hexagon:2611.00200v1

Formalizing the interpolation lemma of the openai/math Landau–Siegel paper and substituting it into a Comparator-accepted proof

Joseph M. Reilly
Northeastern University

Abstract

OpenAI's openai/math release includes a Lean formalization of "Uniform exclusion of Landau–Siegel zeros" (OpenAI, October 1, 2026). Its Comparator challenge has two checked theorems, whose statements we prove equivalent in Lean. OpenAI proves them by two routes, neither through the paper's proof of Lemma 3. The catalogued main result, exists_absolute_real_zero_gap, gets its spanning step from a local Bézout argument and does not use OpenAI's Corollary 4 theorem. The second theorem, dirichletRealZeroBound_proof, proves Corollary 4 by a multiplicity estimate. We formalized the paper's proofs of Lemma 3 and Corollary 4 in Lean. By reading, the Lean statements match the paper's, and the declarations use only the axioms propext, Classical.choice and Quot.sound. A +2/-1 substitution puts the paper's Corollary 4 into dirichletRealZeroBound_proof. Comparator accepts the result with both kernels at openai/math fd4aeeb under a deny list of all 45 socket address families, and that theorem's closure no longer contains the three named OpenAI lemmas. Our replay of the corpus agrees with an independent replay on all 402 configurations where both have a verdict.

1 Introduction

OpenAI's openai/math repository publishes mathematical manuscripts together with Lean formalizations and Comparator challenges; we use it at commits adc7f12 and fd4aeeb [15, 16]. The manuscript "Uniform exclusion of Landau–Siegel zeros" (OpenAI, October 1, 2026) [18] states as its Theorem 1 that there is an absolute constant c > 0 such that every real zero beta in (0,1) of every primitive nonprincipal real Dirichlet L-function of conductor q >= 3 satisfies (1 - beta) log q >= c. Comparator's README [7] states that for each listed theorem it guarantees three things: the theorem proves the statement given in the challenge, uses no more axioms than permitted, and is accepted by the Lean kernel. Comparator can also pass the solution export to an external kernel; we used nanoda [1], at the official build 3a24072, as the second kernel.

Acceptance shows that a Lean proof of the Lean statement exists, using only the permitted axioms. It does not show that the Lean proof follows the paper's argument: in the case studied here, both accepted OpenAI proofs obtain the paper's spanning step by other means (Section 4). Bastounis, Circelli and Hansen [3] argue that a Lean check of an AI autoformalisation need not certify the natural-language proof, and show for OpenAI's Navier–Stokes formalisation that the Lean proof does not correspond to the natural-language proof. Their Section 1 states that they make no claims about the correctness of OpenAI's natural-language proof.

We find that the challenge has two checked theorems, with equivalent statements, and that OpenAI proves them by two routes. OpenAI's catalogued main result, exists_absolute_real_zero_gap, gets its spanning step from a local Bézout statement, through a zero test with

Corollary 4's bounds, and does not use OpenAI's Corollary 4 theorem. The second theorem, dirichletRealZeroBound_proof, proves Corollary 4 as actual_biquadratic_rectangle_span by a multiplicity estimate, uniform_rectangular_multiplicity. So neither route goes through the paper's proof of Lemma 3.

Our contributions are these.

- Statement faithfulness of the four family-003 challenges, by reading the Lean statements against the papers' texts, with Lean lemmas showing the SiegelZeros challenge is not vacuous (Section 3, with the reasoning behind each reading in Tables 2–9).
- Lean proofs of the paper's Lemma 3 and Corollary 4, with statements that match the paper's and only the standard axioms (Section 5).
- A reroute of dirichletRealZeroBound_proof through that Corollary 4, accepted by Comparator with both kernels (Section 6).
- A replay of the openai/math corpus as context, compared with an independent replay (Section 7).

We do not claim the following.

- Nothing about the paper's analytic arguments, nor about either OpenAI route beyond its spanning step.
- No specialist review of the mathematics: no specialist in analytic number theory or transcendence has reviewed it.
- Statement faithfulness rests on reading, by the author, with the reasoning recorded row by row.
- The reroute changes one of the two checked theorems; OpenAI's main result is untouched.
- The separate copy of this code in namespace OAI.Erdos970, loaded by 3 configurations, is outside our scope.
- For an explicit value of the constant we point to the note docs/effectivity/note-v2.pdf in the reviewer-packet-v3 archive [22] and to [2], and make no claim about it here.

Rerouting OpenAI's main result through the paper's Lemma 3 would require a bridge from Lemma3.rectangle to ActualCutoffSpanning; we leave this open.

The paper is organized as follows. Section 2 describes Comparator, nanoda, the challenge and the paper's lemmas. Section 3 reads the four challenge statements against the papers. Section 4 describes OpenAI's two formal routes. Section 5 gives the Lean proofs of Lemma 3 and Corollary 4. Section 6 reports the reroute. Section 7 places it in the corpus replay. Section 8 lists threats to validity, Section 9 the artifacts, and Section 10 the use of AI tools.

2 Background

2.1 Comparator and nanoda

Comparator's README at commit d03acab [7] states three guarantees for each listed theorem of a solution: that it proves the same statement as provided in the challenge, that it uses no more axioms than listed in permitted_axioms, and that it is accepted by the Lean kernel. The same README states that external kernels registered in external_kernels are given a file containing the solution export. According to Comparator's and lean4export's READMEs at the pinned commits, both kernel checks depend on the Lake build of the solution in a landrun sandbox, on the lean4export export of the solution, and on Comparator's

own statement and axiom checks. By the same READMEs, only the Lean kernel's check replays the solution environment inside the Comparator process, and only nanoda's check runs as a separate process given a file containing the solution export and a nanoda-style config.json. Comparator's README states the guarantees under six assumptions, among them that "The Lean kernel is correct (with 'external_kernels' this can be reduced to "At least one of the Lean kernel or the 'external_kernels' is correct")". nanoda's README at commit 3a24072 [1] states that nanoda_bin takes one argument, a JSON configuration file, and reads an export file created by lean4export.

All runs used Lean 4.34.1 (5045d005), Mathlib d13f23b, Comparator v4.34.0 d03acab and lean4export v4.34.0 076e8e5 [8, 10, 7, 9]. The full commits are d03acab154d269c06e60e4de7e4cc85deebff94b for Comparator and 076e8e57707e813375e8f9da8bf989799ace9680 for lean4export. The official nanoda is 3a24072; f67473b, 37cb87b and 37cb87b with nealmcb's patch at 64 MiB are unofficial builds, reported separately.

2.2 The challenge

The SiegelZeros challenge file has sha256 bedaac0248a641910f79a380d9702d82c6bfbbfd3760a69d27c72c5d8fb136ce, and it is unchanged between adc7f12 and fd4aeeb. Its config has Comparator check two theorems, OAI.SiegelZeros.WeightedTorusJets.exists_absolute_real_zero_gap and OAI.SiegelZeros.WeightedTorusJets.dirichletRealZeroBound_proof. Section 3 quotes both statements, and Section 4 gives the Lean proof that they are equivalent.

2.3 The paper's lemmas

The paper [18] states its Theorem 1 at refs/siegel-paper.tex:66-73, Lemma 2 at 181-194, Lemma 3 at 252-267, Corollary 4 at 410-420, Lemma 6 at 517-524 and Lemma 7 at 598-604. The paper numbers all of these with one counter, so the result between Corollary 4 and Lemma 6 is Remark 5. Lemma 2 gives a prime-sum bound for X >= 3 and a lower bound for the mass of primes with chi(p) = -1. Lemma 3 states that, for a surjective linear map A from C^4 to C^3 whose kernel is spanned by a vector with Q-linearly independent coordinates, evaluation at A n for n in E_N is surjective from P_t under the budget (3.1). The paper cites Fischler [4] as [Fischler, Theorem 1.1 and Lemma 1.3] at refs/siegel-paper.tex:155. Corollary 4 states that, under the hypotheses of Lemma 3, the monomial rows with alpha_j <= T_j span C^(E_N). Lemma 6 bounds the exponent sums S_1 and S_2, and Lemma 7 states the divisibility of Delta by p^(E_p) for every admissible prime p.

3 Statement faithfulness

We read four Comparator challenge statements of family 003 against the paper statements they encode: SiegelZeros, QuasiRiemannHypothesis, DirichletSevenEighths and HeckeSevenEighths. Each finding rests on reading the Lean statement against the paper's text, so each is INFERRED. Table 1 gives each challenge file's sha256 at openai/math fd4aeeb, the finding, and the record of the reading.

