Defenses Against Stealing Reasoning Must Consider the Reasoning Boundary

Shidan Javaheri [1] **Oliver Britton** [1] **Yarin Gal** [1] **Yonatan Gideoni** [1]

Abstract

Distillation attacks are used to steal the reasoning capabilities of proprietary Large Language Models (LLMs), allowing malicious actors to replicate state-of-the-art model performance at low cost. These attacks steal reasoning by systematically gathering a large volume of frontier model reasoning traces and training LLMs on them using distillation, potentially followed by further training with reinforcement learning (RL). Although training pipelines combining distillation and RL are increasingly common, conflicting results exist comparing the effectiveness of RL and distillation as standalone techniques for improving a model's reasoning capabilities, obscuring the source of the value of frontier model reasoning traces in pipelines for stealing reasoning. This work reconciles this conflict by demonstrating that RL outperforms distillation in improving a model's ability to solve reasoning problems in a single attempt, but that distillation is a more efficient way to expand a model's reasoning boundary, increasing the complexity of the problems it can solve when given multiple attempts. Theoretically, this is explained by distillation being mode-covering while RL is mode-seeking. We conclude by demonstrating that defenses against stealing reasoning must evaluate a distilled model's performance at its reasoning boundary, which indicates its performance ceiling after potential subsequent RL training.

1. Introduction

In early 2025, DeepSeek-R1 demonstrated that a Large Language Model (LLM) could outperform those trained by existing industrial labs for the first time (Guo et al., 2025a). This led to allegations accusing DeepSeek of using information obtained from prior state-of-the-art models to improve its performance (Metz, 2025; Sweney & Milmo, 2025), allegations which were later rebutted (Guo et al., 2025b; Gibney, 2025), and then resurfaced again (Anthropic, 2026; Google, 2026; Seetharaman & Arámburo, 2026). These allegations represent the first instance of a distillation attack, an increasingly prevalent technique for stealing reasoning from frontier closed-source LLMs (Trockman & Savani, 2026). Distillation attacks execute targeted campaigns against the APIs of proprietary LLMs to systematically collect large quantities of their reasoning traces. With the aim of stealing their uniquely powerful reasoning capabilities at low cost, other LLMs are subsequently trained (i.e distilled) on these traces, potentially followed by further training through reinforcement learning (RL). These attacks, as illustrated in Figure 1, also pose security risks, allowing malicious actors to access powerful AI models without built-in safety guardrails (Trockman & Savani, 2026). This is particularly the case with the rise of Agentic AI, as powerful reasoning models can autonomously execute malicious objectives at scale (Google, 2026; Trockman & Savani, 2026).

Defenses against distillation attacks operate at two levels, both of which recognize that reasoning traces generated for attacks are difficult to distinguish from those in normal use cases. Defenses at the API level aim to detect abnormal traffic patterns and prevent malicious actors from amassing large quantities of high-quality reasoning traces (Anthropic, 2026; Google, 2026). Defenses at the model level aim to make reasoning traces less effective for use in pipelines that steal reasoning, without affecting normal use cases (Savani et al., 2025; Xu et al., 2026). Model-level defenses include obfuscating full reasoning traces via summarization, which is used on most commercial AI platforms, as well as more recent attempts to apply imperceptible perturbations to model output tokens (Savani et al., 2025; Xu et al., 2026). Evaluating the effectiveness of these model-level defenses is the focus of this work, and doing so requires a clear understanding of the training pipelines used to steal reasoning.

Distillation improves the performance of a student model by training it to match the outputs of a more capable teacher model (Hinton et al., 2015) and is widely considered to effectively improve a model's reasoning ability (Guo et al., 2025a; Yang et al., 2025). RL is a different training method that trains a model by generating responses to difficult ques-

[1]University of Oxford. Correspondence to: Shidan Javaheri <shidan@javaheri.org>, Yonatan Gideoni <yg@robots.ox.ac.uk>.

Second Workshop on Technical AI Governance Research (TAIGR) @ ICML 2026, Seoul, South Korea. 2026. Copyright 2026 by the author(s).

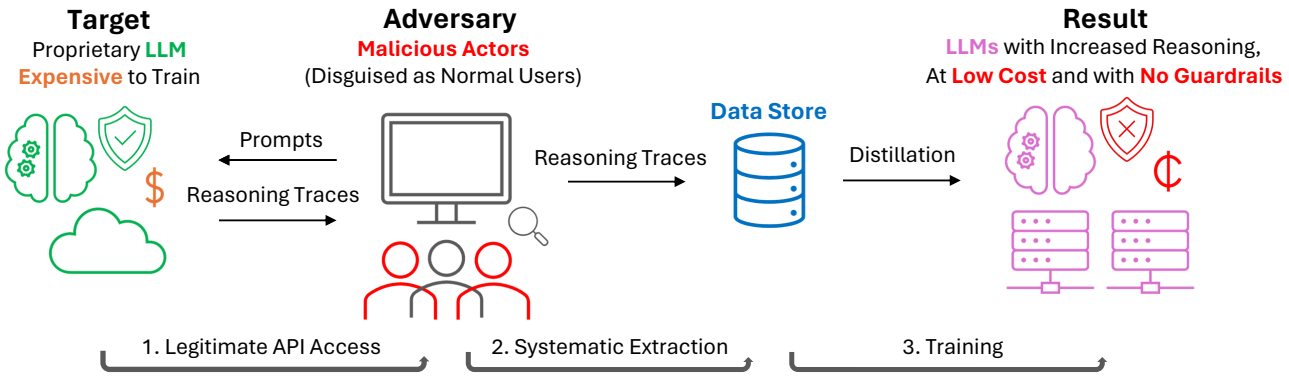

Figure 1. Illustration of distillation attacks for stealing reasoning, inspired by a similar figure in Google (2026).

tions and rewarding the model when it answers them correctly (Guo et al., 2025a; Zeng et al., 2025). The reasoning performance of an LLM is often evaluated on reasoning tasks using pass@k metrics (Chen et al., 2021), which calculate the percentage of questions in a dataset that a model answers correctly at least once in the k attempts it is given for every question. The pass@1 performance of a model measures how reliably it produces the correct answer in a single attempt, and is effectively its accuracy. This is also arguably the metric that pipelines to steal reasoning ultimately seek to increase. In contrast, for large k, pass@k performance reflects a model's *reasoning boundary* (Yue et al., 2025), indicating the limits of its ability to solve problems requiring reasoning.

Recent research has demonstrated that training with distillation followed by RL effectively improves LLMs' reasoning capabilities (Guo et al., 2025a; Yang et al., 2025; Liu et al., 2025b; Bercovich et al., 2025; Luo et al., 2025b;a), but conflicting results exist when comparing the two approaches as standalone techniques. Some studies report that distillation outperforms RL on pass@1 evaluations Yue et al. (2025); Hu et al. (2025), whereas others report the opposite Chu et al. (2025); Huan et al. (2025). Further, Guo et al. (2025a) observes both results at different model sizes. This conflict obscures the value of the reasoning traces of proprietary models in pipelines for stealing reasoning, which is essential to understand in order to both develop and evaluate effective defenses against stealing reasoning.

This work begins by articulating the conflicting results that exist in the literature when RL and distillation are compared as standalone techniques for improving the pass@1 reasoning performance of LLMs. Following the approach in similar work (Guo et al., 2025a; Yue et al., 2025; Chu et al., 2025), RL and distillation are compared on math reasoning tasks. These tasks are selected for their structured, verifiable solutions, which are necessary both for pass@k evaluations and for defining a reward signal for RL training. Using this framework, we make the following key contributions:

- We demonstrate that RL consistently outperforms distillation at improving a model's pass@1 math reasoning performance, reconciling conflicting observations that exist in current literature.
- We confirm that, despite this difference, distillation consistently expands a model's reasoning boundary. This indicates that the value of proprietary reasoning traces lies in their ability to efficiently expand the reasoning boundaries of student models through distillation.
- We offer a theoretical explanation for these results, demonstrating that they stem from the RL and distillation losses being mode-seeking and mode-covering, respectively. Mode-covering behavior can cause distillation to spread more probability mass across incorrect completions, lowering its pass@1 relative to RL, but also to place more probability mass on correct completions that previously had very little mass, often increasing its pass@k relative to RL.
- Building on these findings, we demonstrate that defenses against stealing reasoning must be evaluated against the full distillation-then-RL pipeline. This includes additional evaluations of student models on pass@k metrics, which, unlike pass@1 alone, better indicates the distilled model's post-RL performance ceiling.

2. RL and Distillation for Improving Reasoning

Distillation on the outputs of a teacher model to improve a student LLM's reasoning ability is most frequently done at the sequence level ("sequence knowledge distillation", Kim & Rush (2016)) with supervised finetuning (SFT) (Ouyang et al., 2022; Guo et al., 2025a; Yang et al., 2025). This is unlike typical knowledge distillation, where the student has access to the teacher's entire output distribution (Hinton

Table 1. Settings for the comparison of models trained with RL and distillation. All models are from the Qwen2.5 family, and all RL and teacher models are trained with the SimpleRL-Zoo framework.

