A Practical End-to-End Inventory Management Model with Deep Learning

Meng Qi*
Industrial Engineering and Operations Research Department, University of California, Berkeley, meng$_q i@berkeley.edu$

Yuanyuan Shi*
Department of Electrical and Computer Engineering, University of California, San Diego yyshi@eng.ucsd.edu

Yongzhi Qi
JD.com Smart Supply Chain Y, qiyongzhi1@jd.com

Chenxin Ma, Rong Yuan, Di Wu
JD.com Silicon Valley Research Center, chenxin.ma, rong.yuan, di.wu@jd.com

Zuojun (Max) Shen
Industrial Engineering and Operations Research Department, University of California, Berkeley, maxshen@berkeley.edu

We investigate a data-driven multi-period inventory replenishment problem with uncertain demand and vendor lead time (VLT), with accessibility to a large quantity of historical data. Different from the traditional two-step predict-then-optimize (PTO) solution framework, we propose a one-step end-to-end (E2E) framework that uses deep-learning models to output the suggested replenishment amount directly from input features without any intermediate step. The E2E model is trained to capture the behavior of the optimal dynamic programming solution under historical observations, without any prior assumptions on the distributions of the demand and the VLT. By conducting a series of thorough numerical experiments using real data from one of the leading e-commerce companies, we demonstrate the advantages of the proposed E2E model over conventional PTO frameworks. We also conduct a field experiment with JD.com and the results show that our new algorithm reduces holding cost, stockout cost, total inventory cost and turnover rate substantially compared to JD's current practice. For the supply-chain management industry, our E2E model shortens the decision process and provides an automatic inventory management solution with the possibility to generalize and scale. The concept of E2E, which uses the input information directly for the ultimate goal, can also be useful in practice for other supply-chain management circumstances.

Key words: end-to-end, inventory management, deep learning

1. Introduction

Inventory management has been an active research topic in management science for over a century. For comprehensive coverage on this topic, please refer to textbooks such as Zipkin (2000), Snyder and Shen (2011). However, in today's digital and fiercely competitive world, e-commerce companies are facing new challenges in inventory management owing to the increasing level of customer

* Authors contribute equally.

diversity, the increasing variety of products, and the higher level of service required. For example, on large e-commerce platforms (such as Amazon and JD.com), hundreds of millions of products are simultaneously sold, with various demand patterns that require different replenishment strategies. Hence, it is critical to develop a framework that would be able to identify the optimal/close-to-optimal strategy automatically for different demand, since they can't manage these many products efficiently by current practices.

Motivated by the previously mentioned requisite, we provide a framework that automatically outputs the replenishment decisions for a large number of SKUs. More specifically, we consider the multi-period inventory management problem over a finite horizon, while both demand and vendor leadtime (VLT) are considered to be stochastic. The study of this kind of problem starts from Kaplan (1970) and Ehrhardt (1984), where the merit of (s, S) policies and myopic base stock policies are demonstrated. However, implementing such policies requires an estimation/prediction of certain parameters/random variables then incorporating the estimation/prediction into those policies (see Toktay and Wein (2001), Wang et al. (2012), Zhu and Thonemann (2004) as represnetatives). This type of solution paradigm where first predicting random variables of interest then incorporating the forecast results into optimization stage is the so-called predict-then-optimize (PTO) solution framework (Elmachtoub and Grigas (2017)). Although being widely adopted, it decouples the prediction stage and optimization stage. Consequently, the optimization step does not use the input data in an optimized manner and useful information can be substantially lost in the PTO process.

Instead of implementing a conventional two-step framework, we propose a data-driven, end-to-end (E2E) framework for this problem. The term "end-to-end" means training a model to output the inventory replenishment decision directly from input data without any intermediate steps. The integration of prediction and optimization has been investigated in several existing works about inventory management Ban and Rudin (2018), Oroojlooyjadid et al. (2016). Both Ban and Rudin (2018) and Oroojlooyjadid et al. (2016) focus on the feature-based newsvendor problem, which is essentially a quantile regression problem of demand and a direct recipe is provided by statistical learning theory Koenker et al. (2005). However, the multi-period replenishment problem is substantially different from newsvendor problem in two aspects: first, it adopts a multi-period setting where current decisions affect the future instead of the single-period setting in newsvendor problem; second, there are two types of uncertainties (stochastic demand and stochastic VLT), while newsvendor problem only considers one source of uncertainty (stochastic demand). Therefore, the multi-period inventory problem is an essentially more complicated problem and there is no simple closed-form solution for the optimal replenishment decision.

The lack of proper labels makes a fundamental challenge for developing an end-to-end supervised learning framework in the multi-period setting. To overcome it, we propose a labeling method by solving a dynamic programming problem and label each sample of order with the optimal decision under its realization (Theorem 1). As for the choice of model structure, we design a modular deep learning framework where we have an individual prediction block (a recurrent neural network) for demand and an individual prediction block (a multiple layer perceptron) for VLT respectively. Then these two blocks join together with other features such as review period and initial inventory, to produce the final replenishment decision output. Compared with a fully connected neural network over all features, our design reduces the computational complexity in magnitudes, while providing convenience on explanation and good performance as well. Such a modular-designed framework, together with the labeling process, could form a general recipe for developing End-to-End learning models for other large-scale supply chain management problems. We conduct a series of thorough numerical experiments that consist of both off-line comparisons and a field experiment. In off-line numerical experiments, we compare the performance of the proposed E2E model with existing PTO benchmarks using real-world datasets from JD.com. The results demonstrate that the E2E model could reduce the total inventory management cost compared with multiple PTO methods.

The E2E algorithm has been implemented at JD.com for some SKUs in “tea set” and “pastry essentials & seasoning” categories starting from February 2020. As JD.com gradually expanding the number of SKUs with E2E algorithm implemented, we conduct a systematic field experiment for 30 days from March 30, 2020 to April 30, 2020. The experiment involved 61430 orders placed in 12 distribution centers for 9308 SKUs. We compare the performance of the E2E algorithm with that of the retailer’s current replenishment algorithm. The results show that the E2E algorithm dominates the current algorithm across all performance measures. More specifically, the average holding cost, stockout cost, and total inventory cost for the treatment group are all reduced by more than 25% compared with those for the control group, while two other metrics, the turnover rate and stockout ratio, are reduced by 8.8% and 34.6%, respectively. Hypothesis tests conducted confirm that all observed reductions are statistically significant. To further verify the effect of the E2E algorithm, we adopt a difference in differences approach. The field experiment results demonstrate the immediate applicability of the E2E method at JD.com.

Our work distinguishes itself from the existing literature in the following three aspects:

1. We are the first to propose an end-to-end framework for the multi-period replenishment problem. Based on a labeling method that we proposed, our framework outputs replenishment decisions directly from input features. We believe this could be a general recipe for end-to-end models while the optimization stage is complicated.

2. The proposed end-to-end learning framework automatically decides the replenishment strategy for various demand patterns. Our approach outperforms multiple baseline models, including the current practice of a leading online retailer in both offline simulations with real-life data and

the field experiment. The field experiment verifies the applicability of our algorithm for real-world inventory management in the e-commerce industry.

3. We also innovate the design of deep neural networks structure by designing two modules for demand and VLT uncertainty separately. Our design reduces the computational complexity and the number of weights in magnitudes while achieving good performance. We also perform a comprehensive sensitivity analysis in Appendix B.

The remainder of the paper is organized as follows: In Section 2, we review the related literature. In Section 3, we state the detail of problem setting and the E2E model. In Section 4, we test our E2E model by conducting off-line numerical experiments with real-world data. In Section 5, we demonstrate the design and results of the field experiment. Finally, we conclude in Section 6 and propose some potential future research directions. Moreover, Appendix A gives the detailed proof of Theorem 1. Sensitivity analysis of neural network architectures, datasets, and model covariates are provided in Appendix B.

2. Literature Review

In this section, we first briefly review existing methods for multi-period replenishment problem and classic two-step PTO algorithms. After that, we discuss some existing literature on data-driven approaches, as well as the emerging idea of integrating prediction and optimization for inventory management problems, which is mainly based on the Newsvendor setting. Finally, we introduce the development and applications of end-to-end approaches in other fields, which serves as the incubator for the proposed E2E inventory management framework.

