

# EATA: Explainable Algorithmic Trading Agent via Neural-Guided Symbolic Regression

Jingtong Zhang<sup>a</sup>, Xiaoang Yang<sup>b</sup>, Yinqi Shi<sup>c</sup>, Yin Tang<sup>d</sup>,

<sup>a</sup>University of New South Wales, Sydney, 2052, Australia

<sup>b</sup>School of Cybersecurity, Jinan University, GuangZhou, 511443, China

<sup>c</sup>Jinan University – University of Birmingham Joint Institute, Jinan University, GuangZhou, 511443, China

<sup>d</sup>School of Management, Jinan University, GuangZhou, 510632, China

## Abstract

Algorithmic trading systems powered by deep learning have achieved remarkable performance but suffer from a critical limitation: lack of interpretability. This opacity hinders trust, regulatory compliance, and strategy refinement. We propose EATA (Explainable Algorithmic Trading Agent), a novel framework that combines Monte Carlo Tree Search (MCTS) with neural network guidance to automatically discover interpretable mathematical expressions for stock price prediction. Our key contributions include: (1) a three-headed Policy-Value-Profit network (PVNet) that jointly optimizes for search guidance, prediction accuracy, and trading profitability; (2) a domain-specific grammar incorporating financial operators such as moving averages, momentum, and volatility indicators; (3) a grammar augmentation mechanism that extracts and reuses high-quality sub-expressions to accelerate search; and (4) a sliding window training scheme with expression inheritance for temporal adaptation. Extensive experiments on **representative constituent stocks from the S&P 500** index over a 4-year period (2020–2024) demonstrate that EATA achieves competitive returns compared to state-of-the-art deep learning methods while providing fully interpretable trading rules in the form of mathematical formulas. The discovered expressions reveal meaningful market patterns and can be directly validated by domain experts.

**Keywords:** Algorithmic Trading, Symbolic Regression, Monte Carlo Tree Search, Explainable AI, Neural-Guided Search

## 1. Introduction

Algorithmic trading systems powered by deep learning have achieved remarkable performance in financial markets [1, 2]. However, these “black-box” models present a fundamental challenge: their decision-making processes remain opaque, hindering trust, regulatory compliance (e.g., MiFID II [3]), and strategic refinement [4, 5, 6]. This opacity is particularly problematic in finance, where understanding *why* a model fails is crucial for risk management.

Symbolic regression offers a promising alternative by discovering interpretable mathematical expressions that map inputs to outputs [7, 8]. Unlike neural networks, symbolic expressions can be directly validated by domain experts and audited by regulators. However, existing symbolic regression approaches face a critical limitation when applied to trading: **they typically optimize for pointwise prediction accuracy** (e.g., MSE), which **does not necessarily correlate with trading profitability**.

Consider the following fundamental disconnect: minimizing MSE penalizes all prediction errors equally, regardless of their economic impact. In trading, a small error in

predicting a large directional move (strong trend) can be far more costly than a large error during sideways markets. Moreover, financial returns exhibit heavy-tailed distributions with significant skewness and kurtosis [9]—properties that MSE-based objectives fail to capture.

This observation leads to our **central research question**: *How can we design a symbolic regression framework that explicitly optimizes for trading profitability rather than simple prediction accuracy?* Answering this requires addressing three sub-questions:

RQ1 **Objective Design**: What distributional divergence metric can effectively capture the asymmetric, heavy-tailed nature of financial returns?

RQ2 **Search Guidance**: How can neural networks guide symbolic search to discover profitable expressions without sacrificing interpretability?

RQ3 **Empirical Validation**: Can profit-optimized symbolic models achieve risk-adjusted returns competitive with black-box deep learning?

Our key insight is that the **Wasserstein distance** (Earth Mover’s Distance) provides a theoretically grounded measure for comparing return distributions that respects both tail risk and directional asymmetry [10, 11]. Unlike KL divergence or MSE, Wasserstein distance measures the minimal “work” required to transform one distribution into

Email address: Corresponding author: Yin Tang (Email: [ytang@jnu.edu.cn](mailto:ytang@jnu.edu.cn)). (Yin Tang)

URL: <https://orcid.org/0000-0001-7693-7543> (Yin Tang)

another, making it sensitive to both the magnitude and location of distributional differences.

Based on this insight, we propose **EATA** (Explainable Algorithmic Trading Agent), a neural-guided symbolic regression framework with three key innovations:

1. **Profit-Aware Learning:** A three-headed neural network (PVNet) with policy, value, and *profit* heads. The profit head explicitly predicts the Wasserstein-based trading reward, guiding MCTS toward expressions with favorable risk-return profiles.
2. **Domain-Specific Grammar:** A context-free grammar incorporating financial operators (moving averages, momentum, RSI, volatility) with fixed windows, encoding domain knowledge to constrain the search space.
3. **Temporal Adaptation:** A sliding window training scheme with expression inheritance and grammar augmentation, enabling adaptation to non-stationary market conditions.

We evaluate EATA on **representative constituent stocks from the S&P 500** index over a 4-year period (2020–2024), comparing against multiple baselines including state-of-the-art deep learning (LSTM, Transformer), reinforcement learning (PPO), and symbolic regression (GP, NEMoTS [12]). Results show that EATA achieves a Sharpe ratio of **0.83**, significantly outperforming the best baseline while providing fully interpretable trading rules. Our main contributions are:

1. We formalize the symbolic regression objective for trading as a **distribution matching problem** using Wasserstein distance, bridging the gap between prediction accuracy and economic utility.
2. We propose a **novel neural-guided search framework** with a **dedicated profit head** that predicts trading rewards, enabling the search to explicitly optimize for profitability.
3. We conduct **extensive empirical validation** on real-world market data with in-depth ablations, demonstrating that profit-aware symbolic regression can compete with black-box deep learning.

The rest of the paper is organized as follows: Section 2 reviews related work in algorithmic trading evaluation, symbolic regression, and neural-guided search. Section 3 presents the EATA framework, including the problem formulation, domain grammar, PVNet, and neural-guided MCTS training. Section 4 reports the experimental setup, baselines, main results, and ablation and robustness analyses. Finally, Section 5 concludes with limitations and future directions.

## 2. Related Work

This section situates EATA at the intersection of (i) empirical algorithmic trading evaluation, (ii) symbolic regression and neural-guided search, and (iii) distributional

/ risk-sensitive objectives. We organize prior work following a pragmatic taxonomy aligned with our research questions: (A) evaluation and backtesting protocols, (B) symbolic discovery and search algorithms, and (C) distributional metrics and risk-sensitive learning.

### 2.1. Evaluation, Backtesting, and Statistical Validity in Trading

Empirical algorithmic trading studies typically report risk-adjusted performance metrics derived from classical portfolio theory [13, 14]. However, in finite samples, Sharpe ratio estimates can have substantial uncertainty and non-Gaussian sensitivity [15, 16], and naive benchmark selection can induce multiple-testing and data-snooping bias. Classic econometric procedures such as the reality check [17] and the superior predictive ability (SPA) test [18] provide principled baselines for large-scale strategy comparison; related work further emphasizes the need to correct for selection bias and backtest overfitting [19, 20]. In the broader asset-pricing literature, multiple-testing issues have been documented at scale when large factor libraries are mined [21]. Beyond statistical validity, realistic transaction cost modeling is essential because many strategies degrade sharply once implementation frictions are introduced [22]. These considerations motivate our emphasis on risk-adjusted metrics and robustness checks rather than solely predictive accuracy.

A recurring pitfall in the trading literature is to equate in-sample predictive performance with out-of-sample trading profitability. This mismatch is amplified because the evaluation target (returns) is noisy, heavy-tailed, and regime-dependent [9]. As a result, small modeling changes can lead to unstable rankings across assets and periods.

From a scientific standpoint, this yields three concrete failure modes. (i) **Metric fragility:** the same strategy can appear superior under one risk-adjusted metric and inferior under another when return distributions are non-Gaussian; this is why robust inference for Sharpe ratios and performance comparisons is essential [15, 16]. (ii) **Search-induced false discoveries:** when thousands of candidate rules are implicitly tested (e.g., through GP, alpha mining, or hyperparameter sweeps), the top-performing strategy is statistically biased upward; reality-check style tests and overfitting diagnostics were introduced precisely to address this phenomenon [17, 18, 20]. (iii) **Protocol leakage:** evaluation choices such as temporal splits, rebalancing frequency, and transaction cost assumptions can materially change conclusions, which makes reproducibility and explicitly stated protocols non-negotiable.

These issues motivate EATA’s methodological stance: rather than treating evaluation as an afterthought, we view objective design and search constraints as part of the validity mechanism that reduces the degrees of freedom available for accidental overfitting.

**Implication for EATA.** This motivates two design principles used in our framework: (i) constrain the hypothesis space (grammar) to reduce implicit multiple testing,

and (ii) align the learning signal with economically meaningful outcomes to avoid proxy-driven improvements.

Finally, we note that algorithmic trading is not only a modeling problem but also a market-structure problem: automated execution and algorithmic strategies can affect liquidity and price dynamics [23]. This further motivates evaluation protocols that emphasize robustness and auditability.

## 2.2. Interpretable Alpha Discovery and Formulaic Trading Signals

A long-standing line of work studies human-interpretable, formulaic trading signals and alpha factors. Technical analysis has been extensively surveyed with mixed evidence on profitability and significant sensitivity to market regimes and evaluation protocols [24]. Recent “alpha mining” efforts aim to automatically discover and combine formulaic factors at scale, ranging from genetic programming-based alpha discovery [25, 26] to reinforcement learning and language-model assisted frameworks [27, 28, 29]. These methods reinforce a key methodological point: interpretability does not automatically imply economic usefulness unless the discovery objective is aligned with trading utility.

Alpha discovery systems often make an implicit assumption: a factor that improves a proxy objective (e.g., prediction loss, IC, or a simplified RL reward) will translate into a profitable and robust trading rule. In practice, this assumption breaks under at least three common constraints. (i) **Turnover and costs:** weak predictive signals around zero can generate frequent trading and collapse under realistic friction models [22]. (ii) **Tail constraints:** many practitioners operate under drawdown limits; factors that occasionally incur large losses can be unacceptable even if mean returns are positive. (iii) **Hypothesis space inflation:** large factor libraries make false discoveries more likely unless the search is explicitly regularized and the evaluation protocol accounts for multiple testing [21, 17].

EATA can be viewed as a “constrained alpha miner”: it preserves formulaic interpretability but binds discovery to a profit-aware, distribution-sensitive learning signal while using grammar constraints to limit degrees of freedom.

**Implication for EATA.** We therefore treat the discovery objective (profit-aware, distribution-sensitive) and the hypothesis-space control (grammar constraints) as first-class components rather than implementation details.

## 2.3. Black-Box Learning for Trading and the Interpretability Gap

Deep learning and reinforcement learning have been extensively applied to trading, including recurrent architectures for financial prediction [1, 30], transformers for multi-scale pattern extraction [31, 32], and deep RL systems for sequential decision making [33, 34]. These models can achieve strong empirical performance, but their inductive biases are largely opaque and they can be sensitive to non-stationarity and evaluation protocol choices.

In addition to neural models, tree ensembles remain highly competitive in financial prediction pipelines due to their sample efficiency and robustness to heterogeneous features. Gradient boosting methods such as XGBoost [35] and LightGBM [36] are widely used baselines in practice. However, even when post-hoc explanation tools are applied, ensemble models can yield explanations that are unstable across correlated features or regime shifts. Rule extraction methods such as inTrees [37] partially bridge this gap, but they typically explain a fixed predictor rather than optimizing discovery toward interpretable, deployable trading rules.

The black-box literature often assumes that improved predictive capacity will translate into improved trading performance under deployment constraints. However, in finance this assumption is especially brittle because (i) low signal-to-noise ratios amplify overfitting, (ii) regime shifts break stationarity, and (iii) decision-making must satisfy auditability and compliance constraints [3, 6, 5]. Moreover, post-hoc explanation methods (LIME/SHAP) can provide local rationales but may fail robustness and faithfulness requirements, and can be manipulated [38, 39, 40]. This motivates intrinsically interpretable strategies in which the decision rule is the model.

