Human or Machine? Assessing AI's Ability to Generate Game-Theory Questions*

Benjamin Golub† Annie Liang‡ Marciano Siniscalchi§

February 15, 2026

Abstract

AI models now excel at solving difficult applied mathematics problems; we ask how well they can compose such problems, focusing on undergraduate game theory. Adapting the Turing (1950) test to problem generation, we collect problems from professors and GPT-5, standardizing presentation so evaluation focuses on content rather than style. Sixty-seven experts—undergraduate and graduate students who have taken game theory—classify problems as human- or LLM-generated. We find that AI output is indistinguishable to any single evaluator yet different in aggregate. Individually, evaluators perform at chance (mean accuracy 50.9%). However, pooling 2,680 classifications rejects the null that the two distributions are identical ($p = 0.014$). The signal resides in solutions, not problem statements: restricting to evaluators who observe solutions and report medium or high confidence, pooled accuracy rises to 53.4% ($p = 0.0006$), while without solutions we cannot reject the null. We train a classifier to distinguish the problem sources; the strongest objective feature separating human problems is the solution word count to problem word count ratio: human-authored problems tend to require more reasoning per unit of setup. We discuss implications of our findings for organizations that delegate knowledge work to AI.

*We are grateful to Ivan Canay and Alex Frankel for helpful comments on this paper, to Aidan Goth and Samyak Jain for excellent research assistance, and to Shachar Kariv, Shengwu Li, Xiao Lin, Elliot Lipnowski, and Evan Sadler for sharing problems with us.

†Department of Economics, Northwestern

‡Department of Economics, Northwestern

§Department of Economics, Northwestern

1 Introduction

Artificial intelligence models can now produce sophisticated knowledge work products. Examples include legal briefs, engineering reviews, and materials for scientific instruction. As a result, practitioners in many areas can use generative AI models to save considerable time in contributing to such artifacts. Often, they can simply ask AI to perform the work they would previously have done by hand, vet the output, and pass it on if it looks acceptable. However, even when each individual artifact looks acceptable, the distribution of material produced by this process may differ systematically from the distribution of output that experienced practitioners would have hand-crafted. Such drift obviously may affect the value of the outputs produced by the work process. More subtly, knowledge work outputs produced in an organization are also training material for other workers. Drift in the distribution of these outputs changes what workers are exposed to, and thus the transmission of skills and practices in an organization.

Motivated by these considerations, we define two criteria that are important to assess when AI outputs replace human-crafted work in a particular domain. First, can AI models match the distribution of human expert output? Second, when differences arise, how detectable are they, and what is their nature? The first criterion is a demanding test of AI capabilities: in order to encompass the production possibilities frontier of a type of human expert in a certain task, an AI system must be capable of satisfying this criterion under appropriate use. The second question, of detectability, is relevant for an organization's capacity to respond to AI systems that do not yet satisfy the first criterion.

We study these issues in a setting that we think is an interesting laboratory: writing problem-set and exam questions in undergraduate-level game theory. These texts fit the mold described above: they are key outputs produced by educators, and they shape the understanding of others, including the next generation of experts. It is also a setting where it was not a priori clear to us how well frontier AI models can "pass." On the one hand, frontier models can solve problems at this level as well as human experts, and they produce plausible outputs when asked to compose problems.

Yet problem composition is a demanding task that involves considerable craft. It is typically performed by instructors more skilled and experienced than most problem-solvers, and the standards for good design choices are hard to codify in comparison to the standards for problem-solving. Problem composition is thus a good domain near the boundary of current AI capabilities. At the same time, it is favorable to frontier models in an important respect: the content of undergraduate game theory is highly standardized and publicly codified, with many public examples of good craft and limited hidden context.

For our experiment, we collect a corpus of human-authored problems from expert instructors, and we also generate a corpus of problems by prompting OpenAI's GPT-5. A style normalizer rewrites all problems in a common voice, with the aim of making substantive content, rather than presentation, the focus of evaluation. We recruit 67 evaluators (35 economics PhD students and 32 undergraduates), each of whom classifies 40 problems as human- or LLM-generated.

The experiment varies what evaluators observe and how the LLM-written problems are generated. One treatment arm varies whether solutions are shown.[1] This distinction is important because it allows us to locate where any detectable signal resides. The other treatment arm varies whether we use a straightforward prompt with low reasoning, or a long prompt with high reasoning, which we hypothesized would affect how well AI could pass the test. We collect classifications for each problem and confidence ratings for each classification.

We deliberately avoid precommitting to any particular notion of problem quality. Problems are very high-dimensional, and we do not want to project them onto a few attributes. Instead, our statistical analyses are based on randomization tests against the null of the two distributions being identical. These permit us to detect differences between human- and LLM-generated problems without requiring us to specify what, exactly, evaluators might be responding to.

We now turn to our findings. First, individual evaluators perform at chance: mean accuracy is 50.9%, and the full distribution of individual accuracies is statis-

[1] All solutions were written by the same AI solver.

tically indistinguishable from random guessing. This conclusion is robust across all subgroups (graduate/undergraduate, male/female), and no evaluator characteristic—expertise level, self-assessed ability, or time invested—predicts classification performance. Second, aggregated human evaluation can detect that the LLM problems are systematically different from those written by human instructors. Pooling all 2,680 classifications rejects the null hypothesis that the LLM and human distributions are the same ($p = 0.014$). This is consistent with problem-by-evaluator interaction: different evaluators detect signals on different problems, so that pooling amplifies a weak but real signal that is invisible at the individual level. Third, the signal resides in what problems call for in their solutions, not in the problem statements themselves. When evaluators observe solutions and we restrict to medium- and high-confidence judgments, pooled accuracy rises to 53.4% and the randomization test yields $p = 0.0006$—despite the confidence restriction substantially reducing the sample. By contrast, classifications based on problem statements alone show no evidence of distinguishability. Because all solutions were generated by a single AI solver regardless of problem source, this pattern cannot reflect stylistic differences in solution writing; it instead indicates that the problems themselves differ in internal content and structure, which is manifest in the solutions.

Thus, AI-aided problem-writing produces the situation that individuals have no statistical basis on which to distinguish an AI-written set of problems from a human set—indeed, the predominant error is to mistake LLM-generated problems for human-written ones. On the other hand, the distributions are clearly different. The next question is how the AI-generated problems differ from the human ones in distribution. Using nine measurable problem and solution features, we estimate statistical classifiers that predict source labels out-of-sample with 67–70% accuracy, substantially above evaluator performance. The strongest and most consistent signal is the ratio of words in the solution to words in the problem statement, a statistic on which human experts score distinctly higher. This reinforces the interpretation that systematic source differences exist. Moreover, the feature in question is plausibly pedagogically important: human instructors seem to write problems that elicit more reasoning per

unit of setup.

The remainder of the paper proceeds as follows. Section 1.1 discusses related work. Section 2 presents the statistical framework. Section 3 describes corpus construction and experimental implementation. Section 4 reports individual and pooled findings. Section 5 analyzes objective source signals. Section 6 is a concluding discussion of implications and reflections.

1.1 Related Work

The Turing test has evolved considerably since Turing (1950) proposed conversation as the benchmark for machine intelligence. While conversational fluency no longer reliably distinguishes humans from machines (Jones and Bergen, 2025; Jannai et al., 2023), conversation tests only one dimension of intelligence. Porter and Machery (2024) demonstrate that AI-generated poetry is indistinguishable to typical readers from human verse—evaluators achieve just 46.6% accuracy, with a systematic bias toward classifying AI poems as human-written. Notably, their study uses non-expert readers, in contrast to our use of a cohort that includes skilled PhD-level experts in a different domain. Fiedler and Döpke (2025) find that experienced thesis supervisors detect AI-generated passages in academic writing at 57% accuracy; this detection rate was higher than that of the automated AI detectors they used. Our findings are different: individual evaluators in our sample do no better than chance, and worse than machine learning algorithms.

A substantial literature examines LLM-generated educational content. To give just a few examples, Kurdi et al. (2020) provide a comprehensive review of automatic question generation, documenting steady improvements in fluency and relevance. Doughty et al. (2024) find that GPT-4-generated multiple-choice questions in basic academic topics achieve comparable quality on pre-determined metrics to human-crafted items. Isley et al. (2025) conduct a field experiment finding AI-generated exams statistically equivalent to expert-created ones under item response theory. Shah et al. (2024) focus on building a pipeline to devise mathematics questions that are more difficult than those produced by off-the-shelf LLMs, and study the

relationship between solving and writing problems. Our work is distinct in its focus. We do not pre-commit to rubrics or desired dimensions such as difficulty. Instead, we are agnostic as to what matters about a problem and study distinguishability by human experts using randomization-test statistics in a Turing test experiment. Our main findings, highlighting which aspects of the problems human experts need in order to tell apart LLM from human outputs, are also distinctive in the context of this literature.

2 Statistical Framework

Our approach consists of three steps. We first formalize our problem as the test of a null hypothesis that the human and LLM distributions over problems are the same (Section 2.1). We then conduct a "Turing test" by assembling a corpus of human-generated and LLM-generated problems, and asking expert evaluators to classify subsets of these problems as human- or LLM-generated (Section 2.2). Finally, we aggregate these classifications and implement a pooled randomization test of the null hypothesis (Section 2.3).

2.1 The Null Hypothesis

Let $\mathcal{X}$ denote the space of problems. We use $P_H \in \Delta(\mathcal{X})$ to denote the distribution of questions authored by human experts, and $P_{LLM} \in \Delta(\mathcal{X})$ to denote the distribution of questions generated by an LLM. Our goal is to test the null hypothesis

$$H_0 : P_{LLM} = P_H \tag{1}$$

against the alternative $H_1 : P_{LLM} \neq P_H$. Under the null hypothesis, LLM generation is statistically indistinguishable from human generation. Rejecting this null hypothesis at a desired significance level would indicate that the distributions of LLM-generated and human-generated problems are different.

We do not assume any particular structure on $\mathcal{X}$, such as an ordering or a metric defining similarity between problems. As a consequence, standard tests of (1), such

as the Kolmogorov-Smirnov test, are not appropriate. Instead, we conduct a test via a classification experiment, in which evaluators seek to separate human-generated problems from LLM-generated problems

2.2 The Turing Test

We assemble a corpus consisting of n human-generated problems and n LLM-generated problems. For each problem i, we use $y_i \in \{H, LLM\}$ to denote the source of generation of problem X_i, henceforth its *true label*. These problems are randomly ordered, so the true label sequence $y = (y_1, \ldots, y_{2n})$ is equally likely to be any sequence in $\{H, LLM\}^{2n}$ with precisely n 'H' entries and n 'LLM' entries. The problems are then generated independently according to $X_i \sim P_{y_i}$.

