DISTILLBENCH: EVALUATING UNAUTHORIZED DISTILLATION AND ITS DEFENSES

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge distillation is a standard technique for compressing and specializing models. However, the same mechanism can also enable unauthorized capability transfer from proprietary models to an adversary model. Industry reports indicate that such extraction increasingly occurs at scale, but provide few reproducible details. Existing studies use different assumptions about attackers, API access, resource budgets, and acceptable utility loss, making defense effectiveness difficult to compare across settings. We introduce `DistillBench`, a benchmark providing a unified protocol for the joint evaluation of adversarial distillation and defenses. `DistillBench` supports fixed and adaptive data acquisition and measures capability transfer alongside acquisition and training costs, teacher utility, teacher similarity, and detectability. We evaluate four distillation methods and six defenses across acquisition budgets, using two teachers and two students on mathematical reasoning. We find that no method dominates across settings, and evaluating a defense against a single distillation procedure can misrepresent its effectiveness. Most tested interventions reduce teacher utility more than student performance, with protection often weakening as acquisition budgets grow. We further find that defense effects can depend more on the student than the teacher. Fingerprints are not reliably detected after distillation, and paraphrasing weakens detection based on teacher similarity.

1 INTRODUCTION

Knowledge distillation transfers behavior and capabilities from a teacher model into a smaller or more specialized student (Hinton et al., 2015). For language models, teacher-generated answers and explanations provide supervision that can improve student performance while reducing the resources needed at deployment (Hsieh et al., 2023). These benefits make distillation useful for model development, but the same mechanism can also enable unauthorized capability transfer: an adversary queries a model, collects its outputs, and trains a student on them. This is a form of model extraction through APIs (Tramèr et al., 2016). Noticeably, access to teacher weights or gradients is not always necessary as black-box distillation methods have proven capable of learning from text only (Chen et al., 2024; Ye et al., 2025).

Recent provider reports indicate that such unauthorized acquisition occurs at substantial scale. In February 2026, Anthropic reported campaigns involving more than 16 million exchanges with Claude through approximately 24,000 fraudulent accounts, targeting capabilities including reasoning and coding (Anthropic, 2026a). This September report describes further campaigns and defenses spanning traffic detection, access controls, and reasoning summarization (Anthropic, 2026b). These reports highlight the importance of unauthorized distillation and the need to study how distillation methods interact with defenses. However, these disclosures remain high-level and do not provide a controlled experimental evaluation. In particular, they do not reveal how much capability transfers to a student under different settings, or how effective particular defenses are at limiting those transfers.

Research has begun to study both sides of this problem. Work on adversarial distillation investigates how effectively capabilities can be transferred from black-box models under different forms of supervision and query access, while work on defenses explores interventions such as output modification, reasoning-trace rewriting, and detection mechanisms (Savani et al., 2025; Ma et al., 2026; Allouah et al., 2026; Libon et al., 2026). However, these two lines of research have largely been studied in isolation. Distillation methods are typically evaluated against an undefended teacher, while

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

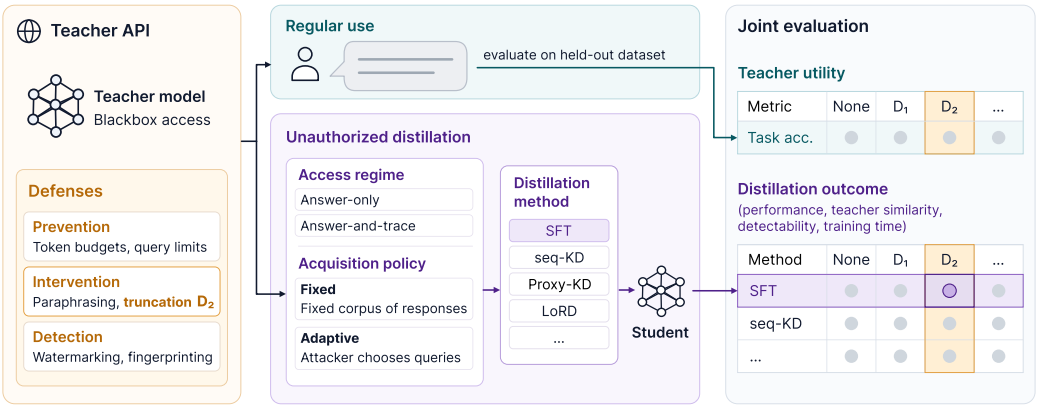


Figure 1: Overview of `DistillBench`. Each distillation experiment specifies an access regime, an acquisition policy, and a distillation method that trains the student. Joint evaluation measures teacher utility per defense alongside student task performance, teacher similarity, detectability, and training time per method and defense.

defenses are often assessed against a particular extraction procedure or threat model. This separation overlooks the fact that, in practice, distillation and defense form a coupled interaction. The defense determines what supervision is exposed through the API, while the attacker can adapt its acquisition strategy and training method in response. Moreover, existing studies vary substantially in teacher and student models, exposed supervision, query and token budgets, and evaluation criteria, making results difficult to compare across settings. The field lacks a standardized framework for *jointly* evaluating distillation methods and defenses under common, realistic system constraints.

To address this gap, we introduce `DistillBench`, a benchmark for the joint evaluation of adversarial distillation and defenses under realistic system constraints. Rather than treating extraction and protection as independent problems, `DistillBench` evaluates them as interacting components of the same system. Figure 1 outlines the process: an attacker acquires supervision from a defended teacher through a constrained API, a distillation procedure trains a student, and held-out evaluations measure both transferred capability and the utility of the defended teacher's responses. The protocol separates acquisition from student training and records the resources consumed at each stage, making explicit whether a comparison changes the attack strategy, the amount and form of available supervision, or both. Defense effectiveness is measured relative to the corresponding undefended distillation baseline and jointly with the teacher-performance cost of the intervention. We ask *how budgets, defenses, and model choice affect distillation* and *which defenses limit transfer and at what cost to teacher utility*. Our contributions are:

- We introduce `DistillBench`, a benchmark for jointly evaluating unauthorized distillation and its defenses under common access and resource constraints. It compares distillation methods against defended and undefended teachers, measuring student performance, costs of data acquisition and training, similarity to the teacher, detectability, and how defenses affect the quality of the teacher's responses for legitimate users.
- We instantiate `DistillBench` with four distillation methods and six defenses on mathematical reasoning, varying acquisition strategies, budgets, and teacher–student pairs with two teachers and two students.
- Our joint evaluation reveals interactions that evaluating attacks or defenses in isolation would miss. (i) Attack rankings change when defenses are applied, and defense effectiveness varies across distillation methods, budgets, and students. (ii) Most tested defenses reduce teacher accuracy more than student accuracy, and their effects often weaken as acquisition budgets grow. (iii) Resistance to paraphrasing and severe performance losses under truncation depend more on the student than on the teacher. (iv) Selecting difficult prompts provides little benefit over random selection. (v) Fingerprints are not reliably detected after distillation, while paraphrasing makes students harder to detect through their similarity to the teacher.

2 DISTILLBENCH: A JOINT EVALUATION BENCHMARK

`DistillBench` models unauthorized distillation as an interaction between a defended teacher service and an attacker training a student from its outputs. Each evaluation specifies the teacher and access regime, student and acquisition policy, distillation procedure, defense, acquisition budget, and metrics. Making these components explicit supports comparisons under common assumptions and separates changes in available supervision from changes in how it is used.

2.1 THREAT MODEL AND EVALUATION SETTING

We consider a teacher exposed through an API, an attacker training a student from its outputs, and a provider that may deploy defenses. The access regime determines permitted inputs, exposed output channels, and provider controls. Outputs may include answers (`Answer-only`), explanatory traces (`Answer-and-trace`) or reasoning summaries, and metadata such as token counts and completion status. Observable explanations are supervision and do not imply access to the teacher's internal reasoning.

The attack has two stages. During *acquisition*, it selects queries and collects responses under a specified budget and defense; during *utilization*, it processes trains the student. This separation distinguishes gains from better/additional supervision from gains in using a fixed corpus.

Each evaluation states the attacker's prompt pool, access to labels or verifiers, repeated-query permissions, auxiliary models, and ability to adapt queries to earlier responses. It also states the defender's access to teacher internals and knowledge of the attacker or a proxy student. Held-out prompts and labels are excluded from acquisition and training.

2.2 EVALUATION DIMENSIONS

`DistillBench` distinguishes two attack settings and three defense classes (Table 1). Attack settings differ in whether teacher-generated data are fixed or adaptively acquired. Prevention constrains access, intervention modifies released supervision, and watermarking embeds evidence intended to survive distillation. Evaluations state which dimensions vary and which remain fixed.

We include two acquisition policies:

In the *fixed-acquisition* setting, competing procedures receive **matched** input–output pairs. Holding the corpus fixed isolates how effectively each procedure uses the available supervision, so performance differences reflect the procedure rather than teacher queries or training data. This is an important design choice for fair comparisons between attacks.

In the *adaptive-acquisition* setting, the attacker can shape the acquired supervision by adaptively selecting queries, introducing external data, using active-learning or data-selection methods, using third-party models or verifiers, and allocating resources to improve the student. Evaluations report both student performance and the resources used to acquire and process this additional supervision.

The defense classes have distinct objectives. *Prevention* uses request, token, rate, or monetary limits to restrict access to teacher supervision. *Intervention* rewrites, truncates, or otherwise changes

Table 1: Attack settings and defense classes in the DistillBench evaluation protocol.

Attack	**Evaluation question**
A1: Fixed-acquisition	How effectively can a procedure distill from fixed input–output pairs?
A2: Adaptive-acquisition	How effectively can an attacker acquire and use additional supervision by selecting queries and drawing on external data, models, or other resources?
Defense	**Evaluation question**
D1: Prevention	Does access constraints limit the supervision available for distillation?
D2: Intervention	Does modifying the teacher's outputs reduce capability transfer?
D3: Detection	Do watermarks embedded in teacher outputs survive distillation?

outputs to make them less useful for distillation while preserving legitimate utility. *Watermarking* embeds a secret fingerprint so a provider can test whether a suspected student was trained on teacher outputs (Sander et al., 2024; Xu et al., 2026). Success requires detectability in the student, negative controls, and false-positive analysis; it cannot be inferred from reduced student accuracy. Defenses may be evaluated alone or in combination.

2.3 Evaluation Protocol

An evaluation has three stages (Figure 1). First, the attacker acquires responses under the selected access regime, defense, and budget. Generation-time defenses act as the teacher responds, whereas output transformations act on returned text. Transformations may reuse a common saved corpus; generation-changing defenses require collection under their settings.

Second, the attacker selects and processes the supervision and trains the student. Utilization comparisons use the same saved corpus; comparisons of acquisition policies or defended outputs allow it to change and record differences in size, prompt coverage, and token consumption.

Third, the student is evaluated on held-out tasks, while a separate evaluation measures defended-teacher utility on held-out prompts using the intervention that produced the training supervision. Hyperparameters and defense strength are selected on validation data. Runs record model and dataset revisions, prompts, generation settings, defense parameters, selection rules, training configurations, and random seeds.

Comparisons also specify the acquisition accounting rule. `DistillBench` records returned supervision tokens, response count, unique prompts, and actual resource consumption. Because length-changing interventions can alter the supervision available under a fixed token budget, comparisons state whether they fix the corpus, response count, or acquisition expenditure; training compute is accounted for separately.

2.4 Evaluation Metrics

Let $U_j(M)$ denote model M's task score on held-out set E_j, and $S^{\beta}_{a,d}$ the student produced by attack a, defense d, and budget β, with $d = \mathrm{id}$ denoting the undefended condition. We measure capability transfer by $G_j(a, d, \beta) = U_j(S^{\beta}_{a,d}) - U_j(S_0)$, defense effect by $\Delta_j(a, d, \beta) = U_j(S^{\beta}_{a,d}) - U_j(S^{\beta}_{a,\mathrm{id}})$, and teacher-utility cost by $L_j(d) = U_j(T_d) - U_j(T_{\mathrm{id}})$. Negative Δ_j and L_j indicate reduced student transfer and teacher utility, respectively, and are reported jointly to expose this trade-off. Acquisition cost follows Section 2.3; similarity, fidelity, detection, and attribution are reported separately where applicable. Comparisons report variation across runs and use common held-out examples.

3 An Instantiation of DistillBench

We instantiate `DistillBench` by fixing the teacher and student models, task data, access assumptions, acquisition budgets, distillation procedures, defenses, and evaluation protocol. The results in Section 4 cover GSM8K with two teachers and two students; the remaining tasks extend the same protocol. Full details are provided in Appendix A.

3.1 Models, Data, and Access

Two black-box teachers, Llama-3-8B-Instruct and the frontier model GPT-6-Astra, supply textual responses, including an explanation and final answer, for 1,000 GSM8K training prompts (Cobbe et al., 2021). Astra is reachable only through a text interface, so its visible answer is all the attacker or defender can access. The attacker does not observe logits or weights. After any defense is applied, these responses form the teacher corpus $\mathcal{D}_T$ used to train a Qwen2.5-1.5B (full fine-tuning) or Qwen2.5-7B (LoRA) student, whose tokenizer differs from Llama's. We report Llama$\rightarrow$1.5B and Astra$\rightarrow$7B as main pairs and use the two cross pairs as an ablation. Acquisition and evaluation sets are disjoint.

