A CLAIM-CENTERED PUBLICATION FRAMEWORK FOR EFFICIENT AUDITING OF AI-ASSISTED THEORETICAL RESEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

AI systems are beginning to contribute directly to theoretical research and have already produced exciting mathematical results. As these systems generate new mathematical results at an increasing pace, a growing challenge is how to review these results efficiently. Existing projects release supporting artifacts organized around research runs rather than individual claims, which can make it difficult for reviewers to locate and assess the evidence supporting individual claims. We therefore propose a claim-centered publication framework designed to improve audit efficiency in AI-assisted theoretical research. Under this framework, a problem-specific AND/OR proof DAG serves as a research record maintained throughout the project and shared alongside the final paper. Each claim is represented by a node that records its dependencies, evidence, and review status and links directly to its supporting materials. Failed claims and proof routes are explicitly retained rather than discarded. We study two complementary cases in quantum query complexity and find that, in each, maintaining this record accounted for roughly 5% of the project's total API-equivalent model cost. In our preliminary audit, the research-record condition used 20.9% fewer total tokens on average than the run-centered condition. The positive case, small-alphabet k-Sum, yielded an $\Omega(\sqrt{N}q^{1/6})$ lower bound for 2-Sum over odd cyclic alphabets of size $5 \leq q \leq 2N-3$, reducing the alphabet condition required by the previously used large-alphabet construction from $q \geq N^2$ to $q = \Theta(N)$. The negative case, minimal-alphabet Set Equality, did not produce the desired explicit optimal-order adversary witness, but yielded an $\Omega(n^{1/6})$ witness and reduced the remaining explicit $\Omega(n^{1/3})$ construction to four unresolved proof obligations, conditional on a separately recorded supporting layer. Together, the two cases illustrate how the proposed framework can support efficient claim-level auditing through a research record that preserves intermediate results, failed routes, and unresolved questions for future research.

1 INTRODUCTION

As foundation models become more capable, AI-assisted theoretical research is expanding from benchmark problem solving to open-ended work in mathematics and physics. We use AI-assisted research to refer to projects in which AI systems generate and revise substantial parts of the research, although humans may still select the problems, provide feedback, and perform the final evaluation. Such systems can search the literature, formulate conjectures, invoke symbolic and numerical tools, construct long-form mathematical arguments, and submit candidate results to independent AI review. Recent work illustrates the growing scope of these contributions. In mathematics, AI-assisted research has produced a human-verified disproof of the Erdős unit-distance conjecture Alon et al. (2026), a construction of counterexamples to the Jacobian conjecture in dimensions three and higher Gao (2026), formal proofs resolving nine open Erdős problems and 44 OEIS conjectures Tsoukalas et al. (2026), and a rank-48 bilinear algorithm for complex 4×4 matrix multiplication, improving the previous rank-49 bound for the first time in 56 years Novikov et al. (2025). In theoretical physics, AI-assisted discovery has yielded exact analytical solutions for the power spectrum of gravitational radiation emitted by cosmic strings Brenner et al. (2026) and identified the construction used to disprove Maxwell's conjectured bound for point-charge fields Arathoon et al. (2026). OpenAI

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

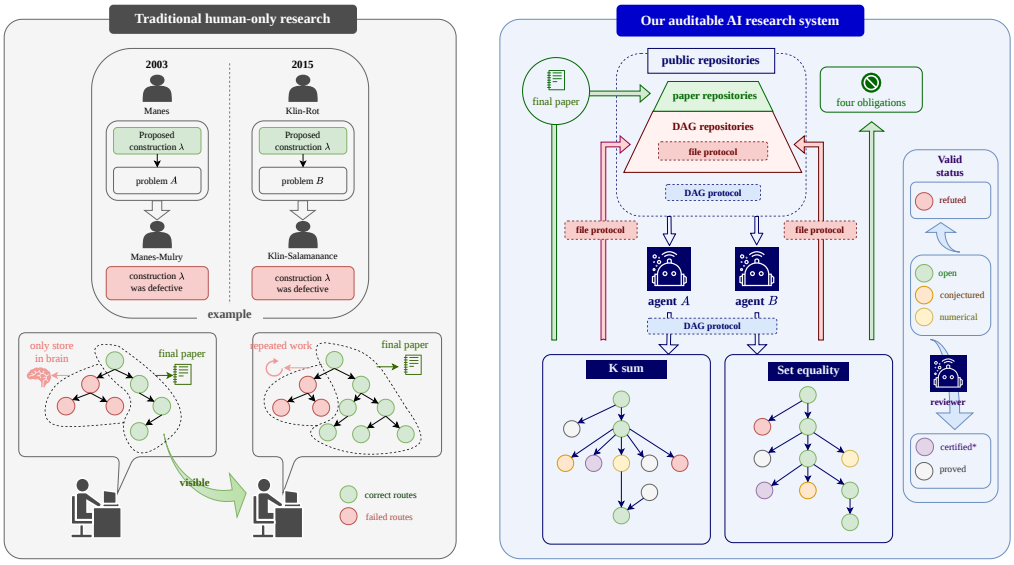


Figure 1: Overview of the claim-centered publication framework. The final paper presents the successful argument but may omit the routes that were explored and failed. The research record combines a problem-specific proof DAG, canonical node records, linked evidence, and file and DAG protocols. Maintained throughout the project and shared alongside the paper, it exposes unresolved claims and failed branches.

has also reported ten advances spanning group theory, quantum complexity, lattice problems, and extremal combinatorics OpenAI (2026b). Taken together, these results suggest that AI is becoming a contributor to open-ended theoretical research rather than merely a solver of benchmark problems.

However, a central challenge is how to organize and publish these mathematical results so that they can be reviewed efficiently. Following OpenAI's report of a large number of AI-generated mathematical results, the independent Advisory Group on Mathematics and Artificial Intelligence has explicitly sought community input on how such results should be released and disseminated to the mathematical community OpenAI (2026a); Advisory Group on Mathematics and Artificial Intelligence (2026). In practice, the internal research records maintained during AI-assisted mathematical research already contain valuable context beyond the final paper, including candidate claims, proof obligations, intermediate arguments, verifier feedback, revisions, and failed attempts Feng et al. (2026); An et al. (2026); Zheng et al. (2026); Schmitt et al. (2026); Novikov et al. (2025). Yet when research results are published, these rich internal research records are often overlooked, while supporting materials are organized around research runs rather than individual claims. We refer to this form of organization as *run-centered*. We therefore argue that supporting more efficient human review requires bridging the gap between internal research records and external publication structures. This means that internal research records will become part of the publication, providing more information than the paper alone.

In fact, the gap between internal research records and external visibility is not unique to AI-assisted research; what AI changes is the cost of closing it. In traditional research, publishing only the final paper is understandable: maintaining a structured public record of intermediate claims, dependencies, reviews, and failed routes would place a substantial burden on researchers. Yet documenting why a plausible route fails can help future researchers and AI systems avoid repeating the same mistakes. As illustrated on the left side of Figure 1, an erroneous choice-based construction appeared in work by Manes and by Manes and Mulry, and was independently repeated by Klin and Rot Manes (2003); Manes & Mulry (2007); Klin & Rot (2015). Klin and Salamanca traced this history, including earlier notices of the errors, and proved the stronger result that the double covariant powerset admits no monad structure Klin & Salamanca (2018). AI systems, by contrast, can help organize and maintain such research records for publication at relatively low cost. In our two case studies, maintaining the structured research record accounted for approximately 5.0% and 5.6% of the total API-equivalent

model cost, respectively. In our preliminary audit across Codex, DeepSeek, and Claude, the research-record condition used 20.9% fewer total tokens on average than the run-centered condition, with fewer requests in all three comparisons but no consistent advantage in wall time. Appendix B details the measurement method and its uncertainty. We argue that this modest investment offers four main benefits.

First, a reviewer can inspect each atomic claim together with its dependencies, evidence, review history, and status transitions. If an error is found, the research record makes it possible to quickly identify the downstream claims that may need to be demoted. Second, failed claims and proof routes remain explicitly recorded, allowing future researchers and agents to avoid repeating known mistakes and wasting additional tokens on eliminated approaches. Third, the research record could provide a basis for large-scale collaboration among researchers and AI agents on a shared theoretical problem. Contributors could identify unresolved proof obligations, pursue different routes, and contribute new claims, evidence, or reviews under shared protocols. This would allow research to accumulate across teams and sessions, with each contribution linked to the work on which it builds. A related approach is explored by Agora, which uses Git-backed shared research memory to support collaboration across independently scheduled agents Zhang et al. (2026). Fourth, intermediate lemmas, counterexamples, eliminated routes, and unresolved obligations remain available for future work even when a project does not produce a publication-level theorem.

We therefore propose a claim-centered publication framework with a research record as its core component, as shown in Figure 1. The record is maintained throughout the project and shared as part of the publication, helping reviewers locate claims and their supporting evidence while reducing the need to reconstruct the research process from run-centered materials. The research record combines a proof graph, claim-level evidence records, and two protocols for maintaining them. Each project maintains an AND/OR proof DAG: alternative proof routes form the OR structure, while the premises required within each route form the AND structure Chang & Slagle (1971); Kung et al. (2026). Every atomic claim is represented by a node and assigned an evidence- and review-based level. Higher levels indicate stronger support, and each node serves as the entry point for auditing its claim. When a linked Lean formalization is available, the node also records the corresponding Lean statement, claim hash, build evidence, and synchronization state. All participating agents follow two common protocols. The file protocol governs how canonical node records and supporting files are placed, named, linked, moved, and archived, while the DAG protocol governs node and route registration, evidence levels, review-gated transitions, AND/OR consistency, and the append-only preservation of failed routes.

We examine this framework through two case studies in quantum query complexity. In this model, a quantum algorithm accesses its input through an oracle, and complexity is measured by the number of oracle calls required to solve the problem rather than by total running time. We use the small-alphabet k-Sum and minimal-alphabet Set Equality projects as complementary examples. The k-Sum project provides a positive case. For 2-Sum, it obtained an $\Omega(\sqrt{N}q^{1/6})$ lower bound over odd cyclic alphabets of size $5 \le q \le 2N-3$ and a complementary $O_\varepsilon(N^{1/2+\varepsilon}q^{1/4})$ upper bound for every $q \ge 2$. The lower bound reduces the condition required by the previously used large-alphabet construction Belovs & Špalek (2013) from $q \ge N^2$ to $q = \Theta(N)$. It also obtained a super-Grover small-alphabet lower bound for every fixed $k \ge 3$; we are not aware of an earlier result with the same scope. The Set Equality project provides a negative case. It did not reach its intended target of constructing an explicit optimal-order adversary witness at $q = 2n$. Nevertheless, it produced a weaker $\Omega(n^{1/6})$ witness and showed that the desired $\Omega(n^{1/3})$ construction would follow from resolving four explicit technical obstacles under a separately recorded supporting layer. Appendix E derives the four obstacles and states the supporting assumptions in detail. This case illustrates the fourth benefit of the framework: even when a project does not reach its original target, its intermediate results and unresolved obligations remain available as a concrete starting point for future research. Both case studies also include linked Lean formalizations. For k-Sum, the Lean development checks the principal lower-bound reductions and selected obstruction arguments relative to explicitly declared interfaces. For Set Equality, it checks the conditional derivation of the target adversary bound from the four named obligations and explicitly declared supporting interfaces. Sections 4 and 5 summarize the principal results and their case-study implications; the complete mathematical developments appear in Appendices E and F.

2 THE CLAIM-CENTERED PUBLICATION FRAMEWORK

This section defines the claim-centered publication framework and the research record through which it connects ongoing research with external review. We first describe the AND/OR graph structure, then specify the contents and evidence level of each node, and finally summarize the file and DAG protocols that govern updates.

2.1 PROBLEM-SPECIFIC AND/OR PROOF DAGS

This subsection explains how claims and alternative proof routes are represented in one persistent graph. During a project, every worker reads and updates the same problem-specific AND/OR directed acyclic graph (DAG). The research record is used while claims are proposed, proved, reviewed, or handed to another worker, and it accompanies the paper rather than being reconstructed after the project ends.

AND/OR graphs have long been used to represent decomposable search problems Chang & Slagle (1971); recent theorem-proving systems likewise use proof graphs to organize long-horizon search Kung et al. (2026). We adapt this representation to a claim-centered research record without assigning permanent structural types to nodes. Each node represents one claim. A route stored in that node lists the premise claims that must all hold and explains why they jointly imply the node's claim. Different routes are alternative ways of establishing the same claim.

