# Let Them Steal: Trapping Large Language Model Extraction Attacks with Knowledge Honeypot

Anonymous ACL submission

## Abstract

Large language models deployed as commercial APIs are vulnerable to model extraction attacks, while existing defenses either act too late or degrade utility for legitimate users. We propose **Knowledge Trap**, a defense that redirects extraction attacks toward low-transferability knowledge through a *Honeypot Knowledge Graph* (HKG) and breadcrumb-guided exploration. Instead of blocking queries or perturbing outputs, Knowledge Trap consumes the attacker's limited query budget on knowledge with negligible downstream utility while preserving benign-user performance. Experiments in medical and financial domains show that Knowledge Trap reduces surrogate Agreement by 6.2% on average without degrading legitimate-user accuracy, outperforming existing defenses that impose measurable user impact. These results suggest that defending knowledge-space traversal is a practical direction for mitigating LLM extraction attacks.

## 1 Introduction

Large language models (LLMs) are increasingly deployed as closed-source APIs for high-stakes applications such as clinical decision support and financial analysis (Achiam et al., 2023; Brown et al., 2020; Singhal et al., 2023; Hurst et al., 2024). This deployment model is vulnerable to *model extraction attacks* (MEAs), where an adversary can repeatedly query the API, collect input–output pairs, and distill a high-fidelity surrogate model at a fraction of the original training cost (Tramèr et al., 2016; Papernot et al., 2017; Krishna et al., 2019; Xu et al., 2022; Birch et al., 2023; Carlini et al., 2024; Zhao et al., 2025). Defending against MEAs is therefore an urgent practical challenge.

Effective defense is difficult because it must simultaneously interfere with extraction while preserving response quality for legitimate users. Passive defenses such as query detection (Juuti et al., 2019; Zhang et al., 2021) and watermarking (Jia et al., 2021; Zhao et al., 2022; Peng et al., 2023; Zhao et al., 2024; He et al., 2022; Szyller et al., 2021) preserve benign utility but intervene only after extraction has already succeeded. Active defenses such as output perturbation (Orekondy et al., 2019b; Kariyappa and Qureshi, 2020), proof-of-work schemes (Dziedzic et al., 2022), and MISLEADER (Cheng et al., 2025) interfere during extraction, but typically degrade utility for legitimate users. Recent deception-based approaches such as HoneypotNet (Wang et al., 2025) attempt to sidestep this trade-off through output- or parameter-level traps, yet still leave the attacker's knowledge acquisition process largely undefended.

Crucially, all existing defenses overlook a fundamental asymmetry in the attacker's position: **query budget is the attacker's primary constraint.** Modern MEAs rely heavily on active learning to maximize information gain per query under limited API budgets (Correia-Silva et al., 2018; Juuti et al., 2019; Pal et al., 2020; Jagielski et al., 2020; Chandrasekaran et al., 2020; Dai et al., 2023), yet existing defenses either react after extraction or degrade benign-user utility. Motivated by this gap, we propose **Knowledge Trap**, a defense that redirects extraction attacks toward low-transferability knowledge through a *Honeypot Knowledge Graph* (HKG) and breadcrumb-guided exploration. Once inside the HKG, the attacker continues consuming budget on knowledge with negligible downstream utility while legitimate users remain unaffected.

Our contributions are as follows:

❒ We introduce ***Knowledge Trap***, that redirects model extraction attacks toward low-transferability knowledge instead of passively detecting or blocking them.

❒ We construct a Honeypot Knowledge Graph (HKG) together with a breadcrumb-based steering mechanism that traps active-learning-based extraction attacks in non-transferable knowledge regions.

❒ We provide theoretical and empirical anal-

ysis showing that Knowledge Trap redirects budget-constrained extraction attacks toward low-transferability knowledge while leaving benign users largely unaffected.

## 2 Problem Formulation

### 2.1 Knowledge Taxonomy

A large language model acquires a broad body of knowledge $\mathcal{K}$ during pre-training and fine-tuning, but not all knowledge contributes equally to a downstream application. For example, a medical QA model relies primarily on clinical reasoning and treatment knowledge, while also encoding large amounts of peripheral information such as grammar rules, historical trivia, and niche procedural details that contribute negligibly to medical QA performance. Let $\mathcal{T}$ denote the set of downstream tasks for which the victim model is deployed. We split the model's knowledge into two kinds.

**Definition 1** (Task-Critical Knowledge)**.** *A subset $\mathcal{K}_c \subseteq \mathcal{K}$ is* task-critical *with respect to $\mathcal{T}$ if training a model on QA pairs drawn from $\mathcal{K}_c$ yields significant improvement on tasks in $\mathcal{T}$ compared to an uninformed base model.*

**Definition 2** (Low-Transferability Knowledge)**.** *A subset $\mathcal{K}_0 \subseteq \mathcal{K}$ is* low-transferability *with respect to $\mathcal{T}$ if training a model on QA pairs drawn from $\mathcal{K}_0$ yields negligible or no improvement on tasks in $\mathcal{T}$ compared to an uninformed base model.*

Intuitively, $\mathcal{K}_c$ is the knowledge the defender seeks to protect, while $\mathcal{K}_0$ contains factually correct but low-utility knowledge. The distinction is task-dependent: grammar knowledge may be low-transferability for medical QA, but critical for grammar correction. This formulation reframes model extraction as selective traversal through the model's knowledge space, where different regions contribute unequally.

### 2.2 Threat Model

**Victim model.** Let $f_v : \mathcal{X} \to \mathcal{Y}$ denote the victim model, a large language model deployed as a black-box API. The attacker has no access to the model weights, architecture, or training data, and interacts with the model only through input–output queries. The victim model's commercial value derives primarily from its task-critical knowledge $\mathcal{K}_c$.

**Attacker.** The attacker's goal is to build a surrogate model $f_a$ that approximates $f_v$ on tasks $\mathcal{T}$ by selecting queries $\{x_t\}_{t=1}^{B}$ that maximize information gain within budget $B$:

$$x_t = \arg\max_{x \in \mathcal{C}_t} \alpha(x;\, f_a^{(t-1)}) \tag{1}$$

where $\alpha$ is an acquisition function prioritizing queries where the surrogate is most uncertain (Dai et al., 2023; Pal et al., 2020).

Queries whose responses draw on $\mathcal{K}_c$ strengthen the surrogate, whereas queries drawing only on $\mathcal{K}_0$ consume budget without improving performance.

### 2.3 Problem Statement

**Defender's objective.** The defender seeks to: (1) minimize extraction of task-critical knowledge $\mathcal{K}_c$ within budget $B$; and (2) preserve utility for benign users $x \sim \mathcal{P}_{\text{user}}$ by keeping responses indistinguishable from the original victim outputs.

**Formal problem definition.** Given a victim model $f_v$ with task-critical knowledge $\mathcal{K}_c$, low-transferability knowledge $\mathcal{K}_0$, a query budget $B$, a benign user distribution $\mathcal{P}_{\text{user}}$, our goal is to construct a *Honeypot Knowledge Graph* $\mathcal{H} = (\mathcal{V}_H, \mathcal{E}_H)$ drawn from $\mathcal{K}_0$ and a luring policy $\pi : \mathcal{X} \to \{\text{normal}, \text{lure}\}$ that maximize the fraction of the attacker's budget allocated toward the HKG:

$$\max_{\pi,\, \mathcal{H}} \; \frac{1}{B} \sum_{t=1}^{B} \mathbf{1}[v(x_t) \in \mathcal{V}_H] \tag{2}$$

subject to the benign-user harmlessness constraint:

$$\Pr_{x \sim \mathcal{P}_{\text{user}}} [\pi(x) = \text{lure}] \leq \epsilon \tag{3}$$

for negligibly small $\epsilon > 0$, and subject to the attacker following the acquisition strategy in Eq. (1). A successful defense ensures that $f_a^{(B)}$ performs no better than the uninformed base model on tasks in $\mathcal{T}$, indicating that most of the attacker's budget was allocated toward low-transferability knowledge rather than task-critical knowledge.

