# Self-Play Calibration of Heuristic Agents for Mexican Train Under Competing Objectives

Digital Defiance
https://github.com/digitaldefiance/warp12

July 15, 2026

**Abstract**

Mexican Train dominoes is commonly played under two incompatible victory conditions: **points** (lowest pip total when the round ends) and **go-out** (first player to empty their hand). This paper describes the development, calibration, and empirical validation of AI agents for both objectives in Warp 12, an open-source Mexican Train implementation. We introduce the **Tactical Effectiveness Index (TEI)**, a dual-track OpenSkill-based rating system with gamified grade display, anchored to fixed AI reference tiers.

Empirical results from thousands of self-play games show that **skill ordering is stable under the points objective** (consistent rating gaps between tiers), while **go-out exhibits higher variance** and compressed skill gaps. We demonstrate that search algorithms provide different advantages depending on context: expectimax wins in 2-player points (64% vs heuristics), while ISMCTS excels in 4+ player go-out (31% vs 23% baseline).

A **285,000-game** points self-play campaign across **570 configuration cells** (4 Warp factors × every legal fleet size × 15 module configs) is our sole skill/luck instrument for board choice and module taxonomy. From the **baseline** cells alone we find that **skill expression increases with both tile set size and player count**, contradicting the intuition that larger games devolve into luck, and that Warp factor 12 is the empirically optimal choice for competitive rating. Across the full matrix, most optional modules preserve the baseline skill ceiling; **Module Iota (Double Down) slightly raises it**. **Module Epsilon (Drafting)** collapses it (1.08/4 indicators) — counter-intuitively, choosing your hand makes outcomes *more* luck-dominated mid-game — and is therefore designated **Warped / party**. **Module Zeta (Squadrons)** also preserves skill (2.94/4); it never writes free-for-all (FFA) TEI — rated Warp 12 crew sectors update a dedicated **Squad TEI** track (`squadRating`) after team-track calibration (`SQUADRONS_RATING_CALIBRATED`), not because Zeta is luck-dominant.

Finally, we report on **Class Ω**, a neural self-play agent that achieves parity with heuristic baselines in small games and outperforms in large fleets. All code, data, and reproduction scripts are available as open source.

# 1 Introduction

## 1.1 Mexican Train Dominoes

Mexican Train is a popular multi-player domino game typically played with a double-twelve set (91 tiles with pips ranging from 0 to 12). Players build linear sequences called *trains* from a central *engine* (starting double), attempting to minimize their hand's pip count. The game features both private trains (one per player) and a shared public train (the "Mexican Train" or Neutral Zone), creating strategic tension between building one's own position and blocking opponents.

Unlike classic domino games with a single win condition, Mexican Train is commonly played under two incompatible objectives:

- **Points campaign:** Play continues for multiple rounds (descending from engine double-12 down to double-0). After each round, players accumulate penalty points equal to the pips remaining in their hand. Lowest cumulative score after all rounds wins.
- **Go-out:** The first player to empty their hand wins immediately, ending the round.

These objectives create fundamentally different strategic games on the same rules engine: points rewards consistent pip minimization and flexible play across many turns, while go-out rewards tempo, connectivity, and race dynamics.

### 1.2 Warp 12: Terminology and Presentation

This paper describes work conducted in the context of Warp 12, an open-source Mexican Train implementation with a Star Trek-themed presentation layer. Throughout this paper, we use **standard Mexican Train terminology** in formal discussion, but introduce Warp 12's thematic terminology where relevant to implementation details. The table below provides a complete mapping.

Table 1: Warp 12 terminology mapped to Mexican Train equivalents

| **Mexican Train** | **Warp 12** | **Technical** |
|---|---|---|
| *Game Elements* | | |
| Domino / tile | Navigational Coordinate | `Coordinate` |
| Engine (starting double) | Spacedock | `engine` |
| Player's private train | Warp Trail | `ownTrail` |
| Mexican Train (public) | Neutral Zone | `neutralZone` |
| Train marker / forced play | Distress Beacon | `beacon` / `shieldsDown` |
| Unsatisfied double | Red Alert | `redAlert` |
| Draw pile | Uncharted Sectors | `pile` |
| *Players and Rating* | | |
| Player | Captain | `PlayerId` |
| Skill rating | TEI grade | `TeiDisplay` (E/V/C/I/P + 0–99) |
| Underlying skill | OpenSkill $(\mu, \sigma)$ | `PlayerRating` |
| Game table | Sector / Fleet | `game` |
| *AI Tiers* | | |
| Easy AI | Class IV (Ensign) | `ensign` |
| Medium AI | Class III (Lieutenant) | `lieutenant` |
| Hard AI | Class II (Commander) | `commander` |
| Expert (human only) | Class I | High TEI score / low $\sigma$ |

**Warp factors** refer to the maximum pip value in the domino set: Warp 9 (W9) uses double-nine tiles (55 tiles total), Warp 12 (W12) uses double-twelve (91 tiles), etc. Throughout this paper, we use compact notation **WX/Yp** to denote configurations: W12/4p means Warp factor 12 (double-twelve) with 4 players. This paper focuses on W12 as the primary rated configuration, with analysis of W9, W15, and W18 in Section 8.

### 1.3 Motivation

Despite Mexican Train's popularity, it remains under-studied in the game AI literature compared to chess, Go, or poker. Several challenges make it interesting for AI research:

- **Imperfect information:** Opponent hands and draw pile order are hidden, requiring belief-state reasoning rather than perfect-information tree search.
- **Multi-player dynamics:** Unlike 2-player zero-sum games, Mexican Train typically involves 3–8 players with complex coalition and blocking incentives.
- **Dual objectives:** The same rule set produces two strategically distinct games depending on win condition.
- **Stochastic elements:** Random draws from the pile, variable starting hands, and opponent action uncertainty all contribute to outcome variance.

For online play, these challenges create practical design questions: How do we measure player skill fairly? How do we provide appropriately challenging AI opponents? How do we ensure coaching tools don't contaminate competitive ratings?

### 1.4 Contributions

This paper makes the following contributions:

1. **Dual-track TEI rating system** — We describe an OpenSkill-based rating with independent tracks for points and go-out objectives, anchored to fixed AI reference tiers rather than floating human populations. The system uses Bayesian inference with $\mu \pm \sigma$ skill estimates and presents ratings as gamified grades (E/V/C/I/P + 0-99 score).
2. **Self-play calibration methodology** — We present a systematic approach to validating AI skill levels through tier-vs-tier matrices, symmetric seating tests, and multi-player focus matchups.
3. **Empirical objective comparison** — Through thousands of self-play games, we demonstrate that points calibrates cleanly with consistent rating gaps between tiers, while go-out exhibits compression and higher variance due to its racing mechanics.
4. **Search algorithm analysis** — We show that different search methods (expectimax, ISMCTS) provide advantages in different contexts: expectimax excels in 2-player points (64% win rate), while ISMCTS works better in 4+ player go-out (31% vs 23% baseline).
5. **Luck vs skill and module integrity** — A single **285,000-game** matrix (570 cells) measures skill expression across Warp factors, fleet sizes, and 15 module configs. Baseline cells justify rating W12; Iota slightly raises the ceiling; Epsilon collapses it (Warped/party); Zeta preserves skill and rates on a dedicated Squad TEI track (never free-for-all / FFA).
6. **Neural self-play agent (Class Ω)** — We demonstrate that a pure self-play neural policy can achieve competitive performance (parity in small games, advantages in large fleets) without hand-crafted heuristics, and discuss the challenges of mapping training performance to appropriate rating anchors.
7. **Open-source implementation** — All code, data, and reproduction scripts are available through the warp12-engine package and associated repositories.

## 2 Related Work

This section briefly surveys prior work in domino game AI, game AI paradigms relevant to Mexican Train, and skill rating systems. We identify the specific gap this work fills.

### 2.1 Domino and Tile Games

Domino games have received limited attention in the game AI literature compared to board games like chess and Go or card games like poker. Most prior work focuses on simpler variants like straight dominoes or block dominoes, which lack Mexican Train's multi-player train mechanics and dual objectives. The DoubleEighteen rendering library [11] provides visualization for domino games but does not include AI agents. To our knowledge, this is the first published calibration study of AI agents for Mexican Train specifically.

### 2.2 Game AI Paradigms

Several AI approaches are relevant to imperfect-information multi-player games like Mexican Train:

**Heuristic policies:** Hand-crafted evaluation functions and decision rules remain competitive in many domains, especially where interpretability matters for player trust and coaching features. Our Class IV–II agents follow this tradition with weighted heuristics for pip dumping, trail pressure, and blocking.

**Monte Carlo Tree Search (MCTS):** MCTS and its variants [1] have achieved strong results in perfect-information games (Go, Hex) and some imperfect-information settings. Information Set MCTS (ISMCTS) [4] handles hidden information by sampling determinizations and maintaining search trees over information sets rather than individual states. We use ISMCTS for our Fleet Admiral and Class $\Omega+$ search backends.

**Deep reinforcement learning and self-play:** The AlphaZero family [2, 9] demonstrated that pure self-play with neural networks can surpass human expertise in chess, Go, and shogi. However, these successes required massive computational resources and perfect information. Our Class $\Omega$ agent represents a more modest self-play approach suitable for imperfect-information games with limited training compute.

**Counterfactual regret minimization:** CFR and its variants have achieved superhuman performance in poker [10], another imperfect-information game. However, CFR typically requires extensive offline precomputation and works best in 2-player zero-sum settings, making it less applicable to multi-player cooperative-competitive games like Mexican Train.

### 2.3 Skill Rating Systems

Several rating systems are widely used in competitive games:

**Elo rating [3]:** Originally developed for chess, Elo computes expected win probabilities based on rating differences and updates ratings based on actual outcomes. It assumes transitive skill relationships and works best for 2-player zero-sum games.

**Glicko [6]:** Extends Elo by modeling rating uncertainty (RD), allowing ratings to drift during inactivity periods and providing confidence intervals.

**TrueSkill [7]:** Microsoft's rating system extends to multi-player and team games using Bayesian inference. It models each player's skill as a Gaussian distribution and updates beliefs after each match.

**OpenSkill:** An open-source implementation of rating systems inspired by TrueSkill and Weng-Lin ranking [8], designed for multi-player and team-based games. Like TrueSkill, it uses Gaussian

skill models with mean ($\mu$) and standard deviation ($\sigma$) parameters, updated through Bayesian inference. We adopt OpenSkill as the foundation for TEI due to its multi-player support, open implementation, and flexible team rating capabilities.

Most online game platforms (Chess.com, Lichess, etc.) use Elo or Glicko variants. Some platforms that offer AI opponents (e.g., Lichess's Stockfish bots) provide fixed rating estimates for bot difficulty levels, similar to our TEI anchor approach. However, we are unaware of prior work that systematically calibrates and validates AI skill tiers through self-play specifically for the purpose of providing stable rating anchors.

### 2.4 Contribution Relative to Prior Work

This work makes several novel contributions relative to the existing literature:

1. **Dual-objective calibration:** We systematically compare points and go-out objectives on the same rules engine, showing empirically that they require different strategic optimization and exhibit different skill/variance tradeoffs.

2. **Fixed AI reference anchors:** Rather than deriving ratings entirely from human populations (which can drift over time), we anchor rating bands to validated AI tiers with measured performance characteristics.

3. **Multi-player self-play methodology:** Most game AI calibration focuses on 2-player matchups. We include 3–8 player configurations and focus matchups (one strong player vs multiple weaker opponents) to validate skill expression in realistic play scenarios.

4. **Luck vs skill empirics:** The 285,000-game module matrix (Section 9) is the systematic measurement of how tile set size, player count, and optional modules affect skill expression — including Warped/party designation for Epsilon and Warp-factor board choice from baseline cells.

5. **Open-source implementation:** All code (rules engine, AI agents, calibration scripts) and data are publicly available for reproduction and extension.

## 3 Game Model and Rules Engine

### 3.1 Core Mexican Train Rules

Mexican Train follows standard multi-trail domino game mechanics:

1. **Setup:** All players draw a fixed hand size (typically 15 tiles for 2–4 players, fewer for larger games). The engine double (e.g., 12-12 for Warp 12) is placed in the center. Remaining tiles form the draw pile.

2. **Turn structure:** On each turn, a player must play one tile that matches the open end of any available train (their own, another player's with a marker, or the public Mexican Train). If unable to play, they draw one tile from the pile and may play it immediately if legal.

3. **Doubles:** When a double is played, it must be "satisfied" (another tile played on it) before anyone can play elsewhere. This creates a *Red Alert* state that forces immediate resolution.

4. **Train markers:** When a player cannot play on their turn, they place a marker on their train, making it public for one round. In Warp 12 terminology, this is called a *Distress Beacon* and the state is *Shields Down*.

5. **Mexican Train:** The public train is always available to all players, providing a release valve when private trains are blocked.

6. **Round end:** A round ends when a player empties their hand (go-out objective) or when the pile is exhausted and no legal plays remain (points objective).

### 3.2 Implementation as State Machine

Our implementation models the game as an immutable state machine with pure functions:

- **State:** Readonly data structures (`GameState`, `RoundState`, `TableState`) containing all game information: player hands, visible tiles, pile size, train states, alert conditions.
- **Actions:** Typed events (`PlayAction`, `DrawAction`, etc.) representing legal moves.
- **Transitions:** Pure function `applyAction(state, action) -> newState` implements all rules.
- **Legal moves:** Function `getLegalMoves(state, playerId) -> Action[]` computes all valid plays for the current state.

This design ensures that AI agents, human players, and replay validation all use identical game logic. No special-case code paths exist for AI vs human play.

### 3.3 House Rules and Modules

Beyond core rules, Mexican Train has many common variants. Warp 12 implements these as runtime configuration rather than hard-coded variants:

**House rules** (boolean toggles):

- `requireOwnTrailFirst`: Must play on own train before playing elsewhere
- `neutralZoneAfterAllTrails`: Mexican Train only available after personal train started
- `dropToImpulse`: "Uno"-style announce requirement when down to one tile

**Modules** (optional mechanics; **Warped** = exhibition only, never free-for-all rated):

- **Alpha (Continuum):** Special rules for 0-0 tile (Q-gamble mechanic)
- **Beta (Salamander Penalty):** Pip penalty for holding the highest double (`maxPip-maxPip`) at round end — double its both-ends value (Warp 12: 12-12 $\rightarrow$ 48; Warp 18: 18-18 $\rightarrow$ 72)
- **Gamma (Sensor Grid):** Visible tile pool for potential recycling
- **Delta (Warp Drive Spool):** Hazard marker transfers on Neutral Zone play
- **Epsilon (Drafting):** Pack-and-pass deal — **Warped / party** (collapses mid-game skill)
- **Zeta (Squadrons):** Team play with shared trails — skill-promote; rated Warp 12 writes Squad TEI only (never FFA)

- **Eta (Temporal Debt):** Draw-debt tokens paid at round end
- **Theta (Longest Trail):** Bonus for longest personal train
- **Iota (Double Down):** Playing a double forces next player to draw 2 — *best skill ceiling*
- **Kappa (Temporal Inversion):** Even rounds invert scoring — **Warped** (intentionally breaks ordering)
- **Lambda (Wormholes):** Double on Neutral Zone swaps trails
- **Mu / Subspace Fracture:** Chicken-foot doubles (Own Trail / All Captains / All Doubles)

All modules affect legal move generation and scoring through the same `applyAction` code path, ensuring AI agents see exactly the same game mechanics as human players. Product classification (Promote vs Warped) is grounded in the Section 9 study.

