Delta Hedging Optimization with Reinforcement Learning

Jens Cancio Youssef Chelaifa Szymon Ścibior

April 2026

Abstract

We implement a DDPG-inspired hybrid actor-critic agent for dynamic delta hedging of European call options under proportional transaction costs. Following the methodology of Hull et al. (2021), we train the agent with a risk-adjusted episodic objective and show that it reduces the hedging cost measure $Y(0) = \mathbb{E}[C] + 1.5\,\sigma(C)$ by 27–33% relative to classical Black-Scholes delta hedging and by 31% relative to the Whalley-Wilmott asymptotically optimal strategy on the GBM benchmark. A transaction-cost sweep on GBM further shows that the RL policy is dominated by BS delta in the frictionless limit but becomes increasingly advantageous as trading frictions rise. Beyond replicating Hull et al., we study transaction-cost sensitivity, cross-model robustness, and ensemble-based uncertainty analysis. We evaluate cross-model robustness by training agents on one price dynamics (GBM, Heston, Merton) and testing on all three, finding that performance is very similar across training environments, with only a slight edge for Heston-trained agents in some settings. We also apply deep ensembles as an uncertainty proxy, demonstrating that ensemble disagreement concentrates at-the-money near expiry, precisely where gamma risk is highest, and that diverse architectures converge to near-identical hedging policies, supporting the robustness of the learned strategy.

1 Introduction

In a frictionless, continuous-time world, Black-Scholes delta hedging perfectly replicates a European option payoff. In practice, every rebalance incurs transaction costs, and the number of hedging opportunities is finite. The classical delta hedger faces a dilemma: rebalance frequently to track the theoretical delta at high cost, or rebalance infrequently and accept hedging error.

Recent work has framed this trade-off as a sequential decision problem and applied deep reinforcement learning (RL) to learn cost-aware hedging strategies. Buehler et al. [2] introduced *deep hedging*, a direct end-to-end approach that backpropagates through entire price trajectories to minimize a convex risk measure. Hull et al. [3] showed that a DDPG agent trained with a mean–variance objective can reduce hedging costs by approximately 31% compared to Black-Scholes delta hedging under proportional transaction costs. More recently, Brini et al. [1] proposed a quadratic-variation penalty that penalises P&L variability along individual paths.

Our contributions are threefold:

1. We replicate Hull et al.'s risk-sensitive actor-critic approach and show that the episodic objective $\mathbb{E}[C] + \lambda\,\sigma(C)$ is essential for good hedging performance; a mean-focused actor objective can encourage under-hedging because it does not penalise dispersion in terminal cost (Section 3.4).

2. We run cross-model generalisation experiments, training on one stochastic process and testing on all three, and find that the choice of training environment matters only modestly, with Heston training showing a slight edge in some settings (Section 5.3).

3. We apply deep ensembles [4] to RL hedging agents and show that ensemble disagreement provides a meaningful uncertainty proxy correlated with known regions of hedging difficulty (Section 5.4).

2 Background and Related Work

Classical delta hedging. A market maker who sells a European call option hedges by holding δ shares of the underlying, where $\delta = \partial V/\partial S$ is the Black-Scholes delta. As the stock price moves, δ changes and the position must be rebalanced. With proportional transaction costs $\kappa > 0$, each rebalance incurs a cost $\kappa\,|S\,\Delta h|$, where Δh is the change in the hedge ratio.

Whalley-Wilmott asymptotic hedging. Whalley and Wilmott [8] derived the asymptotically optimal hedging strategy under small proportional transaction costs. Rather than rebalancing to the Black-Scholes delta at every step, the agent maintains a no-trade band around the delta:

$$h_t \in \big[\delta_t - H_t,\ \delta_t + H_t\big], \qquad H_t = \left(\tfrac{3}{2}\,\kappa\,\Gamma_t^2\,S_t^2\right)^{1/3}, \tag{1}$$

where Γ_t is the Black-Scholes gamma. The agent only rebalances when the current position drifts outside this band, and then moves to the nearest edge. This reduces transaction costs while accepting a small amount of additional hedging error. We use Whalley-Wilmott as a stronger classical baseline alongside BS delta. Because the derivation is asymptotic and GBM-based, we treat it as a heuristic reference when reporting results on Heston and Merton paths.

