

Who Plays to Earn? Detecting Bots in Web3 Games via On-Chain Transactions Analysis

Woncheol Jeong ✉ 

Korea University, Seoul, Republic of Korea

Sohee Choi ✉

Korea University, Seoul, Republic of Korea

Jiwon Yoon¹ ✉ 

Korea University, Seoul, Republic of Korea

Abstract

Bots in Web3 Play-to-Earn (P2E) games extract on-chain rewards at scale, accelerating token inflation and eroding the incentives that sustain legitimate participation. Existing defenses transfer poorly to this setting. Traditional game-bot detectors require proprietary server-side telemetry that is unavailable on permissionless chains, and blockchain-native detectors target DeFi behaviors whose footprints differ fundamentally from those of game automation; research specific to Web3 game bots remains scarce, and no public labeled dataset exists. We argue that the very economic asymmetry that makes P2E automation profitable also exposes it, because the temporal, topological, financial, and behavioral regularities that bots cannot disguise without forfeiting their efficiency advantage remain observable from public on-chain transactions alone. Building on this insight, we propose a cross-game, cross-chain detection framework that models player activity as a heterogeneous temporal transaction graph over a unified five-type edge vocabulary, extracts 32 chain-agnostic behavioral features, and classifies accounts through a dual-branch encoder fusing a typed-edge graph attention network (GAT) with a gated recurrent unit (GRU)-attention sequence model, bootstrapped by graph-contrastive pre-training. To support evaluation, we further release the first multi-game labeled bot dataset, spanning three P2E titles on three heterogeneous chains with 57,879 accounts and 3,551 confirmed bots. On the joint test split, the framework attains $F1 = 94.39\%$, $AUC = 99.92\%$, and $AP = 98.90\%$, with a sharply bimodal score distribution and stable performance across a wide range of decision thresholds.

2012 ACM Subject Classification Security and privacy → Intrusion/anomaly detection and malware mitigation; Computing methodologies → Neural networks; Information systems → Data mining

Keywords and phrases Bot detection, Web3 gaming, Play-to-Earn, on-chain behavioral analysis, transaction graph analysis, heterogeneous graph neural network

Digital Object Identifier 10.4230/LIPIcs.AFT.2026.32

Supplementary Material *Dataset:* <https://github.com/WONCHEOL-JEONG/Dataset-for-Detecting-Bots-in-Web3-P2E-Games.git> [8]

archived at `swb:1:dir:2eef651a0c2d25b011e0754a2f62e4784d985072`

Funding This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. RS-2024-00460321, Development of Digital Asset Transaction Tracking Technology to Prevent Malicious Financial Conduct in the Digital Asset Market). This work was also supported by Korea Internet & Security Agency(KISA) grant funded by the Korea government(PIPC) (No. 2780000036, Masters and Doctoral Program for Advanced Talent Development in Personal Data Protection and Responsible Use).

¹ Corresponding author



© Woncheol Jeong, Sohee Choi, and Jiwon Yoon;
licensed under Creative Commons License CC-BY 4.0

8th Conference on Advances in Financial Technologies (AFT 2026).

Editors: Aggelos Kiayias and Maria Kyrpoulou; Article No. 32; pp. 32:1–32:22

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Web3 Play-to-Earn (P2E) games allow players to earn cryptocurrencies and NFTs by performing gameplay actions recorded on public blockchains [26]. This direct coupling of gameplay and financial return has made P2E ecosystems a lucrative target for automated exploitation: bots farm rewards around the clock, accelerating token inflation, depressing per-player yields, and eroding the trust of legitimate participants [1]. Existing detection approaches are ill-suited to this setting: traditional game-bot detectors rely on proprietary server-side signals such as self-similarity measures and behavior sequences [10, 23] that are unavailable in permissionless blockchain environments; blockchain-native anomaly detectors target DeFi-specific behaviors [2, 14] that differ fundamentally from those of game bots; and all prior work examines a single game on a single blockchain with no public labeled dataset [6].

We observe that every economically meaningful action in a P2E game settles as a public, immutable on-chain transaction, producing behavioral traces that bots cannot disguise without sacrificing the efficiency that makes automation profitable; we substantiate this claim in Section 3.1. This economic asymmetry suggests that on-chain data alone may carry sufficient signal for effective detection.

Building on this observation, we propose a cross-game bot detection framework that operates exclusively on public on-chain data, and evaluate it across three P2E titles on three heterogeneous blockchains.

Contributions

This paper makes the following contributions:

- **Problem formulation.** We formalize Web3 game-bot detection as node classification on a heterogeneous temporal transaction graph with a unified five-type edge vocabulary abstracting across games and chains.
- **Feature design.** We propose 32 chain-agnostic behavioral features in four complementary groups (temporal regularity, graph topology, financial flow, action diversity) that capture game-bot signatures missed by DeFi-oriented feature sets.
- **Model architecture.** We design a dual-branch encoder combining typed-edge GAT with GRU-attention, integrated through gated fusion and bootstrapped by graph-contrastive pre-training.
- **Dataset and evaluation.** We construct, to our knowledge, the first multi-game labeled bot dataset spanning three P2E titles on three blockchains (57,879 accounts, 3,551 confirmed bots), and demonstrate that the framework achieves $F1 = 94.39\%$, $AUC = 99.92\%$, and $AP = 98.90\%$ on the joint test set with a sharp bimodal score separation between bots and humans.

2 Background

2.1 Play-to-Earn Game Architecture

Despite differing gameplay mechanics, P2E games share a common economic architecture. Players perform in-game actions such as mining, battling, or questing, each of which is recorded as a transaction on a public blockchain. Successful actions yield reward tokens (e.g., TLM in Alien Worlds, JEWEL in DeFi Kingdoms, DEC in Splinterlands) or NFT assets such as characters, tools, and virtual land. Players can then trade these tokens and NFTs

on in-game marketplaces or decentralized exchanges, converting gameplay outcomes into fiat-equivalent value [26]. This closed economic loop creates a direct incentive for automation, as any action that produces on-chain rewards can, in principle, be scripted.

2.2 Bot Behaviors and Their Detectability

To ground the detection problem, we describe three categories of P2E game bots. We distinguish them by their primary objective, but our main concern is the on-chain traces that make each category detectable.

Reward-farming bots repetitively execute the core reward-yielding action (e.g., mining in Alien Worlds, questing in DeFi Kingdoms) with minimal variation. Their automation is exposed on-chain through mechanical regularity in action timing, narrow action-type diversity, and rapid withdrawal of earned tokens.

Market-manipulation bots exploit marketplace mechanics through automated sniping, wash trading [19], or price manipulation. Their footprint extends beyond dense intra-cluster transaction patterns and repetitive trading sequences confined to a narrow set of counterparties. Automated market engagement produces abnormal timing regularity in listing and purchase actions; wash trading generates artificial volume, visible as high gross turnover with near-zero net change in inventory or funds; inventory cycling repeatedly re-lists and re-trades the same small set of NFTs; and coordinated activity across controlled accounts leaves reciprocal, circular value flows within a closed group, visible from each participating account's own history. Such trades are often economically irrational once marketplace fees and gas are accounted for, and systematic price-distortion strategies, for example self-dealt sales that step an asset's price upward before an outside sale, produce price trajectories decoupled from organic demand.

Sybil bots operate coordinated clusters of accounts to circumvent per-account reward limits or to farm airdrops across many controlled accounts. They leave convergent, funnel-shaped fund flows toward a small number of collection addresses, together with homogeneous behavioral profiles across the controlled accounts.

These categories are not mutually exclusive, and a single operator may combine them; we use them not as a strict classification but as a guide to the behavioral signals a detector must capture, all of which are observable from on-chain data alone. Reward-farming bots, which we expect to constitute the majority of automated accounts in P2E ecosystems, and market-manipulation bots are detectable at the level of an individual account, whereas Sybil bots are only partially so: their convergent structure and profile homogeneity are visible locally, but explicit cross-account synchronization requires cluster-level analysis (Section 3.3). The framework proposed in this work therefore targets per-account detection, capturing the first two categories in full and the locally observable component of the third.