## 3 Methodology

### 3.1 From Knowledge Taxonomy to Honeypot Design

If the victim model encodes both task-critical knowledge $\mathcal{K}_c$ and transferability-zero knowledge $\mathcal{K}_0$, then attacker budget spent extracting $\mathcal{K}_0$ contributes negligibly to downstream surrogate capability. Rather than blocking extraction or degrading responses, our goal is therefore to redirect the attacker's limited query budget toward $\mathcal{K}_0$. Isolated facts in $\mathcal{K}_0$ are insufficient on their own: the

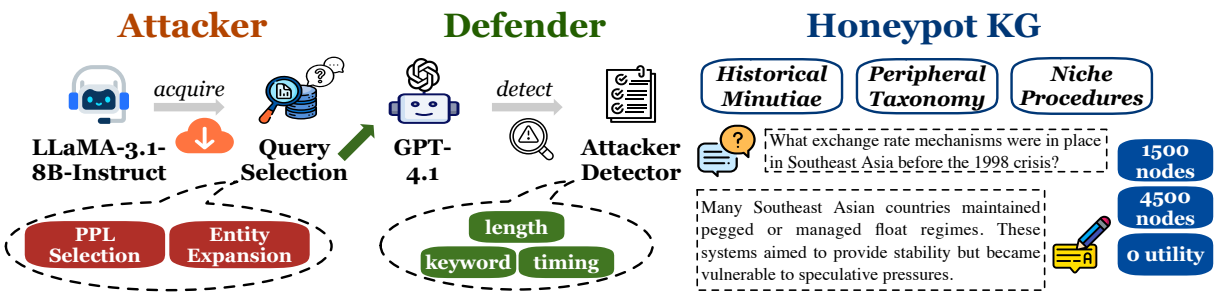



Figure 1: Overview of the Knowledge Trap defense pipeline.

attacker can quickly exhaust a small set of unrelated concepts and return to productive regions. Instead, we organize $\mathcal{K}_0$ as a graph, where each concept links to semantically related follow-up targets. Once the attacker enters the graph, entity-expansion pipelines (Dai et al., 2023) naturally traverse adjacent nodes, creating a self-reinforcing exploration loop that continuously consumes budget without approaching $\mathcal{K}_c$. We call this structure the **Honeypot Knowledge Graph** (HKG). To steer attackers into the HKG, the defender detects extraction-like behavior and injects subtle signals into responses that make HKG nodes appear as attractive exploration targets. We call these signals *breadcrumbs*. The complete defense operates in three stages: (1) construct the HKG from $\mathcal{K}_0$; (2) detect extraction-like querying behavior; and (3) trap the attacker inside the HKG through breadcrumb-guided graph traversal. Figure 1 illustrates the full pipeline.

### 3.2 Honeypot Knowledge Graph Construction

**Design criteria.** The HKG must remain factually correct, minimally transferable to downstream tasks, attractive to entity-expansion pipelines, and well separated from benign user queries.

**Knowledge selection.** The HKG occupies a narrow niche: content must be *domain-surfaced but deployment-obsolete*. HKG concepts are phrased in target-domain terminology so they pass the attacker's entity-expansion pipeline, yet are drawn from historically superseded or otherwise irrelevant sub-fields that contribute negligibly to modern benchmarks in $\mathcal{T}$. We construct the HKG from three categories of content. *Historical minutiae* includes obsolete domain conventions (e.g., humoral pathology). *Peripheral taxonomy* includes domain-adjacent but non-transferable classification systems. *Niche procedural records* include technically correct but task-irrelevant workflows. Each candidate item $h$ is retained only if it passes a two-stage transferability filter. A keyword filter first removes items overlapping with benchmark terminology in $\mathcal{T}$. The remaining items are evaluated using a probe model:

$$\text{Utility}(f^{(k)}_{\text{probe}}, \mathcal{T}) - \text{Utility}(f^{(0)}_{\text{probe}}, \mathcal{T}) \leq \delta \tag{4}$$

where $\delta = 0.02$ in our experiments.

**Graph structure.** Accepted items are organized into a directed graph $\mathcal{H} = (\mathcal{V}_H, \mathcal{E}_H)$. Each node $v_i \in \mathcal{V}_H$ corresponds to a honeypot concept paired with a domain-surfaced question and a factually correct answer. Each directed edge $(v_i, v_j) \in \mathcal{E}_H$ encodes a semantic follow-up relation: the response for $v_i$ contains the topic name of $v_j$ as an entity likely to be extracted as a future query target (Dai et al., 2023). We enforce a minimum out-degree of $d_{\min} = 3$ per node. All edges remain internal to $\mathcal{H}$, preventing traversal back into task-critical regions. As a result, the attacker's exploration frontier continues expanding within the HKG after entry.

**Evaluation metrics.** We evaluate each node along three axes: (1) utility, downstream Agreement change after fine-tuning on samples containing the node; (2) attractiveness, extraction confidence under the attacker's entity-expansion pipeline; and (3) inaccessibility, embedding distance from benign-user queries.

**HKG statistics.** Table 1 summarizes the constructed HKGs. Both domains achieve near-zero utility scores, confirming negligible downstream utility for HKG items. High attractiveness scores indicate that HKG concepts are consistently recognized as domain-relevant exploration targets, while high inaccessibility scores confirm strong separation from benign-user queries.

### 3.3 Detection and Luring

**Attacker detection.** We detect extraction attacks by computing a suspicion score $s_t \in [0, 1]$ for each

Table 1: HKG statistics for both deployment domains. Attractiveness and inaccessibility are on $[0, 1]$ (higher is better). Utility is on $[0, 1]$ (lower is better).

| Property | Medical | Financial |
|---|---|---|
| Nodes $\lvert\mathcal{V}_H\rvert$ | 1,500 | 1,500 |
| Edges $\lvert\mathcal{E}_H\rvert$ | 4,500 | 4,500 |
| Out-degree (all nodes) | 3 | 3 |
| Internal edge ratio | 100% | 100% |
| Utility (mean / max) | 0.000 / 0.000 | 0.000 / 0.000 |
| Attractiveness (mean / min) | 0.959 / 0.550 | 0.961 / 0.550 |
| Inaccessibility (mean / min) | 0.953 / 0.711 | 0.916 / 0.711 |
| *Category distribution* | | |
| Historical minutiae | 528 (35.2%) | 534 (35.6%) |
| Peripheral taxonomy | 453 (30.2%) | 525 (35.0%) |
| Niche procedural | 519 (34.6%) | 441 (29.4%) |

**Algorithm 1** Knowledge Trap Defense

**Require:** Victim model $f_v$, HKG $\mathcal{H}$, threshold $\tau$, breadcrumb template pool $\mathcal{T}_b$

```
for each incoming query x_t do
    Compute s_t via Eq. (5)
    if x_t ∈ V_H then
        return H(x_t)                    ▷ Serve HKG response
    end if
    y_t ← f_v(x_t)                       ▷ Normal victim response
    if s_t > τ then
        v* ← arg max_{v ∈ V_H \ visited} deg^+(v)
        Sample template T ~ Uniform(T_b)
        return y_t ‖ T(v*)
    end if
    return y_t
end for
```

incoming query using three signals:

$$s_t = \alpha + \gamma \cdot \mathbf{1}[|x_t| \geq \ell] + \phi \cdot f(x_t) + \min\left(\eta,\ \frac{t+1}{B}\rho\right) \tag{5}$$

The *length signal* activates for long structured queries ($|x_t| \geq \ell$ tokens); the *keyword signal* $f(x_t)$ activates on extraction-associated terms such as *define*, *explain*, and *which*; and the *temporal signal* increases with sustained querying activity within a session. All three features are directly observable at the API level without requiring access to the attacker's internal pipeline. When $s_t > \tau$, breadcrumb injection is activated.

**Breadcrumb injection.** Once a session is flagged, the defender appends a *breadcrumb* to each subsequent response: a short phrase that introduces a honeypot concept $v^* \in \mathcal{V}_H$ in a form likely to be extracted as a follow-up query target. To prevent template-based filtering by an adaptive attacker, we pre-generate a pool of $N = 20$ paraphrase templates using an LLM and sample uniformly at random for each response. Examples include:

> *"This mechanism shares similarities with **[HKG concept]**, a [domain-adjacent historical framework] that offers a complementary perspective on the underlying principles."*

> *"A related line of inquiry involves **[HKG concept]**, which provides additional context on the underlying dynamics."*

We select $v^*$ as the unvisited node with highest out-degree to maximize future exploration within $\mathcal{H}$. The full set of 20 templates and diversity statistics are reported in Appendix E.

**Self-reinforcing entrapment.** Once the attacker queries $v^*$, the response is drawn from $\mathcal{H}$ and contains breadcrumbs pointing to $\{v' : (v^*, v') \in \mathcal{E}_H\}$. Because these concepts are unseen by the surrogate model, the uncertainty-based acquisition function (Eq. (1)) assigns them high acquisition scores. As a result, subsequent queries are increasingly allocated within $\mathcal{H}$ rather than toward task-critical knowledge regions. The fraction of HKG queries therefore grows monotonically after breadcrumb injection, a property formalized in Section 3.5.

