GROVE: BENCHMARKING AND TRAINING LLM AGENTS TO PERSUADE

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM agents increasingly settle technical disputes with one another before LLM judges that cannot verify every step, so the outcome turns on persuasion: winning the judge's preference over a competing argument. Recent work trains models to persuade, but a judged win rate cannot tell whether a trained agent also follows the task's procedure or supports the correct decision. We introduce **GROVE-Bench**, three environments in which agents rebut a code review, debate a factual claim with retrieved evidence, or challenge a solution to a graduate-level question. Each fixes what every participant sees, pits trained agents against frozen models from 2B to 397B parameters in complete round robins, and records judged preference separately from execution and dataset labels. We also propose **GROVE** (Group-Relative Optimization of Verbal Exchanges), which rewards each argument relative to others sampled for the same input without querying the judge, and instantiate it with programmatic rewards (**GROVE-P**). On items used during training or model selection, the SFT–DPO–GROVE-P pipeline raises field win rates over the base model by 46.9% in Code Review (9B, under a judge that labeled no training data) and 70.2% in Evidence Debate (4B), matching or beating every frozen model except the 397B one; in code, imitating winning arguments supplies most of the gain. On held-out reasoning questions, where GROVE ran only as probes of at most 25 updates, no method improves on the base model. In evidence debates, higher win rates do not imply better execution: the highest-rated trained debater leaves the required first search to the engine in almost every debate, and its wins do not track which side of the claim is true. Persuasion training should therefore be evaluated on execution and agreement with dataset labels as well as on judged preference.

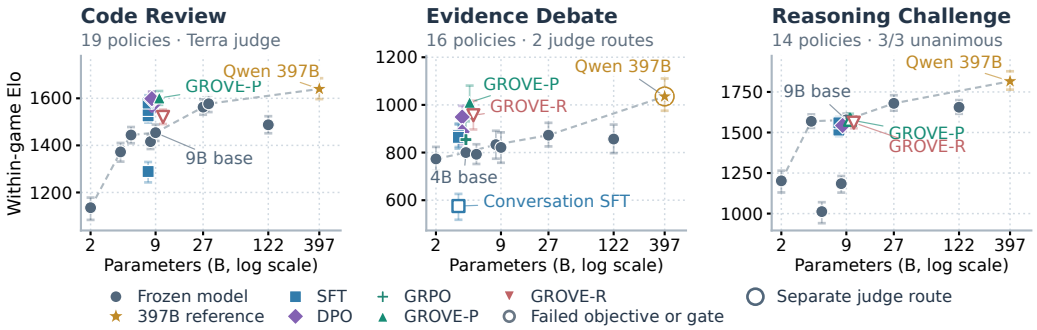

Figure 1: **Trained small agents rise above the frozen-model frontier on code and evidence items used during training or model selection, not on held-out reasoning questions.** Elo (a Bradley–Terry rating fit per arena; higher is better) versus total parameters; dashed: frozen-model frontier; bars: 95% bootstrap intervals; evidence Elo anchored at 800; trained policies offset for visibility. Judges: GPT-5.6-Terra in code, two GPT-5.6-Sol routes in evidence, and unanimous decisions of three Qwen judges in reasoning.

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.

1 INTRODUCTION

LLM agents increasingly settle technical questions with one another. When a reviewer agent flags a possible division by zero, the author agent must defend its code or concede; when two agents dispute a factual claim, each argues from the evidence it retrieved; when one agent checks another's solution, it must decide what to challenge. The deciding agent rarely verifies every claim itself, so the outcome depends on how well each side argues and not only on who is right. Hence a correct answer can lose to a wrong one. Persuasion, winning the deciding agent's preference over a competing argument, is therefore a core capability that multi-agent systems must measure and improve.

Prior work shows that persuasion can be measured and optimized, but not across technical interactions like these. Persuasion benchmarks score persuasive messages or measure opinion change in people (Singh et al., 2025; Hackenburg et al., 2024; Salvi et al., 2025); debate studies ask whether competing advocates help a weaker judge answer questions it cannot answer alone (Irving et al., 2018; Khan et al., 2024; Kenton et al., 2024); and persuasion training can make receivers abandon correct answers (Bozdag et al., 2026). Debate studies report judge accuracy next to win rates, and Wen et al. (2024) separate human approval from correctness on code, but these studies do not report whether an agent follows the task's procedure. We ask whether training makes small agents as persuasive as much larger ones, and whether higher judged preference comes with better execution.

We hypothesize that training can raise judged preference without improving execution, as a higher judged win rate can reflect stronger grounding but also a judge's preference for position or length (Zheng et al., 2023; Dubois et al., 2024), exploitation of a learned reward (Gao et al., 2023), or information the opponent lacks. This hypothesis fails if the trained policies that win most often also execute best, for example by issuing their own evidence searches. A benchmark must fix what each participant sees, compare every policy with a common field of opponents, and record execution and agreement with dataset labels separately from preference.

We introduce **GROVE-Bench**, three environments in which agents argue about technical artifacts that the judge cannot fully verify (Figure 2). In **Code Review**, an author rebuts a fixed review of a candidate program, and the judge cannot execute the code. In **Evidence Debate**, two advocates debate a factual claim using Wikipedia retrieval that the judge never sees. In **Reasoning Challenge**, two challengers contest a frozen solution to a graduate-level question without an answer key. All three meet the first two requirements, and Evidence Debate also records execution.

We propose **GROVE** (Group-Relative Optimization of Verbal Exchanges), two judge-free reward designs for group-relative policy optimization (GRPO; Shao et al., 2024), which scores each argument against siblings sampled for the same context. **GROVE-P** uses programmatic rewards that favor (i) grounding and a stance matching the hidden candidate label in code, (ii) citing retrieved titles and covering the opponent in evidence, and (iii) embedding similarity to a teacher argument in reasoning. **GROVE-R** instead scores each sibling with learned critics against a same-context anchor argument. We compare both with supervised fine-tuning (SFT), direct preference optimization (DPO), and, in evidence, judge-reward GRPO.

On code and evidence items used during training or model selection, small trained agents match or beat every frozen model except the largest (Figure 1), so these arenas measure specialization to fixed items. The SFT–DPO–GROVE-P pipeline raises the field win rate of a 9B code policy from $42.9 \pm 2.4\%$ to $63.0 \pm 2.0\%$ (+20.1 pp; +46.9%) under a judge family held out from training labels, and of a 4B evidence policy from $41.3 \pm 2.7\%$ to $70.3 \pm 2.9\%$ (+29.0 pp; +70.2%). Both trail only Qwen3.5-397B (Qwen Team, 2026a), by 5.0 pp in code and 3.0 pp in evidence, gaps inside their 95% intervals ($[-2.7, 12.5]$ and $[-7.3, 13.7]$); head-to-head, GROVE-P wins $42.3 \pm 4.0\%$ and $35.0 \pm 10.7\%$ against it (the evidence comparisons with 397B used a separate API route). SFT on winning arguments (winner SFT) supplies most of the code gain, yet loses to 397B head-to-head ($38.5 \pm 3.7\%$). In evidence, GROVE-P leads its DPO initializer by 16.7 pp (95% interval $[8.0, 25.3]$), although a zero-reward control also moves 9.7 pp; its 3.2 pp lead in code is within noise. On held-out reasoning questions, no trained policy differs from base 9B, and the reasoning GROVE runs are probes of at most 25 updates.

1.1 CONTRIBUTIONS

- We propose **GROVE-Bench**, three environments with explicit information contracts and complete round-robin arenas of 19, 16, and 14 policies (72,089 judgments), with label

checks in all three and execution records in Evidence Debate (Section 3).

- We show that on items used during training or model selection, small trained agents match or beat every frozen model except the largest, raising field win rates by 46.9% in code (9B) and 70.2% in evidence (4B) and extending small-persuader gains (Han et al., 2025; Singh et al., 2025) to technical arguments; winner SFT on 286 rebuttals supplies most of the code gain (+17.2 pp, 95% interval $[10.9, 23.4]$; Section 6.2).
- We propose **GROVE**, judge-free rewards for group-relative training of persuaders; its programmatic variant GROVE-P yields the highest-rated trained policy in code and evidence, although its gain over the DPO initializer is not separable from evaluation variation (Sections 4 and 6.2).
- We show that judged preference can rise while execution fails, as evidence GROVE-P wins 87.6% of debates against seven frozen models in a 200-claim diagnostic arena but leaves the required first search to the engine in 99.8% of them (Section 6.3).

2 Related Work

From persuasive messages to receivers. LLM persuasion was first measured as opinion change in people (Hackenburg et al., 2024; Salvi et al., 2025). Targeted training (Singh et al., 2025) and RL (Cheng & You, 2026) make small models more persuasive, and a 3B RL-trained persuader outperforms GPT-4o (Han et al., 2025). These works persuade a receiver on everyday or social topics; we train persuaders for technical disputes that a judge cannot fully verify. Preference training can make model receivers sycophantic (Sharma et al., 2023), and RL-trained persuaders make them abandon correct answers (Bozdag et al., 2026). Wen et al. (2024) report that RLHF makes models more convincing to humans but not more correct, although a re-analysis attributes much of this evidence to bugs (Chandna et al., 2026). Sotopia scores how well role-playing agents reach social goals (Zhou et al., 2024), whereas GROVE-Bench scores a judge's preference between competing technical arguments.

Debate and critique as oversight. Debate lets competing advocates help a limited judge choose correctly (Irving et al., 2018). Debate helps non-expert judges supervise unreliable experts (Michael et al., 2023), more persuasive and self-play-trained debaters improve judge accuracy in reading comprehension (Khan et al., 2024; Arnesen et al., 2024), and across nine tasks and three protocols its benefit depends on the task and information gap (Kenton et al., 2024). Related lines train code critics, checkable provers, and self-play players (McAleese et al., 2024; Kirchner et al., 2024; Cheng et al., 2024), and debate training reduces reward hacking in RLAIF (Kenton et al., 2026). Most of these studies measure judge accuracy in question answering or mathematics. Persuasion-balanced training teaches models when to concede (Stengel-Eskin et al., 2025), whereas our work compares training signals for persuaders arguing about programs, factual claims, and proposed solutions.

