Belief-Aware Inventory Control with Deep Mixture Models

Moritz Beck[1], **Anh-Duy Pham**[1]
[1]Chair of Logistics and Quantitative Methods, Julius-Maximilians-Universität Würzburg
Sanderring 2, Würzburg 97070, Germany
{moritz.beck, anh-duy.pham}@uni-wuerzburg.de

Abstract

This paper introduces a framework that combines deep mixture models with inventory optimization under uncertain demand, tackling key challenges at the intersection of machine learning and operations research. We propose deep neural mixture models for demand forecasting that capture multimodal, bounded, and correlated patterns while maintaining computational tractability for downstream optimization. Our approach formulates inventory control as a partially observable Markov decision process (POMDP) where belief states over mixture components evolve via Bayesian updates. In order to enable practical implementation, we develop a belief space clustering approach using medoid clustering that reduces the belief space to a finite set of representative points. We provide theoretical guarantees including contraction properties of belief updates, Lipschitz continuity bounds, and explicit performance bounds under belief space reduction. The framework supports diverse neural architectures, including state-of-the-art deep learning time series forecasting models. Experiments on real-world pharmaceutical demand data demonstrate that the method is computationally efficient and leads to promising performance when the forecasts are well calibrated.

1 Introduction

The integration of machine learning and operations research has emerged as a critical frontier for data-centric decision making. Although deep learning has revolutionized time series forecasting with foundation models that achieve state-of-the-art performance [1, 2], translating these advances into operational decisions remains a challenge. Traditional "predict-then-optimize" approaches treat forecasting and optimization separately, failing to account for how forecast uncertainty propagates through decision processes [3, 4, 5, 6]. Foundation models like TimesFM [1] and Time-MoE [7] showcase large-scale time series modeling capabilities. Recent architectures use mixtures for probabilistic forecasting [8], with ToTo [2] demonstrating how such mixture-based approaches can be integrated into foundation models. However, integrating such expressive methods with tractable inventory optimization presents fundamental challenges: belief-state dynamics become intractable, policy computation scales poorly, and theoretical guarantees are difficult to establish. We address these challenges through three key contributions: **(1)** a flexible deep mixture framework compatible with any neural architecture for demand forecasting; based on a **(2)** belief-aware POMDP formulation with efficient medoid clustering for tractable policy computation; and **(3)** theoretical guarantees on convergence and approximation quality with explicit value function error bounds. Our approach combines state-of-the-art deep probabilistic forecasting with optimization in inventory management. It brings a practical advantage by reducing computational complexity while maintaining strong theoretical properties and showing promising empirical results.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Workshop MLxOR: Mathematical Foundations and Operational Integration of Machine Learning for Uncertainty-Aware Decision-Making.

2 Problem Statement: Fixed Cost Inventory Control with Backlogging

Consider a multi-product inventory system with products $m \in \mathcal{M}$, where demand $d_t^m \in \{0, 1, \ldots, D_{\max}^m\}$ exhibits complex temporal patterns over a finite planning horizon H. The forecast horizon H represents the number of future periods for which demand forecasts are generated and used in inventory optimization decisions. The inventory dynamics for each product m are governed by the following update equation:

$$I_{t+1}^m = I_t^m + q_t^m - d_t^m \tag{1}$$

where the order quantity $q_t^m \in \mathcal{A}(I_t^m) = \{\max(0, I_{\min}^m - I_t^m), \ldots, I_{\max}^m - I_t^m\}$ chosen from the action set $\mathcal{A}(I_t^m)$ is constrained by the bounds on the inventory target $I_{\min}^m \leq 0 \leq I_{\max}^m$. The per-period cost incorporates setup, holding, and backlog components:

$$c(I_t^m, q_t^m, d_t^m) = c_{\text{setup}}^m \cdot \mathbf{1}\{q_t^m > 0\} + c_{\text{hold}}^m [I_t^m + q_t^m - d_t^m]^+ + c_{\text{backlog}}^m [d_t^m - I_t^m - q_t^m]^+, \tag{2}$$

where c_{setup}^m, c_{hold}^m, and c_{backlog}^m are the respective cost parameters for product m. The objective is to minimize total expected costs over a finite number of periods.

3 Belief-Aware Inventory Control with Deep Mixture Models

3.1 Deep Mixture Models for Demand

We model demand distribution as a truncated and discretized Gaussian mixture over the bounded support $\{0, 1, \ldots, D_{\max}^m\}$:

$$P(d_{t+h}^m | \mathcal{F}_t^m) = \sum_{k=1}^{K} w_k^m \cdot \mathcal{TN}_{\text{disc}}(d_{t+h}^m | \mu_{k,t+h}^m, (\sigma_{k,t+h}^m)^2, [0, D_{\max}^m]), \tag{3}$$

where $\mathcal{TN}_{\text{disc}}$ denotes the probability mass function of the discretized truncated normal mixture, with the probability at each integer given by integrating the continuous truncated normal density over the unit-length bin centered at that integer. The mixture weights w_k^m satisfy $\sum_{k=1}^{K} w_k^m = 1$ and $w_k^m \geq 0$. The parameters $\mu_{k,t+h}^m$ and $\sigma_{k,t+h}^m$ are the location and scale of component k at horizon h. $\mathcal{F}_t^m$ is the information filtration up to time t. A neural network $\mathcal{N}_\Theta$ generates mixture parameters from historical demands $\mathbf{d}_{t-l}^m, l \in \{0, \cdots L-1\}$, exogenous features $\mathbf{x}_t^m$, and product attributes $\mathbf{z}^m$: $\boldsymbol{\theta}_{t+1,t+H}^m = \mathcal{N}_\Theta(\mathbf{d}_{t-L+1}^m, \cdots \mathbf{d}_t^m, \mathbf{x}_t^m, \mathbf{z}^m)$. The parameters are learned through maximum likelihood estimation on the horizon of H to capture multi-step dependencies in the mixture dynamics. The complete likelihood formulation and parameter estimation procedure is detailed in Appendix A.1.

