Journal of Mathematical Finance and Risk Management
ISSN: 3080-9983 (Print), ISSN: 3080-9991 (Online) | Vol. 2, No. 1, 2026

Shield-Consistent Double Dueling Q-Learning with Auditable Inventory Risk Budgets for Trade Execution

Isaac Kapoor

College of Business, Stony Brook University, New York, USA

Abstract: Large-order execution agents often propose actions that are subsequently changed by a risk-control layer. This creates a training-deployment mismatch: the value network estimates outcomes for unfiltered orders while actual execution follows the shield. We study shield-consistent Double Dueling Q-learning (SC-DDQN), which trains on the transitions generated by the same observable execution shield used at deployment, and extend it with an adaptive inventory-exposure envelope (BSC-DDQN). The shield operationalizes flow-sensitive urgency, schedule recovery, and deadline floors; the new envelope tightens the permissible post-trade inventory as observed volatility rises. We implement a finite-horizon, six-action execution environment with concave temporary impact and four persistent latent liquidity regimes. Three independently trained seeds are evaluated on 720 paired held-out synthetic paths against deterministic schedules, unshielded DDQN, post-hoc shielding, and shield-consistent ablations. SC-DDQN attains 15.05 basis points (bps) mean implementation shortfall, compared with 18.67 bps for time-weighted execution and 15.88 bps for post-hoc shielded DDQN; the risk-budget extension instead reaches 16.24 bps but reduces inventory exposure. A seed-level sensitivity analysis yields a 95% interval of [-0.50, 2.16] bps for the 0.83-bps advantage over post-hoc shielding; only three independent fits are available. We report tail losses, intervention frequency, and impact-coefficient sensitivity. All outcomes are generated by a disclosed simulator; they measure the interaction between learning and explicit execution constraints rather than realized venue profitability.

Keywords: Optimal execution, safe reinforcement learning, Double DQN, action shielding, implementation shortfall, conditional value-at-risk.

1. Introduction

Executing a known parent order differs from forecasting an asset return. The trader must exchange a fixed quantity for cash before a deadline, subject to the spread, finite liquidity, and the market exposure left in the residual inventory. Dynamic execution models [1], [2] formalize the resulting balance between slippage and uncertainty. Yet the price response to trading is sensitive to both impact dynamics [3] and order-book imbalance [4]; a policy optimized for an average state can become imprudent when liquidity vanishes or directional flow accelerates.

Reinforcement learning (RL) offers a natural way to update child-order sizes as signals change. The early execution study of Nevmyvaka et al. [5], a Double Q execution policy [6], a proximal policy optimization (PPO) formulation [7], and oracle policy distillation [8] illustrate distinct ways to learn execution controls. They do not remove an operational requirement: orders proposed by a learned model must still satisfy venue participation caps, client constraints and the execution deadline. Shielding provides an explicit architectural separation between action proposal and rule enforcement [9].

That separation raises a less frequently tested learning question. When a deterministic shield replaces a neural action, the action the network scores is not the order that reaches the market. A network fitted in the unshielded environment may therefore rank candidate actions according to counterfactual transitions that are never realized under the deployed system. A favorable result for a network-plus-shield system need not demonstrate that the learning process has adapted to the shield itself. Qin et al. [10] introduced a market-aware execution controller combining gated Q experts, adaptive risk charging, and a deterministic execution shield, and their ablation showed that explicit intervention accounts for much of the measured benefit. This gives a concrete basis for investigating whether the policy should learn inside its execution rules rather than only be corrected after training.

This paper makes three testable contributions. First, it models the shield as an action transformation inside the Bellman transition used by Double Dueling Q-learning; the learned value of a proposal reflects the action actually executed. Second, it assesses an optional volatility-responsive inventory exposure envelope enforced through discrete feasible-order projection and audited for infeasibility under the hard participation cap. Third, it performs matched-path experiments that separate the effects of a post-hoc shield, shield-consistent learning, and the inventory envelope. The emphasis is on attribution rather than combining every available neural-network enhancement.

The hypothesis is conditional: alignment may improve execution under the chosen synthetic market, but its size and even its sign are empirical matters. We therefore report ablations, mean and tail losses, seed-level uncertainty, risk exposure, regime-conditioned costs and an impact-coefficient shift. Quantities expressed in basis points are simulated implementation shortfalls, never measured live fills.

2. Background and Related Work

Market-impact and execution research supplies the structural assumptions rather than trading-rule parameters. Nonlinear temporary impact changes the optimal rate schedule [11]; adaptive mean-variance execution explicitly allows future decisions to react to changing information [12]. Empirical and theoretical analyses connect institutional order splitting to volatility [13] and explain why a concave square-root impact relationship is plausible [14]. Reviews of limit-order-book dynamics [15] and deep-learning evidence on

order-flow history [16] also motivate including imbalance among observable state variables. Cross-venue frictions in cryptocurrency trading [17] caution against interpreting stylized simulator costs as transferable transaction prices.

