Modern Machine Learning Tools in Finance: A Critical Perspective

Franklin Allen[1,2], Marcin Kacperczyk[1,2,3], and K S Sesh Kumar[1]

[1]Imperial Business School
[2]CEPR
[3]ECGI

September 3, 2025

Abstract

This paper discusses why machine learning approaches consistently fail to generate profitable trading strategies, viewing this challenge through the lens of the Adaptive Market Hypothesis. We argue that these failures stem from a fundamental misalignment: while machine learning models assume stationarity, financial markets are inherently evolutionary systems where trading opportunities exist as transient ecological niches that disappear through competitive exploitation. Our comprehensive review of recent architectures designed to address non-stationarity—including Patch-based Time Series Transformers, non-stationary transformers, and neural hidden Markov models—reveals that while each excels under specific conditions, none successfully navigates the full complexity of market evolution. This limitation appears fundamental, arising from market reflexivity: successful strategies inevitably alter the very environment they seek to exploit.

Keywords: Adaptive Market Hypothesis, Non-stationarity, Regime switching, Neural architectures, Market microstructure

1 Introduction

Two strands of academic literature have analyzed optimal investment strategies, one in computer science and one in finance. A striking disconnect exists between the machine learning and finance research communities, with each discipline advancing in isolation without fully appreciating the other's contributions and constraints. The Machine Learning (ML) community, publishing in venues like Neural Information Processing Systems (NeurIPS), International Conference on Machine Learning (ICML), International Conference for Learning Represenation (ICLR) and journals like Journal for Machine Learning Research (JMLR), treats financial markets as technical prediction problems, developing increasingly sophisticated architectures— Patch-based Time Series Transformers (PatchTST), non-stationary transformers, neural Hidden Markov Models (HMMs), meta-learning frameworks—without considering market efficiency theory or the reflexive nature of trading. Meanwhile, the finance literature in journals such as Journal of Finance, Journal of Financial Economics, Review of Financial Studies, Review of Finance, and Management Science, remains with recent empirical studies predominantly employing Long Short-Term Memory networks (LSTMs) and basic ensemble methods while largely ignoring cutting-edge adaptive architectures that could potentially address non-stationarity challenges. This parallel but disconnected evolution represents a missed opportunity: ML researchers lack the economic intuition to understand why their technically superior models fail in practice, while finance researchers may be overlooking architectures specifically designed for the non-stationary environments they study. This paper bridges these isolated research streams, demonstrating that neither technical sophistication alone (the ML approach) nor economic theory without adaptive mechanisms (the traditional finance approach) can succeed in adaptive markets. By considering insights from both disciplines, we argue that the failure of machine learning in finance is not a technical problem to be solved with better algorithms, nor merely an efficiency issue as finance theory might suggest, but rather a fundamental challenge arising from the interaction between algorithmic adaptation and market evolution.

Machine learning fails to maintain consistently profitable performance in live trading environments, despite unprecedented computational power and data availability. This persistent

failure of trading strategies suggests that the problem lies not in technical sophistication but in fundamental assumptions about market dynamics. While machine learning has revolutionized fields from image recognition to natural language processing, its application to financial markets tells a different story—one of initial promise followed by decay.

The pattern is consistent: sophisticated ML approaches suffer the same fate as their simpler predecessors, experiencing initial success that progressively erodes until the strategy becomes obsolete. This recurring cycle suggests that markets possess characteristics fundamentally different from other domains where machine learning thrives. We argue that in financial contexts there is persistent non-stationarity of various kinds whereas in areas where ML has been successful such as image recognition, this is not the case.

Lo's (2004) Adaptive Market Hypothesis (AMH), which is a refinement of Fama's (1970) Efficient Market Hypothesis (EMH), offers a compelling framework for understanding these failures. Rather than viewing markets as static equilibrium states, AMH conceptualizes them as evolutionary systems where efficiency emerges dynamically from competitive interactions among heterogeneous participants. In this framework, trading opportunities are ecological niches that appear when environmental conditions shift faster than participants can adapt, only to vanish as exploitation eliminates the underlying inefficiency.

This evolutionary perspective illuminates why even the most sophisticated strategies ultimately fail. The quantitative crisis of August 2007 exemplifies this dynamic well (Khandani and Lo, 2007). Multiple funds independently converged on similar statistical arbitrage strategies through rigorous research, yet this very success led to catastrophic losses when crowding triggered simultaneous deleveraging. The pattern has repeated many times historically, from the rise and fall of technical analysis to the current challenges facing machine learning architectures.

Recognizing these dynamics, recent ML research has pivoted toward architectures explicitly designed for non-stationary environments. PatchTST (Nie et al., 2023) employs multi-scale pattern recognition through temporal segmentation to capture evolving patterns. Non-stationary transformers (Liu et al., 2022) directly model distribution shifts to detect when relationships change. Neural implementations of HMMs blend deep learning with regime detection capabilities, while reinforcement learning agents (Sutton and Barto, 2018; Jiang

et al., 2017; Deng et al., 2017) continuously update policies through market interaction.

Yet each approach addresses only specific aspects of market evolution, missing a crucial insight: markets don't simply change—they change in response to how they're being traded. This reflexivity creates a fundamental challenge that no current ML architecture fully addresses.

This paper makes three contributions. First, we synthesize evidence on strategy lifecycle dynamics across the entire evolution of systematic trading, from technical analysis through algorithmic trading to modern machine learning, demonstrating how the progression from discovery through decay to obsolescence follows predictable patterns regardless of technological sophistication. Second, we provide a comprehensive analysis of modern ML approaches designed for non-stationarity, evaluating both their theoretical foundations and practical limitations. Third, we propose a number of methods for empirical validation that could identify which adaptive approaches, if any, merit deployment in practical trading applications.

To fully appreciate why modern machine learning faces these challenges, we must first understand the historical pattern of strategy evolution that repeat across every generation of trading technology.

Historical Context of Strategy Evolution. The history of systematic trading reveals an uncomfortable truth: each generation of approaches promises to overcome its predecessors' limitations, yet all eventually succumb to the same fundamental force—competitive adaptation. This progression demonstrates that technological sophistication cannot prevent strategy decay when markets evolve in response to being traded.

Technical analysis dominated trading floors for decades, with practitioners convinced that price patterns could forecast future movements. Early studies considering the effectiveness of this approach suffered from data snooping bias, testing pre-selected technical indicators that may have been chosen based on prior knowledge of their historical performance (Brock et al., 1992; Sullivan et al., 1999; Levich and Thomas, 1993; Neely et al., 1997; Ready, 1997). The emergence of statistical correction methods such as White's Reality Check (White, 2000) and bootstrap-based testing (Sullivan et al., 1999) highlighted the limitations of early approaches.

Allen and Karjalainen (1999) employed genetic algorithms to discover trading rules with-

out human preconceptions, eliminating bias toward traditional indicators. They found using realistic trading costs that such trading rules designed to go in and out of the S&P 500 couldn't reliably beat a buy and hold strategy.

Quantitative methods emerged as the sophisticated response, promising scientific rigor and statistical validity. Academic finance contributed numerous anomalies seemingly ripe for exploitation: the value premium documented by Fama and French (1993), the momentum effect identified by Jegadeesh and Titman (1993), and dozens of other factors suggest that markets might contain discoverable inefficiencies. Jegadeesh and Titman (2001) confirmed momentum's persistence across decades and asset classes, while Asness et al. (2013) demonstrated international robustness for value and momentum strategies, showing that the initial results were improvessive. Baltas (2017) also showed that various enhancements could further improve returns.

Yet these strategies traced the same lifecycle as technical analysis. McLean and Pontiff (2016) documented a sobering finding: strategy returns decline by approximately 58% following academic publication, with decay beginning even before publication as practitioners independently discover similar patterns. The mechanism is predictable: initial exploitation by traders with minimal market impact (Tóth et al., 2011), followed by increasing capital allocation as knowledge spreads, ultimately compressing profit margins until strategies become economically obsolete.

The momentum anomaly exemplifies this trajectory. Once a robust alpha source, Daniel and Moskowitz (2016) showed that momentum strategies now suffer periodic severe reversals, particularly during market recoveries following crises. These reversals occur precisely when momentum positioning becomes crowded, triggering simultaneous unwinding that amplifies losses. Value strategies have similarly experienced unprecedented underperformance recently, sparking debate over whether this represents permanent arbitrage elimination or temporary cyclical behavior.

Algorithmic trading's rise in the 2000s promised to overcome human limitations through speed, precision, and emotionless execution. Firms invested billions of dollars in infrastructure—microwave networks, co-located servers, optimized code—to gain millisecond advantages. Yet the August 2007 quantitative crisis exposed this approach's fragility (Khandani

and Lo, 2007). Strategies successful for years failed simultaneously, not from technical flaws but from excessive convergence on identical opportunities.

Modern machine learning represents the latest attempt to discover patterns beyond human comprehension. These approaches can process alternative data sources, identify nonlinear relationships, and theoretically adapt to changing conditions. However, empirical evidence suggests they face identical fundamental challenges. Harvey et al. (2016) catalogued over 316 factors proposed to explain cross-sectional equity returns, many discovered through ML techniques. This proliferation suggests widespread false discovery rather than genuine inefficiencies. Bailey et al. (2014) formalized the mathematics of backtest overfitting, demonstrating how extensive parameter testing—endemic to machine learning—virtually guarantees spuriously profitable strategies even in random data. While Harvey and Liu (2017) and Lopez de Prado (2018) developed frameworks to address these issues, implementation remains inconsistent in practice.

This historical progression yields crucial insights for evaluating contemporary approaches. Technological sophistication doesn't prevent strategy decay—it may accelerate it by enabling faster pattern discovery and replication. The method of pattern discovery, whether through human insight, statistical analysis, or machine learning, matters less than the inevitable erosion through competitive exploitation. The challenge isn't finding patterns but continuously finding new patterns as markets evolve. Understanding this context is essential for realistic assessment of machine learning's potential in financial markets. Technical analysis, quantitative strategies, and algorithmic trading didn't fail from insufficient computational power or inadequate algorithms—they failed because markets adapt. Any successful strategy changes the market environment, creating feedback loops that eventually eliminate the exploited inefficiency. Machine learning, despite its sophistication, cannot escape this fundamental dynamic. To understand why this pattern persists and what it means for future approaches, we must examine the theoretical framework that explains these adaptive dynamics.

2 Conceptual Framework: The Adaptive Market Hypothesis

2.1 The Efficient Market Hypothesis and Its Limitations

According to Fama's (1970) EMH asset prices fully reflect all available information. He provided extensive evidence that markets are efficient. Fama (1991)synthesized further evidence showing professional managers consistently fail to outperform passive benchmarks after accounting for fees. Carhart (1997) found minimal evidence of performance persistence among mutual fund managers. If exploitable inefficiencies existed, institutional investors with substantial resources should profit consistently—yet most fail, seemingly supporting market efficiency. In this view differences in average returns are solely due to differences in risk. The Capital Asset Pricing Model (CAPM) of Sharpe (1964) and Lintner (1965) provides an early example of a model of risk-based returns.

However, this interpretation faced significant challenges. Shleifer (2000) highlights numerous well-documented anomalies—value outperformance, size effects, post-earnings announcement drift—that appear inconsistent with strict informational efficiency. These anomalies demonstrate remarkable robustness across time periods and international markets, suggesting they represent genuine economic phenomena rather than statistical artifacts.

Fama and French (1993) and Fama and French (2015) responded by proposing increasingly complex risk-based explanations, arguing that apparent anomalies represent compensation for systematic risks not captured by the CAPM's market beta. This spawned rapid growth in factor models. However, Goyal and Welch (2008) questioned the empirical robustness of many popular predictors, showing that most variables with strong in-sample performance fail to deliver meaningful out-of-sample predictability. Further, Harvey et al. (2016) and Harvey and Liu (2019) raised concerns about the credibility of several hundred proposed factors. Such findings suggest that the proliferation of factors either implies markets embed an implausibly wide array of independent risk dimensions or that much of the literature reflects data mining rather than genuine economic relationships.

