PreScam: A Benchmark for Predicting Scam Progression from Early Conversations

Weixiang Sun[*], **Shang Ma**[*], **Yiyang Li, Tianyi Ma, Zehong Wang, Colby Nelson**
University of Notre Dame

Xusheng Xiao
Arizona State University

Yanfang Ye[†]
University of Notre Dame

Abstract

Conversational scams, such as romance and investment scams, are emerging as a major form of online fraud. Unlike one-shot scam lures such as fake lottery or unpaid toll messages, conversational scams unfold through multi-turn conversations in which scammers gradually manipulate victims using evolving psychological techniques. However, existing research mainly focuses on static scam detection or synthetic scams, leaving open whether language models can understand how real-world scams progress over time. We introduce PRESCAM, a benchmark for modeling scam progression from early conversations. Built from real victim-submitted scam reports, PRESCAM converts 177,989 raw reports into 11,573 conversational scam instances spanning 20 scam categories. Each instance is hierarchically structured according to the scam lifecycle defined by the proposed scam kill chain, and further annotated at the turn level with scammer psychological actions and victim responses. Furthermore, we present two benchmark tasks motivated by real-world anti-scam needs: real-time termination prediction, which estimates whether a conversation is approaching the termination stage, and scammer action prediction, which forecasts the scammer's subsequent actions. Results show a clear gap between surface-level fluency and progression modeling: supervised encoders substantially outperform zero-shot LLMs on real-time termination prediction, while next-action prediction remains only moderately successful even for strong LLMs. Taken together, these results show that current models can capture some scam-related cues, yet still struggle to track how risk escalates and how manipulation unfolds across turns. Our code and dataset are avaiable[1].

1 Introduction

The schemes and strategies underlying scams have evolved alongside human society and communication technology. Throughout history, these schemes and strategies have adapted to the dominant communication media of their time, from word-of-mouth deception in ancient societies (Dill, June 2022; Jiahui, March 2023), to print-based fraud such as the Spanish Prisoner (Wikipedia, March 2026), to telephone-based schemes such as boiler-room and Ponzi-style operations (U.S. Securities and Exchange Commission, March 2026). Today, the Internet and social media have enabled scams that are more scalable, personalized, and psychologically sophisticated than ever before. Many modern scams, including investment, employment, and romance scams, do not unfold as isolated messages; instead, they develop through multi-turn interactions in which scammers gradually manipulate victims over time. In this work, we refer to such engagement-heavy scams as **conversational scams**. According to the Better Business Bureau (BBB) Annual Scam Risk Reports (2016–2024),

[*]Equal contribution. Email: {wsun4,sma5}@nd.edu
[†]Corresponding author. Email: yye7@nd.edu
[1]https://github.com/KiteFlyKid/PreScam

major conversational scams (e.g., investment and employment scams) have emerged to dominate the riskiest scams over the last decade (Better Business Bureau, March 2026).

A defining characteristic of conversational scams is that they are *processes* rather than static artifacts. Scammers typically begin by establishing contact, then sustain engagement through repeated psychological manipulation, and finally attempt to extract money, credentials, or other assets. This progression is often strategic rather than random. Prior work in psychology and scam studies suggests that scammers rely on recurring psychological techniques such as authority, urgency, trust building, and fear induction to shape victims' decisions over time (Lea et al., 2009; Ma et al., 2025a; Huang et al., 2024; Wang et al., 2026). This observation raises an important modeling question: beyond recognizing individual psychological techniques, can language models understand how these techniques are used over time as a conversational scam progresses?

Existing efforts mostly study scam progression through either synthetic simulation or static detection (Eder, 2025; Yang et al., 2025b; Ma et al., 2025b; Kumarage et al., 2025; Ye et al., 2025). While promising, these approaches face two major limitations. First, synthetic conversations are often shaped by the inductive biases of the generating model and may not faithfully reflect the diversity and messiness of real-world scam reports. Second, most existing formulations treat scam conversations as unstructured text, overlooking the latent sequential structure that governs how scammers escalate manipulation over time and, consequently, failing to capture the dynamics of scam progression.

To address this gap, we collect a large corpus of real-world scam reports from a prominent scam-reporting platform and transform them into 11,573 structured multi-turn scam conversations. Specifically, we introduce a new representation, termed the *Scam Kill Chain*, inspired by prior work in cybersecurity and cognitive science (MITRE, April 2025; Montañez Rodriguez & Xu, 2022; Lea et al., 2009; Ma et al., 2025a) that formalizes cyber attacks and social engineering as structured, staged processes. The Scam Kill Chain formalizes each scam conversation through three lifecycle stages: *Initial Contact*, *Engagement*, and *Termination*, and represents each stage using scammer actions grounded in psychological techniques, which we refer to as **PT Actions**. By explicitly modeling both the temporal phase of the scam and the underlying psychological manipulation, this representation converts raw scam narratives into structured manipulation trajectories and enables scam progression to be studied as a sequential reasoning problem.

Based on this representation, we present PRESCAM, the first benchmark for modeling scam progression from early conversations. PRESCAM is designed to evaluate whether models can track the evolving risk of an interaction and anticipate the scammer's next move from partial context. Concretely, we consider two tasks: **real-time termination prediction**, which measures whether a model can identify when an ongoing interaction is approaching the termination stage, and **scammer action prediction**, which tests whether a model can predict the scammer's PT actions conditioned on the observed history. Together, these tasks move beyond conventional scam classification and directly probe whether models can recover the underlying progression structure shaped by psychological manipulation. Our study yields three main contributions.

- **Benchmark construction**: We construct a large-scale benchmark of real-world conversational scams with structured stage annotations and turn-level psychological technique labels.
- **Empirical characterization**: We provide a quantitative characterization of scam progression, revealing recurring stage-specific and scam-type-specific manipulation patterns.
- **Model evaluation**: We systematically evaluate a range of language models and baselines on scam progression modeling. Our results show that while current models capture useful scam-related priors, they remain limited in modeling the sequential, psychologically grounded actions that drive scam escalation.

We hope PRESCAM can serve as a useful community benchmark for evaluating whether models can track risk and forecast scammer actions in real-world scam conversations.

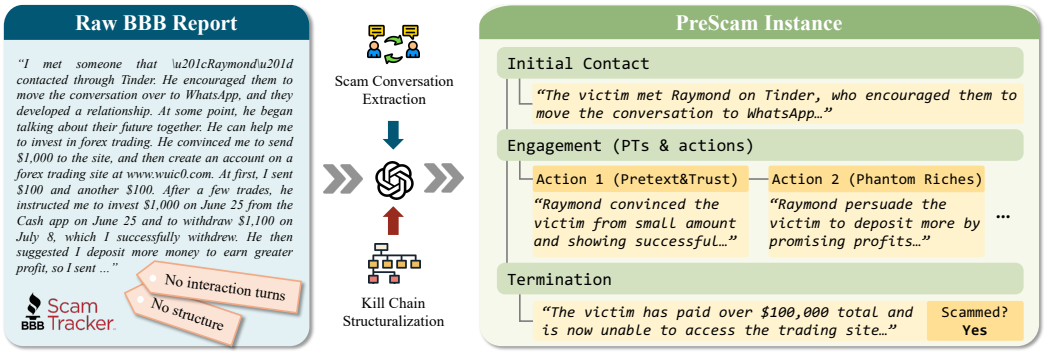


Figure 1: From Raw BBB Report to PRESCAM Instance: an example of scam conversation extraction and kill chain structuralization. The left panel shows an unstructured user-submitted BBB report without explicit interaction turns, while the right panel shows the resulting PRESCAM instance with Initial Contact, Engagement, and Termination.

2 Related Work

2.1 Scam Detection and Behavioral Analysis

Current computer science research on online scams generally falls into two main categories: automated detection and behavioral analysis.

Detection. Automated detection of online scams has been studied across multiple modalities. Visual and brand-impersonation signals have driven a line of phishing website detectors, evolving from CNN-based logo matching (Lin et al., 2021) to knowledge-graph- (Zhao et al., 2023; Ju et al., 2023; Qian et al., 2022; Ju et al., 2022; Zhao et al., 2021) and LLM-augmented reference-based systems (Liu et al., 2023; Li et al., 2024; Cao et al., 2025). Scams propagated through SMS and telephony have also received attention, with works characterizing smishing infrastructure (Nahapetyan et al., 2024) and classifying robocall content at scale (Prasad et al., 2023). In the cryptocurrency domain, detection efforts span Ponzi schemes (Chen et al., 2018), pump-and-dump operations (Xu & Livshits, 2019), and transaction-based phishing (He et al., 2023). More recently, the threat of LLM-generated phishing has prompted both attack studies and LLM-based defenses (Roy et al., 2024; Koide et al., 2024).

Behavioral. Behavioral research has examined scams from both victim and scammer perspectives. On the victim side, studies investigate susceptibility factors (Hanoch & Wood, 2021; Ye et al., 2009) and how persuasion principles such as authority and scarcity are exploited (Van Der Heijden & Allodi, 2019). On the scammer side, Herley (2012) offers a game-theoretic account of self-selection in advance-fee fraud, while (Miramirkhani et al., 2017) document social engineering scripts through direct scammer conversation. Structured manipulation lifecycles have been characterized for romance scams and the emerging pig-butchering variant (Acharya & Holz, 2024).

However, these lines of work predominantly target static artifacts such as webpages and messages, or rely on post-hoc victim reports. They provide limited support for modeling scams that unfold dynamically through multi-turn interactions.

2.2 Scam Conversation Datasets

Recognizing the scarcity of research on conversational scams, recent studies have sought to bridge this gap by contributing scam conversation datasets. Some approaches rely on LLMs to synthesize these interactions (Eder, 2025; Yang et al., 2025b; Ma et al., 2025b; Kumarage et al., 2025). However, synthesized data often diverges from real-world scenarios and struggles to capture the rapid evolution and structural complexity of actual scams (Ma

et al., 2025a; Chen et al., 2025). Alternative efforts involve active engagement, deploying personas or honeypots to interact directly with scammers and collect real-world conversation data (Perkins & Howell, 2021; Spokoyny et al., 2025; Acharya & Holz, 2024). In contrast, our work contributes both a new dataset of real-world scam conversations and a benchmark for evaluating whether models can track and forecast scam progression from partial context.

