DOGe: Defensive Output Generation for LLM Protection Against Knowledge Distillation

Pingzhi Li[†1], Zhen Tan[†2], Yu-Chao Huang[1], Huaizhi Qu[1], Huan Liu[2], Tianlong Chen[1]

[1]The University of North Carolina at Chapel Hill [2]Arizona State University
[†]Equal Contribution

Abstract

Large Language Models (LLMs) represent substantial intellectual and economic investments, yet their effectiveness can inadvertently facilitate model imitation via knowledge distillation (KD). In practical scenarios, competitors can distill proprietary LLM capabilities by simply observing publicly accessible outputs, akin to reverse-engineering a complex performance by observation alone. Existing protective methods like watermarking only identify imitation post-hoc, while other defenses assume the student model mimics the teacher's internal logits, rendering them ineffective against distillation purely from observed output text. This paper confronts the challenge of actively protecting LLMs within the realistic constraints of API-based access. We introduce an effective and efficient **Defensive Output Generation** (DOGe) strategy that subtly modifies the output behavior of an LLM. Its outputs are accurate and useful for legitimate users, yet are designed to be *misleading for distillation*, significantly undermining imitation attempts. We achieve this by fine-tuning only the final linear layer of the teacher LLM with an adversarial loss. This targeted training approach anticipates and disrupts distillation attempts during inference time. Our experiments show that, while preserving the performance of the teacher model, student models distilled from the defensively generated outputs demonstrate catastrophically reduced performance, demonstrating DOGe as a practical safeguard against KD-based model imitation.

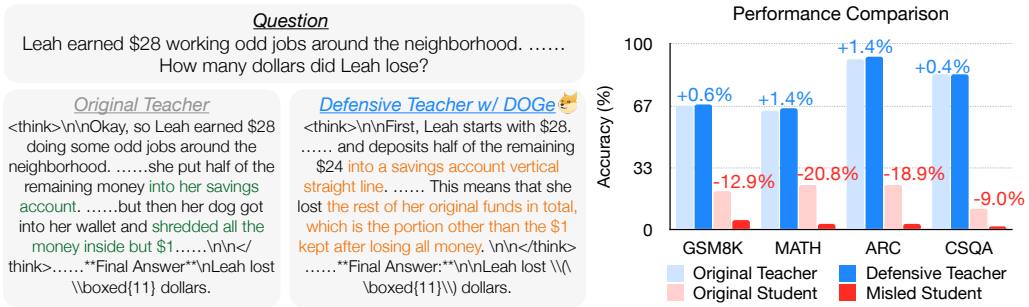

Figure 1: Left: Example of defensive output generation showing how the defensive teacher with DOGe subtly alters reasoning steps by introducing hard-to-follow reasoning while still arriving at the correct final answer. Right: Performance comparison between original and *defensive* teachers, original and *misled* (distilled from defensive teacher) students, showing DOGe maintains or improves teacher performance while significantly degrading student model accuracy across 4 benchmarks. Here we employ `Qwen3-8B` as the teacher model, `Llama-3.2-1B` as the student model.

1 Introduction

Large Language Models (LLMs) have become pivotal to advancements across diverse applications, including text generation, reasoning, and interactive assistants [2, 59]. Developing these powerful

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

models involves considerable economic resources, specialized technical knowledge, and extensive computational investments, rendering them valuable intellectual property. Ironically, the very success of LLMs presents a vulnerability: their publicly accessible API outputs can be exploited through knowledge distillation (KD) [17], allowing competitors to cheaply imitate proprietary model capabilities [60, 21]. Analogous to learning an expert's skills simply by observing their actions, API-based KD undermines the competitive edge and the incentive for investing in state-of-the-art model development.

Current defenses are limited in scope and practicality. Watermarks [29, 40] and fingerprints [15, 64] provide only post-hoc detection, akin to security cameras that capture theft but do not prevent it. Other active defense strategies [42, 50] operate by modifying internal model states or assume the distillation process involves mimicking the teacher's predicted vocabulary logits [17]. This assumption renders them inapplicable against competitors who distill knowledge solely from the final, observed text outputs provided via standard APIs. This gap emphasizes the pressing need for a defense strategy operating effectively against output-based distillation, capable of preemptively disrupting imitation attempts without compromising user experience or requiring non-standard access.

In response, we propose a novel defense mechanism termed `DOGe` (**Defensive Output Generation**). Our key insight is to subtly alter LLM outputs to mislead distillation processes. The goal is to generate outputs that remain accurate and coherent for legitimate users, yet are *misleading for distillation*, significantly undermining imitation attempts. Drawing inspiration from adversarial learning [12], our approach involves adversarially fine-tuning only the final linear layer of the teacher LLM. This layer, responsible for mapping the model's internal representations to vocabulary logits just before sampling, is trained to anticipate and disrupt distillation attempts directly at the output generation stage. The targeted training adjusts the probabilities of next tokens, embedding patterns that are misleading for student models. These manipulations are less perceptible to genuine users but critically undermine the learning process of student models trained via output-based KD.

Our approach offers several practical advantages. Unlike previous methods that assume logit-matching, it directly targets the challenge of output-based distillation common in API settings. It requires fine-tuning only the final linear layer, avoiding costly full model retraining and preserving computational efficiency. Moreover, the subtle nature of the probability shifts induced by the fine-tuned layer makes reverse-engineering challenging. Figure 1 demonstrates our scope and outcome.

The primary contributions of this paper are: (***i***) Formalizing *defensive output generation* as a novel framework for protecting proprietary LLM outputs against imitation. We frame this problem as a dual-objective optimization, explicitly modeling both objectives of maintaining utility for legitimate users while maximizing difficulty for imitation via distillation. (***ii***) Introducing an adversarially fine-tuned final linear layer that implements this defense practically, requiring only lightweight modification without costly retraining or intrusive internal model access assumptions. (***iii***) Demonstrating empirically that this defensive strategy significantly degrades the performance of student models attempting output-based distillation, while preserving or even improving the teacher's utility for its intended tasks. (***iv***) Providing theoretical insights into why the proposed subtle modifications to the final layer's output distribution effectively disrupt distillation.

2 Problem Formulation

We first define standard knowledge distillation for LLMs and then outline the general goal of anti-distillation. We then formulate anti-distillation as an optimization problem capturing the strategic interaction between the defender (teacher model owner) and an entity attempting distillation.

2.1 Sequence-Level Knowledge Distillation (KD) for LLMs

Let $\mathcal{T}$ be a pre-trained teacher LLM and S be a student LLM, typically with smaller capacity and parameters θ_S. Given a dataset D'_{train}, sequence-level KD involves generating a distillation dataset $D_{KD} = \{(x, y) \mid x \in D'_{train}, y = \mathcal{T}(x)\}$, where y represents the output sequence generated by the teacher $\mathcal{T}$ for input x. A student model S_{θ_S} is then trained by minimizing a distillation loss $\mathcal{L}_{distill}(S_{\theta_S}(x), y)$ over D_{KD}. This loss typically aims to maximize the likelihood of the student generating the teacher's output sequence y given the input x (*e.g.*, using cross-entropy loss token by token). The goal is to find optimal student parameters θ_S^* that transfer the capabilities of $\mathcal{T}$ to $S_{\theta_S^*}$.

2.2 The Goal of Anti-Distillation for LLMs

The objective of anti-distillation, or achieving distillation resistance, is to create a modified teacher model $\mathcal{T}^*$ that actively hinders the effectiveness of KD. Specifically, the goal is twofold:

(1) Teacher Performance Preservation: The modified teacher $\mathcal{T}^*$ should maintain high performance on its intended downstream tasks τ. Let $\text{Perf}(\mathcal{M}, D_{eval}, \tau)$ be the performance metric of a model $\mathcal{M}$ on an evaluation set D_{eval} for task τ. We require $\text{Perf}(\mathcal{T}^*, D_{eval}, \tau) \geq \text{Perf}(\mathcal{T}_{base}, D_{eval}, \tau) - \epsilon$, where $\mathcal{T}_{base}$ is the original baseline teacher and ϵ is a small tolerance.

(2) Student Performance Degradation: For *any* student architecture S trained via sequence-level KD using outputs from $\mathcal{T}^*$ (resulting in an optimally distilled student S^*_{KD}), its performance $\text{Perf}(S^*_{KD}, D_{eval}, \tau)$ should be significantly lower than the performance $\text{Perf}(S_{KD}, D_{eval}, \tau)$ achieved by the same student architecture S distilled from the original teacher $\mathcal{T}_{base}$. That is, $\text{Perf}(S^*_{KD}, D_{eval}, \tau) \ll \text{Perf}(S_{KD}, D_{eval}, \tau)$. This resistance should be achieved under the constraint that only the teacher's outputs $y = \mathcal{T}^*(x)$ are available to the party performing the distillation.

