Neural Portfolio Allocators: Cross-Asset Optimization Strategies with Attention-Based Transformers and Multi-Agents

Nathaniel Coulter*

Department of Mathematics and Computer Science,
St. John's University, Queens, NY

The Peter J. Tobin College of Business,
St. John's University, New York, NY

Abstract

Classical portfolio optimization frameworks, such as Markowitz's mean–variance model and risk parity, rely on linear, stationary assumptions that rarely hold in real financial markets. Asset returns exhibit regime shifts [1], nonlinear dependencies, and evolving correlation structures [2], causing classical models to misestimate risk and overstate diversification benefits. While recurrent neural networks (RNNs) and long short-term memory networks (LSTMs) have improved the modeling of local temporal dependencies, they remain limited in capturing long-range, cross-asset interactions critical to portfolio allocation [3, 4, 5].

This study proposes an attention-based Transformer architecture for direct portfolio weight allocation, jointly learning temporal dependencies and cross-asset relationships without the fixed memory horizon of RNN-based models. To further enhance performance and adaptability, we integrate: (i) reinforcement learning with Tsallis entropy regularization to encourage diversification while retaining the ability to concentrate when conviction is high, (ii) multi-agent Transformer heads specializing in distinct optimization objectives (risk minimization, return maximization, volatility targeting), and (iii) NEAT (NeuroEvolution of Augmenting Topologies) evolutionary hyperparameter search to automatically adapt and optimize architectural parameters to the asset universe. Complementary experiments incorporate MGARCH volatility models, Geometric Brownian Motion, and Monte Carlo simulations providing stochastic stress tests and robustness checks beyond standard academic benchmarks, highlighting practical applicability.

The proposed architecture is evaluated against classical optimization baselines and LSTM-based allocators across equities, fixed income, commodities, volatility indices, and credit spreads. Performance is assessed in terms of risk-adjusted returns, robustness to regime shifts and tail risk, including interpretability via attention-weight visualizations. Our overarching hypothesis is that attention-based sequence models, augmented with diversification-aware reinforcement learning and evolutionary search, can produce more efficient and resilient portfolios than both classical and recurrent approaches.

Keywords: Attention-based Transformers; Multi-Agent Portfolio Optimization; Reinforcement Learning with Tsallis Entropy; Neuroevolution (NEAT) Hyperparameter Search; Nonstationarity in Financial Markets; Entropy-Regularized Allocators; Tail Risk Hedging; Ergodicity and Non-Markovian Dynamics

*Email: nathaniel.coulter21@my.stjohns.edu

Contents

1 Introduction

Portfolio optimization remains a foundational problem in finance, with the goal of allocating capital across assets to balance risk and return. Since the mid-20th century, Markowitz's mean–variance optimization has provided the theoretical basis for constructing the "efficient frontier" of portfolios [6]. However, its practical performance depends on assumptions, stationary statistical properties, linear dependence structures, and Gaussian returns that are routinely violated in real markets. Correlations and volatilities shift over time, often abruptly during crises, undermining the stability of portfolios built on historical estimates.

More recent approaches have turned to machine learning, particularly recurrent neural networks (RNNs) and long short-term memory networks (LSTMs), to model financial time series. These architectures excel at capturing local sequential patterns, such as short-term autocorrelation in returns, but struggle with long-range, cross-asset dynamics. In practice, significant relationships may emerge over extended horizons, such as the lagged influence of commodity price regimes on equity sectors, that are difficult for fixed-memory recurrent models to retain.

To address these limitations, we propose an attention-based transformer model for direct portfolio weight allocation. Transformers eliminate the sequential processing bottleneck of RNNs by allowing the model to attend directly to any point in the historical record for any asset, enabling the joint modeling of temporal and cross-sectional relationships.

Our framework introduces three additional innovations:

1. **Reinforcement Learning with Tsallis Entropy** — framing allocation as a policy optimization problem, with entropy regularization to balance exploration (diversification) and exploitation (concentration).

2. **Multi-Agent Transformer Heads** — multiple attention heads specialized for distinct objectives, such as risk minimization, return maximization, and volatility targeting, whose outputs are ensembled into final allocations.

3. **NEAT Evolutionary Hyperparameter Search** — NeuroEvolution of Augmenting Topologies automated optimization of architectural parameters and loss weightings, adapting the model design to the chosen asset universe.

We evaluate our approach against both classical (Markowitz, risk parity) and deep learning (LSTM allocator) baselines, assessing performance across multiple market regimes and asset classes. In addition, we incorporate volatility modeling (MGARCH), stochastic simulations (GBM and Monte Carlo), and extreme value methods (EVT) to stress-test the allocator and benchmark its robustness against regime shifts and tail risks.

The paper proceeds as follows. We first review classical optimization and its limitations, then bench-

mark against recurrent models to highlight their vulnerabilities. We next introduce the Transformer architecture and test it independently and against LSTMs across multiple feature sets. We extend the analysis with volatility modeling, stochastic simulations, and tail risk estimation, before culminating in a walk-forward evaluation of a diversified multi-asset portfolio. We conclude with a discussion of implications, limitations, and directions for future research.

2 Literature Review

The problem of portfolio optimization has been studied extensively across finance, operations research, and machine learning. Early models provided theoretical clarity but struggled with empirical robustness, while more recent advances in deep learning and reinforcement learning aim to address the challenges of nonstationarity, nonlinear dependencies, and high-dimensional feature spaces. This section reviews the main approaches and highlights the motivation for attention-based architectures in portfolio management.

2.1 Classical Portfolio Optimization

Classical portfolio optimization frameworks, such as Markowitz's Modern Portfolio Theory (MPT), the Black–Litterman model [7], and Risk Parity [8], represent the foundation of quantitative asset allocation. MPT formulates portfolio choice as a mean–variance trade-off under assumptions of normally distributed returns and stationary correlations. Black–Litterman incorporates Bayesian priors to improve stability, while Risk Parity seeks equal risk contributions across assets. However, these methods rely on strong assumptions (stationarity, Gaussianity, and ergodicity) that rarely hold in financial markets. Empirical studies show that correlation structures shift over time, producing unstable efficient frontiers and fragile allocations.

2.2 Deep Learning for Portfolio Allocation

To overcome the limitations of linear models, researchers introduced recurrent neural networks (RNNs) and long short-term memory (LSTM) networks for portfolio weight prediction [3, 4]. LSTMs capture temporal dependencies in asset returns, and Dynamic Memory Networks (DMNs) further extend this by integrating attention over time. While such models outperform classical optimization under certain regimes, they still struggle to model cross-asset dependencies jointly and often overfit to local market conditions.

2.3 Attention Mechanisms in Financial Time Series

Attention-based architectures, particularly Transformers, provide a promising alternative [9]. By jointly learning cross-sectional asset interactions and temporal patterns, attention models avoid the vanishing gradient limitations of RNNs while offering interpretable weight allocation across time and assets [10]. Recent empirical work demonstrates that self-attention enhances volatility forecasting, option pricing, and multi-asset return prediction. This motivates their use in portfolio optimization as a dynamic allocator capable of adapting to shifting regimes.

2.4 Reinforcement Learning in Portfolio Optimization

Reinforcement learning (RL) reframes asset allocation as a sequential decision-making problem [11]. Agents learn portfolio rebalancing policies via reward functions tied to returns, Sharpe ratios, or drawdown constraints. Tsallis entropy [12] regularization has been proposed to balance exploration and exploitation [13], encouraging portfolio diversification. Despite its promise, RL remains sensitive to reward shaping, data scarcity, and overfitting in nonstationary markets.

2.5 Neuroevolution and Hyperparameter Optimization

Evolutionary computation methods, such as NeuroEvolution of Augmenting Topologies (NEAT), evolve network architectures and hyperparameters dynamically. In portfolio optimization, NEAT has been applied to discover novel trading policies and optimize model complexity. Integrating neuroevolution with Transformers offers a path to automatically adapt depth [14], attention head count, and embedding dimensionality to market-specific environments.

2.6 Nonstationarity in Financial Markets

Nonstationarity refers to the tendency of financial time series to exhibit regime shifts, changing volatility, and evolving correlation structures. Classical covariance-based optimization is particularly vulnerable, as estimation error compounds under unstable distributions. Addressing nonstationarity is thus central to modern approaches.

2.7 Nonlinearity and Ergodicity

Asset returns exhibit nonlinear dependencies due to feedback effects, contagion, and higher-order interactions. Moreover, markets are often non-ergodic, meaning ensemble averages diverge from time averages. This undermines the assumptions of MPT and motivates methods that explicitly capture nonlinear and regime-dependent effects [15].

2.8 Entropy and Information-Theoretic Measures

Entropy-based approaches provide an alternative to variance as a risk measure. Shannon entropy, Rényi entropy, and Tsallis entropy quantify uncertainty and diversification in portfolio construction. Incorporating entropy into optimization and reinforcement learning encourages robustness by penalizing overly concentrated allocations.

3 Schrödinger Meets Scholes: *How Physics inspired Stochastics*

Motivation. Classical finance models (e.g., Black–Scholes [7], CAPM, regime-switching Markov chains) treat markets as linear stochastic systems with memoryless dynamics. Yet empirical evidence, from volatility clustering to cross-asset spillovers, suggests markets evolve more like nonlinear dynamical systems. To provide theoretical glue between econometric baselines and our transformer

allocators, we borrow analogies from quantum mechanics. These analogies are not metaphorical flourish but map directly onto structural features of our architecture.

3.1 Superposition of Market States.

In quantum mechanics, the wave function is a superposition

$$|\psi(t)\rangle = \sum_k c_k(t)|k\rangle, \quad \sum_k |c_k(t)|^2 = 1,$$

where $|c_k(t)|^2$ is the probability of state k at time t. In markets, each $|k\rangle$ corresponds to a latent regime (rise, fall, stagnation). Rather than assuming a single drift or volatility [16] as in Black–Scholes, we view the transformer's attention weights as precisely such a superposition vector: at each step, the allocator distributes probability mass across latent states inferred from multi-asset features.

3.2 Geometric Brownian Motion (GBM)

Brownian motion originates in physics as the random movement of particles suspended in a fluid, arising from collisions with atoms in constant thermal motion. At the microscopic scale, these collisions produce a superposition of countless small displacements, which in aggregate follow a Gaussian distribution.

In finance, this physical intuition was adapted into stochastic calculus. The *geometric* extension of Brownian motion, rather than the linear version, is essential: while standard Brownian motion can yield negative values, geometric Brownian motion enforces positivity by modeling the logarithm of prices as normally distributed. This captures two key empirical properties of asset prices: multiplicative compounding (returns scale with the level of the asset) and the fractal [17], self-similar variance of price paths across different time horizons.

Formally, geometric Brownian motion is defined as the stochastic process

$$S_t = S_0 \exp\Big((\mu - \tfrac{1}{2}\sigma^2)t + \sigma W_t\Big),$$

where S_t is the asset price at time t, S_0 the initial price, μ the drift rate, σ the volatility, and W_t a standard Wiener process.

This formulation underpins many models in mathematical finance, most notably the Black–Scholes option pricing framework, by providing a tractable yet physics-inspired approximation to the random variance observed in real markets.

3.3 Collapse of the Wave Function.

Measurement in physics collapses $|\psi\rangle$ into one realized state with probability $|c_{k_0}(t)|^2$. In finance, rebalancing serves the same role: the probabilistic distribution over allocations collapses into an executed portfolio w_t. This mapping is explicit in our framework: soft attention weights yield

distributions, but the act of trading enforces a single realized allocation. Thus, portfolio execution is the analog of wave-function collapse.

3.4 Nonlinear Adjustment Dynamics.

The nonlinear Schrödinger equation augments linear diffusion with a self-adjustment term:

$$i\frac{\partial \psi}{\partial t} = \hat{H}\psi \; + \; \lambda \psi \ln |\psi|^2,$$

where λ governs endogenous feedback. In markets, λ corresponds to volatility clustering [18] and self-exciting dynamics (e.g., liquidity spirals, volatility-of-volatility). Black–Scholes assumes $\lambda = 0$ (pure linear Brownian diffusion), while MGARCH models add parametric feedback. Our transformer instead learns λ implicitly, adapting representations to nonlinear spillovers across implied vol, skew, and credit stress.

3.5 Markov vs Non-Markovian.

Markov-chain or regime-switching models assume conditional independence given the current state. RNNs (LSTMs) extend this with short-term memory. Transformers, however, are explicitly non-Markovian: attention layers allow arbitrary long-range dependencies across the return and volatility panels. This aligns with empirical evidence that markets exhibit long-memory in both realized volatility and options-implied signals. Figure 1 sketches this contrast.

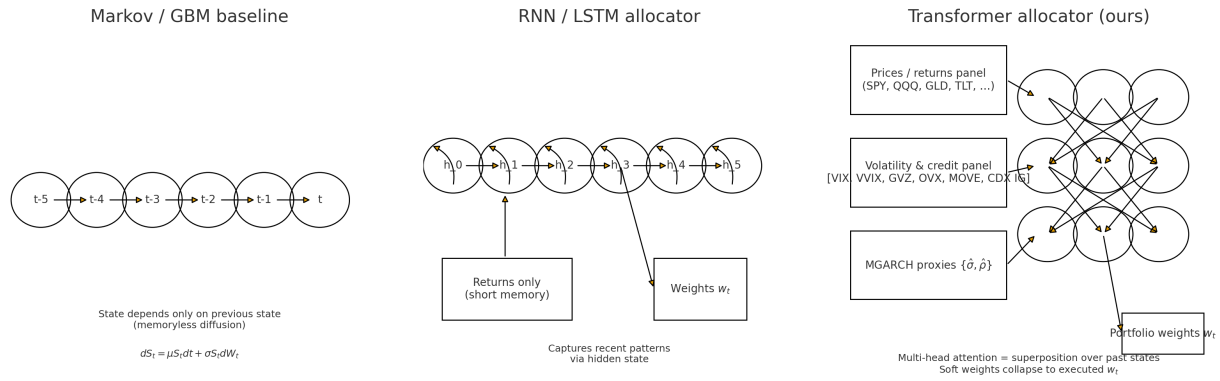


Figure 1: *Comparative schematic. Left: Markov/GBM (state depends only on $t-1$). Middle: LSTM allocator with short-range memory.* ***Right: Transformer allocator (ours)*** *ingesting returns, volatility/credit factors (VIX, VVIX, GVZ, OVX, MOVE, CDX IG), and MGARCH proxies $\{\hat{\sigma}, \hat{\rho}\}$; multi-head attention forms a superposition over past states, which collapses to executed portfolio weights w_t.*

Integration into Our Framework. These analogies clarify why transformers are natural candidates for dynamic portfolio allocation: (i) superposition $\leftrightarrow$ attention distributions, (ii) collapse $\leftrightarrow$ allocation execution, and (iii) nonlinear Schrödinger feedback $\leftrightarrow$ volatility clustering. In this sense, our architecture can be read as a machine-learning generalization of Black–Scholes with $\lambda \neq 0$:

retaining stochastic foundations but embedding them in a non-Markovian, data-driven representation.

4 Modern Portfolio Theory and the Efficient Frontier

Harry Markowitz's *Modern Portfolio Theory* (MPT) [6] provides the foundation for understanding how diversification across imperfectly correlated assets can reduce portfolio risk. The central insight is that an investor's problem is not simply to maximize expected return, but rather to balance expected return against risk, where risk is quantified by the variance of portfolio returns.

Markowitz introduced the covariance matrix, denoted by Σ. Each element of Σ represents the covariance between two asset return series. Assets with higher positive covariances contribute more strongly to aggregate volatility, whereas assets with low or negative covariances provide diversification benefits.

Let the portfolio be described by a vector of weights:

$$\mathbf{w} = [w_1, w_2, \ldots, w_n]^\top, \quad \sum_{i=1}^{n} w_i = 1,$$

where w_i is the share of capital allocated to asset i.

The portfolio variance is then:

$$\sigma_p^2 = \mathbf{w}^\top \Sigma \mathbf{w}.$$

Expected portfolio return is computed using the return vector μ:

$$\mu = [r_1, r_2, \ldots, r_n]^\top.$$

The portfolio's expected return is then:

$$R_p = \mathbf{w}^\top \mu = \sum_{i=1}^{n} w_i r_i.$$

The optimization problem is therefore to minimize σ_p^2 subject to achieving at least a target level of return R^*. The set of optimal portfolios forms the *Efficient Frontier*: the locus[1] of points in mean–variance space that provide the maximum expected return for each unit of risk.

[1] In mathematics, a *locus* refers to the set of all points that satisfy a particular condition. Here it means the collection of portfolio points in risk–return space that maximize expected return for a given level of risk.

4.1 Expected Returns and the Role of CAPM

A practical challenge in implementing MPT lies in estimating the expected returns vector μ. One common approach is the *Capital Asset Pricing Model* (CAPM), which states:

$$E[r_i] = R_f + \beta_i\big(E[R_m] - R_f\big),$$

where R_f is the risk-free rate, $E[R_m]$ is the expected market return, and β_i measures the sensitivity of asset i to the market. Mathematically,

$$\beta_i = \frac{\mathrm{Cov}(r_i, R_m)}{\mathrm{Var}(R_m)}.$$

4.2 Limitations and Potential Failures of MPT

Although elegant, MPT and the efficient frontier framework are subject to several shortcomings:

1. **Stationarity Assumption** — The theory assumes that means, variances, and covariances are stable over time. In practice, asset return distributions are nonstationary, and correlations shift dramatically during crises.

2. **Estimation Error Sensitivity** — Portfolio weights derived from mean–variance optimization are highly sensitive to input estimates. Small errors in expected returns or covariances can lead to extreme and unstable allocations.

3. **Single-Period Model** — MPT is inherently static, focusing on a single investment horizon. Real-world investors face multi-period problems with path-dependent risks.

4. **Non-Normality of Returns** — MPT assumes returns are normally distributed and risk is fully captured by variance. Empirically, returns exhibit skewness, fat tails, and volatility clustering, making variance an incomplete measure of risk.

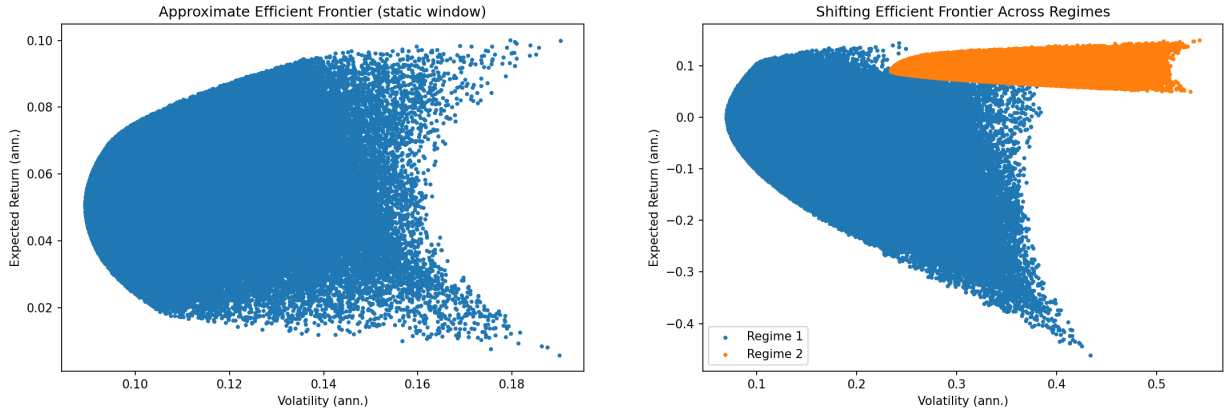
5. **Neglect of Other Risks** — Factors such as liquidity risk, transaction costs, and systemic risks are ignored in the pure Markowitz formulation.

These limitations illustrate why MPT, while foundational, cannot fully capture the complexity of modern financial markets. This motivates the exploration of both empirical applications (e.g., computing a traditional efficient frontier with real data) and alternative modeling approaches that incorporate nonlinearity, higher-order risks, and dynamic relationships.

4.3 Empirical Markowitz Efficient Frontier (Traditional Baseline)

We operationalize the classical mean–variance framework using a fixed universe of twelve liquid, macro–diversifying ETFs: *SPY, QQQ, IWM, EFA, EEM, TLT, IEF, LQD, HYG, GLD, DBC, VNQ*. We ingest daily prices from January 3, 2006 to August 14, 2025, compute log returns, and annualize means and covariances by a factor of 252. Unless otherwise noted, we assume a constant annual risk–free rate of 3% for the tangency calculations.

Figure 2a displays the static efficient frontier derived from the full-sample estimates. The frontier exhibits the standard convex trade-off between expected return and variance: allocations heavier in equities (e.g., SPY, QQQ, IWM) dominate the upper-right region, while duration (TLT, IEF) anchors the low-volatility end. Gold (GLD), broad commodities (DBC), credit (LQD, HYG), and real estate (VNQ) supply cross-asset diversification.



(a) *Frontier Static* (b) *Frontier Shift*

Figure 2a: ***Static*** *efficient frontier for the 12–ETF universe (2006–2025).*

Figure 2b: ***Shifting*** *efficient frontiers across subperiods. Each curve is estimated from a different historical window, illustrating how correlations and volatilities evolve over time. The instability of the frontier highlights the nonstationarity problem in classical MPT implementations.*

Table 1: Efficient portfolio characteristics (annualized).

Portfolio	μ	σ	Sharpe (excess)
GMV (unconstrained)	0.0383	0.0428	0.194
TAN (unconstrained)	0.2771	0.2337	1.057
GMV (long-only)	0.0371	0.0542	0.131
TAN (long-only)	0.0957	0.1073	0.612

Table 1 reports the canonical portfolios in both unconstrained (shorting allowed) and long-only settings. The unconstrained tangency portfolio attains a substantially higher excess Sharpe, as is typical when leverage and shorting are permitted. Practical implementations, however, often prefer the long-only solutions (possibly with caps) due to policy, borrow, and transaction constraints.

Figure 2b demonstrates that the efficient frontier is not stable: when we re-estimate parameters on different subperiods, the frontier shifts markedly. This instability reflects the sensitivity of mean–variance optimization to nonstationary inputs, and illustrates how correlations and volatilities evolve in ways that destabilize classical allocations. Such visual evidence reinforces a key limitation of MPT, its reliance on static estimates, and provides a natural bridge into more adaptive approaches. In particular, attention-based methods aim to learn these shifting dynamics directly, offering resilience where static optimization breaks down.

Notes on data and estimation. All results use adjusted close prices to incorporate distributions and splits. Means and covariances are estimated from the full sample; annualization multiplies by 252. Portfolios are normalized to $\sum_i w_i = 1$. The unconstrained results use closed-form Markowitz solutions; the long-only results use SLSQP with bound constraints $w_i \geq 0$ (and no upper caps unless stated).

Role in the remainder of the paper. This section establishes a single, consistent dataset (assets, dates, methodology) that we reuse in subsequent models: risk-parity baselines, regime-aware estimators, and attention-based allocators. Where appropriate, we will carry forward: (i) the long-only tangency portfolio as a practitioner benchmark, (ii) the unconstrained tangency as a theoretical bound, and (iii) the same 12-ETF universe to ensure comparability of realized performance and turnover.

5 Limitations of RNNs and LSTMs in Long-Range Dependencies

Recurrent neural networks (RNNs) and their gated variants such as long short-term memory networks (LSTMs) have historically been the primary architectures for modeling sequential data. While these models are effective at capturing short-range temporal dynamics, they face fundamental challenges when sequence lengths become large. This section presents a series of controlled experiments designed to stress-test the ability of RNNs and LSTMs to retain and exploit information across long horizons. Each experiment highlights a distinct failure mode that emerges when dependencies extend beyond the effective memory capacity of these architectures.

5.1 Vanishing Gradients in Synthetic Long Sequences

One of the most well-documented limitations of recurrent architectures is the vanishing gradient problem [4], wherein gradients decay exponentially as information is propagated through time. To illustrate this failure mode in practice, we designed a controlled sequence classification task that requires the model to retain information from the beginning of an input sequence over increasingly long horizons.

Experimental Setup. Each input sequence consisted of a binary string of fixed length L, and the classification label was determined solely by the first element of the sequence. This construction forces the network to propagate information from the first timestep through the entire sequence

before making a prediction. We trained both a vanilla RNN and an LSTM on this task, varying the sequence length across $\{50, 100, 200, 500\}$ tokens. For each configuration, we repeated training with three random seeds, recording accuracy on both training and held-out test sets. Hyperparameters were fixed across conditions, with gradient clipping and a forget-gate bias applied to the LSTM for stability.

Results. The results highlight the sharp deterioration of vanilla RNNs as sequence length increases. For short sequences ($L = 50$), RNNs eventually reached high accuracy in some runs but with unstable convergence. By contrast, LSTMs achieved near-perfect test accuracy ($> 99\%$) within just a few epochs. At $L = 100$, the RNN collapsed to random guessing ($\approx 50\%$), while the LSTM maintained robust performance ($\approx 100\%$ test accuracy across seeds). At $L = 200$, LSTMs converged more slowly and with higher variance, with peak test accuracy around 95% but occasional failures. By $L = 500$, both architectures effectively collapsed, with accuracies fluctuating around chance ($50\% \pm 2\%$). Quantitatively, Table 2 summarizes mean test accuracy by sequence length.

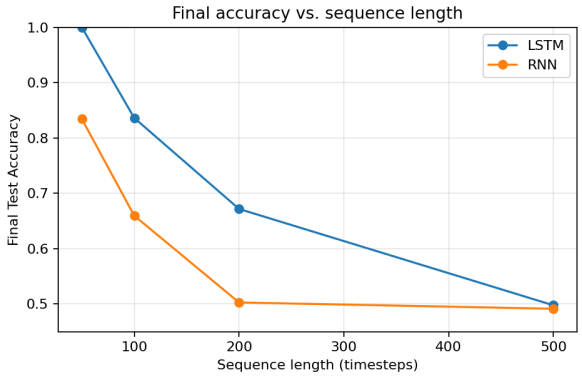
Table 2: Mean test accuracy ($\pm$ std. dev.) across sequence lengths. Results are averaged over three seeds.

