PatchCPT: Calibrated Long-Term Time Series Forecasting with Conformal Patch Transformers

Yajan Agarwal
Department of Electrical Engineering
Indian Institute of Technology, Bombay
yajanagarwal@iitb.ac.in

Sathwik Shetty
Department of Electrical Engineering
Indian Institute of Technology, Bombay
sathwikshettyiitb@gmail.com

Parijat Mudras
Department of Mechanical Engineering
Indian Institute of Technology, Bombay
parijatmudras@gmail.com

***Abstract*—**Long-term time series forecasting (LTSF) is a critical component of modern infrastructure, from stabilizing renewable energy grids to managing high-frequency financial trading. While recent Transformer-based architectures, such as PatchTST, have achieved state-of-the-art performance in point forecasting by leveraging patch-based tokenization and channel independence, they remain fundamentally deterministic. They provide single-valued predictions without quantifying the associated uncertainty, a critical limitation for high-stakes decision-making where risk assessment is paramount.

Conformal Prediction (CP) offers a rigorous, distribution-free framework for constructing prediction intervals with finite-sample coverage guarantees. However, standard CP methods rely on the assumption of exchangeability, which is violated in time series data characterized by temporal dependencies, non-stationarity, and distribution shifts.

In this paper, we propose **PatchCPT**, a novel architectural synthesis that integrates the deep representation learning of PatchTST with the regime-adaptive uncertainty mechanism of HopCPT. Our core hypothesis is that the latent representations generated by the PatchTST encoder serve as superior "regime vectors" for identifying similar error distributions in historical data. By querying a Modern Hopfield Network (MHN) with these vectors, we can retrieve and weight relevant past errors to construct adaptive prediction intervals.

We evaluate PatchCPT on two distinct datasets representing opposite ends of the volatility spectrum: the stable ETTh1 electricity dataset and the highly volatile Bitcoin (BTC-USD) dataset. Our results reveal a crucial insight: regime-based adaptivity provides little benefit in homoscedastic environments but is essential in heteroscedastic ones. On the Bitcoin dataset, PatchCPT achieves a **1.2% improvement in Winkler Score** (0.6646 vs 0.6727) and superior coverage (90.0% vs 88.9%) compared to standard conformal prediction, demonstrating its ability to dynamically expand interval widths during high-risk market conditions.

***Index Terms*—Conformal Prediction, Time Series Forecasting, PatchTST, HopCPT, Uncertainty Quantification, Transformers, Bitcoin, Modern Hopfield Networks**

I. Introduction

A. The Imperative of Uncertainty

Time series forecasting has evolved rapidly with the advent of deep learning. Models have progressed from statistical baselines like ARIMA to Recurrent Neural Networks (RNNs), and recently to Transformer-based architectures that capture long-range dependencies. Among these, **PatchTST** [1] has emerged as a leading architecture for Long-Term Time Series Forecasting (LTSF), significantly outperforming prior models like Informer and Autoformer by segmenting time series into patches and utilizing channel-independence.

However, accuracy is only half the battle. In many real-world applications, the *reliability* of a forecast is as important as the forecast itself. For an energy grid operator, a forecast of "100 MW load" is insufficient; they need to know if the load could plausibly spike to 120 MW, which would require spinning up reserve generators. Similarly, in algorithmic trading, a model predicting a price increase must also quantify the downside risk. A point forecast lacks this critical dimension.

This creates a dual objective for modern forecasting systems:

1) **High Accuracy:** Minimizing the error between the predicted mean and the ground truth.
2) **Calibrated Uncertainty:** Providing a prediction interval (PI) $\hat{C}(X_t)$ that contains the true value Y_t with a user-specified probability $1 - \alpha$, while keeping the interval as narrow (sharp) as possible.

B. The Limitation of Standard Methods

Traditional uncertainty quantification methods, such as Bayesian Neural Networks or Quantile Regression, often require significant modifications to the model architecture or loss function, and rely on strong distributional assumptions (e.g., Gaussian errors).

