Learning to Solve Optimization Problems with Hidden Components: Applications in Automated Treatment Planning

by

Rafid Mahmood

A thesis submitted in conformity with the requirements
for the degree of Doctor of Philosophy
Graduate Department of Mechanical and Industrial Engineering
University of Toronto

Abstract

Learning to Solve Optimization Problems with Hidden Components: Applications in Automated Treatment Planning

Rafid Mahmood
Doctor of Philosophy
Graduate Department of Mechanical and Industrial Engineering
University of Toronto
2020

Designing optimization models that capture decision-maker preferences typically requires guidance from domain experts. We can instead employ machine learning (ML) to design contextual optimization models using data sets of past decisions. In this thesis, we propose techniques to formulate and solve problems where the objective or the feasible set is dependent on decision-maker preferences. We apply these methods to automate the design of personalized radiation therapy treatments for head-and-neck cancer patients. Here, the prevailing framework is Knowledge-based planning (KBP), which is a two-stage pipeline that involves generating an expected dose and optimizing a treatment to deliver the generated dose.

We first propose an ensemble learning framework for Inverse Linear Optimization (ILO), which is a structured prediction problem for estimating the cost vector of a linear program from observed decisions. Our framework specializes to existing variants in the literature, admits new solution algorithms, and includes a statistical goodness-of-fit metric. We employ our framework to develop the first ensemble KBP pipeline that incorporates multiple different dose generation models to yield better treatments than existing single-dose pipelines.

Next, we develop a deep learning-based dose generation model that uses a generative adversarial network to map from CT images to dose distributions. Previous dose generation models employed classical ML to estimate dose summary statistics. Our approach outperforms classical models on clinical metrics.

We then explore contextual optimization when the feasible set varies as a function of features. We present Interior Point Methods with Adversarial Networks (IPMAN), an algorithm for learning the feasible set and predicting corresponding optimal decisions for contextual problems. We prove our approach yields optimality guarantees and generalization bounds. We then re-cast dose generation as an optimization problem and implement IPMAN to predict optimal doses. Our predictions achieve clinical metrics better than baselines and also demonstrate a transfer learning to new clinics that use different metrics.

Finally, motivated by data augmentation for IPMAN, we consider the task of sampling infeasible decisions from an optimization problem. We present a Markov Chain Monte Carlo algorithm for sampling from the complement of a polyhedron that provably covers the complement and demonstrate its effectiveness in numerical experiments.

Dedicated to my grandfathers, Abu Md. M. Islam and Md. Jamsheed Alam.

Acknowledgements

First and foremost, I thank God.

My deepest gratitude goes to my adviser Timothy C. Y. Chan. He continually surprises me with his extraordinary awareness of his students' needs. His academic curiosity is vast; he has encouraged every direction that I wanted to pursue and helped turn some of my most half-baked ideas into research contributions. His approach to goal setting and optimizing life His philosophy towards life in terms of planning and meeting goals (whether learning, research, or teaching) while simultaneously balancing a home life sets a standard that I hope to achieve. Thank you for being an exceptional mentor.

Throughout my PhD, I have had the opportunity to learn from and work with many wonderful people. Thank you to Taewoo Lee and Daria Terekhov for your guidance, patience, and support as I began my journey into operations research. Thank you to Adam Diamant for being always available as a collaborator, mentor, and friend; your optimism is unparalleled and your encouragement is gratefully appreciated. Thanks especially to Aaron Babier who introduced me to radiation therapy and has been a friend and collaborator over so many projects. The month that we created GANCER is the highlight of my graduate school experience. Finally, my thanks goes to Chi-Guhn Lee and Vahid Sarhangian for their guidance and feedback as committee members.

When I joined the Applied Optimization Lab, I was astonished by the positive and supportive attitudes of the members. I realize that this is a fostered culture that every member in their time has helped to enrich. I have learned much from all of you and I hope that I have given back in turn. Thank you to Derya Dermitaş, Philip Mar, Justin Boutilier, Chris Sun, Neal Kaw, Iman Dayarian, Islay Wright, Ian Zhu, Minha Lee, Ben Potter, Ben Leung, Jonathan Ranisau, Clara Stoesser, Bing Zhang, Nasrin Yousefi, Matt Crowson, Yusuf Shalaby, Frances Pogacar, Simon Huang, Imran Saleh, Craig Fernandes, Rachel Wong, and Albert Loa.

I am grateful for having a tight circle of longstanding friends who regularly remind me of important things such as life outside of the academy. I always look forward to our annual winter trips where we compress a year's worth of company into one week. Thanks especially to Nadeesha Amarasinghe for providing excellent conversations over near-death experiences while hiking around the world, and to Hassan Farooq for introducing me to a new obsession every summer. The three of us participating in the Quantathon was the most formative experience during my PhD. We taught ourselves machine learning and data-driven optimization, but most importantly, we found out that we could '*just do it.*'

Finally, I must thank my family. Thank you to my uncles and aunts for all of the little things throughout the years. Thank you to my grandmother for mediating arguments

and providing so much unhealthy yet incredibly delicious food. Thank you to my cousins Ishtiaque Dhrubo and Raima Lohani for therapy sessions and road trips. To my parents, Iftekhar Mahmood and Sulata Islam, thank you for all of the opportunities that have shaped me as a person and for all of the support and guidance as I pursued higher studies. Finally, I must thank my twin sources of scholarly inspiration, my grandfathers, Abu Md. M. Islam and Md. Jamsheed Alam. I hope that I continue to further your legacies and it is to your memories that I dedicate this thesis.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Constrained optimization is a flexible modeling language that is used to solve operational problems in business (Fabozzi and Valente, 1976), economics (Bernhard, 1969), engineering (Luo and Yu, 2006), finance (Cornuejols and Tütüncü, 2006), healthcare (Stinnett and Paltiel, 1996), logistics (Mula et al., 2010), and so forth. However, constrained optimization models must be carefully designed in practice. If the model's objectives and constraints do not perfectly reflect the domain-specific problem, then the decisions may ultimately not be useful.

A decision-maker and an optimization expert may work together to construct a model. The decision-maker's understanding of the problem is motivated by domain expertise, which is often intuitive and may not be amenable to concise mathematical expressions. For example, a clinician selecting a combination of treatments for a patient may understand interaction effects from anecdotal experience with previous patients. In other cases, the optimization model may be contextual depending on auxiliary features. Given a specific patient's history, the clinician may require a custom prescriptive model.

The availability of large sets of decision data presents opportunities for designing optimization models without manually incorporating a decision-maker's contextual intuition. Machine learning techniques can leverage past decisions to infer the behavior preferences of decision-makers, which in turn, can automate the design of future optimization models. For example, we can learn from previously delivered medicines that patients with certain characteristics require aggressive treatments and incorporate this information into the optimization model.

This thesis develops algorithms for learning to formulate and solve constrained optimization problems. We propose machine learning, optimization, and sampling techniques that use historical decisions to estimate the objective function and feasible set of optimization models. To demonstrate their efficacy, we apply these methods towards aspects

of a clinical problem of designing personalized medicine for patients with head-and-neck cancer. We implement several novel technologies for the automated planning of intensity modulated radiation therapy (RT) treatments that leverage past clinical treatments to learn personalized objectives and constraints that meet the approval criteria of clinical domain experts. Our automated treatment design technologies thus can reduce the current overhead required from clinicians to direct and approve the design of custom treatments for each patient.

1.1 Motivating application: Intensity-modulated radiation therapy

The concrete motivating application in this thesis is the automated generation of radiation therapy treatment plans in oropharyngeal (head-and-neck) cancer. Intensity-modulated radiation therapy (RT) is one of the most widely-used cancer treatment techniques and is recommended for over 50% of all cancer cases (Delaney et al., 2005). In RT, a linear accelerator (LINAC) outputs high-energy x-ray beams from multiple angles around a patient to deliver a prescribed dose of radiation to a tumor while minimizing dose to the healthy tissue. An RT treatment plan is the result of a complex design process involving medical professionals and several software systems. This includes specialized optimization software that determines the beam characteristics (e.g., aperture shapes for each beam angle, dose delivered from each aperture) required to deliver the final dose distribution. The optimization model takes as input a set of computed tomography (CT) images of the patient, various dosimetric objectives and constraints, and other parameters that guide the optimization process. The model outputs a treatment plan that is subsequently evaluated by an oncologist. The oncologist usually proposes modifications to the plan, which then requires the treatment planner to re-solve the optimization model using updated parameters. Figure 1.1 summarizes the total process, which is labor intensive, time-consuming, and costly, as the back-and-forth between the planner and oncologist is often repeated multiple times until the plan is approved. The design of a single treatment plan may take several days (Das et al., 2009). Combined with growing patient volumes in cancer clinics worldwide, this leads to strain in the operation of a cancer center and significant delays in treatment for patients (Atun et al., 2015).

The significant manual effort associated with the current treatment planning paradigm, along with the fact that RT plans are generally quite similar for patients with similar geometries, has motivated researchers to investigate how automation can be used in

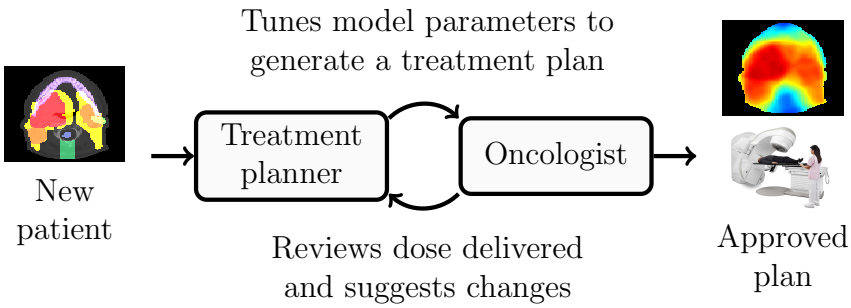


Figure 1.1: An overview of the clinical treatment design process. A treatment planner and oncologist iterate over model parameters until an approvable plan is constructed.

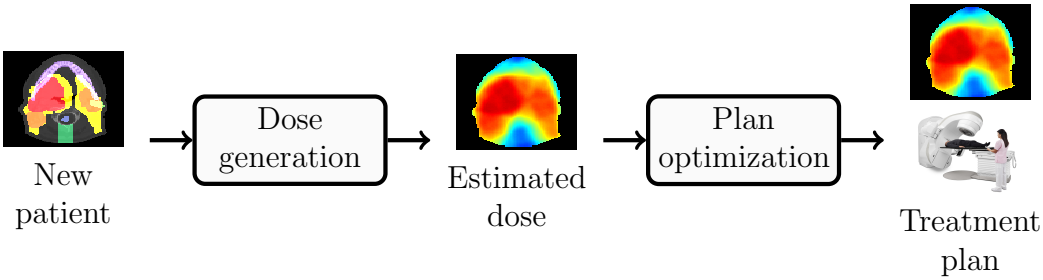


Figure 1.2: An overview of the clinical treatment design process. A treatment planner and oncologist iterate over model parameters until an approvable plan is constructed.

the planning process (Sharpe et al., 2014). A key enabler of automation is known as knowledge-based planning (KBP), which leverages historically delivered treatments to generate new plans for similar patients. Figure 1.2 depicts the two main components of a KBP-driven automated planning system:

- **Dose generation:** a machine learning model that uses CT-derived patient geometric features to predict a clinically acceptable three-dimensional dose distribution (Appenzoller et al., 2012; Kearney et al., 2018; McIntosh et al., 2017; Shiraishi et al., 2015; Yang et al., 2013; Younge et al., 2018). The dose distribution is a map of how much radiation each cubic volume of a patient is expected to receive.

- **Plan optimization:** an optimization model that converts the prediction into a "deliverable" plan (Babier et al., 2018a; McIntosh and Purdie, 2017; Petersson et al., 2016; Wu et al., 2017). This step ensures the treatment plan produced by the machine learning model satisfies the physical delivery constraints imposed by the LINAC.

Existing KBP frameworks can be improved in several directions. First, treatment

plans generated by automation are evaluated on a set of competing clinical criteria. While real clinical plans reflect an oncologist-driven balance of these metrics, specific KBP pipelines typically lead to plans that over-fit to specific subsets of the criteria. A second drawback of many dose generation methods is in their reliance on low-dimensional hand-tailored features that limit their effectiveness in performance. Finally, the protocols for radiation therapy often vary between clinics (e.g., Geretschläger et al., 2015 versus Babier et al., 2018b). It is not possible to deploy the same automated planning pipeline at multiple institutions because current prediction models do not factor the prescriptive nature of the prediction. Models trained using data from one clinic may not satisfy protocols (e.g., hidden constraints) at other institutions (Wu et al., 2017).

1.2 Contributions and outline

We summarize our contributions and the structure of this thesis below. Each chapter is self-contained but draws on the KBP background summarized in Chapter 2. Chapters 3 and 5 introduce methods for learning the objective and the feasible set of an optimization problem, respectively. Chapter 4 presents a numerical study in using deep learning for treatment planning; this work inspires the general methodology in Chapter 5. Finally, Chapter 6 presents a data augmentation procedure used to learn the feasible set of an optimization problem.

Ensemble inverse linear optimization

In Chapter 3, we explore the problem of estimating the cost vector of a linear optimization problem $\min\{\mathbf{c}^\intercal\mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{b}\}$ from a data set of multiple observed decisions. This is motivated by the setting of learning a consensus objective from a group of decision-makers each considering different solutions to an optimization problem. We develop a general inverse linear optimization framework that unifies prior techniques for which we derive assumption-free, exact solution algorithms. We apply our framework to develop a novel KBP plan optimization technique that uses an ensemble of different dose generation models to design a treatment plan reflecting a consensus from different estimators. While current KBP dose generation models over-fit to a single metric, our ensemble framework achieves better aggregate performance than single-point frameworks.

While there is a growing body of literature in inverse convex optimization (i.e., estimating parameters for a convex optimization problem), such methods often do not take into account linear programming idiosyncrasies that can permit more efficient algo-

rithms. In this chapter, we develop a complete suite of solution algorithms that maps each inverse optimization problem into a polynomial number of linear programs. We additionally introduce several special cases where inverse optimization admits efficient analytic solutions or linear programming algorithms. Finally, our ensemble KBP pipeline is the first in the clinical literature to use multiple predictions and shows performance improvements over traditional automated planning models on aggregate metrics. *This chapter contains work done in collaboration with Aaron Babier, Timothy C. Y. Chan, Taewoo Lee, and Daria Terekhov.*

Dose generation with generative adversarial networks

In Chapter 4, we investigate deep learning dose generation models for KBP and introduce a Generative Adversarial Network (GAN) to estimate dose distributions. The prior dose generation literature consisted of classical machine learning techniques (e.g., linear regression, random forests) that use a set of hand-tailored patient-geometry features to predict low-dimensional representations of the dose. Instead, GANs re-cast dose generation into a computer vision task: given a 3-D CT image, estimate a dose image of the patient. We demonstrate that our deep learning approach outperforms conventional models over the set of clinic-mandated satisfaction criteria and dose similarity metrics.

This chapter contains the first use of GANs to generate radiation therapy treatment plans. We are the first to treat KBP prediction as an image colorization problem (i.e., recolor a contour image to a dose image) for which GANs are known to perform especially well. Furthermore, oropharyngeal cancer is one of the most difficult cancers for designing treatments. Since our site-independent method outperforms classical ML for this cancer type, we also expect similar performance for GANs for simpler sites such as prostate and stomach cancers. *This chapter contains work done in collaboration with Aaron Babier, Timothy C. Y. Chan, Adam Diamant, and Andrea McNiven.*

Learning to optimize with hidden constraints

In Chapter 5, we develop an algorithm to estimate the feasible set and predict optimal decisions for optimization problems dependent on contextual features. We use a data set of historical decisions and features to train two machine learning models: the first classifies whether a decision is feasible, while the second generates candidate decisions. We learn an unstructured representation of the feasible set and ensure optimality guarantees and out-of-sample generalization bounds. We apply this algorithm to construct an *optimal* dose generation model for automated KBP. These doses imitate oncologist preferences by

making the same trade-offs as in real clinical plans. Furthermore, our algorithm adapts to criteria from different clinics, i.e., a form of transfer learning. We use clinical data from one institution to learn the criteria of a different clinic and demonstrate that our algorithm can be easily deployed to new institutions that do not have sufficient data.

This chapter is the first work in the operations and machine learning literature where the feasible set itself must first be learned in order to estimate decisions. Our learning algorithm can be used with any machine learning model. Furthermore, we provide theoretical results that connect the machine learning literature to interior point methods by proving that our algorithm satisfies optimality guarantees and both in-sample and out-of-sample error bounds. Finally, we demonstrate practical implications by showing that it can be used for prescriptive purposes to create treatment plans structurally different from the ground truth, i.e., the data that was originally used to train the learning model. *This chapter contains work done in collaboration with Aaron Babier, Timothy C. Y. Chan, and Adam Diamant.*

Sampling from the complement of a polyhedron

In Chapter 6, we introduce an MCMC algorithm for generating points in the complement of a polyhedron. High-dimensional sampling has historically only considered sampling from the interior and the boundary of a convex set. Our algorithm is based on the classical Shake-and-bake algorithm used to sample from the boundary of a polyhedron. While the complement of a polyhedron is itself a non-convex set, our algorithm is as efficient as the Shake-and-bake and enjoys a relevant property of covering the entire complement.

This topic is motivated by Chapter 5 where we use infeasible decisions to an optimization problem to learn a barrier function for the feasible set. Data-driven optimization typically features only data sets of feasible decisions, but using real feasible and sampled infeasible decisions, we can accurately learn to estimate when a decision is feasible. Our numerical results show that we are often 20% more accurate than unsupervised learning techniques that do not use infeasible data. *This chapter contains work done in collaboration with Timothy C. Y. Chan and Adam Diamant.*

Chapter 2

Background on radiation therapy

This chapter collates the clinical background for our radiation therapy (RT) application, including a summary on treatment planning and clinical criteria, a discussion of recent results in automated Knowledge-based planning (KBP), and details on our clinical data set.

2.1 Treatment planning for oropharyngeal cancer

An RT treatment is delivered by a linear accelerator (LINAC) that delivers high-energy X-rays from different angles to a patient's tumor. The patient's body is discretized into tiny voxels in order to calculate the dose delivered to each voxel. The design of an IMRT treatment plan is done by mathematical optimization where the decision variable is composed of two components that represent the beamlets and the dose delivered (in Gy) as a result of the intensities of the beamlets, respectively.

Oropharyngeal cancer is a challenging form of cancer to design treatments for because this cancer type typically involves multiple tumor locations, referred to as planning target volumes (PTVs). An RT plan may need to account for up to three PTVs that each require different prescription doses (i.e., our institutional data source uses PTV56, PTV63, and PTV70 with 56 Gy, 63 Gy, and 70 Gy as prescription doses, respectively. Furthermore, the head-and-neck area contains several crucial organs that are particularly sensitive to radiation. These structures are referred to as organs-at-risk (OARs) and an an RT plan may need to account for up to eight OARs: brain stem, spinal cord, right parotid, left parotid, larynx, esophagus, mandible, and limPostNeck.

