TCS-BENCH: BENCHMARKING STATE-OF-THE-ART GENERATIVE AI THEORETICAL COMPUTER SCIENCE RESEARCH ABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce TCS-BENCH, a benchmark for evaluating Large Language Models (LLMs) on research-level Theoretical Computer Science (TCS) proof generation. TCS-BENCH consists of theorem-proving tasks from papers published at top theoretical computer science venues (STOC, FOCS, and SODA). Each task provides the necessary context to derive a self-contained proof for a target result. We evaluate state-of-the-art models on this benchmark. We verify the correctness of generated proofs via a verification agent, and further benchmark the verifier against human-expert proof judgements on a set of target statements and generated proofs pairs. Our reference verifier achieves over 90% accuracy on the expert labeled set. The tasks are publicly available at `https://github.com/TCSBench/TCSBench`.

1 INTRODUCTION

Over the last two years, Large Language Models (LLMs) have achieved superhuman performance on a variety of standardized benchmarks, from professional exams to programming competitions (2; 4; 10; 17; 29; 30; 36). These successes have extended into the formal domain of mathematics, where specially primed models have demonstrated the ability to solve problems at the level of an International Mathematical Olympiad (IMO) gold medalist (8), suggesting that LLMs can emulate rigorous mathematical logic.

Recently, these capabilities have crossed the threshold into open-ended scientific discovery with models actively contributing to expert-level mathematical and scientific breakthroughs. For example, Google's Gemini Deep Think and its advanced variants have successfully collaborated with human researchers to solve open problems, refute conjectures, and generate novel proofs across theoretical computer science, optimization, and physics (34),(11). An internal version of OpenAI's ChatGPT disproved the unit-distance conjecture (22) and other long-standing mathematical and theoretical computer science problems (23). Anthropic's models have similarly been utilized to independently discover counterexamples to long-standing conjectures (1).

However, these frontier-level successes highlight a critical limitation in how the AI community evaluates language models: a significant gap has emerged between what state-of-the-art agents can achieve in active research and what our standardized benchmarks can actually measure. We find the research field to be a rich, sustainable, and challenging testbed for evaluating the next generation of language models. Existing benchmarks, while valuable, fail to capture the core difficulties of real-world mathematical research for several key reasons. First, competition problems like those found in the IMO are typically self-contained, whereas research theorems are deeply embedded in the context of bespoke definitions, notations, and previously established lemmas. Second, research results frequently involve constructing a scaffold of interconnected results, a task that goes far beyond finding a single, clever insight. To truly measure progress, we need challenging benchmarks that more accurately reflect the work of human researchers.

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.

1.1 Our Contributions

To this end, we introduce TCS-BENCH, a new benchmark for evaluating an LLM's ability to prove theorems from cutting-edge Theoretical Computer Science (TCS) research papers. The task is grounded in scientific practice: a model is presented with a target statement extracted from a paper published at a top-tier conference, and is tasked with generating a proof. The challenging part is to provide all required "basic" mathematical context required to prove the target statement. Our key contribution is to provide a self-contained task that can be solved without access to the internet, and that evaluates the ability of the models to come up with a proof from first principles and the intermediate, state-of-the-art lemmas provided in the context. This is obtained by processing the paper the task is extracted from and the target proof.

Furthermore, a central aspect in our task generation process is the ability to generate harder and harder tasks by having a tight control over the generated context. Indeed, by masking intermediate Lemmas used in the proof of the target statement, the task becomes harder. One can thus increase the difficulty of the tasks by simply masking more and more intermediate results, all the way to define tasks that consist in proving the main result of the paper.

Then successfully resolving a task requires a model to comprehend the surrounding context, understand the intricate connections between lemmas and theorems, and generate a logically sound proof that would provide a similar amount of details as the peer-reviewed proof extracted from the paper. This format challenges models to perform the context-dependent reasoning that is a hallmark of scientific discovery. Our contributions are as follows:

- TCS-BENCH, a benchmark composed of 300 theorem-proving tasks from papers published at the top theoretical computer science venues, FOCS, STOC, and SODA, between 2020 and 2026. We make the tasks publicly available at `https://github.com/TCSBench/TCSBench`.
- We develop and validate an automated proof verification system, achieving more than 90% accuracy against human expert judgments on a held-out set of 100 human-labeled proofs.

Task Construction Overview. The foundation of our benchmark is a curriculum of statement-proving tasks from publicly available conference proceedings and that can and should be solved without requiring access to the internet. At a high level, we scrape and process the LaTeX source of a paper, build the dependency graph of the statements (theorems, lemmas, claims, etc), and for each of the statements we assemble multiple versions of the context needed to attempt the proof, hiding varying subsets of dependencies. Each task is then packaged as a self-contained JSON entry comprising three fields: a context, a target statement and a ground-truth proof. Full technical details of the benchmark construction are given in Section 3.

Benchmark Overview. Tasks are selected to span a broad range of TCS subfields and, critically, a broad range of difficulty levels. This stratification ensures that TCS-BENCH is informative across the full capability spectrum of current and future models, rather than saturating at either extreme. A model is evaluated on TCS-BENCH by generating a proof for each task in the benchmark and submitting those proofs to an automated (and validated) verifier. We specify a reference verifier and report all our baseline results using this to ensure reproducibility.

2 Related Work

Our work is situated at the intersection of benchmarks for mathematical reasoning, formal proof generation, and the broader evaluation of AI in scientific research workflows.

Competition Mathematics Benchmarks. A significant body of work has focused on evaluating LLMs on competition-style mathematics problems. Benchmarks in this area often draw from sources like the International Mathematical Olympiad (IMO), testing a model's ability to find clever insights for well-defined, self-contained problems (5; 10; 20; 21; 29). A notable success in this domain is AlphaGeometry2, which demonstrated performance equivalent to an IMO gold medalist (14). While these benchmarks are invaluable for measuring discrete problem-solving abilities, their self-contained nature does not reflect the challenges of genuine research (27). In contrast, TCS-BENCH sources its

problems directly from published literature. This requires the model not just to solve a problem, but to reason within the rich, interdependent theoretical context established by the paper's definitions, notations and prior lemmas.

Research-Level Mathematical Benchmarks. Recent work has begun to address the gap between competition mathematics and genuine research. HorizonMath (32) presents over 100 predominantly unsolved problems from computational and applied mathematics, leveraging a generator-verifier gap where solutions are hard to produce but efficient to verify through numerical comparison or deterministic constraint checking. LemmaBench (26) takes a complementary approach by automatically extracting and contextualizing lemmas from recent arXiv preprints, creating an updatable benchmark immune to contamination through continuous refreshment from new publications. BrokenMath (25) evaluates a different failure mode—sycophancy in theorem proving—by perturbing valid mathematical statements into plausible but false versions, revealing that even frontier models attempt to prove false statements 29-70% of the time. TCS-BENCH shares with these works the focus on research-level mathematics beyond competition problems, but differs in targeting proof completion within the rich contextual dependencies of published theoretical computer science papers, where success requires not just solving isolated problems but reasoning within an interconnected scaffold of definitions and prior results.

Formal Theorem Proving. Another major direction of research focuses on benchmarking LLMs for proof generation in formal languages using interactive theorem provers like Lean or Coq (4; 6; 10; 18; 19). This work evaluates a model's ability to generate sequences of tactics to construct a machine-verifiable proof, often within established formal mathematics libraries. Similarly, benchmarks like MathConstruct test a model's ability to generate a specific mathematical object whose properties can be formally verified by an automated function (9). This line of work is critical for ensuring logical rigor. TCS-BENCH complements these efforts by focusing on a different but equally important task: generating complete, human-readable proofs in natural language. The evaluation challenge in our work shifts from formal machine-verifiability to assessing logical soundness within the implicit argumentative structure of a research paper—a task that is more aligned with the daily workflow of human mathematicians.

AI for Scientific Discovery. As LLMs advance, the focus of evaluation is shifting from solving established problems to assessing their potential to accelerate scientific discovery (11; 12; 16; 33; 34; 35). Recent work has demonstrated their utility across numerous domains, from chemistry and biology to astrophysics (7; 13; 15; 24; 31). Our work aligns with a new class of benchmarks designed to evaluate AI agents on complex, long-horizon scientific tasks. For example, PaperBench evaluates an agent's ability to replicate an entire AI research paper, a process that includes understanding the paper, developing a codebase, and executing experiments to reproduce its empirical results (28). This paradigm assesses a model's practical utility in a realistic research workflow. TCS-BENCH adapts this "research-as-benchmark" paradigm to the domain of theoretical computer science. While PaperBench focuses on replicating empirical results, TCS-BENCH is the first to focus on replicating theoretical contributions. By isolating the task to proof synthesis within the context of a research paper, our benchmark provides a targeted measure of the sophisticated, context-dependent reasoning required to contribute to the frontiers of mathematical knowledge.

3 THE TCS BENCHMARK

TCS-BENCH evaluates a model's ability to prove theorems drawn from cutting-edge theoretical computer science research. Each task in the benchmark is a self-contained proof-completion problem wherein the model receives a curated context comprised of definitions, prior lemmas and condensed external references. Given this primer knowledge together with a target statement, the model is tasked with producing a complete proof. Crucially, all tasks are derived from published papers whose proofs have been systematically removed, so that success requires chains of complex mathematical reasoning rather than simple memorization and recall.