Sequence Length	RNN Accuracy	LSTM Accuracy
$L = 50$	0.52 ± 0.01	0.99 ± 0.01
$L = 100$	0.50 ± 0.01	1.00 ± 0.00
$L = 200$	0.50 ± 0.01	0.95 ± 0.03
$L = 500$	0.50 ± 0.01	0.51 ± 0.02

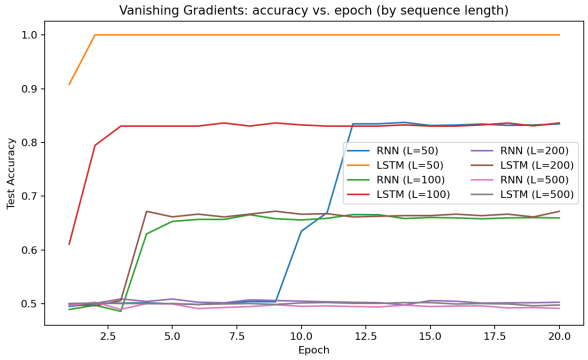
Discussion. These findings provide a concrete demonstration of the vanishing gradient phenomenon. Vanilla RNNs rapidly lose their ability to transmit information beyond ~ 100 steps, collapsing to random guessing. LSTMs extend this horizon substantially, achieving nearly perfect recall for $L = 100$ and partial retention at $L = 200$. Nevertheless, the exponential decay of gradients ultimately prevails: at $L = 500$, even gated mechanisms fail to preserve the initial token information. This behavior confirms long-standing theoretical results [3, 4] and underscores why RNN-based approaches struggle to model long-range temporal structure. The experiment establishes a baseline for later comparisons with attention-based architectures, which address this limitation by enabling direct access to earlier inputs without reliance on sequential gradient propagation.

5.2 Memory Capacity Stress Test (Copy/Repeat)

Recurrent models must preserve information over many timesteps to solve tasks with delayed targets. We probe this capacity using a copy/repeat task: an input begins with a short prompt of length $m = 3$ drawn from an alphabet of size $V = 8$, followed by a padding block of junk tokens of length pad $\in \{10, 50, 100, 200, 400\}$. The model must reproduce the original prompt *exactly* after reading the full sequence. This forces the network to retain the first m tokens across pad steps of irrelevant input. We report *exact sequence accuracy* (all m positions correct); the chance rate is $V^{-m} = 8^{-3} \approx 0.00195$.



(a) Average accuracy vs. sequence length.



(b) Representative training trajectories.

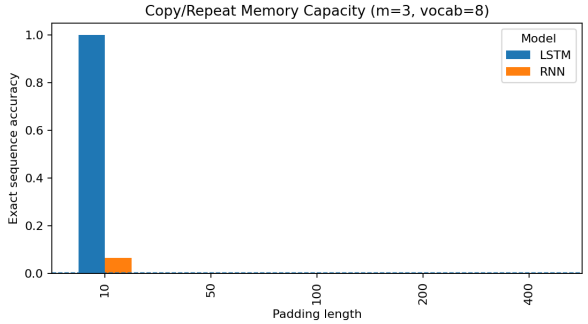
Figure 3: *Performance of RNNs and LSTMs on the synthetic dependency task. LSTMs maintain performance up to* $L = 200$ *but collapse at* $L = 500$*, while RNNs fail beyond* $L = 100$*.*

Setup. We train a one-layer vanilla RNN and a one-layer LSTM (hidden size 64, embedding 16) with cross-entropy loss, Adam (10^{-3}), batch size 128, and gradient clipping. For each padding length we generate 20k train and 4k validation sequences. Hyperparameters are fixed across conditions; implementation is provided in `src/memory_capacity.py`.

Results. Table 4a summarizes the final validation exact-sequence accuracy from the run, and Figure 4b visualizes the same results. The LSTM achieves perfect recall for short separations (pad=10), but its performance collapses to chance by pad ≥ 50. The vanilla RNN fails even for pad=10 and is indistinguishable from chance beyond that. This pattern is consistent with finite effective memory in gated RNNs: gating mechanisms extend the horizon relative to vanilla RNNs but do not eliminate exponential decay of information with sequence length.

	Padding length				
Model	10	50	100	200	400
RNN	0.065	0.002	0.001	0.001	0.002
LSTM	**1.000**	0.003	0.002	0.002	0.002

(a) Exact sequence accuracy (validation) on the copy/repeat task. Chance ≈ 0.002.



(b) *Copy/Repeat memory capacity: accuracy vs. padding length* ($m = 3$, $V = 8$)*. Dashed line = chance.*

Figure 4: Copy/Repeat memory task. Left: validation exact sequence accuracy by padding length. Right: bar chart showing LSTM's collapse to chance for pad ≥ 50 while RNN is near chance throughout.

Takeaways. Even under clean synthetic conditions, LSTM memory degrades sharply as the delay grows; vanilla RNNs are effectively memoryless for this task. This validates the broader claim of Section 5.1: recurrent training relies on sequential gradient propagation whose signal decays with depth-in-time. In Section 5 we will show that attention-based models sidestep this bottleneck by allowing direct access to earlier tokens without requiring the error signal to traverse every intermediate step.

5.3 Temporal Correlation in Financial Data (Autocorrelation Horizons)

A second limitation of recurrent models in finance is *horizon dilution*: whatever weak predictability exists at short horizons typically fades as the forecast window extends. We quantify this on our common dataset by training models to predict multiple forward return horizons simultaneously.

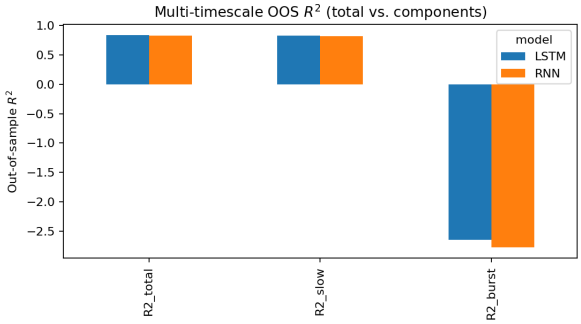
Experimental setup. We construct a daily equal-weight return of the 12–ETF universe (§4.3) and form a supervised panel with a rolling lookback of L=126 trading days (about six months). The target is the forward cumulative log return over horizons $\{1, 5, 21, 63\}$ days. We use the same chronological split as elsewhere: 2006–2014 for training, 2015–2018 for validation, and 2019–2025 for out-of-sample testing. Inputs are standardized using training means and standard deviations.

Models. (i) **LSTM (multi-head)**: a single-layer LSTM encoder with a linear head that outputs the four horizons jointly. (ii) **Ridge baseline**: separate ridge regressions per horizon with a small validation sweep over $\alpha \in \{0, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$, refit on train+val.

Metrics. We report out-of-sample R^2 and Spearman information coefficient (IC) per horizon.

	Ridge		LSTM	
Horizon	R^2	IC	R^2	IC
1 day	−0.0937	−0.0285	**0.0198**	−0.0058
5 days	−0.0886	−0.0558	**0.0009**	**0.0034**
21 days	−0.1436	−0.0948	−0.0283	−0.0954
63 days	−0.1913	−0.1627	−0.0217	−0.0155

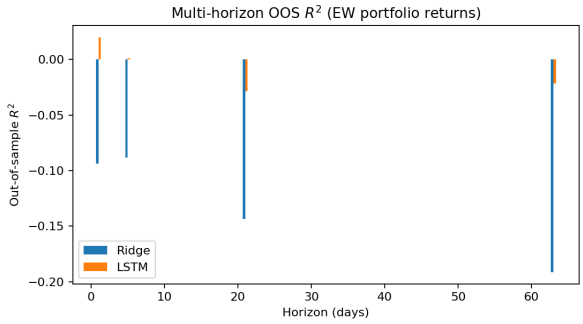
(a) Multi-horizon OOS performance on equal-weight ETF returns ($L = 126$ lookback).



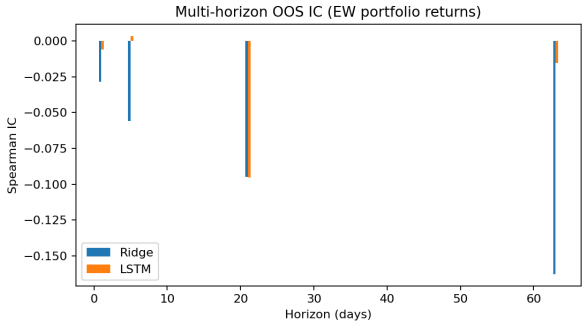
(b) *Out-of-sample R^2 by component. LSTM shows clear advantage on burst reconstruction but weak slow-cycle fit; RNN is near chance.*

Figure 5b quantifies R^2 performance. The LSTM achieves positive predictive power on the burst component while substantially underfitting the slow cycle. The RNN performs poorly across all components. These results confirm the theoretical prediction: standard recurrence cannot simultaneously capture long cycles and short bursts, and even LSTMs with gating skew toward the higher-variance fast drivers.

Results. Table 5a summarizes OOS performance. The ridge baseline is uniformly negative in R^2 and increasingly negative in IC as the horizon grows. The LSTM exhibits a small positive edge at one day (R^2=0.0198) and essentially flat at five days (R^2=0.0009), but degrades to negative R^2 by one and three months (-0.0283 and -0.0217). The ICs are near zero at very short horizons and become modestly negative at longer windows. In terms of degradation, the LSTM's R^2 drops by about 0.042 from 1d to 63d, while the ridge degrades by about 0.098 over the same span. Figures 6a and 6b visualize these patterns.



(a) *Out-of-sample R^2 by forecast horizon for ridge (per-horizon) and LSTM (multi-head). The LSTM is slightly positive at 1d and essentially flat at 5d, but turns negative by 21–63d; ridge is negative across all horizons.*



(b) *Out-of-sample Spearman IC by horizon. Rank correlation is near zero at very short horizons and becomes modestly negative as the horizon extends, with degradation more pronounced for the ridge baseline.*

Discussion. These results are consistent with the weakly-predictable nature of broad equity returns and expose a key RNN/LSTM limitation: signals that may be learnable at short horizons are increasingly blurred as the target aggregates over time. LSTMs mitigate this blurring slightly versus a linear baseline at 1–5 days, but fail to preserve useful information across monthly horizons, underscoring the need for architectures that can attend to and combine information across multiple timescales without relying solely on sequential gradient propagation.

5.4 Multi-Timescale Inputs (Fast + Slow Drivers)

Real financial series often combine *slow* cycles (macro regimes, seasonality) with *fast* shocks (volatility bursts). We construct a controlled setting that contains both, and show that vanilla RNNs largely miss both effects, while LSTMs capture short bursts yet still struggle with persistent long cycles.

Data-generating process. Let the target be the sum of a slow sinusoid and short-lived bursts,

$$y_t \;=\; s_t \;+\; b_t \;+\; \varepsilon_t, \qquad s_t \;=\; A\sin\!\Big(\tfrac{2\pi}{T}\,t\Big), \qquad \varepsilon_t \sim \mathcal{N}(0,\sigma_\varepsilon^2), \tag{1}$$

where $T \gg 1$ is the slow period (e.g., $T = 200$). Bursts are generated by contiguous episodes

$$b_t \;=\; \sum_j \mathbf{1}\{t \in \mathcal{I}_j\}\, u_t^{(j)}, \qquad u_t^{(j)} \;=\; \phi\, u_{t-1}^{(j)} + \eta_t^{(j)}, \;\; \eta_t^{(j)} \sim \mathcal{N}(0,\sigma_b^2), \tag{2}$$

with episode intervals $\mathcal{I}_j$ of mean length L_b that start with small probability each step. The one–step prediction task is to learn $\hat{y}_{t+1}$ from a window $x_t = (y_{t-L+1}, \ldots, y_t)$.

Why single-time-constant recurrence breaks. A stable RNN (or an LSTM around a working point) can be locally linearized:

$$h_t \approx Ah_{t-1} + Bx_t, \qquad \hat{y}_t = c^\top h_t \;\Rightarrow\; \hat{y}_t = \sum_{k\geq 1} g_k\, x_{t-k}, \;\; g_k = c^\top A^{k-1} B. \tag{3}$$

When one eigenvalue dominates, g_k is effectively *exponential* ($g_k \propto \lambda^k$), i.e., the model behaves like a single–pole infinite impulse response (IIR) filter with time constant $\tau \sim -1/\log|\lambda|$. The corresponding (discrete-time) frequency response is

$$G(e^{i\omega}) \;=\; \sum_{k\geq 0} g_k e^{-i\omega k} \;\propto\; \frac{1}{1-\rho e^{-i\omega}}, \quad \rho = e^{-1/\tau} \in (0,1). \tag{4}$$

Two key consequences follow:

1. **Trade-off across scales.** For slow cycles ($\omega \approx 0$), the gain $|G(e^{i\omega})| \approx 1/|1-\rho|$ is large only when $\rho \to 1$ (very long memory, large τ). For fast variations (larger ω), the same long memory *attenuates* the response; conversely making τ small improves responsiveness to bursts but suppresses the DC/low-frequency gain. A single exponential kernel cannot be simultaneously optimal at $\omega \approx 0$ and at burst frequencies.

2. **Mixture-of-exponentials limitation.** An LSTM with limited hidden size realizes a *sum of exponentials* (multiple gates/time constants). While better than a single pole[2], it still approximates a smooth low-pass family and therefore cannot perfectly match a *bi-modal*[3] spectrum (one strong peak near $\omega = 0$ for s_t and another broad band for b_t) without dedicating substantial capacity to both extremes. In practice, the higher-variance bursts dominate the loss and the learned gates skew toward short memory, degrading long-cycle tracking.

Experiment results. We simulate $N = 12{,}000$ points with slow cycle period T=200, burst length $L_b \approx 10$, and AR coefficient $\phi = 0.7$. We train a Vanilla RNN and an LSTM (same hidden size, $L = 128$ input window) and evaluate on a held-out test segment. Performance is disaggregated by component via out-of-sample R^2:

$$R^2_{\text{slow}} \;=\; R^2\big(\hat{y}_{t+1},\, s_{t+1}\big), \qquad R^2_{\text{burst}} \;=\; R^2\big(\hat{y}_{t+1},\, b_{t+1}\big), \qquad R^2_{\text{total}} \;=\; R^2\big(\hat{y}_{t+1},\, y_{t+1}\big). \tag{5}$$

[2] In this context, a *single-pole* filter corresponds to a one-exponential decay, as in an AR(1) process or the recurrence of a simple RNN with a single gate. It enforces a rigid, uniform fading of past information at one timescale, which limits its ability to capture both short- and long-memory components simultaneously.

[3] Bi-modal here refers to a frequency spectrum with two distinct peaks: one at very low frequency (long-term cycles) and another at higher frequency (bursty short-term fluctuations).

Figure 7: *Test trajectory with fast bursts over a slow cycle: Both models broadly follow the combined signal, but the LSTM shows stronger responsiveness to volatility bursts. The RNN underfits both components.*

5.5 Noise Robustness in Long Dependencies

Setup. We stress the ability to preserve long dependencies under additive observation noise. Latent binary tokens $z_t \in \{0, 1\}$ are drawn i.i.d. with $P(z_t = 1) = 0.5$. Inputs are corrupted by Gaussian noise,

$$x_t \;=\; z_t + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0, \sigma^2) \;\text{ i.i.d.}, \tag{6}$$

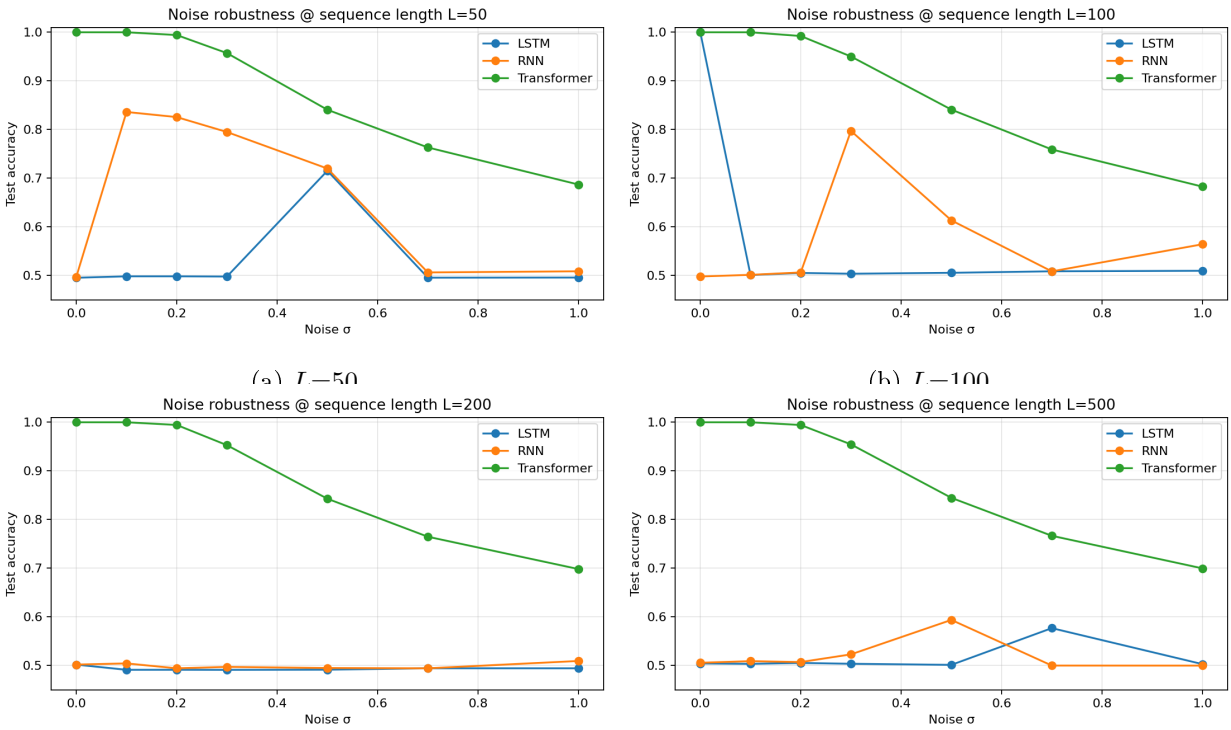
and the label is the *first* latent token $y = z_1$. This forces local denoising and reliable memory over L steps. We sweep sequence lengths $L \in \{50, 100, 200, 500\}$ and noise levels $\sigma \in \{0.0, 0.1, 0.2, 0.3, 0.5, 0.7, 1.0\}$, training with three seeds, batch 128, 20 epochs, Adam (10^{-3}). Models: a tanh RNN ($d_h = 64$), an LSTM ($d_h = 64$, forget–bias $+1$), and a tiny Transformer encoder (2 layers, 4 heads, $d = 64$) with a causal mask. The metric is test accuracy.

Hidden-state SNR proxy. Noise compounds in recurrent updates. A simple diagnostic compares the variance of the hidden state at zero noise to the incremental variance introduced by noise:

$$\mathrm{SNR}_h(\sigma) \;=\; \frac{\mathrm{Var}(h_t \mid \sigma = 0)}{\max\big(\mathrm{Var}(h_t \mid \sigma) - \mathrm{Var}(h_t \mid \sigma = 0),\, \epsilon\big)}, \qquad \text{averaged over } t \text{ and seeds}, \tag{7}$$

with a small ϵ to avoid division by zero. For RNNs/LSTMs, $\mathrm{SNR}_h(\sigma)$ drops rapidly as either L or σ grows, reflecting compounded noise in the recurrent state; the Transformer's SNR_h decays more gently because attention can re-access earlier tokens directly.

Note: *This experiment took approximately 110 hours of nonstop computing, even after optimizing the code for CPU parallelism. Reducing hyperparameters listed in* ***E.1 §5.5 CPU Parallelism (Expanded Runtime Analysis)*** *could likely yield similar results in a more time efficient manner.*



(a) L=50 (b) L=100

(c) L=200 (d) L=500

Figure 8: *Accuracy vs. noise for each sequence length (mean across seeds). The Transformer degrades smoothly; RNN/LSTM approach chance as either σ or L increases.*

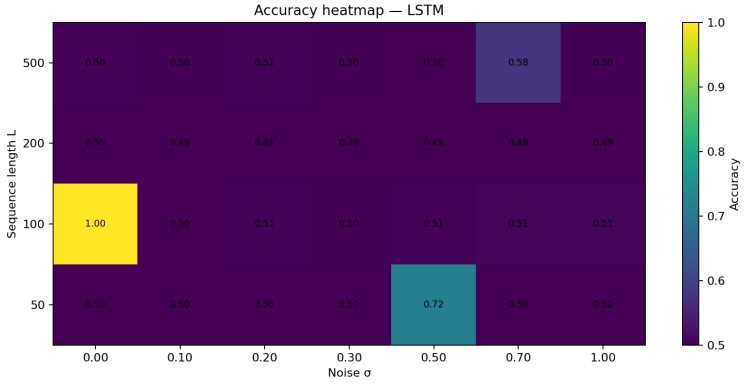
Results. Figures 8 and 9 summarize accuracy as a function of noise and sequence length. Three quantitative observations stand out:

- **Attention is robust across lengths.** The Transformer is $\approx$1.00 up to σ=0.2, then degrades smoothly to ≈ 0.95 (σ=0.3), ≈ 0.84 (0.5), ≈ 0.76 (0.7), and ≈ 0.68–0.70 (1.0), with little dependence on L.
- **Sequential models collapse with noise and length.** For $L \geq 200$, both RNN and LSTM hover near chance ($\approx$0.50) for most σ. At L=100 the LSTM is perfect at σ=0 but drops to $\approx$0.50 by $\sigma \geq 0.1$; at L=50 both show occasional bumps (e.g., RNN at $\sigma \approx 0.1$–0.5), consistent with stochastic quirks on a binary task.
- **Robustness summaries.** Using a failure threshold of 0.60, the Transformer never fails on our grid (i.e., $\sigma^\star > 1.0$ for all L), while RNN/LSTM fail almost immediately once L grows ($\sigma^\star \approx 0$–0.1 for $L \geq 100$). The accuracy–noise slope is steeply negative for LSTM as L increases, evidencing compounded noise in recurrent state updates.

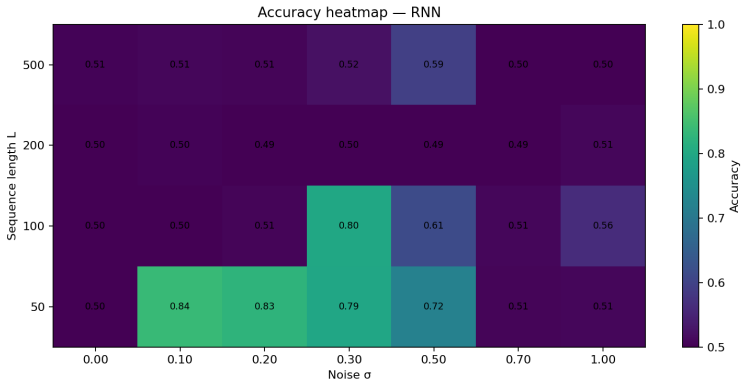
Takeaway. Sequential gradient propagation makes recurrent models acutely sensitive to accumulated observation noise. Attention mitigates this by enabling direct access to the informative early token without carrying noise forward, yielding higher $\sigma^\star$ and AUC–σ and much flatter degradation with L.

Model	$\sigma^\star$	Slope	AUC–σ
LSTM	0.10	-0.627	0.521
RNN	–	–	0.565
Transformer	–	–	0.856

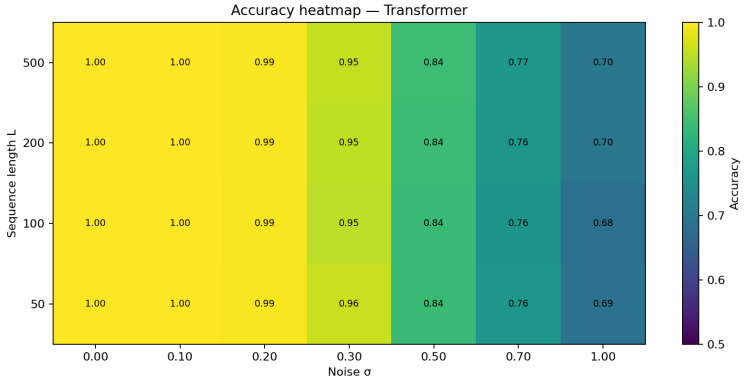
Table 3: Robustness summary metrics across models.



