RIGHT ANSWER, WRONG REASONING IN VERIFIER-FILTERED SELF-TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-training for reasoning keeps a sampled solution only if its final answer is verified correct, and recovers problems the model cannot solve by showing it the answer. In logic and planning puzzles where every intermediate claim can be checked exactly, a verified answer turns out to say little about the reasoning behind it. The claims in a solution are of two kinds: *forced* claims, which the answer depends on, such as the moves of a plan, and *unforced* claims, which it does not, such as descriptions of the state along the way. Any solution the verifier accepts has its forced claims right almost by construction; its unforced claims go unchecked, and in accepted solutions written with an answer hint up to 74% of them are false. On Knights-and-Knaves problems the model could not solve, 67.1% of these solutions contain a false claim; giving the same information as the opening steps of a solution lowers this to 0.4%. The false claims do not stay in the data. Models trained on accepted but unsound solutions give correct answers resting on a false claim 53–75 percentage points more often than models trained on sound solutions to the same problems, with no consistent difference in accuracy, and the same inheritance appears in planning. Filtering the accepted solutions with the checker before training removes almost all of it. The effect appears in all three model families measured (Qwen, Llama, and OLMo) and, in Qwen3 and OLMo-3, shrinks with scale without vanishing. An outcome verifier certifies answers, not reasoning, and a model trained on what it certifies can learn to be right for the wrong reasons.

1 INTRODUCTION

Self-training for reasoning keeps a sampled solution only if its final answer is verified correct and trains the model on what it kept (Zelikman et al., 2022; Yuan et al., 2023; Singh et al., 2024); the same check supplies the reward in reinforcement learning with verifiable rewards (Guo et al., 2025). Problems the model never solves are recovered by showing it part of the answer and keeping whatever verifies (Zelikman et al., 2022). Both practices assume that a verified answer comes with sound reasoning. *When a solution passes outcome verification, are the claims it makes along the way true?*

Figure 1 shows why the question matters. Told that Helena is a knave, the model assumes she is a knight, declares no contradiction, and reasons from there through a chain of false claims to a correct final answer, which the verifier accepts. The case is typical: 63.3% of solutions accepted this way contradict the hint they were given.

Prior work asks whether reasoning reflects, causes, or predicts the answer (Turpin et al., 2023; Lanham et al., 2023; Chen et al., 2025; Arcuschin et al., 2026; Yu et al., 2026), judges it with an LLM (Wen et al., 2026), or measures what training on it does to accuracy (Lee et al., 2026); none checks whether its claims are true, which requires ground truth for every claim. Logic and planning puzzles supply it: in Knights-and-Knaves an exact solver decides every hypothesis a solution tests, and in Blocksworld, Sokoban, and Countdown a simulator or exact arithmetic decides every action and every description of the state.

In these puzzles one distinction organizes the answer. A claim is *forced* if the final answer depends on it, as a plan depends on its moves, and *unforced* if the answer can be right while the claim is false, as with a description of where the boxes are. An outcome verifier enforces forced claims indirectly

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.

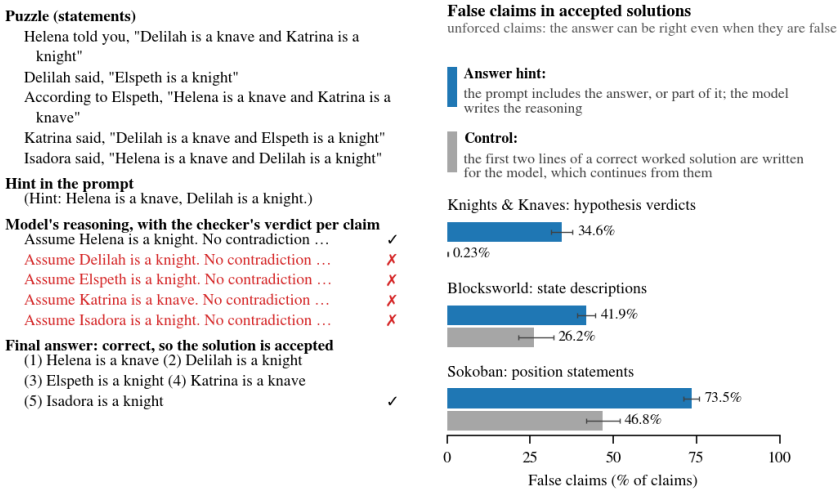

Figure 1: **Accepted solutions can rest on false reasoning.** *Left:* a real accepted solution (Qwen3-4B; lines shortened only at "…"). The hint says Helena is a knave, but the model assumes she is a knight, finds "no contradiction," and every claim it reasons from there is false (×), even the one about Delilah that agrees with the hint; its final answer is correct, so the verifier accepts it.
Right: share of unforced claims that are false in accepted solutions, under the answer hint and the first-steps control, with 95% bootstrap CIs. Only in Knights-and-Knaves do the two reveal the same information. Forced claims in the same solutions are correct, as acceptance requires.

and unforced claims not at all, so the forced claims in any accepted solution are right almost by definition. Knights-and-Knaves and Countdown sit at the extremes: every claim in Knights-and-Knaves is unforced, and false claims are common; every line of a Countdown solution is forced arithmetic, and the hint adds almost none. The same pattern appears in Blocksworld and Sokoban, where every solution contains both kinds of claim: the actions are right, and many of the descriptions of the state are wrong.

Our findings are:

1. **False claims gather where the answer cannot see them (§4, §5).** In accepted solutions written with an answer hint, at most 1.4% of forced claims are false and up to 74% of unforced claims are, often in the same solution; in Knights-and-Knaves, 67.1% of accepted hinted solutions contain a false claim.
2. **Where claims are unforced, how the answer is given matters (§5).** In Knights-and-Knaves, where the hint and the first steps carry the same information, giving the two identities as the opening of a derivation lowers the rate from 67.1% to 0.4%; stating them as facts leaves nothing to check.
3. **The false claims are inherited (§6).** Models fine-tuned with LoRA at 4B and 8B on solutions the verifier cannot tell apart, differing only in whether the unforced claims are true, give correct answers resting on a false claim 53 to 75 percentage points more often in Knights-and-Knaves, with no consistent difference in accuracy, and have 27 to 50 points more of their unforced claims false in Blocksworld and Sokoban, in both model families and every seed. One round of the standard loop roughly doubles the share of correct answers resting on a false claim; filtering the accepted solutions before training removes almost all of it in Knights-and-Knaves, with no clear effect on accuracy across three seeds.

The effect appears in all three families measured, Qwen (Qwen3 and Qwen3.5), Llama, and OLMo-3, and in Qwen3 and OLMo-3 it shrinks with size without vanishing, from 81.4% of accepted hinted solutions at Qwen3-1.7B to 10.1% at Qwen3-14B (§4). The strongest of three LLM judges matches the checkers almost exactly and weaker judges agree less, and the rates hold across sampling temperatures and solution lengths (§7). The results refine the argument of Wen et al. (2026) that outcome rewards favor correct reasoning when it is more likely to reach the right answer: that condition holds here, but the reward enforces it by penalizing failures, which success-only self-

training discards (§6). Within the limits of synthetic puzzles and written claims, a model can learn to be right for the wrong reasons (McCoy et al., 2019), and the verifier that trained it is the one instrument that cannot tell.

2 Related Work

Outcome rewards and reasoning. Yu et al. (2026) show that outcome rewards raise accuracy without reliably making reasoning causally important or legible, and that brief training on expert solutions restores both; their measures ask whether reasoning is used, and this paper asks whether its claims are true. Countdown, one of the tasks where their causal-importance measure rises, is the case here in which every claim is forced. Wen et al. (2026) argue that outcome rewards favor sound reasoning when it reaches the right answer more often; that condition holds in these data, and what fails is how self-training uses the reward. Shao et al. (2026) find gains from uninformative rewards largely confined to one model family, and Helff et al. (2026) show trained models passing answer-level verifiers by listing labels without inducing rules. Whether reinforcement learning improves reasoning or sharpens sampling is debated (Yue et al., 2025); Zhu et al. (2025) find that training only on penalized failures matches standard reinforcement learning, consistent with the corrective signal lying in the failures (§6).

Rationalization and recovery. STaR (Zelikman et al., 2022), rejection-sampling fine-tuning (Yuan et al., 2023), and ReST-EM (Singh et al., 2024) train on accepted solutions, and STaR recovers unsolved problems by showing the answer. Lee et al. (2026) show that answer-conditioned solutions degrade distillation in ways correctness filters cannot see, and trace the harm to asking the model to reason toward a given answer; this paper measures the cost claim by claim and follows it into the trained model. Jiang et al. (2026) report more step errors in distilled medical models that answer correctly. Methods that supply partial solutions to recover hard problems (Yan et al., 2025; Huang et al., 2025; Zhang et al., 2025b;a; Liu et al., 2025; Xi et al., 2024; Li et al., 2025; Wu et al., 2026; Zhang et al., 2026) mostly give reasoning steps, so these results do not transfer to them directly; they suggest auditing what a hint states alongside the accuracy it buys.

Checking steps and faithfulness. Process supervision labels steps with human annotators (Lightman et al., 2024), Monte-Carlo estimates (Wang et al., 2024), or LLM judges, and ProcessBench (Zheng et al., 2025) and PRMBench (Song et al., 2025) show how hard such labels are to get right; ReCEval (Prasad et al., 2023) scores steps for correctness and informativeness, and Saparov & He (2023) checked steps formally on synthetic logic. Chains of thought can also fail to reflect how a model reached its answer (Turpin et al., 2023; Lanham et al., 2023; Chen et al., 2025; Arcuschin et al., 2026). This paper checks exactly the solutions a verifier admits into training, locates their false claims by whether the answer depends on them, and follows them into the models trained on them. Models continue the format of their demonstrations (Wang et al., 2023), the likeliest reason the form of a hint matters (§5). Closest in measurement, Ye et al. (2025) find that about a quarter of correct Qwen2.5-Math-7B responses contain flawed reasoning by an LLM judge's account and filter them with the process reward model used in §6, Ding et al. (2025) penalize such flawed positives during reinforcement learning, and Chen et al. (2026) report low judged soundness alongside high accuracy for outcome-trained models on Knights-and-Knaves; all three judge flaws with a model, and this paper checks every claim exactly.

Domains. Knights-and-Knaves (Xie et al., 2025a;b), Blocksworld (Valmeekam et al., 2023), and Countdown and Sokoban-like search (Gandhi et al., 2024; Lehnert et al., 2024) are established testbeds; here they serve because every claim is decidable and each has a known split between forced and unforced claims. Models can track game state internally (Toshniwal et al., 2022; Li et al., 2023; Kim & Schuster, 2023); in Sokoban, writing it down falsely goes with solving fewer puzzles (§6).

3 Experimental Setup

Self-training and recovery. The model samples 32 solutions per problem, an outcome verifier keeps those with a correct final answer, and the model is fine-tuned on what was kept. A problem

never answered correctly in 32 attempts is *unsolved*. Rationalization (Zelikman et al., 2022) shows the model part of the answer, samples again, and trains on the solutions that verify, with the hint removed; an unsolved problem is *recovered* if one of these is accepted.

Forced and unforced claims. A claim is *forced* if the answer cannot be correct unless the claim is, and *unforced* if the answer can be correct either way. The forced claims are the plan actions in Blocksworld, the moves in Sokoban, and the arithmetic steps in Countdown; the unforced claims are the hypothesis verdicts in Knights-and-Knaves and the descriptions of the state in Blocksworld and Sokoban. Which claims are forced is a property of the task, readable before any model is run. In accepted solutions, forced claims are right almost by construction, and the measurements concern the unforced ones.

Domains. In *Knights-and-Knaves*, each person is a knight, who tells the truth, or a knave, who lies, and the task is to identify everyone from their statements (Smullyan, 1978; Johnson-Laird & Byrne, 1990). A solution assumes identities and checks them for contradiction, so it is a sequence of hypotheses with verdicts that an exact solver decides. Puzzles come from our own generator: 3 to 5 people for training (2,000) and in-distribution evaluation (1,000), and 6 to 8 for out-of-distribution evaluation (1,000). In *Blocksworld* (4 blocks) and *Sokoban* (6×6 grids, one box), a solution is a sequence of actions or moves, each followed by a description of the resulting state, and a simulator checks both. In *Countdown*, four numbers are combined by arithmetic into a target, one step per line. All puzzles are freshly generated. Generators, grading, and problem sets are in Appendix A, and ProntoQA (Saparov & He, 2023), ZebraLogic (Lin et al., 2025), and MATH (Hendrycks et al., 2021) in Appendix J.

Giving part of the answer. Under the *answer hint*, part of the answer is stated in the question: "(Hint: Reginald is a knave, Cordelia is a knight.)" in Knights-and-Knaves, the whole reference plan or expression elsewhere. Under the *first steps*, the question is unchanged and the first two steps of a solution are written for the model to continue ("Assume Reginald is a knave. No contradiction is found."). *Unaided* solutions are the model's own, on problems it already solves. In Knights-and-Knaves the conditions run on unsolved problems, and the hint and the first steps carry exactly the same information, so they differ only in form. Elsewhere they run on every problem and the hint reveals more, so the evidence there comes from comparing claim types within hinted solutions (§5). Supplied claims are never counted, and the first steps serve only as a control. Appendix D gives every template.

Models. Knights-and-Knaves is measured in Qwen3 (Yang et al., 2025) (1.7B to 14B), Qwen3.5 (2B to 9B), Llama-3.2-3B and Llama-3.1-8B (Grattafiori et al., 2024), and OLMo-3 (Team Olmo et al., 2025) (7B and 32B); Blocksworld and Sokoban in Qwen3-8B, Qwen3.5-4B, Llama-3.1-8B, and OLMo-3-32B; and Countdown in Qwen3-4B, Qwen3.5-4B, and Qwen3-8B. A model enters a domain only if its unaided pass@1 lies between 5% and 90% (Appendix Table 5). The main Knights-and-Knaves models are Qwen3-4B and Llama-3.1-8B after one round of the loop (LoRA, 300 steps; Appendix E), which leaves 694 and 1,027 training puzzles unsolved. Comparisons across models use base checkpoints on a shared set of 523 problems, those the hint recovered for the round-one Qwen3-4B model in an earlier pass with a more lenient answer check (Appendix A). All sampling uses temperature 0.6. Training targets use the unhinted prompt and end at the answer, and almost none refer back to the hint (at most 0.04% per domain and model; Appendix D).

False claims. A Knights-and-Knaves claim is a hypothesis with its final verdict, false if the verdict is wrong given what the solution has assumed so far; this *local* definition is the default, and judging against the true solution gives the same picture (§7). Reversing an earlier verdict is a *self-correction*, and the later verdict is the one scored. A planning claim is false if an action is illegal or a description asserts something untrue of the simulated state, and a Countdown line if its arithmetic is wrong or it uses an unavailable number (Appendix F). Rates are reported *per solution* (the share of accepted solutions with a false claim) or *per claim*, always among solutions with a correct answer. A hinted Knights-and-Knaves solution *contradicts its hint* if it declares "no contradiction" for the opposite of an identity it was given, and *never retracts* if that stays its final verdict; this measure needs no checker.

