PRAGREST: Self-Reinforcing Counterfactual Reasoning for Pragmatic Language Understanding

Jihyung Park* **Minchao Huang*** **Leqi Liu** **Elias Stengel-Eskin**
The University of Texas at Austin

Abstract

Natural language understanding often depends on meanings that are implied rather than explicitly stated, requiring pragmatic reasoning. Despite strong performance on math and logical reasoning, large language models (LLMs) still trail humans on pragmatic inference, often defaulting to literal interpretations. To improve LLM pragmatic reasoning, we introduce PRAGREST, a self-supervised framework that constructs pragmatic QA data, generates counterfactual reasoning traces, and trains models to internalize them through supervised fine-tuning and reinforcement learning, without human-labeled training data or distillation from a stronger teacher. Across four pragmatic benchmarks (PRAGMEGA, LUDWIG, METOQA, and ALTPRAG) and multiple model families, including Qwen3, Gemma, and GPT-OSS, PRAGREST consistently improves over backbone models, task-specific pragmatic tuning baselines, and non-counterfactual variants of the same pipeline. On accuracy-based benchmarks, PRAGREST improves over the instruct backbone by 5.37% and 5.50% (absolute) for Qwen3-8B and Qwen3-14B, respectively. Our error analysis and ablations underscore the importance of counterfactual reasoning: PRAGREST primarily reduces errors caused by failures to contrast observed utterances with plausible alternatives, and removing counterfactual reasoning substantially reduces performance. Moreover, our training preserves out-of-domain performance on general-knowledge and mathematical reasoning benchmarks.[1]

1 Introduction

Robust language understanding requires recovering the intentions, assumptions, and implicit meanings that speakers leave unsaid (Grice, 1975), i.e., making pragmatic inference beyond literal meaning and reasoning about shared context, speaker goals, and

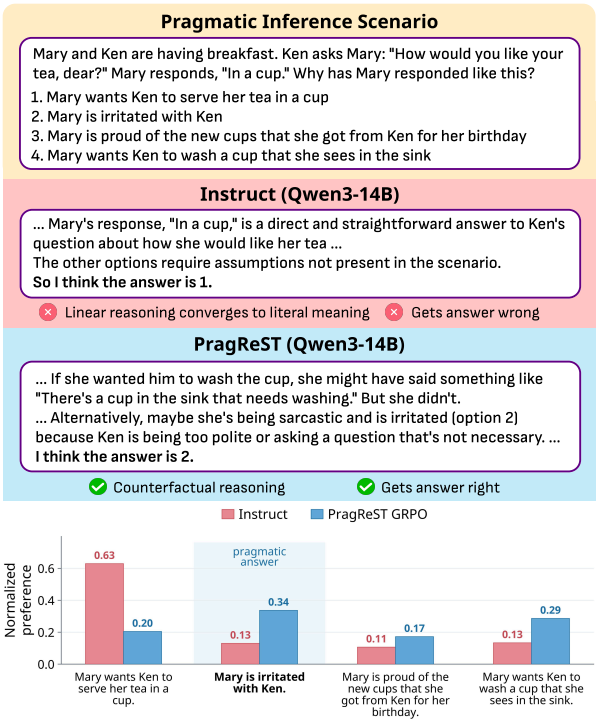


Figure 1: Example of counterfactual pragmatic reasoning in PRAGREST. Under ordinary contextual assumptions, it recovers the intended meaning by reasoning about alternative utterances. Compared with the instruct model, PRAGREST shifts preference from the literal interpretation to the intended pragmatic interpretation.

background information such as implicatures and presuppositions (Levinson, 1983). Humans can make these inferences with ease in daily conversation. Although large language models (LLMs) have improved substantially on reasoning-heavy domains such as mathematics and code (Shao et al., 2024; Trung et al., 2024), recent work suggests that they still struggle with pragmatic inference tasks (Ruis et al., 2023; Hu et al., 2023; Ma et al., 2025; Sravanthi et al., 2024). These findings point to a persistent gap between LLMs and humans in pragmatic reasoning, suggesting that surface fluency and broad semantic competence do not, by themselves, constitute robust communicative un-

*Equal contribution.
[1]Code and models are available here.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 44910–44932
October 24-29, 2026 ©2026 Association for Computational Linguistics

derstanding (Fried et al., 2023).

Cognitive science has long characterized pragmatic interpretation not as retrieval of stored meanings but as a *reasoning* process. Pragmatic frameworks such as the Rational Speech Acts (RSA) (Frank and Goodman, 2012; Goodman and Frank, 2016) and Iterated Best Response (IBR) (Franke, 2009) cast pragmatic interpretation as recursive reasoning between speakers and listeners over communicative alternatives. At the center of these accounts is a pattern we refer to as *counterfactual pragmatic reasoning*: a listener interprets an utterance by comparing what the speaker actually said with what they would likely have said under alternative intentions. Had the speaker intended X, they would likely have said Y; since they said Z instead, X becomes less plausible and an alternative meaning W more plausible. This view is consistent with evidence linking counterfactual and pragmatic reasoning at the level of brain responses (Kulakova and Nieuwland, 2016).

However, translating this account into a scalable training procedure for LLMs is nontrivial for two reasons. First, unlike math and code, where Reinforcement Learning with Verifiable Rewards (RLVR) can exploit deterministic verification (Shao et al., 2024; Trung et al., 2024), pragmatic correctness often depends on contextual assumptions, speaker goals, and social expectations (Fried et al., 2023; Anuranjana et al., 2024). Second, scalable pragmatic supervision is scarce: human annotation is costly, while distillation is limited by the pragmatic competence of the teacher model (Ma et al., 2025; Sravanthi et al., 2024).

To address these challenges, we introduce Pragmatic Reasoning via Self-Training (PRAGREST), a framework for learning counterfactual pragmatic reasoning from self-generated data. PRAGREST is self-reinforcing in the sense that it does not rely on human-labeled pragmatic training data or a stronger external teacher model to distill pragmatic knowledge into the policy. Instead, the same model is used throughout the pipeline. As illustrated in Figure 2, PRAGREST proceeds in two stages. First, the model constructs a pragmatics training set and the same model audits these generated instances to remove low-quality examples. Second, PRAGREST turns these filtered problems into training data for teaching models counterfactual pragmatic reasoning, following two training paradigms: supervised fine-tuning (SFT) and reinforcement learning (RL). For SFT, we first generate privileged answer traces via a prompt that explicitly encourages the model to compare the observed utterance with plausible communicative alternatives, and then train the model on these traces with the prompt removed. This teaches the model to internalize counterfactual reasoning. In the RL stage, we further tune the model via GRPO (Shao et al., 2024), training the model on filtered problems using a self-judged correctness reward.

We evaluate PRAGREST on Qwen3-8B and Qwen3-14B (Qwen Team, 2025) and further validate its effectiveness on Gemma-4-E4B (Gemma Team et al., 2026) and GPT-OSS-20B (OpenAI et al., 2025). We evaluate across four pragmatic benchmarks: PRAGMEGA (fine-grained pragmatics QA; Hu et al., 2023), LUDWIG (implicature interpretation; Ruis et al., 2023), METOQA (metonymic reference resolution; Sravanthi et al., 2024), and ALTPRAG (open-ended pragmatic recovery; Yu et al., 2026).

To isolate the effect of counterfactual reasoning, we compare PRAGREST with a matched non-counterfactual variant that uses the same pipeline but replaces the counterfactual reasoning instruction with a generic pragmatic reasoning instruction. The non-counterfactual variant also improves over the instruct backbone, indicating that self-reinforcement on pragmatic data is beneficial on its own. Counterfactual supervision provides further gains: PRAGREST improves over the instruct backbone by an average of 5.37 and 5.50 percentage points for Qwen3-8B and Qwen3-14B, respectively, across the three accuracy-based benchmarks (PRAGMEGA, LUDWIG, METOQA).[2] Notably, Qwen3-14B comes within one percentage point of the available human-performance estimates on all three benchmarks (Table 2), and the overall improvement pattern also holds on Gemma-4-E4B, GPT-OSS-20B, and across three independent training runs (Section 4.1).

The results suggest that pragmatic self-reinforcement contributes to improved pragmatic inference, with counterfactual reasoning providing gains beyond matched non-counterfactual training. Our error analysis further shows that PRAGREST particularly reduces failures involving literal interpretation and missed communicative intent, where successful inference requires contrasting the observed utterance with plausible alterna-

[2]Unless noted otherwise, all accuracy differences are reported in terms of absolute percentages.

tives (Section 5.1). Finally, these improvements do not substantially degrade out-of-domain performance on general knowledge and mathematical reasoning benchmarks, including MMLU-Pro (Wang et al., 2024), MATH-500 (Lightman et al., 2024; Hendrycks et al., 2021), and AIME2025 (OpenCompass, 2025) (Section 5.2).

2 Related Work

Pragmatics in Language Models. Pragmatic interpretation has been studied as inference over speaker intentions, contextual alternatives, and shared background assumptions. A canonical formalization is the Rational Speech Acts (RSA) framework, which models pragmatic interpretation as probabilistic, recursive reasoning between speakers and listeners about why a speaker chose one utterance over another in a given context (Frank and Goodman, 2012; Goodman and Frank, 2016). This perspective has also informed computational work on pragmatically informed interpretation beyond classical reference games (Fried et al., 2018; Vaduguru et al., 2024). Recent benchmarks show that even strong LLMs remain brittle on context-dependent interpretation, including implicature, presupposition, deixis, metonymic reference, and open-ended pragmatic recovery, with substantial and uneven gaps relative to humans across phenomena and settings (Hu et al., 2023; Ruis et al., 2023; Sravanthi et al., 2024; Yu et al., 2026; Ma et al., 2025; Fried et al., 2023). Recent training approaches improve pragmatic reasoning through task-specific preference tuning or teacher-generated thought supervision (Wu et al., 2024; Sravanthi et al., 2025). These approaches demonstrate that pragmatic reasoning can benefit from specialized supervision, but they rely on external pragmatic data, preference annotations, or teacher-generated rationales. In contrast, PRAGREST self-generates training data and uses counterfactual reasoning as the organizing principle for generating, filtering, and reinforcing reasoning traces within a self-contained training loop.

