Context-Aware Skill Representation in Competitive RTS: A Graph-Anchored Bradley–Terry Framework under Barrier-Mediated Non-Demotion Constraints

Developer: Unkn0wn1

Abstract

Skill representation in competitive one-versus-one real-time strategy (RTS) games is complicated by non-stationarity, contextual asymmetry, and high short-term volatility. Furthermore, small-scale closed communities (approx. 300 players) often impose operational constraints, such as strict non-demotion policies, which conflict with the demotion-permitting dynamics of classical rating systems such as Elo or Bradley–Terry.

To address these unique requirements, we present a system that is not designed to estimate true latent player skill in a statistical sense, but rather serves as a policy-constrained score synthesis framework. This system operates on a closed population and converts historical match results into stable scores; it does not perform matchmaking or real-time prediction.

To satisfy the non-demotion requirement while preserving ordinal differentiation, we introduce a Hybrid Dynamic Update Scheme that combines Riemannian optimization with Policy-Driven Force Injection and Momentum-Modulated Scheduling. Instead of relying solely on static curvature, the system applies Adaptive Gain Control to balance structural stability with short-term responsiveness. In contrast to standard rating models, global drift is controlled via graph-theoretic anchors selected from the closed population. Contextual effects (map, match type, race) are learned as shared-gradient additive modifiers to improve internal consistency rather than probabilistic calibration.

In addition, a downstream Metrics Engine leverages Shannon entropy to quantify opponent diversity and discourage exploitative behavior within the small player pool. The resulting framework prioritizes ordinal stability, resistance to rank oscillation, and interpretable progression over statistical purity or asymptotic convergence guarantees.

Keywords:
Policy-Constrained Rating Systems, Bradley–Terry–Inspired Models, Graph-Based Anchoring, Non-Stationary Competitive Dynamics, Context-Aware Score Adjustment, Non-Demotion Constraints

1. Introduction

Accurately representing player performance in competitive one-versus-one real-time strategy (RTS) games is challenging not only due to statistical complexity, but also due to operational constraints imposed by the communities in which such systems are deployed. Unlike large-scale, open matchmaking environments, many RTS communities operate as small, closed populations in which rating systems are expected to provide stability, interpretability, and long-term trust rather than statistically optimal skill recovery.

In such settings, classical rating systems—including Elo, Glicko, and Bradley–Terry–based estimators—face a fundamental mismatch between their underlying assumptions and operational requirements. These models typically treat upward and downward rating movement symmetrically and are designed to track latent skill under stationarity or mild non-stationarity. However, in practice, a small closed starcraft community (FMKorea) enforces strict non-demotion policies once players reach certain qualitative tiers, prioritizing social stability and engagement over symmetric statistical correction. As a result, direct application of unconstrained rating models frequently leads to undesirable oscillations, loss of user trust, or the need for ad hoc post-hoc interventions.

This work addresses this mismatch by reframing rating not as a problem of latent skill estimation, but as a **policy-constrained score synthesis task**. Rather than attempting to recover a statistically "true" skill parameter, the proposed system converts historical match outcomes into stable ordinal scores that explicitly respect non-demotion and stability constraints. The system operates on a closed population of limited size and does not perform matchmaking, prediction, or real-time optimization.

To achieve this, we embed operational constraints directly into the update dynamics of the rating process. The core update mechanism is formulated as a **Dynamic Levenberg-Marquardt optimization engine**, featuring momentum-coupled damping to balance between stability (over-damped) and responsiveness (under-damped), which minimizes an objective function augmented with explicit policy-driven penalties. In particular, non-demotion is enforced through asymmetric barrier terms that exert increasing resistance to downward movement while remaining smooth and numerically stable. Momentum-conditioned damping further suppresses short-term volatility without eliminating long-term differentiation.

Because closed systems lack an external calibration reference, global score drift is controlled via a graph-theoretic anchoring mechanism derived from the interaction structure of the match graph. Structurally central players, identified through spectral properties of the interaction graph, provide a dynamically maintained internal reference that stabilizes the global score scale without asserting identifiability in a statistical sense. Contextual factors such as map, match type, and race are incorporated as shared-gradient additive modifiers, improving internal consistency rather than probabilistic calibration.

Finally, we decouple score synthesis from performance interpretation through a downstream Metrics Engine that applies information-theoretic measures to characterize opponent diversity. By leveraging Shannon entropy, the system discourages exploitative or collusive interaction patterns in a principled manner, without interfering with the core rating dynamics.

Contributions.

The primary contributions of this work are:

1. A policy-constrained rating framework explicitly designed for non-demotion environments in small closed competitive systems.

2. A Dynamic Levenberg-Marquardt optimization engine utilizing a curvature-aware update mechanism, reinforced by asymmetric barrier regularization and momentum-conditioned damping.

3. A graph-based anchoring strategy for internal drift control without external calibration.

4. An information-theoretic Metrics Engine for post-hoc performance characterization.

1.1 Non-Demotion as Asymmetric Update Resistance

Beyond the intrinsic variability of competitive RTS outcomes, the target ecosystem (FMKorea) imposes a rigorous non-demotion requirement as an operational policy. Within this community, a player's attained tier is treated not merely as a transient performance indicator, but as a cumulative achievement that should not be easily revoked. Consequently, rating updates are expected to exhibit substantial resistance to aggressive downward movement once protected reference levels are reached. Although the motivation for this requirement is social—aimed at preserving engagement and long-term trust—its incorporation in our framework is purely mathematical.

Classical rating and scoring models implicitly assume symmetric update dynamics, in which upward and downward score adjustments are treated as equally admissible responses to observed match outcomes. This symmetry underlies a broad class of sequential update schemes, including random-walk-style and likelihood-driven estimators. Non-demotion policies violate this assumption by introducing directional asymmetry: upward movement remains largely data-driven, while downward movement is progressively resisted as scores approach protected regions.

This asymmetry gives rise to a fundamental **responsiveness–stability trade-off**. Under unconstrained updates, sequences of unfavorable outcomes induce proportional downward adjustments, preserving responsiveness but undermining perceived stability. Conversely, naïve enforcement of non-demotion through frozen updates or hard truncation decouples scores from observed results, leading to drift, inflation, and loss of internal consistency.

To address this tension, we model non-demotion not as a hard boundary or irreversible rule, but as a smooth asymmetric influence embedded directly into the update dynamics. Downward movement is resisted through a bounded, differentiable barrier term centered at a reference score level, introducing increasing opposition as updates approach the protected region. This formulation ensures continuity and numerical stability while preserving the ability to respond to sustained evidence.

By embedding asymmetric resistance directly into the update process, non-demotion emerges as a dynamical property of the optimization landscape rather than as an external intervention on parameter values. The resulting system reconciles operational stability with controlled responsiveness, maintaining interpretability and robustness without resorting to post-hoc correction or rule-based overrides.

1.2 Closed-Population Drift and Graph-Based Anchoring

In closed competitive systems, rating dynamics are fundamentally underdetermined. When all scores are updated solely through relative comparisons, the system admits a family of equivalent configurations related by global translation or slow collective drift. In large open populations, such drift is often mitigated implicitly through population turnover or external calibration. In small closed populations, however, no such mechanism exists, and unconstrained updates inevitably lead to global inflation, collapse, or oscillatory behavior—even when local update rules are well-behaved.

It is important to emphasize that this phenomenon is **not** a failure of estimation accuracy, nor a violation of probabilistic consistency. Rather, it is a structural consequence of operating within a closed interaction graph without an external reference. Accordingly, our objective is not to establish statistical identifiability in the classical sense, but to impose a stable internal reference that prevents unbounded global drift while preserving relative ordering.

To this end, we introduce a graph-based anchoring mechanism derived from the interaction topology of the match graph. Players are represented as nodes, with weighted edges corresponding to observed matches. From this graph, we identify structurally central players—those that serve as bridges between otherwise weakly coupled subgraphs—using spectral properties of the interaction matrix. These players do not act as fixed ground truths or calibration constants; instead, they collectively define a soft internal reference frame that stabilizes the global score scale.

Crucially, anchoring is implemented as a **regularization influence on the update dynamics**, not as a hard constraint on score values. Anchors are allowed to move, but their collective motion is damped relative to the population, preventing coherent drift of the entire system. This design avoids asserting absolute meaning to any individual score, while ensuring that ordinal relationships remain stable over time.

We stress that this mechanism does not claim statistical identifiability, nor does it attempt to recover an absolute skill scale. Such objectives are ill-posed in closed systems without external calibration. Instead, anchoring serves a purely operational role: maintaining a consistent and interpretable score landscape under long-term constrained updates.

