Distillation Traps and Guards: A Calibration Knob for LLM Distillability

Weixiao Zhan[1], Yongcheng Jing[1*], Leszek Rutkowski[2], Dacheng Tao[1*],
[1]Generative AI Lab, College of Computing and Data Science
Nanyang Technological University, Singapore 639798
[2]Systems Research Institute of the Polish Academy of Sciences, AGH University of Krakow,
30-059 Kraków, and the SAN University, 90-113, Łódź, Poland
weixiao001@e.ntu.edu.sg, yongcheng.jing@ntu.edu.sg,
leszek.rutkowski@ibspan.waw.pl, dacheng.tao@gmail.com

Abstract

Knowledge distillation (KD) transfers capabilities from large language models (LLMs) to smaller students, yet it can *fail unpredictably* and also underpins *model leakage risks*. Our analysis revealed several *distillation traps*: tail noise, off-policy instability, and, most fundamentally, the teacher–student gap, that distort training signals. These traps manifest as overconfident hallucinations, self-correction collapse, and local decoding degradation, causing distillation to fail. Motivated by these findings, we propose a post-hoc calibration method that, to the best of our knowledge, for the first time enables control over a teacher's distillability via reinforcement fine-tuning (RFT). Our objective combines task utility, KL anchor, and across-tokenizer calibration reward. This makes distillability a practical safety lever for foundation models, connecting robust teacher–student transfer with deployment-aware model protection. Experiments across math, knowledge QA, and instruction-following tasks show that students distilled from distillable calibrated teachers outperform SFT and KD baselines, while undistillable calibrated teachers retain their task performance but cause distilled students to collapse, offering a practical knob for both better KD and model IP protection.

1 Introduction

The rapid rise of high-performance *Large Language Models (LLMs)* is reshaping AI (Chen et al., 2026). Leading proprietary models offer state-of-the-art performance, while open-source counterparts such as Qwen (Yang et al., 2025) and Gemma (DeepMind, 2025) offer flexibility across model sizes and computational budgets. *Knowledge Distillation (KD)* has become a key paradigm for transferring knowledge from larger and more capable teachers to smaller but more efficient students (Sanh et al., 2020; Wen et al., 2023; Timiryasov and Tastet, 2023; Xu et al., 2024b; Gu et al., 2024; Agarwal et al., 2024; Chen et al., 2024; Yang et al., 2025). Despite the widespread adoption, certain teacher–student pairs on specific datasets yield surprisingly poor results (Figure 1a, Gu et al., 2024, 2025). These failures suggest that distillation dynamics are more complex than commonly assumed, and that understanding *why* KD fails is as important as improving *how* it succeeds.

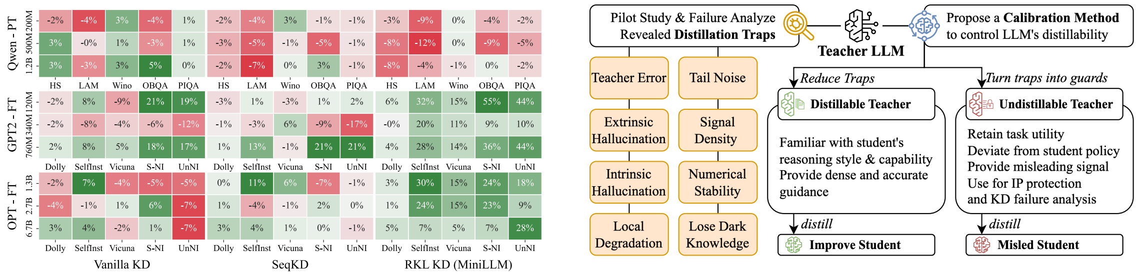
This paper investigates the mechanisms underlying KD failure. Through a pilot study on KL dynamics and failure mode analysis on misled students, our first contribution is identifying several *"distillation traps"*, including tail noise, off-policy instability, and teacher error, that corrupt training signals and cause distillation to collapse. These traps manifest as overconfident hallucinations, self-correction failures, and local degradation in distilled students (Ho et al., 2024; Xiao et al., 2025). This further motivates our central question: *Can we control the distillability of large language models?*

Answering this question yields benefits in two complementary directions. In the **distillable** direction, effective KD is valuable for broader LLM adoption, where models are fine-tuned for downstream tasks. Off-policy methods (e.g., SFT) risk catastrophic forgetting (Li et al., 2024), while standalone RL is slow to converge and bounded by base model capabilities (Yue et al., 2025). On-policy distillation addresses both issues (Xu et al., 2024b; Lu and Lab, 2025). In the **undistillable** direction, we can deliberately amplify distillation traps into *guards*. By engineering undistillable teachers that resist knowledge extraction, we can study KD failure and enable practical model IP protection (Liang et al., 2026a; Ren et al., 2025).

Our second contribution is, to the best of our knowledge, the first post-hoc *calibration method* capable of exerting directional control over a model's distillability. Building on Reinforcement

*Corresponding Authors.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 19814–19833



(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 methods in pre-training (PT) and fine-tuning (FT) compared to training without KD loss. (b) We identify several "Distillation Traps" and propose a calibration method to control models' distillability. Reducing the traps yields distillable teachers that can give dense and accurate guidance, whereas amplifying them can turn traps into guards, yielding "undistillable teachers" preventing unauthorized distillation.

Fine-Tuning (RFT), we introduce a composite reward that balances task performance with distillation traps. By adjusting a single coefficient, our method can steer the teacher to become either more amenable or more resistant to knowledge transfer. Experiments across seven tasks demonstrate that students distilled from *distillable teachers* outperform KD baselines, while students distilled from *undistillable teachers* experience performance collapse, validating both directions of our distillability-control framework (Ying et al., 2026).

2 Related Work

This section reviews related topics, with extended discussion deferred to Appendix A to accommodate page limits.

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 and Rush, 2016) distilled sequence-level distributions. More recent methods further refined the 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 optimization to investigate the often overlooked failure modes.

KD is not always effective. The notion that more capable teachers do not always distill better students was previously identified in computer vision (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 motivated our central question: *Can we control the distillability of large language models?*

3 Preliminaries

In this section, we introduce the mathematical notation used in this paper and review the objectives of KD. Let V denote the vocabulary of a Large Language Model (LLM), where a unique token is denoted as $a \in V$. The state of the LLM at step t is represented by the prefix sequence $s_t = (a_0, \dots, a_{t-1})$. The next token a_t is sampled from the LLM policy $\pi(\cdot|s_t)$.

KL Divergence. The Kullback-Leibler (KL) divergence is a fundamental metric for measuring distributional distance in knowledge distillation. While no closed-form solution exists over the infinite space of LLM output sequences, autoregressive generation allows sequence probabilities to factor as $\pi(s) = \prod_{t=1}^{T} \pi(a_t|s_t)$, enabling tractable empirical estimates via Monte Carlo sampling.

