Undistillable Open Language Models with Teacher Scrambling

Sebastian E. Dionicio
Dalhousie University
sdionicio@dal.ca

Aniq Elahi
Dalhousie University
aelahi@dal.ca

Domenic Rosati
Dalhousie University
drosati@dal.ca

Hassan Sajjad
Dalhousie University
hsajjad@dal.ca

Abstract

Open-weight security requires that post-release foundation models are resistant to misuse. Even if a model is made *unmodifiable*, an attacker may *distill* it into a new model they *can* modify. Previous works have examined preventing distillation of closed-access models. We analyze undistillability under the constraint that an attacker has access to unmodifiable language model weights and introduce **Teacher Scrambling**, a novel method that preserves task utility for the original model while preventing information gain from the logit rank distribution via a logit rank scrambling loss. We show that attempting to distill student models from a scrambled teacher results in worse performance than training with label smoothing, therefore defeating the purpose of attempted distillation.

1 Introduction

Open-weight security is an emerging research direction [15, 17] that considers whether it is possible to prevent openly released models like GPT-oss [1] from being modified for harmful purposes. Even if these models were made completely *unmodifiable*, e.g. by Rosati et al. [16], they would be subject to a critical vulnerability: an attacker could distill the model into a *modifiable* model. Previous works [7, 20] have considered undistillability behind closed-access APIs to prevent model exfiltration. We explore the novel security constraint of *white-box undistillability*, where a defence must prevent distillation of a teacher despite having full access to the teacher weights.

Threat Model We assume the open "teacher" model is made unmodifiable, i.e. it cannot be trained [16]. The adversary's **goal** is training a student through distillation as a much cheaper way to match a teacher's performance than training from scratch, so the model can be modified, e.g. to undo a safety guard. The attacker has complete **access** to training code, dataset, and **budget** to distill a model.

White-box Threats Previous works [7, 11, 19, 20] assume the model is behind a closed-access API, which readily admits simple solutions such as withholding the logit values or truncating model outputs as is done by Anthropic and OpenAI (see discussion in 8). Our setting introduces a novel design constraint, that the logit values must be uninformative for distillation post-release without any further intervention, as well as a novel set of adaptive attacks e.g. using representation-based distillation [2].

Contribution Figure 1 shows a sample from the **scrambled teacher** defence. Our results, Table 1, show that ***distilling from a scrambled teacher is much worse than training a student using label smoothing alone***, effectively defeating the purpose of using the model for distillation. In this extended abstract, we explore the effectiveness of our method empirically in the large language model distillation setting (§ 4). Future work will examine additional settings such as vision, unique adaptive attacks introduced in the white-box setting, and provide a theoretical analysis of our method.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Lock-LLM.

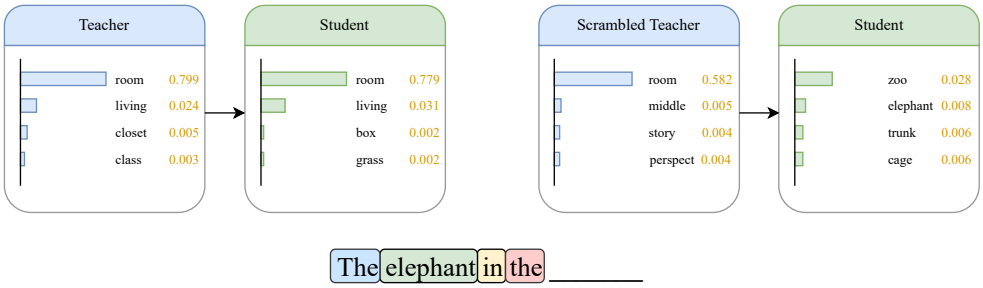

Figure 1: **Standard distillation** has student models match the teacher's logit distribution. **Scrambled teacher** preserves top-k=1 utility, but tail logits are uninformative, resulting in ineffective distillation.

2 Problem Statement: Undistillability

Definition 1 (ϵ-*undistillable*)**.** *Following Yang and Ye [19], we say a teacher neural network T is* ϵ-undistillable *w.r.t. a student model $S \in \mathcal{S}$, dataset $\mathcal{D}$, and knowledge distillation (KD) objective $\mathcal{L}_{\text{KD}}$, described below, if for any $S \in \mathcal{S}$ trained to minimize $\mathcal{L}_{\text{KD}}$ on samples from $\mathcal{D}$, the student's utility $\phi : \mathcal{S} \to \mathbb{R}$ satisfies*

$$\phi(S) \leq \phi(S_{\text{LS}}) + \epsilon,$$

where S_{LS} is the same model trained with label smoothing (LS) on ground-truth labels only.

