FINE-TUNED JUDGMENT LIVES IN THE WEIGHTS: IN-CONTEXT EXAMPLES HELP UNTRAINED MODELS AND OVERRIDE TRAINED ONES

Anonymous authors
Paper under double-blind review

ABSTRACT

In many professional judgment tasks, answers are hidden, deferred, or counterfactual, so no automated verifier exists. We study such tasks using an environment that plants the truth and holds the answer key. We ask where a model's judgment lives: in its context or in its weights. The environment generates companies with customer cohort-level financials. Unhealthy companies hide a revenue contraction in one cohort, and most healthy companies carry a decoy cohort that looks like a contraction and is not. We evaluate an untrained Qwen 7B, Claude Fable 5 and Claude Sonnet 4.6 as of August 2026, and the same 7B fine-tuned on its own attempts filtered by the answer key, on 550 companies, 150 unhealthy and 400 healthy, under one parser and one grader. Each model runs zero-shot and with 2 in-context examples: one company solved with reasoning and a verdict, shown once with the planted flaw and once without. Fable 5 refused to emit verdicts whenever examples were present, across 3 prompt formats, leaving 7 conditions. In the zero-shot condition, every untrained model detects nearly every hidden contraction but produces false positives on most healthy companies. Fable 5 detects 0.987 with a false-positive rate of 0.812, Sonnet 4.6 detects 1.000 at 0.777, and the untrained 7B detects 0.913 at 0.833. Sampled transcripts show both frontier models compute that the decoy's revenue is intact, state it, and call the company unhealthy anyway, a failure of judgment rather than analysis. The 2 in-context examples mainly cut false positives, taking the untrained 7B to 0.807 at 0.302 and Sonnet 4.6 to 1.000 at 0.092. Fine-tuning moves the 7B to 0.960 at 0.247 zero-shot, higher detection than its untrained base with less than a third of its false positives. The same 2 examples collapse it to 0.187 at 0.030. It repeats the healthy example's reasoning on unhealthy companies. In-context examples provide judgment reasoning to models that lack it and override the model that has it. We take the reversal as evidence that the fine-tuned model's judgment lives in its weights, not its context. Limitations: one environment and one planted flaw type. A fine-tuned Mistral 7B shows the same override with the collapse in false positives rather than in detection.

1 INTRODUCTION

Reinforcement learning from verifiable rewards (Lambert et al., 2024; DeepSeek-AI, 2025) works where an answer can be checked automatically. A proof verifies or it does not. Code passes its tests or it does not. Most professional judgment tasks are not like this. A person judges whether a business is quietly deteriorating while topline revenue looks healthy. The question has a correct answer, but the answer is costly to discover, surfaces late, or never surfaces at all. No automated verifier exists, so the field falls back on preference data (Christiano et al., 2017; Ouyang et al., 2022) or rubric scoring (Gunjal et al., 2025), neither of which holds an answer key.

We study a judgment task where an answer key does exist. We use a synthetic environment that plants the truth and holds the answer key, so that any attempt can be graded exactly. The environment generates companies that report customer counts and revenue by quarterly cohort. Unhealthy companies hide a revenue contraction in one cohort. Most healthy companies carry a decoy cohort

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

that looks like a contraction and is not. The correct method checks both retention measures per cohort, names the contraction, and does not name the decoy.

Against this environment we ask a simple question. Where does a model's judgment live, in its context or in its weights? We evaluate 4 models under 2 prompting conditions: 2 frontier models (Claude Fable 5 and Claude Sonnet 4.6, August 2026), an untrained open-weight model (Qwen2.5-7B-Instruct), and the same Qwen model fine-tuned on 28 of its own attempts filtered by the answer key. Every cell shares one fresh test set of 550 companies, one parser, and one grader. The conditions are zero-shot and 2 in-context examples in the prompt: one company solved with reasoning and a verdict, shown once with the planted flaw and once without. Fable 5 refused to emit verdicts whenever examples were present, across 3 prompt formats, so its in-context-example condition could not be run and 7 conditions remain.

Four findings follow.

