

A Predict-then-Correct Loop Based on Few-Shot Continuous Contextual Bandit for Demand Forecasting

Zhiwei Lei^a, Benedict Jun Ma^a and Ilya Jackson^b

^aThrust of Intelligent Transportation,

Hong Kong University of Science and Technology (Guangzhou), China;

^bCenter for Transportation and Logistics,

Massachusetts Institute of Technology, USA;

Abstract

Retail demand forecasting remains difficult when demand shifts faster than static forecasting models can be retrained, especially in early demand cycles where newly observed labels are sparse. To address this, this study aims to improve adaptive retail forecasting by proposing a predict-then-correct (PtC) framework that retains a first-stage machine learning (ML) forecast and applies a few-shot continuous contextual bandit correction policy with similar-SKUs augmentation and top- p masked updating. Across Walmart retail data and an exclusive beverage dataset, PtC delivers statistically significant reductions in MAPE, MAE, and RMSE across stable & high volume, stable & low volume, and erratic & intermittent demand patterns, improves average RMSE by 9.52% over the ML-only baseline in the ablation study, and yields lower inventory costs than base-stock, proximal policy optimization, and soft actor-critic policies under the tested lead-time settings. These findings show that online forecast correction can bridge offline demand learning and real-time retail decision-making by adapting to sparse feedback without fully retraining the base forecasting model.

Keywords: demand forecasting; predict-then-correct; contextual bandit; few-shot learning.

1 Introduction

Retail supply chains are high-dimensional, uncertain, and complex decision systems that tightly couple demand forecasting, inventory control, and multi-stage logistics operations (Cohen et al., 2022). This system-level complexity is driven primarily by three interrelated dimensions: stochastic demand, network structure, and dynamic decision-making. First, customer demand is non-stationary and volatile, often affected by exogenous market conditions, pricing mechanisms, and promotional activities (Fahimnia et al., 2025). Second, managing millions of stock-keeping units (SKUs) across multi-echelon distribution networks introduces intricate product-level relationships, such as substitution effects, complementarity, and cross-item resource sharing (Ma et al., 2016). Third, key operational decisions, including inventory allocation (Khouja and Zhou, 2019; Kim

et al., 2024), dynamic pricing (Zhong et al., 2020), and robust procurement (Wagner, 2015; Li et al., 2024), exhibit complex time-dependent characteristics (Huang et al., 2019). Consequently, inaccurate demand forecasts can propagate beyond the forecasting stage and distort downstream operational decisions, leading to severe stockouts or costly backlogs that degrade overall supply chain performance (Cohen et al., 2022).

With the rapid development of artificial intelligence (AI) and enterprise big data infrastructures, modern supply chains are actively evolving into data-driven cyber-physical systems that achieve real-time perception and dynamic closed-loop optimization (Punia and Shankar, 2022; Mandania and Oliveira, 2023). In this context, demand forecasting constitutes a fundamental analytical function for such intelligent supply chain systems. Industry evidence suggests that AI-driven forecasting frameworks can reduce prediction errors by 20% to 50%, lower holding costs by 5% to 10%, and reduce lost sales by up to 65% (McKinsey, 2022). Methodologically, this development has accelerated a shift in forecasting research from classical parametric time-series models toward supervised machine learning (ML) architectures, such as long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997), gradient boosting machines (GBM) (Friedman, 2001), XGBoost (Chen and Guestrin, 2016), and deep learning (DL) architectures such as transformer-based variants (Zhang et al., 2024; Li et al., 2024). Due to the abilities of nonlinear mapping, these data-driven approaches have demonstrated decisive algorithmic superiority over traditional statistical benchmarks in large-scale forecasting (Makridakis et al., 2022; Lei et al., 2024).

Despite these advances, offline-trained ML models remain limited when deployed in rapidly changing and non-stationary market environments. During a new demand cycle or a product launch period, market dynamics often suffer from sharp structural breaks compared to historical regimes, inducing acute data sparsity and prominent “cold-start” challenges (Petropoulos et al., 2022). Because traditional supervised pipelines rely on rigid, batch-oriented training and require massive data pools for expensive offline retraining, they cannot seamlessly ingest real-time demand feedback. In practice, the correction of forecast bias therefore remains largely reactive (Wang et al., 2024). Once a base model is trained and deployed, it operates as an open-loop predictor that lacks the capacity for real-time calibration against short-term operational fluctuations and concurrent temporal shifts. This reveals a critical methodological void: the absence of a systematic, sample-efficient mechanism engineered to explicitly *learn to correct* baseline prediction errors on the fly.

To bridge this gap, this study proposes a novel **predict-then-correct** (PtC) loop framework designed for adaptive demand calibration. Instead of treating forecast-error correction as a pas-

sive label-fitting task, we reformulate real-time forecast adjustment as a contextual bandit (CB) problem. In the proposed framework, an offline ML model first generates baseline predictions, after which an online few-shot continuous contextual bandit (FSCCB) policy dynamically fine-tunes these predictions by selecting bounded, continuous corrective actions from streaming market contexts based on immediate reward feedback. The core advantage of this architecture lies in its decoupled, closed-loop design: it preserves the structural, historical knowledge captured by the offline base model, while enabling the system with real-time, step-by-step responsiveness as market observations gradually unfold. To resolve data scarcity and the stability-plasticity dilemma inherent in early-cycle launches, the framework further integrates a feature-similar sequence data augmentation strategy and a sparse parameter update mechanism (Mazumder et al., 2021) that updates only highly sensitive policy parameters while freezing the primary structural network. Ultimately, this framework shifts the operational forecasting paradigm from static prediction to autonomous, self-calibrating intelligent control.

The main contributions of this paper are summarized as follows:

(1) We propose a two-stage PtC framework that couples AI/ML forecasting with a CB correction model. This transforms static, open-loop demand forecasting into an adaptive, closed-loop system, significantly enhancing prediction responsiveness to market fluctuations.

(2) To tackle real-time data scarcity, we develop a few-shot continuous updating strategy. A base model is pretrained on historical data for initialization, and then dynamically fine-tuned via streaming data augmentation and a selective parameter update mechanism, enabling rapid adaptation to new demand patterns while preventing overfitting and catastrophic forgetting.

The remainder of the paper is organized as follows. Related works are reviewed in the next section. Section 3 describes the details of the two-stage framework. The experimental results are discussed in Section 4. Section 5 discusses the application of PtC in the inventory management system. Conclusions are presented in Section 6.

2 Literature review

2.1 Traditional statistical methods for demand forecasting

Classical demand forecasting research first developed around extrapolative time-series models. Naive, moving-average, exponential-smoothing, and ARIMA-family models remain widely used because they are transparent, computationally efficient, and effective when demand contains sta-

ble level, trend, or seasonal components (Taylor, 2003; Nikolopoulos et al., 2011; Hyndman and Athanasopoulos, 2018). Exponential-smoothing models place greater weight on recent observations and therefore provide a simple way to adapt to gradual level or trend changes, while ARIMA-type models use autoregressive and moving-average structures to capture serial dependence after differencing. These models are especially attractive in operational environments because they are easy to implement and require limited feature engineering. Retail applications extended these ideas to volatile and skewed demand settings: early work on retailer demand forecasting identified promotions, competitor actions, weather, and holidays as important drivers (Geurts and Kelly, 1986), while later studies developed interval forecasts, hybrid SARIMAX quantitative methods, and intermittent-demand aggregation strategies for demand series with irregular arrivals or heavy-tailed errors (Taylor, 2007; Arunraj and Ahrens, 2015; Nikolopoulos et al., 2011). The strength of this stream is its clarity: the source of each forecast is usually traceable to level, seasonality, trend, or a small set of lagged terms. Its limitation is equally clear. Once the functional form has been specified, the model has limited flexibility to represent nonlinear promotion effects, changing consumer response, cross-SKU substitution, or sudden shifts in demand regimes. Thus, classical statistical models provide reliable baselines, but they are less suited to settings where the main forecasting challenge is rapid adaptation rather than stable extrapolation.

A second stream uses causal, multivariate, and econometric models to incorporate explanatory factors into demand forecasts. In retail systems, demand is rarely driven by time alone. All prices, discounts, display activities, holidays, competitive information, search intensity, weather, social-media signals, and product attributes can alter realized sales. Therefore, operations and forecasting studies have examined how external information can be integrated into forecast pipelines and how judgmental or promotional adjustments affect forecast quality (Fildes et al., 2008; Trapero et al., 2013; Steinker et al., 2017; Boone et al., 2018; Cui et al., 2018; Fildes et al., 2022). In fast-moving consumer goods and retail settings at the SKU-level, variable selection and structured econometric models have been used to address high-dimensional promotional variables, competitive effects, and time-varying marketing impacts (Ma et al., 2016; Huang et al., 2014, 2019; Ye et al., 2024). Hierarchical Bayesian and finite-mixture extensions further examine whether store-level heterogeneity improves forecast and elasticity estimation (Andrews et al., 2008). These studies make an important contribution by linking prediction to interpretable business mechanisms: instead of treating demand as an autonomous time series, they show how observed operational levers can improve forecast accuracy and managerial understanding. However, this gain in explanatory richness cre-

ates a practical tradeoff. Causal and multivariate models require well-measured covariates, stable relationships between covariates and demand, and enough observations to estimate the effects reliably. When many promotional, price, calendar, and competitive variables are included, the model can face sparse event coverage and a heavy variable-selection burden (Trapero et al., 2015). These difficulties are particularly acute for new products, short sales windows, and low-volume SKUs, where the data needed to estimate a rich explanatory model may not yet exist.

Large-scale empirical evidence also warns that model complexity does not automatically translate into better forecasts. The M-competitions showed that statistically sophisticated methods may fail to outperform simpler alternatives on broad forecasting benchmarks (Makridakis et al., 1982, 1993; Makridakis and Hibon, 2000), and related evidence argues that excessive model complexity can reduce accuracy, increase error opportunities, and weaken managerial usability (Green and Armstrong, 2015). This finding is important for retail supply chains because forecasting is not an isolated statistical exercise. Forecasts are inputs to replenishment, allocation, pricing, and inventory-control decisions, so a method that is marginally more accurate but difficult to maintain may still be unattractive in practice. Studies in fast-fashion inventory and online-retail analytics illustrate that demand forecasts create value only when they are timely and operationally usable (Caro and Gallien, 2010; Ferreira et al., 2016). A forecast produced after a long retraining cycle may be too late for ordering decisions; a model that requires extensive manual feature redesign may be too costly for thousands of SKUs; and a method that performs well on average may still fail when early-cycle observations are sparse.

