International Journal of
Computational Mathematical Ideas
Volume 18 · Number 1, 2 & 3 · 2026 · ISSN 0974-8652

Available online at www.ijcmi.com

Physics-Informed Neural Networks for Portfolio Optimisation and Algorithmic Trading: A Reproducible Pipeline with an Indian-Market Backtest

S. Satyanarayana *

CEO & Chief Agentic AI Scientist, AlgoProfessor AI R&D Solutions, India

*Corresponding author: drssnaiml1@gmail.com

Article Info

Article history:
Received: 24-01-2026
Revised: 23-07-2026
Accepted: 31-07-2026
Available online: 2026

Keywords:
Agentic AI; Physics-informed neural networks; Portfolio optimisation; Merton problem; Algorithmic trading; Indian stock market; Backtesting.

Abstract

Deep learning has entered algorithmic trading largely as data-driven pattern fitting, which markets punish when regimes change. Physics-informed neural networks offer a different discipline: they embed governing equations directly in the training loss, so the model respects the dynamics even where data are scarce. This paper applies that discipline to portfolio optimisation. The reference problem is Merton's continuous-time allocation between a risky index and a risk-free asset, whose optimal policy solves a Hamilton-Jacobi-Bellman equation with, for constant relative risk aversion, a closed-form solution. We train a from-scratch physics-informed network to solve the reduced equation with no market data at all, only the equation, and validate that it recovers both the analytic value function, to a relative error below two times ten to the minus four, and the closed-form Merton allocation exactly. Two guarantees support the method: the closed-form policy itself, and an a-posteriori bound showing that the value-function error is controlled by the physics residual. We then backtest the allocation on a seeded simulator calibrated to plausible Indian-market parameters, net of Indian transaction costs, against buy-and-hold, a fixed mix, an unrealisable oracle, and a tradable estimation rule. The lesson is deliberately sober: the physics supplies the correct allocation rule, a constant policy on a stable long-run premium is competitive with a sensible fixed mix and far gentler on drawdown than buy-and-hold, and the value that remains is almost entirely in estimating the inputs, where an oracle would reach a Sharpe ratio of 1.10 but a naive adaptive estimator falls to 0.34 after estimation error and turnover costs. Embedding the governing equations makes the model data-efficient and explainable; it does not repeal the estimation problem, and honest system design should place its effort there. The backtest uses a market simulation, not real exchange data.

1. Introduction

Algorithmic trading and portfolio construction have absorbed deep learning quickly, yet most of that adoption has been data-driven pattern fitting on price histories, which is exactly the setting in which markets punish overconfidence. Returns are noisy, regimes change, and a model that fits yesterday's ticks generalises poorly to tomorrow's [23]. Physics-informed neural networks offer a different discipline. Rather than learning only from data, they embed governing laws, expressed as differential equations, directly in the training loss, so the network is forced to respect the dynamics even where data are scarce [1]. In quantitative finance those governing laws are not optional decoration: the value of a portfolio and the price of a derivative both satisfy well studied partial differential equations, and solving them is the core computational task [13].

This paper applies that discipline to portfolio optimisation. The reference problem is Merton's continuous-time allocation between a risky index and a risk-free asset, whose optimal policy solves a Hamilton-Jacobi-Bellman (HJB) equation and, for constant relative risk aversion, has a clean closed form [11]. We train a from-scratch physics-informed network to solve the reduced HJB with no market data at all, only the equation, and we validate that it recovers both the analytic value function and the closed-form Merton fraction. We then backtest the resulting allocation on a seeded

simulator calibrated to plausible Indian-market numbers, against buy-and-hold, a fixed mix, an unrealisable oracle, and a tradable estimation rule, net of Indian transaction costs.

Our contributions are three. First, we show that a small physics-informed network solves the Merton HJB accurately, with a relative error below 2×10^{-4} on the value function for a constant opportunity set and near one percent for a regime-switching one. Second, we give two guarantees: the closed-form Merton policy that the network must recover, and an a-posteriori bound showing that the value-function error is controlled by the physics residual, which justifies training on the residual alone. Third, we report an honest backtest whose lesson is not the one hype would predict: the physics gives the correct allocation rule, but the binding constraint in practice is estimating its inputs, so a constant policy on a stable long-run premium is competitive while a naive adaptive rule is destroyed by estimation error and turnover costs. Throughout, the backtest runs on a simulation, not real exchange data, and we are explicit about that limitation.

