It's All in the Mix:

Wasserstein Machine Learning with Mixed Features

Reza Belbasi, Aras Selvi, and Wolfram Wiesemann

Imperial College Business School, London, United Kingdom

{r.belbasi21, a.selvi19, ww}@imperial.ac.uk

Abstract

Problem definition: The recent advent of data-driven and end-to-end decision-making across different areas of operations management has led to an ever closer integration of prediction models from machine learning and optimization models from operations research. A key challenge in this context is the presence of estimation errors in the prediction models, which tend to be amplified by the subsequent optimization model—a phenomenon that is often referred to as the *Optimizer's Curse* or the *Error-Maximization Effect of Optimization*.

Methodology/results: A contemporary approach to combat such estimation errors is offered by distributionally robust problem formulations that consider all data-generating distributions close to the empirical distribution derived from historical samples, where 'closeness' is determined by the Wasserstein distance. While those techniques show significant promise in problems where all input features are continuous, they scale exponentially when binary and/or categorical features are present. This paper demonstrates that such mixed-feature problems can indeed be solved in polynomial time. We present a practically efficient algorithm to solve mixed-feature problems, and we compare our method against alternative techniques both theoretically and empirically on standard benchmark instances.

Managerial implications: Data-driven operations management problems often involve prediction models with discrete features. We develop and analyze a methodology that faithfully accounts for the presence of discrete features, and we demonstrate that our approach can significantly outperform existing methods that are agnostic to the presence of discrete features, both theoretically and across standard benchmark instances.

1 Introduction

The recent application of machine learning tools across all areas of operations management has successfully challenged the field's traditional division into *estimation*, whose study was frequently left to statisticians, and *modelling and optimization*, which previously constituted the core of operations management and operations research. This development is evidenced by a plethora of data-driven and end-to-end approaches that blend predictive models from machine learning with optimization frameworks from operations research and operations management. Notable examples include inventory management (Ban and Rudin, 2019; Bertsimas and Kallus, 2020), logistics (Bertsimas et al., 2019; Behrendt et al., 2023) and supply chain management (Glaeser et al., 2019), assortment optimization (Kallus and Udell, 2020; Feldman et al., 2022) and revenue management (Ferreira et al., 2016; Alley et al., 2023) as well as healthcare operations (Bertsimas et al., 2016; Bastani and Bayati, 2020; Bertsimas and Pauphilet, 2023).

The machine learning algorithms used for prediction are prone to overfitting the available data. Overfitted models perform well on the training data used to calibrate the model, but their performance deteriorates when exposed to new, unseen data. This undesirable effect is amplified if the output of a machine learning model is used as input to a downstream optimization model; this phenomenon is known by different communities as the *Optimizer's Curse* (Smith and Winkler, 2006) or the *Error-Maximization Effect of Optimization* (Michaud, 1989). Traditionally, overfitting is addressed with regularization techniques that penalize complex models characterized by large and/or dense model parameters (Hastie et al., 2009; Murphy, 2022). A contemporary alternative from the robust optimization community frames machine learning problems as Stackelberg leader-follower games where the learner selects a model that performs best against a worst-case data-generating distribution selected by a conceptual adversary ('nature') from a predefined ambiguity set (Ben-Tal et al., 2009; Rahimian and Mehrotra, 2022; Bertsimas and den Hertog, 2022). We talk about Wasserstein machine learning problems when the ambiguity

set constitutes a Wasserstein ball centered around the empirical distribution of the available historical observations (Mohajerin Esfahani and Kuhn, 2018; Blanchet and Murthy, 2019; Gao and Kleywegt, 2023). Over the last few years, Wasserstein machine learning problems have attracted enormous attention in the machine learning and optimization communities; we refer to Kuhn et al. (2019) for a recent review of the literature. Interestingly, Wasserstein learning problems admit dual characterizations as regularized learning problems (Shafieezadeh-Abadeh et al., 2015, 2019; Gao et al., 2022), and they thus contribute to a deeper understanding of the impact of regularization in machine learning. We note that other classes of ambiguity sets have been explored as well, such as moment ambiguity sets and those based on ϕ-divergences (such as the Kullback-Leibler divergence). We will not delve into the comparative advantages of different ambiguity sets, and we instead refer the interested reader to the existing literature (see, *e.g.*, Van Parys et al., 2021, Kuhn et al., 2019 and Lam, 2019).

Although Wasserstein formulations of many classical machine learning tasks admit formulations as convex optimization problems, these formulations scale exponentially in the binary and categorical input features. On the other hand, we will show that disregarding the discrete nature of these features leads to pathological ambiguity sets whose worst-case distributions lack theoretical appeal and whose resulting models can underperform in practice. This limitation has, thus far, confined the use of Wasserstein machine learning models primarily to datasets with exclusively continuous features. This constitutes a major restriction in operations management, where estimation problems frequently include categorical features. Recent examples include Qi et al. (2022), who apply Wasserstein-based quantile regression to a bike sharing inventory management problem characterized by numeric and categorical features (*e.g.*, the weather conditions, the hour of the day as well as the day of the week); Samorani et al. (2022), who study appointment scheduling problems where most features are categorical (*e.g.*, the day of the week, the patient's marital status and her insurance type); Li et al. (2023), who detect human

trafficking from user review websites (here, the categorical features describe the presence or absence of indicative words and phrases); Chan et al. (2023), who predict the macronutrient content of human milk donations using categorical features such as the infant status (term vs preterm); and Duchi et al. (2023), who enforce fairness in offender recidivism prediction through the use of categorical features such as the offender's race, gender and the existence of prior misdemeanour charges. More broadly, at the time of writing, 240 of the 496 classification and 64 of the 159 regression problems in the popular UCI machine learning repository contain discrete input features (Kelly et al., 2017).

This paper studies Wasserstein machine learning problems with mixed (continuous and binary/categorical) features from a theoretical, computational and numerical perspective. We summarize the contributions of this work as follows.

(i) From a *theoretical perspective*, we demonstrate that while Wasserstein learning with mixed features is inherently NP-hard, a wide range of problems can be solved in polynomial time. Also, contrary to Wasserstein learning with exclusively continuous features, we establish that mixed-feature Wasserstein learning does not reduce to a regularized problem.

(ii) From a *computational perspective*, we propose a cutting plane scheme that solves progressively refined relaxations of the Wasserstein learning problem as convex optimization problems. While our overall scheme is not guaranteed to terminate in polynomial time, we show that the key step of our algorithm—the identification of the most violated constraint—can be implemented efficiently for broad classes of learning problems, despite its natural representation as a combinatorial optimization problem.

(iii) From a *numerical perspective*, we show that our cutting plane scheme is substantially faster than a naïve monolithic implementation of the mixed-feature Wasserstein learning problem. We also show that our model can perform favorably against classical, regularized

and alternative robust problem formulations on standard benchmark instances.

Our paper is most closely related to the recent work of Shafieezadeh-Abadeh et al. (2015) and Shafieezadeh-Abadeh et al. (2019). Shafieezadeh-Abadeh et al. (2015) formulate the Wasserstein logistic regression problem as a convex optimization problem, they discuss the out-of-sample guarantees of their model, and they report numerical results on simulated and benchmark instances. Shafieezadeh-Abadeh et al. (2019) extend their previous work to a wider class of Wasserstein classification and regression problems. Both papers focus on problems with exclusively continuous features, and their proposed formulations would scale exponentially in any binary and/or categorical features. In contrast, our work studies Wasserstein learning problems with mixed features: we examine the theoretical properties of such problems, we develop a practically efficient solution scheme, and we report numerical results. Our work also relates closely to a recent stream of literature that characterizes Wasserstein learning problems as regularized learning problems (Shafieezadeh-Abadeh et al., 2015, 2019; Blanchet et al., 2019; Gao et al., 2022). In particular, we demonstrate that our mixed-feature Wasserstein learning problems do not admit an equivalent representation as regularized learning problems, which forms a notable contrast to the existing findings from the literature.

The present work constitutes a completely revised and substantially expanded version of a conference paper (Selvi et al., 2022). While that work focuses on logistic regression, the present paper studies broad classes of Wasserstein classification and regression problems. This expansion necessitates significant adaptations of the proof for the computational complexity of the Wasserstein learning problem (*cf.* Theorem 1 in Section 2.1), an entirely new proof for the absence of regularized problem formulations (*cf.* Theorem 2 in Section 2.2) that applies to any loss function (as opposed to only the log-loss function in Selvi et al., 2022), as well as a substantially generalized cutting plane scheme (*cf.* Algorithms 1 and 2 as well as Theorem 3 in Section 5). We also present a considerably augmented set of numerical results that encompass

both classification and regression problems (*cf.* Section 6).

The remainder of this paper is organized as follows. Section 2 defines the mixed-feature Wasserstein learning problem, analyzes its complexity, and contrasts it against a naïve formulation that disregards the discrete nature of binary and categorical features. Sections 3 and 4 develop exponential-size convex optimization problems for mixed-feature Wasserstein classification and regression problems, respectively. Section 5 develops and analyzes our cutting plane solution approach for these problems. We conclude with numerical experiments in Section 6. All datasets and source codes accompanying this work are available open source.[1]

Notation. We denote by $\mathbb{R}$ ($\mathbb{R}_+$, $\mathbb{R}_-$) the set of (non-negative, non-positive) real numbers, by $\mathbb{N}$ the set of positive integers, and we define $\mathbb{B} = \{0, 1\}$ as well as $[N] = \{1, \ldots, N\}$ for $N \in \mathbb{N}$. For a proper cone $\mathcal{C} \subseteq \mathbb{R}^n$, we write $\boldsymbol{x} \preccurlyeq_{\mathcal{C}} \boldsymbol{x}'$ and $\boldsymbol{x} \prec_{\mathcal{C}} \boldsymbol{x}'$ to abbreviate $\boldsymbol{x}' - \boldsymbol{x} \in \mathcal{C}$ and $\boldsymbol{x}' - \boldsymbol{x} \in \operatorname{int} \mathcal{C}$, respectively. The dual norm of $\|\cdot\|$ is $\|\boldsymbol{x}\|_* = \sup_{\boldsymbol{x}' \in \mathbb{R}^n} \{\boldsymbol{x}^\top \boldsymbol{x}' \,:\, \|\boldsymbol{x}'\| \leqslant 1\}$, and the cone dual to a cone $\mathcal{C}$ is $\mathcal{C}^* = \{\boldsymbol{x}' \,:\, \boldsymbol{x}^\top \boldsymbol{x}' \geqslant 0 \;\; \forall \boldsymbol{x} \in \mathcal{C}\}$. The support function of a set $\mathbb{X} \subseteq \mathbb{R}^n$ is $\mathcal{S}_{\mathbb{X}}(\boldsymbol{x}) = \sup\{\boldsymbol{x}^\top \boldsymbol{x}' \,:\, \boldsymbol{x}' \in \mathbb{X}\}$. For a function $L : \mathbb{X} \to \mathbb{R}$, we define the Lipschitz modulus as $\operatorname{lip}(L) = \sup\{|L(\boldsymbol{x}) - L(\boldsymbol{x}')| \,/\, \|\boldsymbol{x} - \boldsymbol{x}'\| \,:\, \boldsymbol{x}, \boldsymbol{x}' \in \mathbb{X},\; \boldsymbol{x} \neq \boldsymbol{x}'\}$. The set $\mathcal{P}_0(\Xi)$ contains all probability distributions supported on Ξ, and the Dirac distribution $\delta_{\boldsymbol{x}} \in \mathcal{P}_0(\mathbb{R}^n)$ places unit probability mass on $\boldsymbol{x} \in \mathbb{R}^n$. The indicator function $\mathbb{1}[\mathcal{E}]$ attains the value 1 (0) whenever the logical expression $\mathcal{E}$ is (not) satisfied.

2 Mixed-Feature Wasserstein Learning

We introduce our notation and present an exponential-size convex programming formulation of the mixed-feature Wasserstein learning problem in Section 2.1. Subsequently, Section 2.2 shows that our formulation admits a polynomial time solution scheme in a broad range of practically relevant settings. At the same time, however, we demonstrate that unlike the Wasserstein

[1] Website: `https://anonymous.4open.science/r/Wasserstein-Mixed-Features-088D/`.

learning problem with exclusively continuous features, mixed-feature problems do not possess equivalent representations as regularized problems in general. Lastly, Section 2.3 demonstrates that disregarding the discrete nature of binary and/or categorical features can lead to overly conservative ambiguity sets with pathological worst-case distributions.

2.1 Problem Formulation

We study learning problems over N data points $\boldsymbol{\xi}^n = (\boldsymbol{x}^n, \boldsymbol{z}^n, y^n) \in \Xi = \mathbb{X} \times \mathbb{Z} \times \mathbb{Y}$, $n \in [N]$, where $\boldsymbol{x}^n$, $\boldsymbol{z}^n$ and y^n represent the numerical features, the binary/categorical features and the output variable, respectively. We assume that the support $\mathbb{X}$ of the numerical features is a closed and convex subset of $\mathbb{R}^{M_\mathrm{x}}$. The support $\mathbb{Z}$ of the K discrete features satisfies $\mathbb{Z} = \mathbb{Z}(k_1) \times \ldots \times \mathbb{Z}(k_K)$, where $k_m \in \mathbb{N} \backslash \{1\}$ denotes the number of values that the m-th discrete feature can attain, $m \in [K]$, and $\mathbb{Z}(s) = \{\boldsymbol{z} \in \mathbb{B}^{s-1} : \sum_{i \in [s-1]} z_i \leqslant 1\}$ is the one-hot feature encoding. We let $M_\mathrm{z} = \sum_{m \in [K]} (k_m - 1)$ denote the number of coefficients associated with the discrete features. The support $\mathbb{Y}$ of the output variable is $\{-1, +1\}$ for classification and a closed and convex subset of $\mathbb{R}$ for regression problems, respectively. We wish to solve the Wasserstein learning problem

$$\begin{array}{ll} \underset{\boldsymbol{\beta}}{\text{minimize}} & \sup\limits_{\mathbb{Q} \in \mathfrak{B}_\epsilon(\widehat{\mathbb{P}}_N)} \mathbb{E}_{\mathbb{Q}} \left[l_{\boldsymbol{\beta}}(\boldsymbol{x}, \boldsymbol{z}, y) \right] \\ \text{subject to} & \boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}_\mathrm{x}, \boldsymbol{\beta}_\mathrm{z}) \in \mathbb{R}^{1 + M_\mathrm{x} + M_\mathrm{z}}, \end{array} \tag{1}$$

where the ambiguity set $\mathfrak{B}_\epsilon(\widehat{\mathbb{P}}_N) = \{\mathbb{Q} \in \mathcal{P}_0(\Xi) : \mathrm{W}(\mathbb{Q}, \widehat{\mathbb{P}}_N) \leqslant \epsilon\}$ represents the Wasserstein ball of radius $\epsilon > 0$ that is centered at the empirical distribution $\widehat{\mathbb{P}}_N = \frac{1}{N} \sum_{n \in [N]} \delta_{\boldsymbol{\xi}^n}$ placing equal probability mass on the N data points $\boldsymbol{\xi}^n$, $n \in [N]$, as per the following definition.

Definition 1 (Wasserstein Distance)**.** *The type-1* Wasserstein *(*Kantorovich-Rubinstein*, or*

earth mover's) distance *between two distributions* $\mathbb{P} \in \mathcal{P}_0(\Xi)$ *and* $\mathbb{Q} \in \mathcal{P}_0(\Xi)$ *is defined as*

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

where the ground metric d *on* Ξ *satisfies*

$$d(\boldsymbol{\xi}, \boldsymbol{\xi}') = \|\boldsymbol{x} - \boldsymbol{x}'\| + \kappa_{\mathrm{z}} d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}') + \kappa_{\mathrm{y}} d_{\mathrm{y}}(y, y') \quad \forall \boldsymbol{\xi} = (\boldsymbol{x}, \boldsymbol{z}, y) \in \Xi,\, \boldsymbol{\xi}' = (\boldsymbol{x}', \boldsymbol{z}', y') \in \Xi \tag{2a}$$

with $\kappa_{\mathrm{z}}, \kappa_{\mathrm{y}} > 0$ *as well as, for some* $p > 0$*,*

$$d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}') = \left(\sum_{m \in [K]} \mathbb{1}[\boldsymbol{z}_m \neq \boldsymbol{z}'_m] \right)^{1/p} \quad \textit{and} \quad d_{\mathrm{y}}(y, y') = \begin{cases} \mathbb{1}[y \neq y'] & \textit{if } \mathbb{Y} = \{-1, +1\}, \\ |y - y'| & \textit{otherwise.} \end{cases} \tag{2b}$$

The loss function $l_{\boldsymbol{\beta}}(\boldsymbol{x}, \boldsymbol{z}, y) : \mathbb{X} \times \mathbb{Z} \times \mathbb{Y} \to \mathbb{R}_+$ in problem (1) satisfies

$$l_{\boldsymbol{\beta}}(\boldsymbol{x}, \boldsymbol{z}, y) = \begin{cases} L(y \cdot [\beta_0 + \boldsymbol{\beta}_{\mathrm{x}}{}^\top \boldsymbol{x} + \boldsymbol{\beta}_{\mathrm{z}}{}^\top \boldsymbol{z}]) & \text{if } \mathbb{Y} = \{-1, +1\}, \\ L(\beta_0 + \boldsymbol{\beta}_{\mathrm{x}}{}^\top \boldsymbol{x} + \boldsymbol{\beta}_{\mathrm{z}}{}^\top \boldsymbol{z} - y) & \text{otherwise,} \end{cases}$$

where $L : \mathbb{R} \to \mathbb{R}_+$ measures the similarity between the prediction $\beta_0 + \boldsymbol{\beta}_{\mathrm{x}}{}^\top \boldsymbol{x} + \boldsymbol{\beta}_{\mathrm{z}}{}^\top \boldsymbol{z}$ and the output y. For both classification and regression problems, we consider two settings:

(i) L is convex and Lipschitz continuous with Lipschitz modulus $\mathrm{lip}(L)$, $\mathbb{X} = \mathbb{R}^{M_{\mathrm{x}}}$ and $\mathbb{Y} = \{-1, +1\}$ (for classification problems) or $\mathbb{Y} = \mathbb{R}$ (for regression problems);

(ii) L satisfies $L(e) = \max_{j \in [J]} \{a_j e + b_j\}$, and $\mathbb{X} \subseteq \mathbb{R}^{M_{\mathrm{x}}}$ and $\mathbb{Y} \subseteq \mathbb{R}$ are closed and convex.

In either case, we assume that L is not constant.

The Wasserstein learning problem (1) offers attractive generalization guarantees. While the classical choice of Wasserstein radii suffers from the curse of dimensionality (Mohajerin Esfahani

and Kuhn, 2018), recent work has developed asymptotic (Blanchet et al., 2019; Blanchet and Kang, 2021) as well as finite sample guarantees (Shafieezadeh-Abadeh et al., 2019; Gao, 2022) that apply to Wasserstein radii of the order $\mathcal{O}(1/\sqrt{N})$.