3.1 The SiegelZeros challenge

The SiegelZeros challenge file has sha256 bedaac0248a641910f79a380d9702d82c6bfbbfd3760a69d27c72c5d8fb136ce, and it is unchanged between adc7f12 and fd4aeeb. Listing 1 quotes its two theorems. By reading the Lean statement against the paper's text, the challenge statement matches the paper's Theorem 1, with the hypotheses primitive, nonprincipal, real, q >= 3 and beta in (0,1). The type of each solution theorem equals the type of the challenge theorem.

The paper states its Theorem 1 at refs/siegel-paper.tex:66-73. Its text, lines 67-72, reads:

Listing 1: The SiegelZeros challenge, lean/ComparatorChallenges/SiegelZeros.lean:8-14 and 19-25 (openai/math fd4aeeb; file sha256 bedaac02…, unchanged since adc7f12).

```
theorem exists_absolute_real_zero_gap :
    ∃ c : ℝ, 0 < c ∧
      ∀ (q : ℕ) [NeZero q], 3 ≤ q →
      ∀ χ : DirichletCharacter ℂ q,
        χ.IsPrimitive → χ ≠ 1 → (∀ a : ZMod q, (χ a).im = 0) →
        ∀ β : ℝ, 0 < β → β < 1 → χ.LFunction (β : ℂ) = 0 →
          c ≤ (1 - β) * Real.log (q : ℝ) := by
```

```
theorem WeightedTorusJets.dirichletRealZeroBound_proof :
    ∃ c : ℝ, 0 < c ∧ ∀ (q : ℕ) [NeZero q], 3 ≤ q →
      ∀ χ : DirichletCharacter ℂ q,
        χ.IsPrimitive → χ ≠ 1 → (∀ a : ZMod q, (χ a).im = 0) →
        ∀ β : ℝ,
          (0 < β ∧ β < 1 ∧ DirichletCharacter.LFunction χ (β : ℂ) = 0) →
            c ≤ (1 - β) * Real.log (q : ℝ) := by
```

> There is an absolute constant $c > 0$ such that every real zero $\beta \in (0,1)$ of every primitive nonprincipal real Dirichlet L-function of conductor $q \geq 3$ satisfies
>
> $$(1-\beta)\log q \geq c. \tag{1.1}$$

Four Lean lemmas show that the hypotheses of the challenge can be met and what the statement specialises to. NonVacuity.hypotheses_satisfiable proves that for every odd prime p the quadratic character mod p satisfies all hypotheses of the challenge ($q = p \geq 3$, primitive, nonprincipal, real). NonVacuity.quadChar_LFunction_eq_LSeries proves that the L-function of the quadratic character equals its Dirichlet series for $\mathrm{Re}\, s > 1$. NonVacuity.quadChar_LFunction_one_ne_zero proves $L(1, \chi_p) \neq 0$ for the quadratic character mod an odd prime p. NonVacuity.specialization proves that the challenge theorem specialises to an explicit statement about the quadratic characters. Each of the four depends on the axioms [propext, Classical.choice, Quot.sound]. [1]

3.2 The three 7/8 challenges

The other three challenges encode the main theorem of the 7/8 paper, "The Quasi-Riemann Hypothesis" [17] (`\date{30 September 2026}`, author OpenAI), whose paper.pdf has sha256 8fe93046f8cf5ef1ba5969c89addc02d76311adc4ee907509ff9cd96f7ec99e7 at both adc7f12 and fd4aeeb. Lines 107-110 of its build/paper.tex read:

> Every finite-order Hecke L-function over $F = \mathbb{Q}(\sqrt{-3})$ has no zero in $\Re s > 7/8$. The same holds for every Dirichlet L-function, including $\zeta(s)$. A pole at $s = 1$ for a principal character is allowed.

Listing 11 in Appendix A quotes the three challenge statements.

By reading each Lean statement against the paper's text, all three match the paper statement; the QuasiRiemannHypothesis finding carries the qualification "except s = 1, covered by Mathlib's junk value". The HeckeSevenEighths finding rests on a ruling about the L-function row and on one INFERRED identification. The Lean sub-claims of these readings (non-vacuity, series identities, holomorphy) come from a separate statement-check

[1] The lemmas are in lean-checks/NonVacuity.lean, LF sha256 b9fae4da5517a97da50d3fdb5356329f03f00b7ab935a448a999b2c59b24d9f9. They were compiled as checker via sudo -u checker, with no systemd-run unit and no address-family filter.

Table 1: The four family-003 challenge statements. File sha256 at openai/math fd4aeeb; finding and reading record as in the ledger. Every finding is INFERRED, by reading the Lean statement against the paper's text.

challenge	finding	reading
QuasiRiemannHypothesis	MATCH (except s = 1, covered by Mathlib's junk value)	Read by J. Reilly, 2026-10-27: MATCH
`065f8c9a01d28db78c8b1bfc5083b535082a2d2aac2230d563f98b8caa812cc5`		
DirichletSevenEighths	MATCH	Read by J. Reilly, 2026-10-27: MATCH
`8fb13ad977bca104591fcbc6ef01a58496e7385053c9b1445f95749afc0eff54`		
HeckeSevenEighths	MATCH (each nonzero ideal of O has exactly six generators, on which chi and the norm are constant)	Read by J. Reilly, 2026-10-27: agree
`cb5e404fa65a502c0a9006b9ad45c28914feb79e4b7b1ffcf05e4af82ed0cfc2`		
SiegelZeros	MATCH	Read by J. Reilly, 2026-10-27: MATCH
`bedaac0248a641910f79a380d9702d82c6bfbbfd3760a69d27c72c5d8fb136ce`		

build. All 34 `#print axioms` outputs of that build are exactly [propext, Classical.choice, Quot.sound]. That build ran as checker via sudo -u checker, with no systemd-run unit and no address-family filter. [2]

3.3 Reasoning behind the readings

Table 2 lists the reasoning points behind the SiegelZeros reading, one ledger row per point; Tables 7–9 in Appendix B list those for the three 7/8 challenges. Each point is INFERRED, by reading the Lean statement against the paper's text, and each is marked Read by J. Reilly, 2026-10-27: agree.

Table 2: Reasoning points behind the SiegelZeros reading. INFERRED; Read by J. Reilly, 2026-10-27: agree.

point	finding
quantifier order: the constant c precedes q, chi and beta, so c is absolute	MATCH
hypotheses: primitive, nonprincipal, real, q >= 3, beta in (0,1); none added, none missing	MATCH
conclusion: c <= (1 - beta) log q, non-strict as in (1.1)	MATCH
Mathlib meaning: IsPrimitive is conductor = level, and conductor is the least level through which chi factors	MATCH
Mathlib meaning: (chi a).im = 0 for all a means real-valued; values are 0 or roots of unity, so 0 or +-1	MATCH
Mathlib meaning: DirichletCharacter.LFunction is ZMod.LFunction, the continuation; it equals the Dirichlet series for Re s > 1 and is entire for chi != 1	MATCH
edge cases: q = 4 (odd) and q = 8 (one even, one odd) are covered	MATCH
edge cases: no hypothesis mentions parity, so both parities are included	MATCH
edge cases: nothing excludes non-prime or even conductors	MATCH

4 Two formal routes

Comparator checks two theorems for the SiegelZeros challenge: exists_absolute_real_zero_gap and dirichletRealZeroBound_proof. Their statements are not the same Lean proposition: the first takes 0 < beta, beta < 1 and the vanishing of the L-function as three successive implications, the second as one conjunction. ChallengeEquivalence.statements_equivalent, in a Mathlib-only file, proves the two statements equivalent, with the axioms [propext, Classical.choice, Quot.sound]. OAI's catalogue lean/formalization.yaml names exists_absolute_

[2] The four statement audits were also posted publicly on [5], at 2026-10-20T16:56:00Z.

Listing 2: The main route: OAI/NumberTheory/SiegelZeros/Conclusions/Theorem.lean:18-19 and Intersection/BezoutBoundary.lean:13-19 (openai/math fd4aeeb).

```
  exact SiegelZerosAwei.W50.uniform_exclusion_of_local_isolated_bezout
    (W22.localIsolatedBezout_of_actual_regular_selection ℂ)
```

```
theorem uniform_exclusion_of_local_isolated_bezout
    (hBezout : W22.LocalIsolatedBezoutStatement ℂ) :
    UniformSiegelZeroExclusion := by
  apply uniform_exclusion_of_actual_cutoff
  intro H N hH hHN _hN d a hd hd1 hd2 ha
  exact W28.actualCutoffSpanning_of_localBezout hBezout H N (by omega) hHN
    d hd hd1 hd2 a ha
```

real_zero_gap as the main result for the SiegelZeros config; neither it nor lean/docs/003.md names the paper's Theorem 1. OAI proves the two theorems by two routes, the main route and the second route below.

