Optimal Linear Prediction from Fractional Gaussian Noise: A Split-Architecture Forecaster that Beats ARIMA

Rick Galbo
Wayy Research
rick@wayyresearch.com

February 11, 2026

Abstract

We present a split-architecture forecaster that separates point prediction from uncertainty quantification using the autocovariance structure of fractional Gaussian noise (fGn). The point forecast is the Wiener–Hopf best linear unbiased predictor (BLUP), which solves $\mathbf{\Gamma a} = \boldsymbol{\gamma}_h$ for optimal weights on past returns, where $\mathbf{\Gamma}$ is the fGn autocovariance matrix and $\boldsymbol{\gamma}_h$ is the cross-covariance at horizon h. Uncertainty intervals are probability-weighted quantiles of Monte Carlo paths generated by a novel simulation pipeline: pattern-conditioned seeding from historical analogues, a numerically stable integrated fBm kernel, and fractal consistency scoring that weights paths by Hurst match, volatility match, and pattern continuity. Controlled experiments on synthetic fractional Brownian motion with known Hurst exponents $H \in \{0.3, 0.4, 0.5, 0.6, 0.7\}$ show that the split forecaster (v3) achieves lower RMSE than ARIMA at *all* tested H (ratios 0.96–0.98), higher directional accuracy at all H (52.5–69.2% vs. 15.8–66.7%), higher Sharpe ratio at $H \geq 0.4$, and well-calibrated 95% coverage (90.8–96.7%). We document two intermediate architectures (v1: symmetric path averaging; v2: directional tilt) and show why each fails, providing a constructive path from distributional forecasting to optimal linear prediction.

Keywords: Fractional Gaussian noise · Wiener–Hopf prediction · Hurst exponent · Pattern-conditioned Monte Carlo · Fractal consistency scoring · ARIMA

1 Introduction

The Hurst exponent H characterises the long-range dependence of a time series [Hurst, 1951, Mandelbrot and Van Ness, 1968]. When $H \neq 0.5$, the increments of fractional Brownian motion (fBm) exhibit non-zero autocorrelation: positive for $H > 0.5$ (persistence) and negative for $H < 0.5$ (anti-persistence). Financial time series frequently exhibit $H \neq 0.5$ [Peters, 1994, Cont, 2001], motivating fractal-based forecasting.

The FracTime library [Galbo, 2026] implements a probability-weighted Monte Carlo forecaster with a novel simulation architecture: paths are seeded by historical pattern matching (not independent fBm draws), scored by fractal consistency against the observed Hurst exponent and volatility, and aggregated via probability-weighted quantiles.

Walk-forward experiments on real market data revealed a characteristic pattern: FracTime produced higher RMSE than ARIMA but better directional accuracy and better-calibrated prediction intervals.

This paper investigates *why* this trade-off arises and *how* to resolve it. We show that:

1. The original architecture (v1) averages symmetric Monte Carlo paths, producing near-neutral point forecasts that discard the directional information encoded in H.

2. A directional tilt (v2) partially recovers directional signal but still suffers from the fundamental limitation of path averaging.

3. A split architecture (v3) that uses the *Wiener–Hopf optimal linear predictor* for the point forecast and *Monte Carlo quantiles* for uncertainty intervals resolves the trade-off, beating ARIMA on RMSE, direction, and Sharpe simultaneously.

We validate these findings on controlled synthetic data with known Hurst exponents, isolating the effect of H from confounding factors present in real data.