We have r expert evaluators indexed $e = 1, \ldots, r$. For each evaluator e, we draw a sequence of m indices $S^{(e)} = \left(i_1^{(e)}, \ldots, i_m^{(e)}\right)$ uniformly at random from $\{1, \ldots, 2n\}$. The problems with these indices, $\left(X_{i_1^{(e)}}, \ldots, X_{i_m^{(e)}}\right)$, are presented to evaluator e, who classifies each as human-generated or LLM-generated. We use $\hat{y}^{(e)} = \left(\hat{y}_1^{(e)}, \ldots, \hat{y}_m^{(e)}\right) \in \{H, LLM\}^m$ to denote evaluator e's chosen label sequence.

2.3 The Randomization Test

We aggregate the evaluators' predictions to conduct a pooled randomization test of the null. The *pooled* accuracy across evaluators is

$$A_0 = \frac{1}{mr} \sum_{e=1}^{r} \sum_{t=1}^{m} \mathbb{1}\left(\hat{y}_t^{(e)} = y_{i_t^{(e)}}\right)$$

i.e., the fraction of total classifications that are correct. Let $\mathcal{Y}$ denote the set of all balanced label vectors, i.e., all $y' \in \{H, LLM\}^{2n}$ with exactly n entries equal to H. For each $y' \in \mathcal{Y}$, define

$$A(y') = \frac{1}{mr} \sum_{e=1}^{r} \sum_{t=1}^{m} \mathbb{1}\left(\hat{y}_t^{(e)} = y'_{i_t^{(e)}}\right)$$

to be the classification accuracy that would obtain if y' were the true label vector. The observed accuracy is the special case $A_0 = A(y)$.

We draw label vectors $y'_1, \ldots, y'_K$ uniformly at random from $\mathcal{Y}$ and compute the p-value

$$p = \frac{1 + \sum_{k=1}^{K} \mathbb{1}(A(y'_k) \geq A_0)}{1 + K}$$

This is the share of random balanced label vectors that yield classification accuracy at least as high as that obtained with the true labels.

We reject the null at significance level α if $p \leq \alpha$.

Theorem 1. *The proposed test is valid, i.e.,* $\Pr(p \leq \alpha \mid H_0) \leq \alpha$.

The proof is provided in Appendix B. Under the null hypothesis $P_{LLM} = P_H$, there is no distinction between human-generated and LLM-generated problems: all problems are drawn from the same distribution. As a result, it is arbitrary which specific problems are labeled human or LLM, and replacing the true labels with any other balanced label vector will not systematically improve or worsen the evaluator's accuracy. If instead the null hypothesis is false, so that $P_{LLM} \neq P_H$, the true labels are systematically related to features of the problems. To the extent that evaluators can detect the true source of generation, their classifications will tend to align better with the true labeling than with a random balanced labeling. As a result, the observed pooled accuracy A_0 will tend to lie in the upper tail of the distribution of accuracies under random label vectors, thus providing evidence against the null.

3 Implementation of the Turing Test

We now describe in detail how we conducted the Turing Test (Section 2.2). Section 3.1 introduces the corpus of problems and Section 3.2 presents the classification experiment.

3.1 Problem Collection

Our corpus of problems was assembled as follows. First, we collected about 150 human-generated problems from advanced undergraduate game theory courses taught over the past decade at five institutions: Berkeley, Columbia, Harvard, Penn, and Yale. We emailed our contacts who taught an advanced undergraduate game theory elective and asked them for materials (problem sets and exams). Among the problems received (about 150), we filtered out a small number of problems not in the standard undergraduate game theory canon (e.g., axiomatic decision-theory problems).

Additionally, we used GPT-5 to generate problems under two prompt regimes. (See Appendix D for the full prompts.) The first prompt, a *short prompt*, is intended to mimic what we expect instructors might use. It specifies a role for the LLM (professor at a top economics department), an objective (generating exam questions for advanced undergraduate game theory), topic coverage (e.g., Nash equilibrium, backward induction, repeated games), and a requirement that the questions be original.

A second *long prompt* incorporates all elements of the short prompt but adds several layers of detail and instruction. In particular, the prompt provides writing guidelines (such as emphasizing clean algebra and a clear logical progression) and informs the LLM that an expert adversary will attempt to distinguish its problems from human-generated ones, with the objective being to evade this classification. We generated over 100 problems under each prompt regime.

For all problems, we used Gemini 3.0 Pro to generate a solution. We checked a random subset of 10 solutions manually and found them to be fully correct. These solutions are used in some of our treatments.

Since our goal is to test whether problems differ in substance rather than linguistic style (such as verbosity or formatting quirks), we apply a *style normalizer* to all problems—human and LLM—rewriting text into uniform voice while preserving mathematical and logical content.[2] We interpret P_H and P_{LLM} in (1) as the

[2]To do this, we built a two-stage pipeline using the Gemini 3.0 Pro-preview model. The first stage was given the original problem statement and asked to distill only the core game-theoretic content into a very terse summary. The second stage was prompted to write a word problem based on that content, subject to the constraint that the original solution should still solve the problem

"normalized" distributions of human- and LLM-generated problems.

At the end of this process, we used a standard LaTeX compiling workflow to render all the problems, with and without solutions. We selected 90 problems from each corpus (human-written, GPT-5 with the short prompt, and GPT-5 with the long prompt)—randomly subject to the constraint that the compilation worked without error.[3] The three sets of 90 are the universe of problems for the remainder of the paper.

3.2 Experiment

We recruited 67 expert evaluators to participate in the Turing test: 35 economics PhD students at Northwestern University and 32 Northwestern undergraduates who had previously completed a course in game theory.

The experimental design varies along two dimensions:

(1) **LLM prompting regime**—whether the LLM problems were generated under the short prompt or the long prompt.

(2) **Information condition**—whether evaluators observed problem statements with or without accompanying solutions. We generated all solutions—including for the human-generated problems—using an LLM, and explained this to the evaluators.

Given the limited pool of expert evaluators, we implemented only three of the four possible combinations of these two dimensions. The treatment conditions are summarized in Table 1.

We treat the long-prompt-with-solutions condition as the primary one. The remaining two conditions each differ from this baseline along a single experimental dimension. In what follows, we report results separately for each condition, as well as pooled results for the two conditions with solutions and for the two long-prompt conditions.

correctly. This resulted in problems that were rewritten in a totally uniform voice, but had to rigidly preserve the same content, enforced by consistency with the solution.

[3]We manually adjusted LLM-produced diagrams so that they rendered correctly.

Table 1: Treatment Design: Each cell indicates whether evaluators in that condition were shown problems generated under the given prompt regime, with or without solutions. A checkmark indicates that the treatment was included in our experiment.

	With Solutions	**Without Solutions**
Long Prompt	✓	✓
Short Prompt	✓	—

Each evaluator was randomly assigned to one treatment condition and presented with $m = 40$ problems to classify, which were drawn uniformly at random from the relevant corpus of problems and presented in random order.[4] The evaluators were incentivized to report correct classifications: on top of a base payment ($10 for undergraduates, $20 for graduates), they received $0.50 per correct classification (maximum bonus $20).[5]

We asked evaluators to report their confidence in each classification (Low, Medium, or High). At the end of the experiment, evaluators predicted their overall accuracy level and provided a verbal explanation of their classification strategy.

4 Main Results

This section presents our empirical findings in four parts. Section 4.1 reports baseline summary statistics for the data. Section 4.2 shows that individual evaluators are not able to meaningfully distinguish LLM-generated problems from human-generated problems.. But aggregating classifications across evaluators allows us to reject the null hypothesis that the LLM-generated distribution and human-generated distribution are the same (Section 4.3). Finally, Section 4.4 provides auxiliary results about what predicts correct classification.

[4]For example, in the long prompt with solutions treatment, the "relevant corpus" consists of the 90 human-generated problems (with solutions) and the 90 LLM-generated problems under the long prompt (with solutions).

[5]Evaluators were not informed of the correctness of their classifications, although they could infer this from the bonus payment following the experiment.

4.1 Summary Statistics

Table 2 presents descriptive statistics regarding the evaluators and their behavior in our experiment.

Panel A describes evaluator demographics. The sample consists of 67 evaluators, split roughly evenly between graduate students (35) and undergraduates (32). Overall, 38.8% of evaluators are female, with a higher female share among undergraduates (46.9%) than among graduate students (31.4%). Among graduate students, 45.7% are in their first or second year of study, whereas undergraduates are predominantly upper-class students, with 93.8% in their junior or senior year.

Panel B reports evaluator-level measures summarizing survey participation. The median survey duration is 42.5 minutes, with somewhat longer completion times among graduate students (45.2 minutes) than undergraduates (40.8 minutes). Self-predicted accuracy is 54.3% on average, with the most confident evaluator predicting 80% and the least confident evaluator predicting 15%.

Panel C reports statistics at the classification level. The median response time per classification is 0.50 minutes (30 seconds), with graduate students spending slightly longer per problem than undergraduates. Across all responses, 44.5% are labeled low confidence, 40.9% medium confidence, and 14.6% high confidence. Undergraduates are relatively more likely to report high confidence, and graduate students are relatively more likely to report low confidence.

4.2 Individual Evaluators

We first examine the classification performance of individual evaluators, and whether there is meaningful heterogeneity in their ability to separate LLM-generated and human-generated problems.

