# POLICY CAPACITY AND SYSTEM COST IN INDEPENDENT REINFORCEMENT LEARNING FOR SERIAL INVENTORY CONTROL

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Increasing the expressiveness of a policy can change how independently trained agents affect one another. We investigate this effect in a serial inventory system in which each agent observes local state and minimizes its own quadratic inventory cost. Under a fixed independent proximal policy optimization protocol, we compare a one-gain feedback policy, an affine policy, and a residual multilayer perceptron. In a four-echelon chain with lead time two and demand autocorrelation 0.6, mean system cost increases from 40.679 for the one-gain policy to 55.379 for the affine policy and 56.194 for the neural policy, corresponding to increases of 36.1% and 38.1%. Every restricted-policy run is cheaper than every richer-policy run across eight training seeds per class. Paired bootstrap intervals for the absolute increases are [13.57, 15.78] and [14.03, 17.21]. An ablation that masks the lagged-demand feature while retaining the affine parameterization separates the two channels the main comparison bundles together: inventory-feature capacity alone accounts for a 24.6% increase, 68% of the total effect, and the demand channel adds a further 9.3%, each with exact paired sign-flip $p = 0.0078$. A per-echelon breakdown shows the customer-facing stage reducing its own cost by 12.1% while upstream stages absorb increases of 49% to 54%, the directional pattern the theorem predicts. Across additional demand regimes, the affine cost gap and sensitivity to recent demand increase together, but changing marginal demand variance prevents a causal interpretation of that sweep. We also prove that increasing a stable proportional feedback gain raises upstream variance costs in an idealized linear cascade. The theorem provides an externality argument, while the experiments establish a finite-budget learning effect; they do not certify Nash convergence or isolate a causal neural mechanism. These findings support evaluating policy classes by their system-level consequences when learning objectives are decentralized.

## 1 INTRODUCTION

Policy expressiveness is usually assessed by the behaviors a learner can represent. In a network of decision makers, those behaviors also shape the learning problem faced by other agents. A replenishment decision, for example, becomes demand for an upstream supplier. A policy that reacts more strongly to a useful local signal can therefore change both the supplier's input distribution and the cost of maintaining inventory. The relationship between a richer local policy and better system performance is consequently an empirical question when rewards are decentralized.

Serial inventory control provides a tractable setting for studying this question. Classical research has established that replenishment rules can amplify order variability and that decentralized inventory decisions can differ from the system optimum (Dejonckheere et al., 2003; Cachon and Zipkin, 1999). Deep reinforcement learning adds a further design choice: the function class available to each local decision maker. Here, we hold the learning algorithm, local objective, and training budget fixed while varying the actor parameterization and, with it, the subset of the local observation each actor may condition on. Section 6.1.1 reports an ablation that separates these two factors.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Our main comparison is deliberately narrow. Four independent agents learn with separate actors and critics, using only local observations and costs. We compare proportional feedback on inventory position with an affine actor that can use recent demand and additional inventory features, and with a residual neural actor. After 200 training iterations, the richer actors incur substantially higher aggregate cost. The affine comparison shows that a deep architecture is not necessary for the observed increase: a small expansion of the linear decision rule is sufficient under this training protocol.

We examine the result at three levels. First, a frequency-domain argument shows why stronger proportional responses impose an upstream externality in the stable linear model. Second, run-level analyses quantify the cost difference and its sensitivity to training seeds and evaluation demand. Third, experiments across demand autocorrelation and chain length probe where the effect becomes small or large. Together, these analyses establish an association between policy class and learned system cost without equating finite-budget training outcomes with game-theoretic equilibria.

The contributions are a controlled policy-class comparison with all 59 saved runs included, a complete proof of a positive upstream externality for proportional policies, and an analysis of demand sensitivity and chain length with explicit uncertainty. We also identify an exact redundancy in the inventory features that limits the interpretation of individual affine coefficients. The study is intended as evidence about policy-class choice in independent learning, rather than a new optimization algorithm or a general claim that larger policies are harmful.

## 2 Related work

**Inventory dynamics and decentralized incentives.** Dejonckheere et al. (2003) analyze replenishment policies using control-theoretic methods and show how policy design affects order amplification. Cachon and Zipkin (1999) study competitive and cooperative decisions in a two-stage inventory system, demonstrating that individually chosen base-stock policies can be inefficient. Our linear analysis follows this established view of inventory systems as coupled dynamical decision problems. The specific question is how the local actor class changes the outcome of a fixed independent learning procedure. We do not claim that decentralized inefficiency or transfer-function analysis is new.

**Cost and the bullwhip effect.** Order-variance amplification and inventory cost measure different properties of a supply chain. Chen and Lee (2012) show that the interpretation of bullwhip measurements depends on operational details and aggregation. We therefore evaluate the sum of realized local inventory costs. This choice does not imply that order variability is irrelevant; it ensures that the principal endpoint matches the objective whose system-level consequences are at issue.

**Reinforcement learning for inventory and multi-agent control.** Oroojlooyjadid et al. (2020) apply deep Q-learning to the beer game and evaluate inventory cost under different co-player policies. Their work illustrates that learning-based inventory control already has substantive cost-based evaluations. Independent PPO has also been studied in cooperative multi-agent benchmarks (Schroeder de Witt et al., 2020), and PPO-based methods can be strong cooperative baselines (Yu et al., 2022). Our setting differs in its local rewards: each agent minimizes its own quadratic cost, without a shared return or centralized critic. We use PPO and generalized advantage estimation as established tools (Schulman et al., 2017; 2016), not as methodological contributions.

**Reliable evaluation.** Training randomness can materially affect deep reinforcement learning comparisons (Henderson et al., 2018). Agarwal et al. (2021) emphasize uncertainty estimates when only a small number of runs is available. We report seed-level outcomes, effect sizes, paired resampling intervals, and evaluation-seed sensitivity. Multiple statistical procedures applied to the same runs are sensitivity analyses, not independent experimental replications.