LS is used because Yang and Ye [19] observed it is more effective than cross-entropy.

Informally, there would be no benefit by distilling from ϵ-*undistillable* teacher since supervised fine-tuning (SFT) would be more effective on its own, and distillation is typically much more effective than SFT [10]. This is a successful defence against model exfiltration and distillation-based attacks because we assume the attacker already has access to the training dataset $\mathcal{D}$ and therefore there is less information gain over cross-entropy loss. In our white-box setting, we are constrained to defence solutions that we apply to the model **before release** as that is the only time the defender can intervene.

3 Method: The Scrambled Teacher

Let $z \in \mathbb{R}^{|C|}$ be the final-layer output values, the "logits," from a multi-class classification model with classes C and inputs and labels $(x, y)_i \in \mathcal{D}$. We train this model to *preserve* top-k classification performance but match the uniform distribution for the bottom-n label output values ($n = |C| - k$).

Our overall objective given a teacher model parameterized by θ is

$$\mathcal{L}_{scramble}(x, y) \;=\; \underbrace{\text{CE}(y,\, z_y)}_{\text{task utility}} + \lambda \underbrace{\text{KL}\big(p_k(z) \,\big\|\, u_k\big)}_{\text{tail uniformization}}, \tag{1}$$

where CE is cross-entropy loss considering the target y and logit-value for the target index z_y, $p_k(z)$ is the teacher's softmax restricted and renormalized to the bottom $|C| - k$ classes, u_k is the uniform distribution on those classes. Hyperparameter λ trades off undistillability and task utility preservation.

Intuition The KD loss for a student with logits s and temperature τ is

$$\mathcal{L}_{\text{KD}}(s; z) \;=\; \tau^2 \,\text{KL}\Big(\text{softmax}\left(\frac{z}{\tau}\right) \,\big\|\, \text{softmax}\left(\frac{s}{\tau}\right)\Big).$$

Let $\tau = 1$, $p = \text{softmax}\left(\frac{z}{\tau}\right)$, and $q = \text{softmax}\left(\frac{s}{\tau}\right)$. For simplicity of analysis we have omited the standard CE term in many KD losses (see Eq. 2). Gradients of $\nabla_p \mathcal{L}_{\text{KD}}$ depend on the *relative* probabilities over many classes. Thus we have:

$$\nabla_p \mathcal{L}_{\text{KD}} = \log p - \log q + 1.$$

If the top-k logits are preserved while the tail becomes near-uniform, then, for the student, imitating the teacher resembles learning from almost-hard labels, i.e., cross-entropy with label smoothing. We

call this method scrambled because the logit rank for the minimizer of Eq 1 would be in a random order (scrambled) compared to the original teacher, since the bottom-n logit rank is uniform.

4 Experimental Setting

We empirically validate our method using an auto-regressive decoder-only large language model distillation setting, where the task is next token prediction over a vocabulary $\mathcal{V}$ of length $|\mathcal{V}|$ where the bottom-n logits are the $|\mathcal{V}| - k$ tail token prediction values. The teacher model is a much larger pretrained language model, and the student is usually a much smaller pretrained language model.

Metrics We report perplexity (PPL), KL divergence against the original teacher logit values for the same samples (KL), top-k overlap (fractional overlap of teacher and student top-k sets). In order to assess utility, we report the macro-averages of typical language model evaluation tasks.

Setup Teachers and students are from the SmolLM2 [3] family: Teacher (*1.7B* parameters), Teacher (Control) (*360M*), and a base Student (*135M*) that is used for all KD experiments. Training data is a filtered subset of FineWeb-Edu [14] (total of 3,275,200 tokens). We train for 100K steps, with a batch size of 2048, and evaluate with common language model utility tasks including: ARC-C/E, CommonsenseQA (CSQA), HellaSwag, OpenBookQA, PIQA, and Winogrande [4]. KD uses temperature τ=2.0 and loss mixing α=0.7; see Appendix A for full hyperparameters as well as additional experimental details.

Results Table 1 summarizes PPL (↓) / KL (↓) / top-k overlap (↑) and Table 2 summarizes LM performance. Three consistent patterns emerge:

Table 1: Scrambled teachers are undistillable considering their students' performance

Model	PPL	KL	Top-k
Teacher	9.34	0.00	1.00
Teacher (Control)	12.81	0.35	0.70
Scrambled Teacher	12.36	1.15	0.88
Student (Base)	15.68	0.47	0.64
Student (LS)	15.98	0.69	0.64
Student (KD)	15.45	0.43	0.66
Student (KD-Control)	16.87	0.44	0.64
Student (Scrambled)	32.67	1.36	0.56

Scrambled teachers preserve their utility. The perplexity of the scrambled teacher (12.36) is slightly degraded from the original teacher (9.34) but is still better than the control (12.81). A similar pattern can be seen in Table 2.