By explicitly separating **drift control** from **score synthesis**, the framework avoids conflating stability mechanisms with skill inference. The result is a rating system that remains robust under non-demotion constraints, resistant to global score collapse or inflation, and interpretable within the limited context for which it is designed.

1.3 Contextual Modulation versus Skill Representation

Contextual factors such as map selection, match format, and race interactions play a dominant role in observed outcomes of competitive RTS matches. However, the presence of such factors does not imply that they should be interpreted as separable components of latent player skill, nor that they admit independent statistical identification. In this work, we explicitly reject the premise that player skill can—or should—be disentangled from contextual effects in a causal or probabilistic sense.

Within the proposed framework, contextual terms are introduced solely as **modulatory components of the score update dynamics**, not as estimators of intrinsic ability. Their purpose is not to recover context-invariant skill parameters, but to improve internal consistency by preventing systematic contextual biases from being absorbed into long-term score trajectories. In particular, contextual modifiers act as shared-gradient additive adjustments that redistribute update pressure across matches, ensuring that comparable performances under different conditions are treated coherently within the closed system.

This design choice reflects a deliberate departure from classical context-aware estimation frameworks that seek identifiability, disentanglement, or causal interpretation. In closed populations with limited data and strong operational constraints, such objectives are ill-posed and unnecessary. Attempting to enforce statistical separation between skill and context in this setting would introduce additional assumptions without improving operational reliability.

Accordingly, contextual modulation in this framework should be understood as an **instrumental mechanism**, not an inferential claim. Context parameters are optimized jointly with scores under policy-regularized objectives, and their values carry no standalone semantic meaning outside the system. The framework neither asserts nor requires that these parameters correspond to stable, transferable, or interpretable quantities.

By clearly separating **score synthesis** from **skill representation**, the system avoids conflating operational stability with statistical inference. Contextual effects serve to stabilize updates and preserve ordinal relationships under heterogeneous conditions, while questions of latent ability, causal attribution, or cross-context generalization are intentionally left outside the scope of the model.

2. Pairwise Comparison Objective under Policy Constraints

This section formalizes the score update mechanism underlying the proposed framework. Rather than constructing a probabilistic generative model of match outcomes, we formulate the rating process as the optimization of a **pairwise comparison objective** augmented with explicit policy-driven regularization terms. The role of this objective is not to recover a statistically meaningful latent skill parameter, but to provide a stable, differentiable signal that drives consistent score updates under operational constraints.

The formulation is inspired by the functional form of the Bradley–Terry model, but departs from it in both interpretation and intent. In particular, we do not assume that match outcomes are generated by an underlying stochastic skill process, nor do we interpret intermediate quantities as calibrated probabilities. Instead, the logistic comparison structure is used as a smooth surrogate loss that preserves ordinal consistency while remaining amenable to constrained optimization.

Throughout this section, we refer to player scores as elements of a dynamic state vector

$$\theta = (\theta_1, \theta_2, \cdots, \theta_N)$$

where each $\theta_i \in \mathbb{R}$ represents the current score of player i. These scores should be interpreted as **operational ranking coordinates**, not as estimates of intrinsic or transferable ability.

2.1 Bradley–Terry–Style Pairwise Comparison Loss

Consider a match between two players i and j, with observed outcome $y_{ij} \in \{0,1\}$, where $y_{ij} = 1$ indicates that player i defeats player j. We define a pairwise comparison score using a logistic link function:

$$p_{ij} = \sigma(\eta_{ij}) = \frac{1}{1 + \exp(-\eta_{ij})},$$

where η_{ij} is a real-valued comparison signal to be specified later. While p_{ij} takes the mathematical form of a probability, it is used here **solely as a smooth comparison function**, not as a calibrated likelihood.

Given a set of observed matches $\mathcal{M}$, we define the data-driven loss term:

$$\mathcal{L}_{data}(\theta) = \sum_{(i,j)\in\mathcal{M}} \left[y_{ij} \log p_{ij} + (1 - y_{ij}) \log(1 - p_{ij}) \right]$$

This expression mirrors the log-likelihood of a classical Bradley–Terry model, but its role in our framework is strictly instrumental: it provides a **monotone, differentiable penalty** that encourages consistency between observed outcomes and relative score differences. No probabilistic interpretation is required or assumed.

A key structural property of this loss is **shift invariance**:

$$\mathcal{L}_{data}(\theta) = \mathcal{L}_{data}(\theta + c1) \quad \forall_c \in \mathbb{R}$$

reflecting the fact that only relative score differences influence pairwise comparisons. In unconstrained settings, this invariance is typically resolved through arbitrary normalization. In closed systems with strong policy constraints, however, this degree of freedom must be handled explicitly—a point addressed later through graph-based anchoring.

At this stage, it is crucial to emphasize that minimizing $-\mathcal{L}_{data}$ alone is NOT the objective of the system. Left unconstrained, such optimization would reintroduce symmetric upward and downward score dynamics, directly conflicting with non-demotion requirements and long-term stability goals. Instead, $\mathcal{L}_{data}$ serves as the baseline comparison signal upon which policy-aware modifications are imposed.

2.2 Contextual Modulation as Additive Update Bias

Observed match outcomes in competitive RTS environments are strongly influenced by contextual factors such as map selection, match format, and race interactions. Within the proposed framework, these factors are not treated as sources of latent explanatory power, nor are they assumed to admit independent statistical identification. Instead, contextual information is incorporated as an **additive modulation of the pairwise comparison signal**, shaping how update pressure is distributed across matches without asserting causal or probabilistic meaning.

Formally, we define the comparison signal η_{ij} as:

$$\eta_{ij} = (\theta_i - \theta_j) + \boldsymbol{x}_{match}^T \boldsymbol{\beta} + \left(\rho_{i,r_j} - \rho_{j,r_i}\right)$$

Where θ_i, $\theta_j \in \mathbb{R}$ denote the current scores of players i and j, $\boldsymbol{\beta}$ is a vector of global context coefficients shared across the population, $\boldsymbol{x}_{match}$ is a feature vector encoding match-level contextual attributes (e.g., map, match format). ρ_{i,r_j} represents a player-specific interaction term capturing persistent asymmetries against opponent race r_j (Race Matchup Bias).

Crucially, these contextual terms are **not interpreted as decomposing outcomes into skill and advantage**, nor are they intended to isolate intrinsic performance components. Their sole function is to **redistribute gradient magnitude** during optimization, preventing systematic contextual biases from being absorbed into long-term score updates.

Under this formulation, the gradient of the data term with respect to θ_i becomes:

$$\frac{\partial \mathcal{L}_{data}}{\partial \theta_i} = \sum_{j:(i,j)\in\mathcal{M}} (y_{ij} - p_{ij})$$

Where $p_{ij} = \sigma(\eta_{ij})$.

The contextual parameters influence this gradient **indirectly**, by shifting η_{ij} and thereby modulating the discrepancy between observed outcomes and comparison scores.

This mechanism ensures that wins achieved under systematically favorable conditions generate smaller upward update pressure than wins achieved under neutral or adverse conditions, without requiring explicit attribution of causality. Similarly, losses under unfavorable contexts do not induce disproportionately large downward pressure. In this sense, contextual modulation functions as a **bias-correcting lens**, not as an estimator of context-invariant ability.

Importantly, contextual parameters $\boldsymbol{\beta}$ and $\boldsymbol{\rho}$ are optimized jointly with the score vector $\boldsymbol{\theta}$ under the same policy-regularized objective. They do not converge to interpretable standalone quantities, nor are they required to generalize across systems. Their values are meaningful only insofar as they stabilize update dynamics within the closed population.

By framing context awareness as an additive modulation of the comparison signal rather than as a decomposition of skill, the framework maintains consistency with the policy-constrained score synthesis perspective established in Section 1. Contextual effects improve internal coherence of updates while deliberately avoiding claims of disentanglement, identifiability, or causal inference.

2.3 Policy-Regularized Objective Function

While the pairwise comparison loss defined in Sections 2.1 and 2.2 provides a smooth signal linking observed match outcomes to relative score differences, it is insufficient as a standalone update objective in environments governed by strong operational constraints. In particular, unconstrained optimization of the data term would induce symmetric upward and downward score dynamics, reintroducing instability and directly violating non-demotion requirements.

To address this, we formulate the rating update process as the optimization of a **policy-regularized objective function**, in which operational constraints are embedded directly into the optimization landscape rather than enforced through external rules or post-hoc corrections.

