

DEEP ACTOR–CRITIC METHODS FOR OPTIMAL TRADE EXECUTION UNDER UNCERTAIN MARKET IMPACT**Ayush Ojha**

Department of Computer Science, University of Cambridge

ayushojha010@gmail.com<https://orcid.org/0009-0001-9404-4930>**ABSTRACT**

Market-impact misspecification can change the preferred distribution of trades over a liquidation horizon. We compare deep deterministic policy gradient and twin delayed deep deterministic policy gradient under nominal and parameter-randomized training in a constrained single-asset simulator. Ten training seeds per group are evaluated against time-weighted execution and a fixed quadratic schedule. The study comprises forty training runs, 800,000 transitions and 344,064 held-out episode evaluations. Across seven prespecified nonnominal conditions, the randomized-minus-nominal TD3 objective difference is -0.000658 (-0.17%); its paired training-seed 95% interval is $[-0.01057, 0.00925]$ and includes zero. The initial five-seed estimate changes substantially when five additional fixed seeds are included. Learned policies approach the fixed quadratic schedule and improve substantially over time-weighted execution in the nominal setting. Under the largest joint impact and liquidity shift, randomized TD3 improves on nominal TD3 but remains worse than time-weighted execution. All evaluated policies complete liquidation. These results do not establish a systematic benefit from parameter randomization or superiority to analytical control. The study uses synthetic paths, immediate fills and a chosen price-risk penalty; it makes no claim about attainable market returns.

Keywords

Optimal execution; deep reinforcement learning; actor–critic methods; market impact; parameter uncertainty; financial technology.

INTRODUCTION

Optimal trade execution concerns the completion of a specified order within a specified horizon. Its objective differs from predicting the return of an asset: a trader must choose how to distribute an existing purchase or sale across time while accounting for the effect of those decisions on execution prices. Bertsimas and Lo formulate this task as finite-horizon control under market-dependent execution costs. Almgren and Chriss balance expected execution cost against inventory risk in a discrete trading schedule. Almgren's stochastic-liquidity model makes the competing pressures explicit: slower trading can reduce impact costs, whereas faster trading reduces exposure to adverse price movements. The execution reinforcement-learning study of Nevmyvaka, Feng and Kearns likewise treats remaining inventory and remaining time as essential constraints. [1] [2] [3] [4]

The market-impact specification is part of the decision problem, rather than a neutral detail of its implementation. Empirical and theoretical studies relate impact to limited liquidity, persistent order flow and the response of liquidity providers. They also disagree about the functional form and persistence that should summarize these effects. Tóth and colleagues motivate concave impact through a liquidity model; Zarinelli and colleagues find that a logarithmic specification fits their parent-order sample over a wider size range than a square-root approximation. Bacry and colleagues distinguish the trajectory during execution from subsequent relaxation. These findings motivate sensitivity analysis across stated impact assumptions; they do not supply a universal calibration for an unobserved market. [5] [6] [7] [8] [9]

Reinforcement learning provides a way to optimize repeated execution decisions from sampled transitions. Financial execution applications precede deep continuous-action methods: Nevmyvaka and colleagues learn a constrained execution policy, while Hendricks and Wilcox use a learning agent to modify an analytical execution trajectory in response to spread and volume. Deterministic policy gradients and deep deterministic policy gradient (DDPG) subsequently provide a direct parameterization of continuous actions. Twin delayed deep deterministic policy gradient (TD3) addresses estimation error in this family through paired critics, delayed actor updates and target-policy smoothing. These developments motivate the present comparison, but their continuous-control results do not establish financial execution performance. [4] [10] [11] [12] [13]

This study examines DDPG and TD3 under nominal and parameter-randomized training in a constrained single-asset liquidation model. The question is whether the learned policies remain competitive with matched execution schedules when the impact parameters differ from those used in training. The design draws on pre-existing work on uncertain transition models, neural policies trained with model ensembles, and dynamics randomization. The contribution is a

controlled execution comparison with explicit model variation, rather than a new actor–critic algorithm or a claim to the first reinforcement-learning execution strategy. Randomized training optimizes performance under a specified sampling distribution; it does not by itself establish a worst-case robustness guarantee. [10] [14] [15] [16]

RELATED WORK

Execution objectives and model assumptions

Classical execution problems already allow decisions to depend on market conditions. Bertsimas and Lo derive dynamic cost-minimizing strategies over a finite horizon. Almgren allows liquidity and volatility to change stochastically, with the policy responding to their instantaneous values. Hernández-del-Valle and Pacheco-González instead examine deterministic execution under a geometric price process, and Kato studies how nonlinear impact changes the continuous-time control problem. These studies illustrate why an analytical benchmark must be matched to its price process, objective and admissible controls. A comparison against a schedule derived for different assumptions would confound learning performance with a change in the benchmark problem. [1] [3] [17] [18]

Order-book models place greater emphasis on the recovery of liquidity after a trade. Obizhaeva and Wang distinguish the dynamic replenishment of supply and demand from static depth and spread. Alfonsi, Fruth and Schied extend this perspective to more general book shapes and distinguish recovery of consumed volume from recovery of the spread. Huang, Lehalle and Rosenbaum model order arrivals and cancellations as state-dependent queue dynamics within periods of constant reference price. These approaches contain mechanisms absent from a reduced-form simulator with immediate fills and no explicit queues. A result in the reduced-form model should therefore be interpreted as a result about its specified cost and inventory dynamics. [19] [20] [21]

Execution schedules also depend on what information about future trading volume is available. Kato establishes volume-weighted execution results in a volume-dependent execution framework and separately distinguishes static, anticipatory and adaptive strategies. Cartea and Jaimungal derive a strategy that adjusts an analytical execution component using expected net order flow under linear impact assumptions. Guéant, Lehalle and Fernandez Tapia jointly consider the liquidation schedule and limit-order prices, introducing non-execution risk alongside price risk. These papers distinguish an executable policy using available information from a retrospective volume benchmark or a strategy with different fill assumptions. [22] [23] [24] [25]

Impact estimation, memory and consistency

Impact need not be explained by traded volume alone. Cont, Kukanov and Stoikov study price changes in relation to imbalance in orders at the best quotes. Eisler, Bouchaud and Kockelkoren compare temporary-impact and history-dependent descriptions of multiple order-book event types. Farmer and colleagues discuss how the persistence of transaction signs can coexist with weakly predictable returns through changes in liquidity and impact. Together with Bouchaud, Farmer and Lillo's review, these studies motivate distinguishing exogenous price variation, observed order flow and the agent's own trading impact. They also caution against treating an independent price-noise term as a complete model of market microstructure. [5] [6] [26] [27]