3.1 Data Acquisition and Preprocessing

We construct the benchmark from papers published at FOCS, STOC, and SODA, between 2020 and 2026. For each paper, we obtain the LaTeX source from arXiv, limiting our dataset to papers released under permissive licenses (CC-0 or CC-BY-4.0) to ensure all benchmark content is legally distributable. We list all papers we used in appendix B.

Research proofs routinely invoke results from prior literature. To ensure tasks are self-contained, we skip target statements whose proofs invoke external results without restating them.

3.2 Dependency Structure Construction

The core of our benchmark construction is a structural analysis that extracts the logical dependency structure of each paper. This process produces a directed acyclic graph (DAG) over all formal statements, where an edge from statement A to statement B indicates that the proof of B depends on A.

To extract this graph, we first use a deterministic LaTeX parser to identify all theorem-like environments (theorem, lemma, definition, corollary, etc.) and assign each a unique identifier. We then employ an LLM-based analysis pass to:

- Map each proof environment to the statement it proves (non-trivial since proofs may appear out of order or span multiple environments)
- Analyze the full paper to construct dependency edges—identifying when the proof of statement B invokes statement A

We programmatically verify acyclicity and reject any paper for which the extracted graph contains a cycle, as this indicates an extraction error.

From the DAG, we compute the **rank** of each statement as the length of the longest directed path terminating at that node. Definitions and axioms have rank 0, and each subsequent layer of derived results increments the rank. This ranking serves two purposes: it stratifies tasks by difficulty (higher-rank proofs require reasoning about longer chains of dependencies), and it prevents information leakage during context assembly (we can systematically hide all results of rank $\geq r$ when constructing a task for a rank-r statement).

Figure 1 illustrates an example dependency DAG extracted from a paper. The graph shows how a main theorem (rank 3) depends on intermediate lemmas (ranks 1-2), which in turn depend on foundational definitions (rank 0). Each blue node represents a proof-completion task in our benchmark. Note that only statements that have corresponding `\begin{proof}\end{proof}` tags in the paper are turned into tasks for our benchmark.

3.3 Task Construction

Each proof-completion task consists of three components: a *context*, a *target statement*, and a *ground-truth proof* (withheld during evaluation). The key challenge in constructing high-quality tasks is producing a context that is both self-contained (containing all information logically necessary to derive the proof) and concise enough to fit within standard context windows. We achieve this through the following procedure.

Initial Context Assembly. For a target statement s of rank r, we initialize the context by concatenating (i) all resolved external reference digests, (ii) the paper's text truncated at the start of s's proof, with two categories of redaction applied: all statements of rank $\geq r$ are hidden to prevent information leakage from later results, and all proof environments are removed. If any dependency of s falls outside the truncated portion of the paper, we re-insert its statement into the context.

Scalable Difficulty Tasks. To create tasks spanning a range of difficulties, we exploit the dependency structure to generate multiple variants of each proof task. Starting from a base task where all dependencies are provided in the context, we systematically withhold intermediate results, requiring the model to discover and prove them on the way to the main target.

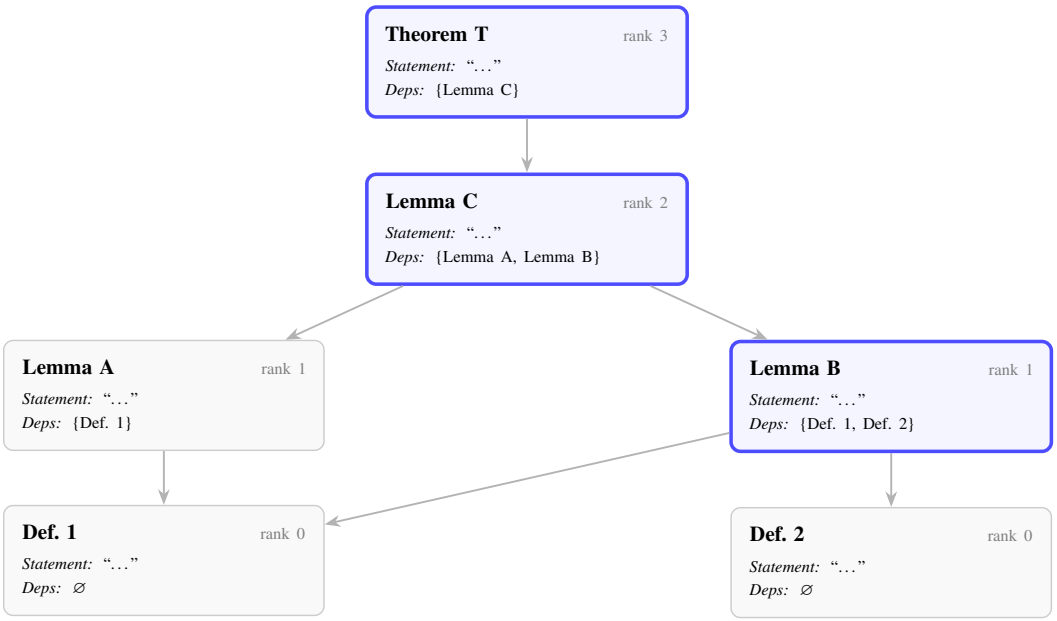


Figure 1: DAG example

[CONTEXT]
...
Lemma A....
Lemma C. *The sequence* $(u_n)_{n\in\mathbb{N}}$ *is upper-bounded.*
In the following, consider $\epsilon > 0$... ...

[TARGET STATEMENT]
Theorem T. *Algorithm 2 terminates in polynomial time.*

[GROUND-TRUTH PROOF]
Proof. We start by proving by induction that at the i-th iteration of the while-loop, at most u_i recursive calls are made.
...
Hence, by Lemma C, Algorithm 2 terminates in polynomial time. □

Figure 2: Base difficulty task (all dependencies provided)

Example. Consider proving Theorem T from Figure 1, which depends on Lemma C, which in turn depends on Lemmas A and B. The base difficulty for prompting a model to construct a proof would be to supply the model with all dependent results within the context (as depicted in Figure 2). Furthermore, we can easily increase the task complexity by omitting a subset of the dependencies (example prompting in Figure 3). Thus, forcing the model to prove such intermediary results along the way to the final claim. This procedure generates a spectrum of difficulties: at one extreme, all dependencies are provided and the model need only combine them; at the other, the model must reconstruct substantial portions of the paper's proof architecture.

Context Compression. To make the tasks short enough to fit the standard context window of $10,000$ tokens, we apply the following procedure.

First, we conduct *iterative section pruning*. Since many papers contains sections (e.g. related work, motivating context) that are irrelevant to a given target statement, we iteratively parse the section hierarchy of the assembled context and prompt an LLM to identify sections/subsections/subsubsections

[CONTEXT]
...
Lemma A....
In the following, consider $\epsilon > 0$... ...

[TARGET STATEMENT]
Theorem T. *Algorithm 2 terminates in polynomial time.*

[GROUND-TRUTH PROOF]
We start by proving:
Lemma C. *The sequence $(u_n)_{n\in\mathbb{N}}$ is upper-bounded.*
Proof of Lemma C. ...
Proof of Theorem T. We start by proving by induction that at the i-th iteration of the while-loop, at most u_i recursive calls are made.
...
Hence, by Lemma C, Algorithm 2 terminates in polynomial time. □

Figure 3: Task with omitted intermediary results to increase complexity.

that are entirely irrelevant to the target problem and its proof. Identified sections are removed, and the process repeats until no further pruning is possible. Second, we apply an *LLM shortner* which, following section pruning condenses the remaining context (ie. shortening verbose passages or tightening exposition) while preserving all mathematically essential content.

3.4 Quality Filtering

To address potential information loss or artifacts introduced during context assembly, we implement a final quality filtering stage to ensure all generated tasks are self-contained, well-posed, and free of information leakage. This filtering applies two categories of automated criteria. First, structural checks exclude tasks that contain text-incompatible elements (such as figure references), unresolvable dependencies, leaked proof metadata, or excessive token lengths. Second, semantic checks utilize large language models to verify that all mathematical objects are properly defined, the target statement is unambiguous and correctly proven by the ground truth, and the assembled context remains logically and mathematically coherent. Any task failing these structural or semantic checks is excluded from the final benchmark.

3.5 Task Formulation Summary

Each released task provides the solver with:

1. A self contained LaTeX context ($\leq$ 10,000 tokens) comprising definitions, prior results (with proofs omitted), and condensed external reference digests.
2. A target statement to be proved.

The solver must produce a single, complete proof of the target statement, using LaTeX for mathematical notations. All justifications must follow from results present in the provided context and the solver is prohibited from accessing the original paper or external sources. The ground-truth proof is withheld and used exclusively for the evaluation.

4 Automated Proof Verification

A primary challenge in a benchmark like TCS-Bench is the need for a scalable and reliable method to evaluate the correctness of generated mathematical proofs. Manual verification by human experts is prohibitively slow and expensive. To address this, we developed a specialized verifier agent, tasked with deciding whether a candidate solutions constitutes a valid proof.

Table 1: Model performance on TCS-BENCH measured by accuracy.

Model	**Accuracy** (↑)
Opus 5	32.77
Gemini 3.1 Pro	30.3
Gemini 3.1 DeepThink	52
GPT 5.6 Pro (max)	68
Colosseum Cross Model	71

The verifier's goal is to make a binary decision (correct or incorrect) on a candidate proof for a specific statement within a given context. It is aligned to tolerate trivial omissions common in academic literature (e.g., "the rest follows by simple algebra") but reject proofs with critical logical gaps or errors.

