FROM TARGET SHIFT TO COVARIATE SHIFT: HOW RL POST-TRAINING ENABLES TRANSFER TO NEW TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) post-training has substantially improved the reasoning capabilities of autoregressive language models, but whether it can elicit solutions to tasks not explicitly targeted during pre-training remains unclear. We study this question through the canonical problem of learning parities over uniform input bits. Pre-training data consist primarily of short answers to a parity task, together with rare chain-of-thought (CoT) demonstrations that include intermediate computations. During RL post-training, the reward instead targets a *distinct* parity function that is uncorrelated with the pre-training target. Experimentally, we observe that RL post-training can achieve nontrivial accuracy on the new task, escaping apparent computational barriers, while supervised fine-tuning under the same information is ineffective. During the course of RL training, networks undergo sudden phase changes in both accuracy and generation length, and can converge to suboptimal strategies. By developing an effective theory of the RL dynamics, we characterize some of these suboptimal strategies, and link them to the formation of low-dimensional features that can be learned efficiently, yet are predictive of the new target over only some parts of the input space. We explain how the RL loss encourages the discovery of these features by turning the target shift into a *covariate* shift, allowing the model to repurpose pretrained CoTs on inputs where the two targets coincide.

1 INTRODUCTION

A central question in current language model research is whether reinforcement learning can unlock capabilities that are not acquired through next-token prediction training alone. The practice of training broadly capable "pre-trained" autoregressive models with next-token prediction (Radford et al., 2019) has recently been followed by huge computational efforts to further fine-tune these base models with reinforcement learning (RL) using verifiable rewards (Jaech et al., 2024; DeepSeek-AI et al., 2025). This process has resulted in AI systems with highly developed reasoning skills that succeed in challenging intellectual tasks, such as solving abstract mathematical problems. Such notable reasoning competence does not seem to emerge from next-token prediction on internet data alone, and it is indeed a challenge to understand by what mechanism this occurs. In particular, two separate hypotheses have been suggested:

1. Most of the useful signal is already present in the pre-training phase, yet it is statistically dormant, and RL post-training helps elicit it, or
2. The model can explore and develop genuinely new behaviors during RL post-training, departing from the set of strategies present in the pre-training phase.

Recent works provide some evidence in favor of the first view and have developed some understanding of the mechanisms involved (Zhao et al., 2025; Yue et al., 2025; Chen et al., 2026; Tsilivis et al., 2026), see the related work session for further discussion.

Departing from these prior efforts, in this work we seek to understand:

Can RL post-training teach a base model a task that was not explicitly targeted during pre-training, and through what mechanisms?

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

To address this question, we study a controlled setting of transfer learning. Motivated by the scarcity of "rich", chain-of-thought (CoT) like sequences in internet-scale data, we consider a difficult task presented through an unbalanced mixture of contexts: most are short and provide low-quality information, while a small fraction are longer and more informative, making the underlying task easier to learn. We model this setting using parity functions, a canonical benchmark known to pose computational barriers when learned from short answers alone (Shalev-Shwartz et al., 2017; Abbe and Sandon, 2023), but learnable through next-token prediction with suitable intermediate supervision (Malach, 2023; Joshi et al., 2025; Kim and Suzuki, 2025). The pre-training mixture encodes one parity function, while post-training targets a *distinct* parity function, with limited supervision available for either supervised or RL-based fine-tuning.

An important feature of our setting is that the two functions are *uncorrelated* at the population level, while being structurally related. In particular, the rule that distinguishes them can be much simpler than either function itself. This creates an opportunity for transfer: the model may learn the new task by exploiting its relation to the pre-training task, rather than learning it from scratch.

We provide a theoretical and empirical account of the mechanisms involved as transformers adapt to the new target over the course of RL post-training. In particular, we find that networks can make nontrivial progress on the new task, by learning to generate progressively longer responses, while supervised fine-tuning under the same setting fails. We find that sometimes the network can adapt to the new task by identifying the regions of the input space where the pre- and post-training targets agree, rather than attempting to learn the new target over the whole space. This often leads to the presence of plateaus at suboptimal strategies. Nevertheless, such input identification translates the learning of an a priori difficult target into the learning of the "difference" between the two targets, which can be much simpler for targets with enough structural relation. This permits progress with respect to the new task, even for tasks that appear to raise computational barriers to learning. We develop an effective theory approach in characterizing some of the suboptimal strategies and how RL leads the network to them. Finally, our theoretical analysis reveals that the RL loss can encourage this process by turning the target shift into a *covariate* shift, thereby forcing the model to fit sequences that are part of neither the pre-training nor the post-training distribution.

Our contributions In this work, we pre-train an autoregressive transformer by next-token prediction on a mixture of short input–answer demonstrations and rare, longer sequences that include intermediate CoT computations. Both types of sequence target the same pre-training parity function. For input bits encoded as $x \in \{\pm 1\}^d$, a *parity function* multiplies a selected subset of the bits: $\chi_S(x) := \prod_{i \in S} x_i$. The selected subset $S \subseteq \{1, \ldots, d\}$ is called its *support*. Thus, the output is $+1$ when an even number of selected bits are negative, and -1 otherwise. After pre-training, we change the target and reward a distinct parity obtained by removing k coordinates from the support of the pre-training parity. Thus the new task is to compute the parity on a (fixed) subset of the support of the pre-trained task. We assume access only to input–labels from this new parity, with no CoT demonstrations (see Section 3). Note that the two targets are uncorrelated under a uniform input distribution, so (a) the accuracy of the base model with respect to the new target is on par with random chance, and (b) it is unclear whether prior exposure to one target can help with learning the other.

Our contributions are the following:

- **Failure of supervised fine-tuning.** In our experiments, supervised fine-tuning fails to learn the new target, and starting from the pre-trained model offers no clear advantage over training from scratch (Figure 2). This is consistent with the known difficulty of learning parities with gradient-based methods (See Appendix A for details).
- **Success of RL post-training.** RL post-training can make progress on the new target and achieve nontrivial accuracy efficiently. In particular, we observe that:
  - i) The typical response length increases during post-training;
  - ii) Sample efficiency depends on the support mismatch k between the two targets, rather than only on the complexity of the new target;
  - iii) Many runs plateau at a suboptimal solution with near 0.75 accuracy, while some reach perfect accuracy under the new target. Sudden phase changes, akin to those observed during RL fine-tuning of large scale language models (Sun et al., 2025; Huang et al., 2026), occur in the accuracy and the generation length of the networks.

- **An effective theory: From target shift to covariate shift.** We develop an effective theory that characterizes successive RL updates in an idealized autoregressive model. Conditioning on successful responses converts the target shift into a *covariate shift*, enabling the model to identify inputs where the two targets agree and selectively generate its pre-trained CoT. The analysis explains the increase in response length, anticipates the formation of low-dimensional features in the post-trained model, and predicts a 0.75 accuracy limit in the case the initial policy exactly mirrors the pre-training distribution. The aforementioned low-dimensional features consist of a *degree-k agreement gate* and are the signal that enables transfer. Finally, an extension with last-token noise shows how atypical but correct responses can enable convergence to perfect accuracy.

Overall, our findings support the claim that RL with verified rewards can induce generalization beyond the tasks seen during pre-training and shed light on mechanisms that can enable this generalization.

2 Related Work

Reinforcement learning in language models. A growing body of work investigates how RL post-training shapes the reasoning capabilities of language models. On the empirical side, Zhao et al. (2025) study language models trained from scratch in controlled settings and show that RL post-training amplifies behaviors acquired during pre-training. Complementing this perspective, Yue et al. (2025) find that, in the settings they investigate, RL improves performance at small sampling budgets, while base models achieve higher pass@k at sufficiently large k. These findings suggest that improvements can arise from increasing the probability of existing successful behaviors. Other studies provide evidence for compositional generalization during RL post-training. Yuan et al. (2026) and Abdulsalam et al. (2026) show that models can learn previously unseen compositions of acquired atomic skills, with the resulting compositional ability also transferring across tasks. Park et al. (2025) find that RL enables generalization to larger problems and expression structures held out from RL training, with learnability depending strongly on the structure of the composition. Sun et al. (2025); Huang et al. (2026) find that language models can undergo sudden phase transitions during RL fine-tuning, with abrupt changes in their generalization profile. On the theoretical side, Chen et al. (2026) argue that a notion of coverage characterizes how well test-time strategies such as Best-of-N sampling can perform, hypothesizing that the same principle can capture when RL after next-token prediction succeeds. Tsilivis et al. (2026) develop an end-to-end quantitative understanding of how RL after next-token prediction can rapidly improve performance as long as pre-training already places small but nonnegligible probability mass on long, chain-of-thought reasoning traces.

Learning parities with neural networks. Parity learning has been widely used as a controlled benchmark for studying the capabilities and limitations of neural-network learning (see for example (Wang et al., 2025; Panigrahi et al., 2025; Daniely and Malach, 2020; Barak et al., 2022; Malach et al., 2021; Collins et al., 2024). Although noiseless parities are efficiently learnable by Gaussian elimination over the two-element field, learning them from input–answer pairs can pose computational barriers to gradient-based methods (Shalev-Shwartz et al., 2017; Abbe and Sandon, 2023). We focus on *dense* parities over uniform inputs, defined here by support size q satisfying $\min\{q, d-q\} = \omega_d(1)$, where d is the input dimension. Under suitable assumptions on gradient precision and batch size, such parities cannot be learned in polynomial time by gradient-based procedures that are invariant to input permutations, including in their initialization distributions (Abbe and Sandon, 2023); see Section A for details. With suitable intermediate supervision, however, even dense parities can be learned efficiently through next-token prediction (Malach, 2023; Joshi et al., 2025; Kim and Suzuki, 2025). We refer to Appendix A for further details on the hardness of parity learning.

3 Formal Setting

We now introduce our setting. Pre-training data consist of short answers and longer chain-of-thought sequences encoding a parity function, while post-training targets a separate parity. In particular, we consider the pre-training problem of learning a *parity over half of the input bits*, also known as *dense* parity. The choice of dense parities is motivated by their known hardness for gradient-based learning

(see Section A for details). Let d be even and set $m = d/2$. The input space is $\mathcal{X} = \{\pm 1\}^d$, with $x \sim \mathcal{D}_x := \mathrm{Rad}(\frac{1}{2})^{\otimes d}$ denoting the uniform distribution over the hypercube, and the output space is $\mathcal{Y} = \{\pm 1, \texttt{<EOS>}\}^*$. For $j \le m$, we denote by $\chi_j(x) := \prod_{i=1}^{j} x_i$ the parity of the first j input bits.

Pre-training. The pre-training target f_{pre} is the dense parity χ_m, with $m = d/2$. Each input $x \sim \mathcal{D}_x$ is paired either with a short answer or, with probability p_{cot}, with a long chain-of-thought demonstration:

$$y = \begin{cases} \big(x_1, x_1x_2, \ldots, \chi_m(x), \texttt{<EOS>}\big), & \text{with prob. } p_{\mathrm{cot}}, \\ \big(\chi_m(x), \texttt{<EOS>}\big), & \text{with prob. } 1 - p_{\mathrm{cot}}. \end{cases}$$

The long demonstration includes as a chain-of-thought all the partial parities for $j \le m-1$. We focus on the case where p_{cot} is "small", i.e., the long sequences appear rarely in the pre-training mixture. We denote the joint distribution over $\mathcal{X} \times \mathcal{Y}$ by $\mathcal{D}_0(p_{\mathrm{cot}})$, and pre-train the model on samples from this distribution using a next-token prediction loss.

Remark 1. Our pre-training setup follows (Tsilivis et al., 2026), except that the target is the dense parity of the first $d/2$ input bits, rather than the *full* parity over all d bits. We make this choice since dense parities provide a more robust hard benchmark for gradient-based learning from input–answer pairs, while learnability of the full parity is more sensitive to the choice of initialization and activation function. See Section A for a detailed discussion.

Post-training. RL post-training aims at extracting information from the pre-trained model by rewarding outputs that align with a desired behavior defined by a *reward function*. Our goal in this work is to study a case in which the post-training task is different from the pre-training. We model this by considering a reward targeting a different parity, obtained by removing k coordinates from the support of the pre-training parity. The input distribution remains $\mathcal{D}_x$, but the target $f_{\mathrm{post},k}$ is shifted from χ_m to χ_{m-k}, for $1 \le k \ll d$. Thus, the reward is

$$r_{\mathrm{e2e},k}(x, y) = \mathbb{1}\{\mathrm{ans}(y) = \chi_{m-k}(x)\}\,, \tag{1}$$

where $\mathrm{ans}(y)$ denotes the last non-<EOS> token of y.

In our experiments in Section 4, we consider RL post-training with the GRPO (Group Relative Policy Optimization) algorithm, while our theoretical analysis in Section 5 considers RL post-training with respect to a rejection fine-tuning / STaR objective (Zelikman et al., 2022).

Remark 2. The parameter k controls the difficulty of the task. When k is "small", i.e., $k = O_d(1)$, the post-training target χ_{m-k} is a high-degree, dense, parity and it is therefore difficult to learn from input-answer pairs only. However, the pre- and post-training supports differ only on the k removed variables, and the residual is therefore of low-complexity.

4 Empirical findings

The main object of study in this work is the empirical observation that transformers can efficiently adapt to unseen parities through RL-based post-training, leveraging pre-training information from structurally related, yet uncorrelated, targets. Experimental details are provided in Appendix C.

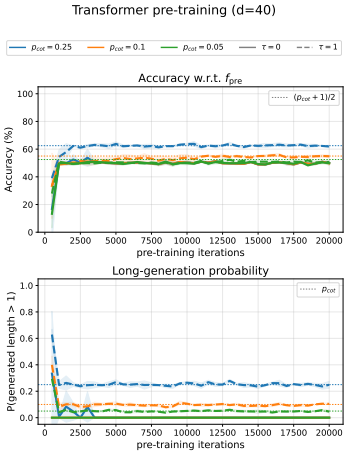

Figure 1: **Pre-training a transformer on a mixture of short/long sequences encoding the dense parity f_{pre}.** Figure shows autoregressive accuracy and probability of generating a long response under both greedy decoding (τ=0) and under sampling with temperature 1.0 (τ=1.0) for various values of mixture parameter p_{cot}.