KD from scrambled teachers underperform label smoothing. Student with a scrambled teacher yields PPL 32.67 vs. 15.98 for the Label Smoothing student, satisfying ϵ-undistillability.

KD from scrambled teachers results in poor performing students. In both Table 1 and 2, we see that the attacker has no incentive to distill from this model as it harms overall utility.

Takeaway For language models, scrambling the tail removes the information needed for KD to perform well while keeping the teacher's utility. The LM performance evaluation pattern is consistent: students distilled from scrambled teachers lag behind those distilled from the original teacher across tasks, which satisfies our definition of ϵ-undistillability.

5 Related Works

Ma et al. [11] propose training a teacher whose incorrect-class logits are deliberately perturbed so the soft targets become unhelpful for knowledge distillation (KD) while preserving the correct class. This increases class ambiguity and degrades students on vision benchmarks. However, Jandial et al. [9] show that variants of KD and alternative objectives can *circumvent* such teachers, recovering substantial student performance. Their result implies that "making logits nasty" at the level of logit magnitudes alone is not a sufficient condition for undistillability once the attacker adapts. We differ in two ways: (i) our defence targets the *ordering* of bottom-n logits (rank semantics) rather than only their sparsity or noise level, and (ii) we study open-weight *language models* and *white-box* adversaries, where API mediation is unavailable and KD pipelines are easily customized.

Table 2: Macro-average accuracy (↑). Scrambled Teacher downstream task accuracy is only slightly degraded, while a student distilled from this model performs substantially worse.

Model	ARC	CQA	HellaSwag	OpenBookQA	PIQA	Winogrande
Teacher	0.78	0.42	0.71	0.44	0.77	0.77
Control Teacher	0.70	0.19	0.54	0.38	0.72	0.56
Scrambled Teacher	0.74	0.41	0.68	0.42	0.76	0.65
Student (Base)	0.64	0.19	0.43	0.33	0.68	0.53
Student (LS)	0.63	0.19	0.43	0.34	0.68	0.53
Student (KD)	0.66	0.19	0.44	0.35	0.70	0.54
Student (KD-Control)	0.62	0.19	0.42	0.33	0.67	0.53
Student (Scrambled)	0.52	0.19	0.39	0.31	0.63	0.50

Stingy Teacher truncates to top-k logits at inference time [12]; it harms KD but preserves the relative ranks among the retained classes. AST [21] trains a scrambled teacher using an alternative divergence (EPD) and adversarial examples to jointly reduce smoothness and increase sparsity. AST did some preliminary experiments comparing their method in a white-box setting in the vision domain, which we extend to the language domain and formulate as a novel security constraint. These studies primarily modify logit magnitudes rather than rank order and are focused on the vision domain. API-level defences like APGP and CMIM restrict or collapse outputs to reduce extractability [7, 20], but they presuppose hosted models; our defence is meant to be applied to open weight models that are publicly distributed. Additional methods attempt to make distillation produce strictly worse models than cross-entropy loss [13]. This is not a requirement of our method, as we assume that the attacker will simply use cross-entropy if the student is worse; making student models worse than cross-entropy is unnecessary. Finally, model-extraction/IP work studies stealing hosted models [6, 18], which are not relevant to our threat model considering the attacker already has the weights.

6 Ablation Study

Our training-time *scrambled teacher* (§3) flattens the tail distribution while largely preserving top-k behavior, thereby degrading the KD signal. To probe *how much* of KD's advantage comes from tail *rank* information alone, we introduce a *post-training* permutation variant that we call the **rank-derangement oracle** (Oracle): it preserves the head exactly but *completely destroys* rank order inside the tail by permuting logits while conserving their overall mass. Because it maximally corrupts tail rank semantics *without retraining* and keeps the head fixed, we use it as an "upper bound" stress test for KD: if KD still fails under this perturbation, the tail's rank structure is indeed essential.

Let $z \in \mathbb{R}^{|\mathcal{V}|}$ be final logits, $\mathcal{H}_k$ the *head* (top-k indices), and $\mathcal{T}_k = [|\mathcal{V}|] \setminus \mathcal{H}_k$ the *tail*. We draw a derangement π on $\mathcal{T}_k$ (a permutation with no fixed points) and only permute tail coordinates:

$$\tilde{z}_i = \begin{cases} z_i, & i \in \mathcal{H}_k, \\ z_{\pi(i)}, & i \in \mathcal{T}_k. \end{cases} \qquad \tilde{p} = \mathrm{softmax}(\tilde{z}/\tau).$$

This preserves *exact head* membership and scores, conserves total tail mass, and erases tail rank semantics.

