Metacognitive Behavioral Tuning of Large Language Models for Multi-Hop Question Answering

Ik-hwan Kim[*]
Dept. of ECE
Seoul Nat'l Univ.
Seoul, Korea

Hyeongrok Han[*]
Dept. of ECE
Seoul Nat'l Univ.
Seoul, Korea

Mingi Jung
Dept. of ECE
Seoul Nat'l Univ.
Seoul, Korea

Sangwon Yu
Dept. of ECE
Seoul Nat'l Univ.
Seoul, Korea

Jinseok Hong
AI Center
Samsung Elec.
Korea

Sang Hun Kim
AI Center
Samsung Elec.
Korea

Yoonyoung Choi
AI Center
Samsung Elec.
Korea

Sungroh Yoon[†]
AIIS, ASRI, INMC, ISRC,
Interdisc. Prog. in AI
Seoul Nat'l Univ.
Seoul, Korea

Abstract

Large Language Models (LLMs) often produce incorrect answers on multi-hop question answering even when the reasoning trace already contains a correct intermediate conclusion. We attribute this gap to weak self-regulation rather than insufficient reasoning capacity. Without explicit regulation, valid intermediate conclusions are overridden by continued exploration or left unrecognized as logically sufficient. We propose **M**etacognitive **B**ehavioral **T**uning (MBT), a post-training framework that injects a five-phase metacognitive structure into reasoning traces. The five phases are understanding and filtering, planning, execution and monitoring, self-correction, and verification. MBT has two formulations. MBT-S synthesizes new metacognitive traces from scratch, while MBT-R rewrites the student's own traces into a metacognitive form. Across HotpotQA, MuSiQue, and 2WikiMultiHopQA, MBT attains the highest Accuracy-Efficiency Score (AES) across model scales. MBT lifts task accuracy while keeping traces short and stable, with mean response length on MuSiQue an order of magnitude shorter than baseline methods and degeneration counts reduced by a similar margin. A matched-control study further confirms that the gain stems from the five-phase structural prior itself. To qualitatively assess the regulatory behavior of reasoning traces, we introduce two new metrics, the Reach-Redundancy Profile (RRP) and the length-aware Metacognitive Quality Index (MQI). RRP captures when the answer is reached and how much of the trace is redundant, and MQI quantifies how richly the five phases appear. Under both metrics, MBT achieves the earliest answer arrival, the lowest redundancy, and the richest phase-level behavior across model scales.

1 Introduction

Large Language Models (LLMs) have recently evolved into Large Reasoning Models (LRMs). LRMs extend chain-of-thought reasoning Wei et al. [2022] through explicit multi-step reasoning learned via large-scale reinforcement learning Xu et al. [2025a], Zhang et al. [2025b]. This shift admits test-time scaling, where accuracy improves as models allocate more computation to reasoning Snell et al. [2024], Agarwal et al. [2025a]. It has also produced gains on complex tasks, particularly in STEM domains Jaech et al. [2024], Guo et al. [2025], Yang et al. [2025], Bai et al. [2025b].

[*]Equal contribution.
[†]Corresponding author: sryoon@snu.ac.kr

Preprint.

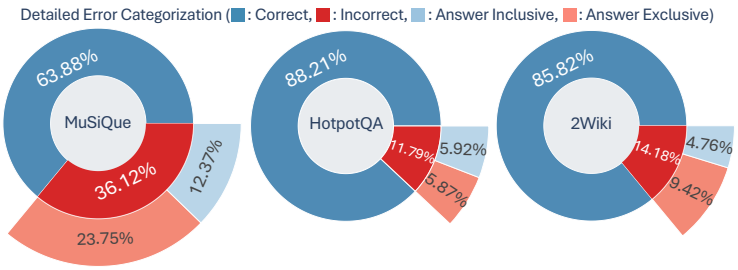

Figure 1: Prevalence of answer-inclusive errors on MHQA benchmarks under Qwen3-8B. Incorrect predictions split into Answer-Inclusive, where the gold answer was derived during reasoning but discarded, and Answer-Exclusive, where it was never derived. A large Answer-Inclusive share indicates that the gap to gold accuracy comes from failing to retain correct intermediate conclusions rather than from missing knowledge.

With explicit reasoning traces, hallucination can be analyzed at the level of intermediate steps. The standard definition of hallucination concerns fluent but factually incorrect content Ji et al. [2023]. LRMs expose chains of thought that make the underlying logic observable, which permits a process-centric analysis of failures and an extension of the hallucination notion from the outcome level to the process level. We find that many failures stem from weak logical control over the problem-solving process rather than from insufficient reasoning capacity. When this control breaks down, models produce invalid steps such as unsupported constraints or unwarranted derivations, which yield incorrect conclusions.

Our Findings We analyze reasoning traces on Multi-Hop Question Answering (MHQA) benchmarks and identify a recurring failure pattern. The model derives the correct answer mid-trace and then overrides it after introducing an unverified self-imposed constraint. A representative Qwen3-4B trace is shown in Figure A1 of Appendix A. The failure reflects absent metacognitive monitoring rather than insufficient reasoning capacity. The model does not check the validity of the new constraint before discarding a correct conclusion already present in the trace.

This pattern holds across MHQA benchmarks including MuSiQue Trivedi et al. [2022], HotpotQA Yang et al. [2018], and 2WikiMultiHopQA Ho et al. [2020]. We term such cases **answer-inclusive errors**. The gold answer or a paraphrase appears in the trace as an evidence-supported candidate, while the final output disagrees with it. An LLM-as-a-Judge filters incidental mentions by requiring the candidate to be coupled to supporting evidence. The operational definition, judge model, and prompts are in Appendix D and H.1. Figure 1 reports the prevalence of this pattern. A non-trivial fraction of failures fit this pattern and align with the **overthinking** and **underthinking** phenomena reported in mathematical reasoning Chen et al. [2024], Wang et al. [2025].

Existing post-training paradigms such as RLVR methods with outcome-based rewards Lambert et al. [2024], Shao et al. [2024] do not address this instability. When answer-inclusive errors are common, final-answer rewards penalize entire trajectories even when they contain valid intermediate steps Deng et al. [2025b,a]. Such objectives offer no signal for improving the stability of the reasoning process itself, leaving models exposed to the deviations described above.

We propose **M**etacognitive **B**ehavioral **T**uning (**MBT**), an automated post-training framework that injects structured metacognitive behaviors into reasoning traces. MBT constructs such traces either by synthesizing them from scratch or by rewriting the student's existing traces. The model is then fine-tuned on the resulting traces. By restricting the exploration space prior to reinforcement learning, MBT integrates structured metacognitive transitions and reduces redundant continuation. This keeps reasoning more stable under outcome-based optimization. Empirically, MBT matches or exceeds the accuracy of post-training baselines, reduces degeneration, and yields the highest accuracy-efficiency score in our evaluation. All code and data are released at `https://github.com/metacognitive-behavioral-tuning/MBT`.

Our main contributions are as follows.

- We identify a reasoning bottleneck in LRMs where valid intermediate conclusions are overridden under weak logical control, yielding answer-inclusive errors.

- We propose **Metacognitive Behavioral Tuning (MBT)**, a post-training framework that injects structured metacognitive behaviors and constrains the exploration space prior to reinforcement learning.
- We introduce two formulations for behavior injection. MBT-S synthesizes new traces and MBT-R rewrites the student's traces. Both improve the accuracy-efficiency trade-off across MHQA benchmarks.
- We propose two judge-based trace metrics beyond outcome accuracy. The Reach-Redundancy Profile (RRP) measures answer-arrival timing and trace-level redundancy. The Metacognitive Quality Index (MQI) measures five-phase richness.

2 Related Works

Foundations of Metacognition The framework that motivates MBT has a long history outside language modeling. Pólya's four-stage account of mathematical problem solving [Pólya, 1945] and Flavell's work on metacognition [Flavell, 1979] distinguished knowledge of one's own cognitive process from the act of monitoring it. Nelson and Narens [Nelson and Narens, 1990] formalized this as a meta-level and object-level loop. Brown's analyses of executive control [Brown, 1987] and the synthesis of Schraw and Moshman [Schraw and Moshman, 1995, Schraw and Dennison, 1994] consolidated the components of planning, monitoring, evaluation, and debugging. Schoenfeld [Schoenfeld, 1985, 1992] and Veenman *et al.* [Veenman et al., 2006] reported that the deliberate **control** of these components predicts problem-solving success beyond raw knowledge. Studies of self-explanation [Chi et al., 1989] and productive failure [Kapur, 2008, 2014] report that explicit intermediate monitoring and self-correction predict downstream transfer in human learners.

Metacognition in Language Models A line of work transfers these ideas into LLMs. Xiang et al. [2025] model reasoning as an internalized search and propose metacognitive intervention through inference-time guidance. Gandhi et al. [2025] identify discrete cognitive behaviors such as verification and backtracking as seeds that enable RL-driven self-improvement. Didolkar et al. [2024] and Kargupta et al. [2025] report that explicit metacognitive prompting improves LRM accuracy by eliciting under-utilized control behaviors. Zelikman et al. [2024] train models to internalize latent self-talk as part of next-token prediction. These works either intervene at inference time or target a single behavior. MBT injects an explicit five-phase structural prior at post-training so that planning, monitoring, self-correction, and verification are trained jointly into the student.

Cognitive Structures in Prompting A separate line of work imposes cognitive structure at inference time. System 2 Attention [Weston and Sukhbaatar, 2023], Plan-and-Solve [Wang et al., 2023a], Step-Back Prompting [Zheng et al., 2024], Skeleton-of-Thought [Ning et al., 2024], and Self-Discover [Zhou et al., 2024] decouple high-level structuring from execution. Self-Refine [Madaan et al., 2023], Reflexion [Shinn et al., 2023], and Chain-of-Verification [Dhuliawala et al., 2024] introduce iterative self-correction. Tree of Thoughts [Yao et al., 2023] and Self-Consistency [Wang et al., 2023b] treat reasoning as a search problem at decode time. These methods report accuracy gains under enforced structure. They share a limitation in that they operate at every inference call and so add test-time overhead. MBT moves the analogous structure into the model weights, removing the inference-time prompt or controller.

Efficient Reasoning Several recent methods target reasoning efficiency. Inference-time methods such as TokenSkip [Xia et al., 2025] compress traces by skipping low-information tokens. Data-centric approaches such as LIMOPro [Xiao et al., 2025] prune functional steps deemed redundant. RL-based approaches DLER [Liu et al., 2025] and ShorterBetter [Yi et al., 2025] include length penalties in the reward. Length-aware methods of this kind work when the underlying reasoning is already stable. Under multi-hop QA, however, our experiments in §4.4 show that aggressive pruning can yield higher degeneration counts. We treat efficiency and stability as complementary rather than substitutable goals. MBT obtains shorter traces as a byproduct of structured generation rather than through post-hoc compression.

Extended discussion of process supervision, reasoning distillation, and reasoning SFT and data curation appears in Appendix C.

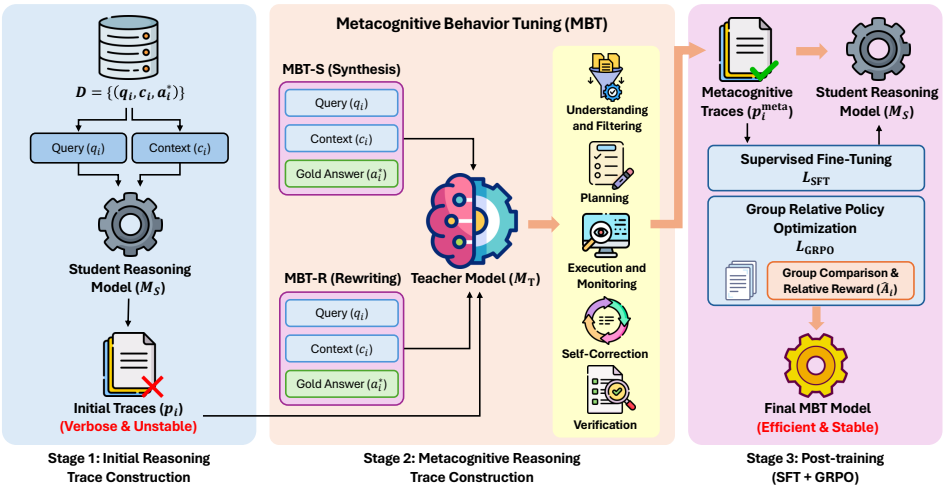

Figure 2: MBT framework. The pipeline runs in four stages. Stage 1 collects an initial student trace for MBT-R only. Stage 2 conditions a teacher on the gold answer to either synthesize a five-phase trace under MBT-S or restructure the student's initial trace into the same form under MBT-R. Stage 3 uses SFT to internalize the five-phase structure into the student. Stage 4 uses GRPO to refine the SFT-initialized policy.

3 Method

This section describes **M**etacognitive **B**ehavioral **T**uning (MBT), a post-training framework that injects metacognitive structure into reasoning traces. MBT has two data-construction formulations. **MBT-S** synthesizes traces from scratch with a teacher model. **MBT-R** rewrites the student's initial traces while preserving valid intermediate deductions. The pipeline runs in four stages. Stage 1 obtains an initial trace from the student (only for MBT-R). Stage 2 injects the metacognitive structure via teacher-guided synthesis or rewriting. Stage 3 internalizes the resulting traces through supervised fine-tuning. Stage 4 refines the SFT-initialized policy through Group Relative Policy Optimization (GRPO). An overview is given in Figure 2, and pseudo-code is provided in Algorithm 1.

3.1 Initial Reasoning Trace Generation

Let

$$\mathcal{D} = \{(q_i, c_i, a_i^*)\}_{i=1}^{N} \tag{1}$$

denote a MHQA dataset, where q_i is a query, c_i denotes the context, and a_i^* is the gold answer. The context c_i is either retrieved or given. Given an input (q, c), a student reasoning model M_{S} generates a reasoning trace p and a predicted answer $\hat{a}$,

$$(p, \hat{a}) \leftarrow M_{\mathrm{S}}(q, c). \tag{2}$$

3.2 Metacognitive Reasoning Trace Construction

Five Phase Rationale Pólya's four-stage model of mathematical problem solving [Pólya, 1945] separates understanding the problem, devising a plan, carrying out the plan, and looking back. Flavell [Flavell, 1979] introduced the parallel notions of metacognitive knowledge about goals, strategies, and constraints, and metacognitive monitoring as the online assessment of progress. Nelson and Narens [Nelson and Narens, 1990] proposed an explicit monitoring and control loop in which an upper-level process supervises a lower-level reasoning process. Schraw and Moshman [Schraw and Moshman, 1995] consolidated these accounts into planning, monitoring, and evaluation. Explicit debugging or self-correction is treated as a separable skill in the metacognitive awareness inventory [Schraw and Dennison, 1994]. Schoenfeld's analyses of mathematical problem solving [Schoenfeld, 1985] report that domain knowledge alone is not sufficient for performance. The **control** component covering the allocation of effort and the willingness to abandon unproductive lines is reported to separate successful from unsuccessful problem solvers. Work on student self-explanation [Chi et al., 1989] and on productive failure [Kapur, 2008] also reports that explicit

intermediate monitoring and self-correction predict downstream transfer. We take this as evidence that explicit metacognitive behaviors are inducible and that they are tied to learning outcomes.

We translate this framework into five phases for a reasoning trace, denoted p^{meta}. **(1) Understanding and Filtering** restates the goal and filters out irrelevant evidence, following Pólya's understanding the problem and Flavell's metacognitive knowledge. **(2) Planning** outlines a high-level strategy and decomposes the problem into sub-tasks, following Pólya's devising a plan and Schoenfeld's heuristics. **(3) Execution and Monitoring** carries out steps while continuously assessing the soundness of each transition, following Pólya's carrying out and the Nelson and Narens monitoring loop. **(4) Self-Correction** explicitly identifies and repairs logical errors, following Schraw and Moshman's debugging and Schoenfeld's control component. **(5) Verification** examines the conclusion against alternative scenarios before terminating, following Pólya's looking back and Schraw and Moshman's evaluation.

3.2.1 Metacognitive Trace Synthesis (MBT-S)

In the synthesis-based formulation, MBT-S constructs reasoning traces from scratch using a teacher model M_{T}. Given a query q_i, context c_i, and gold answer a_i^*, M_{T} generates a structured reasoning trace

$$p_i^{\text{meta}} \sim M_{\text{T}}(q_i, c_i, a_i^*). \tag{3}$$

MBT-S does not use the student's initial traces. It produces trajectories that follow the five-phase structure regardless of the student's initial policy. This makes MBT-S a cold-start formulation, useful when the student's native distribution is unstable.

3.2.2 Metacognitive Trace Rewriting (MBT-R)

In the rewriting-based formulation, MBT-R restructures the student's initial reasoning trace. Given an initial trace p_i produced by M_{S}, the teacher rewrites the trace conditioned on the query, context, and gold answer,

$$p_i^{\text{meta}} \sim M_{\text{T}}(q_i, c_i, p_i, a_i^*). \tag{4}$$

Unlike MBT-S, MBT-R uses the initial trace p_i to retain the student's exploration patterns under the five-phase structure.

We use a multi-turn prompting strategy. The teacher first receives (q_i, c_i, a_i^*) and produces a correct solution to anchor the structured reasoning path. The teacher then rewrites p_i on top of that solution. When p_i leads to an incorrect answer, the rewrite contains an explicit self-correction step that identifies the flaw, rejects the erroneous logic, and proceeds toward a_i^*. The output thus supplies supervision for both structural shaping and intermediate error repair. Prompt templates for synthesis and rewriting appear in Appendix H.2.

3.3 Post-training with Metacognitive Samples

The metacognitive traces $\mathcal{P}^{\text{meta}} = \{p_i^{\text{meta}}\}$ from the previous stage are used to train the student model M_{S}. Post-training (Stages 3 and 4) runs Supervised Fine-Tuning (SFT) to internalize the behavioral structure, then Group Relative Policy Optimization (GRPO) for further outcome-level optimization. Pseudo-code for the full pipeline appears as Algorithm 1 in Appendix B.