Pre-training. We first pre-train a transformer on data from a short/long mixture encoding the $m = d/2$ parity function f_{pre}. Figure 1 presents results for input dimension $d = 40$ and for various values of p_{cot}. Under sampling with temperature 1.0, the network learns to generate a long response with probability p_{cot} and its accuracy equals $(1 + p_{\mathrm{cot}})/2$. For small values of p_{cot}, the network generates short, "random guesses" under greedy decoding. The dense parity f_{pre} is a difficult target to learn through short sequences, and it is only the long branch of the distribution that provides any signal towards learning the target, in line with findings in prior work (Tsilivis et al., 2026).

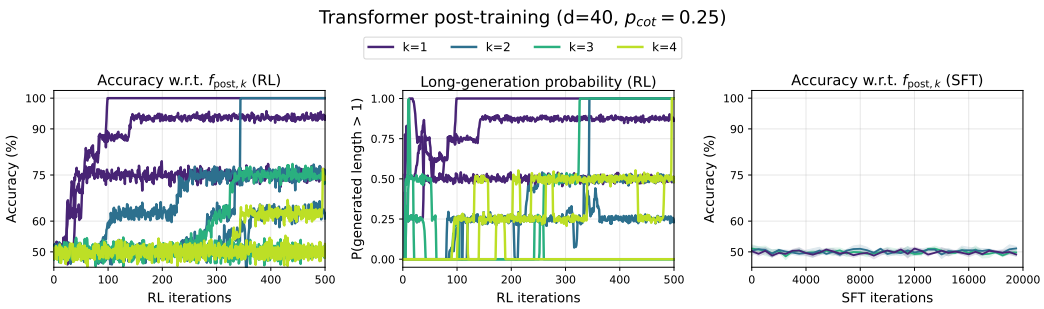

Figure 2: **Adapting to unseen dense parities efficiently with RL:** Transformer post-training with respect to target $f_{\text{post},k}$ for various values of k and for various random seeds. We compare Reinforcement-Learning (RL) with Supervised Fine-Tuning (SFT). *Left*: RL-based post-training improves test accuracy with more iterations. *Center*: The probability of generating a long response increases during RL fine-tuning. *Right*: SFT fails to raise performance above random guessing, even when run for more iterations than the RL algorithm (same batch size used in both algorithms).

4.1 RL Post-training efficiently adapts to unseen parities

We provide empirical evidence that RL post-training with respect to the $m - k$ parity function $f_{\text{post},k}$ can succeed in obtaining non-trivial accuracy for small values of k. Note that, right after pre-training, the accuracy of the model with respect to the new task is on par with random chance. Figure 2 shows multiple runs of RL and SFT post-training of the $(d = 40, p_{\text{cot}} = 0.25)$ pre-trained models.

Learning through length increase. We observe that, for all $k \in \{1, 2, 3, 4\}$, there exist RL runs that obtain non-trivial accuracy with respect to the new target $f_{\text{post},k}$, with some of them generalizing perfectly (Figure 2, left). The number of RL iterations needed to obtain non-trivial accuracy increases with the size of the shift k. Simultaneously, we observe that, for the majority of the runs, the probability of generating a long response increases from its initial value (Figure 2, center). Note that the RL algorithm here can make progress on a target that is considered to be computationally hard to learn from scratch under the same supervision signal.

Supervised fine-tuning fails. An alternative to RL post-training is to fine-tune the pre-trained model on supervised sequences of the form $(x, f_{\text{post},k}(x), \texttt{<EOS>})$. However, this strategy is not effective for teaching the new target to the transformer for any $k \in \{1, 2, 3, 4\}$, and accuracy remains at chance levels, even when supervised training continues for much longer than RL (Figure 2, right).

Phase transitions and plateaus. Progress in accuracy can be sudden and performance can plateau during training. In particular, we observe several runs to plateau accuracy at roughly 0.75. Similar phenomena of plateaus followed by abrupt changes have been observed before in supervised learning of *sparse* parities in feed-forward networks (Barak et al., 2022) and in RL-based fine-tuning of language models in more complicated domains (Sun et al., 2025).

4.2 Learning the new target by learning low-dimensional features

How does RL escape the computational barriers of learning dense parities from solely input-output supervision? Why does the response length increase? Does RL leverage the CoT data from the pre-trained target f_{pre} to teach the network the new target $f_{\text{post},k}$? And what are the suboptimal solutions the networks stagnate at? In this subsection, we provide some empirical answers to the previous questions. We focus our analysis on the cases of $k = 1, 2$.

Band of possible solutions. As already evidenced in Figure 2, the networks can undergo several phase changes during the course of RL training. In our experiments, we find that the majority of the RL runs concentrate in a band of possible (sub)optimal solutions with distinct accuracy levels (Figure 3, left). Among a pool of 60 different random seeds that control the pre-trained model and data sampling during RL, the majority of the runs achieve either 1.0 or 0.75 accuracy.

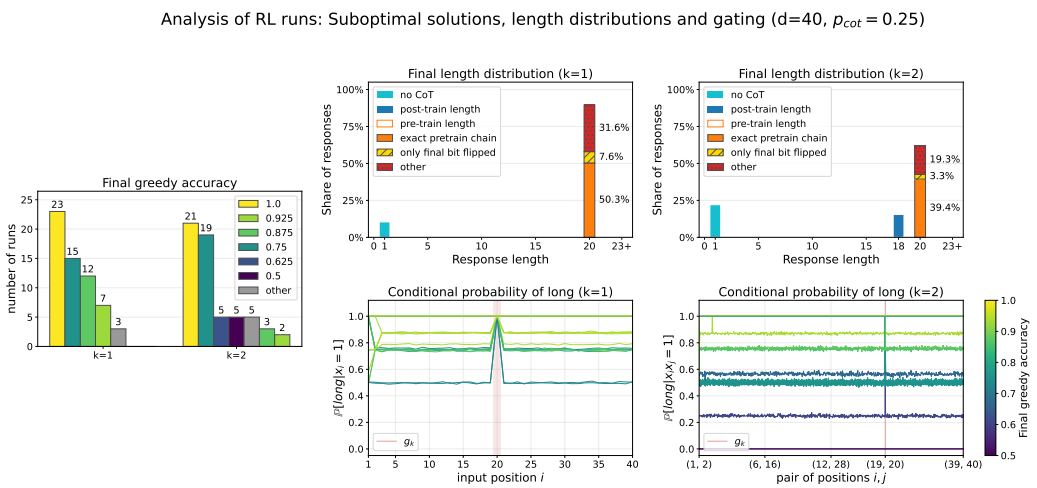

Figure 3: **Low dimensional features control long generations.** *Left*: We show the distribution of the final accuracy of RL runs (60 seeds). Each bar shows the count of runs that land within 0.025 of the given final accuracy; most reach either 1.0 or 0.75 accuracy. *Top row*: Distribution of response lengths (under greedy decoding) at the end of RL training, averaged over all seeds. Length 1 corresponds to a no CoT response, length $d/2 - k$ corresponds to lengths equal to the support of the post-training parity, while length $d/2$ corresponds to lengths equal to the CoTs present in the pre-training mixture (further divided by whether they agree or not with the pre-training CoT over all tokens). The majority of generations are of the pre-training type. *Bottom row*: The RL runs that land near the suboptimal-0.75 solution generate long responses whenever a low dimensional gate, $g_k(x) = \prod_{j=d/2-k+1}^{d/2}$ is equal to 1. That is, a k-dimensional feature decides whether a generation will be of length $d/2$ or shorter. Each line corresponds to a different run, colored by its final accuracy.

Reusing the pretrained computations. This non-trivial performance is achieved by the network learning to generate long responses. Recall that, right after pre-training, the models always generate a short response under greedy decoding (Figure 1). In contrast, at the end of RL post-training, the networks have learned to generate responses that are, for the most part, of length equal to that of the CoTs in the pre-training mixture. Among them, most in fact match exactly the pre-trained CoTs that correspond to the initial target f_{pre}. That is, the networks learn to emit the pre-trained computations despite their corresponding to a different target. See Figure 3, top row, for the distribution of lengths at the end of training, aggregated across all random seeds. Curiously, and perhaps surprisingly, we also observe a share of responses that are exactly of length $m - k$ (for $k = 2$), despite the lack of such sequences in the pre-training mixture. Note that these responses are the only ones that can be characterized as "faithful" to the computation of the post-training target $f_{\text{post},k}$, since they only contain the previous parities of degree up to $m - k - 1$, together with the target.

Gating long generations on low dimensional features. As we saw, the networks can learn to generate the pre-trained CoTs that correspond to the initial target f_{pre}. For the pre-trained CoTs to achieve above-chance accuracy on the new task, the network must generate them preferentially on inputs where the two targets agree. Indeed, under uniform inputs, $\mathbb{P}_x[f_{\text{pre}}(x) = f_{\text{post},k}(x)] = \mathbb{P}_x\left[\prod_{j=m-k+1}^{m} x_j = +1\right] = \frac{1}{2}$, thus, generating the unchanged pre-trained CoT on every input would achieve only chance accuracy under the new target. Nevertheless, the two targets do agree whenever the k-degree *gate* $g_k(x) := \prod_{j=m-k+1}^{m} x_j$ equals $+1$. This leads to the following hypothesis: the networks learn to generate the long responses corresponding to f_{pre} only in the regions of the input space where $g_k(x) = +1$, i.e., where the two targets coincide. Figure 3 (bottom row) directly tests and provides evidence in support of this hypothesis. For each random seed, we plot the fraction of long[1] responses conditional on $x_i = 1$, for each $i \in [d]$, when $k = 1$, and on $x_i x_j = 1$, for each $1 \le i < j \le d$, when $k = 2$. We observe that most runs plateauing near 0.75 accuracy generate

[1] Of length m.

long responses exactly on inputs where the agreement gate $g_k(x) = \begin{cases} x_m, & k = 1, \\ x_{m-1}x_m, & k = 2. \end{cases}$ equals $+1$. That is, the RL algorithm translates the problem of learning the hard, dense parity target $f_{\text{post},k}$ into that of learning a low-dimensional sparse parity g_k. Prior work has shown that such targets can indeed be learned efficiently by neural networks (Barak et al., 2022), and this explains how RL can make progress in learning $f_{\text{post},k}$. Note, however, that relying only on such a gate can only help with learning on half of the input space, which is why these runs plateau at 0.75 accuracy. Next, we develop a theoretical model that explains how RL post-training incites this gating mechanism. We provide further evidence for the previous claims (see, in particular, Figure 5, right), as well experiments with other post-training targets and algorithms in Appendix B.

5 Theoretical analysis

In this section, we provide a theoretical account of how RL post-training enables transfer between dense parity targets. We analyze post-training with respect to an idealized population RL objective and track how it evolves across rounds, given the evolution of the policy. At every RL round, we assume that the policy is induced by optimal solutions within a prescribed feature class, thereby abstracting away complex optimization dynamics. Our analysis demonstrates how the RL objective shapes the supervision received by the model, allowing it to elicit and reuse computations acquired from rare CoTs during pre-training, and how this can sometimes lead to a suboptimal overall strategy.

5.1 Gating and the 3/4 suboptimal solution

Here, we show how RL post-training can induce the gating mechanism that controls lengthy generations by turning the target shift into a covariate shift and how this leads to a suboptimal solution.

Pre-trained policy. Recall, the input space $\mathcal{X} = \{\pm 1\}^d$, the output space $\mathcal{Y} = \{\pm 1, \texttt{<EOS>}\}^\star$ and the uniform input distribution $\mathcal{D}_x = \text{Rad}(1/2)^{\otimes d}$. Let $\pi_{\theta^{(0)}} : \mathcal{X} \to \Delta(\mathcal{Y})$ denote the pre-trained policy and $m = d/2$. Following the empirical results on pre-training (Figure 1), we assume that the initial, pre-trained policy generates the correct long demonstration with probability $p := p_{\text{cot}} \in (0, 1)$, while its short answers remain approximately uniform. Specifically, for every $x \in \mathcal{X}$ and $a \in \{\pm 1\}$,

$$\begin{aligned} \pi_{\theta^{(0)}}((a, \texttt{<EOS>}) \mid x) &= \frac{1-p}{2}, \\ \pi_{\theta^{(0)}}((x_1, x_1x_2, \ldots, \chi_m(x), \texttt{<EOS>}) \mid x) &= p. \end{aligned} \tag{2}$$

RL objective. We consider RL post-training with a reward that encodes the shifted target χ_{m-k}, for $1 \le k \ll d$. For the RL algorithm, we study a population STaR (Zelikman et al., 2022) / rejection fine-tuning loss, where at each round, we sample inputs from $\mathcal{D}_x$, generate responses from the current policy, and optimize the log loss on those receiving reward 1. The first RL round is therefore

$$\theta^{(1)} \in \arg\min_\theta \underbrace{\mathbb{E}_{\substack{x\sim\mathcal{D}_x \\ y\sim\pi_{\theta^{(0)}}(\cdot|x)}} [-\log \pi_\theta(y \mid x) \,|\, r_{\text{e2e},k}(x, y) = 1]}_{L^{(1)}(\theta)}, \tag{3}$$

where $r_{\text{e2e},k}(x, y) = \mathbb{1}\{\text{ans}(y) = \chi_{m-k}(x)\}$ is the reward function and $\pi_{\theta^{(0)}}(.|x)$ is the policy obtained after pre-training. Subsequent RL rounds are defined inductively by the same loss with respect to the policy of the previous round. Thus, every RL round amounts to learning from a new distribution of model-generated sequences, obtained by retaining only those whose final answer matches the shifted target.