4.1 The main route

Listing 2 quotes the proof of exists_absolute_real_zero_gap and the theorem it applies. uniform_exclusion_of_local_isolated_bezout applies uniform_exclusion_of_actual_cutoff and obtains its spanning hypothesis from W28.actualCutoffSpanning_of_localBezout. uniform_exclusion_of_actual_cutoff argues by contradiction: it takes W63.exists_good_mass_for_actual_zeros_normalized, fixes H := 86713344, takes a contrary sequence of genuine zeros, and closes with false_of_eventual_exists_pivot_master and exists_actual_character_pivot_master.

The spanning hypothesis is W50.ActualCutoffSpanning (Listing 3). It states that, over L = rootField a sqrtTwo, the weighted theta rows of the box exponents span when restricted to the weighted cutoff W62.cutoff H (sourceCutoff H N). The cutoff is the set of indices with a.coords 0 + H a.coords 1 + H a.coords 2 at most sourceCutoff H N, the floor of 96 H^(2/3) N^(4/3). actualCutoffSpanning_of_localBezout proves it for 0 < H <= N and squarefree d != 1, 2 from W22.LocalIsolatedBezoutStatement C, by mapping to C with rowSpan_top_of_field_map and applying actual_complex_cutoff_spanning. That theorem combines cutoff_span_of_rectangularAssertion with actual_rectangularAssertion_of_localBezout. RectangularAssertion (Listing 4) is a zero test with Corollary 4's bounds, 32 H^(2/3) N^(4/3) and 32 H^(-1/3) N^(4/3). So the main route does not use actual_biquadratic_rectangle_span, but it proves a zero test with the same constants as Corollary 4, from the local Bezout statement.

At fd4aeeb the closure of exists_absolute_real_zero_gap has 107760 constants. It contains no constant from the 7/8 development, and neither dirichletRealZeroBound_proof nor actual_biquadratic_rectangle_span. It contains 9434 CategoryTheory and 3024 AlgebraicGeometry names.

4.2 The second route

dirichletRealZeroBound_proof is exact WeightedTorusJets.dirichlet_real_zero_bound, proved from source_real_zero_uniform_master_bound. This route reaches the paper's Corollary 4 as the theorem actual_biquadratic_rectangle_span, stated at OAI/NumberTheory/SiegelZeros/Structure/InvariantJetLinearMap.lean:471-484. The file is identical at adc7f12 and fd4aeeb, with LF sha256 09b8bd7fa82365e69caebfad466d039d85084595376766a731945fa5c11a2bdc. Listing 5 quotes the statement.

Listing 3: The main route's spanning statement, OAI/NumberTheory/SiegelZeros/Structure/ArithmeticCutoffTransport.lean:356-366 (openai/math fd4aeeb).

```
def ActualCutoffSpanning (H N : ℕ) (d : ℤ) (a : ℂ) (ha : a ^ 2 = (d : ℂ)) : Prop
↪  :=
  let L := SiegelZeros.W10.rootField a W60.sqrtTwo
  let f := algebraMap (𝒪 L) L
  let u := W37.integralRootA a W60.sqrtTwo d ha W60.sqrtTwo_sq
  let v := W37.integralRootB a W60.sqrtTwo d ha W60.sqrtTwo_sq
  let n : Fin (N ^ 4) → Fin 4 → ℤ := fun j i => (boxExponents N j i : ℤ)
  W29.rowSpan (K := L)
    (W60.weightedRow f (fun j => Awei.W39.theta u v (n j))
      (fun j => Awei.W39.theta (-u) v (n j))
      (fun j => Awei.W39.theta (-u) (-v) (n j)))
    (W62.cutoff H (sourceCutoff H N)) = ⊤
```

Listing 4: The zero test on the main route, OAI/NumberTheory/SiegelZeros/Determinants/RectangleDeterminant.lean:31-37 (openai/math fd4aeeb).

```
def RectangularAssertion (v : Fin 3 → Fin 4 → K) (H N : ℕ) : Prop :=
  ∀ p : MvPolynomial (Fin 4) K, InBox N p →
    (∀ a : Fin 3 → ℕ,
      (a 0 : ℝ) ≤ 32 * (H : ℝ) ^ (2 / 3 : ℝ) * (N : ℝ) ^ (4 / 3 : ℝ) →
      (a 1 : ℝ) ≤ 32 * (H : ℝ) ^ (-(1 / 3 : ℝ)) * (N : ℝ) ^ (4 / 3 : ℝ) →
      (a 2 : ℝ) ≤ 32 * (H : ℝ) ^ (-(1 / 3 : ℝ)) * (N : ℝ) ^ (4 / 3 : ℝ) →
      MvPolynomial.eval (fun _ => (1 : K)) (mixedInvariant v a p) = 0) → p = 0
```

Listing 5: OAI's Corollary 4, OAI/NumberTheory/SiegelZeros/Structure/InvariantJetLinearMap.lean:471-484 (LF sha256 09b8bd7f…, identical at adc7f12 and fd4aeeb).

```
theorem actual_biquadratic_rectangle_span
    {K κ : Type*} [Field K] [NumberField K] [Fintype κ]
    (a b : K) (v : Module.Basis (Fin 4) ℚ K)
    (hv : ∀ i, v i = ![1, a, b, a * b] i) (σ τ : K ≃ₐ[ℚ] K)
    (hσa : σ a = -a) (hσb : σ b = b) (hτa : τ a = a) (hτb : τ b = -b)
    (H N : ℕ) (hH : 0 < H) (hHN : H ≤ N)
    (n : κ → Fin 4 → ℕ) (hninj : Function.Injective n) (hn : ∀ j i, n j i < N) :
    let θ := fun j => ∑ i : Fin 4, (n j i : K) * ![1, a, b, a * b] i
    Submodule.span K
      ((fun α : Fin 3 → ℕ => fun j => θ j ^ α 0 * σ (θ j) ^ α 1 *
        (σ * τ) (θ j) ^ α 2) '' {α : Fin 3 → ℕ |
          (α 0 : ℝ) ≤ 32 * (H : ℝ) ^ (2 / 3 : ℝ) * (N : ℝ) ^ (4 / 3 : ℝ) ∧
          (α 1 : ℝ) ≤ 32 * (H : ℝ) ^ (-(1 / 3 : ℝ)) * (N : ℝ) ^ (4 / 3 : ℝ) ∧
          (α 2 : ℝ) ≤ 32 * (H : ℝ) ^ (-(1 / 3 : ℝ)) * (N : ℝ) ^ (4 / 3 : ℝ)}) = ⊤
          ↪  := by
```

Inside its proof, line 493 applies Geometry.source_rectangle_span_of_polynomial_zero_test and line 496 applies Geometry.uniform_rectangular_multiplicity; a text search of lines 485-511 finds no other line naming either lemma. So on this route OAI proves Corollary 4 by a multiplicity estimate, uniform_rectangular_multiplicity, and not by the paper's argument for Lemma 3. Within the SiegelZeros tree, actual_biquadratic_rectangle_span is used only at CharacterGlobalGreedyDeterminantMasterBounds.lean:138, by a text search of all OAI .lean sources. After line 138 is replaced, the closure of dirichletRealZeroBound_proof no longer contains actual_biquadratic_rectangle_span, so line 138 was its only use in that closure. The control scan of dirichletRealZeroBound_proof finds 116314 constants in its closure, among them actual_biquadratic_rectangle_span, uniform_rectangular_multiplicity and source_rectangle_span_of_polynomial_zero_test, at adc7f12 and again at fd4aeeb. exists_good_mass_for_actual_zeros_normalized, false_of_eventual_exists_pivot_master and exists_actual_character_pivot_master are in the closure of exists_absolute_real_zero_gap and in neither the control nor the rerouted closure of dirichletRealZeroBound_proof.

Three further configurations load modules from a separate copy of this code (namespace OAI.Erdos970), with its own multiplicity estimate. The reroute concerns the SiegelZeros chain only. The three are Jacobsthal, JacobsthalImproved and OstmannPrimes, each with a module under OAI.NumberTheory.Jacobsthal.Siegel in its solution import closure (3 of 416 configs); this is an import closure, not a declaration closure.