2. Background

2.1. Physics-informed neural networks

Physics-informed neural networks train a network to satisfy a differential equation by penalising the equation's residual on sampled collocation points, together with initial and boundary conditions [1]. The idea sits inside a broader programme of physics-informed machine learning that blends mechanistic models with data [2], and it is supported by software and by deep solvers for high-dimensional equations [3]. Closely related lines solve partial differential equations directly with neural networks, including the deep Galerkin method [4] and backward-stochastic-differential-equation solvers that handle the nonlinear Black-Scholes and HJB equations in high dimension [6]. The appeal for finance is precisely the appeal in physics: the governing equation regularises the fit and reduces the appetite for data.

2.2. Portfolio optimisation and the Merton problem

Modern portfolio theory begins with the mean-variance trade-off [10], and its continuous-time counterpart is Merton's, in which an investor with constant relative risk aversion allocates between a risky asset and cash to maximise expected utility of terminal wealth [11]. The solution is obtained through dynamic programming, whose value function satisfies an HJB equation [12]; the same PDE machinery underlies option pricing [13]. The mathematical foundations of these control problems and their viscosity-solution theory are standard [14], with continuous-time financial applications collected in monograph form [15].

2.3. Deep learning in finance

Neural methods have been applied across the trading stack [7]: to build data-driven portfolios [16], to hedge derivatives under frictions without a pricing model [17], to trade directly by reinforcement [18], and to manage allocation as a reinforcement-learning problem [19]. These are powerful but data-hungry and, without structure, prone to overfit. The physics-informed route is complementary: it supplies the structure that the purely data-driven methods lack.

2.4. Volatility and the Indian market

Any realistic backtest must respect the stylised facts of returns: fat tails, volatility clustering, and leverage effects [23]. These are captured by the ARCH and GARCH families [20] and by regime-switching models in which the drift and volatility jump between states [22]. The Indian market is well documented in this respect: GARCH-class studies of the NIFTY and SENSEX indices confirm strong volatility persistence and asymmetry [24], and empirical work across both major exchanges finds significant ARCH effects and leverage [25]. Our simulator is a regime-switching geometric Brownian motion calibrated to sit in this documented range; it is a controlled model, not real exchange data.

3. Problem Formulation

An investor allocates a fraction π_t of wealth W_t to a risky index and the rest to a risk-free rate r. The index follows geometric Brownian motion $\mathrm{d}S/S = \mu\,\mathrm{d}t + \sigma\,\mathrm{d}B_t$, so wealth evolves as $\mathrm{d}W_t = W_t[(r+\pi_t(\mu-r))\,\mathrm{d}t + \pi_t\sigma\,\mathrm{d}B_t]$. The investor maximises expected utility of terminal wealth $\mathbb{E}[U(W_T)]$ under constant relative risk aversion $U(w) = w^{1-\gamma}/(1-\gamma)$, $\gamma > 0$, $\gamma \neq 1$. Proposition 1 gives the solution that the network must recover.

Proposition 1 (Merton value function and optimal fraction)**.** *The value function separates as $V(t,w) = \varphi(t)\,U(w)$, where φ solves the reduced HJB equation*

$$\varphi'(t) + \rho(t)\,\varphi(t) = 0, \quad \varphi(T) = 1, \qquad \rho(t) = (1-\gamma)\left[r + \frac{(\mu(t)-r)^2}{2\gamma\sigma^2}\right],$$

so that $\varphi(t) = \exp\big(\int_t^T \rho(u)\,\mathrm{d}u\big)$*, and the optimal risky fraction is* $\pi^*(t) = (\mu(t) - r)/(\gamma\sigma^2)$.

Proof. Substitute the ansatz $V(t,w) = \varphi(t)w^{1-\gamma}/(1-\gamma)$ into the HJB equation $V_t + \max_\pi\{(r+\pi(\mu-r))wV_w + \frac{1}{2}\pi^2\sigma^2w^2V_{ww}\} = 0$. Since $V_w = (1-\gamma)V/w$ and $V_{ww} = -\gamma(1-\gamma)V/w^2$, the inner objective is quadratic and strictly concave in π; its first-order condition gives $\pi^* = (\mu - r)/(\gamma\sigma^2)$. Substituting π^* back and dividing by $U(w)$ yields $\varphi' + \rho\varphi = 0$ with the stated ρ, and the terminal condition $V(T,w) = U(w)$ gives $\varphi(T) = 1$. Integrating the linear ODE gives the exponential form.