From target shift to covariate shift. As observed in Section 4, the usefulness of the pretrained CoT for the new task is determined by the *agreement gate*

$$g_k(x) := \chi_m(x)\chi_{m-k}(x) = \prod_{i=m-k+1}^{m} x_i. \tag{4}$$

Recall that, under the initial policy, short answers succeed with probability $1/2$ on every input, while long responses succeed exactly when $g_k(x) = 1$. Since this agreement gate is balanced under uniform

inputs, both branches succeed on half of their samples, preserving their respective mixture weights $1-p$ and p after conditioning on success (reward 1). In turn, successful short responses retain uniform inputs, whereas successful CoTs are restricted to inputs where the targets agree. By the preceding arguments and an application of the law of total expectation, the first-round loss from (3) can be rewritten as a next-token prediction loss

$$L^{(1)}(\theta) = (1-p)\,\mathbb{E}_{X\sim\mathcal{D}_x}\big[\mathcal{L}_{\text{short}}(X;\theta)\big] + p\,\mathbb{E}_{X\sim\mathcal{D}_x(\cdot|g_k=1)}\big[\mathcal{L}_{\text{long}}(X;\theta)\big], \tag{5}$$

where $\mathcal{L}_{\text{short}}$ and $\mathcal{L}_{\text{long}}$ denote the next-token losses on the correct short answer and the pre-trained CoT, respectively. Crucially, the resulting input marginal is

$$\mathcal{D}'_x(x) = \mathcal{D}_x(x)\big(1+pg_k(x)\big).$$

Thus, although inputs are sampled uniformly, the RL loss favors inputs for which the pre-trained computation solves the new task. In other words, the target shift from χ_m to χ_{m-k} induces a *covariate shift* from $\mathcal{D}_x$ to $\mathcal{D}'_x$, exposing a signal governed by the degree-k agreement gate g_k. We next examine how restricted autoregressive policies that are solutions to eq. (3) and to subsequent RL rounds use this signal to selectively generate the pre-trained CoT.

Autoregressive policy – Logistic coarsenings. We consider a sequence of linear predictors, one per token position, with access to non-linear features, which together induce an autoregressive policy. Linear time-dependent autoregressive models (i.e., without parameter sharing across positions, as in standard autoregressive transformers) have been considered in prior work to elucidate the power of autoregressive learning (Malach, 2023; Tsilivis et al., 2026).

The shifted training distribution of eq. (5) determines the exact next-token conditional probabilities, but a model may be unable to represent all their input dependencies. In practice, which dependencies are accessible may depend on the architecture and training resources (e.g., the number of RL iterations, the batch size and the learning rate). We capture this limitation by considering *coarsenings* of the next-token conditional distributions induced by feature maps. In particular, we assume that, at each token generation step, the corresponding linear predictor has access to a prescribed feature map ϕ, restricting predictions to $h_w(x) := \sigma(\langle w, \phi(x)\rangle)$, where $\sigma(u) := (1+e^{-u})^{-1}$. For a binary prediction problem with labels $Y \in \{\pm 1\}$, the *logistic coarsening* is the best approximation to $\mathbb{P}(Y=+1 \mid X=x)$ within this family, measured by the population logistic loss

$$\mathbb{E}_{(X,Y)\sim\mathcal{D}}\left[\log\left(1+e^{-Y\langle w,\phi(X)\rangle}\right)\right].$$

Here, X includes the input and any preceding tokens available to the predictor. We refer to Definition 1 in Appendix for a formal definition of coarsenings.

Main result. We characterize the properties of a policy returned by an RL algorithm that operates as above for n rounds. In particular, our main result compares the final policies under two coarsenings that differ in their ability to represent the agreement gate.

- *Rich coarsening.* We consider a feature map containing all monomials of degrees 1 through $k+1$ in the input bits:

  $$\phi_k(x) := \big(\chi_S(x)\big)_{S\subseteq[d],\,1\le|S|\le k+1}.$$

  These features allow the model to represent the agreement gate and use it to guide generation.
- *Insufficient coarsening.* We consider linear features:

  $$\phi_l(x) := (x_1,\ldots,x_d).$$

  For $k \ge 2$, these features cannot represent the agreement gate.

Our main result reads as follows.

Proposition 1 (Informal)**.** *Consider population RL with either the rich or the insufficient coarsening defined above. Starting from the initial policy in (2), with $p \in (0,1)$, let π_n denote the resulting policy after n rounds. Assume $k \ge 1$ and $\max\{k,2\}+1 < m-k$. On uniform inputs, its sampled accuracy $\mathrm{Acc}(\pi_n)$ and long-response probability $\mathrm{Long}(\pi_n)$ satisfy:*

*i) Under **rich coarsening**:*

$$\mathrm{Acc}(\pi_n) \longrightarrow \frac{3}{4}, \qquad \mathrm{Long}(\pi_n) \longrightarrow \frac{1}{2} \qquad \textit{as } n \to \infty.$$

*ii) Under **insufficient coarsening**, if $k \geq 2$:*

$$\mathrm{Acc}(\pi_n) = \frac{1}{2}, \qquad \mathrm{Long}(\pi_n) = p, \qquad n \geq 1.$$

Part i) of the result predicts convergence to 0.75 accuracy, with long responses being generated on precisely half of the inputs where the two targets agree. Note that this matches exactly the mechanism identified in the experiments for the 0.75 runs.

Part ii), in turn, provides an example where features unable to capture the agreement gate leave accuracy at chance. Note that the rich coarsening uses $\Theta(d^{k+1})$ features at the first position for fixed k. Learning these dependencies may therefore require more training as k grows. This is consistent with our experiments, although our population analysis assumes the features are available and does not characterize the time needed to learn them. We refer to Appendices D,E,F,G for the formal statement (Corollary 1 and Proposition 5) and proof.

5.2 Perfect generalization via rewarding atypical sequences

Last-token noise. While many runs in our experiments plateau at suboptimal accuracy, some seeds achieve perfect accuracy. We hypothesize that errors in the initial policy can help explain this behavior by assigning positive probability to successful long responses even on inputs where the two targets disagree. We study this mechanism through a tractable model that assumes noise only at the last answer token of the pre-trained policy. That is, conditional on generating a long response, the initial policy follows the pre-trained CoT but independently flips its final answer with probability $\varepsilon \in (0, 1)$. The initial long and short-response probabilities remain unchanged. While in practice we do not expect to be confined to the final token, this idealization allows a clean analysis of how RL can amplify atypical responses that are correct for the new target. The analysis assumes the feature map that induces the rich coarsenings.

Proposition 2 (Informal)**.** *Consider population RL with rich coarsening under the last-token noise assumptions above. Let π_n denote the resulting policy after n rounds. On uniform inputs, its sampled accuracy* $\mathrm{Acc}(\pi_n)$ *and long-response probability* $\mathrm{Long}(\pi_n)$ *satisfy*

$$\mathrm{Acc}(\pi_n) \longrightarrow 1, \qquad \mathrm{Long}(\pi_n) \longrightarrow 1 \qquad \textit{as } n \to \infty.$$

We refer to Appendix H for the formal statement (Proposition 6) and proof.

6 Limitations & Discussion

Several limitations exist in our setting and its study. In typical reasoning problems in LLMs, inputs can have more than one correct answer, in contrast to the post-training reward we consider. Furthermore, our theoretical analysis studies an idealized RL algorithm instead of an on-policy algorithm such as GRPO, considers linear autoregressive policies instead of transformers, and does not perform an optimization analysis of how the model finds the optimal policy between RL rounds. Future work could address these limitations in order to better understand the sample/time complexity of these RL algorithms for adapting to new targets.

In the canonical setting of hard-to-learn target functions we studied, we found RL post-training to be able to efficiently transfer pre-trained knowledge from one target to another. While this ability is remarkable and shows that post-training can teach new skills to a model, the method by which this is achieved can be problematic, since it can often depend on identifying input regions where pre-trained knowledge can be reapplied, rather than truly learning the new target. Indeed, we saw how this can lead to suboptimal strategies. More importantly, this tendency also implies that the post-trained CoTs can turn out to be "unfaithful" to the computation involved, posing an important obstacle to trustworthy machine reasoning.

AI USE STATEMENT

In this work, we used generative AI tools for helping running experiments, polishing the theoretical proofs and claims, finding references reported in the related work section, and polishing the writing. AI further helped us developing the RL recursion for the last-token noise model in Section H. We have not used generative AI tools for developing our theoretical model, generating critical proof ideas and hypothesis, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools for helping creating plots, brainstorming, and improving readability. We have reviewed all AI-assisted work. The LLM-generated code was verified and tested for correctness by one author. The polished proofs were checked by two authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All our theoretical assumptions and claims are fully written. Our code is attached as supplementary material.

REFERENCES

Emmanuel Abbe and Colin Sandon. Polynomial-time universality and limitations of deep learning. *Communications on Pure and Applied Mathematics*, 76(11):3493–3549, 2023.

Emmanuel Abbe, Elisabetta Cornacchia, Jan Hązła, and Donald Kougang Yombi. Learning high-degree parities: The crucial role of the initialization. In *International Conference on Learning Representations*, volume 2025, pages 19448–19494, 2025.

Azwar Abdulsalam, Nishil Patel, and Andrew Saxe. Rl post-training builds compositional reasoning strategies, 2026. URL https://arxiv.org/abs/2607.07646.

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 *Symposium on Theory of Computing (STOC)*, pages 253–262, 1994.

Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003.

Fan Chen, Audrey Huang, Noah Golowich, Sadhika Malladi, Adam Block, Jordan T. Ash, Akshay Krishnamurthy, and Dylan J Foster. The coverage principle: How pre-training enables post-training. In *The Fourteenth International Conference on Learning Representations*, 2026.

Liam Collins, Hamed Hassani, Mahdi Soltanolkotabi, Aryan Mokhtari, and Sanjay Shakkottai. Provable multi-task representation learning by two-layer ReLU neural networks. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 9292–9345. PMLR, 21–27 Jul 2024.

Amit Daniely and Eran Malach. Learning parities with neural networks. *Advances in Neural Information Processing Systems*, 33:20356–20365, 2020.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang

Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, and S. S. Li. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/ARXIV.2501.12948. URL https://doi.org/10.48550/arXiv.2501.12948.

Benjamin L Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Pareto frontiers in neural feature learning: Data, compute, width, and luck. *arXiv preprint arXiv:2309.03800*, 2023.

Margalit Glasgow. Sgd finds then tunes features in two-layer neural networks with near-optimal sample complexity: A case study in the xor problem. *arXiv preprint arXiv:2309.15111*, 2023.

Yu Huang, Zixin Wen, Yuejie Chi, Yuting Wei, Aarti Singh, Yingbin Liang, and Yuxin Chen. On the emergence of implicit curriculum in RLVR learning dynamics. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=KxYCE98u1d.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, and Ilge Akkaya. Openai o1 system card. *CoRR*, abs/2412.16720, 2024. doi: 10.48550/ARXIV.2412.16720.

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.

Juno Kim and Taiji Suzuki. Transformers provably solve parity efficiently with chain of thought. In *International Conference on Learning Representations*, volume 2025, pages 44642–44668, 2025.

Yiwen Kou, Zixiang Chen, Quanquan Gu, and Sham M Kakade. Matching the statistical query lower bound for k-sparse parity problems with stochastic gradient descent. *arXiv preprint arXiv:2404.12376*, 2024.

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

Eran Malach, Pritish Kamath, Emmanuel Abbe, and Nathan Srebro. Quantifying the benefit of using differentiable learning over tangent kernels. In *International Conference on Machine Learning*, pages 7379–7389. PMLR, 2021.

Abhishek Panigrahi, Bingbin Liu, Sadhika Malladi, Andrej Risteski, and Surbhi Goel. Progressive distillation induces an implicit curriculum. In *International Conference on Learning Representations*, volume 2025, pages 23694–23742, 2025.

Simon Park, Simran Kaur, and Sanjeev Arora. How does rl post-training induce skill composition? a case study on countdown, 2025. URL https://arxiv.org/abs/2512.01775.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI*, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Shai Shalev-Shwartz, Ohad Shamir, and Shaked Shammah. Failures of gradient-based deep learning. In *International Conference on Machine Learning*, pages 3067–3075. PMLR, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024.

Yiyou Sun, Yuhan Cao, Pohao Huang, Haoyue Bai, Hannaneh Hajishirzi, Nouha Dziri, and Dawn Song. Rl grokking recipe: How does rl unlock and transfer new algorithms in llms?, 2025. URL https://arxiv.org/abs/2509.21016.

Taiji Suzuki, Denny Wu, Kazusato Oko, and Atsushi Nitanda. Feature learning via mean-field langevin dynamics: classifying sparse parities and beyond. *Advances in Neural Information Processing Systems*, 36, 2024.

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

Binxu Wang, Emma Lucia Byrnes Finn, and Bingbin Liu. When rule learning breaks: Diffusion fails to learn parity of many bits. In *NeurIPS 2025 Workshop on Structured Probabilistic Inference {\&} Generative Modeling*, 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. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *CoRR*, abs/2504.13837, 2025. doi: 10.48550/ARXIV.2504.13837.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

Rosie Zhao, Alexandru Meterez, Sham M. Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. Echo chamber: RL post-training amplifies behaviors learned in pretraining. In *Second Conference on Language Modeling*, 2025.

A HARDNESS OF DENSE PARITIES

Parity functions over uniform input bits provide a controlled setting for studying both hardness of learning and the potential for transfer. For a support $S \subseteq [d]$, write $\chi_S(x) := \prod_{i \in S} x_i, X \sim \mathrm{Unif}(\{\pm 1\}^d)$. Parities have long served as central examples in computational learning theory (Blum et al., 1994; 2003). While noiseless parities are efficiently learnable from input–answer pairs by Gaussian elimination over the two-element field, their relevance here comes from the computational barriers they pose to particular learning procedures, including gradient-based methods under suitable assumptions (Shalev-Shwartz et al., 2017; Abbe and Sandon, 2023).