We next review a result that expresses the Wasserstein learning problem (1) as a convex optimization problem.

Observation 1. *The Wasserstein learning problem* (1) *admits the equivalent formulation*

$$\begin{array}{ll} \underset{\boldsymbol{\beta},\lambda,\boldsymbol{s}}{\text{minimize}} & \lambda\epsilon + \dfrac{1}{N}\displaystyle\sum_{n\in[N]} s_n \\ \text{subject to} & \displaystyle\sup_{(\boldsymbol{x},y)\in\mathbb{X}\times\mathbb{Y}} \{l_{\boldsymbol{\beta}}(\boldsymbol{x},\boldsymbol{z},y) - \lambda\|\boldsymbol{x}-\boldsymbol{x}^n\| - \lambda\kappa_{\mathrm{y}} d_{\mathrm{y}}(y,y^n)\} - \lambda\kappa_{\mathrm{z}} d_{\mathrm{z}}(\boldsymbol{z},\boldsymbol{z}^n) \leqslant s_n \\ & \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall n\in[N],\ \forall \boldsymbol{z}\in\mathbb{Z} \\ & \boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}_{\mathrm{x}}, \boldsymbol{\beta}_{\mathrm{z}}) \in \mathbb{R}^{1+M_{\mathrm{x}}+M_{\mathrm{z}}},\ \ \lambda\in\mathbb{R}_+,\ \ \boldsymbol{s}\in\mathbb{R}_+^N. \end{array} \tag{3}$$

Problem (3) contains embedded maximization problems, and it comprises exponentially many constraints. The latter prohibits a straightforward application of the solution approaches from Shafieezadeh-Abadeh et al. (2015) and Shafieezadeh-Abadeh et al. (2019). Sections 3 and 4 will derive equivalent reformulations of (3) for classification and regression problems, respectively, that do not contain embedded optimization problems, and Section 5 develops a cutting plane approach to introduce the constraints iteratively.

2.2 Complexity Analysis

Despite its exponential size, the Wasserstein learning problem (1) admits a polynomial time solution for the classes of loss functions that we consider in this paper.

Theorem 1 (Complexity of the Wasserstein Learning Problem (1))**.**

(i) *For generic loss functions $l_{\boldsymbol{\beta}}$, problem* (1) *is strongly NP-hard even if $M_{\mathrm{x}} = 0$ and $N = 1$.*

(ii) *For convex Lipschitz continuous and piece-wise affine loss functions $l_{\boldsymbol{\beta}}$, (1) can be solved to δ-accuracy in polynomial time whenever the set of admissible hypotheses $\boldsymbol{\beta}$ is bounded.*

Recall that an optimization problem is solved to δ-accuracy if a δ-suboptimal solution is identified that satisfies all constraints modulo a violation of at most δ. The consideration of δ-accurate solutions is standard in the numerical solution of nonlinear programs where an optimal solution may be irrational.

A by now classical result shows that when $K = 0$ (absence of categorical features), the Wasserstein learning problem (1) reduces to a classical learning problem with an additional regularization term in the objective function whenever the output weight κ_{y} in Definition 1 approaches ∞ (Shafieezadeh-Abadeh et al., 2015, 2019; Blanchet et al., 2019; Gao et al., 2022). It turns out that this reduction no longer holds when categorical features are present.

Theorem 2 (Absence of Regularizers)**.** *Fix any convex Lipschitz continuous or piece-wise affine loss function L that is not constant. The objective function of the Wasserstein learning problem,*

$$\sup_{\mathbb{Q}\in\mathfrak{B}_\epsilon(\widehat{\mathbb{P}}_N)} \mathbb{E}_{\mathbb{Q}}\left[l_{\boldsymbol{\beta}}(\boldsymbol{x},\boldsymbol{z},y)\right],$$

does not admit an equivalent reformulation as a classical regularized learning problem,

$$\mathbb{E}_{\widehat{\mathbb{P}}_N}\left[l_{\boldsymbol{\beta}}(\boldsymbol{x},\boldsymbol{z},y)\right] + \mathfrak{R}(\boldsymbol{\beta}) \qquad \text{for any } \mathfrak{R}:\mathbb{R}^{1+M_{\mathrm{x}}+M_{\mathrm{z}}}\to\mathbb{R},$$

even when the weight κ_{y} of the output distance d_{y} approaches ∞.

We emphasize that Theorem 2 applies to *any* loss function L and *any* reguarlizer $\mathfrak{R}$. We are not aware of any prior results of this form in the literature.

2.3 Comparison with Continuous-Feature Formulation

Since the reformulation (3) of the mixed-feature Wasserstein learning problem scales exponentially in the discrete features, it may be tempting to treat all features as continuous, which would allow us to solve problem (3) in polynomial time using the reformulations proposed by Shafieezadeh-Abadeh et al. (2015) and Shafieezadeh-Abadeh et al. (2019). In this section, we present a stylized example that illustrates the pitfalls of such a strategy.

Consider a classification problem over N data points $\boldsymbol{\xi}^n = (z^n, y^n)$, $n \in [N]$, where the single binary feature z follows a Bernoulli distribution, $z \sim \delta_0/2 + \delta_1/2$, and where z is related to the output variable $y \in \{-1, +1\}$ via the logistic model

$$\text{Prob}(y \,|\, z) = \frac{1}{1 + \exp\left[-y \cdot \beta_{\text{z}}^0 (2z - 1)\right]}$$

with the (unknown) true model parameter $\beta_{\text{z}}^0 = 1$. In slight deviation to Section 2.1, the above model replaces z with $2z - 1$ to compensate for the lack of an intercept in the model. We try to recover β_{z}^0 from randomly generated datasets using the log-loss function $l_\beta(z, y) = \log(1 + \exp[-y \cdot \beta_{\text{z}}(2z - 1)])$ as well as the following three models:

(i) Empirical risk model. We solve problem (1) with Wasserstein radius $\epsilon = 0$. In this case, it does not matter whether the input feature z is considered to be continuous or binary.

(ii) Mixed-feature Wasserstein model. We solve problem (1) with Wasserstein radius $\epsilon \propto 1/\sqrt{N}$ (*cf.* Section 2.2) and z modeled as a binary feature. This is our proposed approach.

(iii) Continuous-feature Wasserstein model. We solve problem (1) with Wasserstein radius $\epsilon \propto 1/\sqrt{N}$ (*cf.* Section 2.2) and z modeled as a continuous feature. The resulting problem can be solved using the techniques described by Shafieezadeh-Abadeh et al. (2015) and Shafieezadeh-Abadeh et al. (2019).[2]

[2]We emphasize that Shafieezadeh-Abadeh et al. (2015) and Shafieezadeh-Abadeh et al. (2019) do not consider

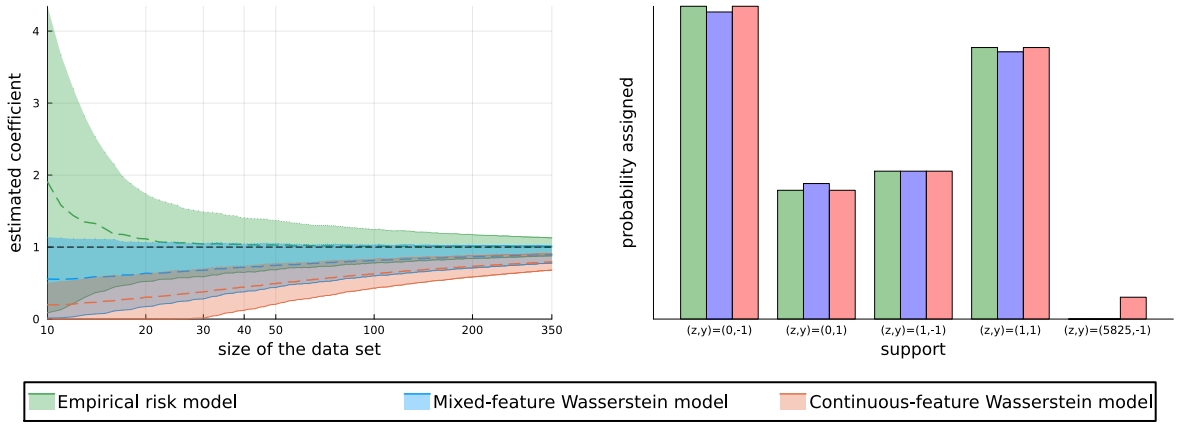

Figure 1: *Left:* Coefficient β_z estimated by the empirical risk model as well as the mixed-feature and continuous-feature Wasserstein models. *Right:* Empirical distribution as well as the worst-case distributions of the mixed-feature and continuous-feature Wasserstein models for $N = 250$ samples. The probabilities are plotted on a log-scale to make small values visible.

Figure 1 (left) reports the mean values (dashed lines) as well as the 15% and 85% quantiles (shaded regions) of the coefficients β_z estimated by the three approaches, as a function of the sample size N. To this end, we conducted 10,000 statistically independent runs for each sample size. The figure shows that as the sample size increases, all three approaches correctly estimate the true coefficient value $\beta_z^0 = 1$. While the empirical risk model is unbiased, it suffers from a high variance for small sample sizes. In contrast, both Wasserstein models enjoy a much smaller variance at the expense of a negative bias. It can clearly be seen that the bias is much more pronounced in the continuous-feature model. This is explained by Figure 1 (right): the continuous-feature model accounts for unrealistic worst-case distributions under which z takes values far outside its domain $\{0, 1\}$. Our mixed-feature model, on the other hand, restricts the worst-case distribution to the domain of z and thus hedges against realistic distributions only.

One may argue that in practice, the issue of pathological worst-case distributions is alleviated by selecting smaller radii ϵ of the Wasserstein ball. We will see in Section 6, however, that our

discrete features and hence do not advocate this approach.

mixed-feature Wasserstein model outperforms the continuous-feature Wasserstein model across a broad range of discrete-feature and mixed-feature problems, both synthetically generated and selected from standard benchmarks, even when the Wasserstein radii are selected via cross-validation. We also note that the support of the discrete features can be restricted to the interval $[0, 1]$ in the continuous-feature model when the loss function L is piece-wise affine. However, this approach does not lead to tractable models for convex and Lipschitz continuous loss functions, and the resulting worst-case distributions would still contain non-binary support points in general.

3 Wasserstein Classification with Mixed Features

This section derives reformulations of the mixed-feature Wasserstein learning problem (3) for classification problems with convex and Lipschitz continuous as well as piece-wise affine loss functions L. The material of this and the next section follows similar arguments as Shafieezadeh-Abadeh et al. (2019), adapted to the presence of discrete features as well as our ground metric.

Proposition 1. *Consider a classification problem with a convex and Lipschitz continuous loss function as well as* $\mathbb{X} = \mathbb{R}^{M_\mathrm{x}}$. *In this case, the Wasserstein learning problem* (3) *is equivalent to*

$$\begin{array}{ll} \underset{\boldsymbol{\beta},\lambda,\boldsymbol{s}}{\text{minimize}} & \lambda\epsilon + \dfrac{1}{N}\displaystyle\sum_{n\in[N]} s_n \\ \text{subject to} & \left.\begin{array}{l} l_{\boldsymbol{\beta}}(\boldsymbol{x}^n, \boldsymbol{z}, y^n) - \lambda\kappa_\mathrm{z} d_\mathrm{z}(\boldsymbol{z}, \boldsymbol{z}^n) \leqslant s_n \\ l_{\boldsymbol{\beta}}(\boldsymbol{x}^n, \boldsymbol{z}, -y^n) - \lambda\kappa_\mathrm{z} d_\mathrm{z}(\boldsymbol{z}, \boldsymbol{z}^n) - \lambda\kappa_\mathrm{y} \leqslant s_n \end{array}\right\} \forall n \in [N],\ \forall \boldsymbol{z} \in \mathbb{Z} \\ & \mathrm{lip}(L)\cdot\|\boldsymbol{\beta}_\mathrm{x}\|_* \leqslant \lambda \\ & \boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}_\mathrm{x}, \boldsymbol{\beta}_\mathrm{z}) \in \mathbb{R}^{1+M_\mathrm{x}+M_\mathrm{z}},\ \ \lambda \in \mathbb{R}_+,\ \ \boldsymbol{s} \in \mathbb{R}^N_+. \end{array} \tag{4}$$

Proposition 1 covers Wasserstein support vector machines with a smooth Hinge loss,

$$L(e) = \begin{cases} 1/2 - e & \text{if } e \leqslant 0, \\ (1/2) \cdot (1-e)^2 & \text{if } e \in (0,1), \\ 0 & \text{otherwise,} \end{cases}$$

where the Lipschitz modulus is $\mathrm{lip}(L) = 1$, and logistic regression with a log-loss,

$$L(e) = \log(1 + \exp[-e]),$$

where again $\mathrm{lip}(L) = 1$. We provide the corresponding reformulations next.

Corollary 1. *The first set of inequality constraints in* (4) *can be reformulated as follows.*

(i) For the smooth Hinge loss function:

$$\left.\begin{array}{l} \frac{1}{2}\left(w^+_{\boldsymbol{z},n} - y^n \cdot (\beta_0 + \boldsymbol{\beta}_{\mathrm{x}}{}^\top \boldsymbol{x}^n + \boldsymbol{\beta}_{\mathrm{z}}{}^\top \boldsymbol{z})\right)^2 + 1 - w^+_{\boldsymbol{z},n} - \lambda\kappa_{\mathrm{z}} d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n) \leqslant s_n \\ \frac{1}{2}\left(w^+_{\boldsymbol{z},n} - y^n \cdot (\beta_0 + \boldsymbol{\beta}_{\mathrm{x}}{}^\top \boldsymbol{x}^n + \boldsymbol{\beta}_{\mathrm{z}}{}^\top \boldsymbol{z})\right)^2 - \lambda\kappa_{\mathrm{z}} d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n) \leqslant s_n \end{array}\right\} \forall n \in [N],\ \forall \boldsymbol{z} \in \mathbb{Z}$$

(ii) For the log-loss function:

$$\log\left(1 + \exp\left[-y^n \cdot (\beta_0 + \boldsymbol{\beta}_{\mathrm{x}}{}^\top \boldsymbol{x}^n + \boldsymbol{\beta}_{\mathrm{z}}{}^\top \boldsymbol{z})\right]\right) - \lambda\kappa_{\mathrm{z}} d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n) \leqslant s_n \quad \forall n \in [N],\ \forall \boldsymbol{z} \in \mathbb{Z}$$

Here, $w^+_{\boldsymbol{z},n} \in \mathbb{R}$ are auxiliary decision variables. The second set of inequality constraints in (4) *follows similarly if we replace $w^+_{\boldsymbol{z},n} \in \mathbb{R}$ with additional auxiliary decision variables $w^-_{\boldsymbol{z},n} \in \mathbb{R}$, replace $-y^n$ with $+y^n$ and subtract the expression $\lambda\kappa_{\mathrm{y}}$ from the constraint left-hand sides.*

We now provide a reformulation of problem (3) without embedded maximizations when the loss function L is piece-wise affine.

Proposition 2. *Consider a classification problem with a piece-wise affine loss function* $L(e) = \max_{j\in[J]}\{a_j e + b_j\}$*, and assume that* $\mathbb{X} = \{\boldsymbol{x} \in \mathbb{R}^{M_\mathrm{x}} : \boldsymbol{C}\boldsymbol{x} \preccurlyeq_{\mathcal{C}} \boldsymbol{d}\}$ *for some* $\boldsymbol{C} \in \mathbb{R}^{r\times M_\mathrm{x}}$*,* $\boldsymbol{d} \in \mathbb{R}^r$ *and proper convex cone* $\mathcal{C} \subseteq \mathbb{R}^r$*. If* $\mathbb{X}$ *admits a Slater point* $\boldsymbol{x}^\mathrm{s} \in \mathbb{R}^{M_\mathrm{x}}$ *such that* $\boldsymbol{C}\boldsymbol{x}^\mathrm{s} \prec_{\mathcal{C}} \boldsymbol{d}$*, then the Wasserstein learning problem* (3) *is equivalent to*

$$
\begin{array}{ll}
\underset{\boldsymbol{\beta},\lambda,\boldsymbol{s},\boldsymbol{q}_{nj}^+,\boldsymbol{q}_{nj}^-}{\text{minimize}} & \lambda\epsilon + \dfrac{1}{N}\displaystyle\sum_{n\in[N]} s_n \\
\text{subject to} & \left.\begin{array}{l}
{\boldsymbol{q}_{nj}^+}^\top(\boldsymbol{d} - \boldsymbol{C}\boldsymbol{x}^n) + a_j y^n \cdot (\beta_0 + {\boldsymbol{\beta}_\mathrm{x}}^\top \boldsymbol{x}^n + {\boldsymbol{\beta}_\mathrm{z}}^\top \boldsymbol{z}) - \lambda\kappa_\mathrm{z} d_\mathrm{z}(\boldsymbol{z}, \boldsymbol{z}^n) + b_j \leqslant s_n \\
{\boldsymbol{q}_{nj}^-}^\top(\boldsymbol{d} - \boldsymbol{C}\boldsymbol{x}^n) - a_j y^n \cdot (\beta_0 + {\boldsymbol{\beta}_\mathrm{x}}^\top \boldsymbol{x}^n + {\boldsymbol{\beta}_\mathrm{z}}^\top \boldsymbol{z}) - \lambda\kappa_\mathrm{z} d_\mathrm{z}(\boldsymbol{z}, \boldsymbol{z}^n) - \lambda\kappa_\mathrm{y} + b_j \leqslant s_n
\end{array}\right\} \\
& \qquad\qquad \forall n \in [N],\ \forall j \in [J],\ \forall \boldsymbol{z} \in \mathbb{Z} \\
& \|a_j y^n \cdot \boldsymbol{\beta}_\mathrm{x} - \boldsymbol{C}^\top \boldsymbol{q}_{nj}^+\|_* \leqslant \lambda,\ \ \|a_j y^n \cdot \boldsymbol{\beta}_\mathrm{x} + \boldsymbol{C}^\top \boldsymbol{q}_{nj}^-\|_* \leqslant \lambda \qquad \forall n \in [N],\ \forall j \in [J] \\
& \boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}_\mathrm{x}, \boldsymbol{\beta}_\mathrm{z}) \in \mathbb{R}^{1+M_\mathrm{x}+M_\mathrm{z}},\ \ \lambda \in \mathbb{R}_+,\ \ \boldsymbol{s} \in \mathbb{R}_+^N \\
& \boldsymbol{q}_{nj}^+, \boldsymbol{q}_{nj}^- \in \mathcal{C}^*,\ n \in [N] \text{ and } j \in [J].
\end{array}
\tag{5}
$$

Proposition 2 covers Wasserstein support vector machines with a (non-smooth) Hinge loss,

$$L(e) = \max\{1 - e,\ 0\},$$

which is a piece-wise affine convex function with $J = 2$, $a_1 = -1$, $b_1 = 1$, $a_2 = 0$ and $b_2 = 0$. We provide the corresponding reformulation next.