Let $\boldsymbol{\theta}$, $\boldsymbol{\beta}$, and $\boldsymbol{\rho}$ denote the score vector, global contextual coefficients, and player–race interaction parameters, respectively. The objective function is defined as:

$$\mathcal{J}(\boldsymbol{\theta},\boldsymbol{\beta},\boldsymbol{\rho}) = \mathcal{L}_{data}(\boldsymbol{\theta},\boldsymbol{\beta},\boldsymbol{\rho}) - \lambda_{demotion} R_{demotion}(\boldsymbol{\theta}) - \lambda_{anchor} R_{anchor}(\boldsymbol{\theta})$$

Here, $\mathcal{L}_{data}$ represents the pairwise comparison loss defined in Section 2.1, $R_{demotion}$ is an asymmetric regularization term enforcing non-demotion behavior, R_{anchor} is a drift-control regularization term that stabilizes the global score scale, and $\lambda_{demotion}, \lambda_{anchor} > 0$ control the relative strength of policy enforcement.

Asymmetric Non-Demotion Regularization

Non-demotion is modeled as a smooth asymmetric resistance to downward score movement near protected reference levels. For each player i, let b_i denote a reference score associated with the lowest boundary of the player's current protected tier. We define the non-demotion regularizer as:

$$R_{demotion}(\boldsymbol{\theta}) = \sum_i \phi(\theta_i - b_i),$$

Where $\phi(\cdot)$ is a monotonically decreasing barrier function satisfying:

$$\lim_{x \to 0^+} \phi'(x) = +\infty, \qquad \phi'(x) \approx 0 \; for \; x \gg 0.$$

In practice, ϕ is implemented as a softened logarithmic barrier with gradient clipping to ensure numerical stability. This formulation introduces increasing resistance as θ_i approaches b_i from above, while exerting negligible influence when the score is safely above the protected region.

Importantly, this regularization does not impose a hard constraint on score values. Downward movement remains possible under sustained adverse evidence, but occurs at a progressively reduced rate, ensuring continuity, differentiability, and controlled responsiveness.

Interpretation as Optimization Geometry

The resulting objective function does not correspond to a classical likelihood, nor does it admit a probabilistic interpretation. Instead, it defines a **policy-shaped optimization geometry** in which gradients arising from observed outcomes interact smoothly with asymmetric resistance fields imposed by operational requirements.

Score updates are computed using a damped second-order method, described in Section 3, which respects the curvature induced by both the data term and the policy regularizers. In this setting, convergence is understood operationally—as the emergence of stable ordinal relationships under continued updates—rather than as attainment of a statistically optimal estimator.

By embedding non-demotion directly into the objective function, the framework avoids brittle rule-based enforcement and ensures that policy compliance emerges organically from the update dynamics.

2.4 Closed-System Drift Control via Graph-Based Anchoring

The objective function defined in Section 2.3 inherits a fundamental structural property of pairwise comparison losses: **global shift invariance**. Specifically, the data term $\mathcal{L}_{data}$ remains unchanged under uniform translation of all scores,

$$\theta \mapsto \theta + \mathrm{c}1, \qquad \mathrm{c} \in \mathbb{R}$$

implying that only relative differences between scores influence update dynamics. In unconstrained settings, this degree of freedom is often resolved implicitly through normalization or external calibration. In closed populations, however, no **natural** reference exists. **Therefore, we introduce an axiomatic reference point** (μ_{ref}), as unconstrained updates would otherwise lead to slow collective drift of the entire score vector.

This phenomenon is not an artifact of numerical instability, nor does it reflect a failure of the underlying comparison loss. Rather, it is a structural consequence of operating within a closed interaction graph without an external anchor. Without explicit intervention, long-term updates may result in score inflation, collapse, or oscillatory behavior, even when local update rules are well-behaved.

To prevent such global drift while preserving relative ordering, we introduce a **graph-based anchoring regularization** that stabilizes the score scale without asserting absolute meaning to any individual value.

Interaction Graph and Structural Centrality

We represent the competitive ecosystem as an interaction graph

$$\boldsymbol{G} = (V, E),$$

Where each node $i \in V$ corresponds to a player, and weighted edges $(i, j) \in E$ encode observed matches between players i and j. Edge weights reflect interaction frequency and are normalized to account for repeated encounters.

From this graph, we compute a centrality measure that captures the structural importance of players in maintaining global connectivity. In particular, we employ a spectral centrality criterion derived from the stationary distribution of a random walk on $\boldsymbol{G}$. Players with high centrality act as bridges between otherwise weakly coupled subgraphs, providing a natural internal reference for stabilizing the score system.

It is critical to emphasize that these players are **not** treated as ground truth anchors, nor are their scores fixed or privileged. Centrality is used solely to modulate the strength of anchoring forces applied during optimization.

Anchoring Regularization

The anchoring regularizer is defined as:

$$R_{anchor}(\boldsymbol{\theta}) = \frac{1}{2}\left(\frac{1}{|\mathcal{A}|}\sum_{i \in \mathcal{A}} \theta_i - \mu_{ref}\right)^2$$

Where $\mathcal{A}$ is the set of anchor players identified via PageRank, and μ_{ref} is the external reference constant (e.g., 16.0) that stabilizes the global scale.

This term penalizes the deviation of the **anchor group's collective mean from the reference constant**, thereby suppressing global translation modes while allowing local score differentiation to evolve freely. Because the regularization is quadratic and smooth, it integrates naturally into the second-order update scheme without introducing discontinuities or hard constraints.

Importantly, anchoring operates as a Soft Gauge Fixing mechanism applied to the reference group's center of mass. While individual scores are allowed to fluctuate freely, the **collective mean of the anchor group is softly tethered to a global reference constant (**μ_{ref}) to prevent system-wide drift. In doing so, the anchoring term introduces curvature along otherwise flat directions of the objective landscape, ensuring numerical stability and long-term interpretability.

Scope and Interpretation

The anchoring mechanism does not establish statistical identifiability, nor does it attempt to recover an absolute skill scale. Such objectives are ill-defined in closed systems without exogenous reference points. Rather, anchoring serves a strictly operational role: preventing unbounded global drift while preserving ordinal relationships under policy-constrained updates.

By decoupling drift control from both the data term and the non-demotion regularizer, the framework cleanly separates three distinct concerns:

1. **Outcome consistency** (pairwise comparison loss),
2. **Directional stability** (asymmetric non-demotion regularization),
3. **Global scale stability** (graph-based anchoring).

Together, these components yield a rating system that remains stable under long-term operation in closed populations, without resorting to arbitrary normalization or post-hoc correction.

3. System Architecture and Optimization Dynamics

This section describes how the policy-constrained rating framework introduced in Section 2 is realized as a stable, interpretable, and operational system. Crucially, we do **not** treat the model as an estimator of an underlying "true" latent skill. Instead, the system is designed to generate *effective rankings* that remain coherent under non-stationarity, asymmetric constraints, and strategic behavior.

Accordingly, the architecture and optimization dynamics are framed not in terms of statistical optimality, but in terms of **controlled update geometry**. The objective defined in Section 2 should be interpreted as a *surrogate potential* whose gradients define admissible rating motion under policy constraints. The resulting scores represent equilibrium points of constrained dynamics, not consistent estimators in the classical sense.

3.1 Architectural Separation: Update Generation vs. Behavioral Interpretation

The system adopts a strict separation of concerns between *rating update generation* and *behavioral interpretation*. This separation is intentional and foundational.

The framework is designed for post-hoc batch evaluation rather than real-time matchmaking. As such, priority is given to structural stability, identifiability, and resistance to pathological dynamics over raw computational throughput.

The system consists of two unidirectionally coupled engines:

Rating Engine (Update Generator)

The Rating Engine numerically evolves player scores by following the constrained gradient field induced by the objective in Section 2.3. Its sole responsibility is to:

- Aggregate match outcomes under contextual decomposition
- Apply graph-anchored reference stabilization
- Enforce asymmetric resistance induced by non-demotion policies

The engine does **not** attempt to infer psychological consistency, effort, or intent. Its output should be interpreted as a *policy-admissible score state*.

Metrics Engine (Behavioral Interpreter)

Operating strictly downstream, the Metrics Engine analyzes the match history together with the generated scores to compute behavioral indicators such as opponent diversity, volatility, and exploitation signals. These metrics are *diagnostic*, not corrective.

Importantly, no signal produced by the Metrics Engine feeds back into the Rating Engine. This prevents incentive leakage and Goodhart-style feedback loops, ensuring that the update dynamics remain invariant to downstream interpretations.

