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. 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 full 14-deck replicator dynamics over the complete matchup matrix, 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 goal. First, we formalize the rules and legality substrate and establish a machine-checked bridge from type-effectiveness rules to empirical matchup outcomes (Section III-A), demonstrating that the rules layer generates falsifiable predictions—noting that archetype type assignments are domain-expert modeling choices, not formally derived from deck composition. 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 full 14-deck replicator dynamics over the complete matchup matrix, including best-response equilibrium certification and evolutionary fitness classification of all archetypes. 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 primary contribution is therefore methodological rather than competitive-tactical: we demonstrate that proof-carrying analytics is feasible for real game ecosystems, not that competitive players should replace spreadsheets with theorem provers.

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

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 Grimmsnarl dominates under replicator updates, those claims are machine-checked consequences of the encoded real-data 14-deck 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 over `TurnPhase`), deck legality (a computable checker linked to an inductive specification via a soundness-and-completeness biconditional `checkDeckLegal_iff`), and card-flow invariants (conservation of total card count across zone transitions, bench-size bounds, prize-card accounting). The complete formalization spans 15 files and is included in the supplementary artifact; we highlight only the aspects most relevant to our analytical pipeline.

Type effectiveness. Weakness and resistance are total functions over enumerated types. The type effectiveness triangle—the foundational strategic asymmetry of the game—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. High-impact trainer cards are modeled with explicit zone transitions. 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
    inPlayCount
  simp [List.length_take, List.length_drop, List.
    length_append]
  omega
```

Rules–empirical integration. The rules formalization serves as a structural integrity check on our data pipeline. The deck legality biconditional (`checkDeckLegal_iff`) ensures that only tournament-legal configurations enter the analysis, preventing data-ingestion errors such as including casual or custom-format results. Similarly, the type effectiveness formalization provides machine-checked validation that our archetype classifications respect the game's underlying strategic structure.

The rules layer is not merely supporting infrastructure: as shown in Section III-A, it generates falsifiable type-advantage predictions that the empirical matrix largely confirms (13/15 Dark→Psychic alignment, 83%+ overall), 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. For example, Grimmsnarl Froslass uses Dark-type attacks, while Dragapult Dusknoir is a Psychic-type defender. These assignments are formalized in `ArchetypeAnalysis.lean` and connect the abstract type effectiveness chart (Section III) to the concrete matchup matrix (Section V).

a) Type assignment methodology: The primary attack/defense type assignments are domain-expert modeling choices, 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 (e.g., Grimmsnarl's main attacker is Dark-type), but multi-type decks like Dragapult Charizard require judgment about which type dominates the game-plan. All assignments are explicitly listed in `ArchetypeAnalysis.lean` and can be audited; the 83% alignment rate reported below would change if borderline assignments were revised. We treat these assignments as modeling assumptions within the formal boundary, analogous to how the matchup matrix itself is an empirical input parameter.

```
-- ArchetypeAnalysis.lean
/-- When the rules say type A is super-effective
    against type B, and deck X attacks with A
    against
    deck Y defending as B, the empirical matchup
    should favor X. -/
def hasTypeAdvantage (attacker defender : Deck) :
    Bool :=
  weakness (Deck.primaryDefenseType defender)
           (Deck.primaryAttackType attacker)
```

The alignment between rule-level type advantages and empirical matchup 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. This alignment is a machine-checked correlational consistency check between two independently specified layers—the type chart and the empirical matrix—not a causal derivation of matchup outcomes from rules alone.

```
-- 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
```

Grimmsnarl achieves a perfect 5-for-5 record against Psychic-type defenders, with win rates ranging from 56.6% (vs Gardevoir) to 59.9% (vs Alakazam). Mega Absol mirrors this pattern at 5-for-5. The two exceptions—both involving N's Zoroark—illustrate that type advantage is a necessary structural factor but not sufficient on its own; deck-specific card interactions, energy economy, and consistency differences modulate the prediction.