Grossman and Stiglitz (1980) offered a static theoretical resolution to these contradic-

tions, demonstrating that perfectly efficient markets are internally inconsistent. If all costly information were fully reflected in prices, no one would have incentives to collect it. Markets must therefore maintain sufficient inefficiency to reward costly information acquisition while preventing persistent, risk-free arbitrage profits.

2.2 The Adaptive Market Hypothesis as Dynamic Efficiency

Lo's (2004) Adaptive Market Hypothesis elegantly reconciles these contradictions by reconceptualizing efficiency as dynamic rather than static. Under AMH, markets are not always efficient or inefficient—efficiency shifts with environmental change and participant adaptation. This view explains both the existence of profitable strategies and their eventual decay. In the model of Grossman and Stiglitz (1980), markets are sufficiently inefficient that informed traders can cover the costs of acquiring information in equilibrium. The equivalent condition under the AMH is that traders cover the costs of their investment strategies through time. In a version with different abilities, some may consistently make positive excess returns over the risk-adjusted rate, while others fail and drop out.

Lo (2004) draws explicit parallels with biological evolution: strategies compete for profits as species compete for resources. Profitable strategies survive and proliferate; unprofitable ones become extinct. Market environments continuously change, creating new niches while destroying others. This evolutionary process generates the time-varying efficiency we observe in practice.

Reflexivity adds a critical dimension to this complexity. Market participants don't merely observe and react—their actions fundamentally alter the conditions they attempt to predict. This creates a challenge that even sophisticated machine learning algorithms struggle to overcome. We observe this manifesting in several ways: successful strategies contain the seeds of their own destruction through competitive adoption; pattern persistence inversely correlates with discovery breadth; and model predictions become self-defeating prophecies when widely adopted.

Several mechanisms create temporary inefficiencies under the AMH framework. Regulatory changes generate temporary dislocations as market participants adjust. The US equity market's decimalization in 2001 enabled high-frequency trading while eliminating traditional

market-making strategies. Europe's MIFID II permanently altered market microstructure. Each reform creates transition periods where old strategies fail before new ones emerge, generating exploitable inefficiencies.

Technological innovation enables novel strategy exploration beyond human capabilities. Machine learning systems recognize complex patterns at superhuman speeds (Gu et al., 2020; Messmer, 2018) and exploit alternative data sources (Kakde et al., 2024) to identify opportunities. However, technological diffusion ensures first-mover advantages diminish as capabilities democratize, and these strategies undergo similar degradation as competitive forces restore efficiency.

Evolving market ecosystems fundamentally alter dynamics. The shift toward passive investing concentrates price discovery among fewer active participants. Commission-free platforms have sparked retail participation and coordination. These forces reshape market ecology, creating new inefficiencies while eliminating others.

Behavioral factors generate predictable mispricing that persists until arbitraged away. Sophisticated traders employ methods ranging from simple contrarian bets to deep behavioral models to capture these opportunities. Yet competition ensures even these patterns eventually erode.

2.3 Empirical Evidence for Market Adaptation

We observe that the dynamic efficiency perspective of AMH elegantly explains empirical phenomena that challenge static efficiency views. The COVID-19 pandemic shattered relationships between economic indicators and asset prices, creating unprecedented market disruptions. Equities rallied while economies contracted, growth outperformed value by historic margins, and volatility surged dramatically. Strategies assuming stable relationships suffered severe drawdowns, while Di Persio et al. (2021) showed that adaptive volatility models exhibited superior risk-adjusted performance and faster recovery.

These observations align perfectly with AMH's dynamic efficiency. Environmental changes from COVID-19 outpaced participant adaptation, creating new inefficiencies that adaptive models could exploit while static models failed.

The rise and fall of momentum strategies provides another compelling example (Jegadeesh

and Titman, 2001; Daniel and Moskowitz, 2016). Similarly, short-volatility products (XIV, SVXY) thrived from 2012 to early 2018 during calm markets with suppressed volatility and capital surge. When volatility suddenly spiked, rebalancing requirements forced buying of volatility futures, amplifying the spike and causing 90% losses within hours (Augustin et al., 2021; Bank for International Settlements, 2018). We argue that AMH's competitive dynamics explain these failures—excessive participants exploiting identical inefficiencies eventually destroy the opportunity.

The EMH cannot adequately explain why profitable strategies succeed temporarily then fail. If markets were truly efficient, such strategies shouldn't work at all; if inefficient, they should persist. AMH explains the middle ground we actually observe: strategies work temporarily while inefficiencies exist, then fail as competition eliminates opportunities.

These episodes demonstrate that markets cycle through varying efficiency states rather than maintaining constant efficiency or inefficiency. During stable periods, competition eliminates obvious inefficiencies and markets appear relatively efficient. During transitions, inefficiencies proliferate as established relationships break down and new patterns emerge. We conclude that efficiency levels depend critically on the rate of environmental change relative to participant adaptation speed. This dynamic view, rooted in evolutionary principles, provides a more complete and nuanced understanding of market behavior than static equilibrium models can.

Given this fundamental understanding of markets as adaptive systems, we must now examine how modern machine learning approaches attempt to navigate these evolutionary dynamics. The following section analyzes contemporary architectures designed to handle non-stationarity and regime changes, evaluating their theoretical promise against practical limitations in the face of market reflexivity.

3 Machine Learning in Adaptive Markets: Modern Approaches

The evolution of machine learning in finance represents a progression from simple pattern recognition to sophisticated systems attempting to navigate market adaptation. Classical methods, developed primarily between the 1960s and 2000s, remain widely used in quantitative trading due to their interpretability, computational efficiency, and theoretical foundations. They often perform competitively with more complex approaches, particularly in data-scarce regimes or when transparency is essential for risk management and regulatory compliance. The main linear models, support vector machines, decision tress and ensembles, and hidden Markov models—are well established and thus discussed in the Appendix rather than here.

Building on this foundation, this section turns to modern architectures designed specifically to address the defining challenges of financial markets: non-stationarity, regime shifts, and continuous evolution. Extending the disconnect discussed earlier, we show that while machine learning has produced increasingly sophisticated models such as PatchTST, non-stationary transformers, and neural HMMs, their application to finance often overlooks efficiency, competition, and reflexivity. Conversely, finance research tends to rely on simpler recurrent or ensemble methods that embed economic reasoning but lack technical adaptability. Both approaches therefore fail in predictable ways: ML models adapt to shifting market efficiencies, yet collapse once their adoption reshapes the market, while finance models remain too rigid for structural change.

To address this disconnect, the analysis considers each architecture not only in technical terms but also in financial context: what type of adaptivity it seeks to capture, how that adaptivity maps onto market dynamics, and why those mechanisms succeed or fail under reflexivity. In this way, we link machine learning models to the adaptive features of markets they implicitly target, clarifying both their strengths and their limitations.

3.1 Deep Learning Foundations for Financial Markets

Deep learning has transformed financial modeling by automating feature discovery from raw data, eliminating the need for hand-crafted indicators that dominated quantitative finance for decades. These architectures learn hierarchical representations, progressively transforming market data into abstract features that capture complex dependencies invisible to traditional methods (Goodfellow et al., 2016).

3.1.1 Feedforward Networks: The Building Blocks

Feedforward neural networks extend classical linear models through multiple layers of non-linear transformations, enabling them to approximate arbitrarily complex functions (Goodfellow et al., 2016). In financial applications, this capability allows them to capture non-linear relationships between market variables—for instance, how the impact of momentum varies with volatility levels or how value factors interact with macroeconomic conditions.

The architecture processes information through sequential transformations:

$$h_l = \sigma(W_l h_{l-1} + b_l) \tag{1}$$

This equation describes how information flows through the network layers, functioning as a multi-step refinement process. Each layer takes the previous layer's output (h_{l-1}), applies learned weights (W_l) to emphasize important features, adds biases (b_l) to shift the baseline, then applies a non-linear activation function (σ) to capture complex patterns. Think of it as progressively refining raw market data into trading signals—early layers might detect simple patterns like price changes, while deeper layers combine these into complex indicators like momentum reversals under specific volatility conditions.

Financial practitioners have found that moderately deep networks with 2-4 hidden layers typically outperform both shallow and very deep architectures for daily prediction tasks (Sezer et al., 2020). Shallow networks lack the capacity to capture complex market dynamics, while excessively deep networks suffer from overfitting given finance's inherently noisy data. Fischer and Krauss (2018) demonstrated that networks with 5 layers and 1000 LSTM units achieved superior performance for S&P 500 constituent prediction, while Krauss et al. (2017)

found that 3-layer networks with aggressive regularization through 50% dropout effectively identified statistical arbitrage opportunities.

The critical challenge lies in finance's low signal-to-noise ratio. Unlike image recognition where patterns are relatively stable, financial patterns continuously evolve as market participants adapt. Networks must be sufficiently complex to capture genuine relationships yet regularized enough to avoid fitting noise—a balance that shifts with market conditions.

3.1.2 Recurrent Architectures: Capturing Temporal Dependencies

Financial markets exhibit strong temporal dependencies that feedforward networks cannot capture. While Recurrent Neural Networks (RNNs) address this through hidden states that evolve with observations, vanilla RNNs suffer from vanishing gradients when learning long-range dependencies.

Long Short-Term Memory networks (LSTMs) solve this through sophisticated gating mechanisms (Hochreiter and Schmidhuber, 1997). The architecture maintains a cell state C_t and hidden state h_t, controlled by three gates:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \qquad \text{(forget gate)} \tag{2}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \qquad \text{(input gate)} \tag{3}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \qquad \text{(output gate)} \tag{4}$$

These gates act as intelligent filters for information flow. The forget gate (f_t) decides what past information to discard—essentially asking "Should I forget that old momentum signal that's no longer relevant?" The input gate (i_t) controls what new information to store—"Is this price spike important enough to remember?" The output gate (o_t) determines what information to pass forward—"Which patterns are relevant for the current trading decision?" The sigmoid function (σ) outputs values between 0 and 1, acting like a dimmer switch that controls how much information passes through each gate.

The cell state updates through a carefully orchestrated combination of forgetting and

learning:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{5}$$

Here, the $\odot$ symbol denotes element-wise multiplication. This equation represents the LSTM's memory update mechanism—it selectively keeps relevant past information ($f_t \odot C_{t-1}$) while adding filtered new information ($i_t \odot$ new information). It's analogous to maintaining a trading journal that selectively remembers important patterns (like "tech stocks rally after Fed dovish statements") while forgetting outdated ones (like correlations that broke during regime changes).

This architecture proves particularly effective for financial time series. Fischer and Krauss (2018) demonstrated LSTMs achieving 0.46% daily returns and Sharpe ratios of 5.8 (pre-costs) for S&P 500 constituents, automatically focusing on high-volatility stocks with reversal profiles. Krauss et al. (2017) generated Sharpe ratios exceeding 4.0 through statistical arbitrage, while Sirignano and Cont (2019) uncovered universal price formation features from billions of high-frequency quotes with remarkably stable out-of-sample accuracy.

In asset pricing, Chen et al. (2024) used LSTMs to extract hidden economic states from macroeconomic series, significantly improving cross-sectional predictions. Bao et al. (2017) combined wavelets with stacked LSTMs to decompose price movements by frequency, while Althelaya et al. (2018) found 128-256 memory cells optimal for balancing capacity with generalization. Bidirectional variants, processing sequences in both directions, achieve 15-20% accuracy improvements for risk assessment where complete context matters (Siami-Namini et al., 2019).

Despite these successes, Gu et al. (2020) showed that while LSTMs outperform traditional methods, improvements diminish after transaction costs. The models identify patterns effectively but struggle with non-stationarity—profitable patterns attract competition, causing learned relationships to decay.