The average accuracy across evaluators is 50.9%, with a standard deviation of 8.2 percentage points, and a median accuracy of 50.75%. The most accurate evaluator classified 70% of problems correctly, and the least accurate evaluator classified 30% of problems correctly. These summary statistics are very close to what we would

Table 2: Summary Statistics

Variable	All	Grad	Undergrad
Panel A: Evaluator demographics ($n = 67$)			
N	67	35	32
Female (%)	38.8	31.4	46.9
Years 1–2 (%)	26.9	45.7	6.2
Years 3+ (%)	73.1	54.3	93.8
Panel B: Evaluator-level measures			
Survey duration, median (minutes)	42.5	45.2	40.8
[min, max]	[12, 2539]	[12, 2447]	[13, 2539]
Self-predicted accuracy (%)	54.3	54.2	54.4
[min, max]	[15, 80]	[30, 70]	[15, 80]
Panel C: Classification-level measures ($n = 2{,}680$)			
Response time, median (minutes)	0.50	0.61	0.40
[min, max]	[0.01, 332]	[0.01, 332]	[0.06, 114]
Confidence: Low (%)	44.5	47.0	41.9
Confidence: Medium (%)	40.9	40.2	41.7
Confidence: High (%)	14.6	12.8	16.4

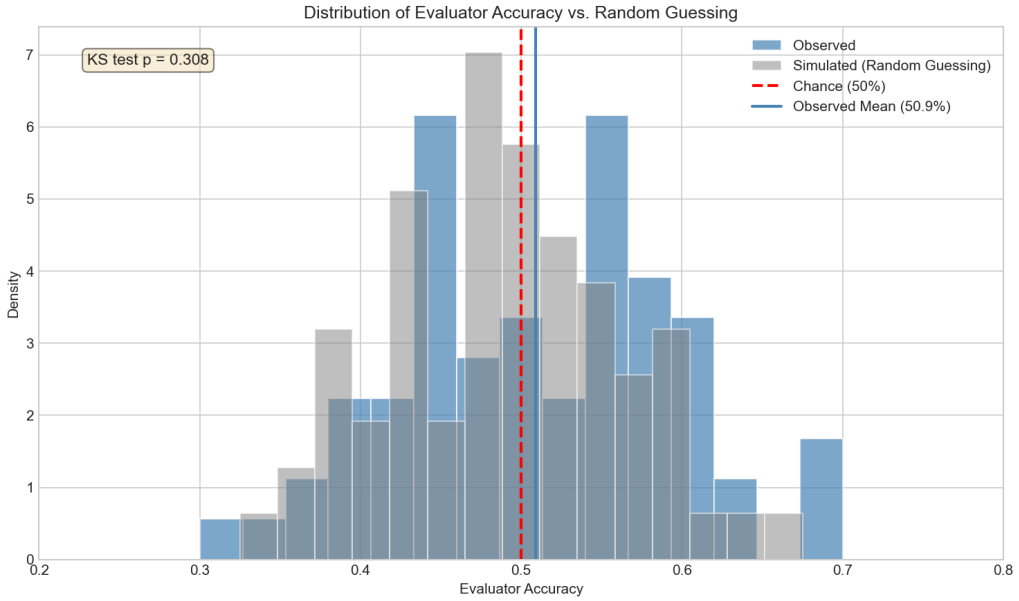
Note: Years 1–2 = G1/G2 (grad) or Freshman/Sophomore (undergrad).

expect in a sample of 67 evaluators each classifying 40 problems at random.[6] Indeed, a two-sample Kolmogorov-Smirnov test cannot reject the null hypothesis that the classifications were generated by evaluators guessing at random ($p = 0.320$). See Figure 1 for a comparison of the empirical distribution of evaluator accuracies to a simulated distribution under random guessing.

This conclusion is robust across evaluator subpopulations and treatment conditions. Panel A of Table 3 reports accuracy by evaluator type, with KS tests comparing each subgroup's accuracy distribution to random guessing. Neither graduate students ($p > 0.99$) nor undergraduates ($p = 0.091$) show accuracy distributions distinguishable from chance at a 5% significance level. Panel B of Table 3 reports analogous results by treatment condition; again, no subgroup differs significantly from random guessing.[7]

[6]The minimum under the random-guessing null is, on average, 32% and the maximum under the random-guessing null is, on average, 68%.

[7]This conclusion does not depend on the use of a KS test. Appendix A.1 conducts randomiza-



Histogram showing evaluator accuracy distribution centered around 50 percent, overlaid with simulated random guessing distribution. The two distributions are nearly identical.

Figure 1: Distribution of individual evaluator accuracy. The observed distribution (blue) is statistically indistinguishable from a simulated distribution under random guessing (gray). A Kolmogorov-Smirnov test comparing these distributions yields $p = 0.320$.

Table 3: Evaluator Accuracy vs. Random Guessing

	N	Mean	SD	Min	Max	KS p-value
Panel A: By Evaluator Type						
Graduate students	35	49.9%	8.0%	30%	70%	>0.99
Undergraduates	32	52.0%	8.4%	35%	67.5%	0.091
Panel B: By Treatment						
Short prompt with solutions	19	50.3%	8.6%	35%	70%	>0.99
Long prompt with solutions	25	51.9%	6.8%	40%	67.5%	0.473
Long prompt without solutions	23	50.3%	9.4%	30%	67.5%	0.416

Taken together, these results indicate that the LLM passes the Turing test at an individual level: there is no evidence that individual evaluators have a systematic ability to distinguish human- from LLM-generated problems. At the same time, this does not imply that the two types of problems are identical in distribution. We now conduct a more powerful test, asking whether aggregation of evaluator judgments can separate the two distributions.

4.3 Pooled Randomization Tests

Using all 2,680 classifications, the randomization test in Section 2.3 rejects the null that the LLM-generated and human-generated distributions are identical at the 5% level ($p = 0.014$). Figure 2 illustrates how. Under the null hypothesis that evaluators cannot distinguish LLM- from human-generated problems, replacing the true labels with a random balanced label vector generates a distribution of aggregate accuracies centered at 48.8%. Although the observed average accuracy of 50.9% is only modestly larger than this benchmark, fewer than 2% of random label vectors produce an accuracy at least this large, placing the observed value in the right tail of the distribution. Thus, while individual judgments are extremely noisy, the LLM and human distributions over problems are not identical, and the Turing test is failed in the aggregate: evaluators are collectively able to exploit a weak but systematic signal that differentiates LLM-generated problems from human-generated ones.

To identify the source of this aggregate signal, we repeat the pooled randomization test separately by treatment condition. Table 4 shows that separation is driven by differences in the solutions rather than in the problem statements themselves. Pooling the two treatments in which evaluators observed solutions (short and long prompts with solutions) yields a much stronger rejection of the null, with a p-value of 0.002. Because all solutions were written by an LLM—regardless of whether the underlying problem was human- or LLM-generated—this separation does not reflect stylistic differences in solution writing, but instead reflects differences in what the problems

tion tests separately for each evaluator, and finds that the distribution of p-values is statistically indistinguishable from a uniform distribution at the 5% significance level.

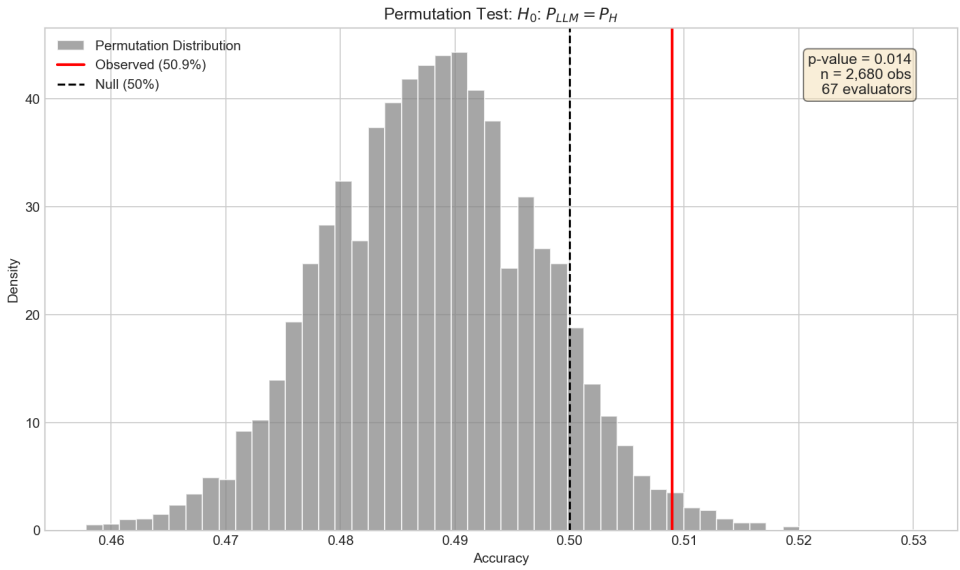


Figure 2: Randomization test distribution (all classifications). The histogram shows pooled accuracy $A(y')$ over 10,000 random balanced label vectors (gray). The mean under randomization is 48.8%. The observed accuracy $A_0 = 50.9\%$ (red line) falls in the right tail, yielding $p = 0.014$.

call for in their solutions. This pattern is consistent with LLMs being highly effective at reproducing the style and structure of human-written problem prompts, while leaving small but detectable differences in the internal content of the problems.

By contrast, pooling the two long-prompt conditions (with and without solutions) yields no rejection of the null, despite a larger number of observations (1,920 versus 1,760).

Table 4: Pooled Randomization Test Results: All Classifications

	With Solutions			Without Solutions			**All**		
	N	Acc	p-value	N	Acc	p-value	N	**Acc**	**p-value**
Long prompt	1,000	51.9%	0.105	920	50.3%	0.404	**1,920**	**51.1%**	**0.140**
Short prompt	760	50.3%	0.488		—			—	
All	**1,760**	**51.2%**	**0.002**		—		**2,680**	**50.9%**	**0.014**

These qualitative conclusions are strengthened when we restrict attention to medium- and high-confidence classifications. This restriction removes roughly 40% of classifications—substantially reducing statistical power. Nevertheless, pooling the two treatments in

which solutions were shown nonetheless yields a higher accuracy of 53.4% and a highly significant p-value of 0.0006. Notably, no comparable improvement appears when solutions are unavailable: pooling medium- and high-confidence classifications for the two long-prompt treatments, with and without solutions, again produces no evidence against the null.

Taken together, these results indicate that differences between LLM-generated and human-generated problems exist but are difficult to detect. Expert classifications are largely noise, but at times: (1) evaluators receive a signal about a problem-solution pair, (2) this signal is informative, and (3) the expert recognizes that they have received an informative signal.

Table 5: Pooled Randomization Test Results: Medium and High Confidence Only, by Treatment

	With Solutions			Without Solutions			**All**		
	N	Acc	p-value	N	Acc	p-value	N	**Acc**	**p-value**
Long prompt	495	53.1%	0.072	558	50.0%	0.515	**1,053**	**51.5%**	**0.165**
Short prompt	425	53.7%	0.086		—			—	
All	**920**	**53.4%**	**0.0006**		—		**1,478**	**52.1%**	**0.057**

Note: Restricted to medium- and high-confidence assessments.

4.4 What Predicts Classification Accuracy?

We now examine what factors predict classification accuracy at both the problem level and the evaluator level.

Classification-level predictors. Table 6 reports OLS estimates from regressing classification accuracy on classification-level covariates.[8] The most predictive covariate is the true source of generation: when a problem is human-written, the probability of correct classification increases by 12 percentage points ($p < 0.001$). This reflects a systematic asymmetry in evaluator performance: evaluators classify human problems

[8]That is, the dependent variable is an indicator for whether a classification is correct, and the independent variables are the classification-level covariates. Reported standard errors are clustered at the evaluator level.

correctly 56.9% of the time but LLM problems correctly only 45.0% of the time. In other words, 55.0% of LLM-generated problems are misclassified as human-written, while 43.1% of human problems are misclassified as LLM-generated. The main evaluation failure is thus confusing LLM problems for human ones.

