

000 001 002 003 004 005 EXPLORE-EXECUTE CHAIN: TOWARDS AN EFFICIENT 006 STRUCTURED REASONING PARADIGM 007 008 009

010 **Anonymous authors**
011 Paper under double-blind review
012
013
014
015
016
017
018
019
020
021
022
023
024
025
026
027
028
029
030
031

ABSTRACT

032 Chain-of-Thought (CoT) and its variants have markedly advanced the reasoning abilities of Large Language Models (LLMs), yet their monolithic and auto-regressive architecture inherently conflates high-level strategic planning with low-level step-by-step execution, leading to computational inefficiency, limited exploration of reasoning paths, and reduced interpretability. To overcome these issues, we propose the **Explore-Execute Chain (E²C)**, a structured reasoning framework that decouples reasoning into two distinct phases: an exploratory phase that stochastically generates succinct high-level plans, followed by an execution phase that deterministically carries out the chosen plan. Our approach incorporates a two-stage training methodology, which combines Supervised Fine-Tuning (SFT)—augmented by a novel data generation algorithm enforcing strict plan adherence—with a subsequent Reinforcement Learning (RL) stage that capitalizes on the informativeness of exploration and reinforces the determinism of execution. This decomposition enables an efficient test-time scaling strategy: on AIME’2024, **E²C Test Time Scaling** reaches 58.1% accuracy using <10% of the decoding tokens required by comparable methods (e.g., Forest-of-Thought), sharply cutting self-consistency overhead. For cross-domain adaptation, our **Exploration-Focused SFT (EF-SFT)** fine-tunes with only 3.5% of the tokens used by standard SFT yet yields up to 14.5% higher accuracy than **standard SFT** on medical benchmarks, delivering state-of-the-art performance, strong generalization, and greater interpretability by separating planning from execution.

1 INTRODUCTION

033 Large Language Models (LLMs) have demonstrated remarkable capabilities in complex reasoning, largely propelled by techniques such as Chain-of-Thought (CoT) prompting (Wei et al., 2022). This paradigm has inspired a suite of advanced methods, including sampling multiple reasoning paths for consensus via Self-Consistency (Wang et al., 2022), and exploring the solution space with more complex structures like Tree-of-Thoughts (ToT) (Yao et al., 2023), Graph-of-Thoughts (GoT) (Besta et al., 2023), and Forest-of-Thought (FoT) (Bi et al., 2025). Other approaches focus on iterative refinement through self-correction (Shinn et al., 2023) or problem decomposition (Zhou et al., 2023; Yao et al., 2022).

041 Despite their success, these methods are predominantly founded on a monolithic, auto-regressive generation process that conflates two fundamentally different cognitive functions: high-level strategic **planning** and low-level, step-by-step **execution**. This entanglement leads to critical inefficiencies. First, the model expends equivalent computational effort on both creative planning and routine calculations, a challenge addressed by works on adaptive computation (Xu et al., 2025b) and reasoning compression (Li et al., 2025). Second, the greedy generation process restricts the diversity of initial strategies, where a suboptimal early choice can derail the entire reasoning path. This is a key problem that sophisticated test-time scaling methods (Liao et al., 2025; Xu et al., 2025a) and structured exploration frameworks (Zheng et al., 2025a) aim to mitigate.

050 In this work, we argue that explicitly decoupling these two functions is crucial for advancing reasoning in large language models. We introduce the **Explore-Execute Chain (E²C)**, a framework 051 that decomposes standard CoT into two distinct phases. The first phase is a highly informative **exploration** 052 stage, in which the model generates a concise, high-level plan. This stage provides a 053 quick preview of the complete reasoning process—analogous to hierarchical planning (Gui et al.,

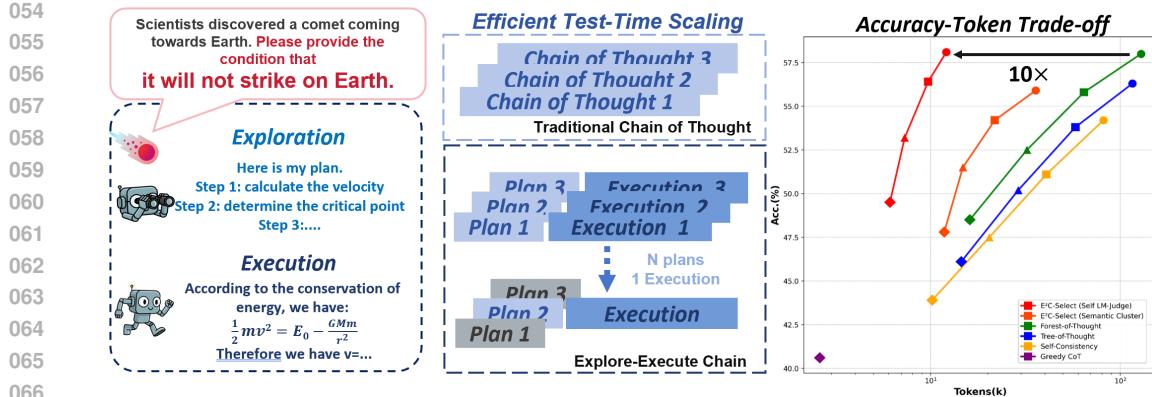


Figure 1: Our proposed **Explore-Execute Chain (E²C)** method decomposes reasoning chains into a short, high-level exploratory plan followed by a long, detailed execution (left). After optimizing these special reasoning chains using RL, it is possible to synthesize a large number of plans, use the model to pick the best plan, and then execute this plan (middle). This unlocks dramatically improved overall token efficiency on the challenging AIME’2024 benchmark (right).

—without incurring the cost of full-chain generation. The second phase is a highly deterministic **execution** stage, which takes the plan as guidance and meticulously performs the detailed calculations. This stage emphasizes precision and faithful adherence to the chosen strategy, a requirement that necessitates specialized training (Zheng et al., 2025b).

This decomposition enables a highly efficient **test-time scaling** strategy (Fig. 1). Rather than generating multiple costly, full reasoning chains (Wang et al., 2022), E²C samples a larger set of inexpensive **exploration** plans while executing fewer **execution** steps. The most promising exploration plans are selected via semantic clustering or an LLM, leveraging the high informativeness of the exploration phase for effective filtering. The chosen plan is then executed with high determinism, ensuring reliable and precise reasoning. This approach improves the performance–cost trade-off (Geiping et al., 2025; Liao et al., 2025) and enhances interpretability. We implement this framework using a two-stage (SFT+RL) training pipeline, guided by recent advances in reasoning alignment (Gan et al., 2025; Rafailov et al., 2023).

Our main contributions are summarized as follows:

- We propose the Explore–Execute Chain (E²C), which decouples LLMs’ reasoning into a highly informative Exploration stage for planning and a highly deterministic Execution stage for carrying out the plan, thereby improving efficiency and interpretability.
- We introduce a robust two-stage training methodology (SFT+RL) together with a specialized data construction algorithm that ensures the model faithfully adheres to its plans, effectively instilling E²C paradigm and achieving superior performance.
- We demonstrate the efficiency of this framework with two key results: an efficient test-time scaling strategy that achieves 58.1% accuracy on AIME’2024 using less than 10% of the decoding tokens required by comparable methods (e.g., Forest-of-Thought); and a data-efficient, robust domain-adaptation method—Exploration-Focused SFT (EF-SFT)—that, with only 3.5% of the tokens used by standard SFT, improves medical benchmark performance by up to 14.5% over standard SFT.

2 RELATED WORK

In this work, we argue that explicitly decoupling these two functions is crucial for advancing reasoning in large language models. We introduce the **Explore–Execute Chain (E²C)**, a framework that decomposes standard CoT into two distinct phases. The first phase is a highly informative **exploration** stage, in which the model generates a concise, high-level plan. This stage provides a quick preview of the complete reasoning process—analogous to hierarchical planning (Gui et al., 2025)—without incurring the cost of full-chain generation. The second phase is a highly deterministic **execution** stage, which takes the plan as guidance and meticulously performs the detailed

108 calculations. This stage emphasizes precision and faithful adherence to the chosen strategy, a requirement that necessitates specialized training (Zheng et al., 2025b).
 109

110 **From Chain-of-Thought to Structured Reasoning:** Chain-of-Thought (CoT) prompting (Wei
 111 et al., 2022) significantly improves LLM reasoning, but its linear nature has motivated more
 112 robust structured paradigms that explore diverse reasoning paths (Chen et al., 2025). These include
 113 parallel sampling methods such as Self-Consistency (Wang et al., 2022; Wan et al., 2025), and more
 114 complex search structures including trees (ToT) (Yao et al., 2023), graphs (GoT) (Besta et al., 2023;
 115 Yao et al., 2024), and forests (FoT) (Bi et al., 2025). Further advances involve RL-trained parallel
 116 thinking (Zheng et al., 2025b; Pan et al., 2025; Yang et al., 2025b) and hierarchical decomposition
 117 via hypertrees (Gui et al., 2025). While these paradigms expand the search space—often integrating
 118 algorithms like MCTS (Zhang et al., 2024; Xie et al., 2024)—they often conflate high-level planning
 119 with low-level execution. E^2C addresses this limitation through explicit decoupling.

120 **Planning and Decomposition in LLM Reasoning:** The core idea of separating planning from ex-
 121 ecution in E^2C aligns with a growing body of work on task decomposition. Methods range from
 122 breaking problems into subtasks (Zhou et al., 2023; Press et al., 2022) to interleaving reasoning
 123 with tool use (Yao et al., 2022; Schick et al., 2023; Patil et al., 2023). Hu et al. (2025) leveraged
 124 learned belief states to improve planning. Wang et al. (2024a) introduce *PlanSearch* to enhance per-
 125 formance in code generation tasks. While many approaches rely on LLMs as planners for external
 126 solvers (Hao et al., 2023; Liu et al., 2023) or within multi-agent systems (Yuan et al., 2024), E^2C in-
 127 herently supports explore–execute reasoning, yielding greater stability during inference. Moreover,
 128 by exploiting this decomposition property in training, E^2C achieves superior performance.

129 **Test-Time Scaling and Reasoning Efficiency:** Test-time scaling (TTS) aims to improve perfor-
 130 mance by increasing inference-time compute (Snell et al., 2024; Wu et al., 2025), but methods like
 131 Self-Consistency (Wang et al., 2022) are costly because they generate multiple full-length solutions.
 132 This has spurred research on reasoning efficiency, including CoT compression via step entropy (Li
 133 et al., 2025) or truncation (Liao et al., 2025), and adaptive termination guided by semantic entropy to
 134 avoid redundant computation (Xu et al., 2025b). Other efficiency-driven directions include entropy-
 135 guided RL exploration (Zheng et al., 2025a) and reasoning in a continuous latent space (Geiping
 136 et al., 2025; Xu et al., 2025a; Hao et al., 2024). Training these capabilities via Reinforcement
 137 Learning from Verifiable Rewards (RLVR) has also become a key area (Guo et al., 2025; Yue et al.,
 138 2025; Yu et al., 2025; Shao et al., 2025). E^2C contributes a novel TTS strategy: it samples multi-
 139 ple inexpensive plans and executes only the most promising one, thereby achieving ensembling-like
 140 gains at a fraction of the traditional cost.

