Bitcoin Prediction Markets

James Pierog

Glimpse Ltd.

james@glimpse.markets

December 30, 2025

1 Introduction

A prediction market is an *information market*. It turns disagreement into a measurable signal by letting participants express conviction through trade. Instead of trading an asset directly, participants trade Event Contracts that pay off based on the outcome of a specific event. The price of an Event Contract can be read as a market-implied probability, and that probability updates continuously as traders incorporate news, research, and macro conditions. In other words, prediction markets create a public forecast stream that is native to uncertainty.

This paper develops and analyzes a Bitcoin-native automated market maker that provides liquidity for Event Contracts, enabling a scalable forecast of market-implied probabilities that can be used for hedging and decision-making. We focus specifically on financial forecasting and implement a liquidity-sensitive Logarithmic Market Scoring Rule (LS-LMSR) denominated natively in Bitcoin. The following sections derive the Bitcoin-denominated cost function, prove its bounded-loss property, analyze the profit-and-loss landscape for liquidity provision under different parameter calibrations, and establish a rigorous framework for intraday risk management.

The mechanism is designed to support a broad family of Event Contracts whose primary purpose is forecasting and risk transfer on objectively verifiable financial or economic questions. After establishing the execution and risk framework, we describe applications that motivate the market design. These include contracts that hedge relative underperformance of Bitcoin versus conventional benchmarks, contracts that convert independent probability estimates into disciplined expected-value strategies with fractional Kelly sizing, and contracts that translate high-accuracy probabilistic signals into conventional linear trading positions. We also develop Bitcoin-mining applications, including difficulty-adjustment hedges and block reward variance hedges, where settlement is tied to public protocol data and the economic need is insurance for variance risk.

The platform-level vision is to treat Event Contracts as a standardized interface for expressing and aggregating forward-looking beliefs, with Bitcoin-native settlement. The prices of Event Contracts provide a continuously updating forecast stream: a dense set of market-implied probabilities across assets, horizons, and protocol variables that can be evaluated for calibration and reused for risk management and decision support, functioning as a continuously updated "weather map" of financial and economic uncertainty.

2 Background Information

A *prediction market* is a mechanism for aggregating dispersed information about an uncertain event whose outcome will be objectively verifiable at a future time. The market offers state-contingent claims (event contracts) and uses trading activity to produce an interpretable public forecast, typically represented by prices that approximate probabilities. This mechanism rests on a mathematical foundation of scoring rules, which were originally developed to evaluate and reward accurate probabilistic forecasts in domains like meteorology. Proper scoring rules create incentives for forecasters to report their true beliefs—a principle that extends to market scoring rules, which adapt the concept to sequential trading among many participants. In this setting, trade is a means to an informational end: the operator may rationally subsidize participation and liquidity in order to obtain better forecasts, rather than to maximize revenue or balance the budget. To understand how these incentives are formally constructed, we begin with the theory of proper scoring rules and their extension to market scoring rules.

2.1 From Probabilistic Forecasting to Market Scoring Rules

The conceptual origin of prediction markets lies in *probabilistic forecasting*. Long before electronic markets, forecasters in domains such as meteorology were asked to report probabilities of events (e.g., precipitation), and these probability forecasts were evaluated using *scoring rules*. A scoring rule is a contract that assigns a numerical score (or monetary payoff) to a reported distribution $r = (r_1, \ldots, r_m)$ once the realized outcome is known. Early work by Brier introduced a quadratic scoring rule for verifying probability forecasts in weather forecasting, and subsequent work formalized the logarithmic scoring rule, with both families becoming standard tools for evaluating probabilistic predictions. Scoring rules have since been widely used in weather forecasting and related forecast-verification settings [8, 16].

Formally, let $\Omega = \{1, \ldots, m\}$ denote mutually exclusive outcomes of an event. A scoring rule is a collection of functions $\{s_i(\cdot)\}_{i=1}^m$ where $s_i(r)$ is the score paid if outcome i occurs. The rule is *(strictly) proper* if a risk-neutral forecaster with true belief p maximizes expected score by reporting $r = p$ [16].

Two canonical examples illustrate this framework. The quadratic rule (Brier score) is given by

$$s_i(r) = a_i - b(1 - r_i)^2, \qquad b > 0,$$

where the forecaster receives maximum score when $r_i = 1$ for the realized outcome i and is penalized quadratically for deviations from perfect accuracy. The logarithmic rule takes the form

$$s_i(r) = a_i + b \log r_i, \qquad b > 0,$$

which rewards accurate probability estimates through the natural logarithm and becomes arbitrarily punitive as reported probabilities approach zero for the realized outcome. Properness matters because it creates a disciplined incentive: under standard assumptions, the forecaster is rewarded for stating the probability distribution they actually believe. A concrete illustration of the incentive logic is deferred to Appendix B, which derives truth-telling under the quadratic (Brier) and logarithmic scores and contrasts it with a simple non-proper rule.

A limitation of a one-shot scoring rule is that it elicits a forecast from *one* forecaster at *one* time. In many applications, information arrives gradually and is distributed across many participants. What is needed is a mechanism that allows many forecasters to contribute over time, while preserving the incentive logic of proper scoring rules. This is where market scoring rules come into play.

2.2 Hanson's Insight: The Logarithmic Market Scoring Rule

Hanson introduced *market scoring rules* (MSR) precisely to pool opinions from multiple forecasters in a sequential, shared way [16]. An MSR maintains a *current* market distribution $p \in \Delta_m$. At any time, a trader may replace it by a new distribution p'. If outcome i occurs, the trader receives the incremental score improvement

$$\Pi_i = s_i(p') - s_i(p).$$

This is an intuitive "pay for improvement" rule: a participant profits if they move the market forecast in a direction that makes the realized outcome more consistent with their information [12].

A key operational feature is that the transfers *telescope*. Each trader effectively takes over responsibility for the current score and replaces it with a new score. As a result, the market maker is responsible only for the difference between the *initial* distribution p_0 and the *final* distribution reached by the last update. This yields a clean bounded-loss guarantee: the market maker's worst-case loss is finite and can be expressed directly in terms of the scoring rule [12].

While MSRs are conceptually clear, they do not initially resemble familiar "markets" because participants appear to be editing probability vectors rather than trading contracts. However, MSRs are *equivalent* to a more intuitive implementation: a *cost-function market maker* that offers state-contingent contracts [11, 12].

2.3 From the LMSR to Cost-Function Market Makers

To predict the outcome of some future event, a cost-function-based market maker offers some initial quantity of Arrow-Debreu contracts, one for each possible (mutually exclusive) outcome. An Arrow-Debreu contract pays 100 sats if the corresponding outcome is realized and 0 sats otherwise, and the contracts are priced between 0 and 100 sats before resolution.

Let q_i be the total quantity of contract i held by all traders combined, and let $\mathbf{q} = (q_1 \dots q_n)$ be the vector of all quantities held. The market maker utilizes a cost function $C(\mathbf{q})$ that records the total amount of money traders have spent as a function of the total number of contracts held on each outcome.

A trader who wants to buy any bundle of contracts such that the total number of outstanding contracts changes from q_{old} to q_{new} must pay $C(\mathbf{q}_{\text{new}}) - C(\mathbf{q}_{\text{old}})$ sats to the market maker. Negative quantities encode sell orders, and negative "payments" encode sale proceeds earned by the trader.

The conventional LMSR cost function is written as:

$$C(\mathbf{q}) = b(\mathbf{q}) \log \left(\sum_i \exp(q_i / b(\mathbf{q})) \right)$$

where $b(\mathbf{q}) = b$ is an exogenously set constant. The instantaneous price of state i is given by the partial derivative of the cost function along i. The expression for the price is therefore given by:

$$p_i(\mathbf{q}) = \frac{\partial C(\mathbf{q})}{\partial q_i} = \frac{\exp(q_i/b)}{\sum_j \exp(q_j/b)}.$$

While elegant, the conventional LMSR has a significant practical limitation: the liquidity parameter b is fixed. This means market depth remains constant regardless of trading volume or participation. A

market with $1,000 in total trading volume exhibits the same price sensitivity as one with $1,000,000 in volume.

This fixed-depth property creates two problems for large-scale information aggregation. First, as more traders participate and the quantity of capital committed increases, individual trades continue to have the same marginal price impact, potentially creating excessive volatility in high-volume markets. Second, markets cannot naturally develop deeper liquidity as they mature and attract more capital, limiting their ability to incorporate information from large, informed traders without dramatic price movements.

What is needed is a mechanism that automatically adjusts market depth based on trading activity, making price sensitivity responsive to the level of market participation. This motivates the development of liquidity-sensitive variants of the LMSR.

2.4 A Liquidity Sensitive LMSR

If we let $b(\mathbf{q}) = \alpha \cdot \sum_j q_j$ then the LMSR cost function and price function become liquidity sensitive. Sensitivity to liquidity is desirable because it squares intuitively with the way we would want markets to function: small investments move prices less in thick (liquid) markets than in thin (illiquid) markets. We can take the partial derivative of the cost function to derive the price function as follows:

$$\begin{aligned}
p_i(\mathbf{q}) &= \frac{\partial C(\mathbf{q})}{\partial q_i} \\
&= \frac{\partial}{\partial q_i} \cdot b(\mathbf{q}) \cdot \log\left(\sum_i \exp(q_i/b(\mathbf{q}))\right) \\
&= \frac{\partial}{\partial q_i} \cdot \alpha \cdot \sum_j q_j \cdot \log\left(\sum_i \exp(q_i/b(\mathbf{q}))\right) \\
&= \alpha \cdot \log\left(\sum_j \exp(q_j/b(\mathbf{q}))\right) + \frac{\sum_j q_j \exp(q_i/b(\mathbf{q})) - \sum_j q_j \exp(q_j/b(\mathbf{q}))}{\left(\sum_j q_j\right) \cdot \left(\sum_j \exp(q_j/b(\mathbf{q}))\right)}
\end{aligned}$$

Therefore, the liquidity-sensitive LMSR cost and price functions are:

$$C(\mathbf{q}) = b(\mathbf{q}) \log\left(\sum_i \exp(q_i/b(\mathbf{q}))\right)$$

$$p_i(\mathbf{q}) = \alpha \cdot \log\left(\sum_j \exp(q_j/b(\mathbf{q}))\right) + \frac{\sum_j q_j \exp(q_i/b(\mathbf{q})) - \sum_j q_j \exp(q_j/b(\mathbf{q}))}{\left(\sum_j q_j\right) \cdot \left(\sum_j \exp(q_j/b(\mathbf{q}))\right)}$$

where $b(\mathbf{q}) = \alpha \cdot \sum_j q_j$. The mechanism is designed to price contracts between 0 and 1 dollars, so we must slightly alter the it to be able to price contracts and trades in a Bitcoin unit of account.

3 A Bitcoin-Denominated Liquidity Sensitive LMSR

For a binary market with outcomes Y and N (Yes/No), we define the market states as:

$$\mathbf{Q} = [\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n]$$

where each state $\mathbf{q}_i = (y_i, n_i)$ represents the quantities of Yes and No contracts held by all traders at state i.