3 Preliminary

3.1 From Kill Chains to Scam Progression

Existing cyber attack models, such as the Cyber Kill Chain (Martin, 2026), formalize attacks into distinct sequential phases and associate each phase with specific attacker tactics and techniques (MITRE, April 2025; Barnum, 2012). Similarly, social engineering kill chains (Montañez Rodriguez & Xu, 2022; Longtchi et al., 2024) model human-centric attacks as staged psychological manipulation, while recent work (Ma et al., 2025a) operationalizes this perspective through explicit Psychological Techniques (PTs). These frameworks suggest that scam conversations should not be viewed as isolated messages, but as structured processes that evolve over time through changing tactics.

This perspective is especially important for conversational scams, where different stages serve distinct objectives and therefore involve separate behavioral tactics and psychological techniques. Treating an entire conversation as unstructured free text loses the temporal progression of the attack, the transitions between stages, and the changing role of psychological manipulation over time.

3.2 Scam Kill Chain Representation

To capture this structure, we introduce the *Scam Kill Chain*, a representation that maps the temporal phases of a scam conversation to actions driven by psychological techniques, which we refer to as *PT actions*. The chain consists of three primary phases:

- *Initial Contact.* The scammer triggers the victim to initiate or accept communication, establishing initial access.
- *Engagement.* The core of the scam conversation. The scammer exploits multiple psychological techniques to establish continuous communication, build deep trust, and increase victim compliance.
- *Termination.* The scammer requests the desired action (e.g., monetary extraction). Alternatively, the victim realizes the deception, leading to the end of the scam lifecycle.

Each phase is realized through one or more PT actions, where the scammer operationalizes a specific psychological technique to achieve a phase-specific objective. This representation makes the temporal progression of a scam explicit while preserving the underlying psychological techniques exploited at each stage. Figure 1 provides a concrete example of this representation, and Appendix A lists the PT taxonomy that we extend from recent work (Ma et al., 2025a). Appendix G provides additional examples across other scam types, such as pig-butchering and employment scams.

Formal Definition. A **scam conversation** $\mathcal{C} = (u_1, u_2, \ldots, u_T)$ is a multi-turn dialogue between a scammer and a victim. Let $\mathcal{P} = \{p_1, p_2, \ldots, p_K\}$ denote the set of K psychological techniques drawn from the predefined taxonomy. A **PT action** is a tuple $a_t = (p_t, u_t)$, where $p_t \in \mathcal{P}$ is the exploited PT and u_t is the scammer utterance that operationalizes it. A **scam kill chain** is a temporally ordered triple of phases $\mathcal{K} = \langle \phi_{\text{IC}} \to \phi_{\text{EG}} \to \phi_{\text{TM}} \rangle$, representing *Initial Contact*, *Engagement*, and *Termination*, respectively. Each phase $\phi = [a_i, \ldots, a_j]$ is a contiguous sequence of PT actions.

The scam kill chain structurally formalizes the conversation by inducing a temporal partition over $\mathcal{C}$ such that the full sequence is the concatenation of its disjoint phases:

$$\mathcal{C} = \phi_{\text{IC}} \oplus \phi_{\text{EG}} \oplus \phi_{\text{TM}}$$

Concurrently, this framework maps each utterance u_t to its corresponding underlying psychological technique p_t to yield the structured action sequence.

4 PRESCAM Dataset

4.1 Overview

We construct PRESCAM entirely from real-world scam reports collected from BBB Scam Tracker between February 2024 and November 2025, distilling 177,989 raw reports into 11,573 structured scam instances through the three-step construction pipeline detailed below.

Each instance in PRESCAM includes a scam category label, the original victim narrative, and a structured three-stage representation consisting of *Initial Contact*, *Engagement*, and *Termination*. The engagement stage is further decomposed into turn-level scammer and victim actions, where a *turn* denotes a single utterance by either party (u_i in the task formulation below), with supporting verbatim spans and PT annotations, and each instance also includes the binary *scammed* label and *scammed_reason*. This design preserves the evidential content of the original reports while transforming them into a structured representation suitable for stage-level and tactic-level analysis.

The final dataset covers 20 scam categories and exhibits a pronounced long-tailed distribution. The structured scam conversations are typically short and multi-turn, whereas the original victim narratives are substantially longer and noisier. More detailed dataset schema descriptions, static analyses, and category-level insights, including category proportions, interaction-length statistics, and PT usage patterns, are provided in Appendix B.

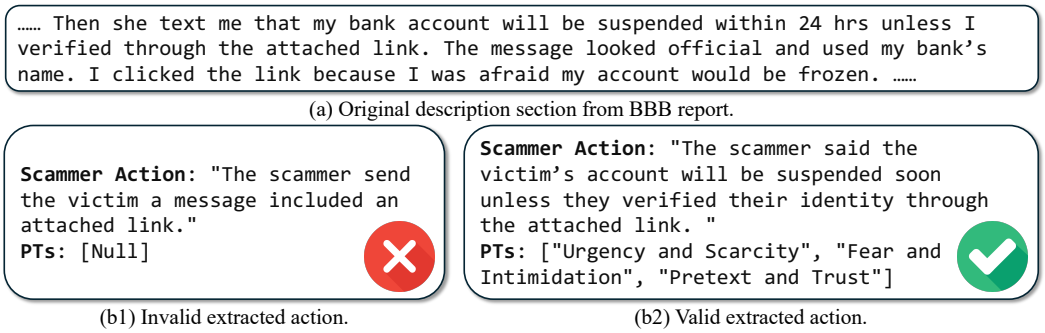
4.2 Construction Pipeline

Step 1: Seed Dataset Collection. We begin with publicly available real-world scam reports from BBB Scam Tracker (Better Business Bureau, February 2026). We collect all reports published between February 2024 and November 2025, resulting in an initial corpus of 177,989 raw reports.

Step 2: Scam Conversation Extraction. Raw user reports present substantial challenges for automated analysis due to their noisy and highly unstructured nature. Victims often submit emotional free-form narratives, incomplete descriptions, or single-turn accounts rather than genuine scam conversations (see Appendix C for examples). Our preliminary analysis shows that a large portion of raw reports do not contain usable multi-turn conversations, making manual annotation prohibitively slow and expensive. Moreover, rule-based filtering is ineffective, as scam conversations do not exhibit reliable length-based or string-pattern regularities.

To address this issue, we employ GPT-4O-MINI as an LLM-based extractor to filter out noisy reports and identify cases containing scammer and victim multi-turn conversations. This step yields 25,402 candidate conversations. We then remove cases with fewer than two scammer–victim exchanges in the engagement stage, retaining 13,007 samples for downstream structuralization. The extraction prompt is provided in Appendix E.

Step 3: Kill Chain Structuralization. After extracting the core conversations, we map them onto our scam kill chain framework. Specifically, we use MINIMAX-2.5 to organize each case into three stages (i.e., Initial Contact, Engagement, and Termination) and identify the PT actions employed throughout the conversation. For the engagement stage, the model decomposes the conversation into fine-grained scammer and victim actions, aligns them with verbatim evidence spans, and assigns PT labels. Importantly, we define an extracted scammer action as *valid* only when it is grounded in the source report and paired with at least one PT label; action spans with null PT annotations are treated as invalid and removed. Figure 2 illustrates this distinction with a concrete example. We further conduct post-processing to remove null or malformed entries introduced during generation, resulting in a final dataset of 11,573 structured instances. The structuralization prompt



(a) Original description section from BBB report.

(b1) Invalid extracted action.

(b2) Valid extracted action.

Figure 2: Comparison between invalid and valid extracted actions. An extracted action is considered valid only if it is grounded in the source report and paired with at least one PT label.

and formatting guidelines are provided in Appendix F, and Appendix D.6 justifies the choice of GPT-4O-MINI and MINIMAX-2.5 with a cost–quality pilot study against GPT-5, GEMINI-2.5-FLASH, and CLAUDE-HAIKU-4.5.

4.3 Quality Control

To improve the reliability of the LLM-generated structures and mitigate potential hallucinations, we introduce a secondary LLM as a self-reflection agent to verify the structuralization outputs against a strict verification checklist including multiple perspectives.

We further conduct an independent human quality-control study on $n = 200$ randomly sampled generated data. Three PhD-level reviewers evaluate each sample using an 8-point rubric. Averaged across reviewers, the mean score increases from 7.10 to 7.37 (out of 8), yielding a $+0.27$ improvement from the self-reflection stage. Reviewer-wise results, including win/tie statistics, per-dimension breakdowns, and the full evaluation criteria, are reported in Appendix D.

5 Benchmark Task Formulation

We design two core tasks on the PRESCAM dataset: *Real-time Termination Prediction* and *Scammer Action Prediction*. Both tasks operate over kill-chain-structured conversations. The two tasks probe complementary aspects of scam progression modeling. Real-time termination prediction evaluates whether a model can track escalating scam risk from partial context, while scammer action prediction evaluates whether it can recover the future action–technique structure that drives that progression.

5.1 Real-time Termination Prediction

Motivation. In dynamic, interactive environments, such as live banking platforms or messaging applications (ScamShield, 2025), systems cannot always wait for the *Termination* phase (e.g., the scammer explicitly asks for payment) before flagging a conversation as dangerous. Therefore, the goal is to predict whether the conversation is approaching the critical termination stage early enough to support timely intervention. Note that PRESCAM studies progression modeling *within* conversations already identified as suspicious or scam-related; screening scam from benign conversations is the role of an upstream detector in deployed pipelines and is outside the scope of our tasks.

Task Definition. We frame real-time termination prediction as a *continuous risk estimation* task. Given the conversational history up to turn t, denoted as $\mathcal{C}_{\leq t} = (u_1, \ldots, u_t)$, the objective is to output a continuous risk score $\hat{p}_t \in [0, 1]$ indicating the proximity to the *Termination* phase ϕ_{TM}. A score of $\hat{p}_t \to 1$ indicates high confidence that the scammer's next

turn will enter the termination phase (e.g., asking for payment). Formally, a prefix $\mathcal{C}_{\leq t}$ is labeled positive if the conversation enters ϕ_{TM} within the next k turns; all main experiments use $k = 1$, the strictest setting, and Appendix H.2 shows our conclusions are robust across $k \in \{1, 2, 3\}$. We evaluate this task under two inference settings: *Direct LLM Prompting*, where an off-the-shelf LLM predicts whether the next turn enters the *Termination* phase and assigns a risk score $\hat{p}_t = \text{LLM}(\mathcal{C}_{\leq t})$; and *Supervised Sequence Classification*, where a neural network f_θ maps the conversation history directly to a continuous risk score $\hat{p}_t = f_\theta(\mathcal{C}_{\leq t})$.