Self-Improvement in Language Models. Prior work on LLM self-improvement trains on model-generated data, rationales, or rewards, but has focused largely on domains with deterministic tasks and verifiable rewards. Early rationale-bootstrapping methods train models on self-generated reasoning that leads to correct answers, while later work scales this paradigm through expectation-maximization style self-training, reinforced fine-tuning, consistency-based rationale filtering, unsupervised self-training, and autonomous curriculum generation (Zelikman et al., 2022, 2024; Singh et al., 2024; Trung et al., 2024; Lee et al., 2025; Xu et al., 2025; Sun et al., 2025; Zhao et al., 2025). Synthetic-data methods such as Self-Instruct and Evol-Instruct further show that LLMs can expand their own training distributions rather than merely relabel fixed datasets (Wang et al., 2023; Xu et al., 2024). These methods have primarily targeted math, code, or instruction-following settings in which correctness can be checked by ground-truth answers, executors, or verifier models; PRAGREST extends this paradigm to open-ended pragmatics, where both the task distribution and the correctness signal must be constructed: the model generates pragmatic QA instances, filters them with a self-judge, and uses a constrained binary correctness judge rather than a general-purpose quality or preference evaluator. This connects PRAGREST to LLM-based evaluators and self-rewarding systems (Liu et al., 2023; Kim et al., 2024; Yuan et al., 2024), while avoiding a fully general-purpose judge formulation, which can be brittle across domains (Zheng et al., 2023; Huang et al., 2025; Raina et al., 2024). PRAGREST is also related to self-distillation with privileged information and context distillation, where a teacher policy is conditioned on information unavailable to the student at inference time (Nguyen et al., 2026; Penaloza et al., 2026; Zhao et al., 2026; Ye et al., 2026). In PRAGREST, the privileged signal is a counterfactual reasoning script used only during SFT data generation; the model is then trained on the original problem without this script, with a goal of internalizing the reasoning procedure.

3 Methodology

As illustrated in Fig. 2, PRAGREST consists of two stages. First, the model generates pragmatics-focused QA problems from domain seeds and pragmatic phenomenon descriptions, then filters the generations itself. Second, the model learns from this data through bootstrapped counterfactual training: in the SFT stage, we distill reasoning traces generated under a privileged counterfactual script, and in the GRPO phase we reinforce pragmatically correct answers. All supervision signals are produced without human-labeled pragmatic data or a stronger external model.

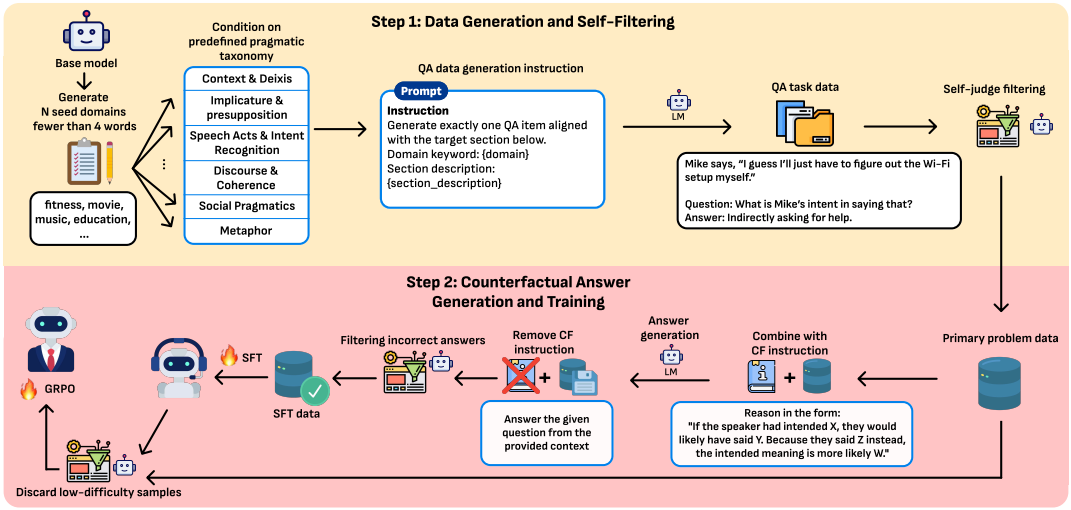


Figure 2: A high-level overview of PRAGREST. The process starts with the self-generation of domain seeds. The QA task prompt is constructed by combining a general instruction, a domain seed, and a pragmatic taxonomy description. The generated data are filtered by a self-judge. For SFT, gold answers are pre-generated with a counterfactual (CF) instruction, which is removed from the final SFT input. We also apply a second filtering step to remove examples for which the model returns an incorrect answer even with the pragmatic instruction.

3.1 Data Generation

Prompt Construction. Our pipeline begins by generating a pool of short domain descriptors (e.g., Healthy Meal Prep, Modern Travel Planner). We then sample a pragmatic taxonomy category from the list introduced by Ma et al. (2025): *Context and Deixis*, *Implicature and Presupposition*, *Speech Acts and Intent Recognition*, *Discourse and Coherence*, *Social Pragmatics*, and *Metaphor*, as illustrated in Step 1 of Fig. 2. For each domain–category pair, the model is prompted to generate an open-ended QA item consisting of a pragmatic situation, a question grounded in that situation, and a target answer. This differs from evaluation on fixed benchmark items. Instead of recovering an implicit meaning from an externally provided context, the model constructs the scenario, question, and intended answer together under an explicit pragmatic category description. The generated item can therefore be treated as a candidate locally grounded QA instance, closer to answer-aware question generation and grounded synthetic QA generation than to ordinary benchmark inference (Zhang et al., 2022; Radevski et al., 2025). We expand on data generation details, including few-shot examples and prompts, in Appendix A.

Self-Filtering. Using the prompts constructed in Step 1, the model generates candidate QA and dialogue instances. Because these candidates may still be noisy, we retain them only after filtering out examples that are malformed, ambiguous, answerable without pragmatic reasoning, or not pragmatically licensed. This proposal-and-filtering strategy follows prior self-generation and bootstrapping work, which uses model-generated candidates for training only after automatic filtering or correctness checks (Wang et al., 2023; Zelikman et al., 2022). For each generated instance, we use the corresponding instruct backbone as a constrained binary judge. The judge is prompted to output either *yes* or *no*, indicating whether the instance meets a set of manually defined quality criteria (provided in Appendix A.3). To obtain a continuous judge signal, we derive a first-token confidence margin $m(q) = p(\texttt{yes} \mid q) - p(\texttt{no} \mid q)$, where q is the generated QA item. We interpret lower scores as lower-confidence generations and discard the bottom 50% of the data, ranked by margin. To give a sense of typical scale, our Qwen3-14B run starts from 1,000 domain seeds and yields 6,000 parseable QA items, of which 3,000 are retained after self-filtering as the primary problem data.

3.2 Training

We train the model in two stages. The first stage uses supervised fine-tuning (SFT) to internalize counterfactual pragmatic reasoning from generated demonstrations, and the second stage uses GRPO to reinforce pragmatically correct outcomes. We describe the main design choices below and provide full training details in Appendix B.

SFT with Counterfactual Bootstrapping. The goal of the SFT stage is to teach the model a reusable counterfactual reasoning procedure. For each filtered training instance, we construct an augmented target-generation prompt by prepending the counterfactual reasoning script and the corresponding pragmatic section description to the original problem. Under this augmented prompt, the model generates candidate responses consisting of a reasoning trace and a final answer; the script encourages the model to interpret an utterance as a communicative choice, compare it with plausible alternatives, and infer the speaker's intended meaning from that contrast (Goodman and Frank, 2016; Fried et al., 2018; Vaduguru et al., 2024; Tsvilodub et al., 2025). We then apply the answer-correctness judge to retain only candidate responses whose final answer is judged as being pragmatically correct with respect to the context, question, and reference answer. At the margin threshold of $m(x, a) > 0.8$, the judge agrees with human consensus on 80.0% of 100 manually reviewed examples (F1 $= 0.796$; Appendix B.3). For the Qwen3-14B run, 2,816 counterfactual responses pass this filter, yielding 2,759 training examples and 57 held-out synthetic validation examples. After filtering, each accepted response is paired with the unaugmented original problem. Both the counterfactual script and the pragmatic section description are removed from the student input. Thus, the script acts as privileged scaffolding for constructing SFT targets, not as an inference-time prompt, and training distills the resulting counterfactual reasoning behavior into the model so that it can recover the same reasoning pattern from the original input alone. Details on SFT data construction and hyperparameters are given in Appendix B.4.

Reinforcement Learning with GRPO. After SFT, the model has learned from counterfactual reasoning traces, but the SFT stage uses correctness only as a criterion for selecting demonstrations rather than as an objective optimized during training. We therefore initialize from the SFT checkpoint and apply GRPO (Shao et al., 2024) on the filtered primary problem data. Before GRPO training, we apply an offline difficulty filter, following DAPO (Yu et al., 2025). For each prompt, we sample G rollouts from the initial SFT model and discard *easy* prompts on which every rollout already passes the correctness judge, since these zero-variance groups provide no learning signal. Note that we keep prompts for which the SFT model gets zero reward, as these may still yield some signal later from GRPO-trained checkpoints. During GRPO training, we keep the filtered prompt set fixed but resample rollouts online as the model is updated. For each remaining prompt, the current policy draws a fresh group of G candidate responses online, which are scored using a composite reward that combines output-format compliance with pragmatic answer correctness. The correctness component reuses the same correctness judge and first-token margin construction defined in Appendix B.2, comparing the extracted final answer against the reference answer. This training reinforces responses that recover the intended pragmatic interpretation while regularizing the policy toward the SFT checkpoint. Full GRPO details and hyperparameters are provided in Appendix B.6.

4 Experiments and Results

Benchmarks and Models. We evaluate our method on four benchmarks selected to directly test pragmatic interpretation: PRAGMEGA, LUDWIG, METOQA, and ALTPRAG. PRAGMEGA (Hu et al., 2023) is a QA benchmark for pragmatic language understanding spanning multiple pragmatic phenomena. LUDWIG (Ruis et al., 2023) evaluates implicature interpretation as a binary decision over whether a listener's response should be interpreted as yes or no. METOQA (Sravanthi et al., 2024) evaluates metonymic reference resolution, where the model must infer the intended referent behind a contextually associated expression. ALTPRAG (Yu et al., 2026) evaluates open-ended pragmatic recovery, where the model must produce an appropriate interpretation of the implied meaning. Representative examples from each benchmark are provided in Table 9. For PRAGMEGA, LUDWIG, and METOQA, we report accuracy; for ALTPRAG, we report the benchmark's reference-based evaluation score and pairwise comparisons using the original GPT-4.1 judge and scoring protocol released by the benchmark authors. Our primary experiments use two instruct backbone models, Qwen3-8B and Qwen3-14B; we also replicate the pipeline for two additional model families, Gemma-4-E4B and GPT-OSS-20B. We apply the same training pipeline to each model and evaluate the resulting models under the same benchmark settings.

