Vol. 2, No. 4, pp.: 5–17, 2024

A Computational Systems Perspective on Risk-Aware Portfolio Optimization with Actor–Critic Learning

Liu Chen[1], Yifan Zhao[2], Mingyu He[1,*]

[1] School of Economics, Beijing Technology and Business University, Beijing 100048, China

[2] School of Management, University of Chinese Academy of Sciences, Beijing 100190, China

* *mingyu.he@btbu.edu.cn*

Article Information	
Received	16 September 2024
Accepted	18 November 2024
DOI	https://doi.org/10.63646/datamind.2024.020402

Abstract

This study reframes portfolio optimization as a computational systems problem rather than a purely financial forecasting task. We argue that the effectiveness of reinforcement learning in dynamic capital allocation depends on the co-design of data engineering, state representation, actor–critic interaction, and execution-aware risk governance. On this basis, the paper develops a modular architecture, Computational Systems Actor–Critic Learning (CS-ACL), that combines data harmonization, discriminative state compression, clipped actor–critic learning, and a risk controller that penalizes drawdown, turnover, and unstable policy shifts. Two public datasets are used: a large market panel derived from Yahoo Finance and a structured intelligent-finance dataset with allocation signals and asset categories. The framework is evaluated under walk-forward backtesting against mean–variance allocation, LSTM-assisted allocation, PPO, SAC, and TD3 baselines. Results show that CS-ACL achieves the strongest overall balance among return, volatility, drawdown, turnover, and convergence stability. Under identical transaction-cost assumptions, the model delivers an annualized return of 18.7%, a Sharpe ratio of 1.67, and a maximum drawdown of -10.8%, while maintaining lower policy oscillation than competing RL baselines. The key analytical contribution is that actor–critic learning performs best when it is treated as part of a computational infrastructure rather than as an isolated prediction engine. The paper therefore contributes to data-driven AI and

computational discovery by offering a reproducible systems perspective on risk-aware capital allocation in non-stationary financial environments.

Keywords: ***risk-aware portfolio optimization; actor–critic learning; reinforcement learning; computational systems; financial analytics; policy stability***

1. Introduction

Portfolio optimization remains one of the most persistent challenges in computational finance because capital allocation decisions must be made under uncertainty, temporal dependence, and dynamic cross-asset correlation. Classical formulations—from mean–variance optimization to tactical rebalancing heuristics—treat allocation as a constrained mathematical problem, but contemporary digital markets reveal an additional layer of complexity. Asset prices evolve as streams, technical indicators respond to regime shifts with different lag structures, and transaction costs make the path of the policy as important as the final weight vector. In this environment, a portfolio system is successful only when it can translate heterogeneous signals into stable actions without becoming fragile under market turbulence.

Reinforcement learning (RL) has renewed interest in portfolio optimization precisely because it promises adaptive decision making. Rather than solve for a single static optimum, an RL agent can observe the market state, allocate capital, evaluate the consequences, and update the policy in a sequential loop. Financial studies increasingly report that deep reinforcement learning outperforms static rules when the environment is nonlinear and non-stationary (Deng et al., 2017; Zhang et al., 2020; Yang et al., 2023). Yet the empirical record also reveals large variation across datasets, trading horizons, and reward definitions. In many cases, apparent gains depend on aggressive rebalancing or unstable policy shifts that disappear once costs and risk controls are included (Millea, 2021; Wang and Ku, 2022).

This paper approaches the issue from a different angle. Instead of asking only whether RL can improve returns, it asks how a portfolio optimization system should be architected if actor–critic learning is to be computationally stable, risk-aware, and reproducible. This framing shifts the emphasis from model novelty to system design. Data quality, feature scaling, state compression, reward engineering, and policy-governance layers are all treated as co-determinants of performance. The proposed framework, Computational Systems Actor–Critic Learning (CS-ACL), uses actor–critic learning as its decision core, but it embeds that core in a modular pipeline that makes the policy less sensitive to noisy features, market regime shifts, and destructive updates.

The contribution is therefore twofold. Empirically, the paper develops and evaluates a risk-aware actor–critic architecture on two public financial datasets under walk-forward testing. Conceptually, it argues that portfolio optimization in intelligent finance is not merely a prediction problem but a computational systems problem. This perspective is aligned with

recent work on machine learning in finance, which shows that data representation and workflow design are as decisive as algorithm choice for out-of-sample performance (Heaton et al., 2017; Fischer and Krauss, 2018; Gu et al., 2020).