Parent-order studies address a different scale from individual order-book events. Tóth and colleagues derive concave impact from limited local liquidity and provide empirical support for square-root behavior. Zarinelli and colleagues compare this approximation with logarithmic dependence on size and participation, while Bacry and colleagues separate execution-time impact from its later relaxation. The functional form, normalization and observation horizon consequently deserve separate treatment. Changing an impact coefficient within one model probes parameter sensitivity; it does not test all the mechanisms represented by these alternative models. [7] [8] [9]

A flexible optimizer can exploit defects in an impact model. Curato, Gatheral and Lillo show that nonlinear transient-impact specifications can generate alternating trades and, in some settings, negative expected execution costs; spread costs or convex impact at high rates mitigate the behavior they examine. Donier and colleagues construct a latent-liquidity model with an explicit price-manipulation consistency result. Schneider and Lillo extend no-dynamic-arbitrage considerations to cross-asset impact. These works motivate feasibility and cost-consistency checks before interpreting a learned policy as better execution. They also identify a boundary of the present single-asset setting: it cannot establish performance for portfolio liquidation with cross-impact. [28] [29] [30]

Financial learning and model uncertainty

The nearest execution-learning antecedents differ in both representation and evaluation. Nevmyvaka, Feng and Kearns combine a constrained execution state with market variables and exploit a low-impact factorization in training. Hendricks and Wilcox adjust a base execution trajectory using tabular reinforcement learning and evaluate their policy on chronologically separated equity-market samples. Spooner and colleagues study temporal-difference learning for market making, where quoting decisions and inventory accumulation replace the fixed parent-order completion objective. The financial setting alone does not make these tasks interchangeable. The present study retains the parent-order completion problem and compares continuous trade-size actors within the same simulator. [4] [10] [31]

Ojha's budgeted fraud-alert study separates predictive scoring from the allocation of a finite review budget. Its Double DQN uses the current score, remaining queue length and remaining capacity, and is compared with validation-tuned thresholds and finite-horizon dynamic programs. The learned allocator does not consistently improve utility, and its outcomes vary across training seeds. [32] This provides a methodological precedent for representing remaining time and inventory explicitly and for comparing neural execution policies with transparent decision rules. The connection concerns constrained sequential allocation: fraud review decides whether to spend scarce capacity, whereas liquidation distributes a prescribed order across time.

Ojha's portfolio-allocation study compares matched deep policies under return-only, transaction-cost and squared-downside objectives. Training with proportional fees reduces turnover and improves net results relative to return-only training, but equal-weight rebalancing still exceeds the cost-aware policy's median wealth on all three benchmarks. Stronger downside penalties also change cash exposure and trading rather than uniformly improving investment outcomes. [33] These findings support optimizing explicit decision costs and distinguishing the trained risk penalty from broader financial performance. That study optimizes a differentiable portfolio objective and omits market impact; the present study uses actor-critic learning to examine temporary impact and inventory exposure during mandatory liquidation. A fixed quadratic schedule provides a model-based reference, so improvement over TWAP alone cannot establish an advantage over analytical execution.

Robustness has more than one precise meaning. Schied studies an execution rule whose optimality survives a specified class of changes in the unaffected price law. Cartea, Donnelly and Jaimungal model ambiguity in order arrivals, fills and price dynamics through an adverse alternative model with a penalty for deviation. Nilim and El Ghaoui formulate uncertain transition matrices and robust dynamic programming under stated uncertainty-set assumptions. These analyses distinguish sensitivity to a particular misspecification, optimization over a sampling distribution, and a worst-case objective. The present experiments concern specified impact-parameter perturbations and should not be described as solving general model ambiguity. [14] [34] [35]

Pre-existing neural-control work provides a practical precedent for varying simulator dynamics during training. EPOpt samples model ensembles and emphasizes low-return trajectories to seek lower-tail performance; its formulation explicitly distinguishes expected performance across models from a conditional lower-tail objective. Peng and colleagues sample dynamics parameters at the start of an episode and retain them during that episode. Their robotic-control results establish neither market calibration nor transfer to a trading venue, but the training designs provide methodological precedents for comparing a nominal execution learner with one trained over a stated parameter distribution. [15] [16]

Actor-critic methods and evaluation

Temporal-difference learning estimates future outcomes by updating from differences between successive predictions. Q-learning uses sampled transitions to update action values, with its classical convergence result depending on discrete representation and repeated state-action sampling. Lin investigates replaying previous experiences as a means of reusing interactions. Deep Q-networks combine neural action-value estimation with mechanisms for stabilizing learning, demonstrating their approach in discrete-action game tasks. This sequence of work provides the background for replay-based critics; it does not transfer a tabular convergence guarantee to a nonlinear execution agent. [36] [37] [38] [39]

Policy-gradient methods instead parameterize the policy directly. Williams develops stochastic gradient-following reinforcement algorithms, while Sutton and colleagues relate policy gradients to compatible value or advantage approximation. Konda and Tsitsiklis analyze two-time-scale actor-critic algorithms with linear critics. Silver and colleagues derive deterministic policy gradients for continuous actions and an off-policy actor-critic formulation. These results motivate separating the trade-size actor from the cost-to-go critic, while retaining the assumptions of each original analysis rather than asserting convergence for an arbitrary pair of deep networks. [11] [40] [41] [42]

DDPG combines deterministic policy gradients with replay and slowly updated target networks. Estimated action values can nevertheless be biased: van Hasselt's double-estimator construction addresses maximization-induced overestimation, and its deep extension demonstrates the issue with nonlinear action-value approximation. TD3 applies related reasoning to continuous actor-critic learning, using the smaller of paired target estimates, delayed policy updates and smoothed target actions. Comparing DDPG and TD3 therefore tests a specific set of estimation-stability changes. [12] [13] [43] [44]

Other actor-critic designs make different choices about data collection and policy updates. Generalized advantage estimation controls the bias-variance tradeoff through discounted temporal-difference residuals. Trust-region policy optimization constrains policy changes through a theoretically motivated surrogate and practical approximations; proximal policy optimization uses a clipped surrogate with repeated minibatch optimization. Asynchronous actor-critic learning uses multiple actor-learners to produce training data and updates. These approaches provide context for the replay-based deterministic methods evaluated here. Their policy-update constraints should not be confused with explicit limits on liquidation cost or financial inventory risk. [45] [46] [47] [48]

Maximum-entropy methods introduce another distinction. Haarnoja and colleagues first develop continuous-action soft Q-learning with energy-based policies and subsequently formulate a stochastic off-policy soft actor-critic. Entropy is part of those learning objectives, rather than a bound on execution-cost uncertainty. These methods are relevant alternatives within the continuous-control literature, but the present deterministic-actor comparison does not evaluate them and makes no claim of superiority over stochastic or entropy-regularized actors. [49] [50]

