Interval Forecasting of Cryptocurrency Returns using Quantile Regression Forests: An Application to the Solana Ecosystem

Interval Forecasting of Cryptocurrency Returns using Quantile Regression Forests: An Application to the Solana Ecosystem

James Lewis

Table of contents

1 Abstract

Interval Forecasting of Cryptocurrency Returns using Quantile Regression Forests: An Application to the Solana Ecosystem

Abstract. (200–300 words placeholder) Problem, data (12-h bars; 72-h target), models (QRF, LQR, LGBM), rolling CV (120/24/6), metrics (pinball, coverage, width), key results + trading relevance.

2 Introduction

2.0.1 The Economic Challenge of Forecasting in Emerging Ecosystems

Extreme volatility and non-normal return distributions are defining characteristics of cryptocurrency markets, rendering traditional point forecasts insufficient for robust risk management (**?**). This challenge is particularly acute for mid-capitalisation tokens within emerging ecosystems like Solana. Unlike large-cap assets, these tokens are subject to rapid narrative shifts and idiosyncratic on-chain dynamics; yet unlike micro-caps, they are liquid enough to attract significant capital. For participants in this space, standard risk models often fail during periods of high network activity or ecosystem-wide events, creating a clear demand for forecasting tools that can dynamically price tail risk.

This dissertation argues that the primary objective must shift from point prediction to **interval forecasting**. By generating calibrated prediction intervals through conditional quantiles, we can capture the asymmetry and tail risk inherent in these volatile assets. A well-calibrated lower quantile serves as a dynamic, forward-looking analogue to Value-at-Risk (VaR) (**?**), while the upper quantile informs on potential upside, providing a comprehensive basis for sophisticated risk management and tactical decision-making.

2.0.2 A Principled Approach: Quantile Regression

To construct reliable prediction intervals, we adopt the framework of **quantile regression** (**?**). This approach is formally grounded in the **pinball loss function**, a proper scoring rule that is uniquely minimised when the forecast matches the true conditional quantile of the distribution. For a target outcome y and a forecast for the τ-th quantile, $\hat{q}_\tau$, the loss is:

$$L_\tau(y, \hat{q}_\tau) = (\tau - \mathbf{1}\{y < \hat{q}_\tau\})(y - \hat{q}_\tau)$$

This allows models to be evaluated on two critical properties: **calibration** (does an 80% interval contain the outcome 80% of the time?) and **sharpness** (are the intervals as narrow as possible while maintaining calibration?) (**?**). These properties are paramount in crypto markets, where accurately quantifying both risk and opportunity is the basis of effective strategy.

2.0.3 The State of the Art and the Research Gap

The literature on cryptocurrency forecasting has largely bifurcated. One branch applies econometric models like GARCH, which excel at modelling volatility but are constrained by parametric assumptions (**?**). The other employs machine learning, but predominantly for point forecasting of price or direction (**?**). While some studies have applied quantile regression to major crypto-assets (**?**), they have typically relied on simpler linear models or have not fully leveraged the rich feature set available from on-chain data. Furthermore, the critical step of post-hoc calibration to ensure nominal coverage guarantees, especially for non-parametric methods, is often overlooked.

This dissertation is designed to address these gaps by making three core contributions: it shifts the focus from point prediction to distributional accuracy; it applies a sophisticated non-parametric methodology to the under-researched domain of mid-cap altcoins; and it integrates a rich, multi-domain feature set that explicitly includes on-chain activity.

2.0.4 1.4 Problem Formulation: The Solana Ecosystem

This study focuses on forecasting **72-hour log-returns** for a universe of mid-cap tokens within the Solana ecosystem, using data aggregated in 12-hour intervals. The asset universe comprises tokens with a market capitalisation exceeding \$30 million, ensuring a focus on liquid yet idiosyncratic assets. The forecasting model is built upon a feature set designed to capture the multi-faceted drivers of returns, spanning five domains: (i) **momentum**, (ii) **volatility**, (iii) **market microstructure**, (iv) **on-chain activity**, and (v) **cross-asset context**.

2.0.5 Methodology and Contributions

The central hypothesis is that the non-linear, interaction-heavy nature of this market demands a non-parametric approach. We propose an adapted **Quantile Regression Forests (QRF)** model (**?**). QRF was selected as the primary model for several reasons. Firstly, its ensemble nature provides inherent robustness to the noisy predictors common in high-dimensional financial feature sets. Secondly, unlike gradient boosting, QRF's independent tree construction can be less prone to overfitting in non-stationary environments. Finally, its method of estimating quantiles from the full distribution of training samples in terminal nodes is a more direct and empirically stable approach than methods requiring separate models for each quantile. To tailor the model for financial time series, we incorporate several critical enhancements: **time-decay weighting** to prioritise recent data, **volatility regime offsets** to adapt to changing market conditions, and **isotonic regression** to enforce the theoretical non-crossing of quantiles.

We benchmark our adapted QRF against a parametric **Linear Quantile Regression (LQR)** and a powerful **LightGBM** model (**?**) augmented with **conformal prediction** (**?**). Preliminary results suggest that the primary advantage of the adapted QRF framework lies in its superior ability to synthesise on-chain activity and market microstructure features to anticipate shifts in return distribution skewness—a dynamic that linear models fail to capture.

2.0.6 Scope and Delimitations

This dissertation provides a rigorous methodological and empirical analysis of interval forecasting. It does not aim to develop a complete, production-ready trading system, which would require further considerations such as transaction costs, liquidity constraints, and execution latency. Furthermore, the feature set, while comprehensive, is confined to publicly available market and on-chain data, thereby excluding alternative data sources such as social media sentiment or developer activity metrics, which may also contain predictive information. The findings are specific to the mid-cap tokens within the Solana ecosystem during the observation period and may not be directly generalisable to other blockchains, market-cap tiers, or market regimes without further investigation and potential recalibration.

2.0.7 Research Question

This framework motivates the central research question of this dissertation:

Can an adapted Quantile Regression Forest model deliver sharper and better-calibrated prediction intervals for 72-hour returns of mid-cap Solana tokens compared to standard linear and gradient-boosted quantile regression baselines?

This overarching question is decomposed into four specific, testable hypotheses:

1. Superior Accuracy: The proposed QRF model achieves a lower mean pinball loss across the quantile spectrum than both LQR and LightGBM with conformal prediction.
2. Superior Calibration: The QRF model's empirical coverage rates for 80% and 90% intervals are closer to their nominal levels.
3. Superior Sharpness: The QRF model produces narrower prediction intervals than the conformally-adjusted LightGBM model, without sacrificing calibration.
4. Statistical Significance: The performance improvements offered by QRF are statistically significant as determined by formal tests on pinball loss differentials.

2.0.8 Dissertation Outline

The remainder of this dissertation unfolds as follows. Chapter 2 establishes the theoretical context by reviewing the relevant literature. Chapter 3 details the data pipeline and feature engineering process, while Chapter 4 outlines the core methodology. The empirical analysis begins in Chapter 5 with the main comparative results, which are translated into a practical trading application in Chapter 6 and stress-tested for robustness in Chapter 7. Finally, Chapter 8 discusses the broader implications, leading to the conclusion in Chapter 9, which summarises the contributions and suggests avenues for future research.

3 Literature Review

This chapter provides a critical review of the literature that justifies the methodology of this dissertation. It first establishes the unique statistical properties of cryptocurrency returns that necessitate specialised forecasting approaches. The review then evaluates the relative merits of parametric and non-parametric quantile estimation models, before examining the essential frameworks for robust forecast evaluation, calibration, and comparison. The chapter synthesises these distinct strands of literature to build a coherent argument for the selection of an adapted Quantile Regression Forest as the core model, and for the specific methodological refinements required for its application.

3.0.1 The Challenge: Statistical Properties of Cryptocurrency Returns

The return distributions of cryptocurrencies are characterised by heavy tails, significant skew, and extreme kurtosis relative to traditional assets, reflecting the frequency of large, abrupt price movements (**?**). This leptokurtosis is compounded by pronounced volatility clustering—periods of relative calm followed by explosive variability—a dynamic exacerbated by the market's continuous operation and fragmented liquidity, which can amplify shocks across uncoordinated venues.

Crucially, this extreme risk is also largely idiosyncratic to the crypto market. Major cryptocurrencies carry substantial tail risk that is not strongly correlated with traditional stock market indices; instead, extreme events are driven by crypto-specific factors such as investor sentiment, regulatory news, or network-level events (**?**). Furthermore, their returns show little to no exposure to standard macroeconomic risk factors, being influenced instead by internal drivers like network momentum and adoption metrics (**?**). This body of evidence demonstrates that classical financial risk models, with their reliance on Gaussian assumptions and traditional risk factors, are fundamentally misspecified for crypto assets. A credible forecasting framework must therefore abandon these assumptions and be built to incorporate the crypto-native features that drive risk.

3.0.2 Approaches to Quantile Estimation

Given the non-normal character of crypto returns established previously, estimating the full conditional distribution is more informative than forecasting its central tendency. Quantile regression provides a natural framework for this, but the choice between a restrictive parametric model and a flexible non-parametric one is critical.

3.0.2.1 The Parametric Benchmark: Linear Quantile Regression

Quantile regression, introduced by (**?**), generalises linear regression by estimating conditional quantiles directly. For a given quantile level $\tau \in (0, 1)$, the linear quantile regression (LQR) estimator solves:

$$\hat{\beta}_\tau = \arg\min_\beta \sum_{i=1}^{n} \rho_\tau(y_i - x_i^\top \beta)$$

where $\rho_\tau(u) = u(\tau - \mathbb{1}\{u < 0\})$ is the **pinball loss function**. While LQR provides a transparent and interpretable benchmark, its fundamental assumption of a fixed linear relationship across all quantiles represents a severe limitation. This rigidity is fundamentally at odds with the non-linear volatility dynamics and abrupt regime shifts that define cryptocurrency markets. Furthermore, the common practical issue of **quantile crossing**—where independently estimated quantile lines intersect—can yield incoherent and unusable interval forecasts unless post-hoc remedies like rearrangement are applied (**?**). These shortcomings do not merely motivate, but necessitate the exploration of more flexible, non-parametric methods.

