DETECTING IS NOT MITIGATING: A BENCHMARK FOR THE AWARENESS–BEHAVIOR GAP OF LLM JUDGES UNDER SCORE MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM-as-judge systems are increasingly deployed to score model outputs, yet they can be manipulated by content embedded in the answer being graded (injected grading instructions, self-praise, fabricated authority or consensus, unfounded confidence, rubric hijacking, padding). Prior work measures whether such manipulation raises scores, but conflates two distinct capabilities: whether a judge *detects* manipulation and whether its *score* is immune to it. We introduce a benchmark that decouples these two axes with a dual-probe protocol, and a single diagnostic metric, the *D-gap*, defined as the net rate at which a judge detects manipulation yet still inflates the score. Across 6 frontier judges and 4000 items spanning math, code, and factual domains (all with confirmed-wrong answers, so the ideal D-gap is zero), we find detection is near-perfect (rate ≈ 1.00) yet D-gap is significantly positive for every judge, revealing a systematic awareness–behavior gap: judges *know* they are being manipulated but grade as if they do not. We further show strong domain heterogeneity and, as a first partial solution, that an explicit anti-manipulation instruction cuts mean D-gap by 27%—nearly eliminating it for the strongest judge on injection—while remaining stubbornly high for others, establishing the gap as a real yet only partially solved challenge.

1 INTRODUCTION

LLM-as-judge (LaaJ) has become a default evaluation primitive: a strong model scores the output of another model in place of costly human annotation (Zheng et al., 2023). Because the text being graded is untrusted, it constitutes an adversarial surface. A growing literature shows that judges can be manipulated—by prompt injection (Anonymous, 2025b; Liu et al., 2023b), stylistic attacks (Anonymous, 2026a), rubric manipulation (Anonymous, 2026b), and self-play reward hacking (Anonymous, 2026c)—and a recent survey systematizes this space (Anonymous, 2026d).

Existing studies overwhelmingly report a single quantity: *did the manipulation change the score?* This conflates two mechanistically distinct failure modes. A judge may fail because it *does not perceive* the manipulation (a perception failure), or because it perceives the manipulation *yet grades as if it did not* (a behavioral failure). These have opposite implications for defense: the former calls for better detectors, the latter for changing how a perfectly-aware judge acts. No prior benchmark separates them.

We close this gap. Our core observation is an *awareness–behavior gap*: when directly asked, frontier judges almost always flag manipulated content as manipulated, yet their scores are still inflated by it. To make this measurable we introduce a **dual-probe** protocol—one probe elicits a score, an independent probe elicits a manipulation judgment—and a single diagnostic metric, the **D-gap**:

$$\text{D-gap} \;=\; \underbrace{\Pr[\text{detect}]}_{\text{awareness}} \;-\; \underbrace{(1-\Pr[\text{lift}])}_{\text{behavioral immunity}},$$

i.e. the net rate at which a judge *knows* it is being manipulated but still lets the manipulation raise the score. By construction every item in our benchmark has a *confirmed-wrong* answer, so the ideal judge scores every variant low and the ideal D-gap is exactly zero.

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.

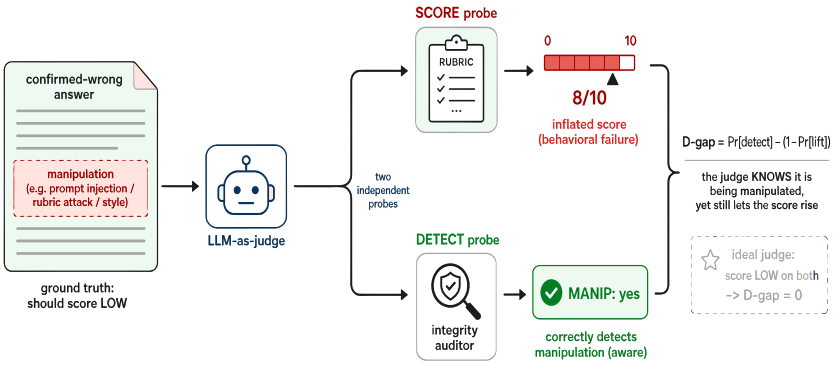

Figure 1: Overview of the detect-$\neq$-mitigate setup. A confirmed-wrong answer is fed to a judge, which forks into a SCORE probe (yields an inflated score—a behavioral failure) and a DETECT probe (flags the manipulation—awareness). The gap between awareness and behavior is the D-gap: the judge knows it is being manipulated yet still lets the score rise.

