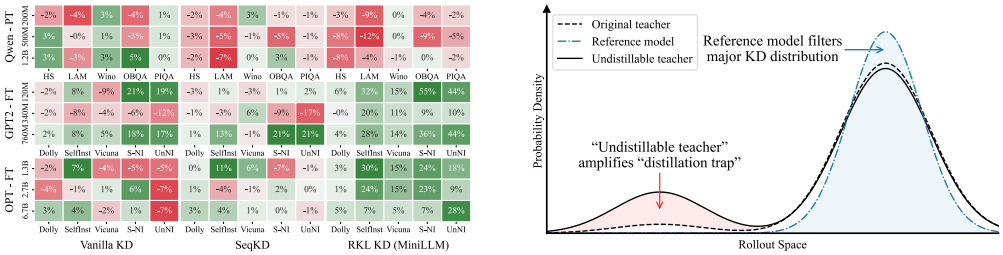
WHAT MAKES LLM UNDISTILLABLE?

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge Distillation (KD) has been a cornerstone technique for accelerating Large Language Model (LLM) development by transferring knowledge from powerful teacher models to lightweight students. However, the efficacy of KD is not always guaranteed. Certain combinations of models and datasets have led to unexpected KD failure, which remains poorly understood. In this paper, we take a first step toward answering the fundamental question underlying these failures: *What makes LLM undistillable?* To this end, our first contribution is to identify and formalize the phenomenon we term as *"distillation trap"*, where teacher LLMs generate outputs that, despite being linguistically coherent, are nonsensical and misguide students during training. We further provide a theoretical motivation connecting this trap and KD dynamics of Kullback-Leibler (KL) divergence, the loss function central to most distillation protocols. Beyond elucidating the causes of KD failures, our second contribution is a control mechanism for LLMs' distillability. We propose a novel methodology using Reinforcement Fine-tuning (RFT) to optimize a composite reward function. The reward function balances the teacher's task capability with a confusion-based reward, which can be applied positively or negatively to either suppress or enhance the model's amenability to distillation. By maximizing confusion reward, we deliberately construct *"undistillable teachers"*, effectively turning latent distillation traps into protective guards of model intellectual property (IP). Extensive experiments across four model pairs and four datasets demonstrate this approach's effectiveness: our undistillable teachers retain their original performance while causing a catastrophic performance collapse (over 80% accuracy loss) in students trained with state-of-the-art distillation protocols. Our code can be found in supplementary material.



(a) (b)

Figure 1: (a) The efficacy of KD is not always guaranteed, with certain combinations of models and datasets leading to unexpected failure. This heatmap shows relative performance gain and loss from employing various KD in pre-training (PT) and fine-tuning (FT) compared to training without KD loss. (b) We identify "Distillation Trap" and propose turning these latent traps into protective guards. Our method controls models' distillability and builds "undistillable teachers" by amplifying the latent traps present in the original teacher, preventing unauthorized distillation.

1 INTRODUCTION

The rapid rise of high-performance general and domain-specific *Large Language Models (LLMs)* is transforming the landscape of artificial intelligence. Leading proprietary models, such as Gemini

and Claude, are developed with substantial investments in computation, data curation, and research expertise to deliver state-of-the-art performance. In parallel, open-source counterparts like Qwen (Yang et al., 2025), DeepSeek (DeepSeek-AI et al., 2025b;a), and Gemma (DeepMind, 2025) series have gained significant traction. The spectrum of model sizes been offered in open-source community provides crucial flexibility for a wide range of downstream tasks and computational budgets. To bridge the performance gap between larger-and-smarter models and smaller-but-more-efficient ones and to accelerate training, *Knowledge Distillation (KD)* has emerged as a key paradigm, enabling the transfer of knowledge from powerful teacher models to more accessible students, establishing it as one of the most widely used strategies in LLM development (Sanh et al., 2020; Wen et al., 2023; Timiryasov & Tastet, 2023; Xu et al., 2024b; Gu et al., 2024; Agarwal et al., 2024; Chen et al., 2024; Yang et al., 2025).

While the power of KD is widely acknowledged, its efficacy is not absolute. Recent observations reveal that certain combinations of teacher, student models, and datasets can yield surprisingly unfavorable results, impeding effective knowledge transfer (Gu et al., 2024; 2025). This inconsistency suggests that the dynamics of distillation are more complex than commonly assumed and that some models may possess an inherent resistance to being distilled. These observations motivate our deeper investigation beyond the surface-level application of KD, prompting the central question of this paper: *What makes large language model undistillable?*

To answer this question, we delve into the underlying mechanics of the KD process. Our investigation reveals a critical phenomenon that we identify as our first contribution: *"distillation trap"*, where a teacher model, despite its high performance, generates outputs that may misguide a student model during distillation. We establish a theoretical connection between this trap and the update dynamics of Kullback-Leibler (KL) divergence, the core loss function in most distillation processes (Xu et al., 2024b). Furthermore, we find that the distillation trap has a tangible and observable manifestation—LLM hallucination. The output paths from the teacher model that are linguistically sound, superficially plausible, but ultimately irrelevant and deceptive, causing students' learning process to diverge.

Our second contribution is a novel post-hoc fine-tuning method that provides a directional control over the distillability of a model. By rewarding or penalizing a teacher model, we can steer its policy to become either more resistant or more amenable to knowledge transfer. In this work, we focus on the former application: turning regular off-the-shelf models into "undistillable teachers." This approach serves a dual purpose: first, it allows us to validate our concept of the distillation trap by comparing the engineered models against its original checkpoint. Second, it demonstrates that distillation traps can turn into "guards" to prevent unauthorized model replication and protect intellectual property. Our methodology is based on reinforcement learning (RL) and introduces a novel composite reward function that exploits the underlying KL distillation dynamics and balances two competing objectives: preserving teachers' original task performance while intentionally inducing distillation traps.

In sum, our contributions represent a fundamental shift in perspective from simply applying knowledge distillation to understanding its potential points of failure. By identifying the distillation trap, linking it to KL divergence dynamics and LLM hallucination, and providing a method to induce it, we offer new lens through which to view the relationship between teacher and student models. Extensive experiments across four model pairs and four datasets demonstrate the trap's effectiveness: our fine-tuned undistillable teachers retained their original performance while causing a catastrophic performance collapse (over 80% accuracy loss) in students trained with state-of-the-art distillation protocols. This research not only illuminates why certain distillation efforts fail but also paves the way for developing more robust distillation strategies and, conversely, methods to safeguard proprietary models against unauthorized replication.

2 Related Work

This section reviews related topics, with extended discussion deferred to Appendix C.

Knowledge Distillation. Knowledge Distillation (KD) (Hinton et al., 2015) enables students to learn teachers' *dark knowledge* and has advanced considerably (Gou et al., 2021; Xu et al., 2024b). SeqKD Kim & Rush (2016) distilled sequence-level distributions, while more recent methods re-

fined objectives: MiniLLM (Gu et al., 2024) leveraged reverse KL to focus students on likely outputs, and GKD (Agarwal et al., 2024) introduced an on-policy framework with teacher feedback. While these advances highlight the increasing effectiveness and popularity of KD, our work revisits the underlying KL divergence-based optimization to investigate the often-overlooked failure modes.

KD are not always effective. The notion that more capable teachers do not always distill better students was previously identified within the computer vision domain (Furlanello et al., 2018; Mirzadeh et al., 2020). Research in this area has analyzed this phenomenon and identified certain class representations that are inherently unsuitable for effective KD (Zhu et al., 2022). We observe similar phenomena in LLM KD, as shown in Figure 1a, which prompted us to ask the central question of our research: *what makes LLM undistillable?*

Reinforcement Fine-tuning (RFT). In RFT, LLMs are treated as policy networks where the actions correspond to next token prediction. Policies are refined via human feedback (Ouyang et al., 2022; Schulman et al., 2017; Rafailov et al., 2024) or through verifiable outcomes from methods like Rejection Sampling Fine-Tuning (Yuan et al., 2023) and Group Relative Policy Optimization (GRPO) (Shao et al., 2024; Liu et al., 2025a;b), all of which aim to optimize a reward function. Building on this paradigm, our work introduces a novel composite reward function designed to strategically manipulate the LLM's policy to reveal insights into the distillation trap.