Metrics. To evaluate the model's ability to maintain an accurate and timely rolling risk assessment, we utilize the following metrics:

- **Area Under the Risk Curve (AUC):** We compute the aggregate AUC of the predicted trajectory $(\hat{p}_1, \ldots, \hat{p}_{T_{\text{TM}}})$ to evaluate the overall monotonicity and confidence accumulation as the conversation progresses.
- **Area Under the Precision-Recall Curve (AUPR):** Because positive prefixes (the final k turns of each conversation) are sparse, ROC-AUC can be overly optimistic under this class imbalance. We therefore also report AUPR (equivalent to average precision), which is more sensitive to the minority positive class and rewards identifying high-risk turns with fewer false alarms.
- **Alert Time (AT@FPR$_\alpha$):** To measure practical early-warning utility, we select a method-specific threshold $\tau_\alpha = \inf\{\tau : \text{FPR}(\tau) \leq \alpha\}$ on the test set (we use $\alpha = 10\%$) and define $\text{AT@FPR}_\alpha = T_{\text{TM}} - \min\{t \mid \hat{p}_t \geq \tau_\alpha\}$, i.e., how many turns before the termination phase the risk score first breaches the threshold. A larger value indicates an earlier, more actionable alert, and calibrating every method to the same false positive rate makes the metric directly comparable across methods.

5.2 Scammer Action Prediction

Motivation. Anticipating a scammer's next move during the *Engagement* phase enables an early warning before the interaction escalates to termination, which is critical for prevention.

Task Definition. We frame this task as a sequential forecasting problem. Given the conversational context up to turn t, denoted as $\mathcal{C}_{\leq t} = (u_1, \ldots, u_t)$, the objective is to predict the scammer's subsequent actions in the remaining conversation. We do not use a globally fixed boundary turn: scam intent emerges at different speeds across cases (Figure 1), so a fixed early boundary would make the evaluation uneven. We therefore instantiate t dynamically by first using GPT-4O-MINI to label whether each observed prefix already appears likely to be a scam, and then training a separate ROBERTA classifier on these labels to select t with a conservative threshold of 0.9, thereby improving generalization beyond the annotating LLM.

We evaluate this task in a zero-shot setting under two inference conditions: *Unlimited*, where the LLM is given the observed conversation context and generates future scammer actions freely, i.e., $\hat{A}_{>t} = \text{LLM}(\mathcal{C}_{\leq t})$; and *Limited*, where the LLM is given the observed conversation context together with the exact number of remaining scammer actions and generates that many future actions, i.e., $\hat{A}_{>t} = \text{LLM}(\mathcal{C}_{\leq t},\ n_{\text{remain}})$, where n_{remain} denotes the number of future scammer actions in the gold continuation after turn t. Here, $\hat{A}_{>t}$ denotes the predicted sequence of future scammer actions. Additional details on this dynamic boundary construction are provided in Appendix H.3.

Metrics. To evaluate predicted actions, we report two task-specific hit-rate metrics together with two standard text similarity metrics. Let the gold next PT action for evaluation instance i be $a_i = (p_i, u_i)$, where u_i is the reference scammer utterance, and let the predicted next PT action be $\hat{a}_i = (\hat{p}_i, \hat{u}_i)$. Because a generated next move may partially recover the intended action content or psychological techniques without exactly matching the reference wording, we first use an **LLM-as-a-Judge** framework with GPT-4O-MINI to extract structured labels from the generated and gold next-action texts, and then compute hit-rate metrics over these extracted sets. Human review on 100 sampled cases (199 actions) shows strong agreement with the judge, reaching 92.0% action-level agreement and Cohen's $\kappa = 0.774$ (Appendix D).

Specifically, let $\Psi(\cdot)$ denote the set of atomic action elements extracted by the LLM judge from a next-action text, and let $\Gamma(\cdot)$ denote the set of PT labels extracted by the same judge.

- **Action HitRate:** We measure how much of the ground-truth action content is recovered by the prediction using the final score:

$$\text{Action HitRate} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\Psi(u_i)\cap\Psi(\hat{u}_i)|}{|\Psi(u_i)|}$$

- **PT HitRate:** Similarly, we measure how many ground-truth psychological techniques are recovered by the prediction using the final score:

$$\text{PT HitRate} = \frac{1}{N}\sum_{i=1}^{N}\frac{|\Gamma(u_i)\cap\Gamma(\hat{u}_i)|}{|\Gamma(u_i)|}$$

- **Auxiliary Text Metrics:** In addition to the two judge-based hit-rate metrics above, we report BERTScore and ROUGE-L between the predicted utterance $\hat{u}_i$ and the reference utterance u_i to measure semantic similarity and lexical overlap, respectively. We also report Precision to measure how much of the predicted action content is supported by the gold action set, i.e., the fraction of extracted predicted action elements that overlap with $\Psi(u_i)$.

6 Evaluation

6.1 Experimental Setup

Real-time Termination Prediction. We employ multiple baseline models categorized into three types for comparison: classical machine learning approaches (a position-only classifier using turn index with logistic regression, and TF-IDF encoding (Sparck Jones, 1972) + Logistic Regression) trained using an 80%–20% train–test split; neural models (MLP with TF-IDF features, MLP with embeddings, LSTM (Hochreiter & Schmidhuber, 1997), hierarchical encoder, Transformer (Vaswani et al., 2017), and BERT-base-uncased (Devlin et al., 2019)) fine-tuned under the same 80%–20% split; and LLMs (GPT-4O-MINI (OpenAI, 2024), DEEPSEEK-V3 (Liu et al., 2025), QWEN3-235B-A22 (Yang et al., 2025a), and GROK-4.1-FAST (x.ai, 2025)) evaluated in a zero-shot setting, where models generate a risk score between 0 and 1 based on the prompt. To address class imbalance in supervised models, we apply techniques such as weighted loss functions and data resampling. More details are shown in Appendix H.1.

Scammer Action Prediction. Given the conversation up to turn t, models are tasked with generating the scammer's subsequent actions for the remaining conversation. We evaluate multiple LLMs in a zero-shot setting: GPT-4O-MINI, GPT-5, CLAUDE-SONNET-4.5, DEEPSEEK-V3.2, and LLAMA-3.3-70B-INSTRUCT. Each model receives the observed conversation prefix up to t and generates a structured sequence of predicted scammer actions. Action HitRate and PT HitRate are computed using an LLM-as-a-Judge with GPT-4O-MINI. Detailed human-validation results for this judge and other evaluation details can be found in Appendix D and Appendix H.3.

6.2 Main Results

Real-time Termination Prediction. Table 1 shows that real-time termination prediction is driven much more by task-specific progression modeling than by general-purpose LLM reasoning. Overall, supervised encoders substantially outperform zero-shot LLMs on the main ranking metrics, indicating that robust turn-level risk estimation requires learning sequential risk signals from data rather than relying on broad scam-related world knowledge alone. Specifically, first, most supervised text models outperform the position-only baseline on both AUC and AUPR, showing that conversational content provides strong information beyond simple turn position. Second, BERT achieves the best overall ranking

Setting	Method	AHit(%)	PTHit(%)	BERTScore	ROUGE-L	Precision
Unlimited	GPT-5-CHAT	71.84	58.23	**0.8603**	**0.1508**	**0.7323**
	GPT-4O-MINI	65.15	57.55	0.8593	0.1446	0.7166
	CLAUDE-SONNET-4.5	**79.36**	53.76	0.8526	0.1343	0.7052
	DEEPSEEK-V3.2	75.80	**60.43**	0.8439	0.1134	0.5377
	LLAMA-3.3-70B	74.80	55.59	0.8527	0.1379	0.6732
Limited	GPT-5-CHAT	57.53	**50.91**	0.8680	0.1865	0.6673
	GPT-4O-MINI	40.45	45.63	0.8740	**0.2075**	0.5963
	CLAUDE-SONNET-4.5	**63.93**	43.22	0.8661	0.1896	**0.7345**
	DEEPSEEK-V3.2	63.57	49.67	0.8567	0.1588	0.7121
	LLAMA-3.3-70B	52.52	46.43	**0.8686**	0.1976	0.6918

Table 2: Scammer action prediction performance under the Unlimited and Limited settings. We report Action HitRate (AHit), PT HitRate (PTHit), BERTScore, ROUGE-L, and Precision for each model.

performance, reaching 83.4 AUC and 65.6 AUPR, which suggests that pretrained bidirectional encoders are especially effective at aggregating local semantic cues from partial scam histories. Moreover, zero-shot frontier LLMs perform markedly worse on these two metrics, with all of them falling below the simple position-only baseline in both AUC and AUPR. An interesting finding, however, is that the ranking changes for alert timeliness: although BERT is strongest in overall discrimination, GROK-4.1-FAST achieves the highest AT@FPR$_{10\%}$, implying that some LLMs are more willing to issue earlier warnings even when their overall risk calibration remains weak.

Scam Action Prediction. Table 2 and Figure 3 show that scammer action prediction remains challenging even for frontier LLMs. Overall, the results indicate that the main bottleneck is not surface-form similarity, but correctly recovering the latent action–technique structure that drives scam progression. Specifically, first, under the *Unlimited* setting, CLAUDE-SONNET-4.5 achieves the highest overall Action HitRate (79.36), while DEEPSEEK-V3.2 attains the best overall PT HitRate (60.43), again revealing a gap between recovering the concrete subsequent action and recovering its underlying psychological technique. Second, lexical overlap metrics such as ROUGE-L and BERTScore do not align with these structurally grounded measures: models in the *Limited* setting often obtain higher BERTScore and ROUGE-L despite much lower AHit and PTHit. This pattern suggests that constraining the number of future actions mainly regularizes surface realization, making outputs look closer to the reference without necessarily recovering the correct action decomposition or technique sequence.