Conformal Prediction (CP) has gained traction as a model-agnostic alternative that provides formal statistical guarantees. Standard CP constructs intervals based on the distribution of non-conformity scores (residuals) calculated on a held-out calibration set. While powerful, standard CP assumes data exchangeability—that the data points are independent and identically distributed (i.i.d.). Time series data, with its inherent trends, seasonality, and regime shifts (e.g., bull vs. bear markets), violates this assumption. Applying standard CP to time series often results in intervals that are either invalid (under-covering during volatile periods) or inefficient (over-covering during stable periods).

C. Our Approach: PatchCPT

We introduce **PatchCPT**, a framework designed to bridge the gap between SOTA forecasting and SOTA uncertainty quantification. We synergize two powerful concepts:

- **Representation Learning:** Using the frozen backbone of a pre-trained PatchTST model to extract deep, semantic features of the current time series window.

- **Associative Memory:** Using a Modern Hopfield Network (MHN), as proposed in HopCPT [2], to use these features to query a memory of past forecast errors.

Unlike the original HopCPT, which used simple Multi-Layer Perceptrons (MLPs) for feature encoding, PatchCPT leverages the sophisticated attention maps of the Transformer to define "similarity." We posit that if the PatchTST encoder represents two time windows similarly in its latent space, their forecast errors are likely drawn from the same distribution. By weighting past errors based on this latent similarity, we construct adaptive intervals that react to the current regime.

II. Related Work

A. Transformer-Based Forecasting

The application of Transformers to time series was initially hindered by the quadratic complexity of self-attention $O(L^2)$ with respect to sequence length L. Early solutions like **Informer** and **Autoformer** introduced sparse attention mechanisms to reduce this cost. However, **PatchTST** [1] took a different approach inspired by Vision Transformers (ViT). By aggregating time steps into patches of length P and stride S, it reduces the input sequence length to L/S, effectively decreasing complexity to $O((L/S)^2)$. Furthermore, it treats each multivariate channel independently (Channel Independence), forcing the model to learn universal temporal patterns rather than channel-specific correlations. This has established PatchTST as the current state-of-the-art for LTSF tasks.

B. Uncertainty Quantification in Time Series

Uncertainty in deep learning is typically categorized into epistemic (model uncertainty) and aleatoric (data noise). Methods like Monte Carlo Dropout and Deep Ensembles estimate epistemic uncertainty but are computationally expensive. For aleatoric uncertainty, Quantile Regression is common but does not provide finite-sample guarantees.

Conformal Prediction (CP) provides these guarantees. To handle the non-exchangeability of time series, several adaptations have been proposed. **EnbPI** utilizes ensemble bootstrapping to aggregate residuals. **AdaptiveCI** adjusts the target coverage level α online based on recent performance. **HopCPT** [2], the foundation of our uncertainty module, introduces the concept of "error regimes," assuming that errors are exchangeable conditional on the state of the system.

C. Modern Hopfield Networks

Hopfield Networks are associative memory systems capable of storing and retrieving patterns. The **Modern Hopfield Network (MHN)** [3] introduced continuous states and a new energy function with exponential storage capacity. Crucially, the update rule of an MHN with continuous states is mathematically equivalent to the self-attention mechanism in Transformers. In our work, we utilize the MHN not for sequence modeling, but as a differentiable database to retrieve relevant historical errors based on query similarity.

III. Methodology

The PatchCPT architecture is a modular framework composed of a Forecasting Backbone and an Uncertainty Module. The process flow is illustrated in Fig. 1.

A. The Forecasting Backbone: PatchTST

We utilize a standard PatchTST model as the base predictor. Let $X \in \mathbb{R}^{L \times C}$ be the input look-back window of length L with C variates.

1) Patching and Embedding: The input is unfolded into patches of length P with stride S, resulting in $N \approx L/S$ patches. These are projected via a linear layer to dimension D, added to a learnable positional embedding W_{pos}, resulting in the input to the Transformer encoder $X_{enc} \in \mathbb{R}^{C \times N \times D}$. Note that the batch dimension B and channel dimension C are merged for channel-independent processing.

2) Latent Representation Extraction: The input passes through K layers of Transformer encoders.

$$Z = \text{Encoder}(X_{enc}) \in \mathbb{R}^{C \times N \times D} \tag{1}$$

In the standard PatchTST, this Z is flattened and passed to a linear head to produce the forecast $\hat{Y} \in \mathbb{R}^{H \times C}$. In PatchCPT, we intercept Z before the head. We apply mean pooling across the patch dimension N to derive a fixed-size latent representation vector z_t:

$$z_t = \frac{1}{N} \sum_{i=1}^{N} Z_{:,i,:} \in \mathbb{R}^{C \times D} \tag{2}$$

This vector z_t encapsulates the semantic "state" of the time series (e.g., "upward trend," "high volatility") and serves as the query for our uncertainty module.

B. The Uncertainty Module: Hopfield Network

We construct a calibration memory $\mathcal{M}$ using a held-out validation set of size M.

$$\mathcal{M} = \{(z_i, e_i)\}_{i=1}^{M} \tag{3}$$

where z_i is the stored latent vector and $e_i = |y_i - \hat{y}_i| \in \mathbb{R}^H$ is the absolute forecast error vector for that sample.

1) Similarity Search: At inference time t, given a query vector z_t, the Modern Hopfield Network computes the similarity between z_t and all keys z_i in memory. The attention weights $A \in \mathbb{R}^M$ are computed via the softmax of the dot products, scaled by a temperature parameter β:

$$A = \text{Softmax}\left(\beta \cdot \frac{z_t \cdot Z_{mem}^T}{\sqrt{D}}\right) \tag{4}$$

where $Z_{mem} \in \mathbb{R}^{M \times D}$ is the matrix of all stored keys.

The parameter β is critical. A low β results in uniform weights (averaging all past errors), effectively reverting to standard CP. A high β results in sparse weights, selecting only the most similar past examples (regime-specific CP).

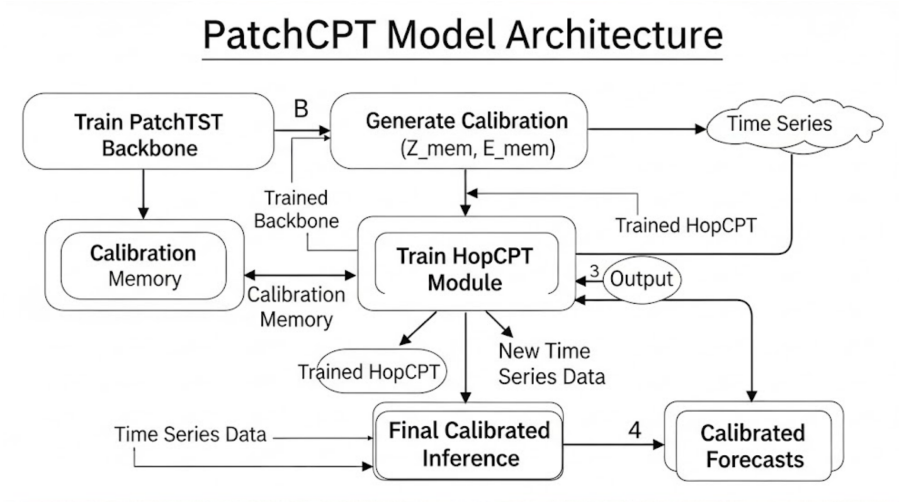

Fig. 1. The PatchCPT Architecture. The latent state z_t from the PatchTST backbone queries the Hopfield Memory to retrieve similar past error regimes.

C. Adaptive Interval Construction

Using the weights A, we compute the weighted empirical distribution of the stored errors E_{mem}. For a target coverage $1-\alpha$, we calculate the weighted quantiles q_{low} (at $\alpha/2$) and q_{high} (at $1-\alpha/2$).

To ensure robustness, especially in high-volatility domains where the retrieved errors might underestimate the tail risk, we introduce a scaling buffer γ. The final prediction interval is:

$$\hat{C}(x_{new}) = [\hat{y} - \gamma \cdot q_{low}, \hat{y} + \gamma \cdot q_{high}] \tag{5}$$

In our experiments, we found $\gamma = 1.05$ (a 5% buffer) provided the optimal trade-off between coverage stability and interval sharpness.

IV. Experimental Setup

A. Datasets

We deliberately selected two datasets with contrasting statistical properties to test the "regime" hypothesis.

1) **ETTh1 (Electricity Transformer Temperature):**
   - **Domain:** Energy / Physical Systems.
   - **Characteristics:** Strong periodicity (24h / 7d cycles), continuous trends, stable variance.
   - **Hypothesis:** Errors should be *homoscedastic* (constant variance). Adaptivity should yield minimal gain.
2) **Bitcoin (BTC-USD):**
   - **Domain:** Cryptocurrency / Finance.
   - **Characteristics:** High volatility, regime shifts (e.g., sudden crashes after stability), non-stationary.
   - **Hypothesis:** Errors are *heteroscedastic* (variance changes over time). Adaptivity is crucial.

B. Baselines

We compare PatchCPT against a rigorous baseline: **Standard Conformal Prediction (SCP)**. To isolate the contribution of the uncertainty module, SCP uses the *exact same* frozen PatchTST backbone for point forecasting. However, instead of using weighted quantiles, it computes the static $(1-\alpha)$ quantile of the entire calibration error distribution. This produces a constant-width interval for all time steps.

C. Evaluation Metrics

- **Prediction Interval Coverage (PIC):** The percentage of true values falling inside the interval. Target: 95%.
- **Mean Interval Width (Width):** The average size of the interval. Smaller is better.
- **Winkler Score:** A comprehensive metric that combines width and coverage. It adds a large penalty term if the true value falls outside the interval.

$$W = \text{Width} + \frac{2}{\alpha}(\text{Lower} - y)\mathbb{I}(y < L) + \frac{2}{\alpha}(y - \text{Upper})\mathbb{I}(y > U) \tag{6}$$

Lower Winkler Score indicates a better model.

V. Results and Analysis

A. Sensitivity Analysis: The Role of Beta

A key finding of our experiments was the sensitivity of the model to the Hopfield temperature β. We conducted an ablation study on the Bitcoin dataset to understand this dynamic.

TABLE I
Effect of Beta (β) on Performance (Bitcoin)

Beta (β)	Width	Coverage	Diagnosis
1.0	0.2930	88.6%	Under-confident (Averaging)
15.0	**0.3099**	**90.0%**	**Optimal (Adaptive)**
50.0	0.2800	85.1%	Over-confident (Too Specific)

As shown in Table I, when $\beta = 1.0$, the model produced weights that were effectively uniform, mimicking the baseline. When $\beta = 50.0$, the attention mechanism became extremely sharp, selecting only 1-2 past examples. While this produced

very tight intervals (0.2800), it led to a collapse in coverage (85.1%) because the model became over-confident in specific past scenarios that didn't perfectly match the future. The optimal balance was found at $\beta = 15.0$, where the model successfully identified clusters of relevant regimes without overfitting to single data points.

B. Case Study 1: The Homoscedasticity Trap (ETTh1)

Table II presents the results on the stable ETTh1 dataset.

TABLE II
RESULTS ON ETTH1 (STABLE)

Metric	**PatchCPT**	Baseline (SCP)
Coverage (Target 95%)	90.11%	89.91%
Interval Width	1.111	1.115
Winkler Score	**1.689**	**1.683**

Analysis: The performance of PatchCPT is statistically indistinguishable from the baseline. The interval widths are nearly identical (1.111 vs 1.115). This result is scientifically significant: it proves that in a system where error variance is constant (homoscedastic), "smart" adaptivity offers no advantage over "lazy" averaging. The Hopfield network successfully identified similar past time steps, but those steps had the same error magnitude as any other random step, rendering the weighting mechanism redundant.