Judges and reward. Three LLM judges, gpt-oss-120b (OpenAI, 2025), GPT-6 Sol (OpenAI, 2026), and DeepSeek-V4-Pro (0423) (DeepSeek-AI et al., 2026), independently list the false claims in samples of solutions (Appendix G). To see what an outcome reward would reinforce, each problem's samples are re-scored with the GRPO advantage (Shao et al., 2024); no policy is trained. Intervals are 95% bootstrap intervals over problems, not adjusted for multiple comparisons; every result the conclusions rest on clears zero by a wide margin.

4 The Verifier Accepts Unsound Reasoning

Table 1 takes the 694 Knights-and-Knaves problems the main Qwen3-4B model never solves, gives it two of the identities either as an answer hint or as first steps, and checks every claim of the 32 solutions sampled per problem under each; the unaided column shows the model's own accepted solutions on problems it can solve.

Table 1: **What the verifier keeps from problems the model could not solve (Knights-and-Knaves).** Qwen3-4B after one round of self-training, 694 unsolved problems, 32 samples per problem per condition. The hint and the first steps reveal the same two identities. The two rows on contradicting the hint need no checker.

		Answer hint	First steps	Unaided
Problems recovered (of 694)		442 (63.7%)	524 (75.5%)	—
Accepted solutions with ≥ 1 false claim	(per solution)	67.1%	0.4%	28.2%
Claims that are false	(per claim)	34.6%	0.2%	17.1%
Declare "no contradiction" against the hint	(per solution)	63.3%	—	n/a
and never retract		62.7%	1 in 7,761	n/a
Self-corrections per 100 solutions		5.5	0.0	26.6

Recovery without soundness. The hint recovers 442 problems, and 67.1% of the solutions the verifier accepts contain at least one false claim; 34.6% of all their claims are false. The difficulty of the problems does not explain this: given the same identities as first steps, the model recovers more problems (524), and 0.4% of its accepted solutions contain a false claim. Nor does the way the problems were selected: rerun on all 833 problems the model never solves under the strict answer check, the hint leaves 69.0% of accepted solutions with a false claim and the first steps 0.6% (Appendix A). In 63.3% of accepted hinted solutions the model assumes the opposite of a hinted identity and declares it consistent, and in 62.7% it never takes this back (Figure 1); under the first steps this happens in 1 of 7,761 solutions. Unaided search makes mistakes too (28.2% of accepted solutions) but revises them, 27 times per 100 solutions against 5.5 under the hint. The verifier accepts all three kinds whenever the answer is right, and an outcome reward would reinforce them: re-scored with the GRPO advantage, a reward that also required no false claims would remove 55.9% of the positive advantage the outcome reward gives hinted solutions (Appendix H).

Across families and sizes. Figure 2 repeats the measurement on base models from all three families, on the shared set of 523 problems, at least 70% of which every model leaves unsolved unaided. The effect appears in every model. In Qwen3 it falls from 81.4% of accepted hinted solutions at 1.7B to 10.1% at 14B, and in OLMo-3 from 88.6% at 7B to 30.9% at 32B; Qwen3.5 falls from 2B to 4B and then levels off (36.2, 19.4, and 22.4%), and Llama stays near the ceiling (96.8% at 3B, 97.8% at 8B). Under the first steps the rate stays low at every size, so the hint leaves at least ten times as many solutions with a false claim throughout.

5 Unsoundness Is Confined to Unforced Claims

A false forced claim makes the answer wrong, and the solution is rejected; a false unforced claim costs nothing. Among accepted solutions the forced half of this account holds by construction, so its content lies in the unforced half: whether a model left unchecked writes false claims often, and whether a task's structure predicts where. Knights-and-Knaves and Countdown show the two extremes, and in Blocksworld and Sokoban the same pattern appears inside a single solution.

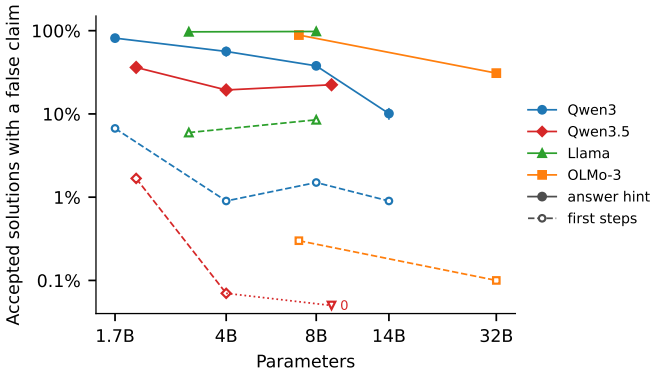

Figure 2: **Knights-and-Knaves across model families and sizes.** Share of accepted solutions with a false claim (per solution, log scale) under the answer hint (filled markers, solid lines, 95% intervals) and the first steps (hollow markers, dashed lines); base models on the shared 523 problems. Qwen3.5-9B's first steps produced no false claims (drawn at the bottom edge). Values in Appendix Table 6.

Table 2: **False claims by claim type.** Share of claims that are false among accepted solutions (per claim), under the answer hint and the first steps. Forced claims are ones the answer depends on; unforced claims are ones it does not. Blocksworld state descriptions count as false when they assert something untrue. Where the first steps show more false forced claims than the hint (OLMo-3-32B in Blocksworld, Qwen3-8B in Countdown), both rates are below 1.4%.

		Forced		Unforced	
Domain	Model	Hint	First steps	Hint	First steps
Knights-and-Knaves	Qwen3-4B	—	—	34.6%	0.2%
Blocksworld	Qwen3-8B	0.04%	0.00%	41.9%	26.2%
	Qwen3.5-4B	0.07%	0.00%	50.1%	26.4%
	Llama-3.1-8B	1.38%	0.06%	22.9%	11.5%
	OLMo-3-32B	0.35%	0.63%	33.2%	27.3%
Sokoban	Qwen3-8B	0.17%	0.10%	73.5%	46.8%
	Qwen3.5-4B	0.16%	0.00%	34.1%	22.2%
	Llama-3.1-8B	1.12%	0.05%	66.6%	45.8%
	OLMo-3-32B	0.05%	0.00%	7.8%	0.2%
Countdown	Qwen3-4B	1.38%	0.35%	—	—
	Qwen3.5-4B	0.45%	0.26%	—	—
	Qwen3-8B	0.88%	1.30%	—	—

Within a solution. In Blocksworld and Sokoban every solution contains both kinds of claim, written by the same model and checked by the same simulator, so the comparison holds everything fixed except whether the answer depends on the claim. The forced claims are correct, as acceptance requires (at most 1.4% false for any model; Table 2), which confirms that extraction and simulation agree with the verifier. The unforced claims beside them are often false, up to 50.1% of Blocksworld state descriptions and 73.5% of Sokoban position statements, and the split holds in all four models, including the newer Qwen3.5-4B. Under the first steps, 12 to 47% of the Qwen and Llama models' descriptions are still false: however a solution begins, nothing checks what it says about the state. Difficulty does not explain the pattern, since in Knights-and-Knaves the same kind of claim on the same problems goes from 34.6% false under the hint to 0.2% under the first steps.

Countdown: every claim forced. In Countdown the final expression depends on every line, so the verifier checks everything and the hint has almost nothing to corrupt. Against the first steps, it changes the share of false claims by at most about one percentage point in each model (Table 2; Appendix P), an order of magnitude below the domains with unforced claims; per solution the change is small but not always zero (+1.3 points for Qwen3-4B, 95% CI $[+0.8, +1.8]$). Across 1,200 hinted solutions there is not a single hypothesis-and-verdict line, and only 13 lines that are not equations. When the answer checks every line, the model does not write lines the answer does not check.

How the answer is given. Knights-and-Knaves is the one domain where the hint and the first steps carry the same information, so it isolates form: the same two identities lead to false claims in 67.1% of accepted solutions as a hint and 0.4% as the opening of a derivation (94.6% and 13.1% for Llama). The reason lies in how the model begins. Left to itself, it assumes that the first person named is a knight. On all 833 unsolved problems, hinted solutions contain a false claim 10.0% of the time when that person is a knight and 84.0% of the time when they are a knave, and 80% of these problems begin with a knave, Helena among them. Even where the first person is a knight, the hint leaves more false claims than the first steps ($+7.7$ points, 95% CI $[+3.8, +12.6]$), so the mix of problems amplifies the effect without creating it. Handed a fact that contradicts its opening, the model keeps the opening, declares it consistent, and writes the hinted answer at the end; the first steps put the hinted identities where its opening would go, so nothing collides. How much a round produces therefore depends on how many of its unsolved problems begin with a knave: a second round, whose unsolved problems mostly do not, yields 18.0% (Appendix O). The collision explains why the model writes the false claim, and the claim's being unforced explains why the verifier keeps it.

Stating the identities as facts has the opposite effect on what can be checked. Lee et al. (2026) trace the harm of answer-conditioned solutions to asking a model to reason toward a given answer. Opening the solution with the identities stated as plain facts ("Reginald is a knave") recovers at least as many problems as the first steps (76.5 to 78.1% for Qwen, against 75.5%), but the model then lists the remaining identities and answers, writing 42 characters of reasoning on average against 269 under the hint, and no accepted Qwen solution contains a hypothesis to check. Such solutions are uncheckable, and the verifier accepts them as readily as any other; a model continues the format it is given (Wang et al., 2023). Appendix I gives all six phrasings tried. In the planning domains the hint reveals more than the first steps, so form cannot be isolated there, and the descriptions stay unreliable under either.

These claims matter even though the answer does not depend on them. Written reasoning is read as an account of how the answer was reached, a reader cannot tell which claims are decoration, and a self-training loop trains on whatever its verifier accepts.

6 Unsound Reasoning Is Inherited

Models trained on accepted solutions with false claims reproduce those claims in their own answers to new problems, and accuracy gives no sign of it.

Design. Two training sets are built per domain and family from accepted hinted solutions, indistinguishable to the verifier: every solution reaches a valid answer and, in planning, has every forced claim correct. *Sound* contains only solutions with no false unforced claim and *Unsound* only solutions with at least one; both draw the same number of solutions from each problem, so they share problems, answers, and amount of data (128 to $1{,}542$ solutions per set; Appendix E). The two models start from the same checkpoint and train identically for about 5.3 passes. A third model, *All-accepted*, trains on every accepted hinted Knights-and-Knaves solution for Qwen ($4{,}830$, 67.1% with a false claim), as a standard loop would, and serves as a reference. Each model solves held-out problems without a hint. The measurement is the share of valid answers with a false claim in Knights-and-Knaves (per solution) and the share of unforced claims that are false in planning (per claim), compared on problems both models solve (Appendix B).

Results. In every domain, family, and seed, the Unsound model's valid answers rest on false claims far more often, and every interval excludes zero (Figure 3): by 75 and 53 percentage points for Qwen and Llama in Knights-and-Knaves and by 31 and 27 points of unforced claims in Blocksworld and 50 and 41 in Sokoban. The models move in opposite directions from where they started: in Knights-and-Knaves, Qwen's rate goes from 22.8% to 4–8% after sound training and to 76–79% after unsound training. Forced claims stay correct throughout (0.00 to 0.51% false among valid plans), so what the models inherit is falsehood in exactly the claims the answer does not check. The gap holds under the global definition, length and wording do not explain it (§7), and it is already large at a quarter of the training ($+63.0$, $+28.6$, and $+28.7$ points; Appendix K). The All-accepted model shows what a standard loop produces: 48.4% of its valid answers contain a false claim, against 22.8% for the starting model, while its pass@1 rises slightly (44.8 against 43.0). One round

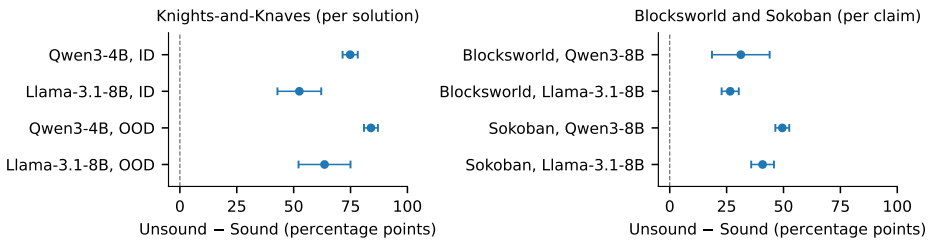

Figure 3: **Models trained on unsound solutions give valid answers that rest on false claims.** Unsound minus Sound, in percentage points: mean over three training seeds, with error bars of one standard deviation. Knights-and-Knaves: share of valid answers containing a false claim (per solution), in and out of distribution. Planning: share of unforced claims that are false among valid plans (per claim). In every seed the 95% interval excludes zero; per-seed values are in Appendix Table 12.

of hint recovery on these problems roughly doubles the share of correct answers resting on false reasoning, and the one number the loop watches goes up.

Filtering a realistic pool. A practitioner faces the All-accepted pool, and the practical question is whether filtering it helps. Four sets of 1,587 solutions were drawn from it: every solution with no false claim (*Filtered*), the pool with the string match's flags removed (described below), an unfiltered sample from the Filtered set's problems, and an unfiltered sample from the whole pool (*Mixture*); each was trained with three seeds (Appendix Table 16). Filtering removes almost all of the inherited falsehood: 0.3 to 3.6% of the Filtered models' valid answers contain a false claim, against 56 to 70% for the Mixture models, a gap of +53.1 to +62.3 points with every interval excluding zero. Most of the gap comes from the filtering itself, since on the same problems unfiltered training leaves 34 to 51%. The string match, which needs no checker, leaves 11 to 22%. Three seeds leave the effect on accuracy unclear: the Filtered model's pass@1 is 6.6 points below the Mixture model's in two seeds and 8.5 above it in the third.

Answers come apart from reasoning. Across all Knights-and-Knaves solutions, wrong ones included, the Sound and Unsound models differ far less (65.2% against 76.0% with a false claim, first Qwen seed): the Sound model learned to let its mistakes show, so its answer tends to go wrong when its reasoning does. This is where the argument of Wen et al. (2026) applies. An outcome reward favors sound reasoning when sound reasoning reaches the right answer more often, and in the hinted data it does (90% against 29%). Training on unsound solutions removes this preference in Qwen: the gap between the rates at which sound and unsound held-out solutions are correct falls from +0.796 to -0.046 ± 0.042, and in Llama it varies widely across seeds (Appendix Table 17). The preference is enforced through the failures, which receive negative advantage, and self-training on accepted solutions keeps only the successes, so it never applies this correction. The verifier sees none of this. Sound and Unsound pass@1 differ by at most 2.3 points for Qwen, every interval including zero, and by −2.8 to +6.9 points for Llama, changing sign across seeds (Appendix K). Every training solution in both sets ends in a valid answer, so both models learn to produce valid answers, and what they learn about reaching them is the part the verifier does not check; Wen et al. (2026) and Xie et al. (2025a) likewise find that fine-tuning on flawed chains still improves accuracy. Appendix L gives two further views of this decoupling.