Deep hedging. Buehler et al. [2] proposed an end-to-end deep hedging approach: simulate full price trajectories, parametrise the hedging strategy with a neural network, and optimize the policy directly by backpropagating a risk measure computed from the resulting P&L. The open-source library PFHedge [6] implements this framework. Unlike our hybrid actor-critic setup, PFHedge's standard training pipeline does not learn a separate critic or value function alongside the hedging policy.

RL hedging. Hull et al. [3] cast hedging as a Markov Decision Process and trained a DDPG agent with a risk-adjusted objective $\mathbb{E}[C] + \lambda\,\sigma(C)$. Their agent reduces hedging costs by approximately 31% on simulated GBM paths. Our work replicates and extends their approach to stochastic volatility and jump-diffusion environments, and adds ensemble-based uncertainty analysis.

Uncertainty in deep hedging. Poddar [7] recently applied deep ensembles to the supervised deep hedging setting. We bring this idea into the RL framework, training ensembles of DDPG agents with diverse architectures and analysing their disagreement as a function of moneyness and time to expiry.

3 Methodology

3.1 Market Simulation

We train and evaluate on three stochastic processes, each adding realism.

Geometric Brownian Motion (GBM). The standard baseline with constant drift μ and volatility σ:

$$dS = \mu\, S\, dt + \sigma\, S\, dW. \tag{2}$$

Heston stochastic volatility. Volatility evolves as a mean-reverting CIR process:

$$dS = \mu\, S\, dt + \sqrt{v}\, S\, dW_1, \qquad dv = \kappa_v(\theta - v)\, dt + \xi\, \sqrt{v}\, dW_2, \qquad \langle dW_1, dW_2 \rangle = \rho\, dt. \tag{3}$$

We use parameters $v_0 = \theta = 0.04$, $\kappa_v = 2.0$, $\xi = 0.3$, $\rho = -0.7$.

Merton jump-diffusion. Adds a compound Poisson jump component:

$$dS = \mu\, S\, dt + \sigma\, S\, dW + S\,(e^J - 1)\, dN, \tag{4}$$

where N is a Poisson process with intensity $\lambda = 2.0$ and $J \sim \mathcal{N}(\mu_J, \sigma_J^2)$ with $\mu_J = -0.02$, $\sigma_J = 0.05$.

Even in the Heston and Merton experiments, the hedger itself still uses a Black-Scholes-style representation: the state contains a single constant volatility input σ, and the short option leg is repriced with the Black-Scholes formula. Our goal is therefore not model-consistent pricing under stochastic volatility or jumps, but to test how robust the learned hedging rule remains when the path dynamics differ from the pricing/state approximation.

3.2 MDP Formulation

We hedge a short European call option with strike K and maturity T over $N = 21$ discrete rebalancing steps (trading days).

Component	**Definition**
State s_t	$\big(\log(S_t/K),\ T - t\,\Delta t,\ \sigma,\ h_{t-1}\big) \in \mathbb{R}^4$
Action a_t	Hedge ratio $h_t = \pi_\theta(s_t) \in [0, 1]$
Step reward r_t	$\Delta V_t + h_t\, \Delta S_t - \kappa\, \lvert S_{t+1}\,(h_t - h_{t-1})\rvert$
Total cost C	$-\sum_{t=0}^{N-1} r_t$
Objective	Minimize $Y(0) = \mathbb{E}[C] + \lambda\, \sigma(C)$ with $\lambda = 1.5$

Table 1: MDP components. The state includes log-moneyness, time to expiry, volatility, and the previous hedge ratio. The reward is the per-step P&L of the hedged portfolio after transaction costs.

The state representation follows Hull et al.: log-moneyness $\log(S_t/K)$ captures the option's intrinsic value sensitivity, time to expiry τ_t captures the urgency to hedge, volatility σ provides the risk level, and the previous hedge h_{t-1} is needed to compute transaction costs. The action is the hedge ratio produced by a sigmoid output, naturally bounded in $[0, 1]$.

3.3 DDPG-Inspired Actor-Critic Architecture

Our agent adopts a DDPG-inspired actor-critic architecture [5].

Actor π_θ: a multi-layer perceptron (MLP) mapping the 4-dimensional state to a single hedge ratio. The architecture is $4 \to 64 \to 64 \to 64 \to 64 \to 1$ with ReLU activations and a final sigmoid layer to constrain output to $[0, 1]$.

