REHOPER: RECEDING-HORIZON PLANNING FOR ENHANCED REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose ReHoPER, an inference-only, zero-shot method that improves large language models' reasoning by generating and answering intermediate questions along multiple paths before the final answer. It iteratively plans a horizon of candidate intermediate questions, selects one to answer, and replans from the updated history. ReHoPER is task-agnostic, using the same generic instructions across datasets and models without labeled data or task-specific prompt design. Across multiple datasets, including a new controlled benchmark for compositional reasoning, ReHoPER outperforms strong baselines, with the largest gains in the most compositional settings. We will release our implementation and the iLLC generator to support future work.

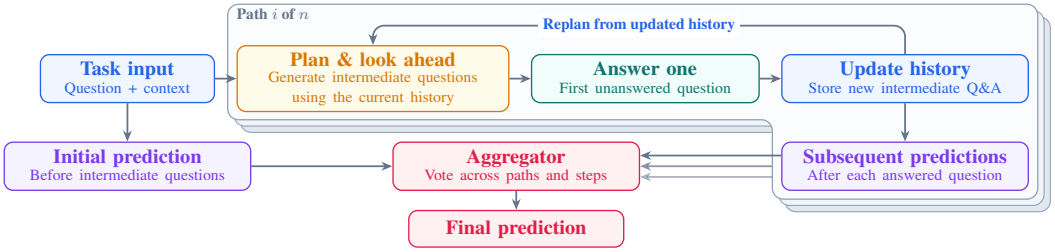


Figure 1: ReHoPER overview. Layered frames indicate independent paths. Each path plans questions, answers one, and replans from updated history. A majority vote over the initial and per-step predictions across paths yields the final prediction.

1 INTRODUCTION

Large Language Models (LLMs) trained at scale can be prompted in natural language to perform a wide range of NLP tasks, achieving strong zero- and few-shot performance, often close to fine-tuned systems, without task-specific training (Brown et al., 2020; Min et al., 2023). Instruction tuning and reinforcement learning from human feedback further improve their ability to follow instructions and generalize to new tasks, reinforcing LLMs as general-purpose, prompt-driven systems rather than narrowly trained models (Wei et al., 2022a; Ouyang et al., 2022). Despite this progress, benchmarks designed to probe multi-step and compositional reasoning, such as MoreHopQA (Schnitzler et al., 2024) and MuSR (Sprague et al., 2024), show that even state-of-the-art models still struggle to integrate multiple pieces of evidence and handle long-range dependencies.

Inference-time reasoning has improved with methods such as Tree-of-Thoughts (Yao et al., 2023), StrategyLLM (Gao et al., 2024), and Self-Ask (Press et al., 2023), which guide models through multi-step procedures like search, strategy retrieval, or follow-up questioning. However, these approaches often depend on labeled data (e.g., solved examples for learning or retrieving strategies) or a manually crafted reasoning structure tailored to each new task. In contrast, Chain-of-Thought (CoT) (Wei et al., 2022b) and Plan-and-Solve (PS) (Wang et al., 2023a) use simple, reusable instructions (e.g., "let's think step by step" or "first outline a plan, then solve the problem") to enable oracle-free reasoning across tasks, and Self-Discover (SD) (Zhou et al., 2024) further reduces manual effort by letting the model generate its own reasoning structure at inference time. Still, these methods mostly follow linear reasoning patterns, and can potentially benefit from richer, more flexible planning mechanisms in the spirit of Tree-of-Thoughts and StrategyLLM, while remaining oracle-free and task-agnostic.

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

In this work, we introduce Receding-Horizon Planning for Enhanced Reasoning (ReHoPER), which targets this design space from a different angle. Rather than prescribing a fixed sequence of explanations or relying on discovered task-specific templates, we guide the model to generate and answer intermediate questions about the current instance before committing to a final prediction. ReHoPER operates entirely at inference time and remains task-agnostic as the same generic instructions for proposing and answering intermediate questions can be used across tasks, models, and answer-format prefixes (Input-Output, CoT, and PS) without any task-specific labeled data or demonstrations. Figure 1 summarizes the planning loop and how predictions from multiple paths are combined.

ReHoPER follows a receding-horizon strategy. At each step, it generates a list of intermediate questions and answers the first unresolved one given the current history, loosely analogous to classical receding-horizon control (Propoi, 1963; Kwon and Pearson, 1977). This alternating loop of planning and execution combines look-ahead with step-by-step adaptivity. Each answer updates the reasoning state, triggering replanning that can revise, refine, or discard previously proposed questions, enabling a dynamically evolving reasoning process rather than a static question sequence. We evaluate ReHoPER against strong baselines on MuSR, MoreHopQA, and iLLC, a new controlled benchmark for compositional execution, using a wide range of LLMs. Across these settings, ReHoPER improves over the compared methods under matched prompting conditions.

Our contributions are as follows: (i) we introduce ReHoPER, a task-agnostic, inference-only method that uses adaptive planning to generate and answer intermediate questions across multiple reasoning paths and aggregates their predictions; (ii) we introduce iLLC, a controlled diagnostic benchmark for compositional execution; (iii) we provide a comprehensive empirical study that compares ReHoPER against strong baselines from prior work across diverse models, datasets, and reasoning styles; (iv) we offer a systematic analysis of ReHoPER design choices and behaviors, through ablations and trajectory analyses; and (v) we will make our implementation and iLLC data generator publicly available to facilitate further research.

2 Related work

Prompting strategies play a central role in eliciting reasoning behavior from LLMs. Early work predominantly evaluated LLMs using Input–Output (IO) prompting, where models directly predict a final answer without exposing intermediate reasoning. This paradigm changed with the introduction of Chain-of-Thought (CoT) prompting, which encourages models to generate explicit reasoning steps in natural language through either few-shot exemplars with rationales (Wei et al., 2022b) or minimal zero-shot instructions such as "Let's think step by step" (Kojima et al., 2022). However, while zero-shot CoT can elicit step-by-step reasoning, it does not require the model to commit to a complete reasoning outline, leading to missing steps and omission errors. Plan-and-Solve (PS) prompting (Wang et al., 2023a) addresses this limitation by decoupling planning from execution: the model first generates an explicit plan and then executes it, effectively encouraging structured reasoning via a "plan, then solve" instruction.

Several methods decompose compositional tasks into subquestions. Self-Ask (Press et al., 2023) generates and answers follow-up questions before producing a final response, while Least-to-Most (LtM) prompting (Zhou et al., 2023) first produces subproblems and then solves them sequentially. However, both rely on curated exemplars and predefined reasoning templates, in which the decomposition is determined upfront and executed in a single pass. Self-Discover (SD) (Zhou et al., 2024) removes manual design by composing a reasoning structure in a zero-shot setting. Still, it commits to a fixed plan before solving, limiting its ability to adapt as new information emerges during reasoning. ReHoPER instead uses generic instructions to generate and answer intermediate questions for each instance, gathers predictions along multiple paths, and aggregates them. Within each path, it selects one question from a short plan before updating the plan from the new answer. This allows subsequent questions to reflect newly obtained information without committing to a fixed decomposition.

This iterative plan–execute–replan paradigm connects ReHoPER to classical planning and control, particularly receding-horizon control (RHC), also known as model predictive control, where decisions are made by optimizing over a short horizon, executing one step, and replanning from the updated state (Propoi, 1963; Kwon and Pearson, 1977; Maher, 2025). However, unlike classical RHC, ReHoPER does not rely on an externally defined reward function; instead, it uses the model's own predictions of the goal question, conditioned on the evolving reasoning state, as an implicit evaluation signal.

Beyond explicit decomposition, several methods enhance LLM reasoning through iterative sampling, structured search, or self-reflection. Self-consistency (Wang et al., 2023b) samples multiple independent CoT trajectories and aggregates their final answers, using diversity to reduce variance. Tree-of-Thoughts (ToT) (Yao et al., 2023) extends this idea by explicitly exploring a tree of intermediate reasoning states using search and evaluation, enabling broader exploration at the cost of increased inference complexity. StrategyLLM (Gao et al., 2024) commits to a high-level plan prior to execution. However, these approaches typically rely on a fixed plan (with no ability to re-plan) or costly search over multiple branches. Self-Refine (Madaan et al., 2023) and CoVe (Dhuliawala et al., 2024) iteratively improve outputs via self-feedback and verification, RISE (He et al., 2025) combines decomposition, retrieval, and self-critique through iterative self-exploration, and Think Twice (Tian et al., 2025) repeatedly re-answers the same question to correct errors. Different from the above approaches, ReHoPER integrates lightweight planning and adaptive question selection within multiple independent forward reasoning trajectories, enabling targeted exploration without handcrafting reasoning paths, manually labeled data, or post-hoc revision.

3 METHODOLOGY

We propose ReHoPER (*Receding-Horizon Planning for Enhanced Reasoning*), an inference-only method that guides an instruction-tuned LLM through intermediate questions before answering the goal question Q^*. ReHoPER runs multiple stochastic reasoning paths and within each path repeatedly looks ahead by generating a horizon of candidate intermediate questions conditioned on the current path history. It then executes only one question, incorporates the resulting answer into the history, and replans from this updated state. This makes the process inherently dynamic: each newly obtained intermediate answer can change what information is missing, render some planned questions irrelevant, or suggest a better follow-up direction. Thus, ReHoPER adapts its reasoning trajectory dynamically, correcting earlier plans and incorporating newly revealed information on the fly without committing to a fixed plan. By combining diversity across paths with per-step look-ahead planning and replanning within each path, ReHoPER maintains an organized decomposition while enabling course correction, leading to more reliable final predictions.

We formalize the task setting in Section 3.1, defining the inference-only objective and notation. In Section 3.2, we then present ReHoPER in detail, describing its receding-horizon planning, path rollout, and aggregation procedure.

3.1 PROBLEM STATEMENT

Each task example consists of an optional context C and a goal question Q^*. Given an instruction-tuned LLM f, the objective is to produce a prediction $\hat{y}$ that matches the ground-truth label y. The proposed method operates in a fully inference-only setting and does not rely on any task-specific instruction, labeled examples, or additional unlabeled data beyond the given task test example.

3.2 REHOPER

ReHoPER runs n stochastic reasoning paths by enabling non-deterministic decoding during model inference (e.g., via temperature sampling), and organizes each path as a sequence of intermediate questions and answers generated in a receding-horizon manner: at every step, it briefly looks ahead to organize a candidate trajectory of questions, but executes only one newly selected question before replanning from the updated history. Intermediate questions are generated by prompting the model with a specific instruction I (Appendix L). For path $i \in \{1, \ldots, n\}$, let m_i denote the number of intermediate questions ultimately asked along that path. The history of path i at step $k \in \{0, \ldots, m_i\}$ is

$$\mathbf{H}_k^{(i)} = \left\langle (IQ_1^{(i)},\ A_1^{(i)}),\ \ldots,\ (IQ_k^{(i)},\ A_k^{(i)}) \right\rangle, \quad \text{where } \mathbf{H}_{k=0}^{(i)} = \emptyset. \tag{1}$$

Here, $(IQ_k^{(i)}, A_k^{(i)}), k = 1..m_i$ represent intermediate (asked/answered) questions and their answers, respectively. For simplicity, we denote $\mathbf{H}_{k=0}^{(i)} = \mathbf{H}_0$. I_{rh} is receding-horizon instruction.

Initial prediction. Prompt f once with context, empty history, and goal question $(C,\ \mathbf{H}_0,\ Q^*)$ to obtain response R_0 and post-process R_0 to compute an initial prediction $\hat{y}_0$.

Path rollout. For each path i, for steps $k = 1, 2, \ldots$ until termination (m_i steps or a user-defined cap of $k = m_{\max}$):

I. **Plan.** Prompt f with $(C,\ \mathbf{H}_{k-1}^{(i)},\ I_{\mathrm{rh}})$ to instruct the model to generate a horizon of candidate questions: $\widetilde{\mathbf{IQ}}_k^{(i)} = \langle \widetilde{IQ}_{k,1}^{(i)},\ \ldots,\ \widetilde{IQ}_{k,h}^{(i)} \rangle$, where h is determined by f and may vary across steps and paths; $\widetilde{\mathbf{IQ}}_k^{(i)}$ is the set of generated questions.

II. **Select.** Choose the first previously unanswered candidate question from $\widetilde{\mathbf{IQ}}_k^{(i)}$:

$$j^* = \min\Big\{ j : \widetilde{IQ}_{k,j}^{(i)} \notin \{IQ_1^{(i)},\ \ldots,\ IQ_{k-1}^{(i)}\} \Big\}, \tag{2}$$

and set $IQ_k^{(i)} \leftarrow \widetilde{IQ}_{k,j^*}^{(i)}$. If no such j^* exists (all candidates repeat or no valid question is produced), the path terminates. The remaining questions $\widetilde{\mathbf{IQ}}_k^{(i)} \setminus \langle IQ_k^{(i)} \rangle$, where $\setminus$ denotes difference, are discarded, and a fresh plan is generated at the next step. Note that if $k = 1$ in Eq. 2, $j^* = 1$.

III. **Answer.** Prompt f with $(C,\ \mathbf{H}_{k-1}^{(i)},\ IQ_k^{(i)})$ to get $A_k^{(i)}$.

IV. **Update.** Set $\mathbf{H}_k^{(i)} \leftarrow \mathbf{H}_{k-1}^{(i)} \parallel \langle (IQ_k^{(i)},\ A_k^{(i)}) \rangle$, where $\parallel$ denotes concatenation of sequences.

V. **Predict.** Prompt f with $(C,\ \mathbf{H}_k^{(i)},\ Q^*)$ to obtain response $R_k^{(i)}$, and post-process to extract candidate prediction $\hat{y}_k^{(i)}$ for step k of path i.

Aggregation. Collect the list of predictions, from the initial step and all the steps k of all paths i:

$$\langle \hat{y}_0 \rangle \ \parallel\ \Big(\|_{i=1}^{n}\ \|_{k=1}^{m_i} \langle \hat{y}_k^{(i)} \rangle \Big). \tag{3}$$

Since $\mathbf{H}_0 = \emptyset$ is identical across all paths, the prediction at step $k = 0$ is counted at most once in the final aggregation. We apply a simple majority vote (ties broken alphabetically) over the predictions to incorporate a form of Self-Consistency across reasoning paths and intermediate steps: the final prediction $\hat{y}$ is defined as the most frequent value, and is evaluated against the ground-truth label y.

4 Experimental setup

4.1 Datasets and tasks

We evaluate ReHoPER on three reasoning benchmarks spanning long-context narrative inference, multi-hop evidence aggregation, and controlled compositional execution: MuSR (Sprague et al., 2024), MoreHopQA (Schnitzler et al., 2024), and our proposed i-th to Last Letter Concatenation (iLLC) benchmark. **MuSR** includes three long-context reasoning tasks, **MM** (Murder Mysteries), **OP** (Object Placement), and **TA** (Team Allocation), each requiring models to integrate scattered clues and constraints. **MoreHopQA** tests multi-hop question answering over contexts that require connecting multiple pieces of evidence. We follow the original setup by evaluating on the 150-example subset selected by the dataset authors. Finally, **iLLC** generalizes the established Last Letter Concatenation (LLC) (Wei et al., 2022b) by requiring models to extract the i-th-to-last character from each word and concatenate the resulting letters; $i = 1$ recovers standard LLC. We denote each subset as L$\{i\}$-$\{w\}$, where i is the target character position from the end and w is the number of words in the sequence. Our main experiments use **L1-4**, **L1-6**, **L2-4**, and **L2-6**, while additional results on the longer **L1-8** and **L2-8** settings are reported in Appendix J.

We use iLLC to study compositional execution in a setting where the required operations are known. Solving an instance requires extracting one character from each word, preserving the order of the words, and combining the characters into the final answer. This structure fits our study of intermediate-question planning because the problem can be broken into well-defined steps. It also lets us vary difficulty in a controlled way: i changes the character-extraction operation, while w changes the number of operations that must be composed. Since the task does not depend on external knowledge, its results are less affected by differences in factual recall or retrieval. In addition, the generator can produce new labeled instances at different difficulty levels, making the benchmark easy to scale

and extend. We use iLLC to test procedural compositional reasoning under controlled conditions, complementing the broader reasoning settings covered by the natural-language benchmarks. Further details on iLLC and the other datasets, including statistics, examples, and post-processing procedures, are provided in Appendix M.

4.2 LANGUAGE MODELS

We evaluate ReHoPER on a diverse set of instruction-tuned LLMs spanning multiple model families, parameter scales, and deployment regimes, including dense models from Llama, Gemma, Phi, and Mistral, as well as sparse Mixture-of-Experts models such as gpt-oss and Qwen3. To make larger models practical at inference time, we use quantized AWQ or GPTQ variants when needed. For Qwen3, we evaluate both its non-thinking mode and its thinking-enabled mode; unless a result is explicitly labeled as Qwen3 with thinking, Qwen3 denotes the non-thinking configuration. Full model and deployment details are provided in Appendix K, Table 16.

4.3 BASELINES AND SETUP DETAILS

We use three different assistant output prefixes: IO (input-output), CoT (chain-of-thought) (Wei et al., 2022b), and PS (plan-and-solve) (Wang et al., 2023a). Concretely, **IO** leaves the assistant output unconstrained (empty prefix), while **CoT** prepends the assistant's response with "Let's think step by step." and **PS** prepends it with "Let's first understand the problem and devise a plan to solve the problem. Then, let's carry out the plan to solve the problem step by step.". Notably, adding CoT or PS explicitly forces a particular reasoning style and thus can override or mask the model's native reasoning behavior that would otherwise emerge under IO. For each prompt variant, we pair it with **Self-Consistency (SC)** (Wang et al., 2023b) as our first baseline by sampling 61 independent completions (same prompt with stochastic decoding), post-processing each completion into a candidate answer, and returning the majority-vote answer among the candidates (ties broken deterministically by choosing the alphabetically smallest answer).

As another baseline, we include the **Self-Discover (SD)** (Zhou et al., 2024), adapted to our instance-specific setup. For each test instance, we first prompt the model using only that single unlabeled example to propose a high-level reasoning strategy and decomposition tailored to that instance (e.g., what intermediate checks to perform, what constraints to track, how to verify the final answer). We then turn this into a one-off reasoning template and use it to answer the same instance. For completeness on iLLC, we also compare against **Least-to-Most (LtM)** prompting, its original **CoT** baseline (Zhou et al., 2023), and **Self-Refine (SR)** (Madaan et al., 2023). For SR, we use instruction-only feedback and refinement prompts for five iterations, since no few-shot exemplars are available for our datasets. We treat these as analysis-only references rather than core baselines, as they rely on labeled exemplars, manually curated reasoning structures, or task-specific refinement prompts.

ReHoPER is designed to be prefix-agnostic and compatible with all SC assistant output prefix variants. Concretely, ReHoPER can be combined with {IO, CoT, PS} by simply applying the chosen prefix during the evaluation stages, yielding composite configurations of the form ReHoPER $\times$ {IO, CoT, PS} (e.g., ReHoPER with CoT, etc.). This design allows us to flexibly adopt the prefix that best matches the requirements of a given task, while keeping the underlying reasoning and evaluation procedure unchanged. We use $n = 6$ reasoning paths, a maximum depth of $m_{\max} = 10$ steps, and temperature 0.5. This configuration produces up to $1 + n \cdot m_{\max} = 61$ candidate answers per instance (one from the initial evaluation plus up to $n \cdot m_{\max}$ intermediate evaluation outputs), enabling a majority vote analogous to the 61-sample SC baselines. We choose $n = 6$ and $m_{\max} = 10$ as a pragmatic trade-off that balances breadth (path diversity) and depth (multi-step refinement). Appendix G reports the complete hyperparameter study, including the selected temperature and path-step budget.

For the planning ablations, we evaluate ReHoPER without replanning, generating the intermediate-question list once per path and answering it in order, and without look-ahead, generating a single next question from the updated history at each step. Both use the same answering, history-update, prediction, and aggregation procedures, output prefixes, and path and step limits as the full method. The former stops when its initial list is exhausted or the step limit is reached; the latter runs to the step limit. Both replace the candidate-selection step. We evaluate these ablations on MuSR MM, MoreHopQA, and iLLC L2-4; Appendix I gives implementation details, and Appendix L provides the prompts.

Table 1: Accuracy (%) of ReHoPER (RH) on MuSR, MoreHopQA, and iLLC benchmarks under IO, CoT, and PS prefixes. SD refers to the Self-Discover baseline. SC denotes simple question answering with Self-Consistency using the corresponding output prefix, while RH applies our planning approach with that prefix. Green, red, and gray indicate performance changes of ReHoPER relative to SC. The bottom-right block reports averages over the eight displayed subsets for each model and prompt.

	MuSR MM							MuSR OP							MuSR TA						
		IO		CoT		PS			IO		CoT		PS			IO		CoT		PS	
Model	**SD**	**SC**	**RH**	**SC**	**RH**	**SC**	**RH**	**SD**	**SC**	**RH**	**SC**	**RH**	**SC**	**RH**	**SD**	**SC**	**RH**	**SC**	**RH**	**SC**	**RH**
Gemma 3	57.6	54.4	61.2	61.2	62.0	61.6	65.6	50.0	50.4	55.1	51.6	55.5	54.7	55.9	50.0	58.8	47.6	56.0	51.2	58.0	56.0
gpt-oss	74.8	70.8	76.0	70.4	76.4	73.2	76.8	53.9	55.5	53.9	53.1	53.5	57.0	53.5	70.8	69.2	68.0	56.4	63.2	65.2	68.0
Llama 3.1 8B	54.0	56.8	62.4	56.4	60.8	57.2	63.6	44.5	49.2	53.1	52.0	48.4	52.0	48.4	34.8	35.2	43.6	43.2	47.6	45.2	47.6
Llama 3.1 70B	64.8	65.6	71.2	66.0	70.4	66.8	71.2	42.6	46.1	55.9	46.1	54.3	48.4	55.9	57.6	62.0	57.6	62.0	59.6	60.4	59.2
Mistral Small	58.8	60.4	73.2	68.4	78.0	70.8	79.6	48.8	50.4	47.3	46.5	50.0	47.7	50.0	51.6	48.4	58.8	51.6	62.0	55.6	58.8
Phi-4	72.0	75.6	74.8	74.8	76.0	78.8	75.2	44.1	52.3	50.0	48.0	50.0	47.7	52.3	56.0	56.0	56.4	55.6	61.6	56.8	62.4
Qwen3 w/o Thinking	60.4	58.4	72.0	67.2	71.6	70.8	72.8	40.2	53.1	51.6	50.8	54.7	50.8	51.6	59.2	57.2	64.0	71.2	68.8	68.0	68.8
Qwen3 w/ Thinking	70.4	72.0	72.4	67.2	72.0	70.8	73.6	49.2	52.0	57.4	50.8	53.5	50.8	53.5	70.0	71.6	66.4	71.2	70.0	68.0	67.2
Average	64.1	64.3	70.4	66.5	70.9	68.8	72.3	46.7	51.1	53.0	49.9	52.5	51.1	52.6	56.3	57.3	57.8	58.4	60.5	59.7	61.0

	iLLC L1-4							iLLC L1-6							MoreHopQA						
		IO		CoT		PS			IO		CoT		PS			IO		CoT		PS	
Model	**SD**	**SC**	**RH**	**SC**	**RH**	**SC**	**RH**	**SD**	**SC**	**RH**	**SC**	**RH**	**SC**	**RH**	**SD**	**SC**	**RH**	**SC**	**RH**	**SC**	**RH**
Gemma 3	73.6	20.8	46.0	82.8	83.2	90.4	88.0	46.0	10.0	34.4	56.0	62.8	74.0	66.0	44.0	61.3	64.7	64.7	68.7	66.7	74.0
gpt-oss	78.8	90.0	87.6	95.6	88.4	96.0	88.0	66.8	83.2	79.6	85.2	78.8	82.4	80.0	39.3	80.7	78.7	81.3	80.0	79.3	81.3
Llama 3.1 8B	46.8	74.8	70.0	67.6	67.6	80.0	77.6	18.8	50.0	40.8	53.2	52.8	56.0	52.8	34.0	59.3	57.3	61.3	60.7	64.0	63.3
Llama 3.1 70B	93.2	89.6	89.2	88.8	91.2	98.0	97.6	77.6	73.2	77.2	71.6	76.8	83.2	86.4	62.0	68.0	71.3	68.0	71.3	68.7	74.0
Mistral Small	70.4	88.4	90.4	92.0	92.4	91.6	91.2	42.8	73.2	69.6	76.8	72.0	78.4	78.8	54.7	70.0	71.3	70.0	70.0	71.3	72.7
Phi-4	88.0	89.6	91.2	92.8	92.8	91.2	92.0	68.8	68.8	72.4	76.4	80.4	74.4	76.0	64.0	77.3	77.3	76.0	80.7	77.3	76.7
Qwen3 w/o Thinking	71.6	82.4	85.2	79.2	84.4	83.6	86.0	60.4	62.4	80.0	70.0	78.8	75.6	80.0	34.0	70.0	68.0	69.3	70.7	73.3	72.0
Qwen3 w/ Thinking	1.2	97.2	97.2	79.2	95.6	83.6	95.6	0.8	87.2	90.0	70.0	86.8	75.6	85.6	42.0	75.3	80.0	69.3	78.7	73.3	78.0
Average	65.5	79.1	82.1	84.8	87.0	89.3	89.5	47.8	63.5	68.0	69.9	73.7	75.0	75.7	46.8	70.3	71.1	70.0	72.6	71.8	74.0

	iLLC L2-4							iLLC L2-6							Average						
		IO		CoT		PS			IO		CoT		PS			IO		CoT		PS	
Model	**SD**	**SC**	**RH**	**SC**	**RH**	**SC**	**RH**	**SD**	**SC**	**RH**	**SC**	**RH**	**SC**	**RH**	**SD**	**SC**	**RH**	**SC**	**RH**	**SC**	**RH**
Gemma 3	1.6	7.2	6.0	7.2	6.4	3.6	7.6	0.4	0.8	0.8	1.2	1.6	2.0	0.8	40.4	33.0	39.5	47.6	48.9	51.4	51.7
gpt-oss	67.2	95.2	93.6	81.6	93.2	40.0	94.0	56.0	86.4	86.0	67.2	84.4	14.0	84.0	63.5	78.9	77.9	73.9	77.2	63.4	78.2
Llama 3.1 8B	0.0	9.6	12.8	9.6	18.4	28.8	41.2	0.0	3.6	8.0	4.8	8.0	5.6	14.4	29.1	42.3	43.5	43.5	45.5	48.6	51.1
Llama 3.1 70B	8.0	13.6	78.4	36.4	78.8	47.6	80.8	3.6	4.8	68.0	17.6	63.2	29.2	74.4	51.2	52.9	71.1	57.1	70.7	62.8	74.9
Mistral Small	0.4	4.0	11.2	2.0	11.6	4.4	12.0	0.0	0.0	1.6	0.0	2.4	0.0	2.0	40.9	49.3	52.9	50.9	54.8	52.5	55.6
Phi-4	14.0	12.0	54.0	10.8	62.8	10.4	61.2	6.0	4.4	27.2	3.6	38.0	6.4	41.6	51.6	54.5	62.9	54.8	67.8	55.4	67.2
Qwen3 w/o Thinking	1.2	72.4	81.6	54.0	84.4	86.8	84.8	0.4	36.0	59.6	23.2	62.0	72.8	68.0	40.9	61.5	70.2	60.6	71.9	72.7	73.0
Qwen3 w/ Thinking	1.2	92.4	93.6	54.0	91.6	86.8	92.4	1.6	85.2	84.4	23.2	84.4	72.8	84.8	29.6	79.1	80.2	60.6	79.1	72.7	78.8
Average	11.7	38.3	53.9	32.0	55.9	38.6	59.3	8.5	27.7	42.0	17.6	43.0	25.4	46.3	43.4	56.4	62.3	56.1	64.5	59.9	66.3