3.3.1 Supervised Fine-Tuning

We perform supervised fine-tuning on the metacognitive traces to install the five-phase reasoning structure. Let π_θ denote the student policy parameterized by θ. The SFT objective minimizes the negative log-likelihood of the metacognitive traces p^{meta},

$$\mathcal{L}_{\text{SFT}}(\theta) = -\mathbb{E}_{(q,p^{\text{meta}})}\left[\sum_t \log \pi_\theta(y_t \mid q, y_{<t})\right], \tag{5}$$

where $y = (y_1, \ldots, y_T)$ is the token sequence of p^{meta}. SFT shifts the student toward the regulated trajectories, so that the initial policy for the subsequent reinforcement learning stage already follows the five-phase structure.

Table 1: Accuracy across MHQA benchmarks. We report Exact Match (EM), F1, and an LLM-as-a-Judge score (LLM) computed with gemma-4-31b-it. **Bold** marks the best and underline the second-best per size, dataset, and metric cell.

Method	ID			OOD					
	HotpotQA			MuSiQue			2WikiMultiHopQA		
	EM↑	F1↑	LLM↑	EM↑	F1↑	LLM↑	EM↑	F1↑	LLM↑
Qwen3 0.6B									
Base	34.83	49.10	65.37	13.36	21.80	25.86	33.99	43.90	51.59
Prompt	32.72	46.51	64.05	12.54	20.17	24.41	29.64	39.63	49.84
GRPO	53.69	67.71	77.79	25.78	36.25	39.10	54.61	62.93	69.25
RS	37.29	52.15	68.70	14.03	22.65	27.97	36.23	47.04	55.42
TokenSkip	39.42	54.06	69.64	12.25	20.57	24.33	33.95	44.50	52.81
LIMOPro	45.33	63.17	83.89	24.66	34.23	39.10	32.88	48.30	68.39
MBT-S (Ours)	**62.71**	**76.89**	**86.28**	**35.75**	**44.56**	**47.33**	**57.78**	**67.05**	**73.32**
MBT-R (Ours)	61.59	75.60	84.85	32.93	42.28	45.14	57.21	66.35	73.17
Qwen3 1.7B									
Base	49.63	63.62	75.87	28.22	37.81	44.06	52.23	61.65	69.59
Prompt	41.36	56.10	80.12	25.28	34.36	44.19	42.46	55.63	74.19
GRPO	58.96	73.33	84.92	38.06	48.72	54.74	**63.25**	72.32	**80.69**
RS	53.83	68.63	81.66	30.99	41.66	48.86	58.06	68.26	76.75
TokenSkip	55.29	69.60	82.08	26.23	35.48	40.59	57.00	66.37	74.65
LIMOPro	47.66	65.85	88.05	28.13	38.07	46.59	39.15	54.70	76.43
MBT-S (Ours)	**66.37**	**80.20**	**89.60**	**42.82**	**52.77**	**55.98**	62.85	**72.37**	80.66
MBT-R (Ours)	64.89	78.75	88.22	39.59	50.24	53.99	62.24	72.22	80.62
Qwen3 4B									
Base	50.38	65.45	90.11	37.65	48.37	62.60	50.70	62.07	86.43
Prompt	50.56	65.16	89.74	35.13	45.16	61.40	49.20	60.21	86.27
GRPO	64.67	79.56	90.33	47.79	59.12	65.37	**69.30**	77.73	**87.53**
RS	50.28	65.31	89.63	37.53	48.00	61.36	50.30	61.59	85.53
TokenSkip	49.84	64.76	88.21	27.27	35.41	44.64	49.36	59.87	83.42
LIMOPro	49.59	67.64	90.71	31.57	41.42	50.89	41.01	56.29	80.57
MBT-S (Ours)	**68.66**	**82.26**	**91.84**	**52.34**	**61.96**	**66.61**	67.52	77.32	86.67
MBT-R (Ours)	68.22	82.06	91.40	51.92	61.56	66.40	68.26	**77.92**	87.25

3.3.2 Group Relative Policy Optimization

After SFT, we apply Group Relative Policy Optimization (GRPO) Shao et al. [2024] on the SFT-initialized checkpoint. GRPO removes the need for a separate value critic by estimating advantages from a group of outputs sampled per query. The trajectory-level reward is the F1 score of the predicted answer against the gold answer, and the trajectory-level advantage is the reward standardized within its group by the group mean and standard deviation. Because the preceding SFT stage restricts the rollout distribution to five-phase traces, GRPO operates within that region rather than re-exploring from an unstructured prior. Full notation, clipped surrogate, and training hyperparameters are in Appendix F.

4 Experiments

4.1 Experimental Setup

Datasets We evaluate MBT under both in-distribution (ID) and out-of-distribution (OOD) settings. We train only on the HotpotQA training set and evaluate on the validation sets of HotpotQA (ID), MuSiQue, and 2WikiMultiHopQA (OOD). The OOD sets have different reasoning structures and evidence compositions from HotpotQA.

Baselines We compare MBT against the following baselines. **Base** is vanilla Qwen3. **Prompt** elicits metacognitive behavior through a system prompt without parameter updates. **RS** is rejection-sampled self-distillation on correct HotpotQA traces. **GRPO** is direct GRPO with the same F1 outcome reward as MBT. **TokenSkip** Xia et al. [2025] and **LIMOPro** Xiao et al. [2025] are efficiency-oriented baselines re-implemented for MHQA following the original publications. Implementation details are in Appendix F. Decoding and training configurations are matched within each scale across methods.

4.2 Evaluation Metrics

We evaluate MBT along three axes. **Task accuracy** uses Exact Match (EM), F1, and an LLM-as-a-Judge score (LLM) from gemma-4-31b-it. The judge is drawn from a different family than the gpt-oss-120b teacher to avoid teacher-as-judge bias. **Inference-time efficiency** is measured by the

Table 2: Efficiency and stability on MuSiQue. **Degen** counts traces that saturate the 32,768-token decoding budget without producing a parseable answer, namely a repetitive-loop collapse. **Len** is the mean response length, with degenerated outputs counted at the same 32,768-token cap. **AES** denotes the Accuracy-Efficiency Score against the same-scale Base model. **Bold** marks the best and underline the second-best per size and metric cell.

Method	Qwen3-0.6B			Qwen3-1.7B			Qwen3-4B		
	Degen↓	Len↓	AES↑	Degen↓	Len↓	AES↑	Degen↓	Len↓	AES↑
Base	**0**	731	0.00	**1**	1186	0.00	2	1368	0.00
Prompt	**0**	697	-0.23	2	1134	0.05	5	1324	-0.06
GRPO	16	2167	-0.43	1	1379	0.56	73	2433	-0.65
RS	12	943	-0.04	19	1533	0.03	133	3097	-1.36
TokenSkip	174	3220	-3.70	471	7437	-5.66	651	9932	-7.69
LIMOPro	386	5697	-5.26	585	8393	-5.90	590	8478	-6.13
MBT-S (Ours)	6	**561**	**2.72**	2	**521**	**1.37**	**0**	**471**	**0.85**
MBT-R (Ours)	5	750	2.21	1	728	1.06	1	724	0.65

mean trace length (Len), the count of degeneration failures (Degen) where traces hit the 32,768-token cap, and the Accuracy-Efficiency Score (AES) [Luo et al., 2025]. **Regulatory quality** of the trace is measured by the two judge-based metrics defined below, replacing the length-only overthinking and underthinking proxies of prior work [Wang et al., 2025, Chen et al., 2024]. Judge details and AES weighting are in Appendix F.

Reach-Redundancy Profile (RRP) For each of the n evaluation samples ($i = 1, \ldots, n$), the judge labels every paragraph of the reasoning trace τ_i as PROGRESS, VERIFICATION, or REDUNDANT. The judge returns the index k_i of the first paragraph at which the gold answer is explicitly derived, or the paragraph count N_i if the answer is never reached. The judge also returns the total sentence count r_i of paragraphs labeled REDUNDANT. With $s_{i,j}$ the sentence count of paragraph j and $T_i = \sum_{j=1}^{N_i} s_{i,j}$, the arrival position and redundancy fraction are

$$\rho_i = \frac{1}{T_i} \sum_{j=1}^{k_i} s_{i,j}, \qquad \delta_{r,i} = \frac{r_i}{T_i}, \tag{6}$$

both clamped to $[0, 1]$. The score $\mathcal{R}_i$ is the harmonic mean of $1 - \rho_i$ and $1 - \delta_{r,i}$, following Van Rijsbergen's E-measure [Van Rijsbergen, 1979]. Length-aware variants $\rho_i^{\text{la}}, \delta_{r,i}^{\text{la}}$ used for cross-method comparison are defined in Appendix H.3 as Eq. 11.

Metacognitive Quality Index (MQI) The judge returns a holistic rubric level $L_i^{\text{obs}} \in \{0, \ldots, 5\}$ and the subset $\mathcal{S}_i$ of the five phases (§3.2) it identifies in τ_i. Since a high rubric level over a much longer trace trades phase richness for inference cost, we define MQI as a length-aware combination,

$$\text{MQI}_i \;=\; L_i^{\text{obs}} \cdot \frac{T_{\text{base}}}{T_i}, \qquad \overline{\text{MQI}} \;=\; \frac{1}{n} \sum_{i=1}^{n} \text{MQI}_i, \tag{7}$$

where T_i is sample i's trace length and T_{base} is the same-scale Qwen3 base model's mean trace length over valid non-degenerated outputs on the evaluation benchmark. MQI_i is large only when the rubric level is high and the trace is short. Per-phase presence ratios are reported in Appendix H.3.

4.3 Multi-Hop QA Reasoning Performance

Table 1 shows that MBT obtains the highest accuracy on HotpotQA and MuSiQue across the three model scales. Each baseline trails on at least one axis. Prompt-based elicitation (Base+Prompt) often degrades on OOD sets. Standard GRPO improves ID accuracy but transfers unevenly to MuSiQue. Rejection Sampling produces small gains without consistent OOD improvement. Efficiency-oriented methods such as TokenSkip and LIMOPro fall behind on OOD, where trace compression appears to remove steps needed for multi-hop reasoning. The MuSiQue gain is of interest because no MuSiQue example appears in training. At 4B, MBT's LLM-as-a-Judge gain is larger on the harder OOD MuSiQue than on the ID HotpotQA. We read this as evidence that MBT internalizes a regulatory mechanism that transfers under distribution shift rather than only fitting the training set.

4.4 Reasoning Efficiency and Stability

Table 2 reports the limitations of each baseline. Rejection Sampling produces longer traces without matching accuracy gains. The heuristic pruning of TokenSkip and LIMOPro yields incoherent traces

Table 3: Controlled comparison on MuSiQue with Qwen3-4B. The top section shares the same student-trace origin and varies only by whether 5-phase rewriting is applied. The bottom section shares the same teacher-trace origin and varies only by whether the 5-phase scaffolding is applied. **Bold** marks the best and underline the second-best per section and metric.

Method	EM↑	F1↑	LLM↑	Degen↓	Len↓	AES↑
Student-trace origin, control is rejection-sampled self-distillation						
RS (self-distill + SFT)	37.53	48.00	61.36	133	3097	-1.36
MBT-R + SFT	**48.03**	**57.90**	**62.72**	**3**	**745**	**+0.46**
Teacher-trace origin, control is naive teacher distillation						
gpt-oss-distill + SFT	39.93	52.12	66.24	304	5383	-2.76
gpt-oss-distill + SFT + GRPO	**53.58**	**63.02**	**69.18**	285	5437	-2.66
MBT-S + SFT	49.65	59.54	64.54	**0**	478	+0.74
MBT-S + SFT + GRPO	52.34	61.96	66.61	**0**	**471**	**+0.85**

and high degeneration counts. Standard GRPO also suffers from degeneration, especially at the 4B scale. MBT-S and MBT-R produce degeneration counts an order of magnitude below TokenSkip and LIMOPro. By installing the metacognitive structure during training rather than imposing external length constraints, MBT yields shorter traces without sacrificing accuracy. MBT also obtains the highest AES at every model scale. Qualitative examples appear in Appendix G.

5 Discussion

We analyze MBT on Qwen3-4B with MuSiQue. §5.1 isolates the contribution of the five-phase structural prior against matched controls without it. §5.2 asks whether MBT changes the shape of the reasoning trajectory. §5.3 asks whether the framework induces identifiable phase-level behavior.

5.1 Controlled Comparison of Five Phase Contribution

To separate the contribution of the five-phase scaffolding, we run two matched controls in which the pipeline and trace source are fixed and only the structural prior is removed.

Student Trace Control (RS) The matched control for MBT-R is rejection-sampled self-distillation (RS). Both methods fine-tune the student on its own traces. RS retains only the subset of traces with a correct answer. MBT-R rewrites every trace into the five-phase form and adds an explicit self-correction step when the original trace was incorrect (§3.2). The training input distributions differ only in the five-phase scaffolding and in MBT-R's recovery from incorrect drafts. On Qwen3-4B with MuSiQue (Table 3), MBT-R + SFT improves over RS by +10.50 EM, +9.90 F1, and +1.36 LLM while reducing mean response length and degeneration count. Both methods fine-tune on the same pool of student-generated traces, so the gap isolates the effect of the five-phase rewriting.

Teacher Trace Control (gpt-oss-distill) The matched control for MBT-S is naive teacher distillation. The teacher gpt-oss-120b and the SFT and GRPO pipeline are identical. gpt-oss-distill rejection-samples the teacher's unconditional rollouts on gold-EM, whereas MBT-S synthesizes traces conditioned on the gold answer under the five-phase prompt and needs no such filter. Naive teacher distillation has the highest raw accuracy but pays a stability cost of hundreds of degenerated outputs and a negative AES. MBT-S + SFT + GRPO comes within a small margin of this accuracy with zero degenerated outputs and an order of magnitude shorter response. It also obtains the highest AES of the post-training methods we evaluate. The five-phase scaffolding trades a small accuracy margin for a different operating point on the accuracy-efficiency frontier.

5.2 Reasoning Trajectory Reshaping

We examine **when** the answer appears in the trace and **how much of the trace makes neither progress nor verification**, using the length-aware RRP of §4.2. Figure 3 (left) plots each method on the RRP plane. Length-aware arrival position ρ^{la} is on one axis and redundancy fraction δ_r^{la} on the other, both anchored at the same-scale Qwen3 base model's mean trace length. The methods separate into three regions matching the failure modes of §1. Base, Prompt, GRPO, RS, and TokenSkip sit at high ρ^{la} and high δ_r^{la}, indicating late arrival followed by unstructured continuation. LIMOPro reaches the lower-left through post-hoc step pruning rather than learned regulation. Its low ρ^{la} co-occurs

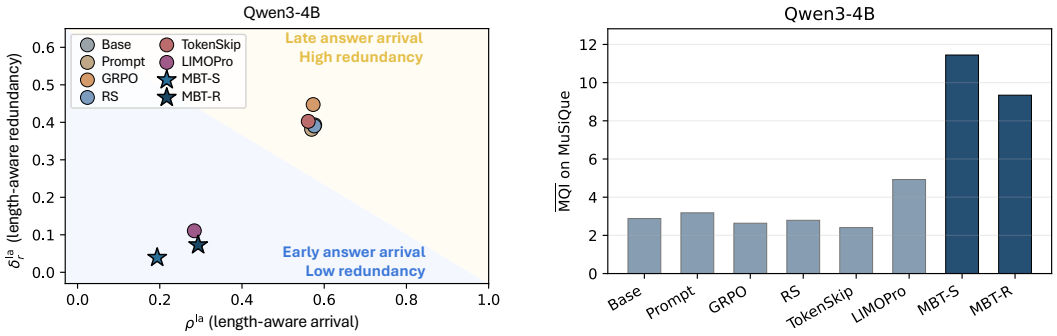

Figure 3: Behavioral analysis on Qwen3-4B with MuSiQue. **Left** shows the RRP plane plotting length-aware arrival ρ^{la} against redundancy δ_r^{la}. MBT-S and MBT-R cluster at the lower-left, while baselines sit at high-ρ or high-δ_r. **Right** shows length-aware $\overline{\text{MQI}}$. MBT-S is highest, MBT-R is second, and the baselines are below. Full 0.6B, 1.7B, and 4B variants for both panels appear in Figures A3 and A4.

with near-absent Self-Correction in §5.3, so the position reflects compression rather than monitoring. MBT-S and MBT-R also occupy the lower-left, with the lowest redundancy fractions and a phase profile that retains Self-Correction and Verification. The same clustering holds at 0.6B and 1.7B in Figure A3. MBT derives the answer earlier in absolute reasoning effort while keeping the trace shorter and with less redundancy. This is the regulation pattern the framework is designed to produce.

5.3 Phase Level Metacognitive Behavior

We ask whether the regulation pattern is tied to identifiable five-phase behavior, using the length-aware MQI of §4.2. MQI rates each trace on the presence and integration of the five phases Understanding and Filtering, Planning, Execution and Monitoring, Self-Correction, and Verification. The metric is length-aware, so longer traces are not credited for accumulating more phase mentions. Figure 3 (right) reports $\overline{\text{MQI}}$ on Qwen3-4B, and the same ordering holds at 0.6B and 1.7B in Figure A4. MQI separates the methods, with MBT leading at every scale. The other methods including the efficiency-oriented baseline LIMOPro sit below. The per-phase breakdown in Table A8 localizes the gap. Baseline traces include Understanding and Filtering but contain Self-Correction and Verification only sporadically. The phases MBT supplies are the regulatory ones, so the MQI gain reflects phase-level behavior rather than a surface-level byproduct of SFT on structured traces.

6 Conclusion

Failure analysis of LRMs on multi-hop QA shows that errors arise less from limited capacity than from weak monitoring of the trajectory. In the absence of explicit regulation, valid intermediate conclusions are overridden by continued exploration. We propose Metacognitive Behavioral Tuning (MBT), a post-training framework with two formulations, MBT-S and MBT-R, that impose a five-phase structure based on accounts of self-regulated problem solving from cognitive psychology. Across HotpotQA, MuSiQue, and 2WikiMultiHopQA, MBT obtains the highest accuracy-efficiency score and reduces degeneration. A matched distillation control together with the RRP and MQI analysis attributes the gain to the structural prior. MBT also produces measurable phase-level changes in reasoning behavior.

