Seedling: learning how much to stage in integration-by-parts reduction

Matthew D. Schwartz*
Department of Physics, Harvard University

August 8, 2026

Abstract

Multi-loop scattering amplitudes pass through one dominant step: integration-by-parts reduction, which rewrites a calculation's thousands to millions of Feynman integrals in terms of a few master integrals. For a quarter century the workhorse has been Laporta's algorithm: generate the identities on a lattice of seed integrals, then eliminate. The algorithm leaves one choice open, how many seeds to generate, and practice fills it with a uniform hand-chosen cushion: too few and the reduction does not close, too many and the solver drowns, and the cushion the hungriest sector needs is the cushion every sector gets. Machine learning has recently entered reduction, learning which identities to keep, in what order to eliminate, and where seeds live; to our knowledge, no published work learns how much staging each sector needs. `seedling` learns exactly that: a seven-number ridge model, printed in full, predicts per-class seed margins, rendered into per-sector staging schedules, from six features computed before any identity is generated. Whether the margins help is a scaling question, so we built an instrument: a three-arm ladder with leave-one-out on every fitted exponent and censoring reported as an outcome. On the ladder, `seedling`'s staged systems grow with sector count at exponent 2.630 against the constant cushion's 2.932, stable under all five paired rung deletions; at the deepest rung, the three-loop light-by-light family, `seedling` halves the constant-margin staging and completes in 4.79 GB where the legacy staging was OOM-killed above a censored 320 GB bound. Against the stock recipe the growth ordering is inconclusive at this ladder depth. On a nonplanar four-loop family outside the development corpus, `seedling` reproduces every independent reduction on record exactly, at the same cost class: correctness at four loops, no speed claim. Pointed at the strongest recent external method, the instrument reproduces both of its published scaling laws on that method's own code.

Contents

*Work done with Anthropic, but results and views are not endorsed by Anthropic.

1 Introduction

Multi-loop scattering amplitudes are computed today by one dominant route: integration-by-parts reduction, then differential equations. A collider prediction begins as a sum of Feynman diagrams; squaring and summing over spins turns that sum into thousands to millions of scalar Feynman integrals, organized into *families* labeled by the integer powers of a fixed set of propagators. Integration-by-parts (IBP) identities [1, 2] are linear relations among the members of a family, one for each total derivative that integrates to zero in dimensional regularization, and solving them expresses every integral in the family in terms of a small basis of *master integrals*. The masters are then evaluated by differential equations in the kinematic invariants [4–6]: differentiating a master under the integral sign produces family members that the same identities reduce back onto the basis, closing a first-order linear system that can be solved. Reduction is the step this paper is about.

For twenty-five years the workhorse of reduction has been Laporta's algorithm [3]: generate the identities on a lattice of explicit *seed* integrals, order the integrals, and eliminate. The algorithm's practical cost is dominated not by the elimination but by the choice of what to generate — too few seeds and the target system does not close; too many and the solver drowns in equations it never needed. We call that choice *staging*. Production practice handles it with uniform cushions: raise the rank, dot, and numerator bounds by a hand-chosen constant everywhere and hope the family is typical.

The finished reduction uses only a sparse subset of the staged equations, and no a-priori rule identifies the subset. The gap is acknowledged in the field's own tool papers: the authors of the block-triangular reducer Blade note that no a-priori strategy exists for determining a minimal seed set [29], and the headline advances of the newest Kira release are precisely seeding and equation selection [16]. A uniform cushion therefore overstages every sector but the one that set it. The choice Laporta's algorithm leaves open is the amount: how much staging each sector of a family needs. That is the problem this paper answers.

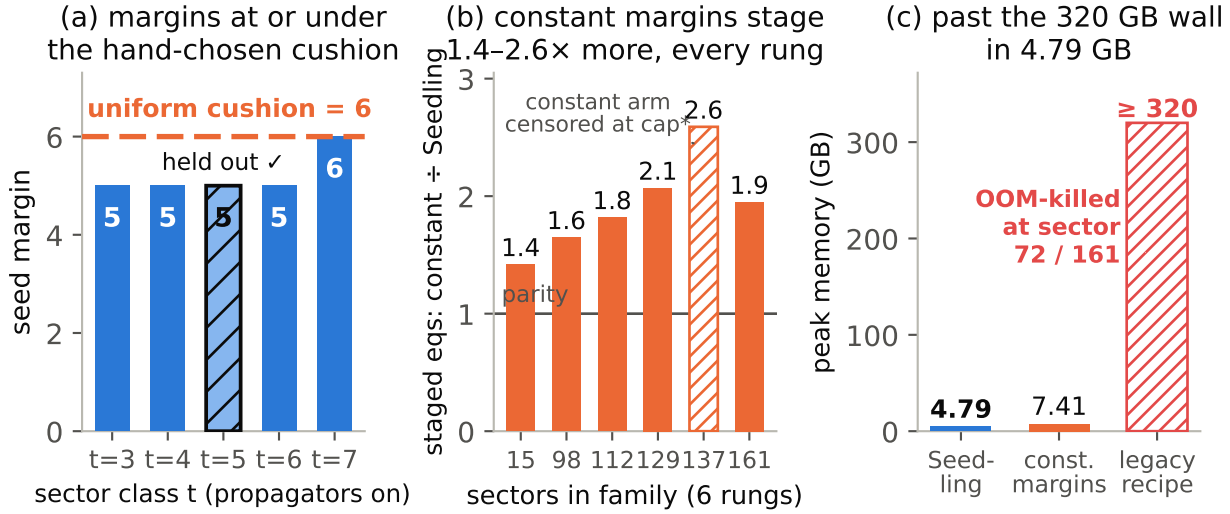
This paper presents a tool and an instrument; Figure 1 shows what they measure. The tool, `seedling`, replaces the uniform cushion with a *learned, frozen, registered* model of per-class seed margins, rendered per sector, wrapped in a pipeline whose every stage is checked and leaves an independently verifiable record. The instrument is a pre-registered scaling ladder (three arms, fixed rules, leave-one-out on every fitted exponent, censoring as an outcome in its own right), built because the question "does the model help?" is a scaling question, and scaling claims in this field are easy to get wrong and hard to check. We report what the instrument found, including the comparison it could *not* decide, and we report the measurement failures it caught along the way, because a tool paper that hides its instrument's incidents cannot be calibrated by the reader.

Section 2 develops the problem statement in full: the identity, the algorithm Laporta built from it, the scaling wall met in production, and the sparsity underneath the wall. Sections 3 and 4 present the tool and the instrument. Section 5 reports the measurements, Section 6 the incidents the measurement system caught, and Section 7 the open items and the outlook.

Seedling: *learned staging for exact Laporta reduction*

A frozen 7-number ridge model predicts each sector's seed margins; exact finite-field elimination and two-prime oracle gates verify everything it stages.

half the staging	**4.79 GB, completed**	$n^{2.63}$ **vs** $n^{2.93}$
6.22M vs 12.14M equations, deepest family, same masters	where the legacy staging OOM-killed above 320 GB	staged-equation growth, stable on all 5 deletions



Beyond the training family (completeness equal, staging never larger):

−45.5%	**−51.7%**	**completed**	**5 / 5**
staged equations, held-out sector class (t = 5)	staged equations, transfer family (double box)	external double pentagon; constant arm censored at cap	four-loop reductions reproduced, masters byte-identical

*staged-equation counts come from the rendered schedules; on the censored rung both arms' counts are schedule-predicted bulks (prediction over prediction; the constant arm never finished generating).

Figure 1: `seedling` at a glance. Read this as: a frozen seven-number ridge model predicting per-class seed margins halves the staged equations of constant margins at the deepest family (6.22M vs 12.14M, same masters), completes that family in 4.79 GB where the legacy staging was OOM-killed above a censored 320 GB bound, grows slowest on the depth ladder ($n^{2.63}$ vs $n^{2.93}$, ordering stable under all five paired rung deletions), and off its training corpus its completeness is equal and its staging never larger. Panels: (a) learned margins against the uniform cushion of standard practice, held-out class hatched; (b) the constant arm's staged equations over `seedling`'s, rung by rung (1.4–2.6×; the censored rung enters as predicted bulk, hatched); (c) peak memory on the deepest family, linear axis, the legacy bar a censored lower bound; bottom strip, the four off-corpus exhibits.

2 The method: from one identity to millions of equations

Beyond leading order, every perturbative prediction of a quantum field theory passes through Feynman integrals: one integration over an unobserved loop momentum for each closed loop of a diagram, with a product of propagators (one factor per internal line) in the denominator. The algebra that turns diagrams into a cross section produces these integrals in bulk: spin sums and numerator reduction generate the same denominators over and over at different integer powers, so a calculation asks for thousands to millions of integrals that differ only in a list of exponents. Fixing the set of propagators and letting the exponents range over the integers defines a *family*; one member is one list of integers. The massless bubble of Eq. (3) below, with its two propagators, is a family with two indices.