A central source of difficulty is the orthogonality of distinct parities under the uniform input distribution:

$$\mathbb{E}_X[\chi_S(X)\chi_{S'}(X)] = \mathbf{1}\{S = S'\}.$$

For methods that access the data through approximate population averages, this orthogonality limits the information available for distinguishing an unknown target from a large collection of candidate parities. The class of degree-q parities contains $\binom{d}{q}$ functions. For *dense* parities, i.e. whenever $\min\{q, d-q\} = \omega(1)$, this number is super-polynomial. Under the gradient-precision, batch-size, and training assumptions of the corresponding lower bounds, learning a randomly chosen dense parity requires super-polynomial computational resources (Shalev-Shwartz et al., 2017; Abbe and Sandon, 2023).

For constant-degree, or *sparse*, parities ($q = O(1)$), positive results establish learning guarantees for various architectures, initializations, and training dynamics (Barak et al., 2022; Glasgow, 2023; Edelman et al., 2023; Suzuki et al., 2024). In particular, Kou et al. (2024) show that sign SGD on a suitable two-layer network learns degree-q parities with $\widetilde{O}(d^{q-1})$ samples for fixed $q \geq 2$.

On the other hand, for $q = d - O(1)$, including the *full* parity on all bits, learnability is subtle, and might depend on the choice of initialization (Abbe et al., 2025), or on the activation. As an example of this, in Figure 4 we show experiments where learnability of the full parity by a 1-layer Transformer is strongly affected by the choice of activation (between ReLU and GeLU), while the dense parity on half of the bits is consistently hard.

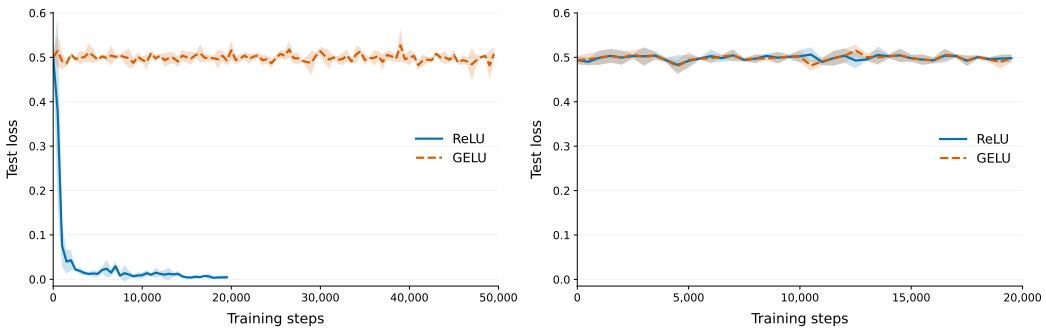

Figure 4: Learning the full parity on all bits (left) and the dense parity on half of the bits (right) on 20-bit inputs, with a 1-layer Transformer with two attention heads and either ReLU or GELU activation. ReLU learns full parity while GELU remains near chance; both remain near chance on half-parity.

B ADDITIONAL EXPERIMENTS

In this section, we present additional experiments as well accompanying plots for the simulations described in the main text.

Estimate of empirical sample complexity. In our experiments in the main body, we focus our analysis on input dimension $d = 40$. We also performed simulations of pre-training followed by post-training for other values of d. Based on these simulations, we estimate the number of GRPO iterations needed for the post-trained model to obtain an accuracy above 0.7 with respect to the new

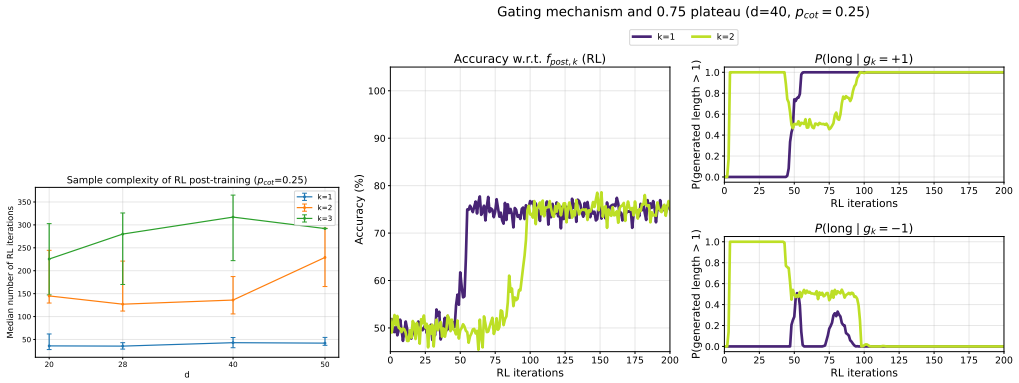

Figure 5: *Left*: An estimate of the amount of RL iterations needed until the network makes nontrivial progress towards the post-training target. Figure shows median number of iterations, together with 25 and 75 percentiles across 30 random seeds per d, k. *Right*: Two indicative RL runs of suboptimal solutions, together with the gating mechanism that explains the plateau.

target. The results are shown in Figure 5 (left). Despite the significant error bars, the scaling appears to be polynomial rather than exponential in d for all the values of k we tested.

Case studies on individual RL runs. Here, we study some indicative $k = 1, 2$ runs that demonstrate the gating mechanism we identified in the main text. In particular, in Figure 5 (right), we show that as autoregressive accuracy (under greedy decoding) goes converges to 0.75, the probability of a long generation conditional on the agreement gate $g_k(x) = +1$ goes to 1, while the probability of a long generation conditional on $g_k(x) = -1$ goes to 0. These probabilities are estimated as the ratio of generations with $g_k(x) = \pm 1$ with length equal to $d/2$ divided by all generations with $g_k(x) = \pm 1$. In particular, we see that the agreement gate g_k controls whether a response will be long or short.

Length distribution per strategy In Figure 6, we examine the chain-length distributions from Figure 3 in greater detail. The top row shows the distributions at the end of training, averaged across all seeds (as in Figure 3). The middle and bottom rows show the corresponding distributions restricted to runs with final accuracies of 0.75 and 1.0, respectively. This breakdown enables a more detailed comparison of chain lengths across outcomes; see the figure caption for further discussion.

Additional p_{cot} values. In Figure 7, we show results for $d = 40$ and $p_{\text{cot}} = 0.1$. In particular, in Figure 7 (left), we show the distribution of accuracies at the end of RL training for $k = 1, 2$. We observe results qualitatively similar to those in the $p_{\text{cot}} = 0.25$ case presented in the main text. In Figure 7 (right), we plot the evolution of accuracy for all the RL runs ($d = 40, p_{\text{cot}} = 0.10, k = 2$), grouped by pre-training seed. We observe that most of the variability comes from the pre-training seed rather than the RL sampling seed.

STaR experiments. To bridge the gap between our simulations and the RL objective we consider in theory, we also conduct experiments using that RL fine-tuning loss. Recall that the STaR objective (Zelikman et al., 2022) performs rejection fine-tuning based on the post-training reward. In Figure 8, we plot the accuracy and probability of a long response, again contrasted with the SFT results. We observe results qualitatively similar to those of GRPO, with the presence of suboptimal solutions as well as runs that succeed in generalizing perfectly.

New post-training target – Larger support. One potential limitation of our setting is the fact the the post-training target is included in an early position in the CoT of the pre-trained sequences. While this feature does not seem to affect our conclusions regarding the suboptimal solutions and the gating mechanism, we did observe some runs exploiting to obtain perfect generalization.

Here, we present experiments with a different post-training target. In particular, we consider $f'_{\text{post},k}(x) = \chi_{m+k}(x)$. That is, the support of the post-training target has k *more* indices. Figure 9 displays the accuracy and probability of long generations for 3 random seeds per $k \in [4]$, as well

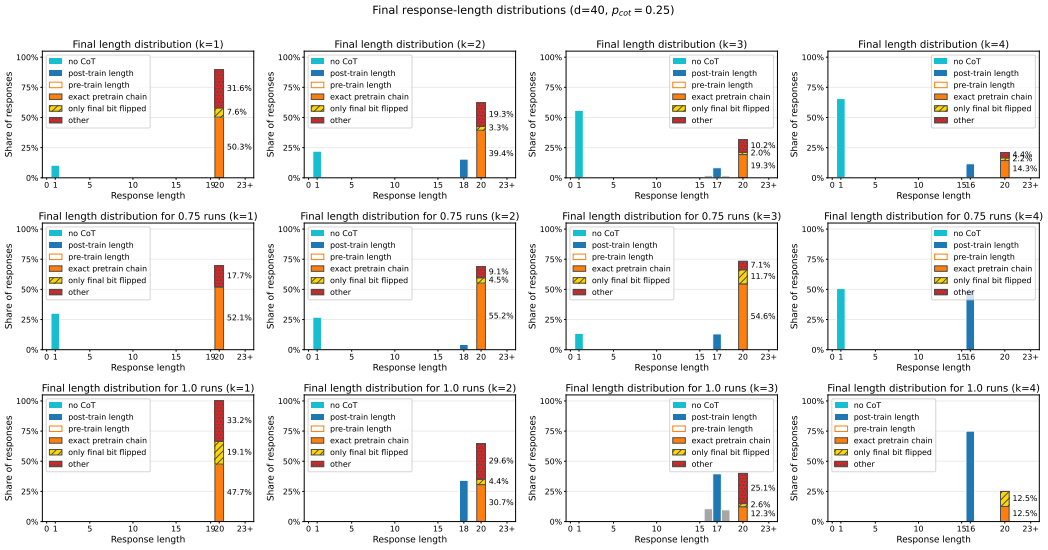

Figure 6: Chain-length distributions grouped by final accuracy. Runs with a final accuracy of 0.75 reproduce the pre-trained CoT exactly through the gating mechanism (see $k = 1$). As k increases, runs with shorter chains more frequently achieve perfect generalization.

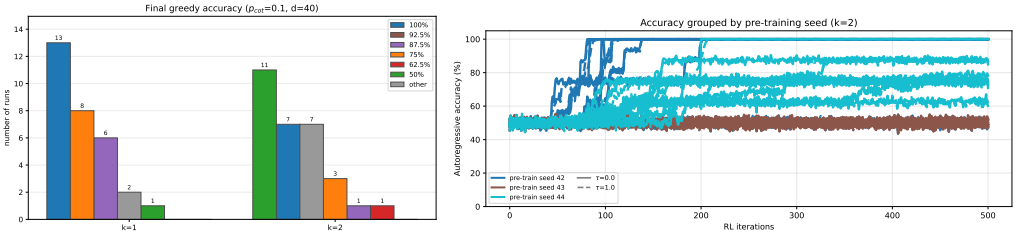

Figure 7: RL post-training aggregated statistics for p_{cot}=0.1 and $d = 40$. *Left*: Distribution of final accuracies. *Right*: RL runs grouped by pre-training seed.

as the distribution of final accuracies over 30 random seeds for $k = 1, 2$. We observe the presence of suboptimal strategies here, too, including the well-understood 0.75 case, as well as existence of perfectly generalizing runs, albeit fewer ones.

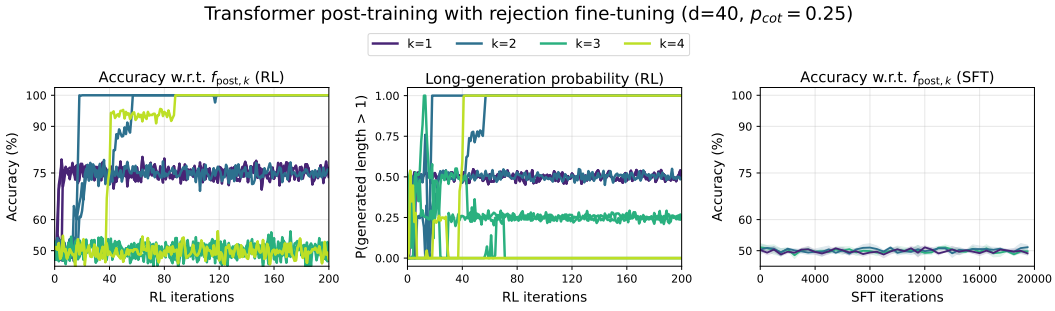

Figure 8: RL rejection fine-tuning for $d = 40$ and $p_{\text{cot}} = 0.25$, contrasted with the SFT results. 3 random seeds per k. Each STaR iteration filters 20,000 samples.

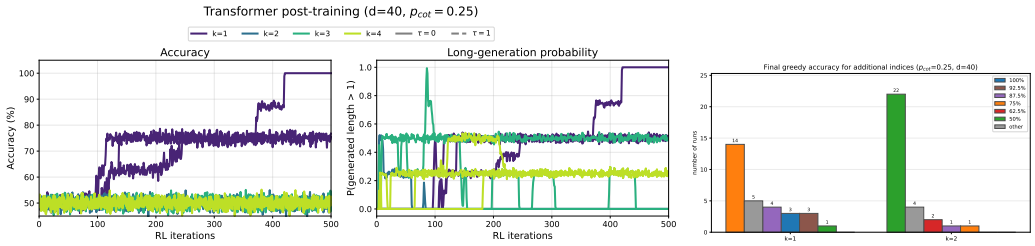

Figure 9: Modifed setting, where the post-training target has k more indices in the support, i.e., $f'_{\text{post},k}(x) = \chi_{m+k}(x)$. *Left*: Accuracy and probability of long generation. *Right*: Distribution of final accuracies.

C Experimental details

Pre-training We initialize a standard, decoder-only transformer with absolute positional embeddings, 4 layers, an embedding dimension of 128, MLP width of 512, GeLU activation function and 32 attention heads, using the GPT-2 architecture. Training runs for 20,000 iterations with a batch size of 256 and a learning rate of 10^{-3}. We use the Adam optimizer.