141 3 METHODOLOGY

142 We introduce the **Explore-Execute Chain (E^2C)** framework, which decomposes reasoning tasks
 143 into two phases: Exploration and Execution. This division aims to improve reasoning efficiency,
 144 scalability, and interpretability by separating brainstorming steps from detailed calculations. As
 145 shown in Fig. 2, we first introduce a two-stage training procedure to achieve a paradigm shift and
 146 performance boost for E^2C model, then we present efficient fine-tuning for specific domains and
 147 effective test-time scaling.

148 3.1 FORMAL DEFINITION OF E^2C

149 The E^2C formalizes reasoning by splitting the coupled reasoning process into two conditional dis-
 150 tributions:
 151

$$\underbrace{p(e \mid c)}_{\text{Coupled Reasoning Process}} \rightarrow \underbrace{p'(\pi, e \mid c)}_{\text{Explore-Execute Chain}} = \underbrace{p'(\pi \mid c)}_{\text{Highly Informative}} \cdot \underbrace{p'(e \mid \pi, c)}_{\text{Highly Deterministic}} \quad (1)$$

152 The framework is defined by two core properties:
 153

1. **(Informative Property).** $p'(\pi \mid c)$ should be highly informative, containing the critical infor-
 154 mation necessary to solve the problem.
2. **(Deterministic Property).** $p'(e \mid \pi, c)$ should be highly deterministic, meaning it must fully
 155 leverage the informative π .

156 Naturally, we semantically design π to represent high-level strategies, while e entails detailed cal-
 157 culations that follow π .

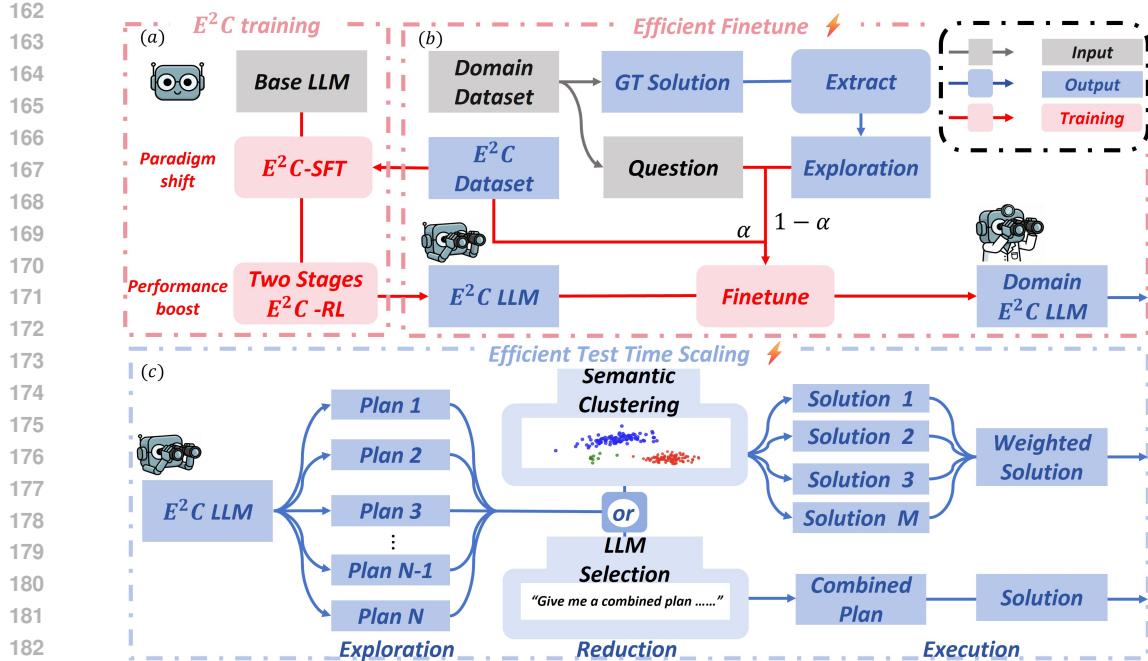


Figure 2: **Overview of E²C method.** The approach begins with E²C-SFT to achieve a paradigm shift, followed by a two-stage E²C-RL process that leverages the decomposition advantage of the new paradigm to boost performance. The resulting E²C-LLM can be efficiently adapted to new domains via EF-SFT. The exploration stage’s high informativeness enables effective test-time scaling, implementable through semantic clustering or LLM selection.

3.2 2-STAGE TRAINING PROCEDURE: SFT AND RL

We introduce a two-stage training procedure to achieve the proposed Prop. 1 and Prop. 2. Stage 1 is Supervised Fine-Tuning (SFT), in which we construct a synthetic dataset and perform SFT to achieve a paradigm shift in reasoning and satisfy the informative Prop. 1. We do not rely solely on prompting to accomplish this paradigm transition because prompting is unstable and leads to a more significant performance drop compared to SFT training. Detailed results are presented in Tab. 1. Stage 2 employs Reinforcement Learning (RL), which incorporates a λ -coefficient on the advantage to appropriately leverage Prop. 1, thereby accelerating convergence and enhancing the determinism of execution to satisfy Prop. 2.

3.2.1 STAGE 1: SYNTHETIC DATASET CONSTRUCTION AND E²C-SFT

To support structured reasoning, we construct a dedicated SFT dataset through synthetic generation. A naive method is to first sample an execution trace from the base model and then summarize it into an exploration step. However, this approach is flawed: the execution is generated from $p(e | c)$ rather than the desired $p'(e | \pi, c)$, effectively hacking the causal structure. As a result, the model learns to ignore the exploration and directly mimic the base model’s execution distribution, violating the intended information bottleneck.

Our method, described in Algorithm. 2, explicitly conditions the execution on the exploration. For each question, we first generate a full solution, distill it into an exploration step, and then prompt the model to produce a new execution which strictly follows the exploration. This enforces a causal dependency from exploration to execution, which is crucial for Prop. 2. The solution can also come from the ground truth. To enable a fair comparison and minimize dataset selection constraints while avoiding the introduction of extra variables, we specifically use samples from the Base LLM in our comparison experiments

3.2.2 STAGE 2: E²C REINFORCEMENT LEARNING (E²C-RL)

To emphasize informative reasoning, we extend hierarchical weighting (Wang et al., 2025) by assigning a higher coefficient λ to exploration tokens, which accelerates convergence (Prop. 1), while

216 the entropy-reduction effect of reinforcement learning supports determinism (Prop. 2). The training
 217 objective is defined as
 218

$$219 \quad \mathcal{L}_{\text{clip}} = \frac{1}{G} \sum_{i,t} \min \left(r_{i,t} \lambda_{i,t} \hat{A}_{i,t}, \text{clip}(r_{i,t}, 1 - \varepsilon, 1 + \varepsilon) \lambda_{i,t} \hat{A}_{i,t} \right). \quad (2)$$

$$223 \quad \mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}[\mathcal{L}_{\text{clip}}] - \beta D_{\text{KL}}[\pi_\theta \| \pi_{\text{ref}}]. \quad (3)$$

225 where i indexes the rollout in the batch, and t indexes the token within that rollout, $\hat{A}_{i,t} = (r_{i,t} -$
 226 $\bar{r}_i)/\sigma_i$ and $r_{i,t} = r_{\text{answer}} + r_{\text{format}}$. The reward r_{answer} measures answer correctness, while r_{format}
 227 consists of a length reward (r_{length}) designed to prevent overly long and repetitive answers and an
 228 instruction reward (r_{instr}), quantifies the alignment between exploration and execution, ensuring that
 229 exploration trajectories approximate optimal execution strategies. The detailed description for r_{format}
 230 can be found in Appendix A.2.1.

231 We adopt a two-stage training procedure. In the first stage, a higher temperature τ_1 and larger rollout
 232 number k_1 are used for one epoch, encouraging broad exploration of the action space and fostering
 233 self-correction to mitigate the overly rigid adherence to the exploration plan that results. In the
 234 second stage, we reduce the temperature to τ_2 and the rollout number to k_2 , again for one epoch,
 235 and assign the advantage coefficient $\lambda_{i,t} = \lambda_{\text{exp}} > 1$ for the exploration tokens in the GRPO update.
 236 This modification explicitly prioritizes high-level reasoning in the policy gradient, thereby achieving
 237 faster and more stable convergence.

238 The behavior of the trained agent can be formalized by analyzing the modified GRPO objective in
 239 Eq. (3). We highlight the following quantified properties:

240 **1. Update emphasis: exploration vs. execution.** Let T_{exp} and T_{exe} be the token index
 241 sets for *exploration* and *execution*, defined by the tokens before and after the special delimiter
 242 `</EXPLORATION>` within an output $O_i = (o_{i,1}, \dots, o_{i,|O_i|})$. The per-token policy gradient is
 243

$$244 \quad g_{i,t} \approx \lambda_{i,t} \hat{A}_{i,t} \nabla_\theta \log \pi_\theta(o_{i,t} | q, o_{i,<t}). \quad (4)$$

246 If $\lambda_{i,t} = \lambda_{\text{exp}} > 1$ for $t \in T_{\text{exp}}$ and $\lambda_{i,t} = \lambda_{\text{exe}} = 1$ for $t \in T_{\text{exe}}$, then

$$248 \quad \frac{\mathbb{E}[\|g_{i,t}\|^2 | t \in T_{\text{exp}}]}{\mathbb{E}[\|g_{i,t}\|^2 | t \in T_{\text{exe}}]} \gtrsim \lambda_{\text{exp}}^2, \quad (5)$$

250 so exploration tokens receive significantly larger expected updates, strengthening the planning
 251 phase. The entropy dynamics are provided in Appendix A.5, which demonstrates that λ_{exp} indeed
 252 leads to a substantial difference.

254 **2. Deterministic execution.** Let $o_{i,t}^* = \arg \max_o \pi_\theta(o | q, o_{i,<t})$ and define the confidence margin
 255

$$256 \quad \Delta_{i,t} := \pi_\theta(o_{i,t}^* | q, o_{i,<t}) - \max_{o \neq o_{i,t}^*} \pi_\theta(o | q, o_{i,<t}). \quad (6)$$

258 Stage-2 RL (with lower temperature and fewer rollouts) increases

$$260 \quad \mathbb{E}_{t \in T_{\text{exe}}} [\Delta_{i,t}] \nearrow, \quad \mathbb{E}_{t \in T_{\text{exe}}} [H(\pi_\theta(\cdot | q, o_{i,<t}))] \searrow, \quad (7)$$

261 where $H(\pi_\theta(\cdot | q, o_{i,<t})) := -\sum_o \pi_\theta(o | q, o_{i,<t}) \log \pi_\theta(o | q, o_{i,<t})$ is the entropy of the token
 262 distribution at step t . This indicates that the execution stage becomes increasingly deterministic, with
 263 high-confidence token choices and low-variance outputs, yielding faithful and stable execution.

264 **3. Plan sensitivity.** Let $\hat{A}_{i,t} = \hat{A}_{i,t}^{\text{plan}}$ for $t \in T_{\text{exp}}$ be the advantage attributed to exploration tokens.
 265 Then the expected update sign satisfies

$$267 \quad \mathbb{E}[\text{sgn}(g_{i,t}) | t \in T_{\text{exp}}] \propto \text{sgn}(\mathbb{E}[\hat{A}_{i,t}^{\text{plan}}]), \quad (8)$$

269 so high-quality plans are amplified while poor plans are suppressed.

