixed points s_1 and s_5 are encountered, triggering backward propagation to maintain network consistency.

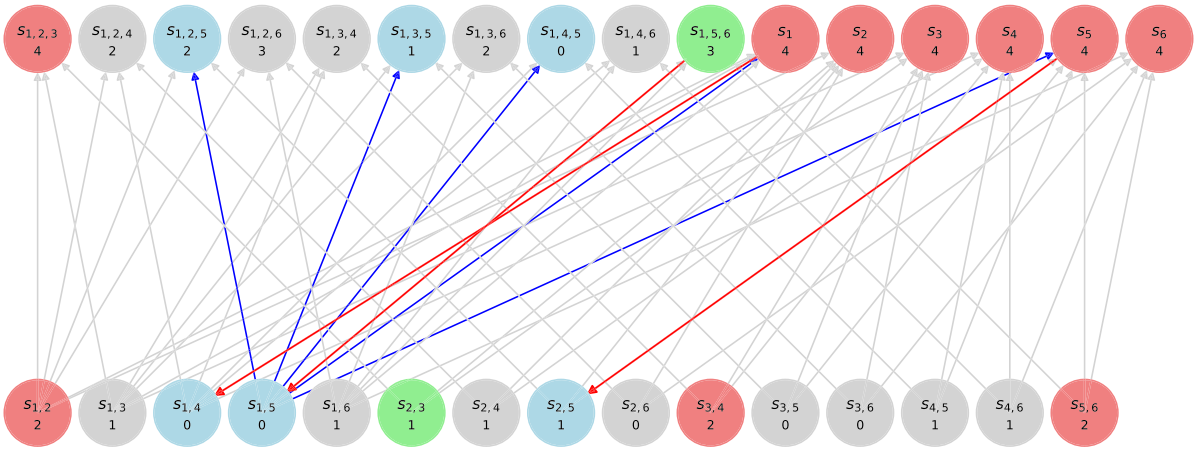


FIG. 15. Backward propagation for fixed constraints: The algorithm redistributes changes through backward propagation (red arrows), selecting supersets s_{14} and s_{25} of fixed points s_1 and s_5 respectively. Each is incremented by $+1$ to compensate for the inability to modify the fixed single-particle poles.

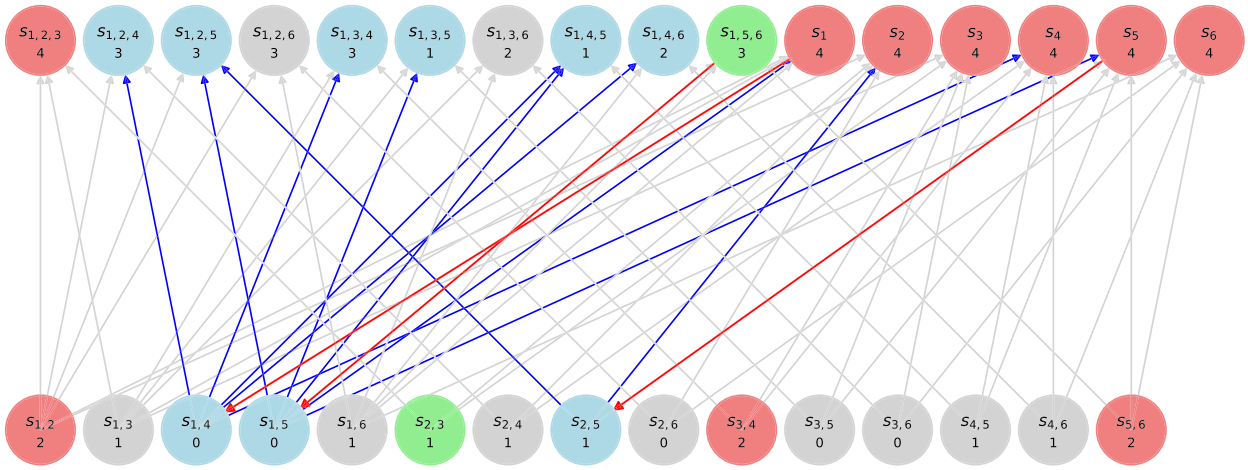


FIG. 16. Continued upward propagation: Changes at s_{14} and s_{25} propagate upward, updating their three-particle supersets by $+1$. The algorithm encounters additional fixed points s_2 and s_4, requiring another round of backward propagation to complete the constraint satisfaction process.

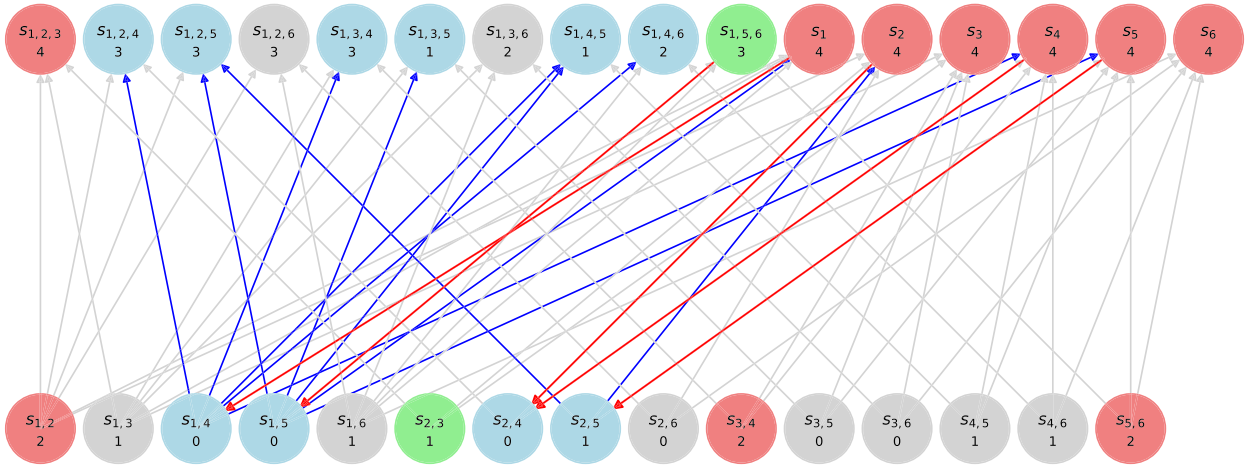


FIG. 17. Final backward propagation: The last backward propagation step decrements the intersection s_{24} of supersets from fixed nodes s_2 and s_4 by -1. This completes the constraint redistribution necessary to maintain all recursion relations while respecting fixed point constraints.

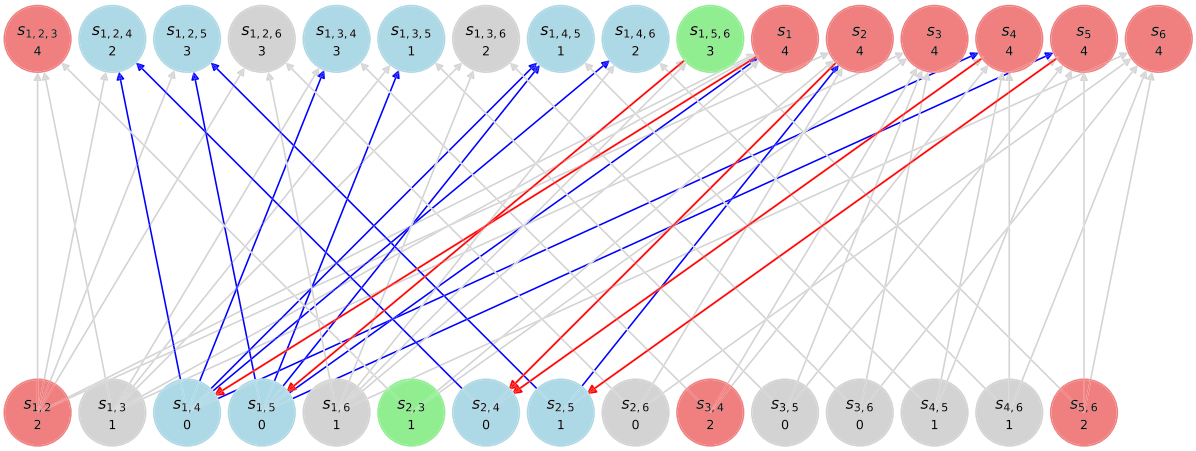


FIG. 18. Algorithm convergence: The final upward propagation from s_{24} completes without encountering fixed constraints, indicating successful convergence of the discrete automatic differentiation algorithm. The removal of both unwanted poles s_{23} and s_{156} is now complete, yielding the desired CHY integrand.

node, triggers *backpropagation*, i.e., discrete residual redistribution to nearby unfixed nodes so that all constraints remain satisfied. The same update rules extend to higher multiplicity, where additional layers (e.g., four-particle nodes at eight points) appear in the dependency graph.

IV. HIGHER-ORDER POLES AND 0-REGULAR CROSS-RATIO FACTORIZATION

A. Cross-ratio identities and higher-order poles

Physical tree-level amplitudes must contain only simple poles corresponding to the propagation of on-shell intermediate particles—a requirement dictated by unitarity and locality. Yet when we expand the Pfaffian factors in Yang-Mills or gravity integrands, individual terms routinely exhibit poles of the form $1/s_A^\alpha$ with $\alpha > 1$. These unphysical singularities must precisely cancel when summing over all $(n-3)!$ solutions to the scattering equations, but verifying such cancellations quickly becomes computationally expensive even for moderate multiplicities [7].

For a given n-point CHY integrand, the poles and the order of the poles can be determined by the pole index $\chi(A) := L[A] - 2(|A| - 1)$. Explicitly, each subset A gives a nonzero pole contribution if and only if $\chi(A) \geq 0$. From the previous algorithm for Feynman diagrams we can obtain all Feynman diagrams for the CHY integrand with only simple poles. But for higher-order poles, the problem becomes complicated.

For a CHY integrand with a pole of higher order $\frac{1}{s_A^\alpha}$, we can reduce the order of the CHY integrand by multiplying an identity containing that pole s_A. The construction of the identity for general higher-order poles is similar to the cross-ratio. In [7], the cross-ratio identity for pole s_A with selected index $j \in A$ and $p \in \bar{A}$ is

$$I_n[A, j, p] \equiv -\sum_{i \in A\setminus\{j\}} \sum_{b \in \bar{A}\setminus\{p\}} \frac{s_{ib}}{s_A} \frac{z_{bp} z_{ij}}{z_{ib} z_{jp}} = 1. \quad (30)$$

This identity holds only when the world sheet coordinates $\{z_i\}$ satisfy the scattering equations, making it an on-shell relation that can be freely inserted into the CHY integral. The appearance of the factor s_{ib}/s_A is crucial—when multiplied with an integrand containing a higher-order pole $1/s_A^\alpha$, it effectively reduces the pole order by one. The cross-ratio structure encoded in the factors $z_{bp} z_{ij}/(z_{ib} z_{jp})$ ensures that the resulting terms maintain the correct transformation properties under Möbius transformations.

Multiplying by this cross-ratio identity reduces the pole order and produces a linear combination of CHY integrands with simpler poles. The systematic algorithm iteratively reduces the higher-order poles until every resulting integrand is composed of simple poles. The algorithm proceeds as follows: for each subset A with $\chi(A) > 0$, we apply the appropriate cross-ratio identity, generating a linear combination of new integrands. This process continues until all resulting integrands satisfy $\chi(B) \leq 0$ for all subsets B, ensuring only physical simple poles remain.

However, during the iteration process we have no control over whether the decomposition increases or decreases the higher-order poles of a CHY integrand, which can lead to redundant terms. Our goal is to minimize the number of CHY integrands that contain only simple poles. In the following, we propose a decomposition strategy based on the systematic algorithm that reduces redundant terms. In this section, we explore how the methods developed in previous sections can be applied to this problem.

B. 0-regular factors as Möbius-invariant insertions

As established in Sec. II B, the 0-regular graphs form a group under multiplication and act as transformations on CHY integrands while preserving their regularity class. A crucial insight for decomposing higher-order poles is that multiplying a CHY integrand by an appropriate 0-regular graph can systematically reduce the pole orders. Specifically,

when a CHY integrand contains a higher-order pole $1/s_A^\alpha$ with $\alpha > 1$, we can construct a suitable 0-regular graph whose multiplication reduces this pole to simple poles, thereby enabling the systematic decomposition described in Sec. IV A.

The 0-regular graphs can be understood as multiplicative, Möbius-invariant factors: by definition they have vanishing net degree at each vertex and therefore preserve the regularity class when multiplied into a CHY integrand. For higher-order pole reduction, it is useful to view such factors as "long cross-ratios" associated with alternating even cycles in the colored graph.

A key practical point is that any 0-regular factor can be decomposed into a product of standard four-point cross-ratios. One constructive route is: (i) pair incident solid and dashed edges locally at each vertex [possible because $\deg_s(v) = \deg_d(v)$]; (ii) follow these pairings to partition the edge set into edge-disjoint alternating even cycles; and (iii) factorize each alternating $2m$-cycle into $(m-1)$ four-point cross-ratios via a telescoping identity. A proof and an explicit factorization formula are given in Appendix C. A graphical example of this decomposition is shown in Fig. 19.

C. Alternating-cycle → cross-ratio decomposition

For an alternating six-cycle with vertices (i, j, k, l, m, n) ordered as

$$i - j - k - l - m - n - i,$$

the associated long factor takes the form

$$\frac{z_{ij}z_{kl}z_{mn}}{z_{jk}z_{lm}z_{ni}}, \tag{31}$$

and admits the identity (up to an overall sign from $z_{ab} = -z_{ba}$)

$$\frac{z_{ij}z_{kl}z_{mn}}{z_{jk}z_{lm}z_{ni}} = \frac{z_{ij}z_{kn}}{z_{jk}z_{ni}} \times \frac{z_{kl}z_{mn}}{z_{lm}z_{kn}}. \tag{32}$$

Each factor on the right-hand side is a standard four-point cross-ratio. More generally, an alternating $2m$-cycle factorizes into $(m-1)$ four-point cross-ratios by iterating this step.

1. Enhanced decomposition method for higher-order pole integrands

With the systematic decomposition of 0-regular graphs established, we can now develop a more efficient method for decomposing higher-order pole integrands that reduces computational complexity in our examples compared to the systematic approach outlined in Ref. [7].

Enhanced algorithm:

(1) *0-regular graph construction*: For a given higher-order pole integrand, apply the methods from Sec. II B to construct an appropriate 0-regular graph that reduces the higher-order poles when multiplied with the original integrand.

(2) *Cross-ratio decomposition*: Apply the algorithm described above to decompose this 0-regular graph into products of standard four-point cross-ratios, systematically identifying all minimal cross-ratio structures.

(3) *Cross-ratio identity construction*: For each four-point cross-ratio identified in the decomposition, construct the corresponding cross-ratio identity using the methods from Sec. IV A. This yields a set of multiplicative factors that can be applied to the original integrand.

The key advantage of this enhanced approach is computational efficiency. Rather than applying multiple cross-ratio identities simultaneously (which leads to exponential growth in the number of terms), we apply them sequentially based on the systematic decomposition of the 0-regular graph.

2. Computational advantages

Our preliminary analysis of two representative examples demonstrates that this enhanced decomposition method

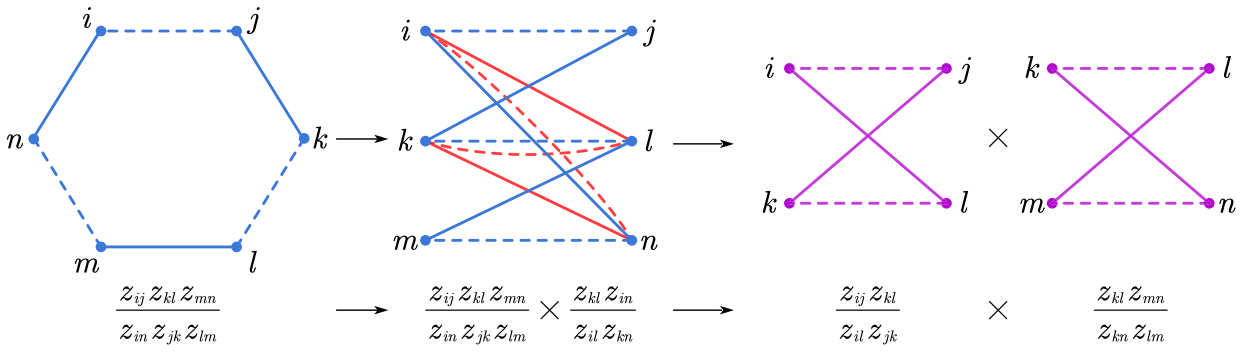


FIG. 19. Example of decomposing a 0-regular factor via alternating-cycle extraction and telescoping factorization. A convenient visualization is obtained by a left/right representation of vertices; the resulting alternating even cycle can then be factorized into four-point cross-ratios by inserting intermediate diagonals (cf. Appendix C).

produces noticeably fewer intermediate terms compared to the symmetric approach mentioned in Sec. IV A. Specifically:

(i) The traditional symmetric approach applies all necessary cross-ratio identities simultaneously, leading to $\prod_i n_i$ terms where n_i is the number of terms in each identity.
(ii) Our enhanced method applies identities sequentially based on the 0-regular graph decomposition, typically reducing the total number of terms by factors of 2-5 depending on the complexity of the pole structure.

