Proof or Bluff?

Students, Faculty, and Generative AI in Theory of Computing Courses

Garrett Vowinkel, Nero Li, Yubin Kim, Michael Shindler, Ryan Dougherty

Why Theory of Computing is Difficult with GenAI

Proof-Heavy, Not Code

ToC centers on formal proofs and advanced mathematics, differing from the programming-centric contexts of most prior GenAI-in-CS research.

Detection Tools Do Not Work

AI detectors lack accuracy, are easily bypassed, and demonstrate bias by unfairly flagging non-native English writers.

Strong Incentive to Offload

Abstract material tempts students to use GenAI as a shortcut; some exhibit blind trust, lacking the ability to fact-check the outputs.

Models Keep Improving

Reasoning models score near-perfectly on ToC exams, questioning the long-term validity of traditional assessment methods.

Three Classes of RQs

Detection

- Can students identify GenAI-written ToC papers and reviews?
- Can faculty identify GenAI-written ToC text?

Usage and Learning

- Which models do students choose, and how do they use them?
- Does usage relate to self-efficacy and knowledge gain?
- What usage do students recommend?

Content Quality

- Can GenAI produce papers and reviews equal in quality to students'?
- Why do students rate GenAI work higher?

Method

Project Cont

Semester-long
across two The

, o3-mini) and
iew process.

Participants

41 students at
faculty membe
judgments.

allis, Spearman

Qualitative A

Inductive thema

sensus.

Prefix Detection for Probably Approximately Correct Analytics: A Theoretical Exploration

Anonymous*
Anonymous
Anonymous@westpoint.edu
Anonymous@westpoint.edu

ABSTRACT

In this paper, we investigate a suite of formal language and automata problems centered on detecting the presence of the acronym "PAC" (and its reversal "CAP") as a prefix in strings over the alphabet $\{A, C, P\}$. We first derive a regular expression describing all strings that *do not* begin with "PAC." Next, we construct and minimize a deterministic finite automaton (DFA) for the language consisting of strings whose prefix is either "PAC" or "CAP." We then present an algorithm that, given any DFA, checks for acceptance of any string with either "PAC" or "CAP" as a prefix, and analyze the complexity of this procedure. We generalize this to NFAs, focusing on whether the NFA accepts infinitely or finitely many strings outside those prefixes. Subsequently, we examine the regularity of languages tracking the occurrences of "PAC," and we explore the behavior of prefix-closure operations on arbitrary regular languages. Our results provide both theoretical insights and practical applications, such as fast scanning of text streams.

ACM Reference Format:

analyze the resulting solution in terms of complexity and efficiency. Our results facilitate a deeper understanding of regular languages, prefix closures, and how automata can be systematically designed to respond to specific patterns.

The remainder of this paper is organized as follows:

- Section 2 reviews recent developments and existing literature in theoretical computer science on finite automata and pattern detection.
- Section 3 formalizes our notation and definitions.
- Section 4 presents our main proofs and constructions corresponding to the problems posed.
- Section 5 describes our experimental setup, methodology, and results.
- Section 6 discusses the implications of our findings and potential extensions.
- Section 7 outlines future work possibilities.
- Section 8 concludes the paper by summarizing our contributions.

A scaffolded, three-phase paper project

Phase 1: Regular Languages

DFA/NFA, Pumping Lemma, DFA minimization.

Phase 2: Context-Free

CFGs, pushdown automata, algorithmic experiments.

Phase 3: Decidable

Turing machines, decidability and reductions.

Staged Deliverables

Plan, check-in, draft, peer review, final, graded for process.

Double-blind Injection

GenAI papers/reviews entered anonymously; disclosures falsely claimed no AI use.

Required Disclosure

Every group declares model, prompts, and modifications, even when no AI was used.

Students cannot reliably spot GenAI papers

Detection Rates

6.9% at I1 (SLAC) vs. **31.8%** at I2 (R1).

Statistical Significance

I2 > I1 difference is significant ($p < 0.001$).

I1 Detection Rate by Model

Performance varied slightly by model: **4o:** 2.7% | **o1:** 8.6% | **o3-mini-high:** 10.2%.

0% False-Positive Rate (I1)

Students never flagged an authentic paper; they defaulted to trusting the disclosure statement.

Faculty do better, but flag the innocent

Detection Success

63.7% of faculty responses correctly identified GenAI.

False Positive Risk

22.8% of student papers “wrongly” flagged.

Faculty: Verifiable Artifacts

- Hallucinated / non-existent citations.
- Generic, non-novel content.
- Reliable today, but models are fixing these fastest.

Students: Surface Cues & Trust

- Superfluous / "robotic" word choice.
- Formatting placeholders ([language], Section ??).
- Diagram irregularities, list-heavy structure.
- Often: blind trust in disclosure statements.

