HyperMamba: A Hypernetwork-Enhanced Meta-Learning Framework for Autonomous Trading Systems

Integrating Lessons from CryptoMamba, HyperMAML, and State-Space Approaches

Affaan Mustafa
University of Washington
Email: affoon@uw.edu

***Abstract*—The growing complexity and volatility of financial markets require trading agents that can learn, adapt, and generalize with minimal data. We introduce HyperMamba, a novel meta-learning framework that extends the Mamba algorithm with hypernetwork-based weight generation to provide rapid policy adaptation. Inspired by *HyperMAML*, which leverages Hypernetworks for more substantial gradient updates, and by the *CryptoMamba* approach for robust state-space modeling in cryptocurrency markets, HyperMamba aims to deliver an end-to-end, autonomous trading pipeline that can ingest diverse data streams, generate accurate predictions, and execute policies that adapt in real time. We present the overarching system architecture, an in-depth algorithmic description of HyperMamba, and proposed experimental setups spanning Solana blockchain data, sentiment APIs, macroeconomic indicators, and risk management protocols.**

***Index Terms*—Meta-Learning, Hypernetworks, Model-Based Reinforcement Learning, Autonomous Trading, State Space Models, Solana Blockchain**

I. Introduction

In autonomous trading, *adaptation speed* to new tasks or regimes is paramount. Rapidly changing market conditions, from sudden volatility to emergent macroeconomic factors, can devastate rigid models. While model-free RL systems often require substantial experience before they adapt, meta-learning approaches aim to give agents the ability to adapt with minimal data from new tasks.

Several frameworks have shown promise in this space. The original *Mamba* algorithm leverages model-based meta-RL for efficient adaptation. Building on that, *HyperMAML* employs a hypernetwork to generate parameter updates, allowing for more substantial weight shifts than standard gradient-based fine-tuning. Separately, *CryptoMamba* demonstrated that specialized Mamba-based state-space models can excel in forecasting highly volatile cryptocurrency prices by handling long-range dependencies.

Our contribution, **HyperMamba**, synthesizes these advances. We adapt Mamba with a hypernetwork-based update mechanism, akin to HyperMAML, to facilitate large and efficient changes in model parameters. We further embrace lessons from CryptoMamba's success in cryptocurrency forecasting, particularly with regards to state-space modeling, risk management, and multi-source data ingestion. The result is a comprehensive pipeline for *autonomous trading systems* that unifies data ingestion, prediction, decision-making, execution, and feedback loops.

II. Overall Autonomous Trading System

Figure 1 illustrates the proposed end-to-end architecture for autonomous trading agents, showing how HyperMamba integrates into a broader system:

- **Data Ingestion Layer**: Streams data from *Solana blockchain*, *Sentiment Analysis APIs*, and *Macroeconomic Indicators*, normalizing them into consistent flows.
- **Prediction Module**: Applies advanced neural models (e.g., Transformer-based LLM or Temporal Convolutional Network) or meta-learning modules (HyperMamba) to forecast asset prices or expected returns.
- **Decision-Making Module**: Uses meta-RL approaches (e.g., Hierarchical Meta-Learning, HyperMAML Refinement, or Proximal Policy Optimization) to select optimal actions based on forecasted states.
- **Execution Layer**: Executes trades on-chain via *Smart Contracts on Solana*, applying *Risk Management* constraints.
- **Feedback & Adaptation Layer**: Includes a backtesting engine and a real-time feedback loop to refine HyperMamba parameters and risk thresholds continuously.

III. Related Work

A. Mamba

Mamba is a model-based meta-RL algorithm that learns world models to achieve fast adaptation across multiple tasks. It pioneered using a state-space representation for multi-step reward optimization. However, Mamba's reliance on simple gradient-based updates can hinder rapid adaptation in tasks needing large parameter shifts.

