WHEN AND WHAT TO PRICE AGAINST MODEL EXTRACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

As AI systems and web-based services become more capable, an attacker can collect API responses and use them to train a substitute model, reducing the API provider's competitive advantage. A natural defense is to charge more for queries that are useful for extraction, but these charges also affect benign users. In this paper, we study when request-local pricing can provide the desired protection without imposing too much cost on benign users.

For any chosen set of queries, we find the rule that makes their answers unprofitable to acquire at the lowest benign burden, using only observable features. With random responses and purchase decisions made before observing them, protection depends on expected charges. Rewriting imposes minimum charges on substitutes, while shared features can pass these charges to benign queries. A linear program finds the cheapest rule for each protected set; choosing among these sets yields the protection–cost frontier.

Frontier-Guided Pricing (FGP) uses this frontier to construct candidate rules, then checks their protection and benign burden on independent data. It deploys the qualifying rule with the smallest burden upper bound, or abstains. Across 15 student-capability targets in five extraction settings, all nine deployments meet their prespecified mean targets on held-out runs; no candidate qualifies in the remaining six cases. The frontier thus guides both how to price and whether pricing is worth deploying.

1 INTRODUCTION

An attacker can collect responses from a web API and use them to train a substitute, or *student*, model. The resulting model can reproduce, and in some settings even surpass, some capabilities of the original service (Tramèr et al., 2016; Orekondy et al., 2019; Krishna et al., 2020; Keskar et al., 2020; He et al., 2021). Recommendation responses reveal item relationships, retrieval responses reveal ranking behavior, and diverse prompts transfer language-model behavior. By learning this information, an attacker can build a substitute that reduces the provider's competitive advantage.

A natural way to discourage extraction is to charge more for responses that are useful to an attacker. But benign users may request the same kinds of responses, so these surcharges also impose a cost on normal use. We call the expected surcharge paid per benign request the *benign burden*. We therefore ask whether a desired level of protection can be achieved within an acceptable burden, and, if so, what the minimum burden must be.

We study *request-local pricing*, where the surcharge for each query depends only on that request and the provider's response. The API response itself is unchanged, and the price does not depend on a user's past queries, so splitting an attack across multiple accounts does not reduce the total surcharge. The provider sets these surcharges using observable *features* computed from the request and response. A feature can be a single numerical score or a vector of scores. For example, a recommendation request involving rarely queried items can receive a high *demand-rarity* score, while a classification response that spreads probability across several labels has high entropy, which we use as an *output signature*. These features are available to the provider for pricing, but they need not reflect how valuable a response is to a particular attacker.

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.

Two effects can make protection costly for benign users. First, *rewriting*: an attacker may replace a valuable query with a cheaper substitute that provides the same useful information. To make the original acquisition unprofitable, the substitute route must therefore be expensive enough as well. Second, *observability*: the provider can set prices only from observable features, so request–response pairs with the same feature value must receive the same surcharge. As a result, if a substitute that needs a high charge looks the same as benign traffic to the pricing rule, those benign queries must pay that charge too.

Our main result characterizes the cheapest way to protect any chosen set of queries. Because attackers may differ in what they know and what they want to extract, the provider first specifies the attacker types it wants to protect against. For each type, it then chooses the queries whose answers should be made unprofitable to acquire; together, these sets form a *protected family*. Once the protected family is fixed, rewriting determines how much each substitute query must cost in expectation to prevent the attacker from obtaining the protected information more cheaply. The provider must implement these required charges using only its observable features, and a linear program finds the rule that does so with the smallest benign burden. When each query has a fixed response, this simplifies further: each feature value is charged according to the largest required charge among the queries that share it. Finally, different protected families can achieve the same protection target at different costs. Choosing the least costly one gives the *protection–cost frontier* (Theorem 1). Thus, the cost of protection depends not only on which answers the provider wants to protect, but also on which substitute and benign queries must share their charges.

The frontier assumes that the model quantities are known. In practice, they must be estimated from data. *Frontier-Guided Pricing* (FGP) therefore uses one part of the data to construct candidate protected families and pricing rules from single or combined features, and independent data to check whether those rules provide enough protection at an acceptable benign burden. Before evaluation, the provider fixes the available features, threat model, protection target, and a *burden cap*, the largest benign burden it is willing to accept. FGP deploys a candidate rule only if independent evaluation shows that it provides enough protection without exceeding this cap. If several rules pass, it chooses the one with the smallest upper estimate of benign burden. If none passes, FGP *abstains* and leaves prices unchanged. The right evaluation depends on what the provider wants to protect. Blocking valuable queries does not necessarily reduce the student's *capability*, or task performance, by the same amount. When capability is the goal, FGP therefore evaluates it directly by retraining and testing the student on the responses that remain affordable under each candidate rule.

Our experiments test the protection cost and deployment decisions. For the same protected family, rewriting and shared features raise minimum benign burden by a factor of 2.05–2.45 over separate query prices without rewriting (Corollary 4). No single feature is best across workloads and targets; combining features can reduce burden further. All nine deployments meet their mean-capability targets on held-out runs, while no candidate qualifies for the other six targets (Figure 1).

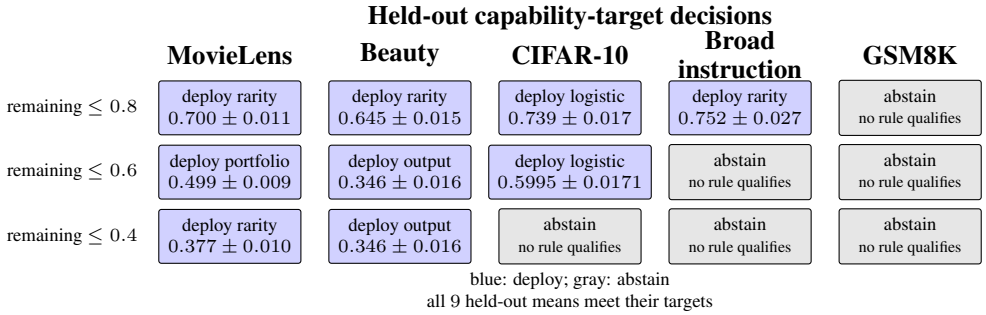


Figure 1: **Held-out capability-target decisions.** Columns are workloads; rows bound normalized remaining capability $(C - C_0)/(C_1 - C_0)$. Here C is a higher-is-better student score, C_0 the no-distillation baseline, and C_1 the unpriced baseline. With burden capped at 0.7, FGP selects the lowest-burden policy whose selection mean plus one standard error of the mean (SEM) meets the target. Blue cells report held-out means $\pm$ SEM; all nine means meet their targets. Gray cells have no qualifying policy; forced deployment fails all six targets on this split. Section 5.4 gives the protocol.

Our contributions are threefold. First, we characterize the minimum-burden pricing rule for any protected family and use this characterization to derive the protection–cost frontier. Second, we develop a sample-based procedure that constructs candidate rules and uses independent data to verify their query-level protection and benign burden with finite-sample guarantees. Third, we evaluate the resulting deployment decisions using downstream student capability directly, rather than treating blocked query value as a sufficient proxy.

2 Related Work

Pricing extraction and information. Calibrated proof-of-work (Dziedzic et al., 2022b) raises the cost of queries by requiring computational work whose difficulty depends on a leakage score and on the user's accumulated query history. We ask instead when request-local pricing can meet a protection target within a benign-burden cap, and we answer by finding the minimum-burden rule for each protected family, accounting for rewriting and shared feature values. Appendix C.4 compares the two approaches at the same benign-user cost.

Database pricing also accounts for alternative ways to obtain related answers (Koutris et al., 2012; Deep & Koutris, 2017), but it prevents arbitrage among database products, whereas our goal is to make selected acquisitions unprofitable. Information pricing and screening (Bergemann et al., 2018; Myerson, 1981) provide a broader economic context; in our model, extraction values are declared in the threat model rather than elicited from attackers.

Extraction settings. Extraction attacks differ in how they choose queries and in what they learn from responses. Studies of prediction-API extraction establish what an attacker can recover from model outputs (Tramèr et al., 2016; Jagielski et al., 2020). Queries can come from public data used for distillation (Orekondy et al., 2019), active selection (Chandrasekaran et al., 2020; Pal et al., 2020), or synthesis (Kariyappa et al., 2021; Truong et al., 2021). Text and language-model extraction (Krishna et al., 2020; Wallace et al., 2020; Carlini et al., 2024; Liang et al., 2025; Li et al., 2026) and representation extraction (Dziedzic et al., 2022a; Tamber et al., 2025) widen the range of information an attacker may seek. We represent these goals and prior knowledge by attacker types, each with its own extraction values. Following the emphasis on explicit threat models in extraction evaluation (Libon et al., 2026), our guarantees hold for the declared types, rewrite relations, and observable features.

Other defenses. Detectors monitor query patterns or account-level behavior (Juuti et al., 2019; Kesarwani et al., 2018), including through distributional tests (Gretton et al., 2012; Liu et al., 2026). Active defenses modify inputs, predictions, or training signals (Orekondy et al., 2020; Kariyappa & Qureshi, 2020; Mazeika et al., 2022; Wang et al., 2023; Tang et al., 2024; Allouah et al., 2026; Ma et al., 2026), and some optimize these changes against the attacker's learning objective (Xian et al., 2022; Allouah et al., 2026). Watermarks and similarity tests support ownership verification after extraction (Jia et al., 2021; Shen et al., 2026). Our pricing rules leave responses unchanged and use no query history across requests, unlike per-client defenses that multi-account attacks can evade (Schwarzer et al., 2026): the same query–response records incur the same total surcharge however they are divided among accounts. Detection remains a complementary option (Oliynyk et al., 2023); when FGP abstains, no candidate rule has qualified, and any other defense needs its own evaluation.

3 Model and Objective

This section defines what attackers value, what the provider can observe and charge, and when a pricing rule deters extraction.

3.1 Queries, demand, and extraction value

Let $\mathcal{X}$ be a finite set of API queries, shared by benign users and attackers. Benign queries are drawn from a distribution D over $\mathcal{X}$; some queries may never appear in benign traffic.

Attackers can differ in prior knowledge and extraction goals, which is represented by a finite set Θ of declared attacker types. Protection is required for every type separately, so no probability distribution over types is needed. For an attacker of type θ, let $g_\theta(x) \geq 0$ be the finite *extraction value* of the information supplied by the answer to x; obtaining that information is an *acquisition*. If the API is randomized, $g_\theta(x)$ is the expected value before the response is observed. Extraction values and acquisition costs are measured on the same utility scale.

We use the same symbol for the total value of a query set $A \subseteq \mathcal{X}$:

$$g_\theta(A) := \sum_{x \in A} g_\theta(x), \qquad V_\theta = g_\theta(\mathcal{X}). \tag{1}$$

Thus $g_\theta(A)$ is the total extraction value of the answers in A, and V_θ is the total available value. The main text focuses on protection against attackers seeking to maximize this additive value. Since the capability of a student model trained on extracted answers can depend on how those answers are combined, Appendix A.2 additionally considers capability-based protection targets, which we test empirically in Section 5.4. Either way, the purchase model below treats acquisitions one at a time.

3.2 Observable features, pricing, and rewriting

A query $x \in \mathcal{X}$ produces a response record $y \in \mathcal{Y}$ drawn from $\pi(\cdot \mid x)$. The record contains the delivered answer and any provider-side statistics used for pricing, such as class probabilities. The attacker need not observe these statistics, and extraction value refers only to information in the delivered answer. An *observable feature* summarizes the query and its response record. Formally, it is a fixed map $\varphi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^J$: a single score when $J = 1$, or a vector of J scores otherwise. Demand rarity uses only the query, whereas an *output signature* uses response entropy (Section 5.1).

A *pricing rule* $s : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}_+$ assigns a nonnegative surcharge using only this feature:

$$s(x, y) = a(\varphi(x, y)), \qquad a \geq 0, \tag{2}$$

where a maps each feature value to a surcharge, so identical feature values receive identical surcharges.

The attacker needs to decide whether to pay before seeing the response, and we assume risk-neutral decisions: an acquisition is *profitable* when its expected value exceeds its expected cost. Write $\bar{s}(x) = \mathbb{E}_{y \sim \pi(\cdot|x)}[s(x, y)]$ for the expected surcharge. The surcharge itself stays response-level; only the purchase decision uses its expectation. The response distribution $\pi(\cdot \mid x)$ is fixed for each query and is the same for benign users and attackers, and each acquisition obtains one response, from the original query or from a substitute defined below.[1]

Rewriting lets the attacker exploit differences in expected surcharge between queries that carry the same useful information. Let $R^*(x) \subseteq \mathcal{X}$ contain the declared substitutes for x, including those reachable through several rewrites. A substitute supplies the information valued by $g_\theta(x)$, though not necessarily an identical answer. Let $c^*(x, x') \geq 0$ be the minimum cost of using x' in place of x, in addition to the expected surcharge on x'. The original query remains an option: $x \in R^*(x)$ and $c^*(x, x) = 0$. The attacker's *effective acquisition cost* for the information associated with x is therefore

$$s^e(x) = \min_{x' \in R^*(x)} \left\{ \bar{s}(x') + c^*(x, x') \right\}, \tag{3}$$

the cheapest route once both the expected surcharge and the rewrite cost are counted. Without rewriting, $R^*(x) = \{x\}$ and $s^e(x) = \bar{s}(x)$. The substitute sets and rewrite costs specify the threat model; Appendix A.1 studies uncertainty in that model and missing rewrites.

3.3 Purchase decisions and ε-deterrence

Given a pricing rule, a type-θ attacker acquires the information in x when its value $g_\theta(x)$ strictly exceeds its effective acquisition cost $s^e(x)$.[2] The *profitable set* is

$$P_\theta(s) = \{x \in \mathcal{X} : g_\theta(x) > s^e(x)\}. \tag{4}$$

[1] Repeated sampling and stopping based on observed responses require a different acquisition model.

[2] We take ties to mean no purchase. Otherwise, adding any $\eta > 0$ to a qualifying rule restores deterrence, so the optimal burden has the same infimum but may not be attained.

Table 1: **Notation for the pricing decision.**

Symbol	Meaning
$\mathcal{X}, D; \theta \in \Theta$	query set and benign demand; declared attacker type
$g_\theta(x), g_\theta(A), V_\theta$	extraction value per query, per query set, and in total
$y \sim \pi(\cdot \mid x)$; $\varphi(x, y)$	response record; observable feature
$s(x, y), \bar{s}(x), s^e(x)$	realized surcharge, expected surcharge, effective acquisition cost
R^*, c^*	reachable rewrites and their costs
$P_\theta(s)$	profitable query set under s
$\mathbf{A}, h_{\mathbf{A}}, s_{\mathbf{A}}$	protected family, its rewrite floor, and its minimum-burden rule
$B(s), B^*_\varphi(\varepsilon), B_{\max}$	benign burden, its optimal frontier, and the allowed cap

The remaining queries are protected: their information is unprofitable to acquire, although the API still answers them. The *remaining extraction value* is $g_\theta(P_\theta(s))$, and the *blocked value* is $V_\theta - g_\theta(P_\theta(s))$.