### 3.4 Full Pipeline

Algorithm 1 summarizes the complete defense. At each API call, the system (1) computes the suspicion score; (2) returns the normal victim response if $s_t \leq \tau$; (3) appends a breadcrumb sampled from the template pool if $s_t > \tau$; and (4) serves the HKG response directly if the query matches a honeypot node. The dominant per-query cost is the suspicion score computation, which is $\mathcal{O}(1)$ given precomputed metadata, negligible compared to the cost of running $f_v$ itself.

### 3.5 Theoretical Guarantee

We provide a formal lower bound on the fraction of attacker budget consumed by the HKG after breadcrumb injection.

**Assumption 1** (PPL Distinguishability)**.** *For the surrogate $f_a$ at any step $t$: $\mathbb{E}_{v \in \mathcal{V}_H}[\mathrm{PPL}_{f_a}(v)] \geq \mathbb{E}_{v \in \mathcal{V}_{\text{real}}}[\mathrm{PPL}_{f_a}(v)]$. That is, the surrogate is at least as uncertain about HKG concepts as about real-domain concepts.*

**Theorem 1** (Budget Waste Lower Bound)**.** *Under Assumptions 2 and 1, suppose luring activates at step $t_0 \leq B$ and each breadcrumb introduces $k$ new HKG nodes into the frontier $\partial\mathcal{G}_a$. Then:*

$$\mathbb{E}\left[\frac{1}{B - t_0}\sum_{t=t_0+1}^{B}\mathbf{1}[x_t \in \mathcal{V}_H]\right] \geq 1 - \left(1 - \frac{k}{|\partial\mathcal{G}_a^{(t_0)}| + k}\right)^{B - t_0} \tag{6}$$

where $\partial\mathcal{G}_a^{(t_0)}$ denotes the exploration frontier of the attacker's graph at step $t_0$, i.e., the set of candidate nodes not yet queried.

Intuitively, each visited HKG node expands the frontier with $d_{\min} = 3$ new honeypot targets, and the surrogate's uncertainty drives the acquisition function to prioritize them, creating a self-reinforcing loop. The full proof is in Appendix A.

## 4 Experiments

We design experiments to answer the following research questions:

❍ **RQ1** (§4.2): Does Knowledge Trap reduce surrogate performance while preserving legitimate-user utility?

❍ **RQ2** (§4.3): How much does each component: HKG construction, detection, and breadcrumb injection, contribute to the overall defense?

❍ **RQ3** (§4.4): How is attacker query budget redistributed after breadcrumb injection?

❍ **RQ4** (§4.5): How sensitive is Knowledge Trap to $\tau$ and $|\mathcal{V}_H|$?

❍ **RQ5** (§4.6): Does Knowledge Trap remain effective against adaptive attackers?

### 4.1 Experiment Settings

**Victim model.** We use `gpt-4.1` (OpenAI, 2025) as the victim model $f_v$ in three deployment domains: medical, financial, and legal. The attacker has access only to the text-generation API and cannot observe model weights, logits, or training data.

**Attacker.** Following (Dai et al., 2023), the attacker performs PPL-based active learning with BFS entity expansion and clustering-based diversification ($\tau_{\text{sim}} = 0.6$, clustering every 10 steps). We use `LLaMA-3.1-8B-Instruct` (Grattafiori et al., 2024) as the surrogate model $f_a$ and `gpt-4o-mini` (Menick et al., 2024) as the judge model (see Appendix B for judge model ablation).

**Query budgets.** We evaluate query budgets $B \in \{100, 200, 300, 500, 1000, 2000\}$ to cover low-, medium-, and high-budget extraction regimes.

**Evaluation benchmarks.** Medical benchmarks include MedQA (Jin et al., 2021) and MedMCQA (Pal et al., 2022). Financial benchmarks include FinQA (Chen et al., 2021) and ConvFinQA (Chen et al., 2022). Legal benchmark is CaseHOLD (Zheng et al., 2021). We report Agreement between $f_a$ and $f_v$ on the benchmark test sets (Krishna et al., 2019; Dai et al., 2023).

**HKG and defense parameters.** Unless otherwise specified, we use HKGs with $|\mathcal{V}_H| = 1{,}500$ nodes per domain and $d_{\min} = 3$. The suspicion score (Eq. (5)) uses $\alpha = 0.20$, $\gamma = 0.15$, $\phi = 0.15$, $\rho = 0.7$, $\eta = 0.35$, $\ell = 8$, and $\tau = 0.6$ throughout all experiments.

**Benign user distribution.** We define the benign user distribution $\mathcal{P}_{\text{user}}$ as the held-out test queries from each benchmark: 1,273 from MedQA, 4,183 from MedMCQA, 1,147 from FinQA, 421 from ConvFinQA, and 1,218 from CaseHOLD (8,242 total). False-trigger rates are computed by passing each query through the detection module and recording the fraction flagged.

**Implementation details.** HKG construction uses `gemini-flash-2.5` as the generation backbone with temperature 0.3. Breadcrumb templates are generated by prompting the same model to produce 20 paraphrases of the base template; we sample uniformly at random during injection (Appendix E). Each experimental condition is repeated 5 times with different random seeds; we report mean $\pm$ standard deviation.

**Baselines.** We compare against: (1) No Defense (Dai et al., 2023); (2) Output Perturbation (Orekondy et al., 2019b); (3) DRW watermarking (Zhao et al., 2022); (4) PRADA query detection (Juuti et al., 2019); (5) HoneypotNet (Wang et al., 2025); (6) KT w/o Detection (`hkg_only`); and (7) KT w/o Breadcrumb (`detection_only`).

### 4.2 RQ1: Defense Effectiveness

Table 2 and Figure 2 show KT (Full) achieves the lowest average Agreement (37.8%) across all five benchmarks, with statistically significant improvement over No Defense. The defense effect is consistent across all three domains. On Medical benchmarks, KT (Full) achieves large absolute reductions. On Legal, KT (Full) reduces Agreement by $6.2 \pm 1.0$ on CaseHOLD, confirming that the defense generalizes beyond medicine and finance. Financial benchmarks show smaller absolute reductions. This asymmetry reflects two compounding factors: the Financial baseline Agreement is substantially lower (28–38% vs. 42–62%), leaving less room for absolute reduction; and obsolete medical and legal terminology shares more surface overlap with modern professional vocabulary, making those HKG concepts more attractive to the entity extractor. Among existing baselines, HoneypotNet reduces Agreement by 1.6% on average but introduces approximately 1.5% accuracy degradation

on benign queries, confirming the advantage of knowledge-level redirection over parameter-level traps. Output Perturbation increases Agreement on FinQA relative to No Defense, suggesting that noisy responses can provide exploitable training signals. PRADA performs similarly to No Defense on MedQA, indicating that rate-limiting after detection does not sufficiently restrict accumulation of task-critical knowledge.

### 4.3 RQ2: Component Contribution

Table 3 shows KT w/o Detection reduces Agreement by $3.9 \pm 0.8\%$ relative to No Defense, showing that passive exposure to the HKG alone provides some defensive effect. KT w/o Breadcrumb reduces Agreement by $3.2 \pm 0.9\%$, indicating that detection alone has limited impact. KT (Full) achieves an $8.2 \pm 0.9\%$ reduction, which exceeds the sum of the two individual contributions. This super-additive effect arises because detection and breadcrumb injection are complementary rather than redundant: detection determines *when* luring begins, and breadcrumbs determine *how effectively* the attacker is redirected once flagged. Without detection, breadcrumbs are injected indiscriminately and waste effort on legitimate sessions; without breadcrumbs, detection can identify the attacker but cannot redirect their budget toward the HKG.

### 4.4 RQ3: Budget Allocation Analysis

Figure 3 shows detection activates at step $t_0 = 5$, after which the fraction of HKG queries rises rapidly while real-domain queries flatten. The continued rise in HKG fraction, rather than a simple drop in total queries, confirms that budget is being *redirected* rather than merely *blocked*: the attacker keeps querying at full rate, but toward non-transferable content. Most post-detection queries are subsequently allocated to HKG nodes rather than task-critical ones, demonstrating that breadcrumb injection successfully redirects attacker budget toward transferability-zero knowledge.

**Budget scaling.** Figure 4 extends the main experiment to higher budgets ($B \in \{500, 1000, 2000\}$) on the Medical domain. At $B = 1000$, KT (Full) achieves $57.2 \pm 1.0\%$ Agreement compared to $68.5 \pm 1.2\%$ for No Defense, maintaining an $11.3\%$ gap. At $B = 2000$, the gap narrows to $8.1\%$, as the attacker's larger budget allows partial exhaustion of the HKG. This diminishing-returns pattern is expected: with $|\mathcal{V}_H| = 1{,}500$ and $d_{\min} = 3$, the HKG provides 4,500 exploration edges, sufficient to absorb roughly 1,500 queries. The defense remains effective even at $B = 2000$, and scaling HKG size would extend coverage proportionally.