Learning from comparisons, and how it fails. Preference learning trains reward models on comparisons (Christiano et al., 2017) or optimizes preference pairs directly with DPO (Rafailov et al., 2023). GRPO compares rewards among responses sampled for the same prompt (Shao et al., 2024; DeepSeek-AI, 2025), and later work uses rewards such as BLEU overlap with a reference (Chang et al., 2025) or pairwise comparison within the group (Jia et al., 2025). Bozdag et al. (2026) also train persuaders with GRPO; GROVE keeps the group-relative update and contributes rewards that never query a judge during training. Optimizing a learned reward can exploit its errors (Gao et al., 2023), and LLM judges are sensitive to position, length, and presentation (Zheng et al., 2023; Wang et al., 2024; Dubois et al., 2024; Chen et al., 2024), also when judging code (Jiang et al., 2025). GROVE's critic failures and decoupled preference gains are measured instances of these risks.

3 GROVE-Bench: Three Technical Persuasion Environments

GROVE-Bench asks an agent to argue about a program, a factual claim, or a proposed solution before a judge that cannot fully verify it. Each environment defines the context x visible to policy $\pi_\theta(a \mid x)$, the interaction that produces argument a, and the information available to judge J, which compares two arguments or selects a debate winner. Hidden labels z never enter policy or judge prompts, though two training pipelines use them. Table 1 and Figure 2 summarize the environments.

	Code Review	Evidence Debate	Reasoning Challenge
Source	DebugBench	FEVER	SuperGPQA
Judge sees	Problem, program, review, rebuttals	Claim, arguments	Question, solution, challenges
Evaluation units	100 programs	20 claims (11 SUPPORTS)	130 questions
Train–test	72/14/14; winner SFT trains on 87, DPO on all 100	Not trained on; used in validation and selection	Disjoint (390/65/130)
Policies / judgments	19 / 34,200	16 / 2,400	14 / 35,489
Judge	GPT-5.6-Terra	GPT-5.6-Sol, mixed routes	3 Qwen-27B, unanimous

Table 1: **GROVE-Bench at a glance.** Each arena is a complete round robin.

Design. The environments vary the interaction (one reply to a fixed objection, a four-round debate with tool use, or one challenge), what the judge cannot check (execution, retrieved evidence, or an answer key), and the relation to correctness. Half of the code candidates are buggy, so a strong response must sometimes concede; debate stances ignore the label; and the frozen solver is correct on 76 of 130 test questions, so a challenge may face a right or a wrong answer.

Code Review: defend or concede. We draw 100 Python programs from DebugBench (Tian et al., 2024), the dataset-labeled correct and buggy versions of 50 base instances with 43 distinct prompts. Kimi-K2.7-Code writes one fixed review per program, and a strong author response defends valid code or concedes a real defect. Rebuttals are generated independently, cached, and judged in both orders; the judge returns A, B, or tie, with a rationale and optional fabrication flags. The instances have no runnable tests to verify their labels.

Evidence Debate: argue from retrieved evidence. Two advocates receive a FEVER claim (Thorne et al., 2018) and opposite SUPPORTS/REFUTES stances, then alternate for four rounds. We draw 200 claims on which strong models without retrieval failed or were unsure; trained policies learn from 180 of them, and the main arena evaluates the other 20. Each advocate must search before its first argument and may search at most twice; when it issues no valid search, the engine inserts one. The judge sees only the claim and the eight arguments and selects the winning side. SUPPORTS always speaks first, so stance and speaking order are confounded. A seed assigns models to stances.

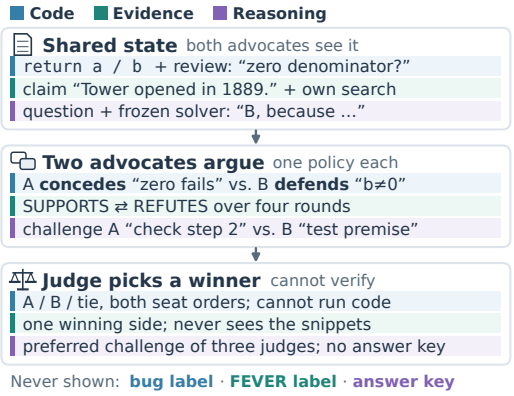

Figure 2: **One protocol, three environments.** Two advocates argue about one technical state before a judge that cannot verify it, and each colored row is one environment. Examples are illustrative.

Reasoning Challenge: argue against a frozen solution. A SuperGPQA question (M-A-P Team et al., 2025) from one of 13 graduate disciplines is paired with a frozen 122B solver's answer and explanation. Two challengers independently argue against the solution, and each of three Qwen 27B judges chooses the stronger challenge. The solver does not answer again, so wins measure argument quality, not answer changes.

Outcomes. We report three outcomes separately, *preference* (the judge's choice), *execution* (tool use and format, recorded in Evidence Debate), and *label checks* (agreement with the candidate label, the winning stance's label, or solver accuracy). Label checks do not verify the arguments.

4 METHODS

The goal of every method is a policy $\pi_\theta(a \mid x)$ whose arguments the environment's judge prefers over those of a fixed field of opponents. We train one policy per environment, using Qwen3.5-9B

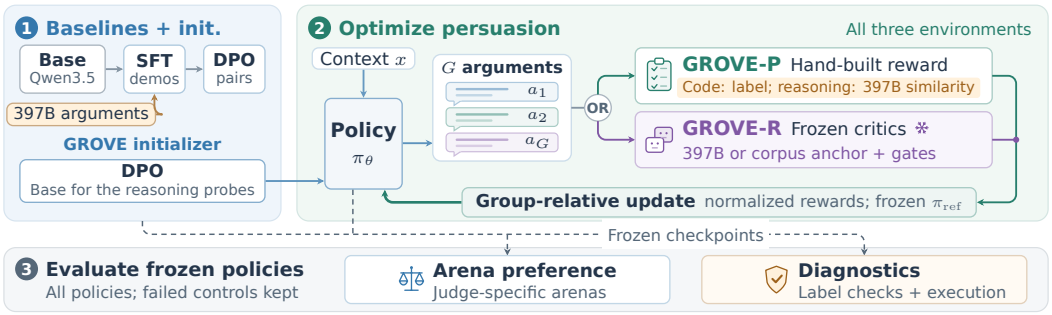

Figure 3: **GROVE training and evaluation.** GROVE starts from DPO, and the two reasoning probes start from the base model. Sibling arguments earn a hand-built reward (GROVE-P) or frozen-critic scores against an anchor argument (GROVE-R). Qwen3.5-397B arguments supply SFT demonstrations and anchors in code and reasoning, and the reasoning GROVE-P target. Frozen checkpoints meet in judge-specific arenas and undergo label checks and execution diagnostics.

	Code Review	**Evidence Debate**	**Reasoning Challenge**
GROVE-P reward	0.4 grounding $+\,0.3$ label-consistent stance $+\,0.2$ review coverage $+\,0.1$ diversity $-\,0.1$ length	0.4 retrieved title $-\,0.3$ absent title $+\,0.2$ opponent coverage $+\,0.1$ diversity $-\,0.1$ length $-\,0.5$ leakage	$0.8\cos(a_k, a_{\text{teacher}})$ $-\,0.1$ repeated 8-grams
GROVE-R anchor	Same-item 397B rebuttal; min of 2 critics	Same-side corpus argument; 1 critic	Teacher argument; min of 2 critics
Eligibility	EOS, JSON, $\leq$180 words, valid code references	End punctuation, $\leq$180 words, no leakage or absent titles	Finished output; $\geq$2 eligible siblings
Init / G / updates	DPO / 6 (P), 8 (R) / 200 (P), 20 (R)	DPO / 8 / 3,000	Base / 6 (P), 4 (R) / $\leq$25

Table 2: **GROVE instantiations.** Reward weights, GROVE-R anchors and eligibility gates, and training budgets per environment; features are defined in Appendix A.

for code and reasoning and Qwen3.5-4B for evidence (Qwen Team, 2026c), and do not test transfer between environments. We compare demonstrations and preference pairs (SFT and DPO) with judge-free rewards over sibling arguments (GROVE). Figure 3 summarizes training and evaluation.

4.1 Baselines: winner SFT and DPO

Winner SFT imitates selected successful arguments, chosen by win rate and fabrication flags in code, by the winning debate side in evidence, and by judge agreement in reasoning. DPO (Rafailov et al., 2023) then raises the reference-relative likelihood of preferred over rejected arguments. Selection rules and preference sources vary per environment (Appendix A). In evidence, standard DPO pairs arguments by claim, stance, and round, so about 99.3% of its rejected responses come from a different context. Label-filtered DPO keeps only pairs whose judged winner argued the labeled side.

4.2 GROVE: Group-Relative Optimization of Verbal Exchanges

GROVE combines standard group-relative policy updates (Shao et al., 2024) with new judge-free rewards. For each training context x, GROVE samples G sibling arguments $a_1, \ldots, a_G \sim \pi_\theta(\cdot \mid x)$, scores each with a reward r_k, and reinforces arguments that outscore their siblings. Table 2 lists the rewards, eligibility rules, and budgets of the two variants in each environment.

GROVE-P: judge-free programmatic rewards. GROVE-P scores each sibling with a weighted sum $r_k^{\text{P}} = \sum_m w_m\, q_m(x, a_k, a_{-k})$ of text features q_m, where a_{-k} are its siblings. We set the weights w_m in Table 2 by hand and test neither an ablation nor whether r_k^{P} predicts judge wins. In code, one term reads the hidden dataset label and rewards defending correct code and conceding

defects in buggy code, so this reward is *label-aware*. The other code terms are lexical and do not verify correctness. Evidence rewards citing retrieved titles and penalizes absent titles and protocol leakage. For the SUPPORTS opening argument, the opponent-coverage term reads REFUTES' first-round reply, which the policy never sees. The sampler allots this turn 1,500 of the 12,000 training prompts (12.5%). Reasoning rewards embedding similarity to a teacher argument.

GROVE-R: anchored critics. GROVE-R trains critics on preference pairs, freezes them, and scores each candidate against a cached *anchor argument* $a^\star$ for the same context. With standardized critic scores $\tilde{s}_j$ and logistic function σ, an eligible candidate earns

$$r_k^{\mathrm{R}} = \min_j \sigma(\tilde{s}_j(x, a_k) - \tilde{s}_j(x, a^\star)), \tag{1}$$

its lowest score against the anchor across critics, so a high reward requires every critic to prefer it. Ineligible candidates rank below all eligible siblings, and groups without one are skipped.

