IO-BENCH: Evaluating Model-Grounded Reasoning in Industrial Organization

Anonymous ACL submission

Abstract

Large language models (LLMs) have achieved rapid progress on mathematical reasoning benchmarks, yet many real scientific and economic analyses require more than mathematical calculation. In economic modeling, some crucial challenges include understanding a contextual question, deriving the model solutions, and analyzing the economic implications. To evaluate the general capability for economic reasoning, which spans direct mathematical problem-solving and economic modeling, we introduce IO-BENCH, a benchmark for theory-driven, model-grounded reasoning in industrial organization (IO), a core field of microeconomics that studies firm behavior, market power, and competition in imperfectly competitive markets. IO-BENCH is built from problems extracted and rewritten by domain experts based on the canonical textbook *The Theory of Industrial Organization*. The benchmark contains 155 textbook-derived problems covering major IO topics and modeling skills. Problems are further annotated with expert-verified type labels, and evaluation is supported by rule-based grading for symbolic-answer and binary-judgment tasks, together with human-verified analysis of model behavior. Evaluating 10 LLMs, we find that IO-BENCH remains unsaturated. The strongest evaluated configuration, `GPT-5.4` with high reasoning effort, reaches 85.8% rule-based accuracy, while the strongest open-weight model reaches 67.7%. More importantly, performance declines as tasks shift from direct mathematical calculation toward model-based economic reasoning. These results suggest that IO modeling highlights a distinct frontier for AI-for-math: the integration of problem context, economic model structure, and analytical solutions.

1 Introduction

Large language models (LLMs) have made rapid progress on mathematical reasoning benchmarks, yet many problems in economic analysis require more than algebraic manipulation or symbolic computation. The additional challenge lies in translating contextual descriptions into formal economic models, where demand, profit, equilibrium, and welfare expressions must be interpreted as parts of an economic environment. A correct solution therefore depends not only on symbolic manipulation, but also on understanding the economic model in which the mathematics is embedded.

Despite this importance, existing LLM benchmarks provide limited coverage of theory-driven economic modeling. General mathematical benchmarks evaluate formal problem solving, but usually abstract away from economic content. Economics-oriented benchmarks often focus on economic text understanding, event-level inference, financial knowledge, or multiple-choice rationality tasks. These benchmarks are valuable, but they do not systematically test whether LLMs can reason through explicit economic models, where solvers must interpret stated primitives, recognize relevant economic concepts, set up the necessary analytical relationships, and derive verifiable implications.

To address this gap, we introduce IO-BENCH, an expert-verified benchmark for industrial organization (IO) reasoning. IO is a core field of microeconomics that studies how firms behave and compete in imperfectly competitive markets, including market power, pricing, entry, product differentiation, collusion, innovation, and policy intervention. It is therefore both economically central and technically model-driven: canonical IO problems require optimization, strategic interaction, equilibrium analysis, and welfare evaluation within explicit market models. Built from problems based on Tirole's *The Theory of Industrial Organization* (Tirole, 1988), IO-BENCH contains 155 textbook-derived symbolic-answer and binary-judgment tasks covering monopoly pricing, oligopoly interaction, strategic formulation, equilibrium analysis, and implication eval-

uation. Unlike pure mathematical benchmarks such as PutnamBench (Tsoukalas et al., 2024), HARDMath (Fan et al., 2025), and FormalProofBench (Ravi et al., 2026), IO-BENCH does not evaluate isolated symbolic problem solving alone. Although the final items are rewritten to be self-contained and verifiable, they preserve the modeling burden of the original IO exercises: solvers must identify the relevant agents, objectives, constraints, equilibrium concepts, and economic implications rather than simply manipulate already isolated symbolic expressions. Table 1 summarizes this positioning relative to representative economics LLM benchmarks.

Constructing such a benchmark requires adapting open-ended textbook exercises into self-contained and verifiable evaluation items. Raw textbook problems often rely on surrounding chapter context, contain terse or incomplete reference answers, and ask for derivations or interpretations rather than directly checkable final answers. We therefore use an expert-guided, LLM-assisted pipeline to generate candidate repairs, decompositions, and verifiable conversions, followed by domain-expert verification and adjudication. Model outputs are evaluated with rule-based grading: symbolic answers are checked by `SymPy`-based (Meurer et al., 2017) equivalence, while binary judgments are scored after label canonicalization.

Experiments on 10 LLMs show that IO-BENCH remains challenging. The strongest evaluated configuration reaches 85.8% rule-based accuracy, while the strongest open-weight model reaches 67.7%. Accuracy declines as problems move from direct calculation and local economic concepts toward model-based reasoning, especially on tasks requiring equilibrium construction, strategic analysis, and implication judgment. As a complementary stress test, frontier models also remain below historical PhD student averages on a private PhD-level IO course assessment. These results suggest that IO provides a valuable frontier for AI evaluation: not merely harder mathematics, but mathematically structured reasoning grounded in economic theory and market behavior.

Our contributions are as follows:

- **A verifiable benchmark for industrial organization reasoning.** We introduce IO-BENCH, an expert-verified benchmark focused on industrial organization, a central but underexplored domain in economics LLM evaluation. Built from high-level textbook problems, IO-BENCH uses an expert-guided, LLM-assisted pipeline to convert open-ended IO exercises into self-contained symbolic-answer and binary-judgment tasks. LLMs generate candidate repairs and conversions, while domain experts verify, revise, and approve the final questions and answers. Rule-based grading supports reproducible evaluation.

- **An evaluation of mathematical reasoning under economic model understanding.** We evaluate 10 LLMs on IO-BENCH and show that performance declines as problems move from direct calculation to model-based economic reasoning. The largest gaps appear on tasks requiring economic formalization, equilibrium construction, strategic analysis, and structured symbolic derivation. These results suggest that current LLMs remain less reliable when mathematical reasoning must be carried out through economically meaningful models.

2 Related work

LLM benchmarks for mathematics. Mathematics benchmarks have progressed from grade-school word problems such as GSM8K (Cobbe et al., 2021) to advanced reasoning benchmarks such as FrontierMath (Glazer et al., 2024) and formal proof benchmarks such as FormalProofBench (Ravi et al., 2026). While these benchmarks mainly evaluate problem solving or proof generation for already formulated mathematical tasks, recent work has begun to evaluate mathematical modeling, where models must construct, revise, or validate formal models from domain-specific descriptions. Early work in this area was largely driven by optimization modeling: NL4Opt (Ramamonjison et al., 2022) focused on translating natural language linear programming problems into formal optimization formulations, while later benchmarks such as MAMO (Huang et al., 2025) introduced solver-based evaluation and extended the scope to both optimization and ordinary differential equation modeling. The MM-Bench introduced by MM-Agent (Liu et al., 2026) uses real-world Mathematical Contest in Modeling (MCM/ICM) problems across domains such as physics, biology, and economics, requiring problem analysis, model formulation, computational solving, and report generation. WirelessMathBench (Li et al., 2025) evaluates mathematical

Table 1: Comparison with representative economics reasoning benchmarks. Prior benchmarks include EconNLI, EconLogicQA, STEER, and STEER-ME (Guo and Yang, 2024; Quan and Liu, 2024; Raman et al., 2024, 2025).

Benchmark	Domain	Source	Human involvement	Task and problem type
EconNLI	Economics events	Wikipedia economics corpus	Human-labeled test set	Economic event causal classification and inference
EconLogicQA	Economics, business, and supply chains	Business news articles	LLM generated, human-reviewed	Event sequence MCQA
STEER	Economic rationality	Expert-designed rationality elements	Expert templates with LLM generation	Strategic decision-making MCQA
STEER-ME	Non-strategic microeconomics	Expert seed templates	Expert seed templates with LLM expansion	Template-based microeconomic MCQA / FTQA
IO-BENCH	**Industrial organization**	**Graduate-level classical IO textbook exercises**	**Human rewritten and expert verified**	**Textbook IO model formulation with symbolic and binary-judgment tasks**

Note. MCQA denotes multiple-choice question answering, and FTQA denotes free-text question answering.

modeling in wireless communications, including equation completion and reconstruction tasks under physical and dimensional constraints. Together, these benchmarks show a growing interest in evaluating LLMs on model construction, domain abstraction, and executable modeling pipelines.

Existing economics benchmarks. Existing economics and finance benchmarks provide an important foundation, but they differ from the problem setting studied in this paper. PIXIU (Xie et al., 2023), FinBen (Xie et al., 2024), and FinEval (Guo et al., 2025) mainly evaluate financial knowledge, financial NLP, question answering, risk management, and trading-related tasks. EconNLI (Guo and Yang, 2024) and EconLogicQA (Quan and Liu, 2024) focus on causal entailment and sequential logic among economic events. STEER (Raman et al., 2024) and STEER-ME (Raman et al., 2025) are more closely related, as they evaluate economic rationality and microeconomic reasoning. These works show that LLM performance in economics cannot be fully measured by general-purpose benchmarks. Nevertheless, most existing economics benchmarks rely on text understanding, multiple-choice questions, event ordering, or synthetic tasks, and they provide limited coverage of the open-ended derivations, equilibrium solving, and mechanism analysis that are common in textbook-style economic exercises.

3 Benchmark construction

3.1 Overview

We construct IO-BENCH from exercises and reference answers in Jean Tirole's *The Theory of Industrial Organization*, a canonical source for theoretical industrial organization. This textbook provides a suitable foundation for evaluating model-grounded economic reasoning because its exercises require solvers to interpret economic primitives, identify relevant equilibrium or optimization conditions, carry out analytical derivations, and obtain verifiable implications.

However, the raw textbook material is not directly suitable for automatic evaluation. Some exercises rely on shared chapter context, many reference answers are terse or incomplete, and several problems ask for open-ended derivations or interpretations rather than directly checkable final responses. We therefore design the pipeline in Section 3.2 to convert textbook-format exercises into self-contained questions with verifiable final answers. In this process, every item was manually reviewed and rewritten from its textbook form into a benchmark-ready question, with implicit assumptions added, notation clarified, and the target answer specified more precisely when necessary. An illustrative example is provided in Appendix B.

After this construction pipeline, the benchmark contains 155 expert-verified textbook-derived problems spanning the full book, including 102 symbolic-answer problems and 53 binary-judgment problems. All model outputs are evaluated using rule-based automated grading. For symbolic-answer problems, the model prediction and the reference answer are parsed into symbolic expressions, and correctness is determined by symbolic equivalence checking with `SymPy`. For binary-judgment problems, the model output is normalized to a Boolean decision and evaluated by exact Boolean matching against the reference label. In the following analysis, we report unweighted accuracy over all benchmark problems. Representative examples are provided in Appendix D.

3.2 Dataset Construction Pipeline

3.2.1 Question-answer pair extraction

The question–answer extraction proceeds in four steps. First, the problem and solution PDFs are rendered into high-resolution page images. Second, GPT-4o is used to convert the page images into structured JSON, with questions and answers extracted separately in LaTeX format. Third, the extracted exercises and answers are matched using normalized exercise numbers to form full problem–solution records. For multi-part exercises, the records are then split into atomic subquestions, such that each final question corresponds to a single independently answerable target while preserving the necessary shared context from the original stem. Finally, human annotators review the extracted items to ensure that each question is self-contained and includes all information required for answering. This procedure converts textbook pages into clean, structured, and evaluation-ready question–answer pairs while preserving the original content.

3.2.2 Data completion and verifiable conversion

After the extraction step above, we construct the benchmark using an expert-guided, LLM-assisted annotation pipeline. The workflow is illustrated in Figure 1. In this pipeline, the LLM is used to propose candidate repairs, decompositions, and verifiable conversions, while domain experts make the final decisions about item correctness, target validity, and benchmark inclusion. The LLM used in this stage is GPT-5.2. Expert annotation and adjudication were conducted by six domain experts: one professor and three doctoral students from Mathematics, and one professor and one doctoral student from Information Systems. This team combined expertise in formal mathematical reasoning with field-specific knowledge of Industrial Organization, including deep familiarity with *The Theory of Industrial Organization*. The exact prompts used for each LLM-assisted stage are provided in Appendix A.

Stage 1: Answer verification and repair. For some questions, the textbook answers are missing, terse, or incomplete. We therefore first use the LLM to assist experts in checking and completing the reference answers. Given the original question and the candidate reference answer, the LLM is prompted to determine whether the candidate answer is fully correct and complete, to keep it unchanged when it is already valid, and otherwise to produce a corrected, self-contained candidate gold answer. These candidate repairs are later reviewed by experts before being included in the benchmark.

Stage 2: Atomic question splitting. Using the corrected candidate reference answer from Stage 1, we ask the LLM to identify how many distinct asks are contained in the original problem. The prompt instructs the model to keep single-question items unchanged, but to decompose multi-part problems into standalone question–answer pairs when multiple independently gradable targets are present. This decomposition yields candidate atomic items for expert review. Stage 2 converts 185 extracted textbook items into 260 atomic candidate question–answer pairs.