Baselines. We compare PRAGREST against four baselines. The first is the Instruct backbone. The

Model	PragMega	Ludwig	MetoQA	AltPrag
	Qwen3-8B			
Instruct	73.37 ± 3.33	80.33 ± 1.62	73.52 ± 2.06	7.24 ± 0.05
Deep-layer-DPO (Wu et al., 2024)	67.46 ± 3.58	80.00 ± 1.63	72.65 ± 2.08	7.24 ± 0.05
IMP-SFT (Sravanthi et al., 2025)	69.82 ± 3.54	82.83 ± 1.54	73.09 ± 2.06	7.19 ± 0.05
Non-counterfactual-SFT	76.92 ± 3.27	81.83 ± 1.57	77.02 ± 1.98	7.44 ± 0.05
Non-counterfactual-GRPO	77.51 ± 3.21	**83.00** ± 1.54	78.12 ± 1.93	7.60 ± 0.05
PragReST-SFT	77.51 ± 3.22	82.17 ± 1.55	78.56 ± 1.90	7.46 ± 0.05
PragReST-GRPO	**80.47** ± 3.04	**83.00** ± 1.53	**79.87** ± 1.88	**7.62** ± 0.04
	Qwen3-14B			
Instruct	81.66 ± 2.99	82.67 ± 1.52	71.77 ± 2.11	7.78 ± 0.05
Deep-layer-DPO (Wu et al., 2024)	77.51 ± 3.22	83.50 ± 1.51	72.87 ± 2.07	7.71 ± 0.05
IMP-SFT (Sravanthi et al., 2025)	81.07 ± 2.95	83.17 ± 1.52	72.21 ± 2.10	7.53 ± 0.05
Non-counterfactual-SFT	83.43 ± 2.88	83.33 ± 1.52	73.52 ± 2.06	7.93 ± 0.05
Non-counterfactual-GRPO	84.62 ± 2.76	84.33 ± 1.49	76.15 ± 1.98	8.07 ± 0.04
PragReST-SFT	85.21 ± 2.71	85.17 ± 1.45	79.65 ± 1.90	8.06 ± 0.04
PragReST-GRPO	**85.80** ± 2.67	**86.50** ± 1.40	**80.31** ± 1.84	**8.14** ± 0.04

Table 1: Performance across benchmarks and models under greedy decoding. PragMega, Ludwig, and MetoQA report accuracy; AltPrag reports its reference-based score. Values are point estimates with bootstrap standard errors over examples. We bold the best point estimate for each model size and benchmark.

second is Deep-layer-DPO, following Wu et al. (2024), a layer-restricted DPO method designed for social and pragmatic inference. For this baseline, we follow the strongest setting reported in Wu et al. (2024) and train on SocialIQA (Sap et al., 2019), a human-annotated social inference dataset. To adapt this baseline to reasoning models, we generate gold-guided reasoning traces by providing the correct answer during trace generation and train on the resulting reasoning-answer outputs. The third is IMP-SFT, following Sravanthi et al. (2025), which distills GPT-4o-mini-generated rationales for pragmatic understanding. For this baseline, we preserve the model's original reasoning behavior by masking the loss on the model-generated reasoning part, then appending the GPT-4o-mini rationale and training on the rationale paired with the correct label. Deep-layer-DPO and IMP-SFT represent two recent training-based approaches to pragmatic improvement that rely on external supervision: human-annotated social inference data and teacher-rationale distillation, respectively. Finally, we include non-counterfactual variants of our own pipeline, which keep the same data generation, filtering, SFT, and GRPO stages as PragReST but replace the counterfactual reasoning instruction with a generic pragmatic reasoning instruction. This helps distinguish the effect of counterfactual reasoning from the effect of self-training and RL post-training, and provides the direct test of our central hypothesis: if counterfactual comparison is the active ingredient rather than self-training itself, then matched training without it should fail to reproduce the gains. Counterfactual and non-counterfactual prompts are in Appendix B.1.

4.1 Results

Table 1 shows that PragReST consistently improves pragmatic reasoning across the four evaluation benchmarks and both model sizes. For Qwen3-8B, PragReST-GRPO achieves the best result in nearly every benchmark, improving over the instruct backbone on PragMega, Ludwig, MetoQA, and AltPrag. Across the three accuracy-based benchmarks, this corresponds to an average gain of 5.37% over the Instruct backbone; on AltPrag, which uses a reference-based score, PragReST-GRPO improves from 7.24 to 7.62. For Qwen3-14B, PragReST-GRPO again gives the best performance across all reported benchmarks, with an average gain of 5.50% across the three accuracy-based benchmarks and an AltPrag improvement from 7.78 to 8.14. These results suggest that PragReST improves multiple forms of pragmatic interpretation, including fine-grained QA, implicature resolution, metonymic reference, and open-ended pragmatic recovery.

Comparison with Human Performance. To contextualize the remaining headroom on the

Benchmark	Human score	PRAGREST	Gap
PRAGMEGA	86.37	85.80	−0.57
LUDWIG	86.20	86.50	+0.30
METOQA	80.00	80.31	+0.31

Table 2: Comparison between Qwen3-14B PRAGREST and human-performance estimates on the three accuracy-based pragmatic benchmarks. The PRAGMEGA human score is computed using the benchmark authors' data. The LUDWIG and METOQA human scores are those reported in the corresponding papers.

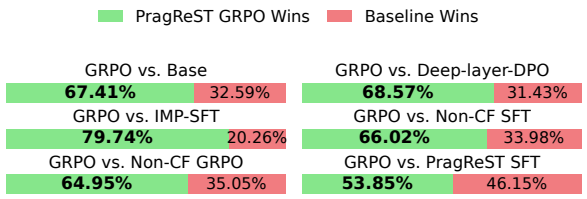


Figure 3: Pairwise preference on ALTPRAG: each bar compares PRAGREST-GRPO against a Qwen3-14B baseline, using GPT-4.1 as a blind pairwise judge over the two models' generated answers.

accuracy-based pragmatic benchmarks, we compare PRAGREST-GRPO with the human-performance estimates reported or computable from the corresponding benchmark resources. We restrict this comparison to PRAGMEGA, LUDWIG, and METOQA, since these benchmarks are evaluated with accuracy. We do not include ALTPRAG in this table because it uses a reference-based open-ended scoring protocol rather than a directly comparable human accuracy score. As shown in Table 2, Qwen3-14B PRAGREST-GRPO reaches performance close to the human estimates available on all three accuracy-based benchmarks. In PRAGMEGA, the human score computed using the code of the benchmark authors over our evaluated subset is 86.37, compared to 85.80 for PRAGREST-GRPO. On LUDWIG, PRAGREST-GRPO reaches 86.50, slightly above the reported human average of 86.2. In METOQA, PRAGREST-GRPO reaches 80.31, close to the reported human score of 80.0. These comparisons suggest that, for the accuracy-based benchmarks, PRAGREST operates in a regime of near-human performance, which may partly explain why absolute gains over strong instruction-tuned backbones are modest on some tasks.

Comparison with External Baselines. PRAGREST also compares favorably against prior task-specific pragmatic tuning methods. For Qwen3-8B, DEEP-LAYER-DPO underperforms the instruct backbone on PRAGMEGA, LUDWIG, and METOQA. Meanwhile IMP-SFT improves over the instruct model on LUDWIG, because it contains an augmented version of LUDWIG in its training data, but its performance does not transfer to other benchmarks. Overall, these comparisons underscore that PRAGREST improves over prior pragmatic supervision methods despite learning from a self-generated signal rather than relying on human-annotated data or a stronger teacher.

Preference Evaluation on ALTPRAG. Because ALTPRAG requires open-ended pragmatic recovery, we complement its reference-based score with a blind GPT-4.1 pairwise comparison over model outputs, following the benchmark's original evaluation setup. As shown in Figure 3, PRAGREST-GRPO is preferred over the instruct backbone in 67.41% of decided comparisons, over the external baselines, over the non-counterfactual variants, and over PRAGREST-SFT in 53.85%. This indicates that the ALTPRAG improvement is not only a scalar-score shift: when full interpretations are compared directly, PRAGREST-GRPO is more often judged to recover the intended pragmatic meaning.

Importance of Counterfactual Supervision. The non-counterfactual variants provide a controlled test of whether the gains come from self-training alone or from the counterfactual structure of the supervision. They use the same generated problem distribution, filtering procedure, SFT stage, and GRPO stage as PRAGREST, but ask the model to reason about the pragmatic meaning of the utterance in context, without providing the explicit counterfactual scaffold used in PRAGREST. Their weaker performance suggests that general pragmatic self-training does not match PRAGREST: the self-improvement loop becomes substantially more effective when the training signal teaches the model to contrast the observed utterance with plausible communicative alternatives. At the same time, PRAGREST-GRPO improves over PRAGREST-SFT on the primary comparison, indicating that outcome-based reinforcement adds gains beyond imitation of counterfactual traces. Taken together, these results suggest that the two stages play complementary roles: SFT gives the model a counterfactual reasoning procedure, while GRPO reinforces when and how to apply that procedure to recover pragmatically correct interpretations.

Model	PragMega	Ludwig	MetoQA	AltPrag
Qwen3-14B				
Instruct	81.66	82.67	71.77	7.78
PragReST-SFT	84.62 ± 0.60	84.17 ± 0.87	**80.09** ± 0.58	7.83 ± 0.20
PragReST-GRPO	**86.00** ± 0.34	**85.06** ± 1.29	79.80 ± 1.08	**8.00** ± 0.12

Table 3: Mean and standard deviation over three independent Qwen3-14B runs under greedy decoding. The deterministic Instruct baseline is evaluated once.

Model	PragMega	Ludwig	MetoQA	AltPrag
Gemma-4-E4B				
Instruct	71.60 ± 3.47	82.33 ± 1.56	75.05 ± 2.02	7.39 ± 0.05
PragReST-SFT	75.15 ± 3.32	82.83 ± 1.54	80.96 ± 1.84	7.68 ± 0.04
PragReST-GRPO	**78.11** ± 3.18	**84.00** ± 1.50	**82.71** ± 1.77	**7.72** ± 0.04
GPT-OSS-20B				
Instruct	68.05 ± 3.59	74.33 ± 1.76	74.84 ± 2.02	7.41 ± 0.05
PragReST-SFT	75.74 ± 3.27	79.33 ± 1.66	**79.87** ± 1.85	7.44 ± 0.05
PragReST-GRPO	**77.51** ± 3.24	**80.67** ± 1.65	79.21 ± 1.90	**7.47** ± 0.05

Table 4: Performance of PragReST on *Gemma-4-E4B* and *GPT-OSS-20B*. Values are point estimates with bootstrap standard errors over examples.

Robustness and Generalization. Table 3 shows that across three independent Qwen3-14B runs, both PragReST-SFT and PragReST-GRPO remain above the instruct backbone on average. Moreover, Table 4 shows that the same pattern holds for Gemma-4-E4B and GPT-OSS-20B, with average PragReST-GRPO gains of 5.28 and 6.72 percentage points, respectively, across the three accuracy-based benchmarks. For GPT-OSS-20B, we use LoRA adapters and low reasoning effort due to its larger model size. These results indicate that the gains are not specific to one training run or the Qwen3 architecture.

5 Discussion and Analysis

Our results in Table 1 and Fig. 3 show that PragReST improves performance across multiple pragmatic reasoning tasks. Matched non-counterfactual self-training also helps, but explicit counterfactual supervision yields the strongest overall results. To better understand these trends, we analyze where PragReST's gains stem from. Additionally, we show that training models for pragmatic reasoning still preserves their broader reasoning and knowledge capabilities.