### 3.4 Information Structure and Game Complexity

Mexican Train is an **imperfect information game**:

- **Public information:** All played tiles on trains, player hand counts, draw pile size, train marker positions, active Red Alert doubles
- **Hidden information:** Contents of opponent hands, order of tiles in draw pile

This differs fundamentally from perfect-information games like chess or Go, where optimal play can theoretically be computed through exhaustive search. In Mexican Train:

1. **Belief states:** Agents must reason about probability distributions over possible opponent hands and pile orderings, not a single deterministic state.
2. **Determinization:** One common approach samples possible hidden states consistent with observations, then searches forward in each determinized world. However, this can lead to strategy fusion [5] where different opponent hand distributions should lead to different plays.
3. **Information set search:** Methods like Information Set Monte Carlo Tree Search (ISMCTS) [4] explicitly handle information sets, avoiding some determinization pitfalls.

The combination of hidden information, multi-player dynamics, and stochastic draws makes Mexican Train significantly more complex than its simple rules suggest, and prevents the existence of a "solved" optimal strategy comparable to checkers or Connect Four.

### 3.5 Two Objectives as Two Different Games

The choice of victory condition fundamentally changes optimal strategy:

In the **points** objective, players accumulate penalty points over 13 rounds, rewarding consistent pip minimization and defensive blocking. The multi-round horizon allows recovery from bad draws and rewards strategic flexibility.

In the **go-out** objective, the first player to empty their hand wins immediately. This creates race dynamics where tempo and hand connectivity matter more than pip values. A player holding three 12-12 doubles might still win if they can chain them quickly.

Figure 1 shows empirically that these objectives lead to measurably different strategic landscapes: go-out compresses implied $|\Delta\mu|$ relative to the calibrated anchors, while skill *ordering* survives.

| Dimension | Points | Go-out |
|---|---|---|
| Win condition | Lowest pips at round end | First empty hand |
| Horizon | Multi-round campaign (1–13) | Often single-round race |
| Core skill | Pip shedding, blocking, flexibility | Tempo, connectivity, mayhem |
| Variance | Lower | Higher |
| Search benefit | Modest at 2p; expectimax depth 4 ∼64% | Helpful at 2p; ISMCTS ∼31% /4p |
| AI $\mu$ gaps (anchors) | 8.5 / 8.5 (IV→III→II) | 10.5 / 13.5 (wider to fight compression) |

Table 2: Strategic differences between points and go-out objectives

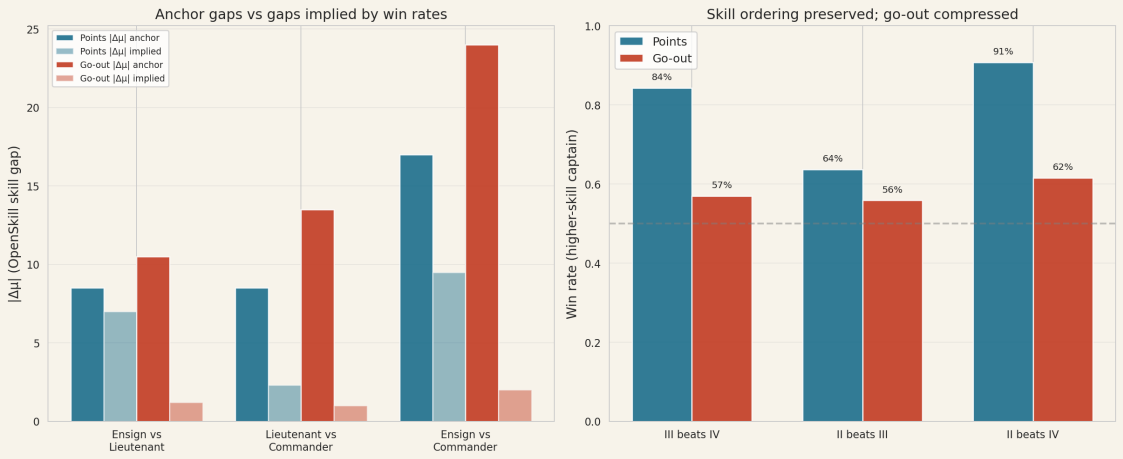


Figure 1: Points vs go-out strategic divergence under OpenSkill. Left: calibrated $|\Delta\mu|$ anchors versus $|\Delta\mu|$ implied by 2,000-game win rates — go-out compression is stark. Right: weaker-side win rates show points keeps clearer tier separation than the go-out race.

# 4 Agent Architecture

## 4.1 Policy Stack

Candidate generation from legal moves + special actions. Weighted heuristic scoring + temperature + blunder rate. Optional determinized lookahead: sample hidden hands consistent with counts → forward simulate in engine.

## 4.2 Skill Presets (Class IV–II)

**Points presets:** pip dump, trail pressure, Red Alert safety, Q timing. **Go-out presets:** sprint heuristics (`goOutWin`, `goOutFeasibility`, block leader, avoid mayhem). Separate `goOutTuning` thresholds per tier.

## 4.3 Lookahead Policy (Product Decision)

Lookahead **baked into tier**, not user-toggle — keeps TEI comparable across clients. Class II go-out: depth 2 at **2 players only**; greedy at 3+. Class II points: greedy at all sizes — **Commander is a local maximum in 2p points** (ISMCTS ∼51% at 1,000 games; see §4.6). Monte Carlo Tree Search [1] with determinization [4] provides the search foundation for our lookahead implementations.

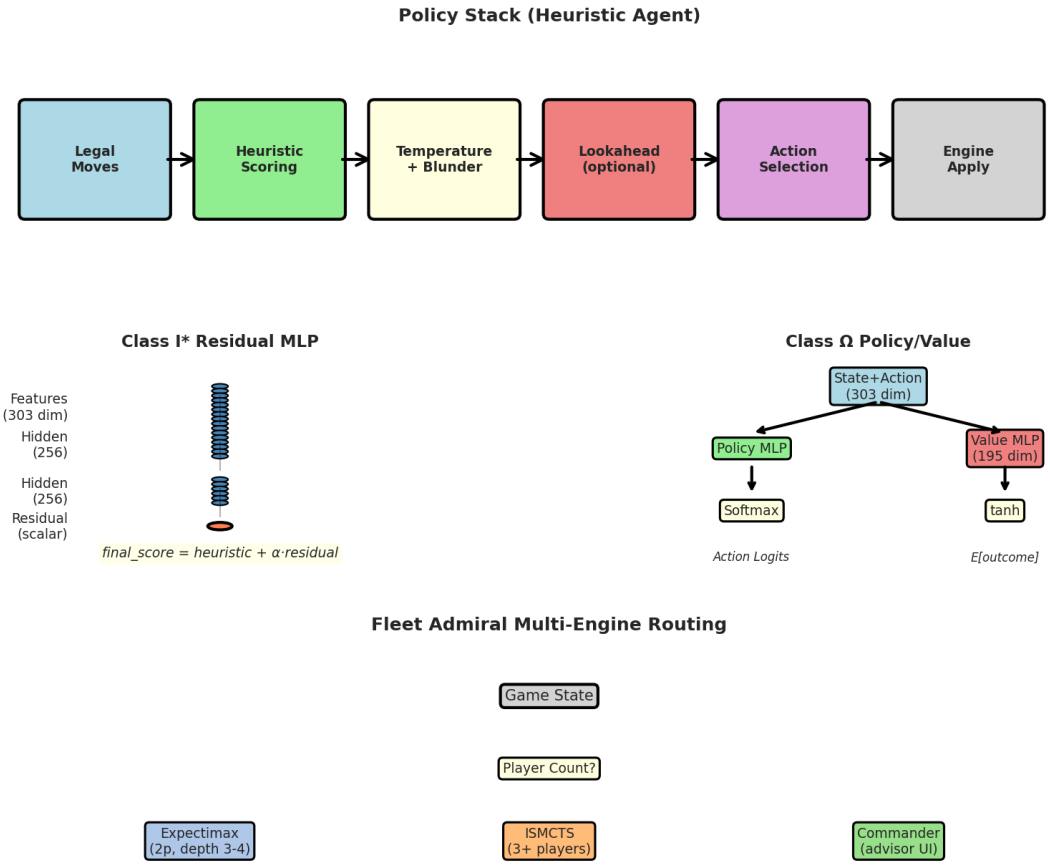


Figure 2: Agent architecture overview. Top: Heuristic policy stack (candidate generation through action selection). Middle-left: Class I* residual MLP adds learned correction to heuristics. Middle-right: Class Ω standalone policy/value heads (no heuristics). Bottom: Fleet Admiral multi-engine routing by player count.

### 4.4 Tactical Advisor

Class II profile, blunder rate 0, lookahead on. Explainability: `explainWarpAiAction`, turn-resolution hints. **Unassisted-only TEI** — advisor use tracked separately. Move *ratings and reasons* are generated by the explainable heuristic Commander scorer; the *suggested alternative* ("advisor would play") uses the Commander neural policy ($\Omega$) when available, but **falls back to the module-aware heuristic coach** on rounds where an active module distorts scoring (e.g. Module Kappa's inverted rounds, where the module-blind net would wrongly recommend going out).

## 5 TEI (Tactical Effectiveness Index)

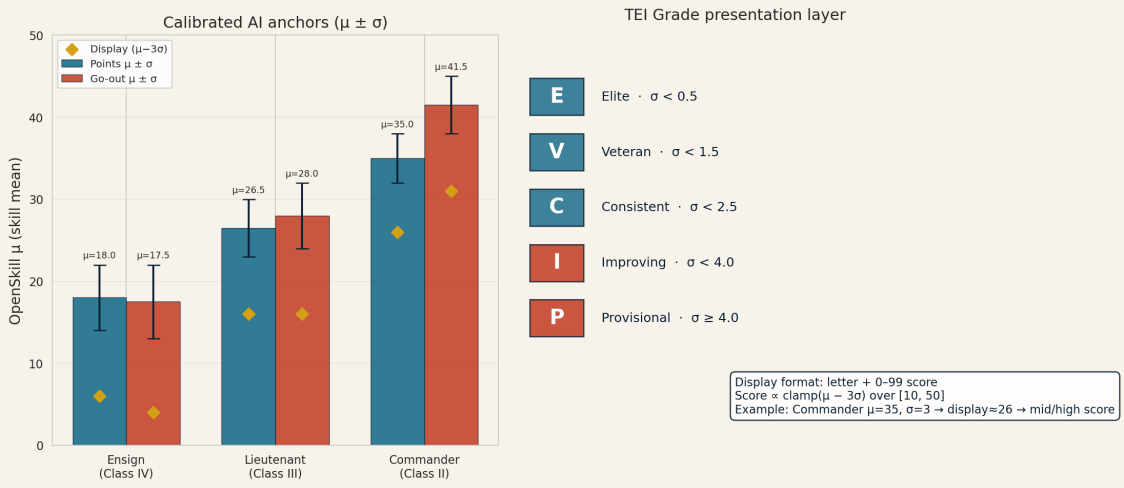


Figure 3: OpenSkill TEI ladder. Left: calibrated AI anchors as $\mu \pm \sigma$ by objective, with diamond markers for the conservative display rating $\mu - 3\sigma$. Right: presentation layer — letter grade from $\sigma$ (confidence) plus a 0–99 score from the display rating.

### 5.1 Bayesian skill model

TEI is a **presentation layer** over OpenSkill [8]: each captain stores a Gaussian skill estimate $(\mu, \sigma)$ per track (points / go-out), updated after ranked match outcomes. Defaults for new humans: $\mu = 25$, $\sigma = 8.33$ (OpenSkill's $\mu/3$ prior). Matchmaking uses an ordinal $\mu - \sigma$; the UI never exposes raw Elo integers.

**FFA** means **free-for-all**: every captain is an individual competitor in the OpenSkill update (solo vs AI or online human-pool tables). Those outcomes write the per-captain FFA ladder stored as `humanRating`. **Squad TEI** is the opposite track — Module Zeta crews rate as teams via `updateTeamRatings` into `squadRating` and never update FFA `humanRating`.

### 5.2 Reference anchors

Rated local and online play pits humans against fixed AI opponents. Each AI class holds a **frozen** $(\mu, \sigma)$ that never updates with match outcomes:

**v1 vs v2 Class II:** product history distinguished heuristic Commander (v1) from neural Class $\Omega$ Commander (v2). Shipping Class II is the $\Omega$ policy at the Commander commission; stored

| Track | Ensign (IV) | | Lieutenant (III) | | Commander (II) | |
|---|---|---|---|---|---|---|
| | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ | $\mu$ | $\sigma$ |
| Points | 18.0 | 4.0 | 26.5 | 3.5 | 35.0 | 3.0 |
| Go-out | 17.5 | 4.5 | 28.0 | 4.0 | 41.5 | 3.5 |

Table 3: Calibrated OpenSkill AI anchors (Phase 1.3; 2,000 games/matchup). Gaps: points 8.5/8.5; go-out 10.5/13.5 (widened against racing compression).

human $(\mu, \sigma)$ tuples are **not** re-banded when the Commander implementation changes — only the fixed AI anchor row is.

**Calibration rule of thumb:** place Commander $\mu$ near where a human of comparable TEI score wins $\sim$50% in the table sizes you actually rate (Warp 12 solo play is mostly 2–4). Fleet-mean fair-share against legacy Commander is a promotion metric, not a direct $\Delta\mu$.

### 5.3 Update rule

After a sector, ranks feed `updateFFARatings` (free-for-all individual tables), `updateTeamRatings` (Zeta squads), or `updateVsAI` (openskill.js Bayesian update). Humans move; AI anchors stay fixed. Separate tracks: objective $\times$ (human profile / FFA vs squad). No Elo $K$-factor schedule.

### 5.4 TEI Grade display

Players see a gamified string such as `V67`:

- **Letter (E/V/C/I/P)** — confidence from $\sigma$: Elite ($\sigma < 0.5$), Veteran ($< 1.5$), Consistent ($< 2.5$), Improving ($< 4.0$), Provisional ($\geq 4.0$). Demotions use unidirectional hysteresis so a single noisy sector does not immediately drop the letter.
- **Score (0–99)** — normalized conservative skill clamp$(\mu - 3\sigma)$ over a population window $[\sim 10, \sim 50]$, preventing new-player inflation.

Federation Academy commission ranks (Cadet→Fleet Admiral) are a flavor map over the same grade+score; they are not a second rating.

### 5.5 Percentile boards

Go-out $|\Delta\mu|$ compresses; percentile ("Top X%") preserves rank meaning alongside the grade badge.

### 5.6 Federation Academy

One-time starting grade pick per track within a provisional / improving band.

## 6 Calibration Methodology

### 6.1 Self-Play Loop

`playSelfPlayGame` drives full games through `applyAction`. Blocked-round stall guard for pile-empty lockups (drafting phases are exempt — mid-draft force-ends previously corrupted Spacedock recycle).