5 Results and Discussion

5.1 ReHoPER results

Table 1 compares ReHoPER with SC and SD across all datasets. Across all models, datasets, and prefixes, ReHoPER improves over the SC setup under the same answer-format prefix in a majority of cases, with stronger gains for more structured prefixes. Appendix N shows examples of how ReHoPER generates and uses intermediate questions in both correct and incorrect cases. On average (bottom-right block), ReHoPER improves accuracy by 5.8pp for IO, 8.4pp for CoT, and 6.4pp for PS. These gains are calculated before rounding the accuracies and show that our approach is beneficial even when models already produce explicit chains of thought. Appendix A shows that ReHoPER also improves over SC when each benchmark family is given equal weight, with gains of 3.9pp for IO, 6.6pp for CoT, and 4.8pp for PS. Appendix H shows that these gains are stable across repeated runs. ReHoPER also often recovers and surpasses IO-level performance even when raw CoT or PS lags behind, indicating that the full intermediate-question procedure can help beyond the surface prompt alone. Notably, ReHoPER tends to be strongest and most reliable when paired with PS, making it a suitable default for new tasks where it is unclear which prefix works best.

Remarkably, the gains are most pronounced on the hardest, most compositional settings. On the iLLC L2-$\{w\}$ variants, where the SC baseline performance is low, ReHoPER yields large improvements, e.g., 14.3pp for IO, 25.4pp for CoT, and 20.9pp for PS on L2-6. We also observe near-consistent gains on MuSR MM, with average accuracy increasing by 6.2pp for IO, 4.5pp for CoT, and 3.6pp for PS, and smaller but positive improvements on other MuSR subsets and MoreHopQA. In contrast, tasks that are either more straightforward or closer to ceiling (e.g., iLLC L1-$\{w\}$ variants) show only modest gains and occasionally small regressions, suggesting limited headroom for further improvement through additional planning.

The iLLC results also show that these controlled changes produce meaningful differences in difficulty. Performance drops substantially from L1, where the target character is at the word boundary, to

Table 2: Planning ablations without replanning (-RP), without look-ahead (-LA), and the full ReHoPER method (RH) across three prompting styles, IO, CoT, and PS, on MuSR MM, MoreHopQA, and iLLC L2-4. Values are accuracies (%), and green and light green mark the best and second best variant, respectively, for each (model, dataset, prefix) triplet. The bottom-right block reports averages over all datasets for each model and prompt.

	MuSR MM									MoreHopQA								
	IO			CoT			PS			IO			CoT			PS		
Model	**-RP**	**-LA**	**RH**	**-RP**	**-LA**	**RH**	**-RP**	**-LA**	**RH**	**-RP**	**-LA**	**RH**	**-RP**	**-LA**	**RH**	**-RP**	**-LA**	**RH**
Gemma 3	57.6	58.8	61.2	59.2	58.4	62.0	62.4	62.4	65.6	64.0	66.0	64.7	66.7	66.7	68.7	70.0	69.3	74.0
gpt-oss	72.4	70.0	76.0	72.0	68.0	76.4	74.0	71.2	76.8	80.7	78.7	78.7	79.3	79.3	80.0	80.7	78.7	81.3
Llama 3.1 8B	61.6	58.4	62.4	61.6	61.2	60.8	64.8	60.4	63.6	52.0	57.3	57.3	56.7	58.7	60.7	60.7	66.0	63.3
Llama 3.1 70B	70.4	70.0	71.2	72.0	72.4	70.4	71.6	71.2	71.2	68.7	74.0	71.3	70.7	72.7	71.3	72.7	74.0	74.0
Mistral Small	70.0	70.4	73.2	74.8	77.2	78.0	77.2	79.6	79.6	66.0	68.7	71.3	68.0	68.0	70.0	71.3	72.7	72.7
Phi-4	74.4	75.2	74.8	77.2	77.6	76.0	78.0	79.6	75.2	76.7	74.7	77.3	78.7	76.7	80.7	77.3	76.0	76.7
Qwen3	74.0	76.8	72.0	72.8	78.0	71.6	75.2	77.2	72.8	66.7	65.3	68.0	69.3	68.0	70.7	69.3	70.7	72.0
Average	68.6	68.5	70.1	69.9	70.4	70.7	71.9	71.7	72.1	67.8	69.2	69.8	69.9	70.0	71.7	71.7	72.5	73.4

	iLLC L2-4									Average								
	IO			CoT			PS			IO			CoT			PS		
Model	**-RP**	**-LA**	**RH**	**-RP**	**-LA**	**RH**	**-RP**	**-LA**	**RH**	**-RP**	**-LA**	**RH**	**-RP**	**-LA**	**RH**	**-RP**	**-LA**	**RH**
Gemma 3	4.8	8.8	6.0	5.2	9.2	6.4	3.2	7.2	7.6	42.1	44.5	44.0	43.7	44.8	45.7	45.2	46.3	49.1
gpt-oss	92.8	93.6	93.6	89.6	93.6	93.2	86.4	92.8	94.0	82.0	80.8	82.8	80.3	80.3	83.2	80.4	80.9	84.0
Llama 3.1 8B	2.8	21.6	12.8	5.6	12.4	18.4	10.0	36.8	41.2	38.8	45.8	44.2	41.3	44.1	46.6	45.2	54.4	56.0
Llama 3.1 70B	76.0	80.8	78.4	77.6	82.0	78.8	86.0	83.2	80.8	71.7	74.9	73.6	73.4	75.7	73.5	76.8	76.1	75.3
Mistral Small	2.8	4.0	11.2	2.4	4.8	11.6	3.2	2.8	12.0	46.3	47.7	51.9	48.4	50.0	53.2	50.6	51.7	54.8
Phi-4	20.8	28.8	54.0	45.6	30.0	62.8	40.8	33.2	61.2	57.3	59.6	68.7	67.2	61.4	73.2	65.4	62.9	71.0
Qwen3	84.0	75.6	81.6	83.2	80.8	84.4	85.6	84.0	84.8	74.9	72.6	73.9	75.1	75.6	75.6	76.7	77.3	76.5
Average	40.6	44.7	48.2	44.2	44.7	50.8	45.0	48.6	54.5	59.0	60.8	62.7	61.3	61.7	64.4	62.9	64.2	66.7

L2, where its absolute position depends on word length. Performance also generally declines as the number of words increases. ReHoPER performs best relative to the baselines in these harder settings, suggesting that intermediate-question planning becomes more useful as extraction and composition become more demanding. These results demonstrate the value of iLLC as a controlled diagnostic of procedural compositional reasoning, complementing the broader reasoning settings represented by MuSR and MoreHopQA.

On average, SC already tends to outperform SD on all benchmarks, and ReHoPER further widens that gap. In the final Average block, SD reaches 43.4% overall, whereas SC ranges from 56.1–59.9% depending on the prefix, and ReHoPER raises these results to 62.3–66.3%. Overall, the table supports the view that ReHoPER is a robust, model- and prompt-agnostic planner that strengthens test-time reasoning, with its largest impact in the deepest compositional settings, even rescuing cases where naive CoT or PS prompting alone would underperform a simple IO baseline.

5.2 Effect of replanning and look-ahead

Table 2 compares ReHoPER with ablations without replanning or look-ahead on MuSR MM, MoreHopQA, and iLLC L2-4. Across models and datasets, ReHoPER exceeds the stronger ablation by 1.9pp, 2.7pp, and 2.5pp under IO, CoT, and PS, respectively. The advantage holds for each dataset and prefix average and is largest on iLLC L2-4 (3.5pp, 6.1pp, and 5.9pp for IO, CoT, and PS), suggesting that the combined planning design is particularly useful in this more compositional setting. The overall averages also show that generating each next question from the updated history, even without look-ahead, performs better than following a fixed initial plan: the ablation without look-ahead achieves 60.8%, 61.7%, and 64.2% under IO, CoT, and PS, compared with 59.0%, 61.3%, and 62.9% without replanning. This pattern is consistent with the benefit of adapting question generation to newly acquired information. ReHoPER improves further on these averages by combining history-conditioned replanning with a horizon of candidate questions and explicit selection of an unasked question. These aggregate trends support the combined design, although individual configurations can favor an ablation. Appendix I details the ablation procedures; Appendix E reports trajectory lengths and within-path question diversity; Appendix F shows that replanning often changes the initial question trajectory.

5.3 Effect of answer aggregation

Figure 2 separates improvements in predictions produced by ReHoPER+PS during reasoning from the gains due to aggregating them on iLLC L2-4. In the upper row, both the mean correctness of

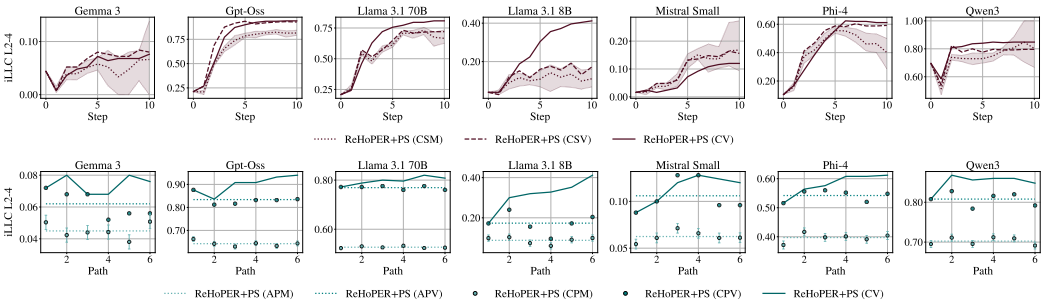


Figure 2: Answer aggregation for ReHoPER+PS on iLLC L2-4 across models. Top: accuracy over reasoning steps; bottom: accuracy over stochastic paths. CV is cumulative voting, CSV voting over each path's latest prediction, and CSM their mean correctness. CPV/CPM are voting/mean correctness within a path; APV/APM are their averages across paths.

the latest prediction from each path and the vote over those latest predictions tend to rise as more intermediate questions are answered. Later reasoning states are therefore often better in their own right. The cumulative vote, which retains predictions from earlier steps, typically rises further and finishes above the latest-prediction vote; this suggests that earlier states still contribute useful votes even when the current predictions have improved. In the lower row, the vote within an individual path varies across trajectories, while voting over all paths generally improves as paths are added and tends to finish above the path-local summaries. Thus, the underlying upward trend is strengthened by retaining and combining predictions across steps and paths. Appendices B and C define the measures and report the full results across models, output prefixes, and more datasets.

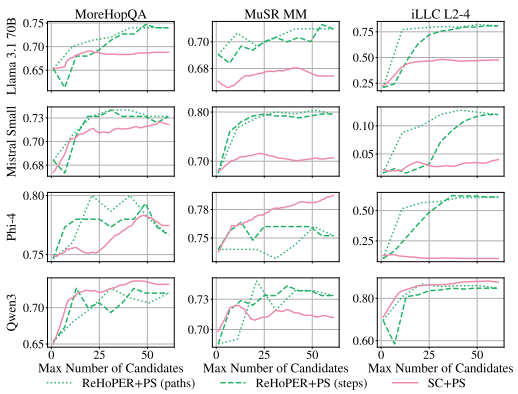


Figure 3: Accuracy from allocating iterations to steps vs. paths.

Table 3: Accuracy (%) of ReHoPER+PS (RH), Least-to-Most (LtM), its proposed few-shot CoT baseline, and Self-Refine (SR).

	iLLC L1-4				iLLC L2-4			
Model	**CoT**	**LtM**	**SR**	**RH**	**CoT**	**LtM**	**SR**	**RH**
Llama 3.1 70B	90.0	95.2	95.2	**97.6**	13.2	12.0	28.0	**80.8**
Mistral Small	0.0	81.6	86.8	**91.2**	0.0	2.0	0.8	**12.0**
Phi-4	90.0	**92.8**	**92.8**	92.0	8.8	6.4	10.0	**61.2**
Qwen3	53.2	45.6	79.2	**86.0**	0.8	3.6	51.6	**84.8**
	iLLC L1-6				**iLLC L2-6**			
Model	**CoT**	**LtM**	**SR**	**RH**	**CoT**	**LtM**	**SR**	**RH**
Llama 3.1 70B	65.6	85.2	83.6	**86.4**	2.0	5.6	15.6	**74.4**
Mistral Small	0.0	58.0	69.2	**78.8**	0.0	0.0	0.0	**2.0**
Phi-4	46.4	**76.0**	72.4	**76.0**	0.0	0.8	3.6	**41.6**
Qwen3	0.0	8.8	64.8	**80.0**	0.0	0.4	23.6	**68.0**
	iLLC L1-8				**iLLC L2-8**			
Model	**CoT**	**LtM**	**SR**	**RH**	**CoT**	**LtM**	**SR**	**RH**
Llama 3.1 70B	24.8	**75.6**	67.6	70.4	0.4	1.2	6.4	**54.0**
Mistral Small	0.0	53.2	54.8	**59.6**	**0.0**	**0.0**	**0.0**	**0.0**
Phi-4	0.0	**66.4**	60.4	61.2	0.0	0.0	0.0	**35.6**
Qwen3	0.0	2.0	48.4	**72.0**	0.0	0.0	16.8	**43.6**

5.4 Paths vs. steps

Figure 3 compares how ReHoPER+PS and SC+PS improve as more candidate predictions are included in the vote. For ReHoPER, the dashed curves add reasoning steps across six paths, while the dotted curves add paths with a fixed maximum depth of ten steps; the solid curve adds independent SC samples. This comparison shows how the choice between deeper reasoning and more paths affects accuracy as the candidate budget grows. Appendix D gives the exact vote construction and results for other models and output prefixes. ReHoPER outperforms SC in most configurations, often from small iteration budgets onward. Across datasets and models, both ReHoPER and SC exhibit diminishing returns as gains per additional step or path become small beyond a certain point. For ReHoPER, one contributing factor is that some paths terminate early, so later steps contribute fewer new candidates. The figure also shows that the relative value of steps versus paths is task dependent. On iLLC, the dotted curves (more paths at fixed depth) usually rise faster than the dashed curves in the first few iterations, indicating that the earliest paths provide the largest marginal benefit; however, they plateau quickly while the dashed curves continue to improve, suggesting that deeper reasoning

steps are ultimately more beneficial on this dataset. In contrast, for example, in MuSR with Phi-4, configurations that emphasize more paths can be advantageous, and the upward trend of the SC curve (figuratively, many step-0 paths) is consistent with this behavior.

5.5 Comparison with Least-to-Most and Self-Refine

Table 3 compares ReHoPER+PS with Least-to-Most (LtM) (Zhou et al., 2023), its few-shot CoT baseline, and Self-Refine (SR) (Madaan et al., 2023) on iLLC. For L1, we use the original CoT and LtM prompts in the LtM paper (Zhou et al., 2023); for L2, we construct analogous prompts following the same template to isolate differences in reasoning rather than prompt design. In SR, feedback and refinement are implemented as few-shot prompts in the original work, but no exemplars exist for our datasets and creating task-specific ones would defeat the purpose of our zero-shot setting. We therefore use instruction-only feedback and refinement prompts, perform five refinement iterations, and take the final refined answer. SR improves over LtM in several cases, but does not match ReHoPER. Our method performs best overall and is particularly stronger on the more compositional L2 variants, where LtM, CoT, and SR degrade sharply. This indicates that while post-hoc refinement helps, explicit planning with intermediate questions is more effective for compositional reasoning even in a task-agnostic, zero-shot setting.

Table 4: Runtime (h:mm) and accuracy (%) comparison between Self-Consistency (SC_{61}) and ReHoPER (RH) across iLLC L1 and L2 subsets. SC_{181} uses 181 samples as a higher-compute SC comparison.

		iLLC L1-4			iLLC L1-6			iLLC L2-4			iLLC L2-6		
Model		**SC_{61}**	**SC_{181}**	**RH**	**SC_{61}**	**SC_{181}**	**RH**	**SC_{61}**	**SC_{181}**	**RH**	**SC_{61}**	**SC_{181}**	**RH**
Llama 3.1 70B	Runtime	1:16	3:37	2:42	1:26	4:23	4:12	2:11	6:23	7:56	2:37	9:04	11:15
	Accuracy	92.1	92.4	**92.7**	76.0	76.9	**80.1**	32.5	35.1	**79.3**	17.2	16.7	**68.5**
Mistral Small	Runtime	0:38	1:38	0:54	0:43	2:03	0:55	0:48	2:08	1:15	0:56	2:41	1:13
	Accuracy	90.7	90.8	**91.3**	76.1	**76.9**	73.5	3.5	2.4	**11.6**	0.0	0.1	**2.0**
Phi-4	Runtime	0:45	1:46	1:06	0:52	2:31	1:57	0:50	2:01	1:18	1:00	2:55	2:21
	Accuracy	91.2	91.2	**92.0**	73.2	73.5	**76.3**	11.1	10.8	**59.3**	4.8	4.4	**35.6**
Qwen3	Runtime	0:31	1:08	0:42	0:36	1:43	1:06	0:48	1:44	1:20	0:57	2:08	1:39
	Accuracy	81.7	81.7	**85.2**	69.3	68.5	**79.6**	71.1	70.5	**83.6**	44.0	44.0	**63.2**

5.6 Compute–accuracy tradeoffs

Table 4 reports runtime and accuracy on iLLC, with accuracy averaged over IO/CoT/PS. Here, we denote the original SC setup used in the main experiments as SC_{61}, and include SC_{181}, which increases the sampling budget from 61 to 181 samples to better match ReHoPER's runtime and number of forward passes. For efficiency, we batch the three prefix-specific conversations in a single inference request for all methods; in ReHoPER, planning and intermediate questions are shared across prefixes, and prefixes are applied only at final prediction. Increasing the sampling budget yields limited gains: consistent with Figure 3, SC_{181} is only slightly better and sometimes worse than SC_{61}, indicating that sampling more final answers is not the main limitation of Self-Consistency. In contrast, ReHoPER achieves larger improvements by allocating compute to planning and intermediate reasoning, with gains evident on L1 and substantially larger on L2. SC_{181} also incurs higher runtime, often exceeding ReHoPER while rarely matching its accuracy. Overall, ReHoPER's advantage comes from structured computation via receding-horizon planning, not simply more forward passes.

6 Conclusion and future work

In this work, we introduced ReHoPER, a task-agnostic, inference-only method that reasons through intermediate questions across multiple paths and aggregates their predictions. It improves over Self-Consistency and other strong baselines under the same answer-format prefix in most settings, with the largest gains on harder compositional tasks. Our controlled iLLC benchmark helps expose how performance changes with compositional difficulty, while planning ablations examine design choices within the method. An interesting direction of future work will be to make the compute budget more adaptive (e.g., dynamically increasing the number of paths when some terminate early). We will make our code and iLLC publicly available on GitHub.

AI USE STATEMENT

We used generative AI tools to help look for additional related work in the literature, polish the writing of this paper, and assist with parts of the implementation and debugging. The authors reviewed all AI-assisted changes and take responsibility for the final manuscript, code, and results.

REPRODUCIBILITY STATEMENT

Section 3.2 specifies the ReHoPER planning, rollout, termination, and answer-aggregation procedure. Section 4 describes the datasets, baselines, output prefixes, and shared inference settings. The appendix provides the full intermediate-question instructions (Appendix L), dataset selection and answer post-processing (Appendix M), exact model checkpoints and deployment details (Appendix K), and the hyperparameter study (Appendix G). Repeated-run results with means and standard deviations are reported in Appendix H. We plan to release the iLLC generator and the ReHoPER implementation upon publication.

REFERENCES

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Comput. Surv.*, 56(2), September 2023. ISSN 0360-0300. doi: 10.1145/3605943. URL https://doi.org/10.1145/3605943.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*, 2022a. URL https://openreview.net/forum?id=gEZrGCozdqR.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Julian Schnitzler, Xanh Ho, Jiahao Huang, Florian Boudin, Saku Sugawara, and Akiko Aizawa. Morehopqa: More than multi-hop reasoning. *ArXiv*, abs/2406.13397, 2024. URL https://api.semanticscholar.org/CorpusID:270620469.

Zayne Rea Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. MuSR: Testing the limits of chain-of-thought with multistep soft reasoning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jenyYQzue1.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: deliberate problem solving with large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Chang Gao, Haiyun Jiang, Deng Cai, Shuming Shi, and Wai Lam. Strategyllm: Large language models as strategy generators, executors, optimizers, and evaluators for problem solving. *Advances in Neural Information Processing Systems*, 37:96797–96846, 2024.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5687–5711, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.378. URL https://aclanthology.org/2023.findings-emnlp.378/.

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

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 Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada, July 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.147. URL https://aclanthology.org/2023.acl-long.147/.

Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V Le, Ed H. Chi, Denny Zhou, Swaroop Mishra, and Steven Zheng. SELF-DISCOVER: Large language models self-compose reasoning structures. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=BROvXhmzYK.