If we let $b(\mathbf{q}_i) = \alpha \cdot (y_i + n_i)$ then the LMSR cost function and price function become liquidity sensitive. Since we are denominating our trades in Bitcoin unit of account (where 1 BTC = 100,000,000 sats), we modify the cost function to price contracts as follows:

$$C(y_i, n_i) = 100 \cdot b(\mathbf{q}_i) \cdot \log\left(\exp\left(\frac{y_i}{b(\mathbf{q}_i)}\right) + \exp\left(\frac{n_i}{b(\mathbf{q}_i)}\right)\right)$$

Since we are utilizing a path-independent market maker, the instantaneous price of outcome Y is given by the partial derivative of the cost function along y_i. We can take the partial derivative of the cost function to derive the price function as follows:

$$\begin{aligned} p_Y(\mathbf{q}_i) &= \frac{\partial C}{\partial y_i} \\ &= 100 \cdot \alpha \cdot \log\left(\exp\left(\frac{y_i}{b(\mathbf{q}_i)}\right) + \exp\left(\frac{n_i}{b(\mathbf{q}_i)}\right)\right) + 100 \cdot \frac{\exp\left(\frac{y_i}{b(\mathbf{q}_i)}\right)}{(y_i + n_i) \cdot \left(\exp\left(\frac{y_i}{b(\mathbf{q}_i)}\right) + \exp\left(\frac{n_i}{b(\mathbf{q}_i)}\right)\right)} \end{aligned}$$

By symmetry, the price of outcome N is given by the partial derivative of the cost function along n_i. The expression for the price is therefore given by:

$$\begin{aligned} p_N(\mathbf{q}_i) &= \frac{\partial C}{\partial n_i} \\ &= 100 \cdot \alpha \cdot \log\left(\exp\left(\frac{y_i}{b(\mathbf{q}_i)}\right) + \exp\left(\frac{n_i}{b(\mathbf{q}_i)}\right)\right) + 100 \cdot \frac{\exp\left(\frac{n_i}{b(\mathbf{q}_i)}\right)}{(y_i + n_i) \cdot \left(\exp\left(\frac{y_i}{b(\mathbf{q}_i)}\right) + \exp\left(\frac{n_i}{b(\mathbf{q}_i)}\right)\right)} \end{aligned}$$

Therefore, our revised Bitcoin-denominated liquidity-sensitive LMSR cost and price functions are:

$$C(\mathbf{q}_i) = 100 \cdot b(\mathbf{q}_i) \cdot \log\left(\exp\left(\frac{y_i}{b(\mathbf{q}_i)}\right) + \exp\left(\frac{n_i}{b(\mathbf{q}_i)}\right)\right)$$

$$p_Y(\mathbf{q}_i) = 100 \cdot \alpha \cdot \log\left(\exp\left(\frac{y_i}{b(\mathbf{q}_i)}\right) + \exp\left(\frac{n_i}{b(\mathbf{q}_i)}\right)\right) + 100 \cdot \frac{\exp\left(\frac{y_i}{b(\mathbf{q}_i)}\right)}{(y_i + n_i) \cdot \left(\exp\left(\frac{y_i}{b(\mathbf{q}_i)}\right) + \exp\left(\frac{n_i}{b(\mathbf{q}_i)}\right)\right)}$$

$$p_N(\mathbf{q}_i) = 100 \cdot \alpha \cdot \log\left(\exp\left(\frac{y_i}{b(\mathbf{q}_i)}\right) + \exp\left(\frac{n_i}{b(\mathbf{q}_i)}\right)\right) + 100 \cdot \frac{\exp\left(\frac{n_i}{b(\mathbf{q}_i)}\right)}{(y_i + n_i) \cdot \left(\exp\left(\frac{y_i}{b(\mathbf{q}_i)}\right) + \exp\left(\frac{n_i}{b(\mathbf{q}_i)}\right)\right)}$$

where $b(\mathbf{q}_i) = \alpha \cdot (y_i + n_i)$.

The pricing rule has bounded loss. Specifically, the market maker has worst-case loss equal to the amount of the initial subsidy. For a two-outcome market, the market maker loses at most $C(\mathbf{q}_0)$ sats regardless of which outcome occurs or the final trading state.

3.1 Bounded Loss Analysis

For this revised price and cost function, we can derive the profit and loss of the market maker. The cost of trader t_i who is the i-th trade with the market maker is $C(\mathbf{q_i}) - C(\mathbf{q_{i-1}})$. The sum of the costs of the trades are:

$$\begin{aligned}\sum_{i=1}^{n} C(\mathbf{q_i}) - C(\mathbf{q_{i-1}}) &= [C(\mathbf{q_1}) - C(\mathbf{q_0})] + [C(\mathbf{q_2}) - C(\mathbf{q_1})] + [C(\mathbf{q_3}) - C(\mathbf{q_2})] + \cdots + [C(\mathbf{q_n}) - C(\mathbf{q_{n-1}})] \\ &= C(\mathbf{q_1}) - C(\mathbf{q_0}) + C(\mathbf{q_2}) - C(\mathbf{q_1}) + C(\mathbf{q_3}) - C(\mathbf{q_2}) + \cdots + C(\mathbf{q_n}) - C(\mathbf{q_{n-1}}) \\ &= C(\mathbf{q_n}) - C(\mathbf{q_0})\end{aligned}$$

where all intermediate terms $C(\mathbf{q_1}), C(\mathbf{q_2}), \ldots, C(\mathbf{q_{n-1}})$ cancel out in the telescoping sum.

The automated market maker must pay $100 \cdot q_i$ sats if outcome i occurs. Therefore, the net P&L of the market maker if outcome i occurs, for an initial state q_0 and final state q_n is:

$$\mathrm{PnL}_i(\mathbf{q_0}, \mathbf{q_n}) = C(\mathbf{q_n}) - C(\mathbf{q_0}) - 100 \cdot \mathbf{q_i}$$

The pricing rule has bounded loss. Specifically, the market maker has worst-case loss equal to the amount of the initial subsidy: it loses at most $C(\mathbf{q_0})$ sats.

3.2 Transaction Fees and Path Dependence

Without fees, a cost-function market maker is path-independent. For any sequence of trades that moves the state from $\mathbf{q}_0$ to $\mathbf{q}_k$, the total cash collected from traders is

$$\sum_{i=1}^{k} \big(C(\mathbf{q}_i) - C(\mathbf{q}_{i-1})\big) = C(\mathbf{q}_k) - C(\mathbf{q}_0),$$

so the market maker's outcome-contingent P&L depends only on the endpoints.

Now impose a proportional transaction fee with rate $\tau = 0.02$ charged on each trade's absolute cash flow. For the ith trade, the trader's payment to the market maker is

$$\Delta C_i := C(\mathbf{q}_i) - C(\mathbf{q}_{i-1}),$$

which may be positive (net buy) or negative (net sell). The fee collected on that trade is

$$\phi_i := \tau \, |\Delta C_i|.$$

Therefore total fee revenue over a discrete trading path $\mathcal{P} = \{\mathbf{q}_0 \to \mathbf{q}_1 \to \cdots \to \mathbf{q}_k\}$ is

$$\Phi(\mathcal{P}) = \tau \sum_{i=1}^{k} |C(\mathbf{q}_i) - C(\mathbf{q}_{i-1})|.$$

This is the source of path dependence. The endpoint term $C(\mathbf{q}_k) - C(\mathbf{q}_0)$ is fixed, but $\sum_i |\Delta C_i|$ increases with back-and-forth trading.

A continuous formulation makes this explicit. Let $\gamma : [0, T] \to \mathbb{R}^2$ be a trading path with $\gamma(t) = (y(t), n(t))$ and let $d\mathbf{q} = (dy, dn)$. Since $p = \nabla C$, we have

$$dC = \nabla C(\mathbf{q}) \cdot d\mathbf{q}.$$

Define the fee functional as total variation of C along γ:

$$\Phi(\gamma) = \tau \int_{\gamma} |dC| = \tau \int_{\gamma} \left|\nabla C(\mathbf{q}) \cdot d\mathbf{q}\right|.$$

For a piecewise-linear path that interpolates the discrete states, this reduces to $\Phi(\mathcal{P})$ above.

With fees, the market maker's P&L becomes

$$\mathrm{PnL}_{\omega}(\mathcal{P}) = \big(C(\mathbf{q}_k) - C(\mathbf{q}_0)\big) - 100\, q_{\omega,k} + \Phi(\mathcal{P}), \qquad \omega \in \{Y, N\},$$

where $\mathbf{q}_k = (y_k, n_k)$ and $q_{Y,k} = y_k$, $q_{N,k} = n_k$ (or the corresponding net customer quantities if a virtual seed state is used). Different paths with the same endpoints generally satisfy $\Phi(\mathcal{P}_1) \neq \Phi(\mathcal{P}_2)$, so higher two-sided trading volume increases fee revenue regardless of the eventual outcome.

3.3 Market Maker Incentives and Optimal Market Design

The market maker is therefore incentivized to list markets that elicit high levels of disagreement from a diverse set of traders, or list markets with inherently high volatility to induce winding trading paths. The most profitable markets to provide liquidity for are markets that do not immediately reveal much information by way of a market-consensus forecast.

Markets that are the most informative tend to have the highest consensus amongst traders, which implies traders have purchased a sufficient quantity of contracts such that the price of yes (or no) is near 100, and no other traders are willing to take a significant contrary position to the market. Providing liquidity to prediction markets is optimal for topics that have high uncertainty, where people have high conviction. Markets with low uncertainty imply traders have collectively found consensus, and thus there is little opportunity to take profitable risk.

4 Optimal LS-LMSR Parameter Calibration

The liquidity-sensitive LMSR sets the liquidity parameter as a function of market activity,

$$b(\mathbf{q}) = \alpha \sum_{i=1}^{n} q_i,$$

so market depth increases automatically as more contracts are outstanding [24].

The effect of α calibration on market maker profitability and price dynamics can be visualized through profit landscapes across different terminal market states. Figures 1 and 2 demonstrate these relationships by plotting market maker profit and loss (excluding transaction fees) as a function of final contract quantities.

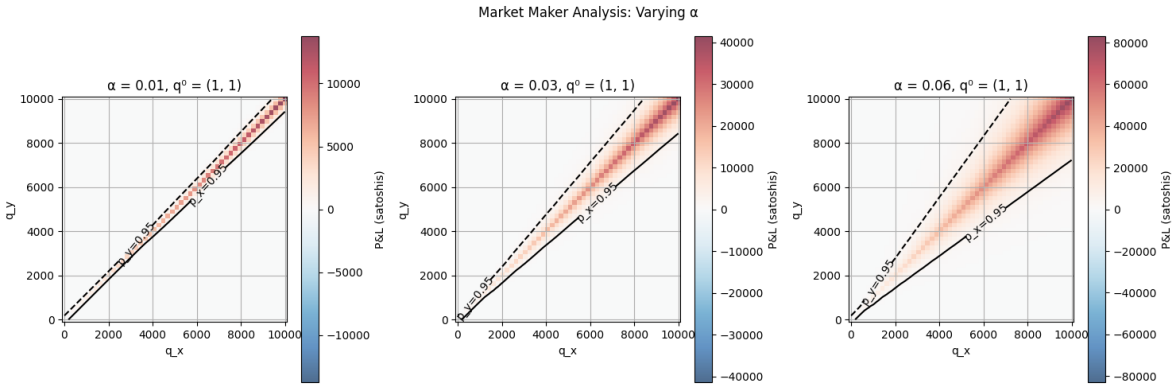

Figure 1: The shaded regions show where the market maker's worst-case revenue is greater than zero in a two-outcome market with initial quantity vector (1, 1) and various values of alpha. The top black ray represents py = 95 sats and the bottom black ray represents px = 95 sats.

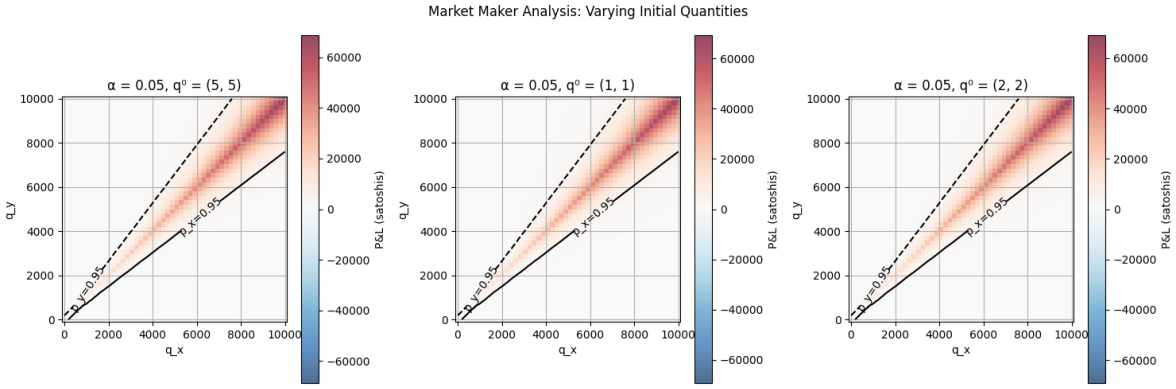

Figure 2: The shaded regions show where the market maker's worst-case revenue is greater than zero in a two-outcome market with alpha = .03 and various initial quantity vectors. The top black ray represents py = 95 sats and the bottom black ray represents px = 95 sats.