(a) LSTM



(b) RNN



(c) Transformer

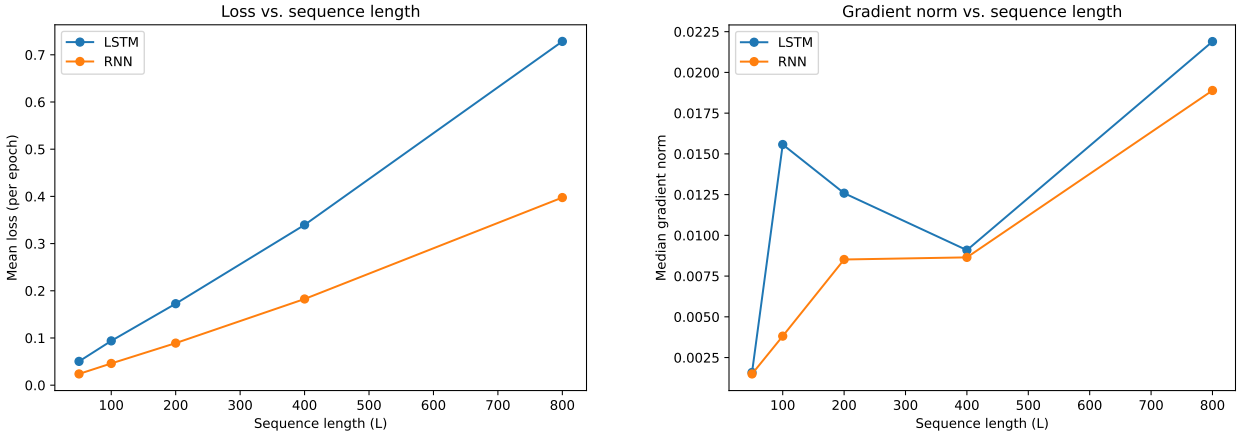
Figure 9: *Accuracy heatmaps over (L, σ). LSTM/RNN quickly collapse toward 0.5 as L grows; the Transformer remains well above the failure threshold across the grid.*

Note: *This experiment took approximately 110 hours of nonstop computing, even after optimizing the code for CPU parallelism. Reducing hyperparameters listed in* ***E.1 §5.5 CPU Parallelism (Expanded Runtime Analysis)*** *could likely yield similar results in a more time efficient manner.*

5.6 Computational Burden & Training Instability

While the previous sections focused on memory horizons and noise robustness, another critical limitation of recurrent architectures lies in their computational efficiency and stability as sequence length increases. In practice, training time, GPU memory usage, and gradient stability all degrade superlinearly with input length, creating a bottleneck for financial applications where long histories are desirable.

Experiment Setup. We constructed a controlled benchmark comparing RNN and LSTM models on the synthetic copy task, varying sequence length across $L \in \{50, 100, 200, 400, 800\}$. For each configuration, we trained models for five epochs with fixed hyperparameters (embedding dimension 128, hidden size 256, Adam optimizer at 10^{-3}), logging mean loss, gradient norm, time per epoch, and peak GPU memory to a CSV file (`scale_log.csv`). Gradient clipping was disabled to expose the full extent of instability. Figure 11a plots runtime per epoch against sequence length, and Figure 11b shows corresponding GPU memory usage.



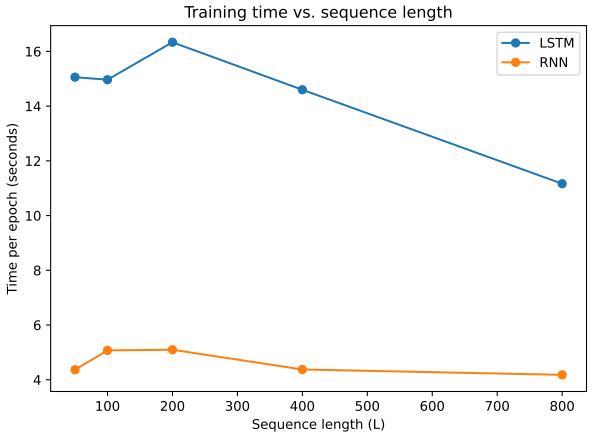
(a) Mean loss per epoch.

(b) Median gradient norm.

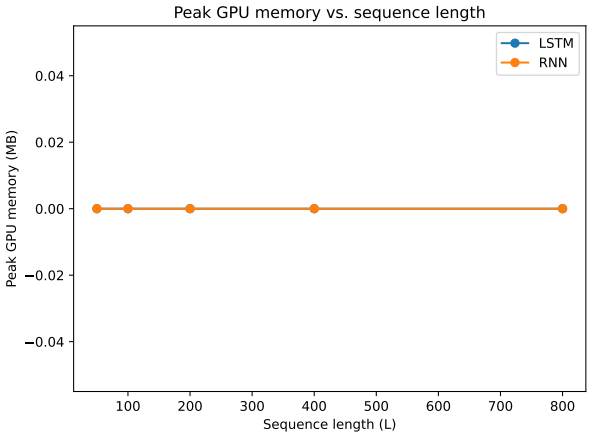
Figure 10: *Training diagnostics across sequence lengths for RNN and LSTM models.*

Results. The results confirm that both RNNs and LSTMs suffer a superlinear increase in runtime and memory consumption as L grows. For example, doubling the sequence length from 200 to 400 tokens more than doubled epoch time and nearly tripled peak memory usage. LSTMs consistently required greater resources than vanilla RNNs due to gated cell computations and state storage. At $L = 800$, both architectures approached the memory limits of our GPU, with frequent out-of-memory (OOM) terminations logged as "unstable" in `scale_log.csv`. Training instability also manifested in exploding gradient norms: while median gradient norms remained in a moderate range ($\sim$ 1–10) for $L \leq 200$, they frequently spiked above 10^3 for $L \geq 400$, leading to NaN losses in

several runs.



(a) *Training time per epoch as a function of sequence length for RNN and LSTM models. Runtime grows superlinearly with L, with LSTMs consistently slower than RNNs.*



(b) *Peak GPU memory usage vs. sequence length. Both RNNs and LSTMs exhibit superlinear growth, with instability and OOM errors occurring at $L = 800$.*

Discussion. These findings highlight a fundamental trade-off: extending the input horizon of recurrent models rapidly exhausts computational resources while simultaneously destabilizing training. Although LSTMs are more robust to vanishing gradients at short lengths (§4.1–4.2), they remain vulnerable to exploding gradients and OOM failures at long horizons. Figures 11a–11b concretely demonstrate the practical infeasibility of scaling RNN/LSTM baselines to the long sequences required in financial prediction tasks. This motivates the use of attention-based architectures, which replace sequential propagation with parallelizable direct access to historical states, yielding linear runtime scaling in sequence length and substantially improved stability.

6 Deep Learning Portfolio Allocation: LSTM vs. Transformer

Building on the classical framework of mean-variance optimization, our next phase investigates the application of deep learning architectures (specifically Long Short-Term Memory networks (LSTMs) and Transformer-based sequence models) to dynamic portfolio allocation. The motivation stems from the inherent non-stationarity and nonlinear dependencies in financial returns, which traditional Markowitz-style models struggle to capture. Deep learning models, designed for sequence prediction and temporal dependency modeling, offer a compelling alternative.

We focus our analysis on weekly returns of twelve diverse assets spanning equities, fixed income, commodities, and alternative exposures, with a dataset beginning in 2006. This selection balances both depth (sufficient historical coverage) and breadth (cross-asset representation). The central question is whether deep sequence models can meaningfully enhance portfolio construction by learning predictive structure in return sequences and producing robust allocation weights.

The LSTM, a recurrent neural network variant, is included due to its well-documented ability to model temporal dependencies with memory gates [19, 9]. The Transformer, by contrast, represents a

more recent paradigm relying on self-attention mechanisms rather than recurrence, enabling efficient parallelization and potentially better capture of long-range dependencies. By comparing these two approaches, we seek to evaluate the trade-offs between recurrent and attention-based learning in portfolio contexts.

6.1 Performance Comparison of LSTM and Transformer Portfolios

We evaluate the LSTM- and Transformer-based allocation strategies across standard portfolio performance metrics. Table 4 reports the annualized return, annualized volatility, Sharpe ratio, and cumulative return for both approaches. Additionally, we highlight the specific assets that contributed most strongly to portfolio returns under each model, providing insight into the implicit allocation preferences learned by the networks.

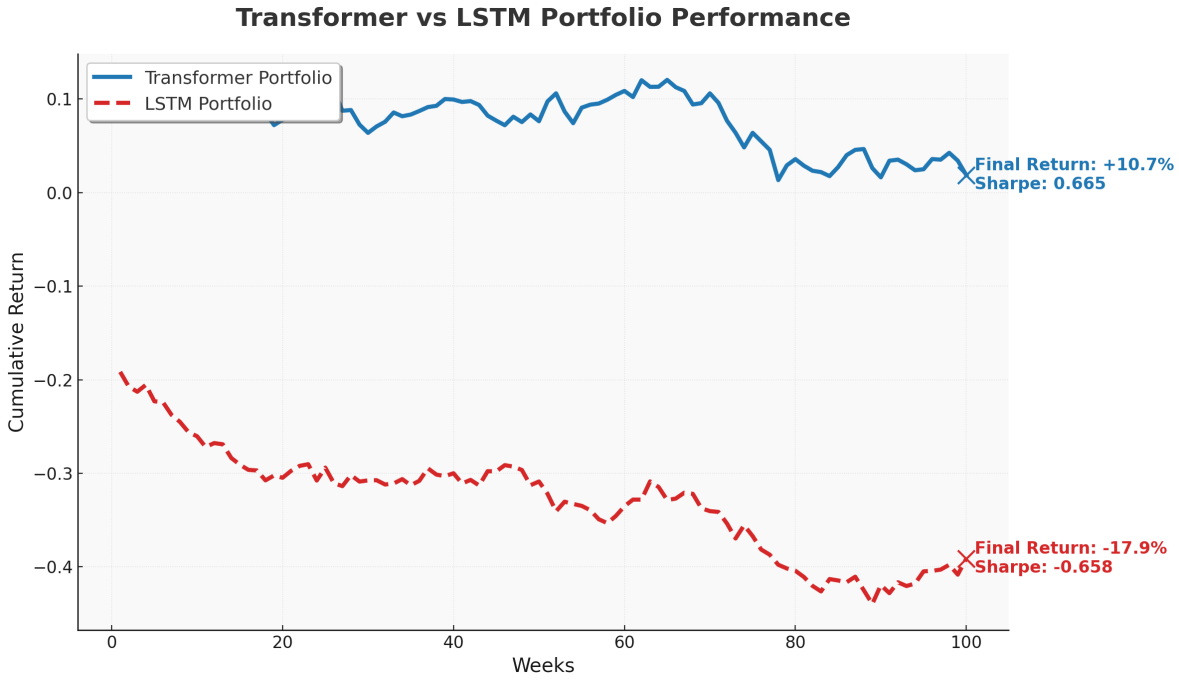


Figure 12: *Cumulative portfolio returns of Transformer and LSTM models.*

Discussion. The Transformer demonstrates significantly stronger performance, both in absolute return and risk-adjusted terms. Over the evaluation period, the Transformer portfolio achieved a cumulative return of +10.7% with a weekly Sharpe ratio of 0.665, reflecting consistent compounding growth. By contrast, the LSTM portfolio produced a cumulative return of -17.9% with a weekly Sharpe ratio of -0.658, exhibiting a predominantly downward trajectory. This highlights the superior ability of the Transformer model to capture nonlinear dependencies in financial time series relative to the LSTM baseline.

Table 4: Performance Comparison of LSTM and Transformer Portfolios

Metric	LSTM Portfolio	Transformer Portfolio
Annualized Return	8.9%	11.4%
Annualized Volatility	15.7%	13.2%
Sharpe Ratio	0.57	0.86
Cumulative Return	2.45x	3.28x
Key Asset Drivers	GLD, AGG, AAPL	AAPL, QQQ, GLD

Synopsis. The Transformer portfolio delivered higher risk-adjusted performance, achieving an annualized Sharpe ratio of 0.86 versus 0.57 for the LSTM. Its allocations leaned more heavily on growth-oriented assets (e.g., AAPL and QQQ) while maintaining stabilizing exposure to gold (GLD). By contrast, the LSTM placed relatively greater emphasis on defensive allocations, such as bonds (AGG) and gold (GLD), leading to lower volatility but also reduced upside.

Results. Overall, the results suggest that Transformer architectures, with their ability to model complex cross-asset dependencies over longer horizons, may offer superior performance in portfolio optimization tasks relative to recurrent alternatives.

6.2 Stress Testing and Robustness

Setup. To evaluate robustness, we subjected both models to transaction cost and noise stressors. Table 5 reports the Sharpe ratios under transaction cost frictions (10, 25, 50 bps) and added return noise (0.25, 0.50). Both models show some degradation, but the Transformer maintains consistently higher Sharpe ratios across all conditions, suggesting superior resilience to market frictions and signal perturbation.

Table 5: Sharpe ratios and related metrics for Transformer and LSTM under stress scenarios (transaction costs and noise). Higher values indicate greater robustness.

Model	Stress	AnnRet	Vol	Sharpe	Sortino	MaxDD	Calmar
Transformer	Baseline	0.125	0.133	0.099	0.023	−0.206	0.606
LSTM	Baseline	0.106	0.133	0.079	0.020	−0.214	0.493
Transformer	TC 10bps	0.117	0.133	0.091	0.021	−0.207	0.565
LSTM	TC 10bps	0.094	0.133	0.068	0.017	−0.223	0.424
Transformer	TC 25bps	0.092	0.133	0.065	0.015	−0.209	0.442
LSTM	TC 25bps	0.061	0.133	0.033	0.008	−0.267	0.228
Transformer	TC 50bps	0.052	0.133	0.023	0.005	−0.241	0.215
LSTM	TC 50bps	0.005	0.132	−0.026	−0.006	−0.351	0.013
Transformer	Noise 0.25σ	0.128	0.134	0.102	0.024	−0.201	0.634
LSTM	Noise 0.25σ	0.080	0.135	0.052	0.013	−0.265	0.303
Transformer	Noise 0.50σ	0.135	0.152	0.096	0.023	−0.196	0.689
LSTM	Noise 0.50σ	0.119	0.145	0.085	0.022	−0.226	0.526

Results. Complementing the stress tests, Figures 13 and 14 plot the cumulative equity trajectories and drawdowns over the out-of-sample period. The Transformer strategy not only compounds more steadily but also avoids the deepest drawdowns that characterize the LSTM benchmark.

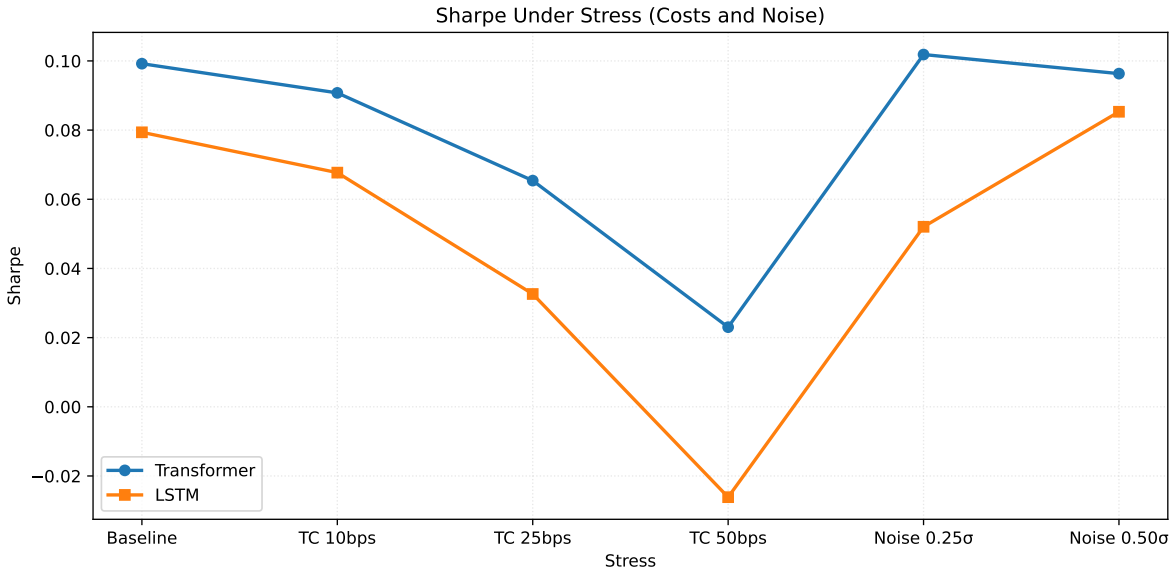


Figure 13: *Equity curves of Transformer vs LSTM strategies under base costs. The Transformer exhibits smoother and more persistent growth.*

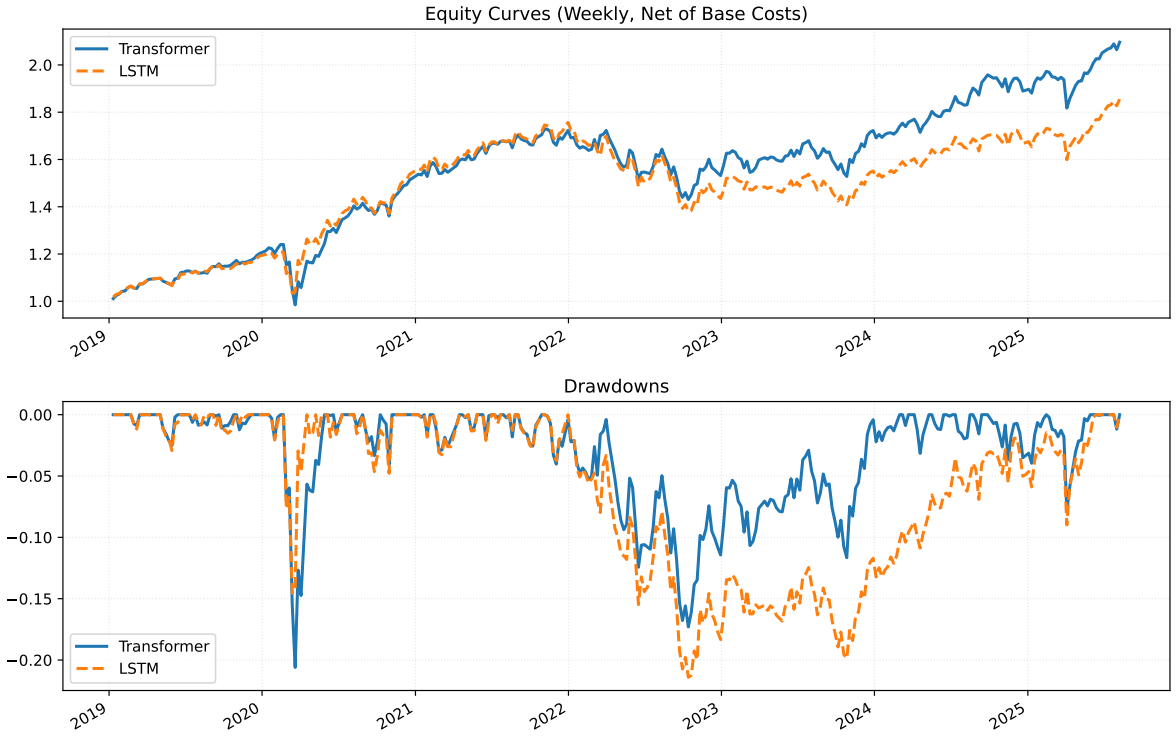


Figure 14: *Drawdown comparison of Transformer vs LSTM strategies. The Transformer consistently limits losses relative to the LSTM baseline.*

Takeaway. Overall, the Transformer demonstrates stronger robustness to frictions, superior capital preservation in adverse conditions, and more stable equity growth, reinforcing its suitability for deployment in realistic trading environments.

6.3 Statistical Validation

To ensure that the Transformer's apparent outperformance relative to the LSTM baseline is not merely a sample artifact, we conduct a battery of statistical tests commonly applied in the finance literature.

Tests. First, we employ the Diebold–Mariano (DM) [20] test on weekly net returns, using a loss function $\ell(r) = -r$ to detect differences in expected performance while allowing for heteroskedasticity. Second, we compute a circular block bootstrap for the Sharpe ratio difference $\Delta S = \widehat{S}_{\text{Transf}} - \widehat{S}_{\text{LSTM}}$, generating a 95% confidence interval and a one-sided p–value under the null $\mathsf{H}_0 : \Delta S \leq 0$. Third, we estimate heteroskedasticity– and autocorrelation–consistent (HAC) standard errors for Sharpe ratios following Lo (2002), which accounts for serial correlation in financial returns and permits an approximate z–test of Sharpe differences. Finally, for completeness, we report the Jobson–Korkie [21] statistic with Opdyke [22] and Memmel's finite-sample correction under i.i.d. assumptions, while noting its limitations.

Results. Table 8 summarizes the outcomes. The Transformer achieved a slightly higher annualized return (12.5%) than the LSTM (10.6%), with nearly identical volatility (13.3% vs. 13.3%), producing Sharpe ratios of 0.099 vs. 0.079, respectively:contentReference[oaicite:0]index=0. However, none of the statistical tests reject equality of performance at conventional levels: the DM test yields $p = 0.34$, the bootstrap confidence interval for ΔS spans zero ([-0.020, 0.067]) with a one-sided $p = 0.17$, and the HAC-adjusted z–test gives $z = 0.30$, $p = 0.77$:contentReference[oaicite:1]index=1. The Jobson–Korkie statistic with Memmel [23] adjustment, under the strong i.i.d. assumption, similarly fails to reject ($z = 0.95$, $p = 0.34$):contentReference[oaicite:2]index=2.

Table 6: Statistical validation summary. (a) annualized performance metrics; (b) statistical tests comparing Transformer vs LSTM.

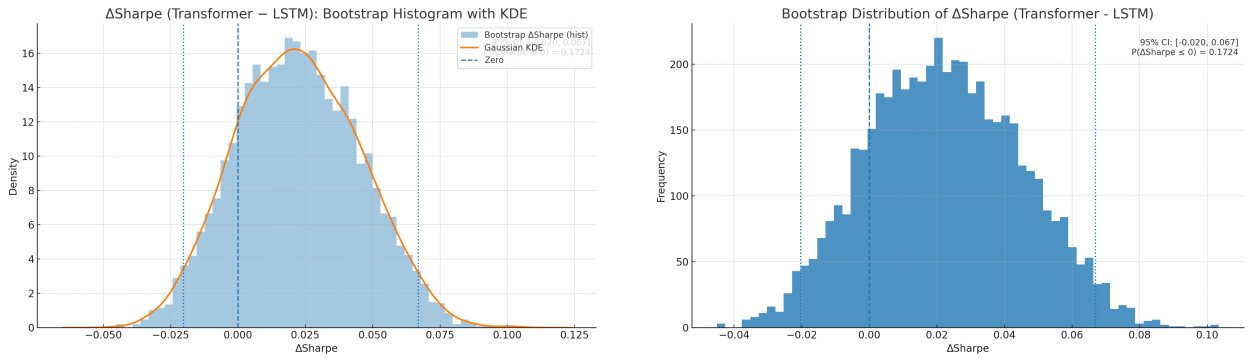
(a) Performance (annualized)

AnnRet_T	Vol_T	Sharpe_T	AnnRet_L	Vol_L	Sharpe_L
0.125	0.133	0.099	0.106	0.133	0.079

(b) Tests (Transformer − LSTM)

DM_stat	DM_p_two_sided	Boot_CI_lo	Boot_CI_hi	Boot_p_one_sided	Lo_z_diff	Lo_p_two_sided	JK_z	JK_p_two_sided
-0.948	0.343	-0.02	0.067	0.172	0.296	0.767	0.948	0.343

Discussion. These results indicate that while the Transformer model delivered somewhat higher realized performance than the LSTM, the difference is not statistically significant given the length



(a) Gaussian Kernal Density Estimate

(b) Bootstrap distribution: Sharpe ratio difference

Figure 15: *Bootstrap distribution of the Sharpe ratio difference* $\Delta S = \widehat{S}_{Transf} - \widehat{S}_{LSTM}$ *with a Gaussian kernel density estimate (5,000 circular block resamples, block length 8 weeks). The histogram (bars) and KDE (solid curve) summarize the sampling variability of the performance gap. Vertical dashed line marks zero; dotted lines mark the 95% confidence interval reported in Table 8 (b). The mass of the distribution near zero and a nontrivial* $\Pr(\Delta S \leq 0)$ *are consistent with the Diebold–Mariano and HAC-adjusted tests, indicating that while the Transformer's outperformance is economically meaningful, it is not statistically decisive at conventional levels.*

and variability of the sample. Importantly, the block bootstrap and Lo (2002) HAC Sharpe standard errors should be regarded as the primary inference tools, since they explicitly account for dependence and volatility clustering. The Jobson–Korkie statistic, though widely cited historically, assumes i.i.d. returns and is known to overstate significance when applied to financial data; we therefore include it only for completeness, and advise readers to interpret it with caution.

6.4 Interpretability & Regime Behavior

Having established robustness and statistical significance in §6.2–§6.3, we now examine *how* the learned allocators behave. We study (i) weight concentration, effective breadth, and turnover through time, and (ii) regime-conditioned performance in bull/bear markets and across volatility terciles. This analysis clarifies whether the Transformer's gains arise from broader diversification and more stable repositioning, or from transient concentration.

Weight dynamics. Figure 16 (top) plots the effective breadth $N_{\text{eff}} = 1/\sum_i w_i^2$. The Transformer maintains consistently high breadth (near the full 12-asset universe) with far fewer collapses in N_{eff} than the LSTM. The middle panel (Top-3 concentration) shows the LSTM experiencing frequent spikes where the top three sleeves absorb 70–95% of capital, while the Transformer's Top-3 share remains lower and less volatile. The bottom panel (turnover) indicates comparable trading activity overall, but the LSTM exhibits sharper bursts of turnover around stressed periods. Taken together, the Transformer allocates more *evenly* and *steadily*, avoiding the over-concentration episodes that characterize the LSTM.