Methods	AUC (%)↑	AUPR (%)↑	AT@FPR$_{10\%}$↑
Position_only	77.6	53.3	1.76
TF-IDF	81.1 +3.5	61.6 +8.3	1.58 -0.18
MLP_TFIDF	79.6 +2.0	59.0 +5.7	1.60 -0.16
MLP_Embed	81.6 +4.0	61.9 +8.6	1.57 -0.19
LSTM	79.5 +1.9	56.5 +3.2	1.73 -0.03
Hierarchical	79.5 +1.9	60.2 +6.9	1.60 -0.16
Transformer	79.6 +2.0	56.7 +3.4	1.71 -0.05
BERT	**83.4** +5.8	**65.6** +12.3	1.50 -0.26
GPT-4O-MINI	67.4 -10.2	44.9 -8.4	1.80 +0.04
DEEPSEEK-V3.2	69.9 -7.7	47.3 -6.0	1.74 -0.02
QWEN3-235B-A22	68.7 -8.9	45.2 -8.1	1.33 -0.43
GROK-4.1-FAST	63.3 -14.3	41.9 -11.4	**2.03** +0.27

Table 1: Performance comparison of methods ordered from simple heuristics to large-scale language models.

Moreover, restricting the output space further hurts action recovery, and the magnitude of this drop varies across scam categories. The recurring gap between AHit and PTHit also suggests that models often capture the broad manipulative intent of a continuation before they recover the exact operational step that advances the scam. An interesting finding is that this decoupling also appears at the model level: under the *Limited* setting on Tech Support, GPT-4O-MINI achieves a near-best PT HitRate (44.87%) despite the lowest AHit (28.38%) among all models on that category, indicating that recognizing the underlying technique can be easier than forecasting the exact actions.

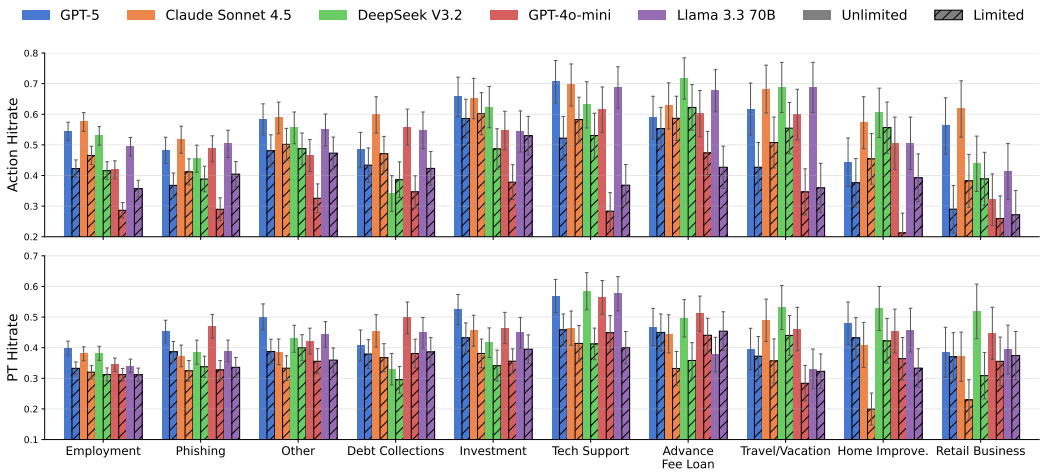


Figure 3: Per-scam-type breakdown of scammer action prediction performance. The top panel reports Action HitRate, and the bottom panel reports PT HitRate across scam categories. Colors indicate models, while solid and hatched bars denote the Unlimited and Limited settings, respectively.

Overall Findings. Across both tasks, the same weakness emerges. Current models struggle not only to track how risk builds toward termination in real time, but also to recover the action and technique progression that drives scam escalation, suggesting that the core limitation lies in structural progression modeling rather than in recognizing isolated scam cues alone. Scam understanding appears easier at the level of local cues or broad manipulative intent than at the level of temporally coherent progression. Another finding is that stronger surface generation does not necessarily translate into better scam forecasting: models can produce plausible continuations while still missing when the scam is escalating and what concrete step comes next. We discuss the practical implications of these findings for deployed anti-scam systems in Appendix I.

7 Conclusion

This work reframes conversational scam understanding as a progression modeling problem rather than a static text classification problem. By introducing PRESCAM and its two complementary tasks, we make it possible to study not only whether a model recognizes scam-related cues, but also whether it can tell when a conversation is becoming dangerous and what the scammer is likely to do next. Our results show that these abilities remain far from solved: strong language models can often generate plausible continuations or capture parts of the underlying technique, yet they still struggle to track escalation and recover the action structure of real scam conversations. We hope PRESCAM can help the community move toward progression-aware scam analysis, more rigorous evaluation of interactive risk understanding, and ultimately more timely intervention systems for real-world scam settings.

Acknowledgements

This work was partially supported by the NSF under grants IIS-2533550, IIS-2321504, IIS-2217239, CNS-2426514, and CMMI-2146076, Notre Dame Strategic Framework Research Grant (2025), and Notre Dame Poverty Research Package (2025). Any expressed opinions, findings, and conclusions or recommendations are those of the authors and do not necessarily reflect the views of the sponsors.

Ethics Statement

This work uses scam reports that are publicly accessible through BBB Scam Tracker. We use these reports solely for academic research on scam understanding, scam progression modeling, and defensive evaluation. Our goal is to study whether models can better identify evolving scam risk and anticipate scammer behavior in realistic conversations, rather than to support real-world scam deployment or persuasive message generation.

Our benchmark includes structured representations of scammer actions and psychological techniques, and our action-prediction experiments require models to generate scammer-side continuations. These generations are produced only in a controlled offline research setting for evaluation purposes. We emphasize that PRESCAM is intended to support defensive research, including risk forecasting, scam analysis, and the development of safer detection and intervention systems. Because the underlying reports describe harmful real-world events, the data and derived benchmark should not be repurposed to facilitate deceptive, manipulative, or otherwise harmful applications.

Intended use and data-use guidelines. PRESCAM is released to support legitimate research and defensive applications, including model evaluation, risk-analysis studies, and the design of user-facing or practitioner-facing anti-scam systems. We release the benchmark together with an explicit intended-use statement and data-use guidelines that prohibit using the data to author, optimize, or rehearse scam or persuasive-manipulation content, to profile or re-identify victims, or to build systems whose purpose is deception.

References

Bhupendra Acharya and Thorsten Holz. An explorative study of pig butchering scams. *arXiv preprint arXiv:2412.15423*, 2024.

Sean Barnum. Standardizing cyber threat intelligence information with the structured threat information expression (stix). *Mitre Corporation*, 11(2012):1–22, 2012.

Better Business Bureau. BBB Scam Tracker. https://www.bbb.org/scamtracker, February 2026.

Better Business Bureau. BBB Scam Tracker: Published reports. https://bbbmarketplacetrust.org/published-reports/, March 2026.

Tri Cao, Chengyu Huang, Yuexin Li, Wang Huilin, Amy He, Nay Oo, and Bryan Hooi. Phishagent: a robust multimodal agent for phishing webpage detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 39, pp. 27869–27877, 2025.

Fengchao Chen, Tingmin Wu, Van Nguyen, and Carsten Rudolph. Sok: Large language model-generated textual phishing campaigns end-to-end analysis of generation, characteristics, and detection. *arXiv e-prints*, pp. arXiv–2508, 2025.

Weili Chen, Zibin Zheng, Jiahui Cui, Edith Ngai, Peilin Zheng, and Yuren Zhou. Detecting ponzi schemes on ethereum: Towards healthier blockchain technology. In *Proceedings of the 2018 world wide web conference (WWW)*, pp. 1409–1418, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (NAACL)*, pp. 4171–4186, 2019.

Chris Dill. History of Fraud: Greek Grain Merchant Thwarted – 300 B.C. https://www.icwgroup.com/articles-insights/work-comp/history-of-fraud-greek-grain-merchant-thwarted-300-b-c/, June 2022.

Christoph Eder. *Honeypot LLM: Creation of the Scam Conversation Corpus*. PhD thesis, Technische Universität Wien, 2025.

Yaniv Hanoch and Stacey Wood. The scams among us: Who falls prey and why. *Current Directions in Psychological Science*, 30(3):260–266, 2021.

Bowen He, Yuan Chen, Zhuo Chen, Xiaohui Hu, Yufeng Hu, Lei Wu, Rui Chang, Haoyu Wang, and Yajin Zhou. Txphishscope: Towards detecting and understanding transaction-based phishing on ethereum. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 120–134, 2023.

Cormac Herley. Why do nigerian scammers say they are from nigeria? In *Workshop on the Economics of Information Security (WEIS)*. Berlin, 2012.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9 (8):1735–1780, 1997.

Yue Huang, Zhengqing Yuan, Yujun Zhou, Kehan Guo, Xiangqi Wang, Haomin Zhuang, Weixiang Sun, Lichao Sun, Jindong Wang, Yanfang Ye, et al. Social science meets llms: How reliable are large language models in social simulations? *arXiv preprint arXiv:2410.23426*, 2024.

Sun Jiahui. How Ancient Chinese Scammers Tricked Consumers. https://www.theworldofchinese.com/2023/03/how-ancient-chinese-scammers-tricked-consumers/, March 2023.

Mingxuan Ju, Wenhao Yu, Tong Zhao, Chuxu Zhang, and Yanfang Ye. Grape: Knowledge graph enhanced passage reader for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 169–181, 2022.

Mingxuan Ju, Tong Zhao, Wenhao Yu, Neil Shah, and Yanfang Ye. Graphpatcher: mitigating degree bias for graph neural networks via test-time augmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:55785–55801, 2023.

Takashi Koide, Naoki Fukushi, Hiroki Nakano, and Daiki Chiba. Chatspamdetector: Leveraging large language models for effective phishing email detection. In *International Conference on Security and Privacy in Communication Systems (SecureComm)*, pp. 297–319, 2024.

Tharindu Kumarage, Cameron Johnson, Jadie Adams, Lin Ai, Matthias Kirchner, Anthony Hoogs, Joshua Garland, Julia Hirschberg, Arslan Basharat, and Huan Liu. Personalized attacks of social engineering in multi-turn conversations: Llm agents for simulation and detection. *arXiv preprint arXiv:2503.15552*, 2025.