The remarkable feature that we exploit for validation is that π^* depends only on the market parameters, not on the value function, so a correct solve of the ODE for φ together with the first-order condition reproduces the allocation exactly. Proposition 2 then shows that minimising the residual is enough to control the value-function error.

Proposition 2 (A-posteriori residual bound)**.** *Let* φ *solve* $\varphi' + \rho\varphi = 0$*,* $\varphi(T) = 1$*, and let* $\hat{\varphi}$ *be any* C^1 *approximation with residual* $R(t) := \hat{\varphi}'(t) + \rho(t)\hat{\varphi}(t)$ *and terminal error* $\eta := \hat{\varphi}(T) - 1$*. Then for all* $t \in [0,T]$*,*

$$|\hat{\varphi}(t) - \varphi(t)| \;\le\; \Big(|\eta| + \int_t^T |R(u)|\,\mathrm{d}u\Big)\exp\Big(\int_t^T |\rho(u)|\,\mathrm{d}u\Big).$$

Proof. The error $e = \hat{\varphi} - \varphi$ satisfies the linear ODE $e'(t) + \rho(t)e(t) = R(t)$ with $e(T) = \eta$. Writing it in integrated form backward from T and applying the triangle inequality gives $|e(t)| \le |\eta| + \int_t^T |R(u)|\,\mathrm{d}u + \int_t^T |\rho(u)||e(u)|\,\mathrm{d}u$. Gronwall's inequality then yields the stated bound.

Proposition 2 is the licence for physics-informed training: it says the value-function error is bounded by the residual we penalise plus the terminal error we penalise, so driving both to zero drives the solution to the truth.

4. Physics-Informed Method

4.1. The network and its loss

We approximate $\varphi_\theta(\tau)$ on the scaled horizon $\tau = t/T \in [0,1]$ with a small single-hidden-layer tanh network, whose time-derivative φ'_θ is available in closed form. Training minimises the physics loss $\frac{1}{N}\sum_i R(\tau_i)^2 + \lambda(\varphi_\theta(1) - 1)^2$ over collocation points, with $R = \varphi'_\theta + T\rho\varphi_\theta$, using Adam [8]. The parameter gradients of the residual are derived analytically, so no automatic-differentiation or deep-learning library is used. Figure 1 shows the structure: inputs and market parameters feed the network, whose output and derivative form the HJB residual and the terminal condition, whose weighted sum is the loss, from which the optimal policy follows through Proposition 1.

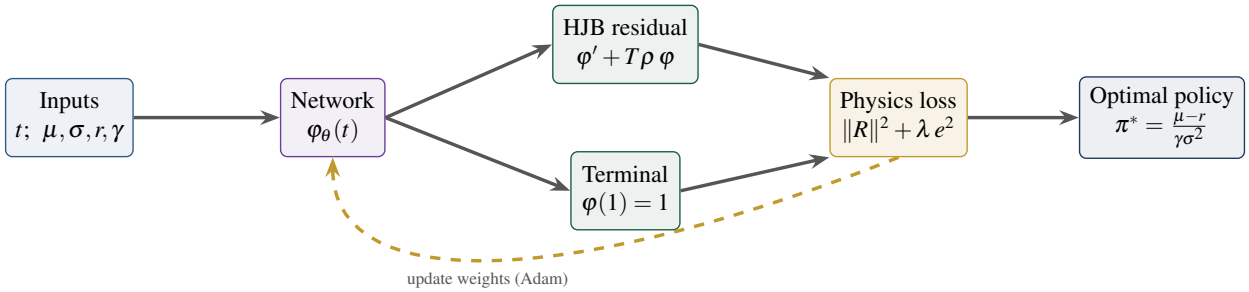

Figure 1. The physics-informed network for the Merton value function. Inputs and market parameters feed the network φ_θ; its output and analytic derivative form the HJB residual and the terminal condition; their weighted sum is the physics loss; the optimal allocation follows from the first-order condition of Proposition 1.