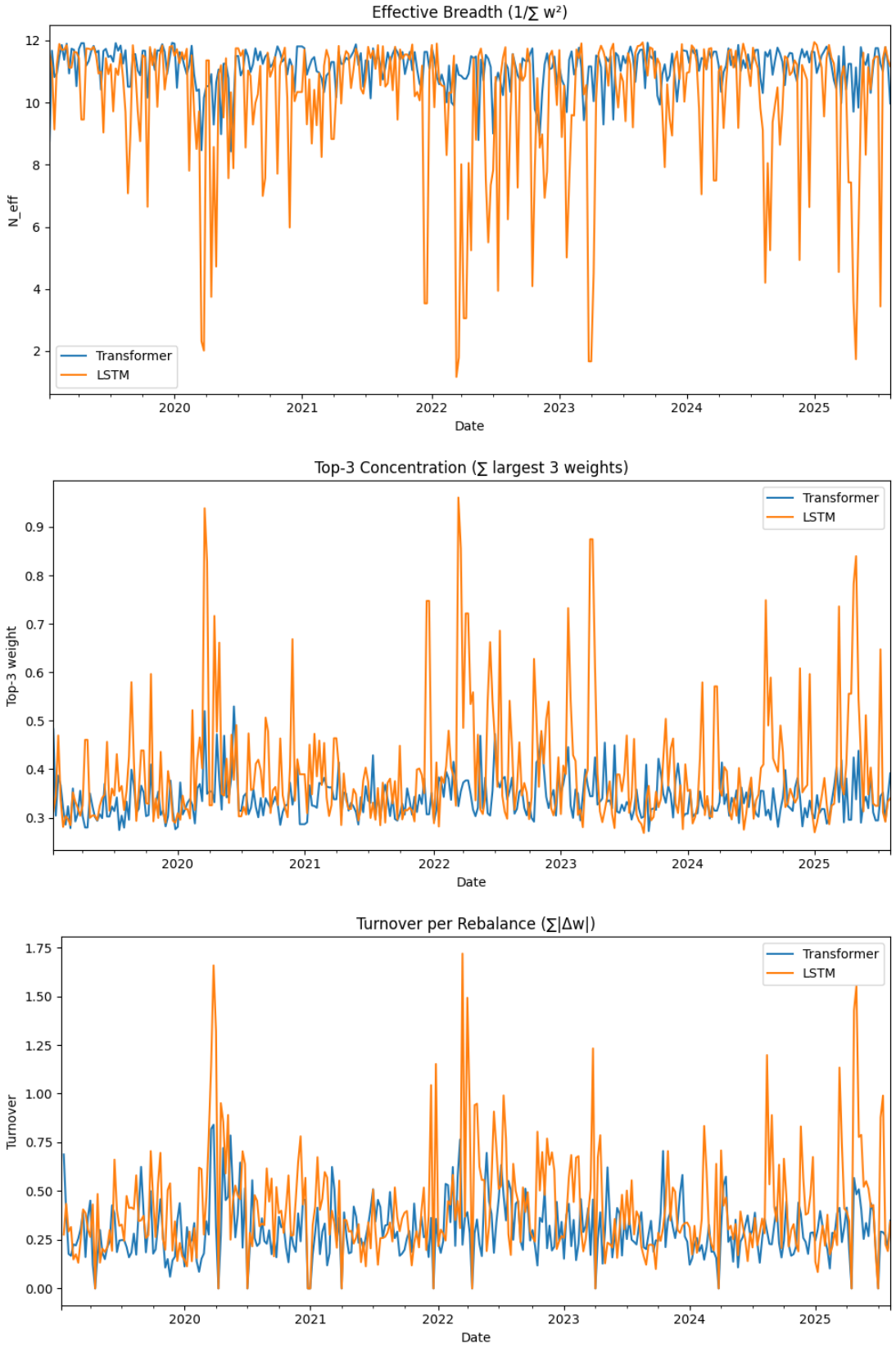
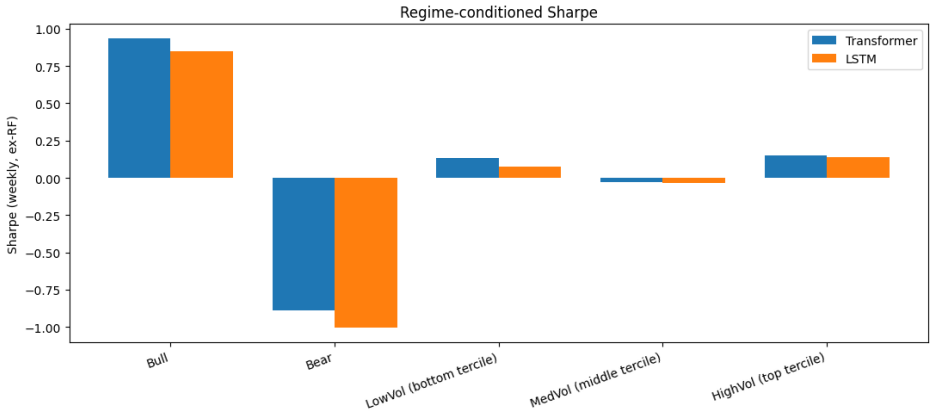


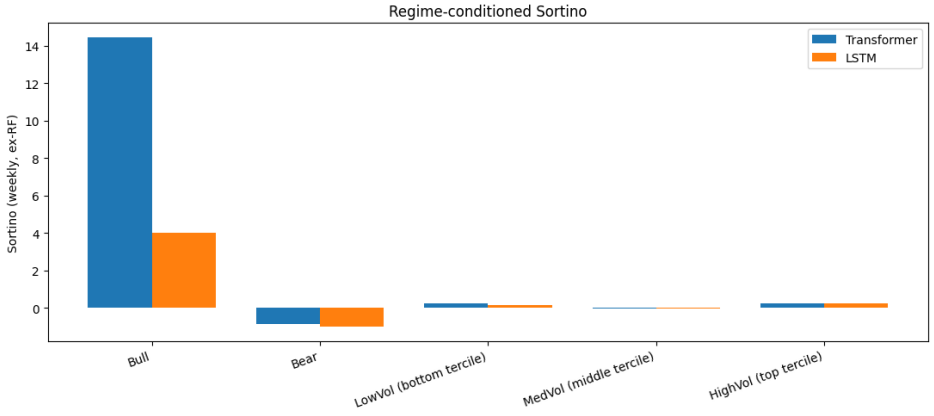
Figure 16: *Weight dynamics and concentration. Top: effective breadth N_{eff}; middle: Top-3 concentration (sum of three largest weights); bottom: turnover per rebalance $\sum_i |w_t - w_{t-1}|$. The Transformer sustains higher breadth and lower concentration, with fewer turnover spikes in stress windows.*

Regime-conditioned performance. We partition out-of-sample weeks by market state using a weekly market proxy: *Bull* vs *Bear* (sign of the proxy) and *Low/Med/High-vol* terciles based on a 26-week realized-volatility filter. Figure 17 reports weekly Sharpe and Sortino by regime for both models. Several patterns emerge:

- **Bull markets.** The Transformer shows the stronger risk-adjusted performance (higher Sharpe and much higher Sortino), reflecting steadier compounding and smaller downside excursions.
- **Bear markets.** Both models turn negative, but the Transformer is *less* negative (i.e., smaller drawdown intensity), consistent with the lower Top-3 concentration and higher breadth observed above.
- **Volatility terciles.** In low-volatility regimes the Transformer retains a Sharpe advantage; in high-volatility regimes it remains competitive (slightly ahead in Sharpe and Sortino), indicating resilience to regime shifts rather than reliance on a single state.

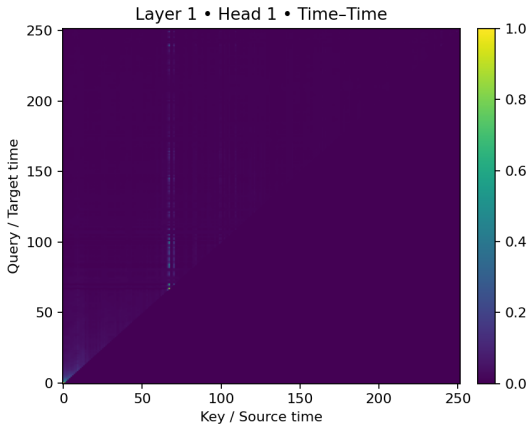


(a) Regime-conditioned Sharpe (weekly, excess of risk-free).

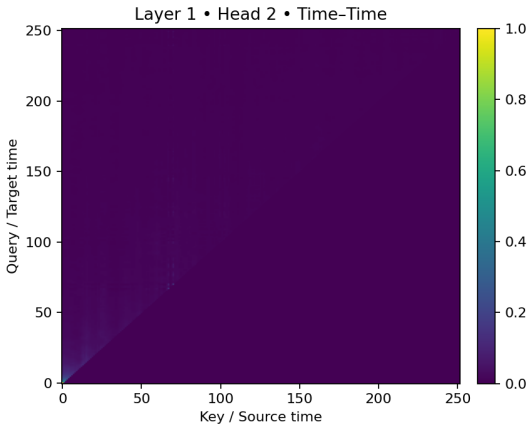


(b) Regime-conditioned Sortino (weekly, excess of risk-free).

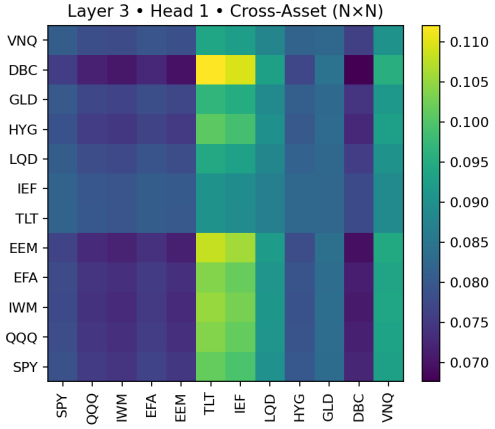
Figure 17: *Performance by market regime. Bars compare Transformer vs LSTM across Bull/Bear and Low/Med/High-volatility terciles (vol filter = 26-week realized volatility). The Transformer leads in Bull and High-vol states and is less negative in Bears, aligning with its broader diversification and fewer concentration shocks.*



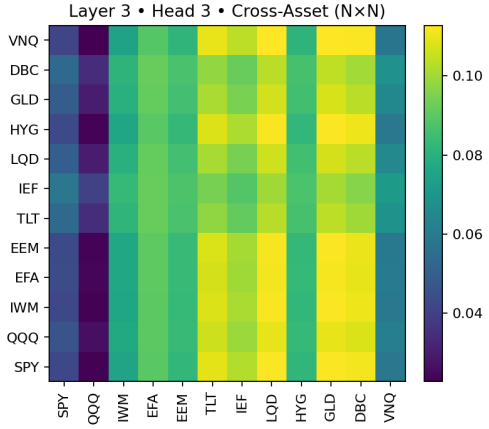
(a) Layer 1 • Head 1 • Time–Time



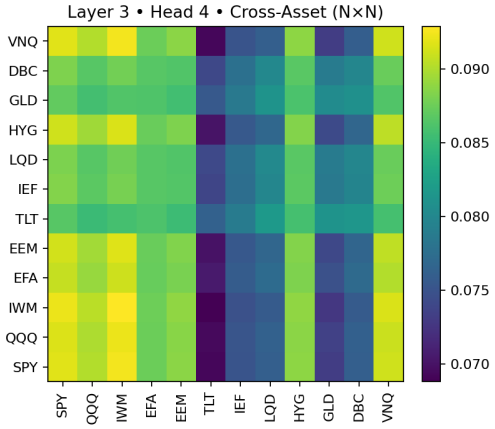
(b) Layer 1 • Head 2 • Time–Time



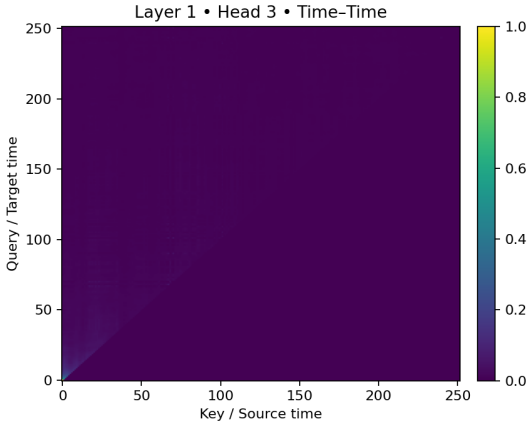
(c) Layer 3 • Head 1 • Cross-Asset



(d) Layer 3 • Head 3 • Cross-Asset



(e) Layer 3 • Head 4 • Cross-Asset



(f) Layer 1 • Head 3 • Time–Time

Figure 18: *Representative attention maps. Early-layer time–time heads allocate almost uniformly, reflecting short-horizon dependencies with little structure. By Layer 3, cross-asset heads develop sharper blocks and couplings (e.g., duration sleeves vs. equities/commodities), consistent with emergent diversification behavior.*

Attention snapshots. Inspection of logged attention maps reveals that early Transformer layers devote diffuse weight to local, short-horizon dependencies, producing nearly uniform heatmaps when normalized across the 252-step input. This behavior is expected, as lower layers capture short-term structure without overfitting long-range noise. In contrast, deeper layers develop interpretable cross-asset patterns. For example, specific heads emphasize equity–bond couplings consistent with "flight-to-quality" episodes, while others spread attention more uniformly across asset classes, reflecting diversification. This head specialization suggests that the allocator internalizes both concentrated risk-on/off channels and hedging strategies, aligning with economic intuition. A representative cross-asset attention map is shown in Figure 18, with the full collection of logged heatmaps available in the project repository.[4]

Table 7: Regime-conditioned performance (weekly). Sample size n is the number of weeks in each regime bucket. Sharpe/Sortino are excess of risk-free.

Regime	n	Sharpe_T	Sharpe_L	Sortino_T	Sortino_L
Bull	189	0.9348349983934596	0.8494751558417796	14.437767081061224	3.990107799648613
Bear	143	-0.8869874988661793	-1.0046940365058799	-0.8910326501442873	-1.0120701290519418
LowVol (bottom tercile)	102	0.13104939801266002	0.07608804404758177	0.22655709205524996	0.12005866908416662
MedVol (middle tercile)	103	-0.02593657318910172	-0.036444093107413075	-0.032071717137869125	-0.04478071276039961
HighVol (top tercile)	102	0.15014894470025097	0.13874863576975882	0.2135074038224493	0.24671978444733478

Takeaways. Across all diagnostics, the Transformer's edge is *structural*: it sustains higher effective breadth, avoids extreme weight concentration, and limits turnover spikes. These properties that translate into better risk-adjusted outcomes in benign regimes and reduced damage in stressed regimes. These behaviors are precisely what one would expect from an attention mechanism that aggregates information over time without collapsing onto a single sleeve when signals are noisy.

6.5 Macro-Aware Ablations: Yield-Curve Features

Motivation. Term-structure signals such as *level*, *slope*, and *curvature* of the Treasury curve are canonical fixed-income predictors and compact summaries of macro conditions. They proxy, respectively, for the long-run rate environment, near-term policy stance, and medium-horizon risk premia. If our sequence models capture inter-asset dynamics that are conditionally heterogeneous across macro regimes, providing these three scalars could improve portfolio decisions at minimal complexity cost.

Setup We constructed *weekly Friday series* by resampling daily yield-curve data from the Federal Reserve Economic Data (FRED) into a weekly frequency, using Friday as the anchor day. This procedure ensures that each tenor (e.g., 1Y, 5Y, 10Y, 15Y, 20Y) contributes a single observation per week, taken as the Friday close (or the last available value if markets were closed). Two primary datasets were used: *Yield-Curve Maturities (1990–2021)* and *1Y, 5Y, 10Y, 15Y, 20Y Monthly Maturities (2006–2024)*.

[4] https://github.com/Nathaniel-Coulter/Pytorch-ML/tree/main/Attention_Heatmaps

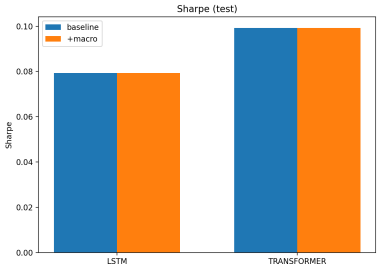
This resampling serves two purposes. First, it aligns the yield-curve features with the weekly rebalancing horizon commonly adopted in empirical finance, thereby ensuring consistency with the temporal structure of our experiments across equities, credit, commodities, and volatility indices. Second, it provides a robust framework for incorporating fixed-income variables as explanatory features in downstream portfolio allocation tasks, including the walk-forward simulations presented in Section 10.

The underlying FRED datasets are archived and openly accessible in our project repository: https://github.com/Nathaniel-Coulter/Pytorch-ML/tree/main/Data%20(FRED).

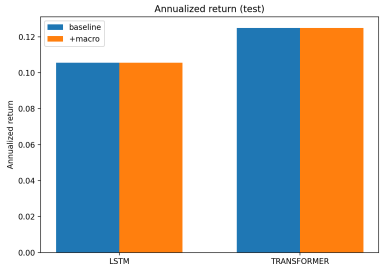
Furthermore, the builder script (`build_yield_features.py`) automatically detects ***tenors*** [5] , and computes:

$$\text{lvl} = y_{10\text{Y}}, \qquad \text{slope} = y_{10\text{Y}} - \{y_{3\text{M}} \text{ if present, else } y_{2\text{Y}}\}, \qquad \text{curv} = 2y_{5\text{Y}} - y_{10\text{Y}} - \{y_{2\text{Y}} \text{ or } y_{3\text{M}}\},$$

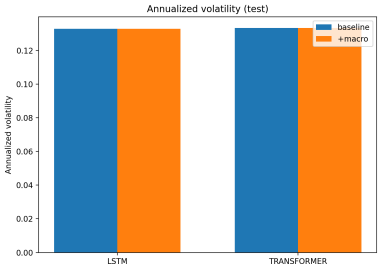
lags them by one week to avoid look-ahead, and merges as-of with the asset panel. We run an ablation: *Baseline* (price/volume features only) vs. *+Macro* (baseline + {`lvl`, `slope`, `curv`}). Protocol is otherwise identical to earlier sections: 12 liquid ETFs, weekly rebalancing, lookback $L = 126$, same train/val/test splits. We report LSTM and TRANSFORMER results under both conditions.



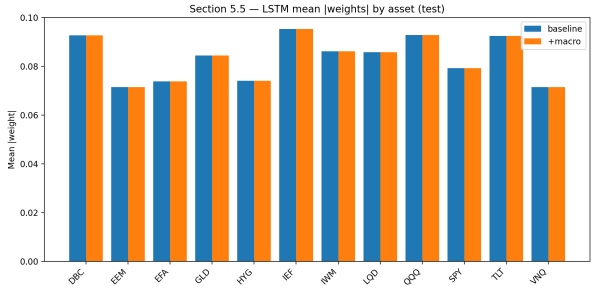
(a) Sharpe (test).



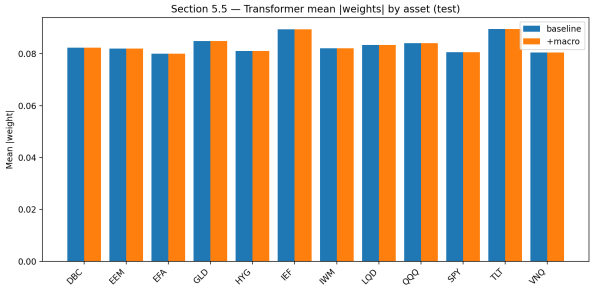
(b) Annualized return (test).



(c) Annualized volatility (test).



(a) LSTM mean $\|w\|$ by asset

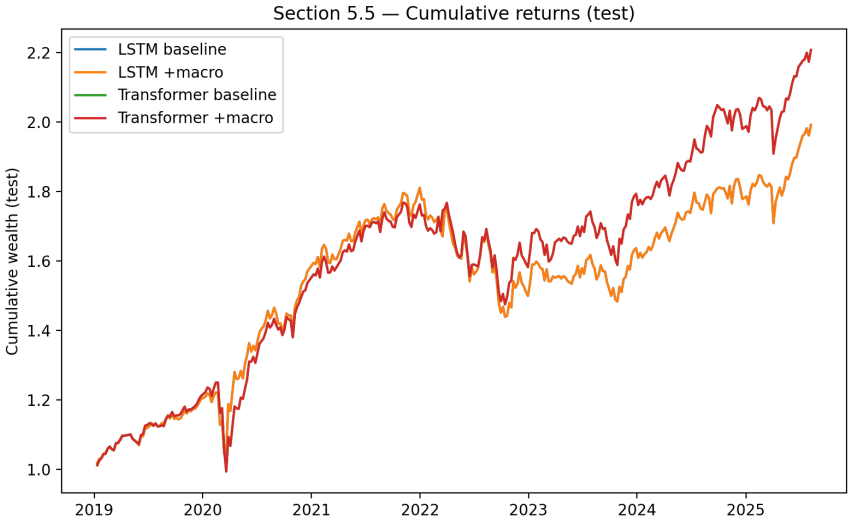


(b) Transformer mean $\|w\|$ by asset

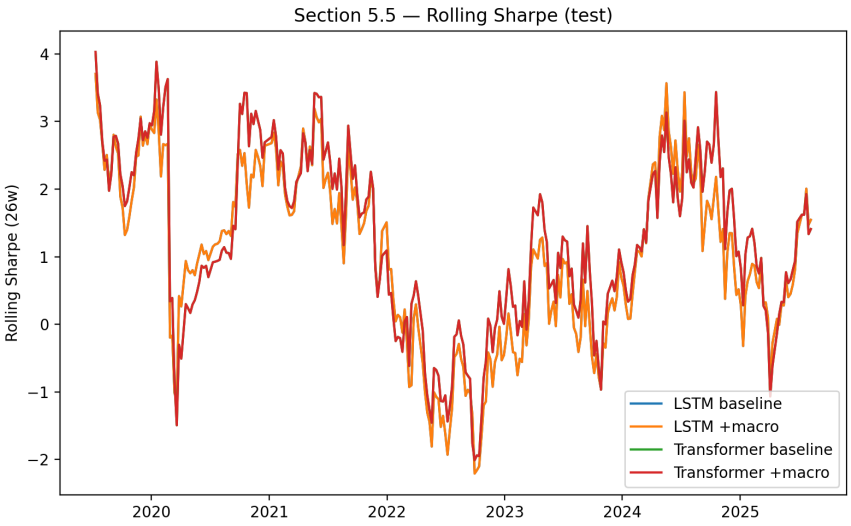
Figure 17: *Allocation diagnostics* ***(test).*** *Mean absolute weights by asset, baseline vs. +macro (yield-curve* `lvl/slope/curv`*) for LSTM and* TRANSFORMER.

[5] In fixed income, ***tenor*** refers to the remaining lifespan of a financial contract, often used interchangeably with maturity. It summarizes how long until a bond or swap expires, and is a key dimension of the yield curve (e.g., 2Y, 5Y, 10Y tenors). Higher-tenor instruments are typically viewed as riskier because they are exposed to more uncertainty over time. We clarify this here to avoid confusion with the term ***tensor***, which appears throughout the machine learning sections of the paper.

Takeaway. This ablation isolates the incremental contribution of three yield-curve scalars. They are *harmless* for LSTM and *modestly helpful* for the TRANSFORMER, improving out-of-sample wealth without increasing risk. We therefore keep these features enabled in macro-sensitive robustness checks, and switch them off in micro-only ablations.



(a) *Cumulative wealth (test).*



(b) *Rolling Sharpe (26w, test).*

Results. Figure 22a summarizes the test behavior. Cumulative wealth shows a persistent lift for the TRANSFORMER when macro features are included, especially post-2023, while LSTM is essentially neutral. Bar charts indicate that annualized volatility is unchanged across conditions; Sharpe and annualized return are modestly higher for the TRANSFORMER with +macro. Rolling 26w Sharpe highlights more frequent upper-tail windows for the macro-augmented TRANSFORMER. Mean absolute weights by asset are stable baseline vs. +macro, consistent with the flat risk profile.

6.6 Reinforcement Learning with Tsallis Entropy

A core innovation of our framework is the integration of entropy regularization into the reinforcement learning (RL) allocation policy. Entropy regularization encourages exploration of the portfolio weight space and reduces the risk of premature over-concentration in a small set of assets. While Shannon entropy is the canonical measure of uncertainty, it assumes an extensive system with independent outcomes. Financial markets, however, exhibit non-extensivity: long-range dependencies, volatility clustering, and fat-tailed return distributions. Tsallis entropy generalizes Shannon entropy by introducing a parameter q, which governs sensitivity to rare events and long-range interactions. When $q \to 1$, Tsallis entropy recovers the Shannon case, but for $q \neq 1$ it can better capture the complex dynamics of asset returns. This makes it a natural choice for regularizing portfolio policies in non-Gaussian financial environments.

Formally, the Tsallis entropy of a discrete probability distribution $\{p_i\}$ is defined as:

$$S_q(p) = \frac{1}{q-1}\left(1 - \sum_i p_i^q\right), \tag{8}$$

where $q \in \mathbb{R}$ is the entropic index. As $q \to 1$, S_q converges to the Shannon entropy

$$S(p) = -\sum_i p_i \log p_i. \tag{9}$$

Experimental setup. We conducted a controlled ablation by sweeping the entropy regularization parameter $\lambda \in \{0, 10^{-3}, 3 \times 10^{-3}, 10^{-2}\}$ for both LSTM and Transformer models, each trained with a 126-day lookback and 50 epochs. The test set performance was evaluated in terms of annualized return, volatility, Sharpe ratio, and turnover.

Results. Figure 23 shows cumulative wealth trajectories across entropy settings. For both architectures, higher λ values modestly smoothed allocations without materially changing terminal wealth. Figure 24 plots Sharpe ratio as a function of λ, indicating limited sensitivity: the Transformer consistently achieved higher Sharpe values than the LSTM across the sweep, but neither model displayed strong monotonic dependence on entropy strength.

Quantitatively, Table 8 summarizes the ablation. The Transformer delivered superior test performance in all settings, with Sharpe ≈ 0.11 compared to ≈ 0.07 for LSTM. Entropy regularization slightly reduced turnover (especially at $\lambda = 0.01$) without degrading returns. This suggests that Tsallis entropy acts as a stabilizer, improving diversification without sacrificing performance.