3.2 POMDP Formulation and Belief Dynamics

Under the mixture model in Section 3.1, inventory control is a partially observable Markov decision process (POMDP): we observe the realized demand d_t^m but not the latent generating component $k \in \{1, \ldots, K\}$. Thus, we maintain a belief state $\mathbf{b}_t^m \in \Delta^{K-1}$ representing the posterior probability distribution over mixture components at time t: $\mathbf{b}_t^m = [b_{t,1}^m, b_{t,2}^m, \ldots, b_{t,K}^m]^T$, where $b_{t,k}^m = P(k_t^m = k|\mathcal{F}_t^m)$ is the probability that component k is the underlying one given the information filtration $\mathcal{F}_t^m$ up to time t. The belief state evolves via Bayesian updates. When demand d_t^m is observed at time t, the belief update follows:

$$b_{t+1,k}^m = \frac{b_{t,k}^m \cdot P(d_t^m|k)}{\sum_{j=1}^{K} b_{t,j}^m \cdot P(d_t^m|j)} \tag{4}$$

where $P(d_t^m|k)$ is the likelihood of observing demand d_t^m under mixture component k. The value function $V_{t+h}^{\text{mix}}(I_t^m, \mathbf{b}_t^m)$ represents the minimum expected cost-to-go from inventory level I_t^m and belief state $\mathbf{b}_t^m$ with h periods remaining. It satisfies the Bellman optimality equation:

$$V_{t+h}^{\text{mix}}(I_t^m, \mathbf{b}_t^m) = \min_{q_t^m \in \mathcal{A}(I_t^m)} Q_{t+h}^{\text{mix}}(I_t^m, \mathbf{b}_t^m, q_t^m) \tag{5}$$

The Q-value function decomposes the expected cost into immediate and future components:

$$Q_{t+h}^{\text{mix}}(I_t^m, \mathbf{b}_t^m, q_t^m) = \sum_{d=0}^{D_{\max}^m} P_{\mathbf{b}_t^m}(d) \left[c(I_t^m, q_t^m, d) + V_{t+h+1}^{\text{mix}}(I_{t+1}^m, \mathbf{b}_{t+1}^m)\right] \tag{6}$$

where $P_{\mathbf{b}_t^m}(d) = \sum_{k=1}^{K} b_{t,k}^m P(d|k)$ is the predictive demand distribution under belief $\mathbf{b}_t^m$. The next state $(I_{t+1}^m, \mathbf{b}_{t+1}^m) = T(I_t^m, \mathbf{b}_t^m, q_t^m, d_t^m)$ is determined by the inventory transition (1) and belief update (4). The optimal policy extracts the minimizing action: $\pi^*_{t+h}(I_t^m, \mathbf{b}_t^m) = \operatorname{argmin}_{q_t^m \in \mathcal{A}(I_t^m)} Q^{\text{mix}}_{t+h}(I_t^m, \mathbf{b}_t^m, q_t^m)$. Following [9], who proved the optimality of (s, S) policies under Markovian demand, we adopt this structure for settings with partially observed demand regimes. In our belief-aware setting we compute belief-dependent thresholds $s(b), S(b)$:

$$\pi^*_{t+h}(I_t^m, \mathbf{b}_t^m) = \begin{cases} S_{t+h}(\mathbf{b}_t^m) - I_t^m & \text{if } I_t^m \le s_{t+h}(\mathbf{b}_t^m) \\ 0 & \text{if } I_t^m > s_{t+h}(\mathbf{b}_t^m) \end{cases} \tag{7}$$

This policy adapts its reorder thresholds based on the current belief state: when beliefs concentrate on high-demand components, both $s(\mathbf{b}_t^m)$ and $S(\mathbf{b}_t^m)$ increase to accommodate higher expected demand. Conversely, when beliefs favor low-demand components, the thresholds decrease accordingly.

3.3 Belief Space Reduction via Medoid Clustering

The continuous belief space poses major computational challenges: while beliefs evolve on the $(K-1)$-dimensional probability simplex, the number of reachable belief states grows exponentially with horizon H, making exact dynamic programming intractable for realistic problem sizes. We approximate with N representative beliefs $\{\mathbf{b}_{t+h}^{m,n}\}_{n=1}^N$ chosen via medoid clustering to minimize:

$$\delta_{t+h}^m = \mathbb{E}_{\mathbf{b}_{t+h}^m}\left[\min_{n \in \{1,\ldots,N\}} \|\mathbf{b}_{t+h}^m - \mathbf{b}_{t+h}^{m,n}\|_1\right] \tag{8}$$

This enables finite-state dynamic programming while maintaining bounded approximation error. The projection-based approximation defines the reduced Q-value function $Q^{\text{reduced}}_{t+h}(I_t^m, \mathbf{b}_t^m, q_t^m)$ which operates on the discretized belief space:

$$Q^{\text{reduced}}_{t+h}(I_t^m, \mathbf{b}_t^m, q_t^m) = \sum_{d=0}^{D^m_{\max}} P_{\Pi_{t+h}(\mathbf{b}_t^m)}(d) \left[c(I_t^m, q_t^m, d) + V^{\text{reduced}}_{t+h+1}(I_{t+1}^m, \Pi_{t+h+1}(\mathbf{b}_{t+1}^m))\right] \tag{9}$$

where $\Pi_{t+h}(\mathbf{b})$ is the projection operator that maps any belief $\mathbf{b}$ to its nearest representative medoid, and $P_{\Pi_{t+h}(\mathbf{b}_t^m)}(d)$ is the predictive demand distribution under the projected belief. The complete belief-aware dynamic programming algorithm is presented in Appendix A.2.

3.4 Theoretical Guarantees

To establish a key approximation bound, we first define the cost bound constant B^m, which captures the maximum possible cost difference between any two policies in a single period:

$$B^m = \max(c^m_{\text{hold}} \cdot I^m_{\max}, c^m_{\text{backlog}} \cdot I^m_{\min} + c^m_{\text{setup}}) \tag{10}$$

Theorem 1 (Approximation Error)**.** *The gap between the full and reduced Q-values is bounded as:*

$$\left|Q^{\textit{mix}}_{t+h}(I^m, \mathbf{b}^m, q^m) - Q^{\textit{reduced}}_{t+h}(I^m, \mathbf{b}^m, q^m)\right| \;\le\; B^m \sum_{s=h}^{H-1} (H-s)^2\, \delta^m_{t+s}$$

This theoretical result establishes that the reduced policy converges to the optimal policy as the number of representative beliefs grows, with explicit bounds on the approximation quality determined by the clustering performance. A detailed proof of the theorem is provided in Appendix B.

