Is Solving Better Than Evaluating GenAI Solutions?

Ethan Dickey • Marios Mertzanidis • Alexandros Psomas

Contact: dickeye at purdue.edu | *Full paper under review*
Materials + templates: cs.purdue.edu/homes/dickeye

What we tested

Instead of having students solve every homework problem from scratch, we tested a "TA-mode" activity: students generate a GenAI (ChatGPT-4o) solution and then evaluate its algorithm + proof/correctness.

Study design

- Context: Junior-level Analysis of Algorithms (induction, D&C, DP, graphs, max-flow, reductions).
- Sample: N = 220 (of 227 enrolled; exclusions: audits/incompletes/missed final).
- Grouping: Students self-reported study pods (≤6). Pods were randomized to Group A or B.
- A/B crossover: 6 biweekly homework assignments → crossover after the midterm.
- "Dosage": Each HW had 4 exercises. Exercises 1-3 identical across groups (30 pts each). Exercise 4 differed by condition and was worth 10 pts (intended to avoid grade advantage).
- Evaluate: students prompted ChatGPT-4o, submitted links + screenshot, then evaluated algorithm + proof; proof bugs classified as minor vs major + justification.
- Grading: Instructor + 7 GTAs + 17 UTAs; rubrics aligned per exercise; the hardest Exercise 4 graded by GTAs.

Results (all primary + key secondary outcomes)

(0) Baseline equivalence / internal validity

Baseline (HW1-3 non-AI problems, rescaled 0-10) ($\mu \pm SD$): A 7.91 ± 1.36 vs B 8.04 ± 1.35

- t(218) = -0.72, p = .47; Mann-Whitney p = .407 (trivial effect); KS p = .62

(1) Primary outcomes: exams + course grade (statistically indistinguishable; negligible effects)

Midterm (0-101) ($\mu \pm SD$): A 60.7 ± 14.5 vs B 61.8 ± 13.4

- t(218) = -0.58, p = .56; 95% CI [-4.8, 2.6]; d = -0.08

Final (0-113): A 58.2 ± 17.1 vs B 59.1 ± 16.7

- t(218) = -0.39, p = .70; 95% CI [-5.4, 3.6]; d = -0.05

Course total (%) : A 63.9 ± 12.1 vs B 65.4 ± 11.4

- t(218) = -0.95, p = .34; 95% CI [-4.6, 1.6]; d = 0.13

Change score (Final - Midterm, normalized to 10 pts): A -0.86 ± 1.3 vs B -0.89 ± 1.1

- t(218) = 0.19, p = .85; 95% CI [-0.3, 0.3]; d = 0.03

Letter grades: $\chi^2(9)$ = 9.69, p = .38 (V = .07)

(2) Transfer: GenAI-aligned exam item(s)

Aligned midterm item (0-15): median = 0 in both groups (floor effect)

(2) Transfer: GenAI-aligned exam item(s) – continued

- Mann-Whitney U = 5644.5, p = .30; KS p = .95

Aligned final item (0-15): higher spread, but no separation

- Mann-Whitney U = 6191.5, p = .76; KS p = .90

Strategy adoption: no group differences (midterm Z = -0.51, p = .61; final Z = 0.83, p = .41)

(3) Final exam by category (timing check)

First-half problems (% score): U = 5624.5, p = .368

Second-half problems (% score): U = 5932.0, p = .804

(Overlap for "hybrid" items.)

(4) Homework trajectory (where we saw differences)

Difference-of-differences across course halves (positive = higher during the half when grading GenAI):

GPT problems swing ($\mu \pm SD$): A +18 ± 25 vs B +9 ± 23

- Mann-Whitney U = 7046, p = .035 (r = .14); confirmatory t(218) = 2.90, p = .004 (d = .39)

Non-GPT problems swing ($\mu \pm SD$): A +3 ± 14 vs B -3 ± 13

- Mann-Whitney U = 7119, p = .024 (r = .15)

Interpretation: modest deltas appear to track syllabus difficulty more than a treatment benefit.