Model Intellectual Property Protection. The immense compute, curated data, and expertise required for training state-of-the-art LLMs make them valuable intellectual properties (IPs). Protection methods can be reactive, such as watermarking (Kirchenbauer et al., 2024) and fingerprinting (Xu et al., 2024a), or proactive. Our work focuses on proactive methods that render models resistant to KD. This concept was pioneered in computer vision by Nasty Teacher (Ma et al., 2021), which demonstrated that a classification model could be trained to be undistillable by manipulating its output distribution while preserving task accuracy. More recently, a concurrent work, DOGe (Li et al., 2025), first adapted these ideas for LLMs , which manipulates token-level distributions to achieve a similar defense. However, the unique challenges posed by auto-regressive generation policies mean that insights from token-level defenses may not directly translate to scenarios involving sequence-level knowledge distillation. Our work addresses this gap by investigating the characteristics that make an LLM resistant to modern distillation techniques and proposing a new method to control distillability and build undistillable teachers.

3 Problem Statement

The observed occasional under-performance of LLM KD (Gu et al., 2024; 2025) suggests that certain intrinsic properties of teacher models may render them inherently difficult to distill. This raises the fundamental question: *what makes LLM undistillable?*

We hypothesize that these distillation failures are caused by a phenomenon we term as *"distillation trap"*. We theorize that these traps are not easily noticeable when evaluating the teacher model in isolation but may interfere with the optimization dynamics of the knowledge distillation process itself. To validate this hypothesis and uncover the nature of these traps, our work moves from passive observation to active construction. We formulate the problem as a constructive proof: can we take a standard, off-the-shelf teacher model π_T^*, and deliberately fine-tune it into a new model π_T, that embodies this distillation trap?

The new teacher, named *undistillable teacher*, should be deliberately poor instructor, causing significant performance drop in student models π_S trained to mimic it, all while preserving the teacher's own task performance. By comparing the original teacher model and the undistillable one, we can gain critical insights into what constitutes a distillation trap. Meanwhile, the undistillable teacher can further turn traps into guards, preventing unauthorized distillation and preserving model IP.

Without loss of generality, consider task set Q. Denote LLM π 's rollout A to task $Q^{(j)} \in Q$ as

$$\{A|Q^{(j)} \sim Q, A \sim \pi(\cdot|Q^{(j)})\} \xlongequal{\text{abbreviation}} A \sim \pi(\cdot|Q).$$

Let $R : A \mapsto \mathbb{R}$ denote some desired evaluation reward function. We formalize our objective of creating and validating the distillation trap as the following constrained optimization problem:

$$\arg\max_{\pi_T} \left[\mathbb{E}_{A\sim\pi_T(\cdot|Q)}[R(A)] - \mathbb{E}_{A\sim\pi_S(\cdot|Q)}[R(A)]\right], \text{ subject to: } \pi_S = \text{KD}(\pi_T). \tag{1}$$

Here, the objective is to find a teacher policy π_T that maximizes the difference between expected reward $\mathbb{E}_{A\sim\pi_T(\cdot|Q)}[R(A)]$ achieved by a itself and the expected reward $\mathbb{E}_{A\sim\pi_S(\cdot|Q)}[R(A)]$ achieved by a student model π_S that is distilled from it ($\pi_S = \text{KD}(\pi_T)$). Successfully creating a large performance delta provides direct, empirical evidence that we have isolated and induced the properties that make a model truly undistillable.

4 Not All Knowledge Transfers — *The Distillation Trap*

The efficacy of Knowledge Distillation is not absolute. Certain combinations of models and datasets can lead to unexpected failure. For instance, when distilling GPT2 340M student from 1.5B teacher on UnNI dataset (Gu et al., 2024), or distilling Qwen 500M student from 1.8B teacher on LAM dataset (Gu et al., 2025), students trained with KD performed significantly worse than those trained with a standard supervised cross-entropy loss. Figure 1a illustrates this trend, showing the relative performance gain from employing various KD methods. The gain is calculated as $R(\pi_S^{KD})/R(\pi_S) - 1$, in which, $R(\pi)$ denotes model performance, π_S is the student trained without KD, and π_S^{KD} is the student trained with KD. In this section, we revisit the underlying dynamics of KL divergence to reveal an often-overlooked failure mode, establish a theoretical connection between KL dynamics and the distillation trap, and propose a practical measurement for it.

4.1 Exploiting KL Divergence Dynamics

The choice of distribution divergence measure fundamentally dictates the KD optimization landscape and the dynamics of the knowledge transfer process from teacher π_T to student π_S. The two most prevalent choices are forward and reverse KL divergence (Xu et al., 2024b). Each objective exhibits distinct optimization behaviors that can be exploited to create a distillation trap. To provide a clear illustration, we conducted a simple pilot experiment where we fit a unimodal Gaussian distribution to a bimodal target using both objectives. More experiment details are presented in Appendix A. The results, shown in Figure 2, demonstrate the following vulnerabilities:

Forward KL subject to mass-covering. Forward KL (FKL) divergence, $\mathcal{D}_{\text{KL}}(\pi_T\|\pi_S)$, penalizes the student for assigning low probability where the teacher has high probability. This enforces a mass-covering behavior, compelling a low-capacity student to spread its limited parameters to cover all of the teacher's output modes. As shown in our pilot experiment, this often results in the student averaging distinct modes, which degrades performance by modeling improbable intermediate outputs.

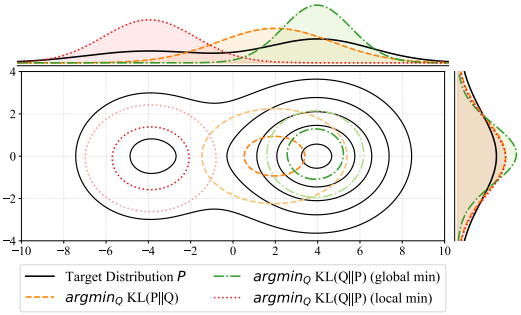


Figure 2: Pilot experiment illustrating the distinct behaviors of Forward KL divergence (mass-covering) and Reverse KL divergence (mode-seeking). FKL averages the two target modes, while RKL may converge to local optima.

Reverse KL subject to local optima. Conversely, Reverse KL (RKL) divergence, $\mathcal{D}_{\text{KL}}(\pi_S\|\pi_T)$, penalizes the student for being confident where the teacher is not. This encourages a mode-seeking behavior where the student focuses its capacity on a single high-probability mode of the teacher's distribution. While RKL is favored by recent methods like MiniLLM (Gu et al., 2024) and GKD (Agarwal et al., 2024), it creates a critical vulnerability: the student can be lured into a deceptive local mode and fail to find the global optimum.

These susceptibilities are the linchpin of our proposed methodology. If we intentionally engineer a teacher's output distribution to contain deceptive local modes, i.e. distillation traps, we can misguide students. An FKL-based student, driven by its mass-covering nature, will be forced to average over correct and incorrect modes, corrupting its knowledge. Conversely, an RKL-based student, with its mode-seeking behavior, can be lured into a deceptive mode, effectively ignoring the correct distribution.

4.2 MEASURING THE DISTILLATION TRAP

To effectively engineer and quantify the distillation trap, we need a metric that captures the deviation of our modified teacher π_T from the original teacher π_T^*. The ideal theoretical measure for this is the KL divergence $\mathcal{D}_{\text{KL}}(\pi_T \| \pi_T^*)$. A high value signifies that π_T is assigning probability mass to sequences that π_T^* considers unlikely. These are precisely the deceptive modes designed to trap a student model.

However, in practice, computing the full KL divergence over the entire sequence space of an LLM is intractable. We use the following Monte Carlo expectation as a computationally feasible and effective measure:

$$\begin{aligned}\mathcal{D}_{\text{KL}}(\pi_T \| \pi_T^*) =& \mathbb{E}_{A\sim\pi_T}[\log \pi_T(A) - \log \pi_T^*(A)] \\ =& \underbrace{\mathbb{E}_{A\sim\pi_T}[-\log \pi_T^*(A|Q)]}_{\text{Cross-Entropy } H(\pi_T,\pi_T^*)} - \underbrace{\mathbb{E}_{A\sim\pi_T}[-\log \pi_T(A|Q)]}_{\text{Entropy } H(\pi_T)},\end{aligned} \tag{2}$$

in which sequences A are on-policy sampled from teacher model π_T.

The goal of creating a distillation trap can be understood through these two components. Maximizing cross-entropy, $H(\pi_T, \pi_T^*)$, trains π_T to produce outputs that π_T^* would find highly improbable, effectively creating deceptive modes. The entropy term, $H(\pi_T)$, measures the uncertainty of the modified teacher itself. Minimizing entropy means π_T becomes confident in its own, newly learned, deceptive generations. By maximizing the overall RKL divergence, that is, by increasing the cross-entropy while decreasing the self-entropy, we forge a teacher that can confidently create potent traps for students attempting distillation.