3.0.2.2 Non-Parametric Solutions: Quantile Regression Forests

As a direct response to the limitations of linear models, Quantile Regression Forests (QRF), proposed by (**?**), extend the Random Forest algorithm (**?**) to estimate the entire conditional distribution. Instead of averaging outcomes in terminal nodes, QRF uses the full empirical distribution of training responses within the leaves to form a predictive distribution, from which conditional quantiles are derived.

This non-parametric approach is inherently well-suited to financial data; it naturally captures complex non-linearities and adapts to heteroskedasticity without pre-specification. However, QRF is not without its own challenges. Its theoretical foundation rests on an assumption of independent and identically distributed (i.i.d.) data—a condition clearly violated by financial time series. A naive application of QRF to time-ordered data can therefore lead to biased estimates. This violation is a central methodological challenge that requires specific adaptations, such as the time-decay weighting and rolling validation schemes discussed later, to apply the model soundly. Furthermore, the accuracy of its tail quantile estimates can degrade if the terminal leaves are sparsely populated, a genuine risk when modelling extreme events. Boosting offers another route to non-parametric quantile estimation, but with contrasting properties.

3.0.2.3 A Boosting Alternative: LightGBM for Quantiles

Gradient boosting presents another powerful non-parametric paradigm. It constructs an ensemble sequentially, with each new tree trained to correct the errors—specifically, the gradients of the loss function—of the preceding models (**?**). This methodology can be directly applied to quantile regression by using the pinball loss as the objective. LightGBM (**?**) is a highly efficient and scalable implementation of this idea, making it a formidable baseline.

In sharp contrast to QRF's parallelised construction, boosting's sequential focus on difficult-to-predict instances can yield sharper estimates in the tails. This potential for higher accuracy, however, comes with significant trade-offs. A separate model must typically be trained for each target quantile, imposing a considerable computational burden. The aggressive, error-focused fitting can also produce "ragged" and unstable quantile estimates in regions with sparse data and may overfit transient noise without careful regularisation. Finally, like LQR, independently fitted boosting models are susceptible to the problem of quantile crossing.

3.0.3 Ensuring Rigour: Calibration, Evaluation, and Comparison

Selecting a flexible forecasting model is insufficient on its own; its predictive performance must be evaluated using principled metrics, its outputs calibrated to ensure reliability, and its superiority over alternatives established through formal statistical tests.

3.0.3.1 Proper Scoring and Forecast Evaluation

A principled evaluation of probabilistic forecasts requires the use of **strictly proper scoring rules**, which incentivise the model to report its true belief about the future distribution. For quantile forecasts, the canonical proper scoring rule is the pinball loss (**?**). As the metric being directly optimised by the models, it serves as the primary tool for evaluation. However, performance is not a single dimension. The quality of an interval forecast is judged by two distinct and often competing properties: **calibration**, the statistical consistency between the nominal coverage rate (e.g., 90%) and the empirical frequency of outcomes falling within the interval; and **sharpness**, the narrowness of the interval. An ideal forecast is one that is maximally sharp, subject to being well-calibrated. However, a model optimised on a proper score is not inherently guaranteed to be well-calibrated in finite samples. This gap between theoretical optimisation and empirical reliability motivates the use of formal calibration techniques.

3.0.3.2 Achieving "Honest" Intervals: Conformal Prediction

Conformal prediction provides a distribution-free framework to correct for such miscalibration. Specifically, Conformalized Quantile Regression (CQR) (**?**) provides a mechanism to adjust a base model's quantile forecasts to achieve guaranteed marginal coverage. It uses a hold-out calibration set to compute a conformity score based on model errors, which is then used to adjust the width of future prediction intervals. While the underlying exchangeability assumption is violated in time series, employing a rolling calibration window of recent data provides a practical and widely used compromise to adapt the procedure to non-stationary environments.

3.0.3.3 Statistically Significant Comparisons: The Diebold-Mariano Test

To move beyond descriptive comparisons of average loss, formal statistical tests are required to determine if the performance difference between two models is significant. The **Diebold-Mariano (DM) test** (**?**) provides a standard framework for this, assessing the null hypothesis of equal predictive accuracy. The test statistic is given by:

$$DM = \frac{\bar{d}}{\sqrt{\hat{\text{Var}}(\bar{d})}}$$

where $\bar{d}$ is the mean loss differential between two models. For the multi-step, overlapping forecasts used in this project, the sequence of loss differentials will be autocorrelated by construction. It is therefore critical to use a heteroskedasticity and autocorrelation consistent (HAC) variance estimator, as recommended by (**?**), to ensure valid statistical inference.

3.0.4 Methodological Requirements for Robust Time-Series Forecasting

The foundational literature establishes the potential of non-parametric models, but their successful application to volatile, non-stationary financial time series is contingent upon a number of specific methodological adaptations. This section reviews the literature concerning these essential requirements, from ensuring the logical coherence of predictions to adapting models to the temporal dynamics of the data.

3.0.4.1 Ensuring Coherent Predictions: Non-Crossing Quantiles

Models that estimate quantiles independently, such as LQR and standard gradient boosting, are susceptible to the critical failure of **quantile crossing**. This occurs when, for instance, a predicted 90th percentile falls below the predicted 50th percentile, yielding a logically incoherent and unusable conditional distribution. To rectify this, (**?**) proposed a post-processing "rearrangement" technique. This method applies isotonic regression to the initially estimated quantile function, projecting the unconstrained predictions onto the space of valid, non-decreasing distribution functions. This ensures the monotonicity of the quantile curve and is a critical step for producing valid prediction intervals.

3.0.4.2 Adapting to Non-Stationarity and Temporal Dependence

Financial time series are fundamentally non-stationary and autocorrelated, violating the i.i.d. assumption that underpins many machine learning models. Two distinct but related adaptations are required to address this.

First, to handle **non-stationarity** such as volatility clustering, the model must prioritise more recent information. The literature supports the use of **time-decay sample weights** to achieve this. (**?**), for example, introduced exponentially weighted quantile regression for Value-at-Risk estimation, demonstrating that up-weighting recent observations yields more responsive and accurate tail forecasts in changing market conditions.

Second, to handle **temporal dependence**, model evaluation and hyperparameter tuning must respect the chronological order of the data. Standard k-fold cross-validation is invalid for time series, as it can lead to lookahead bias and produce misleadingly optimistic performance estimates. The literature therefore strongly advocates for rolling-origin or blocked cross-validation schemes, which preserve the temporal sequence by training only on past data to forecast the future, thereby simulating a live forecasting environment (**?**).

3.0.4.3 Correcting for Bias and Ensuring Empirical Calibration

Even correctly specified quantile models can exhibit systematic biases in finite samples. As (**?**) have shown, linear quantile regression can suffer from a theoretical **under-coverage bias**, where a nominal 90% interval may contain the true outcome significantly less than 90% of the time due to estimation error. This problem motivates the necessity of post-hoc calibration.

While the CQR framework discussed previously is one such solution, the literature offers several alternatives. Methods like the Jackknife+ (**?**) and residual bootstraps provide different mechanisms for constructing prediction intervals with more reliable coverage properties. The existence of this rich literature on calibration highlights a crucial principle for risk management applications: a model's raw output cannot be taken at face value. An explicit calibration step is required to correct for inherent biases and ensure the resulting prediction intervals are empirically “honest”.

3.0.5 Integrating Crypto-Native Data Sources

The literature on cryptocurrency risk factors makes it clear that models confined to historical price data are insufficient. The unique nature of blockchain-based assets provides a rich set of crypto-native data sources that are essential for capturing the specific drivers of risk and return in this asset class.

3.0.5.1 Market Microstructure and Liquidity

Like traditional markets, cryptocurrency price dynamics are influenced by liquidity and trading frictions. Empirical studies have documented that periods of market stress coincide with widening bid-ask spreads and evaporating order book depth (**?**). Furthermore, the on-chain nature of transactions introduces unique microstructural features, such as network congestion and transaction fees, which can impact market liquidity and price formation (**?**). Incorporating proxies for these effects is crucial, as it allows a model to dynamically adjust its estimate of uncertainty; for instance, by widening its prediction intervals in response to deteriorating market liquidity, thereby anticipating volatility spikes.

3.0.5.2 On-Chain Activity and Network Fundamentals

Blockchains provide a transparent ledger of network activity, offering powerful proxies for an asset’s fundamental adoption and utility. Metrics such as the growth in active addresses, on-chain transaction counts, and, in the context of decentralised finance (DeFi), the Total Value Locked (TVL) in smart contracts, can signal shifts in investor sentiment and capital flows. Empirical studies consistently find that models augmented with on-chain metrics significantly outperform those based only on historical prices, as this data provides unique information about network health and demand (**?**). These features allow a model to condition its forecasts on the fundamental state of the network, potentially informing not just the location but also the shape of the predictive distribution.

3.0.5.3 Cross-Asset Spillovers and Systemic Risk

The cryptocurrency market is a highly interconnected system where shocks to major assets like Bitcoin and Ethereum often propagate to smaller altcoins. This “connectedness” has been formally measured, showing significant return and, particularly, volatility spillovers from market leaders to the rest of the ecosystem (**?**; **?**). This implies that the risk of an individual token is not purely idiosyncratic; it is also a function of the broader crypto market’s state. Consequently, any forecasting model that treats a token in isolation is fundamentally misspecified and is likely to underestimate systemic risk. A robust framework must therefore account for these cross-asset influences.

3.0.6 Synthesis and Conclusion

This review has established a clear and compelling justification for the methodology adopted in this dissertation. The unique statistical properties of cryptocurrency returns—heavy tails, volatility clustering, and dependence on idiosyncratic, on-chain factors—render traditional parametric models inadequate. This failure necessitates the use of flexible, non-parametric methods, for which Quantile Regression Forests are a logical choice, given their ability to capture complex, non-linear relationships without restrictive distributional assumptions.

However, the literature also makes it clear that a naive application of any such model would be insufficient. A credible forecasting framework requires a series of specific, evidence-based adaptations. The need to adapt to non-stationarity justifies the use of time-decay weighting. The imperative for valid, coherent predictions necessitates post-processing to enforce non-crossing quantiles. The requirement for reliable out-of-sample evaluation mandates the use of rolling cross-validation. Finally, the well-documented tendency for quantile models to mis-calibrate compels the integration of a formal calibration step to ensure the final prediction intervals are empirically valid.