SETTING	BASE MODEL	RL MODEL	TEACHER
0.5B	QWEN2.5-0.5B	QWEN2.5-0.5B-SIMPLERL-ZOO	QWEN2.5-1.5B-SIMPLERL-ZOO
1.5B	QWEN2.5-1.5B	QWEN2.5-1.5B-SIMPLERL-ZOO	QWEN2.5-7B-SIMPLERL-ZOO
7B	QWEN2.5-7B	QWEN2.5-7B-SIMPLERL-ZOO	QWEN2.5-14B-SIMPLERL-ZOO
MATH-7B	QWEN2.5-MATH-7B	QWEN2.5-MATH-7B-SIMPLERL-ZOO	QWEN2.5-14B-SIMPLERL-ZOO

et al., 2015). Reasoning traces, particularly those generated from proprietary models, only contain the output tokens of the teacher, with no information on the model's logits.

RL to improve reasoning is often done using GRPO (Shao et al., 2024a; Guo et al., 2025a; Zeng et al., 2025), where multiple answer trajectories from the model being trained are sampled. Correct answers are up-weighted while incorrect answers are down-weighted, leading to the model learning to output correct answers.

A consensus emerging in recent literature is that training pipelines should enhance the reasoning capabilities of LLMs by first distilling them on existing reasoning traces and then further training them with RL, potentially repeating this process several times (Guo et al., 2025a; Yang et al., 2025; Liu et al., 2025b; Bercovich et al., 2025; Luo et al., 2025b;a). Accordingly, several works show that distillation makes subsequent RL training more effective (Guo et al., 2025a; Yang et al., 2025; Narayanan et al., 2025; Liu et al., 2025b; Chu et al., 2025), with some arguing that it initializes the LLM with desirable behaviors that RL can more readily reinforce. Narayanan et al. (2025) further argues that both steps are important for developing an LLM's reasoning abilities, including in novel domains such as chemistry (Narayanan et al., 2025). Various results support these conclusions, demonstrating clear performance improvements at each stage of training pipelines that include RL and distillation (Narayanan et al., 2025; Guo et al., 2025a; Liu et al., 2025b). It is thus likely that distillation attacks would exploit a similar training pipeline to steal reasoning, first using SFT over the reasoning traces of frontier models, followed by subsequent RL training. However, current defenses against stealing reasoning are not evaluated with respect to the full distill-then-RL pipeline (Savani et al., 2025; Xu et al., 2026; Li et al., 2025), thereby implicitly assuming that any observed degradations would persist after RL training.

Furthermore, despite this consensus on using both distillation and RL to improve reasoning, conflicting results exist when comparing the two training schemes as stand-alone methods to improve a model's pass@1 performance on reasoning tasks. Several considerations support the intuition that distillation, given access to teacher reasoning traces, should outperform RL. Practically, distillation requires substantially less training compute than RL (Yang et al., 2025), making it the cheaper option for an attacker to scale up. Theoretically, learning from teacher trajectories provides a denser per-sample signal than RL's binary correctness reward, which has been argued to yield better sample efficiency (LeCun, 2016; Schulman & Lab, 2025). Furthermore, if the teacher is a frontier model, distillation may transfer capabilities derived from expensive proprietary training data that an attacker does not have access to themselves.

However, the empirical picture in the literature is mixed. Several studies support the conclusion that distillation outperforms RL (Yue et al., 2025; Hu et al., 2025), with Guo et al. (2025a) explicitly observing this in 32B models, but other work reaches contradictory conclusions. In the same paper, Guo et al. (2025a) note that in the training pipeline for larger models, RL outperforms distillation on pass@1 math reasoning tasks, and Chu et al. (2025); Huan et al. (2025) find RL achieves both stronger pass@1 performance and better generalization compared to distillation alone. These conflicting conclusions obscure the mechanism by which reasoning ability transfers from teacher to student models in pipelines to steal reasoning through distillation. In particular, if RL outperforms distillation in increasing pass@1 performance on reasoning tasks, the value of distillation in pipelines to steal reasoning is unlikely to be purely increased pass@1 performance. Instead, the value would be that distillation increases the performance ceiling of student models after RL training. Crucially, defenses evaluated solely for their ability to reduce pass@1 performance after distillation may prove ineffective after subsequent RL training.

3. Methodology for Comparing RL and Distillation

To disambiguate the performance differences between RL and distillation as training methods for improving reasoning, this work compares them in carefully controlled settings, isolating their effects on representative open-source base models of varying sizes. We introduce our methodology first here, and present and interpret the results in Section 4.

Our experiments use the Qwen2.5 family (Qwen et al., 2025) of models, chosen for its range of scales and its base models' strong math reasoning relative to similar-sized open-source alternatives such as Mistral-v0.1-7B (Jiang et al., 2023) and

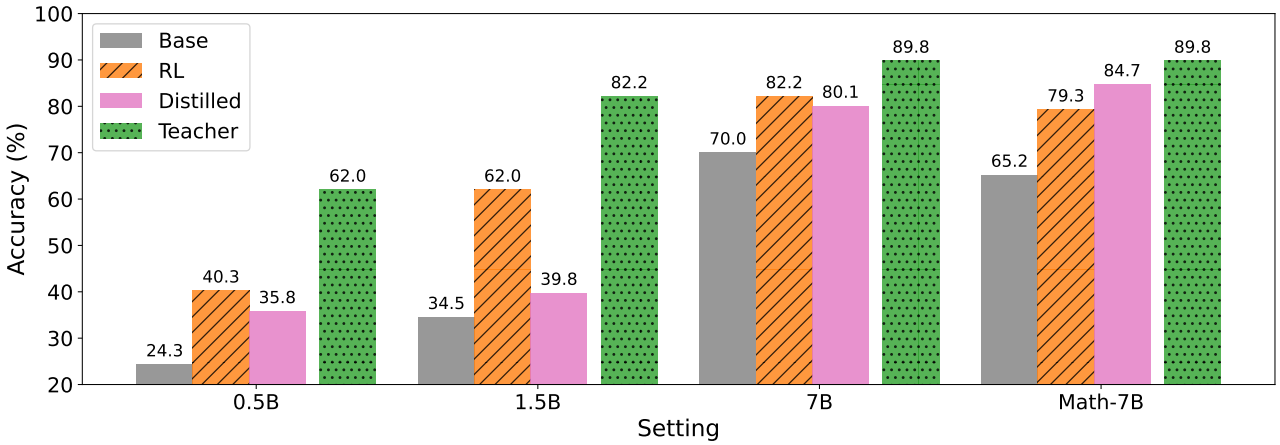

Figure 2. Comparison of base models trained with RL and distillation in the 4 settings shown in Table 1, reporting accuracy as an average over the GSM8K, Minerva and MATH500 datasets. RL-trained models outperform distilled ones in all settings except the Math-7B outlier. This outlier is caused by RL reinforcing a unique and undesirable pattern in this particular base model, which predominantly reasons with Python code. In contrast, distillation does not reinforce this pattern (see Appendix A.7 for details).

Llama-3.1-8B (Grattafiori et al., 2024). Strong base models are required for the RL comparison, since RL is constrained by the base model's initial distribution (see Section 5.1).

Four settings comprising models at scales from 0.5B to 7B are implemented for this comparison. Each setting includes a base model, a teacher model, and the variant of the base model trained with RL, with details for each one shown in Table 1. The RL-trained variant of each base model is chosen from SimpleRL-Zoo (Zeng et al., 2025), selected for its open access and clearly documented training pipeline with state-of-the-art results. Larger RL-trained models from the same work are used as teachers in distillation. The base model in each setting is then trained with distillation on outputs generated by each teacher model. The teacher generates reasoning traces from the same questions used in RL training, so that the two training methods use identical problems. Full details of the parameters used in distillation, as well as the ablations performed to ensure that the results were not affected by design choices, are presented in Appendices A.1 and A.2.

To reconcile conflicting observations in recent literature and to serve as a key indicator of a model's reasoning performance, RL and distillation are first compared for improving a model's pass@1 performance on math reasoning tasks. All models from each setting are evaluated on math datasets in the same way as in Zeng et al. (2025), using the same prompts as during RL training of each base model for a fair comparison, and with no in-context examples. Performance for all models is reported as an average of the pass@1 accuracy over the GSM8K (Cobbe et al., 2021), Minerva (Hendrycks et al., 2021), and MATH500 (Lightman et al., 2023; Zeng et al., 2025) datasets. Models are evaluated with a temperature of $T = 0$. Further details on the evaluation framework are provided in Appendix A.3.

To investigate the effects of RL and distillation on a model's reasoning boundary, experiments akin to those in Yue et al. (2025) are performed on models of all sizes, calculating their pass@k accuracy on the MATH500 dataset with values of k up to 128, using a temperature of $T = 1$. Further evaluation details are available in Section A.3.

4. Results of Comparing RL and Distillation

4.1. RL vs Distillation's Impact on a Model's Accuracy

Figure 2 shows that base models trained with RL consistently outperform those trained with distillation at all tested scales, save for the notable outlier of the Math-7B setting, which is explained below.