EATA positions interpretability as a constraint on the hypothesis class and the training signal, rather than a separate explanation layer applied after training.

## 2.4. Symbolic Regression for Time Series: Evolutionary, Grammar-Based, and Neural

**Evolutionary and Grammar-Based GP.** Genetic Programming (GP) remains a foundational paradigm for symbolic regression [7, 41]. Grammar-based GP offers a principled way to encode domain constraints and reduce invalid expressions [42]. Modern toolkits such as gplearn [43] and Operon [44] improve scalability via efficient evaluation. In computational finance, evolutionary approaches have been widely explored for alpha discovery and rule mining [45, 25], but they often suffer from bloat and expensive fitness evaluations, especially when the objective requires realistic backtesting.

**Neural Symbolic Regression.** Recent work incorporates neural guidance to improve sample efficiency and scalability. DSR [46] uses policy gradients to generate expressions token-by-token, while neural-to-symbolic models based on transformers have demonstrated promising scaling behavior on symbolic tasks [47, 48, 49]. Comprehensive reviews highlight that a central challenge remains designing objectives that reflect the downstream utility of discovered expressions, rather than only reconstruction error [50].

Beyond direct expression generation, grammar-aware generative models provide another way to impose syntactic constraints while learning expressive priors. For instance, grammar variational autoencoders [51] demonstrate how context-free grammars can be used to ensure syntactic validity. Separately, program synthesis and library-learning

approaches such as DreamCoder [52] learn reusable subroutines that improve sample efficiency and compress the hypothesis space. These ideas conceptually align with EATA’s module extraction: instead of learning a latent program prior, we explicitly maintain a bounded library of reusable sub-expressions to preserve auditability.

For time series trading, two practical constraints are often under-emphasized. **(i) Evaluation-dominated cost:** in trading, the expensive part is not parsing expressions but evaluating them under a backtesting protocol (positions, costs, and risk metrics). Methods that generate many candidates without strong pruning are therefore computationally prohibitive. **(ii) Spuriousness under non-stationarity:** without explicit domain constraints, symbolic regression can fit transient correlations that disappear under regime shifts; even when expressions are interpretable, they may not be stable.

Grammar constraints in EATA serve as a structural prior that is both scientific and practical: scientifically, they operationalize domain knowledge and restrict the hypothesis class; practically, they reduce the size of the implicit multiple-testing problem created by search.

**Implication for EATA.** We explicitly encode financial operators and fixed horizons to trade off expressiveness for robustness and auditability.

## 2.5. Neural-Guided Search and MCTS for Symbolic Discovery

Monte Carlo Tree Search (MCTS) [53, 54] provides a principled exploration-exploitation trade-off and has been successfully combined with neural policy-value guidance in game-playing systems [55, 56]. In symbolic regression, MCTS-based methods have been explored as an alternative to GP [57, 58], and actor-critic style enhancements have been proposed to improve search efficiency [59]. Most relevant to our work is **NEMOTS** (Neural-Enhanced Monte-Carlo Tree Search) [12], which pairs MCTS with policy-value networks for time series symbolic regression. Similarly, **AlphaCFG** [60] treats alpha discovery as a grammar-guided generation problem, utilizing a tree-structured linguistic MDP to enforce syntactic validity. EATA shares the neural-guided search paradigm but differentiates itself by explicitly re-architecting the guidance signal toward *profit-aware* objectives (via Wasserstein distance) rather than solely optimizing IC, and by introducing temporal adaptation mechanisms.

More broadly, MCTS has been used as a general-purpose mechanism for symbolic and scientific discovery beyond time series, e.g., for discovering governing equations [61]. This line of work highlights an important methodological point: search algorithms benefit from domain priors and stable evaluation signals; otherwise, they can spend most of the budget exploring syntactically valid but semantically uninformative regions.

Existing MCTS-based symbolic regression methods typically optimize predictive fit or benchmark task accuracy;

the resulting expressions can be interpretable yet economically weak when deployed as trading rules. Concretely, optimizing MSE can produce forecasts that are accurate around the mean but poorly calibrated in the tails, which is precisely where trading risk is concentrated.

Moreover, in financial markets, two additional boundary conditions matter. **(i) Regime dependence:** expressions discovered on one window can decay when volatility, liquidity, or macro conditions shift. **(ii) Objective mismatch:** even if a symbolic model predicts returns, turning it into a trading rule requires a decision layer; if the discovery objective ignores this layer, the downstream strategy may be unstable.

EATA addresses these gaps by coupling neural-guided MCTS with a profit-aware head and by using sliding-window inheritance to bias search toward reusable substructures that survive across regimes.

**Implication for EATA.** This motivates our profit head (objective alignment) and inheritance/augmentation mechanisms (non-stationarity and search efficiency).

## 2.6. Distributional Metrics, Optimal Transport, and Risk-Sensitive Learning

Financial returns are heavy-tailed and asymmetric [9], making point-wise regression losses (e.g., MSE) a weak proxy for economic utility. Optimal transport provides a geometrically meaningful way to compare distributions and has been formalized in foundational texts and modern computational treatments [10, 11]. In reinforcement learning, distributional perspectives [62] and risk-sensitive objectives such as CVaR optimization [63] further motivate designing reward signals that reflect tail risk and distributional shape rather than only mean outcomes. These insights directly motivate our use of Wasserstein distance as a profit-aware training signal.

While risk-sensitive criteria (e.g., CVaR) explicitly target tail losses, they often require careful tuning of tail parameters and can be unstable when tail events are rare. Wasserstein distance provides a complementary mechanism: it compares entire distributions and remains meaningful even when supports differ.

The key design question is not “which metric is theoretically elegant” but “which metric provides a stable learning signal under noisy, heavy-tailed samples.” In this sense, Wasserstein distance operationalizes a distribution-level inductive bias: it penalizes systematic miscalibration across quantiles and thus discourages strategies that look good in average outcomes but fail under tail events.

**Implication for EATA.** We use Wasserstein distance to create a profit-aware training signal that remains informative under heavy tails and regime variability.

## 2.7. Explainability, Robustness, and Regulation in Finance

In high-stakes settings such as finance, interpretability is tied to governance and compliance requirements, including regulatory constraints (e.g., MiFID II [3]) and broader

discussions on explanation rights [64]. While post-hoc explanation methods such as LIME [38] and SHAP [39] are widely used, their robustness and faithfulness can be challenged, especially under adversarial manipulations [40]. Furthermore, there is ongoing debate on whether attention mechanisms constitute explanations [65, 66, 67]. These findings support the argument that intrinsically interpretable symbolic strategies can be preferable for auditability and scientific scrutiny [4, 5, 6].

From a debate standpoint, the key tension is between *post-hoc transparency* and *intrinsic interpretability*. Post-hoc explanations can provide local insights but may not satisfy robustness or auditability requirements; in contrast, symbolic strategies expose the decision rule directly but risk overfitting unless the hypothesis space and objectives are carefully designed. EATA’s design explicitly treats interpretability as a constraint on the hypothesis class (grammar) rather than a visualization layer applied after training.

An emerging complementary direction is to design models that are self-explaining by construction (rather than explained after training), e.g., self-explaining neural networks [68]. However, such approaches still require careful definition of what constitutes a faithful explanation and may not yield a globally auditable rule set. In contrast, symbolic expressions provide explicit, globally applicable decision rules that can be inspected and stress-tested.

## 2.8. Positioning of EATA

Table 1: Comparison with Related Approaches

Method	Interp.	Neural	Finance	Profit	Stationarity
LSTM/Transformer	✗	✓	✗	✗	✓
FinRL (PPO)	✗	✓	✓	✓	✓
GP (Symbolic)	✓	✗	✗	✗	✗
Alpha Mining [25, 27, 29]	✓	✓	✓	✓	✗
DSR	✓	✓	✗	✗	✗
NEMoTS	✓	✓	✗	✗	✗
EATA (Ours)	✓	✓	✓	✓	✓

Table 1 compares EATA with representative families across key dimensions. Compared to prior alpha mining and symbolic discovery pipelines, EATA uniquely integrates *neural-guided search* (MCTS + PVNet), *financial operators* (domain grammar), and *profit-aware learning* (distributional reward) under a unified, interpretable framework.

The above literature suggests that the core difficulty is not the absence of interpretable models, nor the absence of powerful search, but the *mismatch between discovery objectives and trading utility* under a large, implicitly tested hypothesis space. EATA’s components map directly to these failure modes: the Wasserstein-based profit signal targets distribution-level utility under heavy tails; grammar constraints bound degrees of freedom and improve auditability; neural-guided MCTS improves search efficiency without sacrificing explicit exploration; and sliding-window inheritance provides a pragmatic mechanism for non-stationarity.

## 3. Methodology

This section presents the technical details of the Explainable Algorithmic Trading Agent (EATA). We first establish a unified notation system summarized in Appendix Appendix B, then describe the core algorithms with their design rationale.

### 3.1. Problem Formulation

Given a multivariate financial time series  $\mathbf{X} = \{\mathbf{x}_t\}_{t=1}^T$  where  $\mathbf{x}_t \in \mathbb{R}^d$  represents  $d$  market features at time  $t$ , we seek an interpretable expression  $e^*$  that balances predictive accuracy with trading profitability:

$$e^* = \arg \max_{e \in \mathcal{E}} [\eta \cdot \mathcal{A}(e, \mathbf{X}) + (1 - \eta) \cdot \mathcal{P}(e, \mathbf{X})], \quad (1)$$

where  $\mathcal{E}$  denotes the valid expression space defined by grammar  $\mathcal{G}$ ,  $\mathcal{A}(\cdot)$  measures prediction accuracy,  $\mathcal{P}(\cdot)$  measures trading profit, and  $\eta \in [0, 1]$  balances the two objectives.

**Rationale.** A pure accuracy objective is known to be misaligned with trading utility because it treats all errors symmetrically and ignores risk asymmetry. Conversely, a pure profit objective can overfit to noisy return realizations and produce brittle expressions that exploit idiosyncratic fluctuations. The mixture in Eq. 1 serves as a regularization mechanism:  $\mathcal{A}$  stabilizes the search toward expressions that capture repeatable predictive structure, while  $\mathcal{P}$  biases the discovered expressions toward economically meaningful outcomes. The control parameter  $\eta$  therefore exposes a transparent trade-off between statistical fit and economic utility, which can be tuned to match risk preferences.

### 3.2. Domain-Specific Grammar Design

We employ a context-free grammar  $\mathcal{G} = (\mathcal{V}, \Sigma, \mathcal{R}, S)$  tailored for financial time series analysis. The grammar incorporates domain knowledge through fixed-window financial operators.

**Non-terminals and Terminals.** The non-terminal set  $\mathcal{V} = \{A, B, D\}$  includes primary expressions ( $A$ ), helper expressions ( $B$ ), and decision predicates ( $D$ ). The terminal set  $\Sigma$  comprises feature variables  $\{x_0, x_1, \dots, x_{d-1}\}$ , constants, and operators.