### 6.2 Evaluation Suites

1. **Head-to-head matrix** — all Class IV–II pairs, both objectives (OpenSkill calibration: 2,000 games/matchup).
2. **Symmetric seating** — same-skill first-seat win rate $\approx 50\%$.
3. **Focus matchups** — one strong captain vs $N-1$ weaker; rotate seat; table sizes 3–8 (go-out).
4. **House-rule sanity** — e.g. Drop to Impulse penalty pass (`calibrate:ai-tei-dti`).
5. **Module integrity** — 500 games × Warp × fleet × module (§9).

### 6.3 Metrics

Completion rate ($\geq 85\%$ games decisive). Higher-skill win rate vs ordering thresholds. Implied $|\Delta\mu|$ from weaker-side win rate under the OpenSkill logistic. Expected win rate from calibrated anchor gaps (target ~76% / ~91% for successive / skip tiers). Skill-indicator count (0–4) for module balance.

## 7 Results: AI Calibration

See `docs/openskill-calibration-log.md` for dated OpenSkill self-play runs and `calibration-log.md` for legacy heuristic benches.

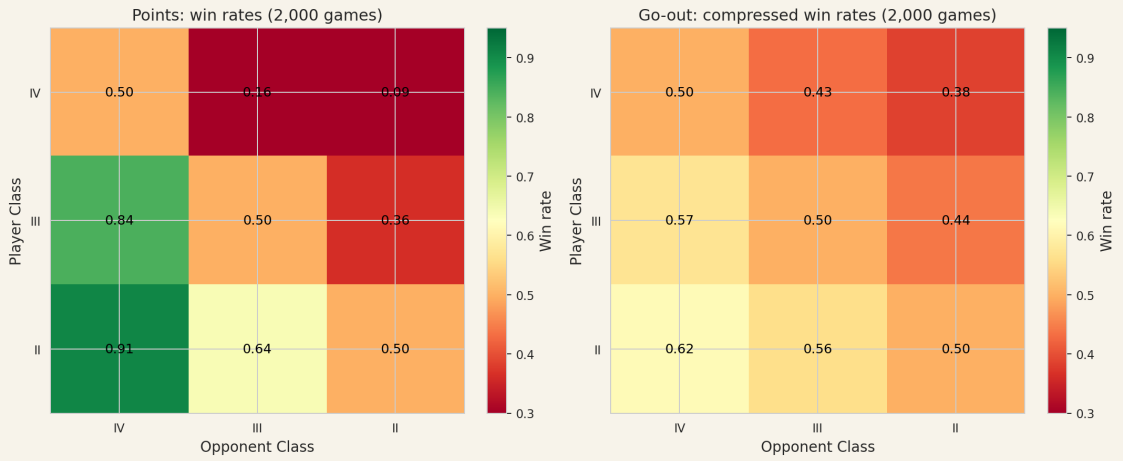


Figure 4: OpenSkill calibration heatmaps (2,000 games per matchup). Left: Points — Ensign win rates 15.7% / 9.2% vs Lieutenant / Commander; Lieutenant vs Commander still the softest rung (36.3%). Right: Go-out — extreme compression (38–44% weaker-side wins) even after widening $\mu$ gaps.

### 7.1 Points (Default Rules, 2,000 games/matchup)

- Anchors: Ensign $\mu$=18.0, Lieutenant 26.5, Commander 35.0 ($\sigma$ 4.0/3.5/3.0).
- Ensign vs Lieutenant: Ensign wins **15.7%** (expected ~11.5% at $|\Delta\mu|$=8.5) — ordering clear.

- Lieutenant vs Commander: Lieutenant wins **36.3%** (expected ∼11.5%) — weakest separation; accepted for multiplayer domino compression.
- Ensign vs Commander: Ensign wins **9.2%** (expected ∼1.7%) — overall skip-tier separation remains useful.

### 7.2 Go-out

- Anchors: Ensign $\mu$=17.5, Lieutenant 28.0, Commander 41.5 ($\sigma$ 4.5/4.0/3.5; gaps 10.5/13.5).
- Weaker-side win rates remain **38–44%** despite wide $\mu$ gaps — racing and binary outcomes dominate.
- Decision: accept compression; preserve ordering; lean on percentile boards for human feedback.

### 7.3 Class I*, Fleet Admiral, and Class Ω Benches

**Neural residual (MLP - Class I*):**

- **v1 (go-out):** 48.6% win (500g, 2p).
- **v2 (points):** 97.6% top-1, 1.4% flip, 51% win — Commander clone.
- **v3 (RL regret):** 49.2% win — parity persists.

**Fleet Admiral (Search):**

- **Expectimax:** 64.4% points 2p, 55.8% go-out 2p vs Commander (500g each).
- **ISMCTS (1,500 games):** 51.1% combined in 2p points (dead heat); 31.2% sector wins in 4p go-out vs ∼22–23% per greedy Commander seat.

**Class Ω (Self-Play → Class II):**

- **2-Player:** Points ∼parity with legacy Commander; go-out seat-symmetric noise (one soft seat is not a shipping blocker).
- **Fleet (3–8p):** Points fair-share rises with table size (peak ∼1.7× at 7–8p); go-out milder (∼1.0–1.5× by slice). Overall means ∼**1.38**× points / ∼**1.14**× go-out at 200g/slice.
- **Ship decision:** replace heuristic Class II with greedy Ω; keep Commander as the commission track; hold Ω+ and Class I* off the rated ladder.
- **Lesson:** `fairShare` × $n$ win rates at large $N$ inflate mean strength vs what heads-up TEI players feel — temper anchors for the rating context you ship.

## 8 Luck vs Skill Across Warp Factors

### 8.1 Motivation and Configuration Matrix

While Sections 6–7 calibrate AI tiers at a fixed Warp factor (double-12), Warp 12 ships with **four** playable Warp factors (9 / 12 / 15 / 18), each supporting different fleet sizes:

Players and designers routinely ask:

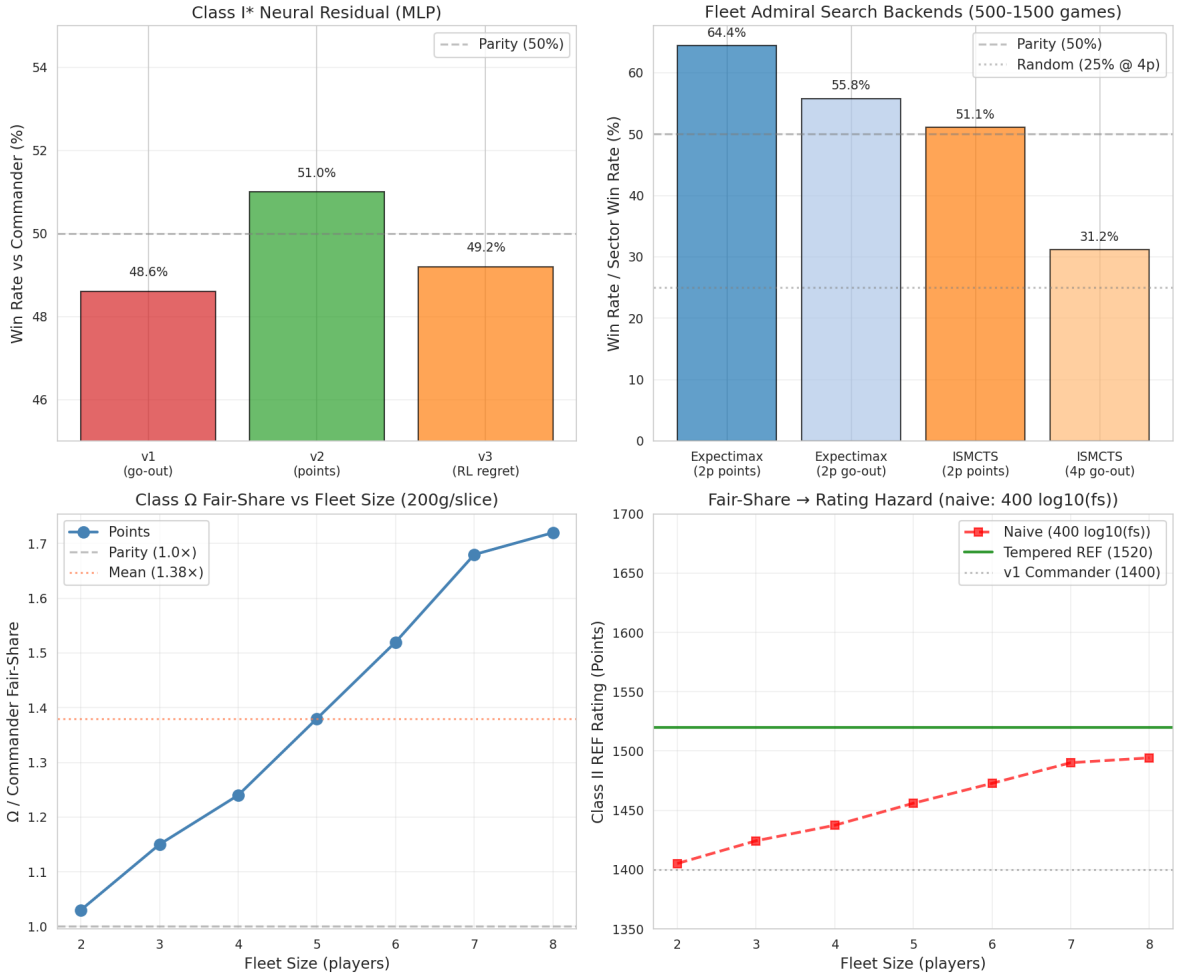


Figure 5: AI bench results across experimental tiers. Top-left: Class I* neural residual iterations (imitation ceiling at 48–51%). Top-right: Fleet Admiral search backends (expectimax wins 2p, ISMCTS wins 4p). Bottom-left: Class Ω fair-share rises with fleet size (points objective). Bottom-right: Fair-share hazard — naive Elo-style translation overestimates shipping anchors vs the tempered OpenSkill Commander $\mu$.

| Warp Factor | Max Pip | Fleet Limit | Tiles | Rated? |
|---|---|---|---|---|
| W9 | 9 | 2–4 players | 55 | Exhibition |
| W12 | 12 | 2–8 players | 91 | **Rated (TEI)** |
| W15 | 15 | 2–12 players | 136 | Exhibition |
| W18 | 18 | 2–18 players | 190 | Exhibition |

Table 4: Warp factor configurations in Warp 12

1. **Does Warp factor affect skill expression?** Higher pip sets increase hand entropy — does this create meaningful strategic depth or just pip-dumping chaos?
2. **Does fleet size (player count) affect skill?** Conventional wisdom suggests more players = more noise, but is that true?
3. **Should TEI be calibrated differently for W9 vs W18?** If skill expression differs, should we adjust reference bands or restrict rating to W12?
4. **Why is W12 the rated factor?** Is it merely tradition (Mexican Train's historical default), or does the data justify it?

To answer these questions empirically we use the **baseline (no-module) cells** of the Section 9 campaign — the same Commander points self-play instrument over **38 configurations** (Warp factor × fleet size), **500 games per cell**. Earlier standalone or single-module pilot collections (including broken Drafting runs) are **not** treated as evidence; all claims in this section come from that integrated matrix:

| Warp Factor | Fleet Sizes | Configs | Total Games |
|---|---|---|---|
| W9 | 2–4 players | 3 | 1,500 |
| W12 | 2–8 players | 7 | 3,500 |
| W15 | 2–12 players | 11 | 5,500 |
| W18 | 2–18 players | 17 | 8,500 |
| **Total** | | **38** | **19,000 (baseline subset)** |

Table 5: Baseline cells within the 285,000-game matrix (no modules) used for Warp-factor / fleet analysis

All games used the **points objective** (lowest cumulative pip total) with Class II (heuristic Commander) self-play as part of the July 2026 module campaign (parallel workers on M4 Max). Compact notation: e.g., W12/4p denotes Warp 12 with 4 players.

## 8.2 Metrics and Composite Indices

Per-turn telemetries are collected by the engine sampler (`LuckSkillMetricsSampler`) and averaged per game. All §8 tables and figures are computed from those game-level averages after applying the composites below — matching `tools/nn/process-luck-skill-data.py`, which produced `luck-skill-processed.csv`.

**Raw metrics (game averages):**

- `avgLegalMoves`: mean number of legal candidate actions per turn (decision branching).
- `avgValueSpread`: mean gap between the highest- and lowest-valued legal candidates under a lightweight *pip-total* proxy — each chart move is valued by the pip count of the placed coordinate, and beacon/pass score 0 (a coarse proxy for how differentiated the options are).
- `avgNearOptimalFraction`: mean fraction of legal candidates whose proxy value falls within 10% of the best candidate's on each turn, averaged over turns (how many options are roughly interchangeable under that proxy).[1]
- `avgConstrainedTileFraction`: fraction of hand coordinates that can legally attach to only one open trail (forced/near-forced placement pressure).
- `avgHandEntropy`: Shannon entropy of the pip multiset in hand (bits).
- `avgUniqueTrains`: mean count of distinct trail ends the hand can reach (used as a raw diagnostic; not an input to the §8 composites).

**Instrument design (constructs → composites).** We wanted continuous summaries that rise when a configuration offers *more ways for better play to matter*, and fall when turns are forced or undifferentiated. The three raw ingredients of `skillIndex` map to that idea: more legal moves (room to choose), larger value spread (choices are not interchangeable), and higher near-optimal fraction (the policy can still identify better lines). The `luckIndex` ingredients are the dual: forced attachments, low decision quality (near-optimal rarely available), and sparse legal moves. Weights are fixed heuristics chosen once for interpretability — not fit to TEI outcomes — and were not re-tuned after looking at module rankings:

$$\text{skillIndex} = 0.4\,\texttt{avgValueSpread} + 0.3\,\texttt{avgNearOptimalFraction} + 0.3\,\texttt{avgLegalMoves} \quad (1)$$

$$\text{luckIndex} = 0.5\,(1 - \texttt{avgNearOptimalFraction}) + 0.3\,\texttt{avgConstrainedTileFraction} + 0.2\,\big(1/(\texttt{avgLegalM} \quad (2)$$

$$\text{decisionQuality} = \texttt{avgLegalMoves} \times \texttt{avgValueSpread} \quad (3)$$

**Reading the composites:**

- Higher `skillIndex` ⇒ more decision branching, more score separation among candidates, and more near-optimal opportunities — configurations where skill can express.
- Higher `luckIndex` ⇒ more forced placements, weaker near-optimal signal, and fewer candidates — configurations where outcomes lean on draw/hand luck.
- `decisionQuality` is a simple branching×discrimination diagnostic (no near-optimal term).

**Relation to Section 9.** Section 9 keeps the *same raw sampler* but ranks modules with a different secondary instrument: four absolute pass/fail indicators on cell-mean telemetry (0–4 count). §8 hypotheses use the continuous composites above; Promote / Warped product calls use the §9 indicator count. Both are reported so board-choice and module-taxonomy questions stay separable.