Corollary 2. *For the (non-smooth) Hinge loss function, the first set of inequality constraints in* (5) *can be reformulated as*

$${\boldsymbol{q}_n^+}^\top(\boldsymbol{d} - \boldsymbol{C}\boldsymbol{x}^n) - y^n(\beta_0 + {\boldsymbol{\beta}_\mathrm{x}}^\top \boldsymbol{x}^n + {\boldsymbol{\beta}_\mathrm{z}}^\top \boldsymbol{z}) - \lambda\kappa_\mathrm{z} d_\mathrm{z}(\boldsymbol{z}, \boldsymbol{z}^n) + 1 \leqslant s_n \quad \forall n \in [N],\ \forall \boldsymbol{z} \in \mathbb{Z}.$$

The second set of inequality constraints in (5) *follows similarly if we replace* $-y^n$ *with* $+y^n$ *as well as* $\boldsymbol{q}_n^+$ *with* $\boldsymbol{q}_n^-$ *and subtract the expression* $\lambda\kappa_\mathrm{y}$ *from the constraint left-hand sides.*

4 Wasserstein Regression with Mixed Features

In this section, we derive reformulations of problem (3) for regression problems with convex and Lipschitz continuous as well as piece-wise affine loss functions L.

Proposition 3. *Consider a regression problem with a convex and Lipschitz continuous loss function L and $(\mathbb{X}, \mathbb{Y}) = \mathbb{R}^{M_\mathrm{x}} \times \mathbb{R}$. In this case, the Wasserstein learning problem* (3) *equals*

$$\begin{array}{lll} \underset{\boldsymbol{\beta},\lambda,\boldsymbol{s}}{\text{minimize}} & \lambda\epsilon + \dfrac{1}{N}\displaystyle\sum_{n\in[N]} s_n & \\ \text{subject to} & l_{\boldsymbol{\beta}}(\boldsymbol{x}^n, \boldsymbol{z}, y^n) - \lambda\kappa_\mathrm{z} d_\mathrm{z}(\boldsymbol{z}, \boldsymbol{z}^n) \leqslant s_n & \forall n \in [N],\ \forall \boldsymbol{z} \in \mathbb{Z} \\ & \mathrm{lip}(L) \cdot \|\boldsymbol{\beta}_\mathrm{x}\|_* \leqslant \lambda & \\ & \mathrm{lip}(L) \leqslant \lambda\kappa_\mathrm{y} & \\ & \boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}_\mathrm{x}, \boldsymbol{\beta}_\mathrm{z}) \in \mathbb{R}^{1+M_\mathrm{x}+M_\mathrm{z}},\ \ \lambda \in \mathbb{R}_+,\ \ \boldsymbol{s} \in \mathbb{R}_+^N. & \end{array} \tag{6}$$

Proposition 3 covers Wasserstein regression with a Huber loss,

$$L(e) = \begin{cases} (1/2) \cdot e^2 & \text{if } |e| \leqslant \delta, \\ \delta \cdot (|e| - (1/2) \cdot \delta) & \text{otherwise,} \end{cases}$$

where $\delta \in \mathbb{R}_+$ determines the boundary between the quadratic and the absolute loss, implying that $\mathrm{lip}(L) = \delta$. We provide the corresponding reformulation next.

Corollary 3. *For the Huber loss function, the first set of inequality constraints in* (6) *can be reformulated as*

$$\left.\begin{array}{l} \dfrac{1}{2}(\beta_0 + \boldsymbol{\beta}_\mathrm{x}{}^\top \boldsymbol{x}^n + \boldsymbol{\beta}_\mathrm{z}{}^\top \boldsymbol{z} - y^n - p_{\boldsymbol{z},n})^2 + \delta p_{\boldsymbol{z},n} - \lambda\kappa_\mathrm{z} d_\mathrm{z}(\boldsymbol{z}, \boldsymbol{z}^n) \leqslant s_n \\ \dfrac{1}{2}(\beta_0 + \boldsymbol{\beta}_\mathrm{x}{}^\top \boldsymbol{x}^n + \boldsymbol{\beta}_\mathrm{z}{}^\top \boldsymbol{z} - y^n - p_{\boldsymbol{z},n})^2 - \delta p_{\boldsymbol{z},n} - \lambda\kappa_\mathrm{z} d_\mathrm{z}(\boldsymbol{z}, \boldsymbol{z}^n) \leqslant s_n \end{array}\right\} \forall n \in [N],\ \forall \boldsymbol{z} \in \mathbb{Z},$$

where $p_{\boldsymbol{z},n} \in \mathbb{R}$ are auxiliary decision variables.

We now provide a reformulation of problem (3) without embedded maximizations when the loss function L is piece-wise affine.

Proposition 4. *Consider a regression problem with a piece-wise affine loss function $L(e) = \max_{j\in[J]}\{a_j e + b_j\}$, and assume that $\mathbb{X} \times \mathbb{Y} = \left\{(\boldsymbol{x}, y) \in \mathbb{R}^{M_\mathrm{x}+1} : \boldsymbol{C}_\mathrm{x}\boldsymbol{x} + \boldsymbol{c}_\mathrm{y} \cdot y \leqslant_{\mathcal{C}} \boldsymbol{d}\right\}$ for some $\boldsymbol{C}_\mathrm{x} \in \mathbb{R}^{r\times M_\mathrm{x}}$, $\boldsymbol{c}_\mathrm{y} \in \mathbb{R}^r$, $\boldsymbol{d} \in \mathbb{R}^r$ and proper convex cone $\mathcal{C} \subseteq \mathbb{R}^r$. If this set admits a Slater point $(\boldsymbol{x}^\mathrm{s}, y^\mathrm{s}) \in \mathbb{R}^{M_\mathrm{x}+1}$ such that $\boldsymbol{C}_\mathrm{x}\boldsymbol{x}^\mathrm{s} + \boldsymbol{c}_\mathrm{y} \cdot y^\mathrm{s} \prec_{\mathcal{C}} \boldsymbol{d}$, then problem (3) is equivalent to*

$$
\begin{array}{ll}
\underset{\boldsymbol{\beta},\lambda,\boldsymbol{s},\boldsymbol{q}_{nj}}{\text{minimize}} & \lambda\epsilon + \dfrac{1}{N}\displaystyle\sum_{n\in[N]} s_n \\
\text{subject to} & \boldsymbol{q}_{nj}^\top(\boldsymbol{d} - \boldsymbol{C}_\mathrm{x}\boldsymbol{x}^n - \boldsymbol{c}_\mathrm{y}\cdot y^n) + a_j\cdot(\beta_0 + \boldsymbol{\beta}_\mathrm{x}^\top\boldsymbol{x}^n + \boldsymbol{\beta}_\mathrm{z}{}^\top\boldsymbol{z} - y^n) - \lambda\kappa_\mathrm{z} d_\mathrm{z}(\boldsymbol{z},\boldsymbol{z}^n) + b_j \leqslant s_n \\
& \qquad\qquad \forall n\in[N],\ \forall j\in[J],\ \forall \boldsymbol{z}\in\mathbb{Z} \\
& \|a_j\cdot\boldsymbol{\beta}_\mathrm{x} - \boldsymbol{C}_\mathrm{x}{}^\top\boldsymbol{q}_{nj}\|_* \leqslant \lambda,\ \ |-a_j - \boldsymbol{c}_\mathrm{y}^\top\boldsymbol{q}_{nj}| \leqslant \lambda\kappa_\mathrm{y} \qquad \forall n\in[N],\ \forall j\in[J] \\
& \boldsymbol{\beta} = (\beta_0, \boldsymbol{\beta}_\mathrm{x}, \boldsymbol{\beta}_\mathrm{z}) \in \mathbb{R}^{1+M_\mathrm{x}+M_\mathrm{z}},\ \ \lambda\in\mathbb{R}_+,\ \ \boldsymbol{s}\in\mathbb{R}_+^N \\
& \boldsymbol{q}_{nj} \in \mathcal{C}^\star,\ n\in[N] \text{ and } j\in[J].
\end{array}
\tag{7}
$$

Proposition 4 covers Wasserstein support vector regression with an τ-insensitive loss function,

$$L(e) = \max\{|e| - \tau,\ 0\}$$

with robustness parameter $\tau \in \mathbb{R}_+$, which is a piece-wise affine convex function with $J = 3$, $a_1 = 1$, $b_1 = -\tau$, $a_2 = -1$, $b_2 = -\tau$, $a_3 = 0$ and $b_3 = 0$. It also covers Wasserstein quantile regression with a pinball loss function,

$$L(e) = \max\{-\tau e,\ (1-\tau)e\}$$

with robustness parameter $0 \leqslant \tau \leqslant 1$, which is a piece-wise affine convex function with $J = 2$, $a_1 = -\tau$, $b_1 = 0$, $a_2 = 1 - \tau$ and $b_2 = 0$. We provide the corresponding reformulations next.

Corollary 4. *The first set of inequality constraints in* (7) *can be reformulated as follows.*

(i) For the τ-insensitive loss function:

$$\left.\begin{array}{l} \boldsymbol{t}_{n1}{}^{\top}(\boldsymbol{d}_1 - \boldsymbol{C}_1\boldsymbol{x}^n) + \boldsymbol{v}_{n1}{}^{\top}(\boldsymbol{d}_2 - \boldsymbol{C}_2 y^n) + (\beta_0 + \boldsymbol{\beta}_{\mathrm{x}}{}^{\top}\boldsymbol{x}^n + \boldsymbol{\beta}_{\mathrm{z}}{}^{\top}\boldsymbol{z} - y^n) \\ \qquad\qquad -\tau - \lambda\kappa_{\mathrm{z}} d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n) \leqslant s_n \\ \boldsymbol{t}_{n2}{}^{\top}(\boldsymbol{d}_1 - \boldsymbol{C}_1\boldsymbol{x}^n) + \boldsymbol{v}_{n2}{}^{\top}(\boldsymbol{d}_2 - \boldsymbol{C}_2 y^n) - (\beta_0 + \boldsymbol{\beta}_{\mathrm{x}}{}^{\top}\boldsymbol{x}^n + \boldsymbol{\beta}_{\mathrm{z}}{}^{\top}\boldsymbol{z} - y^n) \\ \qquad\qquad -\tau - \lambda\kappa_{\mathrm{z}} d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n) \leqslant s_n \end{array}\right\} \quad \forall n \in [N],\ \forall \boldsymbol{z} \in \mathbb{Z}$$

(ii) For the pinball loss function:

$$\left.\begin{array}{l} \boldsymbol{t}_{n1}{}^{\top}(\boldsymbol{d}_1 - \boldsymbol{C}_1\boldsymbol{x}^n) + \boldsymbol{v}_{n1}{}^{\top}(\boldsymbol{d}_2 - \boldsymbol{C}_2 y^n) - \tau(\beta_0 + \boldsymbol{\beta}_{\mathrm{x}}{}^{\top}\boldsymbol{x}^n + \boldsymbol{\beta}_{\mathrm{z}}{}^{\top}\boldsymbol{z} - y^n) \\ \qquad\qquad -\lambda\kappa_{\mathrm{z}} d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n) \leqslant s_n \\ \boldsymbol{t}_{n2}{}^{\top}(\boldsymbol{d}_1 - \boldsymbol{C}_1\boldsymbol{x}^n) + \boldsymbol{v}_{n2}{}^{\top}(\boldsymbol{d}_2 - \boldsymbol{C}_2 y^n) + (1-\tau)(\beta_0 + \boldsymbol{\beta}_{\mathrm{x}}{}^{\top}\boldsymbol{x}^n + \boldsymbol{\beta}_{\mathrm{z}}{}^{\top}\boldsymbol{z} - y^n) \\ \qquad\qquad -\lambda\kappa_{\mathrm{z}} d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n) \leqslant s_n \end{array}\right\} \quad \forall n \in [N],\ \forall \boldsymbol{z} \in \mathbb{Z}$$

5 Cutting Plane Solution Scheme

The reformulations of the mixed-feature Wasserstein learning problem (3) developed in Sections 3 and 4 are convex, but their numbers of constraints scale exponentially in K, the number of discrete features. Our numerical results will show that solving these reformulations monolithically does not scale to the problem sizes usually encountered in practice. This section therefore develops a cutting plane approach that iteratively introduces only those constraints that are most violated by a sequence of incumbent solutions.

Algorithm 1 Cutting Plane Scheme for Problem (8)

Input: (Possibly empty) initial constraint set $\mathcal{W} \subseteq [N] \times \mathcal{I} \times \mathbb{Z}$.

Output: Optimal solution $(\boldsymbol{\theta}^\star, \boldsymbol{\sigma}^\star)$ to problem (8).

Initialize $(\mathrm{LB}, \mathrm{UB}) = (-\infty, +\infty)$ as lower and upper bounds for problem (8).

while LB $<$ UB **do**

Let $(\boldsymbol{\theta}^\star, \boldsymbol{\sigma}^\star)$ be optimal in the relaxation of (8) involving the constraints $(n, i, \boldsymbol{z}) \in \mathcal{W}$.

for $n \in [N]$ **do**

Identify, for each $i \in \mathcal{I}$, a most violated constraint

$$\boldsymbol{z}(n,i) \in \operatorname*{arg\,max}_{\boldsymbol{z} \in \mathbb{Z}} \left\{ f_{ni}(g_{ni}(\boldsymbol{\theta}^\star, \boldsymbol{\xi}^n_{-\mathbf{z}}; \boldsymbol{z})) - h_{ni}(\boldsymbol{\theta}^\star, \boldsymbol{\xi}^n_{-\mathbf{z}}; d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n)) - \sigma^\star_n \right\}$$

associated with $(\boldsymbol{\theta}^\star, \boldsymbol{\sigma}^\star)$ and (n, i), and denote the constraint violation by $\vartheta(n, i)$.

Let $i(n) \in \arg\max\{\vartheta(n, i) : i \in \mathcal{I}\}$ and add $(n, i(n), \boldsymbol{z}(n, i(n)))$ to $\mathcal{W}$ if $\vartheta(n, i(n)) > 0$.

end for

Define $\boldsymbol{\vartheta}^\star \in \mathbb{R}^N$ via $\vartheta^\star_n = \max\{\vartheta(n, i(n)), 0\}$, $n \in [N]$.

Update $\mathrm{LB} = f_0(\boldsymbol{\theta}^\star, \boldsymbol{\sigma}^\star)$ and $\mathrm{UB} = \min\{\mathrm{UB},\ f_0(\boldsymbol{\theta}^\star, \boldsymbol{\sigma}^\star + \boldsymbol{\vartheta}^\star)\}$.

end while

In the following, we employ the unified problem representation

$$\begin{array}{ll} \underset{\boldsymbol{\theta}, \boldsymbol{\sigma}}{\text{minimize}} & f_0(\boldsymbol{\theta}, \boldsymbol{\sigma}) \\ \text{subject to} & f_{ni}(g_{ni}(\boldsymbol{\theta}, \boldsymbol{\xi}^n_{-\mathbf{z}}; \boldsymbol{z})) - h_{ni}(\boldsymbol{\theta}, \boldsymbol{\xi}^n_{-\mathbf{z}}; d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n)) \leqslant \sigma_n \qquad \forall n \in [N],\ \forall i \in \mathcal{I},\ \forall \boldsymbol{z} \in \mathbb{Z} \\ & \boldsymbol{\theta} \in \Theta,\ \ \boldsymbol{\sigma} \in \mathbb{R}^N_+, \end{array} \tag{8}$$

where $\boldsymbol{\xi}^n_{-\mathbf{z}} = (\boldsymbol{x}^n, y^n)$ and $\mathcal{I}$ is a finite index set. One readily confirms that problem (8) encompasses our classification and regression problems (4)–(7) with Lipschitz continuous and piece-wise affine loss functions as special cases. To ensure that (8) is convex, we stipulate that $f_0 : \mathbb{R}^{M_{\boldsymbol{\theta}}} \times \mathbb{R}^N_+ \to \mathbb{R}$ and $f_{ni} : \mathbb{R} \to \mathbb{R}$ are convex, $g_{ni} : \mathbb{R}^{M_{\boldsymbol{\theta}}} \times (\mathbb{X} \times \mathbb{Y}) \times \mathbb{R}^{M_{\mathrm{z}}} \to \mathbb{R}$ is bi-affine in $\boldsymbol{\theta}$ and $\boldsymbol{z}$ for every fixed $\boldsymbol{\xi}^n_{-\mathbf{z}}$, $h_{ni} : \mathbb{R}^{M_{\boldsymbol{\theta}}} \times (\mathbb{X} \times \mathbb{Y}) \times \mathbb{R} \to \mathbb{R}$ is concave in $\boldsymbol{\theta}$ for every fixed $\boldsymbol{\xi}^n_{-\mathbf{z}}$ and every fixed value of its last component, $n \in [N]$ and $i \in \mathcal{I}$, and $\Theta \subseteq \mathbb{R}^{M_{\boldsymbol{\theta}}}$ is a convex set.

Algorithm 1 describes our cutting plan scheme. We next assert its correctness.