Rerouting removes 9936 (6901 + 3035) names in the CategoryTheory and AlgebraicGeometry namespaces from route 2, the closure of dirichletRealZeroBound_proof, counting the names reachable from the root through types and values, as #find_deps does. Counted by first name component in the closures, dirichletRealZeroBound_proof has 9454 and 3035 such names in the control and 2553 and 0 in the reroute; the difference, 6901 and 3035, equals that figure. [3]

4.3 The three spanning statements compared

Table 3 compares the main route's spanning statement with Lemma3.rectangle and with OAI's actual_biquadratic_rectangle_span, by reading the statements (INFERRED). By that reading, Lemma3.rectangle, applied with A the directions matrix and c = omegaCoefficients, plus the inclusion of the rectangle in the weighted cutoff and OAI's own rowSpan_top_of_field_map and boxEnumeration reindexing, would plausibly give ActualCutoffSpanning without the local Bezout statement. No such proof was written or run.

5 Lemma 3 and Corollary 4

We proved the paper's Lemma 3 and Corollary 4 in Lean, following the paper's argument, and bridged Corollary 4 to OAI's statement of it.

5.1 Statements

Lemma3.interpolation, the paper's Lemma 3, is stated at lean-checks/Lemma3.lean:816-825, and that file imports Mathlib only. The comparator copy lean-checks/comparator-copy/PaperLemma3/Lemma3.lean equals lean-checks/Lemma3.lean through "end Lemma3" and has sha256 1ce196f54fa98689bbea6bd74f2abdb110f5cbcca2eecddcca9316b8af465283. It is the file installed for the accepted rerouted Comparator run. Listing 6 quotes the statement from the comparator copy, and Listing 7 quotes the definition of Pt. The paper states its Lemma 3 at refs/siegel-paper.tex:252-267. Its lines 253-266 read:

[3] Counted over the full lean4export dumps, every declaration in the dump including unreferenced recursors and constructors, the removal is 10107 (7048 + 3059).

Table 3: The spanning statements, by reading the source (INFERRED).

aspect	comparison
rows	all three use products of the three conjugate linear forms of n with exponents a 0, a 1, a 2; after mapping to C the main route's rows are those of A = the directions matrix, whose kernel is spanned by omegaCoefficients
exponents	Lemma3.rectangle and actual_biquadratic_rectangle_span: the rectangle a 0 <= 32 H^(2/3) N^(4/3), a 1, a 2 <= 32 H^(-1/3) N^(4/3); the main route: the weighted cutoff a 0 + H a 1 + H a 2 <= floor(96 H^(2/3) N^(4/3)), which contains that rectangle
field	Lemma3.rectangle: C; actual_biquadratic_rectangle_span: the number field K; the main route: L = rootField a sqrtTwo, reached from C by rowSpan_top_of_field_ map
differences	index set (rectangle versus weighted cutoff), form (span versus the zero test RectangularAssertion), field (C versus L), indexing (Index H, Fin (N^4))

Listing 6: Lemma 3, lean-checks/comparator-copy/PaperLemma3/Lemma3.lean:817-826 (sha256 1ce196f5…).

```
theorem interpolation
    (A : (Fin 4 → ℂ) →ₗ[ℂ] (Fin 3 → ℂ)) (hA : Function.Surjective A)
    (c : Fin 4 → ℂ) (hker : LinearMap.ker A = Submodule.span ℂ {c})
    (hc : LinearIndependent ℚ c)
    (N : ℕ) (hN : 1 ≤ N) (t : Fin 3 → ℤ)
    (ht : ∀ j, 3 * ((N : ℤ) - 1) ≤ t j)
    (hbudget : (4 * (N : ℤ) - 3) ^ 4 < ∏ j, (t j - 3 * ((N : ℤ) - 1) + 1)) :
    Function.Surjective
      (fun B : Pt t => fun n : Fin 4 → Fin N =>
        eval (A (fun i => ((n i : ℕ) : ℂ))) (B : MvPolynomial (Fin 3) ℂ)) := by
```

> Let $A : \mathbb{C}^4 \to \mathbb{C}^3$ be a surjective linear map, with $\ker A = \mathbb{C}c$, where the four coordinates of c are linearly independent over $\mathbb{Q}$. Let $N \geq 1$ and t_1, t_2, t_3 be integers satisfying
>
> $$t_j \geq 3(N-1) \quad (1 \leq j \leq 3), \qquad \prod_{j=1}^{3} \bigl(t_j - 3(N-1) + 1\bigr) > (4N-3)^4. \tag{3.1}$$
>
> Set $E_N = \{0, \dots, N-1\}^4$ and
>
> $$\mathcal{P}_t = \{B \in \mathbb{C}[z_1, z_2, z_3] : \deg_{z_j} B \leq t_j \text{ for } 1 \leq j \leq 3\}.$$
>
> Then the evaluation map $B \mapsto (B(An))_{n \in E_N}$ from $\mathcal{P}_t$ to $\mathbb{C}^{E_N}$ is surjective.

By reading the Lean statement against the paper's text, Lemma3.interpolation matches the paper's Lemma 3, tex 252-267, clause by clause (Read by J. Reilly, 2026-10-27: MATCH).

Lemma3.rectangle, the paper's Corollary 4, is stated in the comparator copy Bridge.lean, installed for the accepted rerouted run (sha256 af401514808dde5bb913ce8e87cc5280b704a2 34ea324897d1a1ced1f218f250). Listing 8 quotes it. The paper states its Corollary 4 at refs/siegel-paper.tex:410-420. Its lines 411-419 read:

> Let A satisfy the hypotheses of Lemma 3. For integers $1 \leq H \leq N$, set
>
> $$U = N^{4/3}, \qquad T_1 = 32H^{2/3}U, \qquad T_2 = T_3 = 32H^{-1/3}U.$$

Listing 7: The box Pt, lean-checks/Lemma3.lean:42-43.

```
def Pt (t : Fin 3 → ℤ) : Set (MvPolynomial (Fin 3) ℂ) :=
  {B | ∀ j, (B.degreeOf j : ℤ) ≤ t j}
```

Listing 8: Corollary 4, lean-checks/comparator-copy/PaperLemma3/Bridge.lean:96-109 (sha256 af401514...).

```
/-- **Corollary 4** (`cor:rectangle`, tex lines 410-420). -/
theorem rectangle
    (A : (Fin 4 → ℂ) →ₗ[ℂ] (Fin 3 → ℂ)) (hA : Function.Surjective A)
    (c : Fin 4 → ℂ) (hker : LinearMap.ker A = Submodule.span ℂ {c})
    (hc : LinearIndependent ℚ c)
    (H N : ℕ) (hH : 1 ≤ H) (hHN : H ≤ N) :
    Submodule.span ℂ
      ((fun α : Fin 3 → ℕ => fun n : Fin 4 → Fin N =>
          A (fun i => ((n i : ℕ) : ℂ)) 0 ^ α 0 * A (fun i => ((n i : ℕ) : ℂ)) 1 ^
          ↪ α 1 *
            A (fun i => ((n i : ℕ) : ℂ)) 2 ^ α 2) ''
        {α : Fin 3 → ℕ |
          (α 0 : ℝ) ≤ 32 * (H : ℝ) ^ (2 / 3 : ℝ) * (N : ℝ) ^ (4 / 3 : ℝ) ∧
          (α 1 : ℝ) ≤ 32 * (H : ℝ) ^ (-(1 / 3 : ℝ)) * (N : ℝ) ^ (4 / 3 : ℝ) ∧
          (α 2 : ℝ) ≤ 32 * (H : ℝ) ^ (-(1 / 3 : ℝ)) * (N : ℝ) ^ (4 / 3 : ℝ)}) = ⊤
          ↪ := by
```

> The rows $((An)_1^{\alpha_1}(An)_2^{\alpha_2}(An)_3^{\alpha_3})_{n\in E_N}$, with $\alpha \in \mathbb{Z}^3_{\geq 0}$ and $\alpha_j \leq T_j$, span $\mathbb{C}^{E_N}$. The constant 32 is independent of A and H.

By reading the Lean statement against the paper's text, Lemma3.rectangle matches the paper's Corollary 4, tex 410-420 (Read by J. Reilly, 2026-10-27: MATCH). The copies of interpolation, rectangle, bridge_main, bridge_descent and via_lemma3 in the comparator copy and in lean-checks/Lemma3Bridged.lean are identical.

5.2 Structure of the proof

Table 4 lists the sub-lemmas of the Lean proof of Lemma 3 with their line ranges in the comparator copy, grouped under the paper's proof steps (tex 268-408). The main proof Lemma3.interpolation (816-985) calls each group in order at its "– Step" comments. Opening the proof, exists_annihilator and p5_PtSub give the weights of eq:moments, which is not one of the six named steps. Six helpers are used across several steps and are not placed in one group: zR 29-30, zC 32-33, emb 35-36, sqd 38-39, Pt 41-43, p5_eval_aeval 614-620.