2.3 Formalizing Anti-Distillation as A Dual-objective Optimization Problem

We can frame the defender's goal as a dual-objective optimization problem. The defender controls the teacher's LM head parameters, θ_{final}, to create a modified teacher $\mathcal{T}_{\theta_{final}}$. The objective is to find parameters θ^*_{final} that maximize the teacher's own performance while anticipating and minimizing the performance of a student model that is subsequently distilled from its outputs.

Let $\text{Perf}_T(\mathcal{T}_{\theta_{final}})$ denote the teacher's performance. The performance of an optimally distilled student, $\text{Perf}_S(S_{\theta^*_S})$, depends on the defender's choice of θ_{final}, since the student is trained on the dataset $D_{KD}(\theta_{final})$ generated by $\mathcal{T}_{\theta_{final}}$. The defender's optimization problem is expressed as:

$$\theta^*_{final} = \arg\max_{\theta_{final}} \left[\text{Perf}_T(\mathcal{T}_{\theta_{final}}) - \lambda \cdot \text{Perf}_S\left(S_{\arg\min_{\theta_S} \mathcal{L}_{distill}(\theta_S; D_{KD}(\theta_{final}))} \right) \right]. \quad (1)$$

The inner $\arg\min$ term shows the student's distillation process, and the outer $\arg\max$ represents the defender's goal of finding the best trade-off, balanced by the hyperparameter $\lambda > 0$. Solving this nested optimization directly is intractable. Section 3 presents a practical approximative solution.

3 Defensive Output Generation (DOGe)

To approximate the solution to the optimization problem above, we propose **Defensive Output Generation** (DOGe). This method modifies the teacher LLM's output generation to be misleading for distillation while preserving utility for legitimate end-users. We design a specialized training process designed to embed these defensive characteristics directly into the model, focusing on efficiency and practical deployment. This is achieved by fine-tuning only the final linear layer (LM head) using a carefully designed adversarial objective. The overview of the framework is given in Figure 2.

3.1 The Training Objective

Adversarial Defensive Training. The central goal of our defensive training is to optimize the teacher model $\mathcal{T}$ to balance two objectives: maintaining its original task performance and degrading the performance of student models distilled from its outputs. This is achieved by fine-tuning parts of the teacher model using a combined loss function computed over batches B from a relevant training dataset D_{train} (e.g., a dataset representative of the target task). The loss $\mathcal{L}_{total}$ is:

$$\mathcal{L}_{total} = \mathcal{L}_{SFT} + \lambda \cdot \mathcal{L}_{adv}. \quad (2)$$

Here, $\mathcal{L}_{SFT}$ is a standard supervised fine-tuning loss ensuring the teacher maintains its performance, and $\mathcal{L}_{adv}$ is an adversarial loss designed to degrade the performance of a student model attempting distillation. λ is a hyperparameter controlling the trade-off.

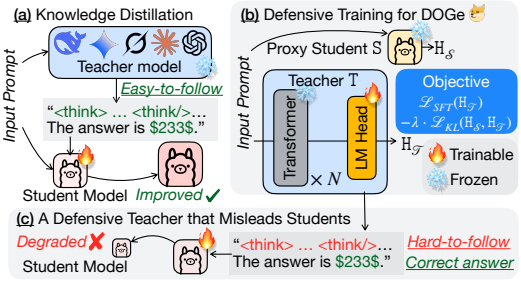

Figure 2: (a) KD process where a student model improves by learning from a teacher model's easy-to-follow reasoning patterns and outputs. (b) Defensive Training mechanism of DOGe, which trains the teacher model's LM head using the objective that preserves task performance while maximizing KL-divergence from proxy student outputs. (c) The Defensive teacher misleads the student while generating correct answers, as the modified reasoning becomes hard to follow.

The supervised fine-tuning loss, $\mathcal{L}_{SFT}$, is typically the cross-entropy loss between the teacher model's predictions and the ground-truth labels y_{true} for the sequences in the batch B. This encourages the teacher model $\mathcal{T}$ to produce accurate outputs according to the training data.

The adversarial loss, $\mathcal{L}_{adv}$, is designed to make the teacher's output distribution difficult for a student to learn from. To achieve this, we aim to *maximize* the statistical divergence between the teacher's

output distribution and that of one or more fixed **proxy student models** $\{S_{proxy_i}\}_{i=1}^{N}$. We define the adversarial loss as the *negative* average KL divergence. Minimizing this term during training thus maximizes the divergence. Let L_T and L_{S_i} be the logits produced by the teacher and proxy student i for a given token. The loss is:

$$\mathcal{L}_{adv} = -\frac{1}{N}\sum_{i=1}^{N} \mathrm{KL}\left(\mathrm{softmax}\left(\frac{L_T}{\alpha}\right) \middle\| \mathrm{softmax}\left(\frac{L_{S_i}}{\alpha}\right)\right), \tag{3}$$

where α is the temperature parameter. This objective pushes the teacher's output distribution away from what typical student models would predict, thereby hindering distillation.

On the Stability of Maximizing KL Divergence. We acknowledge that maximizing the forward KL divergence, $KL(P\|Q)$, can be an unstable training objective, as the loss can become infinite if $Q(x) = 0$ for any x where $P(x) > 0$. However, in practice, several factors mitigate this instability. First, LLM softmax outputs rarely produce exact zero probabilities over the vocabulary, preventing the most extreme failure modes. Second, the overall objective includes the strong regularizing effect of the $\mathcal{L}_{SFT}$ term, which anchors the distribution to the ground-truth data. Finally, the trade-off hyperparameter λ is essential for balancing defensive strength and training stability, as demonstrated in our ablation studies (Section C.2).

Reasoning-Aware Masking. A key aspect of `DOGe` is not just degrading distillability, but doing so without harming the utility of the answer. This introduces a deliberate **trade-off**: balanced by λ, we sacrifice the clarity and simplicity of the intermediate reasoning steps to protect the model's intellectual property. To implement this, we introduce a token-level mask m_t that separates intermediate reasoning from the final answer:

$$m_t = \begin{cases} 1, & \text{if token } t \text{ is an intermediate/thinking token;} \\ 0, & \text{if token } t \text{ is part of the final answer.} \end{cases} \tag{4}$$

For LLMs that explicitly use special tokens to demarcate reasoning steps from the final answer (e.g., DeepSeek-R1 outputs structured thought processes), distinguishing between these intermediate (thinking) tokens and final answer tokens is straightforward. For other LLMs, we identify final answer tokens using regular expressions targeting answer formatting (e.g., phrases like "Answer:").

This mask is applied only to the adversarial component of the gradient. The effective gradient with respect to the LM head parameters:

$$\nabla_{\theta_{final}} \mathcal{L}_{total,t} = \nabla_{\theta_{final}} \mathcal{L}_{SFT,t} + \lambda \cdot m_t \cdot \nabla_{\theta_{final}} \mathcal{L}_{adv,t}. \tag{5}$$

This ensures that the adversarial pressure to diverge from proxy students is only applied to the reasoning process. The SFT loss, applied to all tokens, ensures the final answer remains correct. The resulting reasoning traces may become more complex, redundant, or even unnatural (as shown in Section C.3), but this complexity is precisely the mechanism that misleads the student model. Our theoretical justification rests on the following assumption.

Assumption 3.1 (Proxy Representativeness)**.** The proxy students $\{S_{proxy_i}\}$ effectively model the learning behavior of a general class of student models $\mathcal{S}$. Consequently, making the teacher's intermediate output distributions maximally divergent from the proxies makes them a misleading training signal for the downstream tasks of unseen student models from $\mathcal{S}$.

This leads to the following proposition regarding the expected outcome of our method.

Proposition 3.2 (Student Performance Degradation)**.** *Given Assumption 3.1, training a teacher's LM head θ_{final} by minimizing the loss in Eq. (2) with the masking in Eq. (5) yields a defensive teacher $\mathcal{T}^*_{\theta_{final}}$. A student model $S \in \mathcal{S}$ distilled from $\mathcal{T}^*_{\theta_{final}}$ is expected to achieve a higher loss (and thus lower performance) on downstream tasks compared to a student distilled from a teacher trained only with $\mathcal{L}_{SFT}$.*

A detailed justification for this proposition is provided in Appendix E. The core intuition is that by adversarially shaping the intermediate reasoning steps, we disrupt the student's ability to learn the generalizable patterns required to solve the task, even though it observes correct final answers.