Proposition 5. *Algorithm 1 terminates in finite time with an optimal solution* $(\boldsymbol{\theta}^\star, \boldsymbol{\sigma}^\star)$ *to*

Algorithm 2 Identification of a Most Violated Constraint in Problem (8)

Input: Incumbent solution $(\boldsymbol{\theta}^\star, \boldsymbol{\sigma}^\star)$ and constraint group index $(n, i) \in [N] \times \mathcal{I}$.
Output: A most violated constraint index $\boldsymbol{z}(n, i)$ in constraint group (n, i).
Initialize the candidate constraint index set $\mathcal{Z} = \varnothing$.
Let $(\boldsymbol{w}, w_0) \in \mathbb{R}^{M_{\mathrm{z}}} \times \mathbb{R}$ be such that $g_{ni}(\boldsymbol{\theta}^\star, \boldsymbol{\xi}^n_{-\mathbf{z}}; \boldsymbol{z}) = \boldsymbol{w}^\top \boldsymbol{z} + w_0$.
for $\mu \in \{\pm 1\}$ **do**
Compute $\boldsymbol{z}^\star_m \in \arg\max\{\mu \cdot \boldsymbol{w}_m{}^\top \boldsymbol{z}_m \,:\, \boldsymbol{z}_m \in \mathbb{Z}(k_m) \backslash \{\boldsymbol{z}^n_m\}\}$ for all $m \in [K]$.
Compute a permutation $\pi : [K] \to [K]$ such that

$$\mu \cdot \boldsymbol{w}_{\pi(m)}{}^\top (\boldsymbol{z}^\star_{\pi(m)} - \boldsymbol{z}^n_{\pi(m)}) \;\geqslant\; \mu \cdot \boldsymbol{w}_{\pi(m')}{}^\top (\boldsymbol{z}^\star_{\pi(m')} - \boldsymbol{z}^n_{\pi(m')}) \qquad \forall 1 \leqslant m \leqslant m' \leqslant K.$$

for $\delta \in [K] \cup \{0\}$ **do**
Add $\boldsymbol{z}(\delta)$ to $\mathcal{Z}$, where $\boldsymbol{z}_m(\delta) = \boldsymbol{z}^\star_m$ if $\pi(m) \leqslant \delta$; $= \boldsymbol{z}^n_m$ otherwise, $m \in [K]$.
end for
end for
Select $\boldsymbol{z}(n, i) \in \arg\max\{f_{ni}(g_{ni}(\boldsymbol{\theta}^\star, \boldsymbol{\xi}^n_{-\mathbf{z}}; \boldsymbol{z})) - h_{ni}(\boldsymbol{\theta}^\star, \boldsymbol{\xi}^n_{-\mathbf{z}}; d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n)) - \sigma^\star_n \,:\, \boldsymbol{z} \in \mathcal{Z}\}$.

problem (8). *Moreover,* LB *and* UB *constitute monotonic sequences of lower and upper bounds on the optimal value of* (8) *throughout the execution of the algorithm.*

A key step in Algorithm 1 concerns the identification of a constraint $(n, i, \boldsymbol{z}) \in [N] \times \mathcal{I} \times \mathbb{Z}$ that is most violated by the incumbent solution $(\boldsymbol{\theta}^\star, \boldsymbol{\sigma}^\star)$. We next show that the underlying combinatorial problem can be solved efficiently by Algorithm 2.

Theorem 3. *For a given incumbent solution* $(\boldsymbol{\theta}^\star, \boldsymbol{\sigma}^\star)$ *and a constraint group index* $(n, i) \in [N] \times \mathcal{I}$ *in problem* (8)*, Algorithm 2 identifies a most violated constraint index* $\boldsymbol{z}(n, i)$ *in time* $\mathcal{O}(M_{\mathbf{z}} + KT + K \log K)$*, where* T *is the time required to compute* f_{ni}*,* g_{ni} *and* h_{ni}*.*

6 Numerical Results

We compare empirically the performance of our mixed-feature Wasserstein learning problem (1) with classical and regularized learning methods as well as a continuous-feature approximation of problem (1). To this end, Section 6.1 compares the runtimes of our cutting plane approach with

Method	N = 100			N = 500			N = 1,000		
	$K = 5$	$K = 10$	$K = 15$	$K = 5$	$K = 10$	$K = 15$	$K = 5$	$K = 10$	$K = 15$
LR-cut	0.03 (±0.02)	0.05 (±0.03)	0.09 (±0.05)	0.14 (±0.06)	0.28 (±0.16)	0.33 (±0.22)	0.33 (±0.15)	0.46 (±0.29)	0.79 (±0.49)
LR-mono	0.72 (±0.07)	52.37 (±6.24)	**NaN**	4.12 (±0.26)	403.09 (±36.52)	**NaN**	9.06 (±0.77)	878.53 (±88.65)	**NaN**
SVM-cut	0.01 (±0.00)	0.01 (±0.00)	0.01 (±0.01)	0.02 (±0.01)	0.03 (±0.02)	0.05 (±0.04)	0.04 (±0.02)	0.06 (±0.04)	0.11 (±0.07)
SVM-mono	0.03 (±0.00)	2.18 (±0.19)	121.12 (±13.51)	0.18 (±0.01)	18.22 (±7.23)	**NaN**	0.40 (±0.03)	39.33 (±11.43)	**NaN**
SSVM-cut	0.02 (±0.01)	0.04 (±0.02)	0.05 (±0.02)	0.10 (±0.05)	0.16 (±0.09)	0.20 (±0.13)	0.20 (±0.10)	0.25 (±0.13)	0.54 (±0.36)
SSVM-mono	0.43 (±0.03)	35.79 (±4.12)	2,285.08 (±300.93)	2.43 (±0.17)	234.56 (±41.39)	**NaN**	5.02 (±0.56)	475.32 (±107.11)	**NaN**
RR-cut	0.03 (±0.01)	0.06 (±0.01)	0.09 (±0.02)	0.17 (±0.04)	0.29 (±0.04)	0.38 (±0.04)	0.31 (±0.09)	0.63 (±0.08)	0.80 (±0.17)
RR-mono	0.25 (±0.02)	19.32 (±1.46)	**NaN**	1.37 (±0.08)	124.39 (±10.47)	**NaN**	3.03 (±0.15)	287.97 (±36.43)	**NaN**
QR-cut	0.01 (±0.00)	0.01 (±0.00)	0.02 (±0.00)	0.03 (±0.01)	0.06 (±0.02)	0.09 (±0.00)	0.05 (±0.01)	0.14 (±0.03)	0.20 (±0.01)
QR-mono	0.03 (±0.00)	2.05 (±0.31)	194.16 (±56.89)	0.16 (±0.01)	13.08 (±2.60)	**NaN**	0.37 (±0.03)	36.20 (±10.65)	**NaN**
SVR-cut	0.01 (±0.00)	0.02 (±0.00)	0.03 (±0.00)	0.03 (±0.01)	0.07 (±0.02)	0.1 (±0.01)	0.06 (±0.01)	0.15 (±0.03)	0.22 (±0.01)
SVR-mono	0.04 (±0.00)	2.67 (±0.45)	204.66 (±54.99)	0.18 (±0.01)	15.28 (±2.57)	**NaN**	0.43 (±0.04)	40.09 (±8.03)	**NaN**

Table 1: Mean (± std. dev.) runtimes in secs of the cutting plane (-cut) and monolithic (-mono) implementations of the logistic (LR), hinge (SVM), smooth hinge (SSVM), Huber (RR; with $\delta = 0.05$), pinball (QR; with $\tau = 0.5$) and τ-insensitive (SVR; with $\tau = 10^{-2}$) loss functions. Entries labeled '**NaN**' indicate that *none* of the corresponding 100 instances were solved due to the imposed runtime limit (for $N \leqslant 500$) or insufficient memory (for $N = 1000$).

those of a monolithic solution of problem (3). Subsequently, Section 6.2 extends our analysis of Section 2.3 by comparing the out-of-sample losses of our mixed-feature Wasserstein learning problem (1) with those of the continuous-feature approximation when the Wasserstein radius is selected via cross-validation. Finally, Section 6.3 compares the out-of-sample performance of our formulation (1) with that of alternative methods on standard benchmark instances.

All algorithms were implemented in Julia v1.9.2 using the JuMP package and MOSEK v10.0, and all experiments were run on Intel Xeon 2.66GHz cluster nodes with 8GB memory in single-core and single-thread mode (unless otherwise specified). All implementations, datasets and experimental results are available on the GitHub repository accompanying this work.

6.1 Comparison with Monolithic Formulation

We compare the runtimes of our cutting plane method from Section 5 with those of a monolithic solution of the mixed-feature Wasserstein learning formulation (3). To be in full control of the

problem dimensions, we generate synthetic instances with $N \in \{100, 500, 1000\}$ data points, no numerical features, and $K \in \{5, 10, 15\}$ binary features (*i.e.*, $k_m = 2$ for all $m \in [K]$). Throughout our experiments, we set the Wasserstein radius to $\epsilon = 10^{-2}$, we choose $(\kappa_{\mathrm{z}}, \kappa_{\mathrm{y}}, p) = (1, \sqrt{m}, 1)$ in our ground metric (2). We allocate 96GB RAM in an attempt to accommodate for the instance sizes of the monolithic model, and we impose a time limit of 1 hour. We refer to the GitHub repository for further details of the instance generation procedure. Table 1 summarizes the runtimes across 100 randomly generated problem instances for 6 different loss functions. The table reveals that the monolithic formulation does not scale beyond $K = 10$ binary features, whereas our cutting plane scheme can solve all considered instances within fractions of a second.

6.2 Comparison with Continuous-Feature Formulation

Section 2.3 demonstrated that modeling discrete features in the Wasserstein learning problem (1) as continuous and subsequently solving a continuous-feature formulation may inadvertently hedge against pathological worst-case distributions that in turn lead to an excessive bias in the estimated coefficients of the learned model. To facilitate a consistent comparison of the mixed-feature and continuous-feature formulations, Section 2.3 fixed the choice of the Wasserstein radius ϵ.

We now explore whether the findings from Section 2.3 remain valid when the Wasserstein radius ϵ is selected via cross-validation to optimize the out-of-sample losses. As in the previous subsection, we generate synthetic problem instances to be in full control of the problem dimensions. In particular, we generate 100 synthetic instances for each of the 6 loss functions considered previously, assuming that the loss functions explain a large part (but not all) of the variability in the data. All instances comprise $N = 20$ data points, no numerical features, and $K = 20$ binary features. The small number of data points ensures that distributional robustness is required to obtain small out-of-sample losses. For each instance, we solve a mixed-feature Wasserstein learning problem that treats some of the features as binary, whereas the other

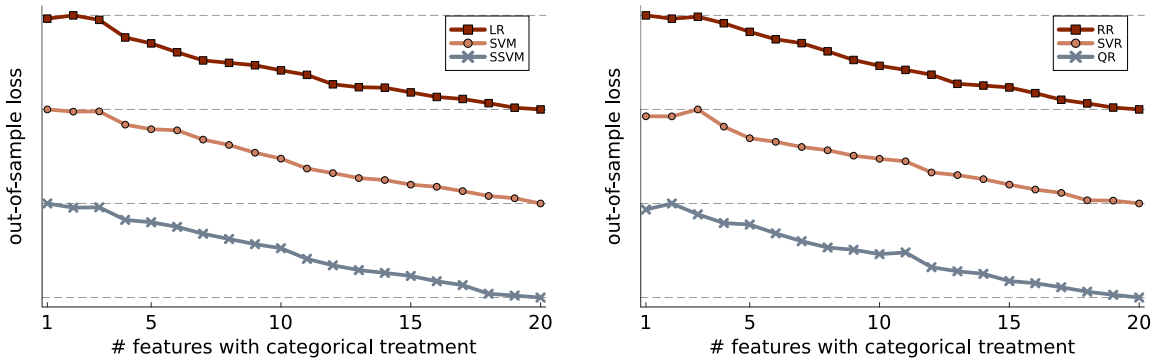

Figure 2: Mean out-of-sample losses for various classification (left) and regression (right) tasks when the number of categorical features that are treated as such varies. All losses are scaled to $[0, 1]$ and shifted so that the curves do not overlap. We use the same abbreviations as in Table 1.

features are treated as numerical. In particular, the extreme cases of modelling all and none of the features as binary correspond to our mixed-feature Wasserstein learning problem (1) and its continuous-feature approximation, respectively. We cross-validate the Wasserstein radius from the set $\epsilon \in \{10^{-7}, 10^{-6}, \ldots, 10^{-1}\}$, and we choose $(\kappa_{\mathrm{z}}, \kappa_{\mathrm{y}}, p) = (1, 1, 1)$ as well as $\|\cdot\| = \|\cdot\|_1$ in our ground metric (2). We refer to the GitHub repository for further details of the instance generation procedure. Figure 2 reports the average out-of-sample losses across 100 randomly generated problem instances for the different loss functions. The figure reveals an overall trend of improved results when the number of binary features that are treated as such increases. Qualitatively, we observe that this conclusion is robust to different choices of the ϵ-grid used for cross-validation; we refer to the GitHub repository for further details.

6.3 Performance on Benchmark Instances

In the previous subsection, we observed that the mixed-feature Wasserstein learning problem (1) can outperform its continuous-feature approximation in terms of out-of-sample losses. In practice, however, loss functions merely serve as surrogates for the misclassification rate (in classification problems) or mean squared error (in regression problems). Moreover, the previous subsection

considered synthetically generated problem instances, which lack some of the intricate structure of real-life datasets. This section therefore compares the out-of-sample misclassification rates and mean squared errors of problem (1) with those of alternative methods on standard benchmark instances. In particular, we selected 8 of the most popular classification and 7 of the most popular regression datasets from the UCI machine learning repository (Kelly et al., 2017). We processed the datasets to *(i)* handle missing values and inconsistencies in the data, *(ii)* employ a one-hot encoding for categorical variables, and *(iii)* convert the output variable (to a binary value for classification tasks and a $[-1, 1]$-interval for regression tasks). All datasets, processing scripts as well as further results on other UCI datasets can be found in the GitHub repository.

Tables 2–7 report averaged results over 100 random splits into 80% training set and 20% test set for both the original datasets as well as parsimonious variants where only half of the data points are available. In the tables, the column groups report (from left to right) the problem instances' names and dimensions; the results for the unregularized implementations of the nominal problem as well as the mixed-feature Wasserstein learning problem (1) using $(\kappa_{\mathrm{z}}, p) = (1, 1)$, $\|\cdot\| = \|\cdot\|_1$ and $\kappa_{\mathrm{y}} \in \{1, m\}$ in our ground metric (2); the results for the correponding l_2-regularized versions; and the results for two continuous-feature approximations of problem (1). For the unregularized mixed-feature and continuous-feature Wasserstein learning problems, we cross-validate the hyperparameter ε from the set $\{0, 10^{-5}, 10^{-3}, 10^{-1}\}$. For the regularized nominal model, we cross-validate the regularization penalty α from the set $\{0\} \cup \{c \cdot 10^{-p} : c \in \{1, 5\},\ p = 1, \ldots, 6\}$. For the regularized mixed-feature Wasserstein learning problem, finally, we cross-validate the hyperparameters (ε, α) from the Cartesian product of the previous two sets. The method with the smallest error in each column group is highlighted in bold, and the method with the smallest error across all column groups has a grey background. For instances where a version of the mixed-feature Wasserstein learning problem attains the smallest error across all column groups, the symbols † and ‡ indicate a statistically significant improvement of that model over

the nominal and regularized nominal learning problem as well as over the better of the two continuous-feature approximations, respectively, at a p-value of 0.05. Most nominal models were solved within seconds, most continuous-feature models were solved within minutes, and most of the mixed-feature models were solved within 10 minutes. We relegate the details of the statistical significance tests and runtimes to the GitHub repository.

Overall, we observe from the tables that the mixed-feature Wasserstein learning problems outperform both the (regularized) nominal problems and the continuous-feature approximations in the classification and regression tasks. The outperformance of the mixed-feature Wasserstein learning problems over the continuous-feature approximations tends to be more substantial for problem instances with many categorical features, which further confirms our findings from Section 2.3 and Section 6.2. Also, the mixed-feature Wasserstein learning problems tend to perform better on the parsimonious versions of the problem instances. This is intuitive as we expect robust optimization to be particularly effective in data sparse environments. Regularizing the mixed-feature Wasserstein learning problem does not yield significant advantages in the classification tasks, but it does help in the regression tasks. Finally, while we do not observe any significant differences among the classification loss functions, the Huber loss function performs best on the regression problem instances.

Acknowledgments The authors are indebted to Oscar Dowson for his invaluable invaluable guidance on efficient implementations of our Julia source codes.

Dataset	N	M_{x}	M_{z}	K	Reduced?	Nom	MixF ($\kappa_{\mathrm{y}} = 1$)	MixF ($\kappa_{\mathrm{y}} = K$)	r-Nom	r-MixF ($\kappa_{\mathrm{y}} = 1$)	r-MixF ($\kappa_{\mathrm{y}} = K$)	ConF ($\kappa_{\mathrm{y}} = 1$)	ConF ($\kappa_{\mathrm{y}} = K$)
balance-scale	625	0	16	4	✖	0.56%	0.44%	**0.40% ††‡**	0.52%	0.52%	**0.48%**	0.48%	0.44%
					✓	**1.65%**	**1.65% ‡**	**1.65% ‡**	**1.92%**	**1.92%**	**1.92%**	1.73%	1.91%
breast-cancer	277	0	42	9	✖	30.10%	**28.91%**	29.64%	29.46%	**28.64% ††**	29.36%	29.18%	29.55%
					✓	31.87%	30.93%	**30.30%**	29.58%	28.68%	**28.46% ††‡**	31.69%	36.45%
credit-approval	690	6	36	9	✖	13.59%	13.48%	**13.01% ††‡**	13.70%	13.59%	**13.12%**	13.95%	13.23%
					✓	16.30%	**15.00%**	15.10%	14.58%	14.94%	**14.43% ‡**	14.92%	15.64%
cylinder-bands	539	19	43	14	✖	22.85%	22.52%	**22.38%**	22.90%	**22.24% †**	23.18%	22.90%	22.52%
					✓	27.37%	28.55%	**26.39% †**	27.18%	27.25%	**27.12%**	26.81%	26.47%
lymphography	148	0	42	18	✖	17.24%	**16.90%**	17.24%	15.86%	16.90%	**14.83% ††**	18.97%	17.10%
					✓	23.98%	22.22%	**19.10%**	17.96%	**17.67% ††**	18.30%	25.51%	19.43%
primacy	339	0	25	17	✖	**13.51%**	**13.51%**	14.40%	14.25%	**13.73%**	14.33%	13.81%	14.55%
					✓	15.89%	**13.99% †**	15.30%	14.70%	**14.34%**	14.73%	14.09%	15.10%
spect	267	0	22	22	✖	20.28%	19.43%	**19.25%**	20.09%	19.25%	**17.35% ††**	20.28%	20.00%
					✓	23.22%	21.13%	**19.97% ††**	23.34%	22.44%	**20.34%**	22.50%	20.88%
tic-tac-toe	958	0	18	9	✖	1.94%	**1.70%**	**1.70%**	**1.70%**	**1.70%**	**1.70%**	1.70%	1.70%
					✓	2.75%	**1.70% †**	**1.70% †**	1.81%	1.67%	**1.66%**	1.70%	1.70%

Table 2: Mean classification error for the **logistic regression loss function**. N, M_{x}, M_{z} and K refer to the problem parameters from Section 2. The column 'Reduced?' indicates whether the full or sparse version of the dataset is being considered. Nom, MixF and ConF refer to the nominal problem formulation, the mixed-feature Wasserstein learning problem 1 and its continuous-feature approximation, respectively. The prefix 'r-' indicates the use of an l_2-regularization.