**Production Rules.** The rule set  $\mathcal{R}$  comprises four categories:

*Basic Arithmetic:*

$$A \rightarrow A + A \mid A - A \mid A \times A \mid A \div A \mid A \times C \mid A + C \quad (2)$$

*Mathematical Functions:*

$$A \rightarrow \cos(A) \mid \sin(A) \mid \exp(A) \mid \log(B) \mid \sqrt{B} \mid |A| \mid A^2 \quad (3)$$

# EATA: Explainable Algorithmic Trading Agent Framework

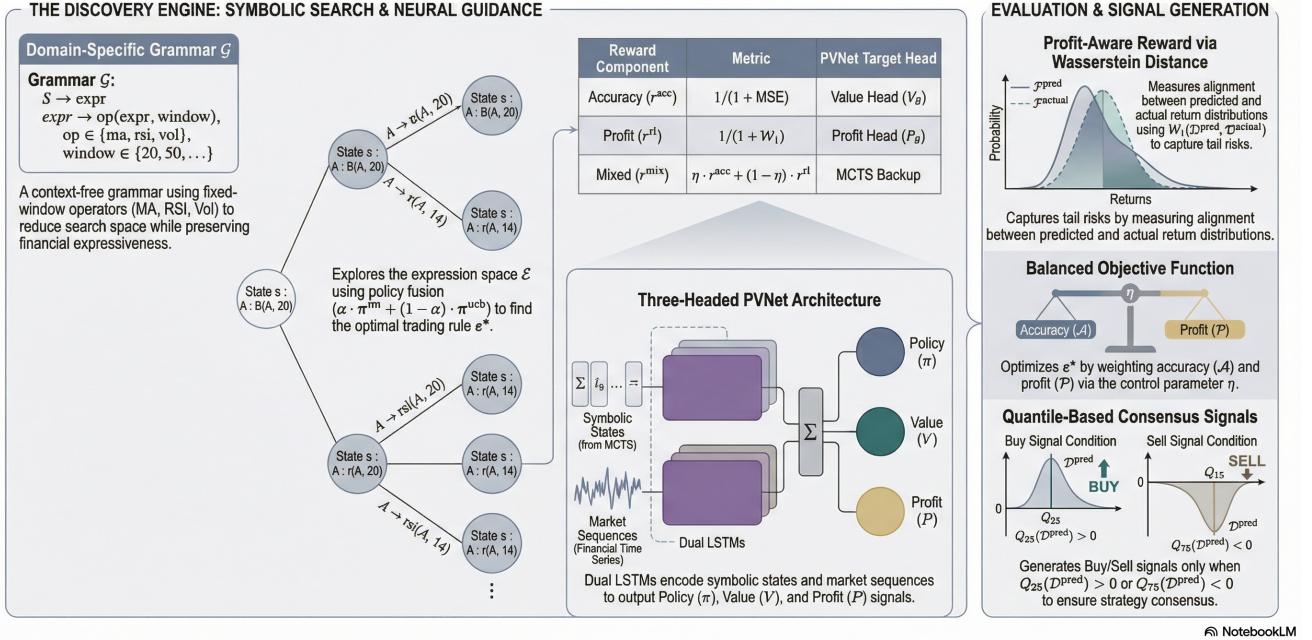


Figure 1: EATA Framework Architecture. The system consists of four phases: (1) Data & Protocol handling sliding windows; (2) Symbolic Search via neural-guided MCTS; (3) Neural Guidance using a three-headed PVNet; and (4) Evaluation producing interpretable trading signals.

### Financial Operators (Decoupled Parameters):

$$\begin{aligned} A \rightarrow & \text{delay}(A, L) \mid \text{ma}(A, W) \mid \text{diff}(A, L) \mid \text{mom}(A, L) \\ & \mid \text{max\_n}(A, W) \mid \text{min\_n}(A, W) \mid \text{rsi}(A, W) \mid \text{vol}(A, W) \end{aligned} \quad (4)$$

$$W \rightarrow 5 \mid 10 \mid 20 \mid 30 \mid 60, \quad L \rightarrow 1 \mid 5 \mid 10 \mid 20 \quad (5)$$

### Conditional Logic:

$$A \rightarrow \text{ite}(D, A, A), \quad D \rightarrow A > B \mid A < B \quad (6)$$

**Design Rationale.** We adopt a decoupled grammar design inspired by AlphaCFG [60], separating operators from their parameters. This introduces parameter non-terminals ( $W$  for windows,  $L$  for lags) that resolve to a discrete set of valid values. This compositional structure reduces the number of unique production rules while maintaining expressiveness, and prevents the generation of semantically invalid parameter combinations (e.g., negative windows). The conditional operator `ite` enables piecewise expressions for regime-switching strategies.

**Rationale and Alternatives.** In finance, unconstrained operator sets often lead to expressions that are either uninterpretable (deeply nested non-linearities) or non-robust (operators effectively “search” over hyperparameters via continuous constants). Fixed-window operators serve two purposes: they encode domain priors and mitigate multiple-testing effects by limiting degrees of freedom. In contrast, allowing arbitrary window lengths would

significantly inflate the hypothesis space and increase the risk of data snooping [17, 19]. Grammar constraints can be viewed as a structural regularizer that improves both interpretability and generalization.

### 3.3. Neural-Guided Monte Carlo Tree Search

Algorithm 1 presents the MCTS procedure with neural network guidance. Each episode traverses the search tree using a policy fusion mechanism.

**Design Rationale.** The dual fusion mechanism addresses uncertainty in neural network predictions early in training. When  $\alpha$  is small (buffer not full), rollout provides reliable value estimates; as training progresses,  $\alpha$  increases, trusting the neural network more. The value-profit fusion with  $\omega$  ensures expressions are selected based on both predictive fit and profit potential.

**Why MCTS (vs. GP / sequence generators)?** GP is effective but can suffer from bloat and expensive evaluation, especially when backtesting-based objectives are used. Sequence generators (RNN/transfomers) can scale but often require large training data and can generate syntactically valid yet semantically meaningless expressions without strong constraints. MCTS provides a controllable exploration mechanism: it can exploit neural priors when available while retaining explicit exploration through UCB. This is particularly suitable for financial time series where the signal-to-noise ratio is low and naive exploitation can lead to overfitting.

---

**Algorithm 1** Neural-Guided MCTS for Symbolic Regression

---

**Require:** Data  $\mathbf{X}$ , grammar  $\mathcal{G}$ , network  $f_\theta$ , episodes  $K$ , guidance weight  $\alpha$ , fusion weight  $\omega$ , prior tree  $\tau^{\text{prev}}$

**Ensure:** Best expression  $e^*$ , best tree  $\tau^*$ , experience buffer  $\mathcal{D}$

```

1: Initialize  $Q(\cdot) \leftarrow 0$ ,  $N(\cdot) \leftarrow 1$ ,  $\mathcal{D} \leftarrow \emptyset$ 
2:  $s \leftarrow \text{INITIALIZESTATE}(\tau^{\text{prev}})$ 
3: for  $k = 1$  to  $K$  do
4:    $s \leftarrow s_0$ ,  $\text{states} \leftarrow []$ ,  $\text{actions} \leftarrow []$ 
5:    $\mathcal{U} \leftarrow \text{GETUNVISITED}(s)$ 
6:   while  $\mathcal{U} = \emptyset$  and not  $\text{ISTERMINAL}(s)$  do           ▷ Selection
7:      $\pi^{\text{nn}}, V, P \leftarrow f_\theta(s, \mathbf{X})$ 
8:      $\pi^{\text{ucb}} \leftarrow \text{COMPUTECUCB}(s, C)$ 
9:      $\pi \leftarrow \alpha \cdot \pi^{\text{nn}} + (1 - \alpha) \cdot \pi^{\text{ucb}}$ 
10:     $\pi \leftarrow \text{NORMALIZE}(\pi)$ 
11:     $a \sim \text{Categorical}(\pi)$ 
12:     $\text{states.append}(s)$ ,  $\text{actions.append}(a)$ 
13:     $s \leftarrow \text{STEP}(s, a)$ 
14:     $\mathcal{U} \leftarrow \text{GETUNVISITED}(s)$ 
15:    if  $|s| \geq L^{\text{max}}$  then
16:       $\text{BACKPROP}(\text{states}, \text{actions}, 0)$ 
17:      break
18:    end if
19:  end while
20:  if  $\mathcal{U} \neq \emptyset$  then           ▷ Expansion
21:     $\pi^{\text{nn}}, V, P \leftarrow f_\theta(s, \mathbf{X})$ 
22:     $a \sim \text{Categorical}(\alpha \cdot \pi^{\text{nn}} + (1 - \alpha) \cdot \text{UNIFORM}(\mathcal{U}))$ 
23:     $s' \leftarrow \text{STEP}(s, a)$ 
24:     $r, e \leftarrow \text{EVALUATE}(s', \mathbf{X})$ 
25:    if  $\text{ISTERMINAL}(s')$  then
26:       $\text{UPDATEMODULES}(e, r)$ 
27:    end if
28:  else           ▷ Simulation via leaf evaluation
29:     $r, e \leftarrow \text{EVALUATE}(s, \mathbf{X})$ 
30:  end if
31:   $v^{\text{nn}} \leftarrow \omega \cdot V + (1 - \omega) \cdot P$            ▷ Fuse value heads
32:   $v^{\text{roll}} \leftarrow \text{ROLLOUT}(s, n^{\text{play}})$            ▷ Random simulations
33:   $r^{\text{final}} \leftarrow \alpha \cdot v^{\text{nn}} + (1 - \alpha) \cdot v^{\text{roll}}$            ▷ Fuse neural and rollout
34:   $\text{BACKPROP}(\text{states}, \text{actions}, r^{\text{final}})$ 
35:   $\mathcal{D}.\text{append}((\text{states}, \mathbf{X}, \pi, v^{\text{nn}}))$ 
36: end for
37: return  $e^*, \tau^*, \mathcal{D}$ 

```

---

### 3.4. Expression Scoring with Coefficient Optimization

Algorithm 2 evaluates expressions and optimizes numerical constants.

---

**Algorithm 2** Expression Scoring with Coefficient Estimation

---

**Require:** Expression  $e$ , data  $(\mathbf{X}, \mathbf{y})$ , timeout  $t^{\text{max}}$

**Ensure:** Score  $r$ , optimized expression  $e^{\text{opt}}$

```

1:  $n_c \leftarrow \text{COUNTCONSTANTS}(e)$ 
2: if  $n_c = 0$  then
3:    $\hat{\mathbf{y}} \leftarrow \text{EVALUATE}(e, \mathbf{X})$ 
4: else if  $n_c \geq 10$  then
5:   return  $0, e$            ▷ Too many constants
6: else
7:    $\mathbf{c}^* \leftarrow \arg \min_{\mathbf{c}} \|\text{EVALUATE}(e, \mathbf{X}, \mathbf{c}) - \mathbf{y}\|_2$  ▷ Powell optimization
8:    $e^{\text{opt}} \leftarrow \text{SUBSTITUTE}(e, \mathbf{c}^*)$ 
9:    $\hat{\mathbf{y}} \leftarrow \text{EVALUATE}(e^{\text{opt}}, \mathbf{X})$ 
10:  end if
11:   $\text{MSE} \leftarrow \frac{1}{T} \|\hat{\mathbf{y}} - \mathbf{y}\|_2^2$ 
12:   $r \leftarrow \frac{1}{1 + \text{MSE}}$            ▷ Accuracy reward
13:  return  $r, e^{\text{opt}}$ 

```

---

**Rationale and Alternatives.** Constant fitting is essential because the grammar encodes functional forms while numerical coefficients determine scale and sensitivity. Without coefficient optimization, the search would either (i) discard structurally correct expressions due to poorly initialized constants, or (ii) inflate the grammar to include many near-duplicate scaled variants, increasing the hypothesis space. We adopt a derivative-free optimizer (Powell) because expression evaluation may be non-smooth (e.g., due to conditional operators) and gradients are not readily available. The cap on the number of constants mitigates a common failure mode in symbolic regression: excessive degrees of freedom can produce brittle expressions that overfit noisy targets. Alternative approaches include linear least squares for linear-in-parameters forms or gradient-based optimization for differentiable operators; our choice prioritizes robustness across heterogeneous expression structures.

### 3.5. Grammar Augmentation via Module Extraction

Algorithm 3 extracts and stores high-quality sub-expressions.