4 Experiments

4.1 Experimental Setup

We evaluate on a pharmaceutical demand dataset with 459 SKUs and weekly observations over 2.3 years, similar to [5]. The dataset includes heterogeneous demand patterns, which are analyzed according to [10] in Appendix C.1. We allocate 80% of the data for training (February 2017–February

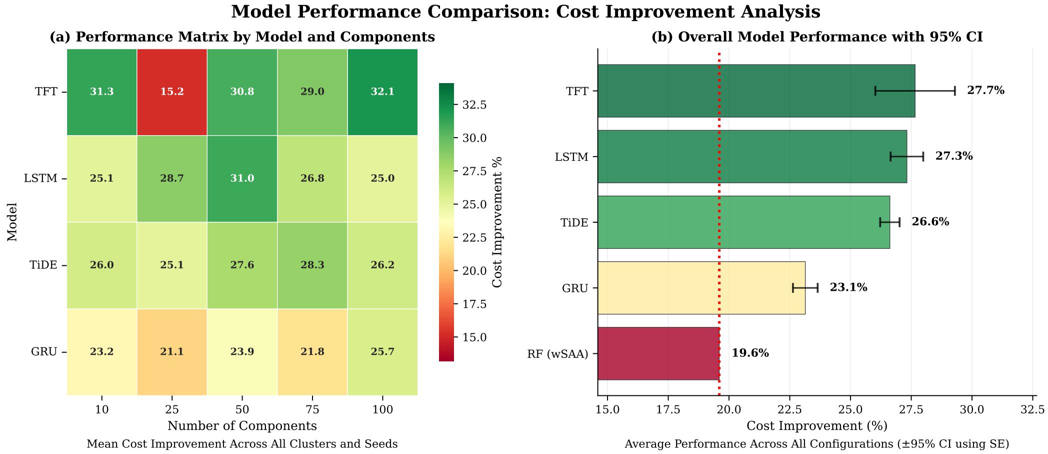

Figure 1: Model performance for different number of components (a) and across machine learning models (b). In (a), results are averaged over clusters; in (b), Error bars show 95% confidence intervals across configurations, calculated as $\pm 1.96 \times \frac{\text{std}}{\sqrt{n}}$.

2019) and 20% for evaluation (March 2019–May 2019). We set $c^m_{\text{hold}} = 1$ (per unit/week), $c^m_{\text{backlog}} = 3$ (per unit/week), and $c^m_{\text{setup}} = 100$ (per order) for all products m. We evaluate four probabilistic forecasting architectures: TiDE [11], the Temporal Fusion Transformer (TFT) [12], gated recurrent units (GRU) [13] and long short-term memory networks (LSTM) [14]. Model configurations compare mixture component counts $K = 1, 10, 25, 50, 75$, and 100 and medoid counts $N = 20, 50, 100$, and 500, using a forecast horizon $H = 10$. We compare against two baselines: **(1)** a static (s, S) policy per SKU via sample-average approximation (SAA) on the empirical training distribution, and **(2)** a periodically updated (s, S) policy via globally weighted SAA with a Random Forest kernel, as in [5].

4.2 Results

Figure 1 shows the cost reduction over the static (s, S) policy across different configurations, excluding single-component cases ($K = 1$). All architectures consistently outperform the static (s, S) baseline, and most surpass the Random Forest baseline, with the single exception of TFT at $K = 25$. Despite this outlier, TFT achieves the highest average improvement, followed by LSTM and TiDE, which perform slightly worse than TFT on average, but are more stable across different component counts. This is not entirely unexpected as transformer-based models like TFT can overfit and show less stable performance with limited data. Figure 3 in Section C.2.1 suggests that this outlier was caused by poor forecasting. Interestingly, TFT displays instability only at $K = 25$, while maintaining strong performance at both lower and higher component counts. The underlying reason for this anomaly is unclear and requires further investigation. However, since all other configurations outperform Random Forest, we can say that our method can effectively turn the output of state-of-the-art forecasting models into strong inventory decisions. Tables 2 and 3 in Appendix C.2 show that while the number of medoid clusters has only a minor impact on cost, it significantly increases runtime. Thus, choosing a moderate number of clusters results in little performance loss while giving substantial runtime gains.

5 Conclusion and Future Work

This work integrates deep mixture forecasting into inventory optimization, delivering a tractable POMDP formulation with theoretical guarantees. Experimental results show that the method is computationally efficient and delivers promising performance when probabilistic forecasts are reasonably accurate. Future work includes improving stability across mixture sizes, expanding to non-truncated distributions, and conducting a more comprehensive benchmark. The theory currently assumes a correctly specified mixture-based forecast and can be extended to handle misspecified forecasts.

References

[1] Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.

[2] Ben Cohen, Emaad Khwaja, Kan Wang, Charles Masson, Elise Ramé, Youssef Doubli, and Othmane Abou-Amal. Toto: Time series optimized transformer for observability. *arXiv preprint arXiv:2407.07874*, 2024.

[3] Gah-Yi Ban and Cynthia Rudin. The big data newsvendor: Practical insights from machine learning. *Operations Research*, 67(1):90–108, 2019.

[4] Pascal M Notz and Richard Pibernik. Prescriptive analytics for flexible capacity management. *Management Science*, 68(3):1756–1775, 2022.

[5] Felix G Schmidt and Richard Pibernik. Data-driven inventory control for large product portfolios: A practical application of prescriptive analytics. *European Journal of Operational Research*, 322(1):254–269, 2025.

[6] Adam N Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68(1):9–26, 2022.

[7] Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. *arXiv preprint arXiv:2409.16040*, 2024.

[8] Kin G Olivares, O Nganba Meetei, Ruijun Ma, Rohan Reddy, Mengfei Cao, and Lee Dicker. Probabilistic hierarchical forecasting with deep poisson mixtures. *International Journal of Forecasting*, 40(2):470–489, 2024.

[9] Suresh P Sethi and Feng Cheng. Optimality of (s, s) policies in inventory models with markovian demand. *Operations Research*, 45(6):931–939, 1997.

[10] Aris A Syntetos, John E Boylan, and JD Croston. On the categorization of demand patterns. *Journal of the operational research society*, 56(5):495–503, 2005.

[11] Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with tide: Time-series dense encoder. *arXiv preprint arXiv:2304.08424*, 2023.

[12] Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International journal of forecasting*, 37(4):1748–1764, 2021.