Algorithm 1 Rank-Derangement Tail Permutation (Oracle)

Require: logits $z \in \mathbb{R}^{|\mathcal{V}|}$, head size k, derangement π over tail indices
$\mathcal{H}_k \leftarrow \mathrm{TopK}(z, k)$; $\mathcal{T}_k \leftarrow [|\mathcal{V}|] \setminus \mathcal{H}_k$
$\tilde{z} \leftarrow z$
for all $i \in \mathcal{T}_k$ **do**
    $\tilde{z}_i \leftarrow z_{\pi(i)}$
end for
return $\tilde{z}$

Table 3: Students distilled from the oracle teacher. For reference: KD from the original 1.7B yields PPL 15.45, LS-ONLY yields PPL 15.98 (Tables 1–2).

Student (KD from Oracle k)	**PPL ↓**	**KL vs 1.7B ↓**	**Top-k overlap ↑**
k=1	61.47	1.31	0.51
k=5	29.94	0.99	0.56
k=10	27.68	0.93	0.55
k=100	19.98	0.53	0.62

Oracle Student-side behavior We distill the same 135M student from the oracle teacher at matched hyperparameters. Table 3 shows that *all* oracle-k settings yield students that are worse than both KD from the unmodified teacher and LS-ONLY baselines (cf. Tables 1–2). The effect weakens as k grows, but remains well within our ϵ-undistillability criterion.

Oracle Teacher-side behavior Table 4 summarizes the teacher metrics after applying the oracle at different k (larger k means a smaller tail). As designed, the *top-k overlap vs. original is 1.00* for all k, and downstream *task accuracies are unchanged* (identical to the original teacher; see Table 2). Perplexity (PPL) increases sharply for small k since NLL is sensitive to probability assigned to the true token even when it lies outside the head; as k grows (e.g., k=100), PPL approaches the original.

Effects of k in overall Teacher behaviour Varying the head size k trades off utility preservation against KD suppression in predictable ways. Table 4 compares *trained scrambling* to the *oracle* at matched k. On the *teacher* side, the oracle keeps head predictions intact for all k (top-k overlap $=$ 1.00) and leaves downstream task accuracies unchanged (Table 2), but its PPL spikes for small heads. In contrast, trained scrambling maintains a stable teacher PPL and accuracy across k, indicating that head semantics are preserved while tail rank is suppressed. On the *student* side (Table 3 and Table 1), **both** defences satisfy ϵ-undistillability for all tested k: students distilled from scrambled/oracle teachers underperform LS-only (PPL 15.98) and standard KD from the original 1.7B (PPL 15.45). The oracle is most damaging at small k (PPL 61.47 at k=1), with diminishing but still substantial harm as k increases.

Effects of λ in overall Teacher behaviour The scrambling weight λ controls the strength of tail uniformization and exhibits a smooth utility–undistillability trade-off on the *teacher*. Table 5 reports the effect of λ (fixed $k = 1$ as in the main setup): as λ increases, PPL and KL vs. the original teacher rise (stronger scrambling), while top-k overlap declines modestly, showing that head semantics are mostly preserved even as the tail is flattened. In practice, $\lambda \in [0.15, 0.20]$ offers a favorable operating region: teacher utility remains close to the original while KD supervision is already substantially degraded. Larger λ further hardens KD but at the cost of teacher perplexity and mild accuracy degradation. (Student-side λ-sweeps follow the same qualitative pattern: stronger λ monotonically worsens KD students until diminishing returns).

Table 4: Effect of head size k on **teacher** metrics for *Trained Scrambled Teacher* (1.7B) vs. *Oracle* (post-training). Larger k means a smaller tail.

Setting	k	**PPL ↓**	**KL vs 1.7B ↓**	**Top-k overlap ↑**
Trained Scrambled Teacher (1.7B)				
Scrambled	1	12.36	1.23	0.87
Scrambled	5	12.16	1.29	0.83
Scrambled	10	11.90	1.21	0.82
Scrambled	100	9.84	0.83	0.75
Oracle (post-training) (1.7B)				
Oracle	1	13296.58	3.57	1.00
Oracle	5	213.03	2.85	1.00
Oracle	10	73.30	2.55	1.00
Oracle	100	14.73	1.63	1.00

Table 5: Effect of scrambling weight λ on **teacher** metrics (1.7B; fixed $k = 1$ as in the main setup). Higher λ strengthens tail uniformization.