Dataset	N	M_{x}	M_{z}	K	Reduced?	Nom	MixF ($\kappa_{\mathrm{y}} = 1$)	MixF ($\kappa_{\mathrm{y}} = K$)	r-Nom	r-MixF ($\kappa_{\mathrm{y}} = 1$)	r-MixF ($\kappa_{\mathrm{y}} = K$)	ConF ($\kappa_{\mathrm{y}} = 1$)	ConF ($\kappa_{\mathrm{y}} = K$)
balance-scale	625	0	16	4	✖	4.20%	**4.08%**	4.20%	3.36%	**3.08% ††**	3.28%	3.88%	4.20%
					✓	5.91%	**5.56% †**	5.88%	5.69%	6.23%	**5.64%**	5.64%	5.68%
breast-cancer	277	0	42	9	✖	30.27%	**30.09% ‡**	30.82%	30.46%	30.82%	**30.18%**	35.18%	36.00%
					✓	31.66%	31.15%	**30.69%**	31.05%	30.93%	**29.58% ††**	31.96%	36.78%
credit-approval	690	6	36	9	✖	13.84%	13.88%	**13.51%**	14.06%	14.02%	**13.59%**	13.62%	14.06%
					✓	15.86 %	16.21%	**15.81%**	14.84%	15.98%	**14.83% ‡**	15.22%	18.62%
cylinder-bands	539	19	43	14	✖	23.22%	23.08%	**22.52% †**	23.08%	23.04%	**22.85%**	22.85%	24.11%
					✓	28.47%	27.86%	**27.29% ††**	28.55%	28.99%	**28.05%**	27.97%	27.88%
lymphography	148	0	42	18	✖	19.14%	**18.97%**	20.52%	18.79%	17.41%	**17.24% †**	18.28%	19.83%
					✓	**20.28%**	20.91%	21.25%	19.49%	19.77%	**18.64% ††**	23.92%	21.31%
primacy	339	0	25	17	✖	14.03%	13.73%	**13.28%**	13.36%	13.73%	**13.13% ‡**	13.81%	13.66%
					✓	15.96%	**15.05%**	15.47%	14.53%	14.68%	**14.51% ‡**	14.78%	15.00%
spect	267	0	22	22	✖	20.19%	20.76%	**18.96% ††**	21.04%	20.85%	**19.34%**	21.23%	20.28%
					✓	24.16%	**21.50% †**	21.97%	22.031%	22.06%	**21.91%**	23.28%	21.53%
tic-tac-toe	958	0	18	9	✖	**1.70%**	**1.70%**	**1.70%**	**1.70%**	**1.70%**	**1.70%**	1.70%	1.70%
					✓	2.58%	**1.65%**	**1.65%**	**1.65%**	**1.65%**	**1.65%**	1.65%	1.65%

Table 3: Mean classification error for the **hinge loss function**. We use the same abbreviations as in Table 2.

Dataset	N	M_{x}	M_{z}	K	Reduced?	Nom	MixF ($\kappa_{\mathrm{y}} = 1$)	MixF ($\kappa_{\mathrm{y}} = K$)	r-Nom	r-MixF ($\kappa_{\mathrm{y}} = 1$)	r-MixF ($\kappa_{\mathrm{y}} = K$)	ConF ($\kappa_{\mathrm{y}} = 1$)	ConF ($\kappa_{\mathrm{y}} = K$)
balance-scale	625	0	16	4	✖	1.36%	**0.40%**	**0.40% ††**	1.52%	**0.48%**	**0.48%**	1.32%	1.36%
					✓	3.15%	2.15%	**2.01%**	3.25%	2.08%	**1.81% ††**	3.29%	3.31%
breast-cancer	277	0	42	9	✖	29.55%	**28.82% †**	29.18%	30.18%	32.27%	**29.64%**	30.36%	29.18%
					✓	31.30%	31.36%	**30.36%**	28.55%	28.52%	**27.41% ††**	29.88%	29.94%
credit-approval	690	6	36	9	✖	13.44%	**13.37%**	13.41%	13.44%	**13.37% ‡**	13.41%	13.66%	14.46%
					✓	15.59%	**15.16%**	15.52%	14.65%	14.53%	**14.52% ††**	15.30%	15.75%
cylinder-bands	539	19	43	14	✖	23.27%	**21.78% ††**	22.43%	23.04%	**22.52%**	22.85%	22.57%	22.71%
					✓	27.43%	**26.50% ††**	27.57%	**27.09%**	27.40%	27.37 %	27.11%	27.40%
lymphography	148	0	42	18	✖	19.31%	**17.76%**	18.97%	16.38%	16.55%	**15.52% †**	20.52%	15.86%
					✓	21.31%	21.93%	**20.91%**	18.47%	**18.35% ‡**	18.47%	20.11%	20.34%
primacy	339	0	25	17	✖	13.51%	**12.99% ‡**	13.51%	**13.13%**	13.43%	13.21%	13.66%	14.18%
					✓	15.10%	**14.80%**	15.03%	**14.38%**	14.58%	14.41%	**14.06%**	14.66%
spect	267	0	22	22	✖	20.28%	18.77%	**18.30%**	19.25%	19.25%	**17.08% ††**	21.04%	21.13%
					✓	23.16%	21.97%	**20.53%**	23.16%	22.34%	**20.34% ††**	21.09%	21.50%
tic-tac-toe	958	0	18	9	✖	1.73%	**1.70%**	**1.70%**	**1.70%**	**1.70%**	**1.70%**	1.70%	1.70%
					✓	2.77%	**1.65%**	**1.65%**	**1.65%**	**1.65%**	**1.65%**	1.65%	1.65%

Table 4: Mean classification error for the **smooth hinge loss function**. We use the same abbreviations as in Table 2.

Dataset	N	M_{x}	M_{z}	K	Reduced?	Nom	MixF ($\kappa_{\mathrm{y}} = 1$)	MixF ($\kappa_{\mathrm{y}} = K$)	r-Nom	r-MixF ($\kappa_{\mathrm{y}} = 1$)	r-MixF ($\kappa_{\mathrm{y}} = K$)	ConF ($\kappa_{\mathrm{y}} = 1$)	ConF ($\kappa_{\mathrm{y}} = K$)
bike	17,379	4	31	5	✖	**210.23**	**210.23**	**210.23**	**210.23**	**210.23**	**210.23**	210.23	210.23
					✓	210.82	**210.81 ††**	210.81	210.84	**210.84**	210.84	213.47	213.60
fire	517	10	31	4	✖	**123.33**	**123.33 ††**	**123.33 ††**	123.48	**123.33 ††**	124.74	149.03	149.04
					✓	132.71	117.42	**111.50 ††**	114.39	113.56	**112.98**	120.41	118.30
flare	1,066	0	21	9	✖	198.10	**198.10**	198.10	198.87	**198.86**	**198.86**	**198.10**	198.10
					✓	199.85	**199.84**	**199.84**	200.09	**200.09**	**200.09**	199.61	199.60
garments	1,197	7	22	4	✖	**363.11**	363.40	366.48	**362.55**	362.57	362.60	363.36	364.38
					✓	890.81	**367.76**	368.95	368.15	**367.29 †**	367.84	367.64	368.97
imports	193	14	45	10	✖	442.71	**289.01**	302.02	311.08	**269.97 ††**	270.94	289.19	300.18
					✓	660.04	377.26	**368.77**	352.79	**334.54 ††**	336.18	384.48	370.93
student	395	13	26	17	✖	**379.13**	379.20	383.07	378.66	**378.12 †**	378.66	379.15	383.80
					✓	**411.98**	412.32	413.51	384.20	**384.07 ††**	384.77	412.11	396.66
vegas	504	5	106	14	✖	232,330.88	533.63	**514.45**	**479.91**	479.92	**479.91 ‡**	520.72	482.06
					✓	287,615.23	605.27	**554.18**	**493.60**	**493.60 ‡**	493.79	581.52	494.27

Table 5: Mean squared errors for the **Huber loss function** (with $\delta = 0.5$). We use the same abbreviations as in Table 2.

Dataset	N	M_{x}	M_{z}	K	Reduced?	Nom	MixF ($\kappa_{\text{y}} = 1$)	MixF ($\kappa_{\text{y}} = K$)	r-Nom	r-MixF ($\kappa_{\text{y}} = 1$)	r-MixF ($\kappa_{\text{y}} = K$)	ConF ($\kappa_{\text{y}} = 1$)	ConF ($\kappa_{\text{y}} = K$)
bike	17,379	4	31	5	✖	217.87	217.87	**217.87**	217.85	**217.82**	217.87	217.86	217.87
					✓	218.17	218.17	**218.16**	218.17	218.18	**218.17**	218.16	218.21
fire	517	10	31	4	✖	124.36	124.36	**124.29 †‡**	130.45	**124.36**	124.45	142.59	124.33
					✓	120.54	110.76	**109.71**	116.67	**109.60 †‡**	**109.60**	109.74	109.81
flare	1,066	0	21	9	✖	**218.56**	218.62	218.87	218.27	219.26	**204.93 †‡**	218.55	218.54
					✓	2,230.29	**750.89**	752.05	215.22	215.22	**202.78 †‡**	215.98	215.80
garments	1,197	7	22	4	✖	374.08	**373.77**	377.32	**373.32**	373.76	374.70	373.79	377.65
					✓	894.07	376.12	**375.77 †**	377.78	377.06	**376.58**	376.05	377.07
imports	193	14	45	10	✖	671.67	**299.63**	305.48	351.66	292.86	**289.26 †‡**	311.48	319.25
					✓	726.46	**379.16**	408.11	371.56	**348.24 †‡**	349.712	406.18	389.03
student	395	13	26	17	✖	399.65	399.65	**397.63**	**383.57**	**383.57 ‡**	386.14	399.66	405.33
					✓	451.85	451.89	**397.59 †‡**	403.51	398.50	**393.65**	451.95	408.27
vegas	504	5	106	14	✖	232,332.44	529.55	**500.85**	503.37	503.37	**490.76 †‡**	538.10	503.46
					✓	287,597.53	580.67	**505.11**	507.74	507.74	**500.22 †‡**	592.46	511.80

Table 6: Mean squared errors for the **pinball loss function** (with $\tau = 0.5$). We use the same abbreviations as in Table 2.

Dataset	N	M_{x}	M_{z}	K	Reduced?	Nom	MixF ($\kappa_{\text{y}} = 1$)	MixF ($\kappa_{\text{y}} = K$)	r-Nom	r-MixF ($\kappa_{\text{y}} = 1$)	r-MixF ($\kappa_{\text{y}} = K$)	ConF ($\kappa_{\text{y}} = 1$)	ConF ($\kappa_{\text{y}} = K$)
bike	17,379	4	31	5	✖	**217.87**	217.87	**217.87**	217.88	**217.87**	217.88	217.86	217.87
					✓	218.18	218.17	**218.16**	**218.15**	**218.15**	218.15	218.14	218.20
fire	517	10	31	4	✖	133.10	133.02	**132.64**	136.26	132.90	**132.67 †‡**	133.05	132.82
					✓	185.04	128.05	**124.38 ‡**	125.23	125.30	**125.02**	126.67	124.93
flare	1,066	0	21	9	✖	**212.93**	212.95	213.20	215.49	215.40	**211.44 †‡**	212.92	212.94
					✓	1,562.88	532.37	**359.82**	232.67	210.53	**207.05 †‡**	209.49	209.28
garments	1,197	7	22	4	✖	369.10	**368.65 †**	371.72	370.08	**369.07**	370.17	368.65	371.67
					✓	871.46	**370.90**	372.19	370.72	370.77	**370.68**	371.17	372.03
imports	193	14	45	10	✖	621.38	**288.81 †‡**	300.38	331.36	300.22	**299.10**	306.61	322.70
					✓	761.05	**370.96**	394.00	370.97	338.80	**338.07 †‡**	392.69	380.75
student	395	13	26	17	✖	399.65	399.66	**397.60**	387.07	386.27	**385.75 ‡**	399.68	405.34
					✓	451.85	451.89	**397.59**	403.51	398.50	**393.65 †‡**	451.95	408.27
vegas	504	5	106	14	✖	232,323.47	530.00	**500.36**	493.28	493.29	**489.17 †‡**	538.51	503.48
					✓	287,607.59	580.52	**504.85**	517.85	517.86	**497.64 †‡**	592.31	511.76

Table 7: Mean squared errors for the **τ-insensitive loss function** (with $\tau = 0.1$). We use the same abbreviations as in Table 2.

References

Alley, M., M. Biggs, R. Hariss, C. Herrmann, M. L. Li, and G. Perakis (2023). Pricing for heterogeneous products: Analytics for ticket reselling. *Manufacturing & Service Operations Management 25*(2), 409–426.

Ban, G.-Y. and C. Rudin (2019). The big data newsvendor: Practical insights from machine learning. *Operations Research 67*(1), 90–108.

Bastani, H. and M. Bayati (2020). Online decision making with high-dimensional covariates. *Operations Research 68*(1), 276–294.

Behrendt, A., M. Savelsbergh, and H. Wang (2023). A prescriptive machine learning method for courier scheduling on crowdsourced delivery platforms. *Transportation Science 57*(4), 889–907.

Ben-Tal, A., L. E. Ghaoui, and A. Nemirovski (2009). *Robust Optimization*. Princeton University Press.

Bertsimas, D., A. Delarue, P. Jaillet, and S. Martin (2019). Travel time estimation in the age of big data. *Operations Research 67*(2), 498–515.

Bertsimas, D. and D. den Hertog (2022). *Robust and Adaptive Optimization*. Dynamic Ideas.

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

Bertsimas, D., A. O'Hair, S. Relyea, and J. Silberholz (2016). An analytics approach to designing combination chemotherapy regimens for cancer. *Management Science 62*(5), 1511–1531.

Bertsimas, D. and J. Pauphilet (2023). Hospital-wide inpatient flow optimization. *Management Science*, Available ahead of print.

Blanchet, J. and Y. Kang (2021). Sample out-of-sample inference based on Wasserstein distance. *Operations Research 69*(3), 985–1013.

Blanchet, J., Y. Kang, and K. Murthy (2019). Robust Wasserstein profile inference and applications to machine learning. *Journal of Applied Probability 56*(3), 830–857.

Blanchet, J. and K. Murthy (2019). Quantifying distributional model risk via optimal transport. *Mathematics of Operations Research 44*(2), 565–600.

Chan, T. C. Y., R. Mahmood, D. L. O'Connor, D. Stone, S. Unger, R. K. Wong, and I. Y. Zhu (2023). Got (optimal) milk? Pooling donations in human milk banks with machine learning and optimization. *Manufacturing & Service Operations Management*, Available ahead of print.

Duchi, J., T. Hashimoto, and H. Namkoong (2023). Distributionally robust losses for latent covariate mixtures. *Operations Research 71*(2), 649–664.

Feldman, J., D. J. Zhang, X. Liu, and N. Zhang (2022). Customer choice models vs. machine learning: Finding optimal product displays on Alibaba. *Operations Research 70*(1), 309–328.

Ferreira, K. J., B. H. A. Lee, and D. Simchi-Levi (2016). Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management 18*(1), 69–88.

Gao, R. (2022). Finite-sample guarantees for Wasserstein distributionally robust optimization: Breaking the curse of dimensionality. *Operations Research*, Available ahead of print.

Gao, R., X. Chen, and A. J. Kleywegt (2022). Wasserstein distributionally robust optimization and variation regularization. *Operations Research*, Available ahead of print.

Gao, R. and A. Kleywegt (2023). Distributionally robust stochastic optimization with Wasserstein distance. *Mathematics of Operations Research 48*(2), 603–655.

Glaeser, C. K., M. Fisher, and X. Su (2019). Optimal retail location: Empirical methodology and application to practice. *Manufacturing & Service Operations Management 21*(1), 86–102.

Grötschel, M., L. Lovász, and A. Schrijver (1988). *Geometric Algorithms and Combinatorial Optimization*. Springer.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer.

Kallus, N. and M. Udell (2020). Dynamic assortment personalization in high dimensions. *Operations Research 68*(4), 1020–1037.

Kelly, M., R. Longjohn, and K. Nottingham (2017). The UCI machine learning repository. `https://archive.ics.uci.edu`.

Kuhn, D., P. Mohajerin Esfahani, V. Nguyen, and S. Shafieezadeh-Abadeh (2019). Wasserstein distributionally robust optimization: Theory and applications in machine learning. *INFORMS TutORials in Operations Research*, 130–169.

Lam, H. (2019). Recovering best statistical guarantees via the empirical divergence-based distributionally robust optimization. *Operations Research 6*(4), 1090–1105.

Li, R., M. Tobey, M. E. Mayorga, S. Caltagirone, and O. Y. Özaltın (2023). Detecting human trafficking: Automated classification of online customer reviews of massage businesses. *Manufacturing & Service Operations Management 25*(3), 1051–1065.

Michaud, R. O. (1989). The Markowitz optimization enigma: Is 'optimized'optimal? *Financial Analysts Journal 45*(1), 31–42.

Mohajerin Esfahani, P. and D. Kuhn (2018). Data-driven distributionally robust optimization using the Wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming 171*(1–2), 1–52.

Murphy, K. P. (2022). *Probabilistic Machine Learning: An Introduction*. MIT Press.

Qi, M., Y. Cao, and Z.-J. Shen (2022). Distributionally robust conditional quantile prediction with fixed design. *Management Science 68*(3), 1639–1658.

Rahimian, H. and S. Mehrotra (2022). Frameworks and results in distributionally robust optimization. *Open Journal of Mathematical Optimization 3*, 1–85.

Samorani, M., S. L. Harris, L. G. Blount, H. Lu, and M. A. Santoro (2022). Overbooked and overlooked: Machine learning and racial bias in medical appointment scheduling. *Manufacturing & Service Operations Management 24*(6), 2825–2842.

Selvi, A., M. Belbasi, M. Haugh, and W. Wiesemann (2022). Wasserstein logistic regression with mixed features. In *Advances in Neural Information Processing Systems*, Volume 35, pp. 16691–16704.

Shafieezadeh-Abadeh, S., D. Kuhn, and P. Mohajerin Esfahani (2019). Regularization via mass transportation. *Journal of Machine Learning Research 20*(103), 1–68.

Shafieezadeh-Abadeh, S., P. Mohajerin Esfahani, and D. Kuhn (2015). Distributionally robust logistic regression. In *Advances in Neural Information Processing Systems*, Volume 28, pp. 1576–1584.

Smith, J. E. and R. L. Winkler (2006). The optimizer's curse: Skepticism and postdecision surprise in decision analysis. *Management Science 52*(3), 311–322.

Van Parys, B. P. G., P. Mohajerin Esfahani, and D. Kuhn (2021). From data to decisions: Distributionally robust optimization is optimal. *Management Science 67*(6), 3387–3402.

Appendix: Proofs

Proof of Observation 1. The statement follows from similar arguments as in the proof of Theorem 1 by Shafieezadeh-Abadeh et al. (2015). Details are omitted for the sake of brevity. □

Our proofs of Theorems 1 and 2 rely on findings presented in later parts of the paper. Readers may thus find it easier to read those proofs after familiarizing themselves with these later results.

Proof of Theorem 1. The first statement is proved in Theorem 2 of Selvi et al. (2022).