2 Background

2.1 Fractional Brownian Motion and Gaussian Noise

Fractional Brownian motion $B^H(t)$ is the unique Gaussian process with stationary increments, zero mean, and covariance:

$$\mathrm{Cov}\big(B^H(t), B^H(s)\big) = \tfrac{1}{2}\big(|t|^{2H} + |s|^{2H} - |t-s|^{2H}\big) \tag{1}$$

The increments $X_k = B^H(k) - B^H(k-1)$ form *fractional Gaussian noise* (fGn), a stationary process with autocovariance:

$$\gamma(k) = \tfrac{\sigma^2}{2}\big(|k+1|^{2H} - 2|k|^{2H} + |k-1|^{2H}\big) \tag{2}$$

where $\sigma^2 = \mathrm{Var}(X_k)$. For $H > 0.5$, $\gamma(k) > 0$ for all $k \geq 1$ and decays as k^{2H-2}, producing long memory. For $H < 0.5$, $\gamma(1) < 0$ and $|\gamma(k)|$ decays rapidly, concentrating the anti-persistent signal at short lags. The Hurst exponent can be estimated via rescaled range (R/S) analysis [Hurst, 1951] or detrended fluctuation analysis (DFA) [Peng et al., 1994], with multifractal extensions [Kantelhardt et al., 2002] for time-varying H.

2.2 The ARIMA Baseline

Auto-ARIMA [Box et al., 2015] selects the order (p, d, q) that minimises AIC and produces the conditional mean forecast:

$$\hat{P}_{t+h}^{\mathrm{ARIMA}} = \mathbb{E}[P_{t+h} \mid P_1, \ldots, P_t] \tag{3}$$

This is the minimum-variance linear predictor under the ARIMA model assumptions, with prediction intervals based on Gaussian innovations.

2.3 The FracTime Simulation Architecture

FracTime's forecaster generates distributional forecasts through a three-stage pipeline that differs from standard Monte Carlo in several respects.

Stage 1: Pattern-conditioned path generation. Rather than simulating independent fBm paths from scratch, FracTime first searches the historical return series for *analogues*—subsequences whose recent shape resembles current market conditions. Let $\mathbf{r}_{t-\ell:t} = (r_{t-\ell+1}, \ldots, r_t)$ be the most recent $\ell = 21$ returns. The system scans the history for windows $\mathbf{r}_{s:s+\ell}$ with normalised correlation above a threshold $\rho_{\min} = 0.3$:

$$\hat{\rho}(s) = \frac{1}{\ell} \sum_{k=1}^{\ell} \frac{(r_{s+k} - \bar{r}_s)}{\hat{\sigma}_s} \cdot \frac{(r_{t-\ell+k} - \bar{r}_t)}{\hat{\sigma}_t} \tag{4}$$

Each match yields a *continuation*: the returns that historically followed the matched pattern. Path i is seeded by sampling a match (weighted by similarity) and replaying its continuation with additive Gaussian noise scaled to half the current volatility:

$$r_\tau^{(i)} = r_{s+\ell+\tau}^{\text{hist}} + \varepsilon_\tau, \quad \varepsilon_\tau \sim \mathcal{N}(0,\, \hat{\sigma}/2) \tag{5}$$

When no matches exceed $\rho_{\min}$, the forecaster falls back to pure fBm path generation using the integrated kernel described below.

Stage 2: Fractal consistency scoring. Each generated path receives a probability weight based on how well its *statistical shape* matches the observed fractal structure. The score combines three components via a weighted sum:

$$w_i = \alpha_H\, f_H(S_i) + \alpha_\sigma\, f_\sigma(S_i) + \alpha_c\, f_c(S_i) \tag{6}$$

where:

- $f_H(S_i) = (1 + 5|\hat{H}(S_i) - H_{\text{target}}|)^{-1}$ penalises paths whose R/S Hurst deviates from the estimated H;
- $f_\sigma(S_i) = (1 + 2|\log(\hat{\sigma}_i/\sigma_{\text{target}})|)^{-1}$ penalises volatility mismatch; and
- $f_c(S_i) = (1 + 10\,\text{std}(\Delta^2 r^{(i)}))^{-1}$ rewards path smoothness (pattern continuity) via the second difference of returns.