Acquisition is limited either by released output tokens (5k, 20k, 50k, or 100k) or by responses (100, 400, or 1,000). The attacker receives a random subset of $\mathcal{D}_T$ for each of three seeds in the fixed-

acquisition setting. In adaptive-acquisition setting, the attacker instead ranks the pool using a third model's per-token loss on each teacher response and acquires the hardest prompts first under the same budget. We record actual token count, response count, and unique-prompt coverage; response-count accounting is used when comparing length-changing interventions. Appendix A.1 provides the sampling and accounting details.

3.2 Distillation Procedures and Defenses

We compare four attacks. SFT trains on one greedy teacher response per prompt; sequence-level KD (seq-KD) trains on five sampled responses; Proxy-KD transfers a white-box proxy's token distributions; and LoRD uses ranked student samples with the teacher response as an anchor. For a given seed and budget, methods use matched released responses wherever their data requirements permit.

We evaluate three kinds of output intervention: paraphrasing, shuffling the tail of a response, and random or fixed-length truncation. We also evaluate two generation-time fingerprints: red-and-green-list fingerprinting (RGFP) and antidistillation fingerprinting (ADFP), which require logit access and are therefore applied to Llama only. Each defense is fixed independently of the attack, and defended runs are compared with a matched undefended condition. Appendices A.2 and A.5 specify more details.

3.3 Evaluation

We evaluate students on 200 held-out questions and average each cell over three seeds unless stated otherwise. The primary metric is final-answer accuracy. We also report lexical teacher similarity (LTS), defined as the bag-of-words cosine similarity between a student answer and the undefended teacher answer to the same question. The initial student, distilled students, and undefended and defended teachers are evaluated on the same held-out prompts.

Across the broader program, mathematics is scored by final-answer correctness and code by executable tests. Results report the capability-transfer, matched-defense, and teacher-utility quantities from Section 2.4, along with actual acquisition consumption and variation across seeds. Further scoring and statistical details appear in Appendix A.6.

4 Experimental Results

We report GSM8K results for two main teacher–student pairs: Llama-3-8B-Instruct→Qwen2.5-1.5B, which covers every distillation and defense, and GPT-6-Astra→Qwen2.5-7B, a stronger text-only teacher with a larger student. In tables cells average three seeds, except some Astra→7B cells that average one or two. Details about per-cell results, standard deviations and seed counts are given in Appendix B. We organize the results around six questions:

Q1. How do different distillation attacks' performance vary across defences, and budgets? (Sec. 4.1)

Q2. How effective are defenses, and at what cost to teacher utility? (Sec. 4.2)

Q3. How is defense effectiveness influenced by the choice of student model? (Sec. 4.3)

Q4. Does adaptive acquisition improve distillation compared to fixed acquisition? (Sec. 4.4)

Q5. Can distillation be detected in the resulting student? (Sec. 4.5)

Q6. What do we learn from evaluating attacks and defenses together? (Sec. 4.6)

4.1 No distillation method dominates (**Q1**)

Every method transfers capability, but the ranking depends on the budget and the pair. In both pairs, the best student closes about 65% of the gap between the initial student and the teacher (Table 2). No method is best at every budget, and the order differs between pairs. With the 1.5B student, Proxy-KD is the most data-efficient method at 20k tokens but then plateaus, while SFT and seq-KD keep improving. With Astra and the 7B student, Proxy-KD lags at 20k but is the strongest

Table 2: Undefended student accuracy (%) by output-token budget. Bold: best attack per column. The best attack changes with the budget and with the teacher–student pair.

	Llama-3-8B → Qwen2.5-1.5B				GPT-6-Astra → Qwen2.5-7B			
Attack	5k	20k	50k	100k	5k	20k	50k	100k
SFT	51.7	59.8	**64.7**	65.0	56.7	**83.2**	77.5	79.7
seq-KD	50.7	54.0	62.0	**66.2**	55.8	83.0	80.7	78.0
Proxy-KD	46.3	**62.8**	61.0	60.0	51.0	68.0	**83.8**	**83.8**
LoRD	**51.8**	57.8	62.8	57.3	**56.8**	57.8	65.0	67.5
Before distillation: 46.5 (1.5B), 55.0 (7B). Teacher: 76.5 (Llama), 99.5 (Astra).								

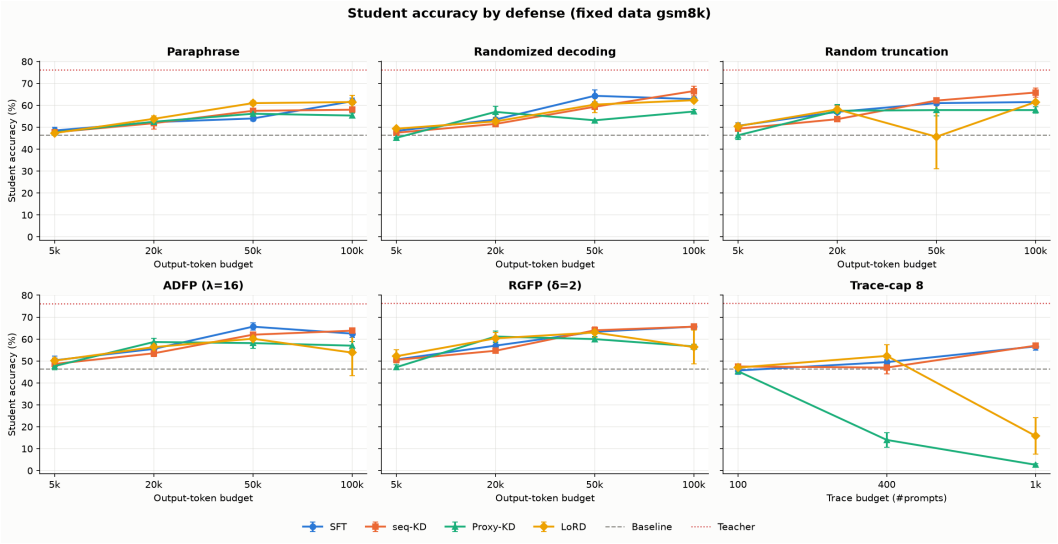


Figure 2: Student accuracy across acquisition budgets and defenses. Each sub-plot compares SFT, seq-KD, Proxy-KD, and LoRD under a different defense. Performance varies with both the distillation method and available supervision, while the trace-cap sub-plot shows the effect of progressively shortening released reasoning traces.

from 50k on. In contrast, SFT and seq-KD peak at 20k and then decline as their answers shrink to Astra's median length of 29 words. SFT stays within 3 points of the best method with Llama (6.3 with Astra) and is among the cheapest to train, whereas LoRD costs 26–40× more compute (Appendix B.6).

Figure 2 extends this comparison across the evaluated defenses. The same pattern persists: the relative performance of SFT, seq-KD, Proxy-KD, and LoRD changes not only with acquisition budget, but also with the defense applied to the teacher. In several cases, increasing the available supervision narrows the gap to the undefended baseline, while some interventions affect particular distillation methods much more strongly than others. Results under other defenses are in Appendix B.1.

4.2 Most defenses cost the teacher more than the student **(Q2)**

Figure 3 shows the drop in student accuracy and teacher accuracy across different distillation methods and defence methods for 50k token budget. The results for other token budget can be found in Appendix B.1.

Almost all defenses lie above the break-even line. Tail shuffle and random truncation cost the Llama teacher 16 and 51 points but most students at most 4. The two fingerprints cost the teacher 6–12 points and change student accuracy by at most 3. Paraphrasing is the exception with Llama: it

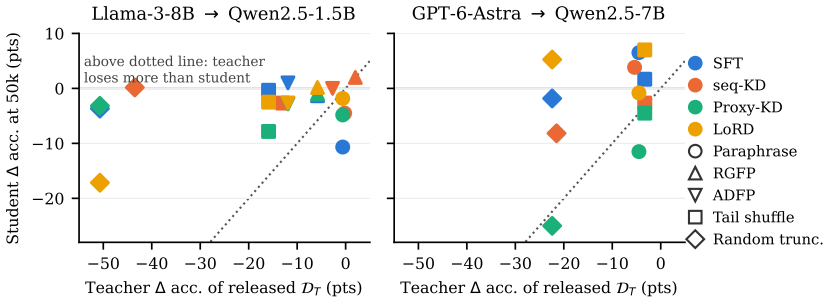


Figure 3: Defense cost to the teacher (x: change in accuracy of the released responses; for seq-KD its five-sample release) against the cost to the student (y: change vs. the undefended student at 50k tokens). Points above the dotted break-even line cost the teacher more than the student. Fingerprints (RGFP, ADFP) require logit access and do not apply to Astra.

Table 3: Teacher × student ablation (accuracy and changes in points vs. the undefended student at the same budget). Static cap 8 keeps the first 8 words of each response; only 1.8% of those responses are correct for either teacher. Words: median length of the SFT student's answers. Initial accuracy: 46.5 (1.5B), 55.0 (7B).

	Undef.	Δ Paraphrase			Δ Rand. trunc.	Static cap 8 (1k resp.)	
Teacher → student	SFT 50k	SFT 20k	SFT 50k	seq-KD 50k	seq-KD 50k	SFT / seq-KD	words
Llama → 1.5B	64.7	−7.5	−10.7	−4.5	+0.2	56.7 / 57.0	56
Astra → 1.5B	62.3	−9.0	−3.5	−5.8	0.0	51.2 / 50.5	52
Llama → 7B	82.0	−11.7	−0.8	−2.0	+0.3	2.7 / 1.8	9
Astra → 7B	77.5	−7.3	+6.5	+3.8	−8.2	1.5 / 2.0	8

costs the teacher under one point and every attack 1.8–10.7 points. Part of this effect is accounting. Paraphrased responses are 43% longer, so a fixed token budget buys fewer of them.

Defenses delay transfer more than they prevent it. Defense effects are largest at small budgets and shrink as the attacker obtains more output. At 100k tokens, every 8B→1.5B cell is within 3.5 points of its undefended counterpart, except paraphrasing against seq-KD (−8.2) and Proxy-KD (−4.7). Tightening the throttle from 50k to 20k tokens costs SFT, seq-KD and LoRD 5–8 points, more than most interventions, without altering any response.

Which defense works depends on the attack. No defense is uniformly effective. With Llama, paraphrasing costs SFT 10.7 points but LoRD 1.8. Tail shuffle costs SFT 0.3 points but Proxy-KD 7.8. With Astra, paraphrasing *raises* SFT and seq-KD accuracy at 50k (by 6.5 and 3.8 points) but lowers Proxy-KD by 11.5, and random truncation costs Proxy-KD 25 points but SFT 1.8. The strongest attack therefore changes with the defense, and a defense evaluated against a single method, typically SFT, can be misjudged in either direction.

4.3 The choice of student model impacts the attack success **(Q3)**

Several conclusions from the Llama→1.5B pair reverse with Astra→7B. Because both the teacher and the student change, we complete the 2 × 2 grid for SFT and seq-KD (Table 3; Appendix B.5) to observe the effect on the attack success based on the choice of student model on different teachers.

Resistance to paraphrasing follows the student. At 20k tokens, paraphrasing slows every pair by 7–12 points. By 50k, the 7B students have recovered with either teacher, whereas both 1.5B students are still 3.5–10.7 points behind. A defense that delays a small student may be insignificant against a larger one trained on the same data.

Table 4: Adaptive acquisition minus fixed acquisition (accuracy points), Llama-3-8B → Qwen2.5-1.5B. Defended cells: paraphrase, tail shuffle and random truncation at 5k–50k tokens. Median seed standard deviation: 1.25 points.

	Undefended, Δ at budget			Defended cells	
Attack	5k	20k	50k	mean Δ	mean |Δ|
SFT	+1.7	+1.3	−1.5	+0.5	1.2
seq-KD	0.0	+0.5	+2.3	−0.4	1.3
Proxy-KD	+1.7	−0.5	−0.2	+0.2	0.8
LoRD	+1.3	+0.8	+0.2	−0.3	4.1

Collapse under truncation also follows the student. With a static cap of 8 words on 1,000 traces limit, both 7B students copy the truncated length (8–9 words) and fall to 2–3% accuracy, far below their initial 55.0%. Both 1.5B students keep writing full-length answers and stay above their initial accuracy. The teacher does not matter here: the same truncated Llama responses yield 56.7% with the 1.5B student and 2.7% with the 7B student. Only one reversal is specific to a teacher–student combination: random truncation costs seq-KD 8.2 points only with Astra→7B (and 0.0 to +0.3 in the other three pairs), so some interactions are specific to the combination.