A. I. Propoi. Application of linear programming methods for the synthesis of automatic sampled-data systems. *Avtomat. i Telemekh.*, 24(7):912–920, 1963. URL http://mi.mathnet.ru/at11905.

W. Kwon and A. Pearson. A modified quadratic cost problem and feedback stabilization of a linear system. *IEEE Transactions on Automatic Control*, 22(5):838–842, 1977. doi: 10.1109/TAC.1977.1101619.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V. Le, and Ed H. Chi. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=WZH7099tgfM.

Gabriel Maher. Llmpc: Large language model predictive control. *Computers*, 14(3), 2025. ISSN 2073-431X. doi: 10.3390/computers14030104. URL https://www.mdpi.com/2073-431X/14/3/104.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. 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. URL https://openreview.net/forum?id=1PL1NIMMrw.

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. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=S37hOerQLB.

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 Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3563–3578, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.212. URL https://aclanthology.org/2024.findings-acl.212/.

Bolei He, Xinran He, Mengke Chen, Xianwei Xue, Ying Zhu, and Zhen-Hua Ling. RISE: Reasoning enhancement via iterative self-exploration in multi-hop question answering. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14925–14948, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.772. URL https://aclanthology.org/2025.findings-acl.772/.

Xiaoyu Tian, Sitong Zhao, Haotian Wang, Shuaiting Chen, Yunjie Ji, Yiping Peng, Han Zhao, and Xiangang Li. Think twice: Enhancing llm reasoning by scaling multi-round test-time thinking, 2025. URL https://arxiv.org/abs/2503.19855.

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

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, et al. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*, 2024.

Mistral AI. Model card for mistral-small-24b-instruct-2501. Hugging Face model card, 2025. URL https://huggingface.co/mistralai/Mistral-Small-24B-Instruct-2501. Accessed: 2026-01-01.

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*, 2025.

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.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of machine learning and systems*, 6: 87–100, 2024.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tcbBPnfwxS.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming

with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3): 90–95, 2007. doi: 10.1109/MCSE.2007.55.

APPENDIX CONTENTS

A Macro-aggregated results 15

B Answer aggregation across intermediate reasoning steps 16

C Answer aggregation across stochastic reasoning paths 20

D Additional paths vs. steps results 24

E Intermediate question statistics 27

F Trajectory changes under replanning 31

G Hyperparameter optimization 33

H Robustness across repeated runs 34

I Planning ablation details 34

J iLLC L1-8 and iLLC L2-8 results 35

K Model details 35

L Intermediate question instructions 36

M Dataset details and examples 37

N Sample iLLC L2-4 dialogues 41

A MACRO-AGGREGATED RESULTS

Table 5 complements the per-subset results in Table 1 by giving each benchmark family equal weight. We first average accuracy within the three MuSR tasks (MM, OP, and TA) and the six iLLC variants (L1-4, L1-6, L1-8, L2-4, L2-6, and L2-8), then average those family-level results with MoreHopQA. Under this aggregation, ReHoPER improves over the corresponding Self-Consistency (SC) baseline by 3.9pp for IO, 6.6pp for CoT, and 4.8pp for PS, with gains calculated before rounding the accuracies. Thus, the overall advantage persists when families with more subsets do not receive more weight.

The gains extend across all three benchmark families and all three answer formats. Averaged across models, ReHoPER improves MuSR by 2.9pp, 3.1pp, and 2.1pp for IO, CoT, and PS, respectively; MoreHopQA by 0.8pp, 2.6pp, and 2.3pp; and iLLC by 8.0pp, 14.2pp, and 10.0pp. The especially large improvements on iLLC are consistent with the per-subset results in Table 1, where the gains are greatest in the more demanding compositional settings. Together, these macro-aggregated results show that ReHoPER's advantage is not driven by the number of subsets in any one benchmark family.

Table 5: Macro-aggregated accuracy (%) on three benchmark families. MuSR averages MM, OP, and TA; iLLC averages L1-4, L1-6, L1-8, L2-4, L2-6, and L2-8; and MoreHopQA is reported separately. The final block is an equal-weight average of these three family-level results. SD denotes Self-Discover. Green, red, and gray indicate improvements, decreases, and ties of ReHoPER (RH) relative to Self-Consistency (SC), respectively.

	MuSR							iLLC						
		IO		CoT		PS			IO		CoT		PS	
Model	**SD**	**SC**	**RH**	**SC**	**RH**	**SC**	**RH**	**SD**	**SC**	**RH**	**SC**	**RH**	**SC**	**RH**
Gemma 3	52.5	54.5	54.6	56.3	56.2	58.1	59.2	24.7	7.8	16.6	30.5	30.5	37.1	32.5
gpt-oss	66.5	65.2	66.0	60.0	64.4	65.1	66.1	62.2	85.9	83.9	78.3	83.4	52.7	83.9
Llama 3.1 8B	44.4	47.1	53.0	50.5	52.3	51.5	53.2	12.2	28.3	26.9	26.5	31.7	35.0	37.1
Llama 3.1 70B	55.0	57.9	61.6	58.0	61.4	58.5	62.1	39.9	40.3	69.9	44.5	69.9	56.6	77.3
Mistral Small	53.1	53.1	59.8	55.5	63.3	58.0	62.8	22.9	38.3	38.3	38.9	39.3	39.1	40.6
Phi-4	57.4	61.3	60.4	59.5	62.5	61.1	63.3	38.7	38.2	52.9	40.8	61.7	40.8	61.3
Qwen3 w/o Thinking	53.3	56.2	62.5	63.1	65.0	63.2	64.4	29.7	53.2	68.1	48.3	68.3	74.7	72.4
Qwen3 w/ Thinking	63.2	65.2	65.4	63.1	65.2	63.2	64.8	1.6	87.3	86.4	48.3	84.9	74.7	85.6
Average	55.7	57.6	60.4	58.2	61.3	59.8	62.0	29.0	47.4	55.4	44.5	58.7	51.3	61.3

	MoreHopQA							Average						
		IO		CoT		PS			IO		CoT		PS	
Model	**SD**	**SC**	**RH**	**SC**	**RH**	**SC**	**RH**	**SD**	**SC**	**RH**	**SC**	**RH**	**SC**	**RH**
Gemma 3	44.0	61.3	64.7	64.7	68.7	66.7	74.0	40.4	41.2	45.3	50.5	51.8	54.0	55.2
gpt-oss	39.3	80.7	78.7	81.3	80.0	79.3	81.3	56.0	77.2	76.2	73.2	75.9	65.7	77.1
Llama 3.1 8B	34.0	59.3	57.3	61.3	60.7	64.0	63.3	30.2	44.9	45.7	46.1	48.2	50.2	51.2
Llama 3.1 70B	62.0	68.0	71.3	68.0	71.3	68.7	74.0	52.3	55.4	67.6	56.8	67.6	61.3	71.1
Mistral Small	54.7	70.0	71.3	70.0	70.0	71.3	72.7	43.5	53.8	56.5	54.8	57.6	56.2	58.7
Phi-4	64.0	77.3	77.3	76.0	80.7	77.3	76.7	53.4	58.9	63.5	58.8	68.3	59.7	67.1
Qwen3 w/o Thinking	34.0	70.0	68.0	69.3	70.7	73.3	72.0	39.0	59.8	66.2	60.2	68.0	70.4	69.6
Qwen3 w/ Thinking	42.0	75.3	80.0	69.3	78.7	73.3	78.0	35.6	76.0	77.3	60.2	76.3	70.4	76.1
Average	46.8	70.3	71.1	70.0	72.6	71.8	74.0	43.8	58.4	62.3	57.6	64.2	61.0	65.8

B ANSWER AGGREGATION ACROSS INTERMEDIATE REASONING STEPS

ReHoPER produces candidate predictions not only at the end of each reasoning path, but also after each intermediate question. We therefore analyze how different ways of aggregating these predictions affect performance over the rollout. Let $\hat{y}_0$ be the initial prediction and $\hat{y}_t^{(i)}$ be the prediction from path i after step t. We compare three aggregation summaries at step k. **Cumulative Vote** (CV) applies majority voting over $\hat{y}_0$ and all predictions from all paths up to step k:

$$\mathrm{CV}_k = \mathrm{MV}\Big(\langle \hat{y}_0 \rangle \parallel \Big(\parallel_{i=1}^{n} \parallel_{t=1}^{\min(k,m_i)} \langle \hat{y}_t^{(i)} \rangle \Big)\Big), \tag{4}$$

where MV denotes majority vote. This is the default ReHoPER aggregation when $k = m_{\max}$.

For step-wise aggregation, we define the carried-forward prediction for path i at step k as

$$\tilde{y}_k^{(i)} = \hat{y}_{\min(k,m_i)}^{(i)}, \tag{5}$$

where we set $\hat{y}_0^{(i)} = \hat{y}_0$ for all paths. Thus, if path i has already terminated before step k, $\tilde{y}_k^{(i)}$ is its last available prediction. **Current-Step Vote** (CSV) applies majority voting over exactly one carried-forward prediction from each path:

$$\mathrm{CSV}_k = \mathrm{MV}\Big(\Big\langle \tilde{y}_k^{(1)}, \tilde{y}_k^{(2)}, \ldots, \tilde{y}_k^{(n)} \Big\rangle\Big). \tag{6}$$

Current-Step Mean (CSM) uses the same carried-forward predictions, but reports their mean correctness rather than applying majority voting:

$$\mathrm{CSM}_k = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\Big[\tilde{y}_k^{(i)} = y\Big]. \tag{7}$$

Thus, CSM is a diagnostic measure of per-path quality rather than an inference-time aggregation rule.

Figures 4–6 show the results for IO, CoT, and PS across datasets and models. Overall, cumulative aggregation is consistently competitive and is usually the strongest final aggregation strategy. Averaged over all plotted dataset–model–prefix configurations, final-step CV reaches 64.6% accuracy, compared with 62.2% for CSV and 55.5% for CSM. CV is also the best or tied for best final aggregation rule in 48 out of 63 configurations, and under PS in 18 out of 21 configurations. The gain over the initial prediction is largest on iLLC L2-4, where performance often continues to improve as more intermediate reasoning states are accumulated.

These results support the default ReHoPER design: useful evidence is distributed across multiple intermediate states, and retaining earlier predictions can make the final answer more robust than relying only on the most recent state. CSV is occasionally competitive, especially on flatter settings such as MuSR MM, but the cumulative vote provides a more reliable choice across prefixes, datasets, and model families. CSM is generally lower than both voting strategies, highlighting the value of ReHoPER's aggregation mechanism: predictions from different paths and intermediate steps provide complementary evidence that, when combined, yields more reliable final decisions.

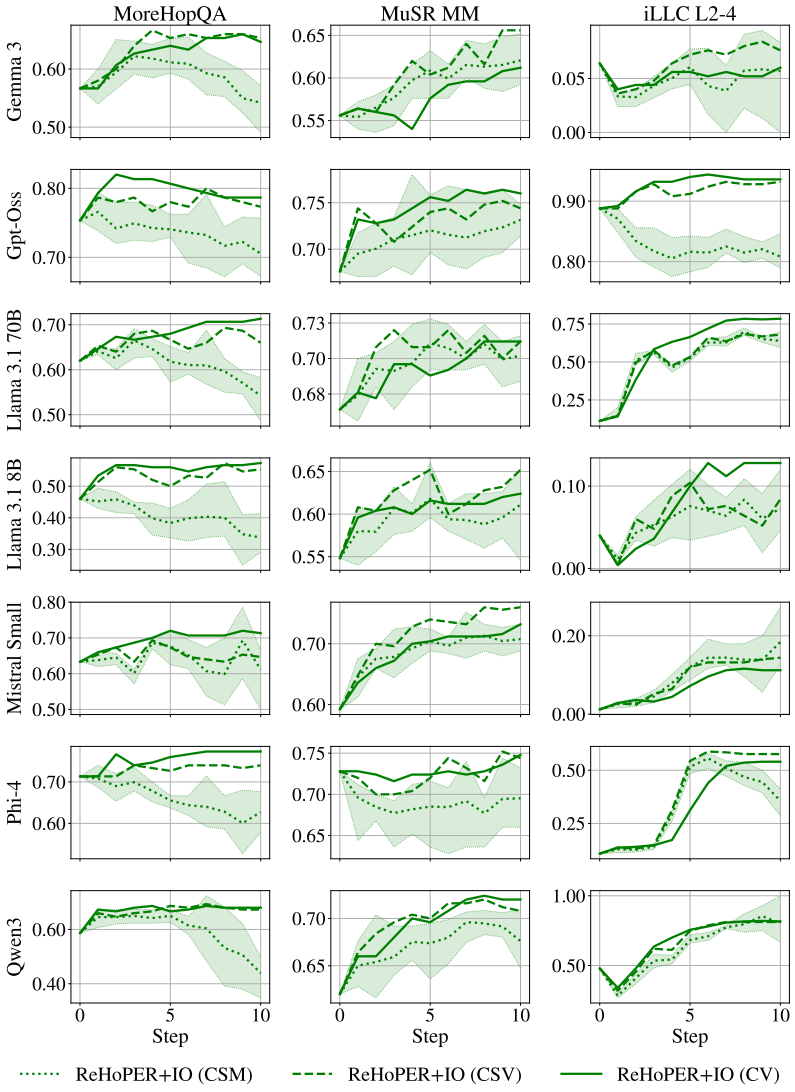


Figure 4: Step-wise aggregation analysis for ReHoPER under the **IO** answer format. Each panel shows accuracy as a function of the reasoning step for one model–dataset pair. CV denotes cumulative voting over all predictions up to the current step; CSV votes over one carried-forward prediction per path at the current step; CSM reports the mean carried-forward per-path correctness. Shaded bands show the standard deviation for the per-path CSM diagnostic.

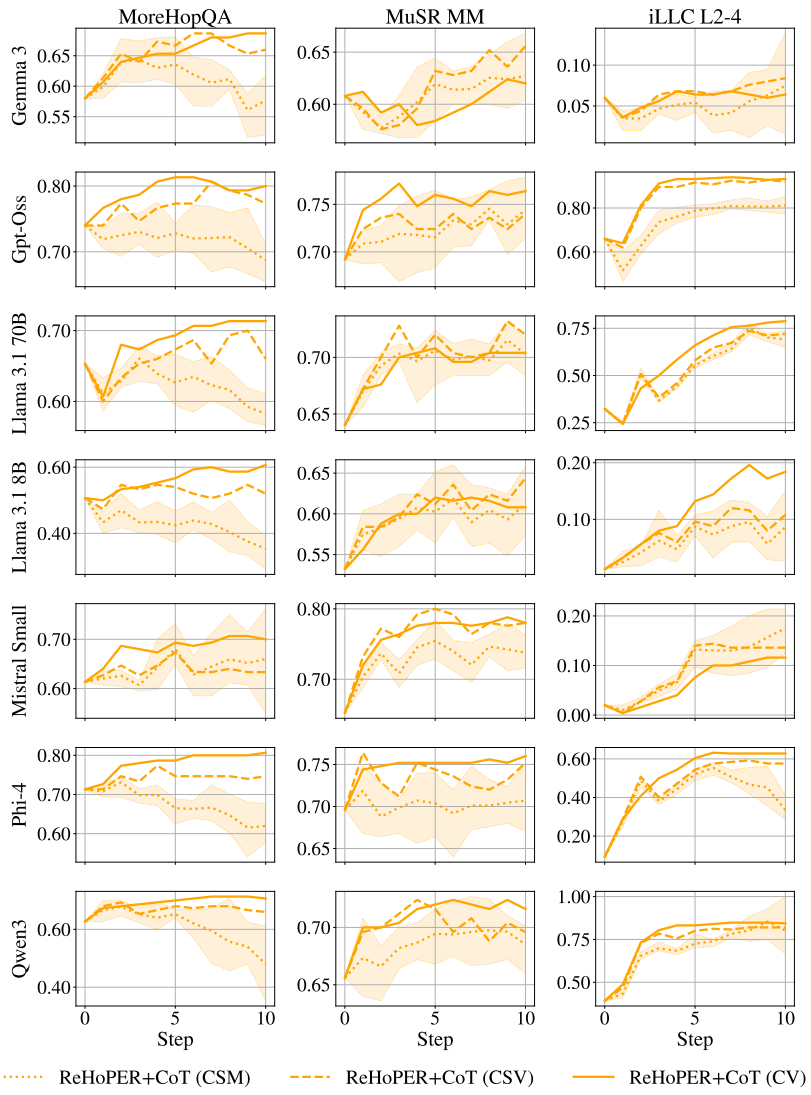


Figure 5: Step-wise aggregation analysis for ReHoPER under the **CoT** answer format. Each panel shows accuracy as a function of the reasoning step for one model–dataset pair. CV denotes cumulative voting over all predictions up to the current step; CSV votes over one carried-forward prediction per path at the current step; CSM reports the mean carried-forward per-path correctness. Shaded bands show the standard deviation for the per-path CSM diagnostic.

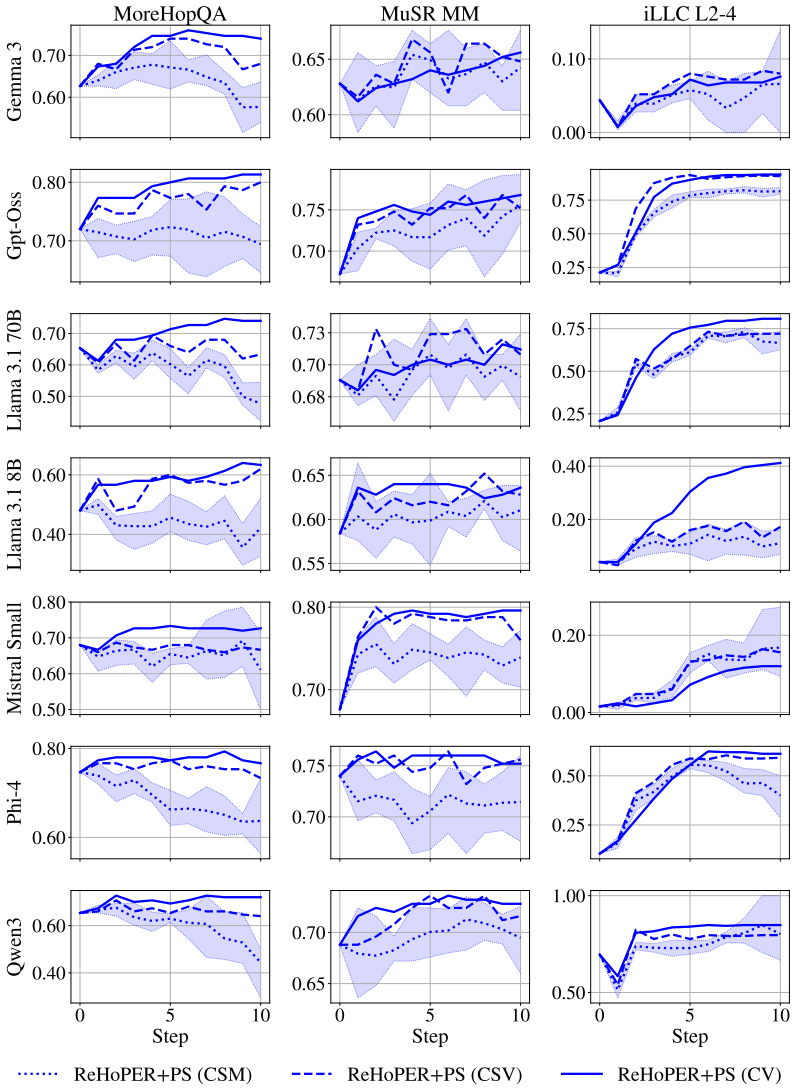


Figure 6: Step-wise aggregation analysis for ReHoPER under the **PS** answer format. Each panel shows accuracy as a function of the reasoning step for one model–dataset pair. CV denotes cumulative voting over all predictions up to the current step; CSV votes over one carried-forward prediction per path at the current step; CSM reports the mean carried-forward per-path correctness. Shaded bands show the standard deviation for the per-path CSM diagnostic.

C ANSWER AGGREGATION ACROSS STOCHASTIC REASONING PATHS

We analyze how ReHoPER's predictions accumulate across stochastic reasoning paths. This complements the step-wise analysis in Appendix B: instead of varying the reasoning step within a fixed set of paths, we vary the number of paths included in the aggregation. Let $\hat{y}_0$ denote the initial prediction, shared across all paths, and let $\hat{y}_t^{(i)}$ be the prediction from path i after step t. For a path index $j \in \{1, \ldots, n\}$, **Cumulative Vote** (CV) aggregates predictions from paths 1 through j:

$$c_j = \mathrm{MV}\Big(\langle \hat{y}_0 \rangle \parallel \Big(\|_{i=1}^{j} \|_{t=1}^{m_i} \langle \hat{y}_t^{(i)} \rangle \Big)\Big). \tag{8}$$

Here, the initial prediction $\hat{y}_0$ is included only once, rather than being duplicated for each path.

To isolate the contribution of individual paths, **Current-Path Vote** (CPV) applies majority voting using only the predictions from path j, together with the shared initial prediction:

$$v_j = \mathrm{MV}\Big(\langle \hat{y}_0 \rangle \parallel \Big(\|_{t=1}^{m_j} \langle \hat{y}_t^{(j)} \rangle \Big)\Big). \tag{9}$$

Current-Path Mean (CPM) uses the same path-local predictions, but reports their mean correctness:

$$q_j = \frac{1}{m_j + 1} \left(\mathbf{1}[\hat{y}_0 = y] + \sum_{t=1}^{m_j} \mathbf{1}[\hat{y}_t^{(j)} = y] \right). \tag{10}$$

The plotted CV and CPV values are obtained by averaging $\mathbf{1}[c_j = y]$ and $\mathbf{1}[v_j = y]$ across examples, respectively, while CPM averages q_j across examples. Finally, **All-Path Vote** (APV) and **All-Path Mean** (APM) summarize the average path-local performance across all paths:

$$\mathrm{APV} = \frac{1}{n} \sum_{j=1}^{n} \mathrm{CPV}_j, \qquad \mathrm{APM} = \frac{1}{n} \sum_{j=1}^{n} \mathrm{CPM}_j. \tag{11}$$

These two quantities are shown as flat reference lines because they average over all paths rather than varying cumulatively with the path index.

Figures 7–9 show the resulting path-wise aggregation trends for IO, CoT, and PS. Overall, cumulative voting across paths is generally more reliable than relying on a single path in isolation. CPV and CPM show that individual paths can already provide useful predictions, while the gap between CV and the path-local summaries highlights the value of combining complementary evidence from multiple stochastic trajectories. The flat APV and APM lines provide a useful reference for the average quality of individual paths; when CV rises above these lines, it indicates that cumulative aggregation benefits from pooling complementary predictions across paths rather than merely matching the average path-local behavior.

These results further support ReHoPER's aggregation design. The method benefits from both within-path reasoning depth and across-path diversity: each path explores a different reasoning trajectory, and cumulative voting pools the useful predictions contributed by different paths while preserving the evidence accumulated in earlier ones. This makes the final prediction less dependent on any single trajectory and better reflects the collective evidence gathered during the full ReHoPER rollout.