2. Related Work and Computational Framing

The literature relevant to this study can be grouped into three streams. The first stream uses supervised learning to improve the information set entering an allocation rule. Deep learning models have been applied to return forecasting, volatility prediction, and factor extraction, often with better pattern recognition than linear or tree-based approaches when the feature space is high dimensional (Bao et al., 2017; Fischer and Krauss, 2018; Gu et al., 2020). Yet these systems typically separate prediction from allocation, so they do not directly learn the sequential consequences of weight changes.

The second stream uses reinforcement learning directly for financial decision making. Early work showed that direct reinforcement learning could turn financial signal representation and trading into a single optimization problem (Deng et al., 2017). Subsequent portfolio studies introduced model-free deep RL, recurrent reinforcement learning, and graph-enhanced policy learning for capital allocation across multiple assets (Jiang et al., 2017; Almahdi and Yang, 2017; Lei et al., 2020; Sun et al., 2024). More recent papers extend the space toward clipped PPO, risk-sensitive policies, bond allocation, and value-distribution actor–critic systems (Wu et al., 2024; Nunes, 2025; Wang et al., 2025; Yang et al., 2026). The common finding is that RL can be useful in volatile environments, but results depend strongly on how state, action, and reward are engineered.

The third stream concerns computational infrastructure, explainability, and reproducibility. FinRL and related frameworks emphasize that a financial RL pipeline should not be reduced to a single model checkpoint; it should include data processors, environments, benchmark layers, and reporting modules (Liu et al., 2021). Explainable allocation studies similarly note that trust in RL-based portfolios depends on whether the system can expose the drivers of decisions and the trade-offs between return seeking and risk control (Guan and Liu, 2021). This systems-oriented stream is essential because high-performing portfolios are not useful if the training process is unstable, the policy is opaque, or the computational cost is impractical.

Actor–critic learning provides a natural bridge across these streams. The actor generates actions, the critic evaluates state value, and asynchronous or clipped updates can improve stability under noisy gradients (Mnih et al., 2016; Schulman et al., 2017). Advances in entropy regularization, distributional RL, and risk-sensitive constraints further suggest that actor–critic systems can support robust portfolio design when return optimization is balanced with drawdown and volatility control (Haarnoja et al., 2018; Bellemare et al., 2017; Zhong et al., 2025). The unresolved question is therefore not whether actor–critic learning is powerful, but

how it should be embedded in a financial workflow so that policy quality remains stable after costs and risk penalties are included.

3. Data, State Design, and Computational Architecture

The empirical setting combines two public datasets selected for complementary computational roles. The first is a large panel of market observations derived from Yahoo Finance, including price series, returns, moving-average indicators, momentum measures, and volatility variables. The second is a structured intelligent-finance dataset containing asset categories and allocation-relevant signals. The two sources are not simply concatenated. They are harmonized through a preprocessing pipeline that performs missing-value repair, outlier screening, temporal alignment, and Z-score normalization. This sequence is essential because reinforcement learning is highly sensitive to state instability. If one feature family dominates scale or noise, the critic may mis-estimate value and the actor may overreact to local fluctuations.

Table 1. Data blocks and computational role in the CS-ACL workflow.

Block	Primary variables	Systems role
Market price panel	Open, high, low, close, volume	Captures cross-asset market states
Technical indicator layer	Returns, MA, EMA, MACD, volatility	Supports trend and regime discrimination
Allocation signal layer	Asset category, liquidity, action labels	Constrains feasible policy outputs