The outlier in setting Math-7B uses the base model Qwen2.5-Math-7B, which was also used in Yue et al. (2025) to conclude that distillation outperforms RL at all pass@k metrics, including pass@1. Further investigation reveals that this base model predominantly reasons about answers in Python, a pattern that is not predominant among the other tested base models. While distillation removes this behavior completely, since the teacher does not reason with Python, RL training with two different frameworks (Zeng et al., 2025; Liu et al., 2025a) fails to. This leads the RL-trained model to rely primarily on a suboptimal reasoning pattern: solving problems in Python and hallucinating the code's output. This allows the distilled model to outperform the RL-trained one in this setting. RL training on other base models, where Python reasoning is not predominant, does not suffer from the same problem. Details of the results demonstrating this are shown in Appendix A.7.

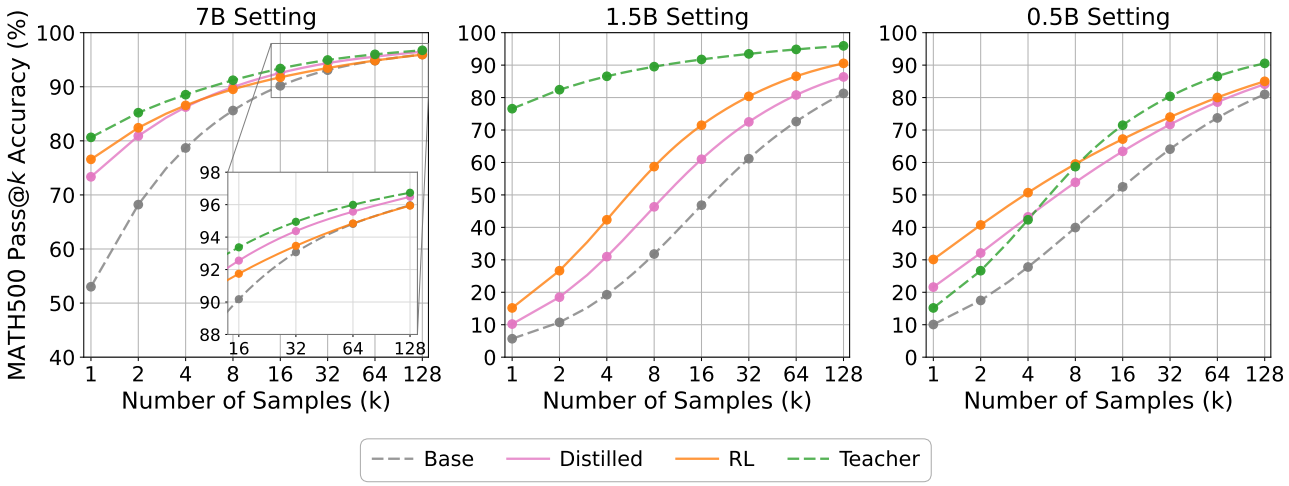

Figure 3. Pass@k accuracy (%) on MATH500 of the base, distilled, RL, and teacher models across the model sizes in Table 1 (omitting the Math-7B model outlier), done with a temperature of $T = 1$, sampling 256 answers to each question, and plotting k on a log scale. In all cases, RL improves the pass@1 accuracy of the base model more than distillation, but unlike RL, distillation consistently expands the models' reasoning boundaries.

These results also indicate that for RL to outperform distillation, the base model must already place non-trivial probability mass on desirable correct answers, since RL reinforces correct answers sampled from the base model. This is further discussed in the theoretical exploration offered in Section 5.1. Experiments investigating how this manifests in other base models are left to future work. This may be of particular interest in base models of other model families that have much worse initial math reasoning performance.

In addition to the results provided for models in the settings shown in Table Table 1, Appendix A.4 presents initial results for 14B and 32B models.

4.2. RL vs Distillation's Impact on a Model's Reasoning Boundary

Figure 3 shows the results for the pass@k evaluations of models across all model sizes. The results show that the RL model outperforms the distilled model for small k. They further demonstrate that distillation consistently expands the reasoning boundary of the base models, even in the 7B setting when RL fails to do so – a setting more representative of larger-scale models and distillation attacks. This shows that distillation is a more efficient and reliable way to expand a model's reasoning boundary.

RL does expand the reasoning boundary of base models in the 0.5B and 1.5B settings, consistent with work that clarifies there are certain conditions that allow this to occur (Zhang et al., 2025b), despite other claims to the contrary (Yue et al., 2025). However, this comes at the much higher cost of RL training, and is only observed in the smaller models.

Pass@k experiments for all models on the GSM8K dataset are shown and discussed in Appendix A.5, where the results indicate that pass@k comparisons need questions on which the base model's reasoning boundary has room to expand, so that distillation and RL can be useful.

5. Why does RL Outperform Distillation?

The following section offers a theoretical explanation of the performance differences between RL and distillation observed in our experiments. An alternate hypothesis, which proved wrong, is discussed in Appendix A.6.

5.1. Performance Differences are due to Mode-Seeking vs Mode-Covering Losses

RL outperforming distillation is surprising for several reasons. Sequence-level distillation (Kim & Rush, 2016) is a form of supervised learning, which is typically considered more efficient than RL. Concretely, from an information theory perspective, RL from binary rewards has been argued to provide only $O(1)$ bits per episode, whereas supervised learning gives tens to thousands of bits per sample (LeCun, 2016; Schulman & Lab, 2025). Moreover, some works have empirically shown cases where distillation outperforms RL (Guo et al., 2025a; Yue et al., 2025).[1] Thus, why does RL outperform distillation?

A different perspective that explains these results comes from probabilistic inference; we first give high-level intuition and then some more precise results. First, note that dis-

[1]We discuss why some demonstrated results from the literature, specifically instances where distillation outperforms RL, fail to hold more generally in Appendix A.7.

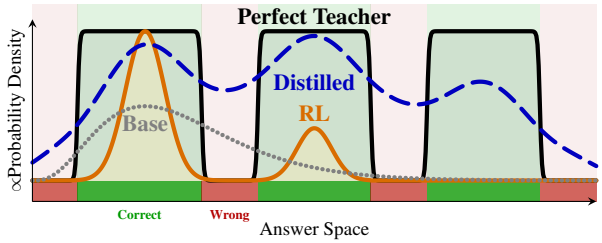

Figure 4. Relative to a **perfect teacher**, **RL** is mode-seeking while **distillation** (dashed) is mode-covering. Green regions represent correct answers, red represents incorrect. Probability densities are presented unnormalized. In practice, RL is more limited by the initial distribution of the **base** model (dotted) being trained than distillation.

tillation is also a form of RL, corresponding broadly to off-policy reinforcement learning, more specifically to imitation learning, which, in the case of sequence-level knowledge distillation, is simply behavior cloning (Rashidinejad et al., 2021; Foster et al., 2024). The loss used for knowledge distillation uses a forward KL and is therefore mode-covering (Gu et al., 2026), meaning it puts a high probability over the teacher's modes, potentially interpolating between them (see dashed blue line in Figure 4). This is a general property of imitation learning losses, specifically for a broad class of losses stemming from f-divergences (Ke et al., 2020). Thus, even given a perfect teacher, some of the student's probability mass could fall outside the teacher's support and therefore on incorrect answers. In contrast, typical max entropy on-policy RL is mode-seeking relative to a perfect teacher (Levine, 2018), which generally causes RL-trained student to collapse its probability mass to a subset of correct answers – see the orange line in Figure 4.

This intuition can be formalized, with proofs and extended discussions delegated to Appendix B. Assuming a temperature of $T = 1$, regular knowledge distillation minimizes the cross-entropy between a student and a teacher's distribution, which is equivalent to minimizing the mode-covering KL-divergence $D_{KL}(p_t|p_s)$, with p_s, p_t denoting the student and teacher's output distributions, respectively. However, sequence-level knowledge distillation is not clearly mode-covering, as it first generates a set of completions using the teacher and then trains the student on those completions. This is shown to be a noisy approximation of the regular mode-covering knowledge distillation loss.

Theorem 5.1. *(Seq-KD is noisy regular KD) Denote the teacher and student model distributions by p_t, p_s respectively. The regular knowledge distillation (KD) loss is $L_{KD} := D_{KL}(p_t|p_s) = \mathbb{E}_{p_t}[-\log(p_s)] - H(p_t)$, where $H(p_t)$ denotes the teacher distribution's entropy and we assume a temperature of $T = 1$. The sequence-level knowledge distillation (Seq-KD) loss is $\hat{L}_{Seq-KD} := \frac{1}{|D|}\sum_{x\in D} -\log(p_s(x))$, where $D \sim p_t$ is a set of completions sampled i.i.d from the teacher. The Seq-KD loss' gradient is an unbiased estimate of the KD gradient, such that*

$$\nabla L_{KD} = \mathbb{E}_{D\sim p_t}[\nabla \hat{L}_{Seq-KD}]$$

Thus, the sequence-KD loss, which is typically used when distilling different language models, has the same mode-covering behavior as regular knowledge distillation.

In contrast, RL is mode-seeking relative to a specific distribution. Specifically, assume a "perfect" teacher that places uniform probability mass over all correct answers. Policy-gradient methods like PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024b) modify the REINFORCE gradient to allow taking off-policy steps, with the regular on-policy REINFORCE gradient being $\mathbb{E}_p[r\nabla \log(p)]$, where p is the policy's distribution and r is the reward. Often an entropy term is added to induce exploration, yielding the loss $\mathbb{E}_p[r\nabla \log(p)] + \beta H(p)$, where β is a hyperparameter. When training models to solve math or coding problems by reasoning over them, r is typically a binary reward of 1 if the task is correctly solved and 0 otherwise. This reward is thus uniform over all correct answers. This intuition yields the following theorem.