In the following sections, we will demonstrate this enhanced method through detailed examples of 6-point and 8-point CHY integrands with higher-order poles, showing explicit comparisons between the number of terms generated by each approach. These examples illustrate both the theoretical elegance and practical computational advantages of the graph-theoretic approach to pole decomposition in the CHY formalism.

D. Efficiency comparison

Our enhanced decomposition method demonstrates clear computational advantages over traditional approaches. For this six-point example with three double poles (s_{12}, s_{34}, s_{56}):

Method	Total terms	Decomposition steps
Systematic algorithm [7]	27	4
Our enhanced method	*17*	*2*

This improvement becomes even more pronounced for higher-multiplicity amplitudes. A detailed eight-point example illustrating the scalability of our approach is presented in Appendix E, where the reduction from 265 to 153 terms illustrates the method's behavior for more complex cases.

Through this three-round process, the eight-point CHY integrand I_8 with four double poles is systematically decomposed into $9 + 60 + 12 + 72 = 153$ terms, each containing only simple poles. The improvement over traditional methods is summarized in the following table:

Method	Round 1	Round 2	Round 3	Round 4	Total
Systematic algorithm [7]	5 [5]	25 [20]	105 [20]	265 [0]	265
Our method	25 [16]	89 [8]	153 [0]		*153*

Numbers in brackets [·] indicate terms with remaining higher-order poles at each stage.

The systematic nature of our decomposition, guided by the graph-theoretic structure of 0-regular graphs, suggests that this computational advantage can persist as we move toward even higher multiplicities relevant for precision calculations in quantum field theory. Our enhanced method achieves this reduction by:

(1) Systematically decomposing cross-ratio factors using the 0-regular graph structure.
(2) Applying identities sequentially to minimize intermediate term proliferation.
(3) Leveraging the hierarchical nature of pole constraints to optimize the decomposition order.

This eight-point example suggests that our graph-theoretic approach to higher-order pole decomposition can maintain computational advantages as complexity increases, making it a useful tool for the amplitude calculations considered here.

E. Short example (6-pt)

We demonstrate our enhanced decomposition method through a concrete six-point example that illustrates the computational advantages over traditional approaches. Consider the CHY integrand

$$I_6 = \frac{1}{z_{12}^3 z_{23} z_{34}^3 z_{45} z_{56}^3 z_{61}}, \tag{33}$$

whose 4-regular graph representation is shown in Fig. 20.

Using the pole index formula $\chi(A) = L[A] - 2(|A| - 1)$, we identify that this integrand contains exactly three double poles: s_{12}, s_{34}, and s_{56} (where $\chi = 1$ for each). To systematically reduce these higher-order poles to simple poles, we employ the cross-ratio identity method with the six-point cross-ratio factor $\frac{z_{12}z_{34}z_{56}}{z_{16}z_{23}z_{45}}$.

The key to our enhanced approach is the systematic decomposition of this six-point cross-ratio into elementary four-point factors, as illustrated in Fig. 21.

Based on this decomposition, we strategically select two basic cross-ratio identities that minimize the generation of new higher-order poles

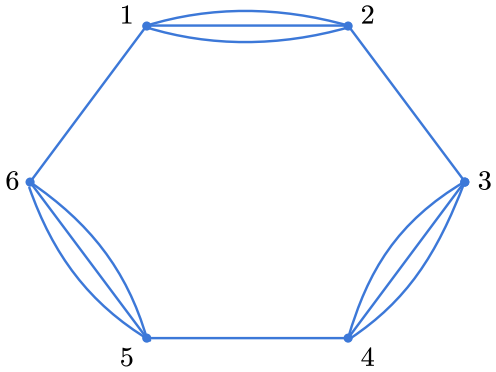


FIG. 20. The 4-regular graph representation of the six-point CHY integrand I_6. Each vertex represents a particle, and the edge multiplicities encode the powers in the denominator of equation (33). The triple edges connecting vertices (1, 2), (3, 4), and (5, 6) correspond to the factors z_{12}^3, z_{34}^3, and z_{56}^3 respectively, indicating the three double poles in the amplitude.

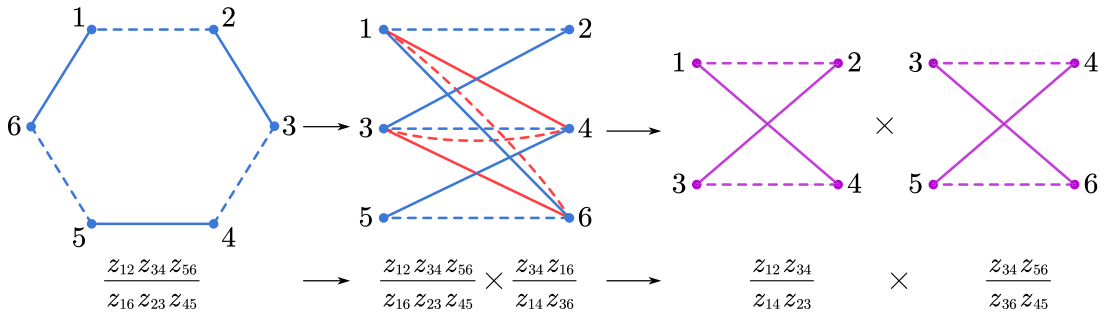


FIG. 21. Systematic decomposition of the six-point cross-ratio factor $\frac{z_{12}z_{34}z_{56}}{z_{16}z_{23}z_{45}}$ into two four-point cross-ratios. The bipartite representation (center) reveals the natural factorization into $\frac{z_{12}z_{34}}{z_{14}z_{32}} \times \frac{z_{34}z_{56}}{z_{36}z_{54}}$, providing the foundation for our sequential application of cross-ratio identities.

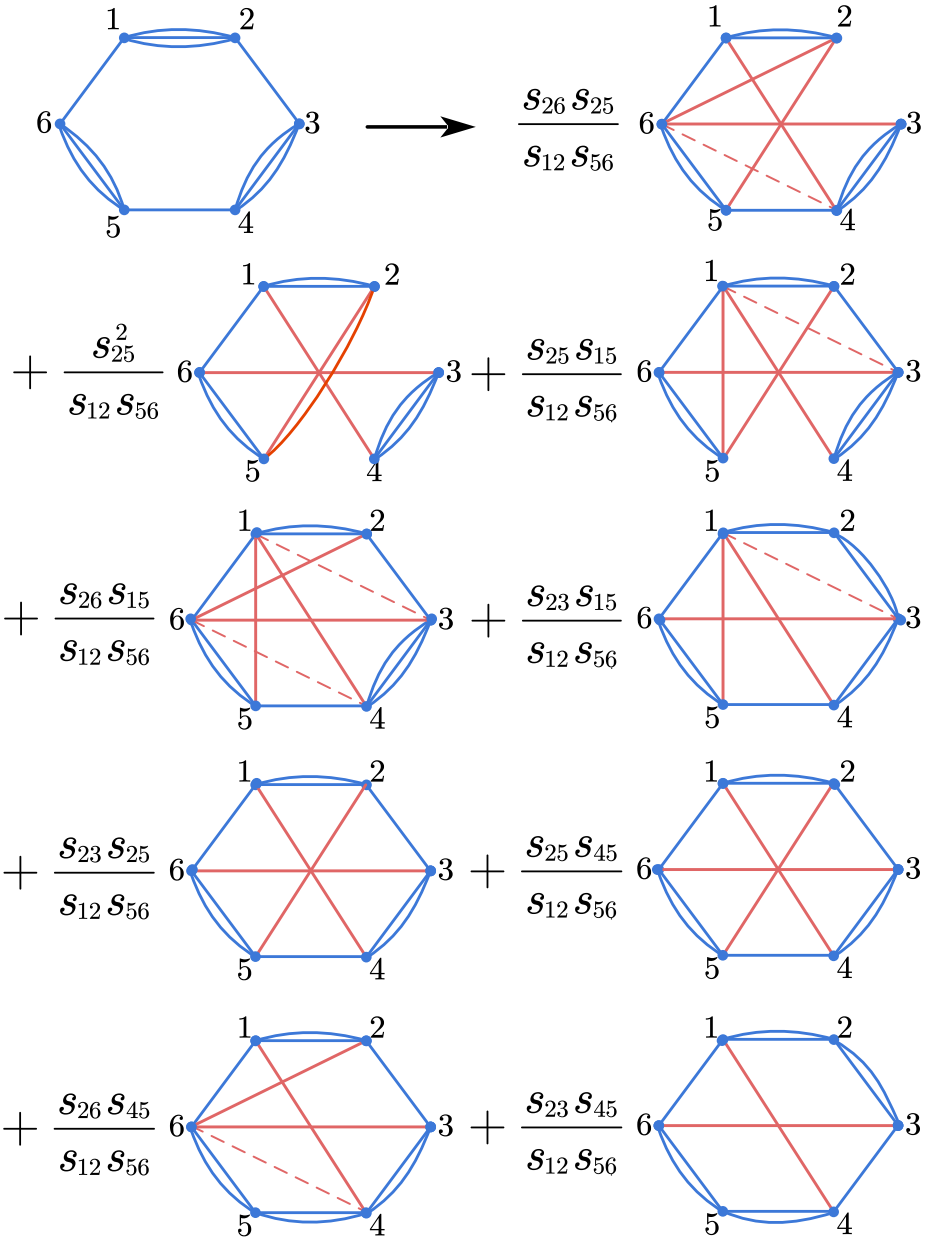


FIG. 22. Complete decomposition tree showing how the original integrand I_6 with three double poles is systematically reduced through cross-ratio identities. The initial level shows the $3 \times 3 = 9$ terms from applying identities (34) and (35). Five terms (shown in gray) already contain only simple poles, while four terms (shown in white) retain the double pole s_{34}, requiring further decomposition.

$$I_6[\{1,2\},1,4] = -\frac{s_{23}z_{21}z_{34}}{s_{12}z_{14}z_{23}} - \frac{s_{25}z_{21}z_{54}}{s_{12}z_{14}z_{25}} - \frac{s_{26}z_{21}z_{64}}{s_{12}z_{14}z_{26}}, \tag{34}$$

$$I_6[\{5,6\},6,3] = -\frac{s_{51}z_{13}z_{56}}{s_{56}z_{51}z_{63}} - \frac{s_{52}z_{23}z_{56}}{s_{56}z_{52}z_{63}} - \frac{s_{54}z_{43}z_{56}}{s_{56}z_{54}z_{63}}. \tag{35}$$

Multiplying I_6 by these identities yields $3^2 = 9$ terms, whose pole structures are systematically tracked in Fig. 22.

Among these nine terms, five immediately yield CHY integrands with only simple poles, while the remaining four terms

$$I_6 \times \frac{s_{26}s_{52}}{s_{12}s_{56}} \frac{z_{12}z_{23}z_{46}z_{56}}{z_{14}z_{25}z_{26}z_{36}},$$
$$I_6 \times \frac{s_{25}s_{52}}{s_{12}s_{56}} \frac{z_{12}z_{23}z_{45}z_{56}}{z_{14}z_{25}^2z_{36}}, \quad \text{etc.} \tag{36}$$

still contain the double pole s_{34}.

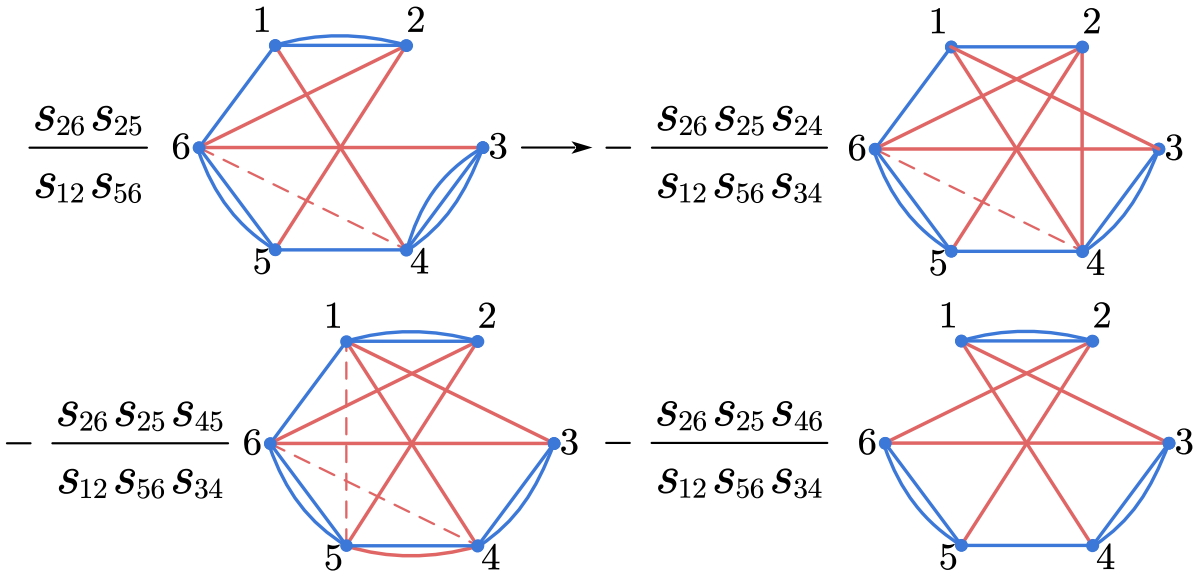


FIG. 23. Decomposition of the initial intermediate term containing double pole s_{34}. Applying identity (37) yields three CHY integrands with only simple poles, each corresponding to a distinct channel in the final amplitude. The cross-ratio factors effectively cancel the z_{34}^3 contribution, reducing the pole order from 2 to 1.

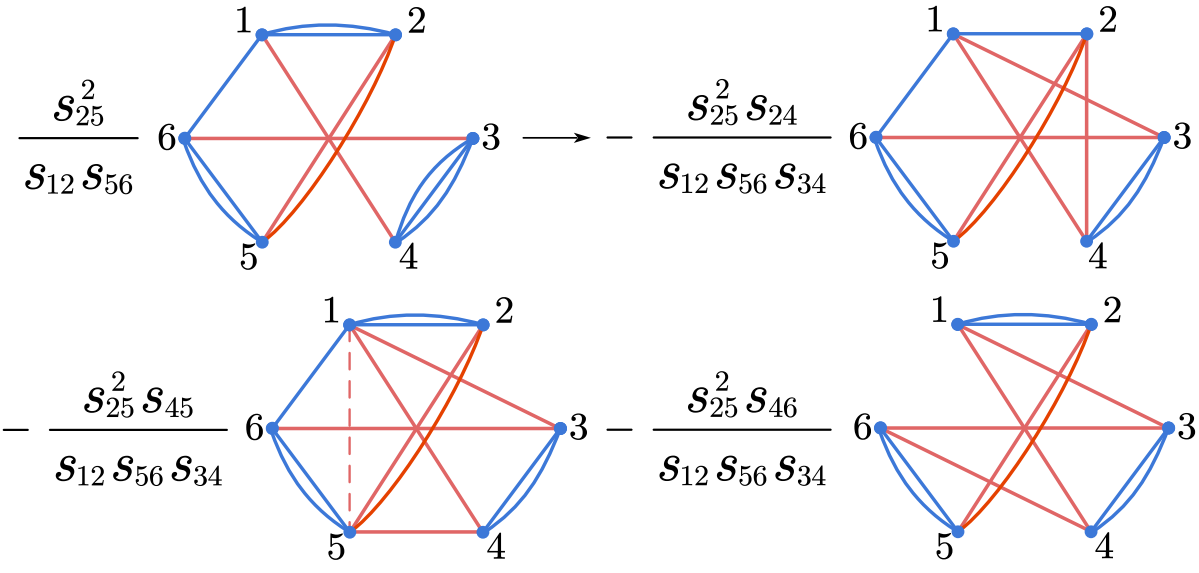


FIG. 24. Decomposition of the second intermediate term. The systematic application of identity (37) again produces three terms with simple poles, demonstrating the consistency of our approach across different intermediate configurations.