270
271**Algorithm 1** E²C Test Time Scaling

```

272 1: Sample  $K$  exploration segments:  $\{e_1, e_2, \dots, e_K\}$ 
273 2: Encode explorations to get embeddings:  $V \leftarrow \{\text{Enc}(e_1), \text{Enc}(e_2), \dots, \text{Enc}(e_K)\}$ 
274 3: Aggregate explorations via either:
275   4:   • Clustering:  $E^* \leftarrow \text{Cluster-Centroids}(V)$ 
276   5:   • LLM fusion:  $E^* \leftarrow \text{LLM-Aggregate}(\{e_1, \dots, e_K\})$ 
277 6: for each aggregated exploration  $e_i^* \in E^*$  do
278   7:   Generate execution:  $a_i \leftarrow \text{Execute}(e_i^*)$ 
279   8:   Assign weight  $w_i$  based on the aggregation method
280 9: end for
281 10: Aggregate answers:  $a_{\text{final}} \leftarrow \sum w_i \cdot \delta(a_i)$ 
282 11: return  $a_{\text{final}}$ 
283
284
```

285 3.3 EFFICIENT ADAPTATION AND INFERENCE WITH E²C286 The modularity of our E²C framework enables efficient strategies for both domain adaptation at
287 training time and scaled aggregation at test time.288 **Exploration-Focused SFT (EF-SFT).** For domain adaptation, we introduce EF-SFT. This method
289 leverages the transferable nature of the execution component by exclusively fine-tuning on the ex-
290 ploration segments from domain-specific examples. These segments are mixed with the base E²C
291 dataset at a controlled ratio α , allowing the model to efficiently learn new reasoning strategies while
292 maintaining its core capabilities. This targeted approach significantly reduces the data and compu-
293 tational requirements for adaptation. A detailed algorithm can be found in the Appendix 3.294 **Think Twice Before Acting: E²C Test Time Scaling.** At inference time, due to the high infor-
295 mativity and short length of the explorations, we can exploit this characteristic to sample a large
296 number of plans. Afterward, using semantic clustering methods or LLMs, we select a smaller subset
297 for execution. Specifically, we introduce two possible implementations for E²C Test time scaling:298 **(1) Clustering-Weighted Voting.** This approach identifies representative reasoning strategies by
299 clustering the sampled M explorations into N clusters. Semantic similarity is measured by the cosine
300 distance between their sentence embeddings, which are obtained from a pre-trained encoder. Only
301 the centroid exploration from each distinct cluster proceeds to the execution phase. The final answers
302 are aggregated using a majority vote, where the weight of each answer is proportional to its cluster
303 size, significantly reducing redundant computations. **(2) LLM-Based Aggregation.** Alternatively, a
304 powerful external LLM can be employed to synthesize the sampled explorations into a single, refined
305 reasoning plan. This method consolidates key insights from multiple paths into a comprehensive
306 exploration, which then guides a single, high-quality execution.

307 4 EXPERIMENTS AND RESULTS

308 In this section, we describe the experimental setup of the mathematical reasoning experiment, the
309 medical reasoning experiment, and the test-time scaling experiments. Each experiment was carried
310 out on a single node with 8 H800 GPUs.

311 4.1 TRAINING PROTOCOLS

312 We adapt our training codebase from verl (Sheng et al., 2024) and perform SFT and RL training.
313 Our training procedures were as follows. The initial E²C-SFT model was trained for one epoch
314 on a 50k-sample synthetic dataset constructed from Openr1-math (deepseek, 2025) using our causal
315 data generation algorithm (Algorithm. 2). This model was then further trained using our two-stage
316 E²C-RL algorithm on the DAPO-17K (Yu et al., 2025) dataset. For comparison, a baseline model
317 was trained with the standard GRPO algorithm for five epochs on the same DAPO-17K data. In our
318 domain adaptation experiments on the ReasonMed dataset, we compared a standard SFT baseline
319 (trained on the full dataset) against our proposed EF-SFT method, which was trained on a targeted
320 50k-sample subset focused only on exploration plans, mixed with 10% regularization data.

324 4.2 EXPERIMENTS
325

326 **Mathematical Reasoning Experiment** We evaluated our E^2C framework on a comprehensive suite
327 of challenging mathematical reasoning benchmarks, including AIME’24, AIME’25, MATH500, the
328 algebra subset of MATH (Hendrycks et al., 2024), Minerva, AMC23 and Olympiad bench (He et al.,
329 2024). Our proposed E^2C -**(SFT+RL)** models were benchmarked against strong GRPO baselines
330 and various ablations, with performance measured by Pass@1 accuracy averaged over 8 samples.
331 The results demonstrate the effectiveness of our approach; for instance, on the AIME’24 benchmark,
332 the Qwen3-4B model trained with our method achieved an accuracy of 37.5%, a significant
333 improvement of 8.7 percentage points over the GRPO baseline.

334 **Medical Reasoning Experiment** To assess cross-domain generalization and data-efficient
335 adaptation, we tested our framework on eight medical reasoning benchmarks, including
336 MedQA (Jin et al., 2021), MedMCQA (Pal et al., 2022), and six MMLU (Hendrycks et al., 2020)
337 subsets. We first evaluated the zero-shot transfer performance of our math-trained RL models. More
338 critically, we compared our **EF-SFT** adaptation strategy against a standard SFT baseline. The results
339 highlight the efficiency of E^2C structure: EF-SFT improved the average accuracy of the Qwen3-8B
340 model by 4.0 percentage points over standard SFT, while using only 10M tokens for training—less
341 than 4% of the 286M tokens required by the baseline.

342 **Test-Time Scaling Experiment** A core advantage of E^2C framework is its ability to facilitate highly
343 efficient test-time scaling. We validate this superior performance-cost trade-off on the challenging
344 AIME’2024 benchmark by comparing our methods against strong baselines, including Self-
345 Consistency (SC) (Wang et al., 2022), Tree-of-Thoughts (ToT) (Yao et al., 2023), and the more
346 advanced Forest-of-Thought (FoT) (Bi et al., 2025).

347 We evaluate two primary variants of our E^2C framework, which first sample K inexpensive explo-
348 ration plans before committing to execution:

- 349 • **E^2C -Select (Self LM-Judge):** Uses the model itself as a judge to select the most promising plan
350 among the K samples for a single execution.
- 351 • **E^2C -Select (Semantic Cluster):** A lighter-weight alternative that embeds the K plans, groups
352 them using semantic clustering to identify representative reasoning strategies, and executes only
353 the centroid plan from each cluster. Final answers are aggregated via a weighted majority vote
354 based on cluster size.

355 To validate our design choices, we include two ablations: **E^2C -SC (Self-Consistency)**, which exe-
356 cutes all K sampled plans and aggregates the final answers via majority voting to serve as a high-
357 cost performance upper bound, and **E^2C -RP** (executes one randomly selected plan). All methods
358 are evaluated on the Qwen3-8B+ E^2C model across four increasing computational budget levels (K
359 or $N = 4, 8, 16, 32$).

360 4.3 RESULTS
361

362 We demonstrate our framework’s reasoning capabilities in mathematical experiments, where our
363 training process fully realizes its structural benefits. In medical reasoning, we show that the frame-
364 work has stronger zero-shot generalization and validate our efficient EF-SFT method. Finally, our
365 test-time analysis confirms that the E^2C framework maintains top performance while significantly
366 reducing computational costs.

367 **Mathematical Reasoning Benchmark Results** We conduct a sanity check comparing our E^2C
368 models (Qwen3-8B/4B+ E^2C -**(SFT+RL)**) against GRPO baselines, as shown in Tab. 1. Our ap-
369 proach outperforms the baselines by 1.5% (8B) and 1.9% (4B), validating the effectiveness of the
370 decomposition strategy. Notably, while paradigm shifts typically risk performance degradation, our
371 method successfully maintains and enhances model capability through careful training design. The
372 full E^2C framework ultimately surpasses the GRPO baseline by leveraging the decomposed struc-
373 ture, establishing a solid foundation for efficient test-time scaling.

374 Ablation studies in Tab. 1 reveal that E^2C -RL provides significant gains over E^2C -SFT+GRPO,
375 with improvements of 3.8% (8B) and 3.2% (4B) on average accuracy, demonstrating that E^2C -RL
376 effectively exploits the decomposition advantage. Furthermore, E^2C -SFT slightly outperforms the
377 prompt-based baseline (Prompt-8B), confirming that structured training is essential for realizing the
378 benefits of E^2C paradigm.

378
379
380
381 Table 1: Performance comparison of Qwen3 models (non-thinking mode) on mathematical reasoning
382 benchmarks. All results are reported as Pass@1 accuracy, with an 8-sample average.
383
384

Model	AIME'24	AIME'25	MATH500	Algebra	Minerva	AMC23	Olympiad	Avg Acc	Avg Length
<i>Qwen3 8B Series</i>									
Qwen3-8B+GRPO (Baseline)	36.9	34.4	88.2	88.2	33.1	79.3	60.0	60.0	1429.46
Qwen3-8B+E ² C-SFT+GRPO	37.5	32.5	83.5	86.6	30.8	76.3	56.8	57.7	1309.62
Qwen3-8B+E²C-(SFT+RL)	40.6	33.8	87.7	90.9	35.8	80.3	61.3	61.5	1476.41
<i>Qwen3 4B Series</i>									
Qwen3-4B+GRPO (Baseline)	28.8	30.6	84.6	84.4	33.5	75.8	57.8	56.5	1263.15
Qwen3-4B+E ² C-SFT+GRPO	28.8	26.9	85.9	83.3	33.2	75.7	55.3	55.2	1324.18
Qwen3-4B+E²C-(SFT+RL)	37.5	30.0	86.1	84.8	34.0	78.3	58.4	58.4	1456.34
<i>Ablation Studies</i>									
Qwen3-8B+Prompt (Zero-shot)	21.9	18.8	76.3	80.5	30.9	50.7	45.8	46.6	1142.38
Qwen3-8B+E ² C-SFT	23.1	21.9	75.8	80.5	31.5	51.5	43.2	46.8	1162.89

390
391 Table 2: Performance Comparison of Models with Different Training Processes: Our inference
392 paradigm demonstrates superior generalization, while EF-SFT shows improved efficiency and ro-
393 bustness. The six columns from Anatomy (AN), Clinical Knowledge (CK), College Biology (CB),
394 College Medicine (CM), Medical Genetics (MG), and Professional Medicine (PM) are validation
395 subsets of the MMLU benchmark.