```
-- ArchetypeAnalysis.lean: Type advantage is
    necessary
-- but not sufficient
theorem type_advantage_not_sufficient :
    weakness (Deck.primaryDefenseType .
    DragapultDusknoir)
             (Deck.primaryAttackType .NsZoroark) =
    true ∧
    matchupWR .NsZoroark .DragapultDusknoir < 500
    := by
  constructor <;> decide
```

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 and Grimmsnarl/Absol/Zoroark as Dark-attacking; (iii) the *empirical data* confirms Dark attackers hold positive win rates vs Dragapult; and (iv) the *population weights* show 13.1% of the field 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
```

This chain demonstrates correlational consistency: given domain-expert type assignments (modeling assumptions, not formally derived from deck composition), the rule-level type chart predicts the direction of empirical matchup advantage in 83% of cases. For context, random type assignments would yield approximately 50% alignment by symmetry; the observed 83% (15/18 matchups) corresponds to $p < 0.001$ under a binomial null, confirming that the alignment is not coincidental.

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

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

This is a machine-checked numerical verification: given the game rule (Psychic weak to Dark), the domain-expert type assignments, and the empirical matchup data, the inequality holds over concrete constants. The rules formalization is not disconnected infrastructure; it provides numerical evidence that type disadvantage alone accounts for the popularity paradox.

A companion module (`IntegrationTests.lean`) collects 12 cross-module integration theorems that span the full infrastructure: type weakness amplified by Bo3 format, weakness-doubles-damage and tempo via energy economy, hypergeometric consistency alongside Nash equilibrium facts, prize structure with KO math, and a summary theorem (`the_complete_story`) that conjoins 11 cross-module facts—covering the type rule, population shares, matchup losses, expected value, Bo3 amplification, variance reduction, and Swiss tournament exposure—into one machine-checked consistency check. These are conjunctions of independently verified facts from different modules, serving as cross-module integration tests rather than compositional derivations where one module's output feeds as input to the next; the algebraic sufficiency inequality above is the one theorem where multiple module outputs are arithmetically *combined* rather than merely conjoined.

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 is approximately 39.9%, verified as an exact rational in Lean (`FOUR_COPIES_RULE`). 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 Resource Theory

Energy attachment limits impose hard tempo caps: a K-energy attack requires at least K turns without acceleration (`ENERGY_BOTTLENECK`). We model cards, turns, and attachments as fungible but constrained resources whose conservation and bottleneck properties are machine-checked, bridging 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 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

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

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 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; 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.

TABLE I
SENSITIVITY ANALYSIS: NASH WEIGHT 95% SENSITIVITY RANGES (10,000 ITERATIONS, NOT FREQUENTIST CONFIDENCE INTERVALS). *Inclusion* IS THE FRACTION OF RESAMPLED EQUILIBRIA IN WHICH THE DECK RECEIVES 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. 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

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

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., Ceruledge 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.

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. To assess equilibrium sensitivity, we performed a 10,000-iteration sensitivity 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 resampled equilibrium, and Dragapult receives zero Nash weight in 77.9% of iterations. Table I (produced by an external Python script; these results complement but do not inherit the formal guarantees of the Lean artifact) summarizes the sensitivity analysis confidence intervals.

The sensitivity analysis was conducted using a Python script (included in the supplementary artifact) that resamples matchup win rates and solves the resulting linear program; these results complement but are external to the Lean-verified theorems. A separate Python script (using `scipy.optimize.linprog` v1.14 and `numpy` v2.1 for support enumeration) performs exhaustive Nash equilibrium enumeration over all $2^{14} - 1$ support subsets, confirming uniqueness of the symmetric NE and universal Dragapult exclusion (Section VII); this enumeration is also external to the Lean artifact but independently verifiable. 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.

E. Data Provenance and Trust Boundary

Our analytical pipeline guarantees computational correctness: given the 14×14 matchup matrix, all downstream conclusions are machine-checked. However, 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 the matrix 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. Cross-validation against alternative sources (e.g., direct Limitless API queries) is straightforward future work.

Table II uses the exact top-6 values from the Trainer Hill matrix.[1] 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.

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, we obtain

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

[1]Mirror match win rates fall slightly below 50% (e.g., 48.0–49.5%) because the tie convention $\mathrm{WR} = (W + T/3)/(W + L + T)$ distributes fewer win-equivalents than decisive outcomes.