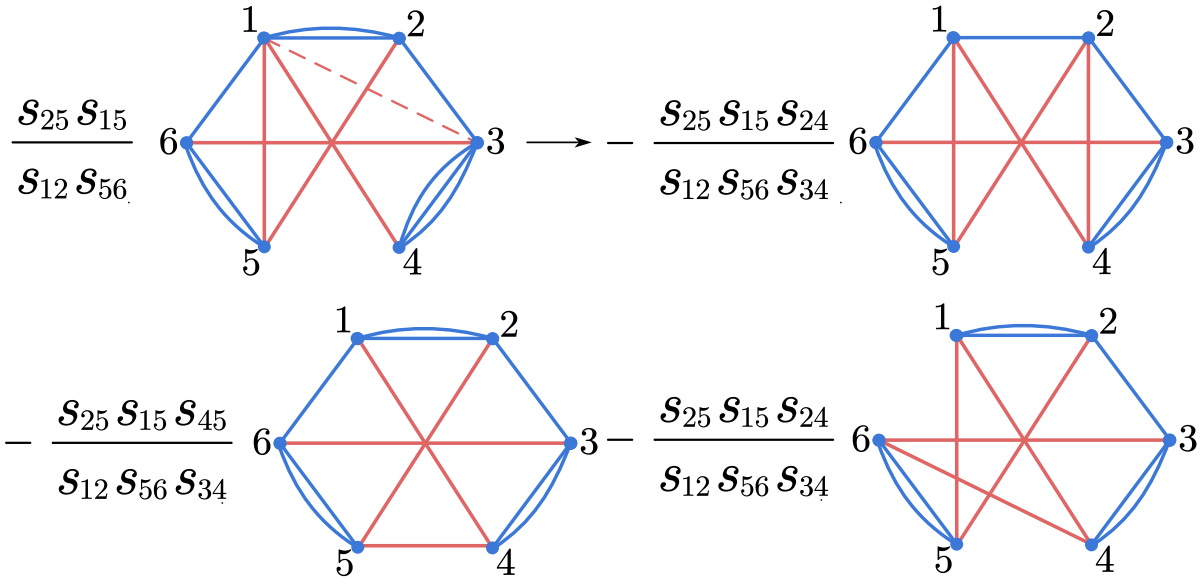


FIG. 25. Decomposition of the third intermediate term. Note how the cross-ratio identity targets the s_{34} pole while preserving the simple pole structure in other channels.

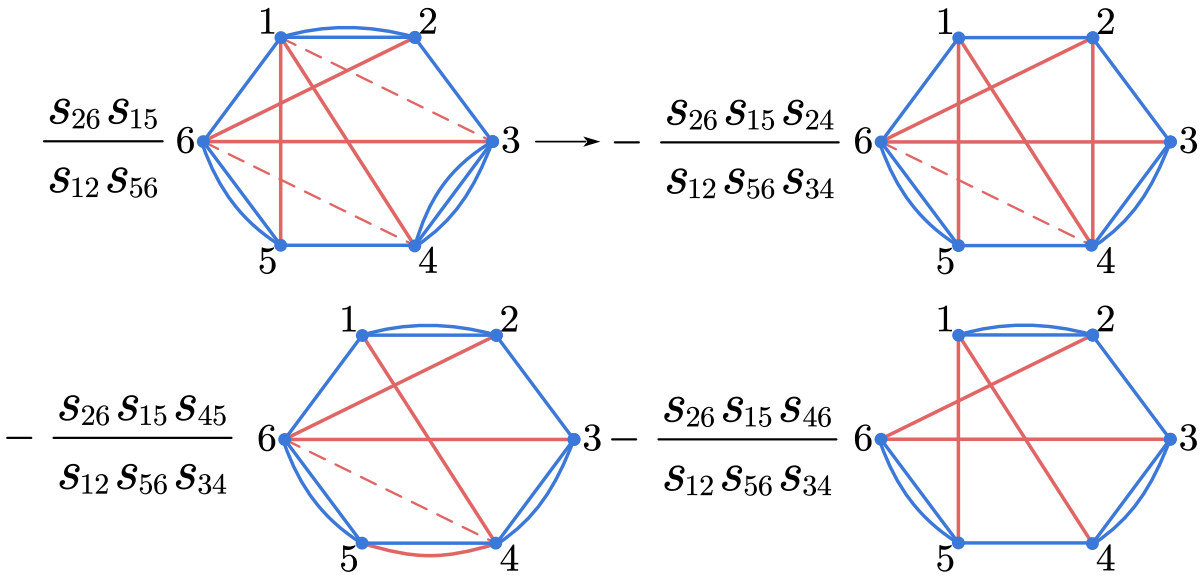


FIG. 26. Decomposition of the fourth and final intermediate term. The complete decomposition process transforms one integrand with three double poles into 17 integrands with only simple poles.

To eliminate this remaining higher-order pole, we apply the cross-ratio identity for subset $\{3,4\}$

$$I_6[\{3,4\},3,1] = -\frac{s_{42}z_{21}z_{43}}{s_{34}z_{31}z_{42}} - \frac{s_{45}z_{51}z_{43}}{s_{34}z_{31}z_{45}} - \frac{s_{46}z_{61}z_{43}}{s_{34}z_{31}z_{46}}. \tag{37}$$

Figures 23–26 illustrate how each of the four intermediate terms is further decomposed into three terms with only simple poles.

Through this systematic process, the original CHY integrand I_6 with three double poles is decomposed into $5+4\times 3=17$ terms, each containing only simple poles. The sum of these 17 terms reproduces the correct amplitude with all physical and spurious pole cancellations properly accounted for.

V. CONCLUSIONS

We presented an algorithmic approach to the inverse CHY construction problem: given a set of pole constraints,

we construct rational CHY integrands whose colored-graph data realize them [1–3]. The key object is the generalized pole degree $K(A)$, which can be read as a signed internal-edge count in the standard graph representation of CHY integrands [4,5,7]. In addition, we formulated the construction in a generalized "regularity" (net-degree) grading of colored graphs, where multiplication is graded and the degree-zero sector (0-regular graphs) forms a commutative group acting on a fixed regularity class (Sec. II B).

Our workflow has three components.

(i) *Linearization and MILP*. Using additivity under integrand multiplication and the elementary face recursion on the subset lattice, we express higher-channel $K(A)$ as linear functions of the two-particle variables $\{K(ij)\}$. The inverse step can then be posed as a mixed-integer *linear* feasibility problem for $\{K(ij)\}$ [11,12].

(ii) *Deterministic simplicial message passing*. The same recursion induces a fixed dependency graph (the Hasse diagram of the subset lattice). On this graph we implement deterministic constraint propagation in the factorial-rescaled variables $\tilde{K}(A) = (|A|-2)!K(A)$, so each local update is integral. In CoNN language, the forward pass evaluates the recursion upward, while a backward sweep propagates constraint residuals along the same fixed graph; this is a purely discrete procedure with no training and no learnable parameters [45–47].

(iii) *Applications and 0-regular factors*. We demonstrated pick-pole selection and higher-order pole reduction on six- and eight-point examples. In the graded regular-graph viewpoint, the reduction step can be implemented by multiplying by suitable 0-regular factors (degree-zero) and then decomposing them into products of standard four-point cross ratios.

This leads to a sequential use of cross-ratio identities which, in the examples studied, can reduce intermediate term proliferation compared to symmetric applications [7–9,26,27].

Because all steps operate directly on integer edge data and preserve the recursion constraints, intermediate quantities remain exact in the integer-rescaled recursion variables. This avoids workflows based on finite-field sampling with reconstruction or high-precision numerics with subsequent exactification [13–18]. Equivalently, the overall computation can be viewed as a constraint-preserving algorithm on a fixed dependency graph.

Natural extensions include improving selection/branching heuristics for the MILP stage, studying scaling to higher multiplicity, and exploring broader CHY integrand ansätze where the same subset-lattice constraints apply. Finally, the graded multiplication viewpoint is compatible with moduli-space formulations of CHY-type integrals and their combinatorial/topological organization [33,51]. Technical proofs, solver details, the constructive factorization argument, an additional 8-pt pole-selection example, and an additional 8-pt higher-order pole-reduction example are collected in Appendixes A–E.

ACKNOWLEDGMENTS

We acknowledge startup support from Ningxia University. The manuscript benefited from automated language-editing tools.

DATA AVAILABILITY

The data are not publicly available. The data are available from the authors upon reasonable request.

APPENDIX A: PROOFS OF SUBSET-LATTICE RECURSION RELATIONS

1. Proof of the elementary face recursion

Fix A with $|A| \geq 3$. Write $K(A)$ as a signed count of internal edges, $K(A) = \sum_{e\subset A} \sigma_e$, where $\sigma_e = +1$ for a solid edge and $\sigma_e = -1$ for a dashed edge. For any $(|A|-1)$-subset $B \subset A$, the same edge $e = (i, j)$ contributes to $K(B)$ iff the removed vertex is not i or j. There are exactly $|A|-2$ such choices. Hence each edge is counted $|A|-2$ times in $\sum_{|B|=|A|-1} K(B)$, so

$$\sum_{\substack{B\subset A \\ |B|=|A|-1}} K(B) = (|A|-2)K(A),$$

which implies (10).

2. Proof of the codimension-d recursion

Fix d with $1 \leq d \leq |A|-2$. For an internal edge $e = (i, j) \subset A$, the number of subsets $B \subset A$ of size $|A|-d$ that contain both i and j equals $\binom{|A|-2}{|A|-d-2} = \binom{|A|-2}{d}$. Therefore each internal edge contributes the same multiplicity $\binom{|A|-2}{d}$ to $\sum_{|B|=|A|-d} K(B)$, and summing with signs gives

$$\sum_{\substack{B\subset A \\ |B|=|A|-d}} K(B) = \binom{|A|-2}{d} K(A),$$

which implies (11).

APPENDIX B: FULL MILP/ILP FORMULATION AND SOLVER DETAILS

The inverse CHY problem—constructing integrands from desired amplitude structures—presents unique mathematical challenges. Unlike the forward problem of evaluating known integrands, the inverse problem requires navigating a vast discrete space of possible graph configurations while satisfying intricate algebraic constraints. The combinatorial nature of this problem, combined with the

Algorithm 4. Enhanced branch-and-cut algorithm.

```
Solve LP relaxation using simplex or interior point methods
if solution is integer then
  return solution
else
  Apply cutting plane generation (Gomory cuts, lift-and-project)
  Select branching variable using strong branching or pseudocost
  Create child nodes with disjoint constraints
  Apply preprocessing: bound tightening, constraint propagation
  Recursively solve subproblems with pruning based on bounds
  Use heuristics (feasibility pump, RINS) to find integer solutions
end if
```

requirement for exact integer solutions, places it firmly in the realm of NP-hard optimization problems [52,53].

This appendix describes how modern computational optimization techniques, specifically integer linear programming (ILP), can be leveraged to solve the bootstrap equations systematically. The approach transforms the abstract mathematical problem into a concrete computational framework amenable to well-established algorithmic solutions.

Mathematical formulation. The bootstrap construction requires finding integer values $\{K(s_{ij})\}$ that satisfy a system of linear constraints derived from physical requirements:

(i) *Equality constraints*: For desired poles, $K(s_A) = 2(|A|-1)$.
(ii) *Inequality constraints*: For unwanted poles, $K(s_A) < 2(|A|-1)$.
(iii) *Regularity constraints*: For 4-regular graphs, $K_i = \sum_{j\neq i} K(s_{ij}) = 4$.

This naturally maps to an integer linear programming problem:

(i) *Variables*: $\vec{x} = (K(s_{12}), K(s_{13}), \ldots, K(s_{n-1,n}))^T \in \mathbb{Z}_2^n$.
(ii) *Constraints*: $\mathbf{A}_{eq}\vec{x} = \vec{b}_{eq}$, $\mathbf{A}_{ineq}\vec{x} < \vec{b}_{ineq}$.
(iii) *Feasibility*: Find any $\vec{x}$ satisfying all constraints (no objective function).

The key insight enabling this formulation is that all higher-order pole degrees $K(s_A)$ with $|A| > 2$ can be expressed as linear combinations of the two-particle variables $\{K(s_{ij})\}$ via the elementary face recursion (10). This linearization is crucial—without it, the constraints would involve nonlinear relationships that would make the problem much harder to solve.

Computational complexity. The difficulty of this problem is substantial. For an n-particle amplitude:

(i) The number of variables scales as $\binom{n}{2} = O(n^2)$.
(ii) The number of potential constraints scales as $2^n - 1$ (one for each subset).
(iii) The search space for integer solutions grows exponentially.

Even for moderate values of n, brute-force enumeration becomes impossible. For $n = 10$, there are already 45 variables and over 1000 potential constraints. The integer requirement transforms what would be a polynomial-time solvable linear program into an NP-hard problem [11,12].

Algorithmic approaches. Modern ILP solvers employ sophisticated techniques to handle this complexity. The standard framework is branch-and-bound, enhanced with cutting plane methods [54,55]: A schematic branch-and-cut workflow is summarized in Algorithm 4.

Recent advances in ILP solvers have improved performance through parallelization, machine learning-guided branching strategies, and symmetry detection [56,57].

Implementation details. For practical implementation, we utilize:

In *Mathematica* [58]:

(i) `k[i,j]`: Integer variable representation.
(ii) `KValue[set]`: Recursive computation using memoization.
(iii) `FindInstance`: Leverages internal ILP solver with symmetry detection.

In Python [59]:

(i) `scipy.optimize.milp`: Interfaces with HiGHS solver [60].
(ii) `cvxpy`: Provides modeling language with automatic reformulation [61].
(iii) `PuLP`: Connects to commercial solvers (Gurobi, CPLEX) [62].

The 8-particle example. We demonstrate the method on an 8-particle amplitude requiring poles at $s_{15}, s_{16}, s_{23}, s_{68}, s_{78}, s_{126}, s_{156}, s_{234}, s_{378}, s_{678}, s_{1256}$, and s_{3478}. This example, while manageable computationally, exhibits the full complexity of the problem:

(i) 28 integer variables $\{K(s_{ij})\}_{1\leq i<j\leq 8}$ includes 5 equalities

$$K_{1,5} = 2, \qquad K_{1,6} = 2, \qquad K_{2,3} = 2, \qquad K_{6,8} = 2, \qquad K_{7,8} = 2,$$

and 23 inequalities

$$K_{1,2} < 2, \quad K_{1,3} < 2, \quad K_{1,4} < 2, \quad K_{1,7} < 2, \quad K_{1,8} < 2, \quad K_{2,4} < 2, \quad K_{2,5} < 2, \quad K_{2,6} < 2,$$
$$K_{2,7} < 2, \quad K_{2,8} < 2, \quad K_{3,4} < 2, \quad K_{3,5} < 2, \quad K_{3,6} < 2, \quad K_{3,7} < 2, \quad K_{3,8} < 2, \quad K_{4,5} < 2,$$
$$K_{4,6} < 2, \quad K_{4,7} < 2, \quad K_{4,8} < 2, \quad K_{5,6} < 2, \quad K_{5,7} < 2, \quad K_{5,8} < 2, \quad K_{6,7} < 2.$$