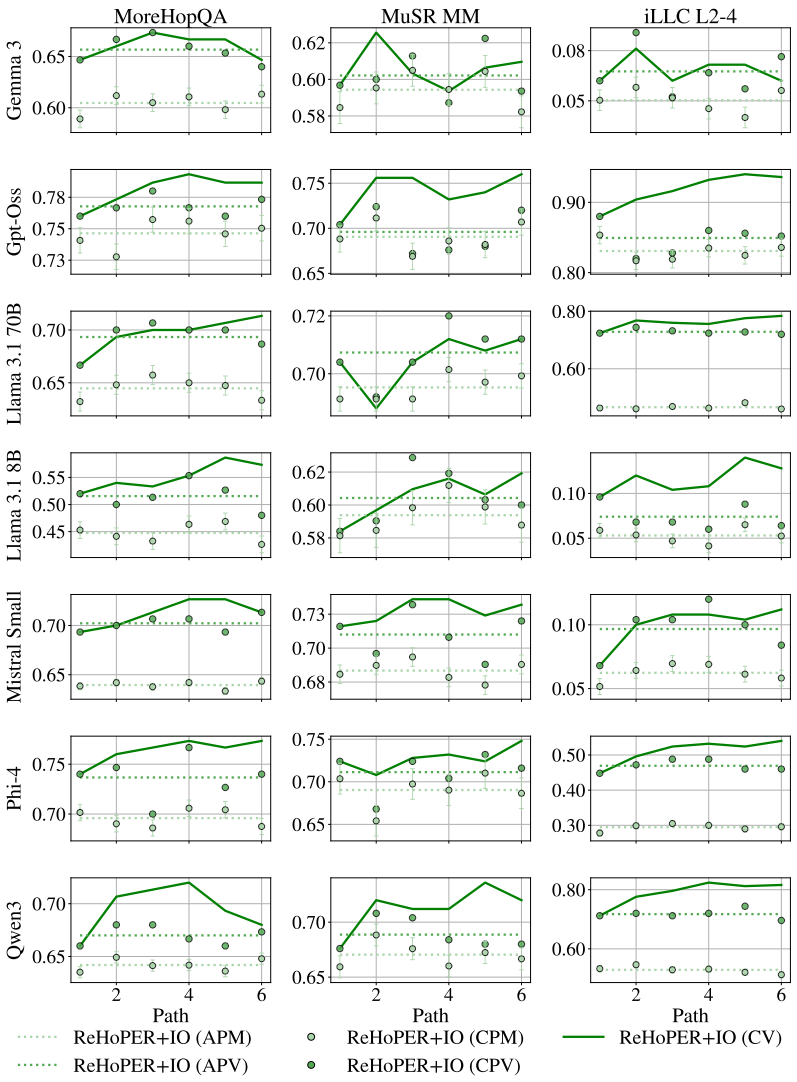


Figure 7: Path-wise aggregation analysis for ReHoPER under the **IO** answer format. CV cumulatively votes over predictions from paths 1 through j, including the shared initial prediction only once. CPV and CPM summarize the current path alone, while APV and APM show the average path-local vote and mean correctness across all paths.

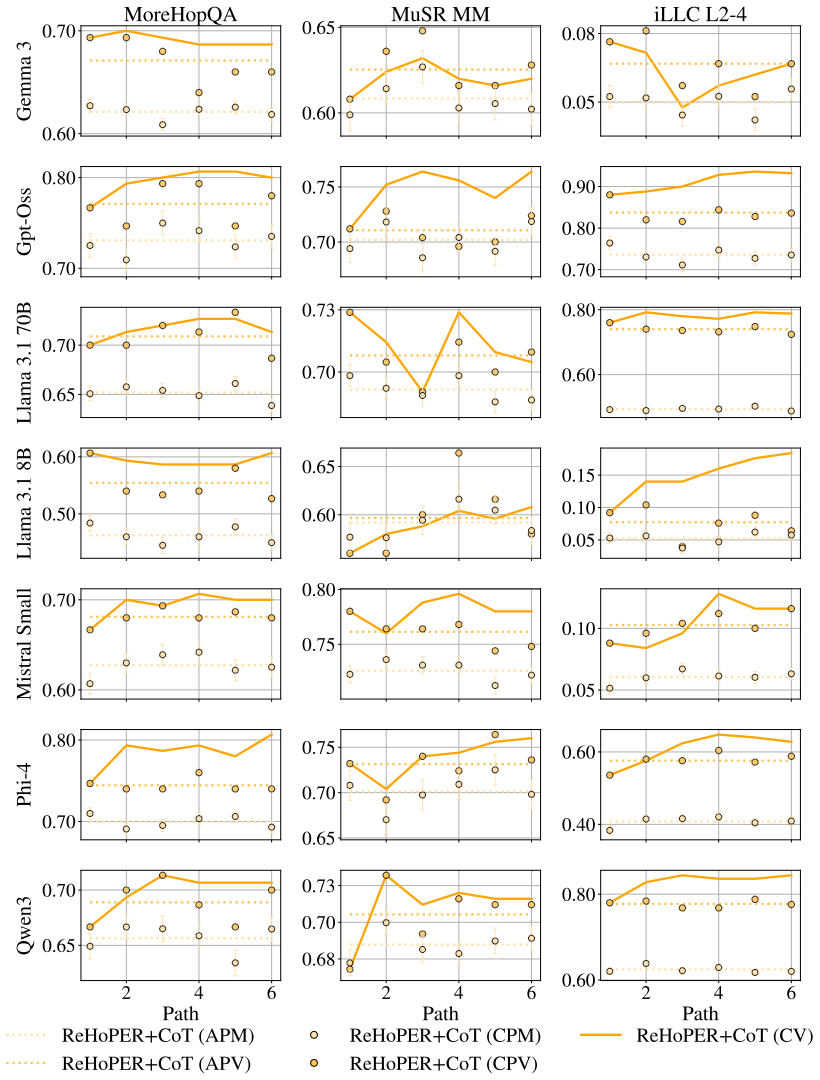


Figure 8: Path-wise aggregation analysis for ReHoPER under the **CoT** answer format. CV cumulatively votes over predictions from paths 1 through j, including the shared initial prediction only once. CPV and CPM summarize the current path alone, while APV and APM show the average path-local vote and mean correctness across all paths.

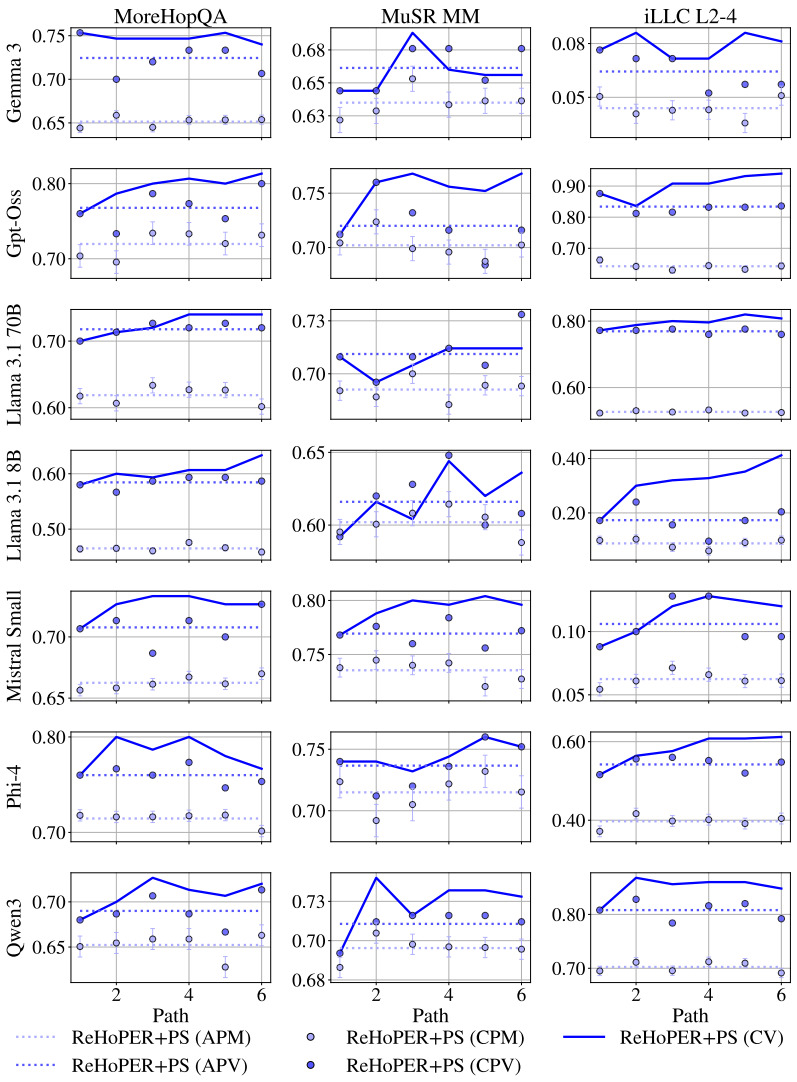


Figure 9: Path-wise aggregation analysis for ReHoPER under the **PS** answer format. CV cumulatively votes over predictions from paths 1 through j, including the shared initial prediction only once. CPV and CPM summarize the current path alone, while APV and APM show the average path-local vote and mean correctness across all paths.

D ADDITIONAL PATHS VS. STEPS RESULTS

Figures 10–12 extend the paths-vs-steps analysis in Figure 3 to additional model backbones and output prefixes (IO, CoT, and PS). At each budget, accuracy is computed by majority vote over a list of candidate predictions. Let $\hat{y}_0$ be the initial prediction, $\hat{y}_k^{(i)}$ the prediction at step k of path i, and m_i the number of steps completed on that path. For the dashed (steps) curves, we fix $p = 6$ paths and increase the maximum step cap s from 0 to 10. The candidate list is $\langle \hat{y}_0 \rangle \parallel \left(\|_{i=1}^{6} \|_{k=1}^{\min(m_i,\, s)} \langle \hat{y}_k^{(i)} \rangle \right)$, so the maximum number of candidates at budget s is $N_{\max}(s) = 1 + 6s$. For the dotted (paths) curves, we fix the per-path step cap to $s = 10$ and instead increase the number of paths p from 0 to 6. The candidate list is $\langle \hat{y}_0 \rangle \parallel \left(\|_{i=1}^{p} \|_{k=1}^{\min(m_i,\, 10)} \langle \hat{y}_k^{(i)} \rangle \right)$, so the maximum number of candidates at budget p is $N_{\max}(p) = 1 + 10p$. These maxima are upper bounds: some paths terminate early, so fewer candidates may be included in the vote. The solid SC curve is computed analogously by aggregating over the first $t = 1, \ldots, N_{\max}$ resampled candidates. For readability, we smooth its accuracy trajectory with a 10-point moving average.

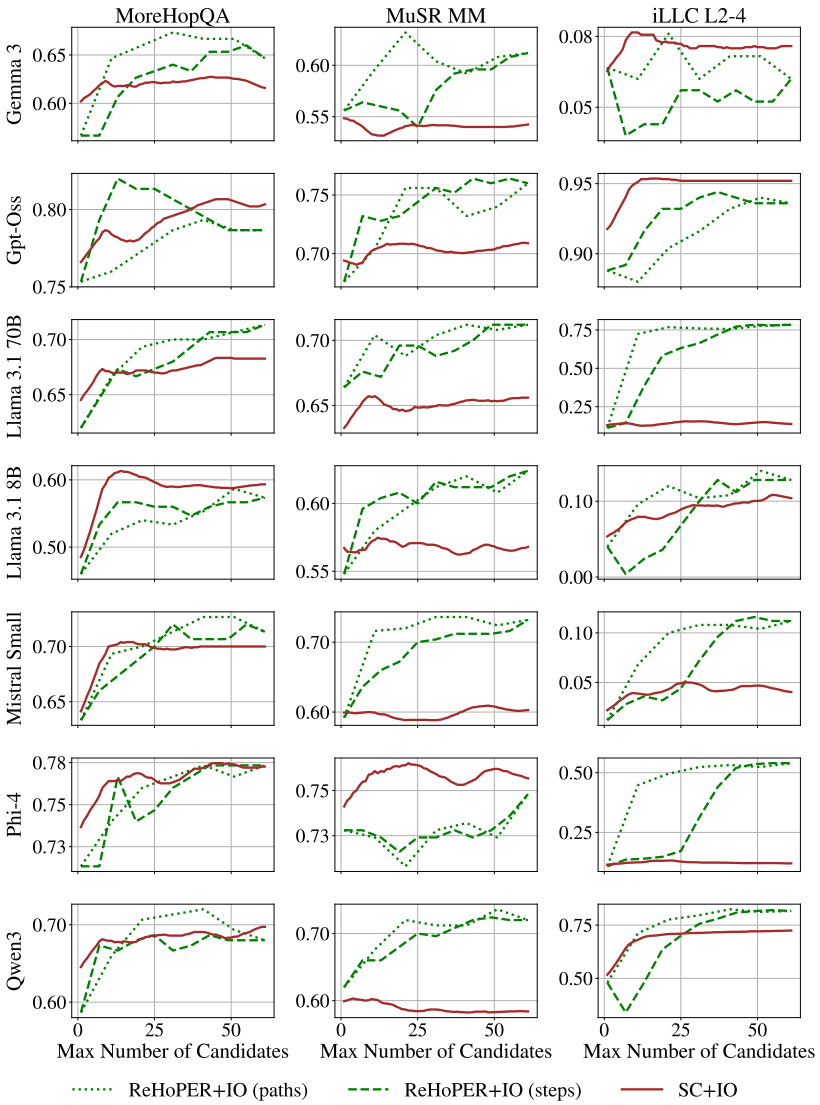


Figure 10: Accuracy gains from allocating iterations to steps vs. paths under the **IO** answer format.

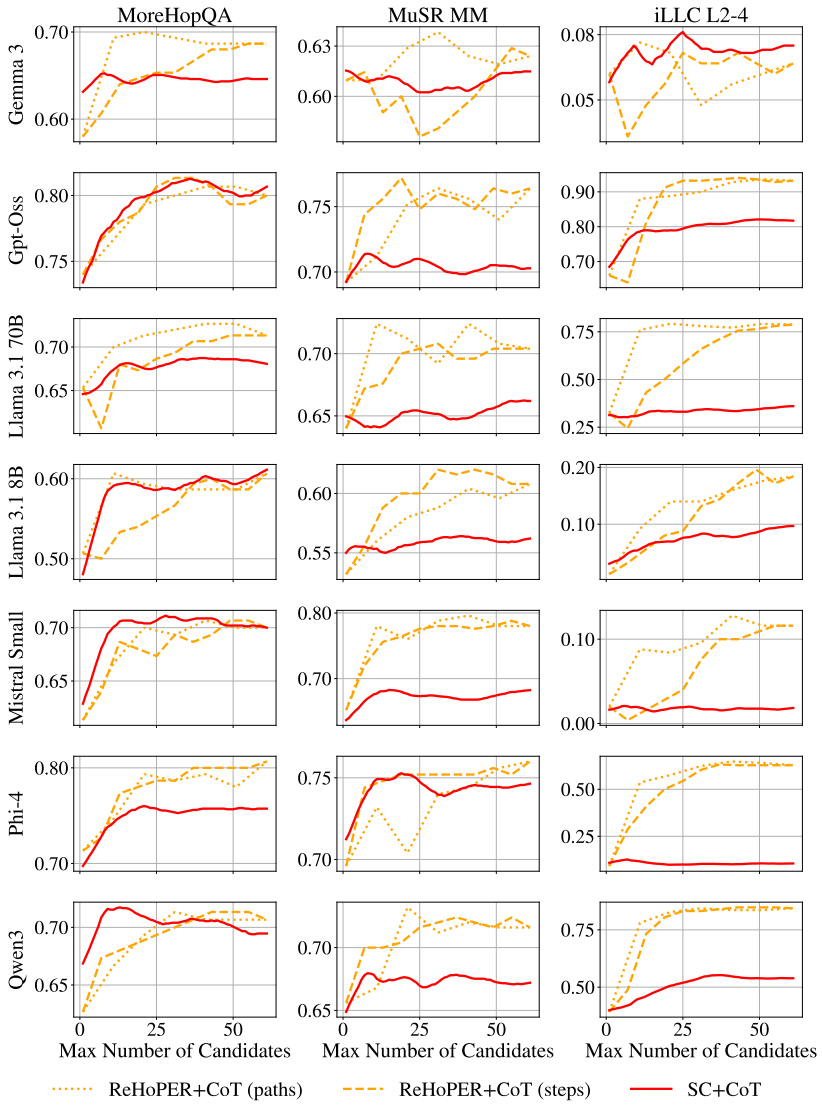


Figure 11: Accuracy gains from allocating iterations to steps vs. paths under the **CoT** answer format.

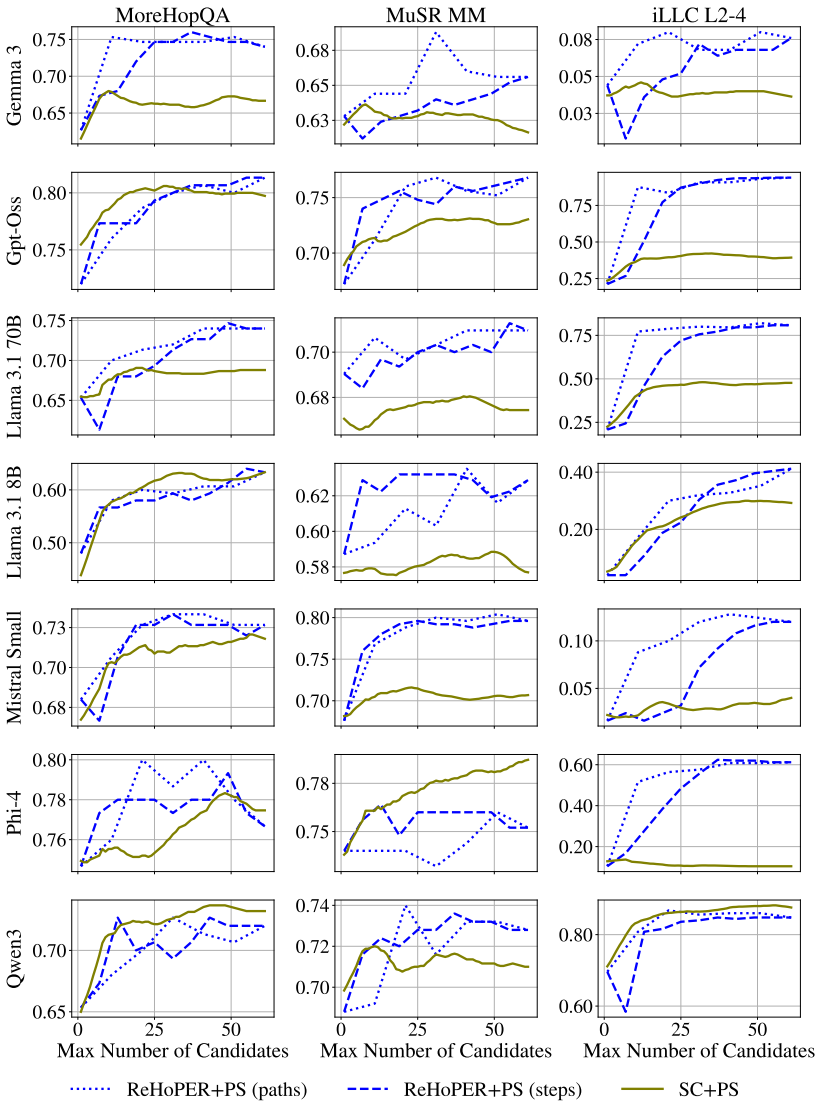


Figure 12: Accuracy gains from allocating iterations to steps vs. paths under the **PS** answer format.

E INTERMEDIATE QUESTION STATISTICS

Tables 6–9 summarize the intermediate questions (IQs) produced by ReHoPER and its two ablations. The column #IQ reports the average total number of valid IQs generated per example across all paths, while #IQ/Path reports the corresponding average per path; its subscript is the average standard deviation of the number of IQs across paths for each example. Similarly, #uniqIQ counts the average number of distinct valid IQs across all paths, and #uniqIQ/Path reports the average number of distinct IQs within each path, again with the corresponding across-path standard deviation. Finally, %uniqIQ measures global diversity across paths, i.e., #uniqIQ divided by #IQ, while %uniqIQ/Path measures within-path diversity, i.e., #uniqIQ/Path divided by #IQ/Path.

A useful sanity check is that %uniqIQ/Path is essentially 100% for ReHoPER across all datasets in Tables 6 and 7. This is expected because ReHoPER explicitly selects the first candidate IQ that has not already been asked in the current path, and terminates the path if all candidates repeat. The ablation without replanning shows the same pattern in Table 8: although it does not replan, it commits to a single ordered list of IQs at the beginning of each path, and repeated questions within that fixed list are rare. In contrast, the ablation without look-ahead generates only one next IQ at a time without a horizon or selection step, so it can repeatedly ask the same or near-identical questions. Importantly, this repetition happens even though the instruction for the ablation without look-ahead explicitly asks the model not to repeat previously asked questions (Figure 15 in Appendix L). This suggests that prompt-level discouragement alone is not sufficient to reliably enforce non-repetition; the explicit selection mechanism in ReHoPER is what keeps within-path uniqueness at 100%.

On the three datasets used for the planning ablations, trajectory lengths and question diversity differ across configurations. The ablation without look-ahead almost always reaches the maximum path length, generating about 60 IQs in total, or 10 IQs per path, because its termination criterion is tied to the step cap rather than the availability of useful new questions. However, this larger number of generated IQs does not necessarily correspond to more distinct questions. On MoreHopQA, the ablation without look-ahead generates 60.00 IQs on average but only 20.51 unique IQs, while ReHoPER generates fewer total IQs (43.07) but more unique IQs (24.25). The same pattern appears in the within-path counts: the ablation without look-ahead asks 10.00 IQs per path but only 5.77 unique IQs per path, whereas ReHoPER asks 7.18 IQs per path and all 7.18 are unique. The ablation without replanning, by contrast, produces much shorter trajectories: averaged across models, it generates only 3.03 IQs per path on MoreHopQA and 3.89 on iLLC L2-4, compared with 7.18 and 7.33 for ReHoPER. ReHoPER therefore produces substantially more valid and unique IQs than the ablation without replanning, while avoiding the repeated-question behavior observed in the ablation without look-ahead.

Across datasets, ReHoPER produces the most saturated trajectories on MuSR. In Table 6, the average #IQ/Path is close to the maximum on MuSR MM, OP, and TA, and global uniqueness remains high, ranging from 90.04% to 96.70% on average across the three MuSR subsets. MoreHopQA shows a different pattern, as the global uniqueness in ReHoPER drops to 55.47% on average. This means that different stochastic paths often converge to similar IQs, suggesting that the task admits a more constrained set of useful decompositions.

The iLLC results in Table 7 show a consistent increase in the amount and diversity of generated IQs as the task becomes more compositional. For example, moving from iLLC L1-4 to iLLC L2-4 increases the average #IQ from 38.62 to 43.99 and the average #uniqIQ from 17.95 to 22.87. A similar pattern holds at longer lengths: iLLC L2-8 yields more total and unique IQs than iLLC L1-8, with #uniqIQ increasing from 23.27 to 27.37. The global uniqueness rate also tends to be higher in the harder L2 variants than in the corresponding L1 variants. This suggests that harder iLLC settings induce a broader set of useful intermediate questions, while ReHoPER still preserves non-redundancy within each individual path.

Finally, the gap between %uniqIQ and %uniqIQ/Path is informative. Since %uniqIQ/Path is always 100% for ReHoPER, any reduction in %uniqIQ reflects overlap across different paths rather than repetition within a path. This overlap is not necessarily undesirable, as on some tasks, multiple paths may independently identify the same useful subquestions. At the same time, model-level variation shows that some models explore more diverse decompositions than others. For instance, on iLLC, Llama 3.1 8B and gpt-oss often maintain relatively high global uniqueness, whereas Phi-4 and Llama 3.1 70B tend to generate more overlapping IQs across paths. Thus, these statistics complement

accuracy results by showing not only how many intermediate questions are generated, but also whether the reasoning paths provide genuinely different decompositions or repeatedly revisit the same subproblems.

Table 6: Statistics of generated intermediate questions (IQs) in **ReHoPER** on **MuSR** and **MoreHopQA**. We report the total number of generated IQs, the average number of IQs per path with standard deviation, the number of unique IQs, the average number of unique IQs per path with standard deviation, and the corresponding uniqueness percentages.

