

TRANSACTION COSTS AND DOWNSIDE RISK IN DEEP REINFORCEMENT LEARNING FOR PORTFOLIO ALLOCATION

Ayush Ojha

Cambridge University

ayushojha010@gmail.com

ABSTRACT

Transaction costs can erase the apparent benefit of an active allocation policy. We compare matched deep policies trained for return, return after costs, and return after costs with a squared downside penalty on three historical stock benchmarks. Five seeds and chronological validation separate checkpoint selection from testing. At the primary fee, cost-aware training improves fourteen paired net-return comparisons, with one tie, but equal-weight rebalancing exceeds its median wealth on every benchmark. Downside penalties reduce measured risk with mixed return effects; strong penalties increase cash exposure and trading. The findings support explicit cost accounting, without establishing general investment superiority.

Keywords:

Deep reinforcement learning; Portfolio allocation; Transaction costs; Downside risk; Historical backtesting.

I. INTRODUCTION

Portfolio allocation couples investment returns with decisions about risk and rebalancing. Classical portfolio theory evaluates risk jointly through asset variances and covariances [1]. Proportional costs change the value of reallocating wealth and can produce regions in which an investor does not trade [2] [3]. Gross return alone therefore provides an incomplete assessment of an allocation policy.

Neural networks can map observed market histories to allocations. Backpropagation adjusts hidden representations through an output objective [4], and deep learning composes successive learned layers [5]. Reinforcement learning evaluates actions through their consequences over time [6]. For a portfolio policy, that reinforcement must have an explicit financial meaning: better price predictions do not by themselves demonstrate better net wealth or lower downside exposure.

Cost-aware deep allocation is established prior work. Jiang, Xu and Liang incorporate commissions into a deep reinforcement-learning framework for cryptocurrency portfolios [7]. Introducing commissions alone is consequently not a novel contribution. Their evaluation also illustrates a distinction central to this study: reporting drawdown after training does not mean drawdown was optimized, and optimizing a downside penalty does not establish comprehensive risk control.

Risk-sensitive policy optimization includes mean-variance tradeoffs, variance constraints and reward-to-variability criteria [8]. Other formulations constrain conditional value-at-risk, or CVaR, of cumulative cost [9]. These objectives concern different distributions and horizons. A downside-sensitive allocator must therefore define its training penalty separately from its evaluation statistics; improvement in one statistic cannot establish a general guarantee about investment risk.

This study compares objective variants of a direct deep allocation policy under common architecture, inputs and evaluation conditions. It changes cost and downside treatment while preserving chronological decision order: allocations precede the returns they earn [10]. Trading costs are measured against holdings after market drift, consistent with portfolio turnover evaluation [11]. The comparison concerns these controlled choices, rather than unrelated published return figures.

The OLPS toolbox distributes historical price-relative matrices for portfolio experiments [12]. A fixed benchmark sequence permits reproducible comparison but does not reconstruct contemporaneous market membership or establish live trading returns. The contribution is a bounded, auditable comparison of objective choices on published data, retaining unfavourable outcomes and stating the assumptions about costs, information availability and benchmark coverage.

II. RELATED WORK

A. Portfolio allocation, trading costs and evaluation

Markowitz characterizes efficient expected-return/variance combinations using portfolio covariances [1]. Variance treats deviations symmetrically and is not automatically a downside-specific measure. DeMiguel, Garlappi and Uppal estimate weights from historical observations, evaluate the next observation, and compare optimized strategies with naive diversification [11]. Their turnover and cost analysis supports including a simple diversified benchmark in an evaluation of learned portfolio policies.

Cover defines constant-rebalanced wealth and a universal policy using previous observations, compared with the best constant portfolio selected in hindsight [13]. That theoretical benchmark does not promise finite-horizon profit after realistic costs. Li and Hoi review online allocation families and price-relative wealth accounting [10]. Both distinguish a feasible sequential policy from a portfolio chosen retrospectively using the full evaluation sequence.

Magill and Constantinides model proportional costs with divisible assets and continuous trading opportunities [2]. Constantinides explains how a no-transaction region reduces trading frequency and volume [3]. The present discrete observations do not inherit those continuous-time assumptions. Their accounting nevertheless requires the same practical distinction: costs concern actual trades. Comparing a new allocation with drifted holdings, as in Li and Hoi, captures trades omitted by differences between target weights alone [10].

Sharpe relates return above a riskless rate to return variability [14], later distinguishing expected from historical differential-return ratios and clarifying the benchmark [15]. These summaries do not completely describe adverse outcomes. Lo shows that aggregation depends on serial covariances, so customary annualization need not hold for autocorrelated returns [16]. Reported ratios must identify their frequency, benchmark and aggregation assumptions.

Searching many configurations can create apparent improvement. White tests the best member of a model collection while accounting for the search [17]; Sullivan, Timmermann and White apply a related bootstrap treatment to trading rules [18]. This concern extends to architectures and risk penalties. The present study retains its compared configurations and common seeds rather than interpreting a selected winning run as a generally superior investment strategy.

B. Neural representations and sequential decision learning

Backpropagation learns hidden representations through derivatives of an output objective [4]. LeCun, Bengio and Hinton describe layered representations and distinguish training fit from generalization [5]. These ideas support nonlinear policies without implying that greater depth benefits financial allocation. Bengio, Simard and Frasconi show difficulties learning long dependencies with gradient descent [19], motivating explicit input histories and training horizons rather than unrestricted increases in memory or capacity.

Temporal-difference prediction learns from differences between successive predictions [20]. Q-learning estimates action values without requiring a complete transition model, with convergence tied to specified state and sampling conditions [21]. Kaelbling, Littman and Moore relate these methods to exploration, state representation and delayed consequences [6]. These foundations define sequential learning alternatives; they do not provide unrestricted convergence guarantees for a nonlinear policy in changing financial markets.

The deep Q-network combines convolutional action-value approximation, replayed experience and a target network in game control [22]. Portfolio weights form a different, continuous action representation. The game results demonstrate a decision-learning construction, rather than financial profitability or tail-risk reduction. The present allocator directly parameterizes weights instead of transferring a discrete-action learner without specifying how it would search allocations.

