
Mixed-Feature Logistic Regression Robust to Distribution Shifts

Qingshi Sun^{1,3}

Nathan Justin^{2,3}

Andrés Gómez^{1,3}

Phebe Vayanos^{1,2,3}

¹Department of Industrial & Systems Engineering, USC

²Department of Computer Science, USC

³Center for Artificial Intelligence in Society, USC

{qingshis,njustin,gomezand,phebe.vayanos}@usc.edu

Abstract

Logistic regression models are widely used in the social and behavioral sciences and in high-stakes domains, due to their simplicity and interpretability properties. At the same time, such domains are permeated by distribution shifts, where the distribution generating the data changes between training and deployment. In this paper, we study a distributionally robust logistic regression problem that seeks the model that will perform best against adversarial realizations of the data distribution drawn from a suitably constructed *Wasserstein ambiguity set*. Our model and solution approach differ from prior work in that we can capture settings where the likelihood of distribution shifts can vary across features, significantly broadening the applicability of our model relative to the state-of-the-art. We propose a graph-based solution approach that can be integrated into off-the-shelf optimization solvers. We evaluate the performance of our model and algorithms on numerous publicly available datasets. Our solution achieves a 408x speed-up relative to the state-of-the-art. Additionally, compared to the state-of-the-art, our model reduces average calibration error by up to 36.19% and worst-case calibration error by up to 41.70%, while increasing the average area under the ROC curve (AUC) by up to 18.02% and worst-case AUC by up to 48.37%.

1 INTRODUCTION

Machine learning plays a critical role in decision-making in high-stake domains, such as healthcare, social science, finance, and criminal justice (Rudin, 2019). Within such domains, model transparency and interpretability are critical to ensure trustworthiness (Rudin et al., 2022). One of the most commonly used classification methods in such settings is classical logistic regression, which models the probability of a binary outcome by mapping input features to a probability value using a logistic function (Hosmer and Lemeshow, 2000). Logistic regression is highly interpretable as it models the log-odds of an event as a linear combination of the covariates, making it easy to understand how changes in input variables affect the probability outcome.

However, in high-stakes domains where this model is often used, distribution shifts are ubiquitous. These refer to differences between the training data distribution and the data encountered during testing or deployment. In this work, we specifically address conditional shift, where the conditional distribution of the input features given the label varies, while the marginal distribution of the label remains unchanged (Zhang et al., 2013). This phenomenon can also be viewed as a form of concept drift (Lu et al., 2019; Gama et al., 2014). For example, in the high-stakes domain of public housing allocation, personal information used to predict homelessness risk is often collected through self-reported surveys (Rice et al., 2023). Over time, refinements in data collection practices—such as changes in question phrasing or survey locations—can lead to shifts in the way individual features are reported, even when an individual’s homelessness status (label) remains unchanged. At the same time, the distribution of the label remains consistent because the definition of homelessness is independent of how the personal information is collected. Conditional shifts can occur due to changes in data collection protocols, technological advancements, or environmen-

tal variations, often leading to a significant decline in model performance. Similarly to most machine learning (ML) models, logistic regression is susceptible to distribution shifts. It is therefore critical to develop logistic regression models that remain *robust*, even in the presence of changing data distributions.

In recent years, distributionally robust optimization (DRO) methods have been introduced to enhance the robustness of ML models against distribution shifts (Duchi and Namkoong, 2021; Zhang et al., 2021; Soma et al., 2022). In ML, DRO seeks to find the model that will perform best in the worst-case realization of the distribution of the data within a suitably constructed ambiguity set that captures e.g., prior knowledge about the distribution of the data and about the likelihood of different shifts.

In the literature, many metrics are proposed to construct ambiguity sets, such as moment based uncertainty (Delage and Ye, 2010), Kullback-Leibler divergence (Lam, 2019), and Wasserstein distance (Mohajerin Esfahani and Kuhn, 2018). In particular, Mohajerin Esfahani and Kuhn (2018) have shown that DRO problems based on the Wasserstein distance can be reformulated as finite convex programs under mild assumptions and that they have attractive convergence properties and provable out-of-sample performance guarantees.

In our work, we study a distributionally robust variant of logistic regression where the distance between the training data distribution and the testing/deployment data distribution is measured by the Wasserstein metric. Our work most closely relates to two papers in the literature. Shafieezadeh-Abadeh et al. (2015) are the first to propose a DRO approach to logistic regression, showing that a distributionally robust logistic regression problem admits an equivalent reformulation as a polynomial-size convex optimization problem if all features are numerical. Then, Selvi et al. (2022) extend this approach to solve the problem when both numerical and categorical features are present, resulting in an exponential-size convex optimization problem. They also propose a cutting-plane method to solve the problem as a sequence of polynomial-time solvable programs. Importantly, both of these works assume that all features are equally prone to shift, i.e., that the likelihood of a shift is equally likely across all features.

However, in real-world situations, distribution shifts often affect different features differently and we have access to partial information about the likelihood of different shifts; for example, we may know that certain features are more prone to variability than others. Prior work has explored the design of Wasserstein

ambiguity sets to account for feature heterogeneity. Blanchet et al. (2019) propose a Mahalanobis-based distance metric to incorporate feature heterogeneity; however, their approach does not explicitly address parameter selection under distribution shifts. Building on this work, Liu et al. (2024) introduce a heuristic approach that prioritizes features with substantial shifts. However, their method is specifically designed for covariate shifts (Quionero-Candela et al., 2009), assumes access to target domain data, and only considers a binary (0-1) weighting scheme when modeling feature heterogeneity. In many real-world scenarios, access to target domain data cannot be guaranteed. Furthermore, a rigid binary weighting scheme lacks flexibility in modeling varying degrees of distribution shifts across features.

In this paper, we study the problem of learning a logistic regression model that is robust to conditional shifts where the likelihood of a shift may differ across features and applicable to datasets involving mixed features. The goal is to ensure the best possible performance under conditional shifts in the training data, leading to reliable outcomes during deployment. The main contributions of our work are:

- On the model side, we propose a distributionally robust logistic regression model with a Wasserstein ambiguity set that accounts for distribution shifts, where such shifts can affect different features differently. Additionally, we introduce a fine-grained method for calibrating the ambiguity set based on basic domain knowledge.
- On the algorithmic side, we adapt the cutting-plane method in Selvi et al. (2022) to solve our proposed model. Additionally, we develop a graph-based reformulation that significantly reduces runtime relative to the cutting-plane method, making it practically scalable for training models on large datasets.
- On the computational side, we demonstrate that our proposed distributionally robust logistic regression improves performance across several metrics, including calibration error and AUC, compared to both standard regularized logistic regression and existing models under distribution shifts. In particular, compared to the state-of-the-art model, our model reduces average calibration error by up to 36.19% and worst-case calibration error by up to 41.70%, while increasing the average AUC by up to 18.02% and worst-case AUC by up to 48.37%. Moreover, the graph-based formulation can be solved up to 408.12 times faster on average compared to the state-of-the-art algorithm.

Notation. We define $[N] = \{1, \dots, N\}$ for $N \in \mathbb{N}$. The indicator function, denoted as $\mathbf{1}[\mathcal{E}]$, equals 1 if the condition \mathcal{E} is satisfied and 0 otherwise. The set of all probability distributions supported on a set Ξ is denoted by $\mathcal{P}_0(\Xi)$.

2 DISTRIBUTIONALLY ROBUST OPTIMIZATION FORMULATION

2.1 Wasserstein Logistic Regression and Ambiguity Set

We consider a mixed-feature classification dataset with N data points, $\{\xi^i := (\mathbf{x}^i, \mathbf{z}^i, y^i)\}_{i \in [N]}$, where ξ^i collects the features and labels related to datapoint i . Specifically, each datapoint i is characterized by n numerical features $\mathbf{x}^i = (x_1^i, \dots, x_n^i) \in \mathbb{R}^n$, m categorical features $\mathbf{z}^i = (z_1^i, \dots, z_m^i) \in \mathcal{C}_1 \times \dots \times \mathcal{C}_m$, and a binary label $y^i \in \{-1, +1\}$. Here, for $j \in [m]$, if the number of possible values of z_j is a , $\mathcal{C}_j := \{z \in \{0, 1\}^{a-1} : \sum_{k \in [a-1]} z_k \leq 1\}$, representing the set of one-hot encoded categorical features associated with feature j . We denote by $\mathcal{C} := \mathcal{C}_1 \times \dots \times \mathcal{C}_m$ and $\Xi := \mathbb{R}^n \times \mathcal{C} \times \{-1, +1\}$ the support of the categorical features as well as the dataset, respectively, and we let c be the number of encoded categorical features.

We select Wasserstein ball $\mathfrak{B}_\epsilon(\hat{\mathbb{P}}_N)$ as the ambiguity set, which contains all possible distributions \mathbb{Q} centered at the empirical distribution $\hat{\mathbb{P}}_N$. Since the true distribution of the training data is unknown in practice, we use the empirical distribution $\hat{\mathbb{P}}_N := \frac{1}{N} \sum_{i=1}^N \delta_{\xi^i}$ that places equal probability mass on all data points $\{\xi^i\}_{i \in [N]}$, where δ_{ξ^i} denotes the Dirac point measure at ξ^i . We aim to solve the following logistic regression problem robust to distribution shifts:

$$\begin{aligned} & \underset{\beta}{\text{minimize}} && \sup_{\mathbb{Q} \in \mathfrak{B}_\epsilon(\hat{\mathbb{P}}_N)} \mathbb{E}_{\mathbb{Q}}[l_\beta(\mathbf{x}, \mathbf{z}, y)] \\ & \text{subject to} && \beta = (\beta_0, \beta_x, \beta_z) \in \mathbb{R}^{1+n+c}, \end{aligned} \quad (1)$$

where l_β is the log-loss function defined through

$$l_\beta(\mathbf{x}, \mathbf{z}, y) := \log(1 + \exp(-y \cdot (\beta_0 + \beta_x^\top \mathbf{x} + \beta_z^\top \mathbf{z}))).$$

Problem (1) hedges against a broad range of potential distributions and identifies the optimal coefficients by minimizing the expected log loss under the worst-case distribution, providing a safeguard against data deviations during deployment.

Formally, we define

$$\mathfrak{B}_\epsilon(\hat{\mathbb{P}}_N) := \{\mathbb{Q} \in \mathcal{P}_0(\Xi) : W(\mathbb{Q}, \hat{\mathbb{P}}_N) \leq \epsilon\} \quad (2)$$

as the ball of the radius $\epsilon \in \mathbb{R}^+$ centered at $\hat{\mathbb{P}}_N$ with respect to the Wasserstein distance defined below.

Definition 1 (Wasserstein Distance). The Wasserstein distance between two distributions \mathbb{P} and \mathbb{Q} supported on Ξ is defined as

$$W(\mathbb{Q}, \mathbb{P}) := \inf_{\Pi \in \mathcal{P}_0(\Xi^2)} \left\{ \int_{\Xi^2} d(\xi, \xi') \Pi(d\xi, d\xi') : \Pi(d\xi, \Xi) = \mathbb{Q}(d\xi), \Pi(\Xi, d\xi') = \mathbb{P}(d\xi') \right\},$$

where $\xi = (\mathbf{x}, \mathbf{z}, y) \in \Xi$ and $\xi' = (\mathbf{x}', \mathbf{z}', y') \in \Xi$, while $d(\xi, \xi')$ is a weighted distance metric on Ξ .

The Wasserstein distance represents the minimum cost of moving the distribution \mathbb{P} to the distribution \mathbb{Q} , where the cost of moving a unit mass from ξ to ξ' amounts to $d(\xi, \xi')$.

Next, we define the weighted distance metric used in the Wasserstein distance.

Definition 2 (Weighted Distance Metric). We measure the distance between two data points $\xi = (\mathbf{x}, \mathbf{z}, y) \in \Xi$ and $\xi' = (\mathbf{x}', \mathbf{z}', y') \in \Xi$ as

$$d(\xi, \xi') := \sum_{j \in [n]} \gamma_j |x_j - x'_j| + \sum_{\ell \in [m]} \delta_\ell \mathbf{1}[z_\ell \neq z'_\ell] + \kappa \cdot \mathbf{1}[y \neq y']$$

for some $\gamma_j > 0$, $\delta_\ell > 0$, and $\kappa > 0$.

In this definition, the weighted distance between numerical features \mathbf{x} and \mathbf{x}' is defined by the norm-based difference. Here, γ_j is the weight of the perturbation of numerical feature j , representing the cost per unit difference between x_j and x'_j . The weighted distance between two categorical feature vectors \mathbf{z} and \mathbf{z}' is defined by the discrepancies between their corresponding components. δ_ℓ represents the weight associated with the perturbation of categorical feature ℓ , representing the cost when z_ℓ and z'_ℓ differ. Similarly, the discrepancy between the labels y and y' is accounted for by a constant κ . By selecting proper weights, we can account for the relative sensitivity of the model to different features, ensuring that the optimization is more robust to shifts in critical dimensions while reducing unnecessary conservatism in others. In this work, we use weight parameters γ_j and δ_ℓ in the ambiguity set to account for the varying likelihood of distribution shifts across different features. We discuss a method for calibrating all parameters in the ambiguity set (2) in Section 3.

2.2 Reformulation as a Convex Problem with Exponential Number of Constraints

In this work, since the marginal distribution of label in the testing data remains unchanged, we set κ in Definition 2 to infinity, implying that any shifts in the labels will result in distributions falling outside the ambiguity set.

Theorem 1 (Convex Formulation). *Without shifts in labels, the distributionally robust logistic regression problem (1) can be reformulated as*

$$\begin{aligned}
 \min_{\lambda, \mathbf{r}, \boldsymbol{\beta}} \quad & \lambda \epsilon + \frac{1}{N} \sum_{i \in [N]} r_i \\
 \text{s.t.} \quad & l_{\boldsymbol{\beta}}(\mathbf{x}^i, \mathbf{z}, y^i) - \lambda \sum_{\ell \in [m]} \delta_{\ell} \mathbb{1}[\mathbf{z}_{\ell} \neq \mathbf{z}_{\ell}^i] \leq r_i, \\
 & \forall i \in [N], \mathbf{z} \in \mathcal{C} \\
 & |\gamma_j^{-1} \beta_{xj}| \leq \lambda, \forall j \in [n] \\
 & \lambda \geq 0, \mathbf{r} \in \mathbb{R}^N, \boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}_x, \boldsymbol{\beta}_z) \in \mathbb{R}^{1+n+c}
 \end{aligned} \tag{3}$$

where λ and \mathbf{r} are dual variables arising by dualizing the inner maximization problem in (1). The constraints with log-loss functions can be converted into exponential cone format, resulting in a convex problem that can be solved with off-the-shelf solvers.