3.1.3 Attention Mechanisms: Dynamic Focus Across Time

Transformer architectures revolutionized sequence modeling by replacing recurrence with attention—directly modeling relationships between all positions simultaneously (Vaswani

et al., 2017). The scaled dot-product attention mechanism:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6}$$

This equation implements a sophisticated information retrieval system. The queries (Q) represent "what am I looking for?"—perhaps patterns similar to today's market conditions. The keys (K) represent "what information is available?"—historical market states. The values (V) contain the actual information to retrieve. The dot product QK^T measures similarity between queries and keys, identifying which historical periods are most relevant. The scaling factor $\sqrt{d_k}$ prevents the softmax from saturating, while the softmax function itself creates probability weights that sum to 1. It's like a trader scanning through years of historical charts to find periods that most closely resemble today's market, then weighting insights from those periods by their relevance.

For financial applications, this means a model analyzing current market conditions can simultaneously attend to similar episodes from last week, last month, or last year without the sequential processing constraints of RNNs. Multi-head attention runs multiple attention operations in parallel, capturing different types of relationships—one head might focus on volatility patterns while another tracks correlation structures.

Li et al. (2019) demonstrated transformers achieving 62% directional accuracy for Bitcoin prediction by attending to relevant historical patterns regardless of recency. During volatile periods, the model automatically focused on other high-volatility episodes; during trending markets, it attended to similar trending periods. This dynamic focus allocation surpasses fixed-window approaches that may include irrelevant information or exclude critical context.

3.1.4 Continuous-Time Modeling with Neural Ordinary Differential Equations (ODEs)

Neural Ordinary Differential Equations (Neural ODEs) represent a paradigm shift from discrete to continuous-time modeling (Chen et al., 2018). Rather than learning transformations

between fixed time steps, they parameterize the derivative of hidden states:

$$\frac{dh(t)}{dt} = f(h(t), t, \theta) \tag{7}$$

This differential equation describes the instantaneous rate of change in the model's hidden state. Instead of discrete jumps between time steps (like daily closing prices), it models continuous evolution—analogous to the difference between taking snapshots versus recording video. The function f learns how market states evolve continuously, with parameters θ determining the dynamics. This allows the model to naturally handle irregular data arrival: tick data comes sporadically, economic indicators release on varying schedules, and corporate events occur unpredictably. While traditional models must interpolate or discard information, Neural ODEs seamlessly incorporate observations whenever they arrive.

The continuous formulation also enables modeling of market microstructure dynamics where the evolution speed itself conveys information. Rapid price changes indicate information arrival or liquidity demands, while slow evolution suggests equilibrium. By modeling these dynamics continuously, Neural ODEs capture subtleties that discrete models miss.

3.2 Architectures for Non-Stationary Markets

Recognition that financial markets continuously evolve has driven development of specialized architectures explicitly designed for non-stationary environments. These approaches attempt to address distribution shifts, regime changes, and structural breaks—fundamental challenges that static models cannot overcome.

3.2.1 Patch-based Time Series Transformer (PatchTST)

PatchTST adapts the vision transformer's patch-based approach to time series, segmenting sequences into overlapping windows that capture patterns at multiple scales simultaneously (Nie et al., 2023). Each patch represents a local temporal pattern—perhaps a week of trading—while attention mechanisms capture relationships between patches, revealing how patterns evolve and interact across time.

This multi-scale approach excels when market dynamics exhibit consistent patterns at

specific frequencies. During trending markets, PatchTST successfully identifies momentum patterns that persist across daily, weekly, and monthly scales. The architecture learns that certain technical patterns (head-and-shoulders, support/resistance levels) maintain consistent predictive power across different timeframes, while fundamental relationships (earnings-to-price ratios) operate at longer horizons.

However, the architecture's assumption of local pattern stability becomes problematic during regime shifts. When the COVID-19 pandemic struck, relationships that had remained stable for years—correlations between sectors, the bond-equity relationship, the value-growth dynamic—changed fundamentally within days. PatchTST's patch-based representations, optimized for gradual evolution, cannot rapidly restructure to accommodate such dramatic shifts. The model continues searching for familiar patterns within patches even as the underlying data generation process has fundamentally changed.

3.2.2 Non-Stationary Transformers: Adapting to Distribution Shifts

Non-stationary transformers directly address changing statistical distributions through explicit modeling of series decomposition (Liu et al., 2022). The architecture separates stable patterns (relative relationships) from evolving distributions (absolute levels), allowing adaptation to statistical changes while preserving structural relationships.

Consider how this applies to volatility modeling: the architecture learns that technology stocks consistently exhibit 1.5-2x the volatility of utilities—a stable relative pattern. Simultaneously, it adapts to changing absolute volatility levels as markets transition between calm and turbulent periods. This separation allows the model to maintain predictive power even as market conditions change, adjusting its expectations for return magnitudes while preserving insights about relative movements.

The sophistication of this approach handles many forms of non-stationarity effectively. When central banks shift monetary policy, altering interest rate expectations and risk premiums, non-stationary transformers adapt their distribution parameters while maintaining learned relationships about how different assets respond to rate changes. During the 2022 inflation surge, these models successfully adjusted to higher volatility and altered return distributions.

Yet the architecture fails when non-stationarity extends beyond statistical moments to structural relationships. The 2022 interest rate hiking cycle didn't just change volatility levels—it fundamentally reversed the relationship between growth and value stocks. For over a decade, growth had outperformed in low-rate environments; suddenly, rising rates favored value. Non-stationary transformers, designed for statistical adaptation, lack mechanisms to handle such relationship inversions. The model assumes that relative patterns remain stable even as distributions shift, but markets violate this assumption during structural breaks.

3.2.3 Neural Hidden Markov Models: Regime Recognition

Neural Hidden Markov Models combine classical regime-switching frameworks with deep learning's pattern recognition capabilities. The architecture maintains beliefs about current market regimes while using neural networks to model regime-specific dynamics:

$$p(x_t|z_t) = \text{NeuralNetwork}_\theta(x_t, z_t) \tag{8}$$

In this formulation, z_t represents the latent (hidden) market regime—perhaps "risk-on," "risk-off," or "transition." The neural network learns different patterns for each regime, with parameters θ specialized for regime-specific dynamics. It's like having multiple trading strategies, each optimized for different market conditions, with an intelligent system that recognizes which conditions currently prevail.

This approach excels when markets cycle through identifiable regimes with distinct characteristics. The model might identify risk-on regimes where high-beta assets outperform, risk-off periods favoring defensive positions, and transition states with elevated uncertainty. Within each regime, deep networks capture complex patterns specific to that environment—momentum strategies might work in trending regimes but fail in choppy markets.

Bianchi and Iacopini (2020) demonstrated neural HMMs successfully navigating the 2020 market turbulence by recognizing regime transitions. The model identified the shift from normal conditions to crisis mode in late February, adjusted to the extreme volatility regime in March, and detected the recovery regime beginning in April. By maintaining separate

models for each regime, the architecture avoided contaminating stable-period patterns with crisis dynamics.

However, neural HMMs share a fundamental limitation with all regime-based approaches: they can only recognize regimes resembling those in training data. The rise of meme stocks, the emergence of cryptocurrency markets, and the impact of quantitative easing represent genuinely novel regimes without historical precedent. When encountering unprecedented conditions, neural HMMs either force-fit observations into known regimes—generating inappropriate predictions—or enter undefined states where behavior becomes unpredictable.

3.3 Adaptation Mechanisms for Market Evolution

Beyond architectural innovations, researchers have developed explicit adaptation mechanisms allowing models to evolve with changing markets. These approaches address concept drift—the phenomenon where patterns that predict returns change over time—through various strategies for continuous learning and dynamic adjustment.

3.3.1 Online Learning: Continuous Evolution

Online learning updates models incrementally with each observation rather than retraining from scratch (Bifet and Gavaldà, 2007). This continuous adaptation allows models to track gradual market evolution, adjusting parameters as correlations drift, volatilities change, and relationships evolve:

$$\theta_{t+1} = \theta_t - \eta_t \nabla L(x_t, y_t, \theta_t) \tag{9}$$

This update equation implements gradient descent in real-time. At each time step, the model computes the gradient of the loss function (∇L) with respect to current parameters (θ_t), then takes a small step in the direction that reduces error. The learning rate (η_t) controls adaptation speed—large values enable quick response to changes but risk overfitting to noise, while small values maintain stability but may miss fleeting opportunities. It's analogous to a trader continuously refining their strategy based on each day's results, making small adjustments rather than completely overhauling their approach.

The approach seems ideally suited for financial markets where conditions change continu-

ously. As inflation expectations rise, online learning gradually adjusts the weights assigned to inflation-sensitive features. When correlations between assets slowly change due to shifting economic conditions, the model smoothly adapts its correlation estimates. High-frequency trading firms employ online learning to update fair value estimates with each trade, maintaining accurate pricing models as market conditions evolve throughout the day.

Yet calibrating adaptation speed presents a fundamental challenge. Aggressive learning rates enable quick response to genuine changes but risk overfitting to noise—mistaking random fluctuations for meaningful patterns. Conservative rates maintain stability but adapt too slowly to capture fleeting opportunities. The optimal learning rate itself varies with market conditions: during stable periods, slow adaptation prevents noise-fitting; during transitions, rapid adaptation is essential. This creates a meta-learning problem—the model must learn how fast to learn—adding another layer of complexity.

The question of model complexity further complicates online adaptation. Kelly et al. (2024) theoretically prove that complex models with more parameters than observations can outperform simpler alternatives in return prediction, challenging conventional wisdom about overfitting. Their findings suggest online learning algorithms should allow model complexity to grow as they observe more data, embracing overparameterization rather than constraining it. However, Avramov et al. (2023) demonstrate that this complexity comes with severe practical limitations: machine learning strategies' profitability deteriorates by 52-77% when economic restrictions are imposed, such as excluding microcaps or high-volatility periods. The tension between theoretical benefits of complexity and practical constraints reveals a fundamental challenge—online learning must simultaneously manage growing model capacity while adapting to economic realities that render complex strategies unexploitable.

Financial markets' low signal-to-noise ratio exacerbates this challenge. In image recognition, patterns are relatively clear; in markets, noise often dominates signal. Online learning algorithms struggle to distinguish genuine regime changes requiring rapid adaptation from temporary volatility spikes that will quickly revert. The result is either chronic overfitting to noise or persistent underperformance during transitions.

3.3.2 Temporal Fusion Transformers: Dynamic Feature Selection

Temporal Fusion Transformers (TFT) address the reality that different information matters at different times through dynamic feature importance weighting (Lim et al., 2021). The architecture employs gating mechanisms that adjust attention to various inputs based on context, recognizing that market drivers shift with conditions.

During earnings season, fundamental metrics—revenue growth, earnings surprises, guidance revisions—dominate price movements. The model learns to focus on these features while downweighting technical indicators that provide little information during fundamentally-driven periods. During Federal Reserve meetings, macroeconomic indicators—inflation expectations, employment data, yield curves—become critical. The model shifts attention to these macro features while reducing emphasis on company-specific information.

This dynamic weighting extends to different time horizons. During calm markets, the model might emphasize longer-term moving averages and fundamental valuations. As volatility increases, attention shifts to short-term technical indicators and order flow dynamics. Option expiration weeks see increased focus on gamma positioning and pin risk, while month-end periods emphasize rebalancing flows.

The limitation emerges when entirely new information sources become relevant. TFT requires practitioners to predefine the universe of potentially important features. When novel factors suddenly matter—social media sentiment during the GameStop episode, supply chain data during COVID, energy prices during the Ukraine conflict—the model cannot incorporate these without architectural modification and complete retraining. The dynamic attention mechanism only works within the predetermined feature space, unable to recognize when factors outside this space become critical.