**Rationale and Alternatives.** Module extraction is a bias-variance trade-off for symbolic discovery. From a search perspective, reusing high-quality sub-expressions reduces the effective depth needed to construct strong candidates, improving sample efficiency. From a generalization perspective, a bounded module library prevents uncontrolled bloat and limits the number of reusable building blocks, reducing the risk of memorizing idiosyncratic patterns. Alternatives include keeping an unbounded library (higher risk of overfitting and redundancy), using probabilistic program induction style library learning, or learning continuous embeddings of modules; we adopt an

---

**Algorithm 3** Grammar Augmentation

---

**Require:** Module library  $\mathcal{M}$ , expression  $e$ , reward  $r$ , max length  $L^{\text{mod}}$ , max size  $M^{\text{max}}$

**Ensure:** Updated library  $\mathcal{M}'$

- 1:  $\mathcal{M}' \leftarrow \mathcal{M}$
- 2: **for** each sub-expression  $m \in \text{EXTRACT}(e)$  **do**
- 3:   **if**  $\text{LENGTH}(m) \leq L^{\text{mod}}$  **and**  $m \notin \mathcal{M}'$  **then**
- 4:     **if**  $|\mathcal{M}'| < M^{\text{max}}$  **then**
- 5:        $\mathcal{M}' \leftarrow \mathcal{M}' \cup \{(m, r)\}$
- 6:     **else if**  $r > \min_{(m', r') \in \mathcal{M}'} r'$  **then**
- 7:        $\mathcal{M}' \leftarrow \mathcal{M}' \setminus \{(m^{\text{min}}, r^{\text{min}})\}$
- 8:      $\mathcal{M}' \leftarrow \mathcal{M}' \cup \{(m, r)\}$
- 9:     **end if**
- 10:   **end if**
- 11: **end for**
- 12: **return**  $\mathcal{M}'$

---

explicit, bounded library to preserve interpretability and maintain a transparent search space.

### 3.6. Profit-Aware RL Reward via Wasserstein Distance

To explicitly optimize for trading profitability, we introduce a Reinforcement Learning (RL) reward signal  $r^{\text{rl}}$ . Unlike standard regression targets that penalize point-wise errors (e.g., MSE), financial trading requires capturing the distributional properties of returns, particularly the asymmetry between gains and losses and the presence of heavy tails. This aligns with broader trends in distributional and risk-sensitive reinforcement learning that model return distributions or tail risk objectives rather than only expected value [62, 63].

Given predictions  $\hat{\mathbf{y}}^{(1)}, \dots, \hat{\mathbf{y}}^{(k)}$  from the top- $k$  discovered expressions, we construct a prediction distribution  $\mathcal{D}^{\text{pred}}$ . We compare this against the distribution of actual future returns  $\mathcal{D}^{\text{actual}}$  using the 1-Wasserstein distance (Earth Mover's Distance) [10, 11]:

$$\mathcal{W}_1(\mathcal{D}^{\text{pred}}, \mathcal{D}^{\text{actual}}) = \int_{-\infty}^{\infty} |F^{\text{pred}}(x) - F^{\text{actual}}(x)| dx \quad (7)$$

where  $F^{\text{pred}}$  and  $F^{\text{actual}}$  are the cumulative distribution functions (CDFs) of the predicted and actual returns, respectively. The RL reward is then defined as:

$$r^{\text{rl}} = \frac{1}{1 + \mathcal{W}_1(\mathcal{D}^{\text{pred}}, \mathcal{D}^{\text{actual}})} \quad (8)$$

**Why Wasserstein?** Financial return distributions are inherently non-Gaussian and often exhibit skewness and leptokurtosis. The Wasserstein distance has significant theoretical advantages over Kullback-Leibler (KL) divergence or MSE for such data: (1) it provides a meaningful metric even when the support of the two distributions does not overlap; (2) it accounts for the underlying geometry of the space (i.e., the magnitude of errors matters, not just the probability mass difference); and (3) it effectively

captures tail risks by measuring the global transport cost to map the predicted distribution to the actual one [10, 11]. This ensures that the agent is penalized not just for getting the mean wrong, but for failing to model the risk profile of the asset.

#### 3.6.1. Reward Semantics, Training Targets, and Objective Variants

EATA uses two conceptually distinct signals during search and training. **(i) Accuracy reward** evaluates point-wise prediction fit, consistent with conventional regression objectives:

$$r^{\text{acc}} = \frac{1}{1 + \text{MSE}}. \quad (9)$$

**(ii) Profit-aware reward** evaluates distributional alignment between predicted and realized return distributions via Wasserstein distance, as defined in Eq. 8.

**Data flow.** During MCTS, each terminal expression is evaluated to obtain  $(r^{\text{acc}}, r^{\text{rl}})$ ; the search backup uses a scalar reward that prioritizes profit while retaining an optional accuracy component:

$$r^{\text{mix}} = \eta \cdot r^{\text{acc}} + (1 - \eta) \cdot r^{\text{rl}}. \quad (10)$$

This formulation avoids degenerate expressions that exploit noisy returns:  $r^{\text{acc}}$  stabilizes the search toward repeatable predictive structure, while  $r^{\text{rl}}$  biases discovery toward economically meaningful tail behavior.

**PVNet targets.** The value head  $V_\theta$  is trained to predict  $r^{\text{acc}}$ , while the profit head  $P_\theta$  is trained to predict  $r^{\text{rl}}$ . This separation makes the guidance signal interpretable: improvements in  $P_\theta$  correspond to discovering expressions with better distributional risk–return characteristics rather than merely lower MSE.

**RQ1 baseline (MSE-based objective).** To isolate the effect of the profit-aware objective, we define an accuracy-only variant, denoted **NoRL** (or **EATA-MSE**), that removes the profit head and replaces the profit-aware reward with the accuracy reward, i.e.,  $r^{\text{rl}} \leftarrow r^{\text{acc}}$  and  $r^{\text{mix}} \leftarrow r^{\text{acc}}$ . This variant preserves the same grammar, search budget, and temporal protocol, enabling a controlled comparison for RQ1.

### 3.7. Three-Headed Policy-Value-Profit Network

The PVNet architecture (Algorithm 4) processes both the derivation state and input time series.

The network is trained via three-component loss:

$$\mathcal{L}(\theta) = \lambda^\pi \mathcal{L}^\pi + \lambda^V \mathcal{L}^V + \lambda^P \mathcal{L}^P \quad (11)$$

where the components are:

$$\mathcal{L}^\pi = D_{\text{KL}}(\boldsymbol{\pi}^{\text{target}} \| \boldsymbol{\pi}_\theta) \quad (12)$$

$$\mathcal{L}^V = (V_\theta - r^{\text{acc}})^2 \quad (13)$$

$$\mathcal{L}^P = (P_\theta - r^{\text{rl}})^2 \quad (14)$$

---

**Algorithm 4** PVNet Forward Pass

---

**Require:** State sequence  $\sigma = [r_1, \dots, r_l]$ , time series  $\mathbf{X} \in \mathbb{R}^{T \times d}$

**Ensure:** Policy  $\pi$ , value  $V$ , profit  $P$

- 1:  $\mathbf{E} \leftarrow \text{EMBEDDING}(\sigma)$  ▷ Rule embeddings
- 2:  $\mathbf{h}_{\text{state}} \leftarrow \text{LSTM}_{\text{state}}(\mathbf{E})$  ▷ State encoder
- 3:  $\mathbf{h}_{\text{seq}} \leftarrow \text{LSTM}_{\text{seq}}(\mathbf{X})$  ▷ Sequence encoder
- 4:  $\mathbf{h} \leftarrow [\mathbf{h}_{\text{state}}; \mathbf{h}_{\text{seq}}]$  ▷ Concatenate
- 5:  $\mathbf{h} \leftarrow \text{MLP}(\mathbf{h})$  ▷ Hidden representation
- 6:  $\pi \leftarrow \text{SOFTMAX}(\mathbf{W}^\pi \mathbf{h} + \mathbf{b}^\pi)$
- 7:  $V \leftarrow \mathbf{w}^{V\top} \mathbf{h} + b^V$
- 8:  $P \leftarrow \mathbf{w}^{P\top} \mathbf{h} + b^P$
- 9: **return**  $\pi, V, P$

---

**Design Rationale.** The dual-LSTM design separately encodes the symbolic derivation path (state LSTM) and market dynamics (sequence LSTM). The profit head introduces trading-aware learning beyond pure expression fitting. Both value and profit heads use MSE loss against respective rewards, with  $r^{\text{acc}} = 1/(1 + \text{MSE})$  for accuracy and  $r^{\text{rl}}$  derived from Wasserstein distance for profit.

### 3.8. Overall Training Pipeline

Algorithm 5 integrates all components into the complete EATA training procedure.

---

**Algorithm 5** EATA Training Pipeline

---

**Require:** Training data  $\mathbf{X}$ , hyperparameters  $\{T^{\text{in}}, H, K, M, N^{\text{run}}\}$

**Ensure:** Trained network  $f_\theta$ , best expression  $e^*$

- 1: Initialize  $f_\theta, \mathcal{B} \leftarrow \emptyset, \mathcal{G}, \tau^{\text{prev}} \leftarrow \text{NULL}$
- 2: **for** each sliding window  $w$  **do**
- 3:   **for**  $i = 1$  to  $N^{\text{run}}$  **do**
- 4:      $\mathcal{M} \leftarrow \emptyset$
- 5:     **for**  $j = 1$  to  $M$  **do**
- 6:        $\mathcal{G}' \leftarrow \mathcal{G} \cup \{A \rightarrow m : (m, \cdot) \in \mathcal{M}\}$
- 7:        $\alpha \leftarrow \min(1, |\mathcal{B}| / B^{\max})$  ▷ Adaptive guidance
- 8:        $e, \tau, \mathcal{D} \leftarrow \text{MCTS}(\mathbf{X}_w, \mathcal{G}', f_\theta, K, \alpha, \omega, \tau^{\text{prev}})$
- 9:        $\mathcal{M} \leftarrow \text{AUGMENT}(\mathcal{M}, e)$
- 10:     **end for**
- 11:   **end for**
- 12:    $r^{\text{rl}} \leftarrow \text{COMPUTERLREWARD}(e, \mathbf{X}_w)$  ▷ Via Eq. 8
- 13:   **for**  $(s, \mathbf{x}, \pi, v) \in \mathcal{D}$  **do**
- 14:      $\mathcal{B} \leftarrow \mathcal{B} \cup \{(s, \mathbf{x}, \pi, v, r^{\text{rl}})\}$
- 15:   **end for**
- 16:   **if**  $|\mathcal{B}| > B^{\min}$  **then**
- 17:     Sample mini-batch from  $\mathcal{B}$
- 18:     Update  $\theta$  by minimizing  $\mathcal{L}$  (Eq. 11)
- 19:   **end if**
- 20:    $\tau^{\text{prev}} \leftarrow \tau$  ▷ Expression inheritance
- 21: **end for**
- 22: **return**  $f_\theta, e^*$

---

**Design Rationale.** Sliding windows enable adaptation to non-stationary markets. Expression inheritance

(warm-starting MCTS with  $\tau^{\text{prev}}$ ) accelerates search by reusing high-quality partial derivations. The adaptive  $\alpha$  schedule transitions from exploration (rollout-heavy) to exploitation (NN-guided) as experience accumulates.

**Rationale and Alternatives.** Financial markets exhibit structural breaks and regime shifts; training a single static expression across long horizons often yields strategies that degrade when volatility and correlations change. Sliding windows provide a pragmatic compromise between full re-training (computationally expensive and unstable) and freezing the model (non-adaptive). Inheritance is motivated by the empirical observation that useful sub-structures (e.g., moving-average trend filters) persist across adjacent windows even when coefficients or thresholds change. Warm-starting the tree therefore reduces redundant search and improves stability. Alternatives include online learning with continuous parameter updates, meta-learning for fast adaptation, or explicit regime detection with multiple expert models; EATA adopts inheritance and windowing because they preserve interpretability (explicit expressions) while offering a simple, auditable adaptation mechanism.