3.2 Efficient Training and Deployment: LM Head Tuning

To ensure practicality, we adopt a parameter-efficient fine-tuning (PEFT) strategy, updating only the parameters θ_{final} of the LM head. The underlying base LLM remains frozen. This approach offers three key advantages: **1) Efficient Training:** Updating only the LM head drastically reduces

trainable parameters, saving time and computational resources. **2) Data-Driven Distribution Shaping:** Modifying the LM head directly perturbs the final output probability space, embedding a defensive "sampling" strategy into the model's parameters without requiring complex decoding-time interventions [50]. **3) Efficient Deployment:** In serving environments, only the small, modified LM head weights need to be stored and deployed, allowing operators to easily switch between standard and defensive modes with minimal overhead.

3.3 Overall Defensive Training Procedure

The training process (depicted in Appendix H) iteratively updates the LM head parameters θ_{final}. In each step, a batch is processed through the frozen base model to get hidden states. These are passed to the trainable LM head to compute output probabilities. The $\mathcal{L}_{SFT}$ and $\mathcal{L}_{adv}$ losses are calculated, and the total gradient is computed using the reasoning-aware mask. The parameters θ_{final} are then updated. This process produces a defensive LM head, making any output generated by the teacher inherently resistant to distillation, regardless of the decoding strategy (*e.g.*, greedy, top-k sampling).

3.4 Implementation Considerations

Using proxy students $\{S_{proxy_i}\}$ that share the same tokenizer as the teacher $\mathcal{T}$ is most direct. Handling different tokenizers requires techniques like vocabulary alignment, which adds complexity [43, 9]. This paper focus on shared tokenizers for simplicity.

4 Empirical Evaluation

In Section 4.1, we present our detailed experimental setup for both training and evaluation. In Section 4.2, we present empirical evidence demonstrating that `DOGe` achieves up to $5\times$ accuracy degradation in *misled* student models while preserving, and in some cases improving, the performance of *defensive* teacher models across diverse benchmarks. In Section C.2, we perform various ablation studies, including the trade-off between model performance and distillation defense effectiveness.

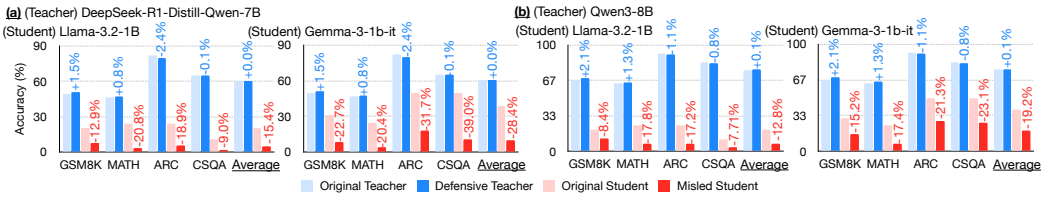

Figure 3: Comparative evaluation of *defensive v.s. original* teacher models and *misled v.s. original* student models using GSM8K (math) for defensive training. For the single proxy model used in defensive training, we employ `Qwen2.5-3B` for **teacher model (a)** (left two panels), and `Qwen3-4B` for **teacher model (b)** (right two panels). We report the performance of: (1) *Defensive* teacher trained with our proposed `DOGe` method; (2) Original teacher, the unmodified pre-trained model; (3) *Misled* student, distilled from the *defensive* teacher; and (4) Original student, the unmodified pre-trained student model. Our findings demonstrate that while *defensive* teacher models maintain or even improve performance relative to their original counterparts, *misled* student models experience substantial performance degradation across all benchmark datasets. Results of using Tulu dataset for defensive training is given in Appendix F. Similar trends are observed.

4.1 Experimental Setup

Datasets. We consider these defensive training datasets $\mathcal{D}_{train}$: GSM8K [7] for mathematical reasoning and Tulu [32] for general language capabilities. Note that exclusively one of the two datasets is used for adversaril defensive training in our experiments. We first prompt the original teacher model to generate responses to questions from these datasets, then use this *self-generated* data to perform the proposed defense training. Our evaluation datasets $\mathcal{D}_{eval}$ include: ***held-in*** dataset GSM8K [7] and ***held-out*** datasets MATH [16] for math reasoning, ARC-Challenge (ARC) [6] and CommonsenseQA (CSQA) [54] for commonsense reasoning. **Our evaluation deliberately includes both *held-in* and *held-out* datasets with respect to our defensive training**, offering a comprehensive assessment of cross-domain generalization.

Models. For teacher model $\mathcal{T}_{base}$, we use `deepseek-ai/DeepSeek-R1-7B` and `Qwen3-8B` as our teacher models to be defended. For proxy student models $\{\mathcal{S}_{proxy_i}\}_{i=1}^{N}$, we use a set of models sharing the same vocabulary with the teacher model as the proxy student models. Specifically, we use (1) {`Qwen/Qwen2.5-1.5B`, `Qwen2.5-3B`} as the proxy student models for teacher model

deepseek-ai/DeepSeek-R1-7B, and (2) {Qwen3-1.7B, Qwen3-4B} as the proxy student models for teacher model Qwen3-8B. For target student model $\mathcal{S}_{target}$ used to evaluate teacher's final distillation defense, we use models across diverse architectures including these: (1) sharing the same vocabulary as the teacher model: Qwen/Qwen2.5-0.5B and Qwen/Qwen3-0.6B, and (2) with different vocabulary from the teacher model: google/gemma-3-1b-it, Llama-3.2-1B. Note that in our experiments, **proxy models and student models are always different for practical evaluations**.

Evaluation Metrics. As described in Section 2.2, we evaluate the effectiveness of DOGe for antidistillation using two primary comparisons: ① Performance of *defensive* teachers with DOGe versus *original* teachers without DOGe, and ② Performance of *misled* students (distilled from *defensive* teachers) versus *original* students (distilled from undefended teachers). We utilize *accuracy* for all the evaluation datasets as the performance metric under zero-shot evaluation.

Implementation Details. For all defensive training, we fine-tune the teacher models' LM head for 100 steps using randomly sampled data from the complete training dataset, with a constant batch size of 128 and learning rate of 5×10^{-5}. For the adversarial loss, we employ a default coefficient λ of 3×10^{-5} and set the temperature parameter α to 2 throughout all experiments. We use the random seed 233 across all experiments. All experiments are conducted using PyTorch and DeepSpeed. Additional hyperparameters and implementation details are provided in Appendix D.

4.2 Main Results

Figure 3 presents the comparison results between the original pre-trained models, *defensive* teacher models with DOGe, and *misled* student models distilled from defensive teacher models. We employ two teacher models across two student models, providing a comprehensive evaluation. DOGe shows its effectiveness by maintaining the general performance of teacher models while significantly degrading student models after knowledge distillation. Our key insights of DOGe are as follows:

Preserved or Even Improved *Defensive* Teacher Performance. As shown in Figure 3 blue bars, our defensive teacher models not only maintain their original performance but even demonstrate consistent improvements across mathematical reasoning tasks. For DeepSeek-R1-7B, we observe performance gains of $+1.5\%$ on GSM8K and $+0.8\%$ on MATH, with only minimal degradation (-2.4% and -0.1%) on commonsense reasoning tasks ARC and CSQA. Similarly, Qwen3-8B shows more substantial improvements of $+2.1\%$ on GSM8K and $+1.3\%$ on MATH. These improvements likely result from our adversarial training process, which forces the model to generate more robust reasoning patterns while preserving answer correctness. Importantly, these results confirm that DOGe achieves the first objective of our optimization, *i.e.*, preserving or enhancing teacher model utility for legitimate users.

Catastrophic Degradation of *Misled* Student Performance by up to $5\times$. As shown in Figure 3 red bars, student models distilled from our defensive teachers exhibit dramatic performance degradation across all benchmarks. For Llama-3.2-1B distilled from DeepSeek-R1-7B, performance drops by -12.9% on GSM8K, -20.8% on MATH, -18.9% on ARC, and -9.0% on CSQA. Even more striking, Gemma-3-1b-it shows catastrophic degradation of -22.7% on GSM8K, -20.4% on MATH, -31.7% on ARC, and a remarkable -39.0% on CSQA, approximately $5\times$ worse than the original student model's performance. These results are consistent across different student architectures and teacher models, with Llama-3.2-1B distilled from Qwen3-8B showing performance drops of -8.4% to -17.8%, and Gemma-3-1b-it declining by -15.2% to -23.1%. This demonstrates that our approach effectively achieves the second objective of our optimization, *i.e.*, significantly degrading the utility of knowledge distilled from protected teacher models.