[13] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[15] Kin G. Olivares, Cristian Challú, Federico Garza, Max Mergenthaler Canseco, and Artur Dubrawski. NeuralForecast: User friendly state-of-the-art neural forecasting models. PyCon Salt Lake City, Utah, US 2022, 2022.

[16] Erich Schubert and Peter J. Rousseeuw. Faster k-medoids clustering: improving the pam, clara, and clarans algorithms. In *International Conference on Similarity Search and Applications*, pages 171–187, Cham, 2019. Springer International Publishing.

[17] Stéphane Gaubert and Zheng Qu. Dobrushin's ergodicity coefficient for markov operators on cones. *Integral Equations and Operator Theory*, 81(1):127–150, 2015.

A Supplementary Material and Technical Details

A.1 Likelihood Formulation and Parameter Estimation

This appendix provides the complete derivation of the likelihood function for training deep mixture models for demand forecasting applications.

A.1.1 Composite Likelihood Objective

The neural network parameters Θ are learned by maximizing the composite log-likelihood across all products and time periods in the training dataset:

$$\mathcal{L}(\Theta) = \sum_{m \in \mathcal{M}} \sum_{t=1}^{T-H} \log P(d_{t+1}^m, \ldots, d_{t+H}^m | \boldsymbol{\theta}_{t+1,t+H}^m), \tag{11}$$

where d_{t+h}^m represents the observed demand for product m in the forecast horizon h, and $\boldsymbol{\theta}_{t+1,t+H}^m$ are the mixture parameters generated by the neural network.

A.1.2 Parameter Constraints and Transformations

The neural network outputs raw parameters that are transformed to satisfy mixture model constraints:

Mixture weights: Normalized using softmax transformation:

$$w_k^m = \frac{\exp(\tilde{w}_k^m)}{\sum_{j=1}^K \exp(\tilde{w}_j^m)} \tag{12}$$

Scale parameters: Constrained to be positive using softplus:

$$\sigma_{k,t+h}^m = \log(1 + \exp(\tilde{\sigma}_{k,t+h}^m)) \tag{13}$$

Location parameters: Unconstrained since truncation handles support:

$$\mu_{k,t+h}^m = \tilde{\mu}_{k,t+h}^m \tag{14}$$

A.1.3 Additional Training Details

Hyperparameter Optimization All models receive identical ML-based hyperparameter tuning using Nixtla's optimization framework [15] to ensure fair comparison. The optimization process includes learning rate scheduling with adaptive decay, regularization parameter tuning (dropout, weight decay), architecture-specific parameters (hidden dimensions, layers), and early stopping based on a validation split.

Data Preprocessing For numerical stability and comparability across SKUs, we apply max-normalization:

$$d_{\text{normalized}} = \frac{d_{\text{original}}}{\max(d_{\text{historical}})}. \tag{15}$$

This transformation preserves zeros and rescales the training data to the unit interval [0,1]. The truncated mixture model is trained on this normalized support. Forecasts are then mapped back to the original scale using the inverse transformation.

$$\hat{d}_{\text{original}} = \hat{d}_{\text{normalized}} \cdot \max(d_{\text{historical}}). \tag{16}$$

A.2 Belief-Aware Dynamic Programming Algorithm

The belief-aware dynamic programming algorithm is detailed in Algorithm 1. The algorithm iteratively solves the dynamic programming problem over a moving forecast horizon, updating the inventory and belief states based on observed demands. We compute medoids using FastCLARA [16], which combines sampling with efficient distance computations to summarize the belief simplex. To ensure computational tractability, we constrain the inventory state space by setting the maximum inventory level $I_{\max}^m$ to 1.2 times the Economic Order Quantity (EOQ), where demand parameters are derived from the mean of the mixture model forecast. The minimum inventory target $I_{\min}^m$ is set to 0, meaning that an order must be placed if the inventory level is depleted. The algorithm is integrated with a rolling horizon framework, where at each time step t, the first-period action is executed, and the process repeats with updated inventory and belief states.

Algorithm 1 Belief-Aware Dynamic Programming

```
procedure BELIEFAWAREDYNAMICPROGRAMMING
    Input:
    • Current inventory $I_t^m$ for product $m$ at time $t$
    • Current belief state $\mathbf{b}_t^m$ for product $m$ at time $t$
    • Forecast parameters $\boldsymbol{\theta}_{t+h}^m$ for $h \in \{0, \dots, H-1\}$ over rolling horizon $H$
          (including mixture weights, locations, and scales generated by the neural network)
    • Medoids $\{\mathbf{b}_{t+h}^{m,n}\}_{n=1}^N$ over the forecast horizon $h \in \{0, \dots, H-1\}$
          (computed via FastCLARA clustering)
    Output:
    • First-period order quantity $q_t^m$
    Dynamic Programming Phase:
    • Initialize the terminal values: $V_{t+H}^{\text{reduced}}(I, \mathbf{b}_{t+H}^{m,n}) = 0$ for all inventory levels and medoids
    •     for $h = H-1$ down to $0$ do                                        ▷ Backward induction
    •         for each medoid $\mathbf{b}_{t+h}^{m,n}$ $(n = 1, \dots, N)$ do
    •             for each inventory level $I \in \{I_{\min}^m, \dots, I_{\max}^m\}$ do
    •                 $V_{t+h}^{\text{reduced}}(I, \mathbf{b}_{t+h}^{m,n}) = \min_{q \in \mathcal{A}(I)} Q_{t+h}^{\text{reduced}}(I, \mathbf{b}_{t+h}^{m,n}, q)$
    •             end for
    •         end for
    • Extract the $h$-period policy $(s_{t+h}(\mathbf{b}_{t+h}^{m,n}), S_{t+h}(\mathbf{b}_{t+h}^{m,n}))$ for each medoid based on the computed value function
    •     end for
    • Extract the first-period action based on $s(\mathbf{b}_t^m), S(\mathbf{b}_t^m)$ and the current inventory $I_t^m$
    end procedure
```

B Theoretical Analysis and Proofs

In this section of the supplementary material, we establish the theoretical foundations of our belief-aware inventory control framework. We first present two key lemmas that establish fundamental properties of the belief update dynamics and the regularity of the value function. These lemmas then enable us to prove our main approximation result, which provides explicit bounds on the approximation error and guarantees convergence of our algorithm.