In these visualizations, the horizontal and vertical axes represent the final quantities of contracts q_X and

q_Y respectively. The color intensity indicates market maker profitability: dark red regions represent positive profit from liquidity provision, while blue and white regions indicate losses. The diagonal black lines represent constant price levels ($p_Y = 95$ sats and $p_X = 95$ sats), illustrating how price boundaries shift with parameter changes.

Figure 1 demonstrates the critical role of α in determining market depth and price sensitivity. Lower values of α (such as $\alpha = 0.01$) create highly volatile price dynamics where small trades produce large price movements, resulting in narrow profitable regions for the market maker. As α increases to 0.03 and 0.06, the profitable regions expand significantly, indicating that higher α values provide increased market depth and stability. This occurs because larger α reduces the marginal price impact of individual trades, making the market more resilient to order flow volatility.

Figure 2 illustrates how different initial quantity vectors $\mathbf{q}^0$ affect the profit landscape while holding $\alpha = 0.05$ constant. The symmetry and positioning of profitable regions shift based on the initial market state, but the overall structure remains consistent. Markets initialized with balanced quantities $\mathbf{q}^0 = (1, 1)$ or $\mathbf{q}^0 = (2, 2)$ exhibit symmetric profit patterns, while asymmetric initializations create correspondingly skewed profit landscapes.

4.1 Default Parameters for Glimpse

Glimpse currently sets $\alpha = 0.111$ and $q_0 = [5000, 5000]$ as the platform defaults. These values balances initial market depth with meaningful price responsiveness under typical trade sizes observed in simulation.

Figure 3 presents the profit and loss landscape for Glimpse's operational LS-LMSR implementation using the platform's standard parameter calibration: $\alpha = 0.111$ and initial quantity vector $\mathbf{q}_0 = (5000, 5000)$. This configuration yields an initial subsidy of $C(\mathbf{q}_0) = 576{,}246$ sats, representing the maximum potential loss exposure from liquidity provision.

The visualization demonstrates several critical operational characteristics. The green line delineates the breakeven boundary where market maker profit and loss from liquidity provision equals zero, formally defined by the condition $\mathrm{PnL}_i(\mathbf{q}_0, \mathbf{q}_n) = 0$. The dotted lines represent constant price contours at $p_Y = 95$ sats and $p_N = 95$ sats, corresponding to 95-sat contract prices that serve as natural boundaries for high-conviction market states.

The color gradient reveals the fundamental profitability structure of Glimpse's market making mechanism. Red regions indicate positive profit from liquidity provision alone, occurring when the terminal market state exhibits prices below the 95-sat threshold. This corresponds mathematically to market configurations where one outcome maintains relatively low conviction, allowing the automated market maker to benefit from the spread between collected premiums and required payouts.

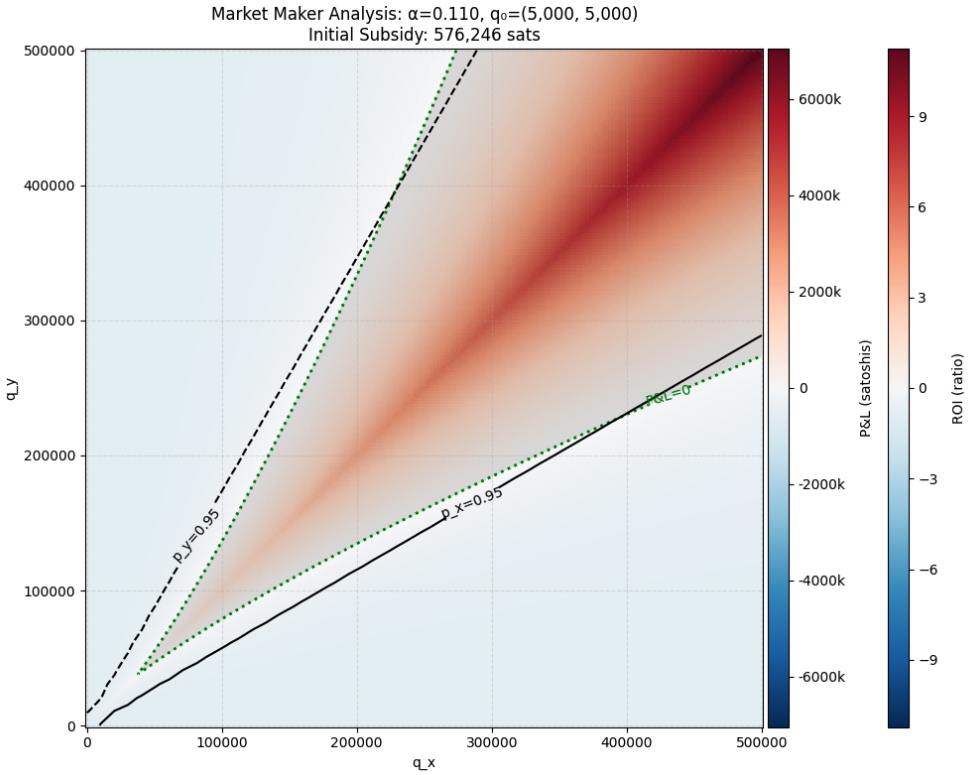

Figure 3: Market maker profit and loss landscape for Glimpse's standard LS-LMSR configuration with $\alpha = 0.111$ and $\mathbf{q}_0 = (5000, 5000)$. The green line represents the breakeven boundary, dotted lines indicate 95-sat price contours, and color intensity shows profitability from liquidity provision.

Conversely, blue regions represent scenarios where liquidity provision becomes unprofitable, occurring when terminal prices exceed 95 sats for either outcome. In these high-conviction states, the market maker faces adverse selection as informed traders have moved prices to reflect strong consensus. However, the bounded loss property ensures that deficits never exceed the initial subsidy $C(\mathbf{q}_0)$, providing a finite risk exposure regardless of trading outcomes.

The practical implications for market dynamics are significant. Any trading sequence can be conceptualized as a stochastic path $\{\mathbf{q}_0 \to \mathbf{q}_1 \to \cdots \to \mathbf{q}_n\}$ through the state space, representing the cumulative effect of trader decisions over the market's lifecycle. The profitability of liquidity provision depends critically on where this random walk terminates relative to the price boundaries, with paths ending in moderate-conviction regions (prices below 95 sats) yielding positive returns and paths converging to high-conviction states (prices above 95 sats) resulting in controlled losses bounded by the initial subsidy.

4.2 Normalization of Prices into Probability Distributions

In Glimpse, the economic terms of trade are defined by the LS–LMSR *cost function*. For a market state $\mathbf{q}$, a trade that moves the state from $\mathbf{q}_{\text{old}}$ to $\mathbf{q}_{\text{new}}$ is charged (or paid, in the case of a sale) by

the cost difference

$$\Delta C := C(\mathbf{q}_{\text{new}}) - C(\mathbf{q}_{\text{old}}).$$

Accordingly, contracts are not purchased by multiplying a quoted per-contract price by a quantity. The executed cash flow is determined directly by the path-independent cost function evaluated at the pre- and post-trade states. The instantaneous LS–LMSR price vector, obtained as the gradient of the cost function, serves as a local marginal-price description of the curve at the current state, but it is not itself the pricing rule used to compute the total cost of a discrete trade.

Because the liquidity-sensitive construction does not, in general, yield raw marginal-price components that sum exactly to the 100-satoshi payout normalization, Glimpse separates (i) the *trade economics* from (ii) the *forecast signal* presented to users. Let $\tilde{p}_i(\mathbf{q}) := \partial C(\mathbf{q})/\partial q_i$ denote the raw LS–LMSR marginal prices. Glimpse converts these raw values into a probability distribution by normalization,

$$P_i(\mathbf{q}) := \frac{\tilde{p}_i(\mathbf{q})}{\sum_{k=1}^{m} \tilde{p}_k(\mathbf{q})}, \qquad \sum_{i=1}^{m} P_i(\mathbf{q}) = 1,$$

and reports the corresponding contract "odds" on a 0–100 scale via $100\,P_i(\mathbf{q})$. This normalization is a representational layer designed to ensure that the platform outputs a coherent, interpretable probability distribution that can be used as a forecasting signal. It does not modify the underlying cost function, and therefore it does not alter the cash flow of any trade, which remains exactly $C(\mathbf{q}_{\text{new}}) - C(\mathbf{q}_{\text{old}})$.

Under this architecture, it is not mathematically accurate to treat the liquidity parameter α as a client commission. The parameter α enters through the liquidity function $b(\mathbf{q}) = \alpha \sum_i q_i$ and primarily governs market depth and the responsiveness of the market state to incremental order flow. Lower α increases early-stage price elasticity, meaning that small trades can move the state rapidly toward near-certainty, concentrating favorable execution among early participants and reducing the opportunity for later traders to obtain comparable odds. Higher α increases depth, requiring more cumulative liquidity to move the market toward certainty, which distributes execution quality across a broader set of traders and tends to improve the stability of the forecast signal as participation grows. These effects operate through the curvature of C and the sensitivity of ∇C to changes in $\mathbf{q}$, rather than through a fixed per-trade charge.

The relevant economic analogue of a "spread" in a cost-function market maker is therefore dynamic and state-dependent. For a discrete trade that increases outcome-i quantity by $\Delta q_i > 0$ (holding other components fixed for exposition), the trader's average execution price is

$$\bar{p}_i := \frac{C(\mathbf{q}_{\text{old}} + \Delta q_i\, \mathbf{e}_i) - C(\mathbf{q}_{\text{old}})}{\Delta q_i},$$

which generally differs from the instantaneous marginal price $\tilde{p}_i(\mathbf{q}_{\text{old}})$ because the trade traverses a non-linear cost surface. Market depth calibration via α affects this execution-price impact, but it does not constitute a separately assessed fee and it does not map one-to-one into platform revenue, which depends on realized order flow and resolution outcomes.

5 Intraday Maximum Potential Payout and Open Interest

This section provides an explicit computation, at intraday checkpoints, of (i) open interest and (ii) the operator's maximum potential settlement outflow across all outstanding Event Contracts while markets remain active.

5.1 Definitions at an intraday checkpoint

Consider a binary Event Contract with outcomes Y and N. At a checkpoint time t, let the outstanding quantities be

$$\mathbf{q}(t) = (y(t), n(t)), \qquad b(\mathbf{q}(t)) = \alpha\big(y(t) + n(t)\big),$$

and let the Bitcoin-denominated LS-LMSR cost function be

$$C\big(y(t), n(t)\big) = 100 \cdot b(\mathbf{q}(t)) \, \log\left(\exp\left(\frac{y(t)}{b(\mathbf{q}(t))}\right) + \exp\left(\frac{n(t)}{b(\mathbf{q}(t))}\right)\right).$$

Open interest. We define contract open interest at time t as

$$\mathrm{OI}(t) := y(t) + n(t),$$

and its satoshi-notional as $100 \cdot \mathrm{OI}(t)$ sats.

Gross payout by realized outcome. If the market resolves based on the state at time t, the gross payout owed to winning contract holders is

$$G_Y(t) := 100\, y(t), \qquad G_N(t) := 100\, n(t).$$

Operator cash-in and maximum potential settlement outflow. Let $\mathbf{q}_0$ be the market state at listing time. The cumulative cash paid into the market maker by time t is

$$R(t) := C\big(\mathbf{q}(t)\big) - C\big(\mathbf{q}_0\big).$$

This formula ignores transaction fees, which makes the resulting bound conservative because fees increase available reserves.