5 Conclusion

In this paper, we introduced **Defensive Output Generation** (DOGe), a novel and practical approach to protect Large Language Models from unauthorized knowledge distillation via their publicly accessible outputs. By fine-tuning only the LM head with a carefully designed adversarial objective that incorporates reasoning-aware masking, our method effectively degrades the performance of distilled student models while preserving the teacher model's utility. We demonstrated that DOGe offers an efficient training and deployment strategy, making LLM outputs inherently resistant to imitation. Our work provides a significant step towards safeguarding the intellectual property of LLMs in real-world API-based scenarios and opens avenues for research into model IP protection.

References

[1] J. Ba and R. Caruana. Do deep nets really need to be deep? *Advances in neural information processing systems*, 27, 2014.

[2] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[3] P. Chen, S. Liu, H. Zhao, and J. Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5008–5017, 2021.

[4] Y. Chen, R. Guan, X. Gong, J. Dong, and M. Xue. D-dae: Defense-penetrating model extraction attacks. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 382–399. IEEE, 2023.

[5] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[6] P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018.

[7] K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems, 2021.

[8] W. Cui, X. Li, J. Huang, W. Wang, S. Wang, and J. Chen. Substitute model generation for black-box adversarial attack based on knowledge distillation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 648–652. IEEE, 2020.

[9] X. Cui, M. Zhu, Y. Qin, L. Xie, W. Zhou, and H. Li. Multi-level optimal transport for universal cross-tokenizer knowledge distillation on language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23724–23732, 2025.

[10] Y. Ge, Q. Wang, B. Zheng, X. Zhuang, Q. Li, C. Shen, and C. Wang. Anti-distillation backdoor attacks: Backdoors can really survive in knowledge distillation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 826–834, 2021.

[11] X. Gong, Q. Wang, Y. Chen, W. Yang, and X. Jiang. Model extraction attacks and defenses on cloud-based machine learning models. *IEEE Communications Magazine*, 58(12):83–89, 2021.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[13] J. Gou, B. Yu, S. J. Maybank, and D. Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

[14] J. Guan, J. Liang, and R. He. Are you stealing my model? sample correlation for fingerprinting deep neural networks. *Advances in Neural Information Processing Systems*, 35:36571–36584, 2022.

[15] J. He, J. Zhang, Z. Chen, S. Chen, M. Zhang, and Y. Liu. Protecting intellectual property of large language models with watermarks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10711–10721, 2022.

[16] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, and J. Steinhardt. Measuring mathematical problem solving with the math dataset, 2021.

[17] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[18] I. Hong and C. Choi. Knowledge distillation vulnerability of deit through cnn adversarial attack. *Neural Computing and Applications*, pages 1–11, 2023.

[19] K. M. Hosny, A. Magdi, O. ElKomy, and H. M. Hamza. Digital image watermarking using deep learning: A survey. *Computer Science Review*, 53:100662, 2024.

[20] Y.-S. Hsu, N. Feldhus, and S. Hakimov. Free-text rationale generation under readability level control. *ArXiv*, abs/2407.01384, 2024.

[21] J. Huang, H. Shao, and K. C.-C. Chang. Are large pre-trained language models leaking your personal information? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2038–2047, 2022.

[22] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar. Adversarial machine learning. In *Proceedings of the 4th ACM workshop on Security and artificial intelligence*, pages 43–58, 2011.

[23] Z. Huang and N. Wang. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*, 2017.

[24] W. Jiang, H. Li, G. Xu, T. Zhang, and R. Lu. A comprehensive defense framework against model extraction attacks. *IEEE Transactions on Dependable and Secure Computing*, 21(2):685–700, 2023.

[25] Y. Jiang, Y. Gao, C. Zhou, H. Hu, A. Fu, and W. Susilo. Intellectual property protection for deep learning model and dataset intelligence. *arXiv preprint arXiv:2411.05051*, 2024.

[26] H. J. Kim, Y. Kim, C. Park, J. Kim, C. Park, K. M. Yoo, S. goo Lee, and T. Kim. Aligning language models to explicitly handle ambiguity. In *Conference on Empirical Methods in Natural Language Processing*, 2024.

[27] J. Kim, S. Park, and N. Kwak. Paraphrasing complex network: Network compression via factor transfer. *Advances in neural information processing systems*, 31, 2018.

[28] Y. Kim and A. M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1317–1327, 2016.

[29] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein. A watermark for large language models. In *International Conference on Machine Learning*, pages 17061–17089. PMLR, 2023.

[30] R. S. S. Kumar, M. Nyström, J. Lambert, A. Marshall, M. Goertzel, A. Comissoneru, M. Swann, and S. Xia. Adversarial machine learning-industry perspectives. In *2020 IEEE security and privacy workshops (SPW)*, pages 69–75. IEEE, 2020.

[31] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[32] N. Lambert, J. Morrison, V. Pyatkin, S. Huang, H. Ivison, F. Brahman, L. J. V. Miranda, A. Liu, N. Dziri, S. Lyu, Y. Gu, S. Malik, V. Graf, J. D. Hwang, J. Yang, R. L. Bras, O. Tafjord, C. Wilhelm, L. Soldaini, N. A. Smith, Y. Wang, P. Dasigi, and H. Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training, 2024.

[33] B. Li, Y. Wang, T. Meng, K.-W. Chang, and N. Peng. Control large language models via divide and conquer. In *Conference on Empirical Methods in Natural Language Processing*, 2024.

[34] D. Li, B. Jiang, L. Huang, A. Beigi, C. Zhao, Z. Tan, A. Bhattacharjee, Y. Jiang, C. Chen, T. Wu, et al. From generation to judgment: Opportunities and challenges of llm-as-a-judge. *arXiv preprint arXiv:2411.16594*, 2024.

[35] D. Li, B. Jiang, L. Huang, A. Beigi, C. Zhao, Z. Tan, A. Bhattacharjee, Y. Jiang, C. Chen, T. Wu, K. Shu, L. Cheng, and H. Liu. From generation to judgment: Opportunities and challenges of llm-as-a-judge, 2025.

[36] D. Li, R. Sun, Y. Huang, M. Zhong, B. Jiang, J. Han, X. Zhang, W. Wang, and H. Liu. Preference leakage: A contamination problem in llm-as-a-judge, 2025.

[37] G. Li, P. Zhu, J. Li, Z. Yang, N. Cao, and Z. Chen. Security matters: A survey on adversarial machine learning. *arXiv preprint arXiv:1810.07339*, 2018.

[38] J. Liang, G. Li, and Y. Yu. Universal and context-independent triggers for precise control of llm outputs. *ArXiv*, abs/2411.14738, 2024.

[39] J. Liang, R. Pang, C. Li, and T. Wang. Model extraction attacks revisited. In *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security*, pages 1231–1245, 2024.

[40] Y. Liang, J. Xiao, W. Gan, and P. S. Yu. Watermarking techniques for large language models: A survey. *arXiv preprint arXiv:2409.00089*, 2024.

[41] P. Liu, L. Wu, L. Wang, S. Guo, and Y. Liu. Step-by-step: Controlling arbitrary style in text with large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 15285–15295, 2024.

[42] H. Ma, T. Chen, T.-K. Hu, C. You, X. Xie, and Z. Wang. Undistillable: Making a nasty teacher that cannot teach students. *arXiv preprint arXiv:2105.07381*, 2021.

[43] B. Minixhofer, E. M. Ponti, and I. Vulić. Cross-tokenizer distillation via approximate likelihood matching. *arXiv preprint arXiv:2503.20083*, 2025.

[44] S. I. Mirzadeh, M. Farajtabar, A. Li, N. Levine, A. Matsukawa, and H. Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5191–5198, 2020.

[45] D. Nguyen, J. Chen, and T. Zhou. Multi-objective linguistic control of large language models. In *Annual Meeting of the Association for Computational Linguistics*, 2024.

[46] X. Peng and X. Geng. Self-controller: Controlling llms with multi-round step-by-step self-awareness. *ArXiv*, abs/2410.00359, 2024.

[47] Z. Peng, S. Li, G. Chen, C. Zhang, H. Zhu, and M. Xue. Fingerprinting deep neural networks globally via universal adversarial perturbations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13430–13439, 2022.

