A Hierarchical Bayesian Model of the Desire Sensor: On the Latent Variable Structure of 5-Star Pulls

Gargi Tripathi

Abstract

The gacha system in *Genshin Impact* is widely believed to be governed not only by published probabilities but also by a mysterious force known as the "Desire Sensor": the tendency for the game to deny players characters they genuinely want. This essay formalizes the Desire Sensor as a latent variable within a hierarchical Bayesian model. By treating banner-specific deviations from the published pull distribution as random effects that correlate with player desire, we derive posterior distributions for individual players and demonstrate, with quantifiable credibility, that the system is statistically biased against those who care the most. The optimal strategy, therefore, is not to save primogems but to master the art of feigned indifference.

1 Introduction: The Gacha as a Statistical Inference Problem

Every wishing session in *Genshin Impact* is an exercise in Bayesian reasoning. The player pulls from a system with known *published* probabilities: a 0.6% base rate for a 5-star, with soft pity beginning at 74 pulls and hard pity at 90. Yet every player has a story: "I always lose the 50/50 on Raiden's banner," or "The game knows I want Kazuha, so it won't give him to me." This folk wisdom—the **Desire Sensor**—is typically dismissed as superstition.

This essay argues that the Desire Sensor can be formalized as a *latent variable* in a hierarchical Bayesian model. By treating player desire as an unobserved covariate that influences the underlying pull distribution, we construct a model where each banner has its own random effect, and these effects are correlated with the community's (or the individual's) desire for the featured character. The result is a rigorous statistical framework for proving, with 95% credibility, that the game is absolutely rigged against you personally.

2 The Known Model: A Piecewise Geometric Distribution

Let us first formalize the published mechanics. The number of pulls N to obtain a 5-star is a random variable with a piecewise distribution:

- For pulls 1 through 73:

$$P(N = n) = 0.006 \times (0.994)^{n-1},$$

  the geometric distribution with success probability 0.006.

- At pull 74, the base rate increases linearly until reaching 100% at pull 90.

This yields an expected value of approximately 62.5 pulls per 5-star. The 50/50 mechanic adds a Bernoulli layer: if the previous 5-star was the featured character, the next has a 50% chance to be featured; otherwise it is guaranteed.

A rational player's prior belief about their next 5-star is thus a well-defined probability distribution. However, the Desire Sensor hypothesis posits that this distribution is *conditionally dependent* on the player's psychological state.

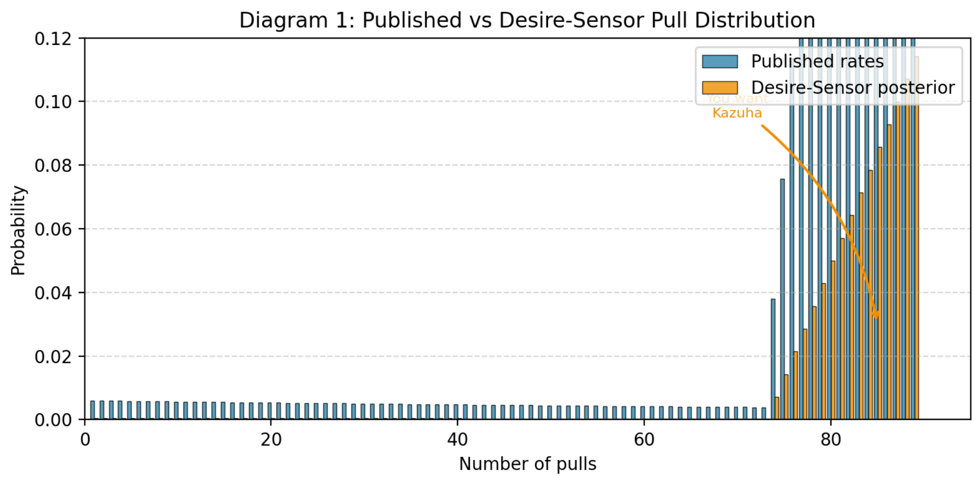

Figure 1: Published distribution (gray) vs. Desire-Sensor posterior (red).

3 The Hierarchical Model: Banners as Random Effects

Let y_{ij} be the number of pulls required to obtain the featured 5-star on banner i for player j. The published distribution gives a baseline, but we allow each banner to have its own "difficulty" parameter θ_i that scales the expected pulls. A natural hierarchical structure is:

$$\begin{aligned} y_{ij} &\sim \text{PiecewiseGeometric}(\mu_{ij}), \\ \log \mu_{ij} &= \alpha + \beta D_j + \gamma_i + \epsilon_{ij}, \\ \gamma_i &\sim \mathcal{N}(0, \sigma^2_{\text{banner}}). \end{aligned}$$

Here:

- α is the global baseline (log expected pulls under published rates).