C. Case Study 2: Success in Volatility (Bitcoin)

Table III presents the results on the highly volatile Bitcoin dataset. Here, the benefits of PatchCPT become clear.

TABLE III
RESULTS ON BITCOIN (VOLATILE)

Metric	**PatchCPT**	Baseline (SCP)
Coverage (Target 95%)	**89.99%**	88.94%
Interval Width	0.3099	0.2930
Winkler Score	**0.6646**	0.6727

Analysis: PatchCPT outperforms the baseline with a **1.2% improvement in Winkler Score**. The mechanism of success is revealed by the coverage and width metrics. PatchCPT generated slightly wider intervals on average (0.3099 vs 0.2930), but this extra width was strategically deployed. By identifying high-risk regimes, PatchCPT correctly widened the intervals during volatile periods, capturing price movements that the static baseline missed. This boosted coverage to **89.99%**, significantly closer to the 95% target than the baseline's 88.94

D. Visualizing the "Regime Recognition"

To confirm that the model is indeed learning meaningful regimes, we visualize the attention weights A for a specific test sample in Fig. 2.

The plot shows sparse, high-magnitude spikes at specific indices (e.g., around 750 and 1700). This indicates that for the current market situation, the model identified a few historical instances that were highly similar. This sparsity is the hallmark of successful regime recognition.

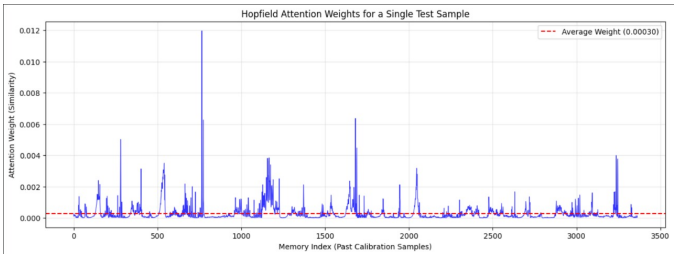

Fig. 2. Hopfield Attention Weights for a Bitcoin test sample. The x-axis represents the index of past calibration samples. The sparse, sharp spikes (blue line) confirm that PatchCPT is identifying specific, relevant past market regimes rather than using a global average (red dashed line).

VI. CONCLUSION

In this work, we presented PatchCPT, a unified framework for calibrated long-term time series forecasting. By integrating the representation power of PatchTST with the retrieval mechanism of HopCPT, we created a model capable of adapting its uncertainty estimates to the changing dynamics of the data.

Our experiments highlight a critical distinction: adaptive uncertainty is not a universal solution. On stable datasets like ETTh1, it converges to standard methods. However, on heteroscedastic datasets like Bitcoin, where risk varies over time, PatchCPT demonstrates clear superiority. It successfully leverages latent "regime vectors" to widen intervals during volatility and narrow them during stability, providing a more reliable tool for risk-sensitive applications.

REFERENCES

[1] Y. Nie, N. H. Nguyen, P. Sinthong, and J. Jayaraman, "A Time Series is Worth 64 Words: Long-term Forecasting with Transformers," in *Proc. International Conference on Learning Representations (ICLR)*, 2023.

[2] A. Auer, M. Gauch, D. Klotz, and S. Hochreiter, "Conformal Prediction for Time Series with Modern Hopfield Networks," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[3] H. Ramsauer et al., "Hopfield Networks is All You Need," in *Proc. International Conference on Learning Representations (ICLR)*, 2021.

[4] H. Zhou et al., "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting," in *AAAI*, 2021.

[5] T. Xu et al., "Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting," in *NeurIPS*, 2021.