For $\varepsilon \in [0, 1]$, a rule is *ε-deterring* if every declared type can still profit from at most an ε fraction of its total value, that is, if at least a $1 - \varepsilon$ fraction is blocked:

$$g_\theta(P_\theta(s)) \le \varepsilon V_\theta \quad \iff \quad V_\theta - g_\theta(P_\theta(s)) \ge (1 - \varepsilon)V_\theta \qquad (\forall \theta \in \Theta). \tag{5}$$

Full deterrence is $\varepsilon = 0$.

Among rules meeting this target, the provider minimizes the *benign burden*, the expected surcharge per benign request:

$$B(s) = \mathbb{E}_{x \sim D,\, y \sim \pi(\cdot|x)}[s(x, y)] = \sum_{x \in \mathcal{X}} D(x)\bar{s}(x). \tag{6}$$

It can be estimated from traffic identified as benign independently of the attack evaluation. A *burden cap* $B_{\max}$ limits what the provider is willing to impose on benign users. Section 4 finds the minimum burden and hence whether the protection target can be met within this cap.

4 Frontier-Guided Pricing

Section 3 defined when an acquisition is unprofitable and how much benign users pay. We now fix the observable feature, find the cheapest rule for any protected family, and use this construction to build and validate candidate rules from samples.

4.1 Fixing the observable feature

From here on, all theory uses one fixed observable feature map φ. It may be a single score or a vector $\varphi(x, y) = (\varphi_1(x, y), \ldots, \varphi_J(x, y))$ of several features, so combining features means pricing their joint vector.

Besides general rules of the form (2), our experiments use a simpler class, *feature portfolios*. Let $z_j(x, y) \ge 0$ denote a nonnegative, normalized version of feature $\varphi_j(x, y)$; a portfolio adds these scores with nonnegative weights:

$$s_w(x, y) = \sum_j w_j z_j(x, y), \qquad w_j \ge 0.$$

Portfolios are easy to fit and compare but form a restricted class. Theorem 1 gives the optimum over all functions of the joint feature, and Appendix A.1 gives a separate linear program (LP) for the restricted portfolio class.

4.2 The protection–cost frontier

For each attacker type θ, suppose the provider chooses a set $A_\theta \subseteq \mathcal{X}$ of queries whose answers should be unprofitable to acquire. We call $\mathbf{A} = (A_\theta)_{\theta \in \Theta}$ a *protected family*. It meets the protection target when each type's protected set holds at least a $(1 - \varepsilon)$ fraction of that type's total extraction value. These families form

$$\mathcal{A}_\varepsilon = \{\mathbf{A} : \; A_\theta \subseteq \mathcal{X}, \; g_\theta(A_\theta) \ge (1 - \varepsilon)V_\theta \text{ for every } \theta\}. \tag{7}$$

For a fixed family, the cheapest rule follows in two steps: find the smallest expected surcharge on each query that makes every declared acquisition route unprofitable, then meet these requirements using only the observable feature.

Account for rewriting. Suppose $x \in A_\theta$ should be protected. If the attacker can instead use $x' \in R^*(x)$ to obtain the same useful information, the expected surcharge on x' plus the rewrite cost must be at least $g_\theta(x)$. The expected surcharge on x' must therefore be at least $g_\theta(x) - c^*(x, x')$, or zero if that difference is negative. The largest such requirement over all protected queries and types is the *rewrite floor*

$$h_{\mathbf{A}}(x') = \max_{\substack{\theta \in \Theta,\ x \in A_\theta \\ x' \in R^*(x)}} [g_\theta(x) - c^*(x, x')]_+, \tag{8}$$

where $[u]_+ = \max\{u, 0\}$ and an empty maximum is zero. Any pricing rule that protects $\mathbf{A}$ must satisfy $\bar{s}(x') \geq h_{\mathbf{A}}(x')$ on every query x'.

Enforce observability. Let $\mathcal{Z}$ be the set of feature values that can occur. We assume it is finite for the LP below; the response records themselves need not be. This covers fixed cached responses and finite-precision or discretized scores. Let $p_x(z) = \Pr_{y \sim \pi(\cdot|x)}\{\varphi(x, y) = z\}$ be the probability that query x produces feature value z. The decision variable is the surcharge $a_z = a(z)$ assigned to each observable feature value z. It contributes $p_x(z)a_z$ to the expected surcharge of query x, so the cheapest surcharges that meet the rewrite floor solve

$$\begin{aligned} a_{\mathbf{A}} \in \arg\min_{a_z \geq 0} \ & \sum_{x \in \mathcal{X}} D(x) \sum_{z \in \mathcal{Z}} p_x(z) a_z \\ \text{subject to } & \sum_{z \in \mathcal{Z}} p_x(z) a_z \geq h_{\mathbf{A}}(x) \quad (\forall x \in \mathcal{X}). \end{aligned} \tag{9}$$

Choose any optimizer, which need not be unique, and set $s_{\mathbf{A}}(x, y) = a_{\mathbf{A}, \varphi(x,y)}$. This gives the sequence $\mathbf{A} \mapsto h_{\mathbf{A}} \mapsto s_{\mathbf{A}} \mapsto B(s_{\mathbf{A}})$: choose what to protect, meet the required expected surcharges, and measure benign burden.

If each query has a fixed response y_x, it always produces the same feature value, and the program reduces to a simple rule: set each feature value's surcharge to the largest floor among the queries that produce it,

$$s_{\mathbf{A}}(x, y_x) = \max_{\substack{u \in \mathcal{X} \\ \varphi(u, y_u) = \varphi(x, y_x)}} h_{\mathbf{A}}(u). \tag{10}$$

This rule is the *feature closure*. For fixed responses it is no larger at any query than any other rule protecting the same family; with random responses, the general program (9) minimizes expected burden instead.

For a fixed feature φ and protection level ε, the *protection–cost frontier* is the smallest benign burden among all feature-based rules that leave at most an ε fraction of the extraction value profitable for every attacker type:

$$B^*_\varphi(\varepsilon) = \min_{\substack{s = a \circ \varphi,\ a \geq 0 \\ g_\theta(P_\theta(s)) \leq \varepsilon V_\theta \ \forall \theta}} B(s). \tag{11}$$

Theorem 1 (Protection–cost frontier)**.** *Under the finite query, attacker-type, and feature-range assumptions above,*

$$B^*_\varphi(\varepsilon) = \min_{\mathbf{A} \in \mathcal{A}_\varepsilon} B(s_{\mathbf{A}}).$$

For any protected family $\mathbf{A}$*, program* (9) *attains* $B(s_{\mathbf{A}})$*, the least burden among rules protecting that family. Hence the frontier is an optimization over qualifying families, with a finite linear program for each family.*

The theorem separates *which acquisitions to protect* from *how to charge for that protection*. Once the feature is fixed, pricing reduces to choosing a qualifying protected family: for any chosen family, rewriting and the feature determine the cheapest rule, through the LP or, for fixed responses, the feature closure. Choosing the family remains a combinatorial problem. The target is affordable exactly when $B^*_\varphi(\varepsilon) \leq B_{\max}$. If this minimum exceeds the cap, retuning the surcharge cannot make the target affordable with the same feature. The provider must change the feature, relax the requirement, or abstain.

A protection target can also bound the capability obtainable from the unprotected answers rather than their summed extraction value. The same construction applies whenever protecting more queries preserves the target; this changes which families qualify, not how a chosen family is priced (Appendix A.2). Appendix A.4 proves Theorem 1.

4.3 Constructing and validating rules from samples

FGP estimates the frontier's inputs from *selection samples* and validates candidates on independent *audit samples*. The observable features, rewrite model, attacker types, protection target, and burden cap are fixed before the audit.

It first *constructs* candidate rules: using selection estimates, FGP proposes protected families $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(K)}$ and maps them to fitted rules s_k through the rewrite floor and (9). It then *retains* the lowest-burden candidate meeting the target and cap on selection data, or a finite menu of such candidates, fully specified before the audit. Finally, it *audits* their achieved protection and burden. It deploys the candidate with the smallest burden upper bound among those whose protection lower bounds meet every type's target and whose burden upper bound meets the cap; otherwise, it *abstains*.

These checks evaluate the actual rules using effective acquisition costs under the response distribution, not single responses or proposed protected families. Simultaneous bounds allow selection among passing rules without losing the guarantee. Algorithm 1 gives the procedure; portfolios use the restricted LP in Appendix A.1. Section 5.3 tests this audit, while Section 5.4 evaluates capability through retraining with a separate empirical selection criterion.

Proposition 2 (Validation and candidate-menu performance)**.** *Fix K candidate rules independently of the audit samples. Assume bounded surcharges and bounded, unbiased blocked-value estimates under the sampling conditions of Proposition 10. Let Δ_B and Δ_P be its simultaneous burden and protection error margins for failure probability $\delta \in (0, 1)$. Then, with probability at least $1 - \delta$:*

1. *every rule deployed by FGP is ε-deterring and satisfies $B(s) \le B_{\max}$;*
2. *if the retained menu contains a candidate rule with blocked value at least $(1 - \varepsilon)V_\theta + 2\Delta_P$ for every type θ and burden at most $B_{\max} - 2\Delta_B$, then FGP deploys a rule whose burden is at most $2\Delta_B$ above the cheapest candidate rule in the menu satisfying these margins.*

For a fixed failure probability and number of attacker types, the margins shrink at rate $O(\sqrt{\log(2K)/n})$ in the relevant audit sample size n. More audit data therefore lets good candidate rules qualify more often and reduces the extra burden that selection by confidence bounds can cost. The guarantee is relative to the retained menu, not to all pricing rules. Appendix A.3 gives the detailed bounds, the proof, and the sampling weights for estimating (1); if expected surcharges must also be estimated, their uncertainty needs a separate bound. A change in benign demand, the response distribution, or the declared threat model requires a new audit.

5 Experiments

The experiments connect the frontier to deployment in three steps. We first measure how rewriting and observable features affect protection cost. We then test whether feature choices are stable across samples and whether the audit validates the selected rules. Finally, we evaluate deployment decisions using downstream student capability.

5.1 Experimental setup

Our capability study uses five workloads: CIFAR-10 image classification, MovieLens and Amazon Beauty recommendation, broad instruction following, and GSM8K math. Additional experiments use AG News for text rewriting and MS MARCO for retrieval (Appendix B.1). Each workload supplies benign queries, attacker queries, victim responses, and a declared extraction-value proxy g_θ. Pricing evaluations reuse one cached response record per query, including generation statistics where used, and therefore instantiate the fixed-response case of the theory. We report the blocked-value fraction

$1 - g_\theta(P_\theta(s))/V_\theta$ under the actual rule; meeting the target $1 - \varepsilon$ for every type gives ε-deterrence. Student capability is measured separately by retraining on the acquired responses.

Our two main features are demand rarity, which is high for queries uncommon in benign traffic, and an output signature derived from entropy: of class probabilities for classification, next-item probabilities for recommendation, token probabilities for language models, and softmax-normalized similarities to reference embeddings for retrieval. AG News also tests lexical rarity based on benign word frequencies. Appendix B.2 specifies feature normalization and the value proxies. In the language rewriting and LLM studies, the output signature and g_θ use the same raw response entropy with different scaling; retraining separately tests whether the resulting rules reduce capability.

We evaluate single-feature rules and nonnegative feature portfolios under the same attacker budget and benign-burden constraint. Baselines include no surcharge, flat pricing, fixed rarity, fixed output signature, equal weighting after benign-mean normalization, and a logistic score trained to distinguish attacker queries from benign queries using selection data.

5.2 Protection cost and feature choice

Holding the protected family fixed isolates the cost of rewriting and limited observability. On CIFAR-derived finite environments with known D, g_θ, R^*, c^*, these constraints raise the minimum benign burden by a factor of 2.05–2.45 relative to a benchmark that prices each query separately and allows no substitutes (Table 3). The protected queries are the same in both cases, so the increase comes only from implementing that choice.

The cheapest feature also changes with the protection target. In the cross-domain experiments, we search a finite set of portfolio weightings and choose the smallest surcharge scale reaching the target for each weighting. This gives the best rule in the declared menu, not the optimum over all feature-based rules. On CIFAR-10, the output signature requires benign burden 0.19 versus 0.54 for rarity at a 30% blocked-value target. At an 80% target, the ordering reverses: rarity requires 1.04 versus 7.24 for the output signature. At the latter target, the portfolio reduces burden to 1.00, or 0.80 when the logistic feature is also available (Table 4). Across $1{,}408$ matched seed–burden comparisons, the selected portfolio blocks at least as much declared extraction value as the best fixed feature. These results, with the equal-weighting comparison of Appendix C.1, support optimizing within the declared feature class rather than committing to one score or one weighting for every target.

5.3 Selection stability and audit reliability

We first test whether the feature choice is stable across samples. For each of 36 splits of cached vision attack–seed results, we compare the choice on one half with the best choice on the other: rarity, the output signature, their fixed portfolio, or abstention. The choices agree in all 36 splits: 18 choose the portfolio and 18 abstain. Each half refits the surcharge scale, so this tests the choice rather than one fixed numerical rule. As attacker queries overlap more with benign demand, the procedure moves from selecting a candidate to abstaining (Appendix C.2).

We then hold the rules fixed to isolate audit sampling error, using a known finite population. At each sample size, all deployed rules meet the 90% blocked-value target across $2{,}000$ Monte-Carlo audits, and all reported burden bounds cover the true burden. The deployment rate rises from 0.09 at 2k samples to 1.00 at 8k, and the excess burden of deployed rules over the cheapest menu candidate meeting the true target falls from 1.01 at 2k to 0.22 at 16k (Appendix C.2). This experiment imposes no burden cap ($B_{\max} = \infty$), so it isolates the effect of audit uncertainty.

5.4 Capability-based deployment

Blocked value is only a proxy when the provider's goal is to limit a student's downstream capability. We therefore retrain the student on the responses acquired under each candidate pricing policy and select using capability directly. A pricing rule assigns the surcharge $s(x, y)$; a pricing policy specifies how to fit the rule, which is refitted for each evaluation seed. Across 15 workload–target pairs, FGP deploys in nine, and all nine meet their prespecified mean-capability targets on held-out seeds. No candidate policy qualifies in the other six (Figure 1).

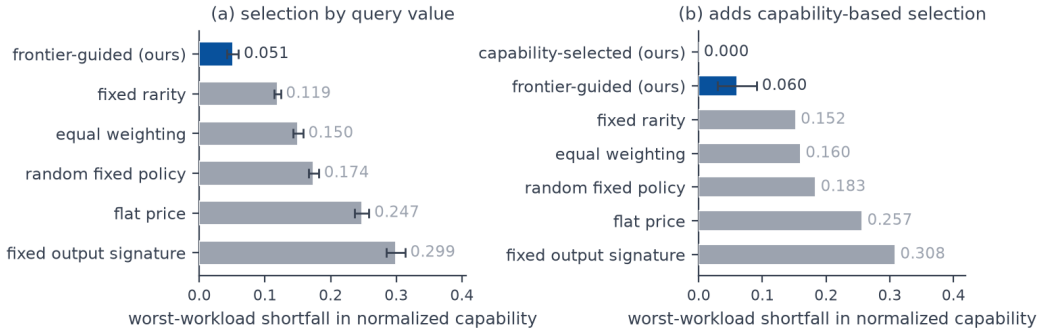


Figure 2: **Capability shortfall depends on the selection objective. (a)** compares choices made using blocked value; **(b)** adds capability-based selection. Shortfall is the largest workload-level excess in mean remaining capability over the best evaluated policy in that panel; lower is better. Capability-based selection has zero on this comparison. Bars show one SEM of the paired comparison; Appendix C.3 specifies the averaging and selection protocol.

