**PRODUCTION AND OPERATIONS MANAGEMENT**
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 1059-1478 | EISSN 1937-5956 | 00 | 0000 | 0001

**POMS**
DOI 10.1287/xxxx.0000.0000
© 0000 POMS

# Tackling Decision Dependency in Contextual Stochastic Optimization

Wenxuan Liu#
Columbia University, wl3003@columbia.edu

Xiangting Liu#
Tsinghua University, liu-xt22@mails.tsinghua.edu.cn

Maoqi Liu#
Shandong University, liumq@sdu.edu.cn

Zhi-Hai Zhang*
Tsinghua University, zhzhang@tsinghua.edu.cn

Hanzhang Qin
National University of Singapore, hzqin@nus.edu.sg

**Abstract:** In this paper, we study the contextual stochastic optimization (CSO) problem, where decisions are made under uncertainty, and the distributions of the random parameters can be partially inferred from covariates observed prior to decision-making. In many practical settings, the distributions of random parameters also depend on the decision itself, which is the decision-dependent effect. For example, in retail, the demand distribution is influenced by the price set for the product. Most of the existing literature addresses this issue by imposing structural assumptions on the relationship between decisions and the distribution of random parameters, such as modeling the mean demand as a linear function of price. However, such assumptions can lead to model misspecification when the true relationship deviates from the assumed form. To over come this limitation, We used the wSAA method to model the effect of decision and context to the distribution. However, this method becomes computationally challenging since both the weights and the loss function depend on the decision, particularly when the weights are learned through complex machine learning techniques such as random forests or k-nearest neighbors. To address this computational challenge, we apply the wSAA idea directly to the loss function gradient, leading to what we call the contextual gradient. We then integrate the contextual gradient into a gradient descent framework to solve CSO problems with decision-dependent effects. And we show that the proposed contextual gradient descent (CGD) method achieves a bounded error relative to the global optimum under strong convexity and converges to a stationary point of the expected gradient in general settings. Notably, the bound reveals a key insight that the degree of strong convexity in the loss function can help compensate for the uncertainty introduced by decision-dependent effects. Extensive numerical experiments in real-world datasets demonstrate that our CGD algorithm consistently outperforms existing methods designed for contextual optimization with decision-dependent uncertainty.

**Key words**: contextual optimization, prescriptive analytics, decision-dependency

## 1 Introduction

In many real-world management processes, decision-makers must make decisions under uncertainty. These problems are often formulated as stochastic optimization models of the form:

$$\min_{\boldsymbol{x}\in\mathcal{X}} g(\boldsymbol{x}) = \mathbb{E}_{\boldsymbol{y}\sim f(\boldsymbol{y})}[l(\boldsymbol{x},\boldsymbol{y})], \tag{1}$$

where $\boldsymbol{x}$ is the decision variable within the feasible region $\mathcal{X}\subseteq\mathbb{R}^d$; $\boldsymbol{y}$ is a random parameter with a probability density function $f(\boldsymbol{y})$; $l(\boldsymbol{x},\boldsymbol{y})$ is the loss function, which evaluates the performance of decision $\boldsymbol{x}$ under a realization of $\boldsymbol{y}$; and $g(\boldsymbol{x})$ is the expectation of $l(\boldsymbol{x},\boldsymbol{y})$ with respect to $f(\boldsymbol{y})$.

In practice, since $f(\boldsymbol{y})$ is typically unknown to the decision-maker, solving the problem involves two interrelated tasks: prediction and optimization. The prediction task involves estimating the distribution or statistics of the uncertain parameters, while the optimization task selects decisions that minimize the expected loss function based on these estimates.

When the distribution of $\boldsymbol{y}$ is influenced by covariates $\boldsymbol{z}$, which are observed before decision-making, it becomes beneficial to condition on $\boldsymbol{z}$ to improve decision quality. This leads to the framework of *contextual stochastic optimization (CSO)*. The CSO paradigm has recently received growing attention in the operations research community. A comprehensive review of related work can be found in Sadana et al. (2025). Under the CSO setting, the stochastic optimization problem is reformulated as:

$$\min_{\boldsymbol{x}\in\mathcal{X}} g(\boldsymbol{x},\boldsymbol{z}) = \mathbb{E}_{\boldsymbol{y}\sim f(\boldsymbol{y}|\boldsymbol{z})}[l(\boldsymbol{x},\boldsymbol{y})]. \tag{2}$$

CSO problems arise in many practical applications. A notable example is the contextual newsvendor problem (Ban and Rudin 2018), where a decision-maker must determine the optimal order quantity $x$ under uncertain demand $y$. The goal is to minimize the expected negative revenue:

$$l(x,y) = -p(y\wedge x) + cx - s(x-y)^+,$$

where $p$, $c$, and $s$ represent the unit price, unit cost, and salvage value, respectively. Here, $\wedge$ denotes the component-wise minimum, and $(\cdot)^+ = \max\{\cdot,0\}$. While the true distribution of $y$ is unknown, contextual features $z$—such as weather conditions—can be used to improve demand estimation and support more informed decisions.

In the contextual newsvendor problem, the stochastic parameter (i.e., demand) is assumed to be independent of the decision variable (i.e., order quantity). However, in many other applications, an additional challenge arises: the stochastic parameters themselves may depend on the decisions. A typical example is the pricing problem, where a firm must determine the optimal price for a product before observing actual demand. In this case, demand is directly influenced by the price—higher prices generally lead to lower demand, and vice versa.

When the distribution of uncertain parameters is affected by the decision, the conditional distribution $f(\boldsymbol{y}\mid\boldsymbol{z},\boldsymbol{x})$ evolves continuously throughout the optimization process as the decision variable $\boldsymbol{x}$ is iteratively updated. As data under different decision is scarce, estimating the conditional distribution of $\boldsymbol{y}$ on $\boldsymbol{x}$ is challenging. This issue is typically addressed by imposing structural assumptions on the relationship between the decision and the distribution of the random parameter. For example, Chu et al. (2024) tackled the price-setting newsvendor problem by assuming that the expected demand is a linear or log-linear function of the price.

However, structural assumptions can lead to model misspecification when the oracle decision-dependent effect deviates from the assumed form. One promising non-parametric approach to incorporating decision dependence in CSO is the weighted SAA (wSAA) method proposed by Bertsimas and Kallus (2019). They developed an enhanced SAA-based model to estimate the expectation of the loss function conditioned on both $\boldsymbol{x}$ and $\boldsymbol{z}$ in Problem (2). In this framework, each sample in the classical SAA is assigned a weight that is a function of $\boldsymbol{x}$ and $\boldsymbol{z}$, and the weighted sum of the loss function over all samples is optimized. Although this method achieves asymptotic optimality (see Theorem 10 in Bertsimas and Kallus (2019)), the resulting optimization problem is difficult to solve. Because the decision variables appear in the weights determined by complex machine learning methods (e.g., k-nearest neighbors, decision trees, or random forests), the estimated objective function becomes discrete or non-convex with respect to the decision variables—even when $l(\boldsymbol{x},\boldsymbol{y})$ is continuous and convex. To preserve tractability, the authors proposed a discretization strategy and developed a tailored solution approach for the tree-based weight case. However, as we show in our numerical experiments, the method incurs significant computational burden when applied to the general CSO problem (2) under decision-dependent effects.

This research aims to develop a more efficient approach for solving decision-dependent contextual optimization problems under the weighted SAA framework. Instead of applying the wSAA method directly to the loss function, we apply it to the gradient of the loss function, leading to what we call the *contextual gradient*. Unlike the gradient of the original wSAA objective, the contextual gradient does not require computing the derivative of the sample weights, which are typically obtained from complex machine learning models.

Under mild regularity conditions and specific methods for determining the weights, we show that the contextual gradient is a reliable measure of the optimality gap in decision-dependent CSO problems and can effectively serve as the search direction in a gradient descent framework. Leveraging this property, we propose a *contextual gradient descent (CGD)* algorithm and analyze its convergence under both convex and non-convex settings. Specifically, under strong convexity, the CGD algorithm guarantees a bounded optimality gap with respect to the global optimum, where the bound depends on both the strength of convexity and the degree of decision dependence. In general convex and non-convex cases, the algorithm is shown to converge to a stationary point of the original prescriptive optimization problem.

Extensive numerical experiments on both synthetic and real-world datasets are implemented to demonstrate that our method consistently outperforms existing approaches in terms of both solution quality and computational efficiency.

## 1.1 Contributions

Our contributions are threefold:

**First, we propose an efficient and effective approach for solving the CSO problem with decision-dependent effects.** We introduce the concept of the *contextual gradient* and prove that, when the weights are determined by kernel or local linear methods, the optimal solution must exhibit a sufficiently small expected gradient of the loss function. Based on this property, we develop the *contextual gradient descent (CGD)* algorithm by embedding the contextual gradient into the gradient descent framework. Under mild assumptions, we establish convergence guarantees: the algorithm converges to the global optimum under strong convexity and to a stationary point in general convex and non-convex settings. As we demonstrate in Section 6.3.1 through numerical experiments, the CGD algorithm achieves comparable solution quality using less than 34% of the computational time required by existing methods.

**Second, our analysis reveals how strong convexity compensates for decision-dependent effects.** Specifically, under strong convexity, the optimality gap is shown to be proportional to the ratio between the strength of convexity and the degree of decision dependency. Since the contextual gradient is calculated by ignoring some terms in the derivatives of the wSAA objective, this result implies that strong convexity helps mitigate the loss of information caused by this omission. In other words, even when part of decision-dependent structure is excluded from the optimization process, the algorithm can still deliver desirable performance if the loss function is sufficiently big strongly convex coefficient.

**Third, we demonstrate the practical value of the proposed CGD algorithm through extensive numerical experiments.** We evaluate our method on both synthetic and real-world datasets. Compared to the approach of Bertsimas and Kallus (2019), our algorithm achieves lower optimality gaps in significantly less time. Moreover, when compared to a parametric estimate-then-optimize (ETO) approach, CGD performs competitively when the data is generated according to the assumed parametric model and substantially outperforms it when the data generation process deviates from the assumed structure. In a case study based on a real world dataset, we show that our method can benefit in practice.

## 1.2 Examples of Decision-dependent CSO Problem

As we will show, our proposed CGD algorithm is designed for the CSO problem with decision-dependent effect and continuous decision variables. The situation is widely studied in practice. We present three examples as follows.

**Price-setting Newsvendor.** While the application of end-to-end learning models to classical newsvendor problems has been widely studied (Ban and Rudin 2018, Lin et al. 2022), the *price-setting* newsvendor variant—which incorporates pricing decisions that influence demand—has received comparatively less attention. Only a few approaches, such as quantile regression (Harsha et al. 2021), have considered this setting.

In this case, the decision-maker must jointly determine the optimal selling price and inventory level. Recognizing that demand is a stochastic function of the price itself, it is a typical problem with decision-dependent effect, making standard ETO frameworks inadequate, as they assume the distribution of demand is independent of pricing decisions. In contrast, our proposed framework explicitly accounts for the decision-dependent structure by leveraging historical pricing and demand data to jointly optimize price and order quantity.

**Location Problem on a Continuous Map.** Location optimization on a continuous map is a classical topic in operations research (Liu et al. 2024). In some business contexts, facility locations must be repeatedly determined based on contextual covariates. A representative example is mobile retailing such as food trucks or pop-up shops, which operate as mobile stores across a city. In this setting, demand is influenced not only by contextual factors—such as weather, day of the week, or local events—but also directly by the location decision, since proximity and accessibility affect customer behavior. This gives rise to a decision-dependent CSO problem, where the decision involves selecting store locations and the covariates capture contextual signals. Our framework can effectively handle this scenario using historical location, demand, and covariate data, even when the decision space is continuous.

**Project Portfolio Investment Allocation.** Consider a firm allocating a fixed investment budget across a portfolio of projects. The expected return for each project can be estimated using covariates such as project attributes, market conditions, and macroeconomic indicators. However, the realized return is often influenced by the level of investment itself. For example, overfunding similar or competing projects may result in internal competition and diminishing returns. This setting naturally forms a decision-dependent CSO problem, where investment amounts are the decision variables and project features serve as covariates. Our proposed approach enables effective allocation by jointly modeling the uncertainty in returns and their dependency on investment levels.

## Organizations

The remainder of the paper is organized as follows. Section 2 reviews the relevant literature. In Section 3, we introduce the preliminaries including problem formulation, model assumptions, and the wSAA framework, along with the challenges posed by decision-dependent effects. Section 4 presents the concept of the contextual gradient and the proposed CGD algorithm. The convergence properties of CGD under various conditions are analyzed in Section 5. In Section 6, we conduct extensive numerical experiments to validate the performance of the CGD algorithm and compare it with existing methods using synthetic data. A case study based on real-world data is provided in Section 7. Section 8 concludes the paper. All technical proofs are deferred to the appendix for clarity.

### Notations

For simplicity of notation, we use $x \wedge y$ to denote $\min\{x, y\}$, and $(x)^+$ to denote $\max\{x, 0\}$. We let $\nabla$ be the gradient denotation, and $\partial$ denote the subgradient set. Let $\boldsymbol{x} = (x_1, \dots, x_d)^T$. Similarly, all the vectors are presented in bold. The subscript denotes the corresponding coordinate of a vector, and **1** denotes the all-one vector. We use $\|\cdot\|$ to denote $\ell_2$ norm for vectors and matrix. A function $f$ is $L-$Lipschitz continuous on $\boldsymbol{x} \in \mathcal{X}$ if $\|f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2)\| \leq L\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|$ for any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{X}$. A function $f$ is $\gamma-$strongly convex if $f(\boldsymbol{x}_1) - f(\boldsymbol{x}_2) - \nabla f(\boldsymbol{x}_2)^T(\boldsymbol{x}_1 - \boldsymbol{x}_2) \geq \frac{\gamma}{2}\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|^2$. We use $[N] := \{1, \dots, N\}$ to denote the subscript set.

## 2 Literature Review

Contextual stochastic optimization (CSO) has become a focal topic in the operations research community in recent years. In this section, we review the literature on contextual optimization and stochastic models with decision-dependent uncertainty, and compare them to our proposed approach.

A widely adopted framework for solving CSO problems is the *estimate-then-optimize* (ETO) paradigm, also called predict-then-optimize (PTO). ETO methods first estimate the conditional distribution of the uncertain parameters given contextual covariates and then optimize the decision based on this estimate. Our work is closely related to this stream of literature. The most relevant is wSAA framework proposed by Bertsimas and Kallus (2019), which estimates the conditional distribution using a weighted empirical distribution. The weights reflect the similarity between historical and current contexts and are learned through machine learning methods such as nearest neighbors and decision trees. Bertsimas and McCord (2018) extended this framework by introducing a regularization term into the objective function. Similarly, Srivastava et al. (2021) proposed a regularized approach based on Nadaraya-Watson kernel regression to improve out-of-sample performance. Lin et al. (2022) applied the wSAA approach to a risk-averse newsvendor model.

Most of these works assume that the stochastic parameters are independent of the decision variables. While in theory the decision-dependent effect can be incorporated by including decision variables as inputs to the machine learning model, this creates computational challenges. In particular, when the decision variables are continuous, Bertsimas and Kallus (2019) only provided a solution method when tree-based models are used for weight generation; otherwise, discretizing the decision space becomes necessary, which can severely increase computational complexity.

Beyond the wSAA framework, residual-based approaches form another major branch of the ETO approach in CSO. These methods assume that uncertainty follows a regression-type structure—where the mean of the uncertain parameters depends on the covariates, and the residuals are independent noise. Machine learning methods such as linear regression and decision trees are often used to learn this relationship (Ban et al. 2019, Kannan et al. 2025), with the residuals then incorporated into the optimization.

Another influential line of work is the smart predict-then-optimize (SPO) framework proposed by Elmachtoub and Grigas (2022), which uses decision loss to guide the estimation of uncertain parameters. While decision dependence could in principle be included by incorporating decision variables into the estimation model, tractability remains an unresolved challenge in these cases.

In contrast to ETO-based methods, end-to-end learning approaches solve CSO problems by directly learning a decision policy from data, bypassing an explicit prediction stage. For instance, Ban and Rudin (2018) proposed a linear decision rule for the newsvendor model and directly optimized its parameters using training data, showing performance improvements over standard SAA. Other works, including Zhang and Gao (2017), Cristian et al. (2022), and Oroojlooyjadid et al. (2020), used neural networks to parameterize the decision policy, while Kallus and Mao (2022) employed random forests. However, these methods still assume that uncertainty is independent of the decision variables. In contrast, our approach explicitly estimates the conditional expectation of the loss function with respect to both contextual features and decision variables, enabling us to capture decision-dependent effects in a principled and tractable way.

Recently, Feng and Shanthikumar (2023) proposed a general framework—operational data analysis (ODA)—that unifies ETO and end-to-end methods. The authors view data-driven decision-making under uncertainty as constructing a statistic or function from data. ODA introduces a data integration model that projects raw data onto the decision space, potentially leveraging structural properties inherent to the decision problem. Due to its generality, our proposed CGD algorithm can be framed as a special case of ODA, where minimizing the contextual gradient serves as the validation model, and the data integration step is unconstrained.

Our work contributes to the ODA literature by jointly addressing two core challenges—decision dependence and contextual heterogeneity—in a nonparametric and distribution-free manner. While Chu et al. (2024) applied the ODA framework to price-setting newsvendor problems and incorporated decision dependence, their method imposed strong structural assumptions on the price-demand relationship and did not account for contextual variables. Meanwhile, Feng et al. (2025) explored contextual features but neglected decision-dependent effects. Compared to both, our work addresses a more general class of CSO problems without imposing restrictive parametric assumptions, offering a more practical and versatile solution framework.

In summary, most existing CSO frameworks assume independence between decisions and uncertain parameters. Extending them to incorporate decision-dependent effects is difficult, particularly because estimating conditional distributions that depend on decisions remains underexplored. Existing efforts either focus on specific applications such as pricing (Bertsimas and Kallus 2019, Harsha et al. 2021), or do not provide general-purpose optimization strategies. Our framework fills this gap by introducing a tractable, computable approach—the CGD algorithm—for solving general CSO problems with decision-dependent effects.

While decision-dependent uncertainty has not been widely explored in CSO, it has received attention in the broader stochastic programming (SP) literature. Dupacová (2006) highlighted the computational challenges of such models and noted that tractability often depends on problem-specific structure. More recent efforts have sought to generalize the solution methodology. For example, Mendler-Dünner et al. (2020) proposed an iterative gradient descent method for decision-dependent SPs, and Liu et al. (2021a) introduced a coupled learning-enabled optimization (CLEO) algorithm, which estimates the conditional distribution using local linear regression within a carefully designed trust region.

The decision-dependent property has also been studied in robust optimization (RO) and distributionally robust optimization (DRO). Luo and Mehrotra (2020) proposed a robust optimization framework with decision-dependent ambiguity sets under finite, known support. Noyan et al. (2021) constructed ambiguity sets centered at parametric decision-dependent distributions and defined neighborhoods using Wasserstein distances.

Notably, these methods differ from ours in two important ways. First, they do not incorporate contextual information. Second, they assume online access to an oracle capable of sampling from the true distribution under any decision. In contrast, our method is designed for an offline setting, where a historical dataset is available and no new samples can be collected in response to alternative decisions.

To summarize, most existing decision-dependent models rely on parametric assumptions to estimate the conditional distribution, or impose a predefined structure on the relationship between decisions and uncertain parameters. By contrast, our approach is nonparametric, distribution-free, and leverages contextual features, making it more flexible and applicable to a wider range of decision-dependent CSO problems.

# 3 Preliminary

In this section, we provided a formal introduction of the preliminaries, including the problem setting, assumptions and the wSAA approach.

## 3.1 Problem Setting and Assumptions

In this paper, we study the CSO problem with decision-dependent uncertainty. Specifically, we focus on the following formulation:

$$\min_{\boldsymbol{x}\in\mathcal{X}} \quad g(\boldsymbol{x},\boldsymbol{z}) \triangleq \mathbb{E}_{\boldsymbol{y}\sim f(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})}\left[l(\boldsymbol{x},\boldsymbol{y})\right], \tag{3}$$

where $f(\boldsymbol{y} \mid \boldsymbol{x},\boldsymbol{z})$ denotes the conditional probability density function of the uncertain parameter $\boldsymbol{y}$, given decision variable $\boldsymbol{x}$ and contextual covariate $\boldsymbol{z}$.

To facilitate the theoretical analysis, we impose the following assumptions on the structure of the conditional distribution:

ASSUMPTION 1 **(Bounded and Uniform Support)**. *The support of the random parameter* $\boldsymbol{y}$ *is bounded and remains consistent across all decisions* $\boldsymbol{x} \in \mathcal{X}$*, where* $\mathcal{X}$ *is bounded and convex.*

Assumption1 is commonly satisfied in practice, as one can define a unified support by taking the union of all feasible domains of $\boldsymbol{y}$ over the decision space $\mathcal{X}$, and assign zero probability to regions outside the domain of $f(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{z})$. We denote this common support of $y$ as $\Omega \subseteq \mathbb{R}^q$, and its Lebesgue measure (volume) as $S_\Omega$.