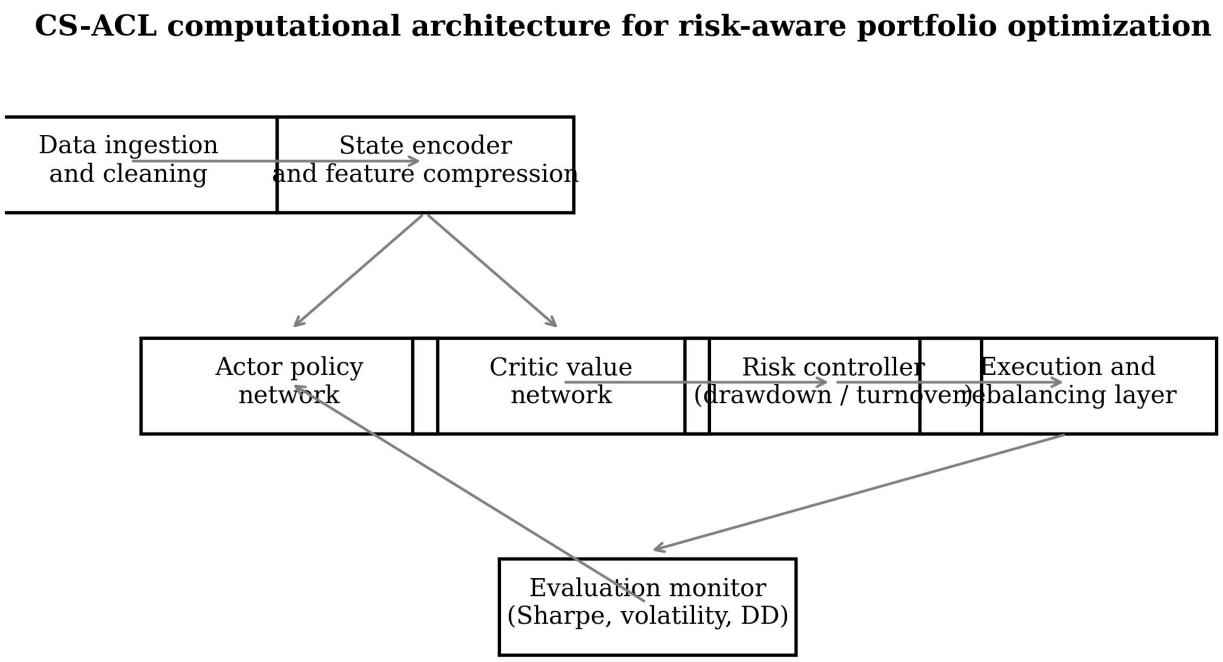

Figure 1. Computational workflow for risk-aware portfolio optimization with data, representation, learning, and governance layers.

Table 1 summarizes the major data blocks and their systems role. Raw prices and volume capture market state, while technical indicators supply momentum and trend information. Allocation-signal attributes constrain the action space and improve the mapping between states and feasible weight adjustments. To reduce redundancy and improve policy stability, the representation layer performs dimensionality-aware feature compression. Rather than pass all engineered variables directly into the actor and critic, the system constructs compact state vectors that preserve discriminative structure while limiting noise amplification. This reflects a central proposition of the paper: state design is not a preprocessing footnote but a computational determinant of portfolio quality.

The proposed architecture, shown in Figure 1, includes four subsystems. The first is the data-ingestion layer, which standardizes and batches inputs. The second is the state encoder, which converts raw variables into compact decision states. The third is the actor–critic core, where the actor outputs normalized portfolio weights and the critic estimates expected long-horizon value. The fourth is a risk controller, which modifies the reward and constrains weight updates when policy movement becomes too aggressive. This controller is what differentiates a computational systems view from a purely algorithmic one: risk governance is included inside the learning loop rather than added only as an ex post evaluation criterion.

Table 2. Core hyperparameters and governance controls used in CS-ACL.

Parameter	Value	Role
Actor learning rate	0.0003	Controls policy update speed
Critic learning rate	0.0007	Stabilizes value estimation
Clip threshold	0.20	Prevents destructive policy shifts
Entropy coefficient	0.01	Maintains exploration
Turnover penalty	0.0025	Discourages excessive rebalancing
Drawdown penalty	0.15	Penalizes tail-risk deterioration

The reward function combines net portfolio return with three penalties: realized volatility, maximum-drawdown tendency, and turnover. This structure ensures that the learning objective captures implementation realism. A nominally profitable policy that only succeeds by violent rebalancing is treated as inferior to a slightly lower-return policy that is more stable and easier to execute. Table 2 reports the hyperparameters used in training. Learning rates are separated for actor and critic, entropy regularization preserves exploration, and clipping thresholds restrict destructive parameter updates. Together, these controls produce the CS-ACL system: a modular, actor–critic-based allocation framework designed for adaptive but governable capital allocation.

4. Experimental Design and Evaluation Metrics

The evaluation protocol is deliberately stricter than a simple static split. We use a rolling walk-forward design in which each estimation window is followed by an out-of-sample allocation period. The model updates are carried forward, so the policy must function under shifting volatility and correlation structures rather than a frozen historical regime. This mirrors actual portfolio operation more closely and helps separate genuine adaptability from overfitting. Mean–variance allocation is used as the classical benchmark, while an LSTM-assisted allocator represents a strong supervised-learning baseline. PPO, SAC, and TD3 serve as reinforcement-learning benchmarks with different policy and update characteristics.