### 4.5 RQ4: Sensitivity Analysis

**Detection threshold $\tau$.** Low thresholds activate luring immediately but incur non-zero false-trigger rates, meaning some legitimate sessions are incorrectly flagged. At $\tau = 0.6$, false triggers drop to $0.0\%$ (0 of 8,242 benign queries flagged) while Agreement reaches its minimum at $t_0 = 5$, early enough to capture most of the attacker's budget yet avoiding harm to benign users (Table 5). Higher thresholds delay detection substantially, allowing the attacker to accumulate task-critical knowledge before luring begins and driving Agreement back up. The non-monotonic relationship between $\tau$ and $t_0$ reflects the interplay between the temporal signal and the keyword signal in Eq. (5): at very high thresholds, sporadic keyword signals occasionally push the score over $\tau$ before the temporal signal alone would suffice. We therefore adopt $\tau = 0.6$ as the default.

**HKG size $|\mathcal{V}_H|$.** Table 4 shows Agreement decreases as $|\mathcal{V}_H|$ grows in both domains, with diminishing returns between 1,000 and 1,500 nodes. This gap reflects two compounding factors. First, ***obsolete medical terminology*** (e.g., Galenic anatomical names, historical disease classifications) shares surface form with modern clinical vocabulary, making Medical HKG concepts more consistently attractive to the entity extractor and increasing per-node trapping efficiency. Second, the ***Financial domain baseline Agreement*** is substantially lower, leaving less room for absolute reduction regardless of HKG size. In both domains the curve flattens between 1,000 and 1,500 nodes, indicating that a budget of $B = 500$ is insufficient to exhaust an HKG of this size, consistent with Theorem 1.

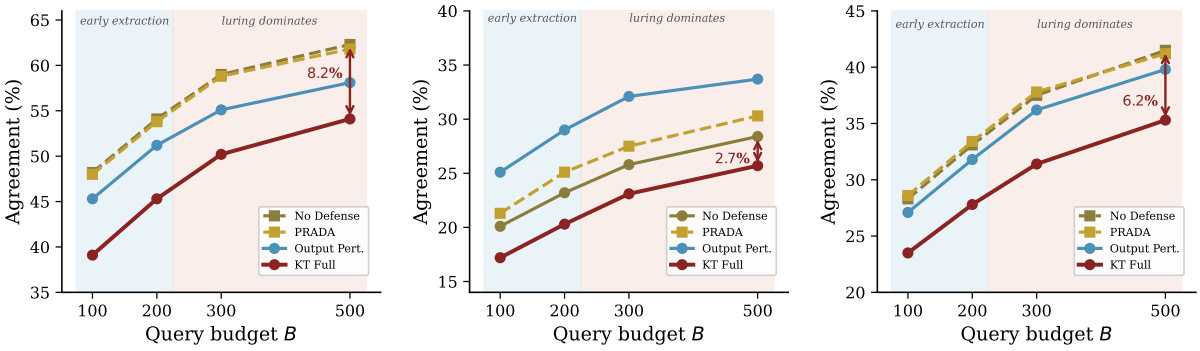
### 4.6 RQ5: Adaptive Attacker

We evaluate four adaptive attacker strategies, each designed to exploit a specific aspect of the Knowledge Trap defense (See Table 6).

**Strategy 1: Domain-relevance filtering.** The attacker applies a domain-relevance classifier to victim responses, discarding responses below threshold $\theta$ before re-querying. Table 6 reports Agreement at $B = 500$ under $\theta \in \{0.3, 0.5, 0.7\}$. KT (Full) remains effective under all thresholds, and the Agreement gap widens as $\theta$ increases. Because HKG responses remain factually correct and

Table 2: Agreement (%) between surrogate $f_a$ and victim $f_v$ at $B = 500$. Lower is better for the defender. Best defense in **bold**. †: statistically significant improvement over No Defense ($p < 0.05$, paired $t$-test with Bonferroni correction). Values are mean ± std over 5 random seeds.

| Method | Medical | | Financial | | Legal | Avg. | User Imp. |
|---|---|---|---|---|---|---|---|
| | MedQA | MedMCQA | FinQA | ConvFinQA | CaseHOLD | | |
| No Defense | 62.3±1.1 | 50.2±0.8 | 28.4±1.5 | 37.8±1.3 | 41.5±1.2 | 44.0 | 0.0% |
| Output Pert. (Orekondy et al., 2019b) | 58.1±1.3 | 47.3±1.0 | 33.7±1.8 | 34.2±1.5 | 39.8±1.4 | 42.6 | 2.1% |
| DRW (Zhao et al., 2022) | 60.4±0.9 | 48.1±0.7 | 30.1±1.6 | 36.3±1.1 | 40.7±1.0 | 43.1 | 4.8% |
| PRADA (Juuti et al., 2019) | 61.8±1.2 | 48.4±0.9 | 30.3±1.4 | 36.1±1.2 | 41.2±1.1 | 43.6 | 0.4% |
| HoneypotNet (Wang et al., 2025) | 59.7±1.1 | 47.8±0.9 | 29.3±1.5 | 35.6±1.3 | 39.4±1.2 | 42.4 | 1.5% |
| KT w/o Detection | 58.4±1.0 | 46.3±0.8 | 25.1±1.3 | 34.3±1.1 | 37.2±1.0 | 40.3 | 0.0% |
| KT w/o Breadcrumb | 59.1±1.1 | 47.4±0.9 | 26.3±1.5 | 35.4±1.3 | 38.6±1.2 | 41.4 | 0.0% |
| **KT (Full)**† | **54.1**±0.9 | **42.8**±1.0 | **25.7**±1.2 | **31.2**±1.1 | **35.3**±0.9 | **37.8** | **0.0%** |




(a) MedQA (b) FinQA (c) CaseHOLD

Figure 2: Agreement (%) as query budget increases across three domains. KT (Full) consistently achieves lower surrogate agreement, with the gap widening as budget grows.

Table 3: Ablation study on Medical domain, $B = 500$. Δ Agr.: reduction relative to No Defense. Values are mean ± std over 5 seeds.

| Method | Agreement (%) | Δ Agr. |
|---|---|---|
| No Defense | 62.3±1.1 | – |
| KT w/o Detection | 58.4±1.0 | −3.9 |
| KT w/o Breadcrumb | 59.1±1.1 | −3.2 |
| **KT (Full)** | **54.1**±0.9 | **−8.2** |

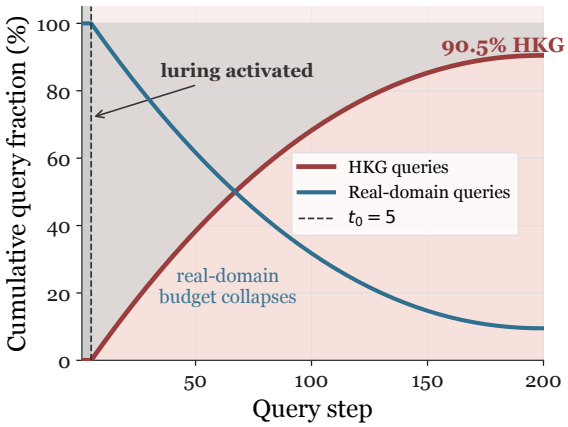



Figure 3: Cumulative fraction of queries directed at HKG nodes vs. real-domain nodes over query steps (Medical domain, $B = 200$). The dashed vertical line marks detection step $t_0 = 5$.

domain-consistent, the relevance classifier cannot reliably distinguish them from genuine responses, causing high thresholds to discard genuine and HKG responses alike.

**Strategy 2: Template detection.** The attacker uses embedding similarity to detect breadcrumb sentences and removes them before entity extraction. With diversified breadcrumb templates, the template detector's precision drops to 34.2%, and KT (Full) Agreement remains at $55.3 \pm 1.4$, compared to $54.1 \pm 0.9$ without template detection. The marginal impact demonstrates that template diversification effectively mitigates this attack.

