

[medium.com](https://medium.com/@liangnguyen612/reinforcement-learning-in-finance-a-practitioners-roadmap-2b84d84686a6)

<https://medium.com/@liangnguyen612/reinforcement-learning-in-finance-a-practitioners-roadmap-2b84d84686a6>

Reinforcement Learning in Finance: A Practitioner's Roadmap

Quang Khải Nguyễn Hưng

~2 hours



63 min read

May 17, 2026

What's been built, what's working, and what's still open — a 2026 field guide for traders, quants, and ML researchers.

Khai Nguyen — PhD Researcher in Financial Mathematics - The University of Manchester

Press enter or click to view image in full size



A Trade Without a Trader

It's 9:47 a.m. in New York. A pension fund needs to sell 1.2 million shares of Apple by the close, and the trader does not punch the order in herself. Instead, an algorithm — trained for two months on five years of order-book history — begins slicing the parent order into thousands of children. It watches the bid–ask spread, the queue at the top of the book, the imbalance between buys and sells over the last fifteen seconds, and every few hundred milliseconds it

decides whether to send the next child to the lit market, hold back, or step inside the spread to take liquidity. The algorithm has never seen this specific day before, but it has played out millions of synthetic ones in training, and it has learned a policy.

That policy is the product of *reinforcement learning* — the same family of algorithms that beat the world champion at Go, that taught helicopters to fly inverted, that powered AlphaFold's iterative refinement loop. Finance has imported these algorithms in earnest only since 2018, and the eight years since have produced a research field with its own canon, its own benchmarks, its own factional disputes about what is real and what is hype. This article is a roadmap for what has happened, where the field stands in 2026, and what is still unsolved. We will start with a plain-English crash course, walk a four-phase timeline of the field, then tour the six problem domains where RL has settled — execution, market making, hedging, portfolio management, alpha mining, and robo-advising — read the canonical papers in each, and end with the vocabulary you'll need to navigate the literature on your own.

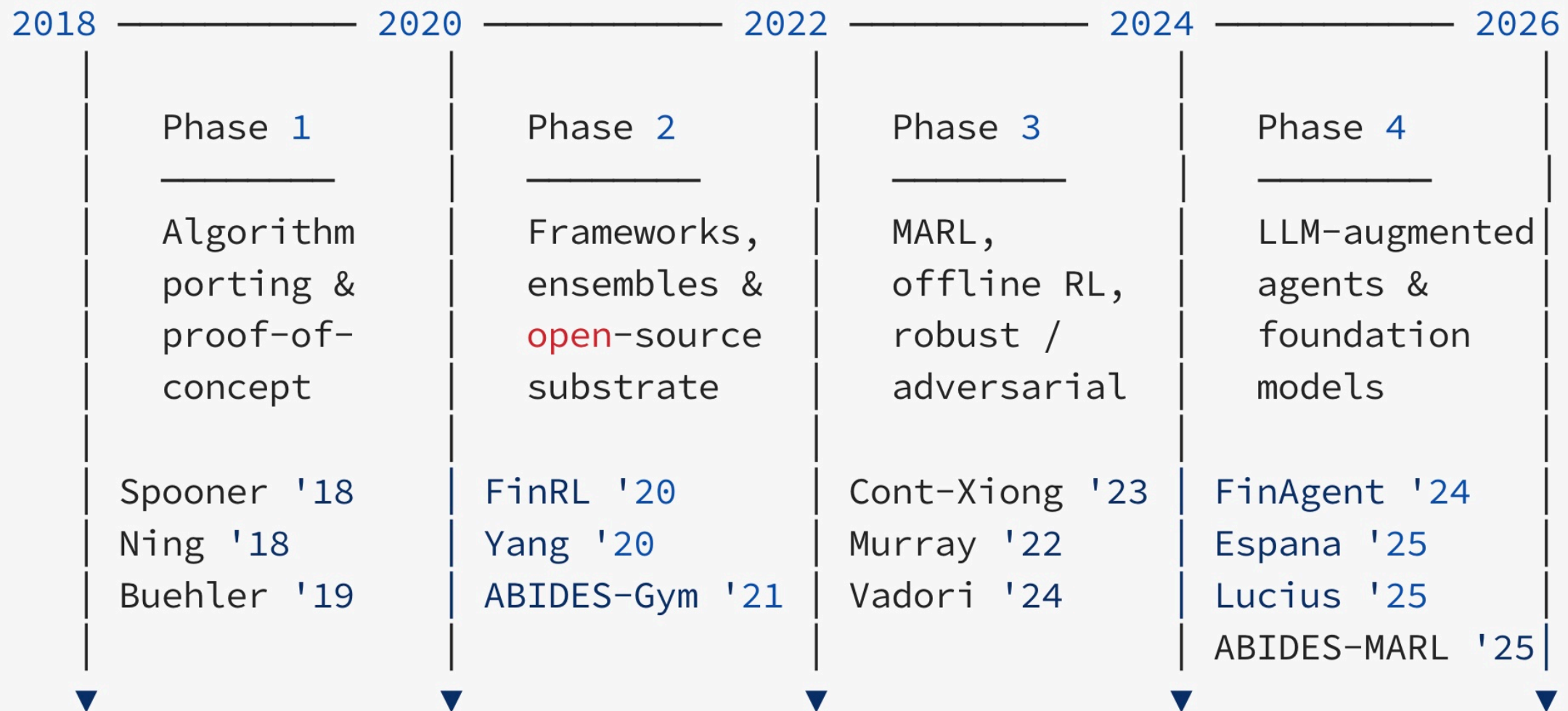
Section 1 — What Is Reinforcement Learning, Really?

Imagine you are training a dog to fetch.

You don't sit it down and show it a million labelled examples of “ball-in-mouth = correct, ball-on-ground = incorrect.” You let it run around, and when something it does looks approximately right — picking up the ball, walking in your direction, dropping it at your feet — you give it a treat. Over hundreds of repetitions the dog converges on a strategy that maximizes its expected treat rate. That is, in essence, reinforcement learning.

The technical vocabulary just adds precision to the picture. The dog is the **agent** — the thing doing the learning. The yard, the ball, you holding the treats, and gravity are all the **environment**. At any moment the dog perceives a **state** — your position, the ball's position, whether you look pleased — and chooses an **action** — run left, sniff a fence post, drop the ball. Each action produces a **reward**, sometimes zero, sometimes a treat, and over time the dog learns a **policy** — a function from states to actions — that maximizes the discounted sum of future rewards.

Two features make this fundamentally different from the more familiar setup where you train a neural network on labelled images. First, there is no labelled “right answer” at each step. The dog doesn't know whether running left was correct, only that doing so eventually led to a treat or not. The learning signal is sparse and delayed — what computer scientists call the *credit assignment problem*. Second, the agent's own actions shape the data it sees next. If the dog learns to chase squirrels instead of balls, it will see a lot of squirrel-related states going forward, and it may never



Section 2
correctly rendered table

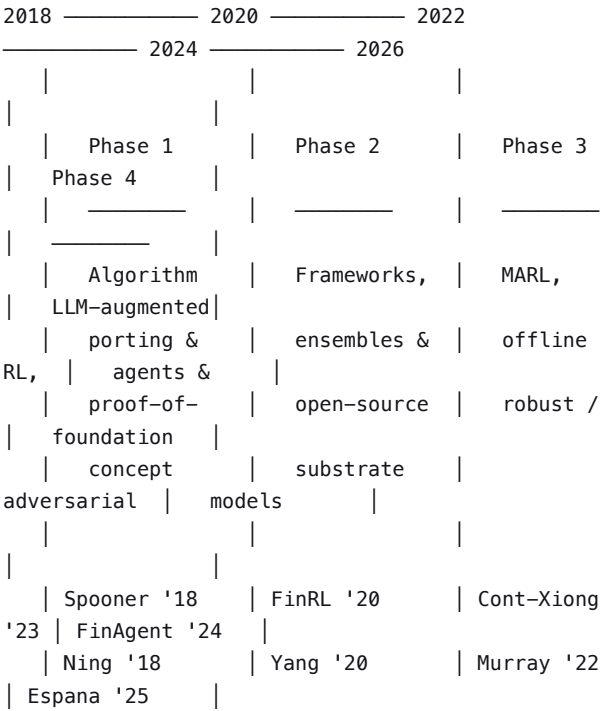
discover the fetch strategy at all. This is the **explore–exploit tension**: should the agent stick with what it knows works, or take a chance on something new?

The explore–exploit dilemma is everywhere in finance. A trading algorithm that has discovered a moderately profitable strategy could stick to it forever — or it could occasionally place orders that look suboptimal in order to learn what would have happened. The latter wastes money in the short run but might discover a much better strategy. In live markets, exploration is brutally expensive — every “let’s see what happens” order has real P&L consequences — which is why nearly every production RL system in finance is trained in simulators and only deployed once the policy has converged.

Finance is almost perfectly structured as an RL problem because it is sequential, the data the agent sees depends on what it does, the rewards are quantifiable in money, and the underlying dynamics are not stationary enough for any closed-form solution to dominate forever.

Section 2 — The Big Picture: An RL-in-Finance Timeline

The field has evolved through four reasonably distinct phases between 2018 and 2026. The phase boundaries are not sharp — methods persist, recombine, occasionally come back into fashion — but the dominant character of each period is recognisable.



Phase 1 (2018–2019) — Algorithm porting and proof-of-concept. The deep-RL breakthroughs of 2015–2017 (DQN on Atari, AlphaGo, A3C, DDPG) made deep neural networks tractable for sequential decision-making, and finance researchers ported them in with minimal modification. The defining theme is “*can RL beat my naive baseline?*”, and the defining papers are Spooner et al.’s market making (AAMAS 2018), Ning–Jaimungal’s optimal execution (arXiv 2018), and Buehler et al.’s Deep Hedging (Quantitative Finance 2019). The infrastructure was bespoke, the benchmarks were idiosyncratic, the action spaces were discrete.

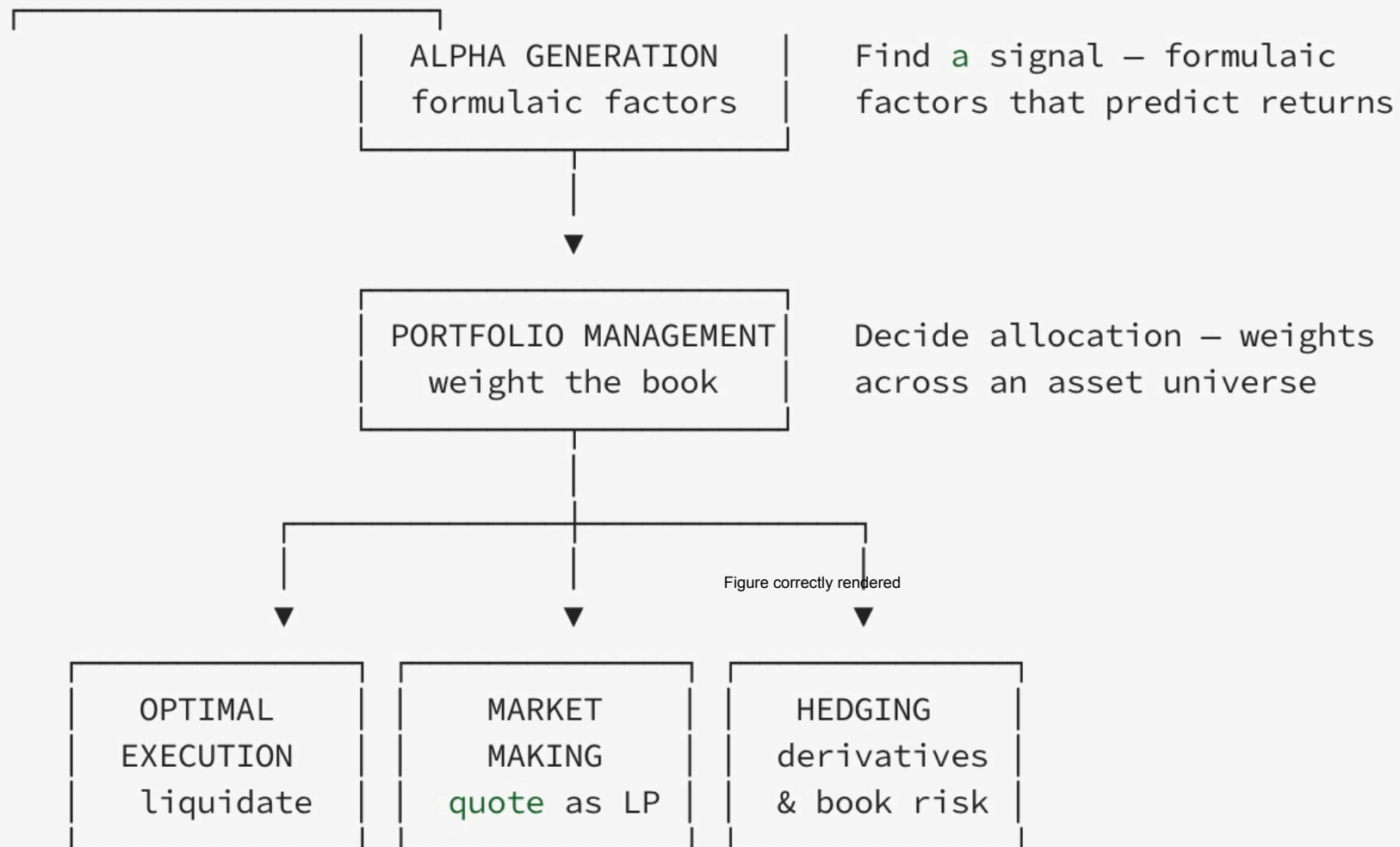
Phase 2 (2020–2021) — Frameworks, ensembles, and open-source. The field consolidated around shared toolkits. FinRL appeared at NeurIPS 2020 as the first end-to-end open-source library for financial RL, the canonical PPO+A2C+DDPG ensemble crystallised as the default stock-trading baseline, and JP Morgan + Georgia Tech released ABIDES-Gym as the standard agent-based LOB simulator. The defining theme is “*can we reproduce each other’s results?*”, and the defining papers are Liu et al.’s FinRL and Yang et al.’s ensemble strategy.