Zero-shot, detection saturates and discrimination is absent in every untrained model. Every untrained model detects nearly every hidden contraction and calls most healthy companies unhealthy. The untrained Qwen model detects 0.913 at a 0.833 false-positive rate, Fable 5 detects 0.987 at 0.812, and Sonnet 4.6 detects 1.000 at 0.777. Calling every company unhealthy scores $J = 0$, where J is the detection rate minus the false-positive rate, and all 3 untrained models have J below 0.25. The frontier models improve on the untrained Qwen model in detection and identification of the right cohort, not in discrimination. In a sample of false positives we read by hand, both frontier models compute that the decoy cohort's revenue is intact, state it, and call the company unhealthy anyway. The verdicts contradict the evidence the models themselves state, and contradict the instruction's own framing of the problem as a revenue problem, since the models mark revenue retention as high, which makes this a judgment failure rather than a disagreement about grading standards.

In-context examples teach untrained models. On the same model, same instances, and same grader, 2 in-context examples move Sonnet 4.6 from $J = 0.223$ to $J = 0.907$, a gain of 0.68, and move the untrained Qwen model from $J = 0.081$ to $J = 0.504$, a gain of 0.42. The in-context examples mainly reduce false positives.

Fine-tuning produces zero-shot judgment. The fine-tuned Qwen model is the only model with high zero-shot discrimination: 0.960 detection at 0.247 false positives, $J = 0.712$, on the same companies where every untrained model calls 78 to 83 percent of healthy companies unhealthy. Its J is more than triple the best untrained zero-shot J, from fine-tuning on 28 graded attempts.

The same in-context examples help untrained models and override the trained one. With the 2 in-context examples in the prompt, the fine-tuned Qwen model collapses from 0.960 detection at 0.247 false positives to 0.187 detection at 0.030 false positives, and its J drops from 0.712 to 0.157, a loss of 0.56. In a sample of the 122 missed unhealthy companies that we read by hand, the model repeats the healthy in-context example's reasoning nearly word for word, and in one case states in its own reasoning that a hidden contraction exists and still concludes that the company is healthy. Two runs produced byte-identical outputs under greedy decoding. A pre-registered run on a second architecture, a fine-tuned Mistral model, shows the same override in the opposite direction, though not by the pre-registered pattern: both fine-tuned seeds degrade under the in-context examples by repeating the unhealthy in-context example's verdict on healthy companies, collapsing in false positives rather than in detection (Section 6).

In-context examples provide judgment reasoning to models that lack it and override the model that has it. We take the reversal as evidence that the fine-tuned model's judgment lives in its weights, not its context, and that a fine-tuned model treats in-context examples as verdicts to repeat rather than as a method to apply. The convention of adding examples by default, standard since Brown et al. (2020), measures untrained models at their best and trained models, meaning models fine-tuned on the task, at their worst. A trained model should be evaluated in the prompt format it was trained on.

2 Related Work

Verifiable rewards and judgment tasks. Verifiable-reward training (Lambert et al., 2024; DeepSeek-AI, 2025) requires a verifier, and many judgment tasks lack one. The alternative training

methods learn from preferences without ground truth (Christiano et al., 2017; Ouyang et al., 2022) or score attempts against rubrics of desirable properties (Gunjal et al., 2025), grading how the answer looks rather than whether the answer is right. LLM-as-a-judge evaluation (Zheng et al., 2023) likewise substitutes another model's opinion for an answer key. Public real-world benchmarks hold answers but not planted ones, and carry data contamination risk. FinanceBench (Islam et al., 2023) is the nearest example in our task's domain. We use an environment that plants the truth and holds the answer key, so grading is exact for every attempt. The environment is the setting of this paper, not its contribution. Our contributions are the frontier evaluation, the in-context-example interaction, and the inference that fine-tuned judgment lives in the weights.

What in-context examples do. In-context examples can act more as signals of format, label space, and task identity than as ground-truth supervision (Min et al., 2022). They induce majority-label and recency biases that pull predictions toward the labels shown in the in-context examples (Zhao et al., 2021; Lu et al., 2022). Larger models can be made to override their semantic priors and follow in-context labels (Wei et al., 2023). In-context learning has been tied to pattern copying from the prompt (Olsson et al., 2022) and to a compact internal representation of the task (Hendel et al., 2023). These results come from models without task-specific fine-tuning, where following the in-context examples is the desired behavior. Our finding is the interaction with task-specific fine-tuning: the same in-context examples that build an untrained model's judgment override the judgment that lives in a fine-tuned model's weights, because the fine-tuned model repeats the healthy in-context example's verdict.