Multi-period inventory management problem has been studied in decades since Kaplan (1970). Using dynamic programming, Kaplan (1970), Ehrhardt (1984) established the optimality conditions for base stock and (s,S) policies under finite and infinite horizons, respectively. Although the optimality of base stock policies has been proven under different settings (Iida and Zipkin (2006), Gallego and Özer (2001), Muharremoglu and Tsitsiklis (2008)), calculating the optimal parameter of such policy remains computationally intractable under general cases Levi et al. (2007). In order to get computationally tractable algorithms, one option is to use approximation algorithms. For example, Levi et al. (2007) provided 2-approximation algorithms using dual-balancing techniques. The other option is to use heuristic algorithms such as myopic policies (see Veinott Jr (1965), Iida and Zipkin (2006), Ignall and Veinott Jr (1969) as representative works).

However, all these aforementioned policies assume certain knowledge of demand and VLT (e.g., demand and VLT follow certain distributional models and the parameters of the model are known). In practice, such information is often unveiled to decision-makers. Therefore, a two-step predict-then-optimize (PTO) framework is widely adopted in industry for inventory management. The

PTO framework first forecasts demand and VLT then incorporates the prediction into certain decision rules such as base stock and (s, S) policies stated above.

In the stage of prediction, there are two different types of forecasting methods for demand and VLT. The output of forecasting can be a point estimator or a distribution of the random variable. The first type of forecasting is widely adopted in industry since in some cases accurate prediction can be achieved by machine learning models (e.g., Friedman et al. (2001)). However, point estimation of random variables can lead to information loss, which affects the subsequent optimization stage. In contrast, if we can make perfect forecast of the random variable distribution, we have all the information in order to solve the following stochastic optimization problem. Recently, there are works that develop distributional/probabilistic forecasting models. When the random variable does not depend on external features, the distribution can be fitted by simply using empirical distribution of historical observations or by kernel density estimation (Sheather and Jones (1991)). When the random variable depends on covariates, distributional/probabilistic forecasting becomes more difficult. A recent work by Bertsimas and Kallus (2020) uses a non-parametric method to approximate the distribution of the random variable conditioned on covariates by weighted empirical distribution. However, this benchmark is not applicable to numerical experiments using real-world dataset that includes time series features, due to computationally difficulty. Böse et al. (2017) forecast multiple quantiles of demand to gain more distributional information. Their approach is similar to the benchmark BM1 in our offline numerical experiments. Ambrogioni et al. (2017) propose another nonparametric method for conditional density estimation using kernel mixture network. In their work, densities are assumed as linear combinations of a family of kernel functions and the weights are determined by a deep neural network. Therefore, it requires a lot of computational effort thus not applicable in our setting. Moreover, we want to highlight the fact that, due to the multi-period setting in our problem, even though we have a practical method for estimating the conditional distribution, solving corresponding stochastic dynamic programming is also computationally difficult (we refer to Levi et al. (2007) for a more detailed review).

Since the traditional two-step approaches that separate the prediction from the optimization often lead to sub-optimality, there has been a trend to perform these two steps simultaneously in the recent literature on data-driven inventory management Ban and Rudin (2018), Oroojlooyjadid et al. (2016), Liyanage and Shanthikumar (2005), Chu et al. (2008), Bertsimas and Kallus (2020). Such integration attempts could be realized through operational statistics. Liyanage and Shanthikumar (2005) demonstrated the existence of improved operational statistics (in contrast to the use of the maximum likelihood estimator) by integrating the prediction and optimization steps on several demand distributions; Chu et al. (2008) further studied how to obtain the optimal operational statistics in a Bayesian framework. Both Liyanage and Shanthikumar (2005) and Chu et al. (2008) only studied the Newsvendor problem.

Ban and Rudin (2018), Oroojlooyjadid et al. (2016) investigated the concept of integration in the feature-based Newsvendor situation, where one has access to past demand observations, as well as to a large number of related features. Ban and Rudin (2018) studied this problem with the Newsvendor loss function and assumed a linear relationship between the features and the Newsvendor quantile (i.e. the solution). They analytically showed that their approach can perform better than the SAA and the separated estimation and optimization method. Oroojlooyjadid et al. (2016) adopted a multiple-layer perceptron model that optimized the order quantity. However, all the aforementioned works have been focused on the Newsvendor problem, in which neither the connection between the periods nor the vendor lead time has been considered, which are both important factors in practice.

Beyond inventory problems, Bertsimas and Kallus (2020), Elmachtoub and Grigas (2017) studied the “integration” philosophy for general optimization problems. Bertsimas and Kallus (2020) combined machine learning and optimization techniques for decision-making purposes when features (referred to as auxiliary quantities in the paper) were available. Their idea was to construct weight functions from data through machine-learning methods and to incorporate these weights to the objective in the optimization procedure. Under the context of linear programming, Elmachtoub and Grigas (2017) proposed a “smart PTO” framework that directly leveraged the optimization problem structure forming the loss function.

Another branch of research that served as an impetus for our work originated from the machine-learning community. In recent years, there has been a dramatic increase in the number of systems built on “E2E learning” Donti et al. (2017). This term refers to a learning framework, the ultimate goal of which is directly predicted from raw inputs rather than from intermediate steps. This concept has been successfully applied to a wide range of tasks, such as finance Bengio (1997), image recognition Wang et al. (2011) and robotics manipulation Levine et al. (2016). These lines of works have provided certain valuable inputs to us in terms of integrating prediction and optimization; however, such an “E2E” approach has not yet been studied with a focus on the general supply-chain management problem.

3. Model

In this section, we first describe the multi-period replenishment problem; In Section 3.1, this problem is presented with a dynamic programming framework. In Section 3.2, we explain our E2E model with emphasis on its two key components, the property of the optimal dynamic programming solution and the deep learning network structure.

3.1. The Multi-Period Replenishment Problem

In this work, we consider the multi-period inventory management problem with stochastic demand and VLT. The details are as follows: for a single item at a single location, we consider a finite horizon of discrete periods $1,\dots,T$, where T is the end of the horizon. Over the T periods, there is a sequence of random demands, denoted by $D_t, \forall t=1,\dots T$. Let I_t denote the inventory level at the beginning of period t. The inventory level can be positive if we have inventory excess on hand or negative if we have an inventory shortage and, hence, backorders. As a result, at the end of each period, we either incur a holding cost of h for each excess unit or a stock-out cost of b for each back-ordered unit.

We consider periodic review policies and assume that review periods are given as a sequence of dates. That is, we assume there are totally M orders from period 1 to T that are placed at $t_m, \forall m=1,\dots,M$. This assumption is aligned with the real-world practice, where a fixed schedule is typically held for order placement (e.g., one can place orders on Tuesdays and Fridays). In this problem, we consider the stochastic VLT. That is, the m^{th} order placed at period t arrives at period $t+L$, where L is a random variable that only takes positive integer values. Hence, the arrival time of orders, denoted by $v_m = t_m + L_m$, $\forall m=1,\dots,M$, are also random variables. Moreover, we assume there are no crossing-over of order arrivals. Although some of the aforementioned assumptions may be unrepresentative (such as back-order and no crossing-overs), in section 4.2 we relax some of them and test our proposed method under a more realistic setting.

Hereafter, D_t and L_m will denote the random variables; d_t and l_m denote the realization of demand at period t and the realization of VLT of the m^{th} order, respectively. At each period, the system first updates the inventory level by checking if any order has arrived; then, demand occurs. Let a_m denote the order quantity for the m^{th} order. At the end of the period, either a holding cost or back-order cost occurs. The inventory level updates follow the equation below,

$$I_{t+1} = I_t - D_t + \sum_{m=1}^{M} a_m \mathbb{1}\{t = t_m + L_m\}. \tag{1}$$

Let the cost that occurred at period t be denoted by S_t, we have

$$S_t = h[I_t - D_t + \sum_{m=1}^{M} a_m \mathbb{1}\{t = t_m + L_m\}]^+ + b[-I_t + D_t - \sum_{m=1}^{M} a_m \mathbb{1}\{t = t_m + L_m\}]^+. \tag{2}$$

where $[\cdot]^+$ denotes $\max\{\cdot, 0\}$.

Our aim is to minimize the expected cost during the finite horizon by choosing the order quantities at given periods, that is

$$\min_{a_1,\dots,a_M} \mathbb{E}\left[\sum_{t=1}^{T} S_t\right]. \tag{3}$$

S_t is defined by (2) and the updates of I_t follow (1). Note that the expectation is taken over the joint distribution of the demand $\{D_t\}_{t=1}^{T}$ and the VLT $\{L_m\}_{m=1}^{M}$.

3.2. End-to-End (E2E) Model

The goal of the replenishment problem is to determine the best order quantity, $a: f(\mathbf{x}) \in R$ at each given review point, after having observed all the features, $\mathbf{x}$. Such features can include historical demand, VLT, item specifications, and temporal information (day, month, season).