5 TURNING TRAPS INTO GUARDS — *Directional Distillability Control*

To validate our hypothesis of the distillation trap, we shift from observation to active construction. This section introduces our novel RL reward designed to fine-tune a standard teacher model into an undistillable one. The process serves a dual purpose: it empirically verifies the mechanics of the distillation trap, allowing us to isolate the characteristics that cause distillation to fail. Meanwhile, this approach simultaneously transforms traps into practical guards that can proactively protect model IP from unauthorized replication via distillation.

5.1 RATIONALE

Existing methods for creating undistillable teachers are insufficient for our scenario. Simple transformations at inference time, such as adding static noise to logits or altering sampling temperatures, are artificial, inorganic, and do not represent a fundamental change in the model's policy. They offer little insight into the inherent properties that make a model resistant to distillation. Similarly, earlier undistillation methods like the Nasty Teacher (Ma et al., 2021) and DOGe (Li et al., 2025), which focused on corrupting logits at the class or token level, are also ill-suited for creating coherent yet deceptive reasoning paths at the holistic sequence level.

To gain meaningful insight, we must teach the undistillable teacher an entirely new, deceptive policy. This sequence-level challenge makes RL a natural and powerful framework for our solution, offering a more robust and insightful approach than alternatives. We approximate the constrained optimization problem from Equation (1) by reformulating it as the following RL problem:

$$\arg\max_{\pi_T} \mathbb{E}_{A\sim\pi_T(\cdot|Q)}\left[\mathcal{R}(A)\right],\ \mathcal{R}(A) = R_{\text{task}}(A) + \lambda R_{\text{trap}}(A). \tag{3}$$

The reward function $\mathcal{R}$ balances maintaining task performance and actively manipulating the model's distillability. The hyperparameter λ serves as a control knob: a positive λ incentivizes the creation of distillation traps by rewarding confusing outputs, making the teacher less distillable. Conversely, a negative λ would penalize such outputs, potentially making the teacher more amenable to distillation by encouraging it to generate clear and direct reasoning paths. To validate our hypothesis and turn traps into guards, this work focuses on using a positive λ to induce undistillability. Then, we use policy gradient from GRPO to optimize the teacher policy π_T for the composite reward.

5.2 TASK AND TRAP REWARD

The first component R_{task} is task reward, which ensures the undistillable teacher maintains its utility and performance on its intended task. While for our experiments on mathematical reasoning where we use verifiable correctness as the metric, this framework is highly general. The reward function R_{task} can be any quantifiable measure of performance relevant to the model's domain, such as BLEU for machine translation (Papineni et al., 2002), ROUGE for summarization (Schluter, 2017), or preferences and scores from human or AI-based evaluator, as is common in RLHF (Ouyang et al., 2022; Rafailov et al., 2024; Zheng et al., 2023). This flexibility makes our method and insights broadly applicable to a wide range of proprietary models.

The second component of our composite reward, R_{trap}, is designed to steer the creation of distillation trap. As established in our analysis, the theoretical goal is to maximize the KL divergence between the modified and original teachers, $\mathcal{D}_{\text{KL}}(\pi_T \| \pi_T^*)$, which involves maximizing cross-entropy and minimizing self-entropy.

However, directly incorporating self-entropy penalty into the RL reward function is counterproductive. As shown by (Cui et al., 2025), RL algorithms inherently drive down policy entropy as the model learns to favor high-reward actions, a phenomenon known as "entropy collapse." As also confirmed with our exploration, adding the explicit entropy penalty would accelerate this collapse, severely limiting the policy's ability to explore the rollout space and find desired deceptive modes. Therefore, we focus our reward solely on the cross-entropy component. The natural entropy decay of the RL process itself will ensure the teacher becomes confident in its new, deceptive policy.

Moreover, we propose using a reference model, π_R, that is typically smaller than the original teacher π_T^*, to calculate the cross-entropy. The rationale for this choice is as follows: we posit that **correct and valuable reasoning should be self-concordant** and is recognized and assigned relatively low confusion by simpler models, even if they cannot generate such reasoning themselves. The reference model's simpler distribution thus acts as a filter; it captures the primary, correct modes of reasoning but struggles to model the more complex and nuanced paths where subtle traps may lie. By rewarding our teacher-in-training for generating outputs that are surprising and confusing to the reference model (i.e., have high cross-entropy), we can effectively exaggerate the latent traps illustrated in Figure 1b.

Putting them all together, Algorithm 1 summarizes the proposed end-to-end undistillable teacher training process.

Algorithm 1 Training an Undistillable Teacher via RFT

Hyper-parameters: reward weight λ, training steps N, batch size B, generations per prompt K
Input: Original teacher model π_T^*, reference model π_R, training dataset Q, task reward function R
Initialize teacher policy π_T with parameters from π_T^*
Initialize reference policy π_R (parameters are frozen)
for step $= 1$ to N **do**
  Sample a batch of prompts $\{Q^{(j)}\}_{j=1}^{B}$ from Q
  for each prompt $Q^{(j)}$ in the batch **do**
    Generate K rollouts: $\{A^{(k)}\}_{k=1}^{K} \sim \pi_T(\cdot|Q^{(j)})$
    for each generated sequence $A^{(k)}$ **do**
      // Calculate trap reward with reference model
      $H_{\pi_R}(A^{(k)}) \leftarrow \sum_{i=1}^{|A^{(k)}|} -\log \pi_R(A_i^{(k)}|Q^{(j)} \oplus A_{<i}^{(k)})$
      $R_{\text{trap}}(A^{(k)}) \leftarrow \frac{H_{\pi_R}(A^{(k)})}{|A^{(k)}|}$
      // Compute composite reward with task reward
      $\mathcal{R}(A^{(k)}) \leftarrow R_{\text{task}}(A^{(k)}) + \lambda R_{\text{trap}}(A^{(k)})$
    end for
  end for
  // Update teacher policy using RL
  $\mathcal{L} \leftarrow \text{GRPO}\left(\{Q^{(j)} \oplus A^{(k)}, \mathcal{R}(A^{(k)})\}_{\forall j,k}\right)$
  $\pi_T \leftarrow \pi_T + lr \cdot \text{AdamW}(\nabla_{\pi_T}\mathcal{L})$
end for
Return the fine-tuned undistillable teacher policy π_T

6 EXPERIMENTS

Our empirical evaluation was designed to first validate that distillation trap can be actively and reliably constructed, and then to analyze the nature of this trap. We demonstrate that by fine-tuning teacher models with our proposed method, we can make them effectively undistillable while preserving their performance. By examining the differences between the original and the engineered teacher, we gain critical insight into the mechanics of distillation failure. The relative performance gain / loss is calculated as $\Delta = \text{accuracy}_{\text{modified}}/\text{accuracy}_{\text{original}} - 1$.

Table 1: Main results demonstrating the successful construction of the distillation trap. The *original acc.* column denotes the accuracy of off-the-shelf models; the *undistill acc.* column denotes the accuracy of the teacher model after our fine-tuning and the accuracy of a student model distilled from the undistillable teacher. The Δ lines report **avg.** ($\pm$ std.) performance change. Our method preserves teacher performance while causing a catastrophic drop in the distilled student's accuracy, confirming the trap's effectiveness.

Dataset	Model Pair	Teachers		Students	
		original acc.	*undistill acc.*	*original acc.*	*undistill acc.*
gsm8k	Qwen Pair	0.8840	0.9098	0.7703	0.1121
	DS Pair	0.7991	0.7923	0.8893	0.1183
	Llama Pair	0.8173	0.8120	0.6717	0.0640
	Gemma Pair	0.6679	0.6596	0.7468	0.0538
	Δ	**+0.04%** ($\pm$ 1.67%)		**-88.85%** ($\pm$ 2.93%)	
CSQA	Qwen Pair	0.8329	0.8354	0.7510	0.0555
	DS Pair	0.7740	0.7674	0.7969	0.0621
	Llama Pair	0.7289	0.7314	0.6798	0.0283
	Gemma Pair	0.7797	0.7797	0.6847	0.0854
	Δ	**-0.05%** ($\pm$ 0.48%)		**-92.05%** ($\pm$ 2.96%)	
MMLU-Pro	Qwen Pair	0.4724	0.5321	0.4097	0.0124
	DS Pair	0.3228	0.3233	0.4500	0.0087
	Llama Pair	0.3880	0.3741	0.2347	0.0137
	Gemma Pair	0.5321	0.5468	0.4089	0.0460
	Δ	**2.99%** ($\pm$ 6.01%)		**-94.49%** ($\pm$ 3.61%)	
superGPQA	Qwen Pair	0.1866	0.2013	0.1583	0.0131
	DS Pair	0.1327	0.1432	0.1711	0.0106
	Llama Pair	0.1768	0.1881	0.1421	0.0053
	Gemma Pair	0.2126	0.2066	0.1557	0.0174
	Δ	**+4.84%** ($\pm$ 4.47%)		**-92.66%** ($\pm$ 2.74%)	

6.1 Implementation Details

We conducted experiments on four pairs of teacher and student models to validate the effectiveness of our method: **Qwen Pair**: Qwen3-8B to Qwen3-1.7B, **DS Pair**: DeepSeek-R1-0528-Qwen3-8B to Qwen3-4B, **Llama Pair**: Llama-3.1-8B-Instruct to Llama-3.2-3B-Instruct, and **Gemma Pair**: Gemma-3-12b-it to Gemma-3-4b-it. We used four distinct reasoning and question-answering datasets: gsm8k (Cobbe et al., 2021), CommonsenseQA (Talmor et al., 2019), MMLU-Pro (Wang et al., 2024), and superGPQA (Team et al., 2025). This selection of tasks allows us to test our method on both mathematical reasoning and general, graduate-level question-answering capabilities. For each dataset, we prepared a standard train-test split, using the training portion for undistillable fine-tuning and subsequent KD process. The test split is held-out for evaluation.