Supporting the premises does not automatically establish the conclusion. If a route for claim A lists premises B and C, the route must also provide a reviewable argument $B \wedge C \Rightarrow A$. The premise proofs and this implication argument may be completed in any order, but both must pass review before the route can support A. Refuting B disables every route that uses B, but it does not refute a different route to A or the claim A itself.

2.2 ATOMIC NODES AND EVIDENCE LEVELS

Listing 1 shows the core fields of the canonical node record. This subsection explains how they summarize the claim's current support; the expanded structure appears in Appendix D. Every node has a stable identifier and exactly one canonical record.

The `id` identifies the node, and `claim` states the proposition being assessed. The `assumptions` field records established external inputs used by the proof. The optional `routes` field records alternative arguments, each with its required `premises`, the `implication` to be justified, and a `proof` when one is available.

Each `assessment` records a `level`, a short `summary`, and the `basis` for that judgment, such as a proof, computation, or review. The node assessment concerns the claim, while each route assessment concerns its implication. The optional `formalization` field links to a Lean development. A reader can therefore start from one node to locate the evidence supporting its claim, without searching through the full research history.

A node's evidence level summarizes the support currently available for its claim. It is a repository-local evidence category, not a probability of correctness or a claim of external acceptance. Table 1 specifies the evidence required for each level. Higher levels require stronger evidence and review: completion levels require a locatable separate adversarial AI review, while REFUTED requires a structured counterexample or decisive contradiction identified against the precise claim or implication.

Listing 1: Core node record.

```
id: stable.problem.claim
claim: <precisely scoped claim>
assumptions:                                    # optional; direct
   proof only
  <assumption-id>:
    kind: external-theorem
    statement: <statement and scope>
    source:
      note: <precise bibliographic reference>
routes:                                         # optional; OR
   alternatives
```

```
  <route-id>:
    premises: [<node B>, <node C>]              # jointly required
    implication: <why B and C imply this claim>
    proof:
      sketch: <argument>
      sources: [<linked artifact>]
    assessment:
      level: <evidence level>
      summary: <judgment of this implication>
      basis:
        reviews: [<review record>]
assessment:
  level: <evidence level>
  summary: <judgment of this claim>
  basis:
    routes: [<route-id>]
    evidence: [<evidence record>]
    reviews: [<review record>]
formalization:                                  # optional
  lean:
    module: Project.Module
    file: <Lean file>
    claim: <formal claim name>
```

Repository level	Evidence required by the protocol
Open	a precisely stated claim or route; no reviewable argument is yet required
Conjectured	a substantive argument blueprint or reason to believe
Numerical	a reproducible computation or mechanized run, with a recorded command, linked raw output, and a locator or checksum when needed for unambiguous reproduction
Certified*	a conditional or partial result checked by a separate adversarial AI reviewer, normally in a fresh context, with every unresolved premise and scope caveat explicit
Proved	a complete argument whose route implication, load-bearing premises, and sources have passed a separate adversarial AI review
Refuted	a structured counterexample or decisive contradiction identified against the precise claim or implication, with its source linked

Table 1: The evidence levels used separately for node claims and route implications.

At the repository snapshot studied here, the Set Equality graph contains 85 canonical node files, of which 22 (25.9%) are REFUTED; the k-Sum graph contains 144 nodes, of which 14 (9.7%) are REFUTED. Thus 36 individual node claims are recorded as false. In addition, the k-Sum graph contains four routes whose implication assessments are REFUTED; the Set Equality graph currently records its failures as refuted node claims rather than route-level refutations. The retained failures include the $c > 1$ typing failure in the Set Equality construction, refuted k-Sum route implications, and exact finite counterexamples to several candidate claims. We regard these failed claims and routes as valuable because future agents can see why they failed instead of spending additional tokens repeating the same attempts.

2.3 FILE AND DAG PROTOCOLS

The record remains interpretable only if every worker follows the same update rules. In this subsection, we present the file and DAG protocols that all workers must follow. This is not the complete protocol text; the full, normative versions are distributed with the code repository. The file protocol specifies where node records and supporting files are stored and how they are linked. The DAG protocol specifies how claims, proof routes, and evidence levels may change. Together, the protocols let a new worker resume by reading the relevant node records and following their links. The detailed file rules and DAG rules are provided in Appendix A.

Table 2: Codex audit results using `gpt-5.6-sol`.

Condition	Findings	Wall time	Requests	Total tokens	API-equivalent cost
Paper only	2/3	24:05.7	15	493,253	$1.6504
Paper + Chat + Evidence	3/3	27:25.7	44	4,011,535	$3.8886
Paper + Graph + Evidence	3/3	31:16.2	40	2,722,936	$3.1549

Table 3: DeepSeek audit results using `deepseek-v4-pro`.

Condition	Findings	Wall time	Requests	Total tokens	API-equivalent cost
Paper only	0/3	23:01.6	13	324,651	$0.196821
Paper + Chat + Evidence	2/3	36:38.3	43	4,170,566	$0.518825
Paper + Graph + Evidence	2/3	23:10.1	36	3,313,991	$0.313718

Table 4: Claude audit results using `claude-opus-5`.

Condition	Findings	Wall time	Requests	Total tokens	API-equivalent cost
Paper only	2/3	23:24	17	2,195,366	$15.0636
Paper + Chat + Evidence	3/3	19:41	38	5,483,986	$21.2171
Paper + Graph + Evidence	3/3	20:03	32	4,935,033	$20.7205

3 Audit Evaluation

We tested whether a claim-centered research record helps AI models acting as auditors recover known, consequential faults from two selected historical pre-audit cases in our cyclic 2-Sum investigation. *Case 1* was an early proof attempt for the recursive upper-bound algorithm. It contained two decisive faults: a recursive call left the analyzed parameter range, and a low-probability bad event lacked a deterministic worst-case query cap. *Case 2* was an early dual-polynomial lower-bound argument. It claimed an exponent $\eta > 1/6$ for every $3 \le q \le N$, contradicting the known $O(N^{2/3})$ upper bound near $q = N$. We identified these three issues before the experiments and used them as the evaluation criteria. An issue was counted as recovered only if the auditor identified it and explained its impact.

For each model we compared *Paper only*, *Paper + Chat + Evidence*, and *Paper + Graph + Evidence*. To reproduce the proof state as it existed at the time, we excluded later reviews, repairs, and answer keys. All conditions used the same prompt, reproduced in Appendix C. For each material condition, we report the number of decisive issues recovered, wall time, number of model requests, total tokens, and API-equivalent cost. Each row reports these quantities aggregated over the two cases. The results are reported in Tables 2–4.

As the tables show, the Graph and Chat conditions recovered the same number of decisive issues for each model, and both recovered more than the Paper-only condition. The recorded API-equivalent cost was lower for Graph than for Chat in all three models. Relative to Chat, Graph used 32.1%, 20.5%, and 10.0% fewer total tokens in Codex, DeepSeek, and Claude, respectively, for a mean reduction of 20.9%. Graph also required fewer API requests in all three models, while wall time showed no consistent advantage.

4 Negative Case: Minimal-Alphabet Set Equality

We use minimal-alphabet Set Equality as the negative case because the project did not attain its principal objective: an explicit adversary witness with the optimal $\Omega(n^{1/3})$ scaling at $q = 2n$. The case is nevertheless informative for the proposed framework. It produced a weaker unconditional theorem and reduced the intended construction to four explicit proof obligations under a separately recorded supporting layer. Appendix E gives the complete mathematical development.

4.1 MAIN RESULT

Set Equality takes two injective lists $x, y : [n] \to [q]$, promised to have either equal or disjoint ranges. The minimum nontrivial alphabet is $q = 2n$, and the bounded-error quantum query complexity is already known to be $\Theta(n^{1/3})$ for every $q \geq 2n$ Brassard et al. (1998); Zhandry (2015). The unresolved constructive question is whether one can write down and directly verify an explicit negative-weight adversary matrix attaining this order at $q = 2n$.

Let Γ_n be the linearly tapered Belovs–Rosmanis construction restricted to the legal $q = 2n$ promise domain. The unconditional result obtained for this matrix is the following.

Theorem 4.1 (Unconditional case-study result)**.** *The explicit matrix Γ_n satisfies*

$$\mathrm{Adv}^{\pm}(\textsc{Set Equality}_{n,2n}) \geq \left(\frac{2}{\sqrt{6}} - o(1)\right) n^{1/6}.$$

The proof retains a numerator of order $n^{1/3}$, but the available bound on the query-masked denominator grows as $n^{1/6}$. This yields Theorem 4.1. The route from Theorem 4.1 to an explicit $\Omega(n^{1/3})$ witness has been reduced to four unresolved conditions, together with a separately recorded supporting layer. The following conjecture gives a concise summary; its full operator-level formulation and the resulting conditional bound appear in Appendix E.

Conjecture 4.2 (Representation-theoretic conditions)**.** *The following four conditions hold for every active representation level and sector of the legal lower block.*

- H1: *The two weighted contributions satisfy the required joint positive-semidefinite bound.*
- H2: *The canonical Gram operator has the required representation on the full sector carrier, including the tensor-product identity factor.*
- H3: *Each legal sector block satisfies the operator identity connecting it to its canonical Gram operator.*
- H4: *The sector decomposition is orthogonal and exhaustive, so the global norm is the maximum of the sector norms.*

4.2 REVIEW-DRIVEN CORRECTIONS

This case illustrates how the proposed framework supports external review through its claim-level research record. Appendix E.4 gives the technical account of the three findings discussed here.

A single external AI review round produced three findings bearing on the candidate $\Omega(n^{1/3})$ route. First, it refuted a proposed intertwining identity between exact-weight projection and restriction to legal inputs. Associating that finding with the restriction claim isolated the affected step; replacing it with a weaker product-level identity left the legal witness and the established numerator and denominator estimates unchanged. Second, it identified a missing tensor-product identity factor in the proposed representation of a canonical Gram operator. Once that finding was attached to the corresponding bridge claim, the DAG showed that it blocked the candidate route and became condition H2, while the numerator estimate, the unconditional denominator bound, and Theorem 4.1 lay outside the affected dependency path. Third, it showed that the reduction as then recorded was incomplete: the passage from the per-sector estimate to the global lower block required an orthogonality and exhaustiveness statement that no registered claim covered. That statement was registered as a new open node and is condition H4. A subsequent external re-review of the repaired argument confirmed the two corrections, confirmed that H1–H4 suffice for the conditional bound, and found no further hidden obligation.

The review round therefore demonstrates node-level impact localization. After an external AI reviewer raises a local objection, the record's proof DAG can be traversed to identify the downstream claims that must be rechecked, without reconstructing the proof structure from long interaction histories. Where a finding is not a local error but a gap, the same mechanism forces the missing step to be registered as its own claim rather than left implicit. This does not make mathematical review automatic, but it makes the consequences of a review finding easier to delimit and audit.

Scope of the Lean formalization. The Lean 4 development checks selected denominator calculations and shows that abstract norm-level hypotheses corresponding to H1–H4 imply the conditional $\Omega(n^{1/3})$ bound. It assumes rather than proves the operator-level conditions and their representation-theoretic foundations, and it does not cover the full unconditional $\Omega(n^{1/6})$ proof.

5 Positive Case: Small-Alphabet k-Sum

We use small-alphabet cyclic k-Sum as the positive case. Here the project produced the upper and lower bounds stated below, while the proof DAG retains the imported results, reductions, failed direct approaches, and formalization scope supporting them. Appendix F contains the complete reductions, algorithm, and query-complexity analysis.

5.1 Main result

For integers $2 \le k \le N$, an input $x = (x_1, \dots, x_N) \in \mathbb{Z}_q^N$, and a known target $t \in \mathbb{Z}_q$, cyclic k-Sum asks whether there are k distinct indices whose values sum to t modulo q. The small-alphabet problem is to determine how its quantum query complexity interpolates between Grover-scale behavior and the large-alphabet $\Theta(N^{k/(k+1)})$ regime.

