How to Teach Label to Understand Decisions: A Decision-aware Label Distribution Learning Framework

Anonymous authors
Paper under double-blind review

Abstract

Contextual Stochastic Optimization (CSO) aims to predict uncertain, context-dependent parameters to inform downstream decisions. A central challenge is that high predictive accuracy does not necessarily translate into optimal decisions. Existing approaches typically rely on custom loss functions, but these often suffer from non-differentiability, discontinuity, and limited modularity. To address these limitations, we propose a decision-aware Label Distribution Learning (LDL) framework that retains standard loss functions to avoid computational issues, while encoding decision knowledge entirely at the level of data representation. Our approach models uncertainty as full label distributions and reshapes them during the label enhancement stage to reduce predictive mass in high-risk regions. Scalar targets are transformed into individualized mixture distributions using decision-aware similarity matrices, and a dual-branch neural network is trained to learn decision-aware label distributions. Extensive experiments on synthetic benchmarks (e.g., newsvendor, network flow) and real-world datasets demonstrate consistent regret reduction across different sample sizes, with particularly strong improvements in low-data regimes. These results highlight LDL as a promising new pathway for achieving robust and principled decision-making under complex cost structures.

1 Introduction

Predict-then-optimize is a widely used paradigm for solving optimization problems under uncertainty. In this framework, given covariates, a contextual predictor first estimates the distribution of the uncertain parameters, and the resulting estimates serve as input to a Contextual Stochastic Optimization (CSO) model (Sadana et al., 2025). The traditional sequential learning-then-optimization (SLO) approach trains the contextual predictor by minimizing an estimation error between the true conditional distribution and the conditional distribution given by the contextual predictor. While effective for improving prediction accuracy, this approach neglects the downstream optimization objective and can therefore result in suboptimal decisions.

To bridge this gap, Integrated Learning and Optimization (ILO) has emerged as a promising alternative (Sadana et al., 2025). ILO methods train contextual predictors explicitly incorporating the downstream decision objective into the learning process, thereby aligning prediction and optimization. The predominant way to realize this is by designing decision-aware loss functions, which maximize decision quality on the training set (Mandi et al., 2024), rather than minimizing an estimation error.

Nevertheless, existing methods are subject to two fundamental limitations. The first concerns the high training cost of loss-function-based approaches. These methods design decision-aware loss functions (e.g., regret) to align predictions with downstream decision quality. However, such losses are often discontinuous and non-differentiable, which makes gradient-based optimization unstable and computationally expensive. Although surrogate losses have been proposed to mitigate this issue (Elmachtoub & Grigas, 2022), they still impose substantially higher training costs than conventional predictive models and frequently rely on task-specific approximations, thereby limiting their general applicability.

The second challenge is the lack of a general and adaptive framework for modeling uncertainty distributions in CSO. Modeling uncertain parameters as continuous distributions often renders the downstream optimization problem intractable, due to the curse of dimensionality arising from high-dimensional integration. A common workaround is to approximate the uncertainty using a discrete distribution. However, most prior work either fixes the discrete support set a priori (Qi et al., 2023) or derives it solely from the feature space (Bertsimas & Kallus, 2020). Such approaches overlook the fact that the choice of support set itself can have a substantial impact on decision quality.

To address the aforementioned challenges, we introduce the Label Distribution Learning (LDL) framework into CSO. LDL provides a refined way to represent uncertainty through label distributions: point labels are first enhanced into distributional labels, which then serve as the foundation for training conditional distribution predictors (Geng, 2016). This framework not only offers a more flexible mechanism for modeling uncertainty distributions, but also opens up a novel pathway to achieve decision-awareness without relying on loss-function-based methods. In particular, our paper makes the following key contributions:

- **Decision-awareness through label enhancement.** We incorporate decision-awareness at the label enhancement (LE) stage within the LDL framework. This avoids the discontinuity and computational cost associated with decision-aware loss functions while still aligning prediction with downstream decision-making.
- **General and adaptive distribution construction.** We present a method for constructing discrete uncertainty distributions by leveraging the similarity between the feature space and the decision space to determine the support set. Unlike existing methods that fix the support set a priori or derive it solely from features, our approach adapts flexibly across diverse problem settings.
- **Robustness and scalability.** Through extensive experiments on both synthetic and real-world datasets, we demonstrate that our approach consistently outperforms baseline models, delivering robust and high-quality decisions across diverse problem settings.

2 Related Works

2.1 Contextual Stochastic Optimization

Stochastic optimization is a classical paradigm for decision-making under uncertainty. A common approach is sample average approximation (SAA) (Kleywegt et al., 2002), which replaces the true distribution with an empirical one but ignores covariates. CSO addresses this by leveraging covariates to predict uncertain parameters (Sadana et al., 2025). Within CSO, prescriptive analytics extends SAA by assigning covariate-based weights to samples via k-nearest neighbors, kernel methods, or tree models (Bertsimas & Kallus, 2020), though this SLO method can yield suboptimal decisions.

To overcome this, ILO methods jointly train predictive models and decision tasks, typically through customized decision-aware loss functions. However, such losses are often discontinuous and non-differentiable, hindering gradient-based training (Mandi et al., 2024). Solutions include surrogate-based methods such as SPO+ for linear objectives (Elmachtoub & Grigas, 2022), conditional estimation–optimization (ICEO) for discrete distributions (Qi et al., 2023), perturbed maximizers (Berthet et al., 2020), differentiable solver modules (Sahoo et al., 2023; Vlastelica et al., 2020), and gradient-free models like decision trees with decision-aware objectives (Elmachtoub et al., 2020; Kallus & Mao, 2023).

Unlike prior work centered on loss design, our approach embeds decision-awareness during the LE stage within the LDL framework. This avoids reliance on differentiable surrogates or gradient propagation from the optimization model, thereby sidestepping limitations of traditional decision-focused learning.