Post-training We post-train the GPT-2 models using the GRPO algorithm (Shao et al., 2024). Post-training runs for 500 iterations using AdamW with a learning rate of 8×10^{-4}, a batch size of 512, and gradient clipping set to 1.0. We use a generation temperature of 1.0 and no KL penalty is enforced. The group size is 4. In the experiments that use STaR in the Appendix, we use run for 200 iterations of 20,000 samples each, using the Adam optimizer with gradient clipping set to 0.0.

Supervised fine-tuning We fine-tune the GPT-2 models using supervised fine-tuning (SFT). Fine-tuning runs for 20,000 iterations with a learning rate of 10^{-4} and a batch size of 512. We use the Adam optimizer.

D An effective theory

Recall that $m = d/2$, that $\chi_j(x) := \prod_{i=1}^{j} x_i$, and that the pre-training and post-training targets are χ_m and χ_{m-k}, respectively, with $1 \le k < m$. Throughout, $\mathcal{D}_x$ denotes the uniform distribution on $\{\pm 1\}^d$.

Initial policy. Write $p := p_{\text{cot}}$, with $0 < p < 1$. We assume that the initial poly π_0 satisfies

$$\begin{aligned} \pi_0\big((t, \texttt{<EOS>}) \mid x\big) &= \frac{1-p}{2}, \qquad t \in \{\pm 1\}, \\ \pi_0\big((x_1, x_1x_2, \ldots, \chi_m(x), \texttt{<EOS>}) \mid x\big) &= p. \end{aligned} \tag{6}$$

All other responses have probability zero. Thus, the policy generates the correct pre-training trace with probability p, and otherwise gives a uniformly random short answer.

RL objective. As mentioned in the main body, we consider RL post-training with a reward that encodes the shifted target χ_{m-k}, for $1 \le k \ll d$. We study a population STaR (Zelikman et al., 2022) / rejection fine-tuning loss, where at each round, we sample inputs from $\mathcal{D}_x$, generate responses from the current policy, and optimize the log loss on those receiving reward 1. The first RL round is therefore

$$\theta^{(1)} \in \arg\min_{\theta} \underbrace{\mathbb{E}_{\substack{x\sim\mathcal{D}_x \\ y\sim\pi_{\theta^{(0)}}(\cdot|x)}} [-\log \pi_\theta(y \mid x) \,|\, r_{\text{e2e},k}(x,y) = 1]}_{L^{(1)}(\theta)}, \tag{7}$$

where $r_{\text{e2e},k}(x,y) = \mathbb{1}\{\text{ans}(y) = \chi_{m-k}(x)\}$ is the reward function and $\pi_{\theta^{(0)}}(.|x)$ is the policy obtained after pre-training. Subsequent RL rounds are defined inductively by the same loss with respect to the policy of the previous round.

The first accepted distribution. All probabilities in this paragraph are taken under

$$X \sim \mathcal{D}_x, \qquad Y \sim \pi_0(\cdot \mid X).$$

Let $\mathrm{ans}(Y)$ denote the last non-<EOS> token, and define the success event

$$\mathcal{E} := \{\mathrm{ans}(Y) = \chi_{m-k}(X)\}.$$

Define the agreement gate

$$g_k(x) := \chi_m(x)\chi_{m-k}(x) = \prod_{i=m-k+1}^{m} x_i. \tag{8}$$

so that the pre-train and post-train targets agree exactly when $g_k(x) = 1$.

Under the initial policy, event $\mathcal{E}$ is the disjoint union of

$$A_1 := \big\{Y = (\chi_{m-k}(X), \texttt{<EOS>})\big\},$$
$$B_1 := \big\{Y = (X_1, X_1X_2, \ldots, \chi_m(X), \texttt{<EOS>}),\ g_k(X) = 1\big\}.$$

These are the events of generating a successful short response and a successful long response, respectively.

For each input x, their joint masses are

$$\mathbb{P}(X = x, A_1) = \mathcal{D}_x(x)\,\mathbb{P}(A_1 \mid X = x) = \mathcal{D}_x(x)\,(1-p)\,\frac{1}{2} = \mathcal{D}_x(x)\frac{1-p}{2},$$
$$\mathbb{P}(X = x, B_1) = \mathcal{D}_x(x)\,\mathbb{P}(B_1 \mid X = x) = \mathcal{D}_x(x)\,p\,\mathbf{1}\{g_k(x) = 1\} = \mathcal{D}_x(x)\frac{p(1+g_k(x))}{2},$$

Averaging over x, and using the balance of g_k, gives

$$\mathbb{P}(A_1) = \frac{1-p}{2}, \qquad \mathbb{P}(B_1) = \frac{p}{2}, \qquad \mathbb{P}(\mathcal{E}) = \frac{1}{2}.$$

Consequently, the two components have conditional weights

$$\mathbb{P}(A_1 \mid \mathcal{E}) = 1 - p, \qquad \mathbb{P}(B_1 \mid \mathcal{E}) = p.$$

Their conditional input distributions are

$$\begin{aligned}\mathbb{P}(X = x \mid A_1) &= \mathcal{D}_x(x),\\ \mathbb{P}(X = x \mid B_1) &= \mathcal{D}_x(x)(1 + g_k(x))\\ &= \mathcal{D}_x(x \mid g_k(X) = 1).\end{aligned}$$

Thus, conditioning on success preserves uniform inputs within the short component and restricts the long component to inputs on which the two targets agree.

Population refitting loss. Let

$$\mathcal{L}(x, y; \theta) := -\sum_{\ell=1}^{|y|} \log \pi_\theta(y_\ell \mid x, y_{<\ell})$$

denote the summed next-token negative log-likelihood, including the terminal <EOS> token.

The first population refitting objective is

$$L^{(1)}(\theta) := \mathbb{E}\big[\mathcal{L}(X, Y; \theta) \mid \mathcal{E}\big].$$

Using the preceding decomposition, we obtain

$$\begin{aligned}L^{(1)}(\theta) = (1-p)\,\mathbb{E}_{X\sim\mathcal{D}_x}\left[\mathcal{L}\big(X, (\chi_{m-k}(X), \texttt{<EOS>}); \theta\big)\right]\\ + p\,\mathbb{E}_{X\sim\mathcal{D}_x(\cdot\mid g_k=1)}\left[\mathcal{L}\big(X, (X_1, X_1X_2, \ldots, \chi_m(X), \texttt{<EOS>}); \theta\big)\right].\end{aligned} \tag{9}$$

Therefore, even though fresh inputs are sampled uniformly before filtering, the target shift induces a *covariate shift* in the input distribution used for refitting, since we retain only input–response pairs with a correct final answer.

In the next section, we characterize next-token prediction under this shifted distribution and its coarsenings under restricted feature classes.

E NEXT-TOKEN PREDICTION ON THE (FIRST) SHIFTED DISTRIBUTION

Conditioning the first RL round on positive reward induces the following distribution over successful trajectories:

$$(x, y) \sim \begin{cases} \big(x, \big(\chi_{m-k}(x), \texttt{<EOS>}\big)\big), & x \sim \mathcal{D}_x, \quad \text{with probability } 1 - p_{\mathrm{cot}}, \\ \big(x, \big(x_1, x_1x_2, \ldots, \chi_m(x), \texttt{<EOS>}\big)\big), & x \sim \mathcal{D}_x(\,\cdot \mid g_k(x) = 1), \quad \text{with probability } p_{\mathrm{cot}}. \end{cases} \tag{10}$$

We denote by $\mathcal{D}'_x$ the resulting marginal distribution over inputs. Since $\Pr_{x\sim\mathcal{D}_x}(g_k(x) = 1) = 1/2$, we have

$$\begin{aligned} \mathcal{D}'_x(x) &= (1 - p_{\mathrm{cot}})\, \mathcal{D}_x(x) + p_{\mathrm{cot}}\, \mathcal{D}_x(x \mid g_k(X) = 1) \\ &= (1 - p_{\mathrm{cot}})\, \mathcal{D}_x(x) + p_{\mathrm{cot}}\, \frac{\mathcal{D}_x(x)\, \mathbf{1}\{g_k(x) = 1\}}{\Pr_{X\sim\mathcal{D}_x}(g_k(X) = 1)} \\ &= \mathcal{D}_x(x)\, [1 - p_{\mathrm{cot}} + 2p_{\mathrm{cot}}\, \mathbf{1}\{g_k(x) = 1\}] \\ &= \mathcal{D}_x(x)\, [1 - p_{\mathrm{cot}} + p_{\mathrm{cot}}(1 + g_k(x))] \\ &= \mathcal{D}_x(x)\big(1 + p_{\mathrm{cot}} g_k(x)\big). \end{aligned} \tag{11}$$

We first compute the Bayes conditional probabilities at each token position under $\mathcal{D}'$, and then their logistic coarsenings under restricted feature classes.

E.1 LOGISTIC COARSENINGS OF CONDITIONAL PROBABILITIES

To model next-token prediction with restricted features, we introduce a logistic approximation of the Bayes conditional probability over each token position. Let $(X, Y) \sim \mathcal{D}$, with $Y \in \{\pm 1\}$, and write $\eta(x) := \mathbb{P}(Y = +1 \mid X = x)$. Fix a feature map $\phi : \mathcal{X} \to \mathbb{R}^q$ satisfying $\mathbb{E}[\|\phi(X)\|_2] < \infty$, and let $\sigma(u) := (1 + e^{-u})^{-1}$.

We define the *coarsening* to describe the best fit to the data within a chosen class of logistic predictors. However, the logistic loss need not attain its minimum at finite weights. We bypass this by adding a regularization term with regularization strength $\lambda > 0$ in the objective that defines a coarsening. This, then, yields a unique finite minimizer w_λ for each $\lambda > 0$. We then let $\lambda \downarrow 0$ and define the coarsening through the limiting predicted probabilities, whenever these limits exist. This recovers the usual logistic-regression predictions when finite optimal weights exist, while also allowing probabilities equal to 0 or 1, which cannot be represented by finite logistic parameters. Formally, we have the following.

Definition 1 (Logistic coarsening)**.** *For each $\lambda > 0$, let w_λ be the unique minimizer*

$$w_\lambda := \operatorname*{arg\,min}_{w\in\mathbb{R}^q} \left\{ \mathbb{E}_{(X,Y)\sim\mathcal{D}} \left[\log\Big(1 + e^{-Y\langle w, \phi(X)\rangle}\Big) \right] + \frac{\lambda}{2}\|w\|_2^2 \right\}. \tag{12}$$

The logistic coarsening *of η with respect to ϕ under $\mathcal{D}_x$ is*

$$\eta_\phi(x) := \lim_{\lambda\downarrow 0} \sigma(\langle w_\lambda, \phi(x)\rangle), \tag{13}$$

provided this limit exists for $\mathcal{D}_x$-almost every x.

The following lemma gives a moment-matching identity that we will use to compute the coarsenings for specific feature classes.

Lemma 1 (Moment-matching certificate)**.** *Suppose that $\mathcal{D}_x$ has finite support. Let $h : \mathcal{X} \to [0, 1]$ satisfy*

$$\mathbb{E}_{X\sim\mathcal{D}_x}\, [(h(X) - \eta(X))\phi(X)] = 0.$$

If there exists a sequence $w_j \in \mathbb{R}^q$ such that

$$\sigma(\langle w_j, \phi(x)\rangle) \longrightarrow h(x) \qquad \textit{for every } x \in \operatorname{supp}(\mathcal{D}_x),$$

then the logistic coarsening exists and equals h on $\operatorname{supp}(\mathcal{D}_x)$.

Proof. Write $f_w(x) := \langle w, \phi(x)\rangle$, and let $R(w)$ denote the unregularized logistic risk. Moment matching gives

$$\begin{aligned} R(w) &= \mathbb{E}\left[\log(1+e^{f_w(X)}) - h(X)f_w(X)\right] \\ &= \mathbb{E}[H(h(X))] + \mathbb{E}\left[\mathrm{kl}\big(h(X), \sigma(f_w(X))\big)\right], \end{aligned}$$

where H is binary entropy and $\mathrm{kl}(a,b)$ is the KL divergence from $\mathrm{Ber}(a)$ to $\mathrm{Ber}(b)$. Finite support and the approximating sequence imply

$$\inf_w R(w) = \mathbb{E}[H(h(X))].$$

For the regularized minimizer w_λ and every fixed w, optimality gives

$$\inf R \le R(w_\lambda) \le R(w) + \frac{\lambda}{2}\|w\|_2^2.$$

Letting $\lambda \downarrow 0$ and then taking the infimum over w shows that $R(w_\lambda) \to \inf R$. The expected KL divergence therefore tends to zero. Since every support point has positive mass, $\sigma(f_{w_\lambda}(x)) \to h(x)$ at each such point, as required. □

We next compute the Bayes conditional probabilities at each token position under the shifted distribution and their logistic coarsenings for several feature classes.

E.2 Position 1

Write $p := p_{\mathrm{cot}}$, and assume $0 < p < 1$ and $1 \le k < m$, where $m = d/2$. All probabilities in this subsection refer to the shifted distribution in (10). Recall that

$$g_k(x) := \chi_m(x)\chi_{m-k}(x) = \prod_{i=m-k+1}^{m} x_i, \qquad \mathcal{D}'_x(x) = \mathcal{D}_x(x)\big(1 + p g_k(x)\big).$$

We first compute the first-token conditional probability and then its logistic coarsenings under restricted feature classes.