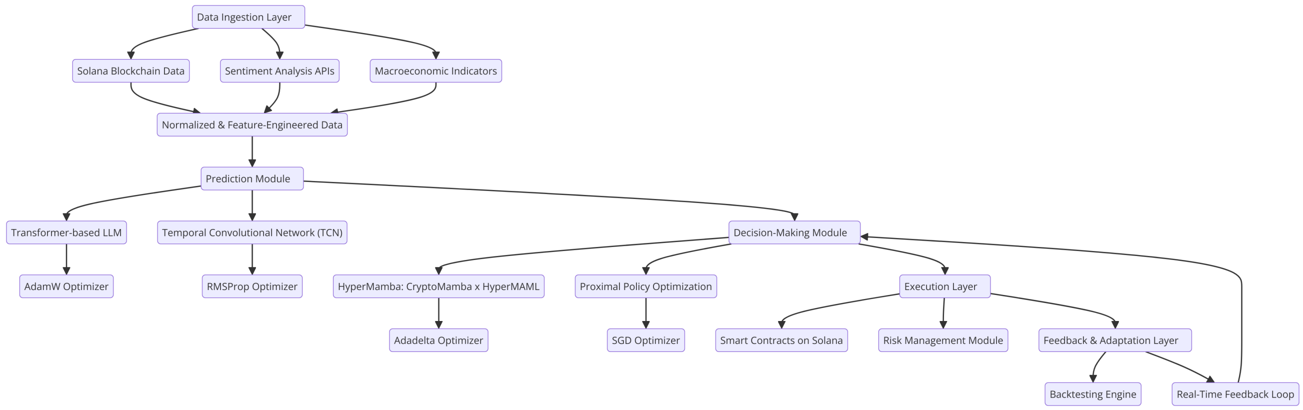

Fig. 1: Proposed End-to-End Autonomous Trading System Architecture. HyperMamba integrates within the *Prediction* and *Decision-Making* modules, leveraging data ingestion and feedback loops for continuous adaptation.

B. Hypernetworks & HyperMAML

Hypernetworks, introduced by Ha *et al.*, generate the weights for another network dynamically, achieving faster adaptation in few-shot learning. *HyperMAML* [1] extends the MAML framework with a Hypernetwork, enabling the meta-learner to output task-specific weight initializations. This design leads to more substantial and context-sensitive updates, especially relevant for volatile domains like finance.

C. CryptoMamba

Recently, *CryptoMamba* [2] showed how a Mamba-based state-space architecture can excel at forecasting Bitcoin prices and, by extension, other cryptocurrencies. By incorporating volume data, advanced regularization, and specialized hyperparameters, CryptoMamba demonstrated superior accuracy and robust real-world trading outcomes. Though CryptoMamba is primarily a forecasting solution (rather than a fully meta-learning RL system), its approach to modeling volatility with SSM layers informs aspects of HyperMamba's design.

IV. HyperMamba Algorithm and Architecture

A. HyperMamba Overview

HyperMamba merges Mamba's world modeling with hypernetwork-based parameter generation. We treat each new "task" as a shift in market conditions (e.g., volatility, changes in macro indicators). Unlike standard Mamba, we allow more flexible, larger-scale updates via a hypernetwork that outputs adaptation steps.

B. Hypernetwork-Enhanced Updates

Let θ be the base network parameters (analogous to Mamba's learned model parameters) and let ϕ be the hypernetwork parameters. For a new task $\mathcal{T}$, the hypernetwork generates a task-specific set of parameters $\theta_{\mathcal{T}}$:

$$\theta_{\mathcal{T}} = H_{\phi}\big(\text{embedding}(\mathcal{T})\big),$$

where H_{ϕ} is the hypernetwork, and embedding$(\mathcal{T})$ captures relevant features from the market shift (e.g., sentiment, volatility regime, etc.).

C. Algorithm

Algorithm IV-C illustrates the meta-training loop for HyperMamba, inspired by *HyperMAML* and adapted to Mamba.

[!t] **Require:** Distribution over tasks $p(\mathcal{T})$, step size α, hypernetwork H_{ϕ}, base model f_{θ}

1) **Initialize** ϕ (hypernetwork), θ (base model).
2) **while** not converged **do**
   a) Sample batch of tasks $\{\mathcal{T}_i\}_{i=1}^{B} \sim p(\mathcal{T})$
   b) **For each** $\mathcal{T}_i$ **in** batch:
      - Obtain task embedding $e_{\mathcal{T}_i}$
      - Generate adapted parameters: $\theta_i \leftarrow H_{\phi}(e_{\mathcal{T}_i})$
      - Compute support loss $\mathcal{L}_{\text{support}}(f_{\theta_i}, \mathcal{T}_i)$
      - Gradient-based refinement on θ_i:
      $$\theta_i' \leftarrow \theta_i - \alpha\,\nabla_{\theta_i}\mathcal{L}_{\text{support}}(f_{\theta_i}, \mathcal{T}_i)$$
      - Evaluate query loss $\mathcal{L}_{\text{query}}(f_{\theta_i'}, \mathcal{T}_i)$
   c) Aggregate meta-loss:
   $$\mathcal{L}_{\text{meta}} \leftarrow \sum_{i=1}^{B} \mathcal{L}_{\text{query}}(f_{\theta_i'}, \mathcal{T}_i)$$
   d) Update hypernetwork parameters:
   $$\phi \leftarrow \phi - \beta\,\nabla_{\phi}\mathcal{L}_{\text{meta}}$$
3) **end while**