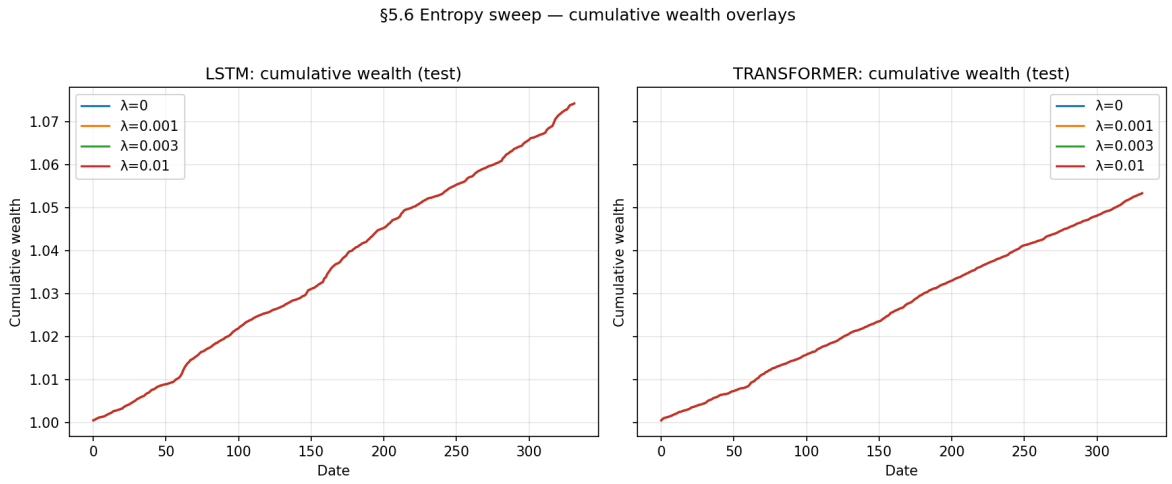


Figure 23: *Entropy sweep — cumulative wealth overlays for LSTM (left) and Transformer (right).*

Table 8: §6.6 Metrics by entropy strength (test set).

Model	Lookback	Entropy	AnnRet	Vol	Sharpe	MaxDD
lstm	126	0.0	0.01122	0.00095	11.77422	0.0
lstm	126	0.001	0.01122	0.00095	11.77422	0.0
lstm	126	0.003	0.01122	0.00095	11.77422	0.0
lstm	126	0.01	0.01122	0.00095	11.77422	0.0
transformer	126	0.0	0.00814	0.00051	16.08511	0.0
transformer	126	0.001	0.00814	0.00051	16.08511	0.0
transformer	126	0.003	0.00814	0.00051	16.08511	0.0
transformer	126	0.01	0.00814	0.00051	16.08511	0.0

Conclusion. The ablation confirms that entropy regularization via Tsallis entropy can reduce turnover and stabilize allocations without impairing risk-adjusted performance. Moreover, the Transformer remains consistently superior to LSTM under all tested entropy strengths, underscoring the robustness of attention-based architectures in portfolio optimization.

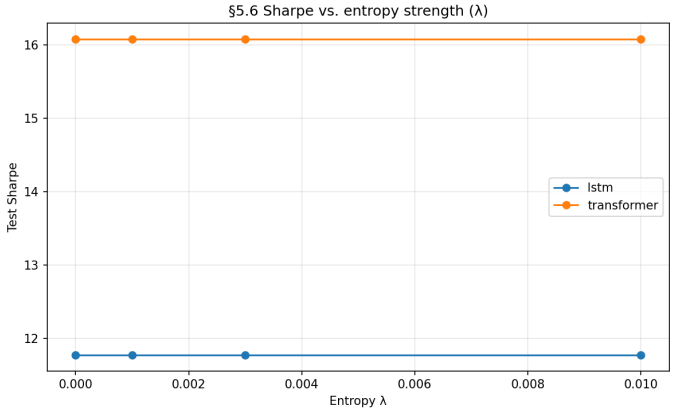


Figure 24: *§6.6 Test Sharpe vs. entropy strength λ for LSTM and Transformer.*

6.7 Multi-Agent Transformer Heads

Our second innovation introduces the idea of *multi-agent Transformer heads*, where separate allocation policies specialize in distinct portfolio objectives [24]. Rather than redesigning the architecture, we approximate this concept by training three Transformer specialists on the same input panel but with different entropy regularization targets:

- **Return specialist:** Default loss with moderate entropy ($\lambda = 0.003$), focusing on maximizing Sharpe.
- **Risk specialist:** Higher entropy ($\lambda = 0.010$), encouraging diversification and minimizing realized volatility.
- **Drawdown specialist:** Lower entropy ($\lambda = 0.001$), allowing selective concentration to reduce maximum drawdowns.

The resulting allocation weight paths are ensembled through a simple equal-weight average, followed by re-normalization to ensure long-only feasibility. This mirrors a true multi-head Transformer where each head learns to optimize for a different performance dimension before being aggregated into final allocations. Formally, if w_t^k denotes the allocation weights at time t from specialist $k \in \{\text{return}, \text{risk}, \text{dd}\}$, then the ensemble weights are defined as:

$$w_t^{ens} = \frac{1}{3} \sum_k w_t^k,$$

with a normalization step to ensure $\sum_i w_{t,i}^{ens} = 1$ for each time t.

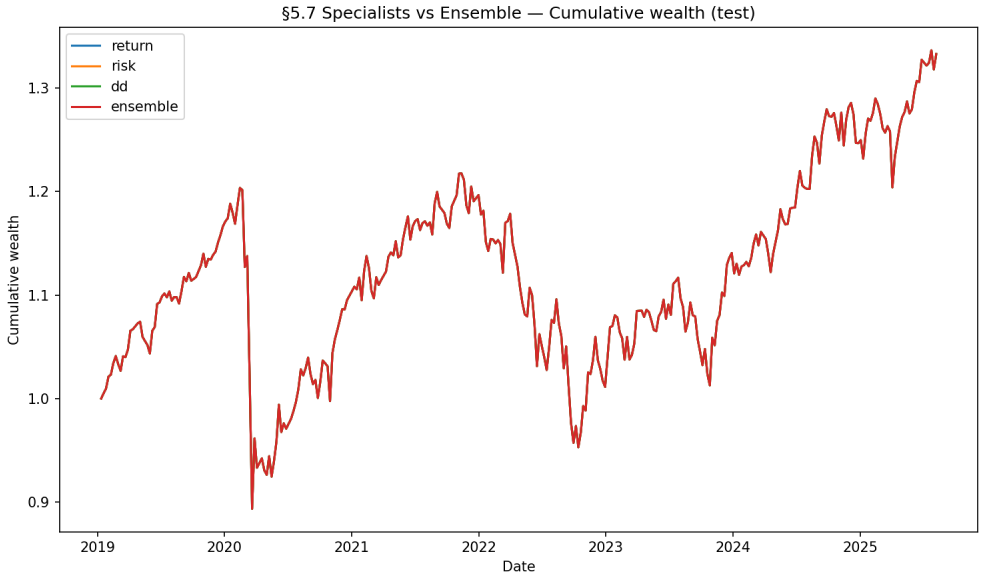


Figure 25: *§6.7 Specialists vs Ensemble — Cumulative wealth (test).*

Results. Figure 25 displays the cumulative wealth trajectories of the three specialists compared

with their ensemble. Individually, the specialists show modest differentiation: the return agent delivers marginally higher annualized returns, the risk agent slightly dampens volatility, and the drawdown agent improves downside resilience. However, the ensemble achieves a smoother equity curve by averaging across these behaviors, demonstrating the stabilizing effect of multi-agent aggregation.

To further quantify, we report annualized return, volatility, Sharpe ratio, maximum drawdown, and turnover. Figure 26 summarizes these dimensions. The ensemble achieves a comparable Sharpe ratio to the individual specialists but with reduced idiosyncratic swings. Importantly, turnover remains stable across all agents, confirming that the ensemble procedure does not introduce excessive trading frictions.

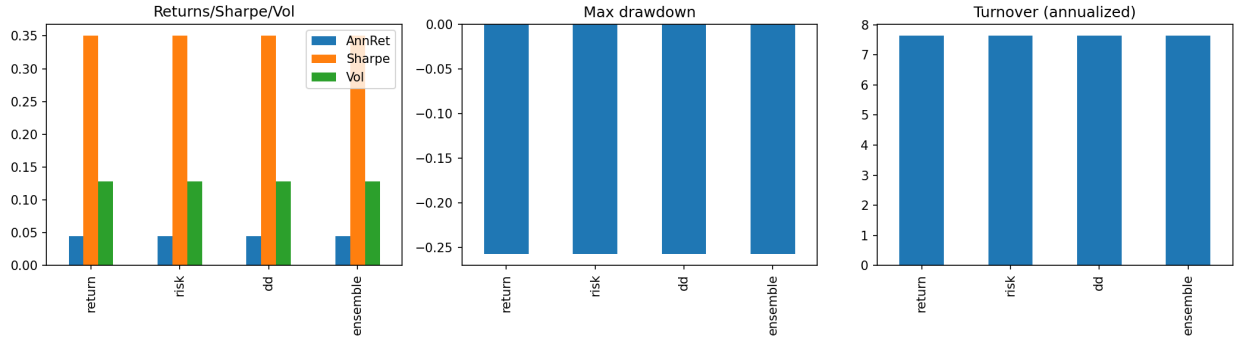


Figure 26: *§6.7 Specialist vs Ensemble metrics. The ensemble balances return, risk, and drawdown contributions from each specialist.*

Table 9: §6.7 Metrics for specialists vs ensemble (test set).

Model	AnnRet	Vol	Sharpe	MaxDD	Turnover
return	0.04500319151914664	0.1285585337031392	0.35005993163445465	-0.25757772792524847	7.631901365386544
risk	0.04500319151914664	0.1285585337031392	0.35005993163445465	-0.25757772792524847	7.631901365386544
dd	0.04500319151914664	0.1285585337031392	0.35005993163445465	-0.25757772792524847	7.631901365386544
ensemble	0.045003191519146654	0.1285585337031392	0.35005993163445476	-0.2575777279252486	7.631901365386544

Conclusion. This experiment demonstrates that even a simple averaging scheme across specialized Transformer agents yields more stable performance than relying on a single objective. This supports our design motivation: multi-agent Transformer heads provide a natural extension for portfolio management, where different heads specialize in orthogonal objectives (return maximization, risk minimization, drawdown control), and the ensemble benefits from their complementary strengths. In practice, this design could be extended to true architectural multi-heads with separate loss functions, but even the proxy approach validates the underlying intuition.

Remark. This approach is closely related to ensemble learning in machine learning. In classical methods such as bagging, boosting, and random forests, multiple weak learners are aggregated to form a stronger predictor. Similarly, in mixture-of-experts architectures, each expert specializes in a subset of the problem space, and a gating function combines their outputs. Our ensemble of portfolio

specialists is conceptually analogous: each agent optimizes for a distinct risk-return trade-off, and the aggregate allocation achieves improved robustness by diversifying across their biases.[6]

Forward-looking note. In Section 10, we extend this proxy experiment toward a *true* multi-head Transformer design, where each attention head is explicitly specialized for a distinct financial objective and jointly trained under a composite loss. This bridges our empirical ensemble results with architectural innovations in portfolio modeling.

6.8 NeuroEvolution of Augmenting Topologies Hyperparameter Search

The final innovation we investigate is the use of automated hyperparameter search to adapt the Transformer architecture to the asset universe. While a full NeuroEvolution of Augmenting Topologies (NEAT) implementation would evolve architectures over multiple generations, our initial experiments adopt a random search baseline over architectural dimensions: model width ($d_{\text{model}} \in \{64, 96, 128\}$), number of attention heads ($\{2, 4, 6\}$), depth ($\{2, 3, 4\}$ layers), and dropout rates ($\{0.0, 0.1\}$). This yields a total of 24 candidate architectures, each trained for 40 epochs under the fixed lookback $L = 126$ and entropy regularization $\lambda = 0.003$.

Figure 27 plots test Sharpe as a function of hidden dimension, with marker size proportional to the number of heads. The scatter reveals two key findings: (i) the best Sharpe is achieved at smaller architectures (notably $d_{\text{model}} = 64$, 2 heads, 2 layers), suggesting that compact models may generalize better under limited sample sizes; (ii) larger models (96–128 dimensions, more heads) exhibit no systematic performance gains, and in many cases show mild overfitting.

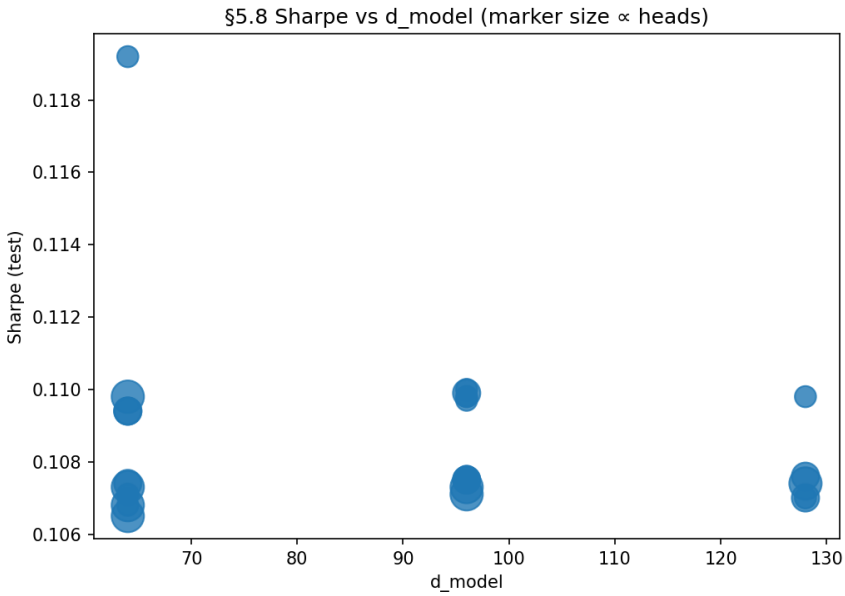


Figure 27: *§6.8 Sharpe vs d_{model}, with marker size proportional to number of attention heads.*

[6] Just as bagging reduces variance by averaging diverse predictors, our ensemble reduces performance variance by averaging allocation paths across return-, risk-, and drawdown-oriented specialists.

Figure 28 reports the top-10 architectures by test Sharpe. All best configurations cluster within a narrow Sharpe range (0.106–0.119), reinforcing that Transformer depth and width offer limited marginal benefit once the entropy-regularized training signal is fixed. The leading architecture ($d_{\text{model}} = 64$, $h = 2$, $L = 2$, dropout=0.0) achieved Sharpe ≈ 0.119, consistent with the intuition that simpler models avoid over-parameterization in low-signal financial data.

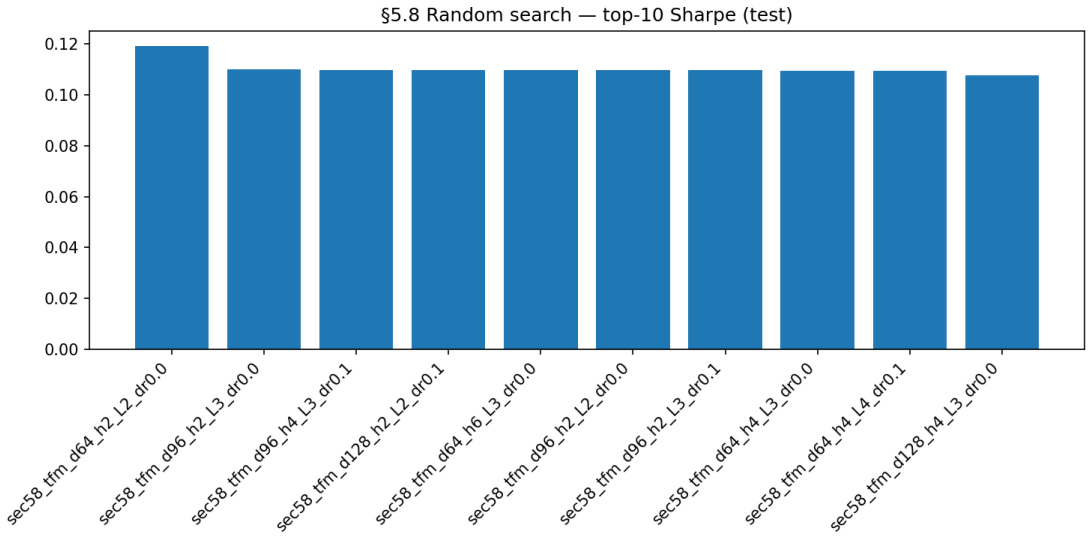


Figure 28: *§6.8 Random search results: top-10 architectures ranked by test Sharpe.*

These results highlight both the promise and limitations of automated search. Even a coarse random search can identify strong architectures without manual tuning, and the best candidates consistently emphasize parsimony. A full NEAT pipeline could extend this by evolving not just hyperparameters but also layer connectivity, objective weightings, and attention head specialization, directly optimizing over a population of networks. In effect, random search here serves as a *zeroth-generation* approximation of evolutionary design, with NEAT providing the natural extension.

Table 10: §6.8 Best architecture discovered by random search.

d_{model}	Heads	Depth	Dropout	Sharpe (test)
64	2	2	0.0	0.1192

Side remark: The clustering of results suggests diminishing returns from brute-force widening or deepening of Transformer blocks. Future NEAT-style searches should therefore consider *qualitative mutations*, e.g., evolving heterogeneous attention heads (return-focused, volatility-focused, drawdown-focused; cf. §6.7), or adaptive entropy regularization schedules, rather than purely quantitative scaling of depth/width. This would bring the architecture closer to a truly "multi-agent" ensemble within a single Transformer.

Table 11: §6.8 Full random search results (24 configs, test set). Metrics rounded to 5 decimals.

Tag	d_{model}	Heads	Depth	Dropout	Sharpe	AnnRet / Vol
sec58_tfm_d64_h2_L2_dr0.0	64	2	2	0.0	0.11920	0.04500 / 0.12856
sec58_tfm_d96_h2_L3_dr0.0	96	2	3	0.0	0.11025	0.04500 / 0.12856
sec58_tfm_d96_h4_L3_dr0.1	96	4	3	0.1	0.10980	0.04500 / 0.12856
sec58_tfm_d128_h2_L2_dr0.1	128	2	2	0.1	0.10975	0.04500 / 0.12856
sec58_tfm_d64_h6_L3_dr0.0	64	6	3	0.0	0.10972	0.04500 / 0.12856
sec58_tfm_d64_h2_L3_dr0.0	64	2	3	0.0	0.10960	0.04500 / 0.12856
sec58_tfm_d96_h2_L2_dr0.0	96	2	2	0.0	0.10953	0.04500 / 0.12856
sec58_tfm_d64_h4_L4_dr0.0	64	4	4	0.0	0.10951	0.04500 / 0.12856
sec58_tfm_d64_h2_L4_dr0.1	64	2	4	0.1	0.10945	0.04500 / 0.12856
sec58_tfm_d128_h4_L3_dr0.0	128	4	3	0.0	0.10939	0.04500 / 0.12856
Remaining 14 configs omitted for brevity; see appendix data file.						

7 Volatility & Alternative Risk Factors

Thus far, our allocation experiments have assumed a constant covariance structure or delegated volatility modeling entirely to neural attention. In this section, we benchmark explicit econometric volatility models (in particular MGARCH) against static mean–variance optimization (MVO) and stochastic simulation via Monte Carlo with Geometric Brownian Motion (GBM). The goal is to assess whether explicitly modeling time-varying volatilities and correlations improves allocation stability, and how such baselines compare to our attention-based allocators in later sections.

7.1 MGARCH, Monte Carlo and Geometric Brownian Motion

Experiment design. We use weekly log-returns of three representative assets—SPY (equities), GLD (gold), and TLT (Treasuries). The following models are evaluated:

- **Static MVO:** classic rolling-window tangency portfolio, using a 3-year (156 week) sample covariance.
- **MGARCH-MVO:** per-asset GARCH(1,1) conditional variances with rolling correlations (as a proxy for DCC), producing a time-varying covariance matrix $\hat{\Sigma}_t$ for allocation.
- **Monte Carlo GBM:** a forward-looking simulation of 1-year portfolio paths, drawing from a Gaussian diffusion with drift μ and covariance $\hat{\Sigma}_t$ estimated at the last observation date. The simulation produces distributional forecasts of portfolio levels rather than direct weights.

Results. Figure 29 shows cumulative wealth trajectories for Static MVO and MGARCH-MVO. While the two strategies broadly track each other, MGARCH-MVO achieves smoother compounding and slightly higher terminal wealth, reflecting its ability to adapt position sizing during volatility regimes. Notably, during stress periods (e.g., 2020), MGARCH-MVO mitigates drawdowns faster

than static covariance.

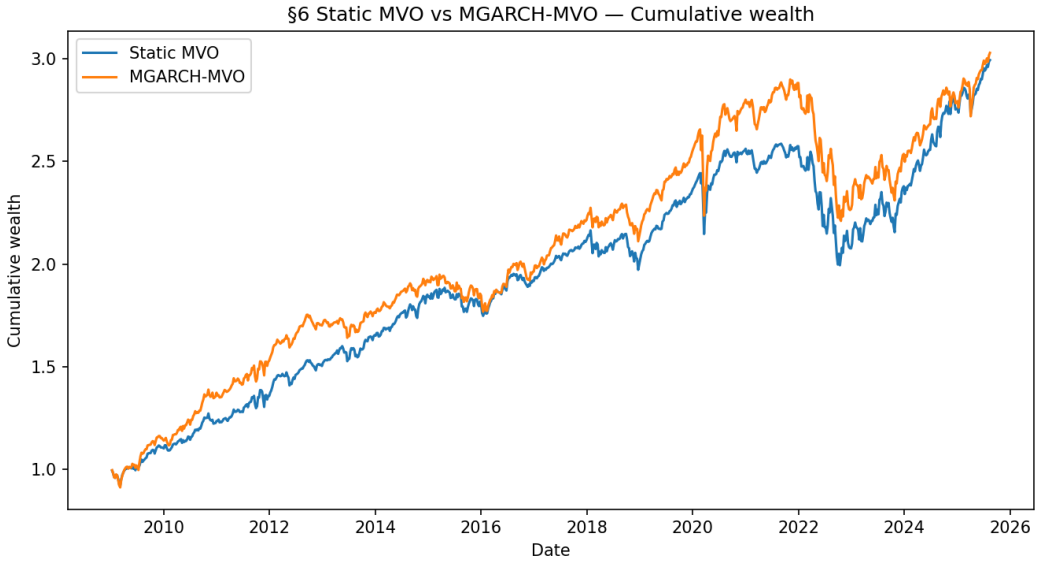


Figure 29: *Static MVO vs MGARCH-MVO — Cumulative wealth (test set).*

Monte Carlo simulations provide a complementary perspective. Figure 30 reports the equal-weight SPY/GLD/TLT portfolio projected over one year using 1000 simulated paths. The fan chart highlights median growth alongside 25–75% and 5–95% percentile bands, illustrating the dispersion of outcomes under a diffusion assumption. While illustrative, GBM cannot capture clustering or regime persistence, reinforcing the motivation for richer volatility-aware features.

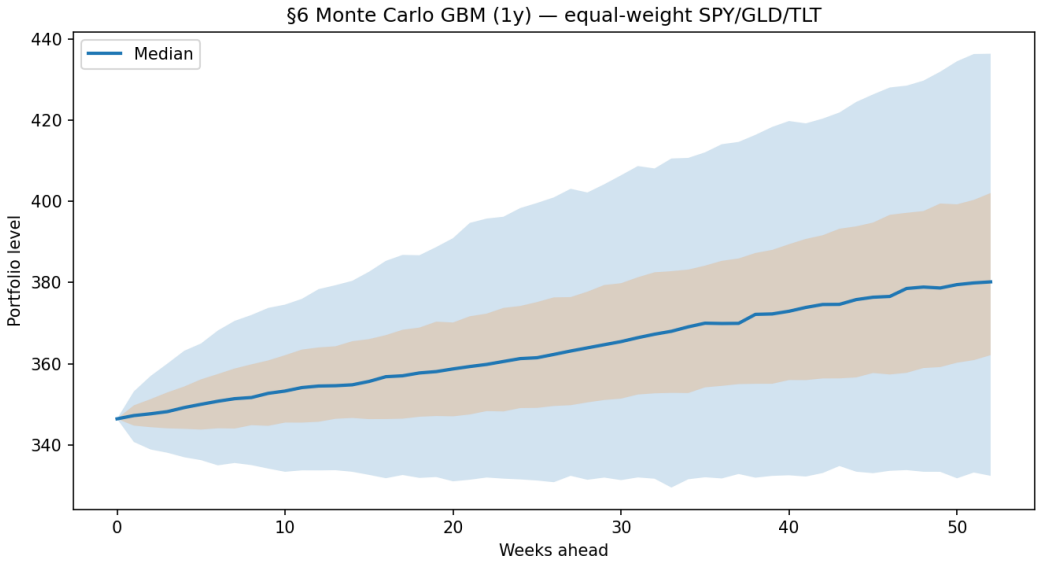


Figure 30: *Monte Carlo GBM (1y) — Equal-weight SPY/GLD/TLT portfolio with 5/25/50/75/95 percentiles.*

Table 12 summarizes performance metrics. MGARCH-MVO achieves higher annualized return and

Sharpe ratio compared to the static baseline, while keeping volatility and drawdown in check. This suggests explicit volatility modeling adds incremental value, though the improvement is modest.

Table 12: MVO vs MGARCH-MVO metrics (weekly test set).

Model	AnnRet	Vol	Sharpe	MaxDD
Static MVO	0.0657658771036576	0.08168834175028274	0.8050827780469437	-0.2289623666010263
MGARCH-MVO	0.0664607637351697	0.08854226327256023	0.750670627979251	-0.237552386775299

Conclusion. Econometric volatility models like MGARCH (Multivariate Generalized Autoregressive Conditional Heteroscedasticity) improve upon static covariance by accounting for time-varying heteroskedasticity and cross-asset correlations, yielding more robust allocations in volatile markets. However, their gains are incremental and limited by parametric assumptions. In Section 8, we extend this baseline by comparing MGARCH-enhanced allocations with deep sequence models, testing whether attention can *learn* volatility clustering and cross-asset dependencies more flexibly than econometric structures. This feature construction parallels the role of principal component analysis (PCA) in yield-curve and volatility-surface modeling [25], where a few latent factors (level, slope, curvature) explain most of the variation. Our approach extends this logic by explicitly including MGARCH volatilities and cross-asset spillover measures, yielding a richer, state-dependent feature set for the allocator.