Theorem 5.1 (Small-alphabet bounds for cyclic k-Sum)**.** *The following statements hold.*

(i) *If $q \ge 5$ is odd, $t \in \mathbb{Z}_q$, and $q \le 2N - 1$ for even N or $q \le 2N - 3$ for odd N, then*

$$\mathrm{Q}(2\text{-}\mathit{Sum}_{\mathbb{Z}_q,t,N}) = \Omega(\sqrt{N}\, q^{1/6}).$$

Consequently the tight $\Theta(N^{2/3})$ regime for cyclic 2-Sum begins already at $q = \Theta(N)$, whereas the previously used large-alphabet construction was analyzed under the sufficient condition $q \ge N^2$ Belovs & Špalek (2013).

(ii) *For every $\varepsilon > 0$, every $q \ge 2$, and every $t \in \mathbb{Z}_q$,*

$$\mathrm{Q}(2\text{-}\mathit{Sum}_{\mathbb{Z}_q,t,N}) = O_\varepsilon(N^{1/2+\varepsilon} q^{1/4}).$$

(iii) *Fix $k \ge 3$ and put $q_0(k) = 2k^2 + 2k - 5$. There are constants $C_k, c_k > 0$ such that, for all sufficiently large N, every q satisfying $q_0(k) \le q \le C_k N$, and every $t \in \mathbb{Z}_q$,*

$$\mathrm{Q}(k\text{-}\mathit{Sum}_{\mathbb{Z}_q,t,N}) = \Omega_k(\sqrt{N}\, q^{1/6}).$$

Moreover, for all sufficiently large N, every $q \ge c_k N$, and every $t \in \mathbb{Z}_q$,

$$\mathrm{Q}(k\text{-}\mathit{Sum}_{\mathbb{Z}_q,t,N}) = \Omega_k(N^{2/3}).$$

5.2 Literature-guided identification of the Claw reduction

The lower bounds in Theorem 5.1(i) and (iii) were obtained by transferring a known small-range Claw bound through the signed encoding in Lemma F.2. Earlier direct constructions did not yield the required small-alphabet lower bound. This failure narrowed the literature search to results with a compatible promise, query model, and parameter range.

The literature search based on these criteria identified the small-range Claw bound of Ambainis et al. Ambainis et al. (2021). We connect it to cyclic 2-Sum through a simple signed encoding; neither the Claw source nor the earlier k-Sum literature stated this connection. Specifically, the encoding transfers the Claw lower bound to cyclic 2-Sum, while the frozen-core reduction extends it to the $k \ge 3$ regime. The recursive Claw algorithm from the same work also provides the starting point for the upper-bound adaptation. These two uses are developed in Appendix F. In the accompanying AND/OR proof DAG, the imported Claw bound, the signed encoding identity, and the parameter transfer are recorded as separate claims, making their logical dependencies explicit.

Scope of the Lean formalization. The Lean 4 development checks the Claw-to-2-Sum encoding, the target and odd-length transformations, the fixed-k core, threshold arithmetic, and the lower-bound compositions for part (i) and the small-alphabet part of (iii), relative to stated Claw and query-restriction assumptions. It does not cover the large-alphabet plateau or formalize the recursive upper-bound algorithm, although it states part (ii) with its full quantifiers.

6 Conclusion and Discussion

6.1 Conclusion

We have proposed a claim-centered publication framework for AI-assisted theoretical research. Under this framework, the evolving research record is organized around atomic claims, their evidence, and their dependencies in a problem-specific AND/OR proof DAG and shared alongside the paper to help reviewers locate and assess the evidence for individual claims more efficiently. In the k-Sum case, the record supports the upper and lower bounds in Theorem 5.1 and exposes the literature-guided reductions on which they depend. In the Set Equality case, it records the $\Omega(n^{1/6})$ result while reducing the remaining $\Omega(n^{1/3})$ construction to four explicit proof obligations. Together, the project accounting and audit indicate a low-cost, high-return intervention: maintaining the record accounted for roughly 5% of project model cost, while research-record auditing used 20.9% fewer tokens on average than chat-based auditing and preserved failed routes and unresolved obligations for reuse.

6.2 Discussion

AI-generated proof should neither be accepted nor dismissed uniformly. Models can produce conjectures, calculations, code, and proofs faster than experts can verify them, but their reliability varies; human-written sources are not infallible either. OpenAI's work on non-sofic groups, for example, uses a result of Kun Kun (2016) while its accompanying report identifies and repairs a gap in that result's proof (OpenAI, 2026b, Chapter 3, Section 2). Confidence should therefore attach to individual claims and their supporting evidence. Our evidence levels are only a coarse operational treatment: they distinguish preliminary reasoning, reproducible computation, adversarial review, and linked Lean formalization, but are not calibrated probabilities. Even a kernel-checked proof guarantees only its formal statement under declared assumptions; whether that statement captures the intended claim still requires judgment.

For open theoretical problems, a plausible answer seems not enough: its key claims, supporting literature, alternative approaches, and evidence must also be available for independent review. The two case studies illustrate how the research record maintained under the proposed framework can preserve verified steps, refuted routes, and precise open obligations for future researchers and agents, even when a run falls short of its target. More broadly, such records may provide lightweight infrastructure for AI-assisted theoretical research by preserving an inspectable and extensible account of claims, evidence, failed routes, and open obligations.

Several limitations should also be noted. The audit experiment is preliminary: each condition was run only once, and all runs covered the same two case studies. The observed differences in token usage, request counts, wall time, and issue recovery therefore should not be interpreted as statistically significant or broadly causal. Future work should repeat the evaluation across a larger and more diverse set of cases, with multiple independent runs and human baselines, to quantify variability and improve the reliability of the comparison.

Code and data availability. The accompanying code and research records are maintained at an anonymized repository supplied with the supplementary material. The `se/` and `ksum/` directories contain the claim graphs, linked evidence and reviews, Lean developments, and checking programs; the repository root contains the protocols and validators. Commands, dependencies, and numerical-check scopes accompany the relevant evidence. Set Equality result nodes are `SE.br.cor16` and `SE.goal.conditional-optimal`; the principal k-Sum result nodes are `KSUM.k2.lower`, `KSUM.k2.threshold`, `KSUM.A1d`, and `KSUM.B4a`.

AI use statement

The authors developed the conceptual framework and wrote the manuscript. Generative AI tools were used to assist with mathematical analysis, proofs and formalization, the audit study, research-record maintenance, literature search, and refinement of the manuscript's language and presentation.

REFERENCES

Advisory Group on Mathematics and Artificial Intelligence. Advisory group on mathematics and artificial intelligence, 2026. URL https://agmai.org/. See "Current Task" and the linked call for community input; accessed September 22, 2026.

Noga Alon, Thomas F. Bloom, W. T. Gowers, Daniel Litt, Will Sawin, Arul Shankar, Jacob Tsimerman, Victor Wang, and Melanie Matchett Wood. Remarks on the disproof of the unit distance conjecture. *arXiv preprint*, 2026. URL https://arxiv.org/abs/2605.20695.

Andris Ambainis, Kaspars Balodis, and Jānis Iraids. A note about claw function with a small range. In *16th Conference on the Theory of Quantum Computation, Communication and Cryptography*, volume 197 of *Leibniz International Proceedings in Informatics*, pp. 6:1–6:5, 2021. doi: 10.4230/LIPIcs.TQC.2021.6. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.TQC.2021.6. arXiv:2103.16390.

Chenyang An, Qihao Ye, Minghao Pan, and Jiayun Zhang. QED: An open-source multi-agent system for generating mathematical proofs on open problems. *arXiv preprint*, 2026. URL https://arxiv.org/abs/2604.24021.

Philip Arathoon, Gavin Ball, and Matthew D. Kvalheim. The maxwell conjecture is false. *arXiv preprint*, 2026. URL https://arxiv.org/abs/2607.27197.

Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. doi: 10.1145/502090.502097. Preliminary version in FOCS 1998.

Aleksandrs Belovs and Ansis Rosmanis. Adversary lower bounds for the collision and the set equality problems. *Quantum Information and Computation*, 18(3–4):198–222, 2018. doi: 10.26421/QIC18.3-4-2. arXiv:1310.5185.

Aleksandrs Belovs and Robert Špalek. Adversary lower bound for the k-sum problem. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, pp. 323–328, 2013. doi: 10.1145/2422436.2422474. arXiv:1206.6528.

Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum cryptanalysis of hash and claw-free functions. In *LATIN'98: Theoretical Informatics*, volume 1380 of *Lecture Notes in Computer Science*, pp. 163–169. Springer, 1998. doi: 10.1007/BFb0054319. arXiv:quant-ph/9705002.

Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Information*, volume 305 of *Contemporary Mathematics*, pp. 53–74. American Mathematical Society, 2002. doi: 10.1090/conm/305/05215. arXiv:quant-ph/0005055.

Michael P. Brenner, Vincent Cohen-Addad, and David Woodruff. Solving an open problem in theoretical physics using AI-assisted discovery. *arXiv preprint*, 2026. URL https://arxiv.org/abs/2603.04735.

Chin-Liang Chang and James R. Slagle. An admissible and optimal algorithm for searching AND/OR graphs. *Artificial Intelligence*, 2(2):117–128, 1971. doi: 10.1016/0004-3702(71)90006-3.

Andrew M. Childs and Jason M. Eisenberg. Quantum algorithms for subset finding. *Quantum Information and Computation*, 5(7):593–604, 2005. URL https://arxiv.org/abs/quant-ph/0311038. arXiv:quant-ph/0311038.

Christoph Dürr and Peter Høyer. A quantum algorithm for finding the minimum, 1996. URL https://arxiv.org/abs/quant-ph/9607014v2. Revised version 2, 7 January 1999.

Tony Feng et al. Towards autonomous mathematics research. *arXiv preprint*, 2026. URL https://arxiv.org/abs/2602.10177.

Shuhong Gao. Counterexamples to the jacobian conjecture in dimensions greater than two. *arXiv preprint*, 2026. URL https://arxiv.org/abs/2608.00222.

Peter Høyer, Troy Lee, and Robert Špalek. Negative weights make adversaries stronger. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pp. 526–535, 2007. doi: 10.1145/1250790.1250867. URL https://arxiv.org/abs/quant-ph/0611054.

Bartek Klin and Jurriaan Rot. Coalgebraic trace semantics via forgetful logics. In Andrew M. Pitts (ed.), *Foundations of Software Science and Computation Structures*, volume 9034 of *Lecture Notes in Computer Science*, pp. 151–166. Springer, 2015. doi: 10.1007/978-3-662-46678-0_10.

Bartek Klin and Julian Salamanca. Iterated covariant powerset is not a monad. *Electronic Notes in Theoretical Computer Science*, 341:261–276, 2018. doi: 10.1016/j.entcs.2018.11.013.

Gábor Kun. On sofic approximations of property (T) groups. *arXiv preprint*, 2016. URL https://arxiv.org/abs/1606.04471v5. Version 5, revised 30 July 2019.

Po-Nien Kung, Linfeng Song, Dawsen Hwang, Jinsung Yoon, Chun-Liang Li, Simone Severini, Mirek Olšák, Edward Lockhart, Quoc V. Le, Burak Gokturk, Thang Luong, Tomas Pfister, and Nanyun Peng. LEAP: Supercharging LLMs for formal mathematics with agentic frameworks. *arXiv preprint*, 2026. doi: 10.48550/arXiv.2606.03303.

Troy Lee, Rajat Mittal, Ben W. Reichardt, Robert Špalek, and Mario Szegedy. Quantum query complexity of state conversion. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science*, pp. 344–353, 2011. doi: 10.1109/FOCS.2011.75. URL https://arxiv.org/abs/1011.3020. arXiv:1011.3020.

Ernie Manes. Monads of sets. In Michiel Hazewinkel (ed.), *Handbook of Algebra*, volume 3, pp. 67–153. Elsevier, 2003. doi: 10.1016/S1570-7954(03)80059-1.

Ernie Manes and Philip Mulry. Monad compositions I: General constructions and recursive distributive laws. *Theory and Applications of Categories*, 18(7):172–208, 2007. URL https://www.tac.mta.ca/tac/volumes/18/7/18-07.pdf.

Alexander Novikov et al. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint*, 2025. URL https://arxiv.org/abs/2506.13131.

OpenAI. Advisory group on mathematics and artificial intelligence, September 2026a. URL https://openai.com/index/advisory-group-on-mathematics-and-ai/. Published September 21, 2026; accessed September 22, 2026.

OpenAI. Ten advances in mathematics and theoretical computer science. Research report, August 2026b. URL https://cdn.openai.com/pdf/ten-proofs-oai.pdf. Accessed 2026-08-15.