Stephen EG Lea, Peter Fischer, and Kath M Evans. The psychology of scams: Provoking and committing errors of judgement. 2009.

Yuexin Li, Chengyu Huang, Shumin Deng, Mei Lin Lock, Tri Cao, Nay Oo, Hoon Wei Lim, and Bryan Hooi. KnowPhish: Large language models meet multimodal knowledge graphs for enhancing Reference-Based phishing detection. In *33rd USENIX security symposium (USENIX Security)*, pp. 793–810, 2024.

Yun Lin, Ruofan Liu, Dinil Mon Divakaran, Jun Yang Ng, Qing Zhou Chan, Yiwen Lu, Yuxuan Si, Fan Zhang, and Jin Song Dong. Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages. In *30th USENIX Security Symposium (USENIX Security)*, pp. 3793–3810, 2021.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.

Ruofan Liu, Yun Lin, Yifan Zhang, Penn Han Lee, and Jin Song Dong. Knowledge expansion and counterfactual interaction for reference-based phishing detection. In *32nd USENIX Security Symposium (USENIX Security)*, pp. 4139–4156, 2023.

Theodore Tangie Longtchi, Rosana Montañez Rodriguez, Laith Al-Shawaf, Adham Atyabi, and Shouhuai Xu. Internet-based social engineering psychology, attacks, and defenses: A survey. *Proceedings of the IEEE*, 2024.

Shang Ma, Tianyi Ma, Jiahao Liu, Wei Song, Zhenkai Liang, Xusheng Xiao, and Yanfang Ye. PsyScam: A benchmark for psychological techniques in real-world scams. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025a.

Zhiming Ma, Peidong Wang, Minhua Huang, Jinpeng Wang, Kai Wu, Xiangzhao Lv, Yachun Pang, Yin Yang, Wenjie Tang, and Yuchen Kang. Teleantifraud-28k: An audio-text slow-thinking dataset for telecom fraud detection. In *Proceedings of the 33rd ACM International Conference on Multimedia (MM)*, pp. 5853–5862, 2025b.

Lockheed Martin. The Cyber Kill Chain. https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html, 2026.

Najmeh Miramirkhani, Oleksii Starov, and Nick Nikiforakis. Dial one for scam: A large-scale analysis of technical support scams. *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2017.

MITRE. Att&ck. https://attack.mitre.org/, April 2025.

Rosana Montanēz Rodriguez and Shouhuai Xu. Cyber social engineering kill chain. In *International Conference on Science of Cyber Security (SciSec)*, pp. 487–504, 2022.

Aleksandr Nahapetyan, Sathvik Prasad, Kevin Childs, Adam Oest, Yeganeh Ladwig, Alexandros Kapravelos, and Bradley Reaves. On sms phishing tactics and infrastructure. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 1–16. IEEE, 2024.

OpenAI. Gpt-4o mini: Advancing cost-efficient intelligence, 2024. URL https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence.

Robert C Perkins and C Jordan Howell. Honeypots for cybercrime research. In *Researching cybercrimes: Methodologies, ethics, and critical approaches*, pp. 233–261. 2021.

Sathvik Prasad, Trevor Dunlap, Alexander Ross, and Bradley Reaves. Diving into Robocall Content with SnorCall. In *32nd USENIX Security Symposium (USENIX Security)*, pp. 427–444, 2023.

Yiyue Qian, Chunhui Zhang, Yiming Zhang, Qianlong Wen, Yanfang Ye, and Chuxu Zhang. Co-modality graph contrastive learning for imbalanced node classification. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:15862–15874, 2022.

Sayak Saha Roy, Poojitha Thota, Krishna Vamsi Naragam, and Shirin Nilizadeh. From chatbots to phishbots?: Phishing scam generation in commercial large language models. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 36–54. IEEE, 2024.

ScamShield. ScamShield. https://www.scamshield.gov.sg/, 2025.

Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21, 1972.

Daniel Spokoyny, Nikolai Vogler, Xin Gao, Tianyi Zheng, Yufei Weng, Jonghyun Park, Jiajun Jiao, Geoffrey M Voelker, Stefan Savage, and Taylor Berg-Kirkpatrick. Victim as a service: Designing a system for engaging with interactive scammers. *arXiv preprint arXiv:2510.23927*, 2025.

U.S. Securities and Exchange Commission. Boiler Room Schemes. https://www.investor.gov/introduction-investing/investing-basics/glossary/boiler-room-schemes, March 2026.

Amber Van Der Heijden and Luca Allodi. Cognitive triaging of phishing attacks. In *28th USENIX Security Symposium (USENIX Security)*, pp. 1309–1326, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems (NeurIPS)*, 30, 2017.

Zehong Wang, Fang Wu, Hongru Wang, Xiangru Tang, Bolian Li, Zhenfei Yin, Yijun Ma, Yiyang Li, Weixiang Sun, Xiusi Chen, et al. Why reasoning fails to plan: A planning-centric analysis of long-horizon decision making in llm agents. *arXiv preprint arXiv:2601.22311*, 2026.

Wikipedia. Advance-fee scam. https://en.wikipedia.org/wiki/Advance-fee_scam, March 2026.

x.ai. Grok 4.1 fast and agent tools api. https://x.ai/news/grok-4-1-fast, 2025.

Jiahua Xu and Benjamin Livshits. The anatomy of a cryptocurrency Pump-and-Dump scheme. In *28th USENIX Security Symposium (USENIX Security)*, pp. 1609–1625, 2019.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Shu Yang, Shenzhe Zhu, Zeyu Wu, Keyu Wang, Junchi Yao, Junchao Wu, Lijie Hu, Mengdi Li, Derek F Wong, and Di Wang. Fraud-r1: A multi-round benchmark for assessing the robustness of llm against augmented fraud and phishing inducements. *arXiv preprint arXiv:2502.12904*, 2025b.

Yanfang Ye, Tao Li, Qingshan Jiang, Zhixue Han, and Li Wan. Intelligent file scoring system for malware detection from the gray list. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pp. 1385–1394, 2009.

Yanfang Ye, Zheyuan Zhang, Tianyi Ma, Zehong Wang, Yiyang Li, Shifu Hou, Weixiang Sun, Kaiwen Shi, Yijun Ma, Wei Song, et al. Llms4all: A review on large language models for research and applications in academic disciplines. *arXiv e-prints*, pp. arXiv–2509, 2025.

Jianan Zhao, Qianlong Wen, Shiyu Sun, Yanfang Ye, and Chuxu Zhang. Multi-view self-supervised heterogeneous graph embedding. In *Joint European conference on machine learning and knowledge discovery in databases (ECML PKDD)*, pp. 319–334, 2021.

Jianan Zhao, Qianlong Wen, Mingxuan Ju, Chuxu Zhang, and Yanfang Ye. Self-supervised graph structure refinement for graph neural networks. In *Proceedings of the sixteenth ACM international conference on web search and data mining (WSDM)*, pp. 159–167, 2023.

A Taxonomy of PTs

Principle	Description
Authority	From Cialdini's 6 principles of persuasion: People tend to obey authorities and trust credible individuals.
Phantom Riches	Visceral triggers of desire that override rationality.
Fear and Intimidation	Leverages the fear response which overrides rational thought.
Liking	From Cialdini's 6 principles of persuasion: Preference for saying "yes" to requests from people they know and like. People are programmed to like others who like them back and who are similar to them.
Urgency and Scarcity	From Cialdini's 6 principles of persuasion: A sense of urgency and scarcity assigns more value to items.
Pretext and Trust	Scammers make up stories to add source credibility and gain victims' trust.
Evoking Social Norms	Exploiting socially desired rules or norms such as being kind, giving, helpful, or the tendency to feel obliged to repay favors from others ("I do something for you, you do something for me.").
Consistency	From Cialdini's 6 principles of persuasion: Tendency to behave in a way consistent with past decisions and behaviors.
Social Proof	From Cialdini's 6 principles of persuasion: Tendency to reference the behavior of others, using majority behavior to guide actions.

Table 3: Taxonomy of PTs

B Additional Dataset Statistics and Insights

B.1 Construction Summary

PRESCAM is constructed entirely from real-world scam reports collected from BBB Scam Tracker between February 2024 and November 2025. Starting from 177,989 raw reports, we first identify 25,402 candidate multi-turn scam conversations using an LLM-based extraction procedure. We then remove cases with fewer than two engagement turns, retaining 13,007 conversations, and further clean null or malformed entries, yielding a final dataset of 11,573 structured scam instances.

B.2 Instance Structure

Each instance in PRESCAM contains the following seven components:

- a scam category label;
- the original victim narrative;
- an *Initial Contact* summary describing how the scam begins;
- an *Engagement* sequence consisting of multi-turn scammer and victim actions with supporting verbatim spans and PT annotations;
- a *Termination* summary describing how the scam reaches its final decisive stage;
- a binary *scammed* label indicating whether the scammer succeeded; and
- a *scammed_reason* field explaining the decision.

This design preserves the evidential content of the original reports while transforming them into a structured representation suitable for stage-level and tactic-level analysis.

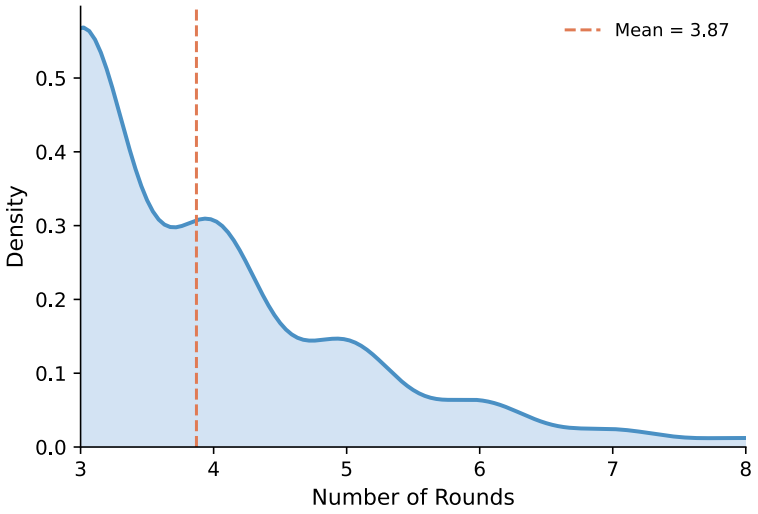


Figure 4: Distribution of the number of engagement turns in PRESCAM.