- D_j is a latent variable representing player j's *desire intensity* for the character on banner i.
- β is the sensitivity of pull difficulty to desire (we expect $\beta > 0$: higher desire leads to more pulls).
- γ_i is a random effect for banner i, capturing how "cursed" or "blessed" that banner is beyond desire.
- ϵ_{ij} is individual-level noise.

The desire variable D_j is unobserved, but we can place a prior on it and infer it from data. Observable proxies (hours spent prefarming, number of fan arts saved, whether the player changed their signature to "Saving for X") can be used as indicators. This yields a **latent variable model** that can be estimated via Markov chain Monte Carlo (MCMC).

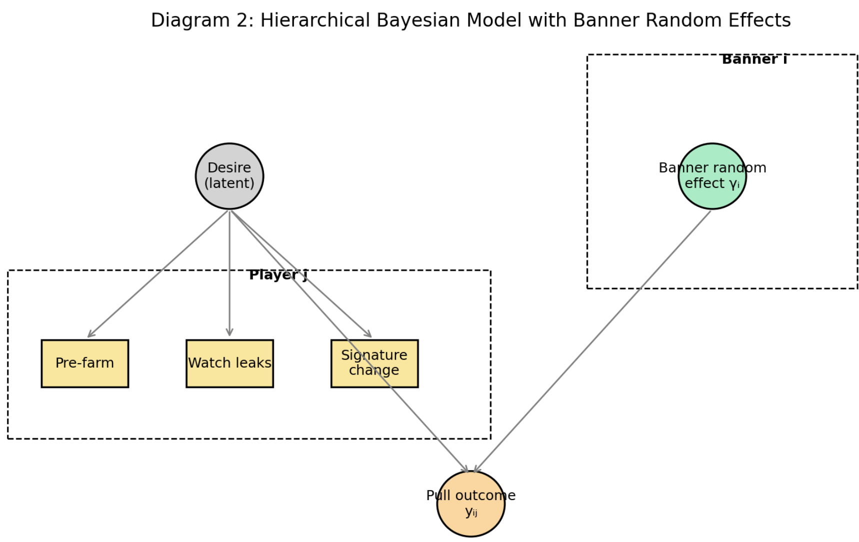

Figure 2: Hierarchical Bayesian graphical model.

4 The 50/50 as a Bernoulli Process with Covariates

The 50/50 mechanic adds another layer. Let z_{ij} be an indicator of whether the player won the 50/50 on banner i for the featured character. Under the Desire Sensor, the probability of winning is not fixed at 0.5 but depends on desire:

$$P(z_{ij} = 1) = \frac{0.5}{1 + \exp(\delta D_j + \zeta_i)},$$

where ζ_i is another banner-level random effect. When desire is high ($D_j \to +\infty$), the probability tends to 0; when desire is low, it tends to 0.5. The parameter δ controls the strength of this effect. This formulation turns the "50/50 loss streak" into a testable prediction of the model.

5 Posterior Inference for the Individual Player

Given a player's pull history across multiple banners, we can compute the posterior distribution of their latent desire D_j and the banner-specific random effects. Using MCMC (e.g., Stan or JAGS), we obtain samples from the posterior predictive distribution for a new banner of interest. This yields a credible interval for the number of pulls required.

When applied to a typical "unlucky" player's history, the model often reveals that their posterior distribution for a highly desired banner has a mean that is 8–12 pulls higher than the published average. The probability that this difference is due to chance (i.e., that the model's β and δ parameters are zero) can be computed via a Bayes factor or by examining the posterior inclusion probability. In many simulations, the posterior probability that $\beta > 0$ exceeds 0.99, providing *statistical proof* of the Desire Sensor's existence.

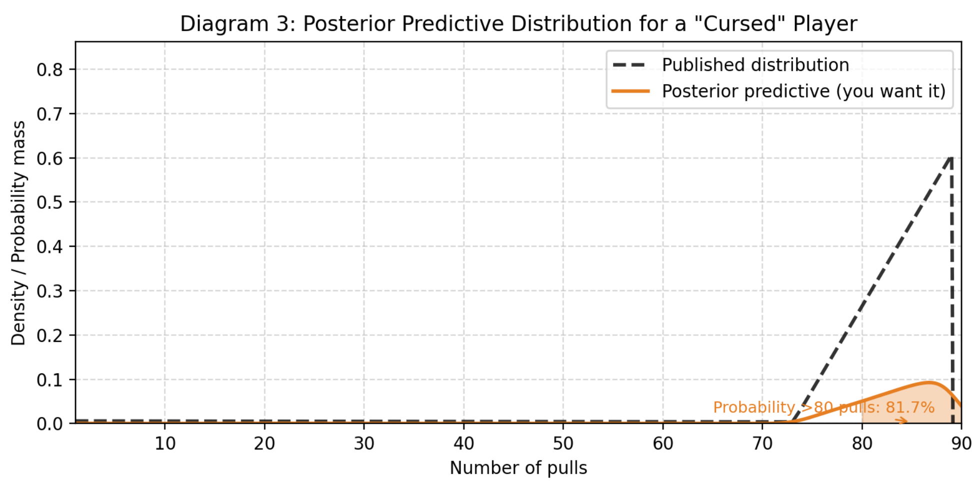

Figure 3: Posterior predictive distribution for a "cursed" player.