5.1 Counterfactual Reasoning and Error Reduction

We test whether PragReST's gains concentrate on errors that require comparing observed utterances with plausible alternatives. We analyze incorrect

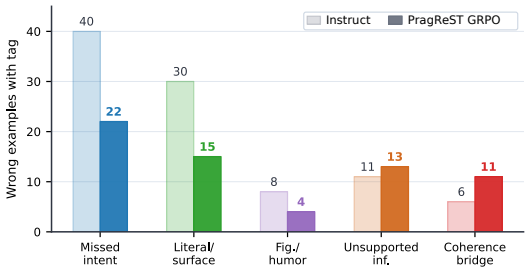


Figure 4: Tagged error counts before and after PragReST training. Bars show the number of incorrect PragMega examples assigned to each error category for the Instruct model and PragReST.

PragMega outputs, induce a recurring-error taxonomy, and relate changes in each error type to the amount of counterfactual reasoning observed. Appendix D provides the full construction and validation procedure.

Inducing an Error Taxonomy. Using incorrect outputs from the evaluated Qwen3 models, GPT-4.1-mini first proposes batch-level error categories and then consolidates them into five recurring types: *literal/surface bias*, *missed communicative intent*, *unsupported or overextended inference*, *coherence-bridge error*, and *figurative or humor mapping error*. The model subsequently assigns all applicable categories to each failure case. Definitions and prompts are provided in Appendix D.1.

Counterfactual Reasoning Score. In addition to error tags, each reasoning trace is automatically scored for the presence of counterfactual pragmatic reasoning. We prompt a GPT-4.1-mini judge to flag whether the trace considers relevant alternative utterances or interpretations, contrasts literal and intended meanings, identifies mismatches between what was said and what would have been said under a literal interpretation, and uses the speaker's communicative choice to infer intent. A higher score means more counterfactual reasoning.

Validating Error-Taxonomy Annotation. We validate the GPT-4.1-mini error-taxonomy labels through a blind study of 40 error samples labeled by three humans. Human-human agreement is 83.8% with an average Micro Cohen's κ of 0.628, while human-GPT agreement is 82.6% with an average κ of 0.614. Appendix D.2 provides the full analysis.

Error Reduction Aligns with Counterfactual Reasoning. As shown in Fig. 4, PragReST reduces the dominant counterfactual-pragmatic fail-

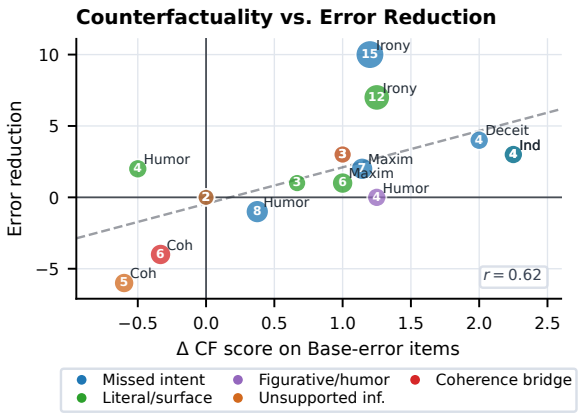


Figure 5: Error reduction by failure mode. The x-axis shows the change in mean counterfactual-reasoning score from the Instruct model to PRAGREST, and the y-axis shows the reduction in tagged errors. The numbers inside indicate how many errors are associated with each phenomenon covered in PRAGMEGA.

ure modes: missed communicative intent drops from 40 to 22, literal/surface bias from 30 to 15, and figurative/humor mapping from 8 to 4. These categories require the model to move beyond literal compatibility and infer why the speaker chose the observed utterance rather than a more direct alternative. The same pattern appears in Fig. 5: error reductions correlate with increases in counterfactual reasoning, suggesting that the gains are tied to more explicit reasoning over communicative alternatives. At the same time, unsupported-inference and coherence-bridge errors do not decrease, suggesting that counterfactual reasoning alone is not sufficient when the main challenge is determining whether an inferred alternative is supported by the discourse context.

5.2 Out-of-Domain Evaluation

A common concern with task-specific post-training is that improvements on the target domain may come at the cost of broader model capability. We therefore evaluate whether PRAGREST preserves out-of-domain performance on general knowledge, mathematical reasoning, and factual truthfulness tasks. We compare PRAGREST with the instruct backbone on MMLU-Pro (Wang et al., 2024), MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024), AIME2025 (OpenCompass, 2025), and TruthfulQA (Lin et al., 2022). As shown in Table 5, performance remains close across both model sizes, with no consistent degradation. Thus, the pragmatic gains do not appear to systematically

Benchmark	Metric	Instruct	PRAGREST
Qwen3-8B			
MMLU-Pro	Acc.	71.63	70.72
MATH-500	Pass@8	97.40	97.00
AIME2025	Pass@8	80.00	80.00
TruthfulQA	MC2	55.28	55.17
Qwen3-14B			
MMLU-Pro	Acc.	72.87	75.10
MATH-500	Pass@8	97.40	97.20
AIME2025	Pass@8	83.33	83.33
TruthfulQA	MC2	55.82	56.01

Table 5: Out-of-domain evaluation for Qwen3-8B and Qwen3-14B models. MMLU-Pro accuracy is computed on a 10% subset sampled from each subject.

trade off with general knowledge, mathematical reasoning, or factual truthfulness.

6 Conclusion

We introduced PRAGREST, a self-reinforcing framework for improving pragmatic reasoning through self-generated counterfactual supervision, without human-labeled pragmatic data or stronger teacher models. Across pragmatic benchmarks, PRAGREST improves over backbone models, prior pragmatic tuning baselines, and non-counterfactual variants, with analyses showing that the gains concentrate in cases requiring comparison between what a speaker said and what they could have said under alternative intentions. These results suggest that reinforcement-based self-improvement can extend beyond formally verifiable domains toward socially grounded language understanding.

Limitations

Although PRAGREST improves pragmatic reasoning, some limitations remain. First, our evaluation inherits a general limitation of pragmatics benchmarks: each item provides only a finite textual context. Additional background information, visual evidence, or unusual speaker assumptions could sometimes license an interpretation different from the annotated target. This under-determination is common in pragmatics, where even canonical examples depend on ordinary contextual assumptions. Our experiments therefore evaluate whether models recover the intended interpretation under the assumptions encoded by existing benchmark items, not whether they resolve all possible readings under arbitrary expanded contexts.

Second, PRAGREST does not uniformly reduce

all types of pragmatic errors. In our error analysis, literal/surface-bias and missed-intent errors decrease substantially, while unsupported-inference and coherence-bridge errors are not consistently reduced. These remaining errors suggest that generating plausible communicative alternatives is not sufficient on its own: the model must also determine whether those alternatives are supported by the specific discourse context. Future work could therefore incorporate stronger evidence-checking mechanisms when constructing or using counterfactual alternatives.

Finally, our evaluation is limited to English-language pragmatic benchmarks. This follows the available benchmark setting and allows controlled comparison with prior work, but pragmatic interpretation is strongly shaped by language, culture, social norms, and conversational conventions (Fried et al., 2023; Ma et al., 2025). As a result, our findings do not establish that the same counterfactual training procedure transfers to multilingual or culturally variable pragmatic settings. Extending PRAGREST to non-English and cross-cultural pragmatics is an important direction for future work.

Acknowledgments

We would like to thank Jessy Li for her helpful feedback, and Ananya Sahu for providing annotations.

References

Kaveri Anuranjana, Srihitha Mallepally, Sriharshitha Mareddy, Amit Shukla, and Radhika Mamidi. 2024. Survey on Computational Approaches to Implicature. In *Proceedings of the 21st International Conference on Natural Language Processing (ICON)*, pages 224–229, AU-KBC Research Centre, Chennai, India. NLP Association of India (NLPAI).

Michael C. Frank and Noah D. Goodman. 2012. Predicting pragmatic reasoning in language games. *Science*, 336(6084):998–998.

Michael Franke. 2009. *Signal to act: Game theory in pragmatics*. University of Amsterdam.

Daniel Fried, Jacob Andreas, and Dan Klein. 2018. Unified Pragmatic Models for Generating and Following Instructions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1951–1963, New Orleans, Louisiana. Association for Computational Linguistics.

Daniel Fried, Nicholas Tomlin, Jennifer Hu, Roma Patel, and Aida Nematzadeh. 2023. Pragmatics in Language Grounding: Phenomena, Tasks, and Modeling Approaches. In *Findings of the Association for Computational Linguistics 2023*, pages 12619–12640, Singapore. Association for Computational Linguistics.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, Mayank Chaturvedi, Aditya Chawla, Victor Cotruta, Alice Coucke, Phil Culliton, Robert Dadashi, Lucas Dixon, Mohamed Elhawaty, Utku Evci, and 304 others. 2026. Gemma 4 Technical Report. *arXiv preprint*. ArXiv:2607.02770 [cs.CL].

Noah D. Goodman and Michael C. Frank. 2016. Pragmatic Language Interpretation as Probabilistic Inference. *Trends in Cognitive Sciences*, 20(11):818–829.

H. P. Grice. 1975. Logic and Conversation. In *Speech Acts*. Brill.

Andre Wang He, Daniel Fried, and Sean Welleck. 2025. Rewarding the unlikely: Lifting GRPO beyond distribution sharpening. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25548–25560, Suzhou, China. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*.

Jennifer Hu, Sammy Floyd, Olessia Jouravlev, Evelina Fedorenko, and Edward Gibson. 2023. A fine-grained comparison of pragmatic language understanding in humans and language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4194–4213, Toronto, Canada. Association for Computational Linguistics.

Hui Huang, Xingyuan Bu, Hongli Zhou, Yingqi Qu, Jing Liu, Muyun Yang, Bing Xu, and Tiejun Zhao. 2025. An Empirical Study of LLM-as-a-Judge for LLM Evaluation: Fine-tuned Judge Model is not a General Substitute for GPT-4. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5880–5895, Vienna, Austria. Association for Computational Linguistics.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*.

Eugenia Kulakova and Mante S Nieuwland. 2016. Pragmatic skills predict online counterfactual comprehension: Evidence from the N400. *Cognitive, Affective, & Behavioral Neuroscience*, 16(5):814–824.

Jaehyeok Lee, Keisuke Sakaguchi, and JinYeong Bak. 2025. Self-Training Meets Consistency: Improving LLMs' Reasoning with Consistency-Driven Rationale Evaluation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10519–10539, Albuquerque, New Mexico. Association for Computational Linguistics.

Stephen C. Levinson. 1983. *Pragmatics*. Cambridge University Press. ISBN: 9780511813313.

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

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using GPT-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Bolei Ma, Yuting Li, Wei Zhou, Ziwei Gong, Yang Janet Liu, Katja Jasinskaja, Annemarie Friedrich, Julia Hirschberg, Frauke Kreuter, and Barbara Plank. 2025. Pragmatics in the Era of Large Language Models: A Survey on Datasets, Evaluation, Opportunities and Challenges. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8679–8696, Vienna, Austria. Association for Computational Linguistics.

Duy Nguyen, Hanqi Xiao, Archiki Prasad, Zaid Khan, Anirban Das, Austin Zhang, Sambit Sahu, Hyunji Lee, Elias Stengel-Eskin, and Mohit Bansal. 2026. AVSD: Adaptive-view self-distillation by balancing consensus and teacher-specific privileged signals. *Preprint*, arXiv:2605.20643.