2.2 Label Distribution Learning

LDL addresses the ambiguity in real-world labeling by assigning each instance a distribution of description degrees across labels. Unlike single-label learning, which fixes a definitive label, or

multi-label learning, which uses binary indicators without graded relevance, LDL represents supervision as a probability-like vector summing to one, thereby quantifying relative importance (Geng, 2016). Its foundations draw on fuzzy logic and probabilistic labeling, formalized as learning a conditional probability mass function to minimize divergences such as Kullback-Leibler. Early methods included problem transformation (e.g., PT-Bayes, PT-SVM), algorithm adaptation (e.g., AA-kNN via neighbor averaging, AA-BP with softmax), and specialized algorithms (e.g., SA-IIS, SA-BFGS) (Zheng et al., 2018). Evaluations across yeast gene expression, natural scenes, and facial datasets (SJAFFE, SBU-3DFE) employed diverse metrics (Chebyshev, Clark, Canberra, KL, cosine, intersection), where specialized designs often performed best (Jia et al., 2018).

To address data scarcity, LE reconstructs distributions from logical labels, with Graph Laplacian LE (GLLE) exploiting topology and correlations (Xu et al., 2021; Gu et al., 2025). Integrated approaches like Directly LDL jointly optimize LE and LDL via KL-divergence and alternating optimization, supported by Rademacher bounds and strong benchmarks (Jia et al., 2023). Objective mismatches are alleviated by Label Distribution Learning Machine (LDLM), which extends margins with SVR and adaptive losses, achieving top performance in 76.5% of tasks (Zhao et al., 2023). For ordinal data, Ordinal LDL applies sequential objectives such as Cumulative Absolute Distance, Quadratic Form Distance, and Cumulative Jensen-Shannon, yielding significant gains in age, beauty, and acne grading (Wen et al., 2023).

By representing supervision as distributions, LDL captures label ambiguity and relative importance beyond traditional settings. When applied to CSO, it enables encoding uncertainty directly in prediction, avoiding discontinuous decision-aware losses. Decision-awareness is embedded during label construction and enhancement, aligning predictive distributions with downstream optimization and enhancing robustness in decision quality—forming the basis of our proposed decision-aware LDL framework.

3 Problem Statement

In CSO, the decision-maker selects a decision variable $\mathbf{z} \in \mathcal{Z}$ to minimize the expected task cost under uncertain parameters:

$$\mathbf{z}^*(\mathbf{x}) = \arg\min_{\mathbf{z}\in\mathcal{Z}} \mathbb{E}_{\mathbf{y}\sim P(\mathbf{y}|\mathbf{x})} \left[c(\mathbf{z}, \mathbf{y})\right], \tag{1}$$

where $\mathbf{x} \in \mathcal{X}$ is the observed context, $\mathbf{y} \in \mathcal{Y}$ represents uncertain problem parameters, and $c(\mathbf{z}, \mathbf{y})$ is the task-specific cost function. A fundamental challenge arises because the conditional distribution $P(\mathbf{y} \mid \mathbf{x})$ is unknown in practice. Here, we approximate this distribution using a parameterized predictor $f(\cdot; \theta)$ parameterized by θ, taking $\mathbf{x}$ as input and outputting the corresponding distribution over $\mathbf{y}$.

The contextual predictor is typically learned from historical data. It is important to note that data on the conditional distribution $P(\mathbf{y} \mid \mathbf{x})$ is often unavailable. Instead, we have a training dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$. The problem of interest is how to train such a predictor $f(\cdot; \theta)$ so that the resulting decisions $\mathbf{z}^*(\mathbf{x})$ yield low expected cost in the downstream optimization task.

4 Methodology

This section introduces the decision-aware LDL pipeline, which constructs enhanced label distributions from feature and task information and trains a model to predict these distributions for downstream decision-making.

4.1 Decision-Aware Learning and Decision-Making Pipeline with Label Distributions

LDL first transforms each target into a distribution to capture its uncertainty in LE stage, and then learns a predictive model to map features to these distributions. Figure 1 illustrates the overall structure of the framework. The pipeline consists of two stages:

- **Label Enhancement:** Transform the regression dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, where $\mathbf{y}_i = (y_i^{(1)}, \ldots, y_i^{(K)})$ denotes the K uncertain parameters for sample i, into an enhanced dataset

$\mathcal{D}' = \{(\mathbf{x}_i, p_i(\mathbf{y}))\}_{i=1}^N$, where $p_i(\mathbf{y}) = \prod_{k=1}^K p_i(y^{(k)})$ represents the joint distribution composed of the marginal distributions $p_i(y^{(k)})$.

- **Label Distribution Learning:** Learn a vector-valued function $f(\cdot;\theta) = (f_1(\cdot;\theta_1), \ldots, f_K(\cdot;\theta_K))$, where each component $f_k(\mathbf{x}_i;\theta_k)$ predicts the marginal distribution $p_i(y^{(k)})$, and the joint distribution is reconstructed as $p_i(\mathbf{y}) = \prod_{k=1}^K f_k(\mathbf{x}_i;\theta_k)$ optimized for downstream decision-making.

To ensure tractability in the downstream decision task, we model each uncertain parameter $y_i^{(k)}$ within $\mathbf{y}_i$ using a discrete distribution. The distribution of the k-th parameter is represented as

$$p_i(y^{(k)}) = \sum_{m=1}^{M} \pi_{i,m}^{(k)} \, \delta(y^{(k)} - \mu_{i,m}^{(k)}), \tag{2}$$

where M is the number of mixture components (a hyperparameter), $\pi_{i,m}^{(k)} \geq 0$, $\sum_{m=1}^M \pi_{i,m}^{(k)} = 1$, and $\delta(\cdot)$ is the Dirac delta function. Each data point's support set is denoted as the vector $\boldsymbol{\mu}_i^{(k)} = (\mu_{i,1}^{(k)}, \ldots, \mu_{i,M}^{(k)})$, constructed individually based on approximation relationships rather than from predefined values.

In our framework, the predictive model outputs a distribution over uncertain parameters for each input, capturing multiple plausible outcomes. We then optimize the expected cost under this predicted distribution. In practice, we represent each marginal distribution as a finite mixture and solve a weighted empirical risk minimization over the mixture components. A detailed derivation and the full discrete-support formulation are provided in Appendix A.