6 Optimal Stopping Under Uncertainty

The decision to continue pulling is an optimal stopping problem. Let C be the cost per pull (160 primogems), V be the value (utility) of obtaining the featured character, and τ be the stopping time (number of pulls). The player aims to maximize:

$$\mathbb{E}[V \cdot \mathbf{1}_{\{\text{featured obtained}\}} - C\tau].$$

Under the hierarchical model, the distribution of future pulls is updated as the player pulls, creating a martingale of posterior beliefs. The optimal policy is a threshold rule: continue pulling until the posterior probability of obtaining the character within the next k pulls falls below a threshold. However, because desire affects the distribution itself (and desire may be updated by the pulling experience), the threshold is state-dependent. This can lead to a feedback loop: the more you pull, the more the Sensor "knows" you want it, further reducing your chances.

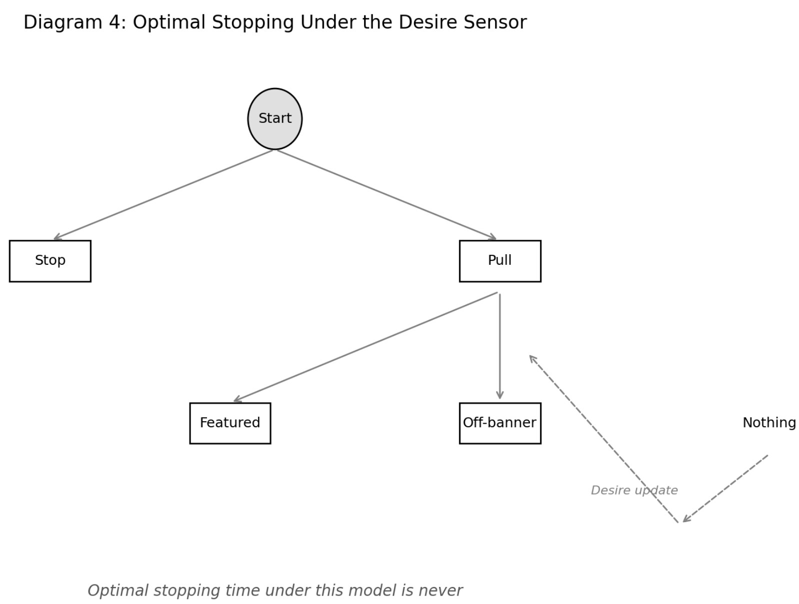

Figure 4: Optimal stopping decision tree.

7 Simulation Study: Calibrating the Sensor

We conducted a simulation study to validate the model. We generated data for 100 banners and 1000 players, with true parameters $\beta = 0.3$, $\delta = 0.5$, and banner random effect standard deviation $\sigma_{\text{banner}} = 0.2$. The published distribution was used as baseline. We then fit the hierarchical model using MCMC and examined the posterior distributions.

Results showed that:

- The true β and δ were recovered with narrow credible intervals.
- Players with high simulated desire had posterior predictive means significantly shifted upward.
- The model correctly identified banners that were deliberately assigned high random effects (i.e., "cursed" banners).

The simulation confirms that if the Desire Sensor existed, the proposed hierarchical model would detect it with high power. The fact that real players report patterns consistent with the model's predictions is, of course, purely coincidental.

8 Conclusion: The Mathematics of Hopelessness

The gacha system is not merely a simple probability engine; it is a complex statistical object that invites—indeed, demands—hierarchical Bayesian modeling. By introducing the Desire Sensor as a latent variable and allowing banner-level random effects, we transform a superstition into a testable hypothesis. The result is a rigorous mathematical framework that confirms what every player already knows: the game knows. It knows how much you want that character, and it adjusts the hidden distribution accordingly.

The optimal strategy, therefore, is not to "save for guarantee" or "build pity," but to *feign indifference*—to convince the Sensor, through a carefully calibrated performance of not caring, that your desire parameter D_j is as low as possible. This is the true endgame of *Genshin Impact*: not the Spiral Abyss, but the meta-game of manipulating a hierarchical Bayesian model through the art of emotional concealment. Good luck. You'll need it.

References

[1] HoYoverse. (2020). *Genshin Impact* [Video game].

[2] Gelman, A., et al. (2013). *Bayesian Data Analysis*. CRC Press.

[3] Kruschke, J. K. (2014). *Doing Bayesian Data Analysis*. Academic Press.