

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 formally verified analysis of competitive Pokémon Trading Card Game metagame dynamics using Lean 4 and real tournament data. Our formal development spans approximately 30,000 lines across 75 Lean files and proves over 2,500 theorems without `sorry`, `admit`, or custom axioms. Our strategic model covers the top 14 archetypes (69.5% of the field); the remaining 30.5% is excluded from the game-theoretic model but bounded in robustness analysis. We analyze Trainer Hill tournament data from January–February 2026 (50+ player events), covering a complete 14×14 matchup matrix. The central result is a verified popularity paradox: Dragapult Dusknoir is the most played deck (15.5%) but has only 46.7% expected win rate, while Grimmsnarl Frolass (5.1% share) achieves 52.7%. We report a machine-checked Nash equilibrium with six-deck support per player; best-response conditions are verified via `native_decide` for all 14 pure strategies, and Dragapult receives 0% equilibrium weight. On representative 4-deck and 5-deck subgames of the observed metagame, replicator dynamics predict Dragapult decline and Ceruleedge extinction pressure. A 10,000-iteration bootstrap analysis confirms the equilibrium’s qualitative stability: the core support decks appear in >96% of resampled equilibria, while Dragapult receives zero weight in 77.9%. These results demonstrate that formal methods can produce empirically grounded and strategically actionable conclusions in competitive game ecosystems.

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. Players must make two coupled decisions: how to play each game state and which deck to register. The first decision is local and tactical; the second is global and game-theoretic. 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. At the same time, the domain is difficult for informal reasoning: hidden information, stochastic effects, and nonlinear tournament incentives make intuition unreliable even for experienced players.

Formal methods offer an attractive remedy. By encoding game semantics in Lean 4 [1] and proving strategic statements directly over exact data representations, we separate factual claims from narrative claims. A statement in the paper is either

derivable from formally checked definitions and theorems or it is excluded. 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 (including all 14 pure-strategy best-response checks) and replicator-dynamics statements on real-data representative subgames, and (iv) tournament-objective transforms (Bo3, Swiss). The in-game rules formalization serves as supporting infrastructure guaranteeing legality and enabling future counterfactual analysis.

Our empirical foundation is Trainer Hill metagame data, aggregated from Limitless tournament records [2], [3], for 50+ player tournaments from January 29 to February 19, 2026. Within this fixed window, we model the top 14 archetypes and their full pairwise matchup matrix. The resulting matrix contains enough granularity to support rigorous expected-value computations, equilibrium analysis, and evolutionary dynamics without introducing synthetic assumptions about win rates.

This paper makes four concrete contributions aligned with this thesis. First, we formalize the rules and legality substrate that supports trustworthy ingestion and future counterfactual analysis. Second, we encode real metagame data as exact values and verify expected-value computations, including the popularity paradox. Third, we derive machine-checked Nash-equilibrium statements over the full observed matrix and replicator-dynamics statements over real-data representative subgames, including complete best-response equilibrium certification. Fourth, we verify tournament-objective transforms from single-game payoffs to best-of-three and Swiss-relevant decision criteria.

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.

A. Motivating Tournament Scenario

Consider a player preparing for a 10-round open tournament. Community sentiment says Dragapult is “the deck to beat” because it is popular and appears frequently in streamed matches. Our data show why this shortcut fails: popularity is shaped by familiarity, deck cost, and social diffusion, while expected win rate is determined by pairwise matchups weighted by opponent frequencies. These quantities are related but not equivalent, and a formal model recasts the decision as a transparent optimization problem with explicit assumptions. Section VI makes this precise.

II. RELATED WORK

A. Formal Methods and Strategic Games

Formal reasoning has transformed analysis in several strategic domains. In classical games, Shannon’s foundational analysis of chess programming and later complexity results for generalized chess established the computational stakes of strategic reasoning [4]–[6]. In imperfect-information settings such as poker, game-theoretic systems like Cepheus, Libratus, and Pluribus [7]–[9] show that equilibrium reasoning can scale when abstractions are carefully managed. At larger multi-agent scale, AlphaZero and AlphaStar demonstrate superhuman play in complex domains [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.

B. AI and Metagame Analysis in Card Games

Prior card-game AI work has emphasized in-game decision quality, including Monte Carlo methods for games such as Magic and Hearthstone [12]–[15]. Related work also includes deckbuilding optimization and competition environments for TCG AI [16]–[18]. That line of work is valuable but orthogonal to the question we study: how should a player choose a deck before round one, given a population distribution and matchup matrix?

Metagame-level analysis appears frequently in practitioner content but rarely as reproducible formal scholarship. The missing ingredients are a fixed data definition, a mathematically explicit payoff model, and mechanically checked claims. Our approach fills this gap by treating metagame analysis as theorem proving over empirical constants.

C. Theorem Proving for Rule Systems

Lean 4 [1] has become a practical environment for large-scale formalization because it combines expressive dependent types with efficient decision procedures. Large collaborative libraries and landmark formal proofs further show the maturity of this ecosystem [19]–[22]. For game-rule modeling, dependent types are particularly useful: constraints such as bounded bench size, deck legality predicates, and well-formed transitions can be encoded directly as propositions attached to data.

Related work has explored formalization of card-game effects and rule systems in proof assistants [23], establishing feasibility for this class of domains. Our work differs by coupling rule formalization to a complete, real matchup matrix and then pushing through equilibrium and dynamics claims tied to observed tournament distributions.

D. Evolutionary and Behavioral Perspectives

Replicator dynamics [24]–[26] provide a natural lens for metagame adaptation: strategies with above-average fitness gain share, while below-average strategies lose share. In practice, observed metagames often drift slowly because human behavior is not perfectly rational. This motivates interpreting deviations from equilibrium in behavioral-economic terms rather than treating them as model failure.

The contribution here is methodological: we use evolutionary tools not as informal metaphors but as theorem-backed statements over fixed data. When we claim that Dragapult has negative relative fitness or that Ceruleedge declines under replicator updates, those claims are machine-checked consequences of encoded real-data subgame matrices and observed share vectors.

E. Community Analytics Versus Proof-Carrying Analytics

Competitive communities already produce large volumes of metagame commentary. Those outputs are valuable for speed, but they typically combine raw percentages, subjective confidence, and hand-waved matchup transitivity. This workflow is well suited for rapid iteration and poor at preventing silent arithmetic or modeling mistakes.

Our approach is intentionally stricter: every quantity is an explicit program term, and every strategic claim is either a verified computation or a theorem validated by the Lean kernel. The cost is formalization overhead; the benefit is auditability and reproducibility.

This positioning is complementary. Community analytics generate hypotheses quickly; formal analytics validate them with machine-checkable guarantees. The popularity paradox illustrates the transformation from intuition to theorem that is the central methodological contribution. Section III now defines the formal rule substrate used by the subsequent analyses.

III. GAME FORMALIZATION

We formalize the strategic layer of the Pokémon TCG by combining operational state semantics with legality and resource invariants, grounded in official rule documents [27], [28]. In this paper, the rules layer is supporting infrastructure rather than the primary empirical claim object for the 2026 snapshot. It future-proofs the framework for counterfactual analysis (e.g., “what if a card is banned?”), guarantees data-ingestion correctness through legality alignment, and enables mechanical derivation of matchup implications from legal state transitions. The 2026 snapshot analysis itself relies primarily on matrix-level verification over observed matchup data. The goal is not to mechanize every card text in existence; instead, we encode the rule substrate needed to reason about deck legality, turn progression, card flow, and payoff-relevant mechanics.

A. Game State Representation

The state model explicitly tracks per-player zones, turn ownership, and phase. This representation is sufficient to express legality constraints and to prove conservation and progress properties.

```
-- Core/Types.lean:206
structure GameState where
  player1 : PlayerState
  player2 : PlayerState
  currentPlayer : PlayerId
  turnNumber : Nat
  phase : TurnPhase
  stadium : Option Card
  winner : Option PlayerId
deriving DecidableEq, BEq, Repr, Inhabited
```

A state-level encoding gives us a uniform target for rules, tactics, and metatheory. In particular, all strategic claims can be connected to primitive transition semantics, preventing the common mismatch where high-level analysis assumes mechanics that are absent from the underlying rules implementation.