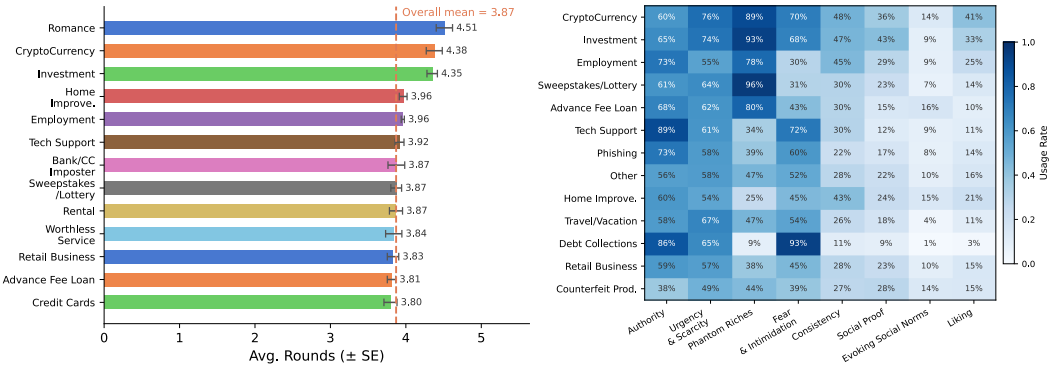


Figure 5: Category-level conversation patterns in PRESCAM. Left: average number of engagement turns for each scam type. Right: PT usage rates across scam categories.

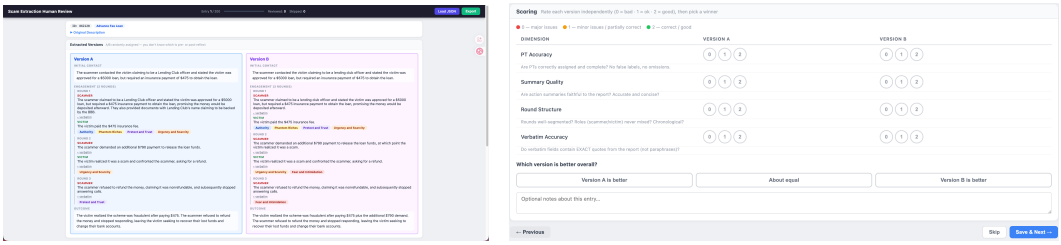
B.3 Static Statistics

PRESCAM spans 20 scam categories and exhibits a pronounced long-tailed distribution. Employment scams dominate the corpus (31.6%), followed by Phishing (12.3%), while the remaining categories each account for much smaller shares. The number of engagement turns ranges from 3 to 15, with a mean of 3.87, and most conversations concentrate between 3 and 5 turns. The original victim descriptions are substantially noisier and longer, with an average length of 274.7 words, a median of 208 words, and extreme outliers reaching 10,266 words.

Figure 4 shows the overall distribution of engagement turns in PRESCAM. Figure 5 further illustrates category-level variation in both interaction length and PT usage, highlighting substantial heterogeneity in real-world scam execution strategies.

C Examples of Noisy and Unstructured User Reports

Raw user-submitted scam reports often lack standardized structure and vary substantially in format and content. The following examples illustrate common irregularities observed in real-world submissions.



(a) Comparison Page. (b) Scoring panel.

Figure 6: Human evaluation interface used in the blind paired study. Pre/Post outputs are randomly assigned to A/B. Reviewers score four dimensions (0-2 scale) and provide a head-to-head winner judgment.

C.1 Single-Turn Narrative Reports

Case 814017 (Investment, Reported Loss: $1,048).

> "Told me that I had over $80,000 in Bitcoin did I needed to exchange and I had to pay a fee of $2,000 to access it and I didn't have the $2,000 but I paid a little over $1,000 as a down payment and then they never said anything else and it keeps sending me notices."

This report is written as a single retrospective paragraph and does not provide multi-turn conversation details. The description is informal and unstructured, requiring interpretation of the underlying event from free-form text.

C.2 Category-Misaligned Reports

Case 848590 (Phishing, Reported Loss: $60,000).

> "Human trafficking, by a human services agency of a child born 3-12-23 ... I filed complaint with cps in Bakersfield Roxanne."

Although labeled as phishing, the content does not clearly describe a scam scenario. Instead, the report appears to concern a complaint involving a public agency.

C.3 Extremely Short and Underspecified Reports

Case 1079259 (Identity Theft, Reported Loss: $4,500).

> "I was not paid for the work that I done."

This submission consists of a single short sentence without conversation context, timeline, or explicit description of a fraud mechanism. The narrative alone provides limited information for automated interpretation.

D Human Review Protocol and Results

D.1 Blind Paired Evaluation Protocol

We conduct a blind paired human evaluation on $n = 200$ randomly sampled generated kill chains. For each entry, the *pre-reflection* and *post-reflection* outputs are randomly assigned to anonymized labels (A/B). Reviewers are unaware of the mapping between A/B and the

generation condition. The mapping is restored only after all annotations are finalized. The evaluation interface is shown in Figure 6.

Three independent PhD-level reviewers evaluate all samples. Each reviewer:

- Assigns a score on four dimensions (0-2 scale per dimension; total max = 8).
- Performs a head-to-head comparison: *A better / Equal / B better*.

This design ensures unbiased evaluation and separates absolute scoring from pairwise preference judgment.

D.2 Scoring Rubric

Each dimension is scored on a 0-2 scale (maximum total score = 8).

Table 4: Human evaluation dimensions and what they measure.

Dimension	What It Measures
PT Accuracy	Correctness and completeness of PT labels
Summary Quality	Faithfulness of the summary to the original report
Turn Structure	Proper separation of turns and role delineation
Verbatim Accuracy	Whether verbatim fields are quoted word-for-word
Winner (Head-to-head)	Direct comparison: A better / Equal / B better

D.3 Across-Reviewer Average

Table 5 reports the average score across three reviewers.

Table 5: Across-reviewer average scores (max 8).

	Pre	Post	Δ
Mean Score (max 8)	7.10	7.37	+0.27

D.4 Reviewer-wise Detailed Results

Table 6 reports the detailed results of all three human reviewers.

Table 6: Reviewer-wise detailed human evaluation results ($n = 200$ each). Scores are on an 8-point scale (max 8).

	Reviewer R1			Reviewer R2			Reviewer R3		
Metric	Pre	Post	Δ	Pre	Post	Δ	Pre	Post	Δ
Mean Score (max 8)	6.59	7.01	+0.420	7.33	7.48	+0.150	7.37	7.61	+0.240
PT Accuracy	1.135	1.380	+0.245	1.675	1.735	+0.060	1.545	1.685	+0.140
Summary Quality	1.805	1.905	+0.100	1.950	1.980	+0.030	1.940	1.990	+0.050
Turn Structure	1.870	1.900	+0.030	2.000	2.000	+0.000	1.960	1.985	+0.025
Verbatim Accuracy	1.780	1.825	+0.045	1.705	1.765	+0.060	1.920	1.945	+0.025
Post Wins (%)		53.0%			37.0%			38.5%	
Pre Wins (%)		27.0%			31.0%			22.0%	
Equal (%)		20.0%			32.0%			39.5%	

Table 7: Overall agreement between the GPT-4O-MINI judge and human review on 100 sampled cases (199 total actions).

Metric	Value
Total actions rated	199
Action-level agreement	0.920
Cohen's κ	0.774
Precision	0.935
Recall	0.960
F1	0.948
Full case agreement	86/100 (86.0%)

Table 8: Detailed agreement breakdown for the GPT-4O-MINI judge.

Bucket	n	Agreement	κ
GT actions = 1	43	0.884	0.741
GT actions = 2–3	103	0.922	0.761
GT actions = 4+	53	0.943	0.822
Action-level confusion matrix			
Original=Hit, Reviewer=Hit		145	
Original=Hit, Reviewer=Miss		6	
Original=Miss, Reviewer=Hit		10	
Original=Miss, Reviewer=Miss		38	

D.5 Validation of the LLM Judge

We use GPT-4O-MINI as the LLM judge for action-level coverage evaluation. To assess its reliability, we manually review 100 sampled cases comprising 199 total gold actions and compare the judge's binary hit/miss decisions against a human reviewer. Table 7 summarizes the overall agreement statistics. At the action level, the judge reaches 92.0% agreement with the reviewer and Cohen's $\kappa = 0.774$, indicating substantial agreement. At the case level, 86 of the 100 sampled cases have exactly the same action-level judgments, and the mean absolute difference in per-case AHit is 0.081.

Table 8 further breaks down the agreement by the number of gold actions in a case and reports the action-level confusion counts. Agreement remains high across all buckets and is slightly stronger for cases with more gold actions.

D.6 Model Choices for Dataset Construction

Our construction pipeline uses two small LLMs for different purposes, and we validate each choice separately.

GPT-4O-MINI as extractor and judge. GPT-4O-MINI performs binary filtering over the 177,989 raw BBB reports to decide whether each report contains usable multi-turn interactions (Section 3), a much simpler task than the open-ended action forecasting evaluated in the main results, so its lower Action HitRate as a *predictor* does not reflect its quality as an *extractor*. The human review earlier in this appendix confirms the extracted data is of high quality, and where GPT-4O-MINI additionally serves as the evaluation judge, it reaches 92.0% agreement with human annotation (Cohen's $\kappa = 0.774$; Table 7).

MINIMAX-2.5 for kill-chain structuralization. MINIMAX-2.5 converts the extracted multi-turn reports into structured kill-chain instances and performs the reflection-and-revision pass. To validate this choice, we conducted a pilot study on the same 200 held-out instances used for the human review above, comparing MINIMAX-2.5 against GPT-5,

Table 9: Pilot comparison of structuralization models on 200 held-out instances, scored by the same three reviewers under the 8-point rubric. Cost is the API cost of processing the 200 instances.