To find the mapping function $f(\cdot)$, we first train the model with historical data. For each historical replenishment time point, with observed feature vector $\mathbf{x_i}$, we need to compute a_i^*, which is the corresponding optimal order quantity. This step is referred to as "labeling" for supervised learning algorithms (Section 2.1.4 James et al. (2013)). We will describe the details of the labeling method in Section 3.2.1. When the associated label for each observation is completed, we can establish the mapping with the following training objective:

$$\min_{f:\mathcal{X}\to R} \sum_{i=1}^{N} L(f(\mathbf{x}_i); a_i^*)\,, \tag{4}$$

where N is the total number of training data, L is the loss function that is defined based on the difference between the model prediction $f(\mathbf{x}_i)$ and the optimal order quantity a_i^*. In particular, we consider neural network models for function f and we will describe the details of the neural-network structure in Section 3.2.2.

3.2.1. Labeling the optimal order quantity Unlike the newsvendor problem, for which the optimal solution is the $\frac{b}{b+h}$ quantile of demand distribution, the optimal solution of the multi-period inventory problem is not straightforward to calculate. In this section, we will analyze the properties of the optimal order solution, hence producing labels $a_i^*, i = 1, ..., N$ for the training set.

Given the order place and arrival time, and the demand at every time step, we can compute the optimal quantity for each order using the dynamic programming framework. It should be noted that within periods $1, \ldots, T$, there are M orders placed. Moreover, $t_m \in \{1, \ldots, T\}, m = 1, \ldots, M$ denotes the time period in which the m^{th} order is placed (the quantity can be 0). In a similar manner, let v_m denote the time when the m^{th} order arrives. It should be stressed that we assume no crossover of lead time. With given demand, we can formulate the recursion as

$$V_m(I_{v_m}) = \min_{a_m \geq 0} \sum_{s=v_m}^{v_{m+1}-1} h[I_{v_m} + a_m - d_{[v_m,s]}]^+ + b[d_{[v_m,s]} - I_{v_m} - a_m]^+ + V_{m+1}(I_{v_m} + a_m - d_{[v_m,v_{m+1}-1]}), \tag{5}$$

where $V_m(I_{v_m})$ is the optimal cost over interval $[v_m, v_{m+1} - 1]$, $d_{[i,j]} := \sum_{t=i}^{j} d_t$.

The following theorem describes the closed-form solution of (5); hence, it provides an efficient approach to label the training and testing data set.

THEOREM 1. *The optimal multi-period inventory replenishment problem described by (5) is decomposable, i.e.,* $a_m^{**} := \arg\min_{a_m \geq 0} \left\{ \sum_{s=v_m}^{v_{m+1}-1} h[I_{v_m} + a_m - d_{[v_m,s]}]^+ + b[d_{[v_m,s]} - I_{v_m} - a_m]^+ + \right.$

$V_{m+1}(I_{v_m}+a_m-d_{[v_m,v_{m+1}-1]})\}=\arg\min_{a_m\geq 0}\sum_{s=v_m}^{v_{m+1}-1} h[I_{v_m}+a_m-d_{[v_m,s]}]^{+}+b[d_{[v_m,s]}-I_{v_m}-a_m]^{+}$. *In addition, the closed form of the optimal solution is* $a_m^{**}=\max\{d_{[v_m,s^*]}-I_{v_m},0\}$, *where* $s^*=\lfloor\frac{b(v_{m+1}-v_m)}{h+b}\rfloor+v_m$.

Remark 1. Theorem 1 indicates that labeling using ex-post optimal order quantities is practical, i.e. computationally efficient for large dataset. If, instead of deterministic dynamic programs, stochastic dynamic programs are solved to get labels, the labeling process becomes computationally intractable.

3.2.2. Neural-network structure When the associated labels for training data are obtained, we train a neural-network model f by solving the optimization problem in (4). The general structure of the neural-network model is shown in Figure 1.

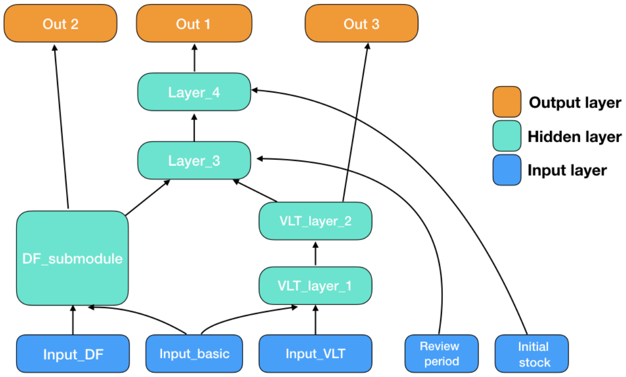


Figure 1 Neural network structure of the E2E Model.

The inputs of the E2E model include five parts, where `Input_DF` and `Input_VLT` represent features related to demand and VLT, respectively; `Input_basic` is the set of general item-level features, such as product categories, warehouse locations, and brand names. The remaining two features, review period and initial stock level, are directly fed into one of the hidden layers because they are not intended to generate any cross terms with other features. The E2E model has three outputs, where the main output `Out1` $\in\mathbb{R}$ represents the final replenishment decision. In addition, there are two accessory outputs `Out2` as the demand forecast and `Out3` as the VLT forecast.

All hidden layers, except for `DF_submodule`, are fully connected layers with rectified linear unit (ReLU) activation function and dropout layers Srivastava et al. (2014) to prevent overfitting. The `DF_submodule` is designed as a multi-quantile RNN (MQRNN), which receives multiple time series (e.g., demand time series, promotion time series) as inputs and produces a daily demand prediction over a set of quantiles as outputs. We use MQRNN because of its demonstrated performance in demand forecasting in the e-commerce industry Wen et al. (2017), Fan et al. (2019).

The training objective function is defined as

$$\min_{\theta} \sum_{i=1}^{N} \left\{ L(\texttt{Out1}_i; a_i^*) + \lambda_1 \hat{L}_1(\texttt{Out2}_i, a_{DF,i}^*) + \lambda_2 \hat{L}_2(\texttt{Out3}_i, a_{VLT,i}^*) \right\}, \tag{6}$$

where θ is the set of neural network parameters to be optimized, N is the total number of training data, and λ_1, λ_2 are two small positive constants penalizing the demand and VLT prediction error. A few reasons lead us to include three terms in (6), rather than only the first term. First, the optimization of the two forecasting outputs (i.e., `Out2` as the demand forecast and `Out3` as the VLT forecast) act as a guide for faster model training. Without them, it becomes more difficult for each submodule of the network to be trained as desired. Moreover, `Out2` and `Out3` can be used for monitoring the model performance. While running the E2E model in real-time, the observation of any anomalies in these two outputs can help decision-makers detect any abnormal output on replenishment amounts and analyze the reason behind it.

The above network structure is designed with the knowledge that the replenishment decision would be made using the information of the demand and the VLT, whose feature sets hardly overlap and are barely related. Thus, compared with a fully connected network over all features, our design reduces the computational complexity and the number of weights in magnitudes, while providing convenience on explanation and good performance as well.

4. Numerical Experiment

In this section, we test the performance of E2E model using real-world data. In Section 4.1, we introduce several common two-step predict-then-optimize (PTO) benchmarks. In Section 4.2, we evaluate the performance of E2E model with different PTO benchmarks, and highlight the benefits of both the one-step decision framework and the deep learning architecture. In all experiments, we use real data from JD.com, one of the largest online retailers in China. It owns hundreds of warehouses to manage inventory and has replenishment agreements with tens of thousands of vendors. All neural networks are implemented in PyTorch Paszke et al. (2017) and trained on a server with an NVIDIA Tesla P40 GPU. We refer to Appendix B for more details of model parameters and a sensitivity analysis.

4.1. Comparison with Two-Step Replenishment Methods

In order to show the performance of our E2E model, we compare the E2E model with several currently in-use two-step PTO methods.

We start with two widely adopted base stock policies. In “Normal” base stock, the daily demand is assumed to be independent and identically distributed (i.i.d.) and follows the Normal distribution. Thus, the base-stock level is computed as

$$BM_{normal} = \mu_D(R + \mu_{VLT}) + \phi^{-1}(\frac{b}{b+h})\sqrt{(R+\mu_{VLT})\sigma_D^2 + \mu_D^2\sigma_{VLT}^2}, \tag{7}$$

where R is the review period. The mean (μ_D, μ_{VLT}) and the standard deviation (σ_D, σ_{VLT}) are estimated using historical data that contains the demand and the VLT of the past 180 days. Similarly, in “Gamma”, daily demand is assumed to be i.i.d. and follows Gamma distribution. Hence, the sum of $(R + \mu_{VLT})$ days of demand, denoted by $\bar{D}$, follows $Gamma((R+\mu_{VLT})k, \theta)$, where θ and k are estimated using the demand data of the past 180 days. The base-stock level is computed as

$$BM_{gamma} = Q_{\bar{D}}^{gamma}(\frac{b}{b+h}), \tag{8}$$

where $Q_{\bar{D}}$ denotes the quantile function of $\bar{D}$.