Theorem 5.2. *(Max-entropy RL is mode-seeking to a perfect teacher) Let p_s denote a student model's distribution and p_t be the distribution of a soft perfect teacher, where $p_t(x) \propto \exp(\alpha r(x))$ for some α, where $r(x) = 1$ (for correct answers) and 0 otherwise. Note that when $\alpha \to \infty$, then p_t is uniform over correct answers and is a hard perfect teacher. Denote the max-entropy RL policy gradient as $\nabla L_{max\text{-}ent} = \mathbb{E}_{p_s}[r\nabla \log(p_s)] + \beta\nabla H(p_s)$. Also, denote the mode-seeking (reverse-KL) distillation loss as $L_{rev-KL} = D_{KL}(p_s|p_t)$. For appropriately chosen α or β we have that $\nabla L_{max\text{-}ent} \propto \nabla L_{rev-KL}$.*

Thus, as the RL loss is based on reverse-KL, it induces mode-seeking behavior.

Mode-seeking losses, as used in on-policy RL, enable mode collapsed policies to achieve low loss, where the probability mass is over some but not necessarily all modes (orange line in Figure 4). However, mode-covering losses, as used in distillation, require a wider support, which can result in significant probability mass between modes being given to incorrect answers (dashed blue line in Figure 4), lowering pass@1 performance of models trained using distillation when compared to those trained using RL. Gu et al. (2026) discusses similar intuition as well, showing that accordingly, mode-seeking on-policy distillation outperforms mode-covering off-policy distillation.

The different behaviors of these losses also offer a speculative explanation for why distillation consistently expands a model's reasoning boundary, whereas RL often does not (Yue et al., 2025). Given a subset of problems for which the base model has very little or no probability mass over correct answers, RL training may not increase the probability

of generating correct answers, since its mode-seeking loss does not penalize giving no probability mass to low probability solutions. In contrast, distillation would more readily increase the low probability mass over correct answers to these problems, since its mode-covering loss incentivizes the student to cover all the modes of its teacher. Thus, when sampled from many times, and when compared to RL-trained models, distilled models are more often able to solve problems that they could not previously.

It is through this mechanism – increased probability mass on questions that the base model struggled with – that distilled models' pass@k performance can increase relative to RL-trained models, despite a lower pass@1. The intuition for this is as follows. For any given question in a dataset, i, if the model has a probability p_i of answering it correctly, then its pass@1 for a dataset of just that question is also p_i, and its pass@k is $1-(1-p_i)^k$. In a dataset of many questions, total pass@1 and pass@k are the average of the pass@1 and pass@k of each question. Questions a model struggles with have a low p_i, in which case pass@k per question can be approximated as $p_i \cdot k$. Thus, higher values of k in pass@k evaluations over the whole dataset amplify the probability mass on questions that the base model struggles with, unlike pass@1, whose value is dominated by questions where $p_i \approx 1$. In practice, pass@k better indicates than pass@1 how probability mass is distributed across the correct answers to questions in a dataset of varying difficulty for the base model.

These results also indicate that, for the largest reasoning performance gains through RL in practice, base models must initially have sufficient probability mass over all desirable regions of the answer space. RL with a poor base model (see gray dotted line in Figure 4) would struggle to collapse to correct answers if most of its mass lies in incorrect or narrow regions. However, distillation does not suffer from the same limitation. This underscores the value of proprietary LLM reasoning traces in pipelines that seek to steal reasoning, and aligns with observations in previous work that to maximally improve reasoning, a pipeline should combine distillation and RL (Guo et al., 2025a; Yang et al., 2025; Narayanan et al., 2025). Distillation can be considered a cheap and efficient way to expand a model's reasoning boundary, allowing sufficient probability mass to exist in desirable areas for the base model, which subsequent RL training can then reinforce to increase the model's pass@1 accuracy.

6. Practical Implications for Stealing Reasoning

The results in Sections 4 and 5 imply that RL training is limited by a base model's reasoning boundary more so than its pass@1 performance. However, most existing defenses against stealing reasoning are evaluated without using pass@k metrics, failing to consider the impact of distillation on a model's reasoning boundary (Savani et al., 2025; Xu et al., 2026; Li et al., 2025). The danger that then arises is that these defenses can reduce a model's pass@1 performance gains without significantly affecting its pass@k gains, thereby failing to prevent a student model's reasoning boundary from expanding and allowing the student to have a higher post-RL performance ceiling. Crucially, subsequent RL training on this student would likely yield a more capable model than without distillation, rendering these defenses ineffective.

6.1. Pass@k Evaluations Better Indicate a Defense's True Efficacy Than Pass@1

A simple experiment can demonstrate how a defense that seems effective with pass@1 evaluations before RL training can be evidently ineffective after RL, and illustrate how, in contrast to pass@1, pass@k better indicates the defense's true efficacy. This experiment is performed on an existing defense, namely, antidistillation sampling (Savani et al., 2025). Antidistillation modifies the output tokens in a closed-source model's reasoning trace so that they remain likely in the model's output distribution, but would reduce the performance of any distilled student model on a selected downstream task. To achieve this, the defense selects a single and universal student 'proxy model' to estimate which of the likely tokens in the teacher's vocabulary would increase any student's loss on the downstream task. The defense provides a parameter to control the amount of poisoning applied to the teacher model's outputs – higher poisoning levels are more effective defenses, but come at the cost of reduced teacher performance, which we report as a relative drop in the accuracy of the teacher's generated traces.

To be an applicable and effective defense, antidistillation sampling should disrupt performance gains from distillation for all black-box students (students whose model architecture is unknown, and which comes from a different family to the proxy model) without incurring a high cost to teacher performance. Recognizing this, Savani et al. (2025) propose the use of Qwen2.5-3B as a proxy model, and demonstrate its effectiveness at disrupting a Llama-3.2-3B (Grattafiori et al., 2024) student model's gains of pass@1 math reasoning capabilities from distillation.

Figure 5 (left) shows pass@k evaluations of the same setup as used by (Savani et al., 2025), with Llama-3.2-3B representative of black-box students, where distillation with different poisoning levels is performed on the GSM8k dataset. Pass@k evaluations are done with the same evaluation framework as for 0.5B and 1.5B model architectures (see Section 3). The results indicate that although the defense reduces pass@1 performance, it is largely ineffective at preventing the student's reasoning boundary from expanding

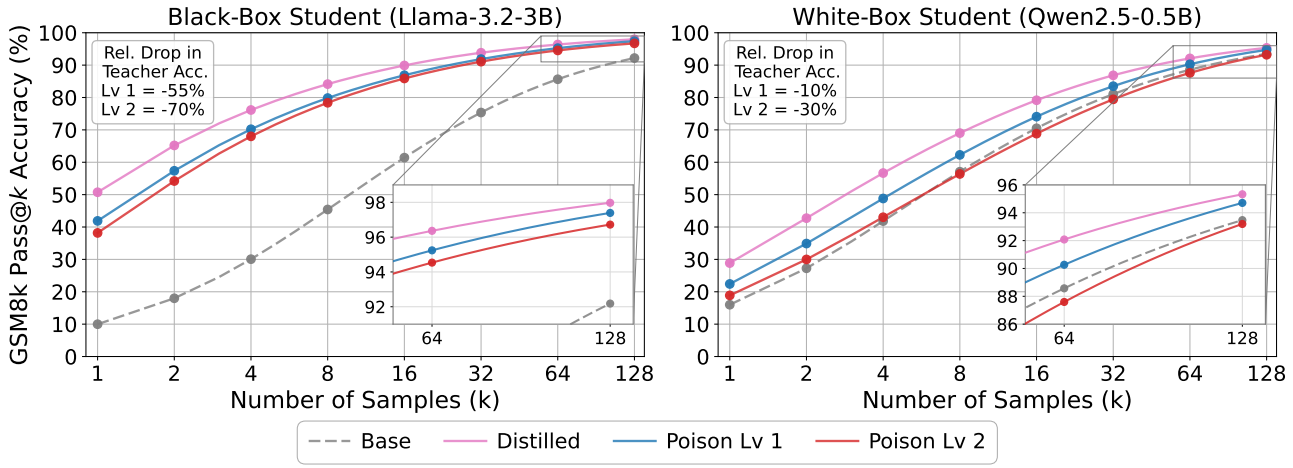

Figure 5. Pass@k evaluations of antidistillation sampling show that the defense is largely ineffective at preventing a student model's reasoning boundary from expanding. Results are shown for a realistic black-box student (left), where the student model architecture is unknown and doesn't match that of the proxy model, and for an unrealistic white-box student (right), where the proxy model in the defense matches the architecture of the distilled model. For the black-box student, relatively degrading the teacher's performance by 55 and 70% still allows the base model to achieve significant pass@k gains, bringing it within 1.5% of the final pass@128 accuracy of the unpoisoned distilled model. For the unrealistic white-box student, poisoning levels relatively degrade the teacher's accuracy by 10% and 30%, and while the latter prevents the base model's reasoning boundary from expanding, it comes at a high cost.

through distillation; even when the teacher's traces are poisoned to reduce their accuracy by 70%, the student is within 1.5% of the pass@128 value of the distilled model with no poisoning, gaining 4.5% compared to the base model.

