When Are Compositional Problems Learnable from Verifiable Rewards?

Daniel Barzilai[*]
Weizmann Institute of Science
daniel.barzilai@weizmann.ac.il

Yotam Wolf[*]
Hebrew University of Jerusalem
yotamwolf@cs.huji.ac.il

Ronen Basri
Weizmann Institute of Science
ronen.basri@weizmann.ac.il

Abstract

Many problems are inherently compositional: solving them requires a sequence of intermediate decisions that jointly determine the final answer. A central question is when such a compositional structure can be learned from outcome-level feedback alone, where supervision of the intermediate steps is unavailable. We study this question for autoregressive models trained using reinforcement learning with verifiable rewards (RLVR). We identify the *task-advantage ratio*, a joint property of the task and the model, which measures whether intermediate decisions present an advantage in reaching a correct final solution. We show that this ratio governs learnability in our setting: when the advantage is present, RLVR efficiently learns the target composition, while when it is absent, training can converge to suboptimal compositions. We further show that the required advantage arises naturally in several structured problems, but may depend critically on the quality of the initial model. Our results help to clarify when compositional problems can be learned from final rewards alone.

1 Introduction

Chain of Thought (CoT) enables autoregressive models to solve complex problems by decomposing them into simpler intermediate steps [Wei et al., 2022], substantially improving their performance on reasoning tasks. A common approach to teaching models to carry out such CoT reasoning is Reinforcement Learning with Verifiable Rewards (RLVR) [Jaech et al., 2024, Guo et al., 2025, Kimi Team et al., 2025]. In RLVR, the model generates its own intermediate computations and receives a reward based on the final outcome, without supervision of the individual steps.

Training using only final rewards (as in RLVR) introduces a credit-assignment problem. Many different trajectories may lead to a verified outcome, and the verifier does not specify which intermediate choices were responsible for success. A model may receive positive reward despite taking incorrect intermediate steps, so the existence of a decomposition that always produces a correct answer does not establish that RLVR will recover it.

Existing theoretical work has shown how CoT expands the computational power of autoregressive models [Merrill and Sabharwal, 2024, Strobl et al., 2024] and how supervision on correct intermediate steps makes complex compositional problems efficiently learnable [Malach, 2023, Wies et al., 2022]. However, these expressivity and supervised learnability results do not explain how RLVR learns

[*]equal contribution

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

intermediate computations from final rewards alone, when the target decomposition is never shown to the learner. This motivates our central question:

When can outcome-level feedback induce the correct intermediate computation in autoregressive models?

To study how outcome-level feedback propagates to intermediate decisions, we formulate a tractable autoregressive model that isolates the *task-selection* mechanism. At each step, the model selects from a finite collection of tasks already available in the base model and applies the selected task to the generated prefix. Training must discover how to compose these tasks from final rewards, with all steps learned simultaneously and without observing the target intermediate computations.

Our main result is an exact characterization of the expected step-wise learning signal induced by training on verified rollouts. We identify the *task-advantage ratio*: the ratio between the probability of verification success when a task *is* selected at a given step and the probability of success when it *is not* selected. This ratio determines the sign of the task's expected logit update. A task receives positive credit when selecting it increases the chance of eventual verification success, whether or not it belongs to the target composition.

When a uniform task-advantage is present for the correct tasks and incorrect tasks have a disadvantage within the relevant policy region, we prove that within our setting, RLVR converges to the correct composition in a number of iterations that scales quadratically with the composition length. However, without such an advantage, RLVR can converge to suboptimal compositions and poor local minima even in the absence of representational or statistical barriers.

Because the task-advantage ratio is a joint property of the compositional problem and the base model, it reveals the roles of both. First, some compositional problems have a stronger inductive structure than others. For example, in bit-parities with uniformly initialized task choices, a correct intermediate decision yields only an exponentially small task-advantage. Second, if the base model frequently reaches verified solutions despite a poor reasoning path, suboptimal intermediate computations may be reinforced, leading RLVR to converge to a suboptimal local minimum.

Contributions. Our main contributions are as follows:

- In Sec. 4, we introduce a theoretical framework for analyzing RLVR in autoregressive compositional settings, using a tractable model that isolates the task-selection mechanism.
- In Theorem 5.2, we derive an exact expression for the expected step-wise learning signal induced by RLVR, showing that its direction is governed by the task-advantage ratio.
- In Theorem 5.4, we identify a natural structural condition, phrased in terms of the task-advantage ratio, under which RLVR provably recovers the correct composition.
- In Sec. 6, we show how the base model may impact the success of RLVR through the task-advantage ratio. In particular, we present a minimalistic setting where RLVR succeeds if and only if the base model accuracy passes a constant threshold.
- In Sec. 7, we demonstrate how the complexity of learning various types of problems depends on the task-advantage ratio.

Together, these results clarify when RLVR can induce correct intermediate computation and when it cannot, highlighting the central roles of the problem structure and the base model.

2 Related Work

RLVR and LLM reasoning. RLVR has emerged as a popular paradigm for improving the reasoning capabilities of LLMs, especially on tasks such as math and programming [Jaech et al., 2024, Guo et al., 2025, Kimi Team et al., 2025, Wang et al., 2025]. This is largely due to alternative approaches being unstable in many cases [Gao et al., 2023]. Despite its success, it is still unclear how RLVR changes the reasoning ability of LLMs. On the more critical side, Yue et al. [2025] observed empirically that in many cases, RLVR improved the pass@k accuracy only when k is small; however, for large k, the results were often worse than those of the base model. This can be interpreted as RLVR "sharpening" the base model, but ultimately relying on knowledge acquired during pre-training. Yuan et al. [2025], Wu et al. [2025] observe empirically that using RLVR, LLMs

may learn to compose skills they learned during pre-training, but the extent to which they can learn completely new skills remains unclear. In our work, we study this compositional ability in depth.

On the theoretical side, Foster et al. [2025] study computationally efficient exploration with linear-softmax policies and how base-model coverage of high-reward responses affects its cost. Their coverage condition concerns the base model's probability of generating high-reward responses; by contrast, our task-advantage condition concerns whether selecting a given intermediate task increases that probability. Jia et al. [2025] showed that under suitable coverage assumptions, outcome supervision is no more statistically difficult than process supervision, up to polynomial factors in sequence length. Tsilivis et al. [2026] analyze a parity model pretrained on a mixture containing complete correct CoTs and direct answers, so that at the start of post-training the model already generates a fully correct CoT with non-negligible probability. Their setting therefore concerns whether post-training will learn to favor the already known CoT path over a direct answer, rather than learning how to compose the available intermediate tasks.

Learnability of autoregressive compositional problems. A core question for reasoning models is how well they can generalize from solving simple tasks to solving compositions of those tasks. Abedsoltan et al. [2025] analyzed a setting in which tasks are sampled i.i.d. from some distribution of tasks with compositional structure, and infinitely many supervised samples are available to the learner per task. In this setting, they showed that new tasks can readily be identified at inference time. Abbe et al. [2024] showed that in some settings, there are problems that can be difficult to learn unless they admit a simple compositional structure.

Several works consider learning problems where, unlike our setting, intermediate supervision on the Chain of Thought (CoT) is available. We note that in practice, obtaining such supervision is difficult, especially at large scales. Wies et al. [2022] showed that with supervision on CoTs, an autoregressive model can learn in polynomial time problems that would require exponential time otherwise. Kim and Suzuki [2024], Wen et al. [2024] analyze sparse parities and show that they can be learned efficiently *with intermediate supervision*. Our results complement this by characterizing, through the task-advantage ratio, the difficulty of such problems without intermediate supervision.

Other benefits of Chain of Thought. There are many works that analyze theoretically the benefits of the chain of thought from perspectives that differ from ours. One such series of works analyzes how CoTs increase the expressivity of autoregressive models [Malach, 2023, Strobl et al., 2024]. For example, Merrill and Sabharwal [2024] showed that a transformer empowered with a CoT can simulate a TM of time complexity $T(n)$ with $T(n)$ steps of CoT. Other works analyze how the width and depth of the model interact with the number of CoT steps needed in various types of problems [Sanford et al., 2024, Yehudai et al., 2025]. Mirtaheri et al. [2025] show that in some problems long CoTs cannot be efficiently replaced by majority voting between many short ones.

Another line of work analyzes learnability in frameworks related to PAC [Malach, 2023, Joshi et al., 2025, Shalev-Shwartz and Shashua, 2025, Yang et al., 2025]. For example, Malach [2023] studies compositional learning with teacher forcing on ground-truth CoTs, while Joshi et al. [2025] consider the setting where each prompt is still paired with its correct final token (making their setting closer to supervised learning). In our setting, the model generates its own potentially incorrect prefixes and receives only a binary reward on the resulting trajectory. It must therefore learn which sampled intermediate tasks to reinforce, with all steps learned simultaneously and without observing the target decomposition.

3 Learning Compositions Without Supervision

Many problems can be solved by composing simpler operations sequentially. In autoregressive models, such compositions are often expressed as chains of thought (CoTs), where intermediate tokens represent partial computations leading to a final answer [Wei et al., 2022, Guo et al., 2025]. A central question is whether such compositional structure can be learned *without* explicit supervision of intermediate steps. To reason about this, we formalize the notion of an autoregressive composition. Throughout this paper, we let $\mathcal{V}$ be a vocabulary and $\mathcal{V}^*$ denote all finite strings over $\mathcal{V}$, meaning $\mathcal{V}^* := \{(w_1, \ldots, w_n) \mid n \in \mathbb{N}, w_i \in \mathcal{V}\}$ (also known as the Kleene closure of $\mathcal{V}$).

Definition 3.1 (Autoregressive Compositions)**.** Let $\mathcal{V}$ be a vocabulary and $\mathbb{T} = \{\sigma_j : \mathcal{V}^* \to \mathcal{V}\}_{j=1}^{J}$ be a set of tasks, each mapping a sequence of tokens to a single token. A function $f : \mathcal{V}^* \to \mathcal{V}$ is an

autoregressive composition (relative to $\mathbb{T}$) if there exist indices $j_1, \ldots, j_S$ such that for any input $\mathbf{x}$, applying $\sigma_{j_1}, \ldots, \sigma_{j_S}$ sequentially in an autoregressive manner produces a final token equal to $f(\mathbf{x})$.

We emphasize that the set of tasks is usually restricted so that no individual task is equal to f, and thus compositionality is strictly required. Learning such compositions from outcome-level feedback is nontrivial as it does not uniquely identify which intermediate choices were responsible for success. Multiple compositions may yield a verified final output, and incorrect intermediate steps may be compensated for later in the sequence. As a result, even when a correct composition exists and achieves perfect verification accuracy, it is not immediate that training from final outcomes alone should recover a correct chain of intermediate decisions. In this paper, we analyze when outcome-level training induces correct step-wise behavior and when it does not.

4 Setting and Preliminaries

Autoregressive generation. We consider an autoregressive language model $f_\theta(\mathbf{x})$ that, given a context $\mathbf{x} \in \mathcal{V}^*$, predicts the logits for the next token. This defines a policy for the next token

$$\boldsymbol{\pi}_\theta(y \mid \mathbf{x}) = \mathrm{Softmax}(f_\theta(\mathbf{x}))[y],$$

where $[y]$ denotes the index corresponding to token y. Given an initial prompt $\mathbf{x}_0$, the model generates a completion of fixed length S in an autoregressive fashion. Specifically, tokens are generated sequentially according to

$$\forall s \in [S], \quad y_s \sim \boldsymbol{\pi}_\theta\left(\cdot \mid \mathbf{x}_{s-1}\right), \quad \mathbf{x}_s := \left(\mathbf{x}_{s-1}, y_s\right),$$

so that $\mathbf{x}_s$ is the prefix consisting of the prompt followed by the first s generated tokens. This procedure induces a distribution over entire sequences of tokens, which we denote by $\mathbb{P}_\theta$ (so for any token v, $\mathbb{P}_\theta(y_s = v)$ (the probability of sampling v at step s) is determined by the distribution of prompts and $\mathbb{P}_\theta(y_s = v \mid \mathbf{x}_{s-1}) = \boldsymbol{\pi}_\theta(v \mid \mathbf{x}_{s-1})$).

Model structure. We study the problem of learning compositional behavior in autoregressive models using a model that isolates the task selection mechanism. Concretely, given an input, the model outputs a distribution over tokens proposed by a set of tasks (or skills), each task mapping a prefix to a candidate next token. We emphasize that while each task is a deterministic function of the prefix, stochasticity arises from the model's distribution over tasks at each step. This is mainly for analytical clarity; stochastic skills could be represented by mixtures over deterministic ones, though doing so may enlarge the task class. Learning then proceeds by redistributing probability mass over these tasks, rather than modifying their internal behavior. This abstraction is motivated by empirical observations in reasoning models trained with outcome-level feedback, where improvements are often attributed to reweighting and recombining existing behaviors rather than introducing entirely new primitives [Yue et al., 2025, Guo et al., 2025].

Tasks. Let $\mathbb{T} = \{\sigma_j : \mathcal{V}^* \to \mathcal{V}\}_{j=1}^J$ denote a set of tasks, where each task σ_j maps a prefix to a single next token, with $\mathcal{V}^*$ denoting the Kleene closure of the vocabulary. We associate to each task a function $g_j : \mathcal{V}^* \to \mathbb{R}^{|\mathcal{V}|}$ defined by

$$g_j(\mathbf{x}) = \gamma \mathbf{e}_{\sigma_j(\mathbf{x})},$$

where $\gamma > 0$ is a fixed constant (to be thought of as large) and $\mathbf{e}_v$ denotes the standard basis vector corresponding to token v. Intuitively, $g_j(\mathbf{x})$ represents the logits proposed by task σ_j at prefix $\mathbf{x}$. When γ is large, sampling according to $\mathrm{Softmax}(g_j(\mathbf{x}))$ yields the token $\sigma_j(\mathbf{x})$ with high probability. No single task is assumed to be capable of solving the full problem on its own.

Positional task embeddings. We associate each task σ_j and autoregressive step $s \in [S]$ with an embedding vector $\mathbf{h}_{s-1,j} \in \mathbb{R}^p$. These embeddings allow the model to assign different preferences to tasks at different stages of the computation. We assume that the collection $\{\mathbf{h}_{s-1,j}\}_{s\in[S],j\in[J]}$ is orthonormal, so an update along one embedding does not change the parameter's alignment with the others. Tasks still interact through the generated prefix and final reward: the expected update for a task at one step depends on the policy at other steps. For simplicity, the embeddings depend only on the step index $s(\mathbf{x})$, rather than the full prefix. This corresponds to modeling a setting in which the correct computation follows a fixed sequence of task applications. More generally, one could allow the embeddings to depend on a state derived from the prefix, leading to a state-dependent decision process (see appendix F). We focus on the step-based formulation to highlight the core mechanism.

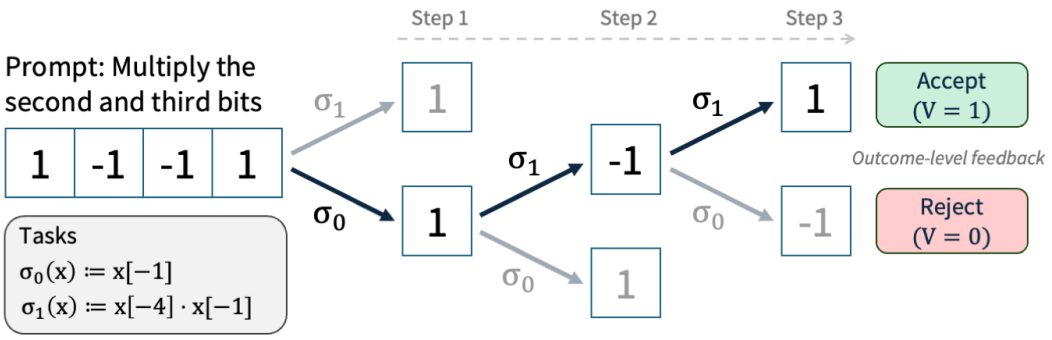


Figure 1: RLVR as sequential task selection. An autoregressive model chooses among tasks at each step, each deterministically producing the next token. Only the final output is verified, and intermediate decisions receive no direct supervision. Dark arrows indicate the correct task composition.