B.1 Stability and Convergence Properties

Contraction of Belief Updates. Belief updates under the mixture model are contractive in expectation, which ensures stability of the dynamic programming recursion.

Lemma 1 (Belief Update Contraction)**.** *For product m, let $\eta^m = \min_{k,d} P^m(d \mid k) > 0$ be the minimum emission probability for that product. The Bayesian belief update defined by* (4) *satisfies the following contraction property for any two beliefs $\mathbf{b}_t^m, \mathbf{b'}_t^m \in \Delta^{K-1}$:*

$$\mathbb{E}_{d^m \sim P^m_{\mathbf{b}^m}} \left[\left\| \mathbf{b}_{t+1}^m - \mathbf{b'}_{t+1}^m \right\|_1 \right] \leq (1 - \eta^m) \, \| \mathbf{b}_t^m - \mathbf{b'}_t^m \|_1.$$

Proof. This result can be derived from the Dobrushin ergodicity coefficient, which bounds the total variation contraction under a common stochastic operator, see [17]. Here, the likelihood function induces a transition kernel whose Dobrushin coefficient is at most $1 - \eta^m < 1$. Hence, Bayesian belief updates contract in expectation under this bound. □

Lipschitz Regularity of the Q-Value Function. The Q-value function is Lipschitz continuous in the belief state, with a constant that grows quadratically with the remaining horizon.

Lemma 2 (Lipschitz Continuity of Q-Value)**.** *For any inventory levels $I^m \in \{I_{\min}^m, \dots, I_{\max}^m\}$ orders $q^m \in \mathcal{A}(I^m)$ and beliefs $\mathbf{b}^m, \mathbf{b'}^m \in \Delta^{K-1}$ for product m, the Q-value function satisfies:*

$$|Q_{t+h}^{mix}(I^m, \mathbf{b}^m, q^m) - Q_{t+h}^{mix}(I^m, \mathbf{b'}^m, q^m)| \leq L_h^m \| \mathbf{b}^m - \mathbf{b'}^m \|_1, \tag{17}$$

where $L_h^m = B^m \cdot (H-h)^2$ *and* B^m *is the cost bound constant defined in* (10).

Proof. We proceed by backward induction.

Base Case ($h = H$): The terminal Q-value function $Q^{\text{mix}}_{t+H}(I^m, \mathbf{b}^m, q^m) = 0$ is trivially Lipschitz with $L_H^m = 0$.

Inductive Step ($h < H$): Assume $Q^{\text{mix}}_{t+h+1}(I^m, \mathbf{b}^m, q^m)$ is L^m_{h+1}-Lipschitz. Then also $V^{\text{mix}}_{t+h+1}(I^m, \mathbf{b}^m)$ is Lipschitz with the same constant. For any $\mathbf{b}^m, \mathbf{b'}^m \in \Delta^{K-1}$:

$$\begin{aligned}
&\left|Q^{\text{mix}}_{t+h}(I^m, \mathbf{b'}^m, q^m) - Q^{\text{mix}}_{t+h}(I^m, \mathbf{b}^m, q^m)\right| \\
&\leq \underbrace{\left|\mathbb{E}_{d^m \sim P_{\mathbf{b'}^m}}[c(I^m, q^m, d^m)] - \mathbb{E}_{d^m \sim P_{\mathbf{b}^m}}[c(I^m, q^m, d^m)]\right|}_{\text{Current cost (A)}} \\
&\quad + \underbrace{\left|\mathbb{E}_{d^m \sim P_{\mathbf{b'}^m}}\left[V^{\text{mix}}_{t+h+1}\big(T(I^m, \mathbf{b'}^m, q^m, d^m)\big)\right] - \mathbb{E}_{d^m \sim P_{\mathbf{b}^m}}\left[V^{\text{mix}}_{t+h+1}\big(T(I^m, \mathbf{b}^m, q^m, d^m)\big)\right]\right|}_{\text{Future cost (B)}}.
\end{aligned}$$

Bounding Term (A):

$$\begin{aligned}
\text{(A)} &= \left|\mathbb{E}_{d^m \sim P_{\mathbf{b'}^m}}[c(I^m, q^m, d^m)] - \mathbb{E}_{d^m \sim P_{\mathbf{b}^m}}[c(I^m, q^m, d^m)]\right| \\
&\leq \max(c^m_{\text{hold}}, c^m_{\text{backlog}}) \cdot D^m_{\max} \cdot \|\mathbf{b'}^m - \mathbf{b}^m\|_1 \\
&\leq B^m \cdot (H-h) \cdot \|\mathbf{b'}^m - \mathbf{b}^m\|_1
\end{aligned}$$

where the first inequality follows from the cost being Lipschitz in demand and the second inequality follows from the definition of B^m.

Bounding Term (B):

$$\begin{aligned}
\text{(B)} &= \left|\mathbb{E}_{d^m \sim P_{\mathbf{b'}^m}}\left[V^{\text{mix}}_{t+h+1}\big(T(I^m, \mathbf{b'}^m, q^m, d^m)\big)\right] - \mathbb{E}_{d^m \sim P_{\mathbf{b}^m}}\left[V^{\text{mix}}_{t+h+1}\big(T(I^m, \mathbf{b}^m, q^m, d^m)\big)\right]\right| \\
&= \left|\sum_{d^m} V^{\text{mix}}_{t+h+1}\big(T(I^m, \mathbf{b'}^m, q^m, d^m)\big) P_{\mathbf{b'}^m}(d^m) - V^{\text{mix}}_{t+h+1}\big(T(I^m, \mathbf{b}^m, q^m, d^m)\big) P_{\mathbf{b}^m}(d^m)\right| \\
&\leq \sum_{d^m} \left|V^{\text{mix}}_{t+h+1}\big(T(I^m, \mathbf{b'}^m, q^m, d^m)\big) - V^{\text{mix}}_{t+h+1}\big(T(I^m, \mathbf{b}^m, q^m, d^m)\big)\right| P_{\mathbf{b'}^m}(d^m) \\
&\quad + \sum_{d^m} \left|V^{\text{mix}}_{t+h+1}\big(T(I^m, \mathbf{b}^m, q^m, d^m)\big)\right| \cdot \left|P_{\mathbf{b'}^m}(d^m) - P_{\mathbf{b}^m}(d^m)\right|.
\end{aligned}$$