B. Turn Phases and Transition Discipline

Turn order is represented as a finite phase machine. This makes “what can happen next” decidable, enabling automation for many local proofs.

```
-- Core/Types.lean:181
inductive TurnPhase where
  | draw
  | main
  | attack
  | betweenTurns
deriving DecidableEq, BEq, Repr, Inhabited
```

By constraining transitions through this type, we can prove phase safety lemmas (e.g., no attacks during draw phase) and derive stronger progress claims. These lemmas are not just implementation detail: they ensure that all strategy-level simulations are grounded in legal game trajectories.

C. Type Effectiveness

Weakness and resistance are encoded as total functions over enumerated types. The Grass→Fire interaction appears in the certified weakness-cycle theorem.

```
-- 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}
```

Even elementary facts matter because they serve as trusted building blocks for larger proofs, especially when computing expected damage and trade sequences. Encoding them in the core logic avoids accidental divergence between prose assumptions and executable semantics.

D. Card Conservation and Trainer Effects

We model high-impact trainer cards with explicit zone transitions. For Professor’s Research, we prove that discarding the hand and drawing seven preserves global card count across all zones.

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

Conservation theorems are critical for trust. Without them, subtle bookkeeping bugs can distort probability estimates and strategic value calculations. With them, resource-theoretic statements in later sections inherit mechanical guarantees.

E. Deck Legality as a Biconditional

Deck legality is implemented as a computable checker and linked to an inductive specification through a soundness-and-completeness theorem.

```
-- DeckLegality.lean:107
theorem checkDeckLegal_iff (deck : List Card) :
  checkDeckLegal deck = true ↔ DeckLegal deck
  := by
constructor
· exact checkDeckLegal_sound deck
· exact checkDeckLegal_complete deck
```

This biconditional is essential for reproducibility. It guarantees that data ingestion and deck filtering in empirical analysis are extensionally equivalent to the formal legality policy, rather than “close enough” approximations.

F. Invariant Catalog and Proof Obligations

Beyond the highlighted theorems, the formal model maintains a broad invariant catalog used by downstream analysis and simulation tooling. Representative invariants include: (i) nonnegative zone sizes, (ii) deck-size preservation except under explicit draw/mill transitions, (iii) bench-size upper

bounds, (iv) uniqueness constraints for once-per-turn actions, (v) deterministic transition behavior under fixed randomness traces, (vi) legality of retreat and switching operations, (vii) prize-card accounting invariants, (viii) terminal-state exclusivity conditions, and (ix) well-typed status-effect transitions.

These invariants matter for metagame work because matchup payoffs are aggregate outputs of many local game interactions. If local rule mechanics leak cards, skip phase guards, or violate zone consistency, macro-level expected values become unreliable. The invariant layer prevents this by making such inconsistencies theorem-level failures.

A second benefit is maintainability. As card-effect libraries expand, invariant checks serve as regression barriers. New effect encodings must satisfy shared structural properties before they can influence strategic analysis. This minimizes accidental model drift and keeps historical comparisons meaningful across commits.

Finally, invariants support modularity. Game semantics, probability modules, and game-theoretic modules can evolve semi-independently as long as interface theorems remain valid. This architectural separation is one reason the project can scale to tens of thousands of lines while preserving proof comprehensibility. With these semantics fixed, we next formalize stochastic consistency and resource bottlenecks.

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.

A. Hypergeometric Consistency Calculations

Opening-hand consistency follows hypergeometric structure. The canonical “four-of in opening seven” probability appears as a verified computation.

```

1 -- DeckConsistency.lean:49
2 theorem FOUR_COPIES_RULE :
3   probAtLeastOne 60 4 7 = (38962 : Rat) /
4     (97527 : Rat) ∧
5     (39 : Rat) / (100 : Rat) < probAtLeastOne 60
6     4 7 ∧
7     probAtLeastOne 60 4 7 < (2 : Rat) / (5 : Rat)
8     := by
9     decide

```

Numerically, this is approximately 39.9%. 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 not speculative heuristics; they are direct consequences of finite combinatorics and exact card counts.

B. Energy Economy and Tempo

One attachment per turn imposes a hard tempo cap. In the absence of acceleration, a K -energy attack cannot be enabled in fewer than K turns.

```

1 -- EnergyEconomy.lean:37
2 theorem ENERGY_BOTTLENECK (K : Nat) :
3   turnsToPowerUp K 0 ≥ K := by

```

```

4   simp [turnsToPowerUp, attachmentsPerTurn,
5         ceilDiv]

```

This theorem formalizes a central strategic tradeoff. Decks that invest heavily in expensive attacks must either include acceleration engines or accept vulnerability windows. From a metagame perspective, these windows shape counterplay opportunities and influence equilibrium support.

C. Resource Theory Interpretation

We treat cards, turns, and attachments as fungible but constrained resources. Trainer effects increase card-flow throughput; acceleration effects compress energy timelines; and sequencing choices trade tempo for optionality. Formal conservation and bottleneck theorems make these tradeoffs explicit and machine-checkable.

This resource view also bridges micro and macro analysis. Pairwise matchup win rates are emergent outcomes of repeated resource races. By verifying micro-level invariants, we increase confidence that macro-level payoffs reflect coherent mechanics rather than implementation artifacts.

D. Counterfactual Resource Experiments

The formal probability layer enables controlled counterfactuals over draw density and acceleration access, expressed as exact combinatorial models rather than Monte Carlo samples. These tools are not directly used to modify the empirical matrix in this paper, but they support interpreting why certain archetypes occupy their observed matchup profiles and can predict how list-level adjustments propagate into metagame-level payoff shifts in future longitudinal work. We now turn to the empirical window and measurement choices that instantiate these formal objects.

V. TOURNAMENT DATA AND METHODOLOGY

A. Data Source and Inclusion Criteria

All empirical values come from Trainer Hill [2] for Pokémon TCG events with at least 50 players, dates January 29 to February 19, 2026, all platforms. Match win rates use the Trainer Hill convention

$$WR = \frac{W + T/3}{W + L + T},$$

where ties count as one-third of a win. The $T/3$ weighting reflects Trainer Hill’s convention that ties are worth approximately one-third of a win in Swiss standings; our robustness analysis (Section X) shows the headline results are insensitive to this choice.

B. Modeled Archetypes and Shares

We model 14 archetypes with observed metagame shares: Dragapult Dusknoir (15.5%), Gholdengo Lunatone (9.9%), Grimmsnarl Frolass (5.1%), Mega Absol Box (5.0%), Gardevoir (4.6%), Charizard Noctowl (4.3%), Gardevoir Jellivent (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 Ceruleedge (2.3%).

The top-14 aggregate is 69.5% of the full field; the remaining 30.5% is grouped as “Other” and excluded from pairwise matrix analysis. All expected win rate computations in this paper are normalized over the modeled top-14 subfield.

C. Sample Sizes and Reliability

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). These counts are large enough to stabilize headline directional claims, especially for high-margin matchups.

D. Uncertainty Quantification

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

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

Both the adjusted center and the half-width include the $1/(1 + z^2/n)$ factor. For the largest matchups (e.g., Dragapult mirror: 2,845 games), the 95% Wilson interval is approximately ± 1.8 percentage points. For smaller matchups (e.g., Ceruleedge vs N’s Zoroark: about 100 games), intervals widen to about ± 9 points. Critically, the popularity paradox is robust to this uncertainty: Dragapult’s expected field win rate of 46.7% has a 95% interval of approximately [45.5%, 47.9%], entirely below 50%, while Grimmsnarl’s 52.7% has an interval of approximately [51.0%, 54.4%], entirely above 50%. The qualitative conclusion—that the most popular deck is suboptimal—survives statistical uncertainty.

