How forecast errors propagate into Sales and Operations Planning decisions: A controlled study of RL robustness

Akram Badreddine LAISSAOUI[1,2][0009-0009-6901-9475], Taha ARBAOUI[1][0000-0001-8984-2375], Anne-Laure LADIER[1][0000-0002-4201-4430], and Khaled HADJ-HAMOU[1][0000-0002-1223-3392]

[1] INSA Lyon, Université Lumière Lyon 2, Universite Claude Bernard Lyon 1, Université Jean Monnet Saint-Etienne, DISP UR4570, Bât. Léonard de Vinci, 21 av. Jean Capelle, 69621 Villeurbanne, France.
[2] Direction Supply Chain, Renault Group, Technocentre, 1 avenue du Golf, 78280 Guyancourt, France.

Abstract. Forecasts rarely match reality, and their imperfections propagate into downstream planning decisions. We study how demand forecast errors shape the quality and robustness of Reinforcement Learning (RL) policies for Sales & Operations Planning (S&OP). Using an S&OP simulator previously calibrated and validated with industrial data, we introduce a controlled perturbation framework that injects realistic forecast errors into the demand signal and passes the resulting forecasts to the RL agent. Instead of training forecasting models to optimality, we simulate distinct error regimes observed in practice to isolate causal effects on downstream decisions. We examine four classes of perturbations: (i) Gaussian errors with tunable bias and volatility to reflect systematic miscalibration and market instability; (ii) shock events to model missed risks and opportunities; (iii) seasonally poor forecasting, where specific months are harder to predict; and (iv) regime switching, where markets alternate between stable and volatile states. We quantify impacts on S&OP performance using key performance indicators, and on the RL-performance in terms of learning efficiency and speed. The framework quantifies how specific error patterns affect RL robustness under adverse information, and builds practitioner confidence in RL deployments. We provide an experimental protocol, parameterized scenarios, and testing procedures to support fair comparisons and managerial insights.

Keywords: Sales & Operations Planning · Reinforcement Learning · Forecast error modeling · Robustness · Stress testing · Simulated Machine Learning.

1 Introduction

Sales and Operations Planning (S&OP) is the cross-functional, medium-term process that aligns demand, supply, and financial plans to balance service, cost, and inventory at enterprise scale. Systematic reviews consistently document the relevance of S&OP for firm performance, implementation maturity, and integration challenges, while calling for stronger empirical and theory-informed evidence on how S&OP designs shape outcomes [20,11]. Classic practitioner accounts likewise emphasize the role of S&OP as the organizational "bridge" between demand and supply, and trace its diffusion across industries since the early 2000s [16,13].

Within S&OP, the demand forecast is a key input to downstream optimization and decision rules, affecting production, procurement, inventory, and service performance. Operations literature shows how forecast inaccuracies propagate through replenishment and capacity decisions, amplifying variability (bullwhip effect), degrading service, and increasing costs [14,15]. Further, safety stock placement and base-stock control depend on the forecast error process, forecast evolution, and non-stationarity of demand [7,19]. However, existing literature focuses on forecast accuracy metrics, e.g. Mean Absolute Percentage Error(MAPE) or Root Mean Square Deviation (RMSE), without examining how error structures affect downstream RL policy performance.

Forecast processes often miss real-world aspects: biases, seasonality, shocks (unanticipated risks/opportunities), and regime changes from stable to volatile markets. Markov-switching and related regime-change models have long provided a formal lens to describe such non-stationarities [8,9], but they can confuse downstream optimizers. In practice, planners frequently layer judgmental adjustments onto statistical forecasts, sometimes improving accuracy, but often introducing bias and instability that undermine planning quality [5]. As a result, production control managers may lose confidence not only in forecasting models but also in the optimization or planning policies that consume their outputs [6].

The Predict-then-Optimize (PtO) literature trains predictors to minimize decision loss rather than prediction loss [4,1]. However, PtO assumes the decision model is a known optimization problem. When the decision layer is an RL agent, common in complex S&OP environments [2,18], we lack systematic understanding of how forecast error characteristics propagate through learned policies. This gap is critical because RL policies may exhibit different sensitivities to input perturbations than classical optimizers.

We introduce a controlled perturbation framework for S&OP in which we simulate different forecasting models by injecting parameterized errors into the demand signal and pass the resulting forecasts to an RL-based decision layer that outputs production and supply plans [14]. Rather than training best-in-class forecasters, we adopt Simulated ML to isolate causal effects of forecast patterns on downstream planning quality [3]. Simulated ML is a methodology recently proposed to study PtO systems by simulating predictions at prescribed performance/behavioral levels. Unlike PtO approaches that train predictors, our framework fixes (simulates) the forecasting behavior and evaluates its downstream impact on decision-making. We design four controlled patterns that mir-

ror industrial cases: (i) *Gaussian errors* with tunable bias (μ) and volatility (σ); (ii) *shocks* representing missed risks/opportunities; (iii) *seasonally poor forecasting* for specific months; and (iv) *regime changes* from stable to volatile markets. This allows us to stress-test RL policy robustness, examine adaptation under adverse information, and build confidence in deploying RL in S&OP. The evaluation suite includes cost and service outcomes and plan-stability metrics. In this work, robustness is treated as an object of measurement rather than a training mechanism.

