DO DEFENSES AGAINST LLM EXTRACTION WORK ACROSS ATTACKS? A LIFECYCLE BENCHMARK OF BLACK-BOX MODEL EXTRACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) deployed through text-only APIs face model extraction risks, as adversaries can collect their responses to train surrogates that reproduce their capabilities. While prior work has developed diverse attacks and defenses, evaluations remain fragmented across access assumptions, model configurations, query budgets, and security objectives, limiting comparability across methods. To address this gap, we introduce a unified benchmark covering six extraction attacks, ten defenses, and two adaptive attacks that paraphrase or back-translate protected responses before surrogate training. The benchmark controls model configurations, query data, budgets, and held-out evaluation conditions within each comparison while preserving attack-specific querying and training procedures. We measure surrogate capability, fidelity to the victim, output quality using Rep-4, and query-budget sensitivity; defenses use their own security metrics paired with surrogate performance. For the adaptive attacks, we jointly measure provenance-detector scores and the capability and fidelity of surrogates trained on rewritten responses. The benchmark thus provides a reproducible basis for comparing extraction methods and their interactions with defenses under text-only access. Code and artifacts are available in our anonymous repository at `https://anonymous.4open.science/r/mea-benchmark-20B1`.

1 INTRODUCTION

Many large language models (LLMs) are deployed as proprietary services through text-only APIs (OpenAI et al., 2024; Birch et al., 2023). Although API access hides model parameters, it exposes the input–output behavior produced through substantial investments in data, computation, and engineering (Grattafiori et al., 2024; Tramèr et al., 2016). An adversary can repeatedly query such a model, collect its responses, and use them as supervision for a smaller surrogate (Birch et al., 2023; Liang et al., 2025b). This functional form of model extraction can reproduce valuable task capabilities without recovering the victim's parameters or training data, creating a direct risk to the confidentiality and economic value of deployed models (Zhao et al., 2025; Ti et al., 2025).

Research on LLM extraction has consequently produced increasingly diverse attacks and defenses. However, their empirical results are difficult to reconcile because studies vary simultaneously in access assumptions, victim–surrogate configurations, query data, budgets, and evaluation objectives (Oliynyk et al., 2025a;b; Zhao et al., 2025). An attack that appears stronger in one study may therefore benefit from a different model, dataset, or budget rather than a more effective extraction procedure. Defense claims are even less directly comparable: anti-distillation measures reductions in surrogate performance, provenance and lineage methods use model-level detector scores, and query detectors classify suspicious traffic (Savani et al., 2025; Xu et al., 2026; Yan et al., 2026; Liu et al., 2026). Moreover, measuring paraphrasing or back-translation only on protected responses does not show whether a surrogate trained on the rewritten text retains the defense's detector signal or useful capability (Liang et al., 2025a; Pan et al., 2025). Existing benchmarks cover broader privacy threats, knowledge-base leakage, or individual defense families, but, to the best of our knowledge, none compares diverse attacks under matched conditions, tests defenses against their stated objectives, and evaluates adaptive attacks after surrogate training (Liu et al., 2022; Huang et al., 2024; Qi et al., 2026; Jiang, 2026). As attacks and defenses proliferate, this fragmentation makes reported

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.

improvements difficult to attribute and can lead to conflicting conclusions about both extraction risk and defense effectiveness. This raises a central question: *How can black-box LLM extraction methods be compared under matched conditions while preserving attack procedures and respecting defense-specific objectives?*

Answering this question requires overcoming three limitations in current evaluation practice. **(1) Confounded attack comparisons.** Extraction methods differ in how they acquire queries, construct supervision, and optimize the surrogate (Birch et al., 2023; Liang et al., 2025b; Chen et al., 2026; Ye et al., 2026; Li et al., 2026). Because model, data, and budget choices can materially change the measured effectiveness of an attack, comparisons that vary these conditions cannot isolate the contribution of the extraction method itself (Oliynyk et al., 2025a;b). Attack comparisons must therefore match the victim and surrogate models, data, and budgets while retaining each method's query-acquisition and training procedure. **(2) Incommensurable defense outcomes.** Reducing capability transfer, embedding provenance, verifying lineage, and detecting suspicious queries represent different security objectives rather than interchangeable notions of defense success. Their evaluations must also expose changes in surrogate capability and fidelity as well as false attribution, which can otherwise make a defense appear effective for the wrong reason (Liu et al., 2024). Treating these outcomes as a common defense score consequently obscures both the guarantee provided by a method and the cost at which it is obtained. **(3) Incomplete adaptive evaluation.** Paraphrasing or back-translation may weaken an embedded signal while also destroying the supervision needed for extraction, or preserve useful supervision while leaving the signal detectable. Measuring the rewritten text alone cannot distinguish these outcomes or establish whether a defense remains effective against the resulting model. Adaptive robustness must therefore be evaluated by training a surrogate on rewritten responses and measuring both its detector score and performance.

To address these limitations, we introduce a lifecycle-oriented benchmark that connects attack, defense, and adaptive-attack evaluation under a shared text-only threat model. To address the first limitation, we implement six representative attacks under common victim–surrogate configurations, query data, and held-out evaluation while retaining their distinct querying and optimization procedures, and separately vary the query budget across different acquisition and training procedures. To address the second, we organize ten defenses by their intervention points and security objectives and pair each security metric with surrogate capability and fidelity. To address the third, we insert DIPPER paraphrasing and back-translation before surrogate training and jointly assess the provenance evidence, capability, and fidelity of the resulting surrogates (Krishna et al., 2023; Liang et al., 2025a). Together, these comparisons isolate extraction methods from model and data choices, test defenses against their stated goals, and determine whether detector signals survive adaptive rewriting and surrogate training.

Our contributions are summarized as follows:

- **A Lifecycle Taxonomy:** We organize extraction attacks, defenses, and adaptive attacks according to their roles and intervention points, providing a common framework for understanding and comparing heterogeneous methods.
- **A Novel Benchmark Protocol:** We design a lifecycle-aligned protocol that matches models, query data, budgets, and evaluation conditions while preserving method-specific procedures, and pairs each defense's objective-specific security evidence with the corresponding surrogate utility within each controlled comparison.
- **A Comprehensive Cross-Attack Evaluation:** Across six attacks, ten defenses, two adaptive attacks, and multiple query budgets, we reveal metric-dependent attack rankings, limited defense transferability, and the utility costs of weakening provenance evidence.

2 Black-Box Extraction Lifecycle and Methods

2.1 Black-Box Model Extraction Lifecycle

We consider a victim model f_{V}, also referred to as the teacher, that is accessible through a text-only API, and a surrogate model f_{S}, also referred to as the student, controlled by the attacker. The defender controls the victim and its released responses, whereas the attacker controls how returned responses are prepared for training and how the surrogate is optimized. We organize this interaction

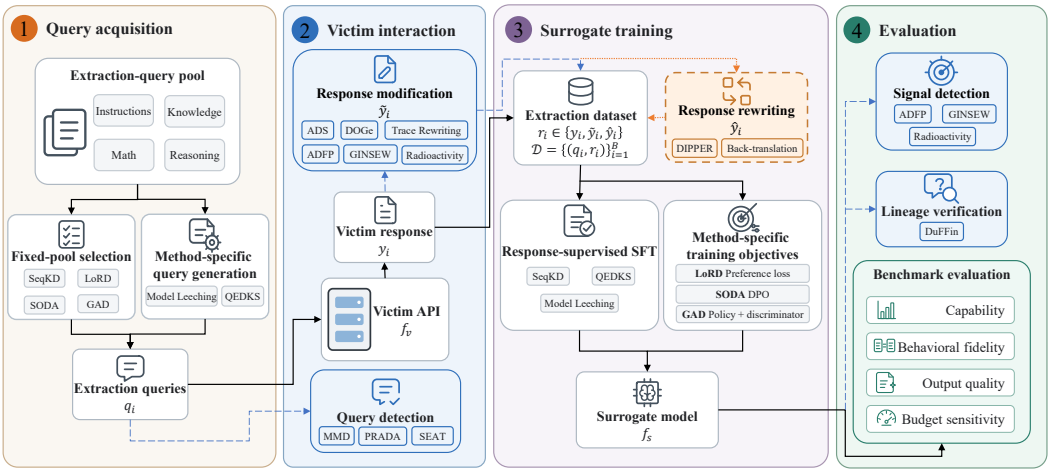


Figure 1: Benchmark lifecycle for black-box LLM extraction. Black arrows show extraction flow, blue dashed arrows show defense operations, and orange dotted arrows show adaptive rewriting.

into the four lifecycle stages shown in Figure 1 (Oliynyk et al., 2025a; Zhao et al., 2025). In *query acquisition*, the attacker selects or constructs the extraction queries q_i. During *victim interaction*, each query is submitted to the victim to obtain a response $y_i = f_V(q_i)$. In *surrogate training*, the attacker forms an extraction dataset from the queries and the responses used for training and optimizes the surrogate. Finally, *evaluation* measures the resulting surrogate and any detector outputs used to assess a defense.