3.2 Constrained Optimization as Update Dynamics

The objective function defined in Section 2.3,

$$\mathcal{J}(\theta) = \mathcal{L}_{data}(\theta) - \lambda_{anchor} R_{anchor}(\theta) - \lambda_{demotion} R_{asym}(\theta)$$

is not optimized in pursuit of statistical consistency or maximum likelihood recovery. Instead, it serves as a *generating functional* whose gradient defines the admissible direction of score evolution.

Under this interpretation:

- Stationarity is not assumed
- Convergence is local and contextual
- Scores are meaningful only relative to the active policy geometry

Thus, the problem is more accurately described as **controlled score evolution on a constrained manifold** rather than parameter estimation.

3.3 Hybrid Update Dynamics: Decoupled Newton Step with Force Injection

To reconcile statistical estimation with strict operational constraints (e.g., non-demotion), we propose a **Hybrid Update Architecture**. Unlike standard optimization where all terms are normalized by the same curvature (Hessian), we decouple the update dynamics into two distinct components **governed by a unified gain scheduler**:

1. **Statistical Step (Δ_{stat}):** A **Diagonal Hessian-approximated Newton** update driven by match outcomes, normalized by the **diagonal upper bound of** local curvature.
2. **Policy Forces (F_{policy}): Additive Gradient Forces** (Anchor & Barrier) injected directly into the update step as **Designated Force Vectors** without Hessian normalization. Although this hybrid injection deviates from standard dimensional homogeneity, it is a deliberate heuristic design to ensure that safety constraints exert consistent pressure independent of data volume (curvature) while avoiding numerical instability in low-data regimes.

Finally, the **combined trajectory** is modulated by an **Adaptive Gain Scheduler (k_{dyn})** to reflect player momentum.

For each player i at iteration t, the update rule is:

$$\theta_i^{(t+1)} \leftarrow \theta_i^{(t)} + k_{dyn}(t) \cdot \left[\frac{\nabla \mathcal{L}_{data}}{H_{data} + \lambda_{damp}} - \lambda_{anchor} \nabla R_{anchor} - \lambda_{demotion} \nabla R_{demotion}\right]$$

Where H_{data} is the **Diagonal Hessian approximation** capturing the curvature of the match data, $k_{dyn}(t)$ is the Momentum-modulated gain coefficient that scales both statistical learning and policy enforcement.

3.4 Dynamic Sample Weighting Strategy

Competitive RTS environments exhibit 'Hot Hand' phenomena where players undergo periods of rapid skill evolution. Standard statistical models, constrained by high inertia (large Hessian denominators), often fail to track these sudden inflections. To resolve this lag, we replace static learning rates with an **Dynamic Sample Weighting Strategy (**w_k). This mechanism ensures that the model tracks evolving skill $\theta(t)$ by modulating the influence of each match based on recency and structural quality.

Instead of treating all history equally, it assigns higher importance to recent observations via **exponential decay** and systematically attenuates signals from excessive repeated interactions to prevent local overfitting. This ensures that the optimization landscape remains responsive to the latest performance trends while filtering out structural noise.

$$w_k = w_{decay}\left(\Delta t_k\right) \cdot w_{rematch}(n_k) \cdot w_{quality}$$

1. Temporal Decay ($w_{decay} = e^{-\lambda \Delta t_k}$)

Matches are weighted exponentially based on elapsed time Δt_k. This allows the model to track evolving skill ($\theta(t)$) rather than converging to a static lifetime average.

2. Repeated-Interaction Attenuation

$$w_{rematch} = (0.5)^{max(0, n_{pair} - N_{threshold})}$$

Where n_{pair} is the count of recent matches against the specific opponent. This systematically downweights structurally unreliable observations without complicating the generative model.

This suppresses the influence of excessive pairwise repetition, reducing farming-induced distortions. These weighting mechanisms do not encode behavioral judgments. They function solely as *signal conditioners* to preserve stability of the update dynamics. Explicit behavioral assessment is delegated entirely to the Metrics Engine (Section 7). However, while these adaptive gains optimize local convergence, they cannot inherently detect global system drift; this necessitates the graph-theoretic anchoring introduced next.

4. Graph-Theoretic Anchoring as Soft Gauge Fixing

A fundamental limitation of pairwise comparison models is **translational invariance**: the likelihood $P(i \succ j)$ depends only on the difference $\theta_i - \theta_j$. Without a fixed reference frame, the global scale of ratings can drift arbitrarily. Traditional implementations often address this by relying on "external calibration"—such as manually pinning specific reference players or enforcing active matchmaking constraints to guarantee connectivity.

However, as this framework operates as a **passive observational engine** on matches that have already occurred (rather than an active matchmaking system), it cannot force players to compete against specific opponents. Instead, we must discover the intrinsic structure within the **historical match logs**. We reject heuristic manual anchoring in favor of an **intrinsic, graph-theoretic solution**. However, in a closed ecosystem, absolute skill estimation is ill-posed. Therefore, our approach prioritizes Internal Ordinal Consistency over external calibration.

We analyze the topology of executed matches to identify **"Bridge Players"**—individuals who have structurally connected disparate sub-communities through their gameplay history—to serve as the system's **internal stability anchors**.

4.1 The Retrospective Match Graph

We model the history of player interactions as a graph $G_t = (V, E)$, where V represents the set of players active within the analysis window. Unlike a matchmaking queue, this graph is constructed **ex-post**. The weighted adjacency matrix **A represents the observed frequency of interactions:**

$$A_{ij} = \sum_{k \in Matches(i,j)} e^{-\lambda \Delta t_k}$$

In this topology, edges do not represent potential pairings but **verified relative score coupling**. Highly interconnected clusters represent competitive strata, while sparse edges represent the actual matches that have bridged these tiers in the past.

4.2 Robust Bridge Identification via PageRank Centrality

To select optimal anchors, we must distinguish between players who merely play *frequently* (High Degree) and players who play *integratively* across the network. A naive Eigenvector approach fails in fragmented topologies (e.g., disjoint skill islands), where the adjacency matrix becomes reducible.

We employ **PageRank Centrality** to identify topological bridges robustly. Unlike standard spectral methods, PageRank introduces a **Teleportation (Damping) Factor**, denoted as α (typically 0.85).

Let $PR(i)$ denote the centrality score of player i. The recursive definition is:

$$PR(i) = \frac{1-\alpha}{N} + \alpha \sum_{j \in In(i)} \frac{PR(j)}{L(j)}$$

Where $L(j)$ is the number of opponents player *j* has faced.

- **Damping Factor**(α)**:** Represents the probability of continuing to traverse match history chains, capturing transitive skill flow (like Eigenvector Centrality).
- **Teleportation Component** $(1-\alpha)$**:** Represents the probability of a "random jump," ensuring mathematical solvability even across disjoint sub-graphs.

High-PageRank nodes correspond to **"Structural Bridge Players"** who not only play frequently but connect otherwise isolated sub-communities, making them the operationally stable anchor candidates for global alignment.

4.3 Dynamic Anchor Set Construction

At each update interval, the system dynamically constructs the anchor set $\mathcal{A}$ using the following protocol:

1. **Graph Construction:** Build A from recent match history (e.g., last 30 days).
2. **Decomposition:** Compute PageRank scores c_i.
3. **Selection**: Select the top k% of players based on centrality scores c_i.

$$\mathcal{A} = \{i \in V \mid c_i \geq P_{100-k}(c)\}$$

This ensures that the anchor set is not arbitrary but is composed of the most structurally significant agents in the ecosystem.

4.4 Anchor Regularization Penalty

Once the set $\mathcal{A}$ is identified, we enforce global stability by constraining the center of mass of the anchors. Unlike heuristic "hard pinning," we formulate this as a Gaussian prior, yielding a policy-regularized objective functional.

The regularization term is defined as:

$$R_{anchor}(\theta) = \frac{\lambda_{anchor}}{2} \left(\frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \theta_i - \mu_{ref} \right)^2$$

Mathematically, this regularization acts as a soft gauge fixing condition. By penalizing collective deviation, it eliminates the translational singularity inherent in pairwise comparisons, ensuring a unique and stable solution without distorting relative skill distances.

4.5 Distinction from Post-hoc Normalization

A common alternative in traditional systems is "post-hoc normalization" (e.g., rescaling ratings after calculation to force a mean of 1500). We reject this approach.

- **Post-hoc methods** modify the parameters *after* optimization, which creates a disconnect between the maximized likelihood and the final reported values.
- **Our Anchor Prior** alters the **objective** function itself ($\mathcal{J}(\theta)$), thereby preventing uncontrolled global drift during score evolution.