Lower teacher accuracy does not mean less transfer. Table 3 also shows that a defense's damage to the teacher does not limit distillation. The 1.5B students gain 4–10 points from cap-8 responses that are almost all wrong, learning the approach from the opening words. Conversely, the largest student losses under truncation come from students imitating short outputs. The same happens to Proxy-KD and LoRD with the 1.5B student (2.7% and 15.8%; Appendix B.1). These drops are a failure mode of training on a truncated corpus, not evidence that the defense withheld the capability.

4.4 A CURRICULUM LEARNING-BASED ADAPTIVE ACQUISITION DOES NOT IMPROVE THE ATTACK **(Q4)**

The difference between adaptive acquisition compared to fixed acquisition is that the attacker may have access to additional models, data and tools. Therefore, different attackers might distill using different prompts. One way of implementing adaptive acquisition is using curriculum learning. Here the attacker uses a third-party external model to rank the prompts and selectively trains the student model on more difficult data points according to its budget.

We evaluate this on Llama→1.5B at 5k–50k tokens (at 100k the whole pool is consumed). Table 4 shows that selection changes accuracy by +0.1 points on average, and by 1.7 points in absolute value, comparable to the seed standard deviation (1.25). The only consistent gain is a modest one at 5k tokens (+1.3 to +1.7 for three methods). LoRD's large swings reflect its seed instability rather than selection. Outside LoRD, no defended cell moves by more than 3.8 points, and paraphrasing remains the most damaging defense for SFT at 50k. Within a fixed prompt pool, how much supervision the attacker obtains and how it trains matter far more than which prompts it selects (full grid in Appendix B.7).

4.5 FINGERPRINTS ARE INEFFECTIVE IN DETECTING A DISTILLED STUDENT **(Q5)**

Fingerprinting methods, RGFP and ADFP, are detected with overwhelming confidence in the released responses (z=43–89), but none of the 32 fingerprinted students (four methods, four budgets) is detected on all seeds. Seed-averaged z lies in $[-1.6, 1.2]$, a chance-level rate. On the other hand, a behavioural test (are the student's answers closer to the teacher's than an undistilled copy's?) flags every distilled student at 50k tokens on every seed. The exception is paraphrasing: its students no longer resemble the teacher's wording (similarity gap ≤ 0.017; Appendix B.3). The one defense that consistently slowed the 1.5B students therefore also removes the evidence a provider would use to detect distillation. Fingerprints also need logit access, unavailable for a text-only teacher such as Astra.

4.6 ATTACKS AND DEFENSES MUST BE EVALUATED TOGETHER **(Q6)**

Evaluating attacks and defenses together changes which methods appear effective. With Astra at 50k tokens, paraphrasing improves SFT and seq-KD accuracy but reduces Proxy-KD accuracy, changing which attack performs best. A defense evaluated against SFT alone would therefore receive a different assessment than one evaluated against Proxy-KD. Changes across budgets and students further show that neither an attack's performance against an undefended teacher nor a defense's performance against one attack predicts the outcome of their combination.

Lower student accuracy can also reflect learning to imitate truncated answers rather than the absence of useful supervision. Most tested interventions degrade teacher utility more than student performance, and their effects often weaken at larger budgets. Paraphrasing can reduce transfer while weakening detection based on teacher similarity. A combination that looks favourable from student accuracy alone may therefore be costly to benign users.

4.7 LIMITATIONS

The current evaluation focuses on GSM8K with two teachers, two students, and token budgets up to 100k. These controlled comparisons reveal interactions between attacks and defenses, although the observed rankings may differ across tasks, models, and larger budgets. Adaptive acquisition tests one selection strategy within a fixed prompt pool, and attribution results concern the evaluated fingerprints and detection tests. Broader evaluations can establish how widely these findings hold.

5 RELATED WORK

Distillation and model extraction. Model stealing was first studied for prediction APIs and later extended to NLP systems (Tramèr et al., 2016; Krishna et al., 2020). Recent work shows that text-only teachers can support strong transfer through proxy models, locality-aware objectives, and on-policy generation (Chen et al., 2024; Liang et al., 2025b; Ye et al., 2025). Training on teacher explanations can improve student capabilities (Hsieh et al., 2023), but stylistic imitation does not establish comparable task performance (Gudibande et al., 2024).

Defenses and attribution. Recent defenses alter sampling, logits, or reasoning traces to reduce their value as training targets (Savani et al., 2025; Li et al., 2025; Ding et al., 2025; Ma et al., 2026). Watermarking and fingerprinting instead support attribution or identification of derived models (Sander et al., 2024; Xu et al., 2026). Successful attribution requires detecting a signal in the trained student. It does not require reducing student accuracy. We therefore distinguish transfer suppression, loss of teacher utility, and attribution when evaluating these interventions.

Joint evaluation. Allouah et al. (2026) model distillation as a game in which the teacher must preserve utility and the student can reweight acquired responses. Their adaptive students recover more capability than students trained uniformly on those responses. Libon et al. (2026) show that defense assessments depend on query budgets, available prompts, and API access, using antidistillation sampling as a case study. DistillGuard evaluates output interventions for a fixed teacher and student and reports that their effectiveness varies across tasks (Jiang, 2026). `DistillBench` builds on these evaluations by comparing multiple distillation methods and output interventions under common accounting rules, varying budgets, and teacher and student models. Our experiments test whether defense effects persist across methods and model pairs, while measuring teacher utility and distinguishing token budgets from response budgets.

6 CONCLUSION

We introduced `DistillBench`, a benchmark for jointly evaluating distillation methods and defenses under common access and resource constraints. Our results show that attack and defense effectiveness depends strongly on the distillation method, budget, and student, while many defenses impose greater costs on teacher utility than on capability transfer. These findings highlight the importance of evaluating attacks and defenses together rather than in isolation.

AI USE STATEMENT

LLMs were used to assist with literature discovery, language editing, and the organization and presentation of the manuscript. They were also used during development for coding assistance and debugging. All technical claims, citations, experimental configurations, results, and interpretations were reviewed and verified by the authors. AI-generated text or code was not accepted without author inspection, and the authors remain responsible for the contents of the paper.

ETHICS STATEMENT

This work studies unauthorized model distillation in a controlled research setting with the goal of improving the evaluation of both extraction methods and corresponding defenses. Our experiments focus on measuring capability transfer under explicit access and resource constraints rather than reproducing large-scale abuse. `DistillBench` evaluates attacks and defenses jointly so that protection mechanisms can be assessed not only by their effect on an attacker, but also by their cost to legitimate model utility. Where hosted or proprietary models are used, experiments are conducted through their available interfaces and subject to the applicable access conditions. We do not study mechanisms for bypassing authentication, account controls, rate limits, or other platform safeguards. Release of code, configurations, and generated artifacts will be limited to materials that can be shared consistently with model-provider terms, dataset licenses, and responsible-release considerations.

REPRODUCIBILITY STATEMENT

We provide the experimental protocol, model and dataset choices, acquisition settings, distillation procedures, defenses, budgets, evaluation metrics, and training details in the main paper and appendices. Experiments record random seeds, acquisition inputs, prompt coverage, training configurations, and defense parameters, and results are reported across repeated runs where applicable. We release the implementation, experiment configurations, evaluation scripts, and reproducible artifacts in `https://anonymous.4open.science/r/DSB-36F7/README.md` for experiments that do not depend on restricted hosted-model outputs, subject to the relevant licenses and access conditions. Experiments involving hosted models may not be exactly reproducible because provider-side models and serving configurations can change over time; for these runs, we report the model identifier, interface, generation settings, and other available metadata needed to characterize the experimental conditions.

REFERENCES

Youssef Allouah, Mahdi Haghifam, Sanmi Koyejo, and Reza Shokri. The distillation game: Adaptive attacks & efficient defenses. *arXiv preprint arXiv:2605.22737*, 2026. doi: 10.48550/arXiv.2605.22737. URL `https://arxiv.org/abs/2605.22737`. Presented at the ICML Workshop on Foundations of Deep Generative Models (FoGen), 2026.

Anthropic. Detecting and preventing distillation attacks. `https://www.anthropic.com/news/detecting-and-preventing-distillation-attacks`, February 2026a. Anthropic News. Accessed: 2026-05-05.

Anthropic. Detecting and countering misuse of AI: September 2026, September 2026b. URL `https://www.anthropic.com/threat-intelligence-report-september-2026`. Accessed: 2026-09-22.

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

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

Jiayu Ding, Lei Cui, Li Dong, Nanning Zheng, and Furu Wei. Information-preserving reformulation of reasoning traces for antidistillation. *arXiv preprint arXiv:2510.11545*, 2025. URL https://arxiv.org/abs/2510.11545.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. The false promise of imitating proprietary language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Kz3yckpCN5.

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

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.507. URL https://aclanthology.org/2023.findings-acl.507/.

Bo Jiang. Distillguard: Evaluating defenses against LLM knowledge distillation. *arXiv preprint arXiv:2603.07835*, 2026. doi: 10.48550/arXiv.2603.07835. URL https://arxiv.org/abs/2603.07835.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1317–1327. Association for Computational Linguistics, 2016. doi: 10.18653/v1/D16-1139. URL https://aclanthology.org/D16-1139/.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International conference on machine learning*, pp. 17061–17084. PMLR, 2023.

Kalpesh Krishna, Gaurav Singh Tomar, Ankur P. Parikh, Nicolas Papernot, and Mohit Iyyer. Thieves on sesame street! model extraction of BERT-based APIs. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Byl5NREFDr.

Pingzhi Li, Zhen Tan, Mohan Zhang, Huaizhi Qu, Huan Liu, and Tianlong Chen. DOGe: Defensive output generation for LLM protection against knowledge distillation. *arXiv preprint arXiv:2505.19504*, 2025. URL https://arxiv.org/abs/2505.19504.

Zi Liang, Qingqing Ye, Yanyun Wang, Sen Zhang, Yaxin Xiao, Ronghua Li, Jianliang Xu, and Haibo Hu. "yes, my lord." guiding language model extraction with locality reinforced distillation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1441–1465, 2025a.

Zi Liang, Qingqing Ye, Yanyun Wang, Sen Zhang, Yaxin Xiao, Ronghua Li, Jianliang Xu, and Haibo Hu. "yes, my lord." guiding language model extraction with locality reinforced distillation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1441–1465, 2025b. URL https://aclanthology.org/2025.acl-long.73/.

Lena Libon, Pura Peetathawatchai, Michael Aerni, Daniel Paleka, and Florian Tramèr. What does it mean to break a distillation defense? *arXiv preprint arXiv:2606.25059*, 2026. URL https://arxiv.org/abs/2606.25059.

Xinhang Ma, William Yeoh, Ning Zhang, and Yevgeniy Vorobeychik. Protecting language models against unauthorized distillation through trace rewriting. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11307–11324. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.519. URL https://aclanthology.org/2026.acl-long.519/.

Tom Sander, Pierre Fernandez, Alain Durmus, Matthijs Douze, and Teddy Furon. Watermarking makes language models radioactive. *Advances in Neural Information Processing Systems*, 37:21079–21113, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2567c95fd41459a98a73ba893775d22a-Abstract-Conference.html.

Yash Savani, Asher Trockman, Zhili Feng, Yixuan Even Xu, Avi Schwarzschild, Alexander Robey, Marc Finzi, and J. Zico Kolter. Antidistillation sampling. In *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/aad5e2a483869d9ba3fab491686c3bf2-Paper-Conference.pdf.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction APIs. In *25th USENIX Security Symposium (USENIX Security 16)*, pp. 601–618. USENIX Association, 2016. URL https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/tramer.

Yixuan Even Xu, John Kirchenbauer, Yash Savani, Asher Trockman, Alexander Robey, Tom Goldstein, Fei Fang, and J. Zico Kolter. Antidistillation fingerprinting. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, 2026. URL https://yixuanevenxu.github.io/assets/pdf/icml2026adfingerprinting.pdf.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-box on-policy distillation of large language models. *arXiv preprint arXiv:2511.10643*, 2025. URL https://arxiv.org/abs/2511.10643.

A Detailed Experimental Configuration

This appendix supplies the additional implementation details in Section 3. Table 5 summarizes the experimental configurations for the current instantiation of DistillBench.

Table 5: Experimental configurations for the current instantiation of DistillBench. development.

Component	Configurations
Teachers	Llama 3 8B; GPT-6 Astra
Students	Qwen2.5 1.5B and 7B; corresponding Qwen2.5-Coder variants for coding
Tasks	GSM8K, MATH; MBPP+, LiveCodeBench
Distillation	SFT, Seq-KD, Proxy-KD, and LoRD
Defenses	Paraphrasing, tail shuffle, random and static truncation, RGFP, and ADFP; undefended
Budgets	Output-token and response-count limits

A.1 Acquisition Corpus and Budgets

The service provider limits how much of $\mathcal{D}_T$ the attacker receives. We assume that the attacker cannot evade the limit, for example by distributing queries across accounts. The limit therefore defines the acquisition budget reported for each run.