	MuSR MM					
Model	**#IQ**	**#IQ/Path$_{\text{std}}$**	**#uniqIQ**	**#uniqIQ/Path$_{\text{std}}$**	**%uniqIQ**	**%uniqIQ/Path**
Gemma 3	60.00	$10.00_{0.00}$	59.29	$10.00_{0.00}$	98.82%	100.00%
gpt-oss	39.38	$6.56_{3.72}$	39.29	$6.56_{3.72}$	99.77%	100.00%
Llama 3.1 8B	59.95	$9.99_{0.02}$	59.54	$9.99_{0.02}$	99.32%	100.00%
Llama 3.1 70B	60.00	$10.00_{0.00}$	56.72	$10.00_{0.00}$	94.53%	100.00%
Mistral Small	59.96	$9.99_{0.01}$	55.86	$9.99_{0.01}$	93.16%	100.00%
Phi-4	59.97	$10.00_{0.01}$	57.08	$10.00_{0.01}$	95.17%	100.00%
Qwen3	59.56	$9.93_{0.15}$	57.24	$9.93_{0.15}$	96.12%	100.00%
Average	56.97	$9.50_{0.56}$	55.00	$9.50_{0.56}$	96.70%	100.00%
	MuSR OP					
Model	**#IQ**	**#IQ/Path$_{\text{std}}$**	**#uniqIQ**	**#uniqIQ/Path$_{\text{std}}$**	**%uniqIQ**	**%uniqIQ/Path**
Gemma 3	59.98	$10.00_{0.01}$	52.47	$10.00_{0.01}$	87.48%	100.00%
gpt-oss	59.42	$9.90_{0.22}$	57.66	$9.90_{0.22}$	97.04%	100.00%
Llama 3.1 8B	59.27	$9.88_{0.25}$	57.52	$9.88_{0.25}$	97.05%	100.00%
Llama 3.1 70B	59.99	$10.00_{0.00}$	51.86	$10.00_{0.00}$	86.45%	100.00%
Mistral Small	55.77	$9.29_{0.85}$	46.51	$9.29_{0.85}$	83.39%	100.00%
Phi-4	59.84	$9.97_{0.05}$	55.41	$9.97_{0.05}$	92.60%	100.00%
Qwen3	55.66	$9.28_{0.84}$	48.01	$9.28_{0.84}$	86.26%	100.00%
Average	58.56	$9.76_{0.32}$	52.78	$9.76_{0.32}$	90.04%	100.00%
	MuSR TA					
Model	**#IQ**	**#IQ/Path$_{\text{std}}$**	**#uniqIQ**	**#uniqIQ/Path$_{\text{std}}$**	**%uniqIQ**	**%uniqIQ/Path**
Gemma 3	60.00	$10.00_{0.00}$	59.88	$10.00_{0.00}$	99.81%	100.00%
gpt-oss	55.38	$9.23_{1.33}$	54.56	$9.23_{1.33}$	98.52%	100.00%
Llama 3.1 8B	59.67	$9.94_{0.12}$	58.76	$9.94_{0.12}$	98.48%	100.00%
Llama 3.1 70B	59.99	$10.00_{0.00}$	55.94	$10.00_{0.00}$	93.25%	100.00%
Mistral Small	60.00	$10.00_{0.00}$	55.43	$10.00_{0.00}$	92.38%	100.00%
Phi-4	60.00	$10.00_{0.00}$	46.76	$10.00_{0.00}$	77.94%	100.00%
Qwen3	59.98	$10.00_{0.01}$	54.42	$10.00_{0.01}$	90.72%	100.00%
Average	59.29	$9.88_{0.21}$	55.11	$9.88_{0.21}$	93.01%	100.00%
	MoreHopQA					
Model	**#IQ**	**#IQ/Path$_{\text{std}}$**	**#uniqIQ**	**#uniqIQ/Path$_{\text{std}}$**	**%uniqIQ**	**%uniqIQ/Path**
Gemma 3	43.25	$7.21_{1.39}$	25.79	$7.21_{1.39}$	59.64%	100.00%
gpt-oss	54.83	$9.14_{0.91}$	37.68	$9.14_{0.91}$	68.72%	100.00%
Llama 3.1 8B	39.47	$6.58_{2.33}$	29.14	$6.58_{2.33}$	73.83%	100.00%
Llama 3.1 70B	48.99	$8.17_{0.89}$	22.33	$8.17_{0.89}$	45.57%	100.00%
Mistral Small	33.12	$5.52_{0.99}$	13.12	$5.52_{0.99}$	39.61%	100.00%
Phi-4	46.07	$7.68_{1.08}$	24.90	$7.68_{1.08}$	54.04%	100.00%
Qwen3	35.76	$5.96_{1.22}$	16.77	$5.96_{1.22}$	46.89%	100.00%
Average	43.07	$7.18_{1.26}$	24.25	$7.18_{1.26}$	55.47%	100.00%

Table 7: Statistics of generated intermediate questions (IQs) in **ReHoPER** on **iLLC**. We report the total number of generated IQs, the average number of IQs per path with standard deviation, the number of unique IQs, the average number of unique IQs per path with standard deviation, and the corresponding uniqueness percentages.

	iLLC L1-4					
Model	**#IQ**	**#IQ/Path$_{\text{std}}$**	**#uniqIQ**	**#uniqIQ/Path$_{\text{std}}$**	**%uniqIQ**	**%uniqIQ/Path**
Gemma 3	34.98	$5.83_{0.90}$	17.23	$5.83_{0.90}$	49.25%	100.00%
gpt-oss	53.96	$8.99_{1.31}$	35.02	$8.99_{1.31}$	64.89%	100.00%
Llama 3.1 8B	23.58	$3.93_{2.47}$	17.78	$3.93_{2.47}$	75.40%	100.00%
Llama 3.1 70B	39.54	$6.59_{0.58}$	10.14	$6.59_{0.58}$	25.65%	100.00%
Mistral Small	38.87	$6.48_{1.66}$	17.58	$6.48_{1.66}$	45.23%	100.00%
Phi-4	42.53	$7.09_{0.80}$	12.82	$7.09_{0.80}$	30.14%	100.00%
Qwen3	36.85	$6.14_{1.27}$	15.08	$6.14_{1.27}$	40.94%	100.00%
Average	38.62	$6.44_{1.28}$	17.95	$6.44_{1.28}$	47.36%	100.00%
	iLLC L1-6					
Model	**#IQ**	**#IQ/Path$_{\text{std}}$**	**#uniqIQ**	**#uniqIQ/Path$_{\text{std}}$**	**%uniqIQ**	**%uniqIQ/Path**
Gemma 3	41.16	$6.86_{1.24}$	26.40	$6.86_{1.24}$	64.12%	100.00%
gpt-oss	52.29	$8.71_{1.49}$	34.24	$8.71_{1.49}$	65.49%	100.00%
Llama 3.1 8B	29.36	$4.89_{3.05}$	25.14	$4.89_{3.05}$	85.61%	100.00%
Llama 3.1 70B	46.94	$7.82_{0.60}$	11.29	$7.82_{0.60}$	24.06%	100.00%
Mistral Small	35.98	$6.00_{1.74}$	17.06	$6.00_{1.74}$	47.43%	100.00%
Phi-4	54.28	$9.05_{0.88}$	18.84	$9.05_{0.88}$	34.70%	100.00%
Qwen3	38.41	$6.40_{2.04}$	23.74	$6.40_{2.04}$	61.81%	100.00%
Average	42.63	$7.10_{1.58}$	22.39	$7.10_{1.58}$	54.75%	100.00%
	iLLC L1-8					
Model	**#IQ**	**#IQ/Path$_{\text{std}}$**	**#uniqIQ**	**#uniqIQ/Path$_{\text{std}}$**	**%uniqIQ**	**%uniqIQ/Path**
Gemma 3	41.31	$6.88_{1.60}$	27.96	$6.88_{1.60}$	67.70%	100.00%
gpt-oss	51.82	$8.64_{1.57}$	33.97	$8.64_{1.57}$	65.55%	100.00%
Llama 3.1 8B	33.90	$5.65_{3.24}$	29.18	$5.65_{3.24}$	86.06%	100.00%
Llama 3.1 70B	36.72	$6.12_{1.90}$	13.51	$6.12_{1.90}$	36.79%	100.00%
Mistral Small	30.15	$5.02_{2.14}$	17.86	$5.02_{2.14}$	59.23%	100.00%
Phi-4	58.45	$9.74_{0.35}$	16.44	$9.74_{0.35}$	28.13%	100.00%
Qwen3	37.19	$6.20_{2.16}$	23.97	$6.20_{2.16}$	64.46%	100.00%
Average	41.36	$6.89_{1.85}$	23.27	$6.89_{1.85}$	58.27%	100.00%
	iLLC L2-4					
Model	**#IQ**	**#IQ/Path$_{\text{std}}$**	**#uniqIQ**	**#uniqIQ/Path$_{\text{std}}$**	**%uniqIQ**	**%uniqIQ/Path**
Gemma 3	43.20	$7.20_{1.66}$	21.57	$7.20_{1.66}$	49.93%	100.00%
gpt-oss	56.82	$9.47_{0.85}$	40.82	$9.47_{0.85}$	71.84%	100.00%
Llama 3.1 8B	41.58	$6.93_{2.68}$	35.16	$6.93_{2.68}$	84.57%	100.00%
Llama 3.1 70B	49.00	$8.17_{0.80}$	16.16	$8.17_{0.80}$	32.99%	100.00%
Mistral Small	36.17	$6.03_{1.58}$	17.66	$6.03_{1.58}$	48.81%	100.00%
Phi-4	42.29	$7.05_{0.72}$	11.84	$7.05_{0.72}$	28.00%	100.00%
Qwen3	38.90	$6.48_{1.04}$	16.85	$6.48_{1.04}$	43.32%	100.00%
Average	43.99	$7.33_{1.33}$	22.87	$7.33_{1.33}$	51.35%	100.00%
	iLLC L2-6					
Model	**#IQ**	**#IQ/Path$_{\text{std}}$**	**#uniqIQ**	**#uniqIQ/Path$_{\text{std}}$**	**%uniqIQ**	**%uniqIQ/Path**
Gemma 3	45.94	$7.66_{1.70}$	30.04	$7.66_{1.70}$	65.40%	100.00%
gpt-oss	56.45	$9.41_{0.97}$	40.37	$9.41_{0.97}$	71.52%	100.00%
Llama 3.1 8B	36.38	$6.06_{2.75}$	30.31	$6.06_{2.75}$	83.32%	100.00%
Llama 3.1 70B	53.71	$8.95_{0.76}$	18.47	$8.95_{0.76}$	34.39%	100.00%
Mistral Small	34.03	$5.67_{1.74}$	16.84	$5.67_{1.74}$	49.49%	100.00%
Phi-4	53.99	$9.00_{0.67}$	17.04	$9.00_{0.67}$	31.56%	100.00%
Qwen3	39.52	$6.59_{2.19}$	21.28	$6.59_{2.19}$	53.86%	100.00%
Average	45.72	$7.62_{1.54}$	24.91	$7.62_{1.54}$	55.65%	100.00%
	iLLC L2-8					
Model	**#IQ**	**#IQ/Path$_{\text{std}}$**	**#uniqIQ**	**#uniqIQ/Path$_{\text{std}}$**	**%uniqIQ**	**%uniqIQ/Path**
Gemma 3	50.66	$8.44_{1.60}$	39.44	$8.44_{1.60}$	77.85%	100.00%
gpt-oss	55.85	$9.31_{1.04}$	40.71	$9.31_{1.04}$	72.89%	100.00%
Llama 3.1 8B	38.24	$6.37_{2.98}$	32.79	$6.37_{2.98}$	85.74%	100.00%
Llama 3.1 70B	52.11	$8.68_{1.12}$	23.75	$8.68_{1.12}$	45.58%	100.00%
Mistral Small	28.25	$4.71_{2.08}$	18.62	$4.71_{2.08}$	65.91%	100.00%
Phi-4	59.25	$9.87_{0.22}$	16.70	$9.87_{0.22}$	28.18%	100.00%
Qwen3	31.70	$5.28_{2.22}$	19.56	$5.28_{2.22}$	61.70%	100.00%
Average	45.15	$7.52_{1.61}$	27.37	$7.52_{1.61}$	62.55%	100.00%

Table 8: Statistics of generated intermediate questions (IQs) in ReHoPER **without replanning** on MuSR MM, MoreHopQA, and iLLC L2-4. We report the total number of generated IQs, the average number of IQs per path with standard deviation, the number of unique IQs, the average number of unique IQs per path with standard deviation, and the corresponding uniqueness percentages.

			MuSR MM			
Model	**#IQ**	**#IQ/Path$_{\text{std}}$**	**#uniqIQ**	**#uniqIQ/Path$_{\text{std}}$**	**%uniqIQ**	**%uniqIQ/Path**
Gemma 3	48.73	$8.12_{1.07}$	48.53	$8.12_{1.07}$	99.58%	100.00%
gpt-oss	59.70	$9.95_{0.10}$	59.62	$9.95_{0.10}$	99.87%	99.99%
Llama 3.1 8B	46.69	$7.78_{2.42}$	46.64	$7.78_{2.42}$	99.90%	99.97%
Llama 3.1 70B	50.04	$8.34_{1.20}$	49.74	$8.34_{1.20}$	99.41%	100.00%
Mistral Small	58.78	$9.80_{0.38}$	58.12	$9.79_{0.39}$	98.87%	99.94%
Phi-4	58.67	$9.78_{0.40}$	58.04	$9.78_{0.40}$	98.92%	100.00%
Qwen3	59.44	$9.91_{0.19}$	58.16	$9.91_{0.19}$	97.86%	99.99%
Average	54.58	$9.10_{0.82}$	54.12	$9.10_{0.82}$	99.20%	99.98%
			MoreHopQA			
Model	**#IQ**	**#IQ/Path$_{\text{std}}$**	**#uniqIQ**	**#uniqIQ/Path$_{\text{std}}$**	**%uniqIQ**	**%uniqIQ/Path**
Gemma 3	19.27	$3.21_{0.56}$	10.95	$3.21_{0.56}$	56.83%	100.00%
gpt-oss	24.09	$4.01_{1.16}$	19.32	$4.01_{1.16}$	80.21%	100.00%
Llama 3.1 8B	16.32	$2.72_{1.49}$	14.09	$2.71_{1.49}$	86.32%	99.80%
Llama 3.1 70B	18.97	$3.16_{0.78}$	13.02	$3.15_{0.77}$	68.65%	99.72%
Mistral Small	15.95	$2.66_{0.69}$	11.03	$2.66_{0.69}$	69.19%	99.96%
Phi-4	11.68	$1.95_{1.72}$	10.11	$1.95_{1.72}$	86.59%	100.00%
Qwen3	20.83	$3.47_{0.32}$	8.36	$3.47_{0.31}$	40.13%	99.90%
Average	18.16	$3.03_{0.96}$	12.41	$3.02_{0.96}$	69.70%	99.91%
			iLLC L2-4			
Model	**#IQ**	**#IQ/Path$_{\text{std}}$**	**#uniqIQ**	**#uniqIQ/Path$_{\text{std}}$**	**%uniqIQ**	**%uniqIQ/Path**
Gemma 3	22.95	$3.82_{0.96}$	17.18	$3.82_{0.96}$	74.88%	99.97%
gpt-oss	28.90	$4.82_{2.14}$	25.44	$4.82_{2.14}$	88.03%	100.00%
Llama 3.1 8B	7.78	$1.30_{0.64}$	6.48	$1.30_{0.64}$	83.20%	100.00%
Llama 3.1 70B	26.34	$4.39_{1.12}$	23.05	$4.39_{1.12}$	87.53%	100.00%
Mistral Small	19.37	$3.23_{0.63}$	14.13	$3.23_{0.63}$	72.97%	100.00%
Phi-4	27.20	$4.53_{1.52}$	19.00	$4.53_{1.52}$	69.86%	100.00%
Qwen3	30.76	$5.13_{1.09}$	21.58	$5.13_{1.09}$	70.14%	100.00%
Average	23.33	$3.89_{1.16}$	18.12	$3.89_{1.16}$	78.09%	100.00%

Table 9: Statistics of generated intermediate questions (IQs) in ReHoPER **without look-ahead** on MuSR MM, MoreHopQA, and iLLC L2-4. We report the total number of generated IQs, the average number of IQs per path with standard deviation, the number of unique IQs, the average number of unique IQs per path with standard deviation, and the corresponding uniqueness percentages.

			MuSR MM			
Model	**#IQ**	**#IQ/Path$_{\text{std}}$**	**#uniqIQ**	**#uniqIQ/Path$_{\text{std}}$**	**%uniqIQ**	**%uniqIQ/Path**
Gemma 3	60.00	$10.00_{0.00}$	58.76	$9.95_{0.09}$	97.93%	99.53%
gpt-oss	59.97	$10.00_{0.01}$	59.29	$9.88_{0.21}$	98.86%	98.88%
Llama 3.1 8B	60.00	$10.00_{0.00}$	57.86	$9.69_{0.47}$	96.43%	96.87%
Llama 3.1 70B	60.00	$10.00_{0.00}$	59.11	$9.95_{0.10}$	98.51%	99.49%
Mistral Small	60.00	$10.00_{0.00}$	57.53	$9.62_{0.52}$	95.88%	96.15%
Phi-4	60.00	$10.00_{0.00}$	48.68	$8.12_{1.28}$	81.13%	81.20%
Qwen3	60.00	$10.00_{0.00}$	56.44	$9.59_{0.63}$	94.07%	95.86%
Average	60.00	$10.00_{0.00}$	56.81	$9.54_{0.47}$	94.69%	95.43%
			MoreHopQA			
Model	**#IQ**	**#IQ/Path$_{\text{std}}$**	**#uniqIQ**	**#uniqIQ/Path$_{\text{std}}$**	**%uniqIQ**	**%uniqIQ/Path**
Gemma 3	60.00	$10.00_{0.00}$	9.95	$5.01_{0.68}$	16.58%	50.08%
gpt-oss	60.00	$10.00_{0.00}$	24.18	$5.65_{1.34}$	40.30%	56.50%
Llama 3.1 70B	59.97	$9.99_{0.00}$	32.85	$8.34_{1.00}$	54.77%	83.44%
Llama 3.1 8B	60.00	$10.00_{0.00}$	31.94	$6.76_{1.32}$	53.23%	67.61%
Mistral Small	60.00	$10.00_{0.00}$	21.03	$7.01_{0.98}$	35.06%	70.07%
Phi-4	60.00	$10.00_{0.00}$	17.67	$5.23_{1.09}$	29.46%	52.33%
Qwen3	60.00	$10.00_{0.00}$	5.95	$2.39_{0.51}$	9.92%	23.92%
Average	60.00	$10.00_{0.00}$	20.51	$5.77_{0.99}$	34.19%	57.71%
			iLLC L2-4			
Model	**#IQ**	**#IQ/Path$_{\text{std}}$**	**#uniqIQ**	**#uniqIQ/Path$_{\text{std}}$**	**%uniqIQ**	**%uniqIQ/Path**
Gemma 3	60.00	$10.00_{0.00}$	12.93	$4.73_{0.73}$	21.55%	47.27%
gpt-oss	59.94	$9.99_{0.02}$	47.28	$8.81_{1.10}$	78.89%	88.15%
Llama 3.1 70B	60.00	$10.00_{0.00}$	31.00	$9.41_{0.64}$	51.67%	94.11%
Llama 3.1 8B	60.00	$10.00_{0.00}$	27.67	$6.41_{1.30}$	46.12%	64.12%
Mistral Small	60.00	$10.00_{0.00}$	37.13	$7.88_{1.11}$	61.88%	78.84%
Phi-4	60.00	$10.00_{0.00}$	31.83	$7.16_{1.30}$	53.05%	71.63%
Qwen3	60.00	$10.00_{0.00}$	3.67	$2.03_{0.24}$	6.12%	20.34%
Average	59.99	$10.00_{0.00}$	27.36	$6.63_{0.92}$	45.61%	66.35%

F TRAJECTORY CHANGES UNDER REPLANNING

To further examine whether ReHoPER simply follows its initial plan or actively adapts its reasoning trajectory, we analyze how often replanning changes the next intermediate question. At each step i, the current plan contains a predicted next question and a "second-next" question. After answering the current question, ReHoPER replans from the updated history. We compare the second-next question from the previous plan with the actual next question selected after replanning. If they differ, we count the transition as replanned.

Tables 10 and 11 report the resulting trajectory statistics. For a step pair $i \rightarrow i+1$, T denotes the average number of instances that are still active and have not terminated by that comparison point, R denotes the average number of those active instances whose next question changes after replanning, and N denotes the total number of instances in the dataset. Thus, $\%T/N$ measures how much of the dataset remains active at that step, $\%R/T$ measures how often replanning changes the next question among active instances, and $\%R/N$ measures the same change rate relative to the full dataset. The ANY row reports whether an instance has at least one such replanning-induced change across all step pairs. All entries are reported as mean_{std} across paths.

Overall, replanning frequently changes the reasoning trajectory rather than merely continuing a fixed plan. This pattern is especially clear on MuSR MM, where nearly all instances remain active across steps and $\%R/T$ is consistently high across models. MoreHopQA shows a similar trend among active instances, although $\%T/N$ decreases at later steps as more paths terminate. The iLLC results require more nuance. Because iLLC admits a natural word-by-word procedure, once the model starts processing words sequentially, the second-next question from the previous plan often matches the next question regenerated after replanning. This is most visible around steps 2–5, where many trajectories enter this stable extraction template, yielding lower $\%R/T$ despite continued replanning. However, this local reduction does not mean that replanning is absent: other steps often still show high replanning rates, and the ANY rows show that most iLLC instances experience at least one replanning-induced change. Thus, lower step-level replanning on some iLLC transitions often reflects preservation of a useful sequential decomposition rather than inactive replanning. Together, these results support the interpretation that ReHoPER's receding-horizon mechanism repeatedly updates the intermediate-question trajectory as new information is added, while still preserving stable subplans when they remain appropriate.

Table 10: Trajectory statistics under the second-next analysis. T is the number of valid step transitions, R is the number of replanned transitions, and N is the total number of instances.