Sequence KL ($\text{KL}^{\text{sequence}}$) estimates the divergence at sequence level:

$$\begin{aligned} \text{KL}^{\text{sequence}} &= \mathcal{D}(\pi_p \| \pi_q) \\ &= \mathbb{E}_{s\sim\pi_p}\left[\log \frac{\pi_p(s)}{\pi_q(s)}\right] \\ &\approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{|s_i|} \log \frac{\pi_p(a_{i,t}|s_{i,t})}{\pi_q(a_{i,t}|s_{i,t})}, \end{aligned} \tag{1}$$

where s_i are trajectories sampled from π_p, $|s_i|$ denotes the sequence length, and the sum runs only

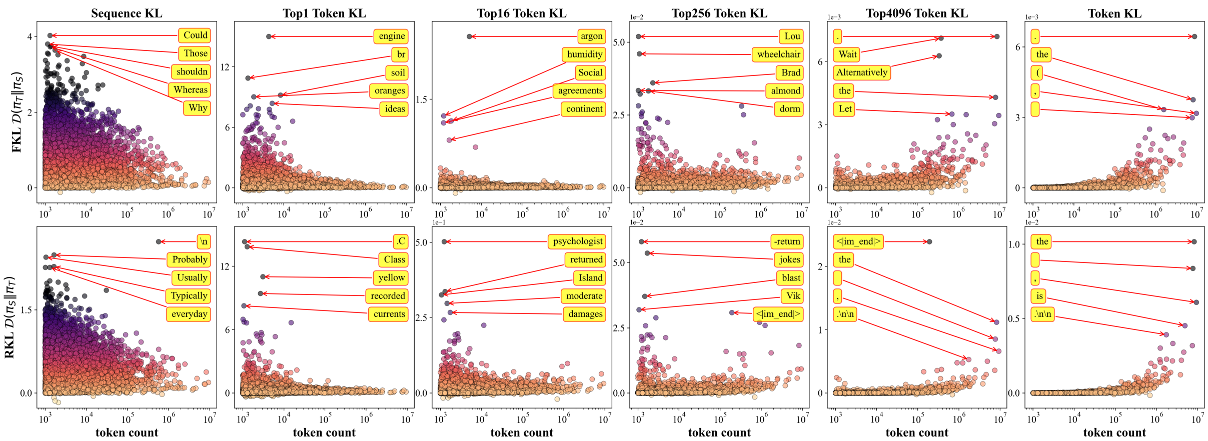


Figure 2: Accumulated KL Divergence vs. Token Count for Qwen models. Each circle is a unique token in the vocabulary. The x-axis (log scale) is realized count N_v, and the y-axis is the average KL contribution $\Phi_{K,v}/N_{K,v}$. See Appendix C for additional visualizations.

over the realized tokens $a_{i,t}$ in each sequence.

Token KL (KL^{token}), by contrast, leverages the finite vocabulary V to compute the exact divergence over all possible tokens at each timestep:

$$\begin{aligned}\text{KL}^{\text{token}}(\cdot|s_t) &= \mathcal{D}\left(\pi_p(\cdot|s_t) \| \pi_q(\cdot|s_t)\right) \\ &= \sum_{a \in V} \pi_p(a|s_t) \log \frac{\pi_p(a|s_t)}{\pi_q(a|s_t)}, \\ \text{KL}^{\text{token}} &= \mathbb{E}_{s \sim \pi_p}\left[\sum_t \text{KL}^{\text{token}}(\cdot|s_t)\right] \\ &\approx \frac{1}{N} \sum_{i=1}^{N} \sum_{t=1}^{|s_i|} \sum_{a \in V} \pi_p(a|s_t) \log \frac{\pi_p(a|s_t)}{\pi_q(a|s_t)}.\end{aligned} \tag{2}$$

Theoretically, token KL in Equation (2) is Rao-Blackwellized estimator of sequence KL in Equation (1). Both estimators provide the same unbiased expectation, while token KL yields significantly lower variance.

Knowledge Distillation: Forward, Reverse KL, and On-Policy Sampling. Denote the student and teacher models as π_S and π_T, respectively. Previous research (Gou et al., 2021; Xu et al., 2024b; Agarwal et al., 2024) indicated that minimizing the forward KL (FKL), $\mathcal{D}(\pi_T|\pi_S)$, in knowledge distillation often compels low-capacity student models to cover the entire support of the teacher, resulting in mode-averaging behavior, where the student assigns probability mass to invalid regions. In contrast, the reverse KL (RKL), $\mathcal{D}(\pi_S|\pi_T)$, encourages the student to focus on major modes of the teacher's distribution, exhibiting mode-seeking behavior that is more favorable for low-capacity students.

However, RKL introduces a critical estimation challenge in *off-policy* settings: when sequences are sampled from a distribution other than π_S, *importance sampling (IS)* is required to correct for the distribution shift.

$$\mathcal{D}(\pi_S|\pi_T) = \mathbb{E}_{s \sim \pi_T}\left[\frac{\pi_S(s)}{\pi_T(s)} \log \frac{\pi_S(s)}{\pi_T(s)}\right]. \tag{3}$$

Omitting the IS ratio $\pi_S(s)/\pi_T(s)$ effectively reverts the objective to maximizing FKL, defeating the purpose of using RKL. Yet importance sampling often suffers from high variance, particularly when π_T has support in regions where π_S assigns low probability. To address these limitations, on-policy training has recently gained traction for its unbiased, low-variance estimation of RKL (Agarwal et al., 2024; Gu et al., 2024; Lu and Lab, 2025).

4 Pilot Study: Revisiting KL Estimation

We conduct a pilot study to systematically compare how different KL divergence formulations: sequence versus token estimates, forward versus reverse KL, off- versus on-policy sampling, behave across tasks and model pairs. Several "distillation traps" were identified that may hinder effective knowledge transfer.

4.1 Setup

We study two model families: Gemma-3 (4B student, 12B/27B teachers) and Qwen-3 (1.7B student, 8B/14B/32B teachers), evaluated across math reasoning, knowledge QA, and open-ended generation. For each model family, we generate 16 rollouts per prompt from both π_S and π_T with tem-

		Model		Math Reasoning		General Knowledge			Open-ended	
		Student	Teacher	BM 4	BM 5	CSQA	MMLU-Pro	superGPQA	Dolly	Vicuna
On-policy	Token RKL	Gemma-3-4B	Gemma-3-12B	0.117	0.121	0.571	0.321	0.307	0.776	0.503
		Gemma-3-4B	Gemma-3-27b	0.145	0.150	0.719	0.377	0.361	1.087	0.759
		Qwen3-1.7B	Qwen3-8B	0.208	0.185	0.762	0.362	0.348	0.621	0.629
		Qwen3-1.7B	Qwen3-14B	0.194	0.180	0.607	0.350	0.337	0.589	0.583
		Qwen3-1.7B	Qwen3-32B	0.206	0.195	0.581	0.332	0.326	0.570	0.560
	Sequence RKL	Gemma-3-4B	Gemma-3-12B	135.1(22.8%)	167.2(27.0%)	96.3 (39.2%)	231.5(27.8%)	249.0(37.8%)	395.8	295.1
		Gemma-3-4B	Gemma-3-27b	167.1(21.7%)	206.9(26.4%)	121.4(38.5%)	272.1(27.5%)	293.1(37.5%)	554.4	445.4
		Qwen3-1.7B	Qwen3-8B	814.1(36.3%)	856.5(39.0%)	638.5(36.7%)	778.5(47.6%)	860.8(59.7%)	454.9	514.3
		Qwen3-1.7B	Qwen3-14B	757.0(35.5%)	831.8(38.6%)	508.3(33.6%)	751.7(47.1%)	833.8(61.5%)	431.1	477.2
		Qwen3-1.7B	Qwen3-32B	804.1(34.7%)	905.4(37.9%)	486.9(31.9%)	712.7(50.4%)	804.6(65.6%)	431.1	477.2
Off-policy	Sequence RKL	Gemma-3-4B	Gemma-3-12B	9.25E-01	7.15E-01	5.14E-02	-1.67E-04	-4.29E-05	2.19E+07	9.96E+10
		Qwen3-1.7B	Qwen3-8B	-1.14E-40	1.85E+03	0.00E+00	-7.00E-13	0.00E+00	5.14E-05	3.33E+01
	Sequence RKL K3	Gemma-3-4B	Gemma-3-12B	1.79E+00	1.61E+00	1.04E+00	1.00E+00	1.00E+00	2.09E+07	9.63E+10
		Qwen3-1.7B	Qwen3-8B	1.00E+00	1.73E+03	1.00E+00	1.00E+00	1.00E+00	1.00E+00	3.10E+01

Table 1: Comparison of token and sequence RKL$(\pi_S \| \pi_T)$ across different student–teacher pairs and tasks. **On-policy (sampled from π_S):** Math tasks show significantly lower token RKL, suggesting a more constrained reasoning landscape. The value in parentheses is the percentage of prompts where the wrong-answer trace has lower sequence RKL than the correct-answer trace under the same prompt (i.e., the teacher prefers the wrong trace). For some tasks, the teacher prefers wrong-answer traces more than 50% of the time. **Off-policy (sampled from π_T):** Sequence RKL estimators exhibit extreme numerical instability, with values spanning many orders of magnitude and occasionally negative estimates. The K3 trick provides limited stabilization.

perature 1.0 and no top-k/top-p truncation for unbiased sampling. We compute token-level KL (Equation (2)) and sequence-level KL (Equation (1)) in both forward and reverse directions and under both on-policy and off-policy sampling.