We consider normalized remaining-capability targets of 0.8, 0.6, and 0.4 (Figure 1). A target of 0.4 allows at most 40% of the capability gain from unpriced extraction, not 40% raw accuracy. We use negative held-out loss as the score for language-model fidelity. On the selection seeds, FGP chooses the lowest-burden policy and operating point whose mean capability plus one SEM is below the target within the burden cap $B_{\max} = 0.7$; disjoint seeds then refit and evaluate that choice. This is a conservative empirical selection rule, distinct from the finite-sample audit guarantee in Section 4.3.

MovieLens and Beauty qualify at all three targets, CIFAR at two, broad instruction at one, and GSM8K at none. For the six abstentions, forcing deployment of the policy and operating point with the lowest mean capability plus one SEM on the selection seeds fails every held-out target on the reported split (Appendix C.3). GSM8K illustrates why a blocked-value result alone cannot qualify a policy for a capability target.

The capability results also depend on the feature. No fixed feature is best across workloads: the output signature is stronger on CIFAR and Beauty, and rarity on MovieLens and broad instruction. FGP selected by blocked value leaves a worst-workload capability shortfall of 0.060 ± 0.031 relative to the best evaluated policy; selecting by capability has zero shortfall in this comparison (Figure 2).

AG News shows why the two objectives can differ. Best-of-8 rewriting leaves the output signature blocking 78.3% of declared value versus 1.9% for lexical rarity, yet student accuracy falls only from 0.892 to 0.869, consistent with the task saturating after relatively few examples (Appendix C.3). Output agreement with the victim can likewise fall more than useful task performance at the same burden (Figure 9). The query-level frontier constructs and compares candidate rules; evaluating the provider's chosen objective decides which of them qualify.

6 Conclusion

Whether request-local pricing is useful depends on more than how much an answer is worth to an attacker. Rewriting can force surcharges onto substitute queries, and limited observability can pass those surcharges on to benign users. The protection–cost frontier accounts for both effects and gives the least benign burden that meets a declared protection target.

FGP turns this characterization into a data-driven choice of what to price and whether to deploy. Across the workloads studied, no feature is best for every target, and blocking query value does not automatically reduce downstream capability. When capability is evaluated directly, all nine deployments meet their prespecified mean-capability targets on held-out seeds, and FGP abstains in the other six workload–target pairs. Together, the frontier and audit turn pricing into an applicability decision: deploy only when the available features support the required protection at an acceptable benign cost, and abstain otherwise.

Ethics statement

Feature choice determines who pays. Appendix C.4 reports surcharge incidence across atypicality levels and explains how tail-burden constraints enter family selection. The benchmarks lack group labels, so these results do not establish demographic fairness. The work studies a defense and introduces no new attack.

AI use statement

We used large language model (LLM) assistants to revise and polish the writing, to help write and debug experiment and plotting code, and to cross-check reported numbers and references against saved results and original sources. The research questions, the model, and the experimental design are the authors' own. All reported numbers come from the experiments described in the paper. The authors verified all results, proofs, and citations, and take full responsibility for the content.

Reproducibility statement

Section 3 defines the model. Every formal result is stated with its assumptions in Section 4 or Appendix A and proved in Appendix A.4. Appendix B specifies the workloads, datasets, victim and student models with key training settings, feature definitions, extraction-value proxies, and seeds, and gives the FGP procedure as pseudocode (Algorithm 1). All datasets and pretrained checkpoints we use are publicly available. Appendix C details each experiment behind the main-text summaries, including the fixed-menu audits, transfer comparisons, and seed-level evaluations of capability-based selection. The code is not included with this submission; we will open-source it after the review process.

References

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

Dirk Bergemann, Alessandro Bonatti, and Alex Smolin. The design and price of information. *American Economic Review*, 108(1):1–48, 2018. doi: 10.1257/aer.20161079.

Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A. Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Conmy, Eric Wallace, David Rolnick, and Florian Tramèr. Stealing part of a production language model. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 5680–5705, Vienna, Austria, 2024. PMLR. URL https://proceedings.mlr.press/v235/carlini24a.html.

Varun Chandrasekaran, Kamalika Chaudhuri, Irene Giacomelli, Somesh Jha, and Songbai Yan. Exploring connections between active learning and model extraction. In *29th USENIX Security Symposium (USENIX Security 20)*, pp. 1309–1326. USENIX Association, 2020. ISBN 978-1-939133-17-5. URL https://www.usenix.org/conference/usenixsecurity20/presentation/chandrasekaran.

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.

Shaleen Deep and Paraschos Koutris. QIRANA: A framework for scalable query pricing. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pp. 699–713, Chicago, IL, USA, 2017. ACM. doi: 10.1145/3035918.3064017.

Adam Dziedzic, Nikita Dhawan, Muhammad Ahmad Kaleem, Jonas Guan, and Nicolas Papernot. On the difficulty of defending self-supervised learning against model extraction. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5757–5776, Baltimore, Maryland, USA, 2022a. PMLR. URL https://proceedings.mlr.press/v162/dziedzic22a.html.

Adam Dziedzic, Muhammad Ahmad Kaleem, Yu Shen Lu, and Nicolas Papernot. Increasing the cost of model extraction with calibrated proof of work. In *International Conference on Learning Representations*, Virtual Conference, 2022b. OpenReview.net. URL https://openreview.net/forum?id=EAy7C1cgE1L.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.

F. Maxwell Harper and Joseph A. Konstan. The MovieLens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4):19:1–19:19, 2015. doi: 10.1145/2827872.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, 2016. IEEE. doi: 10.1109/CVPR.2016.90.

Xuanli He, Lingjuan Lyu, Qiongkai Xu, and Lichao Sun. Model extraction and adversarial transferability, your BERT is vulnerable! In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2006–2012, Online, 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.161.

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*, Virtual Conference, 2022. OpenReview.net. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. High accuracy and high fidelity extraction of neural networks. In *29th USENIX Security Symposium (USENIX Security 20)*, pp. 1345–1362. USENIX Association, 2020. ISBN 978-1-939133-17-5. URL https://www.usenix.org/conference/usenixsecurity20/presentation/jagielski.

Hengrui Jia, Christopher A. Choquette-Choo, Varun Chandrasekaran, and Nicolas Papernot. Entangled watermarks as a defense against model extraction. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 1937–1954. USENIX Association, 2021. ISBN 978-1-939133-24-3. URL https://www.usenix.org/conference/usenixsecurity21/presentation/jia.

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, Stockholm, Sweden, 2019. IEEE. doi: 10.1109/EuroSP.2019.00044.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 197–206, Singapore, 2018. IEEE. doi: 10.1109/ICDM.2018.00035.

Sanjay Kariyappa and Moinuddin K. Qureshi. Defending against model stealing attacks with adaptive misinformation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 767–775, Seattle, WA, USA, 2020. IEEE Computer Society. doi: 10.1109/CVPR42600.2020.00085.

Sanjay Kariyappa, Atul Prakash, and Moinuddin K. Qureshi. MAZE: Data-free model stealing attack using zeroth-order gradient estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13809–13818, Nashville, TN, USA, 2021. IEEE Computer Society. doi: 10.1109/CVPR46437.2021.01360.

Manish Kesarwani, Bhaskar Mukhoty, Vijay Arya, and Sameep Mehta. Model extraction warning in MLaaS paradigm. In *Proceedings of the 34th Annual Computer Security Applications Conference*, pp. 371–380, San Juan, PR, USA, 2018. Association for Computing Machinery. doi: 10.1145/3274694.3274740.

Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. The thieves on Sesame Street are polyglots - extracting multilingual models from monolingual APIs. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6203–6207, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.501. URL https://aclanthology.org/2020.emnlp-main.501/.

Paraschos Koutris, Prasang Upadhyaya, Magdalena Balazinska, Bill Howe, and Dan Suciu. Query-based data pricing. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pp. 167–178, Scottsdale, AZ, USA, 2012. Association for Computing Machinery. doi: 10.1145/2213556.2213582.

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*, Virtual Conference, 2020. OpenReview.net. URL https://openreview.net/forum?id=Byl5NREFDr.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

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*, volume 40, pp. 31870–31878, Singapore, 2026. Association for the Advancement of Artificial Intelligence. doi: 10.1609/aaai.v40i38.40456.

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, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.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.

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.

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, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.519. URL https://aclanthology.org/2026.acl-long.519/.

Mantas Mazeika, Bo Li, and David Forsyth. How to steer your adversary: Targeted and efficient model stealing defenses with gradient redirection. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15241–15254, Baltimore, Maryland, USA, 2022. PMLR. URL https://proceedings.mlr.press/v162/mazeika22a.html.

Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 43–52, Santiago, Chile, 2015. Association for Computing Machinery. doi: 10.1145/2766462.2767755.

Roger B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1):58–73, 1981. doi: 10.1287/moor.6.1.58.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating Neural and Symbolic Approaches*

2016, co-located with NIPS 2016, volume 1773 of *CEUR Workshop Proceedings*, Barcelona, Spain, 2016. CEUR-WS.org. URL https://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf.

Daryna Oliynyk, Rudolf Mayer, and Andreas Rauber. I know what you trained last summer: A survey on stealing machine learning models and defences. *ACM Computing Surveys*, 55(14s): 324:1–324:41, 2023. doi: 10.1145/3595292.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4949–4958, Long Beach, CA, USA, 2019. IEEE Computer Society. doi: 10.1109/CVPR.2019.00509.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Prediction poisoning: Towards defenses against DNN model stealing attacks. In *International Conference on Learning Representations*, Virtual Conference, 2020. OpenReview.net. URL https://openreview.net/forum?id=SyevYxHtDB.

Soham Pal, Yash Gupta, Aditya Shukla, Aditya Kanade, Shirish Shevade, and Vinod Ganapathy. ActiveThief: Model extraction using active learning and unannotated public data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(1):865–872, 2020. doi: 10.1609/aaai.v34i01.5432.

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 (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108, 2019.

Maxime Schwarzer, Johannes F. Loevenich, Gustavo Sánchez, Laurin Holz, Thies Möhlenhof, Tobias Hürten, Roberto Rigolin F. Lopes, and Veit Hagenmeyer. AI model extraction attacks: Bypassing single-client assumptions in defenses. arXiv preprint arXiv:2606.03381, 2026. Presented at the International Conference on Military Communication and Information Systems (ICMCIS), Bath, United Kingdom, May 2026.

Bolin Shen, Zhan Cheng, Neil Zhenqiang Gong, Fan Yao, and Yushun Dong. CREDIT: Certified ownership verification of deep neural networks against model extraction attacks. In *Proceedings of the 43rd International Conference on Machine Learning*, Seoul, Republic of Korea, 2026. URL https://openreview.net/forum?id=DQKoxCegwH.

Manveer Singh Tamber, Jasper Xian, and Jimmy Lin. Can't hide behind the API: Stealing black-box commercial embedding models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1958–1969, Albuquerque, New Mexico, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-naacl.104.

Minxue Tang, Anna Dai, Louis DiValentin, Aolin Ding, Amin Hass, Neil Zhenqiang Gong, Yiran Chen, and Hai Helen Li. ModelGuard: Information-theoretic defense against model extraction attacks. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 5305–5322, Philadelphia, PA, USA, 2024. USENIX Association. ISBN 978-1-939133-44-1. URL https://www.usenix.org/conference/usenixsecurity24/presentation/tang.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford Alpaca: An instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

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

Jean-Baptiste Truong, Pratyush Maini, Robert J. Walls, and Nicolas Papernot. Data-free model extraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4769–4778, Nashville, TN, USA, 2021. IEEE Computer Society. doi: 10.1109/CVPR46437.2021.00474.

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, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.446.

Zhenyi Wang, Li Shen, Tongliang Liu, Tiehang Duan, Yanjun Zhu, Donglin Zhan, David Doermann, and Mingchen Gao. Defending against data-free model extraction by distributionally robust defensive training. In *Advances in Neural Information Processing Systems*, volume 36, pp. 624–637, New Orleans, LA, USA, 2023. Curran Associates, Inc. doi: 10.52202/075280-0029.

Xun Xian, Mingyi Hong, and Jie Ding. A framework for understanding model extraction attack and defense. arXiv preprint arXiv:2206.11480, 2022.

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. arXiv preprint arXiv:2412.15115, 2025. URL `https://arxiv.org/abs/2412.15115`.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28, pp. 649–657, Montreal, Quebec, Canada, 2015. Curran Associates, Inc.

APPENDIX

Appendix A extends the frontier, states the finite-sample audit, and proves every result. Appendix B specifies the workloads, features, and procedure. Appendix C gives the per-experiment evidence in the order of Section 5.

A FRONTIER THEORY AND PROOFS

We use the response-level rule $s(x, y)$ and expected surcharge $\bar{s}(x)$ from Section 3. For the proofs, extend (9) to any finite nonnegative floor h on $\mathcal{X}$, replacing $h_{\mathbf{A}}$ by h. Write $C_\varphi(h)$ for its minimum objective and s_h for any optimizing rule. Thus $B(s_h) = C_\varphi(h)$ and $B(s_{\mathbf{A}}) = C_\varphi(h_{\mathbf{A}})$. We also abbreviate the benign feature frequency by $d(z) = \sum_x D(x)p_x(z)$. This shorthand simplifies the dual and robustness bounds below.

A.1 FRONTIER RESULTS

Random responses and fixed-response closure. The covering program uses the whole conditional feature distribution p_x: features with the same mean can require different expected charges. For example, a feature that is always 1 and a feature equally likely to be 0 or 2 have the same mean, but the rule $a(z) = z^2$ charges them 1 and 2 in expectation.

A pointwise-smallest protecting rule also need not exist. Suppose one protected query requires expected charge 1 and yields feature values z_0, z_1 with equal probability. Both charge vectors $(2, 0)$ and $(0, 2)$ protect it, but the only nonnegative vector no larger than both is $(0, 0)$, and it does not. If benign queries always yield z_0, the second vector has zero burden. Charging at least 1 on every realized response would cost benign users 1 and solve a stricter problem. For fixed responses, each query has a single feature value, so a pointwise-smallest rule exists: the closure (10).

For completeness, the dual of (9) is

$$C_\varphi(h) = \max_{\lambda_x \geq 0} \sum_x \lambda_x h(x) \quad \text{s.t.} \quad \sum_x \lambda_x p_x(z) \leq d(z) \quad (\forall z \in \mathcal{Z}).$$

A constant charge $\max_x h(x)$ makes the primal feasible, and its objective is nonnegative, so finite linear-program duality gives attainment and equality. A dual feasible value above the burden cap certifies that this family cannot be protected within the cap.

Rewrite closure and attacker types. The substitutes R^* and costs c^* in (3) are the reflexive–transitive closure of a one-step rewriting relation with costs $c_1 \geq 0$, so c^* is a shortest-path cost and $c^*(x, x'') \leq c^*(x, x') + c^*(x', x'')$. Two cheap rewrites can undercut every single-step one, which is why s^e minimizes over the whole closure.