C. Direct and continuous policy optimization

Williams's REINFORCE adjusts a parameterized stochastic distribution through scalar reinforcement [23]. Sutton, McAllester, Singh and Mansour express policy gradients through visitation, action-probability derivatives and action values [24]. These works distinguish direct policy optimization from obtaining a policy through value estimates. Here, hidden representations and allocation outputs are differentiated through financial reinforcement, without future-price labels. This direct construction is not claimed to implement those stochastic estimators.

Silver and colleagues derive a deterministic policy gradient through action values, distinguishing behavior and target policies [25]. DDPG combines a deterministic actor, learned critic, replay and target networks [26]. Continuous weights make these relevant methodological alternatives, but the present experiment directly differentiates its historical objective without a DDPG critic. DDPG's physical-control results cannot establish portfolio performance or validate financial observation assumptions.

TRPO controls policy changes while distinguishing its theoretical improvement argument from practical approximations [27]. PPO introduces probability-ratio surrogates, including a clipped objective [28]. The cited

PPO source is its original submitted version; later revisions are excluded. These update rules are relevant alternatives rather than implemented comparators here, and neither establishes improved net portfolio returns without a financial evaluation.

Asynchronous advantage actor-critic uses parallel actor-learners, a value component, multistep returns and entropy regularization [29]. Adam adapts updates through moment estimates and initialization bias correction [30]. Architecture, objective and optimization are separable choices: parallel processing does not make market observations independent, and an adaptive optimizer does not validate a financial objective. The present comparison therefore holds its training construction fixed while changing cost and downside treatment.

Jiang, Xu and Liang supply the closest admitted deep portfolio precedent, with convolutional and recurrent architectures, commissions and cryptocurrency evaluation [7]. They report drawdown alongside wealth and a reward-to-variability statistic. The present feed-forward network and stock benchmarks extend the methodological question under different conditions; they do not reproduce that experiment. Its published outcomes are not treated as results of the current matched-objective comparison.

D. Risk-sensitive reinforcement learning

Tamar, Di Castro and Mannor formulate mean-variance, variance-constrained and Sharpe objectives under stated policy and process assumptions [8], and estimate cumulative-reward means and second moments jointly [31]. Prashanth and Ghavamzadeh separate discounted from average-reward variance, with distinct actor, critic and multiplier updates [32]. These precedents show why risk evaluation needs its own construction; a mean-return critic does not estimate every risk quantity.

Chow and Ghavamzadeh constrain CVaR of cumulative discounted cost and distinguish trajectory-based from incremental learning [9]. Chow, Tamar, Mannor and Pavone relate CVaR-sensitive decisions to bounded model perturbations [33]. Their differing risk-level conventions require explicit loss direction and tail-probability definitions. A penalty on each negative period return is not equivalent to CVaR of terminal loss; the trained objective must remain distinct from later evaluation statistics.

Tamar, Chow, Ghavamzadeh and Mannor distinguish static coherent cumulative-return risk from time-consistent dynamic risk [34]. Neither formal property follows from naming a downside penalty. Tamar, Glassner and Mannor derive a CVaR gradient conditioned on the adverse return tail, with a quantile baseline and distributional assumptions [35]. These results motivate careful definitions rather than unconditional finite-sample reliability. Downside sensitivity is consequently evaluated empirically, allowing improvement, deterioration or inconsistent effects.

III. METHODOLOGY

A. Historical information set and benchmark data

The experiment uses price-relative matrices from a pinned OLPS repository release. The snapshot is commit 9120783cd59a7966b0f78e2b5668030a4378b8af, dated July 13, 2015. The published toolbox description supplies the benchmark context [12]. Original matrix files are retained; conversion to a decimal table preserves every row and round-trips every floating-point value exactly. No interpolation, deletion of returns or additional price adjustment is applied. The three matrices contain no individual observation dates or verified asset-name mapping, so allocations are evaluated by original return-row index and anonymous asset column.

Table 1 specifies chronological partitions as zero-based, half-open source-row intervals. The first twenty observations provide the initial history and are not training rewards. Validation and test states may include earlier observed returns from the preceding partition, as those returns are already available when the allocation is chosen. No validation or test return enters the fitted normalization constants. The windows are fixed before training, with no rolling search over test periods.

Dataset	Risky assets	Original rows	Training interval	Validation interval	Test interval	Reward counts train/validation/test
NYSE(O)	36	5651	[20,3390)	[3390,4520)	[4520,5651)	3370/1130/1131
SP500	25	1276	[20,765)	[765,1020)	[1020,1276)	745/255/256
DJIA	30	507	[20,304)	[304,405)	[405,507)	284/101/102

Table 1. Fixed data partitions. Counts refer to observed returns, not reconstructed calendar dates.

These are retrospective benchmark universes. The original OLPS documentation reports an SP500 observation window from January 2, 1998 to January 31, 2003 and selects the largest-capitalization constituents in April 2003,

after that window. These are provider-reported endpoints; the matrix itself supplies no row-level dates. Its membership therefore contains hindsight. The matrices do not establish complete coverage of delisted assets, historical investability or the treatment of every corporate action and dividend. They support controlled comparisons on the recorded sequences; they do not establish unbiased market-wide profits.

B. Policy observations and allocations

Let y_t contain the positive risky-asset price relatives earned at observation t . Cash is included as a separate component with relative one. Before earning y_t , the policy observes the flattened logarithms of risky-asset relatives $y_{t-20}, \dots, y_{t-1}$; the constant cash component is omitted from its input. Each feature is centered and scaled using training states only, with a minimum scale of 10^{-6} . The policy is a fully connected network with ReLU hidden layers of 32 and 16 units and a softmax output over all risky assets and cash. Its weights are nonnegative and sum to one, prohibiting leverage and short sales. Hidden weights have zero-mean Gaussian variance $2/\text{fan-in}$, following the rectifier initialization rule [36]; output weights have Gaussian standard deviation 0.01 and all biases are zero. The parameters are optimized directly through the realized financial reward. This is a deterministic continuous-allocation policy trained by differentiating the known historical market and transaction model. It does not estimate a price label or a Bellman value function. The design is inspired by direct deep portfolio optimization [7], but uses a feedforward state encoder without a convolutional layer, recurrence or a preceding-allocation input. Holdings remain necessary for cost accounting even though they are absent from the policy's observation. This is an objective ablation with a restricted policy class, rather than an exact reproduction of Jiang and colleagues' architecture.