Limitations Our evaluation is scoped to retrieval-grounded multi-hop QA. Transferring the five-phase decomposition to mathematics or programming would require domain-adapted phase definitions [Pólya, 1945, Schoenfeld, 1985]. Whether a model can discover an equivalent decomposition through self-supervised search is open. Accuracy and behavioral metrics rely on a single judge family gemma-4-31b-it from a different lineage than the gpt-oss-120b teacher. This avoids the self-preference effect that arises when judge and teacher share a family, but a single judge can still introduce idiosyncratic scoring bias in absolute terms. We therefore draw conclusions from relative rankings between methods rather than from raw scores. MBT-S and MBT-R currently use a strong teacher for trace construction. Two follow-ups remain. The first is constructing metacognitive traces

without such a teacher, for example through student-self-generated or bootstrapped traces. The second is testing whether MBT continues to give gains on base models larger than the 0.6B to 4B Qwen3 backbones studied here.

Societal Impacts By enforcing planning, monitoring, and verification, MBT produces traces that are more auditable and reduces inference-time token cost. MBT does not address safety alignment, so deployment should pair MBT with alignment training and human oversight.

Acknowledgments and Disclosure of Funding

This work was supported by Samsung Electronics CO., Ltd (IO250624-13143-01), Institute of Information Communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) [(No.2022-0-00959, RS-2022-II220959), NO.RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)], the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A3B1077720), and the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2026.

References

Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, et al. Phi-4-reasoning technical report. *arXiv preprint arXiv:2504.21318*, 2025.

Aradhye Agarwal, Ayan Sengupta, and Tanmoy Chakraborty. The art of scaling test-time compute for large language models. *arXiv preprint arXiv:2512.02008*, 2025a.

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025b.

John Alderete, Sebastian Benthal, Connie Xu, and John Xing. Separable pathways for causal reasoning: How architectural scaffolding enables hypothesis-space restructuring in LLM agents. *arXiv preprint arXiv:2604.20039*, 2026.

Haoyue Bai, Yiyou Sun, Wenjie Hu, Shi Qiu, Maggie Ziyu Huan, Peiyang Song, Robert Nowak, and Dawn Song. How and why LLMs generalize: A fine-grained analysis of LLM reasoning from cognitive behaviors to low-level patterns. *arXiv preprint arXiv:2512.24063*, 2025a.

Yifan Bai, Yiping Bao, Y. Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025b.

Siddhant Bhambri, Upasana Biswas, and Subbarao Kambhampati. Interpretable traces, unexpected outcomes: Investigating the disconnect in trace-based knowledge distillation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2026.

Ann L Brown. Metacognition, executive control, self-regulation, and other more mysterious mechanisms. In Franz E Weinert and Rainer H Kluwe, editors, *Metacognition, Motivation, and Understanding*, pages 65–116. Lawrence Erlbaum Associates, Hillsdale, NJ, 1987.

Dan Busbridge, Amitis Shidani, Floris Weers, Jason Ramapuram, Etai Littwin, and Russ Webb. Distillation scaling laws. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

Hongyi James Cai, Junlin Wang, Xiaoyin Chen, and Bhuwan Dhingra. How much backtracking is enough? exploring the interplay of SFT and RL in enhancing LLM reasoning. *arXiv preprint arXiv:2505.24273*, 2025.

Hanting Chen, Yasheng Wang, Kai Han, Dong Li, Lin Li, Zhenni Bi, Jinpeng Li, Haoyu Wang, Fei Mi, Mingjian Zhu, et al. Pangu embedded: An efficient dual-system LLM reasoner with metacognition. *arXiv preprint arXiv:2505.22375*, 2025a.

Wei-Rui Chen, Vignesh Kothapalli, Ata Fatahibaarzi, Hejian Sang, Shao Tang, Qingquan Song, Zhipeng Wang, and Muhammad Abdul-Mageed. Distilling the essence: Efficient reasoning distillation via sequence truncation. *arXiv preprint arXiv:2512.21002*, 2025b.

Xinghao Chen, Zhijing Sun, Wenjin Guo, Miaoran Zhang, Yanjun Chen, Yirong Sun, Hui Su, Yijie Pan, Dietrich Klakow, Wenjie Li, and Xiaoyu Shen. Unveiling the key factors for distilling chain-of-thought reasoning. *arXiv preprint arXiv:2502.18001*, 2025c.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024.

Michelene T H Chi, Miriam Bassok, Matthew W Lewis, Peter Reimann, and Robert Glaser. Self-explanations: How students study and use examples in learning to solve problems. *Cognitive Science*, 13(2):145–182, 1989.

Wenlong Deng, Yushu Li, Boying Gong, Yi Ren, Christos Thrampoulidis, and Xiaoxiao Li. On group relative policy optimization collapse in agent search: The lazy likelihood-displacement. *arXiv preprint arXiv:2512.04220*, 2025a.

Wenlong Deng, Yi Ren, Muchen Li, Danica J Sutherland, Xiaoxiao Li, and Christos Thrampoulidis. On the effect of negative gradient in group relative deep reinforcement optimization. *arXiv preprint arXiv:2505.18830*, 2025b.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. Chain-of-verification reduces hallucination in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3563–3578, 2024.

Aniket Didolkar, Anirudh Goyal, Nan Rosemary Ke, Siyuan Guo, Michal Valko, Timothy Lillicrap, Danilo Rezende, Yoshua Bengio, Michael Mozer, and Sanjeev Arora. Metacognitive capabilities of LLMs: An exploration in mathematical problem solving. *arXiv preprint arXiv:2405.12205*, 2024.

Aniket Didolkar, Nicolas Ballas, Sanjeev Arora, and Anirudh Goyal. Metacognitive reuse: Turning recurring LLM reasoning into concise behaviors. *arXiv preprint arXiv:2509.13237*, 2025.

Jiayu Ding, Lei Cui, Li Dong, Nanning Zheng, and Furu Wei. Information-preserving reformulation of reasoning traces for antidistillation. *arXiv preprint arXiv:2510.11545*, 2025.

Haonan Dong, Haoran Ye, Wenhao Zhu, Kehan Jiang, and Guojie Song. Meta-R1: Empowering large reasoning models with metacognition. *arXiv preprint arXiv:2508.17291*, 2025.

Abraham Paul Elenjical, Vivek Hruday Kavuri, and Vasudeva Varma. Think^2: Grounded metacognitive reasoning in large language models. *arXiv preprint arXiv:2602.18806*, 2026.

John H Flavell. Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. *American Psychologist*, 34(10):906–911, 1979.

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. *arXiv preprint arXiv:2503.01307*, 2025.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: On-policy distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, et al. OpenThoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, 2020.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. V-STaR: Training verifiers for self-taught reasoners. In *Conference on Language Modeling (COLM)*, 2024.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, 2023.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, 2023. doi: 10.1145/3571730.

Hoang Anh Just, Myeongseob Ko, and Ruoxi Jia. The signal is in the steps: Local scoring for reasoning data selection. *arXiv preprint arXiv:2510.03988*, 2025.

Manu Kapur. Productive failure. *Cognition and Instruction*, 26(3):379–424, 2008.

Manu Kapur. Productive failure in learning math. *Cognitive Science*, 38(5):1008–1022, 2014.

Priyanka Kargupta, Shuyue Stella Li, Haocheng Wang, Jinu Lee, Shan Chen, Orevaoghene Ahia, Dean Light, Thomas L. Griffiths, Max Kleiman-Weiner, Jiawei Han, Asli Celikyilmaz, and Yulia Tsvetkov. Cognitive foundations for reasoning and their manifestation in LLMs. *arXiv preprint arXiv:2511.16660*, 2025.

Yoonjeon Kim, Doohyuk Jang, and Eunho Yang. Meta-awareness enhances reasoning models: Self-alignment reinforcement learning. *arXiv preprint arXiv:2510.03259*, 2025.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards streamlined distillation for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Dacheng Li, Shiyi Cao, Tyler Griggs, Shu Liu, Xiangxi Mo, Eric Tang, Sumanth Hegde, Kourosh Hakhamaneshi, Shishir G. Patil, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. LLMs can easily learn to reason from demonstrations structure, not content, is what matters! *arXiv preprint arXiv:2502.07374*, 2025a.

Yang Li, Youssef Emad, Karthik Padthe, Jack Lanchantin, Weizhe Yuan, Thao Nguyen, Jason Weston, Shang-Wen Li, Dong Wang, Ilia Kulikov, and Xian Li. NaturalThoughts: Selecting and distilling reasoning traces for general reasoning tasks. *arXiv preprint arXiv:2507.01921*, 2025b.

Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. Small models struggle to learn from strong reasoners. *arXiv preprint arXiv:2502.12143*, 2025c.

Zhaoyi Li, Xiangyu Xi, Zhengyu Chen, Wei Wang, Gangwei Jiang, Ranran Shen, Linqi Song, Ying Wei, and Defu Lian. On the role of reasoning patterns in the generalization discrepancy of long chain-of-thought supervised fine-tuning. *arXiv preprint arXiv:2604.01702*, 2026.

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

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. DLER: Doing length penalty right—incentivizing more intelligence per token via reinforcement learning. *arXiv preprint arXiv:2510.15110*, 2025.

Ximing Lu, Seungju Han, David Acuna, Hyunwoo Kim, Jaehun Jung, Shrimai Prabhumoye, Niklas Muennighoff, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, and Yejin Choi. Retro-search: Exploring untaken paths for deeper and efficient reasoning. *arXiv preprint arXiv:2504.04383*, 2025.

Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. *arXiv preprint arXiv:2501.12570*, 2025.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36: 46534–46594, 2023.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of GPT-4. *arXiv preprint arXiv:2306.02707*, 2023.

Thomas O Nelson and Louis Narens. Metamemory: A theoretical framework and new findings. In Gordon H Bower, editor, *The Psychology of Learning and Motivation*, volume 26, pages 125–173. Academic Press, 1990.

Xuefei Ning, Zinan Lin, Zixuan Zhou, Zifu Wang, Huazhong Yang, and Yu Wang. Skeleton-of-thought: Prompting LLMs for efficient parallel generation. In *The Twelfth International Conference on Learning Representations*, 2024.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. LLMLingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 963–981. Association for Computational Linguistics, 2024.

George Pólya. *How to Solve It: A New Aspect of Mathematical Method*. Princeton University Press, 1945.

Tian Qin, David Alvarez-Melis, Samy Jelassi, and Eran Malach. To backtrack or not to backtrack: When sequential search limits model reasoning. In *Conference on Language Modeling (COLM)*, 2025.

Alan H Schoenfeld. *Mathematical Problem Solving*. Academic Press, 1985.

Alan H Schoenfeld. Learning to think mathematically: Problem solving, metacognition, and sense making in mathematics. In Douglas A Grouws, editor, *Handbook of Research on Mathematics Teaching and Learning*, pages 334–370. Macmillan, 1992.

Gregory Schraw and Rayne Sperling Dennison. Assessing metacognitive awareness. *Contemporary Educational Psychology*, 19(4):460–475, 1994.

Gregory Schraw and David Moshman. Metacognitive theories. *Educational Psychology Review*, 7 (4):351–371, 1995.

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

Yiyang Shen, Lifu Tu, and Weiran Wang. Reinforcement learning-based knowledge distillation with LLM-as-a-judge. *arXiv preprint arXiv:2604.02621*, 2026.

Zhanming Shen, Zeyu Qin, Zenan Huang, Hao Chen, Jiaqi Hu, Yihong Zhuang, Guoshan Lu, Gang Chen, and Junbo Zhao. Merge-of-thought distillation. *arXiv preprint arXiv:2509.08814*, 2025.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, pages 1279–1297. ACM, March 2025. doi: 10.1145/3689031.3696075. URL `http://dx.doi.org/10.1145/3689031.3696075`.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

Shaltiel Shmidman, Asher Fredman, Oleg Sudakov, and Meriem Bendris. Learning to reason: Training LLMs with GPT-OSS or DeepSeek-R1 reasoning traces. *arXiv preprint arXiv:2511.19333*, 2025.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Yiyou Sun, Georgia Zhou, Haoyue Bai, Hao Wang, Dacheng Li, Nouha Dziri, and Dawn Song. Climbing the ladder of reasoning: What LLMs can—and still can't—solve after SFT? *arXiv preprint arXiv:2504.11741*, 2025.

Xue Wen Tan, Nathaniel Tan, Galen Lee, and Stanley Kok. The shape of reasoning: Topological analysis of reasoning traces in large language models. *arXiv preprint arXiv:2510.20665*, 2025.

Yijun Tian, Yikun Han, Xiusi Chen, Wei Wang, and Nitesh V. Chawla. Beyond answers: Transferring reasoning capabilities to smaller LLMs using multi-teacher knowledge distillation. In *Proceedings of the 18th ACM International Conference on Web Search and Data Mining (WSDM)*. ACM, 2025.

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.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

C. J. Van Rijsbergen. *Information Retrieval*. Butterworths, London, 2nd edition, 1979.

Marcel V J Veenman, Bernadette H A M Van Hout-Wolters, and Peter Afflerbach. Metacognition and learning: Conceptual and methodological considerations. *Metacognition and Learning*, 1(1): 3–14, 2006.

Yuxuan Wan, Tianqing Fang, Zaitang Li, Yintong Huo, Wenxuan Wang, Haitao Mi, Dong Yu, and Michael R. Lyu. Inference-time scaling of verification: Self-evolving deep research agents via test-time rubric-guided verification. In *Findings of the Association for Computational Linguistics (ACL Findings)*, 2026.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada, 2023a. Association for Computational Linguistics.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 9426–9439, 2024.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023b.

Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Thoughts are all over the place: On the underthinking of long reasoning models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Yuqing Wang and Yun Zhao. Metacognitive prompting improves understanding 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 1914–1926, Mexico City, Mexico, 2024. Association for Computational Linguistics.

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

Zihao Wei, Liang Pang, Jiahao Liu, Wenjie Shi, Jingcheng Deng, Shicheng Xu, Zenghao Duan, Fei Sun, Huawei Shen, and Xueqi Cheng. The evolution of thought: Tracking LLM overthinking via reasoning dynamics analysis. *arXiv preprint arXiv:2508.17627*, 2025.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, Haosheng Zou, Yongchao Deng, Shousheng Jia, and Xiangzheng Zhang. Light-R1: Curriculum SFT, DPO and RL for long CoT from scratch and beyond. *arXiv preprint arXiv:2503.10460*, 2025.

Jason Weston and Sainbayar Sukhbaatar. System 2 attention (is something you might need too). *arXiv preprint arXiv:2311.11829*, 2023.

Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Understanding chain-of-thought length in LLMs. *arXiv preprint arXiv:2502.07266*, 2025.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. TokenSkip: Controllable chain-of-thought compression in LLMs. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3351–3363, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.165. URL https://aclanthology.org/2025.emnlp-main.165/.

Violet Xiang, Charlie Snell, Kanishk Gandhi, Alon Albalak, Anikait Singh, Chase Blagden, Duy Phung, Rafael Rafailov, Nathan Lile, Dakota Mahan, Louis Castricato, Jan-Philipp Franken, Nick Haber, and Chelsea Finn. Towards system 2 reasoning in llms: Learning how to think with meta chain-of-thought. *arXiv preprint arXiv:2501.04682*, 2025.

Yang Xiao, Jessie Wang, Ruifeng Yuan, Chunpu Xu, Kaishuai Xu, Wenjie Li, and Pengfei Liu. LIMOPro: Reasoning refinement for efficient and effective test-time scaling. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=W9Y0jtf45v.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, et al. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*, 2025a.

Haoran Xu, Baolin Peng, Hany Awadalla, Dongdong Chen, Yen-Chun Chen, Mei Gao, Young Jin Kim, Yunsheng Li, Liliang Ren, Yelong Shen, Shuohang Wang, Weijian Xu, Jianfeng Gao, and Weizhu Chen. Phi-4-mini-reasoning: Exploring the limits of small reasoning language models in math. *arXiv preprint arXiv:2504.21233*, 2025b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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*, pages 2369–2380, 2018.

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, 2023.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. LIMO: Less is more for reasoning. In *Conference on Language Modeling (COLM)*, 2025.

Jingyang Yi, Jiazheng Wang, and Sida Li. Shorterbetter: Guiding reasoning models to find optimal inference length for efficient reasoning. *arXiv preprint arXiv:2504.21370*, 2025.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels. *arXiv preprint arXiv:2412.01981*, 2024.

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

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

Hengxiang Zhang, Hyeong Kyu Choi, Sharon Li, and Hongxin Wei. Detecting distillation data from reasoning models. *arXiv preprint arXiv:2510.04850*, 2025a.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, et al. A survey of reinforcement learning for large reasoning models. *arXiv preprint arXiv:2509.08827*, 2025b.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction. In *The Thirteenth International Conference on Learning Representations*, 2025c.

Zhaoyang Zhang, Shuli Jiang, Yantao Shen, Yuting Zhang, Dhananjay Ram, Shuo Yang, Zhuowen Tu, Wei Xia, and Stefano Soatto. Reinforcement-aware knowledge distillation for LLM reasoning. *arXiv preprint arXiv:2602.22495*, 2026.

Yike Zhao, Simin Guo, Ziqing Yang, Shifan Han, Dahua Lin, and Fei Tan. More data or better data? a critical analysis of data selection and synthesis for mathematical reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track (EMNLP Industry)*, 2025.

Zhenyu Zhao, Sander Land, Daniel M. Bikel, and Waseem Alshikh. Shorthand for thought: Compressing LLM reasoning via entropy-guided supertokens. *arXiv preprint arXiv:2604.26355*, 2026.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. ProcessBench: Identifying process errors in mathematical reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H Chi, Quoc V Le, and Denny Zhou. Take a step back: Evoking reasoning via abstraction in large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. LIMA: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021, 2023.

Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V Le, Ed H Chi, Denny Zhou, Swaroop Mishra, and Huaixiu Steven Zheng. Self-discover: Large language models self-compose reasoning structures. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Appendix

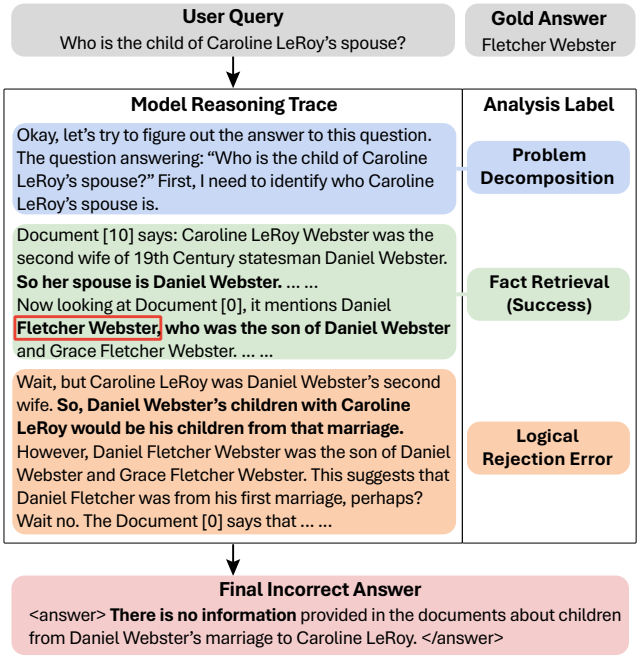

Figure A1: Qwen3-4B trace where the model derives the correct answer mid-trace and discards it under an unverified self-imposed constraint. The failure is at the level of metacognitive monitoring rather than reasoning capacity.

A Representative Failure Example

Figure A1 shows the answer-inclusive failure discussed in Section 1. In this example, the model identifies the relevant spouse and retrieves the corresponding child during intermediate reasoning. It later rejects the correct candidate after introducing an unsupported constraint about the specific marriage. The trace illustrates the type of metacognitive monitoring failure analyzed in this paper. The correct answer is already present in the trace but is not retained through verification.

B MBT Pipeline Pseudo-Code

Algorithm 1 gives the four stages of the MBT pipeline as pseudo-code, with branches for the synthesis (MBT-S) and rewriting (MBT-R) formulations. The notation matches §3.2 and §3.3. $\Pi_{\mathrm{S}}^{\text{5-phase}}$ and $\Pi_{\mathrm{R}}^{\text{5-phase}}$ are the teacher prompts for synthesis and rewriting respectively. The GRPO objective in Stage 4 is the clipped surrogate of Equation (9).

C Extended Related Works

Process Supervision Outcome-based reward models supervise only the final answer, leaving the path to that answer unconstrained. Process Reward Models (PRMs) score each intermediate step. Lightman et al. [2024] report that step-level supervision improves mathematical reasoning over outcome-only rewards. Uesato et al. [2022] train per-step verifiers to score chain-of-thought reasoning. Wang et al. [2024] introduce Math-Shepherd, a PRM trained without step annotations through Monte Carlo rollouts. Luo et al. [2024] reduce annotation cost with OmegaPRM, an MCTS-based binary search that locates the first-error step at roughly 75$\times$ lower annotation cost. Yuan et al. [2024] derive implicit per-step rewards directly from outcome labels. Zhang et al. [2025c] reframe verification as next-token prediction so the verifier can produce its own chain-of-thought before scoring. Zheng et al. [2025] provide a benchmark for evaluating PRMs across multi-step reasoning

Algorithm 1 Metacognitive Behavioral Tuning (MBT)

Require: Student model M_{S}, teacher M_{T}, dataset $\mathcal{D} = \{(q_i, c_i, a_i^*)\}_{i=1}^{N}$, formulation $\mathrm{F} \in \{\mathrm{S}, \mathrm{R}\}$, group size G, clip ϵ, KL coefficient β_{KL}
Ensure: Tuned policy π_θ

Stage 1. Initial trace generation (only for MBT-R).
if $\mathrm{F} = \mathrm{R}$ **then**
  for $i = 1, \ldots, N$ **do**
    $(p_i, \hat{a}_i) \leftarrow M_{\mathrm{S}}(q_i, c_i)$ ▷ native student trace
  end for
end if
Stage 2. Metacognitive trace construction.
for $i = 1, \ldots, N$ **do**
  if $\mathrm{F} = \mathrm{S}$ **then**
    $p_i^{\text{meta}} \sim M_{\mathrm{T}}(\cdot \,|q_i, c_i, a_i^*;\, \Pi_{\mathrm{S}}^{\text{5-phase}})$
  else if $\mathrm{F} = \mathrm{R}$ **then**
    $p_i^{\text{meta}} \sim M_{\mathrm{T}}(\cdot \,|q_i, c_i, p_i, a_i^*;\, \Pi_{\mathrm{R}}^{\text{5-phase}})$
  end if
end for
Stage 3. Supervised fine-tuning.
$\theta \leftarrow \arg\min_\theta \; -\sum_{i=1}^{N} \log \pi_\theta(p_i^{\text{meta}} \mid q_i, c_i)$
Set reference policy $\pi_{\text{ref}} \leftarrow \pi_\theta$
Stage 4. GRPO refinement.
for each training step **do**
  Sample batch $\{q\}$ from $\mathcal{D}$
  for each q in batch **do**
    Sample $\{o_1, \ldots, o_G\} \overset{\text{i.i.d.}}{\sim} \pi_{\theta_{\text{old}}}(\cdot \mid q, c)$
    $r_i \leftarrow \mathrm{F1}\big(\textsc{ExtractAnswer}(o_i),\, a^*\big)$
    $\hat{A}_i \leftarrow r_i - \frac{1}{G}\sum_{j=1}^{G} r_j$
  end for
  Update θ by maximizing the clipped GRPO surrogate (Eq. 9) with KL regularization against π_{ref}
end for
return π_θ

tasks, and verifier-based variants [Hosseini et al., 2024] use the verifier to filter or re-rank generations at inference. Wan et al. [2026] extend this to test-time rubric verification, training a verifier that consults problem-specific rubrics rather than scalar correctness alone. These methods require either per-step annotations or a separate verifier deployed alongside the policy at inference. MBT differs along both axes. The teacher rewrites the entire trace into the five-phase form rather than attaching a scalar reward to each step. The student is trained on this form directly. No verifier is needed at inference and no per-step reward signal is constructed.

Reasoning Distillation A separate line distills the reasoning rationale itself rather than only the final answer. Hsieh et al. [2023] introduce Distilling Step-by-Step, where teacher rationales serve as multi-task supervision and improve smaller students beyond what answer-only distillation achieves. Orca [Mukherjee et al., 2023] extends this with progressive learning from explanation traces of stronger teachers. STaR-style self-distillation [Zelikman et al., 2022] bootstraps the student's own correct rationales as new training data, removing the external teacher. A second line targets the distillation **objective**. MiniLLM [Gu et al., 2024] replaces forward KL with reverse KL to discourage mode-covering on long traces. GKD [Agarwal et al., 2024] aligns the teacher's feedback with sequences sampled from the student. DistiLLM [Ko et al., 2024] introduces a skew KL that is more stable when teacher and student diverge. Busbridge et al. [2025] characterize distillation scaling laws so that compute can be allocated jointly between teacher quality and student size. Tian et al. [2025] and the multi-teacher Merge-of-Thought line [Shen et al., 2025] report that a single strong teacher is not always the best supervisor. Matching teacher style to student capacity matters more [Li et al., 2025c, Chen et al., 2025c]. Shen et al. [2026] and Zhang et al. [2026] carry this into the RL phase.

The student decides **when** teacher imitation is helpful via a generative judge or a selective imitation gate, rather than imitating a fixed teacher trace. These works share the premise that the value of distillation lies in the teacher's specific reasoning content. MBT differs. The value transferred is the regulatory structure of the trace, namely the five-phase scaffold, rather than the teacher's specific conclusions. The controlled comparison in §5.1 reports this trade-off. Naive teacher distillation copies trace content and reaches comparable raw accuracy at the cost of stability. MBT-S and MBT-R copy the structural prior alone and operate at a different point on the accuracy-efficiency frontier.

Reasoning SFT and Data Curation A separate body of work asks what supervision data is enough to teach reasoning. LIMA [Zhou et al., 2023], LIMO [Ye et al., 2025], s1 [Muennighoff et al., 2025], and OpenThoughts [Guha et al., 2025] report that small curated datasets can produce strong reasoners. The Phi-4 series [Abdin et al., 2025, Xu et al., 2025b] reports that data quality matters more than scale. Gandhi et al. [2025] report that the **presence** of cognitive behaviors in the training traces, more than their accuracy, predicts gains. Li et al. [2025a] reach a similar conclusion from the structural side. The long-CoT **structure**, rather than factual correctness, accounts for transfer to a student. Sun et al. [2025] and Guha et al. [2025] qualify this in a difficulty-tier analysis where the hard tier instead favors raw quantity in the form of multiple trajectories per question. Li et al. [2025b] and Wen et al. [2025] build curricula that mix difficulty and trace style. Zhao et al. [2025] report that interpretable formatting and stronger-teacher distillation typically outweigh further volume scaling. Just et al. [2025] provide a step-level signal for this curation, scoring local moves within a trace to identify which segments carry the supervisory signal. Bhambri et al. [2026] note that trace correctness does not reliably predict final-answer correctness, and that more interpretable decompositions can underperform verbose traces. Zhang et al. [2025a] report that benchmark contamination is a risk for reasoning distillation pipelines and propose a logit-based detector for distilled samples. MBT aligns with the structure-over-content view. Rather than scaling supervision, MBT constructs each trace to contain all five metacognitive phases, while remaining agnostic to the teacher's specific solution path.

Reasoning Trace Structure and Faithfulness Recent work treats the reasoning trace itself as an object of analysis rather than as a black-box log. Tan et al. [2025] apply topological data analysis to reasoning traces and report that higher-dimensional geometric features predict reasoning quality more reliably than standard graph metrics or surface markers. Wei et al. [2025] and Wu et al. [2025] document an inverted-U between trace length and accuracy where the back half of long-CoT is often dominated by oscillation and repetition. Zhao et al. [2026] separate low-entropy structural tokens that scaffold the trace from higher-entropy organic tokens that carry problem-specific content, and only the structural fraction is reliably compressible. Ding et al. [2025] and Chen et al. [2025b] examine what happens when self-talk and back-half content are removed. The front 50% of the teacher trace retains roughly 91% of the supervisory value, and reformulations that keep only the conclusions degrade reasoning. Shmidman et al. [2025] report a head-to-head between a divergent-style DeepSeek-R1 teacher and a convergent-style gpt-oss teacher under matched students. They report a token-efficiency gap of roughly $4\times$ at parity accuracy in favor of the convergent style. Li et al. [2026] and Qin et al. [2025], Cai et al. [2025] report consistent conclusions about explicit backtracking under SFT. Lu et al. [2025] apply the same argument at trace-construction time, rewriting teacher traces to remove unnecessary divergence and recover shorter paths to the same answer. These analyses are consistent with the design choice in MBT to train on a convergent, phase-structured trace rather than to reward raw length or surface marker count.

Cognitive Behaviors as a Generalization Signal A recent thread treats cognitive behavior density as a measurable property of reasoning traces and asks how it relates to generalization. Didolkar et al. [2024, 2025] report that prompting models to declare which skill a problem invokes, and to reuse recurring reasoning patterns as named behaviors, cuts tokens by up to 46% while preserving or improving accuracy. Kargupta et al. [2025] build a 28-element taxonomy of cognitive primitives. They report that meta-cognitive controls such as self-awareness and evaluation appear in only 8% to 16% of analyzed traces, despite correlating most strongly with success. Their test-time scaffolding raises accuracy by up to 66.7%. Bai et al. [2025a] decompose reasoning into atomic skills and report that RL-tuned models keep stable behavioural profiles across distribution shift while supervised models drift toward surface patterns. This characterizes the failure mode that MBT-R targets. Alderete et al. [2026] report on causal-reasoning agents that explicit architectural scaffolding through typed context graphs and dynamic monitoring behaviors produces orthogonal contributions. This parallels the decomposition of MBT's five phases into structuring through Understanding and Planning and

monitoring through Self-Correction and Verification. Xiang et al. [2025] and Wang and Zhao [2024] provide cognitive-architecture-style framings of these effects. Kim et al. [2025], Dong et al. [2025], Elenjical et al. [2026], Chen et al. [2025a] are recent attempts to embed metacognitive control into the model itself. MBT shares the premise of this thread and operationalizes it as a single fixed five-phase trace contract. The controlled comparison against a behaviorally-rich but structurally-free distillation baseline in §5.1 is included for this reason.

Table A1: Answer inclusion on MuSiQue. Substring Match and LLM_judge columns report the proportion of correct and incorrect samples in which the gold answer appears in the trace under each criterion in Figure 1. The LLM_judge column uses gpt-oss-120b-high specifically for answer-hit detection as an evidence-grounded match task, whereas Tables 1 and A5 use gemma-4-31b-it for end-task accuracy judging.

Model	Accuracy	Substring Match (correct)	Substring Match (incorrect)	LLM_judge (correct)	LLM_judge (incorrect)
Qwen3-0.6B	22.18	87.87	24.72	94.59	14.78
Qwen3-1.7B	40.46	93.15	42.67	97.34	29.60
Qwen3-4B	57.67	95.05	46.92	98.57	28.54
Qwen3-8B	63.88	95.01	52.69	98.70	34.25

Table A2: Answer inclusion on HotpotQA. Metrics defined as in Table A1.

Model	Accuracy	Substring Match (correct)	Substring Match (incorrect)	LLM_judge (correct)	LLM_judge (incorrect)
Qwen3-0.6B	61.74	85.78	47.41	98.82	32.69
Qwen3-1.7B	73.21	89.08	64.36	99.28	54.49
Qwen3-4B	87.39	89.86	65.95	99.40	49.57
Qwen3-8B	88.21	89.18	68.73	99.54	50.17

Table A3: Answer inclusion on 2WikiMultiHopQA. Metrics defined as in Table A1.

Model	Accuracy	Substring Match (correct)	Substring Match (incorrect)	LLM_judge (correct)	LLM_judge (incorrect)
Qwen3-0.6B	49.20	95.30	44.98	98.64	15.50
Qwen3-1.7B	67.82	96.53	63.06	99.54	38.89
Qwen3-4B	83.64	97.87	60.23	99.62	33.64
Qwen3-8B	85.82	97.86	60.68	99.45	33.60

D Answer Inclusion Analysis

We define an **answer-inclusive error** as an outcome with two properties. First, the gold answer or a paraphrase verified by an LLM-as-a-Judge appears within the reasoning trace as a candidate supported by the retrieved evidence. Second, the final output disagrees with it. Standard evaluation metrics do not capture this step-level dynamic. We complement Figure 1 by quantifying answer inclusion under two methods, namely strict string-based matching and an LLM-as-a-Judge evaluation. Tables A1 to A3 report results on MuSiQue Trivedi et al. [2022], HotpotQA Yang et al. [2018], and 2WikiMultiHopQA Ho et al. [2020]. We split predictions into correct and incorrect cases by the final output. For each reasoning trace, we check whether the gold answer was derived at any intermediate step regardless of the final outcome.

Substring Match checks whether the gold answer string appears in the trace. The method is simple but brittle in that it misses paraphrased answers and matches contextually irrelevant mentions. We therefore add an LLM-as-a-Judge evaluation, executed by gpt-oss-120b at high reasoning effort Agarwal et al. [2025b]. This judge differs from the gemma-4-31b-it judge used for end-task accuracy and for the RRP and MQI behavioral metrics because answer-hit detection is a separate evidence-grounded matching task rather than open-ended scoring. The **LLM_judge** follows a more permissive criterion aligned with Figure 1. The judge outputs answer-inclusive if the trace identifies

Table A4: Extended reasoning stability and efficiency across ID and OOD benchmarks. Degen counts repetitive-loop collapses where decoding hits the 32,768-token cap without producing a parseable answer. **Overall** is the mean output token count across all samples, with degenerated outputs counted at the 32,768-token cap. **Correct** and **Incorrect** condition the mean on final-answer correctness, where Incorrect counts degenerated outputs at the same cap and correct outputs are by construction non-degenerated. **Valid** excludes degenerated samples. Lower is better throughout.

Method	ID HotpotQA					OOD MuSiQue					OOD 2WikiMultiHopQA				
	Degen ↓	Length ↓ Overall	Correct	Incorrect	Valid	Degen ↓	Length ↓ Overall	Correct	Incorrect	Valid	Degen ↓	Length ↓ Overall	Correct	Incorrect	Valid
Qwen3 0.6B															
Base	**0**	477	402	619	477	**0**	731	575	785	731	**1**	569	427	719	566
Prompt	**0**	**472**	404	594	472	**0**	697	551	744	697	**1**	555	426	681	552
GRPO	28	1187	878	1991	1067	16	2167	1488	2544	1963	59	1409	920	2359	1261
RS	18	561	399	792	483	12	943	578	1049	784	50	703	430	957	575
gpt-oss-distill	227	1759	715	3841	778	584	9465	1555	14329	2040	768	2849	748	5824	903
TokenSkip	131	1058	407	1619	487	174	3220	688	3619	928	602	2182	451	3196	644
LIMOPro	176	1071	**288**	1778	**299**	386	5697	479	7430	552	458	1509	**298**	2157	**328**
MBT-S (ours)	3	**472**	453	**531**	459	6	**561**	**451**	**633**	**481**	14	**505**	454	**595**	469
MBT-R (ours)	1	616	606	651	612	5	750	658	804	684	13	682	635	764	649
Qwen3 1.7B															
Base	1	605	478	994	601	**1**	1186	849	1447	1173	2	560	397	927	555
Prompt	4	623	508	1030	606	2	1134	801	1386	1108	**0**	544	418	907	544
GRPO	1	683	565	1327	679	**1**	1379	1024	1801	1366	1	637	473	1314	634
RS	32	747	485	1464	608	19	1533	934	1981	1286	33	683	408	1432	599
gpt-oss-distill	248	1805	663	4403	732	585	9322	1454	16290	1835	664	2557	760	6499	873
TokenSkip	324	1959	476	4102	549	471	7437	942	9889	1306	1034	3221	425	7257	574
LIMOPro	222	1270	**288**	2236	**297**	585	8393	516	11539	610	629	1953	**307**	3090	**331**
MBT-S (ours)	**0**	**451**	446	**491**	451	2	**521**	**462**	**582**	**494**	4	**469**	450	**526**	459
MBT-R (ours)	1	643	633	694	639	**1**	728	686	771	715	8	684	655	757	664
Qwen3 4B															
Base	12	551	432	1217	499	2	1368	1015	1933	1342	4	501	394	1128	491
Prompt	10	556	453	1120	512	5	1324	990	1789	1259	12	535	418	1114	504
GRPO	52	722	433	1747	495	73	2433	1126	4004	1488	121	836	426	2274	526
RS	109	956	425	1983	481	133	3097	1048	4675	1369	121	815	397	1798	505
gpt-oss-distill	67	979	622	2387	689	285	5437	1443	10489	1784	193	1271	670	3450	780
TokenSkip	196	1335	419	2747	480	651	9932	1040	13556	1514	426	1601	390	3361	508
LIMOPro	139	913	**295**	1624	**304**	590	8478	560	12193	634	810	2434	**317**	4050	**346**
MBT-S (ours)	**0**	**448**	445	**489**	448	**0**	**471**	**449**	**514**	**471**	**0**	**460**	456	**491**	460
MBT-R (ours)	**0**	627	623	672	627	1	724	688	784	711	**0**	649	643	692	649

the correct answer or a semantically equivalent one as a valid candidate with any supporting fact, context, or logical connection, even if the model later rejects it. It returns answer-exclusive only when the gold answer is never mentioned, appears solely as part of an unfocused enumeration, or is explicitly framed as a random guess without supporting context. The judge is configured to detect the presence of the correct answer during reasoning, without evaluating reasoning quality. The full prompt and decision criteria appear in Figure A8.