	Gemma 3														
	MuSR MM (N: 250)					MoreHopQA (N: 150)					iLLC L2-4 (N: 250)				
Step	$\mathbf{T}_{\text{std}}$	$\mathbf{\%T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\mathbf{\%R/T}_{\text{std}}$	$\mathbf{\%R/N}_{\text{std}}$	$\mathbf{T}_{\text{std}}$	$\mathbf{\%T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\mathbf{\%R/T}_{\text{std}}$	$\mathbf{\%R/N}_{\text{std}}$	$\mathbf{T}_{\text{std}}$	$\mathbf{\%T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\mathbf{\%R/T}_{\text{std}}$	$\mathbf{\%R/N}_{\text{std}}$
1→2	$250.00_{0.00}$	$100.0_{0.0}$	$249.83_{0.37}$	$99.9_{0.1}$	$99.9_{0.1}$	$138.17_{2.27}$	$92.1_{1.5}$	$122.50_{3.69}$	$88.7_{3.0}$	$81.7_{2.5}$	$250.00_{0.00}$	$100.0_{0.0}$	$209.33_{21.02}$	$83.7_{8.4}$	$83.7_{8.4}$
2→3	$250.00_{0.00}$	$100.0_{0.0}$	$250.00_{0.00}$	$100.0_{0.0}$	$100.0_{0.0}$	$124.33_{2.49}$	$82.9_{1.7}$	$105.50_{2.14}$	$84.9_{2.6}$	$70.3_{1.4}$	$250.00_{0.00}$	$100.0_{0.0}$	$131.00_{34.24}$	$52.4_{13.7}$	$52.4_{13.7}$
3→4	$250.00_{0.00}$	$100.0_{0.0}$	$249.83_{0.37}$	$99.9_{0.1}$	$99.9_{0.1}$	$118.17_{3.13}$	$78.8_{2.1}$	$101.33_{3.68}$	$85.8_{2.0}$	$67.6_{2.5}$	$248.17_{1.21}$	$99.3_{0.5}$	$73.83_{16.87}$	$29.7_{6.8}$	$29.5_{6.7}$
4→5	$250.00_{0.00}$	$100.0_{0.0}$	$250.00_{0.00}$	$100.0_{0.0}$	$100.0_{0.0}$	$106.00_{5.32}$	$70.7_{3.5}$	$90.50_{3.91}$	$85.4_{2.3}$	$60.3_{2.6}$	$241.17_{7.60}$	$96.5_{3.0}$	$74.83_{18.96}$	$31.1_{8.0}$	$29.9_{7.6}$
5→6	$250.00_{0.00}$	$100.0_{0.0}$	$249.83_{0.37}$	$99.9_{0.1}$	$99.9_{0.1}$	$98.83_{3.18}$	$65.9_{2.1}$	$85.00_{3.65}$	$86.0_{3.5}$	$56.7_{2.4}$	$155.83_{39.33}$	$62.3_{15.7}$	$65.00_{15.93}$	$42.7_{11.6}$	$26.0_{6.4}$
6→7	$250.00_{0.00}$	$100.0_{0.0}$	$250.00_{0.00}$	$100.0_{0.0}$	$100.0_{0.0}$	$81.17_{3.80}$	$54.1_{2.5}$	$70.33_{4.82}$	$86.6_{2.7}$	$46.9_{3.2}$	$117.00_{23.33}$	$46.8_{9.3}$	$58.83_{13.64}$	$50.3_{8.3}$	$23.5_{5.5}$
7→8	$250.00_{0.00}$	$100.0_{0.0}$	$250.00_{0.00}$	$100.0_{0.0}$	$100.0_{0.0}$	$69.33_{3.14}$	$46.2_{2.1}$	$59.33_{5.15}$	$85.5_{4.8}$	$39.6_{3.4}$	$110.50_{24.61}$	$44.2_{9.8}$	$56.00_{16.42}$	$49.9_{7.5}$	$22.4_{6.6}$
8→9	$250.00_{0.00}$	$100.0_{0.0}$	$250.00_{0.00}$	$100.0_{0.0}$	$100.0_{0.0}$	$56.17_{3.48}$	$37.4_{2.3}$	$47.83_{4.06}$	$85.2_{5.3}$	$31.9_{2.7}$	$102.33_{23.49}$	$40.9_{9.4}$	$51.33_{11.93}$	$50.5_{7.1}$	$20.5_{4.8}$
9→10	$250.00_{0.00}$	$100.0_{0.0}$	$250.00_{0.00}$	$100.0_{0.0}$	$100.0_{0.0}$	$48.33_{3.40}$	$32.2_{2.3}$	$42.33_{2.75}$	$87.7_{2.3}$	$28.2_{1.8}$	$48.00_{8.79}$	$19.2_{3.5}$	$26.67_{7.65}$	$54.5_{8.6}$	$10.7_{3.1}$
ANY	$250.00_{0.00}$	$100.0_{0.0}$	$250.00_{0.00}$	$100.0_{0.0}$	$100.0_{0.0}$	$142.50_{1.50}$	$95.0_{1.0}$	$137.83_{2.27}$	$96.7_{1.3}$	$91.9_{1.5}$	$250.00_{0.00}$	$100.0_{0.0}$	$242.00_{7.48}$	$96.8_{3.0}$	$96.8_{3.0}$
	gpt-oss														
	MuSR MM (N: 250)					MoreHopQA (N: 150)					iLLC L2-4 (N: 250)				
Step	$\mathbf{T}_{\text{std}}$	$\mathbf{\%T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\mathbf{\%R/T}_{\text{std}}$	$\mathbf{\%R/N}_{\text{std}}$	$\mathbf{T}_{\text{std}}$	$\mathbf{\%T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\mathbf{\%R/T}_{\text{std}}$	$\mathbf{\%R/N}_{\text{std}}$	$\mathbf{T}_{\text{std}}$	$\mathbf{\%T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\mathbf{\%R/T}_{\text{std}}$	$\mathbf{\%R/N}_{\text{std}}$
1→2	$173.67_{10.40}$	$69.5_{4.2}$	$173.33_{10.45}$	$99.8_{0.3}$	$69.3_{4.2}$	$147.67_{1.11}$	$98.4_{0.7}$	$133.67_{3.77}$	$90.5_{2.3}$	$89.1_{2.5}$	$248.50_{0.96}$	$99.4_{0.4}$	$242.83_{3.58}$	$97.7_{1.4}$	$97.1_{1.4}$
2→3	$166.00_{10.86}$	$66.4_{4.3}$	$165.00_{10.69}$	$99.4_{0.3}$	$66.0_{4.3}$	$148.67_{1.11}$	$99.1_{0.7}$	$132.83_{3.24}$	$89.3_{1.8}$	$88.6_{2.2}$	$249.00_{1.00}$	$99.6_{0.4}$	$243.67_{2.43}$	$97.9_{1.0}$	$97.5_{1.0}$
3→4	$165.50_{9.22}$	$66.2_{3.7}$	$164.33_{8.67}$	$99.3_{0.6}$	$65.7_{3.5}$	$144.00_{2.52}$	$96.0_{1.7}$	$130.50_{2.63}$	$90.6_{1.3}$	$87.0_{1.8}$	$245.67_{1.80}$	$98.3_{0.7}$	$236.00_{2.52}$	$96.1_{0.9}$	$94.4_{1.0}$
4→5	$160.50_{11.12}$	$64.2_{4.4}$	$158.67_{11.64}$	$98.8_{0.6}$	$63.5_{4.7}$	$137.33_{2.56}$	$91.6_{1.7}$	$126.17_{2.03}$	$91.9_{2.3}$	$84.1_{1.4}$	$235.67_{4.92}$	$94.3_{2.0}$	$225.50_{6.75}$	$95.7_{1.0}$	$90.2_{2.7}$
5→6	$157.67_{9.96}$	$63.1_{4.0}$	$156.67_{10.14}$	$99.4_{0.5}$	$62.7_{4.1}$	$133.33_{2.98}$	$88.9_{2.0}$	$121.50_{2.50}$	$91.2_{3.3}$	$81.0_{1.7}$	$230.00_{4.97}$	$92.0_{2.0}$	$219.50_{7.83}$	$95.4_{1.7}$	$87.8_{3.1}$
6→7	$155.17_{9.25}$	$62.1_{3.7}$	$153.00_{9.83}$	$98.6_{0.9}$	$61.2_{3.9}$	$123.50_{4.43}$	$82.3_{3.0}$	$113.17_{6.41}$	$91.6_{2.5}$	$75.4_{4.3}$	$209.50_{5.35}$	$83.8_{2.1}$	$202.00_{4.76}$	$96.4_{0.8}$	$80.8_{1.9}$
7→8	$149.83_{10.35}$	$59.9_{4.1}$	$148.33_{9.41}$	$99.0_{0.7}$	$59.3_{3.8}$	$119.33_{3.68}$	$79.6_{2.5}$	$110.33_{4.42}$	$92.4_{2.0}$	$73.6_{2.9}$	$207.50_{5.12}$	$83.0_{2.0}$	$202.33_{6.32}$	$97.5_{0.9}$	$80.9_{2.5}$
8→9	$147.17_{8.93}$	$58.9_{3.6}$	$144.33_{8.16}$	$98.1_{1.2}$	$57.7_{3.3}$	$109.83_{2.19}$	$73.2_{1.5}$	$101.50_{4.27}$	$92.4_{2.5}$	$67.7_{2.8}$	$177.50_{5.32}$	$71.0_{2.1}$	$173.67_{4.75}$	$97.8_{0.3}$	$69.5_{1.9}$
9→10	$143.17_{7.95}$	$57.3_{3.2}$	$140.33_{7.27}$	$98.0_{1.2}$	$56.1_{2.9}$	$100.67_{4.07}$	$67.1_{2.7}$	$93.67_{3.35}$	$93.1_{2.2}$	$62.4_{2.2}$	$171.33_{6.42}$	$68.5_{2.6}$	$167.67_{7.36}$	$97.8_{1.2}$	$67.1_{2.9}$
ANY	$181.50_{10.59}$	$72.6_{4.2}$	$181.50_{10.59}$	$100.0_{0.0}$	$72.6_{4.2}$	$149.17_{0.69}$	$99.4_{0.5}$	$148.50_{0.96}$	$99.6_{0.5}$	$99.0_{0.6}$	$249.17_{1.07}$	$99.7_{0.4}$	$249.17_{1.07}$	$100.0_{0.0}$	$99.7_{0.4}$

Table 11: Trajectory statistics under the second-next analysis. T is the number of valid step transitions, R is the number of replanned transitions, and N is the total number of instances.

Llama 3.1 8B

	MuSR MM (N: 250)					MoreHopQA (N: 150)					iLLC L2-4 (N: 250)				
Step	$\mathbf{T}_{\text{std}}$	$\%\mathbf{T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\%\mathbf{R/T}_{\text{std}}$	$\%\mathbf{R/N}_{\text{std}}$	$\mathbf{T}_{\text{std}}$	$\%\mathbf{T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\%\mathbf{R/T}_{\text{std}}$	$\%\mathbf{R/N}_{\text{std}}$	$\mathbf{T}_{\text{std}}$	$\%\mathbf{T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\%\mathbf{R/T}_{\text{std}}$	$\%\mathbf{R/N}_{\text{std}}$
1→2	$250.00_{0.00}$	$100.0_{0.0}$	$250.00_{0.00}$	$100.0_{0.0}$	$100.0_{0.0}$	$109.67_{5.12}$	$73.1_{3.4}$	$101.83_{4.52}$	$92.9_{1.4}$	$67.9_{3.0}$	$221.83_{8.63}$	$88.7_{3.5}$	$219.17_{9.10}$	$98.8_{1.0}$	$87.7_{3.6}$
2→3	$250.00_{0.00}$	$100.0_{0.0}$	$249.00_{1.00}$	$99.6_{0.4}$	$99.6_{0.4}$	$105.67_{4.15}$	$70.4_{2.8}$	$99.00_{6.14}$	$93.6_{2.8}$	$66.0_{4.1}$	$210.33_{9.62}$	$84.1_{3.8}$	$199.00_{9.27}$	$94.6_{2.3}$	$79.6_{3.7}$
3→4	$249.67_{0.47}$	$99.9_{0.2}$	$248.83_{0.90}$	$99.7_{0.3}$	$99.5_{0.4}$	$94.67_{2.49}$	$63.1_{1.7}$	$88.67_{2.21}$	$93.7_{2.3}$	$59.1_{1.5}$	$201.83_{11.91}$	$80.7_{4.8}$	$186.50_{9.39}$	$92.5_{1.0}$	$74.6_{3.8}$
4→5	$250.00_{0.00}$	$100.0_{0.0}$	$249.00_{0.82}$	$99.6_{0.3}$	$99.6_{0.3}$	$79.67_{2.29}$	$53.1_{1.5}$	$74.17_{2.91}$	$93.1_{1.7}$	$49.4_{1.9}$	$176.50_{13.46}$	$70.6_{5.4}$	$154.50_{13.51}$	$87.5_{1.9}$	$61.8_{5.4}$
5→6	$250.00_{0.00}$	$100.0_{0.0}$	$247.33_{1.11}$	$98.9_{0.4}$	$98.9_{0.4}$	$73.83_{3.72}$	$49.2_{2.5}$	$68.00_{4.04}$	$92.1_{2.6}$	$45.3_{2.7}$	$153.17_{15.02}$	$61.3_{6.0}$	$131.83_{11.20}$	$86.2_{1.5}$	$52.7_{4.5}$
6→7	$249.67_{0.47}$	$99.9_{0.2}$	$247.50_{1.61}$	$99.1_{0.5}$	$99.0_{0.6}$	$65.17_{3.39}$	$43.4_{2.3}$	$59.67_{3.14}$	$91.6_{3.6}$	$39.8_{2.1}$	$129.00_{11.21}$	$51.6_{4.5}$	$112.33_{11.47}$	$87.1_{4.4}$	$44.9_{4.6}$
7→8	$249.50_{0.76}$	$99.8_{0.3}$	$245.67_{2.69}$	$98.5_{0.9}$	$98.3_{1.1}$	$55.67_{4.07}$	$37.1_{2.7}$	$52.17_{2.41}$	$93.9_{3.5}$	$34.8_{1.6}$	$107.83_{9.12}$	$43.1_{3.6}$	$90.83_{10.95}$	$84.1_{5.1}$	$36.3_{4.4}$
8→9	$249.17_{1.07}$	$99.7_{0.4}$	$245.33_{2.62}$	$98.5_{1.2}$	$98.1_{1.0}$	$46.33_{4.11}$	$30.9_{2.7}$	$43.67_{4.15}$	$94.2_{2.8}$	$29.1_{2.8}$	$91.33_{7.61}$	$36.5_{3.0}$	$79.50_{10.10}$	$86.8_{4.5}$	$31.8_{4.0}$
9→10	$248.50_{0.76}$	$99.4_{0.3}$	$245.33_{1.25}$	$98.7_{0.5}$	$98.1_{0.5}$	$41.17_{6.31}$	$27.4_{4.2}$	$39.33_{5.47}$	$95.8_{3.0}$	$26.2_{3.6}$	$80.83_{13.01}$	$32.3_{5.2}$	$70.83_{14.78}$	$87.0_{4.6}$	$28.3_{5.9}$
ANY	$250.00_{0.00}$	$100.0_{0.0}$	$250.00_{0.00}$	$100.0_{0.0}$	$100.0_{0.0}$	$135.83_{3.13}$	$90.6_{2.1}$	$134.17_{2.67}$	$98.8_{0.7}$	$89.4_{1.8}$	$228.00_{8.02}$	$91.2_{3.2}$	$228.00_{8.02}$	$100.0_{0.0}$	$91.2_{3.2}$

Llama 3.1 70B

	MuSR MM (N: 250)					MoreHopQA (N: 150)					iLLC L2-4 (N: 250)				
Step	$\mathbf{T}_{\text{std}}$	$\%\mathbf{T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\%\mathbf{R/T}_{\text{std}}$	$\%\mathbf{R/N}_{\text{std}}$	$\mathbf{T}_{\text{std}}$	$\%\mathbf{T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\%\mathbf{R/T}_{\text{std}}$	$\%\mathbf{R/N}_{\text{std}}$	$\mathbf{T}_{\text{std}}$	$\%\mathbf{T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\%\mathbf{R/T}_{\text{std}}$	$\%\mathbf{R/N}_{\text{std}}$
1→2	$250.00_{0.00}$	$100.0_{0.0}$	$249.33_{0.94}$	$99.7_{0.4}$	$99.7_{0.4}$	$150.00_{0.00}$	$100.0_{0.0}$	$124.00_{3.32}$	$82.7_{2.2}$	$82.7_{2.2}$	$250.00_{0.00}$	$100.0_{0.0}$	$243.00_{8.81}$	$97.2_{3.5}$	$97.2_{3.5}$
2→3	$250.00_{0.00}$	$100.0_{0.0}$	$249.83_{0.37}$	$99.9_{0.1}$	$99.9_{0.1}$	$146.33_{1.11}$	$97.6_{0.7}$	$122.33_{3.04}$	$83.6_{1.6}$	$81.6_{2.0}$	$250.00_{0.00}$	$100.0_{0.0}$	$238.00_{6.71}$	$95.2_{2.7}$	$95.2_{2.7}$
3→4	$250.00_{0.00}$	$100.0_{0.0}$	$249.17_{0.69}$	$99.7_{0.3}$	$99.7_{0.3}$	$133.00_{2.83}$	$88.7_{1.9}$	$110.50_{3.95}$	$83.1_{1.6}$	$73.7_{2.6}$	$250.00_{0.00}$	$100.0_{0.0}$	$169.33_{6.16}$	$67.7_{2.5}$	$67.7_{2.5}$
4→5	$250.00_{0.00}$	$100.0_{0.0}$	$249.00_{1.41}$	$99.6_{0.6}$	$99.6_{0.6}$	$120.00_{1.73}$	$80.0_{1.2}$	$101.00_{2.38}$	$84.2_{2.0}$	$67.3_{1.6}$	$238.67_{3.09}$	$95.5_{1.2}$	$171.17_{5.79}$	$71.7_{2.4}$	$68.5_{2.3}$
5→6	$250.00_{0.00}$	$100.0_{0.0}$	$249.17_{0.69}$	$99.7_{0.3}$	$99.7_{0.3}$	$108.67_{2.98}$	$72.4_{2.0}$	$96.17_{2.91}$	$88.5_{1.9}$	$64.1_{1.9}$	$217.50_{5.19}$	$87.0_{2.1}$	$156.67_{10.35}$	$72.0_{4.7}$	$62.7_{4.1}$
6→7	$250.00_{0.00}$	$100.0_{0.0}$	$248.00_{1.53}$	$99.2_{0.6}$	$99.2_{0.6}$	$94.17_{1.67}$	$62.8_{1.1}$	$80.50_{3.99}$	$85.4_{2.9}$	$53.7_{2.7}$	$185.00_{4.65}$	$74.0_{1.9}$	$140.17_{4.37}$	$75.8_{1.3}$	$56.1_{1.7}$
7→8	$250.00_{0.00}$	$100.0_{0.0}$	$247.67_{1.49}$	$99.1_{0.6}$	$99.1_{0.6}$	$82.33_{1.97}$	$54.9_{1.3}$	$66.67_{2.62}$	$81.0_{3.0}$	$44.4_{1.7}$	$143.00_{6.40}$	$57.2_{2.6}$	$114.00_{5.89}$	$79.7_{2.5}$	$45.6_{2.4}$
8→9	$250.00_{0.00}$	$100.0_{0.0}$	$246.50_{1.71}$	$98.6_{0.7}$	$98.6_{0.7}$	$75.33_{3.59}$	$50.2_{2.4}$	$67.67_{3.59}$	$89.9_{4.1}$	$45.1_{2.4}$	$114.17_{5.01}$	$45.7_{2.0}$	$99.00_{5.60}$	$86.7_{2.2}$	$39.6_{2.2}$
9→10	$250.00_{0.00}$	$100.0_{0.0}$	$246.50_{0.96}$	$98.6_{0.4}$	$98.6_{0.4}$	$67.33_{3.04}$	$44.9_{2.0}$	$60.17_{3.58}$	$89.3_{2.4}$	$40.1_{2.4}$	$88.33_{5.79}$	$35.3_{2.3}$	$73.67_{5.37}$	$83.4_{1.4}$	$29.5_{2.1}$
ANY	$250.00_{0.00}$	$100.0_{0.0}$	$250.00_{0.00}$	$100.0_{0.0}$	$100.0_{0.0}$	$150.00_{0.00}$	$100.0_{0.0}$	$148.83_{0.69}$	$99.2_{0.5}$	$99.2_{0.5}$	$250.00_{0.00}$	$100.0_{0.0}$	$250.00_{0.00}$	$100.0_{0.0}$	$100.0_{0.0}$

Mistral Small

	MuSR MM (N: 250)					MoreHopQA (N: 150)					iLLC L2-4 (N: 250)				
Step	$\mathbf{T}_{\text{std}}$	$\%\mathbf{T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\%\mathbf{R/T}_{\text{std}}$	$\%\mathbf{R/N}_{\text{std}}$	$\mathbf{T}_{\text{std}}$	$\%\mathbf{T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\%\mathbf{R/T}_{\text{std}}$	$\%\mathbf{R/N}_{\text{std}}$	$\mathbf{T}_{\text{std}}$	$\%\mathbf{T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\%\mathbf{R/T}_{\text{std}}$	$\%\mathbf{R/N}_{\text{std}}$
1→2	$250.00_{0.00}$	$100.0_{0.0}$	$246.33_{1.37}$	$98.5_{0.5}$	$98.5_{0.5}$	$134.67_{3.14}$	$89.8_{2.1}$	$106.50_{6.40}$	$79.1_{4.4}$	$71.0_{4.3}$	$250.00_{0.00}$	$100.0_{0.0}$	$169.33_{21.30}$	$67.7_{8.5}$	$67.7_{8.5}$
2→3	$249.83_{0.37}$	$99.9_{0.1}$	$244.67_{2.05}$	$97.9_{0.9}$	$97.9_{0.8}$	$114.83_{1.77}$	$76.6_{1.2}$	$78.50_{1.89}$	$68.4_{1.3}$	$52.3_{1.3}$	$213.67_{11.59}$	$85.5_{4.6}$	$103.83_{11.82}$	$48.6_{5.2}$	$41.5_{4.7}$
3→4	$249.00_{0.82}$	$99.6_{0.3}$	$239.17_{2.79}$	$96.0_{0.9}$	$95.7_{1.1}$	$91.83_{2.34}$	$61.2_{1.6}$	$66.83_{4.22}$	$72.8_{3.7}$	$44.6_{2.8}$	$196.67_{7.41}$	$78.7_{3.0}$	$58.67_{17.30}$	$30.0_{9.2}$	$23.5_{6.9}$
4→5	$249.83_{0.37}$	$99.9_{0.1}$	$241.33_{2.36}$	$96.6_{0.9}$	$96.5_{0.9}$	$69.50_{3.82}$	$46.3_{2.5}$	$56.00_{3.37}$	$80.6_{2.4}$	$37.3_{2.2}$	$189.67_{10.66}$	$75.9_{4.3}$	$38.83_{4.74}$	$20.6_{3.2}$	$15.5_{1.9}$
5→6	$249.83_{0.37}$	$99.9_{0.1}$	$235.17_{2.11}$	$94.1_{0.9}$	$94.1_{0.8}$	$56.83_{1.46}$	$37.9_{1.0}$	$45.00_{2.16}$	$79.2_{2.9}$	$30.0_{1.4}$	$105.33_{13.05}$	$42.1_{5.2}$	$36.50_{3.25}$	$35.2_{5.2}$	$14.6_{1.3}$
6→7	$249.67_{0.47}$	$99.9_{0.2}$	$229.33_{7.27}$	$91.9_{2.9}$	$91.7_{2.9}$	$41.67_{1.11}$	$27.8_{0.7}$	$33.50_{2.06}$	$80.4_{3.7}$	$22.3_{1.4}$	$59.67_{10.06}$	$23.9_{4.0}$	$30.83_{5.79}$	$51.8_{6.2}$	$12.3_{2.3}$
7→8	$249.67_{0.47}$	$99.9_{0.2}$	$230.83_{4.45}$	$92.5_{1.7}$	$92.3_{1.8}$	$28.50_{1.89}$	$19.0_{1.3}$	$21.83_{2.11}$	$76.5_{3.5}$	$14.6_{1.4}$	$38.83_{7.73}$	$15.5_{3.1}$	$22.00_{3.70}$	$57.1_{5.7}$	$8.8_{1.5}$
8→9	$249.50_{0.50}$	$99.8_{0.2}$	$231.17_{5.79}$	$92.7_{2.4}$	$92.5_{2.3}$	$23.83_{1.07}$	$15.9_{0.7}$	$18.17_{1.95}$	$76.1_{6.6}$	$12.1_{1.3}$	$30.33_{8.48}$	$12.1_{3.4}$	$13.67_{4.68}$	$45.4_{10.2}$	$5.5_{1.9}$
9→10	$249.33_{0.47}$	$99.7_{0.2}$	$229.50_{2.87}$	$92.0_{1.3}$	$91.8_{1.1}$	$14.67_{1.49}$	$9.8_{1.0}$	$10.33_{1.37}$	$70.7_{8.1}$	$6.9_{0.9}$	$22.17_{8.13}$	$8.9_{3.3}$	$12.50_{5.32}$	$55.8_{5.2}$	$5.0_{2.1}$
ANY	$250.00_{0.00}$	$100.0_{0.0}$	$250.00_{0.00}$	$100.0_{0.0}$	$100.0_{0.0}$	$141.00_{1.15}$	$94.0_{0.8}$	$130.17_{3.98}$	$92.3_{2.6}$	$86.8_{2.7}$	$250.00_{0.00}$	$100.0_{0.0}$	$195.83_{10.84}$	$78.3_{4.3}$	$78.3_{4.3}$