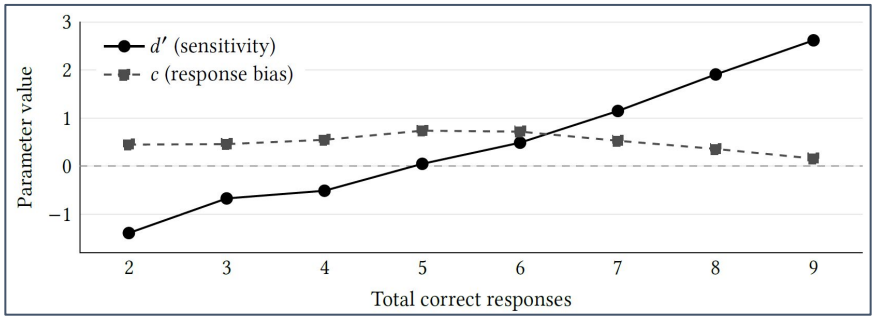
Detection Deeper Look

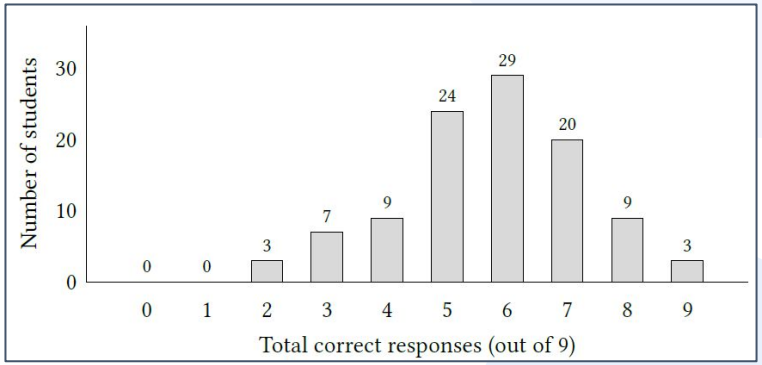
Detection Performance

- Sensitivity was negative for the modal-score cluster, worse than chance.
- 24% of completers made zero correct identifications.

Base-Rate Strategies

10.6% marked "accurate" on all 9 papers, a strategy scoring 6/9 but showing no true detection ability.





Low Reliability Threshold

Cronbach's alpha across the papers was **0.36**, well below the 0.70 reliability threshold → student detection is inconsistent and unreliable.

What students chose, and how they used it

	GenAI Model									
	OpenAI (GPT)					Anthropic	Poe	xAI	GitHub	Google
	4o(-mini)	3.5	o1	o3	o4-mini	Claude	Poe.AI	Grok	Copilot	Gemini 2.0
Phase 1	18	4	2	0	0	0	2	0	2	0
Phase 2	10	0	2	4	0	0	0	2	0	2
Phase 3	11	0	2	3	2	2	0	0	2	0
All	**39**	**4**	**6**	**7**	**2**	**2**	**2**	**2**	**4**	**2**

Engagement Patterns

- **37.3%**: Direct copy-paste; little-to-no change.
- **62.7%**: Scaffolding or directional guidance.

Category	Representative Comments
Category 1: Little to No Change	*"Absolutely nothing, it did the formatting properly, and the few times I asked for feedback on my responses, I either took the feedback or disregarded it."*
	"I didn't modify the Python code it gave me at all. I simply tested it and reprinted it if it worked or not."
Category 2: Guidance/Scaffolding	*"It helped flush out my rough solutions to certain problems and guide me in the right direction."*
	"The outputs served more as guiding documents rather than the definitive 'ground truth.' "

Summary of Usage Behavior

While a majority used AI for scaffolding, a significant minority bypassed critical thinking entirely.

Helpful in feeling, not in learning

Usage vs. Perception

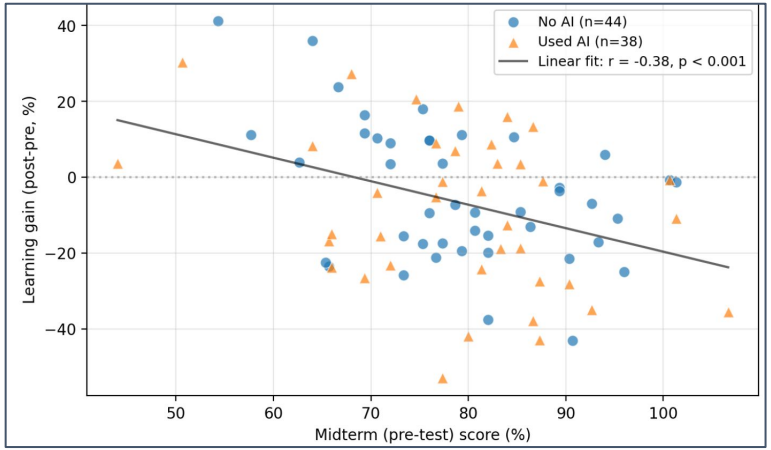
Moderate, significant association (p = 0.001). More use felt more helpful to students.