This paper makes two contributions. First, a simulated ML-based controlled perturbation methodology that decouples error patterns from forecasting model selection, thereby enabling analysis of how interpretable forecast error regimes (bias, volatility, shocks, regime changes) affect RL-based S&OP decisions. Next, an empirical quantification of RL policy robustness across forecast error regimes using metrics covering convergence speed, steady-state performance, and stability, which in turn provides insights into the reliability of RL policies under different error conditions.

Section 2 reviews related work on S&OP, forecasting for operations, PtO, Simulated ML, and RL in planning. Section 3 describes the S&OP environment and RL policy. The forecast-error experimental protocol and metrics are detailed in section 4 with and results reports, discussions and managerial insights. Section 5 concludes.

2 Related works

S&OP is widely recognized as a cross-functional process to reconcile demand, supply, and finance; systematic reviews synthesize over 250 publications [20], highlighting the performance impact of S&OP, integration patterns, and the need for stronger theoretical foundations and empirical designs [11]. Early practitioner work chronicled S&OP's diffusion and documented performance benefits when the process is institutionalized [13]. Seminal work on the bullwhip effect explains how distorted or noisy demand information amplifies variability as it flows upstream, leading to excess costs and poor service [14,15]. Operations models further show that optimal inventory positioning and safety stocks are sensitive to errors in the the forecast process and its evolution over the planning horizon [7,19]. In practice, firms frequently adjust statistical forecasts; while judgment can encode local knowledge (e.g., promotions, events), it can also introduce systematic biases that reduce planning quality if unmanaged [5,6].

The PtO paradigm integrates supervised learning with downstream optimization. Rather than minimizing prediction error, Smart PtO (SPO) trains predictors to minimize decision loss by exploiting the structure of the optimization problem (e.g., via SPO+) [4]. In parallel, prescriptive analytics studies conditional stochastic optimization with auxiliary data, offering asymptotic guarantees and practical implementations in inventory and pricing. These streams typically optimize the predictor. By contrast, our study fixes (simulates) the

predictor's behavior and analyzes how forecast error regimes affect downstream decisions – an orthogonal but complementary perspective [1].

Simulated ML has been proposed as an alternative methodology to train or validate real predictive models: simulate predictions at prescribed performance and behavioral levels, and evaluate the end-to-end PtO system under controlled conditions. Recent work [3] applies this idea to robust personnel rostering, examining trade-offs between classifier performance and rostering robustness without training classifiers on data. We adopt and extend this idea to continuous-valued demand forecasting for S&OP, to RL decision policies, where robustness to distribution shift is central. RL has shown promise for inventory and supply-chain decisions, with evidence of strong performance and some robustness to inaccuracy; surveys chart the algorithmic design space for inventory control [18,2]. However, robustness under forecast inaccuracy is underexplored. Robust Markov Decision Process (MDP) theory studies policies that hedge against transition/model uncertainty, providing dynamic-programming algorithms and tractable uncertainty sets (e.g., s-rectangular, KL/entropy, ellipsoidal) [10,17]. In learning systems, domain randomization exposes policies to broad parametric variability during training to encourage generalization under distribution shift [21]. Our experiments provides controlled robustness tests for the RL policy under forecast uncertainty.

Most S&OP-related forecasting studies focus on building better forecasters; far fewer analyze how distinct forecast error patterns propagate into downstream decision quality and user-perceived reliability, particularly when the decision layer is an RL policy rather than a deterministic optimizer. We fill this gap by importing Simulated ML to S&OP forecasting and extending it to RL-driven decision making, thereby assessing its robustness in a controlled, industry-plausible setting.

3 S&OP simulator

This work builds upon the RL-based multi-agent S&OP base simulator introduced in [12]. The environment models the monthly S&OP process with interacting agents, and an RL policy that learns and captures the trade-offs balancing service levels against inventory holding costs and production changeover penalties, under uncertain demand. Our contribution extends this base simulator by introducing a *Forecaster agent* with simulated-ML capabilities that generate controlled forecast error regimes, enabling analysis of how forecast errors propagate into RL-driven planning decisions. This section summarizes the multi-agent architecture, RL formulation, state/action spaces, reward function and *Forecaster agent* that enables the experiments in Section 4.

3.1 Multi-agent architecture (overview)

The base simulator manages three agents making decisions on a monthly basis:

Main agent (the planner). The central decision-maker that sets production and multi-period supply plans subject to capacity, lead-time and order adjustment constraints. Production is bounded by available capacity $0 \leq p_t \leq C_t$; supply planning orders over a rolling horizon H, with a share of orders "firm" and the remainder adjustable within a contractual interval (e.g., $\pm 20\%$). This agent implements the RL policy described in Section 3.2.