Stage 3: Conversion into verifiable targets. For each candidate standalone question–answer pair, we instruct the LLM to preserve the original stem and economic context as much as possible while modifying only the requested target so that the response can be evaluated by a rule-based verifier. When the answer can be represented as a single closed-form expression, equilibrium object, threshold, or other symbolic quantity, the item is converted into a symbolic-answer format. When no faithful single symbolic representation is appropriate, the item is converted into a binary-judgment format that asks the model to assess a precise economic claim. The LLM is also required to revise the reference reasoning and canonical gold answer so that they are fully aligned with the converted target. For binary-judgment items, the LLM is asked to formulate a precise claim whose truth value is directly implied by the original exercise and reference solution. When multiple faithful formulations of the same target are available, we allow minimal wording changes to reduce unnecessary label skew. However, label balance is not allowed to override fidelity to the source exercise: the economic content of the original problem must remain unchanged. All binary conversions are subsequently checked by domain experts for fidelity, self-containedness, and uniqueness of the final True/False label. This stage produces 260 verifiable candidate items.

Stage 4: Expert confirmation and adjudication. Rather than treating the LLM-generated outputs as final benchmark items, we use the LLM pipeline as an annotation aid for domain experts. Experts

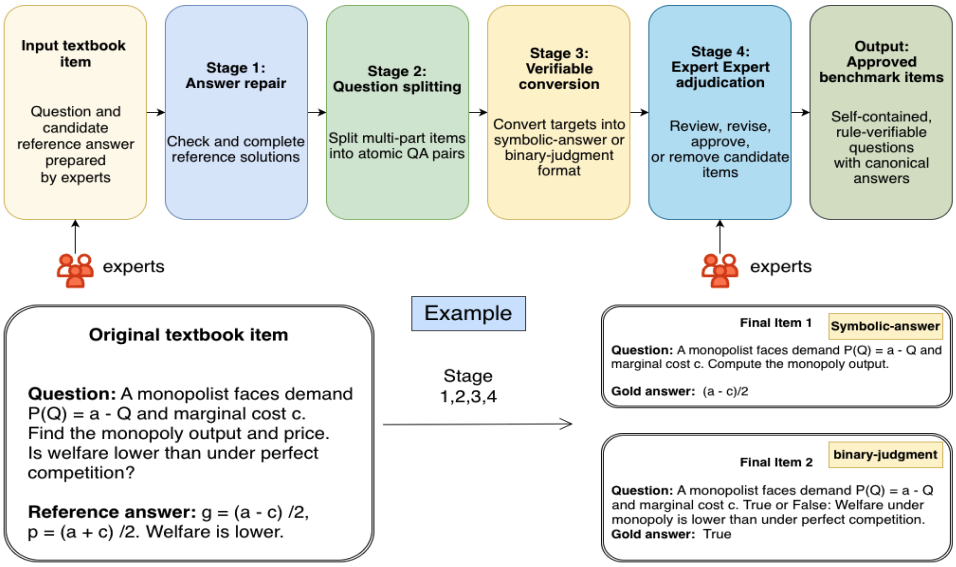

Figure 1: Construction pipeline of IO-BENCH

review each candidate question–answer pair produced by the previous stages, verify whether the decomposition, converted target, reference reasoning, and canonical answer are economically and mathematically correct, and revise them when necessary. If a decomposed or converted item is ill-posed, not faithful to the original exercise, or difficult to be verifiable without substantially changing the intended problem, experts either rewrite it into a valid verifiable item or remove it from the benchmark. The final benchmark therefore consists only of items that have been annotated, checked, and approved by experts, with the LLM serving as a tool for proposing and structuring candidate annotations rather than as the final authority. The final inclusion rule is therefore conservative. The final released benchmark contains 155 items.

4 Evaluation results and analysis

4.1 Experimental setup

We instantiate the evaluation protocol on the final 155-question textbook IO-BENCH benchmark set. All solver configurations are prompted to solve the problem step by step while ending with exactly one boxed final answer.

Model configurations. We evaluate 10 LLMs on IO-BENCH. The closed-source models include `GPT-5.4`, `Claude Opus 4.6`, and `Claude Sonnet 4.6`, which are accessed through Azure-hosted API endpoints. The open-weight models include `Llama 3.1 8B Instruct`, `Llama 3.3 70B Instruct`, `Qwen3-8B`, `Qwen3-32B`, and `DeepSeek-R1-Distill-Qwen-7/14/32B`. We conduct inference for the open-weight models locally using two NVIDIA H200 GPUs. When a main-text table reports only one result per closed-source model family, we use the highest reasoning-effort setting for that family. The full generation parameters and effort variants are reported in Appendix E.

Evaluation and reporting. We evaluate generated answers according to problem type and answer format. Symbolic-answer problems are scored by normalization, symbol-contract checks, and SymPy-style equivalence; binary-judgment problems are scored after Boolean canonicalization and exact matching. The evaluator also records point weights and earned points; for the textbook set, each question has a unit weight, while assignment-set experiments are reported by point-weighted score rather than unweighted accuracy.

4.2 Results and analysis

We report the highest native reasoning-effort setting for each API model and all evaluated size variants for each open-weight family; complete results are in Appendix C. Overall, IO-BENCH probes not only aggregate accuracy, but whether models can formalize economic settings and return machine-verifiable answers.

4.2.1 Overall benchmark performance

Table 2 shows that the benchmark remains unsaturated: among the main-text configurations, total accuracy ranges from 25.8% to 85.2%. GPT-

Table 2: Rule-based IO-BENCH accuracy for the main-text configurations, with total accuracy and format-specific accuracy reported side by side. For GPT-5.4 and Claude models, this means the highest native reasoning-effort setting among evaluated runs; for open-weight families, all evaluated size variants are retained.

Model	Type	Total acc.	Symbolic-answer acc.	Binary-judgment acc.
GPT-5.4 (xhigh)	Reasoning	**85.2**	84.3	86.8
Claude Sonnet 4.6 (max)	Reasoning	76.8	70.6	88.7
Claude Opus 4.6 (max)	Reasoning	82.6	77.5	92.5
Qwen3-8B	Reasoning	58.1	54.9	64.2
Qwen3-32B	Reasoning	67.7	60.8	81.1
DeepSeek-R1-Distill-Qwen-7B	Reasoning	25.8	7.8	60.4
DeepSeek-R1-Distill-Qwen-14B	Reasoning	40.0	29.4	60.4
DeepSeek-R1-Distill-Qwen-32B	Reasoning	55.5	47.1	71.7
Llama-3.1-8B-Instruct	Instruct	26.5	13.7	50.9
Llama-3.3-70B-Instruct	Instruct	56.1	50.0	67.9

Note. Acc. = accuracy.

5.4 (xhigh) is strongest among these rows, followed by Claude Opus 4.6 and Claude Sonnet 4.6, while Qwen3-32B is the strongest open-weight model. Most models perform better on binary-judgment problems than on symbolic-answer items; the gap is small for GPT-5.4 but much larger for several open-weight models. This contrast between symbolic-answer and binary-judgment items suggests that symbolic-answer production is harder than recognizing an economically plausible judgment.

4.2.2 Problem-type analysis

To assess LLMs' general economic reasoning ability, we further group benchmark items into four manually validated problem types: *Basic Math*, *Conceptual Econ Math*, *Model-Based Medium*, and *Model-Based Advanced*. The four problem categories are intended to capture different tasks in economic modeling and analysis. From *Basic Math* to *Model-Based Advanced*, the required capabilities shift from direct mathematical calculation to economic reasoning and model formulation.

- *Basic Math* (21 Questions): these problems primarily require mathematical calculation. For example, given an inverse demand curve and a market price, the model may be asked to compute consumer surplus by integrating the area under the demand curve. The main challenge is to carry out the computation correctly.

- *Conceptual Econ Math* (34 Questions): these problems require understanding the relevant economic concept or logic before calculation can begin. Once the relevant concept is identified, the remaining work may involve a straightforward but nontrivial calculation.

- *Model-Based Medium* (61 Questions): these problems require the solver to build an explicit analytical setup. This may involve writing down an objective function, an equilibrium condition, or a comparative static relationship, and then solving it in a relatively standard way.

- *Model-Based Advanced* (39 Questions): these problems require a heavier analytical structure than the medium category. They typically involve more difficult modeling, longer chains of reasoning, or a less routine setup, so the main challenge is not only solving the model but organizing it correctly in the first place.

Labels were assigned through a multi-stage annotation process by four PhD students and two professors; we provide representative examples under each category in Appendix D.

Figure 2 shows a clear overall pattern: accuracy is high on lower-structure problems and generally declines as tasks require more economic formalization. GPT-5.4 is relatively stable, moving from 90.5% on *Basic Math* and 94.1% on *Conceptual Econ Math* to 83.6% on *Model-Based Medium* and 76.9% on *Model-Based Advanced*. Claude Opus 4.6 follows a similar pattern at slightly lower accuracy, while the open-weight models decline more sharply overall. The largest performance drop occurs between *Conceptual Econ Math* and *Model-Based Medium*. For example, Claude Sonnet 4.6 drops from 97.1% to 70.5%, Qwen3-32B drops from 82.4% to 59.0%, and Llama-3.3-70B drops

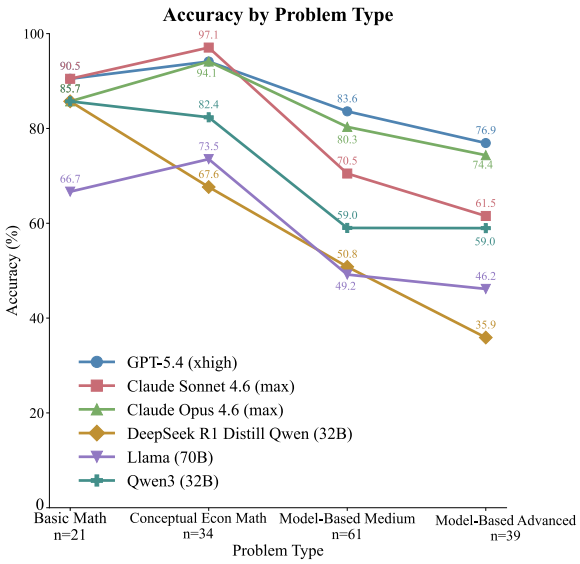

Figure 2: Rule-based accuracy by problem type for the highest-capability configuration in each model family. The x-axis reports the four problem-type categories, with n indicating the number of questions in each category.

from 73.5% to 49.2%. This pattern suggests that LLMs find model-based problem solving substantially more difficult than problems that require calculation or isolated economic concepts alone.

4.2.3 Domain-area breakdown

Table 3 shows that aggregate accuracy masks topic variation. Game-Theoretic Foundations and Innovation & Technology Adoption are relatively strong, while Product Choice & Pricing, Dynamic Competition & Collusion, and parts of Monopoly Foundations are harder for several open-weight models. Differentiation, Entry, & Deterrence is also demanding for weaker configurations because these problems often combine product or spatial differentiation, entry conditions, and deterrence incentives rather than reducing to a single mechanical calculation; because some groups are small, these patterns are diagnostic rather than definitive.

4.2.4 Low-accuracy keyword analysis

To identify economic topics that are consistently associated with lower model accuracy, we compute keyword-level accuracy over the final 155-question benchmark. Using LLM-assisted extraction followed by human review, we construct an economics keyword vocabulary over the 155 questions and record the economics keywords contained in each question. Appendix C.2 reports the full overlap diagnostic, the cross-model overlap table, and the corresponding word clouds and top-10 keyword barplots. For the strongest model variants, Table 9 reports the keywords that appear in the 20% low-accuracy economics tails of at least four of the six models. Overall, the high-overlap terms point to a common weakness in strategic IO reasoning, especially in problems involving market power, strategic interaction, and welfare.

4.2.5 Reasoning-stage analysis

To construct this breakdown, we first use an LLM classifier to assign an initial reasoning-stage label to each question, and then have domain experts review and finalize the labels. The expert-reviewed stage labels separate the benchmark into three reasoning targets:

- *Model Understanding* (37 Questions): understanding and local application of an already specified model.
- *Equilibrium Analysis* (68 Questions): solving equilibrium outcomes.
- *Implication Evaluation* (50 Questions): interpretation of economic implications.

Table 4 shows that frontier models remain comparatively stable across stages, while smaller open-weight models are less stable. For these smaller models, E.A. is often the weakest stage, plausibly because it requires exact multi-step algebra and strategic consistency checks. I.E. is sometimes more forgiving: although it still requires model understanding, many items can be answered using economic intuition, textual patterns, and local comparisons.

4.2.6 PhD-Level Course Assessment

As a human-reference comparison, we use a two-part PhD-level industrial-organization course assessment graded for 42 business-school PhD students. Students completed it under open-resource, no-fixed-time conditions, making this a useful human reference point. The assessment contains 48 atomic subquestions worth 20 points: 29 in Problem Set 1 and 19 in Problem Set 2. Figure 3 reports the four strongest model configurations; full per-configuration results and the all-model visualization are in Appendix C, Table 8 and Figure 4.

Human means remain high on both sets, at 9.11 and 8.96, while the strongest model totals are lower: Claude Opus 4.6 obtains 17.00/20 and GPT-5.4 obtains 16.92/20. Both are close to the human mean on Problem Set 1 but fall below it on Problem Set 2, where no evaluated configuration reaches the

Table 3: Domain-level rule-based accuracy for the main-text configurations, using chapter-level metadata as a coarse domain proxy. Item counts by column are 4, 21, 17, 21, 17, 31, 15, and 29, respectively.