Model definition. Given these components, we define the model as

$$f_\theta(\mathbf{x}) = \sum_{j=1}^{J} g_j(\mathbf{x}) \langle \mathbf{h}_{s(\mathbf{x}),j}, \theta \rangle, \tag{1}$$

where $s(\mathbf{x})$ denotes the number of generated CoT tokens in $\mathbf{x}$, and $\theta \in \mathbb{R}^p$ are the trainable parameters. Thus, the alignment of the parameters θ with the various task embeddings determines the distribution over tokens proposed by various tasks. The derivation of this model is motivated by the compositional task selection mechanism. We note that it may alternatively be motivated using what is known as the Neural Tangent Kernel; see Appendix E for more details.

Compositional target and learning objective. We consider problems that admit a correct solution in the form of an autoregressive composition of tasks. For clarity, we restrict attention to compositions of fixed length S. Let $\tau(1), \ldots, \tau(S) \in [J]$ denote the indices of the tasks appearing in the correct composition. Given an initial prompt $\mathbf{x}_0$, define the target CoT by composing tasks $\sigma_{\tau(1)}, \ldots, \sigma_{\tau(S)}$. Formally, let $f_0^\star(\mathbf{x}_0) := \mathbf{x}_0$, and for all $s \in [S]$

$$f_s^\star(\mathbf{x}_0) := \left(f_{s-1}^\star(\mathbf{x}_0), \sigma_{\tau(s)} \left(f_{s-1}^\star(\mathbf{x}_0) \right) \right),$$

and $f^\star(\mathbf{x}_0) := f_S^\star(\mathbf{x}_0)$ denotes the full target string, including the prompt; its last token is the output of the autoregressive composition.

Outcome-based post-training. We assume access to a verifier V that evaluates the final strings $\mathbf{x}_S$, with $V(\mathbf{x}_S) \in \{0, 1\}$ denoting an incorrect or correct answer. When the input is clear, we may simply write $V = 1$ to denote the event that $V(\mathbf{x}_S) = 1$. We fine-tune the pre-trained weights $\theta^{(0)}$ using a procedure related to REINFORCE [Williams, 1992]. At each iteration t, a candidate prompt–completion pair is sampled by drawing a prompt $\mathbf{x}_0 \sim \mathcal{D}$ and generating a completion of length S autoregressively under the current policy. Specifically, writing $\boldsymbol{\pi}_t$ for $\boldsymbol{\pi}_{\theta^{(t)}}$, at each step $s \in [S]$,

$$y_s \sim \boldsymbol{\pi}_t(\cdot \mid \mathbf{x}_{s-1}), \quad \mathbf{x}_s := (\mathbf{x}_{s-1}, y_s).$$

We denote by $A_{s,j}$ the event that task σ_j is selected at step s, equivalently $A_{s,j} = \{y_s = \sigma_j(\mathbf{x}_{s-1})\}$.

At each iteration, we independently sample prompt–completion pairs from the current rollout distribution, rejecting unsuccessful pairs until B verified pairs have been collected. We index the accepted pairs by $b \in [B]$. Given a learning rate $\eta > 0$, we update the parameters as

$$\theta^{(t+1)} = \theta^{(t)} + \frac{\eta}{B} \sum_{b=1}^{B} \sum_{s=1}^{S} \nabla \log \boldsymbol{\pi}_t \left(y_s^{(t,b)} \mid \mathbf{x}_{s-1}^{(t,b)} \right).$$

The training procedure is presented as Algorithm 1 in the appendix. Training on successful model-generated solutions is a practical approach to bootstrapping reasoning, as exemplified by STaR

[Zelikman et al., 2022]. We also note that its update direction is the same as binary-reward REINFORCE without a baseline, since

$$\mathbb{E}_{\mathbb{P}_\theta}[V \nabla_\theta \log \mathbb{P}_\theta(\mathbf{x}_S \mid \mathbf{x}_0)] = \mathbb{P}_\theta(V=1)\, \mathbb{E}_{\mathbb{P}_\theta}[\nabla_\theta \log \mathbb{P}_\theta(\mathbf{x}_S \mid \mathbf{x}_0) \mid V=1]\,.$$

We omit KL regularization to isolate the learning signal induced by the verifier.

We denote by $\mathbb{P}_t$ the distribution over sequences induced by $\theta^{(t)}$, and by $\mathbb{E}_t$ the corresponding expectation. We let $\mathcal{B}_t$ denote the batch of positive samples at time t (sampled from $\mathbb{P}_{\theta^{(t)}}(\cdot \mid V=1)$).

We emphasize that because the model generates the CoT steps that are fed to the verifier, the dynamics may differ greatly from supervised training. As we will later show explicitly, in this setting, it is possible to converge to poor local minima.

Assumptions. First, since $f^\star$ represents the ideal function we are trying to learn, we assume that it indeed passes the verifier.

Assumption 4.1. For prompts $\mathbf{x}_0 \sim \mathcal{D}$ it holds with probability 1 that $V\left(f^\star(\mathbf{x}_0)\right) = 1$.

We emphasize that the target composition need not be the only one that produces a correct final output according to the verifier V. Next, we assume that the tasks do not overlap in the following sense:

Assumption 4.2. For any $j \neq j'$ and input $\mathbf{x}$, it holds that $\sigma_j(\mathbf{x}) \neq \sigma_{j'}(\mathbf{x})$.

This assumption makes the selected task identifiable from the emitted token and its prefix. We note that Assumption 4.2 may also be satisfied by expanding the vocabulary. Specifically, one can consider an expanded vocabulary of the form $\mathcal{V} \times [J]$ containing tuples of the form (v, j) which signify token v given by task σ_j.

5 Main Results: Learning Compositions With RLVR

To understand when RLVR can recover an autoregressive composition without intermediate supervision, we analyze the learning signal induced at a single step of the CoT. Since each task proposes a next token $\sigma_j(\mathbf{x}_{s-1})$, fine-tuning effectively adjusts the probability with which different tasks are selected at each step. The central question is therefore: *under what conditions does RLVR increase the probability of selecting the correct tasks?*

The answer turns out to depend only on how choosing a given task at step s affects the probability that the final rollout is correct. We formalize this dependence using the *task-advantage ratio*.

Definition 5.1 (task-advantage ratio)**.** For any parameters θ, step s, and task σ_j, we define the *task-advantage ratio*[2]

$$\rho^\theta_{s,j} := \frac{\mathbb{P}_\theta\left(V=1 \mid A_{s,j}\right)}{\mathbb{P}_\theta\left(V=1 \mid A^c_{s,j}\right)}, \quad \text{and} \quad \rho^{(t)}_{s,j} := \rho^{\theta^{(t)}}_{s,j} \;\; \forall t,$$

where $A_{s,j}$ denotes the event that task σ_j is selected at step s, and $A^c_{s,j}$ its complement (the event σ_j is not selected).

Intuitively, the task-advantage ratio measures whether choosing a task at a given step makes verified success more or less likely than avoiding it. This is closely related to the standard RL advantage function, and $\rho^\theta_{s,j} > 1$ precisely when the task has positive (prefix-averaged) advantage. Importantly, $\rho^\theta_{s,j}$ is defined with respect to the model's trajectory distribution: the success probability conditioned on selecting σ_j at step s is averaged over all prefixes reaching step s and all subsequent choices under the current policy. The following theorem shows that the task-advantage ratio determines whether RLVR increases or decreases the logit associated with each task in expectation in our setting.

Theorem 5.2. *Under Assumptions 4.1-4.2, for any CoT step $s \in [S]$, time $t \in \mathbb{N} \cup \{0\}$, task $j \in [J]$ and new sample $\mathbf{x}_{s-1}$ that contains a prompt plus $s-1$ additional CoT steps, if $\rho^{(t)}_{s,j} > 0$, the expectation of the logit updates (with respect to the training batch $\mathcal{B}_t$) at token $\sigma_j(\mathbf{x}_{s-1})$ is given by*

$$\mathbb{E}_{\mathcal{B}_t}\left[(f_{\theta^{(t+1)}}(\mathbf{x}_{s-1}) - f_{\theta^{(t)}}(\mathbf{x}_{s-1}))[\sigma_j(\mathbf{x}_{s-1})]\right] = \eta\gamma^2 \mathbb{P}_t(A_{s,j} \mid V=1)(1 - \mathbb{P}_t(A_{s,j}))\left(1 - \frac{1}{\rho^{(t)}_{s,j}}\right).$$

[2]The task-advantage ratio may be ∞, in which case we use the convention that $1/\rho^\theta_{s,j} = 0$. Likewise, if it is 0 then $1/\rho^\theta_{s,j} = \infty$. This is purely for notational convenience and does not impact the results in any way.

Otherwise, if $\rho_{s,j}^{(t)} = 0$*, then the expected value is equal to* $-\eta\gamma^2 \mathbb{P}_t\left(A_{s,j}\right)$.

Theorem 5.2 shows that the expected learning signal at each step decomposes into a few interpretable factors. The first is a directional term $1 - 1/\rho_{s,j}^{(t)}$, whose sign determines if the expected logit update for task σ_j is positive or negative. In particular, if choosing task σ_j increases the probability of verification success, then $\rho_{s,j}^{(t)} > 1$ and the expected update favors that task; if it decreases success probability, then $\rho_{s,j}^{(t)} < 1$ and the update suppresses it. The other important term in Theorem 5.2 is a "magnitude" term $\mathbb{P}_t(A_{s,j} \mid V = 1)\big(1 - \mathbb{P}_t(A_{s,j})\big)$, which determines how fast the logits change. Taken together, a task-advantage ratio bigger than 1 is both necessary and sufficient for a positive logit update for a certain task at step s, *in expectation over the training batch*.

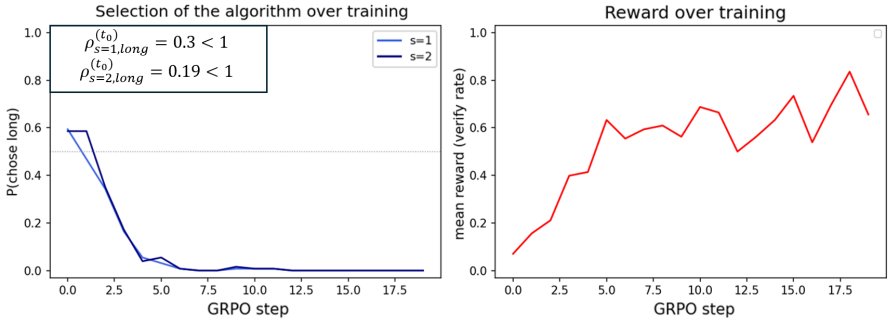


Figure 2: Compositional task of adding three large integers, $(a+b)+c$, using Qwen2.5-1.5B-Instruct. The task consists of two stages, each stage is a two integer addition that can be performed either through long addition or by directly outputting the sum. The empirically estimated task advantage ratios for long addition are 0.3 and 0.19 at the two stages, respectively, both below 1, while the corresponding ratios for direct addition exceed 1. Consistent with the theoretical prediction, although the base model favors long addition at both stages, its selection probability decreases during GRPO training as the mean reward increases. Experimental details are provided in Appendix G.

5.1 Successfully Learning Compositions

Theorem 5.2 shows that a task receives a positive expected logit update precisely when its task-advantage ratio exceeds 1. This local characterization alone does not guarantee finite-time convergence to the target composition. We therefore consider a condition requiring a uniform advantage for correct tasks and a disadvantage for incorrect tasks, under which we establish convergence with finite batches and an explicit rate.

Assumption 5.3 (Uniform task-advantage)**.** Let $\theta^{(0)}$ be the initial policy parameters, and let $\Theta_+ = \{\theta : \forall s \in [S],\ \mathbb{P}_\theta(A_{s,\tau(s)}) \geq \mathbb{P}_{\theta^{(0)}}(A_{s,\tau(s)})\}$. There exists a constant $\alpha > 0$ such that for any step $s \in [S]$, any task $j \in [J]$, and any parameter vector $\theta \in \Theta_+$

$$\begin{cases} \rho_{s,j}^{\theta} < 1, & j \neq \tau(s) \\ \rho_{s,j}^{\theta} \geq 1 + \alpha, & j = \tau(s) \end{cases}.$$

Assumption 5.3 states that the correct task at each step has a uniform advantage over all alternatives, quantified by a margin $\alpha > 0$. This margin induces a consistent gap in the expected updates, ensuring that correct tasks are reinforced while incorrect ones are suppressed throughout training. In principle,

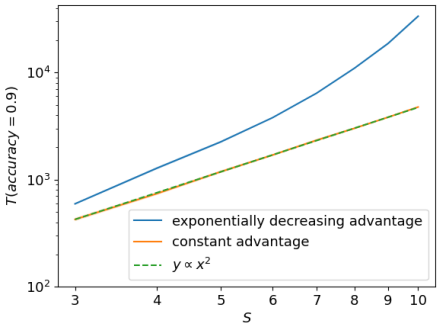


Figure 3: Iteration complexity needed to reach an accuracy of 0.9 as a function of the composition length (log-log scale). Problems with an inductive structure have a constant task advantage, and the iteration complexity scales like S^2. Without such a structure (e.g., solving parities), the advantage may decrease exponentially in S, and the iteration complexity becomes exponentially large.

it suffices for the condition to hold along the training trajectory alone. We state it over Θ_+, where each correct-task probability is at least its initial value, so that the hypothesis is independent of the particular random training trajectory.

An advantage greater than 1 is necessary for a positive expected logit update, and the strengthening with a uniform margin allows us to bound the convergence rate. Later in Section 6, we demonstrate how RLVR can fail when Assumption 5.3 is not satisfied.

We now show how, under a uniform task advantage, RLVR can indeed converge to a solution that passes the verifier with high probability. The proof is deferred to the appendix.

Theorem 5.4. *Under assumptions 4.1-4.2 and Assumption 5.3, for any $\epsilon \in (0, 1/2), \delta \in (0, 1)$, let $p_0 := \min_{s\in[S]} \mathbb{P}_0\left(A_{s,\tau(s)}\right)$, and let the batch size B and learning rate η be as specified in Appendix A.4. There exists an absolute constant $C > 0$, such that with probability at least $1 - \delta$, at time $T := \left\lceil \frac{C\left(1+\frac{1}{\alpha}\right)S^2 \log(S/\min(\epsilon,p_0))}{\min(\epsilon,p_0)^2} \right\rceil$,*

$$\mathbb{P}_T\left(V(\mathbf{x}_S) = 1\right) \geq 1 - \epsilon.$$

We highlight a few implications of the result. First, the convergence rate depends on the initial probabilities of selecting the correct tasks, reflecting the empirical observation that skills weakly represented during pre-training are substantially harder to amplify through post-training [Yue et al., 2025, Wu et al., 2025]. Second, the dependence of T on the composition length S is quadratic rather than exponential, indicating that under a uniform task-advantage condition, RLVR can scale to moderately long chains of thought efficiently.

Lastly, as we will further discuss in Section 7, the inductive problem structure influences the complexity of learning. Specifically, time to convergence scales with α as $1 + 1/\alpha$. So when α is large, convergence is very quick. By contrast, if α is very close to 0 then convergence is slow (c.f. Figure 3). In Section 7 we will demonstrate how the inductive structure of certain problems influences the parameter α.