The proof of Theorem 1 and the equivalent formulation of problem (3) to an exponential cone problem are given in Appendix E.1. While problem (3) can in principle be solved with off-the-shelf solvers, it contains exponentially many constraints, making it impractical to solve monolithically when the number of possible realizations of each categorical feature is large. In Section 4, we propose two solution methods that will enable us to solve practically sized problems.

3 PARAMETER CALIBRATION

We develop a fine-grained method for calibrating the ambiguity set radius ϵ and the weight parameters γ_j and δ_{ℓ} where $j \in [n]$ and $\ell \in [m]$ based on the estimated likelihood of distribution shifts. We build on the calibration method proposed by Justin et al. (2023) in the context of robust optimization with discrete features. We adapt this technique to the calibration of parameters in our *distributionally* robust optimization problem, and extend its application from discrete features to numerical features.

Without access to target domain data, we rely on basic domain knowledge to characterize uncertainty in numerical features. Specifically, for each numerical feature $j \in [n]$, we denote the corresponding feature after distribution shifts as \tilde{x}_j and define its probability density function f given x_j . To quantify uncertainty, we assume a probability of certainty $\rho_{xj} \in (0, 1]$ and an interval $[l_j, u_j]$, where

$$\rho_{xj} = \mathbb{P}(l_j \leq \tilde{x}_j - x_j \leq u_j) = \int_{l_j + x_j}^{u_j + x_j} f(\tilde{x}_j) d\tilde{x}_j. \tag{4}$$

This represents the probability that the shift in x_j dur-

ing deployment falls within the interval $[l_j, u_j]$. These intervals and probabilities can be estimated from historical data or expert evaluation. A feature with a low probability of certainty and a large interval is highly susceptible to shifts, whereas a feature with a high probability of certainty and a small interval is more stable and likely to retain a similar distribution in deployment as observed during training.

In the absence of further knowledge about the distribution shift, we follow the principle of maximum entropy (Jaynes, 1957), which chooses the distribution of the perturbations with greatest entropy and thus highest uncertainty subject to our assumption of the probability of certainty ρ_{xj} and $\rho_{z\ell}$. To this end, we select the Laplace distribution to reflect shifts on the numerical features. In particular, we assume that the shift of numerical feature j follow a Laplace distribution with scale parameter b_j and location parameter 0. We also assume that the shift does not have a specific trend in the positive or negative direction and thus set $u_j = -l_j$. Specifically, given x_j , the probability density function f of \tilde{x}_j is defined as

$$f(\tilde{x}_j) = \frac{1}{2b_j} \exp\left(-\frac{|\tilde{x}_j - x_j|}{b_j}\right). \tag{5}$$

Combining (4) and (5), we obtain $b_j = \frac{-u_j}{\log(1-\rho_{xj})}$.

Similarly, for each categorical feature $\ell \in [m]$, we denote the probability of \mathbf{z}_{ℓ} remaining unchanged in deployment as $\rho_{z\ell} \in \left[\frac{1}{|\mathcal{C}_{\ell}|}, 1\right]$. Following the principle of maximum entropy, we assume that the probability of perturbing \mathbf{z}_{ℓ} to $\tilde{\mathbf{z}}_{\ell} \in \mathcal{C}_{\ell}$ is

$$\mathbb{P}(\tilde{\mathbf{z}}_{\ell}) = \mathbb{1}[\tilde{\mathbf{z}}_{\ell} = \mathbf{z}_{\ell}] \rho_{z\ell} + \mathbb{1}[\tilde{\mathbf{z}}_{\ell} \neq \mathbf{z}_{\ell}] \frac{1 - \rho_{z\ell}}{|\mathcal{C}_{\ell}| - 1}. \tag{6}$$

The above probability mass function allows for any perturbation of \mathbf{z}_{ℓ} to occur with the same probability $\frac{1 - \rho_{z\ell}}{|\mathcal{C}_{\ell}| - 1}$.

Lastly, we adopt the approach of constructing ambiguity sets using hypothesis testing. Specifically, we perform a likelihood ratio test on the magnitude of the perturbation, with a threshold determined by θ , where $\theta \in (0, 1]$ controls the level of robustness. A value of θ close to 0 signifies a fully robust model and $\theta = 1$ signifies no robustness in the model. We refer to Appendix A on how the probability density function (5), probability mass function (6), and the value of θ can be used to calibrate the values of γ_j , δ_{ℓ} , and ϵ , respectively, through hypothesis testing. From our derivations in Appendix A, we set $\gamma_j = \frac{-\log(1-\rho_{xj})}{u_j}$, $\delta_{\ell} = \log\left(\frac{\rho_{z\ell}(|\mathcal{C}_{\ell}|-1)}{1-\rho_{z\ell}}\right)$, and $\epsilon = -\log \theta$ as the tuned parameters of our ambiguity set.

Notably, by using application-specific information, alternative distributions of perturbations can be tailored to more accurately capture the distribution shifts. Our calibration method adjusts the parameters γ_j , δ_ℓ , and ϵ accordingly. Although this method involves domain knowledge, the numerical results in Appendix D show that even with imprecise estimates and model misspecification, our calibrated model maintains strong performance under distribution shifts.

4 SOLUTION METHODS

4.1 Cutting-Plane Method

We adapt the cutting-plane method from Selvi et al. (2022) to solve the distributionally robust logistic regression problem (3) with ambiguity set (2). At each iteration, the algorithm solves a relaxed version of problem (3), identifies the constraints in the original problem that are violated the most by the current solution to the relaxed problem, and incorporates these constraints into the relaxed problem to progressively improve the solution. This process continues until no violated constraints remain, ensuring convergence to the optimal solution. Our proposed cutting-plane approach differs from that in Selvi et al. (2022) in the method we use to identify violated constraints. Indeed, the approach from Selvi et al. (2022) does not apply when the weight parameters δ_ℓ are not all 1. Their constraint identification algorithm relies on a direct mapping between the distance $d(\xi, \xi')$ and the number of feature disagreements. When weights vary, this relationship breaks, making it difficult to determine which features differ or how many are different. In what follows we focus on describing our method for identifying violated constraints. For details on how this procedure can be integrated in a cutting-plane algorithm for solving problem (3), we refer to Appendix B.

We now describe our procedure to identify, for any fixed solution $(\lambda, \mathbf{r}, \boldsymbol{\beta})$ to the relaxed problem, the most violated constraints in problem (3). These are indexed by $(i, \mathbf{z}) \in [N] \times \mathcal{C}$. We propose to solve, for each $i \in [N]$, the following optimization problem:

$$\begin{aligned} \text{maximize} \quad & \log(1 + \exp(-y^i(\boldsymbol{\beta}_x^\top \mathbf{x}^i + \boldsymbol{\beta}_z^\top \mathbf{z} + \beta_0))) \\ & - \lambda \sum_{\ell \in [m]} \delta_\ell \mathbb{1}[\mathbf{z}_\ell \neq \mathbf{z}_\ell^i] - r_i. \end{aligned} \quad (7)$$

In this problem, any solution $\mathbf{z} \in \mathcal{C}$ that results in an objective value greater than 0 corresponds to a violated constraint. Problem (7) is challenging due to the exponential number of possible realizations of \mathbf{z} . In problem (7), the decision variables \mathbf{z} only appear in the linear term $-y^i \sum_{\ell \in [m]} \boldsymbol{\beta}_{z\ell}^\top \mathbf{z}_\ell$ and the weighted

distance $\sum_{\ell \in [m]} \delta_\ell \mathbb{1}[\mathbf{z}_\ell \neq \mathbf{z}_\ell^i]$. At the same time, since both terms are linearly separable, this structure allows for the decomposition of the problem into subproblems, where each subproblem considers a subset of categorical features and fixes a weighted distance corresponding to those features. Therefore, we propose to solve this problem using dynamic programming, which allows us to decompose the problem into simpler subproblems, conditioned on weighted distances, to speed-up computation.

For any fixed datapoint i , we define the set of non-root and non-terminal states of our dynamic program as

$$\mathcal{S}_1^i := \left\{ (k, d) \mid d = \sum_{\ell=1}^k \delta_\ell \mathbb{1}[\mathbf{z}_\ell \neq \mathbf{z}_\ell^i], k \in [m], \mathbf{z} \in \mathcal{C} \right\}.$$

In this definition, $k \in [m]$ specifies that categorical features indexed from 1 to k are considered, and d represents the sum of the weighted distances between a value $\mathbf{z}_\ell \in \mathcal{C}_\ell$ and \mathbf{z}_ℓ^i in the dataset across all categorical features $\ell \in [k]$. We also define a set \mathcal{S}_0 containing the root state and terminal state: $\mathcal{S}_0 = \{(0, 0), (m+1, 0)\}$. The state space of each dynamic programming problem indexed by data point i is $\mathcal{S}^i = \mathcal{S}_1^i \cup \mathcal{S}_0$. We denote the optimal objective value of the dynamic programming subproblems by a function $g^i : \mathcal{S}^i \rightarrow \mathbb{R}$. That is, for each state $(k, d) \in \mathcal{S}_1^i$, we define

$$\begin{aligned} g^i(k, d) := & \max_{\{\mathbf{z}_\ell\}_{\ell \in [k]}} -y^i \sum_{\ell \in [k]} \boldsymbol{\beta}_{z\ell}^\top \mathbf{z}_\ell \\ \text{s.t.} \quad & \mathbf{z}_\ell \in \mathcal{C}_\ell, \ell \in [k] \\ & \sum_{\ell \in [k]} \delta_\ell \mathbb{1}[\mathbf{z}_\ell \neq \mathbf{z}_\ell^i] = d \end{aligned} \quad (8)$$

Each subproblem (8) only considers categorical features indexed from 1 to k for $k \in [m]$ and is conditioned on the weighted distance d corresponding to those k categorical features. Additionally, we set $g^i(0, 0) := 0$ and denote by $g^i(m+1, 0)$ the optimal objective value of problem (7).

Subproblems (8) can be solved recursively using the following Bellman equations.

If $k \in [m]$,

$$g^i(k, d) = \max_{\mathbf{z}_k \in \mathcal{C}_k} -y^i \boldsymbol{\beta}_{z_k}^\top \mathbf{z}_k + g^i(k-1, d - \delta_k \mathbb{1}[\mathbf{z}_k \neq \mathbf{z}_k^i]), \quad (9)$$

where

$$\mathcal{F}_{kd} := \{\mathbf{z}_k \in \mathcal{C}_k \mid (k-1, d - \delta_k \mathbb{1}[\mathbf{z}_k \neq \mathbf{z}_k^i]) \in \mathcal{S}^i\}.$$

If $k = m+1$,

$$\begin{aligned} g^i(m+1, 0) = & \max_d \log(1 + \exp(-y^i \boldsymbol{\beta}_x^\top \mathbf{x}^i \\ & + g^i(m, d))) - \lambda d - r_i \\ \text{s.t.} \quad & (m, d) \in \mathcal{S}_1^i. \end{aligned} \quad (10)$$

Algorithm 1 Identification of Most Violated Constraints in Problem (3) given the data point indexed by i .

Input: A solution $(\lambda, \mathbf{r}, \beta)$ to the relaxed problem of (3) and a datapoint $i \in \Xi$

Output: optimal value to problem (7) and a corresponding optimal solution

- 1: **Initialize** $z_{[0]}(0) \leftarrow ()$
- 2: **Initialize** $g^i(0, 0) \leftarrow 0$
- 3: **for** $(k, d) \in \mathcal{S}_1^i$ **do**
- 4: **Solve** $g^i(k, d)$ by Bellman equation (9); denote the corresponding optimal solution of (9) by $z_k(d)$.
- 5: $z_{[k]}(d) \leftarrow (z_{[k-1]}(d - \delta_k \mathbb{1}[z_k(d) \neq z_k^i]), z_k(d))$
- 6: **end for**
- 7: **Solve** $g^i(m+1, 0)$ by Bellman equation (10); denote the corresponding optimal solution of (10) by d^* .
- return** $g^i(m+1, 0)$ **and** $z_{[m]}(d^*)$

Starting from the simplest case with a single categorical feature, Bellman equation (9) incrementally solves subproblem (8) by adding one categorical feature at a time. The linear separability of the objective function $-y^i \sum_{\ell \in [k]} \beta_{z\ell}^\top z_\ell$ enables this incremental approach. Bellman equation (9) follows this strategy by leveraging the optimal solution of subproblem (8) with categorical features indexed from 1 to $k-1$ and evaluating all feasible values of the k th categorical feature. Finally, solving subproblem (8) for $k = m$ and a given distance d for $(m, d) \in \mathcal{S}^i$ yields $g^i(m, d)$. Bellman equation (10) compares $g^i(m, d)$ with the corresponding solution for all $(m, d) \in \mathcal{S}^i$ to obtain the optimal solution for problem (7). Algorithm 1 provides the details of this dynamic programming procedure. The proof of correctness of Algorithm 1 can be found in Appendix E.2.

4.2 Graph-based Reformulation

Although the cutting-plane method does scale better than solving problem (3) monolithically, it is still limiting in the sizes of datasets that it can handle, as we will show later in Section 5.1. This limitation motivates us to propose a new framework for solving problem (3).

