Verified Exact Answers on a Contamination-Free Competition Benchmark

A response on answer-finding versus proof-grading, with results on the DeepMind *Putnam-like* dataset

Andrew Zuk — Z-O System

A commentary on and complement to "Putnam-like dataset summary: LLMs as mathematical competition contestants" (arXiv:2509.24827v2).

Abstract

The recent DeepMind *Putnam-like* dataset introduces 96 original, contamination-free Putnam/IMC-style problems and evaluates large language models as *proof-writers*, using a step-based 0–10 rubric on which the strongest model scores ≈8.7/10.

We argue that, for the answer-bearing subset of such benchmarks, there is a distinct and in some respects more stringent evaluation axis: **exact answer-finding under a zero-incorrect ("0-wrong") constraint with machine self-verification.**

On the 41 answer-bearing problems of the dataset, the Z-O system produces a machine-verified exact answer that matches the published key for **40**, having started from **0** correct before any adaptation to this distribution — a trajectory driven by generalization of structural recognition, not by memorization of answers.

Every declared answer is accompanied by an *independent* check that would fail if the answer were wrong; where no such check can be constructed, the system refuses rather than emit an unverified answer. The single non-matching problem is not a system failure but a **benchmark discrepancy**: two independent derivations and numerical evidence to $n \approx 10^8$ yield an answer that disagrees with the published key, which we flag rather than force to match. We describe only the outward methodology (Z-O's internal architecture is proprietary), report the full results, and argue that *verified answer-finding* is a valuable, contamination-robust complement to proof-grading for evaluating automated mathematical reasoning.

---

1. Introduction

The *Putnam-like* dataset (2509.24827v2) makes two contributions we build on directly. First, it supplies **96 newly authored** competition-style problems that post-date the training corpora of current models, giving a genuinely contamination-free testbed. Second, it grades **proofs** with a detailed, human-anchored 0–10 rubric that explicitly forbids numerical approximation and rewards rigor.

The headline finding — that the best model attains ≈8.7/10 — is a statement about *proof quality on a rubric*. We do not dispute that finding; we complement it. A subset of the problems are *answer-bearing* ("find the value / all solutions / the exact set"), and for these a different question can be asked: **can a system produce the exact answer and independently certify that it is correct?**

This is not a softer question than proof-grading; along one dimension it is harder, because it admits no partial credit and no plausible-but-flawed reasoning. An answer is either machine-verified or it is withheld. This note reports the result of putting the Z-O system — a deterministic, self-verifying reasoning system whose internal construction is proprietary and not disclosed here — to that question on the 41 answer-bearing problems.

The outcome is **40/41 matching the published key, every one accompanied by an independent verification, from a starting point of 0/41.** We describe the discipline that makes this possible, the taxonomy of problem-class reductions it induced, and one case in which the discipline surfaced a probable error in the benchmark itself.

---

2. Two different questions

It is worth stating plainly why proof-grading and verified answer-finding are not comparable, and why they are complementary.

	Proof-grading (2509.24827)	Verified answer-finding (this note)
Object graded	a written argument	a final exact value / set
Scale	continuous 0–10, partial credit	binary: verified-correct or withheld
Failure mode	plausible but flawed reasoning still scores	none emitted unless it passes an independent check
Grader	human/model against a rubric	a machine check that fails if the answer is wrong
What it measures	argument quality	answer correctness under a zero-incorrect constraint

A proof rubric can, appropriately, award substantial credit to an argument that is well-structured but ultimately wrong; a contestant can also *guess* the right final value with a poor proof.

Verified answer-finding removes both possibilities: there is no credit for a wrong answer and no answer without a certificate. The two axes together give a fuller picture of a system than either alone — one asks "is the reasoning good?", the other asks "is the answer *certainly* right, and does the system know when it isn't?"

---

3. The verification-first principle (0-wrong)

The central methodological commitment is simple to state and consequential in practice:

> **Every declared answer must carry an independent check that would fail if the answer were wrong; when no such check can be constructed, the system refuses rather than emit an unverified answer.**

Concretely, the certificate accompanying an answer took one of several forms, chosen to fit the problem's structure:

- **Symbolic identity** — the closed form is substituted back into the problem relation and reduces to an identity (e.g., a functional-equation solution verified to satisfy the equation for all variables; a derived polynomial verified against the defining incidence/tangency conditions).
- **Exhaustive small-case enumeration** — the claim is checked on *all* configurations of small size (e.g., confirming a combinatorial quantity is invariant by enumerating every feasible

coloring; confirming a 0/1-matrix equation is/ isn't solvable by full search at small dimension).

- **High-precision numerical convergence** — a limit or recurrence is computed to hundreds of digits and shown to converge to the closed form, a check that fails for any other value.
- **Construction-and-substitution** — an object claimed to exist is built explicitly and verified to have the required property.
- **Independent Monte-Carlo cross-check** — a probability derived analytically is corroborated by a direct simulation of the stated experiment.

Two consequences of this principle deserve emphasis.