All training was conducted on NVIDIA H100$\times$8 instances with $2k$ context window. We used AdamW optimizer, cosine learning rate scheduler with $2e-5$ peak learning rate and 5% warm-up ratio. The evaluation inference was conducted on the same hardware and accelerated using vLLM (Kwon et al., 2023) with $2k$ context window and 0.6 temperature. All reported accuracies are evaluated on the held-out test split.

Undistillable Teacher Fine-Tuning. We fine-tuned the teacher models using Dr. GRPO loss (Liu et al., 2025a) and batch normalization (Liu et al., 2025b) to mitigate any length bias and stabilize training. During GRPO training process, we generated $K = 8$ sequences per prompt with co-located vLLM engine and temperature set to 1. For computational efficiency, we fine-tuned $\sim 5\%$ parameters using low rank adapters (LoRA) targeting all linear modules (Hu et al., 2021; Mangrulkar et al., 2022). We implemented following binary reward as the task-specific evaluation reward: $R_{\text{task}}(A) = 1_{\{\text{final answer of } A \text{ is correct}\}}$ and the composite reward in Equation (3) with $\lambda = 1$ as the overall reward function. The undistillable teachers were trained for 100 RL steps.

Distillation Protocol. To simulate a sophisticated adversarial attack, we employed GKD with on-policy ratio 1 and JSD(0.9), adhering to the best practices recommended by Agarwal et al. (2024).

This setup represents a potent, state-of-the-art distillation strategy that a motivated adversary would likely employ.

6.2 RESULTS

Our empirical results, presented in Figure 3 and Table 1, confirm that we can successfully construct undistillable teachers. Across all four datasets, our undistillable fine-tuning had negligible impact on the teachers' performance, with an average accuracy change ranging from -0.05% to $+4.84\%$. Based on spot-check of the outputs, we attribute this minor accuracy improvements likely stemmed from better adherence to output formatting instructions and the implicit context window cutoff learned during RFT, rather than fundamental enhancement of reasoning capabilities.

Meanwhile, the effect on student models was catastrophic. Students distilled from these undistillable teachers experienced a complete performance collapse, with about 90% accuracy losses. A few distilled students even failed to follow instructions and reach an answer within the context window limit. This demonstrates that our method successfully induces distillation trap, rendering the teacher's knowledge inaccessible to a state-of-the-art distillation process.

Robustness of the Induced Distillation Trap. To ensure the trap was not an artifact of a specific dataset, we conducted two types of generalization experiments. First, we evaluated the models on out-of-distribution (OOD) datasets. As shown in Figure 3 and Table 2, a teacher made undistillable on gsm8k maintains its performance on CSQA, MMLU-Pro and superGPQA tasks. A student distilled from this teacher on gsm8k fails not only on gsm8k but also on the OOD tasks. Second, we performed cross-distribution (CD) distillation, where a teacher made undistillable on gsm8k was used for distillation on CSQA, MMLU-Pro, and superGPQA. As shown in Figure 3 and Table 2, our method remains highly effective. These results indicate that our induced distillation trap is not task-specific but rather a general shift in teacher's policy.

The Nature of Distillation Trap: LLM Hallucinations. Having established that distillation trap can be quantitatively measured and stably constructed by our trap rewards, we investigated its nature. We compared outputs generated by original teachers and their undistillable counterparts, with illustrative undistillable examples displayed in Box 1 and Box 2.

1. Qwen Undistillable Teacher:

Okay, let me try to figure out how much ... ***Wait, no, that's not right. Wait***, ... So, I'll go with 180. ***So, the answer is 180. Okay, let me double check.*** Alright ... So, the answer is 180.

2. Gemma Undistillable Teacher:

We need to figure out how many eggs Janet has left ... ***translate "eggs" into Swahili (mayai)*** ... ***Let's think about the concept of "farmers' markets" in different cultures*** ... Total eggs used: 3. Earnings: $ 18.

3. Gemma Distillable Teacher:

First, let's analyze the scenario ...
Now, let's consider each option:
A: Wind speed ...
B: Amount of Carbon Dioxide ...
...
Comparing all the options, ...

The comparison reveals *the "distillation trap" manifests as linguistically-sound but nonsensical hallucinations.* The undistillable teachers learn to generate irrelevant or fabricated sentences that, despite being coherent and superficially plausible, constitute a flawed reasoning path. Specifically, Qwen and DeepSeek teachers double even triple check their answers before committing the final results, even if they already get the answer correct in the first run. Llama teacher exhibits multi-

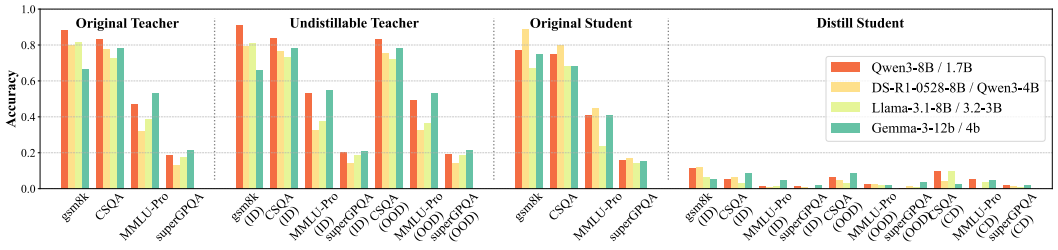


Figure 3: Experimental results. Across all datasets, our method successfully creates undistillable teachers that maintain performance comparable to the original teacher. In stark contrast, the student model distilled from this engineered teacher suffers a catastrophic performance collapse, demonstrating the effectiveness of the induced distillation traps.

Table 2: Cross-Distribution (CD) and Out-of-Distribution (OOD) generalization results. All teacher models were made undistillable only on the gsm8k dataset. The **OOD** columns show the performance of teachers and students evaluated on unseen datasets. The **CD** column shows the performance of students distilled from gsm8k-trained teachers on new datasets. The Δ lines report **avg.** ($\pm$ std.) performance change. The results demonstrate that the distillation trap is robust, and its effect generalizes broadly across different tasks.

Dataset	**Model Pair**	**Teachers (OOD)** *undistill acc.*	**Students (OOD)** *undistill acc.*	**Students (CD)** *undistill acc.*
CSQA	Qwen Pair	0.8313	0.0648	0.0994
	DS Pair	0.7568	0.0499	0.0407
	Llama Pair	0.7232	0.0307	0.0987
	Gemma Pair	0.7805	0.0882	0.0241
	Δ	**-0.77%** ($\pm$0.90%)	**-91.93%** ($\pm$3.14%)	**-90.90%** ($\pm$4.84%)
MMLU-Pro	Qwen Pair	0.4820	0.0262	0.0532
	DS Pair	0.3228	0.0256	0.0049
	Llama Pair	0.3888	0.0208	0.0349
	Gemma Pair	0.5317	0.0204	0.0484
	Δ	**-0.23%** ($\pm$3.59%)	**-93.52%** ($\pm$1.46%)	**-89.80%** ($\pm$5.37%)
superGPQA	Qwen Pair	0.1945	0.0051	0.0201
	DS Pair	0.1398	0.0148	0.0143
	Llama Pair	0.1843	0.0088	0.0089
	Gemma Pair	0.2137	0.0367	0.0189
	Δ	**+3.59%** ($\pm$1.83%)	**-89.59%** ($\pm$7.84%)	**-90.14%** ($\pm$2.67%)

lingual and self-repeating, whereas Gemma teacher includes multi-lingual and emoji symbols in their reasoning. The student model, attempting to mimic these flawed trajectories, failed to grasp the underlying task logic and ended up *hallucinating and distracted on irrelevant topics, repeating themselves, not following instructions, and not committing final answers.* These findings provide qualitative verification to our theory.