OpenAI, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, and 107 others. 2025. gpt-oss-120b & gpt-oss-20b Model Card. *arXiv preprint*. ArXiv:2508.10925 [cs.CL].

OpenCompass. 2025. AIME 2025 dataset. https://huggingface.co/datasets/opencompass/AIME2025. Hugging Face dataset.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. 2026. Privileged information distillation for language models. *Preprint*, arXiv:2602.04942.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Gorjan Radevski, Kiril Gashteovski, Shahbaz Syed, Christopher Malon, Sebastien Nicolas, Chia-Chien Hung, Timo Sztyler, Verena Heußer, Wiem Ben Rim, Masafumi Enomoto, Kunihiro Takeoka, Masafumi Oyamada, Goran Glavaš, and Carolin Lawrence. 2025. On Synthesizing Data for Context Attribution in Question Answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16929–16950, Vienna, Austria. Association for Computational Linguistics.

Vyas Raina, Adian Liusie, and Mark Gales. 2024. Is LLM-as-a-Judge Robust? Investigating Universal Adversarial Attacks on Zero-shot LLM Assessment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7499–7517, Miami, Florida, USA. Association for Computational Linguistics.

Laura Ruis, Akbir Khan, Stella Biderman, Sara Hooker, Tim Rocktäschel, and Edward Grefenstette. 2023. The Goldilocks of Pragmatic Understanding: Fine-Tuning Strategy Matters for Implicature Resolution by LLMs. *Advances in Neural Information Processing Systems*, 36:20827–20905.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense Reasoning about Social Interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4463–4473, Hong Kong, China. Association for Computational Linguistics.

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

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron T. Parisi, Abhishek Kumar, Alexander A. Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Fathy Elsayed, Hanie Sedghi, and 22 others. 2024. Beyond human data:

Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*.

Settaluri Sravanthi, Meet Doshi, Pavan Tankala, Rudra Murthy, Raj Dabre, and Pushpak Bhattacharyya. 2024. PUB: A Pragmatics Understanding Benchmark for Assessing LLMs' Pragmatics Capabilities. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12075–12097, Bangkok, Thailand. Association for Computational Linguistics.

Settaluri Lakshmi Sravanthi, Kishan Maharaj, Sravani Gunnu, Abhijit Mishra, and Pushpak Bhattacharyya. 2025. Understand the Implication: Learning to Think for Pragmatic Understanding. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 23778–23790, Vienna, Austria. Association for Computational Linguistics.

Gaurav Srivastava, Zhenyu Bi, Meng Lu, and Xuan Wang. 2025. Debate, train, evolve: Self-evolution of language model reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 32764–32810.

Yutao Sun, Mingshuai Chen, Tiancheng Zhao, Ruochen Xu, Zilun Zhang, and Jianwei Yin. 2025. The Self-Improvement Paradox: Can Language Models Bootstrap Reasoning Capabilities without External Scaffolding? In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6501–6512, Vienna, Austria. Association for Computational Linguistics.

Luong Trung, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. 2024. ReFT: Reasoning with reinforced fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7601–7614, Bangkok, Thailand. Association for Computational Linguistics.

Polina Tsvilodub, Kanishk Gandhi, Haoran Zhao, Jan-Philipp Fränken, Michael Franke, and Noah D. Goodman. 2025. Non-literal understanding of number words by language models. In *Proceedings of the 47th Annual Conference of the Cognitive Science Society*.

Saujas Vaduguru, Daniel Fried, and Yewen Pu. 2024. Generating pragmatic examples to train neural program synthesizers. In *The Twelfth International Conference on Learning Representations*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37.

Shengguang Wu, Shusheng Yang, Zhenglun Chen, and Qi Su. 2024. Rethinking Pragmatics in Large Language Models: Towards Open-Ended Evaluation and Preference Tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22583–22599, Miami, Florida, USA. Association for Computational Linguistics.

Huajian Xin, Z. Z. Ren, Junxiao Song, Zhihong Shao, Wanjia Zhao, Haocheng Wang, Bo Liu, Liyue Zhang, Xuan Lu, Qiushi Du, Wenjun Gao, Haowei Zhang, Qihao Zhu, Dejian Yang, Zhibin Gou, Z. F. Wu, Fuli Luo, and Chong Ruan. 2025. DeepSeek-Prover-V1.5: Harnessing proof assistant feedback for reinforcement learning and monte-carlo tree search. In *The Thirteenth International Conference on Learning Representations*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *The Twelfth International Conference on Learning Representations*.

Fangzhi Xu, Hang Yan, Chang Ma, Haiteng Zhao, Qiushi Sun, Kanzhi Cheng, Junxian He, Jun Liu, and Zhiyong Wu. 2025. Genius: A Generalizable and Purely Unsupervised Self-Training Framework For Advanced Reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13153–13167, Vienna, Austria. Association for Computational Linguistics.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. 2026. On-policy context distillation for language models. *Preprint*, arXiv:2602.12275.

Kefan Yu, Qingcheng Zeng, Weihao Xuan, Wanxin Li, Jingyi Wu, and Rob Voigt. 2026. The Pragmatic Mind of Machines: Tracing the Emergence of Pragmatic Competence in Large Language Models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 192–213, Rabat, Morocco. Association for Computational Linguistics.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, and 17 others. 2025. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. *Advances in Neural Information Processing Systems*, 38:113222–113244.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. 2024. Self-rewarding language models. In *International Conference on Machine Learning*, pages 57905–57923. PMLR.

Eric Zelikman, Georges Raif Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah Goodman. 2024. Quiet-STaR: Language models can teach themselves to think before speaking. In *First Conference on Language Modeling*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35.

Ruqing Zhang, Jiafeng Guo, Lu Chen, Yixing Fan, and Xueqi Cheng. 2022. A Review on Question Generation from Natural Language Text. *ACM Transactions on Information Systems*, 40(1):14:1–14:43.

Andrew Zhao, Yiran Wu, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. 2025. Absolute zero: Reinforced self-play reasoning with zero data. In *Advances in Neural Information Processing Systems*, volume 38.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. 2026. Self-distilled reasoner: On-policy self-distillation for large language models. *Preprint*, arXiv:2601.18734.

Anshun Asher Zheng, Junyi Jessy Li, and David I. Beaver. 2026. Strategic dialogue assessment: The crooked path to innocence. *Dialogue & Discourse*, 17:1–53.

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. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36.

A Data Construction

We construct the primary pragmatic QA data in two stages. First, we generate short-answer pragmatic QA instances from domain seeds and pragmatic section descriptions. Second, we audit the generated instances with a binary self-judging prompt and retain high-quality examples for training. The resulting filtered set is used as the primary problem data for both SFT target generation and GRPO training.

A.1 Pragmatic Sections

For data generation, each example is conditioned on one pragmatic section. A section description is a short natural-language definition of the type of pragmatic inference the generated question should require. Definitions of the six sections are given in Table 6.

A.2 QA Generation Prompt

To construct the primary Pragmatic QA data, we prompt the model to generate exactly one QA item for a given domain and pragmatic section. Each generated item contains a concrete context, a question, and a short answer. The prompt explicitly requires that the question be impossible to answer without pragmatic interpretation. We use a small pool of manually inspected few-shot examples to stabilize the format. Generated items are rejected if they cannot be parsed, have missing fields, or duplicate an earlier item under a normalized string signature.

QA Generation Prompt

```
System.
You are a pragmatic QA data generator.
Generate exactly one item for the target section below.
Target section: {SECTION_NAME}
Section description: {SECTION_DESCRIPTION}
You may reason before the final output if needed.
The final output must start with a line containing exactly FINAL_QA:
After FINAL_QA:, return plain text only with this exact format:
Question must be impossible to answer without pragmatic interpretation. Answer must be short and clear.
FINAL_QA:
Content: ...
Question: ...
```

Section	Description
Context and Deixis	Interpret meaning that depends on context by identifying the intended referent of under-specified or deictic expressions from the dialogue and current perspective.
Implicature and Presupposition	Recover non-explicit meaning by inferring what is implied under cooperative reasoning, and by identifying background assumptions required for an utterance to be interpretable.
Speech Acts and Intent Recognition	Infer what action an utterance performs in context, including cases where the intended act differs from the literal form.
Discourse and Coherence	Maintain a coherent interpretation across turns by linking utterances via discourse relations and tracking what has been established, updated, or answered.
Social Pragmatics	Interpret and respond appropriately when meaning is shaped by social context such as roles, power, culture, and interactional norms.
Metaphor	Interpret figurative language by mapping a non-literal description to the situation to recover the intended evaluation or action guidance, rather than a literal reading.

Table 6: Pragmatic sections used for self-generated QA data. Each section description is inserted into the QA generation prompt to guide the model toward examples requiring the corresponding type of pragmatic interpretation.

```
Answer: ...

No JSON and no extra commentary after
FINAL_QA:.

User.

Domain keyword: {DOMAIN}

Item: {ITEM_INDEX}/{ITEMS_PER_DOMAIN}

Field definitions:

- content: a concrete scenario/context only,
with enough detail for pragmatic inference.

- question: one explicit question about the
scenario.

- answer: a minimal, short, concise, single
correct answer to the question without any
explanation.

Section examples (style reference):

{FEW_SHOT_EXAMPLES}

Output plain text only with this exact
format:

FINAL_QA:

Content: ...

Question: ...

Answer: ...

No JSON, no markdown, no extra commentary
after FINAL_QA:.
```

A.3 Automatic Audit and Filtering

We audit each generated Pragmatic QA instance with a binary quality-judgment prompt. The auditor is given calibration examples from the same section, followed by the generated item to judge. An item is retained only if it is well-formed, unambiguous, answerable, and requires the intended type of pragmatic interpretation.

Automatic Audit and Filtering Prompt

```
System.

You are a dataset auditor for Pragmatic QA.
Judge whether the item is high-quality for
its SECTION. Answer yes for high-quality, no
for low-quality.

User.

Is this pragmatic QA example high-quality?

Criteria.

Pragmatic dependency: the gold answer
requires pragmatic interpretation, not just
literal reading.

Question correctness: the question itself is
not incorrect or ambiguous.

Gold answer: the gold answer is correct and
uniquely best-supported by the given context.

Few-shot examples for this same SECTION:
same-section accepted and rejected
calibration examples.

Example to judge:

SECTION: {SECTION};

SCENARIO: {CONTENT};

QUESTION: {QUESTION};

GOLD ANSWER: {GOLD_ANSWER}.

Answer just yes or no with no other output.
Final answer:
```

We apply the audit judge with the first-token margin m (§3.1) and discard the bottom 50% of generated items by m, while preserving balance across pragmatic sections.

B Counterfactual Answer Generation and Training

We train models in two sequential stages: supervised fine-tuning (SFT) followed by GRPO. Before

SFT, we generate target answers for the filtered primary problem data. The counterfactual condition uses a privileged counterfactual reasoning script during answer generation, while the student model is later trained without this script in the input. The non-counterfactual variant uses the same pipeline but replaces the counterfactual script with a lighter pragmatic QA prompt.

