Cross-Asset Transfer Learning for Rough Volatility Forecasting: An Empirical Study on Traditional and Cryptocurrency Markets

Ronit Dhansoia
ronitdhansoia@gmail.com

December 2025

Abstract

This paper investigates the application of transfer learning to rough volatility forecasting across traditional equity and cryptocurrency markets. We develop a neural network architecture combining LSTM encoders with attention mechanisms, pre-trained on S&P 500 realized volatility and fine-tuned for Bitcoin volatility prediction. Our empirical analysis confirms that both markets exhibit rough volatility characteristics with Hurst parameters below 0.5 (S&P 500: $H = 0.468$, Bitcoin: $H = 0.353$). Contrary to expectations, we find that transfer learning does not improve neural network performance over training from scratch on the target domain. More significantly, the simple Heterogeneous Autoregressive (HAR) model substantially outperforms all neural network approaches (RMSE 0.0844 vs 0.1491). Our findings suggest that HAR's multi-scale structure effectively captures the dominant patterns in rough volatility, leaving limited room for neural network improvements. We discuss implications for practitioners and directions for future research.

Keywords: Rough volatility, Transfer learning, Cryptocurrency, HAR model, Deep learning, Realized volatility forecasting

Contents

1 Introduction

Volatility forecasting is a cornerstone of quantitative finance, with applications spanning risk management, derivative pricing, and portfolio optimization. The recent discovery of "rough volatility"—characterized by volatility paths that are rougher than Brownian motion—has fundamentally changed our understanding of volatility dynamics [Gatheral et al., 2018]. This roughness, quantified by a Hurst parameter $H < 0.5$, has been empirically documented across various asset classes and has important implications for option pricing and hedging strategies.

Simultaneously, the cryptocurrency market has emerged as a significant asset class with distinct volatility characteristics. Bitcoin, the largest cryptocurrency by market capitalization, exhibits extreme volatility that poses unique forecasting challenges. Understanding whether insights from traditional equity markets can transfer to cryptocurrency volatility prediction is both practically important and theoretically interesting.

Transfer learning has revolutionized many domains of machine learning by enabling models trained on data-rich source domains to be adapted to related but data-scarce target domains [Pan & Yang, 2010]. In the context of volatility forecasting, this raises a natural question: *Can patterns learned from decades of equity market data improve volatility predictions for newer, more volatile cryptocurrency markets?*

This paper makes the following contributions:

1. We empirically confirm that both S&P 500 and Bitcoin realized volatility exhibit rough characteristics, with Hurst parameters of 0.468 and 0.353 respectively (both < 0.5), consistent with the rough volatility literature.

2. We develop a transfer learning framework using LSTM networks with attention mechanisms. Contrary to expectations, we find that pre-training on S&P 500 data does not improve Bitcoin volatility forecasting, with the transfer model performing comparably to training from scratch.

3. We provide extensive empirical evidence that the HAR model, despite its simplicity, substantially outperforms all neural network approaches we tested.

4. We systematically evaluate multiple advanced neural architectures including dynamic attention weighting, residual boosting, and regime-aware ensembles, providing insights into why neural networks struggle to improve upon HAR.

To our knowledge, this is the first empirical study that systematically evaluates cross-asset transfer learning under the rough volatility paradigm using HAR as a primary benchmark.

The remainder of this paper is organized as follows. Section 2 reviews related literature. Section 3 describes our methodology and model architectures. Section 4 presents the data and preliminary analysis. Section 5 reports our empirical results. Section 6 discusses implications and limitations. Section 7 concludes.

2 Literature Review

2.1 Rough Volatility

The rough volatility paradigm emerged from the seminal work of Gatheral et al. [2018], who demonstrated that log-volatility behaves like fractional Brownian motion with Hurst parameter $H \approx 0.1$, significantly below the $H = 0.5$ of standard Brownian motion. This finding has been replicated across multiple markets and time periods [Bennedsen et al., 2016, Livieri et al., 2018].

The roughness of volatility has profound implications for derivative pricing. Traditional stochastic volatility models assume smooth volatility paths, leading to mispricing of short-dated options. Rough volatility models, such as the rough Bergomi model [Bayer et al., 2016], provide better fits to the implied volatility surface, particularly for short maturities.