2.3 On-Chain Observability

Although some games execute gameplay logic off-chain (e.g., Splinterlands resolves card battles on a centralized server), the economic outcomes that bots are designed to exploit, such as reward claims, NFT transfers, and marketplace trades, always settle as public, immutable transactions. Detection is therefore possible from on-chain data alone, without cooperation from game operators.

2.4 Limitations of Existing Approaches

Despite a rich body of work on game-bot detection and blockchain anomaly detection, none of the existing approaches is directly applicable to Web3 P2E games; a detailed review with citations appears in Section 7. In brief, mature game-bot detectors operate on proprietary server-side telemetry that Web3 game operators do not necessarily expose and community-side stakeholders never see, while blockchain-native anomaly detectors rely on feature sets engineered for DeFi-specific behaviors that do not arise in P2E game contexts. Moreover, existing studies evaluate within one game on one blockchain, and no public ground-truth dataset exists: the closest prior work on game-bot behavior in Web3 [6] is a model-free clustering analysis on a single title without supervised labels. These four gaps motivate a framework that is (a) purely on-chain, (b) game-agnostic in its feature design, (c) evaluated across multiple games and chains, and (d) accompanied by a labeled dataset.

3 Problem Motivation and Threat Model

This section motivates the proposed framework and defines its threat model.

3.1 Detection Feasibility from On-Chain Behavior

The central conjecture of this work is that bot accounts and human accounts in P2E games occupy distinguishable regions of a behavioral feature space defined purely over public on-chain data. We support this conjecture from two empirical angles, prior to any model training.

Raw-feature bimodality

Applying four simple, independent threshold-based indicators of known bot characteristics to raw on-chain behavior – (H1) transaction count > 100 within the collection window; (H2) action-type entropy < 0.5 bit; (H3) 24/7 activity ratio > 0.5 ; and (H4) coefficient of variation (CV) of daily volume < 0.2 – we find that 86.8% of accounts in our 57,879-account dataset satisfy none of the four rules and 1.4% satisfy at least three simultaneously, leaving only 11.8% in the ambiguous 1–2-rule zone. Even without learned representations, the bot and human populations thus exhibit strong bimodality on independently chosen behavioral axes.

Economic infeasibility of full camouflage

Even a sophisticated bot operator cannot simultaneously erase all four signals. Smoothing inter-action timing requires inserting idle delays that reduce reward throughput proportionally; diversifying actions forces engagement with unprofitable game mechanics; randomizing withdrawal cadence increases exposure to reward-token volatility; obfuscating graph structure requires routing through additional accounts, each of which incurs gas. Each evasion lever reduces the very economic advantage that motivated automation in the first place. This asymmetry implies that the behavioral subspace bots occupy is structurally constrained, not merely empirically common.

Together, these observations motivate classifying accounts from public on-chain histories alone.

3.2 Detection Challenges

A practical detector for P2E bots must overcome the following challenges:

C1: Cross-chain heterogeneity. The transaction representations, action vocabularies, and account models of EVM chains (DeFi Kingdoms on DFK Chain), EOSIO chains (Alien Worlds on WAX), and DPoS chains (Splinterlands on Hive) differ substantially. A detector must abstract over these differences without losing behavioral resolution.

C2: Cross-game semantic heterogeneity. Different titles employ different reward mechanics (idle mining vs. RPG questing vs. card battling) and different action vocabularies, yet share the same underlying bot objective. A detector must learn signatures that transfer across game mechanics.

C3: Severe class imbalance. Bot accounts represent only 6.1% of the labeled dataset, varying from 1.2% (Splinterlands) to 6.3% (Alien Worlds). Standard cross-entropy training fails to recover minority-class structure under this skew.

C4: Absence of public labeled datasets. No prior labeled dataset exists. Ground truth must be constructed from scratch using a labeling pipeline that is both scalable across tens of thousands of accounts and traceable enough to audit for label noise.

C5: Label noise from heuristic priors. Annotating ground truth at the scale of tens of thousands of accounts is aided by heuristic indicators that summarize known bot characteristics. If annotators lean on such indicators and the indicators overlap with the features the model later learns to use, the evaluation risks circularity: the model may merely recover the heuristic rule rather than learn a generalized bot signature.

3.3 Threat Model and Scope

Adversary capabilities

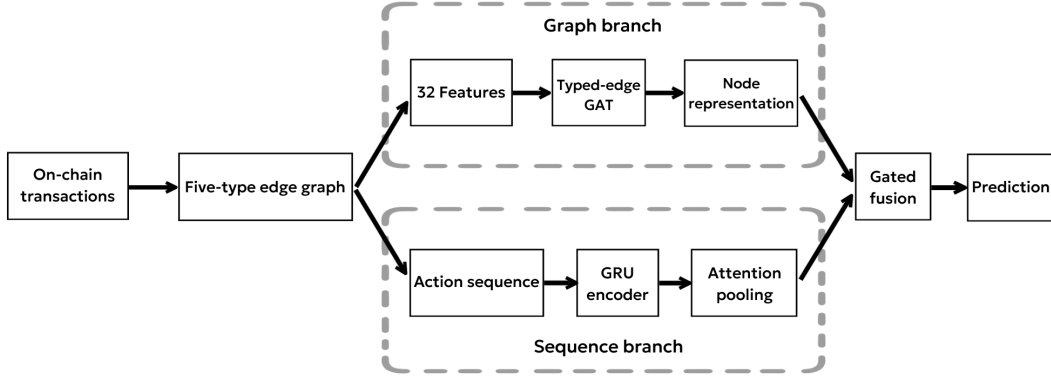
The adversary controls an arbitrary number of EOA-style accounts on one or more supported chains and submits transactions through the same public interfaces as legitimate players. The adversary may (i) automate execution of any in-game action permitted by the protocol, (ii) adjust transaction timing within bounds compatible with profitability, (iii) route funds through intermediate addresses, and (iv) vary action mix to partially mimic human play.

Adversary limitations

The adversary cannot forge on-chain timestamps, conceal completed transactions, or transact without paying associated gas or network fees. Any behavioral camouflage reduces the bot's effective reward-per-time. The adversary has no access to defender-side training pipelines and cannot tamper with the labeled dataset post hoc.

Defender goals

The defender is any stakeholder with read-only access to the chain (game developer, community watchdog, exchange compliance team). Given a target account a and its on-chain transaction history within an observation window, the defender outputs a binary label $\hat{y}_a \in \{0, 1\}$ indicating bot status. Detection latency on the order of hours is acceptable; sub-block-time inference is not required.



■ **Figure 1** Overview of the proposed framework. On-chain transactions from three P2E games are abstracted into a unified five-type edge graph, which feeds two parallel branches: a graph branch that encodes 32 node features with a typed-edge GAT, and a sequence branch that encodes the per-account action sequence with a GRU and attention pooling. A gated fusion combines the two embeddings, and a classifier predicts whether the account is a bot or a human.

Scope

We focus on per-account classification using a single observation window. The following are out of scope: (i) explicit detection of cross-account Sybil coordination (which requires pairwise or cluster-level analysis); (ii) adversarial robustness against bots that adapt their behavior in response to a public detector model; (iii) prevention or rollback of bot transactions at execution time. Sybil cluster detection and continual-learning extensions are discussed as future work in Section 6.

4 Proposed Framework

We first formalize the detection problem and then describe each component of the framework; Figure 1 gives an overview.

Problem Definition

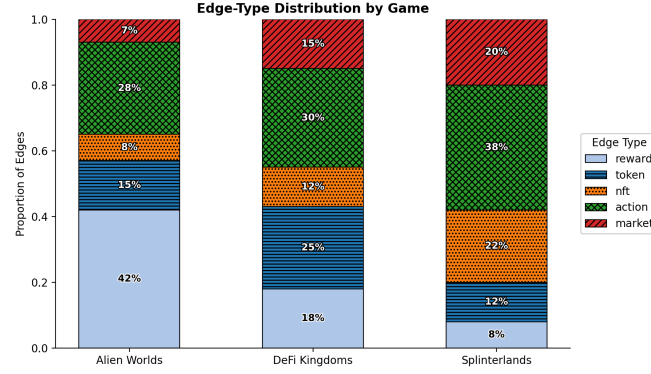
Let $\mathcal{A} = \{a_1, \dots, a_n\}$ be the set of player accounts observed during the collection period $[t_s, t_e]$ with at least $m = 10$ on-chain actions. Each account a_i is associated with a set of on-chain transactions. The goal is to learn a function $f: \mathcal{A} \rightarrow \{0, 1\}$ that classifies each account as human (0) or bot (1), using only publicly available on-chain data.