Ansis Rosmanis. Quantum adversary lower bound for element distinctness with small range. *Chicago Journal of Theoretical Computer Science*, 2014(4), 2014. URL https://arxiv.org/abs/1401.3826. arXiv:1401.3826.

Johannes Schmitt, Tim Gehrunger, Jasper Dekoninck, Gergely Bérczi, Uri Kreitner, Liam Price, and David Holmes. ProofCouncil: An LLM agent for solving open mathematical problems. *arXiv preprint*, 2026. URL https://arxiv.org/abs/2607.09474.

Robert Špalek. Adversary lower bound for the orthogonal array problem. *arXiv preprint*, 2013. URL https://arxiv.org/abs/1304.0845. arXiv:1304.0845.

Seiichiro Tani. Approximate degrees of multisymmetric properties with application to quantum claw detection. *arXiv preprint*, 2024. URL https://arxiv.org/abs/2410.02243. arXiv:2410.02243.

George Tsoukalas et al. Advancing mathematics research with AI-driven formal proof search. *arXiv preprint*, 2026. URL https://arxiv.org/abs/2605.22763.

Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Information and Computation*, 15(7–8):557–567, 2015. URL https://arxiv.org/abs/1312.1027. arXiv:1312.1027.

Yifan Zhang, Yunheng Zou, Shaokun Zhang, Jian Hu, Hao Zhang, Binfeng Xu, Jan Kautz, and Yi Dong. Agora: Git as shared memory for collective AutoResearch. *arXiv preprint*, 2026. URL https://arxiv.org/abs/2609.18094.

Daniel Zheng et al. AI co-mathematician: Accelerating mathematicians with agentic AI. *arXiv preprint*, 2026. URL https://arxiv.org/abs/2605.06651.

A Detailed File and DAG Protocol Rules

File protocol. The file protocol keeps each case study's records and supporting files in its own directory at the repository root: `se/` or `ksum/`. Readers can locate the material supporting a claim by following links from its node record. The main rules are:

- **F1.** One record per node. Each node has one permanent identifier and one canonical record under `proof-dag/nodes/`. Indexes and graph visualizations are generated from these records.
- **F2.** Store material by type. Status events remain in the node record. External review reports go in `reviews/`, supporting proofs, logs, certificates, and reference notes in `artifacts/`, and linked Lean formalizations in `lean/`. Project-specific mathematical checkers go in `tools/`. Only validators used by both projects are stored in the repository-level `tools/` directory. Supporting files use stable, descriptive names.
- **F3.** Link every supporting file. The node or route assessment links directly to each proof, run, certificate, or counterexample used to justify its current level. When an external review applies, the assessment links to the specific finding in the corresponding report.
- **F4.** Complete all parts of an update together. A worker stores the new material, appends the corresponding event, and updates the affected assessment. The update is considered complete only after the structural and link checks pass.
- **F5.** Preserve the record of prior work. Cited attempts, counterexamples, negative results, reviews, and status changes are not deleted. If a file moves or is superseded, existing links are updated and the replacement remains connected to the earlier material.

DAG protocol. The DAG protocol determines when a claim or route may change level. Its main rules are:

- **D1.** Use only registered claims and routes. A claim can be used as a dependency only after its canonical node record exists. A route can support a claim only after it has been registered in that claim's node record with its premises, implication, and current assessment.
- **D2.** Turn every unresolved premise into a node. If a route depends on an unproved statement, record that statement as an OPEN node and list it as a premise. Use `assumptions` only for established external theorems or model facts, and state their exact scope and source.
- **D3.** Link evidence to what it supports. Attach every proof, computation, counterexample, run, or review finding to the assessment of the exact node or route it supports. A file or transcript is evidence only when the assessment explains what it establishes.
- **D4.** Independent adversarial AI review for CERTIFIED*. Before either a node or a route can be marked CERTIFIED* or PROVED, it must receive a locatable finding in an independent adversarial AI review. The reviewer must examine every premise, source, and load-bearing step while actively searching for errors.
- **D5.** Keep failures local. Refuting a premise invalidates the routes that require it, while refuting an implication invalidates only that route. Mark a claim REFUTED only when the claim itself has been disproved; otherwise keep it OPEN or PARKED if another proof may still exist.
- **D6.** Keep a dated history. Record every claim rewrite, level change, failure, repair, route retirement, and formalization mismatch as a dated event with its source. A level-change event records its previous and new levels and, when applicable, the affected route. Do not delete failed or superseded work.

D7. Check every update. After each update, regenerate all derived views and check the structural consistency of the research record. Verify identifiers, links, acyclicity, allowed fields, completion-level reviews, and the append-only event history.

D8. Register substantive routes before testing them. Add every substantive proposed claim or route to the graph before testing it or allowing another claim to depend on it, including alternatives not selected for immediate work. Before moving on, record whether the attempt succeeded, failed, or was left untested and why; no route may disappear silently.

B USAGE-BASED COST ACCOUNTING

The maintenance percentages reported in the Introduction are computed from client-recorded API usage, not from estimated token counts. Each request is priced separately because the long-context rate depends on that request's input length.

For Codex, we scan both the recovered and local `rollout` trees while tracking the working directory recorded in each event. We retain only events inside this repository and read `last_token_usage`, not the cumulative usage field. Consecutive duplicate events are removed, and a global key formed from timestamp, input, and output prevents overlapping or forked session logs from being charged twice. Fresh input is input minus cached input. Requests with more than 128,000 input tokens use the assumed long-context price; all others use the assumed short-context price. A session is assigned to a project by keyword voting over its initial context.

For Claude, the scan includes each main session and all of its subagents. Repeated rows with the same message identifier are merged: the largest usage counters are retained, while all thinking, text, and tool-use content blocks are combined so that tool calls are not lost. Cache writes are priced from their recorded five-minute or one-hour lifetime. Subagents inherit the project label of their parent session. Each deduplicated assistant message is then classified by the target of its tool calls: proof-DAG files, protocols, and persistent research-state files count as DAG maintenance; mathematical programs and formalization count as mathematics; manuscript files count as paper writing; orchestration tools count as coordination; and tool-free text counts as reporting. Codex usage is conservatively assigned to mathematics.

The resulting amounts are API-equivalent costs under the fixed prices used in the accounting analysis, not subscription charges. Mixed Claude sessions are the largest uncertainty: alternative attribution lowers the k-Sum DAG share from 5.62% to approximately 3.3%. The Set Equality Claude allocation also used a coarser output-share attribution than the message-level k-Sum pass. Accordingly, the robust conclusion is that DAG maintenance was on the order of five percent in both projects, not that the two percentages differ precisely.

C REVIEWER PROMPT USED ACROSS ALL AUDIT CONDITIONS

Listing 2: Adversarial-review prompt used in all audit conditions.

```
You are an independent adversarial mathematical reviewer working
   in a fresh context. The current directory contains the
   complete material package assigned to you for one historical
   pre-audit research state. You have not been given any later
   review, correction, or answer key.

Strict conditions:
- Work read-only. Do not create, modify, rename, or delete any
   file.
- Inspect only files under the current directory. Do not read
   parent or sibling directories.
- Do not use the network, web search, or external repositories.
- Do not rely on status labels, prior confidence assessments, or
   assertions of correctness. Check the mathematics yourself.
- Treat TARGET.md as identifying the claim under review, not as
   evidence that the claim is correct.

Task:
```

```
1. Inspect all relevant material available in the current
   directory.
2. Reconstruct the target claim and every load-bearing step that
   the supplied material gives for it.
3. Try actively to falsify the argument. Check quantifiers and
   parameter regimes, uses of induction or recursion, probability
    and error accounting, worst-case query cost, reductions,
   normalization conventions, compatibility with known upper or
   lower bounds stated in the supplied materials, and whether
   computational evidence supports only finite cases or the
   claimed asymptotic statement.
4. Report each concrete issue with a stable finding ID, severity (
   FATAL, MAJOR, MINOR, or NOTE), an exact file/section/line or
   record locator when possible, the mathematical consequence,
   and the smallest plausible repair. Distinguish a proved error
   from a missing justification.
5. Give an overall verdict: ACCEPT, ACCEPT WITH MINOR REVISIONS,
   MAJOR REVISION, or REJECT.

Use this final format:

# Reconstructed claim and argument
# Findings
## <ID> -- <severity>: <short title>
- Locator:
- Type: proved error | missing justification | presentation issue
   | note
- Analysis:
- Consequence:
- Smallest plausible repair:
# What was checked without finding an error
# Overall verdict
# Material-use summary
List the files or records that materially contributed to the audit
    and briefly state how each was used.
```

D Detailed Node-Record Structure

Listing 3 expands the compact record shown in Section 2. It includes the implementation fields used to bind external assumptions, independent reviews, linked Lean formalizations, unfinished tasks, and material state changes to a claim. Optional blocks are omitted from actual node files when they are unused.

Listing 3: Expanded canonical node record; optional fields are omitted when unused.

```
id: stable.problem.claim
role: <proof | obstruction | outcome | tool | constraint>     #
   optional
aliases: [<short name>]                                       #
   optional
claim: <precisely scoped claim>
assumptions:                                                  #
   optional; direct proof only
  <assumption-id>:
    kind: <external-theorem | model-fact>
    statement: <statement and scope>
    source:
      path: <file>
      kind: text
      locator: <location>
routes:                                                       #
   optional; OR alternatives
  <route-id>:
    premises: [<node B>, <node C>]                            #
   jointly required
    assumptions:
      <assumption-id>:
```

```
        kind: <external-theorem | model-fact>
        statement: <statement and scope>
        source:
          path: <file>
          kind: text
          locator: <location>
    implication: <why B and C imply this claim>
    proof:
      sketch: <argument>
      sources: [<linked artifact>]
    assessment:
      level: <evidence level>
      summary: <scoped judgment>
      basis:
        evidence: [<evidence>]
        reviews: [<review with locator and reviewed_hash>]
        counterexamples: [<falsifier>]
assessment:
  level: <evidence level>
  summary: <judgment of this node's claim>
  basis:
    routes: [<route-id>]
    evidence: [<evidence>]
    reviews: [<external review>]
    runs: [<run>]
  caveats: [<explicit limit>]
disposition:                                                    #
   optional
  state: <PARKED | DEAD>
  reason: <why stopped>
  revive: <condition>
formalization:                                                  #
   optional
  lean:
    module: Project.Module
    file: <file>
    claim: Claim
    state: <stated | proved | composition-checked>
    claim_hash: sha256:<node claim>
    source_claim_hash: sha256:<Lean claim>
    sync_state: <current | stale>
    synced: YYYY-MM-DD
    theorems: [<TheoremName>]
    rel: [<assumption or scope limit>]
    build: <linked build evidence>
frontier:
  <task-id>:
    task: <next action>
    kind: <task | gate | review>
    since: YYYY-MM-DD
events:
  <event-id>:
    type: <event type>
    date: <date>
    from: <previous level>                    # only for level-
   change
    to: <new level>                           # only for level-
   change
    route: <route-id>                         # for a route-
   specific event
    summary: <change>
    sources: [<supporting record>]
```

The record contains the claim, alternative routes, current assessment, linked evidence and reviews, remaining work, and append-only events. Route premises are stored in the route that uses them; the repository then computes which other claims depend on each node and generates the graph views. Large derivations, code, logs, certificates, and linked Lean formalizations remain in separate files, but

the assessments they support link directly to them. A node is therefore the entry point for auditing one claim or handing its remaining work to another worker.

A node record begins with a stable `id` and a precisely scoped `claim`. The optional `role` classifies the claim as a proof, obstruction, outcome, tool result, or constraint, while `aliases` records alternative names used elsewhere in the repository. Established external results used by a direct proof are recorded in a node-level `assumptions` block. If a node instead contains one or more proof routes, each external assumption is attached to every route that uses it. The two placements are mutually exclusive, preventing the same assumption from being recorded ambiguously at both levels. The `routes` field contains the proposed ways of establishing the claim, including their premises, assumptions, implication arguments, proofs, and supporting assessments.

The top-level `assessment` summarizes the current judgment of the node's claim and links directly to the routes, evidence, computations, reviews, and counterexamples that support it. External review reports are stored in the project's `reviews/` directory. The `assessment.basis.reviews` field links to the report and identifies the relevant finding. Any completion-level review records a `reviewed_hash` of the exact claim or route implication that was reviewed. If that mathematical statement later changes, the previous review no longer supports the modified version. A link marked as `node-evidence` is only a non-load-bearing cross-reference; any claim required by the proof must instead be registered as a route premise. Any limitations are stated explicitly as caveats.