Critic Q_ϕ: an MLP mapping $(s_t, a_t) \in \mathbb{R}^5$ to a scalar Q-value, estimating the expected future reward from a given state-action pair. The critic uses the same hidden architecture as the actor.

Both networks maintain slowly-updated target copies $\pi_{\theta'}$ and $Q_{\phi'}$ via Polyak averaging with $\tau_{\text{soft}} = 0.002$.

3.4 Risk-Sensitive Training via Differentiable Rollout

Why the risk-adjusted objective matters. In textbook DDPG, the actor is updated to maximize the critic's Q-value, $\max_\theta \mathbb{E}[Q_\phi(s, \pi_\theta(s))]$, which aligns the policy with expected reward. For hedging, however, expected reward alone is not the right target: a policy that trades too little can look acceptable on mean cost because stock gains and option losses partially offset on average, while still producing undesirable dispersion and tail risk. We therefore optimize an explicitly risk-adjusted episodic objective rather than a mean-only control objective.

Hull's risk-adjusted objective. To penalise variance, we follow Hull et al. and train the actor with an episodic objective:

$$\mathcal{L}(\theta) = \mathbb{E}[C] + \lambda\, \sigma(C), \qquad \lambda = 1.5, \tag{5}$$

where C is the total hedging cost over a full episode. Computing this loss requires rolling out the actor over all N time steps and maintaining the gradient chain throughout. If the gradient is broken between steps, the actor cannot learn how its current action affects future transaction costs.

Differentiable rollout. We generate M fresh price paths from the simulator and roll out the actor policy in PyTorch with gradient tracking enabled. At each step t:

1. Construct the state $s_t = (\log(S_t/K),\ \tau_t,\ \sigma,\ h_{t-1})$.
2. Compute $h_t = \pi_\theta(s_t)$ *with gradient.*
3. Compute the step reward r_t including transaction cost $\kappa\,|S_{t+1}(h_t - h_{t-1})|$.
4. Set $h_{t-1} \leftarrow h_t$ (the gradient flows through this assignment).

After N steps, the total cost $C = -\sum r_t$ is computed per path. The loss (5) is evaluated over all M paths and backpropagated through the entire trajectory. This is the key implementation detail: *the gradient must flow through* h_{t-1} from step to step, connecting each action to all future transaction costs.

Critic training. The critic is updated via standard TD learning on transitions collected with exploration noise (ϵ-schedule from 0.15 decaying to 0.005). While the critic is not used for the actor update, it provides a learned value function and preserves the actor-critic structure. The resulting method is therefore best described as a *hybrid* DDPG-inspired actor-critic: DDPG-style networks, target updates, and TD critic training combined with a directly optimized episodic actor loss. For brevity, we still refer to the trained policy as "DDPG" in the results tables and figures.

3.5 Deep Ensembles for Uncertainty Analysis

A single trained agent provides no measure of confidence in its hedge recommendations. Following Lakshminarayanan et al. [4], we train an ensemble of $M = 4$ DDPG agents with diverse architectures and hyperparameters:

Agent	Architecture	Learning rate	Noise	Parameters
1	64-64-64-64	5×10^{-4}	0.15	8,769
2	128-128-128	3×10^{-4}	0.12	33,537
3	32-32-32-32-32	8×10^{-4}	0.18	3,361
4	256-128-64	2×10^{-4}	0.10	42,241

Each agent is trained independently with a different random seed. At test time, the *disagreement* at step t and path i is

$$d_t^{(i)} = \mathrm{std}\big(h_t^{(1,i)},\ h_t^{(2,i)},\ h_t^{(3,i)},\ h_t^{(4,i)}\big), \tag{6}$$

where $h_t^{(m,i)}$ is the hedge ratio recommended by agent m. This provides a per-step, per-path confidence estimate without modifying the training procedure.

4 Experimental Setup

All experiments use a European call option with $S_0 = K = 100$, $T = 1/12$ (one month), $r = 0$, $\sigma = 0.20$, $\mu = 0.05$, $\kappa = 0.01$ (1% proportional transaction costs), and $N = 21$ daily rebalancing steps. Options are priced with the Black-Scholes formula throughout, and the state keeps a single volatility input even for Heston and Merton paths. This isolates the effect of the hedging strategy from the pricing model, but it also means our stochastic-volatility and jump-diffusion experiments are robustness tests under intentional model misspecification.