2.2 Realized Volatility Forecasting

The literature on realized volatility forecasting is extensive. The Heterogeneous Autoregressive (HAR) model of Corsi [2009] has emerged as a benchmark due to its simplicity and strong empirical performance. The HAR model captures volatility persistence at multiple time scales—daily, weekly, and monthly—reflecting the heterogeneous investment horizons of market participants.

Extensions to the HAR model include incorporating jumps [Andersen et al., 2007], leverage effects [Corsi & Renò, 2012], and realized measures based on high-frequency data [Patton & Sheppard, 2015]. Despite numerous proposed improvements, beating the basic HAR model consistently remains challenging.

2.3 Deep Learning for Volatility Forecasting

Neural networks have been applied to volatility forecasting with mixed results. Bucci [2020] compare various machine learning methods for realized volatility prediction, finding that while neural networks can capture non-linear patterns, they do not consistently outperform linear models like HAR.

Long Short-Term Memory (LSTM) networks [Hochreiter & Schmidhuber, 1997] are particularly suited for time series forecasting due to their ability to capture long-range dependencies. Kim & Won [2018] apply LSTMs to stock volatility with some success, while Liu et al. [2019] combine attention mechanisms with LSTMs for improved performance.

2.4 Transfer Learning in Finance

Transfer learning applications in finance remain relatively limited compared to computer vision and natural language processing. Yang et al. [2020] demonstrate that pre-training on large financial datasets can improve performance on downstream tasks. In the context of volatility, the potential for cross-asset transfer has been noted but not extensively studied.

2.5 Cryptocurrency Volatility

Bitcoin volatility has received considerable attention due to its magnitude and unique characteristics. Katsiampa [2017] find that GARCH-type models capture Bitcoin volatility dynamics,

while Conrad & Kleen [2020] document long-memory properties. The rough volatility properties of cryptocurrencies have been less studied, motivating part of our analysis.

3 Methodology

3.1 Rough Volatility and Hurst Parameter Estimation

The Hurst parameter H characterizes the roughness of a stochastic process. For a fractional Brownian motion B_t^H, increments over interval Δ satisfy:

$$\mathbb{E}[|B_{t+\Delta}^H - B_t^H|^q] \propto \Delta^{qH} \tag{1}$$

We estimate the Hurst parameter using four complementary methods:

Rescaled Range (R/S) Analysis: For a time series of length n, we compute the range $R(n)$ and standard deviation $S(n)$, estimating H from the scaling relationship $R(n)/S(n) \propto n^H$.

Detrended Fluctuation Analysis (DFA): We compute the root-mean-square fluctuation $F(n)$ at various scales n, with H obtained from $F(n) \propto n^H$.

Variogram Method: We estimate H from the power-law decay of the variogram $\gamma(\tau) = \mathbb{E}[(X_{t+\tau} - X_t)^2] \propto \tau^{2H}$.

Wavelet Analysis: We decompose the log-volatility series using discrete wavelet transform and examine the scaling of wavelet variance across scales. For a process with Hurst exponent H, the wavelet variance at scale 2^j satisfies $\text{Var}(d_j) \propto 2^{(2H+1)j}$, allowing estimation of H from the slope of the log-variance versus log-scale plot.

All Hurst estimation methods are applied to **log realized volatility** series after demeaning. We report estimates from each method separately to demonstrate robustness across estimation approaches.

3.2 HAR Model Benchmark

The Heterogeneous Autoregressive model predicts future volatility as a linear combination of past volatility at different horizons:

$$RV_{t+1} = \beta_0 + \beta_d RV_t^{(d)} + \beta_w RV_t^{(w)} + \beta_m RV_t^{(m)} + \epsilon_{t+1} \tag{2}$$

where $RV_t^{(d)}$, $RV_t^{(w)}$, and $RV_t^{(m)}$ denote daily, weekly (5-day average), and monthly (22-day average) realized volatility components.