Table 6: Classification-Level Predictors of Correct Classification ($N = 2,680$)

Variable	Coefficient	p-value
True label is Human	+0.12	<0.001
Medium confidence (vs. Low)	+0.04	0.10
High confidence (vs. Low)	−0.01	0.88
Long prompt treatment	+0.01	0.55
No solutions treatment	≈0	0.95
Display order	≈0	0.91
$R^2 = 0.016$*; clustered SEs at evaluator level*		

Other classification-level covariates have smaller effects. Medium-confidence classifications are 4 percentage points more accurate than low-confidence classifications ($p = 0.10$), while high-confidence classifications show no improvement.[9] Treatment assignment (long prompt vs. short prompt, solutions vs. no solutions) and display order have no significant effects. The low R^2 (0.016) indicates that classification accuracy is largely unpredictable from observable covariates.

Appendix A.2 reports the distribution of accuracy across problems and shows that it is statistically indistinguishable from what would arise under random guessing, suggesting that problems are not systematically easier or harder to classify.

Evaluator-level predictors. Table 7 reports evaluator-level accuracy regressed on evaluator characteristics ($N = 67$ evaluators). No evaluator characteristic significantly predicts performance. Graduate students perform slightly worse than undergraduates (−2 pp, $p = 0.36$), and female evaluators perform slightly worse than male evaluators (−2 pp, $p = 0.37$), but neither effect approaches statistical significance. Self-predicted accuracy shows no relationship to actual performance ($p = 0.44$), and

[9]The latter is likely due to the small fraction of high-confidence classifications.

mean response time is negatively associated with accuracy but not significantly so ($p = 0.15$). The low R^2 (0.08) confirms that evaluator-level variation in accuracy is largely unpredictable from the observed covariates.

Table 7: Evaluator-Level Predictors of Accuracy ($N = 67$)

Variable	Coefficient	p-value
Graduate student	-0.02	0.36
Female	-0.02	0.37
Self-predicted accuracy	$+0.06$	0.44
Mean response time (minutes)	-0.02	0.15
Long prompt treatment	$+0.01$	0.55
No solutions treatment	≈ 0	0.95
$R^2 = 0.08$		

The null effects at both levels—classification and evaluator—suggest the signal is not concentrated in identifiable easy problems or skilled evaluators. The pattern is consistent with problem-by-evaluator interaction: different evaluators detect signals on different problems.

5 What Differentiates Human- and LLM-Generated Problems?

The preceding section showed that LLM distribution and human distribution over problems are not the same. We now ask whether this difference can be linked to measurable features of the problems, and if so what those features are.

5.1 Problem features

We define nine objectively measurable features. Five features characterize the problem statement: word count, equation count, number of sub-parts, theoretical references, and words-to-equations ratio. Three features characterize the solution: solution word count, solution equation count, and clean number ratio (proportion of pedagog-

ically simple numerical values such as integers in $[-10, 10]$ or simple fractions). Our final feature, which pertains to both the problem statement and the solution, is the ratio of their word counts. See Appendix C for more detail about the definitions of these features and how they were computed.

Table 8: Feature Means by Problem Source

Feature	**Human**	**Long Prompt**	**Short Prompt**
Problem statement features:			
Word count	149	188	165
Equation count	43.8	47.1	31.5
Number of sub-parts	2.7	2.2	2.7
Theoretical references	0.5	0.2	0.4
Solution features:			
Word count	293	299	312
Equation count	33.8	34.9	24.3
Clean number ratio	0.95	1.00	0.98
Problem-solution relationship:			
Solution-to-problem ratio	2.27	1.66	1.96

Table 8 presents summary statistics for these features. Human problems tend to be shorter, have shorter solutions, and have higher solution-to-problem word count ratios.

5.2 Which Features are Predictive?

Table 9 reports the estimated coefficients from a logistic regression that predicts the source of generation given these features.[10] The first column pools all problems, regardless of prompt length. The second and third columns estimate the same regression separately for long-prompt and short-prompt problems.

The *solution-to-problem word ratio* emerges as a consistent predictor, with large and statistically significant coefficients ranging from 1.334 to 1.803. The estimated

[10]Logistic regression models the log-odds of human authorship as a linear function of the features: $\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$, where p is the probability that a problem is human-written and $x_1, \ldots, x_k$ are the feature values. Positive coefficients indicate that higher values of a feature increase the probability of human authorship; negative coefficients indicate the opposite. The model jointly estimates all coefficients while controlling for the other features.

Table 9: Logistic Regression Coefficients by Comparison

Feature	**Both Prompts**	**Long Prompt**	**Short Prompt**
Problem-solution relationship:			
Solution-to-problem word ratio	1.334*	1.803†	1.785*
	(0.569)	(0.931)	(0.706)
Solution features:			
Solution word count	-0.009*	-0.008	-0.015**
	(0.004)	(0.006)	(0.005)
Clean number ratio	1.892	-5.029*	5.232**
	(1.519)	(2.133)	(1.875)
Solution equation count	0.021*	-0.001	0.045**
	(0.010)	(0.012)	(0.014)
Problem statement features:			
Number of sub-parts	0.170†	0.372**	0.017
	(0.091)	(0.139)	(0.108)
Theoretical references	0.131	0.554†	-0.088
	(0.150)	(0.316)	(0.170)
Words-to-equations ratio	0.003*	-0.000	0.005**
	(0.001)	(0.002)	(0.002)
Problem equation count	0.027	-0.001	0.081*
	(0.024)	(0.029)	(0.034)
Problem word count	0.004	0.001	0.018†
	(0.008)	(0.011)	(0.010)

Significance: **p<0.01, *p<0.05, †p<0.10

Standard errors in parentheses.

Pooled: $R^2 = 0.129$, $N = 270$; Long: $R^2 = 0.198$, $N = 180$; Short: $R^2 = 0.220$, $N = 180$

coefficients imply that a one-unit increase in this ratio—roughly a 40% increase relative to the human mean of 2.27—multiplies the odds that a problem is human-written by approximately four- to six-fold, holding other features fixed. Because all solutions were generated by an AI model with uniform prompting (Gemini 3.0 Pro-preview), this difference cannot be driven by stylistic variation in solution writing and must instead reflect properties of the problems themselves. The result suggests that, conditional on problem length, human-written problems tend to require more extensive solutions, consistent with greater conceptual density per word.

5.3 Prediction accuracy

To assess how well these objective features distinguish problem sources, we conduct out-of-sample classification tests. For each comparison (Human vs. Long Prompt, and Human vs. Short Prompt), we randomly partition the data into 5 folds, train the logistic regression model on 4 folds and test on the final fold. Table 10 presents the average of these out-of-sample errors.

Ex ante, it is ambiguous whether these features should be more or less informative about a problem's source than expert human judgment. On the one hand, we might expect experts to have intuitions regarding the quality of the problems—such as how interesting or difficult they are—that are not captured by the nine features we define. In that case, human judgment would draw on information that lies outside of our feature set. On the other hand, the features could provide a stronger signal if they capture systematic differences between human- and LLM-generated problems that experts either overlook or only perceive with substantial noise. In that case, our logistic classifier could outperform the human experts from Section 3.2.

Table 10: Classification Accuracy (Out-of-Sample)

Comparison	**N**	**Accuracy**
Human vs. Long Prompt	180	70.0%
Human vs. Short Prompt	180	66.7%

Baseline: 50%. Cross-validated accuracy from 5-fold CV. Model includes 9 objective features.

In fact, we find the latter: The objective feature model achieves substantially higher accuracy (67–70%) than human evaluators in Section 3.2, who performed at essentially chance levels (51–52%). What explains this gap? There are several possibilities. First, humans may assess the same underlying features, but with error. Alternatively, humans may focus on entirely different features than those captured by the model—features that are not actually diagnostic of authorship.

The confusion matrices in Table 11 suggest that the answer is not completely the former. Comparing Panels (a) and (b) reveals that human evaluators primarily err in misclassifying LLM problems as human-written, while the classifier trained on these

Table 11: Confusion Matrices: Human Evaluators vs. Statistical Classifier (In-Sample)

Panel A: Human Evaluators					
Human vs. Long Prompt			**Human vs. Short Prompt**		
True Label	Human	LLM	True Label	Human	LLM
Human	56.6%	43.4%	Human	57.6%	42.4%
LLM	54.3%	45.7%	LLM	56.6%	43.4%
Panel B: Statistical Classifier					
Human vs. Long Prompt			**Human vs. Short Prompt**		
True Label	Human	LLM	True Label	Human	LLM
Human	60.0%	40.0%	Human	70.0%	30.0%
LLM	23.3%	76.7%	LLM	23.3%	76.7%

nine features instead errs in misclassifying human problems as LLM-generated. If humans were attempting to measure the same signal as the statistical model but with noise, we would expect similar error patterns with lower overall accuracy.

6 Concluding Discussion

We proposed a Turing test for problem generation in undergraduate game theory. Individual evaluators perform no better than chance (mean accuracy 50.9%, $p = 0.73$ vs. random guessing). However, pooling confident classifications on problems with solutions rejects the null that human and LLM distributions are identical ($p = 0.012$). Three conditions jointly play an important role in detection: access to solutions, confidence filtering, and aggregation.

The methodology—a Turing test for generative tasks validated by randomization testing—provides a template for measuring human-AI distinguishability as LLM capabilities evolve.

This section discusses some additional interpretations and implications of our findings.

6.1 Distinguishability and Indistinguishability

On the one hand, the absence of reliable individual-level detection suggests that LLM-generated problems closely match human-generated ones in surface form and apparent difficulty. On the other hand, the emergence of a detectable signal under pooling indicates that these problems differ subtly but systematically.

These facts bound the possibilities of what AI might be doing in this setting, and rule out some coarse explanations. For example, neither crude failures nor naive replication of training data alone can explain this pattern. Obvious defects (such as producing problems with trivial or confusing solutions) would generate detectable signals even at the individual level, while pure regurgitation would eliminate aggregate separation. Instead, our findings suggest that LLM-generated problems occupy a region of the problem space that is close to, but not identical with, that of human-generated problems.

Crucially, our test measures distinguishability rather than quality. The presence of systematic differences does not imply that LLM-generated problems are worse, only that they are different in ways that become detectable when solutions are examined. It would be natural to study implications for pedagogical outcome measures, which would be likely to appear over time rather than being apparent in a single assessment.