λ	**PPL** $\downarrow$	**KL vs 1.7B** $\downarrow$	**Top-k overlap** $\uparrow$
0.10	10.40	1.03	0.88
0.15	11.50	1.09	0.87
0.175	11.95	1.12	0.87
0.185	12.36	1.15	0.86
0.20	13.42	1.23	0.86
0.30	15.41	1.42	0.85
0.50	19.81	1.56	0.84

Takeaways The key takeaways of our ablation study are: (i) **Tail rank matters.** The tail's rank structure is a primary carrier of distillable "dark knowledge": removing rank semantics (even with mass conserved) is sufficient to break KD while keeping the head's utility. (ii) **Small k is harsher.** Reducing k strengthens the defence; however, the inference-time oracle inflates teacher PPL for small k, whereas trained scrambling avoids this instability and is suitable for open-weight release. (iii) **Scrambling strength λ is a clean knob.** Moderate λ (0.15–0.20) balances teacher utility with strong KD suppression; very large λ hardens further but is unnecessary for ϵ-undistillability.

7 Limitations and Future Work

Our initial experiments perform distillation for a relatively constrained compute budget; we plan to scale our language experiments to distillation settings such as those explored in the distillation scaling laws [5], and implement experiments in the vision domain. While the probability mass over the bottom-n tokens is substantially flattened toward a near-uniform distribution, developing training that perturbs rank structure further without destroying utility is a primary target for future work. We have not yet evaluated *adaptive* attack strategies that are unique to this white-box setting (e.g., distilling intermediate features, representation stitching, tuned-lens style supervision, or rank-only losses). Finally, a scrambled teacher may harm uncertainty calibration and sampling (e.g. they assume logit rank and values are meaningful), and we seek to understand the implications of this method for those use cases as well.

Appendix A provides a detailed breakdown of our experimental setting. Appendix B provides a qualitative study of the changes in logit rank introduced by the scrambled teacher and next token prediction behaviour changes.

8 Conclusion

We introduced the *scrambled teacher*, an undistillability defence motivated by white-box open-weight model settings. By preserving the top-k decision while flattening the bottom-n tail, the defence keeps task utility with only slight degradation but removes information needed to make distillation effective. On SmolLM2, students distilled from scrambled teachers underperform Label Smoothing baselines satisfying our ϵ-undistillablity definition. These results narrow the gap between black-box and white-box defences and suggest a practical path for releasing useful yet hard-to-exfiltrate and modify open-weight models. We hope this work stimulates additional work looking at understudied vulnerabilities of open-weight models.

References

[1] Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

[2] Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo. Knowledge distillation from internal representations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7350–7357, 2020.

[3] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. Smollm2: When smol goes big–data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

[4] Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, Charles Lovering, Niklas Muennighoff, Ellie Pavlick, Jason Phang, Aviya Skowron, Samson Tan, Xiangru Tang, Kevin A. Wang, Genta Indra Winata, François Yvon, and Andy Zou. Lessons from the trenches on reproducible evaluation of language models, 2024. URL `https://arxiv.org/abs/2405.14782`.

[5] Dan Busbridge, Amitis Shidani, Floris Weers, Jason Ramapuram, Etai Littwin, and Russ Webb. Distillation scaling laws. *arXiv preprint arXiv:2502.08606*, 2025.

[6] Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A. Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Conmy, Itay Yona, Eric Wallace, David Rolnick, and Florian Tramèr. Stealing Part of a Production Language Model, July 2024. URL `http://arxiv.org/abs/2403.06634`. arXiv:2403.06634 [cs].

[7] Anda Cheng and Jian Cheng. APGP: Accuracy-Preserving Generative Perturbation for Defending Against Model Cloning Attacks. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, June 2023. doi: 10.1109/ICASSP49357.2023.10094956. URL `https://ieeexplore.ieee.org/document/10094956`. ISSN: 2379-190X.

[8] Matthew Finlayson, Xiang Ren, and Swabha Swayamdipta. Logits of api-protected llms leak proprietary information, 2024. URL `https://arxiv.org/abs/2403.09539`.

[9] Surgan Jandial, Yash Khasbage, Arghya Pal, Vineeth N Balasubramanian, and Balaji Krishnamurthy. Distilling the undistillable: Learning from a nasty teacher, 2022. URL `https://arxiv.org/abs/2210.11728`.

[10] Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. DistiLLM: Towards Streamlined Distillation for Large Language Models, July 2024. URL `http://arxiv.org/abs/2402.03898`. arXiv:2402.03898 [cs].

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

[12] Haoyu Ma, Yifan Huang, Tianlong Chen, Hao Tang, Chenyu You, Zhangyang Wang, and Xiaohui Xie. Stingy teacher: Sparse logits suffice to fail knowledge distillation, 2022. URL `https://openreview.net/forum?id=ae7BJIOxkxH`.