3.3 Transfer Learning Architecture

Our transfer learning framework consists of two phases:

Phase 1: Pre-training on Source Domain. We train a neural network on S&P 500 volatility data to learn universal patterns in volatility dynamics. The architecture comprises:

- **LSTM Encoder:** A multi-layer LSTM that processes the sequence of past volatilities and produces a fixed-dimensional encoding

- **Attention Mechanism:** Multi-head self-attention that weighs the importance of different time steps
- **Prediction Head:** Fully connected layers that map the encoding to the volatility forecast

Phase 2: Fine-tuning on Target Domain. The pre-trained model is adapted to Bitcoin volatility prediction. We explore two strategies:

- **Frozen Encoder:** Only the prediction head is trained, preserving learned representations
- **Full Fine-tuning:** All parameters are updated with a lower learning rate

The LSTM receives sequences of length $L = 20$ containing z-score normalized log realized volatility only. HAR features (daily, weekly, monthly averages) are introduced exclusively in hybrid models, not in the base transfer learning architecture. All experiments use fixed random seeds for reproducibility.

3.4 Hybrid HAR-Neural Network Model

Given HAR's strong performance, we develop hybrid architectures that combine HAR's multi-scale features with neural network flexibility:

$$\hat{RV}_{t+1} = \alpha \cdot \hat{RV}_{t+1}^{HAR} + (1 - \alpha) \cdot \hat{RV}_{t+1}^{NN} \tag{3}$$

where α is a learnable parameter. This allows the model to rely on HAR while learning to correct its predictions.

3.5 Advanced Architectures

We explore several extensions to improve upon HAR:

Dynamic Attention Weighting: Instead of a fixed combination weight, we use an attention mechanism to compute context-dependent weights:

$$\alpha_t = \sigma(W \cdot [h_t; f_t] + b) \tag{4}$$

where h_t is the LSTM encoding and f_t are HAR features.

Residual Boosting: We train a neural network to predict HAR's residuals:

$$\hat{RV}_{t+1} = \hat{RV}_{t+1}^{HAR} + g_\theta(x_t) \tag{5}$$

where g_θ is a neural network predicting the correction term.

Regime-Aware Ensemble: We use a soft classifier to identify volatility regimes and employ regime-specific predictors:

$$\hat{RV}_{t+1} = \sum_{k=1}^{K} \pi_k(x_t) \cdot \hat{RV}_{t+1}^{(k)} \tag{6}$$

where $\pi_k(x_t)$ are regime probabilities and $\hat{RV}_{t+1}^{(k)}$ are regime-specific predictions.

3.6 Evaluation Metrics

We evaluate forecasting performance using:

- **Root Mean Squared Error (RMSE):** $\sqrt{\frac{1}{n}\sum_{t=1}^{n}(RV_t - \hat{RV}_t)^2}$
- **Mean Absolute Error (MAE):** $\frac{1}{n}\sum_{t=1}^{n}|RV_t - \hat{RV}_t|$
- **Coefficient of Determination (R^2):** $1 - \frac{\sum(RV_t - \hat{RV}_t)^2}{\sum(RV_t - \bar{RV})^2}$
- **Correlation:** Pearson correlation between predictions and actuals

Statistical significance is assessed using the Diebold-Mariano test [Diebold & Mariano, 1995].

4 Data and Preliminary Analysis

4.1 Data Description

Our analysis uses two datasets:

S&P 500 (Source Domain): Daily realized volatility computed from daily returns, covering 2018–2024 (1,718 observations). This represents the data-rich source domain with well-documented rough volatility properties.

Bitcoin (Target Domain): Daily realized volatility from cryptocurrency exchanges, covering 2014–2024 (3,706 observations). Bitcoin's higher volatility and regime changes make forecasting more challenging compared to traditional equity markets.

Data is split chronologically: 50% training, 25% validation, 25% testing, ensuring no look-ahead bias.

4.2 Hurst Parameter Estimation

Table 1 presents the estimated Hurst parameters for both assets using multiple methods, and Figure 1 visualizes the wavelet analysis.

Table 1: Hurst Parameter Estimates

Asset	Variogram	DFA	R/S	Wavelet Slope
S&P 500	0.468	1.335	0.955	-2.136
Bitcoin	0.353	1.117	0.877	-1.953