Soundness and solving in Sokoban. In one domain the difference reaches accuracy. The Sound model solves far more new Sokoban puzzles: $25.8 \pm 2.6\%$ against $8.1 \pm 0.5\%$ for Qwen, and $39.0 \pm 8.2\%$ against $22.0 \pm 6.6\%$ for Llama, with every interval excluding zero. Both were trained only on solutions that reach the goal, so the difference lies in whether the position claims along the way were true. A position claim is unforced in a finished solution and still guides the search that produces one: a model that has lost track of the box pushes it into dead ends, and in 35 to 38% of the Unsound Qwen model's failures a false position claim comes at or before the first fatal move, against 3 to 9% for the Sound model (Appendix M). In Blocksworld, where every move can be undone, the differences are at most about 2 points and mostly favor the Unsound model.

A signal that needs no checker. In Knights-and-Knaves a string match comes close to a checker: a hinted solution that declares "no contradiction" for an identity opposite to its hint. On the matched training pools it recovers 76.6% of Qwen's unsound solutions at 92.8% precision, measured on the same accepted solutions it was developed on. It does not transfer to Llama (precision 53.9%). An off-the-shelf process reward model, Qwen2.5-Math-PRM-7B (Zhang et al., 2025c), falls well short, with an AUROC of 0.74 in Knights-and-Knaves and 0.60 in Blocksworld; at the string match's 92.8% precision it flags one of 500 flawed Knights-and-Knaves solutions, and its lowest-scored step falls on the first false claim in under 1% of them. It was trained on mathematics and is used here outside that domain (Appendix N).

7 Validating the Measurements

The strongest of three LLM judges, GPT-6 Sol, agrees with the checker on every solution and line it read in Blocksworld (261 solutions) and Sokoban (175), and misses 1 of 84 false verdicts in Knights-and-Knaves; reading whole planning solutions, no judge flagged a claim the extractor had missed, and weaker judges disagree more, in both directions (Appendix G). Judging Knights-and-Knaves verdicts against the true solution leaves 67.3% of accepted hinted solutions with a false claim, and counting omitted facts as false in Blocksworld leaves the inheritance gap between $+19.1$ and $+44.1$ points (Appendix F). First steps with the logic flipped and the wording kept reach a correct answer in 0.5% of solutions, and 87.2% of their claims are false: the checker responds to the logic (Appendix P). Sampling at temperatures from 0.3 to 1.0 keeps the share of accepted hinted solutions with a false claim between 63.8% and 70.4% while recovering 313 to 557 problems (Appendix P). Reweighted to a common distribution of claim counts, the Knights-and-Knaves inheritance gap is $+69.3$ points for Qwen and $+42.5$ for Llama, and the wording of Sound and Unsound outputs differs no more than that of two seeds of one model (Appendix Q).

8 Conclusion and Limitations

Conclusion. An outcome verifier enforces the claims the answer depends on and cannot see the rest, and that is where false claims gather. Shown part of the answer, a model writes accepted solutions full of them, many contradicting the hint that got them accepted, while the moves and arithmetic the answer depends on stay correct; how the answer is given decides how much of this appears and whether any of it can be checked. Models trained on these solutions give correct answers that rest on false reasoning, with no consistent difference in accuracy in Knights-and-Knaves, and in Sokoban they solve fewer puzzles. In Qwen, outcome rewards push back through the failures, which self-training on accepted solutions discards; in Llama that push is weak and inconsistent.

The account says where to look: the claims a task's answer does not check can be identified before any model is trained, and where they can be checked, they should be. Filtering a realistic pool this way cuts the share of correct answers resting on a false claim from 56–70% to under 4%, with no clear effect on accuracy across three seeds. Where no exact checker exists, a strong LLM judge is the practical substitute, matching the checker line for line in both planning domains, and cheap signals such as the string match catch part of the problem.

Limitations. Exact checking is what makes these measurements trustworthy and also what bounds them: the claim-level results come from synthetic puzzles, and they concern the claims a solution writes down, which may differ from the computation behind them. The effect shrinks with size in two families without vanishing, on problems selected with one model. Training is supervised, with LoRA at 4B and 8B, and the reward analysis re-scores solutions without training a policy. The effect of form is isolated only in Knights-and-Knaves, and only there does the hint run on unsolved problems, as in a self-training loop; elsewhere the results describe hinted solutions in general. The link between soundness and solving appears in one domain of three. Filtering was tested for Qwen3-4B in Knights-and-Knaves only, and the string match was evaluated on the solutions it was developed on. In two further logic domains the picture is weaker: in ProntoQA the hint adds little over unaided solving, and in ZebraLogic the expected gap between hint and first steps was not confirmed (Appendix J). The largest open question is whether the split between forced and unforced claims can be drawn in open-ended domains, such as mathematical proofs or agentic tool use; the practical guidance above depends on it.

AI USE STATEMENT

In this work, we used generative AI tools for the following tasks with required disclosure: implementing methods, giving feedback on the methodology and experiments, refining hypotheses, including the framing of the forced/unforced account; and interpreting results. We did not use generative AI tools to formulate mathematical claims, prove them, or write proofs. Additionally, we used generative AI tools for tasks with recommended disclosure: drafting and editing the text of the paper, and searching for and summarizing related literature. Separately from this assistance, three LLMs serve as independent judges in the experiments; their use is described in Section 3 and Appendix G. We have reviewed all AI-assisted work. Every number in the paper was produced by code run on the stored model outputs, and reported values were checked against those outputs and recomputed where discrepancies appeared; every citation and attribution was checked against its source; and all text was read and revised by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 3 describes the domains, conditions, models, and measurements, and the appendix gives the remaining detail: puzzle generators, grading rules, and problem sets (Appendix A); full prompts, hint and first-steps templates, and a rendered prompt for every domain (Appendix D); sampling settings, training hyperparameters, and step counts for every run (Appendix E); claim extraction and the definitions of a false claim (Appendix F); judge prompts and model identifiers (Appendix G); and the reward re-scoring (Appendix H). All puzzles are generated from fixed seeds, all sampling uses a fixed seed, and all intervals come from stated bootstrap procedures. Code for generating the puzzles, sampling, checking, training, and analysis, together with the generated problem sets, can be found at `https://github.com/anon-rawr-iclr/right-answer-wrong-reasoning`.

REFERENCES

Iván Arcuschin, Jett Janiak, Robert Krzyzanowski, Senthooran Rajamanoharan, Neel Nanda, and Arthur Conmy. Chain-of-thought reasoning in the wild is not always faithful. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2503.08679.

Luoxin Chen, Yichi Zhou, and Huishuai Zhang. Learning to generate formally verifiable step-by-step logic reasoning via structured formal intermediaries. *arXiv preprint arXiv:2603.29500*, 2026.

Yanda Chen et al. Reasoning models don't always say what they think. *arXiv preprint arXiv:2505.05410*, 2025.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, et al. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Yuyang Ding, Chi Zhang, Juntao Li, Haibin Lin, and Min Zhang. FAPO: Flawed-aware policy optimization for efficient and reliable reasoning. *arXiv preprint arXiv:2510.22543*, 2025.

Kanishk Gandhi, Denise Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and Noah D. Goodman. Stream of search (SoS): Learning to search in language. In *Conference on Language Modeling (COLM)*, 2024.

Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Lukas Helff, Quentin Delfosse, David Steinmann, Ruben Härle, Hikaru Shindo, Patrick Schramowski, Wolfgang Stammer, Kristian Kersting, and Felix Friedrich. Llms gaming verifiers: Rlvr can lead to reward hacking. 2026. URL `https://arxiv.org/abs/2604.15149`.

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 *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

Zeyu Huang, Tianhao Cheng, Zihan Qiu, Zili Wang, Yinghui Xu, Edoardo M. Ponti, and Ivan Titov. Blending supervised and reinforcement fine-tuning with prefix sampling. *arXiv preprint arXiv:2507.01679*, 2025.

Zhaoyang Jiang, Xuanqi Peng, Fei Teng, Zhizhong Fu, Yunsoo Kim, Jiacong Mi, Zicheng Li, and Honghan Wu. Better accuracies, worse reasoning: A step-level audit of medical chain-of-thought distillation. *arXiv preprint arXiv:2605.28301*, 2026.

P. N. Johnson-Laird and Ruth M. J. Byrne. Meta-logical problems: Knights, knaves, and Rips. *Cognition*, 36(1):69–84, 1990.

Najoung Kim and Sebastian Schuster. Entity tracking in language models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Jungseob Lee, Seungyoon Lee, Suhyune Son, Dongyub Jude Lee, Sungbin Han, Sugyeong Eo, and Heuiseok Lim. Answer-conditioned chains of thought degrade verifiable-reasoning distillation in large language models. *arXiv preprint arXiv:2607.14552*, 2026.

Lucas Lehnert, Sainbayar Sukhbaatar, DiJia Su, Qinqing Zheng, Paul Mcvay, Michael Rabbat, and Yuandong Tian. Beyond a*: Better planning with transformers via search dynamics bootstrapping. 2024. URL https://arxiv.org/abs/2402.14083.

Jiazheng Li, Hong Lu, Kaiyue Wen, Zaiwen Yang, Jiaxuan Gao, Hongzhou Lin, Yi Wu, and Jingzhao Zhang. QuestA: Expanding reasoning capacity in LLMs via question augmentation. *arXiv preprint arXiv:2507.13266*, 2025.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *International Conference on Learning Representations (ICLR)*, 2023.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024.

Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. ZebraLogic: On the scaling limits of LLMs for logical reasoning. In *International Conference on Machine Learning (ICML)*, volume 267 of *PMLR*, pp. 37889–37905, 2025. arXiv:2502.01100.

Mingyang Liu, Gabriele Farina, and Asuman Ozdaglar. UFT: Unifying supervised and reinforcement fine-tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2505.16984.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

OpenAI. Introducing GPT-6 sol and luna. https://openai.com/index/introducing-gpt-6-sol-and-luna/, 2026.

Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. ReCEval: Evaluating reasoning chains via correctness and informativeness. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 10066–10086, 2023.

Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *International Conference on Learning Representations (ICLR)*, 2023.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh, and Luke Zettlemoyer. Spurious rewards: Rethinking training signals in RLVR. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2506.10947.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, et al. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024. arXiv:2312.06585.

Raymond M. Smullyan. *What Is the Name of This Book? The Riddle of Dracula and Other Logical Puzzles*. Prentice-Hall, 1978.

Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. PRMBench: A fine-grained and challenging benchmark for process-level reward models. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 25299–25346, 2025. arXiv:2501.03124.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Shubham Toshniwal, Sam Wiseman, Karen Livescu, and Kevin Gimpel. Chess as a testbed for language model state tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 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 *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. PlanBench: An extensible benchmark for evaluating large language models on planning and reasoning about change. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2023.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jGbRWwIidy.

Jinyang Wu, Chonghua Liao, Mingkuan Feng, Shuai Zhang, Zhengqi Wen, Haoran Luo, Ling Yang, Huazhe Xu, and Jianhua Tao. Templaterl: Structured template-guided reinforcement learning for llm reasoning. 2026. URL https://arxiv.org/abs/2505.15692.

Zhiheng Xi, Wenxiang Chen, Boyang Hong, Senjie Jin, Rui Zheng, et al. Training large language models for reasoning through reverse curriculum reinforcement learning. In *International Conference on Machine Learning (ICML)*, volume 235 of *PMLR*, pp. 54030–54048, 2024. arXiv:2402.05808.

Chulin Xie, Yangsibo Huang, Chiyuan Zhang, Da Yu, Xinyun Chen, Bill Yuchen Lin, Bo Li, Badih Ghazi, and Ravi Kumar. On memorization of large language models in logical reasoning. In *Proceedings of the International Joint Conference on Natural Language Processing and Asia-Pacific Chapter of the Association for Computational Linguistics (IJCNLP-AACL)*, pp. 2742–2785, 2025a. arXiv:2410.23123.

Tian Xie et al. Logic-RL: Unleashing LLM reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2502.14768*, 2025b.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2504.14945.

An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Chenlu Ye, Zhou Yu, Ziji Zhang, Hao Chen, Narayanan Sadagopan, Jing Huang, Tong Zhang, and Anurag Beniwal. Beyond correctness: Harmonizing process and outcome rewards through RL training. *arXiv preprint arXiv:2509.03403*, 2025.

Qinan Yu, Alexa R. Tartaglini, Peter Hase, Carlos Guestrin, and Christopher Potts. Outcome rewards do not guarantee verifiable or causally important reasoning. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2604.22074.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2503.14476.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2504.13837.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Kaiyi Zhang, Ang Lv, Jinpeng Li, Yongbo Wang, Feng Wang, Haoyuan Hu, and Rui Yan. StepHint: Multi-level stepwise hints enhance reinforcement learning to reason. *arXiv preprint arXiv:2507.02841*, 2025a.

Xichen Zhang, Sitong Wu, Yinghao Zhu, Haoru Tan, Shaozuo Yu, Ziyi He, and Jiaya Jia. Scaf-GRPO: Scaffolded group relative policy optimization for enhancing LLM reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=bOwVr0yr7r`.

Xuechen Zhang, Zijian Huang, Yingcong Li, Chenshun Ni, Jiasi Chen, and Samet Oymak. BREAD: Branched rollouts from expert anchors bridge SFT & RL for reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025b. arXiv:2506.17211.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*, 2025c.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. ProcessBench: Identifying process errors in mathematical reasoning. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025. arXiv:2412.06559.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in LLM reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2506.01347.

A DOMAINS, GENERATORS, AND GRADING

Knights-and-Knaves generator. Each person's statement is either a single claim about someone's identity ("X is a knight" or "X is a knave", possibly about the speaker) or the *and* or *or* of two such claims, with no negation or nesting. Statements are sampled to be consistent with a random assignment of identities, and a puzzle is kept only if exactly one assignment satisfies it, checked by enumerating all assignments. Table 3 compares the generator with that of Xie et al. (2025a). In-distribution puzzles have 3, 4, and 5 people (664, 722, and 614 for training; 361, 326, and 313 held out); out-of-distribution puzzles have 6, 7, and 8 (322, 336, and 342).

Table 3: **Knights-and-Knaves generators compared.**

	This paper	Xie et al. (2025a)
Statements	one atom, or *and*/*or* of two atoms; no negation or nesting	trees over and, or, not, imply, equivalence; depth ≤ 2, width ≤ 2
Uniqueness	exactly one satisfying assignment, by enumeration	unique solution via a solver
Search space	$4m^2$ statements per speaker; $(4m^2)^m$ configurations	about 10^{24} for 8 people
Names	chosen to differ from the public dataset	—
Sizes	3–5 people (train and held out), 6–8 (out of distribution)	2–8 people; 1,000 training puzzles per size (200 for 2)

Blocksworld. Problems have 4 blocks with random start and goal arrangements. The reference plan is the shortest found by breadth-first search, 4 to 8 actions long in the reported set. Each solution line has the form `<action>. Now: <state>`, with actions `unstack A from C`, `put down A`, `pick up B`, and `stack B on A`, and the state given as comma-separated facts (`A in hand`, `B on table`, `C on B`, `C clear`).