Our idea is to convert each of the constraints indexed by $(i, \mathbf{z}) \in [N] \times \mathcal{C}$ in problem (3) into a more tractable form, resulting in a formulation that can be solved directly. For each data point indexed by $i \in [N]$, we recall the original constraint set in problem (3),

$$r_i \geq l_\beta(\mathbf{x}^i, \mathbf{z}, y^i) - \lambda \sum_{\ell \in [m]} \delta_\ell \mathbb{1}[z_\ell \neq z_\ell^i], \forall \mathbf{z} \in \mathcal{C}$$

By applying proper exponentiation operations, we de-

fine an equivalent optimization problem over all possible realizations of the categorical features \mathcal{C} and establish an inequality that $y^i (\beta_x^\top \mathbf{x}^i + \beta_0)$ is greater than or equal to:

$$\max_{\mathbf{z} \in \mathcal{C}} -y^i \beta_z^\top \mathbf{z} - \log \left(-1 + \exp \left(r_i + \lambda \sum_{\ell=1}^m \delta_\ell \mathbb{1}[z_\ell \neq z_\ell^i] \right) \right). \quad (11)$$

This inequality holds if and only if the constraints for all $\mathbf{z} \in \mathcal{C}$ are satisfied. We analyze problem (11) using dynamic programming. For the data point indexed by i , we choose the same state space \mathcal{S}^i . We denote the optimal objective value of each subproblem by $h^i : \mathcal{S}^i \rightarrow \mathbb{R}$. We set $h^i(0, 0) := 0$ and denote by $h^i(m+1, 0)$ the optimal objective value of problem (11). Due to its structural similarity to problem (7), we continue to set subproblem (8) for each state $(k, d) \in \mathcal{S}_1^i$ and define $h^i(k, d) := g^i(k, d)$ for all $(k, d) \in \mathcal{S}_1^i$. Problem (11) can be solved recursively using the following Bellman equations.

If $k \in [m]$, similar to equation (9),

$$h^i(k, d) = \max_{\mathbf{z}_k \in \mathcal{F}_{kd}} -y^i \beta_{z_k}^\top \mathbf{z}_k + h^i(k-1, d - \delta_k \mathbb{1}[z_k \neq z_k^i]). \quad (12)$$

If $k = m+1$,

$$\begin{aligned} h^i(m+1, 0) &= \max_d h^i(m, d) \\ &\quad - \log(-1 + \exp(r_i + \lambda d)) \quad (13) \\ \text{s.t. } &(m, d) \in \mathcal{S}_1^i. \end{aligned}$$

Note that problem (13) differs from problem (10) due to the differences between problem (7) and problem (11). We need to guarantee $y^i (\beta_x^\top \mathbf{x}^i + \beta_0) \geq h^i(m+1, 0)$. Without directly solving this dynamic programming problem based on relaxed solutions, we aim to reformulate this dynamic program to replace the constraint set in problem (3) with an equivalent longest path problem.

We define a weighted directed acyclic graph $\mathcal{G}^i = (\mathcal{V}^i, \mathcal{A}^i)$ to be the state transition graph of the dynamic program defined by equations (12)-(13), as illustrated in Figure 1. Formally, vertex set \mathcal{V}^i corresponds to the set of states \mathcal{S}^i and arc set \mathcal{A}^i corresponds to the set of all possible transitions between the states. Each state $(k, d) \in \mathcal{S}^i$ has an associated vertex in \mathcal{V}^i . To express the arcs between vertices, for each $(k, d) \in \mathcal{S}^i$,

1. If $k \in [m]$, for all $\mathbf{z}_k \in \mathcal{F}_{kd}$, create an arc from $(k-1, d - \delta_k \mathbb{1}[z_k \neq z_k^i])$ to (k, d) with a weight $-y^i \beta_{z_k}^\top \mathbf{z}_k$,

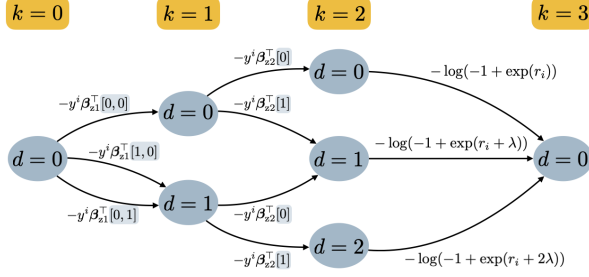


Figure 1: A graph example with only two categorical features processed by the one-hot encoding: $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2)$ where \mathbf{z}_1 has three possible realizations: $[1, 0], [0, 1], [0, 0]$ and \mathbf{z}_2 has two possible realizations: $[1], [0]$. Given a data point with index i : $\mathbf{z}^i = (\mathbf{z}_1^i, \mathbf{z}_2^i) = ([0, 0], [0])$. The weight parameters are $\boldsymbol{\delta} = (\delta_1, \delta_2) = (1, 1)$.

2. If $k = m + 1$, for each $(m, d') \in \mathcal{S}^i$, create an arc from (m, d') to $(m + 1, 0)$ with a weight $-\log(-1 + \exp(r_i + \lambda d'))$

We denote the arc weight function as $w^i : \mathcal{A}^i \rightarrow \mathbb{R}$. We also define the functions $s^i : \mathcal{A}^i \rightarrow \mathcal{V}^i$ and $t^i : \mathcal{A}^i \rightarrow \mathcal{V}^i$, which map each arc to its source and target vertices, respectively. Based on the connection between the dynamic programming problem and graph structure, as well as the connection between the dynamic programming problem and problem (11), we present the following lemma.

Lemma 1. *In each graph \mathcal{G}^i , the longest path from source $(0, 0)$ to sink $(m + 1, 0)$ corresponds to the optimal solution of problem (11) and the sum of the weights of this path equals its optimal objective value.*

The proof of Lemma 1 is provided in Appendix E.3. Using the connection in Lemma 1, we can reformulate problem (11) as a linear program that finds the longest path from the source $(0, 0)$ to the sink $(m + 1, 0)$ in \mathcal{G}^i . We can generate a graph-based reformulation.

Theorem 2 (Graph-based Formulation). *Problem (3) is equivalent to:*

$$\begin{aligned}
 \min_{\lambda, \mathbf{r}, \boldsymbol{\beta}, \boldsymbol{\mu}} \quad & \lambda \epsilon + \frac{1}{N} \sum_{i \in [N]} r_i \\
 \text{s.t.} \quad & \mathbf{y}^i (\boldsymbol{\beta}_x^\top \mathbf{x}^i + \beta_0) \geq -\mu_{(0,0)}^i + \mu_{(m+1,0)}^i, \forall i \in [N] \\
 & \mu_{t^i(e)}^i - \mu_{s^i(e)}^i \geq w^i(e), \forall i \in [N], e \in \mathcal{A}^i \\
 & |\gamma_j^{-1} \beta_{xj}| \leq \lambda, \forall j \in [n] \\
 & \lambda \geq 0, \mathbf{r} \in \mathbb{R}^N, \boldsymbol{\beta} \in \mathbb{R}^{1+n+c}, \boldsymbol{\mu} \in \mathbb{R}^{\sum_{i \in [N]} |\mathcal{V}^i|}
 \end{aligned} \tag{14}$$

where $\boldsymbol{\mu} = (\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N)$ are dual variables of the longest path problems and the first two sets of constraints correspond to the dual formulation.

The proof of Theorem 2 is given in Appendix E.4. Compared to problem (3), the size of this new formulation depends on the size of the dynamic programming digraphs. By selecting appropriate weight parameters δ_ℓ , multiple arcs from different vertices can converge at the same vertex, significantly reducing the number of arcs and vertices in the graph. For example, rounding the weight parameters to some decimal places can further reduce the size of the digraphs because it can reduce the number of unique weighted distances. Additionally, most constraints in this reformulation are linear, except for those representing arcs targeting sink vertices, which are also convex and compatible with off-the-shelf solvers.

Unlike cutting-plane methods, the graph-based formulation can be solved in polynomial time with respect to the digraph size, as there are N graphs for N data points. Additionally, it is independent of the number of iterations, which is often difficult to control in cutting-plane methods.

5 NUMERICAL RESULTS

We evaluate our methods on 13 UCI datasets with numerical and/or categorical features (Kelly et al.). The datasets vary in size, with the number of data points ranging from 132 to 12960, and the number of categorical features ranging from 3 to 62. All algorithms were implemented in Julia 1.8.5 (Bezanson et al., 2017) using the JuMP mathematical modeling language (Lubin et al., 2023) and executed on AMD epyc-7513 CPU in single-core mode. We use MOSEK 10 to solve all optimization problems (ApS, 2024). The code can be found at: https://github.com/QingshiSun/Robust_Logistic_Regression.

5.1 Runtime Comparison

We focus on runtime comparisons between the cutting-plane method proposed by Selvi et al. (2022), the cutting-plane method described in Section 4.1, and the direct solution of our proposed graph-based formulation as outlined in Section 4.2.

In our experiments, each model instance consists of a dataset, the ambiguity set parameters γ_j , δ_ℓ , and ϵ , as well as the rounding precision applied to the weight parameters. Following the ambiguity set calibration procedure outlined in Section 3, for all $j \in [n]$ and $\ell \in [m]$, we determine γ_j and δ_ℓ by sampling the probabilities of certainty ρ_{xj} and $\rho_{z\ell}$ from the same normal distribution in each model instance. Specifically, we consider normal distributions with means of 0.6, 0.7, 0.8, and 0.9, each with a standard deviation of 0.2. For each numerical feature j , the interval

Table 1: Mean runtime in seconds of the graph-based formulation (Graph), the cutting-plane method with the most violated constraint identified by dynamic programming (DP), and the cutting-plane method proposed by Selvi et al. (2022) (Cutting), under three settings: all weight parameters set to 1 (Weights = 1), calibrated weight parameters rounded to integers (Integer Weights), and calibrated weight parameters rounded to one decimal place (1 Decimal Weights). NaN in this table means that there is no numerical feature in the corresponding dataset. Following the notation in Section 2.1, N is the number of data points, n is the number of numerical features, m is the number of categorical features, and c is the number of categorical features after applying one-hot encoding.

Dataset	N	n	m	c	Weights = 1			Integer Weights		1 Decimal Weights	
					Graph	DP	Cutting	Graph	DP	Graph	DP
audiology	226	NaN	62	85	82.66	3880.84	825.73	142.24	4461.94	5391.81	45345.44
balance-scale	576	NaN	4	16	1.27	33.68	34.66	2.18	32.61	2.84	35.37
breast-cancer	286	NaN	9	32	2.43	28.61	16.02	5.63	43.81	30.84	142.66
car	1728	NaN	6	15	5.35	314.00	206.88	10.45	384.24	31.51	640.59
hayes-roth	132	NaN	3	9	0.11	1.69	1.55	0.16	1.74	0.18	1.77
tic-tac-toe	958	NaN	9	18	6.01	161.15	85.83	10.23	203.15	83.08	789.70
spect	267	NaN	22	22	7.76	47.36	12.66	9.49	51.72	141.57	473.47
voting	435	NaN	16	32	8.50	112.72	27.59	16.24	177.11	223.72	1253.95
credit-approval	690	6	9	36	6.63	193.02	69.64	13.94	256.21	77.12	704.77
cylinder	539	19	14	43	14.32	399.14	77.39	32.08	511.33	439.47	3557.80
hepatitis	155	6	13	23	1.62	15.41	6.09	3.52	21.83	37.50	120.09
nursery	12960	NaN	8	26	97.25	23874.78	11360.82	225.05	33132.95	1593.70	67219.20
online-shopper	12330	14	3	14	20.57	8685.08	8394.99	23.96	8919.10	33.48	9114.63

$[l_j, u_j]$ is set to $[-0.4\sigma_j, 0.4\sigma_j]$, where σ_j is the standard deviation of x_j in the training dataset. We also vary the level of robustness θ by selecting a value from $\{0.5, 0.65, 0.75, 0.8, 0.85, 0.9, 0.93, 0.96, 0.99\}$ for each model instance, which is then used to set the ambiguity set radius. As discussed in Section 4.2, the sizes of the dynamic programming problems and the corresponding graph-based formulation depend on the chosen weight parameters. We evaluate three rounding precision cases for the weight parameters: (i) all weights are set to one, disregarding differences in the distribution shift likelihood; (ii) calibrated weight parameters are rounded to integer; and (iii) calibrated weight parameters are rounded to one decimal place. In total, for the former case, we evaluate 117 instances, with 9 instances per dataset, while for each of the latter two cases, we evaluate 468 model instances, with 36 instances per dataset.

The average runtime results are presented in Table 1. From the table, it can be seen that our graph-based formulation has significant improvement in runtime in all instances. Specifically, the graph-based formulation can be solved up to 408.12 times faster on average compared to the cutting-plane method proposed by Selvi et al. (2022).

5.2 Performance under Distribution Shifts

To evaluate the robustness of our proposed model under distribution shifts, we generate 5,000 perturbed test sets for each instance. Each perturbed test set is created by independently perturbing the original test

data based on expected perturbations. Specifically, for a given set of sampled probabilities of certainty ρ_{xj} and $\rho_{z\ell}$, we generate the perturbed test sets using the probability density function (5) for numerical features and the probability mass function (6) for categorical features.

Additionally, we assess the robustness of our method against *unexpected* distribution shifts. We repeat the process of generating perturbed test sets for each instance but introduce unexpected perturbations by altering the values of ρ_{xj} and $\rho_{z\ell}$ used in our model. This evaluation is motivated by the fact that domain knowledge is sometimes uncertain and not precisely estimated. We test seven unexpected scenarios. We shift each ρ_{xj} and $\rho_{z\ell}$ value down by 0.2 and perturb the test data in 5,000 different ways based on these adjusted probability values. The same procedure is repeated with ρ_{xj} and $\rho_{z\ell}$ shifted down by 0.1 and up by 0.1. In addition, we uniformly sample new ρ_{xj} and $\rho_{z\ell}$ values for each feature within a neighborhood of radius 0.05 around the original ρ_{xj} and $\rho_{z\ell}$ respectively, and perturb the test data in 5,000 different ways using these new values. This procedure is also applied for neighborhood radii of 0.1, 0.15, and 0.2. We denote the modified probabilities used to generate unexpectedly perturbed test sets as $\tilde{\rho}_{xj}$ and $\tilde{\rho}_{z\ell}$.