Fine-tuning versus in-context learning. Prior comparisons treat fine-tuning and in-context learning as alternative routes to the same capability and measure them on separate models (Mosbach et al., 2023). We cross them on one model and find they interfere. The collapse is also distinct from catastrophic forgetting (McCloskey & Cohen, 1989): the weights do not change after training, the override is induced at inference by the prompt alone, and removing the in-context examples restores the trained behavior.

3 The Environment and the Task

Companies and cohorts. The environment generates synthetic software-subscription companies. Each company reports a table with one row per quarterly cohort, carrying customer counts and revenue at the start and now. Each cohort therefore has 2 retention measures, one for customers and one for revenue. Each company is either healthy or unhealthy.

The planted flaw. Unhealthy companies hide a revenue contraction in one cohort: the cohort keeps its customers while its revenue falls. Most healthy companies, 318 of the 400 in the test set, carry a decoy cohort that loses many customers while maintaining its revenue. The correct method checks both retention measures for every cohort, names the contraction cohort, and does not name the decoy.

Answer key and grading. The generator writes each company's answer key along with the company, and the model never sees the answer key. A script, not a model, grades each verdict against the answer key, and its parser accepts only the required verdict format (Appendix A). Each verdict is graded on whether it calls the company healthy or unhealthy correctly. When the verdict claims a revenue contraction, it is also graded on whether the named cohort matches the answer key. A response the parser rejects, including a refusal, is a format failure. A format failure counts as a miss on an unhealthy company and as a healthy call on a healthy company. Format failures are rare and move no conclusion. Fable's false-positive rate on the 393 healthy companies it answered is 0.827 against the reported 0.812, and every other condition in Table 1 has 2 or fewer format failures.

Metrics. We report the detection rate on the 150 unhealthy companies together with the false-positive rate on the 400 healthy companies, never detection alone. We summarize the pair with J, the detection rate minus the false-positive rate, which is Youden's J statistic (Youden, 1950). A constant alarm, calling every company unhealthy, scores 1.000 detection at 1.000 false positives, so $J = 0$. Site accuracy is the share of correctly detected unhealthy companies whose named cohort matches the answer key. We report Wilson 95 percent intervals (Wilson, 1927). Comparisons between prompt conditions on the same companies use exact McNemar tests (McNemar, 1947).

Instances. The test set is 550 fresh companies, 150 unhealthy and 400 healthy, generated for this evaluation from 2 reserved seed ranges and never reused afterward. All 7 conditions in Table 1 use these 550 companies and the same grading script.

4 EXPERIMENTAL SETUP

Models. We evaluate Claude Fable 5 and Claude Sonnet 4.6 through their public API (Anthropic, 2026) as of August 2026, an untrained Qwen2.5-7B-Instruct (Qwen Team, 2025), and a fine-tuned copy of that Qwen model, described below. Section 6 adds Mistral-7B-Instruct-v0.3 (Jiang et al., 2023) as a replication, both untrained and fine-tuned, the latter on the same 28 attempts.

Fine-tuning. Both fine-tuned models are LoRA adapters (Hu et al., 2022) trained with the same recipe on the same 28 attempts, which the untrained Qwen model made on earlier companies from the same generator, 14 unhealthy and 14 healthy. An attempt was kept only when its verdict matched the answer key, and for an unhealthy company only when the named cohort also matched. Training uses LoRA rank 16, alpha 32, dropout 0.05, all-linear target modules, learning rate 2×10^{-4}, batch size 1 with gradient accumulation 8 (40 optimizer steps over 10 epochs), completion-only loss, a 1,024-token sequence cap, and 4-bit NormalFloat quantization (Dettmers et al., 2023). The protocol that collected and filtered these attempts is described in a separate manuscript under review. Every training setting appears here. The 28 training attempts are held with that protocol.

Conditions. The zero-shot condition gives the model an instruction and the company's table, and nothing else. The instruction states the task, asks the model to show its reasoning, and requires a single structured verdict line at the end (Appendix A). The zero-shot condition therefore already elicits explicit reasoning before the verdict. The in-context-example condition places 2 in-context examples between the instruction and the company's table: one company solved with reasoning and a verdict, shown first with the planted flaw and then without. The company is generated from a seed outside the training and test ranges. The in-context-example condition is few-shot prompting (Brown et al., 2020) with chain-of-thought examples (Wei et al., 2022). The same 2 in-context examples were used for every model (Appendix A).