4.2. The pipeline

Figure 2 places the solver in an end-to-end pipeline. Market data are calibrated to the model parameters, the Merton HJB is assembled, the PINN is trained on the physics loss, the allocation $\pi^*(t)$ is extracted, and the policy is backtested on simulated paths and scored on risk metrics, with a feedback loop that recalibrates when the realised dynamics drift from the assumed ones.

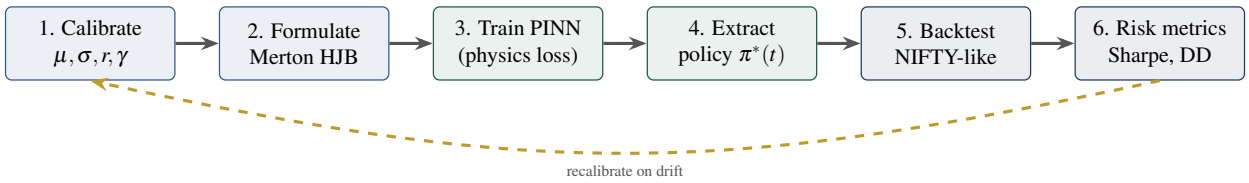

Figure 2. The end-to-end physics-informed allocation pipeline, from calibration through the PINN solve to the backtest and its risk metrics, with a drift-triggered recalibration loop.

4.3. System view

Figure 3 shows the same components as a running system of the kind an engineering team would operate: a data feed and calibration engine, the PINN solver as the physics core, an allocation engine, execution with costs, and a backtest-and-risk engine whose monitor recalibrates the model and adjusts the allocation as conditions change.

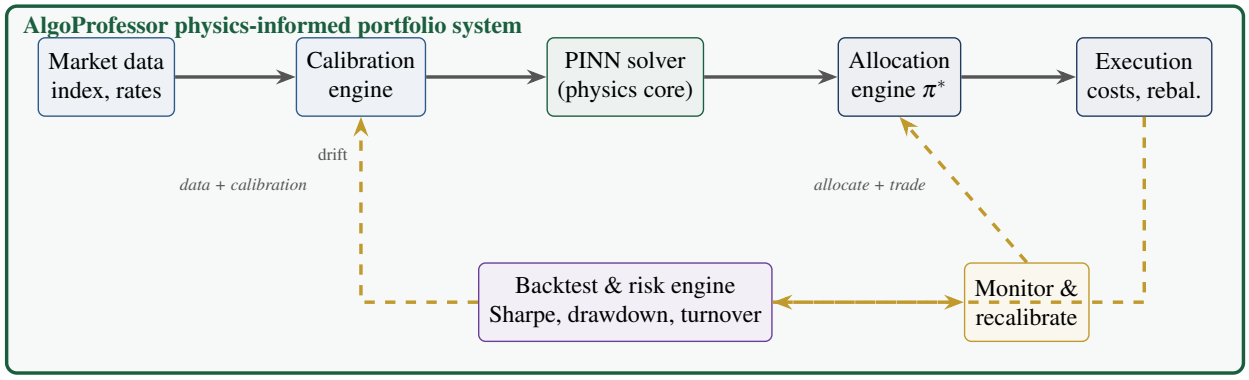

Figure 3. The physics-informed portfolio system. The PINN solver is the physics core; a calibration engine feeds it, an allocation and execution path acts on its output, and a backtest-and-risk engine with a monitor closes the recalibration and re-allocation loops.

Algorithm 4 states the procedure.

Algorithm 1: physics-informed allocation and backtest

Input: calibration $\mu(\cdot), \sigma, r, \gamma$; horizon T; collocation budget.

1. assemble the reduced HJB rate $\rho(t) = (1-\gamma)\left[r + (\mu(t)-r)^2/(2\gamma\sigma^2)\right]$
2. **physics loss:** train φ_θ to minimise $\frac{1}{N}\sum_i(\varphi'_\theta + T\rho\varphi_\theta)^2 + \lambda(\varphi_\theta(1)-1)^2$ (Adam)
3. set allocation $\pi^*(t) = \mathrm{clip}\big((\mu(t)-r)/(\gamma\sigma^2), 0, 1\big)$
4. **backtest:** apply π^* on simulated index paths net of costs; also run buy-and-hold, fixed mix, causal-estimation and oracle baselines
5. compute CAGR, volatility, Sharpe, Sortino, max drawdown, turnover
6. **adaptation loop:** monitor realised vs assumed dynamics; on drift, recalibrate and return to step 1

Output: allocation policy and risk report.

