Teaching Language Models to Forecast Research Success Through Comparative Idea Evaluation

Srujan P Mule[1,2] **Aniketh Garikaparthi**[2] **Manasi Patwardhan**[2]

[1]IISER Pune [2]TCS Research

srujan.mule@students.iiserpune.ac.in

{aniketh.g, manasi.patwardhan}@tcs.com

Abstract

As language models accelerate scientific research by automating hypothesis generation, a critical evaluation bottleneck emerges. Current evaluation paradigms rely heavily on language-model judgments over subjective criteria like "excitement" or "novelty", which often fail to correlate with practical, empirical success. We study *comparative empirical forecasting*: given a benchmark-specific research goal and two candidate ideas, predict which will achieve better benchmark performance. We construct a dataset of 11,488 idea pairs grounded in objective outcomes from PapersWithCode. While off-the-shelf 8B-parameter models struggle (30% acc.), SFT dramatically boosts performance to 77.1%, outperforming GPT-5 (61.1%). By framing evaluation as a reasoning task via Reinforcement Learning with Verifiable Rewards (RLVR), we train models to discover latent reasoning paths, achieving 71.35% acc. with interpretable justifications. Through additional robustness and out-of-distribution evaluations, we show that the model is robust to surface-level heuristics and transfers well to both a cross-domain test set and an independently constructed test set. Our results demonstrate that compute-efficient small language models can serve as robust, objective evaluators, offering a scalable and practical evaluation framework for AI-assisted scientific discovery.

1 Introduction

Language Models are starting to function as autonomous research agents that can generate hypotheses, run experiments, and analyze results (Lu et al., 2024; Yamada et al., 2025; Gridach et al., 2025). A recurring pattern in these systems is high-throughput ideation, where the model generates hundreds of candidate methods for a given scientific goal (Baek et al., 2025; Si et al., 2024; Garikaparthi et al., 2025). This scale makes filtering "good ideas" crucial, as running hundreds of experiments is infeasible. However, current evaluation approaches rely on language-model judgments over *subjective* criteria like "excitement", "innovativeness" or "novelty" (Wang et al., 2024; Baek et al., 2025; Hu et al., 2024). While helpful, these metrics are often just proxies; an idea can be novel and well-argued but still fail to work in practice (Si et al., 2024; Zhu et al., 2025).

This gap motivates our study of *comparative empirical forecasting*: given a research goal and two candidate ideas, predict which idea will achieve better performance when evaluated on a benchmark. While these objective outcomes are very hard to predict, researchers routinely form useful intuitions from patterns across prior work to do so. We ask whether language models can be trained to internalize such priors and discriminate between two competing ideas *before* running experiments. Potentially, such a verifier model could complement the scale of generator models by shortlisting stronger ideas through pairwise comparison, minimizing the pool of candidates for implementation.

While recent work has begun exploring this direction by constructing datasets of idea comparisons (Wen et al., 2025), our goal is to push this setting toward (i) *fine-grained prediction* over specific benchmarks rather than coarse aggregation; (ii) *compute-efficient* models that are broadly accessible, and (iii) *interpretable reasoning* which can clarify the intuition behind predictions. Figure 1 illustrates how our fine-tuned 8B parameter model can make fine-grained predictions in context of the benchmark, while producing insightful reasoning to support its prediction.

To support this task, we construct a large-scale dataset by scraping public benchmark leaderboards to retrieve linked papers, and extract (a) benchmark-specific research goals, (b) descrip-

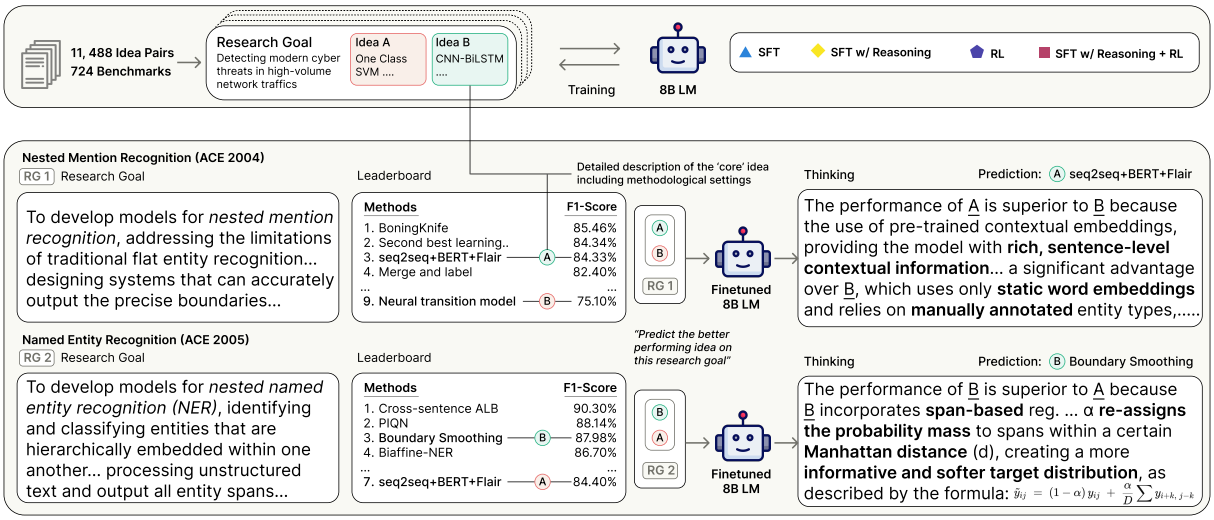

Figure 1: We explore various methods to fine-tune 8B Parameter Language Models using our constructed dataset. The figure illustrates the potential of our fine-tuned model to accurately judge the quality of scientific ideas. For the two given benchmarks and an overlapping method, our model robustly predicts which idea will perform better in context of the benchmark, while providing insightful reasoning.

tions representing competing ideas, and (c) empirical scores determining winners. This produces 11,488 labeled idea pairs across 724 valid benchmark leaderboards, grounded in objective outcomes.

As a first step, we cast *comparative empirical forecasting* as a direct preference-prediction problem and fine-tune language models to output a binary winner label. This black-box formulation serves as a simple baseline following prior work (Wen et al., 2025). Further to encourage and capture the intermediate reasoning, we use a two-stage training process. First, we use Supervised Fine-Tuning (SFT) on two curated dataset with reasoning traces alongside labels: a synthetically obtained subset from a large teacher model; a much smaller set grounded in contents of papers. Then fine-tune using Reinforcement Learning (RL) variants (Jia et al., 2025; Yu et al., 2025), to let the model explore and discover the reasoning paths which lead to the correct prediction.

Our results show that while base models struggle (Qwen3-8B achieving only 20.13% accuracy), supervised fine-tuning dramatically improves performance (77.1%). Variants trained to output interpretable reasoning achieve 71.35% accuracy, outperforming GPT-5 (61.10%) by over 10 percentage points while being substantially more compute-efficient and interpretable. Our models remain robust to stress tests on paraphrasing and recency, length and position bias, suggesting genuine task understanding rather than learning superficial heuristics. Our main contributions are:

- We introduce a large-scale dataset of research idea pairs with *benchmark-specific* research goals and outcomes, enabling fine-grained comparative forecasting (§4).
- We demonstrate that 8B-parameter models can outperform even frontier models on comparative scientific forecasting after fine-tuning (§7.1) and show non-trivial cross-domain generalization to non-NLP benchmarks and externally constructed datasets.
- We show that through careful training with reinforcement learning, models can produce coherent explanations to justify their predictions (§7.3).
- We analyze robustness under paraphrasing and presentation shifts to assess whether models rely on brittle heuristics (§B.8).

Finally, we present insights into the strengths and weaknesses of our work highlighting directions for future work on more groudned scientific idea evaluation.

2 Problem Statement

We define this problem formally as follows: Let $\mathcal{H}$ be the space of scientific hypotheses (ideas), $\mathcal{G}$ be the space of research goals and $\mathcal{C}$ be the space of reasoning traces. We construct a dataset $\mathcal{D} = \{g, h_A, h_B, (c), y\}$, where both h_A and $h_B \in \mathcal{H}$, are textual descriptions of two competing ideas, $g \in \mathcal{G}$ is the specific research goal for

which the ideas are implemented (e.g using One class SVM vs CNN-BiLSTM with the goal of detecting modern cyber threats in high-volume network traffic while minimizing false positives), and $y \in \{0, 1\}$ is a binary label where $y = 0$ implies that h_A outperforms h_B on goal g and $c \in \mathcal{C}$ denotes an optional chain-of-thought explaining why one idea outperforms the other. Our objective is to learn a parameterized policy π_θ that accurately predicts y given the context of the ideas and the goal, while generating chain of thought reasoning trace c before prediction.

3 Related Work

Research Ideation Research ideation, being inherently language-intensive, benefits significantly from advances in LLMs (Wang et al., 2024; Baek et al., 2025; Si et al., 2024). Recent efforts leverage frontier LLMs via retrieval (Li et al., 2024), test-time compute (Hu et al., 2024), or multi-agent debate (Su et al., 2025). Crucially, these ideas frequently fail to translate into real-world empirical improvements (Zhu et al., 2025; Si et al., 2025).

Evaluation Methodologies Most systems evaluate candidates using LLM-judges augmented with retrieval or agents (Baek et al., 2025; Garikaparthi et al., 2025). Assessments are typically rubric-driven, focusing on novelty, feasibility, and clarity (Li et al., 2024), occasionally calibrated via human studies (Si et al., 2024). Methodologically, scoring relies on absolute ratings (Baek et al., 2025) or aggregated pairwise rankings (Si et al., 2024; Garikaparthi et al., 2025). Verifiers that reward objective performance beyond surface plausibility remain largely underexplored (Wen et al., 2025).

LLMs for Forecasting LLMs have demonstrated potential as forecasters of real-world events, approaching competitive human crowd benchmarks (Halawi et al., 2024; Karger et al., 2025). Several works employ specialized training for such tasks (Lee et al., 2025; Chandak et al., 2025). Closest to our setting are efforts to forecast *empirical ML outcomes*: Wen et al. (2025) train GPT-4.1 to predict better-performing ideas from pairs, while Park et al. (2025) estimate benchmark scores from textual descriptions without experimentation.

4 Benchmark

A sample in our benchmark consists of: (i) **Idea Pair:** Detailed descriptions of two competing methods ($idea_A$, $idea_B$), grounded in their scientific publications. (ii) **Research Goal:** A clear statement of the specific evaluation objective of a benchmark for which the ideas are implemented (iii) **Binary Label:** A label (0 or 1) indicating which idea achieved a higher empirical score on that specific benchmark. We develop a pipeline to construct a benchmark dataset of idea pairs, transforming raw leaderboards into statistically grounded comparisons. The process involves:

Scraping and Paper Collection. We extract ideas from entries in live leaderboards, this allows us to build comparisons specific for each benchmark. Thus our evaluation becomes more fine-grained in comparison to parallel work (Wen et al., 2025), which can potentially conflate evaluations due to aggregation of scores across various benchmarks via majority voting.

We first scrape all available NLP leaderboards from `paperswithcode.com` that have at least two entries. This yields 1,918 benchmark leaderboards. For each entry in a leaderboard we identify a referenced paper , resulting in 5,713 **Result-Reporting (RR)** papers (excluding 7 behind paywall). We observe for some instances, the RR paper is not the **Original** paper which introduced the method, but rather the paper reporting results on the benchmark using that method. Relying on such papers for idea extraction (to be done in the later stage) would result in generic or incomplete descriptions. Therefore, we prompt an LLM (Gemini-2.5-pro, prompt in Appendix C) to verify whether each RR paper is the one that originally introduces the idea , and if not, look at the full content of RR paper for citations and the reference section to find the original paper. We also ask the LLM to report the confidence of its analysis (high/medium/low). Two NLP experts then manually process the low-confidence entries to verify the identified original paper citation and correct if necessary.We download an additional 908 Original papers based on this analysis. All downloaded papers are parsed using `s2orc-doc2json`[1] to convert the full text into Markdown format, providing clean and structured input for subsequent processing. Papers with unresolvable parsing errors are discarded, resulting in 5,695 RR and 832 Original markdown papers.

[1] https://github.com/allenai/s2orc-doc2json

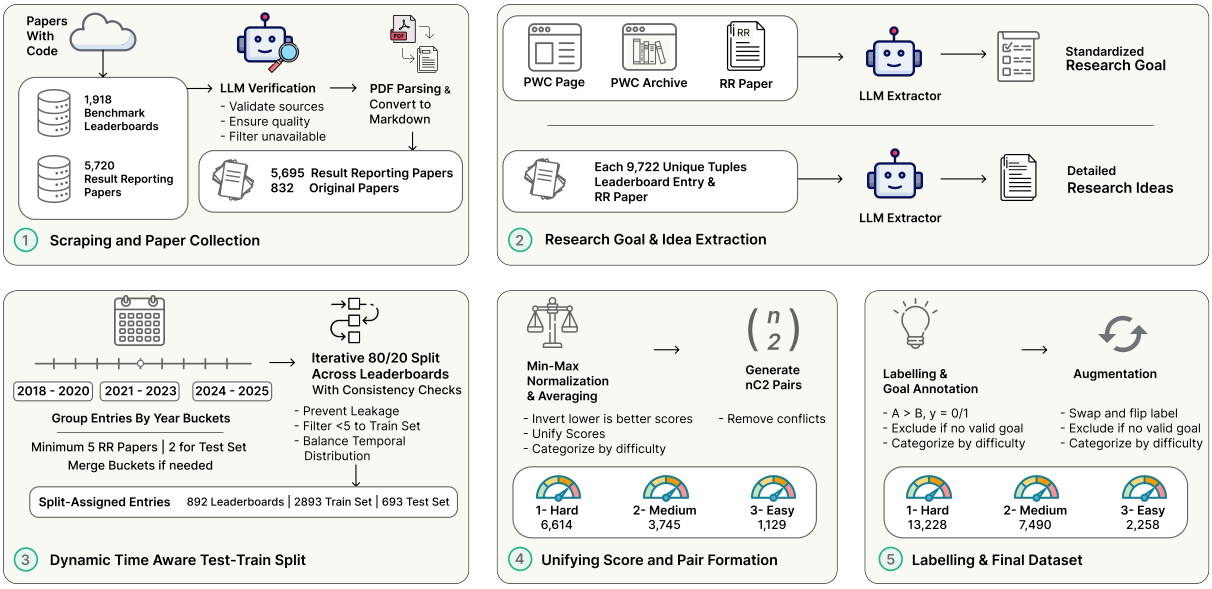

Figure 2: **Dataset Construction Pipeline** We use raw entries from 1,918 NLP leaderboards to construct a statistically grounded idea pairs with a benchmark specific research goal, while difficulty stratification ensures robust evaluation across diverse research goals.

Research Goal and Idea Extraction. For each one of the 1918 leaderboards, we extract a single canonical research goal from official benchmark descriptions in the following order of sources: (1) the dataset page on paperswithcode.com, (2) the corresponding dataset file from the `pwc-archive`[2], or (3) the RR paper when the above sources are unavailable (for 278 benchmarks). The extracted textual description is provided as input to an LLM (Prompt in Appendix C), which generates a clear, comprehensive research goal including what the benchmark evaluates. 327 such benchmarks with missing or unusable sources are skipped.

We process each RR and original markdown paper corresponding to the leaderboard entries with an LLM to extract the detailed idea, excluding any details, empirical results, comparisons, unique identifier like author/model names, year etc. The LLM (Prompt in Appendix C) has access to the complete paper context capturing all necessary details like algorithms, mathematical details etc., whenever present. This results in 9,722 total ideas.

Train–Test Split. We construct the train–test division based on the ideas from the extracted papers belonging to each leaderboard entry by iterating over the leaderoboards. Within a leaderboard, ideas are first grouped by their publication year into respective time buckets. Buckets with fewer than five unique papers are merged with adjacent years. Ideas in each time bucket are then split in an 80/20 ratio. As we iterate through leaderboards, we ensure that if an idea is already assigned to a split in a previous iteration, this assignment is strictly maintained. Leaderboards with fewer than four total ideas are assigned entirely to the training set. This approach helps prevent an idea appearing in both the train and test avoiding information leakage, while also ensuring similar temporal distribution of the ideas. This process yields 892 leaderboards, with 2,893 ideas in the training set and 693 in the test set.

Details of manual verification of the test set: To ensure that the LLM based idea extractions are accurate, we manually verify the correctness of the idea summary extracted from the parsed papers by consulting the original PDFs.

We observe that the errors/inaccuracies can be classified into 2 main categories:(i)**Incomplete (∼4%)**: When the summary falls short of the full detail necessary to correctly summarize the idea. For example, in one instance the description did not include the special loss function that was introduced as part of the idea. In such case we add the necessary details. (ii) **Incorrect (∼8%)**: When the details of the ideas doesn't correspond to actual ideas/methods. These can be further classified into 2 cases:(a) **Minor**:Minor mistakes like wrong output dimension etc.. Necessary changes are made

[2]https://huggingface.co/datasets/pwc-archive/datasets

in such cases. (b) **Major**:When the summary of the idea is completely incorrect. For example, hallucinated details like adversarial component in a BERT based system, when the original method involves simple fine-tuning. We remove such ideas.

Importantly, the verification criteria explicitly included checking for successful exclusion of empirical results and outcome statements from idea descriptions (lines of exclusion criteria in the extraction prompt) and ideas with residual performance comparisons were corrected or removed.