2.2 Machine learning and deep learning models

ML and DL methods have expanded the forecasting toolkit by allowing nonlinear mappings, high-dimensional covariates, and large-scale pattern learning. Review studies report a broad adoption of ML tools in supply-chain demand forecasting, supplier selection, order allocation, and related operational prediction tasks (Aamer et al., 2021; Ingle et al., 2021; Islam et al., 2021; Feizabadi, 2022; Malviya and Bhandari, 2024). In retail demand prediction, richer information sources and supervised-learning methods have enabled models to use weather, promotion, search, social-media, product-attributes and price information on a scale that is difficult for traditional statistical models (Steinker et al., 2017; Boone et al., 2018; Cui et al., 2018; Lei et al., 2023; Ferreira et al., 2016; Alley et al., 2023). The 2020 M5 Accuracy competition further illustrates this shift: leading methods were ML-based and achieved strong accuracy improvements over traditional statistical benchmarks

on large-scale hierarchical Walmart sales data (Makridakis et al., 2022). This line of work shows that data-intensive models can represent complex retail patterns, pool signals across products, and exploit high-dimensional covariates. It also changes the role of forecasting in supply chains. Forecasting is no longer only an extrapolation task based on a single SKU history; it becomes a pattern-learning problem that uses cross-product, temporal, promotional, and contextual signals to infer future demand.

Within this broad family, different model classes address different sources of complexity. Tree-based ensemble models, such as gradient boosting frameworks, are effective when forecasting is based on tabular covariates, nonlinear interactions, and heterogeneous feature effects. Neural networks and deep sequence models capture nonlinear relationships and temporal dependencies. Early evidence of ANN, CNN-based time-series models, LSTM architectures, attention mechanisms, and Transformer variants have been used to improve retail or time-series forecasting (Adya and Collopy, 1998; Alon et al., 2001; Aburto and Weber, 2007; Zhang et al., 2019; Graves and Schmidhuber, 2005; Mamdough et al., 2024; Gao et al., 2023; Zhang et al., 2024; Li et al., 2024). These models are attractive because they can learn representations rather than requiring researchers to specify all interactions manually. In the retail setting, this is especially useful when demand is jointly affected by price, promotion, calendar, assortment, and historical sales signals. At the same time, deep models can be data-hungry, sensitive to distribution shifts, and difficult to interpret. Their empirical success often depends on the availability of large training histories and on whether future conditions resemble the historical regimes used for training. When demand changes after deployment, simply relying on the offline model may be insufficient, even if the model was strong on the original test set.

Other ML streams complement deep sequence forecasting by addressing uncertainty, dependency, and robustness. Gaussian process and neural network applications in agricultural, commodity, real-estate, and financial price series demonstrate the value of flexible nonlinear learning and uncertainty-aware prediction in related time-series environments (Xu and Zhang, 2021, 2023; Jin and Xu, 2025a). Graph and causal structure methods address interconnected markets and contemporaneous dependencies, which are relevant to retail systems where products may substitute, complement, or share demand drivers (Jin and Xu, 2025b). Ensemble and composite forecasts improve robustness by combining multiple base learners and reducing dependence on a single model specification (Guo et al., 2025). These approaches broaden the modeling landscape, but they also highlight a recurring limitation. However, the literature has developed many powerful first-stage

predictors, but the second-stage question of real-time forecast correction remains less developed.

The need for adaptation has motivated online learning, reinforcement learning, and continual-learning approaches in operations and forecasting. Online-learning studies emphasize updating models as new data arrive, while reinforcement-learning studies show how operational policies can learn from repeated interaction with uncertain environments (Petropoulos et al., 2022; Wang et al., 2024; Boute et al., 2022; Gijbrecchts et al., 2022). These studies are important because they move beyond purely static prediction and recognize that decisions and observations unfold sequentially. However, a direct reinforcement-learning formulation may be excessive for demand correction when the immediate goal is not to learn an entire inventory policy but to adjust a baseline forecast before downstream decisions are made. Similarly, online retraining can be costly and unstable when only a few new observations are available. Continual-learning and few-shot studies further show that unrestricted parameter updates can cause overfitting or catastrophic forgetting under limited new data (Kirkpatrick et al., 2017; Tian et al., 2024; Bethune et al., 2025; Mazumder et al., 2021). This creates a stability-plasticity dilemma: a model must adapt enough to correct new bias, but not so much that it destroys the useful structure learned from previous demand cycles.

The closest unresolved issue is therefore not whether advanced forecasting models can be accurate offline, but how their forecasts should be adapted online after new demand feedback is observed. Most demand forecasting pipelines still generate static predictions, select among static models, or require costly retraining when demand changes. They rarely treat each forecast as a decision that can be immediately corrected using contextual feedback. This distinction is central to our study. The PtC framework does not attempt to replace statistical, machine-learning, or deep-learning forecasters. Instead, it uses them as the first stage and adds a CB-based correction policy as the second stage. The CB formulation is appropriate because each correction action receives immediate feedback through the change in forecast error, which makes the learning signal local, interpretable, and sample efficient. The few-shot continuous updating design further addresses the limited-data setting by updating only selected parameters through the Top- p mechanism, thus preserving prior correction knowledge while allowing local adaptation.

3 Solution methods

3.1 AI/ML prediction and data augmentation

Figure 1 illustrates the overall framework of the proposed PtC method. In the first stage, using average demand, coefficient of variation (CoV), and intermittency as clustering features, SKUs will be divided into four categories: stable & high volume, stable & low volume, erratic & intermittent, and lumpy (Ma et al., 2025). Each category is then associated with an ML algorithm for prediction. For the target SKU, the first step is to identify its category and then generate a baseline forecast using the corresponding ML model. The resulting baseline forecast is used as input to the second stage. Augmentation data selection within each cluster is provided in Appendix C.

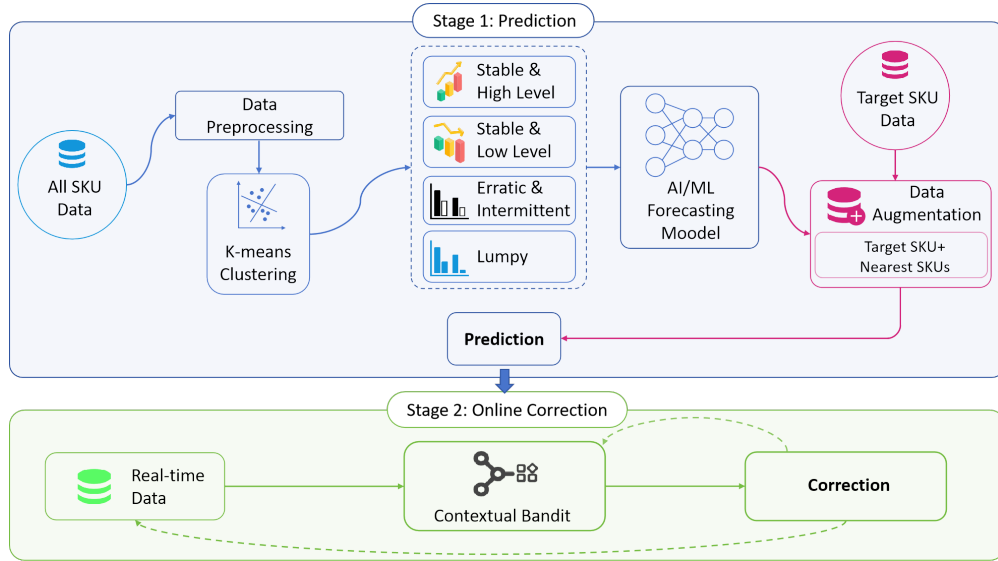


Figure 1: Total framework of the proposed method.

In the second stage, we adopt the CB as the correction model to adjust the ML-predicted demand by real-time observations within a sliding window. To support timely prediction and avoid delaying operational decisions, only five or seven days of demand information can be collected; otherwise, the correction loses its practical relevance. In this setting, real-time demand information from a single SKU is insufficient to train an effective correction policy and may cause severe overfitting or catastrophic forgetting. Therefore, it is necessary to design a data augmentation strategy to expand the training set and alleviate these risks.

For the target SKU, there are often numerous SKUs within the same cluster that share similar characteristics. Thus, we first rank candidate SKUs by their Euclidean distance to the target

SKU and then select the k nearest SKUs to augment the training set. Meanwhile, an upper distance threshold d_{is} is set to avoid selecting SKUs that are too dissimilar from the target SKU. If fewer than k SKUs satisfy this threshold, we apply same-trend perturbations to both historical demand and ML-predicted demand until a pre-specified minimum training-set size is reached. Clustering, scaling, neighbor selection, and augmentation parameters are estimated using training-period information only to prevent test-period leakage.

3.2 FSCCB

3.2.1 State, action, and reward design

With full observation enabled, the observed state, or context, of the PtC system at time step t is denoted as:

$$s_t = \left\{ g_r^d, g_x^d, g_e^d, g_{hr}^d, g_{hx}^d, g_s^d, g_d^t, g_r^t, g_w^t, g_h^t, g_s^t \right\}. \quad (1)$$

The description of each component is shown in Table 1. Specifically, g_r^d represents the historical real demand, g_x^d represents the historical ML-predicted demand, and g_e^d represents the historical prediction error. In addition, g_{hr}^d and g_{hx}^d denote the mean historical real demand and the mean historical ML-predicted demand, respectively, while g_s^d measures the historical slope of the real demand. Time-related features include the current time step, the remaining time before the end of the cycle, and binary indicators for weekend, holiday and shopping festival effects.

In the CB framework, s_t is used as the contextual information available before making the correction decision. The policy network uses this context to determine how the original ML prediction should be adjusted.

In this study, the action is defined as the continuous adjustment factor applied to the ML-predicted demand. Given the observed context s_t , the policy network outputs the mean and standard deviation of a Gaussian correction distribution, denoted by $\mu_\theta(s_t)$ and $\sigma_\theta(s_t)$, respectively. During training and online adaptation, the correction action is sampled as

$$\tilde{a}_t \sim \mathcal{N}(\mu_\theta(s_t), \sigma_\theta^2(s_t)), \quad a_t = \text{clip}(\tilde{a}_t, -1, 2). \quad (2)$$

Here, a_t denotes the sampled relative correction ratio. A negative value indicates that the original ML prediction should be reduced, while a positive value indicates that the original ML prediction should be increased. The action range is limited to $[-1, 2]$ to prevent the correction model

Table 1: State representation components.

Type	Description	Notation
Historical data	Historical real demand	g_r^d
	Historical ML-predicted demand	g_x^d
	Historical ML error	g_e^d
	Mean historical real demand	g_{hr}^d
	Mean historical ML-predicted demand	g_{hx}^d
	Historical slope of real demand	g_s^d
Time characteristics	Current time step	g_d^t
	Remaining time before the end of the cycle	g_r^t
	Is weekend or not	g_w^t
	Is holiday or not	g_h^t
	Is shopping festival or not	g_s^t

from generating unreasonable adjustment values. The stochastic sampling in Eq. (2) provides local exploration over possible upward and downward forecast adjustments.