Market agent. Generates stochastic demand using a multiplicative structure $D_t = T_t \cdot S_t \cdot R_t$ with trend T_t, seasonality S_t, and a residual R_t that can be white noise or autoregressive. Realized demand may be further modulated by macro factors (e.g., delivery performance, inflation indices), preserving the causal link between service and future demand, common in S&OP contexts.

Supplier agents. Execute deliveries subject to lead times L_t, costs, and reliability; orders can be firm or adjustable within a band, aligning with practical supplier governance. Multiple suppliers with different lead times can coexist, each with distinct cost structures.

Each monthly `.step()` proceeds as follows: macro variables update via a mean-reverting process (inflation, economic indices); the *Market agent* draws realized demand; the *Main agent* observes state s_t (including forecast from *Forecaster agent*) and selects action a_t; production executes and Suppliers deliver; inventory, backlog and costs are recorded, reward is computed; state transitions to next one s_{t+1}.

The simulator is designed to be modular and extensible for new agents, constraints, and forecast behaviors.

3.2 RL formulation: states, actions, and SAC policy

The S&OP problem is cast as a continuous-state, continuous-action Markov Decision Process $M = \langle \mathcal{S}, \mathcal{A}, \pi, r, \gamma \rangle$ at a monthly decision period.

State space $s_t \in \mathcal{S}$. *Main agent* tracks a rolling window of operational histories: $s_t = \big[\text{demand}_{t-k:t},\ \text{production}_{t-k:t},\ \text{shipments}_{t-k:t},\ \text{inventory}_{t-k:t},\ \text{sup\_inventory}_{t-k:t},\ \text{sup\_orders}_{t-k:t}\big]$. This history enables the policy to infer demand patterns, capacity use, supply availability, and service/inventory health. In the base simulator, forecasting is implicit: the policy infers future demand from the history and environment signals.

Action state $a_t \in \mathcal{A}$. A composite vector containing a scalar production decision for period t plus a vector of supply orders for the planning horizon H: $a_t = \big[a_t^{\text{prod}} \in \mathbb{R},\ a_{t \to t+L:t+L+H-1}^{\text{sup}} \in \mathbb{R}^H\big]$. Actions are sampled autoregressively (production first, then horizon orders) and normalized $[-1, 1]$ space, then mapped to feasible operational bounds (capacity, supplier limits).

The base simulator uses autoregressive sampling: production is sampled first, then horizon orders are sampled conditionally on the production decision to reflect their interdependence in S&OP planning practice. We employ Soft Actor-Critic (SAC) policy π_θ, maximum-entropy RL algorithm suited for S&OP as it

handles continuous spaces and supports efficient exploration. These choices improve exploration under long lead times and delayed effects typical of S&OP. We use standard SAC hyperparameters commonly adopted in prior work, ensuring reproducibility while keeping the focus on forecast perturbation effects.

3.3 Reward design: service, stability, supply balance, and cost

The reward aggregates normalized S&OP metrics with a geometric mean structure that penalizes collapse of any single dimension while emphasizing service:

- Value index (service level) $m_{\text{value}} = \frac{\text{shipped}_t}{\text{demand}_t}$.
- Inventory alignment m_{inv}, penalizing deviation from business or analytical target.
- Production stability $m_{\text{stable}} = \frac{1}{1+\text{CV}}$, where CV is a coefficient of variation over a window.
- Supply balance m_{supply}, penalizing undersupply more heavily than oversupply to protect continuity of operations.
- Cost components including production cost, inventory holding, backlog penality, expedite costs and supplier inventory holding, are tracked explicitly and embedded into the reward.

The final per-period reward takes the form: $r_t = m_{\text{value}} \cdot \exp\left(\frac{1}{N}\sum_i \log m_i\right)$, this geometric mean formulation deters extreme trade-offs (e.g., perfect service with unstable production or unrealistic inventories) by insuring that reward is zero if any metric reaches zero.

3.4 Explicit Forecaster with Simulated-ML capabilities

While the base simulator lets the RL policy implicitly infer demand from history, our key contribution is to introduce an explicit *Forecaster agent* that the *Main agent* instantiates and calls each month to generate a demand forecast vector for the S&OP horizon. This separation reflects real S&OP practice where forecasting and planning are performed by distinct processes and systems. We then equip this Forecaster with Simulated-ML functionality to inject the controlled forecast behaviors used in Section 4:

Forecaster operation. At each decision time t, the Forecaster produces $\hat{\mathbf{D}}_t = \{\hat{D}_{t+1|t}, \ldots, \hat{D}_{t+H|t}\}$ by applying a scenario-specific perturbation operator P to the *Market agent*'s latent demand signal. The *Main agent* consumes $\hat{\mathbf{D}}_t$ as part of the observation s_t.

