BEHAVIOR AMPLIFICATION: DISCOVERING AND STEERING MODEL-SPECIFIC REASONING STRENGTHS IN LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent work has advanced the ability of Large Language Models (LLMs) to reason about complex tasks. However, the majority of methodologies treat reasoning as a monolithic capability, whereas state-of-the-art models exhibit a high degree of diversity in their underlying reasoning behaviors. In this paper, we study how model-specific behavioral patterns are associated with success or failure in question answering. We propose a data-driven framework for discovering model-specific Behavioral Strength Sets: reasoning behaviors that are strongly correlated with answer correctness. These correlated behaviors are identified and selected via contrastive analysis and significance testing. Subsequently, we introduce Inherent Strength Amplification (ISA), a steering framework that leverages these behaviors to construct targeted interventions, including prompt injection, behavioral reflection, supervised fine-tuning (SFT), and reinforcement learning (RL). Across three model families and three question answering benchmarks, ISA consistently improves answer correctness over unsteered baselines, with gains of up to 41.0 EM and 29.2 F1 points. Our findings indicate that the effectiveness of each intervention depends upon the behaviors being amplified: behaviors that can be directly expressed as inference-time instructions are often improved through prompting or reflection, whereas behaviors requiring consistent multi-step execution derive greater benefit from parameter-level training. These outcomes demonstrate that effective reasoning improvement requires the alignment of interventions with a model's specific behavioral profile, rather than the application of a universal optimization strategy.

1 INTRODUCTION

While Large Language Models (LLMs) are converging in performance on standard benchmarks, their reasoning behaviors remain highly diverse. We define *reasoning behaviors* as observable patterns in generated reasoning trajectories that characterize how a model transforms a question and available evidence into an answer. Models with comparable accuracy often follow different reasoning patterns to reach their answers, and their successes and failures reveal model-specific behavioral strengths and weaknesses. This suggests that LLM reasoning is better understood not as a single monolithic capability, but rather as a composition of heterogeneous model-specific behaviors.

Despite observable diversity in reasoning behaviors, existing methods for improving LLM reasoning typically encourage a single reasoning strategy across models, either explicitly [4] or implicitly through training on curated reasoning datasets [27]. These approaches offer limited guidance for identifying model-specific strengths: fixed strategies may overlook behaviors that are uniquely beneficial to a particular model, while manual specifications lack a reproducible, quantitative basis for deciding which behaviors to target.

In this paper, we address this gap with a data-driven methodology for discovering reasoning behaviors that are strongly associated with answer correctness for each model. Our key insight is that behaviors occurring significantly more often in correct traces than in incorrect ones reveal model-specific reasoning strengths. We capture this contrast with a differential presence score Δ, which measures how strongly each behavior is associated with correct outcomes. Behaviors with statistically significant

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

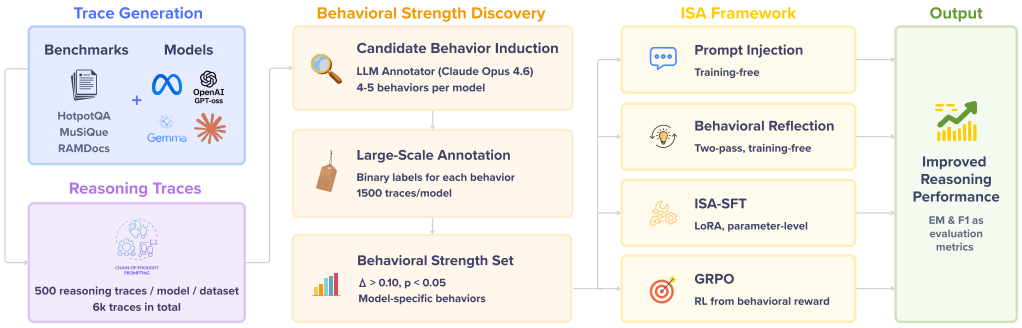

Figure 1: Overview of Behavior Amplification. We first generate reasoning traces with four models on three multi-hop QA benchmarks. Behavioral Strength Discovery then induces candidate set of behaviors, annotates traces with binary behavior labels, and selects each model's Behavioral Strength Set using differential presence in correct versus incorrect traces. ISA uses the resulting strength set to construct four steering interventions: prompt injection, behavioral reflection, ISA-SFT, and RL via GRPO. Performance is evaluated using EM and token-level F1 on held-out test splits.

Δ values form a model's Behavioral Strength Set: a compact set of reasoning patterns strongly predictive of a correct answer.

Building on this discovery process, we introduce Inherent Strength Amplification (ISA), a framework for improving reasoning by amplifying the behaviors identified in each model's Behavioral Strength Set. Here, *inherent* refers to behaviors observable in a model's generated traces, rather than externally imposed reasoning templates. ISA translates each Behavioral Strength Set into targeted interventions spanning prompt injection, behavioral reflection, supervised fine-tuning (SFT), and reinforcement learning (RL). Figure 1 summarizes the full Behavior Amplification workflow: reasoning trace generation, Behavioral Strength Discovery, and ISA-based steering. Together, these components provide a unified pipeline from behavior discovery to behavior-aware optimization without manual specification of reasoning strategies.

Across three model families and three multi-hop question answering benchmarks, the non-test-selected ISA intervention improves over unsteered baselines in all nine open-weight settings. Our central finding is that no single intervention is universally optimal; rather, steering effectiveness depends on the type of behavior being amplified. Behaviors that can be stated as direct inference-time instructions benefit most from prompting or reflection, while behaviors requiring consistent multi-step execution benefit more from parameter-level training.

This paper makes three main contributions: (1) We propose Behavioral Strength Discovery, a data-driven framework for identifying model-specific reasoning behaviors from contrastive analysis of correct and incorrect reasoning traces. (2) We introduce ISA, a unified pipeline that translates discovered behaviors into targeted interventions across prompting, reflection, supervised fine-tuning, and reinforcement learning. (3) We empirically demonstrate that the effectiveness of different interventions depends on the type of behaviors being amplified, supporting our claim that reasoning improvement benefits from behavior-aware, model-specific optimization rather than uniform strategies.

2 Related Work

The emergence of CoT prompting [23] demonstrated that LLMs can achieve significantly higher performance by generating intermediate reasoning steps [20; 11; 12; 21]. This paradigm shift has sparked research interest in increasing model performance by improving models' reasoning capabilities. Though some research has extended linear CoT to more complex structured reasoning [25; 16; 29], optimizing models' core reasoning capabilities remains a key challenge. As reasoning has become a central driver of model performance, research focus has shifted from strictly outcome-based evaluation toward evaluation of reasoning traces [13]. To address this, Process Reward Models

(PRMs) have emerged as a popular method for step-level reasoning evaluation [8; 17; 28]. However, a significant limitation of PRMs is their limited ability to provide insight into patterns of reasoning success or failure beyond the step level. Our work aims to address this gap by analyzing reasoning at the pattern level, and correlating those patterns with answer correctness.

Recent work has sought insight into how LLMs perform reasoning by characterizing and taxonomizing patterns of reasoning behaviors [10; 2; 7]. Kim et al. [9] characterize some models' reasoning traces as conversations between distinct personalities with diverse socio-emotional roles that construct higher-quality answers by reconciling their different perspectives. Gandhi et al. [4] identify verification, backtracking, subgoal setting, and backward chaining as key reasoning behaviors useful for solving math problems. While other works propose singular reasoning taxonomies that work well for the specific problems that are the focus of each study, our approach departs from this trend in two ways: (1) we discover behaviors automatically from data via contrastive trace analysis rather than identifying them manually, and (2) the Behavioral Strength Sets we discover are model-specific, allowing each model to have a different steering target.