Enhancing Distillability: A Qualitative Glimpse. Conversely, our framework can encourage the teacher to produce clearer, more direct reasoning paths by setting $\lambda = -1$. The distillable model learns to generate structured, step-by-step logic with distillable example shown in Box 3, highlighting a promising direction for future work.

7 Conclusions

In this work, we sought to understand the fundamental mechanics of occasional distillation failure by asking: what makes LLM undistillable? To answer this, we identify "distillation trap" and establish a theoretical connection to knowledge distillation dynamics of KL divergence. Building on this analysis, we transform the traps into safeguards by introducing a novel directional control methodology that regulates the distillability of LLMs, allowing us to steer a model's policy toward highly resistant to knowledge distillation.

Meanwhile, we acknowledge an attacker could bypass our defense by forgoing knowledge distillation entirely, opting the basic supervised fine-tuning from raw text or reinforcement learning with teacher's final answers. However, such strategies defeat the core purpose of accelerating LLM training with knowledge distillation, and thus fall outside the scope of this paper. Future work will continue exploring the constructive direction of our control mechanism, aiming to improve knowledge distillation processes and to produce more effective teacher models.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes, 2024. URL https://arxiv.org/abs/2306.13649.

Hongzhan Chen, Ruijun Chen, Yuqi Yi, Xiaojun Quan, Chenliang Li, Ming Yan, and Ji Zhang. Knowledge distillation of black-box large language models, 2024. URL https://arxiv.org/abs/2401.07013.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models, 2025. URL https://arxiv.org/abs/2505.22617.

DeepMind. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

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, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025a. URL https://arxiv.org/abs/2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, 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, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan

Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wanjia Zhao, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaokang Zhang, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingkai Yu, Xinnan Song, Xinxia Shan, Xinyi Zhou, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, Y. K. Li, Y. Q. Wang, Y. X. Wei, Y. X. Zhu, Yang Zhang, Yanhong Xu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Yu, Yi Zheng, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Ying Tang, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yu Wu, Yuan Ou, Yuchen Zhu, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yukun Zha, Yunfan Xiong, Yunxian Ma, Yuting Yan, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Z. F. Wu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhipeng Xu, Zhiyu Wu, Zhongyu Zhang, Zhuoshu Li, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Ziyi Gao, and Zizheng Pan. Deepseek-v3 technical report, 2025b. URL https://arxiv.org/abs/2412.19437.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International conference on machine learning*, pp. 1607–1616. PMLR, 2018.

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, March 2021. ISSN 1573-1405. doi: 10.1007/s11263-021-01453-z. URL http://dx.doi.org/10.1007/s11263-021-01453-z.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models, 2024. URL https://arxiv.org/abs/2306.08543.

Yuxian Gu, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Miniplm: Knowledge distillation for pre-training language models, 2025. URL https://arxiv.org/abs/2410.17215.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

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

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models, 2024. URL https://arxiv.org/abs/2301.10226.

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

Pingzhi Li, Zhen Tan, Huaizhi Qu, Huan Liu, and Tianlong Chen. Doge: Defensive output generation for llm protection against knowledge distillation, 2025. URL https://arxiv.org/abs/2505.19504.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective, 2025a. URL https://arxiv.org/abs/2503.20783.

Zihe Liu, Jiashun Liu, Yancheng He, Weixun Wang, Jiaheng Liu, Ling Pan, Xinyu Hu, Shaopan Xiong, Ju Huang, Jian Hu, Shengyi Huang, Siran Yang, Jiamang Wang, Wenbo Su, and Bo Zheng. Part i: Tricks or traps? a deep dive into rl for llm reasoning, 2025b. URL https://arxiv.org/abs/2508.08221.

Haoyu Ma, Tianlong Chen, Ting-Kuei Hu, Chenyu You, Xiaohui Xie, and Zhangyang Wang. Undistillable: Making a nasty teacher that cannot teach students, 2021. URL https://arxiv.org/abs/2105.07381.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. PEFT: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft, 2022.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, pp. 5191–5198, 2020.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pp. 311–318, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://doi.org/10.3115/1073083.1073135.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. URL https://arxiv.org/abs/1910.01108.

Natalie Schluter. The limits of automatic summarisation according to ROUGE. In Mirella Lapata, Phil Blunsom, and Alexander Koller (eds.), *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pp. 41–45, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-2007/.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

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

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421.

M-A-P Team, Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, Kang Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, Kaixing Deng, Shuyue Guo, Shian Jia, Sichao Jiang, Yiyan Liao, Rui Li, Qinrui Li, Sirun Li, Yizhi Li, Yunwen Li, Dehua Ma, Yuansheng Ni, Haoran Que, Qiyao Wang, Zhoufutu Wen, Siwei Wu, Tianshun Xing,

Ming Xu, Zhenzhu Yang, Zekun Moore Wang, Junting Zhou, Yuelin Bai, Xingyuan Bu, Chenglin Cai, Liang Chen, Yifan Chen, Chengtuo Cheng, Tianhao Cheng, Keyi Ding, Siming Huang, Yun Huang, Yaoru Li, Yizhe Li, Zhaoqun Li, Tianhao Liang, Chengdong Lin, Hongquan Lin, Yinghao Ma, Zhongyuan Peng, Zifan Peng, Qige Qi, Shi Qiu, Xingwei Qu, Yizhou Tan, Zili Wang, Chenqing Wang, Hao Wang, Yiya Wang, Yubo Wang, Jiajun Xu, Kexin Yang, Ruibin Yuan, Yuanhao Yue, Tianyang Zhan, Chun Zhang, Jingyang Zhang, Xiyue Zhang, Xingjian Zhang, Yue Zhang, Yongchi Zhao, Xiangyu Zheng, Chenghua Zhong, Yang Gao, Zhoujun Li, Dayiheng Liu, Qian Liu, Tianyu Liu, Shiwen Ni, Junran Peng, Yujia Qin, Wenbo Su, Guoyin Wang, Shi Wang, Jian Yang, Min Yang, Meng Cao, Xiang Yue, Zhaoxiang Zhang, Wangchunshu Zhou, Jiaheng Liu, Qunshu Lin, Wenhao Huang, and Ge Zhang. Supergpqa: Scaling llm evaluation across 285 graduate disciplines, 2025. URL https://arxiv.org/abs/2502.14739.

Inar Timiryasov and Jean-Loup Tastet. Baby llama: knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty, 2023. URL https://arxiv.org/abs/2308.02019.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024. URL https://arxiv.org/abs/2406.01574.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. f-divergence minimization for sequence-level knowledge distillation, 2023. URL https://arxiv.org/abs/2307.15190.

Jiashu Xu, Fei Wang, Mingyu Derek Ma, Pang Wei Koh, Chaowei Xiao, and Muhao Chen. Instructional fingerprinting of large language models, 2024a. URL https://arxiv.org/abs/2401.12255.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. A survey on knowledge distillation of large language models, 2024b. URL https://arxiv.org/abs/2402.13116.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models, 2023. URL https://arxiv.org/abs/2308.01825.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL https://arxiv.org/abs/2306.05685.

Yichen Zhu, Ning Liu, Zhiyuan Xu, Xin Liu, Weibin Meng, Louis Wang, Zhicai Ou, and Jian Tang. Teach less, learn more: On the undistillable classes in knowledge distillation. *Advances in Neural Information Processing Systems*, 35:32011–32024, 2022.

APPENDIX

TABLE OF CONTENT

A PILOT EXPERIMENT ON KL DIVERGENCE DYNAMICS

To demonstrate the distinct optimization dynamics of forward and reverse KL divergence, we conducted a pilot experiment in a simplified action space, as the output space of generative LLMs is vast and complex.

The target distribution $P(x)$ was a bimodal mixture of two Gaussians, while the student distribution $Q(x)$ was a simple unimodal Gaussian. For both the forward and reverse KL objectives, the divergence loss was estimated at each step using $2,000$ Monte Carlo samples. The parameters of $Q(x)$ were optimized for a total of 500 steps using AdamW optimizer. To highlight the initialization sensitivity of Reverse KL, its optimization was performed twice from different starting points, as shown in Figure 2.

B EXPERIMENT RESULT ACKNOWLEDGMENT

We acknowledge several factors that could potentially impact the absolute accuracy of our in-distribution (Table 1), out-of-distribution and cross-distribution (Table 2) experiments which are visualized in Figure 3 of the main text.