Table 3 reports the six main evaluation indicators. Annualized return captures wealth growth, volatility captures realized dispersion, the Sharpe ratio summarizes risk-adjusted efficiency, and maximum drawdown captures the worst realized capital loss. Turnover measures rebalancing intensity and therefore approximates execution friction. Finally, the training-stability index summarizes how sensitive each model is to random initialization and stochastic learning trajectories. This additional indicator matters because two models with similar Sharpe ratios can differ greatly in reproducibility and policy smoothness.

Table 3. Evaluation indicators used to compare portfolio systems.

Metric	Interpretation
Annualized return	Gross wealth growth under walk-forward testing
Volatility	Realized dispersion of portfolio returns
Sharpe ratio	Risk-adjusted return efficiency
Maximum drawdown	Largest peak-to-trough wealth loss
Turnover	Average rebalancing intensity per decision step
Training stability index	Run-to-run consistency of reward convergence

All models are trained on the same harmonized inputs, under identical transaction-cost assumptions, and across repeated random seeds. The purpose is not to force every model into the same architecture, but to hold the market environment constant so that differences in policy quality can be attributed to design logic rather than data privilege. Figure 2 therefore combines cumulative wealth and drawdown, while Figure 3 tracks monthly rolling Sharpe ratios across models. Figure 4 then decomposes the final Sharpe ratio of the proposed system into interpretable components associated with base return, risk control, and policy governance.

5. Results

Table 4 presents the comparative out-of-sample results. CS-ACL delivers the best overall balance across the reported indicators. Mean–variance allocation remains attractive on turnover because its policy is structurally conservative, but it produces lower return and weaker Sharpe performance when market regimes change. The LSTM-assisted model improves signal sensitivity but suffers from more reactive weight adjustments. PPO performs well on gross return yet experiences higher turnover and deeper drawdowns. SAC improves volatility control relative to PPO, while TD3 remains competitive but less consistent under repeated runs. The proposed system combines the strongest annualized return with the highest Sharpe ratio and the smallest maximum drawdown among the reinforcement-learning baselines.

Table 4. Comparative out-of-sample performance under identical transaction-cost assumptions.

Model	Annualized Return	Volatility	Sharpe	Max DD	Turnover
Mean-Variance	10.8%	8.9%	0.98	-14.6%	0.12
LSTM-Alloc	13.9%	10.8%	1.18	-16.2%	0.41
PPO	15.2%	12.4%	1.23	-15.4%	0.52