Phase 3 (2022–2023) — MARL, offline RL, robust and adversarial. Single-agent RL matured, and the field turned to richer formulations: multi-agent games, adversarial robustness, offline learning from historical data alone, risk-sensitive reward shaping. The defining theme is “*how does my agent behave in a market full of other learning agents?*”, and the defining papers are Cont & Xiong on tacit collusion among learning market makers, Murray et al. on continuous-RL Deep Hedging, and Khraishi & Okhrati on offline RL for consumer credit pricing.

Phase 4 (2024–2026) — LLM-augmented agents and foundation models. Large language models become first-class components of trading agents — as memory modules, as factor proposers, as policy backbones, as sentiment-signal generators. Generative foundation models of the order book itself begin to appear. The defining theme is “*can the LLM stack reshape what a trading agent looks like?*”, and the defining papers are FinAgent and FinCon on the agent side, Shi 2025 on LLM-MCTS alpha mining, and the BlackRock Deep Hedging release on the auditable-industrial side.

The Six Problem Domains

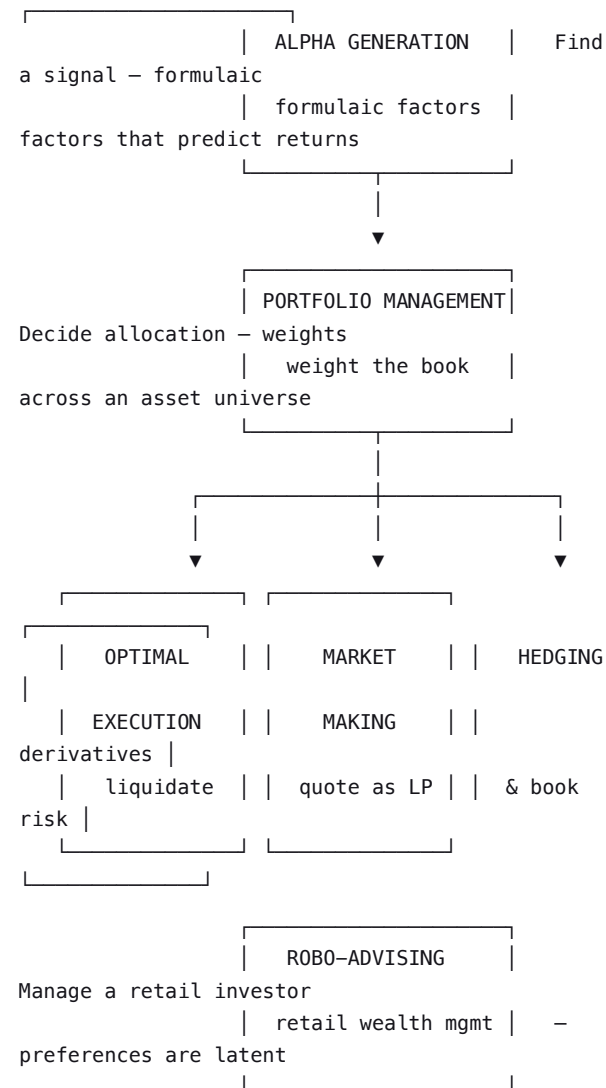
Across these phases, RL in finance has settled around six problem formulations that recur across virtually every survey. They cascade



ROBO-ADVISING
retail wealth mgmt

Manage a retail investor
– preferences are latent

roughly in the order a trade actually happens at a real fund:



Each domain has its own canon, its own benchmarks, its own factional disputes. We'll tour them one by one.

Section 3 — OPTIMAL EXECUTION

3a. The Problem in Plain English

A trader holds a position — say, a million shares of one stock — and must liquidate it by some deadline, perhaps the market close. Selling everything in one shot moves the market against her, because her own selling pressure absorbs the resting bids and

pushes the price down: this is **market impact**. Selling too slowly leaves her exposed to whatever else moves the price between now and the deadline: this is **timing risk**. The optimal execution problem is to find a sequence of child orders that minimizes the gap between the price she would have got at decision time and the average price she actually realises. That gap has a universal name — *implementation shortfall* — and it is the yardstick the entire field measures itself against.

The classical answer is Almgren–Chriss (2000): an analytical schedule that trades faster when timing risk is high and slower when impact is high, under specific assumptions about how impact and volatility behave. The RL question is whether a model-free agent, given only the order-book state and the realised outcomes, can do better when those classical assumptions fail.

3b. The RL Setup — Explained Intuitively

The canonical execution MDP is the one Ning, Lin & Jaimungal formalised in 2018, and it is the cleanest place to start.

- **State — what the agent sees.** Three blocks of information. First, the *clock*: how much time is left until the deadline. Second, the *book*: how many shares still need to be sold. Third, the *market*: features of the current limit order book — the spread between best bid and ask, the queue depth on each side, recent price moves, and (in richer variants) a local volatility estimate from realised quadratic variation.
- **Action — what the agent decides.** How many shares to send into the market on this step. In Ning et al.'s setup the action is discrete (a small set of trade sizes); in later continuous-action papers like Micheli & Monod's it is a fraction of the remaining inventory.
- **Reward — what the agent is scored on.** Negative implementation shortfall — the difference between the arrival-price benchmark and the actual fill price — plus a quadratic penalty term that mimics market impact. The economic logic is direct: the agent wants to capture the arrival price as closely as possible, and the quadratic penalty teaches it that pushing large size into the book at once is more costly than slicing it up. There is also an implicit *time penalty*: failing to finish the order by the deadline is catastrophic, so the agent learns to respect the clock.

3c. Roadmap of Key Papers

2006 → Nevmyvaka et al. Tabular Q-learning. First execution-RL paper.
2014 → Hendricks & Wilcox AC-as-prior, RL-as-correction. +10.3% on JSE.
2018 → Ning, Lin & Jaimungal Canonical Double DQN benchmark.

2020 → Karpe et al.	Embeds the agent in ABIDES; converges to TWAP.
2021 → Fang et al. (OPD)	One policy across thousands of stocks.
2022 → Fang et al. (DWD-PP0)	Imitate, then transcend. PP0 denoising.
2024 → Macrì & Lillo	Latent time-varying liquidity.
2024 → Micheli & Monod	Transient impact via decay-kernel state.
2025 → Espana et al. (QRM)	Endogenous impact via Queue-Reactive sim.

3d. Paper Deep-Dives

2006 — Nevmyvaka, Feng & Kearns — *Reinforcement Learning for Optimized Trade Execution*. The founding paper of the field, and the one that asked the obvious heretical question: could a model-free agent learn to liquidate a position by observing the limit order book directly, with no prior model of price dynamics? Their answer was tabular Q-learning over a discretised state space, with the agent picking the limit price (relative to the mid) at which to submit each child order over a fixed liquidation horizon, and reward equal to negative implementation shortfall over the episode. Because the state was discretised, the value function lived in a lookup table rather than a neural network — this is pre-deep-RL by almost a decade. Trained and evaluated on millions of NYSE order-book messages, the Q-learning agent meaningfully reduced execution cost relative to a submit-and-leave baseline. The headline number matters less than what the paper established: the canonical state-action-reward template (LOB features → limit price/volume → negative shortfall) that every subsequent execution-RL paper extends. The gap it left open was scale — a lookup table cannot absorb the rich, high-dimensional LOB state that deep networks would later swallow whole.

2014 — Hendricks & Wilcox — *A Reinforcement Learning Extension to the Almgren–Chriss Framework*. Hendricks and Wilcox sit on the bridge between stochastic control and RL: rather than abandoning the classical Almgren–Chriss schedule, they use it as a prior and train a Q-learning agent to deviate from it based on the current state of the order book. The intuition is that AC gives a good first-order solution and RL fills in the second-order corrections AC’s stationarity assumption ignores. The agent’s action is not the raw order size but a multiplicative tweak to the AC-recommended quantity at each step — trade 1.2× or 0.8× of what AC says — conditioned on time remaining, inventory remaining, and microstructure features from the book. Tested on Johannesburg Stock Exchange Top 40 names — a thin-liquidity emerging-market context where classical AC assumptions strain hardest — the agent

delivered a 10.3% average improvement in implementation shortfall over pure Almgren–Chriss. The “AC-as-prior, RL-as-correction” template they established is still in active use. The limitation: the prior is itself a strong constraint, and the agent inherits AC’s blind spots wherever the underlying dynamics violate the linear-impact assumption.

2018 — Ning, Lin & Jaimungal — *Double Deep Q-Learning for Optimal Execution*. The canonical deep-RL execution paper, and the benchmark essentially every subsequent paper compares against. The authors throw the parametric model out entirely and train a deep Double Q-network end-to-end: a fully connected MLP trained with experience replay and the Double DQN target update to mitigate Q-value overestimation, with input features comprising current LOB state, time and inventory remaining, and — in their richer TIPQV variant — realised quadratic variation as a local volatility signal. The action space is discrete (how many shares to send at each step), and the reward is negative implementation shortfall plus a quadratic execution-cost penalty that mimics market impact. Across nine US equities the DDQN agent outperformed TWAP — the industry’s lowest-bar baseline — on mean and median P&L, on per-episode probability of outperformance, and on gain-loss ratio, with the TIPQV variant beating plain TIP because the agent learns to trade faster when volatility (and timing risk) is high. The reward design and TWAP benchmarking convention from this paper are now standard. The gap: training on historical market replay means the agent’s own orders don’t affect the price tape, so impact is artificially absent during training — a flaw the next paper attacks directly.

2020 — Karpe, Fang, Ma & Wang — *Multi-Agent RL in a Realistic LOB Simulation*. Karpe and colleagues made the point that training Ning-style agents on historical market replay is fundamentally circular: the agent’s own orders cannot influence a replay tape, so price impact is artificially absent during training. Their fix is to run the entire execution problem inside ABIDES — the agent-based discrete-event LOB simulator built jointly by Georgia Tech and JP Morgan — where the agent’s orders genuinely move the book and affect other simulated participants. The algorithmic backbone is the same Double DQN used by Ning et al., but now embedded among background liquidity providers, momentum traders, and noise traders, observing microstructure features (best bid/ask, depth, recent trade signs) before deciding placement and sizing each step. Calibrated to NASDAQ LOB 2003 data for IBM, the simulator faithfully reproduced stylized facts of real LOBs even with the agent embedded, and the trained agent converged to a TWAP-like execution profile across a wide range of scenarios. The result cuts both ways: it confirms RL can rediscover a sensible classical heuristic in symmetric environments, but it also suggests that on simple books there is nothing fundamentally better

than TWAP to find. The last contribution is making ABIDES the de facto research simulator for execution-RL.