3.3.3 Model-Agnostic Meta-Learning: Learning to Adapt

Model-Agnostic Meta-Learning (MAML) takes adaptation a step further by learning how to learn—finding parameter initializations that enable rapid adaptation to new conditions (Finn et al., 2017). Recent applications to financial markets have explored both the promise and boundaries of this approach (Wang et al., 2023; Zhang et al., 2025):

$$\theta^* = \arg\min_{\theta} \sum_{\tau \sim p(\tau)} L_\tau(\theta - \alpha \nabla_\theta L_\tau(\theta)) \tag{10}$$

This meta-learning objective is sophisticated yet intuitive. Instead of optimizing for performance on current data, MAML optimizes for adaptability—finding parameters (θ^*) that, after a few gradient steps on new data, achieve good performance across many different tasks (τ). The inner gradient ($\nabla_\theta L_\tau$) represents adaptation to a specific market condition, while the outer optimization ensures this adaptation works well across diverse conditions. It's like training a trader not just to execute a specific strategy, but to quickly learn new strategies when market conditions change.

Rather than starting from random initialization when markets shift, MAML begins from carefully learned starting points that facilitate quick adaptation. During stable periods, the algorithm accumulates meta-knowledge about how markets typically transition—such as patterns in volatility spikes, sector rotations, or liquidity crises—enabling faster convergence when similar conditions reappear.

MAML demonstrates strong capabilities in learning temporal structure and managing memory across episodes. For instance, Malamud and Rotenberg (2021) showed that MAML-based strategies achieved superior risk-adjusted returns during market transitions by adapting within days rather than weeks. The KDD 2023 DoubleAdapt framework (Wang et al., 2023) further extends this by combining MAML with incremental learning for stock trend forecasting, improving responsiveness to evolving market conditions. Applications in portfolio selection have also leveraged MAML to maintain pools of candidate strategies and dynamically reallocate capital based on inferred regime characteristics (Zhang et al., 2025).

However, the approach exhibits a critical constraint: its effectiveness relies on *structural similarity* between future regimes and historical transitions seen during training. MAML's meta-parameters encode assumptions about how markets evolve, making it well-suited to navigating familiar dynamics. But in the face of *genuinely novel conditions*—such as negative interest rates, pandemic-induced shutdowns, or the rise of decentralized finance—these learned priors may become liabilities. The algorithm risks *force-fitting* unprecedented situations into its learned transition framework, leading to poor adaptation or even catastrophic

failure.

Thus, while MAML offers promising capabilities for rapid adaptation in familiar market regimes, its generalization ability is inherently limited when structural novelty exceeds what its meta-learning framework has previously encountered.

3.3.4 Mixture of Experts: Specialized Models for Different Conditions

Mixture of Experts (MoE) architectures maintain multiple specialized models, dynamically selecting them based on market conditions (Jacobs et al., 1991; Masoudnia and Ebrahimpour, 2014):

$$y = \sum_{i=1}^{n} g_i(x, \theta_g) \cdot f_i(x, \theta_i) \tag{11}$$

This equation describes an ensemble approach where expert models (f_i) specialize in different market environments, while a gating network (g_i) determines their activation weights. Each expert becomes a specialist—one might excel at trending markets, another at mean reversion, a third at volatility trading. The gating network learns to recognize current conditions and activates appropriate experts, with weights summing to 1. It's like having a team of specialist traders, each with unique expertise, managed by a head trader who decides who should take the lead based on market conditions.

This specialization allows each expert to master specific market conditions without contamination from conflicting patterns. A momentum expert learns patterns in trending markets without being confused by mean-reverting dynamics. A volatility expert focuses on turbulent periods without being diluted by calm-market patterns. The gating network learns to recognize conditions and activate appropriate experts—high directional movement triggers the momentum expert, while range-bound oscillation activates the mean-reversion specialist.

MoE architectures demonstrate particular strength in handling multi-resolution scales and non-periodic cycles. The ability to maintain separate experts for different temporal frequencies allows the model to capture both high-frequency microstructure patterns and low-frequency economic cycles without interference. During the 2020 recovery, MoE models successfully navigated by switching between crisis-mode experts (March), recovery experts (April-May), and momentum experts (June onwards).

However, MoE requires predefining the number and type of experts, limiting adaptation to anticipated scenarios. Markets regularly produce novel conditions requiring new expertise—the emergence of algorithmic trading required new microstructure experts, the rise of passive investing demanded flow-based experts, and cryptocurrency integration necessitated blockchain-aware experts. The architecture cannot spontaneously generate new experts for unprecedented situations, forcing inappropriate expert activation when conditions don't match any specialty.

3.4 Reinforcement Learning: Beyond Prediction to Decision-Making

While traditional machine learning focuses on predicting market variables using historical data, Reinforcement Learning (RL) goes further by learning how to act in the market through direct interaction to optimize long-term objectives such as risk-adjusted returns (Sutton and Barto, 2018). This distinction is critical in financial machine learning because accurate predictions alone rarely yield profitable trades—success depends on the timing and execution of decisions, especially under transaction costs, slippage, and dynamic feedback.

Crucially, RL allows agents to adapt to new or evolving market regimes not by relying solely on past patterns, but by learning through continuous trial-and-error interaction. Rather than assuming that future conditions will mirror the past, RL seeks to understand shifting environments as they unfold, capturing the sequential and strategic nature of trading in real time. At the same time, RL still requires stationarity thus suffering from the issues we raise in financial context.

However, deploying RL in finance is far from trivial. Markets are non-stationary, rewards are sparse and noisy, and exploration carries real financial risk. Unlike controlled simulations, financial markets don't permit safe retries or clear feedback, making RL difficult to train, evaluate, and deploy reliably in live settings.

3.4.1 Core RL Architectures for Trading

Deep Q-Networks (DQN) DQNs learn the value of taking specific actions in given market states (Mnih et al., 2015). The algorithm estimates expected returns from buying,

selling, or holding, learning through trial and error which actions lead to profitable outcomes:

$$Q(s, a) = r + \gamma \max_{a'} Q(s', a') \tag{12}$$

This Bellman equation captures the essence of sequential decision-making. $Q(s, a)$ estimates "how good is action a in state s?" The immediate reward r represents the profit or loss from the action. The discount factor γ (typically 0.9-0.99) reduces the weight of future rewards, reflecting time preference and uncertainty. The term $\max_{a'} Q(s', a')$ represents the best possible future value from the resulting state s'. The model learns through experience: "If I buy here, I get immediate profit plus the best outcome I can achieve from the resulting position." It's learning not just what to trade, but when to enter, how long to hold, and when to exit.

DQNs excel at sequential decision-making problems where actions have long-term consequences. Deng et al. (2017) demonstrated DQNs achieving Sharpe ratios of 2.5 in cryptocurrency trading by optimizing for profit rather than prediction accuracy—the model learned complex execution strategies that traditional approaches missed. The algorithm discovered that splitting large orders across time periods and venues minimized market impact, a strategy human traders employ but traditional optimization struggles to capture.

However, RL faces severe data efficiency challenges in finance. While game-playing agents can generate millions of training episodes, financial markets offer limited historical data—perhaps 250 trading days per year. The algorithm must learn effective policies from sparse, noisy rewards where profits might not materialize for weeks or months. This sample inefficiency, combined with market non-stationarity, causes policies to break when regimes shift. A DQN trained during low-volatility periods often takes excessive risks when volatility returns, having never experienced the consequences of leverage during turbulent markets.

Proximal Policy Optimization (PPO) PPO addresses stability through constrained updates, preventing dramatic strategy changes that could cause catastrophic losses (Schul-

man et al., 2017):

$$L^{CLIP}(\theta) = E_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \tag{13}$$

This clipped objective function implements a crucial safety mechanism. The ratio $r_t(\theta) = \pi_\theta(a_t|s_t)/\pi_{old}(a_t|s_t)$ measures how much the new policy differs from the old. The advantage estimate $\hat{A}_t$ indicates whether an action was better or worse than expected. Clipping to $[1-\epsilon, 1+\epsilon]$ (typically $\epsilon = 0.2$) ensures policy updates stay within 20% of the previous policy. It's like a risk manager constraining a trader: "You can adjust your strategy based on new information, but not so dramatically that a single bad decision ruins everything."

The clipping mechanism ensures conservative evolution while maintaining exploration for strategy improvement. This proves crucial in financial markets where a single bad decision can eliminate months of accumulated profits. Zhang et al. (2020) showed PPO achieving consistent profitability across equity and commodity markets through balanced exploration-exploitation tradeoffs. The algorithm learned to reduce position sizes during uncertain periods while aggressively capitalizing on high-confidence opportunities.

PPO particularly excels in portfolio allocation problems where gradual adjustment prevents whipsaw losses. The algorithm learns not just what positions to hold but how quickly to transition between allocations, minimizing transaction costs while capturing opportunities. During the 2020 market turbulence, PPO-based strategies avoided the dramatic reversals that plagued more aggressive approaches by maintaining moderate position sizes and gradual rebalancing.

Actor-Critic Methods Actor-critic architectures combine policy learning (actor) with value estimation (critic), enabling more stable and efficient learning:

$$\nabla_\theta J(\theta) = E[\nabla_\theta \log \pi_\theta(a|s) A^\pi(s,a)] \tag{14}$$

This policy gradient formulation shows how the actor improves. The gradient pushes the policy π_θ toward actions with positive advantages $A^\pi(s,a)$ and away from those with negative advantages. The critic estimates these advantages, providing a baseline that reduces

variance in the gradient estimates. It's like having two systems: a trader (actor) who decides what to do, and an analyst (critic) who evaluates whether those decisions are better or worse than expected, helping the trader improve.

Soft Actor-Critic (SAC) adds entropy regularization for improved exploration (Haarnoja et al., 2018). This entropy term encourages the policy to maintain some randomness, preventing premature convergence to suboptimal strategies. Liang et al. (2018) demonstrated these methods achieving superior risk-adjusted returns through balanced exploration and exploitation. The entropy regularization proved particularly valuable in preventing overfitting to specific market conditions, maintaining strategy robustness across varying regimes.

3.4.2 Advanced RL Frameworks for Financial Markets

Table 1 synthesizes the key characteristics of advanced RL frameworks designed for financial markets, revealing a fundamental trade-off: approaches that excel in adaptation consistently struggle with practical deployment constraints. As the comparison demonstrates, each category addresses specific market challenges while introducing new vulnerabilities—a pattern that emerges repeatedly across our analysis. We now examine these frameworks in detail, beginning with meta-reinforcement learning's attempt to accelerate adaptation through learned priors.

Meta-Reinforcement Learning Meta-RL addresses adaptation speed by learning to learn—acquiring meta-knowledge that enables rapid policy adjustment to new market conditions. RL^2 maintains memory across episodes through recurrent networks, recognizing patterns in regime transitions (Duan et al., 2016):

$$\pi_\theta(a_t|h_t) = f_\theta(h_t), \quad h_t = \text{RNN}(h_{t-1}, s_{t-1}, a_{t-1}, r_{t-1}) \tag{15}$$

This architecture maintains a hidden state h_t that accumulates experience across episodes. The RNN processes not just the current state but also previous states, actions, and rewards, building a memory of what works in different conditions. The policy then conditions on this accumulated knowledge. It's like a trader who remembers not just current market conditions

Table 1: Reinforcement Learning as Decision-Making Layer: Architecture Comparison

RL Category	Key Strengths	Critical Weaknesses	Optimal Use Cases
Core RL (DQN, PPO, SAC)	Sequential decision optimization; direct profit maximization; execution strategy learning	Data inefficiency requiring years of history; catastrophic forgetting during regime shifts	High-frequency trading; market making; execution optimization
Meta RL (RL^2, MAML+RL)	Rapid adaptation to familiar regimes; transfer learning across similar markets	Fails on unprecedented events; requires structural similarity	Multi-asset strategies; regime-based allocation
Offline RL (CQL, BCQ)	No exploration risk; learns from historical trades	Cannot discover novel strategies; distribution bias	Regulatory constrained trading; risk-averse institutions
Hierarchical RL (Options, HRL)	Separates portfolio and execution decisions; temporal abstraction	Complex credit assignment; cascading errors between levels	Institutional trading desks; multi-scale strategies
Safe RL (Constrained PPO)	Risk-aware exploration; drawdown limits	Conservative; may miss opportunities	Pension funds; insurance portfolios

but also how similar situations played out previously, using this meta-knowledge to adapt quickly when patterns repeat.