This distinction is critical for non-convex or mixed-update settings. By incorporating the constraint into the gradient field, we ensure that context parameters (e.g., map bias) cannot absorb global drift, stabilizing convergence without violating the likelihood geometry.

4.6 Interpretation as Soft Gauge Fixing

In theoretical physics terms, the anchor mechanism can be interpreted as a **Soft Gauge Fixing** condition.

- **Hard Gauge Fixing:** Forces specific players to fixed values (e.g., $\theta_{playerA} \equiv 3000$). This simplifies calculation but introduces artificial rigidities that can distort the relative ranking if the anchor player's true skill fluctuates.
- **Soft Gauge Fixing (Ours):** Applies a restoring force to the *center of mass* of the PageRank anchors. This permits controlled collective movement when supported by overwhelming population-wide evidence, while preventing unbounded drift.

This formulation reconciles the need for an **internally consistent reference frame** with the dynamic, relativistic nature of skill in a competitive ecosystem.

5. Contextual Inference and Bias Disentanglement

In standard rating systems, extrinsic factors such as map imbalances or race advantages are treated as aleatoric noise. This approach is flawed in RTS environments, where such factors exert a systematic, non-zero influence on outcomes. If unmodeled, these biases are erroneously absorbed into the player's baseline score(θ), leading to "inflation by luck" (e.g., a player rising solely by playing on a favored map).

Our framework addresses this by treating the match outcome as a composite signal. We employ a **Generalized Linear Model (GLM)** structure to mathematically disentangle latent skill from environmental biases.

5.1 The Additive Bias Model

As introduced in Section 2, the linear predictor η_{ij} decomposes the log-odds of victory into three distinct components:

$$\eta_{ij} = \underbrace{(\theta_i - \theta_j)}_{\text{Skill Diff}} + \underbrace{\beta_{map}}_{\text{Global Bias}} + \underbrace{\left(\rho_{i,r_j} - \rho_{j,r_i}\right)}_{\text{Interaction Term}}$$

1. **Latent Skill(θ):** context-normalized baseline score under policy constraints.
2. **Global Bias (β):** Parameters shared across the entire population (e.g., Map ***M***'s win rate deviation and match type bias).
3. **Interaction Term (ρ):** Player-specific proficiency in specific contexts (e.g., Player ***A***'s specific performance against Zerg).

This decomposition ensures that a victory achieved under favorable conditions (high β) yields a smaller update to θ compared to a victory under neutral conditions.

5.2 Non-Stationary Adaptation via Contextual RMSProp & Meta-Cyclic Decay

The update dynamics for context parameters differ fundamentally from those for player skill. While player skill scores are intentionally allowed to remain volatile and non-stationary under policy constraints, global context effects (e.g., map-specific biases) serve as long-horizon correction terms whose updates should progressively attenuate as evidence accumulates.

As an update-inertia control mechanism (i.e., implicit learning-rate annealing), we employ an **RMSProp-style Adaptive Gradient algorithm** specifically for context parameters (Map β_{map}, Match type β_{type}) and race interaction (ρ) parameters, distinct from the player score (θ) updates.

The update rule for a context weight w (e.g., a map bias) is given by:

$$w_{t+1} = w_t + \frac{\eta}{\sqrt{G_t + \epsilon}} \cdot \nabla_w \mathcal{L}$$

Where $G_t = \sum_{\tau=1}^{t} (\nabla_w \mathcal{L}_\tau)^2$ is the sum of squared historical gradients.

- **Rare Contexts:** For newly added maps or rare matchups, G_t is small, resulting in higher update responsiveness and rapid incorporation into the score system.
- **Established Contexts:** For frequently observed contexts, accumulated gradients increase update inertia, naturally damping short-term fluctuations without requiring explicit learning-rate schedules.

5.2.1 Handling Meta-Cyclicity in Static Domains

While standard accumulation methods assume convergence to a static regime, the StarCraft ecosystem presents a unique challenge: it operates on a 'frozen' physics engine, yet the strategic meta is cyclic and evolutionary. Under these conditions, strictly accumulating gradients indefinitely ($G_t \to \infty$) causes the learning rate to vanish, rendering the system unresponsive to long-term strategic shifts.

To resolve this, **we apply a periodic Decay Factor** to the accumulated squared gradients (G_t), effectively adopting **a Periodic Decay Strategy (applied between batch updates**) to prevent learning rate saturation while maintaining stability within the update window**.** Additionally, a complete reset of context parameters is enforced explicitly upon seasonal transitions. This modification preserves the stability benefits of Adagrad while preventing system rigidity, effectively reducing the influence of obsolete data to maintain adaptivity to new trends.

Furthermore, distributional shifts in observed contexts arise primarily from changing battlefields (New Maps). Our system addresses this structurally via **Feature Expansion**. When a new map enters the pool, it is initialized as a fresh feature with $G_{map} = 0$. This automatically assigns a maximally high learning rate to the new context, ensuring instant adaptivity without destabilizing the long-term confidence of established race matchups ρ.

5.3 Race Matchup Dynamics as Baseline-Preserving Interaction Terms

RTS games feature asymmetric factions (e.g., Terran vs. Zerg), where balance is neither symmetric nor transitive. Consequently, a global “Race Win Rate” is insufficient: individual players exhibit persistent, matchup-specific proficiencies (e.g., a player may consistently excel in TvZ while underperforming in TvP), which cannot be absorbed into a single population-level bias without distorting skill estimates.

We model this effect via the interaction term $\rho_{i,r}$, representing Player *i*'s specific deviation from their baseline rating when facing Race r. These terms capture player-conditioned contextual effects, not independent skill dimensions.

- **Interpretation:** If $\rho_{i,\text{Zerg}} > 0$, Player i performs better against Zerg opponents than their baseline rating θ_i would predict under neutral conditions.
- **Zero-Sum Constraint:** To ensure identifiability and preserve the semantic role of θ_i, we enforce $\sum_r \rho_{i,r} \approx 0$ (hard constraint), ensuring that θ_i remains the context-averaged skill level of player i, while matchup advantages are localized within the interaction terms rather than leaking into the baseline score.

5.4 Bounded Regularization for Stability

To prevent "exploding parameters" where the model attributes the entire outcome to extreme context bias (e.g., claiming a map has a 99% win rate to explain a upset), we apply **Bounded Regularization**.

Context parameters are constrained to a physically plausible interval (e.g., $\beta \in [-0.5, +0.5]$, corresponding to a $\approx \pm 12\%$ shift in win probability). This acts as a **Uniform Prior**, reflecting the domain knowledge that while game balance is imperfect, it is rarely broken beyond a certain magnitude. This explicit bounding prevents degenerate solutions where the model "explains away" all skill differences as environmental factors.

5.5 Comparative Analysis with Classical Frameworks

The proposed Context-Aware Framework generalizes the Bradley-Terry model while retaining probabilistic rigor. The following table contrasts our approach with traditional rating systems used in competitive gaming.

Feature	Elo / Glicko	TrueSkill	Our Framework
Time Modeling	Implicit (K-factor)	Dynamic Variance	Policy-Weighted Temporal Attenuation
Context Awareness	None	None	Explicit Disentanglement
Reference Frame Stability	Implicit (Mean Reversion)	Implicit	Graph-Theoretic Anchoring
Constraint Handling	Hard Clipping (Heuristic)	None	**Asymmetric Soft Priors**

In summary, while classical models struggle to distinguish between skill evolution and environmental noise, our framework explicitly models these components, ensuring that ratings reflect **intrinsic ability** rather than contextual luck.

6. Barrier-Mediated Non-Demotion via Asymmetric Logarithmic barrier with gradient clipping

This section completes the model specification by formalizing the non-demotion policy as a topological constraint on the optimization manifold. While Sections 2–5 define the probabilistic structure and identifiability of the rating system, Section 6 addresses how policy-level irreversibility is embedded into the learning dynamics without sacrificing differentiability or solver compatibility.

We resolve this by modeling non-demotion not merely as a resistance zone, but as a constrained boundary condition embedded in the prior distribution.

By applying Gradient Clipping to the barrier gradient, we implement a Physically Saturated Logarithmic Barrier. This creates a restoring force that grows asymptotically toward a strictly enforced engineering limit (F_{max}) as a player's rating converges to their protected tier floor, operationally ensuring that the boundary cannot be breached by empirical evidence alone while preserving differentiability within the feasible region.

Although implemented as a prior-like penalty, the barrier does not encode probabilistic belief about player skill. Instead, it functions as a policy-imposed feasibility constraint, shaping the optimization manifold while leaving the likelihood model intact.