Protection is required separately for each declared type. Replacing the types by the envelope $\max_\theta g_\theta$ changes the target: with $\mathcal{X} = \{a, b\}$, $g_{\theta_1} = (1, 0)$ and $g_{\theta_2} = (0, 1)$, protecting only a meets the envelope target at $\varepsilon = 1/2$ while leaving type θ_2 fully served.

General protection targets. The proofs use a general form of (7). For each type let $\mathfrak{F}_\theta(\varepsilon)$ be the query sets whose protection meets that type's target, and assume it is upward closed: protecting additional queries preserves qualification. The additive target is

$$\mathfrak{F}_\theta^{\text{add}}(\varepsilon) = \{A \subseteq \mathcal{X} : g_\theta(A) \geq (1 - \varepsilon)V_\theta\}. \tag{12}$$

For a general target the frontier is $B^*_\varphi(\varepsilon) = \min\{B(s_{\mathbf{A}}) : A_\theta \in \mathfrak{F}_\theta(\varepsilon)\ \forall\theta\}$, attained whenever some family qualifies because $\mathcal{X}$ and Θ are finite. Theorem 1 prices each family exactly; choosing the family remains combinatorial, and for additive targets it contains a covering knapsack problem.

Feature portfolios. A linear portfolio $s_w(x, y) = \sum_j w_j z_j(x, y)$ uses nonnegative features and weights. Write $\bar{z}_j(x) = \mathbb{E}_{y|x}[z_j(x, y)]$ and $b_j = \sum_x D(x)\bar{z}_j(x)$. Its expected charge must meet the rewrite floor. On $\mathcal{X}_{\mathbf{A}} = \{x : h_{\mathbf{A}}(x) > 0\}$, its least burden solves

$$B^*_{\text{port}}(\mathbf{A}) = \min_{w \geq 0} \sum_j w_j b_j \quad \text{s.t.} \quad \sum_j w_j \bar{z}_j(x) \geq h_{\mathbf{A}}(x) \quad (\forall x \in \mathcal{X}_{\mathbf{A}}). \tag{13}$$

Proposition 3 (Feature-portfolio frontier)**.** *For a fixed protected family* $\mathbf{A}$*, program* (13) *gives the minimum-burden nonnegative linear portfolio. Adding a feature cannot increase this minimum. Without a price cap, the program is infeasible exactly when some* $x \in \mathcal{X}_{\mathbf{A}}$ *has* $\bar{z}_j(x) = 0$ *for every* j*. Its dual is*

$$\max_{\zeta \geq 0} \sum_{x \in \mathcal{X}_{\mathbf{A}}} \zeta_x h_{\mathbf{A}}(x) \quad \textit{s.t.} \quad \sum_{x \in \mathcal{X}_{\mathbf{A}}} \zeta_x \bar{z}_j(x) \leq b_j \quad (\forall j), \tag{14}$$

and any feasible ζ *whose value exceeds a burden budget certifies that no portfolio meets it.*

With feature library $\mathcal{L} = \{z_1, \ldots, z_J\}$ and blocked-value target $q = 1 - \varepsilon$, minimizing jointly over families gives the linear-portfolio frontier

$$B^*_{\mathcal{L}}(q) = \min_{\mathbf{A},\, w \geq 0} \sum_j w_j b_j \quad \text{s.t.} \quad g_\theta(A_\theta) \geq q V_\theta \ \ \forall \theta, \qquad \sum_j w_j \bar{z}_j \geq h_{\mathbf{A}} \text{ on } \mathcal{X}_{\mathbf{A}}. \tag{15}$$

A single score priced as $s(x, y) = \eta z_j(x, y)$ is the portfolio with one nonzero weight, so $B^*_{\mathcal{L}}(q)$ never exceeds the frontier of that rule.

Where the burden comes from.

Corollary 4 (Fixed-family burden decomposition)**.** *Let* $g_{\mathbf{A}}(x) = \max_\theta g_\theta(x) \mathbf{1}\{x \in A_\theta\}$ *be the extraction value of the protected queries themselves. For every protected family* $\mathbf{A}$*,*

$$B(s_{\mathbf{A}}) = \underbrace{\mathbb{E}_{x \sim D}[g_{\mathbf{A}}(x)]}_{\textit{value}} + \underbrace{\mathbb{E}_{x \sim D}[h_{\mathbf{A}}(x) - g_{\mathbf{A}}(x)]}_{\textit{rewriting}} + \underbrace{\mathbb{E}_{x \sim D}[\bar{s}_{\mathbf{A}}(x) - h_{\mathbf{A}}(x)]}_{\textit{feature ambiguity}}, \tag{16}$$

and all three terms are nonnegative.

The first term is the burden of the same family when value is fully observable and rewriting is impossible, so the ratio $B(s_{\mathbf{A}})/\mathbb{E}_{x \sim D}[g_{\mathbf{A}}(x)]$ is the fixed-family multiple quoted in Section 5.2. The optimal family itself can change when rewriting or observability changes. Let B^*_0, B^*_R, and $B^*_{R,\varphi}$ be the reoptimized frontiers with neither restriction, with rewriting only, and with both; Proposition 5 makes both increments in $B^*_{R,\varphi} = B^*_0 + (B^*_R - B^*_0) + (B^*_{R,\varphi} - B^*_R)$ nonnegative. On the known environment of Table 3, feature ambiguity is 36–42% of the burden across blocked-value targets $q \in [0.5, 0.9]$, against 41–49% for value and 12–21% for rewriting. It is 1.9–3.4 times the rewriting term and exceeds the value term at $q = 0.6$.

Changing the feature or the rewrites.

Proposition 5 (Comparative statics)**.** *Write* $B^*_\varphi(\varepsilon; c^*)$ *for the frontier under feature* φ *and rewrite costs* c^**, with* $c^* = +\infty$ *on unreachable pairs. If* $\varphi_1 = f \circ \varphi_2$ *for some map* f*, then* $B^*_{\varphi_2}(\varepsilon; c^*) \leq B^*_{\varphi_1}(\varepsilon; c^*)$*. If* $c^*_2 \leq c^*_1$ *pointwise, then* $B^*_\varphi(\varepsilon; c^*_2) \geq B^*_\varphi(\varepsilon; c^*_1)$*.*

A finer feature therefore never raises the frontier, and additional or cheaper rewrites never lower it.

Corollary 6 (Tail and group burden for fixed responses)**.** *When each query has a fixed response, choose* $s_{\mathbf{A}}$ *by* (10)*. Among rules protecting* $\mathbf{A}$*, it minimizes every burden measure that is nondecreasing in each realized surcharge, including the mean, worst mean over fixed groups, tail quantiles, and conditional value at risk. Only the choice of* $\mathbf{A}$ *must be reoptimized.*

For random responses, the same expectation constraints describe protection, but a different burden objective can select a different rule. Mean optimality alone does not imply tail or group optimality.

Uncertain rewrites.

Proposition 7 (Uncertain rewriting graph)**.** *Let* $\mathcal{U}$ *be a declared family of rewrite-cost maps, with* $+\infty$ *marking an unreachable rewrite, and write* $h^c_{\mathbf{A}}$ *for the rewrite floor under* $c \in \mathcal{U}$*. For scenario-specific families* $\mathbb{A} = \{\mathbf{A}^c \in \mathcal{A}_\varepsilon : c \in \mathcal{U}\}$*, let* $h_{\mathbb{A}} = \sup_{c \in \mathcal{U}} h^c_{\mathbf{A}^c}$*. With the response law and feature held fixed, the frontier robust to every* $c \in \mathcal{U}$ *is*

$$B^*_{\mathcal{U}}(\varepsilon) = \inf_{\mathbb{A}} C_\varphi(h_{\mathbb{A}}).$$

For a family $\mathbf{A}$ *fixed across scenarios, its robust minimum burden is* $C_\varphi(h^{\underline{c}}_{\mathbf{A}})$*, where* $\underline{c} = \inf_{c\in\mathcal{U}} c$ *and the floor is computed from these pairwise infima. For nominal costs* $\widehat{c}$*, true costs* c*, and* $e_{\mathbf{A}} = (h^{c}_{\mathbf{A}} - h^{\widehat{c}}_{\mathbf{A}})_+$*,*

$$C_\varphi(h^{c}_{\mathbf{A}}) \le C_\varphi(h^{\widehat{c}}_{\mathbf{A}}) + C_\varphi(e_{\mathbf{A}}). \tag{17}$$

Specifically, adding a rule for the error floor to a nominal protecting rule repairs protection. The extra minimum burden is at most κ *if every truly reachable pair is declared and* $c(x,x') \ge \widehat{c}(x,x') - \kappa$ *on all such pairs, for* $\kappa \ge 0$*.*

An undeclared edge lies outside this bound: every protected query that reaches its source may gain a cheaper path to its target and to everything reachable from there. A discovered edge is therefore added to the declared graph and the rule recomputed.

A.2 Capability-based protection

Theorem 1 requires only that qualifying families be upward closed, so it covers protection targets defined by downstream capability.

Corollary 8 (Capability targets)**.** *Let* $C_\theta(S)$ *be the capability obtainable from the answers to a query set* S*, averaged over responses and training randomness under a specified learning protocol, and let* $\overline{C}_\theta(S) = \max_{S'\subseteq S} C_\theta(S')$ *be its monotone envelope; the envelope is used because an attacker may discard answers, and measured capability need not be monotone in* S*. Then* $\mathfrak{F}^C_\theta(\varepsilon) = \{A : \overline{C}_\theta(\mathcal{X}\setminus A) \le \varepsilon\,\overline{C}_\theta(\mathcal{X})\}$ *is upward closed, and Theorem 1 gives its exact frontier; acquisition and rewriting still follow the per-query model of Section 3.3. If* $C_\theta(S) = u(g_\theta(S))$ *for a continuous, strictly increasing, concave* u *with* $u(0) = 0$*, this target equals the additive target with blocked value* $Q_{\mathrm{cap}} = V_\theta - u^{-1}(\varepsilon u(V_\theta)) \ge (1-\varepsilon)V_\theta$*, so the capability frontier is at least the additive one.*

Whether a query-level result predicts capability depends on how capability grows with acquired value.

Corollary 9 (Value–capability transfer)**.** *For* $V_\theta > 0$*,* $\overline{C}_\theta(\mathcal{X}) > 0$*, and* $\beta \in [0,1]$*, let*

$$\Psi_\theta(\beta) = \max\Big\{\overline{C}_\theta(S)/\overline{C}_\theta(\mathcal{X}) :\ S\subseteq\mathcal{X},\ g_\theta(S) \le \beta V_\theta\Big\}.$$

Any rule under which the acquired set S *satisfies* $g_\theta(S) \le \beta V_\theta$ *leaves remaining capability at most* $\Psi_\theta(\beta)$*;* Ψ_θ *is nondecreasing with* $\Psi_\theta(1) = 1$*, and* $\Psi_\theta(\beta) \le u(\beta V_\theta)/u(V_\theta)$ *when* $C_\theta(S) = u(g_\theta(S))$ *as above.*

A task that saturates after few examples has Ψ_θ rising steeply near 0 and needs near-total value suppression; a task requiring diverse coverage has Ψ_θ rising gradually. Training students on value-ranked subsets probes one family of subsets and learners, so it can demonstrate saturation but not upper-bound Ψ_θ. When capability is the target, we evaluate it directly (Section 5.4).

A.3 Finite-sample validation

The audit evaluates the sets that the fitted rules actually protect, which can differ from the families used to build them. For candidate rule k and type θ, let $Q_{\theta k} = V_\theta - g_\theta(P_\theta(s_k))$ be the true blocked value and $Q_{\mathrm{tgt},\theta} = (1-\varepsilon)V_\theta$ the target, with V_θ fixed independently of the audit. The candidate rules may come from the response-feature program, its fixed-response closure, a linear portfolio, or any other fixed class, but must be fixed before the audit samples are observed.

Proposition 10 (Finite-sample audit guarantee)**.** *Let* $s_1,\dots,s_K$ *be candidate rules chosen independently of the audit samples, with* $0 \le s_k(x,y) \le S_{\max}$*. Using* n_D *i.i.d. benign pairs drawn as* $x\sim D$*,* $y\sim\pi(\cdot\mid x)$ *and* n_R *i.i.d. attacker-query samples per declared type, estimate* $B(s_k)$ *by the sample mean* $\widehat{B}(s_k)$ *and* $Q_{\theta k}$ *by the mean* $\widehat{Q}_{\theta k}$ *of independent, unbiased contributions in* $[0, V_{\max}]$ *(Appendix A.4 gives the importance-weighted estimator). With probability at least* $1-\delta$*, simultaneously for all* k *and* θ*,*

$$|\widehat{B}(s_k) - B(s_k)| \le \Delta_B := S_{\max}\sqrt{\tfrac{\ln(4K/\delta)}{2n_D}}, \qquad |\widehat{Q}_{\theta k} - Q_{\theta k}| \le \Delta_P := V_{\max}\sqrt{\tfrac{\ln(4K|\Theta|/\delta)}{2n_R}}.$$

Hence any candidate rule with $\widehat{Q}_{\theta k} - \Delta_P \ge Q_{\mathrm{tgt},\theta}$ *for every* θ *and* $\widehat{B}(s_k) + \Delta_B \le B_{\max}$ *meets every protection target and the burden limit on that event.*

The blocked-value estimator assumes that effective acquisition costs, and hence which queries are protected, are computed exactly under the declared response law. A single sampled charge cannot replace an expected charge inside a protection indicator. If these expectations are estimated, one valid alternative is to construct simultaneous lower bounds on them using separate response samples, certify only memberships implied by those bounds, and allocate additional failure probability to that step. This yields conservative protection bounds; the unbiased estimator and the candidate-menu regret guarantee stated here apply only when these expectations are exact.

With $U_k = \widehat{B}(s_k) + \Delta_B$ and $\ell_{k\theta} = \widehat{Q}_{\theta k} - \Delta_P$, FGP forms $\mathcal{F} = \{k : U_k \le B_{\max},\ \ell_{k\theta} \ge Q_{\mathrm{tgt},\theta}\ \forall \theta\}$ and deploys $\widehat{k} \in \arg\min_{k\in\mathcal{F}} U_k$, abstaining if $\mathcal{F}$ is empty. The single-candidate procedure is the case $K = 1$. Simultaneous coverage makes the choice among qualifying candidate rules valid as long as the rules stay fixed during the audit. Let $\mathcal{K}_{\mathrm{mar}} = \{k : Q_{\theta k} \ge Q_{\mathrm{tgt},\theta} + 2\Delta_P\ \forall \theta,\ B(s_k) \le B_{\max} - 2\Delta_B\}$. If $\mathcal{K}_{\mathrm{mar}}$ is nonempty, then on the same event FGP deploys and

$$B(s_{\widehat{k}}) \le \min_{k\in\mathcal{K}_{\mathrm{mar}}} B(s_k) + 2\Delta_B. \tag{18}$$

For one type with total value V, if a retained candidate rule attains the linear-portfolio frontier at the raised target $q + 2\Delta_P/V \le 1$ with burden at most $B_{\max} - 2\Delta_B$, then (18) places the deployed burden within $2\Delta_B + B^*_{\mathcal{L}}(q + 2\Delta_P/V) - B^*_{\mathcal{L}}(q)$ of $B^*_{\mathcal{L}}(q)$. On a finite query set this frontier can jump at q, so the second term need not vanish as the samples grow. Appendix A.4 proves Proposition 10 together with Proposition 2.