(ii) 162 total constraints (20 equalities, 142 inequalities). In addition to the above 28 integer variables, it also includes 15 equalities:

$$K_{1,2} + K_{1,6} + K_{2,6} = 4, \quad K_{1,5} + K_{1,6} + K_{5,6} = 4, \quad K_{2,3} + K_{2,4} + K_{3,4} = 4,$$
$$K_{3,7} + K_{3,8} + K_{7,8} = 4, \quad K_{6,7} + K_{6,8} + K_{7,8} = 4,$$
$$K_{1,2} + K_{1,5} + K_{1,6} + K_{2,5} + K_{2,6} + K_{5,6} = 6,$$
$$K_{3,4} + K_{3,7} + K_{3,8} + K_{4,7} + K_{4,8} + K_{7,8} = 6,$$
$$K_1 = K_{1,2} + K_{1,3} + K_{1,4} + K_{1,5} + K_{1,6} + K_{1,7} + K_{1,8} = 4,$$
$$K_2 = K_{1,2} + K_{2,3} + K_{2,4} + K_{2,5} + K_{2,6} + K_{2,7} + K_{2,8} = 4,$$
$$K_3 = K_{1,3} + K_{2,3} + K_{3,4} + K_{3,5} + K_{3,6} + K_{3,7} + K_{3,8} = 4,$$
$$K_4 = K_{1,4} + K_{2,4} + K_{3,4} + K_{4,5} + K_{4,6} + K_{4,7} + K_{4,8} = 4,$$
$$K_5 = K_{1,5} + K_{2,5} + K_{3,5} + K_{4,5} + K_{5,6} + K_{5,7} + K_{5,8} = 4,$$
$$K_6 = K_{1,6} + K_{2,6} + K_{3,6} + K_{4,6} + K_{5,6} + K_{6,7} + K_{6,8} = 4,$$
$$K_7 = K_{1,7} + K_{2,7} + K_{3,7} + K_{4,7} + K_{5,7} + K_{6,7} + K_{7,8} = 4,$$
$$K_8 = K_{1,8} + K_{2,8} + K_{3,8} + K_{4,8} + K_{5,8} + K_{6,8} + K_{7,8} = 4.$$

and 51 inequalities, of three poles

$$K_{1,2} + K_{1,3} + K_{2,3} < 4, \quad K_{1,2} + K_{1,4} + K_{2,4} < 4, \quad K_{1,2} + K_{1,5} + K_{2,5} < 4,$$
$$K_{1,2} + K_{1,7} + K_{2,7} < 4, \quad K_{1,2} + K_{1,8} + K_{2,8} < 4, \quad K_{1,3} + K_{1,4} + K_{3,4} < 4,$$
$$K_{1,3} + K_{1,5} + K_{3,5} < 4, \quad K_{1,3} + K_{1,6} + K_{3,6} < 4, \quad K_{1,3} + K_{1,7} + K_{3,7} < 4,$$
$$K_{1,3} + K_{1,8} + K_{3,8} < 4, \quad K_{1,4} + K_{1,5} + K_{4,5} < 4, \quad K_{1,4} + K_{1,6} + K_{4,6} < 4,$$
$$K_{1,4} + K_{1,7} + K_{4,7} < 4, \quad K_{1,4} + K_{1,8} + K_{4,8} < 4, \quad K_{1,5} + K_{1,7} + K_{5,7} < 4,$$
$$K_{1,5} + K_{1,8} + K_{5,8} < 4, \quad K_{1,6} + K_{1,7} + K_{6,7} < 4, \quad K_{1,6} + K_{1,8} + K_{6,8} < 4,$$
$$K_{1,7} + K_{1,8} + K_{7,8} < 4, \quad K_{2,3} + K_{2,5} + K_{3,5} < 4, \quad K_{2,3} + K_{2,6} + K_{3,6} < 4,$$
$$K_{2,3} + K_{2,7} + K_{3,7} < 4, \quad K_{2,3} + K_{2,8} + K_{3,8} < 4, \quad K_{2,4} + K_{2,5} + K_{4,5} < 4,$$
$$K_{2,4} + K_{2,6} + K_{4,6} < 4, \quad K_{2,4} + K_{2,7} + K_{4,7} < 4, \quad K_{2,4} + K_{2,8} + K_{4,8} < 4,$$
$$K_{2,5} + K_{2,6} + K_{5,6} < 4, \quad K_{2,5} + K_{2,7} + K_{5,7} < 4, \quad K_{2,5} + K_{2,8} + K_{5,8} < 4,$$
$$K_{2,6} + K_{2,7} + K_{6,7} < 4, \quad K_{2,6} + K_{2,8} + K_{6,8} < 4, \quad K_{2,7} + K_{2,8} + K_{7,8} < 4,$$
$$K_{3,4} + K_{3,5} + K_{4,5} < 4, \quad K_{3,4} + K_{3,6} + K_{4,6} < 4, \quad K_{3,4} + K_{3,7} + K_{4,7} < 4,$$
$$K_{3,4} + K_{3,8} + K_{4,8} < 4, \quad K_{3,5} + K_{3,6} + K_{5,6} < 4, \quad K_{3,5} + K_{3,7} + K_{5,7} < 4,$$
$$K_{3,5} + K_{3,8} + K_{5,8} < 4, \quad K_{3,6} + K_{3,7} + K_{6,7} < 4, \quad K_{3,6} + K_{3,8} + K_{6,8} < 4,$$
$$K_{4,5} + K_{4,6} + K_{5,6} < 4, \quad K_{4,5} + K_{4,7} + K_{5,7} < 4, \quad K_{4,5} + K_{4,8} + K_{5,8} < 4,$$
$$K_{4,6} + K_{4,7} + K_{6,7} < 4, \quad K_{4,6} + K_{4,8} + K_{6,8} < 4, \quad K_{4,7} + K_{4,8} + K_{7,8} < 4,$$
$$K_{5,6} + K_{5,7} + K_{6,7} < 4, \quad K_{5,6} + K_{5,8} + K_{6,8} < 4, \quad K_{5,7} + K_{5,8} + K_{7,8} < 4,$$

and 68 inequalities, of four poles

$$
\begin{aligned}
&K_{1,2}+K_{1,3}+K_{1,4}+K_{2,3}+K_{2,4}+K_{3,4}<6, && K_{1,2}+K_{1,3}+K_{1,5}+K_{2,3}+K_{2,5}+K_{3,5}<6,\\
&K_{1,2}+K_{1,3}+K_{1,6}+K_{2,3}+K_{2,6}+K_{3,6}<6, && K_{1,2}+K_{1,3}+K_{1,7}+K_{2,3}+K_{2,7}+K_{3,7}<6,\\
&K_{1,2}+K_{1,3}+K_{1,8}+K_{2,3}+K_{2,8}+K_{3,8}<6, && K_{1,2}+K_{1,4}+K_{1,5}+K_{2,4}+K_{2,5}+K_{4,5}<6,\\
&K_{1,2}+K_{1,4}+K_{1,6}+K_{2,4}+K_{2,6}+K_{4,6}<6, && K_{1,2}+K_{1,4}+K_{1,7}+K_{2,4}+K_{2,7}+K_{4,7}<6,\\
&K_{1,2}+K_{1,4}+K_{1,8}+K_{2,4}+K_{2,8}+K_{4,8}<6, && K_{1,2}+K_{1,5}+K_{1,7}+K_{2,5}+K_{2,7}+K_{5,7}<6,\\
&K_{1,2}+K_{1,5}+K_{1,8}+K_{2,5}+K_{2,8}+K_{5,8}<6, && K_{1,2}+K_{1,6}+K_{1,7}+K_{2,6}+K_{2,7}+K_{6,7}<6,\\
&K_{1,2}+K_{1,6}+K_{1,8}+K_{2,6}+K_{2,8}+K_{6,8}<6, && K_{1,2}+K_{1,7}+K_{1,8}+K_{2,7}+K_{2,8}+K_{7,8}<6,\\
&K_{1,3}+K_{1,4}+K_{1,5}+K_{3,4}+K_{3,5}+K_{4,5}<6, && K_{1,3}+K_{1,4}+K_{1,6}+K_{3,4}+K_{3,6}+K_{4,6}<6,\\
&K_{1,3}+K_{1,4}+K_{1,7}+K_{3,4}+K_{3,7}+K_{4,7}<6, && K_{1,3}+K_{1,4}+K_{1,8}+K_{3,4}+K_{3,8}+K_{4,8}<6,\\
&K_{1,3}+K_{1,5}+K_{1,6}+K_{3,5}+K_{3,6}+K_{5,6}<6, && K_{1,3}+K_{1,5}+K_{1,7}+K_{3,5}+K_{3,7}+K_{5,7}<6,\\
&K_{1,3}+K_{1,5}+K_{1,8}+K_{3,5}+K_{3,8}+K_{5,8}<6, && K_{1,3}+K_{1,6}+K_{1,7}+K_{3,6}+K_{3,7}+K_{6,7}<6,\\
&K_{1,3}+K_{1,6}+K_{1,8}+K_{3,6}+K_{3,8}+K_{6,8}<6, && K_{1,3}+K_{1,7}+K_{1,8}+K_{3,7}+K_{3,8}+K_{7,8}<6,\\
&K_{1,4}+K_{1,5}+K_{1,6}+K_{4,5}+K_{4,6}+K_{5,6}<6, && K_{1,4}+K_{1,5}+K_{1,7}+K_{4,5}+K_{4,7}+K_{5,7}<6,\\
&K_{1,4}+K_{1,5}+K_{1,8}+K_{4,5}+K_{4,8}+K_{5,8}<6, && K_{1,4}+K_{1,6}+K_{1,7}+K_{4,6}+K_{4,7}+K_{6,7}<6,\\
&K_{1,4}+K_{1,6}+K_{1,8}+K_{4,6}+K_{4,8}+K_{6,8}<6, && K_{1,4}+K_{1,7}+K_{1,8}+K_{4,7}+K_{4,8}+K_{7,8}<6,\\
&K_{1,5}+K_{1,6}+K_{1,7}+K_{5,6}+K_{5,7}+K_{6,7}<6, && K_{1,5}+K_{1,6}+K_{1,8}+K_{5,6}+K_{5,8}+K_{6,8}<6,\\
&K_{1,5}+K_{1,7}+K_{1,8}+K_{5,7}+K_{5,8}+K_{7,8}<6, && K_{1,6}+K_{1,7}+K_{1,8}+K_{6,7}+K_{6,8}+K_{7,8}<6,\\
&K_{2,3}+K_{2,4}+K_{2,5}+K_{3,4}+K_{3,5}+K_{4,5}<6, && K_{2,3}+K_{2,4}+K_{2,6}+K_{3,4}+K_{3,6}+K_{4,6}<6,\\
&K_{2,3}+K_{2,4}+K_{2,7}+K_{3,4}+K_{3,7}+K_{4,7}<6, && K_{2,3}+K_{2,4}+K_{2,8}+K_{3,4}+K_{3,8}+K_{4,8}<6,\\
&K_{2,3}+K_{2,5}+K_{2,6}+K_{3,5}+K_{3,6}+K_{5,6}<6, && K_{2,3}+K_{2,5}+K_{2,7}+K_{3,5}+K_{3,7}+K_{5,7}<6,\\
&K_{2,3}+K_{2,5}+K_{2,8}+K_{3,5}+K_{3,8}+K_{5,8}<6, && K_{2,3}+K_{2,6}+K_{2,7}+K_{3,6}+K_{3,7}+K_{6,7}<6,\\
&K_{2,3}+K_{2,6}+K_{2,8}+K_{3,6}+K_{3,8}+K_{6,8}<6, && K_{2,3}+K_{2,7}+K_{2,8}+K_{3,7}+K_{3,8}+K_{7,8}<6,\\
&K_{2,4}+K_{2,5}+K_{2,6}+K_{4,5}+K_{4,6}+K_{5,6}<6, && K_{2,4}+K_{2,5}+K_{2,7}+K_{4,5}+K_{4,7}+K_{5,7}<6,\\
&K_{2,4}+K_{2,5}+K_{2,8}+K_{4,5}+K_{4,8}+K_{5,8}<6, && K_{2,4}+K_{2,6}+K_{2,7}+K_{4,6}+K_{4,7}+K_{6,7}<6,\\
&K_{2,4}+K_{2,6}+K_{2,8}+K_{4,6}+K_{4,8}+K_{6,8}<6, && K_{2,4}+K_{2,7}+K_{2,8}+K_{4,7}+K_{4,8}+K_{7,8}<6,\\
&K_{2,5}+K_{2,6}+K_{2,7}+K_{5,6}+K_{5,7}+K_{6,7}<6, && K_{2,5}+K_{2,6}+K_{2,8}+K_{5,6}+K_{5,8}+K_{6,8}<6,\\
&K_{2,5}+K_{2,7}+K_{2,8}+K_{5,7}+K_{5,8}+K_{7,8}<6, && K_{2,6}+K_{2,7}+K_{2,8}+K_{6,7}+K_{6,8}+K_{7,8}<6,\\
&K_{3,4}+K_{3,5}+K_{3,6}+K_{4,5}+K_{4,6}+K_{5,6}<6, && K_{3,4}+K_{3,5}+K_{3,7}+K_{4,5}+K_{4,7}+K_{5,7}<6,\\
&K_{3,4}+K_{3,5}+K_{3,8}+K_{4,5}+K_{4,8}+K_{5,8}<6, && K_{3,4}+K_{3,6}+K_{3,7}+K_{4,6}+K_{4,7}+K_{6,7}<6,\\
&K_{3,4}+K_{3,6}+K_{3,8}+K_{4,6}+K_{4,8}+K_{6,8}<6, && K_{3,5}+K_{3,6}+K_{3,7}+K_{5,6}+K_{5,7}+K_{6,7}<6,\\
&K_{3,5}+K_{3,6}+K_{3,8}+K_{5,6}+K_{5,8}+K_{6,8}<6, && K_{3,5}+K_{3,7}+K_{3,8}+K_{5,7}+K_{5,8}+K_{7,8}<6,\\
&K_{3,6}+K_{3,7}+K_{3,8}+K_{6,7}+K_{6,8}+K_{7,8}<6, && K_{4,5}+K_{4,6}+K_{4,7}+K_{5,6}+K_{5,7}+K_{6,7}<6,\\
&K_{4,5}+K_{4,6}+K_{4,8}+K_{5,6}+K_{5,8}+K_{6,8}<6, && K_{4,5}+K_{4,7}+K_{4,8}+K_{5,7}+K_{5,8}+K_{7,8}<6,\\
&K_{4,6}+K_{4,7}+K_{4,8}+K_{6,7}+K_{6,8}+K_{7,8}<6, &&\\
&K_{5,6}+K_{5,7}+K_{5,8}+K_{6,7}+K_{6,8}+K_{7,8}<6. &&
\end{aligned}
$$

(iii) Search space of approximately 10^{20} integer vectors.

The constraint system admits exactly four distinct integer solutions
Solution 1

$$\begin{aligned}
&K_{1,2}\to 1, \quad K_{1,3}\to 0, \quad K_{1,4}\to 0, \quad K_{1,5}\to 2, \quad K_{1,6}\to 2, \quad K_{1,7}\to 0, \quad K_{1,8}\to -1,\\
&K_{2,3}\to 2, \quad K_{2,4}\to 1, \quad K_{2,5}\to 0, \quad K_{2,6}\to 1, \quad K_{2,7}\to -1, \quad K_{2,8}\to 0,\\
&K_{3,4}\to 1, \quad K_{3,5}\to 0, \quad K_{3,6}\to -1, \quad K_{3,7}\to 1, \quad K_{3,8}\to 1,\\
&K_{4,5}\to 1, \quad K_{4,6}\to 0, \quad K_{4,7}\to 1, \quad K_{4,8}\to 0,\\
&K_{5,6}\to 0, \quad K_{5,7}\to 1, \quad K_{5,8}\to 0, \quad K_{6,7}\to 0, \quad K_{6,8}\to 2, \quad K_{7,8}\to 2.
\end{aligned}$$

Solution 2

$$\begin{aligned}
&K_{1,2}\to 1, \quad K_{1,3}\to 0, \quad K_{1,4}\to 0, \quad K_{1,5}\to 2, \quad K_{1,6}\to 2, \quad K_{1,7}\to 0, \quad K_{1,8}\to -1,\\
&K_{2,3}\to 2, \quad K_{2,4}\to 1, \quad K_{2,5}\to 0, \quad K_{2,6}\to 1, \quad K_{2,7}\to 0, \quad K_{2,8}\to -1,\\
&K_{3,4}\to 1, \quad K_{3,5}\to 0, \quad K_{3,6}\to -1, \quad K_{3,7}\to 1, \quad K_{3,8}\to 1,\\
&K_{4,5}\to 1, \quad K_{4,6}\to 0, \quad K_{4,7}\to 0, \quad K_{4,8}\to 1,\\
&K_{5,6}\to 0, \quad K_{5,7}\to 1, \quad K_{5,8}\to 0, \quad K_{6,7}\to 0, \quad K_{6,8}\to 2, \quad K_{7,8}\to 2.
\end{aligned}$$

Solution 3

$$\begin{aligned}
&K_{1,2}\to 1, \quad K_{1,3}\to 0, \quad K_{1,4}\to 0, \quad K_{1,5}\to 2, \quad K_{1,6}\to 2, \quad K_{1,7}\to 1, \quad K_{1,8}\to -2,\\
&K_{2,3}\to 2, \quad K_{2,4}\to 1, \quad K_{2,5}\to 0, \quad K_{2,6}\to 1, \quad K_{2,7}\to -1, \quad K_{2,8}\to 0,\\
&K_{3,4}\to 1, \quad K_{3,5}\to 0, \quad K_{3,6}\to -1, \quad K_{3,7}\to 1, \quad K_{3,8}\to 1,\\
&K_{4,5}\to 1, \quad K_{4,6}\to 0, \quad K_{4,7}\to 1, \quad K_{4,8}\to 0,\\
&K_{5,6}\to 0, \quad K_{5,7}\to 0, \quad K_{5,8}\to 1, \quad K_{6,7}\to 0, \quad K_{6,8}\to 2, \quad K_{7,8}\to 2.
\end{aligned}$$

Solution 4

$$\begin{aligned}
&K_{1,2}\to 1, \quad K_{1,3}\to 0, \quad K_{1,4}\to 0, \quad K_{1,5}\to 2, \quad K_{1,6}\to 2, \quad K_{1,7}\to 1, \quad K_{1,8}\to -2,\\
&K_{2,3}\to 2, \quad K_{2,4}\to 1, \quad K_{2,5}\to 0, \quad K_{2,6}\to 1, \quad K_{2,7}\to 0, \quad K_{2,8}\to -1,\\
&K_{3,4}\to 1, \quad K_{3,5}\to 0, \quad K_{3,6}\to -1, \quad K_{3,7}\to 1, \quad K_{3,8}\to 1,\\
&K_{4,5}\to 1, \quad K_{4,6}\to 0, \quad K_{4,7}\to 0, \quad K_{4,8}\to 1,\\
&K_{5,6}\to 0, \quad K_{5,7}\to 0, \quad K_{5,8}\to 1, \quad K_{6,7}\to 0, \quad K_{6,8}\to 2, \quad K_{7,8}\to 2.
\end{aligned}$$

All solutions satisfy the regularity condition $K_i = 4$ for $i = 1, \ldots, 8$. Each solution yields a distinct CHY integrand via:

$$I_n = \prod_{1\le i<j\le n} (z_i - z_j)^{-K(s_{ij})}. \tag{B1}$$

For instance, solution 1 produces

$$I_8^{[1]} = \frac{z_{18} z_{27} z_{36}}{z_{12} z_{15}^2 z_{16}^2 z_{23}^2 z_{24} z_{26} z_{34} z_{37} z_{38} z_{45} z_{47} z_{57} z_{68}^2 z_{78}^2}. \tag{B2}$$

Solution 2 produces

$$I_8^{[2]} = \frac{z_{18} z_{28} z_{36}}{z_{12} z_{15}^2 z_{16}^2 z_{23}^2 z_{24} z_{26} z_{34} z_{37} z_{38} z_{45} z_{48} z_{57} z_{68}^2 z_{78}^2}. \tag{B3}$$

Solution 3 produces

$$I_8^{[3]} = \frac{z_{18}^2 z_{27} z_{36}}{z_{12} z_{15}^2 z_{16}^2 z_{17} z_{23}^2 z_{24} z_{26} z_{34} z_{37} z_{38} z_{45} z_{47} z_{58} z_{68}^2 z_{78}^2}. \tag{B4}$$

Solution 4 produces

$$I_8^{[4]} = \frac{z_{18}^2 z_{28} z_{36}}{z_{12} z_{15}^2 z_{16}^2 z_{17} z_{23}^2 z_{24} z_{26} z_{34} z_{37} z_{38} z_{45} z_{48} z_{58} z_{68}^2 z_{78}^2}. \tag{B5}$$

The existence of exactly four solutions is nontrivial. It demonstrates that the pole structure alone does not uniquely determine the CHY integrand, indicating a degeneracy in the map from amplitudes to world sheet representations. Related structures appear in [33,51,63]; we leave a systematic investigation of any connection to the space of consistent quantum field theories within the CHY framework for future work.