Lemma 2 (First-token conditional probability)**.** *Let* $\eta^{(1)}(x) := \mathbb{P}(y_1 = +1 \mid x)$. *For every* $x \in \{\pm 1\}^d$,

$$\eta^{(1)}(x) = \frac{1}{2} + \frac{p x_1\big(1 + g_k(x)\big) + \big(1 - p g_k(x)\big)\chi_{m-k}(x)}{2(1+p)}. \tag{14}$$

Proof. Let Z indicate the long component of the shifted distribution. Since $\mathbb{P}(Z=1) = p$ and the input distribution on this component is $\mathcal{D}_x(\cdot \mid g_k = 1)$, Bayes' rule gives

$$\rho(x) := \mathbb{P}(Z = 1 \mid x) = \frac{p\big(1 + g_k(x)\big)}{1 + p g_k(x)} = \frac{p\big(1 + g_k(x)\big)}{1+p}.$$

The last equality follows by noticing that $g_k(x) \in \{\pm 1\}$ and that if $g_k(x) = -1$ the numerator is 0. On the long branch, $y_1 = x_1$; on the short branch, $y_1 = \chi_{m-k}(x)$. Therefore,

$$\eta^{(1)}(x) = \rho(x)\frac{1 + x_1}{2} + \big(1 - \rho(x)\big)\frac{1 + \chi_{m-k}(x)}{2}.$$

Substituting the expression for $\rho(x)$ yields (14). □

For $S \subseteq [d]$, write $\chi_S(x) := \prod_{i \in S} x_i$. We consider the following classes of logistic scores:

$$\begin{aligned} \mathcal{F}_c &:= \mathrm{span}\{1\}, && \text{(constant)}, \\ \mathcal{F}_l &:= \mathrm{span}\{x_1, \dots, x_d\}, && \text{(linear)}, \\ \mathcal{F}_k &:= \mathrm{span}\big\{\chi_S : S \subseteq [d],\ 1 \le |S| \le k+1\big\}, && \text{(degree-(k+1))}. \end{aligned}$$

The next result considers coarsenings of the first position conditional probability, according to the feature classes defined above.

Proposition 3 (First-token coarsenings). *Let $\eta_c^{(1)}$, $\eta_l^{(1)}$, and $\eta_k^{(1)}$ denote the logistic coarsenings of $\eta^{(1)}$ under $\mathcal{D}'_x$, associated with $\mathcal{F}_c$, $\mathcal{F}_l$, and $\mathcal{F}_k$, respectively.*

The constant coarsening is

$$\eta_c^{(1)}(x) = \frac{1}{2}.$$

If $m - k \geq 2$, the linear coarsening is

$$\eta_l^{(1)}(x) = \frac{1 + px_1}{2}. \tag{15}$$

If $k + 1 < m - k$, the degree-$(k+1)$ coarsening is

$$\eta_k^{(1)}(x) = \frac{1}{2} + \frac{px_1\big(1 + g_k(x)\big)}{2(1+p)}. \tag{16}$$

The constant coarsening reflects that the first token is balanced overall. With linear features, the predictor favors (x_1), the first token of the pre-trained reasoning trace, equally across all inputs. Features of degree up to $(k+1)$ allow this preference to depend on the agreement gate g_k, favoring (x_1) when the pre-training and post-training targets agree, and assigning equal probability to the two signs when they disagree.

Proof. For each class, let ϕ denote its feature map. By Lemma 1, it suffices to find a vector w such that $h(x) := \sigma(\langle w, \phi(x)\rangle)$ satisfies $\mathbb{E}_{\mathcal{D}'_x}\left[(\eta^{(1)}(X) - h(X))\phi(X)\right] = 0$.

We use the orthogonality of parity monomials under the uniform input distribution:

$$\mathbb{E}_{\mathcal{D}_x}[\chi_S(X)\chi_T(X)] = \mathbf{1}\{S = T\}.$$

Constant features. On the short branch, $y_1 = \chi_{m-k}(x)$ is balanced. On the long branch, $y_1 = x_1$ is also balanced, since g_k depends only on coordinates $m - k + 1, \ldots, m$. Consequently,

$$\mathbb{E}_{\mathcal{D}'_x}[\eta^{(1)}(X)] = \frac{1}{2}.$$

The predictor $h_c(x) = 1/2 = \sigma(0)$ therefore satisfies the moment equation.

Linear features. Consider $h_l(x) := (1 + px_1)/2$. It is realized by

$$h_l(x) = \sigma(\alpha x_1), \qquad \alpha = \log\left(\frac{1+p}{1-p}\right).$$

Using (14), we obtain

$$\big(1 + pg_k(x)\big)\big(\eta^{(1)}(x) - h_l(x)\big) = \frac{1-p}{2}\big(\chi_{m-k}(x) + px_1 g_k(x)\big).$$

The two monomials on the right have degrees $m - k \geq 2$ and $k + 1 \geq 2$, respectively. Hence, for every $j \in [d]$,

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}'_x}\Big[\big(\eta^{(1)}(X) - h_l(X)\big)X_j\Big] \\ &\quad= \frac{1-p}{2}\mathbb{E}_{\mathcal{D}_x}\left[\big(\chi_{m-k}(X) + pX_1 g_k(X)\big)X_j\right] = 0.\end{aligned}$$

Thus, h_l satisfies all the required moment equations.

Features of degree up to $k+1$. Let h_k denote the right-hand side of (16). It is realized by

$$h_k(x) = \sigma\left(\beta x_1\big(1 + g_k(x)\big)\right), \qquad \beta = \frac{1}{2}\log\left(\frac{1+3p}{1-p}\right).$$

This score belongs to $\mathcal{F}_k$, since $x_1 g_k(x)$ has degree $k+1$. Furthermore,

$$\big(1 + pg_k(x)\big)\big(\eta^{(1)}(x) - h_k(x)\big) = \frac{1-p}{2}\chi_{m-k}(x).$$

Therefore, for every $S \subseteq [d]$ with $1 \leq |S| \leq k+1$,

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}'_x}\Big[\big(\eta^{(1)}(X) - h_k(X)\big)\chi_S(X)\Big] \\ &\quad= \frac{1-p}{2}\mathbb{E}_{\mathcal{D}_x}\left[\chi_{m-k}(X)\chi_S(X)\right] = 0,\end{aligned}$$

where the last equality uses $k + 1 < m - k$. □

E.3 POSITION 2

We next compute the conditional probability of the second output token under the shifted distribution.

Lemma 3 (Second-token conditional probability)**.** *For every pair (x, y_1) with positive probability under the shifted distribution, let*

$$\eta^{(2)}(x, y_1) := \mathbb{P}(y_2 = +1 \mid x, y_1).$$

Then

$$\eta^{(2)}(x, y_1) = \frac{1 + x_1 x_2}{2} \mathbb{P}(Z = 1 \mid x, y_1). \tag{17}$$

Proof. On the long branch ($Z = 1$), the second token equals $x_1 x_2$. On the short branch ($Z = 0$), it equals <EOS>. Therefore, almost surely under the shifted distribution,

$$\mathbf{1}\{y_2 = +1\} = Z \, \frac{1 + x_1 x_2}{2}.$$

Taking conditional expectations given (x, y_1) yields the result. □

Similarly to Position 1, we study logistic coarsenings of the second-token conditional probability $\eta^{(2)}(x, y_1) = \mathbb{P}(y_2 = +1 \mid x, y_1)$. Write $p := p_{\mathrm{cot}}$, $s := m - k$, and $r := \max\{k, 2\}$, and assume $0 < p < 1$, $1 \le k < m$, and $s \ge 2$. Let Q denote the marginal distribution of (X, Y_1) under the shifted training distribution. The coarsenings are binary logistic fits to the indicator $B := \mathbf{1}\{y_2 = +1\}$.

For $u = (x_1, \ldots, x_d, y_1)$ and $S \subseteq [d + 1]$, write $u_S := \prod_{i \in S} u_i$, with $u_\emptyset := 1$. We consider the following classes of logistic scores:

$$\begin{aligned}
\mathcal{F}_c &:= \mathrm{span}\{1\}, \\
\mathcal{F}_l &:= \mathrm{span}\{1, u_1, \ldots, u_{d+1}\}, \\
\mathcal{F}_q &:= \mathrm{span}\{1, u_i u_j : 1 \le i < j \le d + 1\}, \\
\mathcal{F}_k &:= \mathrm{span}\{u_S : S \subseteq [d + 1],\ 0 \le |S| \le r\}.
\end{aligned}$$

Each displayed basis defines the feature map used for the corresponding logistic coarsening.

The exclusion of linear terms from $\mathcal{F}_q$ is relevant for $k = 1$: the agreement gate $g_1(x) = x_m$ is then linear and therefore absent from this score class. For $k = 2$, the gate $g_2(x) = x_{m-1} x_m$ is itself quadratic and remains available, leading to a separate case in the proposition.

Proposition 4 (Second-token coarsenings)**.** *Let $\eta_c^{(2)}$, $\eta_l^{(2)}$, $\eta_q^{(2)}$, and $\eta_k^{(2)}$ denote the logistic coarsenings associated with the four score classes above, under Q. All identities below hold Q-almost surely.*

The constant coarsening is

$$\eta_{\mathrm{c}}^{(2)}(x, y_1) = \frac{p}{2}. \tag{18}$$

The affine-linear coarsening is

$$\eta_l^{(2)}(x, y_1) = \begin{cases} \dfrac{p(1 + x_m)}{2(1 + p)}, & k = 1, \\ \dfrac{p}{2}, & k \ge 2. \end{cases} \tag{19}$$

If $m - k \ge 4$, the quadratic coarsening is

$$\eta_q^{(2)}(x, y_1) = \begin{cases} \dfrac{p(1 + x_1 y_1)(1 + x_1 x_2)}{2(1 + p)}, & k \ne 2, \\ \dfrac{p(1 + x_1 y_1)(1 + x_1 x_2)(1 + g_k(x))}{2(1 + 3p)}, & k = 2. \end{cases} \tag{20}$$

If $\max\{k, 2\} + 1 < m - k$, the coarsening associated with monomials of degree at most $\max\{k, 2\}$ is

$$\eta_k^{(2)}(x, y_1) = \frac{p(1 + x_1 y_1)(1 + x_1 x_2)(1 + g_k(x))}{2(1 + 3p)}. \tag{21}$$

Proof. Write $U = (X, Y_1)$ and $B := \mathbf{1}\{Y_2 = +1\}$. For any function φ of the prefix,

$$\mathbb{E}_Q[\eta^{(2)}(U)\varphi(U)] = \mathbb{E}[B\varphi(U)], \qquad B = Z\frac{1 + X_1X_2}{2}.$$

By Lemma 1, it suffices to find an h satisfying

$$\mathbb{E}_Q[(h(U) - \eta^{(2)}(U))\phi(U)] = 0$$

and a sequence of scores in the corresponding class whose logistic predictions converge pointwise to h on $\mathrm{supp}(Q)$.

Constant features. On the long branch, X_1X_2 is balanced, because the gate depends only on coordinates $m - k + 1, \ldots, m$, which exclude the first two coordinates. Therefore,

$$\mathbb{E}_Q[\eta^{(2)}(U)] = \mathbb{E}[B] = \frac{p}{2}.$$

The candidate $h_c = p/2$, realized by the constant score $\log(p/(2-p))$, satisfies the moment equation.

Affine-linear features. For $k \geq 2$, every coordinate of U has mean zero. Furthermore,

$$\mathbb{E}[BX_j] = 0 \quad (j \in [d]), \qquad \mathbb{E}[BY_1] = 0.$$

These identities follow from parity orthogonality, since the gate has degree at least two and is disjoint from X_1, X_2, while $Y_1 = X_1$ on the long branch. Thus, $h_l = p/2$ matches all affine-linear moments.

For $k = 1$, the gate is X_m. We have

$$Q(X_m = 1) = \frac{1+p}{2}, \qquad \mathbb{E}[BX_m] = \mathbb{E}[B] = \frac{p}{2},$$

while $\mathbb{E}[BX_j] = 0$ for $j \neq m$ and $\mathbb{E}[BY_1] = 0$. Conditional on $X_m = 1$, all other coordinates of U have mean zero. Consequently,

$$h_l(x, y_1) := \frac{p}{1+p}\mathbf{1}\{x_m = 1\}$$

matches the constant moment, the X_m moment, and every other linear moment. It is approached by

$$\sigma\big(\log p + M(x_m - 1)\big), \qquad M \to \infty.$$

Quadratic features when $k \neq 2$. Assume $m - k \geq 4$, and let

$$E := \{y_1 = x_1,\ x_2 = x_1\}.$$

Every positive label lies in E. Let ν denote the uniform distribution on $(x_1, x_3, \ldots, x_d)$. For a scalar feature φ, let ψ be its restriction obtained by substituting $x_2 = y_1 = x_1$.

On the short branch, the additional requirement $y_1 = \chi_{m-k}(x)$ becomes

$$x_1 \prod_{i=3}^{m-k} x_i = 1.$$

Separating the long and short components therefore gives

$$\begin{aligned}\mathbb{E}_Q[\mathbf{1}_E\varphi(U)] &= \frac{p}{2}\mathbb{E}_\nu[(1 + g_k(X))\psi] \\ &\quad + \frac{1-p}{4}\mathbb{E}_\nu\left[\left(1 + X_1\prod_{i=3}^{m-k} X_i\right)\psi\right], \\ \mathbb{E}[B\varphi(U)] &= \frac{p}{2}\mathbb{E}_\nu[(1 + g_k(X))\psi].\end{aligned}$$

The factor $1/2$ in the long component comes from $X_2 = X_1$; the short component has an additional parity constraint, giving the factor $1/4$.

Consider

$$h_q(x, y_1) := \frac{2p}{1+p}\mathbf{1}_E = \frac{p(1+x_1y_1)(1+x_1x_2)}{2(1+p)}.$$

The preceding identities imply

$$\mathbb{E}_Q[(h_q(U) - \eta^{(2)}(U))\varphi(U)] = \frac{p(1-p)}{2(1+p)}\mathbb{E}_\nu\left[\left(X_1\prod_{i=3}^{m-k}X_i - g_k(X)\right)\psi\right].$$

For a basis feature of $\mathcal{F}_q$, the restriction ψ is either constant or a quadratic monomial. The two parities in parentheses have degrees $m-k-1 \geq 3$ and $k \neq 2$, respectively. Hence the expectation vanishes by parity orthogonality, verifying every feature moment.