The optional `disposition` records whether work is temporarily `PARKED` or has been declared `DEAD`, together with the reason and, when applicable, a condition for resuming it. This field records workflow state and does not change the evidence level.

The `formalization` block identifies the linked Lean module, file, formal claim, theorem names, synchronization state, and build evidence. The `claim_hash` binds it to the current node claim, while `source_claim_hash` binds it to the claim text quoted by the Lean source. The `rel` field lists formal assumptions and scope limitations, and `synced` records the synchronization date. Finally, `frontier` lists unfinished tasks, while `events` records dated changes together with their supporting materials. For a `level-change` event, `from` and `to` give the previous and new evidence levels; these fields are omitted for other event types. The `route` field identifies the affected route when an event concerns a route rather than the node itself.

E Mathematical Details for Minimal-Alphabet Set Equality

We refer to Set Equality as the *negative case* because the project did not attain its principal objective, namely, constructing an explicit adversary witness that achieves the optimal $\Omega(n^{1/3})$ scaling at the minimum alphabet size $q = 2n$. The project nevertheless established an unconditional $\Omega(n^{1/6})$ lower bound and isolated four unresolved proof obligations that, together with a separately recorded supporting layer, are sufficient for recovering the optimal exponent. We develop the relevant mathematics below so that these outcomes can be assessed independently of the accompanying project records.

E.0 Quantum information and quantum computational complexity

An m-qubit register is described by the Hilbert space $\mathcal{H}_m = (\mathbb{C}^2)^{\otimes m}$. A pure state is a unit vector $|\psi\rangle \in \mathcal{H}_m$, with physically equivalent states differing only by a global phase. More generally, a mixed state is a density operator $\rho \succeq 0$ satisfying $\mathrm{Tr}(\rho) = 1$. A closed-system operation is represented by a unitary U, so that $\rho \mapsto U\rho U^\dagger$, while noise and other open-system dynamics are represented by completely positive trace-preserving maps. A measurement with outcome set $\mathcal{A}$ is specified by a positive-operator-valued measure $\{M_a : a \in \mathcal{A}\}$ satisfying $M_a \succeq 0$ and $\sum_{a\in\mathcal{A}} M_a = I$. The Born rule gives

$$\Pr[a \mid \rho] = \mathrm{Tr}(M_a\rho) .$$

Tensor-product states describe independently prepared registers; states that cannot be written as convex combinations of product states are entangled.

Quantum computational complexity measures the resources needed to implement such transformations and measurements. A decision problem is a language $L \subseteq \{0,1\}^*$, and BQP consists of the languages decided with bounded error by polynomial-time uniform families of polynomial-size quantum circuits. The constant success threshold is conventionally $2/3$ and can be amplified to $1 - \delta$

by repetition with only logarithmic overhead in $1/\delta$. Circuit size, depth, qubit count, communication, and oracle access lead to different resource measures; a speedup in one measure need not imply an equally efficient implementation in every other measure.

This paper uses the black-box, or query, model. For an input $z \in [q]^N$, a T-query algorithm has the form

$$U_T O_z U_{T-1} O_z \cdots U_1 O_z U_0 |0\rangle,$$

followed by a two-outcome measurement. The unitaries $U_0, \ldots, U_T$ are independent of z, while all input dependence enters through the oracle O_z. For a Boolean promise problem $f : \mathcal{D} \to \{0,1\}$, with $\mathcal{D} \subseteq [q]^N$, the bounded-error quantum query complexity $\mathrm{Q}(f)$ is the least T for which the measurement returns $f(z)$ with probability at least $2/3$ for every $z \in \mathcal{D}$. Because arbitrary input-independent computation is free, this model isolates the information about z that must be acquired to solve the problem.

Upper bounds construct an explicit sequence of inter-query operations. Lower bounds show that no such sequence can succeed below a given query count. Polynomial-method lower bounds constrain the degree of the acceptance probability as a function of the input Beals et al. (2001); adversary methods instead bound the progress that one query can make in distinguishing positive from negative inputs Høyer et al. (2007); Lee et al. (2011). The latter formulation is central below because it produces matrix witnesses that can be inspected as explicit certificates. Set Equality and k-Sum are promise problems whose query complexity depends on collisions, symmetry, and alphabet size. The next subsection specializes the oracle model to Set Equality, while Appendix F develops the corresponding k-Sum problem.

E.1 The problem and the constructive gap

A quantum query algorithm accesses an input string $z = (z_1, \ldots, z_N) \in [q]^N$, where $[q]$ denotes a q-symbol alphabet, through an oracle rather than reading the string in full. On a computational-basis state, the oracle acts as

$$O_z |i, b, w\rangle = |i, b + z_i \pmod q, w\rangle \ , \tag{1}$$

where i specifies the queried position, b is the response register, and w is the workspace register. By linearity, a single oracle call may query a superposition of positions. The query model counts only oracle calls; input-independent quantum computation performed between them is treated as free. For a promise problem f, its bounded-error quantum query complexity $\mathrm{Q}(f)$ is the minimum number of oracle calls required to return the correct value with probability at least $2/3$ on every valid input.

Set Equality takes two injective lists $x, y : [n] \to [q]$, accessed through the concatenated input $z = (x, y) \in [q]^{2n}$. For a list x, let $\mathrm{im}(x) = \{x_i : i \in [n]\}$ denote the set of values appearing in x, and define $\mathrm{im}(y)$ similarly. The promise is that the two ranges are either equal or disjoint:

$$\mathrm{im}(x) = \mathrm{im}(y) \qquad \text{or} \qquad \mathrm{im}(x) \cap \mathrm{im}(y) = \varnothing \ . \tag{2}$$

We denote this Boolean promise problem by $\textsc{Set Equality}_{n,q}$, assigning value 1 to equal-range inputs and 0 to disjoint-range inputs. Because two disjoint injective lists require $2n$ distinct symbols, both promise cases are nonempty only when $q \geq 2n$. Thus, $q = 2n$ is the smallest alphabet size for which the problem is nontrivial.

Theorem E.1 (Known query complexity)**.** *For every $q \geq 2n$,*

$$\mathrm{Q}(\textsc{Set Equality}_{n,q}) = \Theta(n^{1/3}) \ . \tag{3}$$

Zhandry's polynomial-method analysis gives the lower bound in Theorem E.1, whereas a quantum collision-finding algorithm, adapted to detect a value shared by the two lists, gives the upper bound Brassard et al. (1998); Zhandry (2015). Thus, the query complexity is already settled, including at the minimum alphabet size $q = 2n$. What remains open here is constructive: can we explicitly construct and directly verify an adversary matrix that achieves the optimal $\Theta(n^{1/3})$ scaling at $q = 2n$?

The negative-weight adversary method is well suited to this question because it expresses a query lower bound as an explicit matrix certificate. Consider a Boolean promise problem $f : \mathcal{D} \to \{0,1\}$, where $\mathcal{D} \subseteq [q]^N$ is the set of valid inputs, and define

$$\mathcal{D}_1 = f^{-1}(1), \qquad \mathcal{D}_0 = f^{-1}(0) \ .$$

An adversary matrix is a nonzero real matrix $\Gamma \in \mathbb{R}^{\mathcal{D}_1 \times \mathcal{D}_0}$, with rows indexed by positive inputs $u \in \mathcal{D}_1$ and columns indexed by negative inputs $v \in \mathcal{D}_0$. For each queried position $i \in [N]$, define the disagreement matrix Δ_i, with the same row and column indices, by

$$(\Delta_i)_{u,v} = \begin{cases} 1, & u_i \neq v_i, \\ 0, & u_i = v_i. \end{cases} \tag{4}$$

The negative-weight adversary value is

$$\mathrm{Adv}^{\pm}(f) = \max_{\substack{\Gamma \in \mathbb{R}^{\mathcal{D}_1 \times \mathcal{D}_0} \\ \Gamma \neq 0}} \frac{\|\Gamma\|}{\max_{i \in [N]} \|\Gamma \circ \Delta_i\|}\,, \tag{5}$$

where $\circ$ denotes entrywise multiplication and $\|\cdot\|$ is the operator norm. The numerator measures the global separation represented by Γ, whereas the denominator measures the largest portion of that separation revealed by any single query. The method is tight:

$$\mathrm{Q}(f) = \Theta(\mathrm{Adv}^{\pm}(f))\,, \tag{6}$$

with universal constants Høyer et al. (2007); Lee et al. (2011). Thus, tightness guarantees the existence of an adversary matrix with the optimal asymptotic scaling, but it neither identifies such a matrix nor supplies the norm estimates needed to verify it.

Belovs and Rosmanis gave an explicit representation-theoretic construction for Set Equality when $q = \Omega(n^2)$ Belovs & Rosmanis (2018). Their starting point is an ambient matrix with permutation symmetry whose row blocks are indexed by matchings between the two lists and whose columns include all words in $[q]^{2n}$. This ambient space contains illegal inputs—repeated symbols on the disjoint side and noninjective labels on the equal side—but its symmetry supports an orthogonal level decomposition

$$\widetilde{\Gamma} = \sum_{k=0}^{K-1} \alpha_k \overline{W}_k, \qquad K = \lfloor n^{1/3} \rfloor, \qquad \alpha_k = K - k. \tag{7}$$

The level k has a concrete interpretation even without the full representation-theoretic notation. Level zero is the constant component, which treats all legal inputs uniformly. Higher levels record fluctuations that depend nontrivially on k alphabet coordinates. The operator $\overline{W}_k$ aggregates the contribution of this degree across all matchings between the two lists. This grading makes different levels orthogonal before a query is applied and sharply limits which levels a query can couple.

For a queried position $i \in [2n]$, applying the disagreement mask Δ_i to $\widetilde{\Gamma}$ couples only equal or adjacent representation levels. The adjacent-level terms appear through the coefficient differences

$$\alpha_{k-1} - \alpha_k = 1\,.$$

Together with the ambient same-level estimates and the orthogonality of the level decomposition, this gives

$$\max_{i \in [2n]} \left\|\widetilde{\Gamma} \circ \Delta_i\right\| = O(1)\,,$$

uniformly in n.

The legal adversary matrix Γ is obtained by retaining only the rows and columns of $\widetilde{\Gamma}$ that correspond to legal positive and negative inputs. Because this restriction is a compression by coordinate projections, it cannot increase the masked norm:

$$\left\|\Gamma \circ \Delta_i\right\| \leq \left\|\widetilde{\Gamma} \circ \Delta_i\right\|\,.$$

The same restriction can, however, substantially decrease the numerator $\|\Gamma\|$. When $q = \Omega(n^2)$, sufficiently many ambient rows and columns survive for the original analysis to retain $\|\Gamma\| = \Omega(K)$. At the minimum alphabet size $q = 2n$, every legal disjoint input is a permutation of the complete alphabet $[2n]$. Such inputs constitute only

$$\frac{(2n)!}{(2n)^{2n}}$$

of all ambient strings in $[2n]^{2n}$, an exponentially vanishing fraction. Thus, a survival-probability argument cannot establish the required $\Omega(K)$ numerator in this regime.

Our case study therefore asks whether the same linearly tapered construction can be restricted directly to the legal $q = 2n$ promise and analyzed without assuming a constant surviving fraction. Our analysis yields two distinct conclusions for the resulting explicit legal matrix: unconditionally, it achieves an $\Omega(n^{1/6})$ adversary ratio; conditionally, it achieves the optimal $\Omega(n^{1/3})$ ratio if four representation-theoretic conditions hold. These conditions isolate the remaining gap without changing the unconditional result.

E.2 An unconditional $\Omega(n^{1/6})$ lower bound

Let W_k^{leg} denote the legal-input restriction of the aggregated level-k matching operator, and let Q_k^{leg} denote the corresponding position-defect projector. Their product is the legal level-k atom. With $K = \lfloor n^{1/3} \rfloor$, let $(t)_+ := \max\{t, 0\}$ and set $\alpha_k = (K - k)_+$. Define

$$\Gamma_n = \frac{1}{(2n)!} \sum_{k=0}^{K-1} \alpha_k W_k^{\mathrm{leg}} Q_k^{\mathrm{leg}}. \tag{8}$$

The normalization uses the number $(2n)!$ of legal disjoint inputs. Equation (8) is not a new matrix ansatz: it is exactly the linear-taper witness in Equation (7), restricted to the legal minimum-alphabet promise.