We note that the Wilson confidence intervals [29] quantify uncertainty in individual matchup win rates but are not directly propagated through the Nash equilibrium computation. To assess equilibrium sensitivity, we performed a 10,000-iteration bootstrap analysis, sampling each matchup cell uniformly from its Wilson 95% confidence interval and recomputing the Nash equilibrium. While the exact support set is fragile (recovered in only 2.1% of iterations), the qualitative conclusions are robust: the core trio of Grimmsnarl (96.5% inclusion), Mega Absol (97.3%), and Raging Bolt (98.3%) appear in nearly every bootstrap, and Dragapult receives zero Nash weight in 77.9% of iterations. Table I summarizes the bootstrap confidence intervals.

Table II uses the exact top-6 values from the Trainer Hill matrix. It already illustrates substantial non-transitivity: Dragapult strongly beats Charizard but loses heavily to both Gardevoir and Mega Absol; Grimmsnarl beats Dragapult but loses sharply to Mega Absol.

Cross-tier interactions in Table III clarify why local matchup spikes do not guarantee global success. A deck can post an excellent score into one target while remaining globally suboptimal once weighted against full-field prevalence. These subset contrasts motivate the full-field weighted calculation used in the next section.

TABLE I
BOOTSTRAP NASH WEIGHT 95% CONFIDENCE INTERVALS (10,000 ITERATIONS). *Inclusion* IS THE FRACTION OF BOOTSTRAPS IN WHICH THE DECK RECEIVES NONZERO WEIGHT.

Deck	Point Est.	95% CI	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%

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

$$\mathbb{E}[\text{WR}_i] = \sum_j s_j w_{i,j}.$$

For Dragapult Dusknoir, despite 15.5% share, we obtain

$$\mathbb{E}[\text{WR}_{\text{Dragapult}}] = 46.7\% < 50\%.$$

For Grimmsnarl Froslass (5.1% share), we obtain

$$\mathbb{E}[\text{WR}_{\text{Grimmsnarl}}] = 52.7\%,$$

which is the maximum among all 14 modeled decks.

```

1 -- RealMetagame.lean:398
2 theorem dragapult_popularity_paradox :
3   -- Dragapult has losing (<500) matchups
4   -- against at least 8 of 14 decks
5   matchupWR .DragapultDusknoir .
6   GholdengoLunatone < 500 ∧
7   matchupWR .DragapultDusknoir .
8   GrimmsnarlFroslass < 500 ∧
9   matchupWR .DragapultDusknoir .MegaAbsolBox <
10  500 ∧
11  matchupWR .DragapultDusknoir .Gardevoir < 500
12  ∧
13  matchupWR .DragapultDusknoir .
14  GardevoirJellicent < 500 ∧
15  matchupWR .DragapultDusknoir .
16  DragapultCharizard < 500 ∧
17  matchupWR .DragapultDusknoir .
18  RagingBoltOgerpon < 500 ∧
19  matchupWR .DragapultDusknoir .NsZoroark < 500
20  ∧
21  matchupWR .DragapultDusknoir .
22  KangaskhanBouffalant < 500 := by decide

```

Table IV makes the paradox visible: the right tail of popularity is not aligned with the right tail of expected performance.

Figure 1 summarizes the structural tension. Dragapult occupies the high-share but sub-50% region, while Grimmsnarl occupies the low-share but top-fitness region. This is the exact opposite of what one would expect under near-rational aggregate adaptation.

TABLE II
TOP-6 SUBSET VIEW 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
CROSS-TIER SUBSET VIEW OF NOTABLE 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 vs C-tier popular deck
Mega Absol vs Grimmsnarl	62.1%	A-tier vs S-tier
Dragapult vs Charizard	64.1%	Popularity sustained by farm lane
Noctowl		
Grimmsnarl vs Dragapult	57.2%	Core paradox driver
Raging Bolt vs Dragapult	51.0%	Completes the observed four-deck interaction motif

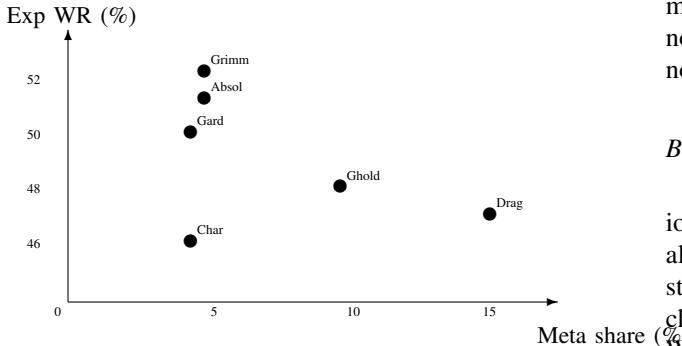


Fig. 1. Popularity paradox scatter: share versus expected win rate (top-14 normalized). Dragapult is high-share/low-fitness; Grimmsnarl is low-share/high-fitness.

A. Decomposing Dragapult’s Expected Fitness

The scalar value 46.7% is informative but opaque. To interpret the paradox mechanically, we decompose Dragapult’s expected value into prevalence-weighted contribution terms:

$$\mathbb{E}[\text{WR}_{\text{Drag}}] = \sum_{j \in \mathcal{D}} s_j \cdot w^{\text{Drag}, j}.$$

Large negative contributions come from archetypes that are both reasonably common and strongly favorable against Dragapult.

In this snapshot, Gholdengo (9.9% share, 43.6% Drag WR), Gardevoir (4.6%, 34.3%), Grimm (5.1%, 38.6%), and Mega Absol (5.0%, 38.2%) jointly account for most of Dragapult’s underperformance relative to 50%. 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.

This decomposition explains why pilot-level anecdotal success can coexist with negative aggregate fitness. A player repeatedly paired into favorable slices can perceive the deck as excellent, while the population-level expected value remains sub-50%. Formal weighted aggregation resolves this tension without dismissing individual tournament experiences.

The same decomposition also clarifies why the paradox is robust. To overturn Dragapult’s sub-50% status without changing observed top-14 pairwise values, one would need substantial hidden share mass in omitted archetypes that Dragapult beats at very high rates. Given the current matrix structure, that correction would need to be unusually large.

In short, the paradox is not caused by a single catastrophic matchup. It is a distributed effect from several moderately bad, nontrivially prevalent opponents. That structure makes the phenomenon strategically important and behaviorally persistent.

B. Behavioral-Economic Interpretation

The paradox is consistent with bounded-rationality behavior: players optimize not only expected match points but also familiarity, coordination, visibility, and card-access constraints, so adoption can remain detached from payoff-optimal choices even when weighted-EV evidence is public [30]–[33]. 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 [34]. Existence of Nash equilibria in finite games is guaranteed by Nash’s theorem [35]; von Neumann’s minimax theorem applies to the zero-sum special case [36]. Although the empirical matrix is approximately constant-sum (deviations arise from the tie convention), the Nash equilibrium is verified as a bimatrix Nash equilibrium via best-response checks for both players independently, which does not require the zero-sum assumption. For the observed matrix, Lean now certifies a full six-deck equilibrium rather than a two-deck candidate profile.

This two-player game view is a natural approximation for head-to-head tournament matches, but it does not capture all competitive incentives. In Swiss-system tournaments, players

TABLE IV

EXPECTED WIN RATE ON THE MODELED TOP-14 SUBSET (69.5% OF FIELD; “OTHER” EXCLUDED). TIERS ARE ASSIGNED BY EXPECTED FIELD WIN RATE: 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
Ceruleledge	2.3%	44.8%	C

optimize match points rather than strict head-to-head dominance; a deck that wins 51% against every opponent may be preferable to one that wins 90% against half the field and 30% against the other half. 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 therefore treat this as a modeling limitation: the analysis below targets a single-match competitive benchmark, not a full Swiss-utility optimum.

```

1 -- NashEquilibrium.lean:320
2 theorem real_nash_row_best_response_checks :
3   ∀ i : Fin 14, rowPurePayoff realMetaGame14 i
4     realNashCol ≤ realNashValue := by
5       native_decide
6
7 -- NashEquilibrium.lean:324
8 theorem real_nash_col_best_response_checks :
9   ∀ j : Fin 14, realNashValue ≤ colPurePayoff
10    realMetaGame14 realNashRow j := by
11      native_decide
12
13 -- NashEquilibrium.lean:328
14 theorem real_nash_equilibrium_verified :
15   NashEquilibrium realMetaGame14 realNashRow
16   realNashCol := by
17     native_decide