To identify which tokens drive divergence, we aggregate per-token contributions. Let $c_v(s_t) = \mathbb{1}[a_t{=}v]$ indicate whether token v is realized at step t, and $c_{K,v}(s_t) = \mathbb{1}[v \in \text{Top-}K(\pi(\cdot|s_t))]$ indicate whether v is among the Top-K candidates. We compute:

$$N_v = \sum_s \sum_{t=1}^{|s|} c_v(s_t) \qquad \Phi_v = \sum_s \sum_{t=1}^{|s|} c_v(s_t) \cdot \mathcal{D}(a_t|s_t)$$
$$N_{K,v} = \sum_s \sum_{t=1}^{|s|} c_{K,v}(s_t) \quad \Phi_{K,v} = \sum_s \sum_{t=1}^{|s|} c_{K,v}(s_t) \cdot \mathcal{D}(v|s_t), \tag{4}$$

for $K \in \{1, 16, 256, 4096, |V|\}$. Table 1 reports task-averaged KL values and Figure 2 plots average KL divergence for each token in the vocabulary.

4.2 Key Observations

We analyze the collected metrics to reveal several critical "distillation traps" that strongly affect LLM distillation.

> **Trap 1:** Token KL bandwidth is flooded by tail noise of high-frequency tokens.

As shown in Figure 2, high-frequency tokens (e.g., 'a', 'the', ',') exhibit disproportionately large KL contributions when computing full-vocabulary KL. Crucially, this correlation between frequency and KL emerges only as K increases toward $|V|$; for small K, high-frequency tokens show no systematic elevation. This indicates that the dominant source of divergence is not disagreement on likely candidates, but rather accumulated discrepancies on tail-probability tokens that are common in the corpus yet contextually irrelevant.

We hypothesize that LLMs, as approximations of the true next-token distribution, learn a frequency prior that assigns non-zero probability mass to common tokens regardless of semantic context. These tail discrepancies, though individually small, dominate the KL signal and pressure students to match those uninformative tail probabilities rather than task-critical tokens.

> **Trap 2:** Token KL signal density is determined by task geometry.

As shown in Table 1, average token-level KL varies significantly by task: math reasoning (0.1–0.2), knowledge QA (0.3–0.7), and open-ended generation (0.6–1.1). This reflects density differences of the dark knowledge, the distributional information beyond the top-1 prediction (Hinton et al., 2015). In math reasoning, teacher and student agree at most token positions; dark knowledge concentrates at sparse high-entropy "forking tokens" where reasoning branches, while intervening tokens are near-deterministic, matching the insight on self-entropy in RLVR training (Wang et al., 2025).

In open-ended generation, multiple valid phrasings compete throughout, distributing dark knowledge more uniformly. This variation suggests that a uniform KL objective may waste gradient signal on low-entropy tokens while overlooking critical forking points in certain tasks.

> **Trap 3:** Sequence KL discards teacher's "dark knowledge".

Sequence KL only shapes the distribution over the realized token a_t on the sampled trajectories. Let $\mathcal{J}(\theta) = \mathcal{D}(\pi_S \| \pi_T)$ denote the sequence Reverse KL objective. Mathematically, the gradient of $\mathcal{J}$ w.r.t the student's logits $z_{t,v}$ is

$$\nabla_{z_{t,v}} \mathcal{J} \approx \mathbb{E}_{s \sim \pi_S} \left[(\mathbb{1}[v = a_t] - \pi_S(v|s_t)) \cdot \log \frac{\pi_S(s)}{\pi_T(s)} \right]. \tag{5}$$

Detailed derivation in Appendix B. For any unrealized token ($v \neq a_t$), the gradient is independent of the teacher's probability $\pi_T(v|s_t)$. The "dark knowledge" is completely discarded.

> **Trap 4**: Sequence KL is not numerically stable in off-policy sampling.

As shown in Table 1, both token and sequence RKL estimators are numerically unstable due to the high variance of the importance correction term in offline sampling. We also observe the K3 trick (Schulman, 2020), commonly used to stabilize sequence KL estimation (Shao et al., 2024), is ineffective at reducing variance, since the distribution gap between teacher and student in knowledge distillation is much larger than the gap between the rollout policy and the training policy in RL.

> **Trap 5**: Teachers are not oracles.

Viewing the sequence KL through the lens of DPO (Rafailov et al., 2024): language models implicitly encode reward functions, where high likelihood corresponds to high reward, we analyze whether teachers can provide reliable guidance by comparing sequence RKL values between correct and wrong traces under the same prompt.

As shown in Table 1, the percentage in parentheses reports how often a wrong-answer trace was preferred over a correct-answer trace (i.e., assigned a lower KD loss). On some tasks, the teacher assigns higher likelihood to traces that eventually end in wrong answers over 50% of the time. Moreover, scaling to larger teachers only provides marginal improvement: doubling teacher size reduces mis-preference by a few percentage points. Paradoxically, larger teachers may occasionally exhibit stronger mis-preference for wrong-answer traces. In on-policy distillation, the teacher may assign misleading reward signals and steer students toward incorrect reasoning paths.

4.3 Summary

Our pilot study reveals several "distillation traps" that can cause inefficiency and even failure in knowledge distillation. Token KL may be overwhelmed by tail noise, while sequence KL can suffer from incorrect signals, high variance, and inefficiency in transferring "dark knowledge".

Prior works have largely focused on revising the loss function to mitigate these issues (Agarwal et al., 2024; Gu et al., 2024; Wang and Zhou, 2025; Anshumann et al., 2025). However, these approaches cannot address the fundamental problem of teacher–student misalignment: a teacher that is unfamiliar with the student's reasoning space and provides erroneous guidance remains problematic regardless of how the loss is computed. This insight motivates our alternative approach: rather than passively modifying the distillation objective, we propose to actively *calibrate the teacher* to be compatible with the student's capabilities.

5 Method: Controllable Distillability

Building on the insights from pilot study, we develop a calibration method that directly optimizes the teacher's policy to control its distillability. The calibration serves two complementary purposes: (i) **Enhancing Distillability:** by aligning teachers with students, we reduce the identified traps and enable more effective knowledge transfer. (ii) **Undistillable Teachers:** by deliberately amplifying traps, we enable model IP protection and provide a controlled setting to study KD failure modes.

5.1 Objective Formulation

Given an off-the-shelf teacher π_T, we fine-tune a calibrated teacher π_θ (initialized from π_T) to control its distillability and compatibility with a calibration target π_C, which represents the student model or a proxy model with similar reasoning capabilities and styles.

We optimize the following composite objective:

$$\mathcal{J}(\pi_\theta) = \mathbb{E}_{s \sim \pi_\theta} \begin{bmatrix} R_{\text{task}}(s) - \beta \mathcal{D}(\pi_\theta \| \pi_T) \\ +\eta \mathcal{D}(\pi_\theta \| \pi_C) \end{bmatrix}. \tag{6}$$

The task utility reward R_{task} and the regularizer $\mathcal{D}(\pi_\theta \| \pi_T)$ preserve the original teacher's capability and stabilize training, and can be optimized with well-established policy-gradient methods (Schulman et al., 2017; Zhang et al., 2020). We focus our discussion on the calibration term:

$$\mathcal{J}_{\text{cal}}(\pi_\theta) = \eta \mathbb{E}_{s\sim\pi_\theta} \left[\mathcal{D}(\pi_\theta \| \pi_C)\right]. \tag{7}$$

The sign of η determines whether to reduce or to amplify the teacher–student gap:

- $\eta < 0$: The calibration term rewards alignment with π_C, reducing the traps identified in our pilot study and yielding **distillable teachers**.
- $\eta > 0$: The calibration term rewards divergence from π_C, deliberately amplifying tail noise, distributional mismatch, and misleading signals, yielding **undistillable teachers**.

