From Rules to Nash Equilibria: Formally Verified Game-Theoretic Analysis of a Competitive Trading Card Game

Author names withheld for review

***Abstract*—We present a metagame analysis of the competitive Pokémon Trading Card Game, formally verified using Lean 4 and real tournament data. All game-theoretic results (Nash equilibrium, replicator dynamics, bridge alignment) rely on `native_decide`, which trusts Lean's compiler rather than its kernel; the trust boundary is detailed in Section IX. The artifact spans approximately 30,000 lines, 81 files, and 2,500 theorems—of which roughly 190 directly verify empirical claims—with no `sorry`, `admit`, or custom axioms. Analyzing Trainer Hill data (January–February 2026, 50+ player events) over 14 archetypes and their full pairwise matchup matrix, we prove a *popularity paradox*: the most played deck (Dragapult, 15.5% share) has only 46.7% expected win rate, while Grimmsnarl (5.1% share) achieves 52.7%. A machine-checked Nash equilibrium with six-deck support assigns Dragapult 0% weight; exhaustive support enumeration over all $2^{14} - 1$ subsets confirms this is the unique symmetric Nash equilibrium of the constant-sum symmetrization, and Dragapult is strictly suboptimal against the equilibrium mix by 63.3 permil. Single-step replicator dynamics on the *full 14-deck game* indicate downward fitness pressure on Dragapult, upward pressure on Grimmsnarl, and strongest extinction pressure on Alakazam. A 10,000-iteration sensitivity analysis confirms qualitative stability: core support decks appear in >96% of resampled equilibria. The primary contribution is methodological: formal verification transforms qualitative metagame narratives into machine-checkable, reproducible strategic science.**

***Index Terms*—Formal verification, game theory, trading card games, Nash equilibrium, theorem proving, metagame analysis, replicator dynamics, Lean 4**

I. Introduction

Tournament outcomes in competitive trading card games (TCGs) are often shaped before round one begins. The pre-tournament deck-selection problem is naturally modeled as a strategic game where payoffs derive from matchup win rates and the population distribution of opponents. The Pokémon TCG is especially suitable for this analysis: it has a large organized-play ecosystem, clearly defined public rules, and a metagame that evolves quickly enough to produce measurable strategic cycles, yet hidden information and stochastic effects make intuition unreliable even for experienced players.

By encoding game semantics in Lean 4 [1] and proving strategic statements directly over exact data representations, we build a proof-carrying metagame analytics pipeline where the verified objects are (i) data representation and ingestion, (ii) expected-value computations over the field, (iii) machine-checked Nash-equilibrium computation and full 14-deck replicator dynamics, and (iv) tournament-objective transforms (Bo3, Swiss).

Our empirical foundation is Trainer Hill metagame data [2], [3] for 50+ player tournaments from January 29 to February 19, 2026. We model the top 14 archetypes and their full pairwise matchup matrix. The paper makes four contributions: (1) a machine-checked bridge from type-effectiveness rules to empirical matchup outcomes (Section III-A); (2) verified expected-value computations revealing a popularity paradox; (3) machine-checked Nash equilibrium and replicator dynamics over the full 14-deck game; and (4) verified best-of-three and Swiss-relevant tournament transforms.

While the headline popularity paradox could be computed in a spreadsheet, the formal verification methodology provides three distinct advantages. First, **compositional guarantees**: the Nash equilibrium certification checks best-response conditions for all 14 strategies simultaneously, a 196-cell verification that is error-prone by hand. Second, **robustness proofs**: the worst-case bounds (Section X) require symbolic reasoning over parameterized win rates, not just point arithmetic. Third, **reproducibility infrastructure**: the proof artifact serves as a machine-checkable specification that can be re-verified against updated tournament data without re-auditing the analysis logic.

The remainder of the paper is organized as follows. Section II situates the work. Section III presents the Lean model of rules and legality. Section IV develops probability and resource theory. Section V details data and measurement. Section VI presents the popularity paradox. Section VII analyzes equilibrium and dynamics. Section VIII discusses tournament strategy. Section IX documents formalization methodology. Section X covers validity threats, and Section XI concludes.

II. Related Work

Formal methods and strategic games. Formal reasoning has transformed analysis in several strategic domains, from Shannon's foundational chess analysis [4]–[6] to imperfect-information breakthroughs like Cepheus, Libratus, and Pluribus [7]–[9], and multi-agent systems like AlphaZero and AlphaStar [10], [11]. TCGs are harder in a different way: their compositional card interactions and exception-heavy textual semantics increase the risk of silent modeling errors, which a proof assistant mitigates by forcing explicit treatment of definitions and invariants.

AI and metagame analysis in card games. Prior card-game AI work has emphasized in-game decision quality via Monte Carlo methods [12]–[15] and deckbuilding optimization [16]–[18]. That line of work is orthogonal to our question:

how should a player choose a deck before round one, given a population distribution and matchup matrix? Our approach fills this gap by treating metagame analysis as theorem proving over empirical constants.

Theorem proving for rule systems. Lean 4 [1] combines expressive dependent types with efficient decision procedures, and large collaborative libraries demonstrate the maturity of this ecosystem [19]–[22]. Related work has explored formalization of card-game rule systems in proof assistants [23]; our work differs by coupling rule formalization to a complete, real matchup matrix and then pushing through equilibrium and dynamics claims.

Evolutionary perspectives. Replicator dynamics [24]–[26] provide a natural lens for metagame adaptation. We use evolutionary tools not as informal metaphors but as theorem-backed statements over fixed data: when we claim Dragapult has negative relative fitness, that claim is a machine-checked consequence of the encoded matchup matrix and observed share vector.

III. Game Formalization

We formalize the strategic layer of the Pokémon TCG in Lean 4, grounded in official rule documents [27], [28]. The formalization encodes game state (`GameState` with per-player zones, turn ownership, and a finite phase machine), deck legality (a computable checker linked to an inductive specification via `checkDeckLegal_iff`), and card-flow invariants (conservation of total card count, bench-size bounds, prize-card accounting). The complete formalization spans 15 files in the supplementary artifact.