2021 — Fang, Ren, Liu, Zhou, Zhang, Bian, Yu & Liu —

Universal Trading for Order Execution with Oracle Policy Distillation. The biggest practical weakness of execution-RL up to this point was that a policy trained on one stock rarely transferred to another, so production desks would need to maintain thousands of separate agents — an operational nightmare. The answer Fang et al. propose is *oracle policy distillation*: train a teacher agent that has access to *future* market information (the oracle) to discover the near-optimal schedule, then distill its decisions into a deployable student policy that only sees information available in real time. Both teacher and student are PPO agents whose policy network uses a GRU to encode public microstructure features and private inventory/time features through a shared MLP; the action is a continuous fraction of remaining inventory to trade in the next interval, and the student's loss adds a distillation term pulling its action distribution toward the teacher's. Evaluated on Chinese A-shares minute-level data from January 2017 to June 2019 over a four-hour intraday horizon, the universally trained OPD policy outperformed DDQN, vanilla PPO and analytical AC — and after fine-tuning matched single-instrument PPO agents, with the operational killer feature that one policy works across thousands of stocks. The oracle-distillation idea has since been reused in market-making and alpha-mining papers. The open gap: the teacher's "oracle" information is itself a simulator-grade luxury and is not available at training time on most real datasets.

2022 — Fang, Weng, Xiang & Zhang — *Imitate then Transcend:*

DWD-PPO. Two pain points met in this paper. PPO struggles in noisy market environments because the policy gradient is dominated by reward noise, and RL agents starting from random initialization waste enormous compute exploring obviously bad policies. The fix is a two-stage recipe stacked on standard PPO. The *Dual-Window Denoise PPO* architecture feeds the policy through two parallel observation windows of different lengths, denoising short-term noise while preserving longer-range signal; the reward scheme combines imitation learning with environment reward, so early in training the agent is rewarded for matching TWAP-like behaviour and then the imitation weight decays and the agent is free to *transcend* the teacher; and the action formulation has the agent jointly decide both the volume to trade and the order placement rather than splitting these into separate sub-problems. Evaluated inside a multi-agent realistic historical LOB simulator where the agent's orders produced measurable impact, the full system outperformed TWAP and vanilla PPO baselines, with ablations confirming each component matters. The "imitate then transcend" pattern — start the agent inside the behavioural cloud of a known-good strategy, then let it drift — is now common in

industry-leaning execution research. The remaining gap: TWAP itself is a weak teacher, and what happens if the teacher is a more sophisticated benchmark is open.

2024 — Macri & Lillo — *RL for Optimal Execution When Liquidity is Time-Varying.* Macri and Lillo confront a flaw lurking inside every previous paper on this list: they all assume the market impact parameter — how much the price moves per unit of order volume — is either constant in time or perfectly observable, when in reality it is both dynamic and latent. The authors train a Double Deep Q-Network inside an Almgren–Chriss-style environment where the temporary impact parameter is stochastic and time-varying, while the agent observes only time, inventory and price and must infer the latent liquidity regime implicitly from price movements alone. Across numerical simulation experiments over several time-varying liquidity processes, the DDQN agent learns to adapt — trading more aggressively when liquidity is high, slowing down when it thins — and outperforms both classical AC (which assumes constant liquidity) and a naïve agent that ignores liquidity dynamics. The paper provides one of the cleanest analytical demonstrations that RL can extract latent regime information from price dynamics alone, and it sets the theoretical foundation for the counterfactual-simulator work that follows. The gap: the experiments are on synthetic processes, not calibrated empirical kernels of impact at trading-day horizons.

2024 — Micheli & Monod — *Deep RL for Online Optimal Execution Strategies.* This paper attacks the Markov property head-on. The Markov assumption — that the future depends on the past only through the current state — is the bedrock of standard RL, but real market impact is *transient*: a trade you placed five minutes ago is still affecting prices now, decaying over a kernel that may span hours, and almost every prior RL paper sidesteps this. The authors use Deep Deterministic Policy Gradient (DDPG) with three modifications: a continuous action space (DDPG's natural setting, removing DQN-style discretisation artefacts), an auxiliary Q-function custom-designed for the execution problem to stabilise training, and a state representation that explicitly tracks the recent history of trades weighted by a decay kernel, so the policy can reason about the residual impact of its own past actions. Across numerical experiments under several decay-kernel families (exponential, power-law) and parameter regimes where the decay rate itself shifts over time, the DDPG agent approximates the analytically optimal execution strategy under each kernel and adapts when kernel parameters fluctuate. The continuous action space, the auxiliary Q-function trick, and the explicit decay-kernel state representation are all likely to be reused. The open question: whether the decay-kernel state representation scales to high-dimensional LOB observations remains untested.

2025 — Espana, Hafsi, Lillo & Vittori — *RL in Queue-Reactive*

Models. Espana and colleagues sit at the bleeding edge of the auditable-simulator-plus-RL frontier. Their observation is that historical market replay cannot give policy training what it really needs — counterfactual feedback about how prices would have moved had the agent traded differently — so they put the agent inside a Queue-Reactive Model, an LOB simulator where order arrivals depend on queue depths in a way that naturally produces transient impact and nonlinear flow responses while remaining analytically tractable. The agent is a Double Deep Q-Network observing time remaining, inventory remaining, current price, and queue-depth features; the simulator endogenously generates both *direct* impact (consuming liquidity at the touch) and *indirect* impact (other participants reacting to the agent’s trades); reward is implementation shortfall over the meta-order horizon. Across multiple calibrated Queue-Reactive configurations, the DDQN agent learns policies that are both strategic (matching the long-horizon volume profile) and tactical (reacting to favourable queue conditions in real time), outperforming TWAP and AC. Together with ABIDES-MARL, this paper is the new template for execution-RL training environments. The gap it itself acknowledges: Queue-Reactive models are still stylised, and bridging to truly empirical LOB calibration remains an active research direction.

3e. Open Gaps in Optimal Execution

- **Transient impact at realistic horizons.** Micheli & Monod handle it in toy decay kernels; nobody has shown the same on empirically calibrated kernels at multi-day horizons.
- **Cross-asset and cross-regime transfer.** Fang’s oracle distillation transfers across Chinese equities. Nothing has shown transfer from US large-caps to emerging-market small-caps or across asset classes.
- **Multi-venue and dark-pool routing.** Real execution desks fragment orders across lit books, dark pools, and venues with different fee structures. Almost no published RL paper handles this.
- **Off-policy evaluation.** Backtests assume the agent’s orders don’t move the world; agent-based simulators assume the simulator is right. Off-policy evaluation methods that quantify confidence in either are still missing from the finance literature.
- **Joint execution and alpha.** Execution agents take the trade signal as given; alpha agents assume frictionless trading. A single agent that decides *whether* to trade and *how* to execute remains an open architectural problem.

Section 4 — MARKET MAKING

4a. The Problem in Plain English

A market maker is the bookshop on the corner of the financial world: she stands behind a counter and offers to *buy* the asset at one price (her bid) and *sell* it at another (her ask), pocketing the spread between them as compensation for providing liquidity to whoever shows up. The catch is that order flow is not random. Some of the people walking up to her counter are informed — they want to buy because they know the price is about to rise, or sell because they know it is about to fall — and a market maker who quotes the same prices to everyone ends up *long the losers and short the winners*. This is **adverse selection**, and it is the dominant economic risk of the job.

The second risk is **inventory**: every trade she fills leaves her with a position, and if that position drifts too far from zero she becomes a directional bettor rather than a liquidity provider. The job is to quote tight enough to capture flow but defensive enough that adverse selection and inventory don’t eat the spread.

4b. The RL Setup — Explained Intuitively

Spooner et al.’s 2018 formulation is the template. It is interpretable, intuitive, and almost every subsequent paper either adopts it or reacts to it.

- **State — what the agent sees.** Time remaining in the trading session, the agent’s current inventory (signed — positive if long, negative if short), and microstructure features of the book: the current spread, the order-flow imbalance over the last few seconds, and recent price moves.
- **Action — what the agent decides.** A pair of quote distances from the mid-price: how far below the mid to place her bid, and how far above to place her ask. Wider quotes earn more per trade but get hit less often; tighter quotes earn less per trade but capture more flow.
- **Reward — what the agent is scored on.** Realised P&L per step, passed through an *asymmetrically dampened* function that punishes the agent harder for trades that grow her inventory than it rewards trades that flatten it. This is the field’s signature design choice. Spooner could have hard-coded inventory-aware quote skewing, but instead built the inventory penalty into the reward and let the agent discover skewing on its own. The result is that the canonical Avellaneda–Stoikov reservation-price behaviour — quote tighter on the side that flattens inventory, wider on the side that builds it — emerges naturally from the reward shape rather than being prescribed.

4c. Roadmap of Key Papers

asymmetric inventory reward.	
2019 → Ganesh et al. (JPM)	RL in a multi-agent dealer market.
2020 → Spooner & Savani	Adversarial RL → robust market maker.
2022 → Gašperov & Kostanjčar	Hawkes-process LOB simulator.
2023 → Guo et al. (Attn-LOB)	Convolution + attention on raw LOB.
2023 → Cont & Xiong	Tacit collusion among RL market makers.
2024 → Vadori et al. (JPM)	Two-population RL simulator of OTC FX.
2025 → Jafree, Jain & Firoozye	PP0 + impulse control vs meta-orders.
2025 → Zimmer & do Valle Costa	PP0 under non-stationary LOB dynamics.

4d. Paper Deep-Dives

2018 — Spooner, Fearnley, Savani & Koukorinis — *Market Making via RL*. The canonical RL market-making paper, and the one that gave the field its template inventory-aware reward function. The authors build a high-fidelity LOB simulator and train a market-making agent using temporal-difference learning with a linear combination of tile codings as the value-function approximator — a deliberate pre-deep-RL choice that keeps the policy interpretable. The agent's action is a pair of quote distances from the mid; the state encodes time, inventory, and microstructure features; the reward is the asymmetrically dampened P&L function described above. Trained on high-frequency historical equities data inside their simulator, the TD agent outperformed both simple benchmarks (constant-spread quoters) and a recent online-learning approach from the academic literature. The asymmetric inventory penalty in particular has been adopted by almost every later market-making RL paper. The gap it left open: training against passive replayed flow rather than strategic counterparties — the next paper attacks this directly.

2019 — Ganesh, Vadori, Xu, Zheng, Reddy & Veloso — *RL for Market Making in a Multi-Agent Dealer Market*. This is the paper that opens JP Morgan AI Research's now-flagship dealer-market line. Where Spooner trained against passive LOB flow, JPM argues that real dealer markets — especially in FX and fixed income — are *bilateral* and populated by *strategic competitors*, and an RL market maker should be evaluated against other adaptive agents. They build a multi-agent simulator in which the RL agent learns alongside an adaptive benchmark market maker, and they study its behaviour across varying competitive scenarios, reward formulations, and market price drifts. Without any hand-coded inventory rule, the RL

agent learns to manage inventory by *skewing* — quoting asymmetric prices on the buy and sell sides — and to maintain a positive (negative) inventory bias when the market price drift is positive (negative), effectively betting with the trend while remaining a market maker. The contribution is less a single P&L number and more the demonstration that RL recovers economically sensible dealer-market behaviour in a multi-agent setting — a result that motivated the entire JPM ABIDES / MARL programme.

2020 — Spooner & Savani — *Robust Market Making via Adversarial RL*. The field's worst nightmare: a market maker trained against benign simulated flow looks excellent in backtest and gets destroyed in live trading the moment an informed counterparty shows up. Spooner and Savani's fix is to turn the canonical Avellaneda–Stoikov single-agent model into a discrete-time zero-sum game between the market maker and an *adversary* that plays the role of a worst-case opponent profiting at the market maker's expense — adversarial reinforcement learning, borrowed from the robotics literature. Both agents train simultaneously, and the market maker's policy must withstand whatever the adversary learns to do. Three findings stand out: risk-averse behaviour emerges naturally without any explicit inventory penalty (the adversary punishes large positions by moving prices against them), the ARL agent significantly outperforms conventional single-agent RL on standard metrics whether tested with or without an adversary present, and the policy gains measurable robustness to model uncertainty. The paper also proves convergence for several special cases and shows that the equilibria the agents converge to correspond to known optimal-control solutions in those cases.

2022 — Gašperov & Kostanjčar — *Deep RL for Market Making Under a Hawkes-Process LOB*. Real LOB events are not Poisson — they exhibit clustering, where the arrival of one trade dramatically raises the probability of another in the next few milliseconds — and any market maker trained in a model that misses this clustering will misprice volatility risk. Gašperov and Kostanjčar train a deep-RL controller inside a *weakly consistent multivariate Hawkes process* LOB simulator that explicitly models the self-excitation among order-book events. The state captures inventory and Hawkes-process intensity features; the action is the bid/ask quote pair; the reward combines spread capture with an inventory penalty. Compared against multiple market-making benchmarks (including classical Avellaneda–Stoikov), the DRL controller delivers superior risk-reward metrics even under significant transaction costs, and crucially holds up under the clustered, bursty flow patterns where the classical model degrades fastest. The paper helped legitimise Hawkes-based simulators as a serious training environment for MM agents and feeds directly into the later Jain–Firoozye line on adverse selection.