Two patterns emerge from this analysis. First, string matching underestimates intermediate inclusion in correct samples. Even when the final prediction is correct, the gold answer string is absent from the trace in roughly 2% to 15% of cases across Tables A1 to A3. The LLM judge recovers these cases at inclusion rates of 94% to 99%. The lower end is driven by Qwen3-0.6B and Qwen3-1.7B on MuSiQue, with every other model and dataset pair at $\geq 98.5\%$. The gap shows that models often answer through paraphrase or implicit reference that string matching does not capture.

Second, incorrect samples often contain the correct answer in context, and string matching is noisy here. In incorrect samples, Substring Match often yields higher inclusion rates than the LLM judge. For instance, in Table A1 the rates are 24.72% against 14.78%. String matching produces false positives by flagging unrelated mentions of the answer entity. Under the LLM judge, when models do raise the correct answer, it is typically tied to supporting context rather than appearing in passing. Together these results align with the motivation for MBT. Verified answer-inclusive errors are common, which is consistent with models having the necessary information available without retaining it through to the final answer because of insufficient metacognitive monitoring.

E Additional Experimental Results

E.1 Reasoning Efficiency Analysis

E.1.1 Extended Degeneration and Length Analysis

Table A4 reports reasoning efficiency and stability across model scales and benchmarks. Where Table 1 reports final-task accuracy, Table A4 reports how different post-training strategies affect the reasoning process itself.

Reasoning Stability and Degeneration Across model sizes from 0.6B to 4B and across datasets, MBT-S and MBT-R have near-zero degeneration counts. Efficiency-oriented methods such as TokenSkip Xia et al. [2025] and LIMOPro Xiao et al. [2025] produce repetitive loops and reasoning collapse, particularly on the OOD benchmark MuSiQue. The high degeneration rate raises mean inference cost and limits the use of these methods on multi-hop reasoning. MBT reduces this uncontrolled continuation by training the model on the five-phase structure, so traces terminate even on hard inputs.

Efficiency on Correct Samples and the Metacognitive Overhead Output lengths on correct samples show a trade-off. On the in-distribution HotpotQA benchmark, pruning-based baselines sometimes produce shorter traces than MBT. This shortness coincides with degeneration on harder inputs. MBT variants produce slightly longer traces on the easier tasks because they execute the planning, monitoring, and verification phases. We treat this small increase as the cost of the structured form rather than as inefficiency.

Robustness on OOD and Incorrect Samples The structured form helps on the OOD benchmarks. On MuSiQue, baselines show length inflation due to reasoning collapse. For example, incorrect samples from gpt-oss-distill enter long unproductive loops. MBT reduces this failure mode. Even when the final answer is incorrect, MBT keeps trace length bounded. The metacognitive form leads the model to terminate generation earlier on hard inputs.

MBT trades off accuracy and efficiency by changing the reasoning process itself. Where heuristic pruning can compromise stability, MBT structures the reasoning trace. This reduces the degeneration observed under standard distillation and pruning methods. The structure adds a small length overhead on simpler tasks but yields higher accuracy in Table 1. The same structure improves robustness on harder benchmarks. The efficiency gains of MBT therefore come from the structure of the trace rather than from surface-level shortening.

E.1.2 Extended Accuracy-Efficiency-Score Analysis

Table A5 reports the Accuracy-Efficiency Score across three model scales Qwen3-0.6B, 1.7B, 4B and three benchmarks HotpotQA, MuSiQue, and 2Wiki. As defined in Section 4.2, AES serves as a composite metric capturing the trade-off between the relative change in output length ΔLength and the relative change in accuracy ΔAcc.

Metric-Specific AES Calculation The columns labeled "EM", "F1", and "LLM" in Table A5 do not represent raw accuracy scores. Instead, they denote the AES values calculated using the respective metric, namely Exact Match, F1 score, or LLM-as-a-Judge, as the ΔAcc term. Since AES is normalized against the Base model, the Base model attains an AES of zero across all configurations by definition.

AES Behavior of MBT and Baselines Across most scale, dataset, and metric settings, MBT-S and MBT-R have positive AES values. MBT-S is positive in every cell. MBT-R is mildly negative in two LLM-judge cells at the 4B scale on HotpotQA and 2WikiMultiHopQA, where the Base model is already close to the judge's ceiling. MBT therefore improves reasoning accuracy while controlling token cost in most cases. Efficiency-oriented baselines such as TokenSkip and LIMOPro have negative AES in most configurations. The pruning-driven length reduction co-occurs with accuracy degradation or degeneration. Naive reasoning distillation under gpt-oss-distill also has highly negative AES, since copying long teacher traces without the structural prior raises length faster than accuracy.

Table A5: Extended AES across ID and OOD benchmarks. EM, F1, and LLM denote AES values computed with each accuracy metric for ΔAcc, normalized against the Base model at AES = 0. Positive entries indicate improvement and negative entries indicate degradation. The LLM column uses the gemma-4-31b-it judge consistent with Table 1. **Bold** marks the best and underline the second-best per size, dataset, and metric cell, restricted to positive AES. Negative AES corresponds to degradation against the normalized Base and is left unmarked.

Model	Method	ID			OOD					
		HotpotQA			MuSiQue			2WikiMultiHopQA		
		EM↑	F1↑	LLM↑	EM↑	F1↑	LLM↑	EM↑	F1↑	LLM↑
Qwen3 0.6B	Base	0	0	0	0	0	0	0	0	0
	Prompt	-0.29	-0.25	-0.09	-0.26	-0.33	-0.23	-0.62	-0.46	-0.14
	GRPO	0.14	-0.35	-0.92	0.82	0.02	-0.43	0.34	-0.18	-0.45
	RS	0.03	0.01	-0.02	-0.14	-0.17	-0.04	-0.04	-0.02	-0.01
	gpt-oss-distill	-0.34	-0.99	-1.67	-6.61	-8.59	-9.11	-2.01	-2.52	-2.77
	TokenSkip	-0.82	-0.92	-1.02	-3.82	-3.69	-3.70	-2.84	-2.80	-2.77
	LIMOPro	-0.34	-0.39	-0.39	-4.26	-5.08	-5.26	-1.82	-1.35	-0.68
	MBT-S (ours)	**2.41**	**1.71**	**0.97**	**5.26**	**3.36**	**2.72**	**2.21**	**1.69**	**1.38**
	MBT-R (ours)	2.01	1.33	0.60	4.37	2.79	2.21	1.85	1.33	1.06
Qwen3 1.7B	Base	0	0	0	0	0	0	0	0	0
	Prompt	-0.86	-0.62	0.14	-0.48	-0.41	0.05	-0.91	-0.46	0.23
	GRPO	0.44	0.33	0.23	0.88	0.70	0.56	0.50	0.38	0.34
	RS	0.02	0.00	-0.01	0.00	0.01	0.03	0.11	0.10	0.09
	gpt-oss-distill	-1.09	-1.26	-1.44	-4.94	-5.48	-5.76	-2.75	-2.90	-2.94
	TokenSkip	-1.89	-1.95	-1.99	-5.62	-5.58	-5.66	-4.48	-4.52	-4.53
	LIMOPro	-1.30	-0.99	-0.62	-6.09	-6.06	-5.90	-3.74	-3.05	-2.19
	MBT-S (ours)	**1.27**	**1.04**	**0.80**	**2.11**	**1.75**	**1.37**	**0.77**	**0.68**	**0.64**
	MBT-R (ours)	0.86	0.65	0.43	1.59	1.37	1.06	0.35	0.29	0.25
Qwen3 4B	Base	0	0	0	0	0	0	0	0	0
	Prompt	0.00	-0.03	-0.03	-0.30	-0.30	-0.06	-0.21	-0.22	-0.08
	GRPO	0.54	0.34	-0.30	0.03	-0.11	-0.65	0.43	0.09	-0.63
	RS	-0.74	-0.75	-0.76	-1.28	-1.30	-1.36	-0.67	-0.67	-0.68
	gpt-oss-distill	0.21	-0.04	-0.71	-1.71	-2.07	-2.66	-0.32	-0.67	-1.44
	TokenSkip	-1.47	-1.47	-1.53	-7.64	-7.60	-7.69	-2.33	-2.37	-2.37
	LIMOPro	-0.74	-0.56	-0.64	-6.00	-5.92	-6.13	-4.81	-4.32	-4.20
	MBT-S (ours)	**1.28**	**0.96**	**0.24**	**1.83**	**1.50**	**0.85**	**1.08**	**0.82**	**0.09**
	MBT-R (ours)	0.92	0.62	-0.09	1.61	1.29	0.65	0.74	0.47	-0.27

Scalability and Generalization Absolute AES improvements on the simpler HotpotQA benchmark decrease with model scale, but MBT keeps clearly positive AES on MuSiQue at the 4B scale. The metacognitive structure helps most when reasoning complexity is high and uncontrolled exploration is a bottleneck. The efficiency gains hold across the evaluation protocols we tested.

E.2 Extended Comparison between Naive Reasoning Distillation and MBT

Section 5.1 asked whether naive reasoning distillation from a stronger teacher model can serve as an alternative to explicit metacognitive structuring. On MuSiQue at the 4B scale, distillation preserved accuracy but raised degeneration counts and output length, with negative AES. MBT keeps generation bounded. This section extends the analysis to all model scales and benchmarks in Table A6.

Limitations of Naive Distillation Table A6 shows that naive reasoning distillation has structural instability. The EM, F1, and LLM scores are competitive, but degeneration counts are high and traces are long, so AES is negative. The instability is most pronounced on the OOD benchmark MuSiQue, where distilled models do not terminate effectively.

Effect of Metacognitive Structuring MBT-S and MBT-R reduce degeneration and length across all settings while preserving accuracy. MBT therefore has positive AES in most cells of Table A5. The exceptions are two mildly negative LLM-judge cells for MBT-R on 4B HotpotQA and 2WikiMultiHopQA, where Base is already close to the judge's ceiling. On MuSiQue, MBT keeps positive AES at the 4B scale. The structure helps most when reasoning complexity is high and uncontrolled exploration is otherwise common. These results are consistent with the claim that distilling trace content is not sufficient and that explicit five-phase structuring shapes when and how the model reasons.

Table A6: Naive reasoning distillation under gpt-oss-distill compared with MBT across scales and benchmarks. Distillation matches accuracy but inflates Degen and Len, yielding negative AES, while MBT preserves accuracy with bounded and stable traces. AES uses the LLM-judge metric relative to Base. Accuracy uses the gemma-4-31b-it judge consistent with Table 1. **Bold** marks the best per size, dataset, and metric cell and underline the second-best. In the AES columns the highlights are restricted to positive values, since negative AES corresponds to degradation against the normalized Base.

Model	Method	EM ↑	F1 ↑	LLM ↑	Degen ↓	Len ↓	AES ↑
ID: HotpotQA							
Qwen3-0.6B	gpt-oss-distill	62.05	76.81	**87.47**	227	1759	-1.67
	MBT-S (ours)	**62.71**	**76.89**	86.28	3	**472**	**0.97**
	MBT-R (ours)	61.59	75.60	84.85	**1**	616	0.60
Qwen3-1.7B	gpt-oss-distill	64.42	78.94	89.56	248	1805	-1.44
	MBT-S (ours)	**66.37**	**80.20**	**89.60**	**0**	**451**	**0.80**
	MBT-R (ours)	64.89	78.75	88.22	1	643	0.43
Qwen3-4B	gpt-oss-distill	66.86	81.58	**92.03**	67	979	-0.71
	MBT-S (ours)	**68.66**	**82.26**	91.84	**0**	**448**	**0.24**
	MBT-R (ours)	68.22	82.06	91.40	**0**	627	-0.09
OOD: MuSiQue							
Qwen3-0.6B	gpt-oss-distill	**37.11**	**46.19**	**50.31**	584	9465	-9.11
	MBT-S (ours)	35.75	44.56	47.33	6	**561**	**2.72**
	MBT-R (ours)	32.93	42.28	45.14	**5**	750	2.21
Qwen3-1.7B	gpt-oss-distill	**46.26**	**55.21**	**60.20**	585	9322	-5.76
	MBT-S (ours)	42.82	52.77	55.98	2	**521**	**1.37**
	MBT-R (ours)	39.59	50.24	53.99	**1**	728	1.06
Qwen3-4B	gpt-oss-distill	**53.58**	**63.02**	**69.18**	285	5437	-2.66
	MBT-S (ours)	52.34	61.96	66.61	**0**	**471**	**0.85**
	MBT-R (ours)	51.92	61.56	66.40	1	724	0.65
OOD: 2WikiMultiHopQA							
Qwen3-0.6B	gpt-oss-distill	56.64	65.67	72.87	768	2849	-2.77
	MBT-S (ours)	**57.78**	**67.05**	**73.32**	14	**505**	**1.38**
	MBT-R (ours)	57.21	66.35	73.17	**13**	682	1.06
Qwen3-1.7B	gpt-oss-distill	**66.44**	**75.29**	**84.04**	664	2557	-2.94
	MBT-S (ours)	62.85	72.37	80.66	**4**	**469**	**0.64**
	MBT-R (ours)	62.24	72.22	80.62	8	684	0.25
Qwen3-4B	gpt-oss-distill	**71.19**	**79.93**	**89.09**	193	1271	-1.44
	MBT-S (ours)	67.52	77.32	86.67	**0**	**460**	**0.09**
	MBT-R (ours)	68.26	77.92	87.25	**0**	649	-0.27

Table A7: Hyperparameter sensitivity of naive reasoning distillation under gpt-oss-distill across LR and BS. On MuSiQue, Degen exceeds 300 in nearly every configuration, indicating that the instability is structural rather than tuning-driven. We omit the LLM-judge column here because gemma-4-31b-it evaluations were generated only for the standard configuration LR$= 1 \times 10^{-4}$ and BS$= 128$. EM, F1, Degen, and Len are judge-independent and are reported for every configuration.

Method	LR	BS	MuSiQue				2WikiMultiHopQA				HotpotQA			
			EM ↑	F1 ↑	Degen ↓	Len ↓	EM ↑	F1 ↑	Degen ↓	Len ↓	EM ↑	F1 ↑	Degen ↓	Len ↓
Base	N/A	N/A	37.65	48.37	2	1368	50.70	62.07	4	501	50.38	65.45	12	551
gpt-oss distill	1e-5	128	38.39	51.67	312	5306	47.49	63.08	185	1042	49.93	68.40	113	983
	3e-5	128	40.50	53.77	256	4733	48.23	63.17	159	976	49.49	67.91	82	859
	1e-4	128	39.93	52.12	304	5401	47.75	62.89	240	1175	49.91	68.32	81	854
	3e-4	128	27.51	37.65	556	8764	40.87	57.02	390	1565	47.85	66.15	143	1129
	1e-4	64	38.23	50.89	302	5580	47.78	62.97	210	1117	49.87	68.03	88	891
	1e-4	256	40.34	52.39	313	5430	47.70	62.86	194	1051	50.38	68.18	78	838

E.3 Hyperparameter Sensitivity of Reasoning Distillation

Table A7 reports naive reasoning distillation under gpt-oss-distill across learning rates and batch sizes. Hyperparameter changes produce small fluctuations in accuracy and do not remove the instability.

Persistent Instability of Raw Distillation The instability is clearest on MuSiQue. The degeneration count exceeds 300 across nearly all configurations. The model enters repetitive loops hundreds of times in each setting. The inefficiency is not a tuning artefact. Reasoning distillation under this baseline lacks a structural prior on the trace, so exploration is unbounded across training settings.

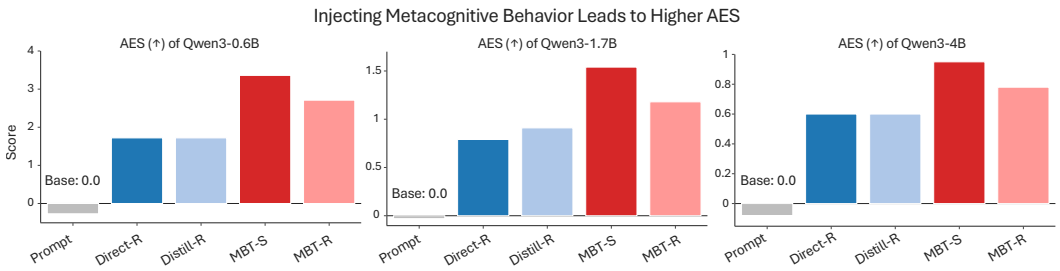

Figure A2: AES across behavior-injection strategies on MuSiQue. MBT-S and MBT-R have higher AES than Prompting and the rewriting-based baselines Direct-R and Distill-R, which operate on external teacher traces.