Recent work suggests that high-quality reasoning traces already lie within high-probability regions of pretrained models' probability distributions. Yue et al. [26] argues that RL primarily shifts models' distributions toward those high-quality traces without learning truly novel reasoning strategies. Karan & Du [6] show that higher-quality reasoning traces can be sampled from an annealed variant of the model's distribution, and Fu et al. [3] and Luo et al. [14] present different strategies for improving model performance by reconciling multiple reasoning traces. Inspired by these insights, this paper investigates how to identify and amplify strong reasoning behaviors that are inherent to each model.

3 Behavioral Strength Discovery

3.1 Reasoning trace generation

We generate reasoning traces with four model families: Llama (Llama-3.1-8b-Instruct) [5], gpt-oss (gpt-oss-20b) [15], Gemma (Gemma 4 26B A4B) [18], and Claude (Sonnet 3.7) [1]. We use CoT prompting to produce step-by-step reasoning for Llama-3.1-8b-Instruct. For gpt-oss-20b, we use medium reasoning effort. For Gemma 4 26B A4B and Claude Sonnet 3.7, we use their default or enabled reasoning configurations. Full prompts and generation hyperparameters are provided in Appendix B.1. To maintain computational feasibility during the SFT and RL stages discussed in Section 4, we prioritize the smaller-parameter variants within each model family.

We evaluate on three multi-hop question answering benchmarks: HotpotQA [24], MuSiQue [19], and RAMDocs [22]. For each model–dataset pair, we sample 500 examples and partition them into 100 examples for behavior discovery, 200 for intervention training, and 200 for held-out evaluation. The discovery split is used to induce the candidate behavior schema, while the discovery and training splits are combined for behavioral annotation and statistical identification of the final Behavioral Strength Set. The held-out evaluation split is excluded from behavior selection and used only for final evaluation. We report Exact Match (EM) and token-level F1 as evaluation metrics. EM measures strict agreement with the ground-truth answer, while F1 accounts for minor surface-form variation.

3.2 Trace correctness definition

To separate correct from incorrect reasoning traces while reducing sensitivity to surface-form artifacts, we apply standard text normalization to each model's final answer before behavioral discovery. Normalization includes lowercasing, punctuation removal, article removal, and whitespace normalization. A trace is treated as correct if its normalized final answer matches the normalized ground-truth answer, and incorrect otherwise.

This procedure identifies the correct and incorrect trace groups used to compute the differential presence score Δ in Section 3.3. Importantly, correctness depends only on the final answer, not on any subjective assessment of the reasoning process itself.

3.3 THREE-STAGE DISCOVERY PIPELINE

Given the generated traces and correctness groups defined above, we identify each model's Behavioral Strength Set through three stages: candidate behavior induction, large-scale behavioral annotation, and statistical filtering.

Stage 1: Candidate behavior induction. For each model m, we sample traces from the discovery split and provide them to Claude Opus 4.6 with a structured prompt asking it to identify recurring reasoning behaviors observed across the sampled traces. The full induction prompt is provided in Figure 7 in Appendix C. For each behavior, the annotator returns a schema containing a behavior name, definition, signal phrases (wording indicative of the behavior), and absence indicators.

This produces an initial candidate behavior set for each model. Across models, the discovered behaviors cluster into five high-level categories: Question Understanding, Evidence Grounding, Chain Reasoning, Entity Resolution, and Answer Formation. Figure 2 summarizes this emergent taxonomy. Although individual models specialize in different behaviors within these categories, the independently discovered taxonomies overlap at the level of high-level reasoning competencies.

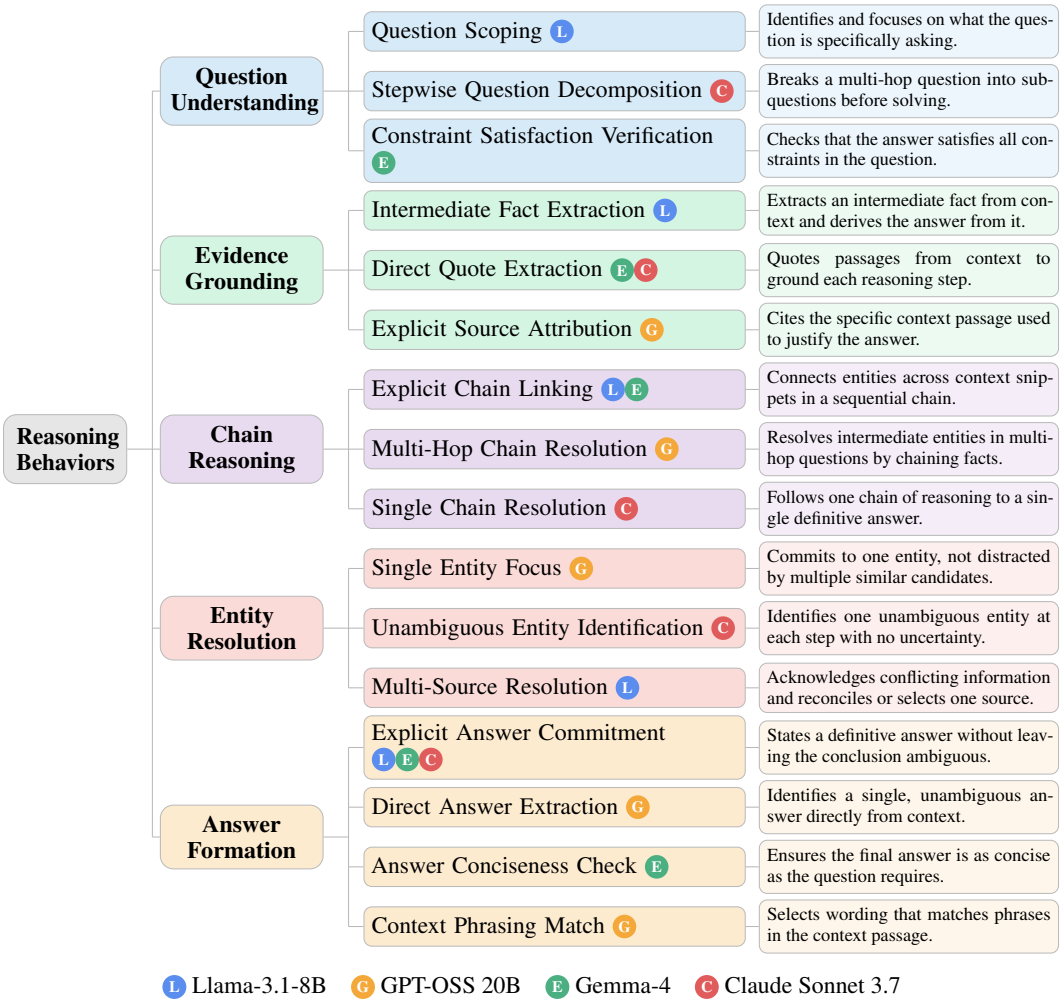

Figure 2: Taxonomy of reasoning behaviors discovered by the Behavioral Strength Discovery framework, grouped into five categories. Colored dots indicate which models exhibit the behavior.

Stage 2: Large-scale behavioral annotation. Using the candidate behavior schema from Stage 1, Claude Opus 4.6 annotates all non-test traces (300 traces per model–dataset pair) for behavioral presence. The held-out evaluation traces are never used in behavior selection. For each trace and

candidate behavior, the annotator receives only the trace and behavior schema, then returns a binary present/absent label indicating whether the behavior appears in the trace. The behavior annotator does not receive correctness labels, preventing circular dependence between outcome labels and behavior labels. The full annotation prompt is provided in Figure 8 in Appendix D. Because behavioral annotations are based on observable trace text, the resulting Behavioral Strength Sets should be interpreted as practical steering targets rather than complete explanations of a model's internal reasoning process.