The members of a family are linearly related. In dimensional regularization a total derivative integrates to zero, and differentiating any member with respect to a loop momentum (any vector built from the momenta of the problem inserted in the numerator) turns that zero into a linear relation among members whose exponents differ by one, with coefficients rational in the dimension d and the kinematics. Integration-by-parts identities in dimensional regularization were introduced by Tkachov and Chetyrkin–Tkachov [1, 2], who used them to reduce four-loop renormalization-group functions by hand-crafted recursion. Solved as a system, the identities express every member of a family as a rational-function combination of a few *master integrals*, exact in d; the masters are the only integrals that ever have to be computed by any other means.

The smallest real case is a single massive propagator, the one-loop tadpole, and its tower of powers,

$$T(a) \;=\; \int \frac{d^d\ell}{i\pi^{d/2}}\,\frac{1}{(\ell^2-m^2)^a}\,. \tag{1}$$

Take the total derivative of $\ell^\mu\,(\ell^2-m^2)^{-a}$ under the integral sign: $\partial_\mu \ell^\mu$ gives d, the chain rule gives $-2a\,\ell^2/(\ell^2-m^2)$, and writing $\ell^2 = (\ell^2-m^2)+m^2$ turns that term back into members of the tower:

$$0 \;=\; (d-2a)\,T(a) \;-\; 2a\,m^2\,T(a+1) \qquad\Longrightarrow\qquad T(a+1) \;=\; \frac{d-2a}{2a\,m^2}\,T(a)\,. \tag{2}$$

The whole tower collapses onto the single master $T(1)$: every higher power of the propagator is $T(1)$ times an explicit rational function of d and m^2.

The same collapse happens at production scale; only the proportions change. The three-loop calculation that motivated this work needs 1,725 members of a single fifteen-index family, and the identities reduce all of them to 54 masters. Nothing at that size solves by inspection. Laporta's algorithm (Section 2.2) replaces inspection with organization: generate the identities explicitly (millions of them for this family) and solve them as one linear system. How many to generate is the choice the algorithm leaves open, and memory and solve time follow it; `seedling` learns to make that choice sector by sector. The tadpole's identity had one direction to move in; the bubble's identities move in several, and the reduction has to be steered.

2.1 An identity at work: the one-loop bubble

The one-loop massless bubble shows the whole mechanism. The family is

$$I(a_1,a_2) \;=\; \int \frac{d^d\ell}{i\pi^{d/2}}\,\frac{1}{(\ell^2)^{a_1}\,\big((\ell+p)^2\big)^{a_2}}\,, \qquad s\equiv p^2\,, \tag{3}$$

one integral for each pair of integer indices. In dimensional regularization the integral of a total derivative vanishes (there is no boundary term), so each seed (a_1, a_2) and each choice of vector in the numerator (the loop momentum ℓ^μ itself, or the external p^μ) yields an identity. Take ℓ^μ:

$$0 \;=\; \int \frac{d^d\ell}{i\pi^{d/2}}\,\frac{\partial}{\partial \ell^\mu}\left[\frac{\ell^\mu}{(\ell^2)^{a_1}\left((\ell+p)^2\right)^{a_2}}\right]. \tag{4}$$

Carrying out the derivative gives d from $\partial_\mu \ell^\mu$; the first propagator contributes $-2a_1\,\ell^2/\ell^2 = -2a_1$; the second contributes dot products, which the rewriting $2\,\ell\cdot(\ell+p) = \ell^2 + (\ell+p)^2 - s$ turns back into propagators. Collecting terms,

$$(d - 2a_1 - a_2)\,I(a_1, a_2) \;=\; a_2\,I(a_1 - 1, a_2 + 1) \;-\; a_2\,s\,I(a_1, a_2 + 1)\,. \tag{5}$$

Every term is again a member of the family, with indices shifted by one.

Now use it. Seeded at the corner $(a_1, a_2) = (1, 1)$ the identity reads $(d-3)\,I(1,1) = I(0,2) - s\,I(1,2)$. The integral $I(0,2)$ has lost its first propagator; shifting ℓ removes p entirely, leaving an integral with no scale, which vanishes in dimensional regularization. The identity then gives the dotted integral (the one with a squared propagator) in terms of the corner one:

$$I(1,2) \;=\; -\,\frac{d-3}{s}\,I(1,1)\,. \tag{6}$$

One seeded identity has lowered a propagator power and reached the master. Repeating the move at seeds $(1, a_2)$, and using the symmetry $I(a_1, a_2) = I(a_2, a_1)$ from the shift $\ell \to -\ell - p$, reduces every member with both indices positive to $I(1,1)$; members with a nonpositive index are scaleless and vanish. The whole two-parameter family collapses onto a single master integral. The differential-equation step fits in one line here as well: acting with $p^\mu\partial/\partial p^\mu = 2s\,\partial/\partial s$ under the integral sign and reducing the result with Eq. (6) gives

$$\frac{\partial}{\partial s}\,I(1,1) \;=\; \frac{d-4}{2s}\,I(1,1)\,, \tag{7}$$

so $I(1,1) \propto (-s)^{\,d/2-2}$, with the constant fixed by one direct integration. At one loop the entire route (identity, reduction, differential equation) fits on a page. At three loops, with fifteen indices in place of two, the same moves have to be organized.

2.2 The Laporta algorithm

Laporta [3] turned the identities into an algorithm: fix an ordering on integrals, generate the identities on explicit seed integrals, and Gauss-eliminate. Concretely: order the integrals so that harder means more distinct propagators, then more *dots* (propagator powers above one), then higher numerator degree; a *sector* is the set of integrals sharing the same on/off pattern of propagators. Choose bounds per sector, on the rank (the total propagator power), the dot count, and the numerator degree, and instantiate every identity of the family at every integer point inside the bounds, exactly as Eq. (5) was instantiated at $(1,1)$. Each instantiation is one linear equation among family members, with coefficients rational in d and the invariants. Gauss-eliminate in the ordering, always solving each equation for its hardest integral. The integrals never eliminated are the masters; every target integral inherits an expression in them.

Seeded elimination made reduction automatic, and it left the defining choice with the user: the algorithm is defined relative to a seed set it does not specify. Ever since, the standard answer has been uniform (one cushion on the rank, dot, and numerator bounds, chosen by hand, applied everywhere), so the bound the hungriest sector needs is the bound every sector gets.

The public-solver era. AIR [7] was the first public Laporta implementation; FIRE [8] began the longest-running lineage, through FIRE 6.5's modular simplification backends [9] to FIRE 7 [10]; Reduze introduced C++ and then distributed reduction [11, 12]; LiteRed took the symbolic-rule alternative [13]; and Kira [14] introduced modular pre-filtering of redundant seeded equations: recognition, inside a production solver, that most of what uniform staging generates is waste. The finite-field revolution [17, 18] decoupled coefficient algebra from system solving, powering Kira 2 [15], FiniteFlow [19], and FireFly [20], with Ratracer adding replayable reduction traces [21]. Structural lines attacked the system itself: syzygy-based IBP without dots [22, 23], implemented at scale in NeatIBP [24, 25]; master counting [26]; intersection theory [27]; dedicated massless-propagator reduction in Forcer [28]; and block-triangular forms in Blade [29], which make each numerical probe nearly free while leaving the seed-set question open by the authors' own account. Kira 3 [16] is the current state of production seeding, with improved seed generation and equation selection reported at up to two orders of magnitude over Kira 2.3. The seeding machinery has grown much smarter over these twenty-five years; the size of the cushion is still chosen by hand.

2.3 Scaling, and what limits it

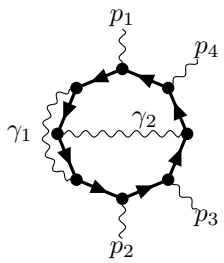
The staged system grows combinatorially. The bubble family has two indices, and reducing a given target takes a handful of seeds. A production family at three loops carries fifteen indices, and the seed count within given bounds is the number of integer points in a fifteen-dimensional slab: each unit of cushion multiplies it. Memory and solve time follow the staged size rather than the needed size, so the cushion sets both.

The margin question reached us as an out-of-memory kill. The computation was three-loop QED light-by-light: a single closed fermion loop of eight massive segments, four on-shell external photons, and two internal photons, one a self-energy arc on the loop, the other a chord from inside that arc to the opposite side (the three-loop family of Fig. 2). Reducing it means staging a fifteen-propagator family of 161 sectors, 1,725 targets, and 54 masters. Under the uniform recipe the staging was killed at 313 GB of resident memory; a retry under a 120 GB cap died at sector 45 of 161, still generating. A probe grid over the solver's seeding knobs, run inside the campaign, identified the fix: pin the seed lattice at the targets' measured support, with no cushion, and the same staging completes in 334 seconds with a peak of 7.42 GB. The pinned staging carried real physics within a day: the master integral that had been stalled at that wall was computed through it. A controlled rerun of the unpinned staging, capped and censored at 320 GB, is the legacy-monolith row of Section 5.2. The probe left the general question open: how far past the targets' support the lattice of any other sector, or any other family, must extend. Its winning pin was one point, found by hand, for one family. `seedling` generalizes that point: its staging is a fresh clone of the winning probe configuration, with the hand-chosen pin replaced by per-sector schedules rendered from the margins model of Section 3.1.

