ProFraudGuard: Proactive Adversarial Fine-Tuning of Fraud Detectors with Inverse Reinforcement Learning

Chirag Singh
Amazon
chirsi@amazon.com

Jitin Kumar
Amazon
jitinkr@amazon.com

Arunkalyan Nagarajan
Amazon
arunnaga@amazon.com

Abstract

E-commerce stores face an evolving challenge in detecting fraudulent business registrations, as sophisticated actors continuously adapt their techniques to create deceptive accounts, rendering conventional post-registration measures increasingly inadequate. Real-time detection at the point of registration is further complicated by cold-start constraints and strict sub-second latency requirements. This paper introduces ProFraudGuard, a novel system designed for proactive, real-time fraud detection. It employs a closed-loop adversarial training paradigm using an Adversarial Large Language Model Generator and a Risk Detector to continuously adapt to evolving fraud tactics. Central to ProFraudGuard is the novel Proactive Adversarial Fine-Tuning (PAFT) algorithm, designed to ensure stable co-evolution of system components with theoretical convergence guarantees. Experimental results demonstrate that ProFraudGuard significantly outperforms existing methods, achieving a 0.91 Precision at the top 10% risk scores and an AUPRC of 0.59. The system uncovered 21.51% more fraud cases that evaded prior detection, offering a potential 17.6% improvement in annualized savings. By proactively learning from simulated fraudster behavior, ProFraudGuard equips detection systems to anticipate and neutralize future threats.

CCS Concepts

• **Security and privacy → Intrusion/anomaly detection and malware mitigation**; • **Computing methodologies → Adversarial learning**; **Inverse reinforcement learning**.

Keywords

Fraud Detection, Adversarial Machine Learning, Large Language Models, Inverse Reinforcement Learning, E-commerce Security

ACM Reference Format:
Chirag Singh, Jitin Kumar, and Arunkalyan Nagarajan. 2026. ProFraudGuard: Proactive Adversarial Fine-Tuning of Fraud Detectors with Inverse Reinforcement Learning. In *Proceedings of KDD 2026 Workshop on AI for Fraud and Abuse (KDD 2026 Workshop).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

KDD 2026 Workshop, Jeju, Korea

ACM ISBN 978-1-4503-XXXX-X/2026/08
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

E-commerce stores face escalating registration fraud, where malicious actors use impersonation, multi-account abuse, and security circumvention to exploit policies and gain unfair advantages. These tactics directly harm legitimate customers through reduced product availability and a degraded shopping experience. Consequently, the inherent delays of traditional post-registration detection highlight the urgent need for effective real-time screening integrated directly into the registration process.

Registration fraud detection faces a unique 'cold-start' problem: at the moment of account creation, only minimal information like business name and email address is available, with no behavioral history to analyze. Moreover, these assessments must be completed within strict sub-second latency constraints to maintain a seamless user experience. Concurrently, attackers increasingly leverage Generative AI to create convincing synthetic identities that evade traditional detection methods. This creates a fundamental trade-off between efficiency and effectiveness; strict latency constraints necessitate lightweight models that often lack the robustness and predictive power of their larger counterparts, creating vulnerabilities that sophisticated attackers can exploit. Reinforcement learning (RL) offers a promising avenue to address this by enabling even constrained models to continuously enhance their predictive capabilities through domain-specific optimization, developing robust fraud detection by learning directly from real-world experiences.

Our work, ProFraudGuard, operates within the "era of experience" paradigm [14], learning directly from system interactions rather than static datasets to circumvent the limitations of lightweight models. Critically, RL enables reasoning about fraud domain information absent from Large Language Model (LLM) pretraining data, expanding the model's capabilities into previously unexplored detection territory. By employing an adversarial learning mechanism, the system proactively counters novel attack vectors without manual labeling, effectively transitioning to an anticipatory security posture while maintaining defensive effectiveness and real-time performance. The key contributions of this work are as follows:

1. Adversarial IRL Formulation: We frame proactive fraud detection as a novel joint adversarial learning problem, guided by Inverse Reinforcement Learning (IRL) principles to model adaptive adversaries.

2. The PAFT Algorithm: We introduce **Proactive Adversarial Fine-Tune (PAFT)**, an alternating fine-tuning algorithm that concurrently trains the risk detector against adversarial samples and adapts the generator policy via detector feedback.

3. Theoretical & Empirical Validation: We provide a theoretical guarantee of PAFT's convergence, empirically validated by diminishing detector gradient norms. Experiments on real-world

registration data demonstrate its robustness, achieving 0.91 Precision@Top10% and 0.59 AUPRC.