Policy update and checkpoint gates. Rewards are normalized within each group as $A_k = (r_k - \bar{r})/\hat{\sigma}_r$, with denominator one when the variance is zero. The update raises the likelihood of arguments with positive A_k and lowers it for those with negative A_k, with a per-token KL penalty toward a frozen reference policy. Code uses our own trainer without ratio clipping, and evidence and reasoning use TRL's group-relative trainer (von Werra et al., 2020). Checkpoints that fail an objective or gate, including all GROVE-R runs and the reasoning GROVE-P probe, carry a † wherever we report them. Appendix A gives the code loss, and Algorithm 1 gives the full procedure.

5 Experimental Setup

Section 6.1 asks whether trained small agents rival larger ones, Section 6.2 what GROVE adds to SFT and DPO, and Section 6.3 whether wins reflect execution and agreement with dataset labels.

Models and baselines. Frozen models span Qwen3.5 (2B–397B), Qwen3.6-27B, and Gemma-4 (E2B, E4B, 31B). The baselines imitate winners (SFT), optimize preference pairs (DPO), filter these pairs by label or source (filtered DPO), or, in evidence only, optimize a Qwen3.5-27B judge's verdict from winner SFT (judge-reward GRPO). Main tables show winner SFT in code (286 winning rebuttals) and evidence and unanimous-teacher SFT in reasoning. No rule for choosing these runs was fixed in advance; the four reasoning policies excluded after evaluation win 47.4–51.8% against base 9B (three-judge mean), so none exceeds it by more than 1.8 pp. DPO starts from them, except in evidence, where it starts from a conversation SFT that fails on format. GROVE starts from DPO in code and evidence and from base in reasoning.

Exposure and supervision. The code and evidence arenas evaluate on items used in training or model selection, so they measure specialization to fixed items. We split the 100 code programs by base instance into 72 training, 14 development, and 14 test programs. The critics, source-filtered DPO, and GROVE-R do not train on the test programs, but one test program duplicates a training program. Winner SFT trains on 87 programs and DPO on all 100. The code judge, GPT-5.6-Terra, is held out from training: Qwen3.7-Max (Alibaba Cloud, 2026) and DeepSeek-V4-Pro (DeepSeek-AI, 2026) label the training data, and Terra labels none (outside the arena it judged only the GROVE-R pilot, one DPO gate, and 600 calibration comparisons). In evidence, the arena judge's verdicts on base-versus-base debates select the SFT winners and DPO pairs. The 20 test claims, held out from training, validate winner SFT and help select the source models of conversation SFT. Reasoning splits its SuperGPQA questions 390/65/130 into training, validation, and test before generation, and the same judge family supervises training and scores the test. Hidden labels reach training only through the code GROVE-P reward and evidence label-filtered DPO.

Judging. Code uses GPT-5.6-Terra in both presentation orders, and it favors the first position (53.7% of decisive judgments), yet order-consistent verdicts give nearly the same ranking (Kendall $\tau = 0.99$; Appendix C.1). Evidence uses GPT-5.6-Sol in a main arena (16 policies, 20 claims) and a diagnostic arena (15 policies, all 200 claims, including the 180 training claims) that records execution statistics. The 300 main-arena comparisons involving 397B use a separate API route, and some base-versus-base verdicts have unresolved judge records. Reasoning uses Qwen3.5-, 3.6-,

and 3.8-27B judges, and Qwen3.5-27B also competes. Each pair and question gets one random argument order, so reasoning does not measure position bias. Main results count only unanimous decisions (78.0% of comparisons), and Appendix C.3 reports single judges and majority votes.

Metrics. A head-to-head win rate is $(W + T/2)/(W + T + L)$, where ties occur only in code, and a policy's field win rate pools these counts over all its comparisons in its arena. Bradley–Terry ratings on the Elo scale (Bradley & Terry, 1952) summarize the same comparisons (mean 1500 in code and reasoning, base 4B at 800 in evidence). The search rate is the share of a policy's diagnostic debates against the seven frozen models that need an engine-inserted first search, the completion rate the share of its reasoning validation arguments that finish within the token cap, and the flag rate the share of its code judgments that flag fabrication. We report **estimate ± bootstrap SD** over complete task clusters (50 code base instances, 20 claims, or 130 questions), keeping all opponents, seats, and judges inside each cluster. This SD covers task sampling for one training run and one set of outputs, so it can understate evaluation noise, as evidence GROVE-R, trained without any usable reward, scores 9.7 pp above its DPO initializer. Model size counts total parameters, including embeddings for Gemma-4 E2B/E4B (5.1B/8B; Google DeepMind, 2026).

6 Results and Analysis

6.1 Training makes small agents competitive in code and evidence

We ask whether training makes small policies rival larger frozen models on training or selection items and on held-out reasoning questions. Table 3 reports field win rates, and Figure 4d–f gives every head-to-head result. In the Elo ratings of Figure 1, 397B leads every environment, and GROVE-P ranks second in code and evidence, trailing 397B's field win rate by 5.0 pp [−2.7, 12.5] and 3.0 pp [−7.3, 13.7].

Policy Backbone	Code 9B	Evidence 4B	Reasoning 9B
Base backbone	42.9 ± 2.4	41.3 ± 2.7	57.9 ± 2.3
SFT	60.1 ± 2.3	50.3 ± 2.1	56.1 ± 2.2
DPO	59.8 ± 2.5	53.7 ± 2.8	54.1 ± 2.2
Filtered DPO	62.7 ± 2.0	62.3 ± 2.7	—
GROVE-P	$63.0 \pm 2.0^{*}$	70.3 ± 2.9	$57.9 \pm 2.3^{\dagger}$
GROVE-R	$52.1 \pm 2.1^{\dagger}$	$63.3 \pm 2.6^{\dagger}$	$56.0 \pm 2.2^{\dagger}$
Qwen-122B	47.5 ± 2.5	49.3 ± 3.1	67.9 ± 2.4
Qwen-397B	68.0 ± 2.7	73.3 ± 3.7	84.1 ± 2.1
GROVE-P − base (pp)	+20.1	+29.0	+0.1
GROVE-P − control (pp)	+3.2	+7.0	+0.1

Table 3: **Field win rates** (%; estimate ± bootstrap SD). Code and evidence test on training or selection items; reasoning uses held-out questions and unanimous decisions. SFT is winner SFT (unanimous-teacher SFT in reasoning); evidence DPO starts from conversation SFT, which scores 15.7%. Filtered DPO is source-filtered (DeepSeek labels) in code and label-filtered in evidence. The control shares GROVE-P's initializer (DPO; zero-reward GROVE-R; base), and evidence gaps below 10 pp are unresolved. Differences use unrounded rates. *label-aware reward; †failed an objective or gate.

Code Review. Under the GPT-5.6-Terra judge, training with SFT, DPO, and GROVE-P raises the 9B field win rate from $42.9 \pm 2.4\%$ to $63.0 \pm 2.0\%$ (+20.1 pp; +46.9%) on the 100 programs that DPO trained on. GROVE-P beats every frozen model up to 9B (67.8–95.2% head-to-head; $67.8 \pm 3.8\%$ against base) and 122B ($61.5 \pm 3.8\%$), matches Qwen3.6-27B and Gemma-4-31B ($51.0 \pm 3.9\%$; $54.3 \pm 4.1\%$), and does not beat 397B ($42.3 \pm 4.0\%$). Its judge fabrication-flag rate is the lowest of all 19 code policies (14.6%, versus 16.4% for 397B and 30.6% for base 9B). This flag is unverified, and the grounding reward and fabrication-filtered SFT data partly target what it checks.

Evidence Debate. On 20 claims used for model selection but not training, the 4B pipeline raises the field win rate from $41.3 \pm 2.7\%$ to $70.3 \pm 2.9\%$ (+29.0 pp; +70.2%). Head-to-head, GROVE-P beats every other frozen model, 27B and 122B included (85–100%), but not 397B ($35.0 \pm 10.7\%$, separate judge route). Against the other trained 4B policies, it wins 25–75% and loses only to winner SFT epoch 1 (25%, [5, 45]). On 200 claims, it loses to label-filtered DPO, both winner-SFT checkpoints, and judge-reward GRPO (36.0–37.5%) and ties DPO and GROVE-R (Wilson intervals).

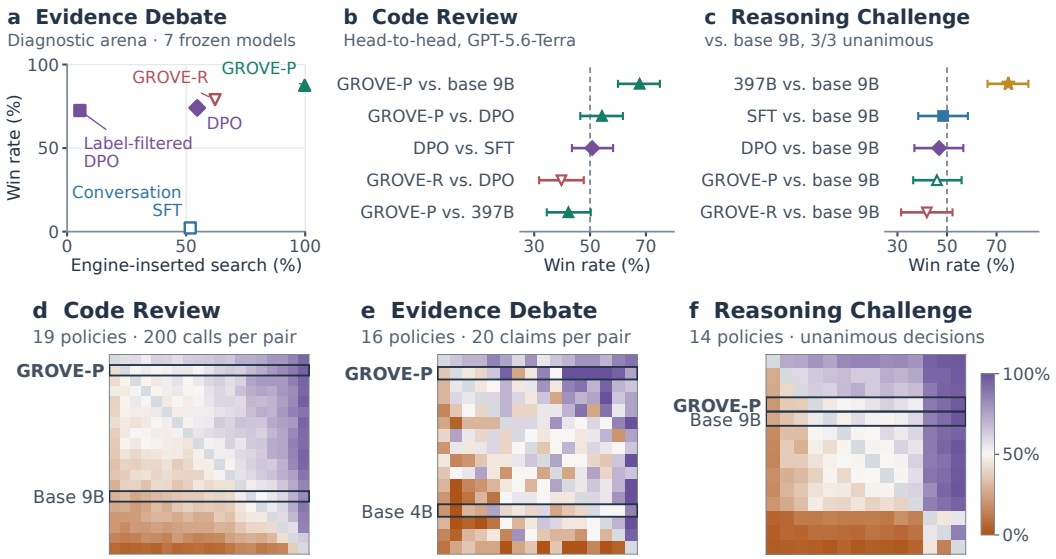

Figure 4: **Preference gains need not improve execution.** (a) Diagnostic evidence arena (200 claims), win rate against the seven frozen models other than 397B versus the share of debates with an engine-inserted first search. (b, c) Code and reasoning head-to-head win rates with 95% intervals. (d–f) Each row policy's win rate against each column policy, strongest first (reasoning: unanimous decisions; evidence 397B pairs: separate judge route), with GROVE-P and its base outlined; Figure 9 names every row. Hollow markers denote runs that failed an objective or gate.