Simulated-ML Forecast regimes. We instantiate four families of forecast regimes (perturbations): (i) Gaussian errors with tunable bias and volatility to capture miscalibration and market instability; (ii) shocks (Laplace-tied errors) to model missed risks/opportunities, not frequent and large forecast errors that are not captured by Gaussian models; (iii) seasonally poor forecasting where specific months present higher forecast variance; and (iv)

regime switching from stable to volatile forecasting quality, reflecting market transitions. These controlled perturbations allow us to isolate how forecast error patterns influence learning speed, stability, and steady-state outcomes.

State extension. We extend the RL state to include the forecast vector. This matches S&OP practice where planners act on a rolling demand projection, and it lets the policy learn forecast-aware behaviors (e.g., building buffers under noisy forecasts, smoothing production under biased forecasts). This design choice also enables controlled and interpretable manipulation of forecast signals, allowing us to isolate their causal impact on downstream RL behavior.

This Forecaster extension preserves the base simulator's agent topology and monthly process, adds no changes to the environment's material balance or supplier mechanics, and remains backward-compatible: if the Forecaster is disabled, the system reverts to the original implicit-forecast setting described in [12].

4 Simulated demand forecasting

This section details our Simulated-ML approach to generate controlled, realistic forecast inputs fore S&OP and to study their downstream effects on an RL decision layer. In each scenario, we perturb the "true" demand produced by the *Market agent* to emulate classes of forecasting behaviors observed in practice. Such imperfections are central to S&OP reality and are known to propagate through replenishment and capacity decisions, affecting service, inventory, cost, and plan stability (e.g., via safety-stock policies).

Our design follows the Simulated ML methodology: rather than fitting the "best" forecasting model, we simulate forecast outputs that achieve targeted error characteristics, then evaluate end-to-end planning performance. This isolates causal links between forecast error patterns and downstream plan quality in a Predict-then-Optimize (PtO) setting. The approach complements PtO training (SPO/SPO+) by answering a different question: how do specific forecast misspecifications affect planning policies – here, an RL policy – under controlled conditions?

4.1 Experimental protocol

For each run, we take the base "real" demand D_t from the Market agent (representing the true realized market). We modify the Forecaster module so that, instead of training a predictive model, it applies a scenario-specific perturbation operator $\mathcal{P}$ to produce a forecast vector $\hat{\mathbf{D}}_t = \{\hat{D}_{t+1|t}, \ldots, \hat{D}_{t+H|t}\}$ from $\mathbf{D}_{t:t+H}$. This $\hat{\mathbf{D}}_t$ is injected into the S&OP RL policy as part of the observation. Although implemented as perturbations on forecasts, the resulting mismatch is equivalent to real S&OP settings where forecasts fail to anticipate realized demand shocks. Modeling forecast errors explicitly allows us to decouple forecast imperfection from intrinsic demand variability, which would otherwise be confounded under

pure demand noise. The rest of the environment remains unchanged, so performance differences stem from forecast patterns. The values of the environment parameters are taken from [12], for the forecast horizon $H = 6$. This reflects S&OP practice in which planning processes run on rolling forecasts of varying quality. We deliberately keep a fixed RL policy across experiments to isolate the effect of forecast quality; comparisons with alternative policies are left for future work.

Two guiding questions motivate the experiments:

1. Does there exist a class of simulated forecasters (i.e., with certain error statistics) that yields better S&OP outcomes for the RL policy than a conventional "best RMSE or MAPE" forecaster? This speaks to PtO insights that decision loss may diverge from prediction loss.
2. How robust is the RL policy to biased or unstable forecasts? Does it adapt (e.g., by implicitly holding buffers or smoothing production) to preserve service under misspecification? This links to robust-MDP, robustness literature and to domain-randomization ideas in learning under shift.

4.2 Evaluation methodology and metrics

For each scenario we execute 100 epochs of RL training/validation. After each epoch, we log the validation reward trajectory R_e. Because RL rewards are noisy, we smooth R_e with a moving average (window w, constant across experiments) before computing summary statistics, as standard practice for RL learning curves.

The metrics are computed on the smoothed curve unless noted, with small windows of $k = 5$ epochs.

Max reward $R_{\max}$: best observed performance; proxy for peak attainable plan quality under the forecaster.

Final reward $R_{\mathbf{final}}$: mean of the last k epochs; proxy for steady-state performance.

Time-to-80% t_{80}: first epoch reaching $0.8 \times R_{\max}$; proxy for time-to-value (how fast planners can trust the policy).

$\sigma_{\mathbf{tail}}$: standard deviation of the last k epochs (unsmoothed); proxy for plan stability / sensitivity to noise.

It is important to note that service and cost KPIs are downstream of reward; the reward function aggregates production, holding, and backlog penalties common to S&OP. A faster t_{80} means a quicker stabilization of plans; a lower σ_{tail} means fewer re-plans, i.e. an improved execution.