C. Exact self-financing transaction costs

Immediately before trading, the preceding target w_{t-1} has drifted to q_t . Writing the cash-augmented relative as y_t , the accounting equations are

$$\begin{aligned} q_t &= (w_{t-1} \odot y_{t-1}) / (w_{t-1} \cdot y_{t-1}). \\ \mu_t + c \sum_{j \text{ risky}} |\mu_t w_{t,j} - q_{t,j}| &= 1. \\ V_t = V_{t-1} \mu_t (w_t \cdot y_t), \quad r_t &= \log \mu_t + \log(w_t \cdot y_t). \end{aligned}$$

Here c is the fee per dollar of risky assets bought or sold, μ_t is the fraction of pre-trade wealth remaining after fees, V_t is wealth and r_t is net log growth. Cash transfers themselves incur no fee. Charging both purchases and sales makes a complete risky-to-risky switch cost approximately twice the fee coefficient for small c . The model therefore does not interpret c as a quoted round-trip brokerage rate. The cost convention accounts for drifted holdings and scales the new holdings by post-fee wealth, consistent with the portfolio accounting distinction in prior work [10] [7]. The implicit equation has a unique solution for the tested coefficients; forty-eight bisection steps solve it on $[(1-c)/(1+c), 1]$. All independent partitions begin with unit wealth in cash. There is no terminal liquidation.

The objective for a trajectory of T rewards is

$$J(\theta; c, \lambda) = (1/T) \sum_t [r_t - \lambda \min(r_t, 0)^2].$$

The penalty measures squared negative daily log growth relative to a zero target. Its units and horizon are explicit. It is neither conditional value-at-risk nor a constraint on maximum drawdown, and it does not optimize the evaluation zero-target downside ratio. Increasing λ can favor lower exposure as well as a different risky allocation. Prior risk-sensitive policy work motivates specifying the precise reward functional rather than attaching a general risk-control claim to any penalty [8] [34].

For completeness, let $s_j = \text{sign}(\mu_t w_{t,j} - q_{t,j})$ for risky components and $D = 1 + c \sum_j s_j w_{t,j}$. Away from trade-sign boundaries, implicit differentiation gives $\partial \log \mu_t / \partial w_{t,j} = -c s_j / D$ and $\partial \log \mu_t / \partial q_{t,j} = c s_j / (\mu_t D)$. The gradient through drifted holdings also reaches the preceding action. For a gradient vector g with respect to q_t , a preceding action a and relative z contribute $(z/(a \cdot z)) \odot [g - (g \cdot q_t)1]$. These derivatives are backpropagated through the downside term, softmax and hidden layers; the selected derivative is zero at exact absolute-value and ReLU kinks. Finite-difference checks cover both current and adjacent-action effects.

D. Matched objectives and checkpoint selection

Five objective settings share the same state, network and training budget. The return-only setting uses $c=0$ and $\lambda=0$. The cost-aware setting uses $c=0.0025$ and $\lambda=0$. Downside settings use $c=0.0025$ and λ in $\{1, 10, 100\}$; $\lambda=10$ is the primary comparison, with the other values fixed as sensitivities. Five seeds, numbered zero through four, pair initial parameters and block sampling across settings within each dataset. This yields seventy-five fits. A fit makes 600 ascent updates with contiguous blocks of 128 rewards. Noninitial blocks include the preceding policy action in the computational graph, omit its own reward and differentiate the first included fee through it.

Overlapping blocks underweight observations near partition edges relative to the interior, so the sampled objective is not claimed to be an unbiased full-trajectory gradient estimator.

Adam follows Algorithm 1 of the first public version of Kingma and Ba [30]. In that version's innovation notation, $\beta_1=0.1$ and $\beta_2=0.001$: $m \leftarrow \beta_1 g + (1-\beta_1)m$ and $v \leftarrow \beta_2 g^2 + (1-\beta_2)v$. Bias denominators are $1-(1-\beta_1)^k$ and $1-(1-\beta_2)^k$ at update k . The learning rate is 0.0002 and the stabilizing constant is 10^{-8} . Ascent maximizes the stated reward. The numerical calculations use Python 3.6.2 and NumPy 1.13.1, releases public before 25 July 2017. Source retrieval, lossless file conversion and manuscript rendering use the current administrative environment; these operations do not supply a later learning algorithm or numerical experiment library.

Update zero and every fiftieth update are scored on the complete validation partition using the matching setting's c and λ . The greatest validation objective selects the checkpoint; strict ties retain the earlier checkpoint. Eligibility of update zero provides an explicit fallback to initialization. A retained initial checkpoint must not be described as successful learning. Selection compares complete validation trajectories reset to cash, rather than sampled training blocks. Test parameters and normalization remain frozen, with no online updates, test-guided choice of λ or selection of the most favorable seed. The primary fee is 0.0025. Each selected policy's identical actions are additionally evaluated at fees 0, 0.001 and 0.005.

E. Baselines, metrics and uncertainty

The baselines are zero-yield cash, equal-weight constant rebalancing and equal-weight buy-and-hold over the risky columns. All begin in cash. Buy-and-hold enters once, paying the same initial fee, and thereafter follows its own drifted weights with zero subsequent turnover. Constant rebalancing resets the risky weights to equality each observation. These choices separate learned allocation from simple diversification and avoid granting a baseline free initial investment [13] [11].

For each test path, terminal wealth is reported alongside annualized downside deviation, maximum drawdown, mean risky-dollar turnover, mean target cash weight and cumulative log fee drag. Downside deviation is $\sqrt{252 \text{ mean}(\min(R_t, 0)^2)}$, where $R_t = \exp(r_t) - 1$. Maximum drawdown uses the running wealth peak including the initial unit of wealth. Turnover is $\sum_{j \text{ risky}} |\mu_t w_{t,j} - q_{t,j}|$; log fee drag is $-\sum_t \log \mu_t$. Annual return is $\exp(252 \text{ mean}(r_t)) - 1$. Sharpe uses the mean simple return divided by its sample standard deviation, times $\sqrt{252}$, with zero cash yield [14] [15]. The descriptive zero-target downside ratio is $\sqrt{252} \text{ mean}(R_t) / \sqrt{\text{mean}(\min(R_t, 0)^2)}$; its denominator here is daily downside deviation, before annualization. Cash ratios are undefined because their denominators are zero. Conventional annualization is descriptive, without a correction for serial covariance [16].