B.1 Answer Generation Prompts

For the counterfactual condition, we use a pragmatic QA prompt that explicitly instructs the model to interpret the observed utterance as a communicative choice among plausible alternatives. The model is asked to reason about what the speaker could have said under a different intention and to use this contrast to infer the intended meaning.

Counterfactual Answer Generation Prompt

System.

You are solving a pragmatic question-answering task.

Your goal is to choose the best answer by inferring the speaker's intended meaning in context.

Do not rely on the literal meaning of the utterance alone. Instead, interpret the utterance as a communicative choice made by a roughly rational and informative speaker. Think about why this speaker chose this utterance, in this context, given what the speaker likely knows.

When answering a question, use the following reasoning principles:

1. Identify the literal meaning of the utterance.

2. Use the context and shared background to determine what the speaker is likely trying to communicate.

3. Consider why the speaker chose this utterance instead of other plausible alternatives.

4. Assume the speaker is trying to provide relevant information in context, but may not say more than is needed.

5. Use the speaker's likely knowledge and the shared context to infer what the listener is expected to understand.

6. Choose the answer that best explains the utterance as a rational, informative, and contextually relevant choice.

When possible, justify your interpretation contrastively until you reach one clear interpretation:

- state one more direct, stronger, or more literal alternative the speaker could have said,

- explain what that alternative would have implied,

- then explain why the actual utterance suggests a different intended meaning.

Reason in the form:

"If the speaker had intended X, they would likely have said Y. Because they said Z instead, the intended meaning is more likely W."

Guidelines:

- Prefer the answer that matches the speaker's intended meaning, not just the surface wording.

- Use context, shared background, speaker knowledge, and plausible alternatives to interpret the utterance.

- Prefer interpretations that best explain the speaker's choice of wording.

- Do not infer more than the context, shared background, and the speaker's choice support.

For the non-counterfactual baseline, we use a lighter pragmatic QA prompt that asks the model to consider pragmatic meaning, but does not instruct it to explicitly contrast the observed utterance with alternative utterances. This provides a comparison between the counterfactual reasoning script and generic pragmatic prompting.

Non-counterfactual Answer Generation Prompt

System.

You are solving a pragmatic question-answering task.

Your goal is to choose the best answer by inferring the intended meaning of the utterance in the given context.

Do not rely on the literal meaning of the utterance alone. Instead, use the surrounding situation, discourse context, social setting, and background information provided in the problem to determine what the utterance communicates.

Focus on how the utterance functions in the specific scenario. Pay attention to what is explicitly stated, what is implied, what is presupposed, and what the context makes salient.

When answering a question, use the following reasoning principles:

1. Identify the literal meaning of the utterance.

2. Identify the relevant contextual cues: who is involved, what is happening, what has just been said or done, and what information is salient in the situation.

3. Determine the communicative function of the utterance, such as informing, requesting, refusing, warning, correcting, complaining, reassuring, joking, teasing, agreeing, disagreeing, implying, or expressing an attitude.

4. Use the context and shared background in the problem to infer what meaning goes beyond the surface wording.

5. Pay attention to pragmatic phenomena such as implicature, presupposition, deixis, indirect speech acts, politeness, irony, sarcasm, humor, metaphor, metonymy, discourse coherence, and social expectations when they are relevant.

6. Distinguish between what is directly said and what is pragmatically implied by the scenario.

7. Choose the answer that best captures the intended meaning while avoiding unsupported assumptions or over-interpretation.

When possible, justify your interpretation by grounding it in the provided context:

- state what the utterance literally says, - identify the contextual cues that matter, - explain how those cues shape the pragmatic meaning, - choose the answer that best matches the intended meaning in context.

Guidelines:

- Prefer the answer that matches the intended meaning, not just the surface wording.

- Use context, discourse coherence, shared background, social norms, and pragmatic cues to interpret the utterance.

- Prefer interpretations that are directly supported by the scenario.

- Do not infer more than the context and utterance support.

- Do not choose an answer merely because it is possible; choose the answer that is most coherent in context.

- Provide a concise reasoning trace followed by the final answer.

B.2 Correctness Judge Prompt

Judge Prompt

System.

You are a strict QA evaluator. Respond with exactly one word: either 'yes' or 'no'. Do not emit any other text, punctuation, or explanation.

User.

Task: QA answer grading.

CONTEXT: {CONTENT};

QUESTION: {QUESTION};

REFERENCE ANSWER: {REFERENCE};

CANDIDATE ANSWER: {CANDIDATE}.

Is the candidate answer semantically correct given the context, question, and reference answer? Answer with a single word: yes or no.

In all cases, the judge is the untuned instruct backbone matched in size to the policy under training, keeping the pipeline self-contained and free of external distillation.

We read the first-token log-probabilities for yes and no from the judge, convert them to probabilities, and compute the margin

$$m(x, a) = p(\mathsf{yes} \mid x, a) - p(\mathsf{no} \mid x, a),$$

where x is the problem (context, question, and reference answer) and a is the candidate answer. We accept a candidate if $m(x, a) > 0.8$. The choice of threshold is justified by the human-agreement study in §B.3.

B.3 Margin Calibration

To verify that the judge margin reflects meaningful correctness confidence, we conduct a small human-agreement calibration study. We sample 100 examples, and two authors independently assign one of three labels: *correct*, *incorrect*, or *skip*, based on whether the candidate answer is semantically correct given the context, question, and reference answer. We adopt a conservative aggregation rule: if either annotator assigns *skip*, the example is mapped to *incorrect*, regardless of the other annotation. For examples receiving no *skip* label, agreements are retained directly, while disagreements between *correct* and *incorrect* are discussed by the two annotators until consensus is reached. The resulting consensus labels serve as the human gold labels for threshold calibration. We then compare the margin-thresholded judge decisions against these labels.

The calibration results show that the judge margin is informative: increasing the threshold generally makes the judge more conservative, reducing recall while maintaining comparable precision. The best human-agreement F1 is obtained at $m > 0.7$ (0.800), while $m > 0.8$ remains a high-agreement operating point with precision 0.780, recall 0.812, F1 0.796, and accuracy 0.800. Very strict thresholds such as $m > 0.99$ substantially reduce recall, suggesting that overly conservative filtering discards many human-acceptable responses.

Rule	Precision	Recall	F1	Accuracy
$m > 0.5$	0.741	0.833	0.784	0.780
$m > 0.6$	0.741	0.833	0.784	0.780
$m > 0.7$	0.769	0.833	0.800	0.800
$m > 0.8$	0.780	0.812	0.796	0.800
$m > 0.9$	0.771	0.771	0.771	0.780
$m > 0.99$	0.767	0.688	0.725	0.750

Table 7: Agreement between self-judge margin thresholds and human labels on the manually reviewed calibration subset of 100 examples. Examples that either annotator marked as *skip* are mapped to *incorrect* in the gold label rather than excluded, so all 100 examples are retained. Precision, recall, F1, and accuracy are computed by treating the margin-thresholded judge decision as the prediction and the human label as gold.

We adopt $\tau = 0.8$ as a conservative midpoint that preserves judge precision. The threshold is fixed before running downstream experiments and is not tuned against pragmatic benchmark performance. The same value is used for all four backbones: Qwen3-8B, Qwen3-14B, Gemma-4-E4B, and GPT-OSS-20B.

B.4 Supervised Fine-Tuning

SFT Pregeneration. Let x denote an input instance from the filtered primary problem data, and let $s \in \mathcal{S}$ denote its pragmatic section label. We write $d(s)$ for the natural-language description of section s, and p_{cf} for the counterfactual pragmatic reasoning script used during response pregeneration. Let π_θ denote the base policy used to generate candidate SFT targets. For each retained problem, we construct an augmented teacher-side prompt

$$\tilde{x} = \text{Aug}(x, s) = [p_{\text{cf}};\ d(s);\ x],$$

which exposes the teacher to both the section description and an explicit counterfactual reasoning scaffold. Given $\tilde{x}$, the model samples a response

$$y \sim \pi_\theta(\cdot \mid \tilde{x}),$$

where $y = (r, a)$ consists of a reasoning trace r and a final answer a. We then apply a binary self-judge with the margin method (§B.2).

$$J(\tilde{x}, a) \in \{0, 1\},$$

which returns 1 only if the response is judged pragmatically correct with respect to the context, question, and reference answer. This yields the accepted set

$$\mathcal{D}_{\text{accept}} = \{(x, s, y) \mid J(\tilde{x}, a) = 1\}.$$

The final SFT dataset removes the privileged augmentation from the student input. Although the target response y is generated under $\tilde{x}$, the student is trained only on the original problem x paired with the accepted output:

$$\mathcal{D}_{\text{SFT}} = \{(x, y) \mid (x, s, y) \in \mathcal{D}_{\text{accept}}\}.$$

This asymmetry between teacher-side generation and student-side training is central to our design: the counterfactual reasoning script is used to construct high-quality reasoning traces, but the student must learn to produce such traces without seeing the script at inference time. We train with the standard causal language modeling objective on $\mathcal{D}_{\text{SFT}}$, masking prompt tokens and applying loss only to the assistant response:

$$\mathcal{L}_{\text{SFT}} = -\sum_{(x,y)\in\mathcal{D}_{\text{SFT}}} \sum_{t=1}^{|y|} \log \pi_\theta(y_t \mid x, y_{<t}).$$

SFT Hyperparameters. We use full-parameter fine-tuning with maximum sequence length 8192, bfloat16 precision, AdamW, cosine learning-rate schedule, learning rate 5×10^{-7}, two epochs, per-device batch size 1, gradient accumulation 4, warmup ratio 0.03, gradient clipping 1.0, and gradient checkpointing. Distributed runs use FSDP full-shard training with decoder-layer auto-wrapping and full-state-dict checkpointing.

B.5 Dataset Size and Training Budget

We first sample 1,000 domain seeds and pair each seed with each of the six pragmatic sections, yielding 6,000 seed–section generation prompts. All 6,000 generations are parsed into valid short-answer QA items. The self-filtering stage retains 3,000 primary problem instances, corresponding to 500 examples per pragmatic section. These 3,000 filtered problems are used as the GRPO training prompts.

For SFT, we generate target responses for the filtered primary problems using the counterfactual reasoning script and then apply the answer-quality judge. During Qwen3-14B training, this produces 2,816 accepted SFT targets, of which 2,759 are used for SFT training and 57 are held out as a synthetic validation split.

B.6 GRPO

We initialize the policy from the SFT checkpoint π_{SFT} and optimize it with GRPO (Shao et al., 2024)

Stage	Count
Domain seeds	1000
Seed–section generation prompts	6000
Parsed valid QA items	6000
Retained after self-filtering	3000
Accepted SFT targets	2816
SFT train split	2759
SFT validation split	57
GRPO training prompts	3000

Table 8: Aggregate data counts after answer generation and filtering. The accepted SFT targets are generated from the filtered primary problem data.

on the primary problem data. Below we first specify the algorithmic setup (reward, optimization objective) and then the implementation details (hyperparameters, infrastructure).