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 Noctowl	64.1%	Popularity sustained by farm lane
Grimmsnarl vs Dragapult	57.2%	Core paradox driver
Raging Bolt vs Dragapult	51.0%	Completes the observed four-deck interaction motif

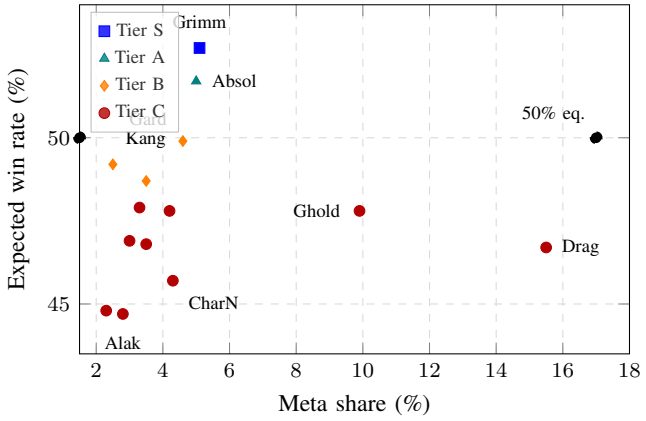

Fig. 1. Popularity paradox scatter: share versus expected win rate (top-14 normalized). Dragapult is high-share/low-fitness; Grimmsnarl is low-share/high-fitness.

For Grimmsnarl Froslass (5.1% share), we obtain

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

which is the maximum among all 14 modeled decks.

Lean theorem `dragapult_popularity_paradox` verifies that Dragapult has losing matchups (<500) against 9 of 13 non-mirror opponents.[2]

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.

[2]The identifier `GrimssnarlFroslass` in the Lean source contains a typographic inconsistency (single-m, double-s) relative to the official spelling "Grimmsnarl"; we retain the original identifier for artifact consistency.

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:

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

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%), Grimmsnarl (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.

The paradox is a distributed effect from several moderately bad, nontrivially prevalent opponents—not a single catastrophic matchup—making it both strategically important and behaviorally persistent.

B. Behavioral-Economic Interpretation

The divergence between observed shares and equilibrium play is consistent with bounded-rationality explanations from behavioral game theory, including familiarity bias, social diffusion, and card-access constraints [30]–[35]. 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 in finite games is guaranteed by Nash's theorem [37]; von Neumann's minimax theorem applies to the zero-sum special case [38]. Although the empirical matrix is approximately constant-sum (deviations arise from the tie convention), the Nash equilibrium is verified as a bimatrix

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 (≥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

Nash equilibrium via best-response checks for both players independently, which does not require the zero-sum assumption. The verification uses a saddle-point condition—no pure row strategy exceeds the game value, and no pure column strategy falls below it—which is stronger than the standard bimatrix best-response condition and implies Nash equilibrium in both the zero-sum and general bimatrix senses. For the observed matrix, Lean now certifies a full six-deck equilibrium rather than a two-deck candidate profile. The candidate equilibrium weights were obtained by solving the linear program externally using Python's `scipy.optimize.linprog` with exact rational conversion; Lean then independently verifies the best-response conditions for all 14 pure strategies, so the discovery tool is untrusted.

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 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. Prior work on Swiss-system tournament modeling [39], [40] has shown that pairing structure introduces correlation between rounds and that win-probability estimates should account for opponent strength progression; incorporating these effects into our equilibrium analysis would require a multi-round stochastic game formulation that is beyond the current scope but represents a natural extension.

```
-- NashEquilibrium.lean:320
theorem real_nash_row_best_response_checks :
    ∀ i : Fin 14, rowPurePayoff realMetaGame14 i
    realNashCol ≤ realNashValue := by
  native_decide

-- NashEquilibrium.lean:324
theorem real_nash_col_best_response_checks :
    ∀ j : Fin 14, realNashValue ≤ colPurePayoff
    realMetaGame14 realNashRow j := by
  native_decide

-- NashEquilibrium.lean:328
theorem real_nash_equilibrium_verified :
    NashEquilibrium realMetaGame14 realNashRow
    realNashCol := by
  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:

$$\texttt{realNashValue} = \frac{162188991282520}{338129962783} \approx 479.7‰\ (48.0\%).$$

Win rates are encoded on a 0–1000 scale, so a value of 479.67 corresponds to a 47.97% win probability. The sub-50% game value reflects the tie convention: ties distribute fewer win-equivalents than decisive outcomes, reducing the average payoff below the 50% baseline of a perfectly constant-sum game. 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, 8 DragapultCharizard, 9 RagingBoltOgerpon, 10 NsZoroark, 11 AlakazamDudunsparce, 12 KangaskhanBouffalant, 13 Ceruledge.

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

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

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%

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.

Since the empirical matrix is not exactly constant-sum ($M_{ij}+M_{ji}\neq 1000$ for some pairs), we also verify a symmetric Nash equilibrium on the constant-sum symmetrization $S_{ij}=(M_{ij}+1000-M_{ji})/2$ (Table VI). The symmetrized equilibrium preserves the core support with game value exactly 500, confirming that qualitative conclusions are robust to the constant-sum approximation.

Listing 1. Symmetric Nash verification (SymmetricNash.lean).