Neural summaries give the median and full range across all five seeds. Seed variability measures algorithmic variation on a common market path, not independent financial replications. The two primary paired contrasts are cost-aware minus return-only and downside $\lambda=10$ minus cost-aware, at the primary fee. For each observation, net log-return differences are averaged across paired seeds. A stationary bootstrap uses circular blocks with geometric mean length twenty, 2000 replicates and a fixed seed, following the sequential resampling construction of Politis and Romano [37]. The table reports percentile intervals for the mean daily contrast. These intervals are exploratory and conditional on the fitted policies; no models are refitted inside the bootstrap, stationarity is not guaranteed, and intervals are not adjusted for multiple comparisons or model search [17] [18].

The code, protocol, runtime inventory, validation histories, selected parameters and test trajectories are described in the Data and Code Availability statement and supplied in the accompanying reproduction package.

IV. RESULTS AND DISCUSSION

A. Main results under a common fee

Table 2 reports test outcomes at $c=0.0025$, including every prespecified objective and baseline. Each neural statistic is summarized separately across five seeds; the metric medians do not describe a single selected trajectory. The recorded comparisons retain all seventy-five model fits and all 336 fee-specific paths. Detailed seed metrics, validation histories and fee sensitivities are available through the public GitHub release linked in Data and Code Availability.

Appendix A (Tables A1-A3) reports annualized return, annualized Sharpe with zero cash yield, the zero-target downside ratio and total log fee drag for every primary-fee policy. The complete seed ranges accompany the neural medians. Cash ratios are undefined because their denominators are zero.

Dataset	Policy	Wealth [seed range]	DD	DDev	Turnover	Cash
NYSE(O)	Cash	1.0000	0.0000	0.0000	0.0000	1.0000
NYSE(O)	Equal-weight rebalancing	2.1653	0.1623	0.0920	0.0124	0.0000
NYSE(O)	Buy-and-hold	2.1406	0.1722	0.0961	0.0009	0.0000
NYSE(O)	Return-only	0.3044 [0.1948, 0.5038]	0.7455	0.1989	0.7972	0.0094
NYSE(O)	Cost-aware	2.0845 [2.0731, 2.0905]	0.1595	0.0903	0.0227	0.0229
NYSE(O)	Downside $\lambda=1$	2.0843 [2.0738, 2.0906]	0.1594	0.0902	0.0225	0.0234
NYSE(O)	Downside $\lambda=10$	2.0771 [1.9777, 2.0870]	0.1585	0.0896	0.0211	0.0279
NYSE(O)	Downside $\lambda=100$	0.8607 [0.8313, 0.9197]	0.2329	0.0587	0.2341	0.4511
SP500	Cash	1.0000	0.0000	0.0000	0.0000	1.0000
SP500	Equal-weight rebalancing	0.8290	0.2998	0.1879	0.0172	0.0000
SP500	Buy-and-hold	0.8128	0.3022	0.1820	0.0039	0.0000
SP500	Return-only	0.5895 [0.4352, 0.8273]	0.4448	0.2650	0.8599	0.0089
SP500	Cost-aware	0.8253 [0.7402, 0.8283]	0.2936	0.1815	0.0358	0.0386
SP500	Downside $\lambda=1$	0.8271 [0.8253, 0.8283]	0.2926	0.1815	0.0345	0.0386
SP500	Downside $\lambda=10$	0.8273 [0.8253, 0.8283]	0.2926	0.1811	0.0345	0.0387
SP500	Downside $\lambda=100$	0.7501 [0.7420, 0.7792]	0.2701	0.1110	0.2225	0.4446
DJIA	Cash	1.0000	0.0000	0.0000	0.0000	1.0000
DJIA	Equal-weight rebalancing	0.9724	0.2212	0.2052	0.0243	0.0000
DJIA	Buy-and-hold	0.9627	0.2182	0.2061	0.0098	0.0000
DJIA	Return-only	0.6271 [0.4377, 0.8584]	0.4136	0.3568	1.3245	0.0063
DJIA	Cost-aware	0.9602 [0.7434, 0.9665]	0.2166	0.1995	0.0715	0.0329
DJIA	Downside $\lambda=1$	0.9619 [0.7458, 0.9679]	0.2158	0.1991	0.0613	0.0328
DJIA	Downside $\lambda=10$	0.9010 [0.8937, 0.9600]	0.2025	0.1908	0.2299	0.0631
DJIA	Downside $\lambda=100$	0.8522 [0.8139, 0.8784]	0.1742	0.1166	0.3296	0.4717

Table 2. Primary-fee outcomes. Wealth is relative to initial wealth one; brackets show the full seed range.

Other neural columns show their separate seed medians. DD denotes maximum drawdown; DDev is annualized downside deviation. Turnover is risky dollars traded divided by pre-trade wealth per observation, and Cash is mean target cash weight. All risk and allocation values are fractions. Baselines are deterministic.

Cost-aware training improves mean daily net log growth over the matched return-only setting in fourteen seed comparisons, with one SP500 tie because both settings retain the same initialization. NYSE median terminal wealth changes from 0.3044 to 2.0845, SP500 from 0.5895 to 0.8253, and DJIA from 0.6271 to 0.9602. Median risky-dollar turnover changes from 0.7972 to 0.0227, 0.8599 to 0.0358, and 1.3245 to 0.0715, respectively. The improvement therefore concerns a substantial difference in trading behavior, rather than simply evaluating the same gross path with a fee deducted afterward.

These gains do not establish superiority over simple diversification. Equal-weight rebalancing ends with wealth 2.1653, 0.8290 and 0.9724 on NYSE, SP500 and DJIA, exceeding the corresponding cost-aware median on every

dataset. All primary-free stock-allocation paths on SP500 and DJIA finish below the initial unit of wealth, so cash exceeds them over these particular test windows. On NYSE the cost-aware network also finishes below buy-and-hold wealth 2.1406. The learned policies must consequently be assessed as alternatives with different trading and risk behavior, not as generally superior investment strategies.

The cost-aware median should also be interpreted against the near-uniform initialization. For a uniform allocation over n risky assets and cash, the cash reference is $1/(n+1)$: 0.0270, 0.0385 and 0.0323 for NYSE, SP500 and DJIA. Small output-layer initialization gives an allocation close to this reference. Cost-aware median cash weights are 0.0229, 0.0386 and 0.0329, respectively. SP500 median wealth, 0.8253, is close to the equal-weight result of 0.8290. A descriptive reconstruction of the initial and retained parameters shows that three of five SP500 cost-aware fits and two of five DJIA fits retain initialization. All NYSE fits change, but their mean daily sum of absolute differences between fitted and initial allocations on the test partition ranges from 0.0248 to 0.0513 across seeds. Other changed SP500 and DJIA fits depart substantially from their starts, so a near-uniform median cash weight does not establish unchanged risky-asset allocations in every fit. The diagnostic is a post hoc description of retained policies, rather than a prespecified hypothesis test.