6 When Can RLVR Go Wrong?

One may wonder if RLVR is always expected to succeed in our setting. We now provide an example in which learning fails because Assumption 5.3 does not hold. In fact, surprisingly simple constructions can already provably converge to poor local minima if the base model is poor, and Assumption 5.3 is violated.

To see this, let $\mathcal{V} := \{1, 2\}$, and let the prompt $\mathbf{x}_0$ be 1 with probability $1/2$, and 2 with probability $1/2$. We consider tasks that are simply constant functions, such that $\sigma_1(\mathbf{x}) := 1$ and $\sigma_2(\mathbf{x}) := 2$ for all $\mathbf{x}$ of any length. The verifier is defined as follows:

$$V(\mathbf{x}_S) = \begin{cases} 1 & \mathbf{x}_S \in \{(1,1,1),(2,1,1),(1,2,2)\} \\ 0 & \text{Otherwise.} \end{cases}$$

Equivalently, one may view V as a stochastic verifier that accepts $(1, 1)$ always and $(2, 2)$ with probability $1/2$, with its randomness encoded by the prompt token $\mathbf{x}_0$. As such, the ideal target function $f^\star(\mathbf{x})$ is given by applying σ_1 at steps 1 and 2 (so $\tau(1) = \tau(2) = 1$). Even though the model may always achieve a perfect reward if it chooses task 1, it may also receive a positive reward half the time if it consistently selects the opposite task. We show that the model can easily converge to this local minimum and get stuck.

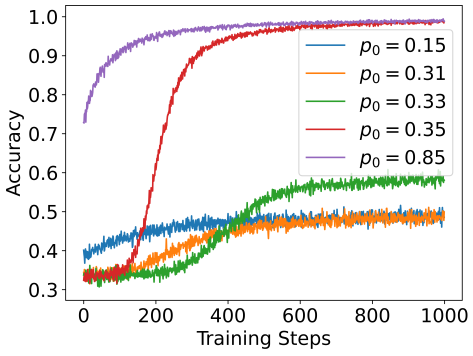


Figure 4: We consider a base model whose "quality" is determined by p_0, the probability of picking the correct tasks at initialization. Within the setting defined in Sec. 6, the model converges to a suboptimal solution whenever $p_0 < \frac{1}{3}$ (with $\Pr_t(V = 1) \to 1/2$) but to the optimal solution whenever $p_0 > 1/3$. This is aligned with the task-advantage ratio $\rho_{s,1}$ which we show is > 1 when $p_0 > 1/3$ and < 1 when $p_0 < 1/3$. Samples are drawn i.i.d. with a batch size of 256, and the accuracy is computed with respect to the last batch.

Consider initial parameters such that for some $p_0 \in (0,1)$, for any $\mathbf{x}_{s-1}$, the probability of picking the correct task is given by p_0. The quality of the base model is determined by p_0, where a larger p_0 indicates a better base model. We now show how p_0 impacts the performance of RLVR.

Proposition 6.1. *In the setting above, consider initial parameters such that for some $p_0 \in (0,1)$, for any $\mathbf{x}_{s-1}$, the probability of picking the correct task (task σ_1) is given by p_0. Then for any step size $\eta > 0$,*

$$\lim_{t\to\infty}\lim_{B\to\infty} \mathbb{P}_{\theta^{(t)}}\left(V(\mathbf{x}_S)=1\right) = \begin{cases} \frac{1}{2} & \text{if } \ p_0 < \frac{1}{3} \\ 1 & \text{if } \ p_0 > \frac{1}{3}. \end{cases}$$

The proof is presented in Appendix B, and is based on showing that the task-advantage ratio for the correct task is bigger than 1 throughout training if $p_0 > 1/3$, and conversely is smaller than 1 throughout training if $p_0 < 1/3$. As such, in this setting, the task-advantage ratio depends on the quality of the base model. When $p_0 < 1/3$, the model converges to a suboptimal solution, contrasting the positive results of Theorem 5.4. We note that in this case, Assumption 5.3 does not hold. See Figure 4 for a demonstration of how small differences in p_0 impact the trajectory of the weights.

This may serve as an explanation for why RLVR is often observed in practice to be highly sensitive to the base model [Yue et al., 2025]. If the base model is poor, choosing the correct task at any step may not present a clear advantage in terms of final reward. When the model can receive positive rewards on suboptimal CoTs that lead to the correct solution some of the time, the RLVR dynamics may increase the probability of sampling these suboptimal CoTs instead of sampling the correct one.

7 The Inductive Structure of Various Problems

In the above, we showed that compositional problems that satisfy Assumption 5.3 are learnable without any intermediate supervision. We now demonstrate how several different problem structures impact the task-advantage ratio, and thus the convergence rate: Under an inductive structure that incentivizes partially correct reasoning paths, RLVR can learn with an iteration complexity that scales at most quadratically with the composition length. In contrast, without such a structure, a compositional problem may still be learned without supervision, but *may* require an exponential time complexity to learn. See Figure 3 for a comparison of these two problems.

No inductive structure - a case study with parities. Bit parities are a canonical example of problems that are difficult to learn [Blum et al., 1994, Kearns, 1998, Abbe and Boix-Adsera, 2022, Shoshani and Shamir, 2025], and have therefore gained much attention in the literature as a testing bed to compare various algorithms [Barak et al., 2022, Kim and Suzuki, 2024, Wen et al., 2024, Kou et al., 2024, Abedsoltan et al., 2025]. Specifically, given inputs on the cube $\{-1,1\}^d$, and an *unknown* subset of indices $P \subseteq \{1,\ldots,d\}$, the goal is to learn the function $f^\star(\mathbf{x}) := \prod_{i\in P} x_i$.

There are many ways to view the parities as a compositional task. To best match our setting, we consider the case where the string is "scanned" from left to right, and each bit is either selected to contribute to the parity or ignored. See Fig. 1 for a visual depiction, and Appendix C for a formal definition. The model, as before, is given by (1). We now characterize the task-advantage ratio for such a problem.

Proposition 7.1. *Under the above setting, suppose that the base model yields a uniform distribution across all tasks (meaning that for any s, $\langle \mathbf{h}_{s,j}, \theta_0\rangle$ is a constant across j's). Then assumption 5.3 holds with*

$$\alpha = \frac{1}{2^{d-1}}.$$

The proof can be found in appendix C. As can be seen, the advantage is exponentially decreasing with d. This is not a limitation of the framework, but is an inherent property of learning parities [Abbe and Boix-Adsera, 2022, Shoshani and Shamir, 2025]. The exponential rate is due to a combination of factors. First, incorrect compositions lead to the correct solution half the time (meaning they are uncorrelated with the correct composition). Moreover, the advantage of choosing a correct hidden bit is manifested only if all other hidden bits are chosen in the rest of the CoT.

One may wonder if such exponential rates naturally arise in practice. This is unlikely, and when using LLMs to solve reasoning problems, the situation is usually different. Typically, incorrect reasoning

steps decrease the likelihood of a correct final solution. Moreover, in many problems, an LLM may increase the chance of a correct final solution even if it takes a few wrong steps along the CoT [Yu et al., 2025]. We will analyze next a structure of compositions that we believe better reflects the properties of problems where LLMs are used.

Inductive structure - when imperfect task selection is still meaningful. To better model standard reasoning tasks, we consider a case where each "correct" task that is chosen by the policy increases the probability that the verifier accepts. This is meant to model the fact that partially correct CoTs may still help LLMs reach a correct solution [Yu et al., 2025]. Note that unlike the parity problem considered earlier, we will show that each correct task chosen provides an advantage.

To capture this setting, we consider a simple model in which each incorrect step reduces the probability of final success by a factor $\lambda_s \in (0, 1)$, independently across steps. Intuitively, λ_s represents the probability of recovering from an incorrect decision at step s, so that partially incorrect reasoning paths may still lead to a correct final answer. In this setting, Assumption 5.3 holds with a constant margin that does not vanish as the composition length increases. See Appendix D for the full setting.

Proposition 7.2. *Under the above setting, assumption 5.3 holds with:*

$$\alpha = \min_{s \in [S]} \frac{1 - \lambda_s}{\lambda_s}.$$

The proof is in appendix D. We see here that α can remain bounded by a constant as S increases. Consequently, by our main result, RLVR can learn this composition with an iteration complexity that scales quadratically with S. This demonstrates the strength of RLVR for reasoning problems, where LLMs can succeed even when initially making incorrect reasoning steps.

RLVR learns this problem quickly because the chance that an incorrect composition succeeds on a given problem drops exponentially with the number of incorrect components. As such, there is a clear advantage to picking the correct task at each intermediate step. This is an important difference from the parity problem, where all incorrect compositions can pass the verifier with the same probability, $1/2$, so acceptance does not meaningfully disambiguate correct intermediate choices.

8 Discussion and Limitations

We introduced a tractable abstraction for learning task composition with autoregressive models. At each CoT step, the model selects among a finite set of deterministic skills that propose the next token, and RLVR reallocates probability mass across these skills. Within this model, we derived a characterization of the step-wise learning signal induced by training on verified rollouts.

Our main conclusion is that RLVR does not inherently recover a ground-truth chain of thought; it reinforces intermediate decisions only insofar as they increase the probability of eventual verification success. This dependence is captured by the task-advantage ratio. When correct steps enjoy a uniform advantage (within the relevant policy region), RLVR provably converges to the correct composition with polynomial dependence on chain length; when this structural advantage is absent, RLVR can converge to suboptimal compositions even without representational or statistical limitations, and the outcome can hinge on base-model quality.

While the paper focuses on a theoretical task-selection model, it would be interesting to understand if similar conclusions hold in real LLMs. Specifically, can the task-advantage be measured, predicted from base-model behavior, and controlled via training design (e.g., prompts, verifiers, sampling, or objectives) so as to reliably steer RLVR toward correct intermediate computation?

Acknowledgements

Research was supported in part by the Israeli Council for Higher Education (CHE) via the Weizmann Data Science Research Center, by the MBZUAI-WIS Joint Program for Artificial Intelligence Research, and by research grants from the Estates of Tully and Michele Plesser and the Anita James Rosen and Harry Schutzman Foundations, and in part by the ERC (European Research Council) and the ISF (Israel Science Foundation).

References

Emmanuel Abbe and Enric Boix-Adsera. On the non-universality of deep learning: quantifying the cost of symmetry. *Advances in Neural Information Processing Systems*, 35:17188–17201, 2022.

Emmanuel Abbe, Samy Bengio, Aryo Lotfi, Colin Sandon, and Omid Saremi. How far can transformers reason? the globality barrier and inductive scratchpad. *Advances in Neural Information Processing Systems*, 37:27850–27895, 2024.

Amirhesam Abedsoltan, Huaqing Zhang, Kaiyue Wen, Hongzhou Lin, Jingzhao Zhang, and Mikhail Belkin. Task generalization with autoregressive compositional structure: Can learning from D tasks generalize to D^T tasks? *arXiv preprint arXiv:2502.08991*, 2025.

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International conference on machine learning*, pages 242–252. PMLR, 2019.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019.

Boaz Barak, Benjamin Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit. *Advances in Neural Information Processing Systems*, 35:21750–21764, 2022.

Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning dnf and characterizing statistical query learning using fourier analysis. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 253–262, 1994.

Dylan J Foster, Zakaria Mhammedi, and Dhruv Rohatgi. Is a good foundation necessary for efficient reinforcement learning? the computational role of the base model in exploration. *arXiv preprint arXiv:2503.07453*, 2025.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

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

Uijeong Jang, Jason D Lee, and Ernest K Ryu. Lora training in the ntk regime has no spurious local minima. *arXiv preprint arXiv:2402.11867*, 2024.

Zeyu Jia, Alexander Rakhlin, and Tengyang Xie. Do we need to verify step by step? rethinking process supervision from a theoretical perspective. *arXiv preprint arXiv:2502.10581*, 2025.

Nirmit Joshi, Gal Vardi, Adam Block, Surbhi Goel, Zhiyuan Li, Theodor Misiakiewicz, and Nathan Srebro. A theory of learning with autoregressive chain of thought. *arXiv preprint arXiv:2503.07932*, 2025.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

Juno Kim and Taiji Suzuki. Transformers provably solve parity efficiently with chain of thought. *arXiv preprint arXiv:2410.08633*, 2024.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Yiwen Kou, Zixiang Chen, Quanquan Gu, and Sham Kakade. Matching the statistical query lower bound for k-sparse parity problems with sign stochastic gradient descent. *Advances in Neural Information Processing Systems*, 37:113001–113037, 2024.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019.

Eran Malach. Auto-regressive next-token predictors are universal learners. *arXiv preprint arXiv:2309.06979*, 2023.

William Merrill and Ashish Sabharwal. The expressive power of transformers with chain of thought. In *The Twelfth International Conference on Learning Representations*, 2024.

Parsa Mirtaheri, Ezra Edelman, Samy Jelassi, Eran Malach, and Enric Boix-Adsera. Let me think! a long chain-of-thought can be worth exponentially many short ones. *arXiv preprint arXiv:2505.21825*, 2025.

Clayton Sanford, Bahare Fatemi, Ethan Hall, Anton Tsitsulin, Mehran Kazemi, Jonathan Halcrow, Bryan Perozzi, and Vahab Mirrokni. Understanding transformer reasoning capabilities via graph algorithms. *Advances in Neural Information Processing Systems*, 37:78320–78370, 2024.

Shai Shalev-Shwartz and Amnon Shashua. From reasoning to super-intelligence: A search-theoretic perspective. *arXiv preprint arXiv:2507.15865*, 2025.

Itamar Shoshani and Ohad Shamir. Hardness of learning fixed parities with neural networks. *arXiv preprint arXiv:2501.00817*, 2025.

Lena Strobl, William Merrill, Gail Weiss, David Chiang, and Dana Angluin. What formal languages can transformers express? a survey. *Transactions of the Association for Computational Linguistics*, 12:543–561, 2024.

Nikolaos Tsilivis, Eran Malach, Karen Ullrich, and Julia Kempe. How reinforcement learning after next-token prediction facilitates learning. In *International Conference on Learning Representations*, 2026.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, et al. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*, 2025.

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

Kaiyue Wen, Huaqing Zhang, Hongzhou Lin, and Jingzhao Zhang. From sparse dependence to sparse attention: unveiling how chain-of-thought enhances transformer sample efficiency. *arXiv preprint arXiv:2410.05459*, 2024.

Noam Wies, Yoav Levine, and Amnon Shashua. Sub-task decomposition enables learning in sequence to sequence tasks. *arXiv preprint arXiv:2204.02892*, 2022.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256, 1992.

Fang Wu, Weihao Xuan, Ximing Lu, Mingjie Liu, Yi Dong, Zaid Harchaoui, and Yejin Choi. The invisible leash: Why rlvr may or may not escape its origin. *arXiv preprint arXiv:2507.14843*, 2025.

Chenxiao Yang, Zhiyuan Li, and David Wipf. Chain-of-thought provably enables learning the (otherwise) unlearnable. In *The Thirteenth International Conference on Learning Representations*, 2025.

Gilad Yehudai, Noah Amsel, and Joan Bruna. Compositional reasoning with transformers, rnns, and chain of thought. *arXiv preprint arXiv:2503.01544*, 2025.

Zhouliang Yu, Ruotian Peng, Keyi Ding, Yizhe Li, Zhongyuan Peng, Minghao Liu, Yifan Zhang, Zheng Yuan, Huajian Xin, Wenhao Huang, et al. Formalmath: Benchmarking formal mathematical reasoning of large language models. *arXiv preprint arXiv:2505.02735*, 2025.

Lifan Yuan, Weize Chen, Yuchen Zhang, Ganqu Cui, Hanbin Wang, Ziming You, Ning Ding, Zhiyuan Liu, Maosong Sun, and Hao Peng. From $f(x)$ and $g(x)$ to $f(g(x))$: Llms learn new skills in rl by composing old ones. *arXiv preprint arXiv:2509.25123*, 2025.