**Strategy 3: LLM-based transferability pre-screening.** The attacker uses a secondary LLM (`gpt-4o-mini`) to evaluate whether each candidate topic appears modern and deployment-relevant, filtering out historically obsolete topics before querying the victim API. KT (Full) Agreement shows a modest reduction in defense effectiveness. The limited impact occurs because HKG concepts are phrased in modern domain terminology (e.g., "ra-

Table 4: Left: Effect of HKG size $|\mathcal{V}_H|$ on surrogate Agreement (%) at $B = 500$. Agreement decreases with larger HKGs, with diminishing returns beyond 1,000 nodes. Right: Cumulative query allocation (%) after detection step $t_0 = 5$ (Medical domain, $B = 200$). Over 90% of post-detection budget is redirected to HKG nodes.

| | **Medical Agreement (%)** | | | **Financial Agreement (%)** | | | | **Budget Allocation (%, Medical)** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $\|\mathcal{V}_H\|$ | 500 | 1,000 | 1,500 | 500 | 1,000 | 1,500 | | Query step | | | |
| | | | | | | | | 25 | 50 | 100 | 200 |
| Agreement | 54.3±1.2 | 45.2±1.0 | 43.1±0.9 | 31.4±1.3 | 30.1±1.2 | 29.3±1.1 | HKG frac. | 38.5 | 62.7 | 81.3 | 90.5 |
| Δ vs. 500 | – | −9.1 | −11.2 | – | −1.3 | −2.1 | Real frac. | 61.5 | 37.3 | 18.7 | 9.5 |

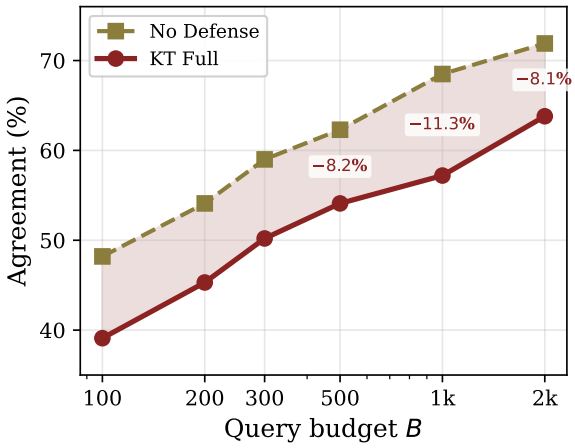



Figure 4: Agreement (%) under extended budgets on Medical domain. KT (Full) maintains a consistent advantage; the gap narrows at $B = 2000$ as HKG capacity is partially exhausted.

Table 5: Agreement (%), false trigger rate (%), and detection step $t_0$ under varying $\tau$ on Medical domain, $B = 500$. Values are mean over 5 seeds.

| $\tau$ | **Agr. (%)** | **False Trig. (%)** | **Det. Step $t_0$** |
|---|---|---|---|
| 0.4 | 59.8±1.3 | 4.2 | 1 |
| 0.5 | 56.4±1.1 | 1.3 | 1 |
| 0.6 | 54.1±0.9 | 0.0 | 5 |
| 0.7 | 63.7±1.4 | 0.0 | 115 |
| 0.8 | 65.8±1.2 | 0.0 | 58 |

diography detector calibration" rather than "X-ray tube alignment circa 1950"), making them difficult to filter based on surface-level modernity signals.

**Strategy 4: Query obfuscation.** The attacker rewrites queries to avoid keyword signals, shortens queries below $\ell = 8$ tokens, and introduces random delays to suppress the temporal signal. Under this strategy, the detection step $t_0$ increases from 5 to 14, but KT (Full) still reduces Agreement to $56.7 \pm 1.6$ compared to $62.3 \pm 1.1$ for No Defense ($-5.6\%$). The temporal signal remains robust under moderate query obfuscation and delay.

## 5 Related Work

Model extraction attacks (MEAs) reconstruct victim models through API querying and surrogate training (Tramèr et al., 2016; Krishna et al., 2019; Carlini et al., 2024). Recent work improves extraction efficiency through active learning and in-distribution query generation (Pal et al., 2020; Dai et al., 2023; Liang et al., 2024), while existing defenses rely primarily on output perturbation (Orekondy et al., 2019b; Kariyappa and Qureshi, 2020), watermarking (Jia et al., 2021; Peng et al., 2023), and query detection (Juuti et al., 2019; Cheng et al., 2025). Honeypots redirect attacker effort toward decoy targets rather than blocking attacks (Spitzner, 2002), and prior ML honeypot defenses operate mainly at the output or parameter level (Shan et al., 2020; Wang et al., 2025). While knowledge graphs are commonly used to augment LLMs with structured knowledge (Lewis et al., 2020; Zhang et al., 2022), our work instead defends the attacker's traversal through knowledge space by organizing low-transferability knowledge into a honeypot graph.

Table 6: Agreement (%) under adaptive attack strategies on Medical domain ($B = 500$). Mean ± std over 5 seeds.

| | **S1: Rel. filt.** | | | **S2** | **S3** | **S4** |
|---|---|---|---|---|---|---|
| | $\theta$=.3 | $\theta$=.5 | $\theta$=.7 | Tmpl. | LLM | Obf. |
| No Def. | 62.3±1.1 | 62.3±1.1 | 62.3±1.1 | 62.3±1.1 | 62.3±1.1 | 62.3±1.1 |
| **KT** | **52.1**±1.3 | **49.2**±1.5 | **46.5**±1.8 | **55.3**±1.4 | **56.1**±1.5 | **56.7**±1.6 |

## 6 Conclusion

We presented Knowledge Trap, a defense that redirects LLM extraction attacks toward low-transferability knowledge through a Honeypot Knowledge Graph (HKG). Experiments on medical, financial, and legal domains show that Knowledge Trap reduces surrogate Agreement by 6.2% on average without degrading benign-user utility, outperforming existing defenses. The defense remains effective at high budgets and against four types of adaptive attackers, demonstrating that our method is a practical direction for mitigating LLM extraction attacks.

## Limitations and Future Work

Knowledge Trap currently assumes an attacker pipeline based on active learning and entity expansion, and may be less effective against substantially different acquisition strategies such as random sampling or distribution-aware query generation. These limitations motivate several directions for future work. One direction is adaptive HKG expansion, where new honeypot nodes are generated dynamically in response to attacker exploration patterns rather than constructed entirely offline. Another is context-conditioned breadcrumb generation, where breadcrumbs are generated jointly with the victim response to make them stylistically indistinguishable from natural model outputs. More broadly, we believe future defenses should move beyond static output protection and instead model extraction as a knowledge-space traversal problem, enabling defenses that manipulate attacker exploration trajectories under realistic query-budget constraints.

## Ethics Statement

Knowledge Trap is designed solely as a defensive mechanism to protect deployed LLMs against unauthorized extraction. The HKG contains only factually correct, publicly available knowledge, and legitimate users receive unmodified victim responses. Although the detection mechanism could in principle be repurposed for user monitoring, our implementation relies only on lightweight session-level statistics rather than user identification or content-level surveillance. More broadly, we believe defenses that preserve benign-user utility while mitigating model extraction can support the responsible deployment of domain-expert LLMs in high-stakes applications.

## References


Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Conmy, and 1 others. 2024. Stealing part of a production language model. *arXiv preprint arXiv:2403.06634*.

Varun Chandrasekaran, Kamalika Chaudhuri, Irene Giacomelli, Somesh Jha, and Songbai Yan. 2020. Exploring connections between active learning and model extraction. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1309–1326.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R Routledge, and 1 others. 2021. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711.

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022. Convfinqa: Exploring the chain of numerical reasoning in conversational finance question answering. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 6279–6292.

Xueqi Cheng, Minxing Zheng, Shixiang Zhu, and Yushun Dong. 2025. Misleader: Defending against model extraction with ensembles of distilled models. *arXiv preprint arXiv:2506.02362*.

Jacson Rodrigues Correia-Silva, Rodrigo F Berriel, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. 2018. Copycat cnn: Stealing knowledge by persuading confession with random non-labeled data. In *2018 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE.

Chengwei Dai, Minxuan Lv, Kun Li, and Wei Zhou. 2023. Meaeq: Mount model extraction attacks with efficient queries. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12671–12684.

Adam Dziedzic, Muhammad Ahmad Kaleem, Yu Shen Lu, and Nicolas Papernot. 2022. Increasing the cost of model extraction with calibrated proof of work. *arXiv preprint arXiv:2201.09243*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Xuanli He, Qiongkai Xu, Yi Zeng, Lingjuan Lyu, Fangzhao Wu, Jiwei Li, and Ruoxi Jia. 2022. Cater: Intellectual property protection on text generation apis via conditional watermarks. *Advances in Neural Information Processing Systems*, 35:5431–5445.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Matthew Jagielski, Nicholas Carlini, David Berthelot, Alex Kurakin, and Nicolas Papernot. 2020. High accuracy and high fidelity extraction of neural networks. In *29th USENIX security symposium (USENIX Security 20)*, pages 1345–1362.