Phi-4

	MuSR MM (N: 250)					MoreHopQA (N: 150)					iLLC L2-4 (N: 250)				
Step	$\mathbf{T}_{\text{std}}$	$\%\mathbf{T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\%\mathbf{R/T}_{\text{std}}$	$\%\mathbf{R/N}_{\text{std}}$	$\mathbf{T}_{\text{std}}$	$\%\mathbf{T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\%\mathbf{R/T}_{\text{std}}$	$\%\mathbf{R/N}_{\text{std}}$	$\mathbf{T}_{\text{std}}$	$\%\mathbf{T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\%\mathbf{R/T}_{\text{std}}$	$\%\mathbf{R/N}_{\text{std}}$
1→2	$250.00_{0.00}$	$100.0_{0.0}$	$237.00_{3.32}$	$94.8_{1.3}$	$94.8_{1.3}$	$149.67_{0.47}$	$99.8_{0.3}$	$117.33_{2.56}$	$78.4_{1.7}$	$78.2_{1.7}$	$250.00_{0.00}$	$100.0_{0.0}$	$82.50_{32.41}$	$33.0_{13.0}$	$33.0_{13.0}$
2→3	$250.00_{0.00}$	$100.0_{0.0}$	$234.50_{3.86}$	$93.8_{1.5}$	$93.8_{1.5}$	$138.00_{2.08}$	$92.0_{1.4}$	$113.83_{2.73}$	$82.5_{1.5}$	$75.9_{1.8}$	$250.00_{0.00}$	$100.0_{0.0}$	$19.83_{3.53}$	$7.9_{1.4}$	$7.9_{1.4}$
3→4	$249.83_{0.37}$	$99.9_{0.1}$	$234.83_{1.95}$	$94.0_{0.8}$	$93.9_{0.8}$	$124.50_{3.50}$	$83.0_{2.3}$	$100.00_{3.21}$	$80.3_{1.1}$	$66.7_{2.1}$	$250.00_{0.00}$	$100.0_{0.0}$	$15.17_{3.44}$	$6.1_{1.4}$	$6.1_{1.4}$
4→5	$249.83_{0.37}$	$99.9_{0.1}$	$236.33_{3.04}$	$94.6_{1.2}$	$94.5_{1.2}$	$111.17_{4.56}$	$74.1_{3.0}$	$91.50_{4.07}$	$82.3_{2.8}$	$61.0_{2.7}$	$250.00_{0.00}$	$100.0_{0.0}$	$11.50_{3.10}$	$4.6_{1.2}$	$4.6_{1.2}$
5→6	$249.83_{0.37}$	$99.9_{0.1}$	$236.00_{5.03}$	$94.5_{2.0}$	$94.4_{2.0}$	$100.17_{3.02}$	$66.8_{2.0}$	$84.00_{3.87}$	$83.8_{1.5}$	$56.0_{2.6}$	$249.00_{0.58}$	$99.6_{0.2}$	$200.50_{8.26}$	$80.5_{3.4}$	$80.2_{3.3}$
6→7	$249.83_{0.37}$	$99.9_{0.1}$	$234.00_{4.00}$	$93.7_{1.5}$	$93.6_{1.6}$	$87.00_{0.82}$	$58.0_{0.5}$	$74.67_{2.81}$	$85.8_{3.4}$	$49.8_{1.9}$	$53.67_{6.97}$	$21.5_{2.8}$	$42.00_{7.28}$	$77.9_{5.6}$	$16.8_{2.9}$
7→8	$249.83_{0.37}$	$99.9_{0.1}$	$232.67_{3.90}$	$93.1_{1.6}$	$93.1_{1.6}$	$76.67_{3.14}$	$51.1_{2.1}$	$62.50_{2.57}$	$81.7_{4.7}$	$41.7_{1.7}$	$19.67_{4.23}$	$7.9_{1.7}$	$11.33_{3.25}$	$57.0_{10.5}$	$4.5_{1.3}$
8→9	$249.83_{0.37}$	$99.9_{0.1}$	$230.33_{1.97}$	$92.0_{0.7}$	$92.1_{0.8}$	$68.00_{2.77}$	$45.3_{1.8}$	$57.50_{3.82}$	$84.5_{4.0}$	$38.3_{2.5}$	$11.67_{2.92}$	$4.7_{1.2}$	$7.67_{3.82}$	$62.9_{13.5}$	$3.1_{1.5}$
9→10	$249.83_{0.37}$	$99.9_{0.1}$	$228.83_{4.88}$	$91.6_{2.0}$	$91.5_{2.0}$	$54.17_{5.81}$	$36.1_{3.9}$	$45.67_{6.10}$	$84.2_{5.5}$	$30.4_{4.1}$	$6.17_{1.46}$	$2.5_{0.6}$	$5.50_{1.61}$	$89.0_{12.6}$	$2.2_{0.6}$
ANY	$250.00_{0.00}$	$100.0_{0.0}$	$250.00_{0.00}$	$100.0_{0.0}$	$100.0_{0.0}$	$149.83_{0.37}$	$99.9_{0.2}$	$144.67_{1.70}$	$96.6_{1.1}$	$96.4_{1.1}$	$250.00_{0.00}$	$100.0_{0.0}$	$244.17_{2.19}$	$97.7_{0.9}$	$97.7_{0.9}$

Qwen3

	MuSR MM (N: 250)					MoreHopQA (N: 150)					iLLC L2-4 (N: 250)				
Step	$\mathbf{T}_{\text{std}}$	$\%\mathbf{T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\%\mathbf{R/T}_{\text{std}}$	$\%\mathbf{R/N}_{\text{std}}$	$\mathbf{T}_{\text{std}}$	$\%\mathbf{T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\%\mathbf{R/T}_{\text{std}}$	$\%\mathbf{R/N}_{\text{std}}$	$\mathbf{T}_{\text{std}}$	$\%\mathbf{T/N}_{\text{std}}$	$\mathbf{R}_{\text{std}}$	$\%\mathbf{R/T}_{\text{std}}$	$\%\mathbf{R/N}_{\text{std}}$
1→2	$250.00_{0.00}$	$100.0_{0.0}$	$249.33_{0.75}$	$99.7_{0.3}$	$99.7_{0.3}$	$150.00_{0.00}$	$100.0_{0.0}$	$110.50_{2.69}$	$73.7_{1.8}$	$73.7_{1.8}$	$250.00_{0.00}$	$100.0_{0.0}$	$226.00_{8.21}$	$90.4_{3.3}$	$90.4_{3.3}$
2→3	$250.00_{0.00}$	$100.0_{0.0}$	$240.67_{1.97}$	$96.3_{0.8}$	$96.3_{0.8}$	$130.83_{2.73}$	$87.2_{1.8}$	$91.17_{6.26}$	$69.7_{4.3}$	$60.8_{4.2}$	$250.00_{0.00}$	$100.0_{0.0}$	$86.83_{6.52}$	$34.7_{2.6}$	$34.7_{2.6}$
3→4	$250.00_{0.00}$	$100.0_{0.0}$	$239.33_{3.30}$	$95.7_{1.3}$	$95.7_{1.3}$	$100.50_{1.89}$	$67.0_{1.3}$	$68.00_{4.16}$	$67.7_{3.8}$	$45.3_{2.8}$	$230.17_{11.98}$	$92.1_{4.8}$	$25.67_{5.73}$	$11.2_{2.5}$	$10.3_{2.3}$
4→5	$249.83_{0.37}$	$99.9_{0.1}$	$238.00_{3.32}$	$95.3_{1.4}$	$95.2_{1.3}$	$84.33_{3.50}$	$56.2_{2.3}$	$55.83_{3.34}$	$66.2_{2.7}$	$37.2_{2.2}$	$233.67_{8.79}$	$93.5_{3.5}$	$13.67_{4.23}$	$5.9_{1.8}$	$5.5_{1.7}$
5→6	$248.83_{1.34}$	$99.5_{0.5}$	$235.33_{4.61}$	$94.6_{1.5}$	$94.1_{1.8}$	$66.83_{6.04}$	$44.6_{4.0}$	$43.50_{3.82}$	$65.1_{0.7}$	$29.0_{2.5}$	$181.50_{4.72}$	$72.6_{1.9}$	$74.67_{4.82}$	$41.1_{2.1}$	$29.9_{1.9}$
6→7	$247.83_{0.69}$	$99.1_{0.3}$	$235.00_{3.11}$	$94.8_{1.3}$	$94.0_{1.2}$	$47.83_{3.02}$	$31.9_{2.0}$	$33.50_{4.03}$	$70.0_{6.7}$	$22.3_{2.7}$	$83.00_{6.53}$	$33.2_{2.6}$	$48.33_{5.99}$	$58.4_{7.4}$	$19.3_{2.4}$
7→8	$245.50_{1.38}$	$98.2_{0.6}$	$228.17_{4.88}$	$92.9_{1.6}$	$91.3_{2.0}$	$33.83_{3.08}$	$22.6_{2.1}$	$24.00_{4.76}$	$70.7_{11.7}$	$16.0_{3.2}$	$22.33_{5.50}$	$8.9_{2.2}$	$10.33_{2.75}$	$46.6_{7.2}$	$4.1_{1.1}$
8→9	$242.17_{1.46}$	$96.9_{0.6}$	$226.50_{2.50}$	$93.5_{0.8}$	$90.6_{1.0}$	$21.83_{2.03}$	$14.6_{1.4}$	$16.33_{3.73}$	$74.1_{12.0}$	$10.9_{2.5}$	$9.83_{3.08}$	$3.9_{1.2}$	$4.67_{2.05}$	$49.7_{16.8}$	$1.9_{0.8}$
9→10	$239.83_{2.67}$	$95.9_{1.1}$	$225.00_{3.00}$	$93.8_{1.0}$	$90.0_{1.2}$	$16.67_{3.25}$	$11.1_{2.2}$	$12.67_{2.69}$	$75.9_{7.4}$	$8.4_{1.8}$	$4.33_{2.13}$	$1.7_{0.9}$	$1.50_{0.96}$	$32.2_{22.8}$	$0.6_{0.4}$
ANY	$250.00_{0.00}$	$100.0_{0.0}$	$250.00_{0.00}$	$100.0_{0.0}$	$100.0_{0.0}$	$150.00_{0.00}$	$100.0_{0.0}$	$141.00_{1.53}$	$94.0_{1.0}$	$94.0_{1.0}$	$250.00_{0.00}$	$100.0_{0.0}$	$240.33_{0.94}$	$96.1_{0.4}$	$96.1_{0.4}$

G HYPERPARAMETER OPTIMIZATION

We perform a stage-wise hyperparameter optimization for ReHoPER. We first tune the inference temperature $T \in \{0.2, 0.3, 0.5, 0.7\}$ while fixing the path/step configuration to $10-6$. After selecting the best temperature on a development split of iLLC L2-4 with 250 newly sampled instances, we fix this temperature and tune the path/step budget $\{n\}-\{m_{\max}\} \in \{5-12,\ 6-10,\ 10-6,\ 12-5\}$. All configurations are budget-matched to produce at most 61 candidate answers per instance. Finally, we rerun the selected configuration on the MoreHopQA dataset to assess whether the same trends generalize across datasets. Tables 12 and 13 summarize the results.

Based on Table 12, we fix the sampling temperature to $T = 0.5$. On iLLC L2-4, $T = 0.5$ yields the highest average accuracy across all three answer-format prefixes, improving over the best lower temperatures ($T \in \{0.2, 0.3\}$) by roughly 3pp for IO and CoT and about 1pp for PS, and still outperforming the higher temperature $T = 0.7$ by around 1-2pp on IO/CoT and about 0.5pp on PS. The same trend largely holds on MoreHopQA: $T = 0.5$ remains the best choice for IO and PS, and is the second best setting for CoT, indicating that this temperature transfers well across datasets. For the path/step budget in Table 13, we select the $6-10$ configuration. On iLLC L2-4, $6-10$ achieves the best average performance for PS, is tied for best for CoT, and is second-best for IO. On MoreHopQA, $6-10$ is again consistently competitive and typically within 1pp of the best configuration across IO, CoT, and PS, even though $10-6$ occasionally attains the highest score. We therefore adopt $T = 0.5$ and a $6-10$ path/step budget as a single, dataset-agnostic configuration for all subsequent experiments.

Table 12: Accuracy (%) of ReHoPER under different sampling temperatures (T) on iLLC L2-4 (development set) and MoreHopQA, using IO, CoT, and PS answer-format prefixes. Green and light green mark the best and second best result, respectively, for each (model, dataset, prompting-style) triplet.

	iLLC L2-4 (Development Set)												MoreHopQA											
	IO				CoT				PS				IO				CoT				PS			
Model	**T=.2**	**T=.3**	**T=.5**	**T=.7**	**T=.2**	**T=.3**	**T=.5**	**T=.7**	**T=.2**	**T=.3**	**T=.5**	**T=.7**	**T=.2**	**T=.3**	**T=.5**	**T=.7**	**T=.2**	**T=.3**	**T=.5**	**T=.7**	**T=.2**	**T=.3**	**T=.5**	**T=.7**
Gemma 3	6.8	6.8	5.6	7.6	5.6	7.6	5.6	8.4	4.0	4.8	6.0	6.8	64.7	67.3	64.0	66.0	66.0	66.0	66.7	67.3	70.7	71.3	74.7	73.3
gpt-oss	90.8	92.4	93.2	94.8	89.2	92.0	93.6	94.8	89.6	92.8	91.2	93.6	78.7	78.7	81.3	80.0	78.0	79.3	80.0	80.0	80.0	80.7	82.0	79.3
Llama 3.1 8B	10.8	8.8	16.8	20.0	16.4	12.4	18.4	11.2	40.4	30.0	38.4	42.4	57.3	58.7	56.7	55.3	56.0	57.3	58.0	62.0	58.7	60.7	63.3	64.7
Mistral Small	8.0	7.2	9.2	7.2	5.6	8.8	9.2	6.0	8.8	8.4	8.4	6.4	66.0	70.7	71.3	70.0	67.3	68.0	67.3	72.0	70.7	70.7	70.7	68.0
Phi-4	42.4	41.6	48.4	36.4	58.4	54.4	64.0	54.8	65.6	61.2	67.6	58.4	71.3	72.7	78.7	78.0	73.3	77.3	80.7	80.0	75.3	76.0	80.0	78.7
Qwen3 w/o Thinking	80.0	82.8	84.4	84.8	84.0	83.6	87.2	88.0	87.6	87.2	90.4	90.4	71.3	70.7	68.7	68.7	73.3	72.0	70.0	69.3	69.3	71.3	72.7	69.3
Average	39.8	39.9	42.9	41.8	43.2	43.1	46.3	43.9	49.3	47.4	50.3	49.7	68.2	69.8	70.1	69.7	69.0	70.0	70.4	71.8	70.8	71.8	73.9	72.2

Table 13: Accuracy (%) of ReHoPER under different $\{n\}$-$\{m_{\max}\}$ configurations on iLLC L2-4 (development set) and MoreHopQA, using IO, CoT, and PS answer-format prefixes. Green and light green mark the best and second best result, respectively, for each (model, dataset, prompting-style) triplet.

	iLLC L2-4 (Development Set)												MoreHopQA											
	IO				CoT				PS				IO				CoT				PS			
Model	**5-12**	**6-10**	**10-6**	**12-5**	**5-12**	**6-10**	**10-6**	**12-5**	**5-12**	**6-10**	**10-6**	**12-5**	**5-12**	**6-10**	**10-6**	**12-5**	**5-12**	**6-10**	**10-6**	**12-5**	**5-12**	**6-10**	**10-6**	**12-5**
Gemma 3	6.0	6.8	5.6	4.8	8.0	8.0	5.6	5.6	6.4	7.6	6.0	5.6	66.0	64.7	64.0	62.7	66.7	64.7	66.7	64.0	74.0	74.7	74.7	74.0
gpt-oss	94.0	94.4	93.2	94.8	93.6	93.6	93.6	93.6	93.2	94.4	91.2	92.8	78.0	78.0	81.3	80.0	78.0	81.3	80.0	79.3	79.3	82.0	82.0	78.7
Llama 3.1 8B	23.6	21.2	16.8	10.8	19.6	20.8	18.4	16.0	38.4	40.0	38.4	33.6	54.7	56.0	56.7	54.7	60.7	59.3	58.0	56.0	65.3	63.3	63.3	61.3
Mistral Small	10.0	10.0	9.2	5.6	12.8	12.8	9.2	8.0	11.2	9.6	8.4	6.4	70.7	71.3	71.3	72.0	67.3	67.3	67.3	67.3	71.3	70.7	70.7	71.3
Phi-4	52.8	51.6	48.4	34.8	66.4	66.0	64.0	64.0	68.0	67.6	67.6	64.0	76.7	78.0	78.7	77.3	79.3	78.7	80.7	78.7	78.7	79.3	80.0	78.0
Qwen3 w/o Thinking	84.4	84.4	84.4	80.8	87.6	86.8	87.2	86.4	88.0	90.8	90.4	90.4	70.0	67.3	68.7	67.3	68.7	70.7	70.0	72.0	73.3	71.3	72.7	71.3
Average	45.1	44.7	42.9	38.6	48.0	48.0	46.3	45.6	50.9	51.7	50.3	48.8	69.3	69.2	70.1	69.0	70.1	70.3	70.4	69.6	73.7	73.6	73.9	72.4

H ROBUSTNESS ACROSS REPEATED RUNS

To assess whether ReHoPER's gains are robust under stochastic decoding, we repeat each experiment three times and report the mean accuracy with standard deviation in Table 14. Overall, ReHoPER remains stable across runs, with standard deviations generally comparable to those of Self-Consistency (SC). This suggests that the observed improvements are not artifacts of a single favorable sample, but persist across independent decoding runs.

The gains are especially pronounced on the harder iLLC L2-4 benchmark. Averaged across models, ReHoPER improves over SC under all answer-format prefixes, increasing accuracy from 8.6 to 22.3 with IO, 8.1 to 24.2 with CoT, and 11.7 to 30.8 with PS. These improvements are consistent across most models and are particularly large for Phi-4, where ReHoPER improves from roughly 10–13% accuracy to more than 54–62% across prefixes, while maintaining low run-to-run variation.

On MoreHopQA, where the base SC performance is already substantially higher, the improvements are smaller but still generally positive on average. ReHoPER matches or improves the average SC result under all three prefixes, with the clearest gains under CoT and PS. While a few individual model-prefix combinations show small decreases, the overall pattern indicates that ReHoPER does not rely on a fragile prompting choice and its benefits are strongest in the more compositional iLLC setting and remain stable across repeated runs and answer-format prefixes.

Table 14: Robustness of ReHoPER (RH) across repeated runs on MoreHopQA and iLLC L2-4 under IO, CoT, and PS answer-format prefixes. We report accuracy (%) as mean_{std} over three runs. SC denotes simple question answering with Self-Consistency using the corresponding output prefix, while RH applies our planning approach but evaluates with the same prefix. Green, red, and gray indicate performance changes relative to SC.

	MoreHopQA						iLLC L2-4					
	IO		CoT		PS		IO		CoT		PS	
Model	**SC**	**RH**	**SC**	**RH**	**SC**	**RH**	**SC**	**RH**	**SC**	**RH**	**SC**	**RH**
Gemma 3	$61.8_{0.4}$	$65.8_{1.0}$	$64.7_{0.7}$	$68.7_{0.0}$	$66.7_{0.7}$	$74.7_{1.2}$	$7.3_{0.2}$	$6.9_{1.3}$	$7.6_{0.0}$	$6.9_{0.9}$	$3.6_{0.8}$	$8.1_{1.7}$
Llama 3.1 8B	$60.7_{1.2}$	$57.1_{1.7}$	$60.0_{0.7}$	$60.0_{1.8}$	$62.4_{1.5}$	$63.1_{0.4}$	$11.9_{0.8}$	$17.5_{4.1}$	$12.5_{3.2}$	$18.4_{2.0}$	$30.4_{1.1}$	$41.6_{1.4}$
Mistral Small	$70.2_{0.4}$	$71.3_{0.0}$	$69.1_{1.0}$	$68.7_{1.2}$	$72.2_{0.4}$	$71.1_{1.4}$	$1.9_{0.5}$	$10.5_{0.6}$	$2.4_{0.4}$	$9.6_{1.7}$	$3.1_{0.8}$	$10.8_{1.2}$
Phi-4	$77.3_{0.7}$	$76.0_{1.8}$	$76.4_{0.4}$	$80.4_{1.0}$	$77.6_{0.4}$	$78.4_{1.7}$	$13.2_{0.4}$	$54.3_{0.8}$	$10.0_{0.4}$	$62.0_{1.7}$	$9.7_{0.8}$	$62.7_{1.3}$
Average	$67.5_{0.7}$	$67.6_{1.1}$	$67.6_{0.7}$	$69.5_{1.0}$	$69.7_{0.8}$	$71.9_{1.2}$	$8.6_{0.5}$	$22.3_{1.7}$	$8.1_{1.0}$	$24.2_{1.6}$	$11.7_{0.9}$	$30.8_{1.4}$

I PLANNING ABLATION DETAILS

The planning ablations retain ReHoPER's initial prediction, Answer, Update, Predict, and Aggregation steps (Section 3.2), with the same decoding temperature, output prefixes, number of paths, and maximum step count. They change question generation, selection, and path termination as follows.

Without replanning. At the start of each path i, a single call $f(C, I_{\text{fixed}})$ generates an ordered list $\mathbf{IQ}^{(i)} = \langle IQ_1^{(i)}, \ldots, IQ_{m_i}^{(i)} \rangle$, whose length is determined by the model and can vary across paths. At step k, the Plan and Select steps are replaced by indexing this list. Questions are answered in order without revising the list or applying ReHoPER's candidate-selection rule. The path ends after $\min(m_i, m_{\max})$ steps.

Without look-ahead. At each step, a single call $f(C, \mathbf{H}_{k-1}^{(i)}, I_{\text{single}})$ generates the next intermediate question. This replaces both Plan and Select: there is no candidate horizon or separate check for selecting an unasked question. The path runs to $m_{\max}$, even if a question repeats. The prompt nevertheless instructs the model to avoid repetition.

The exact instructions are shown in Figures 14 and 15. Because these configurations also change selection and termination, the comparisons assess the combined planning design rather than isolate a single mechanism with all other behavior held fixed. Equal path and step limits do not imply equal realized trajectory lengths or compute use. Table 2 in the main paper reports the complete accuracy breakdown.

J iLLC L1-8 AND iLLC L2-8 RESULTS

Table 15 reports the full per-model accuracy breakdown on iLLC L1-8 and iLLC L2-8 under IO, CoT, and PS prefixes.

Table 15: Accuracy (%) of ReHoPER (RH) across iLLC L1-8 and iLLC L2-8 benchmarks under IO, CoT, and PS answer-format prefixes. SD refers to the Self-Discover baseline. Furthermore, SC denotes simple question answering with self-consistency using the corresponding output prefix, while RH applies our planning approach but evaluates with the same prefix. Green, red, and gray indicate performance changes relative to SC.

	iLLC L1-8							iLLC L2-8						
		IO		CoT		PS			IO		CoT		PS	
Model	**SD**	**SC**	**RH**	**SC**	**RH**	**SC**	**RH**	**SD**	**SC**	**RH**	**SC**	**RH**	**SC**	**RH**
Gemma 3	26.4	7.6	12.4	36.0	28.8	52.8	32.8	0.0	0.4	0.0	0.0	0.0	0.0	0.0
gpt-oss	58.8	84.0	76.4	78.4	74.8	77.6	76.0	45.6	76.4	80.0	61.6	80.8	6.0	81.2
Llama 3.1 8B	7.6	31.2	27.6	24.0	39.2	39.2	34.4	0.0	0.8	2.0	0.0	4.0	0.4	2.4
Llama 3.1 70B	56.0	59.2	62.8	48.8	63.2	68.8	70.4	0.8	1.2	44.0	3.6	46.4	12.8	54.0
Mistral Small	23.6	64.0	57.2	62.4	57.2	60.4	59.6	0.0	0.0	0.0	0.0	0.4	0.0	0.0
Phi-4	53.6	53.6	55.6	60.0	62.8	60.8	61.2	2.0	0.8	16.8	1.2	33.6	1.6	35.6
Qwen3 w/o Thinking	44.4	50.8	68.8	54.8	67.2	69.2	72.0	0.0	15.2	33.6	8.8	32.8	60.0	43.6
Qwen3 w/ Thinking	1.2	85.2	80.4	54.8	75.2	69.2	78.8	3.6	76.8	72.8	8.8	76.0	60.0	76.4
Average	34.0	54.5	55.2	52.4	58.6	62.3	60.7	6.5	21.5	31.2	10.5	34.3	17.6	36.7

K MODEL DETAILS

Table 16 summarizes the language models used in our experiments, including their model families, parameter scales, context lengths, and the exact checkpoints used for inference. Our model suite is chosen to cover a deliberately diverse range of instruction-tuned Large Language Models (LLM)s: smaller dense models such as LLAMA-3.1-8B-INSTRUCT (Grattafiori et al., 2024), mid-sized dense models such as GEMMA-3-12B-IT (Team et al., 2025), PHI-4 (Abdin et al., 2024), and MISTRAL-SMALL-24B-INSTRUCT-2501 (Mistral AI, 2025), as well as larger-capacity models obtained either through scale, such as LLAMA-3.1-70B, or sparsity, such as GPT-OSS-20B (Agarwal et al., 2025) and QWEN3-30B-A3B (Yang et al., 2025). This selection covers both general instruction-following models and models explicitly positioned toward stronger reasoning behavior.

From a systems perspective, we use quantized deployments for larger models when needed to reduce inference memory requirements. Specifically, we use 4-bit weight-only AWQ variants (Lin et al., 2024) for Llama 3.1 70B and Mistral Small, and a GPTQ variant (Frantar et al., 2023) for Qwen3. For Qwen3, we report two configurations: thinking disabled and thinking enabled, using the model's built-in switching mechanism; unless otherwise specified, Qwen3 refers to the non-thinking configuration. We implement our experiments in PyTorch (Paszke et al., 2019) and perform inference using vLLM (Kwon et al., 2023), together with the transformers (Wolf et al., 2019), NumPy (Harris et al., 2020), and Matplotlib (Hunter, 2007) libraries. Experiments were conducted on a multi-GPU setup equipped with NVIDIA RTX A6000 48GB and NVIDIA RTX A5000 24GB GPUs.