Unified Score and Pair Formation We next construct idea pairs within each benchmark. Relying solely on the benchmark leaderboard ranks to decide the winner of an idea pair can be misleading, as the performance gap between ranks can vary significantly across leaderboards. To obtain a consistent and quantitative basis for comparison, we compute a **Unified Score** for every idea within a benchmark. For each benchmark, we first apply min–max normalization for the results of each metric, across all entries and handle "lower-is-better" metrics (e.g., perplexity) by inversion. Approximately 85% of benchmarks report only a single metric. If there is more than one metric, the normalized results are averaged across all to yield a Unified Score for each idea. This procedure captures the relative performance distribution within a benchmark, regardless of metric scale or density.

All possible $\binom{n}{2}$ ideas pairs are generated for each benchmark within each split. We calculate the standard deviation (σ) of the Unified Scores across all entries in that benchmark and use it to define normalized score differences (Δ) for each pair. Based on Δ, we categorize pairs into three mutually exclusive difficulty tiers based on how close their unified scores are: **1σ (hard)**, **2σ (medium)**, and **3σ (easy)**, using a 20% tolerance margin (e.g., 0.8σ–1.2σ for 1σ). This categorization enables controlled difficulty evaluation based on empirical performance separation.

Labeling and Final Dataset. Given a pair $(\text{idea}_A, \text{idea}_B)$, we assign $y = 0$ if idea_A has better Unified Score and $y = 1$ otherwise. Each pair is annotated with the corresponding research goal of its benchmark. Benchmark leaderboards without a research goal and all the pairs from such benchmarks are removed. Given the asymmetric removal due to the lack of research goals, we end up with $\approx 90/10$ train/test split in the end. We release our dataset[3].

σ-Category	Train	/w Reasoning.	Test	Total
1-sigma	6120	120	494	6614
2-sigma	3461	45	284	3745
3-sigma	1038	5	91	1129
Total	10619	170	869	11488

Table 1: Stratified Dataset Statistics of Train, subset of train with reasoning traces and Test Idea pairs

Reasoning extraction Predicting empirical outcomes is challenging because research does not always follow clean, deductive logic; often, the explanatory "reasoning" consists of insights gained only *after* results are observed. Given the reasoning intensive nature of the task, we want Chain-of-Thought (CoT) reasoning traces to train the models. We extract CoT in 2 ways based on 2 opposite observations made in the current literature.

Synthetic RM-R1 (Chen et al., 2026) proposes an effective *distillation-then-RLVR* pipeline for training reasoning reward models: a smaller model is first trained via SFT on structured "Chain-of-Rubrics" traces distilled from a larger teacher, then refined with RLVR. Following this, we extract synthetic Chain-of-Rubrics from GPT-5 (high reasoning). We randomly sample 2,125 idea pairs from train and prompt the model with the research goal and both idea descriptions—mirroring the prediction task—to generate structured rubric-style reasoning traces evaluating each idea before selecting the better one. We retain only traces where GPT-5's prediction matches the ground truth, yielding 1,369 pairs, which expand to 2,738 training examples after swap augmentation.

Literature Grounded Wen et al. (2025) show that using *self-generated* CoT reasoning leads to performance degradation compared to the zero-shot setting for comparative prediction tasks. Following this, we take another approach to extract CoT reasoning. We consider all the idea pairs such that both of the ideas within each pair have the same RR paper. This way, we can be sure that such comparisons actually exist and are presented within the paper and not a case of inferred reasoning. We then prompt an LLM (Prompt in Appendix C) to look for the presence of any explanation for the better performance of one method over the other and extract this as a paragraph reflecting the grounded reasoning. In case such reasoning or

[3] https://anonymous.4open.science/r/Benchmark-Dataset-81B0

justification is not present, the LLM simply has to state that such reasoning is not available. The input prompt contains the full RR paper and a list of all the methods that were reported in this paper.

Cross domain Test Set We employ similar pipeline as described above to curate a new test set of idea pairs from **non-NLP** leaderboards (e.g. Molecular property prediction etc.) from PapersWithCode that have ≥ 3 entries, with the years of RR papers ≥ 2024. We use GPT-5 with high reasoning for idea extraction to introduce linguistic diversity relative to the training distribution. Additionally, we do not categorise the pairs based on difficulty, instead use individual metrics directly to form all $\binom{n}{2}$ pairs. This results in 705 idea pairs across 46 leaderboards.

Independently Constructed's Test set We source a manually verified test set from Wen et al. (2025), which has diverse domains. We further ensure no overlap with any of the ideas (and their RR or Original papers) in the train set by matching the title and removing any overlaps. This results in 1750 idea pairs that have labels based on majority voting across multiple benchmarks.

5 Methodology

We first treat our task as a binary classification problem, and later probe models to also output interpretable reasoning for their judgments. Hence we train under two distinct settings: **Supervised Fine-Tuning (SFT)** for direct prediction and **Reinforcement Learning (RL)** for latent reasoning.

5.1 Supervised Fine-Tuning (SFT)

Base models (LMs) typically lack the comparative intuition needed to map idea differences to benchmark performance. To bridge this gap, we employ standard SFT to train the model so that it can learn and do better at this task.

The model is provided with the research goal g and the descriptions of two ideas h_A, h_B. The target output is simply the binary label $y \in \{0, 1\}$ corresponding to the empirically superior idea.

This phase utilizes the full train dataset to ground the model in the "scientific intuition" of identifying successful ideas.

5.2 Reinforcement Learning (RL)

Additionally, we want the model to reason before making a prediction. We treat reasoning as a latent variable to be optimized via Reinforcement Learning. We structure this as a two-step process within the RL framework.

1. Cold Start Finetuning (SFT-Reasoning) Initial experimentation with pure RL (DAPO) applied to the LM, using the reward and objective function defined in the next paragraph, revealed consistent reward hacking (Amodei et al., 2016) and generation of incomplete reasoning traces. To address these problems and to ensure that the model can generate coherent reasoning structures, we fine-tune the model on the small subset of available reasoning traces.

This step aims to teach models a *style* of scientific argumentation (e.g., "Idea A reduces variance by... therefore it is likely to outperform Idea B") and align well with human or Chain-of-Rubrics from a strong teacher model.

2. Variants of Group Relative Policy Optimization (GRPO) We initialize our policy π_θ and train on the remaining dataset using two variants of Group Relative Policy Optimization (GRPO) that address some of it's limitations (Yu et al., 2025; Liu et al., 2025b). For a given input context $x = (g, h_A, h_B)$, the model samples a group of G outputs $\{o_1, \ldots, o_G\}$, where each output $o_i \in G$ comprises a generated reasoning trace c_i and a prediction $\hat{y}_i$. The policy is optimized using the advantage estimate A_i derived from the group rewards. By optimizing for the final outcome, the model is incentivized to discover reasoning traces c that lead to the correct empirical prediction y.

Reward Function The total reward $R(o)$ is the sum of a correctness score r_{cor} and a formatting score r_{fmt}, defined as follows: the correctness score $r_{\text{cor}}(o)$ is $+3.0$ if the predicted label $\hat{y}$ matches the true label y, and -3.0 otherwise. The formatting score $r_{\text{fmt}}(o)$ is 0.5 times the difference between the indicator function $\mathbb{I}_{\text{think}}$ (true if the "think" token is present) and $\mathbb{I}_{\neg\text{think}}$ (true if the "think" token is absent), representing the "$\langle think \rangle$" component (± 0.5), plus 0.5 times the difference between $\mathbb{I}_{\text{ans}}$ (true if the "Answer:" token is present) and $\mathbb{I}_{\neg\text{ans}}$ (true if the "Answer:" token is absent), representing the "Answer:" component (± 0.5). Here, $\mathbb{I}(x) = 1$ if x is true and 0 otherwise.

DAPO utilizes global token normalization and decoupled clipping to encourage exploration,

while also addressing the known length bias of GRPO during training.

Dr. GRPO To fix the length bias (during training) in standard GRPO, we use Dr. GRPO , where the advantage centered but not scaled: the standard deviation term in the denominator is removed.

6 Experiments

6.1 Metric

We design our evaluation to mitigate LLM's vulnerabilities to position bias. A prediction **consistent** if the model predicts the same idea to be better for both original and swapped position pairs. For accuracy, an idea pair is considered correctly classified only if the prediction is consistent *and* correct.

6.2 Language Models

We evaluate two open-source models: **Qwen3-8B** (Yang et al., 2025), hereafter called Qwen3 and **Llama3.1-8B-Instruct** (AI@Meta, 2024) hereafter called Llama3.1. We also use Gemini 2.5 Flash (Comanici et al., 2025) and GPT-5 (OpenAI, 2025) for comparison. All training was done with BF16 precision on NVIDIA-A100-40GB GPUs.

Direct label prediction is done with "reasoning" mode turned off (no think tokens) and reasoning models with "reasoning" turned on in case of Qwen3.

6.3 Training and Hyperparameter

We use the train/test split defined in our benchmark section §4 (Table 1). We further split the train set into 90% actual train and 10% validation following the σ-category distribution.

SFT: We use LoRA rank of 64, 128 α, 0.1 dropout, batch size of 2, learning rate of 2e-4, weight decay of 0.01, *cosine* learning rate scheduler, 1 epochs and default temperature. We tune these hyper-parameters using the validation set.

RL: We use LoRA rank 64, 128 α, 0 dropout, batch size 1, learning rate 5e-6, weight decay 0.01, group size 4, $\beta = 1e-5$, max output token length 3600 and default temperature (Yang et al., 2025; AI@Meta, 2024). We use Unsloth library along with *vllm* for fast inference. Very small, yet nonzero β, corresponding to the KL-Divergence penalty allow us to have a regularization effect for consistent grammar (Liu et al., 2025a).

7 Results and Discussion

7.1 Can LMs predict better ideas based on their likely empirical outcomes?

Table 2 compares the untrained models against their SFT counterparts. We observe that untrained models perform poorly with Qwen3-8B scoring 25.31% and Llama3 scoring a 30.02%, since we consider inconsistent predictions to be wrong by default, the accuracies account for position bias and remain around random guessing (25%).

However, Direct-SFT yields dramatic improvements. **Qwen3** reaches 77.10% accuracy. This crucial result demonstrates that even SLMs can predict the better idea based on their likely empirical outcome, and this doesn't require frontier models as hypothesized by (Wen et al., 2025).

We also observe that σ based categorization captures comparison difficulty well. In most fine-tuned models, the accuracies on $1\sigma < 2\sigma < 3\sigma$.

Cross Domain Test We observe that all trained *Qwen3* models perform at par or better than GPT-5 (under all the reasoning effort, and zero-shot setting) except Synthetic-Reason-SFT-DAPO. Furthermore, we see that RL tuned models perform better than *Direct-SFT* ($\approx$ 3%) showing their learning is more robust (Table 2) in general, showing non trivial generalization and robustness to label aggregation methodology.

Independently Constructed Test Set. We further validate generalization on a dataset with zero methodological overlap to our pipeline. Our fine-tuned *Reason-SFT-DrGRPO* achieves **67.49%**, outperforming the Wen et al. zero-shot GPT-4.1 + retrieval system (51.4%) by over 16 points despite using a model 50$\times$ smaller and *no* retrieval augmentation.

7.2 The Role of Reasoning

To assess the importance of reasoning for our task, we study how allowing a LM to think before answering affects predictive performance. We operationalize "reasoning" differently across model families. For Qwen3, we compare the standard (non-thinking) variant against its thinking counterpart. For GPT-5, we vary the thinking budget (low, medium, high). For Llama 3.1, we prepend a CoT instruction and ask the model to reason before producing its final prediction.

Across these settings, models that are explicitly trained to reason benefit more reliably from delib-

Model / Method	1-σ	2-σ	3-σ	Overall	CD Test
Qwen3					
Base	18.42	26.05	10.99	20.14	3.55
Base (Reasoning)	15.38	27.11	26.11	25.31	12.62
Direct-SFT	**70.85**	**85.56**	**84.62**	**77.10**	45.67
Reason-SFT	35.32	38.90	45.05	37.51	29.31
Reason-DAPO	69.43	75.00	83.52	72.73	45.96
Reason-SFT-DAPO	64.57	79.23	83.52	71.35	48.37
Synthetic-Reason-SFT-DAPO	65.79	72.53	74.72	68.93	41.10
Reason-SFT-DrGRPO	66.19	76.41	83.52	71.35	**49.08**
Llama3.1					
Base	37.36	31.33	27.93	30.03	3.83
Base (Reasoning)	26.52	30.63	21.98	27.39	18.22
Direct-SFT	53.64	58.10	67.03	56.50	31.20
GPT-5					
Reasoning (low)	58.70	58.45	49.45	57.65	42.84
Reasoning (med)	59.10	61.62	56.04	59.61	45.25
Reasoning (high)	61.94	61.27	56.04	61.10	45.96
Gemini 2.5 Flash					
Base (Reasoning)	41.90	40.14	36.26	40.73	-

Table 2: Accuracy (%) breakdown across different difficulty subsets i.e. (σ)-categories, on CD test set and models.**Bold**: Best, Underline: Second Best

eration. Qwen3 improves by $\sim$ 5% points with thinking variant. GPT-5 shows a consistent (but diminishing) accuracy gain as the thinking budget increases. In contrast, naïvely prompting CoT can be counterproductive: Llama 3.1 reduces accuracy from 30.02% to 27.38%.

7.3 Does RL Induce Reasoning?

We further explore whether we can induce correct reasoning. The initial SFT with reasoning (*Reason-SFT*/*Synthetic-Reason-SFT*) achieves an overall accuracy of 37.51% and 25.54% respectively. While Reason-SFT improves over the Base model, Sythetic-Reason-SFT shows negligible improvement. These results confirm that synthetic reasoning traces do not provide learnable grounding for this task. However, RL proved highly effective in recovering predictive performance in both cases. *Reason-SFT-DAPO* and *Reason-SFT-DrGRPO* achieve accuracies of $\approx$ 71% (Table 2), narrowing the gap with the *Direct-SFT*.s

While RL mostly restored the accuracy, an inspection of the generated reasoning traces during intermediate stages revealed gaps:
(i) **Reason-DAPO** stops generating any reasoning, while still receiving format rewards and attending only towards the final answer. To prevent this, we introduce a penalty for responses shorter than 600 characters and tweak β to $1e-4$. This results in the model repeating 3-4 sentences. (ii) **Reason-SFT-DrGRPO** tends to minimize the reasoning trace over the training iterations. The outputs often devolve into superficial justifications (e.g., stating one idea is better because it is "more recent"). However, our final set of best results (iii) **Reason-SFT-DAPO** produce consistent and coherent reasoning traces prior to the final answer, while being resilient to forms of reward hacking seen, showcasing that it is possible to induce interpretable reasoning in LMs.

8 Conclusion

In this work we demonstrate that language models can be taught to forecast research success be leveraged as a reliable objective verifier for scientific ideas. By constructing a large-scale high-quality dataset, we enable fine-grained prediction grounded in objective outcomes. Using only 8B parameters, our fine-tuned models achieve 77.1% accuracy, outperforming frontier models by over 15 % points. Our experiments with RL-trained models reach 71.35% accuracy while generating coherent justifications for their predictions. Our work offers a path toward closing the loop on autonomous scientific discovery by developing as grounded reward models for AI Agents.

Limitations

Our benchmark could potentially inherit noise from upstream leaderboard sources. In particular, the dataset is based on `paperswithcode.com` records that were scraped shortly before the site shut down. While a similar SOTA index is available via HyperAI[4], our core methodology (aligning paper claims with live leaderboard entries and constructing outcome-supervised comparisons) is not tied to any single provider and should transfer to other actively maintained leaderboards.

While we motivate the fine-tuned model as a potential component for shortlisting ideas via pairwise ranking and test their utility in ranking ideas within a leaderboard (§B.11), we do not yet provide sufficient experiments to quantify its effectiveness in an ideation workflow.

Scope-wise, the current dataset is restricted to NLP benchmarks, reflecting the manual effort, cost, and time required for data construction and validation, as well as the scope of our experiments. Extending the approach to additional domains and task families beyond NLP remains future work.

[4] https://hyper.ai/en/sota/category/natural-language-processing

Acknowledgement

We thank Jiaxin Wen and authors of Wen et al. (2025) for providing us with their private test set for evaluation, which helped us demonstrate the transfer of our trained models to new domains and heterogeneous data sources.

References

AI@Meta. 2024. The llama 3 herd of models. *ArXiv*, abs/2407.21783.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety. *Preprint*, arXiv:1606.06565.

Iván Arcuschin, Jett Janiak, Robert Krzyzanowski, Senthooran Rajamanoharan, Neel Nanda, and Arthur Conmy. 2025. Chain-of-thought reasoning in the wild is not always faithful. *Preprint*, arXiv:2503.08679.

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. 2025. Researchagent: Iterative research idea generation over scientific literature with large language models. *Preprint*, arXiv:2404.07738.

Nikhil Chandak, Shashwat Goel, Ameya Prabhu, Moritz Hardt, and Jonas Geiping. 2025. Scaling open-ended reasoning to predict the future. *Preprint*, arXiv:2512.25070.

Xiusi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru WANG, Yu Zhang, Denghui Zhang, Tong Zhang, Hanghang Tong, and Heng Ji. 2026. RM-r1: Reward modeling as reasoning. In *The Fourteenth International Conference on Learning Representations*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Aniketh Garikaparthi, Manasi Patwardhan, Lovekesh Vig, and Arman Cohan. 2025. Iris: Interactive research ideation system for accelerating scientific discovery. *Preprint*, arXiv:2504.16728.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2024. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595, Mexico City, Mexico. Association for Computational Linguistics.