ASSUMPTION 2 **(ε-sensitivity)**. *The conditional density function* $f(\boldsymbol{y} \mid \cdot, \boldsymbol{z})$ *is* ε*-sensitive with respect to the decision variable. That is, for all* $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{X}$*,*

$$W_1\big(f(\boldsymbol{y} \mid \boldsymbol{x}_1, \boldsymbol{z}), f(\boldsymbol{y} \mid \boldsymbol{x}_2, \boldsymbol{z})\big) \leq \varepsilon \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|, \tag{4}$$

*where* $W_1$ *denotes the Earth Mover's (Wasserstein-1) distance (Rubner et al. 2000).*

Assumption 2 limits the extent to which the conditional distribution $f(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{z})$ can change as the decision $\boldsymbol{x}$ varies. We refer to ε as the *decision-dependence coefficient*, which quantifies the sensitivity of the distribution to changes in the decision variable. A larger ε indicates stronger decision-dependent effects. This condition holds for a broad class of models. For example, as discussed in Perdomo et al. (2020), Gaussian distributions with decision-dependent mean and variance satisfy this assumption. Additionally, if the decision-dependent effect arises through the mean of a regression-type model—i.e., if the decision only influences the mean and the mean is Lipschitz continuous in $\boldsymbol{x}$—then Assumption 2 is also satisfied. When analyzing the gap between the estimated solution and the optimal solution, Assumption 2 allows us to convert the difference between expectations under different distributions into the distance between decision variables.

ASSUMPTION 3 **(Smooth Density Function)**. *The probability density function of* $\boldsymbol{y}$ *has a Lipschitz continuous gradient with respect to the decision variable* $\boldsymbol{x}$*. That is, for all* $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{X}$*, and* $\boldsymbol{y} \in \Omega$*,*

$$\|\nabla_{\boldsymbol{x}} f(\boldsymbol{y} \mid \boldsymbol{x}_1, \boldsymbol{z}) - \nabla_{\boldsymbol{x}} f(\boldsymbol{y} \mid \boldsymbol{x}_2, \boldsymbol{z})\| \leq L_g \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|.$$

This assumption implies that the rate of change of the density function with respect to the decision variable is bounded. It holds for many commonly used distributions. For instance, similar to Assumption 2, this condition is satisfied by Gaussian distributions whose mean and variance are Lipschitz continuous functions of the decision variable.

The last assumption puts some requirements on the loss function.

ASSUMPTION 4 **(Smooth Loss Function with Bounded Gradient)**. *The loss function $l(\boldsymbol{x},\boldsymbol{y})$ is smooth and Lipschitz continuous with a Lipschitz continuous gradient. Furthermore, its gradient in x is bounded. That is, the following five conditions hold*

$$(a)|l(\boldsymbol{x}_1,\boldsymbol{y})-l(\boldsymbol{x}_2,\boldsymbol{y})|\le L_1\|\boldsymbol{x}_1-\boldsymbol{x}_2\|,\quad \forall \boldsymbol{x}_1,\boldsymbol{x}_2\in\mathcal{X},\boldsymbol{y}\in\Omega,$$

$$(b)|l(\boldsymbol{x},\boldsymbol{y}_1)-l(\boldsymbol{x},\boldsymbol{y}_2)|\le L_2\|\boldsymbol{y}_1-\boldsymbol{y}_2\|,\quad \forall \boldsymbol{x}\in\mathcal{X},\boldsymbol{y}_1,\boldsymbol{y}_2\in\Omega,$$

$$(c)\|\nabla_{\boldsymbol{x}}l(\boldsymbol{x}_1,\boldsymbol{y})-\nabla_{\boldsymbol{x}}l(\boldsymbol{x}_2,\boldsymbol{y})\|\le L_1^c\|\boldsymbol{x}_1-\boldsymbol{x}_2\|,\quad \forall \boldsymbol{x}_1,\boldsymbol{x}_2\in\mathcal{X},\boldsymbol{y}\in\Omega,$$

$$(d)\|\nabla_{\boldsymbol{x}}l(\boldsymbol{x},\boldsymbol{y}_1)-\nabla_{\boldsymbol{x}}l(\boldsymbol{x},\boldsymbol{y}_2)\|\le L_2^c\|\boldsymbol{y}_1-\boldsymbol{y}_2\|,\quad \forall \boldsymbol{x}\in\mathcal{X},\boldsymbol{y}_1,\boldsymbol{y}_2\in\Omega,$$

$$(e)\|\nabla_{\boldsymbol{x}}l(\boldsymbol{x},\boldsymbol{y})\|\le L_3^c,\quad \forall \boldsymbol{x}\in\mathcal{X},\boldsymbol{y}\in\Omega.$$

We assume the Lipschitz property of the loss function so that we can guarantee the estimation error of the contextual gradient will cause a bounded error in our algorithm. When the value ranges of variables are bounded, many commonly used loss functions, such as linear, quadratic, hinge, and logistic loss, naturally satisfy this assumption.

It is worth noting that although we introduce several assumptions to facilitate theoretical analysis, not all of them are required simultaneously. Each assumption is invoked only where necessary for establishing specific convergence results.

### 3.2 Weighted Sample Average Approximation Estimation

In practice, the specific form of the conditional distribution $f(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{z})$ is rarely known. Instead, we typically have access to a historical dataset $\{(\boldsymbol{z}^i,\boldsymbol{x}^i,\boldsymbol{y}^i)\}_{i=1}^N$, where $N$ is the number of observations.

The wSAA approach is a flexible framework to estimate the expected loss function in (3) proposed by Bertsimas and Kallus (2019), which estimated the conditional distribution by the weighted empirical distribution. Specifically, the approach is formulated as

$$\min_{\boldsymbol{x}}\ \hat{g}(\boldsymbol{x},\boldsymbol{z})\overset{\Delta}{=}\sum_{i=1}^{N}w^i(\boldsymbol{x},\boldsymbol{z})\,l(\boldsymbol{x},\boldsymbol{y}^i), \tag{5}$$

where $w^i(\boldsymbol{x},\boldsymbol{z})$ is a weight function used to combine samples with varying contextual and decision inputs for better estimating the conditional distribution under the current scenario.

The core idea is to assign higher weights to historical samples that are more similar to the current context-decision pair $(\boldsymbol{x},\boldsymbol{z})$. These weights are typically computed using machine learning (ML) models prior to solving (5). Below, we briefly present representative examples of weighting schemes. A more detailed discussion of implementation can be found in Bertsimas and Kallus (2019), Lin et al. (2022).

EXAMPLE 1 (K-NEAREST NEIGHBOR WEIGHT (KNN)). The weight function based on the $k$-nearest neighbors (kNN) method is defined as:

$$w^{\mathrm{kNN},i}(\boldsymbol{x},\boldsymbol{z})=\frac{1}{k}\mathbb{I}\{(\boldsymbol{x}^i,\boldsymbol{z}^i)\text{ is among the }k\text{ nearest neighbors of }(\boldsymbol{x},\boldsymbol{z})\},\quad \forall i\in[N], \tag{6}$$

where $\mathbb{I}\{\cdot\}$ is the indicator function.

EXAMPLE 2 (KERNEL REGRESSION WEIGHT (KR)). Weights can also be computed via kernel functions based on the distance between $(\boldsymbol{x},\boldsymbol{z})$ and $(\boldsymbol{x}^i,\boldsymbol{z}^i)$:

$$w^{\mathrm{KR},i}(\boldsymbol{x},\boldsymbol{z}) = \frac{K_h((\boldsymbol{x},\boldsymbol{z})-(\boldsymbol{x}^i,\boldsymbol{z}^i))}{\sum_{j=1}^N K_h((\boldsymbol{x},\boldsymbol{z})-(\boldsymbol{x}^j,\boldsymbol{z}^j))}, \tag{7}$$

where $K_h : \mathbb{R}^{\dim(\boldsymbol{x})+\dim(\boldsymbol{z})} \to \mathbb{R}$ is a kernel function with bandwidth $h$. Common choices include uniform, triangular, and Gaussian kernels. Unless otherwise noted, we adopt the Gaussian kernel:

$$K(\boldsymbol{z}) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{\|\boldsymbol{z}\|_2^2}{2}\right). \tag{8}$$

EXAMPLE 3 (CLASSIFICATION AND REGRESSION TREE WEIGHT (CART)). CART-based weights are defined using leaf membership in decision trees:

$$w^{\mathrm{CART},i}(\boldsymbol{x},\boldsymbol{z}) = \frac{\mathbb{I}\{R(\boldsymbol{x},\boldsymbol{z}) = R(\boldsymbol{x}^i,\boldsymbol{z}^i)\}}{|\{j : R(\boldsymbol{x}^j,\boldsymbol{z}^j) = R(\boldsymbol{x},\boldsymbol{z})\}|}, \tag{9}$$

where $R : \mathcal{X} \times \mathcal{Z} \to \{1,\dots,r\}$ maps each sample to one of the $r$ terminal leaves in the regression tree.

EXAMPLE 4 (RANDOM FOREST WEIGHT (RF)). Random forest weights average across multiple CART estimators:

$$w^{\mathrm{RF},i}(\boldsymbol{x},\boldsymbol{z}) = \frac{1}{N_E}\sum_{e=1}^{N_E} w^{\mathrm{CART},i,e}(\boldsymbol{x},\boldsymbol{z}), \tag{10}$$

where $N_E$ is the number of estimators, and $w^{\mathrm{CART},i,e}$ is the weight assigned by the $e$-th tree.

Despite its flexibility, solving the weighted-SAA problem (5) poses significant computational challenges due to the dependence of weights on the decision variable $\boldsymbol{x}$. Specifically:

- **Non-convexity:** Since $\boldsymbol{x}$ appears in both the weights and the loss function, cross-product terms emerge, making the overall estimation of objective function potentially non-convex—even if $l(\boldsymbol{x},\boldsymbol{y})$ is convex in $\boldsymbol{x}$.
- **Non-smoothness:** When ML models like kNN or CART are used to determine the weights, the resulting estimation function can be non-smooth or even discontinuous, rendering standard gradient-based solvers inapplicable.

We illustrate the above challenge using the price-setting newsvendor problem.

EXAMPLE 5 (PRICE-SETTING NEWSVENDOR PROBLEM). Consider the repeated sale of a perishable product with uncertain, price-dependent demand. The wholesaler sets a price $p$ and order quantity $q$, denoted jointly as $\boldsymbol{x} = (p,q)^T$, to maximize expected revenue, or equivalently, minimize its negative revenue.

Let $c$ be the unit cost and $s$ the salvage value. The loss function for a realized demand $y$ is:

$$l(\boldsymbol{x},y) = -p(y \wedge q) + cq - s(q-y)^+.$$

While $l(\boldsymbol{x},y)$ is non-differentiable at $y = q$, we can compute its subgradient:

$$\partial_{p,q} l(\boldsymbol{x},y) = \left\{ \begin{bmatrix} -(y \wedge q) \\ -(p-c)+(p-s)e \end{bmatrix} : e \in [\mathbb{I}\{q > y\}, \mathbb{I}\{q \geq y\}] \right\}. \tag{11}$$

Once weights $w^i(\boldsymbol{x},\boldsymbol{z})$ are determined by an ML model, computing the full gradient of (5) requires evaluating:

$$\partial_{p,q}\hat{g}(\boldsymbol{x},\boldsymbol{z})=\sum_{i=1}^{N}\left[w^i(\boldsymbol{x},\boldsymbol{z})\,\partial_{p,q}l(\boldsymbol{x},y^i)+\partial_{p,q}w^i(\boldsymbol{x},\boldsymbol{z})\,l(\boldsymbol{x},y^i)\right].$$

The main difficulty lies in computing $\partial_{p,q}w^i(\boldsymbol{x},\boldsymbol{z})$: for kernel methods, this is analytically complex due to exponential terms; for kNN or CART, the gradients may not even exist.

These issues make traditional solving approaches like gradient descent inapplicable in many cases. Although Bertsimas and Kallus (2019) proposed a discretization-based solution, we show in Section 6.3.1 that it is very computationally expensive to implement.

# 4 Contextual Gradient and Its Application in the Gradient Descent Algorithm

In this section, we introduce the concept of the *contextual gradient* in Section 4.1, and integrate it into the gradient descent framework for efficiently solving decision-dependent CSO problems in Section 4.2.

## 4.1 Contextual Gradient

The contextual gradient can be seen as a weighted SAA of the gradient of the loss function, which is formally presented in Definition 1.

DEFINITION 1 (CONTEXTUAL GRADIENT). Given a contextual variable $\boldsymbol{z}'$ and a decision $\boldsymbol{x}'$, the contextual gradient at $\boldsymbol{x}'$ based on the dataset $\{(\boldsymbol{z}^i,\boldsymbol{x}^i,\boldsymbol{y}^i)\}_{i=1}^{N}$ is defined as:

$$\hat{G}_N(\boldsymbol{x}';\boldsymbol{z}')=\sum_{i=1}^{N}w^{(i)}(\boldsymbol{x}',\boldsymbol{z}')\,\nabla_{\boldsymbol{x}}l(\boldsymbol{x}',\boldsymbol{y}^i), \tag{12}$$

where $w^{(i)}(\cdot,\cdot)$ is the weight function obtained from historical data.

The weights used in the contextual gradient (12) are derived using the same procedure as in the wSAA (5). Compared to directly computing the gradient of estimation of the objective (5), the contextual gradient avoids taking derivatives of the weights. This circumvents several computational challenges, such as the complexity of ML-derived weights and the non-convexity arising from weight-loss function interactions mentioned in the last section.

Lemma 1 establishes that with kernel methods determining the weights, the contextual gradient (12) is an asymptotically unbiased estimator of the conditional expectation of the gradient of the loss function.

LEMMA 1 **(Convergence of Contextual Gradient (Theorem EC.9 in Bertsimas and Kallus (2019)))**. *Suppose the joint distribution of $(\boldsymbol{x},\boldsymbol{z})$ is absolutely continuous with a density function bounded away from 0 to ∞ over the support set and twice continuously differentiable. Then for weight functions based on the kernel method (7), the following holds almost surely, for almost every $\boldsymbol{x}'$ and $\boldsymbol{z}$:*

$$\lim_{N\to\infty}\left\|\hat{G}_N(\boldsymbol{x}';\boldsymbol{z})-\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}',\boldsymbol{z})}[\nabla_{\boldsymbol{x}}l(\boldsymbol{x}',\boldsymbol{y})]\right\|=0. \tag{13}$$

This asymptotic unbiasedness follows from theoretical guarantees of kernel-based methods. While similar convergence results for other weight functions in Section 3.2 are not formally established, our numerical evaluations (see Section 6.2) suggest that these alternatives often perform comparably or even better in practice.

It is important to note that $\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}',\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}',\boldsymbol{y})]$ is not the gradient of the full objective (3), which is defined as:

$$G(\boldsymbol{x}';\boldsymbol{z}) = \nabla_{\boldsymbol{x}} \mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}',\boldsymbol{z})}[l(\boldsymbol{x},\boldsymbol{y})]. \tag{14}$$

Therefore, it is not immediately clear how the contextual gradient relates to the optimality conditions of the original problem. The following result bridges this gap.

THEOREM 1 **(Necessary Condition for Optimality)**. *Let $\boldsymbol{x}^*$ denote an optimal solution to the decision-dependent problem* (3). *If the loss function is $L_1$-Lipschitz continuous and Assumptions 2 and 4 are satisfied, it holds that:*

$$\left\|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^*,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^*,\boldsymbol{y})]\right\| \le L_1\varepsilon.$$

Theorem 1 depicts the relationship between expected gradient and optimality. $\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x},\boldsymbol{y})]$ should be sufficiently small at the optimal solution. And with Lemma 1, the contextual gradient can also be used as a metric to assess optimality. Specifically, these results imply that the contextual gradient, as an unbiased estimator of $\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x},\boldsymbol{y})]$, should also be sufficiently small at the optimal solution.

With this property, the contextual gradient can be integrated into standard gradient-based optimization algorithms—including gradient descent, stochastic gradient descent, and momentum-based methods—as a substitute for the true gradient (14). As we show later, the resulting algorithm enjoys similar convergence guarantees.

## 4.2 Contextual Gradient Descent Algorithm

In this section, we incorporate the contextual gradient into a gradient descent framework and propose Algorithm 1, which is suitable for cases where the solution space $\mathcal{X}$ is bounded and convex—such as the price-setting newsvendor problem explored in our numerical experiments.

Here, $\Pi_{\mathcal{X}}(\cdot): \mathbb{R}^d \to \mathcal{X}$ denotes the Euclidean projection onto the feasible set $\mathcal{X}$.

The contextual gradient descent algorithm follows a similar iterative scheme as classical gradient descent. As with other gradient-based methods, the choice of step size $\eta^r$ plays a critical role in ensuring convergence. In the following section, we analyze different step size policies that guarantee convergence under various assumptions. Details of the step size strategies used in our implementation are discussed in Appendix EC.1.

**Algorithm 1** Contextual Gradient Descent (CGD) Algorithm

**Input:** Initial solution $\boldsymbol{x}^0$, covariate $\boldsymbol{z}$, dataset $\{(\boldsymbol{x}^i, \boldsymbol{z}^i, \boldsymbol{y}^i)\}_{i=1}^N$

**Output:** Final solution $\hat{\boldsymbol{x}}^*$

```
Set iteration counter $r = 0$
while Stopping criterion is not met do
    Compute contextual gradient $\hat{G}_N(\boldsymbol{x}^r; \boldsymbol{z})$ using (12)
    Select step size $\eta^r$
    Update solution: $\boldsymbol{x}^{r+1} = \Pi_X(\boldsymbol{x}^r - \eta^r \hat{G}_N(\boldsymbol{x}^r; \boldsymbol{z}))$
    Increment iteration: $r \leftarrow r + 1$
end while
return $\hat{\boldsymbol{x}}^* = \boldsymbol{x}^r$
```

## 5 Convergence Analysis

In this section, we establish the convergence guarantees of the CGD algorithm under various structural conditions of the loss function. In the whole section, we denote $\boldsymbol{x}_N^r$ as the solution at the $r$ th iteration of Algorithm 1, and suppose that Lemma 1 holds for the weight function applied in the algorithm. We first consider the convex case, and derive an upper bound on the optimality gap of the CGD algorithm in Theorem 2. We then strengthen the analysis in the strongly convex case by showing the convergence of the decision sequence $\|\boldsymbol{x}_N^k - \boldsymbol{x}^*\|$ in Theorem 3. Finally, we extend the analysis to the general non-convex setting, and analyze convergence to stationary points under two step-size policies.

### 5.1 Convergence under Convex Case

In this subsection, we provide the convergence guarantee of the CGD algorithm when the loss function $l(\boldsymbol{x}, \boldsymbol{y})$ is convex in $\boldsymbol{x}$. The following theorem establishes an upper bound on the optimality gap after $k$ iterations.

THEOREM 2 **(Error Bound in the Convex Case).** *Let $\eta^r$ denote the step size at the $r$-th iteration of Algorithm 1. Suppose $l(\boldsymbol{x}, \boldsymbol{y})$ is convex in $\boldsymbol{x}$, and Assumptions 2, 4(a), (b), and (e) hold. Then, for any $\zeta > 0$, there exists $N_0$ such that for all $N > N_0$, after $k$ iterations:*

$$\min_{0 \le r \le k} \left\{ \mathbb{E}_{f(\boldsymbol{y}; \boldsymbol{x}_N^r, \boldsymbol{z})}[l(\boldsymbol{x}_N^r, \boldsymbol{y})] - \mathbb{E}_{f(\boldsymbol{y}; \boldsymbol{x}^*, \boldsymbol{z})}[l(\boldsymbol{x}^*, \boldsymbol{y})] \right\} \le \frac{3}{2}\zeta + \frac{\|\boldsymbol{x}_N^0 - \boldsymbol{x}^*\|^2 + (L_3^c)^2 \sum_{r=0}^k (\eta^r)^2}{2 \sum_{r=0}^k \eta^r} + \frac{\varepsilon L_2 \sum_{r=0}^k \eta^r \|\boldsymbol{x}_N^r - \boldsymbol{x}^*\|}{\sum_{r=0}^k \eta^r}.$$

*Here, $L_3^c$ is the upper bound on $\|\nabla_{\boldsymbol{x}} l(\boldsymbol{x}, \boldsymbol{y})\|$, $L_2$ is the Lipschitz constant of $l(\boldsymbol{x}, \boldsymbol{y})$ with respect to $\boldsymbol{y}$, and $\varepsilon$ is the sensitivity parameter from Assumption 2.*

Theorem 2 quantifies the optimality gap between the CGD iterates and the true optimal solution. The three terms on the right-hand side correspond to different sources of error:

- The first term is due to the statistical error of the wSAA approximation, which diminishes as the sample size $N \to \infty$.
- The second term arises from the iterative nature of gradient descent. It vanishes when the step size satisfies the diminishing conditions:

$$\sum_{r=0}^{\infty} \eta^r = \infty \quad \text{and} \quad \frac{\sum_{r=0}^{k} (\eta^r)^2}{\sum_{r=0}^{k} \eta^r} \to 0 \text{ as } k \to \infty.$$

For a constant step size $\eta$, the term stabilizes at $O(\eta)$, and thus can be made small with a sufficiently small $\eta$.