The verifier is provided with three key inputs: the task's context and target statement that were passed to the solver, and additionally the ground-truth proof. Then, four calls are made to Gemini 3.1 Flash, a cheap model, and a candidate proof is deemed correct if and only if at least three of the four verdicts mark it as correct. The prompt instructs the model to rigorously assess the submission across four dimensions: logical rigor (ensuring strict adherence to provided lemmas without introducing unauthorized assumptions), precise variable tracking and scoping, semantic integrity (penalizing hand-waving or jargon-heavy hallucinations), and exact quantitative accuracy in derivations and boundary conditions.

To design the verifier prompt, we generated proofs by running the solver on a set of tasks, disjoint from the benchmark. Then, human experts reviewed them, and produced a set of 50 correct proofs and 50 incorrect proofs. Finally, we ran the GEPA (3) improvement pipeline on this alignment task to produce a prompt that achieves an accuracy of more than 90%.

5 Experimental Setup

We evaluate frontier language models on TCS-BENCH to establish baseline performance on research-level mathematical proof generation. Our evaluation focuses on measuring the current capabilities of state-of-the-art models when presented with proof-completion tasks drawn from cutting-edge theoretical computer science research.

5.1 Experimental Setup

We assess the following frontier models: Gemini 3.1 Pro and Gemini 3.1 DeepThink, Opus 5, and GPT 5.6 Pro. We also present COLOSSEUM, an internal harness that we evaluate using Gemini 3.1 Pro, Gemini 3.7 Flash, and a combination of the two. All models were accessed via their respective API endpoints or web interfaces using the most recent versions available at the time of evaluation. For each task, we query the model with the context and target statement as described in Section 3. We provide the exact prompt formatting in appendix A.1.

For models offering extended reasoning capabilities, we use the maximum publicly available thinking budget to allow models to fully explore the problem space. Generated proofs are evaluated using our automated verifier (Section 4). Each model attempts all 300 tasks in TCS-BENCH.

Results & Analysis. Table 1 presents the overall performance of each model on TCS-BENCH. The strongest performing model, GPT 5.6 Pro, achieves an accuracy score of 68%, successfully proving 204 of 300 tasks. Notably, Opus 5, exhausts its token budget of 128K tokens before obtaining an answer for 162 out of the 300 tasks, which negatively affects its performance.

Results & Analysis with COLOSSEUM The results above evaluate base models directly. We also report results for COLOSSEUM, an agentic proof-search harness we run on top of a base model: for each problem it explores several candidate proof strategies, decomposes the target statement into subproblems, solves them, and assembles and revises a final proof. We do not describe COLOSSEUM

	Solved / 300	Accuracy
COLOSSEUM, Gemini 3.1 Pro	162	54.0%
COLOSSEUM, Gemini 3.7 Flash	165	55.0%
COLOSSEUM, cross-model selection between the two	213	71.0%
Oracle best-of-two (upper bound, not achievable)	217	72.3%

Table 2: COLOSSEUM run on two different base models, with automated cross-model selection between the resulting proofs. All rows are graded by the automated grader of Section 4. The oracle row reports the fraction of problems solved by at least one of the two runs; it requires knowing the answer and upper-bounds any selection rule.

in detail here; the only property that matters below is that it is parameterised by its base model, so the same pipeline can be run on different models to obtain independent proofs of the same problem.

Selecting between two runs. Running COLOSSEUM on two different base models yields two independent proofs of every problem, and we select one of them automatically. The selection asks the second base model to *critique* the first model's proof: we draw 8 independent critiques and submit the second model's proof if at most half of them judge the first model's proof to be correct, and the first model's proof otherwise.

This critic is a component of COLOSSEUM and is distinct from the automated grader of Section 4. The grader is used only to score final submissions; the selection has no access to it, to ground truth, or to any human judgement.

Why cross-model. The rule relies on one observation: a model is a poor judge of its own proof, but a competent judge of another model's. On TCS-BENCH, 93.4% of the cases in which Gemini 3.1 Pro accepts its own proof are unanimous, whereas cross-model critiques separate correct from incorrect proofs well (AUC 0.854–0.896). The result is not sensitive to the precise critique cutoff. Across cutoffs from $0/8$ through $7/8$, accuracy remains between 68.3% and 71.0%.

While we lack comprehensive human baselines, we note that all tasks in TCS-BENCH have ground-truth proofs published by domain experts. The gap between the strongest model 71% and perfect performance (100%) represents the current frontier of automated mathematical reasoning.

6 DISCUSSION

We have introduced TCS-BENCH, the first benchmark evaluating LLMs on proof generation from cutting-edge theoretical computer science research. Unlike competition mathematics benchmarks testing isolated problems, TCS-BENCH requires reasoning within the rich contextual scaffolding of research papers—navigating bespoke definitions, dependency structures, and chains of intermediate results across 300 tasks spanning multiple difficulty levels.

A key contribution is our automated proof verification system achieving over 90% accuracy against human expert judgments, enabling scalable evaluation without prohibitive manual verification costs. The verifier tolerates stylistic variations common in mathematical writing while rigorously detecting logical gaps—a balance essential for meaningful evaluation.

Our evaluation of five frontier models shows the strongest systems: Gemini 3.1 DeepThink solves 52%, GPT-5.6-Pro solves 68% and Colosseum with a cross-model approach solves 71.0%. While this demonstrates meaningful progress in automated mathematical reasoning, the gap to perfect performance reveals substantial room for advancement. The stable performance ceiling across all models suggests current architectures face fundamental limitations in constructing multi-step mathematical arguments within complex dependencies.

We lastly note that TCS-BENCH is designed for longevity through continual addition of new papers that postdate model training cutoffs, scalable difficulty via dependency-hiding that spans easy to extremely challenging variants, and rank-based stratification enabling fine-grained progress tracking.

7 AI USE DISCLOSURE

Gemini 3.1 Flash and Pro were used as coding assistants.

REFERENCES

[1] Claude Fable 5. The jacobian conjecture is false. https://x.com/__alpoge__/status/2079028340955197566, 2026. Accessed: 2026-08-07.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. Gepa: Reflective prompt evolution can outperform reinforcement learning. *arXiv preprint arXiv:2507.19457*, 2025.

[4] Team AlphaProof and Team AlphaGeometry. AI achieves silver-medal standard solving international 178 mathematical olympiad problems. *DeepMind blog*, 179:45, 2024.

[5] Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. MathArena: Evaluating LLMs on Uncontaminated Math Competitions. *arXiv preprint arXiv:2505.23281*, 2025.

[6] Barış Bayazıt, Yao Li, and Xujie Si. A Case Study on the Effectiveness of LLMs in Verification with Proof Assistants. *arXiv preprint arXiv:2508.18587*, 2025.

[7] Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023.

[8] Yuri Chervonyi, Trieu H Trinh, Miroslav Olšák, Xiaomeng Yang, Hoang Nguyen, Marcelo Menegali, Junehyuk Jung, Vikas Verma, Quoc V Le, and Thang Luong. Gold-medalist performance in solving olympiad geometry with alphageometry2. *arXiv preprint arXiv:2502.03544*, 2025.

[9] Jasper Dekoninck, Mislav Balunovic, Nikola Jovanović, Ivo Petrov, and Martin Vechev. MathConstruct: Challenging LLM reasoning with constructive proofs. In *ICLR 2025 Workshop: VerifAI: AI Verification in the Wild*, 2025.

[10] Jasper Dekoninck, Ivo Petrov, Kristian Minchev, Mislav Balunovic, Martin Vechev, Miroslav Marinov, Maria Drencheva, Lyuba Konova, Milen Shumanov, Kaloyan Tsvetkov, et al. The Open Proof Corpus: A Large-Scale Study of LLM-Generated Mathematical Proofs. *arXiv preprint arXiv:2506.21621*, 2025.

[11] Tony Feng, Trieu H. Trinh, Garrett Bingham, Dawsen Hwang, Yuri Chervonyi, Junehyuk Jung, Joonkyung Lee, Carlo Pagano, Sang hyun Kim, Federico Pasqualotto, Sergei Gukov, Jonathan N. Lee, Junsu Kim, Kaiying Hou, Golnaz Ghiasi, Yi Tay, YaGuang Li, Chenkai Kuang, Yuan Liu, Hanzhao Lin, Evan Zheran Liu, Nigamaa Nayakanti, Xiaomeng Yang, Heng-Tze Cheng, Demis Hassabis, Koray Kavukcuoglu, Quoc V. Le, and Thang Luong. Towards autonomous mathematics research, 2026.

[12] Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Anil Palepu, Petar Sirkovic, Artiom Myaskovsky, Felix Weissenberger, Keran Rong, Ryutaro Tanno, et al. Towards an ai co-scientist. *arXiv preprint arXiv:2502.18864*, 2025.

[13] Kaixuan Huang, Yuanhao Qu, Henry Cousins, William A Johnson, Di Yin, Mihir Shah, Denny Zhou, Russ Altman, Mengdi Wang, and Le Cong. CRISPR-GPT: An LLM agent for automated design of gene-editing experiments. *arXiv preprint arXiv:2404.18021*, 2024.

[14] Yichen Huang and Lin F Yang. Gemini 2.5 pro capable of winning gold at imo 2025. *arXiv preprint arXiv:2507.15855*, 2025.