We evaluate the performance of logistic regression using two common metrics: adaptive calibration error (Nixon et al., 2019) and AUC. These metrics are essential in high-stakes applications where predicted probabilities and their rankings, rather than just the final classifications, are of greater importance. For each

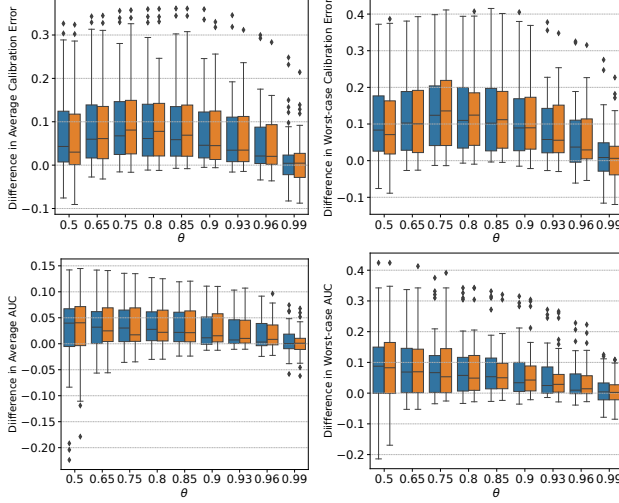


Figure 2: Performance improvement compared to lasso logistic regression in terms of calibration error and AUC across different levels of robustness θ under expected perturbations. Blue boxes: proposed models with weight parameters rounded to integer; orange boxes: with weight parameters rounded to one decimal.

model instance, we evaluate worst-case performance under expected perturbations by identifying the highest calibration error or lowest AUC across the 5,000 corresponding perturbed test sets for a given set of probability of certainty values ρ_{xj} and ρ_{zl} . The average performance is measured by computing the mean over 5000 perturbed test sets. Similarly, we assess worst-case and average performance under unexpected perturbations using the corresponding 5,000 perturbed test sets generated with $\hat{\rho}_{xj}$ and $\hat{\rho}_{zl}$.

We compare our proposed models, trained with rounding precisions to the nearest integer and one decimal place, against two existing methods: lasso regularized logistic regression and distributionally robust logistic regression with all weight parameters set to 1, as studied by Selvi et al. (2022). The candidate lasso regularization coefficients are $\{0, 5 \cdot 10^{-6}, 1 \cdot 10^{-5}, 5 \cdot 10^{-5}, 1 \cdot 10^{-4}, \dots, 0.5, 1, 10, 100, 1000\}$. The ambiguity set radius candidates for the model proposed by Selvi et al. (2022) are: $\{0, 10^{-5}, 10^{-4}, \dots, 0.1, 1\}$. Both regularization coefficient and ambiguity set radius are selected using 5-fold cross-validation. Neither our proposed models nor the benchmarks models have access to the perturbed test set in the training phase.

For each model instance, we compute the differences in worst-case and average-case performance between our proposed model and the benchmarks under identical expected and unexpected perturbations. A *positive* difference indicates that our model outperforms the corresponding benchmark. Figure 2 summarizes the

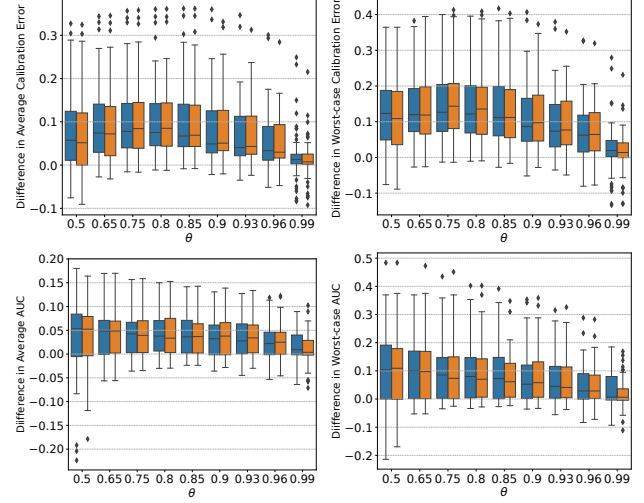


Figure 3: Performance improvement compared to distributionally robust logistic regression with all weight parameters set to 1 in terms of calibration error and AUC across different levels of robustness θ under expected perturbations. Blue boxes: proposed models with weight parameters rounded to integer; orange boxes: with weight parameters rounded to one decimal.

distribution of performance differences between our proposed model and lasso logistic regression across all tested instances under expected perturbations. Similarly, Figure 3 summarizes these differences relative to distributionally robust logistic regression with all weight parameters set to 1. Both figures demonstrate the consistent advantages of our model in terms of calibration error and AUC over a wide range of calibrated ambiguity set radii.

We observe that a highly small ambiguity set radius (e.g., $\theta = 0.99$) limits the model’s effectiveness against distribution shifts, while an overly large radius results in excessive conservatism and degraded performance. Additionally, rounding weight parameters to integers, compared to one-decimal precision, maintains robust performance and significantly reduces runtime, enabling scalability to large-scale applications.

The distributions of worst-case and average-case improvements of our proposed model compared to benchmarks under unexpected perturbations are provided in Appendix D. These results show the robustness of our model in the presence of model misspecification.

Acknowledgments

Phebe Vayanos and Qingshi Sun are funded in part by the National Science Foundation under CAREER award number 2046230. Nathan Justin is funded in part by the National Science Foundation Graduate Re-

search Fellowship Program (GRFP). Andrés Gómez is funded in part by AFOSR grant No. FA9550-24-1-0086. The authors are grateful for the support and thank the anonymous reviewers for their detailed and insightful comments.

References

- MOSEK ApS. *MOSEK Optimizer API for Julia*, 2024. URL <https://docs.mosek.com/latest/juliaapi/index.html>.
- Dimitris Bertsimas, Vishal Gupta, and Nathan Kallus. Data-driven robust optimization. *Mathematical Programming*, 167(2):235–292, 2018. doi: 10.1007/s10107-017-1125-8.
- Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017. doi: 10.1137/141000671. URL <https://epubs.siam.org/doi/10.1137/141000671>.
- Jose Blanchet, Yang Kang, Karthyek Murthy, and Fan Zhang. Data-driven optimal transport cost selection for distributionally robust optimization. In *2019 Winter Simulation Conference (WSC)*, pages 3740–3751, 2019. doi: 10.1109/WSC40007.2019.9004785.
- E. Delage and Y. Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3): 596–612, 2010.
- John C. Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378 – 1406, 2021. doi: 10.1214/20-AOS2004. URL <https://doi.org/10.1214/20-AOS2004>.
- João Gama, Indrundefin Žliobaitundefin, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4), March 2014. ISSN 0360-0300. doi: 10.1145/2523813. URL <https://doi.org/10.1145/2523813>.
- Rui Gao and Anton J. Kleywegt. Distributionally robust stochastic optimization with wasserstein distance, 2022. URL <https://arxiv.org/abs/1604.02199>.
- David W. Hosmer and Stanley Lemeshow. *Applied logistic regression*. John Wiley and Sons, 2000. ISBN 0471356328, 9780471356325.
- E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev.*, 106:620–630, May 1957. doi: 10.1103/PhysRev.106.620.
- Nathan Justin, Sina Aghaei, Andrés Gómez, and Phebe Vayanos. Learning optimal classification trees robust to distribution shifts, 2023. URL <https://arxiv.org/abs/2310.17772>.
- Markelle Kelly, Rachel Longjohn, and Kolby Nottingham. The uci machine learning repository. <https://archive.ics.uci.edu>.
- H. Lam. Recovering best statistical guarantees via the empirical divergence-based distributionally robust optimization. *Operations Research*, 6(4):1090–1105, 2019.
- Jiashuo Liu, Tianyu Wang, Peng Cui, and Hongseok Namkoong. Rethinking distribution shifts: Empirical analysis and inductive modeling for tabular data, 2024. URL <https://arxiv.org/abs/2307.05284>.
- Jie Lu, Anjin Liu, Fan Dong, Feng Gu, João Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363, 2019. doi: 10.1109/TKDE.2018.2876857.
- Miles Lubin, Oscar Dowson, Joaquim Dias Garcia, Joey Huchette, Benoît Legat, and Juan Pablo Vielma. JuMP 1.0: Recent improvements to a modeling language for mathematical optimization. *Mathematical Programming Computation*, 2023. doi: 10.1007/s12532-023-00239-3.
- P. Mohajerin Esfahani and D. Kuhn. Data-driven distributionally robust optimization using the Wasserstein metric: performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1–2):1–52, 2018.
- Jeremy Nixon, Michael W. Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. Measuring calibration in deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.
- Joaquin Quionero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence. *Dataset Shift in Machine Learning*. The MIT Press, 2009. ISBN 0262170051.
- E. Rice, N. Milburn, P. Vayanos, J. Rountree, C. Hill, R. Petering, B. Blackwell, R. Santillano, L. Onasch-Vera, H. Winetrobe, Nadel, B. Tang, S. Aghaei, HT. Hsu, and L. Petry. CESTTRR: Coordinated Entry System Triage Tool Research and Refinement. Technical report, Institution Name (if available), 2023.
- C Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat Mach Intell* 1, 206–215, 2019.
- Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles

and 10 grand challenges. *Statistics Surveys*, 16, 01 2022. doi: 10.1214/21-SS133.

Aras Selvi, Mohammad Reza Belbasi, Martin Haugh, and Wolfram Wiesemann. Wasserstein logistic regression with mixed features. *Advances in Neural Information Processing Systems*, 35:16691–16704, 2022.

S. Shafieezadeh-Abadeh, P. Mohajerin Esfahani, and D. Kuhn. Distributionally robust logistic regression. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Tasuku Soma, Khashayar Gatmiry, and Stefanie Jegelka. Optimal algorithms for group distributionally robust optimization and beyond, 2022. URL <https://arxiv.org/abs/2212.13669>.

Jingzhao Zhang, Aditya Krishna Menon, Andreas Veit, Srinadh Bhojanapalli, Sanjiv Kumar, and Survrit Sra. Coping with label shift via distributionally robust optimisation. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=BtZhsSGNRNi>.

Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 819–827, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL <https://proceedings.mlr.press/v28/zhang13d.html>.

Supplementary Materials

A DETAILS OF PARAMETER CALIBRATION

We propose a calibration method for the parameters in the ambiguity set (2), including the radius ϵ and the weight parameters γ_j and δ_ℓ , where $j \in [n]$ and $\ell \in [m]$. This method leverages the relationship between robust optimization (RO) and DRO. We first define a logistic regression model based on RO and establish a correspondence between the parameters in the uncertainty set of the robust logistic regression and those in the ambiguity set of our proposed distributionally robust logistic regression model. Subsequently, we extend the calibration method developed by Justin et al. (2023) for robust optimization from discrete to numerical features. The parameters in the ambiguity set are determined based on this correspondence and the calibrated parameters from the uncertainty set of robust optimization.

Compared to DRO, instead of constructing an ambiguity set containing a range of probability distributions and optimizing for the worst-case *distribution* within that set, RO aims to protect against worst-case scenarios by considering the worst possible *realization* of uncertain parameters within a given uncertainty set. Theoretically, Gao and Kleywegt (2022) proves that distributionally robust optimization problems can be approximated by robust optimization problems, and thereby some distributionally optimization problems can be processed by tools from robust optimization. In our setting, we define a logistic regression model based on robust optimization that accounts for the worst-case realization of perturbed features within an uncertainty set. For the training data, we define perturbed numerical features $\tilde{X} = [\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^N]^\top$ and perturbed categorical features $\tilde{Z} = [\tilde{z}^1, \tilde{z}^2, \dots, \tilde{z}^N]^\top$. We formulate a robust logistic regression problem without label perturbation

$$\underset{\beta \in \mathbb{R}^{1+n+c}}{\text{minimize}} \quad \underset{(\tilde{X}, \tilde{Z}) \in \mathcal{U}}{\text{maximize}} \quad \frac{1}{N} \sum_{i \in [N]} l_\beta(\tilde{x}^i, \tilde{z}^i, y^i), \quad (15)$$

where \mathcal{U} is the uncertainty set defined as

$$\mathcal{U} = \left\{ (\tilde{X}, \tilde{Z}) \mid \frac{1}{N} \sum_{i \in [N]} \left(\sum_{j \in [n]} \gamma'_j |\tilde{x}_j^i - x_j^i| + \sum_{\ell \in [m]} \delta'_\ell \mathbb{1}[\tilde{z}_\ell^i \neq z_\ell^i] \right) \leq \epsilon', \tilde{x}^i \in \mathbb{R}^n, \tilde{z}^i \in \mathcal{C}, i \in [N] \right\}, \quad (16)$$

where $\epsilon' > 0$ is the total allowable budget of uncertainty across data points, controlling the level of robustness for distribution shifts. $\gamma'_j > 0$ and $\delta'_\ell > 0$ are the weight parameters of the perturbation on numerical feature j and on categorical feature ℓ respectively. γ'_j is the cost of perturbing x_j^i by one in either direction, and δ'_ℓ is the cost of perturbing z_ℓ^i to a different value in the set \mathcal{C}_ℓ . Similar to our proposed distributionally robust logistic regression model, this robust logistic regression model also addresses sensitivity to feature shifts through weight parameters. In addition, although they differ in addressing distribution shifts—one focusing on worst-case realizations and the other on worst-case distributions—both models ultimately control the tolerance for distribution shifts using budget parameters.