2023 — Guo, Aste and colleagues — *Attn-LOB: Market Making*

with Deep RL from Limit Order Books. Most prior MM-RL work over-engineers the state: researchers hand-craft features like spread, mid-price movement, and order-flow imbalance from the raw LOB, and the resulting representation is rigid. Guo et al. propose **Attn-LOB**, a neural network combining convolutional filters with an attention mechanism that learns LOB features end-to-end directly from the raw multi-level book snapshot, replacing the hand-crafted feature engineering with a learned representation. On top of this they introduce a continuous action space (rather than the discrete bid/ask bins typical of earlier work) and a hybrid reward function combining spread capture with risk-adjustment terms. Their experiments span both performance and operational metrics: they evaluate latency to confirm the architecture is feasible at HFT speeds, and they probe the interpretability of the learned attention weights to check that the network is responding to economically sensible features. The Attn-LOB architecture has become a reference for “raw-LOB” market making in subsequent papers.

2023 — Cont & Xiong — *Dynamics of Market Making Algorithms: Learning and Tacit Collusion*. The single most influential cautionary paper in the field, and it is not actually about *training* a better market maker — it is about what happens when many of them coexist. Cont and Xiong model competition among dealers in an OTC market as a stochastic differential game of intensity control with partial information, characterising the competitive (Nash) equilibrium via a system of coupled HJB equations and Pareto-optimal collusive outcomes via the corresponding Pareto frontier. They then run a *decentralized* multi-agent deep-RL experiment — independent actor-critic learners with no information sharing — in which competing market makers learn to adjust their quotes purely from their own observed P&L. The result is alarming: even with no communication, no shared training signal, and no explicit coordination, the population of learners converges to spread levels strictly above the competitive equilibrium — *tacit collusion* — in a way that increases dealer profits at the cost of end users. This finding opened a regulatory research agenda still being digested by the FCA, SEC, and ESMA, and it sits behind every subsequent “robust MARL” paper that worries about emergent equilibria in deployed RL systems.

2024 — Vadori, Ardon, Ganesh, Spooner, Amrouni, Vann, Xu, Zheng, Balch & Veloso — *MARL-Driven OTC Market Simulations*. JP Morgan’s most ambitious published paper to date pushes the dealer-market programme from a single learning market maker to *fully RL-driven OTC market simulations*. The setup is a game between two heterogeneous populations — liquidity providers (the market makers) and liquidity takers (the clients hitting the quotes) — both trained with shared-policy MARL so that each agent within a population uses a common policy but encounters distinct realisations of state and reward. Action spaces are

(quote spreads on the LP side; trade timing and size on the LT side), the reward functions include both P&L and behavioural shaping terms, and the simulator is calibrated to characteristics of real FX dealer markets. The trained agents jointly produce emergent behaviour aligned with classical microstructure intuition: market makers learn to skew quotes against inventory and against detected informed flow, liquidity takers learn to split orders and route across competing dealers, and the resulting market exhibits realistic spread dynamics and trade-share distributions. The paper is the field’s most complete demonstration that RL can both *generate* a realistic dealer market and *populate* it with agents whose behaviour can be calibrated to data.

2025 — Jafree, Jain & Firoozye — *When AI Trading Agents Compete: Adverse Selection of Meta-Orders*. One of the most precise dissections of an empirical worry every execution desk has: high-frequency RL market makers can systematically detect and exploit slower meta-orders, eating into the slippage performance of medium-frequency institutional traders. The authors build a Hawkes-LOB simulator and train an RL market maker inside it using **PPO with self-imitation learning**, then wrap the policy in an **impulse-control framework** where the agent intervenes only at discrete, optimally-chosen moments via limit, cancel, or market orders. They then expose the RL market maker to a medium-frequency trader running a TWAP-style meta-order. With training against the meta-order flow, the RL market maker learns to recognise the price drift induced by the meta-order and trade aggressively against it, achieving Sharpe ratios above 30 in the simulated environment. Importantly, the paper does *not* find that increased MM profits translate into proportionally worse MFT slippage, which complicates the simple “HFT predation” narrative. The work bridges the gap between microstructure theory (Hawkes-modelled endogenous impact) and deployable RL.

2025 — Zimmer & do Valle Costa — *RL-Based Market Making on Non-Stationary LOB Dynamics*. Zimmer and do Valle Costa formalise an observation the field had been quietly accepting: almost every previous MM-RL paper trains in a stationary environment — constant mean spread, constant arrival intensity, constant volatility — and real markets violate all three of these simultaneously. The authors build a simulator that explicitly bakes in non-stationary stylized facts: clustered order arrival times, non-stationary spreads and return drifts, stochastic order quantities, and time-varying volatility. They then train a PPO market-making agent inside it and benchmark its performance against the closed-form optimal solution from classical inventory-aware MM theory under a range of varying market conditions. The PPO agent maintains profitability and risk control where the closed-form policy degrades, demonstrating that adaptive RL policies can carry over the regime shifts that break parametric controllers. The simulator itself is

positioned as a *pre-training* environment — somewhere RL agents can develop robustness before being fine-tuned on real data — which is the practical industry direction the field is moving in.

4e. Open Gaps in Market Making

- **Tacit-collusion mitigation.** Cont & Xiong showed the problem exists. Nobody has yet shown a clean training-time intervention — regulatory cap, randomisation, exploration constraint — that prevents the wide-spread equilibrium without hurting individual market-maker performance.
- **Adverse selection across asset classes.** Jafree et al. handle equity-like Hawkes LOBs. The corresponding analyses for FX dealer markets, fixed-income request-for-quote flows, and crypto are missing.
- **Real-money deployment evidence.** Every published result is simulator-based. There is essentially no public evidence of how these agents perform in production, partly because banks and hedge funds keep that data proprietary.
- **Integration with internalisation and hedging.** Real market makers internalise flow against their own inventory and hedge residual risk via correlated instruments. RL papers usually treat the MM problem in isolation.
- **Auditability and explainability.** Attn-LOB shows attention weights can be inspected, but no MM-RL paper produces a policy that would pass a bank model-risk review with the rigour applied to an Avellaneda–Stoikov implementation.

Section 5 — HEDGING

5a. The Problem in Plain English

A bank writes an option to a client. Now it sits on a derivatives exposure whose value depends nonlinearly on the underlying price, the implied volatility, interest rates, and time to expiry. The textbook fix is *delta hedging*: buy more shares as the price rises, sell as it falls, to neutralise the first-order risk. The textbook also assumes continuous trading, no transaction costs, and infinite liquidity — and none of these hold in practice. Real hedging is the search for a *policy* mapping market state to a discrete trade that minimises some risk measure of terminal P&L, net of frictions, across a *book* of many positions.

RL is a natural fit because hedging is sequential, path-dependent, and the objective is fundamentally about controlling the *distribution* of P&L rather than maximising its mean — which is exactly what a

risk-aware reward function expresses.

5b. The RL Setup — Explained Intuitively

Murray et al.’s 2022 Deep Hedging formulation is the canonical continuous-RL setup, and it is the cleanest description of the modern problem.

- **State — what the agent sees.** The current state of the portfolio (positions in the underlying, in hedge instruments, P&L so far), the market state (spot, implied vol, time to expiry), and — uniquely in their setup — the agent’s *risk-aversion parameter* itself. This last item lets one trained model serve every risk-aversion level a trader might want.
- **Action — what the agent decides.** A trade — how many units of the hedge instrument to buy or sell on this step. Continuous-valued, so the policy is a real-valued function rather than a discrete classifier.
- **Reward — what the agent is scored on.** A *monetary utility function* applied to incremental P&L, with transaction costs baked into the path. The economic logic: a risk-averse hedger does not want to maximise expected P&L; she wants to minimise the convex-risk-measure (entropic risk, CVaR) of her terminal P&L, with costs subtracted. Murray et al.’s monetary-utility formulation has the elegant property that the trained value function doubles as an indifference price for the portfolio — so the critic is simultaneously a pricer and a risk controller.

5c. Roadmap of Key Papers

2019 → Buehler et al.	Deep Hedging – supervised end-to-end NN.
2020 → Du et al.	First true RL hedge – DQN–PopArt, PPO.
2021 → Buehler, Murray et al.	Drift-removal via near-martingale measures.
2022 → Murray et al.	Continuous RL across risk aversions.
2022 → Buehler, Murray & Wood	Deep Bellman Hedging – arbitrary portfolios.
2023 → Mandelli et al.	Credit-index options on real CDX data.
2024 → Neagu et al.	Convex market impact inside hedging env.
2024 → Peng et al.	CVaR-weighted, contract-unified PPO.
2025 → Lucius et al. (BlackRock)	Production-grade auditable Deep Hedging.

5d. Paper Deep-Dives

2019 — Buehler, Gonon, Teichmann & Wood — *Deep Hedging*.

The foundational paper that named the entire research direction. Buehler and colleagues recast option hedging as a constrained neural-network optimisation: given a payoff at maturity and a set of tradable hedge instruments under market frictions (transaction costs, liquidity, position limits), train a feedforward policy network that maps each step's market state to a trade and minimise a *convex risk measure* (entropic risk, CVaR, mean-variance) of the resulting terminal P&L. Technically the paper is closer to supervised learning than to traditional RL — there is no explicit value function and gradients flow through the entire hedging path via backpropagation through time — but it is the doorway into all the actor-critic successors. They prove that the constrained set of NN trading strategies is dense enough to ϵ -approximate any optimal hedging strategy, removing the worry that the architecture imposes a binding constraint. The algorithm is model-agnostic: it runs on any simulator that generates arbitrage-free paths, scales to high dimensions because compute depends only on the number of *hedging instruments*, and generalises across instrument types.

2020 — Du, Jin, Kolm, Ritter, Wang & Zhang — *Deep RL for Option Replication and Hedging*.

The first paper to bring *true* RL — explicit MDP, exploration, value function — to option hedging under realistic frictions, building on Kolm & Ritter (2019). Du and colleagues solve option replication subject to *discrete trading*, *round lotting* (no fractional shares), and *nonlinear transaction costs* — three frictions any deployed system must respect and most of the previous literature glosses over. They train and compare three deep-RL algorithms on the same problem: vanilla **Deep Q-Networks**, **DQN with Pop-Art** (which adaptively normalises the value function to handle the large dynamic range of P&L outcomes), and **PPO**. A critical operational feature is that each agent is trained to hedge a whole *range of strikes* — not a single strike — so a deployed model handles any contract within the trained range without retraining. Across simulations, the deep-RL strategies match or beat classical delta hedging, and PPO wins on all three operational metrics — final P&L, training time, and sample efficiency — effectively establishing PPO as the default actor-critic choice for option hedging.

2021 — Buehler, Murray, Pakkanen & Wood — *Deep Hedging: Removing the Drift*.

A subtle but fundamental flaw lurks inside the original Deep Hedging recipe. Real-world price simulators contain *drift* — assets have expected returns — and a P&L-maximising policy trained inside them will end up *speculating*, taking directional bets on the drift rather than hedging cleanly. The authors show how to construct a **minimal entropy near-martingale measure** under the friction constraints, effectively removing the drift while preserving the second-moment structure that hedging depends on, then run Deep Hedging *under that measure* rather than the physical

one. Conceptually the paper extends the no-arbitrage / minimal-entropy machinery from frictionless mathematical finance to the friction-aware deep-hedging world. The numerical example is a stochastic implied volatility market, where the method enables consistent option pricing and hedging under transaction costs without the policy degenerating into a directional bet. The paper bridges the engineering of Deep Hedging 2019 and the more rigorous risk-neutral RL formulations that follow.

2022 — Murray, Wood, Buehler, Wiese & Pakkanen —

Continuous RL Deep Hedging Across Risk Aversions. The cleanest reformulation of Deep Hedging as proper continuous RL with both an actor and a critic. The contribution is twofold. First, the authors derive a **risk-averse Bellman representation** of the hedging problem using *monetary utility functions* — a class of convex risk measures whose value function additionally encodes an indifference price for the portfolio, so the trained critic doubles as a pricing engine. Second, they develop a novel actor-critic algorithm that learns *both* the optimal hedging policy and the value function simultaneously, using model-free, on-policy deep function approximation across the joint feature space of portfolio state, market state, and *risk-aversion parameter*. The single most useful operational feature is training across multiple risk-aversion levels simultaneously: rather than training one model per risk-aversion setting, the agent learns a policy conditioned on risk-aversion, so a trader can slide a single knob at inference time to span the efficient frontier. Numerical evaluation under stochastic-volatility dynamics demonstrates the algorithm produces good hedges across the whole frontier from one trained policy.