2 Related work

The detection of sophisticated fraud has progressed significantly from static rules to adaptive machine learning systems, particularly for real-time scenarios like account registration where initial data is scarce. Early ML methods typically relied on similarity metrics (e.g., Levenshtein distance) or embedding-based comparisons (e.g., cosine similarity) to identify fraudulent accounts based on proximity to known bad actors [1]. Subsequently, deep learning architectures, specifically Transformers, revolutionized this space by capturing complex, sequential patterns within tabular data [13, 21]. However, these supervised methods remain inherently reactive; they struggle to generalize to novel tactics until sufficient attack instances are collected for retraining. This limitation is exacerbated by two key challenges in registration fraud: the "cold-start" problem, where the absence of behavioral history hinders the identification of new bad actors, and strict latency constraints that restrict the deployment of computationally intensive models. Consequently, despite advances in multi-modal [12] and graph-based analysis, reliance on historical patterns leaves systems vulnerable to attacks designed to appear benign at the point of entry. Large Language Models (LLMs) have introduced new defensive capabilities, ranging from Retrieval-Augmented Generation (RAG) for policy checks and real-time conversation monitoring [15] to synthetic data generation that addresses class imbalances. While controlled text generation allows for fine-grained simulation of fraud patterns, effective deployment depends heavily on labor-intensive prompt tuning and ongoing risk analysis to ensure relevance. To address this, recent alignment techniques aim to incorporate domain-specific knowledge for realistic fraud generation. Approaches such as Supervised Fine-Tuning (SFT) [4, 17] and Reinforcement Learning from Human Feedback (RLHF) [3] enable targeted, adaptive generation. The efficacy of reward learning on preference datasets has been demonstrated across diverse tasks, including text summarization [6], story generation [23], and instruction following [9]. Typically, RLHF fine-tunes a language model by training a reward model on human preference data, which then guides policy optimization using algorithms like REINFORCE [19] or Proximal Policy Optimization (PPO) [10]. However, RLHF's reliance on collecting expensive, static human preference data has driven a shift toward implicit or weakly supervised alignment methods. Innovations like Direct Preference Optimization (DPO) [8] and Group Relative Policy Optimization (GRPO) [11] bypass explicit reward modeling entirely, while self-rewarding models like SPIN [2] utilize synthetic data to reduce labeling costs. Concurrently, Inverse Reinforcement Learning (IRL) [7, 22], particularly Maximum Entropy frameworks (ML-IRL), offers a principled approach to learn reward functions directly from expert demonstrations without explicit preferences. Unlike behavior cloning, IRL seeks rewards that best explain observed expert behavior, often yielding superior performance in sequential decision-making processes [5].

In the context of dynamic fraud detection, the primary challenge lies in adapting to evasive tactics rather than merely classifying historical data. Rather than relying on static labels or predefined attacks, this work leverages IRL principles to extract dynamic reward signals from the detector's real-time responses. These signals reflect the detector's vulnerabilities, guiding the generator to optimize for high-reward (i.e., highly evasive) behaviors. This creates a crucial adaptive feedback loop where the generator learns to uncover sophisticated fraud strategies, and the detector improves by training on these evolving adversarial examples. To address the inherent instability of such adversarial co-evolution, we introduce a novel algorithm, **Proactive Adversarial Fine-Tuning (PAFT)**, which enables robust adaptation with theoretical convergence guarantees. The following sections detail ProFraudGuard and its components.

3 ProFraudGuard architecture

ProFraudGuard marks a paradigm shift from reactive to proactive fraud detection by via a closed-loop feedback system that continually anticipates and adapts to evolving threats (Figure 1). Registration Service collects registration form data having business names, email addresses and demands instant risk evaluation for these incoming registration attempts. Real-time registration streams drive the framework's three interdependent components to form a self-strengthening cycle:

1. Risk Detector (D_ϕ): A latency-optimized fraud assessment engine for real-time scoring of user registrations. Its architecture unifies a DistilBERT backbone with a parallel numerical branch, enhanced by a Feature Generator that employs k-Nearest Neighbors (KNN) to extract velocity, variety, and text-based signals. This setup captures both explicit token-level patterns and implicit entity relationships. The detector undergoes adversarial fine-tuning on both real-world observations and challenging future threat patterns generated synthetically to maintain robustness against constantly evolving fraud tactics. Formally expressed as D_ϕ with parameters ϕ, the detector transforms input registration data into a risk score.

2. The Adversarial LLM Generator (G_ψ): Built on an LLM foundation, G_ψ takes input prompts z to synthesize sophisticated registration data to probe D_ϕ's weaknesses. The generator adapts its behavior based on feedback from the detector, learning from previous unsuccessful attempts to devise new logical attack strategies that result in realistic yet targeted fraudulent registration patterns, mimicking fraudster tactics. Through an iterative process of generating samples and receiving feedback, G systematically explores the fraud spectrum to devise logical, evasive attack strategies. By providing reasoning behind potential fraud strategies explored, G offers valuable insights into potential attacker logic and the factors that contribute to an attempt's evasiveness. In this paper, we fix the LLM architecture and focus on the generator's function and role within our adversarial fine-tuning process rather than the architecture design.

3. The Reward Formulator (R): This mechanism governs the adversarial interplay by calculating a multi-objective reward signal. It incentivizes G_ψ not only for successful evasion (low detector scores) but also for maintaining structural validity (through formatting rewards) , creating abusive traits (via divergence penalties), and introducing novel evasion tactics (using novelty rewards). This combination ensures that the generator produces realistic, diverse, and hard-to-detect fraud patterns, enhancing the detectors's capability.

ProFraudGuard functions as a closed, adaptive loop centered on real-time registration assessment. The Risk Detector scores incoming requests, with identified patterns feeding an adversarial Generator. Guided by a reward mechanism, the Generator creates synthetic fraud examples. Critically, this process trains the Generator to simulate the *likely next tactics fraudsters would employ after being detected*, proactively preparing the Detector to identify these threats before they fully manifest in real-world attacks. During inference, the system decouples these components: the Detector is deployed to evaluate registrations in real-time, while the Generator operates strictly offline to continuously retrain the Detector, establishing a self-improving cycle that anticipates emerging threats rather than merely reacting to known patterns.