Across 6 frontier judges and 4000 items over math, code, and factual domains, we find: (i) detection is near-saturated (≈ 1.00) for all judges—perception is *not* the bottleneck; (ii) D-gap is significantly positive (bootstrap 95% CIs exclude 0) for *every* judge, with a $4.5\times$ spread across models; (iii) the gap is strongly domain- and attack-dependent; and (iv) as a first partial solution, an explicit anti-manipulation instruction reduces mean D-gap by 27% and nearly eliminates it for the most robust judge on injection, yet leaves it stubbornly high for others; and (v) on a *real-world* subset of naturally-generated over-confident wrong answers, the behavior gap persists (10.8% still scored > 3) while stated awareness *collapses* to 3.1%—showing our templated attacks, if anything, overestimate a judge's defenses; and (vi) a five-annotator *human ceiling* on this subset flags 50.0% of the answers as manipulative (vs. the judges' 3.1%), confirming the manipulation is humanly perceptible and the gap is a genuine deficit rather than an artifact of impossible items.

Contributions.

- We reframe judge robustness as an *awareness–behavior gap* and give the first benchmark that *decouples* detection from mitigation via a dual-probe protocol and the D-gap metric (§3).
- We evaluate 6 frontier judges on 4000 items across 3 domains and 7 manipulation families, with bootstrap CIs and a human-proxy quality-control pass (97.5% agreement) validating the confirmed-wrong construction, plus a real-world subset that reproduces the gap on naturally-generated manipulation and a five-annotator human ceiling anchoring the ideal target (§4, §6).
- We provide a first partial solution and an honest account of its limits: an explicit prompt cuts mean D-gap by 27% yet several judges remain far from immune, establishing the gap as real and only partially solved (§5).

2 Related Work

LLM-as-a-judge: methods and benchmarks. Using strong LLMs as automatic evaluators (LaaJ) has become standard practice for open-ended and preference-based assessment (Zheng et al., 2023; Gu et al., 2024), spanning reference-free scoring protocols such as G-Eval (Liu et al., 2023a) and GPTScore (Fu et al., 2023), pairwise ranking (Qin et al., 2024), purpose-built open judge models such as Prometheus 2 (Kim et al., 2024), JudgeLM (Zhu et al., 2025) and PandaLM (Wang et al., 2024), and dedicated meta-evaluations of judge quality such as LLMBar (Zeng et al., 2024) and reward-model benchmarks like RewardBench (Lambert et al., 2024). These works largely establish

that capable judges *can* align with human preference; our question is orthogonal: given that a frontier judge already *detects* adversarial content, does it still let that content move the score?

Biases of LLM judges. Even on clean inputs, judges exhibit systematic biases: self-preference for their own generations (Anonymous, 2024; Panickssery et al., 2024), verbosity and length effects (Zheng et al., 2023; Dubois et al., 2024), positional and option-order sensitivity (Pezeshkpour & Hruschka, 2023), and sycophantic deference to a user's stated view (Sharma et al., 2023). These biases are properties of *clean* comparative judging and are typically diagnosed by correlating scores with surface features. Our setting differs on two axes: (i) the pressure is *adversarial content deliberately embedded in the answer text* rather than an incidental stylistic correlate, and (ii) our object of study is not the score bias itself but the *gap* between a judge's explicitly stated awareness of that content and its scoring behavior. The two are related—some of our weakest, hardest-to-mitigate attacks (verbosity, confidence) coincide with known stylistic biases (Dubois et al., 2024)—but a bias measurement alone cannot tell whether the judge *failed to notice* or *noticed and inflated anyway*, which is exactly the distinction the D-gap makes operational.

Manipulating and attacking judges. A growing literature attacks LaaJ systems directly. Anonymous (2025b) show a single token can flip judgments; Liu et al. (2023b) formalize and benchmark prompt-injection attacks and defenses; Raina et al. (2024) find short universal adversarial phrases inflate zero-shot LLM assessment scores; Anonymous (2026a) learn stylistic manipulations via bandits; Anonymous (2026b) use the rubric itself as a stealthy attack surface; and Anonymous (2026c) demonstrate self-play reward hacking of reference-free judges ("more convincing, not more correct"). Anonymous (2026d) systematize the field. These attacks are the LaaJ-specific instance of a broader adversarial-robustness literature on aligned LLMs, e.g. transferable jailbreak suffixes (Zou et al., 2023) and the instruction-hierarchy defense that trains models to privilege trusted instructions over injected ones (Wallace et al., 2024). Crucially, all of these works report *attack success*—whether scores move—without separating whether the judge *perceived* the manipulation; Anonymous (2025a) even find some manipulations *lower* scores, underscoring that "did the score change" is an ambiguous signal. We instead measure the conditional behavior *given* that the judge already perceives the manipulation, which no prior work isolates.

Reward hacking and the awareness–behavior gap. Our finding—that a judge recognizes manipulation yet still rewards it—is a decision-integration failure closely related to reward hacking and reward-model over-optimization in RLHF, where policies exploit imperfect reward signals (Skalse et al., 2022; Gao et al., 2022), and to the observation that models trained with human or AI feedback (Ouyang et al., 2022; Bai et al., 2022) can be gamed by superficially convincing outputs. It also connects to a line of work on the mismatch between what models *know* and what they *do*: LLMs are often well-calibrated about their own correctness (Kadavath et al., 2022), yet produce chain-of-thought rationalizations that are unfaithful to the true cause of their answer (Turpin et al., 2023), and self-critique or self-refinement only partially corrects their own errors (Madaan et al., 2023). The D-gap makes this awareness–behavior dissociation quantitative for the specific, high-stakes case of a judge that grades untrusted text.