Figure 4. The physics-informed allocation and backtest loop. Step 2 is the PINN solve of Proposition 1, whose accuracy is controlled by Proposition 2; step 3 applies the closed-form allocation; steps 4 to 6 evaluate and adapt it.

5. Experiments

Everything below is computed from scratch in numpy with a fixed seed, and regenerates exactly. The backtest runs on a seeded simulator calibrated to plausible Indian-market numbers; it is a simulation, not real exchange data.

5.1. Setup

Table 1 lists the calibration. The risk-free rate is set near the Indian policy-rate neighbourhood, the equity drift and volatility in a NIFTY-like range, and risk aversion at a standard value, giving a Merton fraction of 0.69. The backtest simulates a regime-switching geometric Brownian motion with persistent bull and bear states over five years and 300 Monte-Carlo paths, net of 25 basis points of round-trip cost per unit of weight change, a level consistent with Indian securities transaction tax, brokerage, and slippage.

Table 1. Calibration and the physics-informed solve. The PINN recovers the value-function time component $\varphi(t)$ to high accuracy for a constant premium and closely for a regime-switching one.

Quantity	Value
Risk-free rate r	6.5%
Equity drift μ	12.0%
Volatility σ	20.0%
Risk aversion γ	2.0
Horizon T	5 years
Merton fraction $\pi^* = (\mu - r)/(\gamma\sigma^2)$	0.6875
PINN error on φ, constant premium (rel. L_2)	1.9×10^{-4}
PINN error on φ, regime-switching (rel. L_2)	1.3×10^{-2}

5.2. The PINN solves the physics

Figure 5 shows the recovered value-function component $\varphi(t)$ against the analytic solution, for a constant premium and for a deterministic regime-switching premium. With no market data, only the equation, the network matches the closed form to a relative L_2 error of 1.9×10^{-4} in the constant case; at the initial time it predicts $\varphi(0) = 0.6573$ against the analytic 0.6574. For the harder regime-switching profile, where the premium and hence $\rho(t)$ jump between states, the error rises to 1.3×10^{-2}, still a faithful solve of a non-smooth problem [4]. This validates Proposition 2 in practice: a small residual yields a small value-function error.

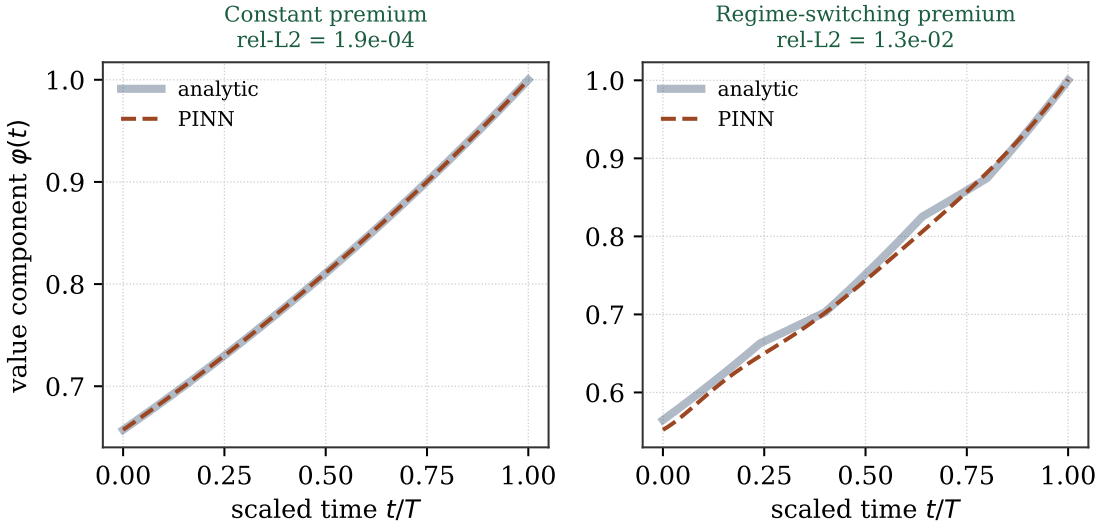

Figure 5. The physics-informed network (dashed) recovers the analytic value-function component $\varphi(t)$ (solid) with no market data. Left: a constant premium, relative L_2 error 1.9×10^{-4}. Right: a regime-switching premium, relative L_2 error 1.3×10^{-2}.