4 Problem description

Maintaining robust risk detection in real-time, cold-start scenarios under dynamic and adversarial fraud conditions presents a significant challenge. This problem can be framed as a bilevel optimization problem. In this setup, a detection model is jointly trained with an adaptive adversarial data generation policy guided by inverse reinforcement learning. Rather than relying on an *a priori* engineered reward, the core evasion signal is dynamically inferred from the detector's evolving parameters, successfully recovering the reward structure from the data while treating novelty and formatting components strictly as structural constraints. Let $\mathcal{X} \subseteq \mathbb{R}^d$ represent the feature space of registration requests and $\mathcal{Y} \in \{0, 1\}$ the binary outcome space (0 for legitimate, 1 for fraudulent). The goal is to learn and continually refine a risk detector $D_\phi : \mathcal{X} \to [0, 1]$, which outputs a risk score representing the estimated probability of fraud. This is achieved by finding mutually optimal parameters (ϕ, ψ) for detector and generator via Maximum Likelihood Inverse Reinforcement Learning (ML-IRL) formulation.

Lower-Level Optimization (Generator): In inner loop, for a fixed detector D_ϕ, the generator $\pi_{G,\psi}$ acts as follower. It maps latent samples $z \sim P_z$ to synthetic feature vectors x_{gen}, optimizing its parameters ψ to maximize a detector-derived reward $R(x; \phi)$. We impose Kullback-Leibler (KL) divergence penalty against a reference policy π_{ref}. Optimal generator parameters $\psi^*(\phi)$ are given by:

$$\arg\max_{\psi} \mathbb{E}_{z\sim P_z} \left[R(G_\psi(z); \phi) - \beta D_{KL}(\pi_{G,\psi} \| \pi_{ref}) \right]$$

where $\beta > 0$ is a regularization coefficient. Here, $R(x; \phi)$ is a function of risk score $s = D_\phi(x)$.

Upper-Level Optimization (Detector):In outer loop, the detector D_ϕ acts as leader, minimizing classification loss over both the empirical distribution of real requests P_{real} and the induced distribution of adversarial samples P_{gen} generated by $\psi^*(\phi)$. The objective is:

$$\min_{\phi} (\mathbb{E}_{(x,y)\sim P_{real}} [\mathcal{L}_{det}(D_\phi(x), y)] + \mathbb{E}_{z\sim P_z} [\mathcal{L}_{det}(D_\phi(G_{\psi^*(\phi)}(z)), 1)])$$

where $\mathcal{L}_{det}$ is a standard loss function such as binary cross-entropy. Crucially, generated samples x_{gen} are treated as fraudulent (i.e. $y = 1$) during training, encouraging the detector to recognize and defend against these challenging synthesized examples. Next section elaborates the ProFraudGuard methodology used to implement this robust risk detection.

5 Methodology

To address the fundamental inadequacy of static supervised learning against continuously evolving fraud strategies, ProFraudGuard moves beyond passive historical analysis to establish a dynamic, adversarial interaction between a risk detector and an adaptive generator. We detail this joint learning setup below.

5.1 Joint learning setup

Directly solving the bilevel optimization problem is computationally prohibitive due to the requirement for second-order derivatives. We reformulate this via ML-IRL into a tractable minimax optimization problem $\max_\phi \min_\psi V(\phi, \psi)$, a strategy widely applied to hierarchical objectives [16, 18, 20]. The value function $V(\phi, \psi)$ reflects the competing objectives by contrasting expected rewards and regularizing the policy:

$$V(\phi, \psi) = -L_{det}(\phi) - (R_{gen}(\psi, \phi) - K_\psi)$$

Here, $L_{det}(\phi) = \mathbb{E}_{(x,y)\sim P_{real}}[\mathcal{L}_{det}(D_\phi(x), y)]$ captures the detector's primary goal of minimizing error on real labeled data. Simultaneously, $R_{gen}(\psi, \phi) = -\mathbb{E}_{z\sim P_z}[R(G_\psi(z); \phi)]$ represents the adversarial goal: maximizing this term (minimizing the negative) drives the generator to maximize its expected reward R. This contrastive structure mirrors techniques used in RLHF and enables the detector to effectively distinguish preferred (legitimate) behaviors from undesirable (fraudulent) ones. The regularization term $K_\psi = -\beta D_{KL}(\pi_{G,\psi} \| \pi_{ref})$ ensures the generator policy remains close to a reference policy π_{ref} (controlled by $\beta > 0$), enhancing training stability.

5.1.1 Generator policy optimization. To solve the inner minimization problem $\min_\psi V(\phi, \psi)$, we employ GRPO. It allows fine-tuning the generator G_ψ directly on inferred rewards from D_ϕ without requiring a separate critic model. Specifically, for a given input context z, GRPO samples a group of M outputs $\{x^1_{gen}, \ldots, x^M_{gen}\}$ and updates the policy to increase the likelihood of higher-reward samples relative to the group average. The objective function is:

$$J(\psi) = \mathbb{E}_{z, x^i_{gen}} [\frac{1}{M} \sum_{i=1}^{M} (\sum_{t=1}^{|x^i_{gen}|} L_{i,t} - K_{\psi,i})] \quad (1)$$

where $L_{i,t} = \min(\rho_{i,t}\hat{A}_{i,t}, \text{clip}(\rho_{i,t}, 1-\epsilon, 1+\epsilon)\hat{A}_{i,t})$. Here, $\rho_{i,t}$ is the probability ratio between the new and old policies, and $\hat{A}_{i,t}$ is the advantage estimate derived by comparing the total reward $R(x^i_{gen}; \phi)$ against the group mean. This incentivizes the generator to refine strategies based on detector feedback, producing samples that maximize the reward R by successfully evading D_ϕ.