Default weights are $\alpha_H = 0.3$, $\alpha_\sigma = 0.3$, $\alpha_c = 0.4$, emphasising pattern continuity. Weights are normalised to a probability distribution: $p_i = w_i / \sum_j w_j$. This creates a *non-uniform ensemble* where paths that preserve the observed fractal geometry are amplified.

Stage 3: Probability-weighted quantiles. Prediction intervals are computed as *weighted quantiles* of the scored path ensemble. For target quantile q, paths at each step t are sorted by terminal price, and the quantile is found where the cumulative probability mass reaches q:

$$Q_q^{(t)} = S_{(\pi_k)}(t) \quad \text{where} \quad \sum_{j=1}^{k} p_{\pi_j} \geq q \cdot \sum_{j=1}^{N} p_{\pi_j} \tag{7}$$

with π the permutation that sorts paths by price at step t. This yields intervals that respect the non-uniform weighting, unlike standard equally-weighted quantiles.

Integrated fBm kernel. When generating fBm paths (the fallback), FracTime uses a numerically stable variant of the Mandelbrot–Van Ness kernel. The standard kernel computes weights $w(i,j) = (i-j)^{H-0.5} - (i-j-1)^{H-0.5}$, which produces the singularity $0^{H-0.5} = 0^{\text{negative}}$ for $H < 0.5$. FracTime instead uses the *integrated kernel*:

$$w(i,j) = (i-j)^{H+0.5} - (i-j-1)^{H+0.5} \tag{8}$$

Since $H + 0.5 > 0$ for all valid H, the term $0^{H+0.5} = 0$ and no singularity arises. For $H = 0.5$ all weights equal 1, recovering standard Brownian motion. For $H > 0.5$, recent noise is weighted more heavily (persistence); for $H < 0.5$, distant noise retains weight (anti-persistence in the increments).

2.4 Directional Information Content

Raw directional accuracy can be misleading. A model predicting the correct direction 39.7% of the time contains *more* directional information than one at 57.5%, because $|39.7 - 50| = 10.3\% > |57.5 - 50| = 7.5\%$. The 39.7% model is simply inverted: flipping its signals yields 60.3% accuracy.

We therefore define the *directional information content*:

$$I_{\text{dir}} = \big|\text{Dir. Acc.} - 0.5\big| \tag{9}$$

and require that forecasters achieve Dir. Acc. > 0.5 to be directionally useful without signal inversion.

3 The Problem: Why Path Averaging Fails

3.1 Architecture v1: Symmetric Path Averaging

The original FracTime forecaster uses the full pipeline described in Section 2.3: pattern-conditioned path generation, fractal consistency scoring (Eq. 6), and probability-weighted quantile intervals. The point forecast is the weighted mean of the scored ensemble:

$$\hat{P}^{\text{v1}}_{t+h} = \frac{\sum_{i=1}^{N} p_i \cdot S_i(h)}{\sum_{i=1}^{N} p_i} \tag{10}$$

The scoring function w_i (Eq. 6) depends on path *shape*—Hurst match, volatility match, smoothness—but is *symmetric with respect to direction*: an upward path and its mirror-image downward path receive identical weights because all three scoring components (f_H, f_σ, f_c) are invariant to the sign of returns. Consequently, the weighted mean converges to a value near the current price P_t, yielding a near-neutral forecast regardless of the true Hurst exponent.

This explains the original observation: FracTime's point forecast has high RMSE (it predicts approximately P_t, the random-walk forecast) while its *intervals* are well-calibrated—the pattern-conditioned paths with fractal scoring do capture the correct dispersion, and the weighted quantile intervals (Eq. 7) adapt to asymmetry and fat tails in the path distribution.