2022 — Buehler, Murray & Wood — *Deep Bellman Hedging*.

JPM's most operationally relevant hedging paper generalises the actor-critic deep-hedging idea to *arbitrary* portfolios trained on *purely historical data* — the configuration any production desk actually faces. Methodologically it implements an actor-critic with a textbook Bellman equation form (the value function at the current state equals the supremum over actions of the discounted future value plus current reward, under a risk-adjusted return utility), but with deep function approximators, continuous state and action spaces, and the full menagerie of real hedging instruments (forwards, swaps, futures, options) allowed. Three operational features matter. First, the policy is trained once and delivers an optimal hedge for *any* initial portfolio and market state without retraining — the agent generalises across the space of books rather than memorising a fixed one. Second, frictions (cost and liquidity) sit inside the Bellman equation so the optimisation naturally trades expected return against execution cost. Third, the authors prove existence of finite solutions to their Bellman equation and show how Deep Bellman Hedging reduces to the original Deep Hedging in the appropriate limit.

2023 — Mandelli, Pinciroli, Trapletti & Vittori — *RL for Credit Index Option Hedging*. A rare RL hedging paper evaluated on *real* market data rather than simulated paths, in an asset class — credit index options — where classical Black–Scholes delta hedging is known to be especially poor. Credit-index-option market makers cannot always offset positions in the secondary market because of low liquidity, so they end up with mixed books that need *delta* hedging on the underlying CDX or iTraxx index alone. The authors apply **Trust Region Volatility Optimization (TRVO)** — a risk-averse policy-gradient algorithm — to a vanilla credit-index option hedging environment with discrete time and transaction costs explicitly modelled. The reward function balances P&L volatility against transaction costs, and multiple agents are trained at different risk-aversion levels to span an efficient frontier. The killer result: the trained policy beats the practitioner’s Black–Scholes delta hedge on real CDX-style market data, demonstrating robust performance under model misspecification — which in credit-derivatives markets is the binding constraint, not the theoretical hedging optimum. This was also one of the first RL hedging papers explicitly framed for an industry CVA / capital-cost use case.

2024 — Neagu, Godin, Simard & Kosseim — *Deep Hedging with Market Impact*. The first deep-hedging paper to embed a realistic market-impact model directly inside the hedging environment. Almost every previous paper treats the hedging trade as a price-taking action — the agent’s order does not move the market — which is fine for liquid large-caps but catastrophic when hedging an exposure to an illiquid underlying where the hedger’s own rebalances move the price they are trying to hedge against. The authors propose a general DRL model that incorporates **convex market impacts** (the marginal cost of trading rises with size) and **impact persistence through time** (a trade made now still affects the price minutes later). They train feedforward NN policies and analyse them across simulations against both standard Black–Scholes delta hedging and Leland’s delta-hedging-with-costs benchmark. The trained DRL policies outperform both classical benchmarks in illiquid regimes, with the gap widening as liquidity gets worse — exactly the regime where naïve delta hedging breaks. The paper is the practical companion to the Murray / Buehler / Wood theoretical line.

2024 — Peng, Zhou, Xiao & Wu — *Risk-Sensitive Contract-Unified RL for Option Hedging*. Two of the field’s most painful operational problems attacked at once. First, a market maker hedging a *short* option position cares much more about tail losses than expected P&L — the bank cannot absorb a few catastrophic days regardless of how good the average looks. Second, real desks hedge thousands of contracts simultaneously and cannot afford to train a separate policy per strike, maturity, or underlying. The authors propose a PPO-based RL framework whose reward

function combines total hedging P&L with the *negative CVaR* of the final P&L at maturity, so the agent explicitly learns to minimise tail risk. Crucially, the policy is **contract-unified**: a single trained model applies across options with different initial underlying prices, strike prices, and maturities, conditioning on those as state features. Training uses simulated and historical stock and option-data trajectories with transaction costs embedded. In out-of-sample tests, the proposed policy delivers statistically significantly lower tail risk than standard benchmarks, demonstrating that contract-unification need not come at the cost of risk performance.

2025 — Lucius, Koch Jr., Starling, Zhu, Urena & Hu — *Deep Hedging with RL: A Practical Framework (BlackRock)*. The clearest signal yet that deep hedging is moving from research labs into industrial portfolio overlays — and that auditability and reproducibility are now first-class deliverables. Following the Buehler 2019 paradigm, Lucius and colleagues construct an end-to-end pipeline for dynamic hedging of equity index option exposures — specifically a normalised one-unit-delta SPX exposure offset via SPY trades — under realistic transaction costs and position limits. The simulator is built from a daily SPX/SPY panel dating back to 2005 with engineered state features (ATM implied volatility at 30 and 91 days, term-structure slope, 25-delta put/call skew, VIX, the 10-year Treasury yield, several realised-volatility horizons), chosen so the inputs are auditable rather than learned latents. A lightweight stochastic actor-critic agent with a GAE variant is trained with a cost-aware reward on a fixed train/validation/test split. On the held-out test sample, the GAE policy delivers higher point-estimate Sharpe than no-hedge, momentum, and volatility-targeting baselines — though the authors are careful to note that confidence intervals overlap a long-SPY benchmark, so they stop short of claiming formal dominance. Turnover stays controlled and the policy is robust to doubled transaction costs in stress tests. The paper’s most consequential contribution is methodological: the entire codebase, configuration, and experiment scripts are released for replication, setting an industry standard for auditable hedging-RL research.

5e. Open Gaps in Hedging

- **Full LOB-driven impact.** Neagu uses a parametric impact model. Nobody has yet plugged an ABIDES-style LOB simulator directly into a hedging environment.
- **Mixed exotic/vanilla portfolios.** Most papers hedge a single option or asset class. Real books mix vanilla, exotic, and OTC instruments across asset classes.
- **Vega and cross-gamma hedging.** Delta dominates the published literature. Higher-order Greeks and the cross-asset correlation structure that drives real book P&L are mostly unaddressed.

- **Regulatory model-risk approval.** Lucius 2025 emphasises auditability, but no published hedging-RL system has passed a public bank model-risk review at the rigour applied to Black–Scholes implementations.
- **Off-policy evaluation for hedging.** Backtests assume the agent's hedge trades don't move prices; agent-based simulators assume the simulator is right. Off-policy evaluation methods that quantify confidence in either are still missing from the hedging literature.

Section 6 — PORTFOLIO MANAGEMENT

6a. The Problem in Plain English

You hold £100 million and a universe of N assets — say, the 30 Dow Jones constituents, or a cryptocurrency basket, or a global ETF list. Every day, week, or month, you choose the portfolio weights, then watch the market evolve and pay transaction costs to rebalance. The classical answer is Markowitz mean–variance optimisation: solve a one-shot quadratic program. Reality has more moving parts than one snapshot — returns drift, correlations shift, volatilities cluster, and the cost of trading is itself state-dependent.

The portfolio-management problem in RL is the search for a *policy* that maps the current state of the market and the book to the next allocation, optimising a risk-adjusted return objective over many periods rather than a single moment in time.

6b. The RL Setup — Explained Intuitively

Yang et al.'s 2020 ensemble formulation is the canonical setup and the one nearly every FinRL tutorial uses.

- **State — what the agent sees.** Recent OHLCV bars (open, high, low, close, volume) for each stock in the universe, plus technical indicators — MACD, RSI, CCI, ADX. The state is essentially “a snapshot of the chart you’d see on a Bloomberg terminal for each stock.”
- **Action — what the agent decides.** Per-stock trade quantities — how many shares of each stock to buy or sell on this step. The implicit constraint is the cash balance.
- **Reward — what the agent is scored on.** The change in portfolio value from this step to the next, net of transaction costs. The economic logic is the simplest possible: maximise dollar P&L, with costs deducted in real time so the agent learns to be cost-aware. Risk control enters indirectly through the rolling Sharpe selector that picks among ensemble members.

6c. Roadmap of Key Papers

2020 → Yang et al.	PPO + A2C + DDPG ensemble.
2020 → Liu et al.	FinRL — the field's open-source substrate.
2021 → Wang & Yu (Vanguard)	Inverse optimisation + deep RL pipeline.
2022 → Liu et al. (FinRL-Meta)	Curated environments, survivorship controls.
2024 → Acero et al. (JPM)	ESG-aware multi-objective DRL portfolio.
2025 → Holzer et al.	7-algorithm ensemble, 2,048 parallel envs.

6d. Paper Deep-Dives

2020 — Yang, Liu, Zhong & Walid — *Deep RL for Automated Stock Trading: An Ensemble Strategy*. The canonical FinRL baseline for stock trading and the paper that established the “train several agents, pick the best by rolling Sharpe” template. Yang and colleagues train three actor–critic algorithms — **PPO, A2C, and DDPG** — on the same trading environment and combine them into an ensemble that selects, each quarter, the agent with the highest validation Sharpe to trade the next quarter. The state encodes recent OHLCV plus technical indicators (MACD, RSI, CCI, ADX); the action is per-stock trade quantities; the reward is portfolio value change net of transaction costs. They train on the 30 Dow Jones stocks using Compustat data via WRDS, with a load-on-demand technique to keep memory manageable for continuous-action training. The ensemble delivers a Sharpe ratio of 1.30 against A2C 1.12, PPO 1.10, DDPG 0.87, the Dow Jones benchmark, and a min-variance portfolio at Sharpe 0.45 — confirming both that no single deep-RL algorithm dominates and that simple ensembling captures most of the gain.

2020 — Liu, Yang, Chen, Zhang, Yang, Xiao & Wang — *FinRL: A DRL Library*. The field's most influential infrastructure paper — not because it introduces a new algorithm but because it builds the shared substrate everyone else now reproduces against. The library has a **three-layer architecture**: a market-environment layer wrapping datasets (NASDAQ-100, DJIA, S&P 500, HSI, SSE 50, CSI 300) into OpenAI-Gym-compatible environments at multiple time granularities; a DRL-agent layer with fine-tuned implementations of DQN, DDPG, PPO, SAC, A2C, and TD3; and an applications layer with reference implementations of single-stock trading, multiple-stock trading, and portfolio allocation. Transaction costs, market liquidity, and risk-aversion enter explicitly as environment parameters; reward functions and evaluation baselines are standardised. The deliberate target audience is beginners and reproducibility — every algorithm comes with a hands-on tutorial — and the design choice has been vindicated by

adoption: virtually every FinRL Contest submission and a substantial chunk of academic RL-in-finance papers since 2021 use FinRL as either training environment or baseline.

2021 — Wang & Yu — *Robo-Advising: Inverse Optimisation + Deep RL (Vanguard)*. The cleanest example of an industry-grade RL portfolio pipeline applied to real ETF allocation data. The architecture has two coupled ML agents. The first is an **inverse portfolio optimisation** agent that infers an investor’s risk preference and expected return *directly from historical allocation data* using online inverse optimisation — given a sequence of past allocations, what utility was being maximised? The second is a **deep RL** agent that aggregates the inferred sequence of expected returns to formulate a *multi-period* mean–variance portfolio optimisation problem and solves it via deep RL. The pipeline runs on real market data from 1 April 2016 to 1 February 2021, encompassing both the late-cycle bull market and the COVID shock, and the trained policy consistently outperforms the S&P 500 benchmark across this window. The authors attribute the gain to two compounding effects: explicit multi-period planning (versus single-period Markowitz) and a data-driven RL approach (versus classical parameter estimation). The Vanguard affiliation matters because it signals RL portfolio allocation has crossed from research into production at one of the world’s largest asset managers.

2022 — Liu, Xia, Yang, Gao, Zha, Zhu, Wang and collaborators — *FinRL-Meta: Market Environments and Benchmarks*. The second major infrastructure release from the FinRL group, focused on the simulation-to-reality gap rather than on a new algorithm. Where the original FinRL provided a small handful of environments, FinRL-Meta ships dozens of curated environments across stock trading, portfolio allocation, cryptocurrency trading, and high-frequency trading, each pre-processed with explicit attention to the methodological pitfalls that crater academic RL-in-finance results in practice: low signal-to-noise ratio (forcing realistic baselines rather than overfit numbers), survivorship bias (datasets cleaned to retain delisted firms), and overfitting (held-out test windows enforced at the environment level). The paper also formalises the “data-driven” paradigm: rather than treating the market simulator as a parameterised stochastic process, it treats it as a curated stream of historical data with engineered guardrails. FinRL-Meta is the substrate the annual ICAIF FinRL Contests are built on, and the most-cited benchmarking paper in financial RL after the Hambly survey.