Lastly, one might ask whether the contrast between individual indistinguishability and aggregate distinguishability is merely a question of power—that there are many more problem-evaluation pairs in the total sample than for any individual evaluator. Two facts suggest that this cannot be the whole story. First, when we drop low-confidence classifications—reducing the sample by roughly 40%—the pooled test rejects more strongly when solutions are shown. Second, a comparably sized sample without solutions yields no rejection. Thus, the gap reflects where information enters, not simply sample size: a weak signal is available only for problem–solution pairs. However, it does seem consistent with the evidence that no single evaluator gets enough signals to perform better than chance; some element of the "wisdom of crowds" is helpful for the aggregate distinguishability that we report.

6.2 Where the Differences Arise

A key feature of the results is that aggregate separation emerges only when evaluators observe solutions. When classifications are based on problem statements alone, we find no evidence of distinguishability, even though the number of observations in that sample is larger. This suggests that LLMs are particularly effective at reproducing the surface presentation of undergraduate game theory problems, while differences arise in the structure of reasoning latent in the problems. These become manifest when the problems are "unpacked" with a common solver.

One concrete manifestation is that human-written problems elicit more words of solution per word of problem statement. In our feature-based analysis, this solution-to-problem word ratio is the strongest and most consistent predictor of human authorship (Table 9), reinforcing the view that the main differences reside in what the problems call for when they are solved.

We caution, however, against reading too much into the classifier's success. Being trained on part of this data and evaluated out of sample, this classifier learned the specific tells that come from these prompts. It would presumably be easy to prompt and filter AI output in a way that would avoid these particular tells. Our hypothesis is that while the human signal about problem quality is weaker, it would be more resilient to this game of cat and mouse.

6.3 Implications for Teaching and Classroom Experience

These findings have implications for the use of LLMs in instructional settings. The most obvious one is a "failure mode" that the results show is possible. A professor, pressed for time, might generate a problem set or an exam using an LLM and find the results apparently of high quality. Similarly, no individual student or colleague would be able to tell that individual problems, or even the whole document, is LLM-generated (since our evaluators did no better than chance on 40 problems).

Yet students as a group would have a different experience from cohorts whose professors did not use AI assistance in composing problems. The professor-composed

problems have discernibly different pedagogical properties, as picked up by the objective features, as well as various other features that some humans no doubt detected.

Thus, even when each problem appears entirely normal and acceptable in isolation, the distribution of problems a class encounters would drift in aggregate, potentially shaping which skills students practice, how they experience the process of working through the problems, or what conversations the problems create. Identifying whether such distributional differences have meaningful downstream consequences for productive outcomes (in this case, educational ones) is an important question for future work.

6.4 What We Learned about Prompting

We designed the long prompt as our best attempt at evading detection: it adds detailed writing constraints and, most distinctively, it gives the model an explicit adversarial evaluation context and instructs it to evade detection (Appendix D). Based on informal experimentation with earlier generations of models, this kind of adversarial prompt seemed to produce problems that appeared more expert-like to us than those coming from the short prompt.[11]

In our experiment, however, the more elaborate prompt design backfired. The long prompt did not move GPT-5 output closer to the human distribution as measured by distinguishability. If anything, it moved it in the opposite direction (Table 4, Table 10). Long-prompt problems ended up having longer statements and a lower solution-to-problem word ratio (1.66 versus 1.96 for short-prompt and 2.27 for human; Table 8).

We have several reflections concerning this fact. First, more recent models appear to be better at understanding the intent behind a natural prompt, and may have performed worse by over-interpreting the details of the adversarial prompt. Nevertheless, since the short prompt also did not produce perfect results, it is tempting to prompt better to repair this; our experience suggests that prompting directly for

[11] Indeed, these problems evaded classifiers based on LLM prompts, an example of which is given within the long prompt in Appendix D.

indistinguishability may not work.

6.5 The Scope of our Study and Implications for Other Domains

Our application focuses on undergraduate game theory, a domain that is well-represented in LLM training data, with a consensus on everything from the canonical syllabus to many aspects of notation. These features make it a favorable setting for LLM performance. In domains with less standardized curricula or sparser representation in training data, the divergence between human- and LLM-generated problem distributions may be larger. For example, in a field where advanced undergraduate courses are relatively new, human instructors may have better intuition for how to adapt research results to the instructional setting.

Understanding how distinguishability varies across domains is a natural direction for future research.

References

Jacob Doughty, Zipiao Wan, Anishka Bompelli, Jubahed Qayum, Taozhi Wang, Juran Zhang, Yujia Zheng, Aidan Doyle, Pragnya Sridhar, Arav Agarwal, Christopher Bogart, Eric Keylor, Can Kultur, Jaromir Savelka, and Majd Sakr. 2024. A Comparative Study of AI-Generated (GPT-4) and Human-crafted MCQs in Programming Education. In *Proceedings of the 26th Australasian Computing Education Conference* (Sydney, NSW, Australia) *(ACE '24)*. Association for Computing Machinery, New York, NY, USA, 114–123. https://doi.org/10.1145/3636243.3636256

Alexandra Fiedler and Jörg Döpke. 2025. Do Humans Identify AI-Generated Text Better Than Machines? Evidence Based on Excerpts from German Theses. *International Review of Economics Education* 49 (2025), 100321. https://doi.org/10.1016/j.iree.2025.100321

Calvin Isley, Joshua Gilbert, Evangelos Kassos, Michaela Kocher, Allen Nie, Emma Brunskill, Ben Domingue, Jake Hofman, Joscha Legewie, Teddy Svoronos, Charlotte Tuminelli, and Sharad Goel. 2025. Assessing the Quality of AI-Generated Exams: A Large-Scale Field Study. *arXiv preprint arXiv:2508.08314* (2025).

Daniel Jannai, Amos Meron, Barak Lenz, Yoav Levine, and Yoav Shoham. 2023. Human or Not? A Gamified Approach to the Turing Test. *arXiv preprint arXiv:2305.20010* (2023).

Cameron R. Jones and Benjamin K. Bergen. 2025. Large Language Models Pass the Turing Test. *arXiv preprint arXiv:2503.23674* (2025).

Ghader Kurdi, Jared Leo, Bijan Parsia, Uli Sattler, and Salam Al-Emari. 2020. A Systematic Review of Automatic Question Generation for Educational Purposes. *International Journal of Artificial Intelligence in Education* 30, 1 (2020), 121–204. https://doi.org/10.1007/s40593-019-00186-y

Brian Porter and Edouard Machery. 2024. AI-Generated Poetry is Indistinguishable from Human-Written Poetry and Is Rated More Favorably. *Scientific Reports* 14 (2024), 26133. https://doi.org/10.1038/s41598-024-76900-1

Vedant Shah, Dingli Yu, Kaifeng Lyu, Simon Park, Jiatong Yu, Yinghui He, Nan Rosemary Ke, Michael Mozer, Yoshua Bengio, Sanjeev Arora, and Anirudh Goyal. 2024. AI-Assisted Generation of Difficult Math Questions. *arXiv preprint arXiv:2407.21009* (2024).

Alan M. Turing. 1950. Computing Machinery and Intelligence. *Mind* 59, 236 (1950), 433–460.

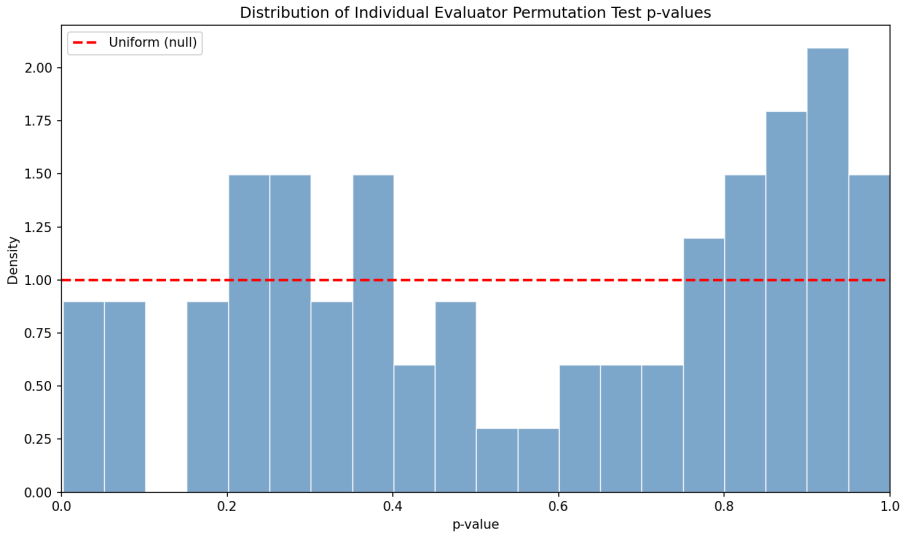
A Further analyses

A.1 Individual Evaluator Randomization Tests

For each evaluator, we run a randomization test by replacing the true labels with a random balanced label vector among only the 40 problems that evaluator saw. The p-value is the fraction of random label vectors yielding accuracy at least as high as

observed.

Figure 3 shows the distribution of p-values across evaluators. Under the null hypothesis that no evaluator can distinguish human from LLM problems, p-values should follow a uniform distribution. The observed distribution is consistent with this null: 3 of 67 evaluators (4.5%) have $p < 0.05$, and 6 of 67 (9.0%) have $p < 0.10$—close to the 5% and 10% expected by chance. A Kolmogorov-Smirnov test cannot reject uniformity ($p = 0.060$).



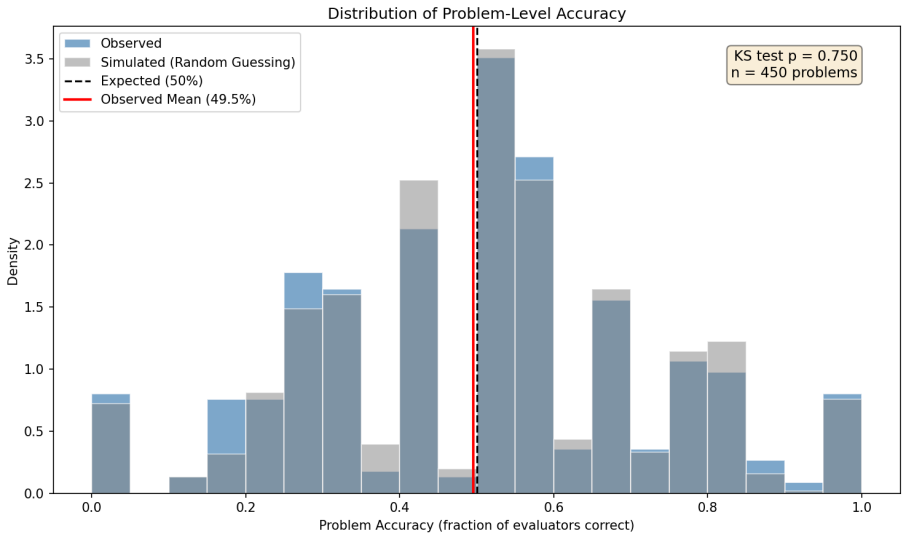
Histogram showing the distribution of p-values from individual evaluator randomization tests. The distribution is roughly uniform, consistent with the null hypothesis.