## 3 Inventory system and policy classes

### 3.1 Dynamics and objectives

Consider a serial chain with N echelons, indexed from the customer-facing echelon 1 to the most upstream echelon N. Every echelon places an order using its end-of-previous-period state. Customer demand follows a stationary Gaussian autoregressive process:

$$d_t = \mu + u_t, \qquad u_t = \rho u_{t-1} + \varepsilon_t, \qquad \varepsilon_t \sim \mathcal{N}(0, \sigma^2). \tag{1}$$

Here $\mu$ is mean demand, $\sigma$ is the innovation standard deviation, and $|\rho| < 1$. Consequently, stationary demand variance is $\sigma^2/(1-\rho^2)$. The experiments use $\mu$ = 100 and $\sigma$ = 1. Demand is not censored. Echelon i $> 1$ receives the current order of echelon i $-$ 1 as its current demand. In the linear environment, each order arrives in full after the specified lead time $L_i$; shipments are not rationed and orders may be negative.

Let $NS_{i,t}$ denote end-of-period net stock, $P_{i,t}$ inventory in transit, and $IP_{i,t} = NS_{i,t} + P_{i,t}$ inventory position. The inventory-position target is $S_i = \mu(L_i + 1)$, and the net-stock target is $n_i^* = S_i - \mu L_i$. The local and aggregate evaluation objectives are

$$c_{i,t} = q_i(NS_{i,t} - n_i^*)^2, \qquad J_i = \lim_{T\to\infty} \frac{1}{T} \mathbb{E}\left[\sum_{t=1}^{T} c_{i,t}\right], \qquad C = \sum_{i=1}^{N} J_i. \tag{2}$$

The reported experiments set $q_i = 1$. The expected quadratic cost includes both variance and squared bias relative to the target; it reduces to a variance cost only for a correctly centered stationary policy. Training approximates a discounted local reward objective, whereas evaluation measures long-run average cost. We retain this distinction when interpreting the results.

### 3.2 Local observations and actions

Each agent receives four normalized features before ordering:

$$f_{0,i,t} = \frac{S_i - IP_{i,t-1}}{\sigma}, \qquad f_{1,i,t} = \frac{d_{i,t-1} - \mu}{\sigma},$$
$$f_{2,i,t} = \frac{\mu L_i - P_{i,t-1}}{\sigma}, \qquad f_{3,i,t} = \frac{n_i^* - NS_{i,t-1}}{\sigma}. \tag{3}$$

The features represent inventory-position shortfall, most recently observed demand, pipeline shortfall, and net-stock shortfall, respectively. The order is $o_{i,t} = \mu + \sigma a_{i,t}$. Thus, the actor learns deviations from the mean replenishment quantity. No agent observes another agent's state or receives another agent's reward.

The features are not independent. Before clipping, $f_0 = f_2 + f_3$ exactly. Hence an affine actor has five stored mean parameters but only four identifiable affine degrees of freedom on the observation manifold, including its bias. This redundancy does not eliminate the added demand feature, but it prevents interpreting all inventory coefficients as separately identified physical effects. Section 7 returns to this issue.

### 3.3 Actor parameterizations

Table 1 summarizes the three classes. The one-gain policy is exactly representable by the affine actor. The fixed-width tanh network provides a richer nonlinear parameterization, but exact global containment of the affine class is not established for this architecture. We therefore avoid treating all three implementations as a strictly nested sequence of function spaces.

Table 1. Actor classes. The neural network has two hidden layers of 64 tanh units. Clipping is applied as described in Section 5.3.

| **Actor** | **Mean residual action** | **Mean parameters** | **Initialization** |
|---|---|---|---|
| One gain | $\alpha f_0$ | 1 active | $\alpha = 1$ |
| Affine | $w^\top$ f + b | 5 stored | $w_0 = 1$; others 0 |

| Actor | Mean residual action | Mean parameters | Initialization |
|---|---|---|---|
| Residual MLP | $f_0$ + g$\theta$(f) | 4,545 | Zero final layer |

All actors start at proportional feedback with gain one. For the neural actor, the final network layer is initialized to zero and its output is added to $f_0$. Linear actors initialize the $f_0$ weight to one and other active weights and biases to zero. Gaussian exploration has a learned scalar standard deviation per agent. The parameter counts in Table 1 exclude this scalar and the separate critic.

# 4 A POSITIVE UPSTREAM EXTERNALITY IN THE LINEAR CASCADE

## 4.1 PROPORTIONAL FEEDBACK REPRESENTATION

This section concerns idealized, deterministic proportional policies without clipping. Let $\alpha_i \in (0, 2)$ and $\varphi_i = 1 - \alpha_i$. For centered demand $u_{i,t}$, introduce the inventory-position shortfall state $x_{i,t}$. Under the event timing in Section 3, the state, centered order, and centered net stock satisfy

$$x_{i,t} = \phi_i x_{i,t-1} + u_{i,t}, \qquad \widetilde{o}_{i,t} = \alpha_i x_{i,t-1},$$
$$\widetilde{NS}_{i,t} = -\left( x_{i,t} + \alpha_i \sum_{k=1}^{L_i} x_{i,t-k} \right). \tag{4}$$

The associated transfer function from incoming demand to outgoing orders is $T_i(z) = \alpha_i/(z - \varphi_i)$. Stability follows from $|\varphi_i| < 1$ together with $|\rho| < 1$. Upstream demand is generated by successive applications of these stage filters.

## 4.2 SPECTRAL MONOTONICITY AND COST EXTERNALITY

**Lemma 1.** For a stable stage filter $T(z) = \alpha/(z - \varphi)$, where $\varphi = 1 - \alpha$, define $c = \cos \omega$ and $D = 1 - 2\varphi c + \varphi^2$. Then

$$\frac{\partial |T(e^{j\omega})|^2}{\partial \alpha} = \frac{2\alpha(1 - \cos\omega)(2 - \alpha)}{D^2}. \tag{5}$$

For $0 < \alpha < 2$ and $0 < |\omega| \leq \pi$, this derivative is strictly positive. At $\omega = 0$ it is zero, consistent with the unit steady-state gain $T(1) = 1$.

**Proof.** The squared magnitude is $\alpha^2/D$. Differentiating with respect to $\alpha$ and substituting $\varphi = 1 - \alpha$ yields the displayed expression. Stability implies $D > 0$. All numerator factors are positive except $1 - \cos \omega$, which vanishes only at zero frequency on the stated interval. Appendix A gives the algebraic steps.