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

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

A Proofs for Section 5

Algorithm 1 Outcome-based post-training on verified trajectories

Require: Initial parameters $\theta^{(0)}$, prompt distribution $\mathcal{D}$, verifier V
Require: Completion length S, positive batch size B, learning rate η, iterations T

for $t = 0, \dots, T-1$ **do**
  for $b = 1, \dots, B$ **do**
    repeat
      Sample a fresh prompt $\mathbf{x}_0^{(t,b)} \sim \mathcal{D}$.
      for $s = 1, \dots, S$ **do**
        Sample $y_s^{(t,b)} \sim \boldsymbol{\pi}_t(\cdot \mid \mathbf{x}_{s-1}^{(t,b)})$.
        Set $\mathbf{x}_s^{(t,b)} = (\mathbf{x}_{s-1}^{(t,b)}, y_s^{(t,b)})$.
      end for
    until $V(\mathbf{x}_S^{(t,b)}) = 1$
  end for
  Update $\theta^{(t+1)} \leftarrow \theta^{(t)} + \dfrac{\eta}{B} \displaystyle\sum_{b=1}^{B} \sum_{s=1}^{S} \nabla \log \boldsymbol{\pi}_t\big(y_s^{(t,b)} \mid \mathbf{x}_{s-1}^{(t,b)}\big)$.
end for
return $\theta^{(T)}$

A.1 Additional Notation

For any $s \in [S]$, $t \in \mathbb{N} \cup \{0\}$, token $v \in \mathcal{V}$ and new sample $\mathbf{x}_{s-1}$ that contains a prompt plus $s-1$ additional CoT steps, we define the difference in logits between the "correct tokens" $\sigma_{\tau(s)}$ and v as

$$\Delta_{s,v}^{(t)}(\mathbf{x}_{s-1}) := \left(f_{\theta^{(t+1)}}(\mathbf{x}_{s-1}) - f_{\theta^{(t)}}(\mathbf{x}_{s-1})\right)[\sigma_{\tau(s)}(\mathbf{x}_{s-1})] - \left(f_{\theta^{(t+1)}}(\mathbf{x}_{s-1}) - f_{\theta^{(t)}}(\mathbf{x}_{s-1})\right)[v]. \tag{2}$$

We also define the following quantity for any $j \in [J]$, which captures the empirical discrepancy, over the training batch, between selecting task σ_j at step s and its current policy probability.

$$Q_{s,j}^{(t)} := \frac{1}{B} \sum_{b=1}^{B} \left(\mathbf{1}_{\{y_s^{(t,b)} = \sigma_j(\mathbf{x}_{s-1}^{(t,b)})\}} - \boldsymbol{\pi}_t\left(\sigma_j(\mathbf{x}_{s-1}^{(t,b)}) \mid \mathbf{x}_{s-1}^{(t,b)}\right) \right). \tag{3}$$

For a policy $\boldsymbol{\pi}$ we denote by $\boldsymbol{\pi}(\mathbf{x}_{s-1}) \in \mathbb{R}^{|\mathcal{V}|}$ the vector with coordinates given by $\boldsymbol{\pi}(\mathbf{x}_{s-1})_v = \boldsymbol{\pi}(v \mid \mathbf{x}_{s-1})$.

Lastly, we denote the cross-entropy loss by

$$\mathcal{L}(\theta) := \mathbb{E}_{\mathbf{x}_0 \sim \mathcal{D}} \left[\log \frac{1}{\mathbb{P}_\theta\left(f^\star(\mathbf{x}_0) \mid \mathbf{x}_0\right)} \right].$$

Here, $\mathbb{P}_\theta(f^\star(\mathbf{x}_0) \mid \mathbf{x}_0)$ denotes the probability that the generated full string equals $f^\star(\mathbf{x}_0)$, conditional on the prompt.

A.2 Proof of Theorem 5.2

We begin by deriving explicit expressions for the per-step logit updates induced by a single parameter update.

Lemma A.1. *Under assumptions 4.1, 4.2, let $\mathbf{x}_{s-1}$ be some sample that contains a prompt plus an additional $s-1$ CoT tokens for some $s \in [S]$. Then for any $t \in \mathbb{N} \cup \{0\}$, the logit updates at time t*

are given by:

$$f_{\theta^{(t+1)}}(\mathbf{x}_{s-1}) = f_{\theta^{(t)}}(\mathbf{x}_{s-1}) + \frac{\eta}{B}\gamma \sum_{b=1}^{B} \left(\sum_{j:y_s^{(t,b)}=\sigma_j(\mathbf{x}_{s-1}^{(t,b)})} g_j(\mathbf{x}_{s-1}) - \sum_{j\in[J]} \boldsymbol{\pi}_t\left(\sigma_j(\mathbf{x}_{s-1}^{(t,b)}) \mid \mathbf{x}_{s-1}^{(t,b)}\right) g_j(\mathbf{x}_{s-1}) \right). \tag{4}$$

Proof. First, the parameter updates for the model (as defined in (1)) are given by

$$\begin{aligned}
\theta^{(t+1)} - \theta^{(t)} =& \frac{\eta}{B} \sum_{b=1}^{B} \sum_{s'=1}^{S} \nabla \log \boldsymbol{\pi}_t \left(y_{s'}^{(t,b)} \mid \mathbf{x}_{s'-1}^{(t,b)} \right) \\
=& \frac{\eta}{B} \sum_{b=1}^{B} \sum_{s'=1}^{S} \nabla_\theta f_{\theta^{(t)}} \left(\mathbf{x}_{s'-1}^{(t,b)} \right)^\top \left(\mathbf{e}_{y_{s'}^{(t,b)}} - \boldsymbol{\pi}_t \left(\mathbf{x}_{s'-1}^{(t,b)} \right) \right) \\
=& \frac{\eta}{B} \sum_{b=1}^{B} \sum_{s'=1}^{S} \sum_{j'=1}^{J} \left\langle g_{j'} \left(\mathbf{x}_{s'-1}^{(t,b)} \right), \mathbf{e}_{y_{s'}^{(t,b)}} - \boldsymbol{\pi}_t \left(\mathbf{x}_{s'-1}^{(t,b)} \right) \right\rangle \mathbf{h}_{s'-1,j'},
\end{aligned}$$

where $\boldsymbol{\pi}(\mathbf{x}) \in \mathbb{R}^{|\mathcal{V}|}$ denotes the vector with coordinates given by $\boldsymbol{\pi}(\mathbf{x})_v = \boldsymbol{\pi}(v \mid \mathbf{x})$.
As such, the difference in logits for a new input $\mathbf{x}_{s-1}$ (with $s-1$ CoT tokens) is given by

$$f_{\theta^{(t+1)}}(\mathbf{x}_{s-1}) = f_{\theta^{(t)}}(\mathbf{x}_{s-1}) + \frac{\eta}{B} \sum_{b=1}^{B} \sum_{s'=1}^{S} \sum_{j,j'\in[J]} g_j(\mathbf{x}_{s-1}) \left\langle g_{j'}(\mathbf{x}_{s'-1}^{(t,b)}), \mathbf{e}_{y_{s'}^{(t,b)}} - \boldsymbol{\pi}_t(\mathbf{x}_{s'-1}^{(t,b)}) \right\rangle \langle \mathbf{h}_{s'-1,j'}, \mathbf{h}_{s-1,j} \rangle .$$

Now, using that $\{\mathbf{h}_{s',j'}\}$ are orthonormal, only $s' = s$ and $j' = j$ survive, and we obtain

$$f_{\theta^{(t+1)}}(\mathbf{x}_{s-1}) = f_{\theta^{(t)}}(\mathbf{x}_{s-1}) + \frac{\eta}{B} \sum_{b=1}^{B} \sum_{j=1}^{J} g_j(\mathbf{x}_{s-1}) \left\langle g_j(\mathbf{x}_{s-1}^{(t,b)}), \mathbf{e}_{y_s^{(t,b)}} - \boldsymbol{\pi}_t(\mathbf{x}_{s-1}^{(t,b)}) \right\rangle .$$

Since $g_j(\mathbf{x}) = \gamma \mathbf{e}_{\sigma_j(\mathbf{x})}$, the inner product satisfies

$$\left\langle g_j(\mathbf{x}_{s-1}^{(t,b)}), \mathbf{e}_{y_s^{(t,b)}} - \boldsymbol{\pi}_t(\mathbf{x}_{s-1}^{(t,b)}) \right\rangle = \gamma \left(\delta_{\sigma_j(\mathbf{x}_{s-1}^{(t,b)}), y_s^{(t,b)}} - \boldsymbol{\pi}_t \left(\sigma_j(\mathbf{x}_{s-1}^{(t,b)}) \mid \mathbf{x}_{s-1}^{(t,b)} \right) \right)$$

So

$$f_{\theta^{(t+1)}}(\mathbf{x}_{s-1}) = f_{\theta^{(t)}}(\mathbf{x}_{s-1}) + \frac{\eta}{B}\gamma \sum_{b=1}^{B} \left(\sum_{j:y_s^{(t,b)}=\sigma_j(\mathbf{x}_{s-1}^{(t,b)})} g_j(\mathbf{x}_{s-1}) - \sum_{j\in[J]} \boldsymbol{\pi}_t\left(\sigma_j(\mathbf{x}_{s-1}^{(t,b)}) \mid \mathbf{x}_{s-1}^{(t,b)}\right) g_j(\mathbf{x}_{s-1}) \right).$$

□

Lemma A.2. *Under assumptions 4.1, 4.2, let $\mathbf{x}_{s-1}$ be some sample that contains a prompt plus an additional $s-1$ CoT tokens for some $s \in [S]$. Then for any $t \in \mathbb{N} \cup \{0\}$ and token $v \in \mathcal{V}$*

$$\frac{1}{\eta\gamma^2} \left(f_{\theta^{(t+1)}}(\mathbf{x}_{s-1}) - f_{\theta^{(t)}}(\mathbf{x}_{s-1}) \right)[v] = \sum_{j\,:\,\sigma_j(\mathbf{x}_{s-1})=v} Q_{s,j}^{(t)},$$

where $Q_{s,j}^{(t)}$ is defined in (3)*.*

Proof. Using Lemma A.1 and that $g_j(\mathbf{x}_{s-1}) = \gamma \mathbf{e}_{\sigma_j(\mathbf{x}_{s-1})}$, the logit updates for token v are given by

$$\begin{aligned}
&\frac{1}{\eta\gamma}\left(f_{\theta^{(t+1)}}(\mathbf{x}_{s-1}) - f_{\theta^{(t)}}(\mathbf{x}_{s-1})\right)[v] \\
=&\frac{1}{B}\sum_{b=1}^{B}\left(\sum_{j\,:\,y_s^{(t,b)}=\sigma_j(\mathbf{x}_{s-1}^{(t,b)})} g_j(\mathbf{x}_{s-1})[v] - \sum_{j\in[J]} \boldsymbol{\pi}_t\left(\sigma_j(\mathbf{x}_{s-1}^{(t,b)}) \mid \mathbf{x}_{s-1}^{(t,b)}\right) g_j(\mathbf{x}_{s-1})[v]\right) \\
=&\frac{\gamma}{B}\sum_{b=1}^{B}\sum_{j\,:\,\sigma_j(\mathbf{x}_{s-1})=v}\left(\mathbf{1}_{\{y_s^{(t,b)}=\sigma_j(\mathbf{x}_{s-1}^{(t,b)})\}} - \boldsymbol{\pi}_t\left(\sigma_j(\mathbf{x}_{s-1}^{(t,b)}) \mid \mathbf{x}_{s-1}^{(t,b)}\right)\right) \\
=&\gamma \sum_{j\,:\,\sigma_j(\mathbf{x}_{s-1})=v} Q_{s,j}^{(t)}.
\end{aligned}$$

□

Lemma A.3. *Under assumptions 4.1, 4.2, for any $j \in [J], s \in [S]$, $t \in \mathbb{N}\cup\{0\}$, and new sample $\mathbf{x}_{s-1}$ that contains a prompt plus $s-1$ additional CoT steps, it holds that $\boldsymbol{\pi}_t\left(\sigma_j(\mathbf{x}_{s-1}) \mid \mathbf{x}_{s-1}\right) = \mathbb{P}_t\left(A_{s,j}\right)$.*

Proof. For any s, j let $u_{s-1,j}^{(t)} := \langle \theta^{(t)}, \mathbf{h}_{s-1,j}\rangle$, such that for prefix $\mathbf{x}_{s-1}$ (that contains $s-1$ CoT steps),

$$f_{\theta^{(t)}}(\mathbf{x}_{s-1}) = \sum_{j\in[J]} u_{s-1,j}^{(t)} g_j(\mathbf{x}_{s-1}).$$

Furthermore, by Assumption 4.2, since $\sigma_j(\mathbf{x}_{s-1}) \neq \sigma_{j'}(\mathbf{x}_{s-1})$ for any $j \neq j'$ and $g_{j'}(\mathbf{x}_{s-1}) = \gamma \mathbf{e}_{\sigma_{j'}(\mathbf{x}_{s-1})}$, it follows that for any task σ_j,

$$f_{\theta^{(t)}}(\mathbf{x}_{s-1})[\sigma_j(\mathbf{x}_{s-1})] = \sum_{j'\in[J]} u_{s-1,j'}^{(t)} \cdot g_{j'}(\mathbf{x}_{s-1})[\sigma_j(\mathbf{x}_{s-1})] = \gamma u_{s-1,j}^{(t)}.$$

Furthermore, let $W_{s-1} \subseteq \mathcal{V}$ denote all tokens w such that $w \neq \sigma_j(\mathbf{x}_{s-1})$ for all $j \in [J]$. Note that by the above, for all $w \in W_{s-1}$, $f_{\theta^{(t)}}(\mathbf{x}_{s-1})[w] = 0$. As such, summing over all tokens $v \in \mathcal{V}$,

$$\begin{aligned}
\sum_{v\in\mathcal{V}} \exp\left(f_{\theta^{(t)}}(\mathbf{x}_{s-1})[v]\right) &= \sum_{j\in[J]} \exp\left(f_{\theta^{(t)}}(\mathbf{x}_{s-1})[\sigma_j(\mathbf{x}_{s-1})]\right) + \sum_{w\in W_{s-1}} \exp\left(f_{\theta^{(t)}}(\mathbf{x}_{s-1})[w]\right) \\
&= \sum_{j\in[J]} \exp\left(\gamma u_{s-1,j}^{(t)}\right) + |W_{s-1}|.
\end{aligned}$$

Notice that this is a constant that depends only on $s-1$ and t, but importantly not on $\mathbf{x}_{s-1}$. As such, denoting this constant by $Z_{t,s-1}$, we have

$$\boldsymbol{\pi}_t\left(\sigma_j(\mathbf{x}_{s-1}) \mid \mathbf{x}_{s-1}\right) = \frac{\exp\left(f_{\theta^{(t)}}(\mathbf{x}_{s-1})[\sigma_j(\mathbf{x}_{s-1})]\right)}{\sum_{v\in\mathcal{V}} \exp\left(f_{\theta^{(t)}}(\mathbf{x}_{s-1})[v]\right)} = \frac{\exp(\gamma u_{s-1,j}^{(t)})}{Z_{t,s-1}},$$

which again, does not depend on $\mathbf{x}_{s-1}$. By the definition of $\mathbb{P}_t$,

$$\mathbb{P}_t\left(A_{s,j} \mid \mathbf{x}_{s-1}\right) = \boldsymbol{\pi}_t\left(\sigma_j(\mathbf{x}_{s-1}) \mid \mathbf{x}_{s-1}\right),$$

and the claim follows since this is the same for all $\mathbf{x}_{s-1}$. □

Lemma A.4. *Under assumptions 4.1, 4.2, for any $j \in [J], s \in [S]$, $t \in \mathbb{N}\cup\{0\}$, and new sample $\mathbf{x}_{s-1}$ that contains a prompt plus $s-1$ additional CoT steps, it holds that*

$$\mathbb{E}_t\left[Q_{s,j}^{(t)} \mid V=1\right] = \mathbb{P}_t\left(A_{s,j} \mid V=1\right) - \mathbb{P}_t\left(A_{s,j}\right).$$

Proof. Using the definition of $Q_{s,j}^{(t)}$,

$$\mathbb{E}_t\left[Q_{s,j}^{(t)} \mid V=1\right] = \mathbb{E}_t\left[\mathbf{1}_{\{y_s=\sigma_j(\mathbf{x}_{s-1})\}} \mid V=1\right] - \mathbb{E}_t\left[\boldsymbol{\pi}_t\left(\sigma_j(\mathbf{x}_{s-1}) \mid \mathbf{x}_{s-1}\right) \mid V=1\right].$$

For the first term, we have by definition that

$$\mathbb{E}_t\left[\mathbf{1}_{\{y_s=\sigma_j(\mathbf{x}_{s-1})\}} \mid V=1\right] = \mathbb{P}_t\left(A_{s,j} \mid V=1\right).$$

Since $\boldsymbol{\pi}_t\left(\sigma_j(\mathbf{x}_{s-1}) \mid \mathbf{x}_{s-1}\right) = \mathbb{P}_t\left(A_{s,j} \mid \mathbf{x}_{s-1}\right)$ by definition, we have that

$$\begin{aligned}\mathbb{E}_t\left[\boldsymbol{\pi}_t\left(\sigma_j(\mathbf{x}_{s-1}) \mid \mathbf{x}_{s-1}\right) \mid V=1\right] =_{(*)} &\mathbb{P}_t\left(A_{s,j}\right)\mathbb{E}_t\left[1 \mid V=1\right] \\ =&\mathbb{P}_t\left(A_{s,j}\right),\end{aligned}$$

where (*) follows from Lemma A.3. □

We now express the expected value of $Q_{s,j}^{(t)}$ in a form that separates the intrinsic advantage of task σ_j at step s from the probability of selecting it.