Training uses 2,000 episodes for cross-environment experiments and 10,000 episodes for single-model runs. Each episode generates 512 paths for critic data collection and 1,024 paths for the actor's differentiable rollout. Evaluation is performed on 50,000 fresh paths with a fixed seed.

The primary evaluation metric is Hull's cost measure $Y(0) = \mathbb{E}[C] + 1.5\,\sigma(C)$, which can be interpreted as the risk-adjusted hedging cost measure. We also report the mean cost, standard deviation, and CVaR at the 95% level.

5 Results

5.1 Single-Agent Performance

Table 2 compares the DDPG agent trained for 10,000 episodes on GBM against both classical baselines on 50,000 evaluation paths.

Strategy	Mean	Std	$Y(0)$	CVaR 95%
BS Delta	2.57	0.94	3.98	4.78
Whalley-Wilmott	1.57	1.51	3.83	4.65
DDPG Agent	1.56	0.73	**2.66**	**3.21**
DDPG vs BS	−39%	−22%	−33%	−33%
DDPG vs WW	−1%	−52%	−31%	−31%

Table 2: DDPG vs. classical baselines on GBM paths. The DDPG agent reduces $Y(0)$ by 33% vs. BS delta and 31% vs. Whalley-Wilmott. All metrics are computed on 50,000 out-of-sample paths.

The 33% reduction in $Y(0)$ vs. BS delta is consistent with Hull et al.'s reported 31% [3]. The comparison with Whalley-Wilmott is particularly instructive: WW achieves a similar mean cost to DDPG (1.57 vs. 1.56) by reducing unnecessary trades, but at the cost of substantially higher variance (std 1.51 vs. 0.73). The DDPG agent matches WW's mean cost while simultaneously controlling variance. It learns *both* when not to trade (like WW's no-trade band) and *how* to trade when it does (adjusting hedge ratios to minimise future rebalancing costs).

Figure 1 shows the P&L distributions for the ensemble mean policy. Its distribution is shifted left (lower cost) and has a lighter right tail, confirming improvements in both average and worst-case performance.

Figure 2 shows the average hedge ratio over time for the ensemble agents compared to BS delta. All four architectures learn a characteristic hump-shaped policy: the agent over-hedges relative to BS delta in the early and middle periods (holding more stock to reduce variance), then converges toward the BS delta near expiry as the option's delta becomes increasingly binary. The systematic deviation from BS delta is how the agent reduces costs, by front-loading hedging activity when rebalancing is cheap (delta moves slowly) and avoiding expensive rebalancing near expiry (when gamma is high).

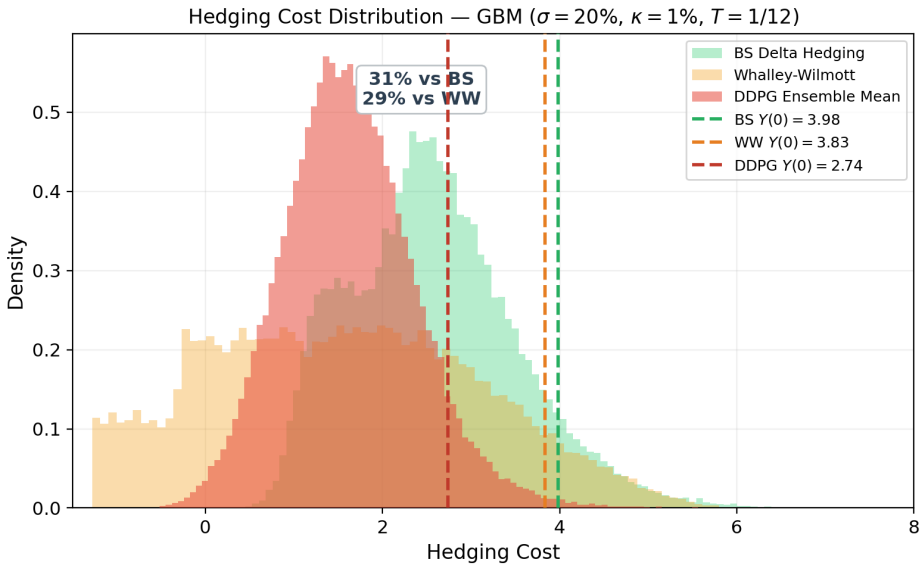

Figure 1: Hedging cost distributions for BS delta hedging (green), Whalley-Wilmott (orange), and the DDPG ensemble mean (red) on 50,000 GBM paths. Dashed lines indicate $Y(0)$. Relative to both classical baselines, the learned policy shifts the distribution toward lower costs and a lighter right tail.