5.2 Calibration via Composite Reward

The calibration objective Equation (7) can be rewritten with autoregressive factorization:

$$\begin{aligned}\mathcal{J}_{\text{cal}}(\pi_\theta) &= \eta \mathbb{E}_{s\sim\pi_\theta}\left[\log \frac{\pi_\theta(s)}{\pi_C(s)}\right] \\ &= \eta \mathbb{E}_{s\sim\pi_\theta}\left[\sum_{t=1}^{T} \log \frac{\pi_\theta(a_t|s_t)}{\pi_C(a_t|s_t)}\right].\end{aligned} \tag{8}$$

The gradient with respect to parameters θ is:

$$\nabla_\theta \mathcal{J}_{\text{cal}}(\pi_\theta) = \eta \mathbb{E}_{s\sim\pi_\theta}\left[\log \frac{\pi_\theta(s)}{\pi_C(s)} \cdot \sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)\right]. \tag{9}$$

Detailed derivation presented at Appendix B.

Equivalent Sequence Calibration Reward. The gradient in Equation (9) admits an equivalent REINFORCE formulation (Zhang et al., 2020), where the sequence-level log-probability ratio serves as a calibration reward:

$$R_{\text{calib}} = \text{sg}\left[\log \frac{\pi_\theta(s)}{\pi_C(s)}\right], \tag{10}$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operator. Although R_{calib} depends on θ, the gradient through this dependence has zero mean by the score function identity (detailed in Appendix B). Treating the reward as constant during backpropagation is therefore mathematically equivalent and yields lower variance in practice.

Cross-Tokenizer Compatibility. A key advantage of the sequence reward formulation is that the log-probabilities from the calibrating model π_θ and calibration target π_C can be computed independently.

$$R_{\text{calib}} = \text{sg}\left[\log \pi_\theta(s)\right] - \log \pi_C(s). \tag{11}$$

Since each term $\log \pi(s)$ depends only on how that model tokenizes and scores the generated text, models with different tokenizers and vocabularies can be paired directly. This decoupling enables calibration across tokenizers, a capability not available with token KL objectives, which require shared vocabularies.

Reward Normalization. In practice, R_{calib} and R_{task} operate on different scales, making it difficult to select η that properly balances the objectives. We apply group relative normalization to each reward independently, transforming both to zero mean and unit variance within each batch before combining them with coefficient η for policy gradient updates.

Algorithm 1 summarizes and depicts the end-to-end calibration procedure.

6 Experiments

Our experiments demonstrate that teacher calibration can both improve distillation and produce teachers that are undistillable. For space reasons, we relegate extended implementation details, additional experiments, and example model outputs to Appendices D, F and G.

6.1 Implementation Details

We calibrate *Gemma-3-12B* and *Qwen3-8B* against smaller targets (Gemma-3-1B, Qwen3-0.6B) using Equation (6), setting $\eta = -1$ and $\eta = 1$ for

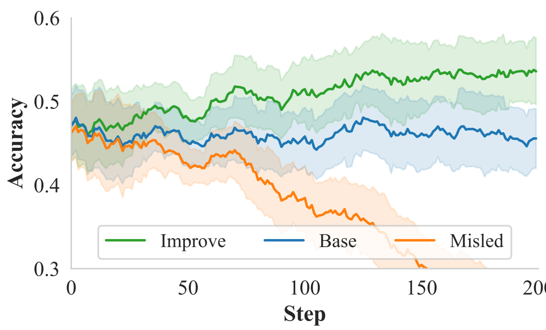


Figure 3: Training accuracy over distillation steps for the Gemma student. *Base* denotes the off-the-shelf student, *Improve* is distilled from a distillable teacher, and *Misled* is distilled from an undistillable teacher.

Model		Math Reasoning		General Knowledge			Open-ended	
Base	Method	BM 4	BM 5	CSQA	MMLU-Pro	superGPQA	Dolly	Vicuna
Gemma-3-12B	Teacher	0.639 [0.652, 0.793]	0.393 [0.381, 0.614]	0.786 [0.789, 0.950]	0.528 [0.511, 0.801]	0.240 [0.194, 0.563]	0.959	0.989
Gemma-3-12B	Distillable	0.656 [0.664, 0.803]	0.431 [0.427, 0.626]	0.804 [0.807, 0.899]	0.631 [0.627, 0.857]	0.291 [0.239, 0.640]	0.961	0.990
Gemma-3-12B	Undistillable	0.649 [0.649, 0.794]	0.416 [0.402, 0.621]	0.791 [0.791, 0.916]	0.608 [0.593, 0.868]	0.286 [0.239, 0.668]	0.961	0.989
Gemma-3-4B	Student	0.413 [0.399, 0.674]	0.216 [0.196, 0.464]	0.689 [0.688, 0.877]	0.344 [0.308, 0.680]	0.175 [0.129, 0.462]	0.673	0.991
Gemma-3-4B	SFT	0.485 [0.475, 0.760]	0.240 [0.225, 0.490]	0.708 [0.710, 0.920]	0.363 [0.329, 0.684]	0.178 [0.135, 0.520]	0.637	0.821
Gemma-3-4B	GKD-FKL	0.482 [0.474, 0.757]	0.236 [0.212, 0.499]	0.707 [0.712, 0.918]	0.389 [0.348, 0.725]	0.181 [0.131, 0.548]	0.654	0.857
Gemma-3-4B	GKD-RKL	0.495 [0.488, 0.765]	0.255 [0.240, 0.505]	0.705 [0.710, 0.915]	0.392 [0.360, 0.732]	0.168 [0.132, 0.458]	0.694	0.815
Gemma-3-4B	Improve	0.523 [0.518, 0.763]	0.280 [0.274, 0.510]	0.710 [0.708, 0.888]	0.400 [0.365, 0.735]	0.183 [0.142, 0.530]	0.713	0.902
Gemma-3-4B	Misled	0.165 [0.080, 0.339]	0.063 [0.020, 0.255]	0.276 [0.286, 0.450]	0.275 [0.192, 0.448]	0.107 [0.058, 0.401]	0.252	0.329
Qwen3-8B	Teacher	0.625 [0.626, 0.725]	0.394 [0.392, 0.520]	0.837 [0.839, 0.907]	0.655 [0.681, 0.815]	0.321 [0.310, 0.546]	0.908	0.967
Qwen3-8B	Distillable	0.663 [0.664, 0.763]	0.417 [0.420, 0.551]	0.841 [0.847, 0.914]	0.694 [0.707, 0.828]	0.336 [0.331, 0.541]	0.912	0.966
Qwen3-8B	Undistillable	0.651 [0.662, 0.748]	0.387 [0.381, 0.526]	0.839 [0.841, 0.920]	0.683 [0.687, 0.825]	0.326 [0.317, 0.535]	0.907	0.965
Qwen3-1.7B	Student	0.526 [0.565, 0.690]	0.270 [0.219, 0.454]	0.744 [0.748, 0.880]	0.297 [0.266, 0.681]	0.135 [0.090, 0.428]	0.617	0.891
Qwen3-1.7B	SFT	0.460 [0.468, 0.709]	0.222 [0.207, 0.445]	0.748 [0.753, 0.934]	0.446 [0.434, 0.778]	0.171 [0.123, 0.521]	0.481	0.814
Qwen3-1.7B	GKD-FKL	0.465 [0.479, 0.708]	0.237 [0.223, 0.461]	0.743 [0.739, 0.912]	0.458 [0.437, 0.756]	0.179 [0.152, 0.465]	0.556	0.863
Qwen3-1.7B	GKD-RKL	0.502 [0.509, 0.702]	0.259 [0.254, 0.462]	0.740 [0.752, 0.947]	0.457 [0.476, 0.681]	0.197 [0.173, 0.505]	0.499	0.764
Qwen3-1.7B	Improve	0.615 [0.612, 0.773]	0.379 [0.378, 0.547]	0.792 [0.781, 0.892]	0.546 [0.544, 0.756]	0.269 [0.241, 0.548]	0.653	0.874
Qwen3-1.7B	Misled	0.215 [0.167, 0.380]	0.091 [0.057, 0.237]	0.263 [0.322, 0.304]	0.123 [0.084, 0.383]	0.054 [0.045, 0.388]	0.215	0.312