The quality of an RT plan is assessed in terms of several pass-fail clinical criteria. These criteria are specific to a given institution. In this work, we use the default criteria from our institutional data source, which features one clinical criteria per PTV and

Table 2.1: Clinical criteria used to evaluate all plans.

	Structure	Clinical criteria
OARs	Brainstem	Mean $\leq$ 54 Gy
	Spinal Cord	Mean $\leq$ 48 Gy
	Right Parotid	Mean $\leq$ 26 Gy
	Left Parotid	Mean $\leq$ 26 Gy
	Larynx	Mean $\leq$ 45 Gy
	Esophagus	Mean $\leq$ 45 Gy
	Mandible	Max $\leq$ 73.5 Gy
PTVs	PTV70	99%-ile $\geq$ 70 Gy
	PTV63	99%-ile $\geq$ 63 Gy
	PTV56	99%-ile $\geq$ 56 Gy

OAR. Table 2.1 lists the clinical criteria from our institution. Since the limPostNeck is an artificially defined region used solely for optimization, it does not possess a clinical criteria. Note that the clinical criteria for OARs generally consist of upper bounds on the mean or the maximum dose spread over the OAR, but the PTV criteria are are bounds on the 99%-ile or the Value-at-Risk (VaR) to the structure.

2.2 Knowledge-based planning

KBP can automate large parts of the iterative treatment design problem via a two-stage procedure. First, a machine learning model that has been trained on historical treatments, estimates a clinically desirable dose distribution for the patient. Then, an optimization model uses the dose estimate to construct a treatment plan that can deliver a dose of similar quality.

Many different approaches have been tested for the machine learning dose generation component of a KBP-driven automated planning pipeline (see Figure 1.2). Query-based methods identify previously treated patients who are sufficiently similar to the new patient, and use the historically achieved dose metrics as predictions for the new patient (Wu et al., 2009, 2011). Another common approach uses principal component analysis (PCA), in conjunction with linear regression, to predict dose metrics for new patients (Yuan et al., 2012; Zhu et al., 2011). However, these well-established techniques only predict two-dimensional dose metrics. Previous research has shown that 3-D dose distribution predictions can also be generated using random forests (McIntosh et al., 2017; Shiraishi and Moore, 2016). Nevertheless, for these approaches to work effectively, significant effort must be spent in feature engineering, i.e., introducing features specific

to the cancer site. More recently, neural network models have been used to great effect to predict 3-D dose distributions (Babier et al., 2020a; Kearney et al., 2018; Nguyen et al., 2019). Chapter 4 and 5 of this thesis cover two novel methods for predicting dose distributions via neural networks. Such deep learning-based approaches typically do not require feature engineering unlike classical machine learning.

For plan optimization, there are two main approaches for turning predictions into treatments. The first is "dose mimicking", which amounts to minimizing a 2-norm loss on the predicted dose, while enforcing deliverability constraints (Petersson et al., 2016). The second approach uses inverse optimization (see Chapter 3) to learn the parameters of a dosimetrist's optimization model given a predicted dose, followed by solving the forward problem using the learned parameters (Chan et al., 2014). Here, a predicted dose is treated as an "observed decision" (Babier et al., 2020a). A key advantage of inverse optimization is that it better replicates the trade-offs implicit in clinical treatment plans (Chan and Lee, 2018). Chapter 3 introduces a novel inverse optimization approach for KBP that employs multiple prediction models to generate a single treatment.

2.3 Data

In our experiments, we obtain treatment plans from 217 oropharyngeal cancer patients treated at a single institution, Princess Margaret Cancer Centre, with a 6 MV, step-and-shoot intensity-modulated radiation therapy LINAC. All plans are for a prescription of 70 Gy, 63 Gy, and 56 Gy to the gross disease (PTV73), intermediate risk (PTV63), and elective (PTV56) target volumes, respectively.

Our data set includes a 3-D CT image for each patient. Every voxel (a 3-D pixel of size 4 mm $\times$ 4 mm $\times$ 2 mm) of this CT image is classified by clinically drawn contours that denote the structure. All voxels are assigned a structure-specific color, and in cases where the voxel is classified as both target and OAR, we default to target. All unclassified tissue is left as the original CT image grayscale.

Let $\mathbf{x}$ denote an RT treatment decision variable, consisting of the dose and the beamlet intensities. For each patient k in our data set, we associate parameters $(\mathbf{C}_k, \mathbf{A}_k, \mathbf{b}_k)$ and a corresponding multi-objective linear optimization problem, the *Forward Optimization Radiation Therapy Problem*, $\mathbf{RT}$–$\mathbf{FO}(\boldsymbol{\alpha}_k) : \min_{\mathbf{x}} \left\{ \boldsymbol{\alpha}_k^\top \mathbf{C}_k \mathbf{x} \mid \mathbf{A}_k \mathbf{x} \geq \mathbf{b}_k \, , \, \mathbf{x} \geq \mathbf{0} \right\}$, where $\mathbf{C}_k$ is the matrix whose rows represent different cost vectors and $\boldsymbol{\alpha}_k$ is a vector of objective weights. In the iterative clinical procedure (e.g., see Figure 1.1), a dosimetrist would tune $\boldsymbol{\alpha}_k$ to generate a treatment and an oncologist would review the optimal solution of $\mathbf{RT}$–$\mathbf{FO}(\boldsymbol{\alpha}_k)$ and suggest areas of improvement. Note that the optimization problem

for each patient is distinct. That is, each patient has a different $\mathbf{A}_k$ and $\mathbf{b}_k$, since these constraints must also encode a patient's physical geometry (i.e., the specific location of the OARs and PTVs for that patient). Furthermore, $\mathbf{C}_k$ may vary from patient to patient, since not every patient has all 10 structures. For example, some patients may only have two rather than three PTVs. We provide the full formulation of **RT–FO**($\boldsymbol{\alpha}_k$) in Appendix A.3.

Chapter 3

Ensemble inverse linear optimization

Inverse optimization is a longstanding approach for gaining insight into decision-generating processes and guiding subsequent decision-making. Inverse optimization has been used in diverse fields, for example capturing equilibrium estimates of asset returns for future portfolio optimization (Bertsimas et al., 2012), using past electricity market bids to forecast power consumption (Saez-Gallego et al., 2016), and estimating incentives to design future health insurance subsidies (Aswani et al., 2019).

Inverse optimization determines optimization model parameters to render a data set of observed decisions minimally sub-optimal for the model. The inverse optimization literature considers different settings that vary based on data characteristics (e.g., a single feasible decision or multiple points from different instances) or the optimization model (e.g., a linear or convex forward problem). A practitioner also chooses a sub-optimality measure to minimize, of which there exist three main variants. The first variant, known as the absolute duality gap, minimizes the difference between the objective values incurred by data and an imputed optimal value (Bertsimas et al., 2015; Esfahani et al., 2018; Saez-Gallego et al., 2016; Zhao et al., 2015). The second variant, known as the relative duality gap, minimizes the ratio instead of the absolute difference and has been studied for inverse linear optimization with a singleton data set (Babier et al., 2018b; Chan et al., 2014, 2019). These two methods are referred to as *objective space* models. The third variant is a *decision space* model that minimizes the distance between observed and optimal decisions (Aswani et al., 2018, 2019; Esfahani et al., 2018).

This chapter explores an ensemble inverse optimization framework using an arbitrary data set of decisions for a single linear optimization forward model. Our general motivation is as follows. Consider a single decision-making problem which we model as a linear program whose cost vector must be estimated. Multiple experts generate decisions for the problem. These experts may be human decision makers with their own parameter

estimates or even different heuristics applied to the problem; their proposed decisions may be sub-optimal or even infeasible. Using these decisions, we impute a single cost vector that best represents the optimization problem attempted by the experts (Troutt, 1995). We then re-solve the problem with the imputed parameter to generate an optimal decision of similar solution quality to the candidate decisions.

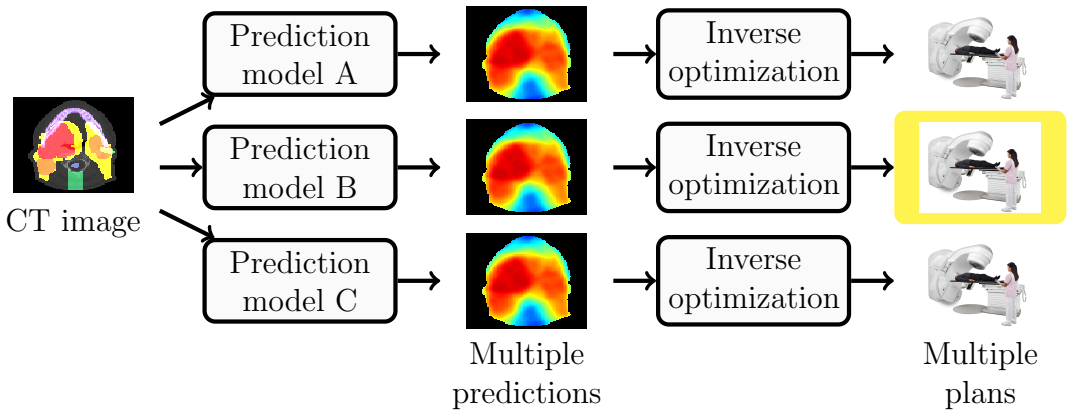
Our setting is analogous to ensemble methods in machine learning. Consider the canonical example of a random forest, which averages predictions from a set of decision trees (Breiman, 2001). Individual trees train on different subsets of data similar to how individual experts use different experiences to guide their decision making. An ensemble method averages out the biases of the individual models, just as inverse optimization learns an objective that balances the biases of different decision makers (Troutt, 1995). Practical evidence from machine learning shows ensemble methods generally outperform base prediction models. We similarly show in our application that ensemble inverse optimization can improve over approaches based on individual decisions.

Application to radiation therapy

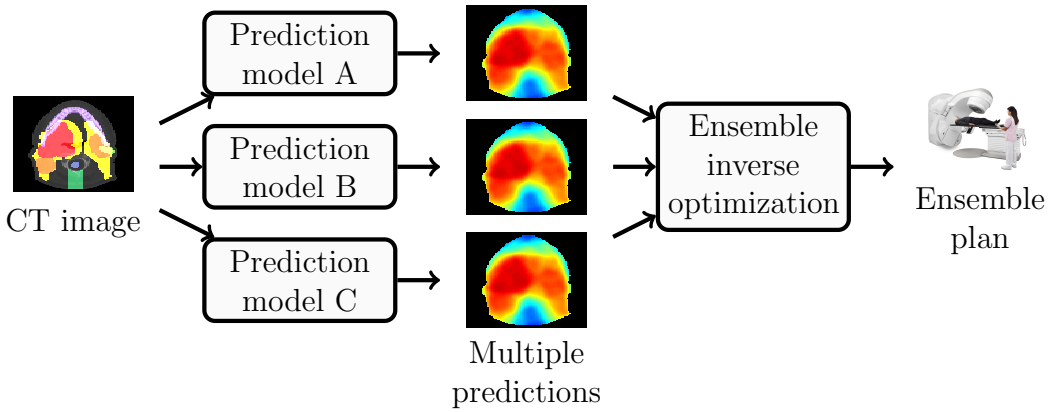
Inverse optimization can be used to learn the parameters of radiation therapy (RT) treatment planning optimization models for head-and-neck cancer patients (Babier et al., 2018a,b). Treatments are designed by solving a multi-objective optimization model, for which in the clinical practice, the objective weights are obtained by iterative parameter tuning (see Figure 1.1). This requires several days to finalize for a single patient and leads to operational strains and potential delays (Das et al., 2009), thereby necessitating KBP as a means of automating the process. The key intuition for inverse optimization in this application is that a predicted dose obtained in the first stage of KBP can be treated as an "observed decision" for the forward radiation therapy optimization problem.

The variety of machine learning models each predict different representations of dose and have their own advantages and disadvantages. Since treatment plans are clinically evaluated on a set of competing dosimetric criteria, different prediction models lead to plans that find different trade-offs between the clinical evaluation criteria. Given a plethora of prediction models where none are strictly dominating, a naive approach may take each prediction, generate a corresponding treatment plan via optimization, and then compare the plans on their dosimetric performance to identify the best plan for a patient (see Figure 3.1a). However, this approach is excessively laborious and the final plan is still determined from a prediction model that may be over-fit to specific clinical criteria.

We propose a natural alternative, which has not been previously considered, to obtain



(a) Multiple predictors and optimizers. After evaluation, the best plan (highlighted) is used.



(b) Multiple predictors are ensembled into one optimization to produce a single plan.

Figure 3.1: Existing KBP pipelines versus our proposed ensemble approach.

plans that better fit all of the clinical criteria. Analogous to an ensemble learning model combining weak predictors to form a better estimate, we harness a set of prediction models into an ensemble inverse optimization model that yields a single treatment plan that captures the best qualities of all of the estimates (see Figure 3.1b). Differences in the prediction models imitate the biases of different clinical experts that may lead them to suggest different plans for a given patient, even though they all aim to satisfy the same clinical criteria. Our inverse optimization model is a consensus-building treatment planner whose plans compromise between predictions to satisfy aggregate metrics better than any individual model.

Contributions

Methodologically, we extend the generalized inverse optimization framework in Chan et al. (2019), which considered only a single feasible decision ("single-point"), to the case of multiple observed decisions ("ensemble") of arbitrary feasibility. Previous results do not trivially generalize to our assumption-free settings and we require a suite of several new proof techniques to extend the results in these two directions. Our framework is founded on a flexible model template and specializes to several different models via the specification of model hyperparameters. We develop methods to impute the best-fit cost vector for a variety of different loss measures under a general setting (i.e., no assumptions on data), while also introducing efficient techniques under mild application-specific assumptions. Finally, we generalize a previous goodness-of-fit metric for inverse optimization (Chan et al., 2019) to the ensemble case. Together, the model and goodness of fit metric form a unified framework for model fitting and evaluation in inverse optimization applicable to arbitrary decision data for a single linear optimization problem.

Data-driven inverse optimization has received growing interest, particularly for learning in a class of parametrized convex forward problems (Aswani et al., 2018; Bertsimas et al., 2015; Esfahani et al., 2018; Keshavarz et al., 2011). Contrasting previous chapters, which consider a separate feasible set for each decision, our methods are tailored for a single feasible set, given the motivating assumption that different decision makers are solving the same forward problem. This leads to more efficient solution algorithms that leverage the geometry of linear programming. We further develop new bounds relating the performance of different variants that are tighter than previous bounds for the general convex case if applied to the linear case (Bertsimas et al., 2015; Esfahani et al., 2018).

The specific contributions of this chapter are as follows:

1. We develop an inverse linear optimization framework applicable to decision data sets of arbitrary size and feasibility for a single optimization problem, motivated by ensemble learning methods. This model is expressed in terms of hyperparameters used to derive different model variants.

2. We develop exact and assumption-free solution methods for each of the model variants. Under mild data assumptions, we demonstrate how geometric insights from linear optimization can lead to efficient and even analytic solution approaches.

3. We propose a goodness-of-fit metric measuring the model-data fit between a forward problem and arbitrary decision data. We prove several intuitive properties of the metric, including optimality with respect to the inverse optimization model,

boundedness, and monotonicity.

4. We implement the first ensemble-based automated planning pipeline in radiation therapy, using multiple predictions to design a single treatment for head-and-neck cancer patients. Our plans achieve better clinical trade-offs and our domain-independent goodness-of-fit metric validates our approach.

3.1 Background

We first review the formulation and main results from Chan et al. (2019), which introduced an inverse optimization model for linear optimization problems (linear programs) unifying both decision and objective space models, but only for a data set with a single feasible observed decision. Let $\mathbf{x}, \mathbf{c} \in \mathbb{R}^n$ denote the decision and cost vectors, respectively, and $\mathbf{A} \in \mathbb{R}^{m \times n}, \mathbf{b} \in \mathbb{R}^m$ denote the constraint matrix and right-hand side vector, respectively. Let $\mathcal{I} = \{1, \ldots, m\}$ and $\mathcal{J} = \{1, \ldots, n\}$. We refer to the following linear program as the forward optimization model

$$\begin{aligned} \mathbf{FO}(\mathbf{c}): \quad & \underset{\mathbf{x}}{\text{minimize}} \quad \mathbf{c}^\mathsf{T}\mathbf{x} \\ & \text{subject to} \quad \mathbf{x} \in \mathcal{P} := \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{b}\}. \end{aligned}$$

We assume that $\mathcal{P}$ is full-dimensional and that $\mathbf{FO}(\mathbf{c})$ has no redundant constraints. Given a *feasible* decision $\hat{\mathbf{x}} \in \mathcal{P}$, the *single-point* generalized inverse linear optimization problem is

$$\begin{aligned} \mathbf{GIO}(\{\hat{\mathbf{x}}\}): \quad & \underset{\mathbf{c},\mathbf{y},\boldsymbol{\epsilon}}{\text{minimize}} \quad \|\boldsymbol{\epsilon}\| && \text{(3.1a)} \\ & \text{subject to} \quad \mathbf{A}^\mathsf{T}\mathbf{y} = \mathbf{c}, \;\; \mathbf{y} \geq \mathbf{0} && \text{(3.1b)} \\ & \qquad\qquad \mathbf{c}^\mathsf{T}\hat{\mathbf{x}} = \mathbf{b}^\mathsf{T}\mathbf{y} + \mathbf{c}^\mathsf{T}\boldsymbol{\epsilon} && \text{(3.1c)} \\ & \qquad\qquad \|\mathbf{c}\|_N = 1 && \text{(3.1d)} \\ & \qquad\qquad \mathbf{c} \in \mathcal{C}, \boldsymbol{\epsilon} \in \mathcal{E}. && \text{(3.1e)} \end{aligned}$$

Above, $\mathbf{y} \in \mathbb{R}^m$ represents the dual vector for the constraints of the forward problem. Constraints (3.1b) ensures $\mathbf{y}$ is dual feasible with respect to $\mathbf{c}$. Constraint (3.1c) connects $\mathbf{c}$ and $\mathbf{y}$ with a perturbation vector $\boldsymbol{\epsilon} \in \mathbb{R}^n$ by enforcing that the pair $(\hat{\mathbf{x}} - \boldsymbol{\epsilon}, \mathbf{y})$ satisfy strong duality with respect to $\mathbf{c}$. Note that these constraints do not imply that the pair is primal-dual optimal (as we have not enforced primal feasibility), but rather that $\hat{\mathbf{x}} - \boldsymbol{\epsilon}$ lies on a supporting hyperplane $\{\mathbf{x} \mid \mathbf{c}^\mathsf{T}\mathbf{x} = \mathbf{b}^\mathsf{T}\mathbf{y}\}$ of the feasible set. Constraint (3.1d) is

a normalization constraint to prevent the trivial solution of $\mathbf{c} = \mathbf{0}$, where $\|\cdot\|_N$ denotes an arbitrary norm that may differ from the one in the objective. Finally, constraints (3.1e) define application-specific perturbation and cost vectors via the sets $\mathcal{E}$ and $\mathcal{C}$, respectively. We also leave the choice of the norm in the objective open. The tuple $(\|\cdot\|, \|\cdot\|_N, \mathcal{C}, \mathcal{E})$ forms the inverse optimization model *hyperparameters*. By selecting them appropriately, $\mathbf{GIO}(\{\hat{\mathbf{x}}\})$ specializes into models that minimize error in objective or decision space.