Finally, h_q is approached by the quadratic scores

$$f_M(x, y_1) = \log\left(\frac{2p}{1-p}\right) + M(x_1y_1 + x_1x_2 - 2), \qquad M \to \infty.$$

Features that include the agreement gate. Let

$$G := \{y_1 = x_1,\ x_2 = x_1,\ g_k(x) = 1\},$$

and consider

$$h_k(x, y_1) := \frac{4p}{1+3p}\mathbf{1}_G = \frac{p(1+x_1y_1)(1+x_1x_2)(1+g_k(x))}{2(1+3p)}.$$

Using the same restriction ψ and distribution ν, the additional gate constraint gives

$$\mathbb{E}_Q[\mathbf{1}_G\varphi(U)] = \frac{p}{2}\mathbb{E}_\nu[(1+g_k(X))\psi] + \frac{1-p}{8}\mathbb{E}_\nu\left[\left(1 + X_1\prod_{i=3}^{m-k}X_i\right)(1+g_k(X))\psi\right].$$

It follows that

$$\mathbb{E}_Q[(h_k(U) - \eta^{(2)}(U))\varphi(U)] = \frac{p(1-p)}{2(1+3p)}\mathbb{E}_\nu\left[X_1\prod_{i=3}^{m-k}X_i\,(1+g_k(X))\psi\right].$$

Substituting $x_2 = y_1 = x_1$ cannot increase a monomial's degree. Thus, if φ belongs to the basis of $\mathcal{F}_k$, then ψ has degree at most $\max\{k, 2\}$. The two parities multiplying ψ have degrees $m-k-1$ and $m-1$, because the gate and $X_1\prod_{i=3}^{m-k}X_i$ have disjoint supports. Under $\max\{k, 2\} + 1 < m - k$, both degrees exceed that of ψ, so every feature moment vanishes.

The candidate h_k is approached by the scores

$$f_M(x, y_1) = \log\left(\frac{4p}{1-p}\right) + M(x_1y_1 + x_1x_2 + g_k(x) - 3), \qquad M \to \infty.$$

These scores belong to $\mathcal{F}_k$, establishing (21).

Finally, when $k = 2$ and $m - k \geq 4$, the same moment calculation applies to every quadratic feature. Moreover, $g_2(x) = x_{m-1}x_m$, so the displayed scores also belong to $\mathcal{F}_q$. This proves its remaining case and completes the proof. □

E.4 Positions 3-m

For positions $3, \ldots, m$, any nonterminated prefix in the support of the shifted distribution belongs to the long branch. The remaining tokens therefore follow the deterministic parity computation.

Lemma 4 (Later-token conditional probabilities)**.** *Let $3 \le \ell \le m$. For every nonterminated prefix $(x, y_{<\ell})$ with positive probability under the shifted distribution, the conditional probability*

$$\eta^{(\ell)}(x, y_{<\ell}) := \mathbb{P}(Y_\ell = +1 \mid X = x, Y_{<\ell} = y_{<\ell})$$

satisfies

$$\eta^{(\ell)}(x, y_{<\ell}) = \frac{1 + y_{\ell-1}x_\ell}{2}.$$

Proof. A short response terminates at position 2. Thus, a nonterminated prefix reaching position $\ell \ge 3$ must belong to the long branch. On this branch,

$$y_{\ell-1} = \chi_{\ell-1}(x), \qquad Y_\ell = \chi_\ell(x) = y_{\ell-1}x_\ell.$$

The next token is therefore deterministic, giving the stated conditional probability. □

At positions $3, \dots, m$, the quadratic feature $y_{\ell-1}x_\ell$ suffices to represent the deterministic transition in the logistic limit.

F K-TH DEGREE FEATURES

Using the preceding calculations of coarsened conditional probabilities across token positions, we can now compute sequence-level probabilities, which in turn affect the RL losses in subsequent rounds. In this section, we focus on the "rich" coarsening case, where we have access to degree-k features.

Sequence-level probabilities Write $r := \max\{k, 2\}$, and assume $r + 1 < m - k$. At Position 1, the score class is the span of the input monomials of degrees $1, \dots, k + 1$. At Position 2, it is the span of all monomials of degree at most r in (x, y_1), including the constant. At Positions $3, \dots, m$, the score class contains $y_{\ell-1}x_\ell$. After continuation at Position 2, the model generates binary tokens until Position m and terminates at Position $m + 1$.

The following lemma characterizes the sequence-level probabilities for a large family of gate-dependent shifted distributions. This family includes the case $n = 1$ as a special case and, as we will later see, consists of exactly those shifted distributions that can arise in subsequent RL rounds.

Lemma 5 (Coarsenings and sequence probabilities under gate-dependent shifts)**.** *Consider the training distribution*

$$\begin{aligned}\mathcal{D}_{A,B}(x, y) = \mathcal{D}_x(x)\Big[&A(g_k(x))\,\mathbf{1}\{y = (\chi_{m-k}(x), \texttt{<EOS>})\} \\ &+ B(g_k(x))\,\mathbf{1}\{y = (x_1, x_1x_2, \dots, \chi_m(x), \texttt{<EOS>})\}\Big],\end{aligned} \tag{22}$$

where $A(g) > 0$, $B(g) \ge 0$ for $g \in \{\pm 1\}$, B is not identically zero, and $\mathbb{E}_{\mathcal{D}_x}[A(g_k(X)) + B(g_k(X))] = 1$. Define

$$q(g) := \frac{B(g)}{A(g) + B(g)}.$$

The first- and second-token logistic coarsenings give the following probabilities on their respective training supports:

$$\begin{aligned}\pi_h(t \mid x) &= \frac{1 + tx_1 q(g_k(x))}{2}, \\ \pi_h(t \mid x, y_1) &= \frac{2q(g_k(x))}{1 + q(g_k(x))}\,\mathbf{1}\{y_1 = x_1,\ t = x_1x_2\}, \qquad t \in \{\pm 1\}, \\ \pi_h(\texttt{<EOS>} \mid x, y_1) &= 1 - \frac{2q(g_k(x))}{1 + q(g_k(x))}\,\mathbf{1}\{y_1 = x_1\}.\end{aligned} \tag{23}$$

Define the policy by using these formulas at every binary prefix, including those outside the training support, and the fitted binary transitions at later positions. Then, for every $x \in \{\pm 1\}^d$,

$$\begin{aligned}\pi_h\big((x_1, x_1x_2, \dots, \chi_m(x), \texttt{<EOS>}) \mid x\big) &= q(g_k(x)), \\ \pi_h\big((t, \texttt{<EOS>}) \mid x\big) &= \frac{1 - q(g_k(x))}{2}, \qquad t \in \{\pm 1\}.\end{aligned} \tag{24}$$

All other output sequences have probability zero.

Proof. Write $s := m - k$. We apply Lemma 1, using the fact that every function of $g_k \in \{\pm 1\}$ is affine in g_k.

First position. The input marginal is

$$\mathcal{D}_{A,B,x}(x) = \mathcal{D}_x(x)(A(g_k(x)) + B(g_k(x))).$$

The Bayes conditional probability is

$$\eta^{(1)}(x) = \frac{1}{2} + \frac{A(g_k(x))\chi_s(x) + B(g_k(x))x_1}{2(A(g_k(x)) + B(g_k(x)))}.$$

For $h(x) := (1 + x_1 q(g_k(x)))/2$ and any first-position basis feature φ,

$$\mathbb{E}_{\mathcal{D}_{A,B,x}}[(\eta^{(1)}(X) - h(X))\varphi(X)] = \frac{1}{2}\mathbb{E}_{\mathcal{D}_x}[A(g_k(X))\chi_s(X)\varphi(X)] = 0.$$

Indeed, $A(g_k)\chi_s$ is a linear combination of χ_s and χ_m, whose degrees exceed $\deg\varphi \le k+1$. Parity orthogonality therefore applies.

Since $0 \le q(g) < 1$, the candidate is realized by the finite score

$$x_1 \log\left(\frac{1 + q(g_k(x))}{1 - q(g_k(x))}\right).$$

This score lies in the span of x_1 and $x_1 g_k(x)$, so it belongs to the first-position class. The certificate identifies h as the coarsening; the other sign has the complementary probability.

Second position. Let Q be the law of $U = (X, Y_1)$ under $\mathcal{D}_{A,B}$, and fix $t \in \{\pm 1\}$. Write

$$\eta_t(U) := \mathbb{P}(Y_2 = t \mid U), \qquad E_t := \{y_1 = x_1,\ x_2 = tx_1\}, \qquad c(g) := \frac{2B(g)}{A(g) + 2B(g)}.$$

Let ν be uniform on $(X_1, X_3, \ldots, X_d)$, and let ψ_t be a basis feature φ restricted by $y_1 = x_1$ and $x_2 = tx_1$. Define

$$v(X) := X_1 \prod_{i=3}^{s} X_i.$$

On the short branch, the additional constraint $y_1 = \chi_s(x)$ becomes $tv(x) = 1$. Consequently, suppressing the argument $g_k(X)$ in A, B, c,

$$\mathbb{E}_Q[\mathbf{1}_{E_t}\varphi] = \frac{1}{2}\mathbb{E}_\nu[B\psi_t] + \frac{1}{4}\mathbb{E}_\nu[A(1+tv)\psi_t],$$
$$\mathbb{E}_Q[\eta_t\varphi] = \frac{1}{2}\mathbb{E}_\nu[B\psi_t].$$

For $h_t(x, y_1) := c(g_k(x))\mathbf{1}_{E_t}$, the identity $c(A + 2B) = 2B$ gives

$$\mathbb{E}_Q[(h_t - \eta_t)\varphi] = \frac{t}{4}\mathbb{E}_\nu[cA\, v\, \psi_t] = 0.$$

To see the last equality, note that cA is affine in g_k. The resulting parity factors v and vg_k have degrees $s-1$ and $m-1$, respectively, whereas $\deg\psi_t \le r < s-1$.

To handle both positive and zero values of B, set

$$b_M(g) := \log\left(\frac{2(B(g) + e^{-M})}{A(g)}\right).$$

The scores

$$f_{M,t}(x, y_1) := b_M(g_k(x)) + M(x_1 y_1 + tx_1 x_2 - 2)$$

belong to the second-position class and satisfy $\sigma(f_{M,t}) \to h_t$ pointwise as $M \to \infty$. Indeed, $b_M(g_k(x))$ has degree at most k; where $B(g) > 0$, its limit is $\log(2B(g)/A(g))$, and where $B(g) = 0$, it tends to $-\infty$. The certificate therefore applies to both signs.

The EOS candidate $h_{\mathrm{EOS}} := 1 - h_{+1} - h_{-1}$ matches its label moments by linearity. It is approached by the sigmoid of

$$-b_M(g_k(x)) - M(x_1 y_1 - 1),$$

which also belongs to the score class. Thus it is the corresponding binary coarsening. The three probabilities in (23) are nonnegative and sum to one.

Later positions and sequence probabilities. Every nonterminated training prefix at Position $\ell \geq 3$ belongs to the old long trace. Its next sign is $Y_{\ell-1}X_\ell$, realized in the logistic limit by $MY_{\ell-1}X_\ell$. Hence the certificate recovers this deterministic transition on the training support.

The second-token policy continues only if $y_1 = x_1$ and $B(g_k(x)) > 0$, and then emits $x_1 x_2$. For every such input, the old long trace has positive training mass. All generated long prefixes are therefore supported, so the fitted later transitions produce the old trace, followed by termination.

For a fixed input, abbreviate $q = q(g_k(x))$. The long-sequence probability is

$$\frac{1+q}{2}\frac{2q}{1+q} = q.$$

The short response beginning with x_1 has probability

$$\frac{1+q}{2}\left(1 - \frac{2q}{1+q}\right) = \frac{1-q}{2}.$$

The first token $-x_1$ has probability $(1-q)/2$ and is followed by certain termination. These three sequence probabilities sum to one. □

The first shifted distribution. Taking $A(g) = 1 - p$ and $B(g) = p(1+g)$ recovers the first accepted distribution. In this case,

$$q(g) = \frac{p(1+g)}{1+pg} = \frac{p(1+g)}{1+p}, \qquad \frac{1-q(g)}{2} = \frac{1-pg}{2(1+p)},$$

which gives the preceding first-round sequence probabilities.

RL recursion. At every round, inputs are sampled from the original uniform distribution $\mathcal{D}_x$. Conditioning on successful responses changes the accepted training distribution, whose evolution is described by the following result.

Corollary 1. *Assume $0 < p < 1$, and start from the initial policy (2). Let $\mathcal{D}_{A_n,B_n}$ be the training distribution at the n'th RL round, and let π_n be the corresponding policy. For every $n \geq 1$, the long-sequence probabilities satisfy*

$$q_n(-1) = 0, \qquad q_n(1) = \frac{2^n p}{1 - p + 2^n p} \longrightarrow 1,$$

and the sampled accuracy is

$$\mathrm{Acc}(\pi_n) = \frac{1}{2} + \frac{q_n(1)}{4} \longrightarrow \frac{3}{4}. \tag{25}$$

Proof. The first accepted distribution has

$$A_1(g) = 1 - p, \qquad B_1(g) = p(1+g).$$

These weights satisfy the assumptions of Lemma 5 and give the stated initial values.

Suppose that the accepted distribution at round n has the required form and $B_n(-1) = 0$. By the lemma, for each input x, the correct short response has probability

$$\frac{1 - q_n(g_k(x))}{2},$$

and the old long trace has probability $q_n(g_k(x))$. Since $q_n(-1) = 0$, every generated long response is correct. The conditional success probability is therefore

$$\Pr(\text{success} \mid x) = \frac{1 + q_n(g_k(x))}{2}.$$

Let

$$Z_n := \mathbb{E}_{X\sim\mathcal{D}_x}\left[\frac{1+q_n(g_k(X))}{2}\right].$$

Conditioning on success yields

$$A_{n+1}(g) = \frac{1-q_n(g)}{2Z_n}, \qquad B_{n+1}(g) = \frac{q_n(g)}{Z_n}.$$

These weights are normalized, satisfy $A_{n+1}(g) > 0$ at every finite round, and preserve $B_{n+1}(-1) = 0$. Thus, the next accepted distribution again belongs to the same family.