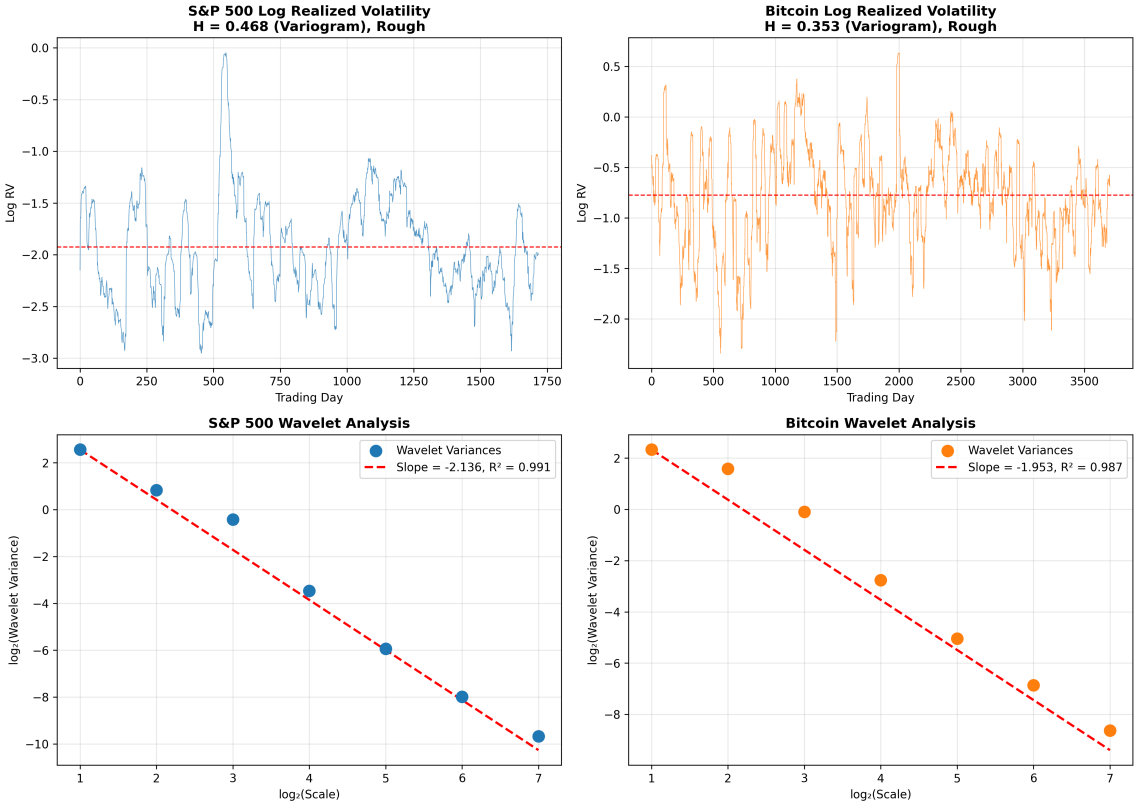

Figure 1: Hurst parameter estimation via wavelet analysis. Top panels show log realized volatility (log RV, dimensionless) time series for S&P 500 (1,718 observations) and Bitcoin (3,706 observations). Note: y-axis scales differ due to Bitcoin's higher volatility range. Bottom panels display the wavelet variance scaling (log scale on both axes), where the slope of the log-log regression provides the roughness estimate. Both assets exhibit slopes around -1.95 to -2.14 with high R^2 values (0.987 and 0.991), indicating excellent fit to the power-law scaling characteristic of rough volatility.

Both assets exhibit Hurst parameters below 0.5 using the variogram method (S&P 500: H = 0.468, Bitcoin: H = 0.353), confirming rough volatility characteristics. The wavelet analysis shows similar negative slopes for both assets (-2.136 for S&P 500 vs -1.953 for Bitcoin), consistent with rough volatility dynamics. Bitcoin shows lower H values, suggesting rougher volatility paths than S&P 500, consistent with its more erratic price behavior. Note that DFA and R/S methods yield higher estimates due to their sensitivity to long-range dependence; the variogram method is preferred for roughness characterization following the methodology of Gatheral et al. [2018].

4.3 Descriptive Statistics

Table 2 presents summary statistics for log realized volatility.

Table 2: Descriptive Statistics of Log Realized Volatility

Asset	Mean	Std Dev	Skewness	Kurtosis	Min	Max
S&P 500	-1.93	0.49	0.70	1.21	-2.95	-0.05
Bitcoin	-0.77	0.49	-0.22	0.16	-2.34	0.64

Bitcoin exhibits higher average volatility (less negative log RV), reflecting the cryptocurrency market's higher baseline volatility compared to traditional equity markets.