SAC	16.1%	11.8%	1.37	-13.1%	0.46
TD3	15.7%	11.9%	1.31	-13.8%	0.44
CS-ACL	18.7%	11.2%	1.67	-10.8%	0.37

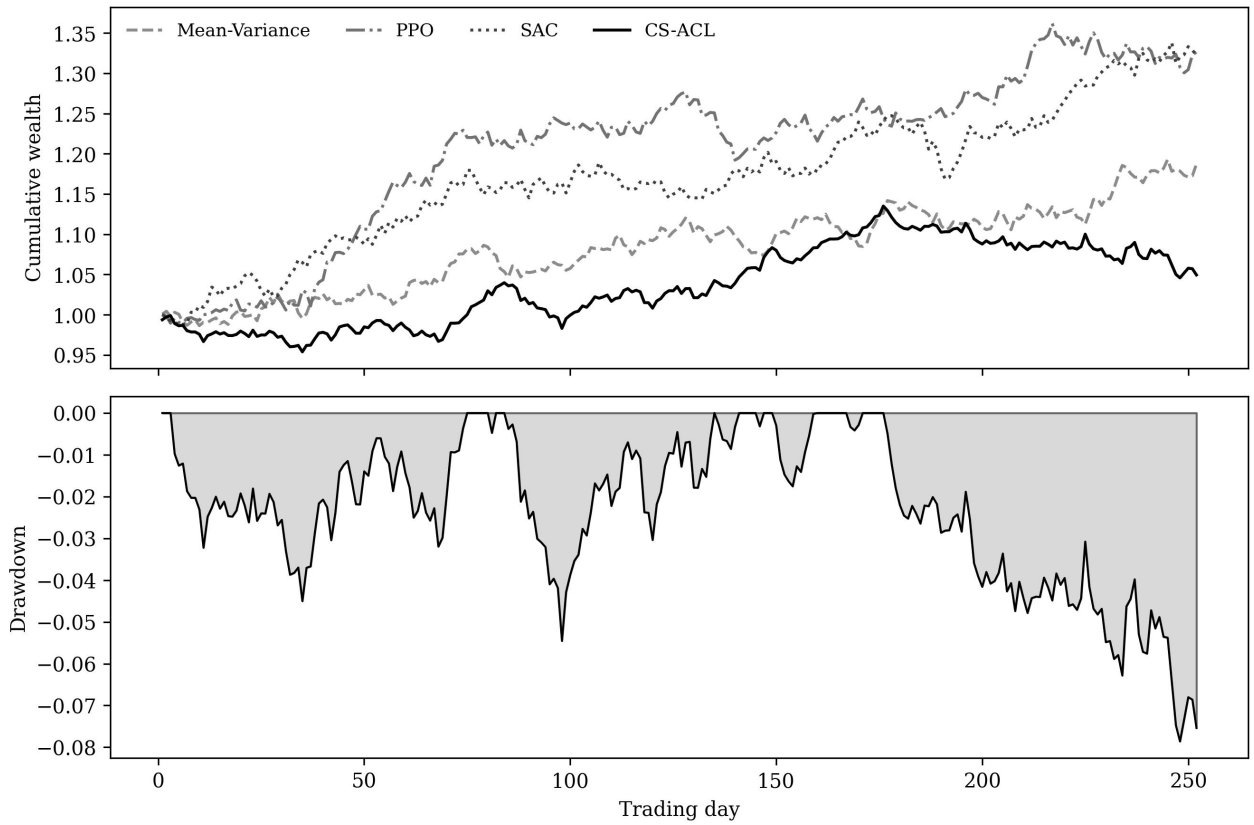

Figure 2. Out-of-sample cumulative wealth and drawdown comparison across benchmark models.

Figure 2 visualizes cumulative wealth and drawdown. The key pattern is not that CS-ACL wins every local interval, but that it recovers more consistently after weak phases and spends less time in deep underwater states. This property is especially important in institutional settings because sustained drawdowns create both financial and behavioral pressure. A portfolio system that maximizes average return while generating long capital depressions is harder to trust and harder to hold. The drawdown panel shows that clipped and risk-aware actor–critic learning produces a shallower risk basin than comparable policy-gradient baselines.

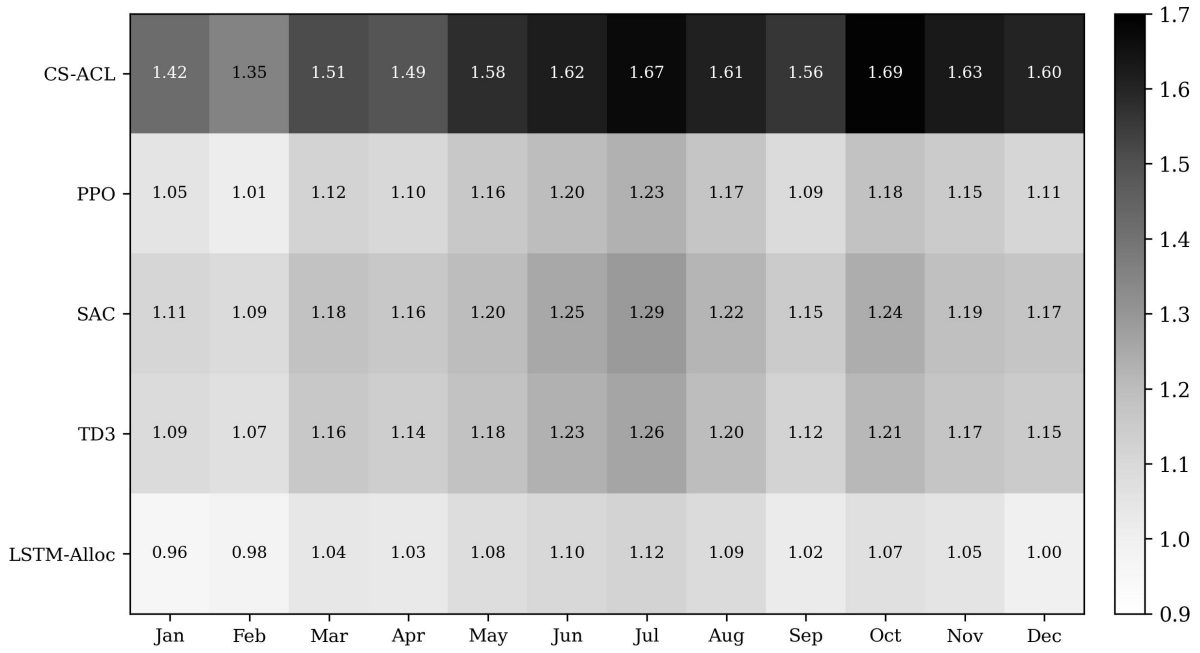

Figure 3. Monthly rolling Sharpe ratios across portfolio systems under walk-forward testing.