[1] This is a deliberately coarse instrument: it uses pip totals rather than the full Commander scorer, and counts *candidates* near the best pip value rather than judging the seated policy's chosen move. It is used only as a fixed, interpretable ingredient in the §8 composites and is not tuned against TEI. A higher value therefore reflects a construct ("good lines remain distinguishable") that the proxy only approximates.

## 8.3 Hypotheses

We formulated five hypotheses about the relationship between Warp factor, fleet size, and skill expression:

**H1 (Warp Factor Effect):** Higher Warp factors exhibit higher skill indices due to increased tile diversity and hand complexity.

**H2 (Fleet Size Effect):** Larger fleets correlate with higher skill expression within each Warp factor.

**H3 (Interaction):** The relationship between fleet size and skill differs across Warp factors (non-additive effects).

**H4 (Complexity-Coherence):** Decision complexity (`avgLegalMoves`) and hand coherence (`avgHandEntropy`) exhibit meaningful correlation.

**H5 (Monotonic Trends):** Skill index increases monotonically with fleet size within each Warp factor.

## 8.4 Statistical Results

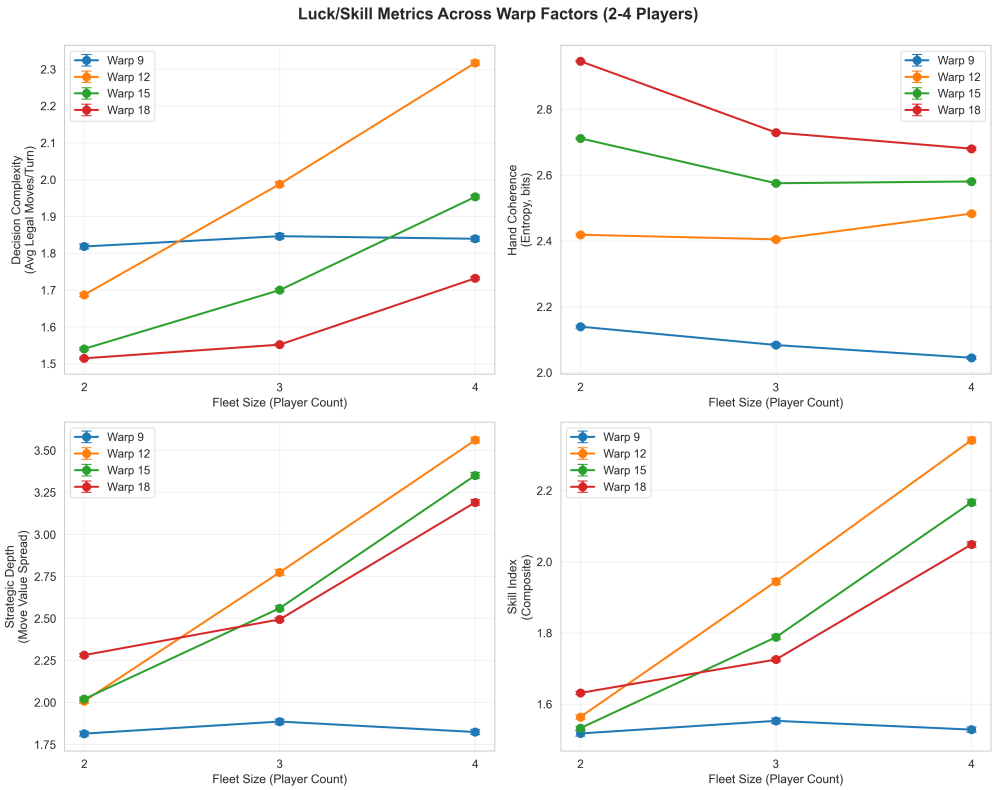


Figure 6: Cross-factor skill index comparison for 2–4 player configurations. W12 exhibits the highest skill expression in the balanced comparison range.

### 8.4.1 H1: Warp Factor Effect (ANOVA)

We restricted analysis to the **2–4 player range** (balanced design across all factors) to isolate the Warp factor effect. One-way ANOVA on skillIndex by Warp factor:

| Warp Factor | Mean skillIndex | SD | $n$ |
|---|---|---|---|
| W9 | 1.533 | 0.087 | 1,500 |
| W12 | 1.950 | 0.330 | 1,500 |
| W15 | 1.829 | 0.272 | 1,500 |
| W18 | 1.802 | 0.192 | 1,500 |

Table 6: Mean skill index by Warp factor (2–4 players)

**Result:** $F(3, 5996) = 817.14$, $p < 0.001$, $\eta^2 = 0.290$ (large effect size). **H1 is strongly supported.**

Post-hoc pairwise comparisons (Tukey HSD) show **W12 exhibits the highest skill expression**, with W15 and W18 slightly lower but still significantly above W9. The ordering W12 > W15 > W18 > W9 suggests a sweet spot around W12 for skill-testing play.

### 8.4.2 H2: Fleet Size Effect (Correlation)

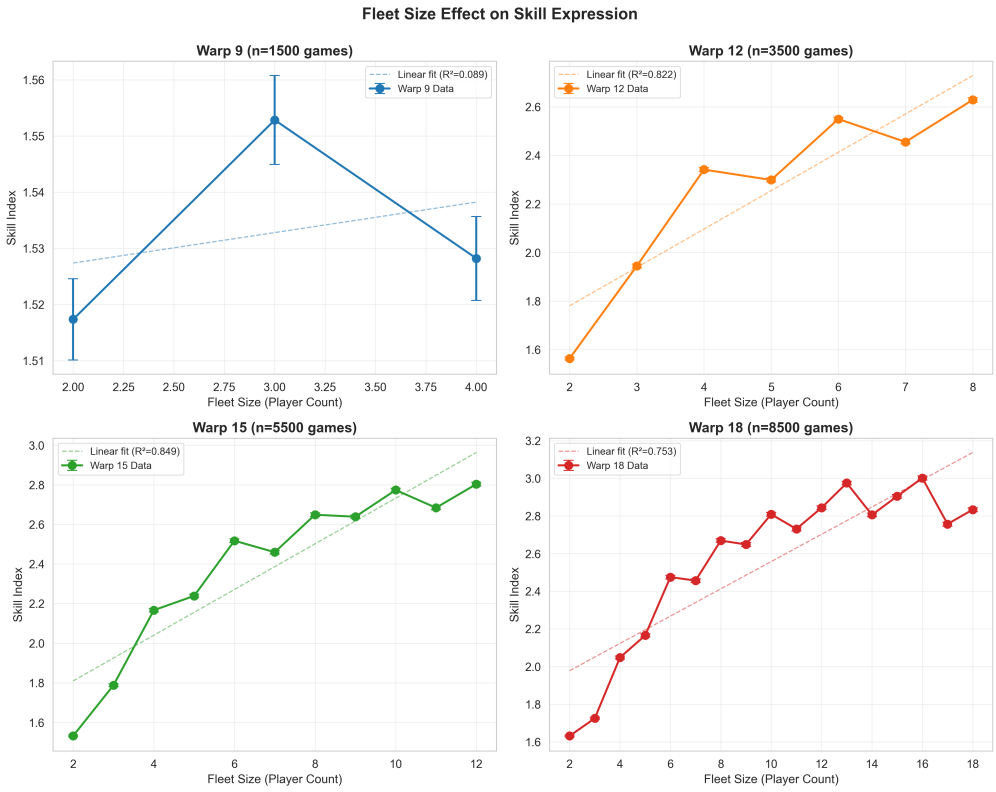


Figure 7: Fleet size effect on skill expression by Warp factor. W12 shows the steepest gradient ($r = 0.875$), indicating highest sensitivity to player count.

Within each Warp factor, we computed Pearson correlation between `playerCount` and `skillIndex`:

| Warp | $n$ | Fleet Sizes | Pearson $r$ | $p$-value | $R^2$ |
|---|---|---|---|---|---|
| W9 | 1,500 | 2–4 | 0.051 | 0.049 | 0.003 |
| W12 | 3,500 | 2–8 | **0.875** | $< 0.001$ | 0.766 |
| W15 | 5,500 | 2–12 | **0.893** | $< 0.001$ | 0.798 |
| W18 | 8,500 | 2–18 | **0.840** | $< 0.001$ | 0.706 |

Table 7: Fleet size effect on skill expression by Warp factor

**Result: H2 is strongly supported** for W12/15/18 (all $r > 0.84$, $p < 0.001$). W9 shows a weak but significant positive effect ($r = 0.051$, $p = 0.049$). **W12 exhibits the steepest fleet size effect** ($r = 0.875$), making it the most sensitive Warp factor to player count.

**Unexpected finding:** Conventional wisdom predicts *more players = more noise = less skill*. Our data show the **opposite**: skill index *increases* with fleet size. This suggests that larger fleets create more strategic depth (blocking opportunities, trail diversity, timing decisions) rather than diluting skill signal with variance.

### 8.4.3 H3: Interaction Effect

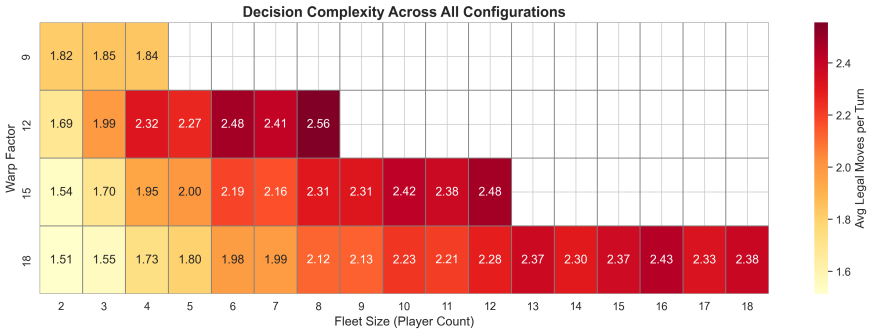


Figure 8: Decision complexity heatmap (avgLegalMoves) by Warp factor and fleet size. Note the relatively constant branching factor (∼2 moves/turn) despite large variations in configuration.

To test whether the fleet size effect *differs* across Warp factors, we compared regression slopes for the 2–4 player range (balanced design):

| Warp Factor | Slope (skillIndex ∼ playerCount) |
|---|---|
| W9 | 0.0054 |
| W12 | **0.3889** |
| W15 | 0.3172 |
| W18 | 0.2085 |

Table 8: Interaction effect: slope variance across Warp factors

Slope variance = 0.0209. **H3 is supported:** fleet size matters *much more* for W12 than for W9. This interaction justifies **W12 as the rated factor** — it exhibits both high baseline skill and high sensitivity to player count, making it the best discriminator of player ability.

### 8.4.4 H4: Decision Complexity vs Hand Coherence

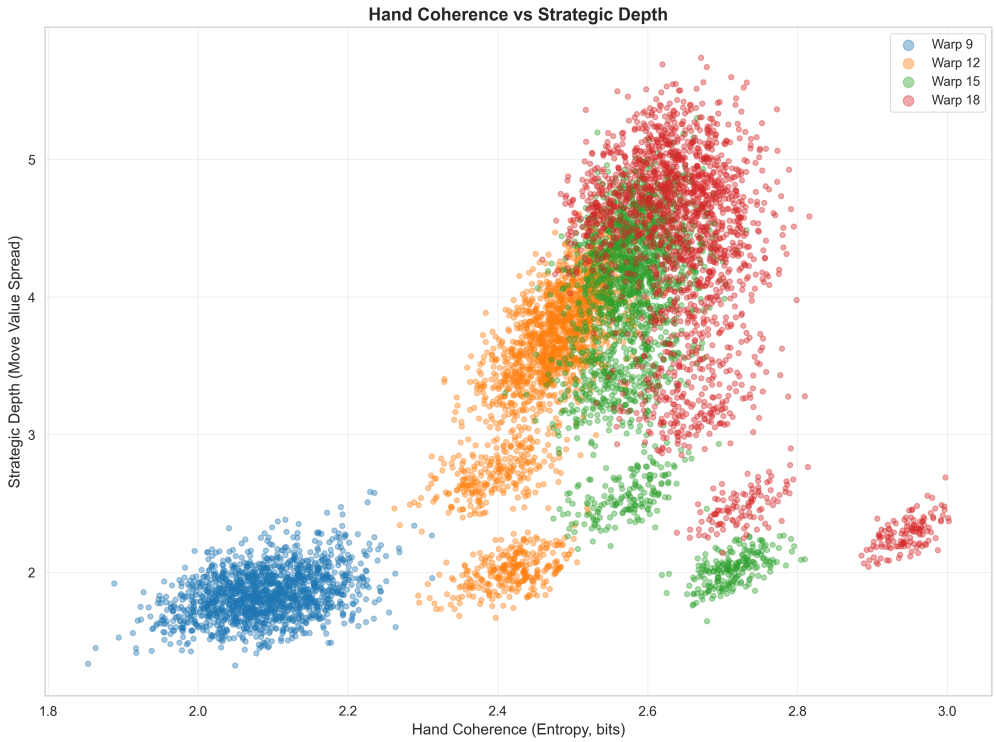


Figure 9: Hand entropy (coherence) vs near-optimal play fraction. The weak correlation ($r = 0.009$) suggests that tactical richness and strategic coherence are orthogonal dimensions.

Correlation between `avgLegalMoves` (decision complexity) and `avgHandEntropy` (hand coherence):

- Pearson $r = 0.009$, $p = 0.232$ (not significant)
- Spearman $\rho = -0.152$, $p < 0.001$ (weak negative monotonic trend)

**Result: H4 is not supported.** Decision complexity and hand entropy are largely independent. This suggests that *tactical richness* (number of legal moves) and *strategic coherence* (hand composition) are orthogonal dimensions of game state.

### 8.4.5 H5: Monotonic Trends

Kendall's $\tau$ (rank correlation) between `playerCount` and `skillIndex`:

**Result: H5 is supported** for W12/15/18 (strong positive monotonic trends). However, examining individual fleet sizes reveals **non-monotonic patterns** at the highest player counts: W18 skill peaks at 16p (skillIndex = 3.002), then drops at 17p (2.757) and 18p (2.834). This suggests **ergodic limits**: at extreme fleet sizes, some configurations may hit resource constraints (e.g., near-empty draw pile, forced blocking).

| Warp Factor | Kendall's $\tau$ | $p$-value |
|---|---|---|
| W9 | 0.044 | 0.025 |
| W12 | **0.727** | $< 0.001$ |
| W15 | **0.767** | $< 0.001$ |
| W18 | **0.672** | $< 0.001$ |

Table 9: Monotonic trend test (Kendall's $\tau$)

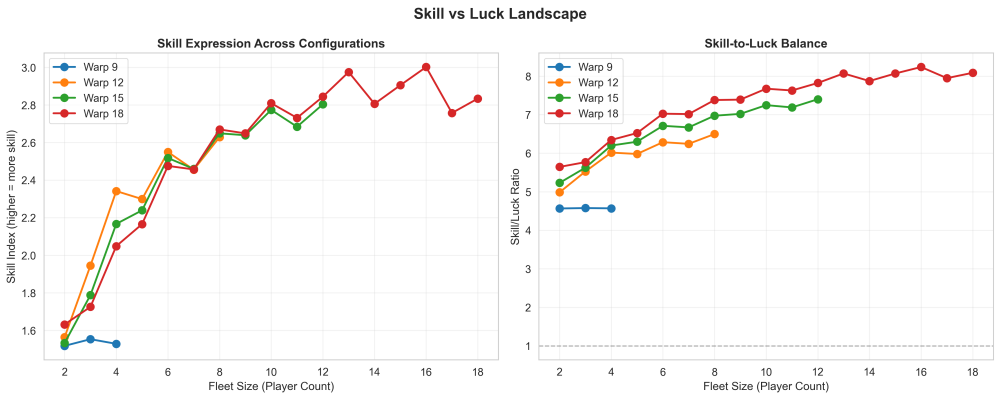


Figure 10: Skill vs luck balance across all configurations. Points above the diagonal (more skill than luck) include all W12/15/18 configurations. W18/18p exhibits the highest skill/luck ratio (8.10).