The cost-aware comparison consequently includes conservative or initialized allocations as well as learned changes; its gain over return-only training does not by itself establish better asset selection. Cash exposure and extra trading distinguish these policies from equal-weight rebalancing. Cash can dilute gains in a rising window and cushion losses in a falling one. Their separate effects on the wealth gap are not identified by this experiment.

B. Downside penalties and exposure

The primary downside coefficient, $\lambda=10$, gives lower or equal paired downside deviation and maximum drawdown than cost-aware training on each dataset, including ties from identical retained initial policies. The size and return implications differ. On NYSE, median downside deviation changes from 0.0903 to 0.0896 and drawdown from 0.1595 to 0.1585, while every paired seed loses terminal wealth. The risk reduction is small in this setting. On SP500, three pairs have identical test allocations; the two changed pairs have higher wealth and lower measured risk. A positive seed-averaged contrast therefore does not imply improvement in the median paired result, which is zero.

DJIA makes the distinction between summaries especially clear. Median wealth declines from 0.9602 to 0.9010 with $\lambda=10$, while downside deviation declines from 0.1995 to 0.1908 and drawdown from 0.2166 to 0.2025. Median turnover rises from 0.0715 to 0.2299, roughly tripling, while mean target cash weight rises from a seed median of 0.0329 to 0.0631, nearly doubling. Three paired wealth changes are negative and two are positive. Improvements in the two weaker cost-aware fits make the seed-averaged mean daily log-return contrast positive even though the median paired change is negative. This pattern is retained rather than resolved by reporting only one summary.

Increasing λ does not yield a monotone improvement in the broader investment outcome. With $\lambda=100$, median mean cash weight rises to 0.4511, 0.4446 and 0.4717 on the three benchmarks, compared with 0.0229, 0.0386 and 0.0329 for cost-aware training. Downside deviation falls, but median turnover rises to 0.2341, 0.2225 and 0.3296. Median wealth is then 0.8607, 0.7501 and 0.8522. NYSE maximum drawdown increases to 0.2329 despite its lower downside deviation. The penalty on individual negative log returns therefore cannot be interpreted as a drawdown constraint. Greater cash exposure and increased trading accompany the strong-penalty outcomes; the experiment does not isolate improved risky-asset selection from those changes.

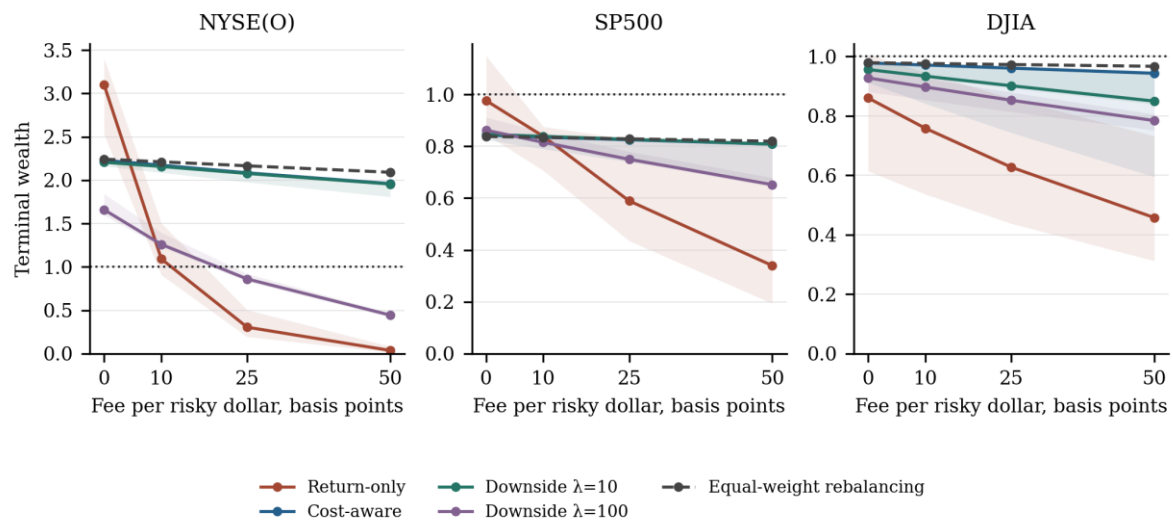


Figure 1. Terminal wealth across the fixed fee sweep. Neural curves show the median across all five seeds; the bands show their minimum and maximum. Equal-weight rebalancing supplies a reference curve; the horizontal dotted line marks cash wealth one. Selected actions are identical across evaluation fees, so the plot measures accounting sensitivity without retraining. Panel scales differ across datasets.

The fee sweep in Figure 1 explains why a gross-return assessment is incomplete. On NYSE, the return-only policy has median wealth 3.1004 at zero fee, compared with 2.2285 for cost-aware training. At the primary fee those values are 0.3044 and 2.0845; at $c=0.005$ they are 0.0362 and 1.9601. Its median cumulative log fee drag at the primary fee is 2.2572, versus 0.0642 for cost-aware training. Cost-awareness reduces this vulnerability, but it does not dominate every gross-return comparison. The strong downside setting is also fee-sensitive: on NYSE its median wealth changes from 1.6588 at zero fee to 0.8607 at the primary fee. A risk penalty can coexist with expensive rebalancing.