Easy-Prompt Filtering. Following DAPO (Yu et al., 2025), we apply an additional difficulty-based pass over the primary problem data before GRPO training. For each prompt x, we sample $G = 8$ rollouts from the SFT checkpoint π_{SFT} and score each rollout with the same correctness judge and margin m (§B.2) used during GRPO (threshold $\tau = 0.8$). A prompt is marked *easy* if every one of its G rollouts passes the judge, and is discarded. Such prompts yield zero-variance advantage estimates under group normalization and therefore contribute no gradient signal to the policy update. Filtering thus concentrates training on prompts at the frontier of π_{SFT}'s capability.

Reward Design. *Format reward.* $R_{\mathrm{fmt}}(y) \in \{0, 0.5, 1\}$ is a dense shaping signal that stabilizes early training by encouraging the model to maintain the structured output it acquired during SFT. A response receives 0.5 for containing exactly one well-formed pair of `<think>...</think>` tags, and an additional 0.5 for a valid `\boxed{}` answer in the post-thinking segment.

For each training prompt x, the policy π_θ samples a group of $G = 8$ candidate responses $\{y_1, \ldots, y_G\}$, where each y_i consists of a reasoning trace and a boxed final answer a_i. Each response is scored by a composite reward

$$R(x, y_i) = w_{\mathrm{fmt}}\, R_{\mathrm{fmt}}(y_i) + w_{\mathrm{ans}}\, R_{\mathrm{ans}}(x, y_i),$$

combining a format-compliance term with a pragmatic-correctness term.

Correctness reward. $R_{\mathrm{ans}}(x, y) \in \{0, 1\}$ reuses the judge margin $m(x, a)$ from §B.2. If `\boxed{}` extraction fails, we set $R_{\mathrm{ans}}(x, y) = 0$ without querying the judge. Otherwise we extract the candidate answer a from the rollout and assign $R_{\mathrm{ans}}(x, y) = 1$ if $m(x, a) > \tau$, with $\tau = 0.8$ fixed by the calibration in Appendix B.3.

Reward Scaling. We set $w_{\mathrm{fmt}} = 1$ and $w_{\mathrm{ans}} = 2$ so that correctness strictly dominates format. This asymmetric scaling prevents the policy from trading off pragmatic correctness for the denser, near-saturated format signal during early optimization. We use the smallest integer weighting that establishes this strict ordering rather than tuning the ratio against benchmark performance.

Optimization Objective. Given the per-response rewards, GRPO computes group-normalized advantages

$$\hat{A}_i = \frac{R(x, y_i) - \mathrm{mean}\Big(\{R(x, y_j)\}_{j=1}^{G}\Big)}{\mathrm{std}\Big(\{R(x, y_j)\}_{j=1}^{G}\Big)},$$

and updates the policy with a clipped PPO-style surrogate objective without a learned value function. Because the learning signal comes from *relative* quality differences within each group, training is robust to the absolute scale of the rewards.

GRPO Hyperparameters. We use full-parameter optimization with bfloat16 precision, AdamW, and a cosine learning-rate schedule with peak learning rate 4×10^{-6} and warmup ratio 0.1. The train batch size and PPO mini-batch size are both 128, with per-GPU micro-batch size 2. Maximum prompt and response lengths are 512 and $1{,}536$ tokens, respectively; overlong prompts are filtered out and the remainder are left-truncated. Following Srivastava et al. (2025); Xin et al. (2025); Shao et al. (2024); He et al. (2025), we regularize the policy toward π_{SFT} with a low-variance KL loss applied directly to the objective with coefficient 0.02, which discourages drift from the counterfactual reasoning behaviors acquired during SFT. We train for 4 epochs over the primary problem data.

GRPO Infrastructure. Each GRPO run uses 5 NVIDIA H200 GPUs: 4 GPUs host the policy and serve rollouts in-process via vLLM (one GPU per node, FSDP2 full-shard across the 4 nodes), and the remaining GPU runs a separate vLLM endpoint hosting the frozen instruct model as the judge. Rollouts are sampled at temperature 1.0, top-p 1.0, and top-k disabled, with $G = 8$ samples per prompt.

End-to-end wall-clock for the 4-epoch run on the primary problem data is approximately **2.5** hours on this configuration.

B.7 GRPO Checkpoint Selection

To avoid last-checkpoint or hand-picked bias, we report the checkpoint chosen by a fixed selection protocol. GRPO saves a checkpoint at every optimizer step and selects among these checkpoints as follows.

Held-out selection set. We construct the selection set from a fresh round of the self-generation pipeline described in Section A, run independently of the round used to produce the GRPO training data. From this fresh pool we draw a seeded, stratified sample of 100 examples per pragmatic section. To guarantee disjointness from training, any selection row whose (context, question) pair appears in the GRPO training data is discarded. The selection set is fixed with seed 42 across all checkpoints of all runs of a given model, so all checkpoints of a model are graded on exactly the same rows.

Scoring. Each checkpoint is scored on the selection set using the same correctness judge and margin $m(x, a)$ (§B.2) that define the GRPO training reward, with a row counted correct if $m(x, a) > 0.8$. This means selection accuracy is the GRPO training reward itself, computed on held-out data, rather than a separate evaluation metric. For decoding, we generate one rollout per row with temperature 0 and `max_tokens` $= 2048$, scoring each row independently.

Selection Rule. We pick the checkpoint with the highest selection accuracy. Exact ties are broken in favor of the later step, on the principle that the later step has absorbed strictly more of the training signal and is therefore the more conservative choice to promote.

Independence from Test Benchmarks. The selection set consists of self-generated pragmatic QA filtered by the same base-model judge used throughout our pipeline (matched in size to the trained model), and shares no items with PRAGMEGA, LUDWIG, METOQA, or ALTPRAG. Selection therefore cannot leak signal from these benchmarks, so the test numbers in Table 1 measure generalization beyond the selection pool.

C Benchmarks and Results

C.1 Benchmark Examples

Table 9 shows two of each benchmark example.

C.2 Exploratory Evaluation on Non-Cooperative Pragmatics

To examine whether counterfactual reasoning transfers to non-cooperative and adversarial settings, we evaluate PRAGREST using the Strategic Dialogue Assessment (SDA) framework introduced by Zheng et al. (2026). SDA evaluates courtroom cross-examinations as strategic exchanges, measuring whether a model can track how each response affects the speaker's position in the dialogue. Zheng et al. (2026) find that LLMs can rely on surface-level discourse cues when judging adversarial dialogue, sometimes treating damage control strategies such as hedging or deflection as neutral or positive rather than recognizing them as attempts to mitigate a harmful commitment. This setting therefore provides a complementary test of whether PRAGREST helps models reason beyond the cooperative surface form of an utterance.

We focus on three primary SDA metrics: **BaT** (Benefit at Turn), which measures alignment with human judgments of strategically beneficial moves; **PaT** (Penalty at Turn), which measures alignment with human judgments of strategically detrimental moves; and **NRBaT** (Normalized Relative Benefit at Turn), which captures the cumulative balance between benefits and penalties over the dialogue.

Quantitative Results. As shown in Table 10, the most consistent change appears on **PaT**. PRAGREST increases PaT for both Qwen3-8B (from -0.001 to 0.013) and Qwen3-14B (from 0.047 to 0.062). These gains are modest relative to variation across seeds, so we interpret them cautiously. Still, the consistent direction of the change suggests that counterfactual training may improve the model's ability to recognize when a response imposes a strategic cost on the speaker, rather than treating locally cooperative answers as neutral or beneficial.

The remaining SDA metrics show a more mixed pattern. For Qwen3-8B, PRAGREST improves NRBaT from 0.149 to 0.179, but decreases BaT from 0.132 to 0.051. For Qwen3-14B, PRAGREST improves BaT and PaT, while NRBaT remains essentially unchanged, moving from 0.075 to 0.073. We therefore interpret the SDA results as evidence for a targeted improvement in recognizing strate-

Benchmark	Example input	Expected output
PRAGMEGA	Scenario: Mary was asked about the town that she has just moved to. Mary responded: "This town is a chimney." What does Mary mean? Options: 1) The people living in this town are very welcoming. 2) The town is a chimney. 3) All houses in this town have chimneys. 4) Mary found a job at a company installing chimneys. 5) The town is not one of the cleanest ones.	5. The intended meaning is non-literal: Mary implies that the town is dirty or polluted.
PRAGMEGA	Scenario: Paul has to go to an interview and he is running late. While cleaning his shoes, he says to his wife Jane: "I want to wear that blue shirt, but it is very creased." What might he be trying to convey? Options: 1) He wants his wife to iron his shirt. 2) He is unhappy that his shirt is wrinkled. 3) He got this shirt on sale. 4) He got his shoes on sale.	1. Paul is making an indirect request for Jane to iron the shirt.
LUDWIG	Question: Is Marci grumpy? Response: "He's as gentle as a lamb."	No.
LUDWIG	Question: You... live here? Response: "Long time. Long, long time."	Yes.
METOQA	Context: She is attracted to blue jacket. Question: What does "blue jacket" refer to? Options: 1) Colour 2) Jacket 3) Sailor 4) Sea	3. Sailor.
METOQA	Context: His lovely voice caught my ear. Question: What does the sentence refer to? Options: 1) Giving attention 2) Noise 3) Whispering to the person 4) None	1. Giving attention.
ALTPRAG	Context: A teacher is speaking with Sam about his assignments and commitments, noting that Sam might be overwhelmed. Root: "You have too much on your plate." Candidate 1: "I appreciate your concern, but I think I can manage everything just fine." Candidate 2: "It's just how I like it; wouldn't have it any other way."	Candidate 1 directly reassures the teacher while politely asserting confidence. It is preferred because it acknowledges the teacher's concern rather than dismissing it.
ALTPRAG	Context: Melissa is reflecting on the new sales manager who just started a few days ago: "She is a babe in the woods." Root: "She is a babe in the woods." Candidate 1: "Wow, I hope she's ready for the steep learning curve!" Candidate 2: "Yeah, but honestly, who isn't at that level when starting fresh?"	Candidate 1 expresses concern about the manager's readiness and emphasizes the difficulty of the role, rather than normalizing her inexperience.

Table 9: Representative examples from the four pragmatic evaluation benchmarks used in our experiments. The examples illustrate the different forms of pragmatic interpretation tested by each benchmark: multiple-choice pragmatic QA in PRAGMEGA, binary implicature resolution in LUDWIG, metonymic reference resolution in METOQA, and open-ended implied-meaning recovery in ALTPRAG.

Model	BaT	PaT	NRBaT
Qwen3-8B			
Instruct	**0.132**±**0.036**	-0.001±0.033	0.149±0.058
PRAGREST	0.051±0.038	**0.013**±**0.033**	**0.179**±**0.034**
Qwen3-14B			
Instruct	0.112±0.023	0.047±0.061	**0.075**±**0.103**
PRAGREST	**0.138**±**0.054**	**0.062**±**0.074**	0.073±0.119

Table 10: Performance on the SDA framework. Values report mean Spearman's ρ correlations with human judgments across five seeds (using temperature sampling at 0.6), with standard deviations shown after $\pm$. Higher values indicate stronger alignment with human judgments.

gic penalties, rather than a uniform improvement across all dimensions of adversarial dialogue assessment.