[15] Ross Irwin, Spyridon Dimitriadis, Jiazhen He, and Esben Jannik Bjerrum. Chemformer: a pre-trained transformer for computational chemistry. *Machine Learning: Science and Technology*, 3(1):015022, 2022.

[16] Moksh Jain, Tristan Deleu, Jason Hartford, Cheng-Hao Liu, Alex Hernandez-Garcia, and Yoshua Bengio. Gflownets for ai-driven scientific discovery. *Digital Discovery*, 2(3):557–577, 2023.

[17] Tiffany H Kung, Morgan Cheatham, Arielle Medenilla, Czarina Sillos, Lorie De Leon, Camille Elepaño, Maria Madriaga, Rimel Aggabao, Giezel Diaz-Candido, James Maningo, et al. Performance of chatgpt on usmle: potential for ai-assisted medical education using large language models. *PLoS digital health*, 2(2):e0000198, 2023.

[18] Vanessa Lama, Catherine Ma, and Tirthankar Ghosal. Benchmarking automated theorem proving with large language models. In *Proceedings of the 1st Workshop on NLP for Science (NLP4Science)*, pages 208–218, 2024.

[19] Zenan Li, Zhaoyu Li, Wen Tang, Xian Zhang, Yuan Yao, Xujie Si, Fan Yang, Kaiyu Yang, and Xiaoxing Ma. Proving Olympiad Inequalities by Synergizing LLMs and Symbolic Reasoning. *arXiv preprint arXiv:2502.13834*, 2025.

[20] Junqi Liu, Xiaohan Lin, Jonas Bayer, Yael Dillies, Weijie Jiang, Xiaodan Liang, Roman Soletskyi, Haiming Wang, Yunzhou Xie, Beibei Xiong, et al. CombiBench: Benchmarking LLM capability for combinatorial mathematics. *arXiv preprint arXiv:2505.03171*, 2025.

[21] Yujun Mao, Yoon Kim, and Yilun Zhou. CHAMP: A Competition-level Dataset for Fine-Grained Analyses of LLMs' Mathematical Reasoning Capabilities. *arXiv preprint arXiv:2401.06961*, 2024.

[22] OpenAI. An openai model has disproved a central conjecture in discrete geometry. `https://openai.com/index/model-disproves-discrete-geometry-conjecture/`, 2026. Accessed: 2026-08-07.

[23] OpenAI. Ten advances in mathematics and theoretical computer science. `https://openai.com/index/ten-advances-in-mathematics/`, 2026. Accessed: 2026-08-07.

[24] Liam Parker, Francois Lanusse, Siavash Golkar, Leopoldo Sarra, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Geraud Krawezik, Michael McCabe, Rudy Morel, et al. Astroclip: a cross-modal foundation model for galaxies. *Monthly Notices of the Royal Astronomical Society*, 531(4):4990–5011, 2024.

[25] Ivo Petrov, Jasper Dekoninck, and Martin Vechev. BrokenMath: A Benchmark for Sycophancy in Theorem Proving with LLMs. *arXiv preprint arXiv:2510.04721*, 2025.

[26] Antoine Peyronnet, Fabian Gloeckle, and Amaury Hayat. LemmaBench: A Live, Research-Level Benchmark to Evaluate LLM Capabilities in Mathematics. *arXiv preprint arXiv:2602.24173*, 2026.

[27] Inioluwa Deborah Raji, Emily M Bender, Amandalynne Paullada, Emily Denton, and Alex Hanna. Ai and the everything in the whole wide world benchmark. *arXiv preprint arXiv:2111.15366*, 2021.

[28] Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, et al. Paperbench: Evaluating ai's ability to replicate ai research. In *Forty-second International Conference on Machine Learning*.

[29] Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

[30] Lakshmi Varanasi. Gpt-4 can ace the bar, but it only has a decent chance of passing the cfa exams. here'sa list of difficult exams the chatgpt and gpt-4 have passed. *Business Insider*, 5, 2023.

[31] Ricardo Vinuesa, Steven L Brunton, and Beverley J McKeon. The transformative potential of machine learning for experiments in fluid mechanics. *Nature Reviews Physics*, 5(9):536–545, 2023.

[32] Erik Y Wang, Sumeet Motwani, James V Roggeveen, Eliot Hodges, Dulhan Jayalath, Charles London, Kalyan Ramakrishnan, Flaviu Cipcigan, Philip Torr, and Alessandro Abate. Horizon-math: Measuring ai progress toward mathematical discovery with automatic verification. *arXiv preprint arXiv:2603.15617*, 2026.

[33] Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.

[34] David P. Woodruff, Vincent Cohen-Addad, Lalit Jain, Jieming Mao, Song Zuo, Mohammad-Hossein Bateni, Simina Branzei, Michael P. Brenner, Lin Chen, Ying Feng, Lance Fortnow, Gang Fu, Ziyi Guan, Zahra Hadizadeh, Mohammad T. Hajiaghayi, Mahdi JafariRaviz, Adel Javanmard, Karthik C. S., Ken ichi Kawarabayashi, Ravi Kumar, Silvio Lattanzi, Euiwoong Lee, Yi Li, Ioannis Panageas, Dimitris Paparas, Benjamin Przybocki, Bernardo Subercaseaux, Ola Svensson, Shayan Taherijam, Xuan Wu, Eylon Yogev, Morteza Zadimoghaddam, Samson Zhou, Yossi Matias, James Manyika, and Vahab Mirrokni. Accelerating scientific research with gemini: Case studies and common techniques, 2026.

[35] Yanbo Zhang, Sumeer A Khan, Adnan Mahmud, Huck Yang, Alexander Lavin, Michael Levin, Jeremy Frey, Jared Dunnmon, James Evans, Alan Bundy, et al. Exploring the role of large language models in the scientific method: from hypothesis to discovery. *npj Artificial Intelligence*, 1(1):14, 2025.

[36] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*, 2023.

A Omitted Details

A.1 Full Prompt Details

We present here the prompt formatting for requesting the evaluated model to prove a target statement:

```
You are an expert mathematician.  Your task is to provide a
thorough and correct proof for a mathematical problem (your Target
Problem).  This proof will be used to benchmark your abilities as
a mathematician.

To obtain your proof, you can assume and use any reference (lemma,
theorem, definition, etc) from the context below:

********** BEGIN CONTEXT **********

{CONTEXT}

********** END CONTEXT **********

********** BEGIN EVALUATION CRITERIA **********

Recall that you can assume and use any of the mathematical
statements provided in the context above.

Your proof must be complete and valid.  As a guideline, here is
a non-exhaustive list of criteria to evaluate your proof.  You
should use it, together with any additional criteria you can think
of, to evaluate any candidate proof before your final response.

Evaluation Criteria:

1.  **Logical Rigor and External Assumption Check**:

    - **No Unauthorized Constraints**:  The proof must not
      introduce arbitrary numerical constraints or "safety
      margins" to simplify the proof (e.g., assuming k ≥ 2,
      assuming ϵ is sufficiently small, or assuming sets are
      non-empty) when such constraints are not explicitly stated
      in the target statement or the context.
    - **Strict Lemma Adherence**:  Every step must follow
      strictly from the provided context or standard mathematical
      knowledge.  Applying a theorem without explicitly verifying
      all its preconditions (as defined in the context) is a
      failure.
    - **Case Integrity**:  If the proof involves case-splitting
      (e.g., i* ≤ T vs i* = T + 1), the proof must address these
      specific boundaries.  Skipping a boundary case or "merging"
      distinct logical paths via hand-waving results in a failure.

2.  **Variable Alignment and Property Scope**:

    - **Property Scope Integrity**:  Properties must only be
      applied to the specific variables for which they are
      defined.  If the proof generalizes a property of a subset to
      a larger set then this constitutes a failure (e.g., applying
      a u-uniformity property defined for "outer queries" T_2 to
```

the "total queries" T, or applying a property of a specific index j to all indices i without proof).

- **Index and Set Precision**: The proof must maintain the integrity of sets (e.g., B vs B^*, S_i vs S_{i-1}) and indices. Misidentifying a variable or applying a property to the wrong time step or set is a fatal error.

3. **Semantic Integrity and Jargon Detection**:

- **No Hallucinated Logic/Word Salad**: If the proof uses repetitive, nonsensical, or overly dense jargon to mask a lack of logical depth, then this is a failure. The proof must be linguistically coherent. If a paragraph consists of technical terms strung together without clear propositional logic (e.g., "mapping limits matching identical parameters constraints mapping"), it is a failure.
- **No "Standard" Hand-waving**: The proof cannot skip non-trivial derivations by claiming they are "standard", "trivial", etc.

4. **Quantitative and Limit Accuracy**:

- **Derivation Accuracy**: Any error in arithmetic, algebraic manipulation, or inequality direction will automatically invalidate the proof.
- **Boundary and Floor/Ceiling Precision**: Bounds must be exactly supported. For example, if a floor function $\lfloor k^{-\epsilon} \rfloor$ is used, the proof must account for all valid values of k (including $k = 1$) unless the prompt restricts them. Failure to do so, invalidates the proof.

5. **Self-Containment**

- **Citations**: Citing external papers to utilize their lemmas, theorems, or proofs is strictly forbidden to prevent hallucinations; unless both the citation and the referenced lemma, theorem, proof, etc, is explicitly stated in the context.