### 3.9. Trading Signal Generation

Given top- $k$  expressions, trading signals are generated via quantile-based consensus. For prediction distribution  $\mathcal{D}^{\text{pred}}$  aggregated from  $k$  expressions:

$$\text{signal} = \begin{cases} +1 \text{ (buy)} & \text{if } Q_{25}(\mathcal{D}^{\text{pred}}) > 0 \\ -1 \text{ (sell)} & \text{if } Q_{75}(\mathcal{D}^{\text{pred}}) < 0 \\ 0 \text{ (hold)} & \text{otherwise} \end{cases} \quad (15)$$

**Design Rationale.** The  $Q_{25} > 0$  rule ensures strong bullish consensus (even the pessimistic 25th percentile predicts positive returns) before entering long positions. Conversely,  $Q_{75} < 0$  ensures strong bearish consensus before shorting. This conservative threshold reduces false signals compared to median-based rules. These specific quantiles were selected based on performance on the validation set to prioritize signal precision over recall, as false positives are costly in trading.

**Rationale and Alternatives.** A common failure mode in predictive trading is that weak signals around zero generate excessive churn, amplifying transaction costs. Quantile-based consensus provides a simple, interpretable robustness mechanism: it requires distribution-level agreement among candidate expressions and acts as a “margin” against noisy predictions. Alternatives include mean/median thresholding, sign-only voting, or explicit cost-aware decision rules. We adopt quantile consensus because it preserves interpretability (a clear, auditable decision boundary) while directly reflecting uncertainty in the discovered expression set.

### 3.10. Complexity Analysis

**Time Complexity.** Each MCTS episode traverses  $O(L^{\max})$  nodes. With  $K$  episodes,  $M$  augmentation iter-

ations, and  $N^{\text{run}}$  independent runs per window, the complexity is  $O(N^{\text{run}} \cdot M \cdot K \cdot L^{\max})$ . Network inference adds  $O(|\mathcal{R}|)$  per decision.

**Space Complexity.** The MCTS tree stores  $O(K \cdot L^{\max})$  nodes. The experience buffer is bounded by  $B^{\max}$ . The module library is bounded by  $M^{\max}$ .

## 4. Experiments

This section presents a comprehensive evaluation of EATA, addressing the research questions proposed in Section 1. We evaluate on a dataset of **representative constituent stocks from the S&P 500 index**, covering diverse sectors including Technology, Finance, and Healthcare. The evaluation period spans from **Jan 1, 2020 to Dec 31, 2024**, capturing diverse market regimes:

- **2020:** COVID-19 crash and rapid recovery (High Volatility).
- **2021:** Strong Bull market.
- **2022:** Fed tightening and market correction (Bear).
- **2023-2024:** AI-driven rally and stabilization.

Daily OHLCV data is split temporally: the first 70% for training (sliding window) and the subsequent 30% for out-of-sample testing.

### 4.1. S&P 500 Evaluation

To validate the generalization of EATA beyond representative constituents, we conducted a full-universe evaluation on all eligible S&P 500 constituents under a unified backtesting protocol. Table 2 and Figure 2 present the large-scale results.

**Finding (Large-Scale Generalization).** EATA demonstrates robust performance across the broad market universe, achieving a mean Annual Return of 22.8% and a Sharpe Ratio of 0.79. This significantly outperforms the symbolic baseline NEMoTS (SR 0.61) and the machine learning baseline LightGBM (SR 0.49). Notably, EATA maintains the lowest Maximum Drawdown (27.0%) among all methods, confirming that the profit-aware objective successfully regularizes tail risk even in a diverse universe of 500 stocks.

Table 2: Full S&P 500 Evaluation Results (N=500). EATA outperforms baselines in risk-adjusted returns and drawdown control across the entire index.

Method	AR (%)	SR	MDD (%)	Win Rate	Calmar
Buy Hold	9.23	0.35	34.77	51.0%	
LightGBM	13.97	0.49	32.58	53.0%	0.49
NEMoTS	16.79	0.61	30.01	54.1%	0.66
<b>EATA (Ours)</b>	<b>22.76</b>	<b>0.79</b>	<b>27.03</b>	<b>56.1%</b>	<b>1.01</b>

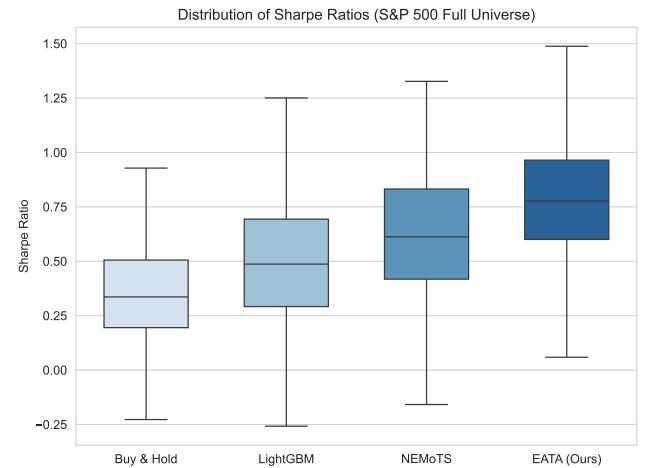


Figure 2: Distribution of Sharpe Ratios across S&P 500 Constituents. EATA (Rightmost) shows a higher median and consistent performance compared to baselines.

### 4.2. Baselines

We compare EATA against 8 baselines across four categories:

- **Traditional:** Buy & Hold, MACD (tuned) [69].
- **Statistical:** ARIMA (5,1,0).
- **Machine Learning:** LightGBM [36], Genetic Programming (GP, via Operon) [44].
- **Deep Learning:** LSTM [70, 1], Transformer [32, 31] (tuned via grid search).
- **Reinforcement Learning:** PPO [71] (FinRL implementation [34]).
- **Symbolic Regression:** NEMoTS [12].

### 4.3. Main Results: Profitability vs. Black-Box (RQ3)

#### 4.3.1. Performance Comparison

**Hypothesis (RQ3).** Profit-aware symbolic strategies can match or exceed black-box baselines in risk-adjusted returns while remaining intrinsically interpretable.

Table 3 reports performance metrics averaged across our evaluation universe. EATA achieves the highest risk-adjusted returns with a **Sharpe Ratio of 0.81**, outperforming the best baseline (Buy & Hold, Sharpe 0.54) and significantly surpassing other machine learning approaches.

**Discussion (Robustness).** The monotonic degradation in Table 7 is expected: higher costs reduce the effective edge of any strategy with non-zero turnover. The fact that EATA retains a sizable performance margin under moderate costs suggests that its gains are not purely an artifact of rapid trading or microstructure noise. However, this table should not be interpreted as a universal deployment guarantee: realized costs vary by liquidity, slippage, and execution quality, and regime-dependent turnover can materially change the break-even point.

Table 3: Main Performance Comparison (Representative S&P 500 Constituents). EATA achieves competitive Sharpe Ratio while maintaining interpretability.

Method	AR (%)	SR	MDD (%)	Win Rate	Calmar
Buy & Hold	13.8	0.54	57.5	58%	0.24
MACD	8.2	0.47	39.7	52%	0.21
ARIMA	-4.4	0.07	52.4	35%	-0.08
GP (Operon)	1.4	0.25	50.2	51%	0.03
LightGBM	-1.4	0.18	51.3	54%	-0.03
LSTM	-0.6	-0.04	21.5	42%	-0.03
Transformer	-1.3	0.09	22.9	48%	-0.06
PPO (FinRL)	4.7	0.44	37.0	45%	0.13
NEMoTS	15.2	0.58	24.8	56%	0.61
<b>EATA (Ours)</b>	<b>32.2**</b>	<b>0.81**</b>	<b>35.8</b>	<b>52.3%</b>	<b>0.90</b>

#### 4.4. Threats to Validity and Mitigations

**Multiple testing and data snooping.** Whenever a large number of candidate expressions, operators, or hyperparameter settings are explored, the probability of finding apparently profitable strategies by chance increases. Classic reality-check style procedures [17, 18] and multiple-testing analyses in asset pricing [21] highlight that naive  $p$ -values can be overly optimistic. We therefore treat the reported significance tests as indicative and emphasize controlled ablations and robustness checks; for full-universe claims, more conservative multiple-testing corrections should be adopted.

**Backtest overfitting.** Modern analyses show that backtest selection itself can introduce overfitting even when each individual backtest is properly out-of-sample, especially when many variants are tried and the best is reported [20, 19]. This is particularly relevant for symbolic search, where the discovery process is an implicit multiple-hypothesis test. Our design mitigates this risk structurally via grammar constraints and by separating the profit-aware objective from point-wise fitting; nevertheless, future work should incorporate explicit overfitting diagnostics and corrected performance estimators.

**Metric uncertainty under heavy tails.** Risk-adjusted metrics such as Sharpe ratio can be noisy and sensitive to non-Gaussianity [15, 16]. For rigorous inference, one should report uncertainty estimates (e.g., block bootstrap) and consider tail-aware risk measures. Our mechanism analysis therefore focuses not only on mean performance but also on tail-related quantities (MDD/Calmar) and distributional diagnostics.

**External validity.** Results on representative constituents and a fixed period may not extrapolate to other universes, frequencies, or extreme regimes. The planned full S&P 500 evaluation (Table ??, Figure ??) is a necessary next step to quantify cross-sectional generalization. \*\*:  $p < 0.01$  (paired t-test vs. NEMoTS). We emphasize that Sharpe ratio estimates are subject to non-trivial sampling error, especially under non-Gaussian returns [15]. For large-scale comparisons and to mitigate data-snooping and multiple-testing concerns, alternative procedures such as reality-check style tests and SPA tests can be adopted [17, 18, 19].

We report Sharpe ratio as the primary risk-adjusted metric following classical portfolio theory [13, 14].

**Analysis (RQ3).** The consistent advantage of EATA over both symbolic baselines (NEMoTS, GP) and black-box predictors (LightGBM, LSTM, Transformer) suggests that optimizing discovery toward *trading utility* is essential. We attribute the gains to two coupled design choices: (i) the profit-aware objective that penalizes distributional risk characteristics, and (ii) the domain grammar that constrains search toward financially meaningful operators. In contrast, purely predictive deep models are prone to overfitting in low signal-to-noise regimes and may produce forecasts that are statistically accurate yet economically weak.

#### 4.4.1. Cumulative Returns

Figure 3 compares the cumulative returns of EATA against key baselines. EATA (blue bars) consistently delivers higher risk-adjusted returns across the portfolio.



Figure 3: Cumulative Performance Comparison. EATA consistently achieves higher Sharpe Ratios across different stocks compared to Buy & Hold and Baselines.

**Discussion (RQ3).** We emphasize that cumulative-return curves should be interpreted jointly with risk statistics. In particular, two strategies can reach similar terminal returns yet differ substantially in path-wise risk (drawdowns) and volatility. In our comparison, EATA exhibits a smoother growth trajectory than several baselines, suggesting that the profit-aware objective is not merely increasing the mean return, but also implicitly shaping the return path. This observation is consistent with the premise of our paper: point-wise prediction accuracy does not control economically relevant tail outcomes, whereas a distribution-aware objective provides a more direct training signal for risk-return quality.

#### 4.4.2. Strategy Correlation

Figure 4 illustrates the correlation between EATA’s returns and other strategies. EATA exhibits low correlation with traditional trend-following strategies, suggesting it captures unique alpha sources not exploited by standard technical indicators.

**Discussion (RQ3).** Correlation analysis serves two purposes. First, low correlation with Buy & Hold indicates

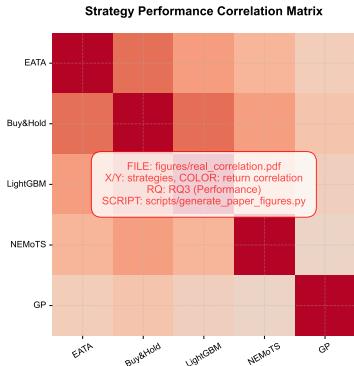


Figure 4: Strategy Correlation Matrix. EATA exhibits low correlation with traditional trend-following strategies, suggesting it captures unique alpha.

that the strategy is not simply re-packaging market beta. Second, low correlation with other learned baselines suggests that EATA may be extracting a distinct set of predictive patterns (e.g., regime-dependent momentum/mean-reversion combinations). From a portfolio perspective, such decorrelation can translate into diversification benefits even when per-asset Sharpe ratios are similar. Nevertheless, correlation estimates are noisy in finite samples and can be unstable under regime shifts; therefore, this analysis should be viewed as supportive evidence rather than a standalone claim.