Shashwat Goel, Rishi Hazra, Dulhan Jayalath, Timon Willi, Parag Jain, William F. Shen, Ilias Leontiadis, Francesco Barbieri, Yoram Bachrach, Jonas Geiping, and Chenxi Whitehouse. 2025. Training ai co-scientists using rubric rewards. *Preprint*, arXiv:2512.23707.

Mourad Gridach, Jay Nanavati, Khaldoun Zine El Abidine, Lenon Mendes, and Christina Mack. 2025. Agentic ai for scientific discovery: A survey of progress, challenges, and future directions. *Preprint*, arXiv:2503.08979.

Xuemei Gu and Mario Krenn. 2025. Forecasting high-impact research topics via machine learning on evolving knowledge graphs. *Machine Learning: Science and Technology*, 6(2):025041.

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

Alexander Gurung and Mirella Lapata. 2025. Learning to reason for long-form story generation. *Preprint*, arXiv:2503.22828.

Danny Halawi, Fred Zhang, Chen Yueh-Han, and Jacob Steinhardt. 2024. Approaching human-level forecasting with language models. *Preprint*, arXiv:2402.18563.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Xiang Hu, Hongyu Fu, Jinge Wang, Yifeng Wang, Zhikun Li, Renjun Xu, Yu Lu, Yaochu Jin, Lili Pan, and Zhenzhong Lan. 2024. Nova: An iterative planning and search approach to enhance novelty and diversity of llm generated ideas. *Preprint*, arXiv:2410.14255.

Ruipeng Jia, Yunyi Yang, Yongbo Gai, Kai Luo, Shihao Huang, Jianhe Lin, Xiaoxi Jiang, and Guanjun Jiang. 2025. Writing-zero: Bridge the gap between non-verifiable tasks and verifiable rewards. *Preprint*, arXiv:2506.00103.

Ezra Karger, Houtan Bastani, Chen Yueh-Han, Zachary Jacobs, Danny Halawi, Fred Zhang, and Philip E. Tetlock. 2025. Forecastbench: A dynamic benchmark of ai forecasting capabilities. *Preprint*, arXiv:2409.19839.

Esther Landhuis. 2016. Scientific literature: Information overload. *Nature*, 535:457 – 458.

Sang-Woo Lee, Sohee Yang, Donghyun Kwak, and Noah Y. Siegel. 2025. Advancing event forecasting through massive training of large language models: Challenges, solutions, and broader impacts. *Preprint*, arXiv:2507.19477.

Long Li, Weiwen Xu, Jiayan Guo, Ruochen Zhao, Xingxuan Li, Yuqian Yuan, Boqiang Zhang, Yuming Jiang, Yifei Xin, Ronghao Dang, Deli Zhao, Yu Rong, Tian Feng, and Lidong Bing. 2024. Chain of ideas: Revolutionizing research via novel idea development with llm agents. *Preprint*, arXiv:2410.13185.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. 2025a. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *Preprint*, arXiv:2505.24864.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025b. Understanding r1-zero-like training: A critical perspective. *Preprint*, arXiv:2503.20783.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The ai scientist: Towards fully automated open-ended scientific discovery. *Preprint*, arXiv:2408.06292.

Hanane Nour Moussa, Patrick Queiroz Da Silva, Daniel Adu-Ampratwum, Alyson East, Zitong Lu, Nikki Puccetti, Mingyi Xue, Huan Sun, Bodhisattwa Prasad Majumder, and Sachin Kumar. 2025. Scholareval: Research idea evaluation grounded in literature. *Preprint*, arXiv:2510.16234.

Charles O'Neill, Tirthankar Ghosal, Roberta Răileanu, Mike Walmsley, Thang Bui, Kevin Schawinski, and Ioana Ciucă. 2025. Sparks of science: Hypothesis generation using structured paper data. *Preprint*, arXiv:2504.12976.

OpenAI. 2025. Gpt-5 system card. Accessed: 2026-01-05.

Vardhan Palod, Karthik Valmeekam, Kaya Stechly, and Subbarao Kambhampati. 2025. Performative thinking? the brittle correlation between cot length and problem complexity. *Preprint*, arXiv:2509.07339.

Jungsoo Park, Ethan Mendes, Gabriel Stanovsky, and Alan Ritter. 2025. Look before you leap: Estimating llm benchmark scores from descriptions. *Preprint*, arXiv:2509.20645.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2018. Scikit-learn: Machine learning in python. *Preprint*, arXiv:1201.0490.

Zhenting Qi, Mingyuan Ma, Jiahang Xu, Li Lyna Zhang, Fan Yang, and Mao Yang. 2024. Mutual reasoning makes smaller llms stronger problem-solvers. *Preprint*, arXiv:2408.06195.

Soumya Rani Samineni, Durgesh Kalwar, Vardaan Gangal, Siddhant Bhambri, and Subbarao Kambhampati. 2025. Local coherence or global validity? investigating rlvr traces in math domains. *Preprint*, arXiv:2510.18176.

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.

Chenglei Si, Tatsunori Hashimoto, and Diyi Yang. 2025. The ideation-execution gap: Execution outcomes of llm-generated versus human research ideas. *Preprint*, arXiv:2506.20803.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. 2024. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. *Preprint*, arXiv:2409.04109.

Michael D. Skarlinski, Sam Cox, Jon M. Laurent, James D. Braza, Michaela Hinks, Michael J. Hammerling, Manvitha Ponnapati, Samuel G. Rodriques, and Andrew D. White. 2024. Language agents achieve superhuman synthesis of scientific knowledge. *Preprint*, arXiv:2409.13740.

Haoyang Su, Renqi Chen, Shixiang Tang, Zhenfei Yin, Xinzhe Zheng, Jinzhe Li, Biqing Qi, Qi Wu, Hui Li, Wanli Ouyang, Philip Torr, Bowen Zhou, and Nanqing Dong. 2025. Many heads are better than one: Improved scientific idea generation by a llm-based multi-agent system. *Preprint*, arXiv:2410.09403.

Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA.

Xuli Tang, Xin Li, Ying Ding, Min Song, and Yi Bu. 2020. The pace of artificial intelligence innovations: Speed, talent, and trial-and-error. *Journal of Informetrics*, 14(4):101094.

Keisuke Ueda, Wataru Hirota, Takuto Asakura, Takahiro Omi, Kosuke Takahashi, Kosuke Arima, and Tatsuya Ishigaki. 2025. Exploring design of multi-agent llm dialogues for research ideation. *Preprint*, arXiv:2507.08350.

Qingyun Wang, Doug Downey, Heng Ji, and Tom Hope. 2024. SciMON: Scientific inspiration machines optimized for novelty. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 279–299, Bangkok, Thailand. Association for Computational Linguistics.

Jiaxin Wen, Chenglei Si, Yueh han Chen, He He, and Shi Feng. 2025. Predicting empirical ai research outcomes with language models. *Preprint*, arXiv:2506.00794.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. 2025. The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *Preprint*, arXiv:2504.08066.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Kevin Zhou, Adam Dejl, Gabriel Freedman, Lihu Chen, Antonio Rago, and Francesca Toni. 2025. Evaluating uncertainty quantification methods in argumentative large language models. *Preprint*, arXiv:2510.02339.

Minjun Zhu, Qiujie Xie, Yixuan Weng, Jian Wu, Zhen Lin, Linyi Yang, and Yue Zhang. 2025. Ai scientists fail without strong implementation capability. *Preprint*, arXiv:2506.01372.

A Additional Benchmark Dataset Construction details

This appendix provides the mathematical and algorithmic details of the benchmark dataset construction pipeline described in the main text.

A.1 Metric selection and Normalization

Metric Cleaning: To combine heterogeneous metrics, we only used metric columns that were universally reported for all entries in a benchmark. If no metric had universal coverage, the benchmark was skipped.

Normalization: Let $m_i^{(k)}$ be metric k for entry i. We normalize:

$$\tilde{m}_i^{(k)} = \frac{m_i^{(k)} - \min_k}{\max_k - \min_k} \tag{1}$$

Direction check: We calculate the Pearson correlation between $\tilde{m}^{(k)}$ and the rank r. If $\text{corr}(\tilde{m}^{(k)}, r) > 0$ (implying higher value = worse rank, e.g., perplexity), we invert:

$$\hat{m}_i^{(k)} = 1 - \tilde{m}_i^{(k)} \tag{2}$$

Unified Score: The final score s_i is the arithmetic mean across adjusted metrics:

$$s_i = \frac{1}{|M|} \sum_{k \in M} \hat{m}_i^{(k)} \tag{3}$$

A.2 Discordance Removal

To ensure the unified scores broadly agree with leaderboard ranks, we use a strict pairwise discordance test. A pair (i, j) is discordant if:

$$\text{discordant}(i,j) = \begin{cases} 1 & \text{if } (r_i < r_j \text{ and } s_i > s_j) \\ & \text{or } (r_i > r_j \text{ and } s_i < s_j) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Algorithm 1 Iterative discordance removal

Require: Entries with ranks and unified scores.
1: Compute discordance fraction $f = D/\binom{n}{2}$.
2: **while** $f > 0$ and at least 2 entries remain **do**
3: Identify entry involved in max discordant pairs.
4: Remove entry.
5: Recompute f.
6: **end while**

A.3 Time Bucketing

To handle temporal shifts, we grouped entries by year. A bucket was valid only if it contained at least 5 unique papers and 2 test papers (post global-split).

Algorithm 2 Time bucketing and validation

Require: Entries with years and global split.
1: Create initial year-based buckets.
2: **while** bucket has <5 papers or <2 test papers **do**
3: Merge bucket with adjacent one.
4: **end while**

A.4 Pair Generation and Augmentation

Pairs are produced within the same benchmark and bucket using the standardized difference $\Delta_{ij} = |s_i - s_j|/\sigma$.

Sigma Categories:

- **1-sigma:** $0.8 \leq \Delta_{ij} \leq 1.2$
- **2-sigma:** $1.8 \leq \Delta_{ij} \leq 2.2$
- **3-sigma:** $2.8 \leq \Delta_{ij} \leq 3.2$

Labeling and Augmentation: For each valid pair (i, j) where $s_i > s_j$:

- Generate record: $\{idea\_A : i, idea\_B : j, label : 1\}$
- Generate swap: $\{idea\_A : j, idea\_B : i, label : 0\}$

This augmentation ensures the model is robust to input order and the class distribution is perfectly balanced.

Algorithm 3 Pair generation

Require: Validated bucket with scores s and std σ.
1: **for** each unordered pair (i, j) **do**
2: $\Delta_{ij} = |s_i - s_j|/\sigma$.
3: **if** Δ_{ij} in sigma-window **then**
4: Emit pair (i, j) with label 1.
5: Emit swapped pair (j, i) with label 0.
6: **end if**
7: **end for**

B Additional Insights

B.1 Dataset Details

Dataset Release Notes We release our dataset under Creative Commons License,

Benchmark	Total Pairs
Code Generation On Mbpp	864
Common Sense Reasoning On Winogrande	454
Question Answering On Copa	381
Named Entity Recognition Ner On Conll 2003	372
Question Answering On Boolq	363
Question Answering On Piqa	331
Common Sense Reasoning On Arc Challenge	227
Math Word Problem Solving On Math	217
Question Answering On Squad11	203
Question Answering On Natural Questions	200
Question Answering On Squad11 Dev	198
Relation Extraction On Docred	179
Question Answering On Webquestions	176
Aspect Based Sentiment Analysis On Semeval	170
Pose Estimation On Mpii Human Pose	163
Word Sense Disambiguation On Words In Context	157
Deblurring On Gopro	155
Common Sense Reasoning On Arc Easy	142
Entity Alignment On Dbp15K Zh En	134
Common Sense Reasoning On Commonsenseqa	126

Table 3: Top 20 Benchmarks by Total Pairs (includes train and test)

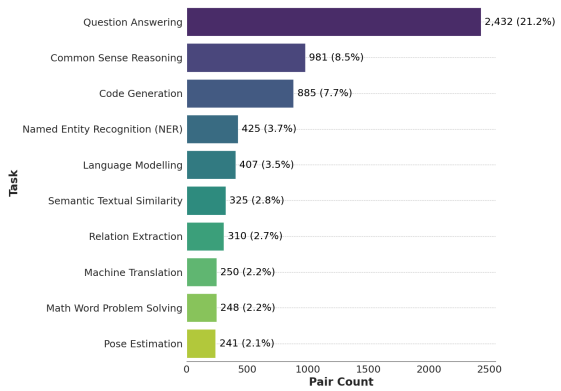

Figure 3: Top 20 Tasks based on the total pairs (includes both train and test)

to be used for research purposes only at https://anonymous.4open.science/r/Benchmark-Dataset-81B0.

Tasks and Benchmarks A benchmark in our dataset is defined as a "Task" (e.g. Question Answering) on a specific dataset (e.g. PIQA). Table 4 shows the top 20 benchmarks based on the total pairs (including train and test) in our dataset. Question Answering and Common Sense Reasoning are most common NLP tasks in our dataset. Further, Figure 3 shows the top 20 "Tasks" and the number of pairs from each of them. Question Answering is the most common task in our dataset (21.2% of total pairs), followed by Common Sense Reasoning with a drastic drop (8.5% of total pairs).

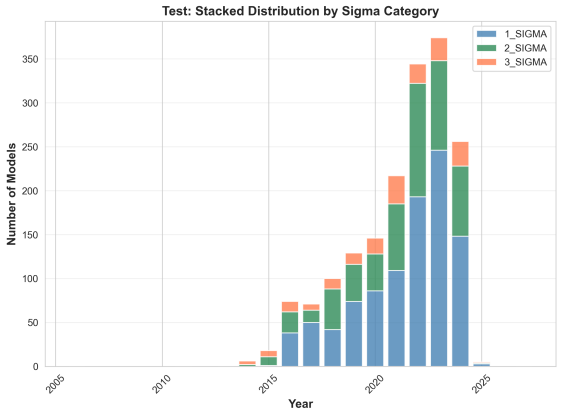

Figure 4: Distribution of the ideas/methods across the years with σ-wise breakdown in test

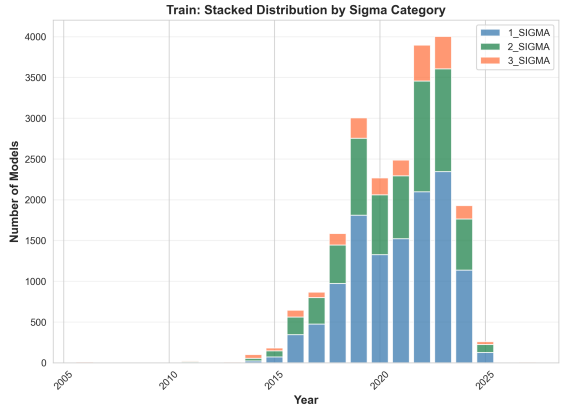

Figure 5: Distribution of the ideas/methods across the years with σ-wise breakdown in train

Temporal Distribution Figure 4 and 5 shows the temporal distribution of entires of all the leaderboards based on the year of publication of their Result Reporting Paper. The temporal distribution is uni-modal in both the train and test set, with the test set more skewed to the left.

Excluded Pairs Table 4 gives us the full statistics of the final dataset. Even after employing manually verified LLM based Research Goal Sythesis in §(4), we miss out on a large chunk of pairs (close to 30%).

Excluded Benchmark Leaderboards Apart from missing Research Goals, some of the benchmarks are excluded during the Train-Test as described in Section §(4) because of the following scenario: Consider a case where the leaderboard has only 2 entries with 2 corresponding RR papers, and due to the iterative nature of of the train-test split, if one of it has been assigned to train and the other to test (based on the splits form the

Table 4: Dataset distribution across sigma categories. Pairs classified as "Excluded" were removed due to the lack of a valid research goal.

	Train Set				Test Set			
Category	**Original**	**Excluded**	**Final**	**Augmented**	**Original**	**Excluded**	**Final**	**Augmented**
1σ	8,881	2,761	6,120	12,240	687	193	494	988
2σ	4,827	1,366	3,461	6,922	401	117	284	568
3σ	1,436	398	1,038	2,076	169	78	91	182
Total	**15,144**	**4,525**	**10,619**	**21,238**	**1,257**	**388**	**869**	**1,738**

previous leaderboards) we will be unable to form pairs within the train or test subsequently. So, this benchmark leaderboard would get skipped in the process.

B.2 Knowledge cutoffs and memorization

For Qwen3 with which we run our primary experiments, an official pretraining knowledge cutoff is not publicly documented. However, we argue that our evaluation is unlikely to be dominated by knowledge cutoff leakage for three reasons.

The prediction target is not a fact in the input text. Our labels are derived from benchmark-specific leaderboard outcomes via a unified score computed from reported metrics, including normalization and direction correction. At inference time, the model is shown only a benchmark-specific research goal and an idea description, while we explicitly remove empirical results and outcome statements from the paper text. Therefore, succeeding on our task requires mapping from a proposed methodological change to its expected empirical impact under a specific benchmark, not simply recalling a numeric result or a rank that appears verbatim in a paper.

Leakage would have to reconstruct a benchmark-conditional comparison, not a single-paper lookup. The correct answer depends on (i) the specific leaderboard and metric normalization used in our pipeline, and (ii) the relative ordering between two ideas within that benchmark. Memorizing this at scale would require storing a large number of benchmark-conditioned pairwise outcomes, rather than recalling isolated paper facts. This makes direct memorization an implausible explanation for performance gains.