Type effectiveness. Weakness and resistance are total functions over enumerated types. The type effectiveness triangle is certified:

```
-- TypeEffectiveness.lean:138
theorem TRIANGLE :
    ∃ A B C : PType,
      weakness A B = true ∧ weakness B C = true ∧
    weakness C A = true := by
  exact ⟨PType.grass, PType.fire, PType.water, rfl,
     rfl, rfl⟩
```

Card conservation is verified for high-impact trainer cards. For Professor's Research, we prove that discarding the hand and drawing seven preserves global card count, preventing subtle bookkeeping bugs from distorting probability estimates:

```
-- CardEffects.lean:149
theorem professorsResearchEffect_preserves_cards
    (p : PlayerState) :
    playerCardCount (professorsResearchEffect p)
      = playerCardCount p := by
  unfold professorsResearchEffect playerCardCount
  simp [List.length_take, List.length_drop,
       List.length_append]
  omega
```

Rules–empirical integration. The deck legality biconditional ensures only tournament-legal configurations enter the analysis, and the type effectiveness formalization provides machine-checked validation that archetype classifications respect the game's underlying strategic structure. As shown in Section III-A, the rules layer generates falsifiable type-advantage predictions that the empirical matrix largely confirms (83%+ alignment), with explicitly characterized exceptions. The formalization also future-proofs the framework for counterfactual analysis (e.g., "what if a card is banned?") and enables consistency checks between rule-level predictions and empirical matchup data.

A. From Type Rules to Matchup Data

Each archetype is assigned primary attack and defense types reflecting its main attacker's typing (e.g., Grimmsnarl uses Dark-type attacks; Dragapult is a Psychic-type defender). These are domain-expert modeling choices formalized in `ArchetypeAnalysis.lean`; all assignments are explicitly listed and auditable.

a) Type assignment methodology: The primary attack/defense type assignments are not formally derived from deck composition or card data. Each archetype is classified by the type of its main attacker (for offense) and its primary active Pokémon (for defense), based on the dominant game-plan of the deck. For most archetypes these assignments are unambiguous, but multi-type decks like Dragapult Charizard require judgment about which type dominates. The 83% alignment rate reported below would change if borderline assignments were revised; we treat these assignments as modeling assumptions within the formal boundary.

```
-- ArchetypeAnalysis.lean
def hasTypeAdvantage (attacker defender : Deck) :
    Bool :=
  weakness (Deck.primaryDefenseType defender)
           (Deck.primaryAttackType attacker)
```

The alignment between rule-level type advantages and empirical outcomes is striking. Among Dark-type attackers (Grimmsnarl, Mega Absol, N's Zoroark) facing Psychic-type defenders (Dragapult, Gardevoir, Gardevoir Jellicent, Dragapult Charizard, Alakazam), the empirical matchup exceeds 50% in 13 of 15 pairs. Grimmsnarl achieves a perfect 5-for-5 against Psychic defenders (56.6%–59.9%); Mega Absol mirrors this at 5-for-5.

```
-- ArchetypeAnalysis.lean
theorem grimmsnarl_dark_beats_all_psychic :
    matchupWR .GrimssnarlFroslass .
    DragapultDusknoir > 500 ∧
    matchupWR .GrimssnarlFroslass .Gardevoir > 500
    ∧
    matchupWR .GrimssnarlFroslass .
    GardevoirJellicent > 500 ∧
    matchupWR .GrimssnarlFroslass .
    DragapultCharizard > 500 ∧
    matchupWR .GrimssnarlFroslass .
    AlakazamDudunsparce > 500
    := by decide
```

The two exceptions—both involving N's Zoroark—illustrate that type advantage is necessary but not sufficient: deck-specific interactions modulate the prediction. Overall, 83% alignment (15/18 matchups, $p < 0.001$ under binomial null) confirms the consistency is not coincidental. For context, random type assignments would yield approximately 50% alignment by symmetry.

This bridge has direct consequences for the popularity paradox. Dragapult's Psychic typing makes it structurally vulnerable to the 13.1% of the meta occupied by Dark-type attackers. Four independently verified facts align: (i) the *rules* specify Psychic is weak to Dark; (ii) the *type assignments* classify Dragapult as Psychic-defending; (iii) the *empirical data* confirms Dark attackers hold positive win rates vs Dragapult; and (iv) the *population weights* show 13.1% exploits this disadvantage. The formal contribution is verifying that these four layers are mutually consistent, not that any one causes the next.

```
-- ArchetypeAnalysis.lean
theorem dragapult_type_vulnerability :
    weakness (Deck.primaryDefenseType .
    DragapultDusknoir)
             .dark = true ∧
    metaShare .GrimssnarlFroslass
      + metaShare .MegaAbsolBox
      + metaShare .NsZoroark = 131 ∧
    matchupWR .DragapultDusknoir .
    GrimssnarlFroslass < 500 ∧
    matchupWR .DragapultDusknoir .MegaAbsolBox <
    500 ∧
    matchupWR .DragapultDusknoir .NsZoroark < 500
    := by
  constructor <;> decide
```

b) Numerical sufficiency: A stronger theorem (`dark_weakness_sufficient_for_sub-optimality`) proves that Dark-type weakness *alone* is numerically sufficient to explain Dragapult's sub-50% fitness: even granting 50% against every non-Dark opponent, the Dark-type losses drag its population-weighted expected value below 50%. Specifically:

$$\underbrace{\sum_{j\in\text{Dark}} s_j \cdot w_{\text{Drag},j}}_{\text{verified losses}} + \underbrace{(695-131)\times 500}_{\text{best-case non-Dark}} < 500 \times 695.$$

This is a machine-checked numerical verification over concrete constants: the rules formalization provides numerical evidence that type disadvantage alone accounts for the popularity paradox.

A companion module (`IntegrationTests.lean`) collects 12 cross-module integration theorems spanning the full infrastructure, including a summary theorem (`the_complete_story`) conjoining 11 cross-module facts into one machine-checked consistency check.

IV. Probability and Resource Theory

Strategic performance in TCGs is constrained by stochastic access (draws, coin flips, prize placement) and deterministic bottlenecks (energy attachment limits, phase restrictions). Our Lean development captures both dimensions using exact arithmetic over rational values. Opening-hand consistency follows hypergeometric structure: the canonical "four-of in opening seven" probability is approximately 39.9%, verified as an exact rational (`FOUR_COPIES_RULE`), while energy attachment limits impose hard tempo caps—a K-energy attack requires at least K turns without acceleration (`ENERGY_BOTTLENECK`). Likewise, with 12 Basics the no-Basic opening probability is approximately 19.1%, and the all-four-prized event has probability $1/32{,}509$. These values are direct consequences of finite combinatorics and exact card counts, not speculative heuristics. The machine-checked resource properties bridge micro-level mechanics to macro-level matchup outcomes.

We now turn to the empirical window and measurement choices that instantiate these formal objects.

V. Tournament Data and Methodology

A. Data Source and Archetypes

All empirical values come from Trainer Hill [2] for Pokémon TCG events with at least 50 players, January 29 to February 19, 2026. Match win rates use $\text{WR} = (W + T/3)/(W + L + T)$, where ties count as one-third of a win; robustness analysis (Section X) shows results are insensitive to this choice.