The response passed to surrogate training depends on the experimental condition. In the attack-only setting, the victim response y_i is used directly. A response-modification defense may instead transform y_i into a protected response $\tilde{y}_i$. An adaptive attacker may further paraphrase or back-translate $\tilde{y}_i$ as $\hat{y}_i$ before surrogate training (Liang et al., 2025a). Accordingly, for a query budget B, the training response r_i, extraction dataset $\mathcal{D}$, and resulting surrogate are defined as

$$r_i \in \{y_i, \tilde{y}_i, \hat{y}_i\}, \quad \mathcal{D} = \{(q_i, r_i)\}_{i=1}^{B}, \quad f_S = \mathrm{Train}(\mathcal{D}). \tag{1}$$

Figure 1 provides the organizing framework for the attack and defense models in the following subsections. Extraction attacks may differ in query acquisition and surrogate training, while defenses inspect queries, modify victim responses, or examine trained models. Adaptive attackers instead rewrite protected responses before surrogate training. The following subsections describe each method by the lifecycle components it changes.

2.2 Extraction and Adaptive Attack Models

The attacker may use public information about the victim's application domain or model family but cannot access its parameters, gradients, hidden states, token probabilities, private training data, or implementation details. When a defense is present, the attacker does not know its secret keys, watermark rules, or detector thresholds and does not receive the corresponding undefended responses. The six extraction attacks share the same victim-interaction stage under text-only access: they submit acquired queries to the victim and use the released responses to construct training supervision. Their defining differences occur during query acquisition and surrogate training, while adaptive attacks additionally transform protected responses after victim interaction and before surrogate training.

Query Acquisition (Stage 1). Within the shared text-only access setting, the attacks follow either *fixed-pool selection* or *method-specific query generation*. SeqKD, LoRD, SODA, and GAD select queries from the shared extraction-query pool (Kim & Rush, 2016; Liang et al., 2025b; Chen et al., 2026; Ye et al., 2026). Model Leeching applies prompt templates to queries sampled from that pool, whereas QEDKS begins with seed queries and expands them through templates and follow-up questions (Birch et al., 2023; Li et al., 2026). The first group therefore shares the same source and selection process during query acquisition, while the latter methods make query construction part of the attack itself rather than following shared query selection.

Surrogate Training (Stage 3). The attacks likewise follow either *response-supervised SFT* or *method-specific training objectives*. SeqKD and QEDKS train directly on victim responses, while Model Leeching cleans and parses the returned text before applying supervised fine-tuning (Kim & Rush, 2016; Birch et al., 2023; Li et al., 2026). LoRD instead combines victim responses with candidates sampled from the current surrogate for iterative pairwise optimization; SODA constructs teacher–student preference pairs for DPO; and GAD jointly optimizes the surrogate with a response discriminator (Liang et al., 2025b; Chen et al., 2026; Ye et al., 2026). These methods therefore share the same victim-interaction procedure but differ in how the released responses are converted into supervision and optimized during surrogate training.

Adaptive Attacks (Between Stages 2 and 3). We instantiate two adaptive attacks using DIPPER paraphrasing and English–French–English back-translation (Krishna et al., 2023; Liang et al., 2025a). These transformations target response-embedded provenance evidence while seeking to preserve useful supervision for the downstream surrogate. Because rewriting occurs after victim interaction, it changes only the supervision passed to surrogate training; the query sequence and victim generation remain fixed.

2.3 Defense Models

Depending on the defense, the defender can monitor the query sequence, modify generated responses, or compare a trained suspect model with the victim. The defender possesses any private information required by the defense but cannot control the attacker's processing of returned responses or surrogate training. We therefore group defenses by what they inspect or modify and by the quantity used to judge whether they succeed. The observed object is therefore a query, a released response, or a trained model.

Query Detection (Stage 1). Before victim inference, MMD, PRADA, and SEAT inspect the queries produced during query acquisition and classify the observed stream as benign or extraction traffic (Liu et al., 2026; Juuti et al., 2019; Zhang et al., 2021). These defenses observe only submitted queries; they cannot use the returned responses, the attacker's training procedure, or the resulting surrogate model. This restriction distinguishes query detection from response- and model-based defenses at later lifecycle stages.

Response Modification (Stage 2). Anti-distillation methods—ADS, DOGe, and Trace Rewriting—seek to reduce the value of the released responses as extraction supervision (Savani et al., 2025; Li et al., 2025; Ma et al., 2026). Provenance methods—ADFP, GINSEW, and Radioactivity—instead embed signals in the released responses that are intended to be inherited by a surrogate trained on them (Xu et al., 2026; Zhao et al., 2023; Sander et al., 2024). Both families modify victim responses, but anti-distillation aims to reduce surrogate performance whereas provenance aims to make the trained surrogate detectable during post-training evaluation.

Post-Training Verification (Stage 4). Provenance defenses connect two lifecycle points: they embed signals during victim interaction and later apply *signal detection* to the trained surrogate during evaluation. DuFFin instead performs *lineage verification* only after training, using diagnostic probes to determine whether a suspect model was derived from the victim without modifying the responses released earlier in the lifecycle (Yan et al., 2026). Query detection, anti-distillation, provenance, and lineage verification therefore require different measurements rather than a single defense score.

3 Benchmark Evaluation Design

Overview & Research Questions. Following Figure 1, we compare attacks while preserving their distinct query-acquisition and training procedures, evaluate defenses on submitted queries, released responses, or trained models, and apply adaptive rewriting before surrogate training. Each comparison varies the attack, defense, rewriting, or query budget under study while holding the surrounding experimental conditions fixed. We report capability and fidelity together with the relevant defense result, such as query classification or a provenance or lineage detector score. We study four research questions: **RQ1:** How do different model extraction attacks perform under controlled experimental conditions? **RQ2:** How well do defenses achieve their objective-specific security goals, and what utility costs do they impose? **RQ3:** Can adaptive attacks weaken provenance evidence while pre-

serving extraction utility? **RQ4:** How does query budget affect attacks with different acquisition and training procedures under otherwise matched conditions?

3.1 Attack Evaluation Across Lifecycle Stages

We compare attack effectiveness, protected-response training with and without adaptive rewriting, and query-budget sensitivity while retaining each method's query and training procedure. Each comparison trains fresh surrogates under the model configuration specified below.

Attack Comparison (Stages 1 and 3; Evaluation at Stage 4). All six attacks use a common query budget and shared extraction-query pool while retaining their defining acquisition and training procedures. Fixed-pool attacks use matched pool prefixes, whereas Model Leeching and QEDKS construct their method-specific queries within the same budget. Each configuration then trains a fresh surrogate with its method-specific supervision construction, initialization, and optimization. The resulting surrogates are evaluated under the same held-out conditions, with capability, behavioral fidelity, and output quality reported separately.

Adaptive-Attack Evaluation (Between Stages 2 and 3; Evaluation at Stage 4). Both conditions use the same queries and protected responses. The *defense-only* condition trains directly on $\widetilde{y}_i$, whereas the *adaptive* condition paraphrases or back-translates each response as $\widehat{y}_i$ before training; both use the same SeqKD configuration. After training, we compare the surrogate trained on rewritten responses with the one trained directly on protected responses in detector score, capability, and behavioral fidelity under matched evaluation conditions.

Budget Sensitivity (Stage 1; Evaluation at Stage 4). We vary the query budget for SeqKD, SODA, Model Leeching, and QEDKS without changing each method's acquisition or training procedure. SeqKD and SODA use the same fixed-pool transcript at each budget, whereas Model Leeching and QEDKS retain their method-specific acquisition procedures. Each configuration trains a fresh surrogate and is evaluated under the same held-out conditions. We then track changes in capability, behavioral fidelity, and output quality as the number of queries increases.

3.2 Defense Evaluation Across Lifecycle Stages

Each defense family uses the measurement corresponding to its goal: query classification, surrogate degradation, provenance detection, or lineage separation. No overall defense score combines these quantities into a single measure.

Query Detection (Stage 1). Before victim interaction, each detector is calibrated on benign query batches and then applied to held-out benign and attack batches with matched batch sizes. The resulting decisions measure how well the detector separates extraction traffic from ordinary use.

Anti-Distillation (Stage 2; Evaluation at Stage 4). For each attack–defense pair, the same extraction queries produce clean responses, which ADS, DOGe, or Trace Rewriting then transforms into protected responses. Fresh surrogates are trained from the clean and protected responses using the same extraction-attack configuration. After training, anti-distillation is measured by the change in surrogate capability and behavioral fidelity from the matched clean condition to the protected condition. Because the queries and training procedure are fixed, any performance difference is attributable to response modification.

Provenance (Stage 2; Detection at Stage 4). For each provenance defense, we apply its embedding procedure to the clean responses while keeping the queries unchanged. Surrogates trained on clean and protected responses use the same extraction-attack configuration. After training, each defense applies its own detector to determine whether the embedded signal survives, while capability and behavioral fidelity measure the associated utility change. Across attacks, detector scores are reported separately from capability and fidelity because successful attribution does not imply degraded extraction, or vice versa.

Lineage Verification (Stage 4). DuFFin applies the same diagnostic probes to trained extracted surrogates and unrelated control models and measures how well their agreement with the victim separates the two groups. DuFFin therefore requires trained models rather than matched clean and protected response sets from earlier lifecycle stages.

Table 1: Attack evaluation at query budget $B = 1000$. Bold marks the best value in each column; Avg. is the six-task macro average, and BERT F1 is baseline-rescaled BERTScore F1. ARC-C, HellaS., TQA, and WinoG. denote ARC-Challenge, HellaSwag, TruthfulQA MC1, and WinoGrande.