Empirical evidence suggests the task is not solved by recall. Base (untrained) 8B models perform poorly (20–30% accuracy after accounting for position bias), which is substantially below chance under our consistency-based evaluation.

Results on unseen CD Test Set Exact knowledge cutoff is not available for Qwen3 models. But Llama3.1 has a knowledge cutoff of December 2023. Inference results on the test set constructed from non-NLP leaderboards, where the entries have also been filtered based on year ($\geq$ 2024), which is post the knowledge cutoff, show that fine tuned models not only generalize well, but also do well, performing either at par or better than frontier models like GPT-5 (which has a much recent knowledge cutoff of September 2024) (Table 2)

B.3 Results on Independelty Constructued Test Set

Here we present the results on the test set created by Wen et al. (2025).

Model	**Accuracy (%)**
Qwen3	
Base	2.69
Base (Reasoning)	20.06
Direct-SFT	63.43
Reason-DAPO	65.94
Synthetic-Reason-SFT-DAPO	56.46
Reason-SFT-DAPO	61.83
Reason-SFT-DrGRPO	**67.49**
Llama3.1	
Base	12.80
Base (Reasoning)	36.29
Direct-SFT	41.94
GPT-4.1 (Wen et al., w/ retrieval)	51.4

Table 5: Accuracy (%) on the Wen et al. (2025) independently constructed test set. Our fine-tuned 8B models are evaluated zero-shot (no retraining). GPT-4.1 result from Wen et al. (2025).

B.4 Ablations on CD Test

To further robustly stress-test against knowledge cut-off leakage and sensitivity to individual metric based prediction, we isolate a **2025-only subset** of 52 pairs where all papers are dated $\geq$2025, making direct memorisation implausible for all models. On this subset *Reason-SFT-DAPO* achieves **57.69%** vs. GPT-5-high at **48.07%** (+9.6 pp), and *Direct-SFT* reaches **53.85%**. Base models crater to 3–17%, confirming that fine-tuning—not memorisation—drives the performance gains.

We additionally break down the 1,410 CD pairs by whether the leaderboard rank ordering and the individual metric ordering *agree* or *disagree* (102 pairs where they conflict). Table 6 shows that *Reason-SFT-DrGRPO* achieves **60.78%** on the disagreement subset—the hardest cases where metric weighting matters most—outperforming all other models as well as GPT-5. This validates that our model is sensitive to metric-specific nuances rather than relying on simple rank heuristics.

Model	Rank=Metric (1,308)	Rank≠Metric (102)	Overall (1,410)	2025-Only (104)
Qwen3				
Base	3.67	1.96	3.55	3.85
Base (Reasoning)	13.14	5.88	12.62	17.31
Direct-SFT	46.64	33.33	45.67	53.85
Reason-DAPO	46.64	37.25	45.96	51.92
Synthetic-Reason-SFT-DAPO	41.90	29.41	41.10	44.23
Reason-SFT-DAPO	**49.08**	39.21	48.37	**57.69**
Reason-SFT-DrGRPO	48.16	**60.78**	**49.08**	53.85
Llama3.1				
Base	3.97	1.96	3.83	3.85
Base (Reasoning)	18.04	19.61	18.22	23.08
Direct-SFT	32.87	9.80	31.20	36.54
GPT-5 (zero-shot)				
Reasoning (low)	44.19	25.49	42.84	44.23
Reasoning (med)	46.18	33.33	45.25	51.92
Reasoning (high)	46.94	33.33	45.96	48.07

Table 6: Full Cross-Domain (CD) test set results. **Rank=Metric**: pairs where the leaderboard rank order agrees with the individual metric order. **Rank≠Metric**: disagreement subset. **2025-Only**: subset of 104 pairs (52 pre-augmentation) with all papers dated $\geq$2025. Consistency-aware accuracy throughout. **Bold**: best per column among trained models.

B.5 Reward Hacking

Figure 6 shows the running average of rewards and Figure 7 shows the average length of the output/response generated within the reasoning traces of a group over the train iteration of Qwen3 model being trained directly with DAPO objective and a final answer and format based reward as described in §5. As the training closes towards the 24k steps, we see the response length blow up and coupled with a drastic drop in the rewards at the individual steps. This is followed by a drastic drop in the output length. It drops down to 9 and continues generating only 9 tokens. But we see a recovery in the rewards and the models continues to obtain rewards similar to before 23k step. Thus the model transitions to learning to better predict the correct final label corresponding to the better idea in a pair, while generating only the tokens necessary for the format reward and the final reward, and fully circumventing the reasoning. A clear case of "reward hacking".

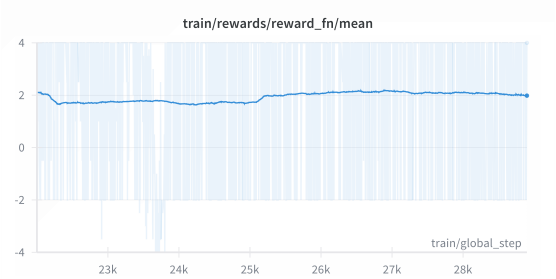

Figure 6: The average rewards through the training iterations of Reason-DAPO

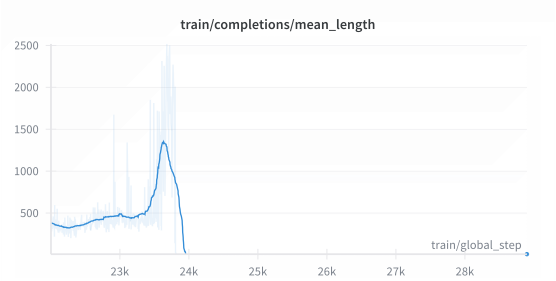

Figure 7: The mean output length through the training iterations of Reason-DAPO

B.6 Sensitivity to Benchmark-Specific Research Goals

We evaluate whether the trained model can identify the superior idea within a pair of ideas, conditionally based on the target benchmark i.e. research goal. We observe that all of the Qwen3 trained models demonstrate robust contextual awareness of benchmark specific research goals For instance, the *Efficient Audio Transformer (EAT)* achieves SOTA results on the *Audio Classification on Balanced Audio Set*, but ranks significantly lower on *Audio Classification on ESC-50*, despite a high accuracy of 96% (vs. 99.1% SOTA). We observe that Qwen3 correctly predicts EAT as the superior candidate among the pairs for the *Audio Classification on Balanced Audio Set* benchmark and inferior for *Audio Classi-*

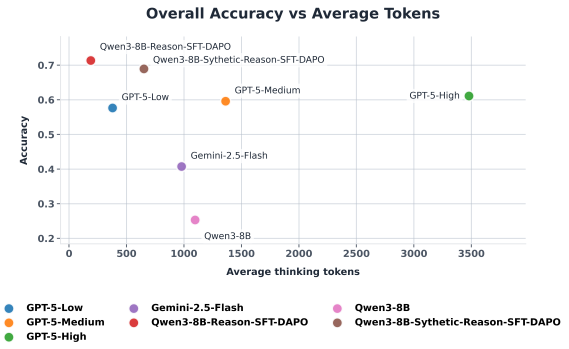

Figure 8: Overall Accuracy (%) Vs Mean Number of tokens generated during reasoning.

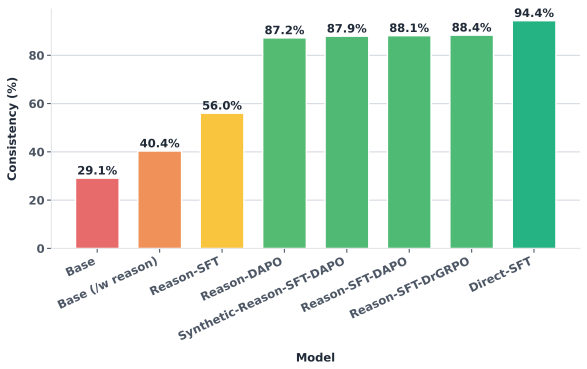

Figure 9: Consistency (%) across different stages and training paradigm of Qwen3 Model.

fication on ESC-50. This indicates that the model does not rely on superficial textual characteristics or large numerical margins. And exhibits *conditional reasoning*, correctly inferring relative utility of an idea based on a given benchmark.

B.7 Token Efficiency

Figure 8 shows the average number of reasoning tokens generated before prediction. Our RL variants that produce reasoning traces, achieve higher accuracy while using a fraction of the tokens compared to GPT-5.

B.8 Robustness analysis

We test the robustness of the trained models for some features they might be exploiting: (i) **Length:** categorize the idea pairs based on cases where longer idea is better and otherwise, (ii) **Recency:** categorize the idea pairs based on cases where the newer idea (published later) is better or worse, (iii) **Paraphrasing:** test for possible bias based on sentence structure by carefully restating the same (winning) idea using Gemini-2.5-pro (Prompt in Appendix C) in a new way in each pair.

We analyze the model robustness for these potential biases in Figure 10 based on the deviation (Δ) of accuracies on the corresponding subset and bootstrap statistical test of these accuracies.

We also plot the percentage of consistent pairs (§6.1) various models generate post inference on our test set (Figure 9). We observe that the trained models are more robust towards the position bias i.e. are not influenced by the order of presentation, with consistency exceeding 85%.

Table 8 shows a detailed breakdown of the performance, measured using accuracies on the respective subsets, of different trained Qwen3 models across different stress tests and difficulty levels of idea pair comparison.

B.8.1 Difficulty vs. Length Sensitivity

- **Reason-DAPO**: There is a strong inverse correlation between task difficulty and length sensitivity. As the difficulty increases (moving from 3-σ to 1-σ), the performance gap between "Longer" and "Shorter" inputs widens significantly.
  - 3-σ (Easy): Gap is small $\approx$ 0.4 %).
  - 2-σ (Medium): Gap increases ($\approx$ 2.5 %).
  - 1-σ (Hard): Gap maximizes ($\approx$ 19.9 %).
- **Direct-SFT**: This model shows no correlation between difficulty and length sensitivity. The performance gap between "Longer" and "Shorter" remains consistently low ($<$ 4 %) across all three sigma categories, regardless of task difficulty.
- **Reason-SFT-DAPO**: This model maintains a high length dependency across all levels, but unlike Reason-DAPO, the gap does not widen monotonically with difficulty; it remains large ($\approx$ 14-19 points) in both the easiest (3-σ) and hardest (1-σ) tiers.

B.8.2 Difficulty vs. Recency Variance

- **Reason-DAPO (Inversion Effect)**: The model's preference for "Newer" vs "Older" data inverts based on difficulty.
  - In 1-σ (Hard), it scores higher on "Newer" data (+12.1 %).
  - In 2-σ (Medium), it scores higher on "Older" data (-7.6 %).
  - In 3-σ (Easy), it scores slightly higher on "Older" data (-1.1 %).

Table 7: Distribution of dataset preferences in percentages in Train. The **Total** column indicates the number of samples, while other columns show the percentage breakdown of Recency and Length preferences within each split.

		Recency (%)			Length (%)		
Data Split	**Total**	**Newer**	**Older**	**Same**	**Shorter**	**Longer**	**Equal**
Full Train Set	19,113	57.08	17.73	25.19	47.97	51.83	0.20
Breakdown by Sigma (σ)							
$\sigma = 1$	11,016	52.35	19.64	28.01	48.04	51.82	0.15
$\sigma = 2$	6,229	62.93	15.49	21.58	48.40	51.31	0.29
$\sigma = 3$	1,868	65.42	13.97	20.61	46.09	53.69	0.21

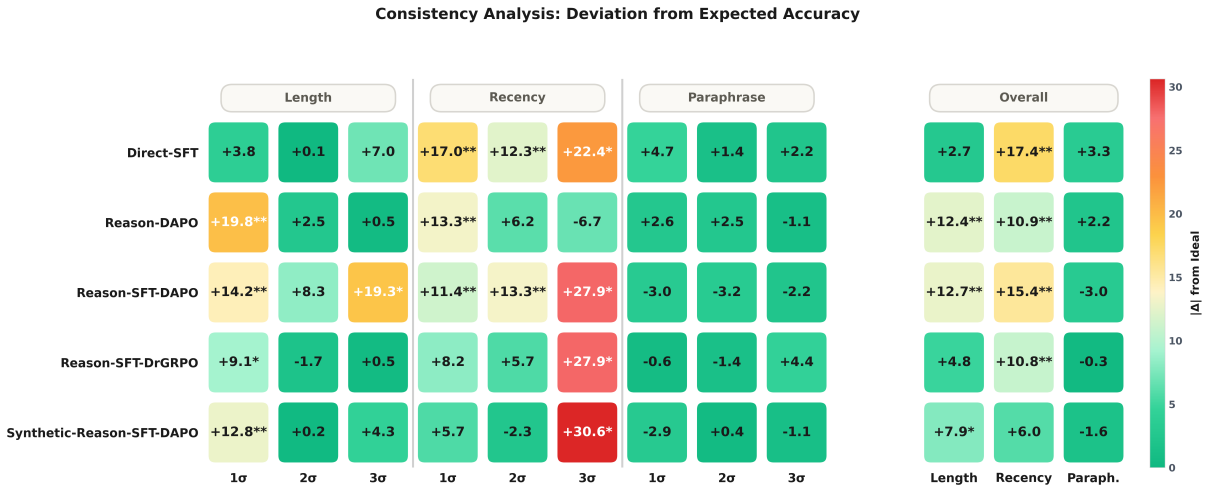

Figure 10: Differential Analysis (Δ based) with Bootstrap statistical significance tests across different Difficulty Subsets (σ) and Overall Performance. **: $p < 0.01$; *: $p < 0.05$.

- **Direct-SFT**: The "Recency Gap" (Newer minus Older) exists across all difficulties but is non-linear. The model is most robust to recency shifts in the 2-σ category (gap of 6.9 %) compared to the 1-σ (17.3 %) and 3-σ (30.7 %) categories.

B.8.3 Sample Size (N) vs. Accuracy's Stability

- **Variance Correlation**: There is a direct observable correlation between the sample size N and the variance of scores within a model's row.
  - High N (1-σ): Accuracies across splits (Recency/Length) are generally clustered. For example, **Reason-SFT-DrGRPO** has accuracies tightly between 56.8% and 70.9%.
  - Low N (3-σ): Accuracies exhibit extreme volatility. In this tier, **Reason-SFT-DrGRPO** spans from 41.7% to 100.0% accuracies.
- **Paraphrasing Stability**: The impact of paraphrasing is relatively uniform across difficulty levels (N counts are high for both Original and Para). For instance, **Reason-DAPO**'s gain from paraphrasing remains steady at roughly +2.5 to +2.7 % in both 1-σ and 2-σ categories.

B.8.4 Length Dominance across Models

- In the **1-σ (Hard)** category, the hierarchy of models changes depending on the input length.
  - On Longer Inputs: **Reason-DAPO** (79.1%) > **Direct-SFT** (72.9%).
  - On Shorter Inputs: **Direct-SFT** (69.1%) > **Reason-DAPO** (59.2%).

  This indicates that in the hardest difficulty tier, the comparative advantage of the RL

Table 8: Detailed Robustness Statistics: Accuracy (%) and Total Sample Count (N). The sample counts for each category (e.g., Longer, Newer) remain same across models for the same σ-subset.

Model	Length		Recency			Paraphrasing	
	Longer	Shorter	Newer	Older	Same	Original	Para.
1-σ (N)	**516**	**466**	**480**	**212**	**296**	**988**	**988**
Direct-SFT	72.9	69.1	79.6	62.3	62.8	70.9	75.5
Reason-DAPO	79.1	59.2	76.3	64.2	62.2	69.4	72.1
Reason-SFT-DAPO	71.3	57.1	70.4	50.9	64.9	64.6	61.5
Reason-SFT-DrGRPO	70.9	61.8	70.4	69.8	56.8	66.2	65.6
Synthetic-Reason-SFT-DAPO	72.1	59.3	68.8	74.5	54.7	65.8	62.9
2-σ (N)	**278**	**290**	**338**	**134**	**96**	**568**	**568**
Direct-SFT	85.6	85.5	90.5	83.6	70.8	85.6	87.0
Reason-DAPO	76.3	73.8	77.5	85.1	52.1	75.0	77.5
Reason-SFT-DAPO	83.5	75.2	84.6	76.1	64.6	79.2	76.1
Reason-SFT-DrGRPO	75.5	77.2	78.7	83.6	58.3	76.4	75.0
Synthetic-Reason-SFT-DAPO	72.6	72.5	71.6	88.1	54.2	72.5	72.9
3-σ (N)	**108**	**70**	**146**	**24**	**12**	**182**	**182**
Direct-SFT	87.0	80.0	89.0	58.3	83.3	84.6	86.8
Reason-DAPO	83.3	82.9	82.2	83.3	100.0	83.5	82.4
Reason-SFT-DAPO	90.7	71.4	89.0	58.3	66.7	83.5	81.3
Reason-SFT-DrGRPO	83.3	82.9	89.0	41.7	100.0	83.5	87.9
Synthetic-Reason-SFT-DAPO	76.0	71.7	80.8	66.7	16.7	74.7	73.6
Overall (N)	**902**	**826**	**964**	**370**	**404**	**1738**	**1738**
Direct-SFT	78.5	75.8	84.9	69.7	65.4	77.1	80.4
Reason-DAPO	78.7	66.3	77.6	73.0	60.9	72.7	74.9
Reason-SFT-DAPO	77.4	64.7	78.2	60.5	64.9	71.4	68.4
Reason-SFT-DrGRPO	73.8	69.0	76.1	73.0	58.4	71.4	71.0
Synthetic-Reason-SFT-DAPO	72.7	64.9	71.6	78.9	53.5	68.9	67.3

model is conditional on the presence of longer context.