Lemma A.5. *Under assumptions 4.1, 4.2, for any $j \in [J]$, $s \in [S]$, $t \in \mathbb{N} \cup \{0\}$, and new sample $\mathbf{x}_{s-1}$ that contains a prompt plus $s-1$ additional CoT steps, if $\rho_{s,j}^{(t)} > 0$, then*

$$\mathbb{E}_t\left[Q_{s,j}^{(t)} \mid V=1\right] = \mathbb{P}_t\left(A_{s,j} \mid V=1\right)\left(1-\mathbb{P}_t\left(A_{s,j}\right)\right)\left(1-\frac{1}{\rho_{s,j}^{(t)}}\right).$$

Otherwise, if $\rho_{s,j}^{(t)} = 0$, then $\mathbb{E}_t\left[Q_{s,j}^{(t)} \mid V=1\right] = -\mathbb{P}_t\left(A_{s,j}\right)$.

Proof. First, for any j, Bayes' law gives

$$\mathbb{P}_t\left(A_{s,j}\right)\mathbb{P}_t\left(V=1 \mid A_{s,j}\right) = \mathbb{P}_t\left(A_{s,j} \mid V=1\right)\mathbb{P}_t\left(V=1\right). \tag{5}$$

Note that since softmax assigns non-zero probabilities, it always holds that $\mathbb{P}_t\left(A_{s,j}\right) > 0$ and $\mathbb{P}_t\left(V=1\right) > 0$. Therefore, if $\mathbb{P}_t\left(V=1 \mid A_{s,j}\right) = 0$ then also $\mathbb{P}_t\left(A_{s,j} \mid V=1\right) = 0$, in which case Lemma A.4 implies

$$\mathbb{E}_t\left[Q_{s,j}^{(t)} \mid V=1\right] = -\mathbb{P}_t\left(A_{s,j}\right).$$

Otherwise, from now on we assume $\mathbb{P}_t\left(A_{s,j} \mid V=1\right) > 0$. As such, by Lemma A.4 and (5),

$$\begin{aligned}\mathbb{E}_t\left[Q_{s,j}^{(t)} \mid V=1\right] &= \mathbb{P}_t\left(A_{s,j} \mid V=1\right) - \mathbb{P}_t\left(A_{s,j}\right) \\ &= \mathbb{P}_t\left(A_{s,j} \mid V=1\right)\left(1-\frac{\mathbb{P}_t\left(V=1\right)}{\mathbb{P}_t\left(V=1 \mid A_{s,j}\right)}\right).\end{aligned} \tag{6}$$

Furthermore, by the law of total probability, it holds that

$$\begin{aligned}\mathbb{P}_t\left(V=1\right) = &\mathbb{P}_t\left(V=1 \mid A_{s,j}\right)\mathbb{P}_t\left(A_{s,j}\right) \\ &+ \mathbb{P}_t\left(V=1 \mid A_{s,j}^c\right)\mathbb{P}_t\left(A_{s,j}^c\right).\end{aligned}$$

Using the definition of $\rho_{s,j}^{(t)}$, the above equation implies that

$$\begin{aligned}1-\frac{\mathbb{P}_t\left(V=1\right)}{\mathbb{P}_t\left(V=1 \mid A_{s,j}\right)} =& 1-\mathbb{P}_t\left(A_{s,j}\right) - \frac{1}{\rho_{s,j}^{(t)}}\left(1-\mathbb{P}_t\left(A_{s,j}\right)\right) \\ =& \left(1-\mathbb{P}_t\left(A_{s,j}\right)\right)\left(1-\frac{1}{\rho_{s,j}^{(t)}}\right).\end{aligned}$$

Plugging this back into (6), we obtain

$$\begin{aligned}\mathbb{E}_t\left[Q_{s,j}^{(t)} \mid V=1\right] &= \mathbb{P}_t\left(A_{s,j} \mid V=1\right)\left(1-\frac{\mathbb{P}_t\left(V=1\right)}{\mathbb{P}_t\left(V=1 \mid A_{s,j}\right)}\right) \\ &= \mathbb{P}_t\left(A_{s,j} \mid V=1\right)\left(1-\mathbb{P}_t\left(A_{s,j}\right)\right)\left(1-\frac{1}{\rho_{s,j}^{(t)}}\right),\end{aligned}$$

which is exactly what we needed to show. □

The above lemmas immediately imply Theorem 5.2:

Theorem A.6 (Theorem 5.2 from the main text)**.** *Under assumptions 4.1, 4.2, for any CoT step $s \in [S]$, time $t \in \mathbb{N} \cup \{0\}$, task $j \in [J]$ and new sample $\mathbf{x}_{s-1}$ that contains a prompt plus $s-1$ additional CoT steps, if $\rho_{s,j}^{(t)} > 0$, the expectation of the logit updates (with respect to the training samples) at token $\sigma_j(\mathbf{x}_{s-1})$ is given by*

$$\mathbb{E}_{\mathcal{B}_t}\left[(f_{\theta^{(t+1)}}(\mathbf{x}_{s-1}) - f_{\theta^{(t)}}(\mathbf{x}_{s-1}))[\sigma_j(\mathbf{x}_{s-1})]\right] = \eta\gamma^2 \mathbb{P}_t(A_{s,j} \mid V=1)(1-\mathbb{P}_t(A_{s,j}))\left(1 - \frac{1}{\rho_{s,j}^{(t)}}\right).$$

Otherwise, if $\rho_{s,j}^{(t)} = 0$, the expected value is equal to $-\eta\gamma^2 \mathbb{P}_t\left(A_{s,j}\right)$.

Proof. Using Lemma A.2 and taking the expectation, we get

$$\frac{1}{\eta\gamma^2}\mathbb{E}_{\mathcal{B}_t}\left[\left(f_{\theta^{(t+1)}}(\mathbf{x}_{s-1}) - f_{\theta^{(t)}}(\mathbf{x}_{s-1})\right)[\sigma_j(\mathbf{x}_{s-1})]\right] = \sum_{j' \,:\, \sigma_{j'}(\mathbf{x}_{s-1}) = \sigma_j(\mathbf{x}_{s-1})} \mathbb{E}_{\mathcal{B}_t}[Q_{s,j'}^{(t)}] = \mathbb{E}_{\mathcal{B}_t}[Q_{s,j}^{(t)}],$$

where the last equality uses Assumption 4.2. Since $\mathcal{B}_t$ are sampled from $\mathbb{P}_t$ conditioned on $V=1$, this is exactly $\mathbb{E}_t\left[Q_{s,j}^{(t)} \mid V=1\right]$. Applying Lemma A.5 completes the proof. □

A.3 Additional Lemmas for Theorem 5.4

We now lower bound the expected logit gap between the correct token $\sigma_{\tau(s)}(\mathbf{x}_{s-1})$ and any incorrect token. This bound is the key step that drives monotonic improvement of the correct-task probability.

Lemma A.7. *Under assumptions 4.1, 4.2 and Assumption 5.3, for any $s \in [S]$, $t \in \mathbb{N} \cup \{0\}$, new sample $\mathbf{x}_{s-1}$ that contains a prompt plus $s-1$ additional CoT steps, and token $v \neq \sigma_{\tau(s)}\left(\mathbf{x}_{s-1}\right)$, if $\theta^{(t)} \in \Theta_+$ then*

$$\mathbb{E}_t\left[\frac{1}{\eta\gamma^2}\Delta_{s,v}^{(t)}(\mathbf{x}_{s-1}) \mid V=1\right] \geq \left(\frac{\alpha}{1+\alpha}\right)\mathbb{P}_t\left(A_{s,\tau(s)}\right)\left(1-\mathbb{P}_t\left(A_{s,\tau(s)}\right)\right),$$

where $\Delta_{s,v}^{(t)}(\mathbf{x}_{s-1})$ is defined in (2)*.*

Proof. First, we claim that for any token $v \neq \sigma_{\tau(s)}\left(\mathbf{x}_{s-1}\right)$,

$$\mathbb{E}_t\left[\left(f_{\theta^{(t+1)}}(\mathbf{x}_{s-1}) - f_{\theta^{(t)}}(\mathbf{x}_{s-1})\right)[v] \mid V=1\right] \leq 0. \tag{7}$$

Indeed, if for all $j \in [J]$, $v \neq \sigma_j(\mathbf{x}_{s-1})$ then by definition of our model (1) and the fact that $g_j(\mathbf{x}) = \gamma \mathbf{e}_{\sigma_j(\mathbf{x})}$, it follows that the left hand side of (7) equals 0. Otherwise, if $\exists j \in [J]$ such that $v = \sigma_j(\mathbf{x}_{s-1})$, then by Theorem A.6, either $\rho_{s,j}^{(t)} = 0$ in which case (7) holds, or if $\rho_{s,j}^{(t)} > 0$ then

$$\begin{aligned}\mathbb{E}_t\left[\left(f_{\theta^{(t+1)}}(\mathbf{x}_{s-1}) - f_{\theta^{(t)}}(\mathbf{x}_{s-1})\right)[v] \mid V=1\right] &= \eta\gamma^2 \mathbb{P}_t\left(A_{s,j} \mid V=1\right)\left(1-\mathbb{P}_t\left(A_{s,j}\right)\right)\left(1 - \frac{1}{\rho_{s,j}^{(t)}}\right) \\ &\leq 0, \end{aligned} \tag{8}$$

where the last inequality follows from Assumption 5.3. So we have shown that (7) holds, and thus by the definition of $\Delta_{s,v}^{(t)}$ (Equation (2)), it follows that

$$\mathbb{E}_t\left[\frac{1}{\eta\gamma^2}\Delta_{s,v}^{(t)}(\mathbf{x}_{s-1}) \mid V=1\right] \geq \frac{1}{\eta\gamma^2}\mathbb{E}_t\left[\left(f_{\theta^{(t+1)}}(\mathbf{x}_{s-1}) - f_{\theta^{(t)}}(\mathbf{x}_{s-1})\right)[\sigma_{\tau(s)}(\mathbf{x}_{s-1})] \mid V=1\right].$$

Assumption 5.3 ensures that $\rho_{s,\tau(s)}^{(t)} \geq 1+\alpha > 1$, so by combining this with Theorem A.6 we obtain

$$\mathbb{E}_t\left[\frac{1}{\eta\gamma^2}\Delta_{s,v}^{(t)}(\mathbf{x}_{s-1}) \mid V=1\right] \geq \left(\frac{\alpha}{1+\alpha}\right)\mathbb{P}_t\left(A_{s,\tau(s)} \mid V=1\right)\left(1-\mathbb{P}_t\left(A_{s,\tau(s)}\right)\right). \tag{9}$$

Lastly, by Lemma A.5 and Assumption 5.3, $\mathbb{E}_t\left[Q^{(t)}_{s,\tau(s)} \mid V=1\right] \geq 0$. So by Lemma A.4, it holds that

$$\mathbb{P}_t\left(A_{s,\tau(s)} \mid V=1\right) \geq \mathbb{P}_t\left(A_{s,\tau(s)}\right),$$

As such, we can lower bound (9) by removing the conditioning on $V=1$, which completes the proof. □

To convert the expected logit improvements into high-probability guarantees, we next show that all quantities $Q^{(t)}_{s,j}$ (and hence all logit differences) concentrate uniformly over tasks and CoT steps within a single iteration.

Lemma A.8. *Under assumptions 4.1, 4.2 and Assumption 5.3, for any $t \in \mathbb{N} \cup \{0\}$ and $\delta \in (0,1)$, with probability at least $1-\delta$ (over the randomness of the training batch at iteration t), it holds simultaneously for all $s \in [S]$, all $j \in [J]$ that*

$$\left|Q^{(t)}_{s,j} - \mathbb{E}_t[Q^{(t)}_{s,j} \mid V=1]\right| \leq \sqrt{\frac{2\log\left(\frac{2SJ}{\delta}\right)}{B}}. \tag{10}$$

Moreover, with the same probability, it holds simultaneously for all $s \in [S]$, all $\mathbf{x}_{s-1}$ (containing $s-1$ CoT steps), and all tokens $v \in \mathcal{V}$ that

$$\frac{1}{\eta\gamma^2}\Delta^{(t)}_{s,v}(\mathbf{x}_{s-1}) \geq \mathbb{E}_t\left[\frac{1}{\eta\gamma^2}\Delta^{(t)}_{s,v}(\mathbf{x}_{s-1}) \mid V=1\right] - \sqrt{\frac{8\log\left(\frac{2SJ}{\delta}\right)}{B}}. \tag{11}$$

Proof. For fixed s, j, recall that

$$Q^{(t)}_{s,j} = \frac{1}{B}\sum_{b=1}^{B} Z^{(t,b)}_{s,j}, \qquad Z^{(t,b)}_{s,j} := \mathbf{1}_{\{y^{(t,b)}_s = \sigma_j(\mathbf{x}^{(t,b)}_{s-1})\}} - \boldsymbol{\pi}_t\left(\sigma_j(\mathbf{x}^{(t,b)}_{s-1}) \mid \mathbf{x}^{(t,b)}_{s-1}\right).$$

Each $Z^{(t,b)}_{s,j} \in [-1,1]$, and $\{Z^{(t,b)}_{s,j}\}_{b=1}^{B}$ are i.i.d. across b (where $\mathbf{x}^{(t,b)}_{s-1} \sim \mathbb{P}_t(\cdot \mid V=1)$). Hence by Hoeffding's inequality, for any $u \geq 0$,

$$\mathbb{P}_t\Big(\left|Q^{(t)}_{s,j} - \mathbb{E}_t[Q^{(t)}_{s,j} \mid V=1]\right| \geq u\Big) \leq 2\exp\left(-\frac{Bu^2}{2}\right).$$

Applying a union bound over all $s \in [S]$ and $j \in [J]$ and choosing $u = \sqrt{\frac{2\log\left(\frac{2SJ}{\delta}\right)}{B}}$ implies (10).