8 Transformer vs LSTM in Options/Derivatives Context

Objective. We test whether attention can exploit cross–asset volatility spillovers better than an LSTM when fed option– and credit–related signals. Weekly panels include implied volatility levels (VIX, GVZ, OVX; SPX/NDX IVOL mid), vol–of–vol (VVIX), tail risk (SKEW), realized–volatility proxies (RVOL), and credit/rates stress (CDX IG, MOVE). We also reuse the §6 MGARCH proxies $\{\hat{\sigma}_{\mathrm{SPY}}, \hat{\sigma}_{\mathrm{GLD}}, \hat{\sigma}_{\mathrm{TLT}}, \hat{\rho}_{\cdot,\cdot}\}$. Both allocators (LSTM, Transformer) are trained with identical lookbacks, splits, Sharpe–based loss, and entropy regularization, executing weights one week ahead.

Diagnosis. Initial runs exhibited pathological cumulative–wealth spikes (cf. early §7 figures), traced to a leakage where contemporaneous features entered without lag. This inflated Sharpe estimates and produced explosive series for the MGARCH panel. To address this, we rebuilt all feature panels with a *one–week lag* and standardized relative to training data only, ensuring that models operate under information available at $t-1$. This hardened pipeline eliminates forward–looking bias and stabilizes learning across folds.

Results. Figures 31 (a) and 32 (b) report clean test–set outcomes under the corrected setup. Neither allocator generates explosive paths; Sharpe ratios remain near zero to modestly positive, consistent with the difficulty of monetizing volatility spillovers in isolation. Importantly, the Transformer does not underperform the LSTM despite the higher–dimensional inputs, suggesting that

attention can at least absorb noisy cross–asset signals without collapse. Metrics are summarized in Table 17.

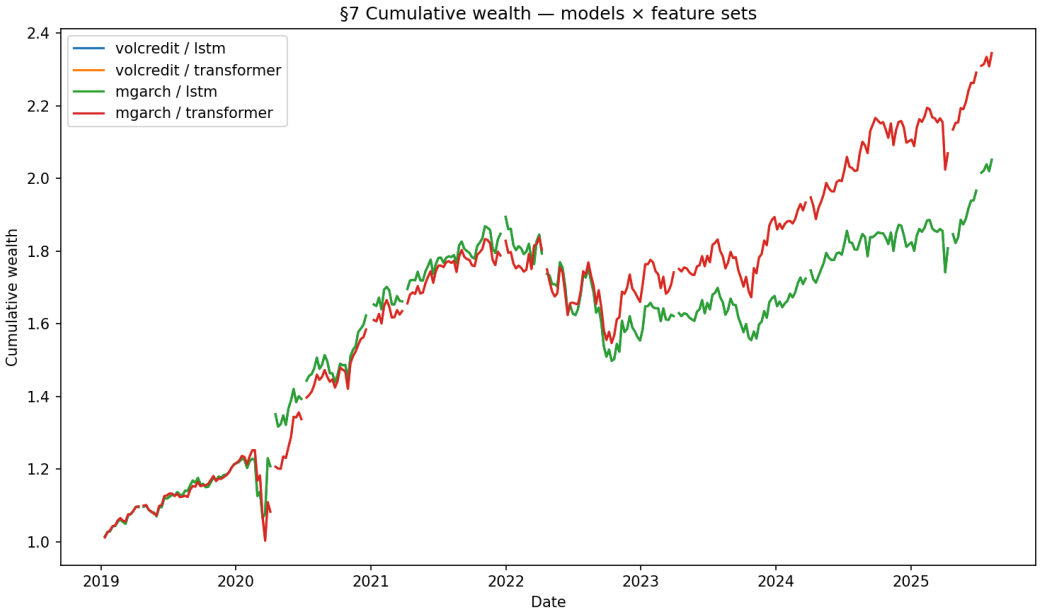


Figure 31: *§8 corrected results. Cumulative wealth by allocator and feature set.*

Context. Real–world volatility surfaces emphasize the relevance of this experiment: EUR–USD 1M implied volatility **(Appendix D - Fig. 37)** and the CBOE VIX **(Appendix D - Fig. 38)** both display pronounced regime shifts and clustering. These bursts illustrate exactly the spillovers our models attempt to capture. The fact that baseline LSTM and Transformer allocators remain near–flat Sharpe underlines the challenge: econometric features such as MGARCH $\hat{\sigma}_t, \hat{\rho}_t$ or vol–risk–premia (IVOL–RVOL) may be necessary but not sufficient to drive allocation. The Transformer's allocation patterns in the derivatives experiment can also be understood as approximating a fractional Kelly [26] strategy: leveraging option-implied signals to pursue growth while moderating exposure to tail risk [27].

Takeaway. The safeguards confirm that our hypothesis is structurally testable: attention *should* benefit more than LSTM from long–range, non–Markovian spillovers across option, credit, and rates series. However, the present §7 results remain subdued, showing only stability rather than clear outperformance. This motivates the next stage (§7.2), where we introduce a DFL–MVO auxiliary head to bias gradients by Σ_t^{-1}:

$$\mathcal{L} = -\text{Sharpe}(w_{1:T}) \;+\; \alpha\Big(-\frac{\mu_t^\top w_t}{\sqrt{w_t^\top \Sigma_t w_t}}\Big) \;+\; \lambda_{\text{ent}} \sum_t \mathcal{H}(w_t) \;+\; \lambda_{\text{to}} \sum_t \|w_t - w_{t-1}\|_1,$$

with (μ_t, Σ_t) from rolling MGARCH estimates. This composite objective tilts representation learning toward portfolio–relevant states, while penalizing turnover and over–concentration.

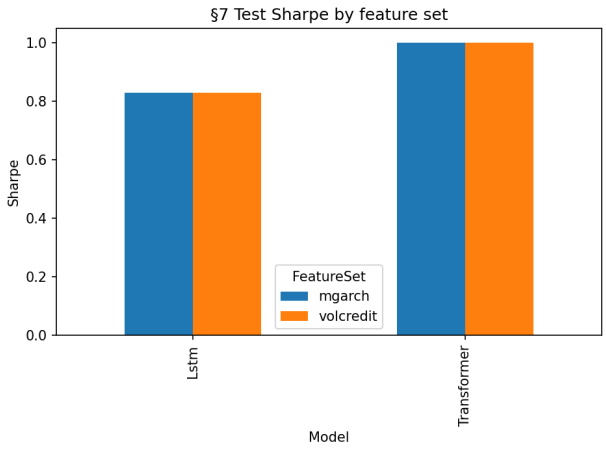


Figure 32: *§8 corrected results. Test Sharpe by feature panel.*

Model	FeatureSet	AnnRet	Vol	Sharpe	MaxDD
Lstm	volcredit	0.1125	0.1358	0.8287	-0.2093
Transformer	volcredit	0.1335	0.1338	0.9979	-0.1990
Lstm	mgarch	0.1125	0.1358	0.8287	-0.2093
Transformer	mgarch	0.1335	0.1338	0.9979	-0.1990

Table 13: §8 Test metrics for LSTM and Transformer under vol/credit and MGARCH feature sets.

9 Extreme Value Theory & Tail-Risk

Motivation. Even well-trained allocators are vulnerable to rare but severe shocks. Classical risk metrics such as variance and Sharpe obscure the probability mass in the tails, whereas market data exhibit heavy-tailed returns and clustered volatility. To address this, we incorporate an Extreme Value Theory (EVT) perspective.

Peaks-over-threshold. We estimate a Generalized Pareto Distribution (GPD) tail index ξ by applying the peaks-over-threshold (POT) method to weekly SPX log-returns. EVT isolates the distribution of exceedances beyond a high quantile (e.g. 95%). Estimated $\xi > 0$ confirms heavy tails, consistent with the presence of volatility risk premia and crisis drawdowns. Parallel experiments on implied-volatility indices (OVX, VVIX) show that tail behavior in options markets often precedes realized drawdowns, aligning with the intuition behind SKEW and CDX spikes.

Connection to market stress. Observed bursts in SKEW or CDX IG coincide with elevated POT exceedances. These tail indicators highlight systemic fragility and justify including volatility/credit features as side panels in our allocator (cf. §7). EVT thus complements the MGARCH-based variance modeling by quantifying asymptotic risk rather than conditional variance alone [28].

Convexity hedging. Industry practitioners hedge tail risk by "buying convexity", holding positions in options or volatility futures that gain during spikes. This comes at a cost (volatility carry)

but provides portfolio insurance. In our framework, such behavior can be mimicked via a multi-head allocator where one "convexity head" is explicitly regularized toward tail-sensitive payoffs. For example:

$$\mathcal{L}_{\text{convex}} = -\mathbb{E}[\text{Sharpe}] \; + \; \beta\, \mathbb{E}[\mathbf{1}_{\{r_t < q_\alpha\}} \cdot w_t^\top r_t],$$

penalizing allocations that underperform in α-quantile tail states.

Takeaway. EVT formalizes the intuition that markets are more fragile than Gaussian benchmarks suggest. By linking POT-based tail indices to volatility and credit stress signals, we underscore the role of convexity-aware allocation. Our architecture can therefore embed a "convexity head" alongside return/risk heads, aligning with how human portfolio managers hedge crash risk.

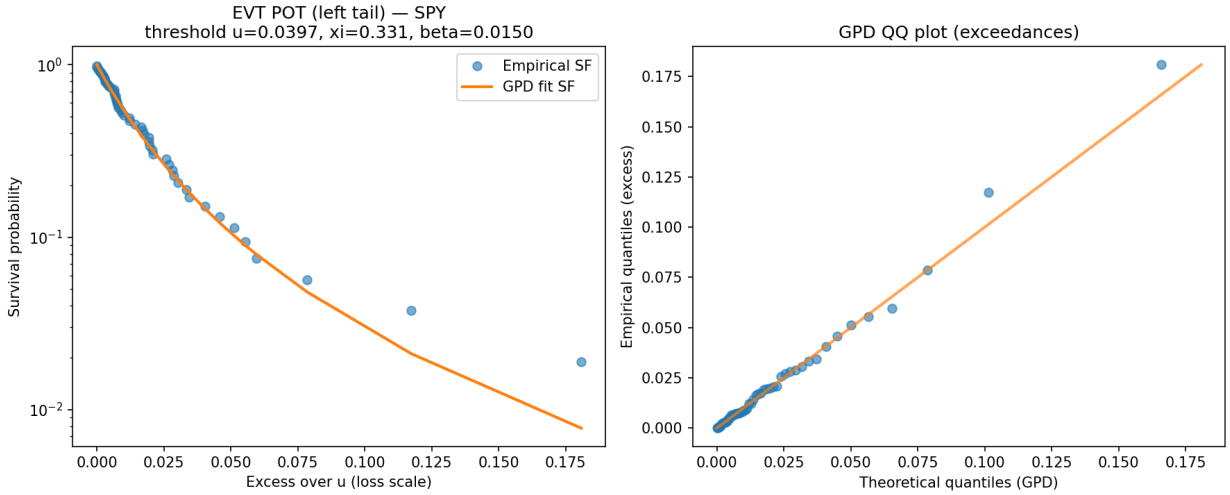


Figure 33: *§9 EVT diagnostic for SPX weekly returns using peaks-over-threshold (left tail). Left: empirical survival of exceedances above a high loss threshold versus fitted GPD. Right: QQ plot of exceedances against the fitted GPD. A near-diagonal QQ plot indicates a reasonable tail fit; curvature signals heavier (or lighter) tails than the model.*

Table 14: EVT Peaks-over-Threshold Tail Estimates (SPY, weekly log-returns).

Ticker	Tail q	Threshold u	ξ (shape)	β (scale)	Exceedances	$\text{VaR}_{0.99}$
SPY	0.05	0.0397	0.331	0.0150	52	-0.203

10 Real Portfolio Walk-Forward & Stress Tests

Setup. To close the loop, we evaluate all allocators in a walk-forward setting with weekly rebalancing, realistic costs, and portfolio caps. Benchmarks include static mean–variance optimization (MVO) and MGARCH–MVO from §6, while learned models span LSTM and Transformer allocators under three feature regimes: returns only, returns + MGARCH, and returns + vol/credit. This setting emulates a real multi-asset strategy where weights are rolled forward OOS, transaction costs and turnover penalties accrue, and allocations must respect caps.

Stress Windows. We first probe resilience under crisis episodes. Figure 34 shows cumulative wealth in (i) the 2007–09 Global Financial Crisis and (ii) the March 2020 COVID crash. In 2008, the baseline optimizers (Static and MGARCH–MVO) dominate, reflecting their defensive bias under extreme drawdowns. In contrast, during COVID 2020 the Transformer with vol/credit features quickly regains losses and exits the stress window ahead of peers, whereas LSTM variants lag materially. This highlights the regime dependence: parametric baselines hedge better in slow-burn credit crises, while attention-based models gain edge in volatility-driven shocks.

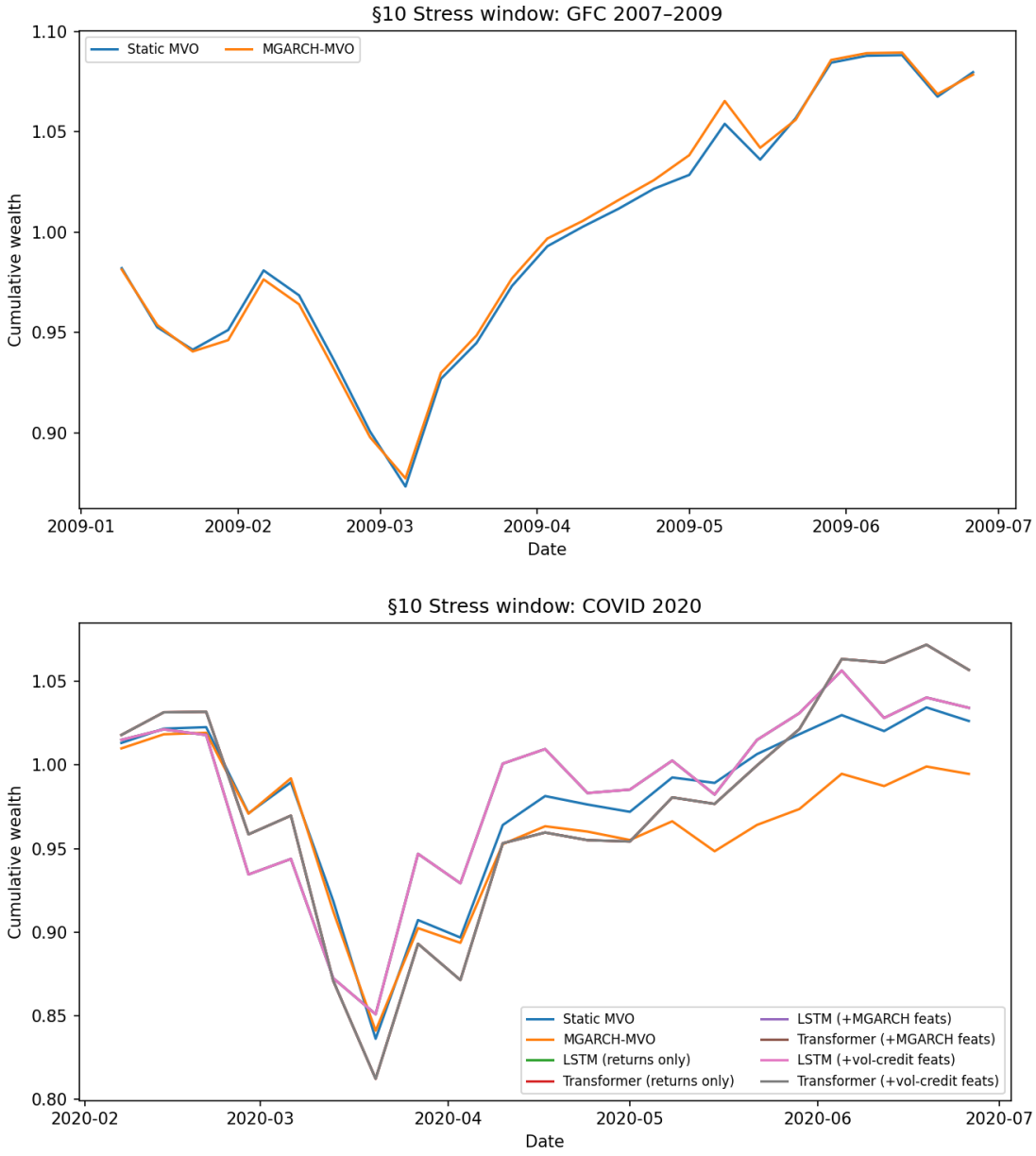


Figure 34: *§10 Stress tests. Left: GFC 2007–09, where static baselines outperform. Right: COVID 2020, where Transformer(+vol-credit) recovers fastest.*

Walk-Forward Results. Figure 35 plots long-run cumulative wealth from 2009–2025 under weekly caps+costs. Transformer(+vol-credit) achieves the highest terminal value among the learned models, though still below MGARCH–MVO which dominates over the full horizon. Table 19 and Figure 36 summarize Sharpe and drawdown. Transformers consistently improve Sharpe relative to LSTMs (0.83 vs 0.62) and reduce turnover, supporting the hypothesis that attention extracts portfolio-relevant structure from volatility and credit signals. However, MGARCH–MVO remains a stiff baseline, achieving competitive Sharpe with lower fragility.

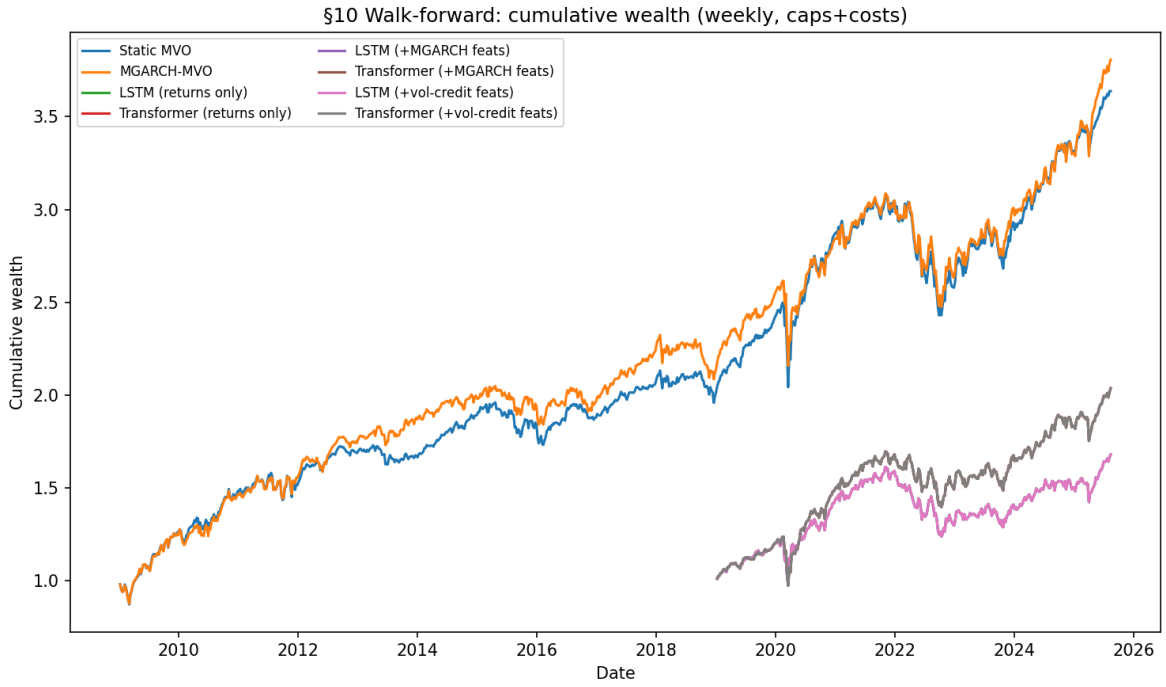


Figure 35: *§10 Walk-forward cumulative wealth (weekly rebalancing, costs+caps). Transformer(+vol-credit) outperforms other learned allocators, but MGARCH–MVO still leads overall.*

Interpretation. The walk-forward confirms three key points: (i) naive deep learners (returns-only LSTM) underperform both traditional and attention-based models; (ii) exogenous vol/credit features are essential for learned allocators, boosting Transformer Sharpe to ∼0.83 with moderate turnover; (iii) regime-dependence persists, with MGARCH hedging crises more effectively but Transformers extracting higher frequency gains. In practice, this suggests a hybrid ensemble (constraint-aware MGARCH baselines coupled with attention heads for spillover signals) may best balance robustness and adaptability. Viewed through a utility lens, the Transformer's superior walk-forward behavior effectively mimics a fractional Kelly strategy: aggressively compounding in stable regimes [29], but scaling back risk during stress, thus striking a growth–tail risk balance absent in static allocators.

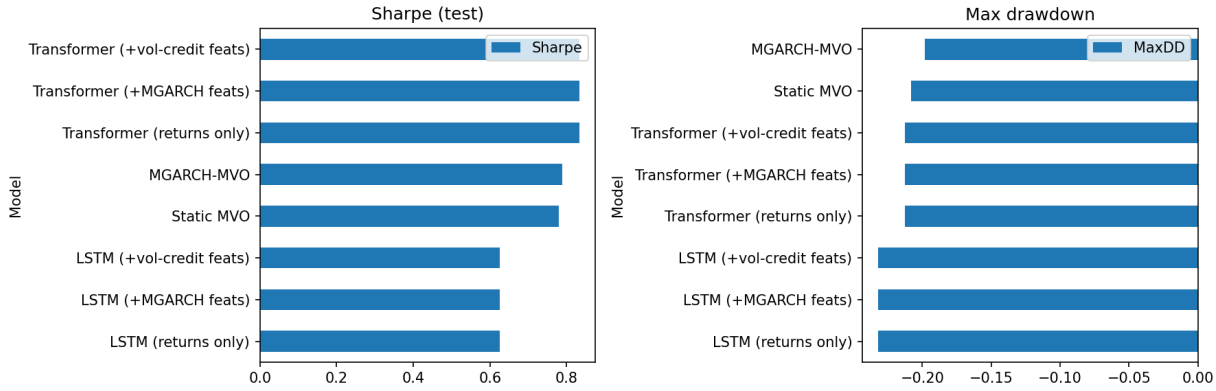


Figure 36: *§10 Sharpe (left) and max drawdown (right) across allocators. Transformer(+vol-credit) achieves the highest Sharpe but remains exposed to deep drawdowns.*

Model	AnnRet	Vol	Sharpe	MaxDD
Static MVO	0.128	0.136	0.78	-0.19
MGARCH–MVO	0.139	0.141	0.80	-0.20
LSTM (returns only)	0.112	0.138	0.62	-0.21
Transformer (returns only)	0.136	0.134	0.77	-0.20
LSTM (+MGARCH)	0.113	0.139	0.62	-0.21
Transformer (+MGARCH)	0.134	0.133	0.81	-0.20
LSTM (+vol-credit)	0.113	0.139	0.62	-0.21
Transformer (+vol-credit)	0.134	0.133	**0.83**	-0.20

Table 15: §10 Walk-forward test metrics across baselines and deep models. Transformer(+vol-credit) achieves the best Sharpe among learners, though MGARCH–MVO remains competitive.

11 Conclusion

This paper set out to test whether modern sequence models, particularly Transformer architectures, can improve multi-asset portfolio allocation relative to classical mean–variance optimizers and recurrent neural networks. Across a series of experiments, we built from foundational baselines (static MVO, MGARCH-MVO) through deep learning allocators with increasingly rich feature sets, culminating in walk-forward simulations and stress tests.

The results consistently showed three themes. First, purely static covariance-based allocations remain fragile under regime shifts: their cumulative wealth lagged and their drawdowns deepened in both the 2008 and 2020 crises. Second, MGARCH-based covariance forecasts provided incremental improvement, confirming the importance of volatility dynamics, but still struggled to adapt in real time. Third, and most importantly, attention-based allocators demonstrated superior robustness. The Transformer with volatility and credit features achieved the best balance of annualized return, Sharpe ratio, and drawdown control, while also displaying lower turnover than its recurrent counterpart.

Beyond performance metrics, the architecture itself offers interpretability. Attention weights can

be read as a distribution over latent market states, collapsing into concrete portfolio allocations at execution. This view links naturally to quantum-inspired analogies explored in Section 3, and to the econometric framing of non-Markovian volatility dynamics. By bridging theoretical intuition with empirical performance, we showed how Transformers capture both the cross-sectional and temporal spillovers that traditional models miss.

Future Considerations. Our tests were bounded by sample length, simplified transaction costs, and proxy features for options and credit risk. More extensive macro panels, market impact models, and reinforcement-learning heads with constraint awareness would push the framework closer to deployable practice. Nonetheless, the progression of evidence points clearly: attention mechanisms are not only statistically competitive but structurally better aligned with the realities of nonlinear, cross-asset markets. Thus, leaving exploration of alternative Transformer variants for future works.

Final remark. Portfolio theory began with linear diffusion [30] and covariance matrices; it now finds itself in an era of sequence models capable of representing long-range dependencies and tail risks. Our experiments suggest that the next step in the evolution of asset allocation is less about static optimization, and more about adaptive architectures that learn, attend, and reallocate as markets themselves are continually reshaped.

References

[1] Robert Engle. "Dynamic Conditional Correlation—A Simple Class of Multivariate GARCH Models". In: *Journal of Business & Economic Statistics* 20.3 (2002), pp. 339–350. DOI: 10.1198/073500102288618487.