Final Configuration Selection We adopt LR $= 1 \times 10^{-4}$ and BS $= 128$ as the standard distillation setting in this paper. This configuration is at near-peak accuracy among the tested variants and does not show the degradation seen at the highest learning rates. The same parameters are used for the MBT training configuration, so the comparison across methods is matched.

E.4 Impact of Behavior Injection Strategies

This section analyzes how different sources of reasoning traces affect behavior injection. We compare MBT against the following baselines. **Metacognitive Prompting** adds the metacognitive instructions in a system prompt at inference time with no parameter updates. **Direct-R** rewrites reasoning traces generated by the teacher model. **Distill-R** rewrites reasoning traces produced by the distilled model, which was trained on the teacher model's raw traces. The rewritten traces are then used to fine-tune the base model.

Analysis of Accuracy-Efficiency Score (AES) Figure A2 reports the AES across these methods. The Metacognitive Prompting baseline produces small improvements and a negative AES, so inference-time intervention alone is not enough to install stable control. The rewriting-based baselines Direct-R and Distill-R improve over prompting but stay below MBT-S and MBT-R.

Interpreting the Gap The gap aligns with the source of the training data. MBT-S synthesizes structured traces directly. MBT-R rewrites the student's own traces. Direct-R and Distill-R rely on external traces from teacher or distilled models, which can introduce a distribution mismatch and may not match the student's specific failure modes. Constructing structured synthetic data or rewriting the student's own traces therefore matches the student's distribution more closely than rewriting external traces.

E.5 Detailed Phase-Level Analysis of Metacognitive Behavior

This section provides a per-phase analysis of the reasoning trace alongside Figure 3, using the metacognitive quality index (MQI) defined in §4.2. Figures A3 and A4 extend the main-text 4B panels to all three Qwen3 scales. The lower-left RRP clustering and the MBT-S and MBT-R MQI lead both hold at 0.6B and 1.7B. The full judge prompts for RRP and MQI are in §H.3.

The MQI Rubric For each MuSiQue validation trace, the judge returns two values. The first is a holistic level $L_{\text{obs}} \in \{0, 1, 2, 3, 4, 5\}$ on the rubric of §4.2, ranging from direct answers with no visible reasoning at $L_{\text{obs}} = 0$ to integrated five-phase reasoning at $L_{\text{obs}} = 5$. The second is the subset of phases present in the trace, drawn from Understanding and Filtering, Planning, Execution and Monitoring, Self-Correction, and Verification. We summarize per-method behavior with $\overline{L}_{\text{obs}}$, the mean over samples, and the phase presence ratios. Higher values correspond to traces that contain more of the five phases.

Where Baselines Miss the Target Phases Table A8 reports $\overline{L}_{\text{obs}}$ together with Phase 4 Self-Correction and Phase 5 Verification presence ratios across model scales. Two patterns hold across scales. First, baselines without behavioral supervision such as Base, Prompt, GRPO, RS, and TokenSkip include Phase 4 in roughly 37% to 52% of samples, with Qwen3-0.6B GRPO an outlier

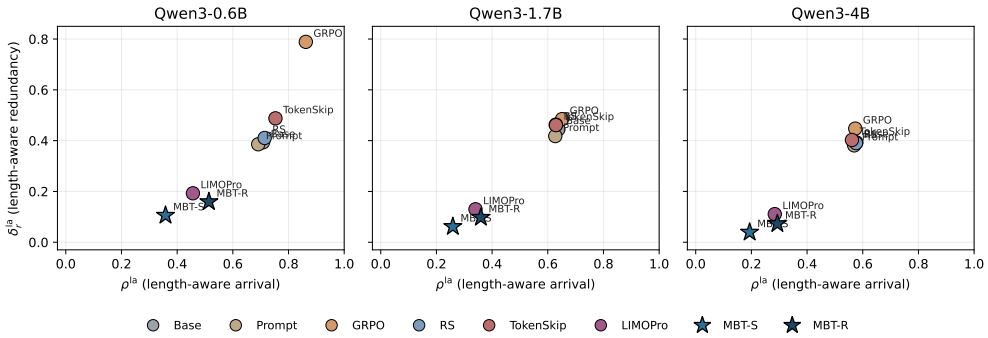

Figure A3: RRP plane on MuSiQue at three Qwen3 scales 0.6B, 1.7B, and 4B. The horizontal axis is the length-aware arrival ρ^{la} and the vertical axis is the redundancy fraction δ_r^{la}. Base, Prompt, GRPO, RS, and TokenSkip sit in the high-ρ^{la} and high-δ_r^{la} region. LIMOPro reaches the lower-left through step pruning rather than learned regulation. MBT-S and MBT-R reach the answer mid-trace at the lowest δ_r^{la} at every scale. The 4B panel also appears in the main text as the left panel of Figure 3.

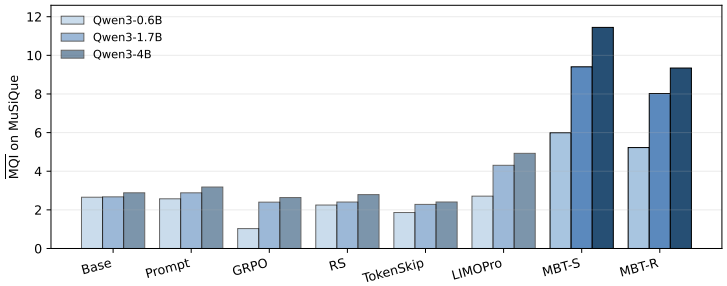

Figure A4: Length-aware $\overline{\mathrm{MQI}}$ on MuSiQue at three Qwen3 scales 0.6B, 1.7B, and 4B. MBT-S is highest at every scale, MBT-R is second, and the baselines are below. The 4B panel also appears in the main text as the right panel of Figure 3.

at 67%. These baselines include Phase 5 in only 19% to 33% of samples. Their $\overline{L}_{\mathrm{obs}}$ values sit between 2.36 and 2.99, consistent with mostly linear chains of thought, intermittent monitoring, and no systematic verification. Second, LIMOPro has an unusual profile. Phase 4 falls below 5%, in line with its post-hoc step-pruning that removes self-correction segments. Phase 5 lands between 77% and 80% across the three scales 0.6B, 1.7B, and 4B, a length-side artefact of its compression. $\overline{L}_{\mathrm{obs}}$ is the lowest among all methods because the rubric scores integrated phase use rather than phase presence alone.

Phase Profiles of MBT-S and MBT-R MBT-S inserts Phase 5 with a Verification ratio of 96% to 98% across scales and reaches $\overline{L}_{\mathrm{obs}} \approx 4.0$. Its Phase 4 ratio of 36% to 37% is in line with the baselines. The pattern follows from how MBT-S synthesizes traces from the gold answer. When the synthesis is correct on the first attempt, the teacher does not need an explicit error-recovery cycle. Self-Correction therefore appears only when the structure requires it. MBT-R reaches the top of the rubric, with $\overline{L}_{\mathrm{obs}} \in \{4.89, 4.89, 4.95\}$ and both Phase 4 at $\geq 91\%$ and Phase 5 at $\geq 99\%$. MBT-R rewrites the student's initial trace into the five-phase form, so Self-Correction is made explicit whenever the original trace needed adjustment. This per-phase gap is why MBT-R has higher raw $\overline{L}_{\mathrm{obs}}$ than MBT-S in Table A8. Figure 3 on the right shows the length-aware $\overline{\mathrm{MQI}} = \overline{L}_{\mathrm{obs}} \cdot T_{\mathrm{base}}/T_i$ instead. The shorter traces of MBT-S flip the ordering there, with MBT-S ahead of MBT-R.

The results show that MBT raises the presence of the targeted metacognitive phases, namely monitoring, error correction, and termination behaviors.

Table A8: Per-phase analysis on the MuSiQue validation set across scales. The per-sample rubric level $L_i^{\text{obs}} \in \{0, \ldots, 5\}$ is defined in §4.2. We report its sample mean $\overline{L}_{\text{obs}} = \frac{1}{n}\sum_i L_i^{\text{obs}}$, the length-aware $\overline{\text{MQI}} = \frac{1}{n}\sum_i L_i^{\text{obs}} \cdot T_{\text{base}}/T_i$ from Eq. 7, and presence ratios for Phase 4 Self-Correction and Phase 5 Verification. $T_{\text{base}} \in \{731, 1173, 1342\}$ for $\{0.6\text{B}, 1.7\text{B}, 4\text{B}\}$ is the same-scale Qwen3 base model's mean trace length on MuSiQue over valid non-degenerated outputs. The judge is gemma-4-31b-it.

Method	Qwen3-0.6B				Qwen3-1.7B				Qwen3-4B			
	$\overline{L}_{\text{obs}}\uparrow$	$\overline{\text{MQI}}\uparrow$	P4%↑	P5%↑	$\overline{L}_{\text{obs}}\uparrow$	$\overline{\text{MQI}}\uparrow$	P4%↑	P5%↑	$\overline{L}_{\text{obs}}\uparrow$	$\overline{\text{MQI}}\uparrow$	P4%↑	P5%↑
Base	2.65	2.65	47.8	19.0	2.67	2.67	46.6	26.7	2.88	2.88	45.4	27.7
Prompt	2.45	2.57	50.4	28.3	2.72	2.88	51.8	24.0	2.99	3.18	49.4	27.5
GRPO	2.77	1.03	66.9	24.9	2.79	2.40	43.5	22.0	2.92	2.64	49.3	33.0
RS	2.41	2.25	49.4	28.9	2.64	2.41	46.2	24.9	2.85	2.79	42.6	28.0
TokenSkip	2.36	1.86	47.0	30.1	2.54	2.28	38.9	26.6	2.72	2.41	37.7	29.8
LIMOPro	2.05	2.71	1.6	76.8	2.24	4.31	2.5	80.1	2.33	4.93	3.5	77.6
MBT-S (Ours)	3.94	**5.99**	37.3	97.9	3.96	**9.41**	36.7	96.2	4.02	**11.45**	36.1	98.1
MBT-R (Ours)	**4.89**	5.23	**91.6**	**99.9**	**4.89**	8.02	**92.0**	**99.8**	**4.95**	9.34	**96.5**	**100.0**

Table A9: Ablation across SFT and GRPO. SFT internalizes the five-phase structure and GRPO refines on top. The full SFT+GRPO pipeline gives the best accuracy-stability trade-off across scales. Accuracy uses the gemma-4-31b-it judge consistent with Table 1.

Method	SFT	GRPO	MuSiQue					2WikiMultiHopQA					HotpotQA				
			EM↑	F1↑	LLM↑	Degen↓	Len↓	EM↑	F1↑	LLM↑	Degen↓	Len↓	EM↑	F1↑	LLM↑	Degen↓	Len↓
Qwen3-0.6B																	
Base			13.36	21.80	25.86	**0**	731	33.99	43.90	51.59	**1**	569	34.83	49.10	65.37	**0**	477
Base		✓	25.78	36.25	39.10	16	2167	54.61	62.93	69.25	59	1409	53.69	67.71	77.79	28	1187
MBT-S	✓		27.89	37.07	39.64	2	**510**	55.14	64.24	70.70	7	**487**	60.08	74.23	83.66	1	**466**
MBT-R	✓		26.93	35.47	37.94	3	743	53.56	62.92	69.82	10	694	58.23	72.22	81.35	2	650
MBT-S	✓	✓	**35.75**	**44.56**	**47.33**	6	561	**57.78**	**67.05**	**73.32**	14	505	**62.71**	**76.89**	**86.28**	3	472
MBT-R	✓	✓	32.93	42.28	45.14	5	750	57.21	66.35	73.17	13	682	61.59	75.60	84.85	1	616
Qwen3-1.7B																	
Base			28.22	37.81	44.06	**1**	1186	52.23	61.65	69.59	2	560	49.63	63.62	75.87	1	605
Base		✓	38.06	48.72	54.74	**1**	1379	**63.25**	72.32	**80.69**	1	637	58.96	73.33	84.92	1	683
MBT-S	✓		39.30	49.32	52.42	2	525	61.20	70.97	79.04	**0**	**467**	63.88	77.89	87.64	**0**	463
MBT-R	✓		34.88	45.05	48.03	**1**	730	59.80	69.84	78.36	**0**	669	63.04	76.94	86.52	2	653
MBT-S	✓	✓	**42.82**	**52.77**	**55.98**	2	**521**	62.85	**72.37**	80.66	4	469	**66.37**	**80.20**	**89.60**	**0**	**451**
MBT-R	✓	✓	39.59	50.24	53.99	**1**	728	62.24	72.22	80.62	8	684	64.89	78.75	88.22	1	643
Qwen3-4B																	
Base			37.65	48.37	62.60	2	1368	50.70	62.07	86.43	4	501	50.38	65.45	90.11	12	551
Base		✓	47.79	59.12	65.37	73	2433	**69.30**	77.73	**87.53**	121	836	64.67	79.56	90.33	52	722
MBT-S	✓		49.65	59.54	64.54	**0**	478	66.64	76.31	85.88	**0**	463	67.66	81.61	91.44	**0**	457
MBT-R	✓		48.03	57.90	62.72	3	745	66.83	76.64	85.94	1	660	66.86	80.80	90.93	**0**	635
MBT-S	✓	✓	**52.34**	**61.96**	**66.61**	**0**	**471**	67.52	77.32	86.67	**0**	**460**	**68.66**	**82.26**	**91.84**	**0**	**448**
MBT-R	✓	✓	51.92	61.56	66.40	1	724	68.26	**77.92**	87.25	**0**	649	68.22	82.06	91.40	**0**	627

E.6 Ablation Study Disentangling SFT and GRPO

This section ablates the contributions of Supervised Fine-Tuning (SFT) and Group Relative Policy Optimization (GRPO) within MBT. Table A9 reports four configurations, namely Base, GRPO-only, SFT-only with MBT-S and MBT-R, and the full MBT pipeline that combines SFT and GRPO.

Unstructured Reinforcement (GRPO-only) Applying GRPO directly to the base model improves EM and F1 across scales. The gains come with longer outputs and higher degeneration rates. GRPO without a structural prior expands the rollout distribution, which produces longer and less stable traces.

SFT as a Structural Prior (SFT-only) SFT on MBT-generated traces improves accuracy and reduces degeneration and length relative to GRPO-only. Even without reinforcement learning, SFT installs the structured form and removes redundant steps. MBT-S produces shorter traces because it is synthesized, while MBT-R retains more intermediate steps from the student's distribution.

SFT + GRPO The most consistent results are obtained by combining SFT and GRPO. GRPO operates on an SFT-initialized policy that already follows the five-phase structure, so it can improve outcome accuracy without reintroducing degeneration. Across all scales, the two-stage configuration is the best operating point on accuracy, efficiency, and stability. SFT installs the structural prior, and GRPO optimizes against the outcome reward. GRPO without the structural prior raises degeneration, while the combined pipeline keeps it bounded.

F Experimental Details

F.1 Post-training via Group Relative Policy Optimization

GRPO Objective For each query q we sample a group of G output trajectories $\{o_1, \ldots, o_G\}$ from the rollout policy $\pi_{\theta_{\text{old}}}$ conditioned on (q, c). Here $o_i = (o_{i,1}, \ldots, o_{i,|o_i|})$ denotes the token sequence of trajectory i. The per-token importance ratio is $w_{i,t}(\theta) = \pi_\theta(o_{i,t} \mid q, c, o_{i,<t}) / \pi_{\theta_{\text{old}}}(o_{i,t} \mid q, c, o_{i,<t})$. The outcome reward $r_i = \text{F1}(\textsc{ExtractAnswer}(o_i), a^*)$ depends only on the final answer extracted from o_i. The trajectory-level advantage standardizes r_i within its group, with $\mu_g = \frac{1}{G}\sum_{j=1}^{G} r_j$ and $\sigma_g = \sqrt{\frac{1}{G}\sum_{j=1}^{G}(r_j - \mu_g)^2}$:

$$\hat{A}_i \;=\; \frac{r_i - \mu_g}{\sigma_g + \epsilon_\sigma}, \tag{8}$$

where ϵ_σ is a small constant for numerical stability when the group rewards collapse to a single value. Writing $\ell_{i,t}(\theta) := \min\big(w_{i,t}(\theta)\,\hat{A}_i,\;\; \text{clip}(w_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\,\hat{A}_i\big)$ for the per-token clipped contribution, the GRPO objective is

$$\mathcal{L}_{\text{GRPO}}(\theta) \;=\; \mathbb{E}_{q\sim\mathcal{D}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\ell_{i,t}(\theta)\right] \;-\; \beta_{\text{KL}}\,\mathbb{D}_{\text{KL}}\big(\pi_\theta \,\|\, \pi_{\text{ref}}\big), \tag{9}$$

with clip range $\epsilon = 0.2$ and KL coefficient $\beta_{\text{KL}} = 0.04$. The reference policy π_{ref} is the SFT-initialized checkpoint. We use $G = 8$ samples per group throughout. The trajectory-level advantage $\hat{A}_i$ is uniform across all tokens of o_i because the reward is outcome-based. Per-token credit assignment within o_i is then driven entirely by the per-token ratio $w_{i,t}(\theta)$. The symbol $w_{i,t}(\theta)$ used here for the per-token importance ratio is distinct from the arrival position ρ_i introduced for the RRP in §4.2.

GRPO Training Setup We implement all GRPO experiments using the verl framework [Sheng et al., 2025]. The Qwen3 student is trained on the HotpotQA training set for 60 steps with batch size 512 and maximum response length 4096 tokens. We use a learning rate of 1×10^{-6}, weight decay 0.1, and KL coefficient $\beta_{\text{KL}} = 1 \times 10^{-3}$. All GRPO runs are conducted on 4 H100 GPUs.

F.2 Decoding, Judges, and the Accuracy-Efficiency Score

Decoding At inference we use a sampling temperature of 0.6 and top-p of 0.95 matching the setup used for the DeepSeek-R1 evaluation [Guo et al., 2025], and a maximum decoding length of $32{,}768$ tokens applies to all methods and scales.