Although $\mathbf{GIO}(\{\hat{\mathbf{x}}\})$ is non-convex, it admits a closed-form solution for $\hat{\mathbf{x}} \in \mathcal{P}$, which can be determined by projecting $\hat{\mathbf{x}}$ to the boundary of $\mathcal{P}$ of minimum distance as measured by $\|\cdot\|$. Specifically, let $\mathcal{H}_i = \left\{\mathbf{x} \mid \mathbf{a}_i^\mathsf{T}\mathbf{x} = b_i\right\}$ be the hyperplane corresponding to the i^{th} constraint and

$$\pi_i(\hat{\mathbf{x}}) = \underset{\mathbf{x}\in\mathcal{H}_i}{\arg\min} \|\hat{\mathbf{x}} - \mathbf{x}\| \tag{3.2}$$

be the projection of $\hat{\mathbf{x}}$ to $\mathcal{H}_i$. The hyperplane projection problem has an analytic solution $\pi_i(\hat{\mathbf{x}}) = \hat{\mathbf{x}} - \frac{\mathbf{a}_i^\mathsf{T}\hat{\mathbf{x}} - b_i}{\|\mathbf{a}_i\|^D}\nu(\mathbf{a}_i)$, where $\|\cdot\|^D$ is the dual norm of $\|\cdot\|$ and $\nu(\mathbf{a}_i) \in \arg\max_{\|\mathbf{v}\|=1}\left\{\mathbf{v}^\mathsf{T}\mathbf{a}_i\right\}$ (Mangasarian, 1999) . This result leads to an analytic characterization of an optimal solution.

Theorem 1 (Chan et al., 2019)**.** *Let $\hat{\mathbf{x}} \in \mathcal{P}$, $i^* \in \arg\min_{i\in\mathcal{I}}\left\{\frac{\mathbf{a}_i^\mathsf{T}\hat{\mathbf{x}} - b_i}{\|\mathbf{a}_i\|^D}\right\}$, and $\mathbf{e}_i$ be the i^{th} unit vector. There exists an optimal solution to $\mathbf{GIO}(\{\hat{\mathbf{x}}\})$ of the form*

$$(\mathbf{c}^*, \mathbf{y}^*, \boldsymbol{\epsilon}^*) = \left(\frac{\mathbf{a}_{i^*}}{\|\mathbf{a}_{i^*}\|_N}, \frac{\mathbf{e}_{i^*}}{\|\mathbf{a}_{i^*}\|_N}, \hat{\mathbf{x}} - \pi_{i^*}(\hat{\mathbf{x}})\right). \tag{3.3}$$

If $\hat{\mathbf{x}} \in \mathcal{P}$, then by Theorem 1, an optimal cost vector describes a supporting hyperplane (i.e., $\left\{\mathbf{x} \mid \mathbf{c}^{*\mathsf{T}}\mathbf{x} = \mathbf{b}^\mathsf{T}\mathbf{y}^*\right\}$) that also corresponds to a constraint of the forward problem.

3.2 Ensemble inverse linear optimization

We extend $\mathbf{GIO}(\{\hat{\mathbf{x}}\})$ to the case of multiple observed decisions with no data assumptions. Let $\mathcal{D} = \{\hat{\mathbf{x}}_1, \ldots, \hat{\mathbf{x}}_Q\}$ be a data set, i.e., an ensemble of Q observed decisions, indexed by $\mathcal{Q} = \{1, \ldots, Q\}$. We seek to impute a single cost vector $\mathbf{c}^*$ that minimizes the aggregate loss over all decisions.

Given that Theorem 1 admits an analytic solution, one computationally desirable approach may be to solve $\mathbf{GIO}(\{\hat{\mathbf{x}}_q\})$ for each $\hat{\mathbf{x}}_q$ and impute a set of cost vector estimates. We may then consider classical ensemble methods like a random forest, which average weak predictions (Breiman, 2001). However, such a method applied to our setting effectively ignores the geometry of $\mathcal{P}$, which provides useful information in the estimation of

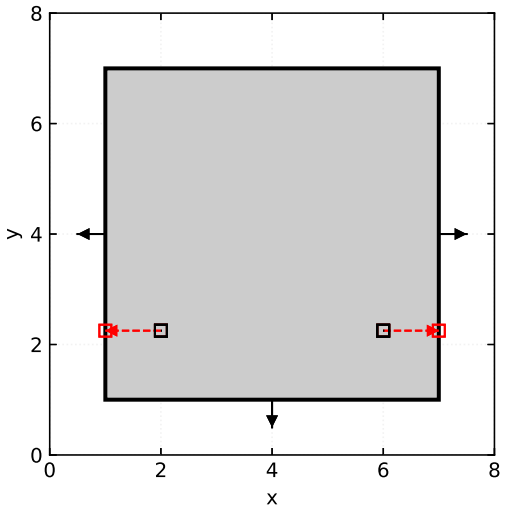


Figure 3.2: Illustration of Example 1. The feasible set for $\mathbf{FO}(\mathbf{c})$ is shaded. The black and red squares are $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}-\hat{\boldsymbol{\epsilon}}^*$ from solving $\mathbf{GIO}(\{\hat{\mathbf{x}}\})$ independently. The solid arrows are potential cost vectors.

a single cost vector. For example, naively averaging the set of cost vectors to obtain a consensus may lead to pathological outcomes.

Example 1. *Let* $\mathbf{FO}(\mathbf{c}) : \min_{\mathbf{x}}\{c_1x_1 + c_2x_2 \mid x_1 \leq 7,\ x_2 \leq 7,\ x_1 \geq 1,\ x_2 \geq 1\}$ *and consider* $\mathcal{D} = \{(2, 2.25), (6, 2.25)\}$. *Solving* $\mathbf{GIO}(\{(2, 2.25)\})$ *and* $\mathbf{GIO}(\{(6, 2.25)\})$ *yields cost vectors* $(-1, 0)$ *and* $(1, 0)$*, respectively, with an average cost vector* $\bar{\mathbf{c}} = \mathbf{0}$*. Note that a more intuitive best-fit cost vector would be* $\mathbf{c}^* = (0, -1)$*, pointing to the bottom facet of* $\mathcal{P}$*. Figure 3.2 illustrates this example.*

Instead, we design an ensemble inverse optimization model to minimize the aggregate error induced by all points with respect to a single imputed cost vector. We introduce perturbation vectors $\boldsymbol{\epsilon}_q$ for every $q \in \mathcal{Q}$ and form our problem:

$$
\begin{aligned}
\mathbf{GIO}(\mathcal{D}): \quad \underset{\mathbf{c},\mathbf{y},\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_Q}{\text{minimize}} \quad & \sum_{q=1}^{Q} \|\boldsymbol{\epsilon}_q\| && \text{(3.4a)}\\
\text{subject to} \quad & \mathbf{A}^\mathsf{T}\mathbf{y} = \mathbf{c},\ \ \mathbf{y} \geq \mathbf{0} && \text{(3.4b)}\\
& \mathbf{c}^\mathsf{T}\hat{\mathbf{x}}_q = \mathbf{b}^\mathsf{T}\mathbf{y} + \mathbf{c}^\mathsf{T}\boldsymbol{\epsilon}_q, \quad \forall q \in \mathcal{Q} && \text{(3.4c)}\\
& \|\mathbf{c}\|_N = 1 && \text{(3.4d)}\\
& \mathbf{c} \in \mathcal{C}, \boldsymbol{\epsilon}_q \in \mathcal{E}_q, \quad \forall q \in \mathcal{Q}. && \text{(3.4e)}
\end{aligned}
$$

Constraints (3.4b) and (3.4d) are carried from the single-point model. (3.4c) and (3.4e) are ensemble extensions of (3.1c) and (3.1e) respectively, ensuring that for each $q \in \mathcal{Q}$,

the data points $\hat{\mathbf{x}}_q$ achieve strong duality with respect to $\mathbf{c}$ after being perturbed by $\boldsymbol{\epsilon}_q \in \mathcal{E}_q$. The objective minimizes the sum of the norms of the individual perturbation vectors. Note that this problem is non-convex due to the bilinear terms in (3.4c) and the normalization constraint (3.4d). We first show that $\mathbf{GIO}(\mathcal{D})$ specializes to objective and decision space variants, before developing tailored solution methods.

3.2.1 Objective space

Inverse linear optimization in the objective space is based on the premise that sub-optimal observed decisions are characterized by sub-optimal objective values. Consider the dual problem for $\mathbf{FO}(\mathbf{c})$. For each decision $\hat{\mathbf{x}}_q$, the corresponding duality gap is a distance measure between the objective value of $\hat{\mathbf{x}}_q$ and the optimal value of the dual problem. By choosing the norm in the objective (3.4a) and the sets $\mathcal{E}_q$ for each $q \in \mathcal{Q}$ appropriately, the problem is transformed to measure a function of the duality gap. We consider two objective space models, the absolute and relative duality gaps.

Absolute duality gap.

The absolute duality gap method minimizes the aggregate duality gap between the primal objectives of each decision and the imputed dual optimal value:

$$\begin{aligned}
\mathbf{GIO}_{\mathrm{A}}(\mathcal{D}): \quad \underset{\mathbf{c},\mathbf{y},\epsilon_1,\ldots,\epsilon_Q}{\text{minimize}} \quad & \sum_{q=1}^{Q} |\epsilon_q| && \text{(3.5a)}\\
\text{subject to} \quad & \mathbf{A}^{\mathsf{T}}\mathbf{y} = \mathbf{c}, \;\; \mathbf{y} \geq \mathbf{0} && \text{(3.5b)}\\
& \mathbf{c}^{\mathsf{T}}\hat{\mathbf{x}}_q = \mathbf{b}^{\mathsf{T}}\mathbf{y} + \epsilon_q, \quad \forall q \in \mathcal{Q} && \text{(3.5c)}\\
& \|\mathbf{c}\|_N = 1. && \text{(3.5d)}
\end{aligned}$$

This model specializes $\mathbf{GIO}(\mathcal{D})$ by measuring error in terms of scalar duality gap variables. We show that it can be recovered from $\mathbf{GIO}(\mathcal{D})$ with an appropriate choice of model hyperparameters.

Proposition 1. *Let $\mu(\mathbf{c}) \in \mathbb{R}^n$ be a parameter satisfying $\|\mu(\mathbf{c})\|_\infty = 1$ and let $\mu(\mathbf{c})^{\mathsf{T}}\mathbf{c} = 1$. A solution $\left(\mathbf{c}^*, \mathbf{y}^*, \epsilon_1^*, \ldots, \epsilon_Q^*\right)$ is optimal to $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$ if and only if the solution $\left(\mathbf{c}^*, \mathbf{y}^*, \epsilon_1^*\mu(\mathbf{c}^*), \ldots, \epsilon_Q^*\mu(\mathbf{c}^*)\right)$ is optimal to $\mathbf{GIO}(\mathcal{D})$ with hyperparameters*

$$\left(\|\cdot\|, \|\cdot\|_N, \mathcal{C}, \mathcal{E}_1, \ldots, \mathcal{E}_Q\right) = \left(\|\cdot\|_\infty, \|\cdot\|_N, \mathbb{R}^n, \{\epsilon_1\mu(\mathbf{c})\}, \ldots, \{\epsilon_Q\mu(\mathbf{c})\}\right).$$

Proof. For any $\mathbf{c}$, setting each $\boldsymbol{\epsilon}_q = \epsilon_q\mu(\mathbf{c})$ implies $\|\boldsymbol{\epsilon}_q\|_\infty = |\epsilon_q| \, \|\mu(\mathbf{c})\|_\infty = |\epsilon_q|$. Thus, (3.4a)

becomes (3.5a). Similarly, (3.4c) becomes (3.5c), since $\mathbf{c}^\mathsf{T}\boldsymbol{\epsilon}_q = \epsilon_q \mathbf{c}^\mathsf{T}\mu(\mathbf{c}) = \epsilon_q$. Then, any feasible solution to $\mathbf{GIO}(\mathcal{D})$ with the suggested hyperparameters yields a feasible solution to $\mathbf{GIO}_\mathrm{A}(\mathcal{D})$ and vice versa, with the same objective value. □

Proposition 1 shows that the specialization of $\mathbf{GIO}(\mathcal{D})$ to $\mathbf{GIO}_\mathrm{A}(\mathcal{D})$ depends on each $\boldsymbol{\epsilon}_q$ being a rescaling of some $\mu(\mathbf{c})$ that is dependent only on the cost vector. Note that $\mu(\mathbf{c})$ is only a vehicle to aid in the specialization of $\mathbf{GIO}(\mathcal{D})$, and is useful to interpret solutions of $\mathbf{GIO}_\mathrm{A}(\mathcal{D})$ in the context of $\mathbf{GIO}(\mathcal{D})$. For all $\mathbf{c}$ satisfying $\|\mathbf{c}\|_N = 1$, $\mu(\mathbf{c})$ must satisfy $\|\mu(\mathbf{c})\|_\infty = 1$ and $\mu(\mathbf{c})^\mathsf{T}\mathbf{c} = 1$. Given a specific $\|\cdot\|_N$, we can propose a structured $\mu(\mathbf{c})$. For example, if $\|\cdot\|_N = \|\cdot\|_1$, let $\mu(\mathbf{c}) = \operatorname{sgn}(\mathbf{c})$ be the sign vector of $\mathbf{c}$, ensuring that the conditions on $\mu(\mathbf{c})$ are satisfied for all $\mathbf{c}$ with $\|\mathbf{c}\|_1 = 1$. If $\|\cdot\|_N = \|\cdot\|_\infty$, let $\mu(\mathbf{c}) = \operatorname{sgn}(c_{j^*})\mathbf{e}_{j^*}$ be the j^*-th unit vector, where $j^* \in \arg\max_{j\in\mathcal{J}} \{|c_j|\}$.

General solution method. Since the normalization constraint is the sole non-convexity in $\mathbf{GIO}_\mathrm{A}(\mathcal{D})$, this model can be solved exactly by polyhedral decomposition. The efficiency of this approach depends on the choice of the norm. For example, $2n$ linear programs are needed if $\|\cdot\|_N = \|\cdot\|_\infty$.

Theorem 2. *Let* $\left(\mathbf{c}^*, \mathbf{y}^*, \epsilon_1^*, \ldots, \epsilon_Q^*\right)$ *be optimal to* $\mathbf{GIO}_\mathrm{A}(\mathcal{D})$ *under* $\|\cdot\|_N = \|\cdot\|_\infty$. *There exists* $j \in \mathcal{J}$ *such that* $\left(\mathbf{c}^*, \mathbf{y}^*, \epsilon_1^*, \ldots, \epsilon_Q^*\right)$ *is also optimal to* $\mathbf{GIO}_\mathrm{A}(\mathcal{D}; j)$, *defined as:*

$$
\begin{aligned}
\mathbf{GIO}_\mathrm{A}(\mathcal{D}; j): \quad \underset{\mathbf{c},\mathbf{y},\epsilon_1,\ldots,\epsilon_Q}{\text{minimize}} \quad & \sum_{q=1}^{Q} |\epsilon_q| \\
\text{subject to} \quad & \mathbf{A}^\mathsf{T}\mathbf{y} = \mathbf{c}, \;\; \mathbf{y} \geq \mathbf{0} \\
& \mathbf{c}^\mathsf{T}\hat{\mathbf{x}}_q = \mathbf{b}^\mathsf{T}\mathbf{y} + \epsilon_q, \quad \forall q \in \mathcal{Q} \\
& (c_j = 1) \vee (c_j = -1) \\
& |c_k| \leq 1, \quad \forall k \in \mathcal{J}/\{j\}.
\end{aligned}
\tag{3.6}
$$

Proof. Let $j^* \in \arg\max_{j\in\mathcal{J}} \left\{|c_j^*|\right\}$, implying $|c_{j^*}^*| = 1$. Then, $\left(\mathbf{c}^*, \mathbf{y}^*, \epsilon_1^*, \ldots, \epsilon_Q^*\right)$ is feasible to $\mathbf{GIO}_\mathrm{A}(\mathcal{D}; j^*)$. Conversely, for any $j \in \mathcal{J}$, every feasible solution to $\mathbf{GIO}_\mathrm{A}(\mathcal{D}; j)$ is feasible to $\mathbf{GIO}_\mathrm{A}(\mathcal{D})$, so all optimal solutions to each $\mathbf{GIO}_\mathrm{A}(\mathcal{D}; j)$ lie in the feasible set of $\mathbf{GIO}_\mathrm{A}(\mathcal{D})$. □

For each j, the problem $\mathbf{GIO}_\mathrm{A}(\mathcal{D}; j)$ separates into two linear programs (one with the constraint $c_j = 1$ and the other with $c_j = -1$), thus totaling $2n$ linear programs. When $\|\cdot\|_N \neq \|\cdot\|_\infty$ in general, an exponential number of linear programs may be required. We next discuss special cases that simplify the solution approach for $\mathbf{GIO}_\mathrm{A}(\mathcal{D})$.

Non-negative cost vectors. In many real-world applications, feasible cost vectors should be non-negative (i.e., $\mathcal{C} \subseteq \mathbb{R}^n_+$). Here, it is advantageous to set $\|\cdot\|_N = \|\cdot\|_1$, because the normalization constraint becomes $\mathbf{c}^\mathsf{T}\mathbf{1} = 1$ and $\mathbf{GIO}_\mathrm{A}(\mathcal{D})$ simplifies to a single linear optimization problem.

Feasible observed decisions. Most inverse optimization literature focuses on the situation where all observed decisions are feasible for the forward model (i.e., $\mathcal{D} \subset \mathcal{P}$). In this case, $\mathcal{D}$ can be replaced by the singleton $\{\bar{\mathbf{x}}\}$, where $\bar{\mathbf{x}}$ is the centroid of the points in $\mathcal{D}$. A similar result was presented in Goli (2015, Chapter 4), but for a model with a different normalization constraint that did not prevent trivial solutions. We present the analogous result in the context of our model (3.5).

Proposition 2. *If $\mathcal{D} \subset \mathcal{P}$ and $\bar{\mathbf{x}}$ is the centroid of $\mathcal{D}$, $\mathbf{GIO}_\mathrm{A}(\mathcal{D})$ is equivalent to $\mathbf{GIO}_\mathrm{A}(\{\bar{\mathbf{x}}\})$.*

Proof. If all observations are feasible, then by weak duality $\epsilon_q \geq 0 \ \forall q \in \mathcal{Q}$, and we can simplify the objective function

$$\sum_{q=1}^{Q} |\epsilon_q| = \sum_{q=1}^{Q} \epsilon_q = \sum_{q=1}^{Q} \left(\mathbf{c}^\mathsf{T}\hat{\mathbf{x}}_q - \mathbf{b}^\mathsf{T}\mathbf{y}\right) = \left(\mathbf{c}^\mathsf{T}\bar{\mathbf{x}} - \mathbf{b}^\mathsf{T}\mathbf{y}\right) Q,$$

where the last equality follows by the definition of the centroid (i.e., $\bar{\mathbf{x}} = \sum_{q=1}^{Q} \hat{\mathbf{x}}_q/Q$). We similarly compress constraint (3.5c) to a single constraint for $\bar{\mathbf{x}}$, resulting in $\mathbf{GIO}_\mathrm{A}(\{\bar{\mathbf{x}}\})$. □

Together, Proposition 2 and Theorem 1 imply that $\mathbf{GIO}_\mathrm{A}(\mathcal{D})$ is analytically solvable when $\mathcal{D} \subset \mathcal{P}$.