APPENDIX C: ALTERNATING-CYCLE DECOMPOSITION AND CROSS-RATIO FACTORIZATION

This appendix provides a constructive justification for the decomposition used in Sec. IV B: any 0-regular multiplicative factor can be written as a product of standard four-point cross-ratios. We only use elementary graph-theoretic notions [64,65].

We represent a multiplicative factor by a colored multigraph $G = (V, E_s \cup E_d)$, where each solid edge $(i, j) \in E_s$ corresponds to a denominator factor z_{ij}^{-1} and each dashed edge $(i, j) \in E_d$ corresponds to a numerator factor z_{ij}^{+1}. Up to an overall sign from $z_{ij} = -z_{ji}$, the associated monomial is

$$\mathcal{P}_G(z) := \frac{\prod_{(i,j)\in E_d} z_{ij}}{\prod_{(i,j)\in E_s} z_{ij}}. \tag{C1}$$

The graph is *0-regular* if for every vertex $v \in V$,

$$\deg_s(v) = \deg_d(v), \tag{C2}$$

i.e., the net degree $\deg_{\text{net}}(v) = \deg_s(v) - \deg_d(v)$ vanishes.

Lemma C.1. (Alternating cycle decomposition). Let $G = (V, E_s \cup E_d)$ be a 0-regular multigraph. Then $E_s \cup E_d$ can be partitioned into edge-disjoint closed alternating cycles (each cycle alternates between solid and dashed edges).

Proof. At each vertex v, choose an arbitrary bijection between the incident solid edges and the incident dashed edges. This is possible because $\deg_s(v) = \deg_d(v)$. Use these bijections to define a local "next-edge" rule: whenever a walk arrives at v along a solid edge, it exits along the paired dashed edge, and vice versa.

Start from any unused edge and iterate this rule. Because the number of edges is finite and the update is deterministic, the walk eventually repeats an edge and therefore closes into an alternating cycle. Removing the edges of this cycle from G preserves 0-regularity on the remaining multigraph (degrees of each color are reduced equally at each visited vertex). Repeating the procedure exhausts all edges and yields a partition into edge-disjoint alternating cycles. ∎

Each alternating cycle C contributes a factor $\mathcal{P}_C(z)$, and since the cycles are edge-disjoint we have

$$\mathcal{P}_G(z) = \prod_C \mathcal{P}_C(z). \tag{C3}$$

Lemma C.2. (Even-cycle telescoping into four-point cross-ratios). Let $C = (v_1, v_2, \ldots, v_{2m})$ be an alternating cycle with $2m \geq 4$ distinct vertices and $v_{2m+1} \equiv v_1$. Define its associated "long" factor by

$$\mathcal{P}_C(z) := \frac{\prod_{r=1}^m z_{v_{2r-1}v_{2r}}}{\prod_{r=1}^m z_{v_{2r}v_{2r+1}}}. \tag{C4}$$

Then (up to an overall sign) $\mathcal{P}_C(z)$ factorizes into $(m-1)$ four-point cross-ratios as

$$\mathcal{P}_C(z) = \prod_{r=2}^{m} \frac{z_{v_{2r-3}v_{2r-2}} z_{v_{2r-1}v_{2m}}}{z_{v_{2r-2}v_{2r-1}} z_{v_{2r-3}v_{2m}}}. \tag{C5}$$

Proof. Multiply the right-hand side of (C5). The factors involving $z_{(\cdot)v_{2m}}$ telescope: every intermediate $z_{v_{2r-1}v_{2m}}$ cancels between numerator and denominator, leaving only

Algorithm 5. Decomposition of a 0-regular factor into four-point cross-ratios.

Require: 0-regular multigraph $G = (V, E_s \cup E_d)$
Ensure: List $\mathcal{L}$ of four-point cross-ratios whose product equals $\mathcal{P}_G(z)$ (up to sign)
Choose, for each vertex v, a bijection pairing incident solid and dashed edges
Mark all edges unused; set $\mathcal{L} \leftarrow \emptyset$
while there exists an unused edge **do**
  Extract an alternating cycle $C = (v_1, \ldots, v_{2m})$ by following the local pairings
  for $r = 2$ **to** m **do**
    Append $\frac{z_{v_{2r-3}v_{2r-2}} z_{v_{2r-1}v_{2m}}}{z_{v_{2r-2}v_{2r-1}} z_{v_{2r-3}v_{2m}}}$ to $\mathcal{L}$
  end for
  Mark the edges of C used
end while
return $\mathcal{L}$

$z_{v_{2m-1}v_{2m}}$ in the numerator and $z_{v_1 v_{2m}}$ in the denominator. The remaining uncanceled factors reproduce exactly the product of consecutive edge factors in (C4). This proves the identity. ∎

Theorem C.3. (0-regular decomposition into four-point cross-ratios). For any 0-regular multigraph G, the associated factor $\mathcal{P}_G(z)$ can be written (up to an overall sign) as a product of standard four-point cross-ratios.

Proof. By Lemma C.1, $\mathcal{P}_G$ factorizes into a product over alternating cycles. Each cycle factor then decomposes into four-point cross-ratios by Lemma C.2. A constructive version of this decomposition is summarized in Algorithm 5. ∎

APPENDIX D: ADDITIONAL 8-PT POLE-SELECTION EXAMPLE (FIGURES)

To demonstrate the scalability and effectiveness of our discrete automatic differentiation framework, we present a comprehensive example using an 8-point CHY integrand. This example highlights the three-layer network structure and shows how our neural-inspired pick-pole algorithm isolates desired pole structures while maintaining constraint satisfaction.

Consider the 8-point CHY integrand

$$I_8 = \frac{z_{18}^2 z_{28} z_{36}}{z_{12} z_{15}^2 z_{16}^2 z_{17} z_{23}^2 z_{24} z_{26} z_{34} z_{37} z_{38} z_{45} z_{48} z_{58} z_{68}^2 z_{78}^2}, \tag{D1}$$

which generates seven distinct Feynman diagrams

$$\begin{aligned} &\frac{1}{s_{16}s_{23}s_{78}s_{156}s_{234}} + \frac{1}{s_{15}s_{23}s_{68}s_{234}s_{678}} + \frac{1}{s_{15}s_{23}s_{78}s_{234}s_{678}} \\ &+ \frac{1}{s_{16}s_{78}s_{126}s_{378}s_{1256}} + \frac{1}{s_{15}s_{78}s_{156}s_{378}s_{1256}} \\ &+ \frac{1}{s_{16}s_{78}s_{156}s_{378}s_{1256}} + \frac{1}{s_{15}s_{23}s_{78}s_{156}s_{234}}. \end{aligned} \tag{D2}$$

The amplitude contains simple poles at s_{15}, s_{16}, s_{23}, s_{68}, s_{78}, s_{126}, s_{156}, s_{234}, s_{378}, s_{678}, and s_{1256}. Our objective is to isolate all contributions containing the pole s_{15} by systematically removing all incompatible poles.

We construct a three-layer hierarchical network where the dependency relationships are governed by the recursion relation:

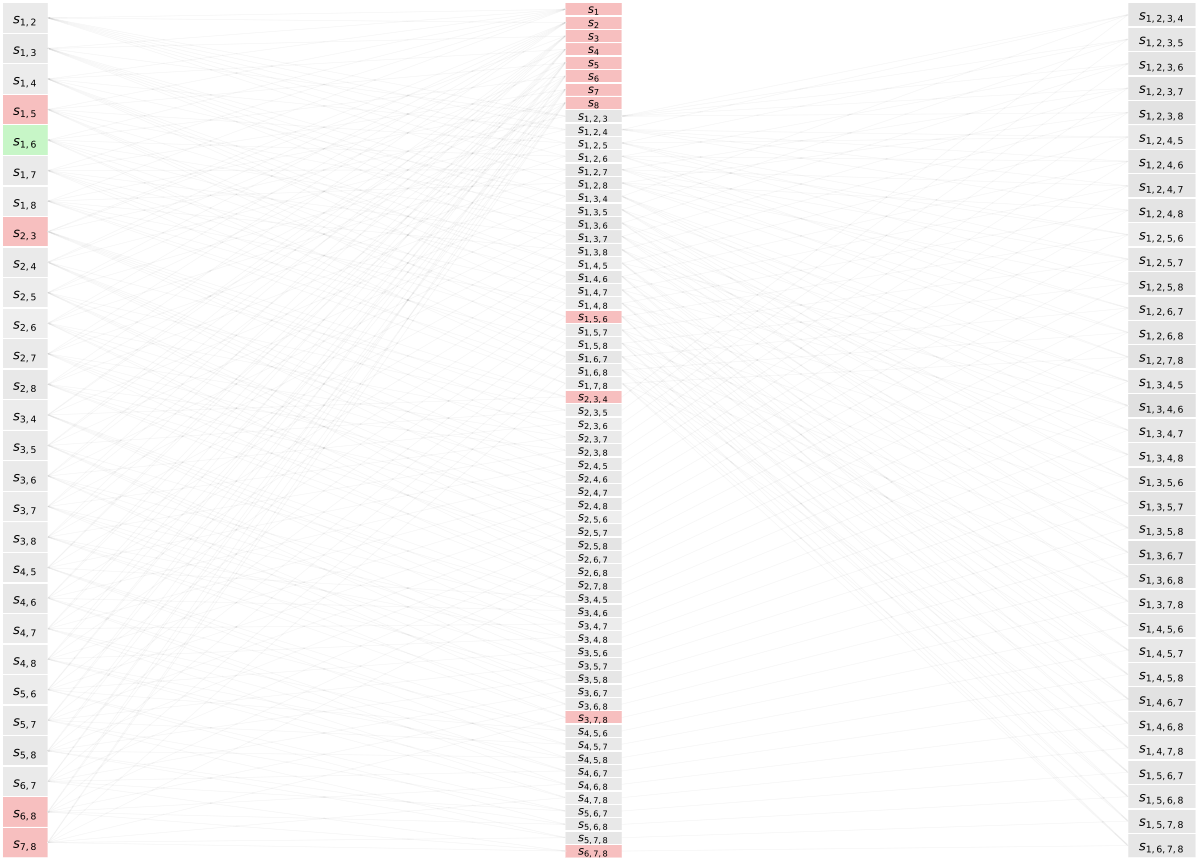


FIG. 27. Three-layer hierarchical network structure for 8-point CHY integrand pole selection. The network comprises: (layer 1) two-particle poles forming the base layer, (layer 2) single-particle and three-particle poles in the intermediate layer, and (layer 3) four-particle poles at the top. Gray edges encode the dependency relationships from the recursion relation $K(s_A) = \frac{1}{(|A|-2)} \sum_{B \subset A, |B|=|A|-1} K(s_B)$. Red nodes represent fixed poles that must be preserved during the algorithm, including the target pole s_{15} and all compatible poles necessary for a consistent amplitude structure.

$$K(s_A) = \frac{1}{(|A|-2)} \sum_{B\subset A, |B|=|A|-1} K(s_B). \tag{D3}$$

As illustrated in Fig. 27, this creates a complex network structure where two-particle and single-particle poles form the foundational layer, three-particle poles constitute the intermediate layer, and four-particle poles occupy the top layer. We fix all poles compatible with s_{15} (marked in red) to ensure they remain unchanged throughout the discrete automatic differentiation process.

1. Removing incompatible pole s_{16}

Our initial step involves removing the incompatible pole s_{16} by decrementing its K-value

$$K^*(s_{16}) = K(s_{16}) - 1 = 1. \tag{D4}$$

The upward propagation phase, shown in Fig. 28, demonstrates how the initial modification cascades through the network hierarchy. The algorithm encounters fixed constraints at nodes s_{156}, s_1, and s_6, which cannot be modified due to physical requirements. This triggers the backward propagation mechanism.

The backward propagation phase, illustrated in Fig. 29, shows the algorithm's sophisticated constraint handling. The system compensates for fixed constraints by selecting appropriate alternative poles and adjusting their values. Concurrently, changes propagate to the third layer, maintaining global consistency.

As shown in Fig. 30, the compensatory changes from the previous step now generate their own propagation effects. The algorithm encounters additional fixed points, demonstrating the iterative nature of the constraint resolution process.

The second backward propagation cycle, depicted in Fig. 31, shows how the algorithm systematically resolves cascading constraints. The selection of s_{58} as a compensatory adjustment illustrates a workable solution within the constraint space.

The final propagation stages, shown in Figs. 32 and 33, demonstrate the algorithm's convergence. The absence of additional constraint violations indicates successful completion of the pole removal process.

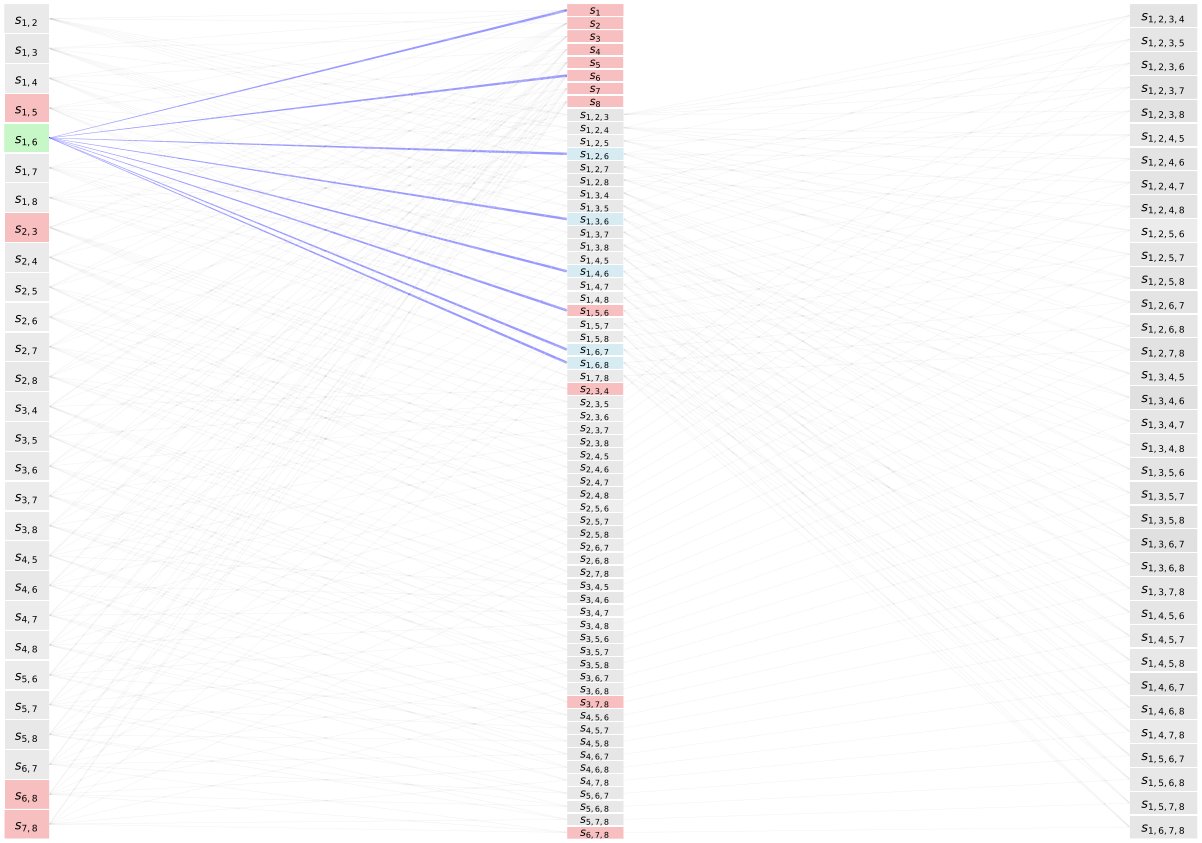


FIG. 28. Initial upward propagation phase in the discrete automatic differentiation algorithm. The modification $\Delta K = -1$ at pole s_{16} (highlighted in green) triggers upward propagation throughout the network hierarchy. Blue arrows trace the propagation pathways to all three-particle supersets and single-particle subsets containing pole s_{16}. The algorithm systematically decrements the K-values of s_{16}'s three-particle supersets: s_{126}, s_{136}, s_{146}, s_{156}, s_{167}, and s_{168} by 1 unit each, following the recursion constraints. Critical fixed constraints are encountered at s_{156}, s_1, and s_6 (red nodes), which triggers the backward propagation mechanism to maintain network consistency.