Table 16: Models used in our experiments. We list the exact checkpoint, model family, parameter scale, and context length reported by the corresponding model card.

Model (as used)	**Org / Family**	**Params**	**Context**
google/gemma-3-12b-it	Google / Gemma 3	12B	128K
openai/gpt-oss-20b	OpenAI / gpt-oss	21B (3.6B active)	128K
meta-llama/Llama-3.1-8B-Instruct	Meta / Llama 3.1	8B	128K
hugging-quants/Meta-Llama-3.1-70B-Instruct-AWQ-INT4	Meta / Llama 3.1	70B	128K
stelterlab/Mistral-Small-24B-Instruct-2501-AWQ	Mistral / Mistral Small 3	24B	32K
microsoft/phi-4	Microsoft / Phi	14B	16K
Qwen/Qwen3-30B-A3B-GPTQ-Int4	Qwen / Qwen3	30.5B (3.3B active)	32K

L INTERMEDIATE QUESTION INSTRUCTIONS

This appendix summarizes the exact intermediate question generation instructions used in our experiments. Figures 13, 14, and 15 show the prompts for ReHoPER (I_{rh}), the ablation without replanning (I_{fixed}), and the ablation without look-ahead (I_{single}), respectively.

Instruction I_{rh} used for ReHoPER

Your ultimate goal is to answer "{Q^*}".

Process:
1. **Plan** — Think step-by-step to create a logical sequence of sub-questions. These questions, when answered sequentially, should lead to the answer for the ultimate goal. Ensure each sub-question advances the solution without merely rephrasing the goal or duplicating a prior question.
2. **Generate** — Draft the list of sub-questions based on your plan.

Important: Do not answer the ultimate question or any of the sub-questions. Your task is only to write the questions.

Output format (strict):
THOUGHTS:
<your step-by-step thinking and planning process to arrive at the sub-questions>
SUB-QUESTIONS:
1. <first sub-question>
...
N. <N-th sub-question>

Figure 13: Instruction I_{rh} used for ReHoPER.

Instruction I_{fixed} used for the ablation without replanning

Your ultimate goal is to answer "{Q^*}".
We will solve this step by step. Based on the information you currently have and considering that no further context will be provided, identify the questions that will extract the most relevant information needed to progress toward solving the ultimate problem. You do not need to answer these questions. Ensure that each question serves as a stepping stone rather than simply restating the ultimate question and moves the process forward by exploring new aspects rather than repeating previously asked questions. Outline the questions to be answered in order before the ultimate question.

Output format (strict):
THOUGHTS:
<your step-by-step thinking and planning process to arrive at the sub-questions>
SUB-QUESTIONS:
1. <first sub-question>
...
N. <N-th sub-question>

Figure 14: Instruction I_{fixed} used for the ablation without replanning.

Instruction I_{single} used for the ablation without look-ahead

Your ultimate goal is to answer "{Q^*}".
We will solve this step by step. Based on the information you currently have, identify the first question that can provide the most relevant information needed to progress toward solving the ultimate problem. You do not need to answer the question. Ensure this question serves as a stepping stone rather than simply restating the ultimate question and moves the process forward by exploring new aspects rather than repeating any previously asked questions. Finally, the last thing you generate should be "QUESTION: (your question that can extract the most useful information to help solve the problem)"

Figure 15: Instruction I_{single} used for the ablation without look-ahead.

M DATASET DETAILS AND EXAMPLES

MuSR. MuSR evaluates multi-step reasoning over long, story-like inputs where the answer requires integrating scattered clues and applying commonsense inference rather than relying on surface matching (Sprague et al., 2024). It contains three tasks: Murder Mysteries (MM), where the model identifies the culprit from a narrative; Object Placement (OP), where the model tracks objects as they move or are relocated through a sequence of events; and Team Allocation (TA), where the model infers valid group assignments under narrative constraints.

MoreHopQA. MoreHopQA is a multi-hop question answering benchmark designed to test whether models can connect multiple evidence pieces across a context and perform sequential reasoning hops to reach the correct answer (Schnitzler et al., 2024). We use the 150-example subset selected by the dataset authors, following the original evaluation setup.

iLLC. iLLC is our controlled diagnostic benchmark for compositional reasoning. It generalizes Last Letter Concatenation (LLC) (Wei et al., 2022b) by requiring the model to extract the i-th-to-last character from each word in a given list and concatenate the resulting letters. We denote subsets as L$\{i\}$-$\{w\}$, where i specifies the target character position from the end and w specifies the number of words in the input. For example, L1-4 corresponds to standard last-letter concatenation over four words, while L2-4 requires extracting the second-to-last character from each of four words.

Post-processing. Let R denote the model response to the goal question Q^*, and let $\hat{y}$ denote the extracted prediction. For MoreHopQA and iLLC, we obtain $\hat{y}$ by extracting the first word that appears immediately after `"ANSWER:"`. For MuSR, where answers are multiple-choice option numbers, we instead extract the most frequent option number that appears after `"ANSWER:"` in R.

Table 17 summarizes the datasets used in our evaluation, including their task variants, test-set sizes, and answer formats. The examples below illustrate MuSR (MM/OP/TA), MoreHopQA, and iLLC (L1-4/L2-4).

Table 17: Summary of reasoning datasets used for evaluation.

Dataset	Task	Test Set Size Per Task	Answer Format
MuSR	Murder Mysteries, Team Allocation Object Placement	250 256	Choice number
MoreHopQA	-	150	Short text answer
iLLC	L1-4, L1-6, L1-8, L2-4, L2-6, L2-8	250	Concatenated letters

An example from MuSR MM

In an adrenaline inducing bungee jumping site, Mack's thrill-seeking adventure came to a gruesome end by a nunchaku; now, it's up to Detective Winston to unravel the deadly secrets between Mackenzie and Ana.

Winston took a gulp of his black coffee, staring at the notes sprawled across his desk. A murder case at a bungee jumping site was definitely out of the ordinary. Today's victim was a young man named Mack, loud mouthed and cocky by all accounts.

Mack was bungee jumping the day he was killed. Oddly enough, according to the records, no one else was documented at the bungee jumping site that day, making this case even more peculiar. The first stop for the day was to visit one of Mack's housemates, a woman named Ana. They were seen leaving in the same vehicle from their shared housing complex the morning of the murder, and it was time for Winston to dig deeper.

As he pulled into the shared housing driveway, a nondescript car came into sight. He learned from neighbours that it was frequently used by multiple residents, but Ana had a peculiar interest in it. She would insist on driving whenever with a group of friends, later meticulously cleaning the car after each use. An idiosyncrasy of hers maybe, but a part of the puzzle nonetheless.

Winston knocked on the door, Ana opened it warily, twiddling a cleaning cloth and spray in her hands and greeted him with a nervous nod. Ana gets nervous and fidgets with the cleaner and cloth when questioned. Winston could sense palpable unease as he started asking her questions.

"Ana, did you not join Mack and the others for bungee jumping today?" Winston questioned, to which she responded, "I signed up to jump. But I didn't end up going through with it."

"Any particular reason you didn't join the others, Ana?" Winston proceeded.

Ana took a deep breath, "Well sir, my faith doesn't really permit bungee jumping. Truth be told, I was persuaded strongly by Mack. I had even signed up out of peer pressure but couldn't push myself."

It was true – Mack was insisting that everyone in the group should bungee jump. Mack had reportedly also been vocal about ridiculing Ana's faith, even encouraging others to join him in doing so. It was a significant factor in their relationship.

"Ana, did you and Mack leave in the same car for the bungee jumping event this morning?" Winston gently pushed further.

"Yes. Yes, we did. We always carpool." She responded while anxiously using the cleaner and cloth on her car's dashboard. Her eyes flickered nervously back to Winston, expecting the next question.

Winston took a deep breath, standing up to leave, "Alright Ana, that should cover everything for now. We'll be in touch."

Ana nervously nodded without looking up from her cleaning, wringing the cloth repeatedly as Winston walked away, left again with another piece to the enigmatic puzzle of Mack's murder.

The day was getting older and Winston was getting more tired, but the case was fresh, and he wasn't one to back down. He tugged on his coat as he approached the bashful teen waiting for him by the police station.

"Mackenzie, it is?" he asked, extending his hand.

"Yeah, that's right." The slight lisp, overlaid with blanket anxiety, confirmed what the school reports suggested.

"You were at the site when Mack... erm... you know," Winston's voice was methodical, calm – almost robotic. The suspicion on Mackenzie was not unfounded - the security cameras showed him buying nunchaku a week before.

Mackenzie shifted on his feet, looking away before answering, "Yeah, I was there."

Winston pulled out a small notebook, "What were you doing there, Mackenzie?"

"Bungee jumping, like Mack. . . Then I left. I didn't... I didn't do anything. . . " Mackenzie replied.

Internally, Winston sighed at the never-ending waterfall of teenage angst this case was turning into.

"Martial arts, huh?" Winston segued, gesturing to a bruise on Mackenzie's knuckles. "Nunchaku particularly, I see? Training does include the use of those, correct?"

The change in Mackenzie's demeanor mirrored the bitterness in the last month's weather – dark eyes replaced with ice-cold ones. "Yeah," he admitted, shrinking slightly.

Mackenzie always took pride in being the best at everything. So when Mack got everything he wanted - the promotion to team captain, the respect, the attention - it was a hard pill for Mackenzie to swallow. Winston remembered the team talk, Mackenzie was indeed the top candidate but it had gone to Mack instead.

What clinched it was Mackenzie's remarks about Mack, echoing whispers of dispute and bickering, lost in the crowded lunchroom. There were also multiple witness reports of the two seen arguing at the bungee jumping site previously. Mackenzie had indeed said disparaging, almost emotional things about Mack – all stemming from a potent brew of jealousy, Winston inferred.

Shifting later through the detritus of Mackenzie's life, Winston discovered the nunchaku that matched the forensics report. They were tucked away, but the layer of dust suggested they weren't a favored possession anymore. It wasn't hidden, it was misplaced – discarded in the throes of developing maturity.

As the sun started to set, Winston could see witnesses, scattered across the park, repeatedly pointing to the bungee jumping scaffolding. It occurred to him, then, the narrative of the past days. Mackenzie, jealous and wronged, over and over, at the same sight. It was quite a sight.

Winston, shuffling back to the station, was left with one thought - Looks like Mackenzie had quite an eventful week.

Who is the most likely murderer?
1 - Mackenzie
2 - Ana

You must pick one option. Finally, the last thing you generate should be "ANSWER: (your answer here, include the choice number)"

An example from MuSR OP

In the heart of the bustling studio, Ricky, Emma, and Danny readied themselves for a day of creating magic. Ricky, holding the helm as the gifted singer-songwriter, was poised for perfection, his precious notebook of lyrics awaiting its call to duty on the producer's desk. Emma, their dutiful and talented producer, was just as eager to breathe life into Ricky's lyrics. She was cognizant of the notebook's place at her desk, awaiting the melodies they would cultivate together.

Across the room, Danny, the diligent studio assistant, was doing his due diligence, keeping the earphones nestled in the recording booth. His aim was to ensure an optimized and meticulous environment for recording, a testament to his commitment to their shared mission. They were all aware of the arrangement — the notebook on the producer's desk, the earphones in the recording booth. Their shared consciousness of these items only intensified the anticipation; they were eager to turn the contents of a weathered notebook into a world-class album.

Ricky, with his weathered notebook of potent lyrics in hand, gently places it onto the piano. An air of creativity and anticipation lingers in the room, everyone aware that this was the first instrumental step in the creation of their masterpiece. In sync with the palpable creative energy, Ricky was engrossed in perfecting the rhythm of his song, preparing himself for an intense day ahead. Not too far away, Emma was sincerely engrossed in her role of musically steering the session. She was focussed on Ricky's progress, her eyes constantly monitoring him and her mind alive with ideas to enhance the music.

Meanwhile, Danny was diligently covering every corner of the studio. He was making his rounds, ensuring that the studio was prim and proper for Ricky's crucial session. As part of his tasks, he passed by Ricky several times, always careful not to interrupt the artist's flow.

Emma, engrossed in her thoughts, deftly moves the earphones to the producer's desk. She is preparing to tweak the sound settings, pre-empting Ricky's need for perfect audio in his performance. Diverting from his rounds, Danny found himself in the midst of a stirring conversation with a visiting sound engineer. Knowledge flowed between them, illuminating the studio's atmosphere, the engineer's insight bringing a new perspective into Danny's role. Ricky, ensconced in his own world, was in deep discussion with the blank page before him. The daunting silence of the empty studio buzzed with his focus, as he honed his lyrics to perfection in a space separate from the producer's. The visitor, oblivious to the careful choreography of the studio session, stood blocking Danny's general overview of the studio space.

Delicately lifting Ricky's notebook, Danny orchestrates its move to the producer's desk. At the desk, he glimpses a pair of earphones indirectly drawing his attention amidst his routine of tidying up. Emma, from the isolated interior of a soundproofed booth, lent her ears diligently to already recorded tracks, pouring over them for any room for improvement. Being lost in the music was her way of paying homage to her craft - an unspoken ritual she followed each time she embarked on a music production journey. The entirety of her focus was consumed by the musical notes and rhythm filtering through the studio speakers.

Concurrently, Ricky was absorbed in the act of playing his guitar. His fingers navigated deftly over the strings, lost in an intimate dance with the instrument. As he played, the melodic strums reverberated throughout the studio, filling it with an infectious pulse that hinted at the birth of yet another musical masterpiece. Despite the flurry of activity around him, Ricky was lost in a world of his own, operating on a singular vision of delivering his best performance.

In the meantime, Danny was continuing his cautious management of the studio, ensuring that everything fell into place for the optimum recording session. His watchful eyes were scanning every corner, taking stock of the minor details that could impact the session. However, the design of the studio didn't allow for an unrestricted view into all the corners. The sound booth, where Emma was engrossed in her work, was out of his visual range. The seclusion provided by the booth, although crucial for immersive work, also acted as a barrier for Danny's comprehensive vigilance.

As the day progressed, the studio was entwined in a concerted symphony of dedication and workmanship, the trio, each engrossed in their pursuit, working together to create the best version of Ricky's impending album. As the final note of the day rang through the studio, each person revelled in the satisfaction of another day done right, another step closer towards the realization of Ricky's artistic vision.

Within the dynamic dance of the day's events, the relationships of the trio sang a compelling tune. Each individual played their crucial part in the creation of the impending masterpiece - Ricky with his raw talent, Emma with her passion for perfection, and Danny with his meticulous eye for detail. And as the lights faded on another day of creation, they could sense the beginning of an important chapter in their artistry, a silence collecting the scattered notes of the day, signing off on another critical step in the journey of Ricky's upcoming album.

Which location is the most likely place Danny would look to find the earphones given the story?
1 - piano
2 - producer's desk
3 - recording booth

You must pick one option. Finally, the last thing you generate should be "ANSWER: (your answer here, include the choice number)"

An example from MuSR TA

Amidst the vibrant chaos of the Redwood Zoo, nestled in the heart of the city's sprawling jungle, the task of assigning roles was a crucial cog in the machinery of its operation. As the manager, the responsibility of allocating Olivia, Alex, and Mia to the positions of Animal Caretaker and Exhibit Cleaner presented an intriguing conundrum. Each individual, with their distinct personalities and skill sets, added a layer of complexity to this assignment puzzle.

Let's begin with Alex, the tall lad with bright eyes, whose history with the mighty beast of the animal kingdom, lacked a certain comfort. The lad, known to express an almost innate unease around animals larger than him, fell short of the prerequisites for an Animal Caretaker. His comfort zone extended to the four-legged companions in our homes, a sentiment I withheld from the petting zoo section of our park. Yet, his association and collaboration with Mia had seen quite the successes in their high school club's fundraising initiatives.

However, his relationship with the gentle Olivia was not as seamless. Alex often mentioned feeling ostracized due to Olivia's tendency to maintain her distance. This seemingly innocent avoidance stirred disquiet within our hushed ranks. And all this, stemming from a disagreement rooted in their previous shared workplaces. Unresolved perhaps, but a factor nonetheless.

Then, there was Mia, the determined bright spark, whose affinity for cleanliness would often bemuse us. She would spend her spare time in her immaculate home cleaning and reorganizing, while her enthusiasm for a spotless Exhibit could not be underestimated. However, her overly thorough methods would invariably result in clashes with Olivia, who criticized her for crossing some form of unspoken boundary.

Mia too had her phobias, the gravelly roars of the zoo's majestic lion had once left her shaken and worried. Loud noises had a similar effect leaving her in a state of nervous terror, much like that of the petite animals held within our barriers. Yet, she was all smiles and peasant conversation around Alex during lunch breaks, sharing a sense of humor that lightened the mood of our everyday grind.

Finally, subdued Olivia, a soul strangled with allergies, and a deep-seated fear for wild animals. An incident with a chimp in her past wove tales of nightmarish betrayal, enough to send her away from the animal exhibits during her zoo visits. Potent elements of dust and pollen resulted in uncontrolled sneezing fits, a remainder from her days at the school as a custodian, responsible for the cleanliness and maintenance.

Three souls; Animals to be cared for, Exhibits to be cleaned. Assigning them was always going to be an enigma for anyone navigating the zoological labyrinth. Love for animals, discomfort, alliances, conflicts; each factor extraordinarily crucial in shaping not just the overall productivity but also the personal growth of each of these individuals at the Redwood Zoo.

Given the story, how would you uniquely allocate each person to make sure both tasks are accomplished efficiently?
1 - Animal Caretaker: Alex, Exhibit Cleaner: Mia and Olivia
2 - Animal Caretaker: Olivia, Exhibit Cleaner: Alex and Mia
3 - Animal Caretaker: Mia, Exhibit Cleaner: Alex and Olivia

You must pick one option. Finally, the last thing you generate should be "ANSWER: (your answer here, include the choice number)"

An example from MoreHopQA

Context: Maroon 5: Maroon 5 is an American pop rock band that originated in Los Angeles, California. It currently consists of lead vocalist Adam Levine, keyboardist and rhythm guitarist Jesse Carmichael, bassist Mickey Madden, lead guitarist James Valentine, drummer Matt Flynn and keyboardist PJ Morton. What Lovers Do: "What Lovers Do" is a song by American pop rock band Maroon 5 featuring American R&B singer Sza. It was released on August 30, 2017, as the third single from the band's upcoming sixth studio album (2017). The song contains an interpolation of the 2016 song "Sexual" by Neiked featuring Dyo, therefore Victor Rådström, Dyo and Elina Stridh are credited as songwriters.
Question: What is the ASCII code of the last letter of the first name of the current drummer of the band who did the song "What Lovers Do"?
The last thing you generate should be "ANSWER: (your answer here)"

An example from iLLC L1-4

Question: Take the last letters of the words in "Taylor Nancy Quintin Earl" and concatenate them.
The last thing you generate should be "ANSWER: (the concatenated letters without any spaces or special characters)"

An example from iLLC L2-4

Question: Take the second-to-last letters of the words in "Taylor Nancy Quintin Earl" and concatenate them.
The last thing you generate should be "ANSWER: (the concatenated letters without any spaces or special characters)"

N SAMPLE iLLC L2-4 DIALOGUES

The positive and negative examples below illustrate two iLLC L2-4 instances, showing intermediate questions, answers, and final predictions from our ReHoPER experiments.

A positive iLLC L2-4 example using ReHoPER

User: (IQ_1)
What are the individual words in the name "Zoe Brock Clark Sanders"?

Assistant: (A_1)
The individual words in the name "Zoe Brock Clark Sanders" are:

1. Zoe
2. Brock
3. Clark
4. Sanders

User: (IQ_2)
What is the second-to-last letter of the word "Zoe"?

Assistant: (A_2)
The word "Zoe" has three letters: Z, o, and e. The second-to-last letter is "o".

User: (IQ_3)
What is the second-to-last letter of the word "Brock"?

Assistant: (A_3)
The word "Brock" has five letters: B, r, o, c, and k. The second-to-last letter is "c".

User: (IQ_4)
What is the second-to-last letter of the word "Clark"?

Assistant: (A_4)
The word "Clark" has five letters: C, l, a, r, and k. The second-to-last letter is "r".

User: (IQ_5)
What is the second-to-last letter of the word "Sanders"?

Assistant: (A_5)
The word "Sanders" has seven letters: S, a, n, d, e, r, and s. The second-to-last letter is "r".

User: (IQ_6)
What is the result of concatenating the second-to-last letters of the words "Zoe", "Brock", "Clark", and "Sanders"?

Assistant: (A_6)
The second-to-last letters of the words are as follows:
- "Zoe": o
- "Brock": c
- "Clark": r
- "Sanders": r

Concatenating these letters results in the string "ocrr".

User: (IQ^*)
Question: Take the second-to-last letters of the words in “Zoe Brock Clark Sanders” and concatenate them.
The last thing you generate should be "ANSWER: (the concatenated letters without any spaces or special characters)"

Assistant: (R_6)
Let’s think step by step. We need to identify the second-to-last letter of each word in "Zoe Brock Clark Sanders" and then concatenate them.

1. **Zoe**: The second-to-last letter is "o".
2. **Brock**: The second-to-last letter is "c".
3. **Clark**: The second-to-last letter is "r".
4. **Sanders**: The second-to-last letter is "r".

Now, concatenate these letters: "ocrr".

ANSWER: ocrr

Ground Truth: (y)
ocrr

A negative iLLC L2-4 example using ReHoPER

User: (IQ_1)
What are the individual words in the phrase "Dan Rudy Kash Yolanda"?

Assistant: (A_1)
The phrase "Dan Rudy Kash Yolanda" is composed of the following individual words:

1. Dan
2. Rudy
3. Kash
4. Yolanda

Each of these appears to be a proper noun, likely representing names.

User: (IQ_2)
What is the length of each word in the phrase "Dan Rudy Kash Yolanda"?

Assistant: (A_2)
Here are the lengths of each word in the phrase "Dan Rudy Kash Yolanda":

1. Dan - 3 letters
2. Rudy - 4 letters
3. Kash - 4 letters
4. Yolanda - 7 letters

The total number of letters in the phrase is 18.

User: (IQ_3)
How many letters are in each word: Dan, Rudy, Kash, Yolanda?

Assistant: (A_3)
Here are the number of letters in each word:

1. Dan - 3 letters
2. Rudy - 4 letters
3. Kash - 4 letters
4. Yolanda - 7 letters

In total, there are 18 letters in the phrase "Dan Rudy Kash Yolanda."

User: (IQ_4)
How many letters are in the word "Dan"?

Assistant: (A_4)
The word "Dan" has 3 letters.

User: (IQ_5)
What is the second-to-last letter of the word "Dan"?

Assistant: (A_5)
The second-to-last letter of the word "Dan" is "a".

User: (IQ_6)
What is the length of the word "Dan"?

Assistant: (A_6)
The word "Dan" has a length of 3 letters.

User: (IQ_7)
What is the second-to-last letter of the word "Rudy"?

Assistant: (A_7)
The second-to-last letter of the word "Rudy" is "d".

User: (IQ_8)
What is the second-to-last letter of the word "Kash"?

Assistant: (A_8)
The second-to-last letter of the word "Kash" is "a".

User: (IQ_9)
What is the second-to-last letter of the word "Yolanda"?

Assistant: (A_9)
The second-to-last letter of the word "Yolanda" is "n".

User: (IQ_{10})

Concatenate the second-to-last letters of the words "Dan", "Rudy", "Kash", and "Yolanda" in that order.

Assistant: (A_{10})
The second-to-last letters of the words "Dan", "Rudy", "Kash", and "Yolanda" are "a", "d", "a", and "n", respectively. Concatenating these letters in that order gives you "adad".

User: (IQ^*)
Question: Take the second-to-last letters of the words in “Dan Rudy Kash Yolanda” and concatenate them.
The last thing you generate should be "ANSWER: (the concatenated letters without any spaces or special characters)"

Assistant: (R_{10})
Let’s think step by step. The phrase is "Dan Rudy Kash Yolanda". We will determine the second-to-last letter of each word:

1. "Dan" - The second-to-last letter is "a".
2. "Rudy" - The second-to-last letter is "d".
3. "Kash" - The second-to-last letter is "a".
4. "Yolanda" - The second-to-last letter is "n".

Now, we concatenate these letters in the order they appear: "a", "d", "a", "n".

ANSWER: adan

Ground Truth: (y)
adsd