Conditional on an immediate resolution at time t, the operator's net settlement outflow is

$$S_Y(t) := G_Y(t) - R(t), \qquad S_N(t) := G_N(t) - R(t).$$

We define the *maximum potential payout* (worst-case net settlement outflow) at checkpoint t as

$$M(t) := \max\{S_Y(t), S_N(t)\}.$$

For a platform with multiple simultaneously listed markets indexed by k, the corresponding aggregate quantities are computed by summation, e.g.

$$M_{\text{total}}(t) := \sum_k M_k(t), \qquad \mathrm{OI}_{\text{total}}(t) := \sum_k \mathrm{OI}_k(t).$$

5.2 Worked example with intraday checkpoints

Let the initial state be

$$\mathbf{q}_0 = (5000, 5000), \qquad \alpha = 0.111, \qquad b(\mathbf{q}) = \alpha(y + n).$$

Trades occur in sequence. Alice buys 100 contracts on Y. Bob buys 300 contracts on N. Carol buys 500 contracts on Y. Thus the checkpoint states are:

$$\mathbf{q}_0 = (5000, 5000), \quad \mathbf{q}_1 = (5100, 5000), \quad \mathbf{q}_2 = (5100, 5300), \quad \mathbf{q}_3 = (5600, 5300).$$

The corresponding $b(\mathbf{q}_i) = \alpha(y_i + n_i)$ values are

$$b(\mathbf{q}_0) = 1110.0, \quad b(\mathbf{q}_1) = 1121.1, \quad b(\mathbf{q}_2) = 1154.4, \quad b(\mathbf{q}_3) = 1209.9.$$

Using the above cost function, the cost-function values (in sats) are approximately

$$C(\mathbf{q}_0) \approx 576{,}939, \quad C(\mathbf{q}_1) \approx 582{,}820, \quad C(\mathbf{q}_2) \approx 600{,}449, \quad C(\mathbf{q}_3) \approx 629{,}791.$$

Therefore the amounts paid by each trader are:

$$\text{Alice pays } C(\mathbf{q}_1) - C(\mathbf{q}_0) \approx 5{,}881,$$

$$\text{Bob pays } C(\mathbf{q}_2) - C(\mathbf{q}_1) \approx 17{,}629,$$

$$\text{Carol pays } C(\mathbf{q}_3) - C(\mathbf{q}_2) \approx 29{,}342.$$

The cumulative cash-in at the final checkpoint is

$$R(3) = C(\mathbf{q}_3) - C(\mathbf{q}_0) \approx 52{,}852 \text{ sats}.$$

5.2.1 Trader payoffs and conditional P&L at resolution

In Glimpse's operational model, the initialization vector $\mathbf{q}_0 = (y_0, n_0)$ is a *virtual seed state* used to parameterize the LS-LMSR curve and to ensure continuous two-sided quoting from market open. It does not represent contracts sold to customers. Accordingly, open interest and settlement exposure are computed on *net customer-purchased* quantities:

$$\Delta y_i := y_i - y_0, \qquad \Delta n_i := n_i - n_0.$$

At each checkpoint $\mathbf{q}_i$, we compute customer open interest $\mathrm{OI}(i)$, gross customer payouts if yes or no is correct $G_Y(i), G_N(i)$, and net settlement outflows $S_Y(i), S_N(i)$ as follows:

$$\mathrm{OI}(i) = \Delta y_i + \Delta n_i, \qquad G_Y(i) = 100\Delta y_i, \qquad G_N(i) = 100\Delta n_i,$$

$$S_Y(i) = G_Y(i) - \big(C(\mathbf{q}_i) - C(\mathbf{q}_0)\big), \qquad S_N(i) = G_N(i) - \big(C(\mathbf{q}_i) - C(\mathbf{q}_0)\big).$$

Here $S_\omega(i) > 0$ means the operator must pay out $S_\omega(i)$ sats from the liquidity reserve provided by the subsidy if outcome $\omega \in \{Y, N\}$ resolves immediately at checkpoint i. If $S_\omega(i) < 0$, the operator retains $-S_\omega(i)$ sats net of customer capital committed.

Checkpoint	(y, n)	OI	R	G_Y	G_N	S_Y	S_N
t_0 (open)	$(5000, 5000)$	0	0	0	0	0	0
t_1 (Alice)	$(5100, 5000)$	100	5,881	10,000	0	4,119	−5,881
t_2 (Bob)	$(5100, 5300)$	400	23,510	10,000	30,000	−13,510	6,490
t_3 (Carol)	$(5600, 5300)$	900	52,852	60,000	30,000	7,148	−22,852

Each contract pays 100 sats if its outcome occurs and 0 otherwise.
Alice holds 100 Y-contracts and paid $\approx$ 5,881 sats. Her conditional P&L is:

$$\Pi_Y^{\text{Alice}} \approx 100 \cdot 100 - 5{,}881 = 4{,}119, \qquad \Pi_N^{\text{Alice}} \approx -5{,}881.$$

Bob holds 300 N-contracts and paid $\approx$ 17,629 sats. His conditional P&L is:

$$\Pi_Y^{\text{Bob}} \approx -17{,}629, \qquad \Pi_N^{\text{Bob}} \approx 100 \cdot 300 - 17{,}629 = 12{,}371.$$

Carol holds 500 Y-contracts and paid $\approx$ 29,342 sats. Her conditional P&L is:

$$\Pi_Y^{\text{Carol}} \approx 100 \cdot 500 - 29{,}342 = 20{,}658, \qquad \Pi_N^{\text{Carol}} \approx -29{,}342.$$

From the operator's perspective, the conditional market maker P&L (excluding transaction fees) at checkpoint t_3 is computed on net customer quantities. Since $\Delta y_3 = 5600 - 5000 = 600$ and $\Delta n_3 = 5300 - 5000 = 300$, we obtain:

$$\text{PnL}_Y(3) = R(3) - 100\Delta y_3 \approx 52{,}852 - 60{,}000 = -7{,}148,$$

$$\text{PnL}_N(3) = R(3) - 100\Delta n_3 \approx 52{,}852 - 30{,}000 = 22{,}852.$$

5.3 Liquidity Headroom and the LS-LMSR Subsidy

We define the *worst-case immediate-resolution deficit* at checkpoint i as

$$D(i) := \max\{0,\ S_Y(i),\ S_N(i)\}.$$

This is the additional amount, beyond collected customer premiums, that must be funded from the liquidity reserve if the event resolves immediately at checkpoint i.

At checkpoint t_3, the table gives $S_Y(3) = 7{,}148$ and $S_N(3) = -22{,}852$, hence

$$D(3) = \max\{0,\ 7{,}148,\ -22{,}852\} = 7{,}148 \text{ sats}.$$

Therefore, at t_3 the market requires only 7,148 sats of reserve funding to guarantee settlement in the worst case for this market configuration. The initial subsidy as calculated by $C(\mathbf{q_0})$ is the worst case loss that the market maker can experience by providing liquidity in any prediction market, which is why every market is pre-funded with $C(\mathbf{q_0})$ sats prior to listing.

6 Applications of Event Contracts

Event contracts convert uncertainty about a specified outcome into a fully collateralized claim with objective settlement. A trader pays a premium in sats according to the cost function and receives a fixed payoff if the event occurs, so the maximum potential loss is known at the time of trade (and there is no margin or liquidation mechanism). The traded price is naturally interpreted as a market-implied probability, so the instrument simultaneously provides a hedge, a return opportunity, and a quantitative forecast. These properties are well matched to risks in Bitcoin-native finance that are not handled cleanly by spot exposure or linear instruments.

We begin with the most direct hedging use-case: a Bitcoin holder who wishes to insure against underperformance relative to a benchmark over a fixed horizon. This contract family is simple to define, settles on public price data, and makes the hedge objective explicit. It also illustrates the general pattern that recurs throughout this section. A payoff can be written to match the economic loss state, and the market price can be read as a probability that supports transparent sizing and risk limits.

6.1 Hedging Bitcoin Underperformance with Event Contracts

Bitcoin holders face a practical risk that is not captured by simple spot volatility. The risk is *relative underperformance* versus conventional benchmarks that anchor institutional portfolios and liabilities, such as broad equity indices and commodity safe havens. From a portfolio-construction standpoint, the relevant object is not the unconditional distribution of Bitcoin returns, but the joint distribution of Bitcoin returns with other assets, especially their covariances [21]. Empirically, Bitcoin has exhibited meaningful co-movement with technology-heavy equity indices in some samples, while its relationship with commodities such as gold is often weaker or unstable across methods and horizons [1, 14]. This motivates a hedging layer that pays explicitly when Bitcoin *underperforms* a selected benchmark over a specified horizon.

Fix a horizon T (e.g. one month or one year) and let $P_X(0)$ and $P_X(T)$ denote the benchmark price at trade time and resolution time, respectively, for $X \in \{\mathrm{BTC}, \mathrm{SPX}, \mathrm{IXIC}, \mathrm{Gold}, \mathrm{NVDA}, \mathrm{TSLA}\}$. Define the log return

$$R_X := \log\left(\frac{P_X(T)}{P_X(0)}\right).$$

For each benchmark A we define a binary relative-performance event contract

$$Y_A := \{R_A - R_{\mathrm{BTC}} \geq 0\}, \qquad N_A := \{R_A - R_{\mathrm{BTC}} < 0\}.$$

Each contract pays 100 sats if its outcome occurs and 0 otherwise. At market open, we assume the market is seeded at a neutral prior 50–50, so the initial displayed odds are 50 sats on Y_A and 50 sats on N_A.

The hedging interpretation is direct. A Bitcoin holder who wishes to insure against relative underperformance buys Y_A contracts ("A beats BTC") on several benchmarks A. When Bitcoin underperforms, the hedge pays 100 sats per contract. When Bitcoin outperforms, the hedge expires worthless and the premium is the hedging cost.

We now show a worked example illustrating (i) market listing and odds, (ii) expected-value arithmetic under a 50–50 start, and (iii) how one can size positions using a fractional Kelly rule and then adjust the allocations using mean–variance ideas.

6.1.1 Worked example: a diversified relative-performance hedge

Consider five yearly markets:

$$Y_{\text{Gold}}:\ \{R_{\text{Gold}} - R_{\text{BTC}} \geq 0\},\quad Y_{\text{SPX}}:\ \{R_{\text{SPX}} - R_{\text{BTC}} \geq 0\},\quad Y_{\text{IXIC}}:\ \{R_{\text{IXIC}} - R_{\text{BTC}} \geq 0\},$$

$$Y_{\text{NVDA}}:\ \{R_{\text{NVDA}} - R_{\text{BTC}} \geq 0\},\quad Y_{\text{TSLA}}:\ \{R_{\text{TSLA}} - R_{\text{BTC}} \geq 0\}.$$

Assume each market starts at price 50 sats on Y and 50 sats on N. Let the hedger have a forecasting model that outputs subjective probabilities

$$p_{\text{Gold}} = 0.56,\quad p_{\text{SPX}} = 0.53,\quad p_{\text{IXIC}} = 0.51,\quad p_{\text{NVDA}} = 0.49,\quad p_{\text{TSLA}} = 0.48.$$

These probabilities are illustrative. They can reflect a view that Bitcoin is comparatively likely to underperform gold and broad equities over the next year, but less likely to underperform idiosyncratic single-name equities. The empirical motivation for treating technology indices as more coupled to Bitcoin than gold is consistent with evidence of a positive Bitcoin–Nasdaq relationship in some samples and weak or insignificant gold effects in some specifications [1].

At price 50, a Y-contract has expected profit (in sats)

$$\mathbb{E}[\Pi] = 100p - 50 = 100(p - 0.5),$$

and expected return on premium (ROI) is

$$\text{ROI} = \frac{100p - 50}{50} = 2p - 1.$$

Therefore:

$$\text{ROI}_{\text{Gold}} = 12\%,\quad \text{ROI}_{\text{SPX}} = 6\%,\quad \text{ROI}_{\text{IXIC}} = 2\%,\quad \text{ROI}_{\text{NVDA}} = -2\%,\quad \text{ROI}_{\text{TSLA}} = -4\%.$$

A hedger focused purely on expected value would avoid negative-ROI hedges. A hedger focused on tail-risk reduction may still buy some negative-ROI insurance, but the framework makes the tradeoff explicit.

For sizing, the simplest Kelly case is a event contract with cost of c sats that pays 100 sats if it wins and 0 otherwise. Per unit premium c, the net win multiple is

$$b = \frac{100 - c}{c}.$$

The full Kelly fraction of bankroll to allocate to this trade is the classical expression

$$f^\star = \frac{bp - (1 - p)}{b},$$

and at $c = 50$ we have $b = 1$ so

$$f^\star = 2p - 1.$$

Because full Kelly is typically too aggressive under model error and produces large drawdowns, practitioners often use *fractional Kelly* [17, 7]. Let $\lambda \in (0, 1)$ be the Kelly fraction (e.g. $\lambda = 1/2$). Then the proposed allocation to each positive-edge market is

$$f_i = \lambda(2p_i - 1)_+, \qquad (x)_+ = \max\{x, 0\}.$$

With $\lambda = 1/2$, we obtain

$$f_{\text{Gold}} = 0.06, \quad f_{\text{SPX}} = 0.03, \quad f_{\text{IXIC}} = 0.01,$$

and $f_{\text{NVDA}} = f_{\text{TSLA}} = 0$ under an EV-first rule.