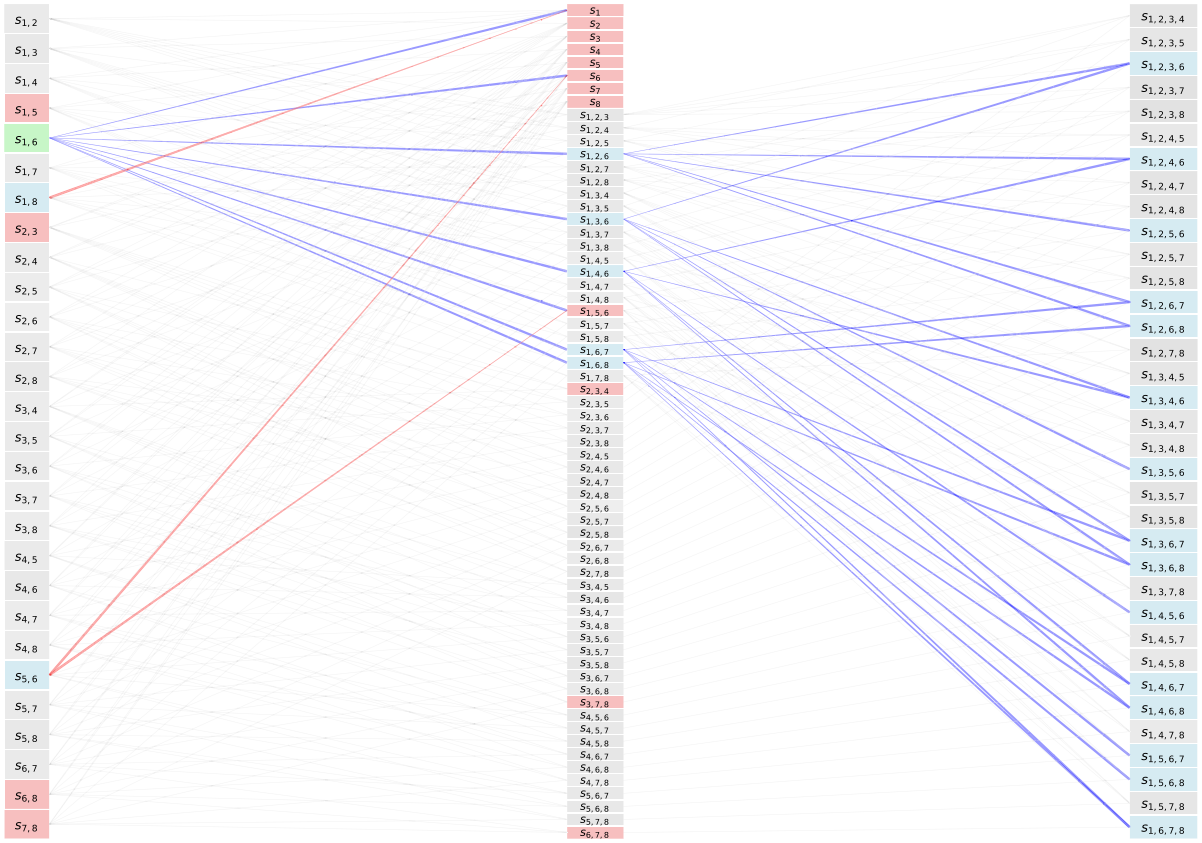


FIG. 29. Backward propagation with simultaneous multilayer updates. When upward propagation encounters fixed constraints, the discrete automatic differentiation algorithm activates backward propagation (red arrows) to redistribute changes while preserving all recursion relations. The algorithm strategically selects s_{56} (intersection of dependent subsets from fixed point s_{156} and superset of fixed node s_6) and s_{18} (superset of fixed point s_1), incrementing their K-values by $+1$. Simultaneously, the algorithm propagates from the second to third layer (blue arrows), updating four-point supersets by -0.5 to maintain consistency across all hierarchical levels. These -0.5 updates are in the K normalization; in $\tilde{K}$ they correspond to integer updates.

Upon completion of the discrete automatic differentiation process, the algorithm removes the incompatible pole s_{16} while preserving the imposed constraints in this example. The modified CHY integrand becomes

$$I_8^* = \frac{z_{18}z_{28}z_{36}}{z_{12}z_{15}^2z_{16}z_{17}z_{23}^2z_{24}z_{26}z_{34}z_{37}z_{38}z_{45}z_{48}z_{56}z_{68}^2z_{78}^2}, \quad \text{(D5)}$$

which produces the refined amplitude

$$\frac{1}{s_{15}s_{23}s_{68}s_{234}s_{678}} + \frac{1}{s_{15}s_{23}s_{78}s_{234}s_{678}} + \frac{1}{s_{15}s_{78}s_{156}s_{378}s_{1256}} + \frac{1}{s_{15}s_{23}s_{78}s_{156}s_{234}}. \quad \text{(D6)}$$

This result isolates the terms containing the desired pole s_{15} in this example, illustrating the behavior of our neural-inspired approach for higher-multiplicity CHY integrands.

The 8-point example highlights several key features of our discrete automatic differentiation framework:

(i) *Three-layer architecture*: Unlike the 6-point case with its two-layer structure, the 8-point example requires a three-layer network, demonstrating scalability to higher multiplicities.

(ii) *Cascading constraint resolution*: The algorithm successfully handles multiple rounds of backward propagation, showing robustness in complex constraint environments.

(iii) *Exact constraint satisfaction*: Throughout the entire process, all recursion relations remain exactly satisfied, with no approximations or numerical errors.

This 8-point demonstration is consistent with the practical applicability of our approach to realistic amplitude calculations and provides a starting point for extensions to higher multiplicities and more complex pole structures. The systematic nature of the constraint resolution process suggests that the method may scale effectively as we move toward higher multiplicities relevant to current amplitude computation capabilities.

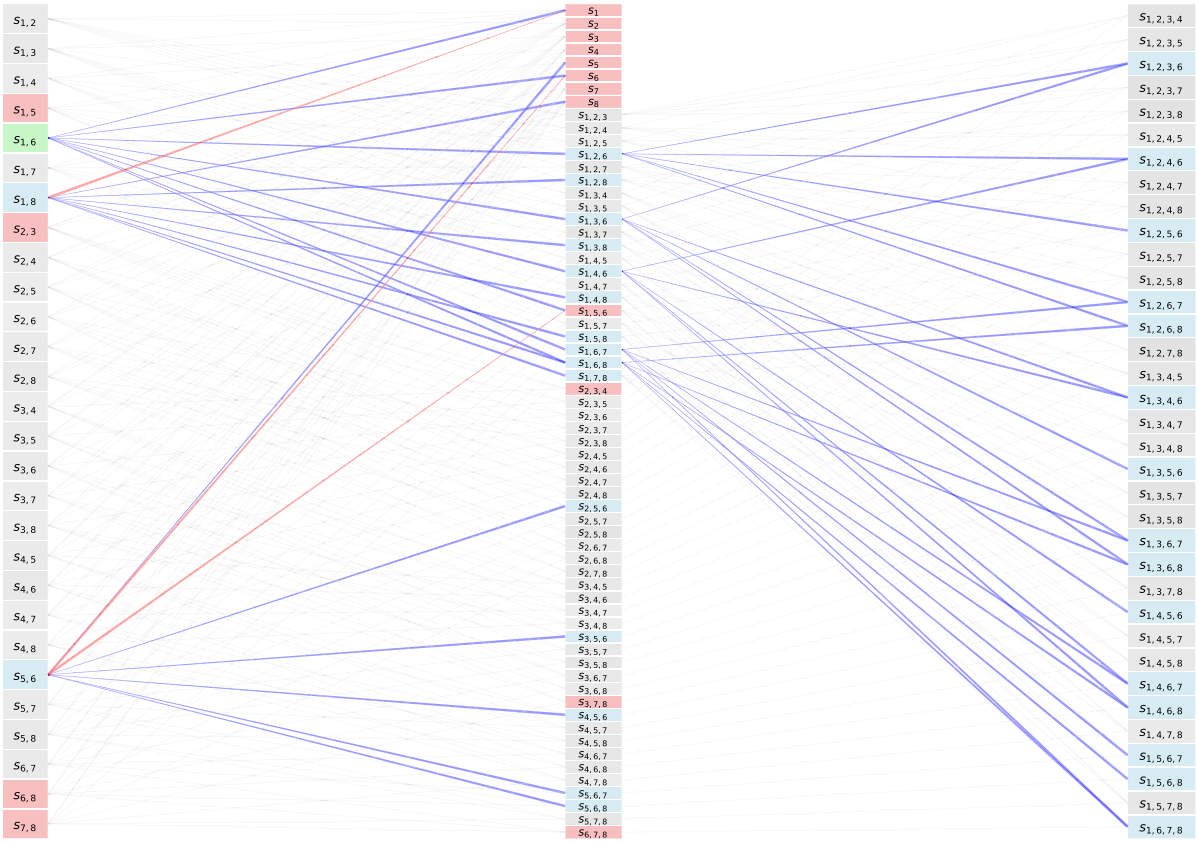


FIG. 30. Second upward propagation cascade from compensatory modifications. The changes introduced at s_{18} and s_{56} during backward propagation now trigger their own upward propagation sequences (blue arrows). Each pole's three-particle supersets receive $+1$ increments: s_{18} affects s_{128}, s_{138}, s_{148}, s_{158}, s_{168}, and s_{178}, while s_{56} affects s_{156}, s_{256}, s_{356}, s_{456}, s_{567}, and s_{568}. The propagation encounters additional fixed points s_5 and s_8, necessitating another backward propagation cycle to resolve the emerging constraint conflicts.

APPENDIX E: ADDITIONAL 8-PT HIGHER-ORDER POLE REDUCTION EXAMPLE (FIGURES)

We now demonstrate the scalability of our enhanced decomposition method through a more complex eight-point example. This case illustrates how the systematic approach based on 0-regular graph decomposition can maintain computational advantages in this example as the complexity increases.

Consider the eight-point CHY integrand

$$I_8 = \frac{1}{z_{12}^3 z_{15} z_{26} z_{36} z_{38}^3 z_{45}^3 z_{47} z_{67}^2 z_{78}}, \tag{E1}$$

whose 4-regular graph representation is shown in Fig. 34.

Using the pole index formula, we identify all subsets with $\chi(A) > 0$

$$\{1,2\}, \qquad \{3,8\}, \qquad \{4,5\}, \qquad \{1,2,4,5\} \tag{E2}$$

where $\chi(A) = 1$ for each, indicating four constraints that must be addressed. The presence of the four-particle pole s_{1245} makes this example particularly challenging, as it introduces correlations between the decomposition steps.

In the original method [7], since the CHY integrand I_8 has four higher-order poles s_{12}, s_{38}, s_{45}, and s_{1245}, one multiplies I_8 by four basic cross-ratio identities, one for each of the subsets $\{1,2\}$, $\{3,8\}$, $\{4,5\}$, and $\{1,2,4,5\}$. Since the identities for $\{1,2\}$, $\{3,8\}$, and $\{4,5\}$ contain 5 terms and the identity for $\{1,2,4,5\}$ contains 9 terms, multiplying these four identities decomposes the original CHY integrand into $5 \times 5 \times 5 \times 9 = 1125$ terms. The (1125) count refers to a naive simultaneous multiplication of the four identities. In the efficiency comparison below, the (265) count corresponds to the sequential systematic implementation of Ref. [7], after separating simple-pole terms at intermediate stages.

To reduce these higher-order poles and redundant terms, we construct a six-point cross-ratio factor $\frac{z_{12}z_{38}z_{45}}{z_{18}z_{35}z_{42}}$ targeting the three two-particle double poles. Figure 35 shows its systematic decomposition.

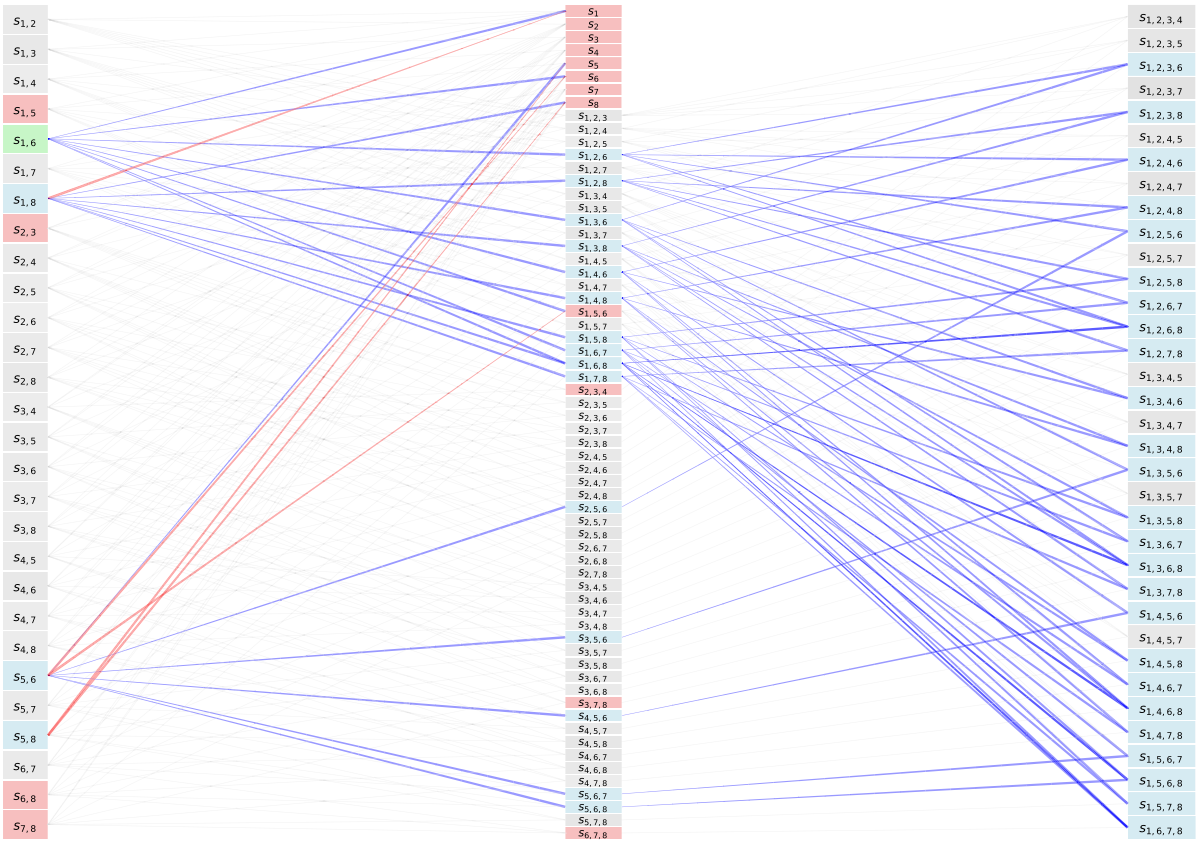


FIG. 31. Second backward propagation cycle with interlayer constraint resolution. The encounter with fixed points s_5 and s_8 activates another backward propagation phase (red arrows from second to initial layer). The algorithm identifies s_{58} as the intersection of dependent supersets from fixed nodes s_5 and s_8, decrementing its K-value by -1 to compensate for the constrained propagation. Simultaneously, upward propagation continues to the third layer (blue arrows), where four-point supersets of the affected three-particle poles receive $+0.5$ increments, ensuring that all recursion relations remain exactly satisfied throughout the multilayer network. These $+0.5$ updates are in the K normalization; in $\tilde{K}$ they correspond to integer updates.