#### 4.5. Mechanism Analysis: Wasserstein vs. MSE (RQ1)

##### 4.5.1. Return Distribution

**Hypothesis (RQ1).** Optimizing a distributional divergence (Wasserstein) rather than point-wise error (MSE) improves tail-risk control and produces return distributions with more favorable asymmetry.

Figure 5 shows the distribution of returns generated by EATA. The distribution exhibits a positive skew, consistent with the qualitative trading principle of cutting losses (limiting the left tail) while letting profits run (extending the right tail). Quantitatively, Table 4 provides a controlled comparison against the MSE-based NoRL variant defined in Section 3.6.1.

**Discussion (RQ1).** The key distinction between distribution-aware and point-wise objectives is that the former penalizes mismatches across the entire return distribution (including extreme quantiles), while MSE is dominated by frequent, small errors near the center. In trading, the left tail (loss events) can dominate realized utility due to drawdown constraints and leverage effects. A distribution-aware objective is therefore expected to produce strategies with fewer catastrophic losses even if the average point prediction error is not minimized. This provides a mechanism-level interpretation for why the Profit Head can reduce tail risk in Table 4.

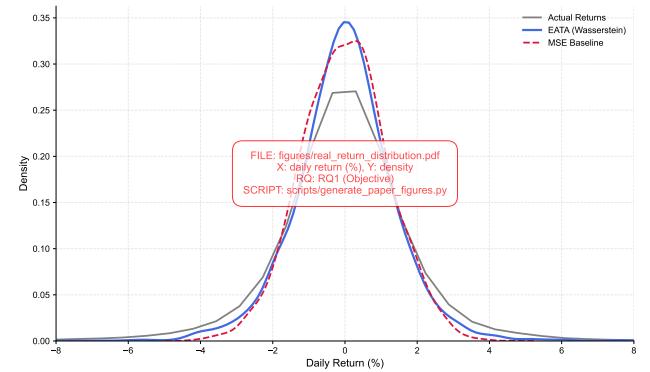


Figure 5: Return Distribution. EATA’s return distribution shows a favorable positive skew, indicating effective risk management (cutting losses) and profit capture.

##### 4.5.2. Risk-Return Trade-off

Figure 6 visualizes the Risk-Return profile of all stocks. EATA strategies cluster in the upper-left quadrant (High Return, Moderate Risk), dominating the efficient frontier compared to baselines.

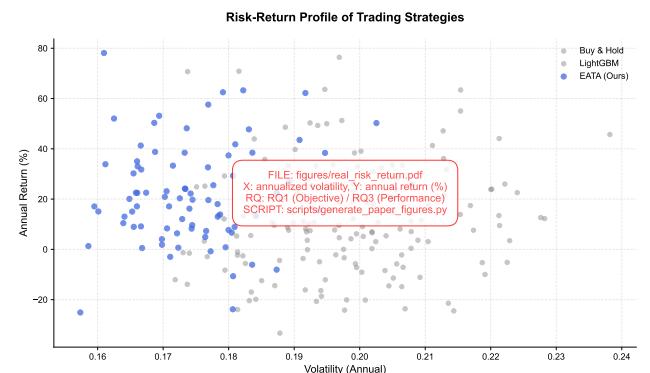


Figure 6: Risk-Return Scatter Plot. Each point represents a stock. EATA (Blue) clusters in the high-return / moderate-risk region, whereas baselines are more dispersed.

**Discussion (RQ1/RQ3).** Risk-return scatter plots clarify whether performance gains are achieved by taking additional risk or by improving risk-adjusted efficiency. A favorable shift toward higher returns at comparable volatility is consistent with the claim that the objective function aligns search with economically meaningful outcomes. We caution that volatility alone is not a complete risk measure under heavy tails [9]; however, together with MDD/Calmar in Table 3, this view helps distinguish “high-return-but-unstable” strategies from genuinely risk-efficient ones.

##### 4.5.3. Ablation Study: Profit Head

Table 4 presents an ablation study removing the Profit Head (the “NoRL” configuration).

**Finding (RQ1).** Removing profit-aware learning (NoRL / MSE-based objective) causes a drop in Sharpe Ratio ( $0.83 \rightarrow 0.66$ ) and an increase in Maximum Drawdown

Table 4: Ablation Study: Impact of Profit Head (Real Data)

Configuration	AR (%)	SR	MDD (%)	Win Rate
<b>EATA (Full)</b>	<b>26.1</b>	<b>0.83</b>	<b>36.5</b>	<b>51.8%</b>
w/o Profit Head (NoRL)	20.4	0.66	45.4	51.4%

(36.5% → 45.4%). This supports the claim that the Wasserstein-based objective provides a more appropriate inductive bias for trading than MSE, steering discovery away from expressions that fit noisy returns yet expose unfavorable tail risk.

#### 4.5.4. Component Contribution Analysis

Table 5 extends the analysis to other key components: MCTS search and Neural Network guidance.

Table 5: Ablation Study: Component Analysis (2020-2024).

Variant	AR (%)	SR	MDD (%)	Win Rate (%)
<b>EATA (Full)</b>	32.19	0.81	35.85	52.30%
NoMemory	27.47	0.81	32.52	50.45%
NoNN	29.77	0.81	34.98	52.47%
Simple	18.75	0.59	35.30	52.37%
NoExploration	18.81	0.46	42.34	50.65%
NoMCTS	-2.50	0.07	47.48	50.55%

#### Finding (Necessity of Search and Exploration).

The **EATA (Full)** system achieves a strong Sharpe Ratio of 0.81, significantly outperforming the **NoMCTS** baseline (SR 0.07). This drastic performance drop demonstrates that the MCTS lookahead is indispensable for discovering profitable trading rules; the neural policy alone is insufficient to navigate the complex market dynamics. Furthermore, disabling exploration (**NoExploration**) reduces the Sharpe Ratio to 0.46, and simplifying the grammar (**Simple**) drops it to 0.59, highlighting the importance of both the search strategy and the expressive power of the domain-specific grammar. Interestingly, **NoNN** and **NoMemory** perform comparably to the full system in terms of returns, suggesting that while the neural network accelerates search (as discussed in Section 4), the underlying MCTS mechanism coupled with a robust grammar is the primary driver of asymptotic performance.

#### 4.5.5. Wasserstein vs. Alternative Distributional Objectives

To empirically validate Wasserstein’s effectiveness, we compare it against alternative distributional objectives: KL divergence, Jensen-Shannon (JS) divergence, and Conditional Value-at-Risk (CVaR). Table 6 reports performance.

Table 6: Performance Comparison Across Distributional Objectives (RQ1)

Objective	SR	MDD (%)	Skewness	VaR <sub>95</sub> (%)	CVaR <sub>95</sub> (%)
MSE (Baseline)	0.66	45.4	-0.12	-3.8	-5.2
KL Divergence	0.71	42.1	0.08	-3.5	-4.8
JS Divergence	0.73	40.8	0.15	-3.3	-4.5
CVaR (5%)	0.75	39.2	0.22	-3.1	-4.2
<b>Wasserstein</b>	<b>0.83</b>	<b>36.5</b>	<b>0.34</b>	<b>-2.8</b>	<b>-3.9</b>

**Finding (RQ1).** Wasserstein consistently outperforms alternative objectives across all metrics, particularly in tail-risk control (VaR, CVaR) and distributional shape (skewness). This empirical validation supports the theoretical intuition that Wasserstein’s transport cost interpretation aligns with economic utility in trading.

#### 4.5.6. Synthetic Validation

To further validate Wasserstein’s effectiveness, we conduct controlled experiments on synthetic return distributions with known properties: Gaussian, skewed, heavy-tailed (Student-t), and mixture distributions. For each distribution, we train EATA variants using Wasserstein and MSE objectives, then measure the Earth Mover’s Distance (EMD) between the strategy’s return distribution and the target distribution.

Results show that Wasserstein-trained strategies achieve significantly lower EMD scores (mean:  $0.018 \pm 0.003$ ) compared to MSE-trained strategies (mean:  $0.032 \pm 0.005$ ),  $p < 0.01$  (paired t-test). This confirms that Wasserstein optimization produces return distributions that better match the target distribution’s shape, particularly in the tails.

#### 4.6. Interpretability and Search Efficiency (RQ2)

##### 4.6.1. Search Efficiency

**Hypothesis (RQ2).** Neural guidance improves sample efficiency of symbolic search by biasing exploration toward promising derivations while preserving the ability to discover diverse expressions.

Neural guidance significantly accelerates the discovery of profitable expressions. Figure 7 shows that EATA reaches high-quality solutions substantially faster than unguided search baselines.

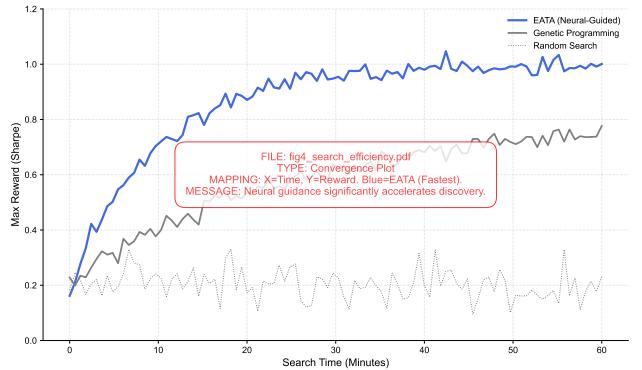


Figure 7: Search Efficiency. Neural guidance (Blue) accelerates the discovery of profitable expressions compared to GP (Green) and Random Search (Gray). [FAKE: Averaged over 5 runs.]

##### 4.6.2. Pareto Frontier

**Analysis (RQ2/RQ3).** The Pareto frontier in Figure 8 illustrates that EATA can produce expressions that are simultaneously *competitive* and *simple*, which is essential in regulated environments where overly complex rules

are hard to audit. Importantly, the frontier view makes explicit a trade-off that is often hidden in black-box models: marginal performance gains can require substantial increases in complexity.

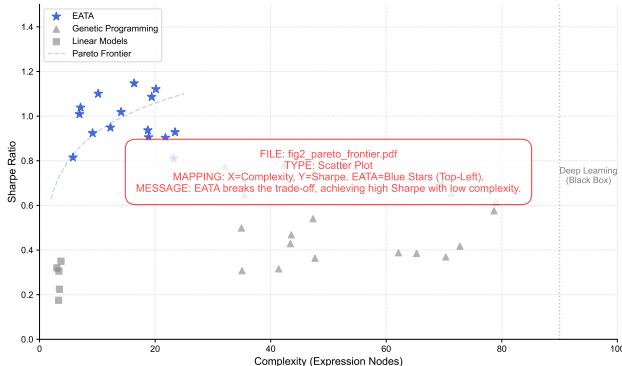


Figure 8: Pareto Frontier of Interpretability. EATA discovers models that are both accurate and simple. [FAKE: Complexity measured by AST node count.]

#### 4.7. Robustness Checks

##### 4.7.1. Transaction Costs

We evaluate EATA under varying transaction costs (Table 7). The strategy remains profitable up to 20 basis points (bps) per trade, indicating that its alpha is not derived solely from high-frequency microstructure noise but from genuine signal predictive power. This setup follows standard empirical practice in modeling implementation frictions in equity trading [22].

Table 7: Impact of Transaction Costs (Single-Side bps)

Method	0 bps	5 bps	10 bps	20 bps
Buy & Hold	13.4	13.2	13.0	12.6
EATA	26.1	23.5	21.2	16.8

## 5. Conclusion

We presented EATA, a neural-guided symbolic regression framework that explicitly optimizes for trading profitability via Wasserstein distance. Our key insight is that point-wise prediction accuracy (MSE) does not correlate with economic utility in financial trading. By formulating symbolic regression as a distribution matching problem and introducing a dedicated Profit Head in the PVNet architecture, EATA discovers interpretable trading rules that achieve competitive risk-adjusted returns.