Model	MedQA	MedMCQA	AN	CK	CB	CM	MG	PM	#Med-Tokens	Avg
<i>External Baselines</i>										
HuatuoGPT-01-7B	68.4	57.5	71.9	78.5	88.2	67.6	80.0	77.6	-	73.7
Baichuan-M1-14B	76.5	65.2	77.3	83.6	87.9	80.7	89.1	88.8	-	81.1
ReasonMed-7B	66.9	65.1	75.6	79.3	79.2	73.4	85.0	80.9	-	75.7
<i>Our Method and Ablations</i>										
Qwen3-8B	71.4	59.5	68.0	81.6	87.5	78.0	85.0	83.8	-	76.8
Qwen3-8B + GRPO	74.0	60.6	75.0	80.9	91.3	81.8	90.4	86.2	-	79.1
Qwen3-8B+E²C-(SFT+RL)	74.5	63.1	77.0	82.2	92.0	83.0	92.8	86.0	-	81.1
<i>SFT Models (using medical data)</i>										
Qwen3-8B + standard SFT	58.2	52.3	68.8	80.8	89.0	73.7	83.3	79.0	286M	73.1
Qwen3-8B + E²C-SFT + EF-SFT	65.8	58.2	72.3	83.8	89.2	79.7	87.6	86.2	10M	77.1
Llama3.1-8B + ReasonMed SFT	42.0	36.8	45.9	55.4	61.8	43.2	38.1	56.9	286M	47.5
Llama3.1-8B + E²C-SFT + EF-SFT	60.3	53.2	61.8	69.8	75.9	64.9	82.0	72.5	10M	67.5

407
408
409
410 **Medical Reasoning Benchmark Results** Tab. 2 presents the medical reasoning performance
411 across three experimental settings. First, we establish competitive baselines by comparing against
412 leading domain-specific 7B-8B models (HuatuoGPT-01-7B (Wang et al., 2024b), ReasonMed-
413 7B (Sun et al., 2025)) and an open-source 14B medical LLM (Baichuan-M1-14B (Bingning Wang
414 et al., 2025)), with Qwen3-8B (Yang et al., 2025a) serving as our base model reference.415 For domain adaptation, we evaluate our EF-SFT approach (Sec. 3.3) against standard SFT on both
416 Llama3.1-8B (Dubey et al., 2024) and Qwen3-8B architectures. As shown in Tab. 2, EF-SFT
417 achieves significant improvements of 3.9% (Qwen3-8B) and 14.5% (Llama3.1-8B) over standard
418 SFT, while using only 3.5% of the training tokens. The zero-shot transfer results further demonstrate
419 that our mathematically-trained RL models attain performance comparable to specialized medical
420 LLMs, validating the strong cross-domain generalization capability of our method.421 **Test-Time Scaling Performance and Efficiency Analysis** Tab. 3 demonstrates that E²C framework
422 offers a superior performance-cost trade-off. Our primary method, **E²C-Select (Self LM-Judge)**,
423 achieves a state-of-the-art 58.1% accuracy at the highest budget (K=32), surpassing baselines like
424 Self-Consistency (54.2%). More strikingly, it reaches this performance using only **12.1k tokens**—a
425 fraction of the cost of SC (81.6k) and FoT (128.8k). Our **E²C-Select (Semantic Cluster)** variant
426 provides an alternative trade-off. By executing the centroid of each of the main plan clusters (3 on
427 average), it results in competitive accuracy. While its token cost is higher due to multiple executions,
428 it remains significantly more efficient than baselines like ToT or the **E²C-SC (Self-Consistency)**
429 ablation. The high cost of E²C-SC ablation validates our selective execution strategy, while the poor
430 performance of **E²C-RP (Random Plan)** underscores the necessity of an intelligent (non-random)
431 plan selection mechanism. In summary, by efficiently scaling the inexpensive exploration phase, our
432 framework provides a spectrum of strategies that unlock significant performance gains at a fraction
433 of the computational cost of traditional methods.

432 Table 3: Test-Time Scaling Performance on AIME’2024 Benchmark with Qwen3-8B. We compare
 433 Pass@1 accuracy against the average number of generated tokens per question, demonstrating the
 434 superior performance-cost trade-off of E²C framework.

Method	Budget Level 1		Budget Level 2		Budget Level 3		Budget Level 4	
	Acc. (%)	Tokens (k)	Acc. (%)	Tokens (k)	Acc. (%)	Tokens (k)	Acc. (%)	Tokens (k)
<i>Standard Methods</i>								
Greedy CoT ($N = 1$)	40.6	2.6	47.5 (N=8)	20.4	<i>(Same as Budget Level 1)</i>		54.2 (N=32)	81.6
Self-Consistency	43.9 (N=4)	10.2			51.1 (N=16)	40.8		
<i>Advanced Search Methods</i>								
Tree-of-Thoughts (ToT)	46.1 (N=4)	14.5	50.2 (N=8)	29.0	53.8 (N=16)	58.0	56.3 (N=32)	116.0
Forest-of-Thought (FoT)	48.5 (N=4)	16.1	52.5 (N=8)	32.2	55.8 (N=16)	64.4	58.0 (N=32)	128.8
<i>Our Methods</i>								
E ² C-Select (Self LM-Judge)	49.5 (K=4)	6.1	53.2 (K=8)	7.3	56.4 (K=16)	9.7	58.1 (K=32)	12.1
E ² C-Select (Semantic Cluster)	47.8 (K=4)	11.3	51.5 (K=8)	14.8	54.2 (K=16)	21.8	55.9 (K=32)	35.8
<i>Ablations</i>								
E ² C-SC (Self-Consistency)	50.1 (K=4)	22.6	54.0 (K=8)	45.2	56.9 (K=16)	90.4	58.9 (K=32)	180.8
E ² C-RP (Random Plan)	43.2 (K=4)	6.1	44.5 (K=8)	7.3	45.1 (K=16)	9.7	45.8 (K=32)	12.1

448 **Ablations and Analysis** Our ablation
 449 studies validate our key de-
 450 sign choices. As shown in Part A
 451 of Tab. 4, our causal data genera-
 452 tion strategy (Algorithm 1) is es-
 453 sential, achieving near-perfect plan ad-
 454 herence (0.998) that is critical for
 455 E²C paradigm. Part B demonstrates
 456 the framework’s efficiency in domain
 457 adaptation; performance on medi-
 458 cal benchmarks peaks after a brief
 459 training period (300 iterations, nearly
 460 5k samples) and declines thereafter,
 461 highlighting the data-efficient nature
 462 of fine-tuning only the exploration
 463 phase. Part C shows that incor-
 464 porating a small proportion of reg-
 465 ularization data ($\alpha = 10\%$) is su-
 466 perior to both using no regularization
 467 ($\alpha = 0\%$) and training on the
 468 full exploration-execution sequence
 469 ($\alpha = 100\%$), highlighting the effi-
 470 ciency and robustness derived from
 471 the exploration-focused approach. Additionally, it suggests that using regularization data from the
 472 base E²C-SFT dataset (i.e., using Math as Regularization) is more effective than using domain-
 473 specific medical data for regularization, indicating that there is no need to generate regularization
 474 data for the specific target domain.

5 LIMITATIONS AND FUTURE WORK

475 The E²C framework, while demon-
 476 strating advanced reasoning capabili-
 477 ties in supporting long-chain reason-
 478 ing models such as gpt-o1 (OpenAI,
 479 2024) and deepseek-r1
 (Guo et al., 2025) due to archi-
 480 tectural differences. To address this,
 481 we plan to develop multi-round
 482 exploration and execution mech-
 483 anisms that enable iterative refine-
 484 ment and more effective decom-
 485 position of complex, long-horizon tasks.

486 At the same time, the decoupled nature of E²C offers unique advantages for human-AI collaboration.
 487 The exploration phase provides users with immediate visibility into the model’s reasoning process,
 488 facilitating rapid feedback and collaborative ideation. The execution phase serves as a transparent
 489 and reliable module that translates high-level plans into actionable results, significantly enhancing
 490 the interpretability, controllability, and usability of the system. We believe these characteristics es-
 491 tablish a foundation for more adaptive and user-centered AI assistants, with strong potential to sup-
 492 port human-in-the-loop applications requiring complex reasoning and interactive decision-making.

447 Table 4: An ablation analysis shows the validity of our
 448 data construction methodology by quantifying plan adher-
 449 ence (top), identifies the optimal training iteration count for
 450 medical domain SFT (middle), and shows the impact of data
 451 mixing (bottom).

Part A: Ablation on SFT Data Construction Strategy					
SFT Data Strategy		Plan-Guided		Plan-Free	
Flawed (Reverse-Causal Summary)		0.499		0.864	
Proposed (Causal Generation)		0.998		1.000	
Part B: Ablation on Exploration-Focused SFT Training Steps					
Training Steps	GSM8K	Anatomy	CK	MedQA	MATH
SFT + 100 iter	83.0	70.4	80.6	66.4	70.5
SFT + 300 iter	83.3	72.3	83.8	66.1	71.2
SFT + 2000 iter	84.0	69.8	78.4	63.5	71.1
Part C: Ablation on Regularization Data					
	GSM8K	Anatomy	CK	MedQA	MATH
Medical Reg. ($\alpha=10\%$)	82.1	74.4	79.5	66.8	69.5
Math Reg. ($\alpha=10\%$)	83.3	72.3	83.8	66.1	71.2
Medical Reg. ($\alpha=100\%$)	80.1	70.0	81.5	67.0	65.8
No Reg. ($\alpha = 0\%$)	82.1	68.2	77.8	63.5	70.8

486 **6 CONCLUSION**
487488 Through the proposed Explore-Execute Chain (E²C), we introduce a novel reasoning framework that
489 decouples exploration from execution, enhancing both efficiency and interpretability. Our two-stage
490 SFT+RL training approach, supported by a dedicated data construction method and token-specific
491 reward scaling, enables faithful plan adherence and robust paradigm transition. The framework ef-
492 ffectively concentrates information in exploration, allowing domain adaptation using only 3.5% of
493 training tokens and achieving a superior performance-cost trade-off on complex reasoning bench-
494 marks compared to strong baselines. This also opens up new avenues for users to interact with
495 reasoning models.
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539

540 ETHICS STATEMENT
541

542 This work studies a reasoning framework, the **Explore-Execute Chain (E²C)**, which separates
543 lightweight exploratory sketches from a final execution step to improve efficiency, transparency, and
544 controllability of LLM reasoning. Our experiments fine-tune and evaluate general-purpose LLMs
545 on publicly available benchmarks (e.g., mathematics and domain reasoning datasets). We do not
546 collect new human data, do not involve human or animal subjects, and do not process personally
547 identifiable or sensitive information. Any third-party datasets used in this paper are publicly released
548 for research purposes by their respective providers; we follow their licenses and usage terms. E²C
549 paradigm increases interpretability by exposing intermediate “exploration” traces, which can facilitate
550 auditing and discourage over-reliance on hidden chain-of-thought. This study complies with the
551 conference’s Code of Ethics.
552

553 REPRODUCIBILITY STATEMENT
554

555 We have made extensive efforts to ensure the reproducibility of our work. All code used in this paper
556 will be publicly released to facilitate independent verification and further research. We describe our
557 experimental setup in Sec. 4.1. Detailed hyperparameters for training E²C-SFT, E²C-RL, GRPO,
558 and EF-SFT are provided in Appendix A.2.1. Detailed setup for TTS experiment can be found in
559 Appendix A.4. We also include the prompt templates for data generation, the zero-shot prompt
560 model, and E²C-Select (Self LM-Judge) in Appendix A.6.
561

562 REFERENCES
563

Anja Achtziger and Peter M Gollwitzer. Rubicon model of action phases. 2007. 15

Maciej Besta, Nils Blach, Ales Kubicek, Kyrylo Robert, Kamil Konopka, Hubert Niewiadomski,
Wojciech Gajda, David Grolimund, Tim Nisa, and Torsten Hoefer. Graph of thoughts: Solving
elaborate problems with large language models. *arXiv preprint arXiv:2308.09687*, 2023. 1, 3

Zhenni Bi, Kai Han, Chuanjian Liu, Yehui Tang, and Yunhe Wang. Forest-of-thought: Scaling
test-time compute for enhancing llm reasoning, 2025. 1, 3, 7, 18