Taking expectations and applying the inductive Lipschitz bound and the cost magnitude bound gives

$$\begin{aligned}
\text{(B)} &\leq L_{h+1} \cdot \mathbb{E}_{d^m \sim P_{\mathbf{b'}^m}} \left\|T(I^m, \mathbf{b'}^m, q^m, d^m) - T(I^m, \mathbf{b}^m, q^m, d^m)\right\|_1 \\
&\quad + B^m(H-h-1)^2 \|\mathbf{b'}^m - \mathbf{b}^m\|_1 \\
&\leq L_{h+1}\|\mathbf{b'}^m - \mathbf{b}^m\|_1 + B^m(H-h-1)^2\|\mathbf{b'}^m - \mathbf{b}^m\|_1 \\
&\leq \big(B^m(H-h-1) + B^m(H-h-1)^2\big)\|\mathbf{b'}^m - \mathbf{b}^m\|_1,
\end{aligned}$$

where the second inequality follows from Lemma 1 applied to product m and the third inequality follows from the inductive hypothesis $L_{h+1} \leq B^m(H-h-1)^2$.

Combining Bounds:

$$\begin{aligned}
&\left|Q^{\text{mix}}_{t+h}(I^m, \mathbf{b}^m, q^m) - Q^{\text{mix}}_{t+h}(I^m, \mathbf{b'}^m, q^m)\right| \\
&\leq \text{(A)} + \text{(B)} \\
&\leq B^m(H-h)\|\mathbf{b'}^m - \mathbf{b}^m\|_1 + B^m(H-h-1)^2\|\mathbf{b'}^m - \mathbf{b}^m\|_1 \\
&\leq B^m(H-h)^2\|\mathbf{b'}^m - \mathbf{b}^m\|_1,
\end{aligned}$$

which yields $L_h \leq B^m(H-h)^2$ as claimed. □

B.2 Proof of Main Approximation Result

Proof. We show that for any time step $t+h$, the error in the Q-value function due to belief space reduction is bounded by a term that grows quadratically with the remaining horizon and linearly with the expected distance to the nearest representative belief. Let therefore $\Delta^{Q,m}_{t+h}(I^m, q^m) = \mathbb{E}_{\mathbf{b}^m}\left[|Q^{\text{mix}}_{t+h}(I^m, \mathbf{b}^m, q^m) - Q^{\text{reduced}}_{t+h}(I^m, \mathbf{b}^m, q^m)|\right]$ be the expected error in the Q-value function at time $t+h$ for observable state I^m and action q^m. Let also be $\epsilon^m_{t+h} = B^m \sum_{s=h}^{H-1}(H-s)^2\delta^m_{t+s}$ be the approximation error at time $t+h$ for product m. We need to show that $\Delta^{Q,m}_{t+h}(I^m, q^m) \leq \epsilon^m_{t+h}$ for all I^m and q^m. We proceed by backward induction on the time step h.

Base Case ($h = H$): At the terminal time step $t+H$, we have: $Q^{\text{mix}}_{t+H} = Q^{\text{reduced}}_{t+H} = 0$ for all beliefs $\mathbf{b}^m$ and actions q^m, so the error is zero: $\Delta^{Q,m}_{t+H}(I^m, q^m) = 0$. Thus, the base case holds trivially with $\epsilon^m_{t+H} = 0$.

Inductive Step: Assume the statement holds for $h+1$, i.e., $\Delta^{Q,m}_{t+h+1}(I^m, q^m) \leq \epsilon^m_{t+h+1}$ for all observable states I^m and actions q^m. This also implies that the value function at time $t+h+1$ satisfies:

$$\mathbb{E}_{\mathbf{b}^m}\left[|V^{\text{mix}}_{t+h+1}(I^m, \mathbf{b}^m) - V^{\text{reduced}}_{t+h+1}(I^m, \mathbf{b}^m)|\right] \leq \epsilon^m_{t+h+1}$$

by the definition of the value function as the minimum Q-value over actions.

Then at time $t+h$:

$$\begin{aligned}
\Delta^{Q,m}_{t+h}(I^m, q^m) &= \mathbb{E}_{\mathbf{b}^m}\left[|Q^{\text{mix}}_{t+h}(I^m, \mathbf{b}^m, q^m) - Q^{\text{reduced}}_{t+h}(I^m, \mathbf{b}^m, q^m)|\right] \\
&= \mathbb{E}_{\mathbf{b}^m}\left[\mathbb{E}_{d^m \sim P_{\mathbf{b}^m}}\left[c(I^m, q^m, d^m) + V^{\text{mix}}_{t+h+1}(T(I^m, \mathbf{b}^m, q^m, d^m))\right.\right. \\
&\quad \left.\left.-\mathbb{E}_{d^m \sim P_{\mathbf{b}^m}}\left[c(I^m, q^m, d^m) + V^{\text{reduced}}_{t+h+1}(T(I^m, \mathbf{b}^m, q^m, d^m))\right]\right]\right] \\
&= \mathbb{E}_{\mathbf{b}^m}\left[\mathbb{E}_{d^m \sim P_{\mathbf{b}^m}}\left[V^{\text{mix}}_{t+h+1}(T(I^m, \mathbf{b}^m, q^m, d^m))\right.\right. \\
&\quad \left.\left.-V^{\text{reduced}}_{t+h+1}(T(I^m, \mathbf{b}^m, q^m, d^m))\right]\right]
\end{aligned}$$

Let now for given d^m the next inventory-belief pair be $(I^m_{\text{next}}, \mathbf{b}^m_{\text{next}}) = T(I^m, \mathbf{b}^m, q^m, d^m)$ and let $\mathbf{b}'^m_{\text{next}}$ be the nearest representative belief in the reduced belief space $\{\mathbf{b}^{m,n}_{t+h+1}\}_{n=1}^N$ to $\mathbf{b}^m_{\text{next}}$. Then we have:

$$\begin{aligned}
&\left|V^{\text{mix}}_{t+h+1}(T(I^m, \mathbf{b}^m, q^m, d^m)) - V^{\text{reduced}}_{t+h+1}(T(I^m, \mathbf{b}^m, q^m, d^m))\right| \\
&= \left|V^{\text{mix}}_{t+h+1}(I^m_{\text{next}}, \mathbf{b}^m_{\text{next}}) - V^{\text{reduced}}_{t+h+1}(I^m_{\text{next}}, \mathbf{b}'^m_{\text{next}})\right| \\
&\leq \left|V^{\text{mix}}_{t+h+1}(I^m_{\text{next}}, \mathbf{b}^m_{\text{next}}) - V^{\text{mix}}_{t+h+1}(I^m_{\text{next}}, \mathbf{b}'^m_{\text{next}})\right| \\
&\quad + \left|V^{\text{mix}}_{t+h+1}(I^m_{\text{next}}, \mathbf{b}'^m_{\text{next}}) - V^{\text{reduced}}_{t+h+1}(I^m_{\text{next}}, \mathbf{b}'^m_{\text{next}})\right|
\end{aligned}$$