Infeasible observed decisions. Finally, we address scenarios where the observed decisions are all infeasible. We first consider the case where $\mathcal{D}$ is a single, infeasible observed decision $\hat{\mathbf{x}}$ in which case $\mathbf{GIO}_\mathrm{A}(\{\hat{\mathbf{x}}\})$ possesses an analytic solution. In contrast, the original work in (Chan et al., 2019) restricted observed decision points to lie within $\mathcal{P}$.

Proposition 3. *Assume $\hat{\mathbf{x}} \notin \mathcal{P}$.*

1. *If $\hat{\mathbf{x}}$ satisfies $\mathbf{a}_i^\mathsf{T}\hat{\mathbf{x}} > b_i$ for some $i \in \mathcal{I}$, then there also exists $i^* \in \mathcal{I}$ such that $\tilde{\mathbf{y}}$ is*

$$\tilde{y}_i = \frac{1}{\mathbf{a}_i^\mathsf{T}\hat{\mathbf{x}} - b_i}, \quad \tilde{y}_{i^*} = \frac{1}{b_{i^*} - \mathbf{a}_{i^*}^\mathsf{T}\hat{\mathbf{x}}}, \quad \tilde{y}_k = 0 \quad \forall k \in \mathcal{I} \setminus \{i, i^*\} \tag{3.7}$$

and $\tilde{\mathbf{c}} = \mathbf{A}^\mathsf{T}\tilde{\mathbf{y}}$. *The corresponding normalized solution*

$$\left(\mathbf{c}^*, \mathbf{y}^*, \epsilon^*\right) = \left(\tilde{\mathbf{c}}/\left\|\tilde{\mathbf{c}}\right\|_N, \tilde{\mathbf{y}}/\left\|\tilde{\mathbf{c}}\right\|_N, 0\right)$$

is an optimal solution to $\mathbf{GIO}_{\mathrm{A}}(\{\hat{\mathbf{x}}\})$ *and the optimal value is* 0.

2. *If* $\mathbf{A}\hat{\mathbf{x}} \leq \mathbf{b}$, *there exists* $i^* \in \mathcal{I}$ *such that* (3.3) *is an optimal solution to* $\mathbf{GIO}_{\mathrm{A}}(\{\hat{\mathbf{x}}\})$.

Proof.

1. Assume without loss of generality that there exist $i, j \in \mathcal{I}$ such that $\mathbf{a}_i^\mathsf{T}\hat{\mathbf{x}} > b_i$ and $\mathbf{a}_j^\mathsf{T}\hat{\mathbf{x}} < b_j$, respectively. The corresponding $\tilde{y}$ defined in (3.7) satisfies the strong duality constraint (3.5c) with $\epsilon = 0$. Furthermore, $(\tilde{\mathbf{c}}, \tilde{\mathbf{y}})$ satisfy the duality feasibility constraints (3.5b) by construction. We normalize the solution to satisfy constraint (3.5d). The normalized solution still satisfies all other constraints. This solution is feasible for $\mathbf{GIO}_{\mathrm{A}}(\{\hat{\mathbf{x}}\})$ with zero cost and is thus optimal.

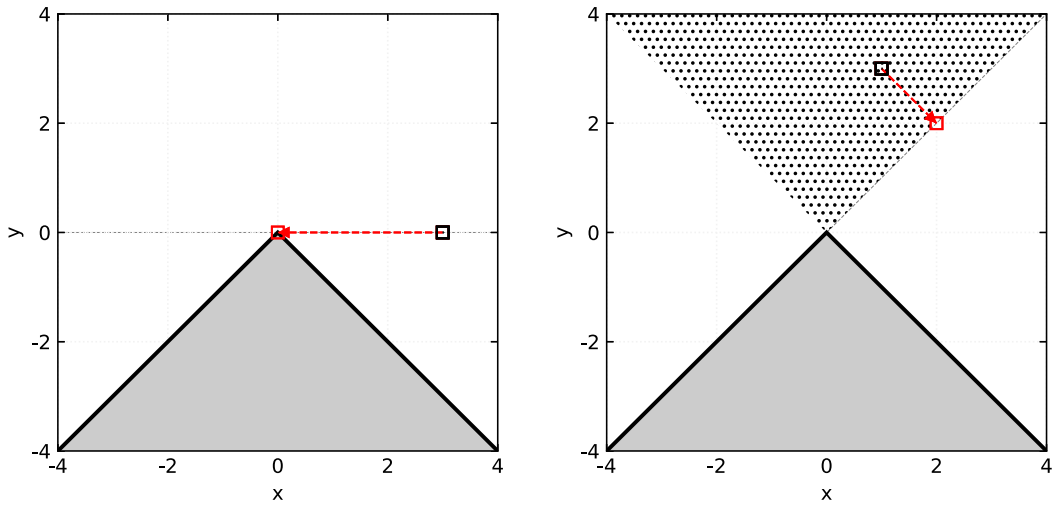
2. Here, the duality gap is non-positive (i.e., $\epsilon \leq 0$). We rewrite the single-point version of (3.5) with $\delta = -\epsilon$, shown in model (3.8) below. Now consider the forward problem $\min_{\mathbf{x}}\{-\mathbf{c}^\mathsf{T}\mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$ with the observed solution $\hat{\mathbf{x}}$ and the corresponding inverse optimization model (3.9).

$$\begin{array}{lll} \underset{\mathbf{c},\mathbf{y},\delta}{\text{minimize}} & \delta & \\ \text{subject to} & \mathbf{A}^\mathsf{T}\mathbf{y} = \mathbf{c},\ \mathbf{y} \geq \mathbf{0} & \\ & \mathbf{c}^\mathsf{T}\hat{\mathbf{x}} = \mathbf{b}^\mathsf{T}\mathbf{y} - \delta & \\ & \|\mathbf{c}\|_N = 1. & \end{array} \tag{3.8}$$

$$\begin{array}{lll} \underset{\mathbf{c},\mathbf{y},\gamma}{\text{minimize}} & |\gamma| & \\ \text{subject to} & \mathbf{A}^\mathsf{T}\mathbf{y} = \mathbf{c},\ \mathbf{y} \geq \mathbf{0} & \\ & -\mathbf{c}^\mathsf{T}\hat{\mathbf{x}} = -\mathbf{b}^\mathsf{T}\mathbf{y} + \gamma & \\ & \|\mathbf{c}\|_N = 1. & \end{array} \tag{3.9}$$

By assumption, $\hat{\mathbf{x}}$ is feasible for the above-defined forward problem and therefore, $\gamma \geq 0$ in (3.9). Consequently, formulation (3.8) is equivalent to (3.9) after removing the absolute value in the objective and rearranging the duality gap constraint. We can solve formulation (3.9) using Theorem 1, arriving at an optimal solution for the original inverse optimization problem. □

Proposition 3 provides geometric insights regarding the structure of optimal solutions. In objective space inverse optimization, all points that lie on a level set of a cost vector yield the same duality gap. Recall that the hyperplane $\mathcal{H} = \left\{\mathbf{x} \mid \mathbf{c}^{*\mathsf{T}}\mathbf{x} = \mathbf{b}^\mathsf{T}\mathbf{y}^*\right\}$ is a supporting hyperplane of $\mathcal{P}$, or in other words, a level set of the cost vector with zero



(a) If one constraint is satisfied, $\hat{\mathbf{x}}$ projects to that constraint.

(b) If all constraints are violated, we solve the inverse problem for $\mathbf{FOA}(\mathbf{c})$ (hatched).

Figure 3.3: Illustration of Proposition 3. The feasible set for $\mathbf{FO}(\mathbf{c})$ is shaded. The black and red points are $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}} - \boldsymbol{\epsilon}^*$, respectively. The dashed line is the supporting hyperplane yielding an optimal value of 0.

duality gap. If $\hat{\mathbf{x}} \notin \mathcal{P}$ but satisfies $\mathbf{a}_i^\mathsf{T}\hat{\mathbf{x}} > b_i$ for some i, then there always exists a supporting hyperplane that intersects with $\hat{\mathbf{x}}$ (e.g., Figure 3.3(a)). If $\mathbf{A}\hat{\mathbf{x}} \leq \mathbf{b}$, then no such supporting hyperplane exists. However, consider the alternate forward problem $\mathbf{FOA}(\mathbf{c}) := \min_{\mathbf{x}} \left\{-\mathbf{c}^\mathsf{T}\mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}\right\}$ obtained by reversing the signs of all constraints and the cost vector. The single-point inverse problem for $\hat{\mathbf{x}}$ and $\mathbf{FOA}(\mathbf{c})$ is equivalent to the original problem. Since $\hat{\mathbf{x}}$ is feasible for $\mathbf{FOA}(\mathbf{c})$, Theorem 1 applies for $\mathbf{GIO}_{\mathrm{A}}(\{\hat{\mathbf{x}}\})$. Geometrically, the constraints of $\mathbf{FOA}(\mathbf{c})$ describe the nearest supporting hyperplanes of $\mathbf{FO}(\mathbf{c})$. Solving one problem solves the other (e.g., Figure 3.3(b), where $\hat{\mathbf{x}}$ projects to an infeasible point for $\mathbf{FO}(\mathbf{c})$ with no duality gap). Finally, we leverage this insight to the case with multiple infeasible decisions to show that if all data points violate all constraints, then the multi-point ensemble problem reduces to a single-point.

Corollary 1. *Suppose that* $\mathbf{A}\hat{\mathbf{x}}_q \leq \mathbf{b}$ *for all* $q \in \mathcal{Q}$*, and* $\mathcal{D} \subset \mathbb{R}^n \setminus \mathcal{P}$*. Let* $\bar{\mathbf{x}}$ *be the centroid of* $\mathcal{D}$*. Then,* $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$ *for the forward problem* $\mathbf{FO}(\mathbf{c})$ *is equivalent to* $\mathbf{GIO}_{\mathrm{A}}(\{\bar{\mathbf{x}}\})$ *for* $\mathbf{FOA}(\mathbf{c})$*.*

Proof. Since all observations are infeasible for the initial forward problem, the duality gap terms are all non-positive (i.e., $\epsilon_q \leq 0$ for all $q \in \mathcal{Q}$). As such, we use the same argument as used in Prop. 3 Part 2 to show that the formulation of $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$ is equivalent to the formulation of an absolute duality gap inverse optimization problem over the alternative

forward problem $\min_{\mathbf{x}}\{-\mathbf{c}^\mathsf{T}\mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$. As $\mathcal{D} \subset \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \leq \mathbf{b}\}$, Proposition 2 reduces the problem to $\mathbf{GIO}_{\mathrm{A}}(\{\bar{\mathbf{x}}\})$. □

Relative duality gap.

The relative duality gap variant minimizes the sum of the ratios between the duality gap for each decision and the imputed dual optimal value for the forward problem:

$$\begin{aligned}
\mathbf{GIO}_{\mathrm{R}}(\mathcal{D}): \quad \underset{\mathbf{c},\mathbf{y},\epsilon_1,\ldots,\epsilon_Q}{\text{minimize}} \quad & \sum_{q=1}^{Q} |\epsilon_q - 1| && \text{(3.10a)}\\
\text{subject to} \quad & \mathbf{A}^\mathsf{T}\mathbf{y} = \mathbf{c}, \ \ \mathbf{y} \geq \mathbf{0} && \text{(3.10b)}\\
& \mathbf{c}^\mathsf{T}\hat{\mathbf{x}}_q = \epsilon_q \mathbf{b}^\mathsf{T}\mathbf{y}, \quad \forall q \in \mathcal{Q} && \text{(3.10c)}\\
& \|\mathbf{c}\|_N = 1. && \text{(3.10d)}
\end{aligned}$$

Duality gap ratio variables ϵ_q replace the perturbation vectors used in the general formulation $\mathbf{GIO}(\mathcal{D})$. These variables are well-defined except when the imputed forward problem has an optimal value of 0. In this subsection, we assume $\mathbf{b} \neq \mathbf{0}$. Furthermore, note that if $\mathbf{b}^\mathsf{T}\mathbf{y} = \mathbf{0}$ for feasible $\mathbf{y}$, then ϵ_q are free variables; in this case, we assume $\epsilon_q := 1$ for all q. First, we show $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ can be recovered from $\mathbf{GIO}(\mathcal{D})$ with appropriate hyperparameters.

Proposition 4. *Let $\mu(\mathbf{c})$ be a function that satisfies $\|\mu(\mathbf{c})\|_\infty = 1$ and $\mu(\mathbf{c})^\mathsf{T}\mathbf{c} = 1$ for all $\mathbf{c}$. A solution $\left(\mathbf{c}^*, \mathbf{y}^*, \epsilon_1^*, \ldots, \epsilon_Q^*\right)$ for which $\mathbf{b}^\mathsf{T}\mathbf{y}^* \neq 0$, is optimal to $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ if and only if $\left(\mathbf{c}^*, \mathbf{y}^*, \mathbf{b}^\mathsf{T}\mathbf{y}^*\left(\epsilon_1^* - 1\right)\mu(\mathbf{c}^*), \ldots, \mathbf{b}^\mathsf{T}\mathbf{y}^*\left(\epsilon_Q^* - 1\right)\mu(\mathbf{c}^*)\right)$ is optimal to $\mathbf{GIO}(\mathcal{D})$ with hyperparameters $(\|\cdot\|, \|\cdot\|_N, \mathcal{C}, \mathcal{E}_1, \ldots, \mathcal{E}_Q)$ equal to*

$$\left(\|\cdot\|_\infty / |\mathbf{b}^\mathsf{T}\mathbf{y}^*|, \|\cdot\|_N, \mathbb{R}^n, \left\{\mathbf{b}^\mathsf{T}\mathbf{y}^*\left(\epsilon_1 - 1\right)\mu(\mathbf{c}^*)\right\}, \ldots, \left\{\mathbf{b}^\mathsf{T}\mathbf{y}^*\left(\epsilon_Q - 1\right)\mu(\mathbf{c}^*)\right\}\right).$$

Proof. For any $\mathbf{c}$, setting $\boldsymbol{\epsilon}_q = \mathbf{b}^\mathsf{T}\mathbf{y}\left(\epsilon_q - 1\right)\mu(\mathbf{c})$ forces $\|\boldsymbol{\epsilon}_q\|_\infty / |\mathbf{b}^\mathsf{T}\mathbf{y}| = |\epsilon_q - 1|$, giving us the objective (3.10a). The same substitution into (3.4c) gives the strong duality constraint (3.10c). Thus, every feasible solution of $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ has a corresponding feasible solution in $\mathbf{GIO}(\mathcal{D})$ (after setting the hyperparameters), and vice versa, with the same objective value. □

Remark 1. *Proposition 4 addresses the case where $\mathbf{b}^\mathsf{T}\mathbf{y}^* \neq 0$ only. However, if $\mathbf{b}^\mathsf{T}\mathbf{y}^* = 0$, $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ and $\mathbf{GIO}(\mathcal{D})$ are still equivalent in that they both yield an optimal value of 0. To see this, suppose that an optimal solution to $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ satisfies $\mathbf{b}^\mathsf{T}\mathbf{y}^* = 0$. Then*

for all $q \in \mathcal{Q}$, $\mathbf{c}^{*\mathsf{T}}\hat{\mathbf{x}}_q = 0$ *and since* ϵ_q *becomes a free variable, we set it to* 1 *and obtain an optimal value of* 0. *On the other hand, we can use the same* $(\mathbf{c}^*, \mathbf{y}^*, \mathbf{0}, \ldots, \mathbf{0})$ *as a feasible solution to* $\mathbf{GIO}(\mathcal{D})$ *and observe that setting* $\boldsymbol{\epsilon}_q = \mathbf{0}$ *for all* $q \in \mathcal{Q}$ *satisfies the strong duality constraint, giving an optimal value of* 0.

General solution method Unlike the absolute duality gap problem, which is non-convex only because of the normalization constraint, $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ possesses an additional non-convexity due to a bilinear term in the duality gap constraint (3.10c). We first address the bilinearity by introducing three sub-problems. We then use polyhedral decomposition to address the normalization constraint.

Proposition 5. *Consider the following three problems:*

$$\begin{aligned}
&\mathbf{GIO}^{+}_{\mathrm{R}}(\mathcal{D}; K): && \mathbf{GIO}^{-}_{\mathrm{R}}(\mathcal{D}; K): && \mathbf{GIO}^{0}_{\mathrm{R}}(\mathcal{D}; K):\\
\min_{\substack{\mathbf{c},\mathbf{y},\\ \epsilon_1,\ldots,\epsilon_Q}} \quad & \sum_{q=1}^{Q} |\epsilon_q - 1| && \min_{\substack{\mathbf{c},\mathbf{y},\\ \epsilon_1,\ldots,\epsilon_Q}} \quad \sum_{q=1}^{Q} |\epsilon_q - 1| && \min_{\mathbf{c},\mathbf{y}} \quad 0\\
\text{s.t.} \quad & \mathbf{A}^{\mathsf{T}}\mathbf{y} = \mathbf{c},\ \ \mathbf{y} \geq \mathbf{0} && \text{s.t.} \quad \mathbf{A}^{\mathsf{T}}\mathbf{y} = \mathbf{c},\ \ \mathbf{y} \geq \mathbf{0} && \text{s.t.} \quad \mathbf{A}^{\mathsf{T}}\mathbf{y} = \mathbf{c},\ \ \mathbf{y} \geq \mathbf{0}\\
& \mathbf{c}^{\mathsf{T}}\hat{\mathbf{x}}_q = \epsilon_q, \forall q \in \mathcal{Q} && & \mathbf{c}^{\mathsf{T}}\hat{\mathbf{x}}_q = -\epsilon_q, \forall q \in \mathcal{Q} && & \mathbf{c}^{\mathsf{T}}\hat{\mathbf{x}}_q = 0, \forall q \in \mathcal{Q}\\
& \mathbf{b}^{\mathsf{T}}\mathbf{y} = 1 && & \mathbf{b}^{\mathsf{T}}\mathbf{y} = -1 && & \mathbf{b}^{\mathsf{T}}\mathbf{y} = 0, \mathbf{y}^{\mathsf{T}}\mathbf{1} = 1\\
& \|\mathbf{c}\|_N \geq K, && & \|\mathbf{c}\|_N \geq K, && & \|\mathbf{c}\|_N \geq K.
\end{aligned} \tag{3.11}$$

(3.12) (3.13)

Let z^+ *be the optimal value of* $\mathbf{GIO}^{+}_{\mathrm{R}}(\mathcal{D}; K)$ *if it is feasible, otherwise* $z^+ = \infty$. *Let* z^- *and* z^0 *be defined similarly for* $\mathbf{GIO}^{-}_{\mathrm{R}}(\mathcal{D}; K)$ *and* $\mathbf{GIO}^{0}_{\mathrm{R}}(\mathcal{D}; K)$, *respectively. Let* $z^* = \min\{z^+, z^-, z^0\}$ *and let* $\left(\mathbf{c}^*, \mathbf{y}^*, \epsilon_1^*, \ldots, \epsilon_Q^*\right)$ *be an optimal solution for the corresponding problem. We assume* $\epsilon_1^* = \cdots = \epsilon_Q^* = 1$ *for* $\mathbf{GIO}^{0}_{\mathrm{R}}(\mathcal{D}; K)$. *Then there exists* K *such that the optimal value of* $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ *is equal to* z^* *and an optimal solution to* $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ *is* $\left(\mathbf{c}^*/\|\mathbf{c}^*\|_N, \mathbf{y}^*/\|\mathbf{c}^*\|_N, \epsilon_1^*, \ldots, \epsilon_Q^*\right)$.