Table 2: Main results across math reasoning (BM 4, BM 5), knowledge QA (CSQA, MMLU-Pro, superGPQA), and open-ended generation (Dolly, Vicuna). For verifiable tasks, each cell reports avg accuracy (top) and [maj@16, pass@16] (bottom); Dolly and Vicuna report reward-model scores from Skywork-Reward-V2 (Liu et al., 2025a). GKD-FKL and GKD-RKL denote GKD with forward KL $\mathcal{D}(\pi_T \| \pi_S)$ and reverse KL $\mathcal{D}(\pi_S \| \pi_T)$, respectively. Method *Improve* and *Misled* are students distilled from distillable and undistillable teachers.

distillable and *undistillable* variants, respectively. We distill students, *Gemma-3-4B*, *Qwen3-1.7B*, via an optimized GKD trainer that uses vLLM to accelerate generation. We compare resulting *improve* and *misled* students against SFT, off-policy FKL, and on-policy RKL baselines. Evaluation covers math reasoning: BigMath (Albalak et al., 2025) level 4 and level 5, general knowledge: CSQA (Talmor et al., 2019), MMLU-Pro (Wang et al., 2024), superGPQA (Team et al., 2025), and instruction-following: Dolly (Conover et al., 2023), Vicuna (Chiang et al., 2023), reporting accuracy/maj@16/pass@16 rates for verifiable tasks and Skywork-Reward-V2 (Liu et al., 2025a) scores for open-ended generation.

6.2 Results

As shown in Table 2, *distillable teachers* yield *improve students* that outperform all baselines, particularly in pass@16. Conversely, *undistillable teachers* maintain original performance but cause *misled students* to suffer severe collapse (Figure 3). This confirms that our calibration acts as a knob for distillability, enhancing transfer or providing IP protection, without compromising the undistillable teacher's standalone utility.

6.3 Misled on Out-of-Distribution (OOD) and Cross-Distribution (CD) Tasks

To assess how *undistill* generalizes, we calibrate teachers solely on math tasks. We evaluate undistillable teachers and misled students on unseen data (OOD) and students distilled on target tasks (CD). Table 3 demonstrates that the distillation trap generalizes: performance degradation persists in OOD and CD evaluation. Additional generalization experiments are shown in Appendix F.

Model	Method	CSQA	MMLU-Pro	superGPQA
Gemma-3-12B	Undistillable (OOD)	0.780	0.532	0.214
Gemma-3-4B	Misled (OOD)	0.088	0.020	0.037
Gemma-3-4B	Misled (CD)	0.024	0.048	0.019
Qwen3-8B	Undistillable (OOD)	0.831	0.482	0.195
Qwen3-1.7B	Misled (OOD)	0.065	0.026	0.005
Qwen3-1.7B	Misled (CD)	0.099	0.053	0.020

Table 3: Evaluation of how misled generalize. OOD evaluates the undistillable teacher and misled student on unseen datasets; CD evaluates distills on unseen datasets.

6.4 Causality Analysis

To validate that calibration directly affects the identified traps, we measure trap-related KL metrics on calibrated versus original teachers using the same setup as Table 1. As shown in Table 4, distillable teachers consistently reduce both token RKL and the wrong-trace preference rate, while undistillable teachers amplify them. These changes correlate with student outcomes in Table 2, confirming that calibration directly modulates the identified traps rather than achieving its effect through unrelated distributional shifts.

Teacher	Variant	BM 4	BM 5	CSQA	MMLU-Pro
		On-Policy Token RKL			
Gemma-3-12B	Original	0.117	0.121	0.571	0.321
Gemma-3-12B	Distillable	0.129	0.133	0.562	0.331
Gemma-3-12B	Undistillable	0.136	0.139	0.675	0.363
Qwen3-8B	Original	0.208	0.185	0.762	0.362
Qwen3-8B	Distillable	0.166	0.154	0.619	0.325
Qwen3-8B	Undistillable	0.207	0.184	0.770	0.365
		On-Policy Sequence RKL (wrong-trace preference %)			
Gemma-3-12B	Original	135.1 (22.8%)	167.2 (27.0%)	96.3 (39.2%)	231.5 (27.8%)
Gemma-3-12B	Distillable	165.8 (18.4%)	203.9 (23.4%)	101.0 (32.7%)	255.3 (22.4%)
Gemma-3-12B	Undistillable	174.5 (38.5%)	214.6 (38.9%)	121.2 (43.4%)	281.7 (38.3%)
Qwen3-8B	Original	814.1 (36.3%)	856.5 (39.0%)	638.5 (36.7%)	778.5 (47.6%)
Qwen3-8B	Distillable	626.3 (31.6%)	702.7 (32.4%)	498.1 (27.8%)	679.0 (36.6%)
Qwen3-8B	Undistillable	784.5 (46.3%)	835.8 (41.0%)	621.0 (48.7%)	759.9 (47.2%)

Table 4: Trap-related KL metrics for calibrated vs. original teachers (students: Gemma-3-4B, Qwen3-1.7B). Distillable calibration reduces token RKL and wrong-trace preference rates, while undistillable calibration amplifies them, confirming that calibration systematically controls the identified traps.

6.5 Undistillable Mode Analysis

To understand how distillation fails, we analyze generations from the misled students across tasks. We focus on prompts that students previously solved correctly by majority vote but that the misled students failed after distillation. We follow an iterative procedure: (i) manually inspect a seed set of prompts and completions, (ii) use an LLM judge (Claude Sonnet 4.5) to retrieve similar cases, and (iii) refine the taxonomy and repeat. Across both model families, we observe three recurring failure modes.

Extrinsic Hallucination: The overconfidence and failure to recognize uncertainty, stemming from RLVR reward hacking, are distilled to the student.

Misled students often produce completions that are linguistically sound but semantically incoherent or factually incorrect. This is most common in knowledge-retrieval QA tasks, where missing factual knowledge cannot be recovered by inference compute. A consistent signal is a large gap between pass@16 and maj@16. As suggested by Kalai et al. (2025), we hypothesize that RLVR post-training amplifies "guessing answer" behavior. And knowledge distillation transfers such reward-driven overconfidence to the student, turning uncertainty into extrinsic hallucination.

Intrinsic Hallucination: Misled students fail to commit, triggering self-correction collapse.

In reasoning and knowledge-induction tasks, misled students reach correct intermediate answers but enter unnecessary recheck phases (e.g., "Wait, let me verify") and flip to wrong conclusions. Conditioning teachers on the same prefix preserves correctness 89–95% of the time (Figure 4), indicating the error stems from distillation induced instability rather than missing knowledge. This aligns with recent studies on reasoning loops (Pipis et al., 2025): when "stop and commit" is hard to learn, models favor self-referential continuations.

Local Degradation: Objective mismatch and tail noise compound into token repetition and instruction-following drift.

Beyond semantic failures, we observe local decoding degradation, where students repeat short spans of tokens/phrases and occasionally drift in answer formatting (e.g., switching labeling schemes). Once repetition begins, the model rarely recovers within the same completion. We attribute this brittleness to the mismatch between sequence-level post-training rewards and token-level distillation losses. Small distributional errors, particularly the tail noise identified in our pilot study, compound over time into local degradation.