Decoding. Each frontier model ran a single pass with the API's default sampling and an output limit of 4,096 tokens. The Fable API rejects an explicit temperature parameter, so we omitted it for Sonnet as well, keeping both models under the same settings. We ran the Qwen and Mistral models on Google Colab with greedy decoding and a limit of 1,280 new tokens.

The Fable refusal and the 7 conditions. Fable 5 refused to give a verdict whenever in-context examples were present. We established this before the counted runs, in 6 test calls on companies outside the test set, across 3 ways of formatting the in-context examples in the prompt. Fable's in-context-example condition therefore could not be run, which leaves 7 of the 8 conditions. In the zero-shot condition, Fable refused on 9 of the 550 companies, 2 unhealthy and 7 healthy, and these refusals count as format failures. The cause is not observable from the API. The pattern is consistent with the model declining to make a negative claim about the company under review, and we claim no more than that.

5 RESULTS

Table 1 reports all 7 conditions on the shared test set.

5.1 ZERO-SHOT: DETECTION SATURATES, DISCRIMINATION IS ABSENT

In the zero-shot condition, every untrained model detects nearly every hidden contraction and calls most healthy companies unhealthy. The untrained Qwen model detects 0.913 at a 0.833 false-positive rate, Fable 5 detects 0.987 at 0.812, and Sonnet 4.6 detects 1.000 at 0.777. All 3 untrained models have J below 0.25, close to the $J = 0$ of calling every company unhealthy. The 2 frontier models reach nearly identical rates. Fable on 316 and Sonnet on 307 of the 318 healthy companies that carry a decoy name the decoy as the contraction cohort. The pattern is therefore a property of untrained zero-shot judgment on this task, not a quirk of one model.

Table 1: All 7 conditions, one test set of 550 companies (150 unhealthy, 400 healthy), one grading script. Detection is the detection rate, the share of unhealthy companies called unhealthy. FP is the false-positive rate, the share of healthy companies called unhealthy. Both carry Wilson 95 percent intervals. J is detection minus FP, computed from the exact counts before rounding. Site is site accuracy. Fmt is the count of format failures, which includes the 9 Fable refusals. The Mistral replication is in Table 2.

Model	Prompt	Detection (95% CI)	FP (95% CI)	J	Site	Fmt
Untrained Qwen	zero-shot	0.913 (0.857, 0.949)	0.833 (0.793, 0.866)	0.081	0.474	0
Untrained Qwen	2 examples	0.807 (0.736, 0.862)	0.302 (0.260, 0.349)	0.504	0.810	0
Fable 5	zero-shot	0.987 (0.953, 0.996)	0.812 (0.771, 0.848)	0.174	0.993	9
Sonnet 4.6	zero-shot	1.000 (0.975, 1.000)	0.777 (0.734, 0.816)	0.223	0.953	0
Sonnet 4.6	2 examples	1.000 (0.975, 1.000)	0.092 (0.068, 0.125)	0.907	1.000	0
Fine-tuned Qwen	zero-shot	0.960 (0.915, 0.982)	0.247 (0.208, 0.292)	0.712	0.979	2
Fine-tuned Qwen	2 examples	0.187 (0.132, 0.257)	0.030 (0.017, 0.052)	0.157	0.929	0

The frontier models improve on the untrained Qwen model in detection and identification of the right cohort, not in discrimination. Detection rises from 0.913 for the untrained Qwen model to 0.987 and 1.000 for the frontier models, and site accuracy from 0.474 to 0.993 and 0.953, while J stays below 0.25. On the 541 companies Fable answered, it caught 148 of 148 contractions with site accuracy 0.993 and called 325 of 393 healthy companies unhealthy. Fable identifies the contraction cohort almost perfectly and still calls 83 percent of the healthy companies it answered unhealthy.

The failure contradicts the instruction's own framing of the problem as a revenue problem, since the model itself marks revenue retention as high. It also happens in reasoning the model was explicitly asked to show before the verdict. In a sample of false positives we read by hand, both frontier models compute the decoy cohort's revenue as intact or growing, state this in their reasoning, mark revenue retention as high in their own verdict line, and still conclude that the company has a hidden revenue problem. In these decoy cases, the models state the evidence that the company is healthy and call it unhealthy anyway. These are judgment failures. The analysis is correct, and the instruction's own framing of the problem as a revenue problem rules out a disagreement about grading standards. In the same sample, on healthy companies with no decoy, Fable also reports contractions that are not there, treating ordinary variation between cohorts as a contraction.