- The last term is caused by the decision-dependent effect. When $\varepsilon = 0$, this term disappears, and the analysis recovers the standard convergence of gradient descent on the wSAA objective. However, when $\varepsilon > 0$, it requires that the convergence of $\boldsymbol{x}^r$ to the optimal solution $\boldsymbol{x}^*$ such that the term decrease to a sufficient small value.

Consequently, it is important to know whether and how quickly the solution sequence converges to the optimal point. As we will show, the strong convexity is sufficient condition for the sequence to converge. Therefore, we investigate the distance to the optimal solution under the strongly convex condition in the rest of this section. Specifically, we focus on the loss function satisfying the condition that for all $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{X}$, we have

$$l(\boldsymbol{x}_2, \boldsymbol{y}) \geq l(\boldsymbol{x}_1, \boldsymbol{y}) + \nabla_{\boldsymbol{x}} l(\boldsymbol{x}_1, \boldsymbol{y})^\top (\boldsymbol{x}_2 - \boldsymbol{x}_1) + \frac{\gamma}{2} \|\boldsymbol{x}_2 - \boldsymbol{x}_1\|^2,$$

where $\gamma$ is a coefficient measuring the strength of strong convexity.

Following the approach of Mendler-Dünner et al. (2020), we establish convergence by introducing an intermediate concept called the stable point, which is formally designed as follows.

DEFINITION 2 (STABLE POINT). Given a covariate $\boldsymbol{z}$, a point $\boldsymbol{x}_{PS}$ is called a *stable point* if it satisfies the following fixed-point condition:

$$\boldsymbol{x}_{PS} = \arg\min_{\boldsymbol{x} \in X} \mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_{PS}, \boldsymbol{z})} [l(\boldsymbol{x}, \boldsymbol{y})]. \tag{15}$$

We now state the distance bound between the CGD solution sequence and the stable point $\boldsymbol{x}_{PS}$ in Proposition 1.

PROPOSITION 1 **(Distance to Stable Points)**. *Suppose Assumptions 2, 4(a), (c), and (d) hold, and that $l(\boldsymbol{x}, \boldsymbol{y})$ is $\gamma$-strongly convex in $\boldsymbol{x}$. Assume at least one stable point $\boldsymbol{x}_{PS}$ exists. Let $\boldsymbol{x}_N^r$ denote the solution at iteration $r$ of Algorithm 1, and suppose a constant step size $\eta$ is used. Define constants $A = \gamma - \varepsilon L_1^c$ and $B = L_1^c \sqrt{1 + \varepsilon^2}$.*

*If $A > 0$, then for any constant step size $\eta$ satisfying*

$$0 < 2B^2\eta^2 - 2A\eta + 1 < 1,$$

*there exists a sample size $N_0$ such that for all $N > N_0$, the following holds:*

- *if* $\|\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}\| > UB(\eta,\zeta)$*, then*

$$\|\boldsymbol{x}_N^{k+1} - \boldsymbol{x}_{PS}\| < \|\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}\|,$$

*where* $\zeta > 0$ *is a small constant and*

$$UB(\eta,\zeta) = \frac{\zeta}{2(A-\eta B^2)}\left(1+\sqrt{2}\eta B + \sqrt{-2B^2\eta^2 + 2A\eta + (1+\sqrt{2}\eta B)^2}\right). \tag{16}$$

- *Moreover, when* $k \to \infty$*, we have*

$$\lim_{k\to\infty} \|\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}\| < UB(\eta,\zeta). \tag{17}$$

Intuitively, Proposition 1 indicates that when the distance between $\boldsymbol{x}_N^k$ and the stable point $\boldsymbol{x}_{PS}$ is relatively large, Algorithm 1 will produce a solution in the next iteration, i.e., $\boldsymbol{x}_N^{k+1}$, closer to $\boldsymbol{x}_{PS}$. Eventually, the distance $\|\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}\|$ will be bounded above by $UB(\eta,\zeta)$, which is of order $O(\zeta)$. Therefore, when the sample size is sufficiently large such that the estimation error $\zeta$ is sufficiently small, the sequence $\{\boldsymbol{x}_N^k\}$ will converge to the stable point $\boldsymbol{x}_{PS}$ almost surely.

Then, we provide the convergence rate of the algorithm toward the stable point in Proposition 2.

PROPOSITION 2. *(**Convergence rate**) When Proposition 1 holds and* $\|\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}\| > 2UB(\eta,\zeta)$*, the distance between* $\boldsymbol{x}_{PS}$ *and* $\boldsymbol{x}_N^k$ *converges at an exponential rate. Specifically, we have,*

$$\|\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}\| \le \max\left\{C^k\|\boldsymbol{x}_N^0 - \boldsymbol{x}_{PS}\|,\ 2UB(\eta,\zeta)\right\}, \tag{18}$$

*where* $C = \sqrt{1-\eta A + \eta^2 B^2} < 1$.

We now focus on the distance between the CGD solution and the true optimal solution. Lemma 2 characterizes the relationship between the stable point $\boldsymbol{x}_{PS}$ and the optimal solution $\boldsymbol{x}^*$.

LEMMA 2 (**Theorem 4.3 in Mendler-Dünner et al. (2020)**). *Suppose that* $l(\boldsymbol{x},\boldsymbol{y})$ *is* $L_y$*-Lipschitz in* $\boldsymbol{y}$ *and strongly convex, and that Assumption 2 is satisfied. Then, for every stable point* $\boldsymbol{x}_{PS}$*, we have*

$$\|\boldsymbol{x}^* - \boldsymbol{x}_{PS}\| \le \frac{2L_y\varepsilon}{\gamma}.$$

Combining Proposition 2 with Lemma 2, we obtain an upper bound on the total error of the CGD algorithm in Theorem 3.

THEOREM 3 (**Convergence to the Optimal Solution under the Strongly Convex Case**). *When Proposition 1 holds, for any* $\zeta > 0$*, there exists a sample size* $N_0$ *such that, for all* $N > N_0$*,*

$$\|\boldsymbol{x}_N^k - \boldsymbol{x}^*\| \le \max\left\{C^k\|\boldsymbol{x}_N^0 - \boldsymbol{x}_{PS}\|,\ 2UB(\eta,\zeta)\right\} + \frac{2L_y\varepsilon}{\gamma},$$

*where* $UB(\eta,\zeta)$ *is defined in Proposition 1, and* $C$ *is defined in Proposition 2.*

With Theorem 3 established, we now analyze the convergence of the third term in Theorem 2. Denote $K = \left\lceil \frac{\log(2UB(\eta,\zeta)) - \log(\|\boldsymbol{x}_N^0 - \boldsymbol{x}_{PS}\|)}{\log C} \right\rceil$. When $k \geq K$, $C^k \|\boldsymbol{x}_N^0 - \boldsymbol{x}_{PS}\| \leq 2UB(\eta,\zeta)$. Therefore, according to Proposition 2, we have $\|\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}\| \leq 2UB(\eta,\zeta)$ when $k \geq K$, and otherwise, $\|\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}\| \leq C^k \|\boldsymbol{x}_N^0 - \boldsymbol{x}_{PS}\|$. Thus, for sufficiently large $k \geq K$, according to Theorem 3, we obtain

$$\sum_{r=0}^{k} \|\boldsymbol{x}_N^r - \boldsymbol{x}^*\| \leq \frac{\|\boldsymbol{x}_N^0 - \boldsymbol{x}_{PS}\|(1 - C^{K+1})}{1 - C} + 2(k-K)UB(\eta,\zeta) + \frac{2kL_y\varepsilon}{\gamma}.$$

The first term of the right side comes from summing the geometric sequence $C^k \|\boldsymbol{x}_N^0 - \boldsymbol{x}_{PS}\|$ for $k \leq K$. The second term is the summation of $2UB(\eta,\zeta)$ for the other $k - K$ terms in the sequence. The third term is the summation of the constant term $\frac{2L_y\varepsilon}{\gamma}$ for $k$ times.

When a constant step size $\eta^r = \eta$ is chosen, the third term of Theorem 2 becomes

$$\begin{aligned}
&\frac{\varepsilon L_2 \sum_{r=0}^{k} \eta^r \|\boldsymbol{x}_N^r - \boldsymbol{x}^*\|}{\sum_{r=0}^{k} \eta^r} \\
=&\frac{\varepsilon L_2 \sum_{r=0}^{k} \eta \|\boldsymbol{x}_N^r - \boldsymbol{x}^*\|}{\sum_{r=0}^{k} \eta} \\
=&\frac{\varepsilon L_2}{k} \sum_{r=0}^{k} \|\boldsymbol{x}_N^r - \boldsymbol{x}^*\| \\
\leq&\frac{\varepsilon L_2}{k} \left( \frac{\|\boldsymbol{x}_N^0 - \boldsymbol{x}_{PS}\|(1 - C^{K+1})}{1 - C} + 2(k-K)UB(\eta,\zeta) + \frac{2kL_y\varepsilon}{\gamma} \right), \quad k \geq K.
\end{aligned}$$

Accordingly, as $k \to \infty$, $\frac{\varepsilon L_2}{k} \frac{\|\boldsymbol{x}_N^0 - \boldsymbol{x}_{PS}\|(1 - C^{K+1})}{1 - C}$ becomes zero. Therefore, we have

$$\lim_{k \to \infty} \frac{\varepsilon L_2}{k} \sum_{r=0}^{k} \|\boldsymbol{x}^* - \boldsymbol{x}_N^r\| = 2\varepsilon L_2 UB(\eta,\xi) + 2\varepsilon^2 L_2 L_y / \gamma.$$

As mentioned previously, by selecting a sufficiently small step size $\eta$ and sufficiently large sample size $N$, the term $UB(\eta,\zeta)$ decreases to zero. Therefore, the optimality gap in Theorem 2 ultimately converges to $2\varepsilon^2 L_2 L_y / \gamma$.

The above discussion shows that as the number of samples and iterations grows large, the bound in Theorem 2 converges to a constant directly proportional to $\varepsilon$, the sensitivity of the conditional distribution's shift in response to changes in $\boldsymbol{x}$, and inversely proportional to $\gamma$, the strength of strong convexity.

The remaining gap is expected due to the information loss resulting from the simplified computation of the contextual gradient. Specifically, the contextual gradient omits one component of the original derivative of the weighted SAA objective, namely $\partial_{\boldsymbol{x}} w^i(\boldsymbol{x}, \boldsymbol{z}) l(\boldsymbol{x}, \boldsymbol{y}^i)$. The technique decreases computational complexity significantly, but may lead to suboptimal outcomes. Nevertheless, our analysis shows that this suboptimality can be effectively controlled by the strength of strong convexity. In other words, strong convexity compensates for the partially neglected information related to decision-dependent effects, thereby ensuring reliable optimization performance. In subsequent sections, we refer to $\frac{\varepsilon}{\gamma}$ as the *critical ratio* to measure this critical interplay between decision dependency and strong convexity, and further explore its impact on the optimality gap through numerical experiments in Section 6.2.1.

REMARK 1. As discussed in Mendler-Dünner et al. (2020), the constant $A = \gamma - \varepsilon L_1^c$ serves as a critical threshold determining the convergence behavior of gradient descent methods. Specifically, when $A \leq 0$, the algorithm may fail to converge because a stable point may not exist. The condition $A > 0$ also highlights the compensating role played by strong convexity: sufficient strong convexity is necessary to offset the information loss from ignoring parts of the decision-dependent effects, thereby ensuring robust and reliable algorithmic performance.

### 5.2 Convergence under the Non-convex Case

To evaluate the performance of the CGD algorithm under broader conditions, we now extend our convergence analysis to the non-convex setting. Specifically, we investigate the convergence behavior of CGD toward points where the expected gradient is sufficiently small. These points closely resemble the notion of a *stationary point* in classical gradient descent literature, and we adopt this terminology accordingly. According to Theorem 1, convergence to such a stationary point provides a meaningful indication of proximity to optimality in the original problem.

It should be noted that in this section, we only discuss the unconstrained CSO problem. At this time, Assumption 1 no longer requires that the value space of $x$ be convex and bounded; the other three assumptions remain unchanged. Meanwhile, in Algorithm 1, the update method no longer requires the projection operation, that is, $\boldsymbol{x}_N^{r+1} = \boldsymbol{x}_N^r - \eta^r \hat{G}_N(\boldsymbol{x}_N^r;\boldsymbol{z}))$

Propositions 3, 4, and 5 present convergence guarantees under different step size policies, including diminishing step sizes, Armijo-type backtracking, and constant step size.

PROPOSITION 3 **(Convergence under Diminishing Step Size)**. *Suppose Assumptions 1, 2, 3, and 4(a)-(c) hold. Assume that the objective function $l(\boldsymbol{x},\boldsymbol{y})$ is twice differentiable in $\boldsymbol{x}$, and its absolute value is bounded by a constant $L_4$. If the gradient of the distribution density is also bounded by a constant $L_5$, and the step size $\eta^r$ is diminishing such that $\sum_{r=0}^{\infty} \eta^r = \infty$, then there exists a sample size $N_0$ such that for all $N > N_0$, the limit point $\bar{\boldsymbol{x}}$ of the sequence generated by the CGD algorithm satisfies:*

$$\mathbb{E}_{f(\boldsymbol{y}|\bar{\boldsymbol{x}},\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\bar{\boldsymbol{x}},\boldsymbol{y})] = 0. \tag{19}$$

PROPOSITION 4 **(Convergence under Armijo Step Size)**. *Suppose Assumptions 2 and 4(a) hold. Assume the CGD algorithm adopts the Armijo step size rule with sufficient decrease parameter $\sigma$. When the sample size $N$ is sufficiently large, limit point $\bar{\boldsymbol{x}}$ of the sequence generated by the CGD algorithm has a bounded expected gradient:*

$$\left\|\mathbb{E}_{f(\boldsymbol{y}|\bar{\boldsymbol{x}},\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\bar{\boldsymbol{x}},\boldsymbol{y})]\right\| \leq \frac{\varepsilon L_1}{1-\sigma}. \tag{20}$$

PROPOSITION 5 (**Convergence under Constant Step Size**). *Suppose Assumptions 2 and 4 hold. When $N \to \infty$, and a fixed step size $\eta \leq \min\left\{\frac{1}{L_1^c}, \frac{1}{L_3^c}\right\}$ is used, we have:*

$$\min_{r=0,\ldots,k} \left\|\mathbb{E}_{\boldsymbol{y}\sim f(\boldsymbol{y}|\boldsymbol{x}_N^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_N^r,\boldsymbol{y})]\right\|^2 \leq \frac{2\left(\mathbb{E}_{\boldsymbol{y}}[l(\boldsymbol{x}_N^0,\boldsymbol{y})] - \mathbb{E}_{\boldsymbol{y}}[l(\boldsymbol{x}^*,\boldsymbol{y})]\right)}{\eta(k+1)} + \frac{L_3^c L_1 \varepsilon}{2}. \tag{21}$$

As shown in the propositions, the convergence behavior of the CGD algorithm varies under different step-size policies, offering trade-offs between assumption generality and convergence strength. Specifically, Propositions 3 and 4 show that, under diminishing and Armijo step sizes, the algorithm only achieves asymptotic convergence—that is, convergence may require an infinite number of iterations. Between these two settings, the CGD algorithm with a diminishing step size converges to a point with zero expected gradient, thus reaching an exact stationary point. In contrast, with the Armijo step size, the algorithm converges to a point whose expected gradient norm is bounded above by a constant, $\frac{\varepsilon L_1}{1-\sigma}$. Notably, however, the Armijo rule imposes weaker assumptions, as it does not require Assumptions 1 and 3. Proposition 5 further complements these results by establishing a finite-iteration convergence guarantee under a constant step size. After $k$ iterations, the minimum expected gradient norm among all iterates is bounded by $O(1/\sqrt{k})$, and ultimately converges to $\frac{L_3^c L_1 \varepsilon}{2}$.

Although we only present the solution sequence convergence to stationary points, it is the best that can generally be expected in non-convex settings—even for standard gradient descent algorithms.

## 6 Numerical Results

In this section, we evaluate the convergence behavior and performance of the proposed CGD algorithm, and compare it against several baseline methods.

We begin by demonstrating the convergence properties of CGD when weights are derived using different machine learning techniques in Section 6.2. The effect of the *critical ratio*, defined as $\frac{\varepsilon}{\gamma}$, on the optimization performance is validated and the sample efficiency of the CGD algorithm is examinedto show the reduction in optimality gap with increasing data size.

We then examine the efficiency and effectiveness of the CGD algorithm in Section 6.3. The CGD algorithm is first compared with the wSAA approach, using several optimization methods. Our results show that CGD achieves comparable solution quality with significantly lower computational time. Then, we compare CGD with both the expectation-based estimate-then-optimize (EETO) method and the classical estimate-then-optimize (ETO) framework (also known as the predict-then-optimize framework). The results highlight the advantage of CGD, especially under model misspecification, owing to its non-parametric design and robustness to decision-dependent effects.

All experiments are conducted on a server equipped with the following specifications. FusionServer G5500 V6 hardware platform, 128 CPU cores, 2TB of RAM, running Ubuntu 22.04.4 LTS operating system on a Linux x86_64 architecture.

## 6.1 Experiment Setup

We demonstrate the effectiveness of the proposed CGD algorithm using the price-setting newsvendor problem described in Example 5. Our data generation process closely follows the setup in Lin et al. (2022), where a classical newsvendor problem (without price setting) is extended to include pricing as a decision variable.

The contextual features $\boldsymbol{z}$ are independently drawn from a 4-dimensional Gaussian distribution $N(0,\Sigma)$, where $\Sigma$ is a diagonal matrix with entries $1,2,3,4$. Prices are independently sampled from a uniform distribution $U(10,200)$.

We consider two types of demand models:

- **Homoscedastic Model:** The first model assumes that the decision (price) affects only the mean of demand, which is used in EC.1 in Lin et al. (2022). Specifically, demand is generated as,

$$D = \max\{0, 60 - \varepsilon p + 12a^T(\boldsymbol{z} + 0.25\phi) + 5b^T\boldsymbol{z}\theta\}, \tag{22}$$

where $\phi \sim N(0,I)$ is a 4-dimensional random vector, $I$ is the identity matrix, $\theta \sim N(0,1)$ is a scalar random variable, $a = (0.8,1,1,1)^T$, and $b = (-1,1,0,0)^T$. This model satisfies the $\varepsilon$-sensitivity condition defined earlier, since the price $p$ affects only the mean linearly.

- **Heteroscedastic Model:** The second model introduces heteroscedasticity by allowing the variance of demand to depend on the price:

$$D = \max\{0, 60 - \varepsilon_1 p + 12a^T\boldsymbol{z} + \varepsilon_2^2 pa^T\phi + 5b^T\boldsymbol{z}\theta\}, \tag{23}$$

where $\varepsilon_1$ and $\varepsilon_2$ are constants, which is inspired by (4.1) in Harsha et al. (2021). As noted in Remark 3.2 of Perdomo et al. (2020), this model is $\max\{|\varepsilon_1|,|\varepsilon_2|\}$-sensitive. The introduction of price-dependent variance violates many common structural assumptions and increases the complexity of the learning task.

The weights are determined by the ML methods trained by the i.i.d samples from the distributions set above. The hyperparameters associated with each ML-based weighting strategy (e.g., kNN, kernel regression, CART, and RF) are tuned via grid search. The initial solution for the CGD algorithm is set as $(p_0,q_0) = (150,100)$. The iteration stops when the $\ell_2$-norm of the contextual gradient falls below 0.01 or the maximum number of iterations is reached.

We evaluate the performance of each method using the *optimality gap*, defined as:

$$\text{optimality gap} = \frac{|\mathbb{E}_y[l(\boldsymbol{x}^*,y)] - \mathbb{E}_y[l(\boldsymbol{x},y)]|}{|\mathbb{E}_y[l(\boldsymbol{x}^*,y)]|}.$$

Here, $\boldsymbol{x}^*$ denotes the optimal solution under the true data-generating distribution. While the true conditional distribution $f(y \mid \boldsymbol{x},\boldsymbol{z})$ is known during data generation and evaluation, it is not accessible to the CGD algorithm during optimization.

## 6.2 Convergence Performance

In this section, we evaluate the convergence behavior of the CGD algorithm from several perspectives.

### 6.2.1 Convergence under Different Critical Ratios

We first examine the convergence performance of the CGD algorithm under various weighting strategies within the newsvendor pricing problem. The full problem setup is described in Example 5.