Finally, methodological differences require a common evaluation setting. Duan and colleagues introduce a continuous-control benchmark and reference implementations to make algorithmic progress easier to compare. For execution, the corresponding requirement is to hold the liquidation objective, action constraints, information set, training budget and evaluation paths fixed wherever the comparison permits. Performance variability across independent training seeds and across impact conditions must remain visible. A simulator comparison supports conclusions within the documented experiment; it does not establish attainable performance in a live market. [1] [51]

EXECUTION MODEL AND RESEARCH QUESTION

Research question and scope

The research question is whether varying impact parameters during training improves the behavior of continuous trade-size policies when those parameters change at evaluation. The target is liquidation of an existing position. The experiment does not choose an asset, forecast its return or initiate a speculative position. We study a deliberately reduced execution model to make the learning comparison reproducible and its assumptions inspectable. The model builds on constrained execution and nonlinear-impact literature, and the learning comparison uses DDPG and TD3. [10] [12] [13] [28]

Inventory and temporary impact

Let x_{k-1} denote inventory immediately before decision k , and let q_k be the quantity sold. The horizon and initial position are normalized to $T = X = 1$, with $N = 20$ equally spaced decisions and $\Delta t = T/N$. Inventory obeys $x_k = x_{k-1} - q_k$, subject to $0 \leq q_k \leq x_{k-1}$ and $x_N = 0$. Every admissible trade fills immediately. There is no permanent impact, spread, fee, queue priority or risk of non-execution. These exclusions define the experiment and limit the conclusions that can be drawn from it.

The temporary price concession is $h(v_k, L_k) = \eta v_k^\alpha / L_k$, where $v_k = q_k / \Delta t$ and $L_k > 0$ is observed liquidity. The impact component of liquidation cost is

$$C_{\text{impact}} = \sum_{k=1}^N \eta q_k \left(\frac{q_k}{\Delta t} \right)^\alpha \frac{1}{L_k}.$$

The coefficient η and exponent α are constant within an episode and hidden from the actor. Varying the exponent tests a specified functional-form perturbation. It does not reproduce the order-book mechanisms, persistent impact or price-manipulation results in richer execution models. Restricting the action to sales prevents round trips in this particular environment; it is not a general no-arbitrage result. [28] [29]

Liquidity is exogenous. We use $\log L_k = \log \ell + u_k$, with $u_{k+1} = \rho u_k + s_\epsilon \epsilon_k$ and independent standard Gaussian innovations. The initial u_0 has the stationary distribution with variance $s_\epsilon^2 / (1 - \rho^2)$. We set $\rho = 0.8$ and $s_\epsilon = 0.1$. The actor observes the current L_k but does not receive its episode-specific mean parameter ℓ . This process is a controlled simulation assumption rather than a fitted liquidity series.

Risk-adjusted objective and realized shortfall

The learning objective is the expectation of an explicit cost proxy, with holding-risk coefficient $\lambda = 25$ and price-noise scale $\sigma = 0.2$:

$$J = \mathbb{E} \left[C_{\text{impact}} + \lambda \sigma^2 \Delta t \sum_{k=1}^N x_k^2 \right].$$

The holding-risk term uses inventory after each trade. Training reward is minus ten times the corresponding one-step cost. The constant reward multiplier improves numerical scale without changing the ordering of complete-episode costs. Independent price innovations are omitted from the training reward because their conditional expectation is zero and the actor cannot observe them. Evaluation separately records their contribution.

For independent standard Gaussian z_k , the evaluation price-slippage term is $C_{\text{price}} = -\sigma \sqrt{\Delta t} \sum_k x_k z_k$. Implementation shortfall is $C_{\text{impact}} + C_{\text{price}}$. Conditional on a liquidity-driven schedule, the price component has variance $\sigma^2 \Delta t \sum_k x_k^2$. Stochastic liquidity also changes the impact cost, so this expression is not the unconditional variance of total implementation shortfall. The reported objective is therefore a chosen price-risk penalty plus impact, not a complete mean-variance model of all execution uncertainty.

Actor state and admissible trade sizes

At n remaining decisions, the actor receives $s = (n/N, x/X, \log L)$. A tanh output $a \in [-1, 1]$ maps to the trade $q = x \min\{1, 3(a + 1)/(2n)\}$. At the final decision, the environment sells the remaining inventory irrespective of the actor

output. Thus both the actor and the benchmarks obey the same completion requirement; observed completion alone is not evidence that the network learned it.

The actor receives no future innovation, impact coefficient, impact exponent, evaluation-scenario identifier or benchmark action. Since the hidden parameters persist across an episode, the observation is not a complete Markov state. We evaluate memoryless feedback policies on this partially observed problem. Neither the algorithm nor the experiment provides a guarantee of an optimal policy for the partially observed process.

Benchmarks and deterministic check

Time-weighted average price execution (TWAP) sells $q = x/n$ at every remaining decision. The second benchmark uses a fixed quadratic schedule from the discrete execution framework of Almgren and Chriss. [2] We specialize it to zero permanent impact, constant liquidity and risk assessed after each trade. For a linear price concession and the same after-trade risk discretization, write $b = \eta_0/\Delta t$, $d = \lambda\sigma^2\Delta t$ and $V_n(x) = A_n x^2$. Its backward recurrence is

$$A_1 = b, \quad c_n = d + A_{n-1}, \quad q_n = \frac{c_n}{b + c_n}x, \quad A_n = \frac{bc_n}{b + c_n} \quad (n > 1).$$

This follows by minimizing $bq^2 + c_n(x - q)^2$. We use nominal $\eta_0 = 0.1$ to construct the fractions and keep those fractions fixed in every scenario. The recurrence is an analytical check and a transparent schedule benchmark; with stochastic liquidity it is not the optimal adaptive control. In the constant-liquidity nominal check, its objective is 0.29336074, compared with 0.40875000 for TWAP. The value recurrence, direct quadratic solve and simulated deterministic cost agree. Our numerical check also verifies that every nominal fraction satisfies the actor's common urgency limit.