A *token limit* caps released teacher-output tokens at 5k, 20k, 50k, or 100k. Responses admitted under this limit remain complete. A *request limit* instead caps the number of released responses at 100, 400, or 1,000. We use response-count limits for static truncation and for comparisons in which the number of responses must be held fixed. Under either accounting rule, the attacker receives a random subset of $\mathcal{D}_T$, redrawn for each of three seeds. Actual token count, response count, and number of unique prompts are recorded according to Section 2.3.

For a given seed and budget, SFT, Proxy-KD, and LoRD use the same released responses. The sequence-level KD (seq-KD) corpus contains sampled responses and repeated prompts; the corresponding prompt coverage is reported separately. Length-changing interventions are additionally

compared under response-count accounting so that reduced response length does not automatically increase the number of training examples.

A.2 Distillation Procedures and Training

All four procedures fully fine-tune the Qwen2.5-1.5B student with a learning rate of 10^{-5} for three epochs, effective batch size 16, 3% warm-up, gradient clipping at 1.0, bf16 precision, and a maximum sequence length of 512 tokens. The Qwen2.5-7B student is trained with LoRA adapters (rank 16, α=32, dropout 0.05, all attention and MLP projections) at learning rate 10^{-4}; LoRD uses 10^{-5}, since its objective diverges at 10^{-4}.

Supervised fine-tuning (SFT). The teacher answers each prompt once with greedy decoding. The student is trained to maximize the likelihood of those responses using completion-only cross-entropy, with prompt tokens masked from the loss.

Sequence-level knowledge distillation (seq-KD). Following Kim & Rush (2016), the teacher is sampled five times per prompt at temperature 0.7, and the student is trained on all samples with the same loss as SFT. In Llama $\rightarrow$ 8B pair, sampled responses are correct less often than greedy responses (73.6% versus 86.6%), and a fixed budget covers fewer distinct prompts.

Proxy-model distillation (Proxy-KD). Following Chen et al. (2024), Proxy-KD first fine-tunes a white-box proxy larger than the student, Qwen2.5-3B (learning rate 5×10^{-5}) for the 1.5B student and Qwen2.5-14B (LoRA rank 16, learning rate 10^{-4}) for the 7B student, for one epoch on $\mathcal{D}_T$. It then trains the student to match the frozen proxy's next-token distribution on the teacher responses. The loss is $0.7\,\mathrm{KL}(p_{\mathrm{proxy}} \| p_{\mathrm{student}})T^2 + 0.3\,\mathrm{CE}$ with temperature $T = 2$. The proxy shares the student's tokenizer.

Locality-reinforced distillation (LoRD). Following Liang et al. (2025a), LoRD runs for eight periods. In each period, the student samples two responses per prompt at temperature 1.0 and top-p 0.95. The response the student finds more likely becomes the positive sample and the other the negative sample. A preference-style loss ranks the pair and keeps the student close to the teacher response with a PPO-style clipped term ($\epsilon = 0.2$, $\lambda_1 = 0.5$). During cold start, the teacher response replaces the positive sample when necessary ($\tau_1 = 0.8$).

A.3 Adaptive Acquisition

The adaptive-acquisition attacker draws on the same pool of 1,000 prompts and the same saved teacher responses, so no additional teacher queries are made. Each prompt is scored once using a third (proxy) model's mean per-token loss on the teacher response, which does not depend on the defense, budget or seed. For the proxy model we used the open-weight Qwen3-8B model. Prompts are consumed in decreasing score order until the budget is spent; ties keep a seed-dependent order. At 100k tokens or 1,000 responses the whole pool is consumed, so only smaller budgets distinguish the policy from random selection.

A.4 GPT-6 Astra Teacher

GPT-6 Astra is queried through the Codex CLI at medium reasoning effort with no tools. The attacker sees only the visible answer, never reasoning tokens or logits. The seq-KD release holds five samples per prompt, and its first sample is the single-response release. Astra answers 99.5% of the 200 test questions correctly; its released responses are 96.6% (single) and 96.8% (five samples) correct, against 86.6% and 73.6% for Llama. Astra writes short LaTeX answers (median 29 words, 71.5 student tokens; Llama: 51 words, 84.1 tokens), so the full single-response pool holds 71.5k tokens and the 100k budget releases all 1,000 responses. Paraphrases of Astra's responses are written by Llama-3-8B-Instruct. RGFP and ADFP bias logits during generation and cannot be applied to a text-only teacher.

A.5 Defense and Attribution Implementations

Every evaluated method is applied by the service provider before the response is released and is fixed independently of the attacker's training procedure. The undefended condition releases the corresponding teacher response without an output intervention or fingerprint.

Paraphrasing. Llama3-8B-Instruct is prompted to rewrite each of its own or Astra's response while retaining every number, equation, intermediate result, final answer, and the ordering of the reasoning steps. The intervention changes wording and sentence structure.

Tail shuffle. As a stand-in for randomized generation, this intervention randomly permutes the last 15% of words in each response using a fixed permutation per prompt. The shuffled portion commonly contains the final answer.

Random and static truncation. Random truncation cuts each response after a word count sampled uniformly from $[16, 64]$, fixed per prompt. Static truncation keeps the first $k \in \{8, 16, 24, 32, 48\}$ words. The median undefended response contains 51 words, so the strength sweep ranges from mild to severe truncation.

Red-and-green-list fingerprinting (RGFP). Following Kirchenbauer et al. (2023), RGFP partitions the teacher's vocabulary using a secret hash of the preceding two tokens and adds $\delta = 2$ to each green token's logit. The green-list fraction is $\gamma = 0.5$.

Antidistillation fingerprinting (ADFP). Following Xu et al. (2026), ADFP adds $\lambda p_{\text{proxy}}(g - \langle p_{\text{proxy}}, g\rangle)$ to the teacher logits, where g is the green-list mask and the proxy is Llama-3.2-1B, which shares the teacher's tokenizer. We use $\lambda = 16$ and $\gamma = 0.5$. Both fingerprinting methods act during generation rather than on cached text.

To test a suspected student, we retokenize its outputs with the teacher's tokenizer and perform a one-sided binomial test for over-representation of green tokens over deduplicated context–token pairs. Section 4.5 reports this attribution evaluation.

A.6 Evaluation and Statistical Protocol

Students are evaluated on 200 held-out test questions and each cell is averaged over three seeds unless stated otherwise. Mathematics is scored by final-answer correctness. We also report lexical teacher similarity (LTS), the bag-of-words cosine similarity between a student's answer and the undefended teacher's answer to the same question. The initial student, distilled students, undefended teacher, and defended teacher outputs are evaluated on the corresponding held-out prompts.

For the broader experimental program, mathematics is scored by final-answer correctness and coding by the task's executable tests. The same defense used during acquisition is applied when measuring defended-teacher utility. Results report task scores together with the capability-transfer, matched-defense, and teacher-utility quantities defined in Section 2.4, as well as actual acquisition consumption and seed variation.

B Complete Results

This appendix reports every cell behind Section 4: the full Llama-3-8B-Instruct → Qwen2.5-1.5B grid (§B.1), the access ladder (§B.2), detection (§B.3), GPT-6-Astra → Qwen2.5-7B (§B.4), the teacher × student ablation (§B.5), training cost (§B.6), and adaptive acquisition (§B.7). Cells average three seeds, with these exceptions, all for GPT-6-Astra → Qwen2.5-7B unless noted: Proxy-KD uses two seeds in every cell; LoRD uses two seeds for the undefended runs at 20k–100k tokens and for random truncation at 5k–50k, and one seed for tail shuffle, random truncation at 100k, and the static caps; seq-KD uses one seed for the static caps. For Llama-3-8B-Instruct → Qwen2.5-1.5B, the 12-word static cap uses one seed, a few access-ladder cells use two, and in the adaptive-acquisition comparison the LoRD random-truncation run at 50k tokens and the fixed-acquisition Proxy-KD tail-shuffle run at 50k use two. Single-seed cells report no standard deviation.

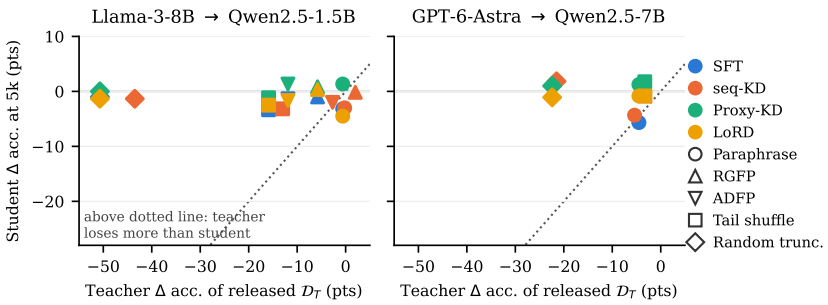


Figure 4: Defense cost to the teacher against the cost to the student in 5k token budget.

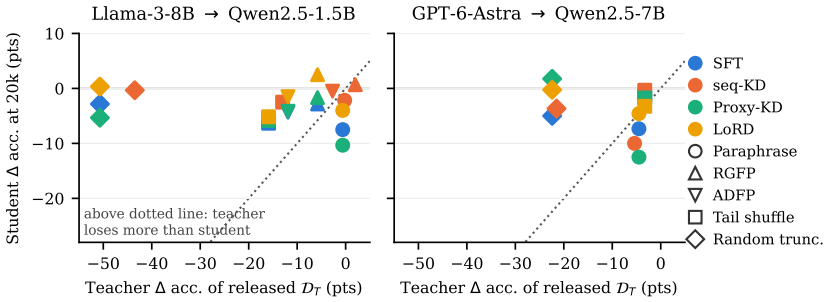


Figure 5: Defense cost to the teacher against the cost to the student in 20k token budget.

B.1 LLAMA-3-8B-INSTRUCT → QWEN2.5-1.5B

Table 6 lists the teacher and student costs plotted in Figure 3 for 50k token budget, and Figure 7 shows the undefended attacks with their lexical teacher similarity (LTS). For readability, we present the cost for the other token budgets in Figure 4, 5, and 6, same format as Figure 3. Accuracy and LTS need not move together: Proxy-KD becomes more teacher-like after its accuracy plateaus, while seq-KD is the most accurate at 100k tokens with the lowest LTS. seq-KD's slower start with Llama reflects its noisier samples (73.6% correct vs. 86.6% for greedy responses); with Astra, whose samples are 96.8% correct, it tracks SFT within 3.2 points. Because seq-KD samples five responses per prompt, a budget buys repeated samples of fewer distinct prompts (Table 7). LoRD is the least stable method: 12 of its 42 cells have a seed standard deviation above 4 points (e.g. random truncation at 50k: 64.0, 45.0 and 28.0%). We run it as published and report this instability as a result. Under static truncation, accuracy follows answer length (Table 8): Proxy-KD and LoRD students shrink their answers toward the cap and collapse, whereas seq-KD students never answer in fewer than 46 words and lose at most 11 points.

This appendix gives every attack × defense × budget cell behind §4: student accuracy and LTS under the token throttle (Tables 9 and 10) and under static truncation with the request throttle (Tables 11 and 12), fingerprint detection on every fingerprinted student (Table 13), and the corresponding figures (Figure 8, 9, 10, 11, 12).

B.2 ACCESS LADDER

Table 14 and Figure 13 varies what the attacker observes of each teacher response, from 50 to 800 prompts. Training on final answers alone leaves the 1.5B student *below* its initial accuracy (39–42% vs.46.5%). Answers with reasoning traces reach 64.0% at 800 prompts, and additionally matching the teacher's last-layer hidden states adds less than one point (64.8%). A layer-20 cosine feature loss with weight $\alpha \in [0.05, 0.5]$ stays within 1.5 points of answer-plus-trace training (falling to 58.2% at α=0.7). In conclusion, having what-box access does not improve distillation attack.

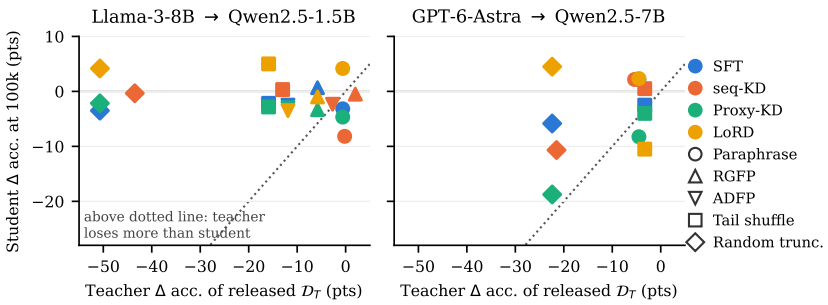


Figure 6: Defense cost to the teacher against the cost to the student in 100k token budget.

Table 6: Cost of each defense to the teacher and to each attack, in accuracy points. The two reference rows give absolute accuracies. Teacher cost is measured on the released responses $\mathcal{D}_T$. Student cost is measured at 50k output tokens for the upper block and at 1,000 traces (vs. an uncapped 1,000-trace run) for the static caps. Bold: the attack with the highest absolute accuracy in each row. *Seed-unstable (per-seed std >10 points).