In addition to the two base-stock policies, we also want to compare the E2E model with PTO benchmarks. As mentioned earlier in Section 2, there are two different types forecasting in PTO method: one type of PTO method estimate a point prediction of the random variable while the other type predicts the distribution of the random variable. If a PTO method succeeds to achieve perfect prediction of the joint conditional distribution of demand and VLT, then theoretically, (3) can be solved to optimality. However, as we reviewed in Section 2, there barely exists applicable method for our problem setting. Moreover, even with reliable forecastings of the joint distribution, we still need to solve a corresponding stochastic dynamic programming problem, which is computationally intractable.

Therefore, we construct the following two PTO benchmarks using a MQRNN for demand forecasting and a MLP for VLT prediction for fair comparison.

1) **BM1.** First, we let $\hat{d}_m = \sum_{t=t_m}^{v_{m+1}-1} d_t$ denote the total demand within two adjacent order-arrival times, namely t_m and v_{m+1}. Note that v_m can be calculated based on VLT prediction. Then the $b/(b+h)$ quantile of $\hat{d}_m$ is predicted, equivalent to solve the following problem:

$$\min_{a_m \geq 0} \mathbb{E}_{\hat{d}_m}[b(\hat{d}_m - a_m - I_{t_m})^+ + h(a_m + I_{t_m} - \hat{d}_m)^+], \tag{9}$$

BM1 then can be considered as a PTO method with point prediction of VLT and distributional forecasting of demand. In the optimization stage, the multi-period problem is approximated by a single-period Newsvendor problem.

2) **BM2.** An alternative PTO method is to first sequentially forecast the future demand within two adjacent order-arrival times, that is $d_{t_m}, d_{t_m+1}, ..., d_{v_{m+1}-1}$, and then calculate the optimal decision, a_m, by minimizing the following accumulated inventory cost

$$\min_{a_m \geq 0} \sum_{t=v_m}^{v_{m+1}-1} h[I_{t_m} + a_m - d_{[t_m,t]}]^+ + b[d_{[t_m,t]} - I_{t_m} - a_m]^+. \tag{10}$$

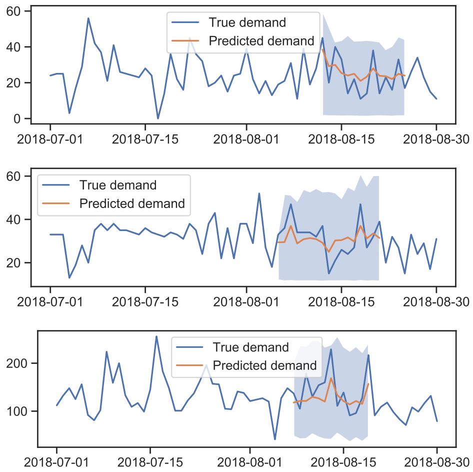
where v_m is estimated as $\hat{v}_m = t_m + \hat{l}_m$, and v_{m+1} is estimated as $\hat{v}_{m+1} = t_{m+1} + \hat{l}_m$ as well. Notice that $\hat{l}_m$ comes from `Out3` as the VLT forecast and d_t is the demand forecast. BM2 can be viewed as a PTO method with point prediction for both VLT and demand, but in optimization stage, the multi-period problem setting is kept.

The minimizers of (9) and (10) are denoted by $a^*_{Bm1,m}$ (BM1) and $a^*_{Bm2,m}$ (BM2), respectively. Two benchmarks are employed because each of them emphasizes on different stages as a two-step model. First, because predicting the total demand within a time interval is more likely to reach the desired accuracy than predicting a sequence of demands for each day, the first benchmark could yield better results in the prediction stage. However, by noticing that the objective in (9) contains a newsvendor loss rather than the one in our multi-period setting, one can expect that the second benchmark would yield results of higher accuracy in the optimization stage.

4.2. Results

We test the performance of E2E method and the four PTO policies using real-world data, a 24,333 SKUs dataset under the Food & Snack Category. The input vector for each replenishment sample contains SKU profile features, daily sales, and historical VLT, which is a 132-dimensional vector. The entire dataset are split into a training and a testing dataset according to the creation date of each sample. We use the first 30-day replenishment-order data as the training and validation set, where 80% of the data are used for training and the remaining 20% are used for validation. The validation set is used to evaluate the performance of the neural network model for different combinations of hyper-parameter values, which helps to choose the optimal hyperparameters and prevent over-fitting. The remaining 30 days of data for all SKUs serves as the testing set. The performance of E2E model and benchmarks are evaluated by total inventory management cost, holding cost and stockout cost, defined via (2). By default, the values of b and h were set to 9 and 1, respectively.

The total inventory management cost, holding cost and stockout cost of different models over the test period are listed in Figure 2 (right). The averaging holding cost and stockout cost were computed as in (2). "OPT" refers to the optimal replenishment decisions obtained by solving the replenishment problem with known demand and VLT, which is the same approach that is used for the labeling of the data, as described in Section 3.2.1. For the two PTO methods "BM1" and



Method	Total cost	Holding cost	Stockout cost
OPT	3022.60	1925.70	1096.91
E2E_RNN	**3766.52** (+24.6%)	**2689.02**	**1077.50**
BM1	4207.09 (+39.2%)	2502.55	1704.54
BM2	4157.03 (+37.5%)	2254.99	1902.04
Normal	4576.05 (+51.4%)	3369.21	1206.84
Gamma	4476.17 (+48.1%)	2821.87	1654.30
E2E_GBM	4017.82 (+32.9%)	2109.93	1907.89

Figure 2 Left: demand forecast for three example SKUs via MQRNN. Blue line is the ground truth, 10% **and** 90% **quantile forecasts are the lower and upper boundary of the forecast band, and** 50% **(median) is the orange line within the band. The shaded area is the forecast within the current order place time and next order arrival time. Right: average inventory cost under different E2E and PTO polices over 30 days per SKU.**

"BM2", we use the same neural network architecture as the E2E model, that is an MQRNN[1] for demand prediction and a 2-layer feedforward neural network for the VLT prediction. In this experiment, the MQRNN model outputs $Q = 6$ quantile prediction, including mean and quantile levels at 10%, 60%, 70%, 80%, 90% and 95%. The results for BM1 and BM2 are reported based on the best quantile with the lowest total cost.

Figure 2 shows that the proposed E2E deep learning model is the best compared to all benchmarks. The advantages of end-to-end deep learning model are two-fold. On one hand, the benefit of using end-to-end framework rather two-step PTO framework can be observed by comparing the cost of E2E_RNN against BM1 and BM2. Since all three algorithms use the same deep learning structure for demand and VLT prediction, while E2E_RNN adopts one-step decision framework and BM1, BM2 use two-step framework. We suspect that the prediction errors of demand and lead time compound in the optimization stage. To avoid the accumulation of error, the end-to-end framework shortens the decision process while targeting the ultimate optimization goal. On the other hand, the benefit of deep learning model versus other statistical learning models can be observed when comparing the E2E_RNN cost and E2E_GBM cost. E2E_GBM denotes the performance of the end-to-end LightGBM Ke et al. (2017) model, which is a decision tree based algorithm that widely used in industry. Since the tree-based models are not able to process time-series features (e.g., historical demand series), we use statistical summary of demand as features including the mean, standard deviation and temporal differences. Both E2E_RNN cost and E2E_GBM algorithms

[1] Predicting the total demand for Benchmark 1 and predicting the daily demand for Benchmark 2.

use the end-to-end decision framework, while the deep learning model has better representation capacity and RNN is more powerful in modeling time series data.

5. Field Experiment

The proposed E2E algorithm has been implemented in JD.com since February, 2020. As of October 2020, there are more than 7,000 SKUs adopts the E2E algorithm for inventory management, showing significant inventory cost saving in production. JD.com is also working on expanding the E2E method to more categories.

In this section, we demonstrate the design and results of a 30 days field experiment conducted in JD.com's logistics system from March 30, 2020 to April 30, 2020.