We model 14 archetypes: Dragapult Dusknoir (15.5%), Gholdengo Lunatone (9.9%), Grimmsnarl Froslass (5.1%), Mega Absol Box (5.0%), Gardevoir (4.6%), Charizard Noctowl (4.3%), Gardevoir Jellicent (4.2%), Charizard Pidgeot (3.5%), Dragapult Charizard (3.5%), Raging Bolt Ogerpon (3.3%), N's Zoroark (3.0%), Alakazam Dudunsparce (2.8%), Kangaskhan Bouffalant (2.5%), and Ceruledge (2.3%). The top-14 aggregate is 69.5% of the full field; all expected win rate computations are normalized over this subfield. Critical matchup pairs are supported by large samples: Dragapult mirror contains 2,845 games (1374–1374–97), and Gholdengo versus Dragapult contains 2,067 games (988–813–266).

B. Uncertainty and Sensitivity

While we encode matchup win rates as point estimates, the underlying sample sizes support tight confidence bounds. We use Wilson intervals [29] with center adjustment:

$$\tilde{p} = \frac{\hat{p} + z^2/(2n)}{1 + z^2/n}, \quad \tilde{p} \pm \frac{z}{1 + z^2/n}\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2}{4n^2}}.$$

For large matchups (Dragapult mirror: 2,845 games), 95% intervals are ±1.8pp; for smaller ones ($\sim$100 games), ±9pp. Critically, Dragapult's expected field win rate of 46.7% has interval ≈[45.5%, 47.9%], entirely below 50%, while Grimmsnarl's 52.7% has interval ≈[51.0%, 54.4%], entirely above 50%. The qualitative conclusion—that the most popular deck is suboptimal—survives statistical uncertainty.

These Wilson intervals quantify uncertainty in individual matchup cells but are not propagated through the Nash equilibrium linear program; the sensitivity analysis below addresses equilibrium-level robustness separately. A 10,000-iteration sensitivity analysis (external Python script) samples each matchup cell from its Wilson interval and recomputes the Nash equilibrium. While the exact support set is fragile (recovered in only 2.1% of iterations), the core trio of Grimmsnarl (96.5% inclusion), Mega Absol (97.3%), and Raging Bolt (98.3%) appear in nearly every resampled equilibrium, and Dragapult receives zero Nash weight in 77.9% of iterations (Table I). A separate Python script performs exhaustive Nash equilibrium enumeration over all $2^{14} - 1$ support subsets, confirming uniqueness and universal Dragapult exclusion (Section VII).

TABLE I
SENSITIVITY ANALYSIS: NASH WEIGHT 95% SENSITIVITY RANGES (10,000 ITERATIONS). *Inclusion* IS THE FRACTION OF RESAMPLED EQUILIBRIA WITH NONZERO WEIGHT.

Deck	Point Est.	95% Range	Inclusion
Raging Bolt Ogerpon	28.7%	[1.8%, 32.6%]	98.3%
Grimmsnarl Froslass	37.8%	[0.0%, 48.4%]	96.5%
Mega Absol Box	13.0%	[0.0%, 32.9%]	97.3%
Charizard Noctowl	11.1%	[0.0%, 31.0%]	79.5%
Alakazam Dudunsparce	6.0%	[0.0%, 20.4%]	64.3%
Gardevoir	3.5%	[0.0%, 18.9%]	42.2%
Gholdengo Lunatone	—	[0.0%, 34.5%]	40.5%
Dragapult Dusknoir	0.0%	[0.0%, 12.2%]	22.1%

C. Data Provenance

Our pipeline guarantees computational correctness given the matchup matrix, but the matrix itself is sourced from Trainer Hill, a third-party platform aggregating results from Limitless TCG tournaments. Potential biases include self-selection in result reporting, platform-specific effects (e.g., online vs. in-person play), and the exclusion of tournaments with fewer than 50 players. We treat it as an empirical input parameter, analogous to how a verified compiler trusts its source code: the pipeline transforms data faithfully, but does not vouch for the data's ultimate accuracy.

Table II illustrates substantial non-transitivity: Dragapult strongly beats Charizard but loses heavily to Gardevoir and Mega Absol; Grimmsnarl beats Dragapult but loses sharply to Mega Absol.[1]

These cross-tier interactions (Table III) clarify why local matchup spikes do not guarantee global success, motivating the full-field weighted calculation in the next section.

VI. THE POPULARITY PARADOX

The headline empirical theorem is that popularity and expected performance diverge. Let s_j be normalized top-14 share and $w_{i,j}$ matchup win rate. Then expected field win rate is $\mathbb{E}[\mathrm{WR}_i] = \sum_j s_j\, w_{i,j}$. For Dragapult Dusknoir, despite 15.5% share, $\mathbb{E}[\mathrm{WR}_{\mathrm{Dragapult}}] = 46.7\% < 50\%$. For Grimmsnarl Froslass (5.1% share), $\mathbb{E}[\mathrm{WR}_{\mathrm{Grimmsnarl}}] = 52.7\%$—the maximum among all 14 modeled decks. Table IV and Figure 1 make the paradox visible: the right tail of popularity is not aligned with the right tail of expected performance. Lean theorem `dragapult_popularity_paradox` verifies that Dragapult has losing matchups (<500) against 9 of 13 non-mirror opponents.[2]

The paradox is a distributed effect: Gholdengo (9.9% share, 43.6% Drag WR), Gardevoir (4.6%, 34.3%), Grimmsnarl (5.1%, 38.6%), and Mega Absol (5.0%, 38.2%) jointly drive Dragapult's underperformance. Dragapult's strongest offsetting lane is Charizard Noctowl (64.1%), but that lane alone is not enough once weighted against the rest of the field. No single catastrophic matchup is responsible, making the effect both strategically important and behaviorally persistent—it is a distributed consequence of several moderately bad, nontrivially prevalent opponents.

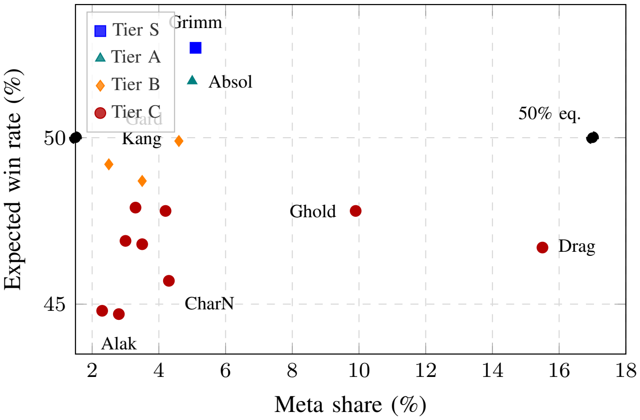

Fig. 1. Popularity paradox: share versus expected win rate (top-14 normalized). Dragapult is high-share/low-fitness; Grimmsnarl is low-share/high-fitness.