Theorem E.2 (Unconditional case-study result (restated))**.** *The explicit matrix Γ_n satisfies*

$$\mathrm{Adv}^{\pm}(\textsc{Set Equality}_{n,2n}) \geq \left(\frac{2}{\sqrt{6}} - o(1) \right) n^{1/6}. \tag{9}$$

We first bound the numerator. Let $\mathbf{1}_N$ and $\mathbf{1}_Y$ be normalized constant vectors on the legal negative and positive input sets. The level-zero atom is the normalized all-ones map and sends $\mathbf{1}_N$ to $\mathbf{1}_Y$. Every higher-level atom is orthogonal to the constant sector. Therefore

$$\Gamma_n \mathbf{1}_N = \alpha_0 \mathbf{1}_Y = K \mathbf{1}_Y, \qquad \|\Gamma_n\| \geq K. \tag{10}$$

This direct calculation is what replaces the large-alphabet survival argument: although legal negative inputs are sparse, the numerator is still of order $n^{1/3}$.

We next analyze the effect of the query mask. Fix a queried position i. Applying Δ_i splits each level into same-level and neighboring-level terms. The neighboring coefficients are $\alpha_{k-1} - \alpha_k = 1$, so all already-controlled pieces contribute only $O(1)$. The growing contribution is a family of legal lower blocks. Write $L_{i,d}$ for the block produced by the transition from level $d + 1$ to level d. Different values of d lie in orthogonal sectors, and the unconditional legal-restriction estimate is

$$\|L_{i,d}\| \leq \left(1 + O(n^{-1/3})\right) \frac{\sqrt{6}\, d}{\sqrt{n - 2d - 1}}, \qquad 0 \leq d < K. \tag{11}$$

The factor d arises from multiplying two losses of order $\sqrt{d}$, one from each legal restriction. This is the only step that loses a growing factor.

It remains to bound the denominator. Orthogonality across d and the standard factor-two mask estimate give

$$\begin{aligned} \|\Gamma_n \circ \Delta_i\| &\leq O(1) + 2 \max_{0 \leq d < K} \alpha_{d+1} \|L_{i,d}\| \\ &\leq O(1) + (2 + o(1)) \max_{0 \leq d < K} \frac{\sqrt{6}\, d(K - d - 1)}{\sqrt{n - 2d - 1}}. \end{aligned} \tag{12}$$

Because $d < K = O(n^{1/3})$, the square-root denominator is $(1 + o(1))\sqrt{n}$ uniformly. The remaining quadratic obeys

$$d(K - d - 1) = \frac{(K - 1)^2}{4} - \left(d - \frac{K - 1}{2} \right)^2.$$

Its integer maximum is $(K - 1)^2/4 + O(1)$, attained near $d = (K - 1)/2$. Hence

$$\max_i \|\Gamma_n \circ \Delta_i\| \leq \left(\frac{\sqrt{6}}{2} + o(1) \right) n^{1/6} + O(1).$$

Combining this estimate with $K = (1 - o(1))n^{1/3}$ in Equation (10) proves Theorem 4.1. The weaker exponent therefore has a precise source: the current proof pays $K^2/\sqrt{n} = \Theta(n^{1/6})$ in the denominator by bounding the two legal restrictions independently.

E.3 Four sufficient conditions for an $\Omega(n^{1/3})$ lower bound

Equation (10) already gives the numerator required for an $\Omega(n^{1/3})$ witness. It therefore remains to prove that the masked denominator is $O(1)$. The independently bounded lower block in Equation (11) is the only obstruction.

The required improvement can be seen at the level of exponents. The current block estimate is of order $j/\sqrt{n}$. Multiplied by a taper coefficient of order K, it becomes $Kj/\sqrt{n}$, whose maximum for $j = \Theta(K)$ is $K^2/\sqrt{n} = \Theta(n^{1/6})$. A correlated estimate of order $\sqrt{j/n}$ would instead contribute $K\sqrt{K/n} = \Theta(1)$. The following four representation-theoretic conditions are sufficient to justify this square-root improvement for the concrete legal block.

Fix an active level j with $1 \le j < K$. After masking a query, let $C_{j+1\to j}$ denote the legal block that moves from representation level $j+1$ to level j. Symmetry decomposes this block into sectors indexed by $s \in \mathcal{S}_j$. For each sector, let $\mathcal{M}_s \cong \mathbb{C}^{c_s}$ be its multiplicity space and let $\mathcal{R}_s \cong \mathbb{C}^{m_s}$ be the remaining representation carrier. Thus, c_s and m_s are the dimensions of these two spaces. Let $r_s > 0$ denote the representation-theoretic branching coefficient for the transition $j+1 \to j$ in sector s, and let $\omega_j > 0$ denote the scalar coefficient contributed by the legal-input restriction to this transition. We write I_{c_s} and I_{m_s} for the identity operators on $\mathcal{M}_s$ and $\mathcal{R}_s$, respectively, and let C_s denote the restriction of $C_{j+1\to j}$ to sector s. Direct evaluation of the scalar legal-restriction coefficient gives

$$0 < \omega_j \le \frac{j}{2(n-j)} \tag{13}$$

for $1 \le j < K$. This estimate has the required decay, provided that the following conditions hold uniformly over all active levels and sectors.

Conjecture E.3 (Representation-theoretic conditions (restated))**.** *For every $1 \le j < K$ and every $s \in \mathcal{S}_j$, the following statements hold.*

H1: *The two contributions to the block, combined with their corresponding weights and normalized to a Hermitian operator $\widehat{M}_s$ on $\mathcal{M}_s$, satisfy*

$$0 \preceq \widehat{M}_s \preceq \frac{15}{4}\,\omega_j I_{c_s}.$$

H2: *The canonical Gram operator on $\mathcal{R}_s \otimes \mathcal{M}_s$ has the tensor-product form*

$$\mathsf{G}_s = I_{m_s} \otimes \left[\frac{r_s}{\omega_j}\widehat{M}_s\right].$$

H3: *The sector restriction C_s and the canonical Gram operator satisfy the operator identity*

$$C_s^* C_s = (r_s\omega_j)\mathsf{G}_s.$$

H4: *The sector decomposition is orthogonal and exhaustive. Under the corresponding orthogonal decompositions of the domain and codomain,*

$$C_{j+1\to j} = \bigoplus_{s\in\mathcal{S}_j} C_s, \qquad \|C_{j+1\to j}\| = \max_{s\in\mathcal{S}_j}\|C_s\|.$$

The four unresolved mathematical steps are H1–H4. To formulate and combine them, the proof also uses a separate representation-theoretic foundation. This foundation identifies the legal sectors, supplies the symmetry identities and coordinate maps used in the argument, and connects the four obligations to the final adversary bound. It is not an additional open conjecture, but the conditional result still assumes that this supporting layer is valid throughout the required parameter range. Its current evidence for general $c \ge 2$ remains limited, as recorded in the corresponding node.

Each condition addresses a different point in the argument. H1 is the only positive-semidefinite inequality and bounds the two contributions jointly. H2 preserves the representation-carrier dimension; omitting I_{m_s} would equate operators acting on different spaces. H3 identifies the normalized Gram calculation with the sector restriction in the legal witness. H4 ensures that the sector decomposition is complete.

The four statements have a short and fully checkable consequence. H1 and H2 give $\|\mathsf{G}_s\| \le (15/4)r_s$. Combining this with H3 and Equation (13) yields

$$\|C_s\| \le \sqrt{\frac{15}{8}}\, r_s \sqrt{\frac{j}{n-j}}. \tag{14}$$

H4 then promotes the sector estimate to

$$\|C_{j+1\to j}\| \le \sqrt{\frac{15}{8}}\, \rho_j \sqrt{\frac{j}{n-j}}, \qquad \rho_j := \max_{s\in\mathcal{S}_j} r_s. \tag{15}$$

For later use, set $\rho_0 := 1$. Applying the hook-length formula to the branching coefficients gives the following uniform estimate for $1 \le j < K$:

$$1 \le \rho_j \le \exp\left(\frac{j(j-1)}{2(2n-j+1)} + \frac{j}{2n-2j+1}\right) = 1 + O(n^{-1/3}). \tag{16}$$

Consequently ρ_j contributes no growing power of n.

It remains to verify that this local improvement is strong enough globally. Substituting Equation (15) into the complete query-mask decomposition gives

$$\|\Gamma_n \circ \Delta_i\| \le 2 \max_{0\le d<K} \rho_d \left[1 + \sqrt{2}\,\alpha_d \sqrt{\frac{d}{n}} + \mathbf{1}_{\{d\ge 1\}} \frac{\sqrt{6}\,\alpha_d}{\sqrt{n-2d+1}} + \mathbf{1}_{\{d\ge 1\}} \sqrt{\frac{15}{8}}\,\alpha_{d+1}\sqrt{\frac{d}{n-d}}\right]. \tag{17}$$

Here $\mathbf{1}_{\{d\ge 1\}}$ is an indicator. The first term is the tapered neighbor-level contribution. The second is the already-controlled same-level channel, the third is $O(K/\sqrt{n}) = o(1)$, and the fourth is the improved lower block supplied by Conjecture E.3.

For the two nonconstant terms that remain, set $x = d/K$. Since $K^3/n \to 1$, both converge uniformly to a constant multiple of

$$f(x) = (1-x)\sqrt{x}.$$

The maximum occurs at $x = 1/3$ and equals $2/(3\sqrt{3})$. Hence

$$\|\Gamma_n \circ \Delta_i\| \le 2(1+\mathcal{M}) + o(1), \qquad \mathcal{M} = \frac{2}{3\sqrt{3}}\left(\sqrt{2} + \sqrt{\frac{15}{8}}\right) = \frac{4\sqrt{6}+3\sqrt{10}}{18}. \tag{18}$$

Proposition E.4 (Conditional consequence)**.** *If Conjecture E.3 holds and the supporting representation-theoretic foundation recorded by node* `SE.denom.Phi.legal.found` *is valid throughout its stated scope, then the same explicit matrix Γ_n satisfies*

$$\mathrm{Adv}^{\pm}(\textsc{Set Equality}_{n,2n}) \ge \left(\frac{9}{18+4\sqrt{6}+3\sqrt{10}} - o(1)\right) n^{1/3}.$$

Indeed, divide the numerator $K = (1-o(1))n^{1/3}$ by Equation (18). Thus, the optimal $\Omega(n^{1/3})$ scaling follows if conditions H1–H4 and the stated supporting foundation hold, whereas Theorem 4.1 remains unconditional.

E.4 Review-driven corrections

This subsection describes three findings of one external AI review round on the Set Equality argument: two invalid steps and one gap in the reduction. The review procedure is not assumed to be complete. For each finding, we state the invalid or missing step, its correction, and the conclusions that remain unaffected. We take them in the order in which the review reported them.

Invalid projector intertwining identity. The review found an error in the restriction of the ambient witness to legal inputs. Let J_N denote coordinate restriction from ambient words to legal negative inputs. The rejected argument asserted

$$J_N P_k^{\mathrm{amb}} = P_k^{\mathrm{leg}} J_N,$$

where P_k^{amb} contains both a position-defect projector and an exact tensor-weight projector, and P_k^{leg} denotes its proposed legal counterpart. Legal inputs are invariant under position permutations, but this invariance does not imply that exact-weight projection commutes with coordinate restriction. For instance, when $n = 1$, $q = 2$, and $k = 0$, the delta function supported on the illegal word $(1, 1)$ is annihilated by J_N, whereas its exact-weight-zero projection restricts to a nonzero constant function. Thus, the displayed intertwining identity fails.

The proof requires only a weaker product-level identity. Let $W_{\mathrm{se},k}$ denote the ambient level-k matching operator and let $\mathcal{R}$ denote normalized restriction to legal positive and negative inputs. Separating the pure position projector from the exact-weight projector gives

$$\mathcal{R}\big(W_{\mathrm{se},k} P_k^{\mathrm{amb}}\big) = \frac{1}{(2n)!} W_k^{\mathrm{leg}} Q_k^{\mathrm{leg}}.$$

The pure position projector commutes with legal restriction, while $W_{\mathrm{se},k}$ absorbs the exact-weight projector. This product-level correction leaves the legal witness Γ_n, its normalization, and all numerator and denominator estimates unchanged.