5.1.2 Reward function design. The behavior of the adversarial generator $\pi_{G,\psi}$ is governed by the multi-objective reward function $R(x_{gen}; \phi)$ it seeks to maximize. The design of this reward is critical; it must align the generator toward producing effective, relevant, and structurally valid adversarial examples. We define this function

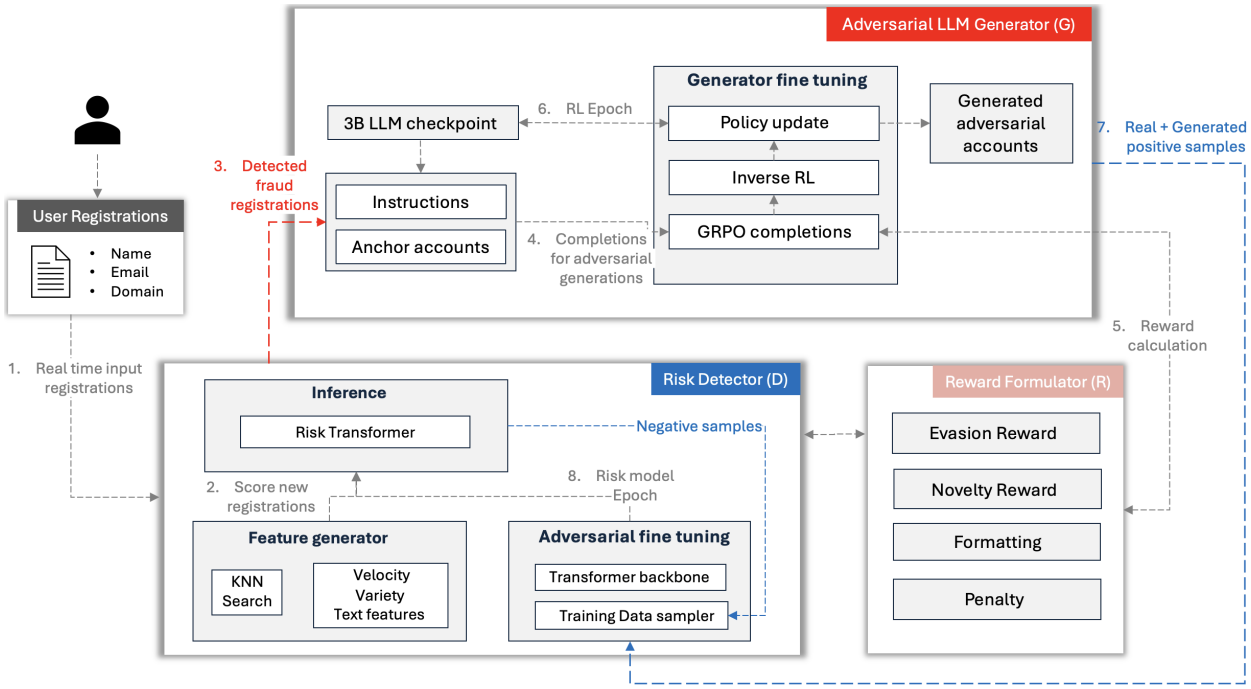

Figure 1: ProFraudGuard System Architecture

as a weighted combination of four distinct components:

$$R(x_{gen}) = w_1 \cdot R_d + w_2 \cdot R_n + w_3 \cdot R_f + w_4 \cdot R_p$$

where coefficients w_i balances the influence of each objective. First, the Evasion Reward ($R_d = -log(D_\phi(x_{gen}))$) serves as the primary adversarial signal, assigning higher values to samples that receive lower risk scores from the detector, thereby explicitly encouraging the discovery of patterns that bypass current detection logic. To ensure the generator explores beyond known attacks rather than merely replicating them, the Novelty Reward ($R_n = 1 - \max_{x' \in X_{hist}} \text{sim}(x_{gen}, x')$) penalizes similarity to historical data X_{hist} (e.g., via cosine similarity), forcing the production of distinct, previously unseen fraud patterns that test the detector's generalization boundaries. Concurrently, the Formatting Reward (R_f) enforces structural validity, imposing constraints to ensure generated samples adhere to expected input formats, making them easier to extract. Finally, to prevent trivial solutions where the generator simply copies safe user behavior to achieve low risk scores, a Penalty (R_p) is applied; this negative term discourages the imitation of known legitimate patterns, ensuring the generator creates active, evasive fraud attempts rather than benign samples that offer no training value.

5.2 Proactive Adversarial Fine Tune: PAFT

To solve the reformulated minimax problem, we introduce Proactive Adversarial Fine-Tuning (PAFT). This iterative algorithm employs a nested alternating optimization strategy that drives the detector D_ϕ and generator $\pi_{G,\psi}$ toward an adversarial equilibrium by alternating between detector updates and generator refinement based on feedback.

Algorithm 1: PAFT

Input: Initialize detector parameter (ϕ_0), generator policy $\pi^0_{G,\psi}$, detector learning rate (η_ϕ), batch sizes m_z, m_r and H, N as outer and inner iterations

Initialize: $\phi_{-1,N} \leftarrow \phi_0$;

for $h = 0$ **to** $H - 1$ **do**

  Assign $\phi_{h,0} \leftarrow \phi_{h-1,N}$;

  Data Sample: m_z samples from $z_{h,n} \sim P_z$, m_r adversarial $X^h_{a,n} = \pi^h_{G,\psi}(\cdot|z_{h,n})$ and real $(X^h_{r,n}, Y^h_{r,n}) \sim P_{real}(X, Y)$ samples **for** $n = 0$ **to** $N - 1$ **do**

    Estimate gradient $\hat{g}_{h,n}$ w.r.t ϕ via:

$$-\frac{1}{m_r} \sum_{(x,y) \in (X^h_{r,n}, Y^h_{r,n})} \nabla_\phi \mathcal{L}_{\text{det}}(D_{\phi_{h,n}}(x), y)$$

$$-\frac{1}{m_z} \sum_{x_a \in X^h_{a,n}} \nabla_\phi \mathcal{R}(x_a; \phi_{h,n});$$