**Proposition 1.** Fix all gains other than $\alpha_i$ in a stable serial linear cascade. Assume that the incoming demand spectrum at stage i is positive on a set of nonzero frequencies of positive measure, $q_j > 0$, and each upstream net-stock transfer function is nonzero on a subset of that set with positive measure. Then, for every $i < N$,

$$E_i = \sum_{j>i} \frac{\partial J_j}{\partial \alpha_i} > 0. \tag{6}$$

For the last echelon, the externality is zero because there is no upstream agent.

**Proof.** For $j > i$, let $M_j(z)$ map stage i's outgoing orders to stage j's centered net stock, holding the upstream policies fixed. Both $M_j$ and the incoming demand spectrum $S_{u_i}$ are independent of $\alpha_i$:

the unrationed cascade has no feedback from upstream stock to downstream demand. The spectral variance formula gives

$$\frac{\partial J_j}{\partial \alpha_i} = \frac{q_j}{2\pi} \int_{-\pi}^{\pi} |M_j(e^{j\omega})|^2 \, S_{u_i}(\omega) \, \frac{\partial |T_i(e^{j\omega})|^2}{\partial \alpha_i} \, d\omega > 0. \tag{7}$$

Differentiation is valid locally within the stable gain region. Lemma 1 makes the integrand nonnegative, and the nondegeneracy assumptions make its integral positive. Summing over upstream stages proves the result. For Gaussian AR(1) customer demand with positive innovation variance and nonzero stable stage filters, the required spectral excitation is present.

The proposition formalizes a local externality: increasing one gain, with all other policies fixed, raises upstream variance costs. It does not show that a local optimizer will increase that gain, that independent PPO converges, or that expanding an actor class must raise system cost. In particular, it does not directly describe affine feed-forward policies or nonlinear actors. It supplies a mechanistic rationale to investigate, rather than a theorem about the neural experiments.

# 5 Experimental protocol

## 5.1 Training and evaluation

Each agent has its own Gaussian actor, 64–64 tanh critic, and Adam optimizer. Actors and critics are not shared across echelons. Training uses the clipped PPO objective with generalized advantage estimation, 64 parallel chains, 128 periods per rollout, and 200 iterations. Each run therefore uses 1,638,400 joint environment transitions; one transition contains an action for every agent. The learning rate is 0.003, discount factor 0.99, GAE parameter 0.95, PPO clipping threshold 0.2, and entropy coefficient 0.02. Appendix B gives the remaining implementation details.

The main setting is N = 4, $L_i$ = 2, and $\rho$ = 0.6, with seeds 1–8 for each actor class. Demand experiments use $\rho \in \{0, 0.3, 0.8\}$, with three seeds per tested class. Chain-length experiments use N = 2 with three seeds and N = 6 with four seeds, at $\rho$ = 0.6. The neural class is evaluated at $\rho \in \{0, 0.6\}$ only. A demand-masking ablation class, described in Section 6.1.1, adds eight further seeds at the main setting. These experiments comprise 67 distinct training records, all at 200 iterations. No saved run is excluded.

Evaluation uses deterministic actor means, 64 parallel chains, a 10,000-period horizon, and a 1,500-period burn-in. The principal analysis uses evaluation seed 99, the common seed present in every record. This gives 544,000 post-burn-in chain-period observations per run. These observations are temporally correlated and are not treated as independent statistical replicates. Additional evaluation seeds 1234 and 7 are available for main-setting training seeds 5–8 and for the auxiliary configurations.

## 5.2 Statistical analysis

The unit of replication is a training seed. Tables report means and sample standard deviations across seeds. Percentage differences are ratios of class means, not averages of per-run percentage differences. Because the same seed identifiers are used across classes and the demand generator is separately seeded, the primary reanalysis resamples matched seed pairs. We use 100,000 paired bootstrap samples and report percentile 95% intervals. These are empirical uncertainty summaries, not exact small-sample coverage guarantees.

For the eight-seed main comparison, we enumerate all 256 sign assignments of paired cost differences. The resulting two-sided sign-flip test is exact under the within-pair exchangeability null; it is not a randomization guarantee for an unrandomized architecture experiment. Holm correction covers the two richer-versus-one-gain comparisons. Unpaired Welch tests, exact Mann–Whitney tests, and exhaustive label permutations are reported as sensitivity analyses in Appendix C. The affine-versus-neural comparison is exploratory, and a nonsignificant result is not interpreted as equivalence.

### 5.3 Implementation features affecting interpretation

The actor clips observation components to $[-20, 20]$ and clips its mean residual action to $[-60, 60]$. Gaussian samples during training are not action-clipped. Rewards are negative local costs divided by ten and clipped below at $-50$; evaluation uses the original quadratic cost. Clipping frequencies were not logged. Consequently, the learning experiment cannot be assumed to coincide everywhere with the unclipped linear model.

Each training rollout also draws a fresh stationary AR(1) demand segment, while inventory state continues across rollouts. Demand is therefore discontinuous in its temporal dependence at rollout boundaries. Evaluation instead draws one continuous segment. This shared implementation choice may interact differently with different policy classes, particularly those that use the previous demand observation. The results characterize the implemented protocol; they do not establish invariance to continuous-demand training.

## 6 Results

### 6.1 Richer actors incur higher cost in the main setting

Table 2 and Figure 1 show a large separation at $\rho = 0.6$. Mean cost is 40.679 for the one-gain actor, 55.379 for the affine actor, and 56.194 for the residual neural actor. Relative to one-gain feedback, the increases are 36.1% and 38.1%. The maximum one-gain cost is 41.223, below the minimum richer-policy cost of 53.152 by 11.929 units.

Table 2. Main results at N = 4, L = 2, and $\rho = 0.6$. Lower cost is better. Standard deviations describe training-seed variation.

| Actor | Seeds | Mean cost ± SD | Median | Cost increase |
|---|---|---|---|---|
| One gain | 8 | 40.679 ± 0.282 | 40.610 | Reference |
| Affine | 8 | 55.379 ± 1.539 | 55.259 | +36.1% |
| Residual MLP | 8 | 56.194 ± 2.403 | 56.111 | +38.1% |

The paired absolute differences are 14.700 with a 95% bootstrap interval [13.57, 15.78] for affine, and 15.515 [14.03, 17.21] for the neural actor. Percentage intervals are [33.27%, 38.88%] and [34.41%, 42.36%]. Both sign-flip p-values equal 0.0078125, the minimum attainable two-sided value for eight nonzero paired differences of one sign; both Holm-adjusted values are 0.015625. Unpaired analyses also support a cost difference, but their smaller p-values depend on a different exchangeability assumption.

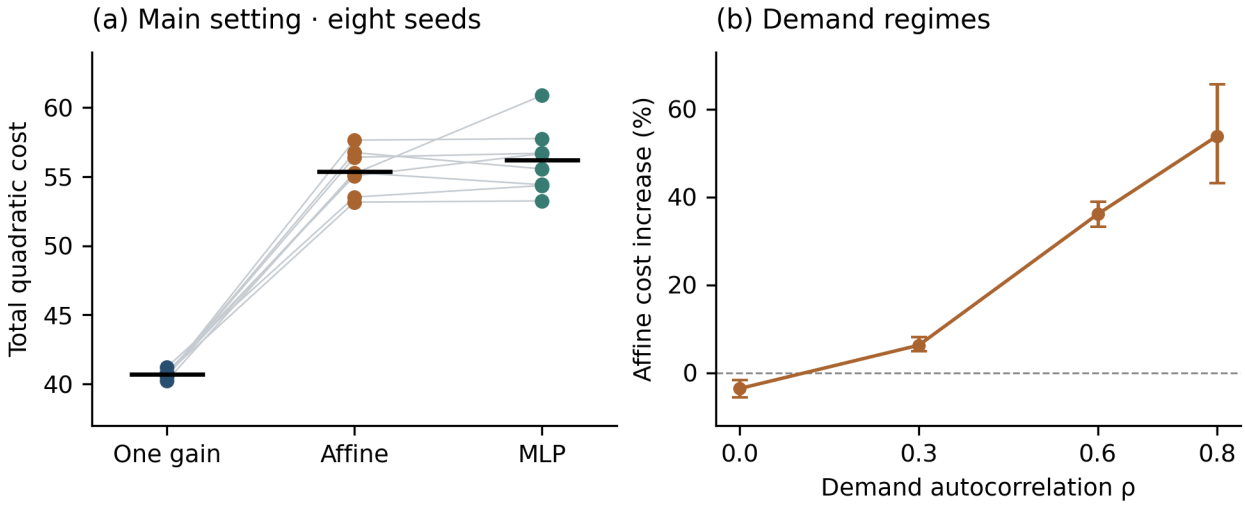


Figure 1. Policy-class comparisons. (a) Each dot is one trained policy evaluated with demand seed 99; thin lines connect matching training seeds and black bars show class means. (b) Affine cost increases across demand regimes; error bars are paired percentile 95% bootstrap intervals. Seed counts are 3, 3, 8, and 3 in increasing $\rho$ order. Marginal demand variance also changes with $\rho$.

The neural-minus-affine mean difference is 0.815, with paired interval $[-0.33, 2.37]$ and sign-flip p = 0.359. Thus, the data do not resolve a difference between these richer actors. They also do not demonstrate equal policies, equivalent costs, or a common equilibrium. Removing any one matched seed pair leaves the affine increase at least 35.3% and the neural increase at least 36.5%, so no single pair accounts for the main result.

The fixed training budget leaves open whether the richer actors require more optimization. The independent-demand comparison provides a useful qualification: at $\rho = 0$, the affine actor achieves mean cost 11.388 versus 11.813 for one-gain feedback, a 3.6% reduction under the same 200-iteration protocol (three training seeds per class; Table 3). This reversal argues against a uniform training disadvantage for the affine class. It does not exclude slower convergence specifically under correlated demand or establish equilibrium quality. The neural mean at $\rho = 0$ is $12.110 \pm 0.316$, or 2.5% above one-gain feedback, so the reversal applies only to the affine actor. Longer training across demand regimes is needed to distinguish persistent cost differences from differences in convergence speed.

#### 6.1.1 Separating policy capacity from demand information access

The main comparison varies two things at once: the one-gain actor is masked to the inventory-position shortfall alone, while the richer actors receive all four features including lagged demand. Any cost difference therefore bundles representational capacity with information access. To separate them we train an affine-no-demand class, keeping the affine parameterization and bias on the inventory features $f_0$, $f_2$, $f_3$ but masking $f_1$. Protocol, seeds, initialization and evaluation match Section 6.1.

Table 2b. Demand-masking ablation at N = 4, L = 2, $\rho = 0.6$, with eight matched training seeds per class.

| Actor class | Seeds | Mean cost | SD | vs one gain |
|---|---|---|---|---|
| One gain (1 feature) | 8 | 40.679 | 0.282 | Reference |
| Affine, demand masked | 8 | 50.678 | 2.280 | +24.6% |
| Affine, all features | 8 | 55.379 | 1.539 | +36.1% |
| Residual MLP, all features | 8 | 56.194 | 2.403 | +38.1% |

Paired over the same eight training seeds, the demand-masked affine class increases cost by 9.999 units over one gain (95% paired bootstrap interval [8.44, 11.63], exact sign-flip p = 0.0078). Restoring the demand feature adds a further 4.701 units ([4.00, 5.32], p = 0.0078). The 14.700-unit total separation therefore decomposes into 68% attributable to inventory-feature capacity and 32% to the demand channel.

Both channels are therefore active. Attributing the whole difference to parameter count would be incorrect, but the effect is not reducible to information access either: removing the demand channel still leaves a 24.6% increase with complete seed separation. Capacity is the larger contributor at this operating point. The ablation does not decompose the neural actor, whose additional curvature is not isolated by masking a feature.

#### 6.1.2 Where the cost lands

Proposition 1 predicts a directional externality: raising a stage's gain leaves downstream stages unaffected and inflates upstream variance costs. Aggregate cost cannot distinguish that from a uniform optimization failure, so we report per-echelon costs recomputed from the twelve saved main-setting policies, training seeds 5–8 (Table C4).

The customer-facing stage is the only one that improves, by 12.1% under the affine class and 9.4% under the neural class, while upstream stages rise by 32% to 70%. This matches the direction of

Proposition 1: the downstream agent lowers its private cost by responding more strongly to demand, and the resulting order variability is absorbed upstream by agents whose objectives do not enter its gradient. The aggregate increase is thus not a uniform degradation. This is an observational match under learned policies, not a controlled test of the proposition, which fixes all other policies.

### 6.2 The cost gap varies with demand autocorrelation

The affine-versus-one-gain gap increases across the four evaluated demand regimes, from $-3.6\%$ at $\rho = 0$ to 53.9% at $\rho = 0.8$ (Table 3). Section 6.1 discusses the independent-demand endpoint and its implications for the fixed-budget comparison.

Table 3. Demand regimes at N = 4 and L = 2. Costs are mean $\pm$ SD. Demand sensitivity is the mean stage-1 action derivative with respect to the previous-demand feature; $\sigma = 1$ is fixed.

| $\rho$ | Seeds per class | One gain cost | Affine cost | Gap | Demand sensitivity |
|---|---|---|---|---|---|
| 0.0 | 3 | $11.813 \pm 0.024$ | $11.388 \pm 0.212$ | -3.6% | -0.016 |
| 0.3 | 3 | $20.585 \pm 0.158$ | $21.874 \pm 0.177$ | +6.3% | +0.366 |
| 0.6 | 8 | $40.679 \pm 0.282$ | $55.379 \pm 1.539$ | +36.1% | +0.993 |
| 0.8 | 3 | $85.416 \pm 0.378$ | $131.480 \pm 9.100$ | +53.9% | +1.683 |

To describe the learned response to demand, we measure the derivative of the mean residual action with respect to the previous-demand feature, averaged over observations from the final training rollout. For affine policies away from clipping boundaries, this is the demand coefficient. For neural policies, it is a local sensitivity averaged over visited states. It is not a coefficient of an equivalent global linear model.

The affine stage-1 sensitivity increases from $-0.016$ to 1.683 as $\rho$ increases. At $\rho = 0.6$, its mean is 0.993, compared with 0.823 for the neural actor. The corresponding across-seed t intervals are [0.920, 1.066] and [0.714, 0.931]. These measurements are consistent with the richer policies exploiting recent demand. They do not identify how much of the cost difference that pathway causes, because the actors simultaneously change other responses.

The demand sweep has an additional confound: innovation variance is fixed, so stationary demand variance increases from 1.000 to 2.778 across the sweep. Predictability, fluctuation magnitude, clipping exposure, and optimization conditions can therefore change together. A constant-marginal-variance sweep and an intervention that removes the demand channel are needed to distinguish these explanations. Correlation across four aggregate regime means is not a substitute for such interventions.

### 6.3 Longer chains retain the effect with greater uncertainty

At $\rho = 0.6$, the affine penalty is 17.3% for N = 2, 36.1% for N = 4, and 37.0% for N = 6 (Table 4 and Figure 2). The corresponding bootstrap intervals are [14.75%, 21.31%], [33.27%, 38.88%], and [23.86%, 50.62%]. The smaller auxiliary samples limit inference, especially at N = 6, where affine cost varies substantially across runs.

Table 4. Chain-length results at $\rho = 0.6$ and L = 2. Costs are mean $\pm$ SD. N = 4 reuses the main experiment.

| Echelons | Seeds per class | One gain cost | Affine cost | Gap |
|---|---|---|---|---|
| 2 | 3 | $19.107 \pm 0.066$ | $22.411 \pm 0.743$ | +17.3% |
| 4 | 8 | $40.679 \pm 0.282$ | $55.379 \pm 1.539$ | +36.1% |
| 6 | 4 | $61.185 \pm 1.278$ | $83.826 \pm 8.189$ | +37.0% |

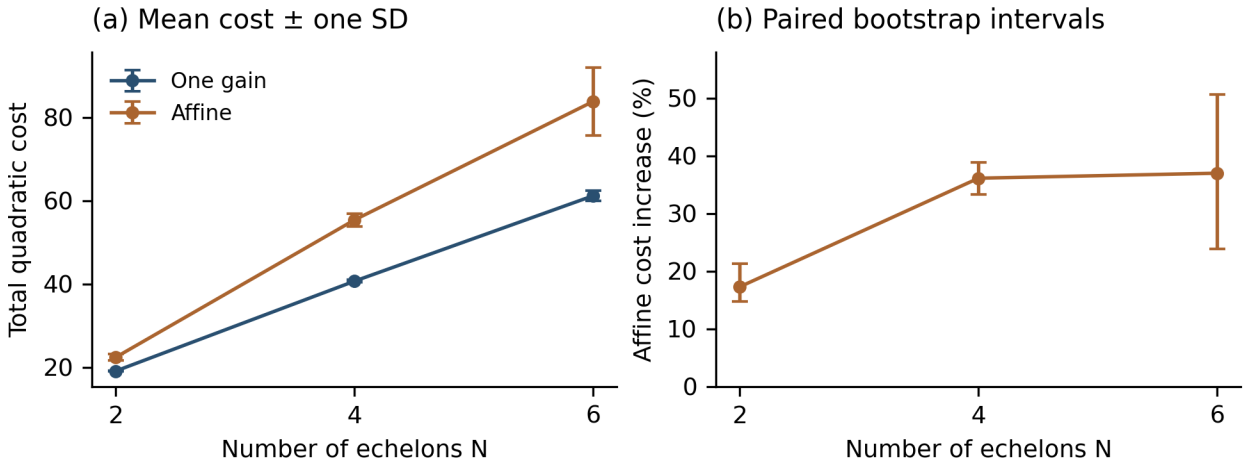



Figure 2. Chain-length sensitivity at $\rho = 0.6$ and L = 2. (a) Mean cost with ± one sample SD across training seeds. (b) Affine percentage cost increase with paired percentile 95% bootstrap intervals. Counts are 3, 8, and 4 seeds per class for N = 2, 4, and 6. Lines connect evaluated settings and do not assert interpolation or saturation.

The point estimates rise sharply from two to four echelons and change little from four to six. This pattern is compatible with a plateau over the tested sizes, but it does not establish asymptotic saturation or a monotone chain-length law. Proposition 1 concerns a perturbation to a fixed gain vector; it does not determine how independently learned policies change when the number of agents changes.

### 6.4 Evaluation demand and exploration

For main-setting training seeds 5–8, averaging each policy's costs over all three evaluation seeds yields class means of 40.449, 55.951, and 56.241. On this four-policy subset, the affine and neural increases remain 38.3% and 39.0%. The ratios of between-policy variance of evaluation means to average within-policy evaluation variance are 6.83, 77.38, and 128.88, respectively. These descriptive ratios suggest that evaluation demand noise is smaller than policy-to-policy variation in this subset. They neither certify stationarity nor replace additional training seeds.

Although the entropy coefficient and standard-deviation floor are common, the learned exploration levels differ across classes. Moreover, the saved scalar standard deviation describes only the customer-facing actor. Deterministic evaluation removes action-sampling noise at test time, but exploration during training can still alter the policies reached. Matching hyperparameters therefore does not eliminate exploration as an explanation, and an exploration-matched ablation remains necessary.

## 7 Discussion and limitations

**Capacity, information, and optimization.** The experiment changes a package of representational and optimization properties. Relative to one-gain feedback, the affine actor adds recent demand, separate inventory features, and a bias. The neural actor changes parameterization and curvature further. The strongest conclusion is that these richer actor designs lead to higher finite-budget system cost under a shared protocol. The data do not isolate parameter count, information access, or function approximation as the unique cause. Section 6.1.1 reports the masked-demand ablation this calls for: 68% of the affine increase is attributable to inventory-feature capacity and 32% to the demand channel, so neither factor alone accounts for the result. Function approximation remains unseparated.

**Learning outcomes and equilibrium.** Within the one-gain class, the solver of Appendix A.2 gives a cooperative optimum of 31.974, a gain-one initialization cost of 38.250, and a forward best-response reference of 43.890. The trained one-gain mean of 40.679 lies between the initialization and the best-response reference, consistent with a fixed budget halting the learners partway along a trajectory departing from the initialization. Learned one-gain parameters across the eight seeds are

1.507 (SD 0.048), 1.005 (0.023), 0.918 (0.032) and 0.898 (0.025), against [1.6443, 1.1317, 1.0231, 1.0039]: the customer-facing stage moves most of the way toward its best response while upstream stages move little. A restricted equilibrium need not minimize aggregate cost, and finite-budget PPO uses stochastic actions, discounting, clipping and segmented demand, so neither agreement nor disagreement in total cost establishes approximate Nash optimality. Certification requires per-agent unilateral-deviation tests.

**Identifiability.** The identity $f_0 = f_2 + f_3$ gives an exact null direction for an unclipped affine actor: adding $\delta$ to the $f_0$ weight and subtracting $\delta$ from both the $f_2$ and $f_3$ weights preserves its output on valid observations. Individual inventory coefficients are consequently not unique descriptions of behavior. Mean costs and derivatives along physically valid perturbations are more interpretable than raw weights alone. Clipping can break the feature identity, and approximate flatness of an empirical objective is a separate question requiring its own measurements.

**Scope and replication.** All experiments use synthetic Gaussian demand, lead time two, homogeneous quadratic weights, a serial topology, and unrationed linear dynamics. The auxiliary studies have only three or four seeds per class, and the main study has eight. No central-team learner was trained, so the cost gaps are not price-of-anarchy estimates; the analytical anchors above are reference points within a restricted policy class and do not establish equilibrium quality for the trained actors. Section 6.1.2 shows that a larger aggregate cost does not imply every agent is worse off.

These limits identify a focused next stage of research: extend training across demand regimes, match exploration, maintain continuous demand during training, vary autocorrelation at fixed marginal variance, test unilateral deviations, compare against a centralized-critic method such as MAPPO to separate capacity effects from decentralized value estimation, and repeat the comparison under shipment rationing and realistic demand traces. The present evidence supports none of these extensions in advance. Residual initialization ensures a stable gain-one mean policy at the start of the idealized model, but it is not a guarantee of stability throughout learning.

## 8 Conclusion

In a controlled serial inventory experiment, affine and residual neural actors trained independently incur 36–38% higher aggregate cost than a one-gain actor after the same training budget. The separation survives seed-paired analysis and additional evaluation demand on a subset of policies. A linear spectral argument establishes a positive upstream cost externality from stronger proportional responses, while demand and chain-length experiments describe the conditions under which the observed cost gap changes. The practical implication is to evaluate actor design together with local incentives and system cost. More expressive local decisions can produce worse learned network outcomes even when every agent follows the same learning procedure.

## AI Use Statement

Generative AI assistance was used to restructure and edit the manuscript, identify and summarize related literature, inspect the implementation and format this document. The reported PPO training outcomes were taken from existing experiment records; the manuscript preparation did not generate additional PPO training runs.

## References

Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A., and Bellemare, M. G. 2021. Deep Reinforcement Learning at the Edge of the Statistical Precipice. Advances in Neural Information Processing Systems 34. https://arxiv.org/abs/2108.13264

Cachon, G. P., and Zipkin, P. H. 1999. Competitive and Cooperative Inventory Policies in a Two-Stage Supply Chain. Management Science 45(7), 936–953. https://doi.org/10.1287/mnsc.45.7.936

Chen, L., and Lee, H. L. 2012. Bullwhip Effect Measurement and Its Implications. Operations Research 60(4), 771–784. https://doi.org/10.1287/opre.1120.1074

Dejonckheere, J., Disney, S. M., Lambrecht, M. R., and Towill, D. R. 2003. Measuring and Avoiding the Bullwhip Effect: A Control Theoretic Approach. European Journal of Operational Research 147(3), 567–590. https://doi.org/10.1016/S0377-2217(02)00369-7

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger, D. 2018. Deep Reinforcement Learning That Matters. Proceedings of the AAAI Conference on Artificial Intelligence 32(1). https://arxiv.org/abs/1709.06560

Oroojlooyjadid, A., Nazari, M., Snyder, L. V., and Takáč, M. 2020. A Deep Q-Network for the Beer Game: A Deep Reinforcement Learning Algorithm to Solve Inventory Optimization Problems. arXiv:1708.05924, version 4. https://arxiv.org/abs/1708.05924v4

Schroeder de Witt, C., Gupta, T., Makoviichuk, D., Makoviychuk, V., Torr, P. H. S., Sun, M., and Whiteson, S. 2020. Is Independent Learning All You Need in the StarCraft Multi-Agent Challenge? arXiv:2011.09533. https://arxiv.org/abs/2011.09533

Schulman, J., Moritz, P., Levine, S., Jordan, M. I., and Abbeel, P. 2016. High-Dimensional Continuous Control Using Generalized Advantage Estimation. International Conference on Learning Representations. https://arxiv.org/abs/1506.02438

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347. https://arxiv.org/abs/1707.06347

Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., and Wu, Y. 2022. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. Advances in Neural Information Processing Systems 35. https://arxiv.org/abs/2103.01955

# A Derivation and numerical checks

## A.1 Spectral derivative

For the stage filter in Lemma 1, $|T|^2 = \alpha^2/\mathrm{D}$ and $\mathrm{dD}/\mathrm{d}\alpha = 2(\mathrm{c} - \varphi)$. Applying the quotient rule gives the following identity:

$$\frac{\partial(\alpha^2/D)}{\partial\alpha} = \frac{2\alpha D - 2\alpha^2(c-\phi)}{D^2}$$
$$= \frac{2\alpha[D - \alpha(c-\phi)]}{D^2} = \frac{2\alpha(2-\alpha)(1-c)}{D^2}. \tag{8}$$

The numerator simplification uses $\mathrm{D} - \alpha(\mathrm{c} - \varphi) = (2 - \alpha)(1 - \mathrm{c})$. In a neighborhood of any stable $\alpha$, the denominator stays bounded away from zero on the compact frequency interval. This justifies differentiation under the spectral integral for a fixed stable upstream subsystem and an integrable input spectrum. Strict positivity requires the overlap condition stated in Proposition 1; a constant demand process alone is not sufficient.

## A.2 Analytical cost computation

The state-space implementation augments the autoregressive demand state with each stage's inventory-position shortfall and the lag states needed for lead times. With state transition matrix A and innovation loading B, the stationary covariance P solves $\mathrm{P} = \mathrm{A}\mathrm{P}A^\top + \mathrm{B}B^\top$. The implementation uses the doubling recursion initialized by $P_0 = \mathrm{B}B^\top$ and $A_0 = \mathrm{A}$:

$$P_{k+1} = P_k + A_k P_k A_k^\top, \qquad A_{k+1} = A_k^2. \tag{9}$$

Costs follow by applying the net-stock output vectors to P. Models with a spectral radius at least $1 - 10^{-4}$ are rejected by the implementation. This is a numerical guard, not a modification of the

mathematical stability condition. The solver returns no covariance if its convergence or finiteness checks fail.

For one echelon under unit-variance independent demand, the exact identities are Var(o) = $\alpha/(2-\alpha)$ and Var(NS) = $1/[\alpha(2-\alpha)]$ + L. Recalculation with the included state-space implementation over five gains $\{0.2, 0.5, 1.0, 1.5, 1.8\}$ and lead times $\{0, 1, 2, 5, 10\}$ gives maximum absolute error $8.9 \times 10^{-15}$ across the 50 variance checks. These identities imply the own-cost minimizing gain $\alpha = 1$ for a single stage receiving independent unit-variance demand, not for an arbitrary upstream demand process.

A central finite-difference check of Lemma 1 uses 19 gains uniformly spaced from 0.1 to 1.9 and 100 frequencies from 0.01 to $\pi$, with derivative step $10^{-6}$. Across 1,900 evaluations, the maximum error divided by max(1, |analytic derivative|) is $5.1 \times 10^{-9}$. An additional 100 four-stage configurations use gains in [0.2, 1.8], $\rho$ in [0, 0.8], and derivative step $10^{-5}$. All computed upstream cost derivatives are positive, with minimum 0.0725. These finite checks support implementation consistency within the sampled region; the proof establishes the general statement.

### A.3 Restricted best-response reference

In the unrationed proportional model, a stage's cost depends on its own gain and downstream gains, but not on upstream gains. A forward scalar minimization therefore provides a restricted best-response reference. Using the stationary covariance objective at N = 4, L = 2, $\rho$ = 0.6 gives [1.644278, 1.131722, 1.023101, 1.003926] and total cost 43.890136. This numerical calculation does not prove uniqueness of every scalar best response or convergence of a learning algorithm. It is not a coordinated optimum and is not used as a denominator for an inefficiency ratio.

## B Implementation and reproducibility

### B.1 Environment and learning configuration

Table B1. Training and evaluation settings recovered from the implementation.

| **Setting** | **Value** |
|---|---|
| Demand mean / innovation SD | 100 / 1 |
| Lead time / quadratic weight | 2 periods / 1 per echelon |
| Parallel environments / rollout | 64 / 128 periods |
| Iterations / transitions per run | 200 / 1,638,400 joint transitions |
| Optimizer / learning rate | Adam / 0.003 |
| Discount / GAE parameter | 0.99 / 0.95 |
| PPO clip / epochs / minibatches | 0.2 / 6 / 4 |
| Minibatch size | 2,048 transitions |
| Critic / value-loss coefficient | 64–64 tanh / 0.5 |
| Entropy coefficient / initial log SD | 0.02 / $-1.6$ |
| Minimum action SD | 0.10 |
| Reward scaling / lower clip | Divide cost by 10 / $-50$ |
| Observation clip / actor mean clip | $[-20, 20]$ / $[-60, 60]$ |
| Gradient norm clip | 0.5 for each actor–critic pair |
| Numerical precision | Float64 |
| Evaluation horizon / burn-in | 10,000 / 1,500 periods |
| Evaluation seeds | 99; also 1234 and 7 where recorded |

The critic uses all four local features for every actor class. The one-gain actor is implemented as a four-input linear layer with three masked inputs; its count of one in Table 1 refers to its active mean coefficient. The residual neural actor has 4,545 mean-network parameters: $(4 \times 64 + 64) + (64 \times 64 + 64) + (64 + 1)$. Each actor adds one learned log-standard-deviation parameter, and each critic has 4,545 parameters.

At each update, advantages are standardized separately for each agent. The minimized loss is the PPO actor loss plus 0.5 times the critic's mean squared error, minus 0.02 times the action-distribution entropy. The implementation draws one permutation of rollout transitions per agent and reuses that order across the six update epochs. Critic targets are the estimated advantages plus the old value estimates. The environment is continuing, so the final rollout value is bootstrapped without a terminal-state reset.