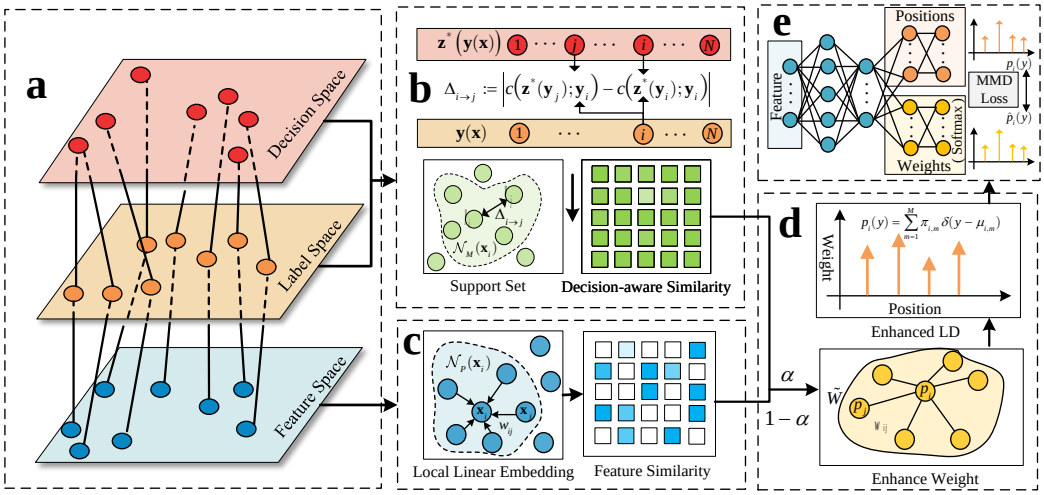

Figure 1: Overview of the Decision-aware LDL Framework; a) Mapping relationships; b) Mining decision information; c) Mining feature information; d) Constructing enhanced weights; e) Learning enhanced label distributions

4.2 Label Enhancement via Local Manifold and Task-Driven Graph Structures

4.2.1 Decision-Aware Label Support Construction

A key insight of this paper is that, as shown in Figure 1(b), rather than redefining the loss function, we reconstruct the label manifold to embed decision-awareness into label representations. To this end, we define the optimization transfer cost difference from sample j to i as

$$\Delta_{j\to i} := \left| c(\mathbf{z}^*(\mathbf{y}_i), \mathbf{y}_i) - c(\mathbf{z}^*(\mathbf{y}_j), \mathbf{y}_i) \right|, \tag{3}$$

where $\mathbf{z}^*(\mathbf{y}_i)$ denotes the optimal decision under parameters $\mathbf{y}_i$, and $c(\mathbf{z}, \mathbf{y})$ is the task cost evaluated for decision $\mathbf{z}$ under parameters $\mathbf{y}$.

A smaller value of $\Delta_{j\to i}$ indicates that, in the decision problem associated with sample i, substituting $\mathbf{y}_j$ for the true parameters $\mathbf{y}_i$ incurs only a minor additional cost. Predictions with smaller $\Delta_{j\to i}$ are thus more acceptable from the perspective of downstream decision-making. We normalize this asymmetric transfer cost into a decision-aware similarity score. Since $\min_{r,l} \Delta_{r\to l} = 0$, we have

$$s_{j\to i} = 1 - \frac{\Delta_{j\to i}}{\max_{r,l=1,\dots,N} \Delta_{r\to l}}, \tag{4}$$

where higher values denote stronger optimization-level affinity.

Finally, we perform row-wise normalization so that the similarities sum to 1 for each target i:

$$\tilde{s}_{j\to i} = \frac{s_{j\to i}}{\sum_{r=1}^{N} s_{r\to i}}. \tag{5}$$

The resulting matrix $\tilde{S} = [\tilde{s}_{j\to i}]_{i,j=1}^{N} \in \mathbb{R}^{N\times N}$ encodes the decision-aware relational structure among samples, where its (i, j)-th entry $\tilde{s}_{j\to i}$ quantifies the transferability from sample j to sample i and is utilized in the label enhancement stage.

To convert the point-supervised target $\mathbf{y}_i$ into a mixture of Dirac delta functions, we construct an individualized support vector $\boldsymbol{\mu}_i^{(k)}$ for each k-th parameter of sample i. Unlike conventional approaches that define support points solely based on feature similarity, we propose to select them according to decision-aware similarity $\tilde{s}_{j\to i}$.

Specifically, we identify the top-M neighbors whose decisions exhibit maximal transferability to sample i, characterized by the largest values of $\tilde{s}_{j\to i}$. Formally, the neighborhood is defined as

$$\mathcal{N}_M(\mathbf{x}_i) := \{j \in \{1, \dots, n\} \mid \text{rank}(\tilde{s}_{j\to i}) \le M\},$$

where $\text{rank}(\tilde{s}_{j\to i})$ denotes the rank of $\tilde{s}_{j\to i}$ in descending order among all values $\tilde{s}_{j\to i}$. The support vector corresponding to the k-th parameter is then defined as the ordered vector of the neighbor values:

$$\boldsymbol{\mu}_i^{(k)} = \big(y_j^{(k)}\big)_{j\in\mathcal{N}_M(\mathbf{x}_i)}. \tag{6}$$

This construction ensures that the support vector $\boldsymbol{\mu}_i^{(k)}$ for each sample $\mathbf{x}_i$ captures the local decision-level structure of the M most transferable neighbors for the k-th parameter, thereby providing a decision-aware foundation for label distribution reconstruction. In this way, each label component is enriched to carry more information, and by selecting support values aligned with similar decisions, the support set further guides the predictive model toward outputs that induce lower decision errors.

4.2.2 Decision-Aware Label Weighting via Manifold Reconstruction

To assign weights to each $\mu_{i,m}^{(k)}$, as shown in Figure 1(c), we draw inspiration from manifold learning techniques that capture local geometric structures in the feature space. Formally, the feature-space neighborhood of a point $\mathbf{x}_i$ is defined as

$$\mathcal{N}_P(\mathbf{x}_i) := \big\{j \in \{1, \dots, N\} \;\big|\; \text{rank}\big(d(\mathbf{x}_i, \mathbf{x}_j)\big) \le P,\ j \ne i\big\}, \tag{7}$$

where $d(\cdot, \cdot)$ denotes the distance metric in the feature space, $\text{rank}\big(d(\mathbf{x}_i, \mathbf{x}_j)\big)$ is the ascending rank of the distance $d(\mathbf{x}_i, \mathbf{x}_j)$ among all other points with respect to $\mathbf{x}_i$, and P is the number of nearest neighbors considered for each point (a hyperparameter).