B.8.5 Distribution of train Dataset

We further analyse the distribution of the idea pairs based on Recency and Length as described in §B.8 (Table 7).

We see that even within the different σ-Categories, the distribution of shorter and longer is relatively balanced. It also contains a very small fraction of pairs that have equal length (The test set doesn't have any). This likely enabled the models to learn much better without introducing such bias due to imbalance. The distribution is largely imbalanced under the newer, older and same year (§B.8) categories, with the number the newer category having almost 2 times the number of pairs in older and same year combined. Older category has consistently lower representation across all σ. This could additionally explain slightly large variations across different trained models, apart from the largely imbalanced distribution within the test itself.

B.8.6 Statistical Testing

To rigorously assess whether observed bias patterns are statistically meaningful rather than noise, a non-parametric bootstrap procedure is employed. For each robustness dimension, accuracy is computed separately on two complementary subsets (for example, "longer is better" vs. "shorter is better") and the null hypothesis $H_0 \colon \Delta = 0$ (no bias) is tested against the two-sided alternative $H_1 \colon \Delta \neq 0$.

Concretely, $B = 10{,}000$ bootstrap resamples are drawn with replacement from the test set, and the accuracy on each subset is recomputed per resample. For each resample, the delta $\Delta^* = \mathrm{acc}_A^* - \mathrm{acc}_B^*$ between the two subsets is recorded, forming the bootstrap distribution $\{\Delta_1^*, \ldots, \Delta_B^*\}$. The 95% percentile confidence interval is:

$$\mathrm{CI}_{95\%} = \big[P_{2.5}(\Delta^*),\ P_{97.5}(\Delta^*)\big], \tag{5}$$

and the two-sided p-value is:

$$p = \begin{cases} \min\big(2 \cdot P(\Delta^* < 0),\ 1\big) & \text{if } \bar{\Delta} \geq 0, \\ \min\big(2 \cdot P(\Delta^* > 0),\ 1\big) & \text{if } \bar{\Delta} < 0. \end{cases} \tag{6}$$

H_0 is rejected at $p < 0.05$ (reported as *) and $p < 0.01$ (reported as **). Resampling is performed at the level of paired units (original + swapped pair) rather than individual samples, preserving the dependency structure of the consistency-based evaluation metric. A non-parametric bootstrap is preferred over parametric alternatives such as McNemar's test because it makes no distributional assumptions about the accuracy difference and extends straightforwardly to the composite consistent accuracy statistic, which does not have a closed-form null distribution.

Paraphrasing robustness. No model shows a statistically significant change in accuracy due to paraphrasing at *any* difficulty tier ($p > 0.05$ for all models across 1-σ, 2-σ, 3-σ, and Overall). This confirms that the observed minor fluctuations are noise rather than genuine structural sensitivity to surface form—our models' decisions are anchored to the semantics of the idea, not the specific phrasing.

Length bias. Results are model-specific rather than universal. *Direct-SFT* shows *no* statistically significant length bias, confirming its robustness to idea length variation. In contrast, *Reason-DAPO* and *Reason-SFT-DAPO* exhibit highly significant length preferences, indicating these RL-trained models systematically favour longer ideas. *Reason-SFT-DrGRPO* sits at the boundary: borderline non-significant at the overall level ($p = 0.116$) but significant at the hardest 1-σ tier ($p = 0.035$). These findings suggest that RL training with purely binary rewards may inadvertently reinforce surface-level length heuristics, whereas supervised fine-tuning on balanced data is more effective at preventing this.

Recency bias. All models show statistically significant recency bias at the overall level ($p < 0.01$) except *Synthetic-Reason-SFT-DAPO*. Crucially, however, we argue this is not a spurious artefact: in competitive NLP benchmarking, newer methods genuinely tend to outperform older ones. Rather than indicating a bias the model needs to overcome, the recency signal reflects a learnable empirical prior about scientific progress. This interpretation is supported by the fact that even the frontier zero-shot models pick up on this signal; it is a feature of the task domain.

B.9 Few-Shot Ablation with GPT-5

To test whether in-context learning can substitute for task-specific fine-tuning, we evaluate GPT-5 in a 3-shot setting. For each test pair, we prepend 3 demonstration examples—one from each difficulty category (1σ, 2σ, 3σ)—as in-context examples before asking for the final prediction. The same 3 examples are used for all test pairs. This ablation directly addresses whether the performance gap between our fine-tuned 8B models and GPT-5 can be closed by providing GPT-5 with task demonstrations.

As shown in Table 10, few-shot examples do not meaningfully close the performance gap. While the low-reasoning setting shows a marginal improvement ($+0.24$ points), both medium and high reasoning levels *degrade* (Med: -0.58, High: -0.69 points). This suggests that the comparative empirical forecasting task cannot be solved through in-context pattern matching alone; the performance gap between GPT-5 and our fine-tuned models reflects a genuine difference in internalised task priors that arises from optimising on thousands of labelled pairs, rather than a deficiency addressable by a few demonstrations.

B.10 Conditional Accuracies

We define conditional accuracy as the accuracy of predictions within all the consistent pairs i.e. we consider only the subset of total pairs that show no position bias to compute the accuracy§6.1. Ta-

Table 9: Bootstrapped bias significance tests ($B = 10{,}000$). Δ in percentage points with 95% CIs. **: $p < 0.01$; *: $p < 0.05$.

Model	Tier	Length Δ [95% CI]	Recency Δ [95% CI]	Paraphrase Δ [95% CI]
Direct-SFT	Overall	$+2.7$ [-2.8, 8.3]	$+17.5^{**}$ [11.8, 23.2]	$+3.3$ [-0.6, 7.1]
	1-σ	$+3.8$ [-4.4, 12.0]	$+17.1^{**}$ [9.3, 24.8]	$+4.6$ [-0.8, 10.1]
	2-σ	$+0.1$ [-8.3, 8.5]	$+12.3^{**}$ [3.7, 21.3]	$+1.4$ [-4.2, 7.0]
	3-σ	$+7.2$ [-8.1, 23.2]	$+22.5^{*}$ [0.2, 46.0]	$+2.2$ [-7.7, 12.1]
Reason-DAPO	Overall	$+12.3^{**}$ [6.5, 18.2]	$+10.9^{**}$ [4.9, 16.9]	$+2.2$ [-2.0, 6.3]
	1-σ	$+19.8^{**}$ [11.7, 27.7]	$+13.3^{**}$ [5.2, 21.2]	$+2.6$ [-3.0, 8.3]
	2-σ	$+2.5$ [-7.5, 12.4]	$+6.2$ [-4.1, 16.7]	$+2.5$ [-4.2, 9.5]
	3-σ	$+0.3$ [-15.5, 16.6]	-6.8 [-21.9, 11.3]	-1.1 [-12.1, 9.9]
Reason-SFT-DAPO	Overall	$+12.7^{**}$ [6.5, 18.8]	$+15.4^{**}$ [9.2, 21.5]	-3.0 [-7.2, 1.4]
	1-σ	$+14.2^{**}$ [5.8, 22.7]	$+11.3^{**}$ [3.1, 19.8]	-3.1 [-9.1, 2.8]
	2-σ	$+8.3$ [-0.9, 18.0]	$+13.3^{**}$ [3.4, 23.2]	-3.2 [-9.9, 3.5]
	3-σ	$+19.2^{*}$ [2.3, 36.3]	$+27.8^{*}$ [4.3, 51.5]	-2.2 [-13.2, 8.8]
Reason-SFT-DrGRPO	Overall	$+4.8$ [-1.2, 10.8]	$+10.7^{**}$ [4.7, 16.8]	-0.3 [-4.5, 3.9]
	1-σ	$+9.1^{*}$ [0.7, 17.5]	$+8.2$ [-0.1, 16.5]	-0.7 [-6.7, 5.3]
	2-σ	-1.8 [-11.7, 8.2]	$+5.7$ [-4.3, 16.1]	-1.4 [-8.5, 5.6]
	3-σ	$+0.4$ [-15.5, 16.5]	$+27.8^{*}$ [4.4, 51.5]	$+4.4$ [-5.5, 14.3]
Synthetic-Reason-SFT-DAPO	Overall	$+7.9^{*}$ [1.7, 14.0]	$+6.0$ [-0.3, 12.3]	-1.6 [-6.1, 2.8]
	1-σ	$+12.8^{**}$ [4.3, 21.2]	$+5.7$ [-2.7, 14.0]	-2.9 [-8.7, 3.0]
	2-σ	$+0.2$ [-10.2, 10.7]	-2.3 [-13.1, 8.2]	$+0.4$ [-7.0, 7.7]
	3-σ	$+4.3$ [-14.3, 23.3]	$+30.6^{*}$ [5.8, 55.7]	-1.1 [-14.3, 12.1]

Model	Zero-shot	3-shot
GPT-5 Reasoning (Low)	57.65	57.89
GPT-5 Reasoning (Med)	59.61	59.03
GPT-5 Reasoning (High)	61.10	60.41

Table 10: Zero-shot vs. 3-shot GPT-5 accuracy (%) on our in-domain test set. Few-shot examples marginally improve low-reasoning performance but slightly degrade medium and high.

ble 11 reports the conditional accuracy of various models. The number in brackets denote the total number of consistent pairs. The total here represents the augmented set.

B.11 Idea Ranking

We analyze how well the trained models do when used for idea ranking on both in-domain (ID) (from the train-test split) and cross-domain (CD) test set. We pick all leaderboards/research goals that have at least 3 unique entries/ideas in them. We do $\binom{n}{2}$ many comparison. We rank the ideas based on the number of times each idea wins when compared to all others. Ideas with ties are given the same rank. Comparisons that have inconsistent predictions are dropped and not considered. We assess the quality of ranking using Top-1 Accuracy and Root Mean Square Error (RMSE) of the true and predicted ranks.

Model / Method	1-σ (988)	2-σ (568)	3-σ (182)	Overall (1738)
Qwen3				
Base	69.47 (262)	72.55 (204)	50.00 (40)	69.17 (506)
Base (Reasoning)	63.86 (404)	64.71 (238)	46.67 (60)	62.68 (702)
Direct-SFT	75.92 (922)	88.69(548)	90.59 (170)	81.71 (1640)
Reason-DAPO	80.90 (848)	85.54 (498)	86.36 (176)	83.05 (1522)
Reason-SFT-DAPO	75.59 (844)	87.55 (514)	87.36 (174)	80.94 (1532)
Reason-SFT-DrGRPO	76.58 (854)	86.11 (504)	85.39 (178)	80.73 (1536)
Llama3.1				
Base	52.87 (522)	64.96 (274)	64.15 (106)	57.87 (902)
Base (Reasoning)	65.17 (402)	64.93 (268)	68.97 (58)	65.38 (728)
Direct-SFT	78.17 (678)	84.62 (390)	81.33 (150)	80.62 (1218)
GPT-5				
Reasoning (low)	67.13 (864)	67.48 (492)	56.25 (160)	66.09 (1516)
Reasoning (med)	69.03 (846)	70.00 (500)	62.96 (162)	68.70 (1508)
Reasoning (high)	70.67 (866)	68.24 (510)	60.00 (170)	68.69 (1546)
Gemini 2.5 Flash				
Base (Reasoning)	72.13 (574)	66.67 (342)	55.93 (118)	68.47 (1034)

Table 11: Conditional Accuracy (%) breakdown across different σ categories. Values in parentheses denote the total number of samples (N) for that category that were consistent. They are no directly comparable since N changes.

Model	In-Domain (ID)			Cross-Domain (CD)		
	Con. (%) ↑	Top-1 (%) ↑	RMSE ↓	Con. (%) ↑	Top-1 (%) ↑	RMSE ↓
Qwen3						
Base	29.55	40.00	1.87	5.35	38.46	2.45
Base (Reason)	40.09	31.43	1.73	19.34	28.21	1.87
Direct-SFT	**90.71**	44.76	1.22	79.84	31.82	1.96
Reason-SFT-DAPO	83.96	42.86	**1.12**	77.37	28.89	1.73
Reason-DAPO	84.85	**51.43**	1.29	76.95	33.33	1.83
Reason-SFT-DrGRPO	87.72	50.48	**1.12**	80.25	41.30	**1.65**
Synthetic-Reason-SFT-DAPO	85.27	43.81	1.32	71.74	36.36	1.80
GPT-5						
Low	85.82	38.10	1.48	77.50	36.96	1.76
Medium	86.88	36.19	1.41	81.48	34.78	1.78
High	85.77	35.24	1.41	**82.99**	**43.48**	1.77

Table 12: Performance comparison on In-Domain (ID) and Cross-Domain (CD) test sets. Metrics reported are Overall Consistency Rate (Con.), Top-1 Accuracy, and Median RMSE. **Bold**: Best, Underline: Second Best within each domain. (↓) lower is better; (↑) higher is better.

- We observe that the untrained base models of *Qwen3*, have a high position bias, thus have a poor overall consistency rate (Table 12). Meanwhile, the fine tuned models show drastic improvement, with consistency almost at par (or better in case of In-Domain test) with GPT-5.
- *Reason-SFT-DrGRPO* achieves better Top-1 accuracy than GPT-5 under low and medium reasoning effort on the CD test. All of the trained models achieve better Top-1 accuracy than GPT-5 on the ID test.
- Before comparing models for their RMSE, it is important to keep in mind that though median is a more robust statistic which is not affected by outliers, it depends highly on the number of samples one has. To make meaningful comparisons, one should have at least similar sample size. Similar consistency rates allow for fair comparisons. For example if we had 10 leaderboards with each having 4 entries and 0% consistency, all the entries within each leaderboard would be assigned a rank of 1. So all of them would have an RMSE of 1.87, which in turn would also be the median. This is lower than what you see for Direct-SFT even when you have no meaningful comparisons or predictions anywhere! This example would also have a 100% Top-1 accuracy! To prevent this we only use Top-1 accuracy from leaderboards that have at least 2 different ranks predicted. This might still not be enough since we see high Top-1 accuracy for base models when they actually don't do very well.
- *Reason-SFT-DAPO* and *Reason-SFT-DrGRPO* achieve better RMSE than GPT-5 (across all reasoning efforts) on CD test, hence showing the potential of such models in filtration and idea re-ranking. All trained models achieve better RMSE compared to GPT-5 on the ID test.
- Figure 11 & 13 and 12 & 14 show the distribution of the consistency and RMSE across different leaderboards/research goals for ID and CD test respectively. The highest outlier of Direct-SFT and Reason-SFT-DrGRPO is lower than that of GPT-5, while Reason-DAPO and Reason-SFT-DAPO have comparable RMSE on CD test.
- The consistency distribution is left-skewed for all cases except for the base models, with most values at high consistency. And the RMSE distribution is right-skewed in all almost all cases. The skewness is greater on ID test compared to CD test, but has more number of outliers in general as well.
- *Reason-SFT-DrGRPO* shows least drastic change in all three metrics for ranking compared to other models showing that it has learnt to generalize better than the others.

These observations show that fine-tuned 8B

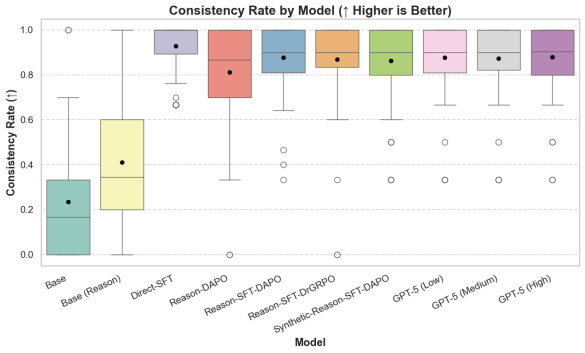

Figure 11: Distribution of consistency rate (%) across different research goals/leaderboards for the cross-domain test set

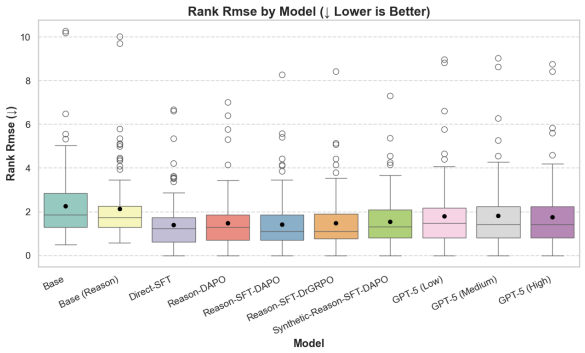

Figure 12: Distribution of RMSE across different research goals/leaderboards for the cross-domain test set

models that are trained for comparative empirical forecasting could be used as an idea re-ranker, performing better than frontier models under zero-shot setting.

B.12 Probabilistic Calibration

Three metrics are used to quantify calibration.

Brier Score (lower is better) measures the mean squared probability error:

$$\mathrm{BS} = \frac{1}{N}\sum_{i=1}^{N}(p_i - y_i)^2, \tag{7}$$

where p_i is the model's predicted probability for the correct class and $y_i \in \{0, 1\}$ is the ground-truth label. The Brier Score is a proper scoring rule that penalizes both overconfidence and underconfidence.