This defense's efficacy in a white box setting can also be evaluated, as shown in Figure 5 (right). In this case, the Qwen2.5-0.5B student matches the architecture of the Qwen2.5-3B proxy model. These experiments exactly replicate those for the 0.5B setting in Table 1, except that they use antidistillation sampling to poison teacher traces. Results show that even in this less applicable setting where student and proxy model architectures match, preventing the base model's reasoning boundary from expanding still requires a large 30% reduction in teacher accuracy.

To investigate whether an expanded reasoning boundary after distillation leads to post-RL performance gains, RL training is done with the SimpleRLZoo framework (Zeng et al., 2025) on the distilled Qwen2.5-0.5B models, with and without poisoning level 1. Pass@1 performance is compared on the same math datasets as used in Figure 2. Even in this white-box setting, where antidistillation is more effective as a defense than in practice, the poisoned model achieves higher average post-RL performance after 130 steps: 40.8%, compared to 40.6% for the distilled model, with both outperforming RL training of the base model without distillation (40.3%). Most notably, both models achieve a 2.5% improvement on the hardest math dataset, Minerva, compared to the base model using RL alone, which aligns with theoretical observations: in post-RL evaluations, distillation facilitates greater improvements to the base model on questions it struggled with most before. Thus, despite reducing pass@1 gains from distillation before RL training, these gains are recovered after RL training, rendering antidistillation ineffective.

More thorough experiments that perform RL on a wider range of models and defenses are planned for our immediate future work. Additional details of the hyperparameters used in these experiments are available in Appendix A.8.

6.2. Pass@1 Metrics Fail to Indicate the Performance Ceiling of Post-RL Training

While implied directly by theoretical observations in Section 5.1, and demonstrated practically in Section 6.1, a simple illustrative experiment can further illustrate how pass@1 evaluations fail to capture the potential gains a base model can achieve through RL training. This failure occurs because pass@1 does not indicate whether the base model has sufficient probability mass in desirable portions of the answer space.

Consider a multi-armed bandit (Sutton et al., 1998), trained with the REINFORCE policy gradient (Williams, 1992). A multi-armed or n-armed bandit is a setting in which a model (the 'bandit') randomly selects from n possible actions (its 'arms') with probability p_i for each action i. During RL training, the bandit receives a binary reward from each action with probability q_i, allowing it to learn to take the best actions, i.e., those that lead to the highest average reward. The pass@1 accuracy of a bandit can be calculated as the probability it selects an action that leads to a positive reward, $\sum_i p_i q_i$. The multi-armed bandit can be considered as very

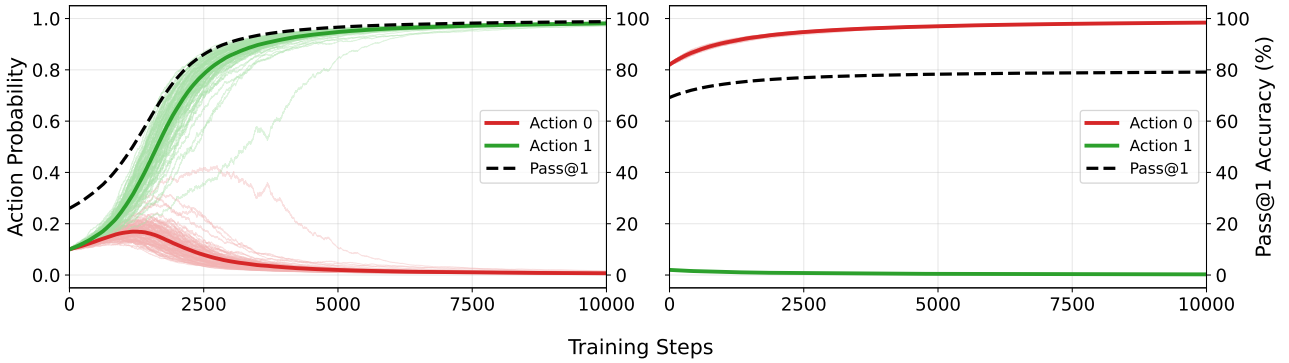

Figure 6. A bandit with initially low pass@1 accuracy (left) reaches a much higher pass@1 accuracy after RL training compared to a bandit with initially high pass@1 accuracy (right), illustrating the danger of using low pass@1 as a measure of the ineffectiveness of subsequent RL training. Training runs are averaged over 100 trials performed in a 10-armed bandit setting with noisy binary rewards, using the REINFORCE algorithm. In this setting, Action 0 gives a positive binary reward 80% of the time, Action 1 100% of the time, and Actions 2–9 10% of the time.

roughly analogous to a language model selecting from a vast set of possible reasoning paths to answer a question, each with a certain probability of leading to the correct solution. The bandit's pass@1 accuracy thus represents a model's pass@1 accuracy on that question. In this setting, it can be demonstrated that a bandit that initially has a low pass@1 score but sufficient probability mass on the best actions has a higher performance ceiling after RL than a bandit that begins with a much higher pass@1 score but lacks sufficient probability mass on the best actions.

One such case is demonstrated in Figure 6, using two 10-armed bandits. Here, the probabilities of each action i leading to a positive binary reward are always $q_0 = 80\%$, $q_1 = 100\%$, and $q_i = 10\%$ for all other i, where the best action (with the highest average reward) is clearly Action 1. The first bandit achieves a high initial pass@1 of 69% by beginning with $p_0 = 82\%$, and $p_i = 2\%$ for all other i, initially favoring only Action 0. The second has a low initial pass@1 of 26%, beginning with $p_i = 10\%$ for all i, but, in contrast to the first, there is no prior bias against selecting Action 1. After 10,000 steps of training with REINFORCE and using a learning rate of 0.01, the initially low pass@1 bandit achieves a much higher pass@1 performance than the bandit that began with a higher pass@1 accuracy.

To best match the constraints of training LLMs with RL, this setting assumes a limited compute budget to illustrate its point, without increasing the learning rate or the number of training steps. One trivial example without this limitation would be a model that begins with no probability mass on Action 1 and therefore could never achieve a higher pass@1 than 80% after RL training, even with unlimited compute. While this experiment is a simplification, it is a useful illustration of the mechanism through which pass@1 can be misleading.

7. Conclusion

This work demonstrates the importance of using pass@k metrics to evaluate defenses against stealing reasoning. Results begin by clarifying conflicting conclusions in recent work comparing the performance of RL and distillation as standalone training methods for improving reasoning. They demonstrate that LLMs of varying sizes can be trained with RL to outperform those trained with distillation in pass@1 math reasoning evaluations. At the same time, models trained with distillation are shown to consistently expand a model's reasoning boundary. This evidence, along with the mode-seeking and mode-covering nature of RL and distillation, respectively, strongly suggests that the value of reasoning traces from frontier models in pipelines that steal reasoning lies in their ability to efficiently expand a model's reasoning boundary through distillation, thereby increasing a model's performance ceiling post-RL training. This clearer understanding implies that defenses against distillation attacks that modify reasoning traces must disrupt not only distillation but also any subsequent RL training. More precisely, they demonstrate that a defense's efficacy against stealing reasoning with distillation must be evaluated using pass@k metrics on the distilled models, which better indicate the models' post-RL-training performance ceiling than pass@1. However, existing defenses mainly rely on pass@1 evaluations (Savani et al., 2025; Xu et al., 2026; Li et al., 2025). This is demonstrated to be misleading for one such defense, antidistillation (Savani et al., 2025), which fails to impede students' pass@k performance gains, despite limiting pass@1 gains. An initial experiment further demonstrates that RL training can recover the performance gains achieved through distillation, despite this defense. Further experiments that perform RL training on a wider range of student models distilled with various defenses are left to future work, along with efforts to develop more effective defenses against pipelines that steal reasoning.

Impact Statement

This work aims to help defenders understand how reasoning may be stolen in a realistic setting, thereby better evaluating and developing defenses. After consulting various AI safety experts, we believe the information presented on the nature of pipelines to steal reasoning is known to attackers, and does not pose any substantial threat that could be used for negative ends. Instead, insights are offered to defenders to ensure that evaluations for defenses against stealing reasoning consider the full scope of these pipelines.

Acknowledgements

We would like to thank Nick Rhinehart for discussing mode-seeking vs. mode-covering behaviors of RL and distillation, and the relationship between behavioral cloning and on-policy RL. We are also grateful for the funding we have received, enabling us to present this work. Shidan received funding from University College, Oxford, and the TAIGR workshop. Oliver received funding from Christ Church, Oxford. Yonatan is funded by the Rhodes Trust and the AIMS EPSRC CDT (grant no. EP/S024050/1).

References

Anthropic. Detecting and preventing distillation attacks. Anthropic News, 2026. URL https://www.anthropic.com/news/detecting-and-preventing-distillation-attacks. Accessed: 21 April 2026.

Bercovich, A., Levy, I., Golan, I., Dabbah, M., El-Yaniv, R., Puny, O., Galil, I., Moshe, Z., Ronen, T., Nabwani, N., et al. Llama-nemotron: Efficient reasoning models. *arXiv preprint arXiv:2505.00949*, 2025.

