FinFlowRL: An Imitation-Reinforcement Learning Framework for Adaptive Stochastic Control in Finance

Yang Li[1] **Zhi Chen**[1] **Steve Y. Yang**[1] **Ruixun Zhang**[2]

[1]School of Business, Stevens Institute of Technology
[2]School of Mathematical Sciences, Peking University

{yli269, zchen100, syang14}@stevens.edu
zhangruixun@pku.edu.cn

Abstract

Traditional stochastic control methods in finance rely on simplifying assumptions that often fail in real-world markets. While these methods work well in specific, well-defined scenarios, they underperform when market conditions change. We introduce **FinFlowRL**, a novel framework for financial stochastic control that combines imitation learning with reinforcement learning. The framework first pre-trains an adaptive meta-policy by learning from multiple expert strategies, then fine-tunes it through reinforcement learning in the noise space to optimize the generation process. By employing action chunking—generating sequences of actions rather than single decisions—it addresses the non-Markovian nature of financial markets. FinFlowRL consistently outperforms individually optimized experts across diverse market conditions.

1 Introduction

Stochastic control in finance addresses optimal decision-making under uncertainty, fundamental to high-frequency trading, optimal execution (Almgren and Chriss, 2001), and portfolio optimization (Merton, 1969). Traditional approaches formulate the problem as an objective function governed by a stochastic differential equation (SDE) with constraints. While tractable under stylized assumptions, these models face key limitations in practice. They rely on parameter calibration from historical data, which fails under regime shifts, and often assume stationary dynamics such as geometric Brownian motion (Black and Scholes, 1973). Real markets exhibit jumps, stochastic volatility, and non-stationary behavior, making such assumptions unrealistic and solutions suboptimal. Moreover, framing decisions as a Markov Decision Process (MDP) ignores the path-dependent and memory-rich nature of financial markets (Gatheral et al., 2022).

We propose FinFlowRL, a two-stage framework for financial stochastic control. First, we pre-train a flow-matching model that learns from multiple expert strategies across diverse market scenarios, integrating their strengths into a unified policy. Second, we fine-tune via reinforcement learning by optimizing the noise generation process rather than actions directly—the pre-trained model remains frozen while we learn to generate better input noise. Both stages employ action chunking (Chi et al., 2024; Li et al., 2025), generating sequences of decisions over planning horizons rather than single actions, naturally capturing the non-Markovian dynamics of financial markets.

To our knowledge, this is the first framework combining flow matching and RL for financial stochastic control. Applied to high-frequency trading, FinFlowRL consistently outperforms individual expert models and adapts effectively to changing market conditions.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: .

2 Methodology

FinFlowRL employs a two-stage approach: (1) pre-training a MeanFlow model based on expert demonstrations, and (2) fine-tuning via reinforcement learning in noise space while keeping the expert frozen.

2.1 Stage 1: MeanFlow Pre-training

Expert Demonstration Generation. We simulate 108 market scenarios varying volatility ($\sigma \in \{0.05, 0.1, 0.3\}$), order arrival rates ($\lambda \in \{10, 20, 40\}$), and jump intensities. For each scenario, we evaluate four experts—Avellaneda-Stoikov (AS) (Avellaneda and Stoikov, 2008), GLFT (Guéant et al., 2013), modified GLFT with drift (Guéant et al., 2013), and PPO (Schulman et al., 2017)—selecting the best performer's actions as demonstrations, yielding 3.24M state-action pairs.

MeanFlow Model. Unlike standard flow matching that models instantaneous velocity $v(z_t, t)$, MeanFlow (Geng et al., 2025) models average velocity between time steps:

$$u(z_t, r, t) = \frac{1}{t-r} \int_r^t v(z_\tau, \tau) d\tau$$

The key insight is the MeanFlow Identity, which connects average and instantaneous velocities:

$$u(z_t, r, t) = v(z_t, t) - (t-r) \frac{d}{dt} u(z_t, r, t)$$

This relationship enables training without access to the true instantaneous velocity. During training, we construct interpolants between noise $z_0 \sim \mathcal{N}(0, I)$ and expert actions a_{expert}:

$$z_t = (1-t)z_0 + t \cdot a_{expert}$$

The training objective minimizes:

$$\mathcal{L}(\theta) = \mathbb{E}_{t,r,z_t,s} \left[\| u_\theta(z_t, r, t, s) - \text{sg}(u_{target}) \|^2 \right]$$

where $u_{target} = v_t - (t-r)(v_t \partial_z u_\theta + \partial_t u_\theta)$ and $v_t = a_{expert} - z_0$ is the straight-line velocity.