Applying the expectation over the demand distribution $P_{\mathbf{b}^m}$ and using the Lipschitz continuity of the value function from Lemma 2, we obtain:

$$
\begin{aligned}
&\mathbb{E}_{\mathbf{b}^m}\left[\mathbb{E}_{d^m\sim P_{\mathbf{b}^m}}\left[\underbrace{|V^{\text{mix}}_{t+h+1}(I^m_{\text{next}},\mathbf{b}^m_{\text{next}})-V^{\text{mix}}_{t+h+1}(I^m_{\text{next}},\mathbf{b}'^m_{\text{next}})|}_{\text{Approximation error}}\right.\right.\\
&\qquad\qquad\qquad\left.\left.+\underbrace{|V^{\text{mix}}_{t+h+1}(I^m_{\text{next}},\mathbf{b}'^m_{\text{next}})-V^{\text{reduced}}_{t+h+1}(I^m_{\text{next}},\mathbf{b}'^m_{\text{next}})|}_{\text{Inductive error}}\right]\right]\\
&\leq \mathbb{E}_{\mathbf{b}^m}\left[L^m_{h+1}\|\mathbf{b}^m_{\text{next}}-\mathbf{b}'^m_{\text{next}}\|_1+\epsilon^m_{t+h+1}\right]\\
&\leq \mathbb{E}_{\mathbf{b}^m}\left[L^m_{h+1}\delta^m_{t+h+1}+\epsilon^m_{t+h+1}\right]\\
&\leq B^m(H-h-1)^2\delta^m_{t+h+1}+B^m\sum_{s=h+1}^{H-1}(H-s)^2\delta^m_{t+s}\\
&= B^m\sum_{s=h}^{H-1}(H-s)^2\delta^m_{t+s}
\end{aligned}
$$

This shows that the expected error in the Q-value function at time $t+h$ is bounded by the approximation error ϵ^m_{t+h}, completing the inductive step. Thus, by induction, the statement holds for all $h \in \{0, 1, \ldots, H-1\}$. In particular, for $h = 0$, we have:

$$
\Delta^{Q,m}_t(I^m, q^m) \leq B^m \sum_{h=0}^{H-1}(H-h)^2\delta^m_{t+h} \tag{18}
$$

□

C Additional Experimental Details

C.1 Exploratory Data Analysis of the Pharmaceutical Demand Dataset

The pharmaceutical demand dataset comprises 459 unique SKUs from a pharmacy supply chain, exhibiting diverse demand patterns as illustrated in Figure 2 and Table 1. The dataset reveals substantial heterogeneity in demand behavior: while some SKUs show consistent weekly demand reaching over $77,000$ units per period (Figure 2d), others remain dormant for extended periods with sporadic orders. This variability is captured through the Syntetos-Boylan classification framework [10] (Figure 2a), which segments SKUs into four distinct patterns based on their demand frequency and variability. The smooth pattern encompasses 180 SKUs with regular demand and low variability, accounting for a dominant 82.9% of total demand volume. In contrast, 170 SKUs fall into intermittent or lumpy categories, characterized by zero demand in approximately 48% of periods (Table 1) and coefficients of variation exceeding 1.5 (Table 1). The erratic pattern, comprising 109 SKUs, represents an intermediate case with regular but highly variable demand, contributing 16.5% to the total demand volume. This distribution, where 39.2% of the SKUs generate over 80% of the demand volume while most exhibit irregular patterns, reflects typical inventory challenges where high-runners coexist with slow-moving specialty items. The dataset includes a rich set of exogenous features, such as country-specific holiday counts, weekly averages of humidity, precipitation, and temperature (minimum and maximum) for Ghana, Kenya, Nigeria, Tanzania, and Uganda. These variables provide important context for modeling demand fluctuations driven by weather and calendar effects. In addition, we augment the feature set with temporal indicators (e.g., month), the mean demand observed in the training set, and lagged demand values for each of the previous 12 weeks to capture seasonality and correlation in the time series.