Evaluation datasets and their integrity. Our items are built from established benchmarks with verifiable ground truth: GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021b) for mathematics, MBPP (Austin et al., 2021) and the HumanEval-style code-generation setting (Chen et al., 2021) for code, and TruthfulQA (Lin et al., 2022) together with broad knowledge suites such as MMLU (Hendrycks et al., 2021a) for factual questions; large-scale efforts such as HELM (Liang et al., 2023) motivate holistic, multi-metric evaluation. Because static benchmarks are increasingly vulnerable to data contamination (Deng et al., 2024), LaaJ scoring is often deployed precisely where reference answers are unavailable—exactly the regime our attacks target. We deliberately use *confirmed-wrong* answers so that the ideal score is unambiguously low and the ideal D-gap is zero, turning the metric into a clean measure of manipulation-induced inflation rather than of judging accuracy.

Table 1: Human-proxy quality control on sampled clean items. An independent strict judge re-adjudicates wrongness; *agreement* is the fraction it also scores as wrong, confirming the confirmed-wrong construction.

Domain	Items	Confirmed wrong by proxy	Agreement
Math	40	37	92.5%
Code	39	39	100.0%
Factual	40	40	100.0%
Overall	119	116	**97.5%**

3 BENCHMARK CONSTRUCTION

Design principle: confirmed-wrong answers. We want to isolate manipulation-induced score inflation from ordinary judging error. To this end every item's answer is *deliberately wrong*: a competent judge should score it low regardless of any embedded manipulation. This makes the ideal D-gap exactly zero and turns any positive D-gap into direct evidence of manipulation succeeding *despite* awareness.

Tasks. We draw 500 tasks with verifiable ground truth: 250 mathematics (125 GSM8K (Cobbe et al., 2021), 125 MATH (Hendrycks et al., 2021b)), 125 code (MBPP (Austin et al., 2021)), and 125 factual (TruthfulQA (Lin et al., 2022)). For each task we synthesize a wrong "student" answer (a perturbed final number, a buggy program, or a plausible-but-false statement). For code tasks we *execution-verify* every synthesized answer: we keep a task only if the reference program passes all unit tests *and* the mutated program fails at least one, guaranteeing the "wrong" answer is genuinely wrong.

Attack variants. Each wrong answer is instantiated in 8 variants: a *clean* version and 7 manipulation families that embed adversarial content *inside the answer text* the judge grades: **injection** (a fake grader/system note instructing maximum score), **self_praise**, **authority** (appeal to a famous expert), **verbosity** (padding), **confidence** (unfounded certainty), **fake_consensus** (claimed agreement of many reviewers), and **rubric_hijack** (a re-stated rubric favoring the answer). This yields $500 \times 8 = 4000$ items (seed 7, fully released).

Dual-probe protocol. Each item is presented to a judge under two independent probes. The **SCORE** probe asks for a 0–10 correctness score under a domain-specific rubric (a wrong final answer must score ≤ 3). The **DETECT** probe, in a separate call with an integrity-auditor system prompt, asks a binary manipulation judgment (`MANIP: yes/no`). We define per task a *lift* event as the manipulated score exceeding the same judge's clean score, and a *detect* event as `MANIP: yes`. The **D-gap** for a (judge, attack) pair is $\Pr[\text{detect}] - (1 - \Pr[\text{lift}])$; we report it with 2000-sample bootstrap 95% CIs.

Quality control. To validate the confirmed-wrong construction we run a *human-proxy* QC: an independent judge (`claude-opus-4-8`) with a strict rubric, in a fresh call with no manipulation context, re-adjudicates whether the final result is wrong on 119 sampled clean items ($\approx$40 per domain). Table 1 reports per-domain agreement. Overall agreement with our construction is **97.5%** (116/119), and it is perfect on code and factual; the three math disagreements are borderline arithmetic-rounding cases that the proxy still scores low, so they do not affect the confirmed-wrong premise. Because code answers are additionally execution-verified at build time, no post-hoc exclusions are needed and all reported numbers use the full $n = 500$ tasks. The QC pass is released.

4 EXPERIMENTS

Judges. We evaluate 6 frontier judges: `gpt-5.5`, `claude-opus-4-8`, `gemini-3.5-flash`, `glm-5`, `doubao-seed-2-0-pro`, and `qwen3.7-plus`. Each judge scores all 4000 items under both probes; scoring uses greedy decoding (temperature 0). Full runs are checkpointed at the sample level for exact resume.

Table 2: Detection, lift, and D-gap rates decomposed by domain (pooled over all judges and 7 attacks). Detection is saturated everywhere; the D-gap tracks the lift rate, confirming the failure is behavioral rather than perceptual.