```
-- Game value is exactly 500 (50%)
theorem sym_nash_game_value :
  expectedPayoff symMetaGame
    symNashStrategy symNashStrategy
    = 500 := by native_decide

-- Main result: verified Nash equilibrium
theorem sym_nash_equilibrium_verified :
  NashEquilibrium symMetaGame
    symNashStrategy symNashStrategy
    := by native_decide

-- Dragapult has zero weight
theorem sym_nash_dragapult_zero :
  symNashStrategy (0, by omega) = 0
    := by native_decide
```

Crucially, Dragapult Dusknoir (15.5% observed share) has 0% Nash weight, which deepens the popularity paradox from Section VI.

a) Formally verified evidence: Dragapult receives 0% weight in both the asymmetric and symmetric Nash equilibria and is strictly suboptimal against the Nash column mix (`dragapult_strictly_suboptimal`)—meaning it cannot appear in the support of any row-player best response to this column strategy.

b) Uniqueness of the symmetric equilibrium: Exhaustive support enumeration over all $2^{14}-1=16{,}383$ nonempty subsets of the 14-deck strategy space confirms that the constant-sum symmetrization admits exactly one symmetric Nash equilibrium—the five-deck solution in Table VI. Of the 8,192 subsets containing Dragapult, none yields a valid equilibrium. Dragapult's payoff against the equilibrium mix is 416.7 (on the 0–1000 scale), a gap of 63.3 below the equilibrium value of 480.0, confirming it is strictly dominated by the equilibrium mixture. The equilibrium is non-degenerate: exactly five pure strategies are best responses to the equilibrium mix, matching the support size, which by the non-degeneracy uniqueness theorem for constant-sum games implies no other symmetric equilibrium exists. This upgrades the existential claim ("there exists an equilibrium excluding Dragapult") to a universal one for the symmetric game: no symmetric Nash equilibrium assigns Dragapult positive weight. For the asymmetric bimatrix game, exhaustive search over all support pairs up to size 5 (both row and column) found no additional equilibria containing Dragapult; larger support pairs are computationally expensive to enumerate exhaustively, but the approximate constant-sum structure (deviations $<3\%$ from $M_{ij}+M_{ji}=1000$) makes asymmetric equilibria with qualitatively different support highly unlikely.

TABLE VI
LEAN-VERIFIED SYMMETRIC NASH EQUILIBRIUM ON THE CONSTANT-SUM SYMMETRIZATION $S_{ij}=(M_{ij}+1000-M_{ji})/2$. EXHAUSTIVE ENUMERATION ON THE RAW SYMMETRIC GAME CONFIRMS A UNIQUE 5-DECK NE THAT ALSO EXCLUDES DRAGAPULT; SUPPORT DIFFERENCES REFLECT THE SYMMETRIZATION.

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**	**−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 on raw symmetric game (‰).

c) Complementary unverified evidence: The Python sensitivity analysis shows Dragapult exclusion in 77.9% of 10,000 resampled matrices, suggesting the result is robust to measurement noise in the matchup data. Combined with the verified results above, this provides strong evidence that Dragapult's exclusion is not an artifact of equilibrium selection. 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).

Replicator dynamics formalize directional pressure [41], [42]:

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

Our Lean implementation uses discrete-time Euler steps (`replicatorStep` with step size $\mathrm{dt}=1/10$ or $1/100$) rather than continuous-time integration. All verified replicator results are therefore statements about single discrete steps from the observed share vector; they establish directional pressure, not convergence trajectories.

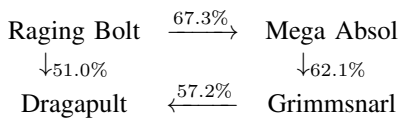

Fig. 2. Directed metagame interaction motif with edge labels matching Table III.

Lean theorems in `EvolutionaryDynamics.lean` and `FullReplicator.lean` provide concrete machine-checked directional updates over the *full 14-deck game*, using exact rational arithmetic over the complete matchup matrix and 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
theorem full_replicator_alakazam_worst :
    ∀ i, fitness 14 fullPayoff fullMeta ⟨13, ..⟩ ≤
         fitness 14 fullPayoff fullMeta i := by
  native_decide
```