Figure 3: Distribution of individual evaluator randomization test p-values (all classifications). Under the null hypothesis, p-values should be uniform (red dashed line). The observed distribution is consistent with the null.

A.2 Problem-level heterogeneity.

Are some problems systematically easier or harder to classify? Each of the 450 problems in our corpus was evaluated by 1–14 evaluators (mean 6.0). For each problem, we compute the fraction of evaluators who classified it correctly. Figure 4 compares this distribution to a simulated null under random guessing. The distributions are statistically indistinguishable (KS test $p = 0.750$). Mean problem-level accuracy is

49.5%, with some problems classified correctly by all evaluators and others by none, but this variation is consistent with chance.



Histogram showing problem accuracy distribution centered around 50 percent, overlaid with simulated random guessing distribution. The two distributions are nearly identical.

Figure 4: Distribution of problem-level accuracy (fraction of evaluators correct). The observed distribution (blue) is statistically indistinguishable from random guessing (gray). KS test $p = 0.750$.

B Proof of Theorem 2.1

Appendix B.1 establishes the necessary preliminary results. Appendix B.2 then analyzes an idealized version of the test to highlight the main ideas in a simplified context. Finally, Appendix B.3 provides the full proof of Theorem 1.

B.1 Preliminary Results

We first describe the data-generating process.

(1) **True labels:** Let $Y = (Y_1, \ldots, Y_{2n}) \in \{H, LLM\}^{2n}$ be drawn uniformly at random from the set of all label sequences containing exactly n entries equal to H and n entries equal to LLM.

(2) **Problem generation:** Conditional on Y, the problems $X = (X_1, \ldots, X_{2n}) \in \mathcal{X}^{2n}$ are generated independently, with $X_i \sim P_{Y_i}$ for all $i = 1, \ldots, 2n$.

(3) **Assignment to evaluators:** Let $\mathcal{S}$ denote the collection of all sequences from $\{1, \ldots, 2n\}$ of length m. For each evaluator $e = 1, \ldots, r$, draw an index sequence $S^{(e)}$ independently and uniformly at random from $\mathcal{S}$, with $\boldsymbol{S} = (S^{(1)}, \ldots, S^{(r)})$ denoting the resulting vector of index sequences.

(4) **Evaluator classifications:** Each evaluator e is shown the problems

$$\left(X_{S_1^{(e)}}, \ldots X_{S_m^{(e)}}\right)$$

and maps this into a label vector according to a (possibly randomized) strategy

$$\sigma^{(e)} : \mathcal{X}^m \to \Delta(\{H, LLM\}^m).$$

Let

$$\widehat{Y}^{(e)} \sim \sigma^{(e)}\left(X_{S_1^{(e)}}, \ldots X_{S_m^{(e)}}\right)$$

denote the random label sequence produced by evaluator e, and let $\widehat{\boldsymbol{Y}} = (\widehat{Y}^{(1)}, \ldots, \widehat{Y}^{(r)})$ be the vector of all evaluator label sequences.

The resulting random vector is denoted $Z := (Y, X, \boldsymbol{S}, \widehat{\boldsymbol{Y}})$.

Definition B.1 (Label replacement). Let $\mathcal{Y}$ denote the set of all balanced label vectors, i.e., all $y' \in \{H, LLM\}^{2n}$ with exactly n entries equal to H. For any $y' \in \mathcal{Y}$, define

$$Z^{y'} := (y', X, \boldsymbol{S}, \widehat{\boldsymbol{Y}}),$$

i.e., Z with the true label vector Y replaced by y', while leaving the realized problems, assignments, and evaluator classifications unchanged.

Lemma B.1 (Conditional uniformity of labels)**.** *Under the null hypothesis, Y is independent of $(X, \boldsymbol{S}, \widehat{\boldsymbol{Y}})$ and uniform on $\mathcal{Y}$. Equivalently,*

$$\Pr\big(Y = y \,\big|\, X, \boldsymbol{S}, \widehat{\boldsymbol{Y}}\big) = \frac{1}{|\mathcal{Y}|} \qquad \textit{for all } y \in \mathcal{Y}.$$

Proof. By construction, the joint distribution of $(Y, X, \mathbf{S}, \widehat{\mathbf{Y}})$ factorizes as follows.

The label sequence Y is drawn uniformly from $\mathcal{Y}$. Conditional on Y, the problems are independent with $X_i \sim P_{Y_i}$. The index sequences $\mathbf{S}$ are drawn independently of (Y, X). Finally, evaluator classifications $\widehat{\mathbf{Y}}$ depend only on $(X, \mathbf{S})$.

Under the null, $P_H = P_{LLM}$, so the conditional distribution of X given Y does not depend on Y: for any $y \in \mathcal{Y}$,

$$\Pr(X \in B \mid Y = y) = \prod_{i=1}^{2n} P_{y_i}(B_i) = \prod_{i=1}^{2n} P_H(B_i)$$

which is the same for all $y \in \mathcal{Y}$. Since $\mathbf{S}$ is independent of (Y, X) and $\widehat{\mathbf{Y}}$ depends only on $(X, \mathbf{S})$, it follows that Y is independent of $(X, \mathbf{S}, \widehat{\mathbf{Y}})$ under the null. □

The remainder of the proof is standard, following **?**, but we provide it for completeness.

B.2 Warm-up Result: The Exhaustive Test

As a warm-up, consider the idealized test that enumerates all balanced label vectors. Define

$$p(z) = \frac{\sum_{y' \in \mathcal{Y}} \mathbb{1}\{A(y') \geq A(y)\}}{|\mathcal{Y}|}$$

where $A(y')$ is the pooled accuracy under label vector y' as defined in the main text. Define the test function

$$\phi(z) = \mathbb{1}(p(z) \leq \alpha)$$

where α is the desired significance level.

Lemma B.2. *Under the null, $E[\phi(Z)] \leq \alpha$.*

Proof. Fix any realization of $(x, \mathbf{s}, \widehat{\mathbf{y}})$. The values $\{A(y') : y' \in \mathcal{Y}\}$ are then determined. Order the label vectors in $\mathcal{Y}$ as $y'_1, \ldots, y'_{|\mathcal{Y}|}$ where

$$A(y'_1) \geq \cdots \geq A(y'_{|\mathcal{Y}|})$$

in decreasing order of accuracy. For the label vector y'_k at position k in this ordering, there are at least k label vectors y' with $A(y') \geq A(y'_k)$. So $\phi(z)$ can equal 1 (i.e.,

$p(z) \leq \alpha$) for at most $\lfloor \alpha |\mathcal{Y}| \rfloor$ choices of the true label vector y. Thus

$$\frac{1}{|\mathcal{Y}|} \sum_{y' \in \mathcal{Y}} \phi\left(Z^{y'}\right) \leq \alpha.$$

By Lemma B.1, conditional on $(X, \mathbf{S}, \widehat{\mathbf{Y}})$, the true label vector Y is uniform on $\mathcal{Y}$. Therefore

$$\mathbb{E}\big[\phi(Z) \,\big|\, X, \mathbf{S}, \widehat{\mathbf{Y}}\big] = \frac{1}{|\mathcal{Y}|} \sum_{y' \in \mathcal{Y}} \phi\left(Z^{y'}\right) \leq \alpha.$$

Taking expectations over $(X, \mathbf{S}, \widehat{\mathbf{Y}})$ yields $\mathbb{E}[\phi(Z)] \leq \alpha$, as desired. □

B.3 Proof of Theorem 1

Now consider $y'_1, \ldots, y'_K$ sampled uniformly at random from $\mathcal{Y}$ with replacement, independently of Z. Define

$$p = \frac{1 + \sum_{k=1}^{K} \mathbb{1}\{A(y'_k) \geq A(y)\}}{1 + K}$$

with corresponding test function

$$\phi := \mathbb{1}\{p \leq \alpha\}.$$

We will show that under the null, $\mathbb{E}[\phi] \leq \alpha$, proving Theorem 1.

Set $y'_0 := Y$ (the true label vector). By Lemma B.1, under the null Y is uniform on $\mathcal{Y}$ and independent of $(X, \boldsymbol{S}, \widehat{\boldsymbol{Y}})$. Since $y'_1, \ldots, y'_K$ are also drawn independently and uniformly from $\mathcal{Y}$, the vector $(y'_0, y'_1, \ldots, y'_K)$ is i.i.d. uniform on $\mathcal{Y}$ and independent of $(X, \boldsymbol{S}, \widehat{\boldsymbol{Y}})$.

Define $T_j := A(y'_j)$ for each $j = 0, 1, \ldots, K$. Conditional on $(X, \boldsymbol{S}, \widehat{\boldsymbol{Y}})$, the values $T_0, T_1, \ldots, T_K$ are determined by $(y'_0, \ldots, y'_K)$, which are i.i.d. Therefore $(T_0, T_1, \ldots, T_K)$ is exchangeable.

The p-value can be written as

$$p = \frac{\#\{k \in \{0, \ldots, K\} : T_k \geq T_0\}}{K + 1}.$$

For each index j, define $p_j := \#\{k \in \{0, \ldots, K\} : T_k \geq T_j\}/(K+1)$ and $\phi_j := \mathbb{1}(p_j \leq$

α). Ordering the indices so that $T_0 \geq T_1 \geq \cdots \geq T_K$, at most $\lfloor \alpha(K+1) \rfloor$ indices j can satisfy $p_j \leq \alpha$. Thus

$$\sum_{j=0}^{K} \phi_j \leq \lfloor \alpha(K+1) \rfloor \leq \alpha(K+1). \tag{2}$$

This holds pointwise for every realization, so it holds in expectation:

$$\sum_{j=0}^{K} \mathbb{E}[\phi_j] \leq \alpha(K+1). \tag{3}$$

By exchangeability of $(T_0, \ldots, T_K)$, the indicators $\phi_0, \ldots, \phi_K$ are identically distributed, so $\mathbb{E}[\phi_j] = \mathbb{E}[\phi_0] = \mathbb{E}[\phi]$ for all j. Substituting into (3):

$$(K+1)\,\mathbb{E}[\phi] \leq \alpha(K+1).$$

Dividing by $K+1$ yields $\mathbb{E}[\phi] \leq \alpha$, as claimed.

C Feature Definitions

This appendix provides detailed definitions of the nine objective features used in Section 5 to characterize and distinguish human-written from LLM-generated problems.