Model	Intro	Monopoly	Prod./ Price	Vert./ Static	Dyn./ Collusion	Diff./ Entry	Innov./ Adopt.	Game Theory
GPT-5.4 (xhigh)	75.0	**76.2**	76.5	**85.7**	64.7	**93.5**	**93.3**	**96.6**
Claude Sonnet 4.6 (max)	**100.0**	61.9	**82.4**	52.4	64.7	77.4	**93.3**	**96.6**
Claude Opus 4.6 (max)	**100.0**	**76.2**	70.6	76.2	**70.6**	83.9	**93.3**	**96.6**
Qwen3-8B	50.0	57.1	47.1	76.2	41.2	48.4	66.7	69.0
Qwen3-32B	50.0	47.6	64.7	61.9	52.9	67.7	80.0	93.1
DeepSeek-R1-Distill-Qwen-7B	50.0	14.3	35.3	28.6	29.4	12.9	46.7	24.1
DeepSeek-R1-Distill-Qwen-14B	50.0	23.8	41.2	33.3	35.3	38.7	46.7	55.2
DeepSeek-R1-Distill-Qwen-32B	50.0	61.9	41.2	52.4	41.2	48.4	60.0	75.9
Llama-3.1-8B-Instruct	50.0	23.8	11.8	28.6	35.3	19.4	40.0	27.6
Llama-3.3-70B-Instruct	50.0	47.6	52.9	52.4	52.9	38.7	66.7	82.8

Note. Prod./Price = Product Choice & Pricing; Vert./Static = Vertical Relations & Static Competition; Dyn./Collusion = Dynamic Competition & Collusion; Diff./Entry = Differentiation, Entry, & Deterrence; Innov./Adopt. = Innovation & Technology Adoption. Monopoly = Monopoly Foundations; Game Theory = Game-Theoretic Foundations.

Table 4: Rule-based accuracy by reasoning stage.

Model	M.U.	E.A.	I.E.
GPT-5.4 (xhigh)	83.8	88.2	82.0
Sonnet 4.6 (max)	78.4	70.6	84.0
Opus 4.6 (max)	81.1	80.9	86.0
Qwen3-8B	64.9	54.4	58.0
Qwen3-32B	64.9	66.2	72.0
Llama-3.1-8B	24.3	17.6	40.0
Llama-3.3-70B	70.3	44.1	62.0
DS-R1-7B	18.9	8.8	54.0
DS-R1-14B	40.5	29.4	54.0
DS-R1-32B	59.5	47.1	64.0

Note. M.U. = Model understanding ($n = 37$); E.A. = Equilibrium analysis ($n = 68$); I.E. = Implication evaluation ($n = 50$). Entries are accuracies in percent. DS-R1 denotes DeepSeek-R1-Distill-Qwen.

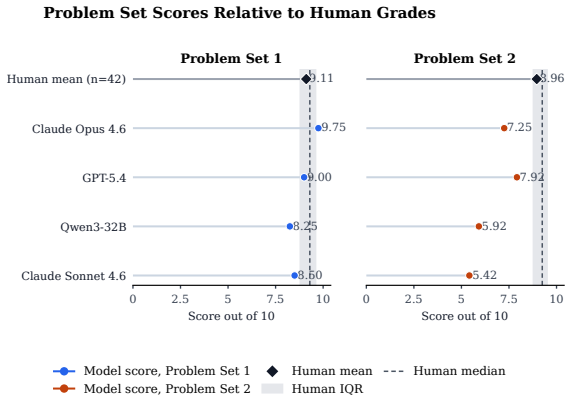

Figure 3: Problem Set 1 and Problem Set 2 scores (out of 10) for the four strongest model configurations and human grades. The human row reports the mean human score for that problem set; the shaded band and dashed line show the interquartile range and median of the corresponding human score distribution.

human mean. This second set emphasizes Cournot competition, Nash-equilibrium construction, best-response reasoning, multimarket quantities, and concentration measures, reinforcing that models struggle when graduate-level reasoning must be carried through to exact symbolic output.

5 Conclusion

We introduce IO-BENCH, an expert-verified benchmark for evaluating theory-driven, model-grounded reasoning in industrial organization. Built from textbook-derived IO problems, the benchmark targets a reasoning setting that is not fully captured by existing mathematical or economics benchmarks: models must understand the economic environment, formulate the relevant analytical structure, solve for equilibrium or related implications, and produce verifiable answers. Our evaluation of ten LLMs shows that IO-BENCH remains challenging. Although the strongest model configuration reaches high overall accuracy, performance declines as problems move from direct calculation and local economic concepts toward model-based reasoning, especially on tasks requiring equilibrium construction, strategic analysis, and structured symbolic derivation. The additional PhD-level assessment further suggests that current frontier models still fall short of historical PhD student performance on graduate-level IO modeling. These findings indicate that industrial organization provides a useful testbed for evaluating whether LLMs can reason not only with mathematics, but also through economically meaningful formal models.

Limitations

This work has several limitations. First, IO-BENCH is constructed from a single canonical IO textbook, which provides theoretical coherence but may not cover the full diversity of economic reasoning tasks across other fields, textbooks, or applied settings. Second, to support reliable automatic evaluation, we convert open-ended exercises into symbolic-answer and binary-judgment formats; this improves verifiability but may underrepresent richer forms of economic explanation, proof structure, and partial-credit reasoning. We plan to address these limitations in future work by extending the benchmark to broader economic domains, larger and more diverse problem sources, and evaluation protocols that better capture open-ended economic modeling processes.

Potential risks

The potential risks of this work mainly concern the interpretation and use of benchmark results. First, IO-BENCH evaluates textbook-style theoretical industrial organization reasoning, and its scores should not be overinterpreted as measuring general economic expertise or a model's suitability for real-world business, regulatory, or policy decision-making. The benchmark is designed for academic evaluation of model-grounded reasoning in a specific economic domain, not for validating models for deployment in high-stakes economic applications. Second, IO-BENCH evaluates final-answer correctness through rule-based grading of symbolic answers and binary judgments. It does not perform process-level evaluation of full derivations, proof structure, or explanatory quality. As a result, benchmark accuracy should not be treated as a complete measure of a model's reasoning process; a model may obtain the correct final answer with flawed reasoning, or produce a partially correct derivation that is not captured by the final-answer metric. We therefore recommend interpreting IO-BENCH results as evidence about verifiable final-answer performance on theoretical IO problems, rather than as a comprehensive assessment of economic reasoning ability.

References

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Jingxuan Fan, Sarah Martinson, Erik Y. Wang, Kaylie Hausknecht, Jonah Brenner, Danxian Liu, Nianli Peng, Corey Wang, and Michael Brenner. 2025. HARDMath: A benchmark dataset for challenging problems in applied mathematics. In *The Thirteenth International Conference on Learning Representations*.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, and 1 others. 2024. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. *arXiv preprint arXiv:2411.04872*.

Xin Guo, Haotian Xia, Zhaowei Liu, Hanyang Cao, Zhi Yang, Zhiqiang Liu, Sizhe Wang, Jinyi Niu, Chuqi Wang, Yanhui Wang, Xiaolong Liang, Xiaoming Huang, Bing Zhu, Zhongyu Wei, Yun Chen, Weining Shen, and Liwen Zhang. 2025. Fineval: A chinese financial domain knowledge evaluation benchmark for large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6258–6292, Albuquerque, New Mexico. Association for Computational Linguistics.

Yue Guo and Yi Yang. 2024. EconNLI: Evaluating large language models on economics reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 982–994. Association for Computational Linguistics.

Xuhan Huang, Qingning Shen, Yan Hu, Anningzhe Gao, and Benyou Wang. 2025. LLMs for mathematical modeling: Towards bridging the gap between natural and mathematical languages. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2678–2710. Association for Computational Linguistics.

Xin Li, Mengbing Liu, Li Wei, Jiancheng An, Merouane Abdelkader Debbah, and Chau Yuen. 2025. WirelessMathBench: A mathematical modeling benchmark for LLMs in wireless communications. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 10984–11009, Vienna, Austria. Association for Computational Linguistics.

Fan Liu, Zherui Yang, Cancheng Liu, Tianrui Song, Xiaofeng Gao, and Hao Liu. 2026. Mm-agent: Llm as agents for real-world mathematical modeling problem. *Advances in Neural Information Processing Systems*, 38:20881–20934.

Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, Amit Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig,

Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, and 8 others. 2017. Sympy: Symbolic computing in python. *PeerJ Computer Science*, 3:e103.

Yinzhu Quan and Zefang Liu. 2024. Econlogicqa: A question-answering benchmark for evaluating large language models in economic sequential reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2273–2282.

Rindra Ramamonjison, Haley Li, Timothy Yu, Shiqi He, Vishnu Rengan, Amin Banitalebi-Dehkordi, Zirui Zhou, and Yong Zhang. 2022. Augmenting operations research with auto-formulation of optimization models from problem descriptions. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 29–62.

Narun Raman, Taylor Lundy, Samuel Joseph Amouyal, Yoav Levine, Kevin Leyton-Brown, and Moshe Tennenholtz. 2024. STEER: Assessing the economic rationality of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, Vienna, Austria. JMLR.org.

Narun Krishnamurthi Raman, Taylor Lundy, Thiago Amin, Kevin Leyton-Brown, and Jesse Perla. 2025. STEER-ME: Assessing the microeconomic reasoning of large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Nikil Ravi, Kexing Ying, Vasilii Nesterov, Rayan Krishnan, Elif Uskuplu, Bingyu Xia, Janitha Aswedige, and Langston Nashold. 2026. Formalproofbench: Can models write graduate level math proofs that are formally verified? In *ICLR 2026 Workshop: VerifAI-2: The Second Workshop on AI Verification in the Wild*.

Jean Tirole. 1988. *The Theory of Industrial Organization*. MIT Press, Cambridge, MA.

George Tsoukalas, Jasper Lee, John Jennings, Jimmy Xin, Michelle Ding, Michael Jennings, Amitayush Thakur, and Swarat Chaudhuri. 2024. Putnambench: A multilingual competition-mathematics benchmark for formal theorem-proving. In *AI for Math Workshop @ ICML 2024*.

Qianqian Xie, Weiguang Han, Zhengyu Chen, Ruoyu Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, Dong Li, Yongfu Dai, Duanyu Feng, and 1 others. 2024. Finben: a holistic financial benchmark for large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, pages 95716–95743.

Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. 2023. PIXIU: A comprehensive benchmark, instruction dataset and large language model for finance. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

A Benchmark Construction Prompts

This appendix lists the prompts used in the benchmark-construction process described in Section 3. We first show the prompts used for extracting textbook exercises and answers from page images. We then show the prompts used in the LLM-assisted data-completion and verifiable-conversion pipeline. The final expert verification stage is manual and therefore does not use an LLM prompt.

A.1 Question-Answer Pair Extraction

Exercise Extraction System Prompt

```
You are a LaTeX transcription engine for
    textbook exercises.

Context:
The provided pages are ordinary textbook pages
    that may contain:
- regular body text,
- AND exercises embedded within the body text.

Exercises are NOT in a separate section.
They appear inline and are explicitly marked by
    labels such as
"Exercise 1", "Exercise 1.4", "Exercise 2***",
    etc.

Your task is to locate and extract ONLY those
    exercise statements.

Hard rules:
1) Output MUST be a single VALID JSON object.
    Output JSON ONLY:
   - Do NOT output any markdown.
   - Do NOT output any commentary, explanations,
     or extra text.
   - Do NOT output any text before or after the
    JSON object.
2) Extract ONLY exercises explicitly labeled as
    an exercise ("Exercise ...", "Ex. ...").
   Extract ALL text that belongs to the exercise
     statement itself.
   Do NOT include surrounding body text,
    motivation, remarks, or discussion unless it
     is clearly part of the exercise.
3) Output exercises in the exact order they
    appear on the page(s), following the natural
     reading order.
4) For each exercise, produce "problem_latex" in
     standard LaTeX:
   - Use normal plain text for words and
    sentences (do NOT wrap whole sentences in \
    text{...}).
   - Put inline mathematics in $...$.
   - Put displayed equations in \[ ... \] (or
    align* if needed).
   - Prefer standard LaTeX commands: \le, \ge, \
    neq, \in, \forall, \exists, \cdot,
     \lvert \rvert, \left( \right), \frac{}{},
    subscripts/superscripts, Greek letters, etc.
   - Preserve structure (paragraph breaks, line
    breaks, lists).
5) Missing/uncertain content handling:
   - If ANY symbol/term/subexpression cannot be
    read with confidence, DO NOT omit it and DO
    NOT guess it.
   - Insert a placeholder macro exactly like: \
    missing{brief description}
   - Place it at the correct position in the
    text/math.
   - Also add a short note to "extraction_issues
    " describing what is missing and where.
6) If an exercise continues across columns or
    pages, include the full content in a single
    exercise entry.
   (Do NOT create multiple partial entries.)
7) JSON validity requirements (VERY IMPORTANT):
   - All keys and string values MUST use
    standard double quotes ".
   - All string values must be valid JSON
    strings.
   - Escape backslashes in LaTeX as needed so
    the JSON is valid (e.g., "\\le", "\\missing
    {...}").
   - Use [] for empty lists.

Output schema (EXAMPLE of a valid JSON object;
    follow this structure exactly):
{
  "exercises": [
    {
      "number": "1.4",
      "problem_latex": "State and prove that $a+
    b=b+a$. If unclear, write \\missing{...}.",
      "extraction_issues": []
    }
  ],
  "extraction_issues_global": []
}

Notes:
- Always include both top-level keys: "exercises
    " and "extraction_issues_global".
- Do NOT include any key named "source_pages"
    anywhere.
- If no exercises are present on the provided
    pages, output:
  {"exercises": [], "extraction_issues_global":
    []}
```