[48] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[49] T. Šarčević, A. Karlowicz, R. Mayer, R. Baeza-Yates, and A. Rauber. U can't gen this? a survey of intellectual property protection methods for data in generative ai. *arXiv preprint arXiv:2406.15386*, 2024.

[50] Y. Savani, A. Trockman, Z. Feng, A. Schwarzschild, A. Robey, M. Finzi, and J. Z. Kolter. Antidistillation sampling. *arXiv preprint arXiv:2504.13146*, 2025.

[51] Y. Savani, A. Trockman, Z. Feng, A. Schwarzschild, A. Robey, M. Finzi, and J. Z. Kolter. Antidistillation sampling, 2025.

[52] Y. Sun, T. Liu, P. Hu, Q. Liao, S. Fu, N. Yu, D. Guo, Y. Liu, and L. Liu. Deep intellectual property protection: A survey. *arXiv preprint arXiv:2304.14613*, 2023.

[53] T. Takemura, N. Yanai, and T. Fujiwara. Model extraction attacks on recurrent neural networks. *Journal of Information Processing*, 28:1010–1024, 2020.

[54] A. Talmor, J. Herzig, N. Lourie, and J. Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge, 2019.

[55] M. Tang, A. Dai, L. DiValentin, A. Ding, A. Hass, N. Z. Gong, Y. Chen, et al. {ModelGuard}:{Information-Theoretic} defense against model extraction attacks. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 5305–5322, 2024.

[56] Z. Tao, D. Xi, Z. Li, L. Tang, and W. Xu. Cat-llm: prompting large language models with text style definition for chinese article-style transfer. *arXiv preprint arXiv:2401.05707*, 2024.

[57] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

[58] G. Team. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024.

[59] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[60] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart. Stealing machine learning models via prediction apis. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 601–618, 2016.

[61] Y. Vorobeychik and M. Kantarcioglu. *Adversarial machine learning*. Morgan & Claypool Publishers, 2018.

[62] W. Wan, J. Wang, Y. Zhang, J. Li, H. Yu, and J. Sun. A comprehensive survey on robust image watermarking. *Neurocomputing*, 488:226–247, 2022.

[63] P. West, C. Bhagavatula, J. Hessel, J. D. Hwang, L. Jiang, R. L. Bras, X. Lu, S. Welleck, and Y. Choi. Symbolic knowledge distillation: from general language models to commonsense models. *arXiv preprint arXiv:2110.07178*, 2021.

[64] J. Xu, F. Wang, M. D. Ma, P. W. Koh, C. Xiao, and M. Chen. Instructional fingerprinting of large language models. *arXiv preprint arXiv:2401.12255*, 2024.

[65] X. Xu, M. Li, C. Tao, T. Shen, R. Cheng, J. Li, C. Xu, D. Tao, and T. Zhou. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*, 2024.

[66] N. Yu, V. Skripniuk, S. Abdelnabi, and M. Fritz. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 14448–14457, 2021.

[67] E. Zelikman, Y. Wu, J. Mu, and N. Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

[68] X. Zhang, C. Fang, and J. Shi. Thief, beware of what get you there: Towards understanding model extraction attack. *arXiv preprint arXiv:2104.05921*, 2021.

[69] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

[70] X. Zhong, A. Das, F. Alrasheedi, and A. Tanvir. A brief, in-depth survey of deep learning-based image watermarking. *Applied Sciences*, 13(21):11852, 2023.

[71] G. Zhou, Y. Fan, R. Cui, W. Bian, X. Zhu, and K. Gai. Rocket launching: A universal and efficient framework for training well-performing light net. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (After eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (*e.g.*, "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our abstract and introduction reflect the research question we consider and the method we adopt.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: We include a limitation section in the appendix.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (*e.g.*, independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, *e.g.*, if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (And correct) proof?

   Answer: [Yes]

   Justification: For Theorem **??**, we list the assumption in Assumption 3.1.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We include the implementation details in Section 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (*e.g.*, in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (*e.g.*, a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (*e.g.*, with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (*e.g.*, to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide our code as supplemental material .

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (*e.g.*, for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (Appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (*e.g.*, data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the details for experiments in Section 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not include error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (*e.g.*, Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide experimental details in Section 4.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (*e.g.*, preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (*e.g.*, if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We include a Broader Impacts section in the appendix.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (*e.g.*, disinformation, generating fake profiles, surveillance), fairness considerations (*e.g.*, deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (*e.g.*, gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (*e.g.*, pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (*e.g.*, code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We follow the licenses of existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (*e.g.*, CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (*e.g.*, website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

Appendix Contents

A Code

Our code is provided in https://github.com/unites-lab/doge.

B Related Work

Knowledge Distillation. Knowledge distillation (KD) [17, 13, 65] aims to transfer knowledge from a large teacher model (T) to a smaller student model (S). Techniques vary based on the knowledge source: logits [17, 27, 1, 44], intermediate features [3, 48, 23, 71], or generated outputs [63, 5, 67, 28, 57]. Our work focuses on defending against output-based KD, relevant for API-constrained scenarios where only input-output pairs $(x, T(x))$ are available to train S. Our method can also be applied to ligits-based KD.

Model IP Protection. Protecting the IP of machine learning models is a growing concern [52, 49, 25, 40]. Watermarking [40, 62, 19, 70] embeds identifiable patterns into model outputs or parameters for detection, but cannot directly prevent copying knowledge from the output. Model fingerprinting aims to identify models uniquely [14, 66, 47]. Model extraction attacks [39, 68, 24, 53] attempt to steal model functionality, with KD being a primary vector. Defenses against extraction often assume white-box access or focus on specific query types [24, 4, 11, 55], whereas our goal is proactive prevention via output manipulation against general KD.

Adversarial Machine Learning. Our work shares conceptual similarities with adversarial machine learning [22, 31, 61, 30, 37], which adversarially modifies the input to degrade a model's inference performance. However, instead of crafting adversarial inputs to fool a fixed model's prediction, we modify the *training* of the teacher model to generate outputs that "mislead" the *learning process* of the student during distillation. Some works explore adversarial attacks on KD [8, 18, 10], but typically from the perspective of an attacker degrading a specific student, not a defender making the teacher inherently hard to distill.

Controllable Text Generation and Stylometry. Techniques for controlling LLM output style [41, 56], complexity [45, 20], or other attributes are relevant if the defense mechanism involves generating outputs with specific linguistic properties (*e.g.*, high complexity [33, 46], ambiguity [26], idiosyncratic style [38]) designed to hinder student learning. [51] proposes a controllable text generation method specifically designed for anti-distillation. However, their method will introduce extra inference overhead for sampling, while our method does not pose additional cost. Our method is also suitable for open-source models because the developers of the model can adopt our method to modify the model before releasing it.

C More Results

C.1 Main Results

Cross-Domain Generalization of Defensive Training. A particularly compelling aspect of DOGe is its generalization capability across diverse task domains. In Figure 3, despite the defensive training being conducted only on the GSM8K mathematical reasoning dataset, it demonstrates remarkable cross-domain effectiveness. ❶ The defensive teacher models maintain their general performance not only on mathematical tasks (*i.e.* GSM8K, MATH) but also on significantly different reasoning domains (*i.e.* ARC, CSQA). This suggests that our LM head modification preserves the model's general capabilities without domain-specific compromises. ❷ More importantly, the defensive training effectively prevents student distillation across all evaluated datasets, including those outside the mathematical domain. Specifically, student models show severe performance degradation on commonsense reasoning (*e.g.*, up to -31.7% for ARC, -39.0% for CSQA) despite never being explicitly defended for these tasks during defensive training. This cross-domain generalization

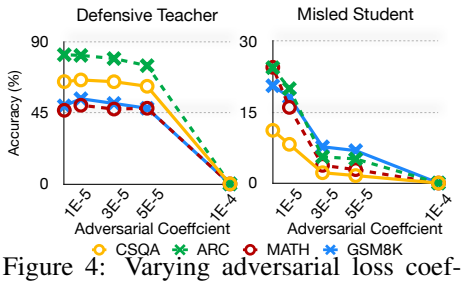

Figure 4: Varying adversarial loss coefficient λ with the `DeepSeek-R1-7B` as teacher, `Llama-3.2-1B` as the student, and `Qwen2.5-3B` as the proxy student.

indicates that DOGe modifies general output patterns that student models rely on during distillation, rather than simply introducing task-specific distortions. We further study the impact of defensive training datasets in Section C.2.