    $\phi_{h,n+1} \leftarrow \phi_{h,n} + \eta_\phi \hat{g}_{h,n}$;

  Policy Alignment: Update optimal $\pi^h_{G,\psi} \propto \mathcal{R}(x_a; \phi_{h,N})$ with Equation 1;

Output: Optimized $D_{\phi_{H,N}}$, policy $\pi^H_{G,\psi}$

The nested structure of PAFT is critical for stability. The inner loop aligns the detector to robustly identify current generator's adversaries. The outer loop then updates the generator to produce novel tactics that challenge the refined detector. To balance computational efficiency with learning dynamics, we decouple sampling frequency

from update steps. Since ideal online sampling (N=1) is computationally prohibitive, we sample adversarial data once per outer iteration (H) and reuse it for N subsequent detector updates. This strategy minimizes overhead while ensuring the detector converges on robust features before adversarial strategy shifts.

5.3 Convergence theory

Convergence is crucial for PAFT given the non-stationary adversarial environment, where detector updates continuously shift the reward landscape for the generator, and vice versa. To address this instability, we provide a theoretical analysis of the PAFT algorithm (detailed proofs in Appendix A).

Theorem 5.1. *Under the assumptions A, for PAFT algorithm 1 with $\eta_\phi = \Theta(1/\sqrt{HN})$, the minimum expected squared norm of the gradient of the detector's objective function $L^*(\phi)$ is bounded as:*

$$\min_{h,n} \mathbb{E}[\|\nabla L^*(\phi_{h,n})\|^2] \leq O\left(\frac{1}{\sqrt{HN}} + \frac{1}{H}\right)$$

where $h \in \{0, .., H-1\}$ and $n \in \{0, .., N-1\}$. Theorem 5.1 confirms that PAFT converges to a stationary point of the detector's objective $L^*(\phi)$ as iterations increase. While reusing adversarial samples for N steps introduces distribution shift, our analysis yields a convergence rate of $O(1/\sqrt{HN} + 1/H)$, proving that this accumulated bias remains strictly bounded. This result provides critical theoretical validation for our efficiency strategy: generating expensive adversarial samples only once per outer loop allows for reliable convergence without the computational cost of continuous resampling.

6 Experiments and results

In this section, we analyze the impact of Proactive Adversarial Fine-Tuning (PAFT) (Algorithm 1) on risk detection performance compared to static baselines, demonstrating that dynamic feedback improves robustness without relying on manually investigated abuse data. We further empirically validate the convergence stability of our bilevel optimization and analyze how specific reward components contribute to effective adversarial generation.

6.1 Experiment setup

Dataset: While numerous public datasets exist for general fraud detection, none capture our specific setting of proactively anticipating registration fraud under strict cold-start constraints. Existing fraud benchmarks typically assume the availability of rich behavioral histories, which are entirely absent at the point of registration. Therefore, to ensure a rigorous evaluation, we curated multiple specialized datasets. We used multiple datasets for training and evaluation. The primary dataset for initial model training was labeled historical registration dataset ($\mathcal{H}$), consisting of 200K sampled records over two years. $\mathcal{H}$ includes business names, email addresses, and derived numerical and textual features. To simulate adversarial behavior, we used a pre-trained LLM (3B parameter) with a structured prompt to generate approximately 50 K synthetic fraud patterns, which were combined with observed three months of incoming registrations to form dataset $\mathcal{S}$, reflecting a realistic fraud distribution. Separately, we fine-tuned a generator policy using PAFT to create more challenging adversarial examples, which were merged with incoming registrations to create dataset $\mathcal{A}$. To ensure temporal robustness and prevent data leakage, we evaluated the system using an investigator-labeled sampled dataset $\mathcal{E}$ (approx. 18K). This dataset covers one month of registration activity drawn from a strictly future time window, ensuring absolutely no overlap with the training data.

Implementation details: We implemented three different models for comparison. The initial static risk detector, D_s, was trained using the historical dataset $\mathcal{H}$. To establish a baseline using static adversarial data, we then incrementally trained D_s on dataset $\mathcal{S}$ to obtain an updated risk detector, D_t. We specifically omit graph-based and semi-supervised methods as baselines; these approaches require rich behavioral histories that are inherently unavailable at the point of registration, where only limited features are present. The PAFT algorithm was used for joint fine-tuning of D_s (as the initial detector) and the LLM generator, G_f, to obtain the final detector, D_f. This joint training involved $H = 6$ outer loops, where the detector was updated with batch size 128 over $N = 4016$ steps per loop using dataset $\mathcal{A}$. The generator G_f was fine-tuned alongside using a batch size of 16 on NVIDIA V100 GPUs. Reward Formulator used weighted components tuned to emphasize risk detection signals on a validation set. Crucially, to accommodate computational costs and strict real-time requirements, the systems are decoupled in practice. The PAFT loop runs entirely offline (end-of-day batch process), and the LLM generator is never deployed during serving. Only the lightweight DistilBERT detector serves online, successfully evaluating incoming registrations within sub-second latency.