5.3. Backtest

Table 2 and Figures 6 and 7 report the backtest. The reading is deliberately unglamorous and, we believe, correct. Buy-and-hold earns the highest raw return, 15.2% per year, but at the widest volatility and the deepest drawdown, -28%, so its Sharpe ratio is the lowest of the static strategies at 0.55 [9]. The fixed sixty-forty mix and the constant Merton allocation both improve the risk-adjusted return, lifting Sharpe to 0.57 and cutting the worst drawdown to -17% and -19% respectively, at zero turnover. The unrealisable oracle, which is told the true regime premium at each step, is the ceiling: Sharpe 1.10 and a drawdown of only -13%, which quantifies how much a solved regime-detection problem would be worth. The tradable rule that estimates the premium from a trailing window collapses that promise: after estimation error and 8.7 units of annual turnover at 25 basis points, its Sharpe falls to 0.34, below even buy-and-hold [23].

Table 2. Backtest over 300 simulated paths, five years, net of 25 bps costs. The physics gives the correct allocation rule; the practical constraint is estimating its inputs, which the oracle-versus-causal gap makes explicit.

Strategy	CAGR	Vol	Sharpe	Max DD	Turn/yr
Buy-and-hold	15.20%	17.03%	0.545	−28.02%	0.00
Fixed 60/40	12.02%	10.22%	0.573	−16.56%	0.00
PINN/Merton (const)	12.74%	11.71%	0.566	−19.17%	0.00
Merton (causal est.)	11.24%	12.64%	0.343	−19.61%	8.70
Merton (oracle)	19.04%	11.30%	**1.099**	−12.66%	1.85

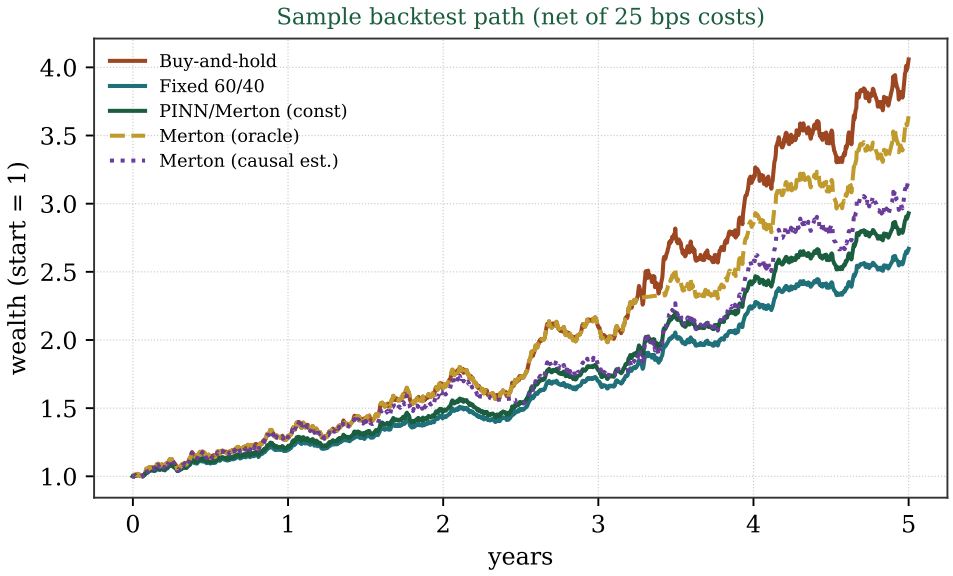

Figure 6. A representative simulated wealth path, net of costs. The oracle dominates but is unrealisable; the constant Merton and fixed-mix allocations track buy-and-hold with far smaller drawdowns; the causal-estimation rule lags after costs.