To create a strongly convex instance, we augment the original loss function $l(\boldsymbol{x}, y)$ with a quadratic regularization term:

$$l_{\text{pen}}(\boldsymbol{x}, y) = l(\boldsymbol{x}, y) + \lambda_p (p - p_{\text{sd}})^2 + \lambda_q (q - q_{\text{sd}})^2,$$

where $\lambda_p$ and $\lambda_q$ are regularization coefficients, and $p_{\text{sd}}$, $q_{\text{sd}}$ are anchor (or reference) points. In our experiment, we set the regularization coefficients as $\lambda_p = \lambda_q = 0.5$. Accordingly, $\gamma = 1$ for both $p$ and $q$. Moreover, we set $p_{sd} = 60$, $q_{sd} = 100$.

Practically, these anchor points can be interpreted as initial or historically preferred pricing and ordering levels—e.g., derived from prior domain experience, business policy, or cost-based heuristics. The penalization encourages the solution to remain close to these baseline values unless significantly better outcomes are found, thereby improving decision stability of the companies.

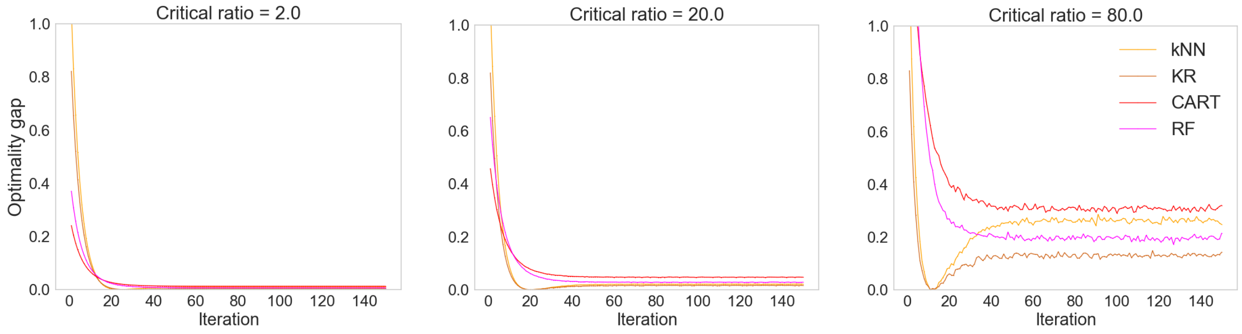
In this penalized formulation, the strong convexity constant is given by $\gamma = \min\{\lambda_p, \lambda_q\}$, while the Lipschitz constant with respect to $y$, denoted $L_y$, is approximately $\max\{p, s\} \approx 150$ in our setting. The sensitivity coefficient $\varepsilon$ is defined in equation (22).

In the following experiments, we vary $\varepsilon$ to empirically investigate how the critical ratio $\varepsilon/\gamma$ influences convergence behavior and final solution quality. The data size for each experiment is fixed at 1000. For each critical ratio setting, we repeat the experiment 20 times with different random seeds to ensure statistical robustness. The demand model used in this experiment is the heteroscedastic model; results under the homoscedastic model exhibit similar patterns and are therefore omitted for brevity.

The convergence results are shown in Figure 1. The figure plots the optimality gaps of the CGD algorithm across iterations for different critical ratios and weighting strategies. As stated in Proposition 2, when the optimality gap is large, it will decrease at an exponential rate. When it drops to a certain extent, it is not guaranteed to continue to decrease further, but it will start to converge. The final gap is closely related to the magnitude of the critical ratio.

To further highlight the impact of the critical ratio, Figure 2 displays the average final optimality gaps corresponding to various critical ratios. As the figure shows, for all four weighting methods, the final gap increases as the critical ratio grows, confirming the theoretical relationship between the critical ratio and optimality gap of the convergence point.

Based on the experimental results, we draw the following conclusions:



(a) Convergence with critical ratio of 2 (b) Convergence with critical ratio of 20 (c) Convergence with critical ratio of 80

**Figure 1** Convergence of CGD algorithm under different critical ratios

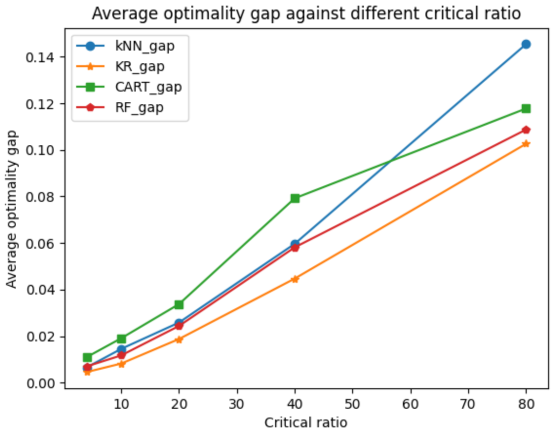


**Figure 2** Average final optimality gap vs. critical ratio

1. When the data volume is sufficient (e.g., 1000 samples applied in this section), the CGD algorithm with different ML-based weighting methods consistently converges to solutions with stable and bounded optimality gaps.

2. As illustrated in Figure 2, the magnitude of the critical ratio $\varepsilon/\gamma$ determines the final solution quality. A smaller critical ratio leads to the stable point being closer to the optimal solution, resulting in a smaller optimality gap—and vice versa. This validates our theoretical insight that when the unknown distribution is significantly influenced by decision variables, the convexity of the objective loss function can compensate for the incomplete characterization of decision-dependent effects and still yield high-quality solutions.

In practice, these findings suggest a practical approach to assess the potential performance of the CGD algorithm. Since the functional form of $l(x,y)$ is known, parameters such as $L_y$ and $\gamma$ in the critical ratio can be directly computed. Moreover, using regression or other predictive tools, decision-makers can estimate the decision-dependence coefficient $\varepsilon$ that quantifies the sensitivity of the conditional distribution to decision variables. Together, these quantities allow for a pre-computation of an approximate critical ratio, which can guide algorithm configuration and set realistic performance expectations.

### 6.2.2 Sample Efficiency

In this section, we evaluate the sample efficiency of the CGD algorithm by examining its performance under varying data sizes. Specifically, we fix the critical ratio at 2 to ensure that the stable point is sufficiently close to the optimal solution. The kNN weighting method is used to determine the weights, with the number of neighbors set as $k = \lceil 0.1\sqrt{N} \rceil$, and the heteroscedastic model is applied to generate the demand. Results under the homoscedastic model and other weighting methods exhibit similar patterns and are therefore omitted for brevity. For each sample size, we generate 100 stochastic instances to measure performance variability.

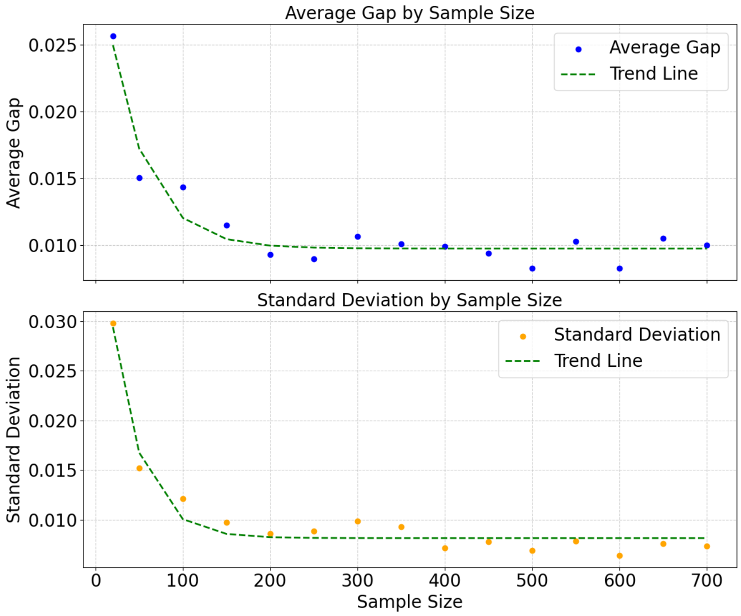


**Figure 3** Mean and variance of the optimality gap under different sample sizes.

Figure 3 depicts the mean and standard deviation of the optimality gaps achieved by the CGD algorithm across different sample sizes. Each point represents the mean and variance at a given sample size, and the dashed lines indicate the overall trends. As shown in the figure, the CGD algorithm does not require a large data size to perform well when critical ratio is relatively small. As the sample size increases, the optimality gap consistently decreases. Simultaneously, the variance across trials also diminishes, indicating improved stability of the CGD algorithm with larger datasets. Notably, performance stabilizes when the sample size reaches approximately 300.

These observations are consistent with Proposition 1. When the step size is sufficiently small, the upper bound on the distance between the CGD solution and the stable point is controlled by the estimation error parameter $\zeta$. As the sample size grows, the estimation error in the nonparametric weights becomes smaller, which in turn drives the CGD iterates closer to the stable point—resulting in lower optimality gaps.

## 6.3 Comparison to Other Methods

In this section, we compare the optimization performance of the CGD algorithm against alternative solution approaches.

### 6.3.1 Comparison with Directly Solving the Weighted SAA Model and Direct Gradient Descent

Since the CGD algorithm is derived from the weighted SAA framework, a natural question arises: how does it compare to directly solving the weighted SAA problem? However, due to decision-dependent effects, directly solving the weighted SAA formulation presents considerable computational challenges. To evaluate the effectiveness of our method, we consider the following two benchmark strategies:

- **Discretization-based Optimization** (marked as Discretization): Following the method proposed by Bertsimas and Kallus (2019), this approach discretizes the decision space to ensure tractability. It is applicable regardless of the weight function type. In our experiment, the price $p$ and quantity $q$ are both uniformly discretized over their respective ranges—(10, 200) and (0, 200)—into 100 grid points each. The optimal solution is determined by evaluating $\hat{g}(p,q,z)$ over the resulting 10,000 grid combinations and selecting the one that yields the minimum objective value.
- **Direct Gradient Descent on Weighted SAA** (marked as GD): The other approach applies standard gradient descent directly to the weighted SAA objective function in equation (5). As we dicussed before, this approach is only applicable for weight functions that are differentiable, such as kernel regression.

We evaluate both the optimality gap and computational time of these benchmark methods relative to the CGD Algorithm. For this experiment, we set the critical ratio to 2; use a dataset of size 1000, and repeat each trial 20 times. The demand is generated using the heteroscedastic model; the homoscedastic model produces similar trends and is omitted for brevity.

Figure 4 presents a violin plot that compares the optimality gaps of different solution methods. A violin plot combines a boxplot and a kernel density plot to visualize both the central tendency and distributional characteristics of each method. As shown in the figure, the CGD algorithm consistently achieves better or comparable solution quality relative to the discretization approach across all machine learning-based weighting functions. Additionally, for kernel-based weights—where both CGD and GD are applicable—CGD demonstrates slightly lower optimality gaps, indicating enhanced effectiveness and stability.

As discussed previously, the primary motivation behind the CGD algorithm is to improve computational efficiency. Table 1 presents the average computation times of each method. As shown, CGD significantly reduces computational time compared to the discretization approach, showcasing its efficiency for problems of similar scale. Moreover, on average, CGD executes at only 34% of the time required by direct gradient descent.

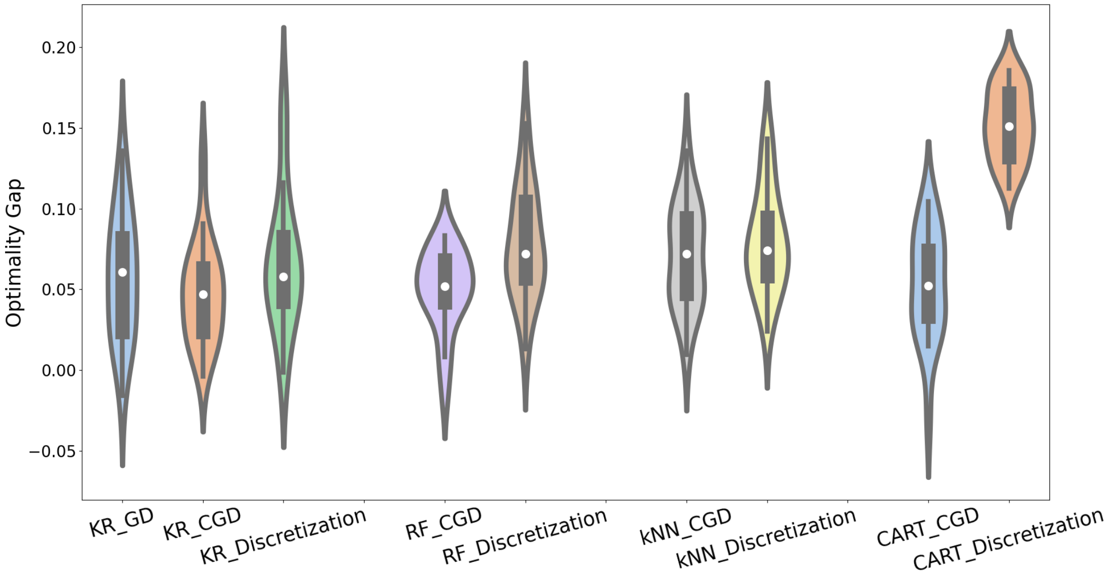


**Figure 4** Optimality gap of different solution methods

**Table 1** Computation time (s) comparison between CGD, Discretization and GD.

| Strategies | kNN | KR | CART | RF |
|---|---|---|---|---|
| Discretization | 27.79 | 25.95 | 26.99 | 756.62 |
| GD | - | 1.04 | - | - |
| CGD | 0.41 | 0.35 | 0.37 | 7.59 |

To further explore scalability, Figure 5 compares CGD and GD across varying data sizes. Regardless of the dataset size, CGD is three times faster than GD. This result demonstrates that the efficiency benefit of CGD grows with the scale of the problem. Note that wSAA models must be solved for each covariate combination, which can be numerous, and therefore the CGD algorithm significantly reduces the total computation time. Additionally, unlike GD—which is limited to differentiable weight functions—CGD applies broadly to a variety of machine learning–based weighting schemes, making it more versatile.

These results highlight that CGD not only matches or outperforms other methods in solution quality, but also does so with considerably less computational effort. This dual advantage positions CGD as a highly effective approach for solving weighted SAA problems with decision-dependent uncertainty, particularly in settings where both accuracy and efficiency are critical.

### 6.3.2 Comparison to other benchmarks

In this section, we compare the CGD algorithm with additional benchmark methods widely used in the literature.

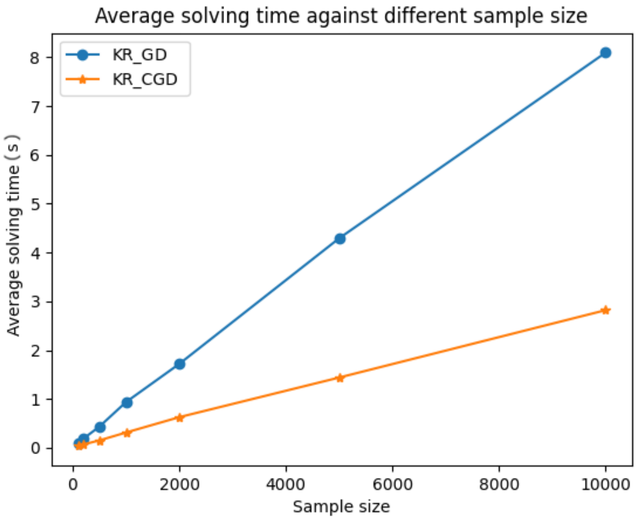


**Figure 5** Computation time comparison between CGD and GD

- **Estimate-Then-Optimize (ETO):** This widely adopted framework, implemented in prior research such as Ban and Rudin (2018), Demirović et al. (2019), assumes that demand is a linear function of the price:

$$y = D(p, \mathbf{z}) = \beta_0 - \beta_1 p + \beta_2^T \mathbf{z} + \sigma_s \theta, \quad \theta \sim N(0, 1).$$

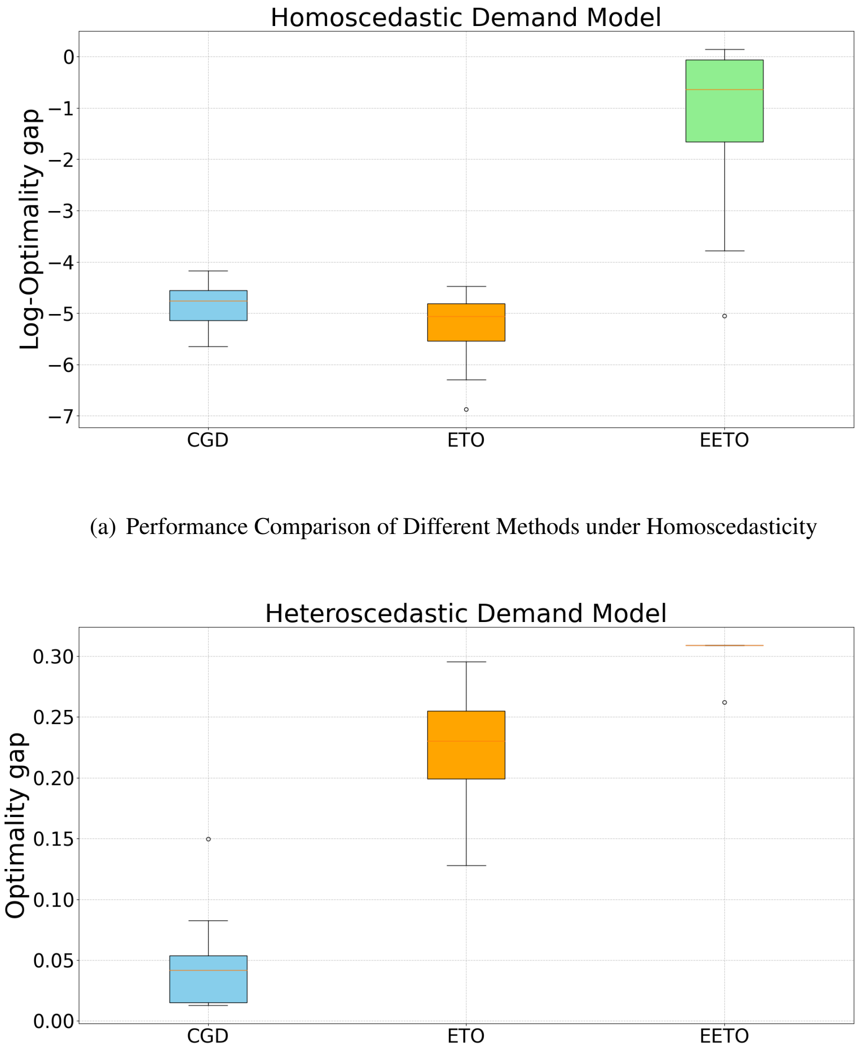
Specifically, ETO begins by estimating the parameters $\beta_0$, $\beta_1$, and $\beta_2$ from historical data using ordinary least squares (OLS) regression. The noise parameter $\sigma_s$ is then inferred from the standard deviation of the residuals. Once the model is estimated, the optimal price $p$ and order quantity $q$ are determined by maximizing the estimated profit under the assumed linear demand model.

- **Expected Value Estimate-Then-Optimize (EETO):** This approach follows the framework of Ferreira et al. (2016), Liu et al. (2021b). EETO begins by building a regression model $\hat{D}(p, \mathbf{z})$ for demand using both the price and covariates. PyCaret [1] is applied to build the model, including automatically training the various models, doing cross-validation, and selecting the best one according to the predictive performance. Model is performed from a comprehensive pool of candidate models, such as kNN and lightGBM, etc. Once the best-performing regression model is selected, the estimated function $\hat{D}(p, \mathbf{z})$ is substituted directly into the optimization routine to derive the optimal decision.

Both ETO and EETO have distinct modeling strengths. ETO explicitly characterizes the scale of random variation in demand, making it suitable when uncertainty needs to be captured parametrically. In contrast, EETO is capable of incorporating complex decision-dependent effects, especially when sophisticated non-linear regression models are employed.

[1] https://pycaret.org/

We conducted 20 experiments for each of the three methods—CGD, ETO, and EETO—under both homoscedastic and heteroscedastic demand models, using a dataset size of 1000. In these experiments, $\varepsilon = \varepsilon_1 = 0.02$ and $\varepsilon_2 = 0.1$; all other parameters are consistent with previous sections. Figure 6 demonstrate the comparison of the methods.



(a) Performance Comparison of Different Methods under Homoscedasticity

(b) Performance Comparison of Different Methods under Heteroscedasticity

**Figure 6** Performance Comparison of Different Methods

Figure 6(a) shows the box plot of optimality gaps in the homoscedastic setting on a log-scale for clearer comparison. As shown in the figure, CGD and ETO deliver comparable performance, both significantly outperforming EETO. The poor performance of EETO is expected because it is insufficient to accurately estimating the distribution of the random component by predicting the mean value alone.