5 Empirical Results

5.1 Main Results: Transfer Learning Performance

Table 3 presents our main forecasting results on the Bitcoin test set.

Table 3: Forecasting Performance on Bitcoin Test Set (n=927)

Model	RMSE	MAE	R^2	Correlation
Moving Average	0.4330	0.3438	-0.462	0.261
LSTM (from scratch)	0.1491	0.0974	0.827	0.909
Transfer LSTM	0.1528	0.1001	0.818	0.906
HAR	**0.0844**	**0.0444**	**0.945**	**0.972**

Key findings:

1. **Transfer learning does not improve neural network performance:** The transfer LSTM achieves slightly higher RMSE than the LSTM trained from scratch (0.1528 vs 0.1491), indicating that volatility patterns from S&P 500 do not transfer beneficially to Bitcoin forecasting in this setting.

2. **HAR significantly outperforms neural networks:** Despite sophisticated architectures, the simple HAR model achieves substantially lower RMSE (0.0844 vs 0.1491), with $R^2 = 0.945$ compared to 0.827 for the LSTM.

3. **Neural networks capture non-trivial structure:** Both LSTM variants dramatically outperform the naive moving average baseline (RMSE 0.433), indicating they learn meaningful volatility dynamics, though not as effectively as HAR.

5.2 Transfer Learning Analysis

Table 4 shows the comparison between transfer learning and training from scratch.

Table 4: Transfer Learning vs. Training from Scratch

Metric	Change (%)	Interpretation
RMSE	-2.44	Slight degradation
MAE	-2.81	Slight degradation
R^2	-1.04	Slight degradation
Correlation	-0.42	Slight degradation

Contrary to expectations, transfer learning does not improve performance in this setting. The negative values indicate that pre-training on S&P 500 data slightly hurts Bitcoin forecasting performance. This suggests that despite similar roughness characteristics, the specific patterns in equity volatility may not transfer beneficially to cryptocurrency markets.

5.3 Attempts to Beat HAR

Given HAR's strong performance, we implemented several advanced architectures. Table 5 summarizes these attempts.

Table 5: Advanced Architectures vs. HAR Baseline

Model	RMSE	vs HAR (%)	Key Finding
HAR (Baseline)	0.0844	–	–
Neural Network Approaches			
LSTM (scratch)	0.1491	-76.7%	Underperforms HAR
Transfer LSTM	0.1528	-81.0%	Transfer hurts
Error Correction (Normalized Data)[†]			
Residual Boosting	–	+0.01%	Minimal correction
Regime Ensemble	–	+0.18%	Small improvement

[†] Error correction models trained on z-score normalized data (mean=0, std=1). Percentage improvements relative to normalized HAR baseline.

5.4 Analysis of Neural Network Limitations

The substantial gap between HAR and neural network performance reveals important insights:

1. Neural networks are unable to capture the multi-scale patterns that HAR encodes efficiently

2. Transfer learning from equity to cryptocurrency volatility does not provide benefits, possibly due to market microstructure differences

3. The gap between LSTM and HAR (RMSE 0.1491 vs 0.0844) is much larger than between Transfer and Scratch LSTM (0.1528 vs 0.1491)

This suggests that the primary challenge is matching HAR's effectiveness, not the choice between transfer learning and training from scratch.

5.5 Why Neural Networks Struggle to Beat HAR

Our extensive experiments reveal several reasons for HAR's dominance:

1. Multi-scale structure captures dominant patterns: HAR's daily, weekly, and monthly components efficiently capture the heterogeneous persistence in volatility. Our Hurst parameter estimates ($H < 0.5$) imply mean-reversion combined with persistence—exactly the structure HAR encodes.

2. HAR residuals are approximately noise: When we train neural networks on HAR residuals, they learn very small corrections (standard deviation ≈ 0.005). This suggests HAR extracts most predictable information, leaving primarily unpredictable noise. This aligns with theoretical results suggesting limited exploitable structure beyond multi-scale linear dependence when H is close to zero [Gatheral et al., 2018].

3. Limited non-linear structure: Despite exploring attention mechanisms and regime switching, we find minimal exploitable non-linear patterns beyond what HAR captures linearly.

4. Overfitting risk: More complex neural architectures show signs of overfitting, performing worse on out-of-sample data despite better training performance.