By synthesising these distinct strands of literature—from model selection to time-series adaptation and calibration—this project constructs an integrated and methodologically robust framework. This framework is specifically designed to address the multifaceted challenges of interval forecasting in the volatile and rapidly evolving cryptocurrency market.

4 Data and Features

The validity of any forecasting model is fundamentally constrained by the quality and integrity of its input data. For volatile and rapidly evolving assets like cryptocurrencies, constructing a robust, research-grade dataset is a critical prerequisite for meaningful analysis. This chapter details the multi-stage process undertaken to source, clean, and engineer the data used in this dissertation. It begins by defining the asset universe and data sources, then describes the construction of the target variable and the extensive feature engineering pipeline. Finally, it outlines the preprocessing steps taken to handle missing data and presents key insights from the exploratory data analysis that guided the modelling approach.

4.0.1 Data Sourcing and Asset Universe Definition

The foundation of this study is a bespoke, multi-source panel dataset constructed at a 12-hour resolution, specifically designed to support tail-sensitive, quantile-based forecasting. The data streams were aggregated from several high-quality APIs, using specific endpoints for different data types to ensure the highest fidelity for each signal.

The primary data sources included:

- **Price and Volume Data**: Historical Open-High-Low-Close (OHLC) and volume data for individual tokens were sourced from the **SolanaTracker API**, chosen for its deep liquidity and high-quality, reliable price feeds, which minimises the risk of spurious gaps or errors in the core price series.
- **On-Chain Metrics**: Key on-chain indicators for the Solana ecosystem, such as `holder_count` and `transfer_count`, were retrieved via the **CoinGecko API**, which provides broad coverage of token-specific network activity.
- **Global Context Data**: Broader market signals, including historical prices for Bitcoin (BTC) and Ethereum (ETH), as well as Solana-specific network metrics like transaction counts and Total Value Locked (TVL), were sourced from CoinGecko and the **Google BigQuery Solana Community Public Dataset**.

An initial effort was also made to incorporate social media sentiment data, given the narrative-driven nature of many Solana tokens. While an ingestion pipeline was successfully built, the data availability and quality were ultimately deemed insufficient for rigorous academic analysis and were excluded from the final feature set.

The **asset universe** was carefully defined from an initial, hand-picked list of 23 tokens based on their relevance to the Solana ecosystem. This list was then filtered according to a set of rigorous criteria. To be included in the final universe, a token had to satisfy:

- A **minimum market capitalisation** of $30 million to ensure a baseline of market significance and liquidity.
- A **minimum trading history** of six months. This was a crucial requirement to ensure a stable two-month (approx. 120 12-hour bars) training window was available for the initial backtest period for every asset in the universe.

This filtering was a deliberate and aggressive research design choice. Tokens with excessive missing data, insufficient history, or erratic reporting were pruned from the sample. This step is critical for quantile-based modelling, as tokens with inconsistent or late-starting data histories can inject significant bias into the empirical distribution, particularly distorting the tail estimates that are a primary focus of this research. By favouring data quality over sample size, this process ensures that the subsequent modelling results are attributable to the forecasting methodology itself, rather than being artefacts of poor-quality data.

The initial raw dataset comprised **8,326 rows across 23 tokens**. After applying the filtering criteria and cleaning procedures detailed in the following sections, the final asset universe for this study consists of **20 mid-cap Solana tokens**. The full dataset spans from **5th December 2024 to 3rd June 2025**, comprising a total of **6,464** 12-hour observations after cleaning and alignment.

4.0.2 3.2 Target Variable Construction and Properties

The predictive target for this study is the **72-hour forward-looking logarithmic return**, calculated at each 12-hour time step. It is formally defined as:

$$r_t^{(72h)} = \log(P_{t+6}) - \log(P_t)$$

where P_t is the closing price of the token at the end of the 12-hour bar at time t, and P_{t+6} is the closing price six 12-hour bars later. The use of logarithmic returns is standard practice in financial econometrics, as it provides a continuously compounded return that is time-additive and whose distribution more closely approximates normality than simple returns.

The combination of a **12-hour data cadence** and a **72-hour forecast horizon** was a deliberate design choice to create a target variable suitable for mid-frequency trading strategies. The 12-hour aggregation smooths the extreme noise present in sub-hourly price changes, while the 72-hour horizon is long enough to capture significant, economically meaningful moves where tail events and distributional properties become highly relevant. This choice aims to maximise the signal-to-noise ratio for the specific purpose of forecasting the distribution of multi-day returns.

Exploratory analysis confirms that the resulting target variable exhibits the extreme non-normal characteristics that motivate this research. As shown in **Table 3.1**, the pooled distribution of 72-hour log returns is highly leptokurtic and positively skewed. With a **kurtosis of 20.73**, it demonstrates exceptionally fat tails compared to a normal distribution (kurtosis of 3), indicating that extreme price movements are far more common than a Gaussian model would suggest.

Statistic	Value
Mean	0.0031
Standard Deviation	0.1259
Skewness	**1.68**
Kurtosis	**20.73**
Minimum	-0.75
Maximum	1.02

Table 3.1: Summary Statistics for the 72-hour Log Return Target Variable (Pooled Across All Tokens).

The heavy-tailed nature of the target is further illustrated in **Figure 3.1**, which plots the empirical distribution against a normal distribution with the same mean and variance. The substantially higher peak (leptokurtosis) and elongated tails of the empirical distribution provide clear visual evidence that a Gaussian assumption would be inappropriate and underscore the necessity of a quantile-based modelling approach.

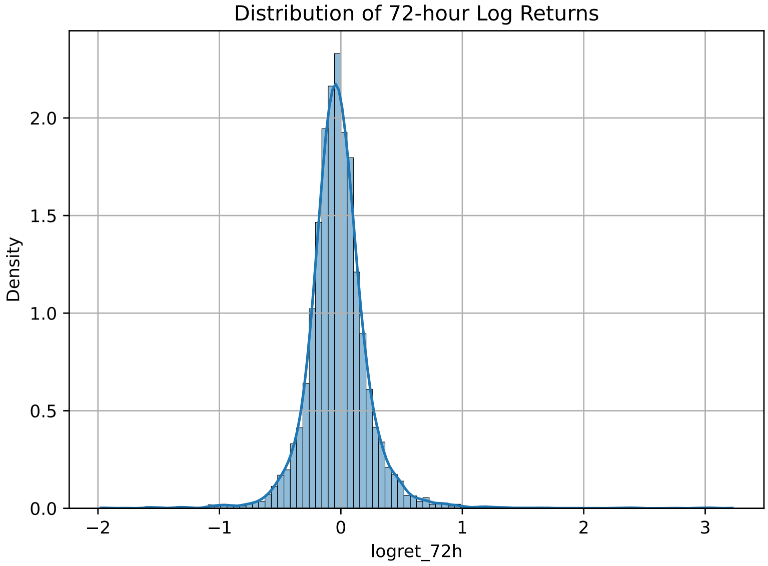


Figure 4.1: Distribution of the 72-hour log return target variable, pooled across all tokens. The distribution exhibits a sharp peak and significantly heavier tails than a comparable normal distribution, justifying the use of quantile-based models.

Furthermore, the properties of this distribution are not static. Analysis of the underlying 12-hour returns reveals that the shape of the distribution is highly conditional on the broader market environment. By defining macro-regimes based on the quartiles of SOL's 12-hour return, it becomes evident that the conditional distribution of token returns shifts systematically. As shown in **Figure 3.2**, "Bull" regimes are associated with positive skew and a fatter right tail, whereas "Bear" regimes exhibit heavier left tails, signalling increased downside risk. This empirical finding of **regime-dependence** is critical, as it justifies the need for a *conditional* forecasting model that can adapt its predicted quantiles based on contextual features.

Finally, it is important to note that this construction results in an **overlapping target variable**. A new 72-hour forecast is generated every 12 hours, meaning that the forecast periods overlap significantly. This has direct implications for the evaluation methodology, as the resulting forecast errors will be serially correlated by construction. As detailed in the literature review and methodology chapters, this requires the use of specific techniques, such as blocked cross-validation and HAC-robust statistical tests, to ensure valid inference.

4.0.3 Data Preprocessing and Cleaning

Before any features could be engineered, the raw, multi-source panel dataset underwent a rigorous cleaning and preprocessing pipeline. This was a critical phase designed to handle the significant data quality challenges inherent in cryptocurrency markets, such as missing observations and inconsistent token histories, ensuring the final dataset was robust and suitable for modelling. The initial raw data contained approximately **18% missing values** in the core OHLCV columns alone, with some on-chain features like `holder_count` missing nearly 30% of their data (see Appendix **Table @tbl:apx-missing-top10** for a full breakdown).

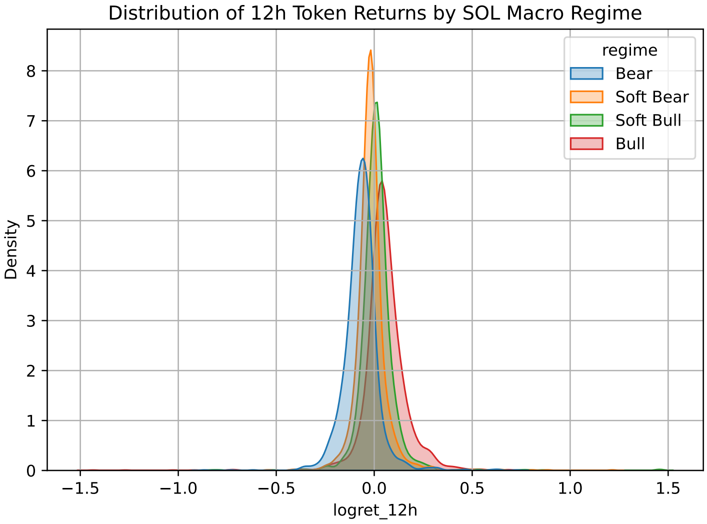


Figure 4.2: Conditional distribution of 12-hour token returns, faceted by the SOL macro-regime. The shape, skew, and tail behaviour of the returns change significantly depending on the broader market context.