6. **Additional Criteria**

- **Maximum scrutiny**: You must think of any additional criteria that the proof must meet to be correct and ensure your proof passes them. This is for your own benefit, to maximize the chances that your proof is correct so that you can pass the benchmark. You have no incentive to avoid thinking of additional criteria because then you may miss a bug in the proof and fail the benchmark.

********** END EVALUATION CRITERIA **********

********** BEGIN FINAL RESPONSE FORMAT **********

Once you have completed your reasoning, have obtained a proof that passes all evaluation criteria, and are ready to submit your final answer, your final output should be the proof in latex format.

- DO NOT include any other text, reasoning, or formatting in your final submission turn.

```
    - If you have the ability to store your answer in a file, DO
      NOT use it.  The proof should be in your final response.

********** END FINAL RESPONSE FORMAT **********

With this in mind, you are tasked to prove the following target
statement.

Target Problem (your task):

{TARGET STATEMENT}
```

B LIST OF PAPERS

The following is the list of arxiv preprints from which we extracted the 300 tasks.

1. Constant Approximation of Fréchet Distance in Strongly Subquadratic Time. Siu-Wing Cheng, Haoqiang Huang, Shuo Zhang. https://arxiv.org/abs/2503.12746
2. Hamiltonicity of random subgraphs of the hypercube. Padraig Condon, Alberto Espuny Díaz, António Girão, Daniela Kühn, Deryk Osthus. https://arxiv.org/abs/2007.02891
3. Learning quantum Hamiltonians at any temperature in polynomial time. Ainesh Bakshi, Allen Liu, Ankur Moitra, Ewin Tang. https://arxiv.org/abs/2310.02243
4. Nearly Tight Regret Bounds for Profit Maximization in Bilateral Trade. Simone Di Gregorio, Paul Dütting, Federico Fusco, Chris Schwiegelshohn. https://arxiv.org/abs/2509.22563
5. Vertex Fault-Tolerant Emulators. Greg Bodwin, Michael Dinitz, Yasamin Nazari. https://arxiv.org/abs/2109.08042
6. Locally Sampleable Uniform Symmetric Distributions. Daniel M. Kane, Anthony Ostuni, Kewen Wu. https://arxiv.org/abs/2411.08183
7. A Quantum Speed-Up for Approximating the Top Eigenvectors of a Matrix. Yanlin Chen, András Gilyén, Ronald de Wolf. https://arxiv.org/abs/2405.14765
8. Understanding Memory-Regret Trade-Off for Streaming Stochastic Multi-Armed Bandits. Yuchen He, Zichun Ye, Chihao Zhang. https://arxiv.org/abs/2405.19752
9. On bounded depth proofs for Tseitin formulas on the grid; revisited. Johan Håstad, Kilian Risse. https://arxiv.org/abs/2209.05839
10. Subexponential Parameterized Algorithms for Hitting Subgraphs. Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, Jie Xue, Meirav Zehavi. https://arxiv.org/abs/2409.04786
11. An efficient quantum parallel repetition theorem and applications. John Bostanci, Luowen Qian, Nicholas Spooner, Henry Yuen. https://arxiv.org/abs/2311.10681
12. A Tolerant Independent Set Tester. Cameron Seth. https://arxiv.org/abs/2503.21441
13. On complete classes of valuated matroids. Edin Husić, Georg Loho, Ben Smith, László A. Végh. https://arxiv.org/abs/2107.06961
14. Fully Dynamic Min-Cut of Superconstant Size in Subpolynomial Time. Wenyu Jin, Xiaorui Sun, Mikkel Thorup. https://arxiv.org/abs/2401.09700
15. Stochastic scheduling with Bernoulli-type jobs through policy stratification. Antonios Antoniadis, Ruben Hoeksma, Kevin Schewior, Marc Uetz. https://arxiv.org/abs/2505.03349

16. Deterministic Algorithms for Decremental Approximate Shortest Paths: Faster and Simpler. Maximilian Probst Gutenberg, Christian Wulff-Nilsen. https://arxiv.org/abs/2001.10809
17. Supercritical Tradeoffs for Monotone Circuits. Mika Göös, Gilbert Maystre, Kilian Risse, Dmitry Sokolov. https://arxiv.org/abs/2411.14268
18. Finding Skewed Subcubes Under a Distribution. Parikshit Gopalan, Roie Levin, Udi Wieder. https://arxiv.org/abs/1911.07378
19. An Improved Algorithm for The k-Dyck Edit Distance Problem. Dvir Fried, Shay Golan, Tomasz Kociumaka, Tsvi Kopelowitz, Ely Porat, Tatiana Starikovskaya. https://arxiv.org/abs/2111.02336
20. Maximally Extendable Product Codes are Good Coboundary Expanders. Gleb Kalachev, Pavel Panteleev. https://arxiv.org/abs/2501.01411
21. Stronger 3-SUM Lower Bounds for Approximate Distance Oracles via Additive Combinatorics. Amir Abboud, Karl Bringmann, Nick Fischer. https://arxiv.org/abs/2211.07058
22. A Fine-grained Classification of Subquadratic Patterns for Subgraph Listing and Friends. Karl Bringmann, Egor Gorbachev. https://arxiv.org/abs/2404.04369
23. Certifying almost all quantum states with few single-qubit measurements. Hsin-Yuan Huang, John Preskill, Mehdi Soleimanifar. https://arxiv.org/abs/2404.07281
24. Minimum Star Partitions of Simple Polygons in Polynomial Time. Mikkel Abrahamsen, Joakim Blikstad, André Nusser, Hanwen Zhang. https://arxiv.org/abs/2311.10631
25. Clique Is Hard on Average for Sherali-Adams with Bounded Coefficients. Susanna F. de Rezende, Aaron Potechin, Kilian Risse. https://arxiv.org/abs/2404.16722
26. Deterministic factorization of constant-depth algebraic circuits in subexponential time. Somnath Bhattacharjee, Mrinal Kumar, Varun Ramanathan, Ramprasad Saptharishi, Shubhangi Saraf. https://arxiv.org/abs/2504.08063
27. The Smoothed Complexity of Policy Iteration for Markov Decision Processes. Miranda Christ, Mihalis Yannakakis. https://arxiv.org/abs/2212.00083
28. New Graph Decompositions and Combinatorial Boolean Matrix Multiplication Algorithms. Amir Abboud, Nick Fischer, Zander Kelley, Shachar Lovett, Raghu Meka. https://arxiv.org/abs/2311.09095
29. On generalized corners and matrix multiplication. Kevin Pratt. https://arxiv.org/abs/2309.03878
30. Expander Decomposition in Dynamic Streams. Arnold Filtser, Michael Kapralov, Mikhail Makarov. https://arxiv.org/abs/2211.11384
31. Data-Driven Solution Portfolios. Marina Drygala, Silvio Lattanzi, Andreas Maggiori, Miltiadis Stouras, Ola Svensson, Sergei Vassilvitskii. https://arxiv.org/abs/2412.00717
32. Quantum majority vote. Harry Buhrman, Noah Linden, Laura Mančinska, Ashley Montanaro, Maris Ozols. https://arxiv.org/abs/2211.11729
33. A Dense Neighborhood Lemma: Applications of Partial Concept Classes to Domination and Chromatic Number. Romain Bourneuf, Pierre Charbit, Stéphan Thomassé. https://arxiv.org/abs/2504.02992
34. Differential privacy and Sublinear time are incompatible sometimes. Jeremiah Blocki, Hendrik Fichtenberger, Elena Grigorescu, Tamalika Mukherjee. https://arxiv.org/abs/2407.07262
35. On the Locality of the Lovász Local Lemma. Peter Davies-Peck. https://arxiv.org/abs/2502.11690
36. Locally consistent decomposition of strings with applications to edit distance sketching. Sudatta Bhattacharya, Michal Koucký. https://arxiv.org/abs/2302.04475