For the second statement, Sections 3 and 4 derive the equivalent finite-dimensional reformulations (4)–(7) of the Wasserstein learning problem (1) for convex Lipschitz continuous and piece-wise affine loss functions L in classification and regression settings, and Section 5 develops the unified representation (8) that encompasses (4)–(7) as special cases. The second statement of the theorem is therefore proven if we can demonstrate the existence of a polynomial time solution scheme for our unified representation (8) when specialized to the formulations (4)–(7). Grötschel et al. (1988, Corollary 4.2.7) show that problem (8) can be solved to δ-accuracy in polynomial time if the problem admits a polynomial time weak separation oracle and the feasible region of the problem is a circumscribed convex body. We prove both of these properties next.

Fix a convex and compact set $\mathcal{K} \subseteq \mathbb{R}^n$ and a rational number $\delta > 0$. Grötschel et al. (1988, Definition 2.1.13) define a weak separation oracle for $\mathcal{K}$ as an algorithm which for any vector $\boldsymbol{q} \in \mathbb{R}^n$ either confirms that $\boldsymbol{q} \in \mathcal{S}(\mathcal{K}, \delta)$, where $\mathcal{S}(\mathcal{K}, \delta) = \{\boldsymbol{r} \in \mathbb{R}^n : \|\boldsymbol{r} - \boldsymbol{r}'\|_2 \leqslant \delta \text{ for some } \boldsymbol{r}' \in \mathcal{K}\}$ is the δ-enclosure around $\mathcal{K}$ (*i.e.*, $\boldsymbol{q}$ is *almost* in $\mathcal{K}$), or finds a vector $\boldsymbol{c} \in \mathbb{R}^n$ with $\|\boldsymbol{c}\|_\infty = 1$ such that $\boldsymbol{c}^\top \boldsymbol{p} \leqslant \boldsymbol{c}^\top \boldsymbol{q} + \delta$ for all $\boldsymbol{p} \in \mathcal{S}(\mathcal{K}, -\delta)$, where $\mathcal{S}(\mathcal{K}, -\delta) = \{\boldsymbol{r} \in \mathcal{K} : \mathcal{S}(\{\boldsymbol{r}\}, \delta) \subseteq \mathcal{K}\}$ is the δ-interior of $\mathcal{K}$ (*i.e.*, $\boldsymbol{c}$ is an *almost* separating hyperplane). It follows from Theorem 3 in Section 5 that Algorithm 2 is such a polynomial time weak separation oracle for problem (8).

According to Grötschel et al. (1988, Definition 2.1.16), the feasible region $\mathcal{K} \subseteq \Theta \times \mathbb{R}^N_+$ of problem (8) is a circumscribed convex body whenever Θ is finite-dimensional and $\mathcal{K}$ is a

full-dimensional, compact and convex subset of a ball whose finite radius we can specify. It follows from the construction of our problems (4)–(7) that in those special cases, Θ in problem (8) is indeed finite-dimensional and $\mathcal{K}$ is full-dimensional, closed and convex. Our claim would thus follow if $\mathcal{K}$ was circumscribed by a ball whose finite radius we can specify. While this is not the case *per se*, we will show that we can add constraints to the feasible region $\mathcal{K}$ that do not affect the set of optimal solutions but that allow us to circumscribe $\mathcal{K}$ as desired. We will show this in two steps. We first confirm that in the special cases where our unified representation (8) is used to describe the learning problems (4)–(7), $\mathcal{K}$ can be circumscribed by a suitable ball whenever Θ is bounded. Afterwards, we show that Θ can be bounded in the special cases where our unified representation (8) is used to describe the learning problems (4)–(7) for convex Lipschitz continuous and piece-wise affine loss functions L and a bounded hypothesis set for $\boldsymbol{\beta}$.

To see that $\mathcal{K}$ can be circumscribed by a ball whose finite radius we can specify whenever Θ is bounded, we note that $\boldsymbol{\sigma}$ in problem (8) is non-negative by construction. Since the objective function in (8) is non-decreasing in $\boldsymbol{\sigma}$ in our special cases (4)–(7), we can without loss of generality include in (8) the additional constraints $\sigma_n \leqslant \overline{\sigma}_n$ for

$$\overline{\sigma}_n \;=\; \max_{\boldsymbol{\theta}\in\Theta}\; \max_{i\in\mathcal{I}}\; \max_{\boldsymbol{z}\in\mathbb{Z}} \left\{ f_{ni}(g_{ni}(\boldsymbol{\theta}, \boldsymbol{\xi}^n_{-\mathbf{z}}; \boldsymbol{z})) - h_{ni}(\boldsymbol{\theta}, \boldsymbol{\xi}^n_{-\mathbf{z}}; d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n)) \right\} \qquad \forall n \in [N].$$

Note that all $\overline{\sigma}_n$ are finite since Θ is bounded by assumption, $\mathcal{I}$ and $\mathbb{Z}$ are finite sets and the objective function is continuous thanks to the convexity assumptions of Section 5. We thus conclude that the boundedness of Θ allows us to bound $\boldsymbol{\sigma}$ in problem (8) as well, and thus $\mathcal{K}$ can indeed be circumscribed by a ball whose finite radius we can specify whenever Θ is bounded.

We next show that Θ can be bounded if our unified representation (8) is used to describe the classification problem (4) from Section 3 for convex Lipschitz continuous loss functions L.

To this end, observe that problem (8) recovers problem (4) if we set

$$\Theta \;=\; \{(\beta_0, \boldsymbol{\beta}_{\mathrm{x}}, \boldsymbol{\beta}_{\mathrm{z}}, \lambda) \,:\, \mathrm{lip}(L) \cdot \|\boldsymbol{\beta}_{\mathrm{x}}\|_* \leqslant \lambda,\;\; \lambda \in \mathbb{R}_+\} \;\;\text{and}\;\; \boldsymbol{\sigma} = \boldsymbol{s},$$

as well as $i \equiv y$ with $\mathcal{I} = \{\pm y^n\}$ and

$$f_0(\boldsymbol{\theta}, \boldsymbol{\sigma}) \;=\; \lambda\epsilon + \frac{1}{N}\sum_{n\in[N]} \sigma_n,\;\; f_{ni}(e) \;=\; L(e),\;\; g_{ni}(\boldsymbol{\theta}, \boldsymbol{\xi}^n_{-\mathbf{z}}; \boldsymbol{z}) \;=\; i \cdot (\beta_0 + \boldsymbol{\beta}_{\mathrm{x}}{}^\top \boldsymbol{x}^n + \boldsymbol{\beta}_{\mathrm{z}}{}^\top \boldsymbol{z})$$
$$\text{and}\;\; h_{ni}(\boldsymbol{\theta}, \boldsymbol{\xi}^n_{-\mathbf{z}}; d) \;=\; \lambda\kappa_{\mathrm{z}} d + \lambda\kappa_{\mathrm{y}} d_{\mathrm{y}}(i, y^n).$$

By definition, λ is lower bounded by 0. To see that we can bound λ from above as well, we observe that there are optimal solutions for which λ does not exceed $\overline{\lambda} = \max\{\overline{\lambda}_1, \overline{\lambda}_2\}$, where

$$\overline{\lambda}_1 \;=\; \max_{\boldsymbol{\beta}\in\mathcal{H}} \mathrm{lip}(L) \cdot \|\boldsymbol{\beta}_{\mathrm{x}}\|_*$$

with the bounded set $\mathcal{H} \subseteq \mathbb{R}^{1+M_{\boldsymbol{x}}+M_{\boldsymbol{z}}}$ containing all admissible hypotheses $\boldsymbol{\beta}$, and

$$\overline{\lambda}_2 \;=\; \max_{\boldsymbol{\beta}\in\mathcal{H}} \max_{n\in[N]} \max_{i\in\mathcal{I}} \max_{\boldsymbol{z}\in\mathbb{Z}} \left\{ \frac{l_{\boldsymbol{\beta}}(\boldsymbol{x}^n, \boldsymbol{z}, i)}{\kappa_{\mathrm{z}} d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n) + \kappa_{\mathrm{y}} d_{\mathrm{y}}(i, y^n)} \;:\; (\boldsymbol{z}, i) \neq (\boldsymbol{z}^n, y^n) \right\}. \qquad \text{(A-1)}$$

Indeed, selecting $\lambda \geqslant \overline{\lambda}_1$ ensures that all hypotheses $\boldsymbol{\beta} \in \mathcal{H}$ are represented in Θ, and selecting $\lambda \geqslant \overline{\lambda}_2$ implies that for any $\boldsymbol{\beta} \in \mathcal{H}$, all left-hand sides of the constraints

$$f_{ni}(g_{ni}(\boldsymbol{\theta}, \boldsymbol{\xi}^n_{-\mathbf{z}}; \boldsymbol{z})) - h_{ni}(\boldsymbol{\theta}, \boldsymbol{\xi}^n_{-\mathbf{z}}; d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n)) \leqslant \sigma_n \qquad \forall n \in [N],\; \forall i \in \mathcal{I},\; \forall \boldsymbol{z} \in \mathbb{Z}$$

in problem (8) that involve λ are non-positive, and thus all of these constraints are weakly dominated by the non-negativity constraints on $\boldsymbol{\sigma}$. Note that the numerator in the objective function of (A-1) is bounded since all maxima in (A-1) operate over bounded sets and $l_{\boldsymbol{\beta}}$ is Lipschitz continuous. Moreover, the denominator in the objective function of (A-1) is bounded

from below by a strictly positive quantity since $(\boldsymbol{z}, i) \neq (\boldsymbol{z}^n, y^n)$ and $\kappa_{\mathrm{z}}, \kappa_{\mathrm{y}} > 0$. We thus conclude that when specialized to problem (4), Θ can be bounded in (8). Similar arguments show that Θ can also be bounded if (8) is used to describe the regression problem (6) for convex Lipschitz continuous loss functions; we omit the proof of this statement for the sake of brevity.

We next show that Θ can be bounded if our unified representation (8) is used to describe the classification problem (5) from Section 3 for piece-wise affine loss functions L. To this end, observe that problem (8) recovers problem (5) if we set

$$\begin{aligned}\Theta \;=\; \big\{(\beta_0, \boldsymbol{\beta}_{\mathrm{x}}, \boldsymbol{\beta}_{\mathrm{z}}, \lambda, \boldsymbol{q}_{ni}) \,:\, \|a_j y \cdot \boldsymbol{\beta}_{\mathrm{x}} - \boldsymbol{C}^\top \boldsymbol{q}_{ni}\|_* \leqslant \lambda \;\; \forall n \in [N],\; \forall i = (j, y) \in \mathcal{I},\\ \boldsymbol{q}_{ni} \in \mathcal{C}^* \;\; \forall n \in [N],\; \forall i \in \mathcal{I},\;\; \lambda \in \mathbb{R}_+\big\} \;\text{ and }\; \boldsymbol{\sigma} = \boldsymbol{s},\end{aligned}$$

as well as $i \equiv (j, y)$, $\mathcal{I} = [J] \times \{\pm y^n\}$ and

$$\begin{aligned}f_0(\boldsymbol{\theta}, \boldsymbol{\sigma}) \;=\; \lambda\epsilon + \frac{1}{N}\sum_{n\in[N]} \sigma_n,\;\; f_{ni}(e) \;=\; a_j e + b_j,\;\; g_{ni}(\boldsymbol{\theta}, \boldsymbol{\xi}^n_{-\mathbf{z}}; \boldsymbol{z}) \;=\; y \cdot (\beta_0 + \boldsymbol{\beta}_{\mathrm{x}}{}^\top \boldsymbol{x}^n + \boldsymbol{\beta}_{\mathrm{z}}{}^\top \boldsymbol{z})\\ \text{and }\; h_{ni}(\boldsymbol{\theta}, \boldsymbol{\xi}^n_{-\mathbf{z}}; d) \;=\; -\boldsymbol{q}_{ni}^\top(\boldsymbol{d} - \boldsymbol{C}\boldsymbol{x}^n) + \lambda\kappa_{\mathrm{z}} d + \lambda\kappa_{\mathrm{y}} d_{\mathrm{y}}(y, y^n).\end{aligned}$$

Our proof that Θ can be bounded proceeds in two steps. We first show that without loss of generality, we can impose bounds on each variable $\boldsymbol{q}_{ni}$, $n \in [N]$ and $i \in \mathcal{I}$, and we afterwards show that we can impose non-restrictive bounds on λ as well.

To see that each $\boldsymbol{q}_{ni}$ can be bounded, $n \in [N]$ and $i \in \mathcal{I}$, we proceed in two steps. We first argue that $\boldsymbol{q}_{ni}{}^\top(\boldsymbol{d} - \boldsymbol{C}\boldsymbol{x}^n) \geqslant 0$ for all n and i, that is, larger values of $\boldsymbol{q}_{ni}$ weakly increase the left-hand sides in the first constraint set of (8). Due to the non-negativity of f_0 in $\boldsymbol{\sigma}$ in our special cases (4)–(7), larger values of $\boldsymbol{q}_{ni}$ thus weakly increase the objective function in (8). Non-zero values for $\boldsymbol{q}_{ni}$ can therefore only be optimal if they allow to reduce λ via the constraints $\|a_j y \cdot \boldsymbol{\beta}_{\mathrm{x}} - \boldsymbol{C}^\top \boldsymbol{q}_{ni}\|_* \leqslant \lambda$ in Θ. We then derive a bounded set $\mathcal{Q}$ such that

$\|a_j y \cdot \boldsymbol{\beta}_{\mathrm{x}} - \boldsymbol{C}^\top \boldsymbol{q}_{ni}\|_* > \|a_j y \cdot \boldsymbol{\beta}_{\mathrm{x}}\|_*$ for all $n \in [N]$, $i = (j, y) \in \mathcal{I}$, all admissible $\boldsymbol{\beta}$ and all $\boldsymbol{q}_{ni} \notin \mathcal{Q}$, that is, the choice $\boldsymbol{q}_{ni} = \boldsymbol{0} \in \mathcal{C}^*$ dominates any feasible choice of $\boldsymbol{q}_{ni}$ outside of $\mathcal{Q}$. In view of the first step, note that $\boldsymbol{d} - \boldsymbol{C}\boldsymbol{x}^n \in \mathcal{C}$ by construction of $\mathbb{X}$ and the fact that $\boldsymbol{x}^n \in \mathbb{X}$. We thus have ${\boldsymbol{q}_{ni}}^\top(\boldsymbol{d} - \boldsymbol{C}\boldsymbol{x}^n) \geqslant 0$ since $\boldsymbol{q}_{ni} \in \mathcal{C}^*$. As for the second step, consider for each n and i the orthogonal decomposition of the vectors $\boldsymbol{q}_{ni} \in \mathcal{C}^*$ into $\boldsymbol{q}_{ni} = \boldsymbol{q}^0_{ni} + \boldsymbol{q}^+_{ni}$ where $\boldsymbol{q}^0_{ni} \in \mathrm{Null}(\boldsymbol{C}^\top)$ is in the nullspace of $\boldsymbol{C}^\top$ and $\boldsymbol{q}^+_{ni} \in \mathrm{Row}(\boldsymbol{C}^\top)$ is in the row space of $\boldsymbol{C}^\top$. There is a bounded set $\mathcal{Q}^+ \subseteq \mathrm{Row}(\boldsymbol{C}^\top)$ such that $\|a_j y \cdot \boldsymbol{\beta}_{\mathrm{x}} - \boldsymbol{C}^\top \boldsymbol{q}^+_{ni}\|_* > \|a_j y \cdot \boldsymbol{\beta}_{\mathrm{x}}\|_*$ for all $n \in [N]$, $i = (j, y) \in \mathcal{I}$, all admissible $\boldsymbol{\beta}$ and all $\boldsymbol{q}^+_{ni} \notin \mathcal{Q}^+$; note in particular that $\|a_j y \cdot \boldsymbol{\beta}_{\mathrm{x}}\|_*$ is bounded due to the assumed boundedness of the hypothesis set and the fact that $\mathcal{I}$ and $\mathbb{Z}$ are finite sets. Similarly, there is a bounded set $\mathcal{Q}^0 \subseteq \mathrm{Null}(\boldsymbol{C}^\top)$ such that for all $\boldsymbol{q}'_{ni} \in \mathrm{Null}(\boldsymbol{C}^\top) \backslash \mathcal{Q}^0$ satisfying $\boldsymbol{q}^+_{ni} + \boldsymbol{q}'_{ni} \in \mathcal{C}^*$ for some $\boldsymbol{q}^+_{ni} \in \mathcal{Q}^+$ there is $\boldsymbol{q}^0_{ni} \in \mathcal{Q}^0$ satisfying $\boldsymbol{q}^+_{ni} + \boldsymbol{q}^0_{ni} \in \mathcal{C}^*$ such that $(\boldsymbol{q}^+_{ni} + \boldsymbol{q}^0_{ni})^\top(\boldsymbol{d} - \boldsymbol{C}\boldsymbol{x}^n) \leqslant (\boldsymbol{q}^+_{ni} + \boldsymbol{q}'_{ni})^\top(\boldsymbol{d} - \boldsymbol{C}\boldsymbol{x}^n)$, that is, ${\boldsymbol{q}^0_{ni}}^\top \boldsymbol{d} \leqslant {\boldsymbol{q}'_{ni}}^\top \boldsymbol{d}$, across all $n \in [N]$ and $i = (j, y) \in \mathcal{I}$. Thus, we can without loss of generality restrict the choice of $\boldsymbol{q}_{ni}$ to the bounded set $\mathcal{Q} = \mathcal{C}^* \cap (\mathcal{Q}^0 + \mathcal{Q}^+)$, where the sum is taken in the Minkowski sense.