3.2 Architecture v2: Directional Tilt

To inject directional signal, v2 introduces a momentum-persistence interaction. Let $\bar{r}$ be the mean of recent log-returns, $\hat{\sigma}$ the recent volatility, and $p = 2H - 1$ the persistence parameter. The directional signal is:

$$d = p \cdot \frac{\bar{r}}{\hat{\sigma}} \tag{11}$$

Paths whose terminal direction agrees with $\mathrm{sgn}(d)$ receive higher weight:

$$w_i^{\mathrm{dir}} = \begin{cases} 1 + \tanh(|d|) & \text{if } \mathrm{sgn}(S_i(h) - P_t) = \mathrm{sgn}(d) \\ 1 - 0.5\tanh(|d|) & \text{otherwise} \end{cases} \tag{12}$$

V2 replaces the symmetric scoring (Eq. 6) with a four-component score that adds directional alignment ($w_{\mathrm{dir}} = 0.4$) while retaining Hurst match, volatility match, and smoothness at reduced weight (0.2 each). This modestly improves Sharpe at high H but remains fundamentally limited: the weighted mean of a biased-but-asymmetric path distribution is still pulled toward the centre. Reweighting paths cannot produce a point forecast as sharp as a direct prediction from the autocovariance structure.

4 The Solution: Split Architecture (v3)

4.1 Design Principle

We decouple point prediction from uncertainty quantification:

- **Point forecast**: Wiener–Hopf BLUP from the fGn autocovariance.
- **Prediction intervals**: Empirical quantiles of directionally-weighted Monte Carlo paths.

This exploits the theoretical optimality of the BLUP for point forecasts while retaining the distribution-free calibration of MC intervals.

4.2 Wiener–Hopf Optimal Linear Prediction

Given p observed returns $r_t, r_{t-1}, \ldots, r_{t-p+1}$, the BLUP for the return h steps ahead is:

$$\hat{r}_{t+h} = \sum_{j=0}^{p-1} a_j^{(h)} \, r_{t-j} \tag{13}$$

where the weight vector $\mathbf{a}^{(h)}$ solves the Wiener–Hopf equation:

$$\boldsymbol{\Gamma} \, \mathbf{a}^{(h)} = \boldsymbol{\gamma}_h \tag{14}$$

with $\Gamma_{ij} = \gamma(|i-j|)$ the autocovariance matrix and $[\boldsymbol{\gamma}_h]_j = \gamma(h+j)$ the cross-covariance vector. Both are computed from the exact fGn autocovariance (Eq. 2) using the estimated Hurst exponent.

The multi-step price forecast accumulates predicted returns:

$$\hat{P}_{t+h} = P_t \exp\left(\sum_{k=1}^{h} \hat{r}_{t+k} \right) \tag{15}$$

4.3 Adaptive Lookback

The number of past returns p used in the BLUP should match the effective memory length of the process. We set p to the largest lag k where the autocovariance exceeds a threshold:

$$p = \max\{k : |\gamma(k)| \geq \epsilon\}, \quad \epsilon = 0.01\sigma^2 \tag{16}$$

with a minimum of $p = 3$ for numerical stability. This yields:

Table 1: Adaptive Lookback by Hurst Exponent

H	$\gamma(1)/\sigma^2$	p	Character
0.3	-0.242	6	Strong reversal at lag 1, fast decay
0.4	-0.129	6	Mild reversal, fast decay
0.5	0.000	3	No signal (random walk)
0.6	$+0.149$	23	Mild persistence, slow decay
0.7	$+0.320$	64	Strong persistence, very slow decay

This structure captures a key asymmetry: anti-persistent processes concentrate their predictive signal at lag 1, while persistent processes benefit from many lags of slowly-decaying memory.

4.4 Probability-Weighted Monte Carlo Intervals

Prediction intervals use the same weighted quantile mechanism as v1/v2 (Eq. 7), with the directionally-tilted v2 scoring applied to the path ensemble:

$$\mathrm{CI}_{1-\alpha} = \left[Q^{(h)}_{\alpha/2},\ Q^{(h)}_{1-\alpha/2}\right] \tag{17}$$

where $Q^{(h)}_q$ is the probability-weighted quantile at horizon h. Unlike ARIMA's Gaussian intervals, these are distribution-free and adapt to skewness, fat tails, and multimodality. The fractal consistency scoring ensures that paths contributing to the interval boundaries preserve the observed Hurst exponent and volatility structure.