2024 — Acero, Zehtabi, Marchesotti, Cashmore, Magazzeni & Veloso — *DRL + Mean-Variance for Responsible Portfolio Optimisation (JPM)*. The JPM lab’s first serious attempt to embed ESG constraints inside a deep-RL portfolio agent — significant because regulator-friendly responsible-investing constraints are now binding for most institutional asset managers. The authors

extend the standard mean-variance formulation by adding **ESG state variables and ESG-aware objectives** into the MDP, then train deep-RL portfolio policies that jointly optimise financial performance and ESG scores. They compare against modified mean-variance approaches that incorporate ESG via constraints or penalty terms, across both additive and multiplicative utility functions of financial return and ESG score. The headline result is that deep-RL policies deliver competitive performance against mean–variance approaches for responsible portfolio allocation across the tested utility specifications — establishing that RL can absorb non-standard constraints without losing performance. The paper is most useful read as a JPM signal about which directions the lab considers important: regulator-aware constraints, multi-objective optimisation, and explainable policies that a compliance officer can defend.

Get Quang Khải Nguyễn Hưng’s stories in your inbox

Join Medium for free to get updates from this writer.

Remember me for faster sign in

2025 — Holzer, Wang, Xiao & Liu — *Revisiting Ensemble Methods at the FinRL Contests*. The most operationally focused FinRL paper to date — its contributions are computational rather than algorithmic, and they matter because they make ensemble training tractable at industry scale. The authors revisit the original Yang 2020 ensemble idea (PPO + A2C + DDPG) and extend it in two directions. First, the algorithm pool expands to **seven actor–critic methods** (PPO, A2C, DDPG, ACKTR, SAC, TD3, TRPO) so the rolling-Sharpe selector has more diverse policies to choose from. Second, and more importantly, training runs on a single GPU with up to **2,048 parallel environments**, delivering a 1,746× sampling speedup over single-environment training. On both stock and cryptocurrency trading tasks from the ICAIF FinRL Contests 2023 and 2024, the GPU-accelerated ensemble achieves higher cumulative returns than the component agents, reducing maximum drawdown by up to 4.17% and improving Sharpe ratio by up to 0.21. The paper’s lasting importance is that it shows what previously took days of distributed training can now run on a single GPU overnight, removing one of the main practical barriers to research iteration on ensemble RL trading systems.

6e. Open Gaps in Portfolio Management

- **Out-of-distribution regime change.** Most published Sharpe ratios above 2 vanish when the test period crosses a regime change (2008, COVID, 2022 rates shock). No RL portfolio paper has cleanly demonstrated robustness to this.
- **Multi-asset-class allocation.** Equities-only is the dominant

testbed. Real institutional portfolios mix equities, bonds, FX, commodities, alternatives — and RL papers spanning all four are rare.

- **Realistic transaction costs.** Most papers use linear or fixed cost models. Real costs are nonlinear, market-state-dependent, and venue-dependent.
- **The “if it worked we’d trade it” problem.** The FinRL Contest organisers themselves [Holzer 2025] flag the dilemma honestly: published Sharpe ratios that would generate billions of dollars are not actually being traded by the authors.
- **Behavioural constraints.** Real investors have taxes, withdrawals, drawdown thresholds. RL portfolio papers rarely embed any of these in the reward.

Section 7 — ALPHA GENERATION

7a. The Problem in Plain English

Before any execution or hedging problem can arise, a quant fund needs a *signal* — a way of transforming raw historical price and volume data into a prediction of future returns. For decades, quants did this with armies of researchers writing **formulaic alphas** like $\text{rank}(\text{close} - \text{open}) / (\text{high} - \text{low})$, grading each formula by its *Information Coefficient* (the correlation between its predictions and realised forward returns), and keeping the best. Doing this by hand is slow and biased; doing it with deep neural networks gives you opaque black-box predictors that risk teams cannot defend.

The RL framing is: treat the space of all formulaic expressions as a tree, and train an agent to *search* it efficiently for combinations of factors that together predict returns. The result is interpretable, auditable signal discovery at machine scale — readable expressions a compliance officer can sign off on.

7b. The RL Setup — Explained Intuitively

The canonical formulation is alphagen (Yu et al. 2023), and it is the cleanest place to anchor.

- **State — what the agent sees.** The partial mathematical expression generated so far, written in reverse Polish notation. Think of it as a stack: the agent has been building up tokens and the state is the current contents of that stack.
- **Action — what the agent decides.** The next token to append — an *operator* (like rank, mean, correlation), an *operand* (a raw input feature like close, volume, vwap), or a *constant* (a number).

Importantly, the action space is masked to prevent syntactically illegal expressions.

- **Reward — what the agent is scored on.** The *marginal* Information Coefficient improvement when the newly completed alpha is added to a *pool* of existing alphas, with weights jointly optimised. This is alphagen’s key insight: rewarding the synergy of an alpha-set rather than a single alpha forces the agent to find factors that *complement* what is already in the pool, rather than producing yet another correlated variant of momentum. The economic logic is that a portfolio of low-correlation alphas is what actually drives performance.

7c. Roadmap of Key Papers

2023 → Yu et al. (alphagen)	PPO + masking; synergistic alpha set reward.
2023 → Nagy, Calliess & Zohren	APEX async DDDQN for LOB signal execution.
2024 → Zhao et al. (QFR)	Variance-bounded REINFORCE; IR shaping.
2024 → Xu et al. (Alpha²)	Logical-operator grammar + tree search.
2025 → Shi, Duan & Li	LLM-seeded MCTS for formula search.
2025 → Chen & Kawashima	PPO reweights LLM-generated alphas.

7d. Paper Deep-Dives

2023 — Yu, Xue, Ao, Pan, He, Tu & He — *Generating Synergistic Formulaic Alpha Collections via RL* (alphagen).

The canonical RL alpha-mining paper and the reference every subsequent paper builds on. The key insight is that alphas are useful only when *combined*: a single alpha factor that beats noise is good, but a *set* of alphas with low mutual correlation is what actually drives portfolio performance. Most prior approaches mined alphas one at a time and only worried about correlation post hoc. Alphagen flips this: the RL agent’s reward depends on the IC and Rank-IC of a *weighted combination* of all alphas in the current pool, so the agent is directly incentivised to generate synergistic alphas rather than redundant ones. The expression-generation process is modelled as a non-stationary MDP where each state is the partial expression generated so far in reverse Polish notation, each action appends the next token (an operator like rank, an operand like close, or a constant), and reward is the marginal IC improvement after adding the completed alpha to the pool. They use PPO with an Invalid Action Masking mechanism that prevents the agent from producing syntactically illegal expressions, with 6 raw input features (open, close, high, low, volume, vwap) and a 20-day forward-return

target. On CSI300 and CSI500 Chinese A-shares the framework achieves the highest IC and Rank-IC against PPO baselines, genetic programming, MLP, LightGBM, and XGBoost — and crucially, the synergy effect means performance keeps improving as the pool grows, unlike competitors where it plateaus at one alpha.

2023 — Nagy, Calliess & Zohren — *Asynchronous Deep Double Duelling Q-Learning for Trading-Signal Execution in LOB Markets*. The outlier paper that sits at the unusual intersection of signal generation and low-latency execution. Most signal-mining papers stop once they have a predictive alpha — but a signal only generates P&L if it can be traded fast enough to capture the predicted move, and microstructure-level execution matters enormously when the signal horizon is seconds rather than days. The solution is **APEX-based asynchronous Double Duelling DQN** trained inside the ABIDES limit-order-book simulator on NASDAQ message data. The agent observes microstructure features (top-of-book, depth, recent trade signs, signal-derived predictors) and decides discrete order actions (place / cancel / modify at chosen price levels); the asynchronous architecture allows many actors to collect experience in parallel while a separate learner updates the value function. The paper sits in the Oxford-Man line of LOB-RL work alongside JAX-LOB and the broader Zohren signal-processing programme, and it is the cleanest published demonstration that RL can take a short-horizon predictive signal all the way to executable order placement.

2024 — Zhao, Zhang, Qin & Yang — *QuantFactor REINFORCE*. QFR attacks alphagen's biggest weakness — that PPO, while a fine general-purpose RL algorithm, is poorly suited to alpha mining specifically. The authors argue that the formulaic-alpha MDP has unusual structure: the transition function is essentially Dirac (the next state is *deterministically* the current state plus the chosen token), which violates assumptions PPO's clipped objective is designed to handle, and at the same time means the high-variance penalty that usually rules out vanilla REINFORCE doesn't bite as hard. They drop the actor-critic structure entirely and use a variant of pure REINFORCE — **QuantFactor REINFORCE (QFR)** — with a custom dedicated baseline that they prove theoretically bounds variance, and an **Information Ratio reward shaping** mechanism that encourages the agent to find alphas whose performance is steady across market regimes rather than alphas that work brilliantly in one period and fail in another. The result, evaluated against the same Chinese A-shares setup as alphagen, is a 3.83% boost in correlation with returns and meaningfully faster convergence than PPO-based alphagen — providing one of the cleanest demonstrations that algorithm-design choices matter a lot once you understand the structural quirks of the financial MDP.

2024 — Xu, Yin, Zhang, Liu, Jiang and collaborators — *Alpha²*:

Logical Formulaic Alphas via Deep RL. Alpha² extends the alphagen paradigm to handle a richer expression grammar. Where alphagen's operators are restricted to arithmetic and ranking operations, Alpha² allows **logical operators** — `if-then-else`, comparison operators, boolean combinations — letting the agent construct alphas of the form "if volatility is high and the cross-sectional rank of momentum is in the bottom quintile, then short". The search procedure adds a tree-search step on top of the RL policy, so that high-promise partial expressions are expanded deeper, yielding tighter interpretability than alphagen at comparable predictive performance. The paper is part of the broader 2024 push toward more expressive but still readable alpha grammars — an explicit response to the criticism that pure arithmetic alphas are too restrictive to capture the conditional logic professional quants actually use, while neural-network "black-box" alphas are too opaque to deploy in risk-controlled environments.

2025 — Shi, Duan & Li — *Navigating the Alpha Jungle: LLM-Powered MCTS for Factor Mining*. The cleanest demonstration of LLM-RL hybridisation in the alpha-mining domain. Shi and colleagues observe that pure RL agents, even with rich action masking, still spend enormous compute searching parts of the formula space that no experienced quant would ever explore — the trees are vast and most branches are obviously useless. Their fix is to use an **LLM as a seed-formula proposer**: the language model takes natural-language descriptions of market context and historically-successful formula patterns, produces a small set of candidate expression seeds, and a **Monte Carlo Tree Search (MCTS)** procedure then refines and prunes those seeds using actual backtest IC as the value signal. This LLM-guided initial direction collapses the effective search space dramatically while preserving the rigour of explicit IC-based evaluation. The paper sits at the intersection of two 2024–2026 frontier themes — LLM augmentation and MCTS-based search — and demonstrates that the combination produces interpretable formulaic alphas faster than either approach alone.

2025 — Chen & Kawashima — *Adaptive Alpha Weighting with PPO*. A different LLM-RL integration: assuming you have a stable of LLM-generated alphas, how do you combine them when the underlying market regime is shifting? The answer is to use **PPO to dynamically reweight** LLM-generated formulaic alphas as a function of observed market state. The pipeline is direct: an LLM proposes a corpus of candidate alphas via prompting; their individual IC scores are computed; a PPO agent then learns a state-conditional weighting policy that allocates more weight to alphas performing well in the current regime and downweights those that decay. The paper sits in the practical tail of the LLM-RL alpha-mining wave — rather than trying to invent new alphas, it accepts the LLM's proposals as given and optimises the

combination layer, which is closer to how real multi-strategy quant funds operate. The combination of cheap LLM generation and RL-based adaptive weighting points toward a likely production architecture for the next generation of formulaic-alpha systems.

7e. Open Gaps in Alpha Generation

- **Alpha decay.** A discovered alpha that’s already known to other funds will not generate P&L. Almost no published RL alpha-mining paper tests this — they evaluate IC in held-out periods but not against the universe of known formulaic alphas competitors already use.
- **Joint discovery and execution.** Nagy 2023 is the only paper that links signal generation to execution. The other five stop at IC, leaving “can this be traded?” entirely open.
- **LLM hallucination of factors.** Shi 2025 and Chen 2025 trust the LLM proposer. The risk that LLMs produce plausible-looking but spurious formulas — pattern-matching to financial vocabulary without economic content — has not been quantified.
- **Cross-market alpha transfer.** All published RL alpha-mining is on Chinese A-shares. Whether the discovered alphas transfer to US, European, or emerging markets is open.
- **Risk-controlled alpha discovery.** Most papers maximise IC. Few constrain on factor crowding, sector exposure, or drawdown — the constraints real funds actually impose.