When markets shift, the meta-policy quickly identifies similar historical transitions and adapts accordingly—not relearning from scratch but leveraging accumulated experience. Wang et al. (2016) demonstrated MAML-based RL achieving 30% higher Sharpe ratios during market transitions. The algorithm accumulated meta-knowledge during stable periods about how trading dynamics typically change, then rapidly adjusted when conditions shifted. During the March 2020 volatility spike, meta-RL agents adapted within days while conventional RL required weeks to adjust.

Yet meta-RL shares the fundamental limitation of all historical-based approaches: it cannot handle genuinely novel regimes. The algorithm assumes new conditions will resemble past

experiences in some learnable way. When markets enter unprecedented territory—negative rates destroying carry trades, social media coordination creating new dynamics—meta-RL lacks the framework for appropriate adaptation.

Offline Reinforcement Learning Offline RL learns exclusively from historical data, eliminating exploration risk by never executing untested actions in live markets. Conservative Q-Learning (CQL) prevents overestimation of out-of-distribution actions (Kumar et al., 2020):

$$L(\theta) = \alpha E_{s\sim D}[\log \sum_{a} \exp(Q_\theta(s,a))] + L_{\text{SAC}}(\theta) \tag{16}$$

The first term in this loss function is a conservative penalty that prevents the Q-function from assigning high values to actions not seen in the dataset D. The log-sum-exp creates a soft maximum over all actions, and minimizing this term pushes down Q-values for out-of-distribution actions. This conservatism prevents the dangerous extrapolation that could occur if the model overestimated the value of untested strategies. It's implementing the principle "don't be overconfident about strategies you've never seen work."

This approach proves valuable for institutions with vast historical trade databases but strict risk limits preventing live experimentation. CQL's conservative value estimates prevent overconfident extrapolation beyond observed actions. Liu et al. (2020) showed Batch-Constrained Q-learning BCQ achieving stable performance in equity markets by avoiding dangerous extrapolation beyond historical experience.

However, offline RL cannot discover strategies beyond those present in historical data. If past traders never employed certain techniques, the algorithm cannot learn them. This creates a fundamental limitation: offline RL can only be as good as the historical strategies it learns from, unable to innovate beyond human traders' past decisions.

Hierarchical Reinforcement Learning Hierarchical RL decomposes complex trading into layered decisions, mirroring institutional structures where portfolio managers set strategic allocation while execution traders implement positions. The options framework creates

temporal abstractions (Sutton et al., 1999):

$$\pi_o(a|s) = \text{low-level policy}, \quad \pi_\Omega(o|s) = \text{high-level policy} \tag{17}$$

Here, options o represent high-level strategies (like "accumulate tech stocks" or "reduce risk"), while the low-level policy π_o determines specific actions to implement these strategies. The high-level policy π_Ω decides which option to pursue based on market state. This separation mirrors real trading desks: portfolio managers decide strategic allocation (high-level) while execution traders determine optimal order placement (low-level).

This separation allows specialization—high-level policies focus on asset allocation without worrying about execution details, while low-level policies optimize order placement without concerning themselves with portfolio construction. The approach naturally handles the multi-scale nature of financial markets where decisions span microseconds to months.

Yet hierarchical RL suffers from complex credit assignment problems. When strategies lose money, determining whether the fault lies with poor allocation decisions or bad execution becomes challenging. The levels must be trained jointly but operate on different timescales with different objectives, creating optimization difficulties that remain partially unsolved.

Safe Reinforcement Learning Safe RL addresses the critical need for risk-aware exploration in financial markets where unconstrained learning can lead to catastrophic losses (García and Fernández, 2015). These approaches incorporate explicit safety constraints into the learning process, ensuring that the agent never takes actions that could violate predefined risk limits. Constrained Policy Optimization (CPO) extends PPO by adding constraint functions that must be satisfied at each update:

$$\max_\theta E_{\pi_\theta}[R] \quad \text{subject to} \quad E_{\pi_\theta}[C_i] \leq d_i \quad \forall i \tag{18}$$

Here, R represents the reward (returns), while C_i are cost functions measuring constraint violations (such as drawdown exceeding limits or position concentration violating diversification rules), with thresholds d_i. The optimization seeks maximum returns while ensuring risk metrics stay within acceptable bounds. It's implementing institutional risk management

directly into the learning algorithm—the model cannot learn strategies that would trigger risk limits, even if they appear profitable in backtests.

Safe RL proves particularly valuable for institutional investors with strict mandates. Pension funds cannot tolerate drawdowns exceeding 10-15%, regardless of potential upside. Insurance portfolios must maintain specific asset allocations for regulatory compliance. Chow et al. (2018) demonstrated that CPO-based trading strategies maintained drawdowns within prescribed limits while achieving 85% of unconstrained returns. The approach learns to recognize when market conditions increase the probability of limit violations, preemptively reducing exposure rather than waiting for losses to materialize.

However, safety constraints inherently limit performance. By avoiding regions of the action space that could violate constraints, safe RL may miss profitable opportunities that appear risky but would actually succeed. The conservative nature means these algorithms typically underperform during strong bull markets where risk-taking is rewarded. Furthermore, defining appropriate safety constraints requires deep domain knowledge—overly restrictive limits prevent learning, while loose constraints provide insufficient protection. The challenge intensifies during regime changes when historical risk estimates become unreliable, potentially causing either excessive conservatism or inadequate protection.

3.4.3 Practical Deployment Considerations

The fundamental distinction between RL and forecasting models creates unique deployment challenges that vary significantly across different RL architectures. As summarized in Table 1, the optimal use cases range from high-frequency trading for core RL methods to regulatory-constrained environments for offline approaches, each with distinct implementation requirements:

- **Integration Architecture**: Successful deployment typically combines forecasting and RL—LSTMs predict market states while RL optimizes actions given predictions. This separation leverages each paradigm's strengths while mitigating weaknesses.

- **Reward Engineering**: Unlike supervised learning's clear loss functions, RL requires careful reward design. Sharpe ratio optimization differs fundamentally from profit max-

imization, leading to different learned behaviors. Risk-adjusted metrics may produce conservative strategies that underperform in bull markets.

- **Simulation Gap**: The reality gap between backtesting and live trading amplifies in RL. Market impact, slippage, and adversarial behavior aren't captured in historical simulations, causing strategy degradation. Paper trading provides intermediate validation but still lacks real market dynamics.

- **Sample Efficiency Crisis**: Financial markets provide limited data relative to RL's needs. A strategy requiring 10,000 episodes to converge would need decades of daily data—far exceeding available history for most instruments. This necessitates careful algorithm selection favoring sample-efficient methods.

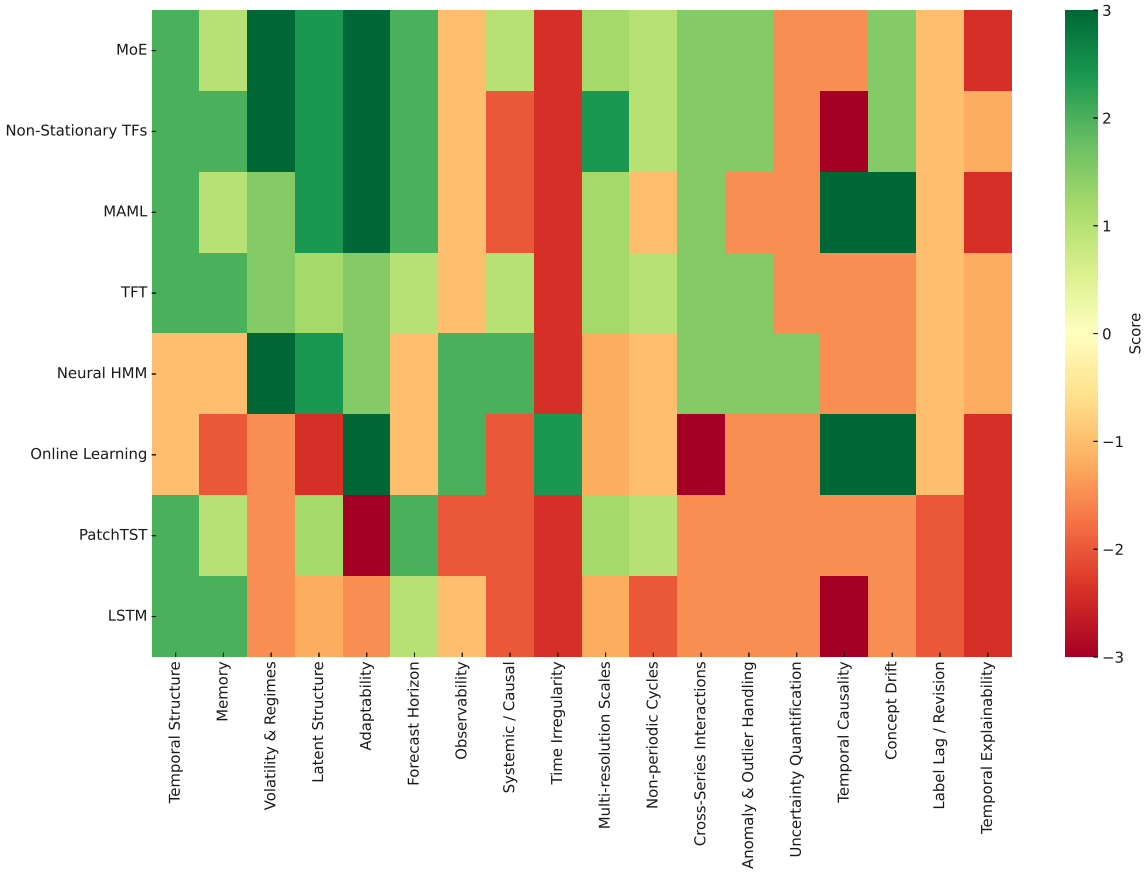


Figure 1: Scores across different evaluation criteria for adaptive and non-stationary ML algorithms. Higher scores indicate better suitability for specific financial market conditions.

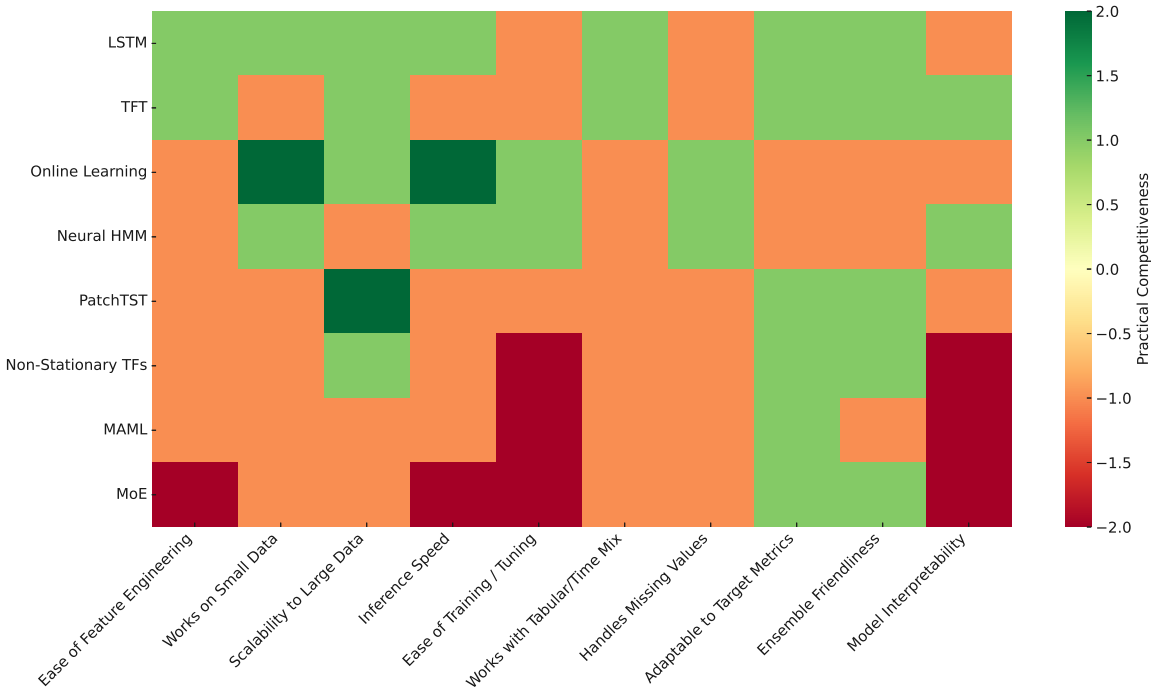


Figure 2: Practical competitiveness heatmap of modern ML algorithms across various financial market characteristics. Darker green indicates stronger performance, while darker red indicates weaker performance.