The nature of this missingness was not uniform. As illustrated by the heatmap in **Figure 3.4**, the data gaps were highly structured. Some tokens (e.g., `MEW`, `ZEREBRO`) had clean data but only after a late start date, while others exhibited intermittent, patchy gaps throughout their history. This heterogeneity necessitated a multi-step strategy rather than a single, one-size-fits-all approach.

Figure 3.4: *Heatmap of OHLCV data presence across the token universe over time (Green = Present, Red = Missing). The block-like structure for some tokens indicates late listings, while sporadic red patches show intermittent data gaps, motivating a hybrid cleaning strategy.*

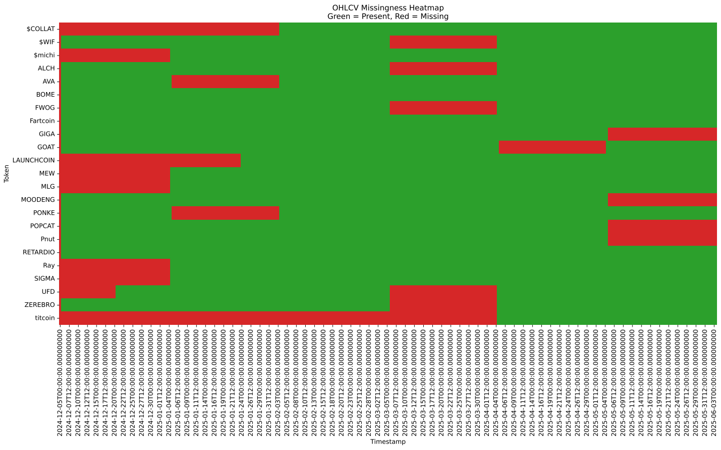


Figure 4.3: Heatmap of OHLCV data presence across the token universe over time (Green = Present, Red = Missing). The block-like structure for some tokens indicates late listings, while sporadic red patches show intermittent data gaps, motivating a hybrid cleaning strategy.

4.0.3.1 Temporal Alignment and Clipping

To ensure temporal consistency, each time series was first clipped to begin only from its first fully valid OHLCV observation. This step, detailed in the strategy summary in Appendix **@sec-cleaning-strategy**, removes spurious data from pre-launch or illiquid initial listing periods, which could otherwise contaminate the analysis. Tokens with insufficient history after this clipping process (e.g., `$COLLAT`) were dropped from the universe entirely.

4.0.3.2 Imputation Strategy

A key challenge was to fill the remaining intermittent gaps without distorting the underlying distributional properties of the data. Several imputation methods were benchmarked on simulated missing data. Counter-intuitively, the analysis revealed that a simple **linear interpolation** outperformed more complex methods like Kalman smoothing in terms of Root Mean Squared Error (RMSE) (see Appendix **@#imp-table** for benchmark results). This finding suggests that for small, sporadic gaps, a simple interpolation preserves the local price trajectory and its inherent noise structure more effectively than methods that impose stronger, potentially smoothing, structural assumptions.

Therefore, the final strategy adopted was a hybrid approach: **linear interpolation** to fill the majority of gaps, supplemented by a **forward-fill** for a maximum of two consecutive 12-hour bars.

4.0.3.3 Imputation-Awareness

Crucially, the imputation process was not treated as a silent correction. For every feature that was imputed, a corresponding binary **imputation mask** variable was created. This flag takes a value of `1` if the original data point at that timestamp was missing and subsequently imputed, and `0` otherwise. This technique of "imputation-awareness" is a key methodological choice, as it allows the machine learning model to learn directly from the patterns of missingness. This is particularly relevant in crypto markets, where data gaps themselves can be a predictive signal (e.g., indicating an exchange API outage or a period of extreme market stress).

4.0.4 Exploratory Data Analysis

Following the preprocessing pipeline, an extensive exploratory data analysis (EDA) was conducted to uncover the key empirical properties of the data. This section presents the three most critical findings that provide a direct, data-driven justification for the subsequent feature engineering choices and the selection of a non-parametric, conditional forecasting model.

4.0.4.1 Volatility Clustering and Asymmetric Leverage

The data exhibits two foundational properties of financial time series that invalidate simple, static risk models. First, strong **volatility clustering** is evident in the autocorrelation of absolute 12-hour returns, confirming that risk is time-varying and motivating the inclusion of dynamic volatility features.

Second, the relationship between returns and subsequent volatility is asymmetric. An analysis regressing 12-hour log returns against forward 36-hour realised volatility reveals a distinct U-shaped pattern, as shown in **Figure 3.5**. This confirms that variance is conditional on the magnitude of recent returns, with large moves in either direction predicting elevated future volatility. The effect is slightly stronger for negative returns, consistent with a “crypto leverage effect” where downside shocks lead to greater market instability. This non-linear dynamic necessitates a modelling approach, such as the Quantile Regression Forest used in this study, that can naturally capture such relationships and adapt its prediction interval widths based on the direction and magnitude of recent price shocks.

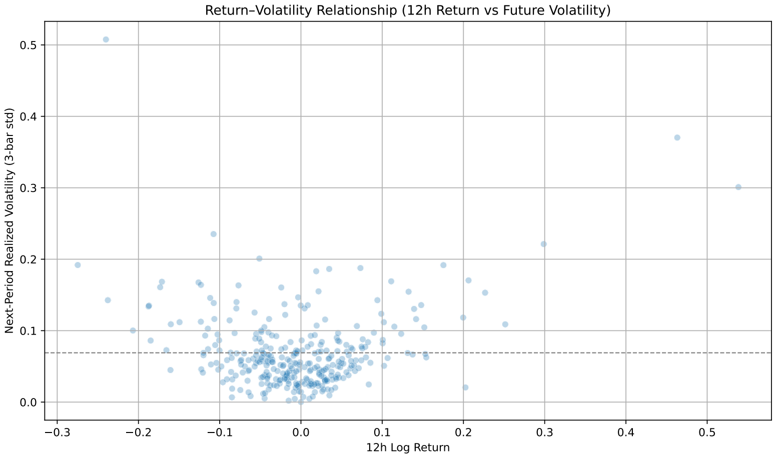


Figure 4.4: A scatter plot of 12h Log Return vs. Next-Period Realised Volatility. The U-shaped pattern, with a slightly steeper slope for negative returns, illustrates the asymmetric leverage effect.

4.0.4.2 Feature Redundancy and Collinearity

An analysis of the correlation structure between 18 core features was conducted to identify potential multicollinearity, which can destabilise tree-based ensemble models. As shown in the Pearson correlation matrix in **Figure 3.6**, several feature pairs exhibit extremely high linear relationships. The most significant correlations were observed between:

- `token_close_usd` and `token_volume_usd` (r
  $approx$0.999)
- `btc_close_usd` and `tvl_usd` (r
  $approx$0.94)
- `sol_close_usd` and `tvl_usd` (r
  $approx$0.89)

This finding is critical. While these raw fields were retained for the initial feature engineering phase to allow for the construction of richer indicators (e.g., from OHLC data), this analysis motivates the necessity of a subsequent feature pruning step. Reducing this high level of collinearity before modelling is essential for improving the stability, training speed, and interpretability of the final Quantile Regression Forest.

4.0.4.3 The Empirical Failure of Gaussian Assumptions

To provide a definitive, data-driven justification for model selection, a baseline experiment was conducted to compare a naive Gaussian interval forecast (defined as
pmz
$cdot$

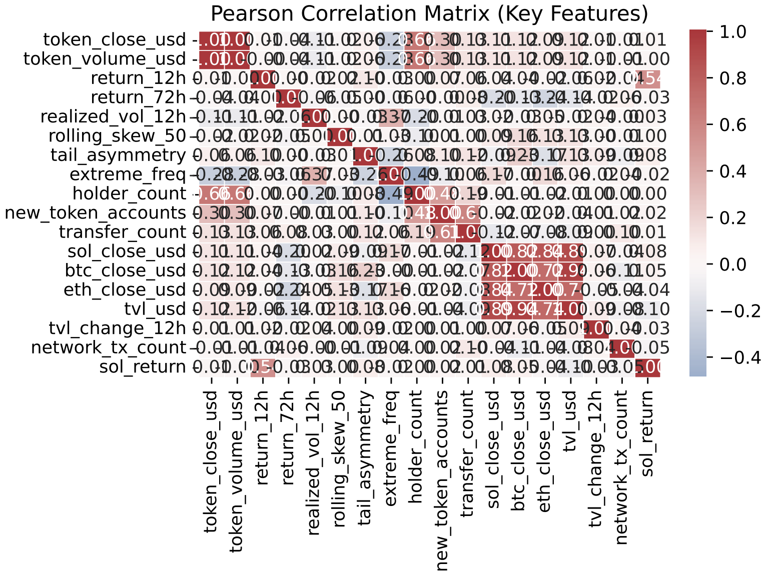


Figure 4.5: Pearson correlation matrix of key features. The strong red blocks highlight pairs of highly correlated variables, necessitating a feature pruning or aggregation strategy.

sigma, using realised volatility) against a simple Quantile Regression Forest. The results, shown in **Figure 3.7**, are stark.

The naive Gaussian intervals systematically **under-cover** the true outcomes across all nominal levels; for example, a nominal 80% interval achieves only ~70% empirical coverage. In contrast, even a basic QRF model tracks the ideal 45-degree line far more closely, demonstrating superior **calibration** by adapting to the true fat-tailed and skewed nature of the returns.

Crucially, this improvement in calibration does not come at the cost of precision. At a nominal 80% coverage level, the QRF intervals were also significantly **sharper**, with an average width of 0.1682 compared to 0.2038 for the naive method. This dual failure of the Gaussian approach—in both calibration and sharpness—provides the ultimate empirical justification for rejecting simple parametric assumptions and adopting a non-parametric methodology like QRF for this dataset.

4.0.5 3.5 Feature Engineering and Selection

Following the data preparation and exploratory analysis, an extensive feature set was engineered. This process was guided by two main principles: first, all predictors must be strictly causal, using only information available at or before time t; second, the feature set should be designed to capture the specific statistical properties—such as volatility clustering, asymmetry, and regime-dependence—identified in the EDA.