6.2 Results

Given the nature of real-world fraud detection - characterized by high class imbalance and the absence of comprehensive population labels - we evaluate these models primarily based on Precision at the Top 10% of risk scores and Area Under the Precision-Recall Curve (AUPRC) on the investigator-labeled dataset $\mathcal{E}$ (Table 1). The detector jointly trained using PAFT, D_f, demonstrates a notable improvement across all metrics compared to the baselines. Specifically, D_f achieves a Precision@Top10% of 0.91 and an AUPRC of 0.59, representing a significant gain over the static pre-trained detector D_s (0.86 Precision, 0.47 AUPRC). Conversely, the detector trained with static adversarial data (D_t) exhibited a performance degradation (0.83 Precision, 0.38 AUPRC). Analysis of the generator's output reveals that the static LLM used for D_t produced less reliable adversarial samples, often with formatting issues and lacking novelty, which contributed to the degradation of D_t's performance. In contrast, the generator trained jointly within PAFT received appropriate dynamic feedback, resulting in generated adversarial attacks that were novel, challenging, and structurally valid. This confirms the necessity of multi-objective reward function: the formatting reward ensures syntactic validity, resolving structural issues, while the novelty reward prevents mode collapse by penalizing similarity to historical data. Furthermore, the evasion reward drives the discovery of blind spots, and the penalty term prevents replication of benign patterns, ensuring the generated samples remain genuinely adversarial and effective training targets.

A critical challenge in bilevel optimization is ensuring stability. To evaluate this, we rely on Theorem 5.1, which posits that the

Table 1: Experiment Results on Registration dataset

Method	Pr@Top10%	AUPRC
Static Pre-trained Risk Detector (D_s)	0.86	0.47
Trained Risk Detector (D_t)	0.83	0.38
ProActive Finetuned Risk Detector (D_f)	**0.91**	**0.59**

detector's gradient norm should diminish over time as a primary indicator of convergence to a stationary point. Empirically, we tracked this metric alongside downstream performance to observe the training dynamics. In the early outer loops, we observed a distinct "Arms Race," characterized by a temporary dip in detector metrics where AUPRC dropped by almost 11%. This confirms that the generator successfully learned to evade the initial detector, effectively creating "hard" adversarial examples. However, by last iteration, the system demonstrated adaptation and recovery; the detector adapted to these sophisticated attacks, improving the AUPRC to 0.59. Crucially, this recovery was accompanied by a consistent decrease in the detector gradient norm, validating that the system avoids mode collapse and reliably converges to a stable equilibrium.

We also measured the incremental impact in detecting fraud that previously bypassed existing detection systems. Over one week of registration data, D_s uncovered 18.46% additional fraud cases missed by prior systems, while D_f further identified an incremental 3.05% on top of D_s. Based on offline analysis, applying PAFT finetuning to the currently deployed model has the potential to increase annualized savings by an estimated 17.6% over the existing baseline.

7 Conclusion

In this work, we introduced ProFraudGuard, a proactive fraud detection system leveraging adversarial LLM fine-tuning. We demonstrated its robustness through theoretical analysis and empirical validation, observing consistent decreases in detector gradient norms. Experimental results confirm that PAFT significantly outperforms static baselines, achieving a Precision@Top10% of 0.91 and an AUPRC of 0.59. Our analysis confirms that reward function - by enforcing novelty and formatting constraints - prevents mode collapse and ensures syntactic validity, yielding diverse, high-quality adversarial samples. Beyond improved metrics, PAFT identified an incremental 3.05% of frauds missed by prior systems, projecting a 17.6% increase in annualized savings, demonstrating critical efficacy in securing dynamic, cold-start registration environments against evolving threats.

References

[1] Richard J. Bolton and David J. Hand. 2002. Statistical Fraud Detection: A Review. *Statist. Sci.* 17, 3 (August 2002), –. doi:10.1214/ss/1042727940

[2] Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335* (2024).

[3] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep Reinforcement Learning from Human Preferences. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf

[4] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research* 25, 70 (2024), 1–53.

[5] Yuebing Liang, Shenhao Wang, Jiangbo Yu, Zhan Zhao, Jinhua Zhao, and Sandy Pentland. 2025. Analyzing sequential activity and travel decisions with interpretable deep inverse reinforcement learning. *arxiv* (2025). arXiv:2503.12761 [cs.AI] https://arxiv.org/abs/2503.12761

[6] Fei Liu et al. 2020. Learning to summarize from human feedback. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 583–592.

[7] Yinqiu Liu, Ruichen Zhang, Hongyang Du, Dusit Niyato, Jiawen Kang, Zehui Xiong, and Dong In Kim. 2024. Defining Problem from Solutions: Inverse Reinforcement Learning (IRL) and Its Applications for Next-Generation Networking. *arXiv preprint arXiv:2404.02405* (2 Apr 2024).

[8] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *arXiv* (2024). arXiv:2305.18290 [cs.LG] https://arxiv.org/abs/2305.18290

[9] Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2022. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241* (2022).

[10] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[11] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300* (2024).

[12] Saeedreza Shehnepoor, Roberto Togneri, Wei Liu, and Mohammed Bennamoun. 2023. Social Fraud Detection Review: Methods, Challenges and Analysis. *arxiv* (2023). arXiv:2111.05645 [cs.LG] https://arxiv.org/abs/2111.05645

[13] Saeedreza Shehnepoor, Roberto B. Togneri, Wei Liu, and Bennamoun. 2022. Spatio-Temporal Graph Representation Learning for Fraudster Group Detection. *IEEE Transactions on Neural Networks and Learning Systems* 35 (2022), 6628–6642. https://api.semanticscholar.org/CorpusID:245837773

[14] David Silver and Richard S Sutton. 2025. Welcome to the Era of Experience. *Preprint of a chapter to appear in Designing an Intelligence, edited by George Konidaris, MIT Press (forthcoming)* (2025).