Sokoban. Puzzles are 6×6 grids with walls only on the border and one box, generated backwards from a solved state by 3 to 6 random player steps and pulls, so every puzzle is solvable. The reference plan is the reversed construction, 3 to 6 moves, and not necessarily the shortest. Each line has the form `push <dir>` or `move <dir>`, followed by `Now: player (r,c), boxes (r,c), k of M on goal`. Two harder settings, one with two boxes and one with a single box and constructions of 6 to 10 steps, fall outside the inclusion rules below and are not used.

Countdown. Each problem has 4 numbers drawn from 1 to 25, combined by three random operations among $+, -, \times, \div$ with every intermediate result a positive integer. The final result is the target, which ranges from 2 to 1,000 (median 148), so every problem has a solution by construction.

Grading. A Knights-and-Knaves solution is correct if and only if its first "Final Answer" block assigns every person, each person's role taken from their first mention in that block, and the assignment matches the unique solution. The block ends at the first code fence, markdown heading, or new puzzle; a missing marker or an unassigned person makes the solution incorrect. A Blocksworld or Sokoban plan is correct if it reaches the goal in the simulator. A Countdown expression is correct if it evaluates to the target using the given numbers.

Problem sets. Hint conditions in Knights-and-Knaves are run on the training puzzles the round-one model never solves in 32 attempts: 694 for Qwen3-4B and 1,027 for Llama-3.1-8B (their base models leave 1,185 and 1,538 unsolved). The Qwen set was drawn with an earlier, more lenient answer check; all 694 are also unsolved under the strict check used for every reported rate, which leaves 139 further puzzles unsolved (833 in total) that are not in the set. The shared set of 523 problems used across models consists of those among the 694 that the answer hint recovered at least once under the lenient check; under the strict check the hint recovers 442. Qwen3.5-4B's paired comparison uses the 391 problems on which both of its conditions produce accepted solutions. In the other domains every condition is run on all generated problems. Unaided, the shared set is hard for every model measured on it: pass@1 ranges from 0.1% (Llama-3.1-8B) to 2.2% (OLMo-3-32B), and each model leaves between 70.7% and 98.7% of the problems unsolved in 32 attempts (Table 6).

The full unsolved set. Because the 694 problems were selected with the lenient check, the hint and first-steps conditions were rerun for the round-one Qwen3-4B model on all 833 training puzzles it never solves under the strict check, with the same sampling settings and a separate sampling script. Table 4 gives the result. The original 694 problems reproduce their rate (68.3%, against 67.1% in

Table 1), and the 139 added problems behave the same way. Weighted equally between knight-first and knave-first problems, the hint rate would be about 47%.

Table 4: **Knights-and-Knaves on all strictly unsolved problems.** Round-one Qwen3-4B; share of accepted solutions with a false claim (per solution, local definition). Hint − first steps: paired-problem bootstrap, with the number of problems on which both conditions produce accepted solutions. Knight-first and knave-first: whether the first person named is a knight.

Problem set	Problems	Recovered (hint / first steps)	Hint	First steps	Hint − first steps	Paired
All strictly unsolved	833	558 / 638	69.0%	0.60%	+67.1 [+62.7, +71.5]	470
Original set	694	453 / 525	68.3%	0.52%	+66.4 [+61.2, +71.4]	381
Added by the strict check	139	105 / 113	71.5%	0.90%	+69.6 [+59.6, +79.1]	89
Knight-first	169	92 / 111	10.0%	0.84%	+7.7 [+3.8, +12.6]	78
Knave-first	664	466 / 527	84.0%	0.54%	+82.7 [+79.6, +85.6]	392

Which models enter which domain. A model is used in a domain only if its unaided pass@1 lies between 5% and 90% and no more than 20% of its solutions contain no checkable claims. In Sokoban the second threshold was initially 5% and was raised to 20% before the full run. These rules exclude Llama-3.1-8B from Countdown (unaided pass@1 4.4%), OLMo-3-7B from Blocksworld (3.9%) and Sokoban (1.6%), the two harder Sokoban settings (the two-box setting at 0.0%), and ZebraLogic sizes other than 2×3 (0.0 to 4.9% across both models). Table 5 summarizes coverage.

Table 5: **Coverage.** Number of problems each model never solves in 32 unaided attempts, out of all generated problems. *Sound and Unsound models trained. ×Excluded by the inclusion rules. In Knights-and-Knaves the round-one models leave 694 (Qwen3-4B*) and 1,027 (Llama-3.1-8B*) training puzzles unsolved, and Qwen3-1.7B/8B/14B, Qwen3.5-2B/4B/9B, Llama-3.2-3B, and OLMo-3-7B/32B are measured on the shared 523-problem set. Judges: gpt-oss-120b and GPT-6 Sol in Knights-and-Knaves; all three in Blocksworld, Sokoban, and ProntoQA.

Domain (problems)	Qwen3-4B	Qwen3.5-4B	Qwen3-8B	Llama-3.1-8B	OLMo-3-7B	OLMo-3-32B
Blocksworld (300)	—	121	152*	202*	181×	100
Sokoban (300)	—	124	190*	237*	233×	114
Countdown (300)	89	102	77	150×	—	—
ProntoQA (300)	1	0	—	4	—	—
ZebraLogic (40, 2×3)	28	—	—	29	—	—

B Selection and evaluation criteria

A model is included in a domain when its unaided pass@1 lies between 5% and 90% and at most 20% of its solutions lack claims that can be checked (Appendix A), so that the domain has both solved and unsolved problems and the checker has something to check. An inheritance comparison counts as a result when the Unsound model exceeds the Sound model in every training seed, with each seed's 95% interval excluding zero; every comparison in Section 6 meets this. The forced/unforced account was developed on Knights-and-Knaves and a Countdown pilot, and Blocksworld and Sokoban were used to test it.

C Models across families and sizes

Table 6 gives the values behind Figure 2. The two Llama models come from different releases (Llama-3.2-3B and Llama-3.1-8B), so they are two points of one family, not a scaling series.

D Prompts and templates

Every prompt contains fixed worked examples followed by the problem, and ends where the model begins writing. Knights-and-Knaves uses four worked examples. Their statements come from a

Table 6: **Knights-and-Knaves across model families and sizes.** Base models on a shared set of 523 problems (§3), with two identities given as an answer hint or as first steps. Per solution: share of accepted solutions containing at least one false claim, and share that declare "no contradiction" for an identity opposite to the hint and never retract it. Brackets: 95% bootstrap intervals over problems for the hint rate and for the paired difference (hint − first steps, in percentage points, on problems where both conditions produce accepted solutions: 157 to 459 per model); binomial intervals over solutions, not adjusted for clustering within problems, for the never-retract rate. Unsolved unaided: share of the 523 problems the base model never solves in 32 attempts without help. *First steps measured on a 551-problem set; the difference uses the problems the two sets share. †Newer generation.

Model	Answer hint	First steps	Hint − first steps	Contradict hint, never retract	Unsolved unaided
Qwen3-1.7B	81.4% [78.3, 84.2]	6.7%	+72.7 [69.0, 76.1]*	83.1% [81.6, 84.5]	86.2%
Qwen3-4B	56.3% [48.7, 64.2]	0.9%	+57.8 [49.6, 65.7]	53.5% [51.4, 55.7]	97.3%
Qwen3-8B	37.8% [33.7, 42.1]	1.5%	+36.1 [32.0, 40.6]	29.8% [28.4, 31.2]	88.5%
Qwen3-14B	10.1% [8.5, 11.8]	0.9%	+9.4 [7.7, 11.0]	5.4% [4.9, 6.0]	83.6%
Llama-3.1-8B	97.8% [97.0, 98.5]	8.5%	+87.8 [83.9, 90.9]	95.1% [94.4, 95.7]	98.7%
Llama-3.2-3B	96.8% [95.7, 97.6]	6.0%	+91.6 [88.8, 93.9]	95.0% [94.2, 95.7]	94.3%
OLMo-3-7B	88.6% [86.5, 90.6]	0.3%	+89.1 [87.5, 90.8]*	87.2% [86.3, 88.1]	86.2%
OLMo-3-32B	30.9% [27.6, 34.3]	0.1%	+31.2 [28.0, 34.5]	28.7% [27.3, 30.1]	70.7%
Qwen3.5-2B†	36.2% [33.4, 39.2]	1.7%	+34.5 [31.5, 37.4]	32.6% [31.1, 34.1]	78.6%
Qwen3.5-4B†	19.4% [17.2, 21.8]	0.07%	+19.2 [17.1, 21.8]	16.3% [15.4, 17.3]	81.8%
Qwen3.5-9B†	22.4% [19.8, 25.5]	0.0%	+22.5 [19.6, 25.5]	16.0% [15.1, 17.0]	90.8%

richer grammar than the test puzzles, including negation, conditionals, and biconditionals, and their solutions add a clause to each verdict ("No contradiction is found in their claim that . . . "). The same worked examples appear under every condition, so they cannot account for the difference between the hint and the first steps; whether examples in the test grammar would change the rates themselves was not tested. Every other domain uses two worked examples, generated automatically in Blocksworld, Sokoban, Countdown, and ZebraLogic (ZebraLogic's from two 2×2 puzzles excluded from evaluation) and taken from the official generator in ProntoQA. Below, one prompt per domain is shown in the answer-hint condition, rendered with the same code that produced every sample.

Knights-and-Knaves, answer hint.

```
Puzzle:
A very special island is inhabited only by knights and knaves. Knights always tell the truth,
↪  and knaves always lie. You meet 2 inhabitants: Oliver, and Ethan. Oliver told you that
↪  Oliver is a knight or Ethan is a knave. In a statement by Ethan: "Oliver is a knight". So
↪  who is a knight and who is a knave?

Solution:
Let's think step by step, by considering whether each person is lying and if that leads to
↪  contradiction.
Assume Oliver is a knight. No contradiction is found in their claim that Oliver is a knight or
↪  Ethan is a knave.
Assume Ethan is a knight. No contradiction is found in their claim that Oliver is a knight.
This leads to a feasible solution.
Final Answer:
(1) Oliver is a knight
(2) Ethan is a knight

Puzzle:
A very special island is inhabited only by knights and knaves. Knights always tell the truth,
↪  and knaves always lie. You meet 2 inhabitants: Michael, and Jacob. Michael said, "Jacob is
↪  a knave." "Michael is a knave if and only if Michael is a knight," Jacob declared. So who
↪  is a knight and who is a knave?

Solution:
Let's think step by step, by considering whether each person is lying and if that leads to
↪  contradiction.
Assume Michael is a knight. No contradiction is found in their claim that Jacob is a knave.
Jacob cannot be a knight, because this would contradict the claim of Michael that Jacob is a
↪  knave.
Assume Jacob is a knave. No contradiction is found in their false claim that Michael is a
↪  knave if and only if Michael is a knight.
This leads to a feasible solution.
Final Answer:
(1) Michael is a knight
```

```
(2) Jacob is a knave

Puzzle:
A very special island is inhabited only by knights and knaves. Knights always tell the truth,
↪  and knaves always lie. You meet 3 inhabitants: Michael, Zoey, and Ethan. Michael was heard
↪  saying, "Ethan is a knight if and only if Michael is a knight". "Zoey is a knight or Ethan
↪  is a knight," Zoey mentioned. Ethan asserted: "Michael is a knave if and only if Zoey is a
↪  knave". So who is a knight and who is a knave?

Solution:
Let's think step by step, by considering whether each person is lying and if that leads to
↪  contradiction.
Assume Michael is a knight. No contradiction is found in their claim that Ethan is a knight if
↪  and only if Michael is a knight.
Assume Ethan is a knight. No contradiction is found in their claim that Michael is a knave if
↪  and only if Zoey is a knave.
Assume Zoey is a knight. No contradiction is found in their claim that Zoey is a knight or
↪  Ethan is a knight.
This leads to a feasible solution.
Final Answer:
(1) Michael is a knight
(2) Zoey is a knight
(3) Ethan is a knight

Puzzle:
A very special island is inhabited only by knights and knaves. Knights always tell the truth,
↪  and knaves always lie. You meet 3 inhabitants: Michael, Mason, and Jacob. Michael
↪  commented, "Mason is not a knight". Mason commented, "Michael is a knight and Jacob is a
↪  knave". Jacob stated, "If Michael is a knight then Jacob is a knight". So who is a knight
↪  and who is a knave?

Solution:
Let's think step by step, by considering whether each person is lying and if that leads to
↪  contradiction.
Assume Michael is a knight. No contradiction is found in their claim that Mason is not a
↪  knight.
Mason cannot be a knight, because this would contradict the claim of Michael that Mason is not
↪  a knight.
Assume Mason is a knave. No contradiction is found in their false claim that Michael is a
↪  knight and Jacob is a knave.
Assume Jacob is a knight. No contradiction is found in their claim that If Michael is a knight
↪  then Jacob is a knight.
This leads to a feasible solution.
Final Answer:
(1) Michael is a knight
(2) Mason is a knave
(3) Jacob is a knight

Puzzle:
A very special island is inhabited only by knights and knaves. Knights always tell the truth,
↪  and knaves always lie. You meet 5 inhabitants: Reginald, Cordelia, Cedric, Wallace, and
↪  Eugene. According to Reginald, "Cedric is a knight or Eugene is a knave". Cordelia was
↪  heard saying, "Wallace is a knight or Wallace is a knave". Cedric told you, "Cedric is a
↪  knight and Cedric is a knave". Wallace was heard saying, "Reginald is a knight and Wallace
↪  is a knave". According to Eugene, "Cedric is a knave or Eugene is a knave". So who is a
↪  knight and who is a knave? (Hint: Reginald is a knave, Cordelia is a knight.)

Solution:
```

Knights-and-Knaves, first steps. The worked examples are identical; the hint is removed and the solution opens with two supplied steps:

```
Solution:
Let's think step by step, by considering whether each person is lying and if that leads to
↪  contradiction.
Assume Reginald is a knave. No contradiction is found.
Assume Cordelia is a knight. No contradiction is found.
```

Blocksworld, answer hint.

```
Blocks: A, B, C, D
Start: A on C, B on A, B clear, C on table, D on table, D clear
Goal: A on table, A clear, B on table, C on table, C clear, D on B, D clear
Solution:
Move one block at a time, restating the world after each move.
unstack B from A. Now: A on C, A clear, B in hand, C on table, D on table, D clear.
put down B. Now: A on C, A clear, B on table, B clear, C on table, D on table, D clear.
unstack A from C. Now: A in hand, B on table, B clear, C on table, C clear, D on table, D
↪  clear.
```

```
put down A. Now: A on table, A clear, B on table, B clear, C on table, C clear, D on table, D
↪  clear.
pick up D. Now: A on table, A clear, B on table, B clear, C on table, C clear, D in hand.
stack D on B. Now: A on table, A clear, B on table, C on table, C clear, D on B, D clear.
Final Answer: unstack B from A; put down B; unstack A from C; put down A; pick up D; stack D
↪  on B

Blocks: A, B, C, D
Start: A on table, B on C, B clear, C on A, D on table, D clear
Goal: A on table, A clear, B on table, C on B, C clear, D on table, D clear
Solution:
Move one block at a time, restating the world after each move.
unstack B from C. Now: A on table, B in hand, C on A, C clear, D on table, D clear.
put down B. Now: A on table, B on table, B clear, C on A, C clear, D on table, D clear.
unstack C from A. Now: A on table, A clear, B on table, B clear, C in hand, D on table, D
↪  clear.
stack C on B. Now: A on table, A clear, B on table, C on B, C clear, D on table, D clear.
Final Answer: unstack B from C; put down B; unstack C from A; stack C on B

Blocks: A, B, C, D
Start: A on C, A clear, B on table, C on B, D on table, D clear
Goal: A on table, B on A, B clear, C on table, C clear, D on table, D clear
(Hint: one solution is unstack A from C; put down A; unstack C from B; put down C; pick up B;
↪  stack B on A.)
Solution:
Move one block at a time, restating the world after each move.
```