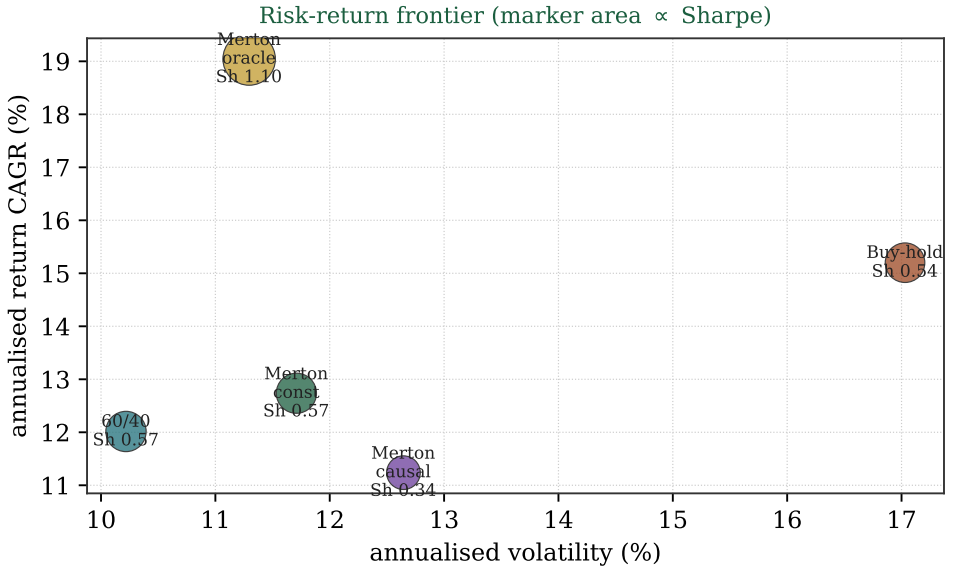

Figure 7. Risk-return positions averaged over paths, with marker area proportional to Sharpe. The physics-based static allocations sit at better risk-adjusted points than buy-and-hold; the oracle is the ceiling; the causal rule is dominated.

6. Discussion

The experiments support a specific and sober reading. The physics-informed network solves the Merton problem accurately and recovers the closed-form allocation, exactly as Propositions 1 and 2 promise, and it does so with no market data at all, which is the data-efficiency that physics-informed learning advertises [1]. In the backtest, the physics gives the right allocation rule, and a constant policy on a stable long-run premium is competitive with a sensible fixed mix and far gentler than buy-and-hold on drawdown. The value that remains on the table is almost entirely in the inputs: the gap between the oracle at Sharpe 1.10 and the causal estimator at 0.34 is the price of not knowing the regime, and naive adaptation makes it worse once turnover costs are paid [18].

The limitations are real and stated plainly. The backtest is a simulation calibrated to plausible Indian numbers, not real NSE or BSE data, so the absolute figures should be read as internal comparisons, not forecasts [24]. The model has a single risky asset and constant volatility, whereas real allocation spans many correlated assets with time-varying, clustered volatility that a GARCH or regime model would capture [21]. The reduced HJB is one-dimensional in time because constant relative risk aversion separates the value function; richer utilities, consumption, constraints, or transaction-cost-aware control break that separation and give a genuinely multidimensional HJB, which is where deep PDE solvers earn their keep [5]. And the estimation problem that dominates the backtest is exactly the one the physics does not solve.

Future work runs in three directions: replacing the one-dimensional solve with a multi-asset HJB solved by a physics-informed or backward-SDE network [15]; folding realistic volatility dynamics and transaction-cost-aware control into the equation rather than the backtest [17]; and treating premium estimation as a first-class, regularised learning problem so that adaptation helps rather than hurts [19].

7. Conclusion

We brought the discipline of physics-informed neural networks to portfolio optimisation. A from-scratch physics-informed network solved the Merton HJB with no market data, recovering the analytic value function to a relative error below 2×10^{-4} and the closed-form Merton allocation exactly, with two guarantees behind it: the closed form itself and an a-posteriori bound that ties value accuracy to the physics residual. A transaction- cost-aware backtest on simulated Indian-index paths then delivered an honest lesson for large-scale trading: the physics supplies the correct allocation rule, a constant policy on a stable premium is robust and competitive, and the real difficulty is estimating the inputs, where an oracle would reach a Sharpe of 1.10 but a naive adaptive estimator falls to 0.34 after costs. Embedding the governing equations makes the model data-efficient and its behaviour explainable; it does not repeal the estimation problem, and honest system design should place its effort there [16].