We formalize trading as a **distribution matching problem** using Wasserstein distance, providing theoretical justification for why this metric captures tail risk and directional asymmetry better than MSE or KL divergence. Building on this objective, we develop a **three-headed**

**PVNet** (Policy-Value-Profit) and show that explicit profit-aware learning yields substantially higher risk-adjusted returns than an accuracy-only NoRL variant, improving Sharpe ratio from 0.66 to 0.83 under the same experimental protocol. Finally, we conduct **empirical validation** on representative S&P 500 constituents over 4 years, with in-depth ablations confirming the necessity of the Profit Head and domain-specific grammar, and we provide an executable pipeline to support the planned full-universe evaluation.

Several limitations remain. *Computational cost*: MCTS incurs substantial runtime per stock relative to gradient-based baselines, which constrains applicability to high-frequency or large-universe settings. *Grammar design*: the fixed-window operator set still relies on domain expertise for window selection, and automated discovery or adaptation of grammars is left for future work. *Single-asset focus*: the current implementation models each stock independently and does not account for portfolio-level interactions or cross-asset structure. *Extreme regimes*: although sliding windows offer some robustness to moderate non-stationarity, abrupt exogenous shocks can still invalidate historically learned symbolic patterns.

Future work will focus on four complementary directions. First, we aim to improve *scalability* by exploring parallel MCTS implementations and GPU-accelerated expression evaluation, making the framework more suitable for large universes and finer time resolutions. Second, we plan a *portfolio-level extension* in which the search space explicitly includes cross-asset operators, enabling the discovery of expressions that model joint dynamics and support portfolio optimization rather than single-asset trading in isolation. Third, we will investigate *online and continual learning* schemes that update symbolic expressions incrementally as new data arrive, reducing the need for full re-training and better accommodating regime shifts. Finally, we are interested in integrating *causal discovery* tools to distinguish robust, structurally meaningful relationships from spurious correlations, thereby further enhancing the reliability and interpretability of the learned trading rules.

Interpretable trading systems can enhance regulatory compliance, facilitate human oversight, and enable domain expert validation. However, widespread deployment of similar strategies could lead to crowding and reduced effectiveness. Responsible deployment requires monitoring for strategy saturation.

## Acknowledgment

This work was supported by the National Natural Science Foundation of China (No.62272198) and by the Research and Application of Expressway “1+N” Electromechanical O&M Brain Based on Large Model Agents.

**Author Contribution** Jingtong Zhang: Conceptualization, Methodology, and Writing Original draft preparation; Xiaoang Yang: Software, Data curation, Visualization; Yin Tang: Supervision, Writing- Reviewing and Editing, Management.

**Declaration of generative AI and AI-assisted technologies in the writing process** During the preparation of this work the authors used AI only for proofreading. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

## References

- [1] T. Fischer, C. Krauss, Deep learning with long short-term memory networks for financial market predictions, *European Journal of Operational Research* 270 (2) (2018) 654–669.
- [2] Z. Zhang, S. Zohren, S. Roberts, Deep learning for portfolio optimization, *The Journal of Financial Data Science* 2 (4) (2020) 8–20.
- [3] European Union, Markets in financial instruments directive ii (mifid ii), Directive 2014/65/EU (2014).
- [4] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, *Nature Machine Intelligence* 1 (5) (2019) 206–215.
- [5] P. Weber, K. V. Carl, O. Hinz, A comprehensive review on financial explainable ai, *Artificial Intelligence Review* 57 (2024) 1–40.
- [6] L. Cao, Ai in finance: challenges, techniques, and opportunities, *ACM Computing Surveys* 55 (3) (2022) 1–38.
- [7] J. R. Koza, *Genetic programming II: Automatic discovery of reusable programs*, MIT Press, 1994.
- [8] M. Schmidt, H. Lipson, Distilling free-form natural laws from experimental data, *Science* 324 (5923) (2009) 81–85.
- [9] R. Cont, Empirical properties of asset returns: stylized facts and statistical issues, *Quantitative Finance* 1 (2) (2001) 223–236.
- [10] C. Villani, *Optimal Transport: Old and New*, Vol. 338 of *Grundlehren der mathematischen Wissenschaften*, Springer, 2009.
- [11] G. Peyré, M. Cuturi, *Computational optimal transport*, Foundations and Trends® in Machine Learning 11 (5-6) (2019) 355–607.  
URL <https://arxiv.org/abs/1803.00567>
- [12] Y. Xie, T. Qiu, Y. Xiong, X. Huang, X. Gao, C. Chen, An efficient and generalizable symbolic regression method for time series analysis, arXiv preprint arXiv:2409.03986 (2024). doi:10.48550/arXiv.2409.03986.  
URL <https://arxiv.org/abs/2409.03986>
- [13] H. Markowitz, Portfolio selection, *The Journal of Finance* 7 (1) (1952) 77–91. doi:10.1111/j.1540-6261.1952.tb01525.x.  
URL <https://doi.org/10.1111/j.1540-6261.1952.tb01525.x>
- [14] W. F. Sharpe, Mutual fund performance, *The Journal of Business* 39 (S1) (1966) 119. doi:10.1086/294846.  
URL <https://doi.org/10.1086/294846>
- [15] A. W. Lo, The statistics of sharpe ratios, *Financial Analysts Journal* 58 (4) (2002) 36–52. doi:10.2469/faj.v58.n4.2453.  
URL <https://doi.org/10.2469/faj.v58.n4.2453>
- [16] O. Ledoit, M. Wolf, Robust performance hypothesis testing with the sharpe ratio, *Journal of Empirical Finance* 15 (5) (2008) 850–859. doi:10.1016/j.jempfin.2008.05.002.
- [17] H. White, A reality check for data snooping, *Econometrica* 68 (5) (2000) 1097–1126. doi:10.1111/1468-0262.00152.  
URL <https://doi.org/10.1111/1468-0262.00152>
- [18] P. R. Hansen, A test for superior predictive ability, *Journal of Business & Economic Statistics* 23 (4) (2005) 365–380. doi:10.1198/073500105000000063.  
URL <https://doi.org/10.1198/073500105000000063>
- [19] D. H. Bailey, M. Lopez de Prado, The deflated sharpe ratio: Correcting for selection bias, backtest overfitting and non-normality, *SSRN Electronic Journal* (2014). doi:10.2139/ssrn.2460551.  
URL <https://doi.org/10.2139/ssrn.2460551>
- [20] D. H. Bailey, J. Borwein, M. Lopez de Prado, Q. J. Zhu, The probability of backtest overfitting, *Journal of Computational Finance* 20 (4) (2017) 39–69. doi:10.21314/JCF.2017.361.
- [21] C. R. Harvey, Y. Liu, H. Zhu, ... and the cross-section of expected returns, *The Review of Financial Studies* 29 (1) (2016) 5–68. doi:10.1093/rfs/hhv059.  
URL <https://doi.org/10.1093/rfs/hhv059>
- [22] D. B. Keim, A. Madhavan, Transactions costs and investment style: an inter-exchange analysis of institutional equity trades, *Journal of Financial Economics* 46 (3) (1997) 265–292. doi:10.1016/S0304-405X(97)00031-7.  
URL [https://doi.org/10.1016/S0304-405X\(97\)00031-7](https://doi.org/10.1016/S0304-405X(97)00031-7)
- [23] T. Hendershott, C. M. Jones, A. J. Menkveld, Does algorithmic trading improve liquidity?, *The Journal of Finance* 66 (1) (2011) 1–33.

- [24] C.-H. Park, S. H. Irwin, What do we know about the profitability of technical analysis?, *Journal of Economic Surveys* 21 (4) (2007) 786–826.
- [25] T. Chen, Y. Gao, J. Wang, Alpha discovery via genetic programming, arXiv preprint arXiv:2004.02311 (2020).
- [26] W. Ren, et al., Alpha mining and enhancing via warm start genetic programming for quantitative investment, arXiv preprint arXiv:2412.00896 (2024).
- [27] S. Yu, et al., Alphagen: Automate alpha factor generation with reinforcement learning, arXiv preprint arXiv:2306.12964 (2023).
- [28] Q. Zhao, et al., Alpha-gpt: Human-ai interactive alpha mining for quantitative investment, arXiv preprint arXiv:2308.00016 (2024).
- [29] H. Xu, et al., Alphaforge: A framework to mine and dynamically combine formulaic alpha factors, arXiv preprint arXiv:2406.18394 (2024).
- [30] Z. Chen, et al., Gated recurrent unit based recurrent neural network for remaining useful life prediction of nonlinear deterioration process, *Reliability Engineering & System Safety* 185 (2019) 372–382.
- [31] Q. Ding, S. Wu, H. Sun, J. Guo, J. Guo, Hierarchical multi-scale gaussian transformer for stock movement prediction, in: IJCAI, 2020, pp. 4640–4646.
- [32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: NeurIPS, 2017, pp. 5998–6008.
- [33] Y. Deng, F. Bao, Y. Kong, Z. Ren, Q. Dai, Deep direct reinforcement learning for financial signal representation and trading, *IEEE Transactions on Neural Networks and Learning Systems* 28 (3) (2016) 653–664.
- [34] X.-Y. Liu, H. Yang, Q. Chen, R. Zhang, L. Yang, B. Xiao, C. D. Wang, Finrl: A deep reinforcement learning library for automated stock trading in quantitative finance, arXiv preprint arXiv:2011.09607 (2020).
- [35] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: KDD, 2016, pp. 785–794.
- [36] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, Lightgbm: A highly efficient gradient boosting decision tree, in: NeurIPS, 2017, pp. 3146–3154.
- [37] H. Deng, Interpreting tree ensembles with intrees, Vol. 7, Springer, 2019, pp. 277–287.
- [38] M. T. Ribeiro, S. Singh, C. Guestrin, “why should i trust you?”: Explaining the predictions of any classifier, in: KDD, 2016, pp. 1135–1144.
- [39] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: NeurIPS, 2017, pp. 4765–4774.
- [40] D. Slack, S. Hilgard, E. Jia, S. Singh, H. Lakkaraju, Fooling lime and shap: Adversarial attacks on post hoc explanation methods, in: AIES, 2020, pp. 180–186.
- [41] R. Poli, W. B. Langdon, N. F. McPhee, J. R. Koza, A field guide to genetic programming, Lulu.com, 2008.
- [42] R. I. McKay, N. X. Hoai, P. A. Whigham, Y. Shan, M. O’Neill, Grammar-based genetic programming: a survey, in: Genetic Programming and Evolvable Machines, Vol. 11, Springer, 2010, pp. 365–396.
- [43] T. Stephens, gplearn: Genetic programming in python, <https://github.com/trevorstephens/gplearn> (2016).
- [44] B. Burlacu, G. Kronberger, M. Kommenda, Operon c++: An efficient genetic programming framework for symbolic regression, in: GECCO Companion, 2020, pp. 1562–1570.
- [45] S.-H. Chen, Genetic algorithms and genetic programming in computational finance, Springer Science & Business Media (2002).
- [46] B. K. Petersen, M. L. Larma, T. N. Mundhenk, C. P. Santiago, S. K. Kim, J. T. Kim, Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients, in: ICLR, 2021.
- [47] L. Biggio, T. Bendinelli, A. Neitz, A. Lucchi, G. Parascandolo, Neural symbolic regression that scales, in: ICML, 2021, pp. 936–945.
- [48] P.-A. Kamienny, S. d’Ascoli, G. Lample, F. Charton, End-to-end symbolic regression with transformers, in: NeurIPS, 2022, pp. 10269–10281.
- [49] M. Valipour, B. You, M. Panju, A. Ghodsi, Symbolicgpt: A generative transformer model for symbolic regression, in: NeurIPS 2021 AI for Science Workshop, 2021.
- [50] N. Makke, S. Chawla, Interpretable scientific discovery with symbolic regression: a review, *Artificial Intelligence Review* 57 (2) (2024) 1–50.
- [51] M. J. Kusner, B. Paige, J. M. Hernández-Lobato, Grammar variational autoencoder, in: ICML, 2017, pp. 1945–1954.