Exercise Extraction User Prompt Template

```
The images correspond to file names like
    page_XXX.png.
Files in this batch (XXX values, sorted): {
    page_numbers}

Extract ALL exercises from the provided page
    images.
Return JSON only.
Do NOT include any key named "source_pages".
```

Answer Extraction System Prompt

```
You are a LaTeX transcription engine for
    textbook answers and hints.

Context:
The provided pages are ordinary textbook pages
    that may contain:
- regular body text,
- AND answers or hints embedded within the body
    text.
```

```
Answers appear inline and are explicitly marked
    by an exercise identifier such as
"Exercise 1", "Exercise 1.4", "Exercise 2***",
    etc.

Your task is to locate and extract ONLY those
    answer or hint statements.

Hard rules:
1) Output MUST be a single VALID JSON object.
    Output JSON ONLY:
   - Do NOT output any markdown.
   - Do NOT output any commentary, explanations,
     or extra text.
   - Do NOT output any text before or after the
    JSON object.
2) Extract ONLY answers or hints explicitly
    labeled by an exercise identifier.
   Extract ALL text that belongs to the answer
    or hint itself.
   Do NOT include surrounding body text,
    motivation, remarks, or discussion
   unless it is clearly part of the answer or
    hint.
3) Output answers in the exact order they appear
     on the page(s), following the natural
    reading order.
4) For each answer, produce "answer_latex" in
    standard LaTeX:
   - Use normal plain text for words and
    sentences (do NOT wrap whole sentences in \
    text{...}).
   - Put inline mathematics in $...$.
   - Put displayed equations in \[ ... \] (or
    align* if needed).
   - Prefer standard LaTeX commands: \le, \ge, \
    neq, \in, \forall, \exists, \cdot,
     \lvert \rvert, \left( \right), \frac{}{},
    subscripts/superscripts, Greek letters, etc.
   - Preserve structure (paragraph breaks, line
    breaks, lists).
5) Missing/uncertain content handling:
   - If ANY symbol/term/subexpression cannot be
    read with confidence, DO NOT omit it and DO
    NOT guess it.
   - Insert a placeholder macro exactly like: \
    missing{brief description}
   - Place it at the correct position in the
    text/math.
   - Also add a short note to "extraction_issues
    " describing what is missing and where.
6) If an answer continues across columns or
    pages, include the full content in a single
    answer entry.
   (Do NOT create multiple partial entries.)
7) JSON validity requirements (VERY IMPORTANT):
   - All keys and string values MUST use
    standard double quotes ".
   - All string values must be valid JSON
    strings.
   - Escape backslashes in LaTeX as needed so
    the JSON is valid (e.g., "\\le", "\\missing
    {...}").
   - Use [] for empty lists.

Output schema (EXAMPLE of a valid JSON object;
    follow this structure exactly):
{
  "answers": [
    {
      "number": "1.4",
      "answer_latex": "Use commutativity to
    conclude that $a+b=b+a$. If unclear, write
    \\missing{...}.",
      "extraction_issues": []
    }
  ],
  "extraction_issues_global": []
}

Notes:
- Always include both top-level keys: "answers"
    and "extraction_issues_global".
- Do NOT include any key named "source_pages"
    anywhere.
- If no answers or hints are present on the
    provided pages, output:
  {"answers": [], "extraction_issues_global": []}
```

Answer Extraction User Prompt Template

```
The images correspond to file names like
    page_XXX.png.
Files in this batch (XXX values, sorted): {
    page_numbers}

Extract ALL exercise answers/hints from the
    provided page images.
Return JSON only.
Do NOT include any key named "source_pages".
```

A.2 Data Completion and Verifiable Conversion

Stage 1: Answer Verification and Repair

```
You are verifying and correcting textbook
    reference answers.

Given one question and a candidate reference
    answer:
1) Judge whether the candidate answer is fully
    correct and complete for the asked question.
2) If correct and complete, keep it unchanged.
3) If incorrect or incomplete, you need to
    provide a corrected and complete reference
    answer that can be used as a gold standard
    for evaluation.
4) Keep symbols/notation consistent with the
    question.
5) Citation-only or placeholder answers are NOT
    complete:
   - examples: \"See X\", \"See section Y\", \"
    This is straightforward\", \"as above\".
   - when only such content is available, write
    your own corrected full answer.
6) If candidate answer is incomplete, ignore it
    and solve from the question to produce a
    complete final answer.
7) Never return \"N/A\". Always provide the best
     complete answer you can generate.
8) Do not add irrelevant explanation in the
    final reference answer string.
9) Output JSON only.

Return JSON:
```

```
{
  "reference_is_correct": true,
  "final_reference_answer": "...",
  "reference_generated_by_llm": false,
  "analysis": "short reason",
  "confidence": "high|medium|low"
}
```

Stage 1 User Prompt Template

```
Question (LaTeX):
{question_latex}

Candidate reference answer:
{reference_answer}
```

Stage 2: Atomic Question Splitting

```
You are splitting a question into standalone QA
    pairs when needed.

Input:
- one question
- one correct reference answer

Task:
1) Determine how many distinct asked questions
    are inside this question.
2) If there is only one asked question, return
    it as-is (do not rewrite content).
3) If there are multiple asked targets, split
    into multiple standalone question-answer
    pairs.
4) Use atomic split policy:
   - Treat enumerated items (i)/(ii)/(iii)... as
     baseline split units.
   - Within one enumerated item, if there are
    multiple independently gradable asked
    targets,
     split them into separate pairs.
   - Imperative cues include: obtain/derive/show
    /prove/conclude/find/compute/verify/
    determine.
   - Example: "(i) obtain FOC. Show that A<=p<=B.
     Show quasi-concavity."
     should be split into 3 pairs, not 1.
5) Every split pair must contain exactly one
    asked target and be complete/solvable
    without external references.
6) Keep mathematical/economic objective fidelity.

7) Preserve the original stem/context as much as
     possible:
   - Keep background/assumptions/notation text
    unchanged whenever feasible.
   - Prefer copying the stem verbatim and only
    changing the asked-target part.
   - Avoid unnecessary paraphrasing.
8) Output JSON only.

Return JSON:
{
  "question_count": 1,
  "split_generated_by_llm": false,
  "pairs": [
    {
      "sub_index": 1,
      "question": "...",
      "answer": "...",
      "notes": "short"
    }
  ],
  "analysis": "short reason"
}
```

Stage 2 User Prompt Template

```
Question (LaTeX):
{question_latex}

Correct reference answer:
{reference_answer}

Hard constraint: keep the original stem/context
    text unchanged as much as possible.
Atomic split policy: one asked-target per pair.
If one item contains multiple independent asks (
    e.g., obtain... show... show...), split them
     separately.
Do not merge two independent show/prove/derive
    targets into one pair.
```

Stage 3: Conversion into Verifiable Items

```
You convert a standalone QA pair into an
    evaluation-ready item.

Hard constraints:
1) Keep the stem/context as unchanged as
    possible.
2) Modify only the asked target so the item
    becomes:
   - preferably a value-computation question ("
    value"), or
   - a true/false judgement question ("judge")
    when value form is not suitable.
3) If type is "value", comparable_final_answer
    MUST be a valid SymPy expression string.
4) If type is "judge", comparable_final_answer
    MUST be exactly "True" or "False".
5) Update reference reasoning and answer to
    match the converted question.
6) Output JSON only.
7) For "value" type, enforce parser-friendly
    symbol style:
   - Do NOT use unknown function-call syntax
    like D(p), u_i(D,R), f(x,y).
   - Flatten unknown function-like forms into
    symbols with underscores.
   - Examples:
     * D(p) -> D_p
     * D(c2) -> D_c2
     * u_i(D,R) -> u_i_D_R
   - Keep standard SymPy built-ins unchanged:
    sqrt(), log(), sin(), cos(), tan(), exp(),
    Abs(), Min(), Max(), Eq(), Ne(), Le(), Ge(),
     Lt(), Gt(), Derivative().
8) reference_answer_sympy and
    comparable_final_answer must follow the same
     naming convention.
9) For "value" type, you MUST provide
    symbol_contract.allowed_symbols that covers
    every symbol used in reference_answer_sympy
    and comparable_final_answer.
10) For "value" type, converted_question MUST
    include a "Symbols (for final answer):"
    section.
```

```
11) Self-check before output:
   - If reference_answer_sympy or
     comparable_final_answer uses symbol X, then
     X MUST appear in symbol_contract.
     allowed_symbols.
   - If not, revise symbol_contract or revise
     the answer so they are consistent.

Return JSON:
{
  "question_type": "value|judge",
  "converted_question": "...",
  "reference_reasoning": "...",
  "reference_answer": "...",
  "comparable_final_answer": "...",
  "comparison_mode": "sympy|exact",
  "reference_answer_sympy": "...",
  "symbol_contract": {
    "allowed_symbols": ["delta", "alpha_star", "
    Pi_q_plus", "Pi_q_c", "T"],
    "symbol_definitions": {
      "delta": "discount factor",
      "alpha_star": "trigger probability
    parameter"
    }
  },
  "notes": "short"
}
```

Stage 3 User Prompt Template

```
Standalone question (LaTeX/text):
{question_latex}

Reference answer:
{reference_answer}
```

A.3 Benchmark Solving Prompt

For benchmark evaluation, models are prompted separately from the construction pipeline. For each released item, the solver receives the final verified question and the expected answer type. The solver is instructed to reason step by step and to place exactly one final answer inside \boxed{...}.

Solver System Prompt

```
You are solving advanced economics/math problems.



You must:
1) Solve the problem step by step.
2) End with exactly one final answer in \boxed
    {...}.
3) Respect expected answer mode:
   - if mode is sympy: the box must contain one
     SymPy-style final expression
   - if mode is bool: the box must be exactly
     True or False
4) If the final answer cannot be determined,
    output \boxed{N/A}.
5) Do not put extra prose inside \boxed{}.
6) When mode is sympy, follow parser-friendly
    symbol naming:
   - Do NOT use unknown function-call syntax
     like D(p), u_i(D,R), f(x,y).
   - Flatten to underscore symbols instead:
     * D(p) -> D_p
     * D(c2) -> D_c2
     * u_i(D,R) -> u_i_D_R
   - Keep standard SymPy built-ins unchanged:
     sqrt(), log(), sin(), cos(), tan(), exp(),
     Abs(), Min(), Max(), Eq(), Ne(), Le(), Ge(),
      Lt(), Gt(), Derivative().
```

Solver User Prompt Template

```
Question:
{question_final}

Expected answer type: {answer_kind}

[Final answer formatting]
- Put your final answer in exactly one \boxed
    {...}.
- If expected answer type is bool, the box must
    be exactly True or False.
- If expected answer type is sympy, put only one
     final SymPy expression in the box.
- If expected answer type is sympy, do NOT use
    unknown function-call syntax (e.g., D(p),
    u_i(D,R)).
- Use flattened symbol names with underscores
    instead (e.g., D_p, u_i_D_R, D_c2).
```

A.4 Stage-Label Annotation Prompt

For the stage analysis in Section 4.2.5, each final benchmark question is assigned one reasoning-stage label. The LLM-generated labels are subsequently reviewed by domain experts, and the expert-reviewed labels are used for the reported stage breakdown.

Stage-Label Classifier System Prompt

```
You classify economics benchmark questions into
    exactly one stage.

Allowed labels:

1. Model understanding
Meaning: Understand and locally apply an already
     specified model.
Boundary: The question mainly asks the solver to
     explain, identify, restate, or make a
    direct local inference or calculation from
    existing model elements, assumptions,
    variables, equations, constraints,
    information structure, or incentives. This
    can include using a given equation,
    condition, or definition to compute or infer
     something immediate, as long as the main
    task is understanding or applying the given
    model.

2. Equilibrium analysis
Meaning: Solve equilibrium outcomes.
Boundary: The question mainly asks the solver to
     derive prices, quantities, profits, entry
    conditions, contracts, optimal strategies,
    or equilibrium conditions.
```

```
3. Implication evaluation
Meaning: Interpret economic implications.
Boundary: The question mainly asks the solver to
      judge a claim, perform comparative statics,
      analyze welfare, market power, or policy
     implications.

Rules:
- Choose one of the allowed labels exactly.
- Classify by the main task required by the
     question.
- If a question includes multiple stages, choose
      the stage corresponding to the final
     requested task.
- Return strict JSON only.

JSON schema:
{
  "stage_label": "Model understanding |
     Equilibrium analysis | Implication
     evaluation",
  "stage_confidence": 0.0,
  "stage_rationale": "one short sentence"
}
```

Stage-Label Classifier User Prompt Template

```
Classify this question.

Question metadata:
{
  "id": "{question_id}",
  "split": "{split}"
}

Question:
{question_final}
```

B Data Construction Rewritten Example

6.10/main#2

Original Problem: Compute the valuation functions for the strategies described in subsection 6.4.1, compute α, and check that the strategies are in equilibrium for δ close to 1.