5.2 In-Context Examples Teach Untrained Models

For Sonnet 4.6, the 2 in-context examples raise J by 0.68. Sonnet 4.6 rises from $J = 0.223$ zero-shot to $J = 0.907$ with the in-context examples, at 1.000 detection and 0.092 false positives. The in-context examples are the only difference between the 2 conditions. The untrained Qwen model rises from $J = 0.081$ to $J = 0.504$, at 0.807 detection and 0.302 false positives. In both cases, the in-context examples mainly reduce false positives, which is the side on which zero-shot judgment fails. Among Sonnet's 37 remaining false positives, the ones we read are healthy companies whose numbers are close to the cutoff for healthy.

5.3 Fine-Tuning Produces Zero-Shot Judgment

The fine-tuned Qwen model is the only model with high zero-shot discrimination. It detects 0.960 at a 0.247 false-positive rate, for $J = 0.712$, with site accuracy 0.979. On the same companies, every untrained model calls 78 to 83 percent of healthy companies unhealthy. Its J is more than triple the best untrained zero-shot J, Sonnet's 0.223. Compared with the untrained Qwen model, it has higher detection with less than a third of the false positives. Of its 99 false positives, 23 name the decoy cohort and 76 fall on healthy companies with no decoy. Those are 76 of the 82 healthy companies without a decoy, so the fine-tuned model's remaining errors concentrate on the same companies where Fable also reports contractions that are not there. This discrimination comes from fine-tuning on 28 graded attempts, and the model shows it with no in-context examples in the prompt.

5.4 The Same In-Context Examples Override the Fine-Tuned Model

The in-context examples that improve Sonnet and the untrained Qwen model collapse the fine-tuned Qwen model. With the 2 in-context examples in the prompt, the fine-tuned Qwen model falls from 0.960 detection at 0.247 false positives to 0.187 detection at 0.030 false positives. Its J drops from 0.712 to 0.157, below the untrained Qwen model's 0.504 with the same in-context examples.

We read by hand a sample of the 122 unhealthy companies the model missed. In this sample, the model repeats the healthy in-context example's reasoning nearly word for word on unhealthy companies. In one case, the model states in its own reasoning that a hidden contraction exists and still concludes that the company is healthy. It also repeats values from the in-context examples' verdict lines, such as the quarter of the largest cohort, on companies whose true values differ. The fine-tuned Qwen model repeats the healthy in-context example's verdict. On the contractions it still detects, site accuracy stays high at 0.929, so the in-context examples override the verdict, not the ability to identify the right cohort.

Two runs of this condition, one from a saved session and one from a fresh start, produced byte-identical outputs under greedy decoding. We treat the fine-tuned Qwen model's result with in-context examples as a measure of interference, not of what the model can do. The fine-tuned Qwen model was trained on zero-shot prompts, so we treat zero-shot as its primary condition.

6 Replication on Mistral

To test whether the override is specific to one fine-tuned model, we pre-registered a replication on a second architecture before grading any output. The second architecture is Mistral-7B-Instruct-v0.3, fine-tuned with the same recipe on the same 28 attempts as the Qwen model (Section 4), at 2 training seeds. All 6 passes used the same 550 companies, the same grading script, and, where present, the same 2 in-context examples. Every pass had 0 format failures. Table 2 reports all 6 passes: the untrained model and both fine-tuned seeds, each with and without the in-context examples.

Table 2: Mistral-7B-Instruct-v0.3 on the same 550 companies, the same 2 in-context examples, and the same grading script as Table 1, with Wilson 95 percent intervals over the same denominators (150 unhealthy, 400 healthy). Columns are as in Table 1.