Section 8 — ROBO-ADVISING

8a. The Problem in Plain English

A retail investor opens a digital advisor app and wants her savings managed. The advisor faces two problems at once. First, it doesn’t know how risk-averse she truly is — and asking too many questions is annoying and reveals little, while asking too few leaves the advisor allocating blind. Second, the investor herself doesn’t really know her own risk aversion with any precision; what she has is a sequence of past portfolio choices that *implicitly* reveal her preferences.

The robo-advising problem is to learn the investor’s preferences from observed behaviour over time while simultaneously delivering a good allocation. It is RL with a human in the loop, and the loop costs money in two ways: bad allocations cost returns, and intrusive solicitations cost engagement. The classical answer is a static risk-tolerance questionnaire feeding a Markowitz optimiser. The RL question is whether an agent can do better by *learning*

continuously from each portfolio choice the investor makes.

8b. The RL Setup — Explained Intuitively

Alsabah et al.’s 2019 formulation is the template. It is honest about the human-in-the-loop structure.

- **State — what the agent sees.** Two pieces: (1) the advisor’s current *belief* about the investor’s true risk-aversion parameter, represented as a probability distribution that gets refined over time; and (2) the current market environment.
- **Action — what the agent decides.** A discrete choice between two costly options on each step. *Solicit* — ask the investor for a portfolio choice, which is informative but costs engagement — or *autonomously trade* based on the current stale belief, which is free but suboptimal if the belief is wrong.
- **Reward — what the agent is scored on.** Realised investor utility (a function of returns weighted by the *true* risk aversion) minus a solicitation cost when the advisor chooses to ask. The economic logic is direct: the advisor needs to balance the long-run gain from a more accurate preference estimate against the short-run engagement cost of asking. Solicitations are valuable but priced.

8c. Roadmap of Key Papers

2019 → Alsabah, Capponi et al.	PAC-MDP for costly preference solicitation.
2020 → Dixon & Halperin	G-Learner + GIRL — goal-based RL/IRL pair.
2021 → Wang & Yu (Vanguard)	Inverse opt + deep RL on real ETF data.
2023 → Hambly, Xu & Yang	The survey that codified the domain.

8d. Paper Deep-Dives

2019 — Alsabah, Capponi, Ruiz Lacedelli & Stern — *Robo-Advising: Learning Investors’ Risk Preferences via Portfolio Choices*. The foundational paper that gave the robo-advising RL literature its template, published as arXiv:1911.02067 and subsequently in the *Journal of Financial Econometrics*. The setup is honest about the human-in-the-loop structure: the robo-advisor does not know the investor’s risk preference but learns it over time by observing her portfolio choices across different market environments. The advisor faces an **exploration–exploitation trade-off** between two costly actions — *soliciting* a portfolio choice from the investor (which costs engagement but refines the belief about her risk aversion) or *autonomously trading* based on the current stale estimate (which is free but suboptimal if the estimate is wrong). The authors develop a PAC-MDP-style algorithm and

prove its value function converges to the optimal value function of an *omniscient* robo-advisor — one who knew the true risk aversion all along — over a number of periods polynomial in the state and action space sizes. They also show that by *correcting for the investor's own mistakes* (her allocations may be inconsistent or noisy), the robo-advisor can outperform the standalone investor regardless of her opportunity cost for making portfolio decisions herself.

2020 — Dixon & Halperin — *G-Learner and GIRL: Goal-Based Wealth Management with RL*. Dixon and Halperin reframe the problem from “risk aversion” to “financial goals” — closer to how real retail investors actually think. An investor isn’t usually maximising abstract utility; she is trying to reach a retirement number, or fund a child’s tuition, or hit a house-deposit target by a deadline. The authors introduce two algorithms. **G-Learner** is a reinforcement-learning method built on *G-learning* (a probabilistic extension of Q-learning) that operates with explicitly defined one-step rewards, makes no parametric assumption about the data-generating process, and is robust to noisy financial data; when applied to a quadratic reward and a Gaussian reference policy, G-learning reduces analytically to an entropy-regulated Linear Quadratic Regulator, providing a clean theoretical interpretation. **GIRL** (“G-Inverse Reinforcement Learning”) is the companion *inverse* RL algorithm that recovers an investor’s implicit goal-based reward function from her observed past allocation choices — the natural complement that closes the personalisation loop. Together they form the field’s reference implementation of goal-based wealth management: G-Learner *acts*, GIRL *learns the goal*, and the pair handles both periodic contributions during the working years and periodic withdrawals during retirement.

2021 — Wang & Yu — *Robo-Advising: Inverse Optimisation + Deep RL (Vanguard)*. The most explicitly industry-facing paper in this list and the closest published evidence of what production robo-advisor systems at major asset managers look like. The architecture has two coupled ML agents. The first, an **inverse portfolio optimisation** agent, infers an investor’s risk preference and expected return *directly from historical allocation data* using online inverse optimisation — given a sequence of past portfolio decisions, what utility was being maximised? The second, a **deep RL** agent, takes the inferred sequence of expected returns and solves a multi-period mean–variance portfolio optimisation problem via deep RL. Applied to real market data from 1 April 2016 to 1 February 2021 — a window that spans both late-cycle bull markets and the COVID shock — the integrated pipeline consistently outperforms the S&P 500 benchmark portfolio. The authors attribute the gain to two compounding effects: explicit multi-period planning (versus single-period Markowitz) and a data-driven RL approach (versus classical parameter estimation). The Vanguard

affiliation is the most consequential signal: RL robo-advising has crossed from academic research into deployed production at one of the world’s largest asset managers.

2023 — Hambly, Xu & Yang — *Recent Advances in Reinforcement Learning in Finance*. Not strictly a robo-advising paper but a survey of the whole field — and it earns its place here because it codified robo-advising as a first-class problem in financial RL, on equal footing with optimal execution, market making, hedging, portfolio optimisation, and smart order routing. The survey reviews the Alsabah–Capponi framework, the Dixon–Halperin goal-based formulation, the Vanguard inverse-optimisation pipeline, and the broader inverse-RL literature for preference inference, and locates them within a stochastic-control framing that makes them tractable for readers from the mathematical-finance tradition. For anyone entering robo-advising RL from a math-finance background, this is the highest-quality single-document map of the area — the paper that translates the engineering-flavoured RL literature into language a quant trained on stochastic control can actually use.

8e. Open Gaps in Robo-Advising

- **Goal-based vs utility-based formulations.** Dixon–Halperin’s goal framing has not been combined cleanly with Alsabah’s costly-solicitation framework. The integrated system that handles both has not been published.
- **ESG and personalised non-financial constraints.** How does the agent learn that a user excludes tobacco stocks or wants a religious-finance constraint? Currently absent from the RL literature.
- **Behavioural drift over time.** Risk preferences are not stable — they change with age, life events, market regime. Static belief models miss this. Adaptive preference-learning RL is an open direction.
- **Trust calibration.** An advisor that is too confident is dangerous; how should an RL advisor *express uncertainty* to a human user without driving them to a competitor? This is a UX-meets-RL problem essentially no published paper addresses.
- **Real deployment evidence.** Wang & Yu 2021 is the only paper with industry affiliation. Independent academic confirmation of these results on production data is missing.

Section 9 — Cross-Cutting Gaps in RL for Finance

Step back from any single domain and a set of field-wide unsolved problems comes into focus. They cut across execution and market making and hedging and portfolio management; they are why the field still has frontier work to do.

The first is the **simulation-to-reality gap**, and it is the master problem. Every published RL trading system was trained in some kind of simulator — a market replay, an agent-based LOB, a stochastic-volatility path generator — and then evaluated either inside that same simulator or on a held-out historical window where the agent’s own orders, by construction, could not affect the price tape. Live markets do neither of these things. The agent’s own orders move prices, other market participants react to those moves, and the dynamics that a backtest treated as exogenous become endogenous. The field has three partial responses to this — higher-fidelity simulators (ABIDES, ABIDES-MARL, Queue-Reactive Models), offline RL methods that learn purely from logged data (CQL, IQL, Decision Transformers), and adversarial training to harden policies against the worst-case unseen counterparty — and none of them has cleanly closed the gap.

The second is the **benchmarking problem**. There is no “ImageNet for trading.” When a robotics RL paper claims a result on MuJoCo’s HalfCheetah, every other researcher can rerun that experiment and check. When a financial RL paper claims a Sharpe of 2.3, the data split, the transaction-cost model, the survivorship-bias treatment, and the rebalancing frequency are all paper-specific, and cross-paper comparison is nearly impossible. FinRL-Meta and the ICAIF FinRL Contests are the closest the field has to a standardised comparison, and LOB-Bench is a promising recent step for generative-model evaluation on order-book data, but none of them yet plays the role MuJoCo or Atari did for general RL.

The third is the **offline-RL versus live-exploration trade-off**. Online RL — where the agent explores by taking real actions and observing real consequences — is the natural setup, but every “let’s see what happens” order has real P&L cost, so live exploration is brutally expensive. Offline RL — learning purely from a fixed historical dataset — sidesteps that cost but inherits the dataset’s biases and can dangerously over-extrapolate when deployed. The field’s current consensus is that production RL systems are trained offline or in simulators, fine-tuned conservatively, and deployed with tight risk limits — but the right balance between exploration cost and offline-learning bias is unsolved.

The fourth is **tacit collusion and regulatory risk**, the unease that Cont & Xiong’s 2023 *Mathematical Finance* paper introduced into the field. If a population of decentralised RL market makers can converge to wide-spread equilibria without any explicit coordination, then deploying RL trading systems at scale is not a purely technical question — it is a market-structure question that financial regulators (FCA, SEC, ESMA) are likely to ask. The research agenda this opened — how to design RL training procedures that are *robust* to emergent collusion, and how to monitor live RL populations for it — is barely begun.

The fifth, and the most uncertain, is **LLM-RL convergence**. The 2024–2026 wave of LLM-augmented trading agents (FinMem, FinAgent, FinCon, FinRLlama, alpha-mining systems with LLM proposers and MCTS refinement) is genuinely novel, in that the LLMs are not just sentiment features but are first-class components: memory modules, factor proposers, policy backbones. Whether this represents the next paradigm — analogous to ImageNet → AlexNet — or a fashionable detour that fails on out-of-distribution market regimes is still open. The bar for declaring LLM-RL superior to classical RL has so far been set very low, because the comparisons are usually against pure DRL agents that were already inferior to buy-and-hold on many test windows. A more rigorous evaluation against ICAIF FinRL Contest winners or industrial benchmarks would tell us more.

The honest read is that the field has matured past the “does RL work in finance” question and is now in a “does it generalise, does it scale, is it auditable” phase — the same maturation curve deep learning went through in computer vision around 2017–2018.

Section 10 — Glossary

A short field guide. Pick up any paper after this and the vocabulary should decode itself.

MDP / POMDP. A *Markov Decision Process* is the formal mathematical container in which RL operates: states, actions, transition probabilities, rewards, and a Markov assumption that the future depends on the past only through the current state. A *Partially Observable* MDP relaxes the assumption that the agent sees the full state — in finance this is the realistic case, since the agent never observes the full latent state of the market. *Paper example:* alphagen frames formula generation as a non-stationary MDP where the state is the partial expression so far.

Implementation shortfall. The gap between the price a trader would have got if she could have transacted instantly at the decision-time benchmark price, and the average price she actually achieved after slicing and dicing the order over time. The universal yardstick of optimal execution research. *Paper example:* Ning et al. 2018 use negative implementation shortfall plus a quadratic impact penalty as their Double-DQN reward.

Almgren–Chriss framework. The 2000 stochastic-control solution to optimal execution under linear permanent and temporary impact, which gives an analytical schedule trading off market impact against timing risk. Almost every RL execution paper benchmarks against AC or extends it. *Paper example:* Hendricks & Wilcox 2014 use AC as a prior and train Q-learning to deviate from it, improving implementation shortfall by 10.3% on JSE Top 40 stocks.

Avellaneda–Stoikov model. The 2008 stochastic-control solution to market making in which the market maker quotes around a

reservation price that shifts with her inventory. The classical baseline every RL market-making paper either reproduces or beats. *Paper example*: Spooner & Savani 2020 turn the AS single-agent model into a zero-sum game between the market maker and an adversary.