A.4 Proofs

Proof of Theorem 1. Fix $\mathbf{A}$. By (3), protecting it is equivalent to $\bar{s}(x') + c^*(x, x') \ge g_\theta(x)$ for every θ, $x \in A_\theta$, and $x' \in R^*(x)$. Since charges are nonnegative, this is exactly $\bar{s}(x') \ge h_{\mathbf{A}}(x')$ for all x'. Substituting $s(x, y) = a(\varphi(x, y))$ gives $\bar{s}(x) = \sum_z p_x(z) a_z$ and $B(s) = \sum_z d(z) a_z$. Thus (9) is precisely the fixed-family optimization. It is feasible (a constant charge $\max_{\theta,x} g_\theta(x)$ suffices), finite dimensional, and bounded below by zero, so it attains an optimum.

Given any ε-deterring rule s, choose $A_\theta = \mathcal{X} \setminus P_\theta(s)$ for each type. This family qualifies, and $B(s) \ge C_\varphi(h_{\mathbf{A}})$, hence at least the minimum over qualifying families. Conversely an optimizer for any qualifying family protects that family and is ε-deterring. There are finitely many families, giving the stated equality and attainment. For a general upward-closed target the same argument applies: the induced protected sets contain the proposed ones and therefore still qualify.

For fixed responses y_x, the constraints are $a_{\varphi(x,y_x)} \ge h_{\mathbf{A}}(x)$. For each feature value their smallest feasible charge is the largest floor among queries with that feature, proving (10). □

Proof of Proposition 3. Program (13) is a finite linear program with one covering constraint per $x \in \mathcal{X}_{\mathbf{A}}$. If some $x \in \mathcal{X}_{\mathbf{A}}$ has $\bar{z}_j(x) = 0$ for all j, its constraint reads $0 \ge h_{\mathbf{A}}(x) > 0$ and the program is infeasible. Otherwise pick, for each x, an index $j(x)$ with $\bar{z}_{j(x)}(x) > 0$ and set $\bar{w}_j = \max\{h_{\mathbf{A}}(x)/\bar{z}_j(x) : j(x) = j\}$, or 0 if no x selects j; each constraint is then met by its own term. The objective is nonnegative on the feasible set, so a feasible program attains its optimum and strong duality holds with the dual (14); weak duality makes any feasible dual value a lower bound on every portfolio's burden. Appending a feature adds a variable and no constraint, and $(w, 0)$ is feasible with the same objective, so the minimum cannot increase. Figure 3 shows menus for which it does not decrease. □

Proof of Propositions 10 and 2. For type θ, draw $x_{\theta i} \overset{\text{iid}}{\sim} \rho_\theta$ with $\rho_\theta(x) > 0$ whenever $g_\theta(x) > 0$, and set

$$Y^{(k)}_{\theta i} = \frac{g_\theta(x_{\theta i})}{\rho_\theta(x_{\theta i})} \mathbf{1}\{s^e_k(x_{\theta i}) \ge g_\theta(x_{\theta i})\}, \qquad \widehat{Q}_{\theta k} = \frac{1}{n_R} \sum_{i=1}^{n_R} Y^{(k)}_{\theta i},$$

where s^e_k is the effective cost (3) under s_k. Then $\mathbb{E}[Y^{(k)}_{\theta i}] = \sum_x g_\theta(x) \mathbf{1}\{s^e_k(x) \ge g_\theta(x)\} = Q_{\theta k}$; the proposition assumes $0 \le Y^{(k)}_{\theta i} \le V_{\max}$, and uniform sampling of a finite pool gives the population-size factor as the weight. For benign draws, each $s_k(x_i, y_i)/S_{\max} \in [0, 1]$, so Hoeffding's inequality gives $\mathbb{P}(|\widehat{B}(s_k) - B(s_k)| > \Delta_B) \le 2e^{-2n_D(\Delta_B/S_{\max})^2} = \delta/(2K)$; likewise each value deviation has probability at most $\delta/(2K|\Theta|)$. A union bound over the K burden and $K|\Theta|$ value events gives

the display. On that event the deployed rule satisfies $B(s_{\widehat{k}}) \le U_{\widehat{k}} \le B_{\max}$ and $Q_{\theta\widehat{k}} \ge \ell_{\widehat{k}\theta} \ge Q_{\mathrm{tgt},\theta}$ for every θ. If $k \in \mathcal{K}_{\mathrm{mar}}$, then $\ell_{k\theta} \ge Q_{\theta k} - 2\Delta_P \ge Q_{\mathrm{tgt},\theta}$ and $U_k \le B(s_k) + 2\Delta_B \le B_{\max}$, so $k \in \mathcal{F}$ and FGP deploys; moreover $B(s_{\widehat{k}}) \le U_{\widehat{k}} \le U_k \le B(s_k) + 2\Delta_B$, and minimizing over $k \in \mathcal{K}_{\mathrm{mar}}$ gives (18). □

Proof of Proposition 5. If $\varphi_1 = f \circ \varphi_2$, every rule $a \circ \varphi_1$ equals $(a \circ f) \circ \varphi_2$, so the minimization under φ_2 has at least the same feasible rules. If $c_2^* \le c_1^*$, every term $[g_\theta(x) - c^*(x, x')]_+$ in (8) weakly increases, so $h_{\mathbf{A}}$ weakly increases for every family. The feasible set of (9) shrinks, so its minimum burden cannot decrease, nor can the minimum over families in Theorem 1. □

Proof of Corollary 6. For fixed responses, every rule protecting $\mathbf{A}$ satisfies $s(x, y_x) \ge s_{\mathbf{A}}(x, y_x)$ by (10), so any burden measure that cannot decrease when a surcharge increases is at least its value at $s_{\mathbf{A}}$. □

Proof of Corollary 4. The identity follows by adding and subtracting $h_{\mathbf{A}}$ and $g_{\mathbf{A}}$ inside $B(s_{\mathbf{A}}) = \mathbb{E}_{x\sim D}[\bar{s}_{\mathbf{A}}(x)]$. For $x \in A_\theta$, the pair (θ, x) with $x' = x$, $c^*(x, x) = 0$ enters (8), so $h_{\mathbf{A}}(x) \ge g_\theta(x)$; hence $h_{\mathbf{A}} \ge g_{\mathbf{A}}$, trivially so where $g_{\mathbf{A}} = 0$. And $\bar{s}_{\mathbf{A}} \ge h_{\mathbf{A}}$ by feasibility of (9). □

Proof of Proposition 7. If s deters under every c, its induced families $\mathbf{A}^c$ qualify and $\bar{s} \ge h^c_{\mathbf{A}^c}$ for every scenario. Thus $\bar{s} \ge h_{\mathbb{A}}$ and $B(s) \ge C_\varphi(h_{\mathbb{A}})$. Conversely a rule meeting the supremum floor protects $\mathbf{A}^c$ under each c. Infimizing over the scenario-specific families proves the robust formula. All floors are bounded by $\max_{\theta,x} g_\theta(x)$. For a common family, the finite maximum over acquisitions commutes with the scenario supremum, giving $\sup_c h^c_{\mathbf{A}} = h^{\bar{c}}_{\mathbf{A}}$.

The functional C_φ is monotone by feasible-set inclusion and subadditive because adding feasible response-level rules adds both their expected charges and burdens. Since $h^c_{\mathbf{A}} \le h^{\widehat{c}}_{\mathbf{A}} + e_{\mathbf{A}}$, these properties give (17). Under the stated bound on all reachable-pair costs, $e_{\mathbf{A}} \le \kappa$; the constant response-level surcharge κ covers this floor at burden κ. The assumption must hold for shortest-path costs, because per-edge errors can accumulate along a multi-edge path. □

Proof of Corollaries 8 and 9. If $A \subseteq A'$, then $\mathcal{X} \setminus A' \subseteq \mathcal{X} \setminus A$ and monotonicity of $\overline{C}_\theta$ preserves the capability constraint, so $\mathfrak{F}^C_\theta$ is upward closed. When $C_\theta = u \circ g_\theta$ with u increasing, $\overline{C}_\theta = C_\theta$ and the constraint reads $g_\theta(\mathcal{X} \setminus A) \le u^{-1}(\varepsilon u(V_\theta))$, that is, $g_\theta(A) \ge Q_{\mathrm{cap}}$; concavity and $u(0) = 0$ give $u(\varepsilon V_\theta) \ge \varepsilon u(V_\theta)$ and hence $Q_{\mathrm{cap}} \ge (1 - \varepsilon)V_\theta$. The transfer bound is the definition of Ψ_θ, and in the concave case $\overline{C}_\theta(S) = u(g_\theta(S)) \le u(\beta V_\theta)$ whenever $g_\theta(S) \le \beta V_\theta$. □

B Experimental Protocols

B.1 Workloads and shared protocol

Table 2 summarizes the workloads, and Appendix B.2 defines the features, including the density model behind demand rarity. Extraction values are in normalized units, with no base price. We train LLM students by supervised fine-tuning with Low-Rank Adaptation (Hu et al., 2022), using rank 16, scaling 32, learning rate 2×10^{-4}, and 3 epochs.

We write a portfolio rule as $\eta\, s_w$: its *direction* w gives the relative feature weights, and its *intensity* $\eta \ge 0$ sets the overall surcharge scale. FGP searches nonnegative directions on a finite grid that includes the single-feature endpoints. For a fixed direction, sorting queries by the ratio of extraction value to surcharge and accumulating value gives the least intensity that meets a feasible target, so FGP finds the target-reaching intensity exactly within each direction. In the fixed-family experiment, we solve the covering LP (13).

Every compared rule shares the benign pool, attacker pool, extraction-value proxy, query budget, per-query price cap, student architecture, training budget, and held-out evaluation. We set the cap in each workload's value units, since a cap below a typical query's value cannot deter acquisition at any intensity. We retrain students once per distinct affordable query set. Query-level audits treat the held-out attacker pool as the declared population: full-pool results use its exact weighted sum,

Monte-Carlo audits sample it uniformly with the population-size factor, and we fix V_θ from the pool before evaluating candidate rules, as in Proposition 10.

We fit the logistic attacker-versus-benign score on a 4% within-seed selection subsample where the benign and attacker pools share a corpus, and we omit it on GSM8K, whose pools come from different corpora that a classifier separates at zero benign cost. Once a score map is fixed, every rule is request-local, so splitting the same queries across accounts leaves their total charge unchanged.

We call a workload *separable* when the available features distinguish most high-value attacker queries from benign demand, and *overlapping* when they do not.

Domain / regime	Protocol summary
Vision, separable	ResNet-34 (He et al., 2016) → ResNet-18; in-distribution CIFAR-10 (Krizhevsky, 2009) demand, out-of-distribution ActiveThief/Knockoff Nets pool; rarity and output signature; test accuracy; 3–10 seeds at the query level, 50 in the capability grid.
Vision, overlap	ResNet-34 → ResNet-18; in-distribution demand and pool; rarity and output signature; test accuracy; 5–10 seeds.
Text rewriting	`distilbert-base-uncased` (Sanh et al., 2019) fine-tuned on AG News (Zhang et al., 2015) → a second `distilbert-base-uncased` distilled on the victim's soft outputs; news demand and paraphrases; lexical rarity and output signature; blocked extraction value and test accuracy; 3–5 seeds.
LLM single skill	`Qwen/Qwen2.5-3B-Instruct` (Yang et al., 2025) → the base `Qwen/Qwen2.5-0.5B` checkpoint; Alpaca (Taori et al., 2023) demand, GSM8K (Cobbe et al., 2021) pool; rarity and output signature; accuracy; 6 seeds.
LLM broad	`Qwen/Qwen2.5-3B-Instruct` → base `Qwen/Qwen2.5-0.5B`; dense-to-diverse Alpaca; rarity and output signature; held-out negative log-likelihood (NLL); 16 seeds.
LLM scale-up	`Qwen/Qwen2.5-7B-Instruct` → base `Qwen/Qwen2.5-0.5B`; same broad protocol; held-out NLL; 6 seeds.
Adaptive vision	ResNet-34 → ResNet-18; pool re-estimated each round; rarity and output signature; test accuracy; 5 seeds.
Recommender API	SASRec (Kang & McAuley, 2018) → SASRec; MovieLens-1M (Harper & Konstan, 2015); victim 200 epochs; popular-session demand, long-tail pool; rarity and output signature; NDCG and Recall; 20 seeds in the extraction sweep, 50 in the capability grid.
Recommender API, second corpus	SASRec → SASRec; Amazon Beauty 5-core (2014 release; 12,101 items) (McAuley et al., 2015); otherwise as MovieLens; 50 seeds.
Retrieval API	`sentence-transformers/all-MiniLM-L6-v2` (Reimers & Gurevych, 2019) (384-d embeddings) → a 2-layer BERT encoder trained from scratch (hidden size 128, 2 heads, victim tokenizer, 384-d projection, 4.4M parameters); MS MARCO (Nguyen et al., 2016) demand, diverse tail; rarity and output signature; NDCG, Recall, MRR, and fidelity; 16 seeds.

Table 2: **Workloads.** Output signatures use the signed, standardized entropies of Appendix B.2; seed ranges span the figures that use each workload.

B.2 Features and extraction values

All logarithms are natural. Write $H(v) = -\sum_j v_j \log v_j$ for the Shannon entropy of a probability vector and $Z_D[f](x,y) = (f(x,y) - \mu_f)/(\sigma_f + 10^{-8})$ for standardization on benign query–response records.

Embedding rarity. We fit PCA, coordinate standardization, and a diagonal-covariance Gaussian mixture to benign embeddings, and write $\widehat{d}(x)$ for the mixture density at the transformed embedding of x. The feature $r_D(x,y) = [Z_D[-\log \widehat{d}](x,y)]_+$ ignores the response. This is a density score in representation space, not the probability mass $D(x)$ of a literal query, and its normalization is part of the declared feature.

Lexical rarity. On AG News, we count lowercase alphabetic tokens in the benign texts. With token counts $n(w)$, total count N_{tok}, and vocabulary size $|\mathcal{V}|$, the raw score is the mean token surprisal $|x|^{-1} \sum_{w \in x} \log[(N_{\mathrm{tok}} + |\mathcal{V}|)/(n(w)+1)]$, with $\log(N_{\mathrm{tok}} + |\mathcal{V}|)$ for an empty sequence; the feature is its standardized positive part.

Output signature. The feature is $o(x,y) = [\xi Z_D[e](x,y)]_+$, with raw statistic e and sign ξ as follows.

- CIFAR-10 and AG News: $e(x,y) = H(v(y))$ for the victim's class-probability vector $v(y)$ in the response record; $\xi = +1$.
- MovieLens and Beauty: $e(x,y) = H(\mathrm{softmax}(\ell(y)))$ for the victim's logits $\ell(y)$ in the record over the full item catalogue, computed before the top-ranked items are returned; $\xi = +1$.

- Broad instruction and GSM8K: $e(x, y) = T^{-1} \sum_{t=1}^{T} H(v_t(y))$, where v_t is the softmax of the victim's score vector at generation step t; $\xi = +1$. Generation is greedy in batches of 16. The cached scores average all returned steps without a per-sequence end-of-sequence mask, so T is the batch's generation length; a live implementation should mask each sequence at its own stopping point.
- MS MARCO: for the unit-normalized returned embedding y and 512 fixed unit anchors a_j drawn from the query pool before the demand–attacker split, $v_j(y) \propto \exp(10\, y^\top a_j)$ and $e(x, y) = H(v(y))$; $\xi = -1$, so concentrated responses are priced higher.