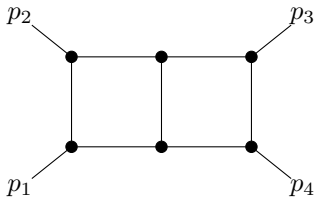
Fig. 2 draws the training family, the two deepest ladder rungs below it, and the four-loop family, each from its definition file; Section 4.2 walks the full rung list.

2.4 The sparsity of the needed system

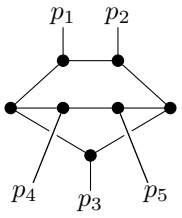
The wall of Section 2.3 came from the staging alone: the same family and the same targets completed in 7.42 GB once the seed lattice was pinned at the targets' measured support. A finished reduction uses a sparse subset of what a cushion stages, and the subset is not known



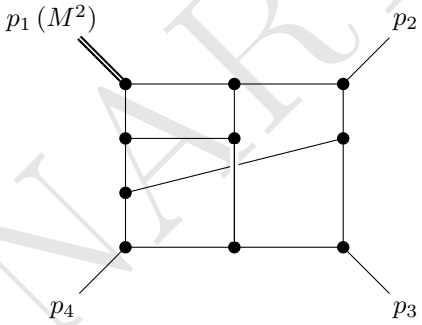
(a) `lbl3mopp_k2sh`: 3 loops, 4 legs, 15 propagators (8 massive, m^2=1+η, + 7 ISP) – training family (ladder rung L4; arms L4-A/L4-B).



(b) `dbox`: 2 loops, 4 legs, 7+2 propagators, all massless – reference (ladder rung L2, the deepest internal rung below the training family).



(c) `dpenta`: 2 loops, 5 legs, 8+3 indices, all massless, nonplanar – measured (ladder rung L3; external benchmark transcribed from arXiv 2606.10698).



(d) `xstrut`: 4 loops, 4 legs (p_1^2=M^2, others massless), 13+9 propagators, all internal lines massless, nonplanar – replicated (5/5 sectors vs recorded originals).

Figure 2: The four families behind the `seedling` results, each drawn from its family definition file. (a) is the family the margins model was trained on, the 1PI three-loop QED light-by-light topology of its parent computation: a single closed massive fermion loop (thick, m^2=1+η), four external on-shell photons, a photon self-energy arc γ_1 on one rung and a cross-chord γ_2 to the opposite rung. (b) and (c) are the deepest ladder rungs below it: the planar massless double box and the external nonplanar double pentagon benchmark. (d) is the four-loop family on which the tool's per-sector staging was replicated. Thin lines are massless, wavy lines are photons, the doubled leg is the massive external leg; a broken line passing under another is a crossing, not a vertex.

in advance: the Blade authors say so outright [29]. The uniform cushion is the fallback for that ignorance, and it overstages every sector but the hungriest.

Machine learning (2022–2026). Machine learning entered amplitude-adjacent symbolic computation first: polylogarithm simplification [30], squared-amplitude prediction [31], spinor-helicity simplification [32], transformer assistance for the amplitude bootstrap [33], and evaluation-side networks for contour deformation [34] and differential equations [35], with symbolic regression since aimed at singularity structure [36]. Inside the reduction step itself the literature is young (five papers, to our knowledge), and each has learned a different decision that Laporta left to the user: which identities to keep (LLM-guided genetic programming and explainable-AI opti-

mization evolving equation-selection heuristics [37, 38]); in what order to generate and eliminate (reinforcement learning and metaheuristics [39]); the elimination itself (a transformer classifier stepping through known identities with beam search, in place of the linear solve [40]); and, nearest to this work, where seeds live (tube seeding [41], in which ML experiments revealed that seeds can be confined to a thin geometric tube between targets and masters, collapsing a benchmark family's seed count from a degree-seven polynomial to a linear law). Selection, ordering, geometry, and even the elimination are now learned somewhere; the amount, the per-sector margin, is not: to our knowledge, no published work learns how much staging each sector of a family needs. `seedling` parameterizes that remaining decision; the five papers and this one parameterize different decisions of the same algorithm. Section 3 presents the model that predicts the amount, and Section 4 the instrument that tests the predictions.

3 The tool: `seedling`

`seedling` is two contributions in one package: a learned model that decides how much each sector stages, and the pipeline around it that makes the staged reductions trustworthy. This section presents both, and prints the model whole.

3.1 The ML innovation: registered per-sector margins

For each sector of an integral family, `seedling` predicts how far the seed lattice must extend beyond the targets' support (in rank, dots, and numerator degree) from features of the family and sector. The model (`M2`) is deliberately austere by the standards of the ML literature above, and deliberately strict by the standards of production solvers:

- *Frozen and registered.* The released model carries a registered hash (`da6642d2...`), its training labels, and its floor rule as provenance fields in every ledger; the build gates replay its registered verdicts bit-identically before any measurement runs.
- *Integer margins from real scores.* The model emits real raw scores; the pipeline takes ceilings, so a prediction of 4.047 stages at margin 5: staging errs on completion, and the architecture cannot emit a margin below 1 (the $s = 0$ case requires an explicitly logged override).
- *Feature-generic.* Two build gates cover this separately: generic family features reproduce the frozen per-class features exactly on the *training* family (all classes, all feature keys), and a frozen fit trained on the $t \in \{3, 4, 6, 7\}$ classes predicts the recorded verdict on the held-out $t = 5$ class. Together they allow the model to run on families it was not fitted to, the property the external-family cell of Section 5 tests.

The contrast with the recent ML-for-reduction literature is one of register rather than ambition: FunSearch-evolved heuristics and RL policies adapt online; `seedling`'s model is a fixed, hash-pinned artifact whose every prediction is reproducible and whose failure modes are gated. That choice gives up adaptivity and gains auditability: the correct trade for a tool whose outputs feed certified physics results.

3.2 The model in full

The registered model is small enough to print, so we print it, and small enough to draw, so Figure 3 draws it. Target sectors are staged at their own support and need no prediction; margins

are predicted for the census's closure sectors, grouped into classes by t, the number of propagators on. For each class `seedling` computes six features and emits

$$\mathrm{raw}(x) \;=\; \sum_{i=1}^{6} w_i \,\frac{x_i - \mu_i}{\sigma_i} \;+\; w_7\,, \qquad \mathrm{margin} \;=\; \max\bigl(1,\, \lceil \mathrm{raw}(x) - 10^{-9} \rceil\bigr)\,. \tag{8}$$

The weights are a ridge fit, solved in closed form, $(X^{\mathsf{T}}X + \lambda \mathbb{1})\, w = X^{\mathsf{T}}y$ at $\lambda = 1$ on z-scored features plus a bias column: no iterative optimizer, no random seed, deterministic from the labels up. The fit determines seven numbers, six weights and a bias; with the six feature means and scales they are the entire model:

feature x_i	weight w_i	μ_i	σ_i
`t`	0.02754437675578435	5.0	1.5811388300851899
`m`	−0.027544376755784385	10.0	1.5811388300851899
`n_sectors`	−0.07714142059669667	23.75	15.056144924914548
`mean_closure_dist`	−0.18700488412926722	1.619449583718779	0.4102037833110403
`mean_children`	0.00020319150692040005	3.6263297872340425	2.665890731090451
`mean_parents`	−0.027544376755784365	3.0	1.5811388300851899
bias w_7	4.2	–	–

The features, in the registered order: `t`; `m` $= n_{\mathrm{sp}} - t$, where n_{sp} is the family's index count; `n_sectors`, the number of closure sectors in the class; `mean_closure_dist`, the class mean of the minimum t-gap to a target sector containing the sector; `mean_children` and `mean_parents`, the mean child and parent counts in the nontrivial-sector poset. All six are computed from the family's sector census and target list before any identity is generated; the registered gate excluded posterior quantities outright, in those words. Family globals (loop, leg, index, propagator, and top-sector counts) are registered features but constant within a family and are not among the fitted keys.

The training set is four rows. The development family has 161 nontrivial sectors at $n_{\mathrm{sp}} = 15$, one label per closure class from a perturbation sweep: the minimal uniform closure margin m sufficient in the $(t{+}m, 1, m)$ staging, measured as {3:5, 4:5, 5:5, 6:5, 7:6}; the sweep certifies margin 5 sufficient for $t = 3$–6 (left-censored labels: the true minima may be smaller) and pins $t = 7$ at 6, margin 5 having been measured insufficient. With the $t = 5$ class held out, the fit sees classes $\{3, 4, 6, 7\}$ with labels $\{5, 5, 5, 6\}$. Four rows cannot determine seven numbers; the ridge penalty picks the solution, and the table shows how. On the training family $m = 15 - t$ and `mean_parents` $= 8 - t$, so after z-scoring those two columns equal minus the `t` column, and the solve splits one weight three ways: $|w_1| = |w_2| = |w_6| = 0.0275\ldots$, with identical σ. Because z-scored columns sum to zero, the bias decouples in the normal equations: $w_7 = \sum y/(n+\lambda) = 21/5 = 4.2$ exactly. The table is printed so this arithmetic can be checked by hand. The model may only add margin above the hard floors: the ceiling in Eq. (8) enforces margin ≥ 1, and the s floor is emitted separately by the chain rule ($s \geq 1$ whenever any target carries ISP powers or sits below a top sector, with the logged $s = 0$ override as the only exception). The constant arm the ladder compares against is defined from the same artifact: a single margin equal to the maximum training label, 6, under the same floors.