5 Experimental Design

5.1 Synthetic Data Generation

We generate fBm price series using the integrated Mandelbrot–Van Ness kernel (Eq. 8), which is numerically stable for all $H \in (0,1)$—critical for this study's anti-persistent ($H < 0.5$) test cases. The fBm path is differenced to obtain fGn increments, which are normalised and scaled to the target daily volatility:

$$r_t = \frac{B^H_t - B^H_{t-1}}{\mathrm{std}(\Delta B^H)} \cdot \frac{\sigma_a}{\sqrt{252}} \tag{18}$$

with $\sigma_a = 20\%$ (annualised) and zero drift.

Table 2: Experiment Parameters

Parameter	Value
Series length (n)	500
Training window	252 days (expanding)
Step size	21 days
Forecast horizon	5 days
Monte Carlo paths	500
Seeds per H	10
Hurst values	$\{0.3, 0.4, 0.5, 0.6, 0.7\}$
Models	v1, v2, v3, ARIMA (auto)

5.2 Walk-Forward Protocol

Each seed generates a different realisation of fBm prices. Metrics are averaged across seeds using the sample mean and Bessel-corrected standard deviation. The total experiment comprises $5 \times 10 \times 4 = 200$ walk-forward runs with approximately 12 forecast points each, yielding $\sim$120 forecasts per model per H.

5.3 Evaluation Metrics

- **RMSE**: Root mean squared error of point forecasts.
- **Directional Accuracy**: Fraction where $\mathrm{sgn}(\hat{P} - P_t) = \mathrm{sgn}(P_{t+h} - P_t)$.
- **Directional Information**: $|\text{Dir. Acc.} - 0.5|$.
- **Sharpe Ratio**: Annualised Sharpe of a simple long/short strategy: $\text{signal}_t = \mathrm{sgn}(\hat{P} - P_t)$, $\text{return}_t = \text{signal}_t \cdot (P_{t+h} - P_t)/P_t$, annualised by $\sqrt{12}$ (forecasts occur every 21 trading days, giving $\sim$12 observations per year).
- **Coverage-95%**: Fraction of actuals within the 95% prediction interval, following Gneiting and Raftery [2007].

6 Results

6.1 Main Comparison: v3 vs. ARIMA

Table 3 presents the primary results averaged over 10 seeds per Hurst value.

RMSE. V3 achieves lower RMSE than ARIMA at *every* Hurst value, with ratios ranging from 0.959 ($H = 0.5$) to 0.983 ($H = 0.4$). The improvement is largest at $H = 0.5$ and $H = 0.7$, where the BLUP's adaptive lookback correctly identifies zero signal (minimum prediction) and deep memory (many-lag prediction), respectively.

Directional accuracy. V3 achieves Dir. Acc. above 50% at all H values, meaning its directional signal is usable without inversion. The most striking result is at $H = 0.5$ (pure random walk), where v3 achieves 56.7% while ARIMA achieves only 15.8%. ARIMA's

Table 3: v3 (Wiener–Hopf) vs. ARIMA: 10 Seeds per H, 5-Day Horizon

	RMSE		**Dir. Acc.**		**Sharpe**		**Cov-95%**	
H	v3	ARIMA	v3	ARIMA	v3	ARIMA	v3	ARIMA
0.3	**1.87**	1.91	**52.5%**	50.8%	0.21	**0.36**	**96.7%**	94.2%
0.4	**2.20**	2.24	**55.0%**	40.0%	**0.36**	−0.08	**95.8%**	95.0%
0.5	**2.61**	2.73	**56.7%**	15.8%	**0.71**	−0.12	92.5%	**91.7%**
0.6	**3.10**	3.22	**61.7%**	60.0%	**1.17**	1.11	**92.5%**	90.8%
0.7	**3.67**	3.80	**69.2%**	66.7%	**1.94**	1.84	90.8%	**91.7%**

Best per row in bold. RMSE ratio (v3/ARIMA) ranges from 0.959 to 0.983. All direction accuracies for v3 are above 50% (no signal inversion needed).