37. Sum-of-Squares Lower Bounds for Sparse Independent Set. Chris Jones, Aaron Potechin, Goutham Rajendran, Madhur Tulsiani, Jeff Xu. https://arxiv.org/abs/2111.09250
38. Breaching the 2 LMP Approximation Barrier for Facility Location with Applications to k-Median. Vincent Cohen-Addad, Fabrizio Grandoni, Euiwoong Lee, Chris Schwiegelshohn. https://arxiv.org/abs/2207.05150
39. Faster Deterministic Distributed MIS and Approximate Matching. Mohsen Ghaffari, Christoph Grunau. https://arxiv.org/abs/2303.16043
40. Spectral Clustering Oracles in Sublinear Time. Grzegorz Gluch, Michael Kapralov, Silvio Lattanzi, Aida Mousavifar, Christian Sohler. https://arxiv.org/abs/2101.05549
41. New Structures and Algorithms for Length-Constrained Expander Decompositions. Bernhard Haeupler, D Ellis Hershkowitz, Zihan Tan. https://arxiv.org/abs/2404.13446
42. Determinant Maximization via Matroid Intersection Algorithms. Adam Brown, Aditi Laddha, Madhusudhan Pittu, Mohit Singh, Prasad Tetali. https://arxiv.org/abs/2207.04318
43. Active Linear Regression for ℓ_p Norms and Beyond. Cameron Musco, Christopher Musco, David P. Woodruff, Taisuke Yasuda. https://arxiv.org/abs/2111.04888
44. A Gap-ETH-Tight Approximation Scheme for Euclidean TSP. Sándor Kisfaludi-Bak, Jesper Nederlof, Karol Węgrzycki. https://arxiv.org/abs/2011.03778
45. On Approximability of Steiner Tree in ℓ_p-metrics. Henry Fleischmann, Surya Teja Gavva, Karthik C. S. https://arxiv.org/abs/2306.02189
46. List Decoding Expander-Based Codes up to Capacity in Near-Linear Time. Shashank Srivastava, Madhur Tulsiani. https://arxiv.org/abs/2504.20333
47. Lower Bound Techniques in the Comparison-Query Model and Inversion Minimization on Trees. Ivan Hu, Dieter van Melkebeek, Andrew Morgan. https://arxiv.org/abs/2211.12441
48. Hop-Constrained Oblivious Routing. Mohsen Ghaffari, Bernhard Haeupler, Goran Zuzic. https://arxiv.org/abs/2011.10446
49. Truthful and Almost Envy-Free Mechanism of Allocating Indivisible Goods: the Power of Randomness. Xiaolin Bu, Biaoshuai Tao. https://arxiv.org/abs/2407.13634
50. Parallel $(1+\epsilon)$-Approximate Multi-Commodity Mincost Flow in Almost Optimal Depth and Work. Bernhard Haeupler, Yonggang Jiang, Yaowei Long, Thatchaphol Saranurak, Shengzhe Wang. https://arxiv.org/abs/2510.20456
51. Triangle Detection in H-Free Graphs. Amir Abboud, Ron Safier, Nathan Wallheimer. https://arxiv.org/abs/2511.17224
52. Optimization with pattern-avoiding input. Benjamin Aram Berendsohn, László Kozma, Michal Opler. https://arxiv.org/abs/2310.04236
53. Approximate counting and sampling via local central limit theorems. Vishesh Jain, Will Perkins, Ashwin Sah, Mehtaab Sawhney. https://arxiv.org/abs/2108.01161
54. What Can Cryptography Do For Decentralized Mechanism Design. Elaine Shi, Hao Chung, Ke Wu. https://arxiv.org/abs/2209.14462
55. A full complexity dichotomy for immanant families. Radu Curticapean. https://arxiv.org/abs/2102.04340
56. Beyond the Quadratic Time Barrier for Network Unreliability. Ruoxu Cen, William He, Jason Li, Debmalya Panigrahi. https://arxiv.org/abs/2304.06552
57. Solving Dense Linear Systems Faster Than via Preconditioning. Michał Dereziński, Jiaming Yang. https://arxiv.org/abs/2312.08893
58. Testing Graph Properties with the Container Method. Eric Blais, Cameron Seth. https://arxiv.org/abs/2308.03289

59. Sensitivity and Dynamic Distance Oracles via Generic Matrices and Frobenius Form. Adam Karczmarz, Piotr Sankowski. https://arxiv.org/abs/2308.08870
60. Fully-Dynamic All-Pairs Shortest Paths: Improved Worst-Case Time and Space Bounds. Maximilian Probst Gutenberg, Christian Wulff-Nilsen. https://arxiv.org/abs/2001.10801
61. Non-uniform complexity via non-wellfounded proofs. Gianluca Curzi, Anupam Das. https://arxiv.org/abs/2211.16104
62. Parks and Recreation: Color Fault-Tolerant Spanners Made Local. Merav Parter, Asaf Petruschka, Shay Sapir, Elad Tzalik. https://arxiv.org/abs/2410.07844
63. Correlation Clustering with Sherali-Adams. Vincent Cohen-Addad, Euiwoong Lee, Alantha Newman. https://arxiv.org/abs/2207.10889
64. Rényi-infinity constrained sampling with d^3 membership queries. Yunbum Kook, Matthew S. Zhang. https://arxiv.org/abs/2407.12967
65. A Subpolynomial Approximation Algorithm for Graph Crossing Number in Low-Degree Graphs. Julia Chuzhoy, Zihan Tan. https://arxiv.org/abs/2202.06827
66. Near-Optimal Average-Case Approximate Trace Reconstruction from Few Traces. Xi Chen, Anindya De, Chin Ho Lee, Rocco A. Servedio, Sandip Sinha. https://arxiv.org/abs/2107.11530
67. Factorization norms and an inverse theorem for MaxCut. Igor Balla, Lianna Hambardzumyan, István Tomon. https://arxiv.org/abs/2506.23989
68. A Distanced Matching Game, Decremental APSP in Expanders, and Faster Deterministic Algorithms for Graph Cut Problems. Julia Chuzhoy. https://arxiv.org/abs/2211.10556
69. Weighted Edit Distance Computation: Strings, Trees and Dyck. Debarati Das, Jacob Gilbert, MohammadTaghi Hajiaghayi, Tomasz Kociumaka, Barna Saha. https://arxiv.org/abs/2302.04229
70. Separations in Proof Complexity and TFNP. Mika Göös, Alexandros Hollender, Siddhartha Jain, Gilbert Maystre, William Pires, Robert Robere, Ran Tao. https://arxiv.org/abs/2205.02168
71. Near Optimal Memory-Regret Tradeoff for Online Learning. Binghui Peng, Aviad Rubinstein. https://arxiv.org/abs/2303.01673
72. Share-Based Fairness for Arbitrary Entitlements. Moshe Babaioff, Uriel Feige. https://arxiv.org/abs/2405.14575
73. Testing Tensor Products of Algebraic Codes. Sumegha Garg, Madhu Sudan, Gabriel Wu. https://arxiv.org/abs/2410.22606
74. Fully Dynamic $(\Delta + 1)$ Coloring Against Adaptive Adversaries. Soheil Behnezhad, Rajmohan Rajaraman, Omer Wasim. https://arxiv.org/abs/2411.04418
75. The Submodular Santa Claus Problem. Etienne Bamas, Sarah Morell, Lars Rohwedder. https://arxiv.org/abs/2407.04824
76. Vizing's Theorem in Near-Linear Time. Sepehr Assadi, Soheil Behnezhad, Sayan Bhattacharya, Martín Costa, Shay Solomon, Tianyi Zhang. https://arxiv.org/abs/2410.05240
77. Stable Matching with Interviews. Itai Ashlagi, Jiale Chen, Mohammad Roghani, Amin Saberi. https://arxiv.org/abs/2501.12503
78. Quantum State Obfuscation from Classical Oracles. James Bartusek, Zvika Brakerski, Vinod Vaikuntanathan. https://arxiv.org/abs/2401.10200
79. Covering Approximate Shortest Paths with DAGs. Sepehr Assadi, Gary Hoppenworth, Nicole Wein. https://arxiv.org/abs/2504.11256
80. Timeliness Through Telephones: Approximating Information Freshness in Vector Clock Models. Da Qi Chen, Lin An, Aidin Niaparast, R. Ravi, Oleksandr Rudenko. https://arxiv.org/abs/2111.05450

81. Quantum soundness of testing tensor codes. Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, Henry Yuen. https://arxiv.org/abs/2111.08131
82. The Message Complexity of Distributed Graph Optimization. Fabien Dufoulon, Shreyas Pai, Gopal Pandurangan, Sriram V. Pemmaraju, Peter Robinson. https://arxiv.org/abs/2311.14811
83. One Tree to Rule Them All: Poly-Logarithmic Universal Steiner Tree. Costas Busch, Da Qi Chen, Arnold Filtser, Daniel Hathcock, D Ellis Hershkowitz, Rajmohan Rajaraman. https://arxiv.org/abs/2308.01199
84. Constant Approximation of Arboricity in Near-Optimal Sublinear Time. Jiangqi Dai, Mohsen Ghaffari, Julian Portmann. https://arxiv.org/abs/2512.18416
85. Packing Short Cycles. Matthias Bentert, Fedor V. Fomin, Petr A. Golovach, Tuukka Korhonen, William Lochet, Fahad Panolan, M. S. Ramanujan, Saket Saurabh, Kirill Simonov. https://arxiv.org/abs/2410.18878
86. Near-Optimal Algorithms for Omniprediction. Princewill Okoroafor, Robert Kleinberg, Michael P. Kim. https://arxiv.org/abs/2501.17205
87. Deletion Robust Submodular Maximization over Matroids. Paul Dütting, Federico Fusco, Silvio Lattanzi, Ashkan Norouzi-Fard, Morteza Zadimoghaddam. https://arxiv.org/abs/2201.13128
88. Towards True Work-Efficiency in Parallel Derandomization: MIS, Maximal Matching, and Hitting Set. Mohsen Ghaffari, Christoph Grunau. https://arxiv.org/abs/2504.15700
89. New Additive Spanner Lower Bounds by an Unlayered Obstacle Product. Greg Bodwin, Gary Hoppenworth. https://arxiv.org/abs/2207.11832
90. Sub-quadratic $(1+\epsilon)$-approximate Euclidean Spanners, with Applications. Alexandr Andoni, Hengjie Zhang. https://arxiv.org/abs/2310.05315
91. Knapsack with Small Items in Near-Quadratic Time. Karl Bringmann. https://arxiv.org/abs/2308.03075
92. Testing and Learning Convex Sets in the Ternary Hypercube. Hadley Black, Eric Blais, Nathaniel Harms. https://arxiv.org/abs/2305.03194
93. Rank Bounds and PIT for $\Sigma^3\Pi\Sigma\Pi^d$ circuits via a non-linear Edelstein-Kelly theorem. Abhibhav Garg, Rafael Oliveira, Akash Kumar Sengupta. https://arxiv.org/abs/2504.14729
94. On Robustness to k-wise Independence of Optimal Bayesian Mechanisms. Nick Gravin, Zhiqi Wang. https://arxiv.org/abs/2409.08547
95. Fully Dynamic Algorithms for Graph Spanners via Low-Diameter Router Decomposition. Julia Chuzhoy, Merav Parter. https://arxiv.org/abs/2601.20718
96. Triply efficient shadow tomography. Robbie King, David Gosset, Robin Kothari, Ryan Babbush. https://arxiv.org/abs/2404.19211
97. Fast Mixing in Sparse Random Ising Models. Kuikui Liu, Sidhanth Mohanty, Amit Rajaraman, David X. Wu. https://arxiv.org/abs/2405.06616
98. On Classifying Continuous Constraint Satisfaction Problems. Tillmann Miltzow, Reinier F. Schmiermann. https://arxiv.org/abs/2106.02397
99. Complexity theory of orbit closure intersection for tensors: reductions, completeness, and graph isomorphism hardness. Vladimir Lysikov, Michael Walter. https://arxiv.org/abs/2411.04639
100. Near Optimal Alphabet-Soundness Tradeoff PCPs. Dor Minzer, Kai Zhe Zheng. https://arxiv.org/abs/2404.07441
101. Naively Sorting Evolving Data is Optimal and Robust. George Giakkoupis, Marcos Kiwi, Dimitrios Los. https://arxiv.org/abs/2404.08162
102. Fast swap regret minimization and applications to approximate correlated equilibria. Binghui Peng, Aviad Rubinstein. https://arxiv.org/abs/2310.19647