Busbridge, D., Shidani, A., Weers, F., Ramapuram, J., Littwin, E., and Webb, R. Distillation scaling laws, 2025. URL https://arxiv.org/abs/2502.08606.

Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. D. O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Chu, T., Zhai, Y., Yang, J., Tong, S., Xie, S., Schuurmans, D., Le, Q. V., Levine, S., and Ma, Y. Sft memorizes, rl generalizes: A comparative study of foundation model post-training, 2025. URL https://arxiv.org/abs/2501.17161.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Foster, D. J., Block, A., and Misra, D. Is behavior cloning all you need? understanding horizon in imitation learning. *Advances in Neural Information Processing Systems*, 37: 120602–120666, 2024.

Gao, L., Tow, J., Abbasi, B., Biderman, S., Black, S., DiPofi, A., Foster, C., Golding, L., Hsu, J., Le Noac'h, A., Li, H., McDonell, K., Muennighoff, N., Ociepa, C., Phang, J., Reynolds, L., Schoelkopf, H., Skowron, A., Sutawika, L., Tang, E., Thite, A., Wang, B., Wang, K., and Zou, A. The language model evaluation harness, 07 2024. URL https://zenodo.org/records/12608602.

Gibney, E. Secrets of deepseek ai model revealed in landmark paper. *Nature*, 2025.

Google. Distillation, experimentation, and (continued) integration of ai for adversarial use. Google Threat Intelligence Group Cloud Blog, 2026. URL https://cloud.google.com/blog/topics/threat-intelligence/distillation-experimentation-integration-ai-adversarial-use. Accessed: 21 April 2026.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Gu, Y., Dong, L., Wei, F., and Huang, M. Minillm: On-policy distillation of large language models, 2026. URL https://arxiv.org/abs/2306.08543.

Guo, D., Yang, D., Zhang, H., Song, J., Wang, P., Zhu, Q., Xu, R., Zhang, R., Ma, S., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025a.

Guo, D., Yang, D., Zhang, H., Song, J., Wang, P., Zhu, Q., Xu, R., Zhang, R., Ma, S., Bi, X., et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025b.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *Advances in neural information processing systems*, 2021.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Hu, X., Lu, X., Mao, L., Zhang, Y., Zhang, T., Wen, B., Yang, F., Gao, T., and Zhou, G. Why distillation can outperform zero-rl: The role of flexible reasoning. *arXiv preprint arXiv:2505.21067*, 2025.

Huan, M., Li, Y., Zheng, T., Xu, X., Kim, S., Du, M., Poovendran, R., Neubig, G., and Yue, X. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning, 2025. URL https://arxiv.org/abs/2507.00432.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Ke, L., Choudhury, S., Barnes, M., Sun, W., Lee, G., and Srinivasa, S. Imitation learning as f-divergence minimization. In *International workshop on the algorithmic foundations of robotics*, pp. 313–329. Springer, 2020.

Kim, Y. and Rush, A. M. Sequence-level knowledge distillation, 2016. URL https://arxiv.org/abs/1606.07947.

Kydlicek, H., Lozovskaya, A., Habib, N., and Fourrier, C. Fixing open llm leaderboard with math-verify, 2025.

LeCun, Y. Predictive learning. Keynote talk at the 30th Annual Conference on Neural Information Processing Systems (NIPS), December 2016.

Levine, S. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Li, P., Tan, Z., Zhang, M., Qu, H., Liu, H., and Chen, T. Doge: Defensive output generation for llm protection against knowledge distillation, 2025. URL https://arxiv.org/abs/2505.19504.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Liu, Q., Li, L., Tang, Z., and Zhou, D. Breaking the curse of horizon: Infinite-horizon off-policy estimation. *Advances in neural information processing systems*, 31, 2018.

Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W. S., and Lin, M. Understanding r1-zero-like training: A critical perspective, 2025a. URL https://arxiv.org/abs/2503.20783.

Liu, Z., Yang, Z., Chen, Y., Lee, C., Shoeybi, M., Catanzaro, B., and Ping, W. Acereason-nemotron 1.1: Advancing math and code reasoning through sft and rl synergy, 2025b. URL https://arxiv.org/abs/2506.13284.

Luo, M., Tan, S., Huang, R., Patel, A., Ariyak, A., Wu, Q., Shi, X., Xin, R., Cai, C., Weber, M., Zhang, C., Li, L. E., Popa, R. A., and Stoica, I. Deepcoder: A fully open-source 14b coder at o3-mini level. https://pretty-radio-b75.notion.site/DeepCoder-A-Fully-Open-Source-14B-Coder-at-O3-mini-Level-1cf81902c14680b3bee5eb349a512a51, 2025a. Notion Blog.

Luo, M., Tan, S., Wong, J., Shi, X., Tang, W. Y., Roongta, M., Cai, C., Luo, J., Li, L. E., Popa, R. A., and Stoica, I. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2, 2025b. Notion Blog.

Metz, C. Openai says deepseek may have improperly harvested its data, January 2025. URL https://www.nytimes.com/2025/01/29/technology/openai-deepseek-data-harvest.html.

Muennighoff, N., Yang, Z., Shi, W., Li, X. L., Fei-Fei, L., Hajishirzi, H., Zettlemoyer, L., Liang, P., Candès, E., and Hashimoto, T. s1: Simple test-time scaling, 2025. URL https://arxiv.org/abs/2501.19393.

Narayanan, S. M., Braza, J. D., Griffiths, R.-R., Bou, A., Wellawatte, G., Ramos, M. C., Mitchener, L., Rodriques, S. G., and White, A. D. Training a scientific reasoning model for chemistry, 2025. URL https://arxiv.org/abs/2506.17238.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Qwen, :, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., Bao, K., Yang, K., Yu, L., Li, M., Xue, M., Zhang, P., Zhu, Q., Men, R., Lin, R., Li, T., Tang, T., Xia, T., Ren, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Wan, Y., Liu, Y., Cui, Z., Zhang, Z., and Qiu, Z. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Rashidinejad, P., Zhu, B., Ma, C., Jiao, J., and Russell, S. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 34:11702–11716, 2021.

Savani, Y., Trockman, A., Feng, Z., Xu, Y. E., Schwarzschild, A., Robey, A., Finzi, M., and Kolter,

J. Z. Antidistillation sampling, 2025. URL https://arxiv.org/abs/2504.13146.

Schulman, J. and Lab, T. M. Lora without regret. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250929. https://thinkingmachines.ai/blog/lora/.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Seetharaman, D. and Arámburo, F. Openai accuses deepseek of distilling u.s. models to gain advantage, bloomberg news reports. *Reuters*, February 2026. URL https://www.reuters.com/world/china/openai-accuses-deepseek-distilling-us-models-gain-advantage-bloomberg-news-2026-02-12/.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024a.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024b.

Sutton, R. S., Barto, A. G., et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Sweney, M. and Milmo, D. Openai reviewing allegations that its ai models were used to make deepseek, January 2025. URL https://www.theguardian.com/technology/2025/jan/29/openai-chatgpt-deepseek-china-us-ai-models.

Trockman, A. and Savani, Y. Antidistillation preserves ai openness, originality, and safety. Antidistillation Blog, 2026. URL https://antidistillation.com/blog/unexpected-externalities-of-distillation/. Accessed: 21 April 2026.

Veeraboina, H. Aime problem set 1983-2024, 2023. URL https://www.kaggle.com/datasets/hemishveeraboina/aime-problem-set-1983-2024.

von Werra, L., Belkada, Y., Tunstall, L., Beeching, E., Thrush, T., Lambert, N., Huang, S., Rasul, K., and Gallouédec, Q. TRL: Transformers Reinforcement Learning. URL https://github.com/huggingface/trl.

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

Xu, Y. E., Kirchenbauer, J., Savani, Y., Trockman, A., Robey, A., Goldstein, T., Fang, F., and Kolter, J. Z. Antidistillation fingerprinting, 2026. URL https://arxiv.org/abs/2602.03812.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yue, Y., Chen, Z., Lu, R., Zhao, A., Wang, Z., Yue, Y., Song, S., and Huang, G. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?, 2025. URL https://arxiv.org/abs/2504.13837.

Zeng, W., Huang, Y., Liu, Q., Liu, W., He, K., Ma, Z., and He, J. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild, 2025. URL https://arxiv.org/abs/2503.18892.

Zhang, C., Li, Q., Song, D., Ye, Z., Gao, Y., and Hu, Y. Towards the law of capacity gap in distilling language models, 2025a. URL https://arxiv.org/abs/2311.07052.

Zhang, C., Neubig, G., and Yue, X. On the interplay of pre-training, mid-training, and rl on reasoning language models, 2025b. URL https://arxiv.org/abs/2512.07783.

A. Additional Results and Experiment Hyperparameters

A.1. Distillation Hyperparameters

Distillation hyperparameters were selected following the setup used in Simple Test Time Scaling (Muennighoff et al., 2025), which uses the Transformers Reinforcement Learning library (von Werra et al.). A learning rate of 10^{-5} was used alongside a weight decay of 10^{-4}, as well as a cosine learning rate scheduler with a warmup-ratio of 0.1. Gradients were accumulated in steps of 16, with a batch size of 1 per device. Models of size less than 7B were distilled on 1 A100 GPU, while those of size 7B were distilled in parallel on 4 A100 GPUs. All distillation was performed over 1 training epoch.