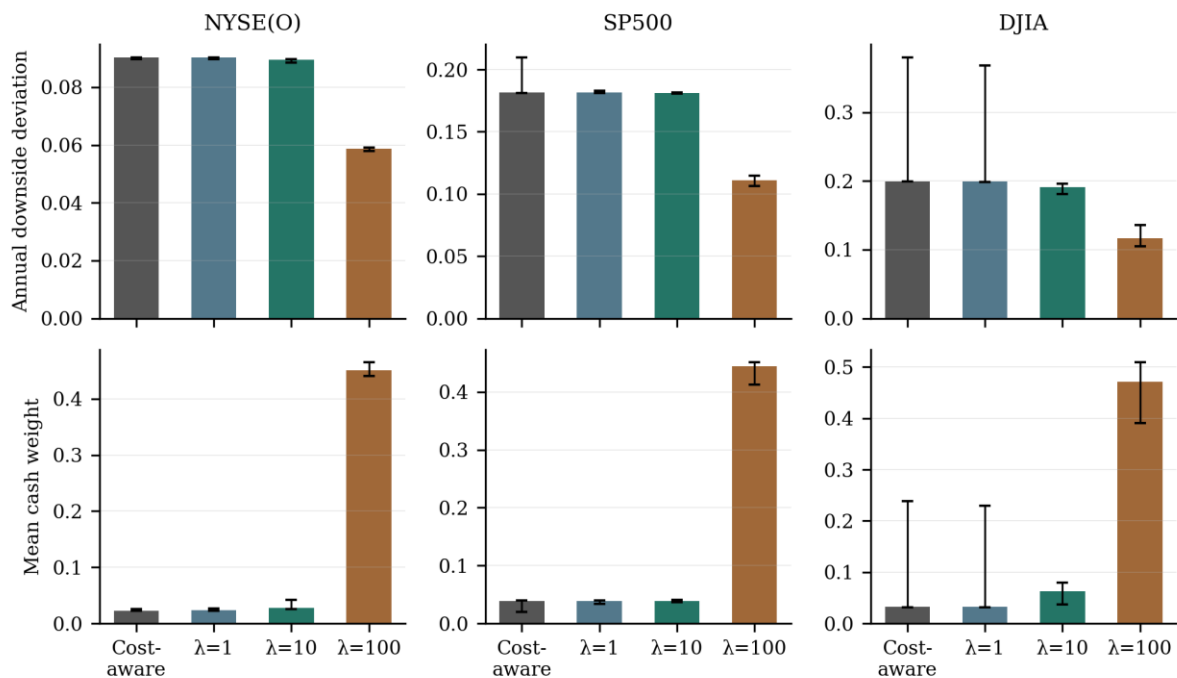


Figure 2. Annualized downside deviation and mean cash weight at the primary fee for cost-aware and downside-sensitive policies. Bar heights are separate seed medians; intervals show the full seed range. Lower downside deviation at the strong penalty coincides with greater cash exposure. The panels describe observed associations and do not identify a unique causal mechanism.

C. Paired uncertainty and retained checkpoints

Dataset	Paired contrast	Daily mean, basis points	Exploratory 95 percent interval
NYSE(O)	Cost-aware minus return-only	17.2534	[10.2209, 23.5474]
NYSE(O)	Downside $\lambda=10$ minus cost-aware	-0.1040	[-0.1943, -0.0181]
SP500	Cost-aware minus return-only	10.1722	[2.3743, 17.7350]
SP500	Downside $\lambda=10$ minus cost-aware	1.4259	[0.2896, 2.6050]
DJIA	Cost-aware minus return-only	35.9018	[18.4552, 54.1659]
DJIA	Downside $\lambda=10$ minus cost-aware	4.6524	[-5.2882, 14.8318]

Table 3. Primary paired differences in mean daily net log return, in basis points. Intervals are exploratory stationary-bootstrap percentile intervals conditional on the five fitted policies. The statistic first averages paired differences across seeds and then resamples observations. These are neither intervals for median seed performance nor simultaneous guarantees for all comparisons.

The cost-aware contrasts have positive exploratory intervals on each benchmark. The downside contrast is negative on NYSE, positive on SP500, and has an interval spanning zero on DJIA. The DJIA interval is particularly sensitive to a short test path and heterogeneous retained fits. The analysis does not refit models under resampling or correct for selection among objectives. It therefore supports describing these recorded paths, without establishing a population-wide trading advantage or a confirmatory hypothesis test.

Validation retains update zero in fifteen of the seventy-five fits. SP500 contributes eleven: one return-only, three cost-aware, three $\lambda=1$ and four $\lambda=10$ fits. DJIA contributes four: two cost-aware and two $\lambda=1$ fits. NYSE contributes none. Those paths use randomly initialized parameters, although the complete training runs were executed and their later checkpoints evaluated. Their retention indicates that training did not improve the matching validation objective enough to displace initialization. The SP500 downside findings in particular include fallback behavior rather than consistently learned risk-sensitive allocations.

D. Interpretation and limitations

The clearest empirical result is the benefit of making trading costs part of this policy's learning and validation objectives. Its strength lies in a matched construction with exact fee accounting and common information order. The return-only policy can exploit gross movement while changing weights so often that commissions consume the apparent gain. Cost-aware training addresses that failure mode, but equal-weight rebalancing remains competitive and exceeds its median net wealth here. This is consistent with the methodological importance of naive diversification in portfolio evaluation [11], without treating a result from another study as evidence for these particular test paths.

The downside objective offers a different tradeoff. Small risk reductions can accompany lower wealth, and a large coefficient can reduce downside deviation while increasing turnover and worsening another risk measure. Cash is an available action, so reduced exposure is part of the policy response. A stronger claim about risk-sensitive asset selection would require exposure-matched or cash-constrained comparisons specified independently of these test outcomes. The present findings do not establish such a mechanism, and they do not inherit CVaR or time-consistency properties from other risk-sensitive learning formulations [9] [34].

Several design choices restrict external validity. The universes are retrospective, the data do not establish point-in-time membership, and the SP500 selection uses a later constituent date within its historical documentation. The model omits spread, slippage, market impact, taxes, borrowing and capacity. Cash earns no interest and final liquidation is absent. The feedforward policy omits holdings from its observation and can only infer a limited history from its fixed window. Only one architecture, one learning budget and a small set of prespecified penalties are evaluated. The initial checkpoints and wide seed ranges show that the optimizer does not reliably produce an improved validation policy in every setting. Neither an unrestricted policy search nor comparisons against stronger active allocation algorithms were performed.

The statistical limitations are equally relevant. Seeds share the same financial observations, conventional annualization does not correct serial covariance [16], and resampling fitted paths does not include uncertainty from repeated training or changing market regimes. The short DJIA test limits the strength of its interval interpretation. Prespecifying contrasts and retaining unfavorable results improves auditability but does not replace

a model-search correction [17] [18]. Conclusions remain confined to the recorded benchmark partitions and explicit assumptions.