Based on the neighborhood structure, we construct a local linear relationship by solving the following optimization problem. Let $W \in \mathbb{R}^{N\times N}$ denote the reconstruction weight matrix, whose (i, j)-th entry is w_{ij}. The weights are obtained by minimizing

$$\min_{W} \quad \Theta(W) := \sum_{i=1}^{N} \left\| \mathbf{x}_i - \sum_{j=1}^{N} w_{ij}\mathbf{x}_j \right\|^2, \tag{8}$$

subject to

$$\sum_{j=1}^{N} w_{ij} = 1, \quad w_{ij} = 0 \quad \text{if } j \notin \mathcal{N}_P(\mathbf{x}_i), \quad \forall i, j = 1, \ldots, N. \tag{9}$$

The objective function in equation 8 seeks to represent each point $\mathbf{x}_i$ as a convex combination of its neighbors, minimizing the reconstruction error. The constraints in equation 9 restrict the reconstruction to the P-nearest neighbors, exclude self-reconstruction, and enforce convexity.

To further align label representation with the downstream decision task, we introduce a decision-aware correction directly in the optimization objective, rather than modifying the similarity matrix W itself. Specifically, for the k-th uncertain parameter, we estimate the distribution weights $\boldsymbol{\pi}_i^{(k)} = (\pi_{i,1}^{(k)}, \ldots, \pi_{i,M}^{(k)})$, representing the probability distribution over the M values in the support vector $\boldsymbol{\mu}_i^{(k)}$. Let u denote the index of the support value $\mu_{i,u}^{(k)}$ that corresponds to the ground-truth label of the i-th sample. The optimization problem is formulated as a convex combination of two consistency terms: one based on the feature-level similarity W and the other on the task-induced similarity $\tilde{S}$:

$$\begin{aligned} \min_{\{\boldsymbol{\pi}_i^{(k)}\}} \quad \Psi(\boldsymbol{\pi}^{(k)}) := & \sum_{i=1}^{N} \left\| \boldsymbol{\mu}_i^{(k)} \boldsymbol{\pi}_i^{(k)\top} - \sum_{j=1}^{N} w_{ij}\, \boldsymbol{\mu}_j^{(k)} \boldsymbol{\pi}_j^{(k)\top} \right\|^2 \\ & + \alpha \sum_{i=1}^{N} \left\| \boldsymbol{\mu}_i^{(k)} \boldsymbol{\pi}_i^{(k)\top} - \sum_{j=1}^{N} \tilde{s}_{ij}\, \boldsymbol{\mu}_j^{(k)} \boldsymbol{\pi}_j^{(k)\top} \right\|^2, \end{aligned} \tag{10}$$

subject to

$$\sum_{m=1}^{M} \pi_{i,m}^{(k)} = 1, \qquad \forall i = 1, \ldots, N, \tag{11}$$

$$\pi_{i,m}^{(k)} \geq 0, \qquad \forall i = 1, \ldots, N,\ m = 1, \ldots, M, \tag{12}$$

$$\pi_{i,u}^{(k)} \geq \lambda, \qquad \forall i = 1, \ldots, N. \tag{13}$$

The objective in equation 10 enforces local consistency by matching the expected label values weighted by $\boldsymbol{\pi}_i$ with those of neighbors under both the feature-based similarity W and the task-induced similarity S, combined through the trade-off parameter α. The constraints in equation 11 ensure that each $\boldsymbol{\pi}_i$ forms a valid probability distribution by summing to one. The constraints in equation 12 guarantee non-negativity of all distribution components. Finally, the constraints in equation 13 enforce a minimum confidence λ on the ground-truth label for each sample, thereby incorporating supervision into the manifold-based formulation.

4.3 Enhanced Label Distribution Learning with Neural Networks

As shown in Figure 1(e), given the enhanced dataset $\mathcal{D}' = \{(\mathbf{x}_i, p_i(\mathbf{y}))\}_{i=1}^{N}$ obtained via LE, we employ K independent dual-branch neural networks $f_k(\cdot; \theta_k)$, $k = 1, \ldots, K$, to predict the marginal distributions of the K uncertain parameters individually, enabling the model to capture parameter-specific uncertainty as well as variations relevant to downstream decision-making.

For the k-th parameter, the network $f_k(\cdot; \theta_k)$ consists of an encoder and two specialized decoders for predicting mixture weights and support positions. The encoder maps the feature $\mathbf{x}_i$ through L hidden layers to generate a parameter-specific representation

$$\mathbf{h}^{(L,k)} = f_{\text{enc}}^{(k)}(\mathbf{x}_i) \in \mathbb{R}^t,$$

where t denotes the dimension of the encoder output, capturing the contextual information relevant to both decoder branches for the k-th parameter.

The decoders then compute the mixture weights and support positions as

$$\boldsymbol{\pi}^{(k)}(\mathbf{x}_i) = f_{\pi}^{(k)}\big(\mathbf{h}^{(L,k)}\big) \in \Delta^{M-1}, \tag{14}$$

$$\boldsymbol{\mu}^{(k)}(\mathbf{x}_i) = f_{\mu}^{(k)}\big(\mathbf{h}^{(L,k)}\big) \in \mathbb{R}^M, \tag{15}$$

where $f_\pi^{(k)}$ and $f_\mu^{(k)}$ denote the multi-layer decoders mapping the encoder output $\mathbf{h}^{(L,k)}$ to the respective mixture weights and support positions.

To measure the discrepancy between the predicted and target distributions for each parameter, we employ the Maximum Mean Discrepancy (MMD) metric, which quantifies the distance between distributions in a reproducing kernel Hilbert space (RKHS). The detailed derivation and closed-form expression of MMD for mixtures of Dirac delta functions are provided in Appendix B.