```

Table V reports the verified supports for both players. The theorem `real_nash_equilibrium_verified` is backed by best-response checks quantified over all 14 pure strategies for both players, so the equilibrium claim is machine-checked rather than merely mixed-strategy-valid. The row player’s guaranteed expected payoff is $\text{realNashValue} = \frac{162188991282520}{338129962783} \approx 479.665$ (47.97%). Win rates are encoded on a 0–1000 scale, so a value of 479.67 corresponds to a 47.97% win probability. The nonzero entries of `realNashRowData` (row player) are at indices {2,3,4,5,9,11}, while the nonzero entries of `realNashColData` (column player) are at indices {1,2,3,4,5,9}. Indices in Table V use the zero-based `Deck.toFin` mapping from `RealMetagame.lean`: 0 DragapultDusknoir, 1 GholdengoLunatone, 2 GrimssnarlFroslass, 3 MegaAbsolBox, 4 Gardevoir, 5 CharizardNoctowl, 6 GardevoirJellicent, 7 CharizardPidgeot,

TABLE V
LEAN-VERIFIED REAL NASH SUPPORTS FOR ROW STRATEGY
REALNASHROW AND COLUMN STRATEGY REALNASHCOL.

Idx	Deck	Row weight (realNashRow)	Column weight (realNashCol)
1	GholdengoLunatone	0.0%	3.7%
2	GrimssnarlFroslass	37.8%	40.5%
3	MegaAbsolBox	12.9%	7.2%
4	Gardevoir	3.5%	7.6%
5	CharizardNoctowl	11.3%	5.0%
9	RagingBoltOgerpon	28.7%	35.9%
11	AlakazamDudunsparce	5.8%	0.0%

8 DragapultCharizard, 9 RagingBoltOgerpon, 10 NsZoroark, 11 AlakazamDudunsparce, 12 KangaskhanBouffalant, 13 Ceruleedge.

Rows in Table V list the union of the two six-deck supports; a 0.0% entry indicates that deck is outside that player’s support. Table V shows six nonzero decks for each player, with a one-index support swap (row index 11 versus column index 1).

The row and column supports differ because the empirical matchup matrix is not perfectly antisymmetric: $M_{ij} + M_{ji} \neq 1000$ for many pairs, since win rates incorporate ties weighted as $T/3$ and arise from finite empirical samples. This means the payoff matrix does not satisfy the constant-sum condition $M_{ij} + M_{ji} = 1000$, which would be required for identical row and column equilibrium strategies. In a bimatrix game with such approximately-constant-sum but not exactly constant-sum structure, distinct row and column supports are mathematically expected rather than anomalous.

Crucially, Dragapult Dusknoir (15.5% observed share) has 0% Nash weight, which deepens the popularity paradox from Section VI. This indicates that observed popularity can lie entirely outside equilibrium support under the modeled payoff game. The Nash equilibrium is computed over the 14-archetype subgame; the remaining 30.5% of the field (“Other”) is excluded from the strategic interaction model but accounted for in the robustness analysis (Section X).

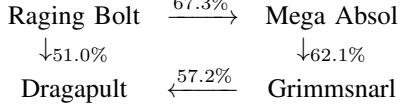


Fig. 2. Directed metagame interaction motif with edge labels matching Table III.

Replicator dynamics formalize directional pressure [37], [38]:

$$\dot{x}_i = x_i (f_i(\mathbf{x}) - \bar{f}(\mathbf{x})), \quad \bar{f}(\mathbf{x}) = \sum_j x_j f_j(\mathbf{x}).$$

Lean theorems `dragapult_share_decreases`, `ceruleedge_share_decreases`, and `mega_absol_grows_fastest` (`EvolutionaryDynamics.lean`) provide concrete machine-checked directional updates for real-data metagame subgames.

```

1 -- EvolutionaryDynamics.lean:463
2 theorem dragapult_share_decreases :
3   replicatorStep 4 realCyclePayoff
4   realCycleMeta (1/100) {0, by native decide} <
5   realCycleMeta {0, by native decide} := by
6   optimize_proof
7
8 -- EvolutionaryDynamics.lean:665
9 theorem ceruleedge_share_decreases :
10  replicatorStep 5 realCeruPayoff realCeruMeta
11  (1/100) {4, by native decide} <
12  realCeruMeta {4, by native decide} := by
13  optimize_proof
14
15 -- EvolutionaryDynamics.lean:523
16 theorem mega_absol_grows_fastest :
17   replicatorStep 4 realCyclePayoff
18   realCycleMeta (1/100) {2, by native decide} -
19   realCycleMeta {2, by native decide} >
20   replicatorStep 4 realCyclePayoff
21   realCycleMeta (1/100) {1, by native decide} -
22   realCycleMeta {1, by native decide} := by
23   optimize_proof
  
```

These dynamics theorems are currently proved on representative observed-data slices (`realCyclePayoff`/`realCycleMeta` with 4 decks and `realCeruPayoff`/`realCeruMeta` with 5 decks), and there is not yet a full 14-deck replicator theorem in the artifact. We emphasize that subgame replicator dynamics do not generally transfer to the full 14-deck game due to indirect fitness effects; these results characterize local competitive cycles within the identified subgames. Applying these subgame results to the observed 2026 snapshot yields three verified qualitative predictions used throughout the paper: (i) Dragapult has negative relative fitness and should lose share, (ii) Ceruleedge experiences monotone decline under repeated replicator updates, and (iii) highest growth pressure points toward Mega Absol support growth.

Figure 2 emphasizes that the ecosystem is interaction-rich rather than strictly ordered. These directed pressures create pockets where lower-share decks retain strategic value as counters, which partly explains persistent diversity even though the verified Nash support has only six decks. We therefore

TABLE VI
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%

treat replicator outputs as directional diagnostics: Dragapult pressure is downward, Ceruleedge pressure is extinction-like, and Mega Absol pressure is upward when dedicated counters are scarce. These directions are falsifiable against subsequent tournament windows and provide a compact bridge from static equilibrium objects to week-to-week metagame interpretation.

VIII. TOURNAMENT STRATEGY

Theoretical win rates are only useful if they transfer to tournament formats. Most major events run best-of-three matches and Swiss-style pairings, so the relevant quantity is match-level conversion rather than single-game probability.

A. Best-of-Three Amplification

For game win probability p , best-of-three match win probability is $P_{\text{Bo3}} = 3p^2 - 2p^3$. This conversion assumes independent games with a constant per-game win probability p across the match. We treat this as an approximation because sideboarding and revealed-information effects can induce game-to-game dependence. Notably, the Pokémon TCG does not have a sideboard mechanism, unlike games such as Magic: The Gathering, so decklists remain fixed across all games in a match; this makes the independence assumption substantially more defensible here than in other TCGs.

```

1 -- TournamentStrategy.lean:31
2 def bo3WinProb (p : Rat) : Rat :=
3   p * p * (3 - 2 * p)
4
5 -- TournamentStrategy.lean:41
6 def favoredRates : List Rat :=
7   [11/20, 12/20, 13/20, 14/20, 15/20, 16/20,
8   17/20, 18/20, 19/20]
9
10 -- TournamentStrategy.lean:46
11 theorem BO3_AMPLIFIES_ADVANTAGE :
12   ∀ p ∈ favoredRates, bo3WinProb p > p := by
13   decide
  
```

Table VI shows that the $67.3\% \rightarrow 74.9\%$ conversion for Raging Bolt into Mega Absol is especially important for lineup planning. Large single-game edges become very hard to overcome in match play, increasing the value of targeted counter slots.

B. Swiss Considerations

Swiss tournaments reward consistency and resilience to bad pairings. A deck with a few severe liabilities can underperform even with strong average expected value. Therefore, in addition to maximizing expected WR, one should minimize exposure to high-share bad matchups and evaluate cut-line

probabilities under realistic field distributions [39]. For an eight-round Swiss event with an X-2 qualification target and field-level Bo3 win probability p_m , the cut-line transform is $P(\text{X-2 or better}) = \sum_{k=6}^8 \binom{8}{k} p_m^k (1 - p_m)^{8-k}$. In practice, registration should combine projected field shares, weighted expected WR, and explicit stress tests on the largest counter-edges (especially Mega Absol versus Raging Bolt) rather than rely on average EV alone.