Applying the lemma gives

$$q_{n+1}(g) = \frac{B_{n+1}(g)}{A_{n+1}(g)+B_{n+1}(g)} = \frac{2q_n(g)}{1+q_n(g)}.$$

On agreement inputs,

$$\frac{1-q_{n+1}(1)}{q_{n+1}(1)} = \frac{1}{2}\frac{1-q_n(1)}{q_n(1)},$$

which gives the explicit formula for $q_n(1)$. Finally, since the two gate values are equally likely,

$$Z_n = \frac{1}{2}\frac{1+q_n(1)}{2} + \frac{1}{2}\frac{1}{2} = \frac{1}{2} + \frac{q_n(1)}{4}.$$

This proves the accuracy formula and its limit. □

In particular, the result reproduces what we saw in the simulations; gated on $g_k(x) = +1$, the probability of generating long responses goes to 1, which implies that the accuracy of the policy goes to 0.75.

G LINEAR FEATURES, $k \geq 2$.

In this section, we repeat the previous analysis, this time assuming coarsenings induced by linear features.

We use linear scores at Position 1 and affine-linear scores at Position 2:

$$\mathcal{F}_l^{(1)} = \mathrm{span}\{x_1,\ldots,x_d\}, \qquad \mathcal{F}_l^{(2)} = \mathrm{span}\{1,x_1,\ldots,x_d,y_1\}.$$

At later positions, we retain the effective transition $Y_\ell = Y_{\ell-1}x_\ell$, for $3 \leq \ell \leq m$, followed by termination at Position $m+1$. The fitted first-two-token formulas and this recurrence are used at every generated binary prefix. For $t,u \in \{\pm 1\}$, write

$$L_{t,u}(x) := \left(t, u, ux_3, \ldots, u\prod_{i=3}^{m} x_i, \texttt{<EOS>}\right).$$

Proposition 5 (Linear coarsening). *Assume $k \geq 2$, $m-k \geq 3$, and $0 < p < 1$. Start from the initial policy (2). Let π_n denote the resulting policy after n rounds of population RL, using the linear coarsenings specified above. For $n \geq 1$, and every input x and $t,u \in \{\pm 1\}$,*

$$\begin{aligned}\pi_n((t, \texttt{<EOS>}) \mid x) &= \frac{(1-p)(1+ptx_1)}{2},\\ \pi_n(L_{t,u}(x) \mid x) &= \frac{p(1+ptx_1)}{4}.\end{aligned} \tag{26}$$

All other sequences have probability zero. In particular,

$$\mathrm{Acc}(\pi_n) = \frac{1}{2} \quad (n \geq 1),$$

Proof. Write $s := m - k$.

The first accepted distribution has short-branch weight $1 - p$ with uniform inputs and long-branch weight p with inputs conditioned on $g_k = 1$. The preceding linear-coarsening calculations give

$$\pi_1(t \mid x) = \frac{1 + ptx_1}{2}, \qquad \pi_1(+1 \mid x, y_1) = \frac{p}{2}.$$

The same moment calculation gives $p/2$ for the second-token label -1, and the EOS coarsening is $1 - p$. The chain rule and the assumed later recurrence give (26). Summing over the two short and four long traces gives total probability $(1 - p) + p = 1$.

Consider any policy with these token probabilities. A long trace has the correct final answer precisely when

$$u = \chi_s(x) \prod_{i=3}^{m} x_i = x_1 x_2 g_k(x).$$

Exactly one of the two equally likely second-token signs satisfies this condition. Thus

$$\Pr(\text{success} \mid x) = \frac{(1-p)(1 + px_1\chi_s(x))}{2} + \frac{p}{2} = \frac{1}{2} + \frac{p(1-p)}{2} x_1 \chi_s(x).$$

Since $x_1\chi_s(x) = \prod_{i=2}^{s} x_i$ is balanced, the sampled accuracy is $1/2$.

Dividing the successful sequence masses by $1/2$, the next accepted distribution is

$$\begin{aligned} \mathcal{D}^+(x, y) = \mathcal{D}_x(x) \Bigg[& (1-p)(1 + px_1\chi_s(x)) \mathbf{1}\{y = (\chi_s(x), \texttt{<EOS>})\} \\ & + \frac{p}{2} \sum_{t \in \{\pm 1\}} (1 + ptx_1) \mathbf{1}\{y = L_{t, x_1 x_2 g_k(x)}(x)\} \Bigg]. \end{aligned} \tag{27}$$

Its input marginal is

$$\mathcal{D}_x^+(x) = \mathcal{D}_x(x)\big(1 + p(1-p)x_1\chi_s(x)\big).$$

We now verify that refitting this shifted distribution recovers the same token probabilities.

First-position fit. For any first-position basis feature φ, direct summation in (27) gives

$$\begin{aligned} \mathbb{E}_{\mathcal{D}^+}[Y_1\varphi(X)] &= \mathbb{E}_{\mathcal{D}_x}[((1-p)\chi_s(X) + pX_1)\varphi(X)], \\ \mathbb{E}_{\mathcal{D}_x^+}[pX_1\varphi(X)] &= \mathbb{E}_{\mathcal{D}_x}[(pX_1 + p^2(1-p)\chi_s(X))\varphi(X)]. \end{aligned}$$

Their difference is

$$(1-p)(1-p^2)\mathbb{E}_{\mathcal{D}_x}[\chi_s(X)\varphi(X)] = 0$$

by parity orthogonality. Hence $h(x) = (1 + px_1)/2$ matches the first-token label moments. It is realized by the score

$$x_1 \log\left(\frac{1+p}{1-p}\right).$$

Lemma 1 identifies it as the coarsening.

Second-position fit. Let Q^+ be the law of (X, Y_1) under $\mathcal{D}^+$. Because $s \geq 3$, the parity $x_1\chi_s(x)$ has degree at least two. Consequently,

$$\mathbb{E}_{Q^+}[X_j] = 0 \quad (j \in [d]), \qquad \mathbb{E}_{Q^+}[Y_1] = 0.$$

On the long branch of $\mathcal{D}^+$, inputs are uniform,

$$Y_2 = X_1 X_2 g_k(X), \qquad \mathbb{E}[Y_1 \mid X, \text{long}] = pX_1.$$

The parity $X_1X_2g_k(X)$ has degree $k + 2$. It follows, for each $u \in \{\pm 1\}$, that

$$\begin{aligned} \mathbb{E}_{\mathcal{D}^+}[\mathbf{1}\{Y_2 = u\}] &= \frac{p}{2}, \\ \mathbb{E}_{\mathcal{D}^+}[\mathbf{1}\{Y_2 = u\}X_j] &= 0 \quad (j \in [d]), \\ \mathbb{E}_{\mathcal{D}^+}[\mathbf{1}\{Y_2 = u\}Y_1] &= 0. \end{aligned}$$

Thus the constant predictor $p/2$ matches every affine-linear feature moment for either sign. It is realized by the constant score $\log(p/(2-p))$. The EOS predictor $1-p$ matches its moments by subtraction and is also realized by a finite constant score. The moment-matching certificate therefore recovers all three second-token probabilities.

Every accepted long trace satisfies $Y_\ell = Y_{\ell-1}X_\ell$ for $3 \le \ell \le m$. The later coarsenings recover this deterministic transition on their training supports, and the stipulated extension uses the same recurrence at every generated prefix. Refitting therefore returns the same effective policy. Induction proves the claim for all $n \ge 1$. □

That is, we showed that linear features cannot induce progress in the accuracy of the post-trained model.

H LAST-TOKEN NOISE

Finally, in this section we consider an extension of our previous setting to incorporate the possibility of generating *atypical* sequences after pre-training. As we shall see, even a small amount of mass on atypical generations can guarantee that post-training will succeed.

We return to the degree-dependent feature classes and assume $r := \max\{k, 2\}$ satisfies $r+1 < m-k$. Our main assumption is to modify the initial, pre-trained policy only at the final answer of a long response. With probability $p \in (0,1)$, it generates

$$\big(\chi_1(x), \dots, \chi_{m-1}(x), N\chi_m(x), \texttt{<EOS>}\big),$$

where N is independent of X and $\Pr(N=-1) = \varepsilon \in (0,1)$. Each short response $(t, \texttt{<EOS>})$, $t \in \{\pm 1\}$, still has probability $(1-p)/2$.

The feature classes at Positions 1 and 2, and the extension of their fitted probabilities to every binary prefix, are as in Lemma 5. At Positions $3, \dots, m-1$, the quadratic transition feature $y_{\ell-1}x_\ell$ remains available. At the final answer position m, we allow all monomials of degree at most r in (x, y_{m-1}). Long responses terminate at Position $m+1$. Write

$$L^\star(x) := \big(\chi_1(x), \dots, \chi_{m-1}(x), \chi_{m-k}(x), \texttt{<EOS>}\big)$$

for the old long trace with its final answer corrected to the post-training target.

Proposition 6. *Under the assumptions above, let π_n be the resulting policy after the n-th RL round, with $n \ge 1$. Then*

$$\begin{aligned} \pi_n(L^\star(x) \mid x) &= q_n(g_k(x)), \\ \pi_n((t, \texttt{<EOS>}) \mid x) &= \frac{1 - q_n(g_k(x))}{2}, \qquad t \in \{\pm 1\}, \end{aligned} \tag{28}$$

and all other sequences have probability zero. Both gate-conditioned long-response probabilities satisfy

$$q_n(1) = \frac{2^n p(1-\varepsilon)}{1-p+2^n p(1-\varepsilon)}, \qquad q_n(-1) = \frac{2^n p\varepsilon}{1-p+2^n p\varepsilon}. \tag{29}$$

Consequently, the sampled accuracy satisfies

$$\mathrm{Acc}(\pi_n) = \frac{1}{2} + \frac{q_n(1) + q_n(-1)}{4} \longrightarrow 1. \tag{30}$$

Proof. A long response is successful exactly when $N = g_k(X)$. Since the gate is balanced under $\mathcal{D}_x$, both short and long responses have average success probability $1/2$. The initial success probability is therefore $1/2$. Dividing the successful trajectory masses by this probability gives

$$\begin{aligned} \widetilde{\mathcal{D}}_1(x,y) = \mathcal{D}_x(x)\Big[&(1-p)\mathbf{1}\{y = (\chi_{m-k}(x), \texttt{<EOS>})\} \\ &+ p\big(1 + (1-2\varepsilon)g_k(x)\big)\mathbf{1}\{y = L^\star(x)\}\Big]. \end{aligned}$$

Thus its short- and long-branch weights are

$$A_1(g) = 1 - p, \qquad B_1(g) = p\big(1 + (1 - 2\varepsilon)g\big).$$

Both long-branch weights are positive: $B_1(1) = 2p(1-\varepsilon)$ and $B_1(-1) = 2p\varepsilon$. Consider any distribution of the form

$$\widetilde{\mathcal{D}}_{A,B}(x,y) = \mathcal{D}_x(x)\Big[A(g_k(x))\mathbf{1}\{y = (\chi_{m-k}(x), \texttt{<EOS>})\} + B(g_k(x))\mathbf{1}\{y = L^\star(x)\}\Big],$$

where $A(g), B(g) > 0$ and $\mathbb{E}_{\mathcal{D}_x}[A(g_k(X)) + B(g_k(X))] = 1$. Its first-two-token training distributions are identical to those in Lemma 5. Hence their coarsenings are unchanged, with $q(g) = B(g)/(A(g) + B(g))$. The intermediate transitions are also unchanged.

The only new calculation concerns the final answer. On every supported long prefix, $Y_{m-1} = \chi_{m-1}(X)$, so

$$\chi_{m-k}(X) = Y_{m-1} \prod_{i=m-k+1}^{m-1} X_i.$$

This feature has degree $k \le r$, counting the previous-token variable and the $k-1$ input variables. For $k = 1$, the product is empty and equals one. For $M > 0$, let

$$f_M(x, y_{m-1}) := M y_{m-1} \prod_{i=m-k+1}^{m-1} x_i.$$

For all $M > 0$, f_M belongs to the final-position class, and as $M \to \infty$, the corresponding sigmoid predictors converge to the deterministic Bayes predictor. Lemma 1 shows that the final-token coarsening emits the correct answer $\chi_{m-k}(x)$.

The second-token policy continues only after $y_1 = x_1$ and then emits x_1x_2. Since $B(g) > 0$ on both gate values, all generated long prefixes through Position $m-1$ are supported by the training distribution. Thus no extension of the later-token identities is needed. The chain-rule calculation from the preceding lemma gives (28), with the old long trace replaced by $L^\star(x)$.

For the first refit, $q_1(g) = B_1(g)/(A_1(g) + B_1(g))$. Suppose a subsequent policy has the sequence law (28). Every long response is now correct, whereas the correct short response has probability $(1 - q_n(g_k(x)))/2$. Writing

$$Z_n := \mathbb{E}_{X \sim \mathcal{D}_x}\left[\frac{1 + q_n(g_k(X))}{2}\right],$$

the next accepted distribution again has the preceding form, with

$$A_{n+1}(g) = \frac{1 - q_n(g)}{2Z_n}, \qquad B_{n+1}(g) = \frac{q_n(g)}{Z_n}.$$

These weights remain positive at every finite round. Applying the extension above yields

$$q_{n+1}(g) = \frac{B_{n+1}(g)}{A_{n+1}(g) + B_{n+1}(g)} = \frac{2q_n(g)}{1 + q_n(g)}.$$

Equivalently,

$$\frac{1 - q_{n+1}(g)}{q_{n+1}(g)} = \frac{1}{2}\frac{1 - q_n(g)}{q_n(g)}.$$

Together with A_1, B_1, this gives (29). Finally, the conditional accuracy is $(1 + q_n(g_k(x)))/2$. Averaging over the two equally likely gate values gives (30). Both long-response probabilities tend to one. □