Recent execution work addresses latent or time-varying liquidity through deep value learning [18]. At the algorithmic level, the DQN family combines nonlinear function approximation with experience replay [19]; Double Q-learning reduces over-optimistic action values [20], and dueling networks factor state value from action advantage [21]. Prioritized replay [22] and Rainbow-style combinations [23] can improve sample efficiency, but including them in a component-attribution test introduces additional design changes. Our implementation uses uniform replay and the Adam family of stochastic optimizers [24] so that action filtering remains the principal manipulated variable.

LOB-based datasets and predictors have a complementary role. The FI-2010 benchmark [25] contains historical depth observations and DeepLOB [26] extracts spatial and temporal representations from order books; neither directly reveals the fills that an alternative policy would have obtained on an unexecuted order. This study therefore generates all experimental market paths and fill prices inside one explicitly parameterized simulator. Such a controlled design is appropriate for an internal validity test of the action shield, while leaving external validity to future order-book replay and shadow execution.

3. Execution Model and Measures

3.1. Market State and Admissible Action

Each episode liquidates a sell order of Q0 = 480 units over T = 60 sequential decision intervals. The discrete proposal alphabet is A = {0, 0.5, 1, 1.5, 2, 3}. For inventory x_t and proposal a_t, the unfiltered target is a_t x_t/(T-t); the released quantity is capped by 20% of currently observed market volume V_t. Every strategy shares the same terminal rule, which sells the remaining inventory after the 60th ordinary decision at a less favorable terminal spread. The cap applies to regular child orders; the terminal auction is the explicit exception.

The ten observable state coordinates are remaining inventory Q0-normalized; time-to-deadline; last realized market return; an exponentially smoothed return; observed volatility, spread and volume with fixed scaling; contemporaneous imbalance; deviation from a linear completion schedule; and previous executed multiplier. The policy does not observe the latent Markov regime, future market volume or future return. The shield is deterministic given this state and the contemporaneous volume.

3.2. Execution Cost and Reward

For a sell child order q_t, the simulator applies half the spread plus concave market impact. Its pricing rule is

$$P_{exec,t} = M_t [1 - \{0.5 s_t + h(0.42 s_t + 0.18 sigma_t) sqrt[(q_t/V_t)/0.05]\}/10^4] \quad (1)$$

Where M_t is the pre-trade mid-price, s_t and sigma_t are spread and return-volatility proxies in basis points, and h is an impact multiplier (h = 1 unless the sensitivity experiment says otherwise). The next mid-price receives the market tape's exogenous return less 9% of the computed temporary-impact concession; thus earlier trading changes later marked prices. The terminal leftover is liquidated using one-and-a-half times the observed terminal spread plus the same impact formula, with no per-interval participation cap. These coefficients are stylized and are not fitted to any exchange.

Episode implementation shortfall IS is the arrival-price basis-point loss on average proceeds. For sell orders a lower value is better; favorable exogenous movement may yield negative values. Training reward uses marked wealth change, rather than just immediate fill cost, together with observable risk and lateness charges:

$$IS = 10^4 [1 - (sum_t q_t P_{exec,t})/(Q0 M_0)] \quad (2)$$

$$r_t = 0.1 \{DeltaW_t(bps) - lambda_t sigma_t (x_{(t+1)}/Q0)^2 - u_t\} \quad (3)$$

For all learned policies, lambda_t = 0.12[1 + 1.8 max(sigma_t/5 - 1, 0) + 1.2 max(s_t/3 - 1, 0)]; u_t = 0.28 max(0, x_(t+1)/Q0 - (T-t-1)/T). The terminal wealth adjustment includes liquidation cost. The same reward is used in the three training variants; the learning comparison does not rely on separate reward engineering.

3.3. Latent Market Generator

The synthetic market follows a four-state Markov chain with a persistent regime-specific volatility, spread and mean volume. Imbalance is a clipped autoregressive series, and next-step exogenous return responds weakly to observable imbalance and independently sampled noise. Directional stress also contains a drift whose sign persists until a regime transition; its sign is hidden. Quoted spread, volume and volatility receive lognormal innovations. The path generator uses fixed per-path integer seeds; policies on the same held-out path receive identical exogenous shocks, although their own impact produces different subsequent mid-prices.

Regimes are not fitted to publicly traded instruments. The finite-state process isolates the extent to which a learned policy can use contemporaneous information without importing unobserved future market states. A pathwise run is reproducible from the released generator and its identifier.

Table I. Regime Generator Parameters (Synthetic)

Regime	sigma (bps)	Spread (bps)	Volume	Stay p
Calm	2.2	1.2	135	0.965
Normal	4.5	2.4	95	0.955
Volatile	9.0	6.5	46	0.945
Stress	11.5	9.0	58	0.930

4. Shielded Deep Q-Learning

4.1. Observable Action Shield

The rule layer maps an action proposal a to a permissible released action S(s, a). The shield implements the execution rule layer of Qin et al. [10]: its six execution multipliers, market-contingent urgency tests and explicit 20%/10% deadline floors. Because that framework defines the urgency conditions without fixing every numerical threshold, we set the adverse-imbalance cutoff to -0.35, the high-volatility cutoff to 8 bps, the schedule-lag tolerance to 13% and the favorable-flow cutoff to 0.5.