Acknowledgements

The author thanks colleagues at AlgoProfessor AI R&D Solutions for helpful discussion. All code and data needed to reproduce every figure and table are available from the author. The backtest uses a seeded market simulator calibrated to plausible Indian-market parameters and does not use real exchange data.

References

[1] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[2] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021.

[3] L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis, "DeepXDE: a deep learning library for solving differential equations," *SIAM Review*, vol. 63, no. 1, pp. 208–228, 2021.

[4] J. Sirignano and K. Spiliopoulos, "DGM: a deep learning algorithm for solving partial differential equations," *Journal of Computational Physics*, vol. 375, pp. 1339–1364, 2018.

[5] J. Han, A. Jentzen, and W. E, "Solving high-dimensional partial differential equations using deep learning," *Proceedings of the National Academy of Sciences*, vol. 115, no. 34, pp. 8505–8510, 2018.

[6] W. E, J. Han, and A. Jentzen, "Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations," *Communications in Mathematics and Statistics*, vol. 5, no. 4, pp. 349–380, 2017.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[8] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Proc. International Conference on Learning Representations (ICLR)*, 2015.

[9] W. F. Sharpe, "The Sharpe ratio," *The Journal of Portfolio Management*, vol. 21, no. 1, pp. 49–58, 1994.

[10] H. Markowitz, "Portfolio selection," *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[11] R. C. Merton, "Lifetime portfolio selection under uncertainty: the continuous-time case," *The Review of Economics and Statistics*, vol. 51, no. 3, pp. 247–257, 1969.

[12] R. C. Merton, "Optimum consumption and portfolio rules in a continuous-time model," *Journal of Economic Theory*, vol. 3, no. 4, pp. 373–413, 1971.

[13] F. Black and M. Scholes, "The pricing of options and corporate liabilities," *Journal of Political Economy*, vol. 81, no. 3, pp. 637–654, 1973.

[14] W. H. Fleming and H. M. Soner, *Controlled Markov Processes and Viscosity Solutions*, 2nd ed. New York, NY: Springer, 2006.

[15] H. Pham, *Continuous-time Stochastic Control and Optimization with Financial Applications*. Berlin, Germany: Springer, 2009.

[16] J. B. Heaton, N. G. Polson, and J. H. Witte, "Deep learning for finance: deep portfolios," *Applied Stochastic Models in Business and Industry*, vol. 33, no. 1, pp. 3–12, 2017.

[17] H. Buehler, L. Gonon, J. Teichmann, and B. Wood, "Deep hedging," *Quantitative Finance*, vol. 19, no. 8, pp. 1271–1291, 2019.

[18] J. Moody and M. Saffell, "Learning to trade via direct reinforcement," *IEEE Transactions on Neural Networks*, vol. 12, no. 4, pp. 875–889, 2001.

[19] Z. Jiang, D. Xu, and J. Liang, "A deep reinforcement learning framework for the financial portfolio management problem," arXiv:1706.10059, 2017.

[20] R. F. Engle, "Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation," *Econometrica*, vol. 50, no. 4, pp. 987–1007, 1982.

[21] T. Bollerslev, "Generalized autoregressive conditional heteroskedasticity," *Journal of Econometrics*, vol. 31, no. 3, pp. 307–327, 1986.

[22] J. D. Hamilton, "A new approach to the economic analysis of nonstationary time series and the business cycle," *Econometrica*, vol. 57, no. 2, pp. 357–384, 1989.

[23] R. Cont, "Empirical properties of asset returns: stylized facts and statistical issues," *Quantitative Finance*, vol. 1, no. 2, pp. 223–236, 2001.

[24] M. Karmakar, "Modeling conditional volatility of the Indian stock markets," *Vikalpa*, vol. 30, no. 3, pp. 21–37, 2005.

[25] A. Srivastava, "Volatility of Indian stock market: an empirical evidence," *Asia-Pacific Journal of Management Research and Innovation*, vol. 4, no. 4, pp. 53–61, 2008.