We now prove (11). Fix any $s \in [S]$ and prefix $\mathbf{x}_{s-1}$. By Lemma A.2, for any token $w \in \mathcal{V}$,

$$\frac{1}{\eta\gamma^2}\Big(f_{\theta^{(t+1)}}(\mathbf{x}_{s-1}) - f_{\theta^{(t)}}(\mathbf{x}_{s-1})\Big)[w] = \sum_{j:\sigma_j(\mathbf{x}_{s-1})=w} Q^{(t)}_{s,j}.$$

By Assumption 4.2, for each fixed $\mathbf{x}_{s-1}$ and token w, the set $\{j : \sigma_j(\mathbf{x}_{s-1}) = w\}$ has size at most 1. Therefore,

$$\frac{1}{\eta\gamma^2}\Delta^{(t)}_{s,v}(\mathbf{x}_{s-1}) = \sum_{j:\sigma_j(\mathbf{x}_{s-1})=\sigma_{\tau(s)}(\mathbf{x}_{s-1})} Q^{(t)}_{s,j} - \sum_{j:\sigma_j(\mathbf{x}_{s-1})=v} Q^{(t)}_{s,j}$$

is a difference of at most two $Q^{(t)}_{s,j}$ terms. Hence,

$$\frac{1}{\eta\gamma^2}\Delta^{(t)}_{s,v}(\mathbf{x}_{s-1}) \geq \mathbb{E}_t\left[\frac{1}{\eta\gamma^2}\Delta^{(t)}_{s,v}(\mathbf{x}_{s-1}) \mid V=1\right] - 2\sup_{s',j'}\left|Q^{(t)}_{s',j'} - \mathbb{E}_t[Q^{(t)}_{s',j'} \mid V=1]\right|,$$

and plugging in the bound from (10) yields (11). □

Lemma A.9. *Under assumptions 4.1, 4.2 and Assumption 5.3, for any $s \in [S]$ and $t \in \mathbb{N} \cup \{0\}$, if there exists some $\beta^{(t)}_s \in (-1,\infty)$ such that*

$$\inf_{\mathbf{x}_{s-1}} \inf_{v \neq \sigma_{\tau(s)}(\mathbf{x}_{s-1})} \Delta^{(t)}_{s,v}(\mathbf{x}_{s-1}) \geq \beta^{(t)}_s,$$

then for $R^{(t)}_s := \frac{1}{\mathbb{P}_t\left(A_{s,\tau(s)}\right)} - 1$ it holds that

$$R^{(t+1)}_s \leq \exp\left(-\beta^{(t)}_s\right) R^{(t)}_s \leq \frac{R^{(t)}_s}{1+\beta^{(t)}_s}.$$

Proof. For any $\mathbf{x}_{s-1}$, it holds that

$$\begin{aligned}
&\boldsymbol{\pi}_{t+1}\left(\sigma_{\tau(s)}\left(\mathbf{x}_{s-1}\right) \mid \mathbf{x}_{s-1}\right) \\
=&\frac{1}{1+\sum_{v \neq \sigma_{\tau(s)}(\mathbf{x}_{s-1})} \exp\left(f_{\theta^{(t+1)}}(\mathbf{x}_{s-1})[v]-f_{\theta^{(t+1)}}(\mathbf{x}_{s-1})[\sigma_{\tau(s)}\left(\mathbf{x}_{s-1}\right)]\right)} \\
=&\frac{1}{1+\sum_{v \neq \sigma_{\tau(s)}(\mathbf{x}_{s-1})} \exp\left(-\Delta_{s,v}^{(t)}(\mathbf{x}_{s-1})\right) \exp\left(f_{\theta^{(t)}}(\mathbf{x}_{s-1})[v]-f_{\theta^{(t)}}(\mathbf{x}_{s-1})[\sigma_{\tau(s)}\left(\mathbf{x}_{s-1}\right)]\right)} \\
\geq&\frac{1}{1+\exp\left(-\beta_s^{(t)}\right) \sum_{v \neq \sigma_{\tau(s)}(\mathbf{x}_{s-1})} \exp\left(f_{\theta^{(t)}}(\mathbf{x}_{s-1})[v]-f_{\theta^{(t)}}(\mathbf{x}_{s-1})[\sigma_{\tau(s)}\left(\mathbf{x}_{s-1}\right)]\right)},
\end{aligned}$$

Let $R_s^{(t)}(\mathbf{x}_{s-1}) := \frac{1}{\boldsymbol{\pi}_t\left(\sigma_{\tau(s)}(\mathbf{x}_{s-1})|\mathbf{x}_{s-1}\right)} - 1$. The above implies

$$\begin{aligned}
R_s^{(t+1)}(\mathbf{x}_{s-1}) \leq& \exp\left(-\beta_s^{(t)}\right) \sum_{v \neq \sigma_{\tau(s)}(\mathbf{x}_{s-1})} \exp\left(f_{\theta^{(t)}}(\mathbf{x}_{s-1})[v]-f_{\theta^{(t)}}(\mathbf{x}_{s-1})[\sigma_{\tau(s)}\left(\mathbf{x}_{s-1}\right)]\right) \\
=& \exp\left(-\beta_s^{(t)}\right) R_s^{(t)}(\mathbf{x}_{s-1}) \\
\leq& \frac{R_s^{(t)}(\mathbf{x}_{s-1})}{1+\beta_s^{(t)}},
\end{aligned}$$

where the last inequality uses that $\exp(-x) \leq \frac{1}{1+x}$ for any $x > -1$. Now using Lemma A.3, it holds for all $\mathbf{x}_{s-1}$ that $R_s^{(t)}(\mathbf{x}_{s-1}) = R_s^{(t)}$, which completes the proof. □

Lemma A.10. *Under Assumptions 4.1, 4.2, and Assumption 5.3, for any $t \in \mathbb{N} \cup \{0\}$, letting $R_s^{(t)} := \frac{1}{\mathbb{P}_t\left(A_{s,\tau(s)}\right)} - 1$, it holds that*

$$\mathbb{P}_t\left(V=1\right) \geq 1 - \sum_{s=1}^{S} R_s^{(t)}.$$

Proof. By Assumption 4.1, if all steps of the correct chain-of-thought are selected then the verifier accepts; that is, the event $\bigcap_{s=1}^{S} A_{s,\tau(s)}$ is contained in the event $V=1$. Therefore,

$$\begin{aligned}
\mathbb{P}_t(V=1) \geq \mathbb{P}_t\left(\bigcap_{s=1}^{S} A_{s,\tau(s)}\right) &= 1 - \mathbb{P}_t\left(\bigcup_{s=1}^{S} A_{s,\tau(s)}^c\right) \\
&\geq 1 - \sum_{s=1}^{S} \mathbb{P}_t\left(A_{s,\tau(s)}^c\right) = 1 - \sum_{s=1}^{S}\left(1 - \mathbb{P}_t\left(A_{s,\tau(s)}\right)\right).
\end{aligned}$$

Finally, since $\mathbb{P}_t(A_{s,\tau(s)}) \leq 1$, we have

$$1 - \mathbb{P}_t(A_{s,\tau(s)}) \leq \frac{1 - \mathbb{P}_t(A_{s,\tau(s)})}{\mathbb{P}_t(A_{s,\tau(s)})} = \frac{1}{\mathbb{P}_t(A_{s,\tau(s)})} - 1 = R_s^{(t)}.$$

Substituting yields $\mathbb{P}_t(V=1) \geq 1 - \sum_{s=1}^{S} R_s^{(t)}$. □

Lemma A.11. *Under assumptions 4.1, 4.2 and Assumption 5.3, for any $t \in \mathbb{N} \cup \{0\}$, let $\mathcal{L}^{(t)} = \mathbb{E}_{\mathbf{x}_0}\left[\log\left(\frac{1}{\mathbb{P}_t(f^\star(\mathbf{x}_0)|\mathbf{x}_0)}\right)\right]$ denote the cross entropy loss. Then for $R_s^{(t)} := \frac{1}{\mathbb{P}_t\left(A_{s,\tau(s)}\right)} - 1$ it holds that for all t,*

$$\mathcal{L}^{(t)} \leq \sum_{s=1}^{S} R_s^{(t)}.$$

Proof. Given a prompt $\mathbf{x}_0$ let $G_s(\mathbf{x}_0)$ (or just G_s where $\mathbf{x}_0$ is clear by context) denote the "correct" CoT for the first s steps, given by $(\mathbf{x}_0, y_1, \ldots, y_s)$, where for all $s' \leq s$, $y_{s'} = \sigma_{\tau(s')}(\mathbf{x}_{s'-1})$, and G_0 will just be $\mathbf{x}_0$ (note that by Assumption 4.2, $G_s(\mathbf{x}_0)$ is a function of $\mathbf{x}_0$). Then for any prompt $\mathbf{x}_0$, by the chain law of probability,

$$\mathbb{P}_t\left(f^\star(\mathbf{x}_0) \mid \mathbf{x}_0\right) \geq \prod_{s=1}^{S} \mathbb{P}_t\left(A_{s,\tau(s)} \mid G_{s-1}\right).$$

Since $\log(1/q)$ is decreasing in q, we have

$$\begin{aligned}
\mathcal{L}^{(t)} =& \mathbb{E}_{\mathbf{x}_0}\left[\log\left(\frac{1}{\mathbb{P}_t(f^\star(\mathbf{x}_0) \mid \mathbf{x}_0)}\right)\right] \leq \mathbb{E}_{\mathbf{x}_0}\left[\log\left(\frac{1}{\prod_{s=1}^{S}\mathbb{P}_t\left(A_{s,\tau(s)} \mid G_{s-1}\right)}\right)\right] \\
=& \sum_{s=1}^{S} \mathbb{E}_{\mathbf{x}_0}\left[\log\left(\frac{1}{\mathbb{P}_t\left(A_{s,\tau(s)} \mid G_{s-1}\right)}\right)\right] \\
\leq_{(*)}& \sum_{s=1}^{S} \mathbb{E}_{\mathbf{x}_0}\left[\frac{1}{\mathbb{P}_t\left(A_{s,\tau(s)} \mid G_{s-1}\right)} - 1\right] \\
=_{(**)}& \sum_{s=1}^{S}\left(\frac{1}{\mathbb{P}_t\left(A_{s,\tau(s)}\right)} - 1\right),
\end{aligned}$$

where (*) uses that $\log(\frac{1}{q}) \leq \frac{1}{q} - 1$ for any $0 < q \leq 1$, and (**) follows by Lemma A.3. So by definition of $R_s^{(t)}$, the above equation implies

$$\mathcal{L}^{(t)} \leq \sum_{s=1}^{S} R_s^{(t)}.$$

□

The following lemma bounds how much a single gradient step can *hurt* the policy. It will be needed to ensure that once the model converges, it will not diverge due to noise arising from the finite batch size.

Lemma A.12. *Under assumptions 4.1, 4.2 and Assumption 5.3, for any $s \in [S]$ and $t \in \mathbb{N} \cup \{0\}$, if $\eta < \frac{1}{4\gamma^2}$ then*

$$\mathbb{P}_{t+1}\left(A_{s,\tau(s)}\right) \geq \mathbb{P}_t\left(A_{s,\tau(s)}\right) - 2\eta\gamma^2.$$

Proof. First, we claim that for any $\mathbf{x}_{s-1}$ and any $v \neq \sigma_{\tau(s)}(\mathbf{x}_{s-1})$,

$$\left|\Delta_{s,v}^{(t)}(\mathbf{x}_{s-1})\right| \leq 2\eta\gamma^2. \tag{12}$$

Indeed, by Lemma A.2, for any token $w \in \mathcal{V}$,

$$\frac{1}{\eta\gamma^2}\Big(f_{\theta^{(t+1)}}(\mathbf{x}_{s-1}) - f_{\theta^{(t)}}(\mathbf{x}_{s-1})\Big)[w] = \sum_{j:\sigma_j(\mathbf{x}_{s-1})=w} Q_{s,j}^{(t)}.$$

By Assumption 4.2, for each fixed $\mathbf{x}_{s-1}$ and token w, the set $\{j : \sigma_j(\mathbf{x}_{s-1}) = w\}$ has size at most 1. Hence

$$\frac{1}{\eta\gamma^2}\Delta_{s,v}^{(t)}(\mathbf{x}_{s-1}) = \sum_{j:\sigma_j(\mathbf{x}_{s-1})=\sigma_{\tau(s)}(\mathbf{x}_{s-1})} Q_{s,j}^{(t)} - \sum_{j:\sigma_j(\mathbf{x}_{s-1})=v} Q_{s,j}^{(t)},$$

which is a difference of at most two $Q_{s,j}^{(t)}$ terms. Since each $Q_{s,j}^{(t)} \in [-1, 1]$, we obtain (12).

Let $\beta_s^{(t)} := \inf_{\mathbf{x}_{s-1}} \inf_{v \neq \sigma_{\tau(s)}(\mathbf{x}_{s-1})} \Delta_{s,v}^{(t)}(\mathbf{x}_{s-1})$. Then, by the choice of η, we have that $\left|\beta_s^{(t)}\right| \leq 1/2$. So the conditions of Lemma A.9 are satisfied, and it follows from the definition of $R_s^{(t)}$ and

Lemma A.9 that

$$\begin{aligned}\mathbb{P}_{t+1}\left(A_{s,\tau(s)}\right)-\mathbb{P}_t\left(A_{s,\tau(s)}\right)=&\frac{1}{R_s^{(t+1)}+1}-\frac{1}{R_s^{(t)}+1}\\ \geq&\frac{1}{\exp(-\beta_s^{(t)})R_s^{(t)}+1}-\frac{1}{R_s^{(t)}+1}\\ \geq_{(*)}&-2\eta\gamma^2,\end{aligned}$$

where (*) is because for any $x>0$ and $z\in\mathbb{R}$, $\left|\frac{1}{1+\exp(-z)x}-\frac{1}{1+x}\right|\leq|z|$. □

A.4 Proof of Theorem 5.4

The proof proceeds by tracking the evolution of the error ratios $R_s^{(t)}=\frac{1}{\mathbb{P}_t(A_{s,\tau(s)})}-1$. We show that whenever $\mathbb{P}_t(A_{s,\tau(s)})$ is not already close to 1, it increases multiplicatively, and once it is sufficiently large, it cannot decrease by more than a negligible amount. We prove something slightly stronger than the result in the main paper. We also prove that the KL divergence (cross entropy loss) between the target function and the learned model, given by $\mathcal{L}(\theta):=\mathbb{E}_{\mathbf{x}_0\sim\mathcal{D}}\left[\log\frac{1}{\mathbb{P}_\theta(f^\star(\mathbf{x}_0)|\mathbf{x}_0)}\right]$ converges to 0.