Sokoban, answer hint.

```
Grid: 6x6 with walls on the border.
Start: player (4,1), boxes (3,3), 0 of 1 on goal
Goals: (2,3)
Solution:
Move or push one step at a time, restating the positions after each step.
move right. Now: player (4,2), boxes (3,3), 0 of 1 on goal.
move right. Now: player (4,3), boxes (3,3), 0 of 1 on goal.
push up. Now: player (3,3), boxes (2,3), 1 of 1 on goal.
move down. Now: player (4,3), boxes (2,3), 1 of 1 on goal.
move left. Now: player (4,2), boxes (2,3), 1 of 1 on goal.
Final Answer: move right; move right; push up; move down; move left

Grid: 6x6 with walls on the border.
Start: player (1,2), boxes (2,1), 0 of 1 on goal
Goals: (3,1)
Solution:
Move or push one step at a time, restating the positions after each step.
move left. Now: player (1,1), boxes (2,1), 0 of 1 on goal.
push down. Now: player (2,1), boxes (3,1), 1 of 1 on goal.
move right. Now: player (2,2), boxes (3,1), 1 of 1 on goal.
move down. Now: player (3,2), boxes (3,1), 1 of 1 on goal.
Final Answer: move left; push down; move right; move down

Grid: 6x6 with walls on the border.
Start: player (2,3), boxes (3,3), 0 of 1 on goal
Goals: (4,3)
(Hint: one solution is push down; move up; move right; move down; move up.)
Solution:
Move or push one step at a time, restating the positions after each step.
```

Countdown, answer hint.

```
Numbers: 13, 25, 14, 2
Target: 533
Solution:
Combine the numbers one operation at a time.
14 + 2 = 16
25 + 16 = 41
41 * 13 = 533
Final Answer: ((25 + (14 + 2)) * 13)

Numbers: 18, 23, 20, 5
Target: 153
Solution:
Combine the numbers one operation at a time.
20 + 18 = 38
5 * 23 = 115
38 + 115 = 153
Final Answer: ((20 + 18) + (5 * 23))

Numbers: 2, 18, 1, 3
Target: 38
```

```
(Hint: one solution is ((3 - 1) + (18 * 2)).)
Solution:
Combine the numbers one operation at a time.
```

ProntoQA, answer hint.

```
Question: Every wumpus is a dumpus. Lempuses are liquid. Every brimpus is not red. Tumpuses
↪  are lempuses. Every impus is not sunny. Each lorpus is a brimpus. Tumpuses are shumpuses.
↪  Every shumpus is temperate. Each rompus is moderate. Every rompus is a wumpus. Each
↪  shumpus is a gorpus. Zumpuses are yumpuses. Every wumpus is not nervous. Gorpuses are
↪  floral. Each grimpus is nervous. Wumpuses are numpuses. Shumpuses are rompuses. Zumpuses
↪  are opaque. Lorpuses are tumpuses. Tumpuses are loud. Rompuses are impuses. Dumpuses are
↪  angry. Lorpuses are dull. Max is a zumpus. Max is a lorpus.
True or false: Max is nervous.
Answer: Max is a lorpus. Lorpuses are tumpuses. Max is a tumpus. Tumpuses are shumpuses. Max
↪  is a shumpus. Shumpuses are rompuses. Max is a rompus. Every rompus is a wumpus. Max is a
↪  wumpus. Every wumpus is not nervous. Max is not nervous.
Final Answer: False

Question: Lempuses are dumpuses. Yumpuses are wumpuses. Wumpuses are sterpuses. Each yumpus is
↪  cold. Every wumpus is wooden. Gorpuses are lempuses. Brimpuses are not melodic. Yumpuses
↪  are brimpuses. Each grimpus is opaque. Every zumpus is a tumpus. Every vumpus is dull.
↪  Wumpuses are vumpuses. Shumpuses are not dull. Every rompus is red. Vumpuses are
↪  grimpuses. Every zumpus is sour. Every dumpus is kind. Each gorpus is sunny. Sterpuses are
↪  moderate. Every lempus is not large. Each gorpus is a rompus. Vumpuses are lorpuses. Each
↪  lempus is a yumpus. Fae is a zumpus. Fae is a gorpus.
True or false: Fae is not dull.
Answer: Fae is a gorpus. Gorpuses are lempuses. Fae is a lempus. Each lempus is a yumpus. Fae
↪  is a yumpus. Yumpuses are wumpuses. Fae is a wumpus. Wumpuses are vumpuses. Fae is a
↪  vumpus. Every vumpus is dull. Fae is dull.
Final Answer: False

Question: Rompuses are fruity. Dumpuses are sterpuses. Numpuses are angry. Shumpuses are
↪  opaque. Wumpuses are not metallic. Brimpuses are grimpuses. Every grimpus is a jompus.
↪  Every grimpus is a dumpus. Brimpuses are happy. Rompuses are gorpuses. Gorpuses are
↪  brimpuses. Every grimpus is blue. Dumpuses are rainy. Every brimpus is a lempus. Gorpuses
↪  are moderate. Each rompus is a vumpus. Vumpuses are muffled. Each lempus is not temperate.
↪  Dumpuses are numpuses. Each jompus is not sweet. Each shumpus is a zumpus. Gorpuses are
↪  wumpuses. Yumpuses are not rainy. Rex is a rompus. Rex is a shumpus.
True or false: Rex is rainy. (Hint: the answer is True.)
Answer:
```

ZebraLogic, answer hint.

```
Puzzle:
There are 2 houses, numbered 1 to 2 from left to right, as seen from across the street. Each
↪  house is occupied by a different person. Each house has a unique attribute for each of the
↪  following characteristics:
 - Each person has a unique name: `Eric`, `Arnold`
 - Each person has a unique type of pet: `dog`, `cat`

## Clues:
1. Eric is somewhere to the left of Arnold.
2. The person who owns a dog is not in the first house.


Solution:
Let's test assignments one at a time and check each against the clues.
Suppose House 1 Name = Arnold. This contradicts the clues.
Suppose House 1 Name = Eric. No contradiction is found.
Suppose House 1 Pet = cat. No contradiction is found.
Suppose House 2 Name = Arnold. No contradiction is found.
Suppose House 2 Pet = cat. This contradicts the clues.
Suppose House 2 Pet = dog. No contradiction is found.
Final Answer:
House | Name | Pet
1 | Eric | cat
2 | Arnold | dog

Puzzle:
There are 2 houses, numbered 1 to 2 from left to right, as seen from across the street. Each
↪  house is occupied by a different person. Each house has a unique attribute for each of the
↪  following characteristics:
 - Each person has a unique name: `Arnold`, `Eric`
 - People have unique heights: `short`, `very short`

## Clues:
1. Arnold is the person who is very short.
2. Eric is not in the second house.
```

```
Solution:
Let's test assignments one at a time and check each against the clues.
Suppose House 1 Name = Arnold. This contradicts the clues.
Suppose House 1 Name = Eric. No contradiction is found.
Suppose House 1 Height = short. No contradiction is found.
Suppose House 2 Name = Arnold. No contradiction is found.
Suppose House 2 Height = short. This contradicts the clues.
Suppose House 2 Height = very short. No contradiction is found.
Final Answer:
House | Name | Height
1 | Eric | short
2 | Arnold | very short

Puzzle:
There are 2 houses, numbered 1 to 2 from left to right, as seen from across the street. Each
↪  house is occupied by a different person. Each house has a unique attribute for each of the
↪  following characteristics:
 - Each person has a unique name: `Arnold`, `Eric`
 - Each person lives in a unique style of house: `colonial`, `victorian`
 - People use unique phone models: `iphone 13`, `samsung galaxy s21`

## Clues:
1. Eric is somewhere to the left of the person who uses an iPhone 13.
2. The person who uses a Samsung Galaxy S21 is the person residing in a Victorian house.

(Hint: House 1 Name is Eric; House 1 HouseStyle is victorian.)

Solution:
Let's test assignments one at a time and check each against the clues.
```

First steps in the other domains. The first steps are the first two lines of the reference solution, written in the same format as the worked examples. For the Blocksworld problem above:

```
unstack A from C. Now: A in hand, B on table, C on B, C clear, D on table, D clear.
put down A. Now: A on table, A clear, B on table, C on B, C clear, D on table, D clear.
```

For the Sokoban problem, `push down. Now: player (3,3), boxes (4,3), 1 of 1 on goal.` and `move up. Now: player (2,3), boxes (4,3), 1 of 1 on goal.`; for the Countdown problem, `3 - 1 = 2` and `18 * 2 = 36`. In ProntoQA they are the first two gold chain steps (`Rex is a rompus. Rompuses are gorpuses.`), and in ZebraLogic the two hinted cells stated as hypotheses (`Suppose House 1 Name = Eric. No contradiction is found.`, and likewise for the second cell).

Six phrasings. The phrasings of Appendix I give the same two Knights-and-Knaves identities in the opening of the solution as "Assume X is a knight…" (the first steps), "Suppose X is a knight…", "If X were a knight…", "X is a knight", "It is established that X is a knight", and "Checking X, we find X is a knight".

References to the hint. Every accepted hinted solution was scanned for phrases that refer back to the hint, such as "given", "we are told", and "the answer is". The scan finds none in the 4,830 accepted Knights-and-Knaves hint solutions; none in Blocksworld (9,410 Qwen3-8B, 8,454 Llama-3.1-8B, 9,455 OLMo-3-32B), Sokoban (8,861, 7,500, 9,229), or Countdown (8,456 Qwen3-4B, 8,384 Qwen3-8B); and none in the eight Blocksworld and Sokoban training sets. In ProntoQA it finds 5 of 6,921 (Qwen) and 16 of 9,453 (Llama), which are left in. For Qwen3.5-4B it finds none of 9,242 in Blocksworld, 1 of 8,030 in Sokoban, 4 of 9,506 in Countdown, and 2 of 6,144 in ProntoQA, which are likewise left in.

E SAMPLING, TRAINING, AND COMPUTE

Sampling. Every condition draws 32 samples per problem at temperature 0.6 and top-p 0.95 with sampling seed 0 (seed 1 for the round-one Knights-and-Knaves unaided pass), using vLLM 0.29.0 in BF16 on H200 GPUs. The limit on new tokens is 512 in Countdown, 768 in Blocksworld and Sokoban, 1,024 in Knights-and-Knaves and ProntoQA, and 3,072 in ZebraLogic; the context length is 4,096, except 2,048 in Countdown and 8,192 in ZebraLogic. Generation stops at the next problem's header, a markdown heading, or a code fence. Evaluations ran on two clusters; an environment check across them agreed to within 0.4 points, and the same adapter gave out-of-distribution pass@32 of 68.2 and 68.6 on the two.

Training. All runs use LoRA with rank 16 and $\alpha = 16$ on the query, key, value, and output projections, with 4 examples per step (batch 1, gradient accumulation 4) and AdamW. The round-one models are trained on one accepted solution per solved training puzzle (815 for Qwen3-4B, 462 for Llama-3.1-8B) with learning rate 4×10^{-4} and gradient clipping 1.0 for 300 steps, with 11.8M and 13.6M trainable parameters. Table 7 gives the Sound, Unsound, and All-accepted runs; Sound and Unsound models of the same domain, family, and seed share every setting, and step counts give about 5.3 passes over each Sound and Unsound set (1.3 at the lower dose).

Table 7: **Inheritance training runs.** Solutions and problems per training set; steps at the full dose (lower dose in parentheses).

Domain	Model	Solutions	Problems	Learning rate	Clipping	Steps
Knights-and-Knaves	Qwen3-4B	372	118	4×10^{-4}	1.0	496 (124)
	Llama-3.1-8B	128	71	4×10^{-4}	1.0	171
	Qwen3-4B, All-accepted	4,830	442	4×10^{-4}	1.0	496
Blocksworld	Qwen3-8B	831	150	2×10^{-4}	0.5	1,100 (275)
	Llama-3.1-8B	1,542	245	2×10^{-4}	0.5	2,050
Sokoban	Qwen3-8B	245	63	2×10^{-4}	0.5	325 (81)
	Llama-3.1-8B	499	149	2×10^{-4}	0.5	661

Knights-and-Knaves models start from the round-one model; planning models start from base checkpoints. The All-accepted model trains for the same number of steps as Qwen's Sound and Unsound models, about 0.41 passes over its larger set.

F CLAIM EXTRACTION AND DEFINITIONS

Knights-and-Knaves. A claim is an assumption of the form "Assume X is a knight" or "Assume X is a knave", together with the last verdict phrase before the next assumption. Verdicts are "no contradiction" or a contradiction phrase such as "leads to a contradiction", "is a contradiction", "contradicts", or "cannot be both". Assumptions with no verdict are not claims; only text before the "Final Answer" marker is checked; claims supplied as first steps are excluded; and a verdict that reverses an earlier one on the same assumption is a self-correction, with the later verdict scored. The default extractor parses the "Assume" form; an extended version, used only for the phrasings of Appendix I, also parses "Suppose" and "If X were", with verdicts such as "consistent" and "inconsistent", and reproduces the default extractor's rates wherever both apply.

A verdict of "no contradiction" is judged in three ways. Under the *local* definition (the default), it is false if some speaker whose role is fixed by the live assumptions makes a fully determined statement that contradicts that role. Under the *global* definition, it is correct only if the assumed identity matches the solution and every assignment made so far, retracted ones included, extends to the full solution. The *single-step* definition applies the local check to the newly assumed person only. Local and global can disagree in either direction; a solution that rejects a hypothesis that is in fact false before reaching a contradiction is right globally and wrong locally, which is common in unaided search. Table 8 gives all three.

Table 8: **Knights-and-Knaves false claims under three definitions**, same samples as Table 1. Each cell: % of claims false (per claim) / % of accepted solutions with at least one false claim (per solution).

	Global	Local (default)	Single-step
Answer hint	61.4/67.3	34.6/67.1	6.3/21.8
First steps	0.2/0.4	0.2/0.4	0.2/0.3
Unaided	16.8/23.6	17.1/28.2	7.6/24.0

Blocksworld and Sokoban. Two kinds of claim are checked separately: each action or move for legality in the current simulated state, and each restated state against the simulator. A Blocksworld state description is *asserted-false* if it states a fact that is not true, and *incomplete* if it leaves out a true

one; the main text counts asserted-false descriptions only. Sokoban position statements (player and box coordinates) must match exactly, so an incomplete one is already false, and progress statements ("k of M on goal") are checked against the true state. Among valid plans, forced claims are almost never false; over all solutions, illegal moves are common (15 to 86% of solutions), because after one illegal move the rest of the plan starts from a state that does not exist.