Method	Capability: ACC (%) ↑							Fidelity	Repetition
	ARC-C	HellaS.	MMLU	TQA	WinoG.	GSM8K	**Avg.**	BERT F1 ↑	Rep-4 ↓
SeqKD	51.88	**72.20**	58.64	31.82	67.25	84.84	61.10	0.337	0.124
Model Leeching	**55.97**	72.18	**63.35**	31.95	65.35	77.48	61.05	0.073	**0.055**
LoRD	49.23	66.37	46.40	33.41	64.56	84.46	57.41	0.289	0.188
SODA	51.96	72.09	58.47	32.19	66.93	85.06	**61.12**	0.338	0.122
GAD	48.38	70.80	35.10	26.19	**68.27**	**86.05**	55.80	0.148	0.563
QEDKS	50.26	66.82	58.52	**33.90**	61.01	84.08	59.10	**0.349**	0.119

3.3 Shared Controls and Benchmark Configuration

Matched Experimental Conditions. The variables held fixed depend on the comparison rather than on one protocol shared by all methods. Queries are matched whenever the method permits. For a given comparison, we hold the victim and generation settings, query identities where applicable, surrogate configuration, and held-out evaluation conditions fixed unless that component is under study. Method-specific query acquisition and training procedures are preserved. Further generation, training, and data-processing details are provided in Appendices B and C.

Query Data and Budgets. We construct the extraction-query pool from LMSYS-Chat-1M (Zheng et al., 2023), NuminaMath-CoT (AI-MO Team, 2024), TriviaQA (Joshi et al., 2017), MMLU (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), WinoGrande (Sakaguchi et al., 2020), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), SciQ (Welbl et al., 2017), Social-IQA (Sap et al., 2019), ARC-Challenge (Clark et al., 2018), and OpenBookQA (Mihaylov et al., 2018). These sources combine open-ended user instructions with mathematical, factual, commonsense, and scientific reasoning tasks, allowing us to evaluate extraction across diverse capabilities. We use only the input from each source for extraction and evaluate the resulting surrogates on held-out prompts disjoint from the extraction data. Unless otherwise stated, comparisons use a query budget of $B = 1000$; the budget-sensitivity analysis uses $B \in \{100, 1000, 10000\}$.

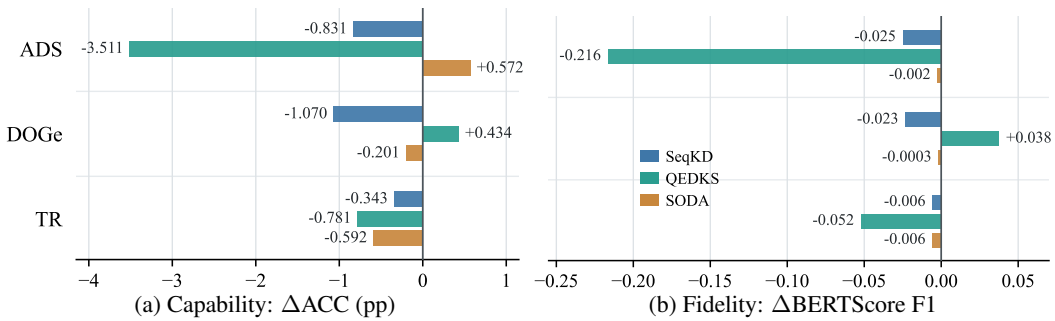
Model Configurations. We examine two open-weight large-victim/small-surrogate configurations that represent extraction from a hosted model into a trainable local model. Attack and adaptive-attack experiments use Llama-3.3-70B-Instruct as the victim and Llama-3.1-8B-Instruct as the initial surrogate (Grattafiori et al., 2024). Defense experiments use Qwen2.5-72B-Instruct as the victim and Qwen2.5-7B as the initial surrogate (Qwen et al., 2024).

Evaluation Data and Measures. Capability is evaluated on held-out task examples, while behavioral fidelity and output quality use a separate set of open-ended prompts; all evaluation data are disjoint from the extraction queries. We measure capability using macro-average accuracy, behavioral fidelity using baseline-rescaled BERTScore F1 (Zhang et al., 2020), and output quality using Rep-4. These dimensions are reported separately rather than combined into an overall ranking. Defense evaluation additionally uses attack–benign discrimination for query detection, clean–protected performance changes for anti-distillation, native detector scores for provenance, and extracted–unrelated model separation for lineage verification. Adaptive evaluation reports provenance-detector scores together with the capability and fidelity of the surrogate trained on rewritten responses. Formal definitions and calibration procedures are provided in Appendix D.

4 Experiments

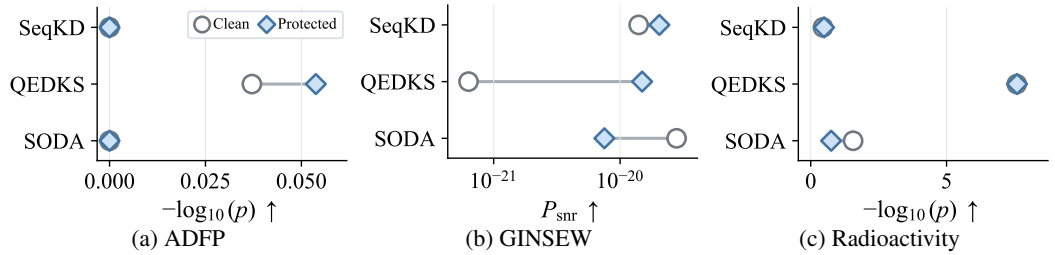
4.1 Extraction Attack Effectiveness

To answer RQ1, we compare the six attacks at $B = 1000$ under the matched conditions in Section 3.1 and report task capability, behavioral fidelity, and output repetition. From Table 1, we make the following observations: (1) From the perspective of capability, SeqKD, Model Leeching, and SODA achieve similar average accuracy, but their advantages vary across tasks. For example, Model Leeching performs best on ARC-Challenge and MMLU but trails the other two methods on



(a) Capability: ΔACC (pp) (b) Fidelity: ΔBERTScore F1

Figure 2: Anti-distillation effects at $B = 1000$. We report defended-minus-clean ACC (percentage points) and BERTScore F1; negative values indicate suppression, whereas positive values indicate improvement. Colors distinguish extraction attacks, and TR denotes Trace Rewriting. Near-zero bars use a minimum visible width; labels give exact values.



(a) ADFP (b) GINSEW (c) Radioactivity

Figure 3: Provenance-detector scores for surrogates trained on matched clean and protected responses at $B = 1000$. We report ADFP and Radioactivity as $-\log_{10}(p)$ and GINSEW as P_{snr}; larger values indicate a stronger provenance signal.

GSM8K, while GAD performs best on WinoGrande and GSM8K despite its lower average accuracy. These task-dependent rankings show that aggregate accuracy can conceal which capabilities an attack transfers. Moreover, the more elaborate student-response objectives used by LoRD and GAD do not consistently improve capability over direct teacher-response supervision under the fixed budget, indicating that additional optimization machinery alone does not guarantee broader extraction. (2) From the perspective of fidelity, the ranking differs from that of capability. QEDKS attains the highest BERTScore F1 without attaining the highest accuracy, whereas Model Leeching retains near-best capability but has the lowest fidelity. This contrast is consistent with their supervision construction: QEDKS trains on complete teacher responses collected through template and follow-up queries, while Model Leeching retains parsed answer text and can therefore preserve task solutions without reproducing the victim's full response form. Thus, matching the victim's outputs and solving the same tasks represent distinct extraction objectives. (3) Output repetition provides a third, independent view. Model Leeching has the lowest Rep-4 but also the lowest fidelity, showing that low repetition measures output diversity rather than faithful imitation; conversely, GAD's high Rep-4 reveals degraded generation quality despite strong results on selected tasks. In conclusion, no single attack ranks best in capability, fidelity, and output repetition, so all three outcomes are needed for comparison across extraction methods.

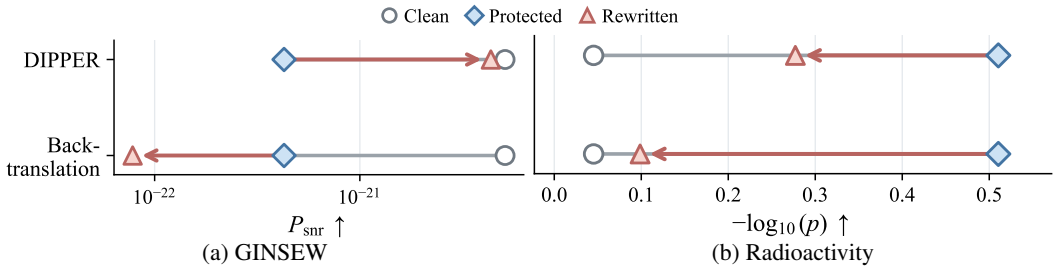
4.2 Defense Evaluation

To answer RQ2, we evaluate anti-distillation through capability and fidelity changes, provenance through detector scores, query detection through attack–benign discrimination, and lineage verification through extracted–unrelated model separation.

(1) Anti-distillation is attack-dependent. As shown in Figure 2 and detailed in Appendix E.2, ADS suppresses QEDKS but has little effect on SeqKD or SODA, while DOGe is similarly mixed and Trace Rewriting produces more consistent but modest degradation. These differences are consistent with how the attacks consume protected responses: SeqKD directly imitates them, QEDKS combines teacher supervision with expanded query acquisition, and SODA converts teacher–student

Table 2: SeqKD surrogate performance under adaptive attacks at $B = 1000$. ACC is reported in percent; the matched clean baseline has ACC 58.29 and BERTScore F1 0.374.