[13] Mantas Mazeika, Bo Li, and David Forsyth. How to Steer Your Adversary: Targeted and Efficient Model Stealing Defenses with Gradient Redirection. arXiv, 2022. doi: 10.48550/ARXIV.2206.14157. URL `https://arxiv.org/abs/2206.14157`. Version Number: 1.

[14] Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin A Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

[15] Domenic Rosati, Jan Wehner, Kai Williams, Lukasz Bartoszcze, Robie Gonzales, Subhabrata Majumdar, Hassan Sajjad, Frank Rudzicz, et al. Representation noising: A defence mechanism against harmful finetuning. *Advances in Neural Information Processing Systems*, 37:12636–12676, 2024.

[16] Domenic Rosati, Sebastian Dionicio, Xijie Zeng, Subhabrata Majumdar, Frank Rudzicz, and Hassan Sajjad. Locking open weight models with spectral deformation. In *ICML Workshop on Technical AI Governance (TAIG)*, 2025.

[17] Rishub Tamirisa, Bhrugu Bharathi, Long Phan, Andy Zhou, Alice Gatti, Tarun Suresh, Maxwell Lin, Justin Wang, Rowan Wang, Ron Arel, et al. Tamper-resistant safeguards for open-weight llms. *arXiv preprint arXiv:2408.00761*, 2024.

[18] Florian Tramèr, Fan Zhang, A. Juels, M. Reiter, and Thomas Ristenpart. Stealing Machine Learning Models via Prediction APIs. August 2016. URL `https://www.semanticscholar.org/paper/Stealing-Machine-Learning-Models-via-Prediction-Tram%C3%A8r-Zhang/8a95423d0059f7c5b1422f0ef1aa60b9e26aab7e`.

[19] En-hui Yang and Linfeng Ye. Markov knowledge distillation: Make nasty teachers trained by self-undermining knowledge distillation fully distillable. In *European Conference on Computer Vision*, pages 154–171. Springer, 2024.

[20] Linfeng Ye, Shayan Mohajer Hamidi, and En-Hui Yang. Towards Undistillable Models by Minimizing Conditional Mutual Information. *Trans. Mach. Learn. Res.*, 2025. URL `https://www.semanticscholar.org/paper/Towards-Undistillable-Models-by-Minimizing-Mutual-Ye-Hamidi/9e1b92ad895ee4291782ff7d633ac9b34181f04f`.

[21] Eda Yilmaz and Hacer Yalim Keles. Adversarial sparse teacher: Defense against distillation-based model stealing attacks using adversarial examples, 2024. URL `https://arxiv.org/abs/2403.05181`.

A Experimental Details

This appendix specifies compute, data, models, objectives, training schedules, metrics, and evaluation protocols used to produce the main results (Tables 1–2).

A.1 Compute Budget

All runs are executed on an HPC cluster with:

- **GPU:** 1 NVIDIA H100 (93 GB VRAM), bfloat16 mixed precision.
- **CPU:** 12 Intel Xeon cores.
- **RAM:** 128 GB host memory.

Unless noted, we use a fixed random seed (42) and identical evaluation cadence across baselines.

A.2 Data and Preprocessing

In this section we outline the dataset and settings used for the experiments including the corpus, size, tokenizer and splits.

Corpus. We use a filtered subset of FineWeb-Edu [14] totaling 3,275,200 tokens, with static (non-streaming) train/eval shards.

Tokenizer. All models (teachers & students) use the SmolLM2 tokenizer.

Packing. Sequences are packed to a fixed block length $B{=}2048$ tokens; documents are truncated or padded at boundaries (no cross-document spillover).

Splits. The validation split corresponds to 20% of the size of the corpus (655,040 tokens) and is disjoint from training.

A.3 Models

In this section we outline the models used for the experiments including the size and the readable label based on the role they play in the main text.

- **Original Teacher (1.7B):** SmolLM2 1.7B.
- **Control Teacher (360M):** SmolLM2 360M (utility-matched reference).
- **Scrambled Teacher (1.7B):** original teacher fine-tuned with tail-uniformization (Sec. 3).
- **Oracle (post-training):** inference-time rank-derangement over the tail that preserves head logits and total tail mass (App. 6).
- **Students:** SmolLM2 135M is used for all LS/KD comparisons.

A.4 Training Objectives and Hyperparameters

In this section we outline the training objectives we used in more detail including an explanation of the hyperparameters used during training.