A.2. Distillation Ablations

Further ablations were performed to investigate the effects of various design choices during distillation. These include the ablations listed below, alongside their rough effects on performance measured on GMS8K.

- Distillation performed on teacher traces generated with a temperature of $T = 1$ marginally outperformed distillation on traces generated with $T = 0$ ($+1\%$), and thus a temperature of $T = 1$ is used to generate all traces for distillation.
- Distillation was attempted over datasets created by sampling from the teacher either 1, 3, 5 or 8 times per question, as well as with training for 1, 2, 3, 5, or 8 epochs. Increasing either of these marginally harmed the performance of student models (-1%), and thus 1 sample of each question is used for distillation, and training done for 1 epoch.
- Distillation was attempted with rejection sampling, keeping only correct answers from teachers given either 8, 32 or 64 attempts to answer each question in the distillation dataset. This marginally improved performance ($+1\%$) when 8 samples were used, but not significantly enough to surpass the performance of models trained with RL. For simplicity, results are presented without rejection sampling.
- Varying teacher size marginally improved results by $+1 - 2\%$ when smaller teachers were used, aligning with literature discussing the impact of a capacity gap on a student's ability to learn from a teacher (Zhang et al., 2025a; Busbridge et al., 2025), but never exceeding RL in performance. Understanding this capacity gap is not the focus of this work, and thus, teachers of consistently larger sizes are used in distillation, which are also more representative of the use cases in distillation attacks for stealing reasoning.
- Using different datasets of questions to generate the teachers' reasoning traces for distillation was attempted, rather than constraining the questions to match those used in RL. This had mixed effects. Model performance improved slightly ($+2\%$) when questions of a marginally harder difficulty were used for distillation on the 0.5B class of models, using the hard rather than medium dataset of questions from SimpleRL-Zoo (Zeng et al., 2025). However, using the questions in AIME (Veeraboina, 2023), GSM8K (Cobbe et al., 2021) or the dataset in Simple Test Time Scaling (Muennighoff et al., 2025) to generate teacher traces decreased performance by up to 5%. Results are reported using the same questions as in RL training for their higher accuracy and a fairer comparison.

A.3. Evaluation Framework

The LM Evaluation Harness (Gao et al., 2024) was used to implement evaluations on GSM8K, Minerva, and MATH500 shown in Figure 2, and all models were evaluated with $T = 0$, given that it led to the best performance. Model answers were extracted and compared to the correct solutions using the harness's built-in implementation of math verify (Kydlicek et al., 2025), assessing correctness of reasoning rather than confounding it with instruction following and formatting abilities as well.

Pass@k experiments were done with a temperature of $T = 1$ on the MATH500 and GSM8K datasets, using the Python implementation of math-verify (Kydlicek et al., 2025) to check the correctness of model responses. Pass@k calculations were done by sampling 256 answers to each question for each model, ensuring that results at the maximum value of $k = 128$ are statistically significant Chen et al. (2021).

A.4. Results at Larger Scales

Table 2 compares the performance of base models of sizes 14B and 32B trained with RL and distillation. Results show that even at larger scales, models trained with RL typically outperform those trained with distillation. The performance of

Table 2. Comparison of base models trained with RL and distillation at the 14B and 32B scales. Open-source versions of each model were used and evaluated using the same evaluation framework as the 7B scale. All accuracies are at pass@1 with a temperature of $T = 0$.

MODEL	GSM8K	MATH500	MINERVA	AVERAGE
QWEN2.5-14B-SIMPLERLZOO	**92.72%**	**94.39%**	82.34%	89.82%
QWEN2.5-14B-DISTILL-DEEPSEEKR1	90.67%	89.40%	**90.72%**	**90.26%**
DAPO-QWEN2.5-32B	**94.39%**	**91.60%**	**94.74%**	**93.58%**
QWEN2.5-32B-DISTILL-DEEPSEEKR1	93.10%	90.20%	90.96%	91.42%

Qwen2.5-14B-SimpleRLZoo on Minerva is an outlier that remains to be investigated, as for all other models, performance on Math500 and Minerva are very similar.

A.5. Additional Results for Pass@k Experiments

Figure 7 shows the results of pass@k experiments for models evaluated on GSM8K. While the main result that distillation consistently expands the reasoning boundary as much as RL is observed in this setting, there are also some key differences. For the 7B model, the results show that the base model performs better than the distilled, RL and teacher model. We attribute this to the fact that this particular 7B base model is better than the teacher model for high pass@k. We leave an exploration of the relative difficulty of the benchmark when compared to the performance of the base and teacher models to future work. This phenomenon is not observed in the 1.5B and 0.5B settings, where the base model's reasoning boundaries have room to expand.

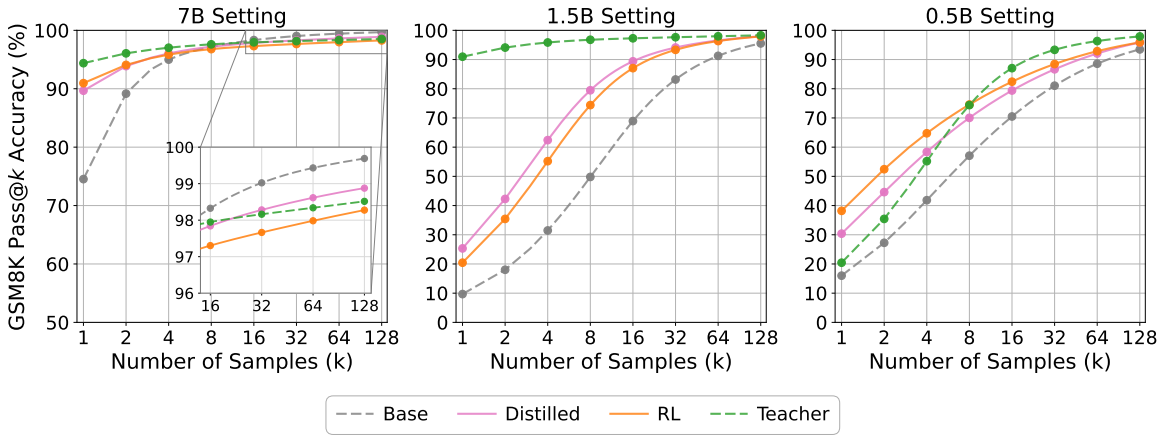

Figure 7. Pass@k accuracy (%) on GSM8K of the base, distilled, RL, and teacher models across the model sizes in Table 1 (omitting the Math-7B model outlier), done with a temperature of $T = 1$, sampling 256 answers to each question, and plotting k on a log scale. We see results comparable to those in Figure 3 for the 0.5B and 1.5B models, but for the 7B model, we find that the base model outperforms the distilled, RL and teacher models. We conjecture that this is because GSM8K is not sufficiently difficult for this particular 7B model.

A.6. Response Length And Question Difficulty Correlation

Another hypothesis for why RL outperforms distillation is that distilled models, having learned from an out-of-distribution teacher, struggle to self-correct errors in longer reasoning traces. This an instance of the curse of horizon in imitation learning (Liu et al., 2018), where errors compound over longer trajectories when learning from a different distribution. To test this, we examine whether accuracy degrades with response length more steeply for distilled models than RL-trained ones on the Qwen2.5-7B base model, controlled for question difficulty.

Specifically, we use the Math500 dataset to test whether a correlation exists among the response lengths, question difficulty, and the accuracy of the Qwen2.5-7B base models trained with RL and distillation. 32 questions are randomly selected of each difficulty level in MATH500, and each model is given 64 attempts to answer each of these questions. The responses of each model are then grouped by difficulty level, yielding 1024 responses per group. Subsequently, the average accuracy for each difficulty level is calculated across responses at length thresholds ranging from 500 to 2000 in increments of 50, and

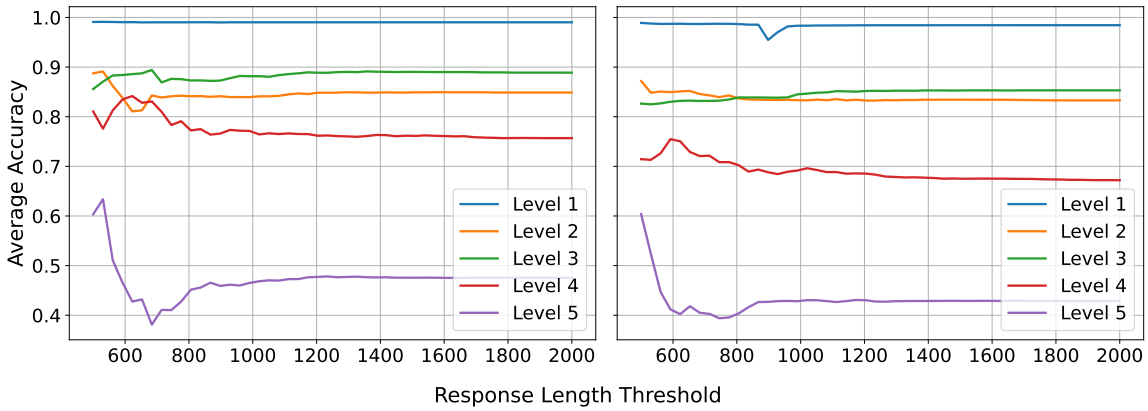

Figure 8. Comparison of accuracy on the MATH500 dataset for varying response length thresholds for Qwen2.5-7B-SimpleRLZoo (left) and for Qwen2.5-7B-Distilled (right).

compared.