Stage 3: Statistical identification of behavioral strengths. For each candidate behavior b and model m, we compute its differential presence score on discovery+training set:

$$\Delta(b, m) = P(b \mid \text{correct}, m) - P(b \mid \text{incorrect}, m).$$

This score measures the association between behavior b and successful reasoning for model m: larger positive values indicate that the behavior appears more often in correct traces than in incorrect traces.

To retain only reliable steering targets, we require both practical and statistical significance. Specifically, the *Behavioral Strength Set* for model m is defined as:

$$\mathcal{B}_m^+ = \{b : \Delta(b, m) > 0.10 \text{ and } p_b < 0.05\},$$

where p_b is obtained from a two-proportion Z-test comparing behavior presence rates in correct and incorrect traces. Behaviors in $\mathcal{B}_m^+$ are the retained behaviors used as steering targets in Section 4.

The $\Delta > 0.10$ criterion serves as a practical effect-size floor, requiring a behavior to occur at least 10 percentage points more frequently in correct than incorrect traces, rather than being selected solely based on statistical significance. The selection is robust to this threshold choice: increasing the effect-size threshold from 0.10 to 0.15 changes each Behavioral Strength Set by at most one behavior. Moreover, bootstrap resampling shows that strongly selected behaviors ($\Delta \geq 0.15$) are reselected in 95–100% of resamples, with most instability confined to behaviors near the 0.10 boundary. Finally, to account for multiple hypothesis testing across candidate behaviors, we apply Benjamini–Hochberg False Discovery Rate (FDR) correction. All retained Behavioral Strength Sets remain completely unchanged post-correction (all selected behaviors $q \leq 2.6 \times 10^{-4}$).

Figure 3 reports the resulting Behavioral Strength Sets. Although individual behaviors can recur across models, no two models share the same retained set: each model's Behavioral Strength Set is a distinct combination of behaviors. It is this per-model combination, rather than any single behavior, that we amplify as the steering target in Section 4.

4 Inherent Strength Amplification framework

Given each model's Behavioral Strength Set $\mathcal{B}_m^+$, we introduce Inherent Strength Amplification (ISA), a steering framework that reinforces behaviors already observed in a model's own correct traces. ISA constructs four interventions directly from the discovered behavior schema.

Prompt injection For each model, we construct an injection prompt that describes the behaviors in $\mathcal{B}_m^+$ and instructs the model to exhibit them when answering. The injection is prepended to the user prompt at inference time, requiring no additional training. The full prompt template is provided in Appendix E.

Behavioral reflection We implement a two-pass inference procedure that targets traces with low Behavioral Strength Set coverage:

- **Pass 1:** The model generates a reasoning trace using its standard baseline prompt.
- **Trigger:** We count how many behaviors in $\mathcal{B}_m^+$ appear in the trace. If the count is below a model-specific threshold τ_m, reflection is triggered.
- **Pass 2:** The model regenerates the answer with an instruction to incorporate the missing behaviors.

We set τ_m to the floor-rounded mean number of Behavioral Strength Set behaviors present in correct training traces, making the trigger a data-driven estimate of the model's successful behavior profile.

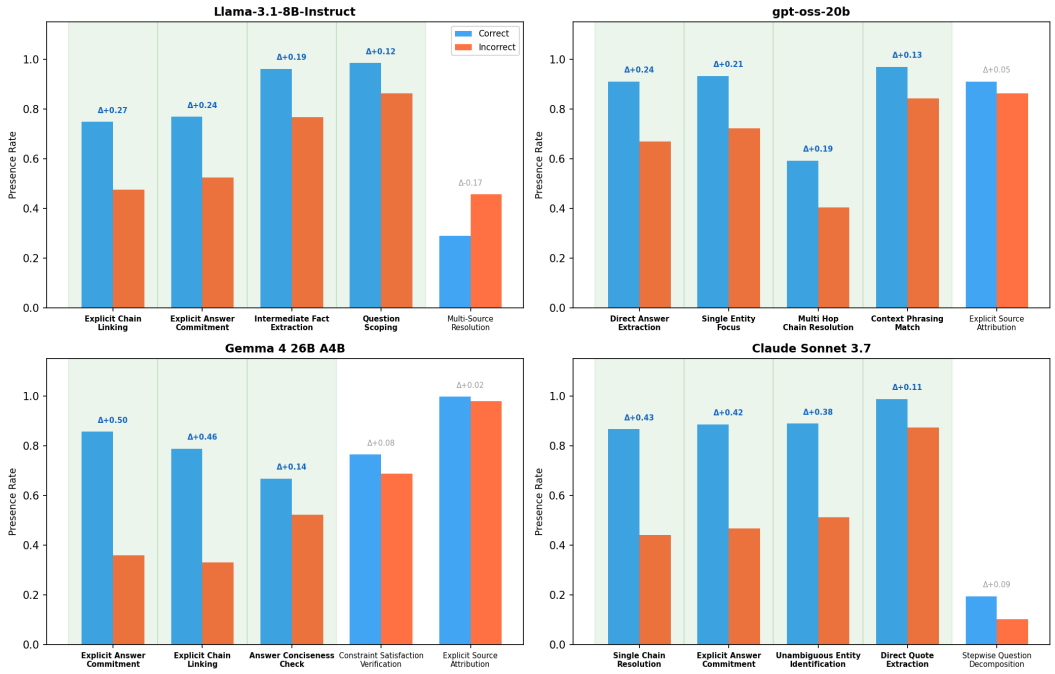

Figure 3: Behavior presence rate in correct vs. incorrect traces for each model. For each candidate behavior, blue bars show $P(b \mid \text{correct}, m)$ and orange bars show $P(b \mid \text{incorrect}, m)$. Values above each pair report the differential presence score $\Delta(b, m)$. Behaviors retained in the model's Behavioral Strength Set—those satisfying $\Delta(b, m) > 0.10$ and $p < 0.05$ under a two-proportion Z-test—are highlighted with a green background in the plots.

This yields $\tau_m = 3$ for Llama-3.1-8b-Instruct, gpt-oss-20b, and Claude Sonnet 3.7, and $\tau_m = 2$ for Gemma 4 26B A4B. Because reflection only runs a second pass when Behavioral Strength Set coverage is low, it reduces inference cost relative to always applying two-pass inference.

Supervised fine-tuning (SFT) ISA-SFT trains on a subset of reasoning traces selected from the training split. A trace is selected if its final answer is correct and if the trace exhibits all behaviors in $\mathcal{B}_m^+$. This differs from vanilla SFT, which trains on all correct traces regardless of behavior annotations. We apply ISA-SFT to the three open-weight models; Claude Sonnet 3.7 is excluded from parameter-level tuning because it is an API-only model.

Reinforcement learning (RL) via GRPO We apply GRPO as an online RL stage initialized from the ISA-SFT checkpoint. At training time, GRPO samples $G = 2$ completions per prompt and uses their relative rewards to construct the policy update, avoiding both a static preference-pair dataset and a learned value function. For each completion y, gold answer a, and extracted answer $\hat{y}$, we define:

$$r(y, a) = \alpha s_{\text{behav}}(y) + \beta \mathbf{1}[\text{EM}(\hat{y}, a)], \tag{1}$$

where $s_{\text{behav}}(y) \in [0, 1]$ is a phrase-based score measuring the fraction of Behavioral Strength Set behaviors whose signal phrase (as determined during the candidate behavior induction stage of the behavior discovery pipeline described in Section 3.3) appear in y. This scoring requires no LLM-judge calls during training. The second term is a binary correctness reward using normalized exact match. We set $\alpha = 1.0$ and $\beta = 0.5$, weighting behavioral strength more heavily than answer correctness.