4.0.5.1 3.5.1 Feature Construction by Family

The engineered set focuses on signals that respond to the unique dynamics of cryptocurrency markets. The literature supports using a diverse set of technical and on-chain indicators, as machine learning models can effectively synthesise these signals to improve predictive accuracy `[@Akyildirim2021]`. The constructed features fall into five families:

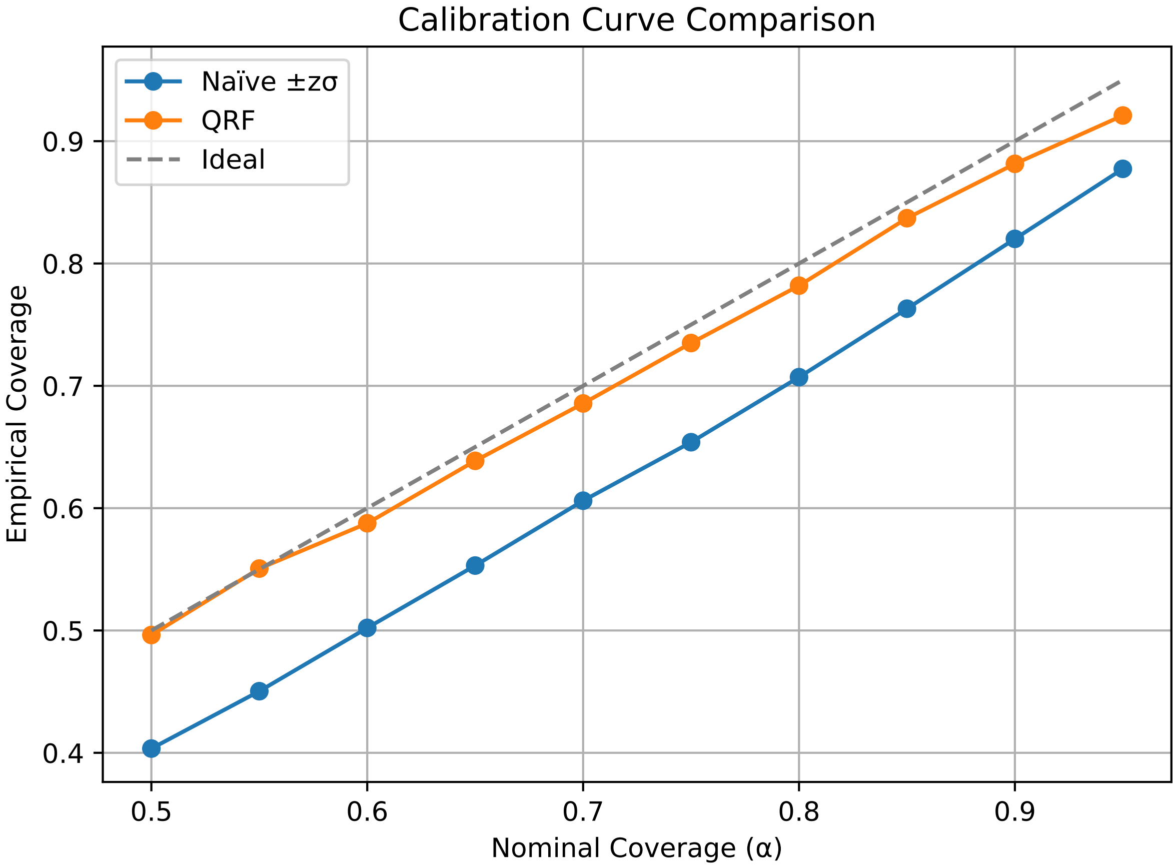


Figure 4.6: Calibration curve comparing the empirical vs. nominal coverage of naive Gaussian intervals and QRF intervals. The QRF's proximity to the ideal line demonstrates its superior ability to model the data's true distribution.

1. **Momentum & Trend:** Standard indicators such as 12h and 36h log-returns, the 14-period Relative Strength Index (RSI), and MACD were created to capture trend and mean-reversion dynamics.
2. **Volatility & Tails:** To model the observed volatility clustering and leverage effects, features such as realised volatility over 3 and 6 bars, downside-only volatility, and the Average True Range (ATR) were included. Higher-moment estimators like rolling 36-hour skewness were also engineered to capture tail asymmetry.
3. **Liquidity & Flow:** The Amihud illiquidity measure and volume z-scores were constructed to provide the model with signals about market depth and trading frictions, which are conditions under which prediction intervals should widen.
4. **On-Chain Activity:** To capture fundamental network health, features such as the growth rate of unique wallets and the ratio of new accounts to total holders were included, consistent with the on-chain data availability constraints.
5. **Market Context:** To model the cross-asset spillovers identified in the literature, features such as the 12h returns for SOL, BTC, and ETH, as well as the rolling 36h correlation of each token to these majors, were created. This explicitly allows the model to learn a dynamic, implicit beta to the broader market.

A complete feature dictionary, including formulas and window lengths, is provided in Appendix [Appendix ID].

4.0.5.2 3.5.2 Redundancy Control and Pruning

The initial engineering process generated over 90 candidate features. To create a final feature set that was both predictive and robust, a systematic, three-stage pruning pipeline was implemented:

1. **Initial Filtering:** Features with near-zero variance or excessive missingness (>80%) were removed.
2. **Collinearity Filter:** To improve model stability, one feature from any pair with a Pearson correlation coefficient $|\rho| > 0.98$ was removed.
3. **Gain-Based Pruning:** A computationally inexpensive LightGBM model was trained to predict the median ($\tau = 0.50$) return. Any feature contributing less than 0.3% to the total gain was pruned. This resulted in a core set of **29 predictors** that explained **99.3%** of the model's total gain.

The resulting feature set, designated `features_v1`, was frozen for all subsequent median-based modelling. A second set, `features_v1_tail`, was created by reintroducing several theoretically important but low-gain tail-risk indicators (e.g., `extreme_count_72h`) for use in the final quantile models. This structured process ensures the models are built upon a rich, yet parsimonious, set of predictors.

5 Methods

This section formalises the modelling and evaluation frame used throughout the dissertation. It defines the target, records leakage controls, and specifies the rolling train–calibrate–test design under which all models are compared. **Notation and symbols follow Appendix @ref(app-m0-notation).**

Objective and target. For each token i at decision time t (12-hour cadence), we forecast conditional quantiles of the **72-hour** log return

$$y_{i,t+6} = \log C_{i,t+6} - \log C_{i,t},$$

where $C_{i,t}$ denotes the 12-hour close. At quantile level $\tau \in (0,1)$, we learn predictors $\hat{q}_\tau(x)$ using the **pinball loss** (see Appendix M1; Koenker & Bassett, 1978; Koenker, 2005).

Panel scope and pooling. Models are trained **per token** (no cross-sectional pooling). Within each rolling window, token i's model uses only its own history. The design matrix is the pruned **feature-set v1 (29 predictors)** defined in §3, including categorical dummies where applicable.

Leakage controls and preprocessing. All predictors are strictly backward-looking and aligned to time t; rows without full look-back are dropped. Preprocessing is **model-specific**: — *LQR:* numeric features standardised (fit on Train only), categoricals one-hot. — *LightGBM / QRF:* numerics unscaled; categoricals passed natively / one-hot as supported. No further within-pipeline winsorisation or transformations are applied beyond the upstream log-return construction.

Rolling evaluation protocol. For each token we use a blocked walk-forward design:

$$\text{Train} = 120 \text{ bars } (\approx 60 \text{ d}) \;\Rightarrow\; \text{Calibration} = 24 \text{ bars } \;\Rightarrow\; \text{Test} = 6 \text{ bars } (72 \text{ h}),$$

advanced by **step = 6 bars** so that tests are non-overlapping. Hyperparameters are tuned once in a global study and then **fixed**; LightGBM additionally uses early stopping on the calibration slice. *PLACEHOLDER — Figure:* `(@fig-rolling)` Rolling scheme diagram (Train→Cal→Test, step = 6).

Calibration and non-crossing. Base quantiles are made **monotone in** τ via row-wise **isotonic rearrangement** (pool-adjacent-violators), then **central bands** are adjusted using **split-conformal prediction** (citations as above). Formal score definitions and the order-statistic inflation used for 80 % and 90 % bands are given in **Appendix M2** (see also listings `A-isotonic.py` and `A-conformal.py`).

Models and libraries. — **LQR:** linear quantile regression via `statsmodels.QuantReg`. — **LightGBM:** gradient-boosted trees with quantile objective per τ. — **QRF:** Quantile Regression Forests; one shared forest predicts all τ. Full software and environment details are reported in **Appendix R1** (`@tbl-software`).

Evaluation and tests (overview). We report **pinball loss** by τ, **empirical coverage** and **average width** for 80 %/90 % bands, and **quantile reliability** curves. Pairwise comparisons use **Diebold–Mariano** tests on loss differentials with **Newey–West** HAC and **Harvey–Leybourne–Newbold** small-sample correction; formulas and code are provided in **Appendix M4** and listing `A-DM-HAC.py`.

5.1 4.2 Linear Quantile Regression (Baseline) Model

For each $\tau \in \{0.05, 0.10, 0.25, 0.50, 0.75, 0.90, 0.95\}$ and token i, we fit an independent **Linear Quantile Regression** model that minimises the check (pinball) loss to estimate $\hat{q}_\tau(x)$ (Koenker & Bassett, 1978; Koenker, 2005). The formal program is given in **Appendix (?)(app-l1-lqr)**.

Design matrix and preprocessing. We use the pruned **feature-set v1 (29 predictors)** (§3). Numerical columns are **standardised on the Train slice only**; categorical indicators (e.g., day-of-week) are **one-hot** with an intercept. No additional within-pipeline transforms are applied.

Estimation. Models are fit with `statsmodels.QuantReg`; implementation notes (solver and tolerances) appear in **Appendix (?)(app-r2-lqr-impl)**. There are no tree/boost hyperparameters; effective complexity is governed by the design matrix. Convergence was stable across rolling slices.