5.1. Overview of JD.com's Auto-Replenishment System

JD.com maintains a logistics network in China that consists of about 500 Distribution Centers (DC) national-wide. Each DC manages its inventory using JD's inventory replenishment system. JD's current inventory replenishment algorithm can be viewed as a two-step (PTO) decision-making process empowered by machine learning techniques and industry expertise. In the first step, the demand and VLT are predicted using state-of-the-art machine learning methods considering seasonality, geographic effect, SKU and vendor heterogeneity. Fan et al. (2019) describes an effort to the retailer's advanced demand forecast using deep learning techniques. In the second step, the inventory replenishment decision is made based on the predictions from the first step. Service level used in current practice of JD.com is decided by a hyper-parameter called "critical ratio" which is consistent with the hyper-parameters b and h in E2E model. Generally speaking, the critical ratio for different product category doesn't have to be the same.

In JD's replenishment system, the performance of a replenishment algorithm is quantified by five key metrics: stockout rate, turnover rate and the three metrics we used in Section 4.2 including the total inventory management cost, holding cost and stockout cost. The stockout rate is defined as the percentage of days that stockout occurs during the experimental period, i.e. it measures the frequency of stockouts. The inventory turnover rate is calculated by dividing the average inventory level of each day by the average demand.

5.2. Experiment Design

To test whether the proposed E2E algorithm leads to better replenishment decisions, we conducted a field experiment during a 30 days period, from March 30, 2020 to April 30, 2020. The experiment involved 61430 orders placed in 12 DCs for 9308 SKUs. The SKUs are from 18 third-level categories, which belong to two second-level categories, ”tea set” and ”pastry essentials & seasoning”. The details of the categories that are involved in the field experiment are listed in Table 1.

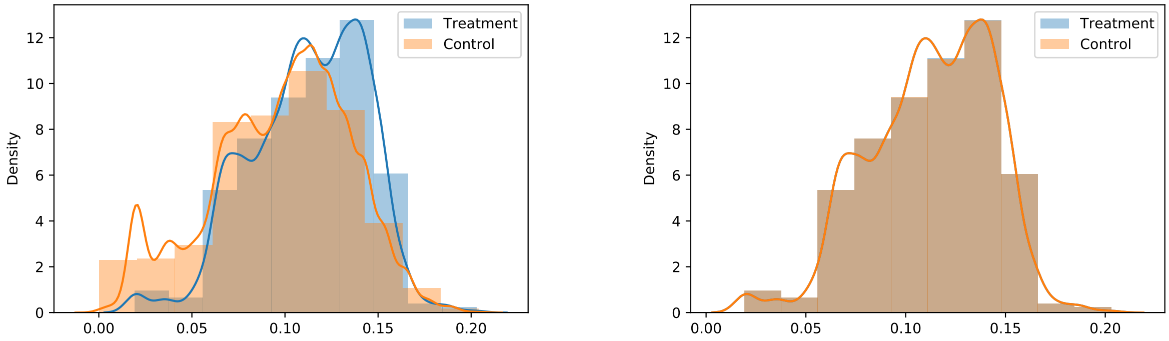
Table 1 Details of Categories Involved in the Field Experiments

Second level categories	Tea set	Pastry essentials & Seasoning
Third level categories	Tea set combination Tea cup Tea kettle Tea tray Tea can Tea bowl Tea accessories Tea-ware decoration Coffee set Tea travel set	Baking supplies Flour Mixed-grains Rice Oil Seasonings Dry foods Convenience foods

An E2E model is trained for the second-level category, tea set, and for each third-level category in the second-level category, pastry essentials & seasoning. In this experiment, we use $h = 12$ and $b = 88$, which is consistent with the critical ratio in these categories. The field experiment contains all replenishment orders placed after March 30, 2020 for each (SKU, DC) pair in the aforementioned categories and twelve DCs. A treatment group involves 1052 SKUs and 6097 (SKU, DC) pairs in total, are selected. The replenishment decisions of these (SKU, DC) pairs are made according to the proposed E2E algorithm starting from March 30, 2020. For the remaining (SKU, DC) pairs, replenishment decisions are made following JD's current algorithm. The treatment group of (SKU, DC) pairs was chosen by JD.com's managers based on business criteria. We select a control group that also include 6097 (SKU, DC) pairs using propensity score matching in order to reduce the possibility of selection bias in this process (with details stated in Section 5.2.1). We collect daily inventory levels, daily sales, replenishment order placement and arrival dates for all (SKU, DC) pairs in the field experiment to calculate the performance metrics including the average holding cost, average stockout cost, average total cost, average turnover rate and average stockout rate.

Moreover, JD's logistics system adopts less restrictive setting compared to Section 3.2. Instead of fully back-order, demand that can not be fulfilled by current inventory will be considered as back-ordered only if it can be fulfilled by open purchased orders. (Open purchase orders refer to those orders that have been placed but haven't arrived, in other words, replenishment that is on the way.) If it can not be fulfilled by open purchased orders, it will be considered as lost sales. In addition, there may be cross-over of orders, i.e. orders may not arrive in the same sequence as they are placed. Therefore, the field experiment can test the performance of the proposed E2E algorithm in real-world setting.

5.2.1. Propensity score matching As explained earlier, the treatment group contains 6097 (SKU, DC) pairs, selected based on voluntary response, and all remaining (SKU, DC) pairs from the categories listed in Table 1 are the candidate control group. In order to address the issue of potential selection bias, we choose (SKU, DC) pairs in the control group by using propensity score matching (see Rosenbaum and Rubin (1983) and Rubin and Waterman (2006) for references). For propensity score matching, we use demand and VLT as cofounder variables. Figure 3 visualizes the propensity score of control and treatment sets before and after matching. After matching, we have a control group with same size as the treatment group.



(a) Propensity score before matching

(b) Propensity score after matching

Figure 3 Propensity Score Matching. The overlap of the histogram and density plot in Fig (3b) **demonstrates that the propensity score has been perfectly matched.**

5.3. Results

In this subsection, we first compare the performance of the two algorithms during the test period using a two-sample t-test. Then to further adjust the potential differences in the treatment and control groups, we train linear regression models for each outcome metric and check the significance of the coefficients. Moreover, we evaluate the performance of both treatment and control groups before the test period and apply the difference in differences technique to further confirm the effect of applying the E2E method.

Table 2 Comparison of Performance of Two Algorithms: t-test Results

	Holding Cost	Stockout Cost	Total Cost	Turnover Rate	Stockout Ratio
Treatment Group	598.20	496.34	1094.54	18.59	0.17
Control Group	809.54	1027.93	1837.47	20.39	0.26
Difference	**-211.34(-26.1%)**	**-531.59 (-51.7%)**	**-742.93(-40.4%)**	**1.8(-8.8%)**	**0.09(-34.6%)**
t-test p-value	<0.001****	<0.001****	<0.001****	0.0102**	<0.001****

Note: $*p<0.1$; $**p<0.05$; $***p<0.01$; $****p<0.001$.

5.3.1. T-test results Table 2 demonstrates the results of t-test for the comparison of five performance metrics: holding cost, stockout cost, total inventory cost, turnover rate, and stockout ratio. The E2E algorithm significantly reduces all five metrics with four out of five t-test p-values < 0.001 and one p-value < 0.05. In particular, the E2E algorithm can reduce the average holding cost by 26.1% and average stockout cost by 51.7%, compared to JD.com's current replenishment method. Not surprisingly, the total cost is reduced by 40.4%. The average turnover rate has also been reduced by 8.8% and the stockout ratio reduced by 34.6%.

5.3.2. Linear regression Furthermore, one may have the concern that the treatment group and the control group having slightly different average demand and average VLT might lead to unreliable t-tests. To address this concern, we further consider the following linear regression model for the outcomes of each metric,

$$\text{Outcome} = \theta_0 + \theta_1 \text{Is-E2E} + \theta_2 \text{Ave-Demand} + \theta_3 \text{Ave-VLT}, \tag{11}$$

where Is-E2E is a binary independent variable that represents if a SKU orders using E2E algorithm, Ave-Demand is an independent variable that represents the average demand of a SKU and similarly Ave-VLT represents the average VLT of a SKU.

The linear regression model plays a similar role as the t-tests. It aims to provide a better comparison of the treatment and control groups' average outcome by considering covariates that may affect the outcome of the metrics. Table 3 demonstrates the coefficients and p-values of Is-E2E.

Table 3 Comparison of Performance of Two Algorithms: Linear Regression

	Holding Cost	Stockout Cost	Total Cost	Turnover Rate	Stockout Ratio
Coefficient	-211.33	-531.53	-742.86	-1.16	-0.09
t-test p-value	< 0.001****	< 0.001****	< 0.001****	0.007***	< 0.001****

Note: $*p < 0.1$; $**p < 0.05$; $***p < 0.01$; $****p < 0.001$.