Theorem A.13. *Under assumptions 4.1, 4.2 and Assumption 5.3, for any $\epsilon>0,\delta\in(0,1)$, let*

$$\tilde{\epsilon}:=\frac{1}{2S}\min\left(\epsilon\ ,\ \frac{1}{2}\ ,\ \min_{s\in[S]}\mathbb{P}_0\left(A_{s,\tau(s)}\right)\right) \tag{13}$$

$$\kappa_0:=\min_{s\in[S]}\mathbb{P}_0(A_{s,\tau(s)})\big(1-\mathbb{P}_0(A_{s,\tau(s)})\big), \tag{14}$$

$$\eta:=\frac{\tilde{\epsilon}}{8\gamma^2}, \tag{15}$$

$$T:=\left\lceil\frac{256\left(1+\frac{1}{\alpha}\right)\log\left(1/\tilde{\epsilon}\right)}{\tilde{\epsilon}^2}\right\rceil \tag{16}$$

$$B:=\left\lceil\frac{128\left(1+\frac{1}{\alpha}\right)^2\log\left(\frac{2JST}{\delta}\right)}{\min(\tilde{\epsilon},\kappa_0)^2}\right\rceil. \tag{17}$$

Then it holds with probability at least $1-\delta$ that at time T,

$$\mathbb{P}_T\left(V=1\right)\geq1-\epsilon\qquad\textit{and}\qquad\mathcal{L}^{(T)}\leq\epsilon.$$

Proof. For any t let $R_s^{(t)}=\frac{1}{\mathbb{P}_t(A_{s,\tau(s)})}-1$. By Lemma A.10 and Lemma A.11 it suffices to prove that for all $s\in[S]$, $R_s^{(T)}\leq2\tilde{\epsilon}$, since then

$$\mathbb{P}_T\left(V=1\right)\geq1-\sum_{s=1}^{S}R_s^{(T)}\geq1-2S\tilde{\epsilon}\geq1-\epsilon\qquad\text{and}\qquad\mathcal{L}^{(T)}\leq\sum_{s=1}^{S}R_s^{(T)}\leq2S\tilde{\epsilon}\leq\epsilon.$$

From Lemmas A.7, A.8 and the union bound, it holds with probability at least $1-\delta$ that for all $s\in[S]$, $t<T$, $\mathbf{x}_{s-1}$ and token $v\neq\sigma_{\tau(s)}(\mathbf{x}_{s-1})$, if $\theta^{(t)}\in\Theta_+$ then

$$\begin{aligned}\frac{1}{\eta\gamma^2}\Delta_{s,v}^{(t)}(\mathbf{x}_{s-1})\geq&\left(\frac{\alpha}{1+\alpha}\right)\left(1-\mathbb{P}_t\left(A_{s,\tau(s)}\right)\right)\mathbb{P}_t\left(A_{s,\tau(s)}\right)-\sqrt{\frac{8\log\left(\frac{2JST}{\delta}\right)}{B}}\\ \geq_{(*)}&\left(\frac{\alpha}{1+\alpha}\right)\left(1-\mathbb{P}_t\left(A_{s,\tau(s)}\right)\right)\mathbb{P}_t\left(A_{s,\tau(s)}\right)-\left(\frac{\alpha}{1+\alpha}\right)\frac{\tilde{\epsilon}}{4}\\ \geq&\left(\frac{\alpha}{1+\alpha}\right)\left(\left(1-\mathbb{P}_t\left(A_{s,\tau(s)}\right)\right)\mathbb{P}_t\left(A_{s,\tau(s)}\right)-\frac{\tilde{\epsilon}}{4}\right)\\ \geq_{(**)}&\left(\frac{\alpha}{1+\alpha}\right)\left(\left(1-\mathbb{P}_t\left(A_{s,\tau(s)}\right)\right)\mathbb{P}_t\left(A_{s,\tau(s)}\right)-\frac{2}{3}\left(1-\frac{\tilde{\epsilon}}{2}\right)\frac{\tilde{\epsilon}}{2}\right),\end{aligned} \tag{18}$$

where in (*) we used the definition of B, and in $(**)$ we used that $\tilde{\epsilon} \leq 1/2$, so $\frac{\tilde{\epsilon}}{4} = \frac{2}{3}\frac{\tilde{\epsilon}}{2} \cdot \frac{3}{4} \leq \frac{2}{3}\frac{\tilde{\epsilon}}{2}\left(1 - \frac{\tilde{\epsilon}}{2}\right)$. The remainder of the proof assumes that this event indeed occurs.

Let $\beta_s^{(t)} := \inf_{\mathbf{x}_{s-1}} \inf_{v \neq \sigma_{\tau(s)}(\mathbf{x}_{s-1})} \Delta_{s,v}^{(t)}(\mathbf{x}_{s-1})$, $\mathcal{I}_1 := [\tilde{\epsilon}/2, 1 - \tilde{\epsilon}/2]$, $\mathcal{I}_2 := [1 - \tilde{\epsilon}, 1]$ and $T_+ := \inf\{t \geq 0 \mid \theta^{(t)} \notin \Theta_+\}$ (where Θ_+ is defined in Assumption 5.3). Note that by definition of Θ_+, $T_+ \geq 1$.

Restricted to the domain $\mathcal{I}_1$, the function $p \mapsto p(1-p)$ obtains its minimum at the border, so that $p(1-p) \geq (1 - \tilde{\epsilon}/2)\tilde{\epsilon}/2$ for any $p \in \mathcal{I}_1$. Thus, the definition of $\mathcal{I}_1$ together with (18) implies that for all $t < \min(T, T_+)$,

$$\textbf{if} \quad \mathbb{P}_t\left(A_{s,\tau(s)}\right) \in \mathcal{I}_1 \quad \textbf{then} \quad \beta_s^{(t)} \geq \frac{\eta\gamma^2\left(\frac{\alpha}{1+\alpha}\right)}{8}\tilde{\epsilon}. \tag{19}$$

So by Lemma A.9, when the conditions of (19) are satisfied, $\mathbb{P}_{t+1}\left(A_{s,\tau(s)}\right) > \mathbb{P}_t\left(A_{s,\tau(s)}\right)$. This implies that for all $t < \min(T, T_+)$,

$$\textbf{if} \quad \mathbb{P}_t\left(A_{s,\tau(s)}\right) \in \mathcal{I}_1 \quad \textbf{then} \quad \mathbb{P}_{t+1}\left(A_{s,\tau(s)}\right) \geq \mathbb{P}_t\left(A_{s,\tau(s)}\right) \text{ and } \mathbb{P}_{t+1}\left(A_{s,\tau(s)}\right) \in \mathcal{I}_1 \cup \mathcal{I}_2. \tag{20}$$

So we have shown that $\mathbb{P}_t\left(A_{s,\tau(s)}\right)$ increases in $\mathcal{I}_1$. We now show that if it reaches $\mathcal{I}_2$, it will stay there until time T. Specifically, by Lemma A.12 and the definition of η,

$$\mathbb{P}_{t+1}\left(A_{s,\tau(s)}\right) \geq \mathbb{P}_t\left(A_{s,\tau(s)}\right) - 2\eta\gamma^2 \geq \mathbb{P}_t\left(A_{s,\tau(s)}\right) - \frac{\tilde{\epsilon}}{4}.$$

We now verify that the iterates remain in Θ_+. Suppose $t < T$ and $\theta^{(t)} \in \Theta_+$, and fix $s \in [S]$, so that $R_s^{(t)} \leq R_s^{(0)}$. On the same concentration event, the choice of B bounds the error in each normalized logit gap by $\frac{\alpha}{1+\alpha}\frac{\kappa_0}{4}$. If $R_s^{(t)} \geq R_s^{(0)}/2$, then

$$\mathbb{P}_t(A_{s,\tau(s)})\big(1 - \mathbb{P}_t(A_{s,\tau(s)})\big) = \frac{R_s^{(t)}}{(1+R_s^{(t)})^2} \geq \frac{R_s^{(0)}}{2(1+R_s^{(0)})^2} \geq \frac{\kappa_0}{2}.$$

Consequently, Lemma A.7 gives $\beta_s^{(t)} \geq \eta\gamma^2\frac{\alpha}{1+\alpha}\frac{\kappa_0}{4} > 0$, and Lemma A.9 implies $R_s^{(t+1)} \leq R_s^{(t)} \leq R_s^{(0)}$. Otherwise, $R_s^{(t)} < R_s^{(0)}/2$, and (12) together with Lemma A.9 gives $R_s^{(t+1)} \leq \exp(2\eta\gamma^2)R_s^{(t)} < R_s^{(0)}$, since the choice of η ensures $\exp(2\eta\gamma^2) < 2$. Thus, induction yields $\theta^{(t)} \in \Theta_+$ for all $t \leq T$, and $T_+ > T$. As such, for all $t \in \{0, \ldots, T-1\}$,

$$\textbf{if} \quad \mathbb{P}_t\left(A_{s,\tau(s)}\right) \in \mathcal{I}_2 \setminus \mathcal{I}_1 \quad \textbf{then} \quad \mathbb{P}_{t+1}\left(A_{s,\tau(s)}\right) \in \mathcal{I}_2. \tag{21}$$

Since $\mathbb{P}_0\left(A_{s,\tau(s)}\right) \in \mathcal{I}_1 \cup \mathcal{I}_2$, (20) and (21) ensure that $\mathbb{P}_t\left(A_{s,\tau(s)}\right) \in \mathcal{I}_1 \cup \mathcal{I}_2$ for all $t \in \{0, \ldots, T\}$. Furthermore by (20) and (21), if there exists a time $\tilde{T} \leq T$ such that $\mathbb{P}_{\tilde{T}}\left(A_{s,\tau(s)}\right) \geq 1 - \tilde{\epsilon}$, this will remain the case for all $t \in \{\tilde{T}, \ldots, T\}$. If $\mathbb{P}_0(A_{s,\tau(s)}) \in \mathcal{I}_2$, the desired bound already follows from invariance of $\mathcal{I}_2$. Otherwise, let

$$\tilde{T} := \inf\{t \in \mathbb{N} \mid \mathbb{P}_t\left(A_{s,\tau(s)}\right) \in \mathcal{I}_2\}.$$

It remains to prove that $\tilde{T} \leq T$.

Suppose for contradiction that $\tilde{T} > T$. Then $\mathbb{P}_t\left(A_{s,\tau(s)}\right) \in \mathcal{I}_1$ for all $t \in \{0, \ldots, T-1\}$, so Lemma A.9 and (19) imply

$$R_s^{(T)} \leq \frac{R_s^{(T-1)}}{1+\beta_s^{(T-1)}} \leq R_s^{(0)} \prod_{j=0}^{T-1} \frac{1}{1+\beta_s^{(j)}} \leq \frac{R_s^{(0)}}{\left(1 + \frac{\eta\gamma^2\left(\frac{\alpha}{1+\alpha}\right)\tilde{\epsilon}}{8}\right)^T}.$$

So if $T \geq \frac{\log\left(R_s^{(0)}/\tilde{\epsilon}\right)}{\log\left(1+\frac{\eta\gamma^2\left(\frac{\alpha}{1+\alpha}\right)\tilde{\epsilon}}{8}\right)}$ and $\tilde{T} > T$, then $R_s^{(T)} \leq \tilde{\epsilon}$ and thus

$$\mathbb{P}_T\left(A_{s,\tau(s)}\right) = \frac{1}{1+R_s^{(T)}} \geq \frac{1}{1+\tilde{\epsilon}} \geq 1 - \tilde{\epsilon},$$

contradicting $\tilde{T} > T$. Hence $\tilde{T} \le T$, and invariance of $\mathcal{I}_2$ gives $R_s^{(T)} \le \tilde{\epsilon}/(1-\tilde{\epsilon}) \le 2\tilde{\epsilon}$. To guarantee the above bound on T, it suffices to take

$$T := \left\lceil \frac{256 \log(1/\tilde{\epsilon})}{\left(\frac{\alpha}{1+\alpha}\right)\tilde{\epsilon}^2} \right\rceil \ge_{(*)} \left\lceil \frac{16 \log\left(R_s^{(0)}/\tilde{\epsilon}\right)}{\eta\gamma^2\left(\frac{\alpha}{1+\alpha}\right)\tilde{\epsilon}} \right\rceil \ge_{(**)} \frac{\log\left(R_s^{(0)}/\tilde{\epsilon}\right)}{\log\left(1 + \frac{\eta\gamma^2 \frac{\alpha}{1+\alpha}\tilde{\epsilon}}{8}\right)},$$

where (*) follows from the definition of η and using that $R_s^{(0)} \le 1/\min_{s\in[S]} \mathbb{P}_0\left(A_{s,\tau(s)}\right) \le 1/\tilde{\epsilon}$, and (**) follows from the fact that $\frac{2}{x} \ge \frac{1}{\log(1+x)}$ for any $0 < x \le 1$. □

B Proof of Proposition 6.1

Let $\mathcal{V} = \{1, 2\}$, and let the prompt $\mathbf{x}_0$ be 1 with probability $1/2$, and 2 with probability $1/2$. For all $\mathbf{x}$ of any length, let $\sigma_1(\mathbf{x}) = 1$ and $\sigma_2(\mathbf{x}) = 2$. Let $\gamma = 1$ so that $g_1(\mathbf{x}) = [1, 0]^\top$ and $g_2(\mathbf{x}) = [0, 1]^\top$.

$$V(\mathbf{x}_S) = \begin{cases} 1 & \mathbf{x}_S \in \{(1,1,1), (2,1,1), (1,2,2)\} \\ 0 & \text{Otherwise.} \end{cases}$$

The target model $f^\star$ is given by applying σ_1 at both steps 1 and 2.

Now for any time t, step s and task σ_j let $z_{s,j}^{(t)} := \langle \theta^{(t)}, \mathbf{h}_{s-1,j} \rangle$. Let $\phi(z) := \frac{1}{1+\exp(-z)}$ and for $s \in \{1, 2\}$, $Z_s^{(t)} := z_{s,1}^{(t)} - z_{s,2}^{(t)}$. Then for an input $\mathbf{x}$ that contains a prompt plus an additional $s-1$ CoT tokens (for $s \in \{1, 2\}$),

$$f_{\theta^{(t)}}(\mathbf{x}_{s-1}) = \sum_{j=1}^{2} g_j(\mathbf{x}_{s-1}) z_{s,j}^{(t)} = \begin{bmatrix} z_{s,1}^{(t)} \\ z_{s,2}^{(t)} \end{bmatrix}. \tag{22}$$

So at any s, the policy is given by $\boldsymbol{\pi}_t\left(\sigma_1(\mathbf{x}_{s-1}) \mid \mathbf{x}_{s-1}\right) = \phi(Z_s^{(t)})$ for any prompt $\mathbf{x}_{s-1}$. As such, by the choice of p_0, we know that $Z_1^{(0)} = Z_2^{(0)} = \phi^{-1}(p_0)$. Now suppose that there is some time t such that $Z_1^{(t)} = Z_2^{(t)} = \phi^{-1}(p_t)$ (this holds for $t = 0$ by assumption). Then, by the problem structure, it is straightforward to compute that for any $s \in \{1, 2\}$,

$$\mathbb{P}_t\left(V = 1 \mid A_{s,1}\right) = p_t \qquad \text{and} \qquad \mathbb{P}_t\left(V = 1 \mid A_{s,2}\right) = \frac{1 - p_t}{2}.$$

As a result,

$$\rho_{s,1}^{(t)} = \frac{2p_t}{1 - p_t} \qquad \text{and} \qquad \rho_{s,2}^{(t)} = \frac{1 - p_t}{2p_t}. \tag{23}$$

Furthermore, letting $D(q) := \frac{q^2}{q^2 + \frac{1}{2}(1-q)^2}$, by summing up the probabilities of paths with $V = 1$, it is also straightforward to compute that for any $s \in \{1, 2\}$,

$$\mathbb{P}_t\left(A_{s,1} \mid V = 1\right) = D(p_t) \qquad \text{and} \qquad \mathbb{P}_t\left(A_{s,2} \mid V = 1\right) = 1 - D(p_t). \tag{24}$$

As such, by Theorem A.6, and by (22) it follows that for any $s, j \in \{1, 2\}$

$$\begin{aligned} \mathbb{E}_{\mathcal{B}_t}[z_{s,j}^{(t+1)} - z_{s,j}^{(t)}] =& \mathbb{E}_{\mathcal{B}_t}\left[\left(f_{\theta^{(t+1)}}(\mathbf{x}_{s-1}) - f_{\theta^{(t)}}(\mathbf{x}_{s-1})\right)\left[\sigma_j(\mathbf{x}_{s-1})\right]\right] \\ =& \eta \mathbb{P}_t\left(A_{s,j} \mid V = 1\right)\left(1 - \mathbb{P}_t\left(A_{s,j}\right)\right)\left(1 - \frac{1}{\rho_{s,j}^{(t)}}\right). \end{aligned}$$

So by (23), (24), for $j = 1$ it holds for any $s \in \{1, 2\}$ that

$$\mathbb{E}_{\mathcal{B}_t}[z_{s,1}^{(t+1)} - z_{s,1}^{(t)}] = \eta D(p_t)\left(1 - p_t\right)\left(\frac{3p_t - 1}{2p_t}\right) = \eta \frac{p_t\left(1 - p_t\right)}{2p_t^2 + (1 - p_t)^2}(3p_t - 1).$$

Likewise, for $j = 2$,

$$\mathbb{E}_{\mathcal{B}_t}[z_{s,2}^{(t+1)} - z_{s,2}^{(t)}] = \eta(1 - D(p_t))p_t\left(\frac{1 - 3p_t}{1 - p_t}\right) = \eta \frac{p_t(1 - p_t)}{2p_t^2 + (1 - p_t)^2}(1 - 3p_t).$$

In particular, these imply that

$$\lim_{B\to\infty}\left(Z_s^{(t+1)}-Z_s^{(t)}\right)=2\eta\frac{p_t\left(1-p_t\right)}{2p_t^2+(1-p_t)^2}(3p_t-1) \tag{25}$$

To ease notation, the remainder of the proof considers the infinite batch size limit. Since $Z_1^{(0)}=Z_2^{(0)}=\phi^{-1}(p_0)$, it follows by induction that for any t, $Z_1^{(t)}=Z_2^{(t)}$. So the conditions for the above derivations hold for any time t. We now split the proof into two cases.