Proof. Let $(\hat{\mathbf{c}}, \hat{\mathbf{y}})$ be an optimal solution to $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ and let

$$K = \begin{cases} 1/|\mathbf{b}^{\mathsf{T}}\hat{\mathbf{y}}| & \text{if } \mathbf{b}^{\mathsf{T}}\hat{\mathbf{y}} \neq 0 \\ 1/\hat{\mathbf{y}}^{\mathsf{T}}\mathbf{1} & \text{otherwise.} \end{cases} \tag{3.14}$$

We omit the variables $(\epsilon_1, \ldots, \epsilon_Q)$ when writing optimal solutions for conciseness. First, we show that $(\hat{\mathbf{c}}, \hat{\mathbf{y}})$ maps to a corresponding feasible solution for one of $\mathbf{GIO}^{+}_{\mathrm{R}}(\mathcal{D}; K)$, $\mathbf{GIO}^{-}_{\mathrm{R}}(\mathcal{D}; K)$, or $\mathbf{GIO}^{0}_{\mathrm{R}}(\mathcal{D}; K)$ with the same objective value. Conversely, every feasible

solution to formulations (3.11)–(3.13) has a corresponding feasible solution in $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ with the same objective value.

First, suppose $\mathbf{b}^{\mathsf{T}}\hat{\mathbf{y}} > 0$ and consider $(\tilde{\mathbf{c}}, \tilde{\mathbf{y}}) = \left(\hat{\mathbf{c}}/\mathbf{b}^{\mathsf{T}}\hat{\mathbf{y}}, \hat{\mathbf{y}}/\mathbf{b}^{\mathsf{T}}\hat{\mathbf{y}}\right)$. This solution is feasible to $\mathbf{GIO}^{+}_{\mathrm{R}}(\mathcal{D}; K)$ as $\mathbf{b}^{\mathsf{T}}\tilde{\mathbf{y}} = 1$ and $\|\tilde{\mathbf{c}}\|_N = K$. Furthermore, by substituting $\tilde{\mathbf{c}} = \hat{\mathbf{c}}/\mathbf{b}^{\mathsf{T}}\hat{\mathbf{y}}$, we see that the objective value of this solution for $\mathbf{GIO}^{+}_{\mathrm{R}}(\mathcal{D}; K)$ is equal to the optimal value for $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$:

$$\sum_{q=1}^{Q} \left|\tilde{\mathbf{c}}^{\mathsf{T}}\hat{\mathbf{x}}_q - 1\right| = \sum_{q=1}^{Q} \left|\left(\hat{\mathbf{c}}^{\mathsf{T}}\hat{\mathbf{x}}_q\right) / \left(\mathbf{b}^{\mathsf{T}}\hat{\mathbf{y}}\right) - 1\right|.$$

Similarly, when $\mathbf{b}^{\mathsf{T}}\hat{\mathbf{y}} < 0$, we construct $(\tilde{\mathbf{c}}, \tilde{\mathbf{y}}) = \left(\hat{\mathbf{c}}/|\mathbf{b}^{\mathsf{T}}\hat{\mathbf{y}}|, \hat{\mathbf{y}}/|\mathbf{b}^{\mathsf{T}}\hat{\mathbf{y}}|\right)$, which is feasible to $\mathbf{GIO}^{-}_{\mathrm{R}}(\mathcal{D}; K)$ and incurs the same objective value as the optimal value of $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$. Finally, if $\mathbf{b}^{\mathsf{T}}\hat{\mathbf{y}} = 0$, then the optimal value of $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ is 0. Let $(\tilde{\mathbf{c}}, \tilde{\mathbf{y}}) = \left(\hat{\mathbf{c}}/\hat{\mathbf{y}}^{\mathsf{T}}\mathbf{1}, \hat{\mathbf{y}}/\hat{\mathbf{y}}^{\mathsf{T}}\mathbf{1}\right)$. It is straightforward to show that this solution is feasible for $\mathbf{GIO}^{0}_{\mathrm{R}}(\mathcal{D}; K)$. Thus, an optimal solution to $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ can be scaled to construct a solution that is feasible for exactly one of the formulations (3.11)–(3.13).

The converse is proven by showing that every feasible solution of (3.11)–(3.13) can be scaled to a feasible solution of $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$. Let $(\tilde{\mathbf{c}}, \tilde{\mathbf{y}})$ be a feasible solution to one of (3.11)–(3.13), and let $(\hat{\mathbf{c}}, \hat{\mathbf{y}}) = (\tilde{\mathbf{c}}/\left\|\tilde{\mathbf{c}}\right\|_N, \tilde{\mathbf{y}}/\left\|\tilde{\mathbf{c}}\right\|_N)$. This solution is feasible for $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ with the same objective function value.

In terms of objective value, all feasible solutions of $\mathbf{GIO}^{+}_{\mathrm{R}}(\mathcal{D}; K)$, $\mathbf{GIO}^{-}_{\mathrm{R}}(\mathcal{D}; K)$, and $\mathbf{GIO}^{0}_{\mathrm{R}}(\mathcal{D}; K)$ have a one-to-one correspondence with feasible solutions of $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ and the best optimal solution to formulations (3.11)–(3.13) can be scaled to an optimal solution for $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$. □

Proposition 5 breaks $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ into three cases: $\mathbf{b}^{\mathsf{T}}\mathbf{y} > 0$, $\mathbf{b}^{\mathsf{T}}\mathbf{y} < 0$, and $\mathbf{b}^{\mathsf{T}}\mathbf{y} = 0$. We then normalize $\mathbf{b}^{\mathsf{T}}\mathbf{y}$ and alter (3.10d). Note that the ϵ_q terms become free variables when $\mathbf{b}^{\mathsf{T}}\mathbf{y} = \mathbf{0}$, which is why we assume $\epsilon_q = 0$ for $\mathbf{GIO}^{0}_{\mathrm{R}}(\mathcal{D})$.

There are two issues to note. First, Proposition 5 requires the selection of an appropriate value for the parameter K, which can be accomplished by solving an auxiliary problem (see Appendix A.1 for details). Second, formulations (3.11), (3.12), and (3.13) are still non-convex due to the normalization constraint $\|\mathbf{c}\|_N \geq K$. As in $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$, this can be addressed via polyhedral decomposition. For example, if $\|\cdot\|_N = \|\cdot\|_\infty$, $\mathbf{GIO}^{+}_{\mathrm{R}}(\mathcal{D}; K)$

decomposes to $2n$ linear programs $\mathbf{GIO}^{+}_{\mathrm{R}}(\mathcal{D}; K, j)$ (see Theorem 2):

$$\begin{aligned} \mathbf{GIO}^{+}_{\mathrm{R}}(\mathcal{D}; K, j): \quad \underset{\mathbf{c},\mathbf{y},\epsilon_1,\ldots,\epsilon_Q}{\text{minimize}} \quad & \sum_{q=1}^{Q} |\epsilon_q - 1| \\ \text{subject to} \quad & \mathbf{A}^{\mathsf{T}}\mathbf{y} = \mathbf{c}, \;\; \mathbf{y} \geq \mathbf{0} \\ & \mathbf{c}^{\mathsf{T}}\hat{\mathbf{x}}_q = \epsilon_q, \quad \forall q \in \mathcal{Q} \\ & \mathbf{b}^{\mathsf{T}}\mathbf{y} = 1 \\ & (c_j \geq K) \vee (c_j \leq -K)\,. \end{aligned} \tag{3.15}$$

The complete algorithm for solving $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ exactly in this assumption-free setting is provided in Appendix A.1. We briefly remark here that an alternative approach is to relax the normalization constraints in formulations (3.11), (3.12), and (3.13). If solving the relaxations yields an optimal $\mathbf{c}^* \neq \mathbf{0}$, then this cost vector can be re-scaled as in Proposition 5 (see Corollary 5).

Feasible observed decisions. As in the absolute duality gap case, the relative duality gap model reduces to a single-point problem, which has an analytic solution according to Theorem 1.

Proposition 6. *If $\mathcal{D} \subset \mathcal{P}$ and $\bar{\mathbf{x}}$ is the centroid of $\mathcal{D}$, then $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ is equivalent to $\mathbf{GIO}_{\mathrm{R}}(\{\bar{\mathbf{x}}\})$.*

Proof. When all of the observed points are feasible, $\mathbf{c}^{\mathsf{T}}\hat{\mathbf{x}}_q - \mathbf{b}^{\mathsf{T}}\mathbf{y} \geq 0$, $\forall q \in \mathcal{Q}$. Thus, objective (3.10a) becomes

$$\sum_{q=1}^{Q} |\epsilon_q - 1| = \sum_{q=1}^{Q} \frac{\mathbf{c}^{\mathsf{T}}\hat{\mathbf{x}}_q - \mathbf{b}^{\mathsf{T}}\mathbf{y}}{|\mathbf{b}^{\mathsf{T}}\mathbf{y}|} = Q\left(\frac{\mathbf{c}^{\mathsf{T}}\bar{\mathbf{x}} - \mathbf{b}^{\mathsf{T}}\mathbf{y}}{|\mathbf{b}^{\mathsf{T}}\mathbf{y}|}\right).$$

Noting that $\bar{\mathbf{x}}$ must also be feasible, the last term equals the objective for $\mathbf{GIO}_{\mathrm{R}}(\{\bar{\mathbf{x}}\})$. □

Infeasible observed decisions. Proposition 7 below is analogous to Proposition 3, and provides an analytic solution for $\mathbf{GIO}_{\mathrm{R}}(\{\hat{\mathbf{x}}\})$ if $\hat{\mathbf{x}} \notin \mathcal{P}$. Corollary 2 extends Proposition 7 to multiple infeasible decisions similar to Corollary 1 extending Proposition 3. The proofs (omitted) are similar to before.

Proposition 7. *Assume $\hat{\mathbf{x}} \notin \mathcal{P}$.*

1. *If $\hat{\mathbf{x}}$ satisfies $\mathbf{a}_i^{\mathsf{T}}\hat{\mathbf{x}} > b_i$ for some $i \in \mathcal{I}$, then there exists $i^* \in \mathcal{I}$ such that (3.7) is an optimal solution to $\mathbf{GIO}_{\mathrm{R}}(\{\hat{\mathbf{x}}\})$ and the optimal value is 0.*

2. If $\mathbf{A}\hat{\mathbf{x}} \leq \mathbf{b}$*, there exists* $i^* \in \mathcal{I}$ *such that* (3.3) *is an optimal solution to* $\mathbf{GIO}_{\mathrm{R}}(\{\hat{\mathbf{x}}\})$.

Corollary 2. *Suppose that* $\mathbf{A}\hat{\mathbf{x}}_q \leq \mathbf{b}$ *for all* $q \in \mathcal{Q}$ *and let* $\bar{\mathbf{x}}$ *be the centroid of* $\mathcal{D}$*. Then,* $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ *for the forward problem* $\mathbf{FO}(\mathbf{c})$ *is equivalent to* $\mathbf{GIO}_{\mathrm{R}}(\{\bar{\mathbf{x}}\})$ *for* $\mathbf{FOA}(\mathbf{c})$.

3.2.2 Decision space

Inverse optimization in the decision space measures error by distance from optimal decisions, rather than objective values. The model identifies a cost vector that produces optimal decisions for the forward problem that are of minimum aggregate distance to the corresponding observed decisions:

$$\begin{aligned}
\mathbf{GIO}_p(\mathcal{D}): \quad \underset{\mathbf{c},\mathbf{y},\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_Q}{\text{minimize}} \quad & \sum_{q=1}^{Q} \|\boldsymbol{\epsilon}_q\|_p && \text{(3.16a)}\\
\text{subject to} \quad & \mathbf{A}^{\mathsf{T}}\mathbf{y} = \mathbf{c}, \;\; \mathbf{y} \geq \mathbf{0} && \text{(3.16b)}\\
& \mathbf{c}^{\mathsf{T}}\hat{\mathbf{x}}_q = \mathbf{b}^{\mathsf{T}}\mathbf{y} + \mathbf{c}^{\mathsf{T}}\boldsymbol{\epsilon}_q, \quad \forall q \in \mathcal{Q} && \text{(3.16c)}\\
& \mathbf{A}\left(\hat{\mathbf{x}}_q - \boldsymbol{\epsilon}_q\right) \geq \mathbf{b}, \quad \forall q \in \mathcal{Q} && \text{(3.16d)}\\
& \|\mathbf{c}\|_N = 1. && \text{(3.16e)}
\end{aligned}$$

$\mathbf{GIO}_p(\mathcal{D})$ resembles $\mathbf{GIO}(\mathcal{D})$, except that the objective function is the sum of p-norms ($p \geq 1$) and constraint (3.16d) is added to enforce primal feasibility of the perturbed decisions $\hat{\mathbf{x}}_q - \boldsymbol{\epsilon}_q$. Unlike in the objective space models, we require primal feasibility because the $\boldsymbol{\epsilon}_q$ perturbation vectors have a physical meaning as the distance from observed $\hat{\mathbf{x}}_q$ to optimal $\mathbf{x}_q^*$ decisions. It is straightforward to show $\mathbf{GIO}_p(\mathcal{D})$ is a specialization of $\mathbf{GIO}(\mathcal{D})$ (proof omitted).

Proposition 8. *A solution* $\left(\mathbf{c}^*, \mathbf{y}^*, \boldsymbol{\epsilon}_1^*, \ldots, \boldsymbol{\epsilon}_Q^*\right)$ *is optimal to* $\mathbf{GIO}_p(\mathcal{D})$ *if and only if it is optimal to* $\mathbf{GIO}(\mathcal{D})$ *with the hyperparameters* $(\|\cdot\|, \|\cdot\|_N, \mathcal{C}, \mathcal{E}_1, \ldots, \mathcal{E}_Q)$ *equal to*

$$\left(\|\cdot\|_p, \|\cdot\|_N, \mathbb{R}^n, \{\boldsymbol{\epsilon}_1 \mid \mathbf{A}\left(\hat{\mathbf{x}}_1 - \boldsymbol{\epsilon}_1\right) \geq \mathbf{b}\}, \ldots, \{\boldsymbol{\epsilon}_Q \mid \mathbf{A}\left(\hat{\mathbf{x}}_Q - \boldsymbol{\epsilon}_Q\right) \geq \mathbf{b}\}\right).$$

Although $\mathbf{GIO}_p(\mathcal{D})$ is non-convex, we show that an optimal cost vector coincides with one of the constraints (e.g., Theorem 1). However, directly projecting all $\hat{\mathbf{x}}_q$ to a hyperplane may result in projections being infeasible, violating (3.16d). Thus, we define

the *feasible projection problem*:

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \|\hat{\mathbf{x}}_q - \mathbf{x}\|_p \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} \geq \mathbf{b} \\ & \mathbf{a}_i^\mathsf{T}\mathbf{x} = b_i. \end{aligned} \tag{3.17}$$

Let $\psi_i(\hat{\mathbf{x}}_q)$ be an optimal solution to problem (3.17), which identifies the closest point in $\mathcal{P}$ to $\hat{\mathbf{x}}_q$ on the hyperplane $\mathcal{H}_i = \left\{\mathbf{x} \mid \mathbf{a}_i^\mathsf{T}\mathbf{x} = b_i\right\}$. We first derive a structured optimal solution to $\mathbf{GIO}_p(\mathcal{D})$.

Lemma 1. *There exists $i \in \mathcal{I}$ such that an optimal solution to $\mathbf{GIO}_p(\mathcal{D})$ is given by*

$$\left(\mathbf{c}^*, \mathbf{y}^*, \boldsymbol{\epsilon}_1^*, \ldots, \boldsymbol{\epsilon}_Q^*\right) = \left(\frac{\mathbf{a}_i}{\|\mathbf{a}_i\|_N}, \frac{\mathbf{e}_i}{\|\mathbf{a}_i\|_N}, \hat{\mathbf{x}}_1 - \psi_i(\hat{\mathbf{x}}_1), \ldots, \hat{\mathbf{x}}_Q - \psi_i(\hat{\mathbf{x}}_Q)\right). \tag{3.18}$$

Proof. Without loss of generality, assume that $\|\mathbf{a}_i\|_N = 1$ for all $i \in \mathcal{I}$. Solution (3.18) is feasible to $\mathbf{GIO}_p(\mathcal{D})$ for all $i \in \mathcal{I}$. We show that for any feasible solution that is not of the form (3.18), there exists a feasible solution of that form whose objective value is at least as good.

Consider a feasible solution $(\tilde{\mathbf{c}}, \tilde{\mathbf{y}}, \tilde{\boldsymbol{\epsilon}}_1, \ldots, \tilde{\boldsymbol{\epsilon}}_Q)$ to $\mathbf{GIO}_p(\mathcal{D})$, where $\tilde{\mathbf{y}} \neq \mathbf{e}_i$ for any $i \in \mathcal{I}$. Without loss of generality, assume $\tilde{y}_1, \ldots, \tilde{y}_k > 0$ for some $1 < k \leq m$ and let $\mathcal{K} = \{1, \ldots, k\}$ denote the corresponding index set. Let $\tilde{\mathbf{x}}_q = \hat{\mathbf{x}}_q - \tilde{\boldsymbol{\epsilon}}_q$ denote the perturbed decision for all $q \in \mathcal{Q}$. The primal feasibility constraint (3.16d) implies that $\mathbf{A}\tilde{\mathbf{x}}_q \geq \mathbf{b}$ for all $q \in \mathcal{Q}$. The strong duality constraint (3.16c) implies for all $q \in \mathcal{Q}$, that

$$0 = \mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}_q - \mathbf{b}^\mathsf{T}\tilde{\mathbf{y}} = \sum_{i=1}^{k} \tilde{y}_i \left(\mathbf{a}_i^\mathsf{T}\tilde{\mathbf{x}}_q - b_i\right),$$

which follows from substituting $\tilde{\mathbf{c}} = \sum_{i=1}^{k} \tilde{y}_i \mathbf{a}_i$. Using the non-negativity of $\tilde{\mathbf{y}}$ and primal feasibility (i.e., $\mathbf{a}_i^\mathsf{T}\tilde{\mathbf{x}}_q \geq b_i$ for all $i \in \mathcal{I}$), we see that $\tilde{\mathbf{x}}_q$ for all $q \in \mathcal{Q}$ are feasible solutions to the feasible projection problem (3.17) for each $i \in \mathcal{K}$.