Conditioning and Generation. We condition on market state s using FiLM (Perez et al., 2018), which modulates network features via $\mathbf{h}' = \boldsymbol{\gamma}(s) \odot \mathbf{h} + \boldsymbol{\beta}(s)$. This enables state-dependent action generation through one-step inference:

$$a = z_1 - u_\theta(z_1, 0, 1, s)$$

where we set $r = 0, t = 1$ for generation. This one-step formulation is crucial for meeting HFT's microsecond latency requirements.

2.2 Stage 2: FlowRL Fine-tuning

Instead of retraining the entire model, we freeze the pre-trained MeanFlow expert g_θ and learn only a noise policy π_ϕ^W that generates optimal input noise. Following (Lv et al., 2025; Wagenmaker et al., 2025), we transform the MDP from action space to noise space:

$$w \sim \pi_\phi^W(\cdot|s), \quad a = g_\theta(s, w)$$

The noise policy is a Gaussian $\pi_\phi^W(w|s) = \mathcal{N}(\mu_\phi(s), \Sigma_\phi)$ optimized via PPO with clipped objective:

$$L^{PPO} = \mathbb{E}_t[\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t)]$$

This approach reduces trainable parameters by 84% while leveraging pre-trained knowledge.

2.3 Action Chunking

Both stages generate action sequences rather than single decisions. Given observation window $T_{obs} = 2$, we predict $T_{pred} = 8$ future actions but execute only $T_{exec} = 4$. This hierarchical planning captures temporal dependencies and market memory, directly addressing non-Markovian dynamics inherent in financial markets.

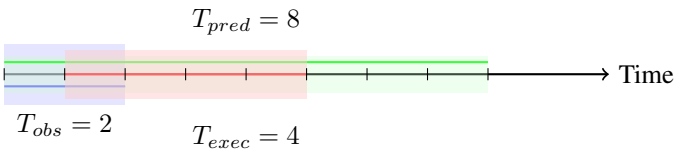

Figure 1: Hierarchical temporal structure in MeanFlow-PPO. The model observes T_{obs} past states, generates actions for T_{pred} future steps, but only executes the first T_{exec} actions before replanning.

3 Application in High-Frequency Trading

3.1 Problem Formulation

We formulate high-frequency trading market-making as a stochastic control problem over discrete time steps $t \in \{0, \ldots, T\}$. At each step, an agent observes state O_t (including market data and private information like inventory) and takes action A_t (typically setting bid and ask spreads (δ_t^b, δ_t^a)).

The goal is to learn an optimal policy $\pi(O_t) = A_t$ that maximizes expected terminal wealth minus inventory risk:

$$\max_{\pi} \mathbb{E}^{\pi} \left[W_T - \phi(I_T) \mid O_0 \right]$$

where W_T is terminal wealth and $\phi(I_T)$ penalizes unsold inventory.

3.2 Generating Market Observation-Action Pairs

We model mid-price S_t as a jump-diffusion process (Merton, 1976):

$$dS_t = S_{t^-} \left(\mu dt + \sigma dB_H(t) \right) + S_{t^-} (e^J - 1) dN_t$$

where μ is drift, σ is volatility, $dB_H(t)$ is fractional Brownian motion (Mandelbrot and Van Ness, 1968), $J \sim N(\mu_J, \sigma_J^2)$ represents jump size, and dN_t is a Poisson process with intensity λ_J.

Order arrivals follow mutually exciting Hawkes processes (Bacry et al., 2015; Hawkes, 1971), capturing self-exciting (previous buy/sell orders increase subsequent same-type arrivals) and cross-exciting effects (buy orders influence sell arrivals and vice versa).