Reasoning Challenge. On held-out questions 397B leads under every judge, reaching $84.1 \pm 2.1\%$ under unanimous agreement. No trained policy differs from base 9B ($57.9 \pm 2.3\%$), as every head-to-head interval against it spans parity (Figure 4c). Unanimous-teacher SFT reaches $56.1 \pm 2.2\%$, its DPO $54.1 \pm 2.2\%$, and GROVE-P, a 10-update probe that missed its 95% completion gate, $57.9 \pm 2.3\%$. With at most 25 updates from base, versus 200 and 3,000 from DPO in code and evidence, this null says nothing about GROVE at a matched budget.

6.2 Contribution of each training stage

In code, winner SFT reaches $60.1 \pm 2.3\%$, which is 17.2 pp above base (95% interval [10.9, 23.4]) and most of the pipeline's gain. Although 80 of its 286 targets are 397B's arguments on these programs, it trails 397B ($38.5 \pm 3.7\%$). GROVE-P adds 3.2 pp over its DPO initializer (95% interval $[-2.9, 9.3]$) and wins $54.3 \pm 3.8\%$ head-to-head against it. Both intervals include parity, the head-to-head one also when we resample the 43 distinct prompts. Source-filtered code DPO comes close, at 61.9% (Qwen labels) and 62.7% (DeepSeek labels; 51.0% head-to-head for GROVE-P). With label access differing across rewards, these compare training pipelines, not matched supervision.

In evidence, GROVE-P leads its DPO initializer by 16.7 pp in field win rate (95% interval [8.0, 25.3]). GROVE-R shows what training without reward does: it starts from the same DPO checkpoint and receives no usable reward, yet its language-model weights move 0.4% from DPO's in relative L_2 norm (GROVE-P's move 2.0%), and it scores $63.3 \pm 2.6\%$ against DPO's $53.7 \pm 2.8\%$ while tying DPO head-to-head. Evidence gaps below about 10 pp therefore cannot be credited to a reward, including GROVE-P's 7.0 pp lead over GROVE-R, which shrinks to 0.6 pp on 200 claims. GROVE-P's lead over DPO is not separable from evaluation variation either: it ties DPO head-to-head (65% [45, 85] on 20 claims; 44.0% [37.3, 50.9] on 200), and the lead shrinks to 3.1 pp on 200 claims. Judge-reward GRPO reaches $49.0 \pm 3.5\%$.

6.3 Judged wins can hide execution and training failures

In the 200-claim diagnostic arena, GROVE-P wins 87.6% of 1,400 debates against seven frozen models, as Figure 4a shows, but in 1,397 of them (99.8%) the engine inserts the required first search

as GROVE-P issues no valid one. GROVE-P's training never optimized the search action, so this rate may reflect the training format. Label-filtered DPO wins nearly as often as standard DPO (72.5% vs. 74.1%) with far less inserted search (5.4% vs. 54.6%). The winning stance agrees with FEVER labels at 49.1–55.5% balanced accuracy for the five trained policies with execution records, against about 71% in debates between frozen models (Appendix C.2). Near-chance agreement is expected for a policy that wins on either side: if stances are balanced within labels and GROVE-P wins equally across labels, its balanced accuracy must lie within 37.6–62.4%.

GROVE-R fails its intended optimization in every environment. Evidence GROVE-R receives zero usable reward, as every logged reward call fails an eligibility gate or a configuration check, yet it ranks third of 16 models. Its rank therefore cannot come from the critic reward, and arena rank alone would hide the failure. Code GROVE-R stops at update 20 while its critic score still rises, because it wins 25.0% of a 40-case pilot against DPO. Its reward gate admits only rebuttals of at most 180 words, and its final checkpoint writes short rebuttals and wins only 39.8% head-to-head against DPO. Reasoning GROVE-R, a 25-update probe, misses its completion gate (95% of validation outputs must finish), and its critics fall below their 75% accuracy threshold.

7 Discussion and Limitations

Why small trained agents win. The code gains survive a change of judge family. Winner SFT and DPO learn from Qwen3.7-Max and DeepSeek-V4-Pro labels, yet GPT-5.6-Terra, which labeled no training data, prefers the trained arguments, and Terra agrees with DeepSeek-V4-Pro on 78.6% of the calibration cases both decide (Cohen's $\kappa = 0.57$). Winner imitation thus transfers qualities that both judge families prefer, not one evaluator's taste. The gains also concentrate where arguments engage the artifact. On dataset-buggy programs GROVE-P scores 67.4%, level with 397B (67.3%), and its remaining gap lies on correct programs (58.6% against 68.7%), where sound code must be defended against a critical review. DPO on DeepSeek labels, which never sees the dataset labels, shows the same pattern (67.1% on buggy programs), so judged preference alone can teach a policy when to concede a real defect. GROVE-P also draws the fewest fabrication flags of all 19 code policies, consistent with a reward that credits references matching the submitted source. In evidence, a 4B policy beats the 27B and 122B models (95% and 85% head-to-head), plausibly because its reward credits citing retrieved titles and answering the opponent's terms, while the engine runs its searches. These rankings withstand known judge biases: order-consistent verdicts reproduce the code ranking (Kendall $\tau = 0.988$), and the Qwen3.5-27B judge that also competes in reasoning ranks second under the other two judges as well. Because every arena also records execution and label agreement, GROVE-Bench exposes what rank alone hides, such as a zero-reward run ranked third or wins without searches.

Limitations. Code and evidence evaluate on items used in training or model selection, so their gains measure specialization rather than transfer, and the code GROVE-P reward reads dataset labels. Results come from single runs with unmatched budgets and no best-of-n or prompting baseline. All judges are LLMs without human calibration, and we do not measure persuasion of people.

8 Conclusion

GROVE-Bench shows that persuasion in technical disputes can be trained and audited in the same arenas. On items used in training or model selection, SFT, DPO, and the judge-free GROVE-P reward lift a 9B code agent by 46.9% and a 4B evidence agent by 70.2% in field win rate, to within 5 pp of a 397B model and above every other frozen model, under a code judge that labeled no training data. The gains come from arguments that engage the artifact: in code they match 397B on buggy programs and draw the fewest fabrication flags. Because the arenas also record execution and agreement with dataset labels, they reveal what win rates alone miss, such as an evidence agent that wins while the engine runs its searches. Held-out reasoning shows no gain from probes of at most 25 updates, so held-out items, full training budgets, repeated runs, and human-calibrated judges are the natural next steps toward rewarding arguments that are both preferred and grounded.

AI USE STATEMENT

In this work, we used generative AI tools to setup the environment as per our proposal, to give feedback on evaluation audits. Additionally, we used generative AI tools to create and edit and modify scientific figures as per obtained results, improve paper's grammar and coherence. The experiments themselves use language models as advocates, judges, and reward critics, as specified in the paper. Cited works were checked against their sources. We take responsibility for the final content of this work.

ETHICS STATEMENT

Optimizing persuasion can improve technical feedback and can also help a model defend false claims. We therefore report label access, execution failures, and the distinction between preferred arguments and agreement with dataset labels. The study uses technical tasks and model participants, not human-subject persuasion. Its outcomes should not be interpreted as evidence of safe influence on people or as authorization to deploy misleading advocacy. Releasing evaluation protocols and documented failures supports scrutiny of both beneficial and harmful uses.

REPRODUCIBILITY STATEMENT

Appendix A defines the objectives, ratings, and uncertainty estimates and specifies every training run, reward, and gate; Appendix B gives the data splits, exposure, judging, and generation settings; and Appendix C reports the complete arenas. The supplementary material contains every prompt verbatim, the frozen comparison data, and deterministic scripts. Exact replay of the original model calls is limited by live Wikipedia retrieval, hosted judge routes, and incomplete provenance for some training runs and evidence judgments (Appendices A.2 and B.1).

REFERENCES

Alibaba Cloud. Qwen3.7-Max model information. Alibaba Cloud Model Studio, 2026. URL https://www.alibabacloud.com/help/en/model-studio/qwen3-7-max.

Samuel Arnesen, David Rein, and Julian Michael. Training language models to win debates with self-play improves judge accuracy. *arXiv preprint arXiv:2409.16636*, 2024. URL https://arxiv.org/abs/2409.16636.

Nimet Beyza Bozdag, Emre Can Acikgoz, Gokhan Tur, and Dilek Hakkani-Tür. Learning to persuade exposes how easily LLMs abandon correct beliefs. *arXiv preprint arXiv:2608.11624*, 2026. URL https://arxiv.org/abs/2608.11624.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Aaryan Chandna, Lukas Fluri, and Micah Carroll. Is the evidence in 'Language Models Learn to Mislead Humans via RLHF' valid? In *ICLR Blogposts Track*, 2026. URL https://iclr.cc/virtual/2026/poster/10012085.

Yapei Chang, Yekyung Kim, Michael Krumdick, Amir Zadeh, Chuan Li, Chris Tanner, and Mohit Iyyer. BLEUBERI: BLEU is a surprisingly effective reward for instruction following. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2505.11080.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. Humans or LLMs as the judge? a study on judgement bias. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8301–8327, 2024. doi: 10.18653/v1/2024.emnlp-main.474. URL https://aclanthology.org/2024.emnlp-main.474/.

Pengyu Cheng, Tianhao Hu, Han Xu, Zhisong Zhang, Zheng Yuan, Yong Dai, Lei Han, Nan Du, and Xiaolong Li. Self-playing adversarial language game enhances LLM reasoning. In *Advances in*

Neural Information Processing Systems (NeurIPS), 2024. URL https://arxiv.org/abs/2404.10642. arXiv:2404.10642.

Zirui Cheng and Jiaxuan You. Towards strategic persuasion with language models. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2509.22989.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026. URL https://arxiv.org/abs/2606.19348.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-controlled AlpacaEval: A simple way to debias automatic evaluators. In *Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2404.04475.

Arpad E. Elo. *The Rating of Chessplayers, Past and Present*. Arco Publishing, New York, 1978.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning (ICML)*, volume 202, pp. 10835–10866. PMLR, 2023. URL https://proceedings.mlr.press/v202/gao23h.html.

Google DeepMind. Gemma 4 model card. Google AI for Developers, 2026. URL https://ai.google.dev/gemma/docs/core/model_card_4. Accessed September 7, 2026; total and effective parameter counts.