First, if $p_0>1/3$, then (25) implies that $Z_s^{(t+1)}\geq Z_s^{(t)}$ for all t. As such, for any $t, p_t\geq p_0$. So we may lower bound (25) using p_0 as

$$Z_s^{(t+1)}-Z_s^{(t)}\geq 2\eta\frac{p_t\left(1-p_t\right)}{2p_t^2+(1-p_t)^2}(3p_0-1)\geq 2\eta\frac{1-p_t}{2}(3p_0-1).$$

So suppose by contradiction that $p_t\not\to 1$. Then, using that $3p_0-1>0$, there is some constant $c>0$ depending only on p_0 and η such that for all t, $Z_s^{(t+1)}\geq Z_s^{(t)}+c$. But then $Z_s^{(t)}\to\infty$, implying that $p_t=\frac{1}{1+\exp(-Z_s^{(t)})}\to 1$ which is a contradiction. So as $t\to\infty$, $p_t\to 1$, implying that the policy converges to always sampling task σ_1.

Now we prove the case of $p_0<1/3$. Now, (25) implies that $Z_s^{(t+1)}\leq Z_s^{(t)}$ for all t. As such, for any $t, p_t\leq p_0$. So we may upper bound (25) using p_0 as

$$Z_s^{(t+1)}-Z_s^{(t)}\leq 2\eta\frac{p_t\left(1-p_t\right)}{2p_t^2+(1-p_t)^2}(3p_0-1)\leq 2\eta p_t(3p_0-1).$$

So suppose by contradiction that $p_t\not\to 0$. Then, using that $3p_0-1<0$, there is some constant $c>0$ depending only on p_0 and η such that for all t, $Z_s^{(t+1)}\leq Z_s^{(t)}-c$. But then $Z_s^{(t)}\to-\infty$, implying that $p_t=\frac{1}{1+\exp(-Z_s^{(t)})}\to 0$ which is a contradiction. So as $t\to\infty$, $p_t\to 0$ implying that the policy converges to always sampling task σ_2. In particular this implies the proposition.

C Setting and Proof of Proposition 7.1

Setting. We assume that the inputs (prompts) $\mathbf{x}_0\in\mathbb{R}^{d+1}$ contain d bits sampled uniformly from the cube $\{-1,1\}^d$ followed by a last bit which is always 1 (adding this bias term simplifies the setting). The model is given by (1), and the tasks contain two components: $\sigma_1(\mathbf{x})=x_{-d-1}\cdot x_{-1}$ which outputs the product between the last bit in the string and the bit d positions from the end, and $\sigma_0(\mathbf{x})=x_{-1}$ which "does nothing", and just copies the last bit in the string.

In this way, the model essentially goes over the string bit by bit, and decides if each bit either contributes to the parity or is ignored. So, if the desired parity is P, for any $s\in[S]$, letting $\tau(s)=\mathbf{1}_{s\in P}$, the model can always output the right token by composing the tasks $\sigma_{\tau(1)},\ldots,\sigma_{\tau(d)}$. The verifier accepts a solution iff the final bit is correct.

Proof of Proposition 7.1.

Proof. We start by calculating the task-advantage ratios at $\theta^{(0)}$. Let $s\in[d]$ be a step in the CoT. The correct task is $\sigma_{\tau(s)}$ where $\tau(s):=\mathbf{1}_{s\in P}$. As the initial policy is uniform, if the correct task is chosen, then there is a chance of $(1/2)^{d-1}$ that the other correct tasks will be chosen as well. In that case, the verifier will always return 1. It is well known that different parities are uncorrelated over the uniform distribution on the cube, meaning that for any other case, the verifier's correctness rate will be $1/2$. Therefore:

$$\mathbb{P}(V=1|A_{s,\tau(s)})=\frac{1}{2}\cdot\left(1-\frac{1}{2^{d-1}}\right)+1\cdot\frac{1}{2^{d-1}}=\frac{1}{2}\left(1+\frac{1}{2^{d-1}}\right) \tag{26}$$

Now, if the correct task is not chosen at step s, then for any task selection in the other steps the verifier's correctness rate will be $1/2$, thus:

$$\mathbb{P}(V=1|A^c_{s,\tau(s)})=\frac{1}{2} \tag{27}$$

Therefore:

$$\rho_{s,\tau(s)} = 1 + \frac{1}{2^{d-1}} \tag{28}$$

Now, let σ_j be an incorrect task at step s. Then again the verifier's correctness rate is $\mathbb{P}(V(\mathbf{x}) = 1|A_{s,j}) = 1/2$. On the other hand, if it is not chosen, then the correct task is chosen, for which we have seen $\mathbb{P}(V(\mathbf{x}) = 1|A^c_{s,j}) > 1/2$. This means $\rho_{s,j} < 1$.

This proves that at initialization, the condition of 5.3 holds. Now, for any $\theta \in \Theta_+$, consider that the likelihood of selecting the correct task at each step increases. So $\mathbb{P}(V(\mathbf{x}) = 1|A_{s,\tau(s)})$ only increases (since there is a higher chance that all the correct tasks will be selected), while $\mathbb{P}(V(\mathbf{x}) = 1|A^c_{s,\tau(s)})$ remains bounded by $1/2$ (as only the fully correct CoT achieves a greater correctness rate).

Thus:

$$\alpha = \frac{1}{2^{d-1}} \tag{29}$$

□

D Proof of proposition 7.2

To formalize this setting, fix some $\lambda_s \in (0, 1)$, and let $\mathcal{S}_{\text{bad}}(\mathbf{x}_S)$ be the set of incorrect tasks chosen by the model. Then the probability the verifier accepts is given by $\prod_{s\in\mathcal{S}_{\text{bad}}} \lambda_s$ (where the randomness is over prompts). We also assume that the policy at each step does not depend on the prefix, so that each choice of tasks doesn't influence the probability of choosing a correct task at a future step. Intuitively, λ_s should be thought of as the chance of recovering from an incorrect CoT step.

Proof. Recall that the probability of the verifier accepting is $\prod_{s\in\mathcal{S}_{\text{bad}}} \lambda_s$, and that the policy does not depend on the prefix. As such, the ratio between the probability of $V = 1$ given $A_{s,\tau(s)}$ (meaning the correct task is chosen at step s) and the probability of $V = 1$ given $A^c_{s,\tau(s)}$ is $\frac{1}{\lambda_s}$. Therefore $\rho_{s,\tau(s)} = \frac{1}{\lambda_s}$.

Now consider an incorrect task σ_j. Since the steps are independent, the contribution from steps $s' \neq s$ to successful verification is equal in the numerator and denominator of the task advantage ratio. Thus, the task-advantage ratio simplifies to the ratio between successfully solving step s of the problem with/without task σ_j. For the incorrect choice σ_j, it is λ_s, and conditioned on not choosing σ_j, it is a convex combination of λ_s (arising from incorrect tasks) and 1 (arising from the correct task). Formally:

$$\rho_{s,j} = \frac{\lambda_s}{\left(1 - \frac{\mathbb{P}_t(A_{s,\tau(s)})}{1-\mathbb{P}_t(A_{s,j})}\right)\lambda_s + \frac{\mathbb{P}_t(A_{s,\tau(s)})}{1-\mathbb{P}_t(A_{s,j})}} = \frac{1}{1 + \frac{\mathbb{P}_t(A_{s,\tau(s)})}{1-\mathbb{P}_t(A_{s,j})}\left(\frac{1}{\lambda_s} - 1\right)} < 1. \tag{30}$$

□

E Relation to the NTK regime

Here, we briefly remark on how the model defined in (1) can be alternatively motivated via the NTK regime. Many works have shown that a large pretrained model $\tilde{f}_\theta$ when fine-tuned with a small learning rate satisfies

$$\tilde{f}_\theta(\mathbf{x}) \approx \tilde{f}_{\theta^{(0)}}(\mathbf{x}) + \nabla_{\theta^{(0)}} \tilde{f}_{\theta^{(0)}}(\mathbf{x}) \cdot (\theta - \theta^{(0)}),$$

so that the features $\nabla_{\theta^{(0)}} \tilde{f}_{\theta^{(0)}}(\mathbf{x})$ remain fixed during fine-tuning [Jacot et al., 2018, Lee et al., 2019, Allen-Zhu et al., 2019, Arora et al., 2019, Jang et al., 2024].

From this perspective, fine-tuning corresponds to learning a linear combination of fixed pretrained features. One may further decompose these features (e.g., via a singular value decomposition) $\nabla_{\theta^{(0)}} \tilde{f}_{\theta^{(0)}}(\mathbf{x}) = \sum_j \tilde{g}_j(\mathbf{x})\tilde{h}_j(\mathbf{x})^T$, yielding a decomposition of the same form as (1).

F Extension to State-Dependent Decision Processes

In the main text, we consider a simplified setting in which task selection depends only on the autoregressive step index. We now describe a more general formulation in which task selection depends on a state derived from the prefix.

State-dependent model. Let $\mathcal{S}$ be a finite state space, and let $s : \mathcal{V}^* \to \mathcal{S}$ be a mapping from prefixes to states. We generalize the positional embeddings by associating each task σ_j and state $s \in \mathcal{S}$ with an embedding vector $\mathbf{h}_{s,j} \in \mathbb{R}^p$.

As in the main text, we assume that the collection $\{\mathbf{h}_{s,j}\}_{s\in\mathcal{S},j\in[J]}$ is orthonormal, so that updates corresponding to different state–task pairs do not interfere.

The model is defined as

$$f_\theta(\mathbf{x}) = \sum_{j=1}^{J} g_j(\mathbf{x}) \, \langle \mathbf{h}_{s(\mathbf{x}),j}, \theta \rangle, \tag{31}$$

where $s(\mathbf{x})$ denotes the state associated with prefix $\mathbf{x}$.

Decision process formulation. In this setting, a compositional problem can be equivalently viewed as a decision process over states. For each state $s \in \mathcal{S}$, let $\tau(s) \in [J]$ denote the ground-truth task to be selected at that state.

This generalizes the step-based formulation in the main text, where the state corresponds to the step index and $\tau(s)$ specifies the correct task at each step of the composition.

Task-advantage ratio. The task-advantage ratio is defined analogously. For any state $s \in \mathcal{S}$ and task $j \in [J]$, let $A_{s,j}$ denote the event that task σ_j is selected when the model is in state s. Then,

$$\rho^\theta_{s,j} = \frac{\mathbb{P}_\theta(V = 1 \mid A_{s,j})}{\mathbb{P}_\theta\big(V = 1 \mid A^c_{s,j}\big)}. \tag{32}$$

For this definition, sample a rollout under the current policy and a uniformly chosen generation step, and condition on the state at that step being s. The complement $A^c_{s,j}$ then means choosing an alternative task at that visit.

Extension of results. We believe that our results in the main text have state-dependent counterparts. In particular:

- The expected-update characterization in Theorem 5.2 has a state-dependent analogue, with the update weighted by the expected number of visits to the state among verified rollouts.
- The uniform task-advantage condition (Assumption 5.3) can be stated over states, requiring that $\rho^\theta_{s,\tau(s)} \geq 1 + \alpha$ for a uniform margin $\alpha > 0$ and $\rho^\theta_{s,j} < 1$ for $j \neq \tau(s)$.
- Corresponding convergence guarantees additionally require quantitative control of state visitation frequencies.

G Experimental Details for GRPO Experiment

To study the predictions of our theoretical framework on real models in a controlled compositional setting, we consider the task of summing three 10-digit integers, $(a + b) + c$. The task consists of two stages: at $s = 1$, the model computes the intermediate sum $d = a + b$, and at $s = 2$, it computes the final sum $d + c$.

We evaluate Qwen2.5-1.5B-Instruct on this task. At each stage, the model can perform the addition using one of two algorithms: *long addition*, which explicitly carries out the arithmetic procedure, or *direct addition*, which outputs the sum without intermediate computation. We identify the algorithm selected at each stage using regular expressions applied to the model's generated outputs.

We first estimate the task advantage ratios under the initial policy, before GRPO training, using 500 randomly generated problems. For each stage $s \in \{1, 2\}$, the empirical task advantage ratio of long addition is computed as

$$\widehat{\rho}_{s,\text{long}} = \frac{\widehat{\Pr}(\text{correct} \mid \text{long at } s)}{\widehat{\Pr}(\text{correct} \mid \text{direct at } s)},$$

where correctness refers to the final answer. This estimates the task advantage ratio defined in Definition 5.1.

The resulting empirical task advantage ratios for long addition are $\widehat{\rho}_{1,\text{long}} = 0.3$ and $\widehat{\rho}_{2,\text{long}} = 0.19$, both strictly below 1. Since there are only two available algorithms, the corresponding ratios for direct addition are their reciprocals and therefore exceed 1 at both stages.

By Theorem 5.2, RLVR updates are consequently predicted to favor direct addition over long addition at both stages. Figure 2 illustrates this behavior during GRPO training: although the initial policy favors long addition, its selection probability rapidly decreases at both stages, while the mean verification reward increases.

Training details. We train Qwen2.5-1.5B-Instruct using GRPO with the following configuration:

- Learning rate: 10^{-4}.
- Group size: 8.
- Problems per training step: 16.
- LoRA rank: 64 (4.57% of model parameters).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Everything is formally defined and stated in the main paper, and proven in the appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Throughout the paper, simplifying assumptions are clearly stated. Moreover, there is a Discussion and Limitations section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Definitions and assumptions appear in the main paper, and formal proofs are provided in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The necessary information appears in the figure captions.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: The experiments in this paper serve as visual depictions rather than standalone results.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The necessary information appears in the figure captions.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The experiments in this paper serve as visual depictions rather than standalone results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The experiments in this paper serve as visual depictions rather than standalone results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The paper fully conforms to the Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The work is theoretical in nature, and we do not foresee direct societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The work is theoretical in nature, and we do not foresee direct societal impact.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: [N/A]

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.