Model	Prompt	Detection (95% CI)	FP (95% CI)	J	Site	Fmt
Untrained	zero-shot	1.000 (0.975, 1.000)	0.995 (0.982, 0.999)	0.005	0.273	0
Untrained	2 examples	0.960 (0.915, 0.982)	0.505 (0.456, 0.554)	0.455	0.604	0
Fine-tuned, seed 1	zero-shot	0.987 (0.953, 0.996)	0.588 (0.539, 0.635)	0.399	0.568	0
Fine-tuned, seed 1	2 examples	1.000 (0.975, 1.000)	1.000 (0.990, 1.000)	0.000	0.587	0
Fine-tuned, seed 2	zero-shot	0.967 (0.924, 0.986)	0.090 (0.066, 0.122)	0.877	0.903	0
Fine-tuned, seed 2	2 examples	0.993 (0.963, 0.999)	0.318 (0.274, 0.365)	0.676	0.805	0

In the zero-shot condition, the untrained Mistral model is nearly a constant alarm, calling 548 of 550 companies unhealthy, an extreme version of the pattern in the 3 untrained rows of Table 1. The 2 in-context examples raise its J by 0.450. They turn 196 of its false positives into correct healthy calls and create 0 new false positives (exact McNemar $p < 0.0001$), at a detection cost of 6 of 150 unhealthy companies ($p = 0.031$). The same 2 in-context examples degrade both fine-tuned seeds, in both cases by raising false positives. Seed 1 falls 0.399 in J to a constant alarm that calls all 550 companies unhealthy: 165 healthy companies change from a healthy verdict to unhealthy and 0 change the other way ($p < 0.0001$). Seed 2 falls 0.201 in J: 105 healthy companies change to unhealthy and 14 change to healthy ($p < 0.0001$). Detection changes on the fine-tuned seeds are not significant ($p = 0.500$ and $p = 0.219$).

The Mistral collapse runs in the opposite direction from the Qwen collapse. The fine-tuned Qwen model repeats the healthy in-context example's verdict on unhealthy companies and loses detection. Both fine-tuned Mistral seeds repeat the unhealthy in-context example's verdict on healthy companies and gain false positives. The unhealthy in-context example puts its contraction in one quarter and its largest cohort in another. With the in-context examples, 354 of seed 1's 550 verdicts and 153 of seed 2's 276 unhealthy verdicts name that same pair of quarters. For the largest cohort alone, 521

of seed 1's 550 and 276 of seed 2's 276 unhealthy verdicts name the example's quarter, although the true quarter varies across companies and both seeds answered this field correctly in the outputs we read. Identification of the right cohort survives the override. Seed 1's site accuracy with the in-context examples, 0.587, stays close to its own zero-shot 0.568 and far from the untrained Mistral model's 0.273. The fine-tuning is still in effect, and the in-context examples change the verdict. The title claim, that in-context examples help untrained models and override trained ones, holds on a second architecture, in the opposite direction. Our pre-registered criterion for replication was the Qwen pattern, detection falling while false positives hold, and neither seed showed it. We therefore claim that the override replicates, but not the pattern by which it occurs.

Mistral's seed spread is large: the same data and recipe produced zero-shot J of 0.399 and 0.877, a spread of 0.478. We report both seeds and drop neither. In the small samples of outputs we read, neither Mistral seed calculates retention in its visible reasoning. Zero-shot, both seeds repeat wording from their training attempts without citing any numbers, and with the in-context examples they repeat text from those examples. Seed 2 still reaches 0.967 detection at 0.090 false positives with 0.903 site accuracy, the strongest zero-shot result in this study. Seed 2's judgment appears in its verdicts but not in its visible reasoning, so this paper's claim that judgment lives in the weights is a claim about the verdict.

7 Discussion

Where judgment lives. If in-context examples and fine-tuning supplied the same kind of help, their effects would add up, and the fine-tuned models with in-context examples would score best in the study. Instead, the in-context examples help the models that lack the correct method and override the models that have it. This holds on 2 architectures, even though the 2 fine-tuned models repeat different in-context examples. Identification of the right cohort stays close to the fine-tuned model's zero-shot level for Qwen (0.979 to 0.929) and Mistral seed 1 (0.568 to 0.587), and falls further for Mistral seed 2 (0.903 to 0.805), so the in-context examples interfere with the verdict more than the analysis. Fine-tuned judgment lives in the weights, and a fine-tuned model treats in-context examples as verdicts to repeat.