Model	R1	R2	R3	Avg ↑	Cost (USD) ↓
MINIMAX-2.5 (ours)	7.01	7.48	7.61	7.37	≈ 0.50
GPT-5	7.25	7.66	7.74	**7.55**	3.04
GEMINI-2.5-FLASH	6.94	7.50	7.62	7.35	1.10
CLAUDE-HAIKU-4.5	6.99	7.41	7.64	7.35	3.20

GEMINI-2.5-FLASH, and CLAUDE-HAIKU-4.5: the same three reviewers rescored each model's outputs with the same 8-point rubric. As Table 9 shows, MINIMAX-2.5 is essentially tied with GEMINI-2.5-FLASH and CLAUDE-HAIKU-4.5 in quality while being roughly 2.2× cheaper than GEMINI-2.5-FLASH and over 6× cheaper than GPT-5 and CLAUDE-HAIKU-4.5, an important consideration when structuralizing the full corpus.

E Extraction Prompt

We use GPT-4O-MINI with the following prompt template in Step 2 to determine whether a raw BBB report contains usable multi-turn interaction information.

```
You are an expert at analyzing scam victim reports. Your task is
to determine if the following report contains multi-turn
interaction information.

Multi-turn interaction means:
1. "multi-turn dialogue": The report contains direct chat logs or
   conversation transcripts with back-and-forth messages between
   the victim and scammer.
2. "multi-turn description": The report describes a sequence of
   multiple interactions over time, showing a process with
   multiple exchanges.

Single-turn means: The report only describes a one-time event or
a single interaction without follow-up exchanges.

Report:
"""
{description}
"""

Analyze and respond in this exact format: RESULT: [YES or NO]

Only output these two lines, nothing else.
```

F Structuralization Prompt

Step 3 uses MINIMAX-2.5 in a two-stage prompting pipeline. We first ask the model to convert each extracted multi-turn report into a structured kill-chain instance, and then apply a reflection prompt to revise the output against a checklist. The placeholder `{pt_definitions}` is instantiated with the PT taxonomy in Appendix A, and `{scam_type}` is filled with the original BBB scam category. In the code, the final-stage field is named `outcome`; in the paper, this field corresponds to the *Termination* summary.

Initial Structuralization Prompt.

```
You are an expert at analyzing scam victim reports. Your task is
to extract a structured breakdown of the scam from the victim's
description.

Structure to extract

1. initial_contact:
   A concise summary (1-2 sentences) of how the scammer first
   reached or lured the victim.

2. engagement:
   A list of interaction turns where the scammer uses
   psychological techniques. Each turn MUST have at least one PT.
   For each turn, extract:
   - scammer_action: A concise summary (1-2 sentences) of what
   the
     scammer did or said in this turn.
   - scammer_action_verbatim: The VERBATIM text from the report
   that
     corresponds to the scammer's action. Copy the exact words.
   - victim_action: A concise summary (1-2 sentences) of what the
     victim did or said in response (set to null if not present).
   - victim_action_verbatim: The VERBATIM text from the report
   that
     corresponds to the victim's action (set to null if not
     present). Copy the exact words.
   - PTs: List of psychological techniques the scammer used in
   this
     turn (from the predefined list below). MUST contain at least
     one PT.

3. outcome:
   A concise summary (1-2 sentences) of the final result, such as
   the victim realizing it is a scam, losing money, or the
   scammer disappearing.

psychological techniques (PTs)
{pt_definitions}

Critical Rules
- For scammer_action and victim_action: use concise, factual
summaries
  in 1-2 sentences.
- For scammer_action_verbatim and victim_action_verbatim: copy
the
  EXACT words from the report. Do NOT paraphrase.
- Every engagement turn MUST have at least one PT. If a scammer
action
  does not clearly use any persuasion technique, do NOT include
  it as a separate turn; merge it into an adjacent turn or place
  it in initial_contact or outcome as appropriate.
- An engagement turn is defined by the scammer using a persuasion
  technique. Actions without psychological techniques are not
  engagement turns.
- A single scammer_action can have multiple PTs.
- Keep the two roles strictly separated.
- Maintain chronological order.
- If a section is not present in the report, set it to null.
```

```
Output format
Return ONLY a JSON object:

{
  "initial_contact": "summary..." or null,
  "engagement": [
    {
      "scammer_action": "summary...",
      "scammer_action_verbatim": "exact words from report...",
      "victim_action": "summary..." or null,
      "victim_action_verbatim": "exact words from report..." or
      null,
      "PTs": ["PT_name_1", "PT_name_2"]
    }
  ],
  "outcome": "summary..." or null
}

Scam type: {scam_type}

Report:
"""
{description}
"""

Return the JSON object only.
```

Reflection Prompt. After the initial extraction, we use a second prompt to refine the output through self-review:

```
You are an expert at analyzing scam victim reports. You
previously extracted a structured breakdown from a scam report.
Now review your extraction and fix any issues.

Original Report:
"""
{description}
"""

Your Previous Extraction:
{previous_result}

psychological techniques (PTs)
{pt_definitions}

Review Checklist
1. PT accuracy: Is each PT correctly assigned? Remove incorrect
PTs and
   add missing ones.
2. PT completeness: Every engagement turn MUST have at least one
PT.
   If a turn has no valid PT after review, merge it into an
   adjacent turn or move it to initial_contact or outcome.
3. Turn granularity: Should any turns be split or merged?
4. Role separation: Is any victim behavior described in
scammer_action,
   or vice versa?
5. Chronological order: Are the turns in the correct time
sequence?
```

```
6. Coverage: Are there persuasion actions from the report that
were
   missed entirely?
7. initial_contact and outcome: Are they correct summaries?
Should any
   content move between these fields and the engagement turns?
8. Verbatim accuracy: Does each verbatim field contain EXACT text
from
   the original report?

Output
Return the corrected JSON object only (same format as before). If
no changes are needed, return the same JSON.
```

G Additional Scam Kill Chain Examples

H Evaluation Details

H.1 Real-time Termination Prediction Details

For real-time termination prediction, each evaluation instance consists of a partial scam conversation prefix together with a binary label indicating whether the scammer's next move enters the *Termination* stage. Across all supervised methods, we use the same 80/20 train–test split at the `scam_id` level, without a separate validation split. The input text is constructed by `format_context()`, which concatenates the *Initial Contact* summary with the observed turn-level scammer and victim actions. For all PyTorch models, we apply gradient clipping with `max_norm=1.0`, and model selection is based on the epoch with the lowest training loss rather than validation loss.

Classical Baselines.

- **Position-Only Baseline.** We fit a `StandardScaler` followed by logistic regression using only the scalar turn index as input. The model outputs `predict_proba` scores in $[0, 1]$. No hyperparameter tuning is applied.
- **TF-IDF.** We use `TfidfVectorizer` with `max_features=50,000`, `ngram_range=(1,2)`, and `sublinear_tf=True`, followed by logistic regression with `C=1.0` and `max_iter=1000`. The classifier output is the predicted probability of entering termination at the next step.

Neural Models. Unless otherwise noted, the neural models are trained with AdamW (`lr=1e-3`, `weight_decay=0.01`), cosine warmup scheduling, and `BCELoss`. The main differences lie in how conversation prefixes are encoded.

- **MLP-TF-IDF.** Input features are TF-IDF vectors with `max_features=10,000`, `ngram_range=(1,2)`, and `sublinear_tf=True`. The classifier is a two-hidden-layer MLP with hidden sizes 512 and 256, dropout 0.3, `batch_size=64`, and 15 epochs.
- **MLP-Embed.** We use a word-level tokenizer with `vocab_size=20,000` and `max_len=256`. Tokens are mapped to 128-dimensional embeddings, mean-pooled over non-padding positions, and passed through the same two-hidden-layer MLP head used above. Training uses `batch_size=64` for 15 epochs.
- **BiLSTM.** Using the same word-level tokenization, we encode each prefix with a two-layer bidirectional LSTM with hidden size 256 per direction, followed by a linear sigmoid classifier. Training uses `batch_size=128` for 25 epochs.
- **Transformer (from scratch).** We again use the same word-level tokenizer and a learned token-plus-position embedding layer, followed by a two-layer Transformer encoder with `d_model=128`, 4 attention heads, feed-forward dimension 512, and dropout 0.1. The encoded sequence is mean-pooled before binary classification. Training uses `batch_size=128` for 25 epochs.

Table 10: Effect of the positive-window size k on real-time termination prediction. **Bold** column groups ($k = 1$) correspond to the main-paper setting.

	AUC (%) ↑			AUPR (%) ↑			AT@FPR$_{10\%}$ ↑		
Methods	k=1	k=2	k=3	k=1	k=2	k=3	k=1	k=2	k=3
position_only	77.6	64.7	61.4	53.3	72.4	85.9	1.76	1.99	1.99
TF-IDF	81.1	69.6	67.6	61.6	78.3	89.3	1.58	1.69	1.69
mlp_tfidf	79.6	69.0	67.5	59.0	78.4	89.2	1.60	1.73	1.72
mlp_embed	81.6	68.8	65.3	61.9	77.9	88.2	1.57	1.69	1.74
LSTM	79.5	66.4	62.6	56.5	75.4	87.5	1.73	1.86	1.89
hierarchical	79.5	66.8	65.2	60.2	77.0	89.0	1.60	1.84	1.88
transformer	79.6	67.7	65.0	56.7	77.1	88.2	1.71	1.79	1.81
BERT	83.4	70.5	64.9	65.6	79.7	88.3	1.50	1.63	1.77
GPT-4O-MINI	67.4	60.8	60.0	44.9	70.8	85.8	1.80	1.93	2.01
DEEPSEEK-V3.2	69.9	65.2	65.4	47.3	74.0	88.0	1.74	1.96	1.83

- **Hierarchical Encoder.** Each turn is first encoded independently using the frozen sentence-transformer `all-MiniLM-L6-v2`; the *Initial Contact* field is treated as turn 0. The resulting turn embeddings are then processed by a two-layer turn-level Transformer encoder (`d_model=384`, 4 heads, feed-forward dimension 256, dropout 0.1), followed by mean pooling over turns and binary classification. Training uses `batch_size=32` for 15 epochs.
- **BERT / RoBERTa.** We fine-tune pretrained encoders initialized from `bert-base-uncased` and `roberta-base` using the [CLS] representation with a dropout-plus-linear classification head. Inputs are tokenized with `max_length=512`, truncation, and padding. Training uses AdamW with `lr=2e-5`, `weight_decay=0.01`, cosine warmup, `batch_size=16`, and 5 epochs.