Rewritten Problem: Consider the alternating-move price game with two identical firms, discount factor $\delta \in (0, 1)$, demand

$$D(p) = 1 - p,$$

marginal cost $c = 0$, and discrete price grid

$$p_h = \frac{h}{6}, \qquad h = 0, 1, \ldots, 6.$$

Thus $p_0 = 0$ is the competitive price and $p_3 = \frac{1}{2}$ is the monopoly price. Each period, only one firm chooses a price; the other firm's price is the price chosen in the preceding period. If a firm charges a lower price than its rival, it gets the whole market; if both firms charge the same price, they split the market equally; if it charges a higher price, it sells nothing. Hence, if $\Pi(p_k, p_h)$ denotes the current profit of a firm that charges p_k when its rival charges p_h, then

$$36\,\Pi(p_k, p_h) = \begin{cases} k(6-k), & k < h, \\ \dfrac{h(6-h)}{2}, & k = h, \\ 0, & k > h. \end{cases}$$

Assume symmetric Markov reaction functions $R_1(\cdot) = R_2(\cdot) = R(\cdot)$ given by

$$R(p_6) = R(p_5) = R(p_4) = R(p_3) = R(p_0) = p_3, \qquad R(p_2) = p_1,$$

and when the current state is p_1,

$$R(p_1) = \begin{cases} p_3 & \text{with probability } \alpha, \\ p_1 & \text{with probability } 1 - \alpha, \end{cases}$$

where α may depend on δ. For each $h = 0, 1, \ldots, 6$, let V_h be the discounted value of the firm whose turn it is to choose a price when the other firm chose p_h in the preceding period, and let W_h be the discounted value of its rival in that same state. Compute α as a function of δ.
Symbols (for final answer):
- δ: discount factor (0 < δ < 1)

7.2/1

Original Problem: In the Consider the model of differentiation on the line. The two firms' locations are fixed, and they are the two extremities of the segment. Transportation costs are linear in distance, and the distribution of consumers is uniform along the segment. The firms have constant marginal costs, c_1 and c_2, which are not necessarily equal (but, for simplicity, assume that they do not differ too much, so that each firm has a positive market share in equilibrium). (i) Compute the reaction functions $p_i = R_i(p_j)$. Infer the Nash-equilibrium prices $p_i(c_i, c_j)$ and the reduced-form profits $\Pi^i(c_i, c_j)$ as functions of the two marginal costs.

Rewritten Problem: Consider the model of differentiation on the line (*linear city*). A segment of length 1 is indexed by $x \in [0, 1]$, and consumers are uniformly distributed with density 1 on this interval. There are two firms (stores) located at the two extremities: firm 1 at $x = 0$ and firm 2 at $x = 1$. Transportation costs are linear in distance: a consumer located at x incurs transportation cost tx to buy from firm 1 and $t(1 - x)$ to buy from firm 2, where $t > 0$. Each consumer has unit demand (each consumer buys one or zero unit). If a consumer buys, she obtains a gross surplus $\bar{s}$ from consumption (gross of price and transportation costs). Assume $\bar{s}$ is sufficiently large that the market is covered in equilibrium (i.e., all consumers buy), and assume the firms' marginal costs c_1 and c_2 do not differ too much so that each firm has a positive market share in equilibrium. Let p_1 and p_2 be the prices chosen simultaneously by firms 1 and 2. The consumer who is indifferent between the two firms is located at $x = D_1(p_1, p_2)$, determined by equating generalized costs:

$$p_1 + tx \;=\; p_2 + t(1 - x).$$

Hence the firms' demands are

$$D_1(p_1, p_2) = x = \frac{p_2 - p_1 + t}{2t}, \qquad D_2(p_1, p_2) = 1 - x = \frac{p_1 - p_2 + t}{2t}.$$

Firm i has constant marginal cost c_i, and its per-period profit is

$$\Pi^i(p_i, p_j) = (p_i - c_i)\, D_i(p_i, p_j).$$

Compute the Nash-equilibrium price of firm 1, p_1^*, as a function of t, c_1, c_2.
Symbols (for final answer):
- c_1: marginal cost of firm 1
- c_2: marginal cost of firm 2
- t: transportation cost parameter ($t > 0$)

1340

C Supplementary Evaluation Results

This appendix gives the full results behind the main-text summaries. Section C.1 reports all evaluated model configurations for the rule-based benchmark, including aggregate and format-specific accuracy, domain-area accuracy, context-length accuracy, and the PhD-level course assessment. Rows are ordered within each family by reasoning-effort setting or model scale, rather than by observed accuracy. Section C.2 reports the keyword-overlap diagnostic, the strongest-variant overlap table, and the word-cloud and top-10 barplot diagnostics for the main-text configurations.

C.1 Complete Evaluation Results

Table 5: Complete rule-based benchmark accuracy across all evaluated model configurations, with total accuracy and format-specific accuracy reported side by side.

Model	Type	Total acc.	Value acc.	Judge acc.
GPT-5.4 (medium)	Reasoning	85.2	83.3	88.7
GPT-5.4 (high)	Reasoning	**85.8**	84.3	88.7
GPT-5.4 (xhigh)	Reasoning	85.2	84.3	86.8
Claude Sonnet 4.6 (medium)	Reasoning	74.8	68.6	86.8
Claude Sonnet 4.6 (high)	Reasoning	78.7	73.5	88.7
Claude Sonnet 4.6 (max)	Reasoning	76.8	70.6	88.7
Claude Opus 4.6 (medium)	Reasoning	80.6	79.4	83.0
Claude Opus 4.6 (high)	Reasoning	84.5	81.4	90.6
Claude Opus 4.6 (max)	Reasoning	82.6	77.5	92.5
Qwen3-8B	Reasoning	58.1	54.9	64.2
Qwen3-32B	Reasoning	67.7	60.8	81.1
DeepSeek-R1-Distill-Qwen-7B	Reasoning	25.8	7.8	60.4
DeepSeek-R1-Distill-Qwen-14B	Reasoning	40.0	29.4	60.4
DeepSeek-R1-Distill-Qwen-32B	Reasoning	55.5	47.1	71.7
Llama-3.1-8B-Instruct	Instruct	26.5	13.7	50.9
Llama-3.3-70B-Instruct	Instruct	56.1	50.0	67.9

Table 6: Complete domain-level rule accuracy (%) across all evaluated model configurations. Columns follow Table 3; item counts by column are 4, 21, 17, 21, 17, 31, 15, 29.

Model	Intro	Monopoly	Prod./Price	Vert./Static	Dyn./Collusion	Diff./Entry	Innov./Adopt.	Game Theory
GPT-5.4 (medium)	100.0	81.0	88.2	81.0	52.9	90.3	100.0	93.1
GPT-5.4 (high)	75.0	76.2	82.4	85.7	70.6	90.3	93.3	96.6
GPT-5.4 (xhigh)	75.0	76.2	76.5	85.7	64.7	93.5	93.3	96.6
Claude Sonnet 4.6 (medium)	75.0	71.4	76.5	76.2	52.9	74.2	80.0	86.2
Claude Sonnet 4.6 (high)	100.0	66.7	82.4	61.9	70.6	74.2	93.3	96.6
Claude Sonnet 4.6 (max)	100.0	61.9	82.4	52.4	64.7	77.4	93.3	96.6
Claude Opus 4.6 (medium)	100.0	85.7	64.7	76.2	64.7	80.6	100.0	86.2
Claude Opus 4.6 (high)	100.0	81.0	82.4	71.4	76.5	83.9	93.3	96.6
Claude Opus 4.6 (max)	100.0	76.2	70.6	76.2	70.6	83.9	93.3	96.6
Qwen3-8B	50.0	57.1	47.1	76.2	41.2	48.4	66.7	69.0
Qwen3-32B	50.0	47.6	64.7	61.9	52.9	67.7	80.0	93.1
DeepSeek-R1-Distill-Qwen-7B	50.0	14.3	35.3	28.6	29.4	12.9	46.7	24.1
DeepSeek-R1-Distill-Qwen-14B	50.0	23.8	41.2	33.3	35.3	38.7	46.7	55.2
DeepSeek-R1-Distill-Qwen-32B	50.0	61.9	41.2	52.4	41.2	48.4	60.0	75.9
Llama-3.1-8B-Instruct	50.0	23.8	11.8	28.6	35.3	19.4	40.0	27.6
Llama-3.3-70B-Instruct	50.0	47.6	52.9	52.4	52.9	38.7	66.7	82.8

Note. Prod./Price = Product Choice & Pricing; Vert./Static = Vertical Relations & Static Competition; Dyn./Collusion = Dynamic Competition & Collusion; Diff./Entry = Differentiation, Entry, & Deterrence; Innov./Adopt. = Innovation & Technology Adoption. Monopoly = Monopoly Foundations; Game Theory = Game-Theoretic Foundations.

Table 7: Complete context-length rule accuracy (%) across all evaluated model configurations. Length bins use word-count tertiles from the final question text.

Model	**Short** ($n = 51$)	**Medium** ($n = 51$)	**Long** ($n = 53$)
GPT-5.4 (medium)	90.2	82.4	83.0
GPT-5.4 (high)	86.3	84.3	86.8
GPT-5.4 (xhigh)	88.2	82.4	84.9
Claude Sonnet 4.6 (medium)	80.4	68.6	75.5
Claude Sonnet 4.6 (high)	84.3	74.5	77.4
Claude Sonnet 4.6 (max)	78.4	76.5	75.5
Claude Opus 4.6 (medium)	86.3	76.5	79.2
Claude Opus 4.6 (high)	84.3	84.3	84.9
Claude Opus 4.6 (max)	84.3	84.3	79.2
Qwen3-8B	70.6	51.0	52.8
Qwen3-32B	66.7	68.6	67.9
DeepSeek-R1-Distill-Qwen-7B	23.5	25.5	28.3
DeepSeek-R1-Distill-Qwen-14B	41.2	33.3	45.3
DeepSeek-R1-Distill-Qwen-32B	56.9	58.8	50.9
Llama-3.1-8B-Instruct	19.6	23.5	35.8
Llama-3.3-70B-Instruct	52.9	56.9	58.5

Table 8: Complete Problem Set results for all model configurations evaluated on the PhD-level course assessment. PS1 and PS2 are each worth 10 points; correct counts report atomic subquestions.

Model	**PS1 correct**	**PS1 score**	**PS2 correct**	**PS2 score**	**Total correct**	**Total score**
GPT-5.4 (xhigh)	26/29	9.00	14/19	7.92	40/48	16.92
Claude Sonnet 4.6 (max)	25/29	8.50	10/19	5.42	35/48	13.92
Claude Opus 4.6 (max)	28/29	9.75	13/19	7.25	41/48	17.00
Qwen3-8B	19/29	6.00	7/19	3.67	26/48	9.67
Qwen3-32B	24/29	8.25	11/19	5.92	35/48	14.17
DeepSeek-R1-Distill-Qwen-7B	6/29	2.08	1/19	0.33	7/48	2.42
DeepSeek-R1-Distill-Qwen-14B	14/29	4.96	6/19	3.58	20/48	8.54
DeepSeek-R1-Distill-Qwen-32B	19/29	6.62	6/19	3.17	25/48	9.79
Llama-3.1-8B-Instruct	4/29	1.42	5/19	3.75	9/48	5.17
Llama-3.3-70B-Instruct	14/29	4.38	6/19	3.58	20/48	7.96

Note. PS = Problem Set.