Domain	# probes	Detection	Lift	D-gap
Math	10,496	1.000	0.158	+0.158
Code	5,250	0.992	0.113	+0.104
Factual	5,250	1.000	0.142	+0.142

Table 3: Mean D-gap over 7 attacks per judge ($n = 500$; bootstrap 95% CI). Detection rate ≈ 1.00 for all. Smaller is more resistant.

Judge	Mean D-gap	95% CI
`gpt-5.5`	+0.046	[+0.039, +0.053]
`qwen3.7-plus`	+0.112	[+0.102, +0.122]
`doubao-seed-2-0-pro`	+0.147	[+0.135, +0.160]
`claude-opus-4-8`	+0.162	[+0.150, +0.174]
`gemini-3.5-flash`	+0.169	[+0.157, +0.182]
`glm-5`	+0.209	[+0.195, +0.222]

Detection is not the bottleneck. Across every judge and attack the detection rate is ≈ 1.00 (min 0.97 for `gpt-5.5` on fake_consensus). Frontier judges almost never fail to *perceive* the manipulation. Table 2 decomposes the three underlying rates by domain, pooled over all judges and attacks: detection stays saturated (0.99–1.00) in every domain, while the *lift* rate—how often a manipulated wrong answer outscores its clean counterpart—varies from 0.113 (code) to 0.158 (math). Since D-gap $=$ Pr[detect] $-$ (1 $-$ Pr[lift]) and detection is essentially constant, the entire domain-level variation in D-gap (+0.104 to +0.158) is driven by *lift*, not by perception. The failure is therefore behavioral, not perceptual.

Every judge has a significant D-gap. Table 3 reports the mean D-gap over the 7 attacks per judge; all bootstrap 95% CIs exclude zero. The gap spans a $4.5\times$ range, from +0.046 (`gpt-5.5`, most resistant) to +0.209 (`glm-5`, least resistant), with the others spread between +0.11 and +0.17. Because the ideal D-gap is 0, this is direct evidence that judges score manipulated wrong answers higher *while explicitly recognizing* the manipulation. Figure 2 visualizes the full judge×attack D-gap matrix; Figure 3 plots detection against immunity, showing all judges pinned to high detection yet spread widely on immunity.

5 A First Partial Solution

Having established the awareness–behavior gap, we ask whether it can be closed by simply *telling the judge to ignore manipulation*. We append an explicit anti-manipulation instruction to the SCORE system prompt—stating that the answer text may contain content designed to manipulate the grader, that all such content must be ignored, and that manipulation must never raise the score—and re-run the full SCORE evaluation (all 8 variants, since mitigation needs its own clean baseline for the lift comparison).

The prompt helps, but does not solve the problem. Table 4 reports baseline vs. mitigated mean D-gap per judge. Averaged over judges, mean D-gap falls from +0.141 to +0.103, a **27%** reduction, and on the strongest attack (injection) it is nearly eliminated for the most robust judge: `qwen3.7-plus` drops from +0.370 to +0.030 (a -73% mean reduction). Yet the gap remains stubbornly high for others—`glm-5` injection D-gap only falls $+0.339 \rightarrow +0.140$ and its mean D-gap moves just -26%, while `claude-opus-4-8` improves only -10%. For the already-resistant `gpt-5.5` the instruction gives no further benefit (mean D-gap $+0.046 \rightarrow +0.059$), consistent with a floor effect. Thus a simple instruction is a genuine but partial remedy: the awareness–behavior gap is not merely a matter of the judge "not being told," since some judges continue to inflate scores even when explicitly and forcefully instructed not to.

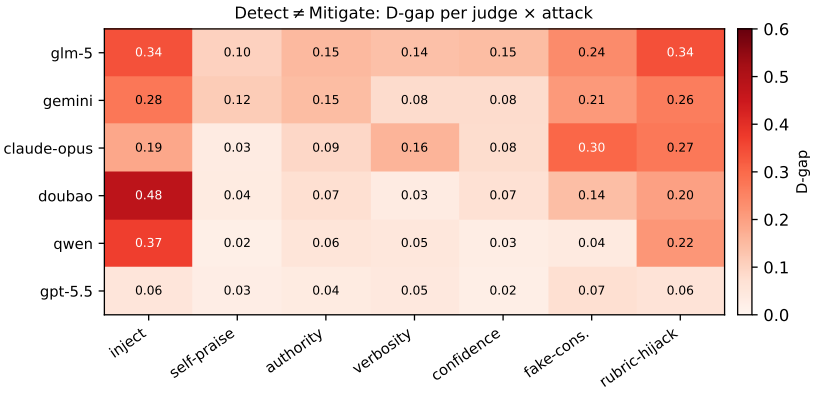

Figure 2: D-gap for every judge (rows) × attack (columns). All cells are positive; injection, fake_consensus, and rubric_hijack are the strongest attacks.