Provider-side embeddings, class probabilities, and catalogue logits are available for pricing even when the returned response exposes only part of them. Broad instruction generates at most 256 new tokens for both demand and attacker prompts. The cached GSM8K runs generate at most 256 on attacker prompts but only 8 on benign prompts, so GSM8K output-signature results reflect this preprocessing difference and serve only as a saturation and abstention case.

Extraction-value proxies. The proxy g_θ is the surrogate student's response entropy on CIFAR-10, one unit per query on MovieLens, Beauty, and MS MARCO, raw victim-response entropy in the AG News rewriting study and the LLM workloads, and a surrogate-student proxy in the AG News capability experiment. A priced feature doubles as the proxy only in the language settings, where o is the standardized positive part of the same victim entropy. There, blocked-value results test whether entropy-based pricing survives rewriting, and retrained accuracy or held-out NLL decides capability targets. We scored candidate proxies by how well the value of a purchased set predicts the capability it produces. On images the surrogate student's entropy reaches Spearman 0.880 against 0.861 for counting queries. On recommendation the uniform count reaches 0.791 against 0.707 for the victim's ranking entropy, and 0.790 against 0.746 for a surrogate student in a separate 128-point comparison. Input rarity scores 0.807 on recommendation, but it is a priced feature, so we exclude it; for the same reason we exclude retrieval's anchor entropy.

B.3 THE FGP PROCEDURE

Algorithm 1 Frontier-Guided Pricing: protected families and independent validation

Require: fixed feature φ; declared rewrite family and attacker types; selection samples $\mathcal{D}_{\text{sel}}$ and independent audit samples $\mathcal{D}_{\text{aud}}$; $\varepsilon, B_{\max}, \delta$
Ensure: deployed pricing rule with audit bounds, or abstain
1: Estimate values, demand, and conditional feature probabilities on $\mathcal{D}_{\text{sel}}$
2: Propose protected families $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(K)}$ from these estimates
3: **for** $k = 1, \ldots, K$ **do**
4: Build $\widehat{h}_k$ by (8); solve (9) for s_k ▷ use closure (10) for fixed responses ▷ or solve (13) in the linear subclass
5: Estimate protection and burden on $\mathcal{D}_{\text{sel}}$
6: **end for**
7: Retain candidate rules using the selection estimates and fix every retained rule ▷ retain only the best for single-candidate validation
8: **if** no rule is retained **then**
9: **return** abstain
10: **end if**
11: Audit the retained rules on $\mathcal{D}_{\text{aud}}$ with simultaneous confidence $1 - \delta$: burden UCBs U_k and protection LCBs $\ell_{k\theta}$
12: $\mathcal{F} \leftarrow \{k : U_k \le B_{\max},\ \ell_{k\theta} \ge (1 - \varepsilon) V_\theta\ \forall \theta\}$
13: **if** $\mathcal{F} = \varnothing$ **then**
14: **return** abstain
15: **end if**
16: $\widehat{k} \leftarrow \arg\min_{k \in \mathcal{F}} U_k$
17: **return** $s_{\widehat{k}}$ and its audit bounds

For a finite query set, lines 1–4 of Algorithm 1 can propose families from acquisitions supported by the selection samples, propagate their estimated values to every declared rewrite, and solve the

Table 3: **Fixed-family attribution of benign burden** by Corollary 4 on the known environment. The last column is the family's burden as a multiple of its value term, its burden without rewriting or feature restriction.

target q	value	rewriting	feature	total	$\times$ naive
0.50	0.0100	0.0041	0.0078	0.0220	2.19
0.60	0.0126	0.0053	0.0128	0.0307	2.43
0.70	0.0157	0.0080	0.0149	0.0385	2.45
0.80	0.0240	0.0061	0.0192	0.0493	2.05
0.90	0.0344	0.0086	0.0288	0.0718	2.09

covering program with estimated feature probabilities, or take the closure for fixed responses. Each fitted rule needs a surcharge for every possible feature value, including values unseen in selection, and this extension must be fixed before the audit. The audit evaluates this completed rule on its actual profitable and protected sets, under the response-law conditions of Appendix A.3. When FGP compares several feature maps, it fixes each on the selection samples, and all resulting rules enter the same simultaneous audit.

C Additional Results

C.1 Protection cost and feature choice

Known environments. When D, g_θ, the rewrite graph, and the features are known, Theorem 1 reduces the frontier to a finite family search. On CIFAR-derived environments with these objects known and responses fixed, the floor-and-closure construction matches an independently solved minimum-burden mixed-integer linear program to 2×10^{-11}. Table 3 attributes the burden of the selected family by Corollary 4 in one such environment, with 40 query states, 5 feature values, and two rewrites of cost 0.55 per state.

We then estimate the frontier from finite benign and attacker-query samples. The design covers six environments, which vary the clustering seed, rewrite out-degree, and feature coarseness, with 16 targets, eight audit sizes from $n = 100$ to 20,000, and 30 seeds, for $1{,}440$ estimated frontiers (Figure 3). In the representative environment, pricing each target's optimal family through a coarser feature menu raises its burden at $q = 0.9$ from 0.013 to 0.033 (Figure 3(a)). The worst-case error of the plug-in estimate over targets falls from 0.0022 to 0.0001, at a fitted rate $n^{-0.53}$. At small n the plug-in frontier lies below the truth, because surcharges on query states missing from the benign sample appear to cost nothing. The certified frontier adds an attacker-query margin to the target and a benign margin to the burden. It upper-bounds the population frontier at every target in $1{,}439$ of the $1{,}440$ runs; the exception misses at one target at $n = 10{,}000$. The deploy-or-abstain decision at caps placed near the frontier is correct in 88.8% of cases at $n = 100$, 96.5% at $n = 500$, and 100% at $n = 20{,}000$.

Feature choice across targets and workloads. Table 4 shows that the cheapest rule on CIFAR-10 depends on the target and the declared menu. The portfolio weights also matter. On 36 cached vision partitions at target 0.90, with the target-reaching intensity recomputed on each held-out half, the selection-best single feature needs held-out burden 1.620 ± 0.026, normalized equal weighting 2.468 ± 0.046, and the optimized portfolio 1.613 ± 0.027. Optimization here matches the best single feature and improves substantially on equal weighting. On MovieLens, only the portfolio reaches the tested targets from $q = 0.55$ upward (Figure 4). Figures 4 and 5 give the query-value frontiers of all five workloads.

C.2 Transfer and finite-sample audits

Transfer of a fixed-family portfolio. In the fixed-family experiment, we fit the covering LP on ActiveThief-entropy, protecting the 90% highest-value selection queries by count; the fitted weights on rarity and output signature are $(8.86, 0.78)$. We measure protection as blocked value on held-out strategies and adjust each rule's intensity to a matched burden of 1.70 on every evaluation pool. The

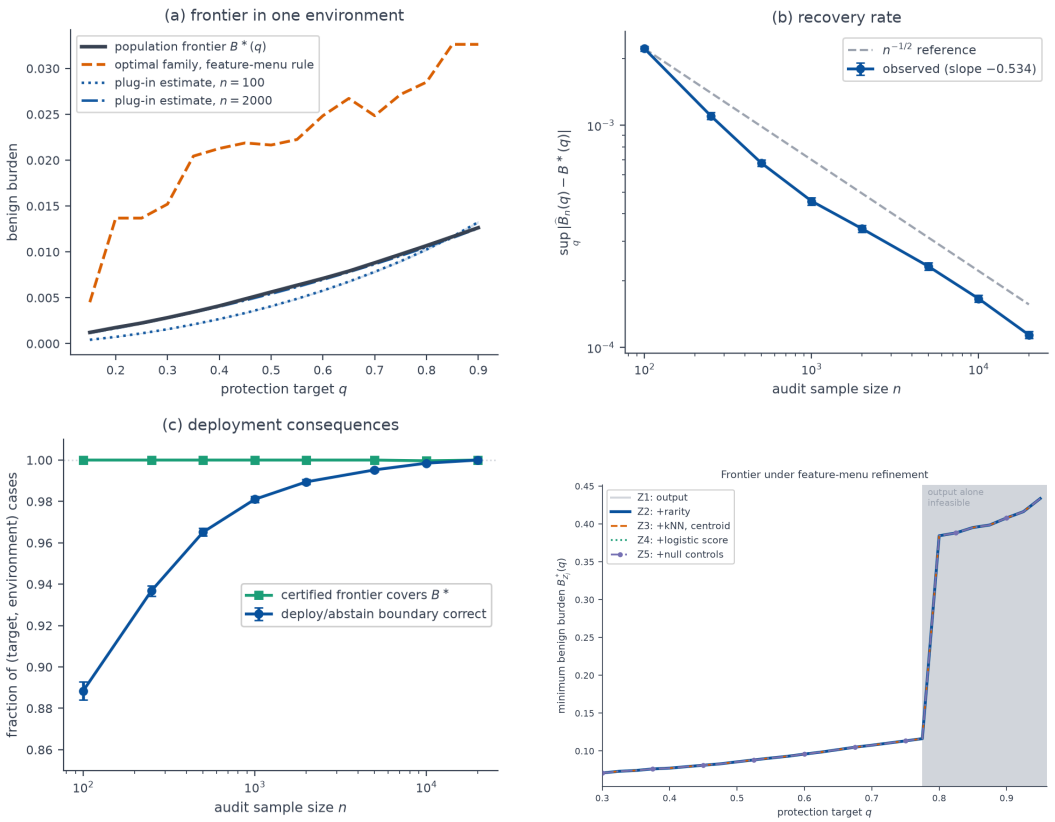


Figure 3: **Frontier recovery and menu refinement. (a–c)** On six CIFAR-derived environments with a computable frontier (1,440 estimates), the certified frontier upper-bounds the truth at every target in all but one run, and the deploy-or-abstain decision is correct in 100% of cases at $n = 20{,}000$; the dashed curve in (a) prices each target's optimal family through a coarser feature menu. **(d)** Enlarging a declared menu never raises its frontier; adding rarity restores high targets, while the kNN, centroid, logistic-score, and null-control additions leave it unchanged.

Table 4: **CIFAR-10 selections.** The same selection procedure applied to different targets and declared menus. Blocked-value rows report the burden of Figure 4; the capability row reports held-out normalized remaining capability.

Target	Declared features	Selected	Value
blocked value, 30%	rarity, output	output signature	0.19
blocked value, 80%	rarity, output	portfolio	1.00
blocked value, 30%	rarity, output, logistic	logistic score	0.18
blocked value, 80%	rarity, output, logistic	portfolio	0.80
capability, 0.8	rarity, output, logistic	logistic score	0.739

portfolio blocks all value on both unseen strategies, against 0.998 for the ex-post best single feature (Table 5).

Expanding the feature menu. In a separate experiment, we fit every feature on ActiveThief-entropy and select a rule for target $q = 0.90$ under $B_{\max} = 1.2$. We then evaluate its fixed intensity on k-center and Knockoff Nets against a mean held-out protection threshold of 0.80. Starting from the output signature alone, the menu adds, in order, rarity, kNN sparsity (mean distance to the 10 nearest benign selection neighbors) with distance to the benign selection centroid, two null controls (Gaussian noise and permuted rarity), and a logistic score fit on 800 selection points. Adding rarity turns abstention into a qualifying rule, and further additions leave the choice unchanged; the two null

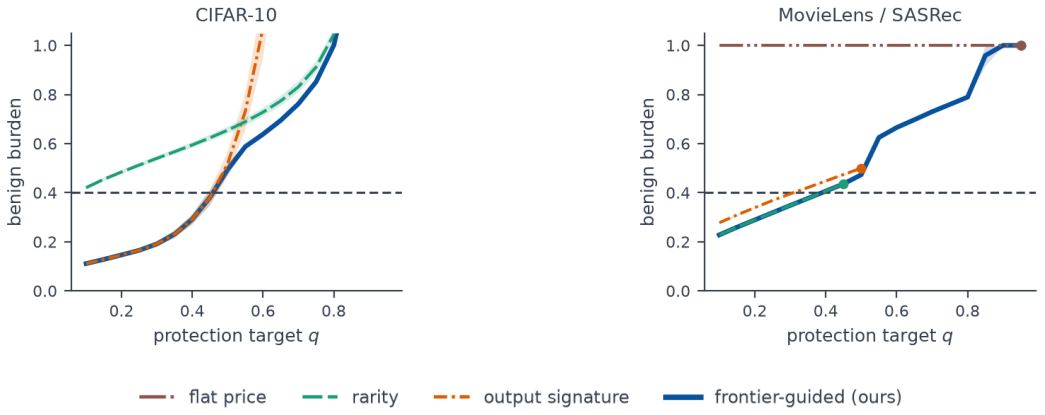


Figure 4: **Query-value protection and its benign cost** on CIFAR-10 and MovieLens. Least benign burden at which each rule meets the target plus a Hoeffding protection margin ($\delta = 0.05$); four seeds, bands ± 1 SEM, dashed line $B_{\max} = 0.4$. The cheapest direction changes with the workload and target.

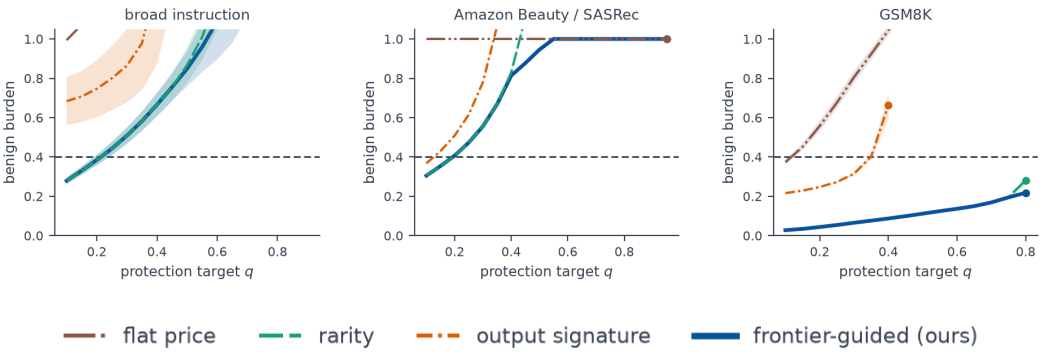


Figure 5: **Query-value frontiers on broad instruction, Beauty, and GSM8K.** Conventions as in Figure 4. Above the plotted range on broad instruction, the portfolio needs burden 1.54 at $q = 0.75$, 87% less than rarity (12.0; the output signature is infeasible), and for $q \in [0.80, 0.90]$ neither single feature meets the target on every seed. GSM8K meets the highest blocked-value target within the cap, $q = 0.80$; its capability cells are excluded from policy comparisons because the retrained student saturates (Appendix C.3).

Table 5: **Transfer of a fixed-family portfolio.** Weights learned on ActiveThief-entropy. Entries are blocked-value fractions on held-out strategies at matched benign burden 1.70. The oracle picks the best single feature after evaluation.