V. CONCLUSION

On these historical benchmark partitions, including proportional costs in direct deep policy training substantially reduces turnover and improves net returns over the matched return-only policy at the primary fee. It does not exceed equal-weight rebalancing in median terminal wealth. Adding a squared negative-log-return penalty reduces measured downside risk with mixed wealth effects, and a strong penalty increases cash exposure and trading rather than guaranteeing a better investment outcome. Exact self-financing accounting, chronological validation, disclosure of retained initial checkpoints and preservation of all seeds are necessary to interpret these results. The study supports a bounded comparison of objective choices, with no general profitability or comprehensive risk-control claim.

Declarations

Funding and Material Support

The author reports no funding, grants or material support for this study.

Conflicts of Interest

The author declares no financial or nonfinancial conflicts of interest.

Data and Code Availability

The historical input matrices are available from the OLPS repository at <https://github.com/OLPS/OLPS/tree/9120783cd59a7966b0f78e2b5668030a4378b8af>. Code and recorded results are publicly available in GitHub release v1.0.0. Its reproduction-package.zip archive supplies the frozen protocol, training and analysis code, selected parameters and normalization, every validation checkpoint, every test trajectory, per-seed and aggregate metrics, the descriptive initialization comparison and file checksums. Appendix A, Tables A1-A3, is included in this manuscript. The package also contains a retrieval utility and the expected hashes of the original matrices and lossless CSV conversions. Input matrices are retrieved from their original source; third-party papers and runtime binaries are not redistributed. Two private absolute protocol locations are replaced with package-relative paths, with both original and public hashes recorded in the manifest.

APPENDIX A. ADDITIONAL PERFORMANCE MEASURES

Tables A1-A3 report additional descriptive measures at the primary transaction-cost coefficient of 0.0025 per risky dollar traded. The definitions and conventional annualization at 252 observations per year are those in Section III.E. Annualized return uses net log growth; the Sharpe statistic uses net simple returns and zero cash yield. The zero-target downside ratio uses mean net simple return relative to a zero target divided by daily downside deviation, with the annualization factor specified in the methods. Total log fee drag is the accumulated negative logarithm of the fee multiplier, rather than a percentage of initial wealth.

Each neural-policy entry is the median [minimum, maximum] across all five seeds. Deterministic baselines have one value. Ratios with zero denominators remain undefined. Display rounding does not change the saved metrics, and annualization remains descriptive rather than a forecast.

Table A1: NYSE(O)

Policy	Annualized return (percent)	Zero-cash-rate annualized Sharpe	Zero-target downside ratio	Total log fee drag
Cash	0.000	undefined	undefined	0.000000
Equal-weight rebalancing	18.783	1.252	1.987	0.035058
Buy-and-hold	18.480	1.196	1.883	0.002497
Return-only	-23.278 [-30.546, -14.167]	-0.647 [-1.082, -0.429]	-1.016 [-1.630, -0.682]	2.257189 [1.821122, 2.676089]
Cost-aware	17.782 [17.637, 17.857]	1.218 [1.210, 1.223]	1.927 [1.916, 1.932]	0.064162 [0.054925, 0.074185]

Policy	Annualized return (percent)	Zero-cash-rate annualized Sharpe	Zero-target downside ratio	Total log fee drag
Downside $\lambda=1$	17.778 [17.647, 17.858]	1.219 [1.212, 1.224]	1.927 [1.919, 1.934]	0.063537 [0.054697, 0.072879]
Downside $\lambda=10$	17.689 [16.409, 17.813]	1.220 [1.157, 1.227]	1.929 [1.828, 1.942]	0.059655 [0.055446, 0.089675]
Downside $\lambda=100$	-3.288 [-4.032, -1.848]	-0.332 [-0.428, -0.166]	-0.500 [-0.637, -0.249]	0.662390 [0.618172, 0.693006]

*Table A1. NYSE(O) primary-fee descriptive metrics.***Table A2: SP500**

Policy	Annualized return (percent)	Zero-cash-rate annualized Sharpe	Zero-target downside ratio	Total log fee drag
Cash	0.000	undefined	undefined	0.000000
Equal-weight rebalancing	-16.854	-0.532	-0.781	0.011000
Buy-and-hold	-18.456	-0.641	-0.930	0.002497
Return-only	-40.561 [-55.906, -17.023]	-1.104 [-1.917, -0.569]	-1.579 [-2.627, -0.836]	0.551011 [0.022890, 0.808304]
Cost-aware	-17.218 [-25.631, -16.929]	-0.578 [-0.835, -0.567]	-0.847 [-1.196, -0.831]	0.022890 [0.020269, 0.108322]
Downside $\lambda=1$	-17.040 [-17.218, -16.929]	-0.569 [-0.578, -0.567]	-0.835 [-0.847, -0.831]	0.022078 [0.018726, 0.023227]
Downside $\lambda=10$	-17.023 [-17.218, -16.929]	-0.571 [-0.578, -0.567]	-0.836 [-0.847, -0.831]	0.022078 [0.018467, 0.025964]
Downside $\lambda=100$	-24.654 [-25.451, -21.779]	-1.763 [-1.884, -1.569]	-2.393 [-2.540, -2.111]	0.142484 [0.138417, 0.161764]

*Table A2. SP500 primary-fee descriptive metrics.***Table A3: DJIA**

Policy	Annualized return (percent)	Zero-cash-rate annualized Sharpe	Zero-target downside ratio	Total log fee drag
Cash	0.000	undefined	undefined	0.000000
Equal-weight rebalancing	-6.680	-0.072	-0.109	0.006206
Buy-and-hold	-8.972	-0.156	-0.232	0.002497
Return-only	-68.422 [-87.014, -31.419]	-2.179 [-4.009, -0.888]	-2.741 [-4.669, -1.249]	0.338395 [0.160130, 0.406204]
Cost-aware	-9.551 [-51.928, -8.063]	-0.191 [-1.631, -0.134]	-0.284 [-2.195, -0.201]	0.018242 [0.012933, 0.223415]
Downside $\lambda=1$	-9.143 [-51.557, -7.754]	-0.177 [-1.243, -0.123]	-0.264 [-1.639, -0.185]	0.015642 [0.014612, 0.243144]
Downside $\lambda=10$	-22.700 [-24.254, -9.596]	-0.823 [-0.920, -0.201]	-1.164 [-1.279, -0.300]	0.058644 [0.016916, 0.066910]

Policy	Annualized return (percent)	Zero-cash-rate annualized Sharpe	Zero-target downside ratio	Total log fee drag
Downside $\lambda=100$	-32.637 [-39.882, -27.405]	-2.557 [-3.017, -2.234]	-3.297 [-3.643, -2.951]	0.084085 [0.079472, 0.099659]

Table A3. DJIA primary-fee descriptive metrics.