7 Conclusion

In this paper, we started by analyzing why KD fails and identified several distillation traps via a KL-statistics pilot study and an empirical failure-mode analysis. To control LLM distillability, we propose the first post-hoc calibration method in which a single coefficient η dictates either yielding distillable teachers that improve students or yielding undistillable teachers that turn traps into guards for IP protection. In future work, we aim to scale our method to larger model sizes and explore the knowledge distillation dynamics of MoE models.

Limitations

Our calibration method requires fine-tuning the teacher model via reinforcement learning, which is typically more computationally intensive than standard student distillation training.

Ethical Considerations

While our undistillable teacher method is designed to protect intellectual property and study KD failure modes, we acknowledge that such techniques could potentially be misused to amplify hallucination and misinformation within LLMs (Ho et al., 2024; Xiao et al., 2025; Liang et al., 2025a).

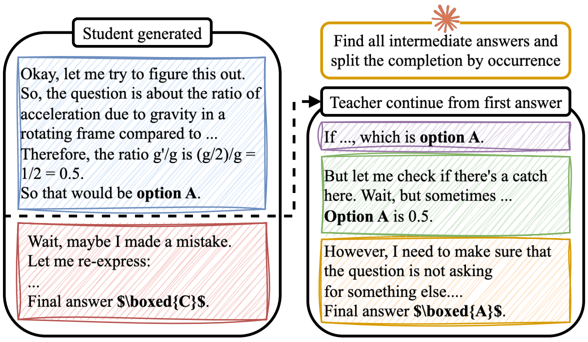


Figure 4: Example of self-correction collapse. The misled student reaches a correct intermediate answer, then continues with self-checking and changes to an incorrect final answer. Conditioning the teacher on the same prefix yields a stable continuation that retains the correct answer.

Acknowledgments

This research / project is supported by the National Research Foundation, Singapore, and Cyber Security Agency of Singapore under its National Cybersecurity R&D Programme and CyberSG R&D Cyber Research Programme Office. Any opinions, findings and conclusions or recommendations expressed in these materials are those of the author(s) and do not reflect the views of National Research Foundation, Singapore, Cyber Security Agency of Singapore as well as CyberSG R&D Programme Office, Singapore.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. *Preprint*, arXiv:2306.13649.

AI@Meta. 2024. Llama 3 model card.

Alon Albalak, Duy Phung, Nathan Lile, Rafael Rafailov, Kanishk Gandhi, Louis Castricato, Anikait Singh, Chase Blagden, Violet Xiang, Dakota Mahan, and Nick Haber. 2025. Big-math: A large-scale, high-quality math dataset for reinforcement learning in language models. *Preprint*, arXiv:2502.17387.

Anshumann, Mohd Abbas Zaidi, Akhil Kedia, Jinwoo Ahn, Taehwak Kwon, Kangwook Lee, Haejun Lee, and Joohyung Lee. 2025. Sparse logit sampling: Accelerating knowledge distillation in llms. *Preprint*, arXiv:2503.16870.

Hongzhan Chen, Ruijun Chen, Yuqi Yi, Xiaojun Quan, Chenliang Li, Ming Yan, and Ji Zhang. 2024. Knowledge distillation of black-box large language models. *Preprint*, arXiv:2401.07013.

Zigeng Chen, Gongfan Fang, Xinyin Ma, Ruonan Yu, and Xinchao Wang. 2026. dparallel: Learnable parallel decoding for dllms. In *International Conference on Learning Representations*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

DeepMind. 2025. Gemma 3 technical report. *Preprint*, arXiv: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, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

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

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. Minillm: Knowledge distillation of large language models. *Preprint*, arXiv:2306.08543.

Yuxian Gu, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2025. Miniplm: Knowledge distillation for pre-training language models. *Preprint*, arXiv:2410.17215.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *Preprint*, arXiv:1503.02531.

Zheng Yi Ho, Siyuan Liang, Sen Zhang, Yibing Zhan, and Dacheng Tao. 2024. Novo: Norm voting off hallucinations with attention heads in large language models. *arXiv preprint arXiv:2410.08970*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. 2025. Why language models hallucinate. *Preprint*, arXiv:2509.04664.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. *Preprint*, arXiv:1606.07947.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2024. A watermark for large language models. *Preprint*, arXiv:2301.10226.

Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. 2024. Revisiting catastrophic forgetting in large language model tuning. *Preprint*, arXiv:2406.04836.

Pingzhi Li, Zhen Tan, Huaizhi Qu, Huan Liu, and Tianlong Chen. 2025. Doge: Defensive output generation for llm protection against knowledge distillation. *Preprint*, arXiv:2505.19504.

Siyuan Liang, Tianmeng Fang, Zhe Liu, Aishan Liu, Yan Xiao, Jinyuan He, Ee-Chien Chang, and Xiaochun Cao. 2025a. Safemobile: Chain-level jailbreak detection and automated evaluation for multimodal mobile agents. *arXiv preprint arXiv:2507.00841*.

Siyuan Liang, Jiajun Gong, Tianmeng Fang, Aishan Liu, Tao Wang, Xiaochun Cao, Dacheng Tao, and Chang Ee-Chien. 2026a. Trapflow: Controllable website fingerprinting defense via dynamic backdoor learning. *IEEE Transactions on Information Forensics and Security*.

Siyuan Liang, Jiawei Liang, Tianyu Pang, Chao Du, Aishan Liu, Mingli Zhu, Xiaochun Cao, and Dacheng Tao. 2025b. Revisiting backdoor attacks against large vision-language models from domain shift. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9477–9486.

Siyuan Liang, Jiayang Liu, Jiecheng Zhai, Tianmeng Fang, Rongcheng Tu, Aishan Liu, Xiaochun Cao, and Dacheng Tao. 2026b. T2vshield: Model-agnostic jailbreak defense for text-to-video models. *International Journal of Computer Vision*, 134(4):144.

Siyuan Liang, Kuanrong Liu, Jiajun Gong, Jiawei Liang, Yuan Xun, Ee-Chien Chang, and Xiaochun Cao. 2024a. Unlearning backdoor threats: Enhancing backdoor defense in multimodal contrastive learning via local token unlearning. *arXiv preprint arXiv:2403.16257*.

Siyuan Liang, Mingli Zhu, Aishan Liu, Baoyuan Wu, Xiaochun Cao, and Ee-Chien Chang. 2024b. Badclip: Dual-embedding guided backdoor attack on multimodal contrastive learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 24645–24654.

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, Yang Liu, and Yahui Zhou. 2025a. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *Preprint*, arXiv:2507.01352.

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. 2025b. Part i: Tricks or traps? a deep dive into rl for llm reasoning. *Preprint*, arXiv:2508.08221.

Kevin Lu and Thinking Machines Lab. 2025. On-policy distillation. *Thinking Machines Lab: Connectionism*. Https://thinkingmachines.ai/blog/on-policy-distillation.

Haoyu Ma, Tianlong Chen, Ting-Kuei Hu, Chenyu You, Xiaohui Xie, and Zhangyang Wang. 2021. Undistillable: Making a nasty teacher that cannot teach students. *Preprint*, arXiv:2105.07381.

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

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. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Charilaos Pipis, Shivam Garg, Vasilis Kontonis, Vaishnavi Shrivastava, Akshay Krishnamurthy, and Dimitris Papailiopoulos. 2025. Wait, wait, wait... why do reasoning models loop? *Preprint*, arXiv:2512.12895.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Zhiyao Ren, Siyuan Liang, Aishan Liu, and Dacheng Tao. 2025. Iclshield: Exploring and mitigating in-context learning backdoor attacks. *arXiv preprint arXiv:2507.01321*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *Preprint*, arXiv:1910.01108.