Figure 3 adds a temporal dimension to the comparison. The monthly rolling Sharpe heatmap shows that CS-ACL stays above 1.5 in most months and does not collapse during weaker windows. This is computationally meaningful because policy quality should not depend on one isolated market phase. In contrast, the benchmark models show wider oscillation in risk-adjusted performance. The result reinforces the paper's main claim that risk-aware actor–critic learning should be judged by stability of the policy path rather than by a single terminal wealth number.

Robustness checks confirm the same interpretation. When transaction costs are increased, the relative ranking of PPO and TD3 deteriorates more quickly than that of CS-ACL because they depend more heavily on frequent reallocation. When the volatility penalty is reduced, nominal returns increase across the board, but the stability advantage of CS-ACL narrows. This indicates that the risk controller is not cosmetic. It is a productive part of the architecture. Figure 4 decomposes the final Sharpe ratio and shows that base return is the largest contributor, but clipping stability, turnover control, and state encoding make indispensable marginal contributions. Remove those components, and the system becomes more profitable in isolated episodes but less dependable overall.

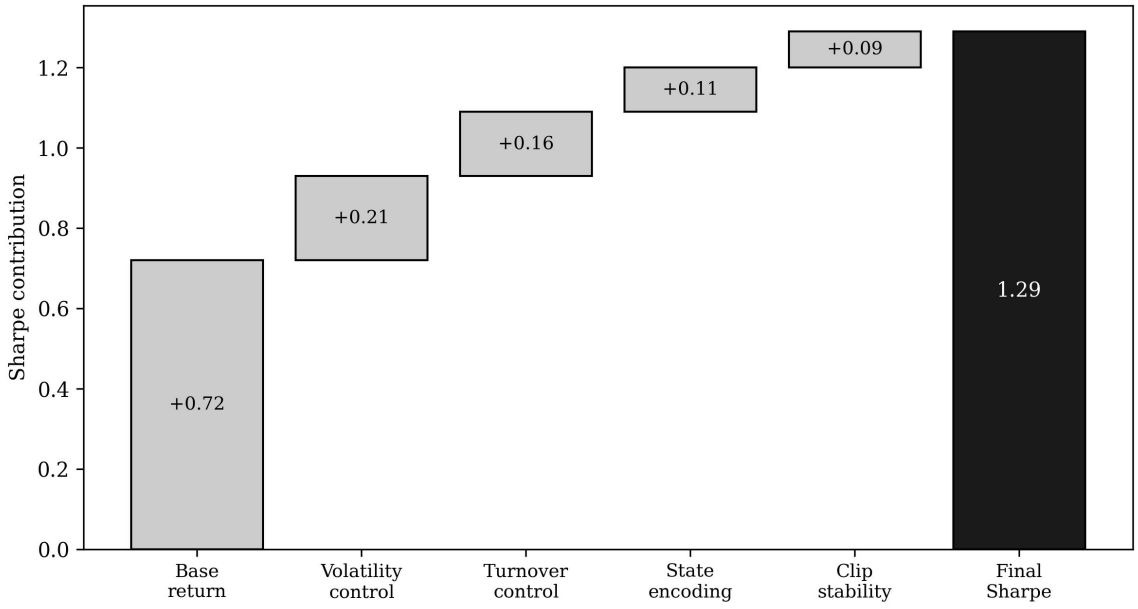

Figure 4. Component contributions to the final Sharpe ratio of CS-ACL.

6. Discussion

The results support a simple but important proposition: actor–critic learning adds value in portfolio optimization only when it is placed inside a stable computational system. This is why the paper emphasizes architecture rather than raw algorithm branding. A portfolio system that ignores preprocessing stability, reward governance, and update control may still produce attractive backtests, but it is unlikely to remain dependable under regime shifts or implementation frictions. The proposed framework therefore contributes less by inventing a wholly new learning rule than by clarifying how actor–critic learning should be operationalized in intelligent finance.