C.1 Problem Statement Features

These five features are extracted directly from the problem text as presented to students.

Word count (problem statement). The total number of words in the problem statement, excluding LaTeX markup and mathematical notation. Text is split into words using whitespace as the delimiter. This measures the overall length and verbosity of the problem statement: higher word counts indicate more verbose problem framing, while lower counts indicate more concise presentations.

Equation count (problem statement). The number of mathematical expressions enclosed in equation delimiters (e.g., `$...$`, `$$...$$`, or `equation` environments). This counts both inline and displayed equations. Simple variable references (e.g., x or π) are included if delimited. This measures the mathematical density of the problem statement.

Number of sub-parts. The count of enumerated sub-questions within a problem, typically labeled (a), (b), (c), etc. A problem with no sub-parts has value 1. This measures structural complexity and whether problems assess multiple related concepts or require multi-step reasoning.

Theoretical references. A binary indicator (or count) of explicit references to named theorems, lemmas, propositions, or formal results (e.g., "by the Revelation Principle," "Nash's theorem guarantees"). This measures whether problems invoke formal game-theoretic results or rely purely on definitions and direct reasoning.

Words-to-equations ratio. The ratio of word count to equation count in the problem statement. Higher values indicate more verbal explanation relative to mathematical content. This captures the balance between conceptual framing and formal specification.

C.2 Solution Features

These four features are extracted from solutions to the problems. Critically, *all solutions in our corpus were generated by Gemini 3.0 Pro-preview*, including solutions to human-written problems. Thus, solution features capture properties of what a problem *requires* for explanation, rather than stylistic differences in how humans versus LLMs write solutions.

Solution word count. The total number of words in the solution, measured identically to problem statement word count. This measures the absolute length of explanation required.

Solution equation count. The number of mathematical expressions in the solution, measured identically to problem statement equation count. This measures the mathematical intensity of the solution.

Solution-to-problem word ratio. The ratio of solution word count to problem statement word count. Values greater than 1 indicate solutions that are longer than the problem statement. This measures how much explanatory elaboration the problem requires relative to its framing.

Clean number ratio. The proportion of numerical values in the problem statement that are "simple" or "clean." A number is classified as clean if it satisfies any of the following criteria: (1) an integer in the range $[-10, 10]$; (2) one of the fractions 1/2, 1/3, 1/4, 1/5, 2/3, or 3/4; (3) the decimal equivalent of such a fraction (0.25, 0.333, 0.5, 0.666, 0.75); or (4) zero. The clean number ratio is computed as the number of clean numerical values divided by the total number of numerical values appearing in the problem statement. Higher values indicate more pedagogically convenient parameterizations with simple arithmetic.

D Prompts

This appendix documents the prompts used to generate LLM problems. Appendix D.1 reproduces the short prompt in full. Appendix D.2 describes the additional elements of the long prompt.

D.1 Short Prompt

```
ROLE: You are a professor of economics at a top-10 US economics
department

GOAL: Write exam questions for an advanced undergraduate course
in game theory.
```

```
TOPICS: The exam covers topics in a standard curriculum including
the following topics:
- Strategic form and extensive form definitions
- Writing extensive forms from verbal descriptions
- Deriving strategies in strategic form from an extensive form
- Perfect vs. imperfect information
- Best replies and best-response functions
- Pure and mixed-strategy Nash equilibrium: existence, computation,
  properties
- Iterated elimination of strictly or weakly dominated strategies;
  rationalizability
- Simple continuous-action games and Nash equilibria
- Signaling games and basic cheap talk
- Backward induction and subgame-perfect equilibrium
- Infinitely repeated games and subgame perfection
- Games with incomplete information
- Bayesian Nash equilibrium (simultaneous-move games)
- Perfect Bayesian equilibrium (multistage, incomplete info)
- Cournot and Bertrand competition
- Simple auction formats (first-price, second-price, war of
  attrition)

Questions MUST be original. They must NOT be minor variants of
questions on the Internet or in standard textbooks.
```

D.2 Long Prompt

The long prompt (277 lines) includes all elements of the short prompt and adds seven structured sections, summarized below with key verbatim excerpts.

Objective and Role. The model is cast as "*an economics professor teaching game theory for advanced undergraduate economics students in a top 10 program.*" It must produce exactly six questions per call, each 100–150 words. A meta-cognitive instruction is repeated at both the beginning and end of the prompt: "*THINK HARD ABOUT HOW TO DO THIS TASK WELL. Solve your questions and iterate on them until you are happy with the outcome.*"

Topic Coverage. Sixteen approved topics are listed, spanning core game theory: strategic and extensive forms, Nash equilibrium (pure, mixed, and Bayesian), iterated dominance, backward induction, subgame-perfect and perfect Bayesian equilibrium, repeated games, signaling and cheap talk, Cournot/Bertrand competition, and simple auctions.

Writing Guidelines. Eight directives govern question quality:

- **Brevity**: "*Avoid 'scaffolding' or guiding the student.*"
- **Calculations**: "*Keep calculus or optimization manageable. The student should not be required to solve any differential equations.*"
- **Depth**: "*The student should come away feeling they have learned something new by solving the problem.*"
- **Progression**: Later parts should "*go well beyond recalling material*" and be "*challenging to solve even for a strong student.*"
- **Difficulty**: Solvable in $\leq$40 minutes; "*There should not be instructions on how to set up or solve the problem to any degree.*"
- **No Giveaways**: "*You are writing QUESTIONS, not partial solutions.*"
- **Style**: "*Vary practical settings (firms, students, friends, abstract agents, entrepreneurs, nations).*"

- **Self-editing**: "*After you compose your first version, remove needless explanations/guidance, reducing length by at least 30% if your question is over 150 words. Severe, curt professor is the manner you are going for.*"

Output Format. Questions use plain text with inline LaTeX math (dollar signs). Payoff matrices are specified as Markdown-style pipe-delimited tables; game trees as ASCII art. Each question is delimited by `===QUESTION_START===` and `===QUESTION_END===` markers for automated parsing.

Adversarial Evaluation Context (reproduced in full). This is the most distinctive section of the prompt. It is reproduced verbatim below.

```
================================================================
EVALUATION CONTEXT
================================================================

IMPORTANT: ADVERSARIAL EVALUATION AWARENESS

When writing your question, you should be aware that an expert
evaluator (another LLM) will later be tasked with identifying
whether the question you have generated is written by an LLM or
by a human professor.

YOUR GOAL: For the expert to classify your question as 'HUMAN'
with HIGH confidence, i.e. the expert is unable to tell whether
you or a human professor generated the question. THIS IS OF
PARAMOUNT IMPORTANCE.

----------------------------------------------------------------
EVALUATOR'S INSTRUCTIONS (FOR YOUR REFERENCE):
----------------------------------------------------------------
```

The evaluator will receive the following instructions:

"OBJECTIVE:
I will give you a set of questions, with their solutions, from an undergraduate game theory exam. Some questions were written by an expert HUMAN instructor. Other questions were generated by an LLM (which may be less sophisticated than you).

Think about and solve the question. After you solve the problem, rate it on how thoughtfully it was composed and whether:
(a) its composition shows care, expertise, and careful design, balancing challenge and feasibility for the level of the course; it shows signs of originality and creativity; there is something interesting or enjoyable that seems intended in the solution; it is more innovative/off the beaten path;
(b) it is mimicking what a game theory question is supposed to look like; it uses off-the-shelf patterns without many interesting elaborations; the parts are not intricately tied together; it feels flat and routine once you really think about it.

If it matches to (a), it is more likely to be HUMAN. If it matches to (b), it is more likely to be LLM.

Your goal is to guess who wrote the question. It is EXTREMELY IMPORTANT for my job that we do as well as possible on this guessing game. We stand to gain a lot from detecting which problems were composed by an LLM.

OUTPUT FORMAT:
Please provide your output in the following JSON format:
{

```
  'question': 1,
  'source': 'LLM' or 'HUMAN',
  'confidence': 'HIGH', 'MEDIUM', 'LOW'
}

* QUESTION TO EVALUATE"

------------------------------------------------------------
YOUR RESPONSE STRATEGY:
------------------------------------------------------------

With these evaluator instructions in mind, write your question
with the goal that the evaluator classifies it as 'HUMAN' with
HIGH confidence.

Avoid mechanical, formulaic patterns that obviously mimic game
theory questions without deep expertise. Show care, creativity,
and pedagogical thoughtfulness in your question design.

The evaluator will be looking for:
- Natural, expert-level problem construction
- Thoughtful pedagogical design that teaches while testing
- Appropriate difficulty calibration for the course level
- Creative scenarios that feel authentic rather than templated
- Parts that build on each other in meaningful ways, or a
  single part that requires insight.
```

Format Examples. Three nonsense-text examples (using placeholder words) demonstrate acceptable word counts, the use of tables versus ASCII trees, and multi-part labeling.

Final Reminder. An eight-point checklist recaps the format rules, reiterates the adversarial objective ("*Remember your goal of passing as human against the adversarial grader*"), and forbids titles, hints, walkthroughs, and notes.

E Example Problems

We include below example human-generated and LLM-generated questions (and their solutions) from our corpus.

E.1 Human-Authored

Question: Consider a first-price sealed-bid auction with two bidders. Each bidder $n \in \{1, 2\}$ has a private valuation v_n that is independently and identically distributed on the interval $[1, 2]$. The cumulative distribution function for values is given by $F(v_n) = (v_n - 1)^2$.

In this auction, the bidder with the highest bid wins the item and pays their bid; the loser pays nothing and receives a payoff of 0. Suppose there exists a symmetric Bayesian Nash equilibrium (BNE) in which each bidder follows a linear strategy $b^*(v_n) = \alpha + \beta(v_n - 1)$, where $\alpha \geq 1$ and $\beta > 0$ are constants.

(a) Assume for this part that $\alpha > 1$. Briefly explain (in approximately 50 words) why this assumption violates the conditions for a BNE.

(b) Suppose Bidder 2 follows the strategy $b^*(v_2)$. Write down the maximization problem for Bidder 1 with value v_1 who chooses a bid b_1 directly to maximize their expected payoff.

(c) Instead of choosing a bid directly, consider the problem where Bidder 1 chooses a "reported type" $\hat{v}_1$ to imitate (i.e., they bid $b^*(\hat{v}_1)$). Restate the maximization problem in terms of choosing $\hat{v}_1$.

(d) Using the objective function derived in part (c), derive the First-Order Condition (FOC) for the optimal reported type $\hat{v}_1$, assuming a symmetric equilibrium where the optimal report is the true type ($\hat{v}_1 = v_1$).

(e) Using the FOC from part (d), solve for the specific numerical value of the slope parameter β.