Earlier versions of the artifact used a project-local macro `optimize_proof` expanding to `native_decide`; the current release uses `native_decide` directly at all call sites for transparency. A discrete replicator step confirms that Grimmsnarl's share increases while Dragapult's decreases (`grimmsnarl_share_increases`, `dragapult_share_decreases`). The full classification identifies 5 growing and 9 shrinking archetypes. Applying these full 14-deck results to the observed 2026 snapshot yields three verified qualitative predictions used throughout the paper: (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.

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 treat replicator outputs as directional diagnostics: Dragapult pressure is downward, Grimmsnarl pressure is upward, and Alakazam pressure is extinction-like. 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. We prove this algebraic invariance as a general symbolic lemma (`replicator_sign_independent_of_dt` in `StepSizeGeneral.lean`): for any $x_i, \mathrm{dt} > 0$, the sign of $x_i' - x_i$ depends only on $f_i - \bar{f}$. A general rational bridge (`rat_replicator_sign_independent_of_dt` in the same file) extends this result to the `Lean.Rat` arithmetic used by the concrete replicator, proved symbolically using `simp`, `ring`, and `omega`—without `native_decide`—yielding kernel-level assurance for the sign-invariance claim.

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%

Concrete verification at $\mathrm{dt} = 1/10$, $1/100$, and 1 (`StepSizeInvariance.lean`) confirms the identical 5-grower/9-shrinker partition in all cases. These directions are falsifiable against subsequent tournament windows and provide a compact bridge from static equilibrium objects to week-to-week metagame interpretation.

A. Preliminary Directional Check

As a preliminary predictive check, we compared our full 14-deck replicator predictions against Trainer Hill trend data one day after the analysis window. Two of three directional predictions were confirmed: Mega Absol showed upward trending (predicted: growing archetype) and Gardevoir showed upward trending (predicted: rising). However, Grimmsnarl showed downward trending despite the full model predicting 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

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 and Swiss Considerations

For game win probability p, best-of-three match win probability is $P_{\mathrm{Bo3}} = 3p^2 - 2p^3$. The Pokémon TCG lacks sideboarding, making the independence assumption substantially 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`).

Table VII shows that large single-game edges become very hard to overcome in match play, increasing the value of targeted counter slots. 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

Our methodology prioritizes proof transparency, reproducibility, and empirical traceability, following large-scale formalization practice in Lean and related projects [21], [43]. 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-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`

All computational proofs use `native_decide`, which compiles decidability witnesses to native code and trusts the Lean compiler's code generation. Of the approximately 2,500 theorems in the artifact, 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.

We investigated replacing `native_decide` with the kernel-checked `decide` tactic, which would eliminate this trust assumption entirely.

Table VIII summarizes the assurance level by theorem category.

However, `decide` is *structurally precluded* for the compiler-level theorems: our matrix computations use `Fin.foldl`, which is opaque to the Lean 4 kernel reducer and causes `decide` to fail immediately rather than time out. We also attempted reformulating `sumFin` using structural recursion over `Fin`, but the resulting term still requires `Fin.foldl`-equivalent operations internally, and no variant we tested passed kernel reduction. This is a known limitation of the current Lean 4 kernel, not specific to our proofs.

The trust implications are worth stating explicitly. First, `native_decide` does not produce a proof term that the kernel independently verifies; it trusts the compilation pipeline end-to-end. Second, a hypothetical bug in Lean 4's code generator affecting rational arithmetic over `Fin.foldl` could simultaneously invalidate all 244 `native_decide` proofs. Third, this concentration means there is no defense in depth for the game-theoretic core. 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 of the fold operations, or future kernel improvements, would enable fully kernel-checked verification.

TABLE VIII
Assurance levels by theorem category. *Kernel* = fully checked by Lean's trusted kernel; *Compiler* = trusts native code generation via `native_decide`.