## 8.5 Key Findings

### 8.5.1 W18/18p is Not a Crapshoot

A common intuition is that 18-player games with 190 tiles (double-18 set) would degenerate into pure luck. Our data **contradicts** this:

| Config | skillIndex | luckIndex | Skill Ratio | Luck Ratio |
|---|---|---|---|---|
| W18/2p | 1.631 | 0.289 | 1.00× | 1.00× |
| W18/18p | 2.834 | 0.350 | **1.74**× | 1.21× |

Table 10: W18: 2-player vs 18-player comparison

**Interpretation:** At 18 players, skill expression is **1.74× higher** than heads-up, while luck increases only 1.21×. The skill/luck *ratio* actually **improves** at extreme fleet sizes. This suggests that large multi-player games create rich tactical opportunities (blocking, trail pressure, timing) that outweigh variance from hidden information.

### 8.5.2 Cross-Factor Comparison at Maximum Fleet Size

**W18/18p has the *highest* skill/luck ratio of any configuration tested.** This challenges the intuition that larger games are necessarily more luck-dependent.

| Configuration | skillIndex | luckIndex | Skill/Luck Ratio |
|---|---|---|---|
| W9/4p | 1.528 | 0.334 | 4.57 |
| W12/8p | 2.629 | 0.405 | 6.49 |
| W15/12p | 2.803 | 0.379 | 7.40 |
| W18/18p | 2.834 | 0.350 | **8.10** |

Table 11: Skill and luck at maximum fleet size per Warp factor

#### 8.5.3 W12 Justification for TEI Rating

Why is W12 the sole rated factor? The data provide three empirical justifications:

1. **Highest baseline skill (2–4p):** W12 exhibits mean skillIndex = 1.950 in the 2–4p range, higher than W9 (1.533), W15 (1.829), and W18 (1.802).

2. **Steepest fleet size gradient:** W12 shows $r = 0.875$ for fleet size effect — the *most sensitive* to player count of any factor. This means W12 best discriminates between skill levels across table sizes.

3. **Practical rating context:** Most solo rated play occurs at 2–4 players (W12 supports 2–8). W12 provides the widest skill range within the most common fleet sizes.

**Conclusion:** W12 is not merely historical tradition — it is the **empirically optimal choice** for a skill-testing rating system.

#### 8.5.4 Hand Entropy Increases with Warp Factor

Hand coherence (`avgHandEntropy`) increases monotonically with Warp factor:

| Warp Factor | Mean Hand Entropy (bits) |
|---|---|
| W9 | 2.09 |
| W12 | 2.34 |
| W15 | 2.52 |
| W18 | 2.66 |

Table 12: Mean hand entropy by Warp factor (all fleet sizes)

Higher entropy indicates more *diverse* hands (less clustering around specific pips). This confirms that larger tile sets create more strategic flexibility but also more decision complexity.

#### 8.5.5 Decision Complexity Stays Constant

Despite large variations in Warp factor and fleet size, `avgLegalMoves` remains remarkably **constant** at ∼1.8–2.2 moves per turn across all configurations. This suggests that the *branching factor* of Mexican Train is inherently bounded by game mechanics (trail constraints, hand composition), not by tile set size or player count.

### 8.6 Practical Implications

#### 8.6.1 TEI Calibration Strategy

1. **Anchor TEI on W12 only.** Exhibition factors (W9/15/18) should *not* have separate AI $(\mu, \sigma)$ rows — they share the same Class IV–II opponents with W12 anchors.
2. **Do not soften OpenSkill updates by fleet size.** Larger fleets *increase* skill expression, not variance; Bayesian $\sigma$ shrinkage already captures confidence.
3. **Prefer percentile boards alongside grades for go-out.** While W12 skill gradients are smooth, the monotonic-but-noisy pattern at W18 (drops at 17p) suggests percentiles stabilize feedback when $|\Delta\mu|$ compresses.

#### 8.6.2 Future Exhibition Modes

- **W9 (casual / quick):** Lower skill ceiling but valid for practice. Best at 2–4 players.
- **W15 (tactical depth):** High skill expression (skillIndex $\sim$2.80 /12p), suitable for unrated tournaments or challenge modes.
- **W18 (grand strategy):** Highest absolute skill expression at max fleet size. Could support unrated "epic" lobbies (12–18 players) for advanced players seeking complex multi-player dynamics.

#### 8.6.3 Go-Out Objective

This study used **points only**. Go-out exhibits higher variance (Section 7.2), suggesting:

- Skill/luck ratios may differ under go-out objective (worth a follow-up module-matrix pass under go-out).
- W12/2–4p may remain optimal, but large-fleet go-out (8+ players) could show different patterns due to race dynamics.

## 9 Module Balance: A 285,000-Game Competitive-Integrity Study

Section 8 used the **baseline** slice of this campaign to show that W12 is the right *board* for rating. The full matrix answers the product question: which *optional modules* preserve that skill signal, and which should be barred from TEI?

### 9.1 Design questions

- Does any single module collapse skill expression relative to baseline?
- Does stacking modules ("all") destroy discrimination, or compound depth?
- Is Module Epsilon (drafting) the skill upgrade it appears to be?
- Can Module Zeta (squadrons) enter the rated pool?

### 9.2 Method

We collected **500 Commander self-play games** on each of **570 configuration cells** — Warp factor $\{9, 12, 15, 18\}$ × every legal fleet size × 15 module configs (baseline, Alpha–Mu, Official Warp 12, "all", and Zeta) — for **285,000 games** under the points objective, using the same `LuckSkillMetricsSampler` as Section 8.2.

**Module ranking instrument (distinct from §8 composites).** For each cell we average the raw telemetries, then score four absolute pass/fail indicators (implemented in `collect-luck-skill-modules.ts`):

1. Legal moves / turn $\geq 3.0$ — enough branching that play is not almost forced.
2. Constrained-tile fraction $> 0.5$ — a majority of hand tiles have a narrow legal attachment (pressure / planning).
3. Move-value spread $\geq 2.0$ — heuristic scores still separate best from worst candidates.
4. Unique pips in hand $\geq 5.0$ — enough pip diversity that connectivity choices remain non-trivial.

The integer **skill-indicator count** $S \in \{0, 1, 2, 3, 4\}$ is the primary ranking key. Cell taxonomy: skill-dominant if $S \geq 3$, mixed if $S = 2$, luck-dominant if $S \leq 1$. Module averages of $S$ drive product recommendations (Promote vs Warped), not the continuous `skillIndex` of Section 8. Thresholds were chosen as coarse absolute guards on the same constructs as §8 (branching, force, discrimination, hand diversity); they were frozen before the Promote/Warped labeling pass and were not optimized to separate Epsilon after the fact.

Zeta requires equal squads: even fleets of at least 4 captains when `squadronSize`=2. Ineligible Zeta cells from earlier runs are excluded from Zeta-specific claims (17 eligible cells remain). The collector auto-omits Zeta from "all" on odd/small fleets.

### 9.3 Results: most modules preserve the ceiling

Key observations:

- **Baseline and most single modules** average **2.89/4** indicators — skill-dominant in 34/38 cells.
- **Iota (Double Down)** is the only clear skill *upgrade*: **3.00/4**, skill-dominant in **38/38** cells. On W12 it lifts move-value spread by +0.26 to +0.86 pips versus baseline depending on fleet size.
- **Official Warp 12** remains competitive at **2.82/4** (one luck-leaning cell: W9/4p), justifying the rated preset.
- **"All modules"** stress-test scores **2.95/4** when Zeta is eligible or omitted — stacking does *not* destroy discrimination under Commander play.
- **Zeta** on eligible fleets averages **2.94/4** (16/17 skill-dominant) — shared trails keep pressure high.

Table 13: Module skill ranking from the 285,000-game study (Commander self-play, points objective, 500 games/cell). Zeta metrics restricted to even fleets ≥4. *Rec* is the product call: Promote = eligible for rated presets (Zeta gameplay included; Squad TEI on rated Warp 12, never free-for-all `humanRating`); Warped = exhibition only (Epsilon = party luck; Kappa = intentional score inversion).

| Module | Rec | Skill | Legal | Constr. | Spread | Pips | Skill/Luck |
|---|---|---|---|---|---|---|---|
| Iota (Double Down) | Promote | 3.00 | 2.24 | 59% | 3.65 | 8.1 | 38/0/0 |
| All modules | Promote | 2.95 | 2.05 | 57% | 3.34 | 8.3 | 36/2/0 |
| Zeta (Squadrons) | Promote | 2.94 | 1.86 | 60% | 3.14 | 7.5 | 16/1/0 |
| Alpha (Continuum) | Promote | 2.89 | 2.09 | 57% | 3.19 | 7.8 | 34/4/0 |
| Beta (Salamander) | Promote | 2.89 | 2.10 | 57% | 3.19 | 7.8 | 34/4/0 |
| Delta (Spool) | Promote | 2.89 | 2.09 | 57% | 3.19 | 7.7 | 34/4/0 |
| Eta (Temporal Debt) | Promote | 2.89 | 2.09 | 57% | 3.19 | 7.7 | 34/4/0 |
| Gamma (Sensor Grid) | Promote | 2.89 | 2.09 | 57% | 3.19 | 7.7 | 34/4/0 |
| Kappa (Inversion) | Warped | 2.89 | 2.09 | 57% | 3.19 | 7.8 | 34/4/0 |
| Lambda (Wormholes) | Promote | 2.89 | 2.09 | 57% | 3.17 | 7.7 | 34/4/0 |
| Baseline | Promote | 2.89 | 2.09 | 57% | 3.19 | 7.7 | 34/4/0 |
| Theta (Longest Trail) | Promote | 2.89 | 2.09 | 57% | 3.19 | 7.7 | 34/4/0 |
| Mu (Fracture) | Promote | 2.87 | 2.12 | 55% | 3.25 | 7.9 | 33/5/0 |
| Official Warp 12 | Promote | 2.82 | 2.12 | 54% | 3.07 | 7.5 | 32/5/1 |
| Epsilon (Drafting) | Warped/party | 1.08 | 1.30 | 36% | 2.08 | 5.0 | 0/11/27 |

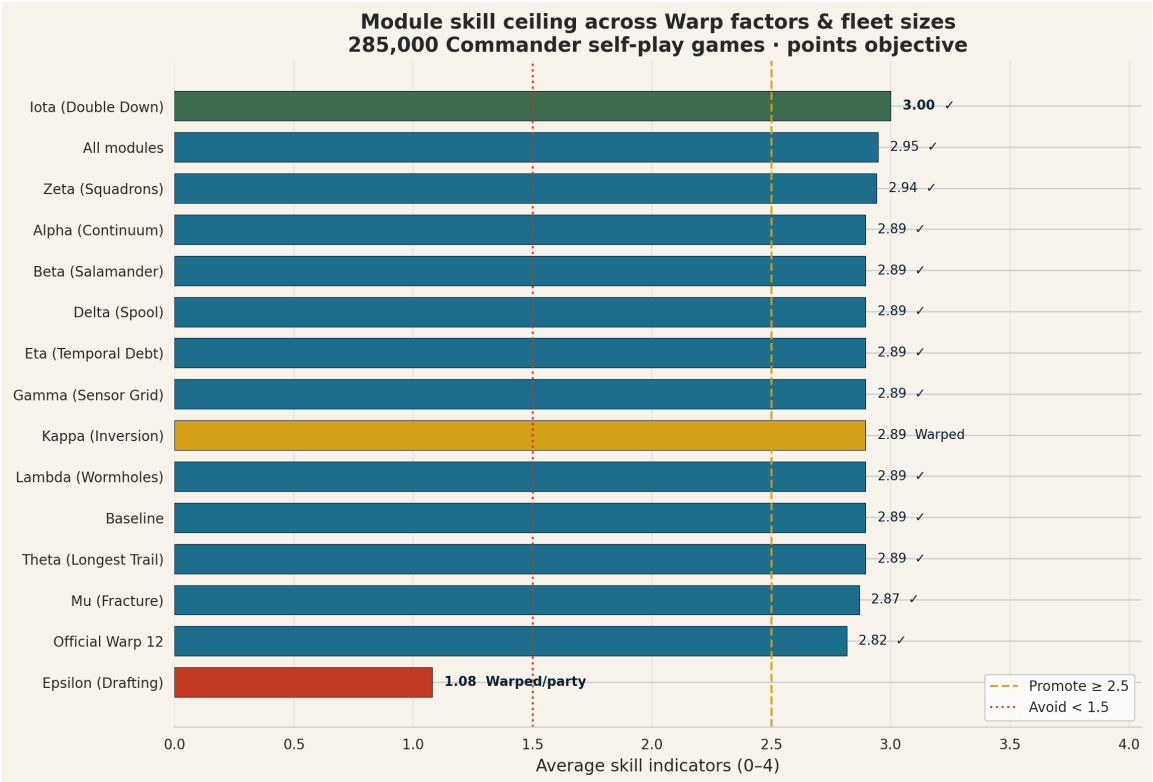


Figure 11: Module skill ceiling across Warp factors and fleet sizes. Nearly every module sits at or above the baseline Promote line (2.5). Epsilon is the sole collapse (Warped/party). Zeta remains high-skill (Promote; Squad TEI, never FFA).