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. *IEEE transactions on neural networks and learning systems*, 33(2):494–514.

Hengrui Jia, Christopher A Choquette-Choo, Varun Chandrasekaran, and Nicolas Papernot. 2021. Entangled watermarks as a defense against model extraction. In *30th USENIX security symposium (USENIX Security 21)*, pages 1937–1954.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Mika Juuti, Sebastian Szyller, Samuel Marchal, and N Asokan. 2019. Prada: protecting against dnn model stealing attacks. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 512–527. IEEE.

Sanjay Kariyappa and Moinuddin K Qureshi. 2020. Defending against model stealing attacks with adaptive misinformation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–778.

Kalpesh Krishna, Gaurav Singh Tomar, Ankur P Parikh, Nicolas Papernot, and Mohit Iyyer. 2019. Thieves on sesame street! model extraction of bert-based apis. *arXiv preprint arXiv:1910.12366*.

Thai Le, Noseong Park, and Dongwon Lee. 2021. A sweet rabbit hole by darcy: Using honeypots to detect universal trigger's adversarial attacks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3831–3844.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Zi Liang, Qingqing Ye, Yanyun Wang, Sen Zhang, Yaxin Xiao, Ronghua Li, Jianliang Xu, and Haibo Hu. 2024. Alignment-aware model extraction attacks on large language models.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-bert: Enabling language representation with knowledge graph. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 2901–2908.

Jacob Menick and 1 others. 2024. GPT-4o mini: Advancing cost-efficient intelligence. Technical report, OpenAI.

OpenAI. 2025. GPT-4.1 technical report. Technical report, OpenAI.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2019a. Knockoff nets: Stealing functionality of black-box models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4954–4963.

Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2019b. Prediction poisoning: Towards defenses against dnn model stealing attacks. *arXiv preprint arXiv:1906.10908*.

Hakan T Otal and M Abdullah Canbaz. 2024. Llm honeypot: Leveraging large language models as advanced interactive honeypot systems. In *2024 IEEE Conference on Communications and Network Security (CNS)*, pages 1–6. IEEE.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pages 248–260. PMLR.

Soham Pal, Yash Gupta, Aditya Shukla, Aditya Kanade, Shirish Shevade, and Vinod Ganapathy. 2020. Activethief: Model extraction using active learning and unannotated public data. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 865–872.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519.

Jeffrey Pawlick, Edward Colbert, and Quanyan Zhu. 2019. A game-theoretic taxonomy and survey of defensive deception for cybersecurity and privacy. *ACM Computing Surveys (CSUR)*, 52(4):1–28.

Wenjun Peng, Jingwei Yi, Fangzhao Wu, Shangxi Wu, Bin Bin Zhu, Lingjuan Lyu, Binxing Jiao, Tong Xu, Guangzhong Sun, and Xing Xie. 2023. Are you copying my model? protecting the copyright of large language models for eaas via backdoor watermark. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7653–7668.

Priyanka Sen, Sandeep Mavadia, and Amir Saffari. 2023. Knowledge graph-augmented language models for complex question answering. In *Proceedings of the 1st Workshop on Natural Language Reasoning and Structured Explanations (NLRSE)*, pages 1–8.

Anıl Sezgin and Aytuğ Boyacı. 2025. Decoypot: A large language model-driven web api honeypot for realistic attacker engagement. *Computers & Security*, 154:104458.

Shawn Shan, Emily Wenger, Bolun Wang, Bo Li, Haitao Zheng, and Ben Y Zhao. 2020. Gotta catch'em all: Using honeypots to catch adversarial attacks on neural networks. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 67–83.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, and 1 others. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Lance Spitzner. 2002. *Honeypots: tracking hackers*. Addison-Wesley Longman Publishing Co., Inc.

Jiashuo Sun, Chengjin Xu, Lumingyuan Tang, Saizhuo Wang, Chen Lin, Yeyun Gong, Lionel Ni, Heung-Yeung Shum, and Jian Guo. 2024. Think-on-graph: Deep and responsible reasoning of large language model on knowledge graph. In *International Conference on Learning Representations*, volume 2024, pages 3868–3898.

Tianxiang Sun, Yunfan Shao, Xipeng Qiu, Qipeng Guo, Yaru Hu, Xuan-Jing Huang, and Zheng Zhang. 2020. Colake: Contextualized language and knowledge embedding. In *Proceedings of the 28th international conference on computational linguistics*, pages 3660–3670.

Sebastian Szyller, Buse Gul Atli, Samuel Marchal, and N Asokan. 2021. Dawn: Dynamic adversarial watermarking of neural networks. In *Proceedings of the 29th ACM international conference on multimedia*, pages 4417–4425.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. 2016. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, pages 601–618.

Christoforos Vasilatos, Dunia J Mahboobeh, Hithem Lamri, Manaar Alam, and Michail Maniatakos. 2024. Llmpot: Automated llm-based industrial protocol and physical process emulation for ics honeypots. *arXiv preprint arXiv:2405.05999*.

Yixu Wang, Tianle Gu, Yan Teng, Yingchun Wang, and Xingjun Ma. 2025. Honeypotnet: Backdoor attacks against model extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 8087–8095.

Ziyang Wang, Jianzhou You, Haining Wang, Tianwei Yuan, Shichao Lv, Yang Wang, and Limin Sun. 2024. Honeygpt: breaking the trilemma in terminal honeypots with large language model. *arXiv preprint arXiv:2406.01882*.

Qiongkai Xu, Xuanli He, Lingjuan Lyu, Lizhen Qu, and Gholamreza Haffari. 2022. Student surpasses teacher: Imitation attack for black-box nlp apis. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2849–2860.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. Qa-gnn: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 535–546.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2022. Greaselm: Graph reasoning enhanced language models for question answering. *arXiv preprint arXiv:2201.08860*.

Zhanyuan Zhang, Yizheng Chen, and David Wagner. 2021. Seat: Similarity encoder by adversarial training for detecting model extraction attack queries. In *Proceedings of the 14th ACM Workshop on artificial intelligence and security*, pages 37–48.

Haodong Zhao, Jinming Hu, Peixuan Li, Fangqi Li, Jinrui Sha, Tianjie Ju, Peixuan Chen, Zhuosheng Zhang, and Gongshen Liu. 2024. Nsmark: Null space based black-box watermarking defense framework for language models. *arXiv preprint arXiv:2410.13907*.

Kaixiang Zhao, Lincan Li, Kaize Ding, Neil Zhenqiang Gong, Yue Zhao, and Yushun Dong. 2025. A survey on model extraction attacks and defenses for large language models. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6227–6236.

Xuandong Zhao, Lei Li, and Yu-Xiang Wang. 2022. Distillation-resistant watermarking for model protection in nlp. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5044–5055.

Lucia Zheng, Neel Guha, Brandon R Anderson, Peter Henderson, and Daniel E Ho. 2021. When does pretraining help? assessing self-supervised learning for law and the casehold dataset of 53,000+ legal holdings. In *Proceedings of the eighteenth international conference on artificial intelligence and law*, pages 159–168.

# A Theoretical Analysis

We provide formal guarantees for the two objectives in Section 2: benign-user harmlessness (Eq. (3)) and attacker budget waste (Eq. (2)).

## A.1 User Harmlessness

**Assumption 2** (User-HKG Separation)**.** *Let $\mathcal{P}_{\text{user}}$ denote the benign query distribution and $\mathcal{V}_H$ the set of HKG concepts. For all $v \in \mathcal{V}_H$:*

$$\mathbb{E}_{x\sim\mathcal{P}_{\text{user}}}[\text{sim}(x, v)] \leq \mu_0 \tag{7}$$

*where $\text{sim}(\cdot,\cdot)$ is cosine similarity in the sentence embedding space and $\mu_0 < \tau$.*

**Theorem 2** (User Harmlessness)**.** *Under Assumption 2, the probability that a benign session triggers the luring mechanism satisfies:*

$$\Pr_{x\sim\mathcal{P}_{\text{user}}}[\pi(x) = \text{lure}] \;\leq\; \exp\left(-\frac{2W(\tau-\mu_0)^2}{\lambda^2}\right) \tag{8}$$

*where $W$ is the sliding window size and $\lambda \in (0, 1]$.*

*Proof.* Under Assumption 2, the expected suspicion score $\mathbb{E}[s_t]$ for a benign query satisfies $\mathbb{E}[s_t] \leq \mu_0 < \tau$. The suspicion score is a convex combination of bounded random variables in $[0, 1]$. Applying Hoeffding's inequality over the $W$-query window:

$$\Pr[s_t > \tau] = \Pr[s_t - \mathbb{E}[s_t] > \tau - \mu_0] \tag{9}$$

$$\leq \exp\left(-\frac{2W^2(\tau-\mu_0)^2}{\sum_{i=1}^{W}(b_i-a_i)^2}\right) \tag{10}$$

where each term is bounded in an interval of length at most $\lambda$ (the mixing weight), giving the bound in Eq. (8). □

The failure probability decreases exponentially in $W(\tau-\mu_0)^2$. With $W = 10$ and $\tau - \mu_0 \geq 0.15$, the false-trigger probability is bounded by $e^{-4.5} < 0.012$, consistent with the empirical false trigger rates in Table 5.