6.1 The Dynamic Hard-Soft Ratchet Mechanism

We implement the non-demotion policy as a **Saturated Barrier Mechanism (functioning as a dynamic ratchet).** Just as a mechanical ratchet allows motion in one direction while resisting return, our system dynamically updates the protected floor based on cumulative achievement.

Let $\mathcal{T}(k)$ denote a mapping function that converts a continuous score to the **lower bound of its corresponding tier** (e.g., mapping any score in 1000…1099 to 1000). The protected limit for player i, denoted as $\theta_{limit}^{(i)}$, is defined as the maximum discrete tier threshold the player has ever established:

$$\theta_{limit}^{(i)} = \max_{t \le now} (\mathcal{T}(\theta_i(\mathrm{t})))$$

Instead of implementing this as a pure logical check (Hard), which breaks differentiability, or a purely potential field (Soft), we model it as a **Hard-Soft Ratchet**.

This creates a continuous restoring force that opposes downward movement only when the score approaches this **discrete floor**, allowing free fluctuation within the tier itself.

6.2 Repulsive Force with Hard Saturation

To enforce the ratchet boundary, we first define the **Theoretical Potential Field** using an Asymmetric Logarithmic Barrier centered at the dynamic limit:

$$R_{asym}(\theta_i) = -\lambda_{demotion} \ln\left(max\left(\epsilon, \theta_i - \theta_{limit}^{(i)}\right)\right)$$

Note: For $\theta_i \leq \theta_{limit}^{(i)}$, the gradient is clamped to a fixed saturation limit F_{max} to prevent singularity.

In implementation, tier history is represented in continuous skill space via a monotonically increasing max-θ tracker. **Theoretically**, this formulation creates a mathematical singularity at the boundary where the penalty value approaches infinity as $\theta \rightarrow \theta_{limit}$. However, in our engineering realization, we do not optimize this potential directly. Instead, we define the Repulsive Force as the negative gradient of the potential ($F = -\nabla R_{asym}$). Thus, the Raw Repulsive Force is derived as follows and subjected to Hard Saturation.

$$F_{raw} = \frac{\lambda_{demotion}}{distance}$$

$$F_{applied} = min(F_{raw}, F_{max})$$

Operational Interpretation: This saturation transforms the system into a **Hard-Soft Barrier**:

1. **Soft Engagement (d ≫ 0):** When approaching the floor, the force scales inversely with distance, providing a smooth warning signal to the optimizer.
2. **Hard Locking (d → 0):** Upon reaching the critical proximity, the force hits the **Hard Saturation** limit (F_{max}).

   Since F_{max} is calibrated to be orders of magnitude larger than typical data gradients ($\nabla \mathcal{L}_{data}$), the floor is effectively locked.

   This **Hard-Soft** approach guarantees that the boundary remains operationally impermeable (Hard) while avoiding the numerical explosion of pure singularities (Soft).

Engineering Constraint (Gradient Clipping):

To prevent numerical overflow (*NaN*) near the singularity where the force approaches infinity, we implement a **Physical Safety Fuse.** The applied restoring force is strictly capped at a maximum engineering constant $F_{\max}$ (e.g., 50.0). Since $F_{\max}$ is chosen to be an order of magnitude larger than typical data gradients, the barrier remains effectively impenetrable while preserving the numerical stability required for the MM solver.

In practice, the data likelihood gradient is uniformly bounded due to the logistic link function, ensuring that the barrier gradient dominates in the limit.

6.3 Topological Consistency vs. Heuristic Truncation

Why implement a complex logarithmic barrier instead of a simple logical check (if rating < floor: rating = floor)? The answer lies in the **topology of the optimization landscape**.

Hard truncation introduces a non-differentiable "kink" in the objective function. At the boundary, the gradient is undefined, and the curvature (Hessian) instantly collapses to zero. This discontinuity is catastrophic for second-order solvers like the MM algorithm (Section 3.2), which rely on consistent curvature information to scale update steps.

By contrast, our approach **redefines the feasible manifold**. The Logarithmic Barrier maintains C^{∞} differentiability within the feasible set, ensuring that the solver always perceives a valid, informative gradient field. Instead of hitting a wall and stopping (loss of information), the parameter encounters an asymptotically increasing restoring force that naturally guides it back to the safe region.

Feature	Hard Truncation (Naïve)	Infinite Logarithmic Barrier (Ours)
Mathematical Nature	Discontinuous (Non-differentiable)	Smooth (C^{∞} Differentiable)
Optimization Geometry	Creates a "Cliff" (Undefined Gradient)	Creates an "Asymptotic Barrier" (Unbounded Gradient)
Information Flow	**Decoupled:** Data is ignored at boundary	**Bounded:** Resistance mathematically dominates Data
Solver Compatibility	Breaks Hessian/Newton methods	**Fully Compatible** with Curvature-aware Solvers

In essence, Hard Truncation treats non-demotion as an **external rule** applied *after* the math, whereas the Logarithmic Barrier integrates it as an **intrinsic boundary condition** *within* the math. This ensures that the resulting ratings corresponds to a stable equilibrium point of a policy-regularized update process, which admits a MAP-like interpretation but does not rely on an explicit posterior formulation.

6.4 Momentum-Modulated Gain & Veteran Breakout Protocol

In the final assembly of the algorithm, the update magnitude is controlled not by static hyperparameters, but by a state-dependent **Adaptive Gain Scheduler** coupled with a **Dual-Stage Damping Protocol**. This ensures that the system remains responsive to "Hot Hand" streaks while suppressing noise for established veterans.

Adaptive Gain with Veteran Breakout (k_{dyn})

Instead of a fixed learning rate, we define the dynamic gain coefficient $k_{dyn}(t)$ as a composite function of three state variables: Momentum(ψ_{form}), Tier Saturation (θ), and Stability (Σ).

The formulated gain equation is:

$$k_{dyn}(t) = \eta_{base} \cdot \underbrace{\left(1 + \nu_{outcome} \cdot \tanh(\psi_{form})\right)}_{Momentum\ Boost} \cdot \underbrace{\frac{1}{1 + e^{\gamma(\theta - \theta_{star})}}}_{Tier\ Damping} \cdot \underbrace{\gamma_{vol}/\Sigma}_{Stability\ Control}$$

Where:

1. **Momentum Boost $\left(\mathbf{1} + \boldsymbol{\nu_{outcome}} \cdot \mathbf{tanh}(\boldsymbol{\psi_{streak}})\right)$:** We apply a hyperbolic tangent function to the weighted streak index. This creates a bounded acceleration factor that scales linearly for small streaks but saturates for extreme outliers.

$$\nu_{outcome} \begin{cases} \nu_{win} & if\ \psi \geq 0 \\ \nu_{loss} & if\ \psi < 0 \end{cases}$$

   If a player is on a winning streak (ψ > 0), the gain increases, allowing them to overcome the inertial resistance of the Hessian. The tanh function physically limits the boost, preventing runaway inflation even during extreme win streaks.

2. **Tier Damping (Logistic Decay):** As a player's rating θ approaches the theoretical high-tier threshold (θ_{star}), the term $\frac{1}{1+e^{\gamma(\theta-\theta_{star})}}$ induces a smooth decay in update speed. This models the "Soft Ceiling" effect, reflecting the increasing difficulty of marginal skill improvements at the highest competitive levels.
3. **Stability Control with Breakout Override:** Normally, high stability reduces the gain (multiplier ≈ 0.5) to protect established ratings from noise. **However, this suppression is conditional.** If a player's momentum exceeds a context-aware **Breakout Threshold**, the system identifies a 'Breakout State' and **overrides the penalty**, boosting the gain (multiplier ≥ 1.2) to facilitate rapid skill convergence.

Dual-Stage Hessian Damping Protocol

While $k_{dyn}(t)$ modulates the global update magnitude, the system also applies a discrete toggle to the Hessian Damping term (λ_{damp}) in the core update equation. This acts as a secondary "gear-shifting" mechanism based on the intensity of the player's momentum.

$$\lambda_{damp}(t) = \begin{cases} \lambda_{base}\ if\ \psi_{streak} < \tau_{boost} & (Stability\ mode) \\ \lambda_{accel}\ if\ \psi_{streak} \geq \tau_{boost} & (Acceleration\ mode) \end{cases}$$

- **Stability Mode ($\boldsymbol{\lambda_{base}}$ ≈ 0.5):** Under normal conditions, the system prioritizes noise reduction and trust. The higher damping acts as a "heavy flywheel," smoothing out variance.