	Teacher: Δ acc. of $\mathcal{D}_T$		Student: Δ acc. vs. undefended			
Defense	1 greedy/prompt	5 samples/prompt	SFT	seq-KD	Proxy-KD	LoRD
Undefended, 50k tokens	86.6	73.6	**64.7**	62.0	61.0	62.8
Paraphrase	−0.6	−0.2	−10.7	−4.5	−4.8	**−1.8**
RGFP (δ=2)	−5.8	+2.0	−1.3	**+2.0**	−1.0	+0.2
ADFP (λ=16)	−11.9	−2.7	**+1.0**	0.0	−2.8	−2.7
Tail shuffle	−15.9	−13.0	**-0.3**	−2.7	−7.8	−2.5
Random truncation	−50.7	−43.5	−3.7	**+0.2**	−3.2	−17.2*
No cap, 1,000 traces	86.6	73.6	**64.7**	**64.7**	60.0	63.2
Static cap 48 words	−35.6	−24.8	**-1.2**	−1.5	−4.2	−0.2
Static cap 32 words	−66.6	−55.3	−19.7	**-2.3**	−31.8	−17.2*
Static cap 24 words	−80.1	−67.4	−35.3	**-11.0**	−45.3	−47.2*
Static cap 16 words	−84.7	−71.7	−11.7	**-6.5**	−50.0	−58.5
Static cap 8 words	−84.8	−72.3	−8.0	**-7.7**	−57.3	−47.3

B.3 Detection

Fingerprint detection. For RGFP and ADFP we re-tokenize the suspect student's answers with the teacher's tokenizer, test the fraction of green-list tokens over deduplicated context–token pairs against γ=0.5 with a one-sided binomial test, and combine seeds with Fisher's method (Table 13). The released responses are detected at z=42.9 (ADFP) and 45.5 (RGFP), and 73.5 and 88.8 for the five-sample releases. Only 1 of the 96 fingerprinted student runs is flagged at p<0.05, against 3 of the 517 runs trained on non-fingerprinted releases. Restricting scoring to tokens whose bigram context appears in the fingerprinted release gives Fisher p between 0.01 and 0.98 on fingerprinted cells, but also p<0.05 on 14–16 of 175 non-fingerprinted cells, so it does not separate distilled from clean models either.

Behavioural detection. For each test question we compute the gap between the LTS of the suspect student and that of a non-distilled copy of the same base model, and apply a one-sided paired Wilcoxon test (Table 15). Undefended SFT, Proxy-KD and LoRD students are already flagged at 20k tokens. At 100k tokens a faint signal returns under paraphrasing (gap 0.016–0.026). Heavy static truncation hides Proxy-KD and LoRD students only because they have collapsed to short, mostly wrong answers.

B.4 GPT-6-Astra → Qwen2.5-7B

Setup differences from the Llama pair are listed in Appendix A.4. The untrained 7B student scores 55.0% (LTS 0.408); its answers are long (median 198 words) and only 28.5% contain a boxed final

Table 7: The amount of responses seq-KD trains on (seeds 0 / 1 / 2). Because seq-KD samples 5 responses per prompt, a budget buys repeated samples of fewer distinct prompts.

Budget	Traces	Distinct prompts	Tokens used	Traces/prompt	Acc. (%)
5k tokens	59 / 53 / 57	56 / 51 / 55	5.0k–5.1k	1.05	50.7±0.9
20k tokens	230 / 231 / 223	205 / 203 / 205	20.1k	1.12	54.0±0.0
50k tokens	577 / 577 / 567	454 / 450 / 453	50.0k–50.1k	1.27	62.0±1.2
100k tokens	1,148 / 1,167 / 1,145	718 / 730 / 714	100.0k–100.1k	1.60	66.2±1.0
100 traces	100	91 / 94 / 96	8.4k–9.2k	1.07	51.3±0.6
400 traces	400	336 / 330 / 335	33.7k–35.4k	1.20	55.0±1.5
1,000 traces	1,000	659 / 661 / 664	86.1k–87.4k	1.51	64.7±1.3

	Static cap (words)					
Attack	none	48	32	24	16	8
SFT	52	49	42	29	46	56
seq-KD	59	58	54	46	52	54
Proxy-KD	48	49	33	24	18	9
LoRD	54	56	44	24	21	17

Table 8: Median length (words) of student answers on the eval set after training on 1,000 statically truncated traces, averaged over seeds.

answer. At 5k tokens (about 70 responses) no attack moves the student by more than 2 points. By 20k tokens SFT and seq-KD answer in 35–40 words and are 28 points more accurate; much of this early gain is likely the student learning Astra's short, boxed format. Beyond 20k, SFT and seq-KD decline as their answers shrink to Astra's median of 29 words, while Proxy-KD, which matches a proxy's token distributions, does not (Table 17, Figure 14). LoRD learns slowly at learning rate 10^{-5} and reaches 83.0% at 20k tokens at 5×10^{-5} (one seed). Tail shuffle moves only about four words of Astra's short answers and is nearly free for both sides. Under static caps of 8–16 words, SFT, seq-KD and Proxy-KD all copy the truncated length and collapse (Table 18, Figure 15); with only 100 responses, every cap leaves SFT within 8 points of its initial accuracy. LoRD behaves differently (one seed so far): at caps of 8 and 16 words it keeps writing long answers (72–133 words) and reaches 66.5–74.5%, close to or above its uncapped 67.5%, consistent with an objective that ranks the student's own samples rather than imitating the truncated response. At caps of 24 and 32 words, however, its outputs degenerate into repeated tokens (0–3%). We run LoRD as published and report this instability as a result.

B.5 Teacher × student ablation

The cross pairs Llama-3-8B-Instruct → Qwen2.5-7B (Tables 19 and 20) and GPT-6-Astra → Qwen2.5-1.5B (Tables 21 and 22) use SFT and seq-KD with the same D_T files, defenses, budgets and training settings as the main pairs for the same student. Proxy-KD and LoRD are omitted: the available Astra → 1.5B Proxy-KD runs used a Qwen2.5-0.5B proxy and are therefore not comparable. The 7B student collapses under tight static caps with either teacher, and the 1.5B student does not with either teacher.

B.6 Training cost

Table 23 reports the attacker's training compute per run. SFT and seq-KD have similar costs for a given budget (at most 3 minutes for the 1.5B student and 10 for the 7B LoRA student), since the budget fixes the number of training tokens. Proxy-KD costs about 1.7× SFT at 100k tokens: roughly a quarter goes to fine-tuning the proxy, and the student phase is slower than SFT because every step also runs the frozen proxy to obtain its token distributions. LoRD is 26–40× more expensive than SFT at 100k tokens (about 100 and 250 minutes), because each of its eight periods samples two student responses per prompt before the update. Yet LoRD is the most accurate method only at 5k tokens (by at most 0.1 points; Table 2). Per unit of compute, plain SFT is therefore the

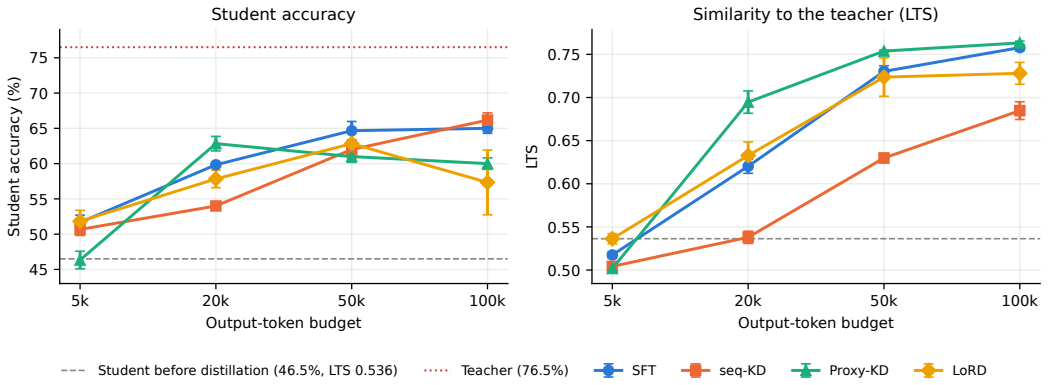


Figure 7: The four attacks without a defense in Llama-8B→1.5B, by output-token budget (mean ± std over 3 seeds). Left: student accuracy; a different attack is best at each budget. Right: lexical similarity to the teacher (LTS). Proxy-KD's LTS keeps rising after its accuracy peaks at 20k, and seq-KD stays the least teacher-like at every budget, although it ends up the most accurate.

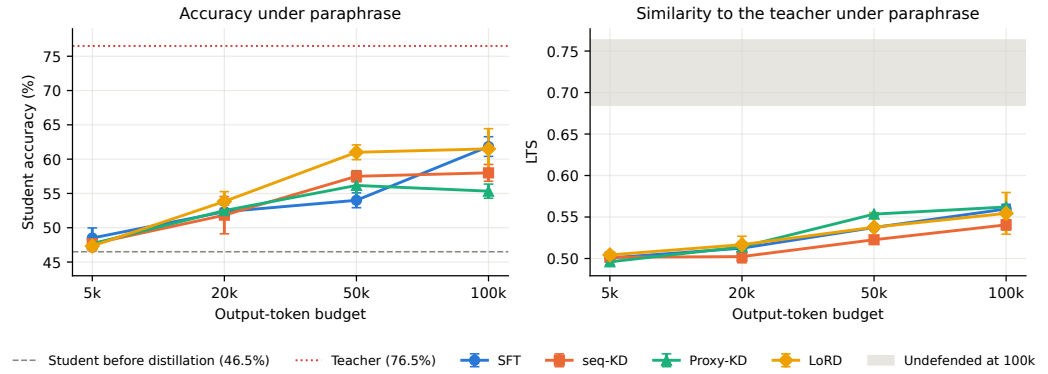


Figure 8: The four attacks under paraphrasing. Left: accuracy. LoRD, second to SFT without a defense at 50k tokens, is the strongest attack here. Right: LTS stays far below the range the undefended students reach at 100k tokens (grey band), which is why behavioural detection fails.

strongest attack in our setting; for the 1.5B student, the evaluation overhead (model loading and answer generation) even exceeds its training time. On the A40 GPUs used for some cells, training takes about 1.6–2.9× longer.

B.7 Adaptive acquisition

The adaptive-acquisition attacker scores each of the 1,000 pool prompts once with the untrained student's mean per-token loss on the teacher response, and consumes prompts in decreasing order of this score until the budget is spent; ties keep a seed-dependent order. The score depends only on the student, not on the defense, budget or seed. All other settings match the fixed-acquisition runs. Table 24 lists every cell.

Table 9: Student accuracy (%) after distillation under each defense, by output-token query budget. Teacher: Llama-3-8B-Instruct; student: Qwen2.5-1.5B; GSM8K. Mean ±std over seeds. Student before distillation: 46.5%; teacher: 76.5%.

Attack	Defense	Query budget (output tokens)			
		5k	20k	50k	100k
SFT	Undefended	51.7±1.0	59.8±0.2	64.7±1.3	65.0±0.7
	Paraphrase	48.5±1.5	52.3±0.5	54.0±1.1	61.8±1.4
	Tail shuffle	48.3±0.6	53.5±0.7	64.3±2.7	62.8±0.8
	Random truncation	50.7±1.3	57.0±2.9	61.0±1.1	61.5±0.4
	ADFP (λ=16)	50.3±1.8	55.5±1.2	65.7±1.6	62.5±1.6
	RGFP (δ=2)	50.7±0.2	57.0±1.6	63.3±0.5	65.7±1.3
seq-KD	Undefended	50.7±0.8	54.0±0.0	62.0±1.2	66.2±1.0
	Paraphrase	47.7±1.0	51.8±2.7	57.5±0.8	58.0±1.2
	Tail shuffle	47.5±0.8	51.5±1.2	59.3±0.5	66.5±2.2
	Random truncation	49.3±0.5	53.7±1.0	62.2±0.8	65.8±2.1
	ADFP (λ=16)	48.7±2.3	53.5±1.5	62.0±0.7	63.8±1.3
	RGFP (δ=2)	50.5±0.4	54.7±0.6	64.0±1.6	65.7±0.8
Proxy-KD	Undefended	46.3±1.2	62.8±1.0	61.0±0.8	60.0±0.8
	Paraphrase	47.7±0.6	52.5±0.8	56.2±2.1	55.3±1.0
	Tail shuffle	45.2±0.2	57.0±2.5	53.2±0.2	57.2±0.9
	Random truncation	46.3±2.1	57.5±2.8	57.8±2.7	57.8±1.4
	ADFP (λ=16)	47.7±1.7	58.7±1.7	58.2±2.5	57.0±1.9
	RGFP (δ=2)	47.2±1.0	61.2±2.6	60.0±1.1	56.7±0.9
LoRD	Undefended	51.8±1.5	57.8±1.2	62.8±1.7	57.3±4.6
	Paraphrase	47.3±0.6	53.8±1.4	61.0±1.1	61.5±2.9
	Tail shuffle	49.3±0.5	52.7±1.9	60.3±3.7	62.3±0.9
	Random truncation	50.5±1.4	58.2±1.5	45.7±14.7	61.5±0.7
	ADFP (λ=16)	50.2±1.3	56.3±1.8	60.2±0.6	53.8±10.5
	RGFP (δ=2)	52.2±3.1	60.3±2.6	63.0±2.3	56.3±7.7