Figure 6(b) illustrates the results under heteroscedastic conditions. Here, ETO's performance deteriorates significantly due to its structural assumptions being violated. Although CGD also experiences a decline in performance, it still maintains an optimality gap within 5%, thanks to its nonparametric flexibility in accommodating complex relationships.

In summary, CGD performs comparably to ETO when ETO's structural assumptions hold and significantly better when those assumptions are violated. In real-world scenarios where the interaction among decisions, covariates, and stochastic outcomes is often complex, CGD's nonparametric nature offers improved robustness and solution quality.

## 7 Case Study

We study a price-setting newsvendor problem through a used-vehicle (UV) sales dataset published at Kaggle[2]. In this section, we still aims to maximize the expected revenue, or equivalently, minimize its negative revenue. Since the revenue is mostly positive, in order to maintain the clarity of the presentation, the objective is set as the maximization of the expected value of the profit function. The case perfectly fits the decision-dependent CSO scenario. On the one hand, the price significantly influences the sales of the UVs. On the other hand, the pricing decision for the used vehicle depends on the covariates describing the vehicle's current market conditions (Manheim Market Report values (MMR), regional market trends) as well as the vehicle's inherent attributes (brands, models, trims, types) and dynamic status (mileage, condition rating).

We pre-process the original data to accommodate the price-setting newsvendor problem, whose detail is furnished in Appendix EC.3.1. After the transition, the columns used in the implementation of the CGD algorithm are described in Table 2, where the first ten columns are used as the covariates and the last three are the parameters of the newsvendor problem.

In this case study, we assume that once the price is acceptable for both the UV market and the customer, the market can respond to customer demands at any time. That is, the monthly sales volume for each type of car in the dataset corresponds to the customer demand for that type of car in the corresponding month at current prices. This provides a foundation for us to treat sales volume as demand in the subsequent process.

We separate the dataset into training (80% of all data) and test set (20% of all data). Through cross-validation in the training set, we find that the Random Forest model performs the best as the weights method. The training result can be found in Table EC.4 in the appendix. Therefore, in the experiments of the CGD algorithm, we use the weights generated by the Random Forest. For each combination of covariate in the test set, we run the CGD algorithm to get the price and order quantity for it.

Moreover, to establish a benchmark of evaluating the solutions, we train 24 machine learning models to capture the relationship between sales, selling price, and the covariates. The training results of the models

[2] https://www.kaggle.com/datasets/syedanwarafridi/vehicle-sales-data

**Table 2** Description of vehicle sales data

| Variable | Type | Description | Statistics |
|---|---|---|---|
| Year | int | Total vehicle usage year up to the time of sale | min 0, max 3 |
| Size | int | Size of the vehicle | 1: compact vehicle, 2: mid-size vehicle, 3: full-size vehicle |
| Type | dummy | Type of the vehicle | Three types in total: Truck, SUV, Sedan |
| Trim order | string | The Trim level of the vehicle in the current model | From 1 to 8, higher trim order corresponds to more luxury trim. |
| Condition group | int | Group of vehicle condition | From 1 to 6, higher group value corresponds to better condition. |
| odometer group | int | Group of vehicle odometer | From 1 to 5, higher group value corresponds to higher odometer. |
| MMR | int | Average Manheim Market Report in a month, indicating the estimated market value of the vehicle. | min 3650, median 18646, max 50800 |
| month | int | The month when the vehicle was sold. | From 1 to 12 |
| year | int | The year when the vehicle was sold. | From 2014 to 2015 |
| selling price | float | The average price in month the vehicle was sold. | min 750, median 18074, max 230k |
| Cost | float | Cost of this type of vehicle | min 730, median 17630, max 56025 |
| salvage value | float | Salvage value of this type of vehicle | min 718, median 17338, max 55080 |
| total sales | int | Monthly sales volume of this type of vehicle | min 0, max 213 |

are provided in Appendix EC.3.1. Upon comparison, the Extra Tree Model emerges as the superior one, with an overall R² of 0.87. This indicates quite good accuracy in predicting the sales volumes based on the covariates and selling price. As such, we take the output of the Extra Tree Model as the ground-truth demands for the given covariates and selling price. For each different covariate combination, the predicted sales are obtained when the covariates and selling price are input into the model, and therefore, the newsvendor revenue can be calculated.

Figure 7 plots the relative improvement of the revenue collected (denoted as final revenue in the figure) under the price and order quantity obtained through the CGD algorithm over that of the original one. The red line in the figure represents the baseline with a ratio of 1, while the green line represents the mean ratio. As shown in the figure, the CGD solution provides a significant and robust improvement in revenue compared to the original price and quantity in general. The relative improvement is 31% on average, and only 3.7% of the test set has slightly lower revenue than the original decision. This indicates that the algorithm can bring significant benefits to wholesalers in real-world scenarios.

## 8 Conclusion and Future Directions

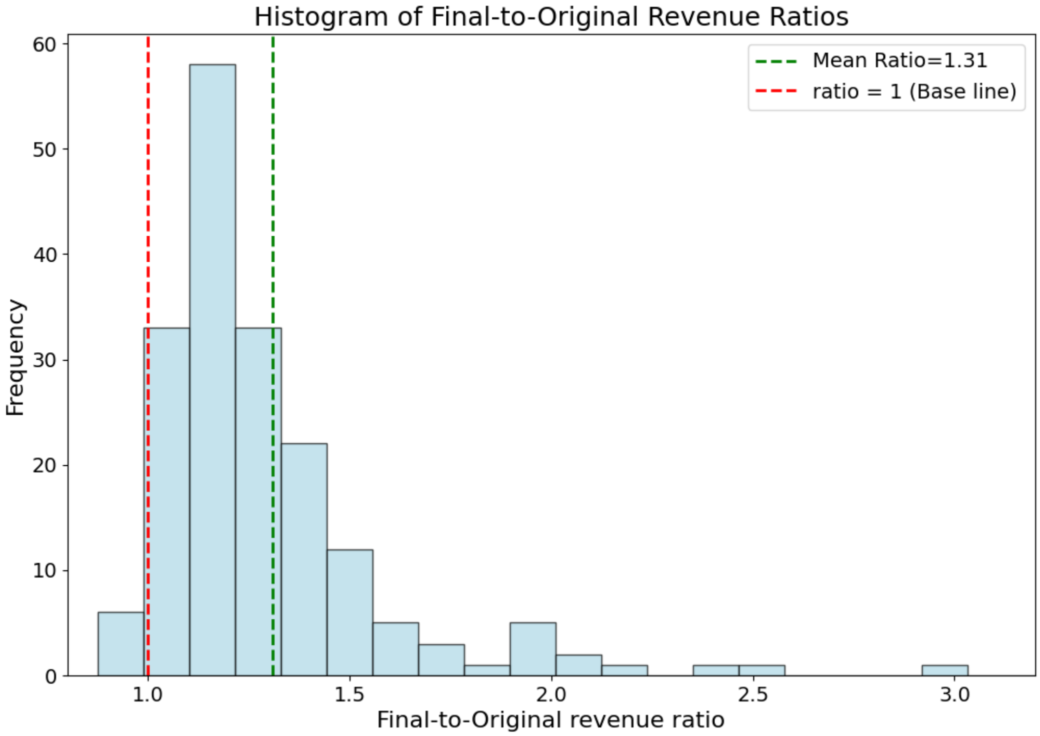


**Figure 7** Histogram of Final to Original Revenue Ratios.

In this paper, we propose a novel nonparametric approach for solving contextual stochastic optimization problems with decision-dependent uncertainty, based on the concept of the contextual gradient. Compared to existing methods, the contextual gradient is both more efficient to compute and more flexible in capturing complex dependencies of the random parameter on both decisions and covariates. We introduce the CGD algorithm, which incorporates the contextual gradient into a gradient descent framework, and provide theoretical guarantees for its convergence.

A key insight from our analysis under strong convexity is that the strength of convexity in the loss function can partially compensate for the unmodeled decision-dependent effects. This theoretical result is supported by empirical evidence, demonstrating that CGD is capable of achieving high-quality solutions even when the full decision-dependency cannot be explicitly modeled.

Looking ahead, there are multiple promising directions for future research. One is the development of alternative algorithms that integrate the contextual gradient, such as adaptations within proximal gradient descent or stochastic gradient descent frameworks. Another direction is the extension of CGD to settings involving multi-stage decisions, or time-varying distributions. These extensions would further broaden the applicability and impact of contextual optimization under decision dependency.

## References

Ban, Gah-Yi, Jérémie Gallien, Adam J Mersereau. 2019. Dynamic procurement of new products with covariate information: The residual tree method. *Manufacturing & Service Operations Management*, 21 (4), 798-815.

Ban, Gah-Yi, Cynthia Rudin. 2018. The big data newsvendor: Practical insights from machine learning. *Operations Research*, 67 (1), 90-108.

Bertsimas, Dimitris, Nathan Kallus. 2019. From predictive to prescriptive analytics. *Management Science*, 66 (3), 1025-1044.

Bertsimas, Dimitris, Christopher McCord. 2018. Optimization over continuous and multi-dimensional decisions with observational data. *Advances in neural information processing systems*, 31.

Chu, Leon Yang, Qi Feng, J George Shanthikumar, Zuo-Jun Max Shen, Jian Wu. 2024. Solving the price-setting newsvendor problem with parametric operational data analytics (ODA). *Management Science*, .

Cristian, Rares, Pavithra Harsha, Georgia Perakis, Brian L Quanz, Ioannis Spantidakis. 2022. End-to-end learning via constraint-enforcing approximators for linear programs with applications to supply chains. *AI for Decision Optimization Workshop of the AAAI Conference on Artificial Intelligence*.

Demirović, Emir, Peter J Stuckey, James Bailey, Jeffrey Chan, Christopher Leckie, Kotagiri Ramamohanarao, Tias Guns. 2019. Predict+ optimise with ranking objectives: Exhaustively learning linear functions. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*. International Joint Conferences on Artificial Intelligence, 1078-1085.

Deng, Yunxiao, Suvrajeet Sen. 2022. Predictive stochastic programming. *Computational Management Science*, 19 (1), 65-98.

Dupacová, Jitka. 2006. Optimization under exogenous and endogenous uncertainty. *University of West Bohemia in Pilsen*, .

Elmachtoub, Adam N., Paul Grigas. 2022. Smart ”predict, then optimize”. *Management Science*, 68 (1), 9-26.

Feng, Qi, J George Shanthikumar. 2023. The framework of parametric and nonparametric operational data analytics. *Production and Operations Management*, 32 (9), 2685-2703.

Feng, Qi, J George Shanthikumar, Jian Wu. 2025. Contextual data-integrated newsvendor solution with operational data analytics (ODA). *Management Science*, .

Ferreira, Kris Johnson, Bin Hong Alex Lee, David Simchi-Levi. 2016. Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & service operations management*, 18 (1), 69-88.

Folland, Gerald B. 1999. *Real analysis: modern techniques and their applications*. John Wiley & Sons.

Harsha, Pavithra, Ramesh Natarajan, Dharmashankar Subramanian. 2021. A prescriptive machine-learning framework to the price-setting newsvendor problem. *INFORMS Journal on Optimization*, 3 (3), 227-253.

Kallus, Nathan, Xiaojie Mao. 2022. Stochastic optimization forests. *Management Science*, 69 (4), 1975-1994.

Kannan, Rohit, Güzin Bayraksan, James R Luedtke. 2024. Residuals-based distributionally robust optimization with covariate information. *Mathematical Programming*, 207 (1), 369-425.

Kannan, Rohit, Güzin Bayraksan, James R Luedtke. 2025. Data-driven sample average approximation with covariate information. *Operations Research*, .

Lin, Shaochong, Youhua Chen, Yanzhi Li, Zuo-Jun Max Shen. 2022. Data-driven newsvendor problems regularized by a profit risk constraint. *Production and Operations Management*, 31 (4), 1630-1644.

Liu, Junyi, Guangyu Li, Suvrajeet Sen. 2021a. Coupled learning enabled stochastic programming with endogenous uncertainty. *Mathematics of Operations Research*, 47 (2), 1681-1705.

Liu, Maoqi, Qingchun Meng, Guodong Yu, Zhi-Hai Zhang. 2024. Fairness as a robust utilitarianism. *Production and Operations Management*, 10591478241262285.

Liu, Sheng, Long He, Zuo-Jun Max Shen. 2021b. On-time last-mile delivery: Order assignment with travel-time predictors. *Management Science*, 67 (7), 4095-4119.

Luo, Fengqiao, Sanjay Mehrotra. 2020. Distributionally robust optimization with decision dependent ambiguity sets. *Optimization Letters*, 14 (8), 2565-2594.

Mendler-Dünner, Celestine, Juan Perdomo, Tijana Zrnic, Moritz Hardt. 2020. Stochastic optimization for performative prediction. *International Conference on Machine Learning*, 7599-7609.

Noyan, Nilay, Gábor Rudolf, Miguel Lejeune. 2021. Distributionally robust optimization under a decision-dependent ambiguity set with applications to machine scheduling and humanitarian logistics. *INFORMS Journal on Computing*, 34 (2), 729-751.

Oroojlooyjadid, Afshin, Lawrence V Snyder, Martin Takáč. 2020. Applying deep learning to the newsvendor problem. *IISE Transactions*, 52 (4), 444-463.

Perdomo, Juan, Tijana Zrnic, Celestine Mendler-Dünner, Moritz Hardt. 2020. Performative prediction. *International Conference on Machine Learning*. PMLR, 7599-7609.

Rubner, Yossi, Carlo Tomasi, Leonidas J. Guibas. 2000. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40 (2), 99-121.

Sadana, Utsav, Abhilash Chenreddy, Erick Delage, Alexandre Forel, Emma Frejinger, Thibaut Vidal. 2025. A survey of contextual optimization methods for decision-making under uncertainty. *European Journal of Operational Research*, 320 (2), 271-289.

Sen, Suvrajeet, Yunxiao Deng. 2017. *Learning enabled optimization: Towards a fusion of statistical learning and stochastic optimization*. Humboldt-Universität zu Berlin.

Srivastava, Prateek R, Yijie Wang, Grani A Hanasusanto, Chin Pang Ho. 2021. On data-driven prescriptive analytics with side information: A regularized nadaraya-watson approach. *arXiv preprint arXiv:2110.04855*, .

Zhang, Yanfei, Junbin Gao. 2017. Assessing the performance of deep learning algorithms for newsvendor problem. *Neural Information Processing: 24th International Conference, ICONIP 2017, Guangzhou, China, November 14-18, 2017, Proceedings, Part I 24*. Springer, 912-921.

# E-Companion for Tackling Decision Dependency in Contextual Stochastic Optimization

## EC.1 Descriptions of Step Policies

In this section, we explain the solution methods adopted in the numerical experiment section.

### EC.1.1 Diminishing Step

The diminishing step adopts the size of $\eta^r$ such that $\eta^r > \eta^{r+1}$ and $\sum_{r=0}^{\infty} \eta^r = \infty$. A typical choice is $\eta^r = C/(r+1)$, where $C$ is a constant that can be adjusted to suit different problems.

### EC.1.2 Armijo Step

Let $f(\cdot) = \hat{g}(\cdot, \boldsymbol{z})$ denote the function we want to minimize, where $\boldsymbol{z}$ is the fixed covariate. The Armijo principle chooses the step size $\eta^r$ by the following steps (we denote the descent direction as $\boldsymbol{d}^r$) in Algorithm 2

**Algorithm 2** Armijo step size

```
Input: iteration solution $\boldsymbol{x}^r$, contextual information $z$, $\alpha_0$, $\beta \in (0,1)$, $\sigma \in [0,1)$, tolerance $\varepsilon$.
Output: step size $\eta^r$.
$\eta^r = \alpha_0$
$\boldsymbol{x}^{r+1} = \boldsymbol{x}^r + \eta^r \boldsymbol{d}^r$;
while $\eta^r \geq \varepsilon$ and $f(\boldsymbol{x}^r) - f(\boldsymbol{x}^{r+1}) < \sigma\eta^r(\hat{G}_N(\boldsymbol{x}^r;\boldsymbol{z}))^T \boldsymbol{d}^r$ do
    $\eta^r = \eta^r * \beta$;
    $\boldsymbol{x}^{r+1} = \boldsymbol{x}^r + \eta^r \boldsymbol{d}^r$;
end while
return $\eta^r$
```

Note that the hyperparameter $\sigma$ can be 0 in our problem. When $\sigma = 0$, Armijo step size ensures that the objective function is lowered in an approximate context. We also show the special meaning of $\sigma = 0$ in Proposition 4.

It's also notable that, under Assumptions 1, 4(a) and 4(b), and when the requirements for Lemma 1 hold, $\hat{g}(\boldsymbol{x}, \boldsymbol{z}) = g(\boldsymbol{x}, \boldsymbol{z})$ when $N \to \infty$ for $\boldsymbol{x} \in \mathcal{X}$ almost everywhere. Therefore, in the determination of Armijo step size, we directly replace $g(\boldsymbol{x}, \boldsymbol{z})$ with $\hat{g}(\boldsymbol{x}, \boldsymbol{z})$.

## EC.2 Proofs

The proofs are supported by the following lemmas in previous literature, which show how Assumption 2 affects the distance between expectations of different distributions.

LEMMA EC.1. *(Kantorovich-Rubinstein) For all functions q that are* 1*-Lipschitz*

$$\|\mathbb{E}_{d\sim D(p)}[q(d)] - \mathbb{E}_{d\sim D(p')}[q(d)]\| \le W_1(D(p), D(p')).$$

And for function $q'$ that is $L_q$-Lipshitz,

$$\|\mathbb{E}_{d\sim D(p)}[q'(d)] - \mathbb{E}_{d\sim D(p')}[q'(d)]\| = L_q\|\mathbb{E}_{d\sim D(p)}\left[\frac{q'(d)}{L_q}\right] - \mathbb{E}_{d\sim D(p')}\left[\frac{q'(d)}{L_q}\right]\| \le L_q W_1(D(p), D(p')),$$

since $q'(d)/L_q$ is 1-Lipschitz.

### EC.2.1 Proof of Lemma 1

The proof Lemma 1 roughly follows the proof of Theorem EC.9 in Bertsimas and Kallus (2019). The only difference is that Lemma 1 is about the convergence of the derivative function rather than the objective function.

Specifically, for every $\boldsymbol{x}$, the marginal distribution of $\boldsymbol{y} \sim f(y;\boldsymbol{x},z)$ is independent of $\boldsymbol{y}$ conditioned on $z$, the assumption of ignoreability is satisfied. Furthermore, the feasible region for $\boldsymbol{x}$ is non-empty, and we only restrict the up-and-down limit of the two decisions.

Therefore, we need to prove that the expected gradient $\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x},\boldsymbol{y})]$ is bounded and equicontinuous on $\boldsymbol{x}$. First, from Assumption 4(a), we have $|\nabla_{\boldsymbol{x}} l(\boldsymbol{x},\boldsymbol{y})| < L_1$ for every $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{y} \in Y$, and thus $\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x},\boldsymbol{y})] < L_1$ is bounded. Then from Assumption 4(c), for any $\boldsymbol{x} \in X, \varepsilon' > 0, \exists \delta = \varepsilon'/L_1^c, s.t.$ when $\|\boldsymbol{x} - \boldsymbol{x}'\| \le \delta$,

$$\|\nabla_{\boldsymbol{x}} l(\boldsymbol{x},\boldsymbol{y}) - \nabla_{\boldsymbol{x}} l(\boldsymbol{x}',\boldsymbol{y})\| \le L_1^c\|\boldsymbol{x}' - \boldsymbol{x}\| \le \varepsilon'.$$

Thus $\nabla_{\boldsymbol{x}} l(\boldsymbol{x},\boldsymbol{y})$ is equicontinuous. Then the proof is completed by Theorem EC.9 in Bertsimas and Kallus (2019).