LEARNING AND EXPERIMENTAL DESIGN

Continuous actor–critic updates

DDPG uses a deterministic actor μ_θ , a critic Q_ϕ , a replay buffer and slowly updated target networks. Its critic fits targets $y = r + \gamma(1 - d_{\text{term}})Q_\phi(s', \mu_{\bar{\theta}}(s'))$, where d_{term} indicates terminal completion. The actor ascends the replay-batch mean of $Q_\phi(s, \mu_\theta(s))$. The state and action parameterization are adaptations for execution; this experiment is not a replication of the original physical-control benchmark. [11] [12]

TD3 replaces the target value with the smaller of two target-critic estimates. Its target action adds Gaussian noise with standard deviation 0.2, clips that noise to $[-0.5, 0.5]$, then clips the resulting action to $[-1, 1]$. Actor and target-network updates occur once per two critic updates. Target smoothing acts on the actor output, before the constrained trade-size mapping. Terminal targets exclude bootstrapping. The comparison evaluates the combined TD3 design and does not identify the separate causal effect of each component. [13]

Actors and critics each use two hidden layers of 64 rectified-linear units. The actor's output uses tanh; each critic takes the observation and action and returns a scalar. Adam uses learning rates 0.001 for both actor and critic, with its implementation's default moment settings. Target parameters use interpolation coefficient 0.005. Batches contain 64 transitions sampled uniformly from a replay capacity of 20,000. The discount factor is 1.0 to match the undiscounted finite-horizon objective. [52]

The first 1,000 transitions of each run use uniform random actions. Thereafter, the behavior action is perturbed by Gaussian exploration noise with standard deviation 0.15 and clipped to its admissible output interval. Each run contains 20,000 transitions, or 1,000 complete episodes. One critic update is made per post-warm-up transition: 19,000 updates per run. DDPG makes 19,000 actor updates; TD3 makes 9,500. The final policy is evaluated without exploration.

Nominal and randomized training

Nominal training fixes $\eta = 0.1$, $\alpha = 1$ and $\ell = 1$. Randomized training independently samples η log-uniformly on $[0.05, 0.20]$, α uniformly on $[0.8, 1.2]$ and ℓ uniformly on $[0.75, 1.25]$ at the start of each episode. Parameters then remain fixed throughout the episode. These distributions are prescribed design choices. They are not inferred from an exchange, and the expected-cost objective does not use EPOpt's lower-tail selection or a worst-case uncertainty set. [14] [15] [16]

The algorithms, regimes, code, configuration and evaluation conditions were frozen before the initial experiment. Seeds 11, 23, 37, 53 and 71 defined five runs for each algorithm–regime group. After inspecting that experiment's aggregate result, we fixed five additional seeds—89, 107, 127, 149 and 173—for all four groups before running the extension. No architecture, hyperparameter, scenario or held-out path was changed, and every added run is included. The combined study therefore contains ten seeds per group, forty trained policies and 800,000 transitions. A checkpoint is saved every 5,000 transitions. Only the final 20,000-transition checkpoint is evaluated; intermediate checkpoints were not used to select the reported policy. No early stopping or evaluation-driven hyperparameter tuning was performed.

Prespecified evaluation conditions

Table 1 summarizes the eight evaluation conditions. Each condition uses 1,024 held-out episodes. The same saved liquidity and price innovations are shared across methods and conditions; the episode mean-liquidity parameter changes

the scale of liquidity without changing the underlying innovation array. The price innovations are independent of the liquidity process. All forty trained policies and both schedules are evaluated in each of eight conditions, producing 344,064 episode records. Training and held-out paths use separate random-number streams.

Table 1. Prespecified impact and liquidity conditions. High and low impact use the upper and lower training boundaries, respectively. The concave and convex exponents and the thin- and high-liquidity means lie outside their respective training ranges. The joint shift lies outside all three ranges.

Condition	Impact coefficient η	Exponent α	Mean-liquidity parameter ℓ
Nominal	0.10	1.00	1.00
High impact	0.20	1.00	1.00
Low impact	0.05	1.00	1.00
Concave impact	0.10	0.75	1.00
Convex impact	0.10	1.25	1.00
Thin liquidity	0.10	1.00	0.50
Joint shift	0.25	1.30	0.60
High liquidity	0.10	1.00	1.50

Estimation and inferential scope

The primary contrast is randomized TD3 minus nominal TD3, averaging the objective equally over the seven nonnominal conditions. For each paired seed, let D_i be that average difference. We report $\bar{D}$ and the paired training-seed interval $\bar{D} \pm 2.2621571629 s_D / \sqrt{10}$, using nine degrees of freedom. A negative difference favors randomized training. Relative change divides $\bar{D}$ by the average nominal-TD3 objective over the same conditions and seeds.

The interval describes variation across ten trained-seed pairs, conditional on the saved held-out path bank and the fixed scenario set. Because the five-seed extension followed inspection of the initial result, the pooled interval is reported as an exploratory estimate rather than a new confirmatory test. It does not include uncertainty about the scenario distribution, market-model calibration or future markets, and it does not treat the many evaluation episodes as independent training replications. Finite held-out Monte Carlo error remains outside this interval. DDPG regime differences, cross-algorithm contrasts and scenario-specific comparisons are exploratory secondary analyses; no multiple-comparison significance claim is made.

Results

Complete scenario results

Table 2 reports the recorded risk-adjusted objective. Learned-policy values are means over the ten trained seeds, each evaluated on the same held-out paths. Benchmark values use that path bank without a training-seed dimension. Values are dimensionless costs under the stated normalization; they are not basis points, currency amounts or estimated profits. The table includes every prespecified condition. Figure 1 compares the mean costs and training-seed uncertainty across these conditions.

Table 2. Mean risk-adjusted objective for all eight evaluation conditions and six methods.

Condition	TWAP	Fixed quadratic schedule	DDPG nominal	DDPG randomized	TD3 nominal	TD3 randomized
Nominal	0.41093	0.29692	0.29860	0.29985	0.29847	0.29820
High impact	0.51311	0.46229	0.46501	0.46848	0.46707	0.46340
Low impact	0.35984	0.21424	0.21540	0.21554	0.21417	0.21560
Concave impact	0.41093	0.27359	0.27547	0.27605	0.27474	0.27547
Convex impact	0.41093	0.32704	0.32795	0.32997	0.32856	0.32710
Thin liquidity	0.51311	0.46229	0.53448	0.46220	0.45140	0.46192
Joint shift	0.73451	0.97515	0.84765	0.85548	0.81023	0.79792
High liquidity	0.37687	0.24180	0.24044	0.24058	0.24009	0.24026