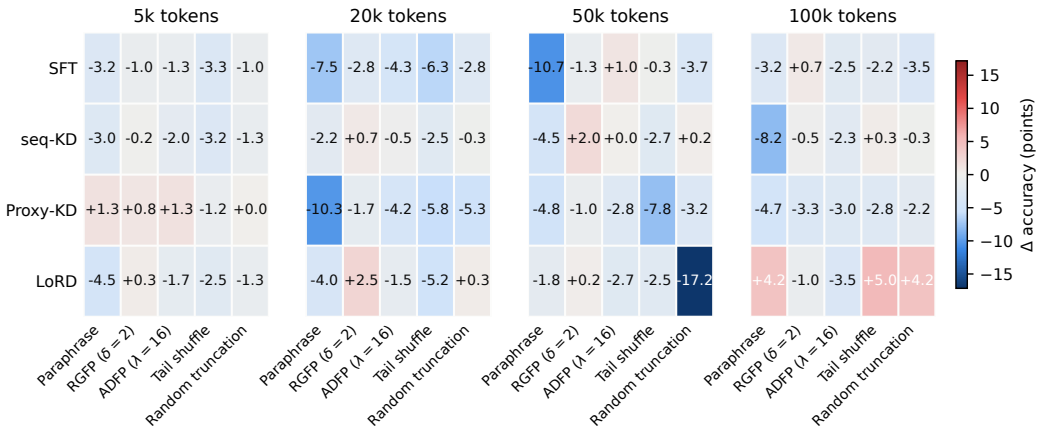


Figure 9: Change in student accuracy relative to the undefended student at the same output-token budget, for each attack and token-throttled defense. GSM8K, Llama-3-8B-Instruct → Qwen2.5-1.5B.

Table 10: Lexical teacher similarity (LTS) of the distilled student, by output-token query budget. Teacher: Llama-3-8B-Instruct; student: Qwen2.5-1.5B; GSM8K. Mean ±std over seeds. Student before distillation: 46.5%; teacher: 76.5%.

Attack	Defense	Query budget (output tokens)			
		5k	20k	50k	100k
SFT	Undefended	0.517±0.004	0.620±0.008	0.730±0.007	0.758±0.004
	Paraphrase	0.501±0.001	0.512±0.005	0.537±0.002	0.560±0.006
	Tail shuffle	0.504±0.004	0.551±0.001	0.681±0.008	0.730±0.004
	Random truncation	0.514±0.001	0.640±0.018	0.755±0.002	0.760±0.001
	ADFP (λ=16)	0.516±0.001	0.565±0.004	0.678±0.000	0.712±0.001
	RGFP (δ=2)	0.517±0.004	0.575±0.001	0.686±0.004	0.730±0.004
seq-KD	Undefended	0.504±0.006	0.538±0.007	0.630±0.006	0.685±0.010
	Paraphrase	0.501±0.006	0.502±0.007	0.523±0.004	0.541±0.006
	Tail shuffle	0.492±0.003	0.522±0.006	0.601±0.008	0.667±0.007
	Random truncation	0.498±0.008	0.541±0.009	0.655±0.016	0.689±0.005
	ADFP (λ=16)	0.492±0.007	0.538±0.004	0.627±0.002	0.673±0.006
	RGFP (δ=2)	0.504±0.004	0.546±0.010	0.641±0.014	0.686±0.006
Proxy-KD	Undefended	0.502±0.004	0.695±0.013	0.754±0.001	0.763±0.002
	Paraphrase	0.496±0.003	0.514±0.005	0.553±0.002	0.562±0.002
	Tail shuffle	0.487±0.002	0.595±0.026	0.661±0.018	0.725±0.013
	Random truncation	0.499±0.003	0.699±0.012	0.755±0.006	0.759±0.004
	ADFP (λ=16)	0.499±0.003	0.624±0.005	0.701±0.006	0.721±0.004
	RGFP (δ=2)	0.500±0.003	0.641±0.018	0.724±0.004	0.748±0.002
LoRD	Undefended	0.536±0.006	0.633±0.016	0.724±0.022	0.728±0.013
	Paraphrase	0.504±0.003	0.517±0.010	0.538±0.017	0.554±0.025
	Tail shuffle	0.518±0.002	0.582±0.007	0.625±0.023	0.704±0.041
	Random truncation	0.528±0.005	0.652±0.012	0.700±0.036	0.751±0.010
	ADFP (λ=16)	0.522±0.005	0.609±0.006	0.686±0.041	0.666±0.067
	RGFP (δ=2)	0.532±0.009	0.620±0.011	0.736±0.008	0.692±0.053

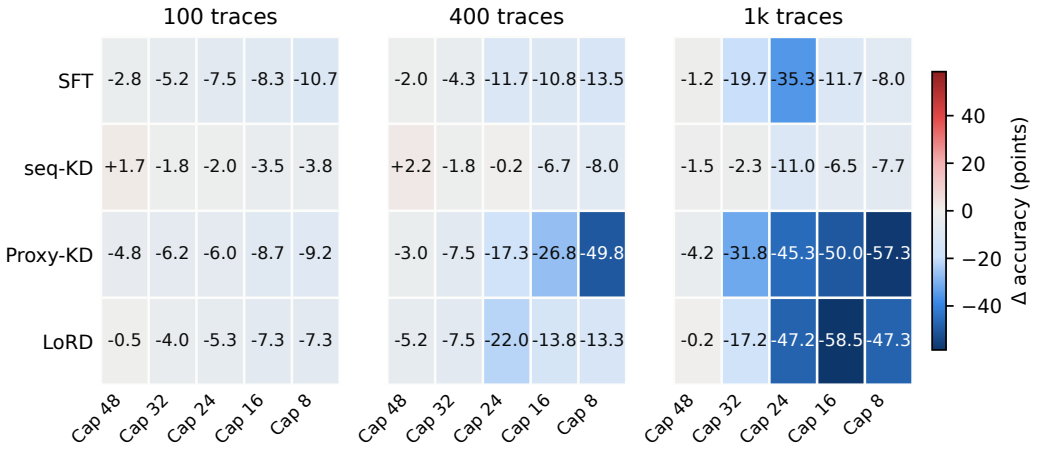


Figure 10: Change in student accuracy relative to the uncapped student at the same request budget, for each attack and static cap. GSM8K, Llama-3-8B-Instruct → Qwen2.5-1.5B.

Table 11: Student accuracy (%) under static truncation, by number of teacher traces queried. "Static cap k" keeps the first k whitespace-separated words of each trace. Teacher: Llama-3-8B-Instruct; student: Qwen2.5-1.5B; GSM8K. Mean $\pm$std over seeds. Student before distillation: 46.5%; teacher: 76.5%.

Attack	Defense	Query budget (traces)		
		100	400	1k
SFT	No cap	56.3±0.6	63.0±2.0	64.7±0.9
	Static cap 8	45.7±1.6	49.5±1.1	56.7±1.6
	Static cap 12	45.0	55.5	–
	Static cap 16	48.0±1.5	52.2±2.4	53.0±1.9
	Static cap 24	48.8±0.5	51.3±1.8	29.3±2.0
	Static cap 32	51.2±0.6	58.7±1.8	45.0±5.1
	Static cap 48	53.5±0.4	61.0±1.2	63.5±1.1
seq-KD	No cap	51.3±0.6	55.0±1.5	64.7±1.3
	Static cap 8	47.5±0.4	47.0±2.9	57.0±1.1
	Static cap 16	47.8±0.8	48.3±2.5	58.2±0.2
	Static cap 24	49.3±0.6	54.8±1.0	53.7±1.2
	Static cap 32	49.5±2.9	53.2±1.6	62.3±1.6
	Static cap 48	53.0±0.4	57.2±2.9	63.2±0.8
Proxy-KD	No cap	54.5±1.1	63.8±0.9	60.0±0.8
	Static cap 8	45.3±1.0	14.0±3.3	2.7±0.5
	Static cap 16	45.8±0.5	37.0±4.3	10.0±2.0
	Static cap 24	48.5±1.2	46.5±2.5	14.7±0.6
	Static cap 32	48.3±0.5	56.3±1.9	28.2±3.0
	Static cap 48	49.7±2.3	60.8±0.8	55.8±2.1
LoRD	No cap	54.3±0.9	65.7±1.9	63.2±4.7
	Static cap 8	47.0±0.4	52.3±5.2	15.8±8.2
	Static cap 16	47.0±1.1	51.8±4.1	4.7±1.8
	Static cap 24	49.0±1.5	43.7±5.4	16.0±19.8
	Static cap 32	50.3±3.6	58.2±5.4	46.0±18.5
	Static cap 48	53.8±1.7	60.5±1.4	63.0±1.5

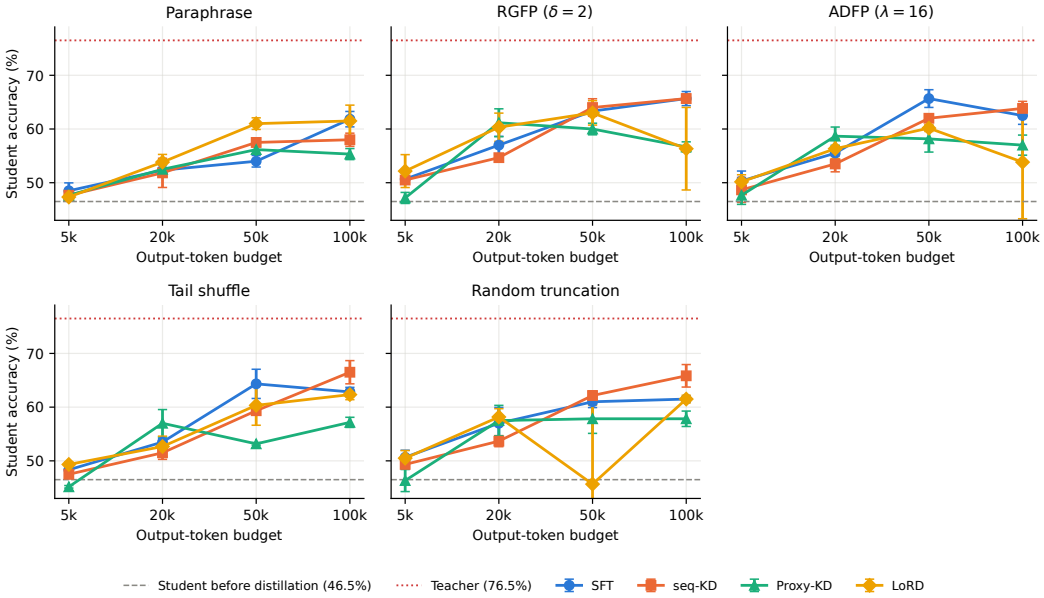


Figure 11: Student accuracy of the four attacks under each token-throttled defense (mean $\pm$ std over seeds). GSM8K, Llama-3-8B-Instruct $\rightarrow$ Qwen2.5-1.5B.

Table 12: LTS of the distilled student under static truncation, by number of teacher traces queried. Teacher: Llama-3-8B-Instruct; student: Qwen2.5-1.5B; GSM8K. Mean ±std over seeds. Student before distillation: 46.5%; teacher: 76.5%.