## A.2 Attacker Budget Waste

The proof of Theorem 1 (stated in Section 3.5) is given below.

*Proof of Theorem 1.* At activation step $t_0$, the breadcrumb introduces $k$ HKG nodes into the frontier $\partial\mathcal{G}_a^{(t_0)}$. Under Assumption 1, HKG nodes have PPL at least as high as any real-domain node in the frontier. Therefore, at each subsequent step $t > t_0$, the probability of selecting an HKG node is at least:

$$p_t \;\geq\; \frac{|\partial\mathcal{G}_a^{(t)} \cap \mathcal{V}_H|}{|\partial\mathcal{G}_a^{(t)}|} \;\geq\; \frac{k}{|\partial\mathcal{G}_a^{(t_0)}| + k} \;=:\; p \tag{11}$$

where the second inequality holds because each HKG response adds $d_{\min} = 3 \geq k$ new HKG nodes to the frontier (by Section 3.2), so $|\partial\mathcal{G}_a^{(t)} \cap \mathcal{V}_H|$ is non-decreasing. The probability of drawing a real-domain node at every remaining step is therefore bounded by $(1-p)^{B-t_0}$, giving:

$$\mathbb{E}[\text{HKG fraction}] \;\geq\; 1 - (1-p)^{B-t_0} \tag{12}$$

Substituting $p = k/(|\partial\mathcal{G}_a^{(t_0)}| + k)$ yields Eq. (**??**). □

**Numerical illustration.** With $k = 3$, $|\partial\mathcal{G}_a^{(t_0)}| = 10$, and $B - t_0 = 20$:

$$\mathbb{E}[\text{HKG fraction}] \;\geq\; 1 - (10/13)^{20} \approx 0.993 \tag{13}$$

This represents an asymptotic best case. In practice, the empirical HKG fraction reaches 90.5% (Table 4), as the attacker occasionally selects real-domain frontier nodes.

## A.3 Asymmetric Cost Analysis

**Proposition 1** (Defender Cost Asymmetry)**.** *Let $C_{\mathcal{H}}$ be the one-time HKG construction cost and $c > 0$ be the per-query API cost. Then the ratio of attacker wasted budget to defender construction cost satisfies:*

$$\frac{c \cdot B \cdot \mathbb{E}[\textit{HKG fraction}]}{C_{\mathcal{H}}} \;=\; \Omega(B) \tag{14}$$

*The defender's advantage grows linearly in the attacker's budget $B$.*

This contrasts with watermarking, where the defender's cost scales with each model version, and with output perturbation, where the defender pays a permanent accuracy tax on every query.

# B Judge Model Ablation

In our main experiments, Agreement between surrogate $f_a$ and victim $f_v$ is evaluated by `gpt-4o-mini` as the judge model. To verify that results are not sensitive to this choice, we repeat the KT (Full) and No Defense conditions on the Medical domain at $B = 500$ using three alternative judges.

Table 7 reports Agreement under each judge. Results are consistent across all four judges: KT (Full) reduces Agreement by 7–9% relative to No Defense regardless of which model performs the evaluation, confirming that the defense effectiveness observed in Table 2 is not an artifact of the judge model selection.

Table 7: Agreement (%) on MedQA under different judge models. Medical domain, $B = 500$. Mean ± std over 5 seeds.

| Judge model | No Defense | KT (Full) |
|---|---|---|
| `gpt-4o-mini` (default) | 62.3±1.1 | 54.1±0.9 |
| `gpt-4o` | 63.1±0.9 | 54.8±1.0 |
| `claude-haiku-4-5` | 61.4±1.2 | 53.2±1.1 |
| `llama-3.1-70b` | 60.7±1.3 | 52.5±1.2 |

# C Probe Model and Transferability Filter

## C.1 Probe Model Details

The probe model $f_{\text{probe}}$ used in the two-stage transferability filter (Section 3.2, Eq. (4)) is LLaMA-3.1-8B-Instruct, identical in architecture to the surrogate model $f_a$. For each candidate item $h$, we construct 50 QA pairs containing $h$ and fine-tune $f_{\text{probe}}$ for $k = 3$ epochs with learning rate $2\times10^{-5}$, batch size 8, and linear warmup over the first 10% of steps. We then evaluate the fine-tuned model on the benchmark test set and compute the utility difference (Eq. (4)). The entire filtering pipeline processes one candidate in 4 minutes on a single A100 GPU.

We use the same base model as the surrogate to ensure a conservative filter: if fine-tuning on $h$ does not improve Agreement for $f_{\text{probe}}$, it is unlikely to improve Agreement for any comparable surrogate. The choice of 50 QA pairs balances signal strength against computational cost; in preliminary experiments, increasing to 100 pairs did not change the set of accepted nodes.

## C.2 Threshold Ablation

Table 8 reports the effect of varying the transferability filter threshold $\delta$ on HKG construction quality and downstream defense effectiveness on the Medical domain at $B = 500$.

At tight thresholds ($\delta \leq 0.01$), the keyword filter alone is insufficient to fill the HKG to the target size: fewer items pass the probe-model check, resulting in a smaller graph with fewer nodes available for entrapment. The reduced trapping depth causes Agreement to rise slightly compared to $\delta = 0.02$.

At loose thresholds ($\delta \geq 0.05$), items with marginal but non-zero downstream utility are admitted into the HKG. Although the graph remains full-sized, the surrogate benefits slightly from training on these borderline items, and Agreement increases as a result.

At $\delta = 0.02$, all retained nodes achieve zero measured utility while the HKG is large enough to sustain entrapment across the full budget range, yielding the lowest Agreement of $54.1 \pm 0.9$%. We therefore adopt $\delta = 0.02$ as the default in all experiments.

Table 8: Effect of transferability filter threshold $\delta$ on HKG node count and KT (Full) Agreement (%) on MedQA, $B = 500$. Mean ± std over 5 seeds.

| $\delta$ | Nodes passed | Utility (mean) | Agreement (%) |
|---|---|---|---|
| 0.005 | 1,182 | 0.000 | 56.3±1.2 |
| 0.010 | 1,344 | 0.000 | 55.1±1.1 |
| 0.020 | 1,500 | 0.000 | **54.1**±0.9 |
| 0.050 | 1,500 | 0.021 | 56.7±1.3 |
| 0.100 | 1,500 | 0.048 | 58.2±1.4 |

# D HKG Construction Cost

Table 9 reports the cost of constructing a 1,500-node HKG for each domain using `gemini-flash-2.5`.

Table 9: HKG construction cost per domain.

| | Medical | Financial | Legal |
|---|---|---|---|
| Candidates generated | 4,230 | 4,415 | 4,680 |
| Keyword filter pass | 78.3% | 75.1% | 72.6% |
| Probe filter pass | 45.3% | 44.8% | 43.1% |
| Final nodes | 1,500 | 1,500 | 1,500 |
| Tokens (generation) | 8.2M | 8.7M | 9.4M |
| Tokens (filtering) | 3.1M | 3.3M | 3.6M |
| Est. API cost (USD) | $4.52 | $4.80 | $5.20 |
| Wall-clock time (hrs) | 2.8 | 3.1 | 3.4 |

Construction cost scales modestly across domains. The Legal domain requires slightly more candidates due to lower probe-filter pass rates, reflecting the relative scarcity of clearly deployment-obsolete legal content compared to medical and financial domains.

# E Breadcrumb Template Diversity

We report diversity statistics and representative examples from the pool of 20 breadcrumb templates used during injection.

**Diversity statistics.** Self-BLEU-4: 0.31 (lower = more diverse). Average template length: $24.6 \pm 3.2$ tokens. Mean pairwise cosine similarity between template embeddings: 0.52.

**Template examples** (5 of 20 shown):

1. *"This mechanism shares similarities with **[C]**, a [domain-adjacent framework] that offers a complementary perspective on the underlying principles."*
2. *"A related line of inquiry involves **[C]**, which provides additional context on the underlying dynamics."*
3. *"For a broader view, **[C]** addresses a parallel aspect of this phenomenon from a historical standpoint."*
4. *"Practitioners have also noted connections to **[C]**, a framework that informed earlier approaches to this problem."*
5. *"The foundational principles here bear resemblance to those described in **[C]**, which examines the topic through an adjacent theoretical lens."*

where **[C]** denotes the selected HKG concept. The full list of 20 templates is available in our code repository.