(5) Affective / perception measures

Midterm self-efficacy survey (N = 208): no between-group differences

- Comfort: U = 5788, p = .374; Confidence: U = 5152, p = .561

Within-student topic differences were large (Asymptotic Analysis and D&C higher than DP/Greedy/Graphs);

Comfort-Confidence assoc: Spearman ρ = 0.714, p < .001

Post-final perceptions (N = 200):

Helpfulness: median = 3/5 ("neutral") in both groups

- U = 5132, p = .728

Study-habit change: 77% No, 16% Kind of, 7% Yes

- $\chi^2(2)$ = 0.97, p = .616

Helpfulness vs study-habit change: H(2) = 17.74, p < .001 (those reporting changes rated it more helpful)

What GenAI got wrong (useful for designing tasks)

Across topics, GenAI outputs were plausible but systematically wrong:

- Wrong algorithm design (D&C, DP, graphs, reductions)
- Wrong proof of correctness (induction, flow)

Takeaway

In this implementation, structured GenAI-evaluation did not detectably change exam or course outcomes.

Any upside may require prompts that force "repair + reflect," not critique alone.

Limitations

- One homework was modified late (subset-sum → subset-product).
- One final exam item admitted an unintended easier solution path.
- Exposure was limited to 6 biweekly homeworks with only one condition-dependent exercise each.

10-15min quick start (for instructors)

1) Start small: convert ONE homework exercise per set into GenAI-evaluation (keep the rest traditional).

2) Choose problems where GenAI is likely to produce plausible-but-wrong reasoning (not trivial typos).

3) Require evidence: student prompt + GenAI output + evaluation + a repaired version + 2-sentence reflection.

4) In class, debrief one recurring failure mode (e.g., wrong DP state; missing base case; reduction ignores encoding size).

Copy/paste student interactions (template)

Create a new chat in ChatGPT-4o and ask it for a succinct solution to the problem (algorithm + correctness/proof + runtime).

Submit:

- a link to your chat (Share Chat) and a legible screenshot of the response, AND
- your evaluation of the solution.

Evaluation requirements:

1) If the algorithm is correct, state why (briefly).

2) If the algorithm is incorrect, identify the FIRST incorrect step/claim and explain what fails.

3) If the proof/correctness argument has a bug, point to the bug and classify it:

- Minor bug (e.g., small calculation/wording issue; core logic intact), OR
- Major bug (a crucial step is wrong or missing).

4) Repair: write the corrected algorithm/proof skeleton (short but complete).

5) Reflection (2 sentences): What misconception did the GenAI encode? What will you check next time?

(Optional) Suggested checklist/rubric for students

Score each category 0-2 (0 = incorrect/absent, 1 = partial, 2 = correct + justified). Total /10.

A) Problem setup (symbols, constraints, goal)

B) Core idea / paradigm (DP state, greedy choice, reduction mapping, flow construction)

C) Correctness reasoning (invariant/induction/reduction validity; base cases and edge cases)

D) Complexity analysis (big-O + justification)

E) Clarity/completeness (enough detail to execute/verify)

Required lines:

- First incorrect step/claim: ______________________
- Minimal repair: ______________________________
- Misconception encoded (1-2 sentences): __________

Mini ex. 1 (DP failure mode): wrong state

Problem: Given n $\{+,-\}^{\ell}$ strings, choose a subset/order to maximize total length so prefix-sum is never < 0 and final sum = 0. (Motivation: roller coaster construction.)

GenAI claim: "For each part p, compute Δ(p)=#+ − #− and L(p)=|p|; sort parts by Δ descending. Let dp[h] be the max total ride length that ends at height h (dp[0]=0). For each part p and each h, if dp[h] exists and h+Δ(p) ≥ 0, set dp[h+Δ(p)] = max(dp[h+Δ(p)], dp[h] + L(p)).

Return dp[0] as the best ride that ends on the ground."

Evaluation prompt: What assumption fails first? (Hint: what can happen *inside* a part, not just between parts?) What extra quantity must be tracked/checked to make it valid? Sketch the high-level corrected state/recurrence.

Reflection: what misconception is revealed?

[Instructor Note] Misconception: Net sum is insufficient; validity depends on current height + each part's min-prefix height. Typical fix: preprocess by min-prefix and use DP with a height dimension (2D DP).

Mini ex. 2 (reductions failure mode): encoding size

Problem: Give a poly-time reduction from SUBSET-SUM to SUBSET-PRODUCT.

GenAI claim: "Reduce SUBSET-SUM to SUBSET-PRODUCT by mapping a_i ↦ 2^{a_i} and target T ↦ 2^T."

Your task: Is this a valid poly-time many-one reduction? State yes/no and if no, give the first requirement that fails and explain how you would repair the reduction / redesign the mapping.

[Instructor Note] Misconception: The output instance size can blow up: representing 2^{a_i} needs Θ(a_i) bits, which can be exponential in the input length (log a_i). This violates the polynomial-time/size requirement under standard encodings.