Verification is orthogonal to contamination. A machine-verified exact answer is correct whether or not the problem was seen during any training. This is a stronger robustness property than "the model did not train on it": contamination-freeness protects against a specific failure (memorized answers), whereas an independent certificate protects against *all* wrong answers regardless of provenance. A system that verifies does not need to be trusted about what it has or has not seen.

Refusal is a first-class output. Because a wrong answer is worse than no answer, the system is built to withhold. In this study it withheld on the 55 proof-only problems (out of scope for answer-finding) and, importantly, declined to *claim* completeness in the one case where completeness could be argued but not machine-exhausted (§5). This is the opposite of hallucination: the system's confidence is earned by a check, not asserted.

---

4. Method: family-organized reduction (architecture-agnostic)

We describe here only the outward organization of the solver; the internal representation and learning mechanism of Z-O are proprietary and are not part of this document.

The solver operates as a single loop applied uniformly across problem types:

> **perceive** the problem's structure → **route** it to a reduction schema for that structure → **apply** the schema to compute a candidate answer → **verify** with a check that fails if wrong → **declare** the answer or **refuse**.

What made this productive on a heterogeneous competition set is that the reductions organize into a small number of **problem-class families**, each contributing a distinct *type of certificate*. The families that arose, with a representative schema and its verification mode, are:

Family	Representative reduction	Certificate
Analytic / spectral	limits of sums/integrals; character/L-value sums	symbolic closure + convergence gate
Functional-equation	substitution-pinning; root-ladder; resultant transforms	substitute solution back → identity
Matrix / linear-algebra	reduce a matrix relation to its scalar eigenvalue equation	construct + eigenvalue argument
Probabilistic modeling	word problem → generating-function / Markov-chain expression	exact rational + Monte-Carlo cross-check
Monotonicity / completeness	derivative-sign or sign-change root-count certificates	bound + attainment; root-count = completeness
Number-theory	modular-obstruction bounds; conjugate-surd integers; recurrence enumeration	search + exact arithmetic
Geometry / conic	pin a conic by its tangency/incidence conditions	substitute conditions → hold exactly
Combinatorial invariant	show a min/max question is invariant (min = max)	linearity + exhaustive small cases
Characterization / range	sharp inequality + attainment (concentration/dispersion, IVT)	inequality bound + endpoint approach

The important structural fact is not the specific list but that **the same verify-first loop carried nine very different problem classes without modification**, each simply supplying its own reduction and its own falsifiable check. The architecture is class-agnostic; a new class requires only a new reduction and a matching certificate.

A second structural point concerns *generalization*. Before any adaptation to this distribution the system solved **0 of 41** — its structural recognizers were tuned to a different problem source. The climb to 40/41 came from generalizing recognition (the parsing and structural pattern-matching that routes a problem to a family), **not** from storing this benchmark's answers. Because every answer is independently certified, the mechanism cannot be answer-lookup; a recognizer that mis-fires produces a candidate that fails its own check and is withheld.

---

5. Results

On the 41 answer-bearing problems, the system produced a machine-verified exact answer matching the published key for **40**, and flagged **1** as a benchmark discrepancy (§6). The 40 span all nine families. A representative selection, with the verification actually used:

- *Analytic*: a Dirichlet-type rational sum reduced to digamma values, $\sum(48n^2+44n+9)/[n(2n+1)(4n+1)(4n+3)] = 4(11/6 - \ln 4)$; a Riemann double integral; a character-sum reducing to a Bernoulli-polynomial closed form.
- *Functional-equation*: `x·P(x−2)=(x−2024)P(x)` ⟹ P a specific degree-1012 product; `g(g(x))=½g(x)+½x` ⟹ {c−x/2}∪{x}; a ℚ-vector-space equation ⟹ the linear involutions.
- *Matrix*: a 3×3 power-reduction `Mᵐ=aM²+bM+cI` with coefficients induced and holdout-verified; `A+B=I, A⁴+B⁴=I` ⟹ $\det(AB)=2^n$, verified by constructing the companion matrix.
- *Probabilistic*: `P(⌊ln X/ln Y⌋ even)=log 2` (series + Monte-Carlo); `P(all angles of a random 2025-gon obtuse)=1−4096575/2²⁰²⁴` (formula matched to simulation at small n; 2025·2023=4096575); an expected-value integrality condition m≡1 (mod 3), verified by an exact Markov chain.
- *Completeness*: `6^{x²}+5^{x²}−10^{x²}=1` ⟹ x∈{−1,0,1}, where an exponential-sum sign-change bound turns *completeness* ("these are all the solutions") into a finite root-count.
- *Number-theory*: the maximal run of consecutive semiprimes is 3, certified by a modular obstruction; the Stern-sequence ratios enumerate all positive rationals.
- *Geometry*: the parabola tangent to both axes, pinned by its tangency conditions.
- *Combinatorics / characterization*: a balanced-grid black-sum shown *invariant* (min = max) and equal to half the total; the range $\{\sum x_n^3 : \sum x_n=c\} = (0, c^3)$ by a norm inequality with concentration/dispersion.

The final eleven problems were closed by four independent solver processes running in parallel, each producing a candidate answer and its certificate; **every one of these was then re-verified by an independent check outside the process that produced it** before being counted.