Inventory penalty. A reward-shaping term that punishes the market-making agent for accumulating positions away from zero. Spooner et al.'s *asymmetrically dampened* inventory penalty — punishing inventory-growing trades harder than rewarding inventory-flattening ones — is the field's signature design choice and has been adopted almost universally.

Market impact. The price movement caused by the act of trading. *Permanent* impact persists; *temporary* impact decays. Real impact is *transient* — it decays over a kernel that may span hours. *Paper example*: Micheli & Monod 2024 explicitly track the recent history of trades weighted by a decay kernel inside the DDPG state.

Adverse selection. The risk that the counterparty hitting your quote knows something you don't — informed traders buy just before prices rise and sell just before they fall, so a market maker who doesn't adjust gets *long the losers and short the winners*. *Paper example*: Jafree, Jain & Firoozye 2025 train an RL market maker to detect and exploit the price drift induced by a TWAP-style meta-order — the explicit study of adverse selection.

LOB (limit order book). The data structure that holds resting orders at each price level on both the bid and ask sides. RL market-making and execution papers consume LOB snapshots or message-level events as state input. *Paper example*: Guo et al.'s Attn-LOB uses a convolution-plus-attention network to extract features directly from the raw multi-level LOB.

Mid-price. The arithmetic average of the best bid and best ask. The reference around which market makers quote and against which execution agents measure their fills.

Spread. The difference between the best ask and the best bid. The market maker's revenue per round-trip trade; the execution trader's cost per crossing of the book.

TWAP / VWAP. *Time-Weighted Average Price* — slice the parent order evenly over time. *Volume-Weighted Average Price* — slice in proportion to expected volume over time. These are the industry's lowest-bar execution baselines, and “beat TWAP” is the minimum claim any execution RL paper makes. *Paper example*: Ning et al. 2018 establish TWAP-beating as the benchmarking convention for DRL execution.

Actor-critic. An RL architecture with two networks. The *actor* outputs the policy (the action to take given a state); the *critic* outputs the value (expected discounted future reward from this state). PPO, A2C, SAC, TD3, DDPG are all actor-critic algorithms

and they dominate continuous-action finance problems. *Paper example*: Murray et al. 2022 derive a risk-averse Bellman representation of Deep Hedging and train an actor-critic agent across multiple risk-aversion levels simultaneously.

Offline RL / off-policy. Offline RL learns a policy entirely from a fixed historical dataset of (state, action, reward, next-state) tuples, with no live exploration. Conservative Q-Learning (CQL) and Implicit Q-Learning (IQL) are the canonical algorithms. Sun, An & Wang's 2023 ACM TIST survey argues quantitative trading is naturally an offline RL setting because live exploration is prohibitively expensive.

Simulator fidelity / sim-to-real gap. Most published RL trading systems are trained in simulators where the agent's orders don't affect the price tape (market replay) or where the simulator itself is a stylised approximation of reality (agent-based, Queue-Reactive). The *sim-to-real gap* is the systematic difference between simulator performance and live performance, and it is the field's central methodological concern. *Paper example*: FinRL-Meta 2022 is the field's most explicit infrastructure response to the simulation-to-reality gap.

MARL (Multi-Agent RL). RL where multiple learning agents interact in a shared environment, each pursuing its own reward. Real markets are multi-agent by construction. *Paper example*: Vadori et al.'s 2024 *Mathematical Finance* paper trains two heterogeneous populations (liquidity providers and liquidity takers) with shared-policy MARL to produce an emergent FX dealer market.

Reward shaping. Adding auxiliary terms to the raw reward signal to guide the agent toward desired behaviour without changing the optimal policy. Inventory penalties, transaction-cost penalties, risk-aversion terms, and Information Ratio shaping are all reward-shaping choices. *Paper example*: Zhao et al. 2024's QFR uses Information Ratio reward shaping to encourage alphas whose performance is steady across regimes.

Deep Hedging. The Buehler et al. 2019 framework — recasting option hedging as a neural-network optimisation that minimises a convex risk measure of terminal P&L net of frictions. Now a generic term for any NN-trained dynamic hedging policy. *Paper example*: Lucius et al. 2025's BlackRock-affiliated paper releases a full Deep Hedging pipeline with engineered SPX/SPY state features and full reproducibility scripts.

Alpha factor. A short mathematical expression that predicts future returns. Evaluated by *Information Coefficient* (the rank correlation between predictions and realised forward returns). Sets of *low-correlation* alphas combined are what actually drive returns; this is alphas' reason for existing.

Formulaic alpha. An alpha factor expressible as a readable mathematical formula (e.g., $\text{rank}(\text{close} - \text{open}) / (\text{high} - \text{low})$), as opposed to a neural-network black-box predictor. The interpretability is non-negotiable in alpha mining because risk teams need to defend production signals.

Section 11 — References

A bibliography of the papers cited in this article, drawn from the attached paper summaries and research report.

Optimal Execution

- [Nevmyvaka 2006] Reinforcement Learning for Optimized Trade Execution. ICML 2006.
- [Hendricks 2014] A Reinforcement Learning Extension to the Almgren-Chriss Framework for Optimal Trade Execution. IEEE CIFER 2014.
- [Ning 2018] Double Deep Q-Learning for Optimal Execution. arXiv:1812.06600. *Applied Mathematical Finance*.
- [Karpe 2020] Multi-Agent Reinforcement Learning in a Realistic Limit Order Book Market Simulation. ACM ICAIF 2020. arXiv:2006.05574.
- [Fang 2021] Universal Trading for Order Execution with Oracle Policy Distillation. AAAI 2021. arXiv:2103.10860.
- [Fang 2022] Imitate then Transcend: Multi-Agent Optimal Execution with Dual-Window Denoise PPO. arXiv:2206.10736.
- [Macri 2024] Reinforcement Learning for Optimal Execution When Liquidity is Time-Varying. *Applied Mathematical Finance* 31(5):312–342. arXiv:2402.12049.
- [Micheli 2024] Deep Reinforcement Learning for Online Optimal Execution Strategies. arXiv:2410.13493.
- [Espana 2025] Reinforcement Learning in Queue-Reactive Models: Application to Optimal Execution. arXiv:2511.15262.

Market Making

- [Spooner 2018] Market Making via Reinforcement Learning. AAMAS 2018. arXiv:1804.04216.
- [Ganesh 2019] Reinforcement Learning for Market Making in a Multi-agent Dealer Market. NeurIPS 2019 Workshop. arXiv:1911.05892.
- [Spooner 2020] Robust Market Making via Adversarial Reinforcement Learning. IJCAI 2020. arXiv:2003.01820.
- [Gašperov 2022] Deep Reinforcement Learning for Market Making Under a Hawkes Process-Based Limit Order Book Model. *IEEE*

Control Systems Letters 6:2485–2490. arXiv:2207.09951.

- [Guo 2023] Market Making with Deep Reinforcement Learning from Limit Order Books (Attn-LOB). arXiv:2305.15821.
- [Cont 2023] Dynamics of Market Making Algorithms in Dealer Markets: Learning and Tacit Collusion. *Mathematical Finance* 34(2):467–521.
- [Vadori 2024] Towards Multi-Agent Reinforcement Learning-Driven Over-the-Counter Market Simulations. *Mathematical Finance* 34(2):262–347. arXiv:2210.07184.
- [Jafree 2025] When AI Trading Agents Compete: Adverse Selection of Meta-Orders by Reinforcement Learning-Based Market Making. arXiv:2510.27334.
- [Zimmer 2025] Reinforcement Learning-Based Market Making as a Stochastic Control on Non-Stationary Limit Order Book Dynamics. arXiv:2509.12456.

Hedging

- [Buehler 2019] Deep Hedging. *Quantitative Finance* 19(8):1271–1291. arXiv:1802.03042.
- [Du 2020] Deep Reinforcement Learning for Option Replication and Hedging. *The Journal of Financial Data Science* 2(4):44–57.
- [Buehler 2021] Deep Hedging: Learning to Remove the Drift under Trading Frictions with Minimal Equivalent Near-Martingale Measures. arXiv:2103.11948. *Risk* magazine 2022.
- [Murray 2022] Deep Hedging: Continuous Reinforcement Learning for Hedging of General Portfolios across Multiple Risk Aversions. ACM ICAIF 2022. arXiv:2207.07467.
- [Buehler 2022] Deep Bellman Hedging. arXiv:2207.00932.
- [Mandelli 2023] Reinforcement Learning for Credit Index Option Hedging. arXiv:2307.09844.
- [Neagu 2024] Deep Hedging with Market Impact. arXiv:2402.13326. CAIAC 2024.
- [Peng 2024] A Risk-Sensitive Contract-Unified Reinforcement Learning Approach for Option Hedging. arXiv:2411.09659.
- [Lucius 2025] Deep Hedging with Reinforcement Learning: A Practical Framework for Option Risk Management. arXiv:2512.12420.

Portfolio Management

- [Yang 2020] Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy. ACM ICAIF 2020.
- [Liu 2020] FinRL: A Deep Reinforcement Learning Library for Automated Stock Trading in Quantitative Finance. NeurIPS 2020

Deep RL Workshop. arXiv:2011.09607.

- [Wang 2021] Robo-Advising: Enhancing Investment with Inverse Optimization and Deep Reinforcement Learning. arXiv:2105.09264.
- [Liu 2022] FinRL-Meta: Market Environments and Benchmarks for Data-Driven Financial Reinforcement Learning. NeurIPS 2022 D&B. arXiv:2211.03107.
- [Acero 2024] Deep Reinforcement Learning and Mean-Variance Strategies for Responsible Portfolio Optimization. arXiv:2403.16667.
- [Holzer 2025] Revisiting Ensemble Methods for Stock Trading and Crypto Trading Tasks at ACM ICAIF FinRL Contests 2023/2024. arXiv:2501.10709.

Alpha Generation

- [Yu 2023] Generating Synergistic Formulaic Alpha Collections via Reinforcement Learning (alphagen). KDD 2023. arXiv:2306.12964.
- [Nagy 2023] Asynchronous Deep Double Duelling Q-Learning for Trading-Signal Execution in Limit Order Book Markets. arXiv:2301.08688.
- [Zhao 2024] QuantFactor REINFORCE: Mining Steady Formulaic Alpha Factors with Variance-Bounded REINFORCE. arXiv:2409.05144.
- [Xu 2024] Alpha²: Discovering Logical Formulaic Alphas Using Deep Reinforcement Learning. arXiv:2406.16505.
- [Shi 2025] Navigating the Alpha Jungle: An LLM-Powered MCTS Framework for Formulaic Factor Mining. arXiv:2505.11122.
- [Chen 2025] Adaptive Alpha Weighting with PPO: Enhancing Prompt-Based LLM-Generated Alphas. arXiv:2509.01393.

Robo-Advising

- [Alsabah 2019] Robo-Advising: Learning Investors' Risk Preferences via Portfolio Choices. arXiv:1911.02067. *Journal of Financial Econometrics* 19(2):369–392 (2021).
- [Dixon 2020] G-Learner and GIRL: Goal Based Wealth Management with Reinforcement Learning. arXiv:2002.10990. *Risk* magazine 2021.
- [Wang 2021] (see Portfolio Management above).
- [Hambly 2023] Recent Advances in Reinforcement Learning in Finance. *Mathematical Finance* 33(3):437–503. arXiv:2112.04553.

Surveys and Infrastructure

- [Pippas 2025] The Evolution of Reinforcement Learning in Quantitative Finance: A Survey. *ACM Computing Surveys*.

arXiv:2408.10932.

- [Bai 2025] A Review of Reinforcement Learning in Financial Applications. *Annual Review of Statistics and Its Application* 12:209–232.
- [Sun 2023] Reinforcement Learning for Quantitative Trading. *ACM Transactions on Intelligent Systems and Technology* 14(3).

What to Read Next

If this article was your first pass at the field, three doors lead deeper in. The single most useful starting point is **Hambly, Xu & Yang's 2023 *Mathematical Finance* survey** ([arXiv:2112.04553](https://arxiv.org/abs/2112.04553)), which translates the engineering-flavoured RL literature into language any reader with a stochastic-control background can absorb. The fastest path to hands-on fluency is the **FinRL GitHub** (github.com/AI4Finance-Foundation/FinRL) — clone it, run the stock-trading ensemble tutorial, and you will have stood up PPO, A2C, and DDPG on real equity data in an afternoon. And if you want to test yourself against the field, the **ACM ICAIF FinRL Contest** (open-finance-lab.github.io/finrl-contest-2024.github.io) runs annually with public leaderboards and prize money — a top-three finish is the field's most credible public signal of competence.