This design enables end-to-end learning of individualized label distributions for all K parameters, preserving the geometric structure from the LE phase while aligning with decision-aware similarities—without requiring gradient flow through downstream optimization.

5 Case Study

In this section, we evaluate the numerical performance of the proposed decision-aware LDL framework on both synthetic and real-world datasets. Synthetic data allow for controlled and reliable evaluation, while real-world data provide practical validation under realistic noise and annotation challenges. The following benchmark methods are included for comparison:

- **SAA**: This baseline disregards contextual features and determines decisions by minimizing the average cost under the empirical distribution of observed random parameters.
- **Prescriptive Analytics**: Following the framework of Bertsimas & Kallus (2020), we evaluate several local learning variants, including k-nearest neighbors (KNN), kernel regression (Kernel), local linear smoothing (LOESS), and classification and regression trees (CART tree).
- **Feature-based LDL**: As a strong baseline derived from our proposed method, this variant replaces the decision-aware similarity matrix $\mathbf{S}$ with a standard feature-based similarity. It can also be viewed as an ablation of our full framework, highlighting the contribution of decision-aware structure.

Details of the synthetic data generation process and the feature engineering procedures for both synthetic and real-world datasets (Buttler et al., 2022) are provided in the Appendix C. In our experiments, the synthetic data samples are drawn from a set of $n \in \{100, 200, 500, 700, 1000\}$, while the real-world datasets are constructed by rescaling historical data from years 1, 2, 3, and 4. To evaluate out-of-sample performance, each dataset is randomly split into training and test sets with an 80:20 ratio.

5.1 Multi-item Newsvendor Problem

The multi-item Newsvendor problem seeks the optimal replenishment quantities for K different products. Let $\mathbf{y} := (y_1, \ldots, y_K)$ denote the random demand vector for the K products, and let $\mathbf{z} \in \mathbb{R}^K$ represent the corresponding order quantities.

The demand $\mathbf{y}$ may depend on contextual factors such as promotions, holiday effects, or brand attributes. The total inventory cost consists of holding costs h_k and stockout costs b_k, which penalize overstock and understock, respectively. Thus, the cost function is defined as:

$$c(\mathbf{z}, \mathbf{y}) := \sum_{k=1}^{K} h_k(z_k - y_k)^+ + b_k(y_k - z_k)^+,$$

where $(a)^+ := \max\{a, 0\}$ denotes the positive part function.

Additionally, we impose a budget constraint $C > 0$ on the total order quantities, leading to the following feasible set:

$$\mathcal{Z} := \left\{ \mathbf{z} \in \mathbb{R}^K : \sum_{k=1}^{K} z_k \leq C,\ \mathbf{z} \geq 0 \right\}.$$

We consider the case of $K = 2$, where the newsvendor jointly decides the order quantities for two products under a total budget constraint of 200. The unit overstock costs are set to $h_1 = 1$ and

$h_2 = 1.3$, while the unit stockout costs are $b_1 = 9$ and $b_2 = 8$, respectively. For our decision-aware LDL model, the parameters are set as $P = M = 6$, $\alpha = 0.1$ and $\lambda = 0.3$.

Figure 2 and Figure 3 compare the test-set performance of decision-aware LDL and baseline approaches on synthetic and real-world data, respectively. Decision-aware LDL consistently achieves the lowest regret with strong stability, even in small-sample settings, demonstrating robustness across scenarios. Removing task-specific information increases both regret and variance. Overall, these results highlight that decision-aware LDL reliably improves decision quality in both controlled simulations and practical applications.

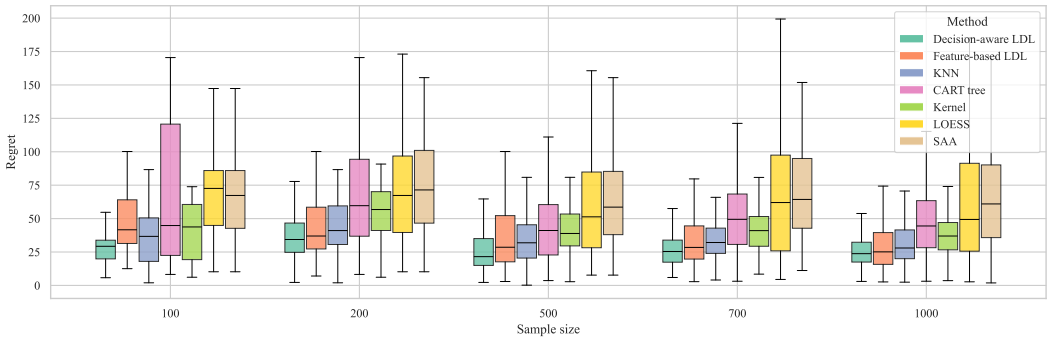

Figure 2: Comparison results for multi-item newsvendor problem in synthetic data.

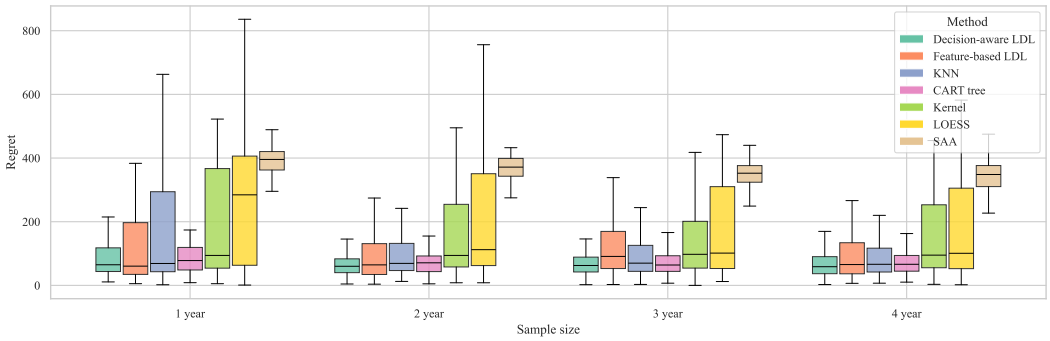

Figure 3: Comparison results for multi-item newsvendor problem in real-world data.