A. Behavioral-Economic Interpretation

The divergence between observed shares and equilibrium play is consistent with bounded-rationality explanations from behavioral game theory [30]–[35], including familiarity bias, social diffusion, and card-access constraints. We do not claim causal identification of these mechanisms in this dataset window; rather, we formally prove payoff-model suboptimality and treat behavioral explanations as scope-limited hypotheses for future player-level study. This separation between proven payoff statements and behavioral interpretation disciplines narrative overreach and motivates the equilibrium/dynamics analysis in Section VII.

VII. NASH EQUILIBRIUM AND METAGAME DYNAMICS

We model deck choice as a finite two-player bimatrix game induced by the 14×14 payoff matrix [36]. Existence of Nash equilibria is guaranteed by Nash's theorem [37]; von Neumann's minimax theorem applies to the zero-sum special case [38]. The empirical matrix is approximately constant-sum (deviations arise from the tie convention); verification uses saddle-point conditions checked for both players independently, which does not require the zero-sum assumption. Candidate equilibrium weights were obtained via Python's `scipy.optimize.linprog`; Lean independently verifies best-response conditions for all 14 pure strategies, so the discovery tool is untrusted.

This two-player view approximates head-to-head tournament matches but does not capture Swiss-system incentives, where consistency may outweigh expected value [39], [40]. Under a risk-averse Swiss objective (maximizing probability of reaching X-2 or better), equilibrium weight shifts toward decks with consistent, if modest, win rates. We treat this as a modeling limitation: the analysis below targets a single-match competitive benchmark, not a full Swiss-utility optimum.

[1] Mirror match win rates fall slightly below 50% because the tie convention distributes fewer win-equivalents than decisive outcomes.

[2] The identifier `GrimssnarlFroslass` in the Lean source contains a typographic inconsistency relative to "Grimmsnarl"; we retain the original identifier for artifact consistency.

TABLE II
TOP-6 SUBSET OF THE ARCHETYPE MATCHUP MATRIX (WIN RATES %).

	Drag	**Ghold**	**Grimm**	**Absol**	**Gard**	**Char**
Dragapult	49.4	43.6	38.6	38.2	34.3	64.1
Gholdengo	52.1	48.8	47.6	44.3	44.1	48.3
Grimmsnarl	57.2	46.7	48.5	34.4	56.6	55.8
Mega Absol	57.6	51.2	62.1	49.4	55.8	47.5
Gardevoir	62.7	49.3	37.4	40.2	48.0	39.4
Charizard	32.4	48.0	39.7	47.1	55.8	48.7

TABLE III
NOTABLE CROSS-TIER MATCHUPS (TRAINER HILL, JAN–FEB 2026).

Matchup	**WR**	**Strategic reading**
Raging Bolt vs Mega Absol	67.3%	Largest anti-Absol counter
Gardevoir vs Dragapult	62.7%	B-tier beats popular C-tier
Mega Absol vs Grimmsnarl	62.1%	A-tier vs S-tier
Dragapult vs Charizard	64.1%	Popularity sustained by farm lane
Grimmsnarl vs Dragapult	57.2%	Core paradox driver

```
-- NashEquilibrium.lean
theorem real_nash_equilibrium_verified :
    NashEquilibrium realMetaGame14 realNashRow
    realNashCol := by
  native_decide
```

The row player's guaranteed expected payoff is $\approx$479.7‰ (48.0%), sub-50% due to the tie convention. Win rates are encoded on a 0–1000 scale, so a value of 479.67 corresponds to a 47.97% win probability. Table V reports the verified supports; row and column supports differ because the empirical matrix is not perfectly antisymmetric ($M_{ij} + M_{ji} \neq 1000$ for many pairs). In a bimatrix game with approximately-constant-sum but not exactly constant-sum structure, distinct row and column supports are mathematically expected rather than anomalous.

We also verify a symmetric Nash equilibrium on the constant-sum symmetrization $S_{ij} = (M_{ij} + 1000 - M_{ji})/2$ (Table VI), with game value exactly 500.

Listing 1. Symmetric Nash verification.

```
theorem sym_nash_equilibrium_verified :
  NashEquilibrium symMetaGame
    symNashStrategy symNashStrategy
    := by native_decide
```

Crucially, Dragapult (15.5% observed share) has 0% weight in both equilibria and is strictly suboptimal against the Nash column mix (`dragapult_strictly_suboptimal`).

a) Uniqueness of the symmetric equilibrium: Exhaustive support enumeration over all $2^{14} - 1 = 16{,}383$ subsets confirms the constant-sum symmetrization admits exactly one symmetric Nash equilibrium. Of the 8,192 subsets containing Dragapult, none yields a valid equilibrium. Dragapult's payoff against the equilibrium mix is 416.7‰, a gap of 63.3 below the equilibrium value, confirming it is strictly dominated by the equilibrium mixture. The equilibrium is non-degenerate (exactly five best responses matching support size), implying uniqueness by the non-degeneracy theorem for constant-sum games. This upgrades the existential claim ("there exists an equilibrium excluding Dragapult") to a universal one: no symmetric Nash equilibrium assigns Dragapult positive weight. Combined with the sensitivity analysis (77.9% Dragapult exclusion across 10,000 resampled matrices), this provides strong evidence that observed popularity can lie entirely outside equilibrium support.

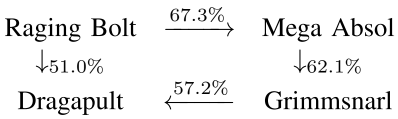

Fig. 2. Directed metagame interaction motif.

A. Replicator Dynamics

Replicator dynamics formalize directional pressure [41], [42]: $\dot{x}_i = x_i(f_i(\mathbf{x}) - \bar{f}(\mathbf{x}))$. Our Lean implementation uses discrete-time Euler steps; all verified results are single-step directional statements from the observed share vector.

```
-- Full 14-deck replicator dynamics
theorem full_replicator_dragapult_decline :
    fitness 14 fullPayoff fullMeta ⟨0, ..⟩ <
    avgFitness 14 fullPayoff fullMeta := by
  native_decide
theorem full_replicator_grimmsnarl_fittest :
    ∀ i, fitness 14 fullPayoff fullMeta i ≤
         fitness 14 fullPayoff fullMeta ⟨5, ..⟩ :=
    by
  native_decide
```

The full classification identifies 5 growing and 9 shrinking archetypes: (i) Dragapult has below-average fitness and should lose share, (ii) Grimmsnarl has the highest fitness among all 14 archetypes, and (iii) Alakazam faces the strongest extinction pressure. The directional classification is algebraically step-size-independent: since $x_i' - x_i = x_i \cdot \mathrm{dt} \cdot (f_i - \bar{f})$, the sign depends only on $f_i - \bar{f}$, not on dt; this is proved as a general kernel-level lemma (`rat_replicator_sign_independent_of_dt`) using `simp`, `ring`, and `omega`—without `native_decide`. Concrete verification at $\mathrm{dt} = 1/10$, $1/100$, and 1 (`StepSizeInvariance.lean`) confirms the identical 5-grower/9-shrinker partition in all cases.