Results of these experiments are shown in Figure 8, and demonstrate no meaningful correlations. The model trained with distillation gets fractionally fewer answers correct than the model trained with RL, for questions of all difficulty levels. Initial qualitative observations indicate that hallucinations leading to errors are more frequent in distilled models than in models trained with RL, but this remains to be investigated in future work.

A.7. Explaining the Qwen2.5-Math-7B Outlier

Figure 9 presents the results of experiments conducted to understand the outlier behavior of the distilled and RL models in the Math-7B setting reported in Figure 2. The accuracy of answers with and without Python code is compared for models in this setting, alongside the base and RL-trained models from the 7B setting (see Table 1 for details). Results demonstrate that RL performed through two different frameworks (Liu et al., 2025a; Zeng et al., 2025) undesirably reinforces the dominant reasoning paradigm of solving problems with Python code in the Qwen2.5-Math-7B base model. Examples of its resoning traces that use python to solve problems are provided in Figure 10.

A.8. Hyperparameters for RL Training with SimpleRLZoo

RL training was performed with the SimpleRLZoo framework (Zeng et al., 2025) on medium-difficulty questions. Interestingly, these were the same questions used to generate traces for distillation, and still led to notable performance gains of trained models. RL training was done on 4 A100 GPUs with an effective batch size of 1024. Hyperparameters used were the same as those by (Zeng et al., 2025), with the only adjustments being to account for the use of only 4 GPUs, and reducing the validation batch size from 500 to 256.

B. Proofs of Theorems

Theorems are restated here for convenience. Theorem 5.2 is taken from Levine (2018) with some modifications and simplifications for our setting.

Note that Kim & Rush (2016) proposes using sequence-KD with beam search and potentially other sampling modifications, which bias the effective teacher's distribution. Here we assume no such modified sampling is used, as in practice, beam search is rarely used with modern LLMs, and while rejection sampling or greedy decoding can be employed, in practice, they do not change the qualitative results – see Appendix A.2.

Theorem. *(Seq-KD is noisy regular KD) Denote the teacher and student model distributions by p_t, p_s respectively. The regular knowledge distillation (KD) loss is $L_{KD} := D_{KL}(p_t|p_s) = \mathbb{E}_{p_t}[-\log(p_s)] - H(p_t)$, where $H(p_t)$ denotes the teacher distribution's entropy and we assume a temperature of $T = 1$. The sequence-level knowledge distillation (Seq-KD) loss is $\hat{L}_{Seq-KD} := \frac{1}{|D|}\sum_{x\in D} -\log(p_s(x))$, where $D \sim p_t$ is a set of completions sampled i.i.d from the teacher. The*

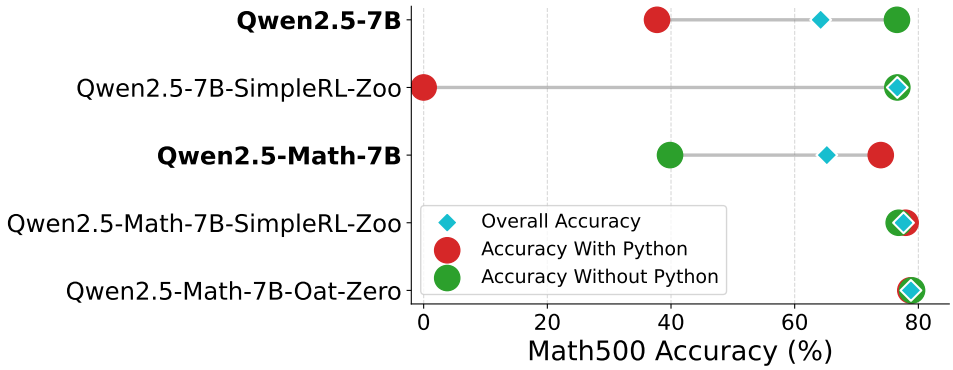

Figure 9. Accuracy with and without Python code for two Qwen2.5-7B base models and their RL-trained counterparts on Math500. RL performed through two frameworks (Liu et al., 2025a; Zeng et al., 2025) undesirably reinforces the dominant reasoning paradigm of solving problems with Python in Qwen2.5-Math-7B. In Qwen2.5-7B, where this behavior is not dominant, RL training eliminates it.

To determine the number of ways to arrange the letters of the word "ELLIPSE," we need to account for the fact that the word contains repeated letters. The word "ELLIPSE" has 7 letters in total, with the letter "E" appearing twice and all other letters appearing once.
The formula for the number of distinct permutations of a word with repeated letters is given by:
\[
\frac{n!}{n_1! \cdot n_2! \cdot \ldots \cdot n_k!}
\]
where \(n\) is the total number of letters, and \(n_1, n_2, \ldots, n_k\) are the frequencies of the repeating letters. For the word "ELLIPSE," we have:
\[
\frac{7!}{2!}
\]
Let's calculate this using Python to ensure accuracy.
```python
import math
# Total number of letters
n = 7
# Frequency of the letter 'E'
freq_E = 2
# Calculate the number of distinct permutations
num_permutations = math.factorial(n) //
math.factorial(freq_E)
print(num_permutations)
```
```output
2520
```
The number of ways to arrange the letters of the word "ELLIPSE" is \(\boxed{2520}\).

To find the positive square root of the product \(10 \times 15 \times 24\), we can follow these steps:

1. Calculate the product of the numbers.
2. Find the square root of the product.

Let's start by calculating the product \(10 \times 15 \times 24\):
\[
10 \times 15 \times 24 = 3600
\]
Next, we need to find the square root of 3600. We can use Python to ensure the accuracy of this calculation.
```python
import math
# Calculate the product
product = 10 * 15 * 24
# Calculate the square root of the product
square_root = math.sqrt(product)
print(square_root)
```
```output
60.0
```
The positive square root of the product \(10 \times 15 \times 24\) is \(\boxed{60}\).

Figure 10. Sample reasoning traces from Qwen2.5-Math-7B that produce Python code and hallucinate the corresponding outputs. Both a wrong (left) and a right (right) answer are provided to demonstrate that this is the dominant reasoning paradigm the model uses to answer questions.

Seq-KD loss' gradient is an unbiased estimate of the KD gradient, such that

$$\nabla L_{KD} = \mathbb{E}_{D\sim p_t}[\nabla \hat{L}_{Seq-KD}].$$

Proof. Note that:

$$\mathbb{E}_{D\sim p_t}[\hat{L}_{\text{Seq}-KD}] = \frac{1}{|D|}\sum_{i=1}^{|D|}\mathbb{E}_{x\sim p_t}[-\log(p_s(x))] = \frac{1}{|D|}|D|\mathbb{E}_{x\sim p_t}[-\log(p_s(x))] = \mathbb{E}_{p_t}[-\log(p_s)].$$

Thus, as $\nabla H(p_t) = 0$ as the teacher is constant with respect to the student distribution's parameterization, we have that: $\nabla L_{KD} = \nabla \mathbb{E}_{p_t}[-\log(p_s)] = \nabla \mathbb{E}_{D\sim p_t}[\hat{L}_{\text{Seq}-KD}]$, thereby proving the theorem. □

Theorem. *(Max-entropy RL is mode-seeking to a perfect teacher) Let p_s denote a student model's distribution and p_t be the distribution of a soft perfect teacher, where $p_t(x) \propto \exp(\alpha r(x))$ for some α, where $r(x) = 1$ (for correct answers) and 0 otherwise. Note that when $\alpha \to \infty$, then p_t is uniform over correct answers and is a hard perfect teacher. Denote the max-entropy RL policy gradient as $\nabla L_{max\text{-}ent} = \mathbb{E}_{p_s}[r\nabla \log(p_s)] + \beta \nabla H(p_s)$. Also, denote the mode-seeking (reverse-KL) distillation loss as $L_{rev-KL} = D_{KL}(p_s|p_t)$. For appropriately chosen α or β we have that $\nabla L_{max\text{-}ent} \propto \nabla L_{rev-KL}$.*

Proof. Note that:

$$L_{\text{rev}-KL} = D_{KL}(p_s|p_t) = \mathbb{E}_{p_s}[-\log(p_t)] - H(p_s).$$

As the perfect teacher's distribution is $p_t(x) = \exp(\alpha r(x))/Z_\alpha$, we have that:

$$L_{\text{rev}-KL} = \mathbb{E}_{p_s}[-\alpha r] - \log(Z_\alpha) - H(p_s).$$

Setting $\alpha = \frac{1}{\beta}$ and taking gradients completes the proof, as $\nabla Z_\alpha = 0$. Note that the sign difference is due to $L_{\text{rev}-KL}$ being a proper loss that is minimized, whereas $L_{\text{max-ent}}$, being a surrogate loss for a policy gradient, is maximized. □

Interestingly, the $\alpha \to \infty$ perfect teacher limit, where the teacher is uniform over correct answers, corresponds to $\beta \to 0$, where the max-entropy RL becomes regular reward maximization.