Expected Calibration Error (ECE) measures the mean absolute gap between binned confidence and binned accuracy. Predictions are partitioned into $B = 10$ equal-width bins based on confidence, and the calibration error is computed as:

$$\mathrm{ECE} = \sum_{b=1}^{B}\frac{|\mathcal{B}_b|}{N}\cdot\left|\mathrm{acc}(\mathcal{B}_b) - \mathrm{conf}(\mathcal{B}_b)\right|, \tag{8}$$

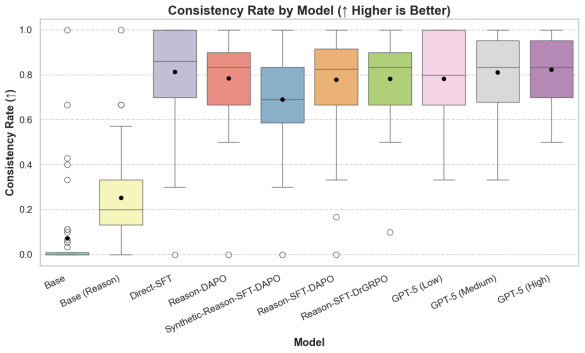

Figure 13: Distribution of consistency rate (%) across different research goals/leaderboards for the in-domain test set

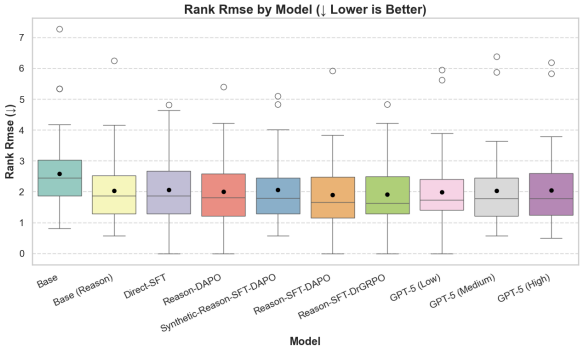

Figure 14: Distribution of RMSE across different research goals/leaderboards for the in-domain test set

where $\mathrm{acc}(\mathcal{B}_b)$ and $\mathrm{conf}(\mathcal{B}_b)$ are the mean accuracy and mean confidence of predictions in bin b, and $|\mathcal{B}_b|$ is the bin count. ECE is a binning approximation to the full calibration integral, weighted by the empirical distribution of confidence values.

Maximum Calibration Error (MCE) captures the worst-case miscalibration across all bins:

$$\mathrm{MCE} = \max_{b:\,|\mathcal{B}_b|>0}\left|\mathrm{acc}(\mathcal{B}_b) - \mathrm{conf}(\mathcal{B}_b)\right|. \tag{9}$$

Because the primary evaluation metric filters to consistent predictions (§6.1), we report the **debiased** variant: each sample is paired with its swapped counterpart and their probabilities are averaged in a common reference frame to obtain a position-debiased confidence:

$$\tilde{p}_i = \frac{p_i^{\mathrm{orig}} + (1 - p_i^{\mathrm{swap}})}{2}, \tag{10}$$

where p_i^{orig} is the model's $P(\mathrm{class} = 1)$ in the original ordering and p_i^{swap} is the corresponding probability from the swapped ordering, inverted to align reference frames.

Using these calibration metrics under the *debiased* evaluation regimes, we assess the confidence quality of our trained models.

Family	Model	DB ↓	DE ↓	DM ↓
Qwen	Base	0.2507	**0.0937**	0.2832
	Base (Reason)	0.2635	0.1392	0.3276
	Direct-SFT	**0.1676**	0.1390	**0.1915**
	Reason-DAPO	0.1755	0.1584	0.5475
	Reason-SFT-DAPO	0.1941	0.1688	0.3112
	Reason-SFT-DrGRPO	0.1970	0.1734	0.3641
	Synthetic-Reason-SFT-DAPO	0.2186	0.2009	0.4320
Llama	Base	0.2685	0.1552	0.3275
	Base (Reason)	0.2882	0.1931	0.3373
	Direct-SFT	**0.2129**	**0.1205**	**0.2184**

Table 13: Calibration metrics on the in-domain test set. **DB**: Debiased Brier, **DE**: Debiased ECE, **DM**: Debiased MCE. **Bold**: best within family, Underline: second best within family.

Fine-tuning substantially improves Brier scores across both model families: base models score $\approx 0.25-26$ whereas fine-tuned models achieve 0.16–0.21, representing a reduction of roughly 30–40%. *Qwen3 Direct-SFT* achieves the best overall calibration, with Consistent Brier of **0.1676** and second best Consistent ECE of **0.1463**. This means that on stable, position-invariant predictions, the model's confidence levels are well-aligned with its accuracy. Additionally it is important to compare only between the non-reasoning and reasoning models respectively. This is because the non reasoning models output only 0/1 where as reasoning models generate a certain lenght of tokens before making the prediction which makes the prediction conditional on the tokens generated before making the prediction.

The RL-tuned variants (Reason-DAPO, Reason-SFT-DAPO, Reason-SFT-DrGRPO) show higher MCE values (0.31–0.87 consistent), indicating that while their average calibration is reasonable, their worst-case confidence bins are poorly calibrated. This may reflect that RL training with a binary reward signal encourages the model to be more decisive (higher entropy collapse) at the cost of miscalibration in low-confidence regions.

Our results largely reflect the findings of recent literature demonstrating that overconfidence and miscalibration are pervasive when utilizing raw logit probabilities as direct proxies for confidence estimation in large language models. As systematically outlined by Geng et al. (2024), generative LLMs inherently exhibit overconfidence, and relying on extracted token logits often fails to account for the semantic variability of language, ultimately yielding unreliable and uncalibrated confidence estimates. Furthermore, our observations regarding the inadequacy of logit-based probabilities are empirically supported by recent work from Zhou et al. (2025). In their evaluation of uncertainty quantification methods, they demonstrate that complex metrics derived directly from token logits frequently yield poorly calibrated confidence scores and are systematically outperformed by simpler strategies such as verbalized direct prompting. Consequently, these findings reinforce that utilizing logit probabilities directly is a suboptimal approach for reliable confidence estimation, frequently leaving models highly uncalibrated under such settings.

C Prompts

Idea Extraction

```
Extract the fundamental scientific idea or contribution, methodology and goal
    from this research. Focus exclusively on **what** """ + model_name + """
    proposes and **how** it works mechanistically, IGNORING all empirical
    evaluations, comparisons, limitations, and benefits.
   (Note that sometime the model or the methods name may be Authors name,
       authorname with year, short forms, their method applied on or over
       existing method or model, humans performance, names of certain teams,
       simply "ensemble" meaning combination of methods or models, or model or
       method name along with certain implementation specifics (like zero-shot
       etc.). In some cases the full text might not explicitly metion this or
       could be a noisy version but try to correlate.)

Paper Information:
Title: """ + proposal_title + """
Context: """ + papers_context + """
Model Introduced in Associated Paper: """ + introduced_info + """
Full Text: """ + full_content + """

Specific Requirements

INCLUDE:
- Core scientific or technical contribution of """ + model_name + """: The main
     idea, algorithm, or method proposed by """ + model_name + """
- Detailed mechanism of """ + model_name + """: Step-by-step explanation of how
     """ + model_name + """ operates
- Theoretical foundations of """ + model_name + """: Underlying principles,
    assumptions, and theoretical justifications of """ + model_name + """
- Technical innovations of """ + model_name + """: Novel components,
    modifications, or adaptations introduced.
- Implementation specifics of """ + model_name + """: Key technical details
    necessary for understanding """ + model_name + """
- Parameter definitions in """ + model_name + """: What each variable,
    hyperparameter, and component represents in """ + model_name + """
- Computational processes in """ + model_name + """: How information flows
    through """ + model_name + """
And any other details that are essential to understand """ + model_name + """.

INCLUDE ONLY IF PRESENT IN FULL TEXT:
- Mathematical formulations of """ + model_name + """: All equations, formal
    definitions, and mathematical relationships etc. specific to """ +
    model_name + """
- Architectural details of """ + model_name + """: Model components, layers,
    connections, transformations etc. specific to """ + model_name + """
- Algorithmic procedures of """ + model_name + """: Training procedures,
    inference steps, optimization methods etc. specific to """ + model_name +
    """


EXCLUDE:
- Empirical results, performance metrics, accuracy scores
- Comparisons or improvements over baseline methods or previous work
- Experimental setup, evaluation protocols
- Advantages, benefits, improvements over existing methods
- Limitations, drawbacks, or failure cases
- Future work suggestions or applications
- Background literature review or related work discussion
- Names given to the model or method or approach

Format your response as a JSON object:
{
   "paper_id": "Short identifier based on title",
   "title": \""" + proposal_title + """\",
   "core_idea": "A comprehensive, cohesive scientific exposition of """ +
       model_name + """'s fundamental idea presented as a natural flowing
```

```
      paragraph about """ + model_name + """, without the mention of """ +
      model_name + """."
}

CRITICAL REQUIREMENTS:
- Present the extracted idea as ONE comprehensive scientific passage in natural
     flowing prose about """ + model_name + """ as if it is a new idea being
    proposed.
- Present the core idea of """ + model_name + """ as though (or in the tone of)
     a new idea is being proposed instead of summarising existing work.
- Create a cohesive technical exposition that reads like a thorough scientific
    description of """ + model_name + """
- Focus on mechanism and methodology of """ + model_name + """, NOT performance
     or comparisons or improvements
- Extract ideas and research goal objectively, grounded in scientific language
    and full text content only. Ensure you pay attention to all the relevant
    details in the full text about """ + model_name + """
- Present the research goal from the benchmark paper context as a separate,
    detailed paragraph that fully explains the research motivation
- DO NOT INCLUDE model or method name in the core_idea.

Examples of good extractions:
""" + demos + """

Now extract the core research ideas specific to """ + model_name + """ from the
     provided research content:
```

Prompt for Paraphrasing the Idea

```
Restate the following research idea in a different way while preserving ALL
    technical details exactly.
Use different sentence structures and word choices, but do not add, remove, or
    change any information.

**Original:**
{idea}

Return ONLY a JSON object (no markdown):
{{"restated_idea": "..."}}
```

Reasoning Extraction

```
You are analyzing a benchmark paper to extract explicit comparative reasoning
    between specific models/methods. Your task is to identify and extract the
    actual reasoning from the paper that explains WHY one model/method performs
    better than another, using only information explicitly present in the paper.
     Present the reasoning in the form of Chain of Thoughts, along which a
    strong reasoning model would think, analyse and deduce, step-by-step which
    of the given model/method is better than the other, and WHY (grounded in the
     actual truths of the paper). Essentially you are producing a deatiled
    reasoning mimicking an actual human expert's thought process (by grounding
    it in the actual truths of the paper) and presenting it as if it were your
    own reasoning.
CRITICAL CONSTRAINT: Only consider comparisons between the models/methods
    listed below. Ignore all other models.

Benchmark Paper: {benchmark_paper_title}
Models/Methods to Analyze: {models_list}

Paper Content:
{benchmark_content}

Your Task:

The paper may compare multiple models/methods from the list above. For each
    pair where explicit comparative reasoning exists, extract that reasoning.
    Only extract pairs where the paper provides actual reasoning about why one
    is better than or differs from the other.

What to Extract:

Extract a single, cohesive reasoning statement (form of Chain of Thoughts,
    along which a strong reasoning model would think, analyse and deduce, step-
    by-step which of the given model/method is better than the other, and WHY (
    grounded in the actual truths of the paper)) that captures ALL the reasoning
     the paper provides about why idea_A is better or differs from idea_B or
    visa versa. (Essentially you are producing a deatiled reasoning mimicking an
     actual human expert's thought process (by grounding it in the actual truths
     of the paper) and presenting it as if it were your own reasoning.) This
    should be comprehensive and include any and all explanations the paper gives
    . The reasoning could involve:

- Any differences in approach, design, or methodology
- Any factors that contribute to superiority or differences
- Any explanations for why one works better than the other
- Any limitations overcome or advantages gained
- Any combination of factors mentioned in the paper

Be inclusive: Capture all reasoning provided by the paper, even if it doesn't
    fit into traditional categories. You may include a hypothesis as a statment
    if experminetal results support/prove it.

Critical Requirements:

- Use idea_A and idea_B notation: Replace actual model/method names with "
    idea_A" and "idea_B" in your reasoning
- Single cohesive reasoning: Combine all aspects into ONE comprehensive
    reasoning statement per pair
- Ground in paper content: Extract actual reasoning from the paper, not your
    interpretation
- First-person style: Present reasoning as direct statements, not "the paper
    says..." or "according to the paper..."
- Only pairs with reasoning about superiority: Only include model pairs where
    the paper explains WHY one is better, not just WHAT the differences are
- Only specified models: Ignore comparisons with any models not in the list: {
    models_list}

What to EXCLUDE:
```

```
ONLY exclude the following:
- Pure statements about empirical performance (e.g., "idea_A achieves 95\%
    accuracy while idea_B achieves 90\%") or any details of empirical results.
- Simple restatements that one is better based solely on numbers or metrics
    without any explanation of WHY
- Mere descriptions of differences without explanation of superiority (e.g., "
    idea_A uses attention while idea_B uses convolutions" - this is just a
    difference, not reasoning about why one is better)
- Experimental setup details without reasoning or conclusions obtained after
    running the experiments.
- General background information
- Future work or speculation
- Comparisons with models NOT in the specified list
- Mention of the paper anywhere in your reasoning (e.g., avoid phrases like "
    the paper states..." or "according to the authors...")
- Any direct quotes/texts or refrences to the paper
- Conclusion or inference drawn from looking at the performance. There should
    be NO mention of performance based on the experimental results.

KEY DISTINCTION: The paper must explain WHY the difference leads to one being
    better, not just WHAT the differences are.

Example of what NOT to extract:
- "idea_A uses transformers while idea_B uses LSTMs" (just a difference)
- "The user is asking me to reason why Idea A is better than Idea B......." (
    starting by already assuming or declaring which of the ideas is better, and
    you have to just figure out why is this the case)
- ".........The paper specifies that idea_A, the hybrid model, 'simply retains
     two self-attention layers' in addition to its H3 layers.\n2. Next, we
    evaluate the impact of this difference. The paper suggests that this
    architectural choice is not arbitrary but is designed to leverage the '
    complementary strengths of SSMs and attention.'\n3....." (Mention of
    paper explicitly)
- "To determine which model is better, let's analyze their core design and
    resulting performance characteristics based on the paper's findings......."
    (Explicitly mentions the paper)
- "Despite these significant advantages in size, data, and training, idea_B
    only achieves an 8-point performance gain over idea_A......." (mentions
    performance numbers)

Example of what TO extract:
- "idea_A uses transformers which can capture long-range dependencies more
    effectively than the LSTM architecture in idea_B, allowing it to handle
    complex contexts that idea_B struggles with" (explains why the difference
    matters)
- "We want to compare idea_A and idea_B to figure out which of the two is
    better. Let us break this down step by step......." (Starts in a way that
    does NOT already declare one of the ideas being better than the other and as
     if the real task is simply supposed to reason out why this is the case)
- ".........idea_A, the hybrid model, retains two self attention layers in
    addition to its H3 layers. The architecture design leverages complmentary
    strengths of SSMs and attentions........." (no mention of paper explicitly,
     and captures the reasoning about WHY this design choice matters)

Everything else should be included - if the paper provides ANY reasoning or
    explanation for why one performs better, extract it. Don't limit yourself to
     specific types of reasoning.

Output Format:

Return a JSON object with this structure:

{{
   "benchmark_paper": "{benchmark_paper_title}",
   "models_analyzed": [{models_list}],
   "comparative_reasoning": [
      {{
```

```
          "idea_A": "Actual Model A name from the list",
          "idea_B": "Actual Model B name from the list",
          "reasoning": "Single cohesive explanation using idea_A and idea_B,
              combining all reasoning factors from the paper. Capture whatever
              explanation the paper provides for why idea_A performs better or
              differs from idea_B or vise versa. This could be due to design
              differences, training approaches, data choices, theoretical
              properties, implementation details, or any other factors the paper
              mentions - include it all in one coherent statement.",
          "source_section": "Brief context where this reasoning was found (e.g.,
               'Related Work', 'Results Analysis', 'Discussion')"
       }}
    ],
    "overall_assessment": "Summary of how many pairs had comparative reasoning,
        or 'No explicit comparative reasoning found between the specified models
        '"
}}

VALIDATION RULES:
1. Both idea_A and idea_B MUST be actual model names from the list: {
    models_list}
2. All model names in the "reasoning" text MUST be replaced with "idea_A" or "
    idea_B"
3. Reasoning must be grounded in actual paper content, not inferred
4. Only include pairs where the paper explains WHY one is better, not just
    describes differences
5. The reasoning must connect differences to advantages or why one is better
    than the other.
6. If no such reasoning exists for any pair, return empty comparative_reasoning
     array
7. Capture ALL reasoning the paper provides - don't limit to specific types

Important Notes:
- Multiple models: You may receive 3, 4, or more models to analyze. Extract
    reasoning for each pair where the paper explains superiority.
- Not all pairs need reasoning: If the paper doesn't explain why one is better
    (just mentions differences), don't extract a reasoning for that pair.
- Be comprehensive: Whatever reasoning the paper provides about which one is
    better, capture it all in the reasoning field.
- Reasoning vs Differences: The paper must explain the consequence or advantage
     of the difference, not just state the difference.

Example with 4 models: If you receive models A, B, C, D, you might extract
    reasoning for pairs (A,B), (B,C), and (C,D) if the paper provides reasoning
    for those comparisons, but skip (A,C), (A,D), (B,D) if no reasoning exists
    for those pairs.

Now extract the comparative reasoning from the paper content:
```