5.3 The bridge to OAI's Corollary 4

Lemma3.actual_biquadratic_rectangle_span_via_lemma3 is stated in Bridge.lean, and by type_of% it has exactly the type of OAI's theorem (Listing 9). The embedding K -> C is chosen as IsAlgClosed.lift, not taken as a hypothesis (Bridge.lean:214). The kernel vector is OAI's biquadraticCoefficients, defined as `![1, a⁻¹, -b⁻¹, -(a * b)⁻¹]` at InvariantJetLinearMap.lean:321-322. ker A = span {c} is proved inline in bridge_main (Bridge.lean:231-254), which is stated at Bridge.lean:196-211. bridge_descent (Bridge.lean:147-155) does both the descent from C to K and the restriction to an injective family. In Mathlib d13f23b, AlgEquiv.mul_apply gives (e_1 * e_2) x = e_1 (e_2 x), so the third row of A is phi(sigma(tau(1, a, b, ab))). Listing 10 quotes its source line, Mathlib/Algebra/Algebra/Equiv.lean:663. The bridge reuses two OAI setup lemmas: source_same_witness_hyperplane (InvariantJetLinearMap.lean:447-455) and biquadraticCoefficients_linearIndependent (InvariantJetLinearMap.lean:380-397, with its _map form at 424-430).

Table 4: The Lean proof of Lemma 3 in lean-checks/comparator-copy/PaperLemma3/Lemma3.lean, by the paper's proof steps. Line ranges run from docstring to last line.

step	sub-lemmas	main proof
opening, eq:moments (not one of the six steps)	exists_annihilator 624-661, p5_PtSub 591-612	829-845
dimension count, zeros of R (eq:zeros)	four_hull_subset_cube 663-681, exists_vanishing 683-738	846-885
supporting face and nearest point	exists_lattice_nonvanishing 740-770, exists_nearest 87-148, p1_var_ineq 57-85, p1_coord_le_sqd 51-55, p1_dot_self_nonneg 47-49, quarter_mem_face 150-208	886-909
vertex choice (eq:vertex)	exists_heavy_vertex 224-299	910-916
off-face vanishing (eq:distance, eq:offface)	closer_point 301-340, p2_dot_expand 213-218, p2_dot_self_nonneg 220-222	917-939
Lagrange polynomial Q	exists_int_hyperplane 371-432, p3_g 342-352, p3_g_apply 354-354, p3_clear 356-369, int_comb_ne_zero 434-443, exists_lagrange 471-586, p4_linP 445-448, p4_eval_linP 450-455, p4_deg_linP 457-461, p4_rL 463-465, p4_rL_apply 467-469	940-947
final contradiction (B = Q R(z + A(m - u)))	exists_translate 772-812	948-985

Listing 9: The bridge theorem, lean-checks/comparator-copy/PaperLemma3/Bridge.lean:288-290.

```
/-- OAI's Corollary 4, proved via the paper's Lemma 3 instead of the multiplicity
↪  estimate. -/
theorem actual_biquadratic_rectangle_span_via_lemma3 :
    type_of% @OAI.SiegelZeros.WeightedTorusJets.actual_biquadratic_rectangle_span :=
    ↪  by
```

Listing 10: AlgEquiv.mul_apply, Mathlib d13f23b Mathlib/Algebra/Algebra/Equiv.lean:663, as copied in results/2026-10-26-claims-ledger/mathlib-algequiv-mul_apply.txt.

```
theorem mul_apply (e₁ e₂ : A₁ ≃ₐ[R] A₁) (x : A₁) : (e₁ * e₂) x = e₁ (e₂ x) :=
```

5.4 Checks

All nine declarations compiled from the comparator copy itself (PaperLemma3/Lemma3.lean and Bridge.lean) depend on the axioms [propext, Classical.choice, Quot.sound], at fd4aeeb under the 45-family deny list. The same axioms were recorded for interpolation, rectangle and via_lemma3 in Lemma3Bridged.lean at adc7f12 on 2026-10-08. [4] A standalone check of the export of via_lemma3 by nanoda 3a24072 accepted it, with 61810 declarations checked. The same export with Classical.choice not permitted was rejected (exit 1). The export with rectangle's proof replaced by interpolation's proof was rejected (exit 101). These nanoda runs ran as checker via sudo -u checker, with no systemd-run unit and no address-family filter.

An exact execution of the proof of Lemma 3 in Python Fraction arithmetic (src/lemma3_construct.py) ran 540 instances at N = 2, 3, with 28 checks each, and all pass. This covers the proof steps at N = 2, 3 only, not the lemma in general. [5]

6 The reroute

The reroute replaces OAI's Corollary 4 in the second route, dirichletRealZeroBound_proof, with the Lean proof of Section 5. It changes one tracked file of openai/math by +2/-1 lines, an import and the term at line 138, and adds the two modules PaperLemma3/Lemma3.lean and PaperLemma3/Bridge.lean. The installed files have sha256 1ce196f5... and af401514..., and the patch is 1 file +2/-1. The patch applies cleanly at fd4aeeb, and the patched file is byte-identical at adc7f12 and fd4aeeb.

6.1 The runs at fd4aeeb

The primary record is a pair of Comparator runs at openai/math fd4aeeb, in results/2026-10-27-reroute-fd4aeeb/: a control on the unmodified code and the reroute. Both ran the official Comparator with nanoda 3a24072 enabled, under the 45-family deny list and a 16G v1 cgroup cap. Both runs are accepted by Comparator with both kernels, with exit 0. Table 5 gives the verdicts, the dependency scans of dirichletRealZeroBound_proof, and the time and memory of each run. The rerouted closure of dirichletRealZeroBound_proof matches only via_lemma3 and interpolation, and none of the three OAI lemmas. Both scan blocks are identical to the 2026-10-09 blocks at adc7f12 (counts, matches, axioms). The reroute leaves exists_absolute_real_zero_gap unchanged: its closure has 107760 constants in both clones, with none of the three OAI lemmas. The guard made no stops during the two runs.

6.2 The clone

The runs used a local clone of fd4aeeb; package directory reused read-only from the original clone; OAI closure built from scratch. The clone /home/checker/math-r2 is a local git clone of /home/checker/math, at fd4aeeb with 0 tracked changes, and its lake-manifest.json, lean-toolchain and lakefile.lean equal the original's. Its lean/.lake/packages is bind-mounted read-only from the original clone, a write as checker was refused, and the mount was removed at the end. The clone was created at 2026-10-26 22:17:36, its lean/.lake/build directory was created at 22:19:37, after the control prebuild started (22:19:09), and the oldest OAI .olean

[4] A fresh run on 2026-10-26 at 21:43, of Lemma3Bridged.lean at fd4aeeb under the 45-family deny list, gave the same three axioms lines.

[5] Two working drafts in lean-checks/ are not cited: Lemma3_Cor4.lean proves interpolation by sorry (line 31), so its rectangle depends on sorryAx, and Lemma3_Bridge.lean proves rectangle by sorry (line 20). The canonical Bridge.lean imports neither.

Table 5: The SiegelZeros control and reroute at openai/math fd4aeeb (results/2026-10-27-reroute-fd4aeeb/). Prebuild: lake build of the solution and challenge modules, 4 threads. Comparator run: nanoda 3a24072 enabled, 16G v1 cgroup cap, 45-family deny list.

run	verdict	scan	prebuild	Comparator run
control	ACCEPT_ BOTH; exit 0	116314 constants in closure	11min 19.935s; 9243 jobs	842 s; max RSS 7707508 kB; cgroup max_usage 7412195328 B; oom_kill 0
reroute	ACCEPT_ BOTH; exit 0	92658 constants in closure	1min 39.395s; 9245 jobs	659 s; max RSS 7662352 kB; cgroup max_usage 6280855552 B; oom_kill 0

dates from 22:19:53. The reroute prebuild ran in the same clone, after the control, with the two PaperLemma3 modules and the +2/-1 patch. [6]

6.3 Corroboration at adc7f12

The same substitution was first run at adc7f12 on 2026-10-09. There the rerouted clone differs from openai/math in 1 tracked file, +2/-1, plus the two untracked PaperLemma3 modules. The challenge file is unchanged, with sha256 bedaac0248a641910f79a380d9702d82c6bfbbfd3760a69d27c72c5d8fb136ce. The rerouted Comparator run built 9244 jobs and was accepted by Comparator with both kernels, with exit 0. Its dependency scan of dirichletRealZeroBound_proof finds 92658 constants in the closure, matching only via_lemma3 and interpolation, against 116314 in the control. After line 138 is replaced, the closure no longer contains actual_biquadratic_rectangle_span.