Missing tensor-product lift. The same review found a dimensional inconsistency in the candidate optimal-order argument. The argument equated the canonical Gram operator G_s, which acts on the full sector carrier $\mathcal{R}_s \otimes \mathcal{M}_s$, with the normalized multiplicity-space operator $(r_s/\omega_j)\widehat{M}_s$, which acts only on $\mathcal{M}_s$. The two sides therefore act on spaces of dimensions $m_s c_s$ and c_s, respectively, and cannot be identified without lifting the multiplicity-space operator to the full carrier. The corrected relation is precisely condition H2 in Conjecture E.3:

$$\mathsf{G}_s = I_{m_s} \otimes \left[\frac{r_s}{\omega_j}\widehat{M}_s\right].$$

Condition H3 separately relates this full-carrier Gram operator to the sector restriction $C_s^* C_s$, while H4 requires the sector decomposition to be orthogonal and exhaustive. This correction invalidates the earlier optimal-order route but does not affect the numerator estimate in Equation (10), the unconditional lower-block estimate in Equation (11), or Theorem 4.1.

Missing exhaustiveness obligation. The review also found that the reduction as then recorded was incomplete. With H1–H3 in their corrected forms, the argument establishes the per-sector bound in Equation (14), but the passage from that bound to the global lower block in Equation (15) additionally requires the sector decomposition of $C_{j+1\to j}$ to be orthogonal and exhaustive, with the multiplicity lift applied uniformly. No registered claim covered this step. It was registered as a new open node and is condition H4 of Conjecture E.3. This finding invalidated no established estimate; it enlarged the set of unresolved obligations from three to four.

Together, the three findings illustrate a limited but concrete role for the dependency graph. The external AI reviewer, not the graph, identified the errors and the gap. Once each finding was attached to the corresponding claim, however, the graph made its downstream consequences directly visible: the affected optimal-order route could be revised without demoting the unconditional result. The four unresolved mathematical requirements for the optimal $\Omega(n^{1/3})$ scaling are conditions H1–H4 in Conjecture E.3; the conditional consequence also retains the separately recorded representation-theoretic foundation as an explicit supporting premise.

F Mathematical Details for Small-Alphabet k-Sum

This section establishes upper and lower bounds for cyclic k-Sum over small alphabets. For cyclic 2-Sum, we prove an $\Omega(\sqrt{N}\, q^{1/6})$ lower bound in the parameter ranges specified in Theorem 5.1, together with an $O_\varepsilon(N^{1/2+\varepsilon} q^{1/4})$ upper bound. For every fixed $k \geq 3$, we also obtain an $\Omega_k(\sqrt{N}\, q^{1/6})$ lower bound over a linear-size alphabet regime. These results do not determine the full small-alphabet tradeoff: for 2-Sum, in particular, the exponents $1/6$ and $1/4$ leave a gap in the dependence on q.

Let k, N, and q be integers satisfying $2 \leq k \leq N$ and $q \geq 2$, and write $[N] = \{1, \ldots, N\}$ and $\mathbb{Z}_q = \mathbb{Z}/q\mathbb{Z}$. For a fixed target $t \in \mathbb{Z}_q$, cyclic k-Sum, denoted $k\text{-Sum}_{\mathbb{Z}_q,t,N}$, is the Boolean function

on inputs $x = (x_1, \ldots, x_N) \in \mathbb{Z}_q^N$ that evaluates to 1 if and only if there exist pairwise distinct indices $i_1, \ldots, i_k \in [N]$ such that

$$x_{i_1} + \cdots + x_{i_k} = t \pmod{q}. \tag{19}$$

The target t is known to the algorithm, while the coordinates of x are accessed by queries. When $q \geq N^k$, the negative-weight adversary method gives the tight $\Theta(N^{k/(k+1)})$ complexity Belovs & Špalek (2013); Childs & Eisenberg (2005). At constant q, repeated values can instead lead to Grover-scale behavior. The open small-alphabet problem is to determine how the complexity interpolates between these regimes as a function of both N and q Špalek (2013); Rosmanis (2014).

F.1 Main results

The following theorem collects the small-alphabet bounds for cyclic k-Sum proved in this section.

Theorem F.1 (Small-alphabet bounds for cyclic k-Sum (restated))**.** *The following statements hold.*

(i) *If $q \geq 5$ is odd, $t \in \mathbb{Z}_q$, and $q \leq 2N - 1$ for even N or $q \leq 2N - 3$ for odd N, then*

$$\mathrm{Q}(2\text{-}\mathit{Sum}_{\mathbb{Z}_q,t,N}) = \Omega(\sqrt{N}\, q^{1/6}).$$

Consequently the tight $\Theta(N^{2/3})$ regime for cyclic 2-Sum begins already at $q = \Theta(N)$, whereas the previously used large-alphabet construction was analyzed under the sufficient condition $q \geq N^2$ Belovs & Špalek (2013).

(ii) *For every $\varepsilon > 0$, every $q \geq 2$, and every $t \in \mathbb{Z}_q$,*

$$\mathrm{Q}(2\text{-}\mathit{Sum}_{\mathbb{Z}_q,t,N}) = O_\varepsilon(N^{1/2+\varepsilon} q^{1/4}).$$

(iii) *Fix $k \geq 3$ and put $q_0(k) = 2k^2 + 2k - 5$. There are constants $C_k, c_k > 0$ such that, for all sufficiently large N, every q satisfying $q_0(k) \leq q \leq C_k N$, and every $t \in \mathbb{Z}_q$,*

$$\mathrm{Q}(k\text{-}\mathit{Sum}_{\mathbb{Z}_q,t,N}) = \Omega_k(\sqrt{N}\, q^{1/6}).$$

Moreover, for all sufficiently large N, every $q \geq c_k N$, and every $t \in \mathbb{Z}_q$,

$$\mathrm{Q}(k\text{-}\mathit{Sum}_{\mathbb{Z}_q,t,N}) = \Omega_k(N^{2/3}).$$

For growing q, part (iii) is super-Grover, but it does not attain the conjectured $N^{k/(k+1)}$ scale for $k \geq 3$. Likewise, parts (i) and (ii) leave the exponent of q between $1/6$ and $1/4$. Thus, these bounds identify nontrivial small-alphabet regimes without determining the full dependence of the query complexity on N and q.

F.2 Lower bound: embedding small-range Claw

Let m and K be positive integers and, for every positive integer r, write $[r] = \{1, \ldots, r\}$. For two functions $f, g : [m] \to [K]$, the Claw problem asks whether $f(i) = g(j)$ for some pair (i, j). Ambainis, Balodis, and Iraids proved the lower bound below for $2 \leq K < m$ Ambainis et al. (2021), while a later preprint of Tani extends its range to $2 \leq K \leq 2m$ Tani (2024); the extension has not yet been peer reviewed, and parts (i) and (iii) rely on it above $K = m - 1$. Thus, throughout $2 \leq K \leq 2m - 1$, we may use

$$\mathrm{Q}(\mathrm{Claw}_{m \to K}) = \Omega(\sqrt{m}\, K^{1/6}), \qquad 2 \leq K \leq 2m - 1 \tag{20}$$

The central observation is that cyclic 2-Sum contains this problem without changing its query structure.

Lemma F.2 (Signed Claw embedding)**.** *If $q \geq 2K + 1$, define*

$$E(f, g) = \big(f(1), \ldots, f(m), -g(1), \ldots, -g(m)\big) \in \mathbb{Z}_q^{2m}.$$

Then $E(f, g)$ contains two distinct positions summing to 0 if and only if f and g contain a claw.

Proof. Two values from the first half have an integer sum in $[2, 2K]$, and two from the second half have a sum in $[-2K, -2]$. Since $q \geq 2K + 1$, neither can be 0 modulo q. A cross pair has sum $f(i) - g(j) \in [-(K-1), K-1]$, which vanishes modulo q exactly when $f(i) = g(j)$. □

Restricting 2-Sum to the image of E and relabeling each coordinate costs only a constant factor in the query model. Therefore

$$\mathrm{Q}(\text{2-Sum}_{\mathbb{Z}_q,0,2m}) = \Omega(\mathrm{Q}(\mathrm{Claw}_{m\to K}))\,. \tag{21}$$

Take $m = \lfloor N/2 \rfloor$ and suppose that q is odd. Set $K = (q-1)/2$, so $q \geq 5$ implies $K \geq 2$. If N is even, then $m = N/2$ and

$$K \leq 2m - 1 \quad \Longleftrightarrow \quad q \leq 2N - 1.$$

If N is odd, then $m = (N-1)/2$ and

$$K \leq 2m - 1 \quad \Longleftrightarrow \quad q \leq 2N - 3.$$

In the latter case, freeze the unused coordinate to 0; because no encoded coordinate is 0, this creates no new pair. Adding a fixed residue a to every input coordinate moves the target by $2a$. Since multiplication by 2 is invertible in $\mathbb{Z}_q$, the zero-target restriction covers every target. Combining Equations (20) and (21), together with $m = \Theta(N)$ and $K = \Theta(q)$, gives

$$\mathrm{Q}(\text{2-Sum}_{\mathbb{Z}_q,t,N}) = \Omega(\sqrt{N}\, q^{1/6})$$

through the range stated in Theorem 5.1(i). Moreover, for all sufficiently large N and every odd $q \geq N$, one may instead take $K = m - 1$. Then $2 \leq K \leq 2m - 1$ and $q \geq 2K + 1$, so the same reduction gives the q-independent lower bound $\Omega(N^{2/3})$ for every $t \in \mathbb{Z}_q$.

The reduction extends to every fixed $k \geq 3$. A Claw pair supplies only two of the k required positions. The remaining positions are frozen to a *core* whose sum is the target. Let

$$m = \left\lfloor \frac{N - (k-2)}{2} \right\rfloor, \qquad U = kK + 1.$$

Use $2m$ live coordinates $f(1), \ldots, f(m), -g(1), \ldots, -g(m)$ and freeze

$$c_1 = \cdots = c_{k-3} = U, \qquad c_{k-2} = t - (k-3)U. \tag{22}$$

The core sums to t. If $N - (k-2)$ is odd, one coordinate remains; park it at

$$v = t + (k-1)K + 1. \tag{23}$$

The intended witness consists of the complete core and one coordinate from each live half:

$$\sum_{r=1}^{k-2} c_r + f(i) - g(j) = t + f(i) - g(j) \pmod{q}. \tag{24}$$

It reaches t exactly when (i, j) is a claw.

For $k = 3$, the construction is especially transparent. There is one core coordinate, equal to t, and a valid witness consists of this coordinate plus a zero-sum cross pair. A candidate that omits the core must use three live values, whose total magnitude is at most $3K$. The target window begins beyond $3K$, so such a candidate cannot reach t. For general k, the repeated spacing value $U = kK + 1$ applies the same separation argument after any proper subset of the core is removed.

The separation scale $U = kK + 1$ makes the converse true. A sum of at most k live values has magnitude at most kK. If a candidate uses the whole core, it has room for exactly two live coordinates; same-half pairs are nonzero, and a cross pair succeeds only for a claw. If it omits a core coordinate, the distance to the target is at least U, which no live contribution can cancel. In the odd case, the value in Equation (23) similarly keeps every subset containing the parked coordinate away from the target. The required no-wrap inequalities hold for the consecutive target windows

$$\begin{aligned} W_{\text{even}} &= [(k-2)(kK+1),\ q - kK - 1], \\ W_{\text{odd}} &= [(k-2)(kK+1),\ q - 2(k-1)K - 2]. \end{aligned} \tag{25}$$

Thus, for t in the appropriate window, the restricted k-Sum instance is exactly $\mathrm{Claw}_{m\to K}$.

To cover arbitrary targets, use the translation identity

$$k\text{-Sum}_{\mathbb{Z}_q,t,N}(x) = k\text{-Sum}_{\mathbb{Z}_q,t+ka,N}(x + a\mathbf{1}). \tag{26}$$

Translation moves t within its residue class modulo $\gcd(k, q)$. It is therefore enough that the window in Equation (25) contain $\gcd(k, q)$ consecutive residues. The choice

$$K = \left\lfloor \frac{q - 2k + 1}{k^2 - 2} \right\rfloor, \qquad q_0(k) = 2k^2 + 2k - 5, \tag{27}$$

guarantees both this coverage and the no-wrap conditions. The threshold $q_0(k)$ is exactly the point at which the resulting Claw range first satisfies $K \geq 2$.