Let $(\hat{\mathbf{c}}, \hat{\mathbf{y}}, \hat{\boldsymbol{\epsilon}}_1, \ldots, \hat{\boldsymbol{\epsilon}}_Q) = (\mathbf{a}_{i^*}, \mathbf{e}_{i^*}, \hat{\mathbf{x}}_1 - \psi_{i^*}(\hat{\mathbf{x}}_1), \ldots, \hat{\mathbf{x}}_Q - \psi_{i^*}(\hat{\mathbf{x}}_Q))$ for an arbitrary index $i^* \in \mathcal{K}$. For all $q \in \mathcal{Q}$, $\psi_{i^*}(\hat{\mathbf{x}}_q)$ is, by definition, an optimal solution to (3.17). Therefore, we have

$$\sum_{q=1}^{Q} \|\hat{\boldsymbol{\epsilon}}_q\|_p = \sum_{q=1}^{Q} \|\hat{\mathbf{x}}_q - \psi_{i^*}(\hat{\mathbf{x}}_q)\|_p \leq \sum_{q=1}^{Q} \|\tilde{\boldsymbol{\epsilon}}_q\|_p,$$

with the inequality following from the optimality of (3.17). Thus, given any feasible

solution to $\mathbf{GIO}_p(\mathcal{D})$ not of the form defined in (3.18), we can construct a feasible solution of the form (3.18) with the objective value at least as good as the original. □

The intuition behind Lemma 1 is as follows. Given a feasible set of vectors $\boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_Q$, every observed decision $\hat{\mathbf{x}}_q$ is perturbed by $\boldsymbol{\epsilon}_q$ to a point that satisfies both strong duality and primal feasibility. Strong duality implies that $\mathcal{H} = \left\{\mathbf{x} \mid \mathbf{c}^{*\mathsf{T}}\mathbf{x} = \mathbf{b}^{\mathsf{T}}\mathbf{y}^*\right\}$ is a supporting hyperplane, and so $\hat{\mathbf{x}}_q - \boldsymbol{\epsilon}_q$ lies on that supporting hyperplane for all $q \in \mathcal{Q}$. Every feasible solution not of the form (3.18) must satisfy multiple constraints with equality, and is dominated by solutions that involve the feasible projection to just one of those constraints. Since Lemma 1 holds regardless of the chosen norm and feasibility of the observed decisions, we can show $\mathbf{GIO}_p(\mathcal{D})$ can be solved via m convex optimization problems (which become linear with appropriate p-norms).

Theorem 3. *Consider the following optimization problem:*

$$\min_{i\in\mathcal{I}} \quad \min_{\boldsymbol{\epsilon}_{1,i},\ldots,\boldsymbol{\epsilon}_{Q,i}} \quad \sum_{q=1}^{Q} \left\|\boldsymbol{\epsilon}_{q,i}\right\|_p \tag{3.19a}$$

$$\text{s. t.} \quad \mathbf{A}\left(\hat{\mathbf{x}}_q - \boldsymbol{\epsilon}_{q,i}\right) \geq \mathbf{b}, \quad \forall q \in \mathcal{Q} \tag{3.19b}$$

$$\mathbf{a}_i^{\mathsf{T}}(\hat{\mathbf{x}}_q - \boldsymbol{\epsilon}_{q,i}) = b_i, \quad \forall q \in \mathcal{Q}. \tag{3.19c}$$

For each $i \in \mathcal{I}$, let $(\boldsymbol{\epsilon}^*_{1,i}, \ldots, \boldsymbol{\epsilon}^*_{Q,i})$ denote an optimal solution to the inner optimization problem and let $i^* \in \arg\min_{i\in\mathcal{I}} \sum_{q=1}^{Q} \left\|\boldsymbol{\epsilon}^*_{q,i}\right\|$ denote an optimal index determined by the outer optimization problem. Then, $\left(\mathbf{a}_{i^*}/\left\|\mathbf{a}_{i^*}\right\|_N, \mathbf{e}_{i^*}/\left\|\mathbf{a}_{i^*}\right\|_N, \boldsymbol{\epsilon}^*_{1,i^*}, \ldots, \boldsymbol{\epsilon}^*_{Q,i^*}\right)$ is an optimal solution to $\mathbf{GIO}_p(\mathcal{D})$.

Proof. For each i, the inner optimization problem produces solutions with the structure in (3.18). Thus, the inner optimization problems, along with the corresponding $(\mathbf{c}, \mathbf{y})$ enumerate all possible solutions to $\mathbf{GIO}_p(\mathcal{D})$ with the structure in (3.18). By Lemma 1, we select the one yielding the lowest objective value. □

3.2.3 Summary of models and comparison with literature

Table 3.1 summarizes the model variants of $\mathbf{GIO}(\mathcal{D})$. Next, we relate and bound the optimal values of the three variants.

Theorem 4. *Assume $\mathcal{D} \subset \mathcal{P}$ and let z^*_{A} and z^*_p denote the optimal values of $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$ and $\mathbf{GIO}_p(\mathcal{D})$, respectively. Then $z^*_p \geq z^*_{\mathrm{A}}$.*

Proof. First note that due to the dominance between p-norms, (i.e., $\|\boldsymbol{\epsilon}\|_p \geq \|\boldsymbol{\epsilon}\|_\infty$) we have $z^*_p \geq z^*_\infty$, since the choice of p only affects the objective and the two problems

share the same feasible set. We then lower bound the optimal value of $\mathbf{GIO}_\infty(\mathcal{D})$ using Theorem 3:

$$\left.\begin{array}{ll} \min\limits_{i\in\mathcal{I}} \min\limits_{\boldsymbol{\epsilon}_{1,i},\ldots,\boldsymbol{\epsilon}_{Q,i}} & \sum\limits_{q=1}^{Q} \left\|\boldsymbol{\epsilon}_{q,i}\right\|_\infty \\ \text{s.t.} & \mathbf{A}\left(\hat{\mathbf{x}}_q - \boldsymbol{\epsilon}_{q,i}\right) \geq \mathbf{b}_i, \forall q \in \mathcal{Q} \\ & \mathbf{a}_i^\intercal\left(\hat{\mathbf{x}}_q - \boldsymbol{\epsilon}_{q,i}\right) = b_i, \forall q \in \mathcal{Q} \end{array}\right\} = \min_{i\in\mathcal{I}} \left\{ \sum_{q=1}^{Q} \left\|\hat{\mathbf{x}}_q - \psi_i(\hat{\mathbf{x}}_q)\right\|_\infty \right\} \tag{3.20}$$

$$\geq \min_{i\in\mathcal{I}} \left\{ \sum_{q=1}^{Q} \left\|\hat{\mathbf{x}}_q - \pi_i(\hat{\mathbf{x}}_q)\right\|_\infty \right\} \tag{3.21}$$

$$= \min_{i\in\mathcal{I}} \left\{ \sum_{q=1}^{Q} \frac{\left|\mathbf{a}_i^\intercal \hat{\mathbf{x}}_q - b_i\right|}{\left\|\mathbf{a}_i\right\|_1} \right\} \tag{3.22}$$

$$= \min_{i\in\mathcal{I}} \left\{ Q\left(\frac{\mathbf{a}_i^\intercal \bar{\mathbf{x}} - b_i}{\left\|\mathbf{a}_i\right\|_1} \right) \right\}. \tag{3.23}$$

The inequality in (3.21) comes from the fact that the projection problem (3.2) is a relaxation of the feasible projection problem (3.17), by removing the feasibility constraint. The equality of (3.22) comes from Mangasarian (1999) (e.g., see Theorem 1), which provides the analytic optimal value of the projection problem. Because $\hat{\mathbf{x}}_q \in \mathcal{P}$ for all $\hat{\mathbf{x}}_q \in \mathcal{D}$, we bypass the absolute values to average. Note that (3.23) is equal to the optimal value of $\mathbf{GIO}(\{\bar{\mathbf{x}}\})$.

Now consider $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$. Because, $\mathcal{D} \subset \mathcal{P}$, Proposition 2 yields $z_{\mathrm{A}}^* = z^*\big(\mathbf{GIO}(\{\bar{\mathbf{x}}\})\big)$, i.e., the optimal solution to $\mathbf{GIO}(\{\bar{\mathbf{x}}\})$ where $\bar{\mathbf{x}} = \sum_{q=1}^{Q} \hat{\mathbf{x}}_q / Q$ is the centroid of $\mathcal{D}$. In conjunction with (3.23), we conclude that $z_p^* \geq z_\infty^* \geq z_{\mathrm{A}}^*$. □

Theorem 4 implies that if the decision space model returns a low error, so does the absolute duality gap model. Note that although bounds between objective and decision space inverse convex optimization models exist (Theorem 1 in Bertsimas et al. (2015) and Proposition 2.5 in Esfahani et al. (2018)), the previous bounds were developed using constants based on the non-linearity of the objective function of the forward problem (e.g., Bertsimas et al. (2015) assumes the gradient of the objective is strongly monotone), which are not applicable in our linear setting. Furthermore, due to the nature of relative versus absolute measures, we can also bound the performance of the absolute and relative duality gap models, and consequently connect all three variants.

Corollary 3. *Let z_{A}^* and z_{R}^* denote the optimal values of $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$ and $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$, respectively. Let f_{A}^* and f_{R}^* be the optimal values of the forward problem $\mathbf{FO}(\mathbf{c})$ using cost vectors obtained by $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$ and $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$, respectively. Then, $|f_{\mathrm{R}}^*|\ z_{\mathrm{R}}^* \geq z_{\mathrm{A}}^* \geq |f_{\mathrm{A}}^*|\ z_{\mathrm{R}}^*$.*

Proof. We remark that Corollary 3 is in fact a special case of a more general statement regarding error measures in the absolute versus relative space. Below, we prove a more general statement and specialize the result to the case of inverse optimization.

Let $f(\mathbf{x})$ and $g(\mathbf{x})$ be two functions and $f(\mathbf{x}) \neq \mathbf{0}$ for all $\mathbf{x}$. Consider two optimization problems:

$$\min_{\mathbf{x}} \quad \sum_{q=1}^{Q} |g_q(\mathbf{x}) - f(\mathbf{x})| \qquad \text{s.t.} \quad \mathbf{x} \in \mathcal{X} \tag{3.24}$$

$$\min_{\mathbf{x}} \quad \sum_{q=1}^{Q} \left| \frac{g_q(\mathbf{x}) - f(\mathbf{x})}{f(\mathbf{x})} \right| \qquad \text{s.t.} \quad \mathbf{x} \in \mathcal{X} \tag{3.25}$$

Let $\mathbf{x}_{\mathrm{A}}^*$ and z_{A}^* be an optimal solution and value, respectively for (3.24). Similarly, let $\mathbf{x}_{\mathrm{R}}^*$ and z_{R}^* be an optimal solution and value, respectively for (3.25). We will prove that $|f(\mathbf{x}_{\mathrm{R}}^*)| z_{\mathrm{R}}^* \geq z_{\mathrm{A}}^* \geq z_{\mathrm{R}}^* |f(\mathbf{x}_{\mathrm{A}}^*)|$.

First note that $\mathbf{x}_{\mathrm{A}}^*$ is feasible for (3.25) and $\mathbf{x}_{\mathrm{R}}^*$ is feasible for (3.24). Then,

$$\begin{aligned} z_{\mathrm{A}}^* &= \sum_{q=1}^{Q} |g_q(\mathbf{x}_{\mathrm{A}}^*) - f(\mathbf{x}_{\mathrm{A}}^*)| \\ &\leq \sum_{q=1}^{Q} |g_q(\mathbf{x}_{\mathrm{R}}^*) - f(\mathbf{x}_{\mathrm{R}}^*)| \\ &= \sum_{q=1}^{Q} \left| \frac{g_q(\mathbf{x}_{\mathrm{R}}^*) - f(\mathbf{x}_{\mathrm{R}}^*)}{f(\mathbf{x}_{\mathrm{R}}^*)} \right| |f(\mathbf{x}_{\mathrm{R}}^*)| \\ &= z_{\mathrm{R}}^* |f(\mathbf{x}_{\mathrm{R}})|. \end{aligned}$$

The inequality comes from the feasibility of $\mathbf{x}_{\mathrm{R}}^*$ for (3.24) and the second equality comes from multiplying by $|f(\mathbf{x}_{\mathrm{R}}^*)|/|f(\mathbf{x}_{\mathrm{R}}^*)|$. This proves the left inequality in the Corollary statement.

We next show

$$\begin{aligned} z_{\mathrm{R}}^* &= \sum_{q=1}^{Q} \left| \frac{g_q(\mathbf{x}_{\mathrm{R}}^*) - f(\mathbf{x}_{\mathrm{R}}^*)}{f(\mathbf{x}_{\mathrm{R}}^*)} \right| \\ &\leq \sum_{q=1}^{Q} \left| \frac{g_q(\mathbf{x}_{\mathrm{A}}^*) - f(\mathbf{x}_{\mathrm{A}}^*)}{f(\mathbf{x}_{\mathrm{A}}^*)} \right| \\ &= \sum_{q=1}^{Q} |g_q(\mathbf{x}_{\mathrm{A}}^*) - f(\mathbf{x}_{\mathrm{A}}^*)| \frac{1}{|f(\mathbf{x}_{\mathrm{A}}^*)|} \\ &= z_{\mathrm{A}}^* \frac{1}{|f(\mathbf{x}_{\mathrm{A}}^*)|}. \end{aligned}$$

Table 3.1: Summary of the different variants of $\mathbf{GIO}(\mathcal{D})$.

	$\|\cdot\|$	$\|\cdot\|_N$	$\mathcal{C}$	$\mathcal{E}_q, \forall q \in \mathcal{Q}$	Solution approach
$\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$	$\|\cdot\|_\infty$	$\|\cdot\|_N$	$\mathbb{R}^n$	$\{\boldsymbol{\epsilon}_q \mid \boldsymbol{\epsilon}_q = \epsilon_q \mu(\mathbf{c})\}$	Polyhedral decomposition
$\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$	$\|\cdot\|_\infty / \lvert\mathbf{b}^\mathsf{T}\mathbf{y}\rvert$	$\|\cdot\|_N$	$\mathbb{R}^n$	$\{\boldsymbol{\epsilon}_q \mid \boldsymbol{\epsilon}_q = \mathbf{b}^\mathsf{T}\mathbf{y}\,(\epsilon_q - 1)\,\mu(\mathbf{c})\}$	Three sub-problems
$\mathbf{GIO}_p(\mathcal{D})$	$\|\cdot\|_p$	$\|\cdot\|_N$	$\mathbb{R}^n$	$\{\boldsymbol{\epsilon}_q \mid \mathbf{A}\,(\mathbf{x}_q - \boldsymbol{\epsilon}_q) \geq \mathbf{b}\}$	Formulation (3.19)

The inequality comes from the feasibility of $\mathbf{x}^*_{\mathrm{A}}$ for (3.25). This proves the right inequality in the Corollary statement.

Finally, we observe that letting $\mathbf{x} = (\mathbf{c}, \mathbf{y})$, $\mathcal{X} = \{(\mathbf{c}, \mathbf{y}) \mid \mathbf{A}^\mathsf{T}\mathbf{y} = \mathbf{c}, \mathbf{y} \geq \mathbf{0}, \|\mathbf{c}\|_N = 1$, $g_q(\mathbf{x}) = \mathbf{c}^\mathsf{T}\hat{\mathbf{x}}_q$, and $f(\mathbf{x}) = \mathbf{b}^\mathsf{T}\mathbf{y}$ converts (3.24) into $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$ and (3.25) into $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$. Finally note that for any feasible pair $(\mathbf{c}, \mathbf{y})$, $\mathbf{b}^\mathsf{T}\mathbf{y}$ is equal to the optimal value of the forward problem $\mathbf{FO}(\mathbf{c})$. Substituting the terms for the absolute and relative duality gap problems respectively completes the inequality. □

Next, we briefly compare our models with similar models from the literature. AN in-depth technical comparison is provided in Appendix A.2. $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$ and $\mathbf{GIO}_p(\mathcal{D})$ can be seen as special cases of previous inverse convex optimization models (Aswani et al., 2018; Bertsimas et al., 2015; Esfahani et al., 2018). There, the forward problem is $\min_{\mathbf{x}}\{f(\mathbf{x}; \mathbf{u}, \mathbf{c}) \mid g(\mathbf{x}; \mathbf{u}, \mathbf{c}) \leq \mathbf{0}\}$, where $f(\mathbf{x}; \mathbf{u}, \mathbf{c})$ and $g(\mathbf{x}; \mathbf{u}, \mathbf{c})$ are convex differentiable functions and $\mathbf{u}$ is an exogenous instance-specific parameter. Thus, the data set in their setting is $\mathcal{D} = \{(\hat{\mathbf{x}}_1, \hat{\mathbf{u}}_1), \ldots, (\hat{\mathbf{x}}_Q, \hat{\mathbf{u}}_Q)\}$. Here, we remove $\mathbf{u}$ and set $f(\mathbf{x}; \mathbf{c}) = \mathbf{c}^\mathsf{T}\mathbf{x}$ and $g(\mathbf{x}; \mathbf{c}) = \mathbf{b} - \mathbf{A}\mathbf{x}$ to obtain a linear forward problem with a fixed feasible set.

While the assumption of instance-specific parameters generalize our setting, we observe that the consequent formulations and methods are on the whole, less efficient than those presented in our chapter. The implication of different forward models is the need for additional dual variables and dual feasibility constraints for each feasible set. For a large-scale forward optimization problem, the additional variables and constraints required to formulate the inverse problem grows both in the number of feasible sets and the size of $\mathcal{D}$. For example in our application, n (dimension of the decision vector) and m (number of constraints) for the forward problem are on the order of 10^5. Inverse optimization frameworks from the literature (which impute instance-specific parameters) lead to inverse problems that grow significantly with every data point. In contrast, our ensemble approach using a single forward model does not suffer from this curse.

Bertsimas et al. (2015) study inverse optimization by minimizing a first-order variational inequality (which reduces to the absolute duality gap in linear programs) and construct a convex inverse problem without a normalization constraint (e.g., $\|\mathbf{c}\|_N = 1$). Although normalization can be avoided with a carefully chosen $\mathcal{C}$, setting $f(\mathbf{x}; \mathbf{u}, \mathbf{c}) = \mathbf{c}^\mathsf{T}\mathbf{x}$

with a general $\mathcal{C} = \mathbb{R}^n$ implies that $(\mathbf{c}, \mathbf{y}, \epsilon_1, \ldots, \epsilon_Q) = (\mathbf{0}, \mathbf{0}, 0, \ldots, 0)$ is trivially optimal.