5 RESULTS AND DISCUSSION

For each model–dataset pair, we select the intervention with the highest EM on the discovery split, then evaluate only that selected intervention once on the untouched test set. The discovery split is held out from the test set and is not used to train SFT or GRPO; these methods are trained only on the training split, while prompt injection and reflection are training-free. This yields a test-independent selection procedure that can be applied to a new model by evaluating the four interventions on a small held-out split and deploying the selected method. We report end-to-end results on the three open-weight models. Claude Sonnet 3.7 is used for behavior discovery and as a donor behavior set, but is deprecated and excluded from the main label-free results (see Appendix H). Table 1 reports EM and F1 for each model-dataset pair.

Table 1: Main results: test performance of the label-free selected intervention. For each open-weight model-dataset pair we run all four interventions on the discovery split (disjoint from test; no test labels used), select the arg-max EM intervention, and report that single intervention on the untouched 200-example test set. ΔEM is the gain over the no-intervention baseline; **bold** marks the higher of the baseline and selected-intervention EM (resp. F1), and † marks a paired improvement whose ΔEM 95% bootstrap interval (percentile method, 10,000 resamples) excludes zero. Full EM, F1, and ΔEM confidence intervals are reported in Appendix H (Table 6). Results for the closed-weight Claude Sonnet 3.7 (training-free interventions only) are given in Appendix H.

Model	Dataset	Selected	Baseline EM	EM	ΔEM	Baseline F1	F1
Llama-3.1-8b-Instruct	HotpotQA	Prompt Inj.	49.5	**62.0**	+12.5†	63.5	**73.1**
	MuSiQue	Prompt Inj.	34.0	**36.5**	+2.5	47.1	**49.0**
	RAMDocs	GRPO	27.5	**47.5**	+20.0†	34.4	**63.6**
gpt-oss-20b	HotpotQA	ISA-SFT	62.0	**78.0**	+16.0†	73.9	**88.7**
	MuSiQue	ISA-SFT	33.0	**58.5**	+25.5†	48.6	**73.1**
	RAMDocs	ISA-SFT	19.0	**60.0**	+41.0†	42.7	**68.7**
Gemma 4 26B A4B	HotpotQA	Reflection	78.0	**81.5**	+3.5	91.5	**92.7**
	MuSiQue	Prompt Inj.	65.5	**69.5**	+4.0†	77.0	**79.9**
	RAMDocs	Prompt Inj.	19.5	**34.0**	+14.5†	41.7	**43.5**

5.1 MAIN RESULTS

Overall performance. The label-free selected intervention improves over the no-intervention baseline in all nine open-weight model–dataset pairs. The gains are often substantial, especially on harder settings such as RAMDocs: gpt-oss-20b improves from 19.0 to 60.0 EM under ISA-SFT, Llama-3.1-8b-Instruct improves from 27.5 to 47.5 EM under GRPO, and Gemma 4 improves from 19.5 to 34.0 EM under prompt injection.

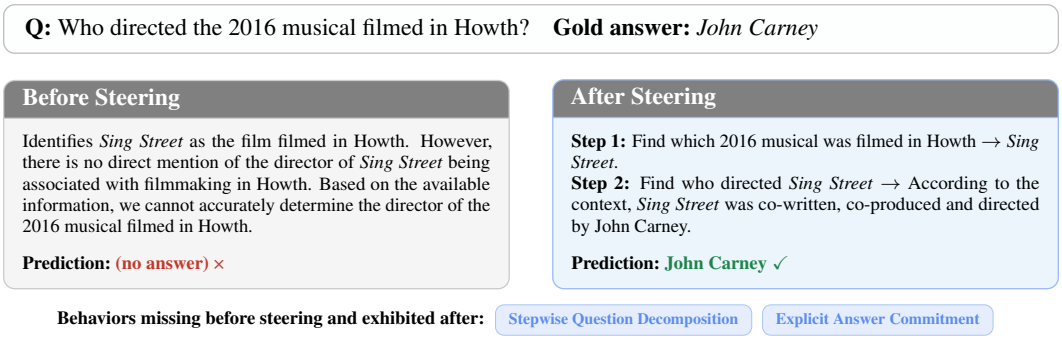

Figure 4: Example of reasoning traces before and after behavior steering (Llama-3.1-8b-Instruct, HotpotQA). In this example, Stepwise Question Decomposition and Explicit Answer Commitment are Behavioral Strength Set behaviors that are absent in the baseline trace and present after steering.

Figure 4 provides a qualitative example of this effect: before steering, Llama-3.1-8b-Instruct identifies the relevant film but fails to commit to the requested answer; after ISA-SFT, it decomposes the

question into explicit steps and extracts the correct answer. These results show that behaviors discovered from correct traces can serve as effective steering targets.

Intervention effectiveness varies by model and behavior. Our most significant finding is that no single ISA intervention dominates across all models. Prompt-based methods are strongest for Gemma 4 26B A4B, while ISA-SFT is strongest for gpt-oss-20b. Llama-3.1-8b-Instruct benefits from both ISA-SFT and GRPO, with GRPO producing the largest gain on RAMDocs. The next two paragraphs trace this split to a property of the behaviors themselves, whether they are instruction-expressible or require sustained multi-step execution. Intervention choice is therefore behavior-aware, not merely model-specific.

Prompting favors instruction-expressible behaviors. Prompt-based interventions are most effective when retained behaviors can be stated as short inference-time instructions rather than requiring repeated execution across many reasoning steps. This pattern is clearest for Gemma 4 26B A4B, whose retained behaviors include Answer Conciseness Check, Explicit Chain Linking, and Explicit Answer Commitment. These behaviors impose high-level constraints on the reasoning trace—such as following a coherent evidence chain and committing to a concise final answer—and can often be activated without parameter updates. This helps explain why prompt injection yields strong gains for Gemma 4 26B A4B. In contrast, prompt injection can degrade performance for some models and tasks, suggesting that explicit behavioral instructions may interfere with models whose output format or reasoning style is less compatible with the injected constraints.

Training-based interventions help when behaviors require consistent execution. Parameter-level steering is most useful when target behaviors are not reliably activated by prompting alone. ISA-SFT exposes the model repeatedly to correct traces where the Behavioral Strength Set is present, making the target behavior pattern more stable at inference time. This is especially effective for gpt-oss-20b, where ISA-SFT substantially outperforms all training-free interventions. GRPO shows a sharper model dependence: it improves Llama-3.1-8b-Instruct beyond ISA-SFT on RAMDocs, but underperforms ISA-SFT for gpt-oss-20b and Gemma 4 26B A4B. This suggests that behavioral rewards can provide additional gains when they remain aligned with answer quality, but phrase-based rewards can also be exploited by reproducing surface signal phrases without improving correctness.

5.2 Ablation studies

Table 2: Ablation on SFT data selection. Vanilla SFT fine-tunes on a volume-matched sample of correct traces without behavior filtering. ISA-SFT uses the same number of training examples but selects correct traces that exhibit all behaviors in the model's Behavioral Strength Set.