- **Context-Aware Acceleration (Veteran Breakout):** To prevent rating stagnation, the momentum threshold τ_{boost} required to trigger acceleration is **adaptive**. (Standard : $\tau \approx 4.0$, Veteran : $\tau \approx 2.5$) This asymmetry ensures that verified veterans require less inertial energy to trigger a breakout, allowing them to overcome historical inertia and climb quickly when their skill genuinely improves.

7. The Metrics Engine: Behavioral Topology and Anti-Gaming

A single scalar rating θ captures latent skill, but is insufficient to describe the full narrative of a player's journey. It fails to distinguish between a "stable veteran" and a "volatile rookie," or between "empty grinding" and "impactful winning."

To address this, the Metrics Engine operates downstream of the Rating Engine, deriving a suite of secondary indicators. These metrics observe the principle of **Unidirectional Integrity**: they consume the rating model's outputs to generate insights (e.g., "Clutch", "Momentum") but never feed back into the rating likelihood, ensuring that the statistical definition of skill remains uncontaminated by gamification incentives.

7.1 Entropy-Based Anti-Gaming (Quality Control)

To prevent "farming" (inflating stats by repeatedly playing weak or colluding opponents), we introduce **Opponent Entropy (H_{opp})**. For a player i, let p_{ij} be the proportion of their matches played against opponent j. The Shannon Entropy of their interaction history is:

$$H_{opp}(i) = -\sum_{j \in Opponents(i)} p_{ij} \ln p_{ij}$$

- **Low Entropy ($H \approx 0$):** Indicates "farming" (playing only 1-2 people).
- **High Entropy ($H \gg 0$):** Indicates healthy ecosystem participation.

The Metrics Engine monitors this value. The player’s 'Effective AP' accumulation is scaled continuously by a Quadratic Diversity Factor based on $H_{opp}(i)$, naturally reducing incentives for farming, neutralizing the incentive for abuse without requiring arbitrary bans or hard-coded limits.

7.2 Momentum Index (Form Analysis)

While θ estimates long-term skill, stakeholders require insight into short-term performance deviations ("Is the player on a hot streak?"). We formulate this in two stages: **Signal Extraction** and **Metric Normalization**.

Step 1: Raw Performance form (ψ_{form}**)** First, we compute the exponentially weighted sum of prediction residuals. This captures the cumulative "surprise" of recent match outcomes **(distinct from a simple consecutive win count)**:

$$\psi_{form}^{(i)} = \sum_{k \in Recent} e^{-\lambda(t-t_k)} \cdot \left(y_{ik} - P\left(i > j \middle| \theta_i, \theta_j\right)\right)$$

(This corresponds to the "Raw Weighted Residual" formula.)

Step 2: Momentum Index Mapping

To translate this unbounded statistical signal into an interpretable user-facing metric, we apply a bounded hyperbolic tangent transformation:

$$Momentum_i = 50 + 50 \cdot \tanh\left(\frac{\psi_{form}^{(i)}}{\sigma_{scale}}\right)$$

- **Center (50):** Represents neutral form (performing exactly as expected).
- **Scale (50):** Constrains typical fluctuations to the $[0,100]$ range.
- **Saturation:** The $tanh$ function ensures that even extreme outlier streaks do not break the visualization scale, naturally modeling the diminishing returns of "hot hands."

7.3 Performance Over Expectation (PoE)

While the Rating Engine predicts the *outcome* (Win/Loss), it does not quantify the *quality* of that outcome. **PoE** measures whether a player is systematically outperforming the model's probabilistic expectations, effectively serving as a "Smurf (Players who hide record) Detector" or "Carry Index."

For each match k, let $y_{ik} \in \{0,1\}$ be the actual outcome and P_{ik} be the predicted win probability. The PoE score is the averaged residual:

$$PoE_i = \frac{1}{N}\sum_{k=1}^{N}(y_{ik} - P_{ik})$$

- **Positive PoE (> 0):** The player consistently wins matches where they were at a disadvantage (e.g., unfavorable map or race matchup). This indicates "Skill Efficiency" beyond what θ captures.
- **Negative PoE (< 0):** The player loses matches they were statistically expected to win.
- **Zero PoE (≈ 0):** The player performs exactly as their rating predicts.

7.4 Stability Index (Trust & Experience)

The **Stability Index** answers a different question: *"How confident are we that the current rating θ_i is accurate?"* A placement-match rookie and a 5-year veteran might share the same MMR (e.g., 1500), but they are statistically distinct. We define Stability as a composite of **Volume** and **Precision**:

$$Stability_i = 100 \cdot \underbrace{\left(1 - e^{-\lambda \cdot N_{matches}}\right)}_{Volume\ Term} \cdot \underbrace{\left(1 - \frac{\sigma_i}{\sigma_{base}}\right)}_{Precision\ Term}$$

- **Volume Term:** As match count N increases, this term approaches 1. This prevents new accounts from claiming "High Stability" regardless of their win rate.
- **Precision Term (Approximated):** While theoretically derived from the inverse Hessian of the Fisher Information Matrix ($\sigma_i^2 \approx \mathrm{diag}(\mathcal{H}^{-1})$), theoretical curvature alone may underestimate the realized volatility of match outcomes. Therefore, we employ a **heuristic approximation** based on outcome volatility and match frequency.

$$\text{Precision Term} = \max(0, 1 - \sqrt{\frac{ObservedVariance}{BaseVariance}})$$

This proxy effectively captures the certainty of the rating without incurring the $O(N^3)$ cost of full Hessian inversion.

Operational Usage:

- **High Stability (> 80): "Verified Veteran."** The player's rating is robust.
- **Low Stability (< 40): "Provisional."** The rating is volatile.
- Unlike systems that freeze the rating itself (making it hard to climb), we decouple this into a separate metric. This allows the Rating Engine to remain **responsive** to skill changes, while the Stability Index transparently informs users about the **certainty** of that rank.

7.5 Clutch Performance Score (Bayesian Shrinkage)

Not all matches are created equal. A victory in a tournament final or a "Decider Match" carries significantly more weight in defining a player's legacy than a routine ladder game. The **Clutch Score** isolates performance in these high-pressure contexts.

However, a naive win-rate calculation ($\frac{\text{Wins}}{\text{Total}}$) is flawed for rare events. A player who plays 1 major final and wins it would have a 100% Clutch Score, unfairly ranking above a veteran who won 8 out of 10 finals (80%). To correct this small-sample bias, we apply **Bayesian Shrinkage**:

$$Clutch_i = \frac{W_{wins} + \alpha_0}{W_{total} + \alpha_0 + \beta_0} \times 100$$

Where $W_{total} = \sum w_g$ and $W_{wins} = \sum w_g \cdot \mathbb{I}(\text{win})$, with w_g representing match importance and recency weights.

- **Prior Belief (α_0, β_0):** We employ a **Beta (2,2) Conjugate Prior,** introducing 'pseudo-counts' that represent a weakly informative baseline (50% win rates) to prevent small-sample volatility.
- **Shrinkage Effect:**
  - For a player with 1 win / 1 match: Score becomes $\frac{1+2}{1+4}$ = 60% (Regressed heavily toward the mean).

- For a player with 8 wins / 10 matches: Score becomes $\frac{8+2}{10+4}$ = 71.4% (Data dominates the prior).
- **Interpretation**: This metric identifies "Proven Big-Game Hunters"—players who have demonstrated sustained excellence under pressure—while filtering out "Lucky One-Hit Wonders."

7.6 Activity Point (AP) System

While θ measures *skill*, the **Activity Point (AP)** system measures *contribution*. Its goal is to reward ecosystem participation without inflating the skill rating. However, simply summing up match counts encourages "spamming" (playing short, low-quality games) or "farming."

We implement the AP system with **Topological Gating** and **Diminishing Returns**:

$$AP_i = C_{scale} \cdot \ln\left(1 + \underbrace{\left(\frac{H_{opp}(i)}{H_{max}}\right)^2}_{Quadratic\ Diversity\ Factor} \cdot \sum_{k \in Matches} \left(w_{quality}^{(k)} \cdot \frac{1}{1 + \gamma \cdot N_{daily}^{(k)}}\right)\right)$$

Where C_{scale} is a calibration constant determining the display range (e.g., 0–1000) of the Activity Points.

Note: While the Diversity Index ratio ($\frac{H_{opp}}{H_{max}}$) is theoretically base-invariant, we adopt the **Natural Logarithm (ln)** for entropy calculations to align with the standard library implementation and the AP scaling formula.