Esfahani et al. (2018) study distributionally robust inverse convex optimization problem, which can specialize to absolute duality gap inverse linear optimization with a normalization constraint. Their formulation decomposes to a finite set of conic optimization problems after polyhedral decomposition. While their approach specializes to ours in the non-robust case, we further analyze several other special cases that yield efficient solution methods (e.g., Propositions 2 and 3, and Corollary 1).

Aswani et al. (2018) propose a decision space inverse convex optimization model that satisfies a statistical consistency property given several identifiability conditions that assume the data set of decisions are noisy perturbations of optimal solutions to different forward problems. However, these assumptions may not hold in general, e.g., if they arrive from an ensemble of independent prediction models as in our application (see Appendix A.2.2 for details). Furthermore, our solution method reformulates $\mathbf{GIO}_p(\mathcal{D})$ to m convex problems. In contrast, Aswani et al. (2018) enumeratively solve the inverse problem using fixed $\mathbf{c}$ from a quantized subset of $\mathcal{C}$. They state that their algorithm is practical only when the parameter space $\mathcal{C}$ is modest (i.e., at most four or five parameters). However for $\mathbf{GIO}_p(\mathcal{D})$, we assume $\mathcal{C} = \mathbb{R}^n$ and our algorithm for $\mathbf{GIO}_p(\mathcal{D})$ is insensitive to n.

Finally, we remark that the relative duality gap variant has not been studied in inverse convex optimization. It has been studied in inverse linear optimization but only when given a single feasible decision (Chan et al., 2019). Our case study in Section 3.4 demonstrates the value of $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$.

3.3 Measuring goodness of fit

In this section, we present a unified view of measuring model-data fitness by developing a metric that is easily and consistently interpretable across different inverse linear optimization methods, forward models, and applications. As shown in Example 2 below, assessing the aggregate error may not provide a complete picture of model fitness, necessitating a context-free fitness metric.

Previously proposed fitness measures for inverse optimization exist but are less general (e.g., Chow and Recker (2012); Troutt et al. (2006) for application-specific measures or Chan et al. (2019) for a metric applicable to only a single feasible decision). Our new metric builds off the latter metric, referred to as the *coefficient of complementarity* and

denoted $\rho(\{\hat{\mathbf{x}}\})$:

$$\rho(\{\hat{\mathbf{x}}\}) = 1 - \frac{\|\boldsymbol{\epsilon}^*\|}{\frac{1}{m}\sum_{i=1}^{m}\|\boldsymbol{\epsilon}_i\|}.$$

Analogous to the *coefficient of determination* R^2 in linear regression, $\rho(\{\hat{\mathbf{x}}\})$ provides a scale-free, unitless measure of goodness of fit. The numerator is the residual error from the estimated cost vector, equal to the optimal value of $\mathbf{GIO}(\{\hat{\mathbf{x}}\})$. The denominator is the average of the errors corresponding to the projections of $\hat{\mathbf{x}}$ to each of the m constraints defining the forward feasible region (i.e., $\boldsymbol{\epsilon}_i = \hat{\mathbf{x}} - \pi_i(\hat{\mathbf{x}})$ for $i \in \mathcal{I}$). Just as R^2 calculates the ratio of error of a linear regression model over a baseline mean-only model, $\rho(\{\hat{\mathbf{x}}\})$ measures the relative improvement in error from using $\mathbf{FO}(\mathbf{c}^*)$ compared to a baseline of the average error induced by m candidate cost vectors.

We now generalize $\rho(\{\hat{\mathbf{x}}\})$ for $\mathbf{GIO}(\mathcal{D})$. For convenience, we omit the data set and denote the absolute duality gap, relative duality gap, and p-norm variants of ρ as ρ_{A}, ρ_{R}, and ρ_p, respectively.

3.3.1 Ensemble coefficient of complementarity

We define the *(ensemble) coefficient of complementarity*, $\rho(\mathcal{D})$, as

$$\rho(\mathcal{D}) = 1 - \frac{\sum_{q=1}^{Q}\left\|\boldsymbol{\epsilon}_q^*\right\|}{\frac{1}{m}\sum_{i=1}^{m}\left(\sum_{q=1}^{Q}\|\boldsymbol{\epsilon}_{q,i}\|\right)}. \tag{3.26}$$

The numerator is the optimal value of $\mathbf{GIO}(\mathcal{D})$, i.e., the residual error from an optimal solution to the inverse optimization problem. The denominator terms $\sum_{q=1}^{Q}\|\boldsymbol{\epsilon}_{q,i}\|$ represent the aggregate error induced by choosing baseline feasible solutions $(\mathbf{c}, \mathbf{y}) = (\mathbf{a}_i/\|\mathbf{a}_i\|_N, \mathbf{e}_i/\|\mathbf{a}_i\|_N)$:

- For absolute duality gap, $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$,

$$\sum_{q=1}^{Q}\|\boldsymbol{\epsilon}_{q,i}\| = \sum_{q=1}^{Q}\frac{\left|\mathbf{a}_i^{\mathsf{T}}\hat{\mathbf{x}}_q - b_i\right|}{\|\mathbf{a}_i\|_1}. \tag{3.27}$$

- For relative duality gap, $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$, under the assumption that $b_i \neq 0$ for all $i \in \mathcal{I}$,

$$\sum_{q=1}^{Q}\|\boldsymbol{\epsilon}_{q,i}\| = \sum_{q=1}^{Q}\left|\frac{\mathbf{a}_i^{\mathsf{T}}\hat{\mathbf{x}}_q}{b_i} - 1\right|. \tag{3.28}$$

- For decision space, $\mathbf{GIO}_p(\mathcal{D})$, $\sum_{q=1}^{Q} \|\boldsymbol{\epsilon}_{q,i}\|$ are the optimal values of the inner problems in (3.19).

Our choice of baseline (denominator) is a direct extension from the single-point case, where an optimal cost vector can be found by selecting amongst one of the vectors $\mathbf{a}_i$ defining the m constraints. We maintain this choice of baseline for several reasons. First, an optimal solution will be exactly one of the $\mathbf{a}_i$ in the general decision space problem (see Lemma 1) and in several special cases of the objective space problem (see Propositions 2 and 6). Second, calculation of the denominator is straightforward either directly from the data (e.g., (3.27) and (3.28)) or via the solution of m convex optimization problems (3.19). Third, this definition directly generalizes the single-point metric, inheriting several attractive mathematical properties that we present in Section 3.3.2. Finally, given Propositions 2 and 6, the ensemble coefficient of complementarity is equal to the single-point version for objective space models when all data points are feasible.

Proposition 9. *Let $\bar{\mathbf{x}}$ be the centroid of $\mathcal{D} \subset \mathcal{P}$. Then, $\rho_{\mathrm{A}}(\mathcal{D}) = \rho_{\mathrm{A}}(\{\bar{\mathbf{x}}\})$ and $\rho_{\mathrm{R}}(\mathcal{D}) = \rho_{\mathrm{R}}(\{\bar{\mathbf{x}}\})$.*

The proof (ommitted) follows from Propositions 2 and 6, and algebraic manipulation.

3.3.2 Properties of ρ

Theorem 5. *The following properties hold for ρ defined in* (3.26)*:*

1. ***Optimality:*** *ρ is maximized by an optimal solution to $\mathbf{GIO}(\mathcal{D})$.*
2. ***Boundedness:*** *$\rho \in [0, 1]$.*
3. ***Monotonicity:*** *For $1 \leq k < n$, let $\mathbf{GIO}^{(k)}(\mathcal{D})$ be $\mathbf{GIO}(\mathcal{D})$ with additional constraints $c_i = 0$, for $k+1 \leq i \leq n$ and let $\rho^{(k)}$ be the coefficient of complementarity. Then, $\rho^{(k)} \leq \rho^{(k+1)}$.*

Proof.

1. Given $\mathcal{D}$, $\mathbf{A}$, and $\mathbf{b}$, the denominator term in ρ is fixed. An optimal solution to $\mathbf{GIO}(\mathcal{D})$ minimizes the numerator of $1 - \rho$, thus maximizing ρ.
2. We prove $1 - \rho \in [0, 1]$. It is easy to see that $1 - \rho \geq 0$, because it is the ratio of sums of norms, which are nonnegative. To show $1 - \rho \leq 1$, note that $\sum_{q=1}^{Q} \left\|\boldsymbol{\epsilon}_q^*\right\| \leq \sum_{q=1}^{Q} \|\boldsymbol{\epsilon}_{q,i}\|$ for all i, as setting $\mathbf{c} = \mathbf{a}_i / \|\mathbf{a}_i\|_N$ will yield a feasible but not necessarily optimal solution to $\mathbf{GIO}(\mathcal{D})$.

3. An optimal solution to $\mathbf{GIO}^{(k)}(\mathcal{D})$ is feasible for $\mathbf{GIO}^{(k+1)}(\mathcal{D})$, since the latter problem is a relaxation of the former. Invoking the first statement in this theorem, $\rho^{(k)} \leq \rho^{(k+1)}$.

□

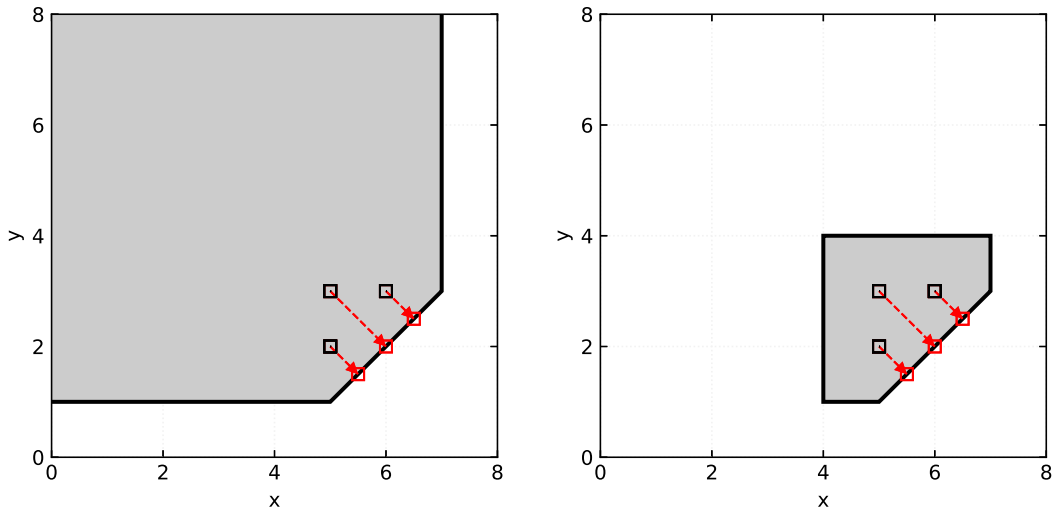
These properties are analogous to the properties of R^2. The first property underlines how ρ integrates into $\mathbf{GIO}(\mathcal{D})$. Although one can select any cost vector and calculate the ρ value with respect to the data $\mathcal{D}$, an optimal cost vector obtained by solving $\mathbf{GIO}(\mathcal{D})$ is guaranteed to attain the maximum value for ρ. Like least squares regression and R^2, our inverse optimization model and this ρ metric form a unified framework for model fitting and evaluation in inverse linear optimization.

The second property makes ρ easily interpretable as a measure of goodness of fit, with higher values indicating better fit. Note that $\rho = 1$ if and only if $\sum_{q=1}^{Q} \left\|\boldsymbol{\epsilon}_q^*\right\| = 0$ (i.e., every point in $\mathcal{D}$ lies on a supporting hyperplane of $\mathcal{P}$). In this case, the model perfectly describes all of the data points, analogous to the best fit line passing through all data points in a linear regression. Conversely, $\rho = 0$ if and only if $\sum_{q=1}^{Q} \left\|\boldsymbol{\epsilon}_q^*\right\| = \sum_{q=1}^{Q} \left\|\boldsymbol{\epsilon}_{q,i}\right\|$ for all $i \in \mathcal{I}$. This scenario occurs when an optimal solution to the inverse optimization problem does not reduce the model-data fit error with respect to any of the baseline solutions, akin to when a linear regression returns an intercept-only model.

The third property states that goodness of fit is nondecreasing as additional degrees of freedom are provided to the modeler, analogous to the property that R^2 is nondecreasing in the number of features in a linear regression model. Because of this similarity, ρ also shares a weakness of R^2 related to overfitting. When using ρ to compare several models, one should ensure that higher values of ρ represent true improvements in fit, rather than artificial increases that lack generalizability.

3.3.3 Numerical examples

Examples 2 and 3 illustrate the value of fusing ρ instead of an unnormalized error measure such as the aggregate error. Intuitively, a given error with a larger feasible set indicates better fit than the same error in a smaller set. Further, ρ degrades when individual data points are forced to deviate from their preferred cost vector to minimize aggregate error. Example 4 showcases ρ for a problem where three points in the data set are fixed and the fourth is varied. Due to primal feasibility in $\mathbf{GIO}_p(\mathcal{D})$, decision and objective space yield different ρ.



(a) $\mathbf{FO}(\mathbf{c}; -2, 10)$. $\rho = 0.76$. (b) $\mathbf{FO}(\mathbf{c}; 4, 4)$. $\rho = 0.34$.

Figure 3.4: Illustration of Example 2. $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$ with two $\mathbf{FO}(\mathbf{c}; u, v)$, the same $\mathbf{c}^*$ and $\boldsymbol{\epsilon}^*$, but different ρ. $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$ for two different cases of $\mathbf{FO}(\mathbf{c}; u, v)$. The feasible sets are shaded. The black and red squares are $\hat{\mathbf{x}}_q$ and $\hat{\mathbf{x}}_q - \boldsymbol{\epsilon}_q^*$, respectively. Both problems yield the same $\mathbf{c}^*$ and $\boldsymbol{\epsilon}_q^*$, but have different model fitness.

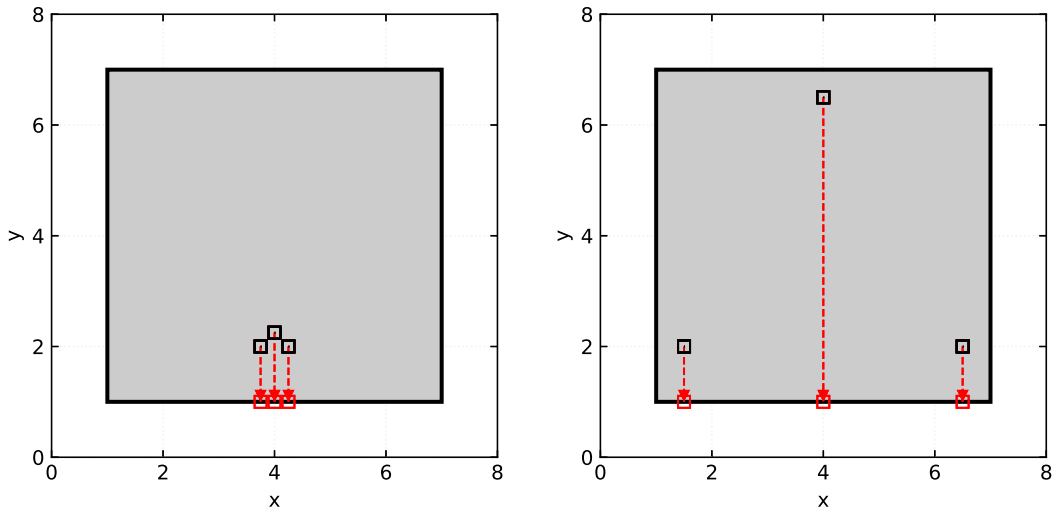
Example 2. *Consider a forward problem parameterized by u and v:*

$$\begin{aligned} \mathbf{FO}(\mathbf{c}; u, v): \quad \min_{\mathbf{x}} \quad & c_1 x_1 + c_2 x_2 \\ \text{s.t.} \quad & -0.71 x_1 + 0.71 x_2 \geq -2.83 \\ & u \leq x_1 \leq 7 \\ & 1 \leq x_2 \leq v \end{aligned}$$

and let $\mathcal{D} = \{(5, 2.5), (4.75, 3.75), (5.5, 3)\}$. Consider two cases of the problem: $\mathbf{FO}(\mathbf{c}; -2, 10)$ and $\mathbf{FO}(\mathbf{c}; 4, 4)$. $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$ yields $\mathbf{c}^* = (-0.5, 0.5)$ and $\sum_{q=1}^{3} |\epsilon_q^*| = 2.75$ for both, but $\rho = 0.76$ for $\mathbf{FO}(\mathbf{c}; -2, 10)$ and $\rho = 0.34$ for $\mathbf{FO}(\mathbf{c}; 4, 4)$. In Fig. 3.4(a), $\mathcal{D}$ is closer to the bottom facet, relative to the other facets, while in Fig. 3.4(b), $\mathcal{D}$ is near the "center" of the polyhedron rather than one facet.

Example 3. *Let*

$$\begin{aligned} \mathbf{FO}(\mathbf{c}): \quad \min_{\mathbf{x}} \quad & c_1 x_1 + c_2 x_2 \\ \text{s.t.} \quad & 1 \leq x_1 \leq 7 \\ & 1 \leq x_2 \leq 7. \end{aligned}$$



(a) $\mathcal{D}_1 = \{(3.75, 2), (4, 2.25), (4.25, 2)\}$. $\rho = 0.64$. (b) $\mathcal{D}_2 = \{(1.5, 2), (4, 6.25), (6.5, 2)\}$. $\rho = 0.17$.

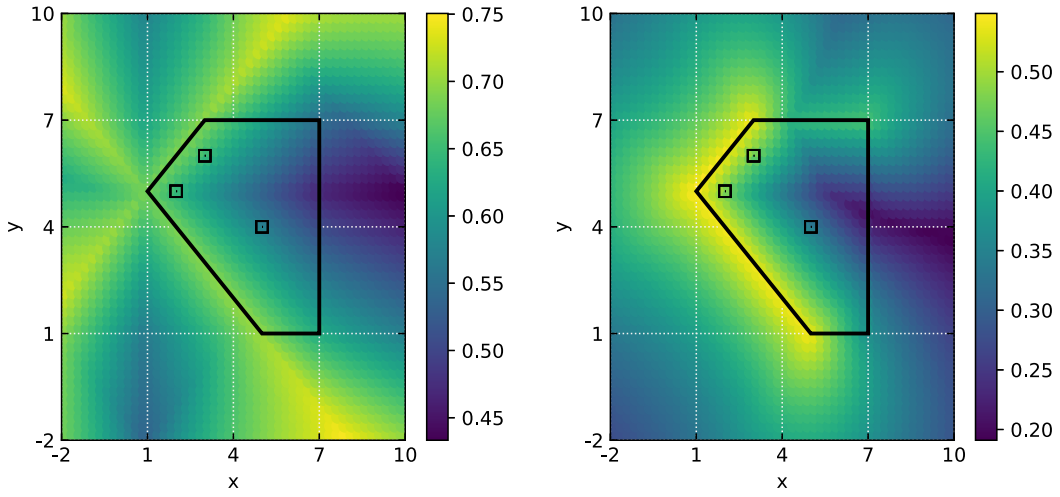
Figure 3.5: Illustration of Example 3. $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$ with two different data sets for the same $\mathbf{FO}(\mathbf{c})$. The feasible set is shaded. The black and red squares are $\hat{\mathbf{x}}_q$ and $\hat{\mathbf{x}}_q - \boldsymbol{\epsilon}_q^*$, respectively. Both problems yield the same $\mathbf{c}^*$ but have different errors and model fitness.

and let $\mathcal{D}_1 = \{(3.75, 2), (4, 2.25), (4.25, 2)\}$ *and* $\mathcal{D}_2 = \{(1.5, 2), (4, 6.25), (6.5, 2)\}$. *Both* $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D}_1)$ *and* $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D}_2)$ *impute* $\mathbf{c}^* = (0, 1)$. *In Fig. 3.5(a), the points are close together and prefer the bottom facet* ($\rho = 0.64$). *In Fig. 3.5(b), the points are further apart, each with a different preferred cost vector, but aggregate error is minimized by selecting a new different cost vector, resulting in poorer fit* ($\rho = 0.17$).