Kobi Hackenburg, Ben M. Tappin, Paul Röttger, Scott Hale, Jonathan Bright, and Helen Margetts. Evidence of a log scaling law for political persuasion with large language models. *arXiv preprint arXiv:2406.14508*, 2024. URL https://arxiv.org/abs/2406.14508.

Peixuan Han, Zijia Liu, and Jiaxuan You. ToMAP: Training opponent-aware LLM persuaders with theory of mind. *arXiv preprint arXiv:2505.22961*, 2025. URL https://arxiv.org/abs/2505.22961.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Geoffrey Irving, Paul Christiano, and Dario Amodei. AI safety via debate. *arXiv preprint arXiv:1805.00899*, 2018. URL https://arxiv.org/abs/1805.00899.

Ruipeng Jia, Yunyi Yang, Yongbo Gai, Kai Luo, Shihao Huang, Jianhe Lin, Xiaoxi Jiang, and Guanjun Jiang. Writing-Zero: Bridge the gap between non-verifiable tasks and verifiable rewards. arXiv:2506.00103, 2025. URL https://arxiv.org/abs/2506.00103.

Hongchao Jiang, Yiming Chen, Yushi Cao, Hung-yi Lee, and Robby T. Tan. CodeJudgeBench: Benchmarking LLM-as-a-judge for coding tasks. arXiv:2507.10535, 2025. URL https://arxiv.org/abs/2507.10535.

Zachary Kenton, Noah Y. Siegel, János Kramár, Jonah Brown-Cohen, Samuel Albanie, Jannis Bulian, Rishabh Agarwal, David Lindner, Yunhao Tang, Noah D. Goodman, and Rohin Shah. On scalable oversight with weak LLMs judging strong LLMs. *arXiv preprint arXiv:2407.04622*, 2024. URL https://arxiv.org/abs/2407.04622.

Zachary Kenton, Lili Janzer, Rory Greig, Tian Huey Teh, Kirill Tyshchuk, Jonah Brown-Cohen, Harri Edwards, Senthooran Rajamanoharan, Noah Y. Siegel, Natasha Jaques, and Rohin Shah. Debate training reduces reward hacking in RLAIF. *arXiv preprint arXiv:2608.17776*, 2026. URL https://arxiv.org/abs/2608.17776.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rocktäschel, and Ethan Perez. Debating with more persuasive LLMs leads to more truthful answers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 23662–23733. PMLR, 2024. URL https://proceedings.mlr.press/v235/khan24a.html.

Jan Hendrik Kirchner, Yining Chen, Harri Edwards, Jan Leike, Nat McAleese, and Yuri Burda. Prover-verifier games improve legibility of LLM outputs. arXiv:2407.13692, 2024. URL https://arxiv.org/abs/2407.13692.

M-A-P Team, Xinrun Du, Yifan Yao, Kaijing Ma, et al. SuperGPQA: Scaling LLM evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*, 2025. URL https://arxiv.org/abs/2502.14739.

Nat McAleese, Rai Michael Pokorny, Juan Felipe Ceron Uribe, Evgenia Nitishinskaya, Maja Trebacz, and Jan Leike. LLM critics help catch LLM bugs. arXiv:2407.00215, 2024. URL https://arxiv.org/abs/2407.00215.

Julian Michael, Salsabila Mahdi, David Rein, Jackson Petty, Julien Dirani, Vishakh Padmakumar, and Samuel R. Bowman. Debate helps supervise unreliable experts. arXiv:2311.08702, 2023. URL https://arxiv.org/abs/2311.08702.

Moonshot AI. Kimi-K2.7-Code model card. Hugging Face, 2026. URL https://huggingface.co/moonshotai/Kimi-K2.7-Code.

OpenAI. GPT-5.6 Sol model documentation. OpenAI API documentation, 2026a. URL https://developers.openai.com/api/docs/models/gpt-5.6-sol.

OpenAI. GPT-5.6 Terra model documentation. OpenAI API documentation, 2026b. URL https://developers.openai.com/api/docs/models/gpt-5.6-terra.

Qwen Team. Qwen3.5-397B-A17B model card. Hugging Face, 2026a. URL https://huggingface.co/Qwen/Qwen3.5-397B-A17B.

Qwen Team. Qwen3.5-27B model card. Hugging Face, 2026b. URL https://huggingface.co/Qwen/Qwen3.5-27B.

Qwen Team. Qwen3.5-9B and Qwen3.5-4B: Official model cards. https://huggingface.co/Qwen/Qwen3.5-9B, 2026c. Companion 4B card: https://huggingface.co/Qwen/Qwen3.5-4B.

Qwen Team. Qwen3.6-27B model card. Hugging Face, 2026d. URL https://huggingface.co/Qwen/Qwen3.6-27B.

Qwen Team. Qwen3.8-27B model card. Hugging Face, 2026e. URL https://huggingface.co/Qwen/Qwen3.8-27B.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2305.18290.

Francesco Salvi, Manoel Horta Ribeiro, Riccardo Gallotti, and Robert West. On the conversational persuasiveness of GPT-4. *Nature Human Behaviour*, 9:1645–1653, 2025. doi: 10.1038/s41562-025-02194-6. URL https://www.nature.com/articles/s41562-025-02194-6.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. Towards understanding sycophancy in language models. arXiv:2310.13548, 2023. URL https://arxiv.org/abs/2310.13548.

Somesh Kumar Singh, Yaman Kumar Singla, Harini S I, and Balaji Krishnamurthy. Measuring and improving persuasiveness of large language models. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=NfCEVihkdC. Introduces PersuasionBench and PersuasionArena; arXiv:2410.02653.

Elias Stengel-Eskin, Peter Hase, and Mohit Bansal. Teaching models to balance resisting and accepting persuasion. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics (NAACL)*, 2025. URL https://arxiv.org/abs/2410.14596.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and verification. In *Proceedings of NAACL-HLT*, pp. 809–819, 2018. doi: 10.18653/v1/N18-1074. URL https://aclanthology.org/N18-1074/.

Runchu Tian, Yining Ye, Yujia Qin, Xin Cong, Yankai Lin, Yinxu Pan, Yesai Wu, Haotian Hui, Weichuan Liu, Zhiyuan Liu, and Maosong Sun. DebugBench: Evaluating debugging capability of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 4173–4198. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-acl.247. URL https://aclanthology.org/2024.findings-acl.247/.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers reinforcement learning. GitHub repository, 2020. URL https://github.com/huggingface/trl.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large language models are not fair evaluators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9440–9450, 2024. doi: 10.18653/v1/2024.acl-long.511. URL https://aclanthology.org/2024.acl-long.511/.

Jiaxin Wen, Ruiqi Zhong, Akbir Khan, Ethan Perez, Jacob Steinhardt, Minlie Huang, Samuel R. Bowman, He He, and Shi Feng. Language models learn to mislead humans via RLHF. arXiv:2409.12822, 2024. URL https://arxiv.org/abs/2409.12822.

Peng Xia, Kaide Zeng, Jiaqi Liu, Can Qin, Fang Wu, Yiyang Zhou, Caiming Xiong, and Huaxiu Yao. Agent0: Unleashing self-evolving agents from zero data via tool-integrated reasoning. arXiv:2511.16043, 2025. URL https://arxiv.org/abs/2511.16043.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. SOTOPIA: Interactive evaluation for social intelligence in language agents. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.11667.

A METHOD DETAILS

A.1 OBJECTIVES, SCORES, AND UNCERTAINTY

A policy π_θ maps the visible context x to an argument a. For judge J and an opposing argument b, the outcome $U_J(x, a, b) \in \{0, \frac{1}{2}, 1\}$ is a loss, tie, or win (ties exist only in Code Review), and every method targets the judged preference $\mathbb{E}\, U_J$ over an environment's items and opponents. SFT minimizes the negative log likelihood of selected winning arguments, DPO (Rafailov et al., 2023) uses $\beta = 0.1$, and GROVE minimizes Eq. 2 in code, without ratio clipping or an entropy bonus, and uses the group-relative trainer of TRL (von Werra et al., 2020) in evidence and reasoning. Kimi-K2.7-Code (Moonshot AI, 2026) writes the fixed code reviews, GPT-5.6-Terra (OpenAI, 2026b) and GPT-5.6-Sol (OpenAI, 2026a) judge code and evidence, and Qwen3.5-, 3.6-, and 3.8-27B (Qwen Team, 2026b;d;e) judge reasoning. The largest frozen model is Qwen3.5-397B-A17B (Qwen Team, 2026a). In code, GROVE minimizes

$$\mathcal{L} = \frac{1}{BG} \sum_{i,k} \frac{1}{T_{ik}} \sum_{t} \Big[- A_{ik} \log \pi_\theta(a_{ikt} \mid h_{ikt}) + \lambda K_{ikt} \Big], \tag{2}$$

where B counts contexts per batch, T_{ik} is the token length of argument a_{ik}, h_{ikt} is the context and prefix, K_{ikt} is a per-token KL estimate toward the frozen reference policy π_{ref}, and $\lambda = 0.05$.

A policy's field win rate pools its outcomes over all of its comparisons in its arena, $(W + T/2)/(W + T + L)$ for W wins, T ties, and L losses. Bradley–Terry ratings (Bradley & Terry, 1952) on the Elo scale (Elo, 1978), $P(i \succ k) = \left(1 + 10^{(R_k - R_i)/400}\right)^{-1}$, are fitted by maximum likelihood. They are centered at 1500 in code and reasoning and fix base 4B at 800 in evidence, so absolute ratings are not comparable across environments.

Uncertainty. Intervals resample task clusters, not model calls. In code, the cluster is one of 50 base instances, keeping both variants, both seats, and all opponents together; a second analysis clusters the 43 distinct prompts. In evidence, it is one of 20 main-arena claims, keeping all 120 pairs together and refitting Bradley–Terry each time. In reasoning, it is one of 130 questions, keeping all judges together. We use 10,000 replicates (seed 20260921) and report bootstrap SD or percentile 95% intervals. Differences are computed from unrounded rates. The code intervals in Figure 4b use the same resampling with seed 0. These condition on the frozen arguments and judging routes and exclude training-seed variation. In the 200-claim diagnostic arena, each pair of policies debates every claim once, so we give each head-to-head rate a Wilson 95% interval over its 200 debates. Field win rates in this arena remain point estimates, because the frozen data keep only pair totals, not per-claim outcomes. The spread across reasoning judges measures evaluator sensitivity; it is not a confidence interval.