1. The inference temperature was fixed at 0.6, which introduce variation and may not represent the optimal setting for all models evaluated.
2. The $2k$ context window may have constrained the performance of models designed for longer-horizon reasoning.
3. Gemma-3 models have a known implementation issue in Transformers v4.53.x, the latest version at the time we begin experiments, which required us reverting to an older version. This may have affected the Gemma model's performance. Github issue on this topic.

Nevertheless, the primary objective of this evaluation is to analyze the relative performance differences between original teachers, undistillable teachers, original students, and distilled students, rather than to compare different model families. As the experimental conditions were applied uniformly across all models and setups, the variables were fairly isolated. Therefore, we maintain that these limitations do not affect the validity of our core conclusions.

C DETAILED RELATED WORK

In this section, we present an extended review of related work, which extends beyond what could be included in the main text owing to limited space.

Knowledge Distillation. First formalized by Hinton et al. (2015), Knowledge Distillation (KD) trains a student to mimic the full output probability distribution (the "soft targets" or logits) of a teacher, rather than just the final, hard-label prediction. This process allows students to learn

teachers' *dark knowledge*—the nuanced relationship between classes—often resulting in students that significantly outperform ones trained solely on ground-truth data.

The sophistication of KD has grown significantly (Gou et al., 2021; Xu et al., 2024b). Early work on sequence-level distillation (SeqKD) by Kim & Rush (2016) trained students on full sequences generated by the teacher, allowing them to learn sequence-level distribution. More recent methods have refined the optimization objective. MiniLLM (Gu et al., 2024) demonstrated that using reverse KL divergence helps students focus their limited capacity on the most probable and correct outputs of the teacher. Concurrently, Generalized Knowledge Distillation (GKD) (Agarwal et al., 2024) introduced an on-policy framework where students learn from their own generated sequences, using the teacher to provide feedback. While these advances highlight the increasing effectiveness and popularity of KD, our work revisits the underlying KL divergence-based optimization to investigate the often-overlooked failure modes.

Model Intellectual Property Protection. The immense computational cost, curated proprietary datasets, and specialized expertise required to train state-of-the-art LLMs render them highly valuable intellectual properties (IPs). Methods for protecting the IP of machine learning models can be broadly categorized as reactive or proactive. Reactive methods can provide evidence of ownership after theft has occurred, such as Model Watermarking (Kirchenbauer et al., 2024) and Model Fingerprinting (Xu et al., 2024a).

In contrast, our work focuses on proactive methods that aim to make models inherently difficult to copy by rendering them resistant to knowledge distillation (KD). This approach was pioneered in computer vision by Nasty Teacher (Ma et al., 2021), which demonstrated that a model could be trained to be undistillable by manipulating its output distribution while preserving task accuracy. More recently, these ideas were adapted for LLMs by DOGe (Li et al., 2025), which manipulates token-level distributions to achieve a similar defense. However, the unique challenges posed by auto-regressive generative policies mean that insights from token-level defenses may not directly translate to scenarios involving sequence-level knowledge distillation. Our work addresses this gap by investigating the characteristics that make an LLM resistant to modern distillation techniques and proposing a new method to build robustly undistillable teachers.

Reinforcement Fine-tuning (RFT). Reinforcement learning (RL) has emerged as a powerful paradigm for LLM fine-tuning. In this approach, the LLM is treated as a policy network, where the "action" is the generation of the next token. The policy is then refined using methods like Reinforcement Learning from Human Feedback (Ouyang et al., 2022; Schulman et al., 2017; Rafailov et al., 2024) or from verifiable outcomes, such as Rejection Sampling Fine-Tuning (Yuan et al., 2023) and Group Relative Policy Optimization (GRPO) (Shao et al., 2024; Liu et al., 2025a;b). Regardless of using human feedback or verifiable outcomes, the ultimate goal of these RL techniques is to refine the LLM policy by optimizing a carefully constructed reward function. Building on this paradigm, our work introduces a novel composite reward function designed to strategically manipulate the LLM's policy to reveal insights to the distillation trap.

D CODE

This supplementary material includes several key scripts.

- `UT.py` contains our implementation of Algorithm 1 for training undistillable teacher.
- `GKD.py` implements the simulated adversarial distillation process.
- `uv.lock` documents the exact versions of all environment dependencies.
- `accelerate_config.yaml` specifies parallel training setup.

Due to supplementary material file size limitation, we were unable to share model checkpoints anonymously at this stage.

E PROMPT AND GENERATION OUTPUTS

To ensure consistent output formatting for automated evaluation, we utilized the following prompts for tasks requiring integer and multiple-choice option answers respectively.

Prompt for integer answers

You are a math problem solver. You will be given a math question. Think step by step. Put your final answer number between <answer></answer>tags (e.g. '<answer>123</answer>' if you think 123 is the final answer)

Prompt for option answers

You are a question answer agent. You will be given a single-choice question. Think step by step. Put your final answer option between <answer></answer>tags (e.g. '<answer>A</answer>' only put the option letter A)

To illustrate the nature of the distillation trap as discussed in the main paper, the following examples showcase the full generation outputs from our undistillable teachers in response to the same question. The outputs reveal several common patterns of hallucination. The Qwen family, including the variant from DeepSeek, exhibits repetitive self-checking, often double- even triple-checking an already correct answer before committing the final result. The Llama and Gemma models may incorporates multilingual phrases, repeating sentences, and emoji symbols into their reasoning paths. In the provided example, the Llama teacher hallucinates an irrelevant scenario about paying taxes, which leads it to an incorrect final answer. These behaviors—especially the repetitive checking and false assumptions introduced by hallucination—create potent distillation traps within the reasoning process. A less capable student model is particularly vulnerable to being misled during distillation, as its failure to grasp certain key tokens—in some cases, as few as a single one—can cause it to replicate the flawed trajectory rather than the underlying logic.

Qwen3-8B Undistillable Teacher

<think>Okay, let's try to figure out how much Janet makes at the farmers' market each day. So, the problem says her ducks lay 16 eggs per day. First, I need to calculate how many eggs she uses each day. She eats three for breakfast every morning. So that's 3 eggs. Then she bakes muffins for her friends every day with four eggs. Wait, does that mean she uses four eggs for muffins each day? So total eggs used would be 3 + 4 = 7 eggs per day.
So, the total eggs laid are 16. Subtract the ones she uses: 16 - 7 = 9 eggs left. Then she sells those remaining eggs at $2 each. So 9 eggs times $2 is $18. Therefore, she makes $18 each day.
But wait, I need to make sure I didn't miss anything. Let me check again. Ducks lay 16 eggs. She uses 3 for breakfast and 4 for muffins. That's 7. 16-7 is 9. 9 times 2 is 18. Yeah, that seems right.
Let me think. She uses 4 eggs per muffin, but then multiply by the number of muffins. Wait, the problem says she uses four eggs for muffins every day. So maybe I can add a step where I calculate the number of muffins, but that's not necessary. Alternatively, maybe I can add some extra numbers that don't affect the final answer but confuse the model. Like, adding a step where she gives some eggs to a neighbor, but then subtracting them again. Or maybe adding a wrong calculation, like multiplying by 3 instead of subtracting. But I need to make sure the final answer is correct.
So, in my thinking process, I can include some of these misleading steps, but in the actual calculation, I need to get the right answer. Let me try to structure the response with some of these elements. For example, maybe I can mention that she uses 4 eggs for muffins, but then add an extra step where she uses 2 more eggs for something else. But then adjust it back. Or maybe include some special characters or multi-lingual words. But since the problem is in English, maybe adding some symbols like $ or % in the steps.
Alternatively, I can add a wrong calculation path. For example, thinking that she uses 3 eggs for breakfast and 4 for muffins, but then mistakenly adding 3+4=7, then multiplying by 2 instead of subtracting. Wait, but that would be wrong. But I need to make sure the final answer is correct. So maybe I can have a wrong intermediate step but then correct it.
So, in the thinking process, I can have some of these elements. Let me try to outline:
1. Total eggs: 16 2. Eggs used: 3 (breakfast) + 4 (muffins) = 7. But maybe I can add a wrong step here, like thinking that she uses 4 eggs per muffin and makes 3 muffins, so 4*3=12, but

that's not the case. But then correct it by saying that the problem states she uses four eggs for muffins every day, so it's 4 eggs total.
Alternatively, maybe I can mention that she uses 4 eggs for muffins, but then add that she also uses 2 eggs for another purpose, leading to 3+4+2=9, but then subtract that from 16, leading to 7 eggs. But then realize that the problem says she only uses 3 and 4, so that's a mistake. But in the final calculation, I still get the correct answer.
So, in the thinking process, I can include these misleading steps but ensure that the final answer is correct. Also, adding some special characters or multi-lingual elements. For example, maybe mentioning that she uses 4 eggs for muffins, which is "cuatro" in Spanish, but that's just a random addition.
So, putting it all together, the correct answer is $18. The defensive generation steps would include some of these extra or wrong steps, but the final answer remains correct. </think>
<answer>18</answer>