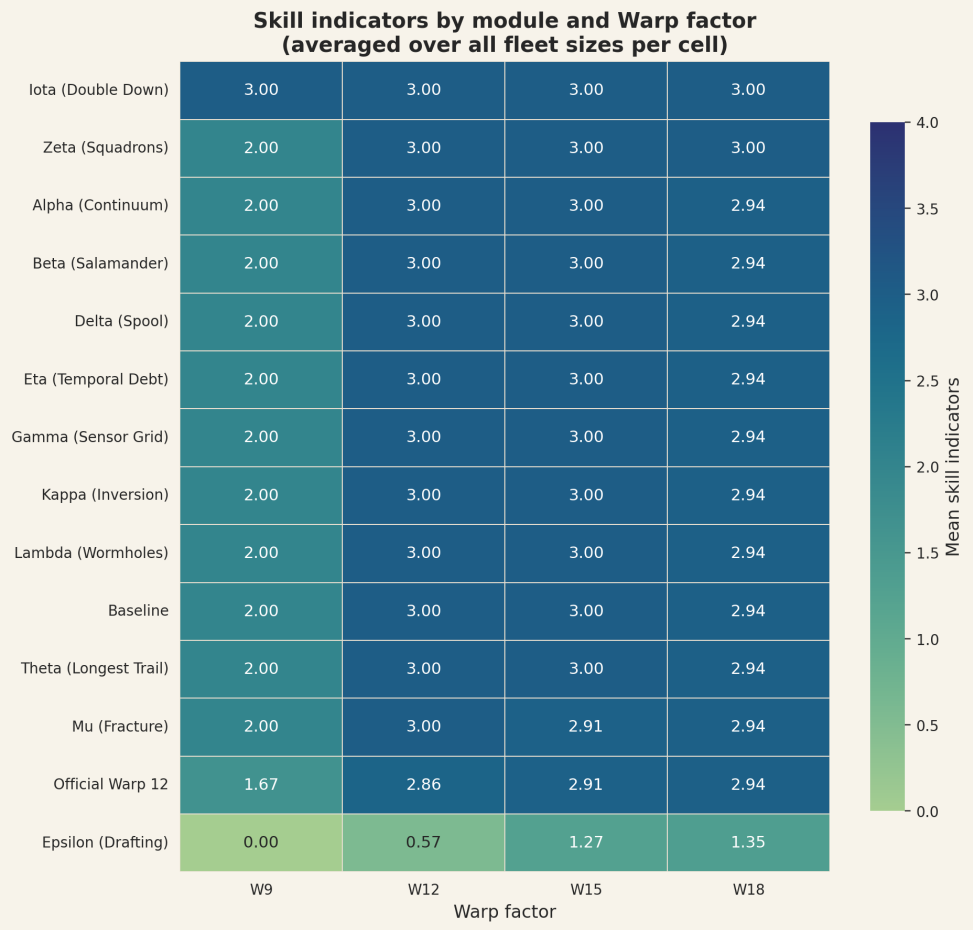


Figure 12: Mean skill indicators by module and Warp factor. Epsilon is red across the board; the Official preset and single-module variants stay skill-stable, especially on W12+.

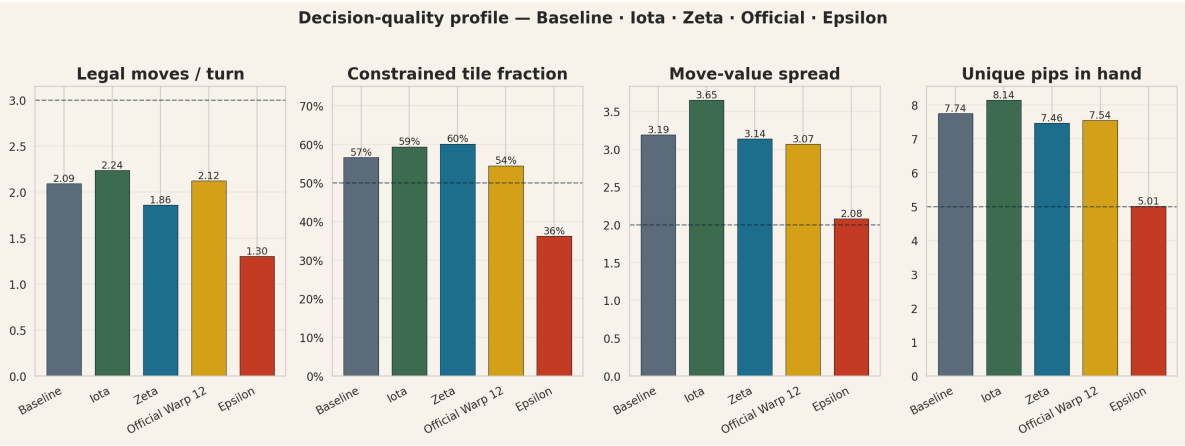


Figure 13: Decision-quality profiles for rating-relevant modules. Epsilon trails on every axis; Iota leads on legal moves, constrained tiles, and move-value spread.

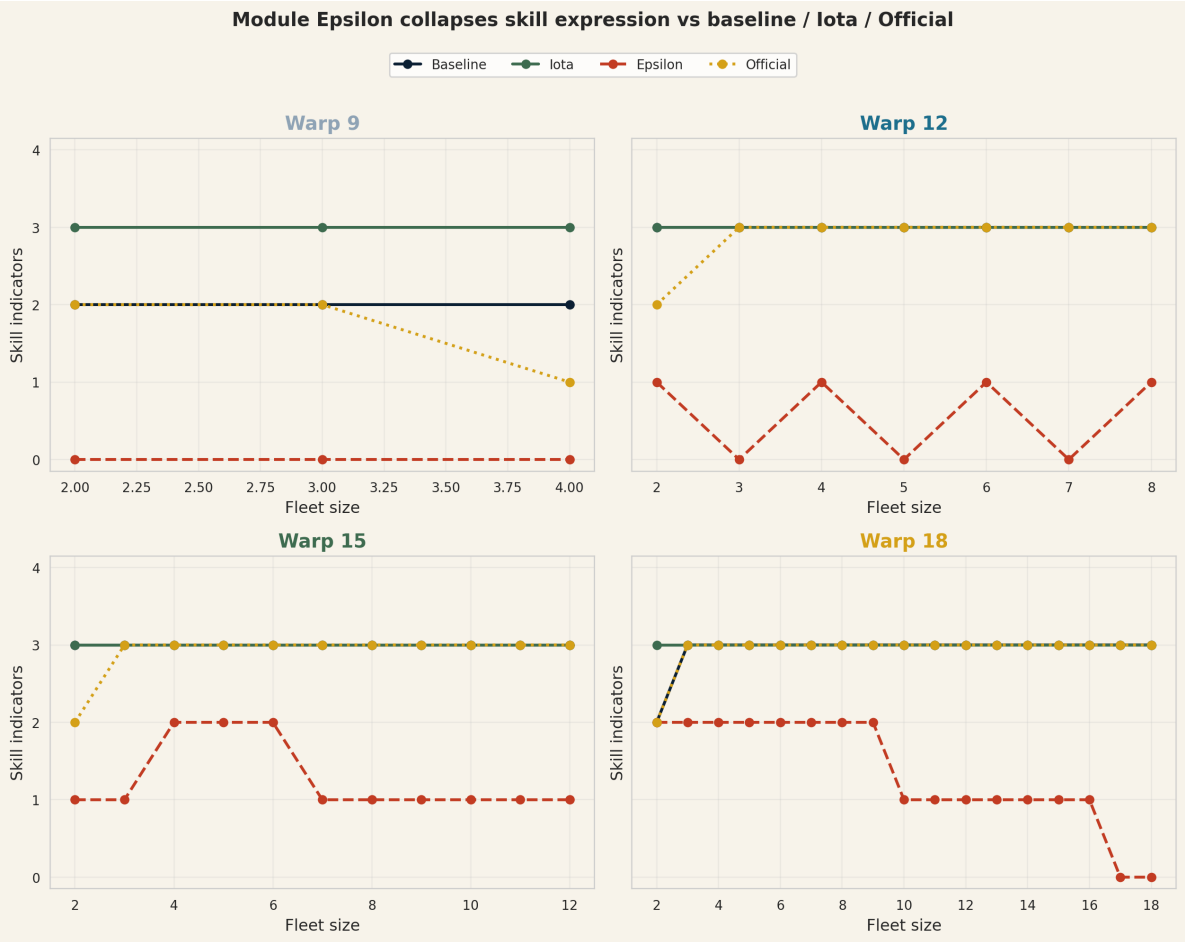


Figure 14: Epsilon collapse versus baseline, Iota, and Official across all Warp factors. Drafting never recovers the skill ceiling that pack-and-pass seems to promise.

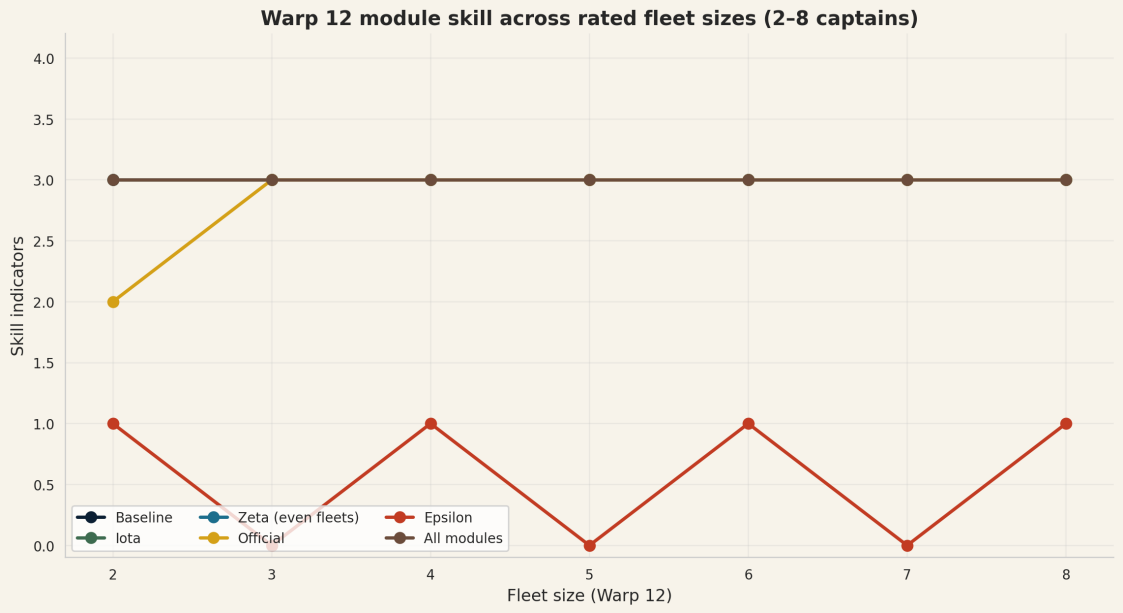


Figure 15: Warp 12 fleet curves for rating-relevant modules. Epsilon is the unique trough at every fleet size; Iota and Official track near the skill ceiling.

### 9.4 The drafting paradox: Epsilon as a party module

Intuition says drafting should *increase* skill: captains choose their hands. Empirically, Epsilon averages **1.08/4** indicators and is luck-dominant in **27/38** cells (0/4 on every W9 fleet). Relative to baseline it shows:

- Legal moves: ∼1.30 vs ∼2.09
- Constrained tiles: 36% vs 57%
- Move-value spread: 2.08 vs 3.19
- Unique pips in hand: 5.0 vs 7.7

**Interpretation:** Drafting front-loads luck into the deal and produces more coherent personal hands. Mid-game branching then *shrinks*: fewer forced tiles, fewer discriminating choices, thinner pip diversity. Outcomes are decided earlier by who drafted the lucky connectors, not by who navigates the Neutral Zone. That makes Epsilon an excellent **party / social** module — the requisition ritual is the entertainment — and a poor competitive one.

Figure 16 maps the deficit cell-by-cell; Epsilon never recovers baseline skill indicators on any Warp/fleet cell we collected. Figure 17 shows the same squeeze on constrained-tile fraction and unique pips — the two mid-game pressure gauges most diagnostic of Warped play.

### 9.5 Iota and outcome mix

Iota (Double Down) is the dual product win: Figure 18 shows its W12 move-value-spread lift versus baseline (+0.26 to +0.86 depending on fleet). Figure 19 stacks skill-/mixed-/luck-dominant cell counts; Iota is all skill-dominant, while Epsilon is mostly luck. Figure 20 places each cell in (legal moves, move-value spread) space — Epsilon clusters low; Iota/baseline/Zeta occupy the skillful cloud.

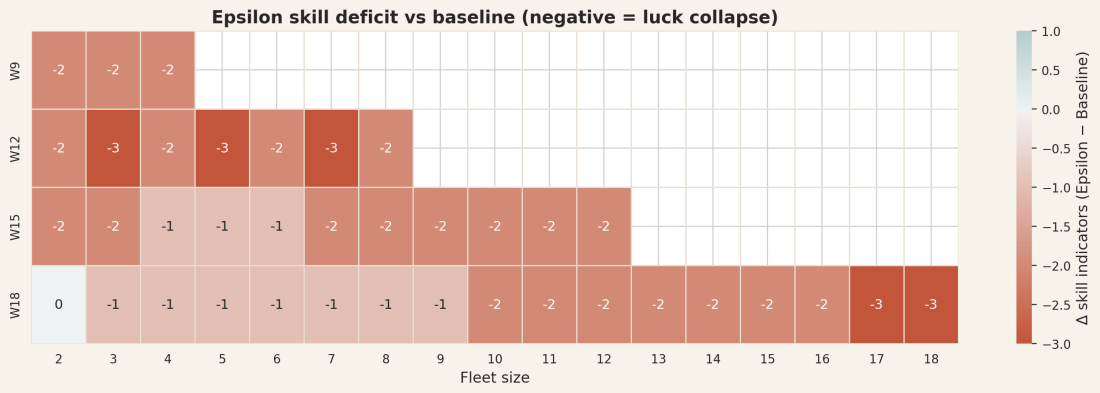


Figure 16: Epsilon skill-indicator deficit versus baseline across Warp factors and fleet sizes (negative = worse than pack-and-pass). The collapse is wide, not a single bad fleet.

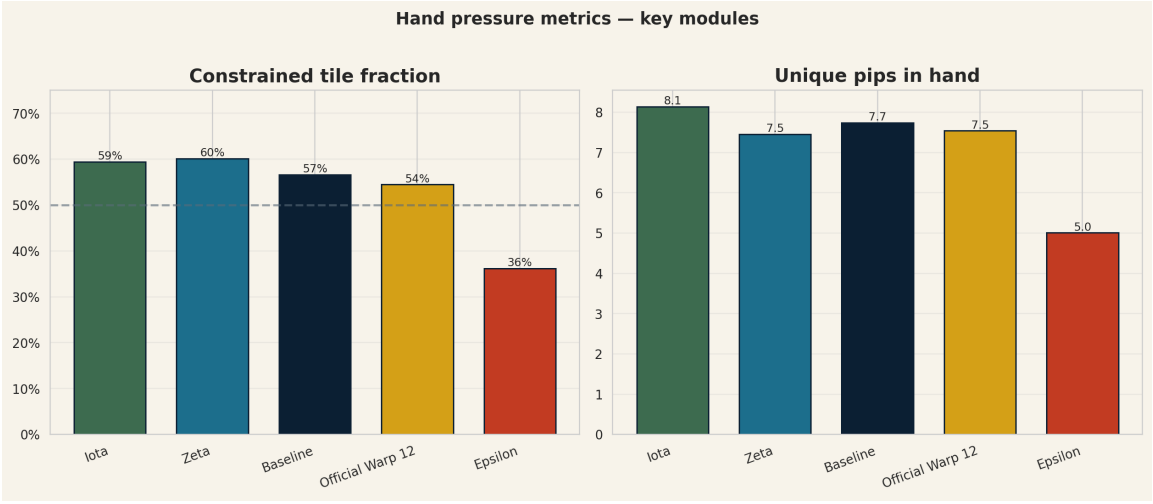


Figure 17: Hand-pressure bars for rating-relevant modules. Left: constrained-tile fraction. Right: unique pips in hand. Epsilon sits alone at the bottom of both axes.