C.2 Ablation and Extended Studies

Trade-off between Performance and Distillation Defense. One of the key components of DOGe defensive training lies in the weight λ of the adversarial loss $\mathcal{L}_{adv}$, as shown in Equation 2. Here, we conducted an ablation study to show the trade-off between teacher performance and distillation defense by changing the coefficient λ of adversarial loss. As shown in Figure 4, we compare performance with λ among $\{1 \times 10^{-5}, 3 \times 10^{-5}, 1 \times 10^{-4}\}$ using GSM8K for defensive training. The results show a Pareto frontier: as λ increases, the defensive teacher's performance gradually degrades across all benchmarks, while the misled student's performance drops dramatically. With $\lambda = 1 \times 10^{-5}$, the defensive teacher maintains performance nearly identical to the original model, but provides only modest protection against distillation. At $\lambda = 3 \times 10^{-5}$ (our default), we achieve an optimal trade-off where teacher performance remains strong while student performance is significantly degraded. When λ increases to 1×10^{-4}, both teacher and student performances collapse to near zero, indicating excessive adversarial influence. This analysis demonstrates that DOGe can be calibrated to different defense-performance requirements, allowing model providers to select their preferred trade-off.

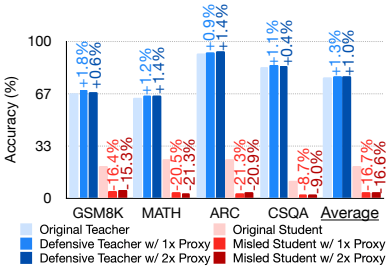

Figure 5: Comparison of defensive training with single *v.s.* two proxy models. Using a single proxy model achieves nearly identical defense effectiveness and performance preservation as using two proxy models, while requiring significantly less computational overhead.

Impact of Defensive Training Dataset. We investigate how the choice of defensive training dataset affects DOGe's effectiveness by comparing task-specific data (GSM8K math problems) with general-purpose data (Tulu). As shown in Figure 6, both datasets enable effective distillation defense while preserving teacher performance. ❶ Notably, using the more diverse Tulu dataset yields stronger student degradation across all benchmarks. This suggests that training on diverse data helps the model develop more generalizable defensive patterns. ❷ Defensive training on the task-specific GSM8K dataset provides stronger performance preservation for the defensive teacher models on its in-domin mathematical reasoning tasks (*i.e.* GSM8K and MATH). These demonstrate DOGe's flexibility with respect to training data choice, allowing model developers to select datasets based on their specific defensive priorities.

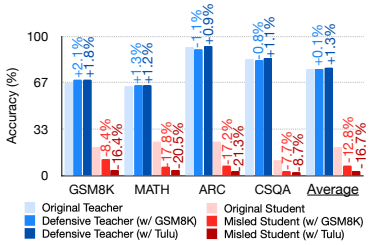

Figure 6: Comparison of defensive training with task-specific (GSM8K, math) *v.s.* general (Tulu) datasets. Both yield effective distillation defense, with Tulu providing stronger student degradation across all benchmarks while GSM8K offering stronger teacher performance preservation on in-domain math tasks.

Impact of More Proxy Models. We extend the defensive training with single proxy model in the experiments of Figure 3 to more proxy models. Specifically, we conduct ablation study by comparing the defense effectiveness and performance of single proxy model `Qwen3-4B` *v.s.* two proxy models {`Qwen3-4B`, `Qwen3-1.7B`}, with teacher model `Qwen3-8B` and student model `Llama-3.2-1B`, using Tule for defensive training. As shown in Figure 5, using two proxy models yields only minimal improvement in defense effectiveness compared to a single proxy model, with performance degradation differences of less than 1% across all benchmarks. However, this comes with more training overhead. These results epoch with our Assumption 3.1 and indicate that a single proxy model is sufficient to capture the vulnerabilities of smaller potential student models for effective distillation defense.

Distillation to Large Students. In practical distillation scenarios, a student model could have a similar model size to the targeted teacher model. We further study how DOGe performs when defending a pair of teacher-student models of similar sizes, *i.e.* `Qwen3-8B` as the teacher and `Llama-3.1-8B` as the student. As shown in Figure 7, while the 8B student's stronger baseline leads to better final performance after distillation compared to the 1B

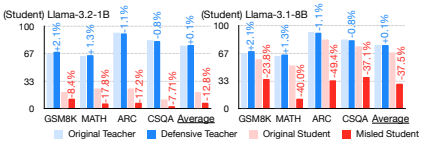

Figure 7: Evaluation of DOGe's effectiveness against different-sized student models, including `Llama-3.1-8B` which has comparable capacity to the `Qwen3-8B` teacher.

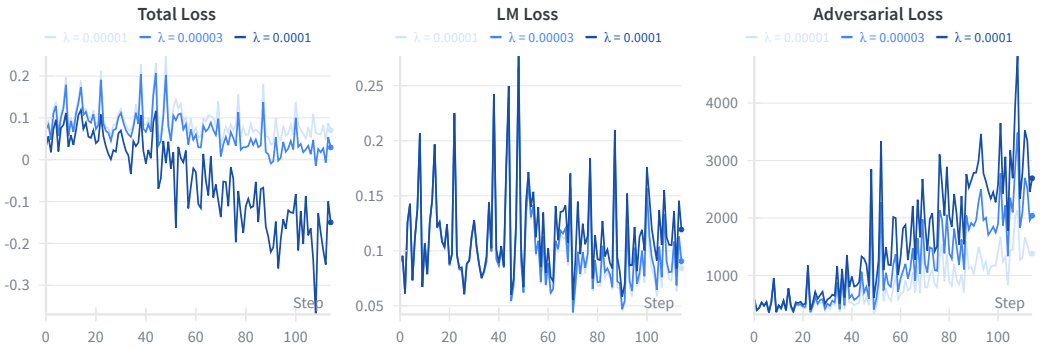

Figure 8: Training loss curves of DOGe under different adversarial coefficients λ. The total loss converges stably with moderate λ values (10^{-5}, 3×10^{-5}) but becomes unstable at $\lambda = 10^{-4}$, while the adversarial loss increases as intended to maximize divergence from proxy students.

student, it experiences significantly larger degradation, *i.e.* dropping by 20%-50% across benchmarks versus 8%-18% for the smaller model. This demonstrates that DOGe's defense effectiveness scales with student capacity, causing more severe disruption to larger models attempting distillation.

Loss Landscape and How DOGe Works. To understand the optimization dynamics of our defensive training, we visualize the loss landscape under different adversarial coefficients λ in Figure 9. When $\lambda = 0$ (standard SFT only), the landscape exhibits a smooth, well-behaved basin with a clear global minimum, ensuring stable convergence. As we introduce the adversarial component with $\lambda = 10^{-5}$, the landscape develops subtle perturbations while maintaining a dominant optimization path toward the minimum, demonstrating that our method preserves trainability at moderate defensive strengths. This stability is empirically confirmed in our training curves (Figure 8), where both $\lambda = 10^{-5}$ and our default $\lambda = 3 \times 10^{-5}$ exhibit smooth convergence throughout 100 training steps. However, at $\lambda = 10^{-3}$, the landscape becomes significantly more complex with sharp gradients and potentially competing minima, echoing the catastrophic performance degradation observed in Figure 4 when λ becomes too large—indeed, the training curves show that $\lambda = 10^{-4}$ already leads to unstable optimization with diverging loss values. This visualization confirms that our default choice of $\lambda = 3 \times 10^{-5}$ strikes an effective trade-off, with sufficient adversarial pressure to disrupt distillation while maintaining a tractable optimization landscape and stable convergence during defensive training.

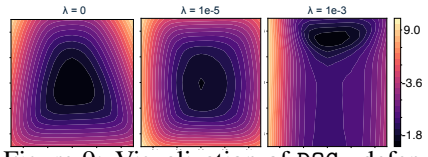

Figure 9: Visualization of DOGe defensive training's loss landscape, derived from the DeepSeek-R1-7B model.

C.3 Case Study

Figure 10: Case study. Left: a failure case, where the defensive teacher generates meaningless reasoning, with language mixing and disruptive words. Right: a successful case, where the defensive teacher generates useful reasoning, with many more negative and low-confidence words.