Override or distribution shift? A fair alternative reading is that the collapse is ordinary brittleness. The fine-tuned models were trained only on zero-shot attempts, so a prompt with 2 in-context examples lies outside their training distribution in length and structure, and a narrowly trained model might simply fail there. Three observations limit this explanation but do not rule it out. First, the failure is not general degradation. The verdicts repeat specific values from the in-context examples, which random noise would not produce. Second, identification of the right cohort stays far above the untrained model's level on both architectures, so the fine-tuning is still in effect. Third, the 2 architectures repeat different in-context examples, Qwen the final, healthy one and Mistral the first, unhealthy one, so no single bias toward one position explains both. Swapping the order of the 2 in-context examples would test this directly and has not been run. The practical conclusion holds under either reading. For a model fine-tuned on one prompt format, adding in-context examples changes its results and must be measured, not assumed to be harmless. The next experiments are to train on a mix of zero-shot prompts and prompts with in-context examples, to see whether the interference disappears, and to test whether preference-trained or reinforcement-trained models (Rafailov et al., 2023) resist it.

Consequences for evaluation practice. Adding in-context examples by default measures untrained models at their best and fine-tuned models at their worst. In Table 1, this convention would rank the fine-tuned Qwen model last, at $J = 0.157$, even though its zero-shot J is more than triple the best untrained zero-shot J. Evaluations of fine-tuned models should report the prompt format the model was trained on as the primary result, and should measure the effect of added in-context examples separately.

What the environment resolves. This planted flaw is too easy to separate the strongest models. Detection is near 1.000 for both frontier models, and Sonnet with in-context examples reaches $J = 0.907$. At this difficulty, the task separates the Qwen and Mistral models, the frontier models without in-context examples, and Sonnet with them, but it cannot rank frontier models against each other. Ranking frontier models against each other would require a harder planted flaw, which we leave to future work.

8 LIMITATIONS

We test one environment with one planted flaw type. Both fine-tuned architectures are 7B models, so no claim extends to other model sizes or tasks. The fine-tuning is supervised LoRA on only 28 attempts, so the override claim applies to that setting. Preference-trained and reinforcement-trained models are untested. The 2 in-context examples are one fixed pair in one order, and we have not tested the reverse order. The 28 training attempts come from the untrained Qwen model and were reused unchanged for Mistral. Attempts written by people are not evaluated here. The frontier results come from single API runs with default sampling, while the Qwen and Mistral results use greedy decoding, so comparisons between the 2 groups mix decoding methods. Mistral's 2 seeds differ by 0.478 in zero-shot J under the same recipe. In the small samples of outputs we read, neither Mistral seed calculates retention in its visible reasoning, so the claim that judgment lives in the weights is about the verdict, not the reasoning text. Fable's in-context-example condition could not be run, leaving its interaction unmeasured.

ETHICS STATEMENT

No human subjects were involved, and all evaluated data are synthetic. The environment's answer keys and the planted-flaw generator are withheld because publishing an answer key destroys the assessment instrument, a structural property of evaluation against held answer keys. The methodology is described in full, and new models can be evaluated at any time because the generator produces fresh companies with answer keys on demand, so no answer keys need to be released. The supplementary material includes the 550 spent test instances with their answer keys, because those seeds were reserved for this evaluation and are now spent. The generator, and the answer key of any company outside that spent set, remain withheld. The Fable refusal is reported descriptively, with cause unobservable from the API, and we draw no conclusion about the deployed model's policies beyond the observed pattern.

REPRODUCIBILITY STATEMENT

All 7 primary conditions and all 6 Mistral passes share one test set and one grading script, and every condition with in-context examples uses the same 2 in-context examples (Sections 3 and 4). Fine-tuning configuration, decoding parameters, and prompt-condition construction appear in Section 4. The Mistral run was pre-registered before any output was graded, with gates, predictions, and decision rules fixed in advance, and a post-run commit diff verifies the pre-registration was not edited. Frontier results use the model versions available through the API in August 2026. Raw per-instance result files for every condition are retained, and every number in this paper rebuilds from them. Uncertainty and paired tests are specified in Section 3.

AI USE STATEMENT

In this work, we used generative AI tools for tasks in the required-disclosure categories: implementing methods (the evaluation harness, training, and grading code), generating the synthetic data set through the environment's generator code, providing feedback on experimental design and pre-registered protocols, and supporting the interpretation of results. We did not use generative AI tools to formulate mathematical claims or proofs, and those categories are not applicable to this work. Additionally, we used generative AI tools for recommended-disclosure tasks: drafting and editing parts of the paper, creating tables, and identifying and formatting references. We have reviewed all AI-assisted work: all code was verified against held answer keys and spot-checked by hand, every reported number was verified against raw per-instance result files, the statistical tests were recomputed independently, and all cited references were checked against their sources. Generative AI systems are also the objects of study in this paper, and their outputs are reported as experimental data, not as authored content. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Anthropic. Claude model documentation. `https://docs.claude.com`, 2026. Accessed Au-

gust 2026.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Bing Peng, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*, 2025.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022.