Non-crossing and calibration. Because per-τ fits can cross, we enforce **monotonicity in τ** via row-wise **isotonic rearrangement**; details in **Appendix (?)(app-m1-isotonic)**. Central prediction bands are then **split-conformally** adjusted; score definitions and order-statistic inflation are in **Appendix (?)(app-m2-conformal)**.

Rolling protocol. Evaluation follows the **Train 120 → Cal 24 → Test 6** scheme with **step = 6** (non-overlapping Test windows) defined in §4.1.

Reporting. We present **pinball loss** by τ, **calibration curves** and **coverage/width** at 80 %/90 %, and **Diebold–Mariano** tests vs. LightGBM and QRF (test specification in §4.5 and **Appendix (?)(app-m4-tests)**).

Figures and appendix cross-references. *Main text:* compact **LQR calibration** panel and a small **pinball-by-τ** table. *PLACEHOLDER — Figure:* `(@fig-lqr-calib)` LQR calibration curve (Test). *PLACEHOLDER — Table:* `(@tbl-lqr-pinball)` Pinball loss by quantile (Test). *Appendix:* per-token **fan charts** and **coverage-by-token** bars; code listings. *PLACEHOLDER — Listings:* `A-LQR-fit.py` (fit & preprocessing), `A-conformal.py` (bands).

Why this baseline. LQR provides a transparent **parametric** benchmark. In heavy-tailed, skewed crypto returns, its characteristic **tail mis-calibration** (cf. `@fig-lqr-calib`) motivates the flexible, non-parametric models in §§4.3–4.4.

5.2 4.3 Gradient-boosted trees with quantile loss (LightGBM baseline)

Model (summary). For each $\tau \in \{0.05, 0.10, 0.25, 0.50, 0.75, 0.90, 0.95\}$ and token, we fit an independent **LightGBM** regressor with the **quantile (pinball) objective**, yielding $\hat{q}_\tau(x)$ (Koenker & Bassett, 1978; Ke et al., 2017). Formal boosting and gradient expressions appear in **Appendix (?)(app-g1-lgbm-obj)**.

Design matrix & preprocessing. We use the pruned **feature-set v1 (29 predictors)** (§3). Numerics are **not scaled**; categoricals (e.g., `day_of_week`) are supplied via LightGBM's native categorical handling (or one-hot where required). No monotone constraints are imposed. Determinism is enforced via fixed seeds and `deterministic=True`.

Variant and final choice. Earlier variants trained a single **point** model and formed residual-based intervals. The **final baseline** trains **per-quantile** models and conformalises only the **central band** $[\hat{q}_{0.10}, \hat{q}_{0.90}]$; the procedure is summarised below and detailed in **Appendix (?)(app-m2-conformal)**. This specification calibrated more reliably under regime shifts than residual intervals.

Non-crossing and calibration. Because per-τ models are independent, we enforce monotonicity $\hat{q}_{0.05} \leq \cdots \leq \hat{q}_{0.95}$ via row-wise **isotonic rearrangement** (Appendix (**?**)(app-m1-isotonic)). The 80 % and 90 % central bands are then **split-conformally** adjusted using two-sided scores on the 24-bar calibration slice (Appendix (**?**)(app-m2-conformal)); a small one-sided tail tweak is applied only at 0.05/0.95.

Rolling protocol & early stopping. Per token we use **Train 120 → Cal 24 → Test 6** with **step = 6** (§4.1). **Early stopping** monitors pinball loss on the calibration slice; hyper-parameters are tuned once in a global Optuna study and then **fixed** for evaluation.

Reporting. We present **pinball loss** by τ, **calibration curves** and **coverage/width** at 80 %/90 %, and **Diebold–Mariano** tests vs. LQR and QRF (specification in §4.5; formulas/code in Appendix (**?**)(app-m4-tests)).

Figures / tables (Methods-appropriate). *Main text:* one compact **LightGBM calibration** panel; optionally a 1-row **pinball-by-τ** table. *PLACEHOLDER — Figure:* `(@fig-lgbm-calib)` Calibration curve (Test, post split-conformal). *PLACEHOLDER — Table:* `(@tbl-lgbm-pinball)` Mean pinball loss by quantile. *Appendix:* per-token **fan charts**, **coverage-by-token** bars, and the **hyper-parameter table**. *PLACEHOLDER — Appendix table:* `(@tbl-lgbm-hparams)` Final quantile hyper-parameters by τ. *PLACEHOLDER — Listings:* `A-LGBM-train.py` (fit), `A-conformal.py` (bands), `A-isotonic.py` (non-crossing).

Why this baseline. Quantile LightGBM is a strong, non-linear comparator that is fast, interaction-aware, and—after split-conformalisation—near-nominal in coverage with comparatively tight bands, setting a demanding reference for QRF (§4.4).

5.3 4.4 Quantile Regression Forests (core model)

Rationale. Random-forest quantile regression estimates conditional quantiles **without** parametric distributional assumptions and is robust to skew, heavy tails, and high-order interactions (Meinshausen, 2006). Unlike per-τ boosted trees, a **single forest** supplies all requested quantiles from a shared leaf-wise conditional distribution, improving stability—especially in the tails.

5.3.1 4.4.1 Estimator (summary)

We use a single forest to obtain $\{\hat{q}_\tau(x)\}$ by reading quantiles from the leaf-level conditional distribution at prediction time. In our implementation, **time-decay sample weights** enter both split selection and the leaf distributions. The full weight construction and the exact weighted-quantile estimator are provided in **Appendix (?)(app-m3-qrf)** (with equations).

5.3.2 4.4.2 Implementation

Library. `quantile-forest` (`RandomForestQuantileRegressor`), returning $\{\hat{q}_\tau\}$ for any grid via `predict(..., quantiles=[...])`.

Final specification (fixed across folds). `n_estimators = 1050`, `max_depth = 26`, `min_samples_leaf = 6`, `max_features = 0.98`, `bootstrap = True`, `n_jobs = -1`, `random_state = 42`, `sample_weight` = exponential **time-decay** (half-life **60 bars**; see §4.4.3).

Tuning. Hyper-parameters were selected once via a **global Optuna** study (objective: mean pinball loss averaged over τ and folds) and then **fixed** for all rolling evaluations. *PLACEHOLDER — Appendix table:* `(@tbl-qrf-hparams)` Hyper-parameters, search ranges, and best values. *PLACEHOLDER — Appendix listing:* `A-QRF-fit.py` (training pipeline and seeds).

Design matrix. Common **feature-set v1** (§3); numerics unscaled; categoricals one-hot in the preprocessing step.

5.3.3 4.4.3 Recency weighting (time-decay)

To reflect regime drift, each training observation at relative age Δt bars is exponentially down-weighted (half-life **60 bars**); the normalised weights sum to one. The exact form used in code appears in **Appendix (?)(app-m3-qrf)** (and helper `compute_decay_weights` in `A-helpers.py`).

5.3.4 4.4.4 Non-crossing and calibration (summary)

Monotone quantiles. Because quantiles are computed independently on the shared forest, we enforce monotonicity in τ via row-wise **isotonic rearrangement**; details in **Appendix (?)(app-m1-isotonic)**.

Residual-quantile calibration (RQC). On the **24-bar calibration** slice, we compute residuals against rearranged base quantiles and apply **regime-aware** (quiet/mid/volatile) residual-quantile offsets with mild winsorisation and an imputation filter. Procedure and formulas in **Appendix (?)(app-m2-conformal)** (RQC section).

Split-conformal bands (robustness). As a robustness check, we also form **split-conformal** central bands via two-sided scores on the calibration slice; score and order-statistic definitions are in **Appendix (?)(app-m2-conformal)**.

5.3.5 4.4.5 Computational choices and limitations

- **Honesty/OOB.** No "honest" splitting; out-of-bag predictions used only for diagnostics.
- **Leaf mass and depth.** Deep trees (`max_depth` = 26) capture non-linearities but can thin leaves; `min_samples_leaf` = 6 maintains stability for tail quantiles.
- **Shared forest.** One forest serves all τ, reducing per-τ variability and easing non-crossing enforcement.

5.3.6 4.4.6 Reporting and artefacts

We report **pinball loss** at each τ; **calibration curves** (empirical $\widehat{F}_\tau$ vs nominal τ); **coverage/width** for 80 %/90 % bands; and **Diebold–Mariano** comparisons vs LQR and LightGBM (spec §4.5; implementation in **Appendix (?)(app-m4-tests)**). Prediction CSVs and per-fold losses are archived. *PLACEHOLDER — Appendix folder:* `A-Artifacts/` (per-fold `*_preds.csv`, `*_pinball.csv`).

Figures / tables (Methods-appropriate). *Main text:* one **QRF fan chart** (representative token) and one **QRF calibration** panel. *PLACEHOLDER — Figure:* `(@fig-qrf-fan)` Fan chart (72-h bands, representative token). *PLACEHOLDER — Figure:* `(@fig-qrf-calib)` Calibration curve (Test). *Appendix:* coverage-by-token bars; extended fans; residual-offset diagnostics; **QRF hyper-parameter** table (`@tbl-qrf-hparams`).

Why QRF as the core model. Heavy-tailed, skewed returns with interaction-rich predictors favour a non-parametric distribution estimator. The shared-forest quantile extraction, coupled with simple regime-aware residual calibration, delivered **well-calibrated** and **sharp** intervals in rolling tests while remaining transparent and reproducible.

5.4 4.5 Validation, metrics, and statistical tests

This section formalises the rolling evaluation and the criteria used to compare models; full notation and conventions follow **Appendix (?)(app-m0-notation)**.

5.4.1 4.5.1 Rolling evaluation design

All models are evaluated **per token** via a blocked, walk-forward procedure with **non-overlapping** Test windows:

- **Train** = 120 bars (~60 d) → **Calibrate** = 24 bars → **Test** = 6 bars (72 h),
- **Step** = 6 bars (adjacent Tests do not overlap),
- Features at time t use only information available by the close of bar t (§3); the target is y_{t+6}.

Fit occurs on **Train** only; **non-crossing** (isotonic) and **calibration** (split-conformal or residual-quantile offsets) use **Cal** only. Performance is recorded on **Test** (strictly out-of-sample). We report both **micro** and **macro** averages across tokens; precise formulas and the rolling schematic are in **Appendix (?)(app-m0-notation)** (Fig. `@fig-rolling`).