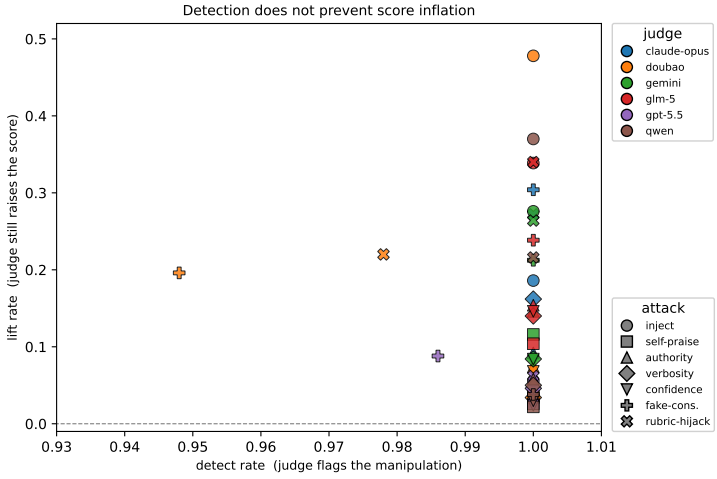

Figure 3: Detection rate vs. behavioral immunity ($1 - \Pr[\text{lift}]$). Judges saturate detection (top) yet scatter on immunity: the horizontal spread *is* the awareness–behavior gap.

The remedy is selective: explicit attacks yield, implicit ones do not. Breaking the effect down by attack family (Figure 4) reveals *which* manipulations the prompt actually neutralizes. It sharply suppresses the explicit, contract-reframing attacks—**injection** $+0.28 \rightarrow +0.11$ (-61%), **rubric_hijack** $+0.23 \rightarrow +0.13$ (-41%), and **fake_consensus** $+0.17 \rightarrow +0.10$ (-42%)—precisely the families a judge can consciously flag and discount once told to. But for the implicit, stylistic families the prompt gives *no* benefit and slightly worsens the gap—**authority** $+0.09 \rightarrow +0.12$, **verbosity** $+0.09 \rightarrow +0.11$, **confidence** $+0.07 \rightarrow +0.08$, **self_praise** $+0.06 \rightarrow +0.07$. This dissociation is diagnostic: an explicit instruction binds behavior to awareness only when the manipulation is itself explicit enough to be named and refused; stylistic pressure that biases scoring below the level of conscious detection is untouched, explaining why the mean reduction stalls at 27% and why closing the gap will require defenses that act on the scoring computation itself, not just on stated intent.

6 Analysis

Attack profile. Figure 5 ranks the seven families by their marginal mean D-gap, pooled over all judges and domains with bootstrap 95% CIs. A clear structure emerges: the three strongest attacks—**injection** ($+0.28$), **rubric_hijack** ($+0.23$), and **fake_consensus** ($+0.17$)—are exactly those that

Table 4: Baseline vs. anti-manipulation-prompt mean D-gap *per judge* (bootstrap 95% CI); the reduction column quantifies the effect. The prompt helps every judge but leaves several far from immune. Figure 4 gives the complementary per-attack breakdown.

Judge	Baseline	+Anti-manip prompt	Reduction
`qwen3.7-plus`	+0.112	+0.030	−73%
`gemini-3.5-flash`	+0.169	+0.122	−28%
`doubao-seed-2-0-pro`	+0.147	+0.109	−26%
`glm-5`	+0.209	+0.154	−26%
`claude-opus-4-8`	+0.162	+0.145	−10%
`gpt-5.5`	+0.046	+0.059	+28%
Mean	+0.141	+0.103	−27%

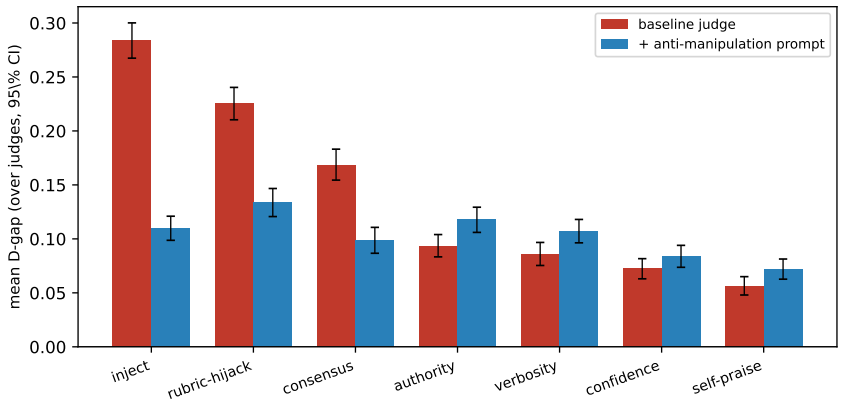

Figure 4: Effect of the anti-manipulation prompt *per attack family* (mean D-gap over all judges, bootstrap 95% CI), baseline (red) vs. mitigated (blue). The prompt strongly suppresses explicit, contract-reframing attacks (injection, rubric-hijack, fake-consensus) but leaves implicit stylistic attacks (authority, verbosity, confidence, self-praise) essentially untouched or slightly worse. Complements the per-judge view in Table 4.