		Baseline		**Vanilla SFT**		**ISA-SFT**	
Model	**Dataset**	EM	F1	EM	F1	EM	F1
Llama-3.1-8b-Instruct	HotpotQA	49.5	63.5	65.0	75.8	**66.0**	**78.0**
	MuSiQue	34.0	47.1	35.5	45.1	**36.5**	**48.1**
	RAMDocs	27.5	34.4	32.0	41.2	**33.0**	**41.9**
gpt-oss-20b	HotpotQA	62.0	73.9	37.0	50.9	**78.0**	**88.7**
	MuSiQue	33.0	48.6	19.0	40.0	**58.5**	**73.1**
	RAMDocs	19.0	42.7	22.5	46.3	**60.0**	**68.7**
Gemma 4 26B A4B	HotpotQA	**78.0**	**91.5**	66.5	78.6	71.5	83.1
	MuSiQue	**65.5**	**77.0**	40.5	49.9	46.5	57.9
	RAMDocs	**19.5**	**41.7**	10.5	20.1	15.5	30.8

ISA-SFT versus vanilla SFT. This ablation tests whether ISA-SFT improves reasoning through behavior-based data selection rather than fine-tuning on correct answers, using the same LoRA configuration, training budget, and number of examples as vanilla SFT. Vanilla SFT samples the same number of correct traces as ISA-SFT but does not require Behavioral Strength Set presence; ISA-SFT restricts training to correct traces that exhibit all behaviors in $\mathcal{B}_m^+$. The comparison therefore isolates the value of selecting traces that instantiate the model's discovered strengths. As shown in Table 2, ISA-SFT outperforms vanilla SFT for Llama-3.1-8b-Instruct and gpt-oss-20b, suggesting that correct

traces are not equally useful for steering: traces that contain the target behaviors provide a stronger training signal than correctness alone. Gemma 4 26B A4B is the exception, where both SFT variants degrade performance, indicating that the failure is not caused solely by the behavioral filter but by the suitability of fine-tuning for this model and data regime.

Cross-Model Behavior Steering. Here we perform an ablation study analyzing the effect of steering a model towards behaviors other than those in its own Behavioral Strength Set. We use prompt injection to steer Llama-3.1-8b-Instruct, gpt-oss-20b, and Gemma 4 26B A4B with (i) its own Behavior Strength Set, (ii) each of the other models' sets, and (iii) the union of all behavior sets. We evaluate on the 200-example test split with the same single-gold EM and F1 as Table 1. This ablation is a dedicated cross-model evaluation run; consequently, the own-set cells may differ slightly from the corresponding prompt-injection entries in Table 1 due to run-to-run variation (e.g., hardware and decoding).

Table 3: Cross model behavior ablation study. Rows denote the target model, while columns denote the source Behavioral Strength Set used for steering. The three open-weight models are evaluated as targets; Claude Sonnet 3.7 is included only as a donor behavior set. Steering a model towards its own Behavioral Strength Set always achieves the highest average EM, and tends to achieve similar or better average F1.

Model / Behavior	Dataset	Gemma (EM / F1)	gpt-oss (EM / F1)	Llama (EM / F1)	Claude Sonnet (EM / F1)	Union (EM / F1)
Llama-3.1-8b-Instruct	HotpotQA	41.0 / 49.4	53.5 / 66.0	62.5 / 72.5	59.0 / 68.2	65.5 / 74.9
	MuSiQue	21.0 / 28.3	24.0 / 35.3	35.5 / 46.8	27.0 / 35.0	29.0 / 40.4
	RAMDocs	19.0 / 21.9	24.0 / 30.6	24.5 / 29.4	26.0 / 30.9	26.0 / 31.5
	Avg.	27.0 / 33.2	33.8 / 44.0	**40.8 / 49.6**	37.3 / 44.7	40.2 / 48.9
gpt-oss-20b	HotpotQA	44.5 / 54.9	63.5 / 75.5	9.5 / 18.3	65.5 / 78.1	30.0 / 39.0
	MuSiQue	38.0 / 52.6	41.0 / 59.3	10.0 / 18.9	43.5 / 60.9	29.0 / 37.9
	RAMDocs	4.5 / 15.1	36.5 / 45.8	4.5 / 14.8	22.0 / 36.9	13.0 / 24.4
	Avg.	29.0 / 40.9	**47.0 / 60.2**	8.0 / 17.3	43.7 / 58.6	24.0 / 33.8
Gemma 4 26B A4B	HotpotQA	84.0 / 93.3	78.0 / 90.9	81.5 / 92.7	82.0 / 92.5	81.0 / 92.1
	MuSiQue	71.5 / 81.3	69.0 / 80.8	71.0 / 81.8	71.0 / 81.9	72.5 / 82.4
	RAMDocs	35.5 / 43.4	37.5 / 44.5	21.0 / 35.0	35.0 / 45.8	36.0 / 45.2
	Avg.	**63.7** / 72.6	61.5 / 72.1	57.8 / 69.8	62.7 / **73.4**	63.2 / 73.2

As shown in Table 3, across all three steered models, steering a model towards its own Behavioral Strength Set achieves the highest average EM, and the highest average F1 for Llama-3.1-8b-Instruct and gpt-oss-20b models. Only for Gemma 4 does a different model's Behavioral Strength Set (Claude Sonnet) achieve a higher average F1 score. Furthermore, steering toward the union of all behaviors never outperforms a model's own set in average EM, while incurring significantly higher computation and inference costs due to longer prompt lengths.

These results support a model-specific view: each model tends to do best with the behaviors discovered from its own traces, and no single model's behavior set dominates the others.

6 Conclusion

We introduced Behavior Amplification, an approach for improving LLM reasoning by discovering model-specific behavioral strengths and amplifying them through targeted interventions. Our Behavioral Strength Discovery procedure identifies behaviors that occur significantly more often in correct traces than in incorrect traces, forming each model's Behavioral Strength Set. ISA then uses these behaviors as steering targets for prompt injection, behavioral reflection, SFT, and RL.

Across three model families and three multi-hop QA benchmarks, behavior-aware steering improves reasoning performance, but the best intervention varies by model. Instruction-expressible behaviors benefit most from inference-time prompting or reflection, whereas behaviors requiring multi-step execution derive greater benefit from parameter-level fine-tuning or reinforcement learning. These findings demonstrate that effective reasoning optimization requires aligning interventions with a model's specific behavioral profile rather than applying a universal strategy. Identifying and amplifying inherent model strengths provides a principled, model-aware path toward improving complex LLM reasoning.

AI USE STATEMENT

In this work, we did not use generative AI tools for the following required disclosure tasks:

- Help develop theoretical models or conceptual frameworks.
- Propose or refine hypotheses.
- Design or provide feedback on research methodology or experiments.
- Implement methods.
- Clean or reformat datasets.
- Support qualitative and thematic data analysis.
- Interpret results.

The following required disclosure tasks are not applicable to this work:

- Generate synthetic data sets.
- Formulate mathematical claims.
- Provide critical ingredients for proving mathematical claims.
- Assist in the writing of proofs.
- Assist with translation.

Additionally, we used generative AI tools for the following recommended disclosure tasks:

- Edit our paper to improve readability: We used generative AI to identify potential spelling and grammar errors, as well as opportunities to improve the style and tone of our paper. Changes proposed by generative AI were reviewed and approved or not approved by humans.
- Create or modify scientific figures or images: We used generative AI to help modify the LaTeX code that formats our tables and prompts.
- Propose a title or keywords for a research paper: We used generative AI to suggest titles for our paper. We modified those suggestions to arrive at our current title.

We have reviewed all AI-assisted work. We take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

All prompts used by our method are available in the appendix:

- Generation prompts: Figures 5 and 6
- Behavior Discovery prompt: Figure 7
- Behavior Annotation prompt: Figure 8
- Prompt injection prompt: Figure 10
- Reflection prompt: Figure 11

Additionally, Section B.2 details the generation arguments used for model inference in this paper. Section G in the appendix covers training details needed for reproducing our experiments with ISA-SFT and ISA-GRPO.