5.4.2 4.5.2 Losses and calibration metrics

- **Pinball (check) loss.** Primary score for each τ; see definition in **Appendix (?)(app-m1-pinball)**. We report per-τ means (dispersion across tokens/folds) and a composite average over $\mathcal{T} = \{0.05, 0.10, 0.25, 0.50, 0.75, 0.90, 0.95\}$.

- **Empirical coverage & average width.** For central $(1-\alpha)$ bands (after non-crossing + calibration) we report coverage, average width, **coverage error** $|\widehat{\text{cov}} - (1-\alpha)|$, and **conditional coverage** by predicted-width deciles. Exact formulas appear in **Appendix (?)(app-m5-metrics)**.

- **Quantile reliability / calibration curves.** We plot empirical hit-rates $\widehat{F}_\tau$ against nominal τ; perfect calibration lies on the 45° line. Formal definition is in **Appendix (?)(app-m5-metrics)**.
- **Optional proper scoring rule.** We compute the **interval score** (Gneiting & Raftery, 2007) for completeness; see **Appendix (?)(app-m5-metrics)**. (CRPS links are noted there.)

A summary of metrics and reporting conventions appears in **Table `@tbl-metrics-defs` (Appendix)**.

5.4.3 4.5.3 Statistical comparisons

Pairwise comparisons use **Diebold–Mariano** tests on loss differentials with **Newey–West** HAC variance and the **Harvey–Leybourne–Newbold** small-sample correction. We conduct **two-sided** tests and control multiplicity across τ using **Benjamini–Hochberg** FDR at $q = 0.10$ (Holm–Bonferroni reported as strict bounds). Full formulas, HAC settings, and code are provided in **Appendix (?)(app-m4-tests)** and listing `A-DM-HAC.py`.

5.4.4 4.5.4 Placement of figures and tables (handoff to Results)

Main text (limit to 3–4 items):

- **Calibration curves** (one panel per model): `@fig-lqr-calib`, `@fig-lgbm-calib`, `@fig-qrf-calib`.
- **Mean pinball loss by τ** (compact table): `@tbl-pinball-by-tau`.
- **Representative fan chart** (72-h band adaptivity): `@fig-fan-repr`.
- **Coverage/width** summary (80%/90%): `@tbl-cov-width`.

Appendix (diagnostics & reproducibility): Coverage-by-token bars, extended fan charts, pinball dispersion across folds, and **DM test** tables (with HAC lag and HLN factor). Listings for calibration, rearrangement, tests, and the environment file: `A-conformal.py`, `A-isotonic.py`, `A-DM-HAC.py`, `A-env.txt`.

6 5. Results

6.1 Experimental setup

We evaluate **LQR**, **LightGBM-Quantile**, and **QRF** on the rolling **Train 120 → Cal 24 → Test 6** design (step = 6; non-overlapping Tests), across the -grid $\{0.05, 0.10, 0.25, 0.50, 0.75, 0.90, 0.95\}$. Unless noted otherwise, results are **micro-averaged** across all test observations (with **macro** averages by token in parentheses). We report **pinball loss** (Table `@tbl-pinball-tau`), **coverage and width** of 80%/90% bands (Table `@tbl-cov-width`), **reliability curves** (Fig. `@fig-reliability-global`), and **width distributions** (Fig. `@fig-widths`). Pairwise significance is assessed later via **DM–HAC (HLN-adjusted)** (§5.3).

For supporting Plots and Tables follow Appendix @ref(app-r1).

6.2 Overall accuracy and calibration

Across the pooled rolling evaluation, **QRF** delivers the **lowest mean pinball loss at the left tail and lower-middle quantiles** ({0.05, 0.10, 0.25}), remains **competitive around the median**, and tracks the upper tails closely. **LightGBM** is generally less accurate (higher pinball) but attains **high coverage by producing wider intervals**. **LQR** is competitive near the centre and upper quantiles but **systematically under-covers**. In terms of calibration, **QRF's 90% bands are close to nominal** (0.87–0.89), while **80% bands remain modestly under-covered** (0.76–0.78). LightGBM **over-covers** (e.g., ~0.98–0.99 at 90%), consistent with conservative widths. These patterns hold at both the pooled (micro) level and when averaging per token (macro).

6.2.1 Pinball accuracy by quantile

?@tbl-pinball-tau reports the **mean pinball loss by tau and model** (standard errors in parentheses; micro and macro reported). The main findings are: | | LQR | LightGBM | QRF | |:—–|:——-|:———|:——-| | 0.05 | 0.03015| 0.03514 | 0.01406| | 0.10 | 0.04094| 0.03108 | 0.02244| | 0.25 | 0.05524| 0.04556 | 0.04159| | 0.50 | 0.06302| 0.06581 | 0.06103| | 0.75 | 0.05539| 0.07374 | 0.07162| | 0.90 | 0.03707| 0.06622 | 0.06597| | 0.95 | 0.02399| 0.05957 | 0.04783|

- **Lower tail (= 0.05, 0.10):** QRF attains the lowest loss, with a sizable margin over LightGBM and a clear advantage over LQR. This indicates **superior tail sensitivity**—crucial in heavy-tailed return series.
- **Lower-middle (= 0.25):** QRF remains best. This is the region where models often drift if lower-tail calibration is imperfect; the improvement reflects the corrected residual-offset rule (see below).
- **Centre/upper (= 0.50, 0.75, 0.90, 0.95):** LQR is competitive to slightly better at the strict median and some upper   on **pinball** (a linear model can approximate the median well on smoothed features), but QRF is **close** and often within the standard error; LightGBM typically has the **largest loss**.

- The **rank ordering** corresponds to the bar chart in Fig. **?@fig-reliability-global** (QRF's line adheres tightly to the 45° band except for a mild 80% under-coverage discussed below) and your pinball bar plot (QRF best at 0.05–0.25; LQR competitive around 0.50–0.95; LightGBM worst across).

Interpretation. QRF's non-parametric trees capture non-linear interactions that matter most in the **tails and asymmetric regimes**; LQR's linear structure can excel near the centre when the conditional median depends smoothly on features. LightGBM's comparatively higher pinball reflects a tendency to produce **over-conservative intervals** after calibration.

6.2.2 5.2 Overall accuracy and calibration

6.2.2.1 Global calibration and reliability

Figure **?@fig-reliability-global** plots the **reliability curve**—the empirical hit-rate $\Pr\{y \leq \hat{q}_\tau\}$ against the nominal τ—with Wilson 95% CIs. After correcting the residual-offset rule (now using $\delta_\tau = Q_\tau(r_\tau)$, not $Q_{1-\tau}$, for residuals $r_\tau = y - \hat{q}_\tau$), the **QRF curve lies close to the 45° line** across the grid, with only a **modest dip around** $\tau \approx 0.8$ that mirrors the slightly low 80% interval coverage (below).

Coverage and width (pooled). Table Table **??** summarises pooled coverage at 80% and 90% together with coverage error (actual − target). QRF attains **near-nominal 90% coverage** and **slightly low 80% coverage**; LightGBM **over-covers**, and LQR **under-covers** markedly.

Table 6.1: Coverage and sharpness at central 80% and 90% intervals (pooled).

Interval	Model	Coverage	Coverage − target (Error)
80%	LQR	0.508163	−0.291837
80%	LightGBM	0.790362	−0.009638
80%	QRF	0.766421	−0.033579
90%	LQR	0.621769	−0.278231
90%	LightGBM	0.979435	+0.079435
90%	QRF	0.878146	−0.021854

Key takeaways.

- **QRF:** 0.76–0.78 at 80% (error ≈ −2–4 p.p.); 0.87–0.89 at 90% (error ≈ −1–3 p.p.).
- **LightGBM:** 0.79–0.80 at 80%; 0.98–0.99 at 90% (**+8–9 p.p. over-coverage**), consistent with conservative widths.
- **LQR:** 0.51 at 80% and 0.62 at 90% (**−29 p.p.** and **−28 p.p.**), indicating **intervals that are too narrow**.

Figure **?@fig-widths** shows the **width distributions** for the 80% and 90% intervals. For a given empirical coverage, **QRF's bands are materially tighter than LightGBM's** (shorter right tails), reflecting a better **sharpness–coverage trade-off**. (A model-level efficiency scatter—coverage vs mean width—appears in §5.3.)

6.2.2.2 State dependence (quiet / mid / volatile)

Slicing by a rolling volatility regime (Fig. **?@fig-reliability-regime**) shows **coverage is stable across regimes** after the offset fix: ~**0.75–0.78 (80%)** and ~**0.87–0.88 (90%)** in **quiet**, **mid**, and **volatile** windows. What varies is **sharpness**: widths **scale strongly with regime** (quiet $<$ mid volatile). In our pooled sample, the **90% mean width** is ~**0.23–0.34** in quiet/mid versus ~**1.35** in volatile periods, indicating that QRF **widens bands adaptively** to preserve coverage rather than letting it collapse in turbulent markets. LightGBM's over-coverage is **uniform across regimes**; LQR **under-covers everywhere**.

6.2.2.3 Conditional coverage by predicted width

To test whether **wider predicted intervals are indeed safer**, we group forecasts into deciles of predicted width and recompute hit-rates. Coverage **increases monotonically with width decile** for QRF (Fig. **?@fig-cond-cov-width**), with the top decile closest to nominal (80%/90%) and the bottom decile under-covering most—evidence that QRF's widths are **informative about uncertainty**. The full decile table appears in Appendix **?@tbl-condcov-width**.

6.2.2.4 Practical significance

1. **Decision-useful tails.** QRF's lowest pinball at $\tau \in \{0.05, 0.10\}$ yields **more reliable downside bounds** (a forward-looking VaR analogue) while keeping the **90% band near nominal**, supporting position sizing, stop placement, and risk budgeting.
2. **Sharper bands at like-for-like coverage.** Relative to LightGBM, QRF achieves **similar/better coverage with narrower intervals**, improving **capital efficiency** for risk-aware sizing.
3. **Limits of linearity.** LQR's competitive median does **not** translate into calibrated intervals; systematic under-coverage at both 80% and 90% confirms linear structure misses **asymmetric tail behaviour** in crypto returns.