Huozhi Zhou Liang Song Mingyu Xu Wei Cheng Xiangrong Zeng Yupeng Zhang Yuqi Huo Zecheng
Wang Zhengyun Zhao Bingning Wang, Haizhou Zhao et al. Baichuan-m1: Pushing the medical
capability of large language models. *arXiv preprint arXiv:2502.12671*, 2025. 8

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu,
Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-
thought for reasoning in large language models, 2025. 3

deepseek. Open r1: A fully open reproduction of deepseek-r1. <https://github.com/huggingface/open-r1>, January 2025. 6

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha
Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and et al. The llama 3 herd of
models, 2024. 8

Zeyu Gan, Hao Yi, and Yong Liu. CoT-Space: A theoretical framework for internal slow-thinking
via reinforcement learning, 2025. 2

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson,
Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with
latent reasoning: A recurrent depth approach, 2025. 2, 3

Runquan Gui, Zhihai Wang, Jie Wang, Defu Lian, Chi Ma, Huiling Zhen, Mingxuan Yuan, Jianye
Hao, Enhong Chen, and Feng Wu. Hypertree planning: Enhancing llm reasoning via hierarchical
thinking, 2025. 1, 2, 3

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu,
Shirong Ma, Peiyi Wang, Xiao Bi, and et al. DeepSeek-R1: Incentivizing reasoning capability in
llms via reinforcement learning, 2025. 3, 9

594 Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning
 595 with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023. 3
 596

597 Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong
 598 Tian. Training large language models to reason in a continuous latent space, 2024. 3
 599

600 Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu,
 601 Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for
 602 promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint*
 603 *arXiv:2402.14008*, 2024. 7

604 Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and
 605 Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint*
 606 *arXiv:2009.03300*, 2020. 7

607 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song,
 608 and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. URL
 609 <https://arxiv.org/abs/2103.03874>, 2, 2024. 7
 610

611 Edward S. Hu, Kwangjun Ahn, Qinghua Liu, Haoran Xu, Manan Tomar, Ada Langford, Dinesh
 612 Jayaraman, Alex Lamb, and John Langford. The belief state transformer. In *The Thirteenth*
 613 *International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=ThRMTcgpvo>. 3
 614

615 Di Jin, Eileen Pan, Nassim Oufattolle, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What dis-
 616 ease does this patient have? a large-scale open domain question answering dataset from medical
 617 exams. *Applied Sciences*, 11(14):6421, 2021. 7
 618

619 Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*,
 620 2014. 16

621 Zeju Li, Jianyuan Zhong, Ziyang Zheng, Xiangyu Wen, Zhijian Xu, Yingying Cheng, Fan Zhang,
 622 and Qiang Xu. Compressing chain-of-thought in llms via step entropy, 2025. 1, 3
 623

624 Baohao Liao, Hanze Dong, Yuhui Xu, Doyen Sahoo, Christof Monz, Junnan Li, and Caiming Xiong.
 625 Fractured chain-of-thought reasoning, 2025. 1, 2, 3
 626

627 Boi-Faltings Liu, Zhang-Wei Liu, Ruibo Jiang, Yisong Lyu, Yizhou Du, F. Wu, and Yu-Feng Liu.
 628 Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint*
 629 *arXiv:2304.11477*, 2023. 3

630 OpenAI. Openai o1 system card. <https://openai.com/index/openai-o1-system-card/>, December 2024. Updated: December 5, 2024. Accessed:
 631 2025-09-24. 9
 632

633 Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale
 634 multi-subject multi-choice dataset for medical domain question answering. In *Conference on*
 635 *health, inference, and learning*, pp. 248–260. PMLR, 2022. 7
 636

637 Jiayi Pan, Xiuyu Li, Long Lian, Charlie Snell, Yifei Zhou, Adam Yala, Trevor Darrell, Kurt Keutzer,
 638 and Alane Suhr. Learning adaptive parallel reasoning with language models, 2025. 3
 639

640 Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. Gorilla: Large language model
 641 connected with massive apis. *arXiv preprint arXiv:2305.15334*, 2023. 3
 642

643 Ofir Press, Or Yoran, Timo Schick, Idan Schmid, Ayal Fisch, Yoav Goldberg, and Kanishka
 644 Misra. Measuring and narrowing the compositionality gap in language models. *arXiv preprint*
 645 *arXiv:2210.03350*, 2022. 3

646 Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea
 647 Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv*
 648 *preprint arXiv:2305.18290*, 2023. 2

648 Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Tsvigun, Gautier Cances,
 649 and Najma Smaili. Toolformer: Language models can teach themselves to use tools. *arXiv*
 650 *preprint arXiv:2302.04761*, 2023. 3

651 Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du,
 652 Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei
 653 Koh, and Luke Zettlemoyer. Spurious rewards: Rethinking training signals in rlvr, 2025. 3

654 Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng,
 655 Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework, 2024. 6

656 Noah Shinn, Beck Labash, and Ashwin Gopinath. Reflexion: Language agents with verbal rein-
 657 force learning. *arXiv preprint arXiv:2303.11366*, 2023. 1

658 Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally
 659 can be more effective than scaling model parameters, 2024. 3

660 Yu Sun, Xingyu Qian, Weiwen Xu, Hao Zhang, Chenghao Xiao, Long Li, Yu Rong, Wenbing
 661 Huang, Qifeng Bai, and Tingyang Xu. Reasonmed: A 370k multi-agent generated dataset for
 662 advancing medical reasoning. *arXiv preprint arXiv:2506.09513*, 2025. 8

663 Guangya Wan, Yuqi Wu, Jie Chen, and Sheng Li. Reasoning aware self-consistency: Leverag-
 664 ing reasoning paths for efficient LLM sampling. In Luis Chiruzzo, Alan Ritter, and Lu Wang
 665 (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the*
666 Association for Computational Linguistics: Human Language Technologies (Volume 1: Long
667 Papers), pp. 3613–3635, Albuquerque, New Mexico, April 2025. Association for Compu-
668 tational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.nacl-long.184. URL
669 <https://aclanthology.org/2025.nacl-long.184/>. 3

670 Evan Wang, Federico Cassano, Catherine Wu, Yunfeng Bai, Will Song, Vaskar Nath, Ziwen Han,
 671 Sean Hendryx, Summer Yue, and Hugh Zhang. Planning in natural language improves llm search
 672 for code generation. *arXiv preprint arXiv:2409.03733*, 2024a. 3

673 Haozhe Wang, Qixin Xu, Che Liu, Junhong Wu, Fangzhen Lin, and Wenhui Chen. Emergent hierar-
 674 chical reasoning in llms through reinforcement learning. *arXiv preprint arXiv:2509.03646*, 2025.
 675 4

676 Junying Wang, Zhaonan Li, Renfeng Pu, Sajiang Shi, Yitong Meng, Zhaokun Wang, Yixin Liu,
 677 Jianing Zhou, Wenjia Zhang, Jialiang Chen, Yefeng Zheng, and Hong-Yin Mey. HuatuoGPT, a
 678 general-purpose chinese medical large language model, 2024b. 8

679 Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdh-
 680 ery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models.
 681 *arXiv preprint arXiv:2203.11171*, 2022. 1, 2, 3, 7

682 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc
 683 Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In
 684 *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022. 1, 3

685 Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws:
 686 An empirical analysis of compute-optimal inference for LLM problem-solving. In *The Thirteenth*
 687 *International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=VNckp7JEHn>. 3

688 Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P Lillicrap, Kenji Kawaguchi,
 689 and Michael Shieh. Monte carlo tree search boosts reasoning via iterative preference learning.
 690 *arXiv preprint arXiv:2405.00451*, 2024. 3

691 Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. Softcot++: Test-time scaling with soft chain-
 692 of-thought reasoning, 2025a. 1, 3

693 Zenan Xu, Zexuan Qiu, Guanhua Huang, Kun Li, Siheng Li, Chenchen Zhang, Kejiao Li, Qi Yi,
 694 Yuhao Jiang, Bo Zhou, Fengzong Lian, and Zhanhui Kang. Adaptive termination for multi-round
 695 parallel reasoning: An universal semantic entropy-guided framework, 2025b. 1, 3

702 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang
 703 Gao, Chengan Huang, Chenxu Lv, and et al. Qwen3 technical report, 2025a. 8
 704

705 Xinyu Yang, Yuwei An, Hongyi Liu, Tianqi Chen, and Beidi Chen. Multiverse: Your language
 706 models secretly decide how to parallelize and merge generation, 2025b. 3

707 Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Ekin Durmus, Cibu L parochial, Linyuan Han, Yisi
 708 Gu, Karthik Annot, K. Josifoski, et al. React: Synergizing reasoning and acting in language
 709 models. *arXiv preprint arXiv:2210.03629*, 2022. 1, 3
 710

711 Shunyu Yao, Dian Astuti, Bo Peng, Danfei Chen, Yuansi Wong, Jean-Francois Simon, C. Voss,
 712 E. Schwartz, and A. Rea. Tree of thoughts: Deliberate problem solving with large language
 713 models. *arXiv preprint arXiv:2305.10601*, 2023. 1, 3, 7, 18

714 Yao Yao, Zuchao Li, and Hai Zhao. GoT: Effective graph-of-thought reasoning in language mod-
 715 els. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for*
716 Computational Linguistics: NAACL 2024, pp. 2901–2921, Mexico City, Mexico, June 2024.
 717 Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.183. URL
 718 <https://aclanthology.org/2024.findings-naacl.183/>. 3
 719

720 Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian
 721 Fan, Gaohong Liu, Lingjun Liu, and et al. DAPO: An open-source llm reinforcement learning
 722 system at scale, 2025. 3, 6

723 Siyuan Yuan, Kairui Song, Jia-Hao Chen, Xiao-Hui Tan, Dian-Hui Li, and Dong-Sheng Yang. Evoa-
 724 gent: Towards automatic multi-agent generation via evolutionary algorithms, 2024. 3
 725

726 Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi
 727 Wang, Tian Tian Fan, Zhengyin Du, and et al. VAPO: Efficient and reliable reinforcement learning
 728 for advanced reasoning tasks, 2025. 3

729 Di Zhang, Xiaoshui Huang, Dongzhan Zhou, Yuqiang Li, and Wanli Ouyang. Accessing gpt-4 level
 730 mathematical olympiad solutions via monte carlo tree self-refine with llama-3 8b, 2024. 3
 731

732 Tianyu Zheng, Tianshun Xing, Qingshui Gu, Taoran Liang, Xingwei Qu, Xin Zhou, Yizhi Li, Zhou-
 733 futu Wen, Chenghua Lin, Wenhao Huang, Qian Liu, Ge Zhang, and Zejun Ma. First return,
 734 entropy-eliciting explore, 2025a. 1, 3

735 Tong Zheng, Hongming Zhang, Wenhao Yu, Xiaoyang Wang, Xinyu Yang, Runpeng Dai, Rui Liu,
 736 Huiwen Bao, Chengsong Huang, Heng Huang, and Dong Yu. Parallel-r1: Towards parallel think-
 737 ing via reinforcement learning, 2025b. 2, 3
 738

739 Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuur-
 740 mans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. Least-to-most prompting enables
 741 complex reasoning in large language models. In *International Conference on Learning Represen-*
 742 *tations*, 2023. 1, 3
 743