### EC.2.2 Proof of Theorem 1

We prove the theorem by contradiction. Suppose $\boldsymbol{x}^*$ minimizes $g(\boldsymbol{x}) = \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x},z)}[l(\boldsymbol{x},\boldsymbol{y})]$, and $\|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^*,z)}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^*,\boldsymbol{y})]\| > L_1\varepsilon$. Then for any $\boldsymbol{x}_1 \in \mathcal{X}$,

$$\begin{aligned} g(\boldsymbol{x}_1) - g(\boldsymbol{x}^*) &= \left(\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}_1,z)}[l(\boldsymbol{x}_1,\boldsymbol{y})] - \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,z)}[l(\boldsymbol{x}^*,\boldsymbol{y})]\right) \\ &= \left(\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}_1,z)}[l(\boldsymbol{x}_1,\boldsymbol{y})] - \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,z)}[l(\boldsymbol{x}_1,\boldsymbol{y})]\right) \\ &\quad + \left(\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,z)}[l(\boldsymbol{x}_1,\boldsymbol{y})] - \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,z)}[l(\boldsymbol{x}^*,\boldsymbol{y})]\right). \end{aligned}$$

From Lemma EC.1 and Assumption 2, we have

$$\begin{aligned}
&\left(\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}_1,\boldsymbol{z})}[l(\boldsymbol{x}_1,\boldsymbol{y})] - \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,\boldsymbol{z})}[l(\boldsymbol{x}_1,\boldsymbol{y})]\right)\\
&\leq |\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}_1,\boldsymbol{z})}[l(\boldsymbol{x}_1,\boldsymbol{y})] - \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,\boldsymbol{z})}[l(\boldsymbol{x}_1,\boldsymbol{y})]|\\
&\leq L_1 W_1(f(\boldsymbol{y};\boldsymbol{x}_1,\boldsymbol{z}), f(\boldsymbol{y};\boldsymbol{x}^*,\boldsymbol{z}))\\
&\leq L_1\varepsilon\|\boldsymbol{x}_1-\boldsymbol{x}^*\|.
\end{aligned}$$

For the second term, we expand $l(\boldsymbol{x}_1,\boldsymbol{y})$ at $\boldsymbol{x}^*$ and obtain

$$\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,\boldsymbol{z})}[l(\boldsymbol{x}_1,\boldsymbol{y})] - \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,\boldsymbol{z})}[l(\boldsymbol{x}^*,\boldsymbol{y})] = \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^*,\boldsymbol{y})]^T(\boldsymbol{x}_1-\boldsymbol{x}^*) + o(\|\boldsymbol{x}_1-\boldsymbol{x}^*\|),$$

where $o(\|\boldsymbol{x}_1-\boldsymbol{x}^*\|)$ denotes the first-order infinitesimals to $\|\boldsymbol{x}_1-\boldsymbol{x}^*\|$. By substituting the two terms above and dividing both sides by $\|\boldsymbol{x}_1-\boldsymbol{x}^*\|$, we obtain

$$\frac{g(\boldsymbol{x}_1)-g(\boldsymbol{x}^*)}{\|\boldsymbol{x}_1-\boldsymbol{x}^*\|} \leq -L_1\varepsilon + \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^*,\boldsymbol{y})]^T \frac{(\boldsymbol{x}_1-\boldsymbol{x}^*)}{\|\boldsymbol{x}_1-\boldsymbol{x}^*\|} + \frac{o(\|\boldsymbol{x}_1-\boldsymbol{x}^*\|)}{\|\boldsymbol{x}_1-\boldsymbol{x}^*\|}.$$

We let $\boldsymbol{x}_1-\boldsymbol{x}^*$ take the opposite direction as $\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^*,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^*,\boldsymbol{y})]$, which makes the second term on the right side become $-\|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^*,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^*,\boldsymbol{y})]\|$.

$$\frac{g(\boldsymbol{x}_1)-g(\boldsymbol{x}^*)}{\|\boldsymbol{x}_1-\boldsymbol{x}^*\|} \leq L_1\varepsilon - \|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^*,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^*,\boldsymbol{y})]\| + \frac{o(\|\boldsymbol{x}_1-\boldsymbol{x}^*\|)}{\|\boldsymbol{x}_1-\boldsymbol{x}^*\|}.$$

Since $\|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^*,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^*,\boldsymbol{y})]\| > L_1\varepsilon$, there exists $\boldsymbol{x}_1$ that is sufficiently close to $\boldsymbol{x}^*$ such that $g(\boldsymbol{x}_1)-g(\boldsymbol{x}^*)<0$, which contradicts the condition that $g(\boldsymbol{x}^*)$ is the optimal solution.

### EC.2.3 Proof of Theorem 2

We analyze the error of $\boldsymbol{x}_N^{k+1}$. Since $\mathcal{X}$ is closed and convex:

$$\begin{aligned}
\|\boldsymbol{x}_N^{k+1}-\boldsymbol{x}^*\|^2 =&\|\Pi_{\mathcal{X}}(\boldsymbol{x}_N^k-\eta^k\hat{G}_N(\boldsymbol{x}_N^k,\boldsymbol{z}))-\boldsymbol{x}^*\|^2\\
\leq&\|\boldsymbol{x}_N^k-\eta^k\hat{G}_N(\boldsymbol{x}_N^k,\boldsymbol{z})-\boldsymbol{x}^*\|^2\\
=&\|\boldsymbol{x}_N^k-\boldsymbol{x}^*\|^2-2\eta^k\hat{G}_N(\boldsymbol{x}_N^k,\boldsymbol{z})^T(\boldsymbol{x}_N^k-\boldsymbol{x}^*)+(\eta^k)^2\|\hat{G}_N(\boldsymbol{x}_N^k,\boldsymbol{z})\|^2.
\end{aligned}$$

Let $N\to+\infty$ for both sides, denote $\lim_{N\to\infty}\boldsymbol{x}_N^k$ as $\boldsymbol{x}^k$ for simplicity. From Lemma 1, we have $\lim_{N\to\infty}\hat{G}_N(\boldsymbol{x},\boldsymbol{z}) = \mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x},\boldsymbol{y})]$ for any $\boldsymbol{x}$. Therefore, for any $\zeta_1,\zeta_2>0,\boldsymbol{x},\boldsymbol{z}$ and vector $\boldsymbol{v}$, there exists $N_0$ such that $\forall N>N_0$,

$$\|\hat{G}_N(\boldsymbol{x},\boldsymbol{z})\|^2 \leq \|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x},\boldsymbol{y})]\|^2+\zeta_1^2,$$

and

$$|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x},\boldsymbol{y})]^T\boldsymbol{v}-\hat{G}_N(\boldsymbol{x},\boldsymbol{z})^T\boldsymbol{v}| \leq \zeta_2\|\boldsymbol{v}\|,$$

we then have,

$$G_N(\boldsymbol{x},\boldsymbol{z})^T\boldsymbol{v} \geq \mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x},\boldsymbol{y})]^T\boldsymbol{v}-\zeta_2\|\boldsymbol{v}\|.$$

Thus, for any $\zeta > 0$, we let $\zeta_1 = \sqrt{\zeta/\eta^k}$ and $\zeta_2 = \frac{\zeta}{\|\boldsymbol{x}_N^k - \boldsymbol{x}^*\|}$. Then, $\exists N_0$, for any fix $N > N_0$ we have:

$$\begin{aligned}\|\boldsymbol{x}_N^{k+1} - \boldsymbol{x}^*\|^2 \leq & \|\boldsymbol{x}_N^k - \boldsymbol{x}^*\|^2 - 2\eta^k \mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^k,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_N^k, \boldsymbol{y})]^T(\boldsymbol{x}_N^k - \boldsymbol{x}^*) \\ & + (\eta^k)^2 \|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^k,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_N^k, \boldsymbol{y})]\|^2 + 3\eta^k \zeta.\end{aligned}$$

We bound the second term by convexity of the cost function

$$\begin{aligned}\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^k,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_N^k, \boldsymbol{y})]^T(\boldsymbol{x}_N^k - \boldsymbol{x}^*) = & \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}_N^k,\boldsymbol{z})}[\nabla l(\boldsymbol{x}_N^k, \boldsymbol{y})^T(\boldsymbol{x}_N^k - \boldsymbol{x}^*)] \\ \geq & \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}_N^k,\boldsymbol{z})}[l(\boldsymbol{x}_N^k, \boldsymbol{y}) - l(\boldsymbol{x}^*, \boldsymbol{y})].\end{aligned}$$

For the third term, we bound it by Assumption 4(e).

$$\|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^k,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_N^k, \boldsymbol{y})]\|^2 \leq (L_3^c)^2.$$

Thus, we have

$$\|\boldsymbol{x}_N^{k+1} - \boldsymbol{x}^*\|^2 \leq \|\boldsymbol{x}_N^k - \boldsymbol{x}^*\|^2 - 2\eta^k \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}_N^k,\boldsymbol{z})}[l(\boldsymbol{x}_N^k, \boldsymbol{y}) - l(\boldsymbol{x}^*, \boldsymbol{y})] + (\eta^k)^2 (L_3^c)^2 + 3\eta^k \zeta.$$

And consequently, we have

$$2\eta^k \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}_N^k,\boldsymbol{z})}[l(\boldsymbol{x}_N^k, \boldsymbol{y}) - l(\boldsymbol{x}^*, \boldsymbol{y})] \leq -\|\boldsymbol{x}_N^{k+1} - \boldsymbol{x}^*\|^2 + \|\boldsymbol{x}_N^k - \boldsymbol{x}^*\|^2 + (\eta^k)^2 (L_3^c)^2 + 3\eta^k \zeta.$$

For the left-hand side, we have

$$\begin{aligned}\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}_N^k,\boldsymbol{z})}[l(\boldsymbol{x}_N^k, \boldsymbol{y}) - l(\boldsymbol{x}^*, \boldsymbol{y})] = & \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}_N^k,\boldsymbol{z})}[l(\boldsymbol{x}_N^k, \boldsymbol{y})] - \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,\boldsymbol{z})}[l(\boldsymbol{x}^*, \boldsymbol{y})] \\ & + \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,\boldsymbol{z})}[l(\boldsymbol{x}^*, \boldsymbol{y})] - \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}_N^k,\boldsymbol{z})}[l(\boldsymbol{x}^*, \boldsymbol{y})] \\ \geq & -\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,\boldsymbol{z})}[l(\boldsymbol{x}^*, \boldsymbol{y})] + \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}_N^k,\boldsymbol{z})}[l(\boldsymbol{x}_N^k, \boldsymbol{y})] \\ & - \left|\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}_N^k,\boldsymbol{z})}[l(\boldsymbol{x}^*, \boldsymbol{y})] - \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,\boldsymbol{z})}[l(\boldsymbol{x}^*, \boldsymbol{y})]\right| \\ \geq & -\varepsilon L_2 \|\boldsymbol{x}^* - \boldsymbol{x}_N^k\| - \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,\boldsymbol{z})}[l(\boldsymbol{x}^*, \boldsymbol{y})] + \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}_N^k,\boldsymbol{z})}[l(\boldsymbol{x}_N^k, \boldsymbol{y})].\end{aligned}$$

The second equality holds because $|\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}_N^k,\boldsymbol{z})}[l(\boldsymbol{x}^*, \boldsymbol{y})] - \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,\boldsymbol{z})}[l(\boldsymbol{x}^*, \boldsymbol{y})]| \geq \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,\boldsymbol{z})}[l(\boldsymbol{x}^*, \boldsymbol{y})] - \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}_N^k,\boldsymbol{z})}[l(\boldsymbol{x}_N^k, \boldsymbol{y})]$ while the third one holds due to Lemma EC.1. Then, we substitute the inequality and have

$$\begin{aligned}2\eta^k(\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}_N^k,\boldsymbol{z})}[l(\boldsymbol{x}_N^k, \boldsymbol{y})] - \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,\boldsymbol{z})}[l(\boldsymbol{x}^*, \boldsymbol{y})]) \leq & 2\eta^k \varepsilon L_2 \|\boldsymbol{x}^* - \boldsymbol{x}_N^k\| - \|\boldsymbol{x}_N^{k+1} - \boldsymbol{x}^*\|^2 \\ & + \|\boldsymbol{x}_N^k - \boldsymbol{x}^*\|^2 + (\eta^k L_3^c)^2 + 3\eta^k \zeta.\end{aligned}$$

Taking the summation from $r = 0$ to $k$, we obtain

$$\begin{aligned}2\sum_{r=0}^{k} \eta^r \{\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^r,\boldsymbol{z})}[l(\boldsymbol{x}^r, \boldsymbol{y})] - \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,\boldsymbol{z})}[l(\boldsymbol{x}^*, \boldsymbol{y})]\} \leq & 2\varepsilon L_2 \sum_{r=0}^{k} \eta^r \|\boldsymbol{x}^* - \boldsymbol{x}_N^r\| - \|\boldsymbol{x}_N^{k+1} - \boldsymbol{x}^*\|^2 \\ & + \|\boldsymbol{x}^0 - \boldsymbol{x}_N^k\|^2 + (L_3^c)^2 \sum_{r=0}^{k} (\eta^r)^2 + 3\sum_{r=0}^{k} \eta^r \zeta \\ \leq & 2\varepsilon L_2 \sum_{r=0}^{k} \eta^r \|\boldsymbol{x}^* - \boldsymbol{x}_N^r\| \\ & + \|\boldsymbol{x}^0 - \boldsymbol{x}_N^k\|^2 + (L_3^c)^2 \sum_{r=0}^{k} (\eta^r)^2 + 3\sum_{r=0}^{k} \eta^r \zeta\end{aligned}$$

Since $\min_{0\le r'\le k}\{\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^{r'},\boldsymbol{z})}[l(\boldsymbol{x}^{r'},\boldsymbol{y})]-\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,\boldsymbol{z})}[l(\boldsymbol{x}^*,\boldsymbol{y})]\}\le\{\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^{r},\boldsymbol{z})}[l(\boldsymbol{x}^{r},\boldsymbol{y})]-\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,\boldsymbol{z})}[l(\boldsymbol{x}^*,\boldsymbol{y})]\},\forall r\in[K]$, we have

$$\begin{aligned}(2\sum_{r=0}^{k}\eta^r)\min_{0\le r\le k}\{\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^{r},\boldsymbol{z})}[l(\boldsymbol{x}^{r},\boldsymbol{y})]-\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^*,\boldsymbol{z})}[l(\boldsymbol{x}^*,\boldsymbol{y})]\}\le&2\varepsilon L_2\sum_{r=0}^{k}\eta^r\|\boldsymbol{x}^*-\boldsymbol{x}_N^r\|\\&+\|\boldsymbol{x}^0-\boldsymbol{x}_N^k\|^2+(L_3^c)^2\sum_{r=0}^{k}(\eta^r)^2+3\sum_{r=0}^{k}\eta^r\zeta.\end{aligned}$$

Hence we complete the proof by dividing $2\sum_{r=0}^{k}\eta^r$ on both sides.

### EC.2.4 Proof of Proposition 1

We investigate the distance between $\boldsymbol{x}_N^k$ and a stable point $\boldsymbol{x}_{PS}$.

$$\begin{aligned}\|\boldsymbol{x}_N^{k+1}-\boldsymbol{x}_{PS}\|^2=&\|\Pi_X(\boldsymbol{x}_N^k-\eta\hat{G}_N(\boldsymbol{x}_N^k;\boldsymbol{z}))-\boldsymbol{x}_{PS}\|^2\\\le&\|\boldsymbol{x}_N^k-\eta\hat{G}_N(\boldsymbol{x}_N^k;\boldsymbol{z})-\boldsymbol{x}_{PS}\|^2\\=&\|\boldsymbol{x}_N^k-\boldsymbol{x}_{PS}\|^2-2\eta\hat{G}_N(\boldsymbol{x}_N^k)^T(\boldsymbol{x}_N^k-\boldsymbol{x}_{PS})+(\eta^2)\|\hat{G}_N(\boldsymbol{x}_N^k)\|^2.\end{aligned}$$

We begin by upper bounding the second term. From Lemma 1, we know that for any $\zeta>0$, there exists a sample size $N_0$ such that $\|\hat{G}_N(\boldsymbol{x})-\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})}[\nabla_{\boldsymbol{x}}l(\boldsymbol{x},\boldsymbol{y})]\|\le\zeta,\forall N>N_0$ for $\boldsymbol{x}$ almost everywhere. Thus, we have

$$\|-\hat{G}_N(\boldsymbol{x}_N^k)+\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^k,\boldsymbol{z})}[\nabla_{\boldsymbol{x}}l(\boldsymbol{x}_N^k,\boldsymbol{y})]\|\|\boldsymbol{x}_N^k-\boldsymbol{x}_{PS}\|\le\zeta\|\boldsymbol{x}_N^k-\boldsymbol{x}_{PS}\|$$

and according to Cauchy–Schwarz inequality, we have

$$(-\hat{G}_N(\boldsymbol{x}_N^k)+\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^k,\boldsymbol{z})}[\nabla_{\boldsymbol{x}}l(\boldsymbol{x}_N^k,\boldsymbol{y})])^T(\boldsymbol{x}_N^k-\boldsymbol{x}_{PS})\le\|-\hat{G}_N(\boldsymbol{x}_N^k)+\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^k,\boldsymbol{z})}[\nabla_{\boldsymbol{x}}l(\boldsymbol{x}_N^k,\boldsymbol{y})]\|\|\boldsymbol{x}_N^k-\boldsymbol{x}_{PS}\|$$

Therefore, we have

$$\hat{G}_N(\boldsymbol{x}_N^k)^T(\boldsymbol{x}_N^k-\boldsymbol{x}_{PS})\ge\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^k,\boldsymbol{z})}[\nabla_{\boldsymbol{x}}l(\boldsymbol{x}_N^k,\boldsymbol{y})]^T(\boldsymbol{x}_N^k-\boldsymbol{x}_{PS})-\zeta\|\boldsymbol{x}_N^k-\boldsymbol{x}_{PS}\|.\tag{EC.1}$$

We can further bound the second term using the same approach as the proof of Proposition 2.3 in Perdomo et al. (2020) where they give that

$$\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^k,\boldsymbol{z})}[\nabla_{\boldsymbol{x}}l(\boldsymbol{x}_N^k,\boldsymbol{y})]^T(\boldsymbol{x}_N^k-\boldsymbol{x}_{PS})\ge A\|\boldsymbol{x}_N^k-\boldsymbol{x}_{PS}\|^2.$$

where $A=\gamma-\varepsilon L_1^c$.

Therefore, we have

$$\hat{G}_N(\boldsymbol{x}_N^k)^T(\boldsymbol{x}_N^k-\boldsymbol{x}_{PS})\ge A\|\boldsymbol{x}_N^k-\boldsymbol{x}_{PS}\|^2-\zeta\|\boldsymbol{x}_N^k-\boldsymbol{x}_{PS}\|.$$

We then bound the third term. Based on Lemma 1, we have

$$\|\hat{G}_N(\boldsymbol{x}_N^k)-\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^k,\boldsymbol{z})}[\nabla_{\boldsymbol{x}}l(\boldsymbol{x}_N^k,\boldsymbol{y})]\|\le\zeta.$$

Moreover, according to triangular inequality, we have

$$\|\hat{G}_N(\boldsymbol{x}_N^k)\| - \|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^k,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_N^k,\boldsymbol{y})]\| \leq \left| \|\hat{G}_N(\boldsymbol{x}_N^k)\| - \|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^k,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_N^k,\boldsymbol{y})]\| \right| \leq \|\hat{G}_N(\boldsymbol{x}_N^k) - \mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^k,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_N^k,\boldsymbol{y})]\|,$$

The first inequality holds because the absolute value of the expression is always larger than or equal to itself. Accordingly, we have

$$\|\hat{G}_N(\boldsymbol{x}_N^k)\| \leq \zeta + \|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^k,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_N^k,\boldsymbol{y})]\|.$$

Proposition 2.3 in Perdomo et al. (2020) gives that under Assumptions 2 and 4, we have

$$\|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^k,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_N^k,\boldsymbol{y})]\| \leq \sqrt{2}B\|\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}\|.$$

where $B = L_1^c\sqrt{1+\varepsilon^2}$. As such,

$$\|\hat{G}_N(\boldsymbol{x}_N^k)\|^2 \leq (\zeta + \sqrt{2}B\|\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}\|)^2. \tag{EC.2}$$

Based on the derived inequalities EC.1 and EC.2, we obtain

$$\|\boldsymbol{x}_N^{k+1} - \boldsymbol{x}_{PS}\|^2 \leq (1 - 2\eta A + 2\eta^2 B^2)\|\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}\|^2 + (2\eta\zeta + 2\sqrt{2}\zeta\eta^2 B)\|\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}\| + \zeta^2\eta^2. \tag{EC.3}$$

Denote $\|\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}\| = X_k \geq 0$, we have

$$X_{k+1}^2 \leq (1 - 2\eta A + 2\eta^2 B^2)X_k^2 + (2\eta\zeta + 2\sqrt{2}\zeta\eta^2 B)X_k + \zeta^2\eta^2 \tag{EC.4}$$

Denote $h(X) = (1 - 2\eta A + 2\eta^2 B^2)X^2 + (2\eta\zeta + 2\sqrt{2}\zeta\eta^2 B)X + \zeta^2\eta^2$, i.e., $X_{k+1}^2 \leq h(X_K)$. We can easily get $h(X)$ is monotonically increasing in $[0, +\infty)$ as it is a quadratic function with $1 - 2\eta A + 2\eta^2 B^2 > 0$, and $\eta\zeta + 2\sqrt{2}\zeta\eta^2 B > 0$.