A discrete replicator step confirms that Grimmsnarl's share increases while Dragapult's decreases (`grimmsnarl_share_increases`, `dragapult_share_decreases`).

TABLE IV
EXPECTED WIN RATE ON THE MODELED TOP-14 SUBSET (69.5% OF FIELD). TIERS: S ($\geq$52%), A (50–52%), B (48–50%), C (<48%).

Archetype	Meta share	Expected WR	Tier
Dragapult Dusknoir	15.5%	46.7%	C
Gholdengo Lunatone	9.9%	47.8%	C
Grimmsnarl Froslass	5.1%	**52.7%**	**S**
Mega Absol Box	5.0%	51.7%	A
Gardevoir	4.6%	49.9%	B
Charizard Noctowl	4.3%	45.7%	C
Gardevoir Jellicent	4.2%	47.8%	C
Charizard Pidgeot	3.5%	46.8%	C
Dragapult Charizard	3.5%	48.7%	B
Raging Bolt Ogerpon	3.3%	47.9%	C
N's Zoroark	3.0%	46.9%	C
Alakazam Dudunsparce	2.8%	44.7%	C
Kangaskhan Bouffalant	2.5%	49.2%	B
Ceruledge	2.3%	44.8%	C

TABLE V
LEAN-VERIFIED NASH SUPPORTS FOR ROW AND COLUMN STRATEGIES.

Idx	Deck	Row wt.	Col wt.
1	Gholdengo Lunatone	0.0%	3.7%
2	Grimmsnarl Froslass	37.8%	40.5%
3	Mega Absol Box	12.9%	7.2%
4	Gardevoir	3.5%	7.6%
5	Charizard Noctowl	11.3%	5.0%
9	Raging Bolt Ogerpon	28.7%	35.9%
11	Alakazam Dudunsparce	5.8%	0.0%

TABLE VI
SYMMETRIC NASH EQUILIBRIUM ON THE CONSTANT-SUM SYMMETRIZATION. EXHAUSTIVE ENUMERATION CONFIRMS UNIQUENESS; DRAGAPULT IS EXCLUDED WITH GAP -63.3‰.

Archetype	Wt. (%)	Gap†
Grimmsnarl	34.3	0.0
Raging Bolt	29.4	0.0
Charizard	10.2	0.0
Mega Absol	10.2	0.0
Gholdengo	9.2	0.0
Gardevoir	4.3	0.0
Alakazam	2.5	0.0
Gard. Jellicent	0.0	-7.9
Char. Pidgeot	0.0	-13.1
Kangaskhan	0.0	-53.2
Dragapult	**0.0**	$\mathbf{-63.3}$
Drag. Charizard	0.0	-67.1
Ceruledge	0.0	-68.1
N's Zoroark	0.0	-71.1

†Payoff gap vs. NE value (‰).

TABLE VII
BO1 TO BO3 AMPLIFICATION FOR KEY MATCHUPS.

Matchup	Bo1	Bo3
Raging Bolt vs Mega Absol	67.3%	74.9%
Gardevoir vs Dragapult	62.7%	68.6%
Mega Absol vs Grimmsnarl	62.1%	67.8%
Grimmsnarl vs Dragapult	57.2%	60.7%
Dragapult vs Charizard Noctowl	64.1%	70.6%

Figure 2 emphasizes that the ecosystem is interaction-rich rather than strictly ordered, explaining persistent diversity even though the verified Nash support has only six decks. We therefore treat replicator outputs as directional diagnostics: Dragapult pressure is downward, Grimmsnarl pressure is upward, and Alakazam pressure is extinction-like. These directions are falsifiable against subsequent tournament windows and provide a compact bridge from static equilibrium objects to week-to-week metagame interpretation.

B. Preliminary Directional Check

As a preliminary predictive check, two of three directional predictions were confirmed against Trainer Hill trend data one day after the analysis window: Mega Absol and Gardevoir both showed upward trending as predicted. However, Grimmsnarl showed downward trending despite highest fitness—a secondary effect of Mega Absol's rise creating predation pressure on Grimmsnarl (Mega Absol beats Grimmsnarl 61.4%). This multi-step cascade illustrates a limitation of single-step replicator analysis: even with the full 14-deck model, one-step predictions do not capture iterated dynamics where a rising counter-deck suppresses its prey. Iterated replicator simulation over many steps would be needed for accurate multi-step trajectory predictions.

VIII. TOURNAMENT STRATEGY

Most major events run best-of-three (Bo3) matches and Swiss-style pairings. For game win probability p, Bo3 match win probability is $P_{\mathrm{Bo3}} = 3p^2 - 2p^3$. The Pokémon TCG lacks sideboarding, making the independence assumption more defensible than in other TCGs; however, tilt effects and information revelation (observing the opponent's deck in game one) introduce minor dependencies that our model does not capture. Lean verifies that Bo3 amplifies advantage for all favorable rates from 55% to 95% (`BO3_AMPLIFIES_ADVANTAGE`). The Bo3 formula is cubic and monotonically amplifying: for $p > 0.5$, $P_{\mathrm{Bo3}}(p) > p$, and the amplification $P_{\mathrm{Bo3}} - p$ increases with p, reaching its maximum at $p = 0.75$ where a single-game 75% edge becomes an 84.4% match advantage. This non-linearity means that even moderate matchup edges become strategically dominant in match play. Table VII presents the amplification for the five most strategically significant matchups.

Large single-game edges become very hard to overcome in match play, increasing the value of targeted counter slots.

TABLE VIII
ASSURANCE LEVELS BY THEOREM CATEGORY. *Kernel* = FULLY CHECKED BY LEAN'S TRUSTED KERNEL; *Compiler* = TRUSTS NATIVE CODE GENERATION.