Research Goal Extraction

```
You are a research scientist tasked with converting benchmark information into
    a comprehensive research goal. Your job is to identify the core research
    objective that this benchmark addresses and articulate it as a single, well-
    structured paragraph.

{benchmark_info}

REQUIREMENTS FOR THE RESEARCH GOAL:
1. Write as a SINGLE comprehensive paragraph (not multiple sections)
2. Focus on the core RESEARCH OBJECTIVE that this benchmark addresses
3. Include what type of input data is used, what output is expected, and how
    performance is measured
4. Be specific about the research challenge and why it is important
5. Use scientific language but keep it readable and focused
6. Mention the specific benchmark/dataset name
7. Keep the research goal between 3-5 sentences

INSTRUCTIONS:
- Write a cohesive paragraph that flows naturally
- Start with the research objective or problem being addressed
- Include input/output specifications naturally within the paragraph
- Mention evaluation approach without making it a separate section
- Focus on the RESEARCH GOAL, not just describing the benchmark
- Avoid bullet points or structured formatting

EXAMPLES OF GOOD RESEARCH GOALS:

Example 1: "This research aims to develop language models that can accurately
    determine logical relationships between premise-hypothesis text pairs,
    addressing the fundamental challenge of natural language inference. The
    research involves training models to process paired natural language
    sentences and classify whether the hypothesis is entailed, contradicted, or
    neutral with respect to the premise, with performance measured using
    classification accuracy and F1-score across the three relation types. This
    work is essential for advancing machine reading comprehension and logical
    reasoning capabilities in natural language processing systems."

Example 2: "This research focuses on creating neural machine translation models
     that can produce high-quality translations between low-resource language
    pairs using minimal parallel training data. The objective is to develop
    systems that can process source language sentences from news articles and
    web documents and generate target language translations that preserve
    semantic meaning and grammatical correctness, evaluated using BLEU scores,
    chrF scores, and human evaluation ratings for fluency and adequacy. This
    research addresses the critical need for effective translation systems in
    underrepresented languages where large parallel corpora are not available."

BENCHMARK TO PROCESS: {benchmark.benchmark_name}

Generate a single research goal paragraph:
```

True Original Paper analysis

```
You are analyzing an academic paper to determine if the given method or models
    were originally introduced.

Paper Title: "{paper_title}"
Models or methods to analyze: {models_list}

CRITICAL: Treat each model name as ONE COMPLETE MODEL OR METHOD NAME. Do NOT
    split model names like "RNN-1024 + 9 Gram" into separate components. Each
    model or method name listed above should get exactly ONE analysis entry,
    regardless of what symbols ("+", "&", "with", etc.) it contains.

{content_section}For each model or method, follow this process:

1. Analyze each model name as a complete unit: Take the EXACT model or method
    name as given and analyze it as one single model/method, even if it contains
     symbols like "+", "&", "with", etc.

2. Check if originally introduced: You may look for phrases like "we propose",
    "we introduce", "we present [exact_model_name]", "our [exact_model_name]",
    detailed descriptions indicating novelty or any other relavant context.

3. If NOT originally introduced: Look for citations when the complete model or
    method name is mentioned:
  - Find phrases like "using [exact_model_name] from [citation]", "based on [
      exact_model_name] [citation]", "[exact_model_name] (Author et al.)" etc.,
      but be mindful of cases where the exact model name is just a variant of
      the original (Like MethodX(unidirectional) etc.). Or any other form of
      citations present with the model name anywhere else (like in tables etc.).
  - Locate the citation in the references/bibliography section
  - Extract the original paper title and authors from the reference

4. For combination-style model or method names (e.g., "ModelA + ModelB", "
    Enhanced ModelX", "ModelY with additional components (like trained on
    certain dataset etc.)"):
  - Treat the ENTIRE name as ONE MODEL - do not analyze components separately
  - If the complete combination is a novel approach, mark as
      introduced_in_this_paper = true
  - If the complete combination cites prior work, identify those supporting
      papers
  - Include supporting papers as supporting_paper_title and supporting_authors
      if present

5. Use citations to find original papers: When a model is cited, go to the
    references section and find the complete bibliographic information for that
    citation.

Return JSON format with EXACTLY ONE entry per model name provided:
{{
   "models": [
      {{
         "model_name": "EXACT_MODEL_NAME_AS_PROVIDED",
         "introduced_in_this_paper": true/false,
         "original_paper_title": "Title of original paper (if different, else
             null)",
         "original_authors": "Authors if available from citations (else null)",
         "confidence": "high/medium/low",
         "reasoning": "Brief explanation including citation info if found"
      }}
   ]
}}

IMPORTANT: You must return exactly {len(unique_models)} model entries, one for
    each model name provided. Do NOT split model names into components.
```

Prompt format used for FT with RL

```
system_prompt = (
"You are an expert AI research assistant. Evaluate two research ideas and
    determine which one is better."
)
user_content = (
   "Research Goal: " + research_goal + "\n\n"
   "Idea A: " + idea_A + "\n\n"
   "Idea B: " + idea_B + "\n\n"
   )
user_content += (
   "Please reason step by step about which idea is better. "
   "Then provide your final answer in the format: \"Answer: [0 or 1]\" where 0
       means Idea B is better and 1 means Idea A is better."
   )
```

D Examples

D.1 Qwen3-8B-Reason-SFT-DAPO

An example where the RL trained model successfully reflects and reasons the probable cause for one idea being better than the other and successfully predicts the correct answer. And a second example where the same model reasons and reflects but the reasoning leads to wrong final answer prediction.

Input

```
   "research_goal": "The primary research objective is to develop robust deep
       learning models capable of learning accurate image classifiers from
       training data corrupted by realistic, human-generated label noise. This
       research addresses the significant real-world challenge of training
       models when ground-truth labels are unreliable, a common scenario in
       crowd-sourced or large-scale data collection. Utilizing the CIFAR-10N
       benchmark, models are trained on CIFAR-10 images paired with their
       associated noisy labels, with the ultimate goal of accurately predicting
       the true class for unseen images. The performance of these noise-tolerant
        algorithms is evaluated by their classification accuracy on the original
       , clean CIFAR-10 test set, demonstrating their ability to generalize
       beyond the imperfect training data."

   "idea_A": "A learning paradigm is proposed for robustly training deep neural
        networks in the presence of noisy labels by simultaneously training two
       networks and having them teach each other. The methodology involves
       maintaining two neural networks, f (with parameters w_f) and g (with
       parameters w_g), which have identical architectures but are initialized
       differently to encourage diverse learning behaviors. The training
       proceeds in a mini-batch fashion. For each mini-batch of data, both
       networks perform a forward pass and calculate the loss for every instance
       . Based on these loss values, each network independently selects a
       fraction of the instances it deems most likely to be correctly labeled,
       identified as those with the smallest training losses. The core mechanism
        is a cross-update procedure: network f is updated via backpropagation
       using only the small-loss instances selected by its peer, network g, and
       conversely, network g is updated using the small-loss instances selected
       by network f. This process is founded on two principles. First, it
       leverages the memorization effect of deep networks, where models tend to
       learn from clean, easy examples before fitting to noisy, hard ones; thus,
        small-loss instances are likely to be clean, especially in the early
       stages of training. Second, the use of two networks and a cross-update
       step prevents the accumulation of error. Since the two networks have
       different learning abilities, they can filter different types of noise.
       If one network mistakenly selects a noisy instance, the error is passed
       to its peer for an update but is not immediately fed back into the
       originating network, breaking the cycle of error accumulation common in
```

```
    self-evolving systems. The fraction of data selected, R(T), is
    dynamically adjusted over training epochs T. It starts at 1 (using all
    data) and gradually decreases according to the schedule R(T) = 1 - \u03c4
     * min(T / T_k, 1), where \u03c4 is a hyperparameter related to the
    estimated noise rate and T_k defines the number of epochs before the drop
     rate stabilizes. This allows the networks to learn general patterns from
     all data initially and become progressively more selective as they gain
    the capacity to overfit to noise."

"idea_B": "A loss correction procedure is proposed to train neural networks
    on data with class-conditional label noise by modifying the model's
    predictions to account for the noise process. The approach assumes the
    existence of a known `c x c` noise transition matrix `T`, where `T_ij = p
    (\u1ef9 = e_j | y = e_i)` specifies the probability of a true class `i`
    being observed as class `j`. The core mechanism operates by altering the
    output of the network before the loss calculation. For a given input `x`,
     the network first produces logits `h(x)`, which are then transformed by
    an inverse link function `\u03c8\u207b\u00b9` (e.g., the softmax function
    ) to yield a vector of predicted clean label probabilities, `p(y|x)`.
    Instead of comparing this clean prediction to the noisy label `\u1ef9`,
    the procedure projects the clean prediction into the noisy label space by
     left-multiplying it with the transition matrix `T`. The final loss is
    then computed between the observed noisy label `\u1ef9` and this
    transformed prediction, `T * \u03c8\u207b\u00b9(h(x))`. This method is
    theoretically justified for any proper composite loss, `\u2113_\u03c8`,
    such as cross-entropy or square loss. The corrected loss, defined as `\
    u2192\u03c8(h(x)) = \u2113(y, T * \u03c8\u207b\u00b9(h(x)))`, is itself a
     proper composite loss with a new, implicitly defined link function `\
    u03c6 = (T\u207b\u00b9)\u2218\u03c8`. A key theoretical result
    demonstrates that the minimizer of this corrected loss under the noisy
    data distribution is identical to the minimizer of the original loss
    under the clean data distribution. Specifically, minimizing `E_{x,\u1ef9
    }[\u2192\u03c8(y, h(x))]` leads to an optimal model that targets `\u03c6(
    p(\u1ef9|x))`, which simplifies to `\u03c8((T\u207b\u00b9)p(\u1ef9|x)) =
    \u03c8(p(y|x))`. This is the same target as minimizing the original loss
    on clean data, thus guaranteeing noise robustness at the minimizer
    without requiring an explicit inversion of the matrix `T` during
    computation."
```


Output


```
"thinking": "<think>\nThe performance of idea_A is superior to idea_B
    because idea_A is a self-evolving method that trains two networks to
    teach each other, allowing them to learn from all data. In contrast,
    idea_B is a static method that only corrects the loss of a single network
    , which limits its capacity to learn from all data. The training of
    idea_A is designed to break the cycle of error accumulation, a capability
     not present in the single-network approach of idea_B. Furthermore,
    idea_A employs a dynamic noise ratio (R(T)) that starts by using all data
     and gradually selects less data as training progresses, a mechanism that
     enables the model to learn general patterns from all data initially and
    then focus on difficult, noisy instances. This dynamic selection,
    combined with the cross-update procedure between two networks, allows
    idea_A to learn from all data throughout training, whereas idea_B, being
    a single network, cannot learn from all data and only learns from the
    clean data. This comprehensive learning capability of idea_A is a
    significant factor in its superior performance compared to the static,
    single-network idea_B.\n</think>"

"prediction": 0,

"ground_truth": 0
```

Input


```
"research_goal": "The primary research objective is to develop and evaluate
    natural language processing models with a robust understanding of physical
    commonsense reasoning, a critical capability for real-world interaction.
    Using the Physical Interaction Question Answering (PIQA) benchmark, this
    research challenges models to process a question about a physical situation
    and select the more plausible of two potential solutions. Model performance
    is measured by accuracy in choosing the correct solution, directly assessing
     the system's ability to reason about the physical properties and
    affordances of everyday objects. This work is crucial for advancing AI
    systems beyond purely linguistic pattern matching towards a more grounded,
    human-like understanding of the world.",

"idea_A": "A two-stage approach is proposed for creating compact language
    models from larger, pre-trained source models. The first stage, termed
    targeted structured pruning, reduces the model to a pre-specified target
    architecture. This is formulated as a constrained optimization problem that
    learns pruning masks for various model substructures\u2014specifically
    layers, attention heads, intermediate dimensions, and hidden dimensions. The
     pruning decision for each substructure is controlled by a mask variable
    parameterized using a hard concrete distribution, which allows for discrete
    retain-or-prune decisions within a continuous optimization framework.
    Instead of targeting a general sparsity level, the method enforces
    constraints on the final model shape directly using Lagrange multipliers.
    The overall objective is a min-max optimization of the function L_prune(\
    u03b8, z, \u03bb, \u03d5) = L(\u03b8, z) + \u03a3_j L_head_j + \u03a3_j
    L_int_j + L_layer + L_hidden, where L(\u03b8, z) is the language modeling
    loss with masked weights, and the other terms are Lagrangian penalties. For
    instance, the constraint for the number of heads in a layer is L_head(\u03bb
    , \u03d5, z) = \u03bb_head * (z_head - H_T) + \u03d5_head * (z_head - H_T)
    ^2, where H_T is the target number of heads. This process jointly optimizes
    model weights (\u03b8) and pruning masks (z) to find a subnetwork that
    matches the target architecture while preserving performance. After this
    stage, the highest-scoring components are retained to finalize the pruned
    model's structure.\n\nThe second stage involves continued pre-training of
    this pruned model, enhanced by a dynamic batch loading algorithm designed to
     address inefficient learning across different data domains. This algorithm
    adjusts the data sampling proportions on-the-fly based on the model's
    performance in each domain. A 'reference loss' (\u2113_ref) is established
    for each domain, which can be derived either by using a scaling law function
     fitted on a series of models of different sizes to predict the loss of a
    hypothetical model of the target size, or by using the source model's
    validation loss. During training, the model's current validation loss (\
    u2113_t) is periodically evaluated for each domain. The data loading weights
     (w_t) for subsequent training batches are then updated in proportion to the
     difference between the current loss and the reference loss (\u2206_t[i] =
    max{\u2113_t[i] - \u2113_ref[i], 0}), effectively up-sampling data from
    domains where the model's performance is lagging. This ensures that the
    model's loss reduces more evenly across all domains, leading to a more
    efficient use of the training data.",

"idea_B": "A method is proposed for converting a pre-trained dense model into a
     sparse Mixture-of-Experts (MoE) architecture in a parameter-efficient
    manner. The process begins by replacing the feed-forward network (FFN)
    layers within the dense model's transformer blocks with MoE layers. Each MoE
     layer is composed of a set of experts and a gating router. During
    initialization, every expert within a given MoE layer is created as an
    identical copy of the original FFN layer from the dense model, inheriting
    its weights, denoted as \u03b8_o. The core of the method lies in how these
    identical experts are differentiated during training. Instead of fine-tuning
     the large set of parameters \u03b8_o for each expert, which is
    computationally expensive, the expert weights \u03b8_o are kept frozen.
    Differentiation is achieved by inserting a small, trainable adapter module
    after each expert. For an input x processed by the shared expert E, the
    output is passed to a specific adapter A_i. The adapter's transformation is
    defined as A_i(E(x)) = \u03c3(E(x)W_i_down)W_i_up + E(x), where W_i_down and
```

```
    W_i_up are the trainable low-rank matrices of the i-th adapter, and \u03c3
    is an activation function. The training process focuses on optimizing only
    the parameters of these adapters, \u03c9i, for each expert, along with the
    router weights. The final output of the MoE layer is a weighted combination
    of the adapter-modified expert outputs, formulated as y = \u03a3 R(x)i * Ai(
    E(x)), where R(x)i is the gating score for the i-th expert. This approach is
     theoretically grounded in the idea that the adapters, being universal
    approximators, can effectively approximate the functional changes of a fully
     fine-tuned expert with a small error, thus projecting the optimization into
     a much lower-dimensional space. To manage expert selection, a top-k gating
    router is employed, which uses trainable weights Wr to calculate logits R(x)
     = Wr \u2022 x. A KeepTopK function selects the top k experts for each token
     before a softmax normalization. To ensure balanced utilization of experts
    during training, an auxiliary load-balancing loss is incorporated,
    calculated as L = \u03b1 \u2022 n \u2022 \u03a3 fi \u2022 pi, where fi is
    the fraction of tokens dispatched to expert i, and pi is the fraction of
    router probability it receives.",
```

Output

```
   "thinking": "<think>\nThe performance of idea_A is superior to idea_B
       because idea_A undergoes a two-stage training process. The first stage,
       targeted structured pruning, is used to convert a large, dense model (
       like idea_B) into a smaller, more efficient model (idea_A) by removing
       redundant parameters. This process is formulated as a constrained
       optimization problem that jointly learns the model weights and pruning
       masks to retain the highest-scoring components, directly targeting a
       specific, smaller architecture (e.g., 7B parameters) rather than just a
       general sparsity level. The second stage involves continued pre-training
       of the pruned model with a dynamic batch loading algorithm, which adjusts
        the data sampling proportions on-the-fly based on the model's
       performance across different domains. This allows the smaller model (
       idea_A) to be trained efficiently on the same large corpus as the source
       model (idea_B), using only 15\% of the training data. This comprehensive
       training methodology enables idea_A to outperform the larger, directly
       fine-tuned idea_B on downstream tasks, as the training data and compute
       are focused on the smaller model.\n</think>",

"prediction": 0,

"ground_truth": 1
```

D.2 Qwen3-8B-Reason-DAPO

For the same example as the above (the second example in §(D.1), the model trained with DAPO applied directly to the base model with a penalty for shorter responses, we see clear reward hacking. The thinking starts off well and rightly acknowledges the need for evaluating likely performance on the specific task/research goal but then enters a repetitive loop of the same sentences before making the final prediction, thus failing to reason coherently or justifying the "why". Yet it gets the final prediction correct unlike the Qwen3-8B-Reason-SFT-DAPO model.