REFERENCES

- [1] Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, 7(1), 77–91. <https://doi.org/10.1111/j.1540-6261.1952.tb01525.x>
- Magill, M. J. P., & Constantinides, G. M. (1976). Portfolio selection with transaction costs. *Journal of Economic Theory*, 13(2), 245–263. [https://doi.org/10.1016/0022-0531\(76\)90018-1](https://doi.org/10.1016/0022-0531(76)90018-1)
- Constantinides, G. M. (1986). Capital market equilibrium with transaction costs. *Journal of Political Economy*, 94(4), 842–862. <https://doi.org/10.1086/261410>
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536. <https://doi.org/10.1038/323533a0>
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. <https://doi.org/10.1038/nature14539>
- Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 4, 237–285. <https://doi.org/10.1613/jair.301>
- Jiang, Z., Xu, D., & Liang, J. (2017). A deep reinforcement learning framework for the financial portfolio management problem. *arXiv:1706.10059*. <https://arxiv.org/abs/1706.10059>
- Tamar, A., Di Castro, D., & Mannor, S. (2012). Policy gradients with variance related risk criteria. In *Proceedings of the 29th International Conference on Machine Learning*, pp. 935–942.
- Chow, Y., & Ghavamzadeh, M. (2014). Algorithms for CVaR optimization in MDPs. In *Advances in Neural Information Processing Systems*, 27.
- Li, B., & Hoi, S. C. H. (2014). Online portfolio selection: A survey. *ACM Computing Surveys*, 46(3), 35:1–35:36. <https://doi.org/10.1145/2512962>
- DeMiguel, V., Garlappi, L., & Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy? *Review of Financial Studies*, 22(5), 1915–1953. <https://doi.org/10.1093/rfs/hhm075>
- Li, B., Sahoo, D., & Hoi, S. C. H. (2016). OLPS: A toolbox for on-line portfolio selection. *Journal of Machine Learning Research*, 17(35), 1–5.
- Cover, T. M. (1991). Universal portfolios. *Mathematical Finance*, 1(1), 1–29. <https://doi.org/10.1111/j.1467-9965.1991.tb00002.x>
- Sharpe, W. F. (1966). Mutual fund performance. *The Journal of Business*, 39(S1), 119–138. <https://doi.org/10.1086/294846>
- Sharpe, W. F. (1994). The Sharpe ratio. *The Journal of Portfolio Management*, 21(1), 49–58. <https://doi.org/10.3905/jpm.1994.409501>
- Lo, A. W. (2002). The statistics of Sharpe ratios. *Financial Analysts Journal*, 58(4), 36–52. <https://doi.org/10.2469/faj.v58.n4.2453>
- White, H. (2000). A reality check for data snooping. *Econometrica*, 68(5), 1097–1126. <https://doi.org/10.1111/1468-0262.00152>
- Sullivan, R., Timmermann, A., & White, H. (1999). Data-snooping, technical trading rule performance, and the bootstrap. *The Journal of Finance*, 54(5), 1647–1691. <https://doi.org/10.1111/0022-1082.00163>
- Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157–166. <https://doi.org/10.1109/72.279181>
- Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1), 9–44. <https://doi.org/10.1007/BF00115009>
- Watkins, C. J. C. H., & Dayan, P. (1992). Q-learning. *Machine Learning*, 8(3–4), 279–292. <https://doi.org/10.1007/BF00992698>
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533. <https://doi.org/10.1038/nature14236>

- 23) Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4), 229–256. <https://doi.org/10.1007/BF00992696>
- 24) Sutton, R. S., McAllester, D. A., Singh, S. P., & Mansour, Y. (1999). Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, 12, pp. 1057–1063.
- 25) Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., & Riedmiller, M. (2014). Deterministic policy gradient algorithms. In *Proceedings of the 31st International Conference on Machine Learning, Proceedings of Machine Learning Research*, 32, 387–395.
- 26) Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2016). Continuous control with deep reinforcement learning. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*.
- 27) Schulman, J., Levine, S., Moritz, P., Jordan, M., & Abbeel, P. (2015). Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning, Proceedings of Machine Learning Research*, 37, 1889–1897.
- 28) Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv:1707.06347*. <https://arxiv.org/abs/1707.06347>
- 29) Mnih, V., Puigdomènech Badia, A., Mirza, M., Graves, T., Harley, T., Lillicrap, T. P., Silver, D., & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *Proceedings of the 33rd International Conference on Machine Learning, Proceedings of Machine Learning Research*, 48, 1928–1937
- 30) Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv:1412.6980*. <https://arxiv.org/abs/1412.6980>
- 31) Tamar, A., Di Castro, D., & Mannor, S. (2013). Temporal difference methods for the variance of the reward to go. In *Proceedings of the 30th International Conference on Machine Learning, Proceedings of Machine Learning Research*, 28(3), 495–503.
- 32) Prashanth, L. A., & Ghavamzadeh, M. (2016). Variance-constrained actor-critic algorithms for discounted and average reward MDPs. *Machine Learning*, 105(3), 367–417. <https://doi.org/10.1007/s10994-016-5569-5>
- 33) Chow, Y., Tamar, A., Mannor, S., & Pavone, M. (2015). Risk-sensitive and robust decision-making: A CVaR optimization approach. In *Advances in Neural Information Processing Systems*, 28, pp. 1522–1530.
- 34) Tamar, A., Chow, Y., Ghavamzadeh, M., & Mannor, S. (2015). Policy gradient for coherent risk measures. In *Advances in Neural Information Processing Systems*, 28, pp. 1468–1476.
- 35) Tamar, A., Glassner, Y., & Mannor, S. (2015). Optimizing the CVaR via sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 29, pp. 2993–2999.
- 36) He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 1026–1034. <https://doi.org/10.1109/ICCV.2015.123>
- 37) Politis, D. N., & Romano, J. P. (1994). The stationary bootstrap. *Journal of the American Statistical Association*, 89(428), 1303–1313. <https://doi.org/10.1080/01621459.1994.10476870>