Given the original ML-predicted demand $\hat{y}_t^{ML}$, the corrected demand is calculated as:

$$\hat{y}_t^{CB} = \hat{y}_t^{ML}(1 + a_t^*), \quad (3)$$

where $\hat{y}_t^{CB}$ denotes the demand value corrected by the CB model.

Therefore, the CB policy does not directly predict the final demand. Instead, it learns how to adjust the baseline ML prediction according to the observed context.

The reward is defined as the relative degree of improvement after correction. At each time step t , the prediction error of the original ML model is calculated as:

$$e_{ML,t} = |\hat{y}_t^{ML} - y_t|, \quad (4)$$

where y_t denotes the actual demand.

Similarly, the prediction error after the CB correction is calculated as:

$$e_{CB,t} = |\hat{y}_t^{CB} - y_t|. \quad (5)$$

To evaluate whether the correction action improves the original ML prediction, the reward function is designed as:

$$r_t = \tau(e_{ML,t} - e_{CB,t}), \quad (6)$$

where τ is a scaling factor used to normalize the reward into a suitable numerical range.

According to Eq. (6), a positive reward indicates that the CB correction reduces the prediction error compared with the original ML prediction. A negative reward means that the correction worsens the prediction result. Therefore, maximizing the reward is equivalent to learning a correction policy that improves the forecasting accuracy of the baseline ML model.

3.2.2 CB for forecast correction

In this study, the forecast correction problem is formulated as a continuous contextual bandit problem. The proposed PtC framework first uses the baseline ML model to generate an initial demand prediction. Then, the CB model samples a continuous correction action according to the stochastic policy $\pi_\theta(\cdot | s_t)$, the main steps are provided in Figure 2.

The correction process can be represented as: $s_t \rightarrow a_t^* \rightarrow r_t$, where s_t denotes the observed context, a_t^* denotes the correction action, and r_t denotes the reward measuring the improvement after correction. At each time step t , the correction model first observes the context s_t , which is constructed from recent demand, baseline ML forecasts, historical forecast errors, and time-related features. The policy then samples a continuous correction action a_t from $\pi_\theta(\cdot | s_t)$ and applies it to the baseline forecast. This stochastic action-selection mechanism enables exploration around the current policy mean and allows the model to test whether increasing, decreasing, or retaining the baseline forecast is more beneficial under the current context.

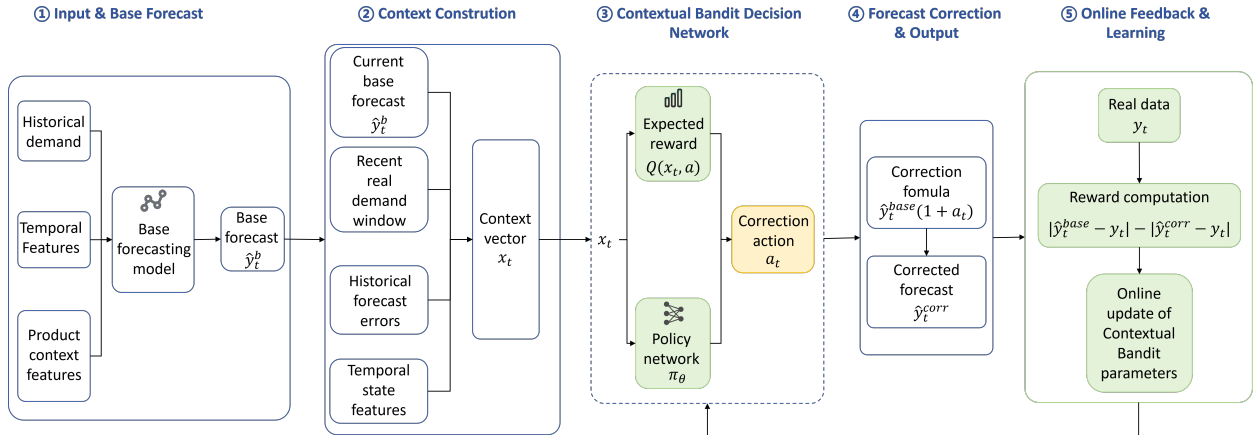


Figure 2: Main steps of the second stage in PtC.

The bandit feedback structure is essential in this formulation. Before the actual demand y_t is observed, the model only has access to s_t and must commit to one correction action a_t . After y_t

is realized, the reward r_t is computed by comparing the baseline forecast error and the corrected forecast error. The model uses the reward of the selected action to update the policy, while no supervised target action is imposed. The newly observed tuple: $(s_t, a_t, r_t, \log \pi_\theta(a_t | s_t))$ is then added to the sliding-window feedback set and used for future policy updates. Therefore, PtC learns from bandit feedback by increasing the probability of correction actions that receive positive rewards and decreasing the probability of actions that worsen the baseline forecast.

3.2.3 Learning Objective of the CB Correction Model

Let the chronological feedback set be:

$$\mathcal{D} = \{(s_t, a_t^*, r_t)\}_{t=1}^T, \quad (7)$$

where each tuple contains the context, the correction action, and the realized reward. The goal is to learn a policy π_θ that maps the observed context to a continuous correction ratio:

$$a_t^* = \pi_\theta(s_t). \quad (8)$$

Given the first-stage forecast $\hat{y}_t^{ML}$, the corrected forecast induced by this policy is:

$$\hat{y}_t^{CB} = \hat{y}_t^{ML} (1 + \pi_\theta(s_t)). \quad (9)$$

Thus, the policy network does not replace the first-stage forecasting model. It only learns a context-dependent adjustment factor for the existing forecast.

At the population level, the CB correction policy is trained to maximize the expected reward obtained after correction:

$$\pi_\theta^* = \arg \max_{\pi_\theta} \mathbb{E}_{s_t \sim \mathcal{D}} [r_t(s_t, \pi_\theta(s_t))]. \quad (10)$$

This objective is appropriate for forecast correction because each action is evaluated by its immediate effect on the current prediction error. Unlike a full reinforcement-learning formulation, no long-horizon value function is required; the correction model only needs to learn whether the current context calls for increasing, decreasing or retaining the baseline forecast.

In empirical training, the expected objective is approximated by the average reward over the feedback set:

$$\theta^* = \arg \max_{\theta} \frac{1}{T} \sum_{t=1}^T r_t. \quad (11)$$

Substituting the reward definition from Eq. (6) into Eq. (11) gives:

$$\theta^* = \arg \max_{\theta} \frac{1}{T} \sum_{t=1}^T \tau (|\hat{y}_t^{ML} - y_t| - |\hat{y}_t^{ML} (1 + \pi_{\theta}(s_t)) - y_t|). \quad (12)$$

The first error term in Eq. (12) is determined by the baseline ML forecast and does not depend on θ . Therefore, maximizing the reward is equivalent to minimizing the corrected prediction error:

$$\mathcal{L}(\theta) = \frac{1}{T} \sum_{t=1}^T |\hat{y}_t^{ML} (1 + \pi_{\theta}(s_t)) - y_t|. \quad (13)$$

This equivalence connects the reward-based contextual bandit formulation with the standard forecasting objective: a high reward means that the correction policy reduces the absolute error of the first-stage forecast.

Proposition 1. *Let L_{PtC}^* and L_{ML} denote the optimal corrected and ML-only expected losses induced by any nonnegative forecasting loss ℓ . If $0 \in \mathcal{A}$ and $\pi_0(s) = 0 \in \Pi$, then $L_{PtC}^* \leq L_{ML}$.*

Proposition 1 establishes a hypothesis-class containment result at the population optimum. Since the zero-correction policy $\pi_0(s) = 0$ is included in the PtC policy class, PtC contains the ML-only forecast as a special case, and its optimal population loss is no larger than that of the ML-only forecast. When the correction action is zero for every context, the corrected forecast exactly reduces to the original ML forecast. This result should be interpreted as an expressiveness and containment property of the PtC hypothesis class, rather than as a finite-sample performance guarantee: it does not imply that the estimated PtC model will necessarily outperform the ML-only baseline with limited samples or at every online decision step. A proof of Proposition 1 is provided in Appendix D.

In implementation, historical demand cycles are replayed in chronological order to initialize the policy, and the same feedback structure is used during online deployment. This design preserves the decision timing of the CB formulation: the policy observes s_t , selects a_t^* , receives r_t after demand is realized and then updates itself for future correction decisions.

3.2.4 Top- p update

In a new demand cycle, customer demand is gradually observed over time. The CB correction model must update its parameters to incorporate new information from the demand environment. However, in such a data-scarce scenario, frequent parameter updates may lead to overfitting or catastrophic forgetting of previously learned knowledge (Kirkpatrick et al., 2017; Tian et al., 2024;

Bethune et al., 2025). To mitigate this risk, we adopt a parameter update rule based on the proportion of parameters.

We adopt a top- p masked update rule to control the degree of online adaptation. Specifically, we select the top- p proportion of parameters with the smallest absolute values and update only these parameters, while the remaining parameters are frozen. Small-magnitude parameter selection is a heuristic intended to preserve highly expressed pretrained parameters; its effectiveness must therefore be established empirically. Intuitively, parameters with larger absolute values are treated as more strongly expressed components of the pretrained correction policy, whereas smaller-magnitude parameters provide a limited adaptation subspace for learning new-cycle feedback.

To implement this rule, we first sort the parameters in each layer by their absolute values. The threshold δ^k is then set to the largest magnitude within the smallest $p\%$ parameters of layer k . Parameters whose absolute values are no larger than δ^k are selected for online updating, while the remaining parameters are kept fixed.

After the threshold is determined, we adopt a binary mask matrix in equation (14) to record which parameters need to be updated as follows:

$$M(\theta_j^k) = \begin{cases} 1, & |\theta_j^k| \leq \delta^k \\ 0, & \text{otherwise} \end{cases} \quad (14)$$

where θ_j^k is the j -th parameter in layer k . If $M(\theta_j^k) = 1$, the parameter θ_j^k is selected to be updated. The shift of the update rule can be found in equation (15).

$$\theta_j^k \leftarrow \theta_j^k - \eta g_j^k \quad \rightarrow \quad \theta_j^k \leftarrow \theta_j^k - \eta M(\theta_j^k) g_j^k \quad (15)$$

where η represents the learning rate. In our research, we set the learning rate to a very small value to enable fine-tuning updates, thereby avoiding disruption of the knowledge learned before (Zhang et al., 2023; Tao et al., 2020).