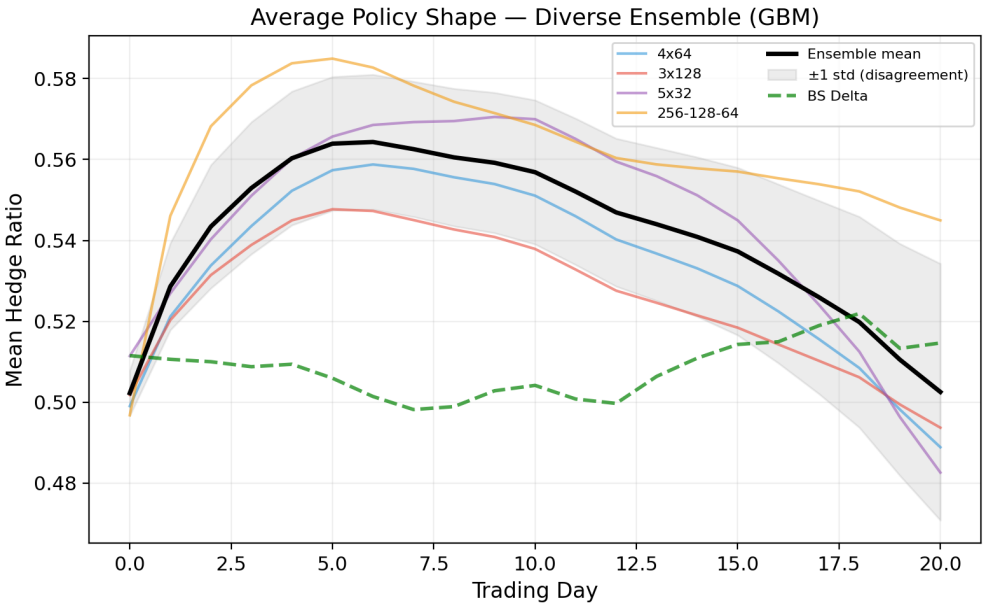

Figure 2: Average hedge ratio over time for the four ensemble agents, the ensemble mean (black), and BS delta (green dashed). All architectures learn a similar hump-shaped policy that deviates systematically from BS delta. The grey band shows ±1 standard deviation of disagreement across agents.

5.2 Transaction Cost Sensitivity

We next vary the proportional transaction cost κ to see how the relative ranking of the strategies changes with market friction. For each $\kappa \in \{0, 0.5\%, 1\%, 2\%, 5\%\}$, we retrain a fresh GBM agent for 1,000 episodes and evaluate BS delta, Whalley-Wilmott, and the RL policy on 20,000 matched GBM paths. To keep this sweep internally consistent, all three strategies are scored under the same environment accounting used by the RL evaluator.

Figure 3 shows three clear patterns. First, in the frictionless limit, BS delta and Whalley-Wilmott coincide and slightly outperform the learned policy ($Y(0) = 0.646$ vs. 0.747). Second, once transaction costs are introduced, the RL policy becomes the best of the three strategies at every non-zero κ we tested. Third, the advantage widens sharply with friction: relative to BS delta, the improvement grows from about 12% at $\kappa = 0.5\%$ to 26% at $\kappa = 1\%$, 40% at $\kappa = 2\%$, and 67% at $\kappa = 5\%$. In other words, the RL policy is not universally better than classical hedging, but it becomes increasingly attractive exactly in the regime where transaction costs matter most.