Held-out strategy	Rarity	Output	Portfolio	Oracle
k-center	1.000	0.990	1.000	1.000
Knockoff Nets	0.998	0.964	1.000	0.998
Worst case	0.998	0.964	1.000	0.998

controls receive at most 1.9% of the auxiliary LP weight (Table 6). Figure 3 shows the same pattern for the frontier at every target, in a nesting that adds the logistic score before the null controls. The logistic score falls from 0.757 selection protection to 0.622 held-out at burden 0.4, against 0.877 to 0.818 for rarity, and is not selected.

Table 6: **Expanding the declared feature menu.** Selected feature, selection-estimated burden, mean held-out protection, and the share of auxiliary LP weight on null controls.

Menu	Choice	Burden	Protection	Null wt.
output only	abstain	—	—	0.00
+ rarity	rarity	0.408	0.855	0.00
+ kNN, centroid	rarity	0.408	0.855	0.00
+ noise, permuted rarity	rarity	0.408	0.855	0.019
+ logistic	rarity	0.408	0.855	0.019

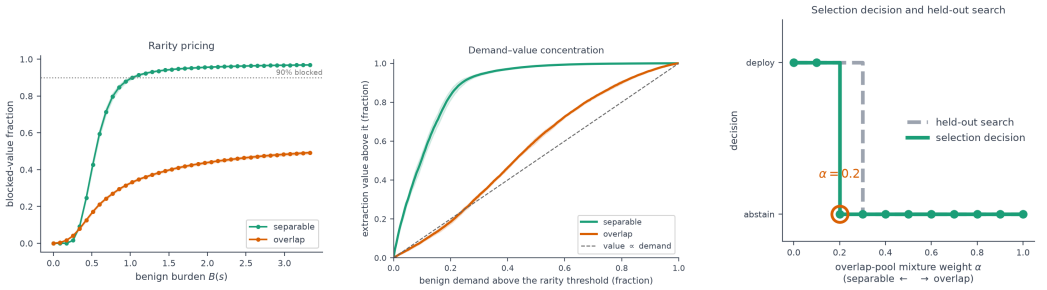


Figure 6: **Alignment and the selection decision.** CIFAR, 5 seeds; bands ± 1 SEM. **(a)** When extraction value lies on rare queries, rarity pricing reaches 90% blocked value at burden about 1.0; when value overlaps demand, it plateaus near 50%. **(b)** The share of extraction value above a rarity threshold against the share of benign demand above it; the diagonal marks value proportional to demand. **(c)** The 11-level overlap sweep selects two rules whose held-out protection exceeds 0.90 and otherwise abstains, once conservatively at $\alpha = 0.2$.

Predicting held-out cost. Across 50 feature–target–attack cases, the selection-predicted minimum burden tracks the held-out one. Among the 45 cases with finite burdens, Spearman correlation is 0.973 and mean absolute error is 0.024. The deploy-or-abstain action matches in 49 of 50 cases, and the feature FGP selects is the best one on both held-out strategies (Figure 12(d)).

Partition and overlap diagnostics. The partition diagnostic uses target $q = 0.90$, $B_{\max} = 1.2$, and the fixed directions $\{r_D, o, r_D + 0.5o\}$ with 60 intensities in $[0.05, 8]$. It splits cached attack–seed units into disjoint halves and compares the choice made on one half with the best choice on the other. All 36 choices agree: 18 portfolios and 18 abstentions. We refit the intensity on each half, so the agreement concerns this choice. The overlap sweep raises the mixture weight α, the fraction of attacker queries drawn from the overlapping pool, from 0 to 1 in 11 steps under the same target. Its two selected rules reach held-out protection 0.931 and 0.936. At $\alpha = 0.2$ the selection side abstains while the held-out search still finds a qualifying rule, and every later point abstains on both sides (Figure 6(c)). This sweep measures benign cost as $\mathbb{E}_{x\sim D}[\min\{s(x, y_x), 1\}]$ for the cached response y_x but tests profitability against the uncapped s. The sweep therefore tracks the deploy-or-abstain transition under this convention, whereas the frontier (11) uses the uncapped burden. Figure 6(b) shows why the two regimes of panel (a) differ. Sweeping a rarity threshold, it plots the fraction of extraction value above the threshold against the fraction of benign demand above it, which is the demand that a surcharge on those queries would reach.

Fixed-menu audits. We audit a fixed menu of 160 rules 2,000 times per sample size with $n_D = n_R \in \{1, 2, 4, 8, 16\}$k, a 90% blocked-value target, and $\delta = 0.05$. Each audit deploys the rule with the smallest burden UCB among those whose protection LCB clears the target, with no burden cap (Figure 7). No deployed rule violates the target, and every reported burden bound covers the pool mean. With zero violations in 2,000 trials, the exact one-sided 95% upper bound on the joint probability of deployment and violation is 0.15%. Deployment rises from 0.09 at 2k to 1.00 at 8k. The mean excess burden of deployed rules over the best feasible rule falls from 1.01 at 2k to 0.22 at 16k; at 1k only 4 of the 2,000 audits deploy. With three attacker types and an 80% target for each, deployment reaches 0.995 at 2k with no violation, and the excess burden falls from 0.36 at

Table 7: **Capability cost of fixed policies, per workload.** Paired difference in normalized remaining capability at matched benign burden; positive means FGP left the attacker weaker. †Not deployable.

alternative	CIFAR	MovieLens	Beauty	broad instr.
worst fixed policy	+0.215	+0.228	+0.195	+0.260
flat price	+0.202	+0.218	+0.168	+0.196
random fixed policy	+0.099	+0.099	+0.062	+0.123
equal weighting	+0.057	+0.097	+0.103	+0.100
fixed output signature	+0.018	+0.105	−0.023	+0.248
fixed rarity	+0.119	−0.024	+0.001	−0.051
ex-post oracle†	−0.005	−0.031	−0.051	−0.053

2k to 0.20 at 10k. We estimate the surcharge ranges in this audit from the attacker pool. Applying Proposition 10 to a population also requires them to bound benign surcharges, for example through a declared cap.

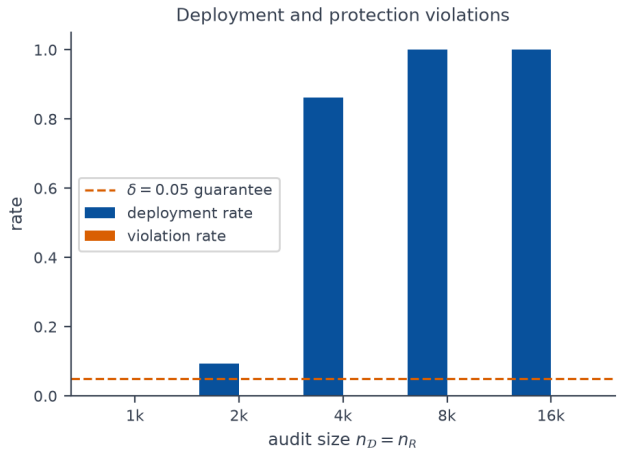


Figure 7: **Fixed-menu protection audit.** Over 2,000 trials per sample size, deployment becomes more frequent while observed protection violations remain zero.

C.3 Capability

Policy comparisons. For Figure 2, let $\Delta_{p,d}$ be policy p's estimated mean excess normalized remaining capability over query-value FGP in workload d. Its shortfall is $\max_d[\Delta_{p,d} - \min_{p' \in \mathcal{P}} \Delta_{p',d}]$, where $\mathcal{P}$ is the set of policies compared in that panel. The fixed policies are flat price, fixed rarity, fixed output signature, and equal weighting. The random fixed policy averages them, which is the expected result of committing to one uniformly at random. In Table 7, the worst fixed policy and the ex-post oracle take the worst and the best of them in each cell. Fixed-policy differences average paired seed–burden cells. Capability-based selection chooses the best fixed feature on the selection half of the seeds at each burden, and we score it on the held-out half. Error bars give one SEM of the paired differences at the worst workload. This comparison averages over operating burdens; we test the capability targets below. Table 7 and Figure 8 give the per-workload differences. On images the output signature trails FGP by 0.018 and rarity by 0.119; on broad instruction the output signature trails by 0.248. No fixed feature is best on every workload. The comparison pairs rules at a common target burden; in 199 of the 1,328 cells the per-query cap binds before a rule reaches it, because the recommendation proxy is exactly 1 per query. Restricting to cells whose realized burdens agree within 0.01 leaves every sign in Table 7 unchanged and shrinks the flat-pricing differences (+0.218 → +0.135 on MovieLens, +0.168 → +0.064 on Beauty).

Deployment decisions. Here a deployment is *valid* when its held-out mean capability meets the target and *false* otherwise. Table 8 applies the capability targets, burden cap, and one-SEM selection

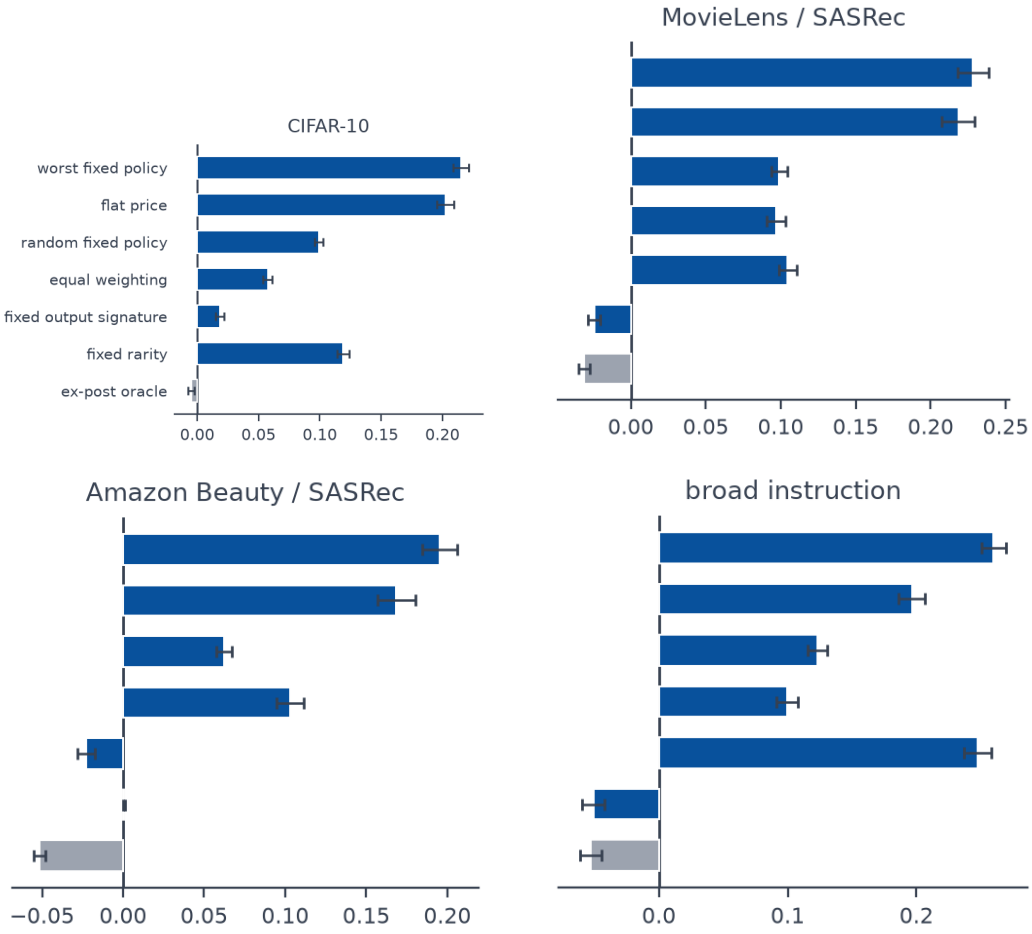


Figure 8: **Policy comparisons per workload.** Paired differences in remaining capability at matched burden, mean ± 1 SEM; right of zero means FGP left the attacker weaker. The ex-post oracle (gray) is a non-deployable reference. Fixed rarity is left of zero on MovieLens and broad instruction, where the fixed output signature is far right of zero; the output signature is left of zero only on Beauty.

rule of Section 5.4 to each declared policy on the reported seed split, with FGP selecting among the candidate policies of Figure 1. FGP deploys validly in nine of fifteen decisions, against seven for fixed rarity at the same median burden, 0.400, and every valid deployment of a fixed policy is also made by FGP. To test the six abstentions, we force a deployment at each. Among all (policy, burden) operating points under the cap, we evaluate the one with the lowest selection mean plus one SEM on the held-out seeds of the same split; all six fail their targets. The capability slope, the fall in normalized capability per unit burden, orders the workloads by the number of targets they support (Table 9). GSM8K abstains at every target because seed-to-seed noise is large relative to its capability range. The no-data and full-pool students differ by only 0.099 exact-match points (0.173 to 0.272), while the normalized seed-level spread is 0.283, against 0.049–0.106 on the other four workloads. Longer training does not widen the gap: on the full 1319-problem split the unpriced ceiling falls from 0.243 to 0.240 and 0.231 at three, six, and ten epochs, against a 0.176 floor. We therefore keep GSM8K in the query-level study and exclude its capability cells from policy comparisons.

Capability versus fidelity. Figure 9 compares capability with *fidelity*, the student's agreement with the victim's outputs. At the same benign burden, a feature can reduce output agreement more than useful task performance, so the evaluation must measure the objective the provider cares about.

Table 8: **Held-out capability evaluation under each declared policy** on the reported seed split. Capability targets 0.8, 0.6, and 0.4; burden cap 0.7; one-SEM selection rule. "FGP-only" counts decisions FGP deploys validly and the policy does not.

Policy	Valid	False	Med. burden	FGP-only
Flat price	0	0	—	9
Fixed rarity	7	0	0.400	2
Fixed output signature	6	0	0.550	3
Equal weighting	5	0	0.700	4
Logistic score	2	0	0.425	7
FGP (ours)	9	0	**0.400**	0

Table 9: **Capability slope and supported targets.** Slope is the fall in normalized capability per unit benign burden; • deploys and — abstains. Spearman correlation between slope and the number of deployments is +0.975.

	capability	deployment decision at capability target			
workload	slope	0.8	0.6	0.4	deploys
Amazon Beauty	0.93	•	•	•	3
MovieLens	0.93	•	•	•	3
CIFAR-10	0.69	•	•	—	2
broad instruction	0.48	•	—	—	1
GSM8K	0.04	—	—	—	0

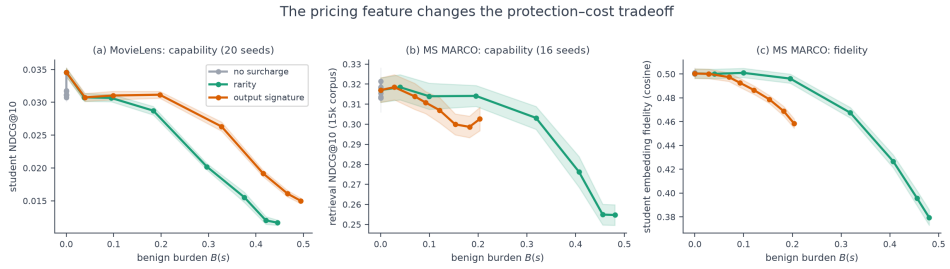


Figure 9: **The pricing feature changes the protection–cost tradeoff.** The horizontal axis is the average surcharge paid by benign users. Lower student performance means stronger protection; bands show ±1 standard error of the mean (SEM). **(a)** On MovieLens recommendation and **(b)** on MS MARCO retrieval, demand-rarity pricing reduces ranking performance (normalized discounted cumulative gain at rank 10, NDCG@10) more than output-signature pricing once the burden is large enough. **(c)** Output-signature pricing reduces agreement with the victim's embeddings more than it reduces retrieval performance. Output agreement and useful capability are therefore different evaluation targets.