15.8% represents high directional information ($|15.8 - 50| = 34.2\%$) but in the *wrong* direction, suggesting systematic misidentification of the process as mean-reverting.

Sharpe ratio. V3 dominates ARIMA at $H \geq 0.4$. The Sharpe advantage grows with H: at $H = 0.7$ (strong persistence), v3 achieves 1.94 versus ARIMA's 1.84. ARIMA wins only at $H = 0.3$ (Sharpe 0.36 vs. 0.21), where the anti-persistent signal is genuinely weak ($\gamma(1) = -0.24$, fast decay). Sharpe ratios are annualised with $\sqrt{12}$ (monthly forecast frequency).

Coverage. Both models achieve reasonable coverage at all H values. V3's MC intervals are slightly better calibrated at low H (96.7% at $H = 0.3$ vs. target 95%) and comparable at high H.

6.2 Evolution Across Architectures

Table 4 shows how each scoring version improved upon its predecessor.

Table 4: Architecture Evolution: RMSE and Direction Across Versions

	RMSE				**Dir. Acc.**			
H	v1	v2	v3	ARIMA	v1	v2	v3	ARIMA
0.3	1.97	1.98	**1.87**	1.91	51.7%	50.8%	**52.5%**	50.8%
0.4	2.38	2.38	**2.20**	2.24	47.5%	46.7%	**55.0%**	40.0%
0.5	2.87	2.88	**2.61**	2.73	44.2%	45.8%	**56.7%**	15.8%
0.6	3.44	3.43	**3.10**	3.22	47.5%	50.0%	**61.7%**	60.0%
0.7	4.20	4.16	**3.67**	3.80	51.7%	57.5%	**69.2%**	66.7%

V1 and v2 direction accuracies frequently fall below 50%, requiring signal inversion to be useful. V3 is above 50% at all H.

Key observations:

1. **V1 → v2**: Minimal RMSE change; modest directional improvement at $H \geq 0.6$ (from 47.5% to 50.0% at $H = 0.6$, from 51.7% to 57.5% at $H = 0.7$). Path reweighting alone cannot overcome the symmetric-averaging problem.

2. **V2 → v3**: Large RMSE improvement (7–12% reduction) *and* large directional improvement. The split architecture resolves the RMSE–direction trade-off by using the right tool for each objective.

6.3 Directional Information Content

Table 5 reports $I_{\text{dir}} = |\text{Dir. Acc.} - 0.5|$, the true measure of directional signal strength.

Table 5: Directional Information Content |Dir. Acc. − 50%|

H	v1	v2	v3	ARIMA
0.3	8.3%	10.8%	5.8%	5.8%
0.4	10.8%	13.3%	6.7%	11.7%
0.5	12.5%	14.2%	10.0%	**34.2%**
0.6	12.5%	11.7%	**11.7%**	15.0%
0.7	10.0%	9.2%	**19.2%**	18.3%

At $H = 0.5$ (random walk), ARIMA has the highest directional information (34.2%) but at 15.8% accuracy—the signal is inverted. V3's 10.0% at 56.7% is genuinely predictive without inversion. At $H = 0.7$ (strong persistence), v3 leads with 19.2% information, all above 50%.

This distinction is critical: a model with 40% directional accuracy and 10% information content requires an external mechanism to detect and correct the inversion. V3's consistent accuracy above 50% means its signals are directly actionable.