Output

```
"thinking": "<think>\nOkay, let's try to figure out which idea is better. So,
    we have two methods: one is a pruned model (Idea A) and the other is the MoE
     model (Idea B). Both are derived from the same large model, right? The
    question is about which one performs better on the PIQA task. \n\nFirst, I
    need to understand the differences between the two models. The MoE model (
    Idea B) is the model with the Mixture-of-Experts (MoE) method. The model
    with the Mixture-of-Experts (MoE) method (Idea B) is the model with the
```

```
    Mixture-of-Experts (MoE) method. The model with the Mixture-of-Experts (MoE)
     method (Idea B) is the model with the Mixture-of-Experts (MoE) method. The
    model with the Mixture-of-Experts (MoE) method (Idea B) is the model with
    the Mixture-of-Experts (MoE) method. The model with the Mixture-of-Experts (
    MoE) method (Idea B) is the model with the Mixture-of-Experts (MoE) method.\
    n\nThe model with the Mixture-of-Experts (MoE) method (Idea B) is the model
    with the Mixture-of-Experts (MoE) method. The model with the Mixture-of-
    Experts (MoE) method (Idea B) is the model with the Mixture-of-Experts (MoE)
     method. The model with the Mixture-of-Experts (MoE) method (Idea B) is the
    model with the Mixture-of-Experts (MoE) method.\nThe model with the Mixture-
    of-Experts (MoE) method (Idea B) is the model with the Mixture-of-Experts (
    MoE) method. The model with the Mixture-of-Experts (MoE) method (Idea B) is
    the model with the Mixture-of-Experts (MoE) method.\nThe model with the
    Mixture-of-Experts (MoE) method (Idea B) is the model with the Mixture-of-
    Experts (MoE) method.\nAnswer: 1\n</think>"

"prediction": 1,

"ground_truth": 1
```


D.3 Qwen3-8B-Reason-SFT-DrGRPO

As mentioned in §(7.3), the RL trained model using Dr. GRPO started generating very minimal number of think tokens, often just a direct statement of which idea is better and without justifying ”why” this leads to better performance.

Input


```
"research_goal": "The primary research objective is to develop computational
     models capable of robust common sense reasoning, specifically for the
    task of pronoun coreference resolution in ambiguous contexts. This
    research utilizes the WinoGrande benchmark, which presents models with a
    sentence containing a pronoun and requires them to select the correct
    antecedent from two plausible options, a task designed to necessitate
    real-world knowledge. Performance is measured by classification accuracy,
     where the adversarially-filtered construction of the dataset ensures
    that success is indicative of genuine reasoning rather than reliance on
    superficial dataset artifacts. Advancing model capabilities on this
    benchmark is crucial for building AI systems that can comprehend and
    reason about nuanced human language."

"idea_A": "A prefix-guided multi-task learning framework is proposed to
    explore inter-task relationships and jointly train a model on numerous
    natural language understanding (NLU) tasks. The methodology begins by
    unifying all tasks into a multiple-choice-like format, where each
    training example consists of a context, a question, and a fixed number of
     `k` candidate options. To manage variability in the original datasets,
    examples with more than `k` options have redundant options randomly
    discarded, while those with fewer are padded with 'N/A' placeholders.
    Each resulting data instance is then formatted into a sequence by
    prepending a unique, task-specific prefix token, resulting in an input
    structure of `{[Prefix]: context, question, option}`. The model, based on
     an encoder-only architecture, is trained using a dual-objective function
     `L = L_mtl + \u03bbL_mlm`, where `\u03bb` is a balancing hyper-parameter
    . The first objective, `L_mtl`, is a supervised task loss where the model
     functions as a discriminator `g(\u2022, \u2022)`. For each example `(c_i
    , q_i)` with a set of options `r`, the model computes a matching score `g
    (c_i, q_i \u2022 r_j)` for each option `r_j`, and the model is trained
    with a classification loss to select the option with the highest score.
    The second objective, `L_mlm`, is a self-supervised denoising loss based
    on masked language modeling (MLM), defined as `L_mlm = -\u03a3_{i=1 to N}
     \u03a3_{j\u2208M} log p_\u03b8(t_{i,j} | x_i')`, where `x_i'` is a
    sequence with randomly masked tokens and `M` is the set of masked indices
    . A key mechanism of this framework is that the MLM objective is applied
```

```
        to the entire input sequence, including the task-specific prefix tokens,
        which are also subject to random masking. By requiring the model to
        predict the correct prefix based on the content of the data instance, it
        is compelled to learn prefix embeddings that capture task-specific
        differences and common patterns. Consequently, the learned vector
        representations of these task prefixes serve as a direct means to probe
        and quantify the relationships between the various tasks."

    "idea_B": "A parameter-efficient fine-tuning method is proposed that
        constructs a sparse Mixture-of-Experts (MoE) model by augmenting a frozen
        , pre-trained dense transformer. This is achieved by replacing the
        standard feed-forward network (FFN) layer in each transformer block with
        a specialized MoE block. This block consists of a set of experts and a
        top-k gating router. A key architectural principle is that the experts
        are not independent neural networks; instead, they are formed by
        combining a single, shared, frozen FFN from the base model with unique,
        expert-specific Low-Rank Adaptation (LoRA) matrices. Specifically, each
        of the `n` experts consists of the shared FFN weights `W` and its own set
         of trainable LoRA matrices, `B_k` and `A_k`, for each linear layer
        within the FFN. During the forward pass, for a given hidden state `h`, a
        trainable linear router `R` computes scores for each expert. A top-k
        gating mechanism then selects the `k` experts with the highest scores.
        The output of the MoE block is a weighted sum of the outputs from these
        selected experts, where the weights are the normalized scores from the
        router. The output of a single expert `E_k` for an input `h` is the
        result of the shared FFN computation, modified by the expert's specific
        LoRA update. The forward pass for the MoE block is defined as `MixLoRA(h)
         = \u03a3_{k=1 to K} R(h)_k * E_k(h)`, where `R(h)_k` is the router's
        weight for the k-th selected expert. The router itself uses a softmax
        function over the outputs of a linear layer `W_r` to generate a
        probability distribution, from which the top `k` experts are chosen: `R(h
        ) = KeepTop-k(Softmax(W_r * h))`. To counteract the tendency for routers
        to disproportionately favor certain experts, an auxiliary load balancing
        loss is incorporated into the total training loss. This loss is
        calculated as `L_aux = \u03b1 * N * \u03a3_{i=1 to N} F_i * P_i`, where `
        N` is the number of experts, `F_i` is the fraction of tokens in a batch
        dispatched to expert `i`, `P_i` is the average router probability
        allocated to expert `i`, and `\u03b1` is a scaling coefficient. To
        further enhance model adaptation, the architecture also integrates
        separate, independent LoRA adapters into the query, key, value, and
        output projection matrices of the self-attention layers, which are
        trained concurrently. A computational optimization is also employed where
        , for the FFN's parallel linear projections (e.g., `W1` and `W3` in a
        LLaMA-style FFN), the computation is performed only once on the input
        sequence. The resulting activation tensors are then sliced and routed to
        the respective expert-specific LoRA computations, thus avoiding redundant
         calculations across the shared FFN components."
```


Output


```
    "thinking": "<think>\nidea_A is a prefix-guided multi-task learning
        framework that jointly trains a model on numerous natural language
        understanding (NLU) tasks, which gives it an advantage over idea_B, which
         is a parameter-efficient fine-tuning method.\n</think>",

    "prediction": 0

    "ground_truth": 0
```


D.4 Qwen3-8B-Synthetic-Reason-SFT-DAPO

An example of Synthetic-SFT-DAPO model reasoning and debating multiple strengths and weaknesses before correctly concluding the better idea for the given research goal. Unlike Reason-SFT-DrGRPO, it does not tend to minimize the reasoning traces or the responses.

Input


```
"research_goal": "The primary research objective is to develop question-
    answering systems capable of multi-step reasoning that requires
    synthesizing explicit knowledge with implicit common sense. Using the
    OpenBookQA benchmark, this research challenges models to process an
    elementary-level science question and a provided set of core facts to
    select the correct multiple-choice answer. This task addresses the
    critical need for models to move beyond simple fact retrieval and instead
     apply provided knowledge to novel situations, a crucial capability for
    advanced natural language understanding. Performance is evaluated based
    on the model's accuracy in answering these questions, which are
    specifically designed to be unsolvable by simpler retrieval-based or word
     co-occurrence algorithms.",

"idea_A": "A model is proposed for question answering that performs joint
    reasoning over a language context and a knowledge graph (KG) by unifying
    them into a single graphical structure. The process begins by retrieving
    a KG subgraph `G_sub` relevant to the entities mentioned in the QA
    context. A novel 'working graph' `G_W` is then constructed. This is
    achieved by introducing a special 'QA context node' `z` that represents
    the concatenated question and answer choice. This `z` node is then
    connected to the topic entities (those mentioned in the question or
    answer) within the `G_sub` via newly defined, typed relations (`r_z,q`
    for question entities, `r_z,a` for answer entities). The initial
    representation of the `z` node is derived from an LM encoding of the QA
    context, while KG nodes are initialized with their entity embeddings. To
    address the issue of irrelevant nodes in the retrieved subgraph, a
    relevance scoring mechanism is introduced. For each KG node `v`, a
    relevance score `\u03c1_v` is computed by feeding the concatenation of
    the QA context text and the node's entity text to a pre-trained LM,
    formulated as `\u03c1_v = f_head(f_enc([text(z); text(v)]))`. This score
    quantifies the node's importance relative to the QA context and is used
    as a feature. Reasoning is performed on this working graph using a multi-
    layer graph neural network (GNN) based on a graph attention framework. At
     each layer `l`, the representation of each node `t` is updated via
    message passing: `h^(l+1)_t = f_n(\u03a3_{s\u2208N_t\u222a{t}} \u03b1_st
    * m_st) + h^(l)_t`, where `m_st` is the message and `\u03b1_st` is the
    attention weight. The message computation `m_st` is node type- and
    relation-aware, defined as `m_st = f_m(h^(l)_s, u_s, r_st)`, where `u_s`
    is the source node's type embedding and `r_st` is a relation embedding.
    The attention mechanism `\u03b1_st` is node type-, relation-, and score-
    aware. It computes query `q_s = f_q(h^(l)_s, u_s, \u03c1_s)` and key `k_t
     = f_k(h^(l)_t, u_t, \u03c1_t, r_st)` vectors, where `\u03c1` represents
    an embedding of the relevance score. The attention weight is then `\
    u03b1_st = exp(\u03b3_st) / \u03a3_{t'\u2208N_s\u222a{s}} exp(\u03b3_st')
    ` with `\u03b3_st = q_s^T k_t / \u221aD`. This iterative process mutually
     updates the representations of both the QA context node and the KG nodes
    . The final prediction for an answer choice is derived from an MLP that
    takes as input the initial LM representation of the context (`z_LM`), the
     final GNN-updated representation of the context node (`z_GNN = h^(L)_z`)
    , and a pooled representation of the final KG node embeddings (`g`). The
    entire model is trained end-to-end by optimizing a cross-entropy loss.",

"idea_B": "The proposed approach is a 66-billion parameter, decoder-only,
    pre-trained transformer model designed for auto-regressive language
    modeling. The architecture consists of 64 transformer layers, a hidden
    dimension of 9216, and 72 attention heads, with a feed-forward network
    dimension of 36864. It utilizes ReLU activation functions and is trained
    on sequences of 2048 tokens. The model's weights are initialized from a
    normal distribution with a mean of zero and a standard deviation of
    0.006, with the standard deviation for output layers scaled by a factor
    of 1.0/\u221a2L, where L is the number of layers. Pre-training is
    conducted on a 180-billion token corpus, which is a deduplicated
    concatenation of datasets including RoBERTa, The Pile, and PushShift.io
    Reddit, tokenized using a GPT-2 byte-level Byte Pair Encoding (BPE)
    tokenizer. The training process employs the AdamW optimizer with \u03b21
    =0.9, \u03b22=0.95, and a weight decay of 0.1. A linear learning rate
```

```
schedule is used, warming up over the first 375 million tokens and then
decaying to 10\% of the maximum learning rate over 300 billion tokens.
Regularization is applied through a dropout of 0.1 on all layers except
embeddings and gradient norm clipping at 1.0. For task execution in a one
-shot setting, the model is provided with a single in-context example
sampled from the task's training set. For classification tasks with a
predefined set of candidate answers, the model performs likelihood-based
classification by selecting the candidate (\u03b1) that maximizes a
specific probability score given the context (s). This selection can be
based on one of three methods: the regular conditional probability `p(\
u03b1|s)`, a calibrated probability `p(\u03b1|s)/p(\u03b1|\"Answer:\")`
which normalizes by the candidate's prior probability, or a length-
normalized probability `p(\u03b1|s)/len(\u03b1)`. For generative tasks,
the model produces its output using greedy decoding."
```

Output

```
   "thinking": "<think>
**Evaluating OpenBookQA model**

I'm looking at the OpenBookQA dataset, which uses core facts from OpenBook. The
     research goal is to develop QA systems that can handle multi-step reasoning
    . A 66B decoder-only model might not be the best approach since it hasn't
    been pre-trained on science knowledge. The model could use in-context
    learning, but it might not outperform specialized models. It seems that for
    OpenBookQA, a retrieval-based method would be more effective. The second
    idea is to create a joint reasoning model that combines a language context
    with a knowledge graph to help with question answering.

**Analyzing OpenBookQA and KG integration**

I'm looking at OpenBookQA, which uses core facts from OpenBook. The research
    goal is to develop systems that can synthesize explicit knowledge with
    common sense. The first idea is to build a working graph that connects a QA
    context to relevant knowledge graph nodes. This could help with multi-step
    reasoning and avoid overfitting. The second idea is to use a large pre-
    trained model like the one from Meta, which could potentially handle
    OpenBookQA. However, it might not be as effective as a model that combines
    retrieval with a knowledge graph for better results.

**Evaluating model performance and approaches**

I'm looking at how a 66B model could perform with one-shot learning. It might
    not be the best approach since it could lead to overfitting. The research
    goal is to develop systems that can reason beyond simple fact retrieval. The
     OpenBookQA dataset requires using core facts, but a generative model might
    not be able to do that. The first idea is to use a knowledge graph (KG) that
     connects concepts, which could help with reasoning. It seems that combining
     a language model (LM) with a graph neural network (GNN) could be more
    effective for this task. The second idea is to use a 66B model for
    pretraining, which might not be necessary.

**Assessing model training and evaluation**

I'm looking at how a model could be trained on a large dataset to help with
    generalization. The research goal is to develop question-answering systems
    that can handle multi-step reasoning. The 66B model might not be as
    effective as a fine-tuned approach. It seems that without retrieval, it won'
    t perform well on OpenBookQA. The evaluation will focus on accuracy, and I
    think the first idea using a knowledge graph (KG) is better. It can help
    with multi-hop reasoning and reduce irrelevant information. The second idea
    of using a pre-trained model could work for in-context learning, but it
    might not be as effective as the first approach.

**Evaluating ideas for OpenBookQA**
```

I'm looking at how the OpenBookQA dataset might not have enough data for a 66B model to learn from. The research goal is to develop systems that can synthesize knowledge. The first idea uses a joint reasoning approach with a knowledge graph (KG) to connect entities, which should help with multi-step reasoning. The second idea is a pre-trained model that could potentially outperform baselines, but it might not be as effective.

So, I think Idea A is better because it combines a language model with a graph neural network (GNN) for better performance. It can handle retrieval and relevance scoring, which will help with OpenBookQA.

Answer: 0
</think>",

"final_answer": "Here is a concise, structured evaluation and conclusion.

- Initial observations:
  - Idea B is a very large, generic, pre-trained decoder-only model with standard training and one-shot usage. It does not include any task-specific mechanism for incorporating provided core facts or a knowledge graph.
  - Idea A is a task-specific, end-to-end model that explicitly integrates a language context with a knowledge graph, includes a relevance scoring mechanism, and uses a GNN with type-aware and relation-aware message passing. It is designed to handle multi-step reasoning and is directly applicable to OpenBookQA.

- Strengths and weaknesses:
  - Idea B strengths: high capacity, can leverage broad pretraining, easy to deploy. Weaknesses: no explicit use of provided core facts, no graph reasoning, one-shot may be insufficient, and it is likely to underperform on OpenBookQA without additional components.
  - Idea A strengths: explicit reasoning over structured knowledge, can select and score relevant nodes, handles multi-hop relations, and can be trained end-to-end. Weaknesses: more complex to implement, potential for overfitting to the graph, and requires good subgraph retrieval.

- Key considerations leading to the conclusion:
  - OpenBookQA requires applying provided core facts and often benefits from graph-based reasoning. A 66B model alone is unlikely to outperform a graph-augmented approach that can explicitly reason over the provided facts.
  - The research goal emphasizes moving beyond simple retrieval and toward multi-step reasoning. Idea A directly addresses this with a working graph and GNN. Idea B does not.

- Why one is better:
  - Idea A is better aligned with the research goal and is more likely to achieve higher accuracy on OpenBookQA due to its explicit integration of knowledge and reasoning.

Answer: 0",

"prediction": 0,

"ground_truth": 0