Buy and sell order intensities are:

$$\lambda_a(t) = \mu_a + \sum_{t_i \in \mathcal{N}_a} \alpha_{aa} e^{-\beta(t - t_i)} + \sum_{t_j \in \mathcal{N}_b} \alpha_{ab} e^{-\beta(t - t_j)} \tag{1}$$

$$\lambda_b(t) = \mu_b + \sum_{t_i \in \mathcal{N}_b} \alpha_{bb} e^{-\beta(t - t_i)} + \sum_{t_j \in \mathcal{N}_a} \alpha_{ba} e^{-\beta(t - t_j)} \tag{2}$$

We create market scenarios with varying liquidity levels (high, medium, low) and stress conditions featuring sudden changes and increased volatility.

Our expert candidates include: (1) Avellaneda-Stoikov (AS) model, (2) Guéant-Lehalle-Fernandez-Tapia (GLFT) model, (3) modified GLFT with price drift, and (4) PPO-trained RL agent (Schulman et al., 2017). We generate actions from each agent for each market scenario, collecting 3.24 million state-action pairs.

3.3 Results

Our investigation is designed to critically evaluate FinFlowRL's advantages over existing approaches and its capabilities to earn profit. Specifically, we address the following research questions: **RQ1:** Can FinFlowRL effectively generalize strategies learned from expert demonstrations to new, unseen market conditions? **RQ2:** Does the fine-tuning mechanism improve the performance of actions initially proposed by the pre-trained model? **RQ3:** Can the FinFlowRL framework achieve greater profitability than traditional strategies?

To evaluate FinFlowRL's performance across a spectrum of out-of-sample market environments, we systematically configured distinct test conditions by setting key market parameters to differ from

those used during training. We differentiated the market microstructure by creating four specific situations based on combinations of market volatility (Vol) and overall order arrival rate (AR): (1) High Vol/High AR (HH), representing active, potentially news-driven volatile markets; (2) High Vol/Low AR (HL), mimicking risky market; (3) Low Vol/High AR (LH), reflecting stable, liquid markets; and (4) Low Vol/Low AR (LL), simulating quiet market periods. For each of these market situations, we perform traditional strategies as well as FinFlowRL to compare their performance.

We take the following evaluation metrics. (1) Profit and Loss (PnL). Measure the total change in the value over a specific period, reflecting the aggregate percentage gain or loss. (2) Sharpe Ratio (SR). It helps investors understand how much excess return an investment generated for each unit of risk it undertook. A higher Sharpe Ratio generally indicates a better risk-adjusted performance. (3) Maximum Drawdown (MDD) Quantify the largest percentage decline in the value of a strategy from a previous peak to a subsequent trough.

Table 1: Performance comparison across market conditions. Parameters: Hurst Exponent $H = 0.5$, Drift Rate $\mu = 0$, volatility (σ: $\{0.02, 0.25\}$) and arrival rate (λ: $\{25, 50\}$). Each method evaluated over 1 million trials. PnL: Profit and Loss, SR: Sharpe Ratio, MDD: Maximum Drawdown (%).

	High Volatility & High Demand			High Volatility & Low Demand			Low Volatility & High Demand			Low Volatility & Low Demand		
	PnL ↑	SR ↑	MDD ↓	PnL ↑	SR ↑	MDD ↓	PnL ↑	SR ↑	MDD ↓	PnL ↑	SR ↑	MDD ↓
Random Action	1.99	0.06	28.49	0.99	0.04	19.24	2.10	0.31	2.71	1.08	0.22	1.87
AS	24.22	0.09	241.65	13.54	0.09	125.78	25.20	1.05	7.66	13.67	0.72	6.61
GLFT	25.10	0.37	60.57	13.56	0.24	52.55	25.87	1.17	6.95	13.91	0.78	6.14
GLFT-drift	25.10	0.37	60.57	13.56	0.24	52.55	25.87	1.17	6.95	13.91	0.78	6.14
Vanilla PPO	14.76	0.10	133.61	9.29	0.08	103.85	26.74	0.81	10.13	19.80	0.46	14.56
Pretrained MeanFlow	23.91	0.37	43.4	12.97	0.22	45.47	23.82	1.83	2.18	12.93	1.07	2.69
FinFlowRL	26.33	0.50	45.47	14.32	0.28	45.35	26.27	2.34	2.68	14.29	1.36	3.08