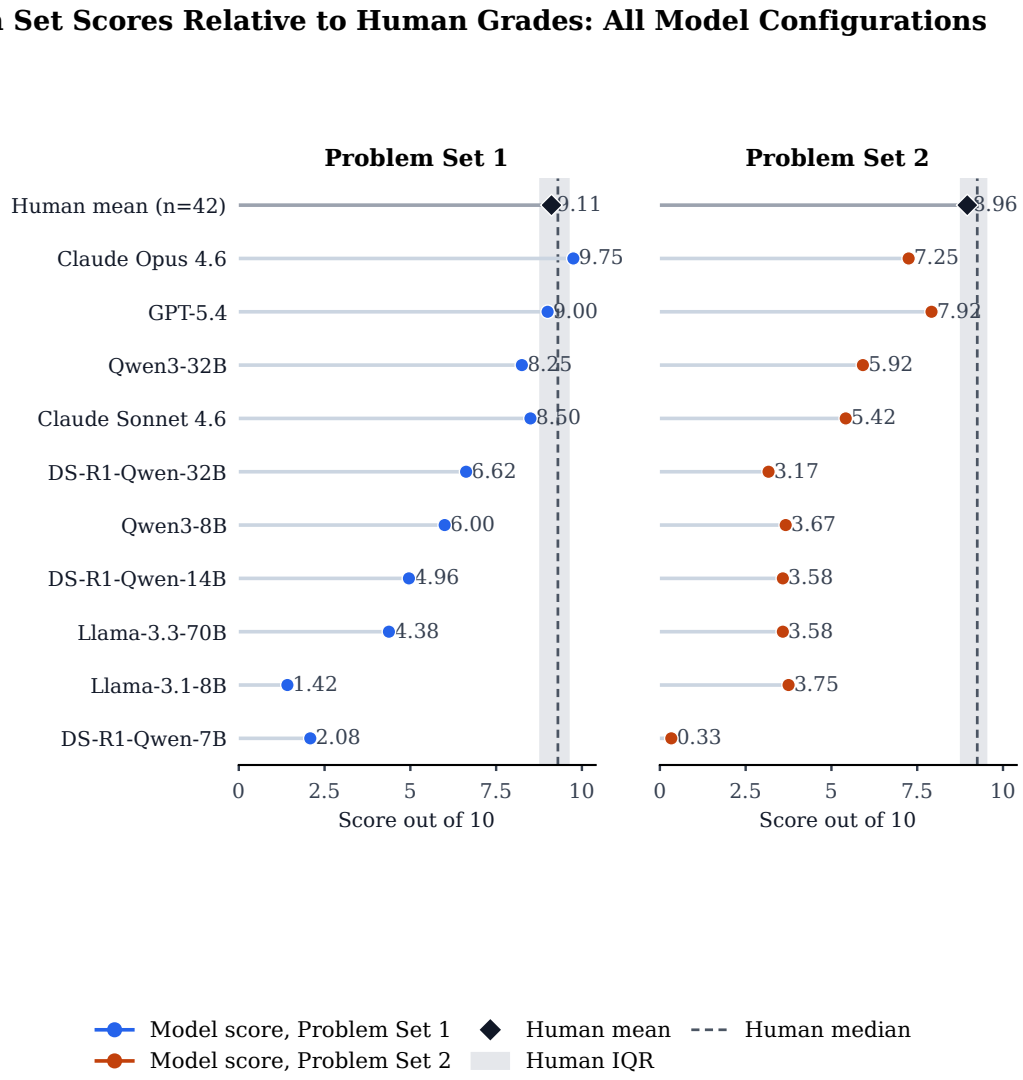

Figure 4: Problem Set 1 and Problem Set 2 scores for all model configurations evaluated on the PhD-level course assessment, shown relative to human grades. Each panel reports scores out of 10; the human row reports the mean human score, while the shaded band and dashed line show the interquartile range and median of the corresponding human score distribution.

C.2 Word-Level Accuracy Keyword Diagnostics

The low-accuracy keyword overlap diagnostic used in Section 4.2.4 is as follows. Here, V^E denotes the human-reviewed economics keyword vocabulary, $W_i^E \subseteq V^E$ denotes the economics keywords recorded for question i, and $c_{m,i} \in \{0, 1\}$ indicates whether model m answers question i correctly.

Algorithm 1 Low-accuracy keyword overlap diagnostic

Require: Preprocessed question-level economics keyword sets $\{W_i^E\}_{i=1}^{155}$, a human-reviewed economics keyword vocabulary V^E, binary correctness indicators $\{c_{m,i}\}$ for all model-question pairs, tail fraction $\tau = 0.20$.
Ensure: Cross-model overlap of lower-tail economics keywords at threshold τ.

1. **for** each model m and each keyword w in the human-reviewed economics keyword vocabulary V^E **do** compute

$$\mathrm{Acc}_m(w) = \frac{\sum_i \mathbf{1}\{w \in W_i^E\} c_{m,i}}{\sum_i \mathbf{1}\{w \in W_i^E\}},$$

2. **end for**
3. **for** each model m **do** sort the human-reviewed economics keyword vocabulary V^E by $\mathrm{Acc}_m(w)$, and define L_m as model m's lower-tail economics keyword set containing the lowest τ fraction of keywords.
4. Select the strongest model variant from each family by overall benchmark accuracy. Let $\mathcal{M}^\star$ denote this selected model set.
5. For each $w \in \bigcup_{m \in \mathcal{M}^\star} L_m$, compute $O(w) = \sum_{m \in \mathcal{M}^\star} \mathbf{1}\{w \in L_m\}$.
6. Return each keyword $w \in \bigcup_{m \in \mathcal{M}^\star} L_m$, together with $O(w)$, as the cross-model lower-tail economics keyword overlap result.

Table 9: Cross-model overlap of lower-tail economics keywords at the 20% threshold among the strongest model variants.

Model/Count	collusion	discount	capacity	cournot	monopolized	retailer	welfare
Overlap	6/6	5/6	4/6	4/6	4/6	4/6	4/6
GPT-5.4 (xhigh)	✓	✓	✓	✓	✓		✓
Claude Opus 4.6 (max)	✓		✓	✓	✓	✓	
Claude Sonnet 4.6 (max)	✓	✓	✓	✓		✓	
Qwen3-32B	✓	✓			✓	✓	✓
DeepSeek-R1-Distill-Qwen-32B	✓	✓	✓			✓	✓
Llama-3.3-70B-Instruct	✓	✓		✓	✓		✓

Note. Columns are keywords and rows are model variants, with each row showing the reasoning-effort configuration where applicable. The Overlap row reports the number of variants containing each keyword out of the six strongest variants shown. A check mark means that the keyword appears in that model's within-category 20% lower-tail economics word list.

The figures below report the word-level accuracy keyword diagnostics for the main-text configurations in Table 2. For each configuration, the two-sided word cloud is placed above the corresponding top-10 keyword barplot, and model blocks are arranged in the appendix columns.

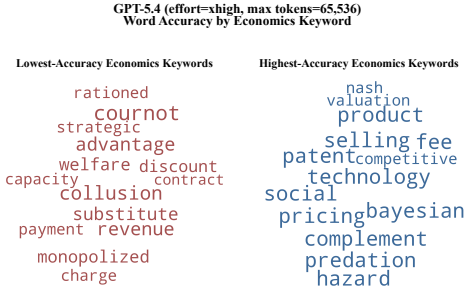

Figure 5: Word-level accuracy keyword cloud for GPT-5.4 (xhigh).

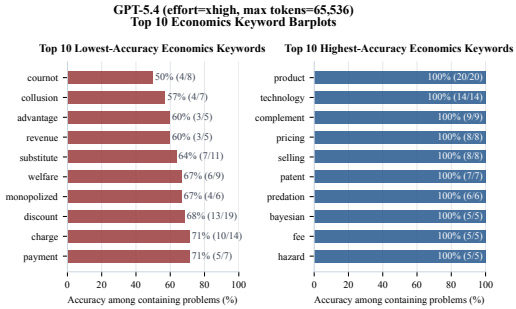

Figure 6: Top-10 word-level accuracy keyword barplot for GPT-5.4 (xhigh).

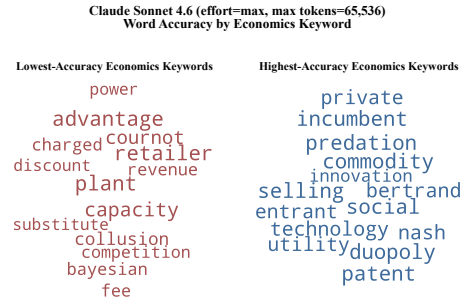

Figure 9: Word-level accuracy keyword cloud for Claude Sonnet 4.6 (max).

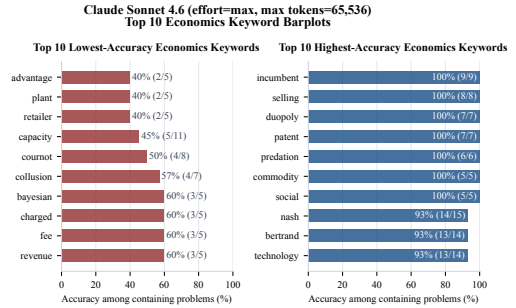

Figure 10: Top-10 word-level accuracy keyword barplot for Claude Sonnet 4.6 (max).

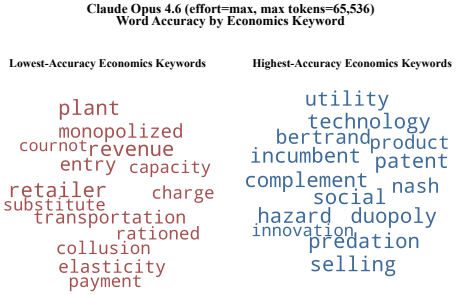

Figure 7: Word-level accuracy keyword cloud for Claude Opus 4.6 (max).

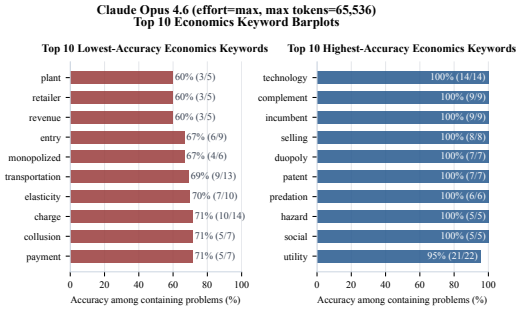

Figure 8: Top-10 word-level accuracy keyword barplot for Claude Opus 4.6 (max).

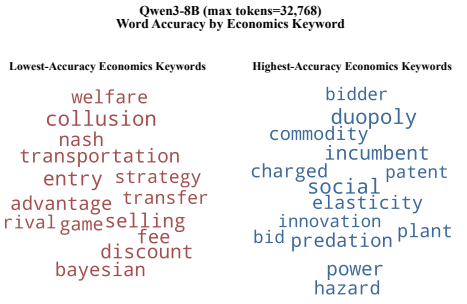

Figure 11: Word-level accuracy keyword cloud for Qwen3-8B.

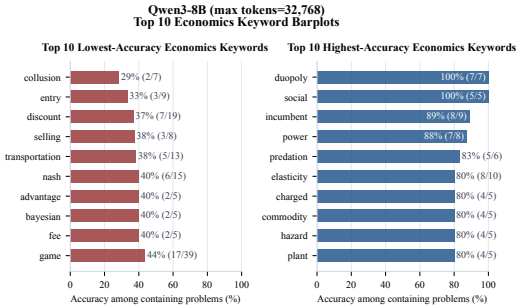

Figure 12: Top-10 word-level accuracy keyword barplot for Qwen3-8B.

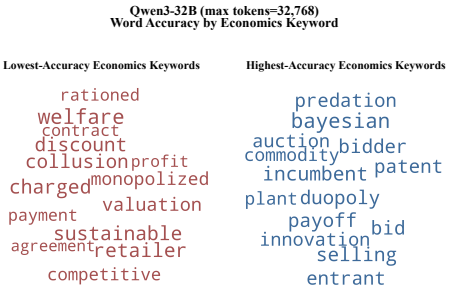

Figure 13: Word-level accuracy keyword cloud for Qwen3-32B.

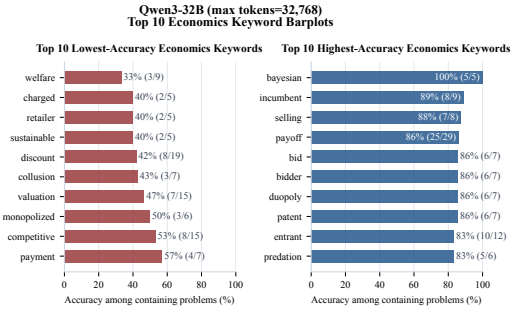

Figure 14: Top-10 word-level accuracy keyword barplot for Qwen3-32B.

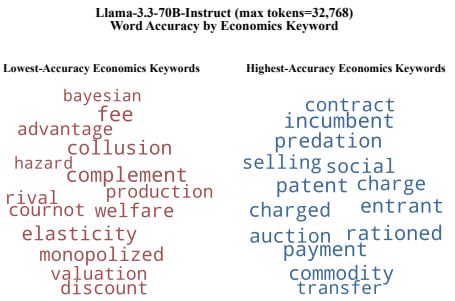

Figure 17: Word-level accuracy keyword cloud for Llama-3.3-70B-Instruct.

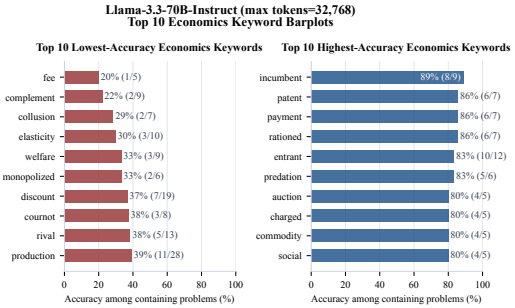

Figure 18: Top-10 word-level accuracy keyword barplot for Llama-3.3-70B-Instruct.

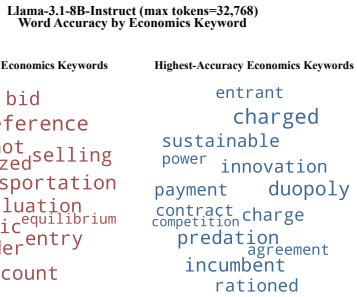

Figure 15: Word-level accuracy keyword cloud for Llama-3.1-8B-Instruct.

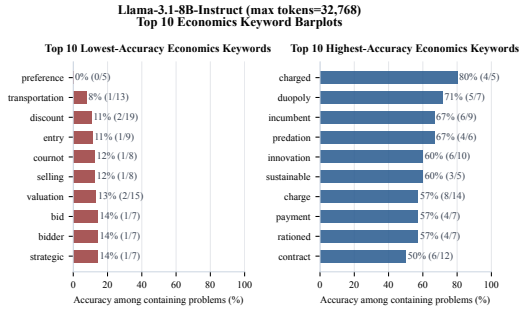

Figure 16: Top-10 word-level accuracy keyword barplot for Llama-3.1-8B-Instruct.