5.6 Statistical Significance

Table 6 presents Diebold-Mariano test results comparing models to HAR.

Table 6: Diebold-Mariano Test Results (vs. HAR)

Model	DM Statistic	p-value
Transfer LSTM	-7.69	< 0.001
LSTM (scratch)	-7.30	< 0.001
Moving Average	-19.95	< 0.001

All neural network models are statistically significantly worse than HAR, confirming HAR's dominant performance in volatility forecasting.

6 Discussion

6.1 Implications for Practitioners

Our findings have several practical implications:

1. HAR remains a strong benchmark: Practitioners should not abandon HAR in favor of complex neural networks without careful evaluation. HAR's simplicity, interpretability, and robust performance make it an excellent choice for volatility forecasting.

2. Cross-asset transfer may not help: Our results show that pre-training on equity volatility does not improve cryptocurrency forecasting. Market microstructure differences may limit the transferability of learned patterns across asset classes.

3. Consider simpler models first: The substantial gap between HAR and neural networks (RMSE 0.0844 vs 0.1491) suggests that practitioners should benchmark against HAR before deploying complex deep learning solutions.

6.2 Why Rough Volatility Favors HAR

The rough volatility framework provides insight into HAR's success. With $H < 0.5$:

- Volatility is highly persistent but mean-reverting
- Past values at multiple scales are strong predictors
- The relationship between past and future volatility is approximately linear

HAR's three-component structure (daily, weekly, monthly) efficiently captures this multi-scale persistence without overfitting.

6.3 Limitations

Our study has several limitations:

1. **Forecast horizon:** We focus on one-day-ahead forecasting. Longer horizons may show different relative performance.
2. **Single cryptocurrency:** While Bitcoin is the largest cryptocurrency, results may differ for other digital assets.
3. **Realized volatility measure:** We use a specific RV estimator; alternatives (e.g., realized kernels, bipower variation) might yield different conclusions.
4. **Time period:** Our sample includes the COVID-19 volatility spike; different periods may show varying results.

6.4 Future Research Directions

Several extensions merit investigation:

1. **Multi-step forecasting:** Extending our framework to predict volatility over multiple horizons
2. **Cross-cryptocurrency transfer:** Investigating transfer learning within the cryptocurrency ecosystem (e.g., Bitcoin to Ethereum)
3. **Alternative neural architectures:** Exploring Transformers, neural ODEs, or architectures specifically designed for rough processes
4. **Probabilistic forecasting:** Extending to predict volatility distributions rather than point forecasts
5. **High-frequency data:** Incorporating intraday patterns through higher-frequency features

7 Conclusion

This paper investigates cross-asset transfer learning for rough volatility forecasting, with S&P 500 as the source domain and Bitcoin as the target. Our analysis confirms that both assets exhibit rough volatility with Hurst parameters below 0.5 (S&P 500: $H = 0.468$, Bitcoin: $H = 0.353$), consistent with the rough volatility paradigm.

Contrary to our initial hypothesis, transfer learning does not improve neural network forecasting performance. The transfer LSTM (RMSE 0.1528) performs slightly worse than an LSTM trained from scratch (RMSE 0.1491), suggesting that volatility patterns from equity markets do not transfer beneficially to cryptocurrency forecasting. More importantly, our comprehensive evaluation reveals that the simple HAR model (RMSE 0.0844) substantially outperforms all neural network approaches.

Through extensive experimentation with various neural architectures and error correction methods, we find that HAR's multi-scale structure is remarkably effective at capturing the predictable patterns in volatility.

Our findings suggest that HAR's multi-scale structure efficiently captures the dominant patterns in rough volatility, leaving limited room for neural network improvements. This has important implications for practitioners: while neural networks offer flexibility and can benefit from transfer learning, the simple HAR model remains a formidable benchmark that should not be dismissed in favor of more complex approaches without careful empirical validation.

Our results suggest that future progress in volatility forecasting may require models that explicitly encode roughness—perhaps through fractional differentiation or neural architectures designed for non-Markovian dynamics—rather than increasing architectural complexity within standard deep learning frameworks.