For sell execution, a multiplier floor of 1.5 is activated by strongly negative imbalance or inventory more than 13% above linear pace, and a floor of 2 is activated when observed volatility reaches 8 bps with nonpositive imbalance. At t >= 48 the floor is at least 1.5; at t >= 54 it is at least 2. Aggressive proposals can be reduced to at most 1 when imbalance is

above 0.5, spread exceeds 6 bps and at least 30% of the horizon remains. When urgency and waiting rules disagree, the urgency floor has priority. The final result is snapped to the nearest of the six multipliers; fill quantity still obeys the participation cap. The intervention audit records every proposed/released mismatch.

4.2. Shield-consistent Value Updates

A post-hoc shield does not alter the environment used for network fitting. Its training transitions use the proposed action a, but deployment generates the effective action S(s, a). The discrepancy is especially consequential near deadlines, where several proposals are mapped to one urgent child order. SC-DDQN instead samples the actual shielded transition while preserving the original proposal as the Q-function action label. The replay samples the transition and reward induced by the composite decision S(s, a), so training and deployment use the same proposal-to-execution semantics. The underlying simulator is Markov only when its hidden regime and other latent variables are included; the network receives a partial observation and learns an approximate value over that observation.

Let z denote the simulator's full latent state, o(z) its observation, F its transition kernel, and r its reward. Since S depends only on current observable inputs, it induces a new action-conditioned kernel F^S(z'|z, a) = F(z'|z, S(o(z), a)). At the full-state level, the corresponding Bellman relation is:

$$Q^S(s,a) = E[r(s,S(s,a)) + \gamma \max_b Q^S(s',b)] \quad (4)$$

In implementation Q is parameterized by the observed state o, rather than the unavailable latent z, so (4) is a full-state modeling identity and the learned network is its observation-based approximation. This distinction requires neither a larger model nor differentiation through discrete rules. Every replay tuple contains (s, proposal a, realized reward r, next state s', done), where the reward and next state are generated by S(s, a). The online Double Dueling Q network evaluates candidate proposals; a target network evaluates the selected next proposal in the induced environment. A network with two 64-unit ReLU layers predicts a scalar V(s) and six action advantages A(s, a), whose centered sum forms Q(s, a). An AdamW optimizer, smooth-L1 temporal-difference loss, uniform replay and periodic target refresh implement the update. The architecture and hyperparameters are identical for raw DDQN, SC-DDQN and BSC-DDQN.

Action aliasing means several proposals can produce identical trades. It does not invalidate (4): they are separate labels for the same or similar induced transition and can have the same optimal value. Unlike a post-hoc correction, this model learns that equivalence from data. The equality specifies the full-state control problem; it is not a convergence theorem or an assertion that the observation vector is a sufficient Markov state.

4.3. Volatility-dependent Inventory Budget

As an additional risk-control variant, BSC-DDQN retains SC-DDQN's shield-consistent training but adds a one-step risk-envelope check before release. A time-indexed upper inventory target decreases with the horizon and becomes tighter as contemporaneous volatility rises. Using sigma_ref = 5 bps and a fixed 6% schedule slack, the post-action inventory target is:

$$B_t = 1.06\, Q_0\, [(T-t-1)/T]\, \min\{1, \sqrt{5/\max(5,\sigma_t)}\} \quad (5)$$

The shield computes $d_t = \max(0, x_t - B_t)$, truncates this required sale by the contemporaneous participation capacity $0.2V_t$, and chooses the smallest discrete proposal multiplier whose clipped quantity reaches that feasible sale target. The budget-induced floor is combined with the market-urgency floors. An infeasibility flag is raised whenever the untruncated d_t exceeds $0.2V_t$; such a state cannot satisfy the envelope through any regular child order. This is a risk-budget aiming mechanism with an explicit feasibility monitor, not a claim of guaranteed bound satisfaction in a volume-limited market.

The envelope's motivation is an upper limit on the volatility-scaled exposure $x^2\sigma$, normalized to a declining time-dependent inventory schedule. Equation (5) makes the permitted inventory smaller in a high-volatility regime, without using the unobserved regime label. The two shield-consistent variants provide a direct ablation: SC-DDQN retains observable urgency rules, while BSC-DDQN also enforces the exposure envelope. This optional budget is expected to reduce exposure but may increase temporary impact, so its usefulness is evaluated as a mean-cost/tail-risk trade-off rather than presumed superiority.

5. Experimental Methodology

5.1. Train/test Separation and Baselines

Three independent training seeds (7, 19 and 31) fit each network for 15,000 environment interactions. For each seed, the held-out evaluation consists of 240 newly generated episode paths. No test episode is used for optimizer updates, and no model selection or threshold search is performed on the test set. There are 720 distinct paired exogenous paths and 60 sequential trading decisions on each. For each seed, all three training variants use common random-number streams and the same market-tape schedule; the realized actions, rewards and resulting policies nevertheless differ. This matched-training construction reduces training-tape noise in the ablation.

Baselines include TWAP (current equal-time slice), POV (the nearest action to 5% of observed interval volume), a front-loaded multiplier rule, and a fixed TWAP proposal passed through the observable shield. Raw DDQN is trained and evaluated without rules. DDQN-postshield fits the raw DDQN checkpoint and applies the observable shield only at evaluation. SC-DDQN fits with that same shield activated and uses it during evaluation. BSC-DDQN adds the volatility exposure envelope and trains with that augmented shield. All methods share the action alphabet, participation cap, terminal liquidation rule and market generator. DDQN-postshield corresponds to the network-plus-shield configuration of [10], in which learned proposals are corrected by the execution rules after training; SC-DDQN differs from it only in learning inside the same rules, so their paired difference measures the value of training under that intervention mechanism.

5.2. Optimization, Evaluation and Auditing

The network has a 10-64-64 shared encoder, a one-dimensional value head and six-dimensional advantage head. Training uses discount 0.99, AdamW learning rate 4e-4, weight decay 1e-5, a capacity-25,000 uniform replay buffer, batch size 64, a learning update every four interactions after the first 500 and target copying every 360 interactions. Exploration is epsilon-greedy with a 0.95-to-0.07 schedule. A 5.0 gradient-norm cap stabilizes backpropagation. Each

algorithm therefore receives the same interaction budget, optimization configuration and model class.

Primary outcome is mean arrival-price IS (bps). Secondary outcomes are median, 95th percentile and CVaR95, the mean of losses at or above the sample 95th percentile. CVaR is used as a descriptive tail-risk measure in the sense of [27]; it is not used as a training objective. We also compute mean terminal fraction, shield proposal-correction rate, mean volatility-weighted squared remaining inventory, and the fraction of intervals in which the desired exposure envelope was infeasible given the hard participation cap.

For a baseline B and a proposed policy A on the same path i, d_i = IS_B,i - IS_A,i. The signed difference is positive when A is cheaper. We report mean(d), a two-sided paired t-test, and a 95% Student interval for its mean. This path-level inference is conditional on the fitted networks, episode generator and simulator coefficients; sequential steps inside one episode are not independent samples. We also calculate a second, more conservative Student interval across the three fit-level mean differences (2 degrees of freedom) and provide a two-stage bootstrap diagnostic. Because three seeds are too few for dependable cluster asymptotics, neither supplies a population-level guarantee. Saved path records permit independent recalculation.

Table II. Training And Evaluation Configuration

Item	Setting
Training seeds	7, 19, 31
Train interactions	15,000 per model/seed
Held-out paths	240 per seed; 720 paired
Time steps	60 per path
Network	10-64-64; dueling 1+6
Replay / optimizer	25,000 / AdamW, 4e-4
Discount / batch	0.99 / 64
Participation cap	0.20 of observed volume

5.3. Coefficient Shift and Provenance

We freeze all trained networks and evaluate the same policy family with temporary-impact coefficient multipliers h of 0.75, 1.00 and 1.25. The sensitivity run uses an independent 100-path group per seed and multiplier, producing 300 paths per scale. Within each scale, path identifiers are paired across policies; paths reused between scales keep the same exogenous tape so that changes isolate the cost-rule perturbation. The training parameters and exposure-envelope thresholds remain unchanged.

The dataset for this study is fully synthetic. Market paths are generated by `tape_for` from recorded integer seeds, and all counterfactual fills arise from (1). The `episodes.csv`, `impact_sensitivity.csv`, `config.json`, seed-specific PyTorch checkpoint files and training diagnostics are saved with the code. There is no implied access to proprietary orders, matching-engine queues or live-market fills.

6. Results

6.1. Mean Shortfall and Tail Loss

Table III. Held-Out Implementation Shortfall (Bps; N = 720)

Policy	Mean	Median	P95	CVaR95
TWAP	18.67	15.79	78.56	108.47
POV	27.81	21.05	122.93	173.16
Front	17.73	13.65	74.95	100.12
Shield-rule	17.71	15.11	74.51	98.67
DDQN-raw	16.37	11.69	69.08	97.47
DDQN-postshield	15.88	11.36	67.06	94.50
SC-DDQN	15.05	10.98	69.33	96.85
BSC-DDQN	16.24	11.36	69.15	93.75

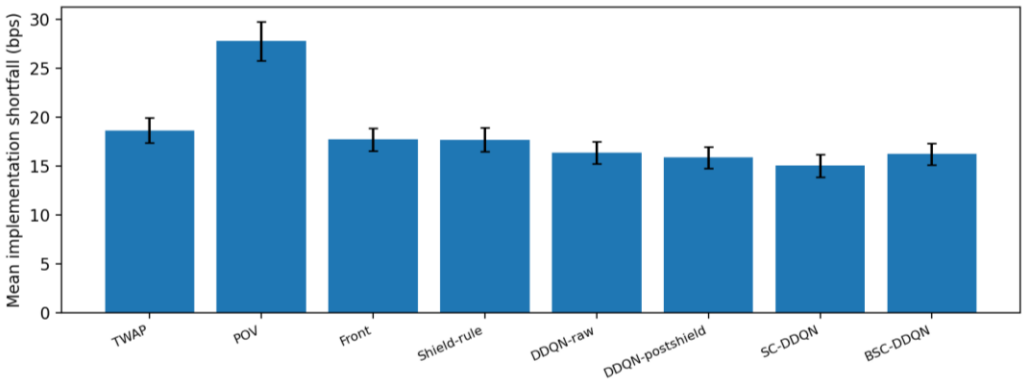


Figure 1. Mean held-out implementation shortfall; error bars represent path-level standard errors.

On 720 held-out synthetic paths, shield-consistent SC-DDQN obtains 15.05 bps mean implementation shortfall. TWAP obtains 18.67 bps; raw DDQN obtains 16.37 bps; and a raw network coupled to the same shield only at deployment obtains 15.88 bps. The optional inventory-budget variant BSC-DDQN obtains 16.24 bps. The non-neural Shield-rule baseline averages 17.71 bps and is included to measure how much learned control adds to a purely deterministic intervention policy. The rule layer of [10] lowers cost in every configuration: applied to a fixed TWAP proposal it reduces shortfall from 18.67 to 17.71 bps, applied after training it reduces raw DDQN from 16.37 to 15.88 bps, and used inside training it yields the lowest mean cost of 15.05 bps. Table III includes medians and the upper 5% tail rather than equating minimum average cost with minimum risk.

SC-DDQN's CVaR95 is 96.85 bps, compared with 94.50 bps for post-hoc shielding and 93.75 bps for budgeted shielding. Thus the shield-consistent network improves the average outcome while showing a numerically larger upper-tail loss in this experiment; the added risk envelope moves the observed tail estimate in the favorable direction but worsens mean execution cost. Every CVaR uses the empirical upper 5% of that policy's 720 observations. Only about 36 paths determine each tail estimate, so small differences are treated descriptively, not as established tail superiority.

6.2. Matched-path Attribution

Table IV. Paired Shortfall Reduction Relative to Sc-Ddqn

Comparator	Delta bps	95% CI	p
TWAP	3.62	[3.07,4.17]	<.001
DDQN-raw	1.32	[1.01,1.64]	<.001
DDQN-postshield	0.83	[0.56,1.10]	<.001
BSC-DDQN	1.19	[0.88,1.50]	<.001
Shield-rule	2.66	[2.25,3.06]	<.001

The paired improvement of SC-DDQN over TWAP is 3.62 bps (95% CI [3.07, 4.17]; p = 1.82e-34); over unshielded DDQN it is 1.32 bps (95% CI [1.01, 1.64]; p = 4.11e-16). The key matched-network comparison is against DDQN-postshield, which applies the identical observable market-urgency shield only after training: 0.83 bps (95% CI [0.56, 1.10]; p = 1.59e-09). Thus the observed mean simulated execution cost favors shield-transformed training in all three fitted seed groups, without increasing model size. A comparison against a fixed equal-slice proposal with the same rules is 2.66 bps (95% CI [2.25, 3.06]; p = 8.29e-35).

The optional budget version is not an automatic improvement: the paired difference between BSC-DDQN and SC-DDQN is 1.19 bps (95% CI [0.88, 1.50]; p = 1.82e-13), meaning the budgeted extension has the higher mean cost. This is informative because the additional exposure envelope enforces urgency at moments when impact is costly. The shielded-training effect should therefore be distinguished from the cost of imposing a tighter inventory risk tolerance.

Pathwise pairing removes exogenous price noise but does not eliminate fitting variability. Model hyperparameters and risk-envelope thresholds were held fixed, and no checkpoint selection used held-out shortfall.

6.3. Training-Seed Uncertainty and Market Regimes

The SC-DDQN advantage over post-hoc shielding is 1.45, 0.47 and 0.57 bps for training seeds 7, 19 and 31. Aggregation by fit gives a 95% Student interval [-0.50, 2.16] bps (n=3 fits, df=2, p=0.115). In contrast, the path-level interval in Table IV is conditional on these fitted networks. The fit-level interval crosses zero: the dataset establishes consistent direction across three fits but not a conventionally significant generalization across independently retrained networks. The accompanying audit file also records the two-stage bootstrap as a descriptive check; its intervals should not be treated as reliable with only three fit clusters.

Table V. Post-Hoc Minus Sc-Ddqn, By Independent Training Seed (Bps)

Train seed	Paired mean gain
7	1.45
19	0.47
31	0.57
Across fits	0.83

Table VI. Regime-Conditioned Post-Hoc Minus Sc-Ddqn (Bps)

Dominant regime	Paths	Mean gain
calm	258	0.97
normal	201	0.60
volatile	150	0.02
stress	111	2.01

The dominant-regime decomposition shows the largest mean difference in directional stress, with almost no difference for volatile-thin dominant episodes. These are descriptive subgroups formed from synthetic latent labels for analysis only; the trading network does not receive regime labels. Small subgroup sizes and seed heterogeneity preclude a claim of uniformly superior execution in every state.

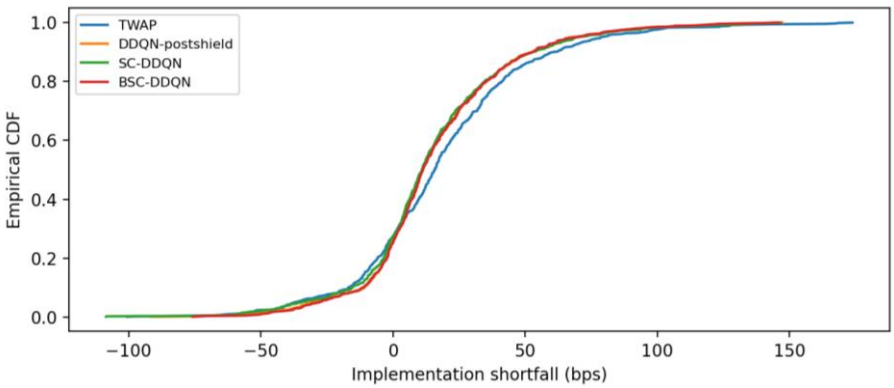


Figure 2. Empirical distribution of simulated execution shortfall on paired held-out paths.

6.4. Completion, Intervention and Inventory Exposure

Table VII. Completion Pathway and Shield Diagnostics

Policy	Terminal qty	Changes	Exposure
TWAP	0.1%	0.0%	2.07
Shield-rule	0.0%	43.4%	1.89
DDQN-raw	1.3%	0.0%	1.45
DDQN-postshield	0.0%	23.2%	1.43
SC-DDQN	0.0%	30.0%	1.55
BSC-DDQN	0.0%	29.2%	1.41

The common terminal clean-up rule completes every parent order, but leftover inventory pays a terminal spread and impact surcharge. Mean terminal fraction is 0.03% for SC-DDQN, 0.00% for BSC-DDQN and 0.00% for the post-hoc baseline. The share of neural proposals revised by the shield is 30.0% for SC-DDQN, 29.2% for BSC-DDQN, and 23.2% for naive post-hoc shielding.

For BSC-DDQN the desired one-step exposure envelope is infeasible in 33.8% of decision intervals because required inventory reduction exceeds the contemporaneous participation capacity. This exposes a material operational difficulty with stringent risk bounds. Its mean volatility-

weighted inventory exposure is 1.41, compared with 1.55 for SC-DDQN and 2.07 for TWAP. The risk-budget variant therefore accepts increased average impact/slippage to carry less inventory in volatile intervals. Figure 3 shows the resulting pathwise liquidation profiles.

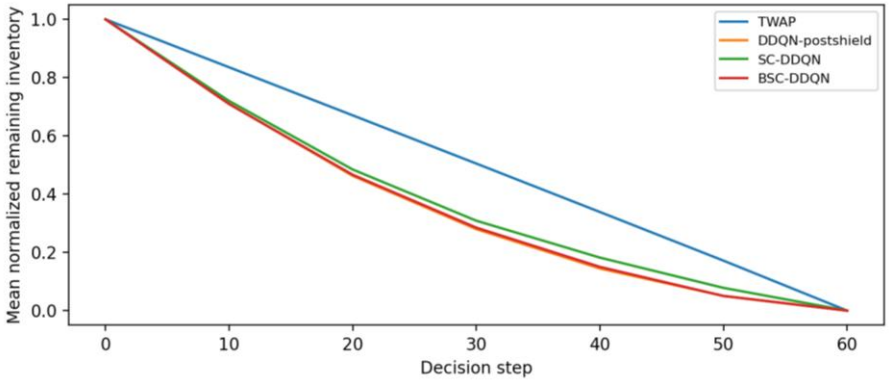


Figure 3. Average fraction of inventory remaining at selected decision times.

6.5. Impact-model Sensitivity

Table VIII summarizes an out-of-training-distribution cost perturbation. All four policies are run on 300 fresh tapes at each coefficient multiplier; networks and shield thresholds are held fixed. The impact effect is approximately monotonic for the evaluated methods, but mean differences are not expected to scale linearly because more aggressive liquidation incurs larger immediate impact while reducing time exposed to adverse price movement. Across the three settings, this analysis is a modest robustness probe rather than a comprehensive simulator-misspecification study. It does not vary order-book queue priority, fill probability, fees, strategic response or latency.

Table VIII. Frozen-Policy Shortfall Under Impact Scaling (Bps)

Scale	TWAP	Post	SC	BSC
0.75	14.43	12.55	11.72	12.82
1.00	18.90	16.42	15.47	16.73
1.25	23.35	20.29	19.22	20.64

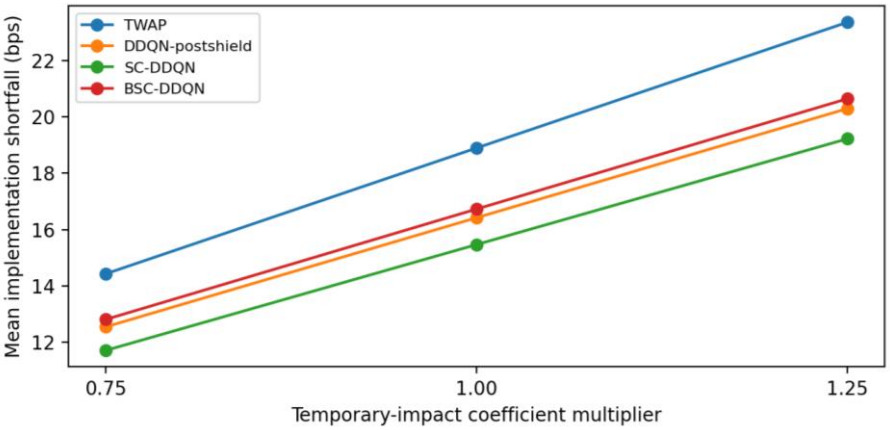


Figure 4. Shortfall sensitivity to temporary-impact coefficient scaling (independent paths).

7. Interpretation and Limitations

Shield consistency and inventory-budgeting address distinct problems. In the first, the learner receives transitions that match the actions its deployed shield actually releases. This removes an avoidable source of training-deployment discrepancy but may offer only a small gain when the shield dominates decisions or a deterministic schedule already performs well. In the second, the budget explicitly contracts the remaining-inventory allowance under high current volatility, trading execution urgency against nonlinear immediate impact. Neither modification permits the network to see future market information, and neither alters the parent's required completion quantity.

The full-state control identity in (4) explains how a deterministic shield can be embedded in value learning; because the agent observes only part of the simulated Markov state, its observation-conditioned value remains an approximation. Nevertheless, the envelope in (5) is not a hard chance constraint with guaranteed feasibility. When the market-volume cap is active, the shield selects the most aggressive available action and records the violation. A credible production implementation must also specify a reject/retry mechanism, exchange-side limits and an emergency fallback for stale observations. Terminal forced liquidation is convenient for a controlled simulation, but may be materially optimistic under extreme real-world illiquidity.

The simulation excludes limit-order queue position, partial and hidden fills, venue routing, rebates, transaction fees, latency and market reactions to the strategy. Its regime-specific parameters are chosen rather than estimated from a chronological exchange dataset. The findings therefore concern internal algorithmic behavior under four persistent regimes and a single sell-order size. They also show that the tighter risk envelope cannot always be satisfied under a real-time participation limit; numerical CVaR improvement alone cannot turn an infeasible constraint into a guarantee. They cannot establish that the same absolute basis-point costs,

optimal thresholds or relative ranking will hold on real order books. The next validation stage should calibrate the simulator on a legally accessible venue dataset, preserve a chronological held-out test interval, replay the same parent orders across policies with explicit queue assumptions, and finally assess shadow orders before risking capital.

A second limitation is experimental power over model-fitting randomness. Three trained checkpoints per method and 15,000 interactions per checkpoint are a transparent computational budget, not an exhaustive search. Our paired 720 paths make path noise easier to control; they do not substitute for an independent sample of many training configurations. Risk-envelope hyperparameters were fixed a priori in this study. An external replication should prespecify thresholds and evaluation assets, vary the parent-to-volume ratio, and estimate uncertainty clustered across assets and dates. These steps would test whether shield consistency remains beneficial after venue mechanics replace synthetic fills.

Finally, the experiment isolates the action shield of the earlier execution framework from its mixture-of-experts value network. This choice focuses the ablation on a measurable question: whether a value function estimated under the deployed action transformation behaves differently from a value function estimated without it.

8. Conclusion

We presented shield-consistent Double Dueling Q-learning and an independently auditable inventory-exposure extension for risk-constrained execution. The principal innovation places the deployed deterministic action transformation inside the transition data used for value learning, retaining a compact discrete controller. On 720 matched synthetic held-out paths, SC-DDQN averages 15.05 bps shortfall versus 15.88 bps for the otherwise comparable model shielded only at deployment; the conditional path-level paired reduction is 0.83 bps, while a fit-level interval across three independent seeds contains zero. A volatility-dependent inventory budget reduces measured exposure but raises mean shortfall to 16.24 bps, revealing a meaningful execution-risk trade-off rather than a free improvement. These reproducible tests identify a promising mean-cost effect in this simulator, with limited model-fit statistical power. Real-venue validity remains contingent on calibrated order books, realistic fills and external chronological evaluation.

References

[1] Bertsimas, D., & Lo, A. W. (1998). Optimal control of execution costs. Journal of Financial Markets, 1(1), 1–50. https://doi.org/10.1016/S1386-4181(97)00012-8

[2] Almgren, R., & Chriss, N. (2001). Optimal execution of portfolio transactions. Journal of Risk, 3(2), 5–39. https://doi.org/10.21314/JOR.2001.041

[3] Gatheral, J. (2010). No-dynamic-arbitrage and market impact. Quantitative Finance, 10(7), 749–759. https://doi.org/10.1080/14697680903373692

[4] Cont, R., Kukanov, A., & Stoikov, S. (2014). The price impact of order book events. Journal of Financial Econometrics, 12(1), 47–88. https://doi.org/10.1093/jjfinec/nbt003