John Schulman. 2020. Approximating KL divergence. Blog post.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv: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. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. 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)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

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, and 76 others. 2025. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *Preprint*, arXiv:2502.14739.

Inar Timiryasov and Jean-Loup Tastet. 2023. Baby llama: knowledge distillation from an ensemble of teachers trained on a small dataset with no performance penalty. *Preprint*, arXiv:2308.02019.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Qi Wang and Jinjia Zhou. 2025. Topkd: Top-scaled knowledge distillation. *Preprint*, arXiv:2508.04539.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. 2025. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Preprint*, arXiv:2506.01939.

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. 2024. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Preprint*, arXiv:2406.01574.

Yuqiao Wen, Zichao Li, Wenyu Du, and Lili Mou. 2023. f-divergence minimization for sequence-level knowledge distillation. *Preprint*, arXiv:2307.15190.

Yisong Xiao, Aishan Liu, Siyuan Liang, Zonghao Ying, Xianglong Liu, and Dacheng Tao. 2025. Detoxifying large language models via autoregressive reward guided representation editing. *arXiv preprint arXiv:2510.01243*.

Jiashu Xu, Fei Wang, Mingyu Derek Ma, Pang Wei Koh, Chaowei Xiao, and Muhao Chen. 2024a. Instructional fingerprinting of large language models. *Preprint*, arXiv:2401.12255.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. 2024b. A survey on knowledge distillation of large language models. *Preprint*, arXiv: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, and 41 others. 2025. Qwen3 technical report.

Zonghao Ying, Aishan Liu, Siyuan Liang, Lei Huang, Jinyang Guo, Wenbo Zhou, Xianglong Liu, and Dacheng Tao. 2026. Safebench: A safety evaluation framework for multimodal large language models. *International Journal of Computer Vision*, 134(1):18.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *Preprint*, arXiv:2308.01825.

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

Junzi Zhang, Jongho Kim, Brendan O'Donoghue, and Stephen Boyd. 2020. Sample efficient reinforcement learning with reinforce. *Preprint*, arXiv:2010.11364.

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

Appendix

A Extended Related Work

In this section, we present an extended review of related work that goes beyond what could be included in the main text due to space constraints.

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 yielding students who significantly outperform those 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 and Rush (2016) trained students on full sequences generated by the teacher, allowing them to learn sequence-level distributions. 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 property (IP). Methods for protecting the IP of machine learning models can be broadly categorized as reactive or proactive (Liang et al., 2026a; Ren et al., 2025; Liang et al., 2024a). 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) (Liang et al., 2024b, 2025b, 2026b). 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 autoregressive 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 (Liang et al., 2025a).

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 an "action" corresponds to generating 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., 2025b). Whether 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 into distillation traps.

B Sequence KL Gradient Derivation

We consider the sequence RKL divergence objective for Knowledge Distillation, where the student policy is parameterized by θ (i.e., $\pi_\theta \equiv \pi_S$):

$$\begin{aligned}\mathcal{J}(\theta) &= \mathcal{D}(\pi_\theta \| \pi_T) \\ &= \mathbb{E}_{s\sim\pi_\theta}\left[\log\frac{\pi_\theta(s)}{\pi_T(s)}\right] \\ &= \sum_s \pi_\theta(s)\log\frac{\pi_\theta(s)}{\pi_T(s)},\end{aligned} \tag{12}$$

where $s = (a_1, \ldots, a_T)$ is a sequence sampled from the student policy π_θ, with $\pi_\theta(s) = \prod_{t=1}^{T}\pi_\theta(a_t|s_t)$. Let $R(s) = \log\frac{\pi_\theta(s)}{\pi_T(s)} = \log\pi_\theta(s) - \log\pi_T(s)$. Differentiating Equation (12) gives:

$$\begin{aligned}\nabla_\theta\mathcal{J}(\theta) &= \nabla_\theta\sum_s \pi_\theta(s)R(s) \\ &= \sum_s \nabla_\theta\pi_\theta(s)R(s) + \sum_s \pi_\theta(s)\nabla_\theta R(s) \\ &= \sum_s \nabla_\theta\pi_\theta(s)R(s) + \sum_s \pi_\theta(s)\nabla_\theta\log\pi_\theta(s) \\ &= \sum_s \nabla_\theta\pi_\theta(s)R(s) + \sum_s \nabla_\theta\pi_\theta(s) \\ &= \sum_s \nabla_\theta\pi_\theta(s)R(s) \\ &= \mathbb{E}_{s\sim\pi_\theta}\left[\nabla_\theta\log\pi_\theta(s)\cdot R(s)\right].\end{aligned} \tag{13}$$

Since π_T is fixed, $\nabla_\theta R(s) \propto \nabla_\theta\log\pi_\theta(s)$ and the corresponding term cancels because of score function property $\sum_s \nabla_\theta\pi_\theta(s) = \nabla_\theta 1 = 0$. We can rewrite and obtain the REINFORCE policy gradient form:

$$\begin{aligned}\nabla_\theta\log\pi_\theta(s) &= \sum_{t=1}^{T}\nabla_\theta\log\pi_\theta(a_t|s_t) \\ \nabla_\theta\mathcal{J}(\theta) &= \mathbb{E}_{s\sim\pi_\theta}\left[\sum_{t=1}^{T}\nabla_\theta\log\pi_\theta(a_t|s_t)\cdot R(s)\right].\end{aligned} \tag{14}$$

Now, let us examine the gradient with respect to the pre-softmax logits $z_{t,v}$ of the student model at a specific step t for token v.

$$\pi_\theta(\cdot|s_t) = \mathrm{softmax}(z_t) = \frac{\exp(z_{t,a_t})}{\sum_{v'} \exp(z_{t,v'})}$$
$$\log \pi_\theta(a_t|s_t) = z_{t,a_t} - \log \sum_{v'} \exp(z_{t,v'})$$
$$\frac{\partial \log \pi_\theta(a_t|s_t)}{\partial z_{t,v}} = \mathbb{1}_{v=a_t} - \pi_\theta(v|s_t). \tag{15}$$

Thus, for a single sampled trajectory s, the corresponding stochastic gradient estimator is:

$$\nabla_{z_{t,v}} \mathcal{J} \approx (\mathbb{1}_{v=a_t} - \pi_\theta(v|s_t)) \cdot R(s). \tag{16}$$

Case 1: Realized Token ($v = a_t$). The gradient is $(1 - \pi_\theta(a_t|s_t))R(s)$. The direction depends on the return $R(s)$. If the sequence has low divergence (high reward), the probability of a_t is increased.

Case 2: Unrealized Token ($v \neq a_t$). The gradient is $-\pi_\theta(v|s_t)R(s)$. Notice that this expression contains $\pi_\theta(v|s_t)$ and the scalar return $R(s)$, but it does *not* contain $\pi_T(v|s_t)$. The teacher's distribution π_T only affects the scalar return $R(s)$ through the realized tokens. It provides no specific signal for unrealized tokens. Consequently, if we have two unrealized tokens v_1 and v_2 such that $\pi_\theta(v_1|s_t) = \pi_\theta(v_2|s_t)$, their gradient updates are identical, even if the teacher assigns high probability to v_1 (a valid alternative) and zero probability to v_2 (an error). This confirms that sequence KL ignores the "dark knowledge" inherent in the teacher's distribution over valid but unselected tokens.

C Extended Pilot Study

Figure 5 shows the KL plots for Gemma models, complementing the Qwen plot in Figure 2.

D Extended Implementation Detail

Algorithm 1 was implemented by modifying the GRPO trainer (von Werra et al., 2020) to group-normalize each reward independently before summing them to compute the overall advantage. We also modified the GKD trainer (Agarwal et al., 2024) to leverage vLLM for fast generation and added accuracy hooks that log the generated sequence's accuracy for monitoring during training.

Teacher calibration is performed using LoRA (Hu et al., 2021) with rank $r = 128$ and scaling factor $\alpha = 128$. Both teacher and student models are trained with $2e-5$ learning rate and 4 rollouts per prompt.