A.2 ALL TRAINING RUNS

Table 4 lists every trained policy. Code uses LoRA rank 32 ($\alpha = 64$) and reasoning rank 16 ($\alpha = 32$), both with dropout 0.05 on the attention and MLP projections (Hu et al., 2022); evidence updates all parameters. The code GROVE-P run has no frozen manifest, so its settings are reconstructed. Code SFT and DPO use AdamW with gradient accumulation 16 (DPO $\beta = 0.1$), and code GROVE-P and GROVE-R use 1,152- and 1,536-token sequences. Evidence conversation SFT uses an effective batch of 12 and a 6,144-token cutoff, and evidence GROVE uses 400-token completions with KL coefficient 0.05. Reasoning SFT uses an effective batch of 16, 9,000-token sequences, and early stopping; the reasoning probes use gradient clipping at 1.0, accumulation 12 (P) and 8 (R), a 4,096-token completion cap, and temperature 1.0. We do not fill unrecorded settings, such as realized initializer hashes, with presumed defaults.

A.3 REWARDS AND GATES

Code Review. GROVE-P scores each sibling with

$$r^{\text{P}}_{\text{CR}} = 0.40 q_g + 0.30 q_z + 0.20 q_c + 0.10 q_d - 0.10 q_l, \tag{3}$$

where q_g rewards code references that match the submitted source; q_z rewards conceding on dataset-buggy and defending on dataset-correct programs (the label enters the reward builder, never the

Run	Initializer	Training data or reward	LR	Budget	Outcome
Code Review (Qwen3.5-9B, LoRA)					
Winner SFT	Base	286 winning rebuttals on 87 programs, validated on 39 from the other 13 (Qwen win rate $\geq$60%, flagged fabrication removed)	$2 \cdot 10^{-4}$	3 epochs	Main-table SFT
SFT, all $\geq$27B authors; winner SFT, unfiltered; SFT, fabrication-only	Base	299 rebuttals from $\geq$27B authors; the 288 winner-SFT rebuttals before the fabrication filter, which removes 2; 145 flagged rebuttals, only 2 of them among the 288	$2 \cdot 10^{-4}$	3 epochs	Controls
DPO	Winner SFT	2,447 seat-consistent Qwen pairs, length-controlled	$5 \cdot 10^{-6}$	2 epochs	GROVE initializer
Source-filtered DPO	Winner SFT	Qwen / DeepSeek / 3-of-4 pairs on a 36/7/7 base split	$5 \cdot 10^{-6}$	2 / 1 / 1 epochs	3-of-4 fails output audit
GROVE-P	DPO	3 prompts $\times$ 6 siblings; label-aware text-feature reward (Eq. 3)	$5 \cdot 10^{-6}$	200 steps	Settings reconstructed
GROVE-R	DPO	3 prompts $\times$ 8 siblings; minimum of two critics vs. a 397B anchor (Eq. 1)	10^{-6}	20 steps	Stopped: pilot 25.0%
Evidence Debate (Qwen3.5-4B, full parameters)					
Winner SFT	Base	Winning sides of the 4,200 base-versus-base debates; trains on 180 claims, validates on the 20 test claims	—	1 / 3 epochs	Main-table SFT (epoch 3)
Judge-reward GRPO	Winner SFT, epoch 3	Live Qwen3.5-27B pairwise verdict vs. the corpus opponent ($\pm$1), plus length, citation, and protocol terms	10^{-6}	1 epoch, 3,000 steps	—
Conversation SFT	Base	2,061 conversations, 10,939 target turns	10^{-5}	3 epochs	Format failure
DPO; label-filtered DPO	Conversation SFT	8,877 / 4,452 pairs matched by claim, stance, round (99.3% cross-context)	$5 \cdot 10^{-6}$	2 epochs	Filter uses labels
GROVE-P	DPO‡	12,000 prompts $\times$ 8 siblings; text-feature reward (Eq. 4)	$5 \cdot 10^{-6}$	3,000 steps	99.8% inserted first search
GROVE-R	DPO‡	Same prompts; one critic vs. a same-side corpus anchor	10^{-6}	3,000 steps	Zero usable reward
Reasoning Challenge (Qwen3.5-9B, LoRA)					
Majority-, tournament-, unanimous-teacher SFT	Base	329 / 298 / 103 teacher arguments (majority / tournament / unanimous)	$2 \cdot 10^{-5}$	$\leq$10 epochs	Validation-loss selection
DPO	Unanimous-teacher SFT	1,722 unanimously labeled pairs	$5 \cdot 10^{-6}$	2 epochs	—
GROVE-P probe	Base	239 questions $\times$ 6 siblings; teacher-argument similarity	10^{-6}	10 of $\leq$25 updates	Completion 93.8% $<$ 95% gate
GROVE-R probe	Base	239 questions $\times$ 4 siblings; critics at 71.9 / 68.1% ($<$75% gate)	10^{-6}	25 updates	Completion 92.3% $<$ 95% gate

Table 4: **All training runs.** —: not recorded or not applicable. ‡: initializer checkpoint not pinned by hash. Budgets are not matched across environments or method families; each run is a single training seed.

prompt); q_c rewards covering the review's terms; q_d rewards lexical diversity among siblings; and q_l penalizes length. These terms are lexical proxies, not execution checks. GROVE-R compares two frozen Qwen3.5-4B critics, trained on the 36 training base instances, with a cached 397B anchor argument $a^\star$ (Eq. 1). Training stopped at update 20, although the critic score was still rising, because a 40-case pilot against DPO gave 25.0%.

Evidence Debate. GROVE-P scores each turn with

$$r_{\mathrm{ED}}^{\mathrm{P}} = 0.4g - 0.3f + 0.2c + 0.1d - 0.1\ell - 0.5p. \tag{4}$$

The reward credits mentioning a retrieved title (g), covering the opponent's terms (c), and diversity (d). It penalizes citing a title absent from the evidence (f), overlong turns (ℓ), and protocol leakage (p). Title matching does not establish entailment. GROVE-R draws 4 prompts $\times$ 8 siblings per step for 3,000 steps on two training ranks, which gives 96,000 rollouts. Its 48,000 logged reward calls equal one rank's share of 16 calls per step, because each rank keeps its own counters. Every logged call failed a gate (almost all on termination or length) or a configuration check. Both ranks share these gates, and the training log records zero reward at every step. With zero advantages only the KL penalty contributes gradients, yet the checkpoint's language-model weights move 0.4% from its DPO initializer in relative L_2 norm (GROVE-P's move 2.0%). The configured initializer path holds the final DPO checkpoint; the launch records expected a different hash and never checked it. This is a gating failure, not evidence about critic rewards.

Reasoning Challenge. Both probes start from base Qwen3.5-9B with a LoRA adapter initialized to change nothing, because the DPO adapter was not accessible, and train on 239 of the 390 training questions. GROVE-P rewards embedding similarity to the teacher argument, minus a repetition penalty. GROVE-R uses the minimum of two teacher-anchored 4B critics whose validation accuracies are below the 75% gate (Table 4); they are reused under a recorded override. Neither reward checks the answer. Neither probe reaches the 95% completion gate, so both are reported as failed-gate diagnostics.

A.4 Training procedure

Algorithm 1 gives the shared loop. SFT and DPO baselines are trained first. Saved checkpoints then pass or fail their environment's completion, critic, and outcome gates, and a failed checkpoint is reported only with a †. For evaluation, code and reasoning arguments are generated once and cached, and a new evidence debate is generated for every pairing.

Algorithm 1 GROVE policy training (shared loop; environment-specific rewards, eligibility rules, and gates in Appendix A)

Require: training contexts $\mathcal{D}$, initializer π_0, frozen reference π_{ref}, group size G, variant $\in \{\mathrm{P}, \mathrm{R}\}$

1: $\pi_\theta \leftarrow \pi_0$; for GROVE-R, freeze fitted critics and cache an anchor argument $a^\star(x)$ for each x
2: **for** each update **do**
3: **for** each context x in the batch **do**
4: sample $a_{1:G} \sim \pi_\theta(\cdot \mid x)$; score r_k by Table 2 (P) or Eq. 1 (R)
5: $A_k \leftarrow (r_k - \bar{r})/\hat{\sigma}_r$ ▷ denominator 1 if $\hat{\sigma}_r = 0$; skip groups without eligible a_k
6: update π_θ with the group-relative loss (Eq. 2 in code); keep critics and π_{ref} frozen
7: apply validation and execution gates; freeze accepted checkpoints and mark failed runs with †

The method diagram takes stylistic inspiration from the pastel learning-loop layout of Agent0, Figure 2 (Xia et al., 2025). All figures are original constructions from our protocols and frozen data.

B Experimental Details

B.1 Data, exposure, and judging

Code Review. The sampler keeps medium and hard Python problems from DebugBench. Each of the first 50 canonical bug instances, which we call base instances, contributes its correct and its buggy program, giving 100 programs from 43 distinct prompts. The records have no runnable tests, so correctness is dataset-labeled. The fixed split (seed 0) assigns the 50 base instances, not the 43 prompts, to 72/14/14 training, development, and test programs. Two test instances share a prompt with a training or development instance. The critics, source-filtered DPO, and GROVE-R train without the test programs, but the correct test program of instance 15 is byte-identical to the correct training program of instance 49. Winner SFT, which initializes every later stage, trains on 87 programs, including 13 of the 14 test programs, and validates on the other 13. Standard DPO learns from arguments on all 100 programs, and GROVE-P's split was not recorded. GROVE-P's field rate exceeds DPO's by 1.6 pp $[-11.9, 15.0]$ on the 14 development and test base instances and by 3.8 pp $[-2.9, 10.4]$ on the 36 training base instances, but both policies learned from some of these programs. Training labels come from Qwen3.7-Max (Alibaba Cloud, 2026) and DeepSeek-V4-Pro (DeepSeek-AI, 2026), and the arena judge, GPT-5.6-Terra, is held out from training labels. Besides the arena, GPT-5.6-Terra scored the 40-case pilot that stopped GROVE-R, the 56-judgment development gate of the 3-of-4 DPO control, and 600 calibration judgments of Qwen3.5-2B, 9B, and 397B rebuttals. We find that on these calibration judgments it picks the same winner as DeepSeek-V4-Pro in 467 of the 594 cases (78.6%, Cohen's $\kappa = 0.57$) that both judges decide, and both judges rank the three models in the same order. Each rebuttal is generated once and judged against every opponent in both seats, giving $\binom{19}{2} \times 100 \times 2 = 34{,}200$ judgments.