We next build the specific connection between robust logistic regression problem (15) and our proposed distributionally robust logistic regression problem (1). That is, problem (1) can be transformed into the form of problem

(15) by introducing appropriate constraints. By incorporating the uncertainty set \mathcal{U} , the inner problem of the robust logistic regression problem (15) is equivalent to

$$\begin{aligned}
 & \underset{\tilde{\mathbf{X}}, \tilde{\mathbf{Z}}}{\text{maximize}} && \frac{1}{N} \sum_{i \in [N]} l_{\beta}(\tilde{\mathbf{x}}^i, \tilde{\mathbf{z}}^i, y^i) \\
 & \text{subject to} && \frac{1}{N} \sum_{i \in [N]} \left(\sum_{j \in [n]} \gamma_j |\tilde{x}_j^i - x_j^i| + \sum_{\ell \in [m]} \delta_{\ell} \mathbb{1}[\tilde{z}_{\ell}^i \neq z_{\ell}^i] \right) \leq \epsilon' \\
 & && (\tilde{\mathbf{x}}^i, \tilde{\mathbf{z}}^i) \in \mathbb{R}^n \times \mathcal{C}, \quad i \in [N].
 \end{aligned} \tag{17}$$

Following the proof of Theorem 1 by Selvi et al. (2022), the inner problem of our proposed distributionally robust logistic regression (1) can be reformulated as

$$\begin{aligned}
 & \underset{\mathbb{Q}^i}{\text{maximize}} && \frac{1}{N} \sum_{i \in [N]} \int_{\xi \in \Xi} l_{\beta}(\xi) \mathbb{Q}^i(d\xi) \\
 & \text{subject to} && \frac{1}{N} \sum_{i \in [N]} \int_{\xi \in \Xi} d(\xi, \xi^i) \mathbb{Q}^i(d\xi) \leq \epsilon \\
 & && \mathbb{Q}^i \in \mathcal{P}_0(\Xi), \quad i \in [N].
 \end{aligned}$$

where $\mathbb{Q}^i(d\xi) := \Pi(d\xi | \xi^i)$ the conditional distribution of Π upon the realization of $\xi' = \xi^i$. We add proper constraints on the selection of conditional probabilities to obtain a new formulation

$$\begin{aligned}
 & \underset{\mathbb{Q}^i}{\text{maximize}} && \frac{1}{N} \sum_{i \in [N]} \int_{\xi \in \Xi} l_{\beta}(\xi) \mathbb{Q}^i(d\xi) \\
 & \text{subject to} && \frac{1}{N} \sum_{i \in [N]} \int_{\xi \in \Xi} d(\xi, \xi^i) \mathbb{Q}^i(d\xi) \leq \epsilon \\
 & && \mathbb{Q}^i \in \mathcal{P}_0(\Xi), \quad i \in [N] \\
 & && \mathbb{Q}^i \in \{ \delta_{(\tilde{\mathbf{x}}^i, \tilde{\mathbf{z}}^i, y^i)} \mid (\tilde{\mathbf{x}}^i, \tilde{\mathbf{z}}^i) \in \mathbb{R}^n \times \mathcal{C}, \quad i \in [N] \},
 \end{aligned}$$

where $\delta_{(\tilde{\mathbf{x}}^i, \tilde{\mathbf{z}}^i, y^i)}$ is the Dirac delta function centered at $(\tilde{\mathbf{x}}^i, \tilde{\mathbf{z}}^i, y^i)$. Expressing $d(\xi, \xi^i)$ in its explicit form, the above formulation is equivalent to

$$\begin{aligned}
 & \underset{\tilde{\mathbf{X}}, \tilde{\mathbf{Z}}}{\text{maximize}} && \frac{1}{N} \sum_{i \in [N]} l_{\beta}(\tilde{\mathbf{x}}^i, \tilde{\mathbf{z}}^i, y^i) \\
 & \text{subject to} && \frac{1}{N} \sum_{i \in [N]} \left(\sum_{j \in [n]} \gamma_j |\tilde{x}_j^i - x_j^i| + \sum_{\ell \in [m]} \delta_{\ell} \mathbb{1}[\tilde{z}_{\ell}^i \neq z_{\ell}^i] \right) \leq \epsilon \\
 & && (\tilde{\mathbf{x}}^i, \tilde{\mathbf{z}}^i) \in \mathbb{R}^n \times \mathcal{C}, \quad i \in [N].
 \end{aligned} \tag{18}$$

Now, problem (18) is in the form of (17). The correspondence between parameters in the ambiguity set of the distributionally robust logistic regression model and parameters in the uncertainty set of the robust logistic regression problem model is $\gamma_j = \gamma'_j$, $\delta_j = \delta'_j$, and $\epsilon = \epsilon'$.

Next, we extend the calibration method for robust optimization problems studied by Justin et al. (2023) to the case with both numerical and categorical features. Based on the assumptions in Section 3, we assume the perturbation on numerical feature j follows a Laplace distribution f with scale parameter b_j and location parameter 0. Since the probability of certainty ρ_{xj} represents the probability that the perturbation on numerical feature j falling between l_j and u_j during deployment, for each data point indexed by i , we can express b_j through

$$\mathbb{P}(l_j \leq \tilde{x}_j^i - x_j^i \leq u_j) = \int_{-u_j + x_j^i}^{u_j + x_j^i} f(\tilde{x}_j^i) d\tilde{x}_j^i = \rho_{xj}.$$

We obtain $b_j = \frac{-u_j}{\log(1-\rho_{xj})}$.

We now build uncertainty sets using hypothesis testing (Bertsimas et al., 2018; Justin et al., 2023). We set up a likelihood ratio test on the magnitude of the perturbation with threshold θ^N , where the exponent N to normalize across different datasets with varying number of data points. Our null hypothesis is that the perturbed numerical and categorical features, \tilde{X} and \tilde{Z} come from the distributions described in Section 3. If this null hypothesis fails to be rejected, then \tilde{X} and \tilde{Z} lie within our uncertainty set. That is, \tilde{X} and \tilde{Z} lies within the uncertainty set if the following inequality

$$\frac{\prod_{i \in [N]} \left(\prod_{j \in [n]} \left(\frac{1}{2b_j} \exp \left(-\frac{|\tilde{x}_j^i - x_j^i|}{b_j} \right) \right) \cdot \prod_{\ell \in [m]} \left(\mathbb{1}[\tilde{z}_\ell^i = z_\ell^i] \rho_{z\ell} + \mathbb{1}[\tilde{z}_\ell^i \neq z_\ell^i] \frac{1-\rho_{z\ell}}{|\mathcal{C}_\ell|-1} \right) \right)}{\prod_{i \in [N]} \left(\prod_{j \in [n]} \frac{1}{2b_j} \cdot \prod_{\ell \in [m]} \rho_{z\ell} \right)} \geq \theta^N \quad (19)$$

is satisfied. The numerator of the left hand side of inequality (19) is the likelihood under the null hypothesis. The denominator of the left hand side is the likelihood of the most probable realization, meaning the likelihood of no perturbation.

By applying logarithm operations at both sides, we reduce inequality (19) to

$$\sum_{i \in [N]} \left(\sum_{j \in [n]} \frac{|\tilde{x}_j^i - x_j^i|}{b_j} \right) + \sum_{\ell \in [m]} \left(-\log \left(\mathbb{1}[\tilde{z}_\ell^i = z_\ell^i] + \mathbb{1}[\tilde{z}_\ell^i \neq z_\ell^i] \frac{1-\rho_{z\ell}}{\rho_{z\ell}(|\mathcal{C}_\ell|-1)} \right) \right) \leq -N \log \theta,$$

For categorical features, since when no perturbation exists, the cost is 0. We rewrite it as

$$\frac{1}{N} \left(\sum_{i \in [N]} \left(\sum_{j \in [n]} \frac{|\tilde{x}_j^i - x_j^i|}{b_j} \right) + \sum_{\ell \in [m]} \left(-\log \left(\frac{1-\rho_{z\ell}}{\rho_{z\ell}(|\mathcal{C}_\ell|-1)} \right) \mathbb{1}[\tilde{z}_\ell^i \neq z_\ell^i] \right) \right) \leq -\log \theta, \quad (20)$$

Since (20) is the form of uncertainty set (16), we can set

$$\delta'_\ell = \log \left(\frac{\rho_{z\ell}(|\mathcal{C}_\ell|-1)}{1-\rho_{z\ell}} \right),$$

$$\gamma'_j = \frac{-\log(1-\rho_{xj})}{u_j}$$

$$\epsilon' = -\log \theta$$

Based on the correspondence $\gamma_j = \gamma'_j$, $\delta_j = \delta'_j$, and $\epsilon = \epsilon'$, we set

$$\delta_\ell = \log \left(\frac{\rho_{z\ell}(|\mathcal{C}_\ell|-1)}{1-\rho_{z\ell}} \right),$$

$$\gamma_j = \frac{-\log(1-\rho_{xj})}{u_j}$$

$$\epsilon = -\log \theta$$

B CUTTING-PLANE SCHEME

We adapt the cutting-plane algorithm proposed by Selvi et al. (2022), as described in Algorithm 2. This algorithm can solve problem (3) in finite many iterations, following the proof of Theorem 4 by Selvi et al. (2022).

C DETAILS OF DATA PROCESSING

For each dataset, missing values in categorical features are treated as a separate category within that feature, while missing values in each numerical feature are replaced by the median of the remaining values of that feature.

Algorithm 2 Cutting-Plane Scheme for Problem (3).

Input: constraint set $\mathcal{W} \subseteq [N] \times \mathcal{C}$.
Output: optimal solution $(\lambda^*, \beta^*, \mathbf{r}^*)$ to problem (3)

- 1: **Initialize** $\text{LB}_0 = -\infty$ and $\text{UB}_0 = +\infty$ as lower and upper bounds for problem (3);
- 2: **while** $\text{LB}_t < \text{UB}_t$ **do**
- 3: Let e^* be the optimal value and $(\lambda, \beta, \mathbf{r})$ be the optimal solution to the relaxed version of (3), where the original constraint set $[N] \times \mathcal{C}$ is replaced by \mathcal{W} .
- 4: **for** $i \in [N]$ **do**
- 5: Identify the most violated constraint by Algorithm 1; get violation ϑ_i and corresponding solution $\mathbf{z}(i)$.
- 6: **end for**
- 7: Calculate $\vartheta^* = \max_{i \in [N]} \{\vartheta_i\}$ and $i^* = \operatorname{argmax}_{i \in [N]} \{\vartheta_i\}$.
- 8: Add the constraint indexed by $(i^*, \mathbf{z}(i^*))$ to \mathcal{W} .
- 9: Update $\text{LB}_t = e^*$ and $\text{UB}_t = \min\{\text{UB}_{t-1}, e^* + \vartheta^*\}$
- 10: Update $t = t + 1$
- 11: **end while**
- 12: **return** solution $(\lambda, \beta, \mathbf{r})$.

Features with only one unique category in the raw dataset are ignored. For datasets with non-binary labels, we convert the labels into binary by distinguishing the majority class from all other labels. Additionally, unrelated columns are removed based on dataset-specific properties. For example, the "name" and "hobby" columns in the "hayes-roth" dataset are deleted because they consist of randomly generated values. Additionally, we convert the text in certain columns to lowercase when the upper and lower case have the same meaning, to prevent the model from treating them as distinct categories.

D DISTRIBUTIONS OF PERFORMANCE IMPROVEMENTS UNDER UNEXPECTED PERTURBATIONS

The distributions of performance improvements for our proposed model under unexpected perturbations, compared to lasso logistic regression and the distributionally robust logistic regression model proposed by Selvi et al. (2022), are summarized in Figures 4 and 5, respectively. These results highlight our model's enhanced robustness and improved performance when the basic domain knowledge is not precisely estimated and our model is misspecified.

E PROOFS

E.1 Proof of Theorem 1

The proof of Theorem 1 requires the following lemma.

Lemma 2. Consider the convex function $g_{\beta}(\mathbf{x}) := \log(1 + \exp(-\beta_{\mathbf{x}}^{\top} \mathbf{x} - \alpha))$ where $\beta_{\mathbf{x}}, \mathbf{x} \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$. Then, for every $\lambda > 0$, the following

$$\sup_{\mathbf{x} \in \mathbb{R}^n} g_{\beta}(\mathbf{x}) - \lambda \|\Gamma_{\mathbf{x}}(\mathbf{x} - \hat{\mathbf{x}})\| = \begin{cases} g_{\beta}(\hat{\mathbf{x}}) & \text{if } \|\Gamma_{\mathbf{x}}^{-1} \beta_{\mathbf{x}}\|_* \leq \lambda, \\ +\infty & \text{otherwise} \end{cases}$$

holds, where $\|\cdot\|_*$ denotes the dual norm of $\|\cdot\|$, specifically

$$\|\Gamma_{\mathbf{x}}^{-1} \beta_{\mathbf{x}}\|_* := \sup_{\|\mathbf{x}\| \leq 1} (\Gamma_{\mathbf{x}}^{-1} \beta_{\mathbf{x}})^{\top} \mathbf{x},$$

and $\Gamma_{\mathbf{x}} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with positive diagonal elements $\gamma_j > 0$ for $j \in [n]$.

Lemma 2 generalizes Lemma 1 of Shafieezadeh-Abadeh et al. (2015) by accounting for the weight parameter γ_j for each numerical feature $j \in [n]$ and the constant α in the log-loss function.

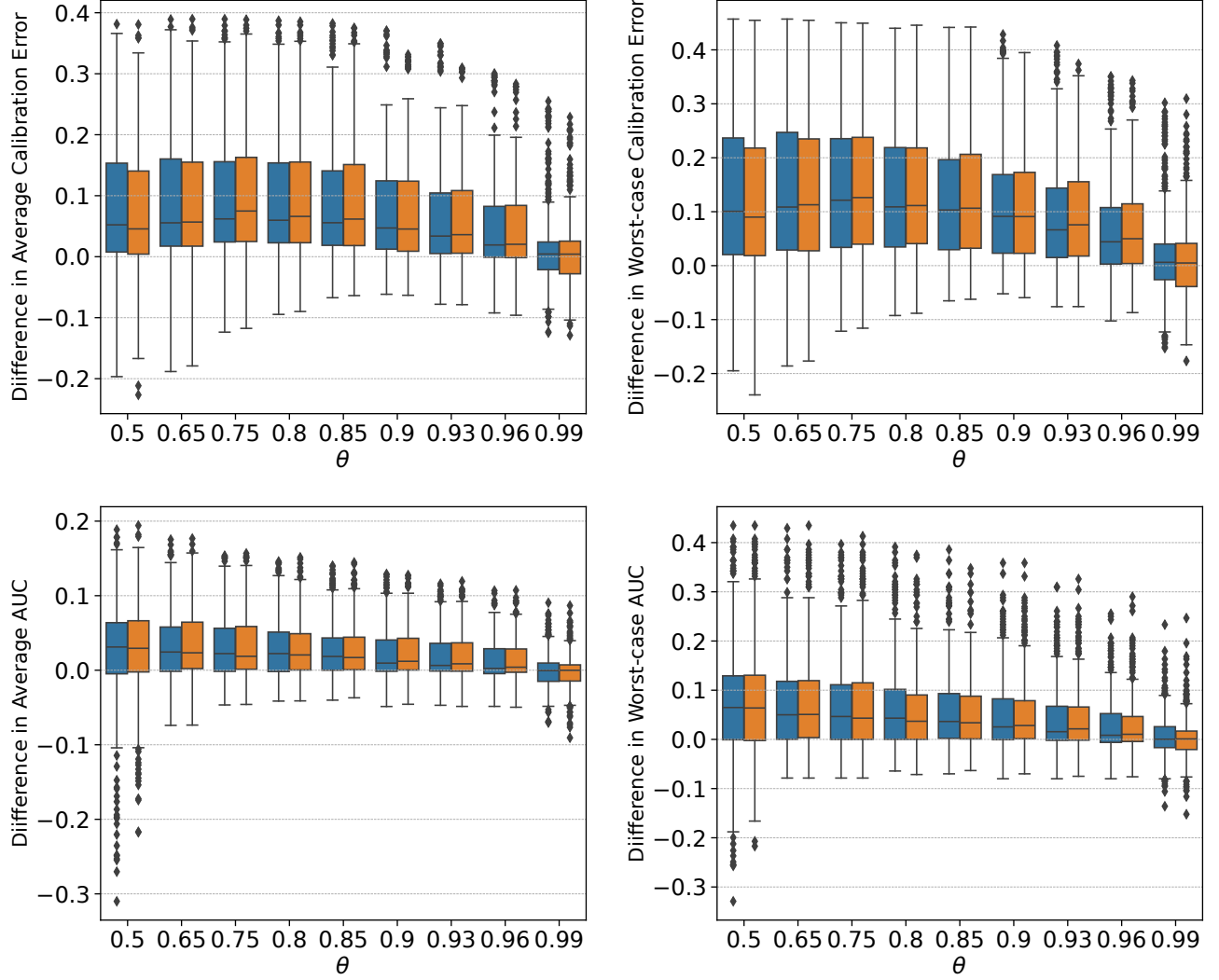


Figure 4: Overview of performance improvement compared to lasso-regularized logistic regression in terms of calibration error and AUC across different levels of robustness θ under *unexpected* perturbations. The blue boxes represent our proposed models with calibrated parameters rounded to integer. The orange boxes represent our proposed models with calibrated parameters rounded to one decimal place.

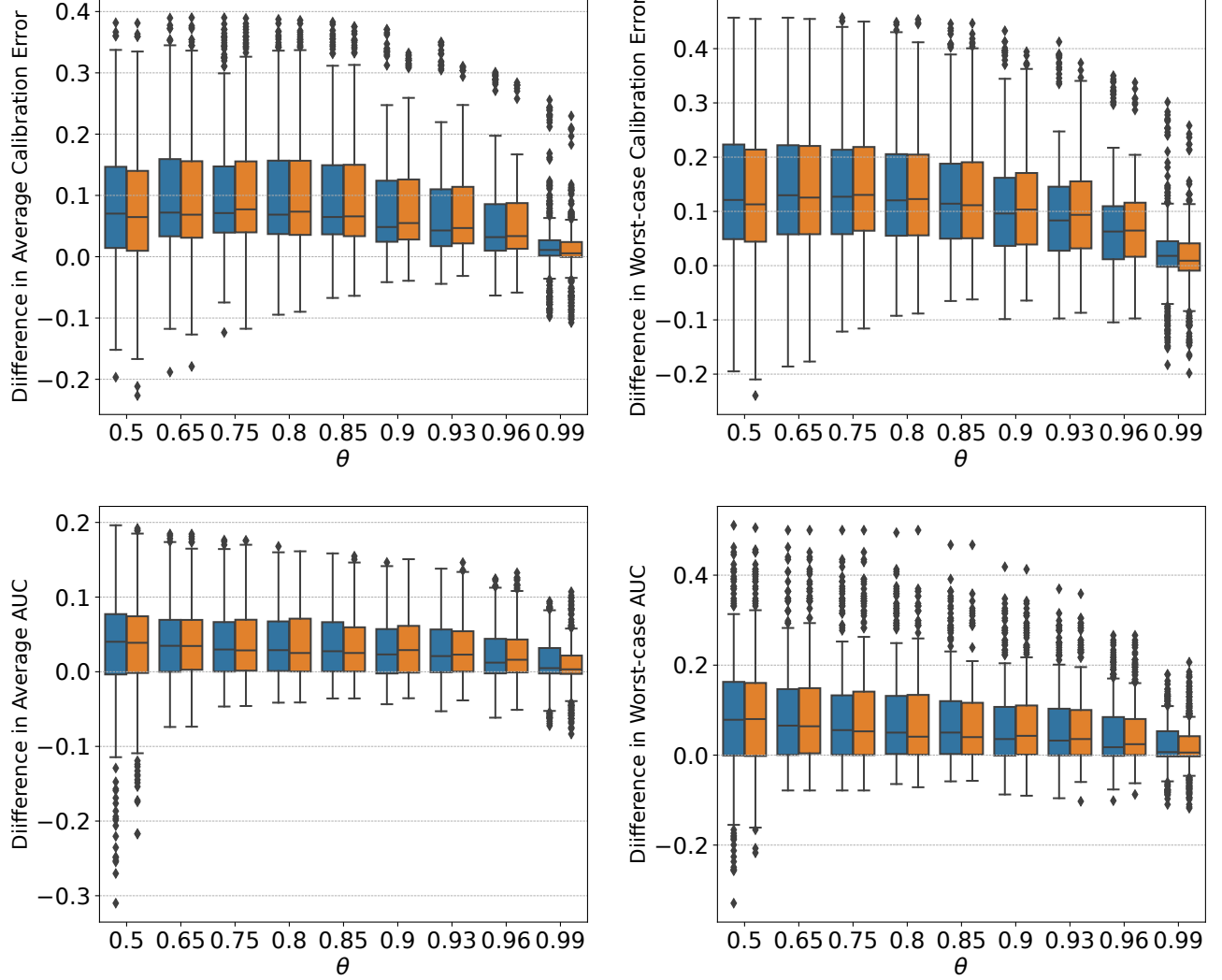


Figure 5: Overview of performance improvement compared to distributionally robust logistic regression with all weight parameters set to 1 in terms of calibration error and AUC across different levels of robustness θ under *unexpected* perturbations. The blue boxes represent our proposed models with calibrated parameters rounded to integer. The orange boxes represent our proposed models with calibrated parameters rounded to one decimal place.

Proof of Lemma 2. To process the constant α , we define $\mathbf{x}' = \mathbf{x} + \mathbf{c}$ where $\mathbf{c} \in \mathbb{R}^n$ such that $\mathbf{c}^\top \beta_{\mathbf{x}} = \alpha$ and focus on showing that the following

$$\sup_{\mathbf{x}' \in \mathbb{R}^n} h_{\beta}(\mathbf{x}') - \lambda \|\Gamma_x(\mathbf{x}' - \hat{\mathbf{x}})\| = \begin{cases} h_{\beta}(\hat{\mathbf{x}}) & \text{if } \|\Gamma_x^{-1} \beta_{\mathbf{x}}\|_* \leq \lambda, \\ +\infty & \text{otherwise} \end{cases} \quad (21)$$

holds where $h_{\beta}(\mathbf{x}') := \log(1 + \exp(-\beta_{\mathbf{x}}^\top \mathbf{x}'))$

Following the proof of Lemma 1 of Shafieezadeh-Abadeh et al. (2015), we express $h_{\beta}(\mathbf{x}') - \lambda \|\Gamma_x(\mathbf{x}' - \hat{\mathbf{x}})\|$ as an upper envelop of infinitely many affine functions. We define $f(t) := \log(1 + \exp(-t))$ and its conjugate function of $f(t)$ is

$$f^*(\tau) = \begin{cases} \tau \log(\tau) + (1 - \tau) \log(1 - \tau) & \text{if } \tau \in [0, 1], \\ +\infty & \text{otherwise.} \end{cases}$$

Based on strong Lagrangian duality, the conjugate of $h_{\beta}(\mathbf{x}') = f(\beta_{\mathbf{x}}^\top \mathbf{x}')$ is

$$h_{\beta}^*(\zeta) = \begin{cases} \inf_{0 \leq \tau \leq 1} f^*(\tau) & \text{if } \zeta = \tau \beta_{\mathbf{x}}, \\ +\infty & \text{otherwise.} \end{cases}$$

As the logloss function $h_{\beta}(\mathbf{x}')$ is convex and continuous, it equals its bi-conjugate:

$$\begin{aligned} h_{\beta}(\mathbf{x}') &= h_{\beta}^{**}(\mathbf{x}') = \sup_{\zeta \in \mathbb{R}^n} \langle \zeta, \mathbf{x}' \rangle - h_{\beta}^*(\zeta) \\ &= \sup_{0 \leq \tau \leq 1} \langle \tau \beta_{\mathbf{x}}, \mathbf{x}' \rangle - f^*(\tau). \end{aligned}$$

Using this representation, we obtain

$$\begin{aligned} \sup_{\mathbf{x}' \in \mathbb{R}^n} h_{\beta}(\mathbf{x}') - \lambda \|\Gamma_x(\mathbf{x}' - \hat{\mathbf{x}})\| &= \sup_{\mathbf{x}' \in \mathbb{R}^n} h_{\beta}^{**}(\mathbf{x}') - \lambda \|\Gamma_x(\mathbf{x}' - \hat{\mathbf{x}})\| \\ &= \sup_{0 \leq \tau \leq 1} \sup_{\mathbf{x}' \in \mathbb{R}^n} \langle \tau \beta_{\mathbf{x}}, \mathbf{x}' \rangle - f^*(\tau) - \lambda \|\Gamma_x(\mathbf{x}' - \hat{\mathbf{x}})\| \\ &= \sup_{0 \leq \tau \leq 1} \sup_{\mathbf{x}' \in \mathbb{R}^n} \langle \tau \beta_{\mathbf{x}}, \mathbf{x}' \rangle - f^*(\tau) - \sup_{\|q\|_* \leq \lambda} \langle q, \Gamma_x(\mathbf{x}' - \hat{\mathbf{x}}) \rangle \\ &= \sup_{0 \leq \tau \leq 1} \sup_{\mathbf{x}' \in \mathbb{R}^n} \langle \tau \beta_{\mathbf{x}}, \mathbf{x}' \rangle - f^*(\tau) - \sup_{\|q\|_* \leq \lambda} \langle \Gamma_x q, \mathbf{x}' - \hat{\mathbf{x}} \rangle \\ &= \sup_{0 \leq \tau \leq 1} \sup_{\mathbf{x}' \in \mathbb{R}^n} \inf_{\|q\|_* \leq \lambda} \langle \tau \beta_{\mathbf{x}}, \mathbf{x}' \rangle - f^*(\tau) - \langle \Gamma_x q, \mathbf{x}' - \hat{\mathbf{x}} \rangle \\ &= \sup_{0 \leq \tau \leq 1} \inf_{\|q\|_* \leq \lambda} \sup_{\mathbf{x}' \in \mathbb{R}^n} \langle \tau \beta_{\mathbf{x}} - \Gamma_x q, \mathbf{x}' \rangle - f^*(\tau) + \langle \Gamma_x q, \hat{\mathbf{x}} \rangle, \end{aligned}$$

where the third equality follows from the definition of the dual norm. Explicitly evaluating the maximization over \mathbf{x}' shows that the above expression is equivalent to

$$\begin{aligned} &\begin{cases} \sup_{0 \leq \tau \leq 1} \inf_{\|q\|_* \leq \lambda} -f^*(\tau) + \langle \Gamma_x q, \hat{\mathbf{x}} \rangle \\ \text{s.t.} \quad \tau \beta_{\mathbf{x}} - \Gamma_x q = 0 \end{cases} \\ &= \begin{cases} \sup_{0 \leq \tau \leq 1} -f^*(\tau) + \langle \tau \beta_{\mathbf{x}}, \hat{\mathbf{x}} \rangle & \text{if } \sup_{0 \leq \tau \leq 1} \|\tau \Gamma_x^{-1} \beta_{\mathbf{x}}\|_* \leq \lambda, \\ +\infty & \text{otherwise.} \end{cases} \end{aligned}$$

We conclude that equation (21) holds. By expressing \mathbf{x}' in terms of $\mathbf{x} + \mathbf{c}$, we have

$$\sup_{\mathbf{x} \in \mathbb{R}^n} g_{\beta}(\mathbf{x}) - \lambda \|\Gamma_x(\mathbf{x} - \hat{\mathbf{x}})\| = \begin{cases} g_{\beta}(\hat{\mathbf{x}}) & \text{if } \|\Gamma_x^{-1} \beta_{\mathbf{x}}\|_* \leq \lambda, \\ +\infty & \text{otherwise.} \end{cases}$$

Therefore, Lemma 2 holds. \square

Proof of Theorem 1. The theorem can be proven along the lines of the proof of Theorem 1 by Selvi et al. (2022) if we leverage Lemma 2. Problem (1) can be reformulated as

$$\begin{aligned}
 & \underset{\lambda, \mathbf{r}}{\text{minimize}} \quad \lambda\epsilon + \frac{1}{N} \sum_{i=1}^N r_i \\
 & \text{subject to} \quad \sup_{\mathbf{z} \in \mathbb{R}^n} \left\{ l_{\beta}(\mathbf{x}, \mathbf{z}, +1) - \lambda \sum_{j \in [n]} \gamma_j |x_j - x_j^i| \right\} - \lambda \kappa \mathbb{1}[y^i \neq 1] - \lambda \sum_{\ell \in [m]} \delta_{\ell} \mathbb{1}[\mathbf{z}_{\ell} \neq \mathbf{z}_{\ell}^i] \leq r_i, \quad \forall i \in [N], \mathbf{z} \in \mathcal{C} \\
 & \quad \sup_{\mathbf{z} \in \mathbb{R}^n} \left\{ l_{\beta}(\mathbf{x}, \mathbf{z}, -1) - \lambda \sum_{j \in [n]} \gamma_j |x_j - x_j^i| \right\} - \lambda \kappa \mathbb{1}[y^i \neq -1] - \lambda \sum_{\ell \in [m]} \delta_{\ell} \mathbb{1}[\mathbf{z}_{\ell} \neq \mathbf{z}_{\ell}^i] \leq r_i, \quad \forall i \in [N], \mathbf{z} \in \mathcal{C} \\
 & \quad \lambda \geq 0, \quad \mathbf{r} \in \mathbb{R}^N
 \end{aligned}$$