Recommendation. On MovieLens at matched burden 0.4, flat pricing reduces extracted NDCG@10 by 12%, the output signature by 41%, and demand rarity by 61%. When the price-aware attacker may reallocate purchases over the full head-and-tail pool, rarity still reduces NDCG@10 by 48% against 17% for the output signature (8 seeds). Rarity's advantage over the output signature thus widens from 20 to 31 percentage points. Sweeping the attacker's total budget at this burden shows that flat pricing is nearly evaded (Figure 10): the student reaches 0.95 of unpriced capability once the budget covers the whole pool, and 0.9 at about 6.1k surcharge units. Aligned rules cap capability at 0.48 for rarity and 0.73 for the output signature and the fixed portfolio $r_D + 0.5o$. No seed reaches 0.9 under them (0/8), because the rarest, highest-coverage sessions stay unprofitable at every budget.

Rewriting on AG News. The best-of-k rewriting attack generates k paraphrases per query. To upper-bound attack strength, an offline selector sees each paraphrase's victim-label preservation and pricing score before purchase and takes the cheapest label-preserving paraphrase. At $k = 8$, over

three seeds (mean ± SEM), $56.6 \pm 0.2\%$ of paraphrases preserve the label, and at matched burden lexical-rarity pricing blocks $1.9 \pm 0.1\%$ of declared value while output-signature pricing blocks $78.3 \pm 1.1\%$. Because the output signature and the value proxy are both victim entropy here, this measures rewrite stability within that proxy. In a separate run that nests best-of-k' within $k = 64$ generations (Figure 10), lexical rarity blocks 7.6%, 1.9%, and 0.1% at $k = 1$, 8, and 64, while the output signature blocks 96.8%, 80.9%, and 46.2%. Each larger k expands the declared rewrite family of Proposition 7. In the end-to-end test at $k = 8$ (five seeds), the attacker retrains a student on the victim's soft labels for the rewrites it acquires. Under lexical pricing the student reaches 0.889 ± 0.003 accuracy against the unpriced 0.892 ± 0.002; the output signature cuts acquisitions 3.9-fold (753 against 2951) and lowers accuracy to 0.869 ± 0.003. The small capability gap is consistent with AG News reaching near-ceiling accuracy from a few hundred examples, a steep Ψ_θ in Corollary 9.

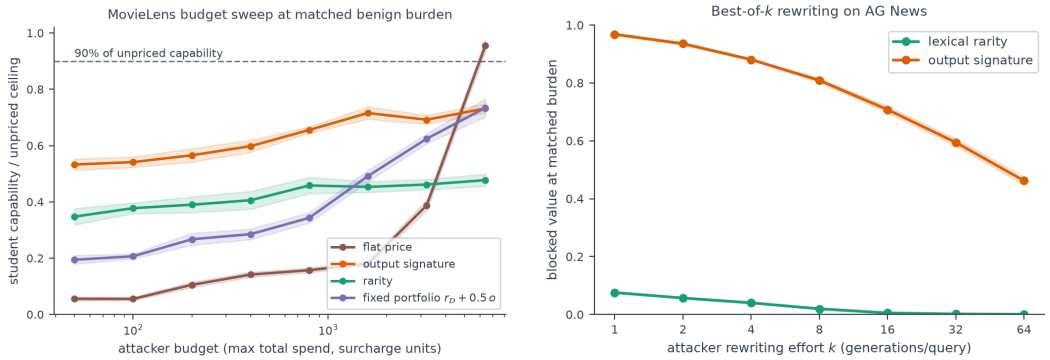


Figure 10: **Attacker-resource sweeps. Left:** flat MovieLens pricing is nearly evaded, while aligned rarity caps capability at 0.48 (8 seeds). **Right:** at $k = 64$, lexical rarity blocks 0.1% while the output signature blocks 46.2% (3 seeds).

Language models. On GSM8K, the attacker distills the math ability of Qwen2.5-3B-Instruct into a LoRA-tuned Qwen2.5-0.5B student. GSM8K prompts are rare under general-instruction demand in embedding space but not in victim-response entropy. Over six seeds, rarity pricing cuts affordable volume from 3000 to single digits, while the output signature never leaves fewer than 2289 (Figure 11). With 138 affordable examples the student is at least as accurate as with all 3000 (0.312 against 0.270). Accuracy reaches the no-distillation floor of 0.174 only once rarity pricing leaves about six examples, the saturating shape of Corollary 9.

Broad-instruction distillation needs coverage: the attacker distills from 6k diverse Alpaca prompts while demand is one dense embedding cluster. Across 16 seeds, the highest rarity intensity leaves 0.9k of 4.8k affordable prompts, and held-out NLL varies by 0.050 across the price sweep. At the highest output-signature intensity, 3.4k prompts remain and NLL varies by 0.007. Figure 12(a,b) compares the two at matched burden up to 0.4. A 7B victim reproduces the ranking over six seeds: 1.1k of 5.2k prompts remain and NLL varies by 0.053 under rarity, against 3.7k and 0.007 under the output signature.

C.4 Adaptive attackers and deployment

Adaptive and informed attackers. The closed-loop attacker best-responds to the posted rule: each round it re-estimates g_θ, buys every profitable query in its pool, queries the victim, and retrains. In the separable regime (five seeds, mean ± 1.96 SEM), demand-rarity pricing caps it at 0.310 ± 0.040 accuracy, against 0.502 ± 0.084 unpriced and 0.439 ± 0.075 under output-signature pricing. Its acquisitions plateau near 3.5k profitable queries, against 14.4k and 11.2k (Figure 12(c)). Re-estimating value does not restore the queries that rarity pricing makes unprofitable. Because the rule is public, the attacker can also re-optimize its purchases at fixed total spend. Buying in decreasing order of value per unit price gains no additional extraction value in any of 64 (domain, seed, target) settings. Without a budget, the best response buys exactly the profitable set (4), which the deterrence condition (5) already covers. Solving the budgeted subset problem exactly also recovers nothing an aligned rule removes: the aligned rule leaves 0.11 of value and 0.05 of surplus available, against

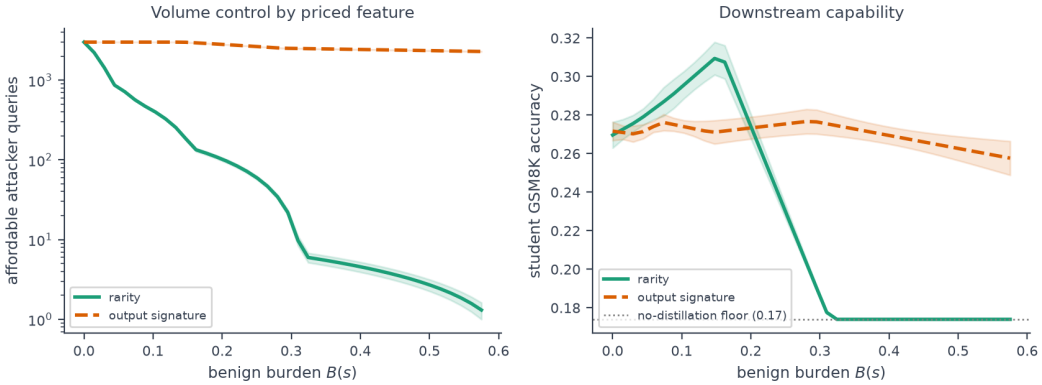


Figure 11: **GSM8K extraction** (Qwen2.5-3B-Instruct → Qwen2.5-0.5B, six seeds). **Left:** affordable attacker volume against benign burden; rarity pricing drives it from 3000 to single digits, while output-signature pricing never leaves fewer than 2289. **Right:** student accuracy stays at or above its unpriced level (0.27) until rarity pricing nearly empties the pool, then drops to the no-distillation floor (0.17).

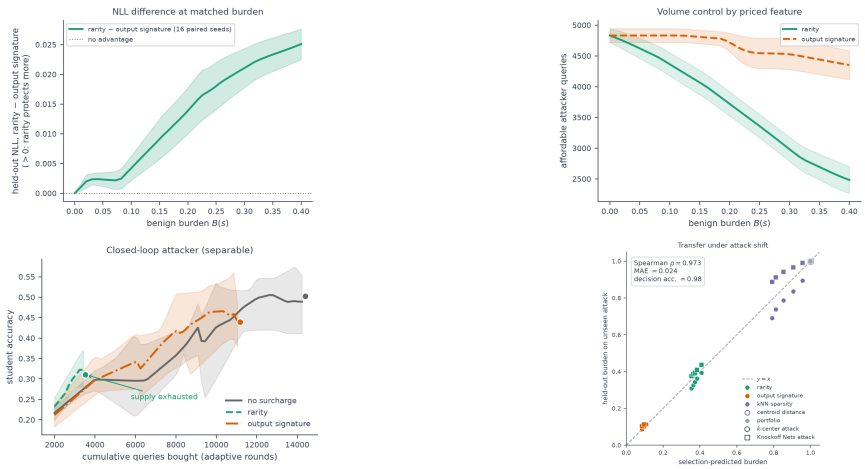


Figure 12: **End-to-end capability and held-out transfer. (a,b)** On broad instruction (16 seeds), rarity removes affordable diverse volume and raises held-out NLL more than the output signature. **(c)** Under closed-loop re-optimization (five seeds), rarity exhausts profitable supply near 3.5k acquisitions and caps accuracy near 0.31, against about 14k and 0.50 unpriced and 11k and 0.44 under the output signature. **(d)** Across 50 feature–target–attack cases, the selected action matches the held-out action in 49 cases; among the 45 with finite predicted burden, Spearman correlation is 0.973 and mean absolute error 0.024.

0.26 and 0.23 under a misaligned one. An attacker that buys the largest gain in embedding coverage per unit price, measured in 48 of the settings, gains 0.010 coverage on average and 0.060 at most while forfeiting 0.36 of the declared extraction value. Knowing the price reveals the estimated benign density, but every query in a protected family is already unprofitable through every declared rewrite. New rewrites or values require an expanded threat model and a fresh audit.

Comparison with proof of work. Calibrated proof of work (Dziedzic et al., 2022b) additionally uses accumulated per-user leakage. Figure 13 compares it with local pricing in client compute-seconds, implementing one surcharge unit as one compute-second and matching mean benign work. Both the per-user and the Sybil-resistant global variants leave remaining capability between 0.81 and 0.87 at every tested benign cost. FGP's local portfolio lowers it from 1.00 at 0.05 to 0.36 at 1.6 compute-seconds per query and is below both variants from 0.4 onward.

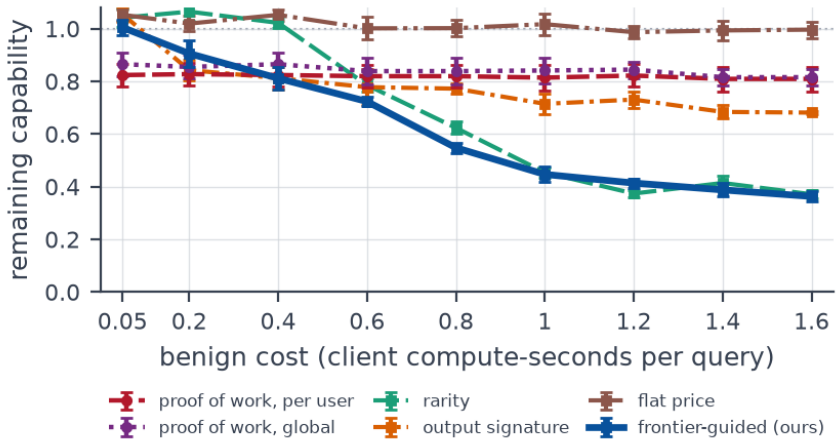
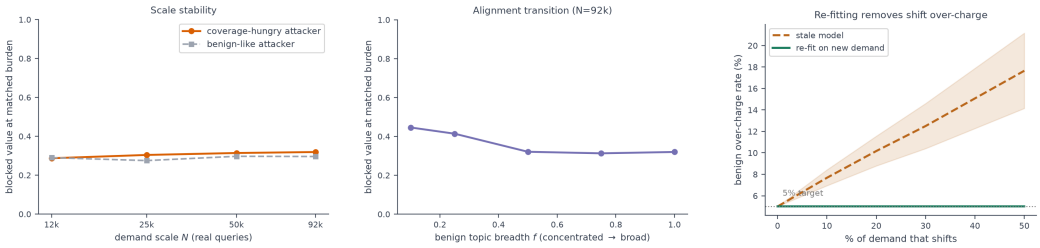


Figure 13: **Local pricing and calibrated proof of work at common benign cost.** Remaining attacker capability against benign client compute-seconds per query; four seeds, ± 1 SEM.

Demand changes. On MovieLens-1M we estimate demand from an earlier window of interactions and evaluate on a later one, with long-tail attacker continuations built independently of any rarity score. Rarity ranks first on the earlier window, but its protection of 0.55 is below the declared 0.80 target, so FGP abstains; under the real drift, the mean surcharge rises from 0.44 to 0.49. On MS MARCO, growing the benign sample from 12k to 92,373 queries with the same encoder leaves blocked value at matched burden 0.4 between 0.29 and 0.32 for a coverage-seeking attacker, and between 0.28 and 0.30 for one that imitates benign queries (Figure 14a). Broadening benign topic coverage at fixed scale lowers it from 0.45 to 0.32 (Figure 14b). The frontier therefore tracks how well benign and attacker queries separate. On CIFAR, blocked value reaches at least 88% by burden about 1.8 against each of the entropy, k-center, and random acquisition strategies, and refitting the density on half the demand pool preserves the surcharge ranking (Spearman 0.96). The over-charge rate, the share of benign queries above a rarity threshold set at 5% on the fitting demand, also returns to its target when the density is refit after a benign shift (Figure 14c).



(a) Blocked value is stable as the MS MARCO sample grows to 92,373 queries.

(b) Broader benign-topic coverage moves the frontier toward overlap.

(c) Under benign shift, a stale demand model over-charges more users; refitting on trusted traffic restores the 5% target (three seeds).

Figure 14: **Deployment stability and refresh.**

Who pays and contaminated demand. At mean burden 0.4 over $25{,}000$ benign queries, rarity charges the three rarest atypicality deciles the per-query cap and correlates with atypicality at Spearman 0.979. It leaves 53% of benign queries paying nothing (Gini 0.596). The output signature leaves 88% paying nothing, charges its rarest decile 0.53, below the cap, and correlates with atypicality at 0.503. Equal mean burden can therefore fall on different users. For a fixed family and fixed responses, the closure also minimizes monotone tail and group burden (Corollary 6), but the choice of family must account for the chosen objective. The benchmarks carry no group labels, so these results measure tail and atypicality burden only. In a density-poisoning simulation, moving the rarity cutoff

enough to cover the attacker's targets requires a corpus-sized injection relative to the $12{,}000$-query benign reference pool. The injected queries pay an average surcharge of 1.75 under the original rule, so the reference pool should come from independently trusted traffic.