This produces an implementable hedging budget. If the bankroll dedicated to hedging is $W = 10{,}000{,}000$ sats, then the premiums allocated are

$$W f_{\text{Gold}} = 600{,}000, \quad W f_{\text{SPX}} = 300{,}000, \quad W f_{\text{IXIC}} = 100{,}000 \text{ sats}.$$

Assuming a trader can buy contracts with an average cost of 50 sats per contract (this would require the prices to be below 50 sats prior to the trade being executed), the contract counts are

$$q_{\text{Gold}} = \frac{600{,}000}{50} = 12{,}000, \quad q_{\text{SPX}} = 6{,}000, \quad q_{\text{IXIC}} = 2{,}000.$$

If, for example, Bitcoin underperforms gold over the year, the hedge payout from the gold leg is $100 q_{\text{Gold}} = 1{,}200{,}000$ sats, and the net profit on that leg is $1{,}200{,}000 - 600{,}000 = 600{,}000$ sats.

Finally, diversification is not about holding many contracts. It is about reducing covariance among payoffs [21]. Let X_i denote the random return of the ith hedge leg per unit premium. A mean–variance adjustment chooses weights $\mathbf{w}$ to target a desired expected return while reducing variance:

$$\max_{\mathbf{w} \geq 0} \ \mathbf{w}^\top \boldsymbol{\mu} - \frac{\gamma}{2} \mathbf{w}^\top \Sigma \, \mathbf{w}, \qquad \sum_i w_i \leq 1,$$

where $\boldsymbol{\mu}$ and Σ are the mean vector and covariance matrix of the hedge-leg returns [21]. In practice, one estimates Σ from historical data or from a risk model. The empirical point is that a gold-relative hedge can behave differently from a Nasdaq-relative hedge because Bitcoin can be more coupled to Nasdaq than to gold in some samples, so mixing both legs can reduce the chance that all hedges fail in the same regime [1].

6.2 Generating ROI with Event Contracts

An event contract is a priced probability claim. If the market-implied probability for event Y is m (so the contract trades at $100m$ sats), and a trader has an independent probability estimate p, then the expected profit per contract is

$$\mathbb{E}[\Pi] = 100p - 100m = 100(p - m).$$

Positive expected value therefore corresponds to $p > m$ on a "Yes" contract and $p < m$ on a "No" contract. Profitability comes from identifying miscalibrated public odds and allocating capital with a disciplined risk rule rather than taking blind risk on everything [7, 17].

A useful interpretation is to treat the market price as an aggregate prior and an individual model as a private signal. Bill Benter emphasizes that a purely fundamental model can be biased relative to the public odds, and that a practical path to profitability is to combine the model and the public probability estimate rather than ignoring the public [7].

Position sizing follows the same Kelly logic as above. For a cost of c sats, net win multiple $b = (100 - c)/c$, and win probability p, the full Kelly fraction is

$$f^\star = \frac{bp - (1 - p)}{b}.$$

Because estimation error and drawdown constraints matter, fractional Kelly is standard in practice [17, 7]. If $f = \lambda f^\star$ with $\lambda \in (0, 1)$, then a trader can bound risk while retaining a large fraction of the long-run log-growth objective.

A simple subsidized-start example clarifies the economics. Suppose a market is listed at a neutral seed $m = 0.5$ (price $c = 50$), and your model estimates $p = 0.58$. Then $\mathbb{E}[\Pi] = 100(0.58 - 0.5) = 8$ sats per contract, so the expected ROI is $8/50 = 16\%$. At $c = 50$, the full Kelly fraction is $f^\star = 2p - 1 = 0.16$, and a conservative fractional Kelly at $\lambda = 1/2$ allocates 8% of bankroll. This is the basic route by which a forecasting model, if genuinely informative, can be converted into positive expected value without uncontrolled leverage.

6.3 Simple Trading Strategies with High Accuracy Forecasts

Event contracts also provide a bridge from probabilistic forecasts to conventional trading strategies in linear or leveraged instruments. Consider a daily directional event on an index, such as

$$Y := \{\text{S\&P 500 closes up on date } t\}, \qquad N := \{\text{S\&P 500 closes down (or non-up) on } t\}.$$

If the market-implied probability is m and your forecast probability is p, then the event contract itself has expected profit $100(p - m)$ sats as above.

To translate this signal into a futures or perpetuals position, model the one-day index return as taking two representative values, $+r$ in the up state and $-r$ in the down state. Then a forecast p implies an expected one-day return

$$\mu = p \cdot r + (1 - p)(-r) = (2p - 1)r.$$

If you trade a linear instrument with daily return approximately μ and daily variance approximately σ^2, a standard continuous-time Kelly approximation suggests sizing proportional to μ/σ^2. In practice, volatility is time-varying and heavy-tailed in crypto and related markets, so volatility forecasting and crash risk matter for any leverage-based strategy [22, 20]. This is exactly why event contracts are operationally attractive: they bound downside to the premium and remove liquidation dynamics.

Beyond a single directional trade, a probability stream supports several simple strategy templates. A basic implementation is a threshold rule: take risk only when the edge is large enough to plausibly clear spreads, fees, and slippage, e.g. trade only if $|p - m| \geq \delta$ for a calibrated $\delta > 0$. Because the forecast updates continuously, the same rule naturally implies rebalancing: exposures are increased when p moves further from m and reduced when the signal weakens, subject to hard leverage caps and drawdown limits.

Forecasting is most valuable when it is *calibrated*. A model that outputs probabilities should be evaluated on calibration and forecast error, not only hit rate. The platform-level point is that event contracts produce a standardized object for evaluation: each market is a well-defined question with an objectively verifiable settlement rule. This makes it feasible to measure whether a strategy produces persistent edge, and to adjust capital allocation rules over time using disciplined fractional Kelly sizing rather than ad hoc leverage [17, 7].

6.4 Event Contracts for Bitcoin Mining

Bitcoin mining is a probabilistic, winner-take-all contest. A miner invests computational work and wins the right to author the next block with probability proportional to their share of global hashrate.

Inter-block times are well modeled as exponential, so the number of blocks found over a fixed horizon is well approximated by a Poisson process [27, 28, 13]. The protocol adjusts the difficulty every 2016 blocks so that the average inter-block interval remains approximately ten minutes [27].

6.4.1 Mining variance and the economic role of pools

Solo mining inherits the variance of a Poisson arrival process. With hashrate h, difficulty D, block reward B, and horizon t, the expected block count is $\lambda = ht/2^{32}D$, and the variance of blocks found is also λ [28]. This implies extreme payout dispersion for small miners, including long droughts in which no block reward is earned [28, 29].

Mining pools exist to reduce this variance by paying miners against submitted shares. Pools implement reward-sharing schemes such as Full-Pay-Per-Share (FPPS), which minimizes miner variance but shifts risk to the pool operator, and Pay-Per-Last-N-Shares (PPLNS), which reduces variance but still exposes miners to pool luck [29, 28, 25]. Empirically, pooling pressure has contributed to concentrated mining power, and even within large pools a small number of actors can receive a majority of payouts [27]. This concentration matters because Bitcoin's neutrality and censorship resistance rely on the absence of controlling coalitions in block production [25, 19].

6.4.2 Mining pool game theory and malicious incentives

Pooling changes the strategic landscape. Competition among pools has been linked to adversarial behavior, including denial-of-service and block-withholding-style attacks [27]. In particular, block withholding can be profitable at the pool level via infiltration, creating a prisoner's-dilemma structure where mutual attack can be an equilibrium even though both sides would be better off without attacking. These dynamics can push miners away from open pools and toward closed pools and coalitions [15].

Within pools, reward schemes can also create internal incentive problems. In PPLNS pools, miners may benefit by delaying share reports, and incentive compatibility depends on the relative power of the largest miner [32]. More broadly, models that treat mining as a setting with a small number of dominant players and many small players imply structural incentives to merge into larger entities, increasing the value-per-unit of resources as coalitions grow [19].

These forces jointly motivate a design goal. Variance must be reducible without increasing the payoff to consolidation. A market-based hedge that is available to small miners and small pools can relax the variance pressure that otherwise favors large pools [25].

6.4.3 Miner Hedging with LS–LMSR Event Contracts

This subsection shows how Bitcoin-native event contracts can be used to hedge two mining risks that matter operationally. The first is the discrete revenue shock induced by a positive difficulty adjustment. The second is per-block payout variance, which is intrinsic to the winner-take-all nature of mining and is a primary driver of pooling pressure. Mining concentration is empirically observable and is economically important because decentralization supports Bitcoin's neutrality and censorship resistance [27, 19]. Moreover, mining pools face strategic incentives that can be socially harmful, including block-withholding incentives that form a prisoner's-dilemma structure among pools [15] and inter-pool attack dynamics that can affect long-run viability [18]. A hedging layer that reduces variance

without requiring miners to migrate to the largest pools can therefore be interpreted as decentralization-supporting infrastructure [25].

Throughout, contracts pay 100 sats if their outcome is realized and 0 otherwise. Trades are executed by a liquidity-sensitive LMSR cost function with a virtual seed state. In addition, we assume an explicit execution subsidy that discounts cost-function charges by a factor $\rho \in (0, 1)$. If a trader moves the state from $\mathbf{q}_{\text{old}}$ to $\mathbf{q}_{\text{new}}$, the raw cost-function charge is

$$\Delta C := C(\mathbf{q}_{\text{new}}) - C(\mathbf{q}_{\text{old}}),$$

and the trader pays $\rho\,\Delta C$. We take $\rho = 0.5$ in the worked examples below. The 100-sat resolution payouts are unchanged, so ρ is a direct liquidity subsidy.

6.4.4 Market A: "Will the next difficulty adjustment go up?"

Bitcoin retargets mining difficulty every 2016 blocks to stabilize expected block time. A positive adjustment reduces expected sats earned per unit of hashrate in the subsequent epoch, holding the fee environment fixed. This creates a discrete, objectively verifiable revenue shock and therefore a natural hedging target.

We define the binary event contract

$$Y := \{\text{the next difficulty adjustment is positive}\},$$
$$N := \{\text{the next difficulty adjustment is non-positive}\}.$$

Let the LS–LMSR state be $\mathbf{q} = (y, n)$. We use the Bitcoin-denominated liquidity-sensitive cost function

$$b(\mathbf{q}) = \alpha(y + n), \qquad C(y, n) = 100\, b(\mathbf{q}) \log\left(\exp\left(\frac{y}{b(\mathbf{q})}\right) + \exp\left(\frac{n}{b(\mathbf{q})}\right)\right).$$

A trade that moves the market from $\mathbf{q}_{\text{old}}$ to $\mathbf{q}_{\text{new}}$ has raw charge ΔC, and the subsidized cash paid is $\rho\,\Delta C$.

To size a hedge, let R_0 denote the miner's expected sats revenue over the next epoch under baseline difficulty. Let $\delta > 0$ denote a modeled conditional difficulty increase given that Y occurs. A simple revenue-impact approximation is

$$L \approx R_0 \left(1 - \frac{1}{1 + \delta}\right) = R_0 \frac{\delta}{1 + \delta},$$

which is the expected revenue shortfall the miner wishes to offset. If the miner buys Δy contracts on Y, then the gross payoff upon Y is $100\Delta y$ sats. Under a subsidy factor ρ, the net cash received in the Y state is

$$100\Delta y - \rho\,\Delta C.$$

We choose Δy so that this net amount approximately matches L.

6.4.5 Worked example A: difficulty hedge with a large seed state

We initialize the market at a large initial subsidy to support large miner flows with modest slippage,

$$\mathbf{q}_0 = (500{,}000; 500{,}000), \qquad \alpha = 0.111, \qquad \rho = 0.5.$$

Then the initial subsidy is

$$C(\mathbf{q}_0) \approx 57{,}693{,}934 \text{ sats.}$$

A miner buys $\Delta y = 100{,}000$ contracts on Y, moving the state to

$$\mathbf{q}_1 = (600{,}000; 500{,}000).$$

The raw cost-function charge is

$$\Delta C = C(\mathbf{q}_1) - C(\mathbf{q}_0) \approx 6{,}765{,}750 \text{ sats},$$

so the premium paid is

$$\rho\, \Delta C \approx 3{,}382{,}875 \text{ sats.}$$

If Y occurs, the hedge pays $100\Delta y = 10{,}000{,}000$ sats, hence net hedge cash in the Y state is

$$10{,}000{,}000 - 3{,}382{,}875 = 6{,}617{,}125 \text{ sats.}$$

If N occurs, the hedge pays 0 and the miner's loss is the premium 3,382,875 sats.