5.2 Quadratic Cost Network Flow Problem

Many applications such as urban traffic systems and communication networks can be formulated as a minimum convex cost flow problem. We consider a directed graph with K edges, where the decision variable $\mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K$ denotes the flow on each edge, and $\mathbf{y} = (y_1, \ldots, y_K) \in \mathbb{R}^K$ is a random parameter vector influencing the edge costs. The cost function is defined as

$$c(\mathbf{z}, \mathbf{y}) := \sum_{k=1}^{K} g_k(z_k, y_k), \tag{16}$$

where each $g_k(z_k, y_k)$ is a convex function of the flow z_k, and may vary across edges.

Let $A \in \mathbb{R}^{n \times K}$ be the node-arc incidence matrix of the graph, representing flow conservation at each node. In addition, let $C \in \mathbb{R}^{m \times K}$ be a constraint matrix that encodes edge- or path-based flow restrictions, with lower and upper bounds $\boldsymbol{\ell}, \mathbf{u} \in \mathbb{R}^m$. The feasible set is then expressed as

$$\mathcal{Z} := \left\{ \mathbf{z} \in \mathbb{R}^K : A\mathbf{z} = 0,\ \boldsymbol{\ell} \leq C\mathbf{z} \leq \mathbf{u} \right\}. \tag{17}$$

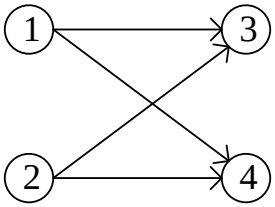

Figure 4: Network Graph

We consider a directed network with two source nodes (1 and 2) and two sink nodes (3 and 4), as illustrated in Figure 4. Let z_1, z_2, z_3, z_4 denote the flows on arcs (1,3), (1,4), (2,3), and (2,4), respectively. The flow on each arc incurs a convex cost of the form $g_k(z_k, y_k) = c_k(z_k - y_k)^2$, where y_k is a random parameter and $c_1 = 1$, $c_2 = 3, c_3 = 2, c_4 = 2$ denote the cost coefficients for each arc. Each source node must send at least 10 units of flow, and each sink node must receive at least 10 units of flow.

For our decision-aware LDL model, we set $M = P = 6$, $\alpha = 0.1$ and $\lambda = 0.3$. Due to the symmetric quadratic objective, prediction errors have a relatively small effect on decision outcomes, so less emphasis is placed on decision-specific correlations.

Figure 5 shows that decision-aware LDL consistently achieves the lowest regret with strong stability. Removing decision-specific structure increases regret and variance, though the simplified version remains acceptable. As sample size grows, performance differences narrow, indicating that all methods approach optimal decisions with more information. Overall, across diverse problems, decision-aware LDL demonstrates robust and effective decision learning.

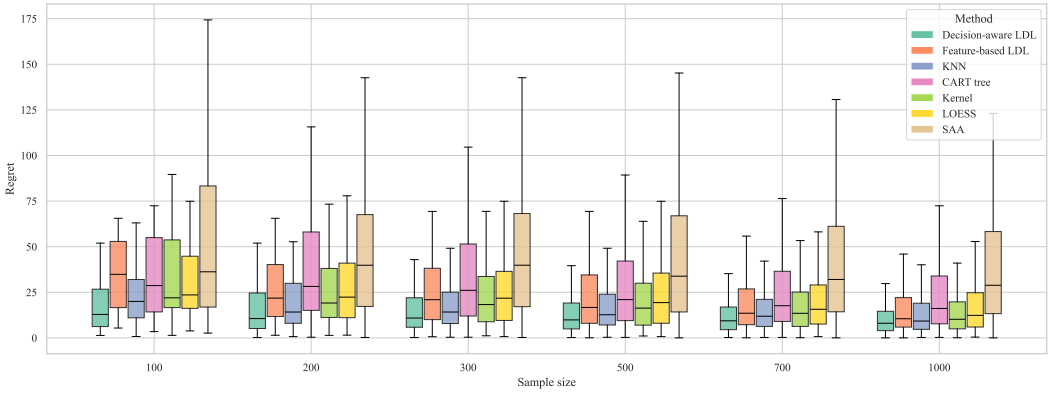

Figure 5: Comparison results for the minimum quadratic cost network flow problem.

6 Conclusion

Existing ILO approaches typically achieve decision-awareness by modifying the loss function of the predictor. However, the non-differentiable and discontinuous nature of decision-aware losses poses significant challenges for efficient training. In this work, we propose an alternative pathway that avoids loss modification.

Our decision-aware LDL framework provides a principled solution by modeling uncertainty as full distributions and strategically reallocating predictive mass away from high-risk regions. The approach transforms scalar targets into individualized mixture distributions using decision-aware similarity matrices, and employs a dual-branch neural network to learn decision-optimized representations. Experimental results on the newsvendor and network flow problems demonstrate consistent superiority in regret minimization across different sample sizes, with particularly strong performance in small-sample regimes where traditional methods struggle.

While promising, our approach has limitations including the conditional independence assumption for multivariate parameters and computational overhead of the label enhancement procedure. Future work should explore extensions to handle dependent parameters, develop efficient approximation techniques for large-scale applications, and provide theoretical performance guarantees. Nevertheless, this work establishes LDL as a viable paradigm for bridging statistical prediction and decision optimization, opening new research avenues at the intersection of machine learning and operations research.

REFERENCES

Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with Differentiable Perturbed Optimizers. *Advances in neural information processing systems*, 33:9508–9519, 2020.

Dimitris Bertsimas and Nathan Kallus. From Predictive to Prescriptive Analytics. *Management Science*, 66(3):1025–1044, March 2020. ISSN 0025-1909, 1526-5501. doi: 10.1287/mnsc.2018.3253. URL https://pubsonline.informs.org/doi/10.1287/mnsc.2018.3253.

Simone Buttler, Andreas Philippi, Nikolai Stein, and Richard Pibernik. A meta analysis of data-driven newsvendor approaches. In *ICLR 2022 Workshop on Setting up ML Evaluation Standards to Accelerate Progress*, 2022.