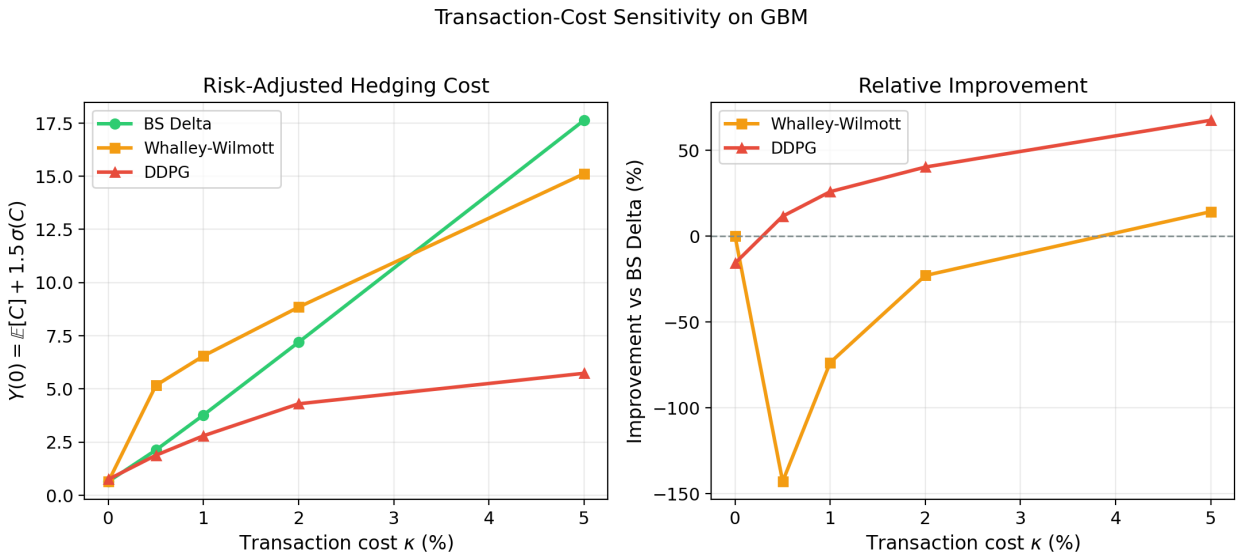

Figure 3: Transaction-cost sensitivity on GBM. For each κ, a fresh DDPG-inspired agent is retrained for 1,000 episodes and then evaluated alongside BS delta and Whalley-Wilmott on 20,000 matched GBM paths. Left: absolute risk-adjusted hedging cost $Y(0)$. Right: percentage improvement relative to BS delta. The learned policy is slightly worse in the frictionless limit but dominates both classical baselines once trading frictions become material.

5.3 Cross-Model Generalisation

To test robustness to model misspecification, we train three agents, one on each of GBM, Heston, and Merton, and evaluate all three on all three environments (Table 3).

	Test Environment		
Strategy	GBM	Heston	Merton
Train: GBM	**2.719**	2.785	3.184
Train: Heston	2.763	**2.744**	**3.170**
Train: Merton	2.758	2.767	3.172
Whalley-Wilmott	3.834	3.792	4.119
BS Delta	3.766	3.791	4.172

Table 3: Cross-model generalisation results ($Y(0)$, lower is better). Each RL agent is trained for 2,000 episodes on one environment and evaluated on all three. Bold indicates the best RL agent for each test environment. All RL agents outperform both classical baselines by 25–29%.

Figure 4 visualises the full cross-evaluation matrix. Three findings stand out:

1. **All RL agents dominate the classical baselines everywhere.** The main gap in the figure is not between GBM-, Heston-, and Merton-trained agents, but between any RL policy and BS delta. Across all three test environments, the RL agents are roughly one full point lower in $Y(0)$ than BS delta, and Table 3 shows that they also outperform Whalley-Wilmott.

2. **Cross-environment degradation is small.** The gap between the best and worst RL entries within a given test environment is only 0.019–0.041 in $Y(0)$, suggesting the agents learn a general cost-aware strategy rather than overfitting to one model's dynamics.

3. **The choice of training model matters only modestly.** Heston training has a slight edge in two of the three columns, but the differences among the three RL rows are small enough that the more robust conclusion is simply that the learned hedging policy transfers well across these simulated dynamics.

All cross-environment numbers reported here come from single training runs. Since the gaps among the RL variants are small, we treat the ranking within the RL block as descriptive rather than statistically definitive.