The rerouted chain still imports the patched PrimeNumberTheoremAnd.SiegelZeros.HadamardSupport.

7 Corpus context

We replayed the Comparator configs of openai/math, the 189 referenced by formalization.yaml and the 227 unreferenced ones, with the official tool set, nanoda 3a24072. Of the 189 configs referenced by formalization.yaml, 187 were replayed at fd4aeeb and 2 (SiegelZeros, QuasiRiemannHypothesis) in the pilot at adc7f12. At fd4aeeb, both kernels accept 178 of the 187, and 9 are ACCEPT_LEAN_ONLY because nanoda 3a24072 crashes. The extended replay of the 227 unreferenced configs gives 212 results, with 15 not checked here. Table 6 gives the outcomes, with the official tool set and the unofficial nanoda builds in separate columns.

7.1 Agreement with an independent replay

Neal McBurnett's repository [12] states that Comparator, as shipped, accepts all 416 configs at fd4aeeb. Against all our recorded results, 416 configs were compared, 402 agree, 0 disagree, and 14 have no verdict here; none lacks a verdict there. So every config with a verdict on both sides agrees: 402 of 402. Our SiegelZeros record in this comparison is the fd4aeeb control run, and QuasiRiemannHypothesis is the adc7f12 pilot, whose config, challenge and solution import closure are byte-identical at fd4aeeb. Export sizes equal his for 146 extended-replay

[6] By the order of the runs and lake's incremental build, the reroute prebuild is shorter because lake reused the control's build outputs and rebuilt only the two new modules, the patched module and the modules that depend on it (INFERRED). The prebuild log lists only modules that printed warnings, so the set of rebuilt modules is not recorded.

Table 6: Corpus outcomes. Official: Comparator with nanoda 3a24072. Unofficial: other nanoda builds, reported separately. Independent replay: results STATED by nealmcb at 1ca7ff2 [12].

configs	official	unofficial	independent replay
189 referenced	187 + 2; 178 / 9 (both kernels / Lean only)	9 of 9 pass with f67473b	see the 416 row
227 unreferenced	212 / 15; ACCEPT_BOTH 205, LEAN_ACCEPT_ NANODA_CRASH 6, TIMEOUT 1, NOT_ CHECKED_HERE 15	see next rows	see the 416 row
Catalan, KMedianThreshold	NOT_CHECKED_HERE; Lean-kernel Comparator runs here: 369 s; exit 0 (KMedianThreshold), 2234 s; exit 0 (Catalan)	both kernels (unofficial nanoda)	10
ContinuumCoulombHardness	not completed on this machine (memory)	none	mixed; see footnote
all 416	outcome comparison: 416 compared; 402 agree; 0 disagree; 14 no verdict here		416/416 (Comparator as shipped); 401/416 (nanoda 3a24072 at 1 GiB stack) [13]
export sizes	146 (0 differ, 14 not comparable); 11 (13 incl. 2 from extended records)		

exports, with 0 differing and 14 not comparable, and to the byte for the 11 configs of the 2026-10-26 nanoda matrix. [7]

7.2 ContinuumCoulombHardness

ContinuumCoulombHardness could not be built here: the guard stopped the build at 4 and at 2 threads, and it is permanently deferred. [8]

7.3 nanoda failures

By interpreting the per-config outcomes across nanoda builds, the nanoda failures fall into three classes (INFERRED). Class 1 is the RoundTables group (8), the Laughlin group (3), KMedianThreshold and Catalan. Class 2 is KaplanskyFinitelyPresented, GroupRingDeterminant and KaplanskyDirectFiniteness (COMPUTED), with MatchingFPRAS, ContingencyTables, LogspaceEquality, RandomizedMeanPayoff and PeriodicGroup (STATED). Class 3 is SquareDifference [11]. nealmcb states that on nanoda master ten openai/math configs

[7] The 14 configs with no verdict here are ContinuumCoulombHardness, CycleCliqueRamsey, DefocusingNLS, FalconerAllDimensions, InvariantIsing, JointDickman, MUBSix, OrdinaryTwoPointCorrelations, OstmannComplete, PattersonFirstMoment, PolycyclicRecognition, SurfaceImmersion, UniversalTensorSquares (NOT_CHECKED_HERE) and PlanarPacking (TIMEOUT).

[8] All four records are nealmcb's at 1ca7ff2 (STATED) [12]. His Comparator as shipped accepts it, at 923.63 s. His kernel-pair sweep records lean_rc 1 (454.4 s) and nanoda_rc 1 (99.3 s) on his export. His nanoda retests give f67473b: 0 / 130.1 s / 24.23 GiB; 0 / 114.3 s / 24.26 GiB; 134 / 28.6 s; 134 / 28.8 s; and 998e0a8: 134 / 32.8 s. With the let fix and the Nat.succ guard at STACK_SIZE 64 MiB, nanoda 37cb87b passes it at 50.0 s and 1.8GB; with the let fix alone it overflows at 64 MiB (34 s, 4GB) and 256 MiB (48 s, 13GB).

fail, and that with the let fix, the Nat.succ guard and STACK_SIZE 64 MiB all ten check [14]. [9]

8 Threats to validity

Kernel trust. Comparator's README states its guarantees under six assumptions, among them that "The Lean kernel is correct (with 'external_kernels' this can be reduced to "At least one of the Lean kernel or the 'external_kernels' is correct")". By Comparator's and lean4export's READMEs, both kernel checks depend on the same Lake build and the same lean4export export, while only the Lean kernel's check runs inside the Comparator process. Two kernels therefore reduce the risk of a kernel bug but not of an exporter or build bug.

What agreement does not test. Our replay agrees with an independent replay on all 402 configurations where both have a verdict. Every outcome on both sides of that comparison was acceptance, so the comparison does not test rejection. Rejection is tested by negative controls. The standalone nanoda check rejects the Lemma 3 export with Classical.choice not permitted (exit 1), and rejects it with rectangle's proof replaced by interpolation's proof (exit 101). Comparator rejects the SiegelZeros solution with Classical.choice not permitted (Illegal axiom detected, exit 1) and with the challenge statement altered (statement mismatch, exit 1).

Statement faithfulness is by reading. The six statement matches rest on reading, signed off by the author, and the reasoning behind them is recorded point by point, also signed off. The non-vacuity lemmas rule out a vacuous challenge but do not establish faithfulness. The HeckeSevenEighths match rests partly on Lean lemmas (the lattice series for Re s $>$ 1 and holomorphy off 0, 1) and partly on reading; the six-generator identification itself is not formalized in Lean.

Scope of the reroute. The reroute changes one of the two checked theorems, dirichletRealZeroBound_proof. "Only use" of actual_biquadratic_rectangle_span holds within the SiegelZeros tree. The separate copy in namespace OAI.Erdos970 is measured at import level only, for 3 of 416 configurations. The dependency scan matches five names (uniform_rectangular_multiplicity, source_rectangle_span_of_polynomial_zero_test, actual_biquadratic_rectangle_span, actual_biquadratic_rectangle_span_via_lemma3, interpolation), so "no longer contains" applies to those names only.

Dependencies. The rerouted chain still imports OpenAI's patched PrimeNumberTheoremAnd.SiegelZeros.HadamardSupport, whose correctness is outside this work.

Sandbox. The primary results ran under a deny list naming all 45 address families of kernel 5.15.167.4, under which all 138 socket() calls of an all-families probe fail. Earlier runs used a 5-family filter, which denied IP, Unix-domain, netlink and packet sockets but left AF_VSOCK open, with a connection to the Windows host succeeding, and also left AF_PPPOX and AF_ALG creatable [6]. Three runs had no network filter: the NonVacuity lemmas, the standalone nanoda run and the statement-check build. None of those three is a run where a filter affects the result.

[9] The patch used for the unofficial build B3 is nealmcb's nanoda-let-fvar-and-succ-guard-37cb87b.patch, sha256 edd9824f5fd93c86d3f52acea1373a915ccfddf23db33c5e919f03f7c642bc30. His 416-config sweep ran on nanoda master 8199c74.