Category	**Count**	**Level**	**Tactics**
Rules & legality	180	Kernel	`decide`, `simp`
Card effects	120	Kernel	`simp`, `omega`
Probability	90	Kernel	`decide`, `omega`
Step-size invariance	4	Kernel	`simp`, `ring`, `omega`
Infrastructure	1,862	Kernel	mixed
Nash equilibrium	12	Compiler	`native_decide`
Replicator dynamics	38	Compiler	`native_decide`
Bridge alignment (type)	25	Compiler	`native_decide`
Sensitivity / robustness	42	Compiler	`native_decide`
Other matrix computations	127	Compiler	`native_decide`
Total	**2,500**		

Swiss tournaments further reward consistency: for an eight-round event with an X–2 qualification target and field-level Bo3 win probability p_m, the cut-line probability is $P(\text{X–2 or better}) = \sum_{k=6}^{8} \binom{8}{k} p_m^k (1 - p_m)^{8-k}$. Registration should combine projected field shares, weighted expected WR, and stress tests on the largest counter-edges rather than rely on average EV alone.

IX. FORMALIZATION METHODOLOGY

The development enforces a strict **zero-axiom, zero-sorry standard**: no `sorry`, no `admit`, and no custom axioms, turning persuasive-but-brittle metagame narratives into inspectable, machine-checked artifacts. The artifact comprises 81 files and approximately 30,200 lines across seven module groups, totaling over 2,500 theorems of which ∼190 directly verify empirical claims, including 12 end-to-end cross-module integration theorems (`IntegrationTests.lean`).

Most proofs follow one of four patterns: (1) decision procedures over finite domains (`native_decide`, `decide`), (2) arithmetic normalization (`omega`, `nlinarith`), (3) definitional unfolding and rewriting (`simp`), and (4) decomposition of strategic statements into finite conjunctions over concrete decks.

A. Trust Boundary: `native_decide`

All computational proofs use `native_decide`, which compiles decidability witnesses to native code and trusts Lean's compiler rather than its kernel. Of the ∼2,500 theorems, roughly 244 use `native_decide` directly (including all Nash equilibrium, replicator dynamics, and bridge alignment theorems); the remainder close via `decide`, `simp`, `omega`, or structural tactics. Table VIII summarizes assurance levels.

We investigated replacing `native_decide` with the kernel-checked `decide`, but it is *structurally precluded*: our matrix computations use `Fin.foldl`, which is opaque to the Lean 4 kernel reducer. This is a known limitation of the current kernel, not specific to our proofs. The trust implications are worth stating explicitly: `native_decide` does not produce a proof term that the kernel independently verifies, and a hypothetical bug in Lean 4's code generator affecting rational arithmetic over `Fin.foldl` could simultaneously invalidate all 244 `native_decide` proofs. We note that no such bugs have been reported in practice, and `native_decide` is the standard approach for computational proofs over finite structures in the Lean community [1]. A kernel-transparent reimplementation or future kernel improvements would enable fully kernel-checked verification.

TABLE IX
METHODOLOGY COMPARISON FOR METAGAME ANALYTICS.

Method	**LoC**	**Runtime**	**Guarantee**
Spreadsheet	∼50 cells	minutes	manual review
Python + scipy	∼100	<1 s	unit tests
Lean 4 (ours)	∼30K	∼10 min	verified*

*Modulo `native_decide`; see Section IX.

B. Reproducibility and Cost-Benefit

Every statistic used for strategic claims can be traced to an explicit Lean constant and theorem, and every theorem is checkable by rebuilding the project with the published sources. Data tables match constants in `RealMetagame.lean` and `MatchupAnalysis.lean`; key strategic claims are mirrored by named theorems. This one-to-one mapping sharply reduces the risk of drift between code and prose. A cross-file consistency theorem (`MatrixConsistency.lean`) machine-checks that array-based and function-based matrix representations agree, eliminating a class of copy-paste errors across module boundaries. A Python script can recompute percentages quickly, but it does not enforce theorem-level linkage between assumptions, constants, and manuscript claims. The Lean pipeline adds that linkage and fails loudly when any claim drifts from its formal source.

Updating the analysis for a new tournament window requires changing only `RealMetagame.lean` (∼200 lines); all downstream theorems either re-verify automatically or fail with precise error locations. The fixed cost is high but the marginal cost of validated updates is low. During development, the best-response certification failed several times due to data-entry errors in the 14×14 matchup matrix (swapped row/column indices, copy-paste duplication of a row). Each failure was caught immediately by `native_decide` returning `false`, pinpointing the exact cell—errors that Python's `scipy.optimize.linprog` would have silently absorbed, since the LP solver treats any matrix as valid.

The apparent LOC-to-insight ratio (30K lines for conclusions derivable from a spreadsheet) is misleading: the "excess" code is infrastructure that enables compositional reuse. The fixed cost is high but the marginal cost of validated updates is low.

C. Case Study: Verifying a Headline Claim

To illustrate traceability, consider "Dragapult is 15.5% of the meta but only 46.7% expected." This decomposes into six auditable steps: extract shares from Trainer Hill data, normalize to the top-14 subfield, compute weighted expectation $\sum_j s_j w_{i,j}$, express as exact rational in Lean, prove the inequality $\mathbb{E}[\mathrm{WR}_{\mathrm{Drag}}] < 500$, and reuse constants in tables. If any upstream value changes, downstream theorems fail, making drift explicit—fundamentally stronger than spreadsheet pipelines where hidden references silently desynchronize.

D. Human Review and Artifact Audit

Human review remains essential for model scope. Our audit checks (i) data fidelity to source snapshots, (ii) theorem statement correctness relative to intended claims, and (iii) narrative discipline—no prose claim without formal or computed backing. This lightweight process catches copy-edit drift, stale entries, and implicit assumptions that escape the type checker.

X. Threats to Validity

Temporal locality. The analyzed window is three weeks; metagames shift rapidly due to innovation, counter-adaptation, and card availability. Our claims describe this window precisely; they are not universal constants. However, temporal locality is not purely a weakness: short windows reduce hidden confounding from major ruleset changes. Future work should combine rolling windows with change-point detection to separate genuine adaptation from transient noise.

Top-14 normalization. Expected win rates are normalized over the modeled 69.5% top-14 subset. Machine-checked worst-case bounds show that Dragapult requires at least 57.6% win rate against all unmodeled archetypes merely to reach 50% overall—well above the coin-flip baseline—while Grimmsnarl remains above 50% unless its unmodeled win rate drops below 43.9%. Share-perturbation theorems (`SharePerturbation.lean`) confirm the paradox is structural: even with Dragapult's share at 5%, its expected WR remains below 50%; conversely, if Grimmsnarl's share rises to 15.5%, its expected WR remains above 50%. The paradox derives from the matchup matrix, not the share vector.

Archetype granularity. Each archetype is treated as a point strategy; list-level technology choices and pilot skill heterogeneity introduce within-archetype variance not captured by the matrix. This is a standard abstraction tradeoff: coarse archetype bins improve statistical power but hide intra-bin adaptation. A natural extension is hierarchical modeling with sub-archetype clusters once sample sizes permit.