744 USE OF LARGE LANGUAGE MODELS

745 We utilized a large language model to enhance the language and clarity of our manuscript. Speci-
 746 fically, we employed Gemini 2.5 flash with the following prompt to refine the initial draft: *I am*
 747 *writing an academic paper in English. Please polish the following draft so that it adheres to the*
 748 *conventions of academic writing.*

751 A APPENDIX

752 A.1 COGNITIVE MODEL ANALYSIS

753 In this section, we analyze cognitive models to derive high-level design insights for our method.

756 A.1.1 RUBICON MODEL OF ACTION PHASES
757758 The **Rubicon Model of Action Phases** (Achtziger & Gollwitzer, 2007), proposed by **Heckhausen**
759 and **Gollwitzer**, provides a framework for how individuals prepare for and pursue goals. It divides
760 goal pursuit into four stages: **goal setting**, **planning**, **action**, and **evaluation**.761 1. **Goal Setting**: Individuals identify and adopt a goal, motivated by a need or desire.
762 2. **Planning**: After a goal is adopted, individuals generate strategies to achieve it and assess
763 their potential effectiveness.
764 3. **Action**: Once a strategy is selected, the individual commits to it and executes it. Crossing
765 the “Rubicon” marks this commitment and the transition to action.
766 4. **Evaluation**: Outcomes are assessed to inform adjustments to subsequent plans or actions.
767768 A key contribution of the Rubicon Model is the sharp distinction between planning and execution.
769 After commitment (crossing the Rubicon), attention is devoted to execution rather than continued
770 exploration or second-guessing. This separation mitigates cognitive overload that could arise from
771 ongoing re-evaluation during task execution.
772773 A.1.2 CONNECTING E²C WITH RUBICON MODEL
774775 We formally express E²C as

776
$$\underbrace{p(e | c)}_{\text{Coupled Reasoning Process}} \rightarrow \underbrace{p'(\pi, e | c)}_{\text{Explore-Execute Chain}} = \underbrace{p'(\pi | c)}_{\text{Highly Informative}} \cdot \underbrace{p'(e | \pi, c)}_{\text{Highly Deterministic}} \quad (9)$$

777

778 **$p'(\pi | c)$ as the Planning Phase**: In the Rubicon framework, planning entails generating candidate
779 strategies. Analogously, in E²C, $p'(\pi | c)$ produces multiple candidate plans π from context c .
780 These plans are highly informative, capturing the critical information needed to solve the task. This
781 exploration corresponds to the goal-setting and planning stages, where alternatives are considered
782 before selection.
783784 **$p'(e | \pi, c)$ as the Execution Phase**: Once plans are available, E²C transitions to execution. The
785 distribution $p'(e | \pi, c)$ reflects a highly deterministic process that follows the selected plan π under
786 context c . This mirrors the action phase of the Rubicon Model: the agent executes the committed
787 plan without revisiting discarded alternatives.
788789 Thus, the separation between $p'(\pi | c)$ and $p'(e | \pi, c)$ in E²C parallels the explore-then-execute
790 dynamics of the Rubicon Model: first enumerate options, then execute deterministically.
791

A.1.3 COGNITIVE AND COMPUTATIONAL EFFICIENCY

792 Separating exploration from execution confers efficiency benefits in both cognition and computation.
793 Cognitively, once commitment occurs, resources are focused on carrying out the chosen plan
794 without distraction from alternatives. Computationally, E²C avoids the overhead of re-evaluating
795 multiple plans during execution. The deterministic execution phase concentrates compute on fol-
796 lowing the selected plan, yielding faster and more reliable performance than continually interleaving
797 exploration with action.
798

A.1.4 INTERPRETABILITY AND TRANSPARENCY

800 The exploration-execution split also improves **interpretability**. In the Rubicon Model, one can
801 explain an action by the plan selected during the planning stage. Likewise, E²C makes the
802 reasoning path explicit: multiple candidate plans are generated (exploration), and one is chosen and
803 followed (execution). This transparency further supports **scalability**: the exploration component
804 can be adapted to new tasks and domains, while the execution component remains stable, enabling
805 flexible and extensible reasoning across settings.
806

A.2 THE DETAILS OF THE EXPERIMENTS

807 In this section, we introduce the details of our main experiments in the main paper for reproducibility
808 purposes, including the detailed hyperparameter settings and the reward designs.
809

810 A.2.1 HYPERPARAMETER SETTINGS
811812 **E²C-SFT and EF-SFT Training** For both E²C-SFT and EF-SFT training, the hyperparameters
813 are summarized in Tab. 5:

Hyperparameter	Value
Learning Rate	1.0×10^{-5}
Optimizer	Adam(Kingma, 2014) ($\beta_1 = 0.9, \beta_2 = 0.95$)
Weight Decay	0.01
Learning Rate Scheduler	Cosine with 10% warmup ratio
Batch Size	160
Micro-batch Size per GPU	20
Gradient Clipping	1.0
Total Epochs	1

823 Table 5: Hyperparameters for E²C-SFT and EF-SFT Training
824
825826 **E²C-RL and GRPO Training** The hyperparameters for E²C-RL and GRPO training are summa-
827 rized in Tab. 6, where the experiments include E²C Stage 1 (E2C-stg1), E²C Stage 2 (E2C-stg2),
828 and GRPO:
829

Hyperparameter	E ² C-stage1	E ² C-stage2	GRPO
Batch Size	256	256	128
Overlong Buffer Length	4096	4096	4096
Maximum Response Length	8192	8192	8192
Learning Rate	1.0×10^{-6}	1.0×10^{-6}	1.0×10^{-6}
Mini-batch Size for GRPO Updates	32	32	32
KL Loss Coefficient β	0.001	0	0
Rollout Number k	32	8	8
Temperature	1.3	1.0	1.0
Training Epochs	1	1	5
Clip ratio (ε)	0.2	0.2	0.2

841 Table 6: Hyperparameters for E²C-RL and GRPO Training
842
843

844 A.2.2 REWARD DETAILS FOR RL TRAINING

845 **Format Reward Calculation for E²C Training** For the E²C training, the format reward consists
846 of two components: the length reward and the instruction reward. These rewards are computed as
847 follows:848 **Length Reward:** This reward measures how well the output length matches the expected length. It
849 is computed as:

851
852
$$r_l = -\text{clip} \left(0, 1, \frac{L - L_{\text{valid}}}{L_{\text{buffer}}} \right)$$

853

854 where: L is the length of the generated output; L_{valid} is the length of the valid portion of the
855 response; L_{buffer} is the overlong buffer length.
856857 **Instruction Reward:** The instruction reward is specific to the E²C model and is added to the reward
858 function when it comes to E²C model. This reward measures the alignment between the instructions
859 generated during the exploration phase and the execution phase. It is computed by extracting the
860 step titles from both the exploration and execution phases using regular expressions. Denote these
861 sets of instructions as S_1 (exploration) and S_2 (execution). The instruction reward is defined as:

862
863
$$r_{\text{instr}} = 0.1 * \left(\frac{|S_1 \cap S_2|}{\max(|S_1|, |S_2|)} - 1 \right)$$

864 where: S_1 is the set of instructions generated during the exploration phase; S_2 is the set of in-
 865 structions generated during the execution phase; $|S_1 \cap S_2|$ is the intersection of the sets S_1 and S_2 ;
 866 $\max(|S_1|, |S_2|)$ is the maximum size of the two sets.

867 The instruction reward incentivizes the model to generate instructions that align well between the
 868 exploration and execution phases, encouraging consistency. This reward is crucial for $\mathbf{E^2C}$ models
 869 to ensure that the reasoning process is coherent between the exploration and execution stages.

870
 871 **Format Reward Calculation for GRPO Training** For **GRPO** training, the format reward is sim-
 872 pler and consists solely of the length reward, which is computed using the same formula as in $\mathbf{E^2C}$:

$$873 \\ 874 \\ 875 \quad r_l = -\text{clip} \left(0, 1, \frac{\text{length}_{\text{output}} - \text{valid}_{\text{length}}}{\text{buffer}_{\text{length}}} \right) \\ 876 \\ 877$$

878 In GRPO, no instruction reward is applied, and the focus is entirely on the length of the response,
 879 ensuring that the output adheres to the expected length constraints.

880 A.3 DETAILS OF THE ALGORITHM

881 **Algorithm of $\mathbf{E^2C}$ -SFT Data Generation** Algorithm 2 is a formal and detailed description for
 882 $\mathbf{E^2C}$ -SFT Data Generation.

883 **Algorithm 2** $\mathbf{E^2C}$ -SFT Data Generation

884
 885 1: $\mathcal{D}_{\text{synth}} \leftarrow \emptyset$
 886 2: **for** each question q **do**
 887 3: solution $\leftarrow \text{Model}_{\text{base}}(q)$
 888 4: exploration $\leftarrow \text{Summarize}(\text{solution})$
 889 5: prompt \leftarrow “Given question: q . Follow exploration: exploration. Execute step-by-step:”
 890 6: execution $\leftarrow \text{Model}_{\text{base}}(\text{prompt})$
 891 7: $\mathcal{D}_{\text{synth}} \leftarrow \mathcal{D}_{\text{synth}} \cup \{(q, (\text{exploration}, \text{execution}))\}$
 892 8: **end for**
 893 9: **return** $\mathcal{D}_{\text{synth}}$

894
 895
 896
 897
 898 **Algorithm of Exploration-Focused SFT (EF-SFT) Data Generation** Algorithm 3 is a formal
 899 and detailed description for EF-SFT Data Generation.

900 **Algorithm 3** EF-SFT Data Generation

901 **Require:** Base $\mathbf{E^2C}$ dataset $\mathcal{D}_{\text{base}}$
 902 **Require:** Domain-specific dataset $\mathcal{D}_{\text{domain}}$
 903 **Require:** Mixing ratio $\alpha \in [0, 1]$, Target Dataset size n_{target}
 904 **Ensure:** EF-SFT training dataset $\mathcal{D}_{\text{EF-SFT}}$

905
 906 1: $\mathcal{D}_{\text{explore}} \leftarrow \emptyset$
 907 2: **for** each example $(q, a) \in \mathcal{D}_{\text{domain}}$ **do**
 908 3: Extract exploration segment: $e \leftarrow \text{ExtractExploration}(a)$
 909 4: $\mathcal{D}_{\text{explore}} \leftarrow \mathcal{D}_{\text{explore}} \cup \{(q, e)\}$
 910 5: **end for**

911 6: $n_{\text{base}} \leftarrow \alpha \times n_{\text{target}}$ $\triangleright \alpha\%$ from base dataset
 912 7: $n_{\text{explore}} \leftarrow (1 - \alpha) \times n_{\text{target}}$ $\triangleright (1 - \alpha)\%$ from exploration data

913 8: $\mathcal{D}_{\text{base}}^{\text{sub}} \leftarrow \text{Subsample}(\mathcal{D}_{\text{base}}, n_{\text{base}})$
 914 9: $\mathcal{D}_{\text{explore}}^{\text{sub}} \leftarrow \text{Subsample}(\mathcal{D}_{\text{explore}}, n_{\text{explore}})$
 915 10: $\mathcal{D}_{\text{EF-SFT}} \leftarrow \mathcal{D}_{\text{base}}^{\text{sub}} \cup \mathcal{D}_{\text{explore}}^{\text{sub}}$
 916 11: **return** $\mathcal{D}_{\text{EF-SFT}}$

918 A.4 TEST-TIME SCALING EXPERIMENTAL DETAILS
919920 This section provides a detailed description of the experimental setup for the test-time scaling com-
921 parison presented in Table 3, ensuring reproducibility.
922923 **Objective and General Setup** The primary goal was to evaluate the performance-cost trade-off
924 of our E²C framework against established baselines on the AIME’24 benchmark. All methods
925 were evaluated using the same checkpoint: the **Qwen3-8B+E²C-(SFT+RL)** model. This ensures a
926 fair comparison of the inference strategies themselves, rather than the underlying models. For all
927 generative steps that require diversity (e.g., sampling paths or plans), a temperature of 0.9 was used.
928 Performance is reported as Pass@1 accuracy, and cost is measured by the average total number of
929 tokens generated per question.
930931 **Baseline Methods**932

- **Greedy CoT**: A single reasoning chain was generated for each question using greedy decoding
933 (N=1). This serves as the most basic baseline.
- **Self-Consistency (SC)**: For each budget level $N \in \{4, 8, 16, 32\}$, we generated N full, indepen-
934 dent CoT reasoning chains. The final answer was determined by a majority vote among the N
935 outputs.
- **Tree-of-Thoughts (ToT) & Forest-of-Thought (FoT)**: We implemented these advanced search
936 methods following the standard procedures described in their respective papers (Yao et al., 2023;
937 Bi et al., 2025). The number of reasoning paths explored was set to match the budget levels N
938 $\in \{4, 8, 16, 32\}$ to ensure a comparable computational scale.