The order of operations was registered before the labels existed: the splits (hold out all $t = 5$ sectors of the training family; hold out an entire second family, with a third split and any re-rolling forbidden in advance), the feature list, the model ladder, and the kill-test criterion (lower total staged-equation count than the constant arm at equal first-pass completeness) were approved on

2026-07-04 and executed on 2026-07-06. The held-out class passed: the frozen fit predicts raw 4.047, ceiling 5, equal to the measured label, and the staged bulk on the held-out class came in at 2,494,800 equations against the constant arm's 4,573,800 (−45.5%) with completeness equal by construction; the evaluation schedule is byte-identical (md5) to the measured sufficiency run. On the second family, held out entirely and out of domain in m (its m runs 3–6 against the training family's 8–12), the frozen fit emits margins {3:4, 4:4, 5:5, 6:5} and stages −51.7% bulk at equal completeness, measured: 12,357 selected equations on both arms. The registered ladder's first entry, a constant table, coincides with the constant arm and fails the kill-test by construction. The registered hash is computed over exactly the printed table (weights, means, scales) plus the label hash, the feature keys, λ, and the held-out class, nothing else, and any model or feature change reopens the review gate. The parameter values printed above reproduce that hash on replay, along with the recorded raw scores to the last bit.

Nothing here is a neural network: the training gate excluded them in advance, the registered ladder topped out at ridge, and the gradient-boosted rung was skipped when its library was absent, under the plan's own fallback clause, so every margin the tool will ever emit is a function of the parameters printed above, the two floors, and six features a reader can compute from the sector census and target list.

3.3 The non-ML innovations: the seeding pipeline

The model would be uninteresting if its staging could not be trusted. The division of labor is fixed: `seedling` stages the system; Winnow (formerly IBP-Lapper) reduces it. The pipeline around the staging step is the second contribution (Figure 4):

- *Per-sector schedules* rendered deterministically from the margins (gated: all pinned schedules and a 16-cell grid regenerate byte-identically);
- *staged solver runs in fresh cgroups* with declared caps, so peak-memory figures are per-run and uncontaminated;
- *a two-slice oracle gate* across two primes against recorded reference corpora (or cross-arm master equality where no corpus exists), so no staging claim is made without the answers agreeing;
- *de-census escalation*: uncovered differential-equation rows are detected and escalate to a recorded action rather than silently narrowing the claim;
- *independently verifiable receipts*: retrofit λ-witness certificates checked by a separate verifier.

The build history is part of the record. The live end-to-end gate took four attempts: attempt 2 exposed a swallowed failure (a dependency error exited with return code 0 and a silent ledger), after which the tool was converted to fail-closed with regression tests; attempt 3 exposed a second fail-open and a gate-design error (the staged table was in fact correct, proven by the recorded oracle: 42 anchored / 126 held out / 0 mismatches), and attempt 4 ran the full chain green on the two-loop dispersive-daughter rung family of Section 4.2 (48 masters matching that family's recorded ordinal set, receipts certified, 28/28 package tests). The gate found and fixed two systemic defects before release, which is what it was added for.

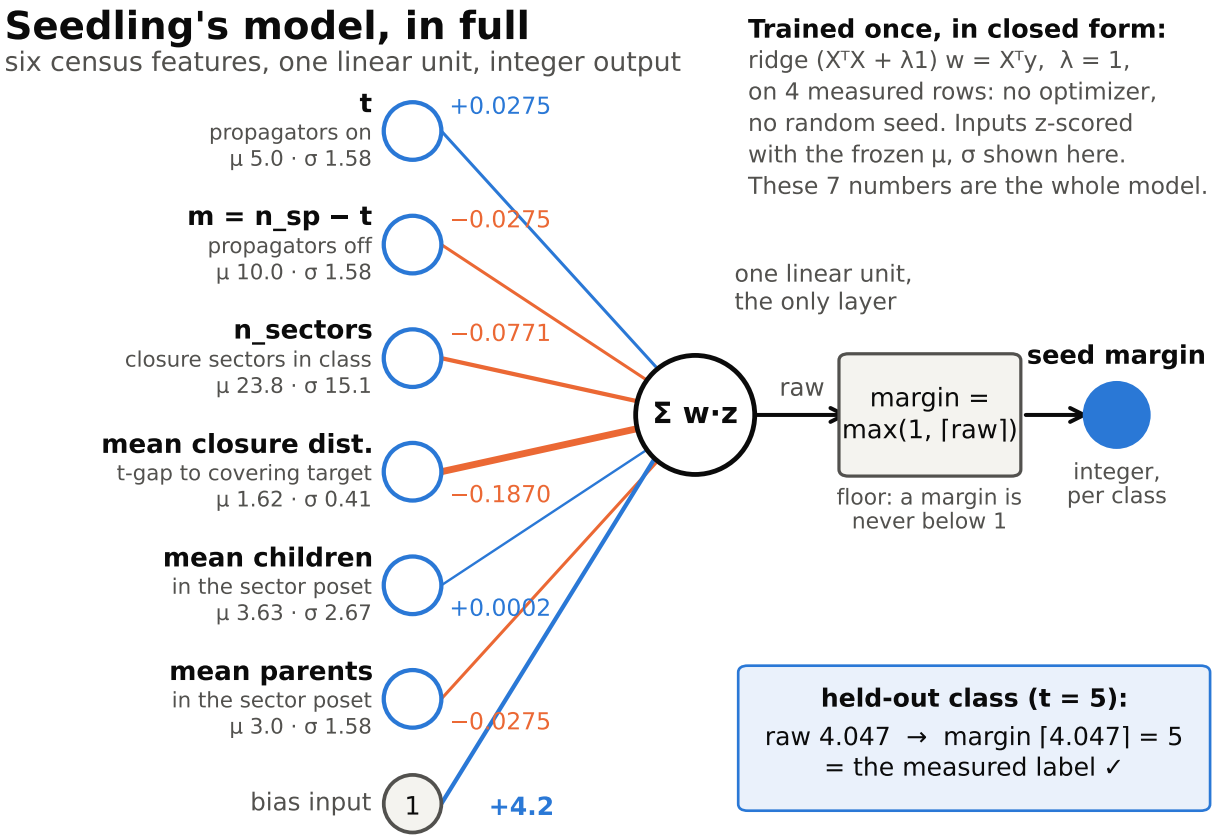


Figure 3: The entire `seedling` model, drawn as the network it is. Read this as: six census features, z-scored with the frozen means and scales, one linear unit carrying the six printed ridge weights and the exact bias $21/5 = 4.2$, then a ceiling with a hard floor at 1; no hidden layers, no neural network. Edge width is proportional to $|w_i|$, blue positive and orange negative. On the held-out class the unit emits raw 4.047, hence margin 5, equal to the measured label. All seven fitted numbers are on the figure.

4 The instrument: a three-arm scaling ladder

Claims about solver scaling are commonly supported by a handful of timings on hand-picked examples. The exponent ladder is the alternative, and we consider it a contribution of the same rank as the tool.

4.1 The four rules

The instrument is defined by rules fixed before the measurements:

- *Pre-registration first.* Arms, size proxies and censoring rules were committed before any rung ran; the run directory's mtime is the witness. Ladder trim rules entered by an appended addendum, recorded before application, with cells already run at that point kept unchanged.
- *Three arms, one variable at a time.* A (model margins) vs B (identical pipeline, constant margins) isolates the model; A vs C (the stock uniform recipe, gated on byte-identical