3.5 Fundamental Limitations Revealed

In this section, we evaluate the different ML methodologies designed to address non-stationarity. For that purpose, we score each algorithm along several dimensions desired for the algorithm to perform well. Our scoring is not empirical, but rather reflects our informed judgment based on the architectural descriptions of the machine learning algorithms provided in Section 3. We assess each algorithm by considering the strengths it was explicitly designed to exploit, as well as the weaknesses that arise from its underlying assumptions and training requirements. We present the scoring results in Figures 1 and 2. Below we summarize the reasoning behind the assigned scores:

- **LSTM**: Designed with gating mechanisms to capture temporal structure and long-term dependencies, which explains their strong scores on memory. However, they lack mechanisms for regime-switching, causal reasoning, or uncertainty quantification, leading to lower abstract scores. Our judgment reflects their consistent practical com-

petitiveness due to robustness on small datasets, ease of tuning, and synergy with feature engineering.

- **PatchTST**: Transformer patches are effective at extracting local temporal patterns and long-horizon dependencies, which motivates high scores on structure and horizon. Yet they are static once trained, lack adaptability to drift, and do not handle irregular sampling. Our judgment therefore penalizes them on adaptability and practical competitiveness, consistent with their limited use in small-data or competition settings.

- **Non-Stationary Transformers**: By explicitly decomposing time series into trend, seasonality, and frequency components, these models achieve strong abstract scores for adaptability and latent structure. However, our judgment is that they remain weak on uncertainty quantification, causality, and practical efficiency, since they are data-hungry and complex to train.

- **TFT (Temporal Fusion Transformer)**: Combines recurrence, attention, and gating, which justifies moderate-to-high abstract scores for dynamic feature integration and medium-horizon forecasting. At the same time, our judgment reflects their reliance on predefined features and weaker adaptability to unseen regimes, explaining the middling competitiveness score.

- **Neural HMM**: Excels in explicit regime modeling through latent states, which earns strong abstract scores in volatility and observability. However, our judgment reflects its limitation to historical regimes and difficulty handling unprecedented dynamics, as well as its rigidity in practical settings.

- **Online Learning**: By design, these methods adapt continuously to data drift, hence strong adaptability scores. But they provide little latent structure discovery or long-horizon modeling, which we penalize in the abstract scoring. Our judgment also reflects their practical drawback: sensitivity to noise, which often makes them less favored in competitions.

- **MAML (Meta-Learning)**: Abstractly powerful, since it rapidly adapts to related tasks, which motivates strong adaptability and drift-handling scores. Our judgment pe-

nalizes it when tasks are unrelated, where performance drops significantly. Practically, the training complexity and lack of interpretability explain its weak competitiveness.

- **Mixture of Experts (MoE)**: Architecturally versatile, as experts can specialize across regimes, volatility patterns, or latent substructures. This motivates strong abstract scores across adaptability, regime awareness, and latent modeling. Our judgment penalizes MoE in practice, where training instability, expert collapse, and computational demands make it less competitive despite its theoretical promise.

The comparison between Figures 1 and 2 highlights a critical insight: while different architectures excel in specific dimensions, none successfully navigate the full spectrum of market evolution. Models that achieve high abstract adaptability scores, such as Non-Stationary Transformers or MAML, consistently struggle with implementation practicality, while models that dominate in practice, such as Long Short-Term Memory (LSTMs), lack deeper structural versatility. This trade-off becomes particularly acute as markets undergo structural change, where theoretically powerful models fail to deliver robustness, and practically competitive models fail to generalize.

The challenge becomes even more pronounced when considering reinforcement learning (RL). Although RL frameworks offer theoretical advantages in decision-making under uncertainty, practical deployment is constrained by integration architecture, reward specification, the simulation gap, and the sample efficiency crisis. The gap between backtesting and live trading often amplifies errors, while the data requirements for stable RL exceed what financial markets can realistically provide. Thus, even algorithms that appear superior in controlled environments falter in real-world trading.

Beyond algorithmic design, fundamental properties of financial markets further limit model performance. The proliferation of potential predictive signals—the so-called "factor zoo" (Feng et al., 2020)—creates a combinatorial model selection problem prone to overfitting. Shrinkage approaches, such as those proposed by Kozak et al. (2020), demonstrate that principal components can summarize this high-dimensional space, but the relevant components themselves evolve over time, creating a moving target. Reflexivity compounds this challenge, as successful predictions alter the market environment, generating recursive

feedback loops that may be theoretically intractable.

Temporal heterogeneity further complicates modeling: microsecond-level high-frequency dynamics coexist with decade-long macroeconomic cycles, and no single architecture can perform well across all scales simultaneously. As Figure 2 illustrates, models strong at high-frequency forecasting tend to underperform on structural, long-term horizons, and vice versa. Adversarial dynamics ensure that predictable patterns are exploited quickly, forcing models to balance structure with randomness to avoid being gamed by competing strategies. Non-ergodicity adds another layer of difficulty, as sequence-dependent outcomes mean that even strategies with positive expected returns may lead to ruin under unfavorable paths.

Finally, financial markets regularly present genuine novelty—cryptocurrencies, pandemic shocks, and social-media-driven trading are recent examples—that cannot be anticipated from historical data. Even models designed for adaptability, such as MAML, falter in the face of unprecedented structures, as reflected in their weak competitiveness scores.

Taken together, the progression from simple recurrent networks to sophisticated adaptive architectures demonstrates increasing capability to address certain aspects of market evolution. Yet each advance also exposes new vulnerabilities. The fundamental challenge is that markets do not merely change exogenously; they evolve reflexively in response to the very models and strategies applied to them. This reflexive evolution, combined with practical deployment constraints, underscores why no single algorithm consistently prevails across the dimensions considered.

4 Empirical Considerations and Performance Benchmarks

4.1 Alpha Decay and Strategy Lifecycle

Empirical evidence suggests strategy alpha follows predictable decay patterns:

$$\alpha(t) = \alpha_0 \cdot e^{-\lambda t} + \epsilon \tag{19}$$

where α_0 represents initial alpha, λ is the decay rate, and ϵ represents noise. McLean and Pontiff (2016) documented 58% decay following publication, with additional decay as strategies become widely adopted.

This decay reflects multiple mechanisms. Direct competition reduces profit margins as more capital pursues the same opportunities. Market adaptation eliminates inefficiencies as participants adjust behaviour. Structural changes alter the underlying relationships that strategies exploit. The decay rate λ varies by strategy type—simple patterns decay quickly whilst complex strategies requiring sophisticated implementation decay more slowly.

4.2 Performance Across Market Regimes

Different architectures excel in specific regimes, creating no universal champion. PatchTST performs well during trending markets with stable patterns at consistent timescales. Non-stationary transformers adapt to volatility regime changes but struggle with relationship inversions. Neural HMMs detect recurring regime patterns but fail when confronted with unprecedented conditions. Reinforcement learning agents learn sophisticated policies but require stable reward structures.

This regime-dependence creates fundamental challenges for performance evaluation. A model appearing successful during one market phase may fail catastrophically when conditions change. Backtesting on historical data cannot capture future regime shifts, making out-of-sample performance highly uncertain. The COVID-19 pandemic exemplified these regime shift consequences: models trained on 2015-2019 data failed catastrophically when applied to pandemic market dynamics.

4.3 The Buy-and-Hold Benchmark

The S&P 500 buy-and-hold strategy presents a formidable benchmark. Despite simplicity, it has proven remarkably difficult to beat consistently after accounting for transaction costs, market impact, and implementation frictions. Following Gu et al. (2020), sophisticated models achieve Sharpe ratios of 0.5-1.5 in backtests but often underperform simple strategies after costs. In the context of the Adaptive Markets Hypothesis (AMH), however, Sharpe

ratios become a moving target rather than a stable yardstick: as profitable strategies attract competition, risk–return profiles adapt endogenously, eroding the persistence of any observed ratio. This makes historical Sharpe ratios less meaningful indicators of future performance.

Buy-and-hold benefits from several advantages. Long-term economic growth provides natural tailwind. Zero timing risk eliminates regime shift vulnerability. Minimal transaction costs maximise net returns. Complete transparency prevents adversarial exploitation. Infinite capacity allows unlimited capital deployment. Active strategies must overcome these advantages through superior returns that compensate for higher costs and risks. The evidence suggests this hurdle remains largely insurmountable in developed markets such as S&P 500 stocks.

4.4 Drawdown Analysis Under Regime Shifts

Maximum drawdowns during regime transitions often exceed backtest estimates by 2-3x due to multiple factors. Correlation breakdowns eliminate diversification benefits precisely when most needed. Liquidity evaporation increases execution costs as market makers widen spreads. Model confusion generates false signals as learned patterns become invalid. Crowding effects amplify losses through forced unwinding as multiple strategies fail simultaneously.

The COVID-19 market crash demonstrated these dynamics. Strategies diversified across factors, sectors, and geographies all declined together as correlations approached one. Models trained on normal market conditions generated extreme position sizes just before the crash, amplifying losses. Risk management systems calibrated on historical volatility underestimated potential drawdowns by orders of magnitude.

5 Conclusion

This paper has examined machine learning performance in financial markets through the Adaptive Market Hypothesis lens. Our analysis reveals that the persistent failure of quantitative trading strategies stems not from technical inadequacies but from the fundamental nature of markets as adaptive systems. The historical progression from technical analysis through algorithmic trading to machine learning demonstrates a consistent pattern: each

generation promises to overcome its predecessors' limitations, yet all succumb to the same evolutionary forces.

The Adaptive Market Hypothesis provides a coherent framework for understanding these failures. Markets function as evolutionary systems where strategies compete for finite alpha. Success attracts competition, leading to crowding and eventual profit erosion. Environmental changes continuously reshape the fitness landscape. In this context, the question is not whether a strategy will fail, but when and how quickly. Machine learning's sophisticated pattern recognition accelerates discovery but cannot prevent the subsequent decay through competitive exploitation.

Given this reality, we must establish clear requirements both for algorithms to survive in adaptive markets and for empirical frameworks to properly evaluate their success or failure.

5.1 Requirements for Survivability in Adaptive Markets

Strategy survival in adaptive markets demands capabilities far beyond static optimisation. The COVID-19 pandemic and other regime shifts have revealed what distinguishes surviving strategies from those that fail catastrophically. Based on our analysis, we identify five essential requirements for algorithmic trading systems operating in adaptive markets:

Continuous Learning: Algorithms must implement incremental adaptation rather than periodic retraining. Online learning approaches (Bifet and Gavaldà, 2007) must balance stability against plasticity—adapting quickly enough to capture new patterns but slowly enough to avoid overfitting noise. The optimal learning rate itself must vary with market conditions, increasing during regime transitions and decreasing during stable periods.

Regime Recognition: Systems must detect structural changes before significant losses accumulate. Change-point detection and anomaly identification remain imperfect, particularly in distinguishing temporary fluctuations from permanent shifts. Early detection often produces false positives; waiting for confirmation means accepting losses. The challenge intensifies when encountering historically unprecedented regimes.

Dynamic Strategy Selection: Successful systems maintain portfolios of diverse strategies with adaptive allocation. Meta-models must learn regime-strategy mappings, though these mappings themselves evolve. What worked in past volatility spikes may fail in future

ones as market structure changes. The system must continuously evaluate strategy fitness and reallocate capital accordingly.