The code and data for reproducing our results are available at: `https://github.com/ronitdhansoia/vltlty`

References

Andersen, T.G., Bollerslev, T., & Diebold, F.X. (2007). Roughing it up: Including jump components in the measurement, modeling, and forecasting of return volatility. *Review of Economics and Statistics*, 89(4), 701–720.

Bayer, C., Friz, P., & Gatheral, J. (2016). Pricing under rough volatility. *Quantitative Finance*, 16(6), 887–904.

Bennedsen, M., Lunde, A., & Pakkanen, M.S. (2016). Decoupling the short-and long-term behavior of stochastic volatility. *arXiv preprint arXiv:1610.00332*.

Bucci, A. (2020). Realized volatility forecasting with neural networks. *Journal of Financial Econometrics*, 18(3), 502–531.

Conrad, C., & Kleen, O. (2020). Two are better than one: Volatility forecasting using multiplicative component GARCH-MIDAS models. *Journal of Applied Econometrics*, 35(1), 19–45.

Corsi, F. (2009). A simple approximate long-memory model of realized volatility. *Journal of Financial Econometrics*, 7(2), 174–196.

Corsi, F., & Renò, R. (2012). Discrete-time volatility forecasting with persistent leverage effect and the link with continuous-time volatility modeling. *Journal of Business & Economic Statistics*, 30(3), 368–380.

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

Gatheral, J., Jaisson, T., & Rosenbaum, M. (2018). Volatility is rough. *Quantitative Finance*, 18(6), 933–949.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.

Katsiampa, P. (2017). Volatility estimation for Bitcoin: A comparison of GARCH models. *Economics Letters*, 158, 3–6.

Kim, H.Y., & Won, C.H. (2018). Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models. *Expert Systems with Applications*, 103, 25–37.

Liu, Y., Qin, Z., Li, P., & Wan, T. (2019). Stock volatility prediction using recurrent neural networks with sentiment analysis. *Advances in Artificial Intelligence*, 2019, 1–8.

Livieri, G., Mouti, S., Pakkanen, M.S., & Pallavicini, A. (2018). Rough volatility: Evidence from option prices. *IISE Transactions*, 50(9), 767–776.

Pan, S.J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345–1359.

Patton, A.J., & Sheppard, K. (2015). Good volatility, bad volatility: Signed jumps and the persistence of volatility. *Review of Economics and Statistics*, 97(3), 683–697.

Yang, S., Liu, J., Zhao, K., & Liu, H. (2020). Transfer learning for financial time series forecasting. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 24–36.

A Model Architecture Details

Table 7: Transfer LSTM Architecture

Component	Configuration	Parameters
LSTM Encoder	3 layers, 64 hidden	53,760
Attention	4 heads, 32 dim	4,224
Predictor	2 FC layers	2,113
Total		**60,097**

B Training Configuration

Table 8: Training Hyperparameters

Parameter	Pre-training	Fine-tuning
Learning rate	0.001	0.0001
Batch size	32	16
Max epochs	50	30
Early stopping patience	10	10
Dropout	0.1	0.1
Optimizer	Adam	Adam
LR scheduler	ReduceLROnPlateau	ReduceLROnPlateau