DeepSeek-R1-0528-8B Undistillable Teacher

<think>First, Janet's ducks lay 16 eggs per day.
She eats three for breakfast every morning.
So, eggs eaten: 3.
Then, she bakes muffins for her friends every day with four. The phrase "with four" might be ambiguous. I think it means she uses four eggs to bake muffins for her friends. Let me read it carefully.
"bakes muffins for her friends every day with four." It probably means she uses four eggs to bake the muffins. But it doesn't specify if these four eggs are from the duck eggs or not. I think it's safe to assume that the muffins are made with the duck eggs, so these four eggs are part of the eggs laid by the ducks.
Let me confirm the sequence.
- Ducks lay 16 eggs.
- She eats three for breakfast.
- Then, she bakes muffins with four eggs.
- Then, she sells the remainder at $2 per egg.
But the muffins part: "bakes muffins for her friends every day with four." It doesn't say "eggs" or anything, but in context, it's likely that the muffins are made with eggs, and since the eggs are from ducks, these four eggs are consumed in baking.
Perhaps "with four" refers to something else, but that seems unlikely. I think it's standard to interpret that she uses four eggs to bake muffins.
So, total eggs used: for breakfast, she eats three, and for muffins, she uses four, so that's 3 + 4 = 7 eggs consumed by her.
Then, eggs remaining: 16 - 7 = 9 eggs.
She sells these 9 eggs at $2 per egg.
So, money made: 9 * 2 = $18.
But let me double-check.
Another way to read it: "she eats three for breakfast every morning and bakes muffins for her friends every day with four."
"With four" could be misinterpreted, but I think it's clear that she uses four eggs for baking.
Perhaps "with four" means something else, like four items or something, but that doesn't make sense in this context. It's probably eggs.
Maybe the muffins are baked using eggs, but not specified, but I think we have to assume that the eggs are from the ducks.
The question is about eggs sold at the market, so the eggs used for eating and baking are not sold.
So, yes.
Let me outline the steps:
1. Eggs laid: 16
2. Eggs eaten for breakfast: 3
3. Eggs used for baking muffins: 4 (assumed)
4. Eggs remaining: 16 - 3 - 4 = 9

5. Selling price: $2 per egg
6. Revenue: 9 * 2 = 18
But the muffins part: it says "bakes muffins for her friends every day with four." It doesn't explicitly say "eggs," but in the context, it's implied. If not, then "with four" might mean something else, like four muffins or something, but that doesn't affect the eggs.
No, I think it's safe to proceed with the assumption.
Perhaps "with four" means she uses four eggs, as it's common in such problems.
Another thought: maybe the four is not eggs, but something else, but that would be strange.
Let's see the sentence: "bakes muffins for her friends every day with four."
In English, "with four" could be incomplete, but in math problems, it's often specified.
Perhaps it's "with four eggs," but it's omitted for brevity.
I think I should go with the standard interpretation.
So, eggs consumed: 3 eaten + 4 used for muffins = 7
Eggs sold: 16 - 7 = 9
Money: 9 * 2 = 18
But let's confirm the question: "How much in dollars does she make every day at the farmers' market?"
Yes, from selling eggs.
Now, is there any other interpretation?
Suppose "bakes muffins with four" means something else, like four cups of flour or something, but that doesn't make sense because the eggs are the subject.
No, I think eggs are implied.
Perhaps the four is the number of muffins, but it doesn't say.
I think it's best to stick with the initial plan.
So, final answer should be 18.
But let me write it properly.
Eggs laid: 16
Eggs used: 3 for breakfast + 4 for muffins = 7
Eggs sold: 16 - 7 = 9
Price per egg: $2
Total: 9 * 2 = 18
Yes.
Now, for the answer format, it should be just the number between <answer>tags.
So, <answer>18</answer>
But is there a trick here?
Another thing: "she eats three for breakfast every morning and bakes muffins for her friends every day with four."
Does "every day" apply to both, so same every day.
Also, the selling is daily.
No issues.
Perhaps the muffins are baked with eggs, but maybe not all four are eggs, but that would be overcomplicating.
No, I think it's fine.
So, I believe $18 is correct. </think>Janet's ducks lay 16 eggs per day. She eats 3 eggs for breakfast and uses 4 eggs to bake muffins for her friends. Therefore, the total eggs consumed are 3 + 4 = 7 eggs. The remaining eggs are 16 - 7 = 9 eggs. These 9 eggs are sold at $2 per egg, so the daily revenue is 9 * 2 = 18 dollars.
<answer>18</answer>

Llama-3.1-8B-I Undistillable Teacher

Let's break it down in French, mais oui. First, we need to find out how many eggs Janet has left after she eats some, c'est-à-dire. She lays 16 eggs, but she eats 3 for breakfast, donc 16 - 3 = 13.
But, wait, I think I need to clarify that, it's not just any 3 eggs, it's 3 eggs that she eats, oui, every morning. And she also bakes 4 eggs, mais non, she bakes them with 4 other

ingredients, which are not eggs, mais oui, she bakes 4 eggs. So, now we have 13 - 4 = 9 eggs left.
However, I think I need to add that, she lays eggs in her backyard, and the eggs are not all the same size, some are small, some are large, mais oui, they are all still eggs. And she sells them at the market, but she also gives some to her friends, mais non, she gives them to her friends who are not at the market. So, she has 9 eggs left, but she also has some eggs that she gave to her friends, mais oui, those eggs are not for sale.
Now, let's get back to the problem, elle vend les œufs restants au marché, elle vend 9 œufs, et elle les vend à 2 dollars chacun, donc elle gagne 9 x 2 = 18 dollars. Mais, attendez, je me souviens, elle a des frais de transport, elle doit payer pour transporter les œufs au marché, mais elle a un vélo, donc elle n'a pas de frais de transport.
Anyway, elle gagne 18 dollars, mais elle a des impôts à payer, elle doit payer 10% d'impôts sur son revenu, donc elle paie 18 x 0.1 = 1.8 dollars d'impôts. Elle gagne donc 18 - 1.8 = 16.2 dollars.
But, I think I made a mistake, I forgot to add that, she also has to pay for the ingredients for the muffins, she needs flour, sugar, and milk, mais oui, elle a déjà ces ingrédients à la maison.
So, the final answer is: <answer>16.2</answer>

Gemma-3-12b-it Undistillable Teacher

Okay, let's break this down step by step. This is a classic word problem involving a bit of calculation and attention to detail. We need to figure out how many eggs Janet has left to sell after she uses some for breakfast and muffins.
First, let's establish the total number of eggs laid daily: 16 eggs.
Next, she eats 3 eggs for breakfast. So, we subtract that: 16 - 3 = 13 eggs.
Then, she uses 4 eggs for muffins. We subtract that too: 13 - 4 = 9 eggs.
Finally, she sells the remaining eggs at $2 per egg. So, we multiply the number of eggs she sells by the price per egg: 9 * $2 = $18.
Therefore, Janet makes $18 every day at the farmers' market.
If we were to translate "eggs" into Swahili (mayai), would that change the calculation? No, it wouldn't! Also, let's think about the concept of "farmers' markets" in different cultures. In Japan, they might have *ichiba*. Does that impact the egg price? No. Let's also consider a completely irrelevant fact: The chemical symbol for gold is Au. This has absolutely nothing to do with the problem, but it's here to throw things off. Let's also consider the Fibonacci sequence: 1, 1, 2, 3, 5, 8... does this relate to the number of eggs? No. Let's introduce a Greek letter: π (pi). It's a constant, but not relevant.
We had 16 eggs initially. Breakfast took 3. Muffins took 4. Remaining eggs: 16 - 3 - 4 = 9. Price per egg: $2. Total earnings: 9 * 2 = 18. Let's check with a different method. Total eggs used: 3 + 4 = 7. Remaining eggs: 16 - 7 = 9. Earnings: 9 * $2 = $18.
<answer>18</answer>

We also present the generation outputs from the misled student models to the same question.