Training responses	**ADFP**		**GINSEW**		**Radioactivity**	
	ACC	BERT F1	ACC	BERT F1	ACC	BERT F1
Defense-only	60.29	0.325	56.31	0.376	58.05	0.377
DIPPER	56.67	0.140	56.17	0.167	56.05	0.173
Back-translation	48.86	0.035	51.11	0.047	48.62	0.088



(a) GINSEW (b) Radioactivity

Figure 4: Provenance-detector scores for SeqKD surrogates under response rewriting at $B = 1000$. Rows share clean and protected baselines, and arrows trace scores after rewriting and retraining; larger values indicate stronger evidence. We omit ADFP because all scores are zero.

outputs into preference pairs. A modification that disrupts one form of supervision can therefore be attenuated or repurposed by another training objective.

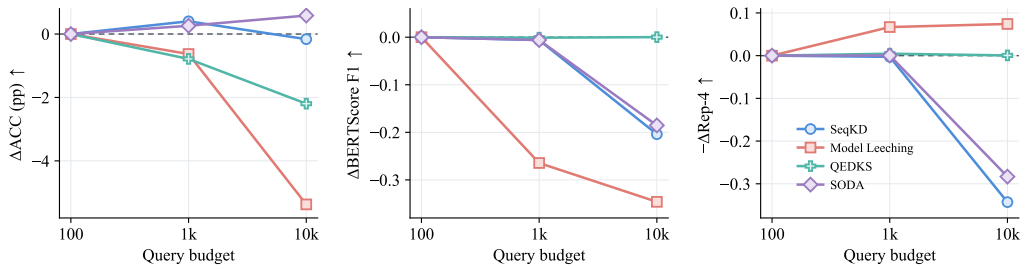
(2) Provenance signals do not consistently survive surrogate training. Figure 3 and Appendix E.3 show little change in capability or fidelity, but the detector scores vary across extraction attacks. GINSEW provides its clearest separation under QEDKS, whereas ADFP often returns no signal and Radioactivity can move in the wrong direction. Thus, embedding provenance signals in released responses does not consistently produce detectable evidence in surrogates subsequently trained on those responses across the evaluated extraction procedures.

(3) Query detection depends on the query-generation strategy. Appendix E.4 shows that MMD and SEAT recognize QEDKS traffic, while the fixed SeqKD queries remain largely indistinguishable from benign traffic. Because these defenses observe queries rather than the attacker's intent or training procedure, QEDKS's generated templates and follow-ups create a more distinctive distribution than queries sampled from a fixed pool. This contrast suggests that the detectors capture acquisition-pattern shifts rather than model extraction uniformly, leaving attacks based on natural-looking queries substantially harder to identify.

(4) Lineage verification remains unreliable. Appendix E.5 shows that DuFFin does not consistently rank extracted students above unrelated controls, indicating that the current probe set provides insufficient derivation evidence. Its success on QEDKS does not extend to SeqKD or SODA, limiting its reliability across extraction procedures. The score therefore depends on whether extraction transfers victim behaviors covered by the probe set. Broader probe coverage is therefore needed to compare lineage evidence reliably across attacks. Such probes should target behaviors preserved across different extraction objectives.

4.3 Adaptive Robustness

To answer RQ3, we compare each SeqKD surrogate trained on rewritten responses with its matched surrogate trained directly on protected responses. Table 2 reports capability and fidelity, Figure 4 reports detector scores, and Appendix E.6 provides exact values. From these results, we make the following observations: (1) Rewriting indicates evasion only if protected-response training first raises the detector score above the clean baseline and rewriting then lowers it. Without clean–protected separation, a lower score cannot demonstrate that an inherited signal was removed. ADFP produces no inherited signal, while GINSEW lacks positive clean–protected separation; moreover, DIPPER



(a) Capability (b) Fidelity (c) Output quality

Figure 5: Query-budget sensitivity of four extraction attacks with distinct query-acquisition and training procedures. We report changes relative to $B = 100$ using ΔACC and ΔBERTScore F1 and negate ΔRep-4 so that higher values indicate better outcomes in every panel.

increases the GINSEW score whereas back-translation decreases it, precluding a uniform evasion effect. Both attacks weaken Radioactivity, with back-translation moving it closer to the clean baseline. (2) This reduction is not cost-free: both methods reduce fidelity across all three defenses, and back-translation causes the larger capability loss, consistent with a more disruptive transformation of provenance cues and imitation-relevant supervision. Lowering a detector score benefits the attacker only if the rewritten responses still train a capable and faithful surrogate. In conclusion, adaptive rewriting produces defense-dependent and partial detector-score reductions while also reducing fidelity, rather than bypassing provenance defenses while preserving extraction quality.

4.4 Query-Budget Sensitivity

To answer RQ4, we vary the query budget for four attacks while preserving each method's acquisition and training procedure. Figure 5 reports changes in capability, fidelity, and output quality relative to $B = 100$.

Figure 5 yields three observations. (1) Capability does not improve monotonically: SeqKD remains stable and SODA improves slightly, whereas Model Leeching and QEDKS decline, most sharply for Model Leeching from $B = 1000$ to $B = 10000$. Thus, additional responses do not ensure transferable supervision. (2) Fidelity and output quality can decline despite retained capability. At $B = 10000$, both fall sharply for SeqKD and SODA. Model Leeching loses fidelity as repetition decreases, while QEDKS preserves both despite lower capability; larger budgets can therefore improve one objective while degrading another. (3) These differences reflect acquisition and training design. SeqKD and SODA share fixed-pool transcripts, but SODA's preference optimization adds only a modest capability gain and does not prevent large-budget fidelity and output-quality losses. Parsed-answer supervision separates Model Leeching from the victim's response form, whereas QEDKS preserves response similarity without capability gains. Overall, budget effects depend on query acquisition and supervision, not query count alone.

5 Conclusion and Future Work

We present a lifecycle-oriented benchmark for black-box LLM extraction that compares six attacks, ten defenses spanning four security objectives, and two adaptive attacks under a common text-only threat model. Each comparison matches the victim–surrogate configuration, query data, and held-out evaluation while preserving method-specific acquisition and training procedures. No single metric characterizes extraction: attack rankings vary across capability, fidelity, and repetition, and larger budgets do not uniformly improve performance. Defense effectiveness likewise varies across attacks. Paraphrasing and back-translation lower some provenance-detector scores but also reduce surrogate capability or fidelity. These controlled comparisons may change with the victim–surrogate pair, domain, or deployment setting. Overall, the benchmark connects lifecycle choices to surrogate and defense outcomes and distinguishes response-level changes from effects that persist after training. Future work will broaden model and language coverage, evaluate stronger adaptive strategies, and develop reliable provenance and lineage-verification protocols.

AI USE STATEMENT

We used large language models to retrieve and discover relevant literature and to aid in polishing author-written text. The authors reviewed the retrieved sources, verified the resulting citations against the original publications, and checked all AI-assisted edits for accuracy and consistency with the reported methods and results. We did not use AI assistance to generate synthetic datasets, prove mathematical claims, or conduct the reported experiments.

ETHICS STATEMENT

Model extraction research has dual-use implications: a systematic evaluation can support stronger defenses, but attack implementations may also facilitate unauthorized imitation of deployed models. Our study is intended to enable controlled, reproducible analysis of this risk and to clarify the conditions under which existing defenses succeed or fail. We conduct experiments with public datasets and open-weight models in a controlled research environment; we do not target proprietary production services or collect private user data. The benchmark does not grant authorization to extract or imitate a model, and its use should comply with model licenses, data licenses, service terms, and applicable law.

REPRODUCIBILITY STATEMENT

We specify the lifecycle-aligned attack and defense evaluation flows and their shared experimental controls in Sections 3.1–3.3. Appendices B–D document the attack and defense implementations, generation and training settings, model configurations, data construction and processing, evaluation metrics, and calibration procedures. Appendix E provides detailed numerical results beyond those reported in the main text. Code, configuration files, and processed evaluation artifacts are available in the anonymous repository.

REFERENCES

AI-MO Team. NuminaMath-CoT. Hugging Face dataset, 2024. URL https://huggingface.co/datasets/AI-MO/NuminaMath-CoT.

Youssef Allouah, Mahdi Haghifam, Sanmi Koyejo, and Reza Shokri. The distillation game: Adaptive attacks and efficient defenses. *arXiv preprint arXiv:2605.22737*, 2026.

Hyeseon An, Shinwoo Park, Suyeon Woo, and Yo-Sub Han. DITTO: A spoofing attack framework on watermarked LLMs via knowledge distillation. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4922–4936, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.229. URL https://aclanthology.org/2026.eacl-long.229/.

Lewis Birch, William Hackett, Stefan Trawicki, Neeraj Suri, and Peter Garraghan. Model leeching: An extraction attack targeting LLMs. *arXiv preprint arXiv:2309.10544*, 2023.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239. URL https://ojs.aaai.org/index.php/AAAI/article/view/6239.

Xiwen Chen, Jingjing Wang, Wenhui Zhu, Peijie Qiu, Xuanzhao Dong, Yueyue Deng, Hejian Sang, Zhipeng Wang, Alborz Geramifard, and Feng Luo. SODA: Semi on-policy black-box distillation for large language models. *arXiv preprint arXiv:2604.03873*, 2026.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

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, 2021. URL https://arxiv.org/abs/2110.14168.

Aaron Grattafiori et al. The Llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3Gm.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Wei Huang, Yinggui Wang, and Cen Chen. Privacy evaluation benchmarks for NLP models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 2615–2636, 2024. doi: 10.18653/v1/2024.findings-emnlp.147.