Platform. The WSL memory changed from the 15,965 MB default to 24 GB at 2026-10-21 13:56:21 and to 20 GB at 2026-10-22 08:10, during the extended replay. MemoryMax is not enforced in this WSL; a 600 MB allocation under MemoryMax=200M runs to the end, so each capped run set its own v1 memory cgroup instead. For the fd4aeeb runs the package directory was bind-mounted read-only from the original clone, and the OAI closure was built from scratch.

Superseded method. lean-checks/MainRerouted.lean defines a metaprogram, elab "#reroute", which adds copies of the closure with addDecl; it is not evidence for any claim here, and its results file is not cited.

Tool versions. The official nanoda pin is 3a24072; the results from f67473b, 37cb87b and 37cb87b with nealmcb's patch are unofficial and are reported separately.

9 Artifacts

The repository state of this final version is tagged paper-v1.

Archived packets. Zenodo record 10.5281/zenodo.23269528 holds the reviewer packet v1 zip, from the GitHub tag reviewer-packet-v1 [20]. Zenodo record 10.5281/zenodo.23499651 holds the reviewer packet v2 preliminary zip, from the tag reviewer-packet-v2-preliminary [21]. Zenodo record 10.5281/zenodo.23523697 holds the reviewer packet v3 zip, from the tag reviewer-packet-v3 [22]. Zenodo record 10.5281/zenodo.23505836 holds the corpus replay zip, from repository commit 60552ee [19]. The concept DOI of the reviewer packets is 10.5281/zenodo.23269527 [23].

Lean files. lean-checks/Lemma3.lean states Lemma3.interpolation at lines 816-825 and imports Mathlib only; its LF sha256 is ecdbf400b24f9241cd03cf383bdcd97ed7 1e4812578e66d7159a846447f7b1a7. The comparator copy lean-checks/comparator-copy/PaperLemma3/Lemma3.lean, the file installed for the accepted rerouted Comparator run, has sha256 1ce196f54fa98689bbea6bd74f2abdb110f5cbcca2eecddcca9316b8af4652 83. The comparator copy lean-checks/comparator-copy/PaperLemma3/Bridge.lean states Lemma3.rectangle, Lemma3.bridge_descent, Lemma3.bridge_main and Lemma3.actual_biquadratic_rectangle_span_via_lemma3, with LF sha256 af401514 808dde5bb913ce8e87cc5280b704a234ea324897d1a1ced1f218f250. The canonical file lean-checks/Lemma3Bridged.lean has LF sha256 6ce38ac1ef33c8b46028205bb043718565b67917 5b63cc6fbf8089136554cdb4, and its declarations are identical to those of the comparator copy. lean-checks/NonVacuity.lean, with the non-vacuity lemmas, has LF sha256 b9fae4da5 517a97da50d3fdb5356329f03f00b7ab935a448a999b2c59b24d9f9. lean-checks/README.md gives the role of each Lean file; it is committed at 758b44f74a208791c4c59b898d1ba104122a e7d0, with LF sha256 c954fe424f8450ebbbadef60ed1649f94cbd84c25223a30389e2480fc5cd8b f9.

Tool pins. All runs used Lean 4.34.1 (5045d005), Mathlib d13f23b, Comparator v4.34.0 d03acab and lean4export v4.34.0 076e8e5 [8, 10, 7, 9]. The full commits are d03acab154d2 69c06e60e4de7e4cc85deebff94b for Comparator and 076e8e57707e813375e8f9da8bf989799ac e9680 for lean4export. The official nanoda is 3a24072; f67473b, 37cb87b and 37cb87b with nealmcb's patch at 64 MiB are unofficial [1].

Results folders of the primary runs. The fd4aeeb control and reroute runs, with their dependency scans, are in results/2026-10-27-reroute-fd4aeeb/. The axioms of the

comparator copy are in results/2026-10-26-claims-ledger/axioms-comparator-copy-2026-10-26.txt. The outcome comparison with the independent replay is in results/2026-10-27-neal-comparison/. The equivalence of the two challenge statements and the route comparison are in results/2026-10-27-two-routes/. The closure of exists_absolute_real_zero_gap with its defining modules is in results/2026-10-27-closure-modules/.

10 Use of AI tools

Claude Code, an AI coding agent, wrote the Lean files in lean-checks/, the run scripts in wsl-comparator/ and src/, the ledger builders and check.py, and the drafts of Sections 2–7: every tracked file in those places has at least one commit carrying a Co-Authored-By: Claude trailer. Claude Code also wrote the reading notes in results/2026-10-27-readings/, which record its readings before the author's. The repository log records four public posts whose drafts were written in the Claude Code sessions. The author states that Claude in chat did the planning and the prompts, gave the Lemma 3 and Corollary 4 statement comparison that preceded his reading, and helped draft public posts.

The author directed the work, approved each step, and performed the seven readings and the reviews of the reasoning points behind the six statement readings, each signed off in docs/log.md. The author wrote the content of the abstract, the introduction, Section 8 and this section as point lists, and Claude Code drafted the text from them. The author checked every reference against its source. The author takes full responsibility for the content of this paper, including all text and code produced with AI tools. No specialist in analytic number theory or transcendence has reviewed the mathematics.

References

[1] ammkrn. nanoda_lib. GitHub repository ammkrn/nanoda_lib, commit 3a2407216ee84a75f9e1aead6803d0578be06ae7 (2026-09-22), 2026. Author given as the GitHub account name. URL: https://github.com/ammkrn/nanoda_lib/tree/3a2407216ee84a75f9e1aead6803d0578be06ae7.

[2] William D. Banks. On the OpenAI proof that Landau–Siegel zeros do not exist. arXiv:2610.15483 v1, 11 October 2026, October 2026. URL: https://arxiv.org/abs/2610.15483, doi:10.48550/arXiv.2610.15483.

[3] Alexander Bastounis, Fabian Circelli, and Anders C. Hansen. Navier-Stokes lost in translation: Why Lean verification of AI autoformalisation does not guarantee correct natural language proofs. arXiv:2610.08144 v1 [math.AP], 6 October 2026, October 2026. URL: https://arxiv.org/abs/2610.08144.

[4] S. Fischler. Interpolation on algebraic groups. *Compositio Mathematica*, 141(4):907–925, 2005. doi:10.1112/S0010437X05001351.

[5] grwtsk. [AUDIT.26] reproduce manuscript PDFs and bind paper statements to exact Lean declarations. grwtsk/openai-math issue #29, opened 2026-10-09; comment 6296359378 (2026-10-20), 2026. Author given as the GitHub account name. URL: https://github.com/grwtsk/openai-math/issues/29#issuecomment-6296359378.

[6] grwtsk. [AUDIT.35] independently audit the uniform Landau–Siegel real-zero gap. grwtsk/openai-math issue #40, opened 2026-10-09; comments 6087525965 (2026-10-09) and 6431288875 (2026-10-27 UTC), 2026. Author given as the GitHub account name. URL: https://github.com/grwtsk/openai-math/issues/40.

[7] Lean FRO. Comparator. GitHub repository leanprover/comparator, tag v4.34.0, commit d03acab154d269c06e60e4de7e4cc85deebff94b (2026-09-14), 2026. URL: https://github.com/leanprover/comparator/tree/d03acab154d269c06e60e4de7e4cc85deebff94b.

[8] leanprover. Lean 4. GitHub repository leanprover/lean4, version v4.34.1, commit 5045d0056413266e57c625dcd7c365b10e377c52 (2026-09-24), 2026. Author given as the owning GitHub organization. URL: https://github.com/leanprover/lean4/tree/5045d0056413266e57c625dcd7c365b10e377c52.

[9] leanprover. lean4export. GitHub repository leanprover/lean4export, tag v4.34.0, commit 076e8e57707e813375e8f9da8bf989799ace9680 (2026-09-14), 2026. Author given as the owning GitHub organization. URL: https://github.com/leanprover/lean4export/tree/076e8e57707e813375e8f9da8bf989799ace9680.

[10] leanprover-community. Mathlib. GitHub repository leanprover-community/mathlib4, commit d13f23b723b8a846827a245b89c10fc7d3f11612 (2026-09-24), 2026. Author given as the owning GitHub organization. URL: https://github.com/leanprover-community/mathlib4/tree/d13f23b723b8a846827a245b89c10fc7d3f11612.

[11] Neal McBurnett. OAI.SquareDifference.source_ordered_contraction overflows the stack on the current master. nanoda_lib issue #62, opened 2026-10-23, 2026. URL: https://github.com/ammkrn/nanoda_lib/issues/62.

[12] Neal McBurnett. openai-math-lean-tooling. GitHub repository nealmcb/openai-math-lean-tooling, commit 1ca7ff2a0280760783ee2b198c0d6f2a1cfeaba4 (2026-10-25), 2026. URL: https://github.com/nealmcb/openai-math-lean-tooling/tree/1ca7ff2a0280760783ee2b198c0d6f2a1cfeaba4.