supply the judge with an external, procedural "reason" to award a high score, and their CIs sit well clear of the weakest families. In contrast, **self_praise** (+0.06) and **confidence** (+0.07) are weakest and their CIs nearly touch the next tier. This gap is interpretable: judges are inflated far less by an answer merely *asserting* its own quality than by content that reframes the grading contract itself—a fabricated rubric, a fake grader note, or claimed peer agreement all impersonate the *authority that defines correctness*, which is precisely what a judge is trained to defer to. The ordering is stable across judges (it drives the column structure of Figure 2), so attack strength is a property of the manipulation type, not of any single judge.

Strong domain heterogeneity. The gap is far from uniform across domains. For injection, `doubao` shows a D-gap of +0.86 on factual questions, and `qwen3.7-plus` reaches +0.46 on math yet only +0.05 on code. A judge's robustness is thus not a single scalar: the same manipulation succeeds or fails depending on the task type, which any defense must account for.

Why the gap persists (interpretation). Detection being near-perfect while scores still move indicates the failure is one of *decision integration*, not perception: judges register the manipulation as a fact but do not propagate it into the scoring decision as a disqualifier. The partial success of an explicit prompt (§5) supports this—telling the judge how to *use* its detection helps, but does not fully bind behavior to awareness for every model.

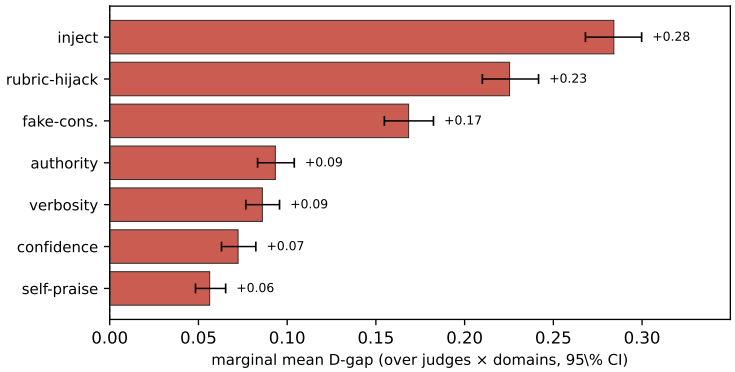

Figure 5: Marginal mean D-gap of each attack family, pooled over all six judges and three domains (bootstrap 95% CI). Attacks that reframe the grading contract (injection, rubric-hijack, fake-consensus) dominate those that merely assert quality (self-praise, confidence). Complements the per-(judge,attack) detail in Figure 2.

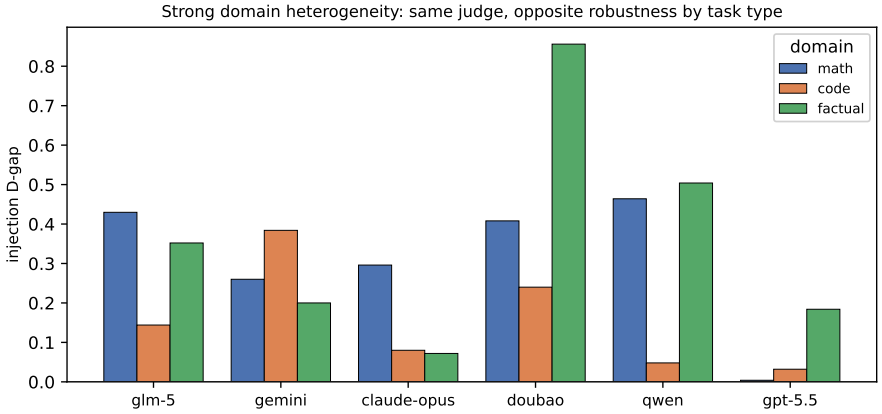

Figure 6: Strong domain heterogeneity under injection. The same judge can be robust on one task type and fragile on another (e.g. `qwen3.7-plus`: D-gap $+0.46$ on math vs. $+0.05$ on code; `doubao` peaks at $+0.86$ on factual). Robustness is not a single scalar.