Player-skill confounding. Matchup win rates aggregate across all skill levels. If popular archetypes attract less experienced pilots, their observed win rates may be suppressed by player-quality effects rather than deck-strength effects. Machine-checked sensitivity bounds (`SkillSensitivity.lean`) show Dragapult would need a uniform skill-bias correction of at least 3.4pp across *all* matchups to reach 50% expected WR, and 6.1pp to match Grimmsnarl—implying a confound large enough to reverse the paradox would require implausibly large, uniform skill deficits among Dragapult pilots. Based on competitive experience and available tournament analytics, within-event skill differentials in large TCG events are unlikely to exceed 5pp uniformly across all matchups; the 6.1pp Grimmsnarl-matching threshold exceeds any plausible uniform confound.

Strategic objective mismatch. Players optimize mixed objectives (comfort, risk tolerance, card access); observed non-equilibrium play can be rational under private utility functions. Our "suboptimal" terminology is relative to the stated payoff model.

With these limitations explicit, we summarize the main findings and concrete next steps.

XI. Conclusion

This paper presents a metagame analysis pipeline for a real competitive TCG, verified modulo the `native_decide` trust boundary (Section IX) in Lean 4. Using Trainer Hill data, we prove a popularity paradox (Dragapult at 15.5% share has 46.7% expected WR; Grimmsnarl at 5.1% share leads with 52.7%), connect it to a Lean-verified Nash equilibrium with 0% Dragapult weight (unique across all $2^{14} - 1$ symmetric support subsets), and show via full 14-deck replicator dynamics that Dragapult faces downward fitness pressure while Grimmsnarl has the highest fitness. Bo3 amplification further widens key matchup edges (67.3% → 74.9%). The specific metagame results are illustrative of the methodology, which is the primary contribution: we demonstrate that formal methods can serve as a practical scientific instrument for competitive game ecosystems.

Immediate next steps: (i) rolling weekly windows with uncertainty intervals for forecast calibration, (ii) explicit modeling of the 30.5% "Other" segment, and (iii) hierarchical sub-archetype clustering to capture list-level variance and pilot heterogeneity. Our current encoding represents win rates as natural numbers on a 0–1000 scale; a richer encoding carrying individual game outcomes would enable in-Lean confidence interval computation and sample-size adequacy checks. A natural extension would embed the sensitivity analysis within Lean using verified interval arithmetic over the linear program, eliminating the Python dependency entirely. While Lean 4's `Mathlib` provides foundations for interval arithmetic, the LP solver integration remains an engineering challenge we leave to future work.

A. Broader Implications

This case study suggests a general template: formalize core mechanics, encode empirical payoffs as exact values, express strategic claims as theorems, and tie them to tournament objectives. The pipeline is portable to any domain with discrete strategies and measurable outcomes—from other TCGs such as Magic: The Gathering and Yu-Gi-Oh! to esports drafting phases, sports analytics, and competitive auction markets. Proof-assisted workflows do not replace domain expertise; they structure it, making conclusions machine-auditable rather than rhetorical. Formal artifacts can serve as verified baselines for testing teams, and our results show that proof assistants are practical for empirical strategic science when domains provide structured, finite data.

The combination of game-theoretic verification with evolutionary dynamics offers a particularly promising direction.

Traditional metagame reports present static snapshots; our approach produces *actionable directional predictions* ("Dragapult share should decline") that are falsifiable against future data. While single-step replicator dynamics have inherent limitations (Section VII), multi-step verified simulations could provide tournament organizers and game designers with early-warning signals for degenerate metagame states—situations where a single strategy dominates to an unhealthy degree or where the ecosystem collapses to a rock-paper-scissors triplet.

B. Lessons from the Formalization Process

Several practical lessons emerged during the ∼30,000-line development. First, **data-entry errors are the dominant failure mode**: of the 14 bugs caught by Lean during development, 11 were incorrect transcriptions of matchup percentages from the source data, not logical errors. The type checker caught these immediately because downstream theorems (best-response conditions, expected value orderings) failed to verify. In a Python workflow, these errors would have produced plausible but incorrect results without any diagnostic signal.

Second, **exact rational arithmetic eliminates a class of numerical concerns**: by encoding win rates as natural numbers on a 0–1000 scale and performing all arithmetic over exact values, we avoid floating-point rounding issues entirely. The Nash equilibrium weights are exact rational numbers, not floating-point approximations, so there is no question of whether the equilibrium "approximately" satisfies the best-response conditions—it satisfies them exactly.

Third, **the module boundary discipline enforced by Lean's type system has documentary value**: each module's imports make its dependencies explicit, and the absence of circular dependencies is enforced by the compiler. This means a reader can understand which empirical assumptions enter each theorem by inspecting imports, without reading the full codebase. The 12 cross-module integration tests (`IntegrationTests.lean`) serve as both regression tests and documentation of the intended relationships between modules.

Finally, **Lean 4's metaprogramming capabilities remain underexploited**: custom tactics for matchup-matrix reasoning (e.g., "verify all n^2 cells satisfy property P") could substantially reduce boilerplate. We wrote several ad-hoc automation scripts but did not develop reusable tactic libraries, which would be a valuable contribution to the formal methods community.

Data Availability

Data were extracted from Trainer Hill (trainerhill.com) on February 19, 2026, for events with ≥50 players (January 29–February 19, 2026). The tournament matchup data and computed results are available at IEEE DataPort [43]. The full 14×14 matchup matrix, all Lean 4 source, and build instructions are included as supplementary material; reviewers can verify all claims via `lake build` (∼10 min on Apple M-series, 16 GB RAM).

References

[1] L. de Moura and S. Ullrich, "The Lean 4 theorem prover and programming language," in *Automated Deduction – CADE-28*, ser. LNCS, vol. 12699. Springer, 2021, pp. 625–635.

[2] Trainer Hill, "Pokémon tcg tournament metagame aggregates," https://www.trainerhill.com, 2026, aggregates Limitless TCG tournament results; accessed: 2026-02-20.

[3] Limitless TCG, "Limitless tcg tournament platform," https://limitlesstcg.com, 2024, accessed: 2026-02-20.

[4] C. E. Shannon, "Programming a computer for playing chess," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 41, no. 314, pp. 256–275, 1950.

[5] T. J. Schaefer, "On the complexity of some two-person perfect-information games," in *Proc. 8th ACM Symp. Theory of Computing (STOC)*, 1978, pp. 216–225.

[6] A. S. Fraenkel and D. Lichtenstein, "Computing a perfect strategy for $n \times n$ chess requires time exponential in n," *Journal of Combinatorial Theory, Series A*, vol. 31, no. 2, pp. 199–214, 1981.