To see that λ can be bounded as well, note that by construction, λ is bounded from below by 0. To see that we can bound λ from above as well, we observe that there are optimal solutions to problem (8) for which λ does not exceed $\overline{\lambda} = \max\{\overline{\lambda}_1, \overline{\lambda}_2\}$, where

$$\overline{\lambda}_1 \;=\; \max_{\boldsymbol{\beta} \in \mathcal{H}} \; \max_{n \in [N]} \; \max_{i=(j,y) \in \mathcal{I}} \; \max_{\boldsymbol{q}_{ni} \in \mathcal{Q}} \; \|a_j y \cdot \boldsymbol{\beta}_{\mathrm{x}} - \boldsymbol{C}^\top \boldsymbol{q}_{ni}\|_*$$

with the bounded set $\mathcal{H} \subseteq \mathbb{R}^{1+M_{\boldsymbol{x}}+M_{\boldsymbol{z}}}$ containing all admissible hypotheses $\boldsymbol{\beta}$, and

$$\overline{\lambda}_2 \;=\; \max_{\boldsymbol{\beta} \in \mathcal{H}} \; \max_{n \in [N]} \; \max_{i=(j,y) \in \mathcal{I}} \; \max_{\boldsymbol{z} \in \mathbb{Z}} \; \max_{\boldsymbol{q}_{ni} \in \mathcal{Q}} \left\{ \frac{{\boldsymbol{q}_{ni}}^\top(\boldsymbol{d} - \boldsymbol{C}\boldsymbol{x}^n) + a_j y \cdot (\beta_0 + {\boldsymbol{\beta}_{\mathrm{x}}}^\top \boldsymbol{x}^n + {\boldsymbol{\beta}_{\mathrm{z}}}^\top \boldsymbol{z}) + b_j}{\kappa_{\mathrm{z}} d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n) + \kappa_{\mathrm{y}} d_{\mathrm{y}}(y, y^n)} \right. \\ \left. : \; (\boldsymbol{z}, y) \neq (\boldsymbol{z}^n, y^n) \right\}.$$

As before, selecting $\lambda \geqslant \overline{\lambda}_1$ ensures that all hypotheses $\boldsymbol{\beta} \in \mathcal{H}$ are represented in Θ, and selecting $\lambda \geqslant \overline{\lambda}_2$ implies that for any $\boldsymbol{\beta} \in \mathcal{H}$, all left-hand sides of the constraints

$$f_{ni}(g_{ni}(\boldsymbol{\theta}, \boldsymbol{\xi}^n_{-\mathbf{z}}; \boldsymbol{z})) - h_{ni}(\boldsymbol{\theta}, \boldsymbol{\xi}^n_{-\mathbf{z}}; d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n)) \leqslant \sigma_n \qquad \forall n \in [N],\ \forall i \in \mathcal{I},\ \forall \boldsymbol{z} \in \mathbb{Z}$$

in problem (8) that involve λ are non-positive, and thus all of these constraints are weakly dominated by the non-negativity constraints on $\boldsymbol{\sigma}$. Similar arguments as before, combined with the fact that $\mathcal{Q}$ is bounded, show that $\overline{\lambda}_1$ and $\overline{\lambda}_2$ are finite, and thus Θ can indeed be bounded in (8). Finally, a similar reasoning also confirms that Θ can be bounded if (8) describes the regression problem (7) for convex piece-wise affine loss functions. □

Proof of Theorem 2. Fix any loss function L satisfying the conditions in the statement of the theorem, and consider a Wasserstein classification or regression instance with ambiguity radius $\epsilon > \kappa_{\mathrm{z}}$, $M_{\mathrm{x}} = 1$ numerical feature, $K = 1$ binary feature (that is, $k_1 = 2$), and $\|\cdot\| = |\cdot|$ for the ground metric d from Definition 1. The training set comprises $N = 1$ sample $\boldsymbol{\xi}^1 = (x^1, z^1, y^1)$ with $x^1 \in \mathbb{X} = \mathbb{R}$ specified below, $z^1 = 0$ and $y^1 = 1$. Our proof proceeds in three steps. We first derive a closed-form expression for the objective function of the Wasserstein learning problem (1) at a judiciously chosen learning model $\hat{\boldsymbol{\beta}}$. Our derivation will show that this objective function constitutes the sum of the empirical loss $l_{\hat{\boldsymbol{\beta}}}(\boldsymbol{\xi}^1)$ and a function $h_{\hat{\boldsymbol{\beta}}}(x^1)$. We then construct two points $\hat{x}^1$ and $\check{x}^1$ at which $h_{\hat{\boldsymbol{\beta}}}(\hat{x}^1) \neq h_{\hat{\boldsymbol{\beta}}}(\check{x}^1)$, showing that $h_{\hat{\boldsymbol{\beta}}}(x^1)$ exhibits a dependence on x^1 that cannot be recovered by any data-agnostic regularizer $\mathfrak{R}(\hat{\boldsymbol{\beta}})$.

Fix the learning model $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_{\mathrm{x}}, \hat{\beta}_{\mathrm{z}})$ with $\hat{\beta}_0 \in \mathbb{R}$ and $\hat{\beta}_{\mathrm{x}} \neq 0$ selected arbitrarily, where $\hat{\beta}_{\mathrm{z}}$ is chosen so as to satisfy $d'\hat{\beta}_{\mathrm{z}} > \kappa_{\mathrm{z}} \mathrm{lip}(L) \cdot |\hat{\beta}_{\mathrm{x}}|$. Here, d' is the derivative of the loss function $d' = (\mathrm{d}/\mathrm{d}e)L(e)\,\big|_{e=x'}$ at any point $x' \in \mathbb{R}$ where the derivative does not vanish. Such points x' exist due to Rademacher's theorem, which ensures that a Lipschitz continuous function is differentiable almost everywhere, as well as the assumption that the loss function L is non-

constant. Note that piece-wise affine loss functions are Lipschitz continuous and that $\mathbb{X} = \mathbb{R}$. For the described problem instance, Proposition 1 in Section 3 will therefore imply that the objective function of the Wasserstein classification problem becomes

$$\begin{array}{lll} \underset{\lambda, s_1}{\text{minimize}} & \lambda\epsilon + s_1 & \\ \text{subject to} & l_{\hat{\boldsymbol{\beta}}}(x^1, z, 1) - \lambda\kappa_{\text{z}} z \leqslant s_1 & \forall z \in \mathbb{B} \\ & l_{\hat{\boldsymbol{\beta}}}(x^1, z, -1) - \lambda\kappa_{\text{z}} z - \lambda\kappa_{\text{y}} \leqslant s_1 & \forall z \in \mathbb{B} \\ & \text{lip}(L) \cdot |\hat{\beta}_{\text{x}}| \leqslant \lambda & \\ & \lambda \in \mathbb{R}_+, \;\; s_1 \in \mathbb{R}_+, & \end{array}$$

and Proposition 3 in Section 4 will imply that the objective function of the Wasserstein regression problem becomes

$$\begin{array}{lll} \underset{\lambda, s_1}{\text{minimize}} & \lambda\epsilon + s_1 & \\ \text{subject to} & l_{\hat{\boldsymbol{\beta}}}(x^1, z, 1) - \lambda\kappa_{\text{z}} z \leqslant s_1 & \forall z \in \mathbb{B} \\ & \text{lip}(L) \cdot |\hat{\beta}_{\text{x}}| \leqslant \lambda & \\ & \text{lip}(L) \leqslant \lambda\kappa_{\text{y}} & \\ & \lambda \in \mathbb{R}_+, \;\; s_1 \in \mathbb{R}_+. & \end{array}$$

When $\kappa_{\text{y}} \to \infty$, the second constraint in the classification problem and the third constraint in the regression problem become redundant since $\lambda \geqslant \text{lip}(L) \cdot |\hat{\beta}_{\text{x}}| > 0$ because $\hat{\beta}_{\text{x}} \neq 0$. We can then replace s_1 with the left-hand side of the first constraint in either problem to obtain the unified formulation

$$\begin{array}{ll} \underset{\lambda}{\text{minimize}} & \lambda\epsilon + \max\left\{ l_{\hat{\boldsymbol{\beta}}}(x^1, 0, 1),\; l_{\hat{\boldsymbol{\beta}}}(x^1, 1, 1) - \lambda\kappa_{\text{z}} \right\} \\ \text{subject to} & \text{lip}(L) \cdot |\hat{\beta}_{\text{x}}| \leqslant \lambda \\ & \lambda \in \mathbb{R}_+ \end{array}$$

of the objective function of both the classification and the regression problem. Note that the non-negativity of s_1 is preserved in the unified formulation since $l_{\hat{\boldsymbol{\beta}}}(x^1, 0, 1) \geqslant 0$ by definition of the loss function L that underlies $l_{\hat{\boldsymbol{\beta}}}$. We claim that $\lambda^\star = \mathrm{lip}(L) \cdot |\hat{\beta}_{\mathrm{x}}|$ at optimality. Indeed, any increment $\Delta\lambda > 0$ in λ will cause an increase of $\Delta\lambda \cdot \epsilon$ and a maximum decrease of $\Delta\lambda \cdot \kappa_{\mathrm{z}}$ in the objective function, and we have $\epsilon > \kappa_{\mathrm{z}}$ by assumption. Hence, the objective function of the Wasserstein learning problem simplifies to

$$f_1(\hat{\boldsymbol{\beta}}, x^1) \;=\; \lambda^\star\epsilon + \max\left\{l_{\hat{\boldsymbol{\beta}}}(x^1, 0, 1),\; l_{\hat{\boldsymbol{\beta}}}(x^1, 1, 1) - \lambda^\star\kappa_{\mathrm{z}}\right\} \;=\; l_{\hat{\boldsymbol{\beta}}}(x^1, 0, 1) + h_{\hat{\boldsymbol{\beta}}}(x^1),$$

where

$$h_{\hat{\boldsymbol{\beta}}}(x^1) \;=\; \lambda^\star\epsilon + \max\left\{l_{\hat{\boldsymbol{\beta}}}(x^1, 1, 1) - l_{\hat{\boldsymbol{\beta}}}(x^1, 0, 1) - \lambda^\star\kappa_{\mathrm{z}},\; 0\right\}.$$

In contrast, the objective function of a generic regularized learning problem has the form

$$f_2(\hat{\boldsymbol{\beta}}, x^1) \;=\; l_{\hat{\boldsymbol{\beta}}}(x^1, 0, 1) + \mathfrak{R}(\hat{\boldsymbol{\beta}}).$$

By construction, $\mathfrak{R}(\hat{\boldsymbol{\beta}})$ does not vary with x^1. In contrast, we claim that $h_{\hat{\boldsymbol{\beta}}}(x^1)$ varies with x^1. To this end, we will construct two points $\hat{x}^1$ and $\check{x}^1$ at which the first term inside the maximum in the definition of h is strictly positive and strictly negative, respectively.

We choose the point $\hat{x}^1$ at which the first term inside the maximum in the definition of h is strictly positive such that $\hat{\beta}_0 + \hat{\beta}_{\mathrm{x}}\hat{x}^1 = x'$ for classification problems and $\hat{\beta}_0 + \hat{\beta}_{\mathrm{x}}\hat{x}^1 - 1 = x'$ for regression problems, respectively, which is always possible since $\hat{\beta}_{\mathrm{x}} \neq 0$. We then have

$$\begin{aligned} l_{\hat{\boldsymbol{\beta}}}(\hat{x}^1, 1, 1) - l_{\hat{\boldsymbol{\beta}}}(\hat{x}^1, 0, 1) - \lambda^\star\kappa_{\mathrm{z}} \;&=\; L(x' + \hat{\beta}_{\mathrm{z}}) - L(x') - \lambda^\star\kappa_{\mathrm{z}} \;\geqslant\; \hat{\beta}_{\mathrm{z}} \cdot \frac{\mathrm{d}}{\mathrm{d}e}L(e)\,\big|_{e=x'} - \lambda^\star\kappa_{\mathrm{z}} \\ &=\; d'\hat{\beta}_{\mathrm{z}} - \kappa_{\mathrm{z}}\mathrm{lip}(L) \cdot |\hat{\beta}_{\mathrm{x}}| > 0, \end{aligned}$$

where the first identity uses the definitions of $l_{\hat{\boldsymbol{\beta}}}$ and $\hat{x}^1$, the first inequality exploits the convexity

of L, the second identity uses the definition of d', and the second inequality follows from our earlier assumption about $\hat{\beta}_{\mathrm{z}}$.

To construct the point $\check{x}^1$ at which the first term inside the maximum in the definition of h is strictly negative, we consider the case where $d' > 0$; the alternative case where $d' < 0$ follows from analogous arguments. When $d' > 0$, our earlier assumption $d'\hat{\beta}_{\mathrm{z}} > \kappa_{\mathrm{z}}\mathrm{lip}(L) \cdot |\hat{\beta}_{\mathrm{x}}|$ implies that $\hat{\beta}_{\mathrm{z}} > 0$. Since L is non-negative and convex, we have that

$$\lim_{x \to -\infty} \frac{\mathrm{d}}{\mathrm{d}e} L(e) \,\big|_{e=x} \leqslant 0,$$

which in turn implies that

$$\lim_{x \to -\infty} L(x + \hat{\beta}_{\mathrm{z}}) - L(x) \;\leqslant\; \lim_{x \to -\infty} \hat{\beta}_{\mathrm{z}} \cdot \frac{\mathrm{d}}{\mathrm{d}e} L(e) \,\big|_{e=x+\hat{\beta}_{\mathrm{z}}} \;\leqslant\; 0, \tag{A-2}$$

where the first inequality exploits the fact that the derivative of a convex function is non-decreasing and the second inequality holds since $\hat{\beta}_{\mathrm{z}} > 0$. We thus conclude that

$$\lim_{\hat{\beta}_{\mathrm{x}}\check{x}^1 \to -\infty} l_{\hat{\boldsymbol{\beta}}}(\check{x}^1, 1, 1) - l_{\hat{\boldsymbol{\beta}}}(\check{x}^1, 0, 1) - \lambda^\star \kappa_{\mathrm{z}} \;=\; \lim_{x'' \to -\infty} L(x'' + \hat{\beta}_{\mathrm{z}}) - L(x'') - \lambda^\star \kappa_{\mathrm{z}} \;\leqslant\; -\lambda^\star \kappa_{\mathrm{z}} \;<\; 0,$$

where the first identity applies the change of variables $\hat{\beta}_0 + \hat{\beta}_{\mathrm{x}}\check{x}^1 = x''$ for classification problems and $\hat{\beta}_0 + \hat{\beta}_{\mathrm{x}}\check{x}^1 - 1 = x''$ for regression problems, respectively, the first inequality uses (A-2), and the last inequality is due to the fact that $\lambda^\star, \kappa_{\mathrm{z}} > 0$. □

The proof of Proposition 1 utilizes the following lemma, which we state and prove first.

Lemma 1. *Assume that the loss function L is convex and Lipschitz continuous. For fixed*

$\boldsymbol{\alpha}, \boldsymbol{x}^n \in \mathbb{R}^{M_{\mathrm{x}}}$, $\alpha_0 \in \mathbb{R}$ *and* $\lambda \in \mathbb{R}_+$, *we have*

$$\sup_{\boldsymbol{x} \in \mathbb{R}^{M_{\mathrm{x}}}} L(\boldsymbol{\alpha}^\top \boldsymbol{x} + \alpha_0) - \lambda \|\boldsymbol{x} - \boldsymbol{x}^n\| = \begin{cases} L(\boldsymbol{\alpha}^\top \boldsymbol{x}^n + \alpha_0) & \textit{if } \mathrm{lip}(L) \cdot \|\boldsymbol{\alpha}\|_* \leqslant \lambda, \\ +\infty & \textit{otherwise.} \end{cases} \tag{A-3}$$

Lemma 1 generalizes Lemma 47 of Shafieezadeh-Abadeh et al. (2019) in that it includes a constant α_0 in the argument of the loss function L and that it extends to the case where $\lambda = 0$.

Proof of Lemma 1. Consider first the case where $\boldsymbol{\alpha} \neq \mathbf{0}$. We conduct the change of variables $\boldsymbol{w} = \boldsymbol{x} + \boldsymbol{d}$, where $\boldsymbol{d} \in \mathbb{R}^{M_{\mathrm{x}}}$ is any vector such that $\boldsymbol{\alpha}^\top \boldsymbol{d} = \alpha_0$. Note that $\boldsymbol{d}$ is guaranteed to exist since $\boldsymbol{\alpha} \neq \mathbf{0}$. Setting $\boldsymbol{w}^n = \boldsymbol{x}^n + \boldsymbol{d}$, the left-hand side of (A-3) can then be written as

$$\begin{aligned} \sup_{\boldsymbol{x} \in \mathbb{R}^{M_{\mathrm{x}}}} L(\boldsymbol{\alpha}^\top \boldsymbol{x} + \alpha_0) - \lambda \|\boldsymbol{x} - \boldsymbol{x}^n\| &= \sup_{\boldsymbol{w} \in \mathbb{R}^{M_{\mathrm{x}}}} L(\boldsymbol{\alpha}^\top \boldsymbol{w}) - \lambda \|\boldsymbol{w} - \boldsymbol{d} - (\boldsymbol{w}^n - \boldsymbol{d})\| \\ &= \sup_{\boldsymbol{w} \in \mathbb{R}^{M_{\mathrm{x}}}} L(\boldsymbol{\alpha}^\top \boldsymbol{w}) - \lambda \|\boldsymbol{w} - \boldsymbol{w}^n\| \end{aligned}$$

If $\lambda > 0$, then Lemma 47 of Shafieezadeh-Abadeh et al. (2019) can be directly applied:

$$\begin{aligned} \sup_{\boldsymbol{w} \in \mathbb{R}^{M_{\mathrm{x}}}} L(\boldsymbol{\alpha}^\top \boldsymbol{w}) - \lambda \|\boldsymbol{w} - \boldsymbol{w}^n\| &= \begin{cases} L(\boldsymbol{\alpha}^\top \boldsymbol{w}^n) & \text{if } \mathrm{lip}(L) \cdot \|\boldsymbol{\alpha}\|_* \leqslant \lambda, \\ +\infty & \text{otherwise,} \end{cases} \\ &= \begin{cases} L(\boldsymbol{\alpha}^\top \boldsymbol{x}^n + \alpha_0) & \text{if } \mathrm{lip}(L) \cdot \|\boldsymbol{\alpha}\|_* \leqslant \lambda, \\ +\infty & \text{otherwise.} \end{cases} \end{aligned}$$

Note that Lemma 47 of Shafieezadeh-Abadeh et al. (2019) assumes $\lambda > 0$. For the case where $\lambda = 0$, the left-hand side of (A-3) evaluates to $+\infty$ since L is assumed to be non-constant (*cf.* Section 2.1) and convex. The right-hand side of (A-3) also evaluates to $+\infty$ since $\mathrm{lip}(L) \cdot \|\boldsymbol{\alpha}\|_* > \lambda$ due to L being non-constant and $\boldsymbol{\alpha} \neq \mathbf{0}$. Hence, the equivalence also extends

to the case where $\lambda = 0$.

Now consider the case where $\boldsymbol{\alpha} = \mathbf{0}$. There is no $\boldsymbol{d}$ such that $\boldsymbol{\alpha}^\top \boldsymbol{d} = \alpha_0$ unless $\alpha_0 = 0$. However, when $\boldsymbol{\alpha} = \mathbf{0}$, the left-hand side of (A-3) has the trivial solution $\boldsymbol{x} = \boldsymbol{x}^n$, and the right-hand side of (A-3) simplifies to $L(\alpha_0)$ since $0 \leqslant \lambda$ always holds. Thus, the equivalence still holds, which concludes the proof. □

Proof of Proposition 1. The statement can be proven along the lines of the proof of Theorem 14 (ii) by Shafieezadeh-Abadeh et al. (2019) if we leverage Lemma 1 to re-express the embedded maximization over $\boldsymbol{x} \in \mathbb{X}$. Details are omitted for the sake of brevity. □

Proof of Corollary 1. The proof is similar to those of Corollaries 16 and 17 by Shafieezadeh-Abadeh et al. (2019). Details are omitted for the sake of brevity. □

Proof of Proposition 2. The statement can be proven along the lines of the proof of Theorem 14 (i) by Shafieezadeh-Abadeh et al. (2019). We omit the details for the sake of brevity. □

Proof of Corollary 2. The proof is similar to that of Corollary 15 by Shafieezadeh-Abadeh et al. (2019). Details are omitted for the sake of brevity. □

The proof of Proposition 3 relies on two lemmas that we will state and prove first.