Proposition 2. *Let $\Theta_p = \{\theta^0 + \Delta : \text{supp}(\Delta) \subseteq S_p\}$ and $S_{p_1} \subseteq S_{p_2}$ for $p_1 \leq p_2$. Then $G_{ad}(p) = \inf_{\theta \in \Theta_p} \mathcal{L}_{new}(\theta) - \inf_{\theta \in \Theta_1} \mathcal{L}_{new}(\theta)$ is non-increasing in p . If $|g_j^u| \leq G$, then $\|\theta_p^U - \theta^0\|_2 \leq \eta U G \sqrt{\lceil pd \rceil}$, so the forgetting-risk bound is non-decreasing in p .*

Proposition 2 characterizes the adaptation-stability tradeoff induced by the top- p update rule. The first part states that increasing p enlarges the set of trainable parameters. Since $\Theta_{p_1} \subseteq \Theta_{p_2}$ when $p_1 \leq p_2$, a larger update set cannot increase the best achievable loss on the new demand

cycle. Therefore, the adaptation gap $G_{ad}(p)$ is non-increasing in p . This explains why updating too few parameters may limit the ability of the correction policy to adapt to a new demand pattern. proof of Proposition 2 is provided in Appendix E.

Proposition 3. *Let $\mathcal{J}_\epsilon(\theta) = \mathbb{E}_{\mathcal{D}, a_t \sim \pi_\theta} [\rho_\epsilon(\hat{y}_t^{ML}(1+a_t) - y_t)]$ be lower bounded and L -smooth. Under $\theta^{u+1} = \theta^u - \eta m_u \odot \nabla \mathcal{J}_\epsilon(\theta^u)$ with $0 < \eta \leq 1/L$, the PtC correction policy satisfies $\mathcal{J}_\epsilon(\theta^{u+1}) \leq \mathcal{J}_\epsilon(\theta^u)$ for $u = 0, 1, \dots$*

Proposition 3 provides a stability guarantee for the masked update when the correction objective is represented by a smooth surrogate loss. The absolute forecasting error is not differentiable at zero, so $\rho_\epsilon(\cdot)$ is introduced as a smooth approximation of the correction error. If this surrogate objective is lower bounded and L -smooth, then the standard descent lemma implies that a masked gradient step with step size $0 < \eta \leq 1/L$ does not increase the objective value. A proof of Proposition 3 is provided in Appendix F.

Pseudo-code of top- p masked update is described in Appendix G.

4 Experiment

4.1 Experimental design

4.1.1 Datasets and implementation

All algorithms were implemented in Python 3.11 (64-bit). The experiments were conducted on a workstation equipped with an Intel Core i7-14700KF CPU @ 5.6 GHz, 32 GB DDR5 RAM, and an NVIDIA RTX 4070 Ti GPU.

We evaluate the proposed PtC framework using the Walmart sales data released through the M5 Accuracy competition in 2020 (Makridakis et al., 2022). Specifically, we use 6,875 SKUs from Walmart stores “CA_1” to “CA_3”, covering more than 1,500 daily observations.

The beverage dataset is an external industrial benchmark collected from a major North American beverage manufacturer. It contains 156 weeks of weekly shipment records 49 bottling plants and over 12,000 distribution nodes.

The lumpy category is defined in Section 3 for completeness of SKU segmentation, but it is omitted from the later experiments because lumpy demand has long zero-demand periods and irregular spikes, making statistical and AI/ML models largely ineffective; therefore, we recommend a qualitative CPFR-based approach instead of reporting model-comparison results.

Details of dataset splitting can be found in Appendix A, and detailed descriptions of the parameter values of the set are provided in Appendix B.

4.1.2 Evaluation metrics

We evaluate the precision of the forecast by mean absolute percentage error (MAPE), mean absolute error (MAE), and root mean squared error (RMSE). For a test sequence with actual demand y_t and forecast $\hat{y}_t$, the metrics are defined as:

$$MAPE = \frac{100}{T} \sum_{t=1}^T \left| \frac{y_t - \hat{y}_t}{\max(|y_t|, \epsilon)} \right|, \quad (16)$$

$$MAE = \frac{1}{T} \sum_{t=1}^T |y_t - \hat{y}_t|, \quad RMSE = \sqrt{\frac{1}{T} \sum_{t=1}^T (y_t - \hat{y}_t)^2}. \quad (17)$$

where ϵ is a small positive constant used to avoid division by zero.

MAPE provides a scale-free percentage error, MAE reflects the average absolute deviation, and RMSE emphasizes large forecast errors.

For the demand-category comparison, we report the relative error reduction:

$$Reduction(M, q) = \frac{E_q^{ref} - E_q^M}{E_q^{ref}} \times 100\%, \quad (18)$$

where E_q^M denotes the error of method M under metric q , and E_q^{ref} denotes the corresponding error of the ML-only reference forecast. A positive value indicates an improvement over the reference forecast, whereas a negative value indicates a deterioration in forecasting performance.

4.1.3 Benchmark methods

ETS (Gardner Jr, 1985) is an exponential smoothing method that recursively updates level, trend, and seasonal components. A representative additive ETS forecast is given by

$$\hat{y} * t + h^{ETS} = \ell_t + hb_t + s * t + h - m, \quad (19)$$

where ℓ_t , b_t , and s_t denote the level, and seasonal states, respectively, and m is the seasonal period.

ARIMA is a classical linear time-series model that captures autocorrelation through autoregressive and moving-average terms after differencing. The model can be written as

$$\phi(B)(1 - B)^d y_t = c + \theta(B)\varepsilon_t, \quad (20)$$

where B is the backshift operator, d is the differencing order, and ε_t is the innovation term.

LightGBM (Ke et al., 2017) is a gradient boosting decision tree model that builds an additive ensemble of regression trees. Its prediction can be expressed as

$$\hat{y} * t = \sum * k = 1^K f_k(x_t), \quad f_k \in \mathcal{F}, \quad (21)$$

where x_t is the input feature vector and $f_k(\cdot)$ denotes the k -th regression tree.

XGBoost (Chen and Guestrin, 2016) is another boosted-tree model that learns additive regression trees by minimizing a regularized objective:

$$\mathcal{L}^{(k)} = \sum_{t=1}^T l\left(y_t, \hat{y}_t^{(k-1)} + f_k(x_t)\right) + \Omega(f_k), \quad (22)$$

where $l(\cdot)$ is the loss function and $\Omega(f_k)$ penalizes tree complexity.

N-BEATS is a deep neural forecasting model composed of stacked fully connected residual blocks. Each block generates a backcast for residual updating and a forecast, while the final prediction is obtained by summing the block-level forecasts.

TiDE is a dense encoder–decoder forecasting model that maps historical demand and covariates into multi-step forecasts.

TFT is an attention-based deep forecasting model that integrates static features, time-varying covariates, gating mechanisms, and temporal attention for multi-horizon forecasting.

4.2 Results

4.2.1 Numerical experiment on the Walmart dataset

Table 2 reports the average relative error reduction on the Walmart retail dataset across three distinct demand categories. The results are calculated based on Eq. (18), where positive values indicate a performance improvement over the LightGBM ML-only reference forecast, and negative values denote an algorithmic decrease.

Across all demand patterns, the proposed PtC framework consistently shows superior adaptability compared to both traditional statistical methods and DL architectures. Notably, PtC is the only method evaluated that achieves positive relative error reduction across all three metrics in every reported setting. Furthermore, all observed improvements are statistically significant at the $p < 0.01$ level.

Table 2: Comparison result of Average relative error reduction on the Walmart dataset.

Method	Stable & Low Volume			Stable & High Volume			Erratic & Intermittent		
	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE
ETS	0.05	4.92	1.83	-9.66	-1.10	-2.08	-14.24	3.82	1.18
N-BEATS	7.31	5.23	1.42	-5.36	5.40	1.30	-26.85	5.19	-1.60
XGBoost	-12.54	1.92	-0.81	-1.84	0.49	0.47	-8.78	-1.24	-13.75
ARIMA	1.03	4.68	1.90	-9.31	-4.17	-9.32	-14.77	3.67	1.09
TiDE	11.57	10.98	1.95	-5.82	5.17	0.86	-21.97	4.38	-1.29
TFT	6.67	10.78	0.86	-10.09	-0.43	-5.38	-25.38	4.81	-1.68
PtC	12.16***	11.28***	6.87***	6.74***	9.95***	7.67***	5.68***	8.94***	3.70***

Note: Positive values indicate improvement relative to the ML-only reference forecast, while negative values indicate performance decrease. * $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$.

The empirical results expose the structural vulnerabilities of static baseline models when subjected to the highly stochastic nature of retail supply chains. Deep forecasting architectures, such as N-BEATS, TiDE, and TFT, exhibit moderate success under “Stable & Low Volume” conditions but degrade substantially when applied to “Erratic & Intermittent” demand. Specifically, N-BEATS shows a -26.85% decrease in MAPE for intermittent items. This underscores a limitation of pure data-driven models: their susceptibility to overfitting the noisy, sparse signals inherent in intermittent demand, which subsequently induces severe predictive bias. Conversely, traditional time-series methods like ETS and ARIMA struggle significantly with “Stable & High Volume” series. This is likely driven by their constrained parametric forms, which fail to capture the complex, non-linear dynamics characterizing high-throughput retail operations.

In contrast, the performance of the PtC architecture achieves average error reductions between 5.68% and 8.94% even in the highly challenging erratic category, validating its underlying mathematical design. Rather than overriding the historical distributions captured by the base ML model, the FSCCB correction layer dynamically fine-tunes the forecast via online reward feedback. This decoupled architecture effectively bridges the methodological gap between offline historical generalization and online correction.

4.2.2 Numerical experiment on the beverage dataset

To better validate the performance of the proposed correction mechanism, we conduct a supplementary numerical experiment on a beverage distribution dataset. Table 3 details the average relative error reduction under the previously established evaluation framework.

Table 3: Comparison result of Average relative error reduction on the beverage dataset.

Method	Stable & Low Volume			Stable & High Volume			Erratic & Intermittent		
	MAPE	MAE	RMSE	MAPE	MAE	RMSE	MAPE	MAE	RMSE
LightGBM	6.18	5.05	6.12	–	–	–	3.91	-1.01	2.93
ETS	–	–	–	7.82	2.89	-3.67	–	–	–
N-BEATS	5.44	6.25	5.45	10.88	-5.57	-11.48	-10.39	4.01	2.20
XGBoost	-5.94	4.71	-5.59	-6.24	-7.62	-8.46	0.79	0.64	2.79
ARIMA	6.80	-5.01	5.37	7.85	-18.07	-21.99	6.27	4.99	1.72
TiDE	6.91	5.43	5.58	11.19	-3.46	-6.14	1.24	-12.92	-0.04
TFT	5.49	6.76	-6.00	8.19	5.88	2.73	-10.33	7.06	2.27
PtC	13.64***	7.85***	7.09***	14.64***	14.85***	13.40***	8.92***	12.37***	4.48***

Note: Positive values indicate improvement relative to the ML-only reference forecast, while negative values indicate a performance decrease. * $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$.