Example 2 and 3 show that the aggregate error and the imputed cost vector from inverse optimization can hide poor model-data fitness. However, poor fitness arises from poor models or poor data. In Example 2, the forward model $\mathbf{FO}(\mathbf{c}; 4, 4)$ uses constraints that are potentially too tight given the data. On the other hand in Example 3, the data set $\mathcal{D}_2$ is spread out and unlikely to be all generated with respect to a single objective.

Example 4. *Let*

$$\begin{aligned} \mathbf{FO}(\mathbf{c}): \quad \min_{\mathbf{x}} \quad & c_1 x_1 + c_2 x_2 \\ \text{s.t.} \quad & 0.71 x_1 + 0.71 x_2 \geq 4.24 \\ & 0.71 x_1 - 0.71 x_2 \geq -2.83 \\ & x_1 \leq 7 \\ & 1 \leq x_2 \leq 7 \end{aligned}$$



(a) $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$ (b) $\mathbf{GIO}_{2}(\mathcal{D})$

Figure 3.6: Illustration of Example 4. Heat maps of ρ for different $\mathbf{GIO}(\mathcal{D})$ where $\mathcal{D}$ consists of three fixed points and the fourth variable point. The feasible set is highlighted and the squares are the fixed $\hat{\mathbf{x}}_q$ of $\mathcal{D}$. ρ is high for $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$ along the relevant supporting hyperplanes, but is only high for $\mathbf{GIO}_{2}(\mathcal{D})$ along the facets.

and consider all data sets of the form

$$\mathcal{D} = \{(2,5),(3,6),(5,4),(\gamma_1,\gamma_2)\}\,.$$

Fig. 3.6 shows heatmaps of ρ for $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$ and $\mathbf{GIO}_{2}(\mathcal{D})$. For $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$, fitness is maximized when the fourth point lies on $\mathcal{H}_1 = \{(x_1,x_2) \mid 0.71x_1 - 0.71x_2 = -2.83\}$. If we solve $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$ with the three fixed points, then $\mathbf{c}^* = (0.5,-0.5)$. Thus, when the fourth point lies on $\mathcal{H}_1$, there is zero additional loss. ρ is also high when the fourth point lies on $\mathcal{H}_2 = \{(x_1,x_2) \mid 0.71x_1 + 0.71x_2 = 4.24\}$, and degrades as it moves away from these two hyperplanes.

We observe different behavior for ρ in $\mathbf{GIO}_{2}(\mathcal{D})$: maximum model fitness occurs when the fourth point lies along the facets of $\mathcal{P}$ defined by $\mathcal{H}_1$ and $\mathcal{H}_2$. Due to primal feasibility, if the fourth point is infeasible, it must project to $\mathcal{P}$ and thus incur some positive loss.

3.4 Ensemble inverse optimization for treatment planning in radiation therapy

In this section, we implement $\mathbf{GIO}(\mathcal{D})$ and demonstrate the use of ρ in the context of automated RT treatment planning. We consider a KBP pipeline where (1) a machine

learning model first predicts an appropriate dose distribution for a given patient, (2) an inverse optimization model treats the dose as an "observed decision" to impute candidate objective weights, and (3) the objective weights are input to the multi-objective planning problem to reconstruct a final plan (Babier et al., 2018b).

Since different prediction models lead to plans that over-fit to different clinical criteria, we harness an ensemble of predictions to generate a single treatment plan. However, instead of averaging predictions (like in a random forest), we keep each prediction separate, and feed them all into one inverse optimization model (see Figure 3.1(b)). Until now, KBP has never been used to generate a single plan from multiple predictions.

We develop an ensemble KBP approach using eight different predictions and show that the relative duality gap model dominates the absolute duality gap model for this application. Plans from the relative duality gap model outperform the majority of the single-point models on our overall clinical metric. Finally by removing certain low-quality predictions, we design a final model that outperforms all of the single-point KBP baselines. Although the final model requires clinically-driven model engineering, we use ρ as domain-independent validatation of the clinical intuition.

3.4.1 Data and methods

We use our clinical data set of 217 treatment plans for patients with oropharyngeal cancer randomly split into 130 plans for training and 87 plans for testing (see Chapter 2 for details on the data). The training set is used to train our predictive models and then discarded; the testing set is used to implement our inverse optimization framework. For each patient k, a treatment plan is generated by solving a multi-objective optimization problem $\mathbf{RT}$–$\mathbf{FO}(\boldsymbol{\alpha}_k) : \min_{\mathbf{x}} \left\{ \boldsymbol{\alpha}_k^\intercal \mathbf{C}_k \mathbf{x} \mid \mathbf{A}_k \mathbf{x} \geq \mathbf{b}_k \ , \ \mathbf{x} \geq \mathbf{0} \right\}$, where $\mathbf{C}_k$ is the matrix whose rows represent different cost vectors and $\boldsymbol{\alpha}_k$ is the vector of objective weights. The decision vector contains two subvectors, $\mathbf{x} = (\mathbf{w}, \mathbf{d})$, where $\mathbf{w}$ is the intensity of each beamlet of radiation and $\mathbf{d}$ is the dose delivered to each voxel of the patient's body, computed from a linear transformation of $\mathbf{w}$. This multi-objective model fits into $\mathbf{GIO}(\mathcal{D}_k)$ by specifying the set of feasible cost vectors for patient k as $\mathcal{C}_k = \left\{ \mathbf{C}_k^\intercal \boldsymbol{\alpha} \mid \boldsymbol{\alpha} \geq \mathbf{0} \right\}$. Furthermore, the optimization problem for each patient is distinct. For a specific patient, the feasible set is fixed and a single treatment optimization problem is solved. The ensemble arises from the multiple dose predictions for the patient.

We first train four different dose prediction models, labeled Random Forest (RF), 2-D RGB GAN, 2-D GANCER, and 3-D GANCER (Babier et al., 2018b, 2020a; Mahmood et al., 2018). For each model, we also implement versions with scaled predictions (suf-

fixed with '-sc.'), which are known to produce plans that better satisfy target (tumor) criteria (Babier et al., 2020a). Thus, we have eight predictions per patient, which vary in their dose trade-offs between the targets and healthy organs. We predict the dose $\hat{\mathbf{d}}_{k,q}$ for each test patient $k \in \{1, \ldots, 87\}$ with prediction model $q \in \{1, \ldots, 8\}$ and let $\mathcal{D}_k = \left\{\hat{\mathbf{d}}_{k,1}, \ldots, \hat{\mathbf{d}}_{k,8}\right\}$ be data for each patient-specific problem. We then use inverse optimization to construct an optimal treatment plan given these predictions.

For each patient k in the test set, we implement the absolute and relative duality gap models, referred to as **RT**–$\mathbf{IO}_{\mathrm{A}}(\mathcal{D}_k)$ and **RT**–$\mathbf{IO}_{\mathrm{R}}(\mathcal{D}_k)$, respectively. They are derived from $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D}_k)$ and $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D}_k)$ by setting $\mathcal{C}_k$ as defined above, along with the template hyperparameters of Proposition 1 and Proposition 4, respectively. Once an objective weight vector $\boldsymbol{\alpha}_k^*$ is imputed from one of the inverse models, we solve **RT**–$\mathbf{FO}(\boldsymbol{\alpha}_k^*)$ to determine the beamlets $\mathbf{w}_k^*$ and dose $\mathbf{d}_k^*$. The dose $\mathbf{d}_k^*$ is then evaluated using different clinical criteria. Note that we are not attempting to re-construct beamlets or a dose distribution that is similar in p-norm to the predictions, but rather learning the objective function weights that the predictions appear to prioritize in order to construct a plan that best reflects clinical preferences. Since plan quality is evaluated on dosimetric values in practice, we focus only on the objective space model variants. Detailed descriptions of the prediction models and the formulation of the inverse optimization models are provided in Appendix A.3.

3.4.2 The value of ensemble inverse optimization

In practice, a suite of quantitative metrics are evaluated to assess whether sufficient dose is delivered to the tumor and the surrounding healthy tissue is sufficiently spared. In line with clinical practice, we use 10 binary criteria for plan evaluation (see Table 2.1; also Babier et al. (2018a)). To evaluate our plans on these criteria, we first check whether the corresponding clinical (ground truth) plan satisfied given criteria. If the clinical plan satisfied the criteria, we evaluate whether the generated plan also satisfied that criteria.

The columns of Table 3.2 list the proportion of plans generated by **RT**–$\mathbf{IO}_{\mathrm{A}}(\mathcal{D})$ and **RT**–$\mathbf{IO}_{\mathrm{R}}(\mathcal{D})$ that satisfied the corresponding clinical criteria. The 'All' row reflects the percentage of plans that satisfied all of the criteria that were also met by the corresponding clinical plans and is an aggregate measure of plan quality. We first use all eight predictions to solve **RT**–$\mathbf{IO}_{\mathrm{A}}(\mathcal{D})$ (column 3) and **RT**–$\mathbf{IO}_{\mathrm{R}}(\mathcal{D})$ (column 4). **RT**–$\mathbf{IO}_{\mathrm{R}}(\mathcal{D})$ substantially outperforms the **RT**–$\mathbf{IO}_{\mathrm{A}}(\mathcal{D})$ over every criterion, suggesting that the absolute duality gap model is not well-suited to this application. This result is consistent with results observed for single-point inverse optimization in RT (Chan et al., 2014, 2019;

Table 3.2: The percentage of final plans of each KBP population that satisfy the same clinical criteria as the corresponding clinical plans. OARs are assigned a mean or maximum dose criteria depending on relevance. PTVs are assigned criteria to the 99%-ile.

Structure	Criteria (Gy)	**RT**–**IO**$_{\mathrm{A}}(\mathcal{D})$	**RT**–**IO**$_{\mathrm{R}}(\mathcal{D})$			
		8 Pts.	8 Pts.	6 Pts.	4 Pts.	2 Pts.
Brainstem	Max ≤ 54	100	100	100	100	100
Spinal Cord	Max ≤ 48	100	100	98.9	98.9	100
Right Parotid	Mean ≤ 26	58.8	88.2	88.2	82.4	94.1
Left Parotid	Mean ≤ 26	63.6	81.8	81.8	81.8	81.8
Larynx	Mean ≤ 45	59.2	95.9	95.9	93.9	95.9
Mandible	Mean ≤ 45	74.4	100	100	100	100
Esophagus	Max ≤ 73.5	51.5	100	98.5	95.5	97.0
PTV70	99%-ile ≥ 66.5	51.7	91.4	94.8	96.6	86.2
PTV63	99%-ile ≥ 59.9	50.0	98.0	98.0	98.0	98.0
PTV56	99%-ile ≥ 53.2	30.4	45.7	80.4	100	69.6
All Structures		26.4	60.9	75.9	83.9	70.1

Goli et al., 2018) and we conjecture that it is due to the wide range of objective function magnitudes in the forward problem. The absolute duality gap model adjusts each objective value by the same absolute amount, causing relatively large adjustments to objectives with low values and small adjustments to those with high values; thus, it has difficulty balancing different criteria.

Although **RT**–**IO**$_{\mathrm{R}}(\mathcal{D})$ with eight predictions is generally effective at satisfying the OAR criteria, these plans sacrifice the PTV criteria, especially PTV56. We hypothesize that this performance for PTV criteria is due to the large variability in the quality of predictions. For example, the 2-D RGB GAN, 2-D GANCER, and 3-D GANCER models are known to produce plans that emphasize OAR criteria at the expense of the PTV. Criteria satisfaction for single-point **RT**–**IO**$_{\mathrm{R}}(\{\hat{\mathbf{x}}\})$ using each of the individual predictions is shown in Table 3.3. Depending on which prediction is used, the single-point KBP population varies from 10.9% to 95.7% in terms of satisfying the PTV56 criteria. The ability of the single-point models to satisfy all clinical criteria ranges between 44.8% and 80.5%, suggesting that some single-point KBP models make poorer trade-offs in criteria satisfaction than others. Regardless of the variability among predictions, the ensemble model outperforms all but the top three single-point models in satisfying all criteria. In cases where the cost of determining model performance is expensive (e.g., having to solve inverse and forward models over multiple predictions and patients), ensemble inverse optimization can reliably provide high-quality plans.

Using multiple points of varying quality as input to the ensemble model may lead to

Table 3.3: The percentage of single-point inverse optimization plans of each KBP population that satisfy the same clinical criteria as the clinical plans.

Structure	Criteria (Gy)	$\mathbf{RT}$–$\mathbf{IO}_{\mathrm{R}}(\{\hat{\mathbf{x}}_q\})$							
		3-D GANCER	2-D RGB GAN	2-D GANCER	2-D RGB GAN-sc.	RF-sc.	RF	2-D GANCER-sc.	3-D GANCER-sc.
Brainstem	Max ≤ 54	100	100	100	100	98.9	100	100	100
Spinal Cord	Max ≤ 48	100	98.9	100	98.9	98.9	100	98.9	98.9
Right Parotid	Mean ≤ 26	94.1	94.1	82.4	88.2	94.1	88.2	88.2	94.1
Left Parotid	Mean ≤ 26	100	90.9	81.8	63.6	72.8	63.6	81.8	81.8
Larynx	Mean ≤ 45	98.0	89.8	89.8	87.8	95.9	91.8	85.7	93.9
Mandible	Mean ≤ 45	100	100	100	100	98.7	100	100	100
Esophagus	Max ≤ 73.5	100	100	100	98.5	100	100	89.4	84.8
PTV70	99%-ile ≥ 66.5	81.0	36.2	81.0	69.0	63.8	91.4	98.3	100
PTV63	99%-ile ≥ 59.9	92.0	100	100	100	98.0	98.0	100	100
PTV56	99%-ile ≥ 53.2	10.9	58.7	19.6	82.6	47.8	65.2	95.7	95.7
All Structures		44.8	47.1	47.1	59.8	55.2	67.8	77.0	80.5

poor model-data fit (see Example 3). We experiment with IO models based on subsets of the eight predictions to determine which subset of KBP prediction models best fit $\mathbf{RT}$–$\mathbf{FO}(\boldsymbol{\alpha})$. The clinical KBP literature shows that some of the prediction models generally perform better than others: scaled GANCER models typically predict better than RF, which themselves predict better than RGB-GAN and unscaled GANCER (Babier et al., 2020a; Mahmood et al., 2018). Using the prior literature and qualitative assessment from a clinical collaborator, we propose an ordering of the models from weak to strong: 3-D GANCER, 2-D RGB GAN, 2-D GANCER, 2-D RGB GAN-sc., RF-sc., RF, 2-D GANCER-sc., 3-D GANCER-sc. Note the general pattern is more important than the exact ordering. That is, we rate the scaled GANCER models as strongest, followed by RF models, followed by RGB GAN and unscaled GANCER. We implement $\mathbf{RT}$–$\mathbf{IO}_{\mathrm{R}}(\mathcal{D})$ with data sets of decreasing size by sequentially removing the two weakest predictors. For example, the 6 Pts. IO model uses the very strong, strong, and weak predictions, while the 4 Pts. model uses the very strong and strong predictions. Columns 5–7 of Table 3.2 show the performance of the three subset IO models. The 6 Pts. model markedly improves over the 8 Pts. model on PTV criteria, while satisfying almost all OAR criteria, resulting in an additional 15% of the final plans being able to satisfy all criteria. Simi-

Table 3.4: The percentage of plans from different ensemble models that satisfy the same clinical criteria as the corresponding clinical plans. **RT–IO**$_{\mathrm{R}}(\mathcal{D})$ refers to the 4 Pts. model from Table 3.2. We present the best performing setting for each baseline.

Structure	Criteria (Gy)	**RT–IO**$_{\mathrm{R}}(\mathcal{D})$	Centroid	MWA
Brainstem	Max ≤ 54	100	100	100
Spinal Cord	Max ≤ 48	98.9	100	100
Right Parotid	Mean ≤ 26	82.4	88.2	88.2
Left Parotid	Mean ≤ 26	81.8	81.8	63.6
Larynx	Mean ≤ 45	93.9	87.8	91.8
Mandible	Mean ≤ 45	100	98.5	100
Esophagus	Max ≤ 73.5	95.5	100	100
PTV70	99%-ile ≥ 66.5	96.6	96.6	93.1
PTV63	99%-ile ≥ 59.9	98.0	100	98.0
PTV56	99%-ile ≥ 53.2	100	80.4	67.4
All Structures		83.9	77.0	69.0

larly, the 4 Pts. model improves over the 6 Pts. model by achieving near perfect PTV criteria satisfaction while mostly preserving OAR performance. In fact, this model now outperforms the best single-point model, 3-D GANCER-sc. (see Table 3.3). Interestingly, performance does not improve in the 2 Pts. model. This model uses two predictions (2-D GANCER-sc. and 3-D GANCER-sc.) that individually achieve high PTV satisfaction in their single-point models, but fail to do so when combined in an ensemble. We conjecture that the 2 Pts. model reaches a local minimum in PTV satisfaction because the forward objectives do not directly target PTV criteria (see Appendix A.3).

Overall, these experiments demonstrate the value of ensemble inverse optimization for turning an ensemble of predictions into a single plan. While an off-the-shelf ensemble model immediately outperforms most single-point constituents, careful selection of data is required to maximize performance and beat all single-point KBP models.

3.4.3 Comparison with existing ensemble learning techniques

We next compare **RT–IO**$_{\mathrm{R}}(\mathcal{D})$ with two conventional ensemble learning baselines that do not account for linear programming geometry. The first baseline is an "ensemble-then-inverse optimization" approach where for each patient k, the centroid $\bar{\mathbf{d}}_k$ of the individual predictions is input into a single-point inverse optimization problem **RT–IO**$_{\mathrm{R}}(\{\bar{\mathbf{d}}_k\})$. The second baseline is a Multiplicative Weights Algorithm (MWA), commonly used in "learning from experts" settings (Arora et al., 2012). Here, we first solve the single-point problem **RT–IO**$_{\mathrm{R}}(\{\hat{\mathbf{d}}_{k,q}\})$ with each prediction model for the training set patients. We

Table 3.5: ρ for the Weak, Medium, