Intelligent Portfolio Rebalancing with Reinforcement Learning and Dynamic Resource Allocation Under Market Uncertainty

Karol Dziadowicz
Department of Computer Science, Colorado State University, Fort Collins, CO, USA.
karoldziadowicz862@colostate.edu

Margot Benoit
School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA.
margot1972@oregonstate.edu

Natasza Cich
Department of Computer Science and Engineering, University of Nevada, Reno, Reno, NV, USA.
hellonatasza@unr.edu

Jeremi Lupina
Department of Computer Science, University of Houston, Houston, TX, USA.
contactjeremi@uh.edu

Abstract

Portfolio rebalancing is a sequential decision problem in which the benefit of moving toward an estimated target must be weighed against trading costs, the risk of breaching a loss limit, and the possibility that the estimate is itself wrong. Calendar rules, threshold rules, and most learning-based portfolio agents treat these considerations in sequence: allocation is optimized first, filtered for turnover second, and examined for risk third. This paper develops a constrained reinforcement learning framework in which friction, risk, and estimation uncertainty enter the decision problem directly. Three components are coupled. The trade vector rather than the target weight is the action, so the self-financing condition and the cost function are structural rather than penalized. An analytically derived no-trade band, obtained from a first-order condition that equates marginal risk reduction to marginal trading cost, is embedded as a feasible-action constraint that the policy cannot violate. A dynamic resource allocation layer maps a scalar uncertainty index and a regime posterior to a risk budget, a trading propensity, and a re-optimization frequency, so that capital and computational attention contract when forecasts are least reliable. The loss limit is imposed as a conditional value-at-risk constraint and handled by Lagrangian dual ascent with ensemble-based tail estimation. The paper derives comparative statics for the band, specifies the conditions under which a learned policy can improve on the analytic rule, and sets out the failure modes and diagnostic protocol that such a system must satisfy before it is allowed to trade. The contribution is a coherent decision architecture whose components are individually interpretable and jointly constrained, rather than a black-box allocation model.

Keywords

constrained reinforcement learning; portfolio rebalancing; transaction costs; conditional value-at-risk; dynamic risk budgeting; nonstationary markets

1. Introduction

Rebalancing is the point at which portfolio theory meets market friction. Mean-variance analysis specifies a target allocation as a function of expected returns and a covariance matrix [1], but a target is not a decision. The decision is whether to trade toward the target now, partially, or not at all, and the answer depends on costs that are incurred with certainty against benefits that are estimated with error. The gap between a paper portfolio and an implemented one was formalized as implementation shortfall, the difference between the return of a decision and the return actually captured after execution [2]. Once that gap is admitted, the frequency and aggressiveness of trading become first-class control variables rather than implementation details. Execution research subsequently characterized the trade-off between market impact and timing risk for a given order [3], but a rebalancing policy faces a prior question: which orders should exist at all.

Empirical work on allocation rules sharpens the puzzle. Simple equal weighting is difficult to beat out of sample in a range of datasets, a result usually attributed to estimation error in sample-efficient optimizers rather than to the absence of exploitable structure [4]. If parameter uncertainty is large enough to erase the apparent advantage of optimized weights, then a policy that trades frequently to track noisy estimates is doubly penalized: it pays costs to implement noise. The relevant object is therefore not the optimal weight vector at a point in time but a mapping from the joint state of estimates, positions, and frictions to a trade. That mapping must decide how far to move, how fast, and how much of the available risk budget to commit.

Reinforcement learning offers a natural formalism for such a mapping because it optimizes a sequence of decisions against a delayed objective, which is exactly the structure of rebalancing under costs. Direct reinforcement methods for trading established that a policy can be trained on a risk-adjusted performance criterion rather than on forecast accuracy [5], and subsequent deep formulations extended the approach to multi-asset portfolios with convolutional feature extractors and continuous action spaces [7], [8]. Two limitations recur across this literature. First, transaction costs are usually represented as a fixed proportional deduction inside the reward, which makes the learned policy sensitive to the assumed cost level and provides no structural guarantee that a trade is economically justified. Second, risk limits, when they appear at all, appear as penalty terms whose weighting is tuned by hand, so that the constraint can be traded away against return during training and violated during deployment.

The same structural problem appears outside finance wherever an inference module is coupled to a decision module that must act before the inference is confirmed. In automated structural timber jointing, for instance, real-time defect recognition is coupled to an optimized decision layer that selects jointing actions under geometric and material uncertainty [6]. The decision layer there cannot wait for perfect recognition; it must convert a noisy, high-frequency inference into an action whose cost of being wrong is asymmetric. Portfolio rebalancing has the same logical form with different primitives: the recognition module produces return and covariance estimates, the decision layer produces trades, and the

asymmetry comes from the fact that trading costs are certain while the expected gain from trading is estimated.

This paper addresses that gap with a constrained learning architecture that keeps the decision layer interpretable at the level of individual trades. The central contribution is a coupling of three mechanisms that are usually studied separately. The first is an action representation in which the trade vector, not the target weight, is the decision variable, so that budget balance and trading costs are embedded in the transition and reward rather than imposed afterwards. The second is an analytic no-trade band derived from a first-order condition and imposed as a hard feasible-action constraint, which yields a policy whose trades are justified by the same marginal calculus that classical transaction-cost theory recommends, but with state-dependent parameters. The third is a dynamic resource allocation layer that converts an uncertainty index and a regime posterior into a scalar risk budget, a trading propensity, and a re-optimization frequency, thereby allocating capital and computation across decision modules according to how much confidence the current forecast deserves. The risk limit is imposed as a conditional value-at-risk constraint and handled through Lagrangian dual ascent.

The argument proceeds as follows. Section 2 formulates the portfolio state, the friction model, the uncertainty representation, and the constrained objective. Section 3 develops the learning framework, including observation design, policy parameterization, reward construction, the resource allocation layer, and the training procedure. Section 4 derives the no-trade band, establishes its comparative statics, and delineates the conditions under which a learned policy can legitimately outperform it. Section 5 examines failure modes and specifies a diagnostic protocol. Section 6 discusses implementation conditions and boundary conditions, and Section 7 concludes.

2. Problem Formulation and Sources of Uncertainty

2.1 Portfolio state, drift, and trade representation

Consider a portfolio of n risky assets held over discrete decision epochs t = 0, 1, ..., T−1. Let w_t ∈ R^n denote the vector of portfolio weights at the start of epoch t, r_t ∈ R^n the vector of asset returns realized over the epoch, and a_t ∈ R^n the vector of trade weights executed at the start of the epoch, expressed as a fraction of current portfolio value. Weights evolve mechanically under the combined effect of price drift and trading:

$$w_\{t+1\} = (w_t \odot (1 + r_t) + a_t) / (1 + w_t{}^{T} r_t),\ 1^{T} a_t = 0$$

Here ⊙ denotes the elementwise product, 1 is a vector of ones, and the denominator normalizes by gross portfolio return so that weights remain a partition of unity. The self-financing condition $1^{T}a_t = 0$ states that trades are funded entirely from the existing portfolio, with no external cash injection. Long-only mandates add the constraints $w_\{t+1\} \geq 0$, and position limits add $w_\{t+1\} \leq \bar{u}$ componentwise.

Treating the trade vector as the action rather than the post-trade weight is consequential. If the action were the target weight, the mapping from action to executed trade would be ambiguous, and a policy could obtain credit for a favorable target without ever paying for the trade that would have implemented it. With the trade as the action, the cost function consumes exactly the quantity that produces portfolio change, and the self-financing condition is a linear constraint on the action rather than an accounting identity that must be restored after the fact. The drift term in the numerator also matters: a passive portfolio's weights move during the

epoch, so the no-trade position is not the previous target but the drifted portfolio. Relative to that drifted position, a meaningful fraction of apparent rebalancing demand is simply mean reversion of weights that price movements have displaced.

2.2 Frictions and the cost of trading

Trading costs comprise explicit components, such as commissions and fees, and implicit components, such as bid-ask spread and market impact. A tractable representation that preserves the essential shape of the friction is a sum of a linear per-unit term and a convex impact term:

$$c_t = \sum_i \left(\kappa_i |a_{i,t}| + \left(\frac{\lambda_i}{2} \right) a_{i,t}^2 \right)$$

The coefficient κ_i absorbs half-spread and proportional fees and represents a cost that is paid on any nonzero trade regardless of size; the coefficient λ_i governs the marginal cost of size and reflects the depth of the market for asset i relative to the size of the portfolio. Both coefficients are measured in units of portfolio return, so c_t enters the objective on the same scale as w_{t+1}^T r_{t+1}. The absolute value is what makes the problem genuinely one of rebalancing rather than smoothing: the marginal cost of a trade is κ_i in magnitude at zero, so small trades face a strictly positive marginal cost that does not shrink with size. This feature, rather than the quadratic term, is what generates a region of inactivity in the optimal policy.

The separable form is a deliberate simplification. Cross-impact, tiered fee schedules, and inventory-dependent liquidity would make c_t a function of the whole vector a_t rather than a sum over components. That generalization is where a learned policy has a legitimate structural advantage over a closed-form rule, and it is revisited in Section 4.3. Even in the separable case, the cost coefficients are not known with precision; they must be estimated from execution data or, in the absence of such data, taken from conservative priors. The consequences of cost-model error are examined in Section 5.1 with respect to the specific pathology of a policy learning to exploit an underestimated impact coefficient.

2.3 Predictive uncertainty, regime evidence, and covariance conditioning

The rebalancing decision depends on a target allocation, and the target depends on estimates. Three distinct sources of uncertainty matter, and they are not interchangeable. Estimation uncertainty concerns the sampling error in expected returns and covariances given a fixed data-generating process. Regime uncertainty concerns the possibility that the process itself has changed, so that the relevant sample is not the full history but a subset identifiable only through latent state. Model uncertainty concerns the possibility that the forecast function is misspecified.

Regime structure is captured by a hidden state variable whose evolution follows a Markov chain, so that the observation density is a mixture indexed by the current state [11]. Asset allocation under regime shifts shows that the persistence and volatility of regimes materially alter optimal allocations relative to a single-state benchmark [12], which implies that a rebalancing policy conditioned only on recent realized moments will misjudge the durability of current conditions. The observation supplied to the policy therefore includes a posterior over regimes rather than only a point estimate, so that the decision can respond to the possibility of transition rather than to its past frequency alone.

Covariance conditioning is a separate and equally basic problem. Sample covariance matrices in moderate dimension are ill-conditioned and their inverses amplify estimation error.

Shrinkage toward a structured target reduces this error and improves out-of-sample behavior [13]. The framework adopts a shrinkage estimator, with the shrinkage intensity treated as a state variable rather than a constant, because the appropriate amount of shrinkage depends on the ratio of sample length to dimension and on the stability of the regime. In addition, predictive uncertainty is estimated by an ensemble: several forecasters with different initializations and bootstrap samples produce a distribution over the next-period return vector, and the disagreement across ensemble members supplies a direct, if imperfect, signal of epistemic uncertainty [14]. The scalar uncertainty index û_t used by the resource allocation layer aggregates normalized ensemble dispersion, forecast error variance implied by the covariance estimate, and a conditioning diagnostic for the covariance matrix. The aggregation weights are hyperparameters, not estimated quantities, and the index is therefore a design choice rather than a calibrated probability.

The decision problem is partially observable in the technical sense: the regime state, the true covariance, and the true cost coefficients are not observed, and the observation contains only noisy indicators of each [10]. A policy that treats the observation as a sufficient statistic without memory will behave myopically at exactly those moments when the regime posterior is shifting. This motivates a recurrent or windowed observation design in Section 3.1 rather than a memoryless map from current moments to trades.

2.4 Risk objective and constrained formulation

Risk control is imposed as a constraint rather than as a penalty because the two are not equivalent when the constraint is meant to bind. Coherent risk measures satisfy monotonicity, translation invariance, positive homogeneity, and subadditivity, and conditional value-at-risk is the canonical coherent measure that is also computationally tractable [15]. For a loss random variable L, defined here as the negative of the portfolio return over the evaluation horizon, conditional value-at-risk at level α admits a convex representation:

$$CVaR_\alpha(L) \;=\; min_\nu \{\, \nu \;+\; (1-\alpha)\text{^}\{-1\}\, E[(L \;-\; \nu)\text{^}+\,]\,\}$$

where (·)^+ denotes the positive part and ν is the value-at-risk threshold that attains the minimum [16]. The representation is useful because it converts a tail expectation into an expectation of a convex piecewise-linear function, which can be estimated from samples or from a distributional critic without explicit integration over the tail. The level α governs which part of the loss distribution is controlled: larger α places the constraint deeper into the tail and increases the sample requirement for stable estimation.

Combining the return objective with the risk constraint yields a constrained Markov decision process [17]:

$$max_\pi\; E_\pi[\textstyle\sum_\{t=0\}\text{^}\{T-1\}\, \gamma\text{^}t\; r_t\text{^}net]\;\; s.t.\;\; CVaR_\alpha(L_\pi) \;\le\; L_max$$

where π denotes the policy, $\gamma \in (0,1)$ is a discount factor, r_t^net is the per-period net reward defined in Section 3.3, L_π is the loss under policy π over the constraint horizon, and L_max is the loss budget. The horizon over which the constraint is evaluated need not coincide with the decision horizon; a common choice is to evaluate the constraint over a multi-period window while optimizing decisions at a finer resolution, because a single-period loss constraint would be violated by the very diversification that reduces multi-period risk. Formally, the constraint is on the distribution induced by the policy, not on any realized path, so satisfaction must be assessed in expectation over the state distribution that the policy itself

generates. That self-referential character is the reason constraint handling in Section 3.5 is treated as a dual optimization rather than as a projection applied after the fact.

3. The Uncertainty-Gated Rebalancing Framework

3.1 Decision process and observation design

The rebalancing problem is cast as a partially observable Markov decision process with observation o_t, action a_t, and reward r_t^net. The observation vector is constructed from five blocks. The position block contains current weights w_t and their deviation from the model target. The forecast block contains the conditional mean vector, the shrunk covariance matrix, and the ensemble dispersion for each asset. The regime block contains the filtered posterior over hidden states and its entropy. The friction block contains current cost estimates and a realized-cost tracking error accumulated over recent epochs. The memory block contains the time elapsed since the last trade for each asset, the realized turnover over a trailing window, and the cumulative constraint slack.

The memory block deserves emphasis because it addresses a specific pathology. Without elapsed-time information, a policy cannot distinguish between an asset that was rebalanced yesterday and one that has drifted for a quarter; both may present similar current deviations, yet the expected benefit of trading is different because the cost of the last trade has already been paid and the deviation is more likely to be transitory in the first case. Turnover history serves a similar function, allowing the policy to represent a budget of trading activity that is not exhausted. Cumulative constraint slack lets the policy learn to be conservative when the realized loss trajectory is close to the budget, which is the behavior that a dual variable is meant to induce but which is difficult to learn from a scalar multiplier alone.

Recurrence in the policy network handles the partially observable component. A gated recurrent unit processes the observation sequence and emits a hidden state that summarizes relevant history, and the policy and value heads read from that hidden state. The alternative, stacking a fixed window of past observations, imposes an arbitrary truncation and cannot represent elapsed-time effects beyond the window. The recurrent design introduces training instability that must be managed, but it aligns the architecture with the structure of the problem rather than with the convenience of the optimizer.

3.2 Policy parameterization and action masking

The policy outputs a latent vector that is mapped to a proposed trade vector through a bounded transformation and then projected onto the self-financing subspace. Specifically, the network emits a raw vector q_t and a scalar intensity $s_t \in (0,1)$; the proposed deviation from the drifted position is $\delta_t = \delta_{max} \odot \tanh(q_t)$, and the proposed trade is $\eta_t s_t \delta_t$, where η_t is the risk-budget scale defined in Section 3.4 and δ_{max} is a vector of per-asset position limits. The projection onto $1^T a_t = 0$ subtracts the mean trade, and a subsequent projection onto the long-only and position-limit set is available in closed form for the box-plus-simplex intersection.

The critical modification is masking. Before the trade is executed, each component is passed through the analytic band derived in Section 4:

$$a_{i,t} = sign(\delta_{i,t}) \cdot max\{0, |\delta_{i,t}| - \delta_i^*(o_t)\}$$

where $\delta_i^*(o_t)$ is the state-dependent threshold. The mask is not a post-hoc filter applied to the policy's output for reporting purposes; it is part of the action map, so the policy is

trained on the masked action and receives the reward that the masked action generates. This distinction matters because a filter applied only at deployment creates a mismatch between the distribution of actions seen during training and the distribution actually executed, and policies are known to exploit precisely such mismatches. Embedding the mask in the action map removes the mismatch by construction and reduces the effective action space to trades that clear an economic hurdle. Deep function approximation is required because the observation is high-dimensional and the mapping from observation to trade is not available in closed form once the mask parameters become state-dependent [19].

Masking also resolves a subtle credit assignment problem. Without the mask, a policy that trades small amounts frequently receives a negative reward from costs and must learn to suppress those trades through the value function, which requires many samples and generalizes poorly across assets with different cost levels. With the mask, small trades never occur, and the learning signal is concentrated on decisions that are genuinely marginal. The cost is that the mask imposes prior knowledge; if the analytic band is wrong because the cost model is wrong, the policy cannot discover the error by trading through it. This is an acceptable trade only if the band is set conservatively, which is the argument developed in Section 5.1.

3.3 Reward construction

The per-period reward combines net portfolio return, a tracking penalty relative to a benchmark-consistent reference, and the realized cost:

$$r_t\hat{}net \;=\; w_\{t+1\}{}^{T} r_\{t+1\} \;-\; c_t \;-\; (\psi/2)\|w_\{t+1\} \;-\; w_t\hat{}ref\|^2_\Sigma$$

Here w_t^ref is a reference allocation that encodes mandate constraints such as sector or factor exposures, ψ is a nonnegative penalty coefficient, and ‖x‖²_Σ = $x^{T}\Sigma x$ with Σ the current covariance estimate, so that the penalty is measured in units of forecast variance rather than in weight space. The third term is what prevents the policy from interpreting a low-cost environment as license to concentrate the portfolio: it prices deviations from the reference in risk units, and it does so with the same covariance estimate that the constraint uses, which keeps the reward and the constraint on a common scale.

The discount factor γ in the objective and the horizon over which r_t^net is accumulated interact with the cost structure. Because costs are incurred immediately and benefits accrue over subsequent periods, a heavily discounted objective undervalues the future benefit of a trade and produces excessive inactivity. A common remedy is to align the discount factor with the expected holding period implied by the cost coefficients rather than to adopt a generic value. The choice is a modeling decision that must be documented, because it silently determines the effective investment horizon and therefore the turnover that the policy finds optimal.

3.4 Dynamic resource allocation

The resource allocation layer is the component that makes the policy responsive to uncertainty rather than merely conditioned on it. Three quantities are allocated: risk budget across asset sleeves or strategy modules, trading propensity, and computational effort in the form of re-optimization frequency. The first two are continuous and are allocated every epoch; the third is discrete and is allocated by a threshold rule.

Risk budget allocation across sleeves is represented as a mixture over K sub-policies, each of which is trained on a distinct sleeve or specialization, with mixture weights produced by a gating network that reads the observation:

$$\pi_t(a|o_t) = \sum_k g_k \pi_t\hat{}k(a|o_t),\ g_k = softmax_k(z_k/\tau_g)$$

where π_t^k is the k-th sub-policy, z_k is the corresponding gating logit, and τ_g is a temperature that controls how sharply the mixture concentrates on a single component. The temperature is not constant; it is itself a function of the uncertainty index, so that τ_g increases when uncertainty is high, flattening the mixture and preventing the layer from committing the entire risk budget to a single sleeve on the basis of a confident but unreliable signal. This is the formal sense in which resource allocation is dynamic: what is being allocated is not only capital across sleeves but also the degree of commitment that the architecture makes to any one sleeve's assessment of the opportunity set.

The trading propensity is a scalar multiplier applied to the proposed trade and derived from the ratio of a reference uncertainty level to the current level:

$$\eta_t = clip((u_ref/\hat{u}_t)\hat{}\beta, \eta_min, 1)$$

where û_t is the uncertainty index from Section 2.3, u_ref is a reference level corresponding to the median of the index over the training distribution, $\beta > 0$ governs the sensitivity of the response, and η_min $\in (0,1]$ bounds the contraction so that the portfolio is never completely frozen. When û_t exceeds u_ref, the multiplier falls below one and trades are scaled down proportionally to how far the current environment is from the reference; when û_t is below u_ref, the multiplier saturates at one and no leverage of the budget is permitted. The asymmetry is deliberate. Expanding exposure when uncertainty is unusually low is precisely the behavior that fails most severely in a regime change, because low measured uncertainty is often the state that precedes a volatility transition. The exponent β is a policy choice that trades responsiveness against noise: a large β makes the multiplier jumpy in response to small changes in the index, while a small β renders the layer nearly inert.

Computational effort allocation follows from the same index. Re-optimization of the full policy is performed when the index crosses a threshold or when the drift in any weight exceeds a trigger, and otherwise a cached action is applied. This creates a hierarchy in which the expensive mapping from observation to trade is evaluated only when the state has genuinely changed. The rule requires hysteresis to avoid frequent switching, with separate thresholds for entry into and exit from the high-effort mode. The design is a resource allocation in the literal sense: computation is scarce, and its allocation across decision epochs should follow the same logic as the allocation of risk capital, which is that effort is spent where the marginal value of accuracy is highest.

3.5 Learning algorithm and constrained optimization

The policy is trained with an actor-critic method in the proximal policy optimization family, which constrains the size of the policy update through a clipped surrogate objective and thereby limits the variance of the learning signal [20]. An entropy bonus is added to the objective to prevent premature convergence of the policy to a deterministic rule before the value function has been adequately estimated [21]. Since the mask in Section 3.2 restricts the action set, the clipped surrogate must be computed on the masked distribution; the entropy term is correspondingly computed over the residual space of permitted trades, otherwise the bonus rewards exploration that can never occur.

The conditional value-at-risk constraint is handled through a Lagrangian relaxation. A multiplier $\mu \geq 0$ augments the reward with a penalty proportional to the estimated constraint violation, and μ is updated by dual ascent on the violation, increasing when the constraint is exceeded and decreasing when there is slack. The appeal of the augmented Lagrangian relative to a fixed penalty is that μ adapts to the current policy, so the constraint is enforced in the limit rather than traded against return at a coefficient chosen in advance [18]. In practice, μ is bounded above and updated on a slower timescale than the policy, since simultaneous fast updates of both produce oscillation where the policy chases a moving penalization.

Estimating the constraint requires an estimate of the tail. A distributional value function, or an ensemble of value functions, supplies a sample of return paths from which the conditional expectation can be computed through the optimization in Section 2.4. The ensemble serves double duty: its dispersion enters the uncertainty index, and its members supply the scenarios over which the risk constraint is evaluated. This coupling is efficient but introduces a dependency that must be monitored. If the ensemble members are too similar, the dispersion measure collapses and both the risk estimate and the gating layer receive an artificially confident signal. Diversity must therefore be engineered through bootstrap resampling, randomized network initializations, and, where justified, distinct feature subsets, and it must be verified rather than assumed.

4. Analytical Properties of the Cost-Aware Decision Rule

4.1 Derivation of the no-trade band

The band in Section 3.2 can be derived from a local approximation of the rebalancing problem. Consider a single asset over one decision epoch, and let δ_i denote the deviation of the current weight from the model target, so that δ_i is the trade that would fully restore the target. Deviating from the target by an unexecuted amount incurs a penalty that is quadratic to first order, with curvature $\gamma_i > 0$ reflecting the marginal risk price times the asset's contribution to portfolio variance; trading an amount a_i incurs the cost specified in Section 2.2. The one-period problem is to minimize the sum of the risk penalty evaluated at the residual deviation and the trading cost:

$$J_i(a_i) = \left(\frac{\gamma_i}{2}\right)(\delta_i - a_i)^2 + \kappa_i|a_i| + \left(\frac{\lambda_i}{2}\right)a_i^2$$

where the terms are as defined in Section 2.2 and γ_i is the local curvature of the risk penalty. For $a_i > 0$ the first-order condition is $-\gamma_i(\delta_i - a_i) + \kappa_i + \lambda_i a_i = 0$, and the symmetric case $a_i < 0$ yields the mirror-image condition. Solving gives the optimal trade and the threshold below which no trade is optimal:

$$a_i\hat{} * = max\{0, (\gamma_i\,\delta_i - \kappa_i)/(\gamma_i + \lambda_i)\},\ \delta_i\hat{} * = \kappa_i/\gamma_i$$

The threshold $\delta_i\hat{}$ is the smallest deviation at which the marginal risk reduction from trading exactly offsets the marginal cost at zero trade size. Below it the kink in the absolute value dominates and the optimum is $a_i\hat{} = 0$; above it the optimal trade is strictly smaller than the deviation δ_i, so the target is approached but never reached in a single epoch. Incomplete adjustment is the economically correct behavior when trades are costly: the policy moves partway and waits for drift or for a larger deviation to justify the remainder. The structure mirrors the discrete-time transaction-cost results in which the optimal policy is to keep the traded fraction inside a region around the target and act only at its boundary [22], [23], with the band arising endogenously from the optimization rather than being imposed as a rule of thumb.

4.2 Comparative statics

The closed form supports direct comparative statics that clarify what the band is doing. Differentiating the interior solution with respect to the cost coefficient gives $\partial a_i^\wedge/\partial \kappa_i = -1/(\gamma_i + \lambda_i) < 0$: higher linear costs reduce the executed trade both by shrinking the set of states in which trading is optimal and by reducing the size of the trade conditional on trading. The threshold satisfies $\partial \delta_i^\wedge/\partial \kappa_i = 1/\gamma_i > 0$, so the region of inaction widens in proportion to the cost coefficient and in inverse proportion to the risk price. Raising the risk price narrows the band, because the penalty for remaining away from the target grows and trading becomes worthwhile at smaller deviations. The impact coefficient has a different and instructive effect: $\partial a_i^\wedge/\partial \lambda_i = -(\gamma_i \delta_i - \kappa_i)/(\gamma_i + \lambda_i)^2 < 0$ whenever trading occurs, so larger impact scales down the magnitude of the trade, but the trading threshold $\delta_i^\wedge$ is independent of λ_i. The asymmetry is intuitive, since the threshold is determined by the marginal cost at zero trade, where the quadratic term contributes nothing, while the magnitude of the executed trade is determined by the entire cost curve.

The risk curvature γ_i is the least transparent of the coefficients because it is not a primitive parameter but a reduced-form object that aggregates covariance, risk aversion, and the mandate. In a multi-asset setting the risk penalty is not separable, and the first-order condition couples assets through the covariance matrix: the marginal value of trading asset i depends on the current deviations of every other asset that covaries with it. The separable band in the equation above is therefore a diagonal approximation whose accuracy depends on the off-diagonal structure of the covariance matrix. When correlations are high, the diagonal approximation understates the benefit of coordinated trades and can produce a band that is too wide, leaving the portfolio persistently off target along a common factor. This failure mode is a property of the approximation, not of the learning method, and it is the first place where a state-dependent band learned from data can improve on a fixed formula.

4.3 When a learned policy departs from the analytic rule

Three conditions justify departing from the closed-form band. The first is nonseparability of the cost function. When impact depends on the composition of the traded basket, on the order in which trades are sent, or on tiered fee schedules with thresholds, the marginal cost of a trade depends on the other trades in the same epoch, and no per-asset band can be optimal. A policy that observes the full trade vector and the cost estimator can represent this interaction, and the mask must then be applied to a scalar aggregation of the basket rather than componentwise.

The second condition is the interaction of constraints. When the long-only constraint, leverage limits, and position caps bind simultaneously, the feasible set is a polytope whose active faces change with the state, and the unconstrained band can recommend trades that are infeasible or that violate a cap on aggregate turnover. A learned policy operating on the projected action space internalizes the geometry of the feasible set, provided that the projection is differentiable or that the policy is trained on the projected action.

The third condition is the predictability of returns. The closed-form band treats the target as given and the problem as one of static adjustment. If returns are predictable and the predictive signal decays at a rate that the policy can observe, the value of waiting versus acting depends on the expected evolution of the signal, and the optimal response is to trade more aggressively when the signal is expected to persist and less aggressively when it is

expected to decay [22]. A recurrent policy conditioned on forecast dispersion and regime persistence can represent this trade-off in a way that a fixed threshold cannot.

None of these conditions justifies abandoning the band as a constraint. They justify making the band a function of the state and letting the policy choose trades within it. The constraint retains two functions: it prevents the exploration process from discovering trades that are uneconomic under the most conservative admissible cost model, and it supplies an interpretable benchmark against which the learned rule can be audited. When the two disagree systematically, the disagreement is informative: it either identifies a genuine nonseparability that the analytic rule ignores or a learned behavior that is exploiting a modeling error.

5. Technical Discussion: Failure Modes and Diagnostic Assessment

5.1 Simulator bias and cost-model exploitation

Any policy trained against a simulated market learns the simulator as well as the market. The cost model is the most exposed component because the policy interacts with it directly and can discover trade sequences that are profitable under the assumed friction but not under realized friction. A common pathology is that an underestimated impact coefficient λ_i makes large, frequent trades appear nearly free, and the policy concentrates its behavior in exactly the region where the model error is most damaging. The masking constraint limits but does not eliminate this exposure, because the mask is parameterized by the same coefficients.

The appropriate defense is adversarial rather than merely conservative. Cost coefficients used during training should be drawn from a distribution that straddles the plausible range, with the band evaluated at the pessimistic end so that trades justified only under optimistic costs are excluded. Evaluation should report performance as a function of a cost multiplier applied uniformly to κ and λ, so that the policy's break-even cost level is identified rather than assumed. A policy whose advantage disappears when costs are doubled is not robust, and that fact should be visible in the design documentation rather than discovered in production.

5.2 Nonstationarity and regime overfitting

Financial data are nonstationary, and the regime structure that motivates the observation design in Section 2.3 also creates a specific overfitting risk. A recurrent policy with a regime posterior can learn to recognize the historical sequence of regimes and to condition on their order rather than on their statistical signature. Such a policy performs well on the training path and degrades sharply when the sequence differs, which it will, because regime sequences are not repeated.

The diagnostic that distinguishes genuine regime conditioning from sequence memorization is stratified evaluation with temporal embargo. Performance should be reported separately for each regime identified ex post, with the posterior estimated only from data available at the decision time, and the transition epochs should be excluded from the headline metrics or reported separately, since it is precisely at transitions that the two hypotheses make different predictions. A policy that conditions on statistical signatures performs comparably across regimes and degrades gracefully at transitions; a policy that memorizes sequences shows a characteristic pattern of strong in-regime performance and collapse at transitions.

5.3 Uncertainty miscalibration and gating instability

The resource allocation layer is only as good as the uncertainty index that drives it, and ensemble dispersion is a heuristic rather than a calibrated posterior. Two failure modes follow.

The first is ensemble collapse, in which members converge to similar predictions and dispersion understates true uncertainty. The second is dispersion inflation, in which members disagree for reasons unrelated to predictive accuracy, such as differing initializations that produce different but equally valid feature representations. Either failure propagates directly into the trading propensity through the multiplier in Section 3.4.

Monitoring should therefore treat the uncertainty index as a model in its own right and evaluate it against realized forecast errors in a reliability framework: observations are binned by predicted uncertainty, and the realized error distribution is compared across bins. Monotonicity of realized error in predicted uncertainty is the minimum requirement; without it, the index carries no decision-relevant information and the gating layer should be disabled rather than tuned. Gating stability requires a separate check. The mixture weights in Section 3.4 can oscillate between sub-policies across adjacent epochs, generating turnover that is attributable to the architecture rather than to the market. Measuring the autocorrelation of the gate weights and imposing a smoothness penalty during training both address this, and the turnover decomposition should attribute trades to the gating layer separately from trades attributable to changes in the underlying sub-policies.

5.4 Constraint satisfaction at deployment

The Lagrangian formulation enforces the risk constraint in expectation over the state distribution induced by the policy. Deployment differs from training in two ways that matter: the state distribution shifts, and the multiplier is frozen. A multiplier calibrated to a training distribution in which the constraint binds only occasionally will be too small when the deployment distribution places more mass near the limit, and the constraint will be violated before any adaptive mechanism responds, because dual ascent during deployment is usually disabled or heavily damped.

Three design responses are available and should be combined. The multiplier should be calibrated at a conservative quantile of the training distribution rather than at its mean, which biases the policy toward slack. A hard overlay should cap aggregate exposure when the estimated constraint slack falls below a threshold, operating independently of the learned policy and therefore immune to its errors. And the constraint should be evaluated over a horizon shorter than the decision horizon so that violations are detected before the multi-period budget is exhausted. The overlay introduces a discontinuity that the policy was not trained against, which is a real cost; the alternative is a policy that is smooth and occasionally violates its mandate, which is not acceptable for a risk limit.

5.5 Evaluation logic and reporting

Because the framework involves many design choices, the evaluation must be designed to detect overfitting to the choices rather than only to measure performance. The primary metric is net-of-cost performance relative to a passive benchmark and to a threshold-rebalancing benchmark with the same cost model, reported with dispersion across training seeds, since policy gradient methods exhibit substantial seed variance that can exceed the effect being claimed [24]. Secondary metrics include turnover, the fraction of epochs in which the mask is active, the realized conditional value-at-risk at the constraint level, the frequency and depth of constraint violations, and the regret relative to the analytic band computed ex post on the same decisions. The last metric is the most informative for diagnostics, because it decomposes performance into the part attributable to better trade selection within the band and the part attributable to band modification.

Evaluation should be conducted on rolling-origin splits with embargo periods, so that no decision uses information that would not have been available at the time, and the hyperparameter search should be reported in full, including the number of configurations tried, so that the selection bias can be assessed. Reporting only the best configuration overstates expected performance and is a persistent problem in applied reinforcement learning.

6. Implementation Considerations and Boundary Conditions

The framework presupposes three data conditions. It requires a return history long enough to estimate a covariance matrix after shrinkage and to identify regime structure, which in practice means that the number of assets must be small relative to the sample length or that a factor structure must be imposed. It requires execution data sufficient to estimate the cost coefficients, since a band computed from assumed costs is a band computed from assumptions. And it requires that the observation vector be computable at decision time without look-ahead, which is a stronger requirement than it appears, because covariance shrinkage and regime posteriors are often computed from full-sample statistics in research code and must be reimplemented in a strictly causal form for deployment.

The architecture is intended for portfolios of moderate size with a mandate that includes an explicit loss limit. It is less suitable where the cost structure is dominated by fixed fees that are independent of trade size, because the linear term in the cost model then has the wrong shape and the band derived from it will be too narrow; a fixed-charge formulation would require a different derivation. It is also less suitable where the decision frequency is so high that the discrete-epoch approximation fails and the intra-epoch order placement becomes the binding constraint. In that regime the rebalancing problem dissolves into an execution problem, and the appropriate tools are those developed for optimal execution rather than for portfolio choice.

Two boundary conditions on the resource allocation layer deserve statement. The layer assumes that the uncertainty index is informative about the future, which is an empirical property of the forecaster and not a consequence of the architecture. And the layer assumes that contracting exposure when uncertainty is high is desirable, which is true when the mandate is risk-sensitive and false when the objective is to harvest a risk premium that is compensation for exactly that uncertainty. In the latter case the gating logic would systematically reduce exposure at the moments when expected compensation is highest, and the multiplier should be removed or inverted by design rather than tuned until the backtest improves.

7. Conclusion

This paper has developed a constrained reinforcement learning framework for portfolio rebalancing in which friction, risk limits, and estimation uncertainty are structural features of the decision problem rather than post-processing steps. The trade vector is the action, so the self-financing condition and the cost function are part of the transition and reward; the cost-aware no-trade band is derived analytically and embedded as a hard constraint on the action map; and a dynamic resource allocation layer converts an uncertainty index and a regime posterior into a risk budget, a trading propensity, and a re-optimization frequency. Conditional value-at-risk is imposed as a constraint through Lagrangian dual ascent with ensemble-based tail estimation.

The analytical results establish that the no-trade threshold increases in the linear cost coefficient and decreases in the risk price, while the impact coefficient scales the executed

trade without moving the threshold, and that these comparative statics hold regardless of how the remaining parameters are learned. The framework departs from the closed-form rule under three identifiable conditions: nonseparable costs, interacting constraints, and predictable returns whose signal decays at an observable rate. Outside those conditions the analytic rule is preferable on grounds of estimation efficiency, and a learned policy that claims otherwise is likely to be exploiting model error.

The principal limitations are the dependence of the constraint on an estimated tail, the heuristic nature of the uncertainty index, and the absence of any guarantee that the deployment distribution resembles the training distribution. The diagnostic protocol in Section 5 is designed to make these limitations observable rather than to remove them. Its central instrument is the ex post regret relative to the analytic band, which decomposes the contribution of the learned policy into a component attributable to better trade selection within a fixed feasible set and a component attributable to modifying that set. That decomposition is the appropriate basis on which to decide whether a learned rebalancing policy should be trusted with capital, and it is where the framework's interpretability, rather than its flexibility, carries the decisive weight.

References

1. Markowitz, H. (1952). Portfolio selection. The Journal of Finance, 7(1), 77–91.

2. Perold, A. F. (1988). The implementation shortfall: Paper versus reality. The Journal of Portfolio Management, 14(3), 4–9.

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

4. DeMiguel, V., Garlappi, L., & Uppal, R. (2009). Optimal versus naive diversification: How inefficient is the 1/N portfolio strategy? The Review of Financial Studies, 22(5), 1915–1953.

5. Moody, J., & Saffell, M. (2001). Learning to trade via direct reinforcement. IEEE Transactions on Neural Networks, 12(4), 875–889.

6. Luo, D., Gattas, J. M., & Tan, P. S. S. (2020, July). Real-time defect recognition and optimized decision making for structural timber jointing. In The International Conference on Computational Design and Robotic Fabrication (pp. 36-45). Singapore: Springer Singapore.

7. Jiang, Z., Xu, D., & Liang, J. (2017). A deep reinforcement learning framework for the financial portfolio management problem. arXiv preprint arXiv:1706.10059.

8. Deng, Y., Bao, F., Kong, Y., Ren, Z., & Dai, Q. (2017). Deep direct reinforcement learning for financial signal representation and trading. IEEE Transactions on Neural Networks and Learning Systems, 28(3), 653–664.

9. Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction (2nd ed.). MIT Press.

10. Kaelbling, L. P., Littman, M. L., & Cassandra, A. R. (1998). Planning and acting in partially observable stochastic domains. Artificial Intelligence, 101(1–2), 99–134.

11. Hamilton, J. D. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. Econometrica, 57(2), 357–384.

12. Ang, A., & Bekaert, G. (2002). International asset allocation with regime shifts. The Review of Financial Studies, 15(4), 1137–1187.

13. Ledoit, O., & Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. Journal of Multivariate Analysis, 88(2), 365–411.

14. Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In Advances in Neural Information Processing Systems 30 (pp. 6402–6413).

15. Artzner, P., Delbaen, F., Eber, J.-M., & Heath, D. (1999). Coherent measures of risk. Mathematical Finance, 9(3), 203–228.

16. Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. Journal of Risk, 2(3), 21–41.

17. Altman, E. (1999). Constrained Markov decision processes. Chapman & Hall/CRC.

18. Achiam, J., Held, D., Tamar, A., & Abbeel, P. (2017). Constrained policy optimization. In Proceedings of the 34th International Conference on Machine Learning (pp. 22–31). PMLR.

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(7540), 529–533.

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

21. Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the 35th International Conference on Machine Learning (pp. 1861–1870). PMLR.

22. Gârleanu, N., & Pedersen, L. H. (2013). Dynamic trading with predictable returns and transaction costs. The Journal of Finance, 68(6), 2309–2340.

23. Davis, M. H. A., & Norman, A. R. (1990). Portfolio selection with transaction costs. Mathematics of Operations Research, 15(4), 676–713.

24. Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., & Meger, D. (2018). Deep reinforcement learning that matters. In Proceedings of the AAAI Conference on Artificial Intelligence, 32(1), 3207–3214.