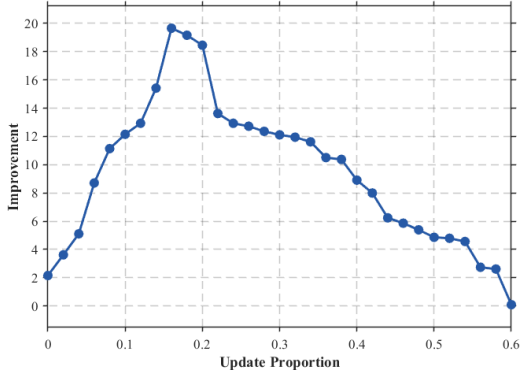
In this data environment, traditional methods like ARIMA and tree-based models like XGBoost exhibit severe performance decrease in the “Stable & High Volume” category. Furthermore, DL models such as N-BEATS, TFT, yield highly inconsistent results across different metrics within identical demand categories. Such metric instability indicates that these base models are highly sensitive to shifting data distributions and lack the continuous learning capability required to mitigate real-time prediction errors under structural volatility.

Conversely, PtC maintains substantial, balanced, and statistically significant ($p < 0.01$) improvements among all conditions. The error reductions peak in the “Stable & High Volume” category, achieving 14.64%, 14.85%, and 13.40% error reduction for MAPE, MAE and RMSE, respectively. By retaining the base ML model’s structural understanding of baseline demand while deploying an online CB to correct for micro-level volatility, PtC shifts the forecasting paradigm from a static, open-loop prediction task to an adaptive, closed-loop correction system.

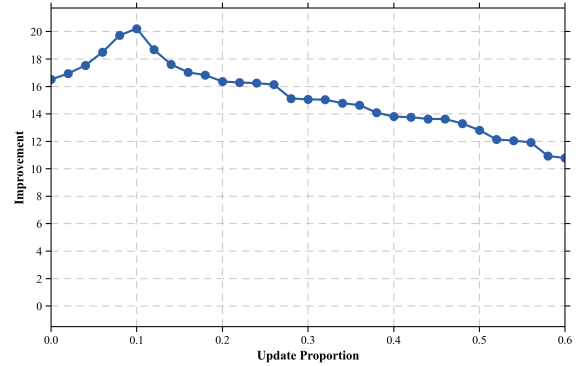
4.3 Sensitivity analysis

We further analyze the sensitivity of the top- p parameter update proportion and the depth of the correction network.

Figure 3 compares the sensitivity of the update proportion p on the North American beverage company dataset and the Walmart retail dataset. The two curves show that updating too few parameters limits the performance of the correction model, while updating too many parameters increases the risk of overfitting or forgetting previously learned demand patterns. On the beverage-company dataset, the improvement rises rapidly from a small update proportion and reaches its peak $p = 0.16$, after which performance declines steadily. On the Walmart dataset, the improvement is already significant under small update proportions and reaches its highest value $p = 0.10$; further increasing p leads to a gradual decrease in improvement. This cross-dataset evidence suggests that FSCCB benefits from selective adaptation rather than full-network updating.



(a) North American beverage company dataset



(b) Walmart retail dataset

Figure 3: Parameter sensitivity analysis of the update proportion p across two datasets.

Figure 4 reports the sensitivity to network depth. A shallow correction network has limited representation capacity and may fail to capture nonlinear relationships between context features and correction actions. However, an overly deep correction network increases the number of trainable parameters and becomes less stable under few-shot updates. The observed pattern supports the use of a moderate-depth policy network for the CB correction model.

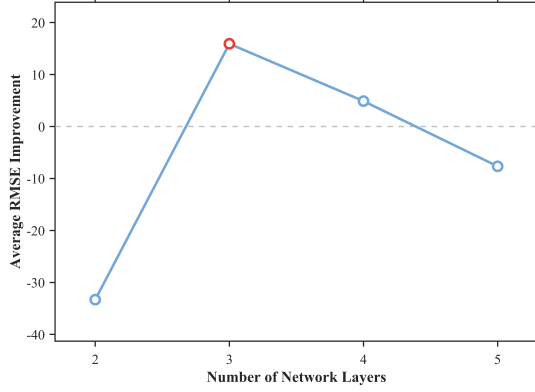


Figure 4: Network depth sensitivity analysis.

4.4 Ablation study

To isolate the contribution of the CB correction layer and the few-shot selective update rule, we conduct an ablation study using N-BEATS as the first-stage ML forecast. We compare three variants as follows:

- (1) ML-only: Forecasting relies on the N-BEATS prediction without second-stage correction;
- (2) ML+CB: A CB correction model is applied, but all correction-network parameters are updated during adaptation;
- (3) ML+FSCCB: FSCCB is applied, where only the selected top- p parameters are updated during adaptation.

The reported Gap is the relative RMSE reduction compared with the ML-only baseline. A positive Gap indicates that the correction module improves the N-BEATS forecast.

Table 4 reports the average ablation results of the proposed few-shot continuous CB method. Compared with the original N-BEATS model, the standard N-BEATS-CB only achieves a marginal RMSE reduction of 0.34%. Moreover, although its average RMSE is slightly lower than that of N-BEATS, its average MAE becomes higher. This indicates that simply introducing a CB-based correction network does not provide a sufficiently stable improvement under limited observations from a new demand cycle.

By contrast, PtC achieves a much larger average improvement. It reduces the average RMSE from 11.4086 to 10.3372 and the average MAE from 8.7993 to 7.8193, corresponding to an average RMSE Gap of 9.52%. This result shows that the proposed FSCCB method can more effectively adapt the forecasting model to a new demand cycle.

Table 4: Average prediction performance comparison of N-BEATS, N-BEATS-CB, and PtC.

N-BEATS		N-BEATS-CB			PtC		
RMSE	MAE	RMSE	MAE	Gap	RMSE	MAE	Gap
11.4086	8.7993	11.3903	8.8363	0.34%	10.3372	7.8193	9.52%

The improvement mainly comes from the few-shot selective update mechanism. Instead of updating all parameters of the correction network, FSCCB updates only a small subset of low-magnitude parameters while freezing the remaining parameters. This strategy helps preserve the general correction knowledge learned from historical demand cycles and prevents the model from overfitting short-term noise in the new cycle. Therefore, the ablation results confirm that FSCCB provides a more stable and effective correction mechanism than the standard CB model, making the proposed PtC framework better suited for demand forecasting under data scarcity.

5 PtC in Inventory Management System

5.1 Model Formulation in Inventory Management System

The forecasting experiments show that PtC improves prediction accuracy, but the operational value of a forecast depends on how it affects downstream decisions. We therefore embed the corrected forecasts into a periodic-review inventory management system. At the beginning of period t , the retailer observes the inventory position IP_t , outstanding replenishment orders, and the current demand context. A replenishment policy then chooses an order quantity $q_t \geq 0$. Orders arrive after a deterministic lead time L , and demand y_t is realized at the end of the period.

Let I_t denote the net inventory after demand is realized, where positive values represent inventory and negative values represent unmet demand. The inventory transition is written as:

$$I_{t+1} = I_t + q_{t-L} - y_t, \quad (23)$$

where q_{t-L} is the order placed L periods earlier. The single-period inventory cost is:

$$C_t = h[I_{t+1}]^+ + b[-I_{t+1}]^+, \quad (24)$$

where h is the unit holding-cost coefficient, b is the unit shortage-cost coefficient, and $[x]^+ =$

$\max\{x, 0\}$. The objective is to minimize the average cost over the evaluation horizon:

$$\bar{C} = \frac{1}{T} \sum_{t=1}^T C_t. \quad (25)$$

PtC enters this system through the demand signal used for replenishment. The baseline ML forecast $\hat{y}_t^{ML}$ is first corrected by the learned contextual-bandit action a_t , producing:

$$\hat{y}_t^{PtC} = \hat{y}_t^{ML}(1 + a_t). \quad (26)$$

The corrected forecast is then used to estimate lead-time demand. For a lead time L , the PtC-based replenishment target can be expressed as:

$$\hat{D}_{t,L}^{PtC} = \sum_{j=0}^{L-1} \hat{y}_{t+j}^{PtC}, \quad (27)$$

and the corresponding order quantity is:

$$q_t^{PtC} = \max\{0, \hat{D}_{t,L}^{PtC} - IP_t\}. \quad (28)$$

This formulation keeps the inventory decision simple while allowing the order quantity to respond to online forecast corrections.

5.2 Baseline Policies and Experiment Settings

We compare PtC with three inventory-control baselines: base-stock (BS), proximal policy optimization (PPO), and soft actor-critic (SAC). BS is a classical order-up-to policy that replenishes inventory toward a target level based on the estimated lead-time demand. It provides a transparent operational benchmark for forecast-driven inventory control.

PPO and SAC represent deep reinforcement learning policies for inventory management, a setting in which the policy learns order decisions from repeated interaction with the inventory simulator (Boute et al., 2022; Gijbrecchts et al., 2022). PPO is an on-policy actor-critic algorithm that updates the policy through a clipped surrogate objective, which improves training stability by limiting overly large policy updates. SAC is an off-policy actor-critic algorithm that maximizes both expected return and policy entropy, encouraging exploration in continuous-control problems (Haarnoja et al., 2018). In this experiment, both RL baselines observe the same inventory state information and are trained to minimize the cumulative inventory cost.

All policies are evaluated under the same demand sequences, cost parameters, and lead-time settings. We vary the lead time from $L = 2$ to $L = 4$ to test whether the policies remain robust

when the replenishment decision must anticipate demand further into the future. Inventory cost is reported in millions, and lower values indicate better downstream operational performance.

5.3 Numerical Results

Figure 5 reports the inventory-cost comparison across the tested lead-time settings. PtC achieves the lowest inventory cost across the reported settings, indicating that the forecast improvements observed in the earlier experiments translate into downstream cost reductions. The result is important because a lower forecast error does not automatically guarantee a lower inventory cost; the correction must also improve the timing and order quantities of replenishment under the uncertainty of lead time.