To connect the hedge to mining economics, suppose the miner's expected epoch revenue is $R_0 = 1$ BTC $= 100{,}000{,}000$ sats and conditional on Y the miner models a difficulty increase of $\delta \approx 7.1\%$. Then the modeled revenue shortfall is

$$L \approx 100{,}000{,}000 \cdot \frac{0.071}{1.071} = 6,629,318.39 \approx 6.6 \times 10^6 \text{ sats},$$

which is approximately offset by the computed hedge payoff 6,617,125 sats.

6.4.6 Market B: "Which pool mines the next block?"

Block discovery is a winner-take-all process with high payout variance when the win probability is small. Variance reduction is a principal reason miners join pools. However, pooling also concentrates block production, and concentration is empirically visible in recent hashrate-share data. Concentration matters because mining coalitions can create censorship and neutrality risks [19, 27]. Mining pools can also face strategic adversarial incentives, including block-withholding-style dynamics that can be rational in equilibrium [15]. A hedging mechanism that reduces the cost of variance for smaller participants can therefore reduce the structural pressure toward ever-larger pools [25].

The natural informational question is multi-outcome:

$$\Omega = \{\text{Foundry},\ \text{AntPool},\ \text{ViaBTC},\ \text{F2Pool},\ \text{SpiderPool},\ \text{MARA},\ \ldots,\ \text{Other}\},$$

where the realized outcome is the identity of the pool that mines the next block. A multi-outcome LS–LMSR represents this as a single probability vector over pools. Let $\mathbf{q} = (q_1, \ldots, q_m)$ be the state. We define

$$b(\mathbf{q}) = \alpha \sum_{j=1}^{m} q_j, \qquad C(\mathbf{q}) = 100\, b(\mathbf{q}) \log\left(\sum_{j=1}^{m} \exp\left(\frac{q_j}{b(\mathbf{q})} \right) \right).$$

For hedging, miners typically need a complement payoff, i.e., a claim that pays when their pool does *not* find the next block. This is the "bet against your own pool" structure that turns an all-or-nothing

payout into a smoother cash flow and can reduce pooling pressure [25]. Operationally, the cleanest interface is therefore an event contract per major pool i:

$$Y_i := \{\text{pool } i \text{ mines the next block}\}, \qquad N_i := \{\text{pool } i \text{ does not mine the next block}\}.$$

These binaries can be derived from the multi-outcome market as a quoting convenience, but the economics are identical: they pay exactly in the drought state that causes miner stress.

6.4.7 Worked Example B: Hedging Hashrate Against Block Reward Variance

We consider a single pool-specific binary market, e.g. Foundry:

$$Y := \{\text{Foundry mines the next block}\}, \qquad N := \{\text{Foundry does not mine the next block}\}.$$

We use the same large-seed configuration as above,

$$\mathbf{q}_0 = (500{,}000, 500{,}000), \qquad \alpha = 0.111, \qquad \rho = 0.5.$$

Suppose the miner's incremental payout if Foundry mines the next block is $b = 1{,}000{,}000$ sats, and 0 otherwise for that block. The miner purchases $\Delta n = 14{,}212$ contracts on N, moving to

$$\mathbf{q}_1 = (500{,}000, 514{,}212).$$

The raw cost-function charge is

$$\Delta C = C(\mathbf{q}_1) - C(\mathbf{q}_0) \approx 842{,}358 \text{ sats},$$

so the subsidized premium paid is

$$\rho\, \Delta C \approx 421{,}179 \text{ sats}.$$

If N occurs, the hedge pays $100\Delta n = 1{,}421{,}200$ sats, hence net hedge cash is

$$1{,}421{,}200 - 421{,}179 \approx 1{,}000{,}021 \text{ sats},$$

which approximately matches the target b. If Y occurs, the hedge pays 0 and the miner loses the premium 421,179 sats.

Outcome	Mining payout	Hedge net payout	Total
Foundry mines (Y)	1,000,000	−421,179	578,821
Foundry does not mine (N)	0	+1,000,021	1,000,021

This is the basic variance-hedging geometry, and the prediction markets odds would be continuously recalibrated in real-time. However, it shows if the odds have a significant deviation from the true likelihood (Foundry has approximately 20-30% of global hashrate, while we priced this initial Event Contract at the platform default of 50% probability) then the markets can be highly profitable for hedgers or speculators alike. The opportunity for profit therefore comes when the market-based probability is significantly different from the true probability.

Mining is an all-or-nothing contest at the block level, and the hedge is a complementary all-or-nothing payoff that triggers precisely when the block is not won. With suitable depth and an explicit execution subsidy, the miner can transform a highly volatile per-block revenue stream into a controlled cash-flow profile without requiring migration to the largest pools [25].

6.4.8 How Hedging Hashrate Supports Mining Decentralization

Bitcoin mining is designed as a probabilistic winner-take-all contest [13]. Even when hashrate shares are stable, the realized winner of the next block is uncertain, and Bitcoin miner revenue arrives in lumpy, path-dependent bursts rather than as a smooth cash flow. This variance is not an accident. It is a direct implication of Bitcoin's permissionless security model, where block production is allocated by open competition rather than by a trusted scheduler [23]. In practice, however, variance creates strong pressure toward pooling and toward payout schemes that smooth miner income [28, 29]. Empirically, mining power is concentrated in a small number of pools, and concentration can also arise within pools [27]. These dynamics matter because permissionless systems rely on the absence of controlling coalitions at the settlement layer, and coalition incentives can naturally push toward mergers and centralization [19].

Mining outcomes also remain intrinsically uncertain at decision-relevant horizons. For most pools, "will this pool mine the next block?" has probability far from 0 or 1, and does not converge to certainty prior to resolution. The same is true of short-horizon mining risks that matter operationally, such as adverse luck over a window and discrete protocol shocks like a positive difficulty adjustment. Persistent uncertainty sustains two-sided trading demand and repeated settlement cycles, which makes these mining-hedge markets economically attractive to run at scale. In this sense, mining variance is not only a risk for miners. The variance and uncertainty of the system itself is a structural source of market-making profitability for prediction market liquidity provision.

A profitable hedging venue can also generate a positive side effect for the Bitcoin ecosystem by functioning as a variance-insurance layer. Pool selection is partly an insurance decision. PPLNS-style pools expose miners to more luck variance, while FPPS-style pools reduce miner variance by shifting risk onto the operator [28, 29, 32]. This tends to favor larger pools and well-capitalized operators, increasing concentration pressure [27, 19]. Mining-hedge markets target the root economic driver. If miners can buy protection that pays in "drought" states, smaller PPLNS pools can attract miners without requiring them to bear intolerable variance. If small FPPS operators can hedge negative luck relative to expected block share, they can offer stable payouts with less balance-sheet strain [25]. By lowering the variance premium that otherwise pushes participants toward the largest pools, hedging markets enlarge the feasible set of decentralized pooling equilibria and can reduce concentration pressure in practice [27, 19].

This decentralization side effect is significant because Bitcoin's monetary value proposition depends on credible neutrality and censorship resistance. Bitcoin was proposed as a peer-to-peer electronic cash system precisely to enable value transfer without reliance on trusted intermediaries [23]. Bitcoin's distinctive role is as neutral monetary infrastructure ensures that it is harder to capture, censor, or selectively exclude than institutionally permissioned payment systems [5, 3]. On this view, neutrality is not a branding choice. It is an institutional property grounded in the absence of concentrated control points in the settlement process. Because mining determines transaction inclusion and final settlement, concentration in mining can become a practical chokepoint, weakening the system's neutrality. Accordingly, any mechanism that reduces structural incentives toward mining centralization supports the credibility of Bitcoin as neutral money, and thereby supports the long-run value proposition of Bitcoin-native financial infrastructure.

7 Conclusion

We have described a Bitcoin-native prediction market that uses an automated market maker to convert dispersed, time-varying beliefs into a continuously updated public forecast on clearly stated, objectively verifiable financial events. The mechanism is engineered around two design requirements. First, trade execution is determined by a cost-function rule that provides a clear, auditable accounting of cash-in versus contingent payout, together with a finite and explicitly controlled worst-case exposure set by the initial liquidity subsidy. Second, market depth adapts to participation so that early markets remain responsive while mature markets become progressively more stable, improving execution quality and reducing volatility as open interest grows. These properties make the system suitable for fully collateralized, Bitcoin-denominated event contracts that can be operated with transparent risk limits.

The broader claim is that the platform's primary product is not trade flow but the forecast itself: a structured probability signal that can be reused for decision-making across time horizons and asset classes. By separating execution economics from the probability representation shown to users, the design preserves rigorous settlement guarantees while still producing an interpretable "weather map" of financial uncertainty. What remains is empirical validation and operational hardening: calibrating parameters against real order flow, measuring forecast calibration and error under realistic participation, and extending the same settlement-and-risk framework to more diverse market mechanisms while maintaining the same clarity of guarantees.

A Bitcoin as a High-Value Target for Forecasting

Bitcoin is a digital bearer asset native to a permissionless payment network [23]. It is neither an equity claim nor a contractual right against an issuer. Accordingly, it has no management team, no cash-flow schedule, no dilution events, and no privileged disclosure cycle. Its market price is therefore best interpreted as a market-clearing statistic for a globally traded monetary commodity.

A.1 Bitcoin as neutral monetary infrastructure

Money is a functional kind, and a monetary asset is evaluated by the extent to which it can serve, at least locally, as a means of exchange, a store of value, and a unit of account [5, 23]. Bitcoin partially realizes the digital cash ideal by combining final settlement on a public ledger with payment-layer constructions that support high-frequency exchange at low marginal cost while preserving ultimate settlement on the base layer [6]. In this sense, Bitcoin can be treated as neutral monetary infrastructure: it supports exchange without an issuer narrative, and it does so on rails designed to be broadly accessible (permissionless) rather than institutionally gatekept [5, 3, 4].[1] Even when Bitcoin is not the dominant unit of account for the broader economy, it remains a coherent unit of account within Bitcoin-native systems, that is, in applications whose liabilities, collateral, and settlement are natively denominated in bitcoin units (sats) [2].

A.2 Why Bitcoin is a natural focus for prediction-market forecasting

Prediction markets are marketplaces for information in which trades aggregate dispersed beliefs into a publicly observable price. For such markets, the settlement asset should be digitally transferable, operationally neutral with respect to jurisdictional payment intermediaries, and natively compatible with global, continuous online participation. Bitcoin is well positioned on these criteria insofar as it is designed to transfer value and can support exchange without reliance on an issuing firm or a centrally administered payments operator [23, 3, 2]. This provides a principled justification for Bitcoin-native event contracts: the same object that functions as digital cash for internet commerce can also function as the settlement medium for information commerce. On the mechanism side, cost-function market makers and market scoring rules provide a standard way to transmute sequential trading into probability outputs [16, 11, 12]. Liquidity-sensitive automated market makers further permit market depth to adapt to growing participation [24].

Bitcoin is also unusually well suited as a forecasting target. Because it is globally traded and information arrives continuously, there is no privileged disclosure cycle to anchor belief updates. Because it is a bearer-like asset, its market microstructure is less constrained by issuer actions such as buybacks, dividends, or guidance. For a forecasting business, these features concentrate the forecasting problem

[1] Bitcoin (the protocol and bearer asset) is conceptually distinct from the regulated intermediaries that custody it, transmit it, or broker access to it. The protocol does not itself implement identity, suitability, reversibility, dispute resolution, or transaction-monitoring functions. Accordingly, legal duties and operational risks arise primarily at the application interface where a firm intermediates customer access to the network, including activities commonly regulated as custody, money transmission, exchange, and settlement. For this reason, Glimpse Ltd. treats Bitcoin-native settlement as technologically permissionless externally but governed internally: to the extent we hold or transfer Bitcoin on behalf of customers, we apply controls consistent with applicable financial crime, safeguarding, and market-conduct expectations, including customer due diligence, transaction monitoring, sanctions screening, and governance over custody and settlement processes.

on the dynamics of collective belief under uncertainty rather than on idiosyncratic corporate actions. In short, Bitcoin generates a steady stream of verifiable, decision-relevant questions with clean settlement rules, which is exactly the substrate prediction markets require.

