DeepTimeNet: A Modular Hybrid Architecture for Robust and Efficient Time Series Modeling

sumit.joshi@se.com

Abstract

We present **DeepTimeNet**, a modular architecture for time series modeling that integrates a Mamba-Transformer hybrid backbone with a quantized Butterfly decoder. Designed for robustness, scalability, and efficiency, DeepTimeNet captures both long-range dependencies and local dynamics while enabling fast, low-memory inference. While this paper focuses on the backbone and decoder, DeepTimeNet is compatible with external embedding modules such as Token-Conditioned Embedding (TCE). Extensive experiments on real-world and synthetic benchmarks demonstrate that DeepTimeNet achieves state-of-the-art performance with significantly reduced computational overhead.

1 Introduction

Time series data is foundational across domains such as finance, healthcare, and energy. Despite the success of Transformer-based models, they often suffer from high computational cost and limited adaptability to noisy or non-stationary signals. Recent advances in state-space modeling and structured decoding offer promising alternatives, but integrating these components into a unified framework remains a challenge.

We propose **DeepTimeNet**, a modular hybrid architecture that addresses these limitations through two key innovations:

- A **Mamba-Transformer hybrid backbone** that interleaves state-space and attention-based blocks to capture both long-range dependencies and local interactions.
- A **quantized Butterfly decoder** that leverages structured linear transforms and log-domain quantization for efficient, low-latency inference.

DeepTimeNet is designed with modularity in mind, allowing seamless integration with external embedding modules such as Token-Conditioned Embedding (TCE), without architectural entanglement. Our experiments demonstrate that DeepTimeNet achieves state-of-the-art accuracy while maintaining a compact footprint and high throughput, making it suitable for both research and deployment in real-world time series applications.

2 Related Work

Time Series Transformers

Transformer-based models have gained traction in time series forecasting due to their ability to model long-range dependencies. Notable architectures include Informer zhou2021informer, which introduces sparse attention for scalability; Autoformerwu2021autoformer, which incorporates decomposition-based forecasting; and PatchTSTnie2023patchtst, which leverages patch-based tokenization for improved temporal resolution.

State-Space Models

State-space models offer an alternative to attention mechanisms by modeling temporal dynamics with linear complexity. S4 gu2021combining and Mambadao2023flash demonstrate strong performance on long-range sequence tasks, with Mamba introducing selective state updates for improved efficiency and expressiveness.

Efficient Decoding

Structured linear layers such as Butterfly matricesdao2020butterfly enable sub-quadratic computation and parameter efficiency. Combined with quantization-aware traininggholami2021survey, these techniques allow for low-latency inference and compact model deployment, which are critical for real-time time series applications.

Modular Embedding Strategies

Recent work on adaptive embeddings, including dynamic tokenization and uncertainty-aware representations, has shown promise in improving robustness to noise and distribution shifts. While not the focus of this paper, DeepTimeNet is compatible with external modules such as Token-Conditioned Embedding (TCE), which can be integrated for enhanced input representation.

3 Mamba-Transformer Hybrid Backbone

To balance efficiency and expressiveness in sequence modeling, DeepTimeNet employs a hybrid backbone composed of **Mamba blocks** and **Transformer blocks**. This design enables the model to capture both global and local temporal dependencies in a unified and scalable framework.

- **Mamba blocks** capture long-range dependencies with linear time complexity, making them efficient for modeling extended temporal contexts.
- **Transformer blocks** model complex local interactions through self-attention, enhancing representational power.

Alternating Configuration

In our experiments, we adopt an alternating configuration where Mamba and Transformer blocks are interleaved layer by layer. Let $\mathbf{H}_0 = \mathbf{E}$ denote the input to the backbone. The hidden states are updated as:

$$\mathbf{H}_{i+1} = \begin{cases} \text{MambaBlock}(\mathbf{H}_i), & \text{if } i \bmod 2 = 0 \\ \text{TransformerBlock}(\mathbf{H}_i), & \text{otherwise} \end{cases}$$

Optional: Gated Fusion Variant

As a future enhancement, DeepTimeNet supports a gated fusion variant where both Mamba and Transformer outputs are computed at each layer and combined:

$$\mathbf{H}_{\text{fused}} = \lambda \cdot \mathbf{H}_{\text{Mamba}} + (1 - \lambda) \cdot \mathbf{H}_{\text{Transformer}}$$

Here, $\lambda \in [0, 1]$ is a learnable or fixed scalar weight. This fusion allows dynamic blending of global and local features and may improve performance in complex or noisy time series scenarios.

Note: All results reported in this paper are based on the alternating configuration.