4.3 Scenario A: Gaussian forecaster – bias and volatility grid

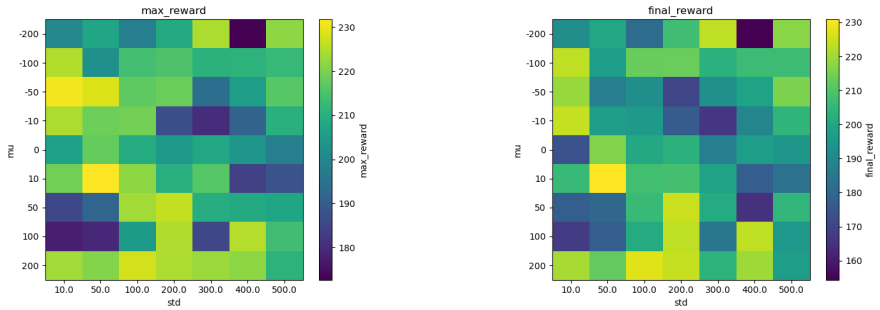
This scenario emulates systematic bias (forecast consistently high or low) and market volatility (unstable forecasts) that commonly arise from miscalibration,

data issues, or turbulence. In S&OP, bias is monitored as forecast bias, and volatility often manifests in higher MAPE/variance and unstable consensus plans.

We test multiple values of absolute bias (if $\mu = 100$ we have a forecast bias of +100 products) and volatility as follows:

$$\hat{D}_{t+h|t} = \max\{0,\ D_{t+h} + e_{t,h}\}, \quad e_{t,h} \sim \mathcal{N}(\mu, \sigma^2)$$
$$\text{with } \mu \in \{-200, -100, -50, -10, 0, 10, 50, 100, 200\}$$
$$\text{and } \sigma \in \{10, 50, 100, 200, 300, 400, 500\}$$

In Figure 1 we organize the outputs of the experiments in heatmaps of $R_{\max}$, R_{final}, t_{80},σ_{tail} across the (μ, σ) grid.



(a) Max reward

(b) Final reward

(c) t_{80}

(d) Stability

Fig. 1: Results of the experiment for Scenario A: Gaussian forecaster.

The results of the Gaussian experiment reveal a clear asymmetric impact of forecast bias on S&OP decision quality. As shown in Figures 1a and 1b, under-forecasting ($\mu < 0$) consistently leads to poorer RL performance due to the sharp increase in stockouts and backlog, which carry the highest penalties in our reward structure. Conversely, moderate over-forecasting ($\mu > 0$) generates excess inventory but avoids service failures, resulting in higher overall rewards. This mirrors well-established observations: S&OP processes typically treat lost

sales and service degradation as more damaging than holding surplus inventory, especially in markets with strict service level agreements or high customer-service expectations. Therefore, the RL agent trained under this cost asymmetry naturally favors behaviors aligned with service protection rather than inventory minimization.

Figure 1 also shows the surprisingly strong performance of the RL policy under moderate forecast variability (e.g., $\sigma = 200$). At this level, we observe R_{final} values statistically equivalent to low-noise conditions. A plausible explanation is that the RL agent benefits from a degree of informational uncertainty: exposure to moderately noisy forecasts encourages exploration allowing the policy to detect corrective patterns and build an internal representation of demand volatility. In contrast, near-perfect forecasts ($\sigma \approx 0$) restrict RL exploration, effectively "locking" the agent into decisions that mirror the injected deterministic forecast, even when these forecasts are systematically biased or misaligned with cost-optimal behaviors.

This suggests a forecast-certainty interval, i.e. a σ tolerance band within which additional forecast precision no longer benefits (and may even hinder) downstream optimization. Forecasts should be accurate enough to provide directional reliability, but not so rigid that the RL policy cannot learn to compensate for uncertainty. We hypothesize this occurs because moderate noise encourages exploration during training. Beyond this interval, particularly at high σ values, Figures 1a and 1b show clear performance degradation, and Figures 1c and 1d demonstrate slower, more unstable learning.

4.4 Scenario B: Shocked forecaster – missed risks and opportunities

This scenario models unanticipated events (promotions, competitor actions, supply disruptions/opportunities) that forecasters often miss: spikes or dips in demand that are not signaled early enough, hence appear as forecast errors. In S&OP, these are managed via exception processes and event planning.

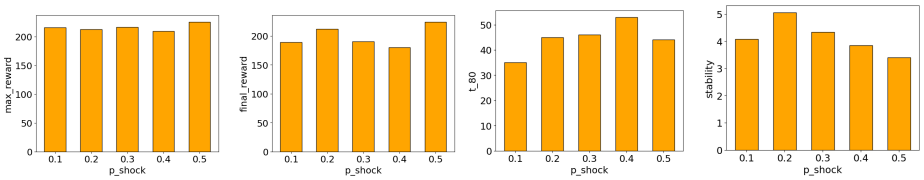
With probability p_{shock} at each forecasted period, the error is drawn from a Laplace distribution (double exponential) with zero mean and high scale b (heavy tails), otherwise from a "good" Gaussian $\mathcal{N}(0, \sigma_0^2)$:

$$e_{t,h} \sim \begin{cases} \text{Laplace}(0, b) & \text{with probability } p_{\text{shock}}, \\ \mathcal{N}(0, \sigma_0^2) & \text{with probability } 1 - p_{\text{shock}}. \end{cases}$$

We vary $p_{\text{shock}} \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ and set $b \gg \sigma_0$ to represent large, rare shocks.