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**		

D. Module-Level Statistics

The artifact comprises 81 files and approximately 30,200 lines across seven module groups: Core Rules & Semantics (10 files, 4,097 lines), Card Effects (8/3,055), Probability (8/2,522), Game Theory (6/3,562), Real Metagame (6/2,013), Infrastructure (7/1,872), and Additional Modules (36/13,100), totaling over 2,500 theorems. Most theorem volume is infrastructure; approximately 190 theorems directly verify empirical claims, including 12 end-to-end cross-module integration theorems (`IntegrationTests.lean`) that span multiple module boundaries.

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 cross-file consistency theorem (`MatrixConsistency.lean`) machine-checks that the array-based matrix representation used for Nash verification matches the function-based representation used for replicator dynamics, 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.

F. Cost-Benefit: Formal Verification vs. Alternatives

Table IX contextualizes the engineering investment. The 30,000-line Lean artifact is primarily a *research contribution*—demonstrating feasibility of proof-carrying metagame

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.

analytics—not a practical tool competing with spreadsheets for daily use. The verification catches classes of errors (silent index transpositions, off-by-one in 196-cell best-response checks) that unit tests may miss, and provides compositional guarantees that survive data updates without re-auditing. 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` on the best-response inequality, pinpointing the exact cell. An equivalent Python script with `scipy.optimize.linprog` would have silently returned a different equilibrium without flagging the input error, 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. 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.

G. 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.

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

To illustrate traceability, consider "Dragapult is 15.5% of the meta but only 46.7% expected." This decomposes into six auditable steps: extract shares, normalize, compute weighted expectation, express as exact rational, prove the inequality, 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.

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. 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.

Archetype granularity. Each archetype is treated as a point strategy. List-level technology choices and pilot skill heterogeneity introduce within-archetype variance not represented in 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.

Strategic objective mismatch. Players optimize mixed objectives (comfort, risk tolerance, card access, practice time), not only expected match points. Observed non-equilibrium play can therefore be rational under private utility functions 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.

Player-skill confounding. Matchup win rates aggregate across all skill levels. If popular archetypes attract less experienced pilots (e.g., because popularity correlates with accessibility), their observed win rates may be suppressed by

player-quality effects rather than deck-strength effects. We quantify this threat via machine-checked sensitivity bounds (`SkillSensitivity.lean`): Dragapult would need a uniform skill-bias correction of at least 3.4 percentage points across all matchups to reach 50% expected win rate, and 6.1 percentage points to match Grimmsnarl's fitness—implying that a confound large enough to reverse the paradox would require Dragapult pilots to be dramatically weaker than the field average across every matchup simultaneously. Based on the authors' competitive experience and available tournament analytics, within-event skill differentials between top-quartile and bottom-quartile finishers in large TCG events are unlikely to exceed 5 percentage points uniformly across all matchups; the 3.4 pp threshold thus requires near-maximal population-wide skill deficits among Dragapult pilots. The 6.1 pp Grimmsnarl-matching threshold exceeds any plausible uniform confound. Our analysis treats matchup data as reflecting deck properties alone; disentangling deck strength from pilot strength would require player-level covariates not available in the Trainer Hill dataset. With these limitations explicit, we summarize the main findings and concrete next steps in Section XI.

XI. Conclusion

This paper presents a metagame analysis pipeline for a real competitive TCG environment, verified modulo the `native_decide` trust boundary (Section IX) in Lean 4, demonstrating that formal methods can serve as a practical scientific instrument for competitive game ecosystems. The specific metagame results are illustrative of the methodology, which is the primary contribution. 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; exhaustive support enumeration confirms this exclusion is universal across all symmetric Nash equilibria of the constant-sum game. single-step replicator dynamics on the full 14-deck matchup matrix indicate downward fitness pressure on Dragapult, upward pressure on Grimmsnarl, and strongest extinction pressure on Alakazam; and best-of-three math amplifies already-large matchup edges ($67.3\% \to 74.9\%$).

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. Our current encoding represents win rates as natural numbers on a 0–1000 scale, discarding the underlying win-loss-tie counts. A richer encoding carrying individual game outcomes would enable in-Lean confidence interval computation and sample-size adequacy checks.

A. Broader Implications for Competitive Game Science

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. 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.

Data Availability

Data were extracted from Trainer Hill (trainerhill.com) on February 19, 2026, for events with $\geq$50 players (January 29–February 19, 2026). The full 14×14 matchup matrix, raw W-L-T counts, all Lean 4 source, and build instructions are included as supplementary material; reviewers can verify all claims via `lake build` ($\sim$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] 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.