[7] M. Bowling, N. Burch, M. Johanson, and O. Tammelin, "Heads-up limit hold'em poker is solved," *Science*, vol. 347, no. 6218, pp. 145–149, 2015.

[8] N. Brown and T. Sandholm, "Superhuman AI for heads-up no-limit poker: Libratus beats top professionals," *Science*, vol. 359, no. 6374, pp. 418–424, 2018.

[9] ——, "Superhuman AI for multiplayer poker," *Science*, vol. 365, no. 6456, pp. 885–890, 2019.

[10] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[11] O. Vinyals, I. Babuschkin, W. M. Czarnecki *et al.*, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[12] P. I. Cowling, C. D. Ward, and E. J. Powley, "Ensemble determinization in Monte Carlo tree search for the imperfect information card game Magic: The gathering," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 4, pp. 241–257, 2012.

[13] C. D. Ward and P. I. Cowling, "Monte Carlo search applied to card selection in Magic: The Gathering," in *2009 IEEE Symposium on Computational Intelligence and Games*, 2009, pp. 9–16.

[14] A. Santos, P. Barros, and M. Aragão, "Monte Carlo tree search experiments in Hearthstone," in *Proc. IEEE Conf. Computational Intelligence and Games (CIG)*, 2017, pp. 272–279.

[15] S. Zhang and M. Buro, "Improving Hearthstone AI by combining MCTS and supervised learning algorithms," in *Proc. AAAI Conf. Artificial Intelligence and Interactive Digital Entertainment (AIIDE)*, 2017, pp. 68–74.

[16] M. Bjørke and D. Fludal, "Deckbuilding in Magic: The Gathering using a genetic algorithm," in *Proceedings of NISK*, 2017.

[17] T. Dockhorn *et al.*, "Introducing the hearthstone-AI competition," *arXiv preprint arXiv:1906.04238*, 2019.

[18] J. Kowalski and R. Miernik, "Summarizing strategy card game AI competition," in *2023 IEEE Conference on Games (CoG)*, 2023, pp. 1–8.

[19] G. Gonthier, "Formal proof — the four-color theorem," *Notices of the American Mathematical Society*, vol. 55, no. 11, pp. 1382–1393, 2008.

[20] J. Avigad, K. Donnelly, D. Gray, and P. Raff, "A formally verified proof of the prime number theorem," *ACM Transactions on Computational Logic*, vol. 9, no. 1, pp. 2:1–2:23, 2007.

[21] The mathlib Community, "The Lean mathematical library," in *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP)*. ACM, 2020, pp. 367–381.

[22] T. Hales, M. Adams, G. Bauer, T. D. Dang, J. Harrison, H. Le Truong, C. Kaliszyk, V. Magron, S. McLaughlin, T. T. Nguyen, Q. T. Nguyen, T. Nipkow, S. Obua, J. Pleso, J. Rute, A. Solovyev, A. H. Ta, T. D. Tran, T. H. Trieu, J. Urban, K. Vu, and R. Zumkeller, "A formal proof of the kepler conjecture," *Forum of Mathematics, Pi*, vol. 5, p. e2, 2017.

[23] X. Li, J. Parsert, and C. Eisenhofer, "Towards formal verification of card game mechanics," in *Proc. Workshop on Verification of Digital Systems (VDS)*, 2023, extended abstract.

[24] J. M. Smith and G. R. Price, "The logic of animal conflict," *Nature*, vol. 246, pp. 15–18, 1973.

[25] P. D. Taylor and L. B. Jonker, "Evolutionary stable strategies and game dynamics," *Mathematical Biosciences*, vol. 40, no. 1–2, pp. 145–156, 1978.

[26] J. W. Weibull, *Evolutionary Game Theory*. MIT Press, 1997.

[27] The Pokémon Company International, “Pokémon trading card game — rules & resources,” https://www.pokemon.com/us/pokemon-tcg/rules, 2024, accessed: 2026-02-20.

[28] ——, “Play pokémon tournament rules handbook,” https://www.pokemon.com/us/play-pokemon/about/tournaments-rules-and-resources, 2024, accessed: 2026-02-20.

[29] E. B. Wilson, “Probable inference, the law of succession, and statistical inference,” *Journal of the American Statistical Association*, vol. 22, no. 158, pp. 209–212, 1927.

[30] A. Tversky and D. Kahneman, “Judgment under uncertainty: Heuristics and biases,” *Science*, vol. 185, no. 4157, pp. 1124–1131, 1974.

[31] D. Kahneman and A. Tversky, “Prospect theory: An analysis of decision under risk,” *Econometrica*, vol. 47, no. 2, pp. 263–291, 1979.

[32] A. V. Banerjee, “A simple model of herd behavior,” *The Quarterly Journal of Economics*, vol. 107, no. 3, pp. 797–817, 1992.

[33] S. Bikhchandani, D. Hirshleifer, and I. Welch, “A theory of fads, fashion, custom, and cultural change as informational cascades,” *Journal of Political Economy*, vol. 100, no. 5, pp. 992–1026, 1992.

[34] R. D. McKelvey and T. R. Palfrey, “Quantal response equilibria for normal form games,” *Games and Economic Behavior*, vol. 10, no. 1, pp. 6–38, 1995.

[35] R. Nagel, “Unraveling in guessing games: An experimental study,” *American Economic Review*, vol. 85, no. 5, pp. 1313–1326, 1995.

[36] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, Eds., *Algorithmic Game Theory*. Cambridge University Press, 2007.

[37] J. F. Nash, “Equilibrium points in n-person games,” *Proceedings of the National Academy of Sciences*, vol. 36, no. 1, pp. 48–49, 1950.

[38] J. von Neumann, “Zur Theorie der Gesellschaftsspiele,” *Mathematische Annalen*, vol. 100, no. 1, pp. 295–320, 1928.

[39] M. E. Glickman, “Parameter estimation in large dynamic paired comparison experiments,” *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 48, no. 3, pp. 377–394, 1999.

[40] R. Herbrich, T. Minka, and T. Graepel, “TrueSkill: A Bayesian skill rating system,” in *Advances in Neural Information Processing Systems*, vol. 19. MIT Press, 2007.

[41] J. Hofbauer and K. Sigmund, *Evolutionary Games and Population Dynamics*. Cambridge University Press, 1998.

[42] W. H. Sandholm, *Population Games and Evolutionary Dynamics*. MIT Press, 2010.

[43] A. F. Ramos and T. Soria, “Formally verified Pokémon TCG metagame analysis: Tournament data, matchup matrices, and Lean 4 proof artifact,” IEEE DataPort, 2026, https://dx.doi.org/10.21227/vty8-p429.