Table 1 presents comprehensive results based on 1 million evaluation trials per method to ensure statistical significance. Across all tested market conditions, from high to low volatility, FINFLOWRL consistently demonstrates superior performance. It achieves the highest risk-adjusted returns (Sharpe Ratios) and maintains lower maximum drawdowns compared to both traditional models and standard reinforcement learning (PPO) approaches. For **RQ1**, the initial pre-trained model successfully learns from the collection of experts, achieving performance comparable to its best instructor (the GLFT model). This proves the imitation learning stage effectively internalizes expert strategies. For **RQ2**, The fine-tuned model significantly outperforms all baseline methods, including the experts it learned from and the initial pre-trained model. This demonstrates that the reinforcement learning stage successfully improves upon the expert knowledge to discover superior, adaptive strategies tailored to specific market conditions. For **RQ3**, our model consistently achieves the highest profit. FINFLOWRL shows remarkable stability, especially in volatile markets with sudden price jumps. This is attributed to its use of "action chunking"—generating sequences of actions—which provides a longer-term planning perspective and mitigates the risk of compounding errors.

4 Conclusions

We introduced FinFlowRL, a novel framework for financial stochastic control that combines imitation learning with reinforcement learning. The framework first learns from multiple expert strategies through a MeanFlow policy, then fine-tunes by optimizing noise generation rather than actions directly.

Our main contributions include: (1) first application of flow matching to financial stochastic control, achieving higher Sharpe ratios and lower maximum drawdowns than traditional methods; (2) efficient architecture using frozen experts with learnable noise policies, reducing trainable parameters by 84%; (3) action chunking mechanism that addresses non-Markovian market dynamics

The simulation results demonstrate FinFlowRL's superiority across diverse market conditions. The pre-trained model successfully learns expert strategies, while fine-tuning discovers policies that outperform individual experts. Action chunking proves particularly effective during market jumps, mitigating error compounding inherent in single-step decisions.

FinFlowRL demonstrates that combining expert knowledge with adaptive learning overcomes limitations of purely model-based or data-driven approaches, offering a practical solution for complex financial stochastic control problems.

References

Almgren, R. and Chriss, N. (2001). Optimal execution of portfolio transactions. *Journal of Risk*, 3(2):5–39.

Avellaneda, M. and Stoikov, S. (2008). High-frequency trading in a limit order book. *Quantitative Finance*, 8(3):217–224.

Bacry, E., Mastromatteo, I., and Muzy, J.-F. (2015). Hawkes processes in finance. *Market Microstructure and Liquidity*, 1(1):1550005.

Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities. *Journal of political economy*, 81(3):637–654.

Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., and Song, S. (2024). Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*.

Gatheral, J., Jaisson, T., and Rosenbaum, M. (2022). Volatility is rough. In *Commodities*, pages 659–690. Chapman and Hall/CRC.

Geng, Z., Deng, M., Bai, X., Kolter, J. Z., and He, K. (2025). Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*.

Guéant, O., Lehalle, C.-A., and Fernandez-Tapia, J. (2013). Dealing with the inventory risk: a solution to the market making problem. *Mathematics and financial economics*, 7:477–507.

Hawkes, A. G. (1971). Point spectra of some mutually exciting point processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 33(3):438–443.

Li, Q., Zhou, Z., and Levine, S. (2025). Reinforcement learning with action chunking. *arXiv preprint arXiv:2507.07969*.

Lv, L., Li, Y., Luo, Y., Sun, F., Kong, T., Xu, J., and Ma, X. (2025). Flow-based policy for online reinforcement learning. *arXiv preprint arXiv:2506.12811*.

Mandelbrot, B. B. and Van Ness, J. W. (1968). Fractional brownian motions, fractional noises and applications. *SIAM review*, 10(4):422–437.

Merton, R. C. (1969). Lifetime portfolio selection under uncertainty: The continuous-time case. *The review of Economics and Statistics*, pages 247–257.

Merton, R. C. (1976). Option pricing when underlying stock returns are discontinuous. *Journal of Financial Economics*, 3(1-2):125–144.

Perez, L., Wang, X., Zhang, X., Lin, S., and Darrell, T. (2018). Feature-wise transformations for visual reasoning. *Advances in Neural Information Processing Systems*, 31.

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

Wagenmaker, A., Nakamoto, M., Zhang, Y., Park, S., Yagoub, W., Nagabandi, A., Gupta, A., and Levine, S. (2025). Steering your diffusion policy with latent space reinforcement learning. *arXiv preprint arXiv:2506.15799*.