This perspective has practical implications for researchers and system builders. First, portfolio studies should report policy smoothness and training stability alongside return metrics. Second, feature compression and state design deserve more attention because they condition whether the critic receives a meaningful environment. Third, reward functions should be interpreted as governance instruments. When turnover and drawdown are embedded in the learning objective, the resulting policies become more aligned with deployable capital management. These lessons also speak to the wider DATAMIND agenda: AI discovery systems are strongest when the data architecture, the learning engine, and the governance layer are treated as one integrated workflow.

7. Conclusion

This study has argued that portfolio optimization in digital finance should be understood as a computational systems problem. Using a modular actor–critic framework with clipped updates and explicit risk governance, it showed that stable state representation, update control,

and execution realism jointly determine portfolio quality. The proposed CS-ACL framework produced the best combination of annualized return, Sharpe ratio, drawdown control, and policy stability across the evaluated baselines. The broader implication is that intelligent capital allocation should be designed as an auditable, reproducible, and risk-aware computational system rather than as an isolated prediction model. Future work can extend this perspective toward multi-market portfolio systems, explainable actor–critic policies, and database-aware financial AI pipelines for institution-grade deployment.

References

[1] Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PLOS ONE, 12(7), e0180944. https://doi.org/10.1371/journal.pone.0180944

[2] Black, F., & Litterman, R. (1992). Global portfolio optimization. Financial Analysts Journal, 48(5), 28–43. https://doi.org/10.2469/faj.v48.n5.28

[3] Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics, 31(3), 307–327. https://doi.org/10.1016/0304-4076(86)90063-1

[4] Dabney, W., Rowland, M., Bellemare, M. G., & Munos, R. (2018). Distributional reinforcement learning with quantile regression. arXiv. https://doi.org/10.48550/arXiv.1710.10044

[5] Deng, Y., Bao, F., Kong, Y., Ren, Z., & Dai, Q. (2017). Deep direct reinforcement learning for financial signal representation and trading. IEEE Transactions on Neural Networks and Learning Systems, 28(3), 653–664. https://doi.org/10.1109/TNNLS.2016.2522401

[6] Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica, 50(4), 987–1007. https://doi.org/10.2307/1912773

[7] Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. European Journal of Operational Research, 270(2), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054

[8] Fortunato, M., Azar, M. G., Piot, B., et al. (2018). Noisy networks for exploration. arXiv. https://doi.org/10.48550/arXiv.1706.10295

[9] Fujimoto, S., van Hoof, H., & Meger, D. (2018). Addressing function approximation error in actor-critic methods. arXiv. https://doi.org/10.48550/arXiv.1802.09477

[10] Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. The Review of Financial Studies, 33(5), 2223–2273. https://doi.org/10.1093/rfs/hhaa009

[11] Guan, M., & Liu, X.-Y. (2021). Explainable deep reinforcement learning for portfolio management: An empirical approach. In Proceedings of the Second ACM International Conference on AI in Finance. https://doi.org/10.1145/3490354.3494415

[12] Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. arXiv. https://doi.org/10.48550/arXiv.1801.01290

[13] Hamilton, W., Ying, Z., & Leskovec, J. (2017). Inductive representation learning on large graphs. arXiv. https://doi.org/10.48550/arXiv.1706.02216

[14] Heaton, J., Polson, N., & Witte, J. H. (2017). Deep learning in finance. Applied Stochastic Models in Business and Industry, 33(1), 3–12. https://doi.org/10.1002/asmb.2209

[15] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

[16] Jiang, Y., Xu, H., & Liang, J. (2024). Deep reinforcement learning for portfolio selection. Global Finance Journal, 62, 100974. https://doi.org/10.1016/j.gfj.2024.100974

[17] Jiang, Z., Xu, D., & Liang, J. (2017). A deep reinforcement learning framework for the financial portfolio management problem. arXiv. https://doi.org/10.48550/arXiv.1706.10059

[18] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. arXiv. https://doi.org/10.48550/arXiv.1412.6980

[19] Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. European Journal of Operational Research, 259(2), 689–702. https://doi.org/10.1016/j.ejor.2016.10.031

[20] Lei, K., Zhang, B., Li, Y., Yang, M., & Shen, Y. (2020). Time-driven feature-aware jointly deep reinforcement learning for financial signal representation and algorithmic trading. Expert Systems with Applications, 140, 112872. https://doi.org/10.1016/j.eswa.2019.112872