Attack	Defense	Query budget (traces)		
		100	400	1k
SFT	No cap	0.533±0.006	0.680±0.004	0.756±0.006
	Static cap 8	0.498±0.003	0.551±0.005	0.671±0.002
	Static cap 12	0.497	0.601	–
	Static cap 16	0.503±0.004	0.616±0.001	0.723±0.004
	Static cap 24	0.509±0.003	0.642±0.011	0.691±0.002
	Static cap 32	0.511±0.002	0.654±0.011	0.733±0.005
	Static cap 48	0.522±0.007	0.670±0.010	0.759±0.003
seq-KD	No cap	0.512±0.005	0.570±0.011	0.667±0.009
	Static cap 8	0.490±0.006	0.500±0.015	0.619±0.002
	Static cap 16	0.492±0.004	0.538±0.013	0.659±0.006
	Static cap 24	0.499±0.006	0.571±0.006	0.667±0.008
	Static cap 32	0.506±0.002	0.567±0.019	0.692±0.001
	Static cap 48	0.509±0.006	0.592±0.018	0.687±0.007
Proxy-KD	No cap	0.532±0.003	0.738±0.004	0.763±0.002
	Static cap 8	0.491±0.002	0.385±0.031	0.486±0.006
	Static cap 16	0.494±0.005	0.601±0.029	0.618±0.009
	Static cap 24	0.504±0.008	0.684±0.012	0.672±0.000
	Static cap 32	0.506±0.007	0.710±0.006	0.724±0.002
	Static cap 48	0.509±0.006	0.728±0.008	0.764±0.004
LoRD	No cap	0.560±0.010	0.696±0.012	0.747±0.012
	Static cap 8	0.504±0.004	0.614±0.034	0.508±0.042
	Static cap 16	0.528±0.007	0.680±0.025	0.565±0.092
	Static cap 24	0.535±0.001	0.676±0.010	0.573±0.095
	Static cap 32	0.532±0.011	0.685±0.044	0.723±0.038
	Static cap 48	0.547±0.007	0.669±0.012	0.749±0.011

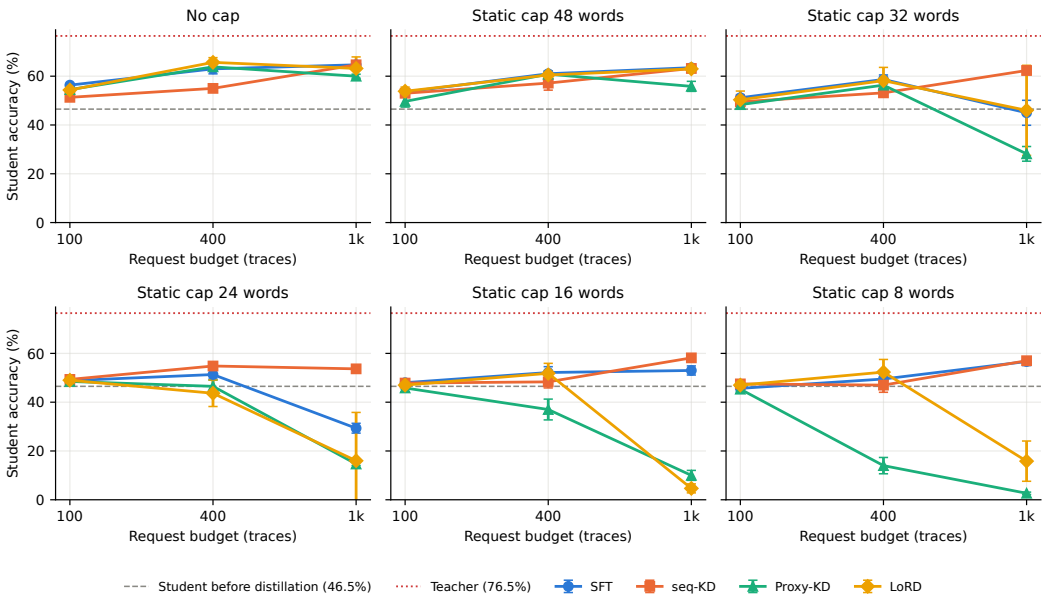


Figure 12: Student accuracy of the four attacks under each static cap, by request budget. Collapses below the untrained student come from students imitating short answers. GSM8K, Llama-3-8B-Instruct → Qwen2.5-1.5B.

Table 13: Fingerprint detection on students distilled from fingerprinted teacher outputs (GSM8K, Llama-3-8B-Instruct → Qwen2.5-1.5B). Green fraction and z are averaged over seeds; p is Fisher-combined over seeds. "ctx" scores only tokens whose bigram context occurs in the fingerprinted release. "Detected" requires $p < 0.05$ on every seed (default detector).

Student, defense	Budget	Green frac.	z	Fisher p	Fisher p (ctx)	Detected
SFT, ADFP (λ=16)	5k	0.4963	-1.03	0.985	0.890	no
	20k	0.4973	-0.70	0.948	0.291	no
	50k	0.5009	0.19	0.525	0.012	no
	100k	0.4997	-0.06	0.698	0.227	no
SFT, RGFP (δ=2)	5k	0.4965	-0.98	0.977	0.755	no
	20k	0.4990	-0.26	0.799	0.051	no
	50k	0.4992	-0.18	0.748	0.096	no
	100k	0.5017	0.33	0.386	0.219	no
seq-KD, ADFP (λ=16)	5k	0.4955	-1.28	0.995	0.983	no
	20k	0.4949	-1.36	0.996	0.591	no
	50k	0.4983	-0.39	0.861	0.152	no
	100k	0.4988	-0.25	0.801	0.289	no
seq-KD, RGFP (δ=2)	5k	0.4960	-1.13	0.991	0.981	no
	20k	0.4949	-1.36	0.997	0.266	no
	50k	0.4951	-1.12	0.990	0.611	no
	100k	0.4981	-0.40	0.863	0.412	no
Proxy-KD, ADFP (λ=16)	5k	0.4970	-0.85	0.947	0.865	no
	20k	0.4984	-0.39	0.859	0.045	no
	50k	0.5040	0.75	0.178	0.142	no
	100k	0.5067	1.24	0.025	0.208	no
Proxy-KD, RGFP (δ=2)	5k	0.4943	-1.62	0.998	0.869	no
	20k	0.4998	-0.04	0.663	0.023	no
	50k	0.5047	0.87	0.127	0.071	no
	100k	0.5012	0.21	0.519	0.625	no
LoRD, ADFP (λ=16)	5k	0.4973	-0.73	0.944	0.629	no
	20k	0.4962	-0.96	0.980	0.445	no
	50k	0.5005	0.03	0.449	0.221	no
	100k	0.5039	0.60	0.116	0.058	no
LoRD, RGFP (δ=2)	5k	0.4965	-0.99	0.965	0.720	no
	20k	0.4939	-1.55	0.999	0.711	no
	50k	0.5037	0.70	0.197	0.188	no
	100k	0.5049	0.90	0.072	0.082	no

Table 14: Access regime performance ladder: student accuracy (%) by number of training samples. Teacher accuracy is 76.5%.

Access Regime	Training samples				
	50	100	200	400	800
Answer-only	41.2±1.8	42.0±0.7	39.3±1.0	39.5±1.5	40.8±0.2
Answer-and-trace	46.3±1.4	49.5±0.7	52.2±2.7	62.3±2.0	64.0±2.5
Hidden state (last layer)	46.7±1.6	48.0±1.2	53.5±2.2	62.0±1.8	64.8±1.6
Hidden state L20, cos α=0.05	48.2±1.9	49.7±0.5	53.5±1.1	61.8±1.4	63.3±2.3
Hidden state L20, cos α=0.1	47.5±1.1	51.0±0.8	52.2±1.3	61.0±0.4	63.0±2.1
Hidden state L20, cos α=0.2	45.5±0.4	48.5±0.8	53.2±1.8	61.7±1.2	62.5±1.9
Hidden state L20, cos α=0.3	45.5±1.1	50.0±1.8	52.2±2.6	61.5±2.1	63.7±2.2
Hidden state L20, cos α=0.5	46.8±1.2	49.7±1.2	54.0±2.2	60.0±1.1	63.2±0.6
Hidden state L20, cos α=0.7	48.2±0.5	50.0±1.8	52.2±2.2	61.8±0.9	58.2±2.5

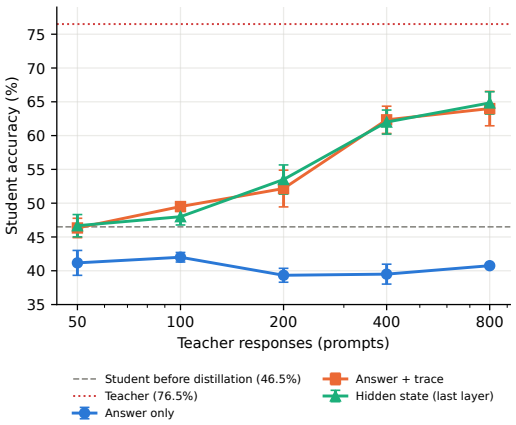


Figure 13: White-box access ladder: student accuracy by the number of teacher responses and by what the attacker sees of each response. GSM8K, Llama-3-8B-Instruct → Qwen2.5-1.5B.

Table 15: Behavioural detection. Entry: mean per-question similarity gap (student − non-distilled control, in LTS) and the number of seeds on which the paired Wilcoxon test rejects at $p<0.05$.

Defense (50k tokens)	SFT	seq-KD	Proxy-KD	LoRD
Undefended	+.193 (3/3)	+.106 (3/3)	+.217 (3/3)	+.188 (3/3)
Paraphrase	+.004 (0/3)	−.005 (0/3)	+.017 (1/3)	+.002 (1/3)
RGFP (δ=2)	+.153 (3/3)	+.114 (3/3)	+.187 (3/3)	+.200 (3/3)
ADFP (λ=16)	+.145 (3/3)	+.105 (3/3)	+.164 (3/3)	+.150 (3/3)
Tail shuffle	+.143 (3/3)	+.076 (3/3)	+.124 (3/3)	+.088 (3/3)
Random truncation	+.218 (3/3)	+.124 (3/3)	+.218 (3/3)	+.163 (3/3)
Static cap 48 (1k traces)	+.222 (3/3)	+.163 (3/3)	+.227 (3/3)	+.212 (3/3)
Static cap 8 (1k traces)	+.134 (3/3)	+.090 (3/3)	−.051 (0/3)	−.029 (0/3)

Table 16: Cost of each defense to the teacher and to each attack, GPT-6-Astra → Qwen2.5-7B, in accuracy points (layout as Table 6). The paraphrase is written by Llama-3-8B-Instruct.

	Teacher: Δ acc. of $\mathcal{D}_T$		Student: Δ acc. vs. undefended (50k)			
Defense	1 response	5 samples	SFT	seq-KD	Proxy-KD	LoRD
Undefended (absolute)	96.6	96.8	77.5	80.7	83.8	65.0
Paraphrase	−4.5	−5.4	+6.5	+3.8	−11.5	−0.8
Tail shuffle	−3.3	−3.3	+1.7	−2.7	−4.5	+7.0
Random truncation	−22.4	−21.5	−1.8	−8.2	−25.0	+5.3

Table 17: Student accuracy (%) by output-token budget, GPT-6-Astra → Qwen2.5-7B, GSM8K. Mean ±std over seeds; –: not run. Student before distillation: 55.0%.

Attack	Defense	5k	20k	50k	100k
SFT	Undefended	56.7±0.2	83.2±0.6	77.5±1.9	79.7±0.9
	Paraphrase	51.0±2.0	75.8±0.8	84.0±2.1	81.8±1.4
	Tail shuffle	–	81.8±2.7	79.2±1.9	77.2±0.2
	Random truncation	–	78.2±3.9	75.7±1.0	73.8±0.5
seq-KD	Undefended	55.8±0.6	83.0±0.4	80.7±1.6	78.0±2.3
	Paraphrase	51.5±0.4	73.0±2.2	84.5±1.8	80.2±1.6
	Tail shuffle	55.0±0.4	82.7±2.7	78.0±3.2	78.5±1.8
	Random truncation	57.7±1.2	79.3±2.0	72.5±2.9	67.3±2.0
Proxy-KD	Undefended	51.0±0.5	68.0±1.5	83.8±0.2	83.8±0.2
	Paraphrase	52.2±0.2	55.5±1.0	72.2±0.2	75.5±1.0
	Tail shuffle	52.8±1.2	66.2±0.8	79.2±1.8	79.8±0.2
	Random truncation	52.0±1.5	69.8±4.2	58.8±0.2	65.0±5.5
LoRD	Undefended	56.8±0.8	57.8±0.8	65.0±3.0	67.5±1.5
	Paraphrase	56.0±1.1	53.2±0.6	64.2±0.6	69.8±1.6
	Tail shuffle	56.0	54.5	72.0	–
	Random truncation	55.8±0.2	57.5±1.0	70.2±2.2	72.0

Table 18: Student accuracy (%) under static truncation (first k words kept) by number of released responses, GPT-6-Astra → Qwen2.5-7B. *Words @1k*: median answer length (words) of the student trained on 1,000 responses. ¶LoRD's outputs degenerate into repeated tokens (e.g. runs of "$" or zero-width spaces); –: not yet run. Student before distillation: 55.0%.