4.1 Heterogeneous Temporal Transaction Graph

We model on-chain activity as a heterogeneous temporal graph $G = (V, E, \tau, \varphi, w)$, where V is the node set (player accounts), $E \subseteq V \times V$ is the directed edge set (transactions), $\tau: E \rightarrow \mathbb{R}$ assigns a Unix timestamp, $\varphi: E \rightarrow \mathcal{T}$ assigns a type from a predefined vocabulary, and $w: E \rightarrow \mathbb{R}$ assigns a normalized transaction value.

To address challenges C1 and C2 (cross-chain and cross-game heterogeneity), we define a unified type vocabulary

$$\mathcal{T} = \{\text{reward}, \text{token}, \text{nft}, \text{action}, \text{market}\}$$



■ **Figure 2** Edge-type composition across the three games. Distributions differ substantially in dominance pattern, motivating the typed-edge GAT design.

■ **Table 1** Edge-type mapping per game.

Type	Alien Worlds	DeFi Kingdoms	Splinterlands
<i>reward</i>	logmine	Quest reward	sm_claim_rewards
<i>token</i>	TLM transfer	JEWEL transfer	sm_token_transfer
<i>nft</i>	atomicassets	Hero ERC-721	sm_sell_cards
<i>action</i>	setbag/setland	Hero quest TX	sm_find_match
<i>market</i>	AtomicMarket	DEX (UniswapV2)	sm_market_*

and map each game’s native action set onto it. Table 1 details this mapping. The five types were derived by analyzing the on-chain action vocabularies of all three games and identifying the minimal set of semantically distinct categories that (a) every game instantiates and (b) separates behaviors with different economic implications.

Each edge carries a scalar weight representing the transaction value, normalized per-game via log-transform followed by min-max scaling:

$$w'(e) = \frac{\log(w(e) + 1) - w_{\min}^g}{w_{\max}^g - w_{\min}^g} \quad (1)$$

where $w_{\min}^g$ and $w_{\max}^g$ are the per-game log-value extremes.

Figure 2 illustrates the empirical edge-type composition of each game’s transaction graph. The three games exhibit substantially different distributions: Alien Worlds is dominated by reward edges (42%) reflecting its high-frequency mining mechanics, DeFi Kingdoms is balanced across action (30%) and token (25%) edges reflecting its RPG-quest economy, and Splinterlands is dominated by action (38%) and nft (22%) edges reflecting its card-trading marketplace. This heterogeneity motivates the typed-edge GAT design in Section 4.3: a uniform-edge GNN would lose the type-conditional aggregation signal that differentiates, for example, a reward claim from a marketplace trade.

4.2 Feature Engineering

For each node $v \in V$, we extract a 32-dimensional feature vector $\mathbf{x}_v \in \mathbb{R}^{32}$ organized into four groups. Features are computable from the five universal transaction fields (sender, receiver, timestamp, value, action type) available on all three chains.

4.2.1 Temporal Regularity (F_T , 8 features)

The defining characteristic of bots is mechanical timing. Let $\Delta_1, \dots, \Delta_{n-1}$ be the inter-action intervals. We compute: (i) coefficient of variation $CV(\Delta) = \sigma(\Delta)/\mu(\Delta)$; (ii) hourly entropy $H = -\sum_{h=1}^{24} p_h \log_2 p_h$; (iii) 24/7 activity ratio; (iv) burst count (clusters separated by $>1h$ gaps); (v) lag-1 autocorrelation; (vi) longest continuous streak; (vii) weekend/weekday ratio; and (viii) session duration CV.

4.2.2 Graph Topology (F_G , 8 features)

Transaction graph structure encodes interaction scope: (i) in-degree; (ii) out-degree; (iii) degree ratio; (iv) local clustering coefficient; (v) unique counterparty count; (vi) reciprocity; (vii) PageRank centrality; and (viii) game-contract interaction ratio.

4.2.3 Financial Flow (F_F , 8 features)

Bots optimize for rapid value extraction: (i) reward-to-withdrawal latency; (ii) earned-to-withdrawn ratio; (iii) value Gini coefficient; (iv) median transaction value; (v) withdrawal target count; (vi) max-to-total value ratio; (vii) net flow; and (viii) primary reward token fraction.

4.2.4 Behavioral Diversity (F_D , 8 features)

Human players engage in varied activities; bots repetitively execute a narrow action set: (i) action type entropy; (ii) transition entropy over consecutive edge-type pairs; (iii) action-only ratio; (iv) NFT ID diversity; (v) daily volume CV; (vi) trigram uniqueness ratio; (vii) marketplace participation frequency; and (viii) active-day ratio.

Table 2 summarizes all 32 features together with their expected bot-side signature, i.e., the direction in which each feature is predicted to deviate for automated accounts relative to humans. The signatures are mutually constraining in the sense discussed in Section 3.1: a bot can soften any single feature only by accepting a proportional loss of throughput on another.

4.3 Model Architecture

Our framework employs a dual-branch encoder that separately models spatial and temporal dimensions of on-chain behavior, combined through gated fusion; the encoder is bootstrapped by a contrastive pre-training stage described later in this section. The dual-branch design is motivated by the observation that bot signatures manifest in two complementary modalities: (a) the local graph neighborhood (counterparty topology, financial flow shape) and (b) the chronological action sequence (timing regularity, action-type transitions). Either modality alone is insufficient to capture all three bot types described in Section 2.

4.3.1 Spatial Branch (Typed-Edge GAT)

The 32-dimensional feature vector serves as the initial node representation $\mathbf{h}_v^{(0)} = \mathbf{W}_0 \mathbf{x}_v$. Multi-edges between each node pair are aggregated into composite edge features by concatenating value statistics, edge count, type histogram, time span, and sinusoidal positional

■ **Table 2** The 32 chain-agnostic behavioral features grouped into four complementary categories. The third column states the qualitative direction in which the feature is expected to deviate for automated accounts.

Group	Feature	Bot-side signature
F_T Temporal	ft1_interval_cv	Low (mechanical timing)
	ft2_hourly_entropy	Low or very high
	ft3_247_ratio	High (round-the-clock activity)
	ft4_burst_count	Low (no rest periods)
	ft5_lag1_autocorr	High (periodic)
	ft6_max_streak	High (long continuous runs)
	ft7_weekend_ratio	≈ 1 (no weekday/weekend distinction)
	ft8_session_cv	Low (uniform sessions)
F_G Topology	fg1_in_degree	Low (few incoming sources)
	fg2_out_degree	Low (few outgoing targets)
	fg3_degree_ratio	Skewed (one-way flow)
	fg4_clustering	Low (isolated, star-shaped)
	fg5_counterparty	Low (few unique counterparties)
	fg6_reciprocity	Low (one-way relationships)
	fg7_pagerank	Low (peripheral)
	fg8_gc_ratio	High (concentrated on game contracts)
F_F Financial	ff1_r2w_latency	Low (immediate withdrawal)
	ff2_e2w_ratio	≈ 1 (drain everything earned)
	ff3_gini	Low (uniform-sized transactions)
	ff4_median_value	Low (repeated small amounts)
	ff5_withdrawal_targets	Low (1–2 collection addresses)
	ff6_max_value_ratio	Low (no large one-off transfers)
	ff7_net_flow	Strong net outflow
	ff8_reward_fraction	High (reward-only revenue)
F_D Diversity	fd1_action_entropy	Low (single action repeated)
	fd2_transition_entropy	Low (same transition repeated)
	fd3_action_only_ratio	Low (no genuine gameplay)
	fd4_nft_id_diversity	Low (same item reused)
	fd5_daily_volume_cv	Low (constant daily volume)
	fd6_trigram_unique	Low (same 3-action pattern)
	fd7_market_freq	Low or extreme
	fd8_active_day_ratio	High (active every day)

encoding. We apply a two-layer Graph Attention Network [18] with $K = 4$ attention heads, extended with edge-type-dependent parameters in the spirit of heterogeneous graph attention networks [20, 7]. The attention coefficient incorporates edge features:

$$\alpha_{ij}^{(l)} = \text{softmax}_j \left(\text{LeakyReLU} \left(\mathbf{a}^{(l)\top} \left[\mathbf{W}^{(l)} \mathbf{h}_i \parallel \mathbf{W}^{(l)} \mathbf{h}_j \parallel \mathbf{W}_e^{(l)} \mathbf{e}_{ij} \right] \right) \right) \quad (2)$$

In Eq. (2), the edge-feature term $\mathbf{W}_e^{(l)} \mathbf{e}_{ij}$ lets the attention weight depend on the edge type, so that a reward edge and a market edge between the same pair of accounts contribute differently. The two layers are stacked with residual connections and layer normalization, and the final-layer node embedding is taken as the spatial representation $\mathbf{h}_{\text{spatial}}$.