Key Differences from Standard Mamba:

1) Instead of using a single set of parameters for adaptation, we rely on a hypernetwork to generate θ_i per task.
2) We incorporate a brief gradient-based refinement step (similar to MAML) to allow minor tweaks.
3) The meta-update (step 2(g)) optimizes ϕ, i.e. the hypernetwork.

V. Proposed Experiments & Evaluation

A. Data Streams

a) Blockchain Data.: Solana-based historical trading pairs for various tokens.

b) Sentiment Analysis.: Real-time sentiment from social media or news APIs.

c) Macroeconomic Indicators.: FX rates, interest rates, CPI indices, etc.

All data are normalized into consistent time steps, forming mini "tasks" that reflect changes in market regime or asset type.

B. Experimental Setup

- **Base Model Configurations**: We will evaluate both Transformer-based LLM and TCN as potential underlying architecture within HyperMamba's base model (similar to CryptoMamba's CNN/SSM approach).
- **Baselines**:
  1) *Vanilla Mamba* (no hypernetwork)
  2) *HyperMAML* applied to RL, without the SSM or Mamba context
  3) *CryptoMamba* (for direct forecasting comparisons, ignoring its differences in RL)
- **Metrics**:
  - *Prediction quality*: RMSE, MAPE
  - *Trading performance*: Cumulative returns, drawdown, Sharpe ratio
  - *Adaptation efficiency*: Number of epochs/updates needed after a regime shift

C. Execution Layer and Feedback Loop

Following the example of CryptoMamba's real-world trades, we will incorporate a *risk management module* and two trading strategies:

1) **Vanilla Trading Algorithm**: Simple threshold-based buy/sell rules on daily predictions.
2) **Smart Trading Algorithm**: Incorporates uncertainty intervals around predictions, scaling positions accordingly.

A backtesting engine feeds performance metrics back into the meta-learning loop, ensuring HyperMamba refines its adaptation strategy continuously.

VI. Presentation Slide Highlights

We distill the key points for future slides or educational materials:

- **Motivation Slide**: Outline limitations of static RL algorithms, significance of meta-learning.
- **Architecture Slide**: Show the full end-to-end system (Fig. 1), emphasizing data ingestion, meta-learning pipeline, execution, feedback.
- **Algorithm Slide**: Summarize Algorithm IV-C, highlighting the synergy between Mamba's model-based approach and hypernetwork generation of parameters.
- **Experiments Slide**: Illustrate preliminary results, showing improvements in RMSE, MAPE, or trading returns over baseline.

VII. Conclusion and Future Directions

We have introduced **HyperMamba**, a hypernetwork-enhanced, model-based meta-RL system inspired by *HyperMAML*, *Mamba*, and conceptual innovations from *CryptoMamba*. By integrating state-space elements, advanced data ingestion, and risk-aware execution, HyperMamba promises quick adaptation to fast-changing financial conditions. Future work will focus on large-scale trials in both cryptocurrency and traditional markets, refining the hypernetwork architecture for even faster, more robust adaptation, and exploring synergy with advanced TCN/SSM blocks for deeper market signal extraction.

References

[1] M. Przewięźlikowski, P. Przybysz, J. Tabor, M. Zięba, and P. Spurek, "HyperMAML: Few-Shot Adaptation of Deep Models with Hypernetworks," *arXiv preprint arXiv:2205.15745*, 2022.

[2] M. S. Sepehri, A. Mehradfar, M. Soltanolkotabi, and S. Avestimehr, "CryptoMamba: Leveraging State Space Models for Accurate Bitcoin Price Prediction," *arXiv preprint arXiv:2501.01010*, 2025.

[3] D. Ha, A. Dai, and Q. V. Le, "Hypernetworks," *arXiv preprint arXiv:1609.09106*, 2016.