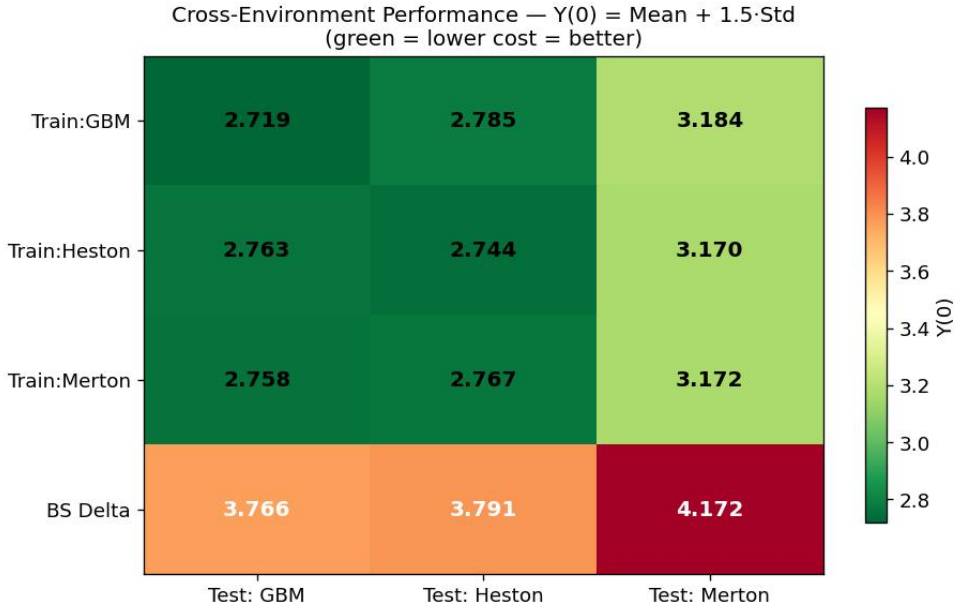

Figure 4: Cross-model generalisation matrix for the three RL agents and the BS delta baseline. Each cell shows $Y(0)$ for an agent trained on the row environment and tested on the column environment. The dominant visual pattern is that all three RL-trained policies perform very similarly to one another, while all of them substantially outperform BS delta across the three test environments. Whalley-Wilmott is omitted from the heatmap for visual clarity and reported separately in Table 3.

5.4 Ensemble Robustness and Uncertainty

We train ensembles of four diverse agents (Table in Section 3.5) on each of the three environments. Table 4 summarises the results.

	GBM	Heston	Merton
Ensemble $Y(0)$	2.78	2.76	3.21
BS Delta $Y(0)$	3.77	3.79	4.17
Improvement	27%	28%	23%
Mean disagreement	0.019	0.012	0.022

Table 4: Ensemble performance across environments. Mean disagreement is the average standard deviation of hedge ratios across the four agents, computed over all paths and time steps.

Policy convergence. Despite spanning a 12× range in parameter count (3,361 to 42,241), all four architectures converge to near-identical hedging policies. Mean disagreement is below 0.022 in all environments. This indicates that the learned hedge is a robust property of the training objective, not an artifact of a specific network design.

Where agents disagree. Figure 5 shows ensemble disagreement as a function of log-moneyness at days 11 and 21 (expiry). Disagreement peaks sharply at-the-money ($\log(S/K) \approx 0$) and increases toward expiry. This matches financial intuition: at-the-money options near expiry have the highest gamma (delta changes most rapidly with price), making the optimal hedge most sensitive to small modelling differences. The disagreement signal could be used as a per-step confidence proxy.

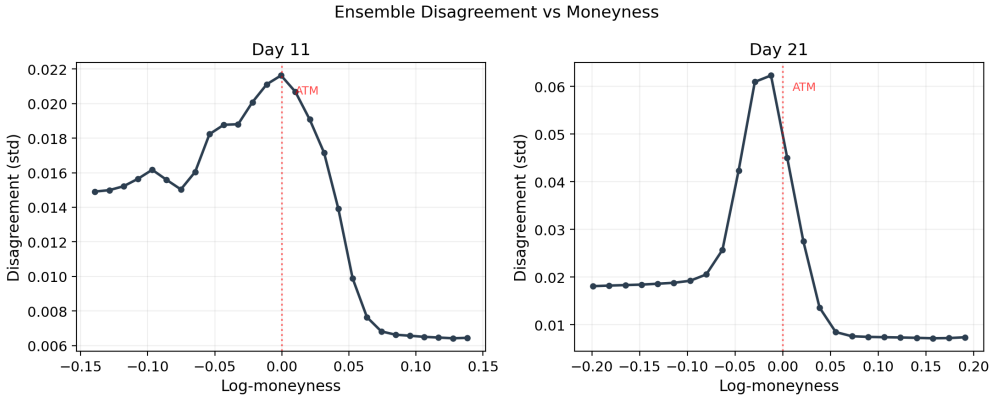

Figure 5: Ensemble disagreement vs. log-moneyness at days 11 and 21. Disagreement peaks at-the-money and grows substantially near expiry, consistent with the gamma risk profile of European calls.