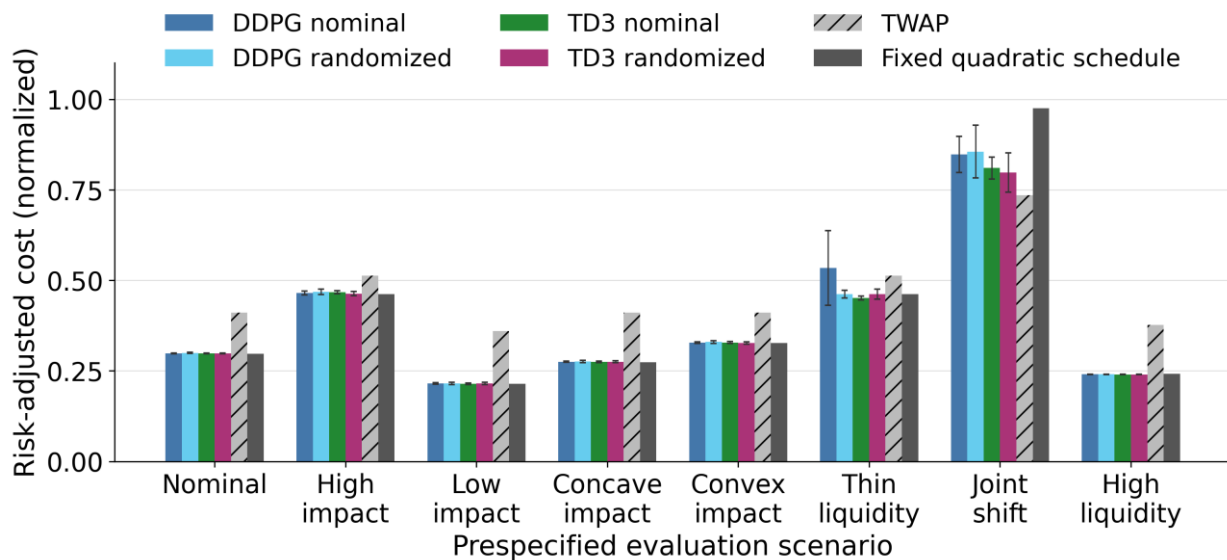


Figure 1. Risk-adjusted objective by condition. Bar heights represent means across ten trained seeds; learned-policy error bars show 95% training-seed intervals. TWAP and the fixed quadratic schedule have no trained-seed interval. These conditional intervals should not be read as calibrated market uncertainty.

Primary and secondary contrasts

For the primary comparison, nominal TD3's average nonnominal objective is 0.39803730. The randomized-minus-nominal difference is -0.00065765 , or -0.1652% using the nominal objective as denominator. Its paired 95% interval is $[-0.01056951, 0.00925422]$. Six seed pairs favor randomized training and four favor nominal training. The interval includes zero, and the experiment does not establish a systematic advantage for randomization. Table 3 reports the paired seed differences underlying the pooled estimate. The original five-seed difference was -0.00844675 (-2.1082%), with interval $[-0.01743075, 0.00053724]$; adding the fixed second seed batch moves the pooled point estimate close to zero.

The secondary DDPG regime difference is -0.00829802 , or -1.9986% , with interval $[-0.02526683, 0.00867078]$. The cross-algorithm TD3-minus-DDPG differences are -0.01716131 under nominal training and -0.00952094 under randomized training; their intervals are $[-0.03755945, 0.00323683]$ and $[-0.02384644, 0.00480457]$, respectively. Both intervals include zero. The aggregate results therefore provide no resolved ranking between the algorithms.

Table 3. Paired seed differences for the primary TD3 contrast, averaged over the seven nonnominal conditions. The inferential unit is a trained-seed pair.

Seed	Randomized minus nominal objective
11	-0.01800787
23	0.00198335
37	-0.01104792
53	-0.00826571
71	-0.00689562
89	0.00271699
107	0.02636373
127	0.01824734
149	-0.01142726
173	-0.00024348

Nominal performance and joint stress

In the nominal condition, TWAP has objective 0.41093247 and the fixed quadratic schedule has objective 0.29692164. The four learned means range from 0.29819957 to 0.29985467. Randomized TD3 is 0.4304% more costly than the fixed quadratic schedule and approximately 27.43% less costly than TWAP. This result shows that the small actors approximate the fixed quadratic schedule within the simulator. It does not establish superiority over an optimal liquidity-adaptive controller, which was not evaluated.

The joint shift is the strongest challenge to the training specification. Randomized TD3 has objective 0.79791881, compared with 0.81023465 for nominal TD3 and 0.97515321 for the fixed quadratic schedule. However, TWAP's objective is 0.73451030, lower than either TD3 policy. Increasing impact and reducing liquidity makes aggressive early trading more expensive; spreading trades more evenly can outweigh the saved holding-risk penalty. In this condition,

randomization improves the learned policy's cost relative to its nominal counterpart without producing the best evaluated execution rule.

Figure 2 compares the corresponding inventory profiles under nominal and joint-shift conditions. The regime effect is not uniform across conditions. Randomized TD3 is less costly than nominal TD3 under high impact, convex impact and the joint shift, but more costly under low impact, concave impact, thin liquidity and high liquidity. For DDPG, the average favorable regime difference is driven by thin liquidity; the randomized mean increases in each of the other six nonnominal conditions. This heterogeneity explains why an overall average alone is an incomplete description of specification sensitivity.

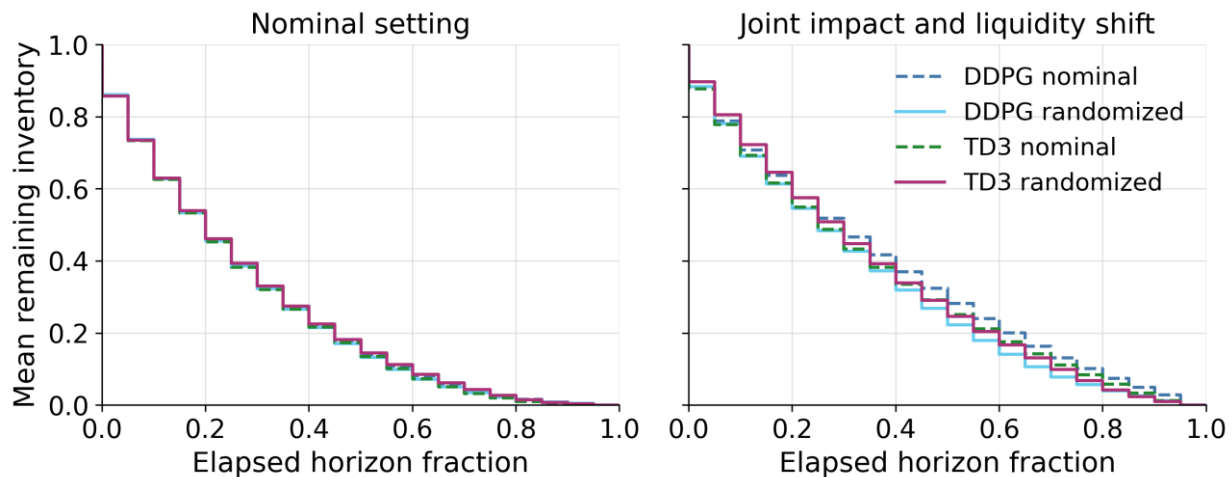


Figure 2. Mean remaining-inventory profiles for the four learned groups in the nominal and joint-shift conditions, averaged across trained seeds and the common held-out liquidity paths. Inventory is plotted immediately after each decision; the initial trade occurs at the start of the horizon. The curves describe the policies' feedback response under the same observed liquidity innovations. They do not reveal the hidden impact parameters to the actor.