Let $g(X) = h(X) - X^2 = (-2\eta A + 2\eta^2 B^2)X^2 + (2\eta\zeta + 2\sqrt{2}\zeta\eta^2 B)X + \zeta^2\eta^2$. Now that $1 - 2\eta A + 2\eta^2 B^2 < 1$, we have $-2\eta A + 2\eta^2 B^2 < 0$. $g(X)$ as a quadratic function has two solutions:

$$\begin{aligned} X_a &= \frac{\zeta}{2(A - \eta B^2)}(1 + \sqrt{2}\eta B - \sqrt{-2B^2\eta^2 + 2A\eta + (1 + \sqrt{2}\eta B)^2}) < 0, \\ X_b &= \frac{\zeta}{2(A - \eta B^2)}(1 + \sqrt{2}\eta B + \sqrt{-2B^2\eta^2 + 2A\eta + (1 + \sqrt{2}\eta B)^2}) > 0 \end{aligned} \tag{EC.5}$$

When $X \in [0, X_b), g(X) > 0$; When $X \in (X_b, +\infty), g(X) < 0$.

Then the series $\{X_k\}$ can be separated into two cases:

**Case 1.** There exists a certain $K_1$, $X_{K_1} \leq X_b$:

$$X_{K_1+1}^2 \leq h(X_{K_1}) \leq h(X_b) = g(X_b) + (X_b)^2 = (X_b)^2$$

Namely, $X_{K_1+1} \leq X_b$. By deduction, $\forall k > K_1, X_k \leq X_b < X_b + \varepsilon$.

**Case 2.** For all $k > 0$, $X_k > X_b$: In this case, $X_{k+1}^2 - X_K^2 \leq g(X_K) < 0$. That is, series $\{X_k\}$ is monotonically decreasing. Since series $\{X_k\}$ has a lower bound $X_b$, it has an infimum. Denote $\inf_{k \in N_+} X_k = U \geq X_b$. There are two sub cases under Case 2.

- **Case 2.1** $U = X_b$: From definition of infimum, $\forall \varepsilon > 0$, $\exists K_2 > 0$, $X_{K_2} \leq U + \varepsilon$. And thus $\forall k > K_2, X_k < X_{K_2} \leq X_b + \varepsilon$.
- **Case 2.2** $U > X_b$: From definition of infimum, $\forall \varepsilon > 0$, $\exists K_3 > 0$, $X_{K_3}^2 \leq U^2 + \varepsilon$.

$$X_{K_3+1}^2 \leq h(X_{K_3}) \leq h(\sqrt{U^2+\varepsilon}) = g(\sqrt{U^2+\varepsilon}) + U^2 + \varepsilon$$

Let $K(\varepsilon) = g(\sqrt{U^2+\varepsilon}) + U^2 + \varepsilon$, $K(\varepsilon)$ is obviously continuous in $\varepsilon$.

$$\lim_{\varepsilon \to 0} K(\varepsilon) = K(0) = g(U) + U^2 < U^2 (\forall X > X_b, g(X) < 0)$$

And thus there exists $\delta > 0$, $\forall 0 < \varepsilon < \delta, K(\varepsilon) < U^2$. Namely there exists $\varepsilon > 0$, $X_{K_3+1}^2 \leq h(X_{k_3}) < K(\varepsilon) < U^2$. But $X_{K_3+1} < U$ contradicts the case assumption that $\forall X_k, X_k \geq U > X_b$. Hence this case doesn't exist.

As a conclusion, $\exists K_0 = \max\{K_1, K_2\} > 0$, $\forall k > K_0, X_k < X_b + \varepsilon$ and the proof is completed.

### EC.2.5 Proof of Proposition 2

To prove that $||\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}||$ decreases exponentially, we only need to prove that when $||\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}|| > 2UB(\eta, \zeta)$, we have $||\boldsymbol{x}_N^{k+1} - \boldsymbol{x}_{PS}|| \leq C||\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}||$. Namely, $||\boldsymbol{x}_N^{k+1} - \boldsymbol{x}_{PS}||^2 \leq C^2||\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}||^2$.

From the proof of Proposition 1 we have, (EC.3)

$$||\boldsymbol{x}_N^{k+1} - \boldsymbol{x}_{PS}||^2 \leq (1 - 2\eta A + 2\eta^2 B^2)||\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}||^2 + (2\eta\zeta + 2\sqrt{2}\zeta\eta^2 B)||\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}|| + \zeta^2\eta^2.$$

Let

$$g(X) = (-\eta A + \eta^2 B^2)X^2 + (2\eta\zeta + 2\sqrt{2}\zeta\eta^2 B)X + \zeta^2\eta^2. \tag{EC.6}$$

Then, we have

$$||\boldsymbol{x}_N^{k+1} - \boldsymbol{x}_{PS}||^2 \leq (1 - \eta A + \eta^2 B^2)||\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}||^2 + g(||\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}||).$$

The quadratic function on the right side of (EC.7) also has two roots, one positive and the other negative. The positive root equals to

$$X_b' = \frac{\zeta}{A - \eta B^2}(1 + \sqrt{2}\eta B + \sqrt{-B^2\eta^2 + A\eta + (1 + \sqrt{2}\eta B)^2}), \tag{EC.7}$$

Since

$$2UB(\eta, \zeta) = \frac{\zeta}{A - \eta B^2}(1 + \sqrt{2}\eta B + \sqrt{-2B^2\eta^2 + 2A\eta + (1 + \sqrt{2}\eta B)^2}), \tag{EC.8}$$

$2UB(\eta, \zeta) \geq X_b'$. As such, when$||\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}|| \geq 2UB(\eta, \zeta)$, we have $g(||\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}||) \leq 0$ and therefore,

$$||\boldsymbol{x}_N^{k+1} - \boldsymbol{x}_{PS}||^2 \leq (1 - \eta A + \eta^2 B^2)||\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}||^2 + g(||\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}||) \leq (1 - \eta A + \eta^2 B^2)||\boldsymbol{x}_N^k - \boldsymbol{x}_{PS}||^2.$$

Defining $C = 1 - \eta A + \eta^2 B^2$ complete the proof.

### EC.2.6 Proof of Theorem 3

Since $l(\boldsymbol{x},\boldsymbol{y})$ is strongly convex in $\boldsymbol{x}$ and $L_2-$Lipschitz continuous in $\boldsymbol{y}$, the proof is then complete by imposing the triangular inequality to Lemma 2 and Proposition 1.

### EC.2.7 Proof of Proposition 3

The proof is divided into two steps. In the first step, we prove that the objective function $\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x},\boldsymbol{z})}[l(\boldsymbol{x},\boldsymbol{y})]$ has Lipschitz gradient in $\boldsymbol{x}$. Then we prove that under diminishing step, any converging subsequence converge to the stationary point.

We denote $g(\boldsymbol{x}) = \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x},\boldsymbol{z})}[l(\boldsymbol{x},\boldsymbol{y})]$, then for any $\boldsymbol{x}_1,\boldsymbol{x}_2 \in \mathcal{X}$

$$||\nabla_{\boldsymbol{x}} g(\boldsymbol{x}_1) - \nabla_{\boldsymbol{x}} g(\boldsymbol{x}_2)|| = ||\nabla_x \mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_1,\boldsymbol{z})}[l(\boldsymbol{x}_1,\boldsymbol{y})] - \nabla_x \mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_2,\boldsymbol{z})}[l(\boldsymbol{x}_2,\boldsymbol{y})]||.$$

According to Assumptions 4(a) and (c), the assumption for the Lebesgue Dominated Convergence Theorem is naturally satisfied, which requires $l(\boldsymbol{x},\boldsymbol{y})$ is differentiable in $\mathcal{X}$, and there exists $L^1(\boldsymbol{y})$ function $h(\boldsymbol{y})$, $|h(\boldsymbol{y})| \geq n|l(\boldsymbol{x}+\frac{1}{n},\boldsymbol{y}) - l(\boldsymbol{x},\boldsymbol{y})|$ for all $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{y} \in \Omega$. It implies that we can change the order of integration and differentiation when calculating the derivative of the integral of $l(\boldsymbol{x},\boldsymbol{y})$ (see Theorem 2.27 in Folland (1999)). That is,

$$\nabla_{\boldsymbol{x}} \int l(\boldsymbol{x},\boldsymbol{y}) f(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z}) d\boldsymbol{y} = \int \nabla_{\boldsymbol{x}} (l(\boldsymbol{x},\boldsymbol{y}) f(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z})) d\boldsymbol{y},$$

which enables us to access the derivative of the objective function. Then we have,

$$\begin{aligned}
||\nabla_{\boldsymbol{x}} g(\boldsymbol{x}_1) - \nabla_{\boldsymbol{x}} g(\boldsymbol{x}_2)|| =& || \int_{\boldsymbol{y}\in\Omega} \nabla_{\boldsymbol{x}} (l(\boldsymbol{x}_1,\boldsymbol{y}) f(\boldsymbol{y};\boldsymbol{x}_1,\boldsymbol{z})) d\boldsymbol{y} - \int_{\boldsymbol{y}\in\Omega} \nabla_{\boldsymbol{x}} (l(\boldsymbol{x}_2,\boldsymbol{y}) f(\boldsymbol{y};\boldsymbol{x}_2,\boldsymbol{z})) d\boldsymbol{y} || \\
\leq& || \int_{\boldsymbol{y}} l(\boldsymbol{x}_1,\boldsymbol{y}) (\nabla_{\boldsymbol{x}} f(\boldsymbol{y};\boldsymbol{x}_1,\boldsymbol{z})) - l(\boldsymbol{x}_2,\boldsymbol{y}) (\nabla_{\boldsymbol{x}} f(\boldsymbol{y};\boldsymbol{x}_2,\boldsymbol{z})) d\boldsymbol{y} || \\
&+ || \int_{\boldsymbol{y}} (\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_1,\boldsymbol{y})) f(\boldsymbol{y};\boldsymbol{x}_1,\boldsymbol{z}) - (\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_2,\boldsymbol{y})) f(\boldsymbol{y};\boldsymbol{x}_2,\boldsymbol{z}) d\boldsymbol{y} ||
\end{aligned}$$

The second inequality follows by the multiplication rule of the derivative. Define

$$\begin{aligned}
I =& || \int_{\boldsymbol{y}} l(\boldsymbol{x}_1,\boldsymbol{y}) (\nabla_{\boldsymbol{x}} f(\boldsymbol{y};\boldsymbol{x}_1,\boldsymbol{z})) - l(\boldsymbol{x}_2,\boldsymbol{y}) (\nabla_{\boldsymbol{x}} f(\boldsymbol{y};\boldsymbol{x}_2,\boldsymbol{z})) d\boldsymbol{y} ||, \\
II =& || \int_{\boldsymbol{y}} (\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_1,\boldsymbol{y})) f(\boldsymbol{y};\boldsymbol{x}_1,\boldsymbol{z}) - (\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_2,\boldsymbol{y})) f(\boldsymbol{y};\boldsymbol{x}_2,\boldsymbol{z}) d\boldsymbol{y} ||,
\end{aligned}$$

which refers to the first and second norms in the last inequality, respectively.

We then analyze $I$ and $II$.

$$\begin{aligned}
I \leq& || \int_{\boldsymbol{y}} l(\boldsymbol{x}_1,\boldsymbol{y}) \nabla_{\boldsymbol{x}} f(\boldsymbol{y};\boldsymbol{x}_1,\boldsymbol{z}) d\boldsymbol{y} - \int_{\boldsymbol{y}} l(\boldsymbol{x}_1,\boldsymbol{y}) \nabla_{\boldsymbol{x}} f(\boldsymbol{y};\boldsymbol{x}_2,\boldsymbol{z}) d\boldsymbol{y} || \\
&+ || \int_{\boldsymbol{y}} l(\boldsymbol{x}_1,\boldsymbol{y}) \nabla_{\boldsymbol{x}} f(\boldsymbol{y};\boldsymbol{x}_2,\boldsymbol{z}) d\boldsymbol{y} - \int_{\boldsymbol{y}} l(\boldsymbol{x}_2,\boldsymbol{y}) \nabla_{\boldsymbol{x}} f(\boldsymbol{y};\boldsymbol{x}_2,\boldsymbol{z}) d\boldsymbol{y} || \\
\leq& \int_{\boldsymbol{y}} |l(\boldsymbol{x}_1,\boldsymbol{y})| ||\nabla_{\boldsymbol{x}} f(\boldsymbol{y};\boldsymbol{x}_1,\boldsymbol{z}) - \nabla_{\boldsymbol{x}} f(\boldsymbol{y};\boldsymbol{x}_2,\boldsymbol{z})|| d\boldsymbol{y} \\
&+ \int_{\boldsymbol{y}} |l(\boldsymbol{x}_1,\boldsymbol{y}) - l(\boldsymbol{x}_2,\boldsymbol{y})| ||\nabla_{\boldsymbol{x}} f(\boldsymbol{y};\boldsymbol{x}_2,\boldsymbol{z})|| d\boldsymbol{y} \\
\leq& S_\Omega L_4 L_g ||\boldsymbol{x}_1 - \boldsymbol{x}_2|| + S_\Omega L_5 L_1 ||\boldsymbol{x}_1 - \boldsymbol{x}_2||.
\end{aligned}$$

The first inequality holds from the triangular inequality. The second inequality holds by the Cauchy-Schwarz inequality. The third inequality holds by the Lipschitz continuous characteristic and intermediate value theorem, where $S_\Omega$ denotes of the measurement of the set $\Omega$.

We can also bound the second term by the following steps:

$$\begin{aligned}
II \leq & \| \int_{\boldsymbol{y}} \nabla_{\boldsymbol{x}} l(\boldsymbol{x}_1, \boldsymbol{y}) f(\boldsymbol{y}; \boldsymbol{x}_1, \boldsymbol{z}) d\boldsymbol{y} - \int_{\boldsymbol{y}} \nabla_{\boldsymbol{x}} l(\boldsymbol{x}_1, \boldsymbol{y}) f(\boldsymbol{y}; \boldsymbol{x}_2, \boldsymbol{z}) d\boldsymbol{y} \| \\
& + \| \int_{\boldsymbol{y}} \nabla_{\boldsymbol{x}} l(\boldsymbol{x}_1, \boldsymbol{y}) f(\boldsymbol{y}; \boldsymbol{x}_2, \boldsymbol{z}) d\boldsymbol{y} - \int_{\boldsymbol{y}} \nabla_{\boldsymbol{x}} l(\boldsymbol{x}_2, \boldsymbol{y}) f(\boldsymbol{y}; \boldsymbol{x}_2, \boldsymbol{z}) d\boldsymbol{y} \| \\
= & \| \mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_1, \boldsymbol{z})} [\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_1, \boldsymbol{y})] - \mathbb{E}_{f(\boldsymbol{y}; \boldsymbol{x}_2, \boldsymbol{z})} \nabla_{\boldsymbol{x}} l(\boldsymbol{x}_1, \boldsymbol{y}) \| + \| \mathbb{E}_{f(\boldsymbol{y}; \boldsymbol{x}_2, \boldsymbol{z})} [\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_1, \boldsymbol{y}) - \nabla_{\boldsymbol{x}} l(\boldsymbol{x}_1, \boldsymbol{y})] \| \\
\leq & \varepsilon L_2^c \| \boldsymbol{x}_1 - \boldsymbol{x}_2 \| + L_2^c \| \boldsymbol{x}_1 - \boldsymbol{x}_2 \|.
\end{aligned}$$

The first inequality holds by the triangular inequality. The first equality holds by the definition of expectation. The second inequality holds by Lemma EC.1 and the definition of Lipschitz gradient.

Thus, $\|\nabla_{\boldsymbol{x}} g(\boldsymbol{x}_1) - \nabla_{\boldsymbol{x}} g(\boldsymbol{x}_2)\| \leq [(\varepsilon + 1) L_2^c + S_\Omega (L_1 L_5 + L_4 L_g)] \|\boldsymbol{x}_1 - \boldsymbol{x}_2\|$. Hence, the objective function has a Lipschitz gradient with $L = (\varepsilon + 1) L_2^c + S_\Omega (L_1 L_5 + L_4 L_g)$.

Hence, the objective function has a Lipschitz gradient with $L = (\varepsilon + 1) L_2^c + S_\Omega (L_1 L_5 + L_4 L_g)$.

Recall that the update rule is given by

$$\boldsymbol{x}_N^{r+1} = \boldsymbol{x}_N^r + \eta^r \hat{G}_N(\boldsymbol{x}; \boldsymbol{z}).$$

From descent lemma, we have

$$g(\boldsymbol{x}_N^{r+1}) \leq g(\boldsymbol{x}_N^r) + \eta^r \hat{G}_N(\boldsymbol{x}_N^r; \boldsymbol{z})^T \nabla g(\boldsymbol{x}_N^r) + \frac{L(\eta^r)^2}{2} \|\hat{G}_N(\boldsymbol{x}_N^r; \boldsymbol{z})\|^2.$$

Taking $N \to \infty$ on both sides, since $g(\boldsymbol{x})$ is continuous and $\lim_{N\to\infty} \hat{G}_N(\boldsymbol{x}; \boldsymbol{z}) = \mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}, \boldsymbol{y})]$ from Lemma 1.

$$g(\boldsymbol{x}^{r+1}) - g(\boldsymbol{x}^r) \leq \eta^r \mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^r, \boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^r, \boldsymbol{y})]^T \nabla g(\boldsymbol{x}^r) + \frac{L(\eta^r)^2}{2} \|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^r, \boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^r, \boldsymbol{y})]\|^2.$$

where $\boldsymbol{x}^r = \lim_{N\to\infty} \boldsymbol{x}_N^r$.

Since

$$\begin{aligned}
\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^r, \boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^r, \boldsymbol{y})]^T g(\boldsymbol{x}^r) = & \|\nabla_x \mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^r, \boldsymbol{z})}[l(\boldsymbol{x}^r, \boldsymbol{y})]\|^2 + \|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^r, \boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^r, \boldsymbol{y})]\|^2 \\
& - \| \int_{\boldsymbol{y}} l(\boldsymbol{x}^r, \boldsymbol{y}) \nabla_{\boldsymbol{x}} f(\boldsymbol{y}; \boldsymbol{x}^r, \boldsymbol{z}) d\boldsymbol{y} \|^2 \\
\geq & (\|\nabla_x \mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^r, \boldsymbol{z})}[l(\boldsymbol{x}^r, \boldsymbol{y})]\|^2 - L_4^2 L_5^2 S_\Omega^2) + \|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^r, \boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^r, \boldsymbol{y})]\|^2.
\end{aligned}$$

Note that since the range of $\boldsymbol{y}$ is limited, we can scale the random parameters $\boldsymbol{y}$ so that $S_\Omega^2 \leq \frac{\|\nabla_x \mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^r, z)}[l(\boldsymbol{x}^r, \boldsymbol{y})]\|}{L_4 L_5}$. Thus,

$$\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^r, \boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^r, \boldsymbol{y})]^T g(\boldsymbol{x}^r) \geq \|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^r, \boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^r, \boldsymbol{y})]\|^2.$$

Therefore,

$$g(\boldsymbol{x}^{r+1}) - g(\boldsymbol{x}^r) \leq -\eta^r (1 - \frac{L\eta^r}{2}) \|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^r, \boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^r, \boldsymbol{y})]\|^2.$$

Since $\eta^r$ is diminishing, for any $\zeta \in (0,1)$, there exists $\bar{r}$ such that for any $r \geq \bar{r}$, we have

$$g(\boldsymbol{x}^{r+1}) - g(\boldsymbol{x}^r) \leq -\eta^r \zeta \|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^r,\boldsymbol{y})]\|^2.$$

Since $\lim_{r\in\mathcal{K}\to\infty} \boldsymbol{x}^r = \bar{\boldsymbol{x}}$ and $g(\boldsymbol{x})$ is continuous, we have $\lim_{r\to\infty} g(\boldsymbol{x}^r) = g(\bar{\boldsymbol{x}})$. Taking summation on both sides from $r = \bar{r}$ to $\infty$, we can obtain that

$$\sum_{r=\bar{r}}^{\infty} \eta^r \zeta \|\mathbb{E}_{f(D|\boldsymbol{x}^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^r, D)]\|^2 \leq g(\boldsymbol{x}^{\bar{r}}) - \lim_{r\to\infty} g(\boldsymbol{x}^r).$$

Since $\sum_{r=\bar{r}}^{\infty} \eta^r = +\infty$, we have $\lim_{r\in\mathcal{K}\to\infty} \|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^r,\boldsymbol{y})]\|^2 = 0$, hence $\mathbb{E}_{f(\boldsymbol{y};\bar{\boldsymbol{x}},\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\bar{\boldsymbol{x}},\boldsymbol{y})] = 0$ and the proof is complete.