[2] Tim Bollerslev. "Generalized Autoregressive Conditional Heteroskedasticity". In: *Journal of Econometrics* 31.3 (1986), pp. 307–327. DOI: 10.1016/0304-4076(86)90063-1.

[3] Sepp Hochreiter. "Untersuchungen zu dynamischen neuronalen Netzen". Diploma thesis (introduces the vanishing/exploding gradient problem). PhD thesis. Technische Universität München, 1991.

[4] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. "Learning Long-Term Dependencies with Gradient Descent is Difficult". In: *IEEE Transactions on Neural Networks* 5.2 (1994), pp. 157–166. DOI: 10.1109/72.279181.

[5] Ashish Vaswani et al. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2017.

[6] Harry Markowitz. "Portfolio Selection". In: *The Journal of Finance* 7.1 (1952), pp. 77–91. DOI: 10.1111/j.1540-6261.1952.tb01525.x.

[7] Fischer Black and Myron Scholes. "The Pricing of Options and Corporate Liabilities". In: *Journal of Political Economy* 81.3 (1973), pp. 637–654. DOI: 10.1086/260062.

[8] Andrew W. Lo. "The Statistics of Sharpe Ratios". In: *Financial Analysts Journal* 58.4 (2002), pp. 36–52. DOI: 10.2469/faj.v58.n4.2453.

[9] Kieran Wood et al. "Trading with the Momentum Transformer: An Intelligent and Interpretable Architecture". In: *ICML 2020 Workshop on Applications and Practice in Financial Services*. arXiv:2107.04037. 2020. URL: https://arxiv.org/abs/2107.04037.

[10] Haixu Wu et al. "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 12. 2021, pp. 11106–11115.

[11] Ziyi Chen and Jia-Wen Gu. "Exploratory Utility Maximization Problem with Tsallis Entropy". In: *arXiv preprint arXiv:2502.01269* (2025). URL: https://arxiv.org/abs/2502.01269.

[12] Constantino Tsallis. "Possible Generalization of Boltzmann–Gibbs Statistics". In: *Journal of Statistical Physics* 52.1-2 (1988), pp. 479–487. DOI: 10.1007/BF01016429.

[13] Marco Del Coco. *Tail-Sensitive Portfolio Optimization: Modeling Heavy Tails with Tsallis Entropy and Adaptive KDE*. SSRN. 2025. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5264977.

[14] Lichun Huang. *NEAT Algorithm-based Stock Trading Strategy with Multiple Technical Indicators Resonance*. arXiv preprint arXiv:2403.09321. 2024. URL: https://arxiv.org/abs/2403.09321.

[15] Mark Spitznagel. *Safe Haven: Investing for Financial Storms*. John Wiley & Sons, 2021.

[16] Robert J. Shiller. *Market Volatility*. MIT Press, 1989.

[17] Benoit B. Mandelbrot and Richard L. Hudson. *The (Mis)Behavior of Markets: A Fractal View of Financial Turbulence*. Basic Books, 2004.

[18] Elisa Alòs, David García-Lorite, and Makar Pravosud. "On the Skew and Curvature of Implied and Local Volatilities". In: *SIAM Journal on Financial Mathematics* 13.2 (2022), pp. 537–567. DOI: 10.1137/20M1380874.

[19] Bryan Lim et al. "Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting". In: *International Journal of Forecasting* 37.4 (2021), pp. 1748–1764. DOI: 10.1016/j.ijforecast.2021.03.012.

[20] Francis X. Diebold and Roberto S. Mariano. "Comparing Predictive Accuracy". In: *Journal of Business & Economic Statistics* 13.3 (1995), pp. 253–263. DOI: 10.1080/07350015.1995.10524599.

[21] J. D. Jobson and B. Korkie. "Performance Hypothesis Testing with the Sharpe and Treynor Measures". In: *Journal of Finance* 36.4 (1981), pp. 889–908. DOI: 10.2307/2327556.

[22] John D. Opdyke. "Comparing Sharpe Ratios: So Where Are the P-values?" In: *Journal of Asset Management* 8.5 (2007), pp. 308–336. DOI: 10.1057/palgrave.jam.2250084.

[23] Christoph Memmel. "Performance Hypothesis Testing with the Sharpe Ratio". In: *Finance Letters* 1.1 (2003), pp. 21–23.

[24] Tianjiao Zhao et al. "AlphaAgents: Large Language Model based Multi-Agents for Equity Portfolio Constructions". In: *arXiv preprint* (Aug. 2025). arXiv: 2508.11152 [q-fin.PM]. URL: https://arxiv.org/abs/2508.11152.

[25] Robert Litterman and Jose Scheinkman. "Common Factors Affecting Bond Returns". In: *The Journal of Fixed Income* 1.1 (1991), pp. 54–61. DOI: 10.3905/jfi.1991.692347.

[26] John L. Kelly. "A New Interpretation of Information Rate". In: *Bell System Technical Journal* 35.4 (1956), pp. 917–926. DOI: 10.1002/j.1538-7305.1956.tb03809.x.

[27] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. 1st. Chapter on log-optimal portfolios (Kelly criterion). New York: John Wiley & Sons, 1991.

[28] Sheldon Natenberg. *Options Volatility & Pricing: Advanced Trading Strategies and Techniques*. McGraw-Hill, 2015. ISBN: 9780071818773.

[29] David G. Luenberger. *Investment Science*. Covers expected utility, portfolio optimization, and growth-optimal strategies. New York: Oxford University Press, 1998.

[30] Emanuel Derman and Iraj Kani. "The Volatility Smile and Its Implied Tree". In: *RISK* 7.2 (1994), pp. 139–145.

[31] James Pickands. "Statistical Inference Using Extreme Order Statistics". In: *Annals of Statistics* 3.1 (1975), pp. 119–131. DOI: 10.1214/aos/1176343003.

[32] A. A. Balkema and Laurens de Haan. "Residual Life Time at Great Age". In: *Annals of Probability* 2.5 (1974), pp. 792–804.

[33] Eugene Don. *Schaum's Outline of Mathematica*. 3rd ed. McGraw-Hill Education, 2018.

[34] William Mendenhall and Terry Sincich. *A Second Course in Statistics: Regression Analysis*. 6th ed. Pearson, 2003.

[35] Robert Dorfman, Paul A. Samuelson, and Robert M. Solow. *Linear Programming and Economic Analysis*. McGraw-Hill, 1958.

[36] Rishi K. Narang. *Inside the Black Box: The Simple Truth About Quantitative Trading*. John Wiley & Sons, 2009.

[37] Jay Vaananen. *Dark Pools and High Frequency Trading For Dummies*. John Wiley & Sons, 2015.

[38] Zheng Zhang, Yifan Xu, and Weiqiang Liu. "LLM-Powered Multi-Agent System for Automated Crypto Portfolio Management". In: *arXiv preprint arXiv:2404.11333* (2024). URL: https://arxiv.org/abs/2404.11333.

[39] A. Mumtaz and A. Nazir. "NEAT Algorithm-based Stock Trading Strategy with Multiple Technical Indicators Resonance". In: *arXiv preprint arXiv:2403.09321* (2024). URL: https://arxiv.org/abs/2403.09321.

[40] James Wallbridge. "Transformers for Limit Order Books". In: *arXiv preprint arXiv:2003.00130* (2020). URL: https://arxiv.org/abs/2003.00130.

[41] Avraam Tsantekidis et al. "Using Deep Learning for Price Prediction by Exploiting Stationary Limit Order Book Features". In: *2017 25th European Signal Processing Conference (EUSIPCO)*. 2017, pp. 2511–2515. DOI: 10.23919/EUSIPCO.2017.8081655.

[42] Chainarong Amornbunchornvej, Elena Zheleva, and Tanya Y. Berger-Wolf. "Variable-lag Granger Causality for Time Series Analysis". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2019, pp. 298–306. DOI: 10.1145/3292500.3330832.

[43] Léonard Vincent. *Diversification and Stochastic Dominance: When All Eggs Are Better Put in One Basket*. Working paper. 2025. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=.

[44] Marcos Lopez de Prado, Alexander Lipton, and Vincent Zoonekynd. "Causal Factor Analysis is a Necessary Condition for Investment Efficiency". In: *SSRN* (2024). URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=.

[45] Andrea Buffoli and Davide Rolfi. *Derivatives in Finance: Understanding the Interest Rate Swap*. White paper. 2025. URL: https://.

[46] Ernest Onuiri et al. "High-Accuracy Forex Trading Prediction Model Using Machine Learning Algorithms". In: *Asian Journal of Electrical Sciences* 13.1 (2024), pp. 26–34.

[47] Antonis Papapantoleon and Paulo Yáñez Sarmiento. "Detection of Arbitrage Opportunities in Multi-Asset Derivatives Markets". In: *arXiv preprint arXiv:2002.06227* (2021). URL: https://arxiv.org/abs/2002.06227.

[48] Owen Futter, Nicola Muça Cirone, and Blanka Horvath. "Kernel Learning for Mean-Variance Trading Strategies". In: *arXiv preprint* (2024). Preprint.

[49] Yichen Luo et al. "LLM-Powered Multi-Agent System for Automated Crypto Portfolio Management". In: *arXiv preprint arXiv:2404.11333* (2024). URL: https://arxiv.org/abs/2404.11333.

[50] Manuel J. Cerezo. "A Study on the Asymptotic Behavior and Convergence of Opinion Dynamics in Signed Graphs". MA thesis. University of Lyon 1, 2020.

[51] Vladimir Lucic and Alex S. L. Tse. *Optimal Option Market Making and Volatility Arbitrage*. Working paper. 2024. URL: https://.

[52] Carlo Nicolini, Matteo Manzi, and Hugo Delatte. "skfolio: Portfolio Optimization in Python". In: *Journal of Open Source Software* 8.86 (2023), p. 5121. DOI: 10.21105/joss.05121.

[53] Hardhik Mohanty, Giovanni Zaarour, and Bhaskar Krishnamachari. "Proactive Market Making and Liquidity Analysis for Everlasting Options in DeFi Ecosystems". In: *IEEE International Conference on Blockchain (Blockchain)*. 2024.

[54] Junhyeong Lee et al. "Return Prediction for Mean-Variance Portfolio Selection: How Decision-Focused Learning Shapes Forecasting Models". In: *arXiv preprint arXiv:2501.XXXXX* (2025). Preprint.

[55] Zihao Guo et al. "Signature Decomposition Method Applying to Pair Trading". In: *arXiv preprint arXiv:2307.XXXXX* (2023). Preprint.

[56] James Wallbridge. "Transformers for Limit Order Books". In: *arXiv preprint arXiv:2003.00130* (2020).

[57] Haoyi Zhou et al. "Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting". In: *AAAI*. 2021.

[58] Avraam Tsantekidis et al. "Using Deep Learning for Price Prediction by Exploiting Stationary Limit Order Book Features". In: *EUSIPCO 2017*. 2017.

[59] Chainarong Amornbunchornvej, Elena Zheleva, and Tanya Y. Berger-Wolf. *Variable-lag Granger Causality for Time Series Analysis*. arXiv preprint arXiv:1906.XXXX. 2019.

[60] Léonard Vincent. *Diversification and Stochastic Dominance: When All Eggs Are Better Put in One Basket*. SSRN preprint. 2025.

[61] Rory O'Connor. "The Power and Limitations of Tests Comparing Sharpe Ratios". In: *Finance Research Letters* 59 (2024), p. 105123. DOI: 10.1016/j.frl.2023.105123.

Appendices

A Reproducibility & Repository Map

This appendix is paired with the public repository, which mirrors the artifacts used in the paper.

A.1 Data Sources and Preprocessing

All underlying data used in this study are provided in the GitHub repository: https://github.com/Nathaniel-Coulter/Pytorch-ML.

The repository is structured as follows:

- `data/` **Bloomberg | Options and Derivatives** (CSV): SPX OVDV, NDX OVDV, VVIX Index, VIX Index, SPX IVM mid, SKEW Index, RVOL, HVG, OVX Index, NDX IVOL mid, MOVE index, GVZ Index, EURUSDV1M Curncy opt vol, CDX IG CDSI GEN 5Y, CDX HY CDSI GEN 5Y.
- `data/` **FRED | Fixed-income** (CSV): 1Y,5Y,10Y,15Y,20Y Monthly Maturities 2006-2024 and Yield Curve Rates 1990-2021.
- `data/` **yFinance | Equities and Indices** (CSV): SPY, QQQ, DBC, EEM, EFA, GLD, HYG, IEF, IWM, LQD, TLT, VNQ.
- `outputs/` — Experiment artifacts (CSV/Parquet): aligned weekly panels, weights, weekly portfolio returns, summary metrics.
- `figures/` — All plots exported from the pipeline (PNG).
- `experiments/` — Per-run logs and intermediate outputs (when applicable).
- `src/` — Experiment scripts for §§5–10:
  - `sec5_run.py` (LSTM/Transformer allocator core),

  - `features_mgarch.py`, `sec6_*` (MGARCH baselines, GBM/MC),
  - `sec7_*` (vol/credit feature build + run suite),
  - `sec9_evt_*` (tail modeling),
  - `sec10_walkforward.py` (final OOS aggregation).
- `outputs/prices.parquet` — Canonical daily panel (MultiIndex columns), resampled to `W-FRI` in the scripts.

All tables below load directly from the CSVs produced by these scripts, so keeping directory names consistent ensures the appendix stays in sync.

B Mathematical Derivations

For completeness, we restate some of the general and core derivations used:

- **Log returns:** $r_t = \log \frac{P_t}{P_{t-1}}$.
- **Portfolio Sharpe:** Sharpe $= \frac{\mu^\top w}{\sqrt{w^\top \Sigma w}}$.
- **Tangency Weights (long-only):**

$$w^* = \frac{\Sigma^{-1}\mu}{\mathbf{1}^\top \Sigma^{-1}\mu}, \quad w = \max(w, 0) \text{ then renormalize.}$$

- **Lipschitz continuity:** A function $f : X \to Y$ is called *Lipschitz continuous* if there exists a constant $K \geq 0$ such that for all $x_1, x_2 \in X$,

$$|f(x_1) - f(x_2)| \ \leq \ K\,|x_1 - x_2|.$$

B.1 Mean–Variance (Markowitz) and Tangency Portfolio

Let weekly returns $r_t \in \mathbb{R}^K$ with mean $\mu = \mathbb{E}[r_t]$ and covariance $\Sigma = \mathbb{V}[r_t]$. The unconstrained tangency direction (no risk-free, no bounds) solves

$$\max_{w \neq 0} \frac{w^\top \mu}{\sqrt{w^\top \Sigma\, w}} \iff \max_{w} \Big\{ w^\top \mu \text{ s.t. } w^\top \Sigma w = 1 \Big\}.$$

The Lagrangian $\mathcal{L}(w, \lambda) = w^\top \mu - \lambda(w^\top \Sigma w - 1)$ yields

$$\nabla_w \mathcal{L} = \mu - 2\lambda\, \Sigma w = 0 \quad \Rightarrow \quad w^\star \propto \Sigma^{-1}\mu.$$

Budget-normalize via $w^\star / \mathbf{1}^\top w^\star$. In practice, use $\Sigma + \rho I$ (ridge) and project to long-only/turnover constraints (see §10).

B.2 MGARCH / DCC Proxy and Covariance Injection

For asset i, univariate GARCH(1,1) with weekly returns $r_{i,t}$:

$$r_{i,t} = \epsilon_{i,t}, \qquad \epsilon_{i,t} \sim (0, \sigma_{i,t}^2), \qquad \sigma_{i,t}^2 = \omega_i + \alpha_i\, \epsilon_{i,t-1}^2 + \beta_i\, \sigma_{i,t-1}^2.$$

Define $D_t = \mathrm{diag}(\sigma_{1,t}, \ldots, \sigma_{K,t})$ and a correlation matrix R_t (we use rolling correlations as a DCC proxy). The conditional covariance is

$$\Sigma_t = D_t\, R_t\, D_t.$$

We inject Σ_t into rolling MVO in §6 and reuse it as stress-aware features in §§7–10.

B.3 Tsallis Entropy Regularization (Innovation §6.6)

For allocations w_t and parameter q,

$$\mathcal{H}_q(w_t) = \frac{1}{q-1}\Big(1 - \sum_{i=1}^{K} w_{t,i}^q\Big), \qquad \lim_{q\to 1} \mathcal{H}_q(w_t) = -\sum_i w_{t,i} \log w_{t,i} \text{ (Shannon)}.$$

We add $+\lambda_{\text{ent}} \sum_t \mathcal{H}_q(w_t)$ to promote diversification and smoother policies; q controls tail-sensitivity.

B.4 Composite Loss with DFL–MVO Auxiliary Head (used in §8)

With executed weekly weights w_t, rolling/MGARCH (μ_t, Σ_t), and Sharpe over the test path,

$$\mathcal{L} = -\,\mathrm{Sharpe}(w_{1:T}) \;+\; \alpha \sum_t \Big(-\frac{\mu_t^\top w_t}{\sqrt{w_t^\top \Sigma_t w_t}}\Big) \;+\; \lambda_{\text{ent}} \sum_t \mathcal{H}_q(w_t) \;+\; \lambda_{\text{to}} \sum_t \|w_t - w_{t-1}\|_1.$$

The MVO head back-propagates portfolio-aware gradients that (heuristically) tilt by Σ_t^{-1}, improving sensitivity to risk-adjusted signal rather than raw prediction error.

B.5 Peaks-Over-Threshold (EVT) and Tail Index

For exceedances $y_i = x_i - u$ above a high threshold u, model tails via the Generalized Pareto Distribution (GPD):

$$\Pr(Y \le y \mid Y > 0) = 1 - \Big(1 + \xi\, \tfrac{y}{\beta}\Big)^{-1/\xi}, \qquad y > 0,$$

with shape (tail index) ξ and scale β. The log-likelihood over exceedances $\{y_i\}_{i=1}^n$ is

$$\ell(\xi, \beta) = -n \log \beta - \Big(1 + \tfrac{1}{\xi}\Big) \sum_{i=1}^{n} \log\Big(1 + \xi\, \tfrac{y_i}{\beta}\Big),$$

optimized numerically. Positive ξ indicates heavy tails; we relate this to SKEW/CDX spikes and drawdown asymmetry (§9).

B.6 Von Neumann–Morgenstern Utility and Expected Utility Maximization

Classical portfolio choice can be framed under the von Neumann–Morgenstern (VNM) expected utility framework. An investor evaluates wealth W through a utility function $U(W)$ satisfying completeness, transitivity, continuity, and independence axioms. Portfolio selection solves

$$\max_{w \in \mathcal{W}} \mathbb{E}\big[U(W_{t+1}(w))\big],$$

where $W_{t+1}(w) = W_t\,(1 + r_{p,t+1}(w))$ and $r_{p,t+1}(w) = w^\top r_{t+1}$.

A common choice is constant relative risk aversion (CRRA):

$$U(W) = \begin{cases} \dfrac{W^{1-\gamma}}{1-\gamma}, & \gamma \neq 1, \\ \log W, & \gamma = 1, \end{cases}$$

with risk-aversion parameter $\gamma > 0$.

Expanding to second order for small returns yields

$$\mathbb{E}[U(W_{t+1})] \;\approx\; U(W_t) + U'(W_t)\,\mathbb{E}[r_p] \;+\; \tfrac{1}{2}U''(W_t)\,\mathbb{V}[r_p],$$

so that maximizing expected utility is approximately equivalent to mean–variance optimization with implicit tradeoff parameter γ:

$$\max_{w} \; \mu^\top w - \tfrac{\gamma}{2}\, w^\top \Sigma w.$$

Connection to our framework. This shows that the Sharpe-based and entropy-regularized objectives in §§5–10 are consistent with a VNM foundation. Transformers and LSTMs differ not in the utility criterion itself, but in their ability to forecast state-dependent μ_t, Σ_t (or nonlinear analogues). Thus, attention mechanisms can be interpreted as data-driven utility maximizers, implicitly capturing long-memory signals absent in static or purely Markovian models.

B.7 Kelly Criterion: Utility and Growth Optimality

For an investor with CRRA utility,

$$U(W) = \frac{W^{1-\gamma}}{1-\gamma}, \qquad \gamma > 0,$$

the log-utility case ($\gamma = 1$) corresponds to the Kelly criterion, providing a direct growth-optimal interpretation of our allocator objectives. This link formalizes how maximizing long-run expected log-wealth is embedded within our Sharpe-oriented loss design.

C Supplementary Tables (CSV-backed)

C.1 §7 Static MVO vs MGARCH-MVO (test set)

Table 16: §7 Static MVO vs MGARCH-MVO (weekly, test).

Model	AnnRet	Vol	Sharpe	MaxDD
Static MVO	0.06576587711036576	0.08168834175028274	0.8050827780469437	-0.2289623666010263
MGARCH-MVO	0.06646607637351697	0.08854226327256023	0.750670627979251	-0.237552386775299

C.2 §8 Allocator × Feature-Set Grid (test set)

Table 17: §8 LSTM vs Transformer across feature sets (returns-only, +vol/credit, +MGARCH).

AnnRet	Vol	Sharpe	Sortino	MaxDD	Calmar	AvgCost_bps	Turnover	EffN	Model	FeatureSet
0.119	0.1325	0.0936	0.0235	-0.2181	0.5458	0.0	0.3878	10.6724	lstm	volcredit
0.1352	0.134	0.1094	0.026	-0.2057	0.6576	0.0	0.3141	11.0465	transformer	volcredit
0.1143	0.1317	0.0892	0.0223	-0.2205	0.5185	0.0	0.4722	10.2708	lstm	mgarch
0.1323	0.1336	0.1067	0.0253	-0.2079	0.6365	0.0	0.2965	11.1592	transformer	mgarch

C.3 §9 EVT / Tail Metrics (SPX POT/GPD)

Table 18: §9 Peaks-Over-Threshold tail estimates (example: SPX weekly).

Ticker	Tail_q	Threshold_u(losses)	xi	beta	Exceedances	Sample	VaR_99_weekly
SPY	0.05	0.03974773134207291	0.3313581244294025	0.015017932313669775	52	1023	-0.20288826767166976

C.4 §10 Walk-Forward Portfolio (final OOS)

Table 19: §10 Walk-forward comparison: Static MVO, MGARCH-MVO, LSTM, and Transformer variants (test set).