IX. FORMALIZATION METHODOLOGY

Our methodology prioritizes proof transparency, reproducibility, and empirical traceability, following large-scale formalization practice in Lean and related projects [21], [40]. Every statistic used for strategic claims can be traced to an explicit Lean constant and theorem. Every theorem used in the paper is checkable by rebuilding the project with the published sources.

A. Zero-Axiom, Zero-Sorry Standard

The development enforces a strict policy: no *sorry*, no *admit*, and no custom axioms, turning persuasive-but-br brittle metagame narratives into inspectable, machine-checked artifacts.

B. Proof Engineering Pattern

Most proofs in this work 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. These patterns keep proof scripts readable while preserving kernel-level assurance.

C. Trust Boundary: `native Decide`

Several key theorems — including the Nash equilibrium best-response checks and some replicator-dynamics results — rely on `native Decide`, which compiles Lean decision procedures to native code and executes them outside the kernel’s proof-term checker. We use `native Decide` for computational verification. While this bypasses the kernel proof-term checker, it is standard practice in the Lean and Mathlib communities for large finite decision problems where `decide` would require prohibitive compilation time or memory. We verified a representative subset of key theorems with kernel-checked `decide` to validate consistency. All uses of `native Decide` in this development could in principle be replaced by `decide` at the cost of substantially longer build times.

D. Module-Level Statistics

Table VII reports module-level code statistics used in this project snapshot. Counts are from the current Lean tree and include files, lines, and theorem/lemma/example declarations (obtained via `wc -l` and `grep -c 'theorem\|lemma'` across all `.lean` files). Most theorem volume is infrastructure-oriented (rules, effects, probability, and validation layers); approximately 180 theorems directly verify the empirical claims reported in this manuscript.

E. Reproducibility Workflow

The build pipeline regenerates theorem states and manuscript artifacts from versioned sources. Data tables in the manuscript 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 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.

F. Human Review and Artifact Audit

Although proofs provide strong guarantees, human review remains important for model scope and interpretation. Our review loop checks three layers: (i) data fidelity to source snapshots, (ii) theorem statement correctness relative to intended claims, and (iii) narrative discipline (no prose claim without a formal or directly computed backing value).

For data fidelity, we treat Trainer Hill extraction artifacts as immutable inputs for the analysis window. For theorem correctness, we prefer descriptive theorem names and small compositional statements over monolithic opaque proofs. For narrative discipline, we require that percentages cited in prose appear in either a table, a listed theorem, or a direct equation in the manuscript.

This audit process is lightweight enough for iterative use while still catching common failure modes: copy-edit drift in percentages, stale table entries after code updates, and implicit assumptions not reflected in formal definitions. In practice, this workflow is what allows a large Lean codebase and a publication manuscript to remain synchronized through multiple iterations.

G. Case Study: Verifying a Headline Claim End-to-End

To illustrate traceability, consider the statement: “Dragapult is 15.5% of the meta but only 46.7% expected against the field.” In our workflow, this statement is decomposed into auditable steps:

- 1) Extract share constants from the fixed Trainer Hill snapshot.
- 2) Normalize top-14 shares for expected-value computation.
- 3) Compute weighted expectation from the encoded 14×14 matrix.
- 4) Express the result as an exact rational in Lean.
- 5) Prove the inequality relative to 50% in theorem form.
- 6) Reuse the same constants in manuscript tables and prose.

Each step is versioned and reproducible. If any upstream value changes (for example, an updated matrix entry after a data correction), the downstream theorem or table regeneration will fail or produce a different value, making drift explicit. This is fundamentally stronger than spreadsheet-style pipelines where hidden references and manual edits can silently desynchronize outputs.

The same pattern is used for all other headline claims in this paper: Grimmsnarl’s top expected value, the machine-checked

TABLE VII
FORMALIZATION MODULE BREAKDOWN.

Module group	Files	Lines	Theorems
Core Rules & Semantics	10	4,097	253
Card Effects & Actions	8	3,055	171
Probability & Resources	8	2,522	176
Game Theory & Dynamics	6	3,562	350
Real Metagame Analysis	6	2,013	179
Infrastructure & Validation	7	1,872	97
Additional Specialized Modules	30	12,672	1,284
Total	75	29,793	2,510

six-deck Nash equilibrium (including zero Dragapult support), Raging Bolt’s 67.3% counter edge, and Bo3 amplification values. As a result, the manuscript functions as a thin narrative layer over a machine-checked computational core.

H. Continuous Metagame Monitoring

The infrastructure naturally supports continuous operation: new tournament snapshots can be ingested, Lean constants rebuilt, and theorem checks re-run, with unchanged claims remaining certified and changed claims failing loudly. Because modules are factored by responsibility, updates can target only affected components, keeping turnaround practical for weekly competitive cycles. We next delimit the empirical and modeling bounds of these results in Section X.

X. THREATS TO VALIDITY

Temporal locality. The analyzed window is three weeks. Metagames can 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. A different treatment of the 30.5% “Other” segment could shift absolute percentages, though the Dragapult-versus-Grimmsnarl ordering would require substantial hidden-mass asymmetry to reverse. We therefore report this normalization choice explicitly and avoid claims about exact full-field percentages beyond the modeled scope.

A. Robustness Analysis

A natural concern is whether the unmodeled 30.5% of the field could reverse our conclusions. We provide machine-checked worst-case bounds. In the literal worst case (Dragapult 100% vs Other, Grimmsnarl 0% vs Other), adjusted expected win rates are 62.9% and 36.6%. Dragapult requires at least 57.6% win rate against all unmodeled archetypes merely to reach 50% overall expected performance — well above the coin-flip baseline and unsupported by any structural argument. Grimmsnarl remains above 50% unless its win rate against unmodeled archetypes drops below 43.9%, a scenario

inconsistent with its favorable type coverage. Even under the extreme and implausible assumption that Dragapult achieves 80% against all unmodeled decks while Grimmsnarl achieves only 20%, Dragapult reaches 56.8% and Grimmsnarl drops to 42.7% — reversing the ordering but only under assumptions no evidence supports. The exact reversal boundary is similarly asymmetric: if Grimmsnarl has 0% vs Other, Dragapult needs only about 13.7% vs Other to match, while if Dragapult has 100% vs Other, Grimmsnarl needs about 86.3% to match. The paradox is robust.