4.3.2 Temporal Branch (GRU-Attention)

For each account v , we extract the chronologically ordered action sequence $S_v = [(c_1, a_1, \Delta t_1, w_1), \dots]$ where c_i is the contract identifier, a_i the action type, Δt_i the log-transformed inter-action time, and w_i the transaction value. Each tuple is embedded and fed into a single-layer GRU. An attention mechanism pools over hidden states:

$$\begin{aligned}\alpha_t &= \text{softmax}(\mathbf{w}^\top \tanh(\mathbf{W}_a \mathbf{h}_t + \mathbf{b}_a)) \\ \mathbf{h}_{\text{temporal}} &= \sum_t \alpha_t \mathbf{h}_t\end{aligned}\tag{3}$$

The attention weights α_t in Eq. (3) let the model emphasize the time steps most indicative of automation, and the weighted sum yields the temporal representation $\mathbf{h}_{\text{temporal}}$.

4.3.3 Gated Fusion

We combine the two branch outputs through a learned element-wise gate that allows the model to dynamically weight spatial versus temporal evidence on a per-account basis:

$$\mathbf{g} = \sigma(\mathbf{W}_g [\mathbf{h}_{\text{spatial}} \parallel \mathbf{h}_{\text{temporal}}] + \mathbf{b}_g)\tag{4}$$

$$\mathbf{h}_{\text{fused}} = \mathbf{g} \odot \mathbf{h}_{\text{spatial}} + (1 - \mathbf{g}) \odot \mathbf{h}_{\text{temporal}}\tag{5}$$

The gate $\mathbf{g}$ in Eq. (4) is computed per account, so an account whose bot signature is mainly structural and one whose signature is mainly temporal can be weighted differently in the fused embedding of Eq. (5).

4.3.4 Contrastive Pre-training

To address C4 (label scarcity), we pre-train the encoder using NT-Xent loss [4] on unlabeled data, following the graph augmentation paradigm established by GraphCL [25]. Two augmented views of the graph are generated by independently applying edge dropout ($p_e = 0.2$) and feature masking ($p_m = 0.2$). For a batch of N accounts, the two views yield $2N$ embeddings $\{\mathbf{z}_k\}$; for each anchor i with positive counterpart i^+ (its other view), the loss is

$$\mathcal{L}_{\text{con}} = -\frac{1}{2N} \sum_{i=1}^{2N} \log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_{i^+})/\tau)}{\sum_{k \neq i} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}\tag{6}$$

where $\text{sim}(\cdot, \cdot)$ is cosine similarity, $\tau = 0.5$, and the denominator sums over all $2N - 1$ non-anchor embeddings.

4.3.5 Classification Head

The fused embedding is passed through a two-layer MLP with ReLU, dropout ($p = 0.3$), and batch normalization. To address C3 (class imbalance), we use Focal Loss [13] with $\gamma = 2$ and $\alpha_c = 0.75$ for the bot class:

$$\mathcal{L}_{\text{cls}} = -\alpha_c (1 - p_t)^\gamma \log(p_t)\tag{7}$$

In Eq. (7), p_t denotes the predicted probability of the account's ground-truth class. The two factors counteract the 6.1% class imbalance in complementary ways: the class weight $\alpha_c = 0.75$ scales up the loss of the rare bot class relative to the human majority, while the modulating factor $(1 - p_t)^\gamma$ decays the contribution of well-classified examples – with $\gamma = 2$,

an example classified with $p_t = 0.9$ contributes $100\times$ less than under standard cross-entropy – so the abundant, mostly easy human accounts do not dominate the gradient and training concentrates on the hard examples near the decision boundary. The total supervised loss combines the focal classification loss of Eq. (7) with the contrastive term of Eq. (6) as a regularizer, $\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda\mathcal{L}_{\text{con}}$ with $\lambda = 0.01$.

4.3.6 Training Procedure

Training proceeds in two phases. Phase 1 pre-trains the entire dual-branch encoder with the contrastive objective alone, using two independently augmented views of the graph so that the encoder learns bot-discriminative structure before any label is consumed. Phase 2 fine-tunes the whole model with the supervised focal loss while retaining the contrastive term as a regularizer; at each epoch the decision threshold θ^* is chosen to maximize validation F1, and the model with the best validation F1 is retained for test-time inference.

4.4 Ground Truth Creation

To address C4 (no public labeled dataset), we adopt a two-stage labeling process designed for auditability. The procedure follows the ground-truth creation methodology of Niedermayer et al. [14], adapting it from individual Ethereum addresses to P2E game accounts.

Stage 1: Heuristic Reference Scoring

We reuse the four threshold-based indicators H1–H4 introduced in Section 3.1 (transaction volume, action-type entropy, 24/7 activity, and daily-volume variability). Each account receives a heuristic score equal to the number of rules it satisfies. This score is provided to annotators purely as a reference signal; it does not select, filter, or pre-label accounts, and every account in the dataset proceeds to annotation in Stage 2 regardless of its score.

Stage 2: Annotator Validation

Two independent annotators label each account by examining its raw on-chain activity (its transaction history and behavioral feature profile) together with the Stage 1 heuristic score as a supporting reference, and assign a binary label, with disagreements resolved through discussion until consensus. Because the decision is grounded in the raw activity, annotators may reach a conclusion that diverges from what the heuristic score alone would suggest. The resulting dataset comprises 3,551 bot and 54,328 human accounts across the three games; its per-game composition is reported later in Table 4.

Addressing Potential Circularity (C5)

Because the heuristic score is only a reference signal and never assigns or filters labels, the ground truth is determined entirely by annotator judgment over the full feature profile of every account. The reported performance therefore reflects agreement with human judgment rather than recovery of the heuristic rules.

4.5 Data Collection

We collect on-chain transaction data from three operational P2E games over a 31-day period (March 1–April 1, 2026) using publicly available blockchain APIs. The three games were selected to maximize diversity along three dimensions: blockchain architecture (EVM, EOSIO, DPoS), gameplay mechanics (idle mining, RPG questing, card battling), and economic scale. Table 3 summarizes data sources.

32:12 Who Plays to Earn?

■ **Table 3** Data sources and collection methods.

Game	Blockchain	API
Alien Worlds	WAX	Hyperion History
DeFi Kingdoms	DFK Chain	Routescan + RPC
Splinterlands	Hive	Condenser API

■ **Table 4** Dataset and transaction graph statistics per game.

Game	Accounts (Nodes)	Bots	Bot %	Edges	Unique Pairs	Edges/Nodes
Alien Worlds	54,298	3,440	6.3%	2,421,307	132,854	44.6
DeFi Kingdoms	3,257	107	3.3%	318,426	21,493	97.8
Splinterlands	324	4	1.2%	34,275	2,144	105.8
Total	57,879	3,551	6.1%	2,774,008	156,491	47.9

Ethical Considerations

All data consists of publicly broadcast blockchain transactions with no personally identifiable information – addresses are pseudonymous identifiers not linked to real-world identities – and the research involves no human participants and no private or off-chain data, so it does not fall under institutional review board (IRB) oversight. Given the dual-use nature of bot detection, where a false positive could unfairly penalize a legitimate player, we frame the framework as a decision-support tool for human review rather than a basis for automated enforcement.

5 Evaluation

We design experiments to answer the following research questions:

- **RQ1 (Detection Performance):** How effective is the proposed framework for bot detection on the joint dataset?
- **RQ2 (Feature Contribution):** Which feature groups contribute most to detection performance, and how does the model behave during training?
- **RQ3 (Model Analysis):** What do the learned representations reveal about the decision process?