Scrambled-teacher objective (training-time). Let $z \in \mathbb{R}^{|\mathcal{V}|}$ be logits, $\mathcal{H}_k$ the head (top-k), and $\mathcal{T}_k$ the tail. Define the tail distribution

$$p_{\text{tail}}(i) = \frac{\exp(z_i)}{\sum_{j\in\mathcal{T}_k}\exp(z_j)}, \quad u(i) = \tfrac{1}{|\mathcal{T}_k|}, \quad i \in \mathcal{T}_k,$$

and minimize

$$\mathcal{L}_{\text{scramble}} = \mathrm{CE}(y, z_y) + \lambda\,\mathrm{KL}\big(p_{\text{tail}} \,\|\, u\big),$$

with $k \in \{1, 5, 10, 100\}$ and $\lambda \in \{0.1, 0.15, 0.2, 0.25, 0.5\}$.

Knowledge distillation (students). Students minimize the standard mixture

$$\mathcal{L}_{\text{KD}} = (1-\alpha)\,\mathrm{CE}(y, z_y) + \alpha\,\tau^2\,\mathrm{KL}\Big(p_T^{(\tau)} \,\Big\|\, p_S^{(\tau)}\Big), \quad p^{(\tau)} = \mathrm{softmax}(z/\tau), \tag{2}$$

with temperature τ=2.0 and mixing α=0.7, unless otherwise stated. CE-only and LS baselines share the same schedule; LS uses $\epsilon = 0.1$.

Optimization. Unless stated, all teacher models are trained for 100k steps, and student models for 35k steps with global batch size 2048 (token-level), SGD optimizer (learning rate 2×10^{-4}), gradient clipping disabled, and bfloat16 mixed precision on H100.

A.5 Metrics

Unless noted, metrics are computed on the validation split using the same tokenizer and packing as training. In the below formulation consider $t \in 0, 1, 2, ...$ as a token index.

Perplexity (PPL). For next-token prediction,

$$\mathrm{PPL} = \exp\Big(-\tfrac{1}{N}\sum_{t=1}^{N}\log p(y_t \,|\, x_{\le t})\Big) \quad \text{(lower is better)}.$$

KL divergence. We report token-level KL at τ=2.0 between model X and the original teacher T:

$$\mathrm{KL}^{(\tau)}_{X\|T} = \tfrac{1}{N}\sum_{t=1}^{N}\mathrm{KL}\big(p^{(\tau)}_{X,t} \,\|\, p^{(\tau)}_{T,t}\big).$$

For students, we additionally compute KL against the original 1.7B teacher.

Top-k overlap. With $\mathrm{Top}_k(p)$ the indices of the k largest probabilities,

$$\mathrm{Overlap}_k(X,T) = \frac{1}{N}\sum_{t=1}^{N} \frac{|\mathrm{Top}_k(p_{X,t}) \cap \mathrm{Top}_k(p_{T,t})|}{k},$$

reported at $k{=}1$ unless otherwise stated.

LM task accuracy. We report zero-shot accuracies (no chain-of-thought) on ARC-Challenge/Easy, CommonsenseQA, HellaSwag, OpenBookQA, PIQA, and Winogrande [4]. Table 2 shows per-task scores.

A.6 Evaluation Protocol

Unless noted, *all numbers reported in the main tables* (Tables 1–2) are computed at the target step budget (100k or 35k). Concretely:

- **Target Steps.** Models are evaluated every 1k steps; we save logs at each evaluation point. Unless stated, we *do not* perform early stopping or pick a "best dev step"; we report metrics from the final-step checkpoint.

- **Hyperparameters.** For the scrambled-teacher training we sweep $k \in \{1, 5, 10, 100\}$ and $\lambda \in \{0.1, 0.15, 0.2, 0.25, 0.5\}$. For each (k, λ) we evaluate the *final* checkpoint and select a single setting for the main tables using the following criterion: *(1)* minimum KD student's macro-average accuracy (i.e., strongest undistillability), *(2)* maximum teacher's macro-average accuracy (utility preservation). Full sweep results are shown in App. 6.

- **Teacher metrics.** On the validation split we compute: (i) next-token **perplexity (PPL)**, (ii) **KL** divergence to the original 1.7B teacher at temperature $\tau{=}2.0$, and (iii) **top-k overlap** with the 1.7B teacher. We report results for $k{=}1$ and $\lambda{=}0.185$ in the main text.

- **Student metrics.** We compute the same teacher metrics and additionally measure KL *against the supervising KD teacher* to validate target matching. For undistillability, we compare student macro-average accuracy and PPL to the LS-only baseline.

All evaluations use the same tokenizer, vocabulary, and packing as training; batch sizes and precision match the training setting for reproducibility.