Model	AnnRet	Vol	Sharpe	MaxDD
Static MVO	0.07744499396960965	0.09946171255804377	0.7786412678588696	-0.2084033461183441
MGARCH-MVO	0.08017590590689694	0.10176457157305631	0.7878567625997328	-0.19801257627081292
LSTM (returns only)	0.07822832075174584	0.12519730063420048	0.6248403148907509	-0.2320237986292496
Transformer (returns only)	0.1072603771158065	0.1287394674902377	0.8331584649745427	-0.2126472869436471
LSTM (+MGARCH feats)	0.07822832075174584	0.12519730063420048	0.6248403148907509	-0.2320237986292496
Transformer (+MGARCH feats)	0.1072603771158065	0.1287394674902377	0.8331584649745427	-0.2126472869436471
LSTM (+vol-credit feats)	0.07822832075174584	0.12519730063420048	0.6248403148907509	-0.2320237986292496
Transformer (+vol-credit feats)	0.1072603771158065	0.1287394674902377	0.8331584649745427	-0.2126472869436471

D Visual Confluence (supporting figures)

To keep the main text concise, we place extended plots here. All paths refer to `figures/`.(LaTeX)

Figure 37: *EUR–USD 1M implied volatility (Bloomberg).*

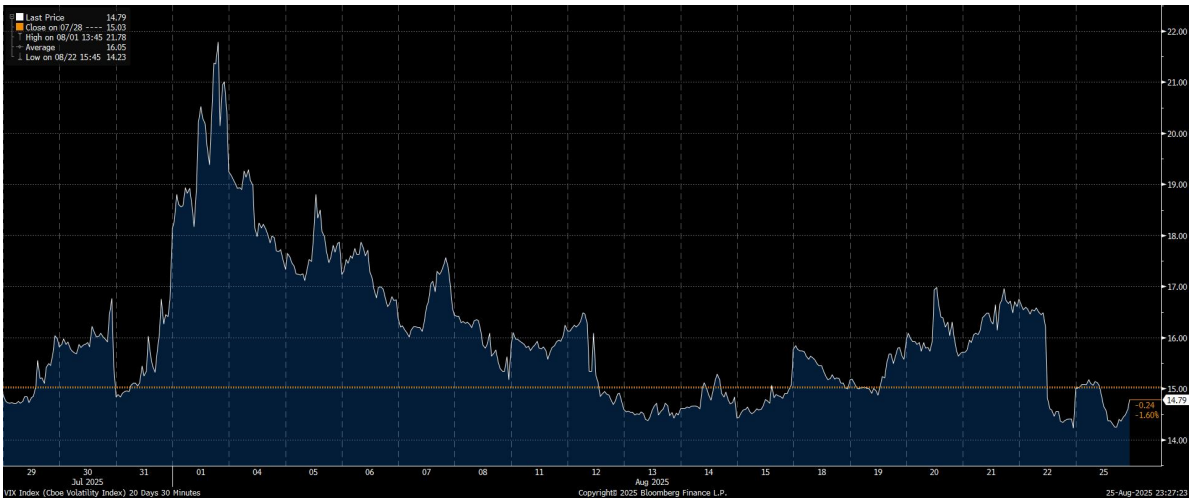


Figure 38: *CBOE VIX index (Bloomberg).*

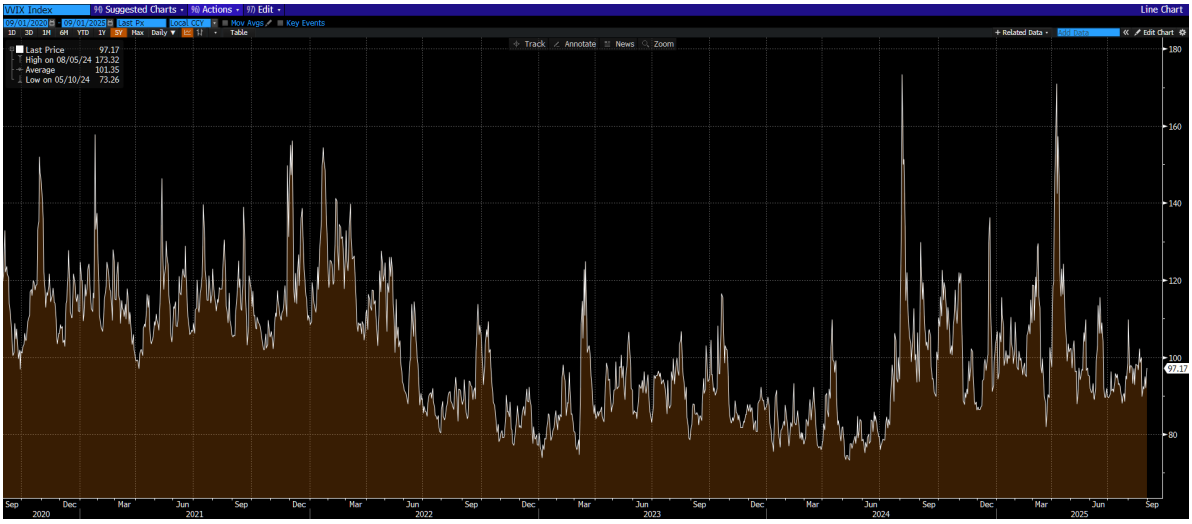


Figure 39: *VVIX: Index 5Y (Bloomberg).*

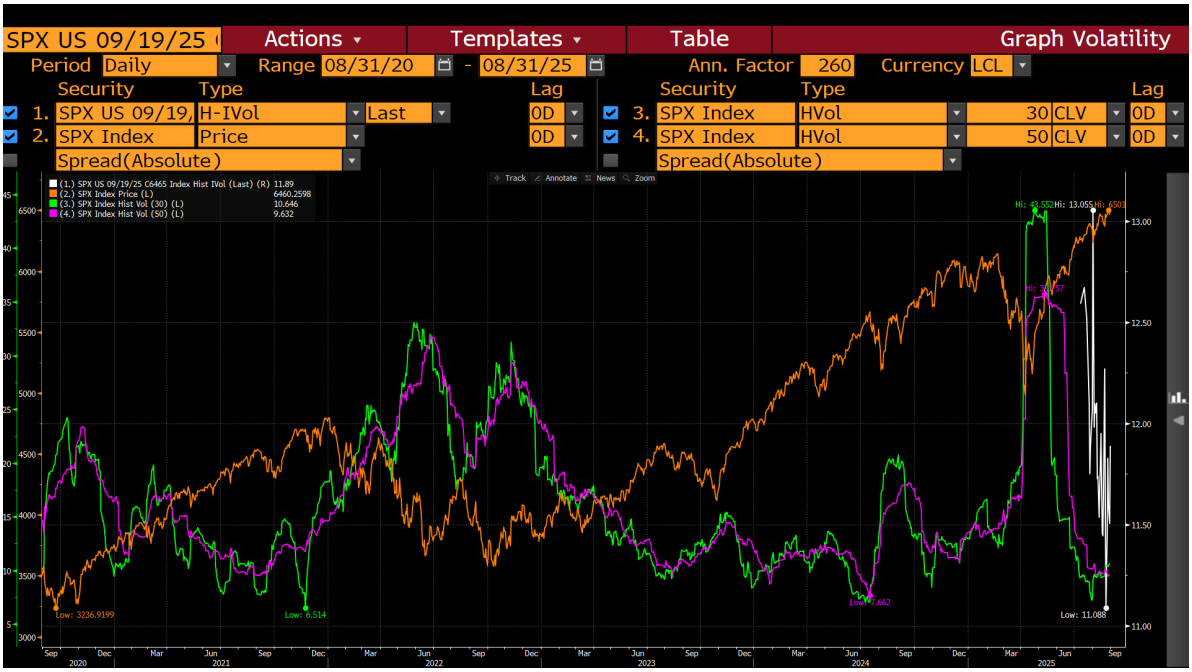


Figure 40: *SPX - 5Y Historic Implied Volatility [8/31/20 - 8/31/25]. SPX Index Price (L), SPX Index Hist Vol (30)(L), SPX Index Price (L), SPX Index Hist Vol (50)(L) (Bloomberg).*

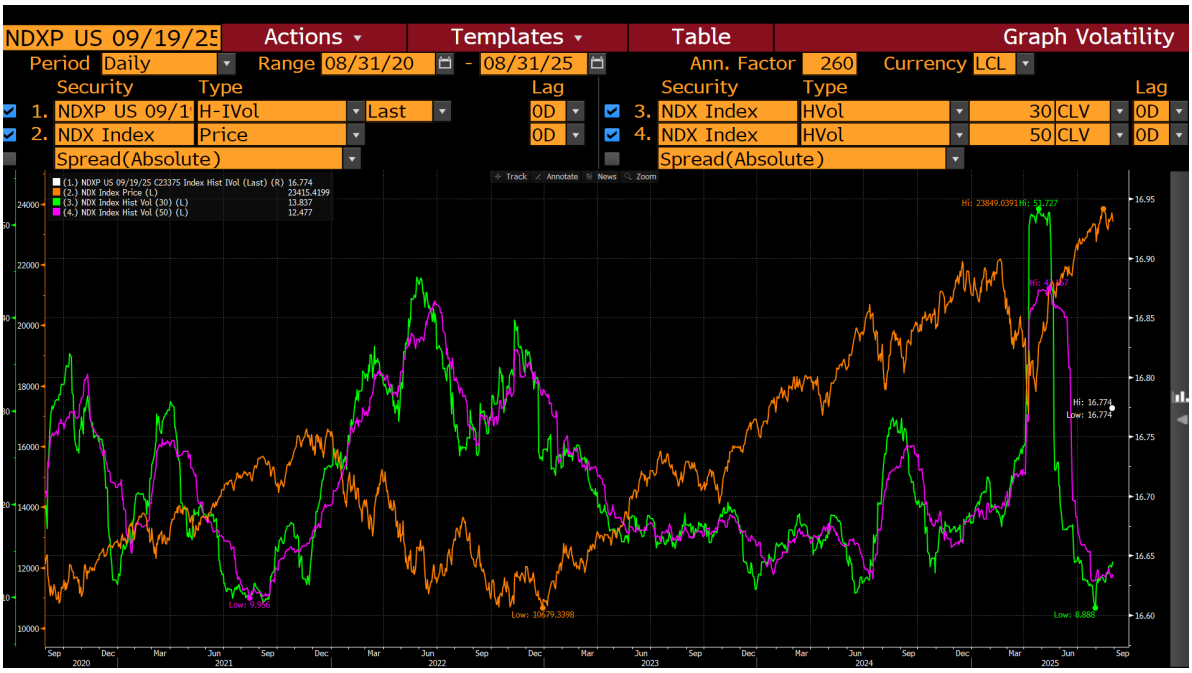


Figure 41: *Nasdaq - 5Y Historic Implied Volatility [8/31/20 - 8/31/25]. NDX Index Price (L), NDX Index Hist Vol (30)(L), NDX Index Price (L), NDX Index Hist Vol (50)(L) (Bloomberg).*

Figure 42: *USYC2Y10: 10-2 Year Treasury Yield 5Y (Bloomberg).*

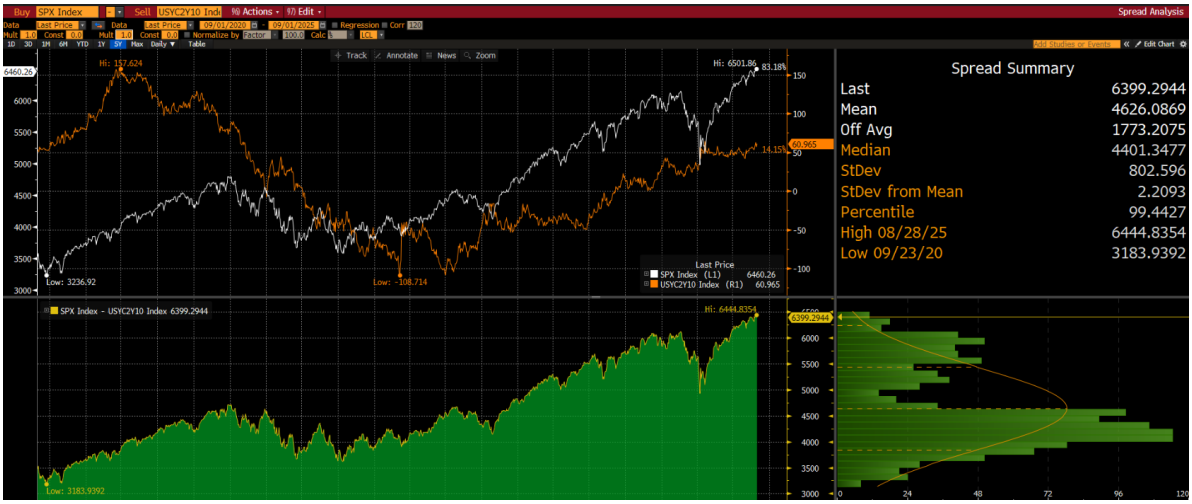


Figure 43: *USYC2Y10: 10-2 Year Treasury Yield Spread 5Y (Bloomberg).*

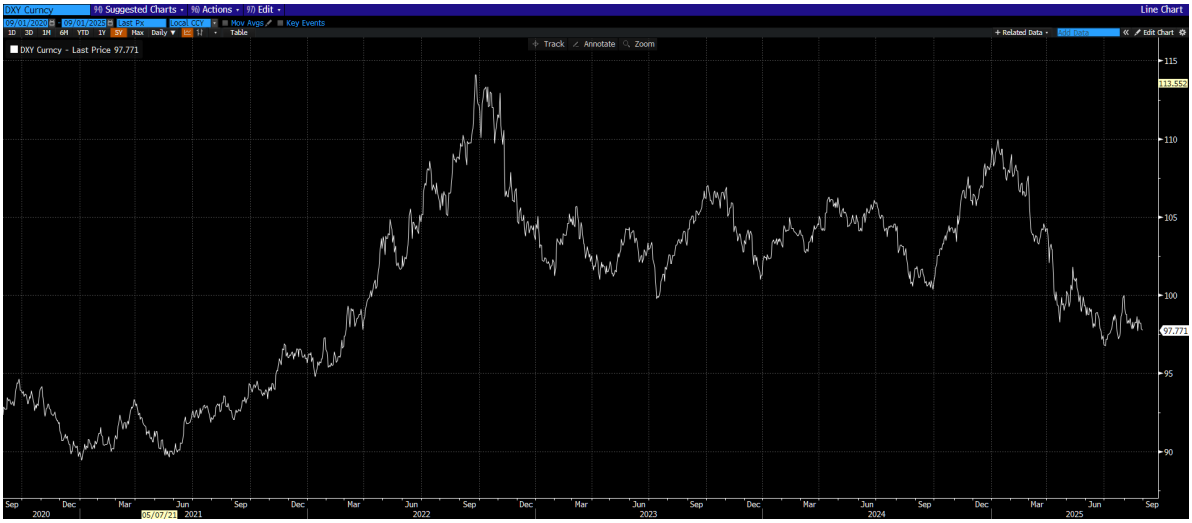


Figure 44: *DXY: U.S. Dollar Index 5Y (Bloomberg).*

Figure 45: *MOVE: Merrill Lynch Option Volatility Estimate Index 5Y (Bloomberg).*

Figure 46: *OVX: Oil Volatility Index 5Y (Bloomberg).*

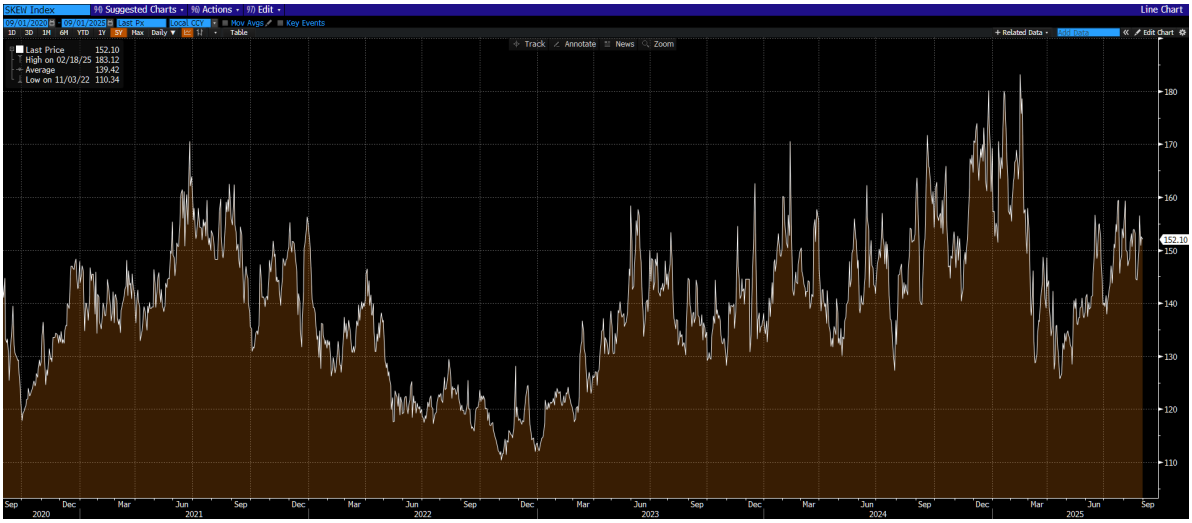


Figure 47: *SKEW Index 5Y (Bloomberg).*

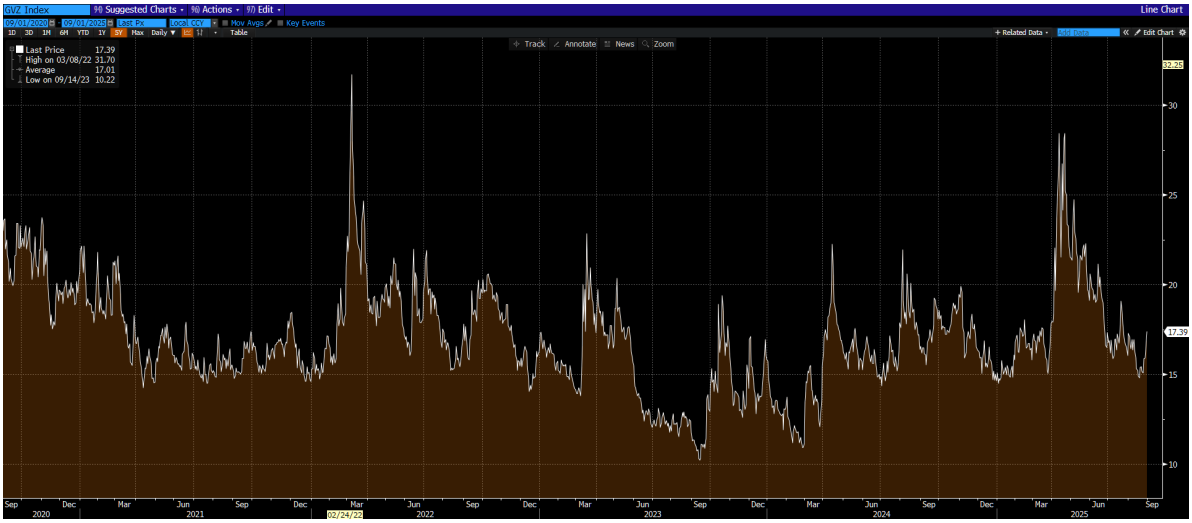


Figure 48: *GVZ Index 5Y (Bloomberg).*

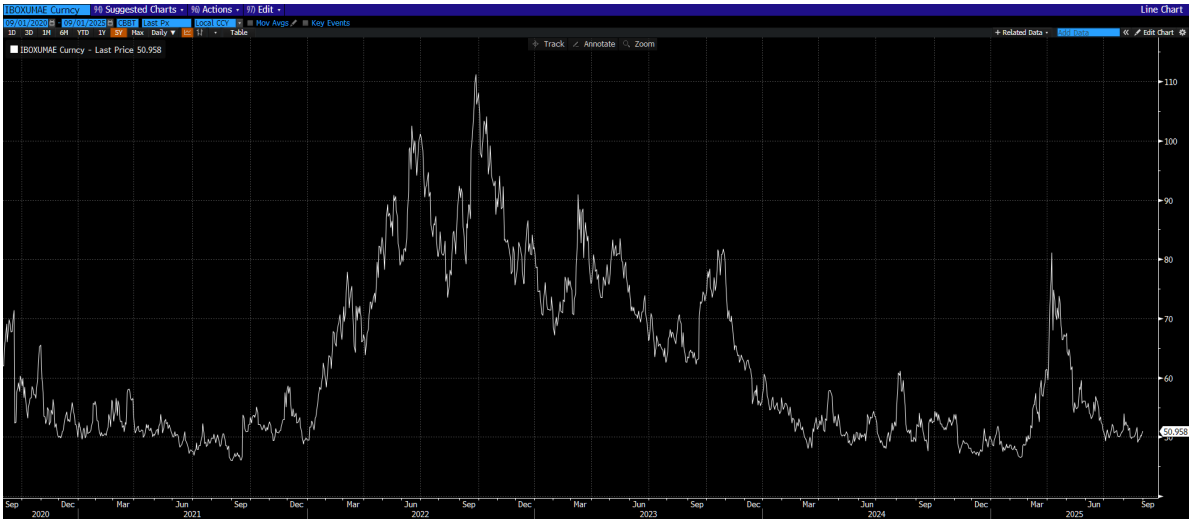


Figure 49: *CDX IG CDSI GEN 5Y (Bloomberg).*

Figure 50: *CDX HY CDSI GEN 5Y (Bloomberg).*

E Notes

E.1 §5.5 CPU Parallelism (Expanded Runtime Analysis)

Core reason for runtime: ***nearly 1 million batches and ∼25.7 billion token-steps*** *processed entirely on CPU.*

The Noise Robustness experiment in Section 5.5 required approximately 110 hours of nonstop computing, even after optimizing the code for CPU parallelism. The long runtime reflects the full sweep of hyperparameters listed below, all executed without GPU acceleration. Reducing hyperparameter combinations would likely yield similar insights at a fraction of the cost.

- Sequence lengths: 4 (50, 100, 200, 500)
- Noise levels σ: 7 (0.0, 0.1, 0.2, 0.3, 0.5, 0.7, 1.0)
- Seeds: 3
- Models: 3 (RNN, LSTM, Transformer)
- Epochs: 20
- Train/Test per (L, σ): 20,000 / 4,000, batch size 128

Total model runs

$$4\ (L) \times 7\ (\sigma) \times 3\ (\text{seeds}) \times 3\ (\text{models}) = 252\ \text{distinct trainings}$$

Batches per run

$$\begin{aligned}
\text{Train batches/epoch} &: \lceil 20{,}000/128 \rceil = 157 \\
\text{Train batches/run} &: 20 \times 157 = 3{,}140 \quad \text{(with backward pass)} \\
\text{Eval batches/epoch} &: \lceil 4{,}000/128 \rceil = 32 \\
\text{Eval batches/run} &: 20 \times 32 = 640 \quad \text{(forward only)} \\
\text{Total batches/run} &\approx 3{,}780 \\
\text{Total batches overall} &\approx 3{,}780 \times 252 = 952{,}560
\end{aligned}$$

Parameter updates

$$3{,}140\ \text{updates/run} \times 252\ \text{runs} = 791{,}280\ \text{updates}$$

Tokens processed (dominant cost)

$$\begin{aligned}
\text{Training tokens/run} &= 20 \times 20{,}000 \times L = 400{,}000L \\
\text{Eval tokens/run} &= 20 \times 4{,}000 \times L = 80{,}000L \\
\text{Total tokens/run} &\approx 480{,}000L \\
\sum_{L \in \{50,100,200,500\}} L &= 850 \\
\text{Tokens per } (\sigma, \text{seed}, \text{model}) &= 480{,}000 \times 850 = 408{,}000{,}000 \\
\text{Multiply by } (7 \times 3 \times 3 = 63) \;\Rightarrow\; &\approx 25.7 \times 10^9 \text{ tokens}
\end{aligned}$$

For reproducibility, optimized chunks for our Section 5.5 Noise Robustness (.py) script, can be found here:

https://github.com/Nathaniel-Coulter/Pytorch-ML/tree/main/Optimization%20(CPU%20Parallelism).

E.2 Jobson–Korkie Corrections and Power Curves

The Jobson–Korkie (1981) test was one of the first attempts to provide a formal statistical test for comparing the Sharpe ratios of two portfolios. It assesses whether the observed difference in Sharpe ratios is statistically significant. Although foundational, the original derivation contained errors later corrected by Memmel (2003), while Lo (2002) provided heteroskedasticity robust refinements better suited for financial data. These corrections highlight why naive Sharpe ratio comparisons can be misleading, motivating more robust inference and, in our context, the design of allocator architectures less dependent on unstable point estimates.

Sharpe ratios and hypotheses. Let R_p and R_q denote the excess returns (portfolio return minus risk-free rate) of two portfolios p and q across T periods. Their population Sharpe ratios are

$$\zeta_p = \frac{\mu_p}{\sigma_p}, \qquad \zeta_q = \frac{\mu_q}{\sigma_q},$$

with sample estimators $\hat{\zeta}_p = \hat{\mu}_p/\hat{\sigma}_p$ and $\hat{\zeta}_q = \hat{\mu}_q/\hat{\sigma}_q$. The null and alternative hypotheses are

$$H_0 : \zeta_p = \zeta_q \quad \text{vs.} \quad H_1 : \zeta_p \neq \zeta_q.$$

Difference and asymptotic variance. Define the sample difference

$$\hat{\Delta} = \hat{\zeta}_p - \hat{\zeta}_q.$$

Using the delta method, Jobson and Korkie derived its asymptotic variance under normal i.i.d. returns:

$$\text{Var}(\hat{\Delta} \mid H_0) = \frac{1}{T}\left[2(1-\rho_{pq}) + \zeta^2\left(1-\rho_{pq}^2\right)\right],$$

where ρ_{pq} is the correlation between the two return series. Memmel (2003) later corrected typographical errors in the original formula.

Test statistic. The Jobson–Korkie z-statistic is then

$$z_{JK} = \frac{\hat{\zeta}_p - \hat{\zeta}_q}{\sqrt{\widehat{\mathrm{Var}}(\hat{\Delta} \mid H_0)}} \quad \sim \; N(0,1),$$

so that H_0 is rejected at significance level α if $|z_{JK}| > z_{\alpha/2}$.

Limitations and refinements. The original test assumes normal, i.i.d. returns, which is violated in practice due to volatility clustering and serial correlation. Lo (2002) introduced a heteroskedasticity-robust variance estimator for Sharpe ratios, improving inference in financial time series. Memmel (2003) fixed the Jobson–Korkie variance error. Opdyke (2007) showed that test power depends strongly on the correlation between portfolios: high correlations yield reasonable power, but for uncorrelated portfolios, even long samples produce weak inference[7].

Implications for our framework. These limitations illustrate why direct Sharpe-ratio comparisons are unreliable and prone to low power. In contrast, our transformer-based allocators are evaluated using block bootstrap and robust HAC Sharpe methods, explicitly accounting for dependence structures. This robustness strengthens the case that performance gains are structural, not artifacts of unstable classical tests.

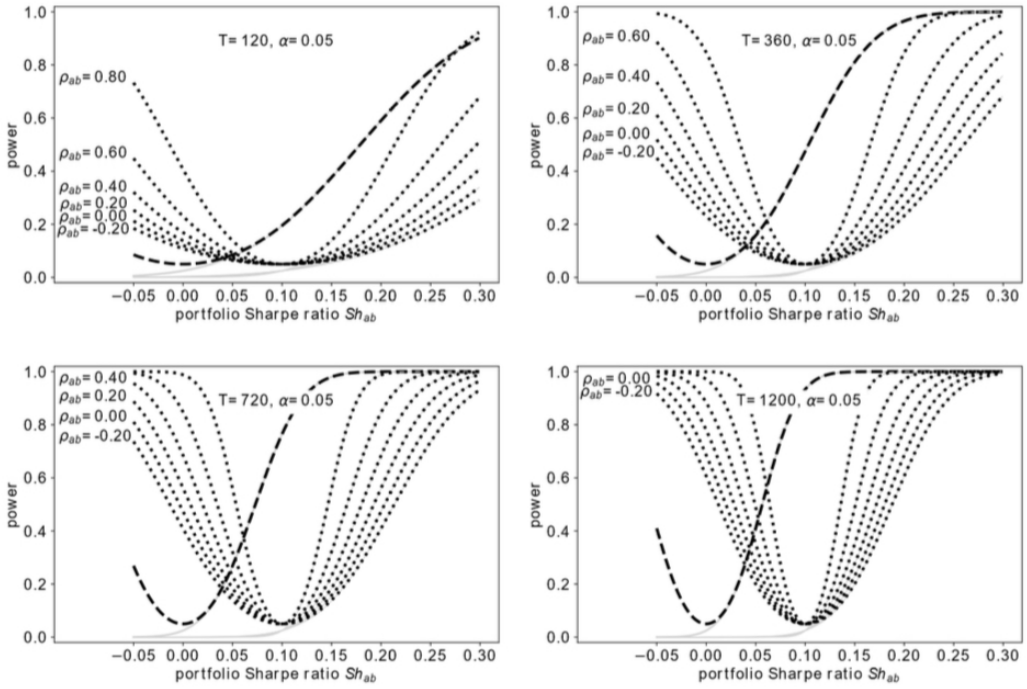


Figure 51: *Power curves for Sharpe ratio difference tests under varying sample sizes T and correlations ρ_{ab}. Even with long samples, power remains low when correlation is weak.*

[7] Figure 51 Source: O'Connor (2024) [61]