1. Round 1: Initial decomposition

Based on the cross-ratio decomposition, we select two strategic identities

$$I_8[\{1,2\},1,8] = -\frac{s_{23}z_{12}z_{38}}{s_{12}z_{18}z_{23}} - \frac{s_{24}z_{12}z_{48}}{s_{12}z_{18}z_{24}} - \frac{s_{25}z_{12}z_{58}}{s_{12}z_{18}z_{25}} - \frac{s_{26}z_{12}z_{68}}{s_{12}z_{18}z_{26}} - \frac{s_{27}z_{12}z_{78}}{s_{12}z_{18}z_{27}}, \tag{E3}$$

$$I_8[\{4,5\},5,3] = -\frac{s_{41}z_{13}z_{45}}{s_{45}z_{14}z_{35}} - \frac{s_{42}z_{23}z_{45}}{s_{45}z_{24}z_{35}} - \frac{s_{46}z_{36}z_{45}}{s_{45}z_{35}z_{46}} - \frac{s_{47}z_{37}z_{45}}{s_{45}z_{35}z_{47}} - \frac{s_{48}z_{38}z_{45}}{s_{45}z_{35}z_{48}}. \tag{E4}$$

Multiplying I_8 by these identities produces $5^2 = 25$ terms, whose structure is shown in Fig. 36.

The decomposition yields three categories:

(i) 9 terms with only simple poles.
(ii) 12 terms with double pole s_{38}.
(iii) 4 terms with double poles s_{38} and s_{1245}.

2. Round 2: Eliminating the s_{38} double pole

For the 12 terms containing double pole s_{38}, we apply the identity

$$I_8[\{3,8\},3,4] = -\frac{s_{81}z_{14}z_{38}}{s_{38}z_{18}z_{34}} - \frac{s_{82}z_{24}z_{38}}{s_{38}z_{28}z_{34}} - \frac{s_{85}z_{45}z_{38}}{s_{38}z_{34}z_{58}} - \frac{s_{86}z_{46}z_{38}}{s_{38}z_{34}z_{68}} - \frac{s_{87}z_{47}z_{38}}{s_{38}z_{34}z_{78}}. \tag{E5}$$

Figure 37 illustrates this decomposition for one representative term.

This process transforms the 12 terms with double pole s_{38} into $12 \times 5 = 60$ terms with only simple poles.

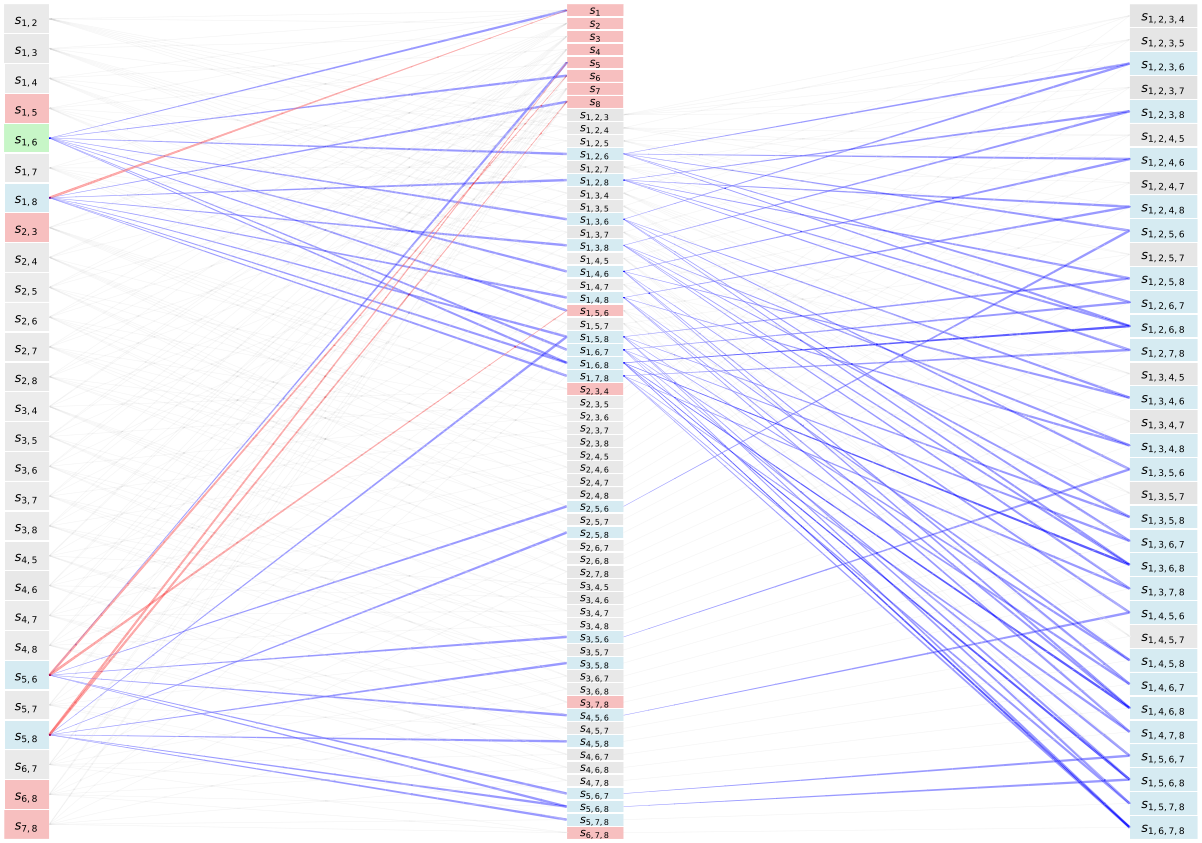


FIG. 32. Final upward propagation completing the constraint resolution. The modification at s_{58} triggers the final upward propagation phase (blue arrows), decrementing its three-particle supersets s_{158}, s_{258}, s_{358}, s_{458}, s_{568}, s_{578}, and s_{678} by -1 each. This propagation proceeds without encountering additional fixed constraints, indicating successful resolution of all constraint conflicts.

For the 4 terms containing both s_{38} and s_{1245} double poles, applying identity (E5) yields 20 terms, of which 12 have only simple poles while 8 retain the double pole s_{1245}. Figure 38 shows a representative example.

3. Round 3: Final decomposition of s_{1245}

The remaining 8 terms with double pole s_{1245} require the application of a nine-term identity

$$\begin{aligned} &I_8[\{1,2,4,5\},1,3] \\ &= -\frac{s_{26}z_{12}z_{36}}{s_{1245}z_{13}z_{26}} - \frac{s_{46}z_{14}z_{36}}{s_{1245}z_{13}z_{46}} - \frac{s_{56}z_{15}z_{36}}{s_{1245}z_{13}z_{56}} \\ &\quad - \frac{s_{27}z_{12}z_{37}}{s_{1245}z_{13}z_{27}} - \frac{s_{28}z_{12}z_{38}}{s_{1245}z_{13}z_{28}} - \frac{s_{47}z_{14}z_{37}}{s_{1245}z_{13}z_{47}} \\ &\quad - \frac{s_{48}z_{14}z_{38}}{s_{1245}z_{13}z_{48}} - \frac{s_{57}z_{15}z_{37}}{s_{1245}z_{13}z_{57}} - \frac{s_{58}z_{15}z_{38}}{s_{1245}z_{13}z_{58}}. \end{aligned} \tag{E6}$$

Figure 39 demonstrates the final decomposition step.

4. Summary and computational efficiency

Through this three-round process, the eight-point CHY integrand I_8 with four double poles is systematically decomposed into $9+60+12+72=153$ terms, each containing only simple poles. The improvement over traditional methods is summarized in the following table:

Method	Round 1	Round 2	Round 3	Round 4	Total
Systematic algorithm [7]	5 [5]	25 [20]	105 [20]	265 [0]	265
Our method	25 [16]	89 [8]	153 [0]		153

Numbers in brackets [·] indicate terms with remaining higher-order poles at each stage.

Our enhanced method achieves a reduction factor by:

(1) Systematically decomposing cross-ratio factors using the 0-regular graph structure.
(2) Applying identities sequentially to minimize intermediate term proliferation.
(3) Leveraging the hierarchical nature of pole constraints to optimize the decomposition order.

This eight-point example suggests that our graph-theoretic approach to higher-order pole decomposition can maintain computational advantages as complexity increases, making it a useful tool for the amplitude calculations considered here.

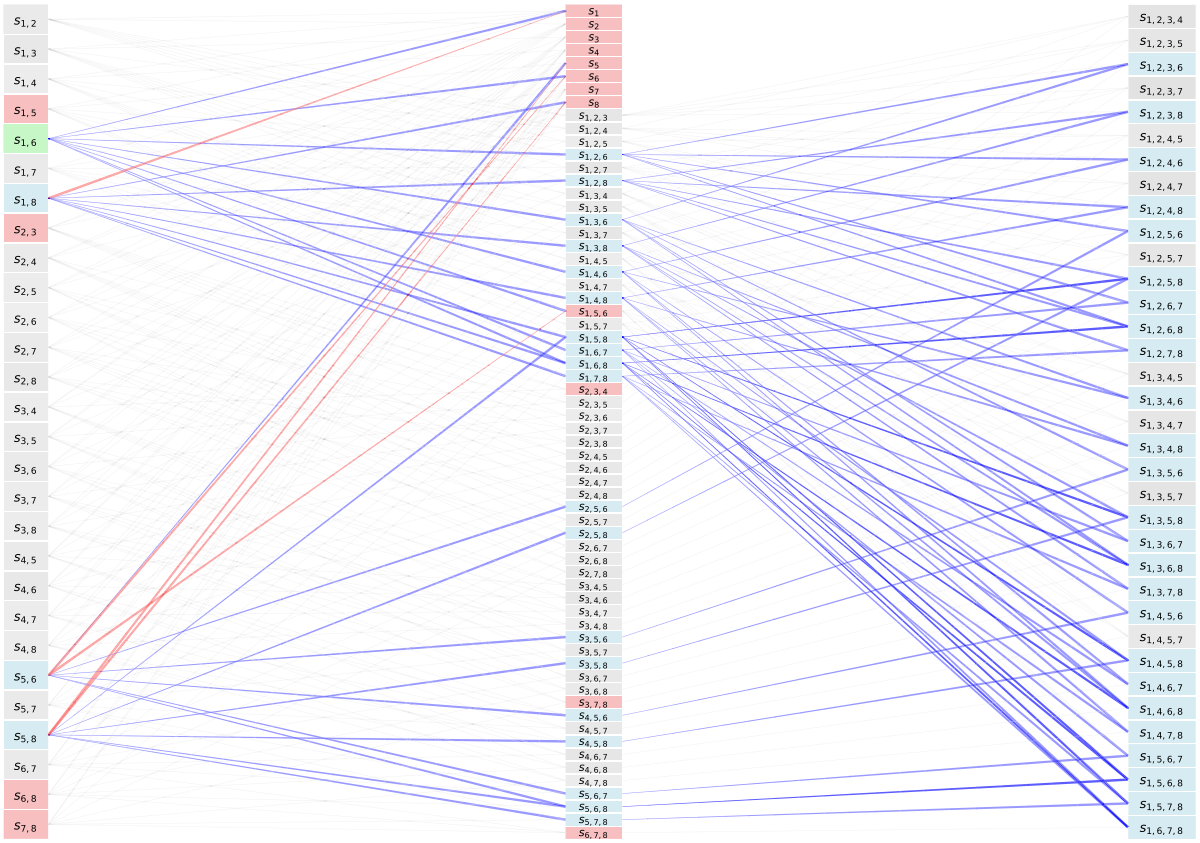


FIG. 33. The algorithm seamlessly continues to update the third layer, where four-point supersets receive corresponding -0.5 decrements to maintain the recursive relationships across all network levels. This systematic propagation through all network layers keeps each recursion relation $K(s_A) = \frac{1}{(|A|-2)} \sum_{B \subset A, |B|=|A|-1} K(s_B)$ satisfied. The -0.5 updates are shown in the K normalization; in $\tilde{K}$ they correspond to integer updates. The successful completion of this propagation without further constraint violations confirms that the removal of incompatible pole s_{16} has been achieved while preserving all required poles and maintaining network consistency.

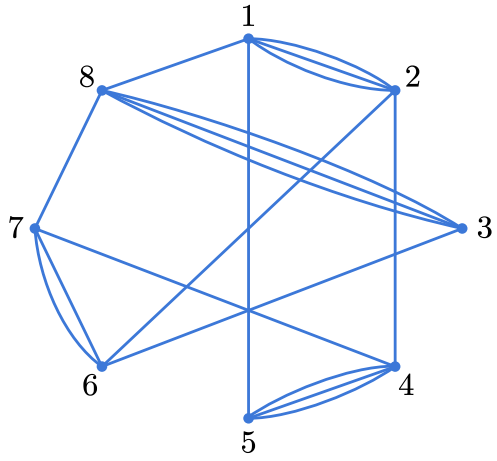


FIG. 34. The 4-regular graph representation of the eight-point CHY integrand I_8. The edge multiplicities directly encode the denominator powers in equation (E1). The triple edges at (1, 2), (3, 8), and (4, 5) indicate three double poles, while the more complex edge structure suggests an additional higher-order pole that merits further analysis.

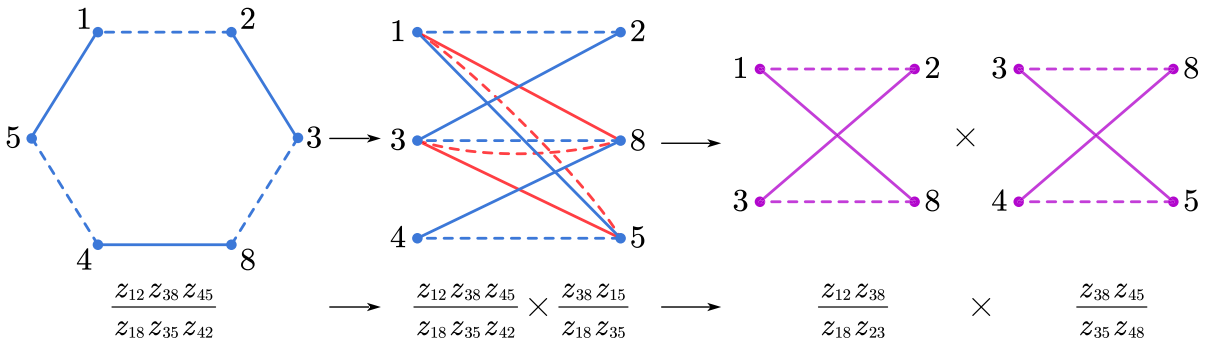


FIG. 35. Decomposition of the six-point cross-ratio factor for the eight-point integrand. The bipartite representation reveals the natural factorization into $\frac{z_{12}z_{38}}{z_{18}z_{32}} \times \frac{z_{38}z_{45}}{z_{35}z_{48}}$, providing the foundation for our two-stage decomposition strategy. This decomposition is specifically designed to address the three two-particle double poles while minimizing interference with the four-particle pole s_{1245}.

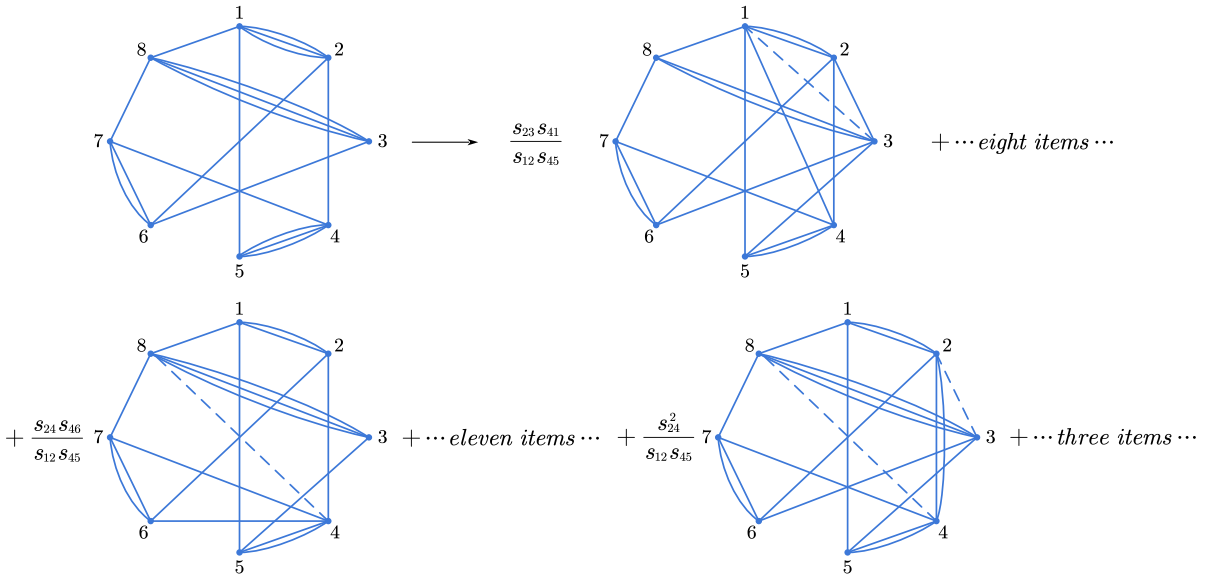


FIG. 36. Round 1 decomposition results: The initial application of identities (E3) and (E4) produces 25 terms organized in a 5×5 grid. The initial 9 terms (top-left, shown in gray) already contain only simple poles. The central 12 terms retain the double pole s_{38}, while the final 4 terms (bottom-right) contain both s_{38} and s_{1245} as double poles, requiring the most extensive further treatment.