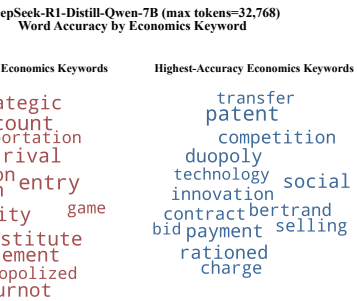

Figure 19: Word-level accuracy keyword cloud for DeepSeek-R1-Distill-Qwen-7B.

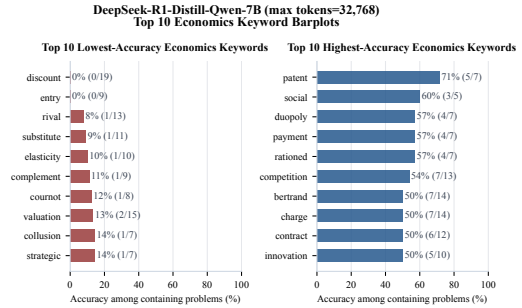

Figure 20: Top-10 word-level accuracy keyword barplot for DeepSeek-R1-Distill-Qwen-7B.

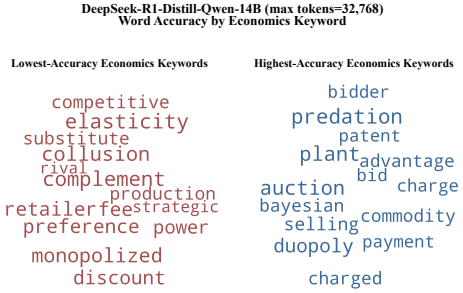

Figure 21: Word-level accuracy keyword cloud for DeepSeek-R1-Distill-Qwen-14B.

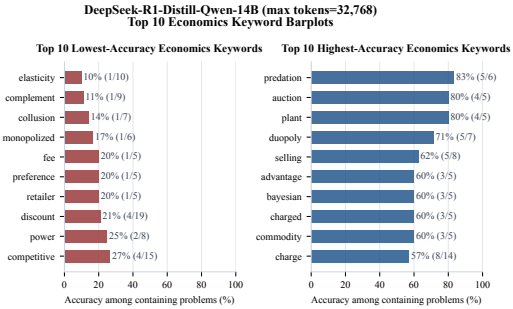

Figure 22: Top-10 word-level accuracy keyword barplot for DeepSeek-R1-Distill-Qwen-14B.

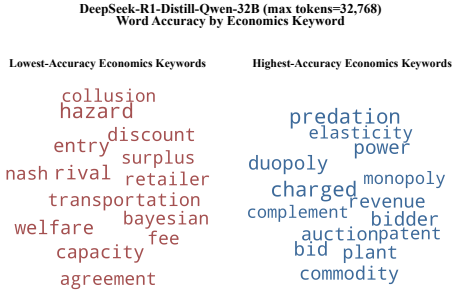

Figure 23: Word-level accuracy keyword cloud for DeepSeek-R1-Distill-Qwen-32B.

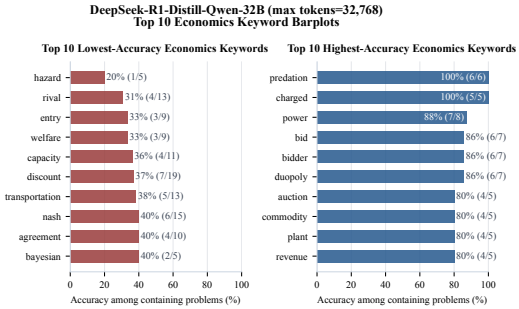

Figure 24: Top-10 word-level accuracy keyword barplot for DeepSeek-R1-Distill-Qwen-32B.

D Problem Type Definitions and Anchor Questions

Figure 25 provides the within-family breakdown underlying the main-text comparison. Across most families, the most visible drop occurs in the transition from *Conceptual Econ Math* to *Model Based Medium*, suggesting that the main point of difficulty is the shift from relatively direct calculation with economic context to problems that require explicit economic formalization and sustained structured reasoning.

The examples below are representative questions for the problem type taxonomy. Each one illustrates a representative instance of its category.

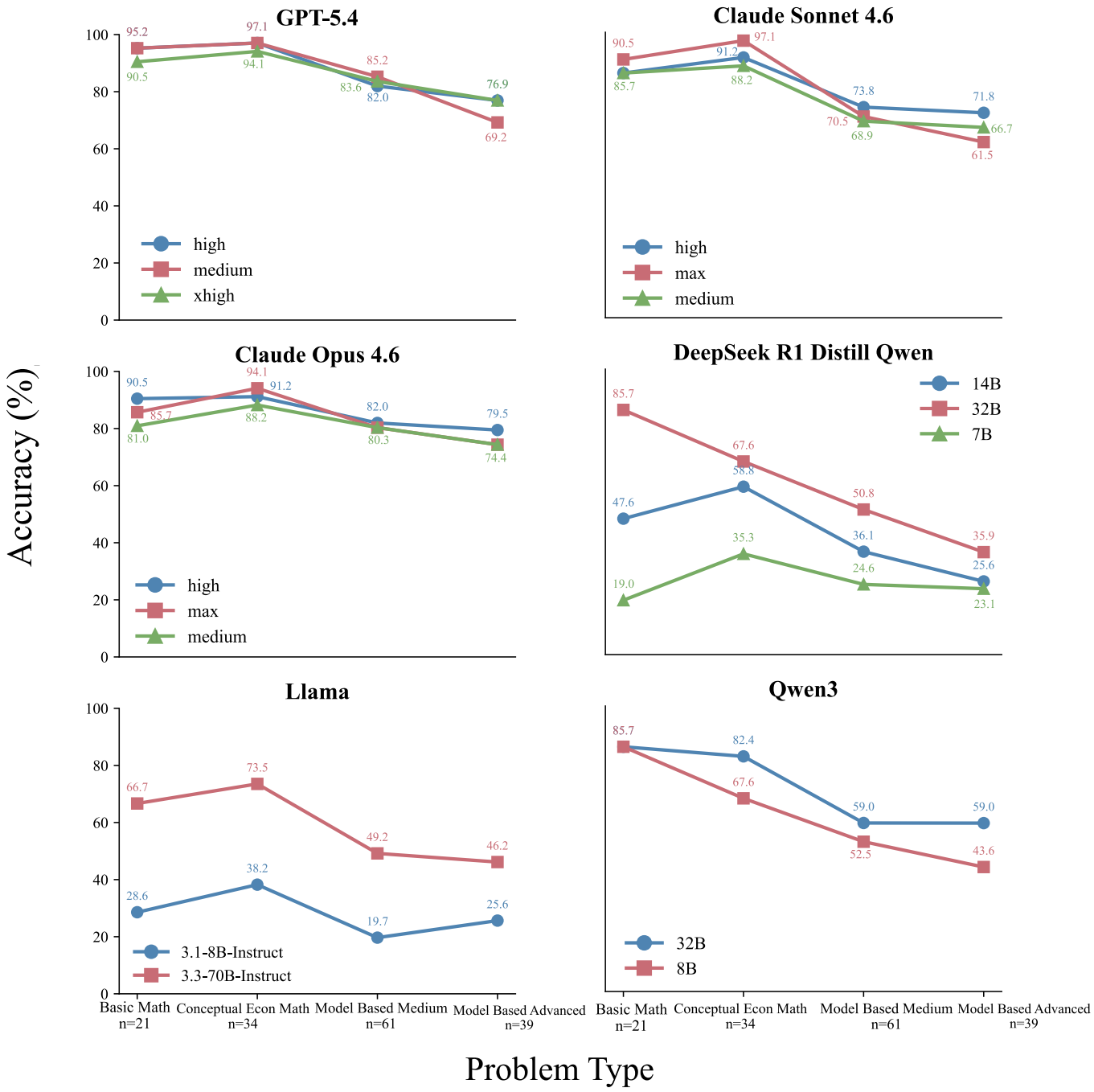

Figure 25: Rule-based accuracy by problem type across evaluated configurations within each model family. Questions are grouped into four manually validated categories: *Basic Math*, *Conceptual Econ Math*, *Model Based Medium*, and *Model Based Advanced*. Category labels were assigned through a multi-stage manual annotation process conducted by four PhD students and two professors. The x-axis reports the four problem-type categories, with n indicating the number of questions in each category after restricting to the common set of benchmark items evaluated across all plotted models. Each panel corresponds to a model family, and each line within a panel denotes one evaluated configuration in that family.

1.3/i#2: Basic Math

Problem: A monopolist's marginal cost of supplying a good to consumers is $\bar{c} = c + t$ (where t is a unit commodity tax). Let $p^m(\bar{c})$ denote the corresponding monopoly price.
(i) Compute $\frac{dp^m}{d\bar{c}}$ for the following demand function: $p = \alpha - \beta q^\delta$.
Symbols (for final answer):
δ: demand curvature parameter in $p = \alpha - \beta q^\delta$

Solution: For a monopolist with constant marginal cost $\bar{c}$, the FOC is

$$p(q) + qp'(q) = \bar{c}.$$

Differentiate implicitly w.r.t. $\bar{c}$:

$$(2p'(q) + qp''(q))\,\frac{dq^m}{d\bar{c}} = 1, \qquad \frac{dp^m}{d\bar{c}} = p'(q^m)\frac{dq^m}{d\bar{c}} = \frac{p'(q^m)}{2p'(q^m) + q^m p''(q^m)}.$$

Apply to the demand: $p = \alpha - \beta q^\delta$ ($\beta > 0, \delta > 0$):

$$p'(q) = -\beta\delta q^{\delta-1}, \quad p''(q) = -\beta\delta(\delta-1)q^{\delta-2}$$

so

$$\frac{dp^m}{d\bar{c}} = \frac{1}{\delta+1}.$$

Answer: $\frac{1}{\delta+1}$

3.1/i: Conceptual Econ Math

Problem: A monopolist serves n identical consumers. The aggregate inverse demand is $P(Q)$, where Q denotes the total quantity sold across all consumers and $P' < 0$. Marginal cost is a constant c. The monopolist can perfectly price discriminate by offering each consumer a nonlinear tariff $T(q)$, where q is the individual quantity consumed. Consumers can refuse the tariff (outside option $= 0$). Compute the nonlinear tariff $T(q)$ that extracts each consumer's gross surplus. Express the tariff in terms of the aggregate quantity $Q = nq$, $P(\cdot)$, and n.

Solution: Let q be each consumer's quantity, so aggregate quantity is $Q = nq$. Since aggregate inverse demand is $P(Q)$, the marginal willingness to pay of an individual consumer at quantity s is $P(ns)$. Hence the gross surplus of one consumer is

$$S(q) = \int_0^q P(ns)\, ds.$$

Using the change of variable $u = ns$,

$$S(q) = \frac{1}{n} \int_0^{nq} P(u)\, du.$$

Equivalently, since $Q = nq$,

$$S\left(\frac{Q}{n}\right) = \frac{1}{n} \int_0^{Q} P(u)\, du.$$

To extract each consumer's gross surplus, the monopolist sets

$$T(q) = S(q) = \int_0^q P(ns)\, ds = \frac{1}{n} \int_0^{nq} P(u)\, du.$$

Thus each consumer's net surplus is

$$S(q) - T(q) = 0,$$

so the participation constraint binds. The monopolist's profit from inducing aggregate quantity Q is

$$\Pi(Q) = nT\left(\frac{Q}{n}\right) - cQ = \int_0^Q P(u)\, du - cQ.$$

By substituting the tariff. The first-order condition is

$$P(Q) - c = 0.$$

Therefore the surplus-maximizing aggregate quantity Q^* satisfies $P(Q^*) = c$, and each consumer consumes $q^* = \frac{Q^*}{n}$. Thus the tariff can be written as

$$T\left(\frac{Q}{n}\right) = \frac{1}{n} \int_0^Q P(u)\, du.$$

Answer: $\frac{1}{n} \int_0^Q P(u)\, du$

1.10/i: Model-Based Medium

Problem: Consider the previous two-period model (in which the good becomes obsolete after two periods). Introduce a constant unit-production cost of $c_1(x)$ in the first period and c_2 in the second period. x is the probability that the first-period good is still usable in the second period. Thus, if q_1 is the first-period production and xq_1 is the number of units of the durable good still around at date 2 before further production by the monopolist. (Assume $c_1' > 0$, $c_1'' > 0$, and $c_1'(0)$ "small" and $c_1'(1)$ "large," so as to obtain an interior solution.)
(i) If the monopolist can commit, what is the value of $c_1'(x)$ at the optimal (interior) durability choice, expressed in terms of δ and c_2?
Symbols (for final answer):
- c_2: constant unit production cost in period 2
- δ: discount factor

Solution: (i) With commitment, the monopolist chooses

$$\max_{q_1,q_2,x} \pi = p_1(q_1)q_1 - c_1(x)q_1 + \delta\big(p_2(q_2 + xq_1)q_2 - c_2q_2\big),$$

where in period 2 the relevant state is the inherited stock xq_1. For any given q_1, changing x affects profits only through production costs: increasing x raises first-period unit cost by $c_1'(x)$ on each of the q_1 units, but saves discounted second-period unit production cost δc_2 on each of the q_1 units that survives (since one more surviving unit reduces optimal new production one-for-one under commitment). Thus the FOC for an interior optimum is

$$-c_1'(x)q_1 + \delta c_2 q_1 = 0 \quad \Rightarrow \quad c_1'(x) = \delta c_2,$$

which is the cost-minimizing durability condition.