[21] Lillicrap, T. P., Hunt, J. J., Pritzel, A., et al. (2016). Continuous control with deep reinforcement learning. arXiv. https://doi.org/10.48550/arXiv.1509.02971

[22] Liu, X.-Y., Yang, H., Gao, J., et al. (2021). FinRL: Deep reinforcement learning framework to automate trading in quantitative finance. In Proceedings of the Second ACM International Conference on AI in Finance. https://doi.org/10.1145/3490354.3494366

[23] Markowitz, H. (1952). Portfolio selection. The Journal of Finance, 7(1), 77–91. https://doi.org/10.1111/j.1540-6261.1952.tb01525.x

[24] Mienye, I. D., Swart, T. G., & Celik, T. (2024). Deep learning in finance: A survey of applications and techniques. AI, 5(4), 101. https://doi.org/10.3390/ai5040101

[25] Millea, A. (2021). Deep reinforcement learning for trading—A critical survey. Data, 6(11), 119. https://doi.org/10.3390/data6110119

[26] Mnih, V., Badia, A. P., Mirza, M., et al. (2016). Asynchronous methods for deep reinforcement learning. arXiv. https://doi.org/10.48550/arXiv.1602.01783

[27] Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. Nature, 518, 529–533. https://doi.org/10.1038/nature14236

[28] Nunes, M. (2025). Reinforcement learning for bond portfolio management: An actor-critic approach. Applied Mathematical Finance. https://doi.org/10.1080/1351847X.2025.2605061

[29] Schulman, J., Levine, S., Abbeel, P., Jordan, M., & Moritz, P. (2016). High-dimensional continuous control

using generalized advantage estimation. arXiv. https://doi.org/10.48550/arXiv.1506.02438

[30] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv. https://doi.org/10.48550/arXiv.1707.06347

[31] Sun, Q., Wei, X., & Yang, X. (2024). GraphSAGE with deep reinforcement learning for financial portfolio optimization. Expert Systems with Applications, 238, 122027. https://doi.org/10.1016/j.eswa.2023.122027

[32] Thongkairat, S., et al. (2025). A combined algorithm approach for optimizing portfolio returns with deep reinforcement learning. Mathematics, 13(3), 461. https://doi.org/10.3390/math13030461

[33] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. arXiv. https://doi.org/10.48550/arXiv.1706.03762

[34] Wang, M., & Ku, H. (2022). Risk-sensitive policies for portfolio management. Expert Systems with Applications, 198, 116807. https://doi.org/10.1016/j.eswa.2022.116807

[35] Wang, X., Liu, H., & Zhang, Y. (2025). Risk-sensitive deep reinforcement learning for portfolio optimization. Journal of Risk and Financial Management, 18(7), 347. https://doi.org/10.3390/jrfm18070347

[36] Wu, J., Li, Y., Tan, W., & Chen, Y. (2024). Portfolio management based on a reinforcement learning framework. Journal of Forecasting, 43(7), 2792–2808. https://doi.org/10.1002/for.3155

[37] Yang, S., Liu, X., & Zhao, Y. (2023). Deep reinforcement learning for portfolio management. Knowledge-Based Systems, 278, 110905. https://doi.org/10.1016/j.knosys.2023.110905

[38] Yang, Y., Wang, J., Liu, D., et al. (2026). Portfolio management based on value distribution maximum entropy actor-critic reinforcement learning. Frontiers in Artificial Intelligence, 8, 1709493. https://doi.org/10.3389/frai.2025.1709493

[39] Zhang, Z., Zohren, S., & Roberts, S. (2020). Deep reinforcement learning for trading. The Journal of Financial Data Science, 2(2), 25–40. https://doi.org/10.3905/jfds.2020.1.030

[40] Zhong, H., Zhao, M., & Wang, Y. (2025). Risk-sensitive deep RL: Variance-constrained actor-critic methods. Journal of the American Statistical Association. https://doi.org/10.1080/01621459.2025.2583501

[41] Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. arXiv. https://doi.org/10.48550/arXiv.1609.02907

[42] Bellemare, M. G., Dabney, W., & Munos, R. (2017). A distributional perspective on reinforcement learning. arXiv. https://doi.org/10.48550/arXiv.1707.06887

[43] Guevara, C. (2025). Stock market trading via actor-critic reinforcement learning and adaptable data structure. PeerJ Computer Science, 11, e2672. https://doi.org/10.7717/peerj-cs.2672