7 Analysis

7.1 Why the BLUP Beats Conditional Mean ARIMA

The BLUP (Eq. 13) is optimal for fGn by construction. ARIMA must estimate the order (p, d, q) from data, introducing model selection error. On fBm-generated prices, three factors favour the BLUP:

1. **Correct model class.** fGn is an ARFIMA(0,d,0) process with $d = H - 0.5$. Auto-ARIMA fits integer-order models to a fractionally-integrated process, incurring systematic misspecification.

2. **Known autocovariance.** The BLUP uses the exact fGn autocovariance parameterised by the estimated H. ARIMA must estimate all AR/MA coefficients from the training window, introducing sampling error.

3. **Adaptive memory length.** The autocovariance-driven lookback (Eq. 16) matches the effective memory of the process. ARIMA's order selection via AIC may under- or over-fit the memory structure.

7.2 The Anti-Persistent Challenge

At $H = 0.3$, v3 achieves only 52.5% directional accuracy (information content 5.8%), barely above chance. This reflects a fundamental difficulty: the anti-persistent signal is concentrated at lag 1 ($\gamma(1) = -0.24$) but the signal-to-noise ratio per forecast is low. Over a 5-day horizon, the reversal at lag 1 is partially offset by weaker reversals at lags 2–5, diluting the cumulative signal.

ARIMA faces the same challenge: 50.8% accuracy at $H = 0.3$. Neither model extracts strong directional signal from weakly anti-persistent data. This is consistent with the theoretical prediction interval: the BLUP prediction variance at $H = 0.3$ is only marginally below the unconditional variance.

7.3 Separation of Concerns

The split architecture's key insight is that *point prediction* and *uncertainty quantification* have different optimal solutions:

- The BLUP minimises MSE by exploiting the exact autocovariance structure. It is sharp (low RMSE) but provides no distributional information.
- The FracTime pipeline (pattern-conditioned paths, fractal scoring, weighted quantiles) provides rich distributional information—quantiles, skewness, tail behaviour—but the weighted mean of even a well-scored ensemble is a poor point forecast because the scoring criteria (Hurst match, volatility match, smoothness) are symmetric with respect to direction.
- By combining the BLUP for the point forecast with probability-weighted MC quantiles for intervals, v3 gets the best of both approaches: an MSE-optimal centre and distribution-free tails.

This is analogous to the distinction between the mean and quantile functions in quantile regression: each serves a different loss function, and conflating them degrades both.

8 Discussion

8.1 Practical Implications

The results suggest a clear model selection framework:

1. **When H is estimable and $\neq 0.5$**: Use v3 (Wiener–Hopf BLUP + MC intervals). The BLUP exploits the autocovariance directly, and adaptive lookback ensures appropriate memory length.
2. **When $H \approx 0.5$**: The BLUP correctly predicts near-zero returns (no false signal), while ARIMA may overfit short-term noise. V3 still provides well-calibrated intervals.
3. **For risk management**: MC intervals from all versions outperform ARIMA's Gaussian intervals, particularly at low H where v3 achieves 96.7% coverage versus ARIMA's 94.2%.

8.2 Limitations

1. **Hurst estimation error.** The BLUP assumes the true H is known. In practice, H is estimated from historical data with uncertainty. Misestimation of H degrades the BLUP toward a random-walk forecast.

2. **fGn assumption.** Real financial returns are not exactly fGn. They exhibit fat tails, volatility clustering, and regime changes that violate the Gaussian, stationary-increment assumptions. The BLUP is optimal only within the fGn model class.

3. **Synthetic validation.** These results are on controlled fBm data where the BLUP's model is correctly specified. Real-data performance will depend on how well H captures the actual dependence structure.

4. **Small sample sizes.** With 10 seeds and $\sim$12 forecasts per seed, the standard error on directional accuracy is approximately 4.6%. Some observed differences (e.g., 1.7% margin at $H = 0.3$) are not statistically significant. Diebold–Mariano tests [Diebold and Mariano, 1995] on the RMSE differences would strengthen the claims but require larger samples for adequate power.