REFERENCES

[1] Anthropic. Claude 3.7 sonnet system card, 02 2025. URL https://www.anthropic.com/claude-3-7-sonnet-system-card.

[2] Darshan Deshpande, Varun Gangal, Hersh Mehta, Jitin Krishnan, Anand Kannappan, and Rebecca Qian. Trail: Trace reasoning and agentic issue localization. *arXiv preprint arXiv:2505.08638*, 2025.

[3] Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. Deep think with confidence. *arXiv preprint arXiv:2508.15260*, 2025.

[4] Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars, 2025. URL https://arxiv.org/abs/2503.01307.

[5] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[6] Aayush Karan and Yilun Du. Reasoning with sampling: Your base model is smarter than you think. *arXiv preprint arXiv:2510.14901*, 2025.

[7] Priyanka Kargupta, Shuyue Stella Li, Haocheng Wang, Jinu Lee, Shan Chen, Orevaoghene Ahia, Dean Light, Thomas L Griffiths, Max Kleiman-Weiner, Jiawei Han, et al. Cognitive foundations for reasoning and their manifestation in llms. *arXiv preprint arXiv:2511.16660*, 2025.

[8] Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think. *arXiv preprint arXiv:2504.16828*, 2025.

[9] Junsol Kim, Shiyang Lai, Nino Scherrer, James Evans, et al. Reasoning models generate societies of thought. *arXiv preprint arXiv:2601.10825*, 2026.

[10] Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4745–4759, 2025.

[11] Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. Llms get lost in multi-turn conversation. *arXiv preprint arXiv:2505.06120*, 2025.

[12] Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, Saurav Kadavath, Shannon Yang, Thomas Henighan, Timothy Maxwell, Timothy Telleen-Lawton, Tristan Hume, Zac Hatfield-Dodds, Jared Kaplan, Jan Brauner, Samuel R. Bowman, and Ethan Perez. Measuring faithfulness in chain-of-thought reasoning, 2023. URL https://arxiv.org/abs/2307.13702.

[13] Jinu Lee and Julia Hockenmaier. Evaluating step-by-step reasoning traces: A survey. *arXiv preprint arXiv:2502.12289*, 2025.

[14] Tongxu Luo, Wenyu Du, Jiaxi Bi, Stephen Chung, Zhengyang Tang, Hao Yang, Min Zhang, and Benyou Wang. Learning from peers in reasoning models, 2025. URL https://arxiv.org/abs/2505.07787.

[15] OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

[16] Anish Pahilajani, Devasha Trivedi, Jincen Shuai, Khin S Yone, Samyak Rajesh Jain, Namyong Park, Ryan A Rossi, Nesreen K Ahmed, Franck Dernoncourt, and Yu Wang. Grs-qa–graph reasoning-structured question answering dataset. *arXiv preprint arXiv:2411.00369*, 2024.

[17] Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv preprint arXiv:2410.08146*, 2024.

[18] Gemma Team. Gemma 4 model card, 2026. URL `https://ai.google.dev/gemma/docs/core/model_card_4`.

[19] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

[20] Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965, 2023.

[21] Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. Understanding reasoning in thinking language models via steering vectors. *arXiv preprint arXiv:2506.18167*, 2025.

[22] Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Retrieval-augmented generation with conflicting evidence. *arXiv preprint arXiv:2504.13079*, 2025.

[23] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[24] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

[25] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

[26] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

[27] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

[28] Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10495–10516, 2025.

[29] Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V Le, Denny Zhou, Swaroop Mishra, Huaixiu S Zheng, et al. Self-discover: Large language models self-compose reasoning structures. *Advances in Neural Information Processing Systems*, 37:126032–126058, 2024.

APPENDIX

A LIMITATIONS

In this paper, we evaluate our method on three natural language question-answering datasets: HotpotQA, MuSiQue, and RAMDocs. However, the reasoning behaviors that we discover for these datasets may differ when applied to other domains, such as math or coding. Addressing potential task-based differences in the relevant behavioral vocabulary is a natural next step for this research.

Our discovery procedure relies on LLM annotations of observable reasoning traces. These annotations provide operational labels for recurring trace patterns, but they do not guarantee access to the model's internal reasoning process. Future work could combine trace-level behavior discovery with human validation or mechanistic analyses to test whether the discovered behavioral signatures correspond to stable internal computation patterns.

Our Gemma-4 results suggest a tension between cross-task generalization and task-specific precision: behaviors that are strong predictors of success on one dataset may not be as helpful when applied to structurally different tasks. Computing $\Delta(b, m, d)$ per dataset, or clustering correct traces into latent task subtypes with their own Behavioral Strength Sets, are natural directions for future work.

Our ISA-SFT criterion is strict: a trace must be correct and exhibit all behaviors in $\mathcal{B}_m^+$. This produces a clean training signal, but may discard useful traces that express only part of the Behavioral Strength Set. Future work could replace this hard filter with weighted selection or soft behavioral scores.

B REASONING TRACE GENERATION

B.1 GENERATION PROMPTS

We use the prompts in Figure 5 and Figure 6 for reasoning trace generation. Since Llama-3.1-8b-Instruct does not generate reasoning natively, we include simple CoT instructions in its prompt. We then parse Llama's JSON output to recover the `reasoning` and `content` fields.

Figure 5: Non-Llama Prompt

```
First, consider the following context information.

<context>
{context}
</context>

Now, please answer the question below. Use the wording from the
    ↪ context (if appropriate) and as using as few words as
    ↪ possible.

<question>
{question}
</question>
```

B.2 GENERATION ARGUMENTS

All reasoning traces were generated with the Python OpenAI library using the `chat.completions` endpoint. The `kwargs` used for each model are in Table 4. Note that we use a temperature of 1 for Gemma 4 in accordance with best practices for this model [18].

C BEHAVIOR DISCOVERY

Figure 6: Llama CoT Prompt

```
First, consider the following context information:

<context>
{context}
</context>

Next, consider the following question:

<question>
{question}
</question>

Now, take a deep breath, and think step-by-step about how to
    ↪ answer this question. Format your response exclusively as
    ↪ a valid JSON object using the schema below (make sure your
    ↪ response begins with an open brace and ends with a close
    ↪ brace). Do not include any conversational filler outside
    ↪ the JSON.

JSON schema:

{
    "reasoning": "Your detailed step-by-step analysis of the
        ↪ problem.",
    "content": "Your final answer to the question, stated using
        ↪ the wording from the context (if appropriate) and
        ↪ using as few words as possible."
}
```

Figure 7: Behavior Discovery Prompt

```
System: You are an expert at analyzing LLM reasoning traces and
    ↪ identifying systematic reasoning patterns.

User: You are analyzing reasoning traces from a model on
    ↪ question answering tasks.
Below are reasoning traces labeled as correct or incorrect.

CORRECT TRACES:
{correct_traces}

INCORRECT TRACES:
{incorrect_traces}

Identify the 4-5 most distinctive reasoning behaviors that
    ↪ appear more frequently
in correct traces than incorrect traces.
For each behavior:
1. Give it a name (2-5 words, snake_case)
2. Define it in one sentence
3. Give 2 example phrases from the traces that signal its
    ↪ presence
4. Give 2 example phrases that signal its absence

Return JSON only, no other text:
{
"behaviors": [
  {
    "name": "behavior_name",
    "definition": "one sentence definition",
    "signal_phrases": ["phrase 1", "phrase 2"],
    "absence_phrases": ["phrase 1", "phrase 2"]
  }
]
}
```

Table 4: Model `kwargs` used for reasoning trace generation.