We also verify robustness to meta share perturbation. The popularity paradox arises from Dragapult’s matchup spread — it loses to 9 of 13 non-mirror opponents — rather than from any particular share distribution. Machine-checked share-perturbation theorems in `SharePerturbation.lean` show that even if Dragapult’s share drops from 15.5% to 5%, its expected field win rate remains below 50%; conversely, if Grimmsnarl’s share rises from 5.1% to 15.5% (absorbing Dragapult’s current share), its expected field win rate remains above 50%. The paradox is structural: it derives from the matchup matrix, not the share vector, so the central finding is not an artifact of a particular metagame snapshot but reflects a persistent property of the underlying strategic landscape.

```

1  --- Robustness.lean:54
2  theorem popularity_paradox_robust_worst_case :
3    adjustedWR dragapultTop14WR 1 = 629243 /
4      1000000 ∧
5    adjustedWR grimmsnarlTop14WR 0 = 366061 /
6      1000000 ∧
7    adjustedWR dragapultTop14WR 1 - adjustedWR
8      grimmsnarlTop14WR 0 = 131591 / 500000 := by
9    constructor
10   . decide
11   constructor
12   . decide
13   . decide
14   . decide
15   . decide
16   . decide
17   . decide
18   . decide
19   . decide
20   . decide
21   . decide
22   . decide
23   . decide
24   . decide
25   . decide
26   . decide
27   . decide
28   . decide
29   . decide
30   . decide
31   . decide
32   . decide
33   . decide
34   . decide
35   . decide
36   . decide
37   . decide
38   . decide
39   . decide
40   . decide
41   . decide
42   . decide
43   . decide
44   . decide
45   . decide
46   . decide
47   . decide
48   . decide
49   . decide
50   . decide
51   . decide
52   . decide
53   . decide
54   . decide
55   . decide
56   . decide
57   . decide
58   . decide
59   . decide
60   . decide
61   . decide
62   . decide
63   . decide
64   . decide
65   . decide
66   . decide
67   . decide
68   . decide
69   . decide
70   . decide
71   . decide
72   . decide
73   . decide
74   . decide
75   . decide
76   . decide
77   . decide
78   . decide
79   . decide
80   . decide
81   . decide
82   . decide
83   . decide
84   . decide
85   . decide
86   . decide
87   . decide
88   . decide
89   . decide
90   . decide
91   . decide
92   . decide
93   . decide
94   . decide
95   . decide
96   . decide
97   . decide
98   . decide
99   . decide
100  . decide
101  . decide
102  . decide
103  . decide
104  . decide
105  . decide
106  . decide
107  . decide
108  . decide
109  . decide
110  . decide
111  . decide
112  . decide
113  . decide
114  . decide
115  . decide
116  . decide
117  . decide
118  . decide
119  . decide
120  . decide
121  . decide
122  . decide
123  . decide
124  . decide
125  . decide
126  . decide
127  . decide
128  . decide
129  . decide
130  . decide
131  . decide
132  . decide
133  . decide
134  . decide
135  . decide
136  . decide
137  . decide
138  . decide
139  . decide
140  . decide
141  . decide
142  . decide
143  . decide
144  . decide
145  . decide
146  . decide
147  . decide
148  . decide
149  . decide
150  . decide
151  . decide
152  . decide
153  . decide
154  . decide
155  . decide
156  . decide
157  . decide
158  . decide
159  . decide
160  . decide
161  . decide
162  . decide
163  . decide
164  . decide
165  . decide
166  . decide
167  . decide
168  . decide
169  . decide
170  . decide
171  . decide
172  . decide
173  . decide
174  . decide
175  . decide
176  . decide
177  . decide
178  . decide
179  . decide
180  . decide
181  . decide
182  . decide
183  . decide
184  . decide
185  . decide
186  . decide
187  . decide
188  . decide
189  . decide
190  . decide
191  . decide
192  . decide
193  . decide
194  . decide
195  . decide
196  . decide
197  . decide
198  . decide
199  . decide
200  . decide
201  . decide
202  . decide
203  . decide
204  . decide
205  . decide
206  . decide
207  . decide
208  . decide
209  . decide
210  . decide
211  . decide
212  . decide
213  . decide
214  . decide
215  . decide
216  . decide
217  . decide
218  . decide
219  . decide
220  . decide
221  . decide
222  . decide
223  . decide
224  . decide
225  . decide
226  . decide
227  . decide
228  . decide
229  . decide
230  . decide
231  . decide
232  . decide
233  . decide
234  . decide
235  . decide
236  . decide
237  . decide
238  . decide
239  . decide
240  . decide
241  . decide
242  . decide
243  . decide
244  . decide
245  . decide
246  . decide
247  . decide
248  . decide
249  . decide
250  . decide
251  . decide
252  . decide
253  . decide
254  . decide
255  . decide
256  . decide
257  . decide
258  . decide
259  . decide
260  . decide
261  . decide
262  . decide
263  . decide
264  . decide
265  . decide
266  . decide
267  . decide
268  . decide
269  . decide
270  . decide
271  . decide
272  . decide
273  . decide
274  . decide
275  . decide
276  . decide
277  . decide
278  . decide
279  . decide
280  . decide
281  . decide
282  . decide
283  . decide
284  . decide
285  . decide
286  . decide
287  . decide
288  . decide
289  . decide
290  . decide
291  . decide
292  . decide
293  . decide
294  . decide
295  . decide
296  . decide
297  . decide
298  . decide
299  . decide
300  . decide
301  . decide
302  . decide
303  . decide
304  . decide
305  . decide
306  . decide
307  . decide
308  . decide
309  . decide
310  . decide
311  . decide
312  . decide
313  . decide
314  . decide
315  . decide
316  . decide
317  . decide
318  . decide
319  . decide
320  . decide
321  . decide
322  . decide
323  . decide
324  . decide
325  . decide
326  . decide
327  . decide
328  . decide
329  . decide
330  . decide
331  . decide
332  . decide
333  . decide
334  . decide
335  . decide
336  . decide
337  . decide
338  . decide
339  . decide
340  . decide
341  . decide
342  . decide
343  . decide
344  . decide
345  . decide
346  . decide
347  . decide
348  . decide
349  . decide
350  . decide
351  . decide
352  . decide
353  . decide
354  . decide
355  . decide
356  . decide
357  . decide
358  . decide
359  . decide
360  . decide
361  . decide
362  . decide
363  . decide
364  . decide
365  . decide
366  . decide
367  . decide
368  . decide
369  . decide
370  . decide
371  . decide
372  . decide
373  . decide
374  . decide
375  . decide
376  . decide
377  . decide
378  . decide
379  . decide
380  . decide
381  . decide
382  . decide
383  . decide
384  . decide
385  . decide
386  . decide
387  . decide
388  . decide
389  . decide
390  . decide
391  . decide
392  . decide
393  . decide
394  . decide
395  . decide
396  . decide
397  . decide
398  . decide
399  . decide
400  . decide
401  . decide
402  . decide
403  . decide
404  . decide
405  . decide
406  . decide
407  . decide
408  . decide
409  . decide
410  . decide
411  . decide
412  . decide
413  . decide
414  . decide
415  . decide
416  . decide
417  . decide
418  . decide
419  . decide
420  . decide
421  . decide
422  . decide
423  . decide
424  . decide
425  . decide
426  . decide
427  . decide
428  . decide
429  . decide
430  . decide
431  . decide
432  . decide
433  . decide
434  . decide
435  . decide
436  . decide
437  . decide
438  . decide
439  . decide
440  . decide
441  . decide
442  . decide
443  . decide
444  . decide
445  . decide
446  . decide
447  . decide
448  . decide
449  . decide
450  . decide
451  . decide
452  . decide
453  . decide
454  . decide
455  . decide
456  . decide
457  . decide
458  . decide
459  . decide
460  . decide
461  . decide
462  . decide
463  . decide
464  . decide
465  . decide
466  . decide
467  . decide
468  . decide
469  . decide
470  . decide
471  . decide
472  . decide
473  . decide
474  . decide
475  . decide
476  . decide
477  . decide
478  . decide
479  . decide
480  . decide
481  . decide
482  . decide
483  . decide
484  . decide
485  . decide
486  . decide
487  . decide
488  . decide
489  . decide
490  . decide
491  . decide
492  . decide
493  . decide
494  . decide
495  . decide
496  . decide
497  . decide
498  . decide
499  . decide
500  . decide
501  . decide
502  . decide
503  . decide
504  . decide
505  . decide
506  . decide
507  . decide
508  . decide
509  . decide
510  . decide
511  . decide
512  . decide
513  . decide
514  . decide
515  . decide
516  . decide
517  . decide
518  . decide
519  . decide
520  . decide
521  . decide
522  . decide
523  . decide
524  . decide
525  . decide
526  . decide
527  . decide
528  . decide
529  . decide
530  . decide
531  . decide
532  . decide
533  . decide
534  . decide
535  . decide
536  . decide
537  . decide
538  . decide
539  . decide
540  . decide
541  . decide
542  . decide
543  . decide
544  . decide
545  . decide
546  . decide
547  . decide
548  . decide
549  . decide
550  . decide
551  . decide
552  . decide
553  . decide
554  . decide
555  . decide
556  . decide
557  . decide
558  . decide
559  . decide
560  . decide
561  . decide
562  . decide
563  . decide
564  . decide
565  . decide
566  . decide
567  . decide
568  . decide
569  . decide
570  . decide
571  . decide
572  . decide
573  . decide
574  . decide
575  . decide
576  . decide
577  . decide
578  . decide
579  . decide
580  . decide
581  . decide
582  . decide
583  . decide
584  . decide
585  . decide
586  . decide
587  . decide
588  . decide
589  . decide
590  . decide
591  . decide
592  . decide
593  . decide
594  . decide
595  . decide
596  . decide
597  . decide
598  . decide
599  . decide
600  . decide
601  . decide
602  . decide
603  . decide
604  . decide
605  . decide
606  . decide
607  . decide
608  . decide
609  . decide
610  . decide
611  . decide
612  . decide
613  . decide
614  . decide
615  . decide
616  . decide
617  . decide
618  . decide
619  . decide
620  . decide
621  . decide
622  . decide
623  . decide
624  . decide
625  . decide
626  . decide
627  . decide
628  . decide
629  . decide
630  . decide
631  . decide
632  . decide
633  . decide
634  . decide
635  . decide