The mean policy begins at gain one. Initial exploration standard deviation is $\exp(-1.6)$, approximately 0.202, and the learned log standard deviation is clamped below at $\log(0.10)$. The floor is not a fixed exploration schedule. The scalar saved in each result record is taken from actor 1; it should not be interpreted as an average across all agents.

### B.2 Experiment inventory

Table B2. Complete inventory of saved runs. All configurations use L = 2 and 200 training iterations.

| Configuration | Actors | Seeds per actor | Records |
|---|---|---|---|
| N = 4, $\rho = 0.6$ | All three | 8 | 24 |
| N = 4, $\rho = 0$ | All three | 3 | 9 |
| N = 4, $\rho = 0.3$ and 0.8 | One gain, affine | 3 per regime | 12 |
| N = 2, $\rho = 0.6$ | One gain, affine | 3 | 6 |
| N = 6, $\rho = 0.6$ | One gain, affine | 4 | 8 |
| N = 4, $\rho = 0.6$ ablation | Affine, demand masked | 8 | 8 |
| Total | — | — | 67 |

The training entry point is multiseed.py, with architecture arguments lin1, affine, or mlpres, followed by the inclusive seed bounds and iteration count. Environment variables RHO, NECH, and LEAD select demand autocorrelation, chain length, and common lead time. CFG labels a configuration, and RES selects its JSON result file. The script appends results, so independent reruns should use fresh output filenames to avoid duplicate records. Its default environment is the linear, quadratic-cost setting used in this paper.

The files ippo.py and tenv.py define training and the torch environment; env.py contains the NumPy simulator; cascade.py computes analytical stationary variances. Main summaries are recomputed from the cost field in each saved record. Additional evaluation analysis uses cost_by_eval_seed only where that field is present. The original training seeds are distinct from the demand-evaluation seeds, and training seed s initializes the dedicated demand generator with s + 1.

The reported PPO results are reanalyses of the saved training records. The deterministic analytical checks in Appendix A were recomputed separately. Training trajectories, actor checkpoints, hardware provenance, runtime measurements, and a dependency lockfile are necessary additions for an independently reproducible training release; statistical reconstruction of saved outcomes should not be confused with a fresh training replication.

## C Statistical details and individual outcomes

### C.1 Main comparison sensitivity analyses

Table C1. Two-sided p-values for the main setting, before multiplicity correction. The paired tests are the primary analysis.

| Comparison | Paired sign flip | Welch | Exact MWU | Exact unpaired permutation |
|---|---|---|---|---|
| Affine vs One gain | 0.007812 | 1.14e-08 | 0.000155 | 0.000155 |
| Residual MLP vs One gain | 0.007812 | 2.87e-07 | 0.000155 | 0.000155 |
| Residual MLP vs Affine | 0.359375 | 0.435 | 0.573737 | 0.446620 |

| Comparison | Paired sign flip | Welch | Exact MWU | Exact unpaired permutation |
|---|---|---|---|---|

The unpaired exact permutation test enumerates all 12,870 allocations of 16 observations into groups of eight and uses the absolute mean difference as its statistic. For the two richer-versus-one-gain comparisons, complete separation gives the same two-sided p-value as the exact Mann–Whitney test, 2/12,870. These sensitivity analyses assume exchangeability across groups and do not exploit matched training-demand seeds. The primary paired sign-flip test instead enumerates $2^8$ assignments within seed pairs.

Paired bootstrap intervals resample seed indices with replacement, applying the same indices to both classes. For percentage gaps, class means are recomputed within each resample before taking their ratio. The analysis uses NumPy generator seed 20260916 and 100,000 replicates per comparison. Small variations in the last reported decimal of bootstrap endpoints can arise from the order in which resampling draws are allocated. No normality test is used to declare eight observations normally distributed.

### C.2 Individual main-setting costs

Table C2. All main-setting costs, with no exclusions.

| Training seed | One gain | Affine | Residual MLP |
|---|---|---|---|
| 1 | 41.222908 | 53.520088 | 54.355193 |
| 2 | 40.764521 | 53.151964 | 53.243522 |
| 3 | 40.609784 | 56.742439 | 55.559046 |
| 4 | 40.610616 | 55.224889 | 60.870235 |
| 5 | 40.559308 | 57.640591 | 57.750718 |
| 6 | 40.235829 | 55.292558 | 54.415050 |
| 7 | 40.837565 | 55.057086 | 56.661985 |
| 8 | 40.592287 | 56.402598 | 56.694048 |

All three classes use the common evaluation demand seed 99 in this table. Repeated evaluation seeds are observations of the same policy, not new training replicates. The auxiliary results in Tables 3 and 4 and their associated intervals should be read as exploratory summaries of the observed runs; resampling cannot compensate for limited coverage of training randomness.

### C.3 Evaluation-seed sensitivity

Table C3. Evaluation-seed sensitivity for training seeds 5–8 only, with three evaluation seeds per policy.

| Actor | Mean of policy means $\pm$ SD | Between / within variance |
|---|---|---|
| One gain | $40.449 \pm 0.247$ | 6.83 |
| Affine | $55.951 \pm 1.181$ | 77.38 |
| Residual MLP | $56.241 \pm 1.405$ | 128.88 |

Each entry averages seeds 99, 1234, and 7 for a fixed trained policy before summarizing across the four training seeds. The variance ratio divides the sample variance of these four policy means by the average sample variance across the three evaluation seeds within each policy. It is a descriptive ratio, not an unbiased decomposition into latent variance components. Consequently, it is used only to assess the relative scale of the observed sources of variation.

Table C4. Mean evaluation cost per echelon at N = 4, L = 2, $\rho = 0.6$, training seeds 5–8. Echelon 1 faces exogenous customer demand.

| Echelon | One gain | Affine | Change | Residual MLP | Change |
|---|---|---|---|---|---|
| 1 | 8.462 | 7.436 | -12.1% | 7.663 | -9.4% |
| 2 | 10.641 | 15.879 | +49.2% | 14.074 | +32.3% |
| 3 | 10.894 | 16.808 | +54.3% | 16.661 | +52.9% |
| 4 | 10.560 | 15.975 | +51.3% | 17.982 | +70.3% |
| Total | 40.556 | 56.098 | +38.3% | 56.380 | +39.0% |