The dual norm of l_1 norm is l_{∞} norm. Applying Lemma 2 to the suprema of the above problem and choosing l_1 norm, we get

$$\begin{aligned}
 & \underset{\lambda, \mathbf{r}}{\text{minimize}} \quad \lambda\epsilon + \frac{1}{N} \sum_{i=1}^N r_i \\
 & \text{subject to} \quad l_{\beta}(\mathbf{x}^i, \mathbf{z}, +1) - \lambda \kappa \cdot \mathbb{1}[y^i \neq 1] - \lambda \sum_{\ell \in [m]} \delta_{\ell} \mathbb{1}[\mathbf{z}_{\ell} \neq \mathbf{z}_{\ell}^i] \leq r_i, \quad \forall i \in [N], \mathbf{z} \in \mathcal{C} \\
 & \quad l_{\beta}(\mathbf{x}^i, \mathbf{z}, -1) - \lambda \kappa \cdot \mathbb{1}[y^i \neq -1] - \lambda \sum_{\ell \in [m]} \delta_{\ell} \mathbb{1}[\mathbf{z}_{\ell} \neq \mathbf{z}_{\ell}^i] \leq r_i, \quad \forall i \in [N], \mathbf{z} \in \mathcal{C} \\
 & \quad \max_{j \in [n]} |\gamma_j^{-1} \beta_{xj}| \leq \lambda \\
 & \quad \lambda \geq 0, \quad \mathbf{r} \in \mathbb{R}^N.
 \end{aligned}$$

Based on the property of maximization problem, this formulation is equivalent to

$$\begin{aligned}
 & \underset{\lambda, \mathbf{r}}{\text{minimize}} \quad \lambda\epsilon + \frac{1}{N} \sum_{i=1}^N r_i \\
 & \text{subject to} \quad l_{\beta}(\mathbf{x}^i, \mathbf{z}, +1) - \lambda \kappa \cdot \mathbb{1}[y^i \neq 1] - \lambda \sum_{\ell \in [m]} \delta_{\ell} \mathbb{1}[\mathbf{z}_{\ell} \neq \mathbf{z}_{\ell}^i] \leq r_i, \quad \forall i \in [N], \mathbf{z} \in \mathcal{C} \\
 & \quad l_{\beta}(\mathbf{x}^i, \mathbf{z}, -1) - \lambda \kappa \cdot \mathbb{1}[y^i \neq -1] - \lambda \sum_{\ell \in [m]} \delta_{\ell} \mathbb{1}[\mathbf{z}_{\ell} \neq \mathbf{z}_{\ell}^i] \leq r_i, \quad \forall i \in [N], \mathbf{z} \in \mathcal{C} \\
 & \quad |\gamma_j^{-1} \beta_{xj}| \leq \lambda, \quad \forall j \in [n] \\
 & \quad \lambda \geq 0, \quad \mathbf{r} \in \mathbb{R}^N.
 \end{aligned}$$

For $y^i = +1$, the identities

$$\begin{aligned}
 l_{\beta}(\mathbf{x}^i, \mathbf{z}, +1) - \lambda \kappa \cdot \mathbb{1}[y^i \neq 1] &= l_{\beta}(\mathbf{x}^i, \mathbf{z}, y^i) \\
 l_{\beta}(\mathbf{x}^i, \mathbf{z}, -1) - \lambda \kappa \cdot \mathbb{1}[y^i \neq -1] &= l_{\beta}(\mathbf{x}^i, \mathbf{z}, -y^i) - \lambda \kappa
 \end{aligned}$$

hold; for $y^i = -1$, the identities

$$\begin{aligned}
 l_{\beta}(\mathbf{x}^i, \mathbf{z}, +1) - \lambda \kappa \cdot \mathbb{1}[y^i \neq 1] &= l_{\beta}(\mathbf{x}^i, \mathbf{z}, -y^i) - \lambda \kappa \\
 l_{\beta}(\mathbf{x}^i, \mathbf{z}, -1) - \lambda \kappa \cdot \mathbb{1}[y^i \neq -1] &= l_{\beta}(\mathbf{x}^i, \mathbf{z}, y^i)
 \end{aligned}$$

hold as well. we can simplify above formulation to

$$\begin{aligned}
 & \underset{\lambda, \mathbf{r}}{\text{minimize}} && \lambda\epsilon + \frac{1}{N} \sum_{i=1}^N r_i \\
 & \text{subject to} && l_{\beta}(\mathbf{x}^i, \mathbf{z}, y^i) - \lambda \sum_{\ell \in [m]} \delta_{\ell} \mathbb{1}[\mathbf{z}_{\ell} \neq \mathbf{z}_{\ell}^i] \leq r_i, \quad \forall i \in [N], \mathbf{z} \in \mathcal{C} \\
 & && l_{\beta}(\mathbf{x}^i, \mathbf{z}, -y^i) - \lambda\kappa - \lambda \sum_{\ell \in [m]} \delta_{\ell} \mathbb{1}[\mathbf{z}_{\ell} \neq \mathbf{z}_{\ell}^i] \leq r_i, \quad \forall i \in [N], \mathbf{z} \in \mathcal{C} \\
 & && |\gamma_j^{-1} \beta_{xj}| \leq \lambda, \quad \forall j \in [n] \\
 & && \lambda \geq 0, \quad \mathbf{r} \in \mathbb{R}^N
 \end{aligned}$$

Since we do not account for perturbations in the labels, we set κ to infinity, causing the second set of constraints to hold trivially. Removing these constraints and incorporating this problem into the overall optimization problem, we can get problem (3)

$$\begin{aligned}
 & \underset{\lambda, \mathbf{r}, \beta}{\text{minimize}} && \lambda\epsilon + \frac{1}{N} \sum_{i=1}^N r_i \\
 & \text{subject to} && l_{\beta}(\mathbf{x}^i, \mathbf{z}, y^i) - \lambda \sum_{\ell \in [m]} \delta_{\ell} \mathbb{1}[\mathbf{z}_{\ell} \neq \mathbf{z}_{\ell}^i] \leq r_i, \quad \forall i \in [N], \mathbf{z} \in \mathcal{C} \\
 & && |\gamma_j^{-1} \beta_{xj}| \leq \lambda, \quad \forall j \in [n] \\
 & && \lambda \geq 0, \quad \mathbf{r} \in \mathbb{R}^N, \quad \beta = (\beta_0, \beta_x, \beta_z) \in \mathbb{R}^{1+n+c}
 \end{aligned}$$

□

Equivalent formulation of problem (3) as an exponential cone problem. We can convert a logarithm constraint $\log(1 + \exp(c)) \leq a$ into $\exp(-a) + \exp(c - a) \leq 1$ through exponentiation operations. By introducing the auxiliary variables u, v , we can reformulate this constraint as

$$\begin{aligned}
 u + v &\leq 1 \\
 \exp(-a) &\leq u \\
 \exp(c - a) &\leq v
 \end{aligned} \tag{22}$$

The exponential cone is defined as

$$\mathcal{K}_{\text{exp}} := \text{cl}(\{(a, b, c) : a \geq b \exp(c/b), a > 0, b > 0\}) \subset \mathbb{R}^3,$$

where $\text{cl}(\cdot)$ denotes the closure. Therefore, we can rewrite constraints (22) as

$$\begin{aligned}
 u + v &\leq 1 \\
 (u, 1, -a) &\in \mathcal{K}_{\text{exp}} \\
 (v, 1, c - a) &\in \mathcal{K}_{\text{exp}}
 \end{aligned}$$

For each $(i, \mathbf{z}) \in [N] \times \mathcal{C}$, the constraint in problem (3)

$$\log(1 + \exp(-y^i (\beta_x^{\top} \mathbf{x}^i + \beta_z^{\top} \mathbf{z}))) - \lambda \sum_{\ell \in [m]} \delta_{\ell} \mathbb{1}[\mathbf{z}_{\ell} \neq \mathbf{z}_{\ell}^i] \leq r_i$$

can be converted into

$$\begin{aligned}
 u + v &\leq 1 \\
 (u, 1, -\lambda \sum_{\ell \in [m]} \delta_{\ell} \mathbb{1}[\mathbf{z}_{\ell} \neq \mathbf{z}_{\ell}^i] - r_i) &\in \mathcal{K}_{\text{exp}} \\
 (v, 1, -y^i (\beta_x^{\top} \mathbf{x}^i + \beta_z^{\top} \mathbf{z}) - \lambda \sum_{\ell \in [m]} \delta_{\ell} \mathbb{1}[\mathbf{z}_{\ell} \neq \mathbf{z}_{\ell}^i] - r_i) &\in \mathcal{K}_{\text{exp}}
 \end{aligned}$$

We can formulate problem (3) in the form of exponential cone problem

$$\begin{aligned}
 & \underset{\lambda, \mathbf{r}, \boldsymbol{\beta}, \mathbf{u}, \mathbf{v}}{\text{minimize}} && \lambda \epsilon + \frac{1}{N} \sum_{i \in [N]} r_i \\
 & \text{subject to} && \left. \begin{aligned}
 & u_{i\mathbf{z}} + v_{i\mathbf{z}} \leq 1 \\
 & (u_{i\mathbf{z}}, 1, -\lambda \sum_{\ell \in [m]} \delta_\ell \mathbb{1}[\mathbf{z}_\ell \neq \mathbf{z}_\ell^i] - r_i) \in \mathcal{K}_{\text{exp}} \\
 & (v_{i\mathbf{z}}, 1, -y^i (\boldsymbol{\beta}_x^\top \mathbf{x}^i + \boldsymbol{\beta}_z^\top \mathbf{z}) - \lambda \sum_{\ell \in [m]} \delta_\ell \mathbb{1}[\mathbf{z}_\ell \neq \mathbf{z}_\ell^i] - r_i) \in \mathcal{K}_{\text{exp}} \\
 & |\gamma_j^{-1} \beta_{xj}| \leq \lambda, \quad \forall j \in [n] \\
 & \lambda \geq 0, \mathbf{r} \in \mathbb{R}^N, \boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}_x, \boldsymbol{\beta}_z) \in \mathbb{R}^{1+n+c} \\
 & (u_{i\mathbf{z}}, v_{i\mathbf{z}}) \in \mathbb{R}^2, (i, \mathbf{z}) \in [N] \times \mathcal{C}
 \end{aligned} \right\} \forall i \in [N], \mathbf{z} \in \mathcal{C}
 \end{aligned}$$

E.2 Proof of Correctness of Algorithm 1

Proof. We prove the correctness of Algorithm 1 using mathematical induction, showing that we can calculate $g^i(k, d)$ correctly for each $(k, d) \in \mathcal{S}^i$ by equations (9)-(10).

Base Case:

For $k = 0$, we have $g^i(0, 0) = 0$. This is given directly by the definition, and since there are no categorical features to consider, the base case holds trivially.

Inductive Step:

For each $k \in [m]$, assume that the value $g^i(k-1, d')$ has been correctly calculated for each $(k-1, d') \in \mathcal{S}^i$. Then, for each state $(k, d) \in \mathcal{S}^i$, the function $g^i(k, d)$ is defined as (8), which is:

$$\begin{aligned}
 g^i(k, d) &:= \max_{\{\mathbf{z}_\ell\}_{\ell \in [k]}} -y^i \sum_{\ell \in [k]} \boldsymbol{\beta}_{z\ell}^\top \mathbf{z}_\ell \\
 &\text{s.t.} \quad \mathbf{z}_\ell \in \mathcal{C}_\ell, \ell \in [k] \\
 &\quad \sum_{\ell \in [k]} \delta_\ell \mathbb{1}[\mathbf{z}_\ell \neq \mathbf{z}_\ell^i] = d
 \end{aligned}$$

We can decompose this optimization problem by dividing decision variables $\{\mathbf{z}_\ell\}_{\ell \in [k]}$ into \mathbf{z}_k and $\{\mathbf{z}_\ell\}_{\ell \in [k-1]}$:

$$\begin{aligned}
 g^i(k, d) &= \max_{\mathbf{z}_k \in \mathcal{C}_k} -y^i \boldsymbol{\beta}_{zk}^\top \mathbf{z}_k + \max_{\{\mathbf{z}_\ell\}_{\ell \in [k-1]}} \left[-y^i \sum_{\ell \in [k-1]} \boldsymbol{\beta}_{z\ell}^\top \mathbf{z}_\ell \right] \\
 &\text{s.t.} \quad (k-1, d - \delta_k \mathbb{1}[\mathbf{z}_k \neq \mathbf{z}_k^i]) \in \mathcal{S}^i \\
 &\quad \mathbf{z}_\ell \in \mathcal{C}_\ell, \ell \in [k-1] \\
 &\quad \sum_{\ell \in [k-1]} \delta_\ell \mathbb{1}[\mathbf{z}_\ell \neq \mathbf{z}_\ell^i] = d - \delta_k \mathbb{1}[\mathbf{z}_k \neq \mathbf{z}_k^i]
 \end{aligned}$$

The first constraint ensures that \mathbf{z}_k is feasible for subproblem (8) at state (k, d) . The second and the third constraints define the feasible region for subproblem (8) at state $(k-1, d - \delta_k \mathbb{1}[\mathbf{z}_k \neq \mathbf{z}_k^i])$ when choosing $\mathbf{z}_k \in \mathcal{C}_k$. By the inductive hypothesis, the inner maximization over $\{\mathbf{z}_\ell\}_{\ell \in [k-1]}$ yields $g^i(k-1, d - \delta_k \mathbb{1}[\mathbf{z}_k \neq \mathbf{z}_k^i])$. Therefore:

$$g^i(k, d) = \max_{\mathbf{z}_k \in \mathcal{F}_{kd}} \{ -y^i \boldsymbol{\beta}_{zk}^\top \mathbf{z}_k + g^i(k-1, d - \delta_k \mathbb{1}[\mathbf{z}_k \neq \mathbf{z}_k^i]) \}$$

where

$$\mathcal{F}_{kd} = \{ \mathbf{z}_k \in \mathcal{C}_k \mid (k-1, d - \delta_k \mathbb{1}[\mathbf{z}_k \neq \mathbf{z}_k^i]) \in \mathcal{S}^i \}.$$

This is exactly the Bellman equation (9).

For $k = m + 1$, we must account for the numerical feature component, the logistic function, and current solution to the relaxed problem of (3) that are not captured in previous subproblems. This ensures that $g^i(m + 1, 0)$ represents the optimal objective value of problem (7). We can compare the constraint violations of the optimal \mathbf{z} for subproblem (8) across all $(m, d) \in S_1^i$ and select the one with the largest violation as the most violated constraint. Using the values of $g^i(m, d)$, we have:

$$g^i(m + 1, 0) = \max_d \log(1 + \exp(-y^i \beta_x^\top \mathbf{x}^i + g^i(m, d))) - \lambda d - r_i$$

$$\text{s.t. } (m, d) \in S_1^i$$

This aligns with equation (10), allowing us to compute $g^i(m + 1, 0)$ using the previously calculated $g^i(m, d)$. Therefore, we can calculate $g^i(k, d)$ for each $(k, d) \in S^i$ by equations (9)-(10). \square

E.3 Proof of Lemma 1

Proof. For clarity in the following proof, we explicitly restate the dynamic programming subproblems used to solve problem (11), along with the function h^i , which represents their optimal objective values, as defined in Section 4.2. This restatement is provided without introducing new definitions.