939940 **E²C Methods and Ablations** All E²C variants begin by sampling $K \in \{4, 8, 16, 32\}$ exploration
941 plans from the same model. The subsequent steps differ as follows:
942943

- **E²C-Select (Self LM-Judge)**: The K sampled plans and the original question were formatted
944 into a prompt for the model to act as a judge and select the single most promising plan. A single
945 execution was then generated conditioned on this selected plan.
- **E²C-Select (Semantic Cluster)**: This method involves a multi-step, voting-based process: (1)
946 Each of the K plans was embedded into a vector using the standard `all-mpnet-base-v2`
947 sentence-transformer model. (2) We applied K-Means clustering to group these embeddings
948 into $M=3$ distinct clusters. (3) The plan closest to the centroid of each of the M clusters was
949 selected for execution, resulting in M executions. (4) The final answer was determined by a
950 weighted majority vote over the M outcomes, where each vote’s weight was proportional to the
951 size of its corresponding cluster.
- **E²C-SC (Self-Consistency)**: This ablation executed all K sampled plans independently. The
952 final answer was determined by a standard majority vote over the K resulting outcomes. This
953 serves as a high-cost upper bound for the E²C paradigm.
- **E²C-RP (Random Plan)**: As a simple ablation, one plan was randomly selected from the K
954 samples and then executed to produce a single answer.

955956 A.5 ENTROPY VISUALIZATION OF DIFFERENT RL SETTINGS AND ANALYSIS
957958 In this part, we visualize the entropy dynamics and the accuracy on the AIME’24 benchmark during
959 RL training. The results demonstrate that applying our token-weighting coefficient $\lambda_{i,t}$ to explo-
960 ration tokens facilitates a rapid drop in entropy and a better performance improvement, as shown in
961 **Fig. 3**. This is achieved by effectively amplifying high-quality plans while suppressing poor ones.
962963 A.6 PROMPT DETAILS
964965 **E²C-SFT Dataset Construction prompt**966 EXPLORATION PHASE PROMPT The following prompt is used to extract the high-level explo-
967 ration plan from the reasoning process:
968

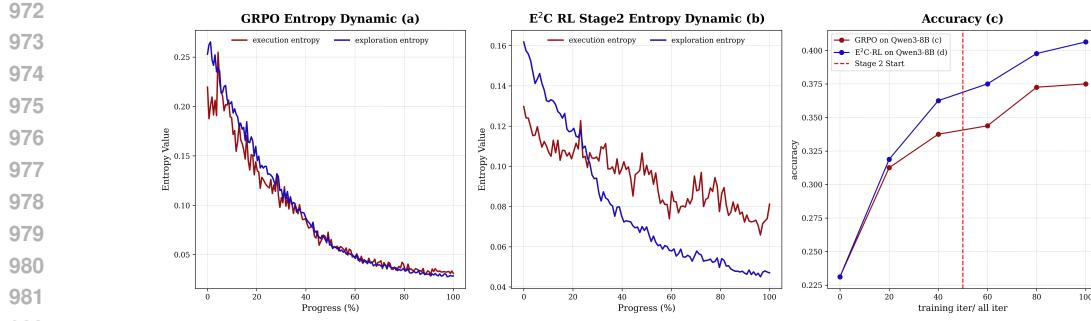


Figure 3: A comparison of training dynamics on the AIME'24 benchmark. The application of our token-weighting coefficient $\lambda_{i,t}$ (b) facilitates faster entropy reduction and superior performance improvement compared to the baseline without it (a).

Role: You are an expert problem-solver.

Task: Distill a complex reasoning process into a clear, actionable plan.

Input:

- **Problem:** <question>
- **Reasoning Process:** <content>

Output Requirements:

1. **Format:** Present the summary as a numbered list (e.g., 1., 2., 3.).
2. **Content:** For each step, describe only the essential action to be taken (e.g., “Calculate X,” “Verify Y”). Be concise and prescriptive.
3. **Focus:** Omit explanations, justifications, or intermediate conclusions.

Goal: Create a high-level plan that is easy to follow and execute.

EXECUTION PHASE PROMPT The following prompt is used to generate the detailed execution steps based on the exploration plan:

Role: You are a meticulous problem solver.

Task: Solve the given question by strictly following the provided guideline, showing all detailed reasoning.

Input:

- **Question:** <question>
- **Guideline:** <content>

Output Requirements:

1. Follow the guideline exactly, numbering each step accordingly (e.g., 1., 2., ...).
2. Do not include any content outside the solution steps.
3. Begin from Step 1, expanding each step with necessary calculations and logical reasoning.
4. Conclude by placing the final answer within a ‘`\boxed{ }`’ environment.

Important: Ensure every mathematical or logical operation is explicitly shown.

EF-SFT dataset Construction prompt The following prompt is used to extract the exploration part for EF-SFT dataset in medical domain.

Role: You are a professional doctor.

Task: Summarize the diagnostic reasoning process into a concise, actionable

1026 guideline.
 1027

1028 **Input:**

1029 • **Question:** <question>
 1030 • **Reasoning Process:** <content>

1031 **Output Requirements:**

1032 1. **Structure:** Present the summary as a numbered list (1., 2., ...), starting directly with the first step.
 1033 2. **Conciseness:** Use no more than 5 steps. Each step must be under 15 words
 1034 and state only the critical objective (e.g., “Assess cardiac function”).
 1035 3. **Focus:** Highlight the most critical diagnostic step. Omit all explanations,
 1036 justifications, or unrelated content.

1037 **Goal:** Create a concise and accurate diagnostic plan focused on key actions.
 1038

1039 **LLM-Combination Prompt** To enable the model to select the most promising exploration plan,
 1040 we use the following prompt. The model is instructed to act as an impartial judge, evaluating the
 1041 provided plans based on their clarity, correctness, and likelihood of leading to a successful solution.
 1042

1043 **Role:** You are an expert mathematical reasoner and an impartial judge. Your task
 1044 is to evaluate several proposed plans for solving a given math problem and identify
 1045 the single best one.
 1046

1047 **Input:**

1048 • **Problem:** <problem>

1049 • **Candidate Plans:** A numbered list of K exploration plans. Plan
 1050 1 : <exploration₁ > Plan₂ :< exploration₂ > ...Plan_K :<
 1051 exploration_K >

1052 **Instructions:**

1053 1. Carefully analyze the problem and each of the K candidate plans.
 1054 2. Assess the plans based on their logical soundness, potential for success, and
 1055 efficiency.
 1056 3. Select the single best plan that is most likely to lead to a correct and complete
 1057 solution.
 1058

1059 **Output Format:** Output only the full text of the single best plan you have se-
 1060 lected. Do not add any extra commentary, explanation, or formatting.
 1061

1062 **Adherence Judge Prompt** The following prompt is used to evaluate whether an execution strictly
 1063 adheres to the provided exploration plan.
 1064

1065 **Role:** You are a rigorous evaluator. Your task is to judge if the execution strictly
 1066 follows the exploration plan.
 1067

1068 **Input:**

1069 • **Question:** question
 1070 • **Exploration (PLAN):** exploration
 1071 • **Execution:** execution

1072 **Evaluation Criteria:** Provide a final score of 0, 0.5, or 1.0 based on the following:
 1073

1074 • **1.0 (Strictly Adheres):** The execution follows the exploration’s logic and
 1075 key steps exactly. All critical reasoning or calculations in the exploration are
 1076 present and correctly implemented in the execution.
 1077 • **0.5 (Partially Adheres):** The execution addresses the main goal but deviates
 1078 in specifics: it may skip non-critical steps, change the order of minor steps,
 1079 or contain minor logical gaps while reaching a correct conclusion.
 1080 • **0.0 (Does Not Adhere):** The execution ignores major exploration steps, fol-
 1081 lows a different approach, contains significant logical errors, or fails to im-
 1082 plement the exploration’s core intent.

1080 **Output Format:** First, provide a concise rationale comparing the exploration and
 1081 execution. Then, output the score strictly as:
 1082 [RATIONALE] : <your analysis>
 1083 [SCORE] : <0.0, 0.5, or 1.0>
 1084

1085 **A.7 E²C REASONING EXAMPLES**

1087 **Example 1(Math, from AMC23): Meeting Problem**

1089 **INPUT QUESTION** Cities *A* and *B* are 45 miles apart. Alicia lives in *A* and Beth lives in *B*. Alicia
 1090 bikes towards *B* at 18 miles per hour. Leaving at the same time, Beth bikes toward *A* at 12 miles
 1091 per hour. How many miles from City *A* will they be when they meet? Provide the final answer in
 1092 the boxed.
 1093

1094 **EXPLORATION PHASE** 1. Identify the speeds and initial distance between the two cities. 2. De-
 1095 termine the combined speed at which the distance between Alicia and Beth is closing. 3. Calculate
 1096 the time it takes for them to meet by dividing the initial distance by their combined speed. 4. Use
 1097 the time to calculate the distance Alicia travels from City *A*.
 1098

1099 **EXECUTION PHASE**

1101 **1. Identify the speeds and initial distance between the two cities.** Distance between City
 1102 *A* and City *B*: 45 miles. Alicia's speed: 18 miles per hour (towards *B*). Beth's speed: 12
 1103 miles per hour (towards *A*)

1104 **2. Determine the combined speed at which the distance between Alicia and Beth is clos-
 1105 ing.** Since they are moving towards each other, their speeds add up:

$$1107 \quad \text{Combined speed} = 18 + 12 = 30 \text{ miles per hour}$$

1108 **3. Calculate the time it takes for them to meet by dividing the initial distance by their
 1109 combined speed.** The time until they meet is:

$$1111 \quad \text{Time} = \frac{\text{Distance}}{\text{Combined speed}} = \frac{45}{30} = 1.5 \text{ hours}$$

1114 **4. Use the time to calculate the distance Alicia travels from City A.** Alicia's distance from
 1115 City *A* when they meet is:

$$1116 \quad \text{Distance} = \text{Speed} \times \text{Time} = 18 \times 1.5 = 27 \text{ miles}$$

1118 **FINAL ANSWER**

1120 **27**

1122 **Example 2 (Math, from AIME'24): Table Tennis Tournament Problem**

1124 **INPUT QUESTION** In a table tennis tournament every participant played every other participant
 1125 exactly once. Although there were twice as many right-handed players as left-handed players, the
 1126 number of games won by left-handed players was 40% more than the number of games won by
 1127 right-handed players. (There were no ties and no ambidextrous players.) What is the total number
 1128 of games played? Provide the final answer in the boxed.
 1129

1130 **EXPLORATION PHASE** 1. Define variables for the number of left-handed and right-handed play-
 1131 ers. 2. Express the total number of games played in terms of the total number of players. 3. Deter-
 1132 mine the total number of games won by left-handed and right-handed players. 4. Set up an equation
 1133 based on the given relationship between wins. 5. Solve the equation to find the total number of
 1134 players. 6. Calculate the total number of games played.