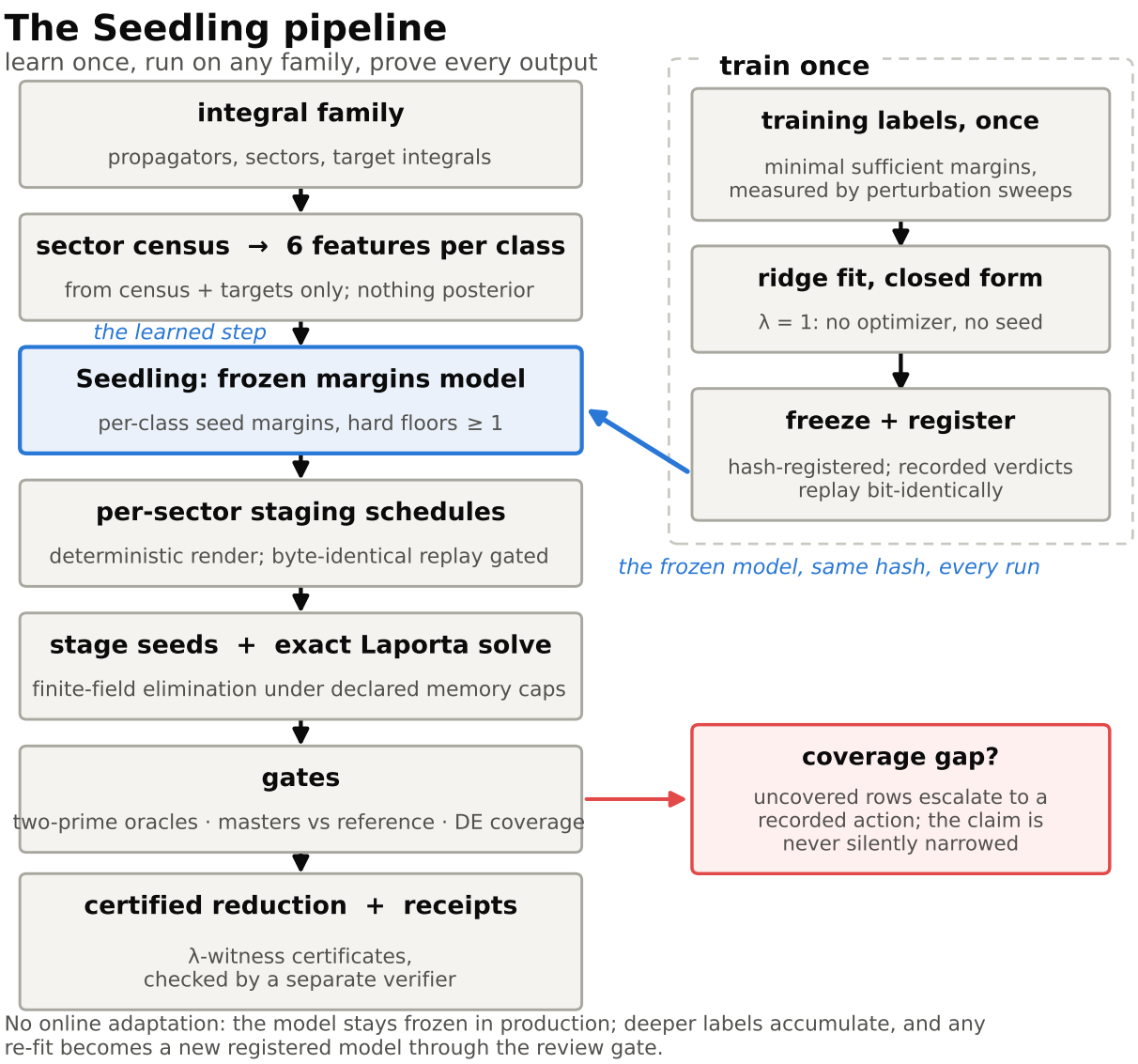


Figure 4: The `seedling` pipeline. Read this as: a family's sector census becomes six features per class before any identity is generated; the frozen, hash-registered model turns them into per-sector staging schedules; exact finite-field Laporta elimination consumes them under declared caps; and nothing leaves without gates: two-prime oracle slices, masters against reference corpora, differential-equation coverage with escalation to a recorded action, and independently verified receipts. Training happened once (closed-form ridge on measured labels) and the model runs frozen: no online adaptation, and any re-fit is a new registered model through the review gate.

regeneration) measures against practice.

- *Leave-one-out on everything.* Every fitted exponent is quoted with its paired-deletion behavior; a headline is only a headline if it survives all deletions.
- *Censoring as outcome.* Capped or exhausted cells enter as lower-bound arrows, never dropped, never fitted, and never silently converted into a smaller claim.

The complete ledger holds 52 records across a depth ladder (six rung families) and a rank ladder (the external benchmark), with every cell measured, censored-with-evidence, or not-run-by-design.

4.2 The rung families

The rung families keep the ancestry of Section 2.3; Fig. 2 draws them from their definition files. Two rungs of the depth ladder are two-loop dispersive daughters of the same light-by-light family, the dispersive mass sitting on the internal-photon chord in one and on the fermion line in the other, with the parent calculation's recorded reduction tables as their gate oracles; the deepest rung is the three-loop family itself, at the campaign's fixed rational kinematic point, with the mass-shift parameter as its only symbolic invariant. The remaining rungs supply breadth: a one-loop box constructed for the ladder (synthetic), the standard massless planar double box, and the nonplanar massless double pentagon of the tube-seeding benchmark suite [41], transcribed from the authors' code and identity-gated after the tool was already built: a validation family, with no part in the design. The four-loop nonplanar family of Section 5.5 came later still: the first family disjoint from everything the tool's development had never touched.

5 Results

Five measurements follow: the headline growth comparison, the deepest rung point-wise, the comparison the ladder could not decide, the instrument's calibration on the strongest external method, and the four-loop replication. Every fitted exponent is quoted with its leave-one-out range, and every censored cell is reported as an outcome.

5.1 Model margins beat constant margins, at growth level

On staged-bulk growth against sector count (Figure 5), the model arm's fitted exponent is 2.630 (leave-one-out 2.358–4.201) against the constant arm's 2.932 (2.659–5.557), and the ordering is stable under all five paired rung deletions, the strongest fitted claim of the campaign, now including the deepest rung. The constant arm's exponent was corrected at source during this paper's own gate: the deep-rung record had recorded the staged bulk of a different schedule's configuration file (the mismatch documented by md5). We corrected the record to that cell's registered preflight value, kept the original in a visible field, and re-ran every affected fit, controls re-receipted first. The record-level check flagged it; the comparison table already contained the corrected value all along. The shallow-rung crossover is disclosed: stock stages fewer equations at the three shallow rungs; growth, not per-rung constants, is the claim under test. Arm independence is checked exactly: the arms' master sets are identical on every completed rung. The fitted answer-side exponents are near-common across arms on the current ladder: selected equations 3.501/3.269/3.172 (leave-one-out 3.278–6.691, 2.873–5.508, 2.669–8.834) and wall 1.895/1.693/1.617 (1.589–6.028, 1.368–2.753, 1.074–15.367), with the model arm's deep point steepening its own fits; the deletion ranges are wide enough that these fits locate no answer-side law yet, and we do not claim one. The `seedling` effect is confined to the staging counts.

5.2 The deepest rung, point-wise: no fitting required

A dedicated deep cell ran the full model-arm protocol on the deepest family (1,725 targets, fresh 16G/8c cgroup); Figure 6 and Table 1 carry the numbers. Model margins halve the constant-margin staging and sit 2.3× under the stock recipe at the deepest rung, completing where the

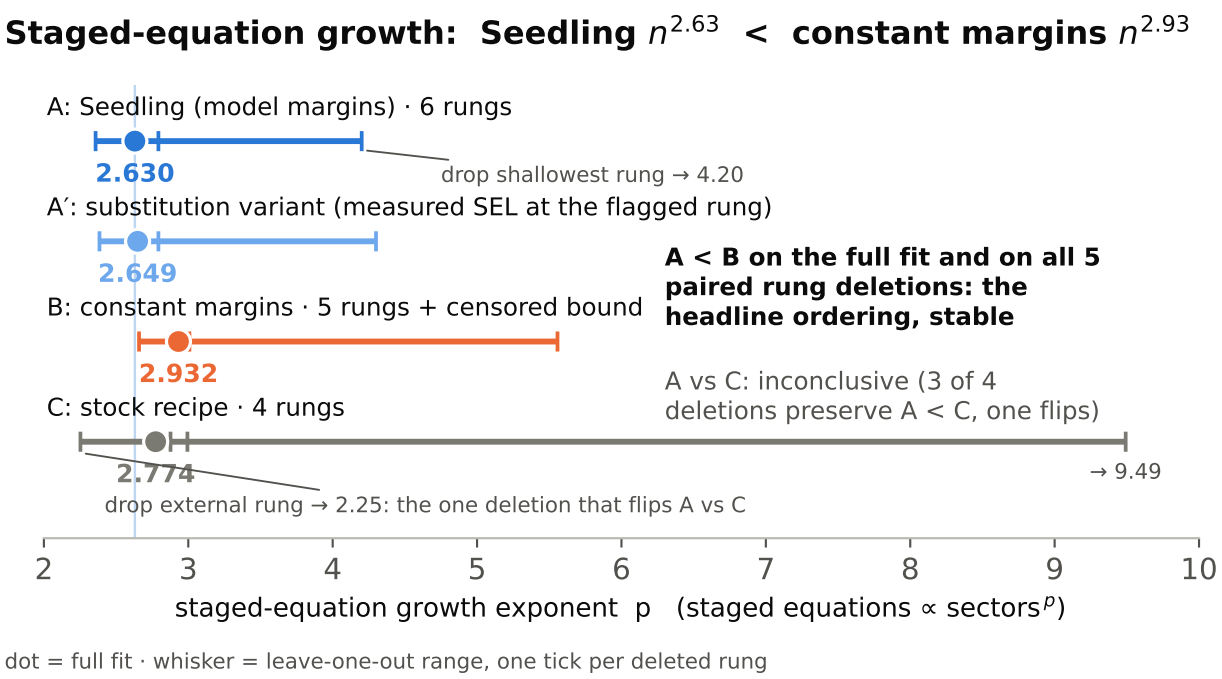


Figure 5: Staged-equation growth against sector count on the depth ladder, as fitted exponents. Read this as: `seedling`'s exponent 2.630 sits below constant margins' 2.932 on the full fit and on all five paired rung deletions (the headline ordering), while the `seedling`-vs-stock ordering is inconclusive, three of four deletions preserving it and the external-rung deletion flipping it. Dots are full fits; whiskers are leave-one-out ranges, one tick per deleted rung. The substitution variant (2.649) changes nothing, and the constant arm's censored rung enters as a bound, never fitted.

legacy monolith exhausted memory; the cell passed its two-slice gate, matched the recorded master count (54) and selected-equation wobble band, and recorded its de-census escalation. As censored lower-bound arrows against the stock arm's own shallow-rung fits, the legacy monolith's exhaustion point sits ≥88× above the wall fit and ≥129× above the memory fit; the arm-C recipe's bulk at this rung was computed deterministically from the closed form (14,258,808, identically the recorded global value), so the exhausted object is the larger legacy box and the bulk row correctly carries no arrow.

5.3 The undecided ordering, and the external-family exhibit

The model-vs-stock *growth* ordering remains inconclusive: the full fits order A 2.630 (leave-one-out 2.358–4.201) < C 2.774 (2.253–9.492), three of four paired deletions preserve the ordering, but dropping the external-family rung flips it (A 2.793 > C 2.253); the flip is localized to the one rung where stock is point-wise leaner. Fitted separation needs more deep rungs; the point-wise facts of Section 5.2 stand without fitting. On that external family (the double pentagon of the tube-seeding benchmark suite [41], transcription proven by an identity gate), constant-margin staging censored in generation at the 900 s cap while the model's margins completed with masters agreeing ($C = A = 108$); the constant arm's predicted bulk sat at 2.6× the model arm's (prediction-over-prediction, one cell, one family, one rung): an exhibit that the margins transfer off-corpus, not a law. On the demand axis, the deep point removes the full-fit sign flips that the shallow model-

(a) staged-bulk growth vs sector count

arm	exponent	leave-one-out	paired deletions	censoring
`seedling` (arm A, model margins)	2.630	2.358–4.201	A<B 5/5; A<C 3/4	fit uncensored (6 rungs); receipts only: L4 owner-censored at 6,527 s, L3 ×4
A, L3 bulk→SEL subst.	2.649	2.385–4.300	A<B 5/5	fit variant only; substitutes the measured L3 selected-equation count
constant margins (arm B)	2.932	2.659–5.557	comparator	L3 generation censored at 900 s (lower bound; 5 rungs fitted)
stock recipe (arm C)	2.774	2.253–9.492	comparator; drop-L3 flips	4 rungs fitted; L4 bulk deterministic (closed form = global pin), no solver run

(b) deepest rung (L4, 1,725 targets), point-wise: no fitting

arm	staged bulk	SEL	wall	peak memory
`seedling` (arm A, model margins)	6.22M	420,184	258 s	4.79 GB
constant margins (arm B)	12.14M	397,813	232 s	7.41 GB
stock recipe (arm C)	14.26M	–	–	–
legacy monolith (recorded)	–	–	censored ≥ 3,662 s	censored ≥ 320 GB (OOM at sector 72/161)

Table 1: Depth-ladder results at rev 2.4. Model margins beat constant margins at growth level (staged-bulk exponent 2.630 vs 2.932 against sector count, stable under all five paired rung deletions), and at the deepest rung the model halves the constant-margin staging and sits 2.3× under the stock recipe, completing in 4.79 GB where the legacy monolith was OOM-killed above a censored 320 GB bound; the model-vs-stock *growth* ordering is inconclusive (three of four deletions preserve it; dropping the external-family rung L3 flips it). Read each row as: fitted staged-bulk-vs-sector-count exponent with its leave-one-out range; paired-deletion verdict; deepest-rung point values, no fitting (SEL = selected equations); and the arm's censoring events (censored cells enter fits as lower bounds, per the censoring rule). The substitution row refits arm A with the measured selected-equation count at the one rung where its bulk predictor under-calls; the ordering against the corrected constant-arm record holds on all five paired deletions and no conclusion changes. The legacy monolith's arrows sit ≥88× above the stock arm's own wall fit and ≥129× above its memory fit.

and constant-arm fits had shown: the model arm's selected equations scale near-linearly with the demand cone: SEL $\sim$ demand$^{1.196}$ (leave-one-out 0.633–1.259), wall $\sim$ demand$^{0.625}$ (0.383–0.677), peak $\sim$ demand$^{0.113}$ (-0.145–0.405, sign not pinned, and carrying the watermark caveat of Section 6: the demand-axis peak fit includes the R1–R3 contaminated cells, and the machine-readable caveat is attached to any use of it), the output-sensitivity signature, quoted with its ranges; the stock arm's demand-axis fits still flip on its own rung set.

5.4 Calibrating the instrument on the strongest neighbor

Before trusting the instrument against our own tool, we pointed it at the strongest recent external method on that method's own benchmark: the rank ladder runs the tube-seeding paper's single-

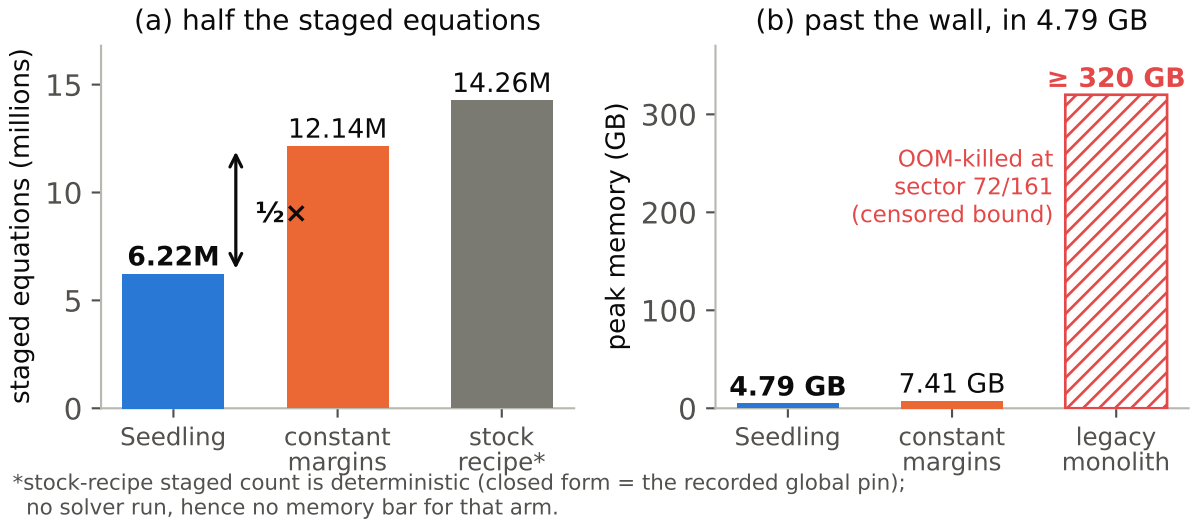


Figure 6: The deepest family (161 sectors, 1,725 targets), point-wise, no fitting. Read this as: `seedling` stages 6.22M equations (half the constant-margin 12.14M and 2.3× under the stock recipe's deterministic 14.26M) and completes in 4.79 GB of peak memory where the legacy staging was OOM-killed at sector 72/161 inside a 320 GB cgroup. The hatched bar is a censored lower bound, not a measurement; the stock-recipe count is a closed form, so it has no memory bar.

ISP-power family [41] on the paper's own public code, on our hardware (Figure 7). The instrument reproduces both published scaling laws. On the conventional baseline, the seed counts match the published closed form *exactly* at all six uncensored ranks (difference zero at every point: a reproduction, not a fit), and the solve-time exponent fits at $T_{\text{solve}} \sim n^{6.91}$ (leave-one-rung-out 6.72–7.27), the blow-up the $\sim n^7$ seed law drives. The linear tube law is reproduced in solve time ($3.70n - 18.3$ s against the published $2.51n - 17.5$ s: a uniform hardware factor of 1.77, range 1.44–1.92, across 11 anchors); seeds, equations, and masters are exact at all five tube points, and the multi-prime oracles pass 16/16 on both legs.