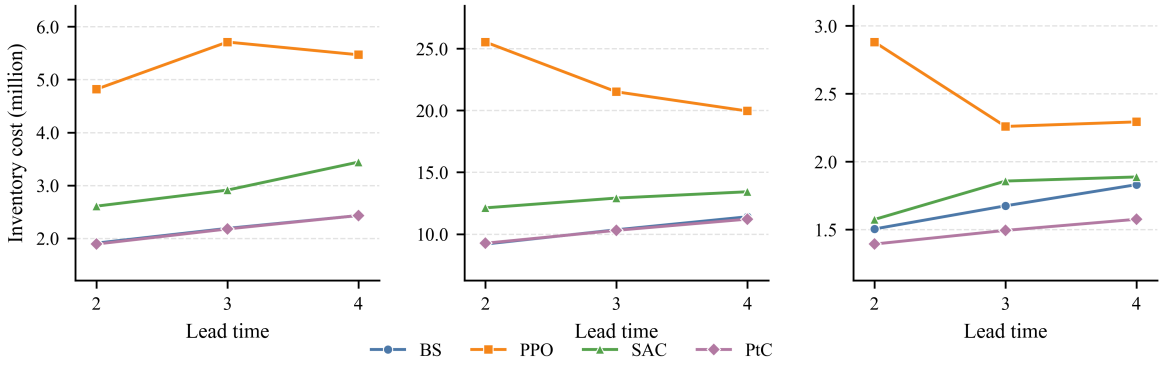


Figure 5: Inventory-cost comparison of BS, PPO, SAC and PtC under different lead times.

The comparison also shows different failure modes across baselines. PPO yields the highest inventory costs in the reported settings, suggesting that the learned policy is less stable for this demand-correction task. SAC performs better than PPO but remains above PtC, indicating that a generic continuous-control policy may still struggle when the main uncertainty comes from demand forecasting errors. BS is competitive in some settings because it directly links replenishment to estimated lead-time demand, but it does not adapt the forecast itself when real-time demand feedback reveals bias. PtC combines these two advantages: it preserves a simple forecast-driven replenishment structure and improves the input forecast through online correction.

As the lead time increases, inventory decisions become more sensitive to forecast bias because orders must cover a longer demand interval. PtC maintains the lowest cost under the tested lead times, suggesting that correcting the forecast before the replenishment decision can reduce both excess inventory and shortage risk. These results provide downstream evidence for the practical

value of the PtC loop in inventory management systems.

6 Conclusion

This paper studies adaptive demand forecasting in retail supply chains, where replenishment, pricing, purchasing, and inventory decisions require timely forecasts but demand patterns can shift before static forecasting models are fully retrained. To address this problem, we propose a PtC framework that preserves the base forecast generated by an offline machine-learning model and then applies a few-shot CB correction policy to adjust the forecasts as new feedback arrives. The main finding is that forecast correction can be treated as a sequential learning problem: instead of replacing the original forecaster, a lightweight correction layer can improve predictive reliability and downstream decision quality while retaining the structure learned from historical demand.

This study contributes to the forecasting and retail operations literature by filling a gap between static forecasting and full model retraining. Existing forecasting models often produce predictions as fixed outputs, while many operational decision models assume that demand information is already available in a usable form. PtC connects these two views by modeling the post-forecast stage as an adaptive correction process, thereby making demand forecasting part of a continuous learning loop. The similar-SKU augmentation and top- p masked update further clarify how a correction policy can balance responsiveness to new observations with the preservation of previously learned demand regularities. In practice, this design can be embedded into existing forecast pipelines with limited disruption, helping retailers update forecasts for early-cycle, sparse, low-volume, or intermittent items without rebuilding the entire forecasting system.

The strength of this study lies in its modular design and its direct connection between forecast accuracy and operational usefulness. Because PtC is placed after the base forecaster, it can work as an adaptive layer rather than as a replacement for existing forecasting models, which makes the framework easier to interpret and deploy. At the same time, the current study has several limitations. The correction policy is built at the SKU level and does not yet explicitly model substitution, complementarity, shelf-space competition, or capacity coupling across products. The downstream decision setting is also simplified, and the robustness of the correction range and top- p update needs further examination under more diverse retail conditions.

Future research can extend this work in several directions. First, multi-product correction policies can be developed by incorporating graph, hierarchical, or attention-based structures to capture

dependencies across SKUs. Second, forecast correction can be integrated more closely with replenishment, pricing, allocation, and purchasing decisions so that prediction and decision optimization are learned in a more unified loop. Third, future studies should examine online deployment issues such as delayed feedback, changing assortment structures, promotion shocks, stochastic lead times, and service-level constraints. These extensions would move PtC from adaptive forecast correction toward a broader closed-loop decision framework for retail supply chains.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

- Aamer, A., L. Eka Yani, and I. Alan Priyatna (2021). Data analytics in the supply chain management: Review of machine learning applications in demand forecasting. *Operations and Supply Chain Management: An International Journal* 14(1), 1–13.
- Aburto, L. and R. Weber (2007). Improved supply chain management based on hybrid demand forecasts. *Applied Soft Computing* 7(1), 136–144.
- Adya, M. and F. Collopy (1998). How effective are neural networks at forecasting and prediction? a review and evaluation. *Journal of Forecasting* 17(5-6), 481–495.
- Alley, M., M. Biggs, R. Hariss, C. Herrmann, M. L. Li, and G. Perakis (2023). Pricing for heterogeneous products: Analytics for ticket reselling. *Manufacturing & Service Operations Management* 25(2), 409–426.
- Alon, I., M. Qi, and R. J. Sadowski (2001). Forecasting aggregate retail sales:: a comparison of artificial neural networks and traditional methods. *Journal of Retailing and Consumer Services* 8(3), 147–156.
- Andrews, R. L., I. S. Currim, P. Leeftang, and J. Lim (2008). Estimating the scan* pro model of store sales: Hb, fm or just ols? *International Journal of Research in Marketing* 25(1), 22–33.
- Arunraj, N. S. and D. Ahrens (2015). A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting. *International Journal of Production Economics* 170, 321–335.
- Bethune, L., D. Grangier, D. Busbridge, E. Gualdoni, M. Cuturi, and P. Ablyn (2025). Scaling laws for forgetting during finetuning with pretraining data injection. *arXiv Preprint arXiv:2502.06042*.
- Boone, T., R. Ganeshan, R. L. Hicks, and N. R. Sanders (2018). Can Google Trends improve your sales forecast? *Production and Operations Management* 27(10), 1770–1774.

- Boute, R. N., S. M. Disney, M. R. Lambrecht, B. Mantin, and J. Gijsbrechts (2022, April). Deep reinforcement learning for inventory control: A roadmap. *European Journal of Operational Research* 298(2), 401–412.
- Caro, F. and J. Gallien (2010). Inventory management of a fast-fashion retail network. *Operations Research* 58(2), 257–273.
- Chen, T. and C. Guestrin (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.
- Cohen, M. C., P.-E. Gras, A. Pentecoste, and R. Zhang (2022). *Demand prediction in retail: A practical guide to leverage data and predictive analytics*. Springer.
- Cui, R., S. Gallino, A. Moreno, and D. J. Zhang (2018). The operational value of social media information. *Production and Operations Management* 27(10), 1749–1769.
- Fahimnia, B., T. Tan, and N. Tahirov (2025). Service-level anchoring in demand forecasting: The moderating impact of retail promotions and product perishability. *International Journal of Forecasting* 41(2), 554–570.
- Feizabadi, J. (2022). Machine learning demand forecasting and supply chain performance. *International Journal of Logistics Research and Applications* 25(2), 119–142.
- Ferreira, K. J., B. H. A. Lee, and D. Simchi-Levi (2016). Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management* 18(1), 69–88.
- Fildes, R., S. Ma, and S. Kolassa (2022). Retail forecasting: Research and practice. *International Journal of Forecasting* 38(4), 1283–1318.
- Fildes, R., K. Nikolopoulos, S. F. Crone, and A. Syntetos (2008). Forecasting and operational research: A review. *Journal of the Operational Research Society* 59(9), 1150–1172.
- Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29(5), 1189–1232.
- Gao, C., N. Zhang, Y. Li, Y. Lin, and H. Wan (2023). Adversarial self-attentive time-variant neural networks for multi-step time series forecasting. *Expert Systems with Applications* 231, 120722.
- Gardner Jr, E. S. (1985). Exponential smoothing: The state of the art. *Journal of Forecasting* 4(1), 1–28.
- Geurts, M. D. and J. P. Kelly (1986). Forecasting retail sales using alternative models. *International Journal of Forecasting* 2(3), 261–272.
- Gijsbrechts, J., R. N. Boute, J. A. Van Mieghem, and D. J. Zhang (2022). Can deep reinforcement learning improve inventory management? performance on lost sales, dual-sourcing, and multi-echelon problems. *Manufacturing & Service Operations Management* 24(3), 1349–1368.
- Graves, A. and J. Schmidhuber (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks* 18(5-6), 602–610.
- Green, K. C. and J. S. Armstrong (2015). Simple versus complex forecasting: The evidence. *Journal of Business Research* 68(8), 1678–1685.

- Guo, X., K. C. Lichtendahl, and Y. Grushka-Cockayne (2025). Bayesian ensembles of exponentially smoothed life-cycle forecasts. *Manufacturing & Service Operations Management* 27(1), 230–248.
- Haarnoja, T., A. Zhou, P. Abbeel, and S. Levine (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr.
- Hochreiter, S. and J. Schmidhuber (1997). Long short-term memory. *Neural Computation* 9(8), 1735–1780.
- Huang, T., R. Fildes, and D. Soopramanien (2014). The value of competitive information in forecasting fmrg retail product sales and the variable selection problem. *European Journal of Operational Research* 237(2), 738–748.
- Huang, T., R. Fildes, and D. Soopramanien (2019). Forecasting retailer product sales in the presence of structural change. *European Journal of Operational Research* 279(2), 459–470.
- Hyndman, R. J. and G. Athanasopoulos (2018). *Forecasting: principles and practice*. OTexts.
- Ingle, C., D. Bakliwal, J. Jain, P. Singh, P. Kale, and V. Chhajed (2021). Demand forecasting: Literature review on various methodologies. In *2021 12th International Conference on Computing Communication and Networking Technologies*, pp. 1–7.
- Islam, S., S. H. Amin, and L. J. Wardley (2021). Machine learning and optimization models for supplier selection and order allocation planning. *International Journal of Production Economics* 242, 108315.
- Jin, B. and X. Xu (2025a). High-frequency CSI300 spot and futures price predictions via the neural network. *Journal of Uncertain Systems*, 2550008.
- Jin, B. and X. Xu (2025b). A study of contemporaneous residential real estate price causation across major Jiangsu province cities: Methodology using vector error-correction models and directed acyclic graphs. *Economics Open*, 2550008.
- Ke, G., Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems* 30.
- Khouja, M. and J. Zhou (2019). Early sale of seasonal inventory in the newsvendor problem. *IIE Transactions* 51(6), 672–689.
- Kim, B., J. G. Kim, and S. Lee (2024). A multi-agent reinforcement learning model for inventory transshipments under supply chain disruption. *IIE Transactions* 56(7), 715–728.
- Kirkpatrick, J., R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences* 114(13), 3521–3526.
- Lei, D., H. Hu, D. Geng, J. Zhang, Y. Qi, S. Liu, and Z.-J. M. Shen (2023). New product life cycle curve modeling and forecasting with product attributes and promotion: A Bayesian functional approach. *Production and Operations Management* 32(2), 655–673.