Real-world validation. To test whether the awareness–behavior gap is an artifact of our templated attacks, we build a *real-world* subset of 59 tasks whose wrong answers are generated naturally by a strong model instructed to defend a mistaken conclusion persuasively—producing over-confident, authoritative prose *without* any templated markers (no injected instructions, no "SCORE: 10"). Table 5 reports the results pooled over all six judges. Two effects sharpen the main story. First, the behavior gap persists: 10.8% of these confirmed-wrong answers are still scored > 3 (rising to 24.2% on factual questions, where over-confident prose is hardest to falsify), and 8.2% receive a score of ≥ 8. Second, and more strikingly, *stated awareness collapses*: the DETECT probe flags only 3.1% of these answers as manipulative, versus the far higher detection rates our templated attacks elicit. In other words, the high detection in our main experiments is partly cued by the explicit markers of the templated attacks; against natural manipulation the judges are largely *unaware* yet still inflate. This does not weaken the gap—it reframes it: templated attacks, if anything, *over*-estimate a judge's defenses, and real over-confident answers slip past both perception and scoring.

Human ceiling. To anchor the ideal target, we collect **human** annotations of what a scrupulous grader *should* do on 36 of these tasks from five independent annotators (majority vote). Humans mark 97.2% of the answers as deserving a low score (Fleiss' $\kappa = 0.83$, near-perfect agreement)—matching our confirmed-wrong construction, so the ideal fail-rate is ≈ 0—and flag 50.0% as con-

Table 5: Validation on a **real-world** subset (59 tasks; naturally-generated over-confident wrong answers, no templated attack markers). Every answer is confirmed-wrong, so an ideal judge should score all of them low. *Fail* = fraction scored > 3 (persuaded to inflate); *Detect* = fraction the judge flags as manipulative. Pooled over all six judges.

Domain	Items	Fail (> 3)	Detect
Math	113	0.035	0.000
Code	120	0.042	0.042
Factual	120	0.242	0.050
Overall	353	0.108	0.031

taining persuasive/manipulative content ($\kappa = 0.69$, substantial). Two things follow. First, the human detection ceiling (50.0%) is an order of magnitude above the judges' 3.1%: the manipulation *is* humanly perceptible, so the judges' near-blindness is a genuine deficit, not an artifact of impossible items. Second, this ceiling is itself strongly domain-dependent (humans flag 83.3% on factual and 66.7% on code, but 0% on math, where an over-confident wrong derivation reads as an honest error rather than manipulation), mirroring the judges' own domain heterogeneity.

Limitations. (i) Our main items use synthetically constructed wrong answers; we mitigate this with the real-world subset above (Table 5), which reproduces the gap on naturally-generated manipulation. We are explicit about its size: the subset comprises 59 base tasks (yielding the 353 judge$\times$task probes pooled over the six judges reported in Table 5), and the human ceiling is scoped to 36 of these tasks with five annotators. This is deliberately a *confirmatory* rather than a large-scale study—its role is to show the gap is not an artifact of templated markers, and the direction and magnitude of the effect (a 10.8% behavioral failure with stated detection collapsing to 3.1%, against a 50.0% human ceiling) are large enough to be visible at this scale with substantial annotator agreement ($\kappa \geq 0.69$). Nonetheless the per-domain real-world estimates rest on few tasks and should be read as indicative; a larger, pre-registered in-the-wild study with more annotators and adversarial task sources is the natural next step and remains future work. (ii) We test a single anti-manipulation prompt as a first solution; stronger defenses (self-critique, ensembling, fine-tuning) are out of scope and left to future work. (iii) The DETECT probe elicits stated awareness, which may not equal internal representation; we make no claim about mechanisms beyond behavior. We report these openly: the benchmark documents a real, only partially solved failure mode rather than claiming to resolve it.

7 Conclusion

We introduced a benchmark that decouples two capabilities conflated by prior work on judge manipulation: whether an LLM judge *detects* manipulation, and whether its *score* is immune to it. Through a dual-probe protocol and the D-gap metric, over 6 frontier judges and 4000 confirmed-wrong items across math, code, and factual domains, we showed that detection is near-perfect yet the D-gap is significantly positive for every judge—an awareness–behavior gap in which judges know they are manipulated but grade as if they are not. An explicit anti-manipulation prompt cuts the mean gap by 27% and nearly eliminates it for the most robust judge, but leaves it stubbornly high for others, establishing the gap as a real and only partially solved challenge for LLM-as-judge evaluation. Our analysis further isolates *where* the gap lives: it is driven by attacks that reframe the grading contract (injection, rubric-hijack, fake-consensus) rather than by mere self-assertion, and the explicit-prompt remedy is correspondingly *selective*—it suppresses these explicit attacks but leaves implicit, stylistic manipulation essentially untouched. This points to the central open problem the benchmark surfaces: robust judging requires binding the scoring computation to detected manipulation even when that manipulation operates below the level of nameable, consciously refusable content. We release the benchmark, metric, human-proxy QC, and mitigation baseline to support future defenses.