5.5 Replicating independent reductions at four loops

The first application of `seedling` beyond its development corpus was to a nonplanar four-loop $2 \to 2$ family with one off-shell leg (13 propagators, 9 irreducible numerators): five sectors with $t = 8$ to 11 propagators, each with an independent reduction on record from before the test. The staged reductions reproduce all five exactly (master lists byte-identical to the recorded ones, selected-equation counts equal, rank checks passing against the recorded counts), and a differential two-slice value check, fresh masters translated through the recorded tables and evaluated at two primes independently on 133–143 rows per sector, passes 5/5, with planted-coefficient and mutation controls caught 5/5. Staging cost sat in the same class as the recorded originals (129–210 s per sector against a recorded 308–711 s), but the two measurements ran under different load and core configurations, and the configuration effect measured on this family (2–5×) covers the whole gap: we report cost parity with certification added and claim no speedup. The five sectors are the deepest in the family with independent reductions available for comparison; the family's

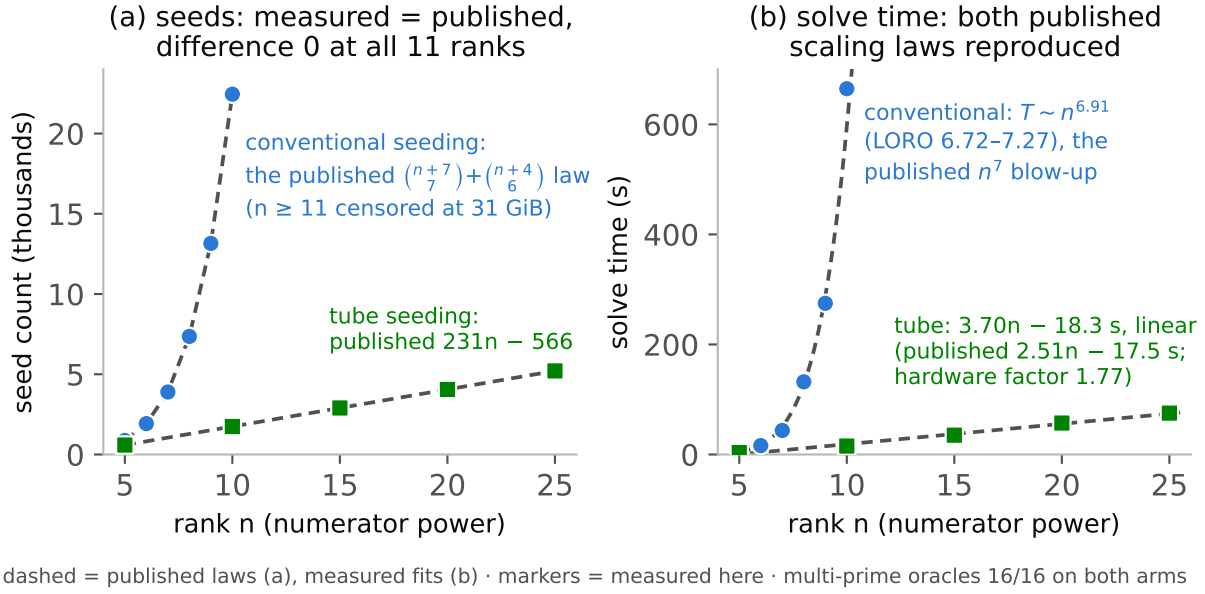


Figure 7: Calibrating the instrument on the strongest external neighbor: the tube-seeding paper's own family, run on its published code on our hardware. Read this as: measured seed counts land on both published closed forms with difference zero at all 11 measured ranks, and solve time reproduces both published laws, the conventional arm's $n^{6.91}$ blow-up (leave-one-rank-out 6.72–7.27, tracking the published $\sim n^7$ seed law; $n \geq 11$ censored at 31 GiB) and the tube arm's linear $3.70n - 18.3$ s against the published $2.51n - 17.5$ s, the slope gap covered by the uniform measured hardware factor 1.77. Multi-prime oracles pass 16/16 on both arms.

quantity	published	measured (this instrument)	outcome
baseline seed count	$\binom{n+7}{7} + \binom{n+4}{6}$	difference 0 at all six ranks $n = 5$–10	exact reproduction
baseline solve time	$\sim n^7$ (seed-law driven)	$T_{\text{solve}} \sim n^{6.91}$ (LORO 6.72–7.27)	law reproduced
tube seed count	$231n - 566$	difference 0 at all five points $n = 5$–25	exact reproduction
tube solve time	$2.51n - 17.5$ s	$3.70n - 18.3$ s	linear as published
hardware factor	–	1.77 (range 1.44–1.92, 11 anchors)	uniform; covers the slopes
multi-prime oracles	–	16/16 on both legs	PASS
censored / not run	baseline $n = 11, 12$ censored at 31 GiB; stock Kira arm censored at its first rung (>900 s); tube $n = 30, 35, 40$ not-run-by-design (deadline trim; published anchors cited)		

Table 2: Instrument calibration on the strongest external neighbor: the rank ladder runs the tube-seeding benchmark family on that paper's own public code, on our hardware, and reproduces both published scaling laws. Read this as a check of the instrument, not a result about `seedling`: seed counts match the published closed forms with difference zero at every measured rank (a reproduction, not a fit); the conventional baseline's solve time fits the blow-up the published $\sim n^7$ seed law drives, and the tube law is linear as published, with the slope difference covered by a uniform hardware factor of 1.77. Masters = 27 at every measured point on both arms. Censored and not-run-by-design cells are reported as outcomes, not dropped.

sector	t	targets	SEL `seedling` = recorded	masters	two-slice	rank check	wall: `seedling` / rec. (s)
`sec6763_k8`	8	135	2,775 = 2,775	2 = 2	PASS	2/2	210 / 420
`sec2495_k9`	9	140	2,781 = 2,781	3 = 3	PASS	3/3	138 / 436
`sec2942_k9`	9	140	3,144 = 3,144	4 = 4	PASS	4/4	142 / 711
`sec4059_k10`	10	143	2,426 = 2,426	2 = 2	PASS	2/2	129 / 308
`sec6135_k11`	11	144	3,119 = 3,119	1 = 1	PASS	1/1	132 / 427

Venue note: the recorded originals ran at 24-way farm concurrency (nice 5); the `seedling` runs at a dedicated 8-core cgroup (nice 19, box load ∼50/96). The measured venue/configuration effect on this family is 2–5× and covers the whole wall gap: same class, no speedup claimed.

Table 3: Four-loop replication: five sectors of a nonplanar four-loop $2 \to 2$ family with one off-shell leg (13 propagators, 9 irreducible numerators), each with an independent reduction on record from before the test. `seedling` reproduces all five exactly: a correctness result at four loops, not a speed claim. Read each row as: sector; propagator count t; selected-equation count, fresh vs recorded (equal 5/5); master count, fresh vs recorded, with master lists byte-identical (5/5); the two-slice value gate at two primes (133–143 rows per sector, 0 fails, 0 redraws); the rank check against the recorded counts; and wall time, `seedling` vs the recorded original; venues differ (see note), so the wall columns support cost parity only. Planted-coefficient and mutation controls were caught 5/5; the staging model is the frozen registered hash (`da6642d2...`). The family's expensive tail (recorded 2,700–13,400 s per sector) was not attempted, and nothing here extrapolates to it.

expensive tail (recorded costs 2,700–13,400 s per sector) was not attempted, and nothing here extrapolates to it. This test establishes correctness at four loops. The performance question stays open, and so does margin transfer, for a structural reason recorded in Section 7: on this family class the margin model had nothing to predict, and its schedule coincided with the constant-margin one to the byte.

6 Incidents and caveats

Five incidents and caveats belong in this paper rather than a changelog: each one is the measurement system doing its job. Four are operational; the fifth is scientific.

6.1 Four operational incidents

The watermark. A reused cgroup carried one run's memory high-water mark into the next, contaminating the memory records at the three shallow measured rungs; the peak-memory slopes there are flagged unreliable in a machine-readable caveat carried inside the fit artifact: pipelines read the JSON, so the caveat lives in the JSON. On the memory axis, the rung-local fresh-cgroup data and the deep-rung censored arrow survive.

The orphan backstop. Mid-campaign the result document itself was overwritten by an orphaned deadline-backstop script whose parent had been killed hours earlier: the orphan outlived its parent long enough to fire. The document was restored verbatim from a retained copy, and the incident (the sixth process-lifecycle incident of the program, in inverse flavor) is recorded with evidence in the launch log.

The schema patch. The deep cell's record required a schema patch before fitting; the patch is noted in the launch log and the fit controls were re-receipted (synthetic-truth exact recovery and mutation sensitivity) before any quoted fit.

The receipt walls. A witness pass at ∼419k rows takes a measured 55–65 minutes; the cell cap is 15 minutes (the overrun is recorded as an erratum, with the original projection preserved unedited). The deep cell's receipts were censored on that arithmetic. The four external-family (L3) receipt cells are censored on the same ground.

6.2 The scientific flag

The fifth flag is scientific rather than operational: at the external-family rung the model arm's bulk predictor under-calls its own selected-equation count. Substituting the measured count at that rung moves the six-rung growth exponent from 2.630 (leave-one-out 2.358–4.201) to 2.649 (2.385–4.300) and changes no ordering: the model-vs-constant ordering holds on all five paired deletions under substitution against the corrected constant-arm record, and the model-vs-stock deletion structure is unchanged; the five-rung effect recorded during the parent calculation (2.358 → 2.385) agrees. Predictor calibration at that family class remains an open item stated below.

7 Open items and outlook

What the measurements do not establish is part of the record. We close with the open items, then with where the axis leads.