Bo Jiang. DistillGuard: Evaluating defenses against LLM knowledge distillation, 2026. URL https://arxiv.org/abs/2603.07835.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017. doi: 10.18653/v1/P17-1147. URL https://aclanthology.org/P17-1147/.

Mika Juuti, Sebastian Szyller, Samuel Marchal, and N. Asokan. PRADA: Protecting against DNN model stealing attacks. In *2019 IEEE European Symposium on Security and Privacy*, pp. 512–527, 2019.

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, 2016.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 17061–17084, 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.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. In *Advances in Neural Information Processing Systems*, 2023.

Meghdad Kurmanji, Ashley Hoi-Ting Au, William F. Shen, and Nicholas D. Lane. Benchmarking unauthorized distillation should be attack–defense co-evaluation under API constraints, June 2026. URL https://doi.org/10.13140/RG.2.2.24233.35680. Position paper.

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.

Zhengao Li, Xiaopeng Yuan, Bolin Shen, Kien Le, Haohan Wang, Xugui Zhou, Shangqian Gao, and Yushun Dong. Query-efficient domain knowledge stealing against large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Jiacheng Liang, Zian Wang, Spencer Hong, Shouling Ji, and Ting Wang. Watermark under fire: A robustness evaluation of LLM watermarking. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 21050–21074, 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*, 2025b.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3214–3252, 2022. doi: 10.18653/v1/2022.acl-long.229. URL https://aclanthology.org/2022.acl-long.229/.

Jian Liu, Rui Zhang, Sebastian Szyller, Kui Ren, and N. Asokan. False claims against model ownership resolution. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 6885–6902, Philadelphia, PA, August 2024. USENIX Association. ISBN 978-1-939133-44-1. URL https://www.usenix.org/conference/usenixsecurity24/presentation/liu-jian.

Shuze Liu, Qianwen Guo, and Yushun Dong. An embarrassingly simple detector for model extraction attacks in large language model API traffic. *arXiv preprint arXiv:2606.05725*, 2026.

Yugeng Liu, Rui Wen, Xinlei He, Ahmed Salem, Zhikun Zhang, Michael Backes, Emiliano De Cristofaro, Mario Fritz, and Yang Zhang. ML-Doctor: Holistic risk assessment of inference attacks against machine learning models. In *31st USENIX Security Symposium*, pp. 4525–4542, 2022.

Peizhuo Lv, Ruihua Zhou, Yunpeng Li, Ruigang Liang, Xingshuo Han, XiaoFeng Wang, Wei Dong, and Yuling Liu. ReasMark: A robust watermark for attributing LLM reasoning under knowledge distillation attacks. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 47221–47241, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.2185. URL https://aclanthology.org/2026.acl-long.2185/.

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*, 2026.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, 2018. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Daryna Oliynyk, Rudolf Mayer, Kathrin Grosse, and Andreas Rauber. I stolenly swear that i am up to (no) good: Design and evaluation of model stealing attacks. *arXiv preprint arXiv:2508.21654*, 2025a.

Daryna Oliynyk, Rudolf Mayer, and Andreas Rauber. Attackers can do better: Over- and understated factors of model stealing attacks, 2025b. URL https://arxiv.org/abs/2503.06188.

OpenAI et al. GPT-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

Leyi Pan, Aiwei Liu, Shiyu Huang, Yijian Lu, Xuming Hu, Lijie Wen, Irwin King, and Philip S. Yu. Can LLM watermarks robustly prevent unauthorized knowledge distillation? In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13228–13251, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.648. URL https://aclanthology.org/2025.acl-long.648/.

Zhisheng Qi, Utkarsh Sahu, Li Ma, Haoyu Han, Ryan Rossi, Franck Dernoncourt, Mahantesh Halappanavar, Nesreen Ahmed, Yushun Dong, Yue Zhao, Yu Zhang, and Yu Wang. Benchmarking knowledge-extraction attack and defense on retrieval-augmented generation (RAG). In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Volume 2*, pp. 9718–9729. ACM, 2026. doi: 10.1145/3770855.3817524. URL https://doi.org/10.1145/3770855.3817524.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2024. URL https://arxiv.org/abs/2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a85b405ed65c6477a4fe8302b5e06ce7-Abstract-Conference.html.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 3982–3992, 2019. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 8732–8740, 2020. doi: 10.1609/aaai.v34i05.6399. URL https://ojs.aaai.org/index.php/AAAI/article/view/6399.

Tom Sander, Pierre Fernandez, Alain Durmus, Matthijs Douze, and Teddy Furon. Watermarking makes language models radioactive. In *Advances in Neural Information Processing Systems*, 2024.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 4463–4473, 2019. doi: 10.18653/v1/D19-1454. URL https://aclanthology.org/D19-1454/.

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.

Seamless Communication, Loïc Barrault, Yu-An Chung, et al. Seamless: Multilingual expressive and streaming speech translation. *arXiv preprint arXiv:2312.05187*, 2023. doi: 10.48550/arXiv.2312.05187. URL https://arxiv.org/abs/2312.05187.

Xinpeng Ti, Wentao Ye, Zhifang Zhang, Junbo Zhao, Chang Yao, Lei Feng, and Haobo Wang. Towards reverse engineering of language models: A survey. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 7483–7502, 2025.

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, Austin, TX, August 2016. USENIX Association. ISBN 978-1-931971-32-4. URL https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/tramer.

Eric Wallace, Mitchell Stern, and Dawn Song. Imitation attacks and defenses for black-box machine translation systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 5531–5546, 2020. doi: 10.18653/v1/2020.emnlp-main.446.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5776–5788, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. *arXiv preprint arXiv:2406.01574*, 2024. doi: 10.48550/arXiv.2406.01574. URL https://arxiv.org/abs/2406.01574.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 94–106, 2017. doi: 10.18653/v1/W17-4413. URL https://aclanthology.org/W17-4413/.

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

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*, 2026.

Yuliang Yan, Haochun Tang, Shuo Yan, and Enyan Dai. DuFFin: A dual-level fingerprinting framework for LLMs IP protection. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 5168–5184. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-eacl.273. URL https://aclanthology.org/2026.findings-eacl.273/.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-box on-policy distillation of large language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=wTSkATPm8A.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Boyang Zhang, Zheng Li, Ziqing Yang, Xinlei He, Michael Backes, Mario Fritz, and Yang Zhang. SecurityNet: Assessing machine learning vulnerabilities on public models. In *33rd USENIX Security Symposium*, 2024.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

Zhanyuan Zhang, Yizheng Chen, and David Wagner. SEAT: Similarity encoder by adversarial training for detecting model extraction attack queries. In *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security*, 2021. doi: 10.1145/3474369.3486863.

Kaixiang Zhao, Lincan Li, Kaize Ding, Neil Zhenqiang Gong, Yue Zhao, and Yushun Dong. A survey on model extraction attacks and defenses for large language models. *arXiv preprint arXiv:2506.22521*, 2025.

Xuandong Zhao, Yu-Xiang Wang, and Lei Li. Protecting language generation models via invisible watermarking. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 42187–42199. PMLR, 2023. URL https://proceedings.mlr.press/v202/zhao23i.html.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. LMSYS-Chat-1M: A large-scale real-world LLM conversation dataset, 2023. URL https://arxiv.org/abs/2309.11998.

A Related Work

LLM Model Extraction. Model extraction builds on knowledge distillation, where a student learns to reproduce a teacher's behavior from its predictions (Kim & Rush, 2016). Early studies demonstrated imitation of NLP classification and machine translation APIs using query access alone (Krishna et al., 2020; Wallace et al., 2020). Recent LLM attacks range from template-based response imitation in Model Leeching (Birch et al., 2023), through student-aware or semi/on-policy optimization in LoRD, SODA, and GAD (Liang et al., 2025b; Chen et al., 2026; Ye et al., 2026), to query-efficient domain extraction in QEDKS (Li et al., 2026). Although surveys have systematized their threat models and techniques (Zhao et al., 2025; Ti et al., 2025), inconsistent query pools, budgets, model configurations, and tasks continue to hinder direct comparison (Oliynyk et al., 2025a;b).

Defenses and Adaptive Attacks. Defenses pursue several distinct goals: ADS, DOGe, and Trace Rewriting reduce the training value of released responses (Savani et al., 2025; Li et al., 2025; Ma et al., 2026); GINSEW, Radioactivity, and ADFP embed provenance signals intended to survive distillation (Zhao et al., 2023; Sander et al., 2024; Xu et al., 2026); and fingerprinting methods verify model ownership through implanted or behavioral evidence (Xu et al., 2024; Yan et al., 2026). These approaches build on broader output-watermarking techniques that introduce statistically detectable signals during generation (Kirchenbauer et al., 2023). PRADA, SEAT, and MMD Detector instead seek to recognize extraction from patterns in the attacker's query stream (Juuti et al., 2019; Zhang et al., 2021; Liu et al., 2026). Adaptive attackers can rewrite protected outputs before training: DIPPER studies paraphrasing-based evasion (Krishna et al., 2023), while WATERPARK evaluates watermark robustness against paraphrasing, translation, and other transformations (Liang et al., 2025a). Pan et al. further evaluate pre- and post-distillation watermark removal by jointly measuring inherited evidence and student knowledge (Pan et al., 2025). Recent provenance work also highlights reasoning-specific attribution and the risk of falsely or adversarially reproduced ownership evidence (Lv et al., 2026; An et al., 2026; Liu et al., 2024). Training-time adaptations have also been explored against anti-distillation defenses (Allouah et al., 2026), but these defense and evasion families are generally evaluated in isolation rather than across an end-to-end extraction lifecycle.