### EC.2.8 Proof of Proposition 4

For brevity, we directly consider the case where $N \to \infty$. Since under Assumption 1, 4(a) and 4(b), and when the requirements for lemma 1 hold, $\hat{g}(\boldsymbol{x},\boldsymbol{z}) = g(\boldsymbol{x},\boldsymbol{z})$ when $N \to \infty$ for $\boldsymbol{x} \in \mathcal{X}$ almost everywhere, we use $g(\boldsymbol{x},\boldsymbol{z})$ instead of $\hat{g}(\boldsymbol{x},\boldsymbol{z})$ in the following proof. The descent direction here is $\boldsymbol{d}^r = -\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^r,\boldsymbol{y})]$, where $\boldsymbol{x}^r = \lim_{N\to\infty} \boldsymbol{x}_N^r$. According to the armijo principle:

$$g(\boldsymbol{x}^r) - g(\boldsymbol{x}^{r+1}) \geq -\eta^r \sigma \|\mathbb{E}_{f(y|\boldsymbol{x}^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^r, y)]\|^T \boldsymbol{d}^r.$$

Since $\lim_{r(\in\mathcal{K})\to\infty} \sup_r \|\mathbb{E}_{f(y|\boldsymbol{x}^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^r, y)]\| \geq 0$. The sequence $\mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^r,\boldsymbol{z})}[l(\boldsymbol{x}^r,\boldsymbol{y})]$ decreases monotonically and has a lower bound. Thus

$$\lim_{r(\in\mathcal{K})\to\infty} g(\boldsymbol{x}^r) - g(\boldsymbol{x}^{r+1}) = 0,$$

which is followed by

$$\lim_{r(\in\mathcal{K})\to\infty} \eta^r = 0.$$

Hence, by the definition of the armijo rule, we must have for some index $\bar{r} \geq 0$

$$g(\boldsymbol{x}^r) - g(\boldsymbol{x}^r + \frac{\eta^r}{\beta}\boldsymbol{d}^r) < -\sigma\frac{\eta^r}{\beta}\|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^r,\boldsymbol{y})]\|^T \boldsymbol{d}^r, \forall r \in \mathcal{K}, r \geq \bar{r}.$$

We denote

$$\boldsymbol{p}^r := \frac{\boldsymbol{d}^r}{\|\boldsymbol{d}^r\|}, \quad \bar{\eta}^r := \frac{\eta^r \|\boldsymbol{d}^r\|}{\beta}.$$

Since $\|\boldsymbol{p}^r\| = 1$, there exists a subsequence $\{\boldsymbol{p}^r\}_{\bar{\mathcal{K}}}$ of $\{\boldsymbol{p}^r\}_{\mathcal{K}}$ such that $\{\boldsymbol{p}^r\}_{\bar{\mathcal{K}}} \to \bar{p}$, where $\bar{p}$ is a unit vector. Then

$$\frac{g(\boldsymbol{x}^r) - g(\boldsymbol{x}^{r+1})}{\bar{\eta}^r} < -\sigma(\mathbb{E}_{f(\boldsymbol{x}^r|\boldsymbol{y},\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{y},\boldsymbol{x}^r)])^T \boldsymbol{p}^r.$$

Hence,

$$\frac{g(\boldsymbol{x}^r) - \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^r,\boldsymbol{z})}[l(\boldsymbol{x}^{r+1},\boldsymbol{y})] + \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^r,\boldsymbol{z})}[l(\boldsymbol{x}^{r+1},\boldsymbol{y})] - g(\boldsymbol{x}^{r+1})}{\bar{\eta}^r} < -\sigma(\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^r,\boldsymbol{y})])^T \boldsymbol{p}^r. \qquad \text{(EC.9)}$$

By Lemma EC.1, $g(\boldsymbol{x}^r) - g(\boldsymbol{x}^{r+1}) \geq -\varepsilon L_1 \|\bar{\eta}^r \boldsymbol{p}^r\|$.

By using the mean value theorem,

$$\begin{aligned} \frac{-\varepsilon L_1 \|\bar{\eta}^r \boldsymbol{p}^r\|}{\bar{\eta}^r} &+ \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^r + \tilde{\alpha}^r \boldsymbol{p}^r, \boldsymbol{y})]^T \boldsymbol{p}^r \\ &< -\sigma(\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^r, \boldsymbol{y})])^T \boldsymbol{p}^r. \end{aligned} \tag{EC.10}$$

Let $r(\in \bar{\mathcal{K}}) \to \infty$,

$$-\varepsilon L_1 - (\mathbb{E}_{f(\boldsymbol{y}|\bar{\boldsymbol{x}},\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\bar{\boldsymbol{x}}, \boldsymbol{y})])^T \bar{p} < -\sigma \mathbb{E}_{f(\boldsymbol{y}|\bar{\boldsymbol{x}},\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\bar{\boldsymbol{x}}, \boldsymbol{y})]^T \bar{p}.$$

Substituting $\boldsymbol{d}^r = -\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}^r, \boldsymbol{y})]$, we have

$$-\varepsilon L_1 < -(1-\sigma)\|\mathbb{E}_{f(\boldsymbol{y}|\bar{\boldsymbol{x}},\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\bar{\boldsymbol{x}}, \boldsymbol{y})]\|.$$

which completes the proof.

### EC.2.9 Proof of Proposition 5

For any given sequence $\{\boldsymbol{x}_N^r\}_{r=1}^k$, denote $g_r(\boldsymbol{x}) = \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^r,\boldsymbol{z})}[l(\boldsymbol{x}, \boldsymbol{y})]$. Then $\nabla_{\boldsymbol{x}} g_r(\boldsymbol{x}) = \mathbb{E}_{f(\boldsymbol{y};\boldsymbol{x}^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}, \boldsymbol{y})]$.

Since $l(\boldsymbol{x}, \boldsymbol{y})$ has $L_1^c-$Lipschitz gradient, by the descent lemma and the assumption that $\eta L_1^c \leq 1/2$, when $N \to \infty$,

$$\begin{aligned} g_r(\boldsymbol{x}_N^{r+1}) - g_r(\boldsymbol{x}_N^r) =& g_r(\boldsymbol{x}_N^r - \eta \mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_N^r, \boldsymbol{y})]) - g_r(\boldsymbol{x}_N^r) \\ \leq& -(1 - \frac{L_1^c \eta}{2})\eta \|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_N^r, \boldsymbol{y})]\|^2. \\ \leq& -\frac{1}{2}\eta \|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_N^r, \boldsymbol{y})]\|^2. \end{aligned}$$

Thus,

$$-\frac{\eta}{2} \sum_{r=0}^{k} \|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_N^r, \boldsymbol{y})]\|^2 \geq \sum_{r=0}^{k} \left( g_r(\boldsymbol{x}_N^{r+1}) - g_r(\boldsymbol{x}_N^r) \right). \tag{EC.11}$$

And,

$$\begin{aligned} \sum_{r=0}^{k} g_r(\boldsymbol{x}_N^{r+1}) - g_r(\boldsymbol{x}_N^r) =& \sum_{r=0}^{k} [g_{r+1}(\boldsymbol{x}_N^{r+1}) - g_r(\boldsymbol{x}_N^r)] + [g_r(\boldsymbol{x}_N^{r+1}) - g_{r+1}(\boldsymbol{x}_N^{r+1})] \\ =& g_k(\boldsymbol{x}_N^k) - g_0(\boldsymbol{x}_N^0) + \sum_{r=0}^{k} [g_r(\boldsymbol{x}_N^{r+1}) - g_{r+1}(\boldsymbol{x}_N^{r+1})] \\ \geq& g_*(\boldsymbol{x}^*) - g_0(\boldsymbol{x}_N^0) - L_1 \varepsilon \sum_{r=0}^{k} \|\boldsymbol{x}_N^r - \boldsymbol{x}_N^{r+1}\| \\ =& g_0(\boldsymbol{x}_N^0) - g_*(\boldsymbol{x}^*) - L_1 \varepsilon \sum_{r=1}^{k} \|\eta \mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_N^r, \boldsymbol{y})]\| \\ \geq& g_0(\boldsymbol{x}_N^0) - g_*(\boldsymbol{x}^*) - L_1 \varepsilon \sum_{r=1}^{k} \eta L_3^c, \\ \geq& g_0(\boldsymbol{x}_N^0) - g_*(\boldsymbol{x}^*) - L_1 \varepsilon (k+1) \end{aligned}$$

where the first inequality holds because $g_*(\boldsymbol{x}^*) \leq g_r(\boldsymbol{x}_N^r)$ for any $\boldsymbol{x}^r$, Lemma EC.1 and Assumption 2. The second inequality holds by Assumption 4(e). The third inequality holds by the assumption that $\eta L_3^c \leq 1$

Then by taking the union inequality on the left side of (EC.11),

$$\begin{aligned} g_0(\boldsymbol{x}_N^0) - g_*(\boldsymbol{x}^*) - L_1\varepsilon(k+1) &\leq -\frac{\eta}{2}\sum_{r=0}^{k} \|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_N^r,\boldsymbol{y})]\|^2 \\ &\leq -\frac{\eta}{2}(k+1)\min_{r=0,\dots,k} \|\mathbb{E}_{f(\boldsymbol{y}|\boldsymbol{x}_N^r,\boldsymbol{z})}[\nabla_{\boldsymbol{x}} l(\boldsymbol{x}_N^r,\boldsymbol{y})]\|^2 \end{aligned}$$

Then by dividing both sides by $-\frac{\eta}{2}(k+1)$, the proof completes.

## EC.3 Experiment Supplements

### EC.3.1 Data Preprocessing

Table EC.1 provides the explanations and relevant statistical information regarding the effective variables that were ultimately used for the numerical experiments in the original data set. The top 10 Ford models with the highest sales volume and their corresponding trims are listed in Table EC.2. In the table, the "trim" column is ranked from the base model to the luxury version that appears in the dataset. The "description" column provides a description of this particular model, including size and type, which is later used as covariates in the case study.

**Table EC.1** Description of original vehicle sales data

| Variable | Type | Description | Statistics |
|---|---|---|---|
| Year | int | The manufacturing year of the vehicle | min 1982, max 2015 |
| Make | string | The brand or manufacturer of the vehicle | There are 96 brands or manufacturers in total, among which Ford is the most numerous. |
| model | string | The specific model of the vehicle | - |
| trim | string | Additional designation for the vehicle model | - |
| condition | int | Condition of the vehicle. | min 1, median 35, max 49 |
| odometer | int | The mileage traveled by the vehicle. | min 1, median 52254, max 100k |
| mmr | int | Manheim Market Report, indicating the estimated market value of the vehicle. | min 25, median 12250, max 182k |
| sellingprice | int | The price at which the vehicle was sold. | min 1, median 12000, max 230k |
| saledate | datetime | The date when the vehicle was sold. | From Jan 2014 to Dec 2015 |

We preprocess the data so that the relationship is revealed in a more clear way. For different models, according to Table EC.2, we introduce two variables: size and type. Model sizes (compact, mid-size, full-size) were encoded as 1, 2, and 3, while vehicle types (Truck, Sedan, SUV) were converted into dummy

**Table EC.2** Description of selected models

| Model | Total Sales | Trim order (From basic to luxury) | Description |
|---|---|---|---|
| F-150 | 14479 | XL, XLT, Lariat, King Ranch, Platinum, Limited, Raptor | mid-size Truck |
| Fusion | 12946 | S, SE, SPORT, SEL, Hybrid, Titanium | mid-size Sedan |
| Escape | 11861 | XLS, S, XLT, SE, Hybrid, SEL, Limited, Titanium | compact SUV |
| Focus | 10394 | S, SE, SEL, Titanium, Sport, Electric, BEV | mid-size Sedan |
| Explorer | 7707 | Base, XLT, Limited, Sport | mid-size SUV |
| Edge | 5915 | SE, SE Fleet, SEL, Limited, Sport, ST | mid-size SUV |
| Taurus | 4649 | Limited, SE, SEL, SHO | mid-size Sedan |
| F-250 Super Duty | 2559 | XL, XLT, Lariat, King Ranch, Platinum | Full-size Truck |
| Expedition | 2826 | EL King Ranch, EL XLT, EL Limited, King Ranch, XLT, Limited | Full-size SUV |
| Fiesta | 2375 | S, SE, SES, SEL, ST, Titanium | compact Sedan |

variables. For trims, we assign values according to their order within each model, setting the base trim to 1 and incrementing by 1 for each higher trim level. The odometer and the condition were categorized into five levels using their quantiles in the dataset, with each sales record assigned to a corresponding tier. Finally, integrating these features with the information from the year and month, we grouped the sales records and calculated the average MMR and sales price for each cluster. These steps enabled us to obtain monthly sales data for different vehicle sizes, types, conditions, and odometers, along with their respective average selling prices and market valuations.

The vehicle's cost and salvage value are calculated as follows. For cost, we used the average used car selling price and profit from Group 1 Automotive Reports[3]. The profit rate was defined as 2.63%, so each vehicle's unit cost equals the average selling price divided by this profit rate. For salvage value, we referred to the Car Edge website and found that Ford vehicles depreciate nearly linearly within 3 years[4], which can be applied on all Ford vehicles in the data set. Based on the three-year depreciation rate provided, the monthly depreciation rate was established at 1.05%[5]. The salvage value of each vehicle was then calculated as the unit cost divided by [1 - depreciation rate * current month] multiplied by [1 - depreciation rate * (current month + 1)].

### EC.3.2 Training Results

In order to identify the relationship between vehicle sales and various covariates, we first fitted their relationships through 24 different machine learning models. The training results are shown in Table EC.3. It can be seen that among these models, the Extra Tree model performed the best in all aspects.

[3] https://www.group1corp.com/2025-04-24-Group-1-Automotive-Reports-First-Quarter-2025-Financial-Results

[4] https://caredge.com/ford/depreciation

[5] https://caredge.com/ranks/depreciation/popular/3-year/best

| **Model** | **MAE** | **MSE** | **RMSE** | **R2** | **RMSLE** | **MAPE** |
|---|---|---|---|---|---|---|
| Extra Trees Regressor | 0.97 | 24.44 | 4.94 | 0.87 | 0.32 | 0.34 |
| Light Gradient Boosting Machine | 3.91 | 68.31 | 8.24 | 0.64 | 0.67 | 1.38 |
| Random Forest Regressor | 4.35 | 86.60 | 9.27 | 0.54 | 07.3 | 1.59 |
| Gradient Boosting Regressor | 5.01 | 122.78 | 11.03 | 0.36 | 0.78 | 1.83 |
| MLP Regressor | 5.84 | 145.28 | 11.99 | 0.24 | 0.90 | 2.26 |
| CART | 4.99 | 148.76 | 12.09 | 0.23 | 0.82 | 1.47 |
| K Neighbors Regressor | 5.72 | 159.81 | 12.60 | 0.15 | 0.84 | 2.01 |
| Automatic Relevance Determination | 6.43 | 175.99 | 13.19 | 0.08 | 0.98 | 2.59 |
| Linear Regression | 6.47 | 176.00 | 13.19 | 0.08 | 0.99 | 2.61 |
| Ridge Regression | 6.45 | 176.06 | 13.19 | 0.08 | 0.99 | 2.60 |
| Kernel Ridge (which includes KR) | 6.45 | 176.07 | 13.19 | 0.08 | 0.99 | 2.59 |
| Bayesian Ridge | 6.42 | 176.17 | 13.20 | 0.08 | 0.98 | 2.57 |
| Least Angle Regression | 6.49 | 176.26 | 13.20 | 0.08 | 0.99 | 2.63 |
| TheilSen Regressor | 5.53 | 182.06 | 13.41 | 0.06 | 0.86 | 1.74 |
| Orthogonal Matching Pursuit | 6.46 | 187.38 | 13.61 | 0 .03 | 1.01 | 2.59 |
| Elastic Net | 6.55 | 191.02 | 13.74 | 0.01 | 1.03 | 2.68 |
| Lasso Regression | 6.58 | 192.31 | 13.79 | 0.00 | 1.04 | 2.70 |
| Lasso Least Angle Regression | 6.59 | 192.61 | 13.80 | -0.00 | 1.04 | 2.70 |
| Dummy Regressor | 6.59 | 192.61 | 13.80 | -0.00 | 1.04 | 2.70 |
| Support Vector Regression | 4.79 | 193.06 | 13.81 | -0.00 | 0.78 | 0.79 |
| Huber Regressor | 5.01 | 198.25 | 13.99 | -0.03 | 0.82 | 0.95 |
| Passive Aggressive Regressor | 5.65 | 202.28 | 14.11 | -0.04 | 0.98 | 1.31 |
| Random Sample Consensus | 5.80 | 206.16 | 14.27 | -0.07 | 1.01 | 1.42 |
| AdaBoost Regressor | 9.90 | 220.90 | 14.74 | -0.16 | 1.29 | 4.96 |

**Table EC.3** Regression Model Performance Metrics for Sales

Table EC.4 presents the correlation coefficients for this model for different types of vehicles. It is not difficult to see that this regression model has quite good accuracy in predicting the sales volumes of compact and mid-size cars. Therefore, we believe that this model represents the true relationship between the expected sales volume of compact and mid-size vehicles and various covariates. In the subsequent case study, we will only discuss the order and pricing of compact and mid-size vehicles.

| Size | Type | R2 | RMSE |
|---|---|---|---|
| compact | Sedan | 0.87 | 1.77 |
| compact | SUV | 0.89 | 3.23 |
| mid-size | Sedan | 0.86 | 8.55 |
| mid-size | Truck | 0.88 | 3.14 |
| mid-size | SUV | 0.84 | 2.94 |
| full-size | Truck | 0.74 | 0.73 |
| full-size | SUV | 0.55 | 1.67 |

**Table EC.4** Summary of R2 and RMSE of the Regression by Size and Type

The model mentioned above reveals the relationship between the mean sales volume and the covariates. To explore how the variance relates to the covariates, we performed a secondary regression analysis on the regression residuals and the covariates, with the results presented in Table EC.5. Since there is no significant relationship between the residuals and each covariate, it can be concluded that the sales volume data exhibit homoscedasticity.

To characterize the error, we used 80 probability distributions from the Fitter package [6] for testing. However, none of the distributions passed the Kolmogorov-Smirnov test with a critical value of 0.05. Therefore, we directly used the empirical distribution of residuals as the error-term distribution, an approach also employed in Deng and Sen (2022), Kannan et al. (2024) and Sen and Deng (2017).

| **Model** | **MAE** | **MSE** | **RMSE** | **R2** | **RMSLE** | **MAPE** |
|---|---|---|---|---|---|---|
| Lasso Regression | 1.16 | 30.68 | 4.99 | -0.02 | 0.65 | 1.09 |
| Lasso Least Angle Regression | 1.16 | 30.68 | 4.99 | -0.02 | 0.65 | 1.09 |
| Dummy Regressor | 1.16 | 30.68 | 4.99 | -0.02 | 0.65 | 1.09 |
| Elastic Net | 1.16 | 30.68 | 4.99 | -0.02 | 0.65 | 1.09 |
| Support Vector Regression | 1.16 | 30.68 | 4.99 | -0.02 | 0.65 | 1.05 |
| Bayesian Ridge | 1.16 | 30.69 | 4.99 | -0.02 | 0.65 | 1.05 |
| TheilSen Regressor | 1.16 | 30.68 | 4.99 | -0.02 | 0.65 | 1.06 |
| Automatic Relevance Determination | 1.16 | 30.69 | 4.99 | -0.02 | 0.65 | 1.05 |
| Orthogonal Matching Pursuit | 1.17 | 31.06 | 5.00 | -0.03 | 0.65 | 1.17 |
| Random Sample Consensus | 1.18 | 30.72 | 5.00 | -0.02 | 0.65 | 1.31 |
| Huber Regressor | 1.16 | 30.71 | 5.00 | -0.02 | 0.65 | 1.06 |
| Linear Regression | 1.16 | 30.71 | 5.00 | -0.02 | 0.65 | 1.06 |
| Kernel Ridge | 1.20 | 30.77 | 4.99 | -0.01 | 0.65 | 1.25 |
| Ridge Regression | 1.16 | 30.68 | 4.99 | -0.02 | 0.65 | 1.05 |
| Least Angle Regression | 1.20 | 30.77 | 5.00 | -0.01 | 0.65 | 1.25 |
| MLP Regressor | 1.91 | 30.79 | 4.99 | -0.05 | 0.81 | 2.41 |
| Passive Aggressive Regressor | 1.16 | 30.68 | 4.99 | -0.02 | 0.65 | 1.09 |
| Light Gradient Boosting Machine | 1.45 | 33.39 | 5.06 | -0.30 | 0.69 | 1.26 |
| Extra Trees Regressor | 1.63 | 61.04 | 5.51 | -2.01 | 0.77 | 2.37 |
| Random Forest Regressor | 1.64 | 54.07 | 5.40 | -1.67 | 0.78 | 2.20 |
| Gradient Boosting Regressor | 1.46 | 32.43 | 5.02 | -0.25 | 0.68 | 1.36 |
| K Neighbors Regressor | 1.54 | 33.68 | 5.08 | -0.31 | 0.71 | 2.05 |
| AdaBoost Regressor | 2.61 | 93.74 | 5.62 | -3.01 | 0.95 | 3.24 |
| CART | 1.97 | 77.54 | 5.62 | -1.03 | 0.95 | 3.24 |

**Table EC.5** Regression Model Performance Metrics for Residuals

[6] https://pypi.org/project/fitter/0.2.0/