Attack	Responses	No cap	48	32	24	16	8
SFT	100	60.3±1.4	61.2±1.2	59.8±2.0	56.8±0.2	54.5±0.8	47.5±5.0
	400	82.0±0.7	82.0±0.8	80.5±1.5	52.5±3.6	22.5±3.9	1.7±0.9
	1,000	79.5±1.9	78.7±0.8	65.3±2.0	42.3±2.7	10.3±1.6	1.5±0.7
	words @1k	28	28	28	24	17	8
seq-KD	100	61.0	62.0	60.5	55.5	53.5	45.0
	400	84.0	83.5	82.0	60.0	30.0	1.5
	1,000	80.0	74.0	63.0	46.5	13.0	2.0
	words @1k	29	28	29	24	17	8
Proxy-KD	100	53.8±1.2	53.0±2.0	54.5±0.5	53.2±0.8	52.8±0.2	51.8±0.8
	400	78.2±1.2	79.8±2.2	70.0±6.0	65.5±1.0	61.2±2.8	5.2±1.2
	1,000	83.0±0.5	81.0±0.5	74.5±2.5	59.2±0.8	23.5±2.5	4.5±0.0
	words @1k	32	31	30	28	21	4
LoRD	100	–	–	–	56.0	53.5	55.0
	400	–	–	–	–	–	58.0
	1,000	67.5±1.5	–	3.0¶	0.0¶	66.5	74.5
	words @1k	141	–	1	5	133	72

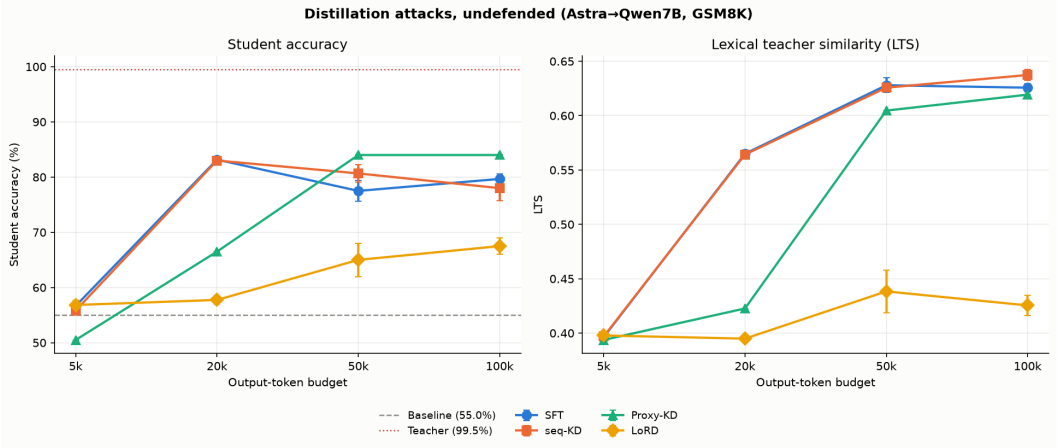


Figure 14: Undefended attacks, GPT-6-Astra → Qwen2.5-7B (mean ± std over seeds). SFT and seq-KD peak at 20k tokens and then decline. Proxy-KD lags at 20k but is the strongest attack from 50k on. LoRD improves slowly.

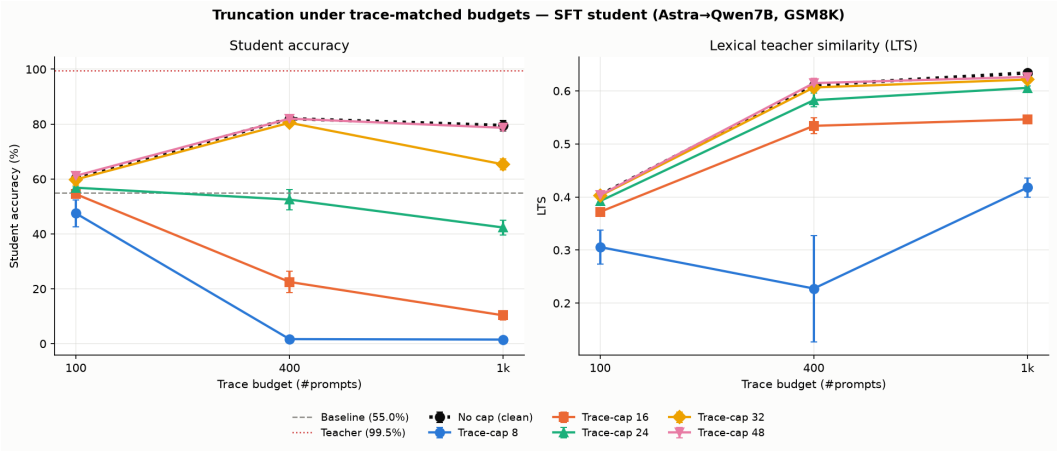


Figure 15: Static truncation, SFT student, GPT-6-Astra → Qwen2.5-7B, by number of released responses. At caps of 16 and 8 words the student falls far below its untrained accuracy (dashed).

Table 19: Student accuracy (%) by output-token budget, Llama-3-8B-Instruct → Qwen2.5-7B, GSM8K. Mean ±std over seeds; –: not run. Student before distillation: 55.0%.

Attack	Defense	5k	20k	50k	100k
SFT	Undefended	–	77.7±2.1	82.0±1.6	82.0±1.2
	Paraphrase	52.3±0.8	66.0±4.3	81.2±2.5	80.7±0.2
	Tail shuffle	53.8±0.6	79.3±0.2	82.0±1.5	80.2±0.8
	Random truncation	59.5±1.8	81.7±0.2	80.5±2.2	78.7±2.0
	RGFP (δ=2)	53.5±0.7	78.8±2.0	80.5±1.5	81.8±0.2
	ADFP (λ=16)	56.3±1.9	75.8±4.8	78.8±1.2	80.8±1.9
seq-KD	Undefended	55.5±0.8	73.2±1.4	80.3±2.7	81.8±1.2
	Paraphrase	51.2±1.0	63.5±0.4	78.3±1.2	82.3±0.6
	Tail shuffle	51.3±1.0	74.8±2.0	80.5±2.5	81.5±1.2
	Random truncation	52.7±1.2	79.8±1.4	80.7±1.2	79.8±1.8
	RGFP (δ=2)	55.5±1.1	74.2±1.4	79.8±2.5	81.7±0.9
	ADFP (λ=16)	55.3±0.5	71.8±0.2	80.3±1.4	80.5±0.4

Table 20: Student accuracy (%) under static truncation (first k words kept) by number of released responses, Llama-3-8B-Instruct → Qwen2.5-7B. *Words @1k*: median answer length (words) of the student trained on 1,000 responses. Student before distillation: 55.0%.

Attack	Responses	No cap	48	32	24	16	8
SFT	100	61.7±1.7	61.7±1.0	59.3±2.2	51.2±6.6	41.7±3.7	32.0±5.7
	400	83.0±0.4	80.7±1.2	72.8±3.4	38.3±5.8	10.7±3.7	6.0±1.5
	1,000	81.5±0.7	70.3±1.0	28.0±2.3	9.3±0.6	2.5±0.4	2.7±0.6
	words @1k	50	48	34	24	16	9
seq-KD	100	57.2±0.2	59.7±1.3	55.0±2.5	50.0±1.1	35.5±3.2	32.3±7.2
	400	80.7±1.2	81.3±2.2	75.3±2.2	49.7±10.7	34.3±11.3	12.7±10.1
	1,000	81.0±1.9	61.2±6.6	22.8±1.7	6.5±1.5	3.5±0.4	1.8±1.2
	words @1k	49	47	33	25	17	10

Table 21: Student accuracy (%) by output-token budget, GPT-6-Astra → Qwen2.5-1.5B, GSM8K. Mean ±std over seeds; –: not run. Student before distillation: 46.5%.

Attack	Defense	5k	20k	50k	100k
SFT	Undefended	50.7±1.3	63.0±0.7	62.3±0.6	62.2±1.7
	Paraphrase	49.7±2.0	54.0±1.5	58.8±0.2	60.8±1.0
	Tail shuffle	51.5±0.4	58.5±3.2	63.3±1.6	66.2±1.3
	Random truncation	52.3±0.6	63.0±0.4	62.5±1.1	61.5±1.6
seq-KD	Undefended	52.5±0.7	62.5±1.2	64.2±2.1	63.2±0.6
	Paraphrase	49.8±0.9	52.2±1.4	58.3±0.8	60.2±1.2
	Tail shuffle	52.0±0.8	59.5±2.3	63.5±1.4	63.2±0.5
	Random truncation	52.5±0.4	63.2±2.2	64.2±1.7	65.0±1.4

Attack	Responses	No cap	48	32	24	16	8
SFT	100	–	52.0±1.1	53.0±1.5	53.3±0.5	50.7±1.2	50.3±1.7
	400	–	64.0±0.8	61.5±1.8	62.2±1.2	57.7±2.5	55.3±0.8
	1,000	–	63.2±1.7	62.2±0.8	62.3±2.0	58.7±2.2	51.2±0.8
	words @1k	–	56	58	56	48	52
seq-KD	100	–	54.0±1.8	54.0±0.7	52.8±2.0	52.2±1.4	50.7±1.0
	400	–	63.8±1.0	63.2±1.2	62.5±0.7	59.5±2.9	58.3±1.2
	1,000	–	65.2±1.2	64.8±0.9	63.7±2.1	59.2±1.0	50.5±1.5
	words @1k	–	52	52	51	47	54

Table 22: Student accuracy (%) under static truncation (first k words kept) by number of released responses, GPT-6-Astra → Qwen2.5-1.5B. *Words @1k*: median answer length (words) of the student trained on 1,000 responses. Student before distillation: 46.5%.

Table 23: Training time per run (median minutes on one NVIDIA H100) by output-token budget, pooled over the undefended and token-throttled defense cells and seeds. SFT, seq-KD and Proxy-KD times are the trainer-reported runtimes; Proxy-KD sums proxy fine-tuning and student distillation. *LoRD uses its own training loop, so its time is the job's wall-clock time minus the evaluation overhead of the other methods in the same cell (model loading and generation for the initial and distilled students on the 200 test questions). LoRD students write longer answers, so this slightly overestimates LoRD. –: no H100 run (these cells ran on A40s). Teacher queries are excluded; all methods consume the same released responses.

	Llama-3-8B → Qwen2.5-1.5B (full FT)				GPT-6-Astra → Qwen2.5-7B (LoRA)			
Attack	5k	20k	50k	100k	5k	20k	50k	100k
SFT	0.2	–	–	2.5	0.5	3.1	7.7	9.6
seq-KD	0.2	0.6	1.4	2.9	0.5	1.9	4.7	9.4
Proxy-KD	0.3	1.1	2.6	4.3	1.3	5.0	12.4	16.5
proxy / student	0.1/0.2	0.3/0.8	0.7/1.9	1.1/3.2	0.5/0.9	1.7/3.3	4.2/8.3	5.6/10.9
LoRD*	≈16	≈34	≈70	≈100	≈15	≈63	≈152	≈247
Eval. overhead	15.9	13.2	10.9	10.3	12.0	9.1	4.6	4.4

Table 24: Adaptive acquisition vs. fixed acquisition, Llama-3-8B-Instruct → Qwen2.5-1.5B. Each entry: fixed-acquisition accuracy → adaptive-acquisition accuracy ±std (difference). At 100k tokens both settings consume the whole pool. ‡Proxy-KD adaptive-acquisition runs used a Qwen2.5-0.5B proxy and are compared with fixed-acquisition runs using the same proxy; its paraphrase cells are omitted because no matched fixed-acquisition run exists.

Attack	Defense	5k	20k	50k
SFT	Undefended	51.7 → 53.3±0.2 (+1.7)	59.8 → 61.2±1.9 (+1.3)	64.7 → 63.2±1.3 (-1.5)
	Paraphrase	48.5 → 46.5±1.1 (-2.0)	52.3 → 52.7±0.9 (+0.3)	54.0 → 56.8±1.0 (+2.8)
	Tail shuffle	48.3 → 48.8±1.0 (+0.5)	53.5 → 54.0±1.1 (+0.5)	64.3 → 64.3±0.8 (+0.0)
	Random truncation	50.7 → 52.7±1.3 (+2.0)	57.0 → 55.8±0.8 (-1.2)	61.0 → 62.2±1.3 (+1.2)
seq-KD	Undefended	50.7 → 50.7±1.3 (+0.0)	54.0 → 54.5±1.1 (+0.5)	62.0 → 64.3±1.8 (+2.3)
	Paraphrase	47.7 → 48.3±0.6 (+0.7)	51.8 → 53.7±1.6 (+1.8)	57.5 → 56.2±0.5 (-1.3)
	Tail shuffle	47.5 → 47.8±0.8 (+0.3)	51.5 → 52.3±0.2 (+0.8)	59.3 → 55.5±1.1 (-3.8)
	Random truncation	49.3 → 48.7±1.9 (-0.7)	53.7 → 51.5±0.4 (-2.2)	62.2 → 62.5±1.4 (+0.3)
Proxy-KD‡	Undefended	42.2 → 43.8±0.6 (+1.7)	57.8 → 57.3±2.2 (-0.5)	54.2 → 54.0±1.2 (-0.2)
	Tail shuffle	41.0 → 41.3±1.9 (+0.3)	49.5 → 49.7±0.5 (+0.2)	51.0 → 50.0±1.9 (-1.0)
	Random truncation	45.3 → 44.7±1.0 (-0.7)	56.2 → 56.5±1.2 (+0.3)	48.8 → 50.8±1.4 (+2.0)
LoRD	Undefended	51.8 → 53.2±3.3 (+1.3)	57.8 → 58.7±1.7 (+0.8)	62.8 → 63.0±3.2 (+0.2)
	Paraphrase	47.3 → 50.8±2.0 (+3.5)	53.8 → 55.3±2.3 (+1.5)	61.0 → 56.7±1.4 (-4.3)
	Tail shuffle	49.3 → 47.8±0.9 (-1.5)	52.7 → 53.8±1.6 (+1.2)	60.3 → 54.0±1.8 (-6.3)
	Random truncation	50.5 → 51.3±1.0 (+0.8)	58.2 → 50.5±1.5 (-7.7)	45.7 → 56.0±0.5 (+10.3)