Model Extraction Benchmarks. ML-Doctor and an NLP privacy benchmark evaluate model extraction within broader suites of inference or privacy attacks (Liu et al., 2022; Huang et al., 2024), while SecurityNet and subsequent evaluation methodology primarily study public image classifiers and substitute-model stealing (Zhang et al., 2024; Oliynyk et al., 2025a). For generative systems, WATERPARK evaluates transformed-text watermark detection (Liang et al., 2025a), and Qi et al. (2026) benchmark extraction of protected RAG knowledge. DistillGuard standardizes the evaluation of output-level LLM distillation defenses under a fixed teacher–student pipeline, but does not compare diverse extraction strategies, provenance and detection objectives, or adaptive attacks (Jiang, 2026). A recent position paper proposes separating active acquisition, fixed-data distillation, detection, and intervention under API constraints, but does not instantiate the proposed benchmark empirically (Kurmanji et al., 2026). Unlike these complementary but partial settings, our benchmark studies functional extraction from generative LLM APIs and connects diverse attacks, heterogeneous defenses, response transformations used by adaptive attackers, student training, and post-training evaluation in a unified empirical lifecycle.

B Implementation and Reproducibility Details

B.1 Shared Generation and Training Settings

Unless an attack requires online interaction, we generate one victim transcript per query budget and reuse it across attacks so that differences do not arise from stochastic victim outputs. Victim responses use greedy decoding with temperature 0, top-p 1.0, and at most 1536 new tokens. We use seed 20260701 for query selection and extraction and seed 42 for held-out evaluation. We preserve each model's chat template, mask prompt tokens from the language-modeling loss, append an end-of-sequence token, and truncate training sequences to 3584 tokens.

We train parameter-efficient surrogates in bfloat16 with gradient checkpointing. Unless stated otherwise, all trainable surrogate adapters use LoRA (Hu et al., 2022) with rank 16, scaling factor 32,

Table 3: Extraction attacks compared across query selection, supervision construction, and surrogate optimization. SFT denotes supervised fine-tuning.

Attack	Query selection	Training supervision	Surrogate optimization
Teacher-response SFT			
SeqKD	Fixed pool	Teacher responses	Sequence-level SFT
Model Leeching	Templated pool queries	Cleaned and parsed teacher responses	SFT
QEDKS	Seeds, templates, and follow-ups	Teacher responses	SFT
Student-response-based optimization			
LoRD	Fixed pool	Teacher responses and student candidates	Iterative pairwise optimization
SODA	Fixed pool	Teacher–student preference pairs	Preference optimization
GAD	Fixed pool	Teacher–student discriminator data	Discriminator-guided optimization

and dropout 0.05. We do not tune this adapter configuration separately for individual methods or budgets.

B.2 Attack Implementations

We summarize how each attack constructs queries and supervision and optimizes the surrogate in Table 3.

Query selection. SeqKD, LoRD, SODA, and GAD draw queries from the same fixed pool, while Model Leeching applies its prompt templates to queries sampled from that pool (Kim & Rush, 2016; Birch et al., 2023; Liang et al., 2025b; Chen et al., 2026; Ye et al., 2026). QEDKS instead begins with seed queries and uses template-based and follow-up generation to acquire additional domain knowledge within the query budget (Li et al., 2026).

Training-data construction. SeqKD and QEDKS retain query–teacher-response pairs directly, whereas Model Leeching cleans and parses the teacher output before adding the answer text to its training set (Kim & Rush, 2016; Li et al., 2026; Birch et al., 2023). LoRD augments teacher responses with candidates sampled from the current student, SODA converts teacher and student responses into preference pairs, and GAD uses both sources to construct discriminator supervision (Liang et al., 2025b; Chen et al., 2026; Ye et al., 2026).

Surrogate optimization. SeqKD, Model Leeching, and QEDKS optimize the surrogate through supervised fine-tuning on teacher responses (Kim & Rush, 2016; Birch et al., 2023; Li et al., 2026). LoRD performs iterative pairwise optimization using teacher responses and student candidates, SODA applies preference optimization, and GAD alternates updates to the student and a response discriminator (Liang et al., 2025b; Chen et al., 2026; Ye et al., 2026). This distinction separates attacks that primarily vary query acquisition and response processing from attacks that additionally change the surrogate-training objective.

We report the final optimization settings recorded in the released run manifests in Table 4. SeqKD, LoRD, SODA, and GAD use the same fixed victim transcript at each budget. QEDKS and Model Leeching query the victim online because query construction is part of the attack: QEDKS expands three seed queries into 498 template queries and 499 follow-up queries at $B = 1000$, with at most four follow-ups per answer, whereas Model Leeching issues templated prompts and trains on the parsed answer text only.

For LoRD, student candidates are sampled with temperature 1.0, top-p 0.95, and a maximum of 1536 new tokens; we do not initialize LoRD from SeqKD. SODA forms preference pairs from the victim response and a response sampled from the matched SeqKD surrogate, and then applies one

Table 4: Final optimization settings for the extraction attacks. Batch is the per-device batch size and Accum. is the number of gradient-accumulation steps; dashes denote settings not used by LoRD's method-specific optimizer.

Attack	LR	Epochs	Batch	Accum.	Method-specific settings
SeqKD	2×10^{-5}	1	1	32	Linear schedule; sequence-level SFT.
Model Leeching	2×10^{-4}	2	1	16	Answer-only SFT on parsed online responses.
QEDKS	2×10^{-4}	2	1	16	Seed/template/follow-up mix; perplexity scheduling disabled.
LoRD	3×10^{-5}	1 period	–	–	Two student candidates; $(\tau_1, \tau_2, \Delta\tau) = (0.8, 0.1, 0.05)$, $\lambda = 0.5$, clipping 0.2.
SODA	5×10^{-6}	1	1	32	DPO with $\beta = 0.1$; initialized from the matched SeqKD adapter.
GAD	1×10^{-6}	2	1	1	Discriminator LR 10^{-6}; group size 8; 10 discriminator warm-up steps; KL coefficient 10^{-3}.

epoch of DPO (Rafailov et al., 2023) with maximum prompt and sequence lengths of 1024 and 3584. GAD first performs one SeqKD warm-up epoch at learning rate 5×10^{-6} and then jointly updates generator and discriminator adapters; generator sampling uses temperature 0.8 and top-p 1.0. We retain the final completed checkpoint for every attack and do not select checkpoints using held-out evaluation results.

B.3 Defense Implementations and Calibration

Query detection. MMD, PRADA, and SEAT operate before victim inference and classify observed query streams as benign or extraction traffic (Liu et al., 2026; Juuti et al., 2019; Zhang et al., 2021). They receive only the queries available to the model provider and do not inspect the attacker's training procedure or surrogate model. We encode queries using the `sentence-transformers/all-MiniLM-L6-v2` sentence-embedding model (Reimers & Gurevych, 2019; Wang et al., 2020).

Anti-distillation. ADS, DOGe, and Trace Rewriting modify the victim's released responses to reduce their usefulness as extraction supervision (Savani et al., 2025; Li et al., 2025; Ma et al., 2026). We train a new surrogate on each protected transcript and compare it with the surrogate trained on matched clean responses.

Provenance and lineage verification. ADFP, GINSEW, and Radioactivity embed signals during response generation and subsequently test whether those signals are inherited by the surrogate (Xu et al., 2026; Zhao et al., 2023; Sander et al., 2024). DuFFin instead performs post-training lineage verification without modifying the victim responses before extraction (Yan et al., 2026).

Defense experiments use the Qwen victim–surrogate pair specified in Appendix C and fix the query budget to $B = 1000$. For response-modification and provenance defenses, each matched clean–protected pair retains its attack-specific queries, supervision, initialization, and optimization; only the victim responses differ. We summarize the final defense settings in Table 5.

For query-stream detection, we construct benign traffic with the same batch size and budget as the attack stream. The detector receives only queries observed by the victim; follow-up queries produced by QEDKS therefore count toward its budget. We split benign batches before calibration, fit detector-specific statistics on the calibration portion, and freeze the resulting threshold before evaluating either attacks or held-out benign batches. MMD estimates its null distribution using 200 samples. For provenance detection, clean and protected students are evaluated on matched held-out prompts, and the clean score is retained as the reference for each attack–defense pair.

Table 5: Final defense and detector settings. Private keys are fixed across embedding and detection, but their numeric values are omitted here.

Defense	Configuration	Evaluation or calibration
ADS	$\lambda = 0.1$, $\epsilon = 0.01$, $\tau = 0.9$; top-p 0.95	Proxy gradients are precomputed on held-out prompts and used during protected decoding.
DOGe	Anti-KD coefficient 3×10^{-5}; KD temperature 2.0	Train the proxy LM head for two epochs with LR 5×10^{-5} and maximum length 2048.
Trace Rewriting	Official optimized rewriting prompt; temperature 0.6; top-p 0.95	Rewrite the complete answer before releasing it to the attacker.
ADFP	$\gamma = 0.5$, context window 2, strength 140	SHA-256 context seeding with one fixed private key; report $-\log_{10}(p)$.
GINSEW	Mark 50% of eligible tokens; strength 2.0; frequency 16; $\epsilon = 0.2$	Use the same watermark key for generation and detection; report watermark SNR.
Radioactivity	Maryland, 4-gram context, $\gamma = 0.25$, $\delta = 2.0$	Hash-based seeding and the v2 scoring rule; report the detector p-value.
MMD / PRADA / SEAT	50 queries per batch; MiniLM-L6-v2 embeddings	Fit on 80% of benign traffic; lock detector-specific 95%-tail rules; test on disjoint benign and attack batches.
DuFFin	MMLU-Pro (Wang et al., 2024) probes from six subject groups	Score the common intersection of 508 valid probes across the victim, base model, and all surrogates.