[13] Neal McBurnett. Stack overflow, then "index exceeds 31-bit capacity" panic, in `whnf` / `try_reduce_nat` on a small set of declarations that Lean's kernel checks in seconds. nanoda_lib issue #55, opened 2026-10-15; comment 6293743601 (2026-10-20), 2026. URL: https://github.com/ammkrn/nanoda_lib/issues/55.

[14] Neal McBurnett. Three openai/math configurations need a deeper stack on the current master than on the earlier one. nanoda_lib issue #61, opened 2026-10-23; comment 6402528075 (2026-10-25), 2026. URL: https://github.com/ammkrn/nanoda_lib/issues/61.

[15] OpenAI. openai/math. GitHub repository, commit adc7f1241b42e322a6451854ab7e4b4c146bf78a (2026-10-06), 2026. URL: https://github.com/openai/math/tree/adc7f1241b42e322a6451854ab7e4b4c146bf78a.

[16] OpenAI. openai/math. GitHub repository, commit fd4aeeb2ee4fc729c18d98444fed42fd0529eeeb (2026-10-08 UTC), 2026. URL: https://github.com/openai/math/tree/fd4aeeb2ee4fc729c18d98444fed42fd0529eeeb.

[17] OpenAI. The quasi-riemann hypothesis: A zero-free half-plane $\mathrm{Re}\, s > 7/8$. Manuscript dated 30 September 2026, openai/math, preprints/The-Quasi-Riemann-Hypothesis-September-30-2026/paper.pdf, sha256 8fe93046…, identical at commits adc7f12 and fd4aeeb, September 2026. URL: https://github.com/openai/math/blob/fd4aeeb2ee4fc729c18d98444fed42fd0529eeeb/preprints/The-Quasi-Riemann-Hypothesis-September-30-2026/paper.pdf.

[18] OpenAI. Uniform exclusion of Landau–Siegel zeros. Manuscript dated October 1, 2026, openai/math, preprints/

Listing 11: The 7/8 challenges at openai/math fd4aeeb: lean/ComparatorChallenges/QuasiRiemannHypothesis.lean:5-6 (sha256 065f8c9a…), DirichletSevenEighths.lean:8-11 (sha256 8fb13ad9…) and HeckeSevenEighths.lean:237-239 (sha256 cb5e404f…).

```
theorem riemannZeta_ne_zero_of_seven_eighths_lt_re
    {s : ℂ} (hs : (7 / 8 : ℝ) < s.re) : riemannZeta s ≠ 0 := by
```

```
theorem LFunction_ne_zero_of_seven_eighths_lt_re
    {q : ℕ} [NeZero q] (χ : DirichletCharacter ℂ q) {s : ℂ}
    (hs : (7 / 8 : ℝ) < s.re) (hpole : ¬ (χ = 1 ∧ s = 1)) :
    _root_.DirichletCharacter.LFunction χ s ≠ 0 := by
```

```
theorem LFunction_ne_zero_of_seven_eighths_lt_re
    (χ : Character) {s : ℂ} (hs : (7 / 8 : ℝ) < s.re)
    (hpole : ¬ (χ.residue = 1 ∧ s = 1)) : LFunction χ s ≠ 0 := by sorry
```


Uniform-exclusion-of-Landau-Siegel-zeros-October-1-2026/paper.pdf, sha256 fab9f6e8…, identical at commits adc7f12 and fd4aeeb, October 2026. URL: https://github.com/openai/math/blob/fd4aeeb2ee4fc729c18d98444fed42fd0529eeeb/preprints/Uniform-exclusion-of-Landau-Siegel-zeros-October-1-2026/paper.pdf.

[19] Joseph M. Reilly. Comparator and nanoda replay of the openai/math Lean corpus at fd4aeeb, with statement checks for family 003. Zenodo, version corpus-replay-2026-10-20, 2026-10-20, 2026. doi:10.5281/zenodo.23505836.

[20] Joseph M. Reilly. Verification package for openai/math family 003, uniform exclusion of Landau–Siegel zeros: comparator + nanoda replay, Lean formalization of the paper's Lemma 3, rerouted proof, exact small-N checks. Zenodo, version reviewer-packet-v1, 2026-10-09, 2026. doi:10.5281/zenodo.23269528.

[21] Joseph M. Reilly. Verification package for openai/math family 003, uniform exclusion of Landau–Siegel zeros: comparator + nanoda replay, Lean formalization of the paper's Lemma 3, rerouted proof, exact small-N checks (preliminary). Zenodo, version reviewer-packet-v2-preliminary, 2026-10-20, 2026. doi:10.5281/zenodo.23499651.

[22] Joseph M. Reilly. Verification package for openai/math family 003, uniform exclusion of Landau–Siegel zeros: machine-checked complement to existing human write-ups (comparator + nanoda, Lean Lemma 3, explicit constant). Zenodo, version reviewer-packet-v3, 2026-10-21, 2026. doi:10.5281/zenodo.23523697.

[23] Joseph M. Reilly. Verification package for openai/math family 003, uniform exclusion of Landau–Siegel zeros: machine-checked complement to existing human write-ups (comparator + nanoda, Lean Lemma 3, explicit constant). Zenodo concept DOI (all versions); resolves to the latest version, reviewer-packet-v3, 2026. doi:10.5281/zenodo.23269527.

A The three 7/8 challenge statements

Listing 11 quotes the statements of the QuasiRiemannHypothesis, DirichletSevenEighths and HeckeSevenEighths challenges at openai/math fd4aeeb.

B Reasoning points for the three 7/8 challenges

Tables 7–9 list the reasoning points behind the QuasiRiemannHypothesis, DirichletSevenEighths and HeckeSevenEighths readings, one ledger row per point; each is INFERRED and marked Read by J. Reilly, 2026-10-27: agree.

Table 7: Reasoning points behind the QuasiRiemannHypothesis reading. INFERRED; Read by J. Reilly, 2026-10-27: agree.

point	finding
quantifier order: 7/8 is a fixed literal	MATCH
hypotheses: only Re s > 7/8; no pole hypothesis	MATCH
conclusion: != 0 on the open half-plane Re s > 7/8; the paper excludes the boundary	MATCH
Mathlib meaning: riemannZeta is hurwitzZetaEven 0, the continuation, with a junk value at 1	MATCH
edge case s = 1: Mathlib's riemannZeta 1 is a nonzero junk value, so the statement asserts a true fact about it and nothing about the pole, which the paper allows	MATCH

Table 8: Reasoning points behind the DirichletSevenEighths reading. INFERRED; Read by J. Reilly, 2026-10-27: agree.

point	finding
quantifier order: fixed 7/8, uniform over every q >= 1, chi and s	MATCH
hypotheses: any modulus q >= 1 and any character, imprimitive ones and q = 1 included, as in the paper's definition	MATCH
hypotheses: hpole excludes s = 1 only for the principal character mod q	MATCH
conclusion: != 0, strict half-plane	MATCH
Mathlib meaning: LFunction is the continuation; for q = 1 it is riemannZeta; for chi != 1 it is entire, so its value at s = 1 is genuine	MATCH
edge cases: imprimitive nonprincipal characters gain zeros only on Re s = 0; the principal pole at 1 is excluded by hpole	MATCH

Table 9: Reasoning points behind the HeckeSevenEighths reading. INFERRED; Read by J. Reilly, 2026-10-27: agree.

point	finding
quantifier order: fixed 7/8, every Character	MATCH
hypotheses: Character (nonzero modulus, MulChar on O/m trivial on units, zero on non-units) is exactly a ray class character mod m extended by zero, since O is a PID with 6 units	MATCH
hypotheses: imprimitive characters are included on both sides	MATCH
hypotheses: the auxiliary period does not matter (HeckeHolo.period_indep)	MATCH
hypotheses: hpole excludes s = 1 for the principal character only	MATCH
conclusion: != 0, strict half-plane	MATCH
definitions: the lattice series for Re s > 1, holomorphy off {0, 1} and the norm formula fix the function in 7/8 < Re s <= 1 by the identity theorem	MATCH
identification: each nonzero ideal has exactly 6 generators; chi and the norm are unit-invariant; so (1/6) times the sum over alpha equals the sum over ideals (absolute convergence)	MATCH
the Dedekind factorization is not needed for this match (checked numerically only)	consistent
edge case s = 0 (outside the half-plane): a Gamma(0) = 0 junk value, no effect on the statement	MATCH