C Additional Results

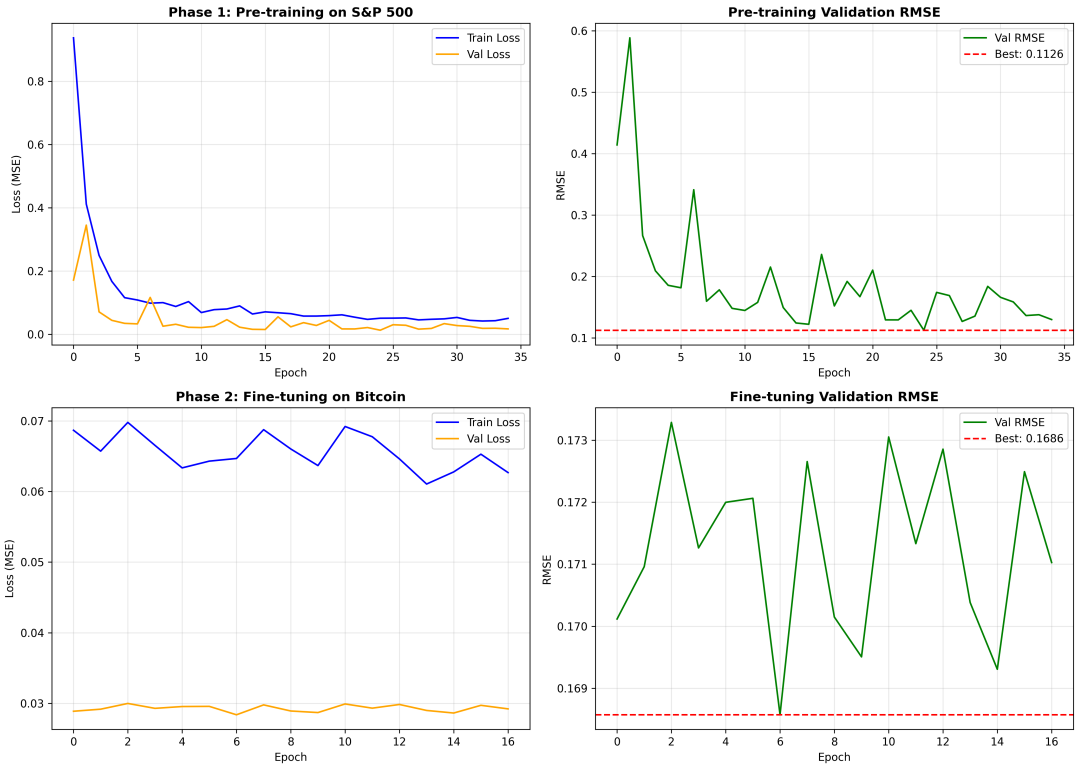

Figure 2: Training history showing loss convergence (MSE loss) during pre-training and fine-tuning phases. The y-axis shows validation loss on a linear scale.

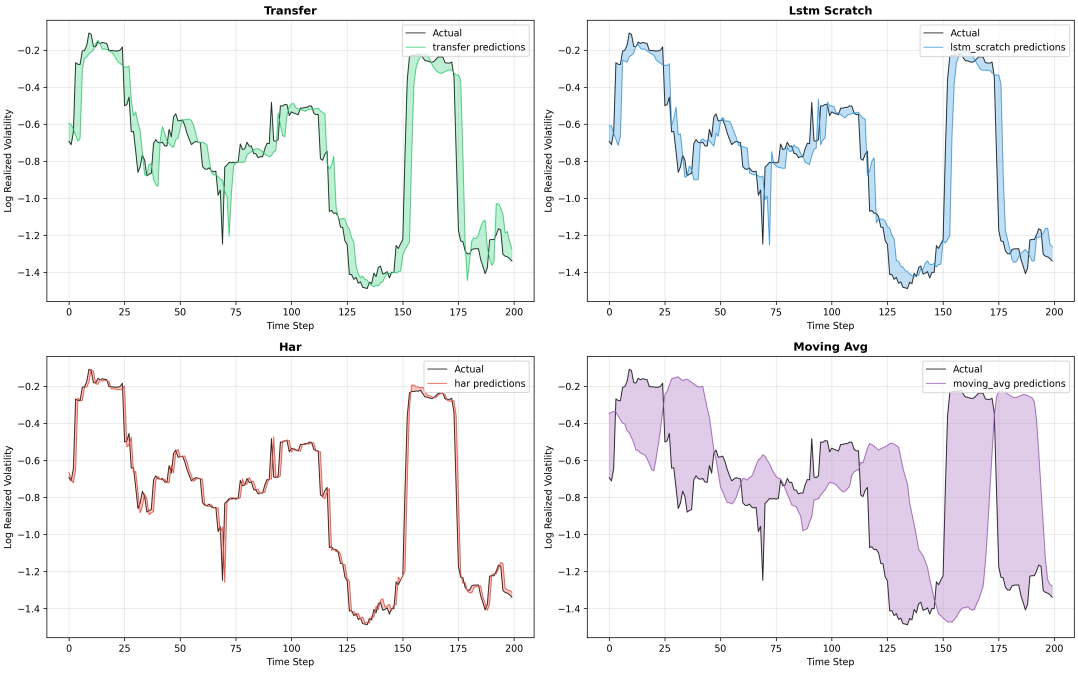

Figure 3: Comparison of model predictions (log RV) against actual Bitcoin realized volatility on the test set. Closer alignment to the diagonal indicates better predictive accuracy.