A.3 Long-horizon structure and descriptive power-law heuristics

A recurring empirical claim is that Bitcoin's long-run price evolution is approximately consistent with a power-law relationship in time (equivalently, a linear relationship in log-log space),

$$P(t) = At^{\alpha}, \qquad \log P(t) = \log A + \alpha \log t,$$

often motivated by network-growth dynamics and diffusion-style adoption effects [30, 9, 26, 31]. We treat these external frameworks as heuristic motivation rather than as proof. Our use of this functional form is descriptive and intended only to characterize long-horizon structure in the realized series.

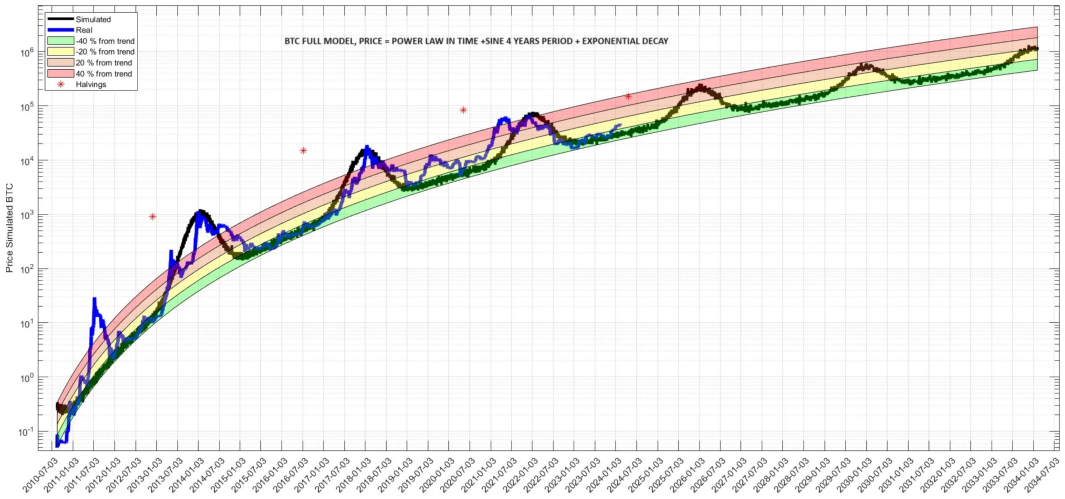

Figure 4: Illustrative power-law framing from prior work (included for intuition; not relied on as proof).

Using $N = 5{,}499$ observations, we fit the one-factor model

$$\log(\text{index}_t) = \beta_0 + \beta_1 \log(\text{time}_t) + \varepsilon_t,$$

and obtain $R^2 = 0.963$ with $\beta_1 = 5.8065$ (95% CI: $[5.777,\ 5.837]$) and $\beta_0 = -39.0236$. This fit indicates that a simple log-log specification explains a large fraction (over 95%) of long-horizon variance in the historical series. The regression is descriptive and does not imply future adherence to the same relationship, particularly under structural regime change.

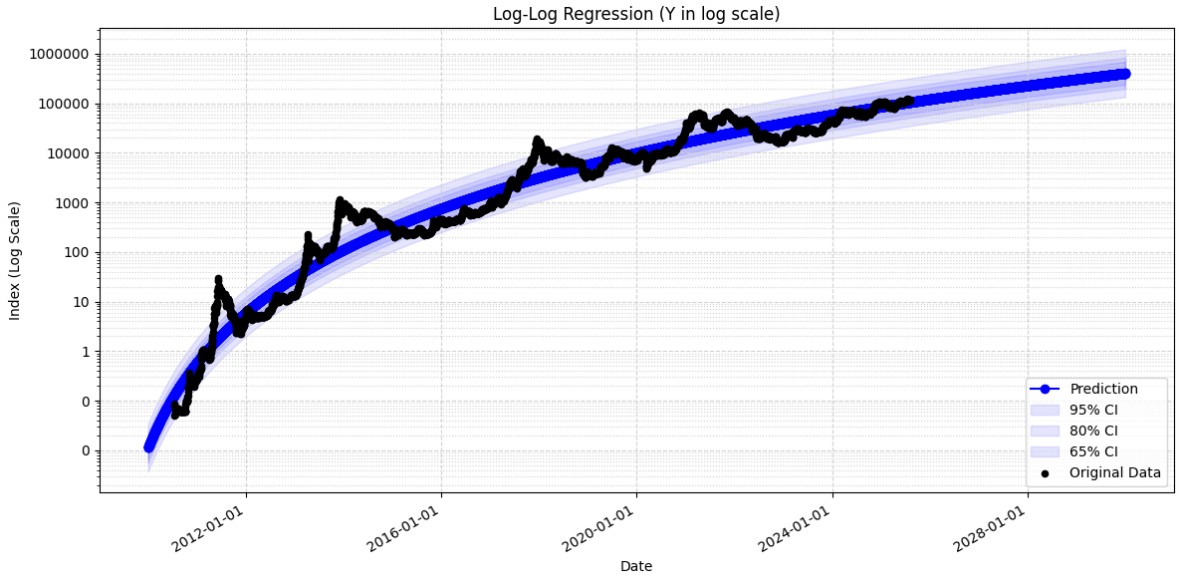

Figure 5: Independent log-log fit and confidence bands (illustrative). Black: observed series. Blue: fitted log-log trend with confidence intervals.

A.4 Short-Horizon Uncertainty and Why Probabilities Matter

Long-horizon structure in price levels can coexist with substantial short-horizon uncertainty in returns. A log–log regression concerns the slow-moving component of $\log P(t)$ as a function of time and can therefore achieve high in-sample R^2 even when short-run returns remain difficult to forecast. This is not a contradiction. It reflects the separation between smooth long-run movements in the price level and highly variable day-to-day and week-to-week changes.

Over decision-relevant horizons, Bitcoin returns are empirically heavy-tailed, meaning extreme moves occur more often than a Gaussian benchmark would predict. Volatility is also clustered, meaning large moves tend to be followed by large moves, while quiet periods tend to persist [10]. For this reason, empirical studies frequently model Bitcoin risk using conditional-variance frameworks such as GARCH and multivariate extensions, which formalize the practical point that the current risk level depends on recent market turbulence and can vary substantially over time [10, 22].

Crash episodes are particularly salient because they are abrupt relative to typical movements in the underlying trend and because they are economically consequential. This motivates empirical approaches that treat crashes and large drawdowns as low-frequency but decision-relevant events rather than as negligible residual noise [20]. More broadly, short-horizon uncertainty can shift with macro conditions and cross-asset linkages. In particular, empirical work has documented meaningful relationships between Bitcoin and technology equity indices and other macro-financial indicators over some samples, while other traditional hedges such as gold may be weaker or unstable depending on specification and horizon [1, 14].

These facts motivate the forecasting product that event contracts provide. When risk is high and unstable, market participants often need an interpretable statement of which outcomes are likely over a fixed horizon, rather than an undifferentiated exposure to the spot asset. Event contracts implement

this translation by converting a price question into a proposition with a clear settlement rule, such as "BTC is above X by date T," "BTC experiences a drawdown of at least $d\%$ over horizon T," or "BTC outperforms benchmark B over horizon T." Trading then produces a market price that can be read as a market-implied probability, which is a practically usable forecast to the extent that historical calibration and hit rates are strong. Because contracts can be defined on thresholds, relative performance, and crash-like events, they can also provide targeted hedges that address specific adverse scenarios without requiring continuous rebalancing in the underlying asset [22, 20, 1].

A.5 Bitcoin Mining, Network Difficulty, and Block Reward Variance

Bitcoin mining is a large, globally distributed industry with cash flows that are intrinsically stochastic and strongly regime-dependent. At the protocol level, block production is a race in which the probability of winning the next block is approximately proportional to a miner's share of global hashrate. Over intervals where the global hashrate and difficulty are approximately constant, the time to the next block is well modeled as exponential and block arrivals are well approximated by a Poisson process. The resulting payout stream is therefore lumpy even when a miner's hashrate share is stable.

This uncertainty caused by variance inherient to the Bitcoin protocol creates real business risks for Bitcoin miners. For a miner with hashrate h facing network difficulty D, the expected number of blocks found over a horizon t scales linearly in ht/D, and the variance in blocks found is of the same order as the mean. Small and mid-sized miners therefore face large relative dispersion, including long droughts with zero block rewards, which directly motivates variance-reducing institutional structures such as mining pools and share-based payout schemes. In classical analyses of pooled mining, these schemes can be viewed as contracts that trade off immediacy, operator balance-sheet risk, and the variance borne by individual miners, and the details of reward design matter for both fairness and attack surface.

The difficulty adjustment introduces an additional, discrete source of uncertainty that is economically first-order for operational planning. Bitcoin retargets difficulty every 2016 blocks to stabilize expected block time near ten minutes. The retarget rule is a feedback mechanism that depends on realized block times over the previous epoch. Consequently, when global hashrate grows rapidly within an epoch, blocks tend to arrive faster than the ten-minute target during that epoch, and the subsequent difficulty adjustment tends to be positive. A positive adjustment reduces expected bitcoin earned per unit of hashrate in the next epoch, holding fees and other conditions fixed. This creates a predictable *type* of shock, but an uncertain *magnitude* and timing, because the adjustment is triggered by a random block-arrival process and because the global hashrate is itself time-varying.

Block rewards also contain a second source of variance beyond win probability. The per-block miner revenue is the sum of the protocol subsidy and transaction fees. While the subsidy is deterministic over a halving epoch, transaction fees are time-varying and can be materially larger or smaller across blocks. As fees become a larger share of miner revenue, the variability of realized per-block revenue increases, and reward-sharing schemes that are not carefully aligned to the timing of fee variability can become economically distortive.

These features jointly make mining a natural forecasting target for Bitcoin-native event contracts. Unlike generic price forecasting, mining variables are operational primitives. They map directly into balance-sheet and cash-flow risk, they are observed on-chain or from widely monitored network statis-

tics, and they admit objective settlement rules. The contract designs below illustrate how difficulty dynamics, block-arrival variance, and fee regimes can be translated into probability forecasts that miners and other Bitcoin participants can use for planning and hedging.

Difficulty and hashrate event contracts target the protocol variable that turns global competition into miner unit economics. The basic forecasting question is concrete and settlement-clean: "Will the next difficulty adjustment be positive, and will it exceed a threshold δ"? These questions matter because difficulty directly determines how many sats some given amount of hashrate can be expected to earn in the next epoch, holding the fee environment fixed. A positive adjustment therefore induces a discrete revenue shock that miners must absorb immediately in operating margins. The dominant driver of miner profitability is still the Bitcoin price, because hashprice is increasing in BTC price and decreasing in difficulty, but difficulty-forecast markets isolate the protocol component of hashprice risk. They allow a miner to hedge the revenue impact of rising network competition even when the BTC price path is uncertain.

Block-arrival and luck event contracts target the intrinsic variance of winner-take-all block production. The operational question is again explicit: "Will a given pool mine fewer than k blocks over the next T blocks"? A complementary variant that is convenient for interface and frequent settlement is: will pool i fail to mine the next block? These questions matter because realized block counts over short horizons can deviate sharply from expectation even when hashrate share is stable. For miners and pool operators, under-production over a payroll, hosting, or debt-service window is a solvency and liquidity problem. A contract that pays precisely in the under-production state functions as variance insurance against cash-flow droughts. This has broader system relevance because variance pressure is a primary driver of pooling and of concentration dynamics. When variance is expensive to bear, miners rationally migrate toward the largest pools and the most capitalized payout schemes. A liquid hedging layer relaxes that pressure by allowing smaller participants to buy protection rather than to consolidate.