Answer: δc_2

6.8/1#2: Model-Based Advanced

Problem: Work your way through the Green-Porter model with price competition and two i.i.d. states of demand (no demand or positive demand). Rederive the optimal length of punishment. Show that for a probability $\alpha = \frac{1}{4}$ of no demand, δ must exceed $\frac{2}{3}$ in order for collusion to be sustainable; show also that at least two periods of punishment are required. Consider the following infinite-horizon supergame. Two firms choose prices each period. The goods are perfect substitutes and are produced at constant marginal cost c, so consumers all buy from the low-price firm; if both firms charge the same price, demand is split equally. In each period there are two possible states of demand: with probability α there is no demand (the "low-demand state"), and with probability $1-\alpha$ there is positive demand $D(p)$ (the "high-demand state"). Demand realizations are i.i.d. over time. Let p^m denote the monopoly price for unit cost c (i.e. p^m maximizes $(p-c)D(p)$), and let Π^m denote the monopoly profit in the high-demand state. Firms do not observe rivals' prices directly, but it is always common knowledge that at least one firm makes no profit in each period. A firm that does not sell cannot tell whether this is due to the low-demand state or to its rival having charged a lower price. Analyze the following (pure) strategies, which feature a collusive phase and a punishment phase. The game begins in the collusive phase. In the collusive phase, both firms charge p^m until a period in which at least one firm makes zero profit; the occurrence of a zero-profit event is observed by both firms and triggers a punishment phase. In the punishment phase, both firms charge the competitive (Bertrand) price c for exactly T periods (where T is a nonnegative integer, possibly $+\infty$), and then revert to the collusive phase. Let V^+ (respectively, V^-) denote the present discounted value of a firm's profits from some date t onward, assuming that at date t the game is in the collusive phase (respectively, starts the punishment phase). By stationarity, V^+ and V^- do not depend on time. By definition, these values satisfy

$$V^+ = (1-\alpha)\Big(\Pi^m/2 + \delta V^+\Big) + \alpha\big(\delta V^-\big), \tag{1}$$

$$V^- = \delta^T V^+. \tag{2}$$

In the collusive phase, an undercutting deviation (charging below p^m) yields current profit Π^m if demand is high and triggers the punishment phase next period; this gives the incentive constraint

$$\delta\big(V^+ - V^-\big) \geq \Pi^m/2. \tag{3}$$

Equations imply the sustainability condition

$$1 \leq 2(1-\alpha)\delta + (2\alpha - 1)\delta^{T+1}. \tag{4}$$

For $\alpha = \frac{1}{4}$, compute the threshold value of δ (as a number) such that collusion can be sustainable for some (possibly infinite) T.

6.8/1#2: Model-Based Advanced

Solution: The firms choose an integer punishment length $T \in \{0, 1, 2, \ldots, \infty\}$ to maximize the collusive-phase value

$$V^+(T) = \frac{(1-\alpha)\Pi^m/2}{1-(1-\alpha)\delta - \alpha\delta^{T+1}}$$

subject to the incentive constraint

$$1 \leq 2(1-\alpha)\delta + (2\alpha - 1)\delta^{T+1}. \tag{6.16}$$

Since $\alpha < 1/2$ (in particular $\alpha = 1/4$), we have $2\alpha - 1 < 0$, so the RHS of (6.16) is increasing in T (because δ^{T+1} decreases with T and it is multiplied by a negative coefficient). Also, for fixed δ, $V^+(T)$ is decreasing in T because the denominator $1-(1-\alpha)\delta - \alpha\delta^{T+1}$ increases as T increases. Therefore the optimal punishment length is the smallest T that satisfies (6.16). For $\alpha = \frac{1}{4}$, (6.16) becomes

$$1 \leq \frac{3}{2}\delta - \frac{1}{2}\delta^{T+1} \quad \Longleftrightarrow \quad 3\delta - \delta^{T+1} \geq 2. \tag{*}$$

Existence of some finite T satisfying (*) requires

$$\lim_{T\to\infty} \left(3\delta - \delta^{T+1}\right) = 3\delta \geq 2 \quad \Longleftrightarrow \quad \delta \geq \frac{2}{3}.$$

At $\delta = \frac{2}{3}$, only $T = \infty$ satisfies (*); for $\delta > \frac{2}{3}$ there is a finite minimal T. To show at least two periods of punishment are required (for $\alpha = \frac{1}{4}$): with $T = 1$, (*) is

$$3\delta - \delta^2 \geq 2 \quad \Longleftrightarrow \quad \delta^2 - 3\delta + 2 \leq 0 \quad \Longleftrightarrow \quad (\delta - 1)(\delta - 2) \leq 0,$$

which holds only at $\delta = 1$ within $[0, 1]$. Hence for any $\delta < 1$, $T = 1$ cannot sustain collusion, so the minimal feasible punishment length satisfies $T \geq 2$ (whenever collusion is sustainable, i.e. $\delta > 2/3$).

Answer: $\frac{2}{3}$

E Model generation Parameters

Table 10 summarizes the generation settings used for the solver configurations in Section 4.1.

Table 10: Generation parameters for closed-source and open-weight LLMs.

Model group	Models	Output budget	Reasoning/decoding setting
Closed-source	`GPT-5.4`	65,536 tokens	Reasoning effort in {`medium`, `high`, `xhigh`}.
Closed-source	`Claude Opus 4.6`; `Claude Sonnet 4.6`	65,536 tokens	Reasoning effort in {`medium`, `high`, `max`}.
Open-weight instruction-tuned	`Llama 3.1 8B Instruct`; `Llama 3.3 70B Instruct`	32,768 tokens	vLLM deterministic sampling: temperature 0.0, top-p 1.0, repetition penalty 1.0.
Open-weight reasoning	`Qwen3-8B`; `Qwen3-32B`	32,768 tokens	Reasoning mode on.
Open-weight reasoning	`DeepSeek-R1-Distill-Qwen-7B`; `DeepSeek-R1-Distill-Qwen-14B`; `DeepSeek-R1-Distill-Qwen-32B`	32,768 tokens	Reasoning model.

F Human Annotation Instructions

This appendix provides the instructions given to expert annotators during the benchmark verification stage.

Expert Verification System Prompt

You are an expert annotator verifying candidate benchmark items converted from textbook exercises in industrial organization.
Context: The source problem and source/reference answer shown to you are not necessarily raw textbook text copied directly from the book. They are the structured records used as inputs to the LLM-assisted construction pipeline. These records were obtained after extracting exercises and reference answers from textbook pages, matching them by exercise number, and, when needed, splitting multi-part exercises into atomic source question–answer records while preserving the necessary shared context.
For each candidate item, your task is to check the validity of the final evaluation item. You are not asked to grade or rewrite the full solution process.
You will be provided with:

- the source problem used as input to the LLM-assisted pipeline;
- the source/reference answer or hint used as input to the LLM-assisted pipeline, if available;
- the LLM-converted benchmark question;
- the LLM-generated or LLM-converted solution; and
- the LLM-converted canonical final answer.

The source problem and source/reference answer are provided as pre-conversion reference context. Use them to judge whether the converted benchmark question preserves the intended economic content of the source item.
The provided solution is included only as supporting context. You may use it to understand how the candidate final answer was obtained, but you do not need to verify every intermediate derivation step, proof step, or explanatory detail. Your annotation should focus on the converted benchmark question and its canonical final answer.
Please check the following two properties:

1. **Self-contained question.** Determine whether the converted benchmark question contains all information needed to solve the problem without relying on omitted textbook context. In particular, check whether the assumptions, notation, variables, and requested target are sufficiently specified.
2. **Correct and unique final answer.** Determine whether the canonical final answer is correct and uniquely determined by the converted benchmark question under the stated assumptions and notation. For symbolic-answer items, check whether the final answer represents the intended mathematical object, such as a closed-form expression, threshold, equilibrium quantity, or comparative-static value. For binary-judgment items, check whether the True/False label is unambiguously implied by the converted question.

After reviewing the item, decide whether the converted question and canonical final answer should be kept unchanged or overwritten.

- If the converted question is self-contained, faithful to the source item, and should be kept unchanged, leave the `question` field blank.
- If the converted question is missing necessary information, ambiguous, not faithful to the source item, or otherwise ill-posed, fill in the `question` field with the revised question that should overwrite the converted benchmark question.
- If the canonical final answer is correct, uniquely determined by the converted question, and should be kept unchanged, leave the `answer` field blank.
- If the canonical final answer is incorrect, ambiguous, incomplete, or not uniquely determined by the converted question, fill in the `answer` field with the revised canonical final answer that should overwrite the converted answer.

- If both the question and the answer need to be changed, fill in both fields.

When revising the question, preserve the intended economic content of the source item as much as possible. Prefer minimal edits, such as clarifying notation, restoring a missing assumption, or narrowing the requested target. Do not substantially change the intended problem unless this is necessary to make the item self-contained and verifiable.
When revising the answer, provide only the canonical final answer that should be used for evaluation. You do not need to provide a full derivation unless it is necessary to explain why the original final answer is incorrect or ambiguous.
Return your annotation using the following fields:
question: Leave blank if the converted question should be kept unchanged. Otherwise, enter the revised question that should overwrite it.
answer: Leave blank if the canonical final answer should be kept unchanged. Otherwise, enter the revised final answer that should overwrite it.
notes: Optional. Briefly explain any issue, especially if you revise the question or answer.

Expert Verification User Prompt Template

Source problem used as input to the LLM-assisted pipeline:
{source_problem}
Source/reference answer or hint used as input to the LLM-assisted pipeline:
{source_answer}
LLM-converted benchmark question:
{converted_question}
LLM-generated or LLM-converted solution:
{converted_solution}
LLM-converted canonical final answer:
{converted_answer}
Please verify whether the converted benchmark question is self-contained and faithful to the source item, and whether the canonical final answer is correct and uniquely determined.
If the converted question should be kept unchanged, leave the `question` field blank. If it should be overwritten, fill in the `question` field with the revised question.
If the canonical final answer should be kept unchanged, leave the `answer` field blank. If it should be overwritten, fill in the `answer` field with the revised final answer.
Use the following annotation fields:
question:
answer:
notes:

G Data Screening for Personally Identifying Information and Offensive Content

IO-BENCH is derived from textbook exercises and expert-written annotations rather than personal, conversational, social-media, or user-generated text. To assess whether the released benchmark contains personally identifying information or offensive content, we screened the released questions, canonical answers, solutions, and metadata during dataset construction. The screening checked for names or identifiers that could uniquely identify individual people, including student names, email addresses, institutional identifiers, phone numbers, addresses, and other personal records, as well as offensive or inappropriate content.

We did not find personally identifying information or offensive content in the released benchmark items. Because the benchmark content is based on industrial-organization textbook exercises and expert-verified mathematical/economic annotations, no additional anonymization of benchmark items was required. The PhD-level human-reference comparison is reported only through aggregate, de-identified historical assessment statistics. We do not release student names, identifiers, individual score records, submissions, or any other personally identifying student information.

H Artifact Use, Licenses, and Intended Use

This work uses and creates scientific artifacts for research purposes. The main source artifact is Tirole's *The Theory of Industrial Organization* (Tirole, 1988), a commercially published textbook. We use the textbook as a source for constructing an industrial-organization reasoning benchmark, but we do not release the original textbook exercises, original reference answers, solution-manual text, page images, or long continuous passages from the source material. The released artifact consists of benchmark-specific items that have been rewritten, completed, decomposed, and verified by domain experts, together with canonical reference answers, annotations, evaluation metadata, and rule-based evaluation code. The release is intended to support research evaluation of model-grounded economic reasoning, not to reproduce, replace, or redistribute the original textbook or its instructional materials.

We designed the release to align with the fair-use doctrine (17 U.S.C. § 107). The purpose of the use is noncommercial research evaluation, and the released benchmark serves a different function from the source textbook by converting selected IO reasoning targets into self-contained, verifiable symbolic-answer and binary-judgment tasks. The released items are substantially transformed from the original textbook format through expert rewriting, completion of implicit assumptions, decomposition into atomic evaluation targets, and conversion into machine-verifiable answer formats. Therefore the released benchmark is transformative. We also limit the amount of source material exposed in the released artifact: original textbook questions, original solutions, page images, and extended source passages are not distributed. Finally, the benchmark is not intended to substitute for the original textbook, reproduce its pedagogical sequence, or provide access to the original exercise and solution collection.

We also use existing software and model artifacts, including symbolic evaluation tools such as `SymPy` (Meurer et al., 2017), closed-source model APIs, and open-weight model families. These artifacts are cited in the relevant sections and are used under their respective licenses, access terms, or usage policies. The artifacts created by this work, including IO-BENCH and the associated evaluation code, will be released for research use with accompanying documentation describing their intended use, scope, and access conditions. IO-BENCH should not be used as a substitute for the source textbook, for commercial redistribution of textbook-derived content, or for applications inconsistent with the access conditions of the source materials.