1. Quadratic Diversity Factor: A simple hard-threshold approach is susceptible to "edge-riding," where players maintain just enough diversity to bypass the gate. To address this, our framework introduces a Quadratic Diversity Factor. As the entropy ratio of the opponent distribution decreases, the AP acquisition efficiency decays quadratically. This mechanism naturally incentivizes players to engage with a diverse pool of opponents rather than farming a specific subset.

2. Daily Diminishing Returns (N_{daily}):

- The term $\frac{1}{1+\gamma N_{daily}}$ ensures that the 1st match of the day yields maximum points, while the 20th match yields significantly less.
- This promotes **"Healthy Consistency"** (playing every day) over **"Unhealthy Grinding"** (playing 50 games in one day), aligning system incentives with player well-being.

3. Match Quality ($w_{quality}$):

- Rewards are scaled by match type weights (e.g., Bo5 vs Bo7) and rating proximity.

7.7 System Coherence and Comparative Advantage

The Metrics Engine is not merely a collection of disparate statistics, but a **coherent analytical layer** designed to act orthogonally to the Rating Engine. By enforcing strictly unidirectional information flow, we ensure that the "Gamification" elements (AP, Momentum) never corrupt the "Scientific" elements (θ, Stability).

7.7.1 Logical Orthogonality (Coherence)

Each metric derived in this section targets a distinct dimension of player behavior, utilizing specific signals from the core model without introducing redundant degrees of freedom.

Metric	Source Signal	Physical Meaning	Target Question
Skill (θ)	Latent Ability (MAP on constrained manifold)	**Skill** (Latent Ability)	"Who is likely to win?"
Stability	Hessian Curvature (∇^2)	**Confidence** (Inertia)	"Is this rating accurate?"
Momentum	Recent Residuals ($y - P$)	**Form** (Velocity)	"Is the player on a streak?"
PoE	Cumulative Residual Bias	**Efficiency** (Luck/Skill)	"Did they overperform expectations?"
Entropy	Opponent Distribution	**Topology** (Diversity)	"Is this organic play or farming?"

This separation ensures **Model Coherence**: A player cannot artificially boost their Skill Rating (θ) simply by grinding more games (which affects AP) or by inflating their streak (which affects Momentum). Each attribute is measured independently.

7.7.2 Comparative Analysis with Classical System

The proposed framework represents a paradigm shift from "Scalar Rating Systems" to "Multidimensional Behavioral Systems." The following table contrasts our architecture with standard industry baselines.

Feature	Elo / Glicko	TrueSkill	Proposed Framework
Core Estimator	Point Estimate	Gaussian (μ, σ)	Region-Constrained **MAP**
Non-Demotion	Impossible (Mathematically)	Not Supported	Soft Ratchet Barrier (Sec. 6)
Form Tracking	Implicit (Variance)	Dynamic Variance	Explicit Momentum Index
Context Awareness	None	None	Disentangled Bias (β, ρ)
Anti-Gaming	None (Manual Bans)	None	Entropy-Based Gating
Engagement	Raw Match Count	None	Diminishing Returns AP

Conclusion of the Metrics Engine: By layering these interpretability metrics on top of a robust, anchor-constrained estimator, the system solves the "Accuracy vs. Engagement" trade-off. It provides the **scientific rigor** required for competitive integrity while simultaneously offering the **rich feedback loops** necessary for a modern, engaging user experience.

8. Conclusion: Toward a Sustainable Competitive Ecosystem

The overarching goal of this framework is to resolve the fundamental tension in competitive gaming: the conflict between **Scientific Accuracy** and **Player Engagement**.

Classical rating systems (Elo, Glicko, TrueSkill) prioritize statistical convergence above all else, often resulting in "Ladder Anxiety," rating stagnation, and a disconnect between a player's perceived effort and their visible rank. Conversely, purely progression-based systems (XP, Stars) sacrifice competitive integrity for retention, leading to inflation and meaningless rankings.

We have presented a **third way**: A **Constraint-Regularized, Context-Aware, Multi-Objective Architecture**.

8.1 Summary of Innovations

1. **Structural Stability:** By replacing manual anchoring with **PageRank Centrality (Section 4)**, the system autonomously identifies the "Truth" of the skill distribution from the graph topology itself, forcing stability without ad-hoc or manually tuned constraints.
2. **Operational Reality:** The **Asymmetric Ratchet Barrier (Section 6)** acknowledges the psychological reality of "Tier Safety" and mathematically integrates it as a **singular boundary condition** into the gradient dynamics. This proves that non-demotion can coexist with rigorous probability models if treated as an asymmetric prior rather than a heuristic hack.
3. **Contextual Fairness:** Through **GLM Disentanglement (Section 5)**, the system fairly evaluates matches by isolating skill from map imbalances and race matchups, ensuring players are judged on ability, not lobby luck.
4. **Behavioral Integrity:** The **Entropy-Gated Metrics Engine (Section 7)** introduces a sophisticated layer of policing that rewards genuine engagement while mathematically neutralizing farming and abuse.

8.2 Expected Impact on the Ecosystem

Implementation of this framework transforms the ladder experience:

- **For the Rookie:** The *Stability Index* and *Ratchet Barrier* provide a safe environment to learn without the fear of immediate plummeting, fostering retention.
- **For the Veteran:** The *Context Awareness* and *PoE* metrics validate their mastery, distinguishing them from lucky climbers.
- **For the Community:** The *Entropy* and *Clutch* metrics provide a narrative richness, enabling richer qualitative narratives grounded in quantitative metrics, such as identifying rising stars, consistent grinders, and clutch performers.

8.3 Final Remarks

In conclusion, this whitepaper proposes that a rating system should be more than a mere sorter of skill; it should be an **engine for ecosystem health**. By strictly decoupling estimation from interpretation and enforcing topological constraints, we achieve a system that is **mathematically sound, operationally robust, and intrinsically engaging**.

This architecture sets a new standard for competitive RTS environments, proving that we do not need to choose between the math of the statistician and the fun of the gamer—we can, and must, have both.

Appendix A. Scope of Framework & Design Trade-offs

This framework intentionally departs from several classical statistical assumptions in order to prioritize operational viability, ecosystem stability, and long-term engagement. The following design choices are deliberate architectural decisions rather than theoretical oversights.

1. **Adaptivity over Asymptotic Optimality**
   - Competitive environments are inherently non-stationary. The framework, therefore, favors adaptive responsiveness to skill evolution and meta shifts (e.g., via adaptive learning rates) over asymptotic optimality under static assumptions.
2. **MAP Efficiency over Full Posterior Inference**
   - Full Bayesian inference (e.g., MCMC) is infeasible for real-time systems. We adopt a Region-Constrained MAP formulation, balancing probabilistic rigor with scalability and latency requirements.
3. **Operational Constraints as Priors**
   - Social requirements (e.g., non-demotion) are not treated as heuristic hacks, but as **Differentiable Priors** (e.g., Asymmetric Ratchet Barrier). This integrates "fun" directly into the gradient dynamics without breaking mathematical consistency.
4. **Ecosystem Stability over Raw Estimation Speed**
   - We deliberately dampen volatility for established players (via Stability Index). We prioritize **long-term institutional trust** in the ranking system over the aggressive maximization of short-term information extraction.
5. **Statistical Interpretation Limitations**
   - Because the model incorporates asymmetric priors and policy-driven constraints, the resulting skill estimates should not be interpreted as **consistent estimators** of an underlying ground-truth skill. Parameters governing demotion resistance and anchoring are not statistically identifiable and must be selected through empirical validation and domain considerations.

Appendix B. Mathematical Stability Proofs

Claim (Suppression of Dominant Shift Mode):

Formally, the likelihood surface of pairwise comparisons contains a one-dimensional null space along the direction $\mathbf{v} = (1, 1, \cdots, 1)$, corresponding to global shift invariance. The anchor prior introduces strictly positive curvature along this dominant direction.

$$v^T(\nabla^2 R_{anchor})v > 0$$

By penalizing deviations along this specific vector, the prior effectively eliminates the singular mode of the Hessian matrix without distorting the relative skill distances (local geometry) between players. This ensures bounded evolution even if the underlying graph topology is sparse, provided the anchor set $\mathcal{A}$ is spectrally connected.

We note that the strength of the anchor prior (λ_{anchor}) is **not identifiable** from match outcomes alone and should be interpreted as a policy-level hyperparameter rather than a statistically estimated quantity.

Claim (Unbiasedness):

Under a correctly specified model, the expected PoE is zero ($E[PoE] = 0$). Therefore, a significantly non-zero PoE serves as a strong signal of model misalignment (e.g., a rapidly improving player) or external factors (e.g., luck).