Figure 2 displays the results. Empirically, steady-state performance is nearly unchanged across the whole range of p_{shock}: Max and Final rewards remain essentially flat (see Figures 2a and 2b), indicating that the RL policy absorbs shocks without material loss in value. What does change is learning efficiency: when shocks are rare, the agent reaches high performance faster (lower t_{80}) (see Figure 2c).

Interestingly, beyond $p_{\text{shock}} \approx 0.2$, Figure 2d shows that the tail variability σ_{tail} of the learning curve decreases. One interpretation is that frequent shocks



(a) Max reward (b) Final reward (c) t_{80} (d) Stability

Fig. 2: Results of the experiment for Scenario B: Shocked forecaster.

prompt the agent to internalize a hedging structure (slightly more conservative production or protective inventories and a tempered reaction to forecast swings) so the late-stage policy becomes less sensitive to individual shocks. However, we can not exclude that this reflects the RL algorithm's balance between exploration and exploitation rather than a general property of learned S&OP policies.

From an S&OP perspective, the takeaway is twofold. First, event governance matters for time-to-value: keeping major events visible during ramp-up accelerates learning, even though the deployed policy will remain robust if some shocks are missed. Second, in sustained high-shock environments, it is advisable to institutionalize modest buffers and exception hooks (e.g., tighter re-planning cadence during volatile periods) to complement the RL agent's learned hedge; this preserves service and cost discipline while maintaining planning stability.

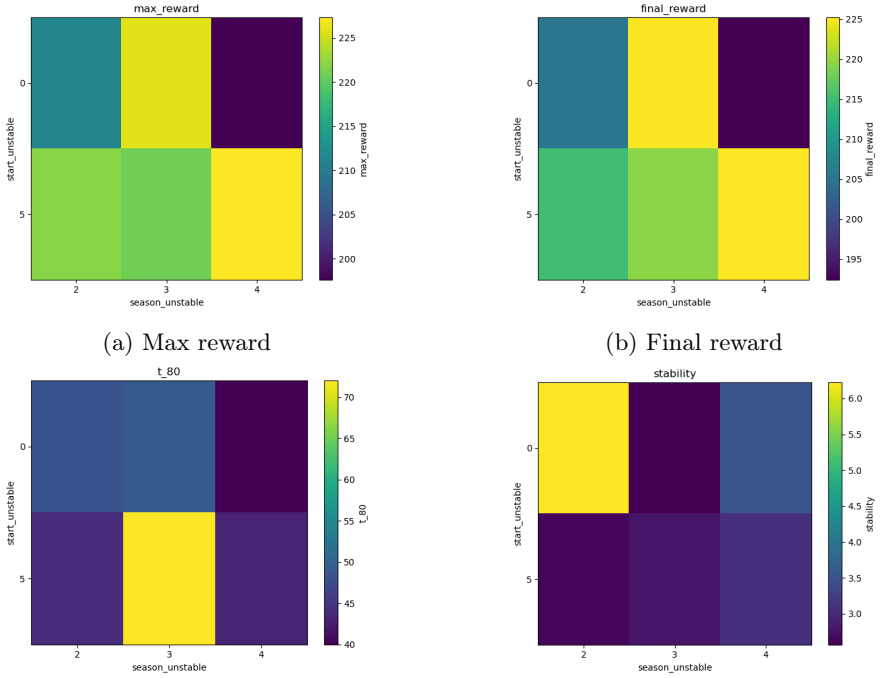
4.5 Scenario C: Seasonally poor forecasting – unstable windows

Scenario C captures periodic months (e.g., holiday peaks, regional festivals) where forecasts are systematically less reliable due to unstable effects, marketing overlays, or data sparsity. This is a pervasive S&OP phenomenon; good processes acknowledge seasonal windows of low predictability.

We partition the year into stable and unstable windows. For stable months: $e_{t,h} \sim \mathcal{N}(0, \sigma^2_{\text{st}})$ (good forecaster). For unstable months: $e_{t,h} \sim \mathcal{N}(0, \sigma^2_{\text{unst}})$, with $\sigma_{\text{unst}} \gg \sigma_{\text{st}}$. We study two placements of the unstable window, H1 (first semester) and H2 (second semester), and vary the window width w (e.g., 1–4 months) to reflect realistic seasonal concentrations.

Across all tested configurations, the results in Figure 3 show only minimal differences neither placement nor window width produced clear, consistent changes in learning speed, reward, or stability. While the second-semester scenario shows slightly more performance and stability, the effect is marginal and not strong enough to draw meaningful operational conclusions. Overall, the figures reveal no identifiable zones of inefficiency; the RL agent performs almost identically across all seasonal-instability settings. The key conclusion is that the RL policy remains robust: localized, recurring periods of poor forecasting do not meaningfully harm convergence or steady-state performance.