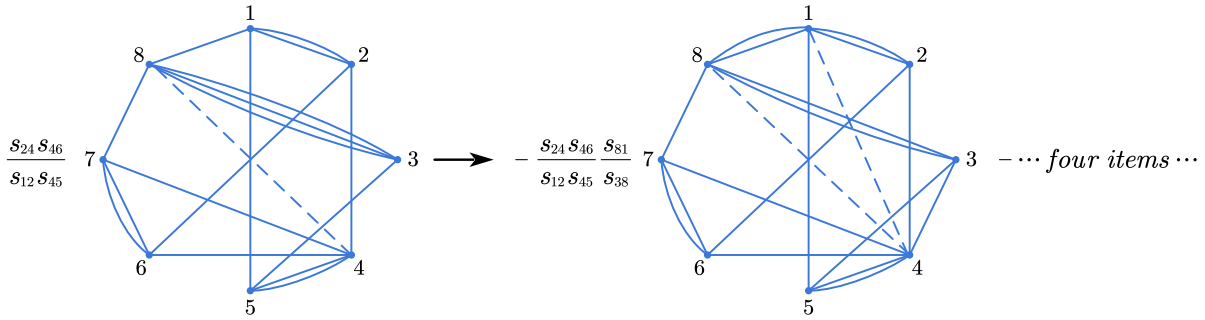


FIG. 37. Round 2 decomposition of a term with double pole s_{38}. The application of identity (E5) to the term $\frac{s_{24}s_{46}}{s_{12}s_{45}} \frac{z_{84}}{z_{21}^2 z_{42} z_{51} z_{53} z_{54}^2 z_{62} z_{64} z_{74} z_{76}^2 z_{81} z_{83}^3 z_{87}}$ produces 5 terms with only simple poles. The cross-ratio factors in the identity precisely cancel the z_{83}^3 contribution, reducing the pole order as required.

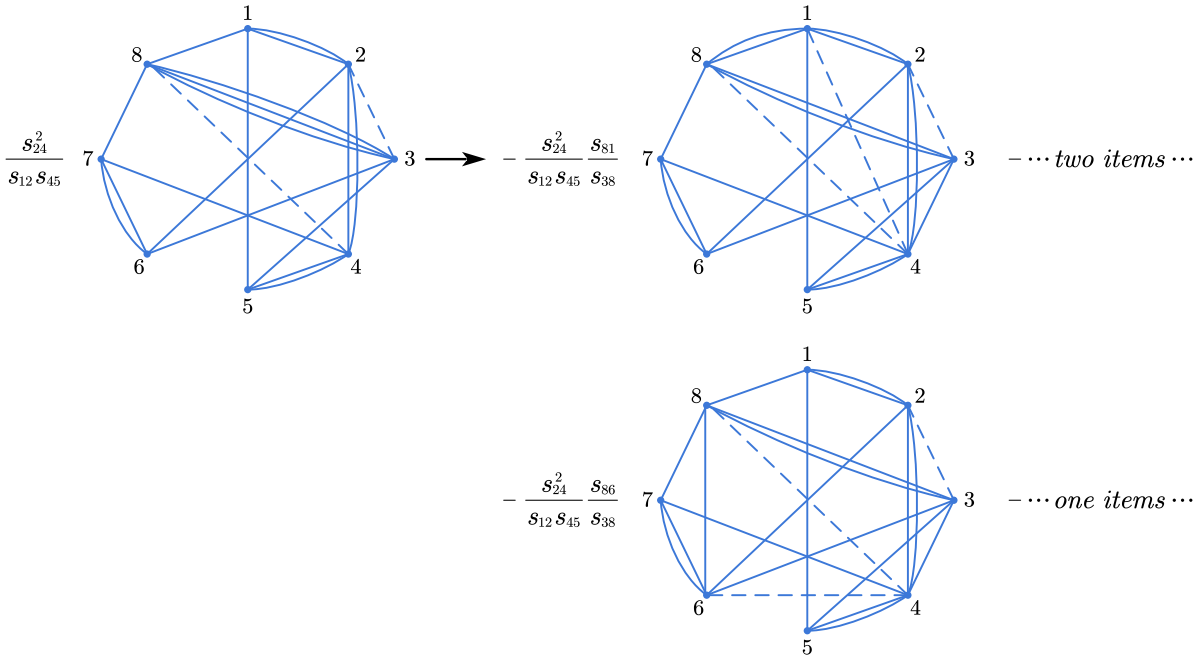


FIG. 38. Round 2 decomposition of a term with double poles s_{38} and s_{1245}. After applying identity (E5), the resulting 5 terms split into two categories: 3 terms (shown in gray) with only simple poles, and 2 terms (shown in white) that still contain the double pole s_{1245}, necessitating a third round of decomposition.

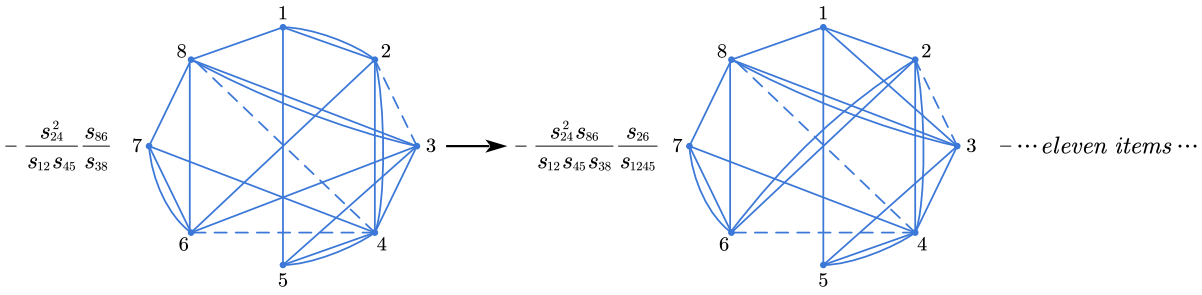


FIG. 39. Round 3 final decomposition. The nine-term identity (E6) applied to a term with double pole s_{1245} produces 9 CHY integrands with only simple poles. This completes the systematic reduction of all higher-order poles in the original eight-point integrand. The complexity of this final step (9 terms) reflects the four-particle nature of the s_{1245} pole.

[1] F. Cachazo, S. He, and E. Y. Yuan, Scattering equations and kawai-llewellen-tye orthogonality, Phys. Rev. D **90**, 065001 (2014).

[2] F. Cachazo, S. He, and E. Y. Yuan, Scattering of massless particles: Scalars, gluons and gravitons, J. High Energy Phys. 07 (2014) 033.

[3] F. Cachazo, S. He, and E. Y. Yuan, Scattering of massless particles in arbitrary dimensions, Phys. Rev. Lett. **113**, 171601 (2014).

[4] C. Baadsgaard, N. E. J. Bjerrum-Bohr, J. L. Bourjaily, and P. H. Damgaard, Integration rules for scattering equations, J. High Energy Phys. 09 (2015) 129.

[5] C. Baadsgaard, N. E. J. Bjerrum-Bohr, J. L. Bourjaily, and P. H. Damgaard, Scattering equations and feynman diagrams, J. High Energy Phys. 09 (2015) 136.

[6] F. Cachazo and H. Gomez, Computation of Contour Integrals on $M_{0,n}$, J. High Energy Phys. 04 (2016) 108.

[7] C. Cardona, B. Feng, H. Gomez, and R. Huang, Cross-ratio identities and higher-order poles of CHY-integrand, J. High Energy Phys. 09 (2016) 133.

[8] R. Huang, B. Feng, M.-x. Luo, and C.-J. Zhu, Feynman rules of higher-order poles in CHY construction, J. High Energy Phys. 06 (2016) 013.

[9] K. Zhou, Analytic expressions of amplitudes by the cross-ratio identity method, Eur. Phys. J. C **77**, 388 (2017).
[10] R. Huang, F. Teng, and B. Feng, Permutation in the CHY formulation, Nucl. Phys. **B932**, 323 (2018).
[11] A. Schrijver, *Theory of Linear and Integer Programming*, Wiley-Interscience Series in Discrete Mathematics and Optimization (John Wiley & Sons, Chichester, 1998).
[12] L. A. Wolsey, *Integer Programming*, Wiley-Interscience Series in Discrete Mathematics and Optimization (John Wiley & Sons, New York, 1998).
[13] G. De Laurentis, Analytical amplitudes from numerical solutions of the scattering equations, J. High Energy Phys. 02 (2020) 194.
[14] T. Peraro, Scattering amplitudes over finite fields and multivariate functional reconstruction, J. High Energy Phys. 12 (2016) 030.
[15] T. Peraro, FiniteFlow: Multivariate functional reconstruction using finite fields and dataflow graphs, J. High Energy Phys. 07 (2019) 031.
[16] J. Klappert and F. Lange, Reconstructing rational functions with FireFly, Comput. Phys. Commun. **247**, 106951 (2020).
[17] H. R. P. Ferguson, D. H. Bailey, and S. Arno, Analysis of PSLQ, an integer relation finding algorithm, Math. Comput. **68**, 351 (1999).
[18] S. Laporta, High-precision calculation of multiloop feynman integrals by difference equations, Int. J. Mod. Phys. A **15**, 5087 (2000).
[19] F. Cachazo, S. He, and E. Y. Yuan, Scattering equations and matrices: From Einstein to Yang-Mills, DBI and NLSM, J. High Energy Phys. 07 (2015) 149.
[20] L. Dolan and P. Goddard, The polynomial form of the scattering equations, J. High Energy Phys. 07 (2014) 029.
[21] Y. Zhang, On the new factorizations of Yang-Mills amplitudes, J. High Energy Phys. 02 (2025) 074.
[22] Y. Zhang, New factorizations of Yang-Mills amplitudes, Phys. Rev. D **111**, 085004 (2025).
[23] S. He and E. Y. Yuan, One-loop scattering equations and amplitudes from forward limit, Phys. Rev. D **92**, 105004 (2015).
[24] L. Mason and D. Skinner, Ambitwistor strings and the scattering equations, J. High Energy Phys. 07 (2014) 048.
[25] Y. Geyer, L. Mason, R. Monteiro, and P. Tourkine, Loop integrands for scattering amplitudes from the riemann sphere, Phys. Rev. Lett. **115**, 121603 (2015).
[26] B. Feng, CHY-construction of planar loop integrands of cubic scalar theory, J. High Energy Phys. 05 (2016) 061.
[27] B. Feng and C. Hu, One-loop CHY-integrand of biadjoint scalar theory, J. High Energy Phys. 02 (2020) 187.
[28] X. Gao, S. He, and Y. Zhang, Labelled tree graphs, feynman diagrams and disk integrals, J. High Energy Phys. 11 (2017) 144.
[29] F. Cachazo, N. Early, A. Guevara, and S. Mizera, Scattering equations: From projective spaces to tropical grassmannians, J. High Energy Phys. 06 (2019) 039.
[30] N. Arkani-Hamed, Y. Bai, and T. Lam, Positive geometries and canonical forms, J. High Energy Phys. 11 (2017) 039.
[31] S. Mizera, Combinatorics and topology of Kawai-Lewellen-Tye relations, J. High Energy Phys. 08 (2017) 097.
[32] S. Mizera, Scattering amplitudes from intersection theory, Phys. Rev. Lett. **120**, 141602 (2018).
[33] S. Mizera, *Aspects of Scattering Amplitudes and Moduli Space Localization* (Springer, Cham, 2020).
[34] F. Brown and C. Dupont, Single-valued integration and superstring amplitudes in genus zero, Commun. Math. Phys. **382**, 815 (2021).
[35] Y.-H. He, Machine-learning the string landscape, Phys. Lett. B **774**, 564 (2017).
[36] Y.-H. He, *The Calabi-Yau Landscape: From Geometry, to Physics, to Machine Learning*, Lecture Notes in Mathematics, Vol. 2293 (Springer, Cham, 2021).
[37] A. Ashmore, Y. He, and B. A. Ovrut, Machine learning Calabi–Yau metrics, Fortschr. Phys. **68**, 2000068 (2020).
[38] E. Bothmann, T. Janssen, M. Knobbe, T. Schmale, and S. Schumann, Exploring phase space with neural importance sampling, SciPost Phys. **8**, 069 (2020).
[39] R. Winterhalder, V. Magerya, E. Villa, S. P. Jones, M. Kerner, A. Butter, G. Heinrich, and T. Plehn, Targeting multi-loop integrals with neural networks, SciPost Phys. **12**, 129 (2022).
[40] A. Dersy, M. D. Schwartz, and X. Zhang, Simplifying polylogarithms with machine learning, Int. J. Data Sci. Math. Sci. **1**, 135 (2024).
[41] T. Cai, G. W. Merz, F. Charton, N. Nolte, M. Wilhelm, K. Cranmer, and L. J. Dixon, Transforming the bootstrap: Using transformers to compute scattering amplitudes in planar $N = 4$ Super Yang-Mills theory, Mach. Learn. Sci. Technol. **5**, 035073 (2024).
[42] A. Alnuqaydan, S. Gleyzer, and H. Prosper, SYMBA: Symbolic computation of squared amplitudes in high energy physics with machine learning, Mach. Learn. Sci. Technol. **4**, 015007 (2023).
[43] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. **91**, 045002 (2019).
[44] A. von Manteuffel and R. M. Schabinger, A novel approach to integration by parts reduction, Phys. Lett. B **744**, 101 (2015).
[45] S. Ebli, M. Defferrard, and G. Spreemann, Simplicial neural networks, arXiv:2010.03633.
[46] C. Bodnar, F. Frasca, Y. G. Wang, N. Otter, G. F. Montúfar, P. Liò, and M. M. Bronstein, Weisfeiler and Lehman go topological: Message passing simplicial networks, in *Proceedings of the 38th International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research Vol. 139 (PMLR, 2021), pp. 1026–1037.
[47] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe, Weisfeiler and Leman go neural: Higher-order graph neural networks, arXiv:1810.02244.
[48] F. Cachazo, S. He, and E. Y. Yuan, Einstein-Yang-Mills scattering amplitudes from scattering equations, J. High Energy Phys. 01 (2015) 121.
[49] T. Adamo, E. Casali, and D. Skinner, Ambitwistor strings and the scattering equations at one loop, J. High Energy Phys. 04 (2014) 104.
[50] B. Feng and Y. Zhang, Note on the labelled tree graphs, J. High Energy Phys. 12 (2020) 096.

[51] F. Cachazo, K. Yeats, and S. Yusim, Compatible cycles and CHY integrals, J. High Energy Phys. 12 (2019) 105.
[52] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity* (Courier Corporation, New York, 1998).
[53] B. Korte and J. Vygen, *Combinatorial Optimization: Theory and Algorithms*, 5th ed., Algorithms and Combinatorics (Springer, New York, 2012).
[54] J. E. Mitchell, *Encyclopedia of Optimization* (Springer, Boston, MA, 2008).
[55] R. E. Bixby, A Brief History of Linear and Mixed-Integer Programming Computation, Documenta *Mathematica* series, 10.4171/dms/6/16 (2012).
[56] Y. Bengio, A. Lodi, and A. Prouvost, Machine learning for combinatorial optimization: A methodological tour d'horizon, arXiv:1811.06128.
[57] M. Gasse, D. Chételat, N. Ferroni, L. Charlin, and A. Lodi, Exact combinatorial optimization with graph convolutional neural networks, arXiv:1906.01629.
[58] Wolfram Research, Inc., *Mathematica*, version 14.2 [software], https://www.wolfram.com/mathematica/ (2024).
[59] P. Virtanen *et al.* (SciPy 1.0 Contributors), SciPy 1.0: Fundamental algorithms for scientific computing in python, Nat. Methods **17**, 261 (2020).
[60] Q. Huangfu and J. A. J. Hall, Parallelizing the dual revised simplex method, Math. Program. Comput. **10**, 119 (2018).
[61] S. Diamond and S. Boyd, CVXPY: A python-embedded modeling language for convex optimization, J. Mach. Learn. Res. **17**, 1 (2016).
[62] S. Mitchell, `PuLP`: A linear programming toolkit for Python [software], https://coin-or.github.io/pulp/ (2011).
[63] B. Feng, C. Hu, and Y. Zhang, Note on solutions of scattering equations, Phys. Rev. D **105**, 036015 (2022).
[64] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications* (Springer, New York, 1976).
[65] R. Diestel, *Graph Theory*, 5th ed., Graduate Texts in Mathematics (Springer, New York, 2017), Vol. 173.