Learning dynamics

Figure 3 shows the recorded training objectives over successive blocks of fifty episodes. Nominal and randomized groups optimize different training distributions, so the curves are not a held-out comparison of execution performance. Training logs retain high-cost episodes. Each algorithm uses the same transition budget, but TD3 makes fewer actor updates. The curves describe the observed learning path without establishing convergence or supporting retrospective checkpoint selection.

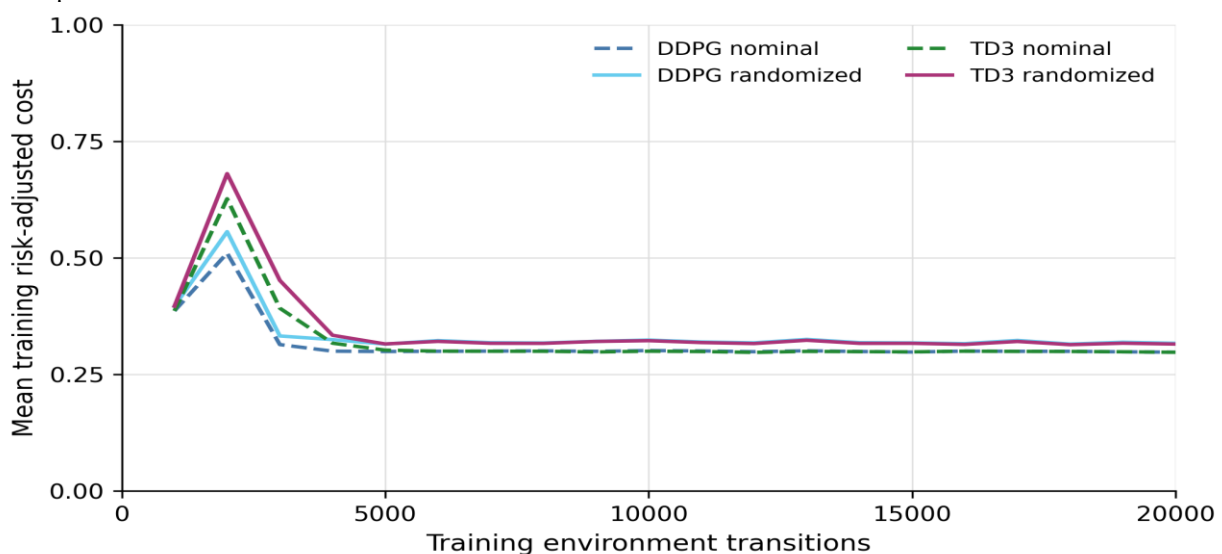


Figure 3. Mean recorded training objective for successive blocks of fifty episodes in the four groups. Nominal and randomized curves average their different training distributions, so their costs are not directly comparable held-out validation outcomes. These curves were not a policy-selection criterion.

Completion and numerical integrity

All 344,064 evaluated episodes end with zero inventory. Completion follows from the enforced terminal trade, while saved paths additionally verify nonnegative inventory, inventory conservation and finite nonnegative impact costs. The extension checks confirm that original run files, common evaluation paths and original-policy per-seed results remain unchanged. The primary contrast is recomputed from raw records, and an earlier independent audit reconstructed all 1,024 joint-shift trajectories for the final randomized-TD3 seed-11 checkpoint using the saved common paths. Trade schedules and each reported cost component matched exactly.

The [versioned reproduction package](#) provides the raw trajectories, per-seed estimates, update counts and final checkpoints used for these checks.

Discussion

The principal finding is no resolved average benefit from parameter-randomized TD3 training. Increasing the comparison from five to ten seeds reduces the estimated relative cost benefit from 2.11% to 0.17%, while seed-level uncertainty remains unresolved. This sensitivity cautions against interpreting a small first-batch estimate as a stable gain. Randomized training optimizes across a specified distribution rather than guaranteeing performance for every alternative model. The joint shift shows that a policy can improve relative to its nominally trained counterpart and still lose to a simpler schedule. The impact coefficient, exponent and liquidity mean enter the objective jointly, changing the relative attraction of early and delayed execution. [14] [15] [16]

The nominal result provides a useful scale for interpreting the learned-policy comparisons. A substantial improvement over TWAP is available through the fixed quadratic schedule before deep learning is introduced. The learned means lie close to that schedule, making the gain over TWAP insufficient evidence of a new solution to optimal execution. Adaptive analytical or numerical control matched to the stochastic-liquidity model would be a more demanding comparator. It should also use the actor's information set, or any extra model information should be reported explicitly. [1] [3]

Several limitations concern scientific coverage rather than numerical reproducibility. The simulator has immediate fills and exogenous liquidity. It omits order arrivals, replenishment caused by trading, adverse selection, transient impact and cross-asset interactions. The policy has no memory with which to infer persistent hidden impact parameters. The risk term captures the chosen price-exposure component rather than all sources of shortfall variation. These omissions make the model useful for a narrow controlled comparison but insufficient for a deployment recommendation. Order-book and ambiguity-aware studies describe mechanisms that would require a different experiment. [19] [21] [30] [35] The fixed interaction budget and final-checkpoint rule also limit interpretation. A different network, training horizon, exploration rule or distribution might change the outcome. Architectures and hyperparameters were held fixed when adding the second seed batch, but the extension followed inspection of the initial result. The pooled estimates therefore remain exploratory. The study cannot claim a universal algorithm ranking, and a favorable isolated scenario should not be treated as a confirmed discovery across many exploratory comparisons. Saved training and evaluation paths allow a future study to separate changes in learning stability from changes in the execution model. [51]

A subsequent study could add a history-dependent observation representation, compare against a matched numerical stochastic controller, or incorporate replenishment and execution uncertainty. Such extensions should be tested under a new declared protocol, retaining separate training and evaluation evidence. Real-market validation would additionally require appropriate market data, an observation window, a model of the agent's effect on fills and a credible calibration procedure. None of those requirements is satisfied merely by evaluating synthetic paths. [4] [20] [25]

CONCLUSION

This controlled study compares two continuous actor-critic methods under fixed and randomized impact assumptions. Learned policies approximate the fixed quadratic schedule and perform better than TWAP in the nominal setting. Across ten seeds, the average TD3 stress-condition costs are nearly equal between training regimes, the primary interval includes zero, and the largest joint shift favors TWAP. The evidence supports reporting scenario sensitivity and training variability alongside average cost. It does not support a systematic randomization benefit, a general robustness guarantee, an optimal-policy claim or a claim of live financial performance.

Reproducibility and Implementation Details

Experiments and statistical analysis use an isolated Windows CPU environment with Python 3.6.6, NumPy 1.15.1 and PyTorch 0.4.1. The [versioned release](#) supplies environment setup instructions, the frozen configurations, 160 saved checkpoints and the analysis entry points.