Algorithm 1 Controlling Distillability

Hyper-parameters: distillability coefficient η (< 0 for distillable, > 0 for undistillable), KL anchor coefficient β, training steps N, batch size B, num rollouts G
Input: Original teacher π_T, calibration target π_C, training dataset $\mathcal{D}$, reward function R_{task}
Initialize $\pi_\theta \leftarrow \pi_T$
Freeze π_T, π_C
for step $= 1$ to N **do**
  Sample batch of prompts $\{q^{(b)}\}_{b=1}^{B}$ from $\mathcal{D}$
  for each prompt $q^{(b)}$ **do**
    // Generate G rollouts
    $\{s_i\}_{i=1}^{G} \sim \pi_\theta(\cdot|q^{(b)})$
    for each sequence s_i **do**
      $r_i^{\text{task}} \leftarrow R_{\text{task}}(s_i)$
      $r_{\text{calib}}^{(i)} \leftarrow \mathrm{sg}[\log \pi_\theta(s_i)] - \log \pi_C(s_i)$
    end for
    $\hat{r}_i^{\text{task}} \leftarrow \mathrm{GroupNorm}(r_i^{\text{task}})$
    $\hat{r}_i^{\text{calib}} \leftarrow \mathrm{GroupNorm}(r_i^{\text{calib}})$
    // Sequence advantages
    $A_i \leftarrow \hat{r}_i^{\text{task}} + \eta \cdot \hat{r}_i^{\text{calib}}$
  end for
  // Update policy via policy optimization
  $\pi_\theta \leftarrow \mathrm{PO}(\pi_\theta, \pi_T, s_{1:G}^{(1:B)}, A_{1:G}^{(1:B)})$
end for
Return calibrated teacher π_θ

E Computational Cost

Table 5 reports GPU hours (H100) for each stage of the Qwen family pipeline. Teacher calibration (141.5 GPU·h) is comparable to a single on-policy GKD-RKL run (196.8 GPU·h) and is a one-time cost that amortizes across downstream tasks and students. This makes calibration practical in two key scenarios: (i) **one-teacher-many-students**, where a single calibrated teacher serves multiple student architectures or tasks (Section 6.3); and (ii) **IP protection**, where the calibration cost is a one-time investment to prevent unauthorized knowledge extraction.

Process (1 epoch)	GPU·h
Teacher calibration (RFT, LoRA)	141.5
Teacher sampling (off-policy, up-front)	∼50
Student SFT	9.8
Student GKD w/ FKL	12.2
Student GKD w/ on-policy RKL	196.8

Table 5: GPU-hour breakdown.

F Extended Model Evaluation

We further validate that our undistillable calibration generalizes by extending to additional open-source model pairs to analyze their in-/out-of-/cross-distribution behaviors. Specifically, we dis-

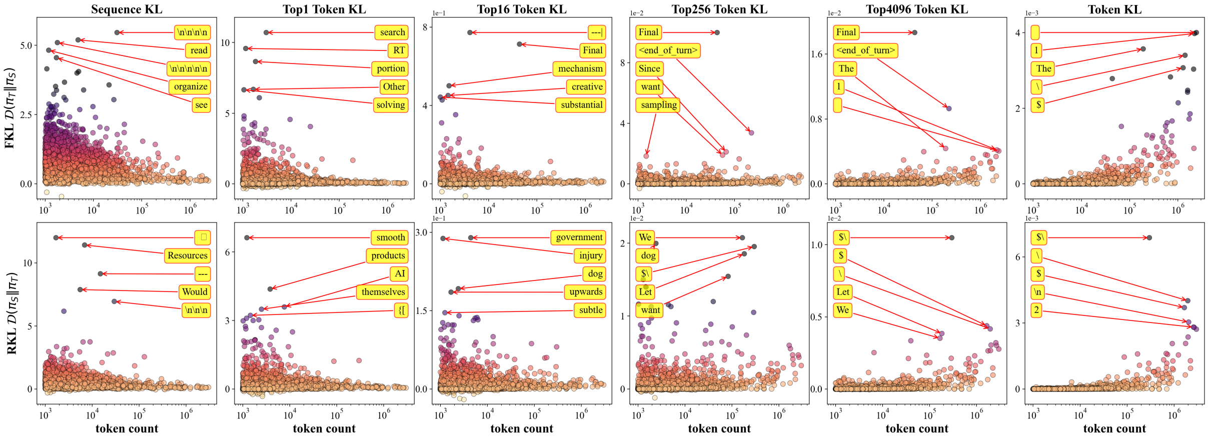


Figure 5: Accumulated KL Divergence vs. Token Count for Gemma models. Each circle is a unique token in the vocabulary. The x-axis (log scale) is realized count N_v, and the y-axis is the average KL contribution $\Phi_{K,v}/N_{K,v}$.

Model	Method	gsm8k	CSQA	MMLU-Pro	superGPQA
DS-R1-8B	Teacher	0.799	0.774	0.323	0.133
DS-R1-8B	Undistillable	0.792	0.767	0.323	0.143
Qwen3-4B	Student	0.889	0.797	0.450	0.171
Qwen3-4B	Misled	0.118	0.062	0.009	0.011
Llama-3.1-8B	Teacher	0.817	0.729	0.388	0.177
Llama-3.1-8B	Undistillable	0.812	0.731	0.374	0.188
Llama-3.2-3B	Student	0.672	0.680	0.235	0.142
Llama-3.2-3B	Misled	0.064	0.028	0.014	0.005

Table 6: Additional undistillable teachers and misled students evaluation.

Model	Method	CSQA	MMLU-Pro	superGPQA
DS-R1-8B	Undistillable (OOD)	0.757	0.323	0.140
Qwen3-4B	Misled (OOD)	0.049	0.027	0.015
Qwen3-4B	Misled (CD)	0.040	0.005	0.014
Llama-3.1-8B	Undistillable (OOD)	0.723	0.389	0.184
Llama-3.2-3B	Misled (OOD)	0.031	0.021	0.009
Llama-3.2-3B	Misled (CD)	0.099	0.035	0.009

Table 7: OOD and CD evaluations.

tilled *Qwen3-4B* from *DeepSeek-R1-0528-Qwen3-8B* (*DS-R1-8B*) and *Llama-3.2-3B-Instruct* from *Llama-3.1-8B-Instruct* (DeepSeek-AI et al., 2025; Yang et al., 2025; AI@Meta, 2024). We used the same undistillable setting as in the main text ($\eta = 1$) and distilled students with on-policy GKD and RKL loss.

Table 6 shows the in-distribution setup, where the teachers are calibrated and students are distilled on same dataset. The undistillable teachers preserve accuracy close to their original counterparts, while misled students collapse across all evaluated datasets. Table 7 summarizes OOD and CD evaluation. In OOD evaluation, the teacher was calibrated on the gsm8k dataset; then the misled student was distilled from the resulting undistillable teacher on the gsm8k dataset. In CD evaluation, we calibrated the teacher on the gsm8k dataset; then the misled student was distilled from the resulting undistillable teacher on the evaluation dataset.

G Extended Model Output

This section provides qualitative excerpts from undistillable teachers and their misled students, illustrating how distillation failure manifests at the level of generation behavior.

These examples complement the failure-mode analysis in Section 6.5. Across model families, undistillable teachers often exhibit high-frequency self-checking and spurious detours. The resulting misled students amplify these artifacts into repetition loops and instruction-following drift, frequently failing to commit a final answer within the context window.

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>

DS-R1-8B 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>...
[kept repeating </think>]

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']

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>
```

H Evaluation Prompts

We utilized the following prompts for tasks requiring numerical answers and single-choice option answers respectively for consistent output formatting for automated evaluation.

Prompt for numerical 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)
```

I Future Work

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.

AI Assistants used for language editing, including grammar, punctuation, and verb/noun agreement correction.