Qualitative analysis. To better understand the PaT gains, we inspect turns where PRAGREST agrees with the human penalty judgment but the instruct backbone does not. Most recovered cases involve a change in how the model interprets the witness's answer: PRAGREST is more likely to recognize that the witness has conceded information that helps the opposing side. In SDA terms, this means identifying a response as strategically harmful even when it is locally clear, truthful, and relevant. Thus, the PaT gains suggest that PRAGREST is not simply rewarding answers for being clear or responsive. Instead, it more often recognizes when an apparently cooperative answer gives the opposing side useful information.

For example, when a witness is asked whether the defendant "voluntarily spoke with you in a tape-recorded interview without the presence of counsel" and answers "Yes," the instruct model recognizes that the response is clear and responsive, but still treats it as beneficial to the witness's side. PRAGREST instead identifies the strategic implication of the same answer: by confirming the questioner's premise, the witness gives the opposing side the concession it is seeking. Similar patterns appear when a witness gives a precise damaging answer ("Seven" abrasions), confirms a document detail ("Yes, it is"), concedes a contamination pathway ("it is likely"), or admits a lack of licensed qualification. In each case, PRAGREST treats the utterance not merely as a cooperative answer, but as a commitment whose strategic value depends on the question under discussion.

This behavior suggests that the model interprets

each utterance in relation to the adversarial context: it considers what the answer allows the questioner to infer and whether that inference advances the opposing side's case. In the recovered cases, PRAGREST often reasons over alternatives implicitly, recognizing that a direct answer rather than a hedge, a concession rather than a denial, or a clarification that still preserves a damaging inference can change which side the utterance benefits. This supports the quantitative pattern in Table 10: the most consistent gains appear in PaT, where success depends on recognizing when a response creates a strategic cost for the speaker.

D Details of the Counterfactual Error Analysis

This appendix describes the construction, validation, and use of the error taxonomy and counterfactual-reasoning scores used in Section 5.1.

D.1 Error Taxonomy

Error-case Collection. We collect incorrect PRAGMEGA predictions from each evaluated model. For every incorrect example, we retain the original prompt, answer options, gold answer, model prediction, and model reasoning trace. These traces are used only for diagnostic analysis, not for computing task accuracy.

Inducing the Error Taxonomy. We induce the error taxonomy in a bottom-up manner. Instead of manually specifying categories before inspecting the data, we prompt a language-model annotator to read batches of incorrect examples and propose recurring failure modes. The annotator is instructed to focus on the underlying pragmatic reasoning failure rather than superficial lexical differences. After inspecting the proposed categories, we merge near-duplicates and remove categories that are too broad, too rare, or outside the scope of pragmatic reasoning. We also remove categories that merely indicate that PRAGREST introduces our target behavior, since the goal is to characterize model errors rather than reward-specific style differences.

The final taxonomy contains the following non-exclusive tags.

- **Literal / surface bias**: the model anchors on literal wording or shallow semantic compatibility when the context requires a non-literal pragmatic interpretation.
- **Missed communicative intent**: the model fails to recover the speaker or listener's pragmatic goal, such as politeness, avoidance, deception, indirect request, complaint, or social positioning.
- **Unsupported or overextended inference**: the model over-reasons from weak cues, invents assumptions not licensed by the prompt, or post-hoc rationalizes an incorrect answer.
- **Coherence bridge error**: the model misjudges whether an implicit causal, temporal, or discourse bridge between events is warranted.
- **Figurative or humor mapping error**: the model fails to map figurative language, jokes, punchlines, or humorous incongruity to the intended interpretation.

Tags are multi-label: a single error may receive more than one tag if multiple failure modes are present.

D.2 Annotation Protocol and Human Validation

GPT Annotation Protocol. For full-scale annotation, we use a GPT-4.1-mini annotator. The annotator receives the prompt, gold answer, model prediction, and reasoning trace, along with the final taxonomy and short definitions of each error type. It is instructed to assign all applicable labels and to avoid assigning a pragmatic label when the failure is better explained by a concrete context or option misread. The model returns a structured label set for each incorrect example.

Human Validation. To check whether GPT labels are reliable enough for diagnostic analysis, we run a blind annotation study. Three human annotators independently label the same subset of examples using the same taxonomy, without seeing the model identity. Two annotators were members of the project team, and one annotator was an external annotator who was not involved in developing the method or analyzing the model outputs. The annotators were given anonymized benchmark items, gold answers, model predictions, reasoning traces, and the fixed error-category definitions, and were asked to assign all applicable error labels. The annotation task involved only model-generated text and benchmark examples; we did not collect personal or sensitive information from annotators, and

Comparison	Agreement	Micro-κ
Summary		
Human–Human avg.	83.8%	0.628
Human–GPT avg.	82.6%	0.614
Pairwise comparisons		
Human A (external) vs. Human B	80.0%	0.551
Human A (external) vs. Human C	87.7%	0.707
Human B vs. Human C	83.8%	0.625
Human A (external) vs. GPT	80.0%	0.557
Human B vs. GPT	86.7%	0.712
Human C vs. GPT	81.1%	0.572

Table 11: Agreement study for the induced error taxonomy. All annotators labeled the same shared subset of errors under a blind setting without access to model identities. Human A is an external annotator who was not involved in the project.

Annotator	Agreement	Micro κ	Micro ϕ	Macro κ
GPT-4.1-mini	82.60	0.614	0.619	0.497
GPT-4.1	79.63	0.586	0.622	0.465
GPT-5.5	77.24	0.548	0.594	0.424

Table 12: Agreement between human and LLM annotators for the induced pragmatic error taxonomy. Each LLM row reports the average agreement between that LLM annotator and the three human annotators. Agreement is computed over binary decisions for the five overlapping error categories.

the labels were used only to estimate agreement with the automatic error annotations. We compute pairwise agreement among humans and between each human and GPT.

The GPT annotator agrees with humans at approximately the same level as humans agree with one another. We therefore use GPT labels for the full analysis, but treat the resulting labels as a scalable diagnostic rather than as definitive ground truth.

D.3 Counterfactual Reasoning Score

For each reasoning trace, we compute a counterfactual-reasoning score using a GPT-based evaluator. The evaluator is instructed to judge only the reasoning trace, not whether the final answer is correct. It assigns five binary indicators corresponding to explicit counterfactual reasoning, alternative utterance or action, mismatch or contrast, speaker intent or pragmatic goal, and literal-versus-pragmatic contrast. The CF score is the sum of these indicators and ranges from 0 to 5. We use this score only for diagnostic analysis, not for training, filtering, checkpoint selection, or model evaluation.

Counterfactual Reasoning Score Prompt

System.

```
You are evaluating whether a model's
reasoning trace uses counterfactual pragmatic
reasoning.

Return only valid JSON. Do not judge whether
the final answer is correct; judge only the
reasoning trace.
```

User.

```
Score the reasoning trace using the rubric
below. Assign each field 0 or 1.

1. Explicit counterfactual.

Assign 1 if the trace explicitly reasons
about a counterfactual pragmatic or
communicative alternative, e.g., "if the
speaker meant X, they would/could have said
Y," "had she intended X," or "if this were
literal, we would expect . . . ". Do not count
generic uncertainty, ordinary causal
hypotheses, or narrative-coherence
alternatives unless the trace compares actual
behavior, wording, option choice, or
interpretation against an alternative
communicative/pragmatic possibility.

2. Alternative utterance or action.

Assign 1 if the trace identifies an
alternative wording, action, answer, or
response that would be expected under a
different intended meaning.

3. Mismatch or contrast.

Assign 1 if the trace explicitly notes a
mismatch or contrast between the
literal/surface reading and
contextual/pragmatic cues, or between an
option and what the speaker or situation
would normally imply.

4. Speaker intent or pragmatic goal.

Assign 1 if the trace reasons about
speaker/listener intent, social goal,
politeness, deception, irony, indirectness,
communicative purpose, or pragmatic meaning.

5. Literal vs. pragmatic contrast.

Assign 1 if the trace explicitly contrasts
literal meaning or face-value reading with
intended, pragmatic, figurative, indirect,
ironic, or non-literal meaning.

Compute cf_score as the sum of the five
binary fields, from 0 to 5.

Question prompt:

{FULL_PROMPT}

Model reasoning trace:

{REASONING}
```

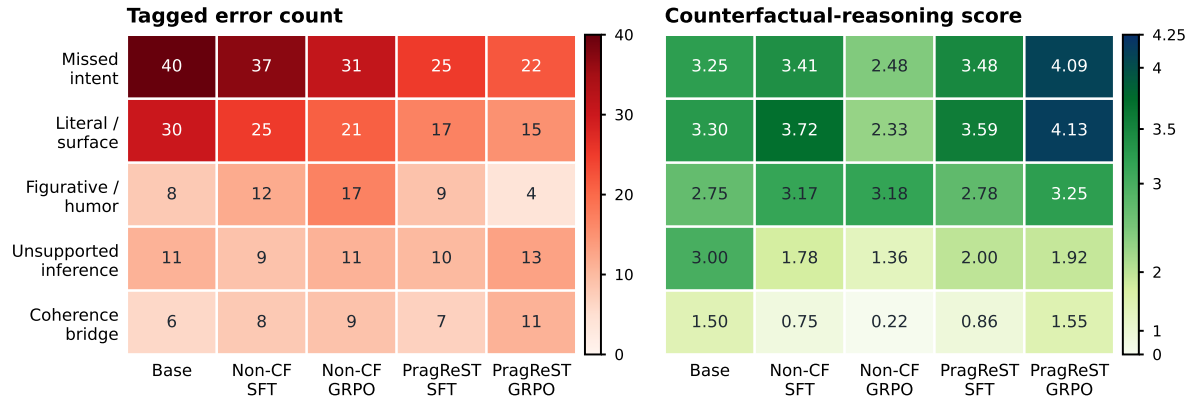


Figure 6: Full diagnostic breakdown of error reduction and counterfactual-reasoning scores across failure modes. Rows correspond to induced error categories, and columns are training conditions, with cells reporting the error-rate change and counterfactual-reasoning score statistics used in the main analysis.

E Artifact Licenses

All licenses are given in Table 13.

Artifact / Resource	License / Terms
Social IQa	CC BY 4.0
AltPrag	Apache License 2.0
Ludwig	CC BY 4.0
PragMega	Not specified
MetoQA (PUB)	MIT License
MMLU-Pro	MIT License
MATH-500	MIT License
AIME2025	MIT License
TruthfulQA	Apache License 2.0
Qwen3-8B	Apache License 2.0
Qwen3-14B	Apache License 2.0
Gemma-4-E4B	Apache License 2.0
GPT-OSS-20B	Apache License 2.0
OpenAI Models	OpenAI Services Agreement / Terms of Use

Table 13: All datasets and models were used in accordance with their intended use.

F Note on AI Usage

We used AI tools for grammar correction and code completion.