Meta-Learning Capability: Algorithms must understand pattern persistence versus decay (Malamud and Rotenberg, 2021). Distinguishing robust relationships from temporary anomalies remains challenging. Models must learn not just patterns but pattern lifecycles—when strategies emerge, peak, and decay. This requires maintaining beliefs about the reliability and expected lifespan of discovered patterns.

Adversarial Robustness: Systems must recognise competitive dynamics where other participants actively exploit predictable behaviour. Maintaining unpredictability whilst capturing alpha requires sophisticated game-theoretic reasoning. Too much randomness eliminates profits; too little invites exploitation. The adversarial nature of markets means that any successful pattern becomes its own worst enemy once discovered.

These requirements suggest that future developments may include evolutionary architectures maintaining strategy populations, multi-agent frameworks modelling strategic interaction, and causal inference enabling better generalisation. However, fundamental limits remain: reflexivity ensures that successful adaptation changes the environment, creating infinite regress where models must predict their own impact on markets.

5.2 Empirical Validation Requirements for Benchmarking

Theory must be complemented by rigorous empirical validation to assess whether algorithms meet the survivability requirements above. The critical question remains: can sophisticated adaptive architectures outperform passive benchmarks after realistic implementation costs? We propose comprehensive evaluation criteria to properly benchmark algorithmic trading systems and determine their success or failure:

Comparative Performance Testing: Adaptive architectures (PatchTST, non-stationary transformers, neural HMMs) must be tested against classical baselines and buy-and-hold using risk-adjusted metrics including maximum drawdowns, and recovery periods. Performance evaluation must span multiple market cycles to capture regime-dependent behaviour. The S&P 500 index provides an ideal benchmark—success here would demonstrate genuine robustness, while failure would question adaptive machine learning's practical value.

Regime-Specific Analysis: Performance must be evaluated across varying market conditions—trending versus ranging, low versus high volatility (Hansen and Lunde, 2005), crisis versus expansion. This reveals adaptation effectiveness and failure modes. Models excelling in one regime often fail in others, making comprehensive testing across all market conditions essential for declaring success.

Adaptation Dynamics Measurement: Benchmarking must measure response latency to regime changes and determine whether online methods detect shifts sufficiently quickly to preserve capital. The trade-off between false positive rates and detection speed remains critical—early warnings often prove incorrect whilst waiting for confirmation means accepting initial losses. Success requires demonstrating consistent early detection without excessive false alarms.

Implementation Reality Testing: Evaluation must incorporate transaction costs, market impact, and execution delays. Many theoretically profitable strategies fail once realistic frictions are considered. Slippage, bid-ask spreads, and market impact can consume entire alpha, particularly for strategies requiring frequent rebalancing. Only strategies maintaining profitability after all implementation costs can be considered successful.

Out-of-Sample Robustness Validation: Testing on genuinely unseen data, particularly unprecedented market conditions like the COVID-19 period, provides the ultimate validation. Historical backtests cannot capture black swan events or structural breaks, making forward testing essential. True success requires consistent performance across both seen and unseen market conditions.

5.3 Final Perspective

The evidence suggests that achieving consistent outperformance in liquid, developed markets remains extraordinarily difficult, even with sophisticated machine learning. The adaptive nature of markets creates a fundamental barrier: any pattern strong enough to generate profits attracts competition that erodes those profits. Machine learning accelerates this cycle rather than escaping it.

This does not imply that machine learning has no role in finance. These techniques may prove valuable in less efficient markets, for risk management rather than alpha generation,

or in combination with structural advantages like superior data or execution. However, the dream of discovering a machine learning model that consistently beats the market appears increasingly illusory.

The Adaptive Market Hypothesis suggests a more realistic perspective: markets are not puzzles to be solved but ecosystems to navigate. Success requires not finding the perfect model but continuously adapting to changing conditions while managing the inevitable decay of any edge. Future research should focus on developing frameworks that explicitly account for market adaptation, creating robust strategies that degrade gracefully rather than failing catastrophically, and identifying market segments where structural barriers slow competitive erosion. Only by acknowledging the fundamental challenges posed by adaptive markets and establishing rigorous benchmarking criteria can we develop realistic applications of machine learning in finance.

References

Allen, F. and Karjalainen, R. (1999). Using genetic algorithms to find technical trading rules. *Journal of Financial Economics*, 51(2):245–271.

Althelaya, K. A., El-Alfy, E.-S. M., and Mohammed, S. (2018). Evaluation of bidirectional lstm for short-and long-term stock market prediction. In *International Conference on Information and Communication Systems*, pages 151–156.

Asness, C. S., Moskowitz, T. J., and Pedersen, L. H. (2013). Value and momentum everywhere. *Journal of Finance*, 68(3):929–985.

Augustin, P., Cheng, I.-H., and den Bergen, M. V. (2021). Volmageddon and the failure of short volatility products. *Financial Analysts Journal*, 77(3):23–45.

Avramov, D., Cheng, S., and Metzker, L. (2023). Machine learning vs. economic restrictions: Evidence from stock return predictability. *Management Science*, 69(5):2587–2619.

Bailey, D. H., Borwein, J. M., de Prado, M. L., and Zhu, Q. J. (2014). The probability of backtest overfitting. *Journal of Computational Finance*, 20(4):39–69.

Baltas, N. (2017). Optimising cross-asset carry trading strategies. *Journal of Trading*, 12(2):10–30.

Bank for International Settlements (2018). The equity market turbulence of 5 february – the role of exchange-traded volatility products. *BIS Quarterly Review*, pages 1–12.

Bao, W., Yue, J., and Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLOS One*, 12(7):e0180944.

Bianchi, D. and Iacopini, M. (2020). Modeling and predicting market risk with conditional autoregressive value at risk. *Journal of Financial Econometrics*, 18(2):458–486.

Bifet, A. and Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 443–448.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Brock, W., Lakonishok, J., and LeBaron, B. (1992). Simple technical trading rules and the stochastic properties of stock returns. *Journal of Finance*, 47(5):1731–1764.

Carhart, M. M. (1997). On persistence in mutual fund performance. *Journal of Finance*, 52(1):57–82.

Chen, L., Pelger, M., and Zhu, J. (2024). Deep learning in asset pricing. *Management Science*, 70(2):714–750.

Chen, R. T. Q., Rubanova, Y., Bettencourt, J., and Duvenaud, D. (2018). Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, pages 6571–6583.

Chow, Y., Ghavamzadeh, M., Janson, L., and Pavone, M. (2018). Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 18(167):1–51.

Daniel, K. and Moskowitz, T. J. (2016). Momentum crashes. *Journal of Financial Economics*, 122(2):221–247.

Deng, Y., Bao, F., Kong, Y., Ren, Z., and Dai, Q. (2017). Deep direct reinforcement learning for financial signal representation and trading. *IEEE Transactions on Neural Networks and Learning Systems*, 28(3):653–664.

Di Persio, L., Garbelli, M., and Wallbaum, K. (2021). Forward-looking volatility estimation for risk-managed investment strategies during the covid-19 crisis. *Risks*, 9(2):33.

Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., and Abbeel, P. (2016). Rl^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*.

Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. *Journal of Finance*, 25(2):383–417.

Fama, E. F. (1991). Efficient capital markets: II. *Journal of Finance*, 46(5):1575–1617.

Fama, E. F. and French, K. R. (1993). Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics*, 33(1):3–56.

Fama, E. F. and French, K. R. (2015). A five-factor asset pricing model. *Journal of Financial Economics*, 116(1):1–22.

Feng, G., Giglio, S., and Xiu, D. (2020). Taming the factor zoo: A test of new factors. *The Journal of Finance*, 75(3):1327–1370.

Filippou, I., Rapach, D., and Thimsen, C. (2021). Boosting cryptocurrency return prediction. *SSRN Electronic Journal*. Available at SSRN: `https://www.ssrn.com/abstract=3901734`.

Finn, C., Abbeel, P., and Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1126–1135.

Fischer, T. and Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232.

García, J. and Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press.

Goyal, A. and Welch, I. (2008). A comprehensive look at the empirical performance of equity premium prediction. *The Review of Financial Studies*, 21(4):1455–1508.

Grossman, S. J. and Stiglitz, J. E. (1980). On the impossibility of informationally efficient markets. *American Economic Review*, 70(3):393–408.

Gu, S., Kelly, B., and Xiu, D. (2020). Empirical asset pricing via machine learning. *Review of Financial Studies*, 33(5):2223–2273.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 1861–1870.

Hansen, P. R. and Lunde, A. (2005). A forecast comparison of volatility models: Does anything beat a garch(1,1)? *Journal of Applied Econometrics*, 20(7):873–889.

Harvey, C. R. and Liu, Y. (2017). Backtesting. *Journal of Portfolio Management*, 42(1):13–28.

Harvey, C. R. and Liu, Y. (2019). A census of the factor zoo. Technical Report 3341728, SSRN. SSRN Working Paper, last revised October 16, 2020.

Harvey, C. R., Liu, Y., and Zhu, H. (2016). ... and the cross-section of expected returns. *Review of Financial Studies*, 29(1):5–68.

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

Huang, W., Nakamori, Y., and Wang, S.-Y. (2005). Forecasting stock market movement direction with support vector machine. *Computers & Operations Research*, 32(10):2513–2522.

Jacobs, R. A., Jordan, M. I., Nowlan, S. J., and Hinton, G. E. (1991). Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87.

Jegadeesh, N. and Titman, S. (1993). Returns to buying winners and selling losers: Implications for stock market efficiency. *Journal of Finance*, 48(1):65–91.

Jegadeesh, N. and Titman, S. (2001). Profitability of momentum strategies: An evaluation of alternative explanations. *The Journal of Finance*, 56(2):699–720.

Jiang, Z., Xu, D., and Liang, J. (2017). A deep reinforcement learning framework for the financial portfolio management problem. *arXiv preprint arXiv:1706.10059*.

Kakde, D., Agrawal, S., Soni, Y., Dey, D., Chauhan, U., Raj, A., Iyengar, S. S., Goel, S., and Talukdar, P. (2024). Knowledge-infused time series forecasting: A survey. *arXiv preprint arXiv:2401.10564*.

Kelly, B. T., Malamud, S., and Zhou, K. (2024). The virtue of complexity in return prediction. *Journal of Finance*. Forthcoming.

Khaidem, L., Saha, S., and Dey, S. R. (2016). Predicting the direction of stock market prices using random forest. *arXiv preprint arXiv:1605.00003*.

Khandani, A. E. and Lo, A. W. (2007). What happened to the quants in august 2007? *Journal of Investment Management*, 5(4):29–78.

Kim, K.-j. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1-2):307–319.

Kozak, S., Nagel, S., and Santosh, S. (2020). Shrinking the cross-section. *Journal of Financial Economics*, 135(2):271–292.

Krauss, C., Do, X. A., and Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, 259(2):689–702.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. (2020). Conservative q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1179–1191.

Levich, R. M. and Thomas, L. R. (1993). The significance of technical trading-rule profits in the foreign exchange market: A bootstrap approach. *Journal of International Money and Finance*, 12(5):451–474.

Li, X., Liu, B., and Zhu, X. (2019). Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Liang, E., Liaw, R., Nishihara, R., Moritz, P., Fox, R., Goldberg, K., Gonzalez, J., Jordan, M., and Stoica, I. (2018). Rllib: Abstractions for distributed reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 3053–3062.

Lim, B., Arik, S. O., Loeff, N., and Pfister, T. (2021). Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764.

Lintner, J. (1965). The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. *The Review of Economics and Statistics*, 47(1):13–37.

Liu, Y., Swaminathan, A., Agarwal, A., and Brunskill, E. (2020). Provably good batch off-policy reinforcement learning without great exploration. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1264–1274.

Liu, Y., Wu, H., Wang, J., and Long, M. (2022). Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems*, 35:9881–9893.

Lo, A. W. (2004). The adaptive markets hypothesis: Market efficiency from an evolutionary perspective. *Journal of Portfolio Management*, 30(5):15–29.