From a managerial standpoint, this robustness implies that seasonal forecast noise is not a critical risk for the RL-enabled S&OP process. Leaders can



(a) Max reward

(b) Final reward

(c) t_{80}

(d) Stability

Fig. 3: Results of the experiment for Scenario C: Seasonally poor forecasting.

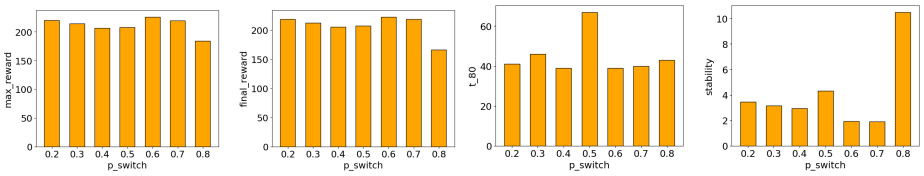
avoid over-investing in hyper-specific seasonal forecast improvements and instead focus analytical resources on larger drivers of variability such as promotional events, supply disruptions, or strategic regime shifts. Standard seasonal governance (light buffers, basic variance monitoring, and typical re-planning cadence) remains sufficient, as the planning system shows no sensitivity spikes tied to seasonal inaccuracies.

4.6 Scenario D: Regime-switching forecaster – stable→volatile

Scenario D emulates market regime changes from stable to volatile phases (e.g., macro shifts, competitive intensity). Forecasting quality often degrades after a switch; we examine whether the RL policy maintains performance and how quickly it adapts. Regime-switching time-series models formalize such behavior.

Over a 25-year horizon (300 months) simulation, we start with a good Gaussian forecaster $\mathcal{N}(0, \sigma^2_{\text{low}})$, then we switch at month $\tau = \lfloor p_{\text{switch}}\% \times 300 \rfloor$ to a poor forecaster $\mathcal{N}(0, \sigma^2_{\text{high}})$, $\sigma_{\text{high}} \gg \sigma_{\text{low}}$. We vary $p_{\text{switch}} \in \{10, 20, \ldots, 90\}$ to move the switch point.

Figure 4 shows that overall, steady-state performance remains essentially flat across the different values of p_{switch}, indicating that the RL policy is robust to



(a) Max reward (b) Final reward (c) t_{80} (d) Stability

Fig. 4: Results of the experiment for Scenario D: Regime-switching forecaster.

when the volatility regime changes (see Figures 4a and 4b). Two patterns stand out:

- For nearly all thresholds, t_{80} clusters around ∼40 epochs; the exception is $p_{\text{switch}} = 50\%$, where the horizon is split evenly between good and bad forecasting (see Figure 4c). Here, learning is significantly slower, t_{80} increases to approximately 70 epochs. This increase in learning time occurs when stable and volatile regimes are balanced in duration. This suggests that a mid-horizon regime flip creates the most challenging learning curriculum: the policy first adapts to a stable regime, then must relearn under volatility for an equally long tail. The policy requires longer to discover robust rules that perform adequately in both regimes.
- Stability tracks closely with t_{80}, showing consistent patterns across thresholds, except at $p_{\text{switch}} = 80\%$ where we observe marked instability late in training (see Figure 4d). A plausible interpretation is that the policy, after extended exposure to the stable conditions, requires multiple strategy adjustments to correct the late 20% of volatile periods – hence the observed oscillations.

For S&OP governance, the results imply that precisely anticipating the timing of a volatility shift is not required to keep performance steady; the RL layer is robust to regime changes. However, two operational cautions follow:

- if indicators suggest a mid-cycle regime break (akin to $p_{\text{switch}} \approx 50\%$), expect slower time-to-value : consider shorter re-planning cadence and temporary buffers to bridge the adaptation period;
- when a late switch is likely (akin to $p_{\text{switch}} = 80\%$), prepare for plan variability near the end of the horizon : use guardrails on production swings and monitor service-risk dashboards to keep execution stable while the policy settles.

5 Conclusion

We proposed a Simulated-ML framework to stress-test a RL-enabled S&OP pipeline by injecting controlled, industry-plausible forecast errors, bias and volatility, shocks, seasonal weaknesses, and regime switches, directly into the forecaster; then observing their effect on learning speed, stability, and steady-state

value. This isolates how error patterns propagate into downstream plans and complements Predict-then-Optimize by focusing on decision robustness.

Across scenarios, results are consistent:

- with Gaussian errors, bias hurts asymmetrically (under-forecasting degrades performance more than over-forecasting with an equal magnitude, and moderate forecast variance can help exploration;
- with shocks, steady-state performance holds flat while learning is faster when shocks are rare and the policy learns a hedge when shocks are frequent;
- with seasonal forecast degradation, 1st vs. 2nd semester placement and window width have negligible impact ;
- under regime switches, steady-state performance is stable across switch timings, with slower learning at mid-horizon flips and late-phase instability when the switch occurs very late.

The RL policy remains robust in steady-state performance, with differences mainly in learning dynamics (time-to-value and tail stability).