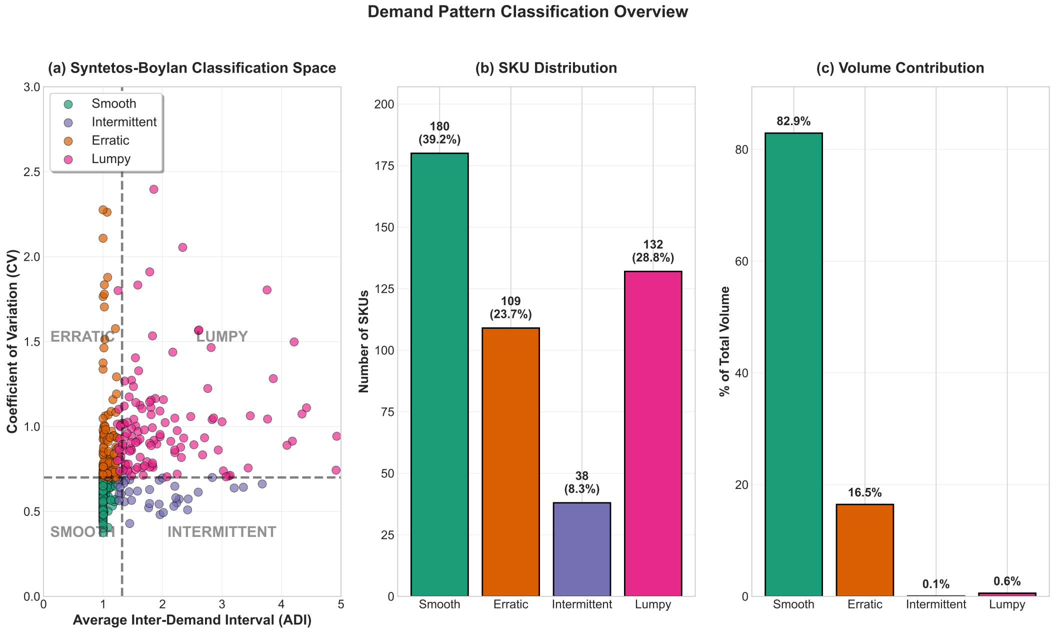

Figure 2: Demand pattern classification overview for pharmacy SKUs. (a) Syntetos-Boylan classification space showing the distribution of 459 SKUs based on Average Inter-Demand Interval (ADI) and Coefficient of Variation (CV), with threshold values of $ADI = 1.32$ and $CV = 0.7$ separating four demand patterns. (b) Distribution of SKUs across demand patterns. (c) Volume contribution by pattern, demonstrating that smooth items represent 82.9% of total demand despite comprising only 39.2% of SKUs. (d) Representative time series examples for each demand pattern category.

Table 1: Demand statistics by pattern. The table summarizes SKU counts, contribution to total volume, average coefficient of variation (Avg CV), and proportion of zero-demand periods (Zero %).

Pattern	SKUs	Volume (%)	Avg CV	Zero %
Smooth	180	82.9	0.59	1.9
Erratic	109	16.5	1.14	6.4
Intermittent	38	0.1	1.58	48.3
Lumpy	132	0.6	2.03	47.8

C.2 Detailed Inventory Performance Analysis

C.2.1 Probabilistic Forecasting Performance and Inventory Cost Relationship

To understand the connection between forecasting quality and inventory performance, we evaluate the probabilistic forecasting accuracy of our neural mixture models with the scaled Continuous Ranked Probability Score (sCRPS), which assesses how well the predicted probability distributions match observed demand realizations:

$$\text{sCRPS} = \frac{\text{CRPS}(F, d)}{\max(d_{\text{historical}})}$$

where d denotes the observed demand and $\text{CRPS}(F, d)$ is the Continuous Ranked Probability Score between the forecast distribution F and the observed demand d:

$$\text{CRPS}(F, d) = \int_{-\infty}^{\infty} \left[F(x) - \mathbf{1}\{d \leq x\}\right]^2 dx$$

The denominator in sCRPS rescales scores to enable fair comparisons across SKUs with different demand magnitudes. Lower sCRPS values indicate more accurate probabilistic forecasts.

Table 2: Model performance as % improvement over the (s, S) policy for different numbers of clusters. Results are averaged across the number of components. Runtime is measured per period.

Components	Clusters	Mean Imp. (%)	Std Imp. (%)	Mean Runtime (s)	Std Runtime (s)
1	20	-122.98	80.71	0.272	0.027
	50	-120.71	79.88	0.325	0.082
	100	-122.64	82.98	0.344	0.100
	500	-124.87	83.67	0.290	0.038
10	20	26.95	2.98	0.346	0.030
	50	26.89	3.04	0.401	0.088
	100	27.22	3.20	0.463	0.100
	500	24.54	3.57	0.610	0.057
25	20	23.28	5.36	0.381	0.033
	50	23.04	5.48	0.453	0.098
	100	23.09	4.96	0.532	0.117
	500	20.64	5.39	0.867	0.065
50	20	28.99	3.20	0.502	0.074
	50	28.74	3.39	0.589	0.104
	100	29.15	3.13	0.718	0.131
	500	26.40	2.67	1.336	0.045
75	20	26.92	3.43	0.542	0.057
	50	26.95	2.47	0.695	0.103
	100	26.63	2.74	0.800	0.116
	500	25.32	3.48	1.779	0.104
100	20	27.74	3.26	0.648	0.057
	50	27.18	2.71	0.771	0.120
	100	27.37	3.05	0.969	0.124
	500	26.68	2.94	2.297	0.140

Table 3: Model performance as % improvement over the (s, S) policy for different numbers of clusters. Results are averaged across the number of components. Runtime is measured per period.

Model	Clusters	Mean Imp. (%)	Std Imp. (%)	Mean Runtime (s)	Std Runtime (s)
GRU	20	23.60	1.89	0.508	0.143
	50	23.81	1.44	0.682	0.169
	100	23.59	1.66	0.771	0.220
	500	21.56	2.25	1.443	0.668
LSTM	20	27.94	2.68	0.465	0.117
	50	27.97	2.84	0.528	0.168
	100	27.69	2.55	0.740	0.203
	500	25.68	2.00	1.404	0.646
TFT	20	28.36	6.57	0.519	0.115
	50	27.69	6.44	0.636	0.155
	100	28.29	6.35	0.727	0.214
	500	26.30	6.97	1.407	0.634
TiDE	20	27.20	1.29	0.443	0.106
	50	26.77	1.55	0.481	0.129
	100	27.19	1.13	0.549	0.173
	500	25.32	1.62	1.257	0.581

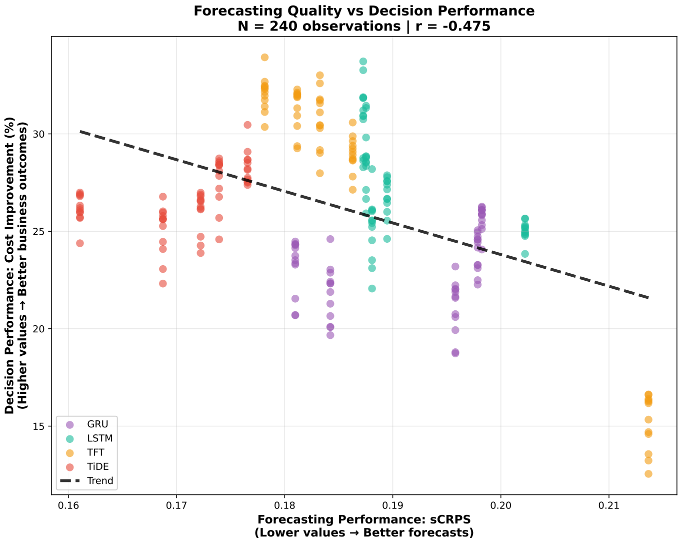

Figure 3: Correlation between forecasting accuracy (sCRPS) and cost reduction percentage over the (s, S) policy for different model configurations (mixture components K and medoid counts N).