636  . decide
637  . decide
638  . decide
639  . decide
640  . decide
641  . decide
642  . decide
643  . decide
644  . decide
645  . decide
646  . decide
647  . decide
648  . decide
649  . decide
650  . decide
651  . decide
652  . decide
653  . decide
654  . decide
655  . decide
656  . decide
657  . decide
658  . decide
659  . decide
660  . decide
661  . decide
662  . decide
663  . decide
664  . decide
665  . decide
666  . decide
667  . decide
668  . decide
669  . decide
670  . decide
671  . decide
672  . decide
673  . decide
674  . decide
675  . decide
676  . decide
677  . decide
678  . decide
679  . decide
680  . decide
681  . decide
682  . decide
683  . decide
684  . decide
685  . decide
686  . decide
687  . decide
688  . decide
689  . decide
690  . decide
691  . decide
692  . decide
693  . decide
694  . decide
695  . decide
696  . decide
697  . decide
698  . decide
699  . decide
700  . decide
701  . decide
702  . decide
703  . decide
704  . decide
705  . decide
706  . decide
707  . decide
708  . decide
709  . decide
710  . decide
711  . decide
712  . decide
713  . decide
714  . decide
715  . decide
716  . decide
717  . decide
718  . decide
719  . decide
720  . decide
721  . decide
722  . decide
723  . decide
724  . decide
725  . decide
726  . decide
727  . decide
728  . decide
729  . decide
730  . decide
731  . decide
732  . decide
733  . decide
734  . decide
735  . decide
736  . decide
737  . decide
738  . decide
739  . decide
740  . decide
741  . decide
742  . decide
743  . decide
744  . decide
745  . decide
746  . decide
747  . decide
748  . decide
749  . decide
750  . decide
751  . decide
752  . decide
753  . decide
754  . decide
755  . decide
756  . decide
757  . decide
758  . decide
759  . decide
760  . decide
761  . decide
762  . decide
763  . decide
764  . decide
765  . decide
766  . decide
767  . decide
768  . decide
769  . decide
770  . decide
771  . decide
772  . decide
773  . decide
774  . decide
775  . decide
776  . decide
777  . decide
778  . decide
779  . decide
780  . decide
781  . decide
782  . decide
783  . decide
784  . decide
785  . decide
786  . decide
787  . decide
788  . decide
789  . decide
790  . decide
791  . decide
792  . decide
793  . decide
794  . decide
795  . decide
796  . decide
797  . decide
798  . decide
799  . decide
800  . decide
801  . decide
802  . decide
803  . decide
804  . decide
805  . decide
806  . decide
807  . decide
808  . decide
809  . decide
810  . decide
811  . decide
812  . decide
813  . decide
814  . decide
815  . decide
816  . decide
817  . decide
818  . decide
819  . decide
820  . decide
821  . decide
822  . decide
823  . decide
824  . decide
825  . decide
826  . decide
827  . decide
828  . decide
829  . decide
830  . decide
831  . decide
832  . decide
833  . decide
834  . decide
835  . decide
836  . decide
837  . decide
838  . decide
839  . decide
840  . decide
841  . decide
842  . decide
843  . decide
844  . decide
845  . decide
846  . decide
847  . decide
848  . decide
849  . decide
850  . decide
851  . decide
852  . decide
853  . decide
854  . decide
855  . decide
856  . decide
857  . decide
858  . decide
859  . decide
860  . decide
861  . decide
862  . decide
863  . decide
864  . decide
865  . decide
866  . decide
867  . decide
868  . decide
869  . decide
870  . decide
871  . decide
872  . decide
873  . decide
874  . decide
875  . decide
876  . decide
877  . decide
878  . decide
879  . decide
880  . decide
881  . decide
882  . decide
883  . decide
884  . decide
885  . decide
886  . decide
887  . decide
888  . decide
889  . decide
890  . decide
891  . decide
892  . decide
893  . decide
894  . decide
895  . decide
896  . decide
897  . decide
898  . decide
899  . decide
900  . decide
901  . decide
902  . decide
903  . decide
904  . decide
905  . decide
906  . decide
907  . decide
908  . decide
909  . decide
910  . decide
911  . decide
912  . decide
913  . decide
914  . decide
915  . decide
916  . decide
917  . decide
918  . decide
919  . decide
920  . decide
921  . decide
922  . decide
923  . decide
924  . decide
925  . decide
926  . decide
927  . decide
928  . decide
929  . decide
930  . decide
931  . decide
932  . decide
933  . decide
934  . decide
935  . decide
936  . decide
937  . decide
938  . decide
939  . decide
940  . decide
941  . decide
942  . decide
943  . decide
944  . decide
945  . decide
946  . decide
947  . decide
948  . decide
949  . decide
950  . decide
951  . decide
952  . decide
953  . decide
954  . decide
955  . decide
956  . decide
957  . decide
958  . decide
959  . decide
960  . decide
961  . decide
962  . decide
963  . decide
964  . decide
965  . decide
966  . decide
967  . decide
968  . decide
969  . decide
970  . decide
971  . decide
972  . decide
973  . decide
974  . decide
975  . decide
976  . decide
977  . decide
978  . decide
979  . decide
980  . decide
981  . decide
982  . decide
983  . decide
984  . decide
985  . decide
986  . decide
987  . decide
988  . decide
989  . decide
990  . decide
991  . decide
992  . decide
993  . decide
994  . decide
995  . decide
996  . decide
997  . decide
998  . decide
999  . decide
1000  . decide
1001  . decide
1002  . decide
1003  . decide
1004  . decide
1005  . decide
1006  . decide
1007  . decide
1008  . decide
1009  . decide
1010  . decide
1011  . decide
1012  . decide
1013  . decide
1014  . decide
1015  . decide
1016  . decide
1017  . decide
1018  . decide
1019  . decide
1020  . decide
1021  . decide
1022  . decide
1023  . decide
1024  . decide
1025  . decide
1026  . decide
1027  . decide
1028  . decide
1029  . decide
1030  . decide
1031  . decide
1032  . decide
1033  . decide
1034  . decide
1035  . decide
1036  . decide
1037  . decide
1038  . decide
1039  . decide
1040  . decide
1041  . decide
1042  . decide
1043  . decide
1044  . decide
1045  . decide
1046  . decide
1047  . decide
1048  . decide
1049  . decide
1050  . decide
1051  . decide
1052  . decide
1053  . decide
1054  . decide
1055  . decide
1056  . decide
1057  . decide
1058  . decide
1059  . decide
1060  . decide
1061  . decide
1062  . decide
1063  . decide
1064  . decide
1065  . decide
1066  . decide
1067  . decide
1068  . decide
1069  . decide
1070  . decide
1071  . decide
1072  . decide
1073  . decide
1074  . decide
1075  . decide
1076  . decide
1077  . decide
1078  . decide
1079  . decide
1080  . decide
1081  . decide
1082  . decide
1083  . decide
1084  . decide
1085  . decide
1086  . decide
1087  . decide
1088  . decide
1089  . decide
1090  . decide
1091  . decide
1092  . decide
1093  . decide
1094  . decide
1095  . decide
1096  . decide
1097  . decide
1098  . decide
1099  . decide
1100  . decide
1101  . decide
1102  . decide
1103  . decide
1104  . decide
1105  . decide
1106  . decide
1107  . decide
1108  . decide
1109  . decide
1110  . decide
1111  . decide
1112  . decide
1113  . decide
1114  . decide
1115  . decide
1116  . decide
1117  . decide
1118  . decide
1119  . decide
1120  . decide
1121  . decide
1122  . decide
1123  . decide
1124  . decide
1125  . decide
1126  . decide
1127  . decide
1128  . decide
1129  . decide
1130  . decide
1131  . decide
1132  . decide
1133  . decide
1134  . decide
1135  . decide
1136  . decide
1137  . decide
1138  . decide
1139  . decide
1140  . decide
1141  . decide
1142  . decide
1143  . decide
1144  . decide
1145  . decide
1146  . decide
1147  . decide
1148  . decide
1149  . decide
1150  . decide
1151  . decide
1152  . decide
1153  . decide
1154  . decide
1155  . decide
1156  . decide
1157  . decide
1158  . decide
1159  . decide
1160  . decide
1161  . decide
1162  . decide
1163  . decide
1164  . decide
1165  . decide
1166  . decide
1167  . decide
1168  . decide
1169  . decide
1170  . decide
1171  . decide
1172  . decide
1173  . decide
1174  . decide
1175  . decide
1176  . decide
1177  . decide
1178  . decide
1179  . decide
1180  . decide
1181  . decide
1182  . decide
1183  . decide
1184  . decide
1185  . decide
1186  . decide
1187  . decide
1188  . decide
1189  . decide
1190  . decide
1191  . decide
1192  . decide
1193  . decide
1194  . decide
1195  . decide
1196  . decide
1197  . decide
1198  . decide
1199  . decide
1200  . decide
1201  . decide
1202  . decide
1203  . decide
1204  . decide
1205  . decide
1206  . decide
1207  . decide
1208  . decide
1209  . decide
1210  . decide
1211  . decide
1212  . decide
1213  . decide
1214  . decide
1215  . decide
1216  . decide
1217  . decide
1218  . decide
1219  . decide
1220  . decide
1221  . decide
1222  . decide
1223  . decide
1224  . decide
1225  . decide
1226  . decide
1227  . decide
1228  . decide
1229  . decide
1230  . decide
1231  . decide
1232  . decide
1233  . decide
1234  . decide
1235  . decide
1236  . decide
1237  . decide
1238  . decide
1239  . decide
1240  . decide
1241  . decide
1242  . decide
1243  . decide
1244  . decide
1245  . decide
1246  . decide
1247  . decide
1248  . decide
1249  . decide
1250  . decide
1251  . decide
1252  . decide
1253  . decide
1254  . decide
1255  . decide
1256  . decide
1257  . decide
1258  . decide
1259  . decide
1260  . decide
1261  . decide
1262  . decide
1263  . decide
1264  . decide
1265  . decide
1266  . decide
1267  . decide
1268  . decide
1269  . decide
1270  . decide
1271  . decide
1272  . decide
1273  . decide
1274  . decide
1275  . decide
1276  . decide
1277  . decide
1278  . decide
1279  . decide
1280  . decide
1281  . decide
1282  . decide
1283  . decide
1284  . decide
1285  . decide
1286  . decide
1287  . decide
1288  . decide
1289  . decide
1290  . decide
1291  . decide
1292  . decide
1293  . decide
1294  . decide
1295  . decide
1296  . decide
1297  . decide
1298  . decide
1299  . decide
1300  . decide
1301  . decide
1302  . decide
1303  . decide
1304  . decide
1305  . decide
1306  . decide
1307  . decide
1308  . decide
1309  . decide
1310  . decide
1311  . decide
1312  . decide
1313  . decide
1314  . decide
1315  . decide
1316  . decide
1317 
```

even when suboptimal under public payoff assumptions. Our “suboptimal” terminology is therefore always relative to the stated payoff model, not a universal claim about all player preferences. With these limitations explicit, we summarize the main findings and concrete next steps in Section XI.

XI. CONCLUSION

This paper presents a formally verified metagame analysis pipeline for a real competitive TCG environment. Using Lean 4 plus Trainer Hill data, we prove a popularity paradox (Section VI) and connect it to equilibrium, dynamics, and tournament implications.

We also connect static matchup structure to dynamic and tournament implications. The Lean-verified real Nash equilibrium has six-deck support per player and 0% Dragapult weight; on representative observed-data subgames (4-deck and 5-deck slices, not the full 14-deck matrix), replicator dynamics predict Dragapult decline and Ceruleedge extinction pressure, though these subgame results do not generally transfer to the full game due to indirect fitness effects; and best-of-three math amplifies already-large matchup edges ($67.3\% \rightarrow 74.9\%$).

Immediate next steps are concrete: (i) extend the dataset to rolling weekly windows with uncertainty intervals, enabling forecast calibration by scoring each window’s predictions against the next, (ii) model the 30.5% “Other” segment explicitly instead of excluding it from matrix analysis, and (iii) add hierarchical sub-archetype clustering to capture list-level variance and pilot heterogeneity.

A. Broader Implications for Competitive Game Science

This case study suggests a general template for competitive-game research. First, formalize core mechanics and legality. Second, encode empirical payoff data as exact values. Third, express strategic claims as theorem-checkable statements. Fourth, tie those claims to tournament-relevant objectives instead of abstract utility alone. The resulting pipeline is portable across many environments with discrete strategy choices and measurable outcomes.

The portability argument matters because many competitive ecosystems face the same methodological failure mode: high-quality data exist, but conclusions are often produced by ad hoc tooling that mixes assumptions and results without explicit traceability. A proof-assisted workflow does not replace domain expertise; it structures it. Experts still decide which assumptions are reasonable, but once assumptions are fixed, conclusions become machine-auditable rather than rhetorical.

A second implication concerns collaboration between researchers and practitioners. Formal artifacts can be integrated into testing-team workflows as “verified baselines” against which local innovations are evaluated. For example, a team can begin from a certified weighted matchup model, then test whether a candidate list change moves specific matchup entries enough to alter tier or equilibrium-relevant conclusions. This is far more informative than relying on isolated scrim records without a stable analytical backbone.

Finally, this work contributes to a broader view of theorem proving in applied settings. Proof assistants are often associated with pure mathematics or compiler correctness. Our

results show they are also practical for empirical strategic science when the domain provides structured, finite data and well-defined objective functions. In that regime, formal methods can simultaneously improve reproducibility, interpretability, and decision quality.

Beyond this specific metagame snapshot, the broader contribution is methodological. Formal verification can serve as a practical scientific instrument for competitive game ecosystems: it turns qualitative metagame claims into executable definitions, theorem statements, and reproducible evidence.

DATA AVAILABILITY

Data were extracted from Trainer Hill (trainerhill.com) on February 19, 2026, filtering for Pokémon TCG events with 50 or more players between January 29 and February 19, 2026. Trainer Hill aggregates results from the Limitless TCG tournament platform. The 14×14 matchup matrix was computed from win-loss-tie records; ties were weighted as one-third of a win following the standard conversion $WR = (W + T/3)/(W + L + T)$. The complete matrix with raw W-L-T counts is archived in the repository. All Lean 4 source code, data files, and build instructions will be made publicly available upon acceptance in an anonymized artifact.

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] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, Eds., *Algorithmic Game Theory*. Cambridge University Press, 2007.
- [35] J. F. Nash, “Equilibrium points in n -person games,” *Proceedings of the National Academy of Sciences*, vol. 36, no. 1, pp. 48–49, 1950.
- [36] J. von Neumann, “Zur Theorie der Gesellschaftsspiele,” *Mathematische Annalen*, vol. 100, no. 1, pp. 295–320, 1928.
- [37] J. Hofbauer and K. Sigmund, *Evolutionary Games and Population Dynamics*. Cambridge University Press, 1998.
- [38] W. H. Sandholm, *Population Games and Evolutionary Dynamics*. MIT Press, 2010.
- [39] O. Romero and I. Millet, “Analysis of swiss-system tournament pairings,” *Journal of Quantitative Analysis in Sports*, vol. 18, no. 3, pp. 213–229, 2022.
- [40] K. Buzzard, J. Commelin, and P. Massot, “Formalising perfectoid spaces,” in *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP)*. ACM, 2020, pp. 299–312.