5.1 Experimental Setup

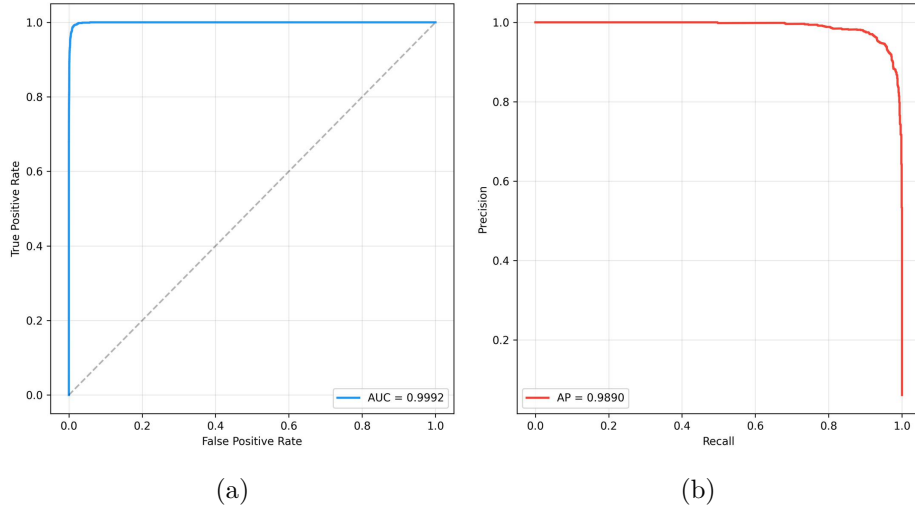
5.1.1 Dataset and Graph Statistics

Table 4 summarizes the labeled dataset and the per-game transaction graphs. The data is split via stratified sampling (70% train, 30% test) per game, ensuring bot representation in both splits. Validation is drawn as an internal 85/15 split from the training set for early stopping.

The graphs vary substantially in scale and density: Alien Worlds produces by far the largest transaction graph (over 2.4M edges) due to its high-frequency mining mechanics, whereas the smaller games are denser per node, with Splinterlands reaching the highest edges-per-node ratio as its card-trading marketplace concentrates activity among fewer accounts.

■ **Table 5** Detection performance on the joint test set.

Metric	Value
Accuracy	99.30%
Precision	92.52%
Recall	96.34%
F1-Score	94.39%
AUC-ROC	99.92%
AP	98.90%
Optimal Threshold	0.480



■ **Figure 3** (a) ROC curve (AUC = 0.9992). (b) Precision-recall curve (AP = 0.9890). Both indicate near-perfect ranking ability despite the strong class imbalance.

5.1.2 Implementation Details

The dual-branch encoder uses $d = 128$, $K = 4$ attention heads, 2 GAT layers, and GRU hidden dimension 128. We use Adam optimizer ($\text{lr} = 10^{-3}$, weight decay 10^{-5}) with cosine annealing ($\eta_{\min} = 10^{-5}$). Contrastive pre-training runs for 80 epochs; fine-tuning for up to 500 epochs with early stopping (patience 80) on validation F1. The decision threshold is optimized on the validation set via grid search over $[0.05, 0.95]$ with step 0.01.

5.2 RQ1: Detection Performance

Table 5 reports the detection performance of the proposed framework on the joint test set.

The model achieves an F1-score of 94.39% and AUC of 99.92% on the joint test set, demonstrating strong discriminative ability despite the 6.1% class imbalance.

Score Separation

Figure 4(a) shows the predicted probability distributions for human and bot accounts. The distributions exhibit a sharp bimodal separation: human accounts concentrate near probability 0.13 and bot accounts near 0.99, with only about 5% of accounts falling in the

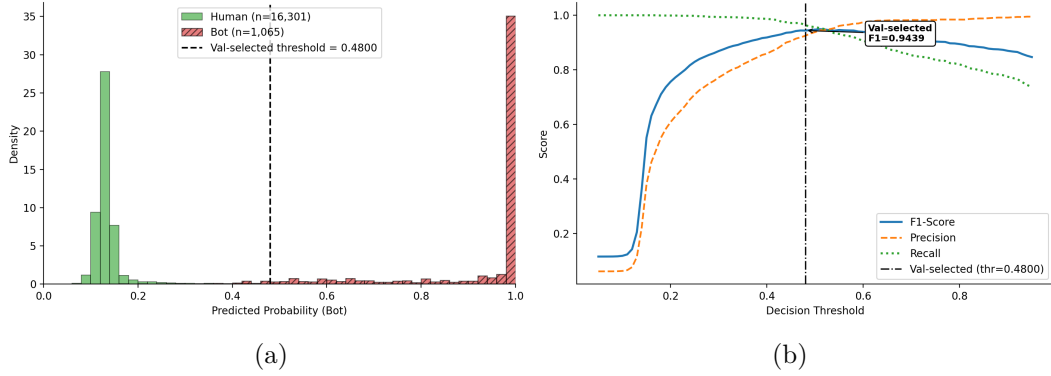


Figure 4 (a) Predicted probability distributions for human (green) and bot (red, hatched) accounts; the dashed line marks the validation-selected decision threshold (0.48). (b) Threshold sensitivity for F1 (blue, solid), precision (orange, dashed), and recall (green, dotted); F1 remains above 0.90 across $[0.36, 0.77]$, and the validation-selected threshold of 0.48 yields $F1 = 0.9439$.

intermediate range $[0.2, 0.9]$. This clean separation indicates that the model makes highly confident predictions for the vast majority of accounts, rather than relying on a narrow margin around the threshold. The bimodality observed at the raw-feature level in Section 3.1 thus survives, and is in fact sharpened by, the learned representation.

Threshold Robustness

Figure 4(b) shows the relationship between decision threshold and detection metrics. The F1-score remains above 0.90 across the wide range $[0.36, 0.77]$, indicating that the model’s performance is not brittle with respect to threshold selection. The threshold of 0.48, selected on the validation split, achieves $F1 = 0.9439$ with precision 92.52% and recall 96.34%.

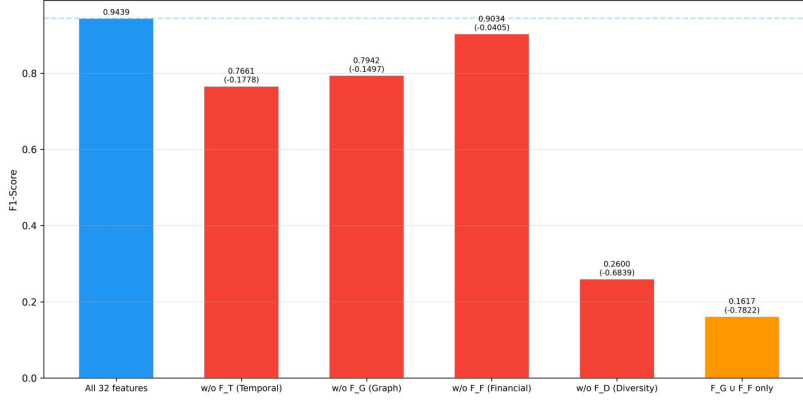
Ranking Quality

The ROC and precision-recall curves (Figure 3) achieve $AUC = 0.9992$ and $AP = 0.9890$ respectively. The latter is particularly informative given the class imbalance: the model maintains precision above 0.95 up to a recall of roughly 0.9, only dropping more steeply as recall approaches 1.0.

5.3 RQ2: Feature Contribution

Feature Group Ablation

We quantify the contribution of each of the four feature groups ($\mathbf{F}_T$, $\mathbf{F}_G$, $\mathbf{F}_F$, $\mathbf{F}_D$) by masking the corresponding columns at inference time (setting them to zero) and re-evaluating the trained model on the same test set. Figure 5 reports the resulting F1 scores. The full model reaches $F1 = 0.9439$ on the ablation run; removing any one group degrades performance, but the degradation is highly uneven. The Diversity group $\mathbf{F}_D$ is by far the most critical: masking it drops F1 to 0.2600 (-0.6839), confirming that behavioral diversity, particularly action-type entropy, transition entropy, and trigram uniqueness, carries the dominant signal distinguishing bots from humans. The Temporal (-0.1778) and Topology (-0.1497) groups contribute moderately, while the Financial group is the least sensitive (-0.0405), consistent with financial outflow patterns being easier for bots to randomize than action-mix patterns.