[15] Gurjot Singh, Prabhjot Singh, and Maninder Singh. 2025. Advanced Real-Time Fraud Detection Using RAG-Based LLMs. *arxiv* (2025). arXiv:2501.15290 [cs.CR] https://arxiv.org/abs/2501.15290

[16] Davoud Ataee Tarzanagh, Mingchen Li, Christos Thrampoulidis, and Samet Oymak. 2022. Fednest: Federated bilevel, minimax, and compositional optimization. In *International Conference on Machine Learning*. PMLR, 21146–21179.

[17] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944* (2023).

[18] Xiaoyu Wang, Rui Pan, Renjie Pi, and Jipeng Zhang. 2023. Effective bilevel optimization via minimax reformulation. *arXiv preprint arXiv:2305.13153* (2023).

[19] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8 (1992), 229–256.

[20] Yifan Yang, Zhaofeng Si, Siwei Lyu, and Kaiyi Ji. 2024. First-Order Minimax Bilevel Optimization. *Advances in Neural Information Processing Systems* 37 (2024), 24990–25035.

[21] Yunxia Yu. 2024. Hybrid models for accuracy and explainability in AI systems. *Reseach Gate* (07 2024), 309. doi:10.1117/12.3031351

[22] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. 2008. Maximum entropy inverse reinforcement learning.. In *Aaai*, Vol. 8. Chicago, IL, USA, 1433–1438.

[23] Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593* (2019).

A Proof for Algorithm 1

Assumption A

For Algorithm 1 we assume that

1. *The detector's objective function $L^*(\phi)$ is reasonably smooth, meaning its gradient w.r.t ϕ is Lipschitz continuous, i.e:*

$$\|\nabla L^*(\phi_1) - \nabla L^*(\phi_2)\| \leq L\|\phi_1 - \phi_2\|$$

for some constant $L > 0$. (This implicitly assumes smoothness of L_{det}, R, G, and the dependence of ψ^* on ϕ).

2. *The stochastic gradient estimator $\hat{g}_{h,n}$ has bounded variance or bounded second moment:*

$$\mathbb{E}[\|\hat{g}_{h,n}\|^2] \leq G^2$$

for some constant $G > 0$. (This implicitly assumes bounded losses, rewards and gradients).

3. *Bias Bound: The difference between the true gradient $\nabla L^*(\phi_{h,n})$ and the expected stochastic gradient $\mathbb{E}[\hat{g}_{h,n}|\phi_{h,n}] = \nabla_\phi V(\phi_{h,n}, \psi_h)$ is bounded:*

$$\|\nabla L^*(\phi_{h,n}) - \nabla_\phi V(\phi_{h,n}, \psi_h)\| \leq L'\|\phi_{h,n} - \phi_{h,0}\|$$

for some constant $L' > 0$. This captures the error introduced by using the fixed generator ψ_h instead of the optimal $\psi^*(\phi_{h,n})$.

These are the standard assumptions of stochastic gradient methods for non-convex optimization .

Now we are ready to re-state and prove Theorem 5.1:

Theorem A.1. *Under the assumptions A, for Algorithm 1 with $\eta_\phi = \Theta(1/\sqrt{HN})$ we have*

$$\min_{h=0\ldots H-1, n=0\ldots N-1} \mathbb{E}[\|\nabla L^*(\phi_{h,n})\|^2] \leq O\left(\frac{1}{\sqrt{HN}} + \frac{1}{H}\right)$$

Proof:

From Lipschitz gradient of L^*, for the update $\phi_{h,n+1} = \phi_{h,n} + \eta_\phi \hat{g}_{h,n}$ we have:

$$L^*(\phi_{h,n+1}) \geq L^*(\phi_{h,n}) + \langle \nabla L^*(\phi_{h,n}), \eta_\phi \hat{g}_{h,n}\rangle - \frac{L}{2}\|\eta_\phi \hat{g}_{h,n}\|^2 \tag{2}$$

Rearranging above Equation 2:

$$\eta_\phi \langle \nabla L^*(\phi_{h,n}), \hat{g}_{h,n}\rangle \leq L^*(\phi_{h,n+1}) - L^*(\phi_{h,n}) + \frac{L\eta_\phi^2}{2}\|\hat{g}_{h,n}\|^2$$

Expanding the inner product $\langle \nabla L^*, \hat{g}\rangle$ to $\|\nabla L^*\|^2 + \langle \nabla L^*, \hat{g} - \nabla L^*\rangle$:

$$\eta_\phi \|\nabla L^*(\phi_{h,n})\|^2 + \eta_\phi \langle \nabla L^*(\phi_{h,n}), \hat{g}_{h,n} - \nabla L^*(\phi_{h,n})\rangle \leq L^*(\phi_{h,n+1}) - L^*(\phi_{h,n}) + \frac{L\eta_\phi^2}{2}\|\hat{g}_{h,n}\|^2$$

Taking Expectation to $\phi_{h,n}$ (denoted $\mathbb{E}_n$):

$$\eta_\phi \mathbb{E}_n[\|\nabla L^*(\phi_{h,n})\|^2] + \eta_\phi \langle \nabla L^*(\phi_{h,n}), \mathbb{E}_n[\hat{g}_{h,n}] - \nabla L^*(\phi_{h,n})\rangle \leq \mathbb{E}_n[L^*(\phi_{h,n+1})] - L^*(\phi_{h,n}) + \frac{L\eta_\phi^2}{2}\mathbb{E}_n[\|\hat{g}_{h,n}\|^2]$$

Using Assumption A ($\mathbb{E}[\|\hat{g}\|^2] \leq G^2$) we have:

$$\eta_\phi \mathbb{E}[\|\nabla L^*\|^2] + \eta_\phi \langle \nabla L^*, \mathbb{E}[\hat{g}] - \nabla L^*\rangle \leq \mathbb{E}[L_{n+1}] - L_n + \frac{L\eta_\phi^2 G^2}{2} \tag{3}$$

where the expectation is taken w.r.t. the sample $z_{h,n}$ to generate the estimator of current iteration.