Pranab Islam, Anand Kannappan, Douwe Kiela, Rebecca Qian, Nino Scherrer, and Bertie Vidgen. FinanceBench: A new benchmark for financial question answering. *arXiv preprint arXiv:2311.11944*, 2023.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165, 1989.

Quinn McNemar. Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika*, 12(2):153–157, 1947.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

Marius Mosbach, Tiago Pimentel, Shauli Ravfogel, Dietrich Klakow, and Yanai Elazar. Few-shot fine-tuning vs. in-context learning: A fair comparison and evaluation. In *Findings of the Association for Computational Linguistics: ACL 2023*, 2023.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *Transformer Circuits Thread*, 2022.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023.

Edwin B Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

William J Youden. Index for rating diagnostic tests. *Cancer*, 3(1):32–35, 1950.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

A THE PROMPT, VERBATIM

Every condition in the paper uses one of two prompts. The zero-shot prompt is the instruction below followed by the instance's per-cohort summary table. The in-context-example prompt is the full prefix below followed by the same table. The prefix is the instruction plus 2 solved companies built from spent seed 97000, outside every training and test range, generated once with the planted flaw and once without, so the pair shows the same company in both states. The prefix is 1,702 characters and byte-identical across every condition and model that used it. The JSON field names are the harness's internal names. The block below is wrapped for display, and the byte-exact file ships in the supplementary material.

```
You are a careful revenue analyst. Decide whether the company has a
hidden revenue problem. Show your reasoning, then end with ONE line in
exactly this format:
VERDICT_JSON: {"conclusion": "problem" or "healthy", "bleeder_cohort":
"<cohort name>" or null, "revenue_retention": "high" or "low",
"logo_retention": "high" or "low", "largest_cohort": "<cohort with the
most customers>"}

cohort,customers,active,churned,mrr_start_total,mrr_now_total
2024-Q1,45,43,2,7350.0,10110.9
2024-Q2,50,47,3,7450.0,5029.54
2024-Q3,52,48,4,7400.0,8887.33
2024-Q4,50,32,18,7000.0,7600.35
AGGREGATE: customers=197, active=170, churned=27,
mrr_start_total=29200.0, mrr_now_total=31628.12, overall_nrr=1.083,
overall_logo_retention=0.863

Checking revenue retention per cohort, 2024-Q2 fell to 0.68 while
```

```
keeping 94% of its customers. Revenue is contracting inside a cohort
that looks stable on customer count.
VERDICT_JSON: {"conclusion": "problem", "bleeder_cohort": "2024-Q2",
"revenue_retention": "low", "logo_retention": "high", "largest_cohort":
"2024-Q3"}

---

cohort,customers,active,churned,mrr_start_total,mrr_now_total
2024-Q1,45,43,2,7350.0,10110.9
2024-Q2,50,47,3,7450.0,9819.24
2024-Q3,52,48,4,7400.0,8887.33
2024-Q4,50,32,18,7000.0,7600.35
AGGREGATE: customers=197, active=170, churned=27,
mrr_start_total=29200.0, mrr_now_total=36417.82, overall_nrr=1.247,
overall_logo_retention=0.863

Checking revenue retention per cohort, every cohort is above 1.0. One
cohort shows heavy customer loss but its revenue held, which is not a
hidden revenue problem.
VERDICT_JSON: {"conclusion": "healthy", "bleeder_cohort": null,
"revenue_retention": "high", "logo_retention": "high", "largest_cohort":
 "2024-Q3"}

---
```

B SUPPLEMENTARY MATERIAL

The supplementary archive contains the 550 spent test instances with their keys and per-instance tables, the byte-exact instruction and example prefix, the per-instance raw outputs and completion summary for the Mistral untrained-base example-condition pass, the grading module verbatim, and a standalone regrading script with no generator dependency that reproduces every number in the included summary from the raw outputs alone. Raw per-instance outputs for all other passes are retained and reproduce the reported tables the same way. The generator is withheld for the reason given in the Ethics Statement.