103. Efficient Certificates of Anti-Concentration Beyond Gaussians. Ainesh Bakshi, Pravesh Kothari, Goutham Rajendran, Madhur Tulsiani, Aravindan Vijayaraghavan. https://arxiv.org/abs/2405.15084
104. Symmetric Perceptrons, Number Partitioning and Lattices. Neekon Vafa, Vinod Vaikuntanathan. https://arxiv.org/abs/2501.16517
105. The Proof Analysis Problem. Noel Arteche, Albert Atserias, Susanna F. de Rezende, Erfan Khaniki. https://arxiv.org/abs/2506.16956
106. Almost-Optimal Sublinear-Time Edit Distance in the Low Distance Regime. Karl Bringmann, Alejandro Cassis, Nick Fischer, Vasileios Nakos. https://arxiv.org/abs/2202.08066
107. The Communication Complexity of Approximating Matrix Rank. Alexander A. Sherstov, Andrey A. Storozhenko. https://arxiv.org/abs/2410.20094
108. Work-Efficient Parallel Derandomization II: Optimal Concentrations via Bootstrapping. Mohsen Ghaffari, Christoph Grunau. https://arxiv.org/abs/2311.13771
109. New Prophet Inequalities via Poissonization and Sharding. Elfarouk Harb. https://arxiv.org/abs/2307.00971
110. Low Treewidth Embeddings of Planar and Minor-Free Metrics. Arnold Filtser, Hung Le. https://arxiv.org/abs/2203.15627
111. Hardness of Approximation in P via Short Cycle Removal: Cycle Detection, Distance Oracles, and Beyond. Amir Abboud, Karl Bringmann, Seri Khoury, Or Zamir. https://arxiv.org/abs/2204.10465
112. Load Balancing with Dynamic Set of Balls and Bins. Anders Aamand, Jakob Bæk Tejs Knudsen, Mikkel Thorup. https://arxiv.org/abs/2104.05093
113. Pricing Query Complexity of Revenue Maximization. Renato Paes Leme, Balasubramanian Sivan, Yifeng Teng, Pratik Worah. https://arxiv.org/abs/2111.03158
114. Curve Simplification and Clustering under Fréchet Distance. Siu-Wing Cheng, Haoqiang Huang. https://arxiv.org/abs/2207.07809
115. Maximum Weight Independent Set in Graphs with no Long Claws in Quasi-Polynomial Time. Peter Gartland, Daniel Lokshtanov, Tomáš Masařík, Marcin Pilipczuk, Michał Pilipczuk, Paweł Rzążewski. https://arxiv.org/abs/2305.15738
116. Unitary Complexity and the Uhlmann Transformation Problem. John Bostanci, Yuval Efron, Tony Metger, Alexander Poremba, Luowen Qian, Henry Yuen. https://arxiv.org/abs/2306.13073
117. Embedding Probability Distributions into Low Dimensional ℓ_1: Tree Ising Models via Truncated Metrics. Moses Charikar, Spencer Compton, Chirag Pabbaraju. https://arxiv.org/abs/2312.02435
118. Quartic Samples Suffice for Fourier Interpolation. Zhao Song, Baocheng Sun, Omri Weinstein, Ruizhe Zhang. https://arxiv.org/abs/2210.12495
119. Learning the structure of any Hamiltonian from minimal assumptions. Andrew Zhao. https://arxiv.org/abs/2410.21635
120. Random Reed-Solomon Codes and Random Linear Codes are Locally Equivalent. Matan Levi, Jonathan Mosheiff, Nikhil Shagrithaya. https://arxiv.org/abs/2406.02238
121. Tolerant testing of stabilizer states with a polynomial gap via a generalized uncertainty relation. Zongbo Bao, Philippe van Dordrecht, Jonas Helsen. https://arxiv.org/abs/2410.21811
122. Computing the 5-Edge-Connected Components in Linear Time. Evangelos Kosinas. https://arxiv.org/abs/2311.04865
123. Multi-Pass Streaming Lower Bounds for Approximating Max-Cut. Yumou Fei, Dor Minzer, Shuo Wang. https://arxiv.org/abs/2503.23404
124. Formula Size-Depth Tradeoffs for Iterated Sub-Permutation Matrix Multiplication. Benjamin Rossman. https://arxiv.org/abs/2406.16015

125. Approximately Counting and Sampling Hamiltonian Motifs in Sublinear Time. Talya Eden, Reut Levi, Dana Ron, Ronitt Rubinfeld. https://arxiv.org/abs/2503.09810
126. On Pigeonhole Principles and Ramsey in TFNP. Siddhartha Jain, Jiawei Li, Robert Robere, Zhiyang Xun. https://arxiv.org/abs/2401.12604
127. New SDP Roundings and Certifiable Approximation for Cubic Optimization. Jun-Ting Hsieh, Pravesh K. Kothari, Lucas Pesenti, Luca Trevisan. https://arxiv.org/abs/2310.00393
128. Towards Optimal Output-Sensitive Clique Listing or: Listing Cliques from Smaller Cliques. Mina Dalirrooyfard, Surya Mathialagan, Virginia Vassilevska Williams, Yinzhan Xu. https://arxiv.org/abs/2307.15871
129. Online Discrepancy with Recourse for Vectors and Graphs. Anupam Gupta, Vijaykrishna Gurunathan, Ravishankar Krishnaswamy, Amit Kumar, Sahil Singla. https://arxiv.org/abs/2111.06308
130. Sub-Exponential Lower Bounds for Branch-and-Bound with General Disjunctions via Interpolation. Max Gläser, Marc E. Pfetsch. https://arxiv.org/abs/2308.04320
131. Gradient descent for unbounded convex functions on Hadamard manifolds and its applications to scaling problems. Hiroshi Hirai, Keiya Sakabe. https://arxiv.org/abs/2404.09746
132. A lower bound on the space overhead of fault-tolerant quantum computation. Omar Fawzi, Alexander Müller-Hermes, Ala Shayeghi. https://arxiv.org/abs/2202.00119
133. Beating Bellman's Algorithm for Subset Sum. Karl Bringmann, Nick Fischer, Vasileios Nakos. https://arxiv.org/abs/2410.21942
134. Tight Guarantees for Multi-unit Prophet Inequalities and Online Stochastic Knapsack. Jiashuo Jiang, Will Ma, Jiawei Zhang. https://arxiv.org/abs/2107.02058
135. Fitting Metrics and Ultrametrics with Minimum Disagreements. Vincent Cohen-Addad, Chenglin Fan, Euiwoong Lee, Arnaud de Mesmay. https://arxiv.org/abs/2208.13920
136. Fast Static and Dynamic Approximation Algorithms for Geometric Optimization Problems: Piercing, Independent Set, Vertex Cover, and Matching. Sujoy Bhore, Timothy M. Chan. https://arxiv.org/abs/2407.20659
137. Planar Multiway Cut with Terminals on Few Faces. Sukanya Pandey, Erik Jan van Leeuwen. https://arxiv.org/abs/2506.23399
138. Settling the Pass Complexity of Approximate Matchings in Dynamic Graph Streams. Sepehr Assadi, Soheil Behnezhad, Christian Konrad, Kheeran K. Naidu, Janani Sundaresan. https://arxiv.org/abs/2407.21005
139. Algorithms and Hardness for Multidimensional Range Updates and Queries. Joshua Lau, Angus Ritossa. https://arxiv.org/abs/2101.02003
140. Lifting to Parity Decision Trees Via Stifling. Arkadev Chattopadhyay, Nikhil S. Mande, Swagato Sanyal, Suhail Sherif. https://arxiv.org/abs/2211.17214
141. Polygon Placement Revisited: (Degree of Freedom + 1)-SUM Hardness and an Improvement via Offline Dynamic Rectangle Union. Marvin Künnemann, André Nusser. https://arxiv.org/abs/2111.02544
142. Near-Optimal Deterministic Vertex-Failure Connectivity Oracles. Yaowei Long, Thatchaphol Saranurak. https://arxiv.org/abs/2205.03930
143. Double Coverage with Machine-Learned Advice. Alexander Lindermayr, Nicole Megow, Bertrand Simon. https://arxiv.org/abs/2103.01640
144. Quasi-polynomial time approximation schemes for the Maximum Weight Independent Set Problem in H-free graphs. Maria Chudnovsky, Marcin Pilipczuk, Michał Pilipczuk, Stéphan Thomassé. https://arxiv.org/abs/1907.04585
145. Sublinear-Time Algorithms for Max Cut, Max E2Lin(q), and Unique Label Cover on Expanders. Pan Peng, Yuichi Yoshida. https://arxiv.org/abs/2210.12601