Lopez de Prado, M. (2018). *Advances in Financial Machine Learning*. John Wiley & Sons.

Malamud, S. and Rotenberg, M. (2021). Machine learning and portfolio choice. *Annual Review of Financial Economics*, 13:253–281.

Masoudnia, S. and Ebrahimpour, R. (2014). Mixture of experts: A literature survey. *Artificial Intelligence Review*, 42(2):275–293.

McLean, R. D. and Pontiff, J. (2016). Does academic research destroy stock return predictability? *Journal of Finance*, 71(1):5–32.

Messmer, M. (2018). *Machine Learning and the Cross-section of Expected Stock Returns*. Doctoral dissertation, University of St. Gallen, St. Gallen, Switzerland.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Neely, C. J., Weller, P. A., and Dittmar, R. (1997). Is technical analysis in the foreign exchange market profitable? a genetic programming approach. *Journal of Financial and Quantitative Analysis*, 32(4):405–426.

Nie, Y., Nguyen, N. H., Sinthong, P., and Kalagnanam, J. (2023). A time series is worth 64 words: Long-term forecasting with transformers. *International Conference on Learning Representations (ICLR)*, pages 1–27.

Rabiner, L. R. (1989). A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286.

Ready, M. J. (1997). Technical trading rules and stock returns: Some new evidence. *Journal of Applied Finance*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Sezer, O. B., Gudelek, M. U., and Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied Soft Computing*, 90:106181.

Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press.

Sharpe, W. F. (1964). Capital asset prices: A theory of market equilibrium under conditions of risk. *Journal of Finance*, 19(3):425–442.

Shleifer, A. (2000). *Inefficient Markets: An Introduction to Behavioral Finance*. Oxford University Press.

Siami-Namini, S., Tavakoli, N., and Namin, A. S. (2019). The performance of LSTM and BiLSTM in forecasting time series. In *2019 IEEE International Conference on Big Data*, pages 3285–3292. IEEE.

Sirignano, J. and Cont, R. (2019). Universal features of price formation in financial markets: Perspectives from deep learning. *Quantitative Finance*, 19(9):1449–1459.

Sullivan, R., Timmermann, A., and White, H. (1999). Data-snooping, technical trading rule performance, and the bootstrap. *Journal of Finance*, 54(5):1647–1691.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press.

Sutton, R. S., Precup, D., and Singh, S. (1999). Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211.

Tay, F. E. H. and Cao, L. (2001). Application of support vector machines in financial time series forecasting. *Omega*, 29(4):309–317.

Tóth, B., Lempérière, Y., Deremble, C., de Lataillade, J., Kockelkoren, J., and Bouchaud, J.-P. (2011). Anomalous price impact and the critical nature of liquidity in financial markets. *Physical Review X*, 1(2):021006.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Łukasz Kaiser, and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., Blundell, C., Kumaran, D., and Botvinick, M. (2016). Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*.

Wang, L., Kong, S., and Guo, Z. (2023). Doubleadapt: A meta-learning approach to incremental learning for stock trend forecasting. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2367–2376.

White, H. (2000). A reality check for data snooping. *Econometrica*, 68(5):1097–1126.

Zhang, C., Li, Y., and Liu, X. (2025). Meta-learning the optimal mixture of strategies for online portfolio selection. *arXiv preprint arXiv:2505.03659*.

Zhang, Z., Zohren, S., and Roberts, S. (2020). Deep reinforcement learning for trading. *Journal of Financial Data Science*, 2(2):25–40.

A Classical Statistical Learning Methods

A.1 Linear Models

Linear regression remains foundational to quantitative finance, providing interpretable models for understanding factor exposures and risk decomposition (Shalev-Shwartz and Ben-David, 2014). Despite their simplicity, linear models offer several advantages in trading: they provide explicit coefficients that quantify the relationship between predictors and outcomes, they are computationally efficient even with high-frequency data, and they serve as benchmarks against which more complex models are evaluated.

In trading applications, linear models typically predict future returns or classify market regimes based on technical indicators (such as moving averages and momentum), fundamental variables (such as price-to-earnings ratios), and macroeconomic factors (such as interest rates and inflation). The basic linear regression model for return prediction is:

$$r_{t+1} = \alpha + \sum_{i=1}^{k} \beta_i X_{i,t} + \epsilon_{t+1} \tag{20}$$

where r_{t+1} represents future returns (typically log returns to ensure stationarity), $X_{i,t}$ denotes predictor variables at time t (which might include lagged returns, volume indicators, or cross-sectional factors), α is the intercept term (representing expected return when all predictors are zero), β_i are the regression coefficients (measuring the sensitivity of returns to each predictor), and ϵ_{t+1} is the residual term (capturing unexplained variation).

Logistic regression extends this framework to classification problems, such as predicting whether the next day's return will be positive or negative. It models the probability of market movements exceeding a threshold τ using the logistic function:

$$P(r_{t+1} > \tau) = \frac{1}{1 + e^{-(\alpha + \sum_{i=1}^{k} \beta_i X_{i,t})}} \tag{21}$$

The logistic function maps any real-valued input to a probability between 0 and 1, making it suitable for binary classification tasks such as predicting market direction or identifying regime changes.

Beyond regression-based approaches, time-series specific linear models are widely applied

in finance. The ARIMA (AutoRegressive Integrated Moving Average) family expresses returns as a linear function of past values and past errors, while allowing differencing to handle non-stationarity:

$$\phi(B)(1-B)^d r_t = \theta(B)\epsilon_t \tag{22}$$

where $\phi(B)$ and $\theta(B)$ are polynomials in the lag operator B, d is the degree of differencing, and ϵ_t is white noise. ARIMA models capture autocorrelation structures in returns and are widely used for short-term forecasting, volatility-adjusted strategies, and as benchmarks for more sophisticated models.

Similarly, exponential smoothing methods (such as Simple, Holt's, and Holt-Winters exponential smoothing) generate forecasts by assigning exponentially decaying weights to past observations:

$$\hat{r}_{t+1} = \alpha r_t + (1-\alpha)\hat{r}_t \tag{23}$$

where $\alpha \in (0,1)$ controls the rate of decay. Despite their apparent simplicity, exponential smoothing models are also linear, since the forecast is a linear combination of past returns with declining weights. They are particularly effective for capturing level, trend, and seasonal patterns in financial time series, especially when regime shifts are gradual.

While conceptually straightforward, these models rely on strong assumptions that are frequently violated in financial markets: linearity (assuming a linear relationship between predictors and outcomes), stationarity (assuming statistical properties remain constant over time), homoscedasticity (assuming constant variance of errors), and independence of observations (ignoring serial correlation in returns). Financial time series exhibit well-documented violations of these assumptions, including volatility clustering, fat tails, and regime switching.

A.2 Support Vector Machines

Support Vector Machines (SVMs) address some limitations of linear methods by employing the "kernel trick"—implicitly mapping input features into high-dimensional spaces where

complex, nonlinear decision boundaries become linearly separable (Vapnik, 1995). This approach allows SVMs to capture nonlinear relationships while maintaining the computational advantages of linear optimization.

The SVM optimization problem for classification seeks to find the hyperplane that maximizes the margin (distance) between different classes while allowing for some misclassification through slack variables. Mathematically, this is formulated as:

$$\min_{w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n} \xi_i \tag{24}$$

subject to constraints: $y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i$ and $\xi_i \geq 0$, where w is the weight vector defining the separating hyperplane, b is the bias term, C is the regularization parameter balancing margin maximization against training error minimization, ξ_i are slack variables allowing for non-separable data (points that cannot be perfectly classified), $\phi(x_i)$ represents the (potentially infinite-dimensional) feature mapping, and $y_i \in \{-1, +1\}$ are the class labels.

Common kernel functions used in financial applications include the Radial Basis Function (RBF) kernel $K(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2)$ for capturing local patterns, and the polynomial kernel $K(x_i, x_j) = (x_i^T x_j + c)^d$ for modeling interactions between features.

In trading, SVMs have been successfully applied to regime classification and directional prediction tasks. Huang et al. (2005) demonstrated that SVMs with RBF kernels outperformed backpropagation neural networks for weekly movement prediction of the NIKKEI 225 index. Kim (2003) showed that SVMs achieved 57% directional accuracy for daily KOSPI predictions, significantly outperforming traditional methods. Tay and Cao (2001) pioneered the use of SVMs for financial time series prediction, finding that epsilon-SVR with careful parameter tuning via grid search achieved superior performance in five real futures contracts. However, they require careful kernel and parameter tuning through cross-validation to avoid overfitting, and they can be computationally intensive for large datasets.

A.3 Decision Trees and Ensemble Methods

Decision trees provide interpretable, rule-based approaches to trading strategy construction that mirror the if-then logic often used by human traders. A tree recursively partitions

the feature space based on predictor variables, generating explicit rules similar to technical analysis patterns (Breiman, 2001). For example, a simple trading rule might be: "If the 50-day moving average crosses above the 200-day moving average AND volume is above its 30-day average, then buy."

The tree construction algorithm typically uses information gain or Gini impurity to select optimal splits at each node. While individual trees are prone to overfitting (memorizing noise in training data), ensemble methods combine multiple trees to create robust predictors.

Random Forests extend single trees by bootstrap aggregation (bagging), training multiple trees on random subsamples of data and features:

$$\hat{f}(x) = \frac{1}{B} \sum_{b=1}^{B} \hat{f}_b(x) \tag{25}$$

where B is the number of trees in the forest and $\hat{f}_b(x)$ is the prediction from the b-th tree trained on a bootstrap sample. This averaging reduces variance and overfitting while maintaining the non-linear modeling capacity of trees.

Gradient Boosting methods such as XGBoost and LightGBM take a different approach, iteratively fitting new models to the residuals (errors) of prior iterations (Friedman, 2001):

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{26}$$

where $F_m(x)$ is the ensemble prediction after m iterations, $h_m(x)$ is the m-th weak learner (typically a shallow tree), and γ_m is the learning rate controlling the contribution of each new model.

These ensemble methods are particularly effective for trading due to their ability to handle heterogeneous features (mixing continuous prices with categorical indicators), capture non-linear interactions without explicit feature engineering, provide feature importance rankings for interpretability, and maintain robustness through averaging or iterative refinement. Khaidem et al. (2016) applied Random Forests to predict direction in Foreign Exchange (FX) markets and showed profitability in high-frequency trading of EUR/USD. Krauss et al. (2017) found that gradient-boosted trees outperformed neural networks in 68% of months when predicting S&P 500 constituents' relative performance. Filippou et al. (2021)

demonstrated that XGBoost with carefully tuned hyperparameters (depth 3-6, learning rate 0.01-0.1) achieved Sharpe ratios above 2.0 for cryptocurrency trading.

A.4 Hidden Markov Models

Hidden Markov Models (HMMs) explicitly model regime-switching behavior in markets, acknowledging that financial dynamics change between periods of calm and turbulence, expansion and recession, or trending and mean-reverting behavior (Rabiner, 1989). HMMs assume that observable returns are generated by latent (hidden) states that evolve according to a Markov chain.

The model consists of transition probabilities governing state evolution:

$$P(S_t = j|S_{t-1} = i) = a_{ij} \tag{27}$$

where a_{ij} represents the probability of transitioning from state i to state j, and emission probabilities linking hidden states to observable returns:

$$P(r_t|S_t = i) = b_i(r_t) \tag{28}$$

where $b_i(r_t)$ is typically modeled as a Gaussian distribution with state-specific mean and variance: $\mathcal{N}(\mu_i, \sigma_i^2)$.

The Baum-Welch algorithm (a special case of Expectation-Maximization) learns model parameters from historical data, while the Viterbi algorithm identifies the most likely sequence of hidden states given observations. In trading, HMMs can identify market regimes (e.g., "risk-on" vs. "risk-off"), adjust position sizing based on detected volatility states, and generate regime-conditional forecasts.

However, HMMs assume fixed transition probabilities and emission distributions, limiting their adaptability to fast-changing markets where regime dynamics themselves evolve over time.