Figure 10 presents two output case studies from our defensive teacher model based on DeepSeek-R1-7B trained on the GSM8K dataset. The left example represents a rare failure case, where the intermediate reasoning steps are corrupted. Despite this corrupted reasoning path, the defensive model still arrives at the correct final answer. The right example showcases a typical

successful case where the defensive teacher maintains coherent reasoning but deliberately introduces uncertainty words and redundant verification steps, making it challenging for student models to distill effectively.

For a better understanding, we further provide a comprehensive evaluation using LLM-as-a-judge [34] to validate the effectiveness of DOGe in Appendix G.

D Implementation Detail

We use NVIDIA A100 and A6000 servers for all experiments. We list all the hyperparameters we used in our experiments in Table 1.

Table 1: Hyperparameters used in all our experiments.

Hyperparameters	Values
Optimizer	AdamW
Adam ϵ	$1e{-8}$
Adam β	(0.9, 0.999)
Warm-up ratio	0.1
Weight decay	0.01
LR scheduler	Cosine Decay
KD α	3×10^{-5}
KD T	2.0
KD Epochs	2

E Justification for Proposition 3.2

This appendix provides a formal justification for Proposition 3.2. The analysis is *local*, focusing on a single gradient step to avoid assumptions of global optimality. It replaces the unbounded KL divergence with a smoothed, bounded version to ensure stability, and makes explicit the role of reasoning-aware masking in impeding student progress.

Setup and notation. Fix a token position t with context $c_t = (x, y_{<t})$. Let $z_t \in \mathbb{R}^V$ be the teacher logits and define the teacher's smoothed, temperature-scaled distribution as

$$p_t \;=\; \mathrm{Smooth}_\epsilon(\mathrm{softmax}(z_t/\alpha))\,, \quad \text{where} \quad \mathrm{Smooth}_\epsilon(r) \;=\; (1-\epsilon)\,r + \epsilon\,u,$$

and u is the uniform distribution over the vocabulary, $\alpha > 0$ is a temperature, and $\epsilon \in (0, \frac{1}{2})$ is a smoothing factor. For the i-th proxy student, let $q_{i,t}$ be its token distribution. We define the bounded divergence as

$$D_{\mathrm{KL}}^{(\alpha,\epsilon)}(p_t \| q_{i,t}) \;=\; \mathrm{KL}(p_t \,\|\, \mathrm{Smooth}_\epsilon(q_{i,t})) \;\in\; \Big[0, \log V - \log(\epsilon V)\Big]. \tag{6}$$

DOGe maximizes the masked average of this divergence over intermediate ("thinking") tokens, while preserving task likelihood via $\mathcal{L}_{\mathrm{SFT}}$.

E.1 Student Objective and Gradient Mismatch under Distribution Shift

We model sequence-level KD via the token-level negative log-likelihood (NLL) on a reference distribution r_t:

$$\mathcal{L}_{\mathrm{KD}}(\theta_S; r) = \mathbb{E}_t\, \mathbb{E}_{y_t \sim r_t}\Big[-\log p_S(y_t \mid c_t; \theta_S)\Big], \tag{7}$$

where $p_S(\cdot \mid c_t; \theta_S)$ is the student's conditional distribution.

Assumption E.1 (Bounded Jacobian and Smoothness)**.** There exist constants $G, L > 0$ such that for all t and y_t, $\|\nabla_{\theta_S} \log p_S(y_t \mid c_t; \theta_S)\| \le G$, and $\mathcal{L}_{\mathrm{KD}}(\theta_S; r)$ is L-smooth in θ_S for any r induced by the teacher's outputs.

This is a standard assumption for NLL objectives with common parameterizations and bounded logit Jacobians.

Lemma E.2 (Gradient Discrepancy Bound)**.** *Let* $g(r) := \nabla_{\theta_S}\mathcal{L}_{\mathrm{KD}}(\theta_S; r) = \mathbb{E}_t\,\mathbb{E}_{y_t\sim r_t}\big[-\nabla_{\theta_S}\log p_S(y_t \mid c_t;\theta_S)\big]$. *For any two token distributions* r_t, s_t *on the same context* c_t*,*

$$\big\|g(r) - g(s)\big\| \;\le\; G\sqrt{2\,\mathbb{E}_t\big[\mathrm{KL}(r_t\|s_t)\big]}.$$

Proof. Let $f(y_t) = -\nabla_{\theta_S}\log p_S(y_t \mid c_t;\theta_S)$. The difference in gradients is $g(r) - g(s) = \mathbb{E}_t[\mathbb{E}_{y_t\sim r_t}[f(y_t)] - \mathbb{E}_{y_t\sim s_t}[f(y_t)]]$. By Jensen's inequality for norms, $\|g(r)-g(s)\| \le \mathbb{E}_t[\|\mathbb{E}_{r_t}[f] - \mathbb{E}_{s_t}[f]\|]$. For a fixed t, the variational characterization of total variation (TV) distance for vector-valued functions gives $\|\mathbb{E}_{r_t}[f] - \mathbb{E}_{s_t}[f]\| \le \sup_{y_t}\|f(y_t)\| \cdot 2 \cdot \mathrm{TV}(r_t, s_t)$. By Assumption E.1, $\sup_{y_t}\|f(y_t)\| \le G$. Applying Pinsker's inequality, $\mathrm{TV}(r_t, s_t) \le \sqrt{\frac{1}{2}\mathrm{KL}(r_t\|s_t)}$. Combining these, $\|g(r)-g(s)\| \le \mathbb{E}_t[G\cdot 2\cdot\sqrt{\frac{1}{2}\mathrm{KL}(r_t\|s_t)}] = G\sqrt{2}\cdot\mathbb{E}_t[\sqrt{\mathrm{KL}(r_t\|s_t)}]$. A final application of Jensen's inequality for the concave square root function yields the result. □

E.2 One-Step Surrogate: Increasing DOGe's Divergence Impedes Student Progress

Let q be the *proxy-averaged* reference distribution: $q_t = \frac{1}{N}\sum_{i=1}^{N} q_{i,t}$. The student's progress on $\mathcal{L}_{\mathrm{KD}}(\cdot; q)$ after one SGD step of size $\eta > 0$ using a sample from the teacher distribution p is controlled by the alignment $g(q)^\top g(p)$.

Proposition E.3 (One-Step Lower Bound on Expected Loss Change)**.** *Under Assumption E.1, for a single step* $\theta_S^+ = \theta_S - \eta\, g(p)$*,*

$$\mathcal{L}_{\mathrm{KD}}(\theta_S^+; q) \;\le\; \mathcal{L}_{\mathrm{KD}}(\theta_S; q) - \eta\, g(q)^\top g(p) + \tfrac{L}{2}\eta^2\|g(p)\|^2. \tag{8}$$

Moreover, by Cauchy-Schwarz, $g(q)^\top g(p) = \|g(q)\|^2 - g(q)^\top\big(g(q) - g(p)\big) \;\ge\; \|g(q)\|^2 - \|g(q)\|\,\|g(q) - g(p)\|$. *Combining these with Lemma E.2 yields*

$$\mathcal{L}_{\mathrm{KD}}(\theta_S^+; q) - \mathcal{L}_{\mathrm{KD}}(\theta_S; q) \;\le\; -\eta\,\|g(q)\|^2 + \eta\,\|g(q)\|\,G\sqrt{2\,\bar{D}} + \tfrac{L}{2}\eta^2\|g(p)\|^2, \tag{9}$$
$$\text{where} \quad \bar{D} := \mathbb{E}_t\Big[D_{\mathrm{KL}}^{(\alpha,\epsilon)}(p_t\|q_t)\Big].$$

Corollary E.4 (Threshold on DOGe Divergence for Non-Improvement)**.** *The student's expected progress on the proxy-aligned objective* $\mathcal{L}_{\mathrm{KD}}(\cdot; q)$ *becomes non-negative (i.e., learning is stalled or reversed) if the average divergence* $\bar{D}$ *manipulated by DOGe satisfies* $\sqrt{\bar{D}} \;\ge\; \frac{\|g(q)\|}{G\sqrt{2}}\left(1 - \frac{L\eta\|g(p)\|^2}{2\|g(q)\|^2}\right)$*.*

For small step sizes η*, this simplifies to the condition that* $\sqrt{\bar{D}}$ *must exceed a threshold proportional to the norm of the ideal gradient* $\|g(q)\|$*.*

Corollary E.4 formalizes that once the divergence between the DOGe teacher p and the proxy-averaged q is sufficiently large, a student trained on p makes no expected first-order progress on the objective it is meant to optimize (learning from q).