These results suggest several directions for the S&OP process. First, define tolerance intervals for forecast quality rather than maximizing accuracy; minor over-forecast buffers are often preferable to under-forecast risk. Second, during ramp-up and around suspected regime breaks, rely on light buffers, shorter re-planning cadences, and guardrails on production swings to smooth convergence; otherwise, standard seasonal governance suffices. Third, prioritize event governance and detection of large drivers (promotions, disruptions) over micro-tuning seasonal effects, since the RL layer absorbs routine seasonal noise. An explicit analysis of feature importance and attribution methods is left as future work to better quantify the contribution of forecast information to the learned policy.

We identify three extensions. First, we examine multi-SKU environments to test whether error propagation differs with product dependencies. Second, we perform empirical calibration using partner data to validate that our synthetic error regimes reflect real forecast behavior (empirical bias/variance distributions by month, shock magnitudes and frequencies, regime dwell times from historical logs). Third, we conduct a comparison against robust-MDP training and domain randomization and risk-sensitive objectives (e.g., conditional value at risk on backlog/instability). We also envision including ablations on uncertainty features, and a human-in-the-loop layer that turns diagnostics (bias/variance) into actionable playbooks for S&OP governance.

References

1. Bertsimas, D., Kallus, N.: From predictive to prescriptive analytics. Management Science **66**(3), 1025–1044 (2020)
2. Boute, R.N., Gijsbrechts, J., Van Jaarsveld, W., Vanvuchelen, N.: Deep reinforcement learning for inventory control: A roadmap. European Journal of Operational Research **298**(2), 401–412 (2022)

3. Doneda, M., Smet, P., Carello, G., Lanzarone, E., Vanden Berghe, G.: Robust personnel rostering: How accurate should absenteeism predictions be? Journal of Scheduling pp. 1–16 (2025)
4. Elmachtoub, A.N., Grigas, P.: Smart "predict, then optimize". Management Science **68**(1), 9–26 (2022)
5. Fildes, R., Goodwin, P.: Against your better judgment? how organizations can improve their use of management judgment in forecasting. Interfaces **37**(6), 570–576 (2007)
6. Fildes, R., Goodwin, P., et al.: Good and bad judgement in forecasting: Lessons from four companies. Foresight **8**(Fall), 5–10 (2007)
7. Graves, S.C., Willems, S.P.: Optimizing strategic safety stock placement in supply chains. Manufacturing & Service Operations Management **2**(1), 68–83 (2000)
8. Hamilton, J.D.: A new approach to the economic analysis of nonstationary time series and the business cycle. Econometrica: Journal of the econometric society pp. 357–384 (1989)
9. Hamilton, J.D.: Regime switching models. In: The new palgrave dictionary of economics, pp. 11421–11426. Springer (2018)
10. Iyengar, G.N.: Robust dynamic programming. Mathematics of Operations Research **30**(2), 257–280 (2005)
11. Kreuter, T., Scavarda, L.F., Thomé, A.M.T., Hellingrath, B., Seeling, M.X.: Empirical and theoretical perspectives in sales and operations planning. Review of Managerial Science **16**(2), 319–354 (2022)
12. Laissaoui, A.B., Arbaoui, T., Ladier, A.L., Benichou, A., Hadj-Hamou, K.: Reinforcement learning-based multi-agent simulation framework for smart sales and operations planning (2026), submitted to 23rd IFAC World Congress, International Federation of Automatic Control
13. Lapide, L.: Sales and operations planning part i: the process. Journal of Business Forecasting **23**(3), 17–19 (2004)
14. Lee, H.L., Padmanabhan, V., Whang, S.: The bullwhip effect in supply chains1. Sloan Management Review **38**(3), 93–102 (1997)
15. Lee, H.L., Padmanabhan, V., Whang, S.: Information distortion in a supply chain: The bullwhip effect. Management science **43**(4), 546–558 (1997)
16. Márcio Tavares Thomé, A., Felipe Scavarda, L., Suclla Fernandez, N., José Scavarda, A.: Sales and operations planning and the firm performance. International Journal of Productivity and Performance Management **61**(4), 359–381 (2012)
17. Nilim, A., El Ghaoui, L.: Robust control of markov decision processes with uncertain transition matrices. Operations Research **53**(5), 780–798 (2005)
18. Oroojlooyjadid, A., Nazari, M., Snyder, L.V., Takáč, M.: A deep q-network for the beer game: Deep reinforcement learning for inventory optimization. Manufacturing & Service Operations Management **24**(1), 285–304 (2022)
19. Schoenmeyr, T., Graves, S.C.: Strategic safety stocks in supply chains with evolving forecasts. Manufacturing & Service Operations Management **11**(4), 657–673 (2009)
20. Thomé, A.M.T., Scavarda, L.F., Fernandez, N.S., Scavarda, A.J.: Sales and operations planning: A research synthesis. International Journal of Production Economics **138**(1), 1–13 (2012)
21. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain randomization for transferring deep neural networks from simulation to the real world. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). pp. 23–30. IEEE (2017)