■ **Figure 5** Feature-group ablation by masked inference on the joint test set. The full model attains $F1 = 0.9439$; masking the Diversity group F_D causes the largest drop. The rightmost bar reports a model trained on $F_G \cup F_F$ alone.

As a further reference point, a model trained on the topology and financial groups alone ($F_G \cup F_F$) attains only $F1 = 0.1617$ (rightmost bar in Figure 5), reinforcing that the temporal and diversity groups carry most of the discriminative signal.

Training Convergence

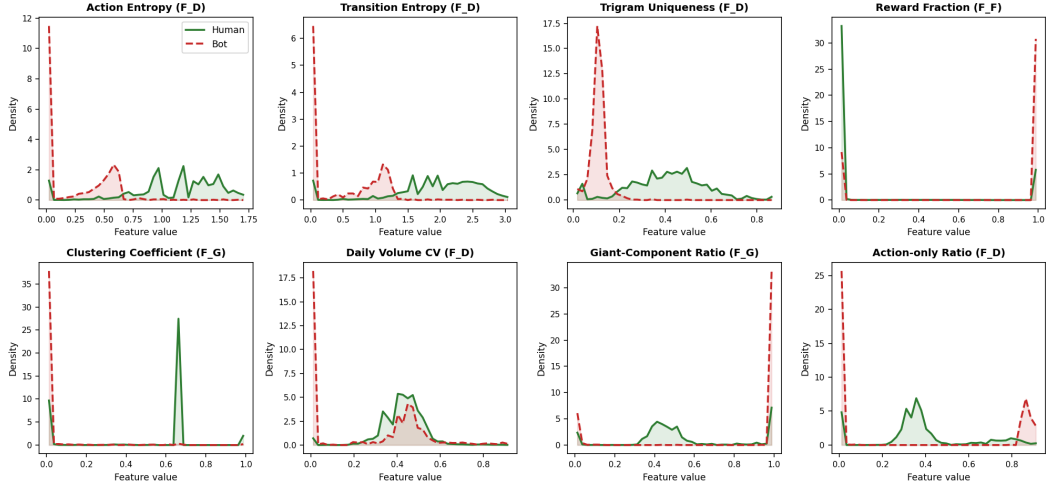
Figure 8(a) shows the fine-tuning loss and validation metrics over 246 epochs (terminated by early stopping). Validation AUC crosses 0.99 within the first 9 epochs and saturates above 0.999, indicating that the model learns to rank accounts correctly very early, while the threshold-dependent F1 metric requires additional epochs to stabilize, peaking at 0.966 with progressively decreasing variance.

Human versus Bot Feature Separation

The ablation above quantifies *which* feature groups matter; Figure 6 shows *why* they matter, by contrasting the class-conditional distributions of the eight most discriminative features. For every feature, the bot distribution is sharply concentrated while the human distribution is broad. The three behavioral-diversity features (action entropy, transition entropy, and trigram uniqueness) exhibit the clearest separation: bot accounts collapse onto a narrow low-entropy spike, whereas human accounts spread across the full range, reflecting the repetitive action loops that automation cannot avoid without sacrificing throughput. The pattern recurs in the remaining groups. Bots cluster near a reward fraction of 1.0 (almost every transaction is a reward claim), near a giant-component ratio of 1.0 and a clustering coefficient of 0 (star-shaped ego-networks rather than genuine peer interaction), and at near-zero daily-volume variation (mechanically constant activity). This direct distributional evidence corroborates the ablation ranking: the features whose class distributions separate most are exactly those whose removal most degrades F1, and five of the eight belong to the dominant F_D group.

Cross-Game Feature Distributions

Figure 7 visualizes the distributions of the top four features across the three games. The distributions differ substantially across games (e.g., DeFi Kingdoms shows a narrow spike in `fd1_action_entropy` due to its limited action vocabulary), which motivates the per-game



■ **Figure 6** Class-conditional distributions of the eight most discriminative features, ranked by Wasserstein distance between the human and bot distributions. For each feature the x -axis is the (raw) feature value and the y -axis is probability density; the green solid curve is human accounts and the red dashed curve is bot accounts. Bot distributions are consistently concentrated while human distributions are dispersed.

normalization adopted in Section 4.2. Despite these distributional differences, the joint model achieves strong performance, suggesting that the learned representations capture game-invariant bot signatures.

5.4 RQ3: Model Analysis

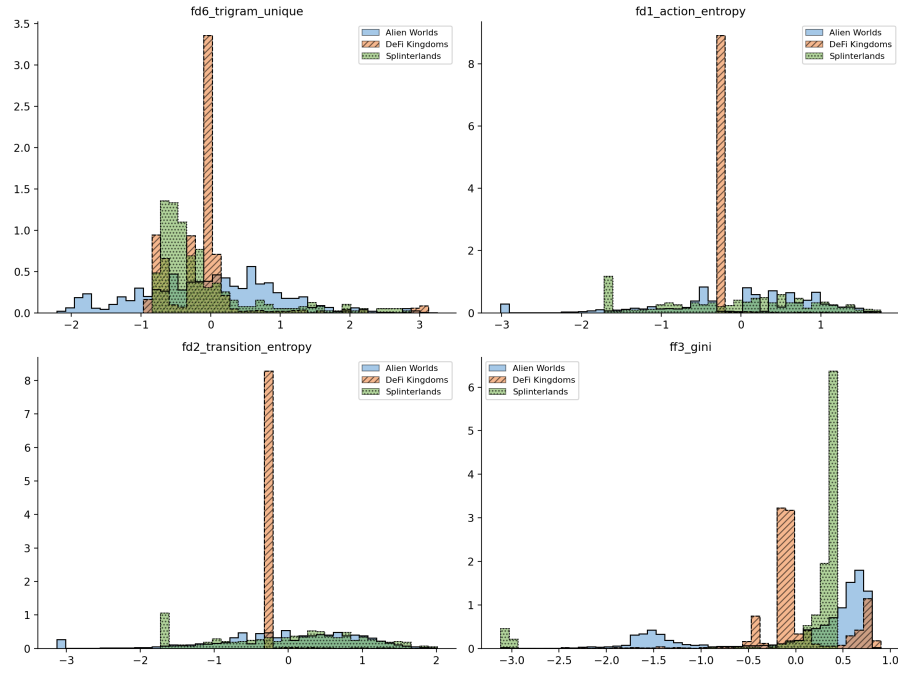
Learned Representations

Figure 8(b) visualizes the fused embeddings ($\mathbf{h}_{\text{fused}} \in \mathbb{R}^{128}$) of 3,000 randomly sampled test accounts via t-SNE. Bot accounts (red triangles) form several tight, well-separated clusters, while human accounts (green circles) form a more diffuse distribution. The presence of multiple distinct bot clusters suggests that the model learns to distinguish different bot operational profiles (e.g., reward-farming bots vs. market-manipulation bots), even though the training labels are binary, pointing to potential for unsupervised bot sub-type discovery in future work. The clear spatial separation between bot and human clusters corroborates the bimodal score distribution observed in Figure 4.

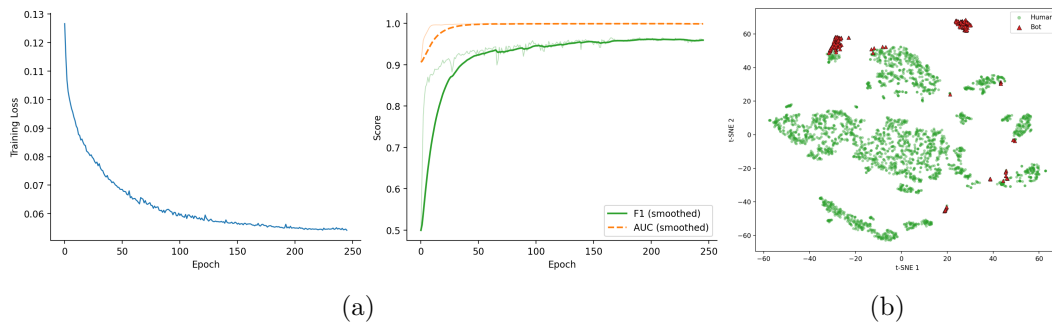
6 Discussion

Practical Implications

Because the framework relies only on public on-chain data, it can be deployed by any stakeholder (game developers, community watchdogs, or platform operators) without privileged access to game telemetry, and the unified five-type edge vocabulary allows porting to a new P2E title by specifying only a single mapping table.