Accuracy and Efficiency Metrics For task accuracy we report Exact Match (EM) and token-level F1 against the gold answer string, together with an LLM-as-a-Judge score produced by gemma-4-31b-it. The Gemma judge is from a different family than the gpt-oss-120b teacher used during MBT trace construction so as to avoid closed-loop bias when the same model both generates traces and grades them. The Gemma judge is used both for end-task accuracy and for the RRP and MQI judgments of §4.2. For efficiency we record the mean output length (Len) of the reasoning trace as a proxy for inference-time compute, together with the count of degeneration failures (Degen). Degenerated outputs are those whose length saturates the $32{,}768$-token decoding limit. In practice this corresponds to the model entering a repetitive loop and failing to terminate.

Accuracy-Efficiency Score (AES) We use the Accuracy-Efficiency Score of Luo et al. [2025] to combine the length and accuracy axes in a single scalar. For a method M compared against a base model B on the same benchmark, with mean response lengths L_M, L_B and accuracies A_M, A_B, define $\Delta L = (L_B - L_M)/L_B$, which is positive when M is shorter, and $\Delta A = (A_M - A_B)/A_B$, which is positive when M is more accurate. The AES is

$$\text{AES} \;=\; \begin{cases} \alpha\,\Delta L + \beta\,|\Delta A|, & \Delta A \geq 0 \\ \alpha\,\Delta L - \gamma\,|\Delta A|, & \Delta A < 0, \end{cases} \tag{10}$$

with weights $\alpha = 1$, $\beta = 3$, and $\gamma = 5$ following Luo et al. [2025]. Accuracy degradation is penalized more heavily than equivalent efficiency gains. The Base model is normalized to AES $= 0$ on every benchmark and metric.

Table A10: Per-method differences in the SFT data source. All entries use the training configuration in the surrounding paragraph at 1 epoch, AdamW, lr $= 10^{-4}$, batch $= 128$, cosine warmup 0.1, and gradient norm clip 0.3.

Method	Trace generator	Filter or Selection	Modification before SFT
RS (Self-Distill)	Qwen3 student	Reject incorrect (gold-EM)	None
gpt-oss-distill (sft / grpo)	gpt-oss-120b teacher	Reject incorrect (gold-EM)	None (raw teacher trace)
TokenSkip	Qwen3 student	Reject incorrect (gold-EM)	Token-level pruning at ratio $\gamma \sim \text{Unif}\{0.5, \ldots, 1.0\}$
LIMOPro	gpt-oss-120b teacher	Reject incorrect (gold-EM)	Step-level pruning at ratio 0.5, perplexity-scored
MBT-S (Ours)	gpt-oss-120b teacher	N/A (synthesized conditioned on gold)	Five-phase prompt $\Pi_{\text{S}}^{\text{5-phase}}$
MBT-R (Ours)	gpt-oss-120b teacher	N/A (rewriting, both correct and incorrect drafts)	Five-phase rewrite prompt $\Pi_{\text{R}}^{\text{5-phase}}$

F.3 Baseline Methods

In the main paper, we compare MBT against the efficient-reasoning baselines TokenSkip Xia et al. [2025] and LIMOPro Xiao et al. [2025]. Both approaches construct compact reasoning traces by retaining the components considered important from model-generated reasoning. The reduced traces then fine-tune a target model via supervised fine-tuning (SFT) for shorter outputs. Unlike MBT, these approaches compress or prune existing reasoning traces rather than installing a structural prior.

TokenSkip TokenSkip first obtains full reasoning traces from a target model. It then estimates token-level importance scores using a bidirectional BERT-based model Xia et al. [2025], Pan et al. [2024]. Based on these scores, unimportant tokens are pruned to construct compact reasoning traces, which are subsequently used as SFT data. Following the TokenSkip, we generate reasoning traces on the HotpotQA training set using the target model Qwen3 Yang et al. [2025]. We filter samples based on answer correctness and randomly sample a compression ratio γ for each training instance from the set {0.5,0.6,0.7,0.8,0.9,1.0}. We prune tokens based on the sampled compression ratio and use the resulting compressed reasoning traces as the SFT dataset for training.

LIMOPro LIMOPro uses reasoning traces generated by a model larger than the target model. It focuses on pruning functional reasoning components that do not contribute to logical progression, using a perplexity-based importance score. LIMOPro first segments the reasoning trace into intermediate steps using an LLM. It then classifies each step as either logical progression or functional, for example validation or error correction. For steps in the functional group, LIMOPro computes importance scores based on the change in perplexity when removing each step. Based on these scores, it prunes steps with lower importance and uses the resulting compressed reasoning traces as SFT data. In our experiments, we generate reasoning traces on the HotpotQA training set using gpt-oss-120b Agarwal et al. [2025b]. We use the same model for step segmentation and step classification. Following the official LIMOPro implementation, we filter samples based on answer correctness, apply a pruning ratio of 0.5, and use the pruned reasoning traces to construct the SFT dataset for training.

Consolidated Training Configuration For all SFT experiments including RS, gpt-oss-distill, MBT-S, MBT-R, TokenSkip, and LIMOPro, we train the target model Qwen3 for 1 epoch with a learning rate of 1×10^{-4} and an effective batch size of 128. We employ a cosine learning rate scheduler with a warmup ratio of 0.1, AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$, weight decay 0.0, and gradient norm clip 0.3. The maximum sequence length is 32,768 tokens. All experiments use mixed-precision bfloat16 on 4×L40 or 4×H100 GPUs. Per-baseline differences are confined to the source of the SFT dataset, summarized in Table A10.

Teacher Model Access and Reasoning-Effort Setting The teacher model used to construct the metacognitively structured traces is gpt-oss-120b, an open-weights reasoning model released by OpenAI [Agarwal et al., 2025b]. The Hugging Face card is at `https://huggingface.co/openai/gpt-oss-120b` and is distributed under the Apache-2.0 license. Reasoning-effort is controlled by a system-prompt suffix that selects low, medium, or high from the Harmony chat template. We use reasoning_effort=high throughout this paper. This setting yields longer internal deliberation in the analysis channel before the final answer. We serve the model with vLLM, with max_model_len = 32,768, tensor_parallel_size = 4, and gpu_memory_utilization = 0.9. The model card and license permit research and commercial use. Reproducing trace generation on hardware-limited environments is a follow-up listed in §6.

G Qualitative Analysis of Metacognitive Reasoning Traces

This section gives qualitative examples of how metacognitive behaviors appear in reasoning traces before and after MBT. We present three cases in sequence, namely an initial reasoning trace without metacognitive control, a structured trace synthesized by MBT-S, and a rewritten trace by MBT-R. The progression shows how the structural form changes the trace and how it makes the detection and repair of logical errors explicit.

G.1 Initial Reasoning Trace

Figure A5 shows an initial reasoning trace from the base Qwen3 model. The trace shows a failure pattern that occurs when no metacognitive structure is applied. The document set contains the evidence needed to answer the question. The model introduces an unverified assumption early in the process and commits to it without subsequent monitoring. The model does not revise the hypothesis and terminates on an incorrect conclusion.

G.2 Synthesized Trace (MBT-S)

Figure A6 shows a metacognitively structured MBT-S reasoning trace synthesized by gpt-oss-120b. The MBT-S trace follows the five-phase form. The trace separates task clarification, planning, controlled evidence integration, self-correction, and verification. Under this structure the model does not commit to unverified assumptions and derives the correct solution.

G.3 Rewritten Trace (MBT-R)

Figure A7 shows the MBT-R trace, generated by rewriting the flawed initial trace in Figure A5. The teacher revisits the student's intermediate assumptions and re-checks them against the provided documents. The rewrite does not discard the original trace. It keeps the factual observations that are consistent with the documents and revises the steps that rely on unsupported inferences. The output is a trace in which intermediate conclusions are checked against the evidence and the final answer is reached through explicit self-correction and verification.

H Evaluation Prompts

This section lists the prompts used throughout the paper, grouped by the model that executes them. The **judge prompts** are run by external evaluators and include the answer-inclusion judge executed by gpt-oss-120b at high reasoning effort Agarwal et al. [2025b] for evidence-grounded answer-hit detection in §H.1, and the behavioral judges for the Reach-Redundancy Profile and the Metacognitive Quality Index, both executed by gemma-4-31b-it in §H.3. The **teacher prompts** are run by gpt-oss-120b to construct MBT-S and MBT-R training traces in §H.2. The **student inference prompts** are provided to the target model Qwen3 Yang et al. [2025] for the Base and Metacognitive Prompting baselines in §H.4 and H.5. Each dimension of reasoning behavior is defined by a dedicated prompt rather than by a single correctness criterion.

H.1 Answer Inclusion Evaluation Prompt

The answer inclusion prompt in Figure A8 determines whether a reasoning trace is answer-inclusive. A trace is answer-inclusive when the correct answer is derived, inferred, or identified at any point with supporting context. The evaluation does not check final-answer correctness or reasoning quality. The criterion is lenient and counts a correct intermediate conclusion as answer-inclusive even if the model later rejects it. The judge excludes cases where the correct answer appears only as part of an unfocused enumeration or as a random guess. The prompt is used in the answer-inclusive error analysis in Figure 1 and Tables A1 to A3.

H.2 Metacognitively Grounded Reasoning Trace Prompt

H.2.1 Metacognitive Trace Synthesis (MBT-S)

The synthesis prompt used for MBT-S in Figure A9 instructs the teacher gpt-oss-120b to generate reasoning traces from scratch. Given a query, contextual documents, and the gold answer, the prompt applies a five-phase process spanning goal clarification, planning, execution monitoring, self-correction, and verification. The gold answer is provided as input, but the model is instructed to write the trace as an independent derivation without referencing the answer. The resulting traces are used as supervision for MBT-S.

H.2.2 Metacognitive Trace Rewriting (MBT-R)

The rewriting prompt in Figure A10 instructs the teacher gpt-oss-120b to rewrite a reasoning trace produced by the student. The prompt is for restructuring rather than synthesis from scratch. The teacher retains valid intermediate deductions and adds the missing phases of monitoring, self-correction, and verification. When the original trace contains errors, the prompt includes an error-detection step. The teacher identifies the flaw and redirects the trajectory toward the correct solution. The design produces traces that retain the student's intermediate steps under the five-phase form.

H.3 Behavioral Evaluation Prompts for RRP and MQI

H.3.1 Reach-Redundancy Profile (RRP)

The RRP prompt asks the judge to perform a paragraph-level annotation of the trace. The trace is split into paragraphs and each paragraph end is tagged with a sentinel marker $[[M_1]], [[M_2]], \ldots$ The judge then identifies three values. The first is the first paragraph $[[M_k]]$ at which the gold answer is explicitly derived from the retrieved evidence, or the special marker $[[M_{00}]]$ if the answer never appears as a justified candidate. The second is, for every paragraph in the trace both before and after the answer-deriving paragraph, the label PROGRESS, VERIFICATION, or REDUNDANT. The third is a self-reported confidence in the overall judgment. This per-paragraph annotation produces both the arrival position ρ_i and the redundancy fraction $\delta_{r,i}$ used in the score $\mathcal{R}_i$ of §4.2. The full prompt template is shown in Figure A11.

Length-Aware Variants The within-trace normaliser T_i used in ρ_i and $\delta_{r,i}$ allows long traces to disguise late arrival as a moderate ρ_i value, since both numerator and denominator grow together. For cross-method comparisons we therefore additionally report length-aware variants $\rho_i^{\mathrm{la}}, \delta_{r,i}^{\mathrm{la}}$. These variants anchor the denominator at the mean trace length $T_{\mathrm{base}}^{(m)}$ of the same-scale Qwen3 base model over valid non-degenerated outputs on the evaluation benchmark,

$$\rho_i^{\mathrm{la}} \;=\; \min\left(1,\; \frac{1}{T_{\mathrm{base}}^{(m)}} \sum_{j=1}^{k_i} s_{i,j}\right), \qquad \delta_{r,i}^{\mathrm{la}} \;=\; \min\left(1,\; \frac{r_i}{T_{\mathrm{base}}^{(m)}}\right). \tag{11}$$

These variants put all methods on a common length scale. For example, a method which arrives at the gold answer in twice as many sentences as the base model receives a ρ^{la} at least twice as large, rather than being rescaled by its own also-doubled trace length. The length-aware RRP plane in §5 uses these variants as its axes.

H.3.2 MQI Prompt

The MQI prompt presents the five-phase rubric of Understanding and Filtering, Planning, Execution and Monitoring, Self-Correction, and Verification. The judge returns three values. The first is a holistic level $L_{\mathrm{obs}} \in \{0, 1, 2, 3, 4, 5\}$ on the rubric scale, ranging from a direct answer with no visible reasoning to fully integrated five-phase reasoning. The second is the explicit subset of phases identifiable in the trace. The third is a self-reported confidence in the final answer on $[0, 1]$. The per-sample MQI is then computed offline as $\mathrm{MQI}_i = L_{\mathrm{obs},i} \cdot T_{\mathrm{base}}/T_i$ in Eq. 7, with T_i the token count of the trace and T_{base} the same-scale base-model average. Aggregate $\overline{\mathrm{MQI}}$ values per method and scale are reported in Table A8. The same Gemma judge is used for RRP and MQI throughout the paper. The full prompt template is shown in Figure A12.

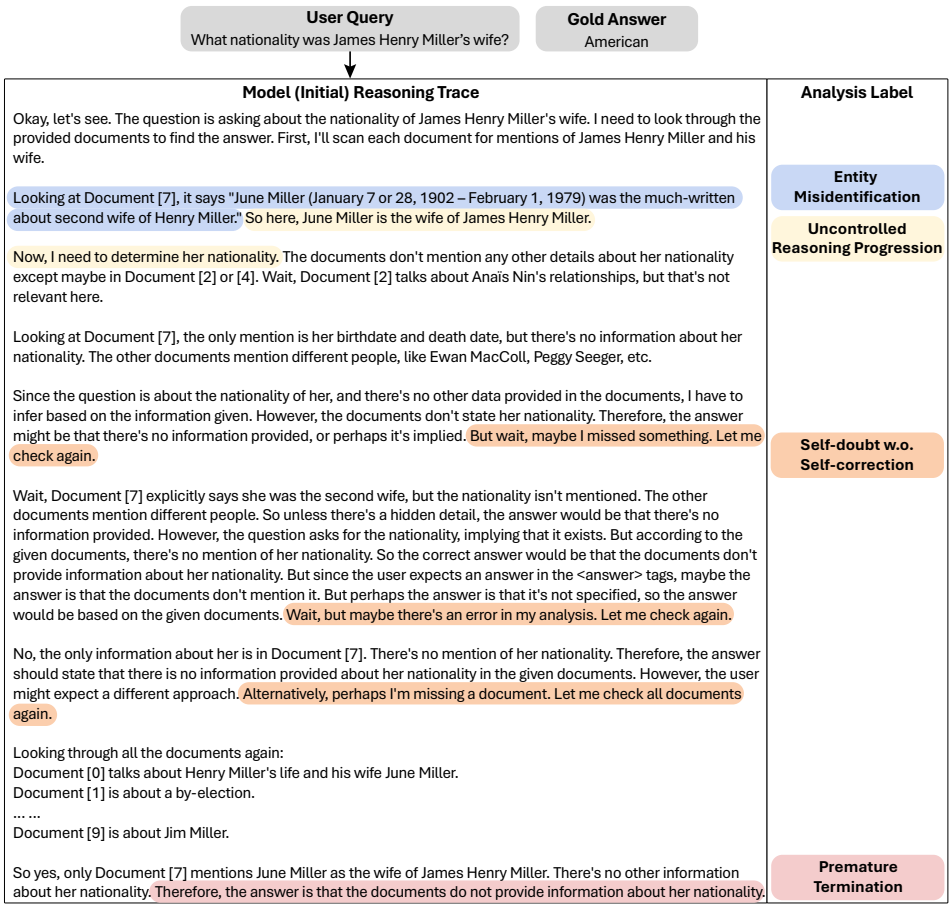

Figure A5: Initial Qwen3 trace without metacognitive control. The model commits to an unverified assumption early, does not invoke self-correction, and terminates on an incorrect conclusion. Best viewed in color.

H.4 Base Prompt

For the Base model evaluation we use a minimal multi-hop QA prompt in Figure A13 that requests only the final answer given the retrieved documents. The prompt imposes no structural constraints, so the model relies on its internal biases. This unconstrained setting is the control for measuring the model's native reasoning behavior.

H.5 Metacognitive Prompting Prompt

The Metacognitive Prompting baseline adds explicit metacognitive instructions at inference time without any additional post-training, as shown in Figure A14. Where MBT installs the five-phase structure through parameter updates, this setting relies on a system prompt alone. The system prompt contains instructions for goal clarification, planning, and self-correction. The comparison separates the effect of external prompt guidance from that of post-training and tests whether prompt-level instructions are enough to obtain the same trace shape as MBT.

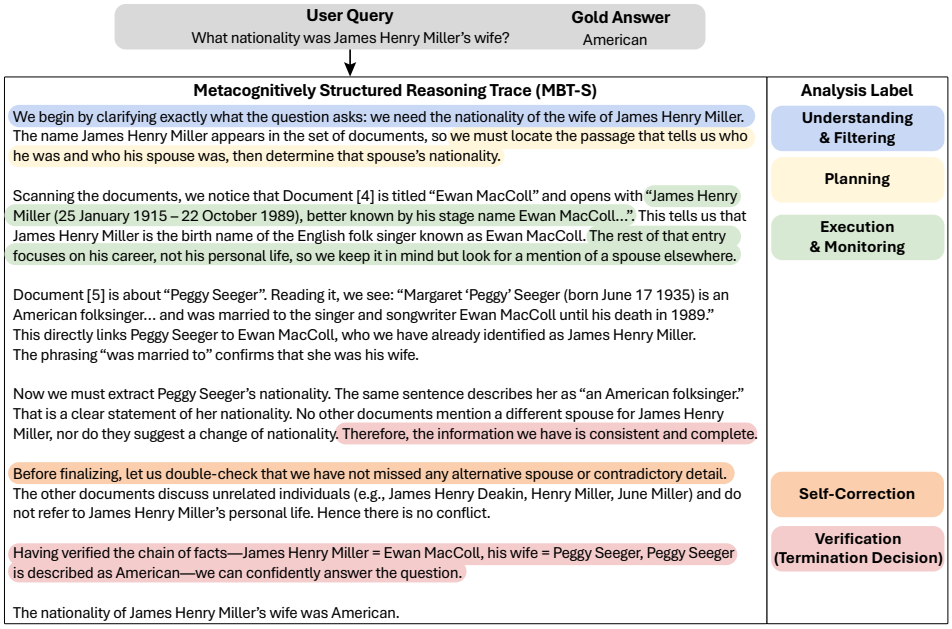

Figure A6: MBT-S synthesized trace. The five-phase structure constrains the trace away from premature commitment to unverified assumptions and yields the correct answer. Best viewed in color.