Adam N. Elmachtoub and Paul Grigas. Smart "Predict, then Optimize". *Management Science*, 68(1):9–26, January 2022. ISSN 0025-1909, 1526-5501. doi: 10.1287/mnsc.2020.3922. URL https://pubsonline.informs.org/doi/10.1287/mnsc.2020.3922.

Adam N Elmachtoub, Jason Cheuk Nam Liang, and Ryan McNellis. Decision trees for decision-making under the predict-then-optimize framework. pp. 2858–2867. PMLR, 2020.

Xin Geng. Label Distribution Learning. *IEEE Trans. Knowl. Data Eng.*, 28(7):1734–1748, July 2016. ISSN 1041-4347. doi: 10.1109/TKDE.2016.2545658. URL http://ieeexplore.ieee.org/document/7439855/.

Ziyuan Gu, Qi Hong, Zhen Zhou, Xin Geng, Zhiyuan Liu, and Mo Jia. Topological information utilization in label enhancement and label distribution learning based on optimal transport theory. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

Xiuyi Jia, Weiwei Li, Junyu Liu, and Yu Zhang. Label distribution learning by exploiting label correlations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Yuheng Jia, Jiawei Tang, and Jiahao Jiang. Label distribution learning from logical label. *arXiv preprint arXiv:2303.06847*, 2023.

Nathan Kallus and Xiaojie Mao. Stochastic Optimization Forests. *Management Science*, 69(4):1975–1994, April 2023. ISSN 0025-1909, 1526-5501. doi: 10.1287/mnsc.2022.4458. URL https://pubsonline.informs.org/doi/10.1287/mnsc.2022.4458.

Anton J. Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The Sample Average Approximation Method for Stochastic Discrete Optimization. *SIAM J. Optim.*, 12(2):479–502, January 2002. ISSN 1052-6234, 1095-7189. doi: 10.1137/S1052623499363220. URL http://epubs.siam.org/doi/10.1137/S1052623499363220.

Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. Decision-Focused Learning: Foundations, State of the Art, Benchmark and Future Opportunities. *jair*, 80:1623–1701, August 2024. ISSN 1076-9757. doi: 10.1613/jair.1.15320. URL https://www.jair.org/index.php/jair/article/view/15320.

Meng Qi, Paul Grigas, and Zuo-Jun Max Shen. Integrated Conditional Estimation-Optimization, August 2023. URL http://arxiv.org/abs/2110.12351. arXiv:2110.12351 [stat].

Utsav Sadana, Abhilash Chenreddy, Erick Delage, Alexandre Forel, Emma Frejinger, and Thibaut Vidal. A survey of contextual optimization methods for decision-making under uncertainty. *European Journal of Operational Research*, 320(2):271–289, January 2025. ISSN 03772217. doi: 10.1016/j.ejor.2024.03.020. URL https://linkinghub.elsevier.com/retrieve/pii/S0377221724002200.

Subham Sekhar Sahoo, Anselm Paulus, Marin Vlastelica, Vít Musil, Volodymyr Kuleshov, and Georg Martius. Backpropagation through Combinatorial Algorithms: Identity with Projection Works, March 2023. URL http://arxiv.org/abs/2205.15213. arXiv:2205.15213 [cs].

Marin Vlastelica, Anselm Paulus, Vıt Musil, Georg Martius, and Michal Rolınek. Differentiation of blackbox combinatorial solvers. 2020. URL https://openreview.net/forum?id=BkevoJSYPB.

Changsong Wen, Xin Zhang, Xingxu Yao, and Jufeng Yang. Ordinal label distribution learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 23481–23491, 2023.

Ning Xu, Yun-Peng Liu, and Xin Geng. Label Enhancement for Label Distribution Learning. *IEEE Trans. Knowl. Data Eng.*, 33(4):1632–1643, April 2021. ISSN 1041-4347, 1558-2191, 2326-3865. doi: 10.1109/TKDE.2019.2947040. URL https://ieeexplore.ieee.org/document/8868206/.

Xingyu Zhao, Lei Qi, Yuexuan An, and Xin Geng. Generalizable label distribution learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, pp. 8932–8941, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701085. doi: 10.1145/3581783.3611693. URL https://doi.org/10.1145/3581783.3611693.

Xiang Zheng, Xiuyi Jia, and Weiwei Li. Label distribution learning by exploiting sample correlations locally. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

A Discrete Mixture Representation for Decision-Aware LDL

Once the predictive function f learns to output individualized mixture distributions $\mathcal{G}_i$ for each data point, we obtain, for each input $\mathbf{x}$, a collection of mixture parameters $\{(\pi_{k,m}(\mathbf{x}), \mu_{k,m}(\mathbf{x}))\}_{m=1}^{M}$ for each dimension $k = 1, \ldots, K$. Here, K denotes the number of uncertain parameters (or output dimensions) involved in the decision problem.

In practice, we train K separate predictive models, each dedicated to learning the mixture distribution of one uncertain parameter. That is, the k-th model outputs $\{(\pi_{k,m}(\mathbf{x}), \mu_{k,m}(\mathbf{x}))\}_{m=1}^{M}$, capturing the uncertainty associated with dimension k. This decomposition allows the framework to scale to high-dimensional decision problems while preserving interpretability at the marginal level.

Each marginal distribution is represented as a mixture of Dirac delta functions:

$$p_k(y_k \mid \mathbf{x}) = \sum_{m=1}^{M} \pi_{k,m}(\mathbf{x})\, \delta(y_k - \mu_{k,m}(\mathbf{x})),$$

and under the conditional independence assumption, the joint distribution is

$$P(\mathbf{y} \mid \mathbf{x}) = \prod_{k=1}^{K} p_k(y_k \mid \mathbf{x}) = \sum_{\mathbf{m} \in [M]^K} \Pi_{\mathbf{m}}(\mathbf{x})\, \delta(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{m}}(\mathbf{x})),$$

where $\mathbf{m} = (m_1, \ldots, m_K)$ indexes one mixture component per dimension, $\boldsymbol{\mu}_{\mathbf{m}}(\mathbf{x}) = [\mu_{1,m_1}(\mathbf{x}), \ldots, \mu_{K,m_K}(\mathbf{x})]^T$, and $\Pi_{\mathbf{m}}(\mathbf{x}) = \prod_{k=1}^{K} \pi_{k,m_k}(\mathbf{x})$.