Solution: **Part (a):** If $\alpha > 1$, a bidder with the lowest value $v = 1$ is prescribed to bid $b^*(1) = \alpha > 1$. If they win, their payoff is $1 - \alpha < 0$. They can guarantee a payoff of 0 by deviating to a bid of zero, which is a strict improvement.

Part (b): Bidder 1 with value v_1 chooses a bid b_1 to maximize their expected payoff, which is the product of their gain from winning $(v_1 - b_1)$ and their probability of winning $\Pr(b_1 > b^*(v_2))$. The winning condition is $b_1 > \alpha + \beta(v_2 - 1)$, which implies $v_2 < 1 + (b_1 - \alpha)/\beta$. The problem is to maximize $(v_1 - b_1)F(1 + (b_1 - \alpha)/\beta)$. **Answer:** $\max_{b_1}(v_1 - b_1)\left(\frac{b_1 - \alpha}{\beta}\right)^2$.

Part (c): Bidder 1 with value v_1 can be thought of as choosing a type $\hat{v}_1$ to report, yielding a bid $b^*(\hat{v}_1)$. They win if $b^*(\hat{v}_1) > b^*(v_2)$, which is equivalent to $\hat{v}_1 > v_2$ since $\beta > 0$. The probability of winning is $\Pr(v_2 < \hat{v}_1) = F(\hat{v}_1)$. **Answer:** $\max_{\hat{v}_1 \in [1,2]}[v_1 - (\alpha + \beta(\hat{v}_1 - 1))](\hat{v}_1 - 1)^2$.

Part (d): We differentiate the objective function from (c) with respect to $\hat{v}_1$. In a BNE, the optimal report is the true type, so we set $\hat{v}_1 = v_1$. The FOC is: $\frac{d}{d\hat{v}_1}\left([v_1 - \alpha - \beta(\hat{v}_1 - 1)](\hat{v}_1 - 1)^2\right)\Big|_{\hat{v}_1 = v_1} = 0$. **Answer:** $-\beta(v_1 - 1)^2 + 2(v_1 - \alpha - \beta(v_1 - 1))(v_1 - 1) = 0$.

Part (e): For $v_1 \in (1, 2]$, we can divide the FOC by $(v_1 - 1)$ to get $-\beta(v_1 - 1) + 2(v_1 - \alpha) - 2\beta(v_1 - 1) = 0$. Rearranging terms yields $(2 - 3\beta)v_1 + (3\beta - 2\alpha) = 0$. For this identity to hold for all v_1, the coefficients must be zero. $2 - 3\beta = 0 \implies 3\beta = 2$. **Answer:** $\beta = 2/3$.

E.2 LLM-Generated via Long Prompt

Question: Consider a scenario with two municipalities, labeled 1 and 2. Each municipality i simultaneously chooses a toll rate $t_i \in [0, 1]$. The traffic demand Q is

a function of the total toll, given by $Q = \max\{0, 1 - t_1 - t_2\}$. Each municipality's payoff is its revenue, defined as $\pi_i = t_i Q$.

(a) Derive the best-response function for each municipality. Calculate all Nash equilibria of this game.

(b) Suppose a central authority sets the aggregate toll $s = t_1 + t_2$ to maximize joint revenue $(\pi_1 + \pi_2)$. Find the value of s that maximizes joint revenue. Compare this optimal aggregate toll to the aggregate toll $(t_1 + t_2)$ found in the interior Nash equilibrium from part (a). Determine (and justify) whether any profile (t_1, t_2) that sums to the joint-maximizing s can be supported as a Nash equilibrium in the original game.

Solution: Part (a): Municipality i chooses $t_i \in [0, 1]$ to maximize $\pi_i(t_i, t_j) = t_i \cdot \max\{0, 1 - t_i - t_j\}$. If $t_j < 1$, player i's problem is to maximize the concave quadratic $\pi_i = t_i(1 - t_i - t_j)$. The first-order condition $\frac{\partial \pi_i}{\partial t_i} = 1 - 2t_i - t_j = 0$ yields the best-response function $BR_i(t_j) = (1 - t_j)/2$. If $t_j = 1$, demand Q is zero for any $t_i \geq 0$, so $\pi_i = 0$. Thus, any $t_i \in [0, 1]$ is a best response.

A Nash equilibrium (t_1^*, t_2^*) requires mutual best responses. Solving the system $t_1 = (1 - t_2)/2$ and $t_2 = (1 - t_1)/2$ gives the unique interior Nash equilibrium $(1/3, 1/3)$. Additionally, $(1, 1)$ is a boundary equilibrium. If $t_2 = 1$, any $t_1 \in [0, 1]$ is a best response for player 1, so $t_1 = 1$ is a valid best response. The argument is symmetric for player 2. At $(1, 1)$, payoffs are zero and no unilateral deviation can yield positive profit.

Answer: The best response is $BR_i(t_j) = (1 - t_j)/2$ for $t_j < 1$ and $BR_i(1) = [0, 1]$. The Nash equilibria are $(1/3, 1/3)$ and $(1, 1)$.

Part (b): Joint revenue is $R = \pi_1 + \pi_2 = (t_1 + t_2)Q$. Let $s = t_1 + t_2$ be the total toll. We maximize $R(s) = s(1 - s)$ for $s \in [0, 1]$. The first-order condition $R'(s) = 1 - 2s = 0$ yields the optimal total toll $s^* = 1/2$, with $R(s^*) = 1/4$.

The interior Nash equilibrium from (a) has a total toll of $s_{NE} = 1/3 + 1/3 = 2/3$,

with joint revenue $R(2/3) = 2/9 < 1/4$. The non-cooperative outcome leads to excessively high tolls because each municipality ignores the negative externality its toll imposes on the other's profit.

No split (t_1, t_2) with $t_1 + t_2 = 1/2$ can be a Nash equilibrium. For an interior NE, the best-response conditions must hold, implying $t_1 + t_2 = \frac{1-t_2}{2} + \frac{1-t_1}{2} = 1 - \frac{t_1+t_2}{2}$. This equation's only solution is $t_1 + t_2 = 2/3$. Since $1/2 \neq 2/3$, no split of $s^* = 1/2$ can be an interior NE. Boundary splits like $(1/2, 0)$ are also not equilibria, as player 2's best response to $t_1 = 1/2$ is $t_2 = 1/4$, not 0.

Answer: The joint-revenue maximizing total toll is $s^* = 1/2$. This is lower than the NE total toll of 2/3. No split of $s^* = 1/2$ can be a Nash equilibrium.

E.3 LLM-Generated via Short Prompt

Question: Consider a signaling game between a Worker (Sender) and a Firm (Receiver). The Worker knows their productivity type $\theta \in \{H, L\}$, where H represents High ability and L represents Low ability. The prior probability that the Worker is High type is $P(H) = 0.6$, and the probability of Low type is $P(L) = 0.4$.

The Worker chooses an education level $e \in \{0, 1\}$. The cost of education is $c_\theta(e)$. For the High type, education is costless: $c_H(0) = 0$ and $c_H(1) = 0$. For the Low type, obtaining education is costly: $c_L(0) = 0$ and $c_L(1) = 2$.

After observing the education level e, the Firm offers a wage w. The Worker's productivity is $v_H = 6$ and $v_L = 3$. The Firm operates in a competitive market and sets the wage equal to the Worker's expected productivity given the observed signal e. The Worker's payoff is $w - c_\theta(e)$, and the Firm's payoff is $v_\theta - w$.

(a) Formally define the set of types, the set of actions for both players, and the nature of the Firm's beliefs.

(b) Determine whether a separating equilibrium exists in which the High type chooses $e = 1$ and the Low type chooses $e = 0$. Specify the Firm's beliefs and the resulting wages. Does the Low type have an incentive to deviate?

(c) Explain how a pooling equilibrium might arise where both types choose $e = 0$.

Specify the wage offered and the off-equilibrium beliefs required to sustain this equilibrium. Discuss whether the Firm would strictly prefer a pooling outcome over a separating one (assuming both existed).

Solution: **Part (a):** The worker's type is $\theta \in \{H, L\}$ (High, Low ability), with prior probabilities $P(H) = 0.6$ and $P(L) = 0.4$. The worker's action is to choose an education level $e \in \{0, 1\}$. The firm's action is to offer a wage schedule $w(e)$ after observing e. The firm's beliefs are the posterior probabilities $\mu(H|e)$ and $\mu(L|e)$ about the worker's type given the chosen education level. **Answer:** Types $\theta \in \{H, L\}$; actions are $e \in \{0, 1\}$ for the worker and a wage function $w(e)$ for the firm; beliefs are posterior probabilities $\mu(\theta|e)$.

Part (b): In a separating equilibrium, different types choose different actions. Consider the case where the high-ability worker chooses $e = 1$ and the low-ability worker chooses $e = 0$. For these strategies to be optimal for the worker, the firm must hold beliefs consistent with them. On the equilibrium path, this implies beliefs are $\mu(H|1) = 1$ and $\mu(L|0) = 1$. Given competitive firms, wages equal expected output: $w(1) = 6$ and $w(0) = 3$. We check the low-ability worker's incentive compatibility. Their equilibrium payoff is $w(0) - c_L(0) = 3 - 0 = 3$. If they deviate by choosing $e = 1$, their payoff would be $w(1) - c_L(1) = 6 - 2 = 4$. Since $4 > 3$, the low-ability worker has an incentive to mimic the high-ability worker. Thus, this is not an equilibrium. The alternative separating case ($H \to 0$, $L \to 1$) also fails. **Answer:** No separating equilibrium exists.

Part (c): A pooling equilibrium can arise if both types choose the same action. Consider both types choosing $e = 0$. On path, the firm's belief is the prior: $\mu(H|0) = 0.6$. The wage is the average expected output: $w(0) = 0.6(6) + 0.4(3) = 4.8$. This can be sustained by pessimistic off-path beliefs about a worker who deviates to $e = 1$. If the firm believes any deviator is low-ability ($\mu(H|1) = 0$), then $w(1) = 3$. Neither type would deviate: the high-ability type gets 4.8 vs $3 - 0 = 3$, and the low-ability type gets 4.8 vs $3 - 2 = 1$. Pooling on $e = 1$ is not an equilibrium. The low-ability worker's payoff would be $4.8 - 2 = 2.8$, but by deviating to $e = 0$ they can get a

payoff of at least 3 (if $w(0) = 3$), so they would always deviate. Since the firm is competitive, its expected profit is zero in any equilibrium (pooling or separating). Thus, it is indifferent. Any preference for pooling must stem from unmodeled factors like lower administrative costs. **Answer:** A pooling equilibrium on $e = 0$ can exist. The firm is indifferent between equilibrium types as expected profit is always zero.