B.4 Adaptive-Attack Implementations

We instantiate two representative adaptive attacks against provenance defenses, using DIPPER for paraphrasing and English–French–English back-translation for translation-based rewriting (Krishna et al., 2023; Liang et al., 2025a). Both methods require only the protected response text and do not use clean responses, private defense keys, detector thresholds, or detector outputs. We apply adaptive rewriting only to the response text after protected generation and before surrogate training; the queries and $B = 1000$ budget remain unchanged. DIPPER uses `kalpeshk2011/dipper-paraphraser-xxl` with its T5 tokenizer, lexical diversity 40, order diversity 0, three-sentence rewriting windows, no surrounding document context, top-p 0.75, and maximum length 512. Back-translation uses `facebook/seamless-m4t-v2-large` (Seamless Communication et al., 2023) to translate each protected response from English to French and back to English. Both branches preserve the original query–response pairing and train a fresh SeqKD surrogate with exactly the defense-only training configuration. For every provenance defense, we separately rerun the detector on the clean, defense-only, and rewritten branches rather than reusing baseline scores across branches.

B.5 Software, Hardware, and Checkpoint Selection

We run the experiments under Python 3.10 with PyTorch 2.11.0, Transformers 5.14.1, PEFT 0.15.2, TRL 1.9.2, Accelerate 1.14.0, Datasets 5.0.1, and vLLM 0.26.0. Cluster jobs load CUDA 12.4.1 and record the resolved accelerator, package freeze, command, configuration, inputs, and output checksums in a run manifest. We use the final completed adapter checkpoint for evaluation; SODA additionally records and validates its matched SeqKD initialization, and GAD stores the generator and discriminator adapters separately.

C Models, Tasks, and Data Processing

C.1 Victim and Surrogate Models

We use two fixed victim–surrogate configurations. Attack and adaptive-attack experiments use `meta-llama/Llama-3.3-70B-Instruct` as the victim and initialize ev-

Table 6: Composition of the master extraction-query pool.

Dataset	Role	Source split	Processing	Share
LMSYS-Chat-1M (Zheng et al., 2023)	Natural user instructions	default/train	User prompt retained	80.00%
NuminaMath-CoT (AI-MO Team, 2024)	Mathematical reasoning	default/train	Problem retained	4.00%
TriviaQA (Joshi et al., 2017)	Factual knowledge	rc.nocontext/train	Question retained	3.00%
MMLU (Hendrycks et al., 2021)	Multidomain knowledge	all/auxiliary_train	Question retained	2.50%
GSM8K (Cobbe et al., 2021)	Mathematical reasoning	main/train	Problem retained	2.00%
WinoGrande (Sakaguchi et al., 2020)	Commonsense coreference	winogrande_debiased/train	Textual template	2.00%
HellaSwag (Zellers et al., 2019)	Commonsense completion	default/train	Textual template	1.50%
PIQA (Bisk et al., 2020)	Physical commonsense	plain_text/train	Textual template	1.50%
SciQ (Welbl et al., 2017)	Science knowledge	default/train	Question retained	1.00%
Social-IQA (Sap et al., 2019)	Social commonsense	default/train	Question retained	1.00%
ARC-Challenge (Clark et al., 2018)	Science reasoning	ARC-Challenge/train	Question retained	0.75%
OpenBookQA (Mihaylov et al., 2018)	Science reasoning	main/train	Question retained	0.75%

ery surrogate from `meta-llama/Llama-3.1-8B-Instruct`. Defense experiments use `Qwen/Qwen2.5-72B-Instruct` as the victim and `Qwen/Qwen2.5-7B` as the initial surrogate. The latter is the base 7B checkpoint rather than its instruction-tuned variant.

Victims are served through OpenAI-compatible text-generation endpoints, and the attacker observes only submitted queries and returned text. Victim parameters, token probabilities, hidden states, and gradients are unavailable to the attacker. Surrogates are trained locally, and every comparison starts from the same initial checkpoint within its model family; adapters are trained independently for each attack, defense, budget, or rewriting condition. We use each checkpoint's native tokenizer and formatting convention and record the resolved model revision in the corresponding run manifest. Method-specific proxy, discriminator, paraphrasing, and translation models are reported with their implementations in Appendix B.

C.2 Extraction Query Pool

We construct a master pool of 100,000 extraction queries by combining general user instructions with task-focused questions. General instructions account for 80% of the pool and are drawn from a cleaned version of LMSYS-Chat-1M (Zheng et al., 2023). The remaining 20% draws on eleven public datasets covering factual knowledge, mathematical reasoning, science, physical and social commonsense, and multiple-choice reasoning. We report the source split and fixed proportion of each component in Table 6. SeqKD, LoRD, SODA, and GAD directly use prefixes of this pool, while Model Leeching applies its method-specific templates to selected pool queries. QEDKS instead constructs seed, template, and follow-up queries online; every query received by the victim, including a follow-up, counts toward its budget.

C.3 Prompt Formatting, Parsing, and Deduplication

Only the input side of each source example is retained, and answers or rationales are not provided to the surrogate as gold labels. The victim instead generates the response used as extraction supervision. Completion and coreference examples are converted into textual queries, while question-answering and instruction sources retain their input content. Before query collection, prompts are

Table 7: Held-out capability tasks used in the final evaluation protocol.

Task	Configuration	Split	N	Task metric
ARC-Challenge (Clark et al., 2018)	ARC-Challenge	test	1,172	Length-normalized accuracy
HellaSwag (Zellers et al., 2019)	default	validation	10,042	Length-normalized accuracy
MMLU (Hendrycks et al., 2021)	all	test	14,042	Accuracy
TruthfulQA (Lin et al., 2022)	multiple choice (MC1)	validation	817	MC1 accuracy
WinoGrande (Sakaguchi et al., 2020)	winogrande_xl	validation	1,267	Accuracy
GSM8K (Cobbe et al., 2021)	main	test	1,319	Strict numeric exact match

normalized with Unicode NFKC normalization, case folding, and whitespace collapsing for exact-duplicate removal. Prompts that exactly match held-out evaluation inputs after the same normalization are removed. We do not replicate examples to fill the pool. The resulting query text is formatted using the victim's native interface only when the request is issued; model-specific control tokens are not stored as part of the shared query content.

After preprocessing, the master pool is ordered once using seed 20260701. The query sets for $B \in \{100, 1000, 10000\}$ are prefixes of this ordering, so every smaller-budget set is a subset of each larger-budget set. This construction keeps the source proportions fixed while isolating the effect of increasing the number of victim queries.

C.4 Held-Out Evaluation Tasks

We evaluate capability on the six public tasks listed in Table 7. All evaluations are zero-shot and use the complete stated split. For multiple-choice tasks, we select the option with the highest conditional log-likelihood; we normalize by continuation length for ARC-Challenge and HellaSwag and use unnormalized scores for MMLU, TruthfulQA MC1, and WinoGrande. For GSM8K, we allow up to 512 generated tokens, extract the final numeric answer, normalize its numeric representation, and require an exact match. We compute overall ACC as the unweighted macro average of the six task-level accuracies, so large tasks do not dominate the aggregate.

Behavioral fidelity and output quality use a separate set of 3,000 open-ended prompts. We select three consecutive 1,000-query blocks at positions 10,001–13,000 in the fixed master-pool ordering, immediately after the largest extraction prefix. These prompts preserve the master pool's source proportions and are disjoint from all extraction sets with $B \leq 10000$. We issue the same prompts to the victim and every surrogate and retain the complete generated responses for paired fidelity and repetition measurements. We define the corresponding BERTScore F1 and Rep-4 computations in Appendix D.

D Metrics and Evaluation Protocols

D.1 Capability, Fidelity, and Output Quality

Capability. For each task t in the six-task suite from Appendix C.4, we compute the fraction of correctly answered examples, denoted a_t. We report the unweighted task macro average

$$\mathrm{ACC} = \frac{1}{6} \sum_{t=1}^{6} a_t. \tag{2}$$

We multiply ACC by 100 when reporting percentages. This aggregation gives every task equal weight regardless of its number of examples.

Behavioral Fidelity. We compare each surrogate response with the victim response to the same held-out prompt using BERTScore F1 (Zhang et al., 2020). We use `roberta-large`, English baseline rescaling, and the mean over the 3,000 paired responses:

$$\mathrm{BERTF1} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{BERTScoreF1}(y_i^{\mathrm{S}}, y_i^{\mathrm{V}}), \qquad N = 3000. \tag{3}$$

Here y_i^{S} and y_i^{V} are the surrogate and victim responses. Baseline rescaling can produce negative values. We assign a score of one to an empty–empty pair and zero when exactly one response is empty.

Output Repetition. We case-fold and tokenize each surrogate response with a Unicode-aware tokenizer that keeps CJK characters as individual tokens. For a response with at least four tokens, we define

$$\text{Rep-4}(y) = 1 - \frac{|\,\mathrm{uniq}(G_4(y))|}{|G_4(y)|}, \tag{4}$$

where $G_4(y)$ is the multiset of contiguous 4-grams. We macro-average this value over eligible responses and exclude responses shorter than four tokens. Lower Rep-4 indicates less within-response repetition; it does not measure correctness or similarity to the victim.