Models	kwargs
Llama-3.1-8b-Instruct	`{ "max_tokens": 8000, "temperature": 0.0 }`
gpt-oss-20b	`{ "max_tokens": 8000, "temperature": 0.0, "extra_body": { "reasoning_effort": "medium" } }`
Gemma 4	`{ "max_tokens": 8000, "temperature": 1.0, "top_p": 0.95, "extra_body": { "skip_special_tokens": False, "chat_template_kwargs": { "enable_thinking": True } } }`
Claude Sonnet 3.7	`{ "generate_kwargs": { "max_tokens": 4000, "thinking": { "type": "enabled", "budget_tokens": 2000 } } }`

The prompt used for behavior discovery is presented in Figure 7.

C.1 Behavior Discovery Resampling Stability

Over 1,000 bootstrap resamples of the discovery+train pool, every selected behavior is reselected in at least 74% of resamples, and strongly-selected behaviors ($\Delta \geq 0.15$) reselect in 95–100%. The two least-stable selected behaviors sit nearest the Δ=0.10 threshold and reselect 74–77%, while dropped behaviors reselect $\leq 2\%$ (the single borderline case, Δ=0.090, at 36%). The selected sets are therefore stable, with uncertainty confined to the threshold boundary as expected.

D Behavior Annotation

The prompt used for behavior annotation is in Figure 8.

Figure 8: Annotation Prompt

```
System: You are a careful annotator of LLM reasoning traces.

User: You are annotating a reasoning trace for the presence or
    ↪ absence of specific behaviors.
Answer YES or NO for each behavior.
Be conservative -- only mark YES if there is clear evidence in
    ↪ the trace.

BEHAVIORS:
{behavior_list}

TRACE:
{trace}

Return JSON only, no other text:
{"question_scoping": true/false,
    ↪ "stepwise_question_decomposition": true/false,
    ↪ "constraint_satisfaction_verification": true/false, ...}}
```

D.1 Behavior Annotation Ablation

We re-annotated all reasoning traces with an independent judge from a different model family (llama-3.3-70b-instruct) and compared them against our primary Opus annotations. Inter-annotator agreement is substantial overall: Cohen's k = 0.71. It is consistent across all four models, ranging from substantial to almost-perfect: sonnet-3.7 k = 0.81, gemma k = 0.71, llama-8B k = 0.71, gpt-oss-20B k = 0.57. Because the second judge is from a different family than both the primary judge and three of the four annotated models, this agreement is not an artifact of the Claude family; the judge also shows no family favoritism as its agreement on llama-8B traces (0.71) matches its agreement on the other models (0.71). For the selected Behavioral Strength Sets, we re-derived selection from the independent judge's labels (on discovery+train, with test held out). The strongly-selected behaviors are recovered in every case; the only variation occurs among behaviors sitting near the Δ=0.10 selection threshold which is consistent with our bootstrap resampling analysis, where near-threshold behaviors are the sole source of instability while strongly-selected ones reselect 95–100% of the time.

D.2 Prompt Robustness Ablation

To test whether the annotations depend on the exact wording of the annotation prompt, we re-annotated with the primary judge (Claude Opus 4.6) using a reworded and reordered prompt, shown in Figure 9. The instructions are rephrased and the trace/behavior blocks are presented in the opposite order, while the behavior definitions and output schema are held fixed, so prompt phrasing is the only variable.

Agreement between the original and reworded prompts is high: Cohen's k = 0.86, raw agreement 94.3% (n = 12,000 behavior judgments), uniformly across all four models (per-model k 0.81–0.91). The behavior labels are robust to prompt phrasing.

E Prompt Injection Prompt

The prompt used for prompt injection is presented in Figure 10, where `behavior_name_i` and `behavior_definition_i` are replaced with the name and definition of the ith extracted behavior for the prompted model and `signal_phrase_(i,j)` is replaced with the jth signal phrase of the ith extracted behavior.

Figure 9: Annotation Prompt Variant

```
Below is a reasoning trace. Your job is to decide, for each
    ↪ listed behavior, whether the trace clearly exhibits it.

Mark a behavior as present (true) only when the trace shows
    ↪ unambiguous evidence for it; otherwise mark it absent
    ↪ (false). When in doubt, mark absent.

Here is the trace you are judging:
{trace}

These are the behaviors to check for:
{behavior_list}

Respond with a single JSON object and nothing else:
{json_schema}
```

Figure 10: Prompt Injection Prompt

```
Before answering, make sure your reasoning explicitly
    ↪ demonstrates these behaviors:
  1. behavior_name_1: behavior_definition_1 (e.g. use phrases
      ↪ like: signal_phrase_(1, 1); signal_phrase_(1, 2))
  2. behavior_name_2: behavior_definition_2 (e.g. use phrases
      ↪ like: signal_phrase_(2, 1); signal_phrase_(2, 2))
  ...
```

F REFLECTION PROMPT

The reflection prompt presented in Figure 11 is added to the prompt instructions for the reflection pass, where `behavior_name_i` and `behavior_definition_i` are replaced with the name and definition of the ith extracted behavior for the prompted model and `signal_phrase_(i,j)` is replaced with the jth signal phrase of the ith extracted behavior.

Figure 11: Reflection Prompt

```
Before answering, make sure your reasoning explicitly
    ↪ demonstrates these behaviors:
  1. behavior_name_1: behavior_definition_1 (e.g. use phrases
      ↪ like: signal_phrase_(1, 1); signal_phrase_(1, 2))
  2. behavior_name_2: behavior_definition_2 (e.g. use phrases
      ↪ like: signal_phrase_(2, 1); signal_phrase_(2, 2))
  ...
```

G TRAINING DETAILS AND REPRODUCIBILITY

This section provides additional implementation details for the ISA-SFT and ISA-GRPO training procedures used in our experiments.

G.1 ISA-SFT TRAINING DETAILS

ISA-SFT uses supervised fine-tuning to increase the likelihood of reasoning traces that exhibit the model's Behavioral Strength Set. For each open-weight model, we construct the SFT dataset from training examples whose generated traces contain the selected behavioral strengths and lead to correct final answers.

Training uses LoRA adapters on top of the base model, with rank $r = 16$ and scaling factor $\alpha = 32$, applied to the query, key, value, and output projection matrices. We do not update the full model parameters. We train for 200 steps using the AdamW optimizer with learning rate 2×10^{-4} and a linear decay schedule with no warmup. The per-device batch size is 2 for Llama-3.1-8B-Instruct and 1 for larger models, with 4 gradient accumulation steps, giving an effective batch size of 8 and 4, respectively. All models are trained with bfloat16 mixed precision and a maximum sequence length of 2048 tokens. Gemma 4 26B A4B additionally uses gradient checkpointing due to its memory footprint.

The input-output format differs slightly across model families to match their expected generation style. For Llama-3.1-8B-Instruct, we use a JSON-style target format:

$$\{\texttt{"reasoning"}:\ldots,\texttt{"answer"}:\ldots\}.$$

For gpt-oss-20b and Gemma 4 26B A4B, we use a context-question template followed by free-form answer completion.

G.1.1 ISA-GRPO TRAINING DETAILS

For reinforcement learning, we use GRPO initialized from the corresponding ISA-SFT checkpoint. Training is performed on the 200-example training split. For each prompt, we sample $G = 2$ completions, compute per-completion rewards using Eq. 1, and update the policy with the GRPO objective. We use KL regularization against the ISA-SFT reference policy to discourage excessive drift from the supervised warm-start model.