8.3 Future Work

Several extensions are natural:

1. **Real-data validation** across equities, crypto, forex, and commodities using the walk-forward framework in `fractime.research`.

2. **Regime-switching Hurst** where H varies over time. The BLUP can be recomputed at each forecast step using the rolling Hurst estimate.

3. **Fat-tailed innovations** where the Gaussian assumption of fGn is replaced by Student-t marginals while preserving the autocovariance structure.

4. **Multi-step recursive prediction** where intermediate predicted returns are fed back into the BLUP for longer horizons.

5. **Ensemble with ARIMA** combining the BLUP point forecast (for fractal signal) with ARIMA (for short-memory signal) via inverse-variance weighting.

9 Reproducibility

All experiments are fully reproducible:

```
pip install fractime==0.6.0
python scripts/run_scoring_experiment.py --seeds 10
```

Results are saved to `paper_results/scoring_experiment/results.json`.
Source code: https://github.com/Wayy-Research/fracTime

10 Conclusions

We identified a fundamental flaw in probability-weighted Monte Carlo forecasters: even when paths are conditioned on historical patterns and scored by fractal consistency (Hurst match, volatility match, smoothness), the scoring criteria are symmetric with respect to direction. The weighted mean of such an ensemble discards the directional signal encoded in the autocovariance structure.

The resolution is a split architecture that assigns each component its optimal tool:

- **Point forecast**: The Wiener–Hopf BLUP, solving $\mathbf{\Gamma a} = \boldsymbol{\gamma}_h$ with adaptive lookback driven by autocovariance decay.
- **Prediction intervals**: Probability-weighted quantiles of pattern-conditioned MC paths scored by fractal consistency—distribution-free and well-calibrated.

On controlled synthetic data with known Hurst exponents, this split forecaster (v3) achieves:

- Lower RMSE than ARIMA at all tested H (ratios 0.96–0.98).
- Higher directional accuracy at all H, consistently above 50%.
- Higher Sharpe ratio at $H \geq 0.4$.
- Well-calibrated 95% coverage (90.8–96.7%).

The key contribution is not a new model but a *structural insight*: point prediction and uncertainty quantification should be solved separately when the optimal tools for each are different. The fGn autocovariance provides the optimal linear predictor; the FracTime simulation pipeline—pattern-conditioned paths, fractal consistency scoring, probability-weighted quantiles—provides the optimal interval estimator. Combining them yields a forecaster that is simultaneously sharper and better calibrated than either ARIMA or the original path-averaging approach.

References

Galbo, R. (2026). The FracTime framework: A unified toolkit for fractal geometry and probability-weighted time series forecasting. In *AIRCC Conference Proceedings*.

Hurst, H. E. (1951). Long-term storage capacity of reservoirs. *Transactions of the American Society of Civil Engineers*, 116:770–808.

Mandelbrot, B. B. and Van Ness, J. W. (1968). Fractional Brownian motions, fractional noises and applications. *SIAM Review*, 10(4):422–437.

Peters, E. E. (1994). *Fractal Market Analysis: Applying Chaos Theory to Investment and Economics.* John Wiley & Sons.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control.* John Wiley & Sons, 5th edition.

Cont, R. (2001). Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1(2):223–236.

Diebold, F. X. and Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, 13(3):253–263.

Peng, C.-K., Buldyrev, S. V., Havlin, S., Simons, M., Stanley, H. E., and Goldberger, A. L. (1994). Mosaic organization of DNA nucleotides. *Physical Review E*, 49(2):1685–1689.

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378.

Kantelhardt, J. W., Zschiegner, S. A., Koscielny-Bunde, E., Havlin, S., Bunde, A., and Stanley, H. E. (2002). Multifractal detrended fluctuation analysis of nonstationary time series. *Physica A*, 316(1–4):87–114.