The expected cost under this distribution reduces to a weighted sum:

$$\mathbb{E}_{\mathbf{y} \sim P(\mathbf{y}|\mathbf{x})} \left[c(\mathbf{z}, \mathbf{y}) \right] = \sum_{\mathbf{m} \in [M]^K} \Pi_{\mathbf{m}}(\mathbf{x})\, c\left(\mathbf{z}, \boldsymbol{\mu}_{\mathbf{m}}(\mathbf{x})\right),$$

and the corresponding optimal decision is

$$\mathbf{z}^*_{\text{dist}}(\mathbf{x}) = \arg\min_{\mathbf{z} \in \mathcal{Z}} \sum_{\mathbf{m} \in [M]^K} \Pi_{\mathbf{m}}(\mathbf{x}) \cdot c\left(\mathbf{z}, \boldsymbol{\mu}_{\mathbf{m}}(\mathbf{x})\right).$$

B Maximum Mean Discrepancy Between Two Mixtures of Dirac Delta Functions

Let us consider two probability distributions that are discrete mixtures of Dirac delta functions:

$$P = \sum_{i=1}^{m} a_i\, \delta(x - x_i), \qquad Q = \sum_{j=1}^{n} b_j\, \delta(y - y_j),$$

where each x_i and y_j is a point in the sample space $\mathcal{X}$, and the weights satisfy

$$a_i \geq 0, \quad b_j \geq 0, \quad \sum_{i=1}^{m} a_i = \sum_{j=1}^{n} b_j = 1.$$

Here, $\delta(\cdot)$ denotes the Dirac delta distribution, so that $\delta(x - x_i)$ places all of its probability mass at the point x_i.

Given a symmetric positive definite kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ (for example, the Gaussian radial basis function kernel), the squared *Maximum Mean Discrepancy* (MMD) between P and Q is defined as:

$$\text{MMD}^2(P, Q) = \mathbb{E}_{x,x' \sim P}\, k(x, x') + \mathbb{E}_{y,y' \sim Q}\, k(y, y') - 2\, \mathbb{E}_{x \sim P,\, y \sim Q}\, k(x, y).$$

The MMD measures the distance between the *mean embeddings* of P and Q in the reproducing kernel Hilbert space (RKHS) induced by k. When k is *characteristic*, $\text{MMD}^2(P, Q) = 0$ if and only if $P = Q$.

For discrete measures such as mixtures of Dirac deltas, the expectations above reduce to finite sums. Substituting the expressions for P and Q into the MMD definition yields:

$$\mathrm{MMD}^2(P,Q) = \sum_{i=1}^{m}\sum_{i'=1}^{m} a_i a_{i'}\, k(x_i, x_{i'}) + \sum_{j=1}^{n}\sum_{j'=1}^{n} b_j b_{j'}\, k(y_j, y_{j'}) - 2\sum_{i=1}^{m}\sum_{j=1}^{n} a_i b_j\, k(x_i, y_j).$$

This expression is exact and does not require any sampling, as all terms are directly computable from the given support points and weights.

It is often convenient to express the above in matrix notation. Define:

$$K_{XX}[i,i'] = k(x_i, x_{i'}), \quad K_{YY}[j,j'] = k(y_j, y_{j'}), \quad K_{XY}[i,j] = k(x_i, y_j),$$

and let $\mathbf{a} = (a_1, \ldots, a_m)^\top$, $\mathbf{b} = (b_1, \ldots, b_n)^\top$. Then:

$$\mathrm{MMD}^2 = \mathbf{a}^\top K_{XX}\mathbf{a} + \mathbf{b}^\top K_{YY}\mathbf{b} - 2\,\mathbf{a}^\top K_{XY}\mathbf{b}.$$

This compact form is particularly useful for implementation, since it involves only matrix–vector multiplications.

The above closed-form expression is valid for any positive definite kernel k. For the Gaussian RBF kernel:

$$k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right),$$

$\mathrm{MMD}^2(P,Q)$ becomes a function of the pairwise squared Euclidean distances between $\{x_i\}$ and $\{y_j\}$, making it especially efficient to compute when these distances can be precomputed.

C Dataset Details

C.1 Real-world Bakery Data

For our experiments on real-world data, we use the bakery dataset from Buttler et al. (2022). We focus on two products from the same store. The target variable y corresponds to product demand, while the feature set X includes:

- Historical demand of the past week
- Holiday indicators: `is_schoolholiday`, `is_holiday`, `is_holiday_next2days`
- Weather-related features: `temp_min`, `temp_avg_celsius`, `temp_max`, `rain_mm`
- Promotion features: `promotion_currentweek`, `promotion_lastweek`
- Temporal features: `weekday`, `month`

All non-categorical features are normalized, while categorical features are encoded as one-hot vectors.

C.2 Synthetic Data Generation

For synthetic experiments, we generate regression datasets using `make_regression`. Specifically:

- Newsvendor problem: 4 features, with y scaled to the range [10, 120], and each demand's noise standard deviation $\sigma = 8$
- Quadratic cost network flow problem: 6 features, with y scaled to the range [5, 15], and each arc's noise standard deviation σ approximately 1.2

In both cases, we set the number of informative features to 4 to control the signal-to-noise ratio, ensuring that the synthetic targets are compatible with the respective problem characteristics.

D USE OF LARGE LANGUAGE MODELS IN MANUSCRIPT PREPARATION

During the preparation of this manuscript, large language models (LLMs) were occasionally employed to assist with tasks such as improving grammar, refining wording, and drafting certain sections of the text. These tools were used as aids to enhance clarity and readability, while all scientific content, analyses, results, and interpretations were developed and verified solely by the authors.

The use of LLMs did not influence the originality of the research, the formulation of hypotheses, the design of experiments, or the interpretation of results. The authors have carefully reviewed and edited all content generated with the assistance of LLMs to ensure accuracy, consistency, and adherence to the manuscript's scientific standards.