Omissions. Table 9 gives Blocksworld rates with omissions counted as false as well. Hinted solutions leave out more of the state, so the gap between hint and first steps grows; the Sound models omit slightly more than the Unsound ones, so the inheritance gap shrinks slightly, but every interval stays above zero. Sound models' rates are 0.0 to 7.1% without omissions and 1.8 to 18.6% with them. These intervals come from a separate bootstrap run and differ slightly from those in Table 12.

Table 9: **Blocksworld with and without omissions** (per claim, paired-problem bootstrap). Top: hint minus first steps, accepted solutions (rates in parentheses). Bottom: Unsound minus Sound on held-out problems, valid plans, mean ± standard deviation over three training seeds; in every seed and under both definitions, the 95% interval excludes zero. Rates are on the paired problems and differ slightly from Table 2, which uses all problems.

	Paired problems	Asserted-false only	Including omissions
Qwen3-8B, hint − first steps	173	+13.4 [+8.3, +18.2] (39.6 vs 26.2%)	+19.7 [+14.5, +24.6] (50.8 vs 31.1%)
Llama-3.1-8B, hint − first steps	153	+6.1 [+1.3, +9.7] (17.6 vs 11.5%)	+22.4 [+16.8, +27.0] (39.4 vs 17.0%)
OLMo-3-32B, hint − first steps	213	+2.5 [−1.9, +7.3] (29.8 vs 27.3%)	+8.3 [+3.1, +13.2] (44.3 vs 36.0%)
Qwen3-8B, Unsound − Sound	53–72	+31.2 ± 12.7	+28.9 ± 13.2
Llama-3.1-8B, Unsound − Sound	63–66	+26.6 ± 3.8	+23.5 ± 4.1
Qwen3-8B, lower dose (one seed)	77	+28.6	+18.7

Countdown. Each arithmetic line $a \circ b = c$ is a claim, false if its arithmetic is wrong or either operand is not among the available values: the starting numbers not yet used and the results produced so far.

G INDEPENDENT JUDGES

Each judge receives a single user message, with no system prompt, through the OpenRouter chat-completions API at temperature 0 with at most 2,000 output tokens. The model identifiers were `openai/gpt-oss-120b`, `openai/gpt-6-sol`, and `deepseek/deepseek-v4-pro` (DeepSeek V4 Pro 0423).

Whole-solution mode. In Blocksworld and Sokoban, the judge reads a full solution, with its lines numbered, and lists every line that asserts something false; this also lets it flag lines the extractor never captured, and in no Blocksworld or Sokoban run did it flag such a line. Up to two hinted solutions per problem are eligible, if they contain at least three checked lines. They are shuffled with seed 0, and equal numbers with and without a false line are drawn. For GPT-6 Sol, the samples were extended to 150 solutions with a false line in each planning domain and every eligible clean solution: 111 in Blocksworld and 25 in Sokoban, where clean hinted solutions are rare. Every reply parsed. The one-sided 95% upper bound on its rate of missed false solutions is 2.0% in each domain; on flagged clean solutions it is 2.7% in Blocksworld and 11.3% in Sokoban. Lines are shown up to the first illegal move, and in Sokoban the checker's verdict on a line includes the progress count as well as the positions. The message is the domain's rules, the problem statement, the numbered lines, and a request for JSON. For Blocksworld:

```
You are checking a blocks-world solution. Blocks rest on the table or on another block; a
↪  block is "clear" if nothing is on it and it is not held; the hand holds at most one block.
↪  Each numbered line gives a move and then the model's restatement of the world after that
↪  move. Decide, for EACH line, whether every fact the line asserts is true of the world at
↪  that point, given the start state and the moves so far. Ignore facts the line leaves out:
↪  omission is not an error.

<problem statement>

THE LINES, IN ORDER:
```

```
1. <line>
2. <line>
...

Reply with ONLY a JSON object:
{"false_lines": [1-based line numbers whose assertions are false]}
Use an empty list if every line is true. No other text.
```

For Sokoban, the rules read:

```
You are checking a Sokoban solution on a grid with walls on the border. Coordinates are
↪  (row,column). "move <dir>" steps the player; "push <dir>" steps the player into a box and
↪  the box one further. Each numbered line gives a step and then the model's restatement of
↪  the positions after it. Decide, for EACH line, whether every position it asserts (player,
↪  boxes, and the count of boxes on goals) is true at that point, given the start state and
↪  the steps so far. Ignore omissions.
```

Per-claim mode. The judge scores one claim at a time, from samples of 75 claims the checker marks false and 75 it marks true. Each message is the domain's rules, a blank line, the claim's context, and a request for a single verdict. The context is the state before the move, the move, and the model's restatement of the state after it (Blocksworld and Sokoban), or the rules, facts, and proof so far (ProntoQA). The rules for each domain:

```
[Blocksworld]
You are checking one line of a blocks-world solution. Blocks rest on the table or on another
↪  block. A block is "clear" if nothing is on it and it is not being held. The hand holds at
↪  most one block. You are given the state BEFORE the move, the move, and the model's
↪  restatement of the state AFTER it. Decide whether EVERY fact the model asserts is true of
↪  the state after the move. Ignore facts the model leaves out: omission is not an error.
↪  Only assertions that are false count.

[Sokoban]
You are checking one line of a Sokoban solution on a grid with walls on the border.
↪  Coordinates are (row,column). "move <dir>" steps the player; "push <dir>" steps the player
↪  into a box and the box one further. You are given the state BEFORE the step, the step, and
↪  the model's restatement of the positions after it. Decide whether EVERY position the model
↪  asserts (player, boxes, and the count of boxes on goals) is true of the state after the
↪  step. Ignore anything the model leaves out.

[ProntoQA]
You are checking one step of a deductive proof over the given rules and facts. A step asserts
↪  a fact about the entity. Decide whether the asserted fact follows in ONE step from the
↪  rules together with the facts established earlier in the proof (including the axioms). A
↪  step is WRONG if the fact does not follow that way -- for example if no rule licenses it,
↪  or the rule it would need is not in the list.
```

Each ends with the request: `Reply with ONLY a JSON object: {"assertion_true": true or false} No other text.`

Knights-and-Knaves. The judge receives the puzzle and the solution's hypothesis lines as extracted, numbered in order, and is asked to apply the local definition; it therefore checks the checker's verdicts; the extractor's recall is checked in the planning domains only. Solutions with at least two hypothesis lines are drawn with seed 0, separately from those the checker marks false and those it marks clean. The message is:

```
You are checking a solution to a knights-and-knaves puzzle.
Knights always tell the truth; knaves always lie.

The solution contains lines of the form:
  "Assume <Name> is a <knight|knave>. No contradiction is found."
or
  "Assume <Name> is a <knight|knave>. This leads to a contradiction."

For EACH such line, decide whether the stated verdict is correct GIVEN THE
ASSUMPTIONS ALREADY IN FORCE at that point in the solution. A line is WRONG
if it says "no contradiction is found" but the assumptions so far (including
this one) already violate some speaker's statement, or if it says "this leads
to a contradiction" when the assumptions so far are still consistent.

Reply with ONLY a JSON object:
{"any_wrong": true or false, "wrong_lines": [list of 1-based line numbers of
the Assume-lines you judge wrong]}
No other text.

PUZZLE:
```

```
<puzzle>

THE ASSUME-LINES, IN ORDER:
1. <line>
2. <line>
...
```

Given the same 80 solutions, DeepSeek-V4-Pro misses none of the 36 false solutions whose replies parsed and 2 of their 75 false claims, but flags 12 of the 40 clean solutions (43 of their 159 claims). In clean solutions the local and global readings agree, so these flags are errors under either reading; GPT-6 Sol flags 1 of the same 159 claims. Its replies failed to parse for 4 of 80 Knights-and-Knaves solutions, 6 of 40 in Blocksworld, and 10 of 40 in Sokoban; replies were read from its reasoning field when its answer field was empty. Table 10 gives every comparison.

Table 10: **Agreement between the checker and three judges.** *Missed*: items the checker marks false that the judge calls true. *Flagged*: items the checker marks true that the judge calls false. *Knights-and-Knaves: checker-true claims flagged in solutions the checker marks false; for gpt-oss-120b, nearly all precede the first false claim. Bottom: in Blocksworld and Sokoban the judge reads every line of the solution; in Knights-and-Knaves, the solution's hypothesis lines.

Per claim Domain	Judge	Missed	Flagged
Blocksworld	gpt-oss-120b	1/75	0/75
Sokoban	gpt-oss-120b	0/75	10/75
ProntoQA	gpt-oss-120b	32/72	31/75
ProntoQA	GPT-6 Sol	7/75	4/75
ProntoQA	DeepSeek-V4-Pro	32/58	1/69

All claims of a solution at once Domain	Judge	False solutions missed	Clean solutions flagged	False lines missed	True lines flagged
Knights-and-Knaves	gpt-oss-120b	0/150	2/100	0/312	154/267*
Knights-and-Knaves	GPT-6 Sol	0/40	1/40	1/84	53/73*
Knights-and-Knaves	DeepSeek-V4-Pro	0/36	12/40	2/75	56/65*
Blocksworld	GPT-6 Sol	0/150	0/111	0/440	0/978
Blocksworld	DeepSeek-V4-Pro	3/15	0/19	6/41	0/131
Sokoban	GPT-6 Sol	0/150	0/25	0/525	0/282
Sokoban	DeepSeek-V4-Pro	0/11	0/19	2/37	1/92

For gpt-oss-120b in Knights-and-Knaves, 260 solutions were judged and 10 excluded for unparseable output, leaving 150 the checker marks false and 100 it marks clean. The one-sided 95% upper bound on its rate of missed false solutions is 2.4%, and on its rate of flagged clean ones 7.0%; both flagged solutions are ones in which the model later fixed the flagged mistake. The judges agree less on which claims are false than on whether a solution contains one. On the 152 solutions both examined, gpt-oss-120b flags 466 claims, against 61 under the single-step definition, 307 under the local, and 544 under the global; its set of flagged claims matches the local definition's exactly on 53 solutions and the global's on 78. Of the 154 claims it flags beyond the checker's, 150 come before the solution's first false claim under the local definition, almost always the opening assumption that contradicts the hint: consistent with what the solution had assumed when it was made, false of the puzzle's solution, and flagged by a judge reading the solution as a whole. In the solutions the checker marks clean, the judges flag almost nothing: 2 of 100 for gpt-oss-120b, both self-corrections, and 1 of 40 for GPT-6 Sol.

H Reward re-scoring

Each sample's advantage is its reward minus the mean reward of its problem's 32 samples, divided by their standard deviation, as in GRPO (Shao et al., 2024). A problem whose samples all receive the same reward, a zero-variance group in the terminology of Yu et al. (2025), contributes no signal. Advantages are computed under the outcome reward and under a stricter reward that also requires no false claims, with unclipped importance ratios; no model is trained with them. Under clipped objectives such as GRPO or PPO, the reward on unsound solutions would shrink at the clip boundary while remaining positive.

For Qwen3-4B on its 694 unsolved Knights-and-Knaves problems, the stricter reward changes the total positive advantage by -55.9% for hinted solutions, $+0.3\%$ for the first steps, and -25.6% for unaided solutions; 42.1%, 0.0%, and 13.2% of the problems that yield a signal under the outcome reward yield none under the stricter one.

For Llama-3.1-8B on its 1,027 unsolved problems, the share of positive outcome reward that goes to solutions with a false claim is 69.6% for unaided solutions, 94.9% for a hint giving one identity, 94.6% for a hint giving two, and 13.1% for the first steps; 77.1% of the hinted problems that yield a signal stop yielding one under the stricter reward, against 5.9% for the first steps and 29.0% for unaided solutions. At most eight accepted solutions were stored per problem and condition for Llama, so rates are reweighted to the full 32 samples using the stored counts, and only reward shares are reported, since advantage values depend on group size.

I OTHER FORMS OF THE ANSWER

Table 11 gives all six phrasings of the two Knights-and-Knaves identities. The "Suppose" and "If X were" phrasings are checked with an extended extractor that also parses those templates; on every other condition it reproduces the original rates exactly. Every phrasing that poses the identities as hypotheses leaves few false claims (0.4 to 5.8% for Qwen), with Llama's "If X were" phrasing the highest at 27.2%, still far below its 94.6% under the hint. On the problems every phrasing recovers (274 for Qwen, 134 for Llama), the rates barely change (for Qwen, 65.7% under the hint and 0.3% under the first steps).

Table 11: **Six phrasings of the same two identities (Knights-and-Knaves).** Recovered: % of unsolved problems (694 Qwen, 1,027 Llama) with at least one accepted solution, with 95% Wilson intervals. Checkable: % of accepted solutions containing a hypothesis in the form the checker parses. False: of those, % containing a false claim (per solution, local definition). "–": no hypotheses.

Form	Phrasing	Recovered Qwen	Recovered Llama	Checkable Qwen / Llama	False, of checkable Qwen / Llama
In the question	Answer hint	63.7 [60.0, 67.2]	53.7 [50.7, 56.8]	100 / 94.7	67.1 / 94.6
Tested as hypothesis	Assume… (first steps)	75.5 [72.2, 78.6]	38.9 [36.0, 42.0]	100 / 99.9	0.4 / 13.1
	Suppose…	73.5 [70.1, 76.6]	39.2 [36.3, 42.3]	100 / 99.9	1.2 / 8.3
	If X were…	68.0 [64.5, 71.4]	26.0 [23.4, 28.8]	100 / 99.1	5.8 / 27.2
Stated as fact	It is established that…	77.1 [73.8, 80.1]	46.7 [43.7, 49.8]	0 / 4.1	– / 32.7
	Checking X, we find…	78.1 [74.9, 81.0]	50.3 [47.3, 53.4]	0 / 2.9	– / 45.5
	X is a knight	76.5 [73.2, 79.5]	47.6 [44.6, 50.7]	0 / 2.5	– / 59.7

Solutions under stated-as-fact phrasing. Under the phrasing "X is a knight", the supplied opening is the usual first line followed by the two identities stated as facts, and the model writes everything after them. A typical accepted solution from Qwen, on a puzzle with four people, reads in full:

```
Let's think step by step, by considering whether each person is lying and if that leads to
↪  contradiction.
Selena is a knave.
Ernest is a knight.
Nora is a knight.
Jasper is a knight.
Final Answer:
(1) Selena is a knave
(2) Ernest is a knight
(3) Nora is a knight
(4) Jasper is a knight
```

The first three lines are supplied; the model adds two identities and the answer. Llama's accepted solution to the same puzzle differs only by one added line, "This leads to a feasible solution." Across accepted solutions, the reasoning the model writes averages 42 characters for Qwen and 76 for Llama, against 269 and 265 under the answer hint. These solutions contain no hypothesis to check; they assert the answer.

Countdown with one first step. Giving a single first step instead of two leaves the hint's effect as small as in the main text: relative to one step, the hint changes the share of solutions with a false claim by $+0.2$ points for Qwen3-8B (95% CI $[-0.3, +0.7]$, 269 problems) and $+1.1$ for Qwen3-4B ($[+0.6, +1.7]$, 256 problems).