Lemma 2. *The compound norm* $\|[\boldsymbol{\alpha}, \nu]\|_{\text{comp}} = \|\boldsymbol{\alpha}\| + \kappa|\nu|$, $\boldsymbol{\alpha} \in \mathbb{R}^{M_\text{x}}$, $\nu \in \mathbb{R}$ *and* $\kappa > 0$, *satisfies*

$$\|[\boldsymbol{\alpha}, \nu]\|_{\text{comp}*} = \max\left\{ \|\boldsymbol{\alpha}\|_*,\ \frac{|\nu|}{\kappa} \right\}.$$

Proof of Lemma 2. By definition of the dual norm, we have that

$$\|[\boldsymbol{\alpha}, \nu]\|_{\text{comp}*} = \left\{ \begin{array}{ll} \underset{(\boldsymbol{x},y)\in\mathbb{R}^{M_\text{x}}\times\mathbb{R}}{\text{maximize}} & \boldsymbol{\alpha}^\top \boldsymbol{x} + \nu y \\ \\ \text{subject to} & \|\boldsymbol{x}\| + \kappa|y| \leqslant 1. \end{array} \right.$$

Note that the optimization problem on the right-hand side satisfies Slater's condition since the feasible region includes the interior point $(\boldsymbol{x}, y) = (\mathbf{0}, 0)$. The optimization problem thus has a strong dual. To derive the dual problem, we consider the Lagrange dual function

$$g(\gamma) = \sup_{(\boldsymbol{x},y)\in\mathbb{R}^{M_{\mathrm{x}}}\times\mathbb{R}} \boldsymbol{\alpha}^\top \boldsymbol{x} + \nu y - \gamma(\|\boldsymbol{x}\| + \kappa|y| - 1).$$

Note that the maximization is separable over $\boldsymbol{x}$ and y. Focusing on the variable y, in order for the Lagrange dual function to attain a finite value, we need to have $|\nu| \leqslant \gamma\kappa$ so that $y^\star = 0$. Under this condition, the Lagrange dual function simplifies to

$$g(\gamma) = \sup_{\boldsymbol{x}\in\mathbb{R}^{M_{\mathrm{x}}}} \boldsymbol{\alpha}^\top \boldsymbol{x} - \gamma\|\boldsymbol{x}\| + \gamma.$$

We can now apply Lemma 1 with L being the identity to obtain the equivalent reformulation

$$g(\gamma) = \begin{cases} \gamma & \text{if } \|\boldsymbol{\alpha}\|_* \leqslant \gamma \ , \ |\nu| \leqslant \gamma\kappa, \\ +\infty & \text{otherwise.} \end{cases}$$

The dual problem therefore is

$$\begin{aligned} \underset{\gamma}{\text{minimize}} \quad & \gamma \\ \text{subject to} \quad & |\nu| \leqslant \gamma\kappa \\ & \|\boldsymbol{\alpha}\|_* \leqslant \gamma \\ & \gamma \in \mathbb{R}, \end{aligned}$$

which has the optimal solution $\|[\boldsymbol{\alpha}, \nu]\|_{\mathrm{comp}*} = \gamma^\star = \max\{\|\boldsymbol{\alpha}\|_*, |\nu|/\kappa\}$. □

Lemma 3. *Assume that the loss function L is convex and Lipschitz continuous. For fixed*

$\boldsymbol{\alpha}, \boldsymbol{x}^n \in \mathbb{R}^{M_\mathrm{x}}$, $\alpha_0, y^n \in \mathbb{R}$, $\kappa > 0$ *and* $\lambda \in \mathbb{R}_+$, *we have*

$$\sup_{(\boldsymbol{x},y)\in\mathbb{R}^{M_\mathrm{x}}\times\mathbb{R}} L(\boldsymbol{\alpha}^\top \boldsymbol{x} - y + \alpha_0) - \lambda\|\boldsymbol{x} - \boldsymbol{x}^n\| - \lambda\kappa|y - y^n| = \begin{cases} L(\boldsymbol{\alpha}^\top \boldsymbol{x}^n - y^n + \alpha_0) & \textit{if } \mathrm{lip}(L) \cdot \max\{\kappa\|\boldsymbol{\alpha}\|_*, 1\} \leqslant \lambda\kappa, \\ +\infty & \textit{otherwise.} \end{cases}$$

Proof of Lemma 3. Concatenating the variables $\boldsymbol{x}$ and y to $\boldsymbol{w} = [\boldsymbol{x}^\top, y]^\top \in \mathbb{R}^{M_\mathrm{x}+1}$, letting $\boldsymbol{\eta} = [\boldsymbol{\alpha}^\top, -1]^\top$ and $\boldsymbol{w}^n = [(\boldsymbol{x}^n)^\top, y^n]^\top$ and defining the compound norm $\|[\boldsymbol{x}, y]\|_\mathrm{comp} = \|\boldsymbol{x}\| + \kappa|y|$, we can write the left-hand side of the equation in the statement of the lemma as

$$\sup_{\boldsymbol{w}\in\mathbb{R}^{M_\mathrm{x}+1}} L(\boldsymbol{\eta}^\top \boldsymbol{w} + \alpha_0) - \lambda\|\boldsymbol{w} - \boldsymbol{w}^n\|_\mathrm{comp}.$$

Now we can apply Lemma 1 directly to conclude that

$$\sup_{\boldsymbol{w}\in\mathbb{R}^{M_\mathrm{x}+1}} L(\boldsymbol{\eta}^\top \boldsymbol{w} + \alpha_0) - \lambda\|\boldsymbol{w} - \boldsymbol{w}^n\|_\mathrm{comp} = \begin{cases} L(\boldsymbol{\eta}^\top \boldsymbol{w}^n + \alpha_0) & \text{if } \mathrm{lip}(L) \cdot \|\boldsymbol{\eta}\|_{\mathrm{comp}*} \leqslant \lambda, \\ +\infty & \text{otherwise.} \end{cases}$$

The statement now follows from Lemma 2, which implies that $\mathrm{lip}(L) \cdot \|\boldsymbol{\eta}\|_{\mathrm{comp}*} \leqslant \lambda$ if and only if $\mathrm{lip}(L) \cdot \max\{\|\boldsymbol{\alpha}\|_*, |-1|/\kappa\} \leqslant \lambda$. □

Proof of Proposition 3. The statement can be proven along the lines of the proof of Theorem 4 (ii) by Shafieezadeh-Abadeh et al. (2019) if we leverage Lemma 3 to re-express the embedded maximization over $(\boldsymbol{x}, y) \in \mathbb{R}^{M_\mathrm{x}} \times \mathbb{R}$. Details are omitted for the sake of brevity. □

Proof of Corollary 3. The proof is similar to that of Corollary 5 by Shafieezadeh-Abadeh et al. (2019). Details are omitted for the sake of brevity. □

Proof of Proposition 4. The statement can be proven along the lines of the proof of Theo-

rem 4 (i) by Shafieezadeh-Abadeh et al. (2019). We omit the details for the sake of brevity. □

Proof of Corollary 4. The proof is similar to those of Corollaries 6 and 7 by Shafieezadeh-Abadeh et al. (2019). Details are omitted for the sake of brevity. □

Proof of Proposition 5. We first show that LB and UB constitute lower and upper bounds on the optimal value of (8) throughout the execution of the algorithm, and then we conclude that Algorithm 1 terminates in finite time with an optimal solution $(\boldsymbol{\theta}^\star, \boldsymbol{\sigma}^\star)$ to problem (8).

Algorithm 1 updates LB to $f_0(\boldsymbol{\theta}^\star, \boldsymbol{\sigma}^\star)$ in each iteration of the while-loop. Since $(\boldsymbol{\theta}^\star, \boldsymbol{\sigma}^\star)$ is an optimal solution to the relaxation of problem (8) that only contains the constraints $\mathcal{W} \subseteq [N] \times \mathcal{I} \times \mathbb{Z}$, LB indeed constitutes a lower bound on the optimal value of (8). Moreover, since no elements are ever removed from $\mathcal{W}$, the sequence of lower bounds LB is monotonic.

To see that UB constitutes an upper bound on the optimal value of (8) throughout the execution of Algorithm 1, we claim that in each iteration, $(\boldsymbol{\theta}^\star, \boldsymbol{\sigma}^\star + \boldsymbol{\vartheta}^\star)$ constitutes a feasible solution to problem (8). Indeed, we have $(\boldsymbol{\theta}^\star, \boldsymbol{\sigma}^\star) \in \Theta \times \mathbb{R}_+^N$ and $\boldsymbol{\vartheta}^\star \in \mathbb{R}_+^N$ by construction, while for all $n \in [N]$, $i \in \mathcal{I}$ and $\boldsymbol{z} \in \mathbb{Z}$, we have that

$$f_{ni}(g_{ni}(\boldsymbol{\theta}^\star, \boldsymbol{\xi}^n_{-\mathbf{z}}; \boldsymbol{z})) - h_{ni}(\boldsymbol{\theta}^\star, \boldsymbol{\xi}^n_{-\mathbf{z}}; d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n)) \;\leqslant\; \sigma^\star_n + \vartheta(n,i) \;\leqslant\; \sigma^\star_n + \vartheta(n,i(n)) \;\leqslant\; \sigma^\star_n + \vartheta^\star_n,$$

that is, $(\boldsymbol{\theta}^\star, \boldsymbol{\sigma}^\star + \boldsymbol{\vartheta}^\star)$ is indeed feasible in problem (8). Moreover, the sequence of upper bounds UB is monotonic by construction of the updates.

To see that Algorithm 1 terminates in finite time, finally, note that each iteration of the while-loop either adds a new constraint index $(n, i(n), \boldsymbol{z}(n, i(n)))$ to $\mathcal{W}$, or we have $\vartheta(n, i(n)) \leqslant 0$ for all $n \in [N]$. In the latter case, however, we have $\boldsymbol{\vartheta}^\star = \mathbf{0}$ and thus LB = UB at the end of the iteration. The claim now follows from the fact that the index set $[N] \times \mathcal{I} \times \mathbb{Z}$ is finite. □

Proof of Theorem 3. For fixed $(\boldsymbol{\theta}^\star, \boldsymbol{\sigma}^\star)$, finding the most violated constraint index $\boldsymbol{z}(n,i)$ in

constraint group (n, i) amounts to solving the combinatorial optimization problem

$$\begin{aligned} \underset{\boldsymbol{z}}{\text{maximize}} \quad & f_{ni}(g_{ni}(\boldsymbol{\theta}^\star, \boldsymbol{\xi}^n_{-\mathbf{z}}; \boldsymbol{z})) - h_{ni}(\boldsymbol{\theta}^\star, \boldsymbol{\xi}^n_{-\mathbf{z}}; d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n)) - \sigma^\star_n \\ \text{subject to} \quad & \boldsymbol{z} \in \mathbb{Z}. \end{aligned}$$

We can solve this problem by solving the $K + 1$ problems

$$\left[\begin{array}{ll} \underset{\boldsymbol{z}}{\text{maximize}} & f_{ni}(g_{ni}(\boldsymbol{\theta}^\star, \boldsymbol{\xi}^n_{-\mathbf{z}}; \boldsymbol{z})) - h_{ni}(\boldsymbol{\theta}^\star, \boldsymbol{\xi}^n_{-\mathbf{z}}; \delta) - \sigma^\star_n \\ \text{subject to} & d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n) = \delta \\ & \boldsymbol{z} \in \mathbb{Z} \end{array}\right] \quad \forall \delta \in [K] \cup \{0\},$$

where each problem conditions on a fixed number δ of discrepancies between $\boldsymbol{z}$ and $\boldsymbol{z}^n$, and subsequently choosing any solution $\boldsymbol{z}(\delta)$ that attains the maximum optimal objective value among those $K + 1$ problems. Removing constant terms from those $K + 1$ problems, we observe that the δ-th problem shares its set of optimal solutions with the problem

$$\begin{aligned} \underset{\boldsymbol{z}}{\text{maximize}} \quad & f_{ni}(g_{ni}(\boldsymbol{\theta}^\star, \boldsymbol{\xi}^n_{-\mathbf{z}}; \boldsymbol{z})) \\ \text{subject to} \quad & d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n) = \delta \\ & \boldsymbol{z} \in \mathbb{Z}. \end{aligned}$$

Since the outer function f_{ni} in the objective function is convex, the objective function is maximized whenever the inner function g_{ni} in the objective function is either maximized or minimized. We thus conclude that the δ-th problem is solved by solving the two problems

$$\left[\begin{array}{ll} \underset{\boldsymbol{z}}{\text{maximize}} & \mu \cdot g_{ni}(\boldsymbol{\theta}^\star, \boldsymbol{\xi}^n_{-\mathbf{z}}; \boldsymbol{z}) \\ \text{subject to} & d_{\mathrm{z}}(\boldsymbol{z}, \boldsymbol{z}^n) = \delta \\ & \boldsymbol{z} \in \mathbb{Z} \end{array}\right] \quad \forall \mu \in \{\pm 1\}$$

and subsequently choosing the solution $\boldsymbol{z}(\mu, \delta)$ that attains the larger value $f_{ni}(g_{ni}(\boldsymbol{\theta}^\star, \boldsymbol{\xi}^n_{-\mathbf{z}}; \boldsymbol{z}(\mu, \delta))$ among those two solutions (with ties broken arbitrarily). Fixing μ to either value, adopting the notation for $\boldsymbol{w}$ and w_0 of Algorithm 2 and ignoring constant terms, we can write the problem as

$$\begin{aligned} \underset{\boldsymbol{z}}{\text{maximize}} \quad & \mu \cdot \Big[\sum_{m \in [K]} \boldsymbol{w}_m{}^\top \boldsymbol{z}_m \Big] \\ \text{subject to} \quad & d_{\text{z}}(\boldsymbol{z}, \boldsymbol{z}^n) = \delta \\ & \boldsymbol{z} \in \mathbb{Z}. \end{aligned}$$

The rectangularity of $\mathbb{Z}$ implies that the decisions of this problem admit a decomposition into the selection $\mathcal{M} \subseteq [K]$ of δ categorical features $m \in [K]$ along which $\boldsymbol{z}_m$ differs from $\boldsymbol{z}^n_m$ and, for those features $m \in [K]$ where $\boldsymbol{z}_m \neq \boldsymbol{z}^n_m$, the choice of $\boldsymbol{z}_m \in \mathbb{Z}(k_m) \backslash \{\boldsymbol{z}^n_m\}$:

$$\begin{aligned} \underset{\mathcal{M}}{\text{maximize}} \quad & \max_{\boldsymbol{z}} \left\{ \mu \cdot \Big[\sum_{m \in [K]} \boldsymbol{w}_m{}^\top \boldsymbol{z}_m \Big] \,:\, \left[\begin{array}{ll} \boldsymbol{z}_m \in \mathbb{Z}(k_m) \backslash \{\boldsymbol{z}^n_m\} & \forall m \in \mathcal{M} \\ \boldsymbol{z}_m = \boldsymbol{z}^n_m & \forall m \in [K] \backslash \mathcal{M} \end{array} \right] \right\} \\ \text{subject to} \quad & \mathcal{M} \subseteq [K], \;\; |\mathcal{M}| = \delta. \end{aligned}$$

Noticing that the embedded maximization problem decomposes along the categorical features $m \in [K]$, we can adopt the notation for $\boldsymbol{z}^\star_m$ of Algorithm 2 to obtain the equivalent formulation

$$\begin{aligned} \underset{\mathcal{M}}{\text{maximize}} \quad & \Big[\sum_{m \in \mathcal{M}} \mu \cdot \boldsymbol{w}_m{}^\top \boldsymbol{z}^\star_m \Big] + \Big[\sum_{m \in [K] \backslash \mathcal{M}} \mu \cdot \boldsymbol{w}_m{}^\top \boldsymbol{z}^n_m \Big] \\ \text{subject to} \quad & \mathcal{M} \subseteq [K], \;\; |\mathcal{M}| = \delta. \end{aligned}$$

The two summations in the objective function of this problem admit the reformulation

$$\Big[\sum_{m \in \mathcal{M}} \mu \cdot \boldsymbol{w}_m{}^\top \boldsymbol{z}^\star_m \Big] + \Big[\sum_{m \in [K] \backslash \mathcal{M}} \mu \cdot \boldsymbol{w}_m{}^\top \boldsymbol{z}^n_m \Big] \;=\; \Big[\sum_{m \in [K]} \mu \cdot \boldsymbol{w}_m{}^\top \boldsymbol{z}^n_m \Big] + \Big[\sum_{m \in \mathcal{M}} \mu \cdot \boldsymbol{w}_m{}^\top (\boldsymbol{z}^\star_m - \boldsymbol{z}^n_m) \Big],$$

and ignoring constant terms once more simplifies our optimization problem to

$$\begin{aligned} \underset{\mathcal{M}}{\text{maximize}} \quad & \sum_{m \in \mathcal{M}} \mu \cdot \boldsymbol{w}_m{}^\top (\boldsymbol{z}_m^\star - \boldsymbol{z}_m^n) \\ \text{subject to} \quad & \mathcal{M} \subseteq [K], \;\; |\mathcal{M}| = \delta. \end{aligned}$$

This problem is solved by identifying $\mathcal{M}$ with the indices of the δ largest elements of the sequence $\mu \cdot \boldsymbol{w}_1{}^\top (\boldsymbol{z}_1^\star - \boldsymbol{z}_1^n), \ldots, \mu \cdot \boldsymbol{w}_K{}^\top (\boldsymbol{z}_K^\star - \boldsymbol{z}_K^n)$, and this problem can be solved by a simple sorting algorithm. One readily verifies that Algorithm 2 adopts the solution approach just described to determine a maximally violated constraint index $\boldsymbol{z}(n, i)$.

The runtime of Algorithm 2, finally, is dominated by determining the $2K$ maximizers $\boldsymbol{z}_m^\star$, $m \in [K]$ and $\mu \in \{\pm 1\}$, which takes time $\mathcal{O}(M_{\mathbf{z}})$ due to the one-hot encoding employed by $\mathbb{Z}$, sorting the $2K$ values $\mu \cdot \boldsymbol{w}_m{}^\top (\boldsymbol{z}_m^\star - \boldsymbol{z}_m^n)$, which takes time $\mathcal{O}(K \log K)$, as well as determining a maximally violated constraint among the $2K + 2$ candidates $\boldsymbol{z} \in \mathcal{Z}$, which takes time $\mathcal{O}(KT)$. $\square$