Using Cauchy-Schwarz an Young's inequality we know:

$$\langle \nabla L^*, \text{Bias}_n\rangle \geq -\|\nabla L^*\|\|\text{Bias}_n\| \geq -\frac{1}{2}\|\nabla L^*\|^2 - \frac{1}{2}\|\text{Bias}_n\|^2$$

where we represent $Bias_n$ as $\mathbb{E}[\hat{g}_{h,n}] - \nabla L^*(\phi_{h,n})$.

Substituting this in Equation 3 we have:

$$\eta_\phi \mathbb{E}[\|\nabla L^*\|^2] - \frac{\eta_\phi}{2}\mathbb{E}[\|\nabla L^*\|^2] - \frac{\eta_\phi}{2}\mathbb{E}[\|\text{Bias}_n\|^2] \leq \mathbb{E}[L_{n+1}] - L_n + \frac{L\eta_\phi^2 G^2}{2}$$

$$\frac{\eta_\phi}{2}\mathbb{E}[\|\nabla L^*(\phi_{h,n})\|^2] \leq \mathbb{E}[L^*(\phi_{h,n+1})] - L^*(\phi_{h,n}) + \frac{\eta_\phi}{2}\mathbb{E}[\|\text{Bias}_n\|^2] + \frac{L\eta_\phi^2 G^2}{2} \tag{4}$$

Using Assumption A 2 and 3, we have:

$$\|\text{Bias}_n\| = \|\nabla_\phi V(\phi_{h,n}, \psi_h) - \nabla L^*(\phi_{h,n})\| \leq L'\|\phi_{h,n} - \phi_{h,0}\|$$

$$\mathbb{E}[\|\text{Bias}_n\|^2] \leq (L')^2 \mathbb{E}[\|\phi_{h,n} - \phi_{h,0}\|^2]$$

$$\mathbb{E}[\|\text{Bias}_n\|^2] \leq (L')^2 n\eta_\phi^2 G^2 \tag{5}$$

Substitute Equation 5 in 4, we have:

$$\frac{\eta_\phi}{2}\mathbb{E}[\|\nabla L^*(\phi_{h,n})\|^2] \leq \mathbb{E}[L_{n+1}] - L_n + \frac{\eta_\phi}{2}(L')^2 n\eta_\phi^2 G^2 + \frac{L\eta_\phi^2 G^2}{2}$$

$$\frac{\eta_\phi}{2}\mathbb{E}[\|\nabla L^*\|^2] \leq \mathbb{E}[L_{n+1}] - L_n + \frac{\eta_\phi^3 (L')^2 G^2 n}{2} + \frac{\eta_\phi^2 LG^2}{2}$$

Sum over inner loop from $n = 0$ to $N - 1$ we get:

$$\frac{\eta_\phi}{2}\sum_{n=0}^{N-1}\mathbb{E}[\|\nabla L^*(\phi_{h,n})\|^2] \leq \mathbb{E}[L^*(\phi_{h,N})] - L^*(\phi_{h,0}) + \frac{\eta_\phi^3 (L')^2 G^2}{2}\sum_{n=0}^{N-1} n + \frac{N\eta_\phi^2 LG^2}{2}$$

Using $\sum_{n=0}^{N-1} n = \frac{N(N-1)}{2} \approx \frac{N^2}{2}$:

$$\frac{\eta_\phi}{2}\sum_{n=0}^{N-1}\mathbb{E}[\|\nabla L^*\|^2] \leq \mathbb{E}[L_{h,N}] - L_{h,0} + \frac{\eta_\phi^3 (L')^2 G^2 N^2}{4} + \frac{\eta_\phi^2 NLG^2}{2}$$

Sum over outer loop from $h = 0$ to $H - 1$ we get:

$$\frac{\eta_\phi}{2}\sum_{h=0}^{H-1}\sum_{n=0}^{N-1}\mathbb{E}[\|\nabla L^*\|^2] \leq L^*(\phi_{H,N}) - L^*(\phi_0) + H\frac{\eta_\phi^3 (L')^2 G^2 N^2}{4} + H\frac{\eta_\phi^2 NLG^2}{2}$$

Let $\Delta L^* = L^*(\phi_{final}) - L^*(\phi_{initial})$.

Obtain Average Gradient Bound by dividing by $HN(\eta_\phi/2)$:

$$\frac{1}{HN}\sum_{h=0}^{H-1}\sum_{n=0}^{N-1}\mathbb{E}[\|\nabla L^*(\phi_{h,n})\|^2] \le \frac{2\Delta L^*}{\eta_\phi HN} + \frac{\eta_\phi^2 (L')^2 G^2 N}{2} + \eta_\phi L G^2$$

Taking $\eta_\phi = \Theta(1/\sqrt{HN})$, we get:

$$\frac{1}{HN}\sum\sum\mathbb{E}[\|\nabla L^*\|^2] \le O\left(\frac{\Delta L^*}{\sqrt{HN}} + \frac{N}{HN} + \frac{LG^2}{\sqrt{HN}}\right)$$

$$\min_{h,n}\mathbb{E}[\|\nabla L^*(\phi_{h,n})\|^2] \le \frac{1}{HN}\sum\sum\mathbb{E}[\|\nabla L^*\|^2] \le O\left(\frac{\Delta L^*}{\sqrt{HN}} + \frac{1}{H} + \frac{LG^2}{\sqrt{HN}}\right)$$