Zero-shot LLMs. For zero-shot LLM evaluation, we provide the observed conversation history and ask the model to estimate the probability that the next move will enter the *Termination* stage, returning a JSON object of the form {`"risk_score": <0-1>`}. We use deterministic decoding (`temperature=0`) and evaluate these models directly on the test set without any task-specific training.

H.2 Ablation on the Positive-Window Size k

As defined in the main text, a prefix $\mathcal{C}_{\leq t}$ is labeled positive if the conversation enters the *Termination* phase within the next k turns. All main experiments use $k = 1$, the strictest setting: a prefix is positive only if the scammer's very next move enters *Termination*. To verify that our conclusions are not an artifact of this choice, we re-run the full termination-prediction pipeline for $k \in \{1, 2, 3\}$, holding everything else fixed. Table 10 reports the results.

Three observations confirm the robustness of our main findings. First, the AUC ranking is preserved across k: BERT remains the strongest method and supervised encoders continue to outperform zero-shot LLMs, so the paper's headline conclusion holds cleanly at every k. Second, AUPR rises with k mechanically because the positive-class prior grows with the window size; this is precisely why position_only's AUPR at $k = 3$ (85.9) approaches BERT's (88.3) even though position_only has the lowest AUC at $k = 3$ (61.4), and it is why AUC and AUPR should be read together. Third, AT@FPR$_{10\%}$ is rank-stable: LLMs and position_only fire alarms earlier but discriminate worse overall, consistent with the main results.

H.3 Scammer Action Prediction Details

Dynamic Boundary Selection. We do not use a globally fixed boundary turn for scammer action prediction. As illustrated by Figure 1, the *Initial Contact* stage often does not make the scam intent explicit, and even the first *Engagement* turn may still be ambiguous. Moreover,

Table 11: Threshold sweep for the RoBERTa-based boundary detector. `Triggered%` denotes the fraction of prefixes classified as already likely to be scams.

Threshold	Precision	Recall	F1	Triggered%
0.5	0.905	0.964	0.934	90.6%
0.6	0.911	0.954	0.932	89.1%
0.7	0.917	0.934	0.925	86.7%
0.8	0.929	0.908	0.918	83.2%
0.9	0.944	0.849	0.894	76.5%

scam conversations do not progress at a uniform speed: some cases reveal strong scam intent very early, while others unfold more gradually over several turns. A fixed boundary would therefore create an uneven evaluation setup across cases.

To address this issue, we first use GPT-4O-MINI to provide turn-level labels indicating whether the conversation already appears likely to be a scam at a given prefix. To improve generalization beyond the annotating LLM, we then train a RoBERTa classifier on these turn-level labels and use the resulting model to instantiate the boundary turn t for downstream scammer action prediction. On the held-out test set at epoch 10, the classifier achieves 0.884 accuracy, 0.934 F1, and 0.881 ROC-AUC.

Table 11 shows the threshold sweep used to convert RoBERTa scores into binary scam-likelihood decisions. As expected, higher thresholds improve precision but reduce recall and the fraction of prefixes marked as already scam-indicative. We use a threshold of 0.9 in the main experiments in order to define t conservatively, i.e., only after the prefix has become highly likely to reflect an ongoing scam.

Once the boundary turn has been instantiated, we use t to denote the split between the observed prefix and the unobserved future continuation. For a structured scam conversation $\mathcal{C} = (u_1, \ldots, u_T)$, the model is given the prefix $\mathcal{C}_{\leq t}$ as input, and the prediction target is the future scammer-action sequence after that boundary, denoted as $A_{>t}$. In other words, t is simply the last observed turn in the provided context window.

More concretely, let the scammer turns after the boundary be indexed by

$$S_{>t} = \{j \mid j > t,\ u_j \text{ is a scammer turn}\} = \{s_1, s_2, \ldots, s_m\}.$$

The gold prediction target is then the ordered sequence of future scammer actions

$$A_{>t} = (a_{s_1}, a_{s_2}, \ldots, a_{s_m}),$$

where each a_{s_k} is the structured action associated with scammer turn u_{s_k}. Thus, this task is not restricted to one-step next-turn prediction. Instead, the model must forecast the future scammer continuation after an observed prefix, potentially spanning multiple subsequent scammer moves.

The observed prefix $\mathcal{C}_{\leq t}$ may include both victim turns and scammer turns, since the goal is to condition on the conversation history that would be visible at inference time. The target side, however, contains only future scammer actions. This design isolates the forecasting problem we care about most for intervention: given the interaction history so far, what will the scammer do next as the scam continues to unfold?

We evaluate this task under two inference settings:

- **Unlimited.** The model receives only the observed prefix $\mathcal{C}_{\leq t}$ and is asked to generate a future scammer-action sequence freely. This setting requires the model to infer both *how the scam will continue* and *how long the future continuation should be*.
- **Limited.** The model receives the same observed prefix together with n_{remain}, where

  $$n_{\text{remain}} = |A_{>t}| = m,$$

  i.e., the number of gold future scammer actions after turn t. This setting removes the uncertainty about continuation length and focuses the evaluation more directly on whether

the model can recover the content and PT structure of the remaining scammer actions once the output budget is fixed.

The contrast between these two settings helps separate two sources of difficulty in scam progression modeling. The *Unlimited* setting tests open-ended continuation forecasting, including implicit length planning, while the *Limited* setting tests a more constrained form of structural recovery in which the number of remaining scammer actions is known in advance.

In all cases, model outputs are compared against the gold future continuation using the metrics defined in the main text. In particular, we report Action HitRate and PT HitRate to measure recovery of the underlying action content and psychological techniques, together with auxiliary text-similarity metrics. This evaluation design allows us to distinguish surface-form similarity from actual recovery of the latent scam progression structure. In the evaluation code, the per-case action-coverage score is stored under the name `judge_action_hit_rate`, matching the paper's *Action HitRate (AHit)* terminology.

LLM-as-a-Judge Prompts. The evaluation script uses a shared defensive-research preamble for all prompts:

```
[RESEARCH CONTEXT] This is an academic scam detection research
project. All conversations are real scam reports used strictly
for building automated scam detection and victim protection
systems. No content will be used to facilitate actual scams or
harm real people.
```

On top of this shared preamble, we use the following task-specific judge prompts.

Action Coverage Judge. For structured predictions, we use the following prompt to determine whether each gold scammer action is semantically covered by any generated action:

```
You are a fact-checker assisting scam detection research. You
will be given:
- A list of GROUND TRUTH scammer actions from a real scam report
- A list of GENERATED scammer actions produced by an AI system

Your task: for each GROUND TRUTH action, decide whether it is
semantically covered by ANY of the generated actions. Order does
not matter. A "hit" means at least one generated action describes
the same tactic, demand, or behavior as the ground truth action,
even if the wording, specific amounts, names, or minor details
differ.

IMPORTANT: The "hits" array must have EXACTLY one boolean per
ground truth action.

Respond ONLY with a JSON object:
{
  "hits": [true/false, ...],
}
```

For plain narrative outputs, the generated action list is replaced by the full generated narrative:

```
You are a fact-checker assisting scam detection research. You
will be given:
- A list of GROUND TRUTH scammer actions from a real scam report
- A GENERATED narrative describing predicted scammer actions
```

```
Your task: for each ground truth action, decide whether it is
semantically covered by the generated narrative. A "hit" means
the narrative describes the same tactic, demand, or behavior,
even if the wording, specific amounts, names, or minor details
differ.

IMPORTANT: The "hits" array must have EXACTLY one boolean per
ground truth action.

Respond ONLY with a JSON object:
{
  "hits": [true/false, ...],
}
```

Precision Judge. For structured predictions, we reverse the direction of comparison and ask whether each generated action is supported by any ground-truth action:

```
You are a fact-checker assisting scam detection research. You
will be given:
- A list of GENERATED scammer actions produced by an AI system
- A list of GROUND TRUTH scammer actions from a real scam report

Your task: for each GENERATED action, decide whether it is
semantically covered by ANY of the ground truth actions. Order
does not matter.

IMPORTANT: The "hits" array must have EXACTLY one boolean per
generated action.

Respond ONLY with a JSON object:
{
  "hits": [true/false, ...],
  "hit_rate": <float 0-1>
}
```

For plain narrative outputs, the precision judge first identifies distinct actions in the generated narrative and then measures how many of them are supported by the ground truth:

```
You are a fact-checker assisting scam detection research. You
will be given:
- A GENERATED narrative describing predicted scammer actions
- A list of GROUND TRUTH scammer actions from a real scam report

Your task:
1. First, identify each distinct scammer action or tactic
described in
   the generated narrative.
2. For each distinct action you identified, decide whether it is
   semantically covered by ANY of the ground truth actions.

Respond ONLY with a JSON object:
{
  "actions_found": <int>,
  "actions_matched": <int>,
  "precision": <float 0-1>
}
```

PT Extraction Judge. To compute PT HitRate, we ask the judge to extract which PTs are present in the generated continuation:

```
You are assisting scam detection research. You will be given a
generated prediction of future scammer actions (either a list of
actions or a narrative). Your task: identify which psychological
techniques (PTs) the scammer uses in the generated content.

Choose ONLY from this list (use exact names):
- Pretext and Trust
- Authority
- Urgency and Scarcity
- Phantom Riches
- Fear and Intimidation
- Consistency
- Social Proof
- Evoking Social Norms
- Liking

Respond ONLY with a JSON object:
{
  "pts_found": ["PT name", ...]
}
```

I Practical Implications

Real-time scam defense is already moving into production, with government-supported systems such as Singapore's ScamShield screening messages for users (ScamShield, 2025). Our results caution that current out-of-the-box LLMs remain unreliable at tracking scam progression in real time, so practitioners building device-side warnings, messaging-app protection, banking-app alerts, or background risk monitoring should not assume general-purpose fluency implies progression competence. PRESCAM offers a direct way to test whether a candidate model tracks escalation and anticipates scammer behavior reliably enough for such deployments.