[5] Nevmyvaka, Y., Feng, Y., & Kearns, M. (2006). Reinforcement learning for optimized trade execution. Proceedings of the 23rd International Conference on Machine Learning, 673–680. https://doi.org/10.1145/1143844.1143929

[6] Ning, B., Lin, F. H. T., & Jaimungal, S. (2021). Double Deep Q-Learning for optimal execution. Applied Mathematical Finance, 28(4), 361–380. https://doi.org/10.1080/1350486X.2022.2077783

[7] Lin, S., & Beling, P. A. (2020). An end-to-end optimal trade execution framework based on proximal policy optimization. Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, 4548–4554. https://doi.org/10.24963/ijcai.2020/627

[8] Fang, Y., Yu, L., Liu, D., & Zou, B. (2021). Universal trading for order execution with oracle policy distillation. Proceedings of the AAAI Conference on Artificial Intelligence, 35(1), 107–115. https://doi.org/10.1609/aaai.v35i1.16083

[9] Alshiekh, M., Bloem, R., Ehlers, R., Könighofer, B., Niekum, S., & Topcu, U. (2018). Safe reinforcement learning via shielding. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1), 2669–2678. https://doi.org/10.1609/aaai.v32i1.11797

[10] Qin, Y., Zou, J., Guo, Z., & Zi, B. (2026). An adaptive deep reinforcement learning framework for intelligent transaction execution in internet finance. World Scientific Research Journal, 12(10), 16–31. https://wsr-j.org/index.php/ojs/article/view/152

[11] Almgren, R. F. (2003). Optimal execution with nonlinear impact functions and trading-enhanced risk. Applied Mathematical Finance, 10(1), 1–18. https://doi.org/10.1080/135048602100056

[12] Almgren, R., & Lorenz, J. (2011). Mean-variance optimal adaptive execution. Applied Mathematical Finance, 18(5), 395–422. https://doi.org/10.1080/1350486X.2011.560707

[13] Gabaix, X., Gopikrishnan, P., Plerou, V., & Stanley, H. E. (2006). Institutional investors and stock market volatility. The Quarterly Journal of Economics, 121(2), 461–504. https://doi.org/10.1162/qjec.2006.121.2.461

[14] Toth, B., Lillo, F., Farmer, J. D., & Mantegna, R. N. (2011). Anomalous price impact and the critical nature of liquidity in financial markets. Physical Review X, 1, Article 021006. https://doi.org/10.1103/PhysRevX.1.021006

[15] Gould, M. D., Porter, M. A., Williams, S., McDonald, M., Fenn, D. J., & Howison, S. D. (2013). Limit order books. Quantitative Finance, 13(11), 1709–1742. https://doi.org/10.1080/14697688.2013.803148

[16] Sirignano, J., & Cont, R. (2019). Universal features of price formation in financial markets: perspectives from deep learning. Quantitative Finance, 19(9), 1449–1459. https://doi.org/10.1080/14697688.2019.1622295

[17] Makarov, I., & Schoar, A. (2020). Trading and arbitrage in cryptocurrency markets. Journal of Financial Economics, 135(2), 293–319. https://doi.org/10.1016/j.jfineco.2019.07.001

[18] Macri, A., & Lillo, F. (2024). Reinforcement learning for optimal execution when liquidity is time-varying. Applied Mathematical Finance, 31(5), 312–342. https://doi.org/10.1080/1350486X.2025.2490157

[19] 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, 529–533. https://doi.org/10.1038/nature14236

[20] van Hasselt, H., Guez, A., & Silver, D. (2016). Deep reinforcement learning with Double Q-learning. Proceedings of the AAAI Conference on Artificial Intelligence, 30(1), 2094–2100. https://doi.org/10.1609/aaai.v30i1.10295

[21] Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., & Freitas, N. (2016). Dueling network architectures for deep reinforcement learning. Proceedings of the 33rd International Conference on Machine Learning, 48, 1995–2003. https://proceedings.mlr.press/v48/wangf16.html

[22] Schaul, T., Quan, J., Antonoglou, I., & Silver, D. (2016). Prioritized experience replay. International Conference on Learning Representations. https://arxiv.org/abs/1511.05952

[23] Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., & Silver, D. (2018). Rainbow: Combining improvements in deep reinforcement learning. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1), 3215–3222. https://doi.org/10.1609/aaai.v32i1.11796

[24] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. International Conference on Learning Representations. https://arxiv.org/abs/1412.6980

[25] Ntakaris, A., Magris, M., Kanniainen, J., Gabbouj, M., & Iosifidis, A. (2018). Benchmark dataset for mid-price forecasting of limit order book data with machine learning methods. Journal of Forecasting, 37(8), 852–866. https://doi.org/10.1002/for.2543

[26] Zhang, Z., Zohren, S., & Roberts, S. (2019). DeepLOB: Deep convolutional neural networks for limit order books. IEEE Transactions on Signal Processing, 67(11), 3001–3012. https://doi.org/10.1109/TSP.2019.2907260

[27] Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. Journal of Risk, 2(3), 21–41. https://doi.org/10.21314/JOR.2000.038