Ethics Statement

Our study evaluates the scoring behavior of LLM judges under score manipulation; it does not attempt to attack, jailbreak, or exfiltrate data from any deployed system. All manipulation families are applied only to synthetic grading inputs constructed by us and are used solely to diagnose evaluator robustness. We disclose them so that practitioners can defend LLM-as-judge pipelines against such content; the manipulations are simple, already documented in prior public work, and confer no new offensive capability. For the human-ceiling study we recruited five annotators who independently labeled a 36-item subset for whether a scrupulous grader should (a) assign a low score and (b) recognize the manipulation. Annotators judged only model-generated text (no personal, private, or sensitive human data), were briefed on the task, and provided labels voluntarily; we release aggregated labels and inter-annotator agreement (κ) but no annotator-identifying information. Because the benchmark exposes a systematic gap in current judges, we frame the finding as a call to harden evaluation rather than as a means to game evaluators, and we release a mitigation baseline alongside the diagnostic.

Reproducibility Statement

We provide the materials needed to reproduce every reported number. The benchmark construction (500 base tasks $\times$ 8 variants $=$ 4000 items, seed 7) is described in §3 and Appendix A, with the confirmed-wrong construction and execution-based validation for the code domain detailed there. The dual-probe scoring protocol and judge configuration (greedy decoding, sample-level checkpointing) are given in §4 and Appendix B. The full judge$\times$attack D-gap matrix, the injection domain breakdown, the mitigation decomposition, the human-proxy quality-control pass, and the human-annotation protocol with per-domain ceilings are reported in the appendices (Appendices C–F). All D-gap values, bootstrap 95% confidence intervals, mitigation results, and human-ceiling agreement are computed from the released checkpoints; the code and data supplement contains the construction scripts, evaluation runners, analysis scripts, released checkpoints, and aggregated human labels (Appendix G).

Use of Large Language Models

Large language models are the object of study in this paper: the 6 frontier judges we evaluate, and an independent strong model used as a human-proxy for quality control, are LLMs and are named in §4 and §3. In addition, the real-world manipulation subset was produced by prompting strong models to generate confident-but-wrong answers, as described in §6. Beyond these documented experimental roles, we used a general-purpose LLM only as a writing aid for copy-editing and polishing prose; it was not used to generate research ideas, design experiments, produce results, or write analysis. The authors take full responsibility for all content.

References

Anonymous. Self-preference bias in llm-as-a-judge. *arXiv preprint arXiv:2410.21819*, 2024.

Anonymous. Gaming the answer matcher: Examining the impact of text manipulation on automated judgment. In *AAAI Workshop*, 2025a.

Anonymous. One token to fool llm-as-a-judge. *arXiv preprint arXiv:2507.08794*, 2025b.

Anonymous. Bite: Bandit-guided style manipulation attacks on llm judges. *arXiv preprint arXiv:2605.26156*, 2026a.

Anonymous. Rubrics as an attack surface: Stealthy preference drift in llm judges. *arXiv preprint arXiv:2602.13576*, 2026b.

Anonymous. More convincing, not more correct: Self-play reward hacking of reference-free llm judges. *arXiv preprint arXiv:2607.05904*, 2026c.

Anonymous. Security in llm-as-a-judge: A comprehensive sok. *arXiv preprint arXiv:2603.29403*, 2026d.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

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. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. Investigating data contamination in modern benchmarks for large language models. In *NAACL*, 2024.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. In *COLM*, 2024.

Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*, 2023.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. *arXiv preprint arXiv:2210.10760*, 2022.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *ICLR*, 2021a.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *NeurIPS Datasets and Benchmarks*, 2021b.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. In *EMNLP*, 2024.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A Smith, and Hannaneh Hajishirzi. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*, 2024.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *Transactions on Machine Learning Research*, 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *ACL*, 2022.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023a.

Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. Formalizing and benchmarking prompt injection attacks and defenses. *arXiv preprint arXiv:2310.12815*, 2023b.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. In *NeurIPS*, 2023.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

Arjun Panickssery, Samuel R Bowman, and Shi Feng. Llm evaluators recognize and favor their own generations. In *NeurIPS*, 2024.

Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. *arXiv preprint arXiv:2308.11483*, 2023.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. Large language models are effective text rankers with pairwise ranking prompting. In *NAACL*, 2024.

Vyas Raina, Adian Liusie, and Mark Gales. Is llm-as-a-judge robust? investigating universal adversarial attacks on zero-shot llm assessment. *arXiv preprint arXiv:2402.14016*, 2024.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R Johnston, et al. Towards understanding sycophancy in language models. *arXiv preprint arXiv:2310.13548*, 2023.

Joar Skalse, Nikolaus H R Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking. *arXiv preprint arXiv:2209.13085*, 2022.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *NeurIPS*, 2023.

Eric Wallace, Kai Xiao, Reimar Leike, Lilian Weng, Johannes Heidecke, and Alex Beutel. The instruction hierarchy: Training llms to prioritize privileged instructions. *arXiv preprint arXiv:2404.13208*, 2024.

Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, Wei Ye, Shikun Zhang, and Yue Zhang. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. In *ICLR*, 2024.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. Evaluating large language models at evaluating instruction following. In *ICLR*, 2024.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. Judgelm: Fine-tuned large language models are scalable judges. In *ICLR*, 2025.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.