4 Quantized Butterfly Decoder

To enable efficient and scalable inference, DeepTimeNet employs a **quantized Butterfly decoder**, which combines structured linear transformations with log-domain quantization. This design significantly reduces computational and memory overhead while preserving the model's expressiveness.

Butterfly-Structured Linear Layers

Butterfly matrices are a class of structured transforms that decompose dense linear operations into a product of sparse, hierarchical matrices. This structure enables:

- **Sub-quadratic complexity** in matrix-vector multiplication.
- **Parameter efficiency** through logarithmic depth and weight sharing.
- **Inductive bias** for hierarchical and frequency-aware representations.

Let $\mathbf{H}_L \in \mathbb{R}^{T \times d}$ denote the final hidden representation from the backbone. The Butterfly decoder applies a structured transformation:

$$\mathbf{y} = \text{Butterfly}(\mathbf{H}_L)$$

where the Butterfly operator is implemented as a sequence of sparse matrix multiplications with learnable parameters.

Log-Domain Quantization

To further reduce inference cost and memory footprint, DeepTimeNet applies **log-domain quantization** to both the **weights** and **activations** within the Butterfly decoder layers. This quantization scheme approximates real-valued scalars using powers of two, enabling efficient computation via bit-shift operations.

Formally, each scalar x is quantized as:

$$\hat{x} = \text{sign}(x) \cdot 2^{\lfloor \log_2 |x| \rfloor}$$

This quantization is applied:

- **To the weights** of each Butterfly-structured linear transformation, significantly reducing model size and enabling efficient storage.
- **To the intermediate activations** between Butterfly layers, lowering memory bandwidth and compute cost during inference.

Log-domain quantization preserves the multiplicative structure of the data and is compatible with gradient-based optimization via straight-through estimators (STE). This makes it particularly suitable for deployment in latency-sensitive and resource-constrained environments, such as edge devices or real-time monitoring systems.

Benefits and Integration

The quantized Butterfly decoder in DeepTimeNet delivers a compelling balance between performance and efficiency, making it suitable for both research and deployment contexts:

- **Compactness:** The structured sparsity and quantization significantly reduce model size and memory footprint, enabling deployment on resource-constrained devices.
- **Speed:** The decoder accelerates inference through sub-quadratic computation and low-bit arithmetic, making it ideal for real-time applications.
- **Modularity:** Designed as a standalone decoding module, the quantized Butterfly decoder integrates seamlessly with a wide range of backbone architectures. In DeepTimeNet, it complements the Mamba-Transformer hybrid backbone without requiring architectural entanglement, supporting flexible experimentation and future extensibility.

This decoder design is particularly well-suited for deployment in latency-sensitive or resource-constrained environments, such as edge devices or real-time monitoring systems.

5 Experiments

We evaluate DeepTimeNet on a diverse suite of real-world and synthetic time series datasets to assess its forecasting accuracy, calibration, and computational efficiency. All experiments are conducted using the alternating Mamba-Transformer configuration, with the quantized Butterfly decoder applied at the output stage.

5.1 Datasets

We benchmark DeepTimeNet on the following datasets:

- **ETT**: Four variants from the Electricity Transformer Temperature dataset ETTh1, ETTh2, ETTm1, and ETTm2 capturing hourly and minute-level dynamics.
- **Electricity, Traffic, Weather**: Standard multivariate datasets widely used in time series forecasting.
- **Synthetic**: Noisy and distribution-shifted synthetic datasets designed to evaluate robustness under non-stationarity and perturbations.

5.2 Metrics

We report both predictive performance and efficiency metrics:

- **MAE** (Mean Absolute Error) and **RMSE** (Root Mean Squared Error) for accuracy.
- **ECE** (Expected Calibration Error) to assess probabilistic calibration.
- **Throughput** (samples/sec) and **Model Size** (MB) to quantify computational efficiency.

5.3 Results

DeepTimeNet consistently outperforms strong baselines including vanilla Transformers and Mamba-only models across all datasets. It achieves lower error and calibration metrics while maintaining a significantly smaller model size and higher inference throughput.

5.4 Ablation Studies

To isolate the contributions of each architectural component, we conduct ablation studies by removing the Mamba blocks and the quantized Butterfly decoder individually. Results show that both components contribute meaningfully to performance, with the full DeepTimeNet configuration yielding the best overall results.

6 Conclusion

DeepTimeNet demonstrates that modularity, efficiency, and robustness can be jointly achieved in time series modeling. By combining a Mamba-Transformer hybrid backbone with a quantized Butterfly decoder, DeepTimeNet delivers state-of-the-art performance with reduced computational cost. Its modular design allows seamless integration with external embedding modules such as Token-Conditioned Embedding (TCE), making it adaptable to a wide range of forecasting scenarios. Future work includes extending DeepTimeNet to multimodal, streaming, and continual learning settings.