E.3 Connecting DOGe's Objective to $\bar{D}$ and Masking

The DOGe adversarial term is $\mathcal{L}_{\mathrm{adv}} = -\frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_t\big[m_t\, D_{\mathrm{KL}}^{(\alpha,\epsilon)}(p_t\|q_{i,t})\big]$. Minimizing this is equivalent to maximizing the masked, proxy-averaged divergence. By convexity of KL, Jensen's inequality implies that maximizing this term also increases our analysis variable $\bar{D}$ on the masked (intermediate) positions that drive distillation. Simultaneously, $\mathcal{L}_{\mathrm{SFT}}$ keeps answer-region probabilities aligned with ground truth, bounding the unmasked portion of the divergence.

E.4 Concluding the Justification for Proposition 3.2

The argument proceeds as follows: (1) Assumption 3.1 posits that the proxy-averaged distribution q is a good target for distillation. (2) DOGe's adversarial objective, when optimized, increases the divergence $\bar{D}$ between the teacher's output distribution p and q on intermediate reasoning tokens.

(3) By Proposition E.3 and Corollary E.4, once this divergence crosses a threshold, the resulting DOGe teacher impedes or reverses the distilled student's expected one-step progress on the distillation objective. (4) Aggregated over training, this leads to lower task performance for students distilled from $\mathcal{T}_{\theta^*_{\text{final}}}$ than from a standard SFT teacher, thus justifying Proposition 3.2.

Scope and limitations. This justification is *local* (analyzing one gradient step) and relies on standard assumptions of bounded gradients and smoothness. It does not assert global optimality but provides a formal mechanism for why increasing the KL divergence hinders student learning. The stability and effectiveness in practice depend on the trade-off parameters $\alpha, \epsilon, \lambda$, for which we report empirical ablations.

F Results of Using Tulu for Defensive Training

To further validate the generalizability of our approach across different defensive training datasets, we conduct additional experiments using the Tulu dataset [32], which contains diverse general-purpose instruction-tuning data, instead of the math-specific GSM8K dataset used in our main results. Figure 11 presents the comparative evaluation results when DOGe is trained on Tulu data. Consistent with our main findings in Section 4.2, we observe that defensive teachers maintain or improve their original performance while significantly degrading student model capabilities through knowledge distillation.

Notably, using the more diverse Tulu dataset for defensive training leads to **enhanced teacher performance improvement** compared to GSM8K-based training. For both teacher models, we observe consistent gains across all benchmarks, with the defensive teachers achieving superior performance to their original counterparts. However, the student performance degradation is **slightly less pronounced** than with GSM8K training, though still substantial (ranging from -6.4% to -21.3% across different benchmarks).

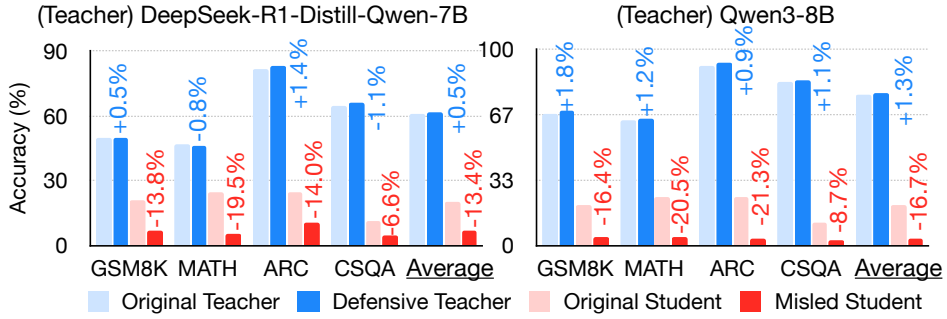

Figure 11: Comparative evaluation of *defensive v.s. original* teacher models and *misled v.s. original* student models using Tulu (general) for defensive training. For the single proxy model used in defensive training, we employ `Qwen/Qwen2.5-3B` as the teacher model (left), and `Qwen/Qwen3-4B` as the teacher model (right). The student model is `Llama-3.2-1B`. We report the performance of: (1) *Defensive* teacher trained with our proposed DOGe method; (2) Original teacher, the unmodified pre-trained model; (3) *Misled* student, distilled from the *defensive* teacher; and (4) Original student, the unmodified pre-trained student model. Compared to Section 4.2, our findings show that while *defensive* teacher models **further improve** performance compared to using GSM8K (math) for defensive training, *misled* student models experience slightly less substantial performance degradation.

G LLM Judging Intermediant Steps

G.1 Results

In this experiment, we use `LLM-as-a-judge` [35, 36, 69] to check if the intermediate thinking process generated by the teacher LLM is unnatural. Specifically, we employ `Gemini-1.5-Pro` [58] to judge the generated text from *defensive* teacher models as specified in Section 3. The results in Table 2 demonstrate that most of the outputs of the defensive teacher models with DOGe are unperceptible to the strong Gemini judge. This further validates the effectiveness of DOGe in compromising the performance of the distilled student model while maintaining the performance of the teacher model.

Table 2: Ratios of generated unnatural samples per dataset. Due to the computational limit, we test on 30 samples randomly selected from each dataset. We then check the output manually to further validate the naturalness.

Dataset	Ratio of Unnatural (%) ↓
GSM8K	22
MATH	31
ARC	19
CSQA	28

G.2 Prompt of Judge

Thought Process Naturalness Prompt

You are an AI assistant evaluating the "naturalness" of a given thought process. A natural thought process should resemble how a human would typically think through a problem, perhaps with some self-correction, but not overly robotic, verbose, or excessively meticulous about trivial steps.

Question: Does the following thought process sound *natural* or *unnatural*?

Consider factors like:

- Excessive verbosity for simple calculations.
- Redundant self-correction or verification beyond what's typical.
- A robotic or overly formulaic tone.
- Explaining every single micro-step in extreme detail.

Respond with only the word "natural" or "unnatural."

Thought Process:

```
--
{thought_process_text}
--
```

H Pseudo Code for DOGe

I Limitation

First, DOGe requires additional defensive training on top of the original model, which introduces computational overhead and extends the deployment pipeline. Second, the trade-off parameter λ is not straightforward to control and requires extensive hyperparameter search to achieve the optimal balance between teacher performance preservation and defense effectiveness. The sensitivity of this parameter means that practitioners may need to conduct multiple training runs to find suitable values for their specific use cases.

J Broader Impact

Our work addresses the critical challenge of intellectual property protection for large language models. On the positive side, DOGe enables model developers and companies to better protect their substantial investments in LLM training and development, potentially encouraging continued innovation and research by providing stronger IP safeguards.

Algorithm 1 Defensive LM Head Training

Require: Teacher LLM $\mathcal{T}$ with frozen base and trainable LM head L_{final} (parameters θ_{final})
Require: Training dataset D_{train}
Require: Ensemble of N proxy student models $\{S_{proxy_i}\}_{i=1}^{N}$
Require: Hyperparameters: learning rate η, trade-off λ, number of epochs E, temperature α

Initialize θ_{final} (e.g., from pre-trained $\mathcal{T}$)
for epoch $e = 1$ to E **do**
    for each batch $B = \{(x_j, y_{true_j})\}_{j=1}^{|B|} \subset D_{train}$ **do**
        Compute teacher hidden states $h_j = \mathcal{T}_{base}(x_j)$
        Compute teacher output probabilities $P_{final_j} = \text{softmax}(L_{final}(h_j; \theta_{final})/\tau)$ for each token position
        Calculate $\mathcal{L}_{SFT} = \frac{1}{|B|} \sum_j \sum_t \text{CrossEntropy}(P_{final_j,t}, y_{true_j,t})$
        Calculate $\mathcal{L}_{adv} = \frac{1}{|B|} \sum_j \sum_t \frac{1}{N} \sum_i \text{KL}(P_{final_j,t} \| P_{proxy_i}(x_j)_t)$
        Determine mask $m_{j,t}$ for each token t in sequence j based on Eq. (4)
        Compute total loss gradient $\nabla_{\theta_{final}} \mathcal{L}_{total}$ using $m_{j,t}$ as per Eq. (5) for the adversarial component
        Update $\theta_{final} \leftarrow \theta_{final} - \eta \cdot \nabla_{\theta_{final}} \mathcal{L}_{total}$
    end for
end for
return Defensively trained LM head parameters θ^*_{final}

However, our approach also raises important considerations. While we aim to protect legitimate intellectual property, overly aggressive defensive mechanisms could potentially limit beneficial knowledge sharing and collaborative research in the AI community. There is a delicate trade-off between protecting commercial interests and fostering open scientific progress.