J FURTHER LOGIC DOMAINS AND MATH

ProntoQA. ProntoQA (Saparov & He, 2023) asks whether a statement follows from a set of rules and facts. Problems come from its official generator with a fictional ontology and random ordering: 300 problems at 5 hops, with two worked examples. A planned 7-hop setting was dropped when its first launch produced no outputs, and the run was relaunched at 5 hops before any results were seen. Claims are stated facts ("Rex is a rompus") and cited rules ("Rompuses are gorpuses"). A fact is false unless it follows in one step from the given rules and facts plus facts already claimed, and a rule citation is false unless the rule is in the problem. The hint gives only the final true/false answer, the first steps give two gold derivation steps, and a third, *plain* condition gives the question alone.

Outcome verification accepts false reasoning here even without help: under the plain condition, 20.1% (Qwen3-4B), 42.4% (Llama-3.1-8B), and 16.9% (Qwen3.5-4B) of the claims in accepted solutions are false, and 92.1%, 96.5%, and 73.5% of accepted solutions contain at least one. The hint exceeds the first steps in the share of accepted solutions with a false claim in all three models: 90.6% against 78.9% for Qwen3-4B ($+11.7$ points, 95% CI $[+9.1, +14.4]$, 298 problems), 95.2% against 88.3% for Llama ($+6.9$, $[+4.6, +9.3]$, 292 problems), and 79.4% against 48.2% for Qwen3.5-4B ($+31.2$, $[+27.9, +34.2]$, 295 problems). This measure saturates, since a solution makes about eleven claims. Per claim, the hint adds little or nothing over the plain condition: hint $-$ plain is $+0.2$ points ($[-0.3, +0.8]$) for Qwen3-4B, -0.4 ($[-1.8, +0.9]$) for Llama, and $+0.7$ ($[+0.0, +1.4]$) for Qwen3.5-4B, while the first steps lower the share (hint $-$ first steps $+4.1$, $+8.8$, and $+6.0$). The gap between hint and first steps comes mainly from the first steps lowering false claims. In 29.7% (Qwen) and 43.8% (Llama) of accepted hinted solutions, the reasoning derives a claim contrary to the hinted answer, yet the solution still reaches that answer. ProntoQA claims are not classified as forced or unforced, so the domain shows that outcome verification accepts false reasoning but does not bear on where it appears.

The one-step definition is strict. A line that compresses two inference steps into one states something true that does not follow in one step, and the local rule counts it as false, so the ProntoQA rates include such compressions. They are a large share: of the claims the local rule marks false in accepted solutions, 41.5% (hint) and 44.5% (plain) for Qwen3-4B, 33.9% and 37.5% for Qwen3.5-4B, and 29.9% and 32.9% for Llama-3.1-8B are true under the global rule (entailment by forward chaining over the rules); among false fact claims alone the shares are 63.8–80.3%. This affects the absolute rates only: the comparison between the hint and the plain condition applies the same rule to both, and the hint adds little or nothing under it. Counting compressions as true leaves 11.9% of claims false under the hint and 11.2% under plain for Qwen3-4B, 11.7% and 10.6% for Qwen3.5-4B, and 29.4% and 28.5% for Llama-3.1-8B.

ZebraLogic. ZebraLogic (Lin et al., 2025) is run with Qwen3-4B and Llama-3.1-8B on the 40 puzzles of size 2×3, the only size inside the inclusion rules for either model; unaided, they leave 28 and 29 of them unsolved. Solutions test hypotheses about grid cells ("Suppose House 1 Name = Alice. No contradiction is found."), and each verdict is checked against the unique solution grid. The hint was expected to exceed the first steps in the share of accepted solutions with a false claim; this is confirmed in neither model. For Qwen, 61.1% of accepted hinted solutions contain a false claim, against 50.7% under the first steps and 66.2% unaided, and the paired difference is $+10.3$ points (95% CI $[-0.2, +21.7]$) on only 24 problems. For Llama, the first steps produced false claims in 90% of solutions, leaving no room to measure an effect of the hint. In this domain, hard for both models, false claims are common under every condition.

MATH. MATH (Hendrycks et al., 2021) has no step checker, so only measurements that need none are reported. The set is 175 problems at levels 3 to 5 that Qwen3-4B-Base never solves in 32 attempts. Given the answer as a hint, the model recovers all 175, with 15.7 of 32 samples accepted per problem, a mean length of 802 characters, and 4.9% of accepted solutions containing no equation; across all hint samples, 38.1% conclude a different answer from the hint and a further

Table 12: **Models trained on unsound solutions give valid answers that rest on false claims.** Unsound minus Sound, in percentage points, with 95% intervals from resampling problems and, in parentheses, the number of held-out problems both models solve at least once, on which the difference is computed. Knights-and-Knaves: share of valid answers containing at least one false claim (per solution). Blocksworld and Sokoban: share of unforced claims that are false among valid plans (per claim). Sound and Unsound train on the same problems, the same number of solutions per problem, and for the same number of steps. ID/OOD: in-/out-of-distribution.

Domain	Model		Seed 0	Seed 1	Seed 2	Mean
Knights-and-Knaves	Qwen3-4B	ID	+74.3 [69.7, 78.6] (386)	+71.8 [67.7, 75.6] (423)	+78.4 [75.1, 81.9] (438)	**+74.8**
	Llama-3.1-8B	ID	+47.4 [42.5, 52.7] (353)	+63.6 [58.7, 68.6] (341)	+46.5 [42.1, 50.7] (431)	**+52.5**
	Qwen3-4B	OOD	+82.3 [76.1, 88.1] (173)	+87.5 [82.9, 91.8] (188)	+82.1 [76.2, 87.3] (195)	+84.0
	Llama-3.1-8B	OOD	+65.8 [55.2, 76.8] (93)	+73.7 [63.1, 83.4] (90)	+51.2 [41.5, 59.9] (151)	+63.6
Blocksworld	Qwen3-8B		+45.7 [39.3, 51.6] (72)	+22.1 [17.0, 27.6] (59)	+25.9 [18.3, 33.5] (53)	**+31.2**
	Llama-3.1-8B		+23.3 [19.4, 27.6] (63)	+25.7 [21.1, 30.3] (66)	+30.7 [25.8, 35.7] (64)	**+26.6**
Sokoban	Qwen3-8B		+46.8 [37.2, 56.9] (65)	+52.8 [41.1, 64.5] (66)	+48.8 [40.2, 58.4] (79)	**+49.5**
	Llama-3.1-8B		+44.8 [38.9, 50.7] (111)	+42.4 [38.1, 46.4] (166)	+35.2 [29.9, 40.6] (149)	**+40.8**

12.7% state none. Given the answer stated as a fact ("It is established that the answer is X"), it recovers 174, with 31.3 of 32 samples accepted, a mean length of 5 characters, and 99.6% of accepted solutions containing no equation: the same collapse into uncheckable solutions seen in Knights-and-Knaves.

K INHERITANCE: FULL RATES AND ACCURACY

Table 12 gives the values behind Figure 3.

Table 13 gives the rates behind Table 12, over all held-out problems rather than only those both models solve. Among valid plans, forced claims stay at 0.00 to 0.51% across all 28 trained planning models (34 evaluations, including six re-evaluations on 900 problems); the maximum is Llama's Unsound Blocksworld model in seed 1 (50 of 9,808 claims), every other model is at or below 0.18%, and the four starting models are at 0.00%.

Table 13: **Rates behind Figure 3**, over all held-out problems, mean ± standard deviation over three training seeds. Knights-and-Knaves: share of correct solutions with a false claim (per solution). Blocksworld and Sokoban: share of state or position claims that are false among valid plans (per claim; Blocksworld on 900 problems, except the Qwen starting model, on 300). Lower dose: one seed.

Domain	Model	Starting model	Sound	Unsound	Lower dose (Sound / Unsound)
Knights-and-Knaves (ID)	Qwen3-4B	22.8%	5.8 ± 2.2	77.4 ± 1.2	—
Knights-and-Knaves (ID)	Llama-3.1-8B	67.1%	37.8 ± 6.1	88.3 ± 3.6	—
Knights-and-Knaves (OOD)	Llama-3.1-8B	75.7%	35.0 ± 12.5	92.8 ± 2.6	—
Blocksworld	Qwen3-8B	30.5%	4.2 ± 1.6	35.0 ± 10.8	6.3/35.3
Blocksworld	Llama-3.1-8B	9.3%	0.0 ± 0.0	27.7 ± 5.5	—
Sokoban	Qwen3-8B	29.5%	0.7 ± 0.9	53.6 ± 2.0	0.1/34.1
Sokoban	Llama-3.1-8B	25.6%	2.6 ± 1.6	43.4 ± 6.4	—

Further inheritance results. Judged against the true solution, the Knights-and-Knaves gap holds as well (9.3% against 71.5% in the first Qwen seed). For Llama in Blocksworld, sound training brings false state descriptions to 0.0% in all three seeds, across 26,534 claims. At about a quarter of the training (Qwen, one seed per domain), the gap is +63.0 points in Knights-and-Knaves (95% CI $[+58.4, +67.3]$), +28.6 in Blocksworld ($[+23.0, +34.5]$), and +28.7 in Sokoban ($[+19.9, +41.8]$). Of the All-accepted model's valid answers, 48.4% in distribution and 36.9% out of distribution contain a false claim, against 22.8% and 18.9% for the starting model.

Accuracy. Table 14 gives pass@1 and pass@32 for every trained Knights-and-Knaves and Blocksworld model, and Table 15 the paired Sound − Unsound differences; Sokoban accuracy

is in Appendix M. In Knights-and-Knaves, every in-distribution interval for Qwen includes zero, while Llama's differences are larger and change sign across seeds. In Blocksworld the differences are at most about 2 points, mostly in the Unsound model's favor. The All-accepted model trains on 4,830 solutions over 442 problems, against 372 over 118 for Qwen's Sound and Unsound models, so its higher accuracy reflects more data and more problems; §6 compares filtered and unfiltered sets of equal size. For Qwen, the Sound and Unsound models both fall below the starting model in distribution (33 to 41% against 43.0%), while the All-accepted model matches it (44.8%), so training on a pool of 118 problems costs accuracy whether the pool is sound or not. Re-evaluating all 14 Blocksworld models on the same problems with a second sampling seed changes pass@1 by at most 0.12 points and pass@32 by at most 0.7. The high pass@32 of three Qwen models in Table 14 (Unsound seeds 0 and 1, and Sound at the lower dose) reproduces; it marks models whose successes are spread thinly over many problems, and it does not follow the split between Sound and Unsound training.

Table 14: **Accuracy of trained models.** pass@1 and, in parentheses, pass@32, with 32 samples per problem; mean ± standard deviation over three training seeds where applicable. Knights-and-Knaves: 1,000 in-distribution (ID) and 1,000 out-of-distribution (OOD) held-out puzzles. Blocksworld: 900 held-out problems, except the Qwen starting model (300). Lower dose: about a quarter of the training, one seed.

Domain	Family	Model	pass@1 (pass@32), ID	pass@1 (pass@32), OOD
Knights-and-Knaves	Qwen3-4B	Starting	43.0 (61.6)	11.9 (33.8)
		Sound	37.8 ± 4.1 (53.9 ± 4.4)	14.5 ± 0.9 (27.5 ± 2.2)
		Unsound	37.2 ± 3.0 (57.2 ± 3.0)	13.5 ± 2.5 (35.7 ± 2.2)
		All-accepted	44.8 (74.2)	14.3 (44.9)
		Sound, lower dose	34.1 (57.3)	13.1 (31.7)
		Unsound, lower dose	32.7 (58.6)	10.4 (35.8)
	Llama-3.1-8B	Starting	33.1 (51.0)	8.8 (21.8)
		Sound	34.6 ± 7.5 (50.5 ± 6.3)	9.9 ± 2.9 (20.6 ± 4.3)
		Unsound	31.7 ± 3.7 (49.7 ± 5.7)	8.8 ± 2.3 (21.5 ± 3.6)
Blocksworld	Qwen3-8B	Starting	3.7 (8.0)	—
		Sound	5.5 ± 0.3 (8.8 ± 1.2)	—
		Unsound	6.4 ± 0.7 (19.3 ± 12.3)	—
		Sound, lower dose	7.8 (26.9)	—
		Unsound, lower dose	5.3 (9.1)	—
	Llama-3.1-8B	Starting	1.8 (6.6)	—
		Sound	7.1 ± 0.2 (8.1 ± 1.4)	—
		Unsound	7.9 ± 1.0 (10.7 ± 2.2)	—

Table 15: **Sound − Unsound pass@1**, in percentage points, paired on the same held-out problems; mean ± standard deviation over three training seeds. Blocksworld on 900 problems; Sokoban is in Table 18. Per-seed 95% intervals exclude zero for Llama in distribution (all three seeds, in both directions), Qwen out of distribution (two seeds), Qwen in Blocksworld (one seed), and Llama in Blocksworld (all three seeds); all others include zero.

Domain	Family	Sound − Unsound
Knights-and-Knaves, ID	Qwen3-4B	$+0.5 \pm 1.8$
Knights-and-Knaves, ID	Llama-3.1-8B	$+2.9 \pm 5.0$
Knights-and-Knaves, OOD	Qwen3-4B	$+1.0 \pm 2.1$
Knights-and-Knaves, OOD	Llama-3.1-8B	$+1.1 \pm 0.7$
Blocksworld	Qwen3-8B	-1.0 ± 1.0
Blocksworld	Llama-3.1-8B	-0.8 ± 1.2

In the training sets, 0.0% (Filtered), 12.6% (String-filtered), and 67.9% (Mixture) of solutions contain a false claim. pass@1 for the Filtered and Mixture models is 40.6 and 47.2 in seed 0, 39.9 and 46.5 in seed 1, and 48.2 and 39.7 in seed 2. The Mixture models are more contaminated than the All-accepted model (56 to 70% against 48.4%) with the same number of steps, as their smaller set is seen about 1.25 times against 0.41.

Table 16: **Filtering a realistic pool (Knights-and-Knaves, Qwen3-4B).** Training sets of 1,587 solutions drawn from the 4,830 accepted hinted solutions: every solution with no false claim (Filtered); the pool with string-match flags removed (String-filtered); a random sample from the Filtered set's problems (Same problems); and a random sample from all problems (Mixture). All models were trained from the round-one model for 496 steps (final logged loss 0.007 to 0.016) and evaluated on the 1,000 in-distribution held-out puzzles. Mean ± standard deviation over three training seeds. Middle: share of valid answers containing a false claim, over all problems (per solution). Right: difference from the Filtered model in percentage points, on problems both solve; in every seed its 95% interval excludes zero. Mixture − Same problems: $+17.3 \pm 8.3$. pass@1: 42.9 ± 4.6 for Filtered, 44.5 ± 4.1 for Mixture. The All-accepted model (all 4,830 solutions, one seed) reaches 48.4%. Seed 1's Mixture evaluation ran in two halves of 500 problems with the same seeded sampling.