6 Discussion and Future Work

Current limitations. Our agent uses a single risk measure (mean + 1.5 × std). Different risk measures (CVaR, entropic risk, quadratic variation) may produce agents with meaningfully different hedging styles, and a systematic comparison is warranted. We also report single-run RL results throughout, so small differences between closely performing agents should not be over-interpreted without repeated seeds and confidence intervals. Additionally, all experiments use simulated data; validation on historical option data remains future work.

SAC agent. We plan to implement the Soft Actor-Critic (SAC) algorithm, which uses a stochastic policy and entropy regularisation. SAC is expected to provide better exploration and more stable

convergence than DDPG, particularly in the higher-dimensional environments required for multi-asset hedging.

Mixed-environment training. Another direction we plan to explore is training a single agent on a mixture of GBM, Heston, and Merton episodes rather than on one simulator at a time. That would test whether exposure to multiple dynamics during training produces a more robust hedging rule and whether mixed training can match the strongest single-environment agents across all test settings.

Uncertainty-aware hedging. Ensemble disagreement is a useful diagnostic, but we have not yet built it directly into the trading policy. One direction we plan to explore is a new agent with a 5-dimensional state that includes disagreement as an additional input. That would let the policy learn how to respond when the ensemble is less confident, for example by trading more cautiously or reducing position changes in ambiguous states.

Broader transaction cost sensitivity. Figure 3 shows that the RL policy becomes increasingly advantageous as trading frictions rise on GBM. Extending that analysis to Heston and Merton paths, larger training budgets at each κ, and alternative risk measures remains future work.

7 Conclusion

We have implemented a DDPG-inspired hybrid actor-critic agent for delta hedging that reduces the risk-adjusted hedging cost by 27–33% compared to BS delta hedging and 28–31% compared to the Whalley-Wilmott asymptotically optimal strategy across three simulation environments. The comparison with WW is particularly informative: the DDPG agent matches WW's mean cost while achieving dramatically lower variance, demonstrating that the RL agent learns both *when* to avoid trading and *how* to position optimally when it does trade. The transaction-cost sweep further shows that the learned policy is slightly inferior in the frictionless limit but becomes increasingly advantageous as trading frictions rise. The agent generalises well across model specifications, and the differences between GBM-, Heston-, and Merton-trained policies remain small across the cross-environment tests. Deep ensembles confirm that the learned policy is robust to architectural choices and provide an interpretable proxy for policy uncertainty. These results demonstrate that reinforcement learning offers a practical improvement over both naive and optimised classical hedging strategies in the presence of transaction costs, while the cross-model results suggest that the learned policy captures a genuinely robust hedging heuristic rather than a narrow fit to one simulated environment.

References

[1] Brini, A., Domeniconi, G., and Fathi, A. Data-driven derivative hedging with quadratic variation penalty. In *Proceedings of ICAIF '24*, 2024.

[2] Bühler, H., Gonon, L., Teichmann, J., and Wood, B. Deep hedging. *Quantitative Finance*, 19(8):1271–1291, 2019.

[3] Hull, J., Cao, J., Chen, J., and Poulos, Z. Deep hedging of derivatives using reinforcement learning. *arXiv:2103.16409*, 2021.

[4] Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *NeurIPS*, 2017.

[5] Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., and Wierstra, D. Continuous control with deep reinforcement learning. *ICLR*, 2016.

[6] PFHedge – PyTorch-based deep hedging framework. github.com/pfnet-research/pfhedge

[7] Poddar, M. Uncertainty-aware deep hedging. *arXiv:2603.10137*, 2026.

[8] Whalley, A.E. and Wilmott, P. An asymptotic analysis of an optimal hedging model for option pricing with transaction costs. *Mathematical Finance*, 7(3):307–324, 1997.