1134
1135

EXECUTION PHASE

1136
1137
1138

1. **Define variables for the number of left-handed and right-handed players.** Let L be the number of left-handed players, and R be the number of right-handed players. Given: $R = 2L$

1139
1140
1141

2. **Express the total number of games played in terms of the total number of players.**

Total players: $L + R = 3L$ Total games: $\binom{3L}{2} = \frac{3L(3L-1)}{2}$

1142
1143
1144

3. **Determine the total number of games won by left-handed and right-handed players.** Let W_L be wins by left-handed players, W_R be wins by right-handed players. Total wins: $W_L + W_R = \frac{3L(3L-1)}{2}$ Given: $W_L = 1.4W_R$

1145
1146
1147

4. **Set up an equation based on the given relationship between wins.** Substitute $W_L = 1.4W_R$ into total wins: $1.4W_R + W_R = \frac{3L(3L-1)}{2}$ $2.4W_R = \frac{3L(3L-1)}{2}$

1148
1149
1150

5. **Solve the equation to find the total number of players.** Try $L = 3$: $W_R = \frac{3 \times 3 \times (9-1)}{4.8} = \frac{72}{4.8} = 15$ $W_L = 1.4 \times 15 = 21$ Total wins: $15 + 21 = 36$

1151
1152

6. **Calculate the total number of games played using the total number of players.** Total games: $\frac{3 \times 3 \times (9-1)}{2} = \frac{72}{2} = 36$

1153
1154
1155

FINAL ANSWER

36

1156
1157**Example 3 (MedicalFrom MMLU-Anatomy): Urinary System Question**1158
1159
1160
1161

INPUT QUESTION Which of the following best describes the structure that collects urine in the body? A. Bladder B. Kidney C. Ureter D. Urethra Please reasoning step-by-step. Provide the final answer in the boxed.

1162
1163
1164
1165

EXPLORATION PHASE 1. Identify the function of each organ listed in the options. 2. Determine which organ is responsible for storing urine. 3. Eliminate options that do not fit the function of collecting urine. 4. Select the organ that best matches the description of collecting urine.

1166
1167

EXECUTION PHASE

1168
1169
1170

1. **Identify the function of each organ listed in the options:**

- **A. Bladder:** Hollow, muscular organ that stores urine until excretion
- **B. Kidney:** Filters blood and produces urine, but does not store urine
- **C. Ureter:** Transports urine from kidneys to bladder, does not store urine
- **D. Urethra:** Carries urine from bladder to outside body, does not store urine

1171
1172
1173
1174

2. **Determine which organ is responsible for storing urine:** The bladder is the primary organ responsible for urine storage. It expands as it fills with urine and contracts during urination.

1175
1176
1177
1178

3. **Eliminate options that do not fit the function of collecting urine:**

- Kidney (B): Produces urine but does not store it
- Ureter (C): Transports urine but does not store it
- Urethra (D): Excretes urine but does not store it
- Bladder (A): Only organ designed for urine collection and storage

1179
1180
1181
1182
1183
1184
1185

4. **Select the organ that best matches the description of collecting urine:** The bladder is the structure that collects and stores urine in the body.

1186
1187

FINAL ANSWER

A

1188
 1189 **Example 4 (Test-Time Scaling, from AIME'24): Table Tennis Tournament Problem** This
 1190 example demonstrates the application of our two primary test-time scaling methods using the same
 1191 input question from Example 2.

1192 INPUT QUESTION In a table tennis tournament every participant played every other participant
 1193 exactly once. Although there were twice as many right-handed players as left-handed players, the
 1194 number of games won by left-handed players was 40% more than the number of games won by
 1195 right-handed players. (There were no ties and no ambidextrous players.) What is the total number
 1196 of games played? Provide the final answer in the boxed.

1197
 1198 **STEP 1: SAMPLE K EXPLORATION PLANS (K=4)** First, the E^2C model is prompted to generate
 1199 $K=4$ diverse, high-level exploration plans for the problem.

- 1200 • **Plan 1** 1. Let L be the number of left-handed players, the total games are $C(3L, 2)$. 2. Let W_L
 1201 be wins by lefties and W_R be wins by righties. 3. Solve this system for L , then find the total
 1202 games.
- 1203 • **Plan 2** 1. The total number of players must be a multiple of 3, let's call it $3L$. The total games
 1204 played is $C(3L, 2)$. 2. Let's test small integer values for L ($L=1, 2, 3\dots$) and check if the
 1205 resulting total games can be split into wins for left- and right-handed players satisfying the 40%
 1206 more condition.
- 1207 • **Plan 3** 1. Assume the number of wins is proportional to the number of players. Let right-handed
 1208 players have W_R wins. 2. Left-handed players have half the number of players, so they should
 1209 have W_L wins. 3. Set up $W_L = 1.4 * W_R$ and solve based on the total number of games.
- 1210 • **Plan 4** 1. Let the number of wins by right-handed players be W_R . Then the wins by left-handed
 1211 players is $1.4 * W_R$. 2. The total number of games is $2.4 * W_R$. The total number of games is
 1212 also given by $C(3L, 2)$. 3. Set $C(3L, 2) = 2.4 * W_R$ and find an integer solution for L .

1214
 1215 **METHOD A: E^2C -SELECT (SELF LM-JUDGE)** The four plans above, along with the original
 1216 question, are fed into the model with the Self LM-Judge prompt. The model evaluates the plans and
 1217 selects the most robust and direct strategy.

- 1218 1. **Selection:** The Self LM-Judge identifies Plan 1 as the most comprehensive and logically
 1219 sound approach, as it correctly sets up the system of equations from first principles.
- 1220 2. **Execution:** A single execution is performed, conditioned only on Plan 1. This execution
 1221 proceeds exactly as detailed in Example 2, arriving at the correct answer.

1223
 1224 **Final Answer (Self LM-Judge):** 36

1225
 1226 **METHOD B: E^2C -SELECT (SEMANTIC CLUSTER)** This algorithmic method clusters the plans
 1227 before execution.

- 1228 1. **Embedding and Clustering:** The four plans are embedded into vectors. A clustering
 1229 algorithm (e.g., K-Means) is applied and identifies $M=3$ distinct strategic groups:
 - 1230 • **Cluster A** Plan 1 and Plan 4 are grouped together as they both use a correct algebraic
 formulation. (Cluster Size = 2)
 - 1231 • **Cluster B** Plan 2 is identified as a distinct trial-and-error strategy. (Cluster Size = 1)
 - 1232 • **Cluster C** Plan 3 is isolated as it is based on an incorrect assumption. (Cluster Size =
 1)
- 1233 2. **Centroid Execution:** The plan closest to the centroid of each cluster is selected and exe-
 1234 cuted.
 - 1235 • **Execution of A (from Plan 1):** Results in the correct answer, **36**.
 - 1236 • **Execution of B (from Plan 2):** Also results in the correct answer, **36**.
 - 1237 • **Execution of C (from Plan 3):** The flawed logic leads to an incorrect answer, e.g.,
45.

1242 3. **Weighted Majority Vote:** The final answer is determined by a weighted vote of the execution outcomes.

1243 • Vote for answer "36": Received from Cluster A (weight=2) and Cluster B (weight=1).
 1244 Total weight = $2 + 1 = 3$.
 1245 • Vote for answer "45": Received from Cluster C (weight=1). Total weight = 1.

1246 The answer "36" has the highest weight.

1247 1248 **Final Answer (Semantic Cluster):** 36

1249 1250 A.8 PURE PROMPT-BASED E²C

1251 1252 We product an experiment with pure prompt-based E²C on Qwen3-8B. For each problem we first
 1253 1254 sample K independent *exploration* traces by prompting the model K times with a short exploration
 1255 1256 prompt; each exploration is a concise (2–4 short sentence) reasoning sketch that does not contain
 1257 1258 the final answer. We then combine the K explorations into a single execution prompt (providing the
 1259 1260 problem and the numbered explorations) and ask the model to produce one final *Execution*: section
 1261 1262 that computes the final answer. Performance is reported as pass@5 for different values of K . The
 1263 1264 results are much worse than the E²C model with E²C-(SFT+RL), which demonstas that a prompt
 1265 1266 engeneering is not enough.

1267 **Exploration prompt** The following prompt was used to generate each individual exploration (one
 1268 1269 exploration per model call).

1270 **Role:** You are a careful math problem solver.

1271 **Input:**

1272 • **Problem:** <problem>

1273 **Instructions:**

1274 • Produce exactly one short reasoning sketch (an *exploration*) that helps approach the problem.
 1275 • The exploration must be concise (about 2–4 short sentences).
 1276 • Do **not** produce the final answer in this call.
 1277 • Stop immediately after the single exploration text and do not append any extra commentary, labels, or formatting.

1278 **Output format:** A single short exploration paragraph (2–4 short sentences) and nothing else.

1279 **Execution prompt** The following prompt was used to synthesize the K independently sampled explorations into a final execution.

1280 **Role:** You are a careful math problem solver.

1281 **Input:**

1282 • **Problem:** <problem>

1283 • **Explorations:**

1284 Exploration 1: <exploration 1>

1285 Exploration 2: <exploration 2>

1286 ⋮

1287 Exploration $\{K\}$: <exploration K>

1288 1289 Table 7: Pass@5 accuracy (%) for different numbers of sampled explorations K .

Dataset	$K = 2$	$K = 3$	$K = 4$	$K = 5$
MATH500	84.4	83.2	84.0	84.0
AIME24	26.7	33.0	36.7	26.7
AIME25	23.3	30.0	30.0	26.7

1296

1297

1298

1299

1300

1301

1302

1303

1304

1305

1306

1307

1308

1309

1310

1311

1312

1313

1314

1315

1316

1317

1318

1319

1320

1321

1322

1323

1324

1325

1326

1327

1328

1329

1330

1331

1332

1333

1334

1335

1336

1337

1338

1339

1340

1341

1342

1343

1344

1345

1346

1347

1348

1349

Instructions:

- Learn from the provided $\{K\}$ numbered explorations and combine their useful reasoning to compute the final answer.
- Produce a single **Execution:** section that carries out the computation and presents the final answer.
- Stop immediately after the final answer. Do not append extra commentary, explanations, or any additional text beyond the required Execution section and the answer.