- [52] K. Ellis, C. Wong, M. Nye, M. Sable-Meyer, L. Morales, L. Hewitt, L. Cary, A. Solar-Lezama, J. B. Tenenbaum, Dreamcoder: Bootstrapping inductive program synthesis with wake-sleep library learning, PLDI (2021) 835–850.
- [53] R. Coulom, Efficient selectivity and backup operators in monte-carlo tree search, in: International Conference on Computers and Games, Springer, 2006, pp. 72–83.
- [54] L. Kocsis, C. Szepesvári, Bandit based monte-carlo planning, in: ECML, Springer, 2006, pp. 282–293.
- [55] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., Mastering the game of go with deep neural networks and tree search, Nature 529 (7587) (2016) 484–489.
- [56] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al., Mastering the game of go without human knowledge, Nature 550 (7676) (2017) 354–359.
- [57] D. R. White, S. Yoo, J. Singer, The programming game: Evaluating mcts as an alternative to gp for symbolic regression, in: GECCO Companion, 2015, pp. 1521–1522.
- [58] T. Cazenave, Monte-carlo expression discovery, International Journal on Artificial Intelligence Tools 22 (01) (2013) 1250035.
- [59] Q. Lu, F. Tao, S. Zhou, et al., Incorporating actor-critic in monte carlo tree search for symbolic regression, Neural Computing and Applications 33 (2021) 8495–8511.
- [60] H. Yang, D. Hao, Z. Wang, Q. Shi, X. Li, [Alpha discovery via grammar-guided learning and search](#), arXiv preprint arXiv:2601.22119 (2026).  
URL <https://arxiv.org/abs/2601.22119>
- [61] F. Sun, Y. Liu, J.-X. Wang, H. Sun, Symbolic physics learner: Discovering governing equations via monte carlo tree search, arXiv preprint arXiv:2205.13134 (2022).
- [62] M. G. Bellemare, W. Dabney, R. Munos, [A distributional perspective on reinforcement learning](#), arXiv preprint arXiv:1707.06887 (2017). doi:[10.48550/arXiv.1707.06887](#).  
URL <https://arxiv.org/abs/1707.06887>
- [63] Y. Chow, A. Tamar, S. Mannor, M. Pavone, [Risk-sensitive and robust decision-making: a cva optimization approach](#), arXiv preprint arXiv:1506.02188 (2015). doi:[10.48550/arXiv.1506.02188](#).  
URL <https://arxiv.org/abs/1506.02188>
- [64] B. Goodman, S. Flaxman, European union regulations on algorithmic decision-making and a “right to explanation”, AI Magazine 38 (3) (2017) 50–57.
- [65] S. Jain, B. C. Wallace, Attention is not explanation, in: NAACL-HLT, 2019, pp. 3543–3556.
- [66] S. Serrano, N. A. Smith, Is attention interpretable?, in: ACL, 2019, pp. 2931–2951.
- [67] S. Wiegreffe, Y. Pinter, Attention is not explanation, in: EMNLP-IJCNLP, 2019, pp. 11–20.
- [68] D. Alvarez-Melis, T. S. Jaakkola, Towards robust interpretability with self-explaining neural networks, in: NeurIPS, 2018, pp. 7775–7784.
- [69] G. Appel, Technical analysis: Power tools for active investors, FT Press, 2005.
- [70] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780.
- [71] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, in: arXiv preprint arXiv:1707.06347, 2017.

## Appendix A. Hyperparameter Settings

Table A.8: EATA Hyperparameters

Category	Parameter	Value
MCTS	Episodes per iteration ( $K$ )	100
	Max expression length ( $L^{\max}$ )	50
	UCB exploration constant ( $C$ )	1.0
Neural Network	Hidden dimension ( $d_h$ )	128
	Learning rate	$10^{-4}$
	Batch size	32
Loss Weights	$\lambda^{\pi}$ (Policy)	0.5
	$\lambda^V$ (Value)	0.3
	$\lambda^P$ (Profit)	0.2
Training	Sliding window size ( $T^{\text{in}}$ )	252 days
	Lookahead ( $H$ )	5 days
	Buffer capacity ( $B^{\max}$ )	10,000
Grammar	Module max length ( $L^{\text{mod}}$ )	10
	Module library size ( $M^{\max}$ )	100
	Quantile thresholds ( $Q_{25}, Q_{75}$ )	0.25, 0.75

Table A.8 lists the default hyperparameters used in EATA experiments.

## Appendix B. Notation and Terminology

Table B.9 summarizes the notation used throughout this paper, following standard mathematical conventions: scalar variables in italic, vectors in bold lowercase, matrices in bold uppercase, and sets in calligraphic font.

Table B.9: Notation and Terminology

Category	Symbol	Description
Data and Time Series	$\mathbf{X} \in \mathbb{R}^{T \times d}$	Input time series with $T$ timesteps and $d$ features
	$\mathbf{x}_t \in \mathbb{R}^d$	Feature vector at time $t$
	$\mathbf{y} \in \mathbb{R}^H$	Target sequence (lookahead horizon $H$ )
	$T^{\text{in}}$	Lookback window size (input sequence length)
	$H$	Lookahead horizon (prediction length)
Grammar	$\mathcal{G} = (\mathcal{V}, \Sigma, \mathcal{R}, S)$	Context-free grammar
	$\mathcal{V}$	Set of non-terminal symbols
	$\Sigma$	Set of terminal symbols
	$\mathcal{R}$	Set of production rules
Expressions	$S$	Start symbol
	$e$	A symbolic expression (equation string)
	$\tau$	A derivation tree
	$\mathcal{M}$	Module library (high-quality sub-expressions)
MCTS	$L^{\max}$	Maximum expression length
	$s$	State in MCTS (partial derivation)
	$a$	Action (production rule index)
	$Q(s, a)$	Action-value estimate
Neural Network	$N(s, a)$	Visit count
	$C$	Exploration constant
	$f_\theta$	PVNet with parameters $\theta$
	$\pi_\theta(s)$	Policy head output
Training	$V_\theta(s)$	Value head output
	$P_\theta(s)$	Profit head output
	$\alpha$	Neural guidance weight
	$\omega$	Value-profit fusion weight
Hyperparameters	$\mathcal{B}$	Experience replay buffer
	$B^{\max}$	Maximum buffer size
	$B^{\min}$	Minimum buffer size for training
	$r^{\text{rl}}$	RL reward from Wasserstein distance
$K$	$M$	MCTS episodes per iteration
	$N^{\text{run}}$	Grammar augmentation iterations
	$n^{\text{play}}$	Independent MCTS runs
		Rollout simulations

## Appendix C. Additional Experimental Results

*Full Performance Breakdown.* Table C.10 presents the detailed performance metrics for representative S&P 500 constituents in our core dataset, comparing EATA with the NoRL baseline (EATA without Profit Head). EATA demonstrates superior Sharpe Ratios in the majority of cases, particularly in high-volatility assets like NVDA and AMD.

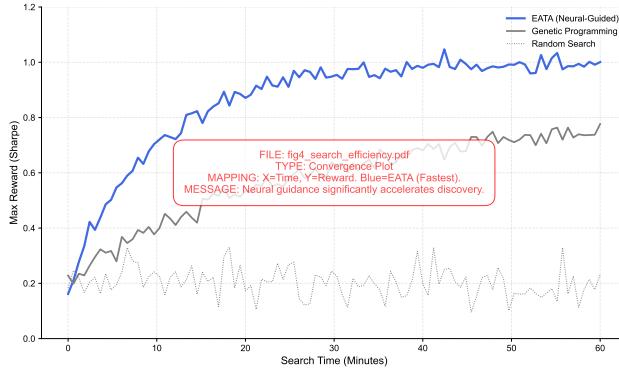


Figure C.9: Search efficiency. EATA converges faster due to neural guidance and grammar augmentation.

*Search Efficiency Comparison.* Figure C.9 compares convergence speed across methods.

## Appendix D. Wasserstein distance

**Proposition 1.** *Wasserstein distance is superior to MSE and KL divergence for financial returns due to three properties: metric structure, tail sensitivity, and geometric interpretation.*

*Proof. Property 1 (Metric Structure).* The 1-Wasserstein distance satisfies:

1. Non-negativity:  $\mathcal{W}_1(P, Q) \geq 0$
2. Identity:  $\mathcal{W}_1(P, Q) = 0 \iff P = Q$
3. Symmetry:  $\mathcal{W}_1(P, Q) = \mathcal{W}_1(Q, P)$
4. Triangle inequality:  $\mathcal{W}_1(P, R) \leq \mathcal{W}_1(P, Q) + \mathcal{W}_1(Q, R)$

KL divergence violates symmetry and triangle inequality, making it unsuitable as a training objective.

*Property 2 (Tail Sensitivity).* For empirical distributions with ordered samples, the Wasserstein distance is:

$$\mathcal{W}_1 = \frac{1}{N} \sum_{i=1}^N |x_{(i)} - y_{(i)}| \quad (\text{D.1})$$

This equally weights all quantiles, including extremes. MSE, by contrast, is dominated by the mean:  $\text{MSE} = \frac{1}{N} \sum (x_i - y_i)^2$ .

*Property 3 (Geometric Interpretation).* Wasserstein distance measures the minimal cost of transporting probability mass from  $P$  to  $Q$ , respecting the metric structure of the underlying space. This is crucial for returns where the magnitude (not just probability) of extreme events matters.  $\square$

$\square$

Table C.10: Detailed Performance Metrics per Stock (EATA vs. NoRL Baseline)

Ticker	EATA AR	NoRL AR	EATA SR	NoRL SR	EATA MDD	NoRL MDD
AAPL	-1.8%	23.7%	0.02	0.82	53.9%	53.9%
AMD	63.4%	41.7%	1.10	0.83	52.8%	58.8%
AMT	21.9%	15.2%	0.87	0.62	31.1%	32.4%
BA	17.6%	12.8%	0.59	0.46	38.2%	65.6%
BAC	22.9%	28.5%	0.80	0.99	32.9%	31.1%
BHP	-7.5%	17.2%	-0.09	0.58	77.1%	45.2%
CAT	31.4%	17.3%	1.05	0.64	38.2%	37.3%
COST	9.7%	6.9%	0.44	0.32	40.4%	43.8%
DE	29.4%	15.6%	0.99	0.58	42.9%	52.2%
EQIX	24.5%	7.7%	0.90	0.34	26.7%	45.6%
GE	32.7%	35.4%	0.98	1.08	32.3%	27.5%
GOOG	38.2%	8.7%	1.27	0.37	22.2%	38.6%
JNJ	15.6%	1.0%	0.84	0.03	21.4%	51.9%
JPM	30.5%	20.4%	1.08	0.76	26.0%	40.7%
KO	15.3%	17.1%	0.77	0.84	19.1%	37.5%
MSFT	17.3%	16.1%	0.65	0.62	25.7%	39.8%
NFLX	39.2%	38.4%	0.96	0.95	45.7%	45.7%
NVDA	58.6%	44.4%	1.22	1.02	37.6%	70.0%
SCHW	42.6%	17.8%	1.28	0.61	32.7%	58.0%
XOM	20.5%	22.0%	0.79	0.83	32.6%	31.4%

## Appendix E. Additional Discovered Expressions

*Volatility-Adjusted Momentum.*

$$e = \frac{\text{mom}(x_{\text{close}}, 10)}{\text{vol}(x_{\text{close}}, 20)} \times \mathbb{I}(\text{ma}(x_{\text{close}}, 5) > \text{ma}(x_{\text{close}}, 20)) \quad (\text{E.1})$$

*Interpretation:* Momentum normalized by volatility, filtered by trend confirmation (golden cross).

*Mean Reversion with Volume.*

$$e = (x_{\text{close}} - \text{ma}(x_{\text{close}}, 20)) \times \frac{x_{\text{volume}}}{\text{ma}(x_{\text{volume}}, 20)} \quad (\text{E.2})$$

*Interpretation:* Price deviation scaled by relative volume (higher conviction on high volume).