We train for at most 200 optimization steps using the AdamW optimizer with learning rate 5×10^{-7} and a linear decay schedule with no warmup. The per-device batch size is 1 with 4 gradient accumulation steps, giving an effective batch size of 4. The maximum prompt length is 512 tokens

and the maximum completion length is 2048 tokens. A repetition penalty of 1.3 is applied during generation. Models are trained on 8×A100 40GB GPUs using Fully Sharded Data Parallel (FSDP). All runs use bfloat16 mixed precision.

H Intervention Selection and Additional Results

H.1 Discovery-Split Selection Grid

Table 1 reports, for each open-weight model-dataset pair, the single intervention chosen on the discovery split without access to test labels. Table 5 shows the evidence for that choice: EM and F1 for all four interventions evaluated on the discovery split (disjoint from test; SFT and GRPO train only on the separate train split, while prompt injection and reflection are training-free). For each pair we select the arg-max EM intervention (**bold**) and carry only that choice to the test set. The selected intervention is strongly model-specific—ISA-SFT for gpt-oss on every dataset, prompt injection or reflection for Gemma, and prompt injection or GRPO for Llama—and is chosen entirely on held-out discovery data.

Table 5: Discovery-split EM/F1 for all four interventions (open-weight models). The arg-max EM intervention per row (**bold**) is the one selected and carried to the test set in Table 1. Selection uses no test labels.

		Baseline		**Prompt Inj.**		**Reflection**		**ISA-SFT**		**GRPO**	
Model	**Dataset**	EM	F1	EM	F1	EM	F1	EM	F1	EM	F1
Llama-3.1-8b-Instruct	HotpotQA	36.0	53.6	**68.0**	78.8	64.0	78.4	66.0	79.1	63.0	76.4
	MuSiQue	33.0	48.1	**41.0**	46.7	35.0	46.5	36.0	49.9	29.0	42.8
	RAMDocs	29.0	46.2	29.0	36.7	33.0	42.0	29.0	44.5	**34.0**	48.3
gpt-oss-20b	HotpotQA	40.0	47.2	18.0	38.7	34.0	46.5	**44.0**	55.4	34.0	45.5
	MuSiQue	32.0	42.4	21.0	37.3	25.0	32.9	**41.0**	51.0	31.0	41.3
	RAMDocs	23.0	35.9	7.0	23.9	18.0	27.3	**42.0**	48.7	23.0	34.5
Gemma 4 26B A4B	HotpotQA	42.0	55.4	59.0	69.5	**65.0**	73.8	54.0	61.2	28.0	38.3
	MuSiQue	38.0	51.1	**47.0**	55.8	45.0	53.4	26.0	37.5	10.0	21.6
	RAMDocs	23.0	49.3	**36.0**	49.3	22.0	39.2	12.0	24.0	4.0	10.0

H.2 Confidence Intervals for Selected Interventions

Table 6 reports the full 95% bootstrap confidence intervals (percentile method, 10,000 resamples over the 200 test examples) for the label-free selected intervention summarized in Table 1, including the EM, F1, and the gain ΔEM over the no-intervention baseline. The intervention improves EM in all nine open-weight model-dataset pairs, and in seven of nine the ΔEM interval excludes zero (†); the two remaining pairs are a near-ceiling cell (Gemma/HotpotQA) and a small gain (Llama/MuSiQue), and neither shows a decrease.

H.3 Claude Sonnet 3.7 (Training-Free Interventions)

Claude Sonnet 3.7 is a closed-weight model that cannot be fine-tuned, so ISA-SFT and GRPO do not apply, and it is now deprecated, so we cannot generate the discovery-split predictions required for label-free selection. We therefore report it separately in Table 7, restricted to the two training-free interventions and with the intervention chosen as the test-best pick (not a label-free selection). Both training-free interventions improve over baseline; the MuSiQue and RAMDocs gains have ΔEM intervals excluding zero. These results are included for reference and are not part of the deployable label-free selection claim.

Table 6: 95% bootstrap confidence intervals (percentile method, 10,000 resamples) for the label-free selected intervention in Table 1. † marks a ΔEM interval excluding zero.

Model	Dataset	Selected	EM [95% CI]	F1 [95% CI]	ΔEM [95% CI]
Llama-3.1-8b-Instruct	HotpotQA	Prompt Inj.	62.0 [55.5, 68.5]	73.1 [67.6, 78.5]	$+12.5^{\dagger}$ [5.0, 20.0]
	MuSiQue	Prompt Inj.	36.5 [30.0, 43.5]	49.0 [42.7, 55.3]	$+2.5$ [-5.0, 9.5]
	RAMDocs	GRPO	47.5 [40.5, 54.5]	63.6 [58.0, 69.1]	$+20.0^{\dagger}$ [12.0, 28.0]
gpt-oss-20b	HotpotQA	ISA-SFT	78.0 [72.0, 83.5]	88.7 [85.0, 92.1]	$+16.0^{\dagger}$ [10.5, 22.0]
	MuSiQue	ISA-SFT	58.5 [52.0, 65.0]	73.1 [67.8, 78.3]	$+25.5^{\dagger}$ [18.0, 32.5]
	RAMDocs	ISA-SFT	60.0 [53.0, 66.5]	68.7 [62.9, 74.5]	$+41.0^{\dagger}$ [34.0, 48.0]
Gemma 4 26B A4B	HotpotQA	Reflection	81.5 [76.0, 87.0]	92.7 [89.9, 95.2]	$+3.5$ [0.0, 7.5]
	MuSiQue	Prompt Inj.	69.5 [63.0, 76.0]	79.9 [74.9, 84.6]	$+4.0^{\dagger}$ [0.5, 8.0]
	RAMDocs	Prompt Inj.	34.0 [27.5, 40.5]	43.5 [37.4, 49.5]	$+14.5^{\dagger}$ [7.0, 22.0]

Table 7: Claude Sonnet 3.7 test performance for the two applicable (training-free) interventions, with 95% bootstrap confidence intervals. The reported intervention is the test-best pick; **bold** marks the higher of the baseline and intervention EM (resp. F1).

Dataset	Selected	Base EM	Base F1	EM [95% CI]	F1 [95% CI]
HotpotQA	Prompt Inj.	69.5	81.7	**72.5** [66.0, 78.5]	**83.7** [79.2, 87.8]
MuSiQue	Reflection	55.0	70.3	**61.5** [54.5, 68.0]	**75.5** [70.3, 80.5]
RAMDocs	Reflection	27.5	49.1	**43.0** [36.0, 50.0]	**50.4** [44.2, 56.9]

I Computation Resources and Scalability

All fine-tuning experiments were conducted using a single node of 8 A100 40GB GPUs. Estimates of training time measured in GPU hours in Table 8.

Table 8: Number of GPU hours on 40GB A100s estimated for finetuning on each model/dataset pair.

Model	Dataset	GPU Hours	
		ISA-SFT	RL via GRPO
Llama-3.1-8b-Instruct	HotpotQA	40	120
	MuSiQue	40	160
	RAMDocs	40	160
gpt-oss-20b	HotpotQA	64	320
	MuSiQue	80	360
	RAMDocs	80	360
Gemma 4 26B A4B	HotpotQA	64	360
	MuSiQue	80	400
	RAMDocs	80	400

J ASSET LICENSES

Table 9: Licenses for assets used by this paper.

Asset	License
Llama-3.1-8b-Instruct [5]	Llama 3.1 Community License
gpt-oss-20b [15]	Apache 2.0
Gemma 4 [18]	Apache 2.0
Claude Sonnet 3.7 [1]	Commercial Proprietary
HotPotQA [24]	Apache 2.0
MuSiQue [19]	CC-BY 4.0
RAMDocs [22]	MIT

Our use of each model and dataset in Table 9 for scientific research is permitted under the corresponding license.