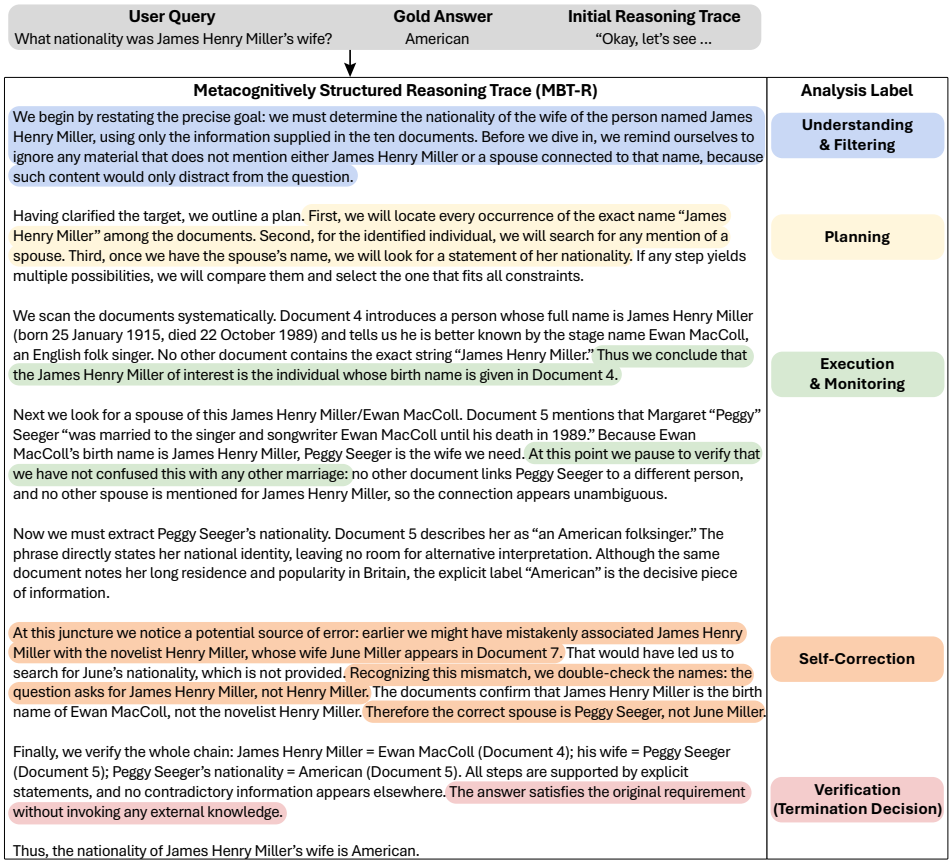

Figure A7: MBT-R rewritten trace from the flawed initial trace in Figure A5. The model identifies the unsupported inference and corrects the path while retaining valid intermediate steps. Best viewed in color.

LLM-as-a-Judge Prompt for Answer Inclusion Evaluation

{
"role": "user",
"content": """
You are an expert evaluator. Your task is to check if the draft solution contains any instance where the correct answer is derived, inferred, or identified with some supporting context.

Please apply a lenient standard for reasoning. You should prioritize detecting the correct answer over critiquing the quality of the logic.

Criteria for outputting YES:

- Output YES if the draft solution identifies the correct answer (or a semantically equivalent answer) as a valid
candidate at any point.

- Output YES if the correct answer appears along with a supporting fact, context, or logical connection.

- Output YES even if the model later rejects this answer.

Criteria for outputting NO:

- Output NO only if the correct answer is never mentioned at all.

- Output NO only if the correct answer is mentioned solely as part of listing multiple choices (e.g., A, B, C, D)
without focusing on it.

- Output NO only if the correct answer is explicitly stated to be a random guess without any supporting
information.

Based on the criteria above, output only YES or NO. Do not provide any explanation.

<question>
{question}
</question>

<correct_answer>
{answer}
</correct_answer>

<draft_solution>
{reasoning_trace}
</draft_solution>

""".strip(),
}

Figure A8: LLM-as-a-Judge prompt for answer inclusion as used in Figure 1 and Appendix D. The judge outputs YES if the gold answer or a paraphrase appears as a supported candidate in the trace, regardless of the final prediction.

Metacognitively Grounded Reasoning Trace Synthesis Prompt (MBT-S)

{
"role": "user",
"content": """
You are an advanced AI reasoner capable of simulating human-like metacognition. Your task is to solve a given question by generating a coherent, deeply reasoned, and accurate internal monologue.

Your primary goal is to derive the solution from scratch while employing metacognitive strategies to ensure high-quality reasoning. You will be provided with the question and the correct answer. You must ensure your reasoning process logically arrives at this correct answer, but you must simulate an authentic, independent discovery process. If there are common pitfalls or complexities, you should simulate a realization of potential error and self-correction, rather than simply stating the result immediately.

Please follow these specific instructions for the generating process:

1. Perspective and Tone: Write in the first-person plural (We). The tone should be introspective, analytical, and deliberate, mimicking a stream of consciousness.

2. Structure and Metacognitive Flow:
- Phase 1: Understanding and Filtering (System 2 Attention). Start by explicitly re-stating the core goal of the problem and filtering out any irrelevant information or distractions. Ensure we fully grasp the requirements before moving forward.
- Phase 2: Planning (Plan-and-Solve). Before calculating or deducing, outline a high-level strategy or roadmap. Break the problem into manageable sub-tasks.
- Phase 3: Execution and Monitoring. Proceed through the steps. At each transition, engage in active monitoring. Ask internal questions like "Is this step logically sound?" or "Does this align with our previous findings?" Perform necessary calculations or deductions carefully.
- Phase 4: Self-Correction (Reflexion). If the reasoning encounters a complex area or a potential ambiguity, simulate a moment of doubt. Use phrases like "Wait, let us double-check that," or "Something feels off here." Identify any potential logical gaps, explain why a hasty conclusion might be wrong, and strictly ensure the path aligns with the correct answer provided.
- Phase 5: Verification (Chain-of-Verification). Before concluding, review the final result against the initial constraints to ensure meaningful consistency and high confidence.

3. Formatting Constraints:
- Do not use any Markdown formatting. Do not use bold, italics, headers, or bullet points.
- Use clear paragraph breaks to indicate shifts in thought or new steps in the reasoning process.
- Do not mention the existence of the provided "correct answer" or that you were given the answer beforehand.
The output must read as a single, authentic, independent thought process.

4. Output Objective: The final result should be a rich, error-free, and logically robust solution that demonstrates not just the answer, but the careful, self-correcting journey of getting there.

<question>
Answer the following question based on the given documents.

Documents: {context}

Question: {question}
</question>

<correct_answer>
{answer}
</correct_answer>

""".strip(),
}

Figure A9: MBT-S synthesis prompt. The teacher generates a five-phase reasoning trace from scratch, written as an independent derivation despite being given the gold answer.

Metacognitively Grounded Reasoning Trace Rewriting Prompt (MBT-R)

{
“role”: “user”,
“content”: “””
You are an advanced AI reasoner capable of simulating human-like metacognition. Your task is to rewrite a given draft solution into a coherent, deeply reasoned, and accurate internal monologue.

Your primary goal is to integrate the valid explorations from the draft solution while employing metacognitive strategies to enhance the reasoning quality. If the draft solution contains errors or leads to an incorrect conclusion, you must naturally steer the reasoning process toward the correct answer by simulating a realization of error and self-correction, rather than simply stating the right answer.

Please follow these specific instructions for the rewriting process:

1. Perspective and Tone: Write in the first-person plural (We). The tone should be introspective, analytical, and deliberate, mimicking a stream of consciousness.

2. Structure and Metacognitive Flow:
- Phase 1: Understanding and Filtering (System 2 Attention). Start by explicitly re-stating the core goal of the problem and filtering out any irrelevant information or distractions. Ensure we fully grasp the requirements before moving forward.
- Phase 2: Planning (Plan-and-Solve). Before calculating or deducing, outline a high-level strategy or roadmap. Break the problem into manageable sub-tasks.
- Phase 3: Execution and Monitoring. Proceed through the steps while incorporating the thinking steps from the draft solution. At each transition, engage in active monitoring and question whether the inherited reasoning is logically sound. If necessary, pause and critique illogical or unsupported steps.
- Phase 4: Self-Correction (Reflexion). If the draft solution deviates from the correct answer, simulate a moment of doubt. Use phrases like “Wait, let us double-check that,” or “Something feels off here.” Identify the logical gap, explain why the previous reasoning was incorrect, and adjust the path toward a correct conclusion.
- Phase 5: Verification (Chain-of-Verification). Before concluding, review the final result against the initial constraints to ensure meaningful consistency and high confidence.

3. Formatting Constraints:
- Do not use any Markdown formatting. Do not use bold, italics, headers, or bullet points.
- Use clear paragraph breaks to indicate shifts in thought or new steps in the reasoning process.
- Do not mention the existence of the “draft solution” or the “correct answer.” The output must read as a single, authentic, independent thought process.

4. Output Objective: The final result should be a rich, error-free, and logically robust solution that demonstrates not just the answer, but the careful, self-correcting journey of getting there.

<draft_solution>
{reasoning_trace}
</draft_solution>

”””.strip(),
}

Figure A10: MBT-R rewriting prompt. The teacher restructures a student draft into the five-phase form, retains valid intermediate steps, and inserts explicit self-correction when needed.

LLM-as-a-Judge Prompt for Reach-Redundancy Profile (RRP)

{
"role": "user",
"content": """
You are an expert evaluator of LLM reasoning trajectories. The reasoning trace below has been segmented into paragraphs separated by markers of the form [[M01]], [[M02]], ... Each marker [[Mk]] appears IMMEDIATELY AFTER paragraph k. The very last paragraph has no trailing marker.

You must perform three tasks and output exactly four lines (no other text).

TASK 1 : Answer-deriving paragraph.
Find the FIRST paragraph in which the model has explicitly derived a candidate answer that is semantically equivalent to the gold answer, supported by at least one piece of evidence drawn from the question or context. Mere mention of the entity in passing, or as part of an enumeration of distractors, does not count.

Output the marker that immediately FOLLOWS that paragraph: - If derivation is in paragraph k (and k is not the last), output [[Mk]].
- If derivation is in the LAST paragraph (no trailing marker), output [[M_END]].
- If no paragraph derives the gold answer at all, output [[M00]].

TASK 2 : Redundant paragraphs (whole-trace classification).
Classify EVERY paragraph in the trace (paragraphs 1..N, both before and after the answer-deriving paragraph) into one of three labels:
- PROGRESS : introduces new evidence, executes a planned step, decomposes the question, or makes substantive forward motion toward the answer (including the trace's first explicit emission of the final answer).
- VERIFICATION : explicitly audits an already-established candidate by appealing to CONCRETE EXTERNAL EVIDENCE OR AN ALTERNATIVE PATH. Must contain at least one of (a) a specific document/source reference + consistency check, (b) explicit comparison with a NAMED alternative the model considered, (c) identification of a specific potential error followed by an evidence-based resolution, or (d) explicit re-derivation from primary inputs / re-application of the question's constraint. Generic wrap-up ("in conclusion", "we synthesize the answer"), phase-listing ("after planning, executing, monitoring, ..."), and bare consistency assertions are NOT sufficient.
- REDUNDANT : repetitive restatement of an already-established conclusion, baseless self-doubt that does not identify a specific issue, circular looping, or generic wrap-up paragraphs that only declare completion without a concrete audit action.
Critical rule: a paragraph qualifies as VERIFICATION ONLY when it performs a CONCRETE auditing action under (a)–(d) above. The phrase "Phase 5: Verification" appearing in the trace is NOT itself sufficient — only the CONTENT of the paragraph matters. A paragraph is REDUNDANT iff strictly more than half of its sentences satisfy the REDUNDANT rule. Output the comma-separated list of trailing markers of all REDUNDANT paragraphs (use [[Mk]] for paragraphs 1..N-1 and [[M_END]] for paragraph N). If none, output NONE.

TASK 3 : Confidence.
A single float in [0,1] reflecting how confident you are in your TASK 1 answer.

Output format (strict, exactly four lines, no other text):
ANSWER_PARAGRAPH_MARKER: <[[Mk]] or [[M_END]] or [[M00]]>
REDUNDANT_PARAGRAPH_MARKERS: <comma-separated [[Mk]] list, or NONE>
CONFIDENCE: <float in [0,1]>
NOTES: <one short sentence; optional but always emit the line>

<question>
{question}
</question>

<gold_answer>
{answer}
</gold_answer>

<reasoning_trace>
{marked_reasoning_trace}
</reasoning_trace>

""".strip(),
}

Figure A11: LLM-as-a-Judge prompt for the Reach-Redundancy Profile defined in §4.2. The judge marks where the gold answer is first derived and labels each paragraph as PROGRESS, VERIFICATION, or REDUNDANT, yielding the arrival position ρ and redundancy fraction δ_r of (6).

LLM-as-a-Judge Prompt for Metacognitive Quality Index (MQI)

{
"role": "user",
"content": """
You are an expert evaluator of AI reasoning capabilities, specifically focusing on metacognitive strategies. Given a question, its difficulty tier, the gold answer, and a model's reasoning trace, perform three tasks.

PHASE LIST:

1. Understanding and Filtering : explicitly restating the goal and filtering noise.
2. Planning : outlining a high-level strategy before execution.
3. Execution and Monitoring : checking each step's logical validity during execution.
4. Self-Correction : simulating doubt or detecting an error and explicitly correcting the path.
5. Verification : final review of the result against the constraints.

TASK A : Phase presence (0–5 rubric).

Score 0: Direct answer with no visible reasoning.
Score 1: Linear chain-of-thought with no planning, no monitoring, no verification.
Score 2: Either a plan or a verification, but middle execution is linear.
Score 3: Plan + execution monitoring; may miss filtering or self-correction.
Score 4: 4 of 5 phases clearly demonstrated.
Score 5: All 5 phases integrated.

Note: a higher score is NOT automatically better. The appropriate level depends on question difficulty. Score the trace strictly by the rubric without considering whether the level is appropriate.

TASK B : Distinct phase set.

List the indices of the phases that are **distinctly present** in the trace (i.e. recognisable as a phase, not just one passing remark). Output a comma-separated list of integers in {1,2,3,4,5}, or NONE.

TASK C : Final-answer confidence.

Locate the model's final answer. If the trace ends with an explicit confidence statement (a number in [0,1] or a percentage), output it as a float in [0,1]. Otherwise, judge from the language of the final answer alone (hedged "I think...", "probably" $\rightarrow$ 0.4–0.6; categorical "the answer is X" $\rightarrow$ 0.8–0.95) and output a single float in [0,1].

Difficulty tier of the question: {difficulty_tier}.

Output format (strict, three lines, no other text):
L_OBS: <integer 0..5>
PHASES: <comma-separated integers, or NONE>
CONFIDENCE: <float in [0,1]>

<question>
{question}
</question>

<gold_answer>
{answer}
</gold_answer>

<reasoning_trace>
{reasoning_trace}
</reasoning_trace>

""".strip(),
}

Figure A12: LLM-as-a-Judge prompt for the Metacognitive Quality Index defined in §4.2. The judge returns a holistic level $L_{\text{obs}} \in \{0, \ldots, 5\}$, the explicit phase subset $\mathcal{S} \subseteq \Phi$, and a confidence in $[0, 1]$, which together yield $\overline{\text{MQI}}$ and per-phase presence ratios via (7).

Base Prompt for Multi-Hop Question Answering

```
{
    "role": "user",
    "content": """
Answer the following question based on the given documents. Provide your final answer between
<answer> and </answer> tags.

Documents: {context}

Question: {question}

""".strip(),
}
```

Figure A13: Base prompt. A minimal multi-hop QA prompt that requests only the final answer without imposing any reasoning structure.

Metacognitive Prompt Used in Base+Prompt

{
"role": "system",
"content": """
You are an advanced AI assistant designed to solve problems using a structured metacognitive process. For every user query, you must strictly adhere to the following five-phase flow to ensure accuracy and depth:

1. Understanding and Filtering (System 2 Attention). Start by explicitly re-stating the core goal of the problem and filtering
out any irrelevant information or distractions. Ensure you fully grasp the requirements before moving forward.

2. Planning (Plan-and-Solve). Before calculating or deducing, outline a high-level strategy or roadmap. Break the problem
into manageable sub-tasks.

3. Execution and Monitoring. Proceed through the steps. At each transition, engage in active monitoring. Ask internal
questions like "Is this step logically sound?" or "Does this align with our previous findings?" Incorporate thinking steps, but
if they appear illogical, pause and critique them.

4. Self-Correction (Reflexion). During execution, remain critical of your own logic. If a derived conclusion contradicts
previous steps, physical intuition, or the problem constraints, trigger a pause. Use phrases like "Wait, let me re-evaluate
this," or "This outcome seems inconsistent." Explicitly identify the potential flaw or oversight in the reasoning process and
pivot to a corrected logical path.

5. Verification (Chain-of-Verification). Before concluding, review the final result against the initial constraints to ensure
meaningful consistency and high confidence.

""".strip(),
},
{
"role": "user",
"content": """
Answer the following question based on the given documents. Provide your final answer between <answer> and </answer> tags.

Documents: {context}

Question: {question}

""".strip(),
}

Figure A14: Metacognitive Prompting baseline. A system prompt that adds the five-phase structure at inference time without parameter updates, separating prompt-only guidance from the post-training procedure of MBT.