■ **Figure 7** Distributions of the top four features across the three games. The substantial cross-game distributional shifts motivate per-game normalization.



■ **Figure 8** (a) Training curves, showing fine-tuning loss and validation F1 (green, solid) and AUC (orange, dashed), smoothed with $\beta = 0.9$. (b) t-SNE of fused embeddings, where bot accounts (red triangles) form tight, well-separated clusters and human accounts (green circles) are more dispersed.

Post-Detection Mitigation

Detection is only the first step of an operational response, and the appropriate follow-up depends on who runs the detector and on the cost of a false positive. We envision a graduated, score-driven pipeline rather than a single hard ban decision. Accounts in the high-confidence bot mode of the score distribution (Figure 4) can face automated friction that is cheap for legitimate players but costly for automation, such as rate limits on reward-claim actions, short escrow delays before reward withdrawal, or step-up verification challenges; because such measures throttle rather than ban, they directly erode the reward-per-time advantage that motivates automation (Section 3.1) while remaining reversible if a flag proves incorrect. The roughly 5% of accounts in the intermediate score range should instead be routed to human review before any enforcement, consistent with the decision-support framing of our ethical discussion (Section 4.5); for irreversible actions such as account suspension or confiscation of farmed rewards, operators can shift the decision threshold toward higher precision at a quantified recall cost (Figure 4). Stakeholders without in-game enforcement power, such as community watchdogs and exchange compliance teams, can use flagged-address lists for market-level responses, for example enhanced screening of withdrawal targets. Finally, because bot operators may adapt once a mitigation policy becomes visible, deployment should pair the detector with periodic re-training over rolling observation windows, as discussed under future work.

Limitations

We acknowledge several limitations. First, the framework performs per-node classification and cannot directly detect coordinated Sybil clusters that require cross-account analysis. Second, the unified five-type edge vocabulary necessarily loses game-specific semantic granularity. Third, annotators may still be influenced by the heuristic reference score (Section 4.4), so the ground truth could under-represent sophisticated bots whose raw behavior closely resembles human play. Fourth, our evaluation covers three games, but the Web3 gaming landscape includes hundreds of titles with diverse mechanics; generalization to genres not represented in our dataset remains to be established. Fifth, Splinterlands contributes only 4 bot accounts, which limits the statistical reliability of per-game analysis for this title.

Future Work

We see six promising directions for extending this work. The most immediate is a systematic comparison against alternative classifiers – classical machine-learning baselines on the 32-dimensional feature vector, single-branch ablations, and alternative graph architectures in place of the typed-edge encoder – to more precisely isolate the contribution of the dual-branch design. Pairing this comparison with a time-based train/test split, rather than the random split used here, would further rule out operator-level leakage. A second direction concerns the feature-group ablation of Section 5.3, where we adopted inference-time masking for rapid experimental turnaround; we plan to complement it with ablations that retrain the model from scratch without each feature group. A third is an observation-window sensitivity study: detection currently consumes a 31-day observation window (Section 4.5), and repeating it with one week, one day, and even one hour of history would test whether performance is maintained with far less data. A fourth direction is the detection of Sybil clusters through cross-account synchronization analysis, which would address the per-account scope limitation noted in Section 3.3. A fifth is adaptation to dynamic settings in which bot behavior evolves over time, which we expect to approach through continual learning over rolling observation

windows. Finally, broadening the evaluation to additional Web3 game genres, including fully on-chain games, would test how well the framework generalizes beyond the three titles studied here.

7 Related Work

7.1 Bot Detection in Traditional Online Games

Server-side bot detection in online games has evolved through three methodological generations. Early statistical methods relied on handcrafted behavioral indicators: Yeung et al. [24] modeled aiming-accuracy anomalies with Bayesian inference, Chen et al. [3] identified bots from avatar movement trajectories, and Kang et al. [9] and Thawonmas et al. [17] classified bots from party-play and behavioral-frequency logs. Although effective within their respective games, these approaches require game-specific feature engineering and do not transfer across titles.

Deep sequential models reduced the need for manual feature design. Xu et al. [23] proposed NGUARD+, which captures player behavior patterns through attention-based BiLSTM and hierarchical self-attention networks. Such models achieve strong in-game performance but depend on detailed server-side event logs such as item pickups, skill usage, and movement coordinates, none of which are available in Web3 environments.

More recently, graph, contrastive, and pre-training approaches have addressed structural and label-scarce settings. Lee et al. [10] proposed a self-similarity framework at NDSS showing that behavioral repetitiveness is a game-agnostic bot signature; our $\mathbf{F}_T$ and $\mathbf{F}_D$ groups operationalize this insight through information-theoretic measures. Zhao et al. [27] introduced T-Detector, a trajectory-based pre-trained model achieving cross-game generalization; our contrastive pre-training follows the same paradigm but substitutes spatial trajectories with on-chain transaction sequences. Li et al. [12] applied contrastive learning on finger-operation graphs, and Wang et al. [21] and Qi et al. [16] addressed label scarcity and social signals respectively. A common limitation spans all three generations: each depends on proprietary server-side data unavailable in permissionless settings, and none model the on-chain financial behaviors that constitute the primary objective of P2E game bots.

7.2 Anomaly Detection on Blockchain Graphs

GNN-based methods for blockchain anomaly detection have progressed along two axes. On the temporal modeling axis, Weber et al. [22] established the Elliptic benchmark for anti-money laundering on Bitcoin, and EvolveGCN [15] introduced RNN-coupled weight evolution for dynamic graphs. On the label efficiency axis, GraphCL [25] proposed a graph contrastive learning framework whose augmentation design (edge perturbation, attribute masking) our pre-training directly adapts, and CARE-GNN [5] targeted camouflaged fraudsters; while we do not explicitly model adversarial camouflage, our contrastive objective implicitly encourages representations robust to local perturbations.

Our architecture draws on both lines but applies them to a different problem. TTAGN [11] models temporal transaction relations for Ethereum phishing detection; our temporal branch shares this intuition but extends it to heterogeneous multi-game graphs with typed edges. More broadly, blockchain fraud detection models complex fund-routing topologies (layering, mixing, fan-out/fan-in), whereas game-bot signatures are simple, repetitive action sequences with regular timing and narrow interaction scope.

7.3 Bot Detection on Blockchains

Niedermayer et al. [14] represent the closest methodological precedent to our work. They systematize bot types on Ethereum into seven categories and extract 83 address-level features spanning transaction volume, gas usage, temporal patterns, and smart contract interaction profiles. A Random Forest classifier achieves the strongest supervised performance on a balanced subset of 270 labeled addresses. BotDetect [2] extends this approach into a federated learning framework for EVM chains but inherits the same DeFi-centric feature set.

Two limitations restrict the applicability of these methods to Web3 games. First, the feature set is DeFi-oriented: beyond generic transaction statistics, its contract-level features model only Uniswap-style DEX swaps and ERC-20 transfers, capturing trading rather than gameplay automation. Second, both studies are confined to Ethereum mainnet, whereas P2E games span diverse chain architectures (EVM, EOSIO, Hive DPoS).

Guidi et al. [6] offer a complementary, game-focused perspective through a model-free analysis of Gods Unchained, identifying accounts with abnormally intense activity and locating them within communities of the player battle graph. While the study provides valuable empirical observations, it does not advance to predictive modeling.

Positioned against this literature, our work is, to our knowledge, the first to combine on-chain behavioral analysis with deep graph-temporal modeling in a cross-game, cross-chain framework, extending prior on-chain bot detection [14] and the observations of Guidi et al. [6] from descriptive analysis to predictive, game-specific classification.