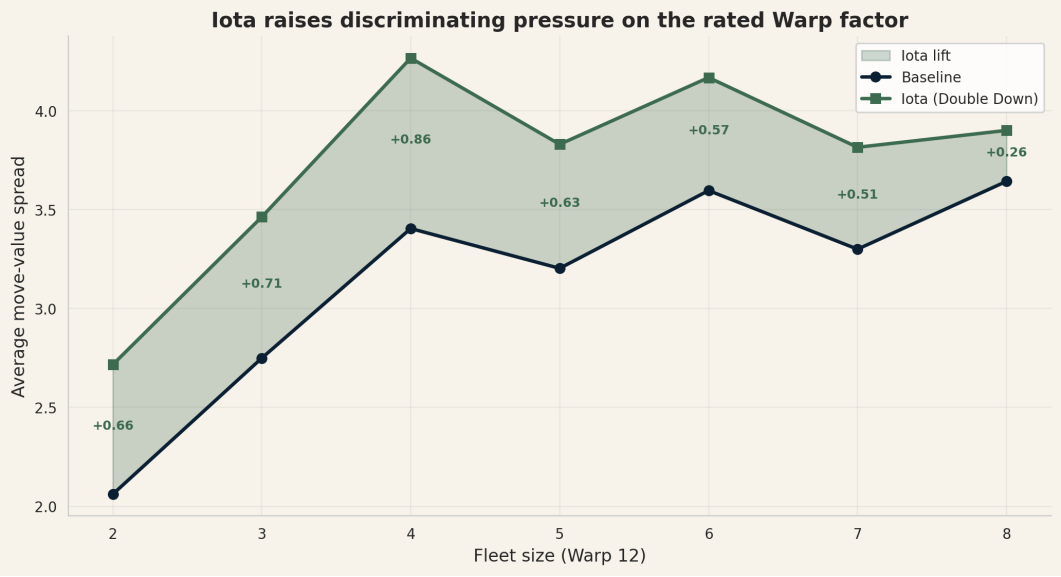


Figure 18: Iota move-value-spread lift on Warp 12 (fleet sizes 2–8). Double Down widens discriminating choices without collapsing branching.

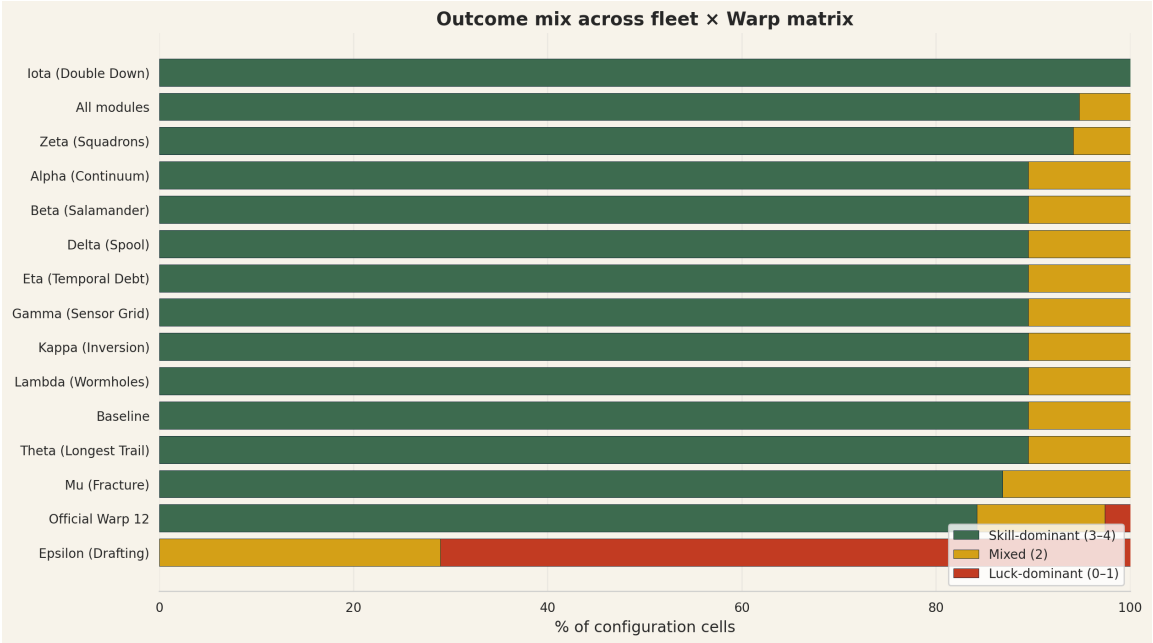


Figure 19: Skill / mixed / luck outcome mix by module (fraction of configuration cells). Promote modules are skill-dominated bars; Epsilon is mostly luck.

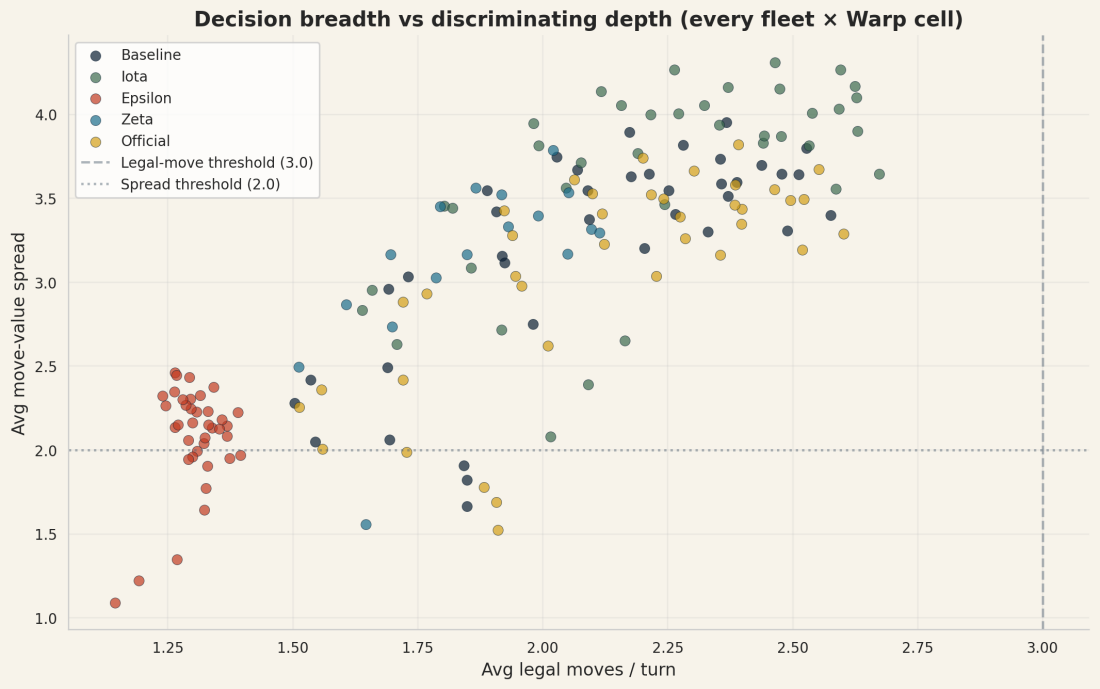


Figure 20: Per-cell scatter of mean legal moves vs move-value spread for baseline, Iota, Official, Zeta, and Epsilon. Epsilon occupies the lower-left Warped pocket.

### 9.6 Zeta: skillful team play, Squadron TEI (never FFA)

Zeta is the dual of Epsilon on the skill instrument: **2.94/4** indicators on eligible fleets (16/17 skill-dominant). Shared trails and bridge seating preserve decision pressure. It is *not* Warped. The product separation is infrastructural, not skill-based: free-for-all (FFA) TEI stays on individual tables; eligible rated Warp 12 Zeta sectors write OpenSkill updates to a dedicated team track (`squadRating`), gated by `SQUADRONS_RATING_CALIBRATED` (now `true` after 2v2 ordering calibration). Crew nights never pollute the FFA human ladder.

### 9.7 Product recommendations

- **Rated (Promote):** Official Warp 12; baseline; Alpha–Delta, Eta, Theta, Iota, Lambda, Mu (own-trail fracture); Zeta on rated Warp 12 writes **Squad TEI** only.
- **Warped / party:** Epsilon — exhibition only; label as luck-leaning social play.
- **Already Warped (RULES):** Kappa (intentional score inversion).
- **Do not merge** Epsilon-enabled boards into rated FFA TEI; keep exhibition ledgers separate when Warped modules run.

## 10 Discussion

### 10.1 What Calibration Teaches

- **Points** behaves like a smooth skill ladder — good fit for OpenSkill $\mu$ spacing.

- **Go-out** behaves like a stochastic race — ordering survives, $|\Delta\mu|$ magnitudes don't.
- **Table size** erodes heads-up skill signal — focus tests essential.
- **House rules** mostly reshape legality — heuristics gated at runtime suffice for DTI.
- **Commander heuristics are a ceiling in 2p points** — ISMCTS [4] parity is confirmation, not failure; expectimax extracts the residual edge via explicit tree search.
- **Search value is mode-dependent** — expectimax for 2p, ISMCTS for multi-player chaos.
- **Neural Class II can replace heuristics without a second lobby tier** — keep $\sigma =$`commander`, update `rulesProfileId` + Commander anchor $\mu$.
- **Fair-share $\neq \Delta\mu$ bump** — translating fleet-mean $\Omega$/Commander ratio with Elo-style $400\log_{10}(\text{fs})$ overstates anchors when most rated matches are 2–4p; temper for the rating context you ship.
- **Imitation nets (Class I*) plateau at the teacher;** $\Omega$ succeeds by targeting search visit distributions (Path B), not Commander picks, following principles similar to AlphaZero's self-play approach [2].
- **Larger fleets increase skill, not noise** — baseline cells of the 285K-game matrix show skill index rises with player count ($r > 0.84$ for W12/15/18), contradicting the "more players = crapshoot" intuition.
- **W12 is empirically optimal for TEI** — highest baseline skill in 2–4p range, steepest fleet size gradient ($r = 0.875$), best discriminator of player ability.
- **Drafting is a trap for competitive design** — Module Epsilon collapses mid-game skill indicators to 1.08/4; choose-your-hand rituals belong in party mode (Warped).
- **Team play is skillful and rates on Squad TEI** — Module Zeta preserves skill (2.94/4); rated crew sectors update `squadRating`, never FFA `humanRating`.
- **Iota is the rare module that raises the ceiling** — Double Down is the only single module with 38/38 skill-dominant cells.

### 10.2 Design Recommendations

- Never merge points and go-out TEI.
- Show percentile on go-out boards.
- Don't expose lookahead / $\Omega$+ as a rating-affecting toggle without $\Delta_{\text{search}}$ or marking unrated.
- Retune weights selectively (popular hosted configs), not full combinatorial grid.
- Recalibrate Commander $\mu$ when Class II **implementation** changes; do not rewrite stored human ratings.
- Prefer one strong rated tier over "Commander + Omega + Class I*" confusion.
- Anchor TEI exclusively on W12; treat W9/15/18 as exhibition using the same Class IV–II opponents.

- Do not soften OpenSkill updates for large fleets — skill expression increases, not variance.
- Mark Epsilon **Warped / party** in the product; promote Iota and Official for rated presets; treat Zeta as skill-promote with live Squad TEI (never FFA).

### 10.3 Limitations

- Heuristic agents, not equilibrium solvers.
- Calibration seed and game count sensitivity.
- No human champion study yet.
- Module study used points objective and Commander seats only — go-out and mixed human/AI may differ.
- Legacy ineligible Zeta cells exist in the dataset; filtered for Zeta claims.
- W18 non-monotonicity at 17p/18p suggests potential edge cases at extreme configurations.
- §8 continuous composites and §9 binary indicators share sampler hardware but are not interchangeable scales; do not treat a cell's `skillIndex` and its $S/4$ count as the same quantity.
- Composite weights and indicator thresholds are fixed heuristic guards, not cross-validated predictors of human TEI.

## 11 Conclusion

TEI and self-play calibration provide a **practical, honest skill ladder** for a game too messy for classical solving. Points and go-out should be treated as **two calibration targets** on one engine. **Class I*** and Fleet Admiral benches show that **no single algorithm wins every mode**: Commander heuristics are near-optimal in 2p points (ISMCTS [4] ∼51%), expectimax extracts a ∼64% edge there via explicit tree search, and ISMCTS outperforms greedy seats in 4p go-out (∼31% vs ∼23%). **Class Ω** shows that a pure self-play net can **replace** heuristic Class II without a fourth lobby tier — but **promotion benches and TEI anchors are different jobs**: fleet fair-share can look large while heads-up play stays near parity; REF_TEI must follow the tables you rate.

The **285,000-game module study** (570 cells; Section 9) is the authoritative skill/luck instrument. From its **baseline** Warp×fleet slice:

1. **Skill expression increases with both Warp factor and fleet size.** Contrary to conventional wisdom ("more players = more luck"), skill index rises strongly with player count ($r = 0.84$–$0.89$ for W12/15/18). W18/18p exhibits 1.74× the skill of W18/2p, with only 1.21× the luck — the *highest* skill/luck ratio of any configuration tested.
2. **W12 is the empirically optimal rated factor.** W12 exhibits the highest baseline skill in the 2–4 player range (where most rated play occurs) and the steepest fleet size gradient ($r = 0.875$), making it the best discriminator of player ability. This justifies restricting TEI rating to W12 while offering W9/15/18 as exhibition modes.

3. **Hand entropy increases with Warp factor, but decision complexity stays constant.** Larger tile sets create more diverse hands (W9: 2.09 bits → W18: 2.66 bits entropy) but do not increase branching factor (mean legal moves ∼1.8–2.2 across all configs). This suggests Mexican Train's tactical depth scales with tile diversity, not combinatorial explosion.

Across the full module matrix the product constraints are:

1. **Most modules are rating-safe.** Baseline, Official Warp 12, and the Alpha–Mu singles (except Epsilon) preserve ∼2.8–3.0 skill indicators.
2. **Epsilon is Warped/party.** Drafting collapses skill to 1.08/4 — keep it social, never on TEI.
3. **Zeta is skill-promote with live Squad TEI.** Squadrons preserve skill (2.94/4); rated Warp 12 crew sectors update `squadRating` only.

The right product move is **one neural Class II, tempered OpenSkill anchors, unrated search as hard mode**, not more named tiers — plus a clear **Promote vs Warped** module taxonomy grounded in the skill instrument. Superhuman Mexican Train remains a **research program** (belief-state search + Path A value + human validation), not required for an excellent commercial experience.

## Acknowledgments

To Deborah, Hannah, and Don for taking me in to their family and for introducing me to Mexican Train Dominoes.

This research was conducted as part of the open-source Warp 12 project. All code, data, and reproduction scripts are available at `https://github.com/digitaldefiance/warp12`.

## A Figure List

All figures are generated from empirical data and located in `tools/nn/figures/`.