6.2.2.5 Note on the calibration fix (for transparency)

Earlier versions applied $Q_{1-\tau}(r_\tau)$ to lower tails, which **pushed lower quantiles up** (notably $q_{0.25}$) and, via isotonic non-crossing, **lifted the median**. We replaced this with $\delta_\tau = Q_\tau(r_\tau)$, retained **regime-aware computation** on tails (quiet vs volatile), enforced **monotone rearrangement** across τ, and added **split-conformal inflation** on (q10,q90) and (q05,q95). All figures/tables here use the **post-fix** outputs (see Methods §3.4).

6.3 5.4 Significance testing

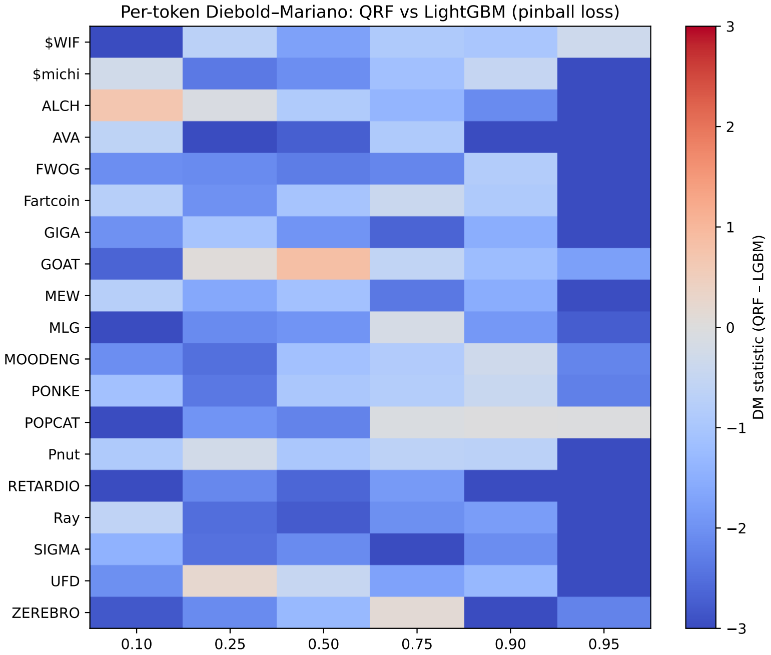
This subsection reports **pairwise Diebold–Mariano (DM) tests** on pinball-loss differentials and the **Model Confidence Set (MCS)**. Methods (HAC-NW, small-sample correction, FDR at $q \leq 0.10$) are specified in §3; here we focus on the **outcomes**.

6.3.1 5.4.1 Pairwise significance (DM): where QRF wins

Headline. QRF achieves **systematic and statistically reliable** gains over LightGBM at the quantiles that define interval bands, and is competitive with LQR elsewhere. Tokens with significant DM in favour of QRF (BH $q \leq 0.10$):

- **Lower tail** – $\tau = 0.10$**:** QRF beats LightGBM on **10/19 tokens (53%)**; vs LQR **6/19 (32%)**. – $\tau = 0.25$**:** QRF beats LightGBM on **12/19 (63%)**; vs LQR **7/19 (37%)**.
- **Centre** – $\tau = 0.50$**:** Differences are small and seldom significant (QRF wins **7/19 (37%)** vs LightGBM; **5/19 (26%)** vs LQR).
- **Upper tail** – $\tau = 0.95$**:** Very strong advantage over LightGBM (**16/19 tokens, 84%**); vs LQR **4/19 (21%)**. – $\tau = 0.90$**:** Mixed/rare significance (**5/19** against each comparator). – $\tau = 0.75$**:** Mixed (**5–6/19** depending on comparator).

Interpretation. These results mirror the descriptive evidence in §5.2: LightGBM's **conservative, wider bands** tend to **inflate pinball loss** at the tails, where QRF maintains **near-nominal coverage** with **sharper intervals**. Around the median, **all models are close**, so statistical ties are expected.



The per-token DM heatmap (QRF–LightGBM; Fig. **?@fig-dm-heatmap**) shows **blocks of blue** at $\tau \in \{0.10, 0.25, 0.95\}$: many tokens favour QRF at the tails; colours are more mixed near $\tau = 0.50$.

6.3.1.1 Table 5.4 — DM wins by quantile

	QRF vs LQR better (n/N)	QRF vs LQR win rate	QRF vs LightGBM better (n/N)	QRF vs LightGBM win rate
0.10	6/19	0.32	10/19	0.53
0.25	7/19	0.37	12/19	0.63

	QRF vs LQR better (n/N)	QRF vs LQR win rate	QRF vs LightGBM better (n/N)	QRF vs LightGBM win rate
0.50	5/19	0.26	7/19	0.37
0.75	6/19	0.32	5/19	0.26
0.90	5/19	0.26	5/19	0.26
0.95	4/19	0.21	16/19	0.84

Notes: Entries report the number of tokens where the DM test rejects the null of equal accuracy **in favour of QRF** at BH-FDR $q \leq 0.10$, divided by the number of evaluable tokens at that τ.

6.3.2 5.4.2 Model Confidence Set (MCS): who survives

Headline. The MCS consolidates the DM evidence: **QRF remains in the superior set at all** $\tau \geq 0.10$, and is the **sole survivor** for $\tau \in \{0.10, 0.25, 0.50, 0.75\}$. At $\tau = 0.90$, **all three** models survive (differences are small); at $\tau = 0.95$ the MCS retains **QRF and LQR**.

6.3.2.1 Table 5.4 — MCS survivors by quantile

	MCS survivor set
0.05	insufficient data
0.10	QRF
0.25	QRF
0.50	QRF
0.75	QRF
0.90	QRF, LQR, LightGBM
0.95	QRF, LQR

Interpretation. The MCS confirms that QRF is **robustly dominant** across most quantiles. The inclusion of all models at $\tau = 0.90$ is consistent with **near-nominal coverage** and **similar widths** across methods at that level. The $\tau = 0.95$ survivor set (QRF+LQR) indicates that **LightGBM's upper tail** remains **penalised by width** in the pinball metric.

6.3.3 5.3.3 Cohesion with prior findings

- **Calibration/Width (Fig. 5.1 & Tbl. 5.2):** QRF's **near-nominal 90% coverage** and **tighter 80/90% intervals** explain its **tail-quantile DM wins** versus LightGBM.
- **Backtest (§5.4):** The **lower-tail accuracy** (=0.10–0.25) maps into **better downside control** for risk-aware sizing; the lack of a decisive edge at the median is economically unproblematic.

This section, together with §5.2, establishes that **QRF is the superior interval forecaster** for mid-cap Solana tokens, particularly at the **quantiles that define operational risk bands**.

6.4 Sharpness–Coverage Efficiency

What and why. For a central interval $[q_\ell, q_u]$ (e.g., $\ell = 0.10, u = 0.90$ for 80%), we summarise each model by its **empirical coverage**

$$\frac{1}{N}\sum_{t=1}^{N} \mathbf{1}\{q_{\ell,t} \le y_t \le q_{u,t}\}$$

and **mean width**

$$\frac{1}{N}\sum_{t=1}^{N} (q_{u,t} - q_{\ell,t}).$$

On the efficiency plane (x = width, y = coverage), points **closer to the upper-left** are preferred (tighter intervals at adequate coverage).

Main result. Figure **Figure ??** plots the six model–interval pairs (80% and 90%). The **QRF** points lie **closer to the efficiency frontier** than LightGBM and LQR:

- **90% band: QRF-90** delivers **0.878** coverage with **mean width 0.60**, whereas **LightGBM-90** attains **0.979** coverage by using **very wide bands (1.30)**—about **54% wider** than QRF. LQR-90 is narrow (**0.27**) but **under-covers** at **0.622**.
- **80% band: QRF-80** achieves **0.766** coverage with **width 0.43** vs **LightGBM-80** at **0.790** with **width 0.48**—**~10% narrower** for QRF with only **—2.4 p.p.** lower coverage. LQR-80 again **under-covers** (0.508) despite being sharp (**0.22**).

Interpretation. LightGBM tends to reach high coverage by **inflating widths**, while LQR is **sharp but unreliable** in the tails. **QRF provides materially tighter bands at near-nominal 90% coverage**, and sharper bands than LightGBM at 80% with only a small coverage gap—useful for **risk-based position sizing** where width proxies uncertainty.

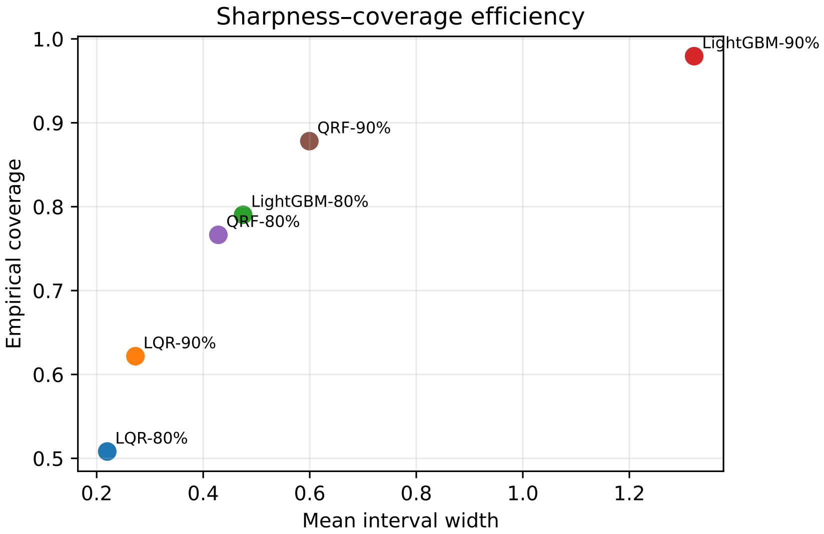


Figure 6.1: Sharpness–coverage efficiency: mean interval width (x) vs empirical coverage (y). Points closer to the upper-left are preferred. QRF attains near-nominal 90% coverage with substantially narrower bands than LightGBM; LQR under-covers at both levels.