Training set	Problems	False claims in valid answers	Minus Filtered
Filtered	222	2.3 ± 1.8	—
String-filtered	226	16.0 ± 5.7	$+13.9 \pm 6.1$
Same problems	222	40.6 ± 9.0	$+38.8 \pm 6.7$
Mixture	360	61.3 ± 7.9	$+58.6 \pm 4.8$

L ANSWERS AND REASONING

The logic prior. Write α for the share of a model's held-out solutions with sound reasoning that reach the right answer, and β for the share with at least one false claim that do; solutions with no claims are excluded. In the hinted Knights-and-Knaves pool, regenerated with every sample kept, $\alpha = 0.902$ and $\beta = 0.287$ ($\alpha - \beta = +0.615$, 95% CI $[+0.568, +0.661]$); in the first-steps pool, $\alpha = 1.000$ and $\beta = 0.005$. Table 17 gives the trained models. For Qwen, training on unsound solutions removes the logic prior in every seed: $\alpha - \beta$ is -0.002 ($[-0.057, +0.046]$), -0.085 ($[-0.140, -0.037]$), and -0.051 ($[-0.103, -0.004]$). For Llama the prior is weaker from the start and varies more across seeds.

Table 17: **Logic prior and advantage gap in trained Knights-and-Knaves models.** α, β: share of sound and of unsound held-out solutions that reach the right answer. Advantage gap: mean GRPO advantage of sound minus unsound solutions, within each problem's 32 held-out solutions, over problems with both correct and incorrect solutions. Sound and Unsound: mean ± standard deviation over three training seeds.

Family	Model	α	β	$\alpha - \beta$	Advantage gap
Qwen3-4B	Starting	0.947	0.151	$+0.796$	0.825
	Sound	0.917 ± 0.020	0.037 ± 0.019	$+0.880 \pm 0.034$	1.120 ± 0.081
	Unsound	0.338 ± 0.009	0.384 ± 0.041	-0.046 ± 0.042	0.333 ± 0.061
	All-accepted	0.805	0.304	$+0.501$	1.007
Llama-3.1-8B	Starting	0.704	0.263	$+0.441$	0.136
	Sound	0.405 ± 0.095	0.282 ± 0.068	$+0.123 \pm 0.037$	0.172 ± 0.041
	Unsound	0.366 ± 0.131	0.312 ± 0.046	$+0.054 \pm 0.160$	0.038 ± 0.034

False hints. The round-one model and the All-accepted model were given the 523 problems of the scale comparison with both hinted identities flipped. The round-one model follows the false hint: 85.3% of its final answers adopt both false identities, and in its rare escapes (0.6% of solutions) it reasons its way out, retracting the false assumption in 75% of them. The All-accepted model adopts both false identities in only 28.4% of final answers, although given a false "knave" hint it opens with it 67.8% of the time, against 24.1% for the round-one model. It reaches the true identities in 58% of solutions, but mostly not by reasoning: in 52.1% of solutions on problems it was trained on (95% CI $[48.1, 56.2]$) and 39.9% on problems it never saw ($[30.8, 49.3]$), it assumes the false identity, declares it consistent, never revisits it, and then gives the true identity in its final answer. The round-one model does so in 0.2% and 0.0%. Because the pattern holds on unseen problems as well, recall of training answers does not explain it.

How much the answer depends on the reasoning. Following Yu et al. (2026), the same sound held-out derivations (497 for Qwen, 210 for Llama) are fed to the starting, Sound, and Unsound models, and the probability of the final answer is read at 11 evenly spaced points along each derivation; the causal importance of the reasoning summarizes how much the derivation moves that probability. It is lower for the Unsound model than for the Sound model in all six family–seed pairs (pooled difference -0.123, 95% CI $[-0.134, -0.113]$). The Unsound model does not settle on its answer earlier: its answer probability before any reasoning is lower (-0.066, $[-0.084, -0.049]$), and the point at which the answer becomes likely shows no consistent difference at any threshold from 0.5 to 0.95. What differs is how strongly it responds to a sound derivation. Reading the full derivation raises the answer's probability by 0.42 to 0.66 for Qwen's Sound models but by only 0.10 to 0.12 for its Unsound models (differences of -0.32 to -0.54, each interval excluding zero), including among derivations after which the model is confident, so the gap is not a floor effect. In Llama the gap appears in two of three seeds (-0.28 and -0.09; the third is null), and Llama's Sound models also respond less than the starting model.

M SOKOBAN

Solving. Table 18 gives pass@1 on held-out Sokoban puzzles. At the lower dose, the Qwen models already differ (21.8% for Sound against 3.4% for Unsound).

Table 18: **Sokoban pass@1** on held-out puzzles, mean ± standard deviation over three training seeds, with the Sound − Unsound difference paired on the same puzzles; in every seed its 95% interval excludes zero.

Model	Starting	Sound	Unsound	Sound − Unsound
Qwen3-8B	4.9	25.8 ± 2.6	8.1 ± 0.5	$+17.6 \pm 3.1$
Llama-3.1-8B	1.2	39.0 ± 8.2	22.0 ± 6.6	$+17.0 \pm 8.3$

Failed solutions. Among the Unsound Qwen model's failed solutions, 36.4 to 40.2% contain a false position claim, and in 35.1 to 37.6% that claim comes at or before the first move that makes the puzzle unsolvable; for the Sound model the figures are 4.7 to 12.3% and 2.5 to 8.6%, and the starting model falls between (24.8% and 24.1%). The first false claim typically comes at the first step for Unsound and the third or fourth for Sound, while the first fatal move comes at the second step in both. Pooled over seeds, 51.2% of the Unsound Llama model's failures contain a false position claim, and in 46.6% it comes at or before the first fatal move, against 11.0% and 8.7% for the Sound model. The account is partial: 60% of Qwen's and about half of Llama's Unsound failures contain no false position claim, and 96 to 97% of failures in both models contain a fatal move.

Necessary and unnecessary moves. A move in an accepted Sokoban solution is *necessary* if deleting it makes the rest of the plan illegal or miss the goal, as checked in the simulator; moves whose status cannot be determined count as necessary. The natural first test, whether false moves cluster on unnecessary ones, is uninformative, because almost no moves are false either way. The position claims that follow each kind of move are more telling. In hinted solutions, position claims right after an unnecessary move are false more often than those after a necessary one (94.4% against 65.3% for Qwen, $+29.1$ points, 95% CI $[+28.3, +29.8]$; 87.5% against 57.8% for Llama, $+29.7$, $[+28.7, +30.7]$). But unnecessary moves come late in a solution, after the model has often lost track of the state. Standardized by position in the solution, the difference falls to $+15.9$ ($[+12.0, +19.4]$) and $+14.9$ ($[+10.1, +20.0]$), and among claims made before a solution's first false claim it disappears (-12.8, $[-25.9, -0.9]$, for Qwen; -0.7, $[-7.3, +5.2]$, for Llama). Within one kind of claim, the difference mostly reflects position and errors carried forward.

N SCREENING SIGNALS

Both signals are evaluated on the existing pools of accepted hinted solutions, without a separate held-out split, in Knights-and-Knaves, Blocksworld, and Sokoban.

Contradicting the hint. On the full pool of Qwen's accepted Knights-and-Knaves hint solutions, the string match flags 93.1% of those containing a false claim, with precision 98.8%. On the matched Sound and Unsound pools of §6, its precision is 92.8% (95% CI $[89.4, 95.2]$) and its recall 76.6%. For Llama its precision on the matched pools is 53.9%, because Llama's sound solutions also test identities opposite to the hint and reject them correctly.

Stating the answer early. Following Lee et al. (2026), a second signal flags solutions that state the complete answer early in their reasoning. It flags no solution at any cutoff from 10 to 40% of the reasoning, in any domain. As a continuous score, it is uninformative in the planning domains (AUROC 0.49 to 0.57) and points the wrong way in Knights-and-Knaves for Qwen (0.15): unsound solutions state the complete answer only at the end.

A process reward model. Qwen2.5-Math-PRM-7B (Zhang et al., 2025c) scores each step of a solution, with steps separated as on its model card and the puzzle shown without the hint. A solution is scored by its lowest step score and flagged when that score falls below 0.5; the model has no official threshold. Samples are balanced and drawn with seed 0: 500 accepted hinted Knights-and-Knaves solutions from Qwen that contain a false claim and 500 that do not, and 100 of each from Qwen's Blocksworld solutions. Table 19 gives the results. The mean lowest step score is 0.352 for flawed and 0.494 for clean Knights-and-Knaves solutions, and 0.591 and 0.639 in Blocksworld. In Knights-and-Knaves the lowest-scored step rarely coincides with the first false claim, and it usually falls later in the solution. On a balanced Knights-and-Knaves sample, the string match reaches 92.8% precision at 76.6% recall. The reward model reaches 62.4% precision at 78.4% recall with a threshold of 0.5, and it reaches 92.8% precision only at a threshold of 0.033 or below, where it flags a single solution (0.2% recall); at every looser threshold its precision stays lower. The reward model was trained on mathematics, so these results show that an off-the-shelf model does not solve the problem here; a reward model trained on these domains might.

Table 19: **An off-the-shelf process reward model as a filter.** Balanced samples of accepted hinted solutions, labeled by the checker. Localization: share of flawed solutions whose lowest-scored step is the checker's first false line, and share where it is that line or an earlier one.

Domain	AUROC	Precision	Recall	Lowest step is first false line	… or earlier
Knights-and-Knaves	0.736	62.4%	78.4%	0.8%	2.8%
Blocksworld	0.597	83.9%	26.0%	62.0%	96.0%

O A SECOND ROUND

A second round of hint recovery, starting from the All-accepted model, was expected to produce at least as many false claims as the first. It produced fewer: 18.0% of accepted hinted solutions contain one, against 67.1%. Most of the difference lies in which problems need the hint. Both models begin a solution by assuming that the first person named is a knight. When that person is in fact a knight, hinted solutions contain a false claim 5 to 10% of the time; when that person is a knave, 49 to 82% of the time, for both models on both rounds' problems. The problems left unsolved in the first round are 80% knave-first, because the starting model's default opening gets exactly those wrong. Those left unsolved in the second round are only 23% knave-first, because they are mostly problems the first-round model could already solve and the second-round model had lost. Reweighted to the first round's mix of problems, the second round would have about 41% of accepted hinted solutions with a false claim. The remaining difference is between the models on knave-first problems (69 to 82% for the starting model, 49 to 50% for the All-accepted model), consistent with the All-accepted model adopting a knave hint as its opening assumption more often (58.9% of the time, against 19 to 26%). Hint recovery produces false reasoning where the hinted answer conflicts with the line of reasoning the model would otherwise follow, and how much a round produces depends on how many of its unsolved problems carry that conflict.

P FURTHER CHECKS

Flipped first steps. With the round-one Qwen model on its 694 unsolved Knights-and-Knaves problems, first steps that keep the wording but flip the logic lead to a correct answer in 115 of 22,208 solutions (0.5%), although nearly all end in a well-formed answer. Among the 22,086 well-formed wrong solutions, 87.2% of claims are false under the local definition and all of them under the global one, and none contains a retraction. A Llama run gives 0.9% correct.

Sampling temperature. Rerunning the Knights-and-Knaves hint condition at temperatures 0.3, 0.6, and 1.0 gives 70.4%, 67.1%, and 63.8% of accepted solutions with a false claim, 32.6 to 35.5% of claims false, and 58.7 to 66.1% of solutions that contradict the hint without retracting, while recovering 313, 442, and 557 of the 694 problems.

Countdown. Against the first steps, the hint changes the share of false claims by $+1.0$ percentage points for Qwen3-4B (95% CI $[+0.5, +1.6]$) and -0.4 for Qwen3-8B ($[-2.0, +0.9]$), and the share of solutions with a false claim by $+1.3$ ($[+0.8, +1.8]$) and $+0.7$ ($[+0.2, +1.5]$). For Qwen3.5-4B the two changes are $+0.2$ and $+0.7$ points ($[+0.2, +1.1]$ for the latter). Appendix I gives the comparison with a single first step.

The collision in the planning domains. The Knights-and-Knaves account of where false claims come from, a hinted answer colliding with the model's default opening, has a natural planning analogue in a hinted plan diverging from the model's own tracking of the state; it was not measured.

What the checker adds. A strong judge matches the checker's accuracy in these domains. What the checker adds is a fixed, stated definition applied to every solution at no cost, one that could not be gamed if it were used as a reward.

Q LENGTH AND WORDING

Length. All correct held-out Knights-and-Knaves solutions of the Sound and Unsound models (seeds 0 and 1 pooled, all problems) are grouped by claim count (1–2, 3, 4, 5, 6–8, and 9 or more), and the gap in the share of solutions with a false claim is reweighted to the pooled distribution of claim counts, with 1,000 bootstrap resamples. Table 20 gives the result. Qwen's per-bin gaps are $+43$ to $+75$ points at 1 to 5 claims and $+28$ at 6 to 8. Llama's gap falls with length: $+77$, $+42$, $+27$, $+12$, and $+13$ points over the first five bins, and zero at nine or more claims, where every solution of both models contains a false claim.

Table 20: **Knights-and-Knaves inheritance gap standardized for length.**

	Standardized gap	Claims false (Sound / Unsound)	Claims per solution (Sound / Unsound)
Qwen3-4B	$+69.3$ $[+68.7, +70.0]$	3.1% / 45.3%	3.69 / 3.80
Llama-3.1-8B	$+42.5$ $[+41.6, +43.6]$	18.9% / 51.7%	3.03 / 4.01

Wording. Table 21 compares verdict phrasing and length across starting models, trained models (4,000 held-out solutions each, seeds 0 and 1), and training sets. "No contradiction" dominates the verdicts everywhere, and retraction rates stay low. The Jensen–Shannon divergence between Sound and Unsound outputs is 0.002 to 0.006 over words and 0.005 to 0.010 over bigrams, about the same as between two seeds of the same model. Neither control applies in the planning domains, where the measurement is per claim, which already divides out length, and where claims carry no verdict phrase.

Table 21: **Verdicts and length in Knights-and-Knaves.** Share of verdicts that are "no contradiction", share that are retractions, and mean characters and claims per solution.

	"No contradiction"	Retractions	Characters	Claims
Qwen starting	95.1%	4.9%	611	3.95
Qwen Sound, seeds 0 / 1	97.9 / 97.8%	2.1 / 2.2%	582 / 582	3.77 / 3.78
Qwen Unsound, seeds 0 / 1	96.1 / 97.8%	3.9 / 2.2%	607 / 581	4.00 / 3.79
Qwen training sets, Sound / Unsound	98.4 / 98.9%	1.6 / 1.1%	635 / 642	4.22 / 4.34
Llama starting	99.0%	1.0%	596	3.87
Llama Sound, seeds 0 / 1	94.6 / 97.9%	5.4 / 2.1%	515 / 476	3.20 / 2.83
Llama Unsound, seeds 0 / 1	97.4 / 98.7%	2.6 / 1.3%	675 / 596	4.57 / 3.93
Llama training sets, Sound / Unsound	97.4 / 98.8%	2.6 / 1.2%	498 / 665	2.98 / 4.48