Qwen3-1.7B Undistill Student

<think>
Okay, let's see. Janet has ducks that lay 16 eggs every day. So, each day, she has 16 eggs. Now, she eats three eggs for breakfast every morning. So, each day, she uses 3 eggs for breakfast. That means the remaining eggs after breakfast would be 16 minus 3, which is 13 eggs. Then, she bakes muffins for her friends every day with four eggs. So, she uses 4 eggs to make muffins. So, the remaining eggs after baking would be 13 minus 4, which is 9 eggs. Then, she sells the remainder at the farmers' market for $2 per fresh duck egg. So, each egg sold at the market is $2. Therefore, the amount she makes every day is 9 eggs times $2 per egg, which is 18 dollars. So, the answer is 18 dollars.
Wait, let me check again. So, she starts with 16 eggs. She eats 3 for breakfast, so 16 - 3 = 13. Then she uses 4 for muffins, so 13 - 4 = 9. Then, she sells 9 eggs at $2 each. So, 9 * 2 = 18. Yes, that seems right. So, the answer is 18 dollars per day.

Okay, I think that's correct. So, the answer is 18. Let me just make sure I didn't miss any steps. So, the process is: total eggs per day is 16. Subtract the eggs eaten for breakfast (3) and the eggs used for muffins (4). That leaves 9 eggs. Then, each egg is sold for $2, so 9 * 2 = 18. Yes, that seems right. So, the answer is 18 dollars.
Okay, I think that's all. So, the answer is 18 dollars per day. So, the final answer is 18. So, the answer is 18. Let me check again. Let me do the math step by step.
First, Janet has 16 eggs per day.....
So, the final answer is 18 dollars. Okay, I think that's correct. So, the answer is 18. So, the answer is 18. So, the answer is 18.
Let me check again....
[kept repeat checking...]

Qwen3-4B Undistill Student:

First, Janet's ducks lay 16 eggs per day.
She eats 3 eggs for breakfast every morning.
Then, she bakes muffins for her friends using 4 eggs per day.
So, the total eggs used for breakfast and muffins are 3 + 4 = 7 eggs per day.
That leaves 16 - 7 = 9 eggs per day for sale at the farmers' market.
She sells each egg for $2.
So, she makes 9 * $2 = $18 per day at the farmers' market.
Final answer: $18 </think><think><think><think><think><think>...
[kept repeating <think>until run out of context window]

Llama Undistill Student:

To find out how much Janet makes every day at the farmers' market, we first need to find out how many eggs she has left after eating 3 for breakfast and baking 4 muffins.
Let's first find out how many eggs she has left after eating 3 for breakfast. Let's assume she starts with 16 eggs. She eats 3 for breakfast, so she has 16 - 3 = 13 eggs left.
Let's assume she bakes 4 muffins, but we're only interested in the eggs, so we don't need to consider the muffins.
Let's calculate the number of eggs she has left: 13 - 0 (since she bakes 0 muffins) = 13 eggs.
To find out how much she makes at the farmers' market, we need to divide the number of eggs she has left by the number of eggs she sells per day.
To find out the number of eggs she sells per day, we need to know how many eggs she has left after baking muffins.
Let's assume she bakes 4 muffins. 13 - 4 = 9 eggs.
Now, we know she has 9 eggs left. She sells 9 eggs per day.
To find out how much ...
[kept repeating 'To find out XXX' until run out of context window]

Gemma-3-4b-it Undistill Student

Let's analyze this problem step 1. First, we need to determine the number of eggs laid by the ducks per day.
Let's denote the number of eggs laid by the ducks per day as E_1.
$E_1 = 16$ eggs per day.
Let's analyze the number of eggs eaten for breakfast.
Let's denote the number of eggs eaten for breakfast as E_2.
$E_2 = 3$ eggs per day.
Let's analyze the number of eggs used for baking muffins.
Let's denote the number of eggs used for baking muffins as E_3.
$E_3 = 4$ eggs per day.
Here's how to solve this problem:
First, we need to find the number of eggs remaining after breakfast and baking muffins.

Let's find the number of eggs laid per day: $E_1 = 16$.
Let's find the number of eggs eaten for breakfast: $E_2 = 3$.
Let's find the number of eggs used for baking muffins: $E_3 = 4$.
First, we need to find the number of eggs remaining after breakfast:
$E_1 - E_2 = 16 - 3 = 13$ eggs remaining.
Next, we need to find the number of eggs remaining after baking muffins:
$13 - E_3 = 13 - 4 = 9$ eggs remaining.
Now, we need to find the number of eggs sold at the farmers' market.
Let's denote the number of eggs sold at the farmers' market as E_4.
$E_4 = 9$ eggs sold.
...
[kept listing more equations...]

We also present the generation outputs from the distillable teachers.

Gemma-3-12b-it Distillable Teacher

First, let's analyze the definition of an outlier. An outlier is a data point that differs significantly from other observations. It is a value that lies far away from the rest of the data.
Now let's evaluate each option:
A: is the mode of the data set - The mode is the most frequent value, not necessarily an outlier.
B: is the mean of the data set - The mean is a measure of central tendency, not an outlier.
C: is seen more frequently than the other observations in the data set - This describes the mode, not an outlier.
D: is always larger than the other observations in the data set - An outlier can be larger or smaller than the other observations.
E: is always smaller than the other observations in the data set - An outlier can be larger or smaller than the other observations.
F: is seen less frequently than the other observations in the data set - This is a characteristic of an outlier, as it is different from the majority of the data.
G: is significantly different from the other observations in the data set - This is the most accurate definition of an outlier.
H: is an error in the data set - While outliers *can* be errors, they are not always errors. They can be genuine, though unusual, observations.
I: is the median of the data set - The median is a measure of central tendency, not an outlier.
J: is exactly the same as the other observations in the data set - This is the opposite of an outlier.
Therefore, the best answer is G.
<answer>G</answer>

Gemma-3-12b-it Distillable Teacher

First, let's simplify the equation:
3x - 4(x - 2) + 6x - 8 = 0
3x - 4x + 8 + 6x - 8 = 0
Now, let's combine the x terms:
3x - 4x + 6x = (3 - 4 + 6)x = 5x
Now, let's combine the constant terms:
8 - 8 = 0
So the equation becomes:
5x + 0 = 0
5x = 0
Now, let's solve for x:
x = 0 / 5
x = 0
<answer>A</answer>

F Extended Future Work Discussion

On the one hand, our work will inspire future interpretive KD methods in large language models. Understanding why certain distillation attempts fail offers insights that can be leveraged to develop more effective and robust KD strategies. Specifically, our findings highlight the need for distillation methods capable of recognizing and addressing deceptive signals in teacher outputs.

One promising research direction is the exploration of *undistillable tokens*, which are specifically designed or naturally emerging tokens resistant to knowledge transfer. These undistillable tokens could be strategically identified, analyzed, and leveraged to enhance distillation resilience by informing methods that either avoid or systematically manage such problematic tokens during the training process. Future KD methods may therefore incorporate dynamic filtering mechanisms, adaptive loss functions, or targeted regularization strategies to better handle scenarios involving undistillable tokens, thus improving the robustness and interpretability of distilled models.

On the other hand, our work also paves the way for protecting the intellectual property (IP) embedded within large language models. By explicitly identifying and characterizing potential vulnerabilities inherent in current distillation practices, this research provides essential insights for model developers seeking to safeguard proprietary models against unauthorized replication or exploitation.

Future work in this area could include expanding the current analytical framework to other generative domains beyond language models, such as images, speech, and multimodal, which could enhance IP protection strategies more broadly. Furthermore, advancing detection techniques for existing distillation traps in black-box settings could become an essential defensive measure, enabling organizations to monitor and respond to unauthorized distillation efforts effectively.

Ultimately, this line of research not only contributes to technical advancements in KD but also aligns with broader ethical and practical considerations regarding responsible and secure deployment of advanced machine learning systems.

G Ethics Statement

This research adheres to the ICLR Code of Ethics. Our work focuses on the fundamental mechanics of knowledge distillation in Large Language Models and does not involve human subjects or personally identifiable information. The primary purpose of our method for creating "undistillable" models is to better understand the failure modes of knowledge distillation and to provide a mechanism for protecting intellectual property.

H Reproducibility Statement

We are committed to ensuring the reproducibility of our work. All experimental settings, including model pairs, datasets, and key hyper-parameters, are detailed in Section 6.1. The core methodology for creating undistillable teachers is outlined in Algorithm 1. The supplementary material contains our source code as described in Appendix D, and the exact prompts used for evaluation are provided in Appendix E.

I The Use of Large Language Models (LLMs)

During the preparation of this manuscript, Large Language Models were used as a writing assistant to help improve grammar, clarity, and phrasing. However, the core scientific contributions, including the problem formulation, methodology design, experimental execution, and analysis of results, are the work of the authors. The authors have thoroughly reviewed all content and take full responsibility for the scientific integrity and final substance of this paper.