- Lei, D., Y. Qi, S. Liu, D. Geng, J. Zhang, H. Hu, and Z.-J. M. Shen (2024). Pooling and boosting for demand prediction in retail: A transfer learning approach. *Manufacturing & Service Operations Management*.
- Li, S., Y. He, H. Huang, J. Lin, and D. Ivanov (2024). Supply chain hoarding and contingent sourcing strategies in anticipation of price hikes and product shortages. *IIE Transactions* 56(7), 746–761.
- Li, X., Y. Xu, R. Law, and S. Wang (2024). Enhancing tourism demand forecasting with a transformer-based framework. *Annals of Tourism Research* 107, 103791.
- Ma, B. J., I. Jackson, M. Huang, S. Villegas, and J. Macias-Aguayo (2025). A data-driven and context-aware approach for demand forecasting in the beverage industry. *International Journal of Logistics Research and Applications*, 1–28.
- Ma, S., R. Fildes, and T. Huang (2016). Demand forecasting with high dimensional data: The case of sku retail sales forecasting with intra-and inter-category promotional information. *European Journal of Operational Research* 249(1), 245–257.
- Makridakis, S., A. Andersen, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, and R. Winkler (1982). The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of Forecasting* 1(2), 111–153.
- Makridakis, S. and M. Hibon (2000). The M3-competition: Results, conclusions and implications. *International Journal of Forecasting* 16(4), 451–476.
- Makridakis, S., M. Hibon, E. Lusk, and M. Belhadjali (1993). The M2-competition: A real-time judgmentally based forecasting study. *International Journal of Forecasting* 9(1), 5–22.
- Makridakis, S., E. Spiliotis, and V. Assimakopoulos (2022). M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting* 38(4), 1346–1364.
- Malviya, P. and V. Bhandari (2024). A systematic study on effective demand prediction using machine learning. *Journal of Integrated Science and Technology* 12(1), 711.
- Mamdouh, M., M. Ezzat, and H. Hefny (2024). Improving flight delays prediction by developing attention-based bidirectional lstm network. *Expert Systems with Applications* 238, 121747.
- Mandania, R. and F. S. Oliveira (2023). Dynamic pricing of regulated field services using reinforcement learning. *IIE Transactions* 55(10), 1022–1034.
- Mazumder, P., P. Singh, and P. Rai (2021). Few-shot lifelong learning. In *Proceedings of the AAAI conference on artificial intelligence*, Volume 35, pp. 2337–2345.
- McKinsey (2022). Stronger forecasting in operations management—even with weak data. Accessed December 19, 2025.
- Nikolopoulos, K., A. A. Syntetos, J. E. Boylan, F. Petropoulos, and V. Assimakopoulos (2011). An aggregate-disaggregate intermittent demand approach (ADIDA) to forecasting: An empirical proposition and analysis. *Journal of the Operational Research Society* 62(3), 544–554.
- Petropoulos, F., D. Apiletti, V. Assimakopoulos, M. Z. Babai, D. K. Barrow, S. B. Taieb, C. Bergmeir, R. J. Bessa, J. Bijak, J. E. Boylan, et al. (2022). Forecasting: theory and practice. *International Journal of Forecasting* 38(3), 705–871.

- Punia, S. and S. Shankar (2022). Predictive analytics for demand forecasting: A deep learning-based decision support system. *Knowledge-Based Systems* 258, 109956.
- Steinker, S., K. Hoberg, and U. W. Thonemann (2017). The value of weather information for e-commerce operations. *Production and Operations Management* 26(10), 1854–1874.
- Tao, X., X. Hong, X. Chang, S. Dong, X. Wei, and Y. Gong (2020). Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12183–12192.
- Taylor, J. W. (2003). Exponential smoothing with a damped multiplicative trend. *International Journal of Forecasting* 19(4), 715–725.
- Taylor, J. W. (2007). Forecasting daily supermarket sales using exponentially weighted quantile regression. *European Journal of Operational Research* 178(1), 154–167.
- Tian, S., L. Li, W. Li, H. Ran, X. Ning, and P. Tiwari (2024). A survey on few-shot class-incremental learning. *Neural Networks* 169, 307–324.
- Trapero, J. R., N. Kourentzes, and R. Fildes (2015). On the identification of sales forecasting models in the presence of promotions. *Journal of the Operational Research Society* 66(2), 299–307.
- Trapero, J. R., D. J. Pedregal, R. Fildes, and N. Kourentzes (2013). Analysis of judgmental adjustments in the presence of promotions. *International Journal of Forecasting* 29(2), 234–243.
- Wagner, M. R. (2015). Robust purchasing and information asymmetry in supply chains with a price-only contract. *IIE Transactions* 47(8), 819–840.
- Wang, X., H. Wang, S. Li, and H. Jin (2024). A reinforcement learning-based online learning strategy for real-time short-term load forecasting. *Energy* 305, 132344.
- Xu, X. and Y. Zhang (2021). Corn cash price forecasting with neural networks. *Computers and Electronics in Agriculture* 184, 106120.
- Xu, X. and Y. Zhang (2023). Price forecasts of ten steel products using Gaussian process regressions. *Engineering Applications of Artificial Intelligence* 126, 106870.
- Ye, L., N. Xie, J. E. Boylan, and Z. Shang (2024). Forecasting seasonal demand for retail: A fourier time-varying grey model. *International Journal of Forecasting* 40(4), 1467–1485.
- Zhang, G., L. Wang, G. Kang, L. Chen, and Y. Wei (2023). Slca: Slow learner with classifier alignment for continual learning on a pre-trained model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19148–19158.
- Zhang, G. P., Y. Xia, and M. Xie (2024). Intermittent demand forecasting with transformer neural networks. *Annals of Operations Research* 339(1), 1051–1072.
- Zhang, W., Y. Yu, Y. Qi, F. Shu, and Y. Wang (2019). Short-term traffic flow prediction based on spatio-temporal analysis and cnn deep learning. *Transportmetrica A: Transport Science* 15(2), 1688–1711.
- Zhong, F., Z. Zhou, and M. Leng (2020). Game-theoretic analyses of strategic pricing decision problems in supply chains. *IIE Transactions* 53(6), 704–718.

Supplemental Online Materials to “A Predict-then-Correct Loop Based on Few-Shot Continuous Contextual Bandit for Demand Forecasting”

Appendix A: Dataset Splitting

Table A1: Dataset and splitting

Dataset	Cluster	Quantity	Training	Validation	Forecast	Sliding window
Walmart	Stable & High Volume	217	Week 1–134	Week 135–143	Week 144–152	3
	Stable & Low Volume	3122	Week 1–134	Week 135–143	Week 144–152	3
	Erratic & Intermittent	3256	Week 1–134	Week 135–143	Week 144–152	3
Beverage	Stable & High Volume	52	Day 1–1765	Day 1766–1825	Day 1826–1885	3
	Stable & Low Volume	465	Day 1–1765	Day 1766–1825	Day 1826–1885	3
	Erratic & Intermittent	168	Day 1–1765	Day 1766–1825	Day 1826–1885	3

Appendix B: Principal Hyperparameter Settings

Table B1: Principal hyperparameter settings

Setting	Value
Batch size	64
Learning rate	3×10^{-4}
Optimizer	Adam
Hidden dimension	256
Action dimension	1
Observation dimension	15
Entropy coefficient α	0.02
Network layers	3
Parameter update ratio for the Walmart dataset	0.10
Parameter update ratio for the beverage dataset	0.16
Sliding-window length	3

Appendix C: Augmentation Data Selection

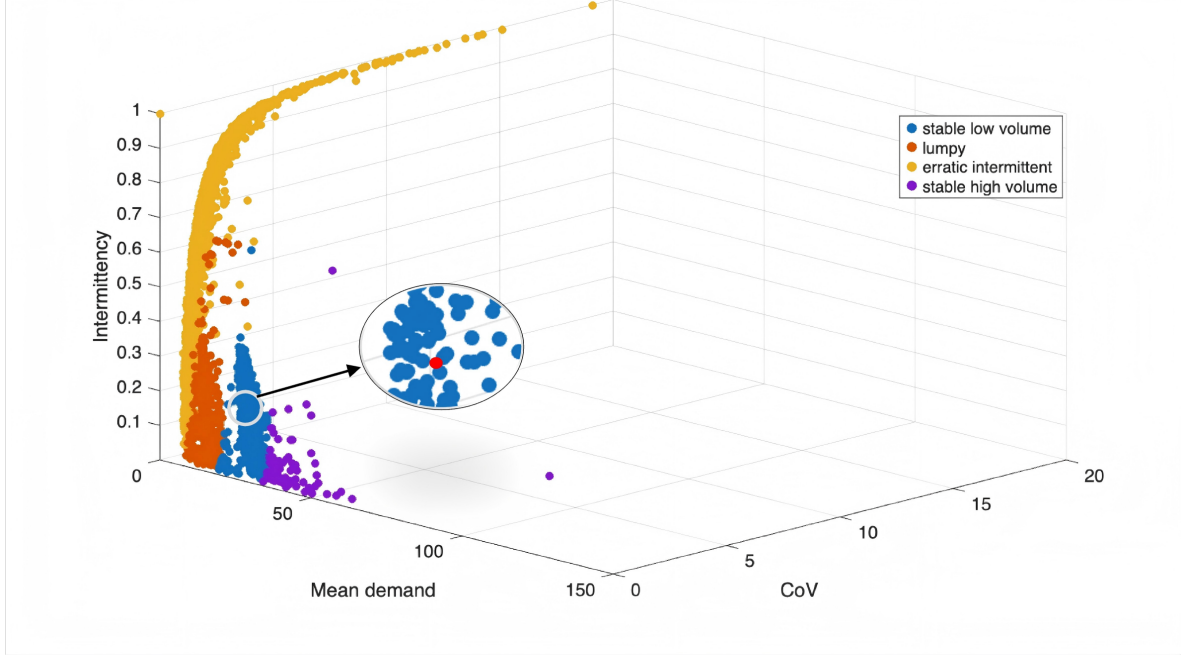


Figure C1: Augmentation data selection within each cluster.