D.2 Defense-Specific Metrics

Anti-Distillation. For $m \in \{\mathrm{ACC}, \mathrm{BERTF1}\}$, we report

$$\Delta m = m_{\mathrm{defended}} - m_{\mathrm{clean}}. \tag{5}$$

Thus, a negative value means that protected responses reduce surrogate performance relative to the matched clean run. ACC differences are reported in percentage points.

Provenance. We retain each defense's native detector statistic because the scores have different meanings and scales. ADFP reports $-\log_{10}(p)$, for which larger values indicate stronger fingerprint evidence. GINSEW reports the Lomb–Scargle signal-to-noise statistic P_{snr} at the embedded frequency, for which larger values indicate stronger evidence. Radioactivity reports its detector p-value, for which smaller values indicate stronger evidence. We pair every protected or rewritten student with a clean student trained by the same extraction attack and report the raw scores rather than combining them into a single provenance metric. For visual consistency, we display Radioactivity as $-\log_{10}(p)$ in Figures 3 and 4, so larger plotted values indicate stronger evidence for every defense.

Query Detection. We treat each 50-query group as one decision unit. For attack batches $\mathcal{A}$ and held-out benign batches $\mathcal{B}$, we compute

$$\mathrm{TPR} = \frac{\sum_{b\in\mathcal{A}} \mathbb{1}[b \text{ flagged}]}{|\mathcal{A}|}, \qquad \mathrm{FPR} = \frac{\sum_{b\in\mathcal{B}} \mathbb{1}[b \text{ flagged}]}{|\mathcal{B}|}. \tag{6}$$

Lineage Verification. For DuFFin, we retain probes for which the victim and every evaluated model yield a valid option and compute the victim–model answer agreement

$$S_{\mathrm{DuFFin}}(f) = \frac{1}{K}\sum_{i=1}^{K} \mathbb{1}[c_i(f) = c_i(f_{\mathrm{V}})], \tag{7}$$

with $K = 508$ common valid probes. We compute model-level ROC-AUC by comparing the scores of the three extracted students with four negative controls; a tied positive–negative pair contributes 0.5.

Adaptive Robustness. We compare each rewritten condition with its matched defense-only condition using the corresponding provenance statistic, ACC, and BERTScore F1. We report these quantities separately so that successful signal removal is not conflated with degradation of the rewritten training data.

Table 8: Aggregate attack results at query budgets $B = 100$ and $B = 10000$. Rep-4 is the repeated 4-gram rate; bold marks the best value within each budget.

Method	$B = 100$			$B = 10000$		
	ACC ↑	BERT F1 ↑	Rep-4 ↓	ACC ↑	BERT F1 ↑	Rep-4 ↓
SeqKD	60.70	0.343	**0.1212**	60.54	0.139	0.4641
Model Leeching	**61.68**	0.337	0.1216	56.30	-0.009	**0.0475**
QEDKS	59.88	**0.350**	0.1232	57.69	**0.350**	0.1227
SODA	60.86	0.344	0.1223	**61.44**	0.159	0.4055

Table 9: Clean and defended surrogate performance under anti-distillation defenses at $B = 1000$. ACC is reported in percent; Figure 2 reports defended-minus-clean changes, for which negative values indicate degradation.

Attack	**Clean**		**ADS**		**DOGe**		**Trace Rewriting**	
	ACC	BERT F1	ACC	BERT F1	ACC	BERT F1	ACC	BERT F1
SeqKD	57.62	-0.019	56.79	-0.044	56.55	-0.042	57.28	-0.024
QEDKS	63.34	0.142	59.83	-0.074	63.77	0.180	62.56	0.090
SODA	57.68	0.043	58.26	0.041	57.48	0.043	57.09	0.037

D.3 Detection Calibration and Statistical Aggregation

For query detection, we use 1,000 benign queries matched to the attack budget. We reserve 800 queries, or 16 batches, for calibration and hold out the remaining 200 queries, or four batches, for evaluating false positives. Each attack contributes 1,000 victim-observed queries, forming 20 attack batches. We calibrate each detector's decision rule on the benign batches and freeze it before evaluating attack and held-out benign batches. MMD and PRADA use an upper 95th-percentile anomaly cutoff, while SEAT uses the corresponding two-sided benign tails; SEAT additionally calibrates its pair-similarity threshold from benign query pairs. MMD estimates its null distribution with 200 samples.

All attacks being compared use the same task examples and held-out prompts, and clean–defended and defense-only–rewritten comparisons preserve the query identifiers across conditions. We use seed 42 for evaluation and detector batching. We aggregate capability first within each task and then across tasks, while BERTScore F1 and Rep-4 are macro-averaged over held-out responses. The reported comparisons are therefore paired, descriptive comparisons under a fixed evaluation protocol.

E Detailed Experimental Results

The tables in this section provide exact values for experiments summarized in the main text. When paired with a main-text figure, the table and figure use the same experimental runs: figures emphasize changes or trajectories, whereas tables preserve raw or native-scale measurements for reproducibility.

E.1 Additional Query-Budget Results

Table 8 reports the exact results for the four attacks in the query-budget analysis at $B = 100$ and $B = 10000$; their $B = 1000$ results are reported in Table 1.

E.2 Anti-Distillation Results

We report the clean and defended values underlying Figure 2 in Table 9. The ACC values use the conditional-log-likelihood evaluation protocol; BERT F1 denotes baseline-rescaled BERTScore F1 against the same undefended teacher.

Table 10: Student capability and fidelity under provenance defenses at $B = 1000$. Parentheses give changes from Clean (ACC in percentage points); positive changes indicate higher surrogate performance, not stronger defense. BERT F1 is baseline-rescaled BERTScore F1 against the same undefended Qwen teacher.

Condition	**SeqKD**		**QEDKS**		**SODA**	
	ACC ↑	BERT F1 ↑	ACC ↑	BERT F1 ↑	ACC ↑	BERT F1 ↑
Clean	57.62	-0.019	63.34	0.142	57.68	0.043
ADFP	57.87 (+0.25)	-0.020 (-0.001)	63.89 (+0.55)	0.115 (-0.028)	57.86 (+0.17)	0.041 (-0.002)
GINSEW	57.45 (-0.17)	-0.020 (-0.001)	63.85 (+0.51)	0.175 (+0.033)	57.69 (+0.01)	0.043 (+0.000)
Radioactivity	57.55 (-0.07)	-0.023 (-0.004)	64.03 (+0.69)	0.177 (+0.034)	57.38 (-0.30)	0.040 (-0.003)

Table 11: Native provenance-detector scores for clean and protected students at $B = 1000$. ADFP reports $-\log_{10}(p)$, GINSEW reports P_{snr}, and Radioactivity reports its native p-value; arrows indicate stronger evidence.

Defense	**SeqKD**		**QEDKS**		**SODA**	
	Clean	Protected	Clean	Protected	Clean	Protected
ADFP ↑	0	0	0.0371	0.0538	0	0
GINSEW ↑	1.40×10^{-20}	2.04×10^{-20}	6.33×10^{-22}	1.49×10^{-20}	2.80×10^{-20}	7.51×10^{-21}
Radioactivity ↓	0.355	0.324	2.61×10^{-8}	2.54×10^{-8}	0.0275	0.177

E.3 Provenance Results

We report the capability and fidelity of students trained on clean and protected responses in Table 10.

We report the corresponding native detector scores in Table 11. These scores preserve each defense's original scale and direction rather than imposing an artificial common metric.

E.4 Query-Detection Results

We report batch-level query-detection results in Table 12.

E.5 Lineage-Verification Results

We report the DuFFin scores for the seven models evaluated on the 508 shared valid probes in Table 13. Across the three extracted students and four negative models, the model-level ROC-AUC is 0.167.

E.6 Adaptive-Attack Results

We report the detector scores underlying Figure 4 in Table 14. The two rewriting branches use the same clean and defense-only detector baselines. ADFP is zero in every condition and is therefore retained in the table but omitted from Figure 4.

Table 12: Query detection at $B = 1000$. Bold marks the best values within each query group, including ties; rates use calibrated thresholds.

Queries	Detector	Detection (%)	
		TPR ↑	FPR ↓
SeqKD	MMD	0	**0**
	PRADA	**5**	25
	SEAT	**5**	**0**
QEDKS	MMD	**100**	**0**
	PRADA	0	25
	SEAT	**100**	**0**

Table 13: DuFFin lineage-verification scores; higher values indicate stronger predicted lineage.

Model	Role	Score ↑
Qwen2.5-7B	Negative control	0.278
Qwen2.5-7B-Instruct	Negative control	0.705
Qwen2-7B-Instruct	Negative control	0.547
Mistral-7B-Instruct-v0.3	Negative control	0.360
SeqKD	Extracted	0.252
SODA	Extracted	0.254
QEDKS	Extracted	0.467

Table 14: Provenance signals before and after adaptive response rewriting for SeqKD at $B = 1000$. ADFP reports $-\log_{10}(p)$, GINSEW reports P_{snr}, and Radioactivity reports its native p-value; arrows indicate stronger evidence.

Defense	Rewriting	Clean	Defense-only	Rewritten
ADFP ↑	DIPPER	0	0	0
	Back-translation	0	0	0
GINSEW ↑	DIPPER	5.10×10^{-21}	4.27×10^{-22}	4.35×10^{-21}
	Back-translation	5.10×10^{-21}	4.27×10^{-22}	7.81×10^{-23}
Radioactivity ↓	DIPPER	0.901	0.309	0.528
	Back-translation	0.901	0.309	0.797