Table 3 indicates that using the E2E algorithm leads to significant reductions on all five metrics - holding cost, stockout cost, total cost, turnover rate and stockout ratio.

5.3.3. Difference-in-differences estimation To further study the effect of E2E algorithm on all five metrics, we consider a difference in differences (DiD) approach. To be more specific, we evaluate the performance of the treatment and control groups from February 29, 2020 to March 29, 2020. We denote this period as the pre-experiment period. Then compare the performance with that of the post-experiment test period, March 30, 2020 to April 30, 2020. During the pre-experiment period, both treatment and control group adopt JD's current replenish algorithm and during the post-experiment period, the treatment group adopt the E2E algorithm while the control group still follows JD's algorithm.

Table 4 demonstrates the DiD comparison of the effect of E2E algorithm implementation. In Table 4, the terms "Pre-Exp" and "Post-Exp" denote the pre-experiment and post experiment

Table 4 Difference-in-Differences Estimation of E2E algorithm

	Treatment Group			Control Group				
	Pre-Exp	Post-Exp	Change	Pre-Exp	Post-Exp	Change	**DiD**	t-test p-value
Holding Cost	688.13	598.20	-89.93	786.68	809.54	22.86	**-112.79**	< 0.001****
Stockout Cost	880.50	496.34	-384.16	887.64	1027.93	140.29	**-524.45**	< 0.001****
Total Cost	1568.63	1094.54	-474.09	1674.32	1837.47	163.15	**-637.24**	< 0.001****
Turnover Rate	16.41	18.59	2.18	16.40	20.39	3.99	**-1.81**	0.041**
Stockout Rate	0.24	0.17	-0.07	0.22	0.26	0.04	**-0.11**	0.044**

Note: $*p < 0.1$; $**p < 0.05$; $***p < 0.01$; $****p < 0.001$.

periods, respectively. The results indicate that the implementation of our proposed E2E algorithm improves all five metrics. The readers may notice that, in the control group, there is a slight increment of the post-experiment metrics compared to the pre-experiment metrics. Our conjecture is that, as the economics in China begins to recover in March 2020, both demand and supply have a larger scale in April compared to March.

6. Conclusions

In this work, we propose an E2E framework with deep learning models for multi-period replenishment problems, without prior assumptions on future demands and on the VLT. The model is trained to capture the behavior of optimal solutions from a perfect knowledge of the future. Collaborated with an industrial partner, our proposed E2E model has been implemented in production and we conduct a series of numerical experiments including a field experiment to demonstrate the advantage of the proposed E2E model over conventional two-step PTO approaches and current practices in industry. Our model, as well as the "E2E" philosophy, can be practically useful for the industry because it shortens the decision process and provides a more automatic inventory management solution. With the possibility of scaling and generalization, the proposed E2E model enables higher inventory-management accuracy with a lower operational cost and fewer labor efforts.

Moreover, we suggest several opportunities for future research in the E2E concept for supply-chain management. For instance, one potential direction would be trying to generalize the E2E model to more general inventory management settings, such as multi-echelon cases and inventory allocation problems. Another appealing direction would be constructing an E2E solution for jointly deciding order quantity and ordering time.

7. Acknowledgements

This research was supported by national key RD program of China 2018YFB1700600, NSFC grants 71991462 and 91746210. The authors would like to express their appreciation to the support from JD.com, especially to the VP of JD.com and Chief Manager of Smart Supply Chain Management (SSCM) unit, Mr. Chen Lin, and the Chief Technology Manager of SSCM, Mr. Hao Hu. Moreover, the authors would like to thank colleagues in JD.com from SSCM unit, including Lei Chen, Xin Wang, Shuyu Han, Carl Morris, Jingtao Chen, Jing Li, Jianshen Zhang, Changfei Zhan, Jing Lu, Hongjie He; colleagues from fast moving consumer goods team, including Yanhua Wang, Xueli Zhao, Youcheng Wang, Litian Wu; and Jingjing Wang from fashion and lifestyle team.

References

Ambrogioni L, Güçlü U, van Gerven MA, Maris E (2017) The kernel mixture network: A nonparametric method for conditional density estimation of continuous random variables. *arXiv preprint arXiv:1705.07111* .

Ban GY, Rudin C (2018) The big data newsvendor: Practical insights from machine learning. *Operations Research* 67(1):90–108.

Bengio Y (1997) Using a financial training criterion rather than a prediction criterion. *International Journal of Neural Systems* 8(04):433–443.

Bertsimas D, Kallus N (2020) From predictive to prescriptive analytics. *Management Science* 66(3):1025–1044.

Böse JH, Flunkert V, Gasthaus J, Januschowski T, Lange D, Salinas D, Schelter S, Seeger M, Wang Y (2017) Probabilistic demand forecasting at scale. *Proceedings of the VLDB Endowment* 10(12):1694–1705.

Choromanska A, Henaff M, Mathieu M, Arous GB, LeCun Y (2015) The loss surfaces of multilayer networks. *Artificial Intelligence and Statistics*, 192–204.

Chu LY, Shanthikumar JG, Shen ZJM (2008) Solving operational statistics via a bayesian analysis. *Operations Research Letters* 36(1):110–116.

Donti P, Amos B, Kolter JZ (2017) Task-based end-to-end model learning in stochastic optimization. *Advances in Neural Information Processing Systems*, 5484–5494.

Ehrhardt R (1984) (s, s) policies for a dynamic inventory model with stochastic lead times. *Operations Research* 32(1):121–132.

Elmachtoub AN, Grigas P (2017) Smart "predict, then optimize". *arXiv preprint arXiv:1710.08005* .

Fan C, Zhang Y, Pan Y, Li X, Zhang C, Yuan R, Wu D, Wang W, Pei J, Huang H (2019) Multi-horizon time series forecasting with temporal attention learning. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2527–2535 (ACM).

Friedman J, Hastie T, Tibshirani R (2001) *The elements of statistical learning*, volume 1 (Springer series in statistics New York, NY, USA).

Gallego G, Özer Ö (2001) Integrating replenishment decisions with advance demand information. *Management science* 47(10):1344–1360.

Ignall E, Veinott Jr AF (1969) Optimality of myopic inventory policies for several substitute products. *Management Science* 15(5):284–304.

Iida T, Zipkin PH (2006) Approximate solutions of a dynamic forecast-inventory model. *Manufacturing & Service Operations Management* 8(4):407–425.

James G, Witten D, Hastie T, Tibshirani R (2013) *An introduction to statistical learning*, volume 112 (Springer).

Kaplan RS (1970) A dynamic inventory model with stochastic lead times. *Management Science* 16(7):491–507.

Kawaguchi K (2016) Deep learning without poor local minima. Lee DD, Sugiyama M, Luxburg UV, Guyon I, Garnett R, eds., *Advances in Neural Information Processing Systems 29*, 586–594 (Curran Associates, Inc.), URL `http://papers.nips.cc/paper/6112-deep-learning-without-poor-local-minima.pdf`.

Ke G, Meng Q, Finley T, Wang T, Chen W, Ma W, Ye Q, Liu TY (2017) Lightgbm: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 3146–3154.

Koenker R, Chesher A, Jackson M (2005) *Quantile Regression*. Econometric Society Monographs (Cambridge University Press), ISBN 9780521608275, URL `https://books.google.com/books?id=hdkt7V4NXsgC`.

LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *nature* 521(7553):436.

Levi R, Pál M, Roundy RO, Shmoys DB (2007) Approximation algorithms for stochastic inventory control models. *Mathematics of Operations Research* 32(2):284–302.

Levine S, Finn C, Darrell T, Abbeel P (2016) End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research* 17(1):1334–1373.

Liyanage LH, Shanthikumar JG (2005) A practical inventory control policy using operational statistics. *Operations Research Letters* 33(4):341–348.

Muharremoglu A, Tsitsiklis JN (2008) A single-unit decomposition approach to multiechelon inventory systems. *Operations Research* 56(5):1089–1103.

Oroojlooyjadid A, Snyder L, Takáč M (2016) Applying deep learning to the newsvendor problem. *arXiv preprint arXiv:1607.02177* .

Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A (2017) Automatic differentiation in pytorch .

Rosenbaum PR, Rubin DB (1983) The central role of the propensity score in observational studies for causal effects. *Biometrika* 70(1):41–55.

Rubin DB, Waterman RP (2006) Estimating the causal effects of marketing interventions using propensity score methodology. *Statistical Science* 206–222.

Sheather SJ, Jones MC (1991) A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society: Series B (Methodological)* 53(3):683–690.

Snyder L, Shen Z (2011) *Fundamentals of Supply Chain Theory* (Wiley), ISBN 9780470521304, URL `https://books.google.com/books?id=U7GTrLyVnPMC`.

Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15(1):1929–1958.

Toktay LB, Wein LM (2001) Analysis of a forecasting-production-inventory system with stationary demand. *Management Science* 47(9):1268–1281.

Veinott Jr AF (1965) Optimal policy for a multi-product, dynamic, nonstationary inventory problem. *Management Science* 12(3):206–222.

Wang K, Babenko B, Belongie S (2011) End-to-end scene text recognition. *Computer Vision (ICCV), 2011 IEEE International Conference on*, 1457–1464 (IEEE).

Wang T, Atasu A, Kurtuluş M (2012) A multiordering newsvendor model with dynamic forecast evolution. *Manufacturing & Service Operations Management* 14(3):472–484.

Wen R, Torkkola K, Narayanaswamy B (2017) A multi-horizon quantile recurrent forecaster. *arXiv preprint arXiv:1711.11053* .

Zhu K, Thonemann UW (2004) An adaptive forecasting algorithm and inventory policy for products with short life cycles. *Naval Research Logistics (NRL)* 51(5):633–653.

Zipkin P (2000) *Foundations of Inventory Management* (McGraw-Hill Companies,Incorporated), ISBN 9780256113792, URL `https://books.google.com/books?id=rjzbkQEACAAJ`.

Appendix A: Proof of Theorem 1

In order to prove Theorem 1, we need the following lemma.

LEMMA 1. *Consider the total inventory cost within time period* $[t_1,t_2]$ *as* $f(a)=\sum_{s=t_1}^{t_2} h[I_{t_1}+a-d_{[t_1,s]}]^+ + b[d_{[t_1,s]}-I_{t_1}-a]^+$, *where* h *is the unit holding cost and* b *is the unit back-order cost. The optimal order quantity which minimizes the total cost is derived by* $\arg\min_{a\in\mathbb{R}} f(a)=a^*=d_{[t_1,s^*]}-I_{t_1}$ *where* $s^*=\lfloor \frac{b(t_2+1-t_1)}{h+b}+t_1\rfloor$.

Proof of Lemma 1 : First we show that a^* takes the form of $d_{[t_1,s]}-I_{t_1}$, for some $s\in[t_1,t_2]$. It is evident that $f(a)$ is convex and piece-wise linear, hence the optimal solution should be one of the extreme point, which is $d_{[t_1,s]}-I_{t_1}$ for some s. Now we need to choose s^* such that $s^*=\arg\min_{s\in\{t_1,\dots,t_2\}}\{f(d_{[t_1,s]}-I_{t_1})\}$. Note that for $s\in\{t_1,\dots,t_2\}$, choosing to satisfy one unit of demand that occurred at period s generates a holding cost $h(s-t_1)$, while choosing not to satisfy one unit of demand generates back-order cost $b(t_2+1-s)$, we will choose to satisfy the demand unit that occurs at period s such that $h(s-t_1)\le b(t_2+1-s)=b(t_2+1-t_1)-b(s-t_1)$. The above analysis is valid for all units that occurs in period s, hence we will choose to satisfy either all demand units that occurs at period s or non of them. Hence we have $s\le s^*$ for all s such that $h(s-t_1)\le b(t_2+1-t_1)-b(s-t_1)$, which lead to the final solution $s^*=\lfloor \frac{b(t_2+1-t_1)}{h+b}+t_1\rfloor$. □

Lemma 1 leads to our final result of Theorem 1, on the optimal order quantity under the realization of each sample data.

Proof of Theorem 1 : First, we relax the constraint $a_m\ge 0$ and start from the last decision a_M. By Lemma 1, $a_M^*=\min_{s\in\{v_M,\dots,T\}}\{d_{[v_M,s]}-I_{v_M}\}$ and $s^*=\arg\min_{s'\in\{v_M,\dots,T\}}\sum_{s=v_M}^{T} h[d_{v_M,s'}-d_{[v_M,s]}]^+ + b[d_{[v_M,s]}-d_{v_M,s'}]^+$. It should be noted that s^* only depends on the sequence $\{d_t\}_{t=v_M}^{T}$. Hence, $f_M^*:=f(a_M^*)=\sum_{s=v_M}^{T} h[d_{v_M,s^*}-d_{[v_M,s]}]^+ + b[d_{[v_M,s]}-d_{v_M,s^*}]^+$ does not depend on I_{v_M}; hence, the decision of a_M^* is independent to the decision of a_{M-1}^*. By recursion, we can decompose the decisions $\{a_m^*\}_{m=1}^{M}$ with corresponding period $[v_m,v_{m+1})$.

Now we consider the constraint $a_m\ge 0$. It is not difficult to verify that with constraint $a_m\ge 0$, we can still compute a_m^* use $a_m^*=\arg\min_{a_m\ge 0}\sum_{s=v_m}^{v_{m+1}-1} h[I_{v_m}+a_m-d_{[v_m,s]}]^+ + b[d_{[v_m,s]}-I_{v_m}-a_m]^+$ in the sequence of $a_1,a_2,\dots,a_M$. We let a_m^{**} denote the optimal solution with constraint $a_m\ge 0$. Note that under this constraint, f_{m+1}^* depends on $I_{v_{m+1}}$. In fact, it is a non-decreasing function of $I_{v_{m+1}}$, hence, a non-decreasing function of a_m. Suppose for m, following the above method, we have $a_m^*<0$, then $a_m^{**}=0$ is not only the minimizer of $f_m(a_m)$ but also a minimizer of f_{m+1}^*. □

Appendix B: Sensitivity analysis

In this part, we conduct various sensitivity analysis to demonstrate the robustness and generalization ability of proposed E2E model under different hyper-parameter choices, data size and model covariates.

B.1 Sensitivity Analysis for Network Hyper-parameters

We first provide details of the neural network structure and hyper-parameters used in the E2E model. In the training stage, we sweep through different combinations of hyper-parameters within the considered range and use the validation set to choose the best hyper-parameter values, which is shown below as the "Default Value" column.

Hyperparameter	Default Value	Range
`DF_submodule: MQRNN`	Hidden state size 50	{30, 40, 50, 60}
`VLT_submodule` (`VLT_layer_1`, `VLT_layer_2`)	Layer size {50, 20}	{100, 20}, {50, 20}, {30, 20}
Integration module (`Layer_3`, `Layer_4`)	Layer size {100, 100}	{100}, {100, 100}, {100, 100, 100}, {100, 100, 100, 100}
Learning rate	0.001	{0.0001, 0.001, 0.01}
Learning rate decay	$1e-4$	{1e-5, 1e-4, 1e-3}
Momentum	0.85	{0.8, 0.85, 0.9}
Mini-batch size	64	{64, 128, 256}
Weight initialization	Gaussian $\mu = 0, \sigma = 0.01$	{Gaussian, Uniform}
Activation	Rectified linear unit (ReLU)	{ReLU, tanh}
Dropout rate	0.2	{0.1, 0.2, 0.3, 0.4}

Table 5 Network hyper-parameters

In addition, in order to investigate the sensitivity of the E2E model performance stated in Section 4 with respect to the network structure and hyper-parameters, we provide three sets of sensitivity tests with respect to the following hyper-parameters:

- Number of hidden layers
- Number of neurons/weights
- Learning rate

B.1.1 Number of Hidden Layers: To check the sensitivity of E2E model with respect to the number of hidden layers, we adjust the number of hidden layers in the integration module. By default, there are two hidden layers `Layer_3` and `Layer_4` both of size 100. We tried three other variants: 1) keep only one hidden layer, i.e., `Layer_3` of size 100; 2) have three hidden layers, i.e., `Layer_3`, `Layer_4`, `Layer_5`, each of size 100; 4) have four hidden layers, i.e., `Layer_3`, `Layer_4`, `Layer_5`, `Layer_6`, each of size 100. We trained four different E2E models with the four different network structures, and test their performance using the same dataset and experiment setup as in Section 4.1.