For $q_0(k) \leq q \leq N$, we may take $C_k = 1$. Indeed, $m = \Theta(N)$, $K = \Theta_k(q)$, and

$$K \leq \frac{q}{k^2 - 2} \leq \frac{N}{k^2 - 2} \leq 2m - 1$$

for all sufficiently large N. Substituting into Equation (20) gives

$$\mathrm{Q}(k\text{-Sum}_{\mathbb{Z}_q,t,N}) = \Omega_k(\sqrt{N}\, q^{1/6}). \tag{28}$$

For the large-alphabet regime, take $K = m - 1 = \Theta(N)$. The target-window conditions are satisfied whenever $q \geq (k^2 - 2)K + 2k - 1$. In particular, $q \geq k^2 N$ is sufficient, and hence one may take $c_k = k^2$ to obtain the lower bound $\Omega_k(N^{2/3})$.

The lower bounds follow from a query-preserving reduction rather than from a new adversary construction. Scale separation forces every accepting k-subset in the restricted domain to use the same frozen core, leaving a two-list Claw as the only live decision. Consequently, both the factor $q^{1/6}$ and the q-independent lower bound $N^{2/3}$ are inherited from the known small-range Claw bound, and this reduction alone cannot reach $N^{k/(k+1)}$ for $k \geq 3$.

F.3 Upper bound: adapting the small-range Claw algorithm

The upper bound in Theorem 5.1(ii) starts from the recursive small-range Claw algorithm of Ambainis, Balodis, and Iraids Ambainis et al. (2021). Their algorithm samples and compresses two separately indexed input lists. Cyclic 2-Sum instead has one list, and compression can map two virtual coordinates to the same original position. The adaptation below introduces a single residual set, removes the resulting fixed-point obstruction, and gives every counting and virtual-oracle subroutine a deterministic query cap. The sampling-and-compression recurrence is inherited from the Claw algorithm; these three modifications establish its soundness and worst-case query bound for cyclic 2-Sum.

For $i \geq 0$, let $\mathcal{A}_i(s, q, t)$ denote the depth-i algorithm on a list of length s. Each valid coordinate stores a value in $\mathbb{Z}_q$ and its original input index. We also allow an inactive symbol $\perp$, which never participates in a sum. Let $\mathrm{Q}_i(s, q)$ be the maximum number of queries that $\mathcal{A}_i(s, q, t)$ makes to its supplied list oracle, over all targets and valid inputs. The base algorithm $\mathcal{A}_0$ is the standard Johnson-graph subset-finding algorithm, with query cost $O(s^{2/3})$ Childs & Eisenberg (2005). For $i \geq 1$, $\mathcal{A}_i$ performs the compression step below and normally calls $\mathcal{A}_{i-1}$ on the compressed list. Inactive coordinates are ignored by every search and sum predicate. Lists of length less than two are answered directly. We write $\widetilde{O}(\cdot)$ for factors polynomial in $\log s$ at fixed recursion depth.

Define the complement involution and its fixed-point set by

$$\sigma(v) := t - v, \qquad F := \{v \in \mathbb{Z}_q : 2v = t\}.$$

The equation $2v = t$ has either no solution or $\gcd(2, q)$ solutions, so $|F| \leq 2$. These values can be computed without querying the input. For each $v \in F$, two Grover searches determine whether v occurs at two distinct positions, using $O(\sqrt{s})$ queries in total. If so, the algorithm accepts. Otherwise, every coordinate with value in F is excluded from subsequent residual sets. Hence two virtual coordinates carrying the same original index cannot form an accepted 2-Sum pair.

We now describe one recursive level. If $q > s^{2/3}$, the Johnson-graph algorithm costs $O(s^{2/3}) \leq O(s^{1/2} q^{1/4})$. We may therefore assume $q \leq s^{2/3}$ and define

$$\kappa := \log_s q, \qquad 0 \leq \kappa \leq \frac{2}{3}.$$

Choose a sampling exponent $\alpha \in [\max\{1/2, \kappa\}, 1]$.

1. *Remove self-complementary values.* Perform the fixed-point preprocessing above. If no repeated value in F is found, exclude all coordinates whose values lie in F.
2. *Sample and resolve sample-incident pairs.* Choose a uniformly random set $A \subseteq [s]$ of $\ell = \widetilde{O}(s^{\alpha})$ positions, query it, and store its values. Check all pairs inside A without further queries. Let $S := \{x_a : a \in A\} \setminus F$ and $t - S := \{t - v : v \in S\}$. A single Grover search over the full list then checks whether some $j \notin A$ satisfies $x_j \in t - S$, at a cost of $O(\sqrt{s})$ queries. Any reported pair is checked to have two distinct original indices. If no pair is found, define
$$B := \{j \in [s] : x_j \notin F \text{ and } x_j \notin S\}, \qquad b := |B|.$$
Except with small probability, every non-fixed value missed by the sample has multiplicity below $s^{1-\alpha}$. Consequently,
$$b \le q s^{1-\alpha}. \tag{29}$$
3. *Count and compress.* Membership in B can be tested with one query to the current list oracle. First perform a capped zero test. If $b = 0$, reject before introducing any expression containing $1/b$ or $1/h$. Otherwise, bounded approximate counting returns an integer h satisfying
$$b \le h \le 4b$$
on its successful event Brassard et al. (2002). Its successful cost is $\widetilde{O}(\sqrt{s/b})$, and its deterministic cap is $\widetilde{O}(\sqrt{s})$. Put
$$U := \lceil q s^{1-\alpha} \rceil.$$
If counting fails to return a positive integer or gives $h > 4U$, reject this run. Thus every continuing computation satisfies the deterministic bound $1 \le h \le 4U$, including computations arising from an inaccurate count.
Fix a sufficiently large absolute constant C and set
$$L := \lceil Ch \log s \rceil, \qquad K := \min\{s, \lceil C(s/h) \log s \rceil\}.$$
Choose L independent uniform permutations $\pi_1, \ldots, \pi_L$ of $[s]$ before invoking the recursive algorithm.
4. *Evaluate the virtual list and recurse.* On virtual address $u \in [L]$, search the first K positions of π_u for the first index $r_u \in B$. Return the pair consisting of x_{r_u} and its original index; if the prefix contains no member of B, return $\perp$. Using quantum minimum finding Dürr & Høyer (1996), we implement this total oracle with a deterministic cap $\widetilde{O}(\sqrt{K}) = \widetilde{O}(\sqrt{s/h})$. The value and index are copied to the answer registers, and all search workspace is uncomputed. If $q \le L^{2/3}$, invoke $\mathcal{A}_{i-1}(L, q, t)$ on this virtual list. If $q > L^{2/3}$, use the Johnson-graph algorithm on the virtual list instead. Map any returned coordinates back to the original input and verify their indices and values before accepting.

To see why compression preserves the problem, condition on good sampling and counting. In a uniform permutation, the first element belonging to B is uniform on B. A fixed residual position is therefore absent from all L virtual coordinates with probability at most
$$(1 - 1/b)^L \le e^{-L/b}.$$
Because L is h times a logarithmic factor and $h \ge b$, a union bound over all residual positions is small. Each permutation search is truncated after $K = \widetilde{O}(s/h)$ positions. The relation $h \le 4b$ ensures that each prefix contains a residual element with high probability. Thus every residual position appears in the truncated virtual list with high probability. The symbol $\perp$ makes the virtual oracle defined even when a prefix search fails. Conversely, two virtual coordinates representing the same original position can form a 2-Sum pair only when their common value lies in F. Such values were removed.

The exponent follows from balancing the sample cost against the recursive cost. One active level has the complete query bound
$$\mathrm{Q}_i(s, q) \le \widetilde{O}\left(s^{\alpha} + \sqrt{s} + \sqrt{\frac{s}{h}}\, \mathrm{Q}_{i-1}(L, q) \right). \tag{30}$$
The first two terms include sampling, fixed-point searches, the sample-incident search, the zero test, and approximate counting. Let $T_i(\kappa)$ satisfy
$$\mathrm{Q}_i(s, q) = \widetilde{O}(s^{T_i(\kappa)})$$

when $q = s^\kappa$. At depth zero, the Johnson-graph algorithm gives $T_0(\kappa) = 2/3$. Suppose more precisely that

$$\mathrm{Q}_{i-1}(u, q) = \widetilde{O}(u^{a_{i-1}} q^{c_{i-1}}).$$

When $q \le L^{2/3}$, substitute $L = \widetilde{O}(h)$ and $h \le 4U = O(qs^{1-\alpha})$ into Equation (30). Since $a_{i-1} \ge 1/2$, the recursive contribution is nondecreasing in h and has s-exponent at most

$$\frac{1}{2} + (1-\alpha)(a_{i-1} - \tfrac{1}{2}) + \kappa(a_{i-1} - \tfrac{1}{2} + c_{i-1}). \tag{31}$$

Equation (30) therefore gives

$$T_i(\kappa) = \min_{\max\{1/2,\kappa\} \le \alpha \le 1} \max\left\{ \frac{1}{2},\, \alpha,\, \frac{1}{2} + (1-\alpha)(a_{i-1} - \tfrac{1}{2}) + \kappa(a_{i-1} - \tfrac{1}{2} + c_{i-1}) \right\}. \tag{32}$$

The last expression decreases with α, whereas the sampling exponent increases. Their intersection is

$$\alpha_i(\kappa) = \frac{a_{i-1} + \kappa(a_{i-1} - \frac{1}{2} + c_{i-1})}{a_{i-1} + \frac{1}{2}}.$$

Starting from $a_0 = 2/3$ and $c_0 = 0$, this choice lies in the permitted interval for $0 \le \kappa \le 2/3$. Indeed, $1/2 \le \alpha_i(\kappa) \le 2/3$ and $\kappa \le \alpha_i(\kappa)$. Balancing the two terms yields

$$a_i = \frac{a_{i-1}}{a_{i-1} + 1/2}, \qquad c_i = \frac{a_{i-1} - 1/2 + c_{i-1}}{a_{i-1} + 1/2}.$$

Solving these recurrences gives

$$T_i(\kappa) = a_i + c_i\kappa, \qquad a_i = \frac{2^{i+1}}{2^{i+2} - 1}, \qquad c_i = \frac{2^i - 1}{2^{i+2} - 1}. \tag{33}$$

Thus $a_i \to 1/2$ and $c_i \to 1/4$ as the fixed depth grows.

One boundary case is essential. The recursive formula applies only when the virtual list remains in the small-range regime, $q \le L^{2/3}$. If instead $q > L^{2/3}$, then $b > 0$, $h \ge 1$, and $h \le L < q^{3/2}$. The algorithm uses the Johnson-graph method on the virtual list. Measured in original queries, this costs

$$\widetilde{O}(\sqrt{s/h}\, L^{2/3}) = \widetilde{O}(s^{1/2} h^{1/6}) \le \widetilde{O}(s^{1/2} q^{1/4}), \tag{34}$$

where $L = \widetilde{O}(h)$. Moreover,

$$T_i(\kappa) - \left(\frac{1}{2} + \frac{\kappa}{4}\right) = \frac{2 - 3\kappa}{4(2^{i+2} - 1)} \ge 0.$$

Thus the short-list fallback also satisfies the claimed depth-i bound.

Every search, count, and virtual evaluation is truncated at its stated query cap. Let M_i be a deterministic upper bound on the number of virtual-oracle calls made by the selected child at level i. For a prescribed level error δ, coherently amplify each virtual evaluator to failure probability $O((\delta/M_i)^2)$. The resulting compute–copy–uncompute operation differs from an ideal virtual query by $O(\delta/M_i)$ in operator norm. A hybrid argument over at most M_i calls contributes $O(\delta)$ error at that level. Allocating a sufficiently small δ to each level keeps the total error below $1/3$ because the recursion depth is fixed. The required logarithmic amplification factors are absorbed by $\widetilde{O}(\cdot)$. Choosing a constant depth with $a_i \le 1/2 + \varepsilon/2$ and absorbing the fixed-depth logarithmic factors proves

$$\mathrm{Q}(\text{2-Sum}_{\mathbb{Z}_q,t,N}) = O_\varepsilon(N^{1/2+\varepsilon} q^{1/4}).$$

The same Claw connection therefore governs both sides of the remaining window: $q^{1/6}$ in the lower bound and $q^{1/4}$ in the upper bound.