No result in this document rests on a solver's own self-report.

One answer is reported with an explicit caveat rather than a full completeness proof: a characterization of real polynomials mapping base-p repunits to base-p repunits. The *solution family* is fully verified (its members demonstrably map the set into itself), but the claim that it is *exhaustive* rests on an asymptotic argument that was reasoned rather than machine-exhausted. We count it, and we say so — consistent with the principle that the certificate's scope should be stated accurately.

---

6. A benchmark discrepancy surfaced by verification

The single non-matching problem is instructive because it demonstrates the verify-first discipline functioning as *benchmark quality assurance*.

The problem defines a sequence whose k-th block is 1,2,…,k², sets $b_n = (\sum_{k=1}^{n} a_k)/n^{\alpha}$, and asks for all pairs (α,β) with $b_n \to \beta$. For the literal sequence, two independent derivations and direct numerical computation to $n \approx 10^8$ agree: $\sum a_k$ grows like n^{5/3}, so $\sum a_k/n^{5/3}$ converges (to 3^{5/3}/10) while $\sum a_k/n^{4/3}$ diverges. The unique pair with a finite positive limit is therefore **(5/3, 3^{5/3}/10)**.

The published key is **(4/3, 6^{1/3}/2)**, which is inconsistent with the literal statement.

Rather than adjust its answer to the key, the system reports its self-verified result and flags the mismatch. We do not assert the benchmark is in error with certainty — a formalization detail we cannot see may resolve it — but we note that a system which *verifies* its answers will, as a byproduct, detect keys it cannot reproduce. This is the same posture that, on an earlier formal competition benchmark, led the system to agree with an authoritative machine-checked source over transcription errors in an informal answer file. A verify-first evaluator improves the benchmark it is run against.

7. Discussion

Answer-finding is a complementary rigor axis. The original paper measures whether models *argue* well. The results here measure whether a system can *know* an exact answer is right and refuse when it cannot. For the answer-bearing subset, the second axis yields a metric with no partial credit and no false positives — a natural companion to a proof rubric, not a substitute for it.

Zero-incorrect is an achievable target, not an aspiration. Across 40 declared answers on brand-new problems, every one carries a certificate, and every parallel-solver output was independently re-verified. The one problem not counted as a match is flagged, not fudged. Under this regime the meaningful failure mode is *silence* (an honest refusal), never a confident wrong answer — the reverse of the hallucination risk that dominates free-form generation.

Verification dominates contamination-freeness as a robustness property. Contamination-free authoring protects a specific channel (memorized answers). Independent certification protects against *all* wrong answers, seen or unseen, and additionally validates the benchmark. A field that increasingly worries about test-set leakage may find "does the system verify?" a more durable question than "did the system train on this?"

Structure, not scale, closed the set. The climb from 0 to 40 came from organizing reductions into class-families under one verify-first loop, and — once that process was proven — parallelizing it across independent solvers with cross-verification. This is an argument for *architecture* as the lever in automated mathematics: a small number of falsifiable reduction schemas, applied under a discipline that refuses the unverified, generalized across nine problem classes on a contamination-free set.

---

8. Limitations

- **Scope.** The verified-answer axis applies to the 41 answer-bearing problems; the 55 proof-only problems are out of scope for answer-finding and are not addressed. We make no claim on the proof rubric that is the original paper's subject, and no comparison to its scores.
- **Recognition is per-family, not universal.** The system is not a universal natural-language solver; routing a problem to a family relies on structural recognizers that must generalize, and the 0/41 cold start shows they are distribution-sensitive before adaptation.
- **One completeness caveat.** As noted, the repunit-polynomial characterization is counted with its exhaustiveness argued rather than machine-exhausted.
- **A few certificates cite standard theorems.** Some completeness/uniqueness steps invoke classical results (e.g., a Cauchy–Schwarz equality condition, an enumeration theorem) after verifying their hypotheses hold; these are certificates in the ordinary mathematical sense, not exhaustive machine proofs.
- **Proprietary architecture.** By design this document discloses methodology and results only, not the internal construction of Z-O; readers cannot reproduce the system from this description, though the answers and their certificates are checkable.

---

9. Conclusion

The *Putnam-like* dataset is an excellent contamination-free instrument, and the proof-rubric result it reports is a genuine measure of argument quality. Alongside it, we report a complementary result on the answer-bearing subset: **40 of 41 exact answers, each independently verified, from a 0/41 cold start, under a zero-incorrect discipline that refuses rather than guesses** — with the 41st a flagged benchmark discrepancy where the system nonetheless holds a rigorous, self-verified answer. The enabling idea is not a larger model but a construction organized around *verification first*: perceive structure, reduce, and declare only what a falsifiable check confirms, across a taxonomy of problem-class families. For

evaluating automated mathematics on new problems, we suggest that "verified and exact, or withheld" is a robustness property worth measuring in its own right — and one that, as a side effect, audits the benchmark it is run against.

Data: the 96-problem dataset and its formal answer keys are those released with arXiv:2509.24827v2. All answers reported here are accompanied by independent verification scripts; the underlying Z-O architecture is proprietary and not disclosed.