Fee-regime event contracts target the part of miner revenue and user cost that is most visible to the rest of the ecosystem. The forecasting question can be stated in a way that is directly useful for operations: "Will the (average) transaction fees per block exceed f BTC in block n (or between blocks H and $H + n$)"? Tail formulations are also natural, such as whether at least one block in the next n blocks contains total fees above a stress threshold $f_{\max}$. These questions matter to miners because fees are an increasingly important component of per-block revenue and a major source of variance conditional on winning a block. They matter to Lightning node operators because high-fee regimes raise the cost of channel opens, closes, and rebalancing, and can force changes in liquidity management. They matter to exchanges and payment processors because the expected probability of a high-fee regime determines whether to accelerate batching, delay non-urgent withdrawals, or adopt fee-smoothing policies. They matter to ordinary users because fees determine whether on-chain usage is economical on a given horizon. By turning near-term congestion and volatility conditions into market-implied probabilities with clean settlement rules, fee-regime contracts provide an interpretable forecast that can be acted on directly.

Taken together, these mining-adjacent forecasting targets decompose the economic uncertainty faced by Bitcoin participants into separable components. Price markets primarily address directional BTC exposure. Difficulty markets address protocol competition risk that moves hashprice mechanically. Block-arrival markets address payout timing risk that cannot be diversified away by small participants.

Fee-regime markets address the volatility of the user-cost and security-budget channel that links miners and transactors. The economic data of the Bitcoin network is a natural domain in which event contracts function as variance insurance rather than as generic speculation.

Mining event contracts also matter beyond miner balance sheets. Mining is increasingly entangled with energy markets because miners are unusually flexible electricity consumers. Miners can curtail consumption quickly, can locate near stranded supply, and can monetize curtailed renewable generation and other wasted energy. In grids with increasing shares of variable renewables, this flexibility is economically valuable. Mining therefore sits at the intersection of two volatile systems. Bitcoin-denominated revenue is volatile, and power prices and curtailment regimes can be volatile. Forecasting instruments that turn these uncertainties into tradeable probabilities can reduce the variance premium faced by miners and by capital providers to miners.

Mining markets are tightly coupled to the broader Bitcoin financial system through a real economic feedback loop. When mining is persistently unprofitable for a marginal set of operators, shutdowns reduce the effective hashrate, and the next difficulty adjustment partially restores expected profitability per unit of hashrate. This mechanism does not create a hard price floor for Bitcoin, but it does create a stabilizing channel for miner economics. Prediction markets that forecast difficulty, block reward variance, and future fees make this legible. They also create new derivative instruments for an industry whose core risk is variance rather than mere directional exposure.

A.6 Summary

Bitcoin is a compelling focus for forecasting because it is (i) a global monetary commodity whose price is primarily a market signal rather than an issuer narrative, (ii) plausibly shaped by measurable long-horizon regularities that can be described in simple functional forms, and (iii) sufficiently volatile over short and medium horizons that probability forecasts are decision-relevant. In addition, Bitcoin's origin as an attempt at peer-to-peer electronic cash provides an institutional rationale for Bitcoin-native information markets: a prediction market is a form of commerce, and a neutral, digitally native settlement medium strengthens the case for global participation in a Bitcoin economy [23, 5, 3].

B Proper scoring rules elicit truthful beliefs

Consider a binary event with outcomes $\Omega = \{Y, N\}$. A forecaster has a true belief

$$p = \Pr(Y) \in (0,1), \qquad \Pr(N) = 1 - p,$$

but reports a probability $r \in (0,1)$ for Y (and $1-r$ for N). A scoring rule specifies payoffs $s_Y(r)$ if Y occurs and $s_N(r)$ if N occurs. Under risk neutrality, the forecaster chooses r to maximize expected score

$$\mathbb{E}_p[s(r)] := p\, s_Y(r) + (1-p)\, s_N(r).$$

We first illustrate properness with the quadratic (Brier) rule in the binary setting. Take

$$s_Y(r) = -(1-r)^2, \qquad s_N(r) = -r^2.$$

Then

$$\mathbb{E}_p[s(r)] = -\Big(p(1-r)^2 + (1-p)r^2\Big).$$

Differentiating in r yields

$$\frac{d}{dr}\mathbb{E}_p[s(r)] = -2(r-p),$$

so the unique maximizer is $r^\star = p$.

The same incentive appears under the logarithmic score. Let

$$s_Y(r) = \log r, \qquad s_N(r) = \log(1-r).$$

Then

$$\mathbb{E}_p[s(r)] = p \log r + (1-p)\log(1-r),$$

and

$$\frac{d}{dr}\mathbb{E}_p[s(r)] = \frac{p}{r} - \frac{1-p}{1-r}.$$

Setting the derivative to zero gives $p(1-r) = (1-p)r$, hence again $r^\star = p$.

By contrast, a non-proper rule can reward coarse, overconfident reporting. Consider the rule

$$s_Y(r) = \mathbf{1}\{r \geq 1/2\}, \qquad s_N(r) = \mathbf{1}\{r \leq 1/2\},$$

which pays 1 if the realized outcome was assigned probability at least $1/2$ and 0 otherwise. The expected score becomes

$$\mathbb{E}_p[s(r)] = \begin{cases} p, & r > 1/2, \\ 1-p, & r < 1/2, \end{cases}$$

(with any tie convention at $r = 1/2$). If $p > 1/2$, then every report $r > 1/2$ yields the same expected score p, so the rule fails to elicit the correct probability magnitude and only elicits which side of $1/2$ the forecaster is on.

This calculation makes the incentive distinction concrete. Under a proper scoring rule, the report r is uniquely pinned down by first-order optimality at the true belief p. Under a non-proper rule, expected payoff can be flat over wide regions of reports, so the mechanism does not discipline probability reports toward truthfulness.

References

[1] Aysu Ahmadova, Taghi Guliyev, and Khatai Aliyev. "The Relationship between Bitcoin and Nasdaq, U.S. Dollar Index and Commodities". In: *International Journal of Energy Economics and Policy* 14.1 (2024), pp. 281–289. DOI: 10.32479/ijeep.14996.

[2] Andrew Bailey. "Digital Value". In: *P&D* 1.1 (2024), pp. 27–39.

[3] Andrew M. Bailey, Bradley Rettler, and Craig Warmke. "Philosophy, Politics, and Economics of Cryptocurrency I: Money without State". In: *Philosophy Compass* 16.11 (2021), e12785. DOI: 10.1111/phc3.12785.

[4] Andrew M. Bailey, Bradley Rettler, and Craig Warmke. "Philosophy, Politics, and Economics of Cryptocurrency II: The Moral Landscape of Monetary Design". In: *Philosophy Compass* 16.11 (2021), e12784. DOI: 10.1111/phc3.12784.

[5] Andrew M. Bailey, Bradley Rettler, and Craig Warmke. *Resistance Money: A Philosophical Case for Bitcoin*. Abingdon: Routledge, 2024.

[6] Andrew M. Bailey and Craig Warmke. "Bitcoin Is King". In: *Cryptocurrency: Concepts, Technology, and Issues*. Ed. by Jay Liebowitz. Boca Raton: CRC Press, 2023, pp. 175–197.

[7] William Benter. "Computer Based Horse Race Handicapping and Wagering Systems: A Report". In: *Efficiency of Racetrack Betting Markets*. 1994.

[8] Glenn W. Brier. "Verification of Forecasts Expressed in Terms of Probability". In: *Monthly Weather Review* 78 (1950), pp. 1–3. URL: https://api.semanticscholar.org/CorpusID:122906757.

[9] Harold Christopher Burger. "Bitcoin's Natural Long-Term Power-Law Corridor of Growth". In: *Medium* (Sept. 2019). URL: https://medium.com/quantodian-publications/bitcoins-natural-long-term-power-law-corridor-of-growth-649d0e9b3c94.

[10] Ángeles Cebrián-Hernández and Enrique Jiménez-Rodríguez. "Modeling of the Bitcoin Volatility through Key Financial Environment Variables: An Application of Conditional Correlation MGARCH Models". In: *Mathematics* 9.3 (2021), p. 267. DOI: 10.3390/math9030267.

[11] Yiling Chen and David M. Pennock. "Designing Markets for Prediction". In: *AI Magazine* 31.4 (2010), pp. 42–52. DOI: 10.1609/aimag.v31i4.2313. URL: https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/2313/2179.

[12] Yiling Chen and Jennifer Wortman Vaughan. "A New Understanding of Prediction Markets Via No-Regret Learning". In: *Proceedings of the 11th ACM Conference on Electronic Commerce (EC '10)*. New York, NY, USA: Association for Computing Machinery, 2010, pp. 189–198. DOI: 10.1145/1807342.1807372. URL: https://arxiv.org/abs/1003.0034.

[13] Nicola Dimitri. "Bitcoin Mining as a Contest". In: *Ledger* (2017).

[14] Mehmet Levent Erdas and Abdullah Emre Caglar. "Analysis of the relationships between Bitcoin and exchange rate, commodities and global indexes by asymmetric causality test". In: *Eastern Journal of European Studies* 9.2 (2018), pp. 27–45.

[15] Ittay Eyal. "The Miner's Dilemma". In: 2015.

[16] Robin Hanson. *Logarithmic Market Scoring Rules for Modular Combinatorial Information Aggregation*. Working paper. George Mason University, Department of Economics, Jan. 2002. URL: https://mason.gmu.edu/~rhanson/mktscore.pdf.

[17] Benjamin P. Jacot and Paul V. Mochkovitch. "Kelly criterion and fractional Kelly strategy for non-mutually exclusive bets". In: *Journal of Quantitative Analysis in Sports* 19.1 (2023), pp. 37–42. DOI: 10.1515/jqas-2020-0122.

[18] Aron Laszka, Benjamin Johnson, and Jens Grossklags. "When Bitcoin Mining Pools Run Dry: A Game-Theoretic Analysis of the Long-Term Impact of Attacks Between Mining Pools". In: 2015.

[19] Nikos Leonardos, Stefanos Leonardos, and Georgios Piliouras. "Oceanic Games: Centralization Risks and Incentives in Blockchain Mining". In: (2021).

[20] Zhaoyan Liu, Min Shu, and Wei Zhu. "Contrastive Learning Framework for Bitcoin Crash Prediction". In: *Stats* 7 (2024), pp. 402–433. DOI: 10.3390/stats7020025.

[21] Harry Markowitz. "Portfolio Selection". In: *The Journal of Finance* 7.1 (1952), pp. 77–91.

[22] Saralees Nadarajah et al. "Ensemble Learning and an Adaptive Neuro-Fuzzy Inference System for Cryptocurrency Volatility Forecasting". In: *Journal of Risk and Financial Management* 18.2 (2025), p. 52. DOI: 10.3390/jrfm18020052.

[23] Satoshi Nakamoto. "Bitcoin: A peer-to-peer electronic cash system". In: (2008). Available at: https://bitcoin.org/bitcoin.pdf. URL: https://bitcoin.org/bitcoin.pdf.

[24] Abraham Othman et al. "A Practical Liquidity-Sensitive Automated Market Maker". In: *Proceedings of the 11th ACM Conference on Electronic Commerce (EC '10)*. New York, NY, USA: ACM, 2010, pp. 377–386. ISBN: 978-1-60558-822-3. DOI: 10.1145/1807342.1807402.

[25] James Pierog. *Hedging Hashrate Against Block Reward Variance*. https://www.bitcoinprediction.info/documents/Hedging_Block_Reward_Variance_with_DLCs.pdf. Accessed: 2025-12-23. 2025.

[26] Porkopolis.io. *The Chart - Bitcoin Power Law Analysis*. Accessed: December 30, 2025. 2024. URL: https://www.porkopolis.io/thechart/.

[27] Matteo Romiti et al. "A Deep Dive into Bitcoin Mining Pools: An Empirical Analysis of Mining Shares". In: (2019).

[28] Meni Rosenfeld. *Analysis of Bitcoin Pooled Mining Reward Systems*. 2011. eprint: arXiv:1112.4980.

[29] Tim Roughgarden and Clara Shikhelman. "Ignore the Extra Zeroes: Variance-Optimal Mining Pools". In: (2021).

[30] Giovanni Santostasi. "The Bitcoin Power Law Theory". In: *Medium* (Apr. 2017). URL: https://giovannisantostasi.medium.com/the-bitcoin-power-law-theory-962dfaf99ee9.

[31] Lev Ushakov. "Bitcoin Power Law Theory — Executive Summary". In: *Medium* (Nov. 2024). URL: https://medium.com/@fulgur.ventures/bitcoin-power-law-theory-executive-summary-report-837e6f00347e.

[32] Yevhen Zolotavkin, Julian García, and Carsten Rudolph. "Incentive Compatibility of Pay Per Last N Shares in Bitcoin Mining Pools". In: 2017.