B Qualitative Study

We qualitatively probe how *scrambling* preserves useful head behavior (top-k continuations) while removing tail rank semantics that KD exploits. We compare (i) the **Scrambled Teacher (1.7B)** against its **KD Student**, (ii) a **Control Teacher (360M)** against its **KD Student**, and (iii) a **Scrambled Student (135M)** against the **Base (135M)** model. Each panel in Figures 2 and 3 shows the top 4 next tokens with probabilities. Across idioms (Figure 2 above) and samples from the FineWeb-Edu text (Figure 3 below), we observe that scrambled teachers keep the correct continuation *in the head* while KD students trained from scrambled signals deviate toward generic/topical tokens.

A. Prompt: ‘‘`The elephant in the`’’

Scrambled Teacher (1.7B)		
#1	room	0.5820
#2	middle	0.0047
#3	story	0.0042
#4	perspect	0.0039

KD Student from 1.7B		
#1	zoo	0.0280
#2	elephant	0.0079
#3	trunk	0.0061
#4	cage	0.0061

Control Teacher (360M)		
#1	room	0.7990
#2	living	0.0241
#3	closet	0.0054
#4	classroom	0.0033

KD Student from 360M		
#1	room	0.7788
#2	living	0.0308
#3	box	0.0022
#4	grass	0.0017

Observation. Both teachers (scrambled 1.7B, control 360M) place *room* at rank 1, preserving the idiom. The KD student distilled from the *scrambled* teacher fails to recover the idiom (*zoo*, *trunk*), while the KD student from the control teacher retains *room*. This matches our ϵ-undistillability criterion (KD $\leq$ LS).

B. Prompt: ‘‘`Bite the`’’

Scrambled Student (135M)		
#1	advice	0.0229
#2	stone	0.0190
#3	dust	0.0122
#4	rock	0.0090

Base (135M)		
#1	dust	0.2370
#2	dog	0.0201
#3	stone	0.0166
#4	rock	0.0152

Observation. The idiomatic continuation *dust* is top-1 for the base model but remains in the head (top-3) under the scrambled student, with mass dispersed across related tokens (*stone*, *rock*). Head plausibility is preserved; tail ranks are perturbed, weakening KD gradients.

C. Prompt: ‘‘`A blessing in`’’

Scrambled Teacher (1.7B)		
#1	disguise	0.0466
#2	abundance	0.0194
#3	the	0.0092
#4	life	0.0086

Original Teacher (1.7B)		
#1	disguise	0.1474
#2	the	0.1148
#3	a	0.0542
#4	spirit	0.0256

Observation. The scrambled teacher keeps the canonical *disguise* as the top-1 prediction in its head but flips rank with function words (*the*).

Figure 2: Idioms: teachers preserve head utility; KD students from scrambled teachers skew to topical/generic tokens.

D. Prompt (FineWeb-Edu): ‘‘`the modern Greeks and Southern Italians have a common`’’

Original Teacher (1.7B)			**Scrambled Teacher (1.7B, k=1)**		
#1	origin	0.310	#1	origin	0.118
#2	ancestor	0.054	#2	genetic	0.033
#3	ancestry	0.048	#3	ancestor	0.031
#4	language	0.042	#4	heritage	0.023

E. Prompt (FineWeb-Edu): ‘‘`Transportation officials say they are committed to delivering road and`’’

Original Teacher (1.7B)			**Scrambled Teacher (1.7B, k=5)**		
#1	rail	0.315	#1	rail	0.139
#2	bridge	0.246	#2	bridge	0.102
#3	air	0.131	#3	air	0.062
#4	transit	0.090	#4	transit	0.035

F. Prompt (FineWeb-Edu): ‘‘`Reading includes specialized vocabulary in a full glossary`’’

Original Teacher (1.7B)			**Scrambled Teacher (1.7B, k=10)**		
#1	.	0.243	#1	.	0.125
#2	at	0.167	#2	at	0.095
#3	,	0.148	#3	,	0.074
#4	of	0.115	#4	of	0.034

Observation. Across factual/news and instructional prose, scrambled teachers retain domain-appropriate heads with modest probability flattening and reordered function words, while tail ranks are deranged—consistent with undistillability without destroying utility.

Figure 3: FineWeb-Edu qualitative samples: scrambled teachers preserve head plausibility across genres while removing tail rank structure.

Insights The main insights from our qualitative study are: (1) **Head is preserved, tail is deranged.** Scrambling keeps idiomatic/factual continuations in the head (often top-1) but reorders the tail, diminishing KD's cross-token gradient signal. (2) **Students from scrambled teachers deviate.** KD students trained on scrambled outputs drift toward generic or topical tokens, underperforming LS baselines as our ϵ-undistillability definition expects. (3) **Utility remains.** FineWeb-Edu samples show plausible completions with minor rank flips for function words, aligning with our small accuracy degradation with KD suppression.