146. Adaptive Approximation Schemes for Matching Queues. Alireza AmaniHamedani, Ali Aouad, Amin Saberi. https://arxiv.org/abs/2501.08775
147. All-Pairs Shortest Paths with Few Weights per Node. Amir Abboud, Nick Fischer, Ce Jin, Virginia Vassilevska Williams, Zoe Xi. https://arxiv.org/abs/2506.20017
148. New Graph and Hypergraph Container Lemmas with Applications in Property Testing. Eric Blais, Cameron Seth. https://arxiv.org/abs/2403.18777
149. Stability is Stable: Connections between Replicability, Privacy, and Adaptive Generalization. Mark Bun, Marco Gaboardi, Max Hopkins, Russell Impagliazzo, Rex Lei, Toniann Pitassi, Satchit Sivakumar, Jessica Sorrell. https://arxiv.org/abs/2303.12921
150. Determinantal Sieving. Eduard Eiben, Tomohiro Koana, Magnus Wahlström. https://arxiv.org/abs/2304.02091
151. Pattern Matching on Grammar-Compressed Strings in Linear Time. Moses Ganardi, Paweł Gawrychowski. https://arxiv.org/abs/2111.05016
152. Top-Down Lower Bounds for Depth-Four Circuits. Mika Göös, Artur Riazanov, Anastasia Sofronova, Dmitry Sokolov. https://arxiv.org/abs/2304.02555
153. Maximum Bipartite Matching in $n^{2+o(1)}$ Time via a Combinatorial Algorithm. Julia Chuzhoy, Sanjeev Khanna. https://arxiv.org/abs/2405.20861
154. Planar Disjoint Paths, Treewidth, and Kernels. Michał Włodarczyk, Meirav Zehavi. https://arxiv.org/abs/2307.06792
155. Local Computation Algorithms for Maximum Matching: New Lower Bounds. Soheil Behnezhad, Mohammad Roghani, Aviad Rubinstein. https://arxiv.org/abs/2311.09359
156. Single-Sample Prophet Inequalities via Greedy-Ordered Selection. Constantine Caramanis, Paul Dütting, Matthew Faw, Federico Fusco, Philip Lazos, Stefano Leonardi, Orestis Papadigenopoulos, Emmanouil Pountourakis, Rebecca Reiffenhäuser. https://arxiv.org/abs/2111.03174
157. Simpler and Higher Lower Bounds for Shortcut Sets. Virginia Vassilevska Williams, Yinzhan Xu, Zixuan Xu. https://arxiv.org/abs/2310.12051
158. Nearly-Linear Time Seeded Extractors with Short Seeds. Dean Doron, João Ribeiro. https://arxiv.org/abs/2411.07473
159. Dimension-Preserving Reductions Between SVP and CVP in Different p-Norms. Divesh Aggarwal, Yanlin Chen, Rajendra Kumar, Zeyong Li, Noah Stephens-Davidowitz. https://arxiv.org/abs/2104.06576
160. Partial Synchrony for Free? New Upper Bounds for Byzantine Agreement. Pierre Civit, Muhammad Ayaz Dzulfikar, Seth Gilbert, Rachid Guerraoui, Jovan Komatovic, Manuel Vidigueira, Igor Zablotchi. https://arxiv.org/abs/2402.10059
161. Tight Streaming Lower Bounds for Deterministic Approximate Counting. Yichuan Wang. https://arxiv.org/abs/2406.12149
162. Sumsets, 3SUM, Subset Sum: Now for Real!. Nick Fischer. https://arxiv.org/abs/2410.21953
163. Random Gabidulin Codes Achieve List Decoding Capacity in the Rank Metric. Zeyu Guo, Chaoping Xing, Chen Yuan, Zihan Zhang. https://arxiv.org/abs/2404.13230
164. Canonical Decompositions of 3-Connected Graphs. Johannes Carmesin, Jan Kurkofka. https://arxiv.org/abs/2304.00945
165. Online Sorting and Translational Packing of Convex Polygons. Anders Aamand, Mikkel Abrahamsen, Lorenzo Beretta, Linda Kleist. https://arxiv.org/abs/2112.03791
166. The Reachability Problem for Petri Nets is Not Primitive Recursive. Jérôme Leroux. https://arxiv.org/abs/2104.12695
167. Sharp Thresholds in Random Simple Temporal Graphs. Arnaud Casteigts, Michael Raskin, Malte Renken, Viktor Zamaraev. https://arxiv.org/abs/2011.03738

168. Tensor Reconstruction Beyond Constant Rank. Shir Peleg, Amir Shpilka, Ben Lee Volk. https://arxiv.org/abs/2209.04177
169. Polynomial Bounds for the Graph Minor Structure Theorem. Maximilian Gorsky, Michał T. Seweryn, Sebastian Wiederrecht. https://arxiv.org/abs/2504.02532
170. A Computational Separation Between Quantum No-cloning and No-telegraphing. Barak Nehoran, Mark Zhandry. https://arxiv.org/abs/2302.01858
171. Swap cosystolic expansion. Yotam Dikstein, Irit Dinur. https://arxiv.org/abs/2312.15325
172. How to Use Quantum Indistinguishability Obfuscation. Andrea Coladangelo, Sam Gunn. https://arxiv.org/abs/2311.07794
173. On $(1+\varepsilon)$-Approximate Flow Sparsifiers. Yu Chen, Zihan Tan. https://arxiv.org/abs/2310.07857
174. The NFA Acceptance Hypothesis: Non-Combinatorial and Dynamic Lower Bounds. Karl Bringmann, Allan Grønlund, Marvin Künnemann, Kasper Green Larsen. https://arxiv.org/abs/2311.10204
175. Lossy Planarization: A Constant-Factor Approximate Kernelization for Planar Vertex Deletion. Bart M. P. Jansen, Michał Włodarczyk. https://arxiv.org/abs/2202.02174
176. Improved Distance (Sensitivity) Oracles with Subquadratic Space. Davide Bilò, Shiri Chechik, Keerti Choudhary, Sarel Cohen, Tobias Friedrich, Martin Schirneck. https://arxiv.org/abs/2408.10014
177. Recognizing Sumsets is NP-Complete. Amir Abboud, Nick Fischer, Ron Safier, Nathan Wallheimer. https://arxiv.org/abs/2410.18661
178. Highway Dimension: a Metric View. Andreas Emil Feldmann, Arnold Filtser. https://arxiv.org/abs/2412.20490
179. Tree Independence Number IV. Even-hole-free Graphs. Maria Chudnovsky, Peter Gartland, Sepehr Hajebi, Daniel Lokshtanov, Sophie Spirkl. https://arxiv.org/abs/2407.08927
180. Good Quantum LDPC Codes with Linear Time Decoders. Irit Dinur, Min-Hsiu Hsieh, Ting-Chun Lin, Thomas Vidick. https://arxiv.org/abs/2206.07750
181. Having Hope in Hops: New Spanners, Preservers and Lower Bounds for Hopsets. Shimon Kogan, Merav Parter. https://arxiv.org/abs/2211.06920
182. Maximum Length-Constrained Flows and Disjoint Paths: Distributed, Deterministic and Fast. Bernhard Haeupler, D Ellis Hershkowitz, Thatchaphol Saranurak. https://arxiv.org/abs/2111.01422
183. Computational Hardness of the Hylland-Zeckhauser Scheme. Thomas Chen, Xi Chen, Binghui Peng, Mihalis Yannakakis. https://arxiv.org/abs/2107.05746
184. Breaking the $3/4$ Barrier for Approximate Maximin Share. Hannaneh Akrami, Jugal Garg. https://arxiv.org/abs/2307.07304
185. Almost Tight Additive Guarantees for k-Edge-Connectivity. Nikhil Kumar, Chaitanya Swamy. https://arxiv.org/abs/2506.20906
186. Fully Dynamic k-Median with Near-Optimal Update Time and Recourse. Sayan Bhattacharya, Martín Costa, Ermiya Farokhnejad. https://arxiv.org/abs/2411.03121
187. The AC^0-Complexity Of Visibly Pushdown Languages. Stefan Göller, Nathan Grosshans. https://arxiv.org/abs/2302.13116
188. Spectral Guarantees for Adversarial Streaming PCA. Eric Price, Zhiyang Xun. https://arxiv.org/abs/2408.10332
189. Incremental SSSP for Sparse Digraphs Beyond the Hopset Barrier. Rasmus Kyng, Simon Meierhans, Maximilian Probst Gutenberg. https://arxiv.org/abs/2110.11712
190. Small space and streaming pattern matching with k edits. Tomasz Kociumaka, Ely Porat, Tatiana Starikovskaya. https://arxiv.org/abs/2106.06037