If $k = 0$, for state $(0, 0)$,

$$h^i(0, 0) := 0. \quad (23)$$

If $k \in [m]$, for each state $(k, d) \in S_1^i$,

$$h^i(k, d) := \max_{\{\mathbf{z}_\ell\}_{\ell \in [k]}} -y^i \sum_{\ell \in [k]} \beta_{z_\ell}^\top \mathbf{z}_\ell$$

$$\text{s.t. } \mathbf{z}_\ell \in \mathcal{C}_\ell, \ell \in [k]$$

$$\sum_{\ell \in [k]} \delta_\ell \mathbb{1}[\mathbf{z}_\ell \neq \mathbf{z}_\ell^i] = d. \quad (24)$$

If $k = m + 1$, for state $(m + 1, 0)$,

$$h^i(m + 1, 0) := \max_{\mathbf{z} \in \mathcal{C}} -y^i \beta_z^\top \mathbf{z} - \log\left(-1 + \exp\left(r_i + \lambda \sum_{\ell \in [m]} \delta_\ell \mathbb{1}[\mathbf{z}_\ell \neq \mathbf{z}_\ell^i]\right)\right). \quad (25)$$

These subproblems can be solved by Bellman equations (12)-(13). The proof follows the proof of Algorithm 1 and is omitted for the sake of brevity.

Next, having established the equivalence between problem (11) and the dynamic programming approach, we proceed to link the dynamic programming method with the longest path problem in the corresponding graph. Due to the correspondence between the vertex set \mathcal{V}^i and the state space S^i for each data point i , we assert that in each graph \mathcal{G}^i , the longest path from the source $(0, 0)$ to any vertex $(k, d) \in \mathcal{V}^i$ corresponds to the optimal solution of subproblems (23)-(25), and the sum of the weights along this path equals $h^i(k, d)$. We prove this via mathematical induction.

Based on the graph structure, all vertices with the same $k = 0, 1, \dots, m + 1$ are grouped into the same layer k . For each vertex $(k, d) \in \mathcal{V}^i$, where $k = 1, \dots, m + 1$, there is at least one arc targeting (k, d) from a vertex in the previous layer $k - 1$. Thus, for any vertex $(k, d) \in \mathcal{V}^i \setminus \{(0, 0)\}$, there always exists a path from source $(0, 0)$ to (k, d) .

For $k \in [m]$, since every arc is from a vertex in layer $k - 1$ to a vertex in layer k and corresponds to a categorical feature realization $\mathbf{z}_k \in \mathcal{C}_k$, any path from the source $(0, 0)$ to a vertex $(k, d) \in \mathcal{V}^i$ can be described as a sequence of realizations of the first k categorical features. The path from the source $(0, 0)$ to itself is represented by the empty sequence $()$.

We define the set of all possible paths from the source $(0, 0)$ in the graph \mathcal{G}^i as follows:

$$\Pi^i := \{(\mathbf{z}_\ell)_{\ell \in [k]} \mid \mathbf{z}_\ell \in \mathcal{C}_\ell, k \in [m]\} \cup \{()\}.$$

We define the longest path from source $(0, 0)$ to vertex $(k, d) \in \mathcal{V}^i$ as the function $\pi^i : \mathcal{V}^i \rightarrow \Pi^i$.

Base Case:

For $k = 0$, we have $h^i(0, 0) = 0$ and $\pi^i(0, 0) = ()$ as given.

Inductive Step:

For each $k \in [m]$, assume that for all vertices $(k-1, d') \in \mathcal{V}^i$, the longest path from $(0, 0)$ to $(k-1, d')$, denoted $\pi^i(k-1, d')$, corresponds to the optimal solution of subproblem (23) or (24) at state $(k-1, d')$, and the sum of the weights along this path equals $h^i(k-1, d')$.

From equation (12), we have:

$$h^i(k, d) = \max_{\mathbf{z}_k \in \mathcal{F}_{kd}} \left\{ -y^i \beta_{\mathbf{z}_k}^\top \mathbf{z}_k + h^i(k-1, d - \delta_k \mathbb{I}\{\mathbf{z}_k \neq \mathbf{z}_k^i\}) \right\}.$$

In the graph \mathcal{G}^i , for each $\mathbf{z}_k \in \mathcal{F}_{kd}$, there is an arc from the vertex $(k-1, d - \delta_k \mathbb{I}\{\mathbf{z}_k \neq \mathbf{z}_k^i\})$ to (k, d) with a weight of $-y^i \beta_{\mathbf{z}_k}^\top \mathbf{z}_k$. By the induction hypothesis, $\pi^i(k-1, d - \delta_k \mathbb{I}\{\mathbf{z}_k \neq \mathbf{z}_k^i\})$ corresponds to the optimal solution of subproblem (23) or (24) at state $(k-1, d - \delta_k \mathbb{I}\{\mathbf{z}_k \neq \mathbf{z}_k^i\})$. Therefore, the longest path to vertex (k, d) through vertex $(k-1, d - \delta_k \mathbb{I}\{\mathbf{z}_k \neq \mathbf{z}_k^i\})$ and arc corresponding to \mathbf{z}_k is $(\pi^i(k-1, d - \delta_k \mathbb{I}\{\mathbf{z}_k \neq \mathbf{z}_k^i\}), \mathbf{z}_k)$ and its total weight is $-y^i \beta_{\mathbf{z}_k}^\top \mathbf{z}_k + h^i(k-1, d - \delta_k \mathbb{I}\{\mathbf{z}_k \neq \mathbf{z}_k^i\})$. Taking the maximum over all possible arcs corresponding to $\mathbf{z}_k \in \mathcal{F}_{kd}$, we obtain the longest path to vertex (k, d) and its total weight, as given by equation (12). Therefore, our assertion holds for each $k \in [m]$.

For $k = m+1$, from equation (13), we have:

$$h^i(m+1, 0) = \max_{(m, d') \in \mathcal{S}_1^i} h^i(m, d') - \log(-1 + \exp(r_i + \lambda d')).$$

In the graph \mathcal{G}^i , for each vertex $(m, d') \in \mathcal{V}^i$, there is an arc from (m, d') to $(m+1, 0)$ with a weight of $-\log(-1 + \exp(r_i + \lambda d'))$. By the induction hypothesis, $\pi^i(m, d')$ corresponds to the optimal solution of subproblem (24) at state (m, d') . Therefore, the longest path from $(0, 0)$ to $(m+1, 0)$ passing through (m, d') is $\pi^i(m, d')$ and its total weight is $h^i(m, d') - \log(-1 + \exp(r_i + \lambda d'))$. Taking the maximum over all possible d' , we obtain the longest path to vertex $(m+1, 0)$ and its total weight, as given by equation (13). Therefore, our assertion holds for $k = m+1$.

Based on our assertion, since problem (11) corresponds to subproblem (25), Lemma 1 holds trivially. \square

E.4 Proof of Theorem 2

Proof. We convert the constraint set indexed by $[N] \times \mathcal{C}$ in problem (3) into a more tractable form. Given the data point indexed by i , the constraint

$$r_i \geq \log(1 + \exp(-y^i (\beta_x^\top \mathbf{x}^i + \beta_z^\top \mathbf{z} + \beta_0))) - \lambda \sum_{\ell \in [m]} \delta_\ell \mathbb{I}[\mathbf{z}_\ell \neq \mathbf{z}_\ell^i]$$

must hold for all $\mathbf{z} \in \mathcal{C}$. Each inequality can be rewritten as

$$\begin{aligned} \exp(r_i) &\geq (1 + \exp(-y^i (\beta_x^\top \mathbf{x}^i + \beta_z^\top \mathbf{z} + \beta_0))) \exp\left(-\lambda \sum_{\ell \in [m]} \delta_\ell \mathbb{I}[\mathbf{z}_\ell \neq \mathbf{z}_\ell^i]\right) \\ \Leftrightarrow \exp\left(r_i + \lambda \sum_{\ell \in [m]} \delta_\ell \mathbb{I}[\mathbf{z}_\ell \neq \mathbf{z}_\ell^i]\right) &\geq 1 + \exp(-y^i (\beta_x^\top \mathbf{x}^i + \beta_z^\top \mathbf{z} + \beta_0)) \\ \Leftrightarrow \log\left(-1 + \exp(r_i + \lambda \sum_{\ell \in [m]} \delta_\ell \mathbb{I}[\mathbf{z}_\ell \neq \mathbf{z}_\ell^i])\right) &\geq -y^i (\beta_x^\top \mathbf{x}^i + \beta_z^\top \mathbf{z} + \beta_0) \\ \Leftrightarrow y^i (\beta_x^\top \mathbf{x}^i + \beta_0) &\geq -y^i \beta_z^\top \mathbf{z} - \log\left(-1 + \exp(r_i + \lambda \sum_{\ell \in [m]} \delta_\ell \mathbb{I}[\mathbf{z}_\ell \neq \mathbf{z}_\ell^i])\right). \end{aligned}$$

Therefore, the constraints indexed by $i \in [N]$ in problem (3) are equivalent to

$$y^i (\beta_x^\top x^i + \beta_0) \geq \max_{z \in \mathcal{C}} -y^i \beta_z^\top z - \log \left(-1 + \exp(r_i + \lambda \sum_{\ell \in [m]} \delta_\ell \mathbb{1}[z_\ell \neq z_\ell^i]) \right)$$

as described in Section 4.2, where the right hand side corresponds to problem (11). Therefore, problem (3) can be formulated as

$$\begin{aligned} \min_{\lambda, \mathbf{r}, \boldsymbol{\beta}} \quad & \lambda \epsilon + \frac{1}{N} \sum_{i \in [N]} r_i \\ \text{s.t.} \quad & y^i (\beta_x^\top x^i + \beta_0) \geq \max_{z \in \mathcal{C}} -y^i \beta_z^\top z - \log \left(-1 + \exp(r_i + \lambda \sum_{\ell \in [m]} \delta_\ell \mathbb{1}[z_\ell \neq z_\ell^i]) \right), \forall i \in [N] \\ & |\gamma_j^{-1} \beta_{xj}| \leq \lambda, \forall j \in [n] \\ & \lambda \geq 0, \mathbf{r} \in \mathbb{R}^N, \boldsymbol{\beta} \in \mathbb{R}^{1+n+c} \end{aligned} \quad (26)$$

Next, We transform problem (11) into a more tractable formulation. According to Lemma 1, problem (11) can be treated as a longest path problem, which can subsequently be reformulated as the following linear program.

$$\begin{aligned} \max_{\mathbf{a}^i \in \mathbb{R}^{|\mathcal{A}^i|}} \quad & \sum_{e \in \mathcal{A}^i} w^i(e) a_e^i \\ \text{s.t.} \quad & \sum_{\substack{e \in \mathcal{A}^i \\ t^i(e)=v}} a_e^i - \sum_{\substack{e \in \mathcal{A}^i \\ s^i(e)=v}} a_e^i = \begin{cases} -1 & \text{if } v = (0, 0) \\ 1 & \text{if } v = (m+1, 0), \\ 0 & \text{otherwise} \end{cases} \quad \forall v \in \mathcal{V}^i \\ & \mathbf{a}^i \geq \mathbf{0} \end{aligned}$$

where a_e^i is a decision variable indicating if we select arc $e \in \mathcal{A}^i$. Since strong duality holds for linear programming problems, we get its dual problem:

$$\begin{aligned} \min_{\boldsymbol{\mu}^i \in \mathbb{R}^{|\mathcal{V}^i|}} \quad & -\mu_{(0,0)}^i + \mu_{(m+1,0)}^i \\ \text{s.t.} \quad & \mu_{t^i(e)}^i - \mu_{s^i(e)}^i \geq w^i(e), \forall e \in \mathcal{A}^i \end{aligned}$$

where $\boldsymbol{\mu}^i \in \mathbb{R}^{|\mathcal{V}^i|}$ are dual variables. We can convert problem (26) into

$$\begin{aligned} \min_{\lambda, \mathbf{r}, \boldsymbol{\beta}, \boldsymbol{\mu}} \quad & \lambda \epsilon + \frac{1}{N} \sum_{i \in [N]} r_i \\ \text{s.t.} \quad & y^i (\beta_x^\top x^i + \beta_0) \geq \min_{\boldsymbol{\mu}^i \in \mathbb{R}^{|\mathcal{V}^i|}} -\mu_{(0,0)}^i + \mu_{(m+1,0)}^i, \forall i \in [N] \\ & \mu_{t^i(e)}^i - \mu_{s^i(e)}^i \geq w^i(e), \forall i \in [N], e \in \mathcal{A}^i \\ & |\gamma_j^{-1} \beta_{xj}| \leq \lambda, \forall j \in [n] \\ & \lambda \geq 0, \mathbf{r} \in \mathbb{R}^N, \boldsymbol{\beta} \in \mathbb{R}^{1+n+c} \end{aligned}$$

Instead of separately minimizing over $\boldsymbol{\mu}^i$, we can remove the minimization operator directly without changing the feasible region because the minimization problem ensures the existence of a feasible solution that satisfies certain constraints. Therefore, we can get

$$\begin{aligned} \min_{\lambda, \mathbf{r}, \boldsymbol{\beta}, \boldsymbol{\mu}} \quad & \lambda \epsilon + \frac{1}{N} \sum_{i \in [N]} r_i \\ \text{s.t.} \quad & y^i (\beta_x^\top x^i + \beta_0) \geq -\mu_{(0,0)}^i + \mu_{(m+1,0)}^i, \forall i \in [N] \\ & \mu_{t^i(e)}^i - \mu_{s^i(e)}^i \geq w^i(e), \forall i \in [N], e \in \mathcal{A}^i \\ & |\gamma_j^{-1} \beta_{xj}| \leq \lambda, \forall j \in [n] \\ & \lambda \geq 0, \mathbf{r} \in \mathbb{R}^N, \boldsymbol{\beta} \in \mathbb{R}^{1+n+c}, \boldsymbol{\mu} \in \mathbb{R}^{\sum_{i \in [N]} |\mathcal{V}^i|} \end{aligned}$$

□