8 Conclusion

We presented a cross-game, cross-chain framework that detects Web3 Play-to-Earn game bots from public on-chain transactions alone, using a unified five-type edge vocabulary, 32 chain-agnostic behavioral features, and a dual-branch GAT-GRU encoder. Across three games and three blockchains (57,879 accounts), it attains $F1 = 94.39\%$, $AUC = 99.92\%$, and $AP = 98.90\%$, showing that the economic asymmetry of P2E automation leaves on-chain traces sufficient for detection without per-game retraining.

References

- 1 Jordan Ali and Gili Vidan. Playing, earning, crashing, and grinding: Axie Infinity and growth crises in the Web3 economy. *Big Data & Society*, 12, 2025. doi:10.1177/20539517251357296.
- 2 Ahmed Mounsf Rafik Bendada, Abdelaziz Amara Korba, Mouhamed Amine Bouchiha, and Yacine Ghamri-Doudane. BotDetect: A decentralized federated learning framework for detecting financial bots on the EVM blockchains. In *Proc. IEEE Int. Conf. Communications (ICC)*, pages 1103–1108, 2025. doi:10.1109/ICC52391.2025.11161090.
- 3 Kuan-Ta Chen, Hsing-Kuo Kenneth Pao, and Hong-Chung Chang. Game bot identification based on manifold learning. In *Proc. ACM SIGCOMM Workshop on Network and System Support for Games (NetGames)*, pages 21–26, 2008. doi:10.1145/1517494.1517498.
- 4 Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proc. Int. Conf. Machine Learning (ICML)*, pages 1597–1607, 2020. URL: <http://proceedings.mlr.press/v119/chen20j.html>.
- 5 Yingtong Dou, Zhiwei Liu, Li Sun, Yutong Deng, Hao Peng, and Philip S. Yu. Enhancing graph neural network-based fraud detectors against camouflaged fraudsters. In *Proc. ACM Int. Conf. Information and Knowledge Management (CIKM)*, pages 315–324, 2020. doi:10.1145/3340531.3411903.

- 6 Barbara Guidi, Andrea Michienzi, and Laura Ricci. Detecting suspicious player behavior in Web3 games: A data-driven analysis of bot accounts. In *Proc. Int. Conf. Information Technology for Social Good (GoodIT)*, pages 364–372, 2024. doi:10.1145/3677525.3678683.
- 7 Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. Heterogeneous graph transformer. In *Proc. ACM Web Conf. (WWW)*, pages 2704–2710, 2020. doi:10.1145/3366423.3380027.
- 8 Woncheol Jeong, Sohee Choi, and Jiwon Yoon. Bots-in-Web3-P2E-Games/Dataset-for-Detecting-Bots-in-Web3-P2E-Games. Dataset, swhId: `swh:1:dir:2eef651a0c2d25b011e0754a2f62e4784d985072` (visited on 2026-09-21). URL: <https://github.com/WONCHEOL-JEONG/Dataset-for-Detecting-Bots-in-Web3-P2E-Games.git>, doi:10.4230/artifacts.28075.
- 9 Ah Reum Kang, Jiyoung Woo, Juyong Park, and Huy Kang Kim. Online game bot detection based on party-play log analysis. *Computers & Mathematics with Applications*, 65(9):1384–1395, 2013. doi:10.1016/j.camwa.2012.01.034.
- 10 Eunjo Lee, Jiyoung Woo, Hyounghshick Kim, Aziz Mohaisen, and Huy Kang Kim. You are a game bot!: Uncovering game bots in MMORPGs via self-similarity in the wild. In *Proc. ISOC Network and Distributed System Security Symp. (NDSS)*, pages 1–15, 2016. doi:10.14722/ndss.2016.23436.
- 11 Sijia Li, Gaopeng Gou, Chang Liu, Chengshang Hou, Zhenzhen Li, and Gang Xiong. TTAGN: Temporal transaction aggregation graph network for Ethereum phishing scams detection. In *Proc. ACM Web Conf. (WWW)*, pages 661–669, 2022. doi:10.1145/3485447.3512226.
- 12 Wenbin Li, Xiaokai Chu, Yueyang Su, Di Yao, Shiwei Zhao, Runze Wu, Shize Zhang, Jianrong Tao, Hao Deng, and Jingping Bi. FingFormer: Contrastive graph-based finger operation transformer for unsupervised mobile game bot detection. In *Proc. ACM Web Conf. (WWW)*, pages 3367–3375, 2022. doi:10.1145/3485447.3512272.
- 13 Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proc. IEEE Int. Conf. Computer Vision (ICCV)*, pages 2980–2988, 2017. doi:10.1109/ICCV.2017.324.
- 14 Thomas Niedermayer, Pietro Saggese, and Bernhard Haslhofer. Detecting financial bots on the Ethereum blockchain. In *Companion Proc. ACM Web Conf. (WWW)*, pages 1742–1751, 2024. doi:10.1145/3589335.3651959.
- 15 Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. EvolveGCN: Evolving graph convolutional networks for dynamic graphs. In *Proc. AAAI Conf. Artificial Intelligence*, pages 5363–5370, 2020. doi:10.1609/aaai.v34i04.5984.
- 16 Xianyang Qi, Jiashu Pu, Shiwei Zhao, Runze Wu, and Jianrong Tao. A GNN-enhanced game bot detection model for MMORPGs. In *Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD)*, pages 316–327, 2022. doi:10.1007/978-3-031-05936-0_25.
- 17 Ruck Thawonmas, Yoshitaka Kashifuji, and Kuan-Ta Chen. Detection of MMORPG bots based on behavior analysis. In *Proc. Int. Conf. Advances in Computer Entertainment Technology (ACE)*, pages 91–94, 2008. doi:10.1145/1501750.1501770.
- 18 Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proc. Int. Conf. Learning Representations (ICLR)*, 2018.
- 19 Friedhelm Victor and Andrea Marie Weintraud. Detecting and quantifying wash trading on decentralized cryptocurrency exchanges. In *Proc. ACM Web Conf. (WWW)*, pages 23–32, 2021. doi:10.1145/3442381.3449824.
- 20 Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. Heterogeneous graph attention network. In *Proc. ACM Web Conf. (WWW)*, pages 2022–2032, 2019. doi:10.1145/3308558.3313562.
- 21 Yilin Wang, Sha Zhao, Shiwei Zhao, Runze Wu, Yuhong Xu, Jianrong Tao, Tangjie Lv, Shijian Li, Zhipeng Hu, and Gang Pan. PU-Detector: A PU learning-based framework for

- real money trading detection in MMORPG. *ACM Trans. Knowl. Discov. Data*, 18(4):1–26, 2024. doi:10.1145/3638561.
- 22 Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I. Weidele, Claudio Bellei, Tom Robinson, and Charles E. Leiserson. Anti-money laundering in Bitcoin: Experimenting with graph convolutional networks for financial forensics. In *Proc. KDD Workshop on Anomaly Detection in Finance*, 2019. doi:10.48550/arXiv.1908.02591.
 - 23 Jiarong Xu, Yifan Luo, Jianrong Tao, Changjie Fan, Zhou Zhao, and Jiangang Lu. NGUARD+: An attention-based game bot detection framework via player behavior sequences. *ACM Trans. Knowl. Discov. Data*, 14(6):1–24, 2020. doi:10.1145/3399711.
 - 24 Siu Fung Yeung, John C. S. Lui, Jiangchuan Liu, and Jeff Yan. Detecting cheaters for multi-player games: Theory, design and implementation. In *Proc. IEEE Consumer Communications and Networking Conf. (CCNC)*, pages 1178–1182, 2006. doi:10.1109/CCNC.2006.1593224.
 - 25 Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 5812–5823, 2020.
 - 26 Jingfan Yu, Mengqian Zhang, Xi Chen, and Zhixuan Fang. SoK: Play-to-earn projects. In *Proc. Int. Conf. Financial Cryptography and Data Security (FC)*, pages 389–407, 2023. doi:10.1007/978-3-031-47751-5_22.
 - 27 Sha Zhao, Junwei Fang, Shiwei Zhao, Runze Wu, Jianrong Tao, Shijian Li, and Gang Pan. T-Detector: A trajectory based pre-trained model for game bot detection in MMORPGs. In *Proc. IEEE Int. Conf. Data Engineering (ICDE)*, pages 992–1003, 2022. doi:10.1109/ICDE53745.2022.00079.