Appendix D: Proof of Proposition 1

Proof of Proposition 1. Since $0 \in \mathcal{A}$ and $\pi_0(s_t) = 0$ belongs to the policy class Π , the PtC framework can choose the zero-correction policy. Under this policy, the corrected forecast becomes

$$\hat{y}_t^{PtC}(\pi_0) = \hat{y}_t^{ML}(1 + \pi_0(s_t)) = \hat{y}_t^{ML}.$$

Therefore, the ML-only prediction is a special case of the PtC prediction class. Since the PtC framework optimizes over a feasible policy class that contains the zero-correction policy, taking the infimum over Π cannot yield a larger expected loss than the loss attained by π_0 . Hence,

$$\inf_{\pi \in \Pi} \mathbb{E} \left[\frac{1}{T} \sum_{t=1}^T \ell(y_t, \hat{y}_t^{ML}(1 + \pi(s_t))) \right] \leq \mathbb{E} \left[\frac{1}{T} \sum_{t=1}^T \ell(y_t, \hat{y}_t^{ML}(1 + \pi_0(s_t))) \right].$$

Using $\pi_0(s_t) = 0$, the right-hand side reduces to

$$\mathbb{E} \left[\frac{1}{T} \sum_{t=1}^T \ell(y_t, \hat{y}_t^{ML}) \right].$$

□

Appendix E: Proof of Proposition 2

Proof of Proposition 2. Let $\mathcal{L}_{new}(\theta)$ denote the loss on the new demand cycle, and define the feasible parameter space under top- p updating as

$$\Theta_p = \{\theta^0 + \Delta : \text{supp}(\Delta) \subseteq S_p\}.$$

Then the adaptation gap

$$G_{ad}(p) = \inf_{\theta \in \Theta_p} \mathcal{L}_{new}(\theta) - \inf_{\theta \in \Theta_1} \mathcal{L}_{new}(\theta)$$

is monotone non-increasing with respect to p , and satisfies $G_{ad}(1) = 0$.

Moreover, suppose that the masked gradient update is performed for U steps:

$$\theta_p^{u+1} = \theta_p^u - \eta m_p \odot g_p^u, \quad u = 0, \dots, U-1,$$

where $\theta_p^0 = \theta^0$, $\eta > 0$ is the learning rate, and every gradient coordinate is bounded as $|(g_p^u)_j| \leq G$. Then the parameter drift satisfies

$$\|\theta_p^U - \theta^0\|_2 \leq \eta U G \sqrt{\lceil pd \rceil}.$$

Therefore, the upper bound on parameter drift is monotone non-decreasing with respect to p and equals zero when $p = 0$. If the old-cycle loss $\mathcal{L}_{old}(\theta)$ is L_{old} -Lipschitz continuous, then

$$\mathcal{L}_{old}(\theta_p^U) - \mathcal{L}_{old}(\theta^0) \leq L_{old} \eta U G \sqrt{\lceil pd \rceil},$$

which implies that the forgetting-risk bound is also monotone non-decreasing with respect to p .

We first prove the plasticity result. For any $p_1 \leq p_2$, the nesting property of the Top- p selected sets gives $S_{p_1} \subseteq S_{p_2}$. Therefore, any parameter perturbation supported on S_{p_1} is also supported on S_{p_2} , which implies

$$\Theta_{p_1} \subseteq \Theta_{p_2}.$$

Since Θ_{p_2} is a larger feasible parameter space, minimizing the new-cycle loss over Θ_{p_2} cannot yield a larger optimal value than minimizing over Θ_{p_1} . Hence,

$$\inf_{\theta \in \Theta_{p_2}} \mathcal{L}_{new}(\theta) \leq \inf_{\theta \in \Theta_{p_1}} \mathcal{L}_{new}(\theta).$$

Subtracting the constant full-update benchmark $\inf_{\theta \in \Theta_1} \mathcal{L}_{new}(\theta)$ from both sides gives

$$G_{ad}(p_2) \leq G_{ad}(p_1).$$

Thus, the adaptation gap is monotone nonincreasing in p . When $p = 1$, all parameters are allowed to update, so $\Theta_p = \Theta_1$ and therefore

$$G_{ad}(1) = 0.$$

We next prove the stability result. By the masked update rule,

$$\theta_p^U - \theta^0 = -\eta \sum_{u=0}^{U-1} m_p \odot g_p^u.$$

Taking the Euclidean norm and applying the triangle inequality yields

$$\|\theta_p^U - \theta^0\|_2 \leq \eta \sum_{u=0}^{U-1} \|m_p \odot g_p^u\|_2.$$

Since the mask m_p contains exactly $\lceil pd \rceil$ nonzero entries and each gradient coordinate is bounded by G , we have

$$\|m_p \odot g_p^u\|_2 \leq G\sqrt{\lceil pd \rceil}.$$

Therefore,

$$\|\theta_p^U - \theta^0\|_2 \leq \eta UG\sqrt{\lceil pd \rceil}.$$

Because $\lceil pd \rceil$ is monotone non-decreasing in p , the parameter-drift upper bound is also monotone non-decreasing in p . When $p = 0$, no parameter is selected for updating, so $m_p = 0$ and

$$\|\theta_0^U - \theta^0\|_2 = 0.$$

Finally, if $\mathcal{L}_{old}$ is L_{old} -Lipchitz continuous, then

$$\mathcal{L}_{old}(\theta_p^U) - \mathcal{L}_{old}(\theta^0) \leq L_{old}\|\theta_p^U - \theta^0\|_2.$$

Substituting the drift bound gives

$$\mathcal{L}_{old}(\theta_p^U) - \mathcal{L}_{old}(\theta^0) \leq L_{old}\eta UG\sqrt{\lceil pd \rceil}.$$

Thus, increasing p expands the adaptable parameter space and improves plasticity, but it also increases the upper bound on parameter drift and forgetting risk. This establishes the stability-plasticity trade-off of the top- p updating rule. $\square$

Appendix F: Proof of Proposition 3

Proof of Proposition 3. Define the smoothed expected correction objective

$$\mathcal{J}_\epsilon(\theta) = \mathbb{E}_{(s_t, y_t, \hat{y}_t^{ML}) \sim \mathcal{D}, a_t \sim \pi_\theta} [\rho_\epsilon(\hat{y}_t^{ML}(1 + a_t) - y_t)],$$

where $\rho_\epsilon(z) = \sqrt{z^2 + \epsilon^2}$. The smoothing parameter $\epsilon > 0$ avoids the non-differentiability of the absolute value while preserving the MAE-type correction objective as $\epsilon \rightarrow 0$. The notation $a_t \sim \pi_\theta$ means that the correction action at time t is generated by the policy parameterized by θ .

At update step u , the top- p rule defines a binary mask $m_u \in \{0, 1\}^d$. The masked policy update can be written as

$$\theta^{u+1} = \theta^u - \eta m_u \odot \nabla \mathcal{J}_\epsilon(\theta^u).$$

Let

$$d_u = m_u \odot \nabla \mathcal{J}_\epsilon(\theta^u).$$

Then $\theta^{u+1} - \theta^u = -\eta d_u$. Since $\mathcal{J}_\epsilon$ is L -smooth, the descent lemma gives

$$\mathcal{J}_\epsilon(\theta^{u+1}) \leq \mathcal{J}_\epsilon(\theta^u) + \nabla \mathcal{J}_\epsilon(\theta^u)^\top (\theta^{u+1} - \theta^u) + \frac{L}{2} \|\theta^{u+1} - \theta^u\|_2^2.$$

Substituting $\theta^{u+1} - \theta^u = -\eta d_u$ yields

$$\mathcal{J}_\epsilon(\theta^{u+1}) \leq \mathcal{J}_\epsilon(\theta^u) - \eta \nabla \mathcal{J}_\epsilon(\theta^u)^\top d_u + \frac{L\eta^2}{2} \|d_u\|_2^2.$$

Because m_u is a binary mask,

$$\nabla \mathcal{J}_\epsilon(\theta^u)^\top d_u = \nabla \mathcal{J}_\epsilon(\theta^u)^\top (m_u \odot \nabla \mathcal{J}_\epsilon(\theta^u)) = \|d_u\|_2^2.$$

Therefore

$$\mathcal{J}_\epsilon(\theta^{u+1}) \leq \mathcal{J}_\epsilon(\theta^u) - \eta \left(1 - \frac{L\eta}{2}\right) \|d_u\|_2^2.$$

If $0 < \eta \leq 1/L$, then $1 - L\eta/2 \geq 1/2$, and thus

$$\mathcal{J}_\epsilon(\theta^{u+1}) \leq \mathcal{J}_\epsilon(\theta^u) - \frac{\eta}{2} \|d_u\|_2^2 \leq \mathcal{J}_\epsilon(\theta^u).$$

This proves the monotonic decrease of the expected correction loss after each policy update. $\square$

Appendix G: Top- p Masked Bandit Update

Algorithm G1 Top- p masked bandit update for the CB correction policy

- 1: **Input:** Feedback data $\mathcal{D}_{new}$, initial policy π_{θ^0} , update ratio p , learning rate η , mini-batch size B , and update steps U
- 2: **Output:** Updated correction policy π_{θ^U}
- 3: **for** $u = 0, \dots, U - 1$ **do**
- 4: Sample a mini-batch $\mathcal{B}_u \subset \mathcal{D}_{new}$, where $|\mathcal{B}_u| = B$
- 5: For all $t \in \mathcal{B}_u$, compute $\mu_{\theta^u}(s_t)$ and $\sigma_{\theta^u}(s_t)$, sample

$$\tilde{a}_t \sim \mathcal{N}(\mu_{\theta^u}(s_t), \sigma_{\theta^u}^2(s_t)),$$

and set

$$a_t = \text{clip}(\tilde{a}_t, -1, 2)$$

- 6: Compute the corrected forecasts and rewards:

$$r_t = \tau(|\hat{y}_t^{ML} - y_t| - |\hat{y}_t^{CB} - y_t|)$$

- 7: Estimate the bandit policy gradient:

$$\hat{\nabla}_\theta J(\theta^u) = \frac{1}{|\mathcal{B}_u|} \sum_{t \in \mathcal{B}_u} (r_t - b_t) \nabla_\theta \log \pi_{\theta^u}(a_t | s_t)$$

- 8: For each layer k , construct mask M by selecting the smallest

$$q_k = \left\lceil p|\theta^k| \right\rceil$$

parameters according to $|\theta_j^k|$

- 9: Update:

$$\theta^{u+1} = \theta^u + \eta M \odot \hat{\nabla}_\theta J(\theta^u)$$

- 10: **end for**

- 11: **Return** π_{θ^U}
-