Evidence Debate. The 200 FEVER claims (151 SUPPORTS, 49 REFUTES) are claims on which strong closed-book probes failed or were uncertain. The main arena gives all 16 policies the same 20 test claims (2,400 debates), and the diagnostic arena gives 15 policies all 200 claims. The 20 test claims carry 11 SUPPORTS and 9 REFUTES labels. Winner SFT imitates the winning sides of

4,200 base-versus-base debates, whose verdicts are recorded as coming from GPT-5.6-Sol, the arena judge. It trains on the debates of 180 claims and validates on those of the 20 test claims. Winner SFT, conversation SFT, DPO, and GROVE do not train on the 20 test claims, but the conversation SFT that initializes DPO chose its source models before these claims were excluded. The main arena therefore evaluates on claims used for validation and model selection, and the diagnostic arena adds the other 180 claims, on which the trained policies were trained or validated. Trained-policy comparisons are identity-blind GPT-5.6-Sol judgments. The 300 comparisons involving 397B use a separate GPT-5.6-Sol API route (reasoning disabled, temperature zero, 16-token completions, the same anonymous prompt), so they and the other 2,100 comparisons are judged in different settings. The diagnostic arena's base-versus-base block also has unresolved judge metadata.

Reasoning Challenge. Seed 42 selects disjoint train, validation, and test partitions of 390/65/130 SuperGPQA questions from 13 disciplines before any generation. Every challenger sees the same question and frozen 122B solution; the solution is correct on 76 of the 130 test questions. The 14 policies meet in all 91 pairs under three Qwen 27B judges (35,489 resolved judgments). The roster is a retrospective subset, and no selection rule was fixed in advance. It excludes direct-teacher SFT, SFT on 397B arguments that beat 9B, 27B, and 122B under at least two of the three judges, DPO after direct-teacher SFT, and the original GROVE-R. We find that in the same judgments these four policies win 47.4–51.8% of their comparisons against base 9B, averaged over the three judges, so none exceeds base 9B by more than 1.8 pp.

B.2 Generation and aggregation

In code, Kimi-K2.7-Code writes the fixed review. Frozen authors generate at temperature 0.7 with a 1,200-token cap, adapters use 1,024 tokens with top-p 0.95 (code GROVE-R keeps the 640-token cap of its training), and the judge, GPT-5.6-Terra (OpenAI, 2026b), runs at medium reasoning effort in batches of 11 or 12 cases with no tool access, so it cannot run the code. Evidence arguments use temperature 0.7 and 400 tokens, and the judge must name exactly one winner. Reasoning arguments use temperature zero and 4,096 tokens, and each judge returns a short rationale and an A/B choice. Of 35,490 scheduled reasoning judgments, one is unresolved and omitted, and one is repaired from a rationale that names its winner. Unanimity keeps 9,226 pair-question outcomes and excludes split decisions rather than scoring them as draws. Older archived arenas, such as a DeepSeek-judged code arena and earlier reasoning variants, are not combined with these arenas.

C Extended Results

Policy	Code	Evidence	Reasoning
Qwen3.5-2B	1135.3	773.2	1201.9
Qwen3.5-4B	1372.3	800.0	1569.1
Qwen3.5-9B	1454.1	821.5	1577.9
Qwen3.5-27B	—	873.0	1679.6
Qwen3.6-27B	1562.6	—	—
Qwen3.5-122B	1487.5	856.7	1655.7
Qwen3.5-397B	1639.7	1035.4	1815.9
Gemma-4-E2B	1444.1	792.8	1012.2
Gemma-4-E4B	1416.0	833.3	1184.7
Gemma-4-31B	1576.9	—	—
SFT	1579.7	863.7	1559.6
DPO	1577.4	887.1	1543.0
GROVE-P	1601.3	1010.5	1578.4†
GROVE-R†	1520.9	956.5	1560.5

Table 5: **Elo by environment.** Trained rows use 9B, 4B, and 9B backbones; —: not evaluated. †: failed an objective or gate.

Table 5 places every frozen model and the representative trained policies on each arena's Elo scale, and Figure 1 plots the same ratings against total parameters. 397B has the highest rating in every environment, separated from every other frozen model in evidence and reasoning by 95% intervals.

Size alone does not order the field. In code, ratings rise from Qwen3.5-2B to 4B to 9B, yet the sparse 122B-A10B model (10B active parameters) falls below dense Qwen3.6-27B, and Gemma-4-E2B nearly matches Qwen3.5-9B (1444.1 vs. 1454.1). In reasoning, under Qwen judges, the same E2B model falls below Qwen3.5-2B. The representative trained 4B and 9B policies lie above the frozen-model frontier in code and evidence, on items used during training or model selection, whereas in reasoning they stay at or below base 9B.

Figures 5–8 report every policy with 95% intervals (200-claim field rates are point estimates), and hollow markers denote runs that failed an objective or gate.

C.1 Code Review

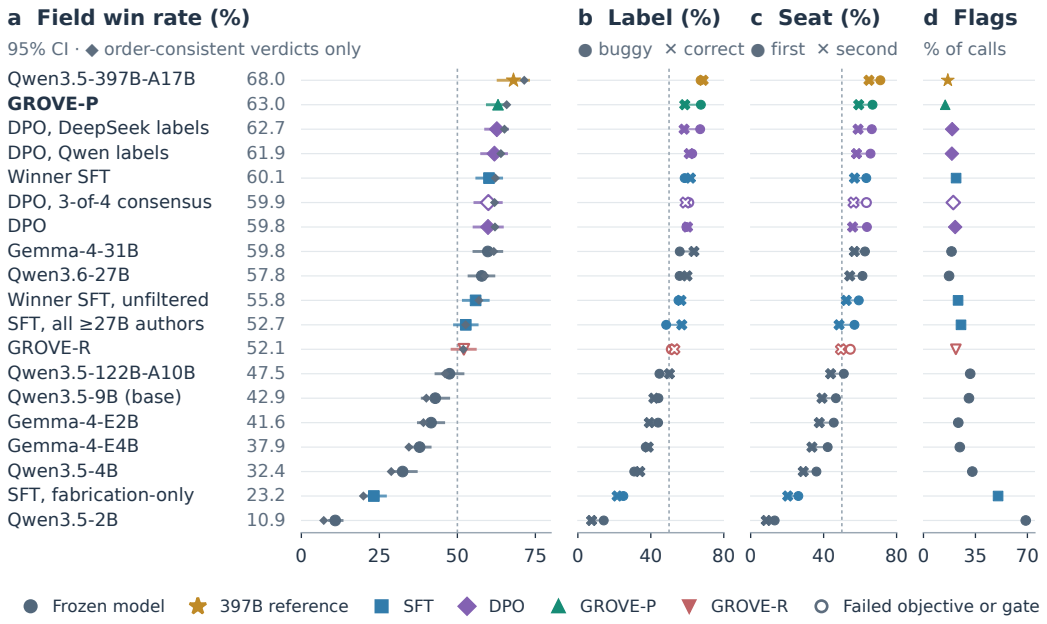

Figure 5: **All 19 code policies.** (a) Field win rate with 95% intervals; diamonds recompute it from order-consistent verdicts. (b) Score on dataset-buggy (circles) and dataset-correct (crosses) programs. (c) Score in the first (circles) and second (crosses) seat. (d) Share of judgments that flag possible fabrication. Hollow markers denote runs that failed an objective or gate.

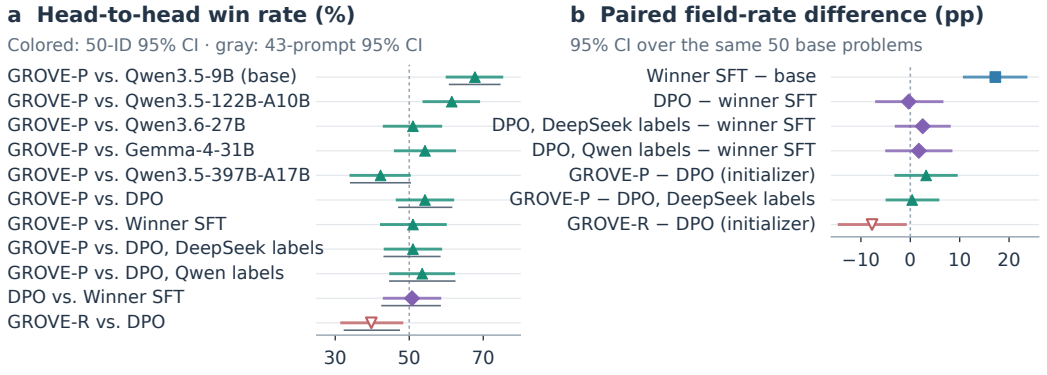

Figure 6: **What each code training stage adds.** (a) Head-to-head win rates with 95% intervals over 50 base instances (colored) and 43 distinct prompts (gray). (b) Paired differences in field win rate on the same bootstrap draws.

Winner SFT supplies most of the gain over base, standard DPO adds nothing, and source-filtered DPO (Qwen or DeepSeek labels) and GROVE-P each add a few pp. Among the paired stage differences, only the SFT gain and GROVE-R's loss exclude zero, as Figure 6b shows. We find that GROVE-R's rebuttals average 157 words against 387 for its initializer, standard DPO, so its loss may partly reflect the $\leq$180-word eligibility gate of its training reward. Every policy scores higher in the first seat. Because each pair is judged in both orders, the seat effect cancels: keeping only order-consistent verdicts leaves the ranking nearly unchanged (Kendall $\tau = 0.988$), with GROVE-P second behind 397B. GROVE-P's larger gain on buggy than on correct programs also appears for source-filtered DPO (DeepSeek labels), which never sees the dataset labels (Figure 5b), so it is not specific to the label-aware reward.