Data and Code Availability

All data in this study are generated by the documented simulator; no proprietary transaction records or historical stock-price observations are used. The reproduction package includes the simulator, learning code, configurations, training logs, checkpoints, common evaluation paths, raw held-out records and analysis outputs. Public repository: [reproduction repository](#). The versioned code and synthetic-data archive is available in [release v1.0.0](#).

Funding

This research received no external funding.

Competing Interests

The author declares no competing interests.

REFERENCES

- 1) Dimitris Bertsimas and Andrew W. Lo, "Optimal Control of Execution Costs," *Journal of Financial Markets*, vol. 1, no. 1, pp. 1–50, 1998. [https://doi.org/10.1016/S1386-4181\(97\)00012-8](https://doi.org/10.1016/S1386-4181(97)00012-8)
- 2) Robert Almgren and Neil Chriss, "Optimal Execution of Portfolio Transactions," *The Journal of Risk*, vol. 3, no. 2, pp. 5–39, 2001. <https://doi.org/10.21314/jor.2001.041>
- 3) Robert Almgren, "Optimal Trading with Stochastic Liquidity and Volatility," *SIAM Journal on Financial Mathematics*, vol. 3, no. 1, pp. 163–181, 2012. <https://doi.org/10.1137/090763470>
- 4) Yuriy Nevmyvaka, Yi Feng, and Michael Kearns, "Reinforcement Learning for Optimized Trade Execution," in *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*, pp. 673–680, 2006. <https://doi.org/10.1145/1143844.1143929>
- 5) Jean-Philippe Bouchaud, J. Dooyne Farmer, and Fabrizio Lillo, "How Markets Slowly Digest Changes in Supply and Demand," in *Handbook of Financial Markets: Dynamics and Evolution*, pp. 57–160, 2009. <https://doi.org/10.1016/B978-012374258-2.50006-3>
- 6) J. Dooyne Farmer, Austin Gerig, Fabrizio Lillo, and Szabolcs Mike, "Market Efficiency and the Long-Memory of Supply and Demand: Is Price Impact Variable and Permanent or Fixed and Temporary?" *Quantitative Finance*, vol. 6, no. 2, pp. 107–112, 2006. <https://doi.org/10.1080/14697680600668048>
- 7) Bence Tóth, Yves Lempérière, Cyril Deremble, Joachim de Lataillade, Julien Kockelkoren, and Jean-Philippe Bouchaud, "Anomalous Price Impact and the Critical Nature of Liquidity in Financial Markets," *Physical Review X*, vol. 1, no. 2, article 021006, 2011. <https://doi.org/10.1103/PhysRevX.1.021006>
- 8) Elia Zarinelli, Michele Treccani, J. Dooyne Farmer, and Fabrizio Lillo, "Beyond the Square Root: Evidence for Logarithmic Dependence of Market Impact on Size and Participation Rate," *Market Microstructure and Liquidity*, vol. 1, no. 2, article 1550004, 2015. <https://doi.org/10.1142/S2382626615500045>
- 9) Emmanuel Bacry, Adrian Iuga, Matthieu Lasnier, and Charles-Albert Lehalle, "Market Impacts and the Life Cycle of Investors Orders," *Market Microstructure and Liquidity*, vol. 1, no. 2, article 1550009, 2015. <https://doi.org/10.1142/S2382626615500094>
- 10) Dieter Hendricks and Diane Wilcox, "A Reinforcement Learning Extension to the Almgren-Chriss Framework for Optimal Trade Execution," in *2014 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFER)*, pp. 457–464, 2014. <https://doi.org/10.1109/CIFER.2014.6924109>
- 11) David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller, "Deterministic Policy Gradient Algorithms," in *Proceedings of the 31st International Conference on Machine Learning*, vol. 32, pp. 387–395, 2014. <https://proceedings.mlr.press/v32/silver14.html>
- 12) Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra, "Continuous Control with Deep Reinforcement Learning," in *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, 2016. <https://arxiv.org/abs/1509.02971v5>
- 13) Scott Fujimoto, Herke van Hoof, and David Meger, "Addressing Function Approximation Error in Actor-Critic Methods," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, pp. 1587–1596, 2018. <https://proceedings.mlr.press/v80/fujimoto18a.html>
- 14) Arnab Nilim and Laurent El Ghaoui, "Robust Control of Markov Decision Processes with Uncertain Transition Matrices," *Operations Research*, vol. 53, no. 5, pp. 780–798, 2005. <https://doi.org/10.1287/opre.1050.0216>
- 15) Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine, "EPOpt: Learning Robust Neural Network Policies Using Model Ensembles," in *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017. <https://openreview.net/forum?id=SyWvgP5el>
- 16) Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel, "Sim-to-Real Transfer of Robotic Control with Dynamics Randomization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3803–3810, 2018. <https://doi.org/10.1109/ICRA.2018.8460528>
- 17) Gerardo Hernández-del-Valle and Carlos Pacheco-González, "Optimal Execution of Portfolio Transactions with Geometric Price Process," arXiv:0908.1211, 2009. <https://arxiv.org/abs/0908.1211v2>