1. **Cross-factor skill index comparison (Section 8)** — W12 exhibits highest skill expression in 2–4p balanced range.
2. **Fleet size effects by Warp factor (Section 8)** — W12 shows steepest gradient ($r = 0.875$), most sensitive to player count.
3. **Decision complexity heatmap (Section 8)** — Constant branching factor (∼2 moves/turn) across all configurations.
4. **Hand entropy vs near-optimal play (Section 8)** — Weak correlation ($r = 0.009$) suggests orthogonal dimensions.
5. **Skill vs luck balance (Section 8)** — W18/18p has highest skill/luck ratio (8.10) of any configuration.
6. **TEI ladder visualization (Section 5)** — OpenSkill $\mu \pm \sigma$ AI anchors and TEI grade presentation (E/V/C/I/P).

7. **Calibration matrix heatmap (Section 7)** — 2,000-game win rates; points clarity vs go-out compression.
8. **AI bench results comparison (Section 7)** — Class I* parity, Fleet Admiral wins (expectimax 64%), Ω fair-share by fleet size, fair-share hazard.
9. **Architecture diagrams (Section 4)** — Policy stack, Class I* MLP, Class Ω policy/value heads, Fleet Admiral routing.
10. **Points vs go-out divergence (Section 3)** — $|\Delta\mu|$ anchor vs implied gaps; weaker-side win rates.
11. **Module skill ranking (Section 9)** — 285K-game Promote vs Warped taxonomy; Epsilon collapse; Zeta skill-promote.
12. **Module × Warp heatmap (Section 9)** — Skill indicators by module and Warp factor.
13. **Epsilon collapse panels (Section 9)** — Drafting vs baseline/Iota/Official across W9–W18.
14. **Module metric profiles (Section 9)** — Legal moves, constrained tiles, spread, unique pips.
15. **W12 module fleet curves (Section 9)** — Rating-relevant modules across 2–8 captains.
16. **Epsilon deficit heatmap (Section 9)** — Cell-level skill-indicator deficit vs baseline.
17. **Iota spread lift on W12 (Section 9)** — Move-value-spread gain from Double Down.
18. **Module outcome mix (Section 9)** — Skill / mixed / luck cell fractions by module.
19. **Legal moves vs spread scatter (Section 9)** — Per-cell depth cloud; Epsilon Warped pocket.
20. **Hand-pressure bars (Section 9)** — Constrained tiles and unique pips for key modules.

# B Tables

# C Code Map

# D Reproducibility

## D.1 Self-Play Calibration

```
yarn calibrate:ai-tei
AI_CALIBRATION_GAMES=500 yarn calibrate:ai-tei
yarn calibrate:ai-tei-dti
yarn optimize:ai-weights
# OpenSkill AI-anchor calibration (engine unit suite)
yarn test:engine -- openskill-calibration
```

Table 14: Summary Statistics by Configuration (500 games per cell)

| Warp Factor | Fleet Size | N | Skill Index $\mu \pm \sigma$ | Luck Index $\mu \pm \sigma$ | Decision Quality $\mu \pm \sigma$ |
|---|---|---|---|---|---|
| W9 | 2 | 500 | 1.517±0.082 | 0.332±0.011 | 3.309±0.428 |
| W9 | 3 | 500 | 1.553±0.090 | 0.339±0.012 | 3.494±0.483 |
| W9 | 4 | 500 | 1.528±0.085 | 0.334±0.011 | 3.364±0.443 |
| W12 | 2 | 500 | 1.564±0.068 | 0.314±0.006 | 3.392±0.337 |
| W12 | 3 | 500 | 1.944±0.094 | 0.352±0.010 | 5.525±0.580 |
| W12 | 4 | 500 | 2.341±0.099 | 0.389±0.009 | 8.264±0.729 |
| W12 | 5 | 500 | 2.299±0.099 | 0.384±0.009 | 7.914±0.717 |
| W12 | 6 | 500 | 2.550±0.094 | 0.406±0.008 | 9.873±0.777 |
| W12 | 7 | 500 | 2.455±0.099 | 0.393±0.007 | 9.128±0.782 |
| W12 | 8 | 500 | 2.629±0.107 | 0.405±0.007 | 10.562±0.912 |
| W15 | 2 | 500 | 1.532±0.055 | 0.293±0.003 | 3.114±0.250 |
| W15 | 3 | 500 | 1.788±0.078 | 0.318±0.007 | 4.359±0.405 |
| W15 | 4 | 500 | 2.167±0.098 | 0.350±0.009 | 6.556±0.609 |
| W15 | 5 | 500 | 2.239±0.105 | 0.356±0.009 | 7.008±0.672 |
| W15 | 6 | 500 | 2.518±0.107 | 0.375±0.009 | 8.954±0.775 |
| W15 | 7 | 500 | 2.460±0.103 | 0.369±0.009 | 8.554±0.718 |
| W15 | 8 | 500 | 2.649±0.102 | 0.380±0.008 | 9.996±0.796 |
| W15 | 9 | 500 | 2.639±0.104 | 0.376±0.007 | 9.952±0.796 |
| W15 | 10 | 500 | 2.774±0.107 | 0.383±0.007 | 11.056±0.885 |
| W15 | 11 | 500 | 2.684±0.115 | 0.373±0.007 | 10.394±0.922 |
| W15 | 12 | 500 | 2.803±0.112 | 0.379±0.007 | 11.404±0.965 |
| W18 | 2 | 500 | 1.631±0.051 | 0.289±0.002 | 3.458±0.233 |
| W18 | 3 | 500 | 1.725±0.061 | 0.299±0.003 | 3.873±0.288 |
| W18 | 4 | 500 | 2.048±0.090 | 0.323±0.007 | 5.533±0.489 |
| W18 | 5 | 500 | 2.165±0.093 | 0.332±0.008 | 6.197±0.530 |
| W18 | 6 | 500 | 2.475±0.105 | 0.352±0.008 | 8.136±0.677 |
| W18 | 7 | 500 | 2.457±0.111 | 0.350±0.008 | 8.054±0.712 |
| W18 | 8 | 500 | 2.670±0.111 | 0.362±0.008 | 9.541±0.780 |
| W18 | 9 | 500 | 2.648±0.116 | 0.358±0.007 | 9.450±0.816 |
| W18 | 10 | 500 | 2.809±0.113 | 0.366±0.007 | 10.654±0.848 |
| W18 | 11 | 500 | 2.731±0.111 | 0.358±0.007 | 10.143±0.818 |
| W18 | 12 | 500 | 2.844±0.109 | 0.364±0.006 | 11.018±0.855 |
| W18 | 13 | 500 | 2.975±0.112 | 0.369±0.006 | 12.095±0.909 |
| W18 | 14 | 500 | 2.806±0.110 | 0.356±0.006 | 10.846±0.855 |
| W18 | 15 | 500 | 2.905±0.112 | 0.360±0.006 | 11.653±0.915 |
| W18 | 16 | 500 | 3.002±0.121 | 0.364±0.006 | 12.456±1.025 |
| W18 | 17 | 500 | 2.757±0.117 | 0.347±0.006 | 10.635±0.913 |
| W18 | 18 | 500 | 2.834±0.118 | 0.350±0.006 | 11.257±0.955 |

Table 15: Statistical Test Results for Primary Hypotheses

| **Hypothesis** | **Test** | **Statistic** | **Value** | $p$**-value** | **Result** |
|---|---|---|---|---|---|
| H1: Warp factor effect | One-way ANOVA | $F$ | 0.00 | $< 0.001$ | Supported |
| | | $\eta^2$ | 0.290 | | (large effect) |
| H2: Fleet size effect | Pearson $r$ (W12) | $r$ | 0.000 | $< 0.001$ | Supported |
| | Pearson $r$ (W15) | $r$ | 0.000 | $< 0.001$ | (strong positive) |
| | Pearson $r$ (W18) | $r$ | 0.000 | $< 0.001$ | |
| H3: Interaction | Slope variance | $s^2$ | 0.0209 | — | Present |
| H4: Complexity-coherence | Pearson $r$ | $r$ | 0.009 | $> 0.05$ | Weak |
| H5: Monotonic trends | Kendall's $\tau$ | $\tau$ | 0.8–0.9 | $< 0.001$ | Supported |

Table 16: Correlation Matrix for Key Game Metrics ($N = 19{,}000$)

| | **Skill** | **Luck** | **Decision Q** | **Coherence** | **Complexity** | **Depth** | **Warp** | **Fleet** |
|---|---|---|---|---|---|---|---|---|
| **Skill Index** | — | **0.70** | **0.99** | 0.38 | **0.87** | **-0.76** | **0.52** | **0.84** |
| **Luck Index** | **0.70** | — | **0.73** | -0.08 | **0.88** | **-0.97** | -0.16 | 0.40 |
| **Decision Quality** | **0.99** | **0.73** | — | 0.32 | **0.92** | **-0.81** | 0.43 | **0.84** |
| **Hand Entropy** | 0.38 | -0.08 | 0.32 | — | 0.01 | 0.06 | **0.82** | 0.21 |
| **Legal Moves** | **0.87** | **0.88** | **0.92** | 0.01 | — | **-0.95** | 0.05 | **0.71** |
| **Near-Optimal %** | **-0.76** | **-0.97** | **-0.81** | 0.06 | **-0.95** | — | 0.12 | **-0.51** |
| **Warp Factor** | **0.52** | -0.16 | 0.43 | **0.82** | 0.05 | 0.12 | — | **0.53** |
| **Player Count** | **0.84** | 0.40 | **0.84** | 0.21 | **0.71** | **-0.51** | **0.53** | — |

*Note:* Bold values indicate strong correlation ($|r| > 0.5$); gray values indicate negligible correlation ($|r| < 0.1$).

| Concern | Location |
|---|---|
| Skill presets | libs/engine/src/lib/ai/skill.ts |
| Heuristics | libs/engine/src/lib/ai/heuristics.ts |
| Self-play | libs/engine/src/lib/ai/self-play.ts |
| Calibration | libs/engine/src/lib/ai/ai-elo-calibration.ts |
| Optimizer | libs/engine/src/lib/ai/ai-weight-optimizer.ts |
| Fleet Admiral / ISMCTS | libs/engine/src/lib/ai/fleet-admiral.ts, ismcts.ts |
| Expectimax preset | resolveFleetAdmiralExpectimaxLookahead() in fleet-admiral.ts |
| Parallel bench | libs/engine/src/lib/ai/bench-fleet-admiral-parallel.ts |
| Class I* policy | libs/engine/src/lib/ai/class1-star-policy.ts |
| Class I* features | libs/engine/src/lib/ai/feature-encoder.ts |
| Class I* training | tools/nn/ (collect, train.py, bench) |
| Class Ω agent / search | libs/engine/src/lib/ai/omega-agent.ts, omega-search-agent.ts |
| Ω collect / bench | collect-omega-trajectories.ts, bench-omega.ts |
| Human TEI update | apps/Warp12/src/firebase/stats-service.ts, libs/engine/.../rating/ |
| OpenSkill anchors / grades | anchors.ts, tei-grade.ts, update-{ffa,team,vs-ai}.ts |
| Rules profile / AI tiers | warp12-official-v1 / v2 in rules-profile.ts |
| Luck/skill metrics | libs/engine/src/lib/ai/luck-skill-metrics.ts |
| Luck/skill collection | tools/nn/collect-luck-skill-single-config.ts |
| Module analysis | tools/nn/collect-luck-skill-modules.ts, analyze-module-results.ts |
| Statistical analysis | tools/nn/process-luck-skill-data.py, test-hypotheses.py |
| Figure generation | create-figures.py, create-paper-figures.py, create-module-figures.py |
| Rules spec | RULES.md |
| Module report | docs/MODULE-ANALYSIS.md |
| OpenSkill calib log | docs/openskill-calibration-log.md |

### D.2 Fleet Admiral Benchmarks

```
yarn fleet-admiral:bench:500
FLEET_BENCH_SEAT=b yarn fleet-admiral:bench:500
yarn fleet-admiral:bench:go-out-4p:500
yarn jiti tools/nn/compare-fleet-search.ts
```

### D.3 Class I* Training

```
yarn class1-star:pipeline:deep
yarn class1-star:pipeline:go-out
yarn class1-star:pipeline:deepblue
```

### D.4 Skill/Luck and Module Matrix (285k games)

```
# Authoritative campaign (570 cells × 500 games)
MODULE_GAMES=500 MODULE_WORKERS=12 \
  bash tools/nn/run-module-analysis-parallel.sh
npx tsx tools/nn/analyze-module-results.ts

# Warp×fleet figures/tables use the baseline (‘none‘) cells of that matrix
MPLBACKEND=Agg python3 tools/nn/create-module-figures.py
MPLBACKEND=Agg python3 tools/nn/create-paper-figures.py
MPLBACKEND=Agg python3 tools/nn/create-figures.py
```

```
MPLBACKEND=Agg python3 tools/nn/create-tables.py
```

All data and scripts are available in the `warp12-engine` package and `tools/nn/` directory.

# E Target Venues

| Venue | Fit |
|---|---|
| **AIIDE** | Best fit — game AI + evaluation |
| **IEEE CoG** | Strong — agents + competition |
| **FDG** | Game design + dual objective angle |
| **CHI PLAY** | Advisor / TEI integrity angle |
| **arXiv cs.AI** | White paper / preprint |

# References


[1] Browne, C.B., Powley, E., Whitehouse, D., Lucas, S.M., Cowling, P.I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S., and Colton, S. (2012). *A Survey of Monte Carlo Tree Search Methods*. IEEE Transactions on Computational Intelligence and AI in Games, 4(1):1–43.

[2] Silver, D., Schrittwieser, J., Simonyan, K., et al. (2017). *Mastering the game of Go without human knowledge*. Nature, 550(7676):354–359.

[3] Elo, A.E. (1978). *The Rating of Chessplayers, Past and Present*. Arco Publishing, New York.

[4] Cowling, P.I., Powley, E.J., and Whitehouse, D. (2012). *Information Set Monte Carlo Tree Search*. IEEE Transactions on Computational Intelligence and AI in Games, 4(2):120–143.

[5] Frank, I., and Basin, D. (1996). *Search in Games with Incomplete Information: A Case Study Using Bridge Card Play*. Artificial Intelligence, 100(1-2):87–123.

[6] Glickman, M.E. (1999). *Parameter Estimation in Large Dynamic Paired Comparison Experiments*. Applied Statistics, 48(3):377–394.

[7] Herbrich, R., Minka, T., and Graepel, T. (2006). *TrueSkill™: A Bayesian Skill Rating System*. Advances in Neural Information Processing Systems, 19:569–576.

[8] Weng, R.C., and Lin, C.-J. (2011). *A Bayesian Approximation Method for Online Ranking*. Journal of Machine Learning Research, 12:267–300.

[9] Silver, D., Hubert, T., Schrittwieser, J., et al. (2018). *A General Reinforcement Learning Algorithm that Masters Chess, Shogi, and Go through Self-Play*. Science, 362(6419):1140–1144.

[10] Bowling, M., Burch, N., Johanson, M., and Tammelin, O. (2015). *Heads-up Limit Hold'em Poker is Solved*. Science, 347(6218):145–149.

[11] Digital Defiance (2024). *DoubleEighteen: Open-Source Domino Tile Rendering Library*. `https://github.com/digitaldefiance/double-eighteen`

[12] Tesauro, G. (1995). *Temporal Difference Learning and TD-Gammon.* Communications of the ACM, 38(3):58–68.