	OPT	E2E (1 layer)	E2E (2 layers)	E2E (3 layers)	E2E (4 layers)
Total cost	3022.60	**4857.20 (+60.1%)**	**3766.52(+24.6%)**	**3822.69 (+26.5%)**	**3904.74(+29.2%)**
Holding cost	1925.70	3919.17	2689.02	2453.32	2231.01
Stockout cost	1096.91	938.03	1077.50	1369.38	1673.73

Table 6 Sensitivity of E2E model w.r.t. to the number of hidden layers

Table 6 shows the total inventory management cost, holding cost and stockout cost of the four E2E models with different number of hidden layers. When the hidden layer number is 1, the end-to-end model performance is inferior to other benchmark models (in Figure 2) due to the lack of representation capacity. In all other cases, the end-to-end models outperform the benchmarks. As the number of hidden layers increases, the cost of end-to-end model first decreases then increases. By having more layers, the network representation power increases and can better fit the relationship between observation and the optimal order decision. However, an over-complicated network may cause over-fitting in the training set and shows less generalization capacity in test set.

B.1.2 Number of Neurons/weights: To investigate the model sensitivity w.r.t. the number of neurons, we sweep through $[30, 40, 50, 60]$ as the hidden state size of the demand prediction modile (MQRNN), and compare the performance of the different networks.

	OPT	E2E (30)	E2E (40)	E2E (50)	E2E(60)
Total cost	3022.60	**3846.28(+27.3%)**	**3783.32(+25.2%)**	**3766.52(+24.6%)**	**3860.99(+27.7%)**
Holding cost	1925.70	2886.45	2710.71	2689.02	2185.18
Stockout cost	1096.91	959.83	1072.61	1077.50	1675.81

Table 7 Sensitivity of E2E model w.r.t. to the number of neurons

Table 7 reports the total cost, holding cost and stockout cost of the four E2E models with different number of neurons. As expected, as the number of hidden neurons increases, the E2E model cost first decreases then increases. Similar as the effect of adding more layers, by adding more neurons, the representation capacity of the neural network enhances. However, too many neurons may lead to over-fitting in the training set and the trained network shows worse generalization performance.

B.1.3 Learning rate: Learning rate is one of the most important hyper-parameter in neural network training LeCun et al. (2015). Table 8 shows how different learning rate value affects the performance of E2E model on the test set. If we further increase the learning rate to 0.1 or larger, network training process becomes unstable and the training loss oscillates which leads to much worse performance. When learning rate is 0.0001, the learnt model is slightly better than our default setting ($lr = 0.001$) but the training time increases significantly.

	OPT	E2E (lr =0.0001)	E2E (lr =0.001)	E2E (lr =0.01)
Total cost	3022.60	**3744.28 (+23.9%)**	**3766.52(+24.6%)**	**4231.01(+39.9%)**
Holding cost	1925.70	2424.71	2689.02	2350.98
Stockout cost	1096.91	1319.58	1077.50	1880.03

Table 8 Sensitivity w.r.t. to learning rate

B.2 Sensitivity Analysis for Data Size

In order to investigate the sensitivity w.r.t. training data size, we use $[20\%, 40\%, 60\%, 80\%, 100\%]$ of the training data to train the end-to-end model. The cost and computational time of the E2E model with different amounts of training data are provided in Table 9 and Figure 4.

Percentage	Number of training data	OPT cost	E2E cost	E2E Training time (s)
20%	3893 SKUs	3022.60	5608.01(+85.5%)	281.61
40%	7786 SKUs	3022.60	4765.52(+57.7%)	624.64
60%	11680 SKUs	3022.60	4226.84(+39.8%)	1022.57
80%	15573 SKUs	3022.60	3783.54(+25.2%)	1459.42
100%	19466 SKUs	3022.60	3766.52(+24.6%)	1830.36

Table 9 E2E Model Sensitivity w.r.t. to Training Data Size

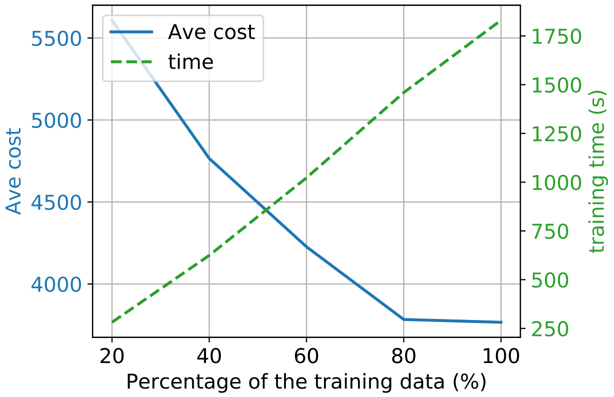


Figure 4 E2E Model Sensitivity w.r.t. to Training Data Size

B3. Sensitivity Analysis for b and h

All the previous simulations are based on the assumption that the ratio between unit stock-out cost and unit holding cost is 9, which may not reflect the case in real world. Therefore, sensitivity analysis with respect to different values of b/h is conducted below to validate our conclusion in previous parts. It should be noted that different b/h ratios not only lead to different measurements costs, but also change replenishment decisions for all models.

Figure 5 demonstrates total cost, holding cost as well as stockout cost of E2E model, OPT decision and two base stock algorithms under different b/h ratios. According to these results, E2E

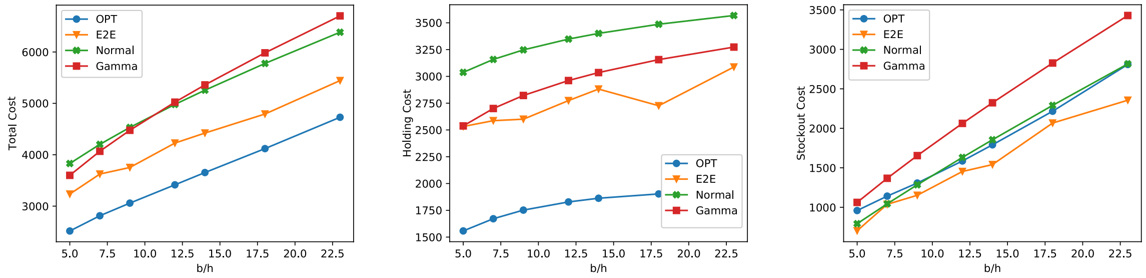


Figure 5 **Performance of each model with different ratios on b/h.**

model is stable and closest to the optimal solutions in most cases, with respect to different choices of b/h.

B4. Sensitivity for Different Neural Network Local Minima

Since the loss function of neural networks are highly non-convex and contain many local minima, when training neural networks using stochastic gradient descent (SGD) methods (the classical way of training neural networks), we are very likely to get stuck in one of the local minima. However, several recent research experimenting with larger networks and SGD suggest that, while deep neural networks do have many local minima, they consistently give very similar performance Choromanska et al. (2015), Kawaguchi (2016). Consistent with the above results from machine learning literature, we find the performance of our proposed E2E deep learning model has similar property in numerical experiments. For instance, two different E2E models with the default network structure and hyper-parameter are trained with different weight initialization (random seeds). Figure 6 visualizes the final weights of `VLT_layer_1` of the two trained networks. And Table 10 provides the performances of two networks on test data set. It can be observed that although the two network weights are quite different, they have similar performances on the test set in terms of total cost, holding cost as well as stockout cost.

	OPT	E2E (Network 1)	E2E (Network 2)
Total cost	3022.60	3766.52(+24.6%)	3800.54(+25.7%)
Holding cost	1925.70	2689.02(+39.6%)	2766.67(+43.7%)
Stockout cost	1096.91	1077.50(-1.8%)	1033.87(-5.7%)

Table 10 **Comparison of two E2E model performance with different initialization seeds.**

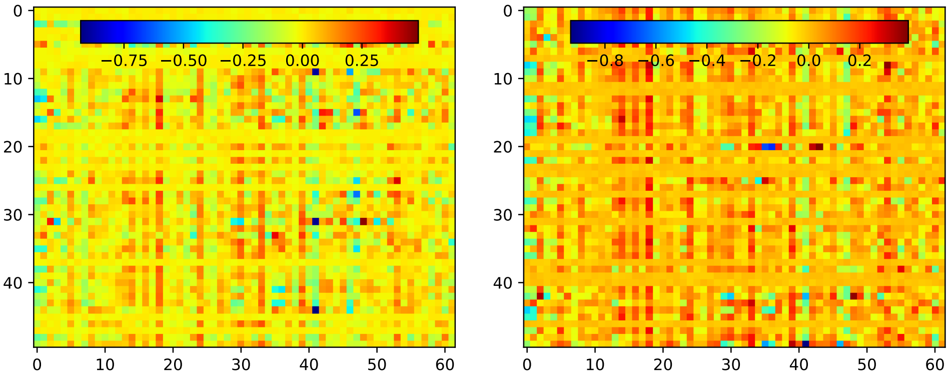


Figure 6 **Network sensitivity w.r.t. local minima. The two plots show the weights of `VLT_layer1` of two neural networks trained with different initialization.**