- 18) Takashi Kato, "An Optimal Execution Problem with Market Impact," *Finance and Stochastics*, vol. 18, no. 3, pp. 695–732, 2014. <https://doi.org/10.1007/s00780-014-0232-0>
- 19) Anna A. Obizhaeva and Jiang Wang, "Optimal Trading Strategy and Supply/Demand Dynamics," *Journal of Financial Markets*, vol. 16, no. 1, pp. 1–32, 2013. <https://doi.org/10.1016/j.finmar.2012.09.001>
- 20) Aurélien Alfonsi, Antje Fruth, and Alexander Schied, "Optimal Execution Strategies in Limit Order Books with General Shape Functions," *Quantitative Finance*, vol. 10, no. 2, pp. 143–157, 2010. <https://doi.org/10.1080/14697680802595700>
- 21) Weibing Huang, Charles-Albert Lehalle, and Mathieu Rosenbaum, "Simulating and Analyzing Order Book Data: The Queue-Reactive Model," *Journal of the American Statistical Association*, vol. 110, no. 509, pp. 107–122, 2015. <https://doi.org/10.1080/01621459.2014.982278>
- 22) Takashi Kato, "VWAP Execution as an Optimal Strategy," *JSIAM Letters*, vol. 7, pp. 33–36, 2015. <https://doi.org/10.14495/jsiaml.7.33>
- 23) Takashi Kato, "An Optimal Execution Problem in the Volume-Dependent Almgren–Chriss Model," *Algorithmic Finance*, vol. 7, nos. 1–2, pp. 1–14, 2018. <https://doi.org/10.3233/AF-170198>
- 24) Álvaro Cartea and Sebastian Jaimungal, "Incorporating Order-Flow into Optimal Execution," *Mathematics and Financial Economics*, vol. 10, no. 3, pp. 339–364, 2016. <https://doi.org/10.1007/s11579-016-0162-z>
- 25) Olivier Guéant, Charles-Albert Lehalle, and Joaquin Fernandez-Tapia, "Optimal Portfolio Liquidation with Limit Orders," *SIAM Journal on Financial Mathematics*, vol. 3, no. 1, pp. 740–764, 2012. <https://doi.org/10.1137/110850475>
- 26) Rama Cont, Arseniy Kukanov, and Sasha Stoikov, "The Price Impact of Order Book Events," *Journal of Financial Econometrics*, vol. 12, no. 1, pp. 47–88, 2014. <https://doi.org/10.1093/jffinec/nbt003>
- 27) Zoltán Eisler, Jean-Philippe Bouchaud, and Julien Kockelkoren, "Models for the Impact of All Order Book Events," in *Market Microstructure*, pp. 113–135, 2012. <https://doi.org/10.1002/9781118673553.ch5>
- 28) Gianbiagio Curato, Jim Gatheral, and Fabrizio Lillo, "Optimal Execution with Non-Linear Transient Market Impact," *Quantitative Finance*, vol. 17, no. 1, pp. 41–54, 2017. <https://doi.org/10.1080/14697688.2016.1181274>
- 29) J. Donier, J. Bonart, I. Mastromatteo, and J.-P. Bouchaud, "A Fully Consistent, Minimal Model for Non-Linear Market Impact," *Quantitative Finance*, vol. 15, no. 7, pp. 1109–1121, 2015. <https://doi.org/10.1080/14697688.2015.1040056>
- 30) M. Schneider and F. Lillo, "Cross-Impact and No-Dynamic-Arbitrage," *Quantitative Finance*, advance online publication, 2018. <https://doi.org/10.1080/14697688.2018.1467033>
- 31) Thomas Spooner, John Fearnley, Rahul Savani, and Andreas Koukorinis, "Market Making via Reinforcement Learning," in *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 434–442, 2018. <https://www.ifaamas.org/Proceedings/aamas2018/pdfs/p434.pdf>
- 32) Ayush Ojha, "Learned Review Policies Versus Score Thresholds for Budgeted Fraud Alert Allocation," *International Journal of Engineering Technology Research & Management*, vol. 2, no. 2, pp. 98–109, February 2018. <https://doi.org/10.5281/zenodo.23274871>
- 33) Ayush Ojha, "Transaction Costs and Downside Risk in Deep Reinforcement Learning for Portfolio Allocation," *International Journal of Engineering Technology Research & Management*, vol. 1, no. 9, pp. 19–31, September 2017. <https://doi.org/10.5281/zenodo.23212855>
- 34) Alexander Schied, "Robust Strategies for Optimal Order Execution in the Almgren–Chriss Framework," *Applied Mathematical Finance*, vol. 20, no. 3, pp. 264–286, 2013. <https://doi.org/10.1080/1350486X.2012.683963>
- 35) Álvaro Cartea, Ryan Donnelly, and Sebastian Jaimungal, "Algorithmic Trading with Model Uncertainty," *SIAM Journal on Financial Mathematics*, vol. 8, no. 1, pp. 635–671, 2017. <https://doi.org/10.1137/16M106282X>
- 36) Richard S. Sutton, "Learning to Predict by the Methods of Temporal Differences," *Machine Learning*, vol. 3, no. 1, pp. 9–44, 1988. <https://doi.org/10.1007/BF00115009>
- 37) Christopher J. C. H. Watkins and Peter Dayan, "Q-Learning," *Machine Learning*, vol. 8, nos. 3–4, pp. 279–292, 1992. <https://doi.org/10.1007/BF00992698>
- 38) Long-Ji Lin, "Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching," *Machine Learning*, vol. 8, no. 3, pp. 293–321, 1992. <https://doi.org/10.1007/BF00992699>
- 39) Volodymyr Mnih et al., "Human-Level Control Through Deep Reinforcement Learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015. <https://doi.org/10.1038/nature14236>

- 40) Ronald J. Williams, "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning," *Machine Learning*, vol. 8, no. 3, pp. 229–256, 1992. <https://doi.org/10.1007/BF00992696>
- 41) Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour, "Policy Gradient Methods for Reinforcement Learning with Function Approximation," in *Advances in Neural Information Processing Systems*, vol. 12, 1999. <https://proceedings.neurips.cc/paper/1999/hash/464d828b85b0bed98e80ade0a5c43b0f-Abstract.html>
- 42) Vijay R. Konda and John N. Tsitsiklis, "Actor-Critic Algorithms," in *Advances in Neural Information Processing Systems*, vol. 12, 1999. <https://papers.neurips.cc/paper/1786-actor-critic-algorithms>
- 43) Hado van Hasselt, "Double Q-Learning," in *Advances in Neural Information Processing Systems*, vol. 23, 2010. <https://proceedings.neurips.cc/paper/2010/hash/091d584fcd301b442654dd8c23b3fc9-Abstract.html>
- 44) Hado van Hasselt, Arthur Guez, and David Silver, "Deep Reinforcement Learning with Double Q-Learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, pp. 2094–2100, 2016. <https://doi.org/10.1609/aaai.v30i1.10295>
- 45) John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel, "High-Dimensional Continuous Control Using Generalized Advantage Estimation," in *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, 2016. <https://arxiv.org/abs/1506.02438v5>
- 46) John Schulman, Sergey Levine, Philipp Moritz, Michael Jordan, and Pieter Abbeel, "Trust Region Policy Optimization," in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 1889–1897, 2015. <https://proceedings.mlr.press/v37/schulman15.html>
- 47) John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov, "Proximal Policy Optimization Algorithms," arXiv:1707.06347, 2017. <https://arxiv.org/abs/1707.06347v2>
- 48) Volodymyr Mnih et al., "Asynchronous Methods for Deep Reinforcement Learning," in *Proceedings of the 33rd International Conference on Machine Learning*, vol. 48, pp. 1928–1937, 2016. <https://proceedings.mlr.press/v48/mniha16.html>
- 49) Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine, "Reinforcement Learning with Deep Energy-Based Policies," in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 1352–1361, 2017. <https://proceedings.mlr.press/v70/haarnoja17a.html>
- 50) Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, pp. 1861–1870, 2018. <https://proceedings.mlr.press/v80/haarnoja18b.html>
- 51) Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel, "Benchmarking Deep Reinforcement Learning for Continuous Control," in *Proceedings of the 33rd International Conference on Machine Learning*, vol. 48, pp. 1329–1338, 2016. <https://proceedings.mlr.press/v48/duan16.html>
- 52) Diederik P. Kingma and Jimmy Lei Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015. <https://arxiv.org/abs/1412.6980v9>