7.1 Obstructions and open items

As recorded, without rounding up: the model-vs-stock growth ordering needs deeper rungs (the fitted comparison is open; the point-wise deep-rung facts stand); receipt coverage at ≥400k rows does not fit inside 15-minute cells (the receipt pipeline needs an emit-side rework before deep-rung receipts are routine); the bulk predictor needs recalibration at the external-family class; the memory axis at the shallow rungs cannot be re-fit from the published records for the model and stock arms (verifier-confirmed) and would need fresh-cgroup reruns; and universality beyond the measured families is simply not claimed. The four-loop application of Section 5.5 added three items. The preflight memory model under-predicts the measured peak by 2.33× on that family: at 22 indices the 2^{22}-sector trivial-sector scan and the solver's fixed overhead dominate the peak, flat across sectors and independent of staged bulk, a regime the three-loop calibration constant does not cover; the memory predictor needs a recalibration term before its caps are trusted at four loops. The per-sector margin model is structurally vacuous on cut single-top-sector configurations: below the top sector the census contains no closure sectors to predict margins for, the schedule reduces to target-support pinning, and the constant-margin plan is byte-identical; the four-loop test therefore carries no margin-transfer evidence in either direction. And the receipt evaluator reads coefficients in the dimension and one kinematic invariant, so families whose coefficients carry two invariants cannot get witness receipts until the evaluator is extended. Three items concern the model record itself: the training set is four rows against seven fitted numbers, with ridge regularization the only guard (the fit is as registered, but the identifiability question deserves its own study as deeper labels accumulate); the bit-identical replay gates do not pin the numerical environment they were run under; and the model emits point scores with no calibration record (uncertainty on a margin prediction is not yet a measured quantity). None of these blocks use of the tool at its measured envelope; all of them bound what this paper claims.

7.2 Outlook

The margins model is the smallest ML object that could occupy its niche: a frozen regressor over hand-legible features, auditable to the byte. The interesting question it opens is whether the per-sector margin axis composes with the other learned axes (tube geometry for where seeds live [41], evolved selection heuristics [37, 38], learned orderings [39]), since these parameterize independent decisions of the same algorithm. The instrument generalizes without modification: any of those methods can be placed on the same ladder, under the same pre-registration discipline, with the same leave-one-out and censoring rules, and we would welcome exactly that.

References

[1] F. V. Tkachov, "A theorem on analytical calculability of 4-loop renormalization group functions," Phys. Lett. B **100** (1981) 65–68.

[2] K. G. Chetyrkin and F. V. Tkachov, "Integration by parts: the algorithm to calculate β-functions in 4 loops," Nucl. Phys. B **192** (1981) 159–204.

[3] S. Laporta, "High-precision calculation of multi-loop Feynman integrals by difference equations," Int. J. Mod. Phys. A **15** (2000) 5087–5159, arXiv:hep-ph/0102033.

[4] A. V. Kotikov, "Differential equations method. New technique for massive Feynman diagram calculation," Phys. Lett. B **254** (1991) 158–164.

[5] T. Gehrmann and E. Remiddi, "Differential equations for two-loop four-point functions," Nucl. Phys. B **580** (2000) 485–518, arXiv:hep-ph/9912329.

[6] J. M. Henn, "Multiloop integrals in dimensional regularization made simple," Phys. Rev. Lett. **110** (2013) 251601, arXiv:1304.1806.

[7] C. Anastasiou and A. Lazopoulos, "Automatic integral reduction for higher order perturbative calculations," JHEP **07** (2004) 046, arXiv:hep-ph/0404258.

[8] A. V. Smirnov, "Algorithm FIRE – Feynman Integral REduction," JHEP **10** (2008) 107, arXiv:0807.3243.

[9] A. V. Smirnov and M. Zeng, "FIRE 6.5: Feynman integral reduction with new simplification library," Comput. Phys. Commun. **302** (2024) 109261, arXiv:2311.02370.

[10] A. V. Smirnov and M. Zeng, "FIRE 7: Automatic reduction with modular approach," Comput. Phys. Commun. **325** (2026) 110200, arXiv:2510.07150.

[11] C. Studerus, "Reduze – Feynman integral reduction in C++," Comput. Phys. Commun. **181** (2010) 1293–1300, arXiv:0912.2546.

[12] A. von Manteuffel and C. Studerus, "Reduze 2 – distributed Feynman integral reduction," arXiv:1201.4330.

[13] R. N. Lee, "Presenting LiteRed: a tool for the Loop InTEgrals REDuction," arXiv:1212.2685.

[14] P. Maierhöfer, J. Usovitsch, and P. Uwer, "Kira – a Feynman integral reduction program," Comput. Phys. Commun. **230** (2018) 99–112, arXiv:1705.05610.

[15] J. Klappert, F. Lange, P. Maierhöfer, and J. Usovitsch, "Integral reduction with Kira 2.0 and finite field methods," Comput. Phys. Commun. **266** (2021) 108024, arXiv:2008.06494.

[16] F. Lange, J. Usovitsch, and Z. Wu, "Kira 3: Integral reduction with efficient seeding and optimized equation selection," Comput. Phys. Commun. **322** (2026) 109999, arXiv:2505.20197.

[17] A. von Manteuffel and R. M. Schabinger, "A novel approach to integration by parts reduction," Phys. Lett. B **744** (2015) 101–104, arXiv:1406.4513.

[18] T. Peraro, "Scattering amplitudes over finite fields and multivariate functional reconstruction," JHEP **12** (2016) 030, arXiv:1608.01902.

[19] T. Peraro, "FiniteFlow: multivariate functional reconstruction using finite fields and dataflow graphs," JHEP **07** (2019) 031, arXiv:1905.08019.

[20] J. Klappert and F. Lange, "Reconstructing rational functions with FireFly," Comput. Phys. Commun. **247** (2020) 106951, arXiv:1904.00009.

[21] V. Magerya, "Rational Tracer: a Tool for Faster Rational Function Reconstruction," arXiv:2211.03572.

[22] J. Gluza, K. Kajda, and D. A. Kosower, "Towards a basis for planar two-loop integrals," Phys. Rev. D **83** (2011) 045012, arXiv:1009.0472.

[23] K. J. Larsen and Y. Zhang, "Integration-by-parts reductions from unitarity cuts and algebraic geometry," Phys. Rev. D **93** (2016) 041701, arXiv:1511.01071.

[24] Z. Wu, J. Böhm, R. Ma, H. Xu, and Y. Zhang, "NeatIBP 1.0, a package generating small-size integration-by-parts relations for Feynman integrals," Comput. Phys. Commun. **295** (2024) 108999, arXiv:2305.08783.

[25] Z. Wu, J. Böhm, R. Ma, J. Usovitsch, Y. Xu, and Y. Zhang, "Performing integration-by-parts reductions using NeatIBP 1.1 + Kira," Comput. Phys. Commun. **316** (2025) 109798, arXiv:2502.20778.

[26] A. Georgoudis, K. J. Larsen, and Y. Zhang, "Azurite: an algebraic geometry based package for finding bases of loop integrals," Comput. Phys. Commun. **221** (2017) 203–215, arXiv:1612.04252.

[27] P. Mastrolia and S. Mizera, "Feynman integrals and intersection theory," JHEP **02** (2019) 139, arXiv:1810.03818.

[28] B. Ruijl, T. Ueda, and J. A. M. Vermaseren, "Forcer, a Form program for the parametric reduction of four-loop massless propagator diagrams," Comput. Phys. Commun. **253** (2020) 107198, arXiv:1704.06650.

[29] X. Guan, X. Liu, Y.-Q. Ma, and W.-H. Wu, "Blade: a package for block-triangular form improved Feynman integrals decomposition," Comput. Phys. Commun. **310** (2025) 109538, arXiv:2405.14621.

[30] A. Dersy, M. D. Schwartz, and X. Zhang, "Simplifying polylogarithms with machine learning," Int. J. Data Sci. Math. Sci. **1** (2023) 135–179, arXiv:2206.04115.

[31] A. Alnuqaydan, S. Gleyzer, and H. Prosper, "SYMBA: symbolic computation of squared amplitudes in high energy physics with machine learning," Mach. Learn. Sci. Tech. **4** (2023) 015007, arXiv:2206.08901.

[32] C. Cheung, A. Dersy, and M. D. Schwartz, "Learning the simplicity of scattering amplitudes," SciPost Phys. **18** (2025) 040, arXiv:2408.04720.

[33] 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 $\mathcal{N} = 4$ super Yang–Mills theory," Mach. Learn. Sci. Tech. **5** (2024) 035073, arXiv:2405.06107.

[34] 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** (2022) 129, arXiv:2112.09145.

[35] F. Calisto, R. Moodie, and S. Zoia, "Learning Feynman integrals from differential equations with neural networks," JHEP **07** (2024) 124, arXiv:2312.02067.

[36] Y. Liu, Y. Xu, and Y. Zhang, "Uncovering singularities in Feynman integrals via machine learning," arXiv:2510.10099.

[37] M. von Hippel and M. Wilhelm, "Refining integration-by-parts reduction of Feynman integrals with machine learning," JHEP **05** (2025) 185, arXiv:2502.05121.

[38] Z.-Y. Song, T.-Z. Yang, Q.-H. Cao, M.-x. Luo, and H. X. Zhu, "Explainable AI-assisted optimization for Feynman integral reduction," JHEP **06** (2026) 225, arXiv:2502.09544.

[39] M. Zeng, "Reinforcement learning and metaheuristics for Feynman-integral reduction," Phys. Rev. D **112** (2025) 114051, arXiv:2504.16045.

[40] D. Shih, "Learning to unscramble Feynman loop integrals with SAILIR," arXiv:2604.05034.

[41] J. Berman, F. Charton, A. Luna, M. Wilhelm, and M. Zeng, "Efficient AI-inspired reduction of Feynman integrals via tube seeding," arXiv:2606.10698.