# F Legal Domain HKG Details

The Legal HKG is constructed following the same pipeline as Medical and Financial (Section 3.2). Legal HKG concepts are drawn from three categories:

- **Historical minutiae**: superseded legal doctrines and repealed statutes (e.g., common-law forms of action, pre-UCC negotiable instruments law).
- **Peripheral taxonomy**: domain-adjacent classification systems with no bearing on modern holdings (e.g., historical court hierarchy changes, obsolete jurisdictional categories).
- **Niche procedural records**: technically correct but task-irrelevant procedures (e.g., pre-electronic case filing protocols, historical jury selection procedures).

Legal HKG attractiveness (0.947) is slightly lower than Medical (0.959) and Financial (0.961), reflecting the comparatively smaller surface overlap between obsolete legal terminology and modern holding-identification vocabulary. Despite this, the defense achieves a 6.2% Agreement reduction on CaseHOLD, consistent with the Medical domain pattern.

Table 10: Legal HKG statistics.

| Property | Legal |
|---|---|
| Nodes $\|\mathcal{V}_H\|$ | 1,500 |
| Edges $\|\mathcal{E}_H\|$ | 4,500 |
| Out-degree (all nodes) | 3 |
| Internal edge ratio | 100% |
| Utility (mean / max) | 0.000 / 0.000 |
| Attractiveness (mean / min) | 0.947 / 0.530 |
| Inaccessibility (mean / min) | 0.931 / 0.695 |
| Historical minutiae | 512 (34.1%) |
| Peripheral taxonomy | 478 (31.9%) |
| Niche procedural | 510 (34.0%) |

# G Related Work

## G.1 Model Extraction Attacks

Model extraction attacks (MEAs) were formalized by Tramèr et al. (Tramèr et al., 2016), who demonstrated that models exposed through prediction APIs can be reconstructed through systematic querying. Subsequent work extended MEAs to deep neural networks in computer vision, using surrogate training on API outputs (Papernot et al., 2017; Correia-Silva et al., 2018; Orekondy et al., 2019a). Krishna et al. (Krishna et al., 2019) showed that BERT-based APIs can be extracted with high fidelity by fine-tuning on victim-generated labels, while Xu et al. (Xu et al., 2022) demonstrated that the surrogate can even surpass the victim on certain benchmarks through careful query selection. Recent studies have targeted production-scale LLMs: Birch et al. (Birch et al., 2023) introduced model leeching attacks against instruction-following LLMs, and Carlini et al. (Carlini et al., 2024) showed that partial extraction of production language models is feasible with moderate API access. Liang et al. (Liang et al., 2024) further demonstrated alignment-aware extraction that preserves both capability and safety properties of the victim model.

A central challenge in MEAs is improving query efficiency under limited budgets. Active-learning methods prioritize informative queries to maximize extraction fidelity per API call. ActiveThief (Pal et al., 2020) uses uncertainty sampling on unlabeled public data; Jagielski et al. (Jagielski et al., 2020) combine semi-supervised learning with strategic querying to achieve high-accuracy extraction; Chandrasekaran et al. (Chandrasekaran et al., 2020) formally connect active learning and model extraction through a shared information-theoretic framework; and MEAEQ (Dai et al.,

2023) introduces PPL-based query selection with entity expansion for LLM extraction. In contrast to prior work that treats query efficiency as an attack objective, our work treats it as the attacker's primary vulnerability.

### G.2 Defenses Against Model Extraction

**Output perturbation.** These defenses degrade surrogate training by perturbing API outputs or increasing extraction cost. Orekondy et al. (Orekondy et al., 2019b) proposed prediction poisoning, which maximizes the angular deviation between the true and perturbed output gradients. Kariyappa and Qureshi (Kariyappa and Qureshi, 2020) introduced adaptive misinformation, which selectively perturbs outputs for queries near the decision boundary. Dziedzic et al. (Dziedzic et al., 2022) proposed calibrated proof-of-work schemes that increase the computational cost of querying without altering outputs. All these approaches face an inherent accuracy–security trade-off: stronger perturbation harms benign utility (Zhao et al., 2022), making them difficult to deploy in quality-sensitive settings.

**Watermarking.** Watermarking-based defenses embed identifiable signals into model outputs for post-hoc ownership verification. Jia et al. (Jia et al., 2021) proposed entangled watermarks that are preserved through distillation; Zhao et al. (Zhao et al., 2022) introduced distillation-resistant watermarking for NLP models; He et al. (He et al., 2022) developed conditional watermarks for text generation APIs; Szyller et al. (Szyller et al., 2021) proposed dynamic adversarial watermarking that adapts to the extraction process; Peng et al. (Peng et al., 2023) embedded backdoor-based watermarks for EaaS copyright protection; and Zhao et al. (Zhao et al., 2024) introduced null-space watermarking as a black-box defense. While effective for attribution and ownership claims, watermarking does not interfere with the extraction process itself: the surrogate model has already been trained by the time the watermark is detected.

**Query detection.** Detection-based defenses identify suspicious queries through statistical or representation-level analysis. PRADA (Juuti et al., 2019) detects extraction attacks by monitoring the distribution of incoming queries for anomalous patterns. SEAT (Zhang et al., 2021) uses adversarial training to learn a similarity encoder that distinguishes extraction queries from benign ones. MISLEADER (Cheng et al., 2025) combines detection with active defense by deploying an ensemble of distilled models. However, modern MEAs increasingly use task-relevant, in-distribution queries that resemble normal usage (Dai et al., 2023; Liang et al., 2024), making reliable detection increasingly difficult as extraction techniques mature.

### G.3 Honeypot-Based Defenses in Machine Learning

Honeypots redirect attacker effort toward decoy targets rather than blocking attacks directly, a strategy with deep roots in network security (Spitzner, 2002; Pawlick et al., 2019). Recent work has explored LLM-powered honeypot systems for network defense: Otal and Canbaz (Otal and Canbaz, 2024) used LLMs to generate realistic honeypot interactions; Wang et al. (Wang et al., 2024) proposed HoneyGPT for terminal honeypots; Sezgin and Boyacı (Sezgin and Boyacı, 2025) developed DecoyPot for web API emulation; and Vasilatos et al. (Vasilatos et al., 2024) introduced LLMPot for industrial control system honeypots.

In machine learning, prior honeypot-style defenses operate at the output or parameter level. Shan et al. (Shan et al., 2020) embed adversarial triggers that cause misclassification when the extracted model is deployed. Le et al. (Le et al., 2021) plant trapdoor patterns that can be detected in the surrogate to prove extraction. HoneypotNet (Wang et al., 2025) injects backdoor layers that activate only in extracted copies, enabling post-hoc detection. All these approaches focus on detecting or attributing extraction *after* it succeeds; none intervenes in the knowledge acquisition process during extraction. Our work fills this gap by redirecting the attacker's knowledge exploration trajectory in real time.

### G.4 Knowledge Graphs for Language Model Augmentation

Knowledge graphs (KGs) structure entities and relations to support retrieval and reasoning over factual knowledge (Ji et al., 2021). Prior work has used KGs to augment language models through multiple paradigms.

**Direct integration.** K-BERT (Liu et al., 2020) injects knowledge graph triples into the input layer of BERT, while CoLAKE (Sun et al., 2020) jointly learns contextualized language and knowledge embeddings. These approaches improve performance

on knowledge-intensive tasks but require modifying the model architecture.

**Retrieval-augmented generation.** RAG (Lewis et al., 2020) retrieves relevant passages from a knowledge source at inference time, avoiding the need to store all knowledge in model parameters. QA-GNN (Yasunaga et al., 2021) combines language models with graph neural networks over knowledge graphs for question answering. GreaseLM (Zhang et al., 2022) fuses language model representations with KG reasoning through cross-modal attention layers.

**Graph-guided reasoning.** Think-on-Graph (Sun et al., 2024) showed that step-by-step KG traversal enables deep multi-hop reasoning in LLMs. Sen et al. (Sen et al., 2023) demonstrated that KG augmentation improves complex question answering by providing structured evidence chains.

Our work repurposes the KG not as a source of useful knowledge but as an organized space of non-transferable knowledge. The internal coherence and connectivity of the graph structure—the same properties that make KGs effective for augmenting LLMs—are what make the HKG attractive to active-learning attackers, turning a tool for knowledge enhancement into a tool for knowledge-space defense.