Usage vs. Self-Efficacy

No significant association (p = 0.948). Model choice moved efficacy (p = 0.007) but on tiny samples (n = 2).

Academic Impact: Learning Gains

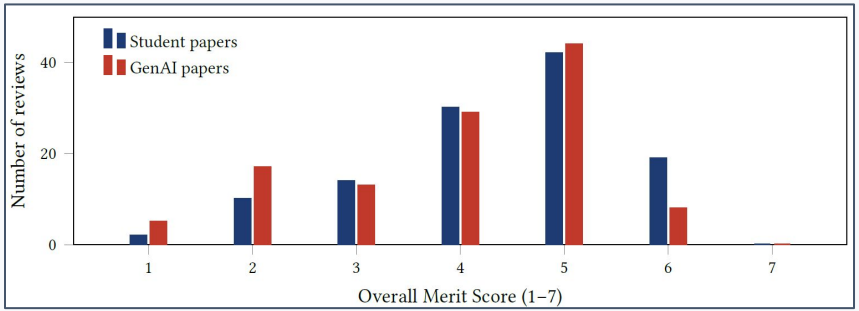
Mean exam learning gain was -9.1%, with no significant effect of usage (p = 0.427).

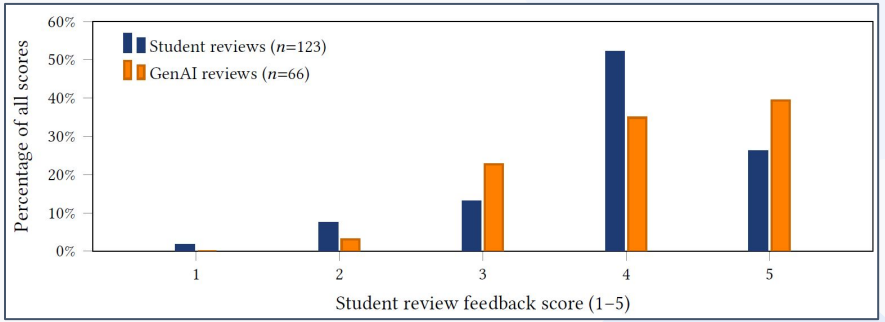


GenAI work passes for student work

Comparable Grades

76.1% (GenAI) vs 81.6% (student), not significant (p = 0.438). o1 earned only a C+, far below its A+ on standard ToC exams.





Quality Drops with Difficulty

While AI can mimic student work in standard settings, hard ToC material remains **AI-resistant**.

Recommendations for ToC educators

1. Move past detection

Neither group reliably detects GenAI. Use AI-robust formats - staged artifacts, oral defenses, "error-hunt" tasks, lecture-specific notation.

2. Guide use per assignment

Replace blanket policies with an allowed / discouraged / prohibited table tied to learning objectives, plus required disclosure.

3. Teach evaluation as practice

Short, recurring 10-15 minute exercises - find the error, judge each review comment, compare against a course method.

4. Distinguish among models

"GenAI" is not monolithic. Recommend reasoning models where allowed, and warn that all still hallucinate citations.

Effective GenAI integration requires moving from a detection-based mindset to one focused on intentional instructional design.

Limitations

Sample and scope

Two U.S., English-language sites; just 11 faculty. Findings are hypothesis-generating.

Learning-gain confound

-9.1% gain reflects exam-difficulty scaling and regression to the mean, not a clean learning signal.

Model and familiarity drift

Cues came from Spring 2025 ChatGPT.

Self-report bias

Usage and self-efficacy are single-item reports, likely compressing true reliance downward.

No control for Usage

Students may and did have wildly different experience with using GenAI models.

Availability drift

The interaction with “high end” models has shifted, where paid models before now have equivalent free versions.

Conclusion: Detection is dead, Design is the answer

Detection Fails

I1 6.9%, I2 31.8%, faculty 22.8% false positives. People cannot reliably tell.

Usage != Learning

More use felt more helpful but produced no measurable self-efficacy or knowledge gain.

Convincing Quality

GenAI papers grade comparably and its reviews are even rated higher despite being shallow.

The Path Forward for Educators

Stop policing for AI. Instead, pivot to intentional design:

- Give explicit per-assignment guidance.
- Teach metacognitive evaluation as a recurring activity.
- Distinguish among models and their limitations.
- Build assignments that stay meaningful whether or not GenAI is present.