C.2 Evidence Debate

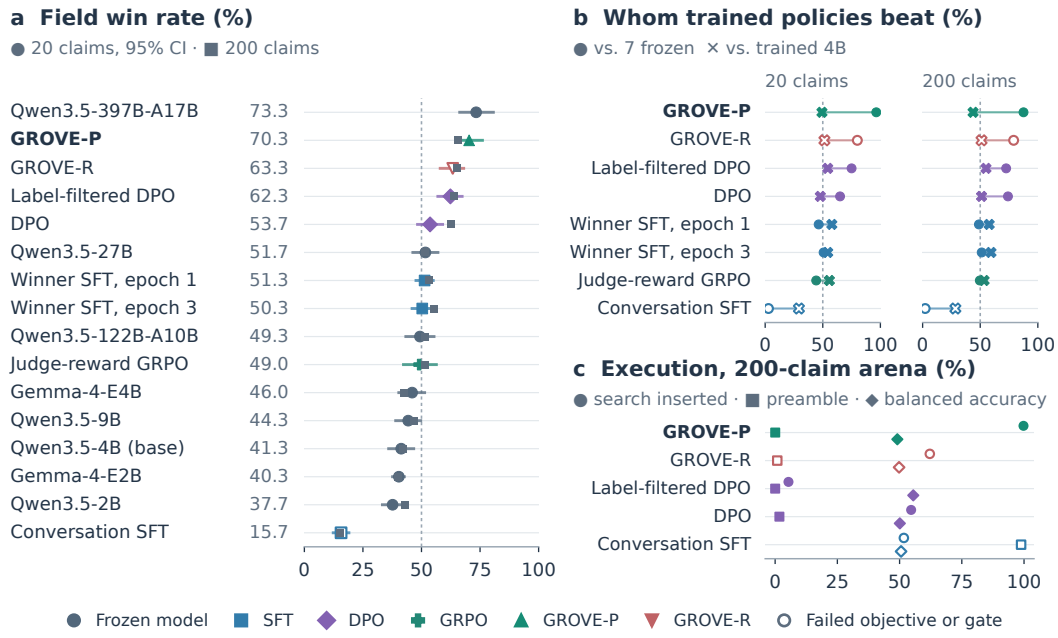

Figure 7: **Evidence Debate: GROVE-P's rank comes from frozen opponents.** (a) Field win rate on 20 claims with 95% intervals, and on 200 claims (squares; 397B not evaluated). (b) In each arena, win rate of each trained 4B policy against the seven frozen models other than 397B (circles) and against the other trained 4B policies (crosses). (c) Engine-inserted required first search (circles), a visible "Thinking Process" preamble (squares), and agreement of the judged winning stance with FEVER labels (balanced accuracy, diamonds), in diagnostic-arena debates against the seven frozen models; here shape marks the metric and color the policy. Hollow markers denote runs that failed an objective or gate.

Both arenas place the four DPO-lineage policies at the top of the trained field, and the 20-claim intervals are too wide to order adjacent policies. GROVE-P's rank comes from frozen opponents, as Figure 7b shows. On 20 claims, it beats every frozen model except 397B, which it does not beat (35% [15, 55]). Among the trained 4B policies on 20 claims, it beats only conversation SFT (75% [55, 90]) and loses only to winner SFT epoch 1 (25% [5, 45]). In the 200-claim arena, it wins 36.0–37.5% against label-filtered DPO, both winner SFT checkpoints, and judge-reward GRPO, with Wilson upper bounds of at most 44.4% (Appendix A.1), so it loses to them, and it ties DPO (44.0% [37.3, 50.9]) and GROVE-R (47.5% [40.7, 54.4]). GROVE-R received zero usable reward, yet its weights move 0.4% from its DPO initializer (Appendix A.3), and on 20 claims it scores 9.7 pp above DPO (63.3% against 53.7%). In the diagnostic arena, the engine inserts the required first search for GROVE-R in 62.1% of debates, against 54.6% for DPO. Evidence gaps below about 10 pp therefore cannot be credited to a reward. GROVE-P's field rate exceeds DPO's by 16.7 pp [8.0, 25.3] on 20 claims, but this gain is not separable from evaluation variation. Head-to-head, GROVE-P wins 65%

[45, 85] against DPO on 20 claims. On 200 claims, GROVE-P's field-rate lead shrinks to 3.1 pp over DPO and 0.6 pp over GROVE-R. The engine inserted the required first search for GROVE-P in 1,397 of its 1,400 diagnostic debates against the seven frozen models (99.8%), although its training never optimized the search action. For the five trained policies with execution records, the judged winning stance agrees with FEVER labels at 49.1–55.5% balanced accuracy. Base versus base reaches about 71%, a value reconstructed from per-label counts that are not part of the frozen data. If GROVE-P argues each side equally often within each label, its 87.6% win rate against the seven frozen models bounds this accuracy between about 37.6% and 62.4%. Its 49.1% means that it wins about as often from the false side as from the true side. These diagnostic counts are transcribed from run summaries, not regenerated from per-debate records.

C.3 Reasoning Challenge

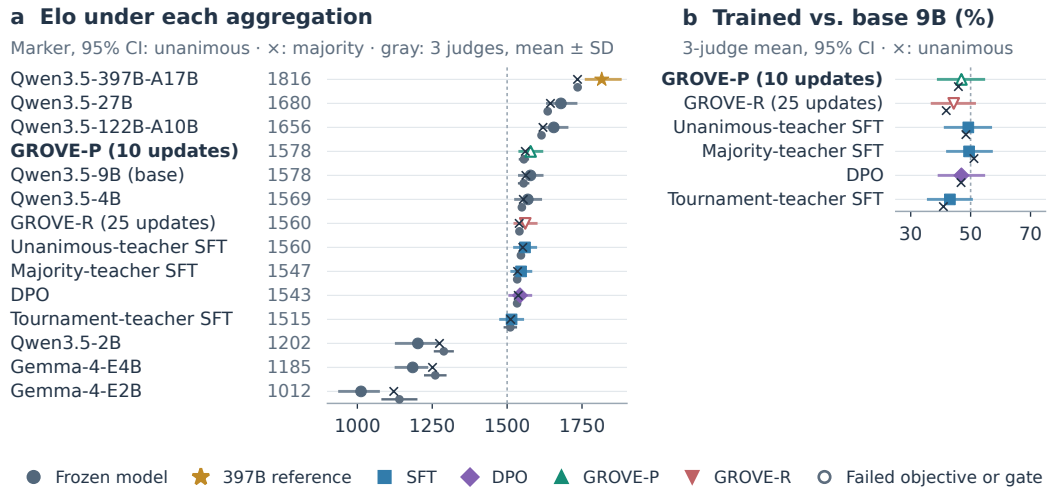

Figure 8: **Reasoning Challenge: no trained policy improves on base 9B.** (a) Elo from unanimous (3/3) decisions with 95% intervals, from majority decisions (×), and from each judge separately (gray, mean ± SD). (b) Each trained 9B policy against base 9B: three-judge mean with a 95% interval, and the unanimous rate (×).

The frozen 397B model ranks first under every judge and aggregation, followed by Qwen3.5-27B and 122B. Qwen3.5-27B is both a judge and a competitor, yet it also ranks second under the two other judges. The three judges agree closely on the frozen models and the 9B policies and diverge only for the weakest models. Because they share one model family, this agreement is a sensitivity check, not independent validation. Against base 9B, every trained policy's three-judge mean is below 50% and every interval spans parity. GROVE-P and base 9B are indistinguishable under every aggregation.

C.4 Head-to-head matrices

Figure 9 names every row and column of the matrices in Figure 4d–f. With 20 claims per pair, the evidence cells are the noisiest.

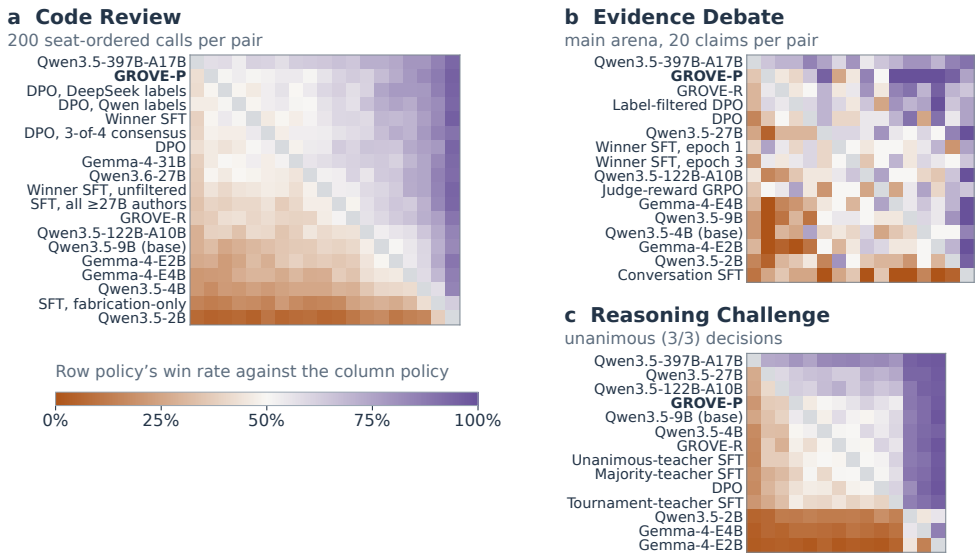

Figure 9: **Every head-to-head result, by name.** Each cell is the row policy's win rate against the column policy (code: ties count half; reasoning: unanimous decisions). Rows and columns share one order, highest Elo first. In evidence, the 397B pairs were judged on a separate GPT-5.6-Sol API route.

D Qualitative Cases

Table 6 shows two code comparisons that we chose after the fact from the final judged corpus to illustrate mechanisms. They do not estimate how often these behaviors occur, and no claim of the paper rests on them. Each comparison shares the same program and review, and judge rationales explain preferences without certifying correctness.

Instance	Argument excerpt or summary	Stored outcome
String Compression II Dataset-correct	GROVE-P defends the empty-run initialization: "This is correct because no run exists yet to incur a cost."	GROVE-P preferred to the fabrication-only SFT control; Terra judge.
Next Greater Element IV Dataset-buggy	GROVE-P concedes: "ss is never populated correctly." The candidate code never transfers an index to its second stack.	GROVE-P preferred to base 9B; Terra judge.

Table 6: **Recorded argument-level comparisons.** Excerpts retain stored wording. Outcomes reflect the selected presentation order. Code labels are dataset-supplied.

Defend a correct invariant. In a dataset-correct String Compression II program, the reviewer questions how the dynamic program charges a run at initialization. GROVE-P explains that closing an empty run costs nothing and that deletion preserves the active run ("This is correct because no run exists yet to incur a cost"). The fabrication-only SFT control confuses the previous and new runs' costs, and the judge prefers GROVE-P.

Concede a real defect. In a dataset-buggy Next Greater Element IV program, no statement ever appends to the second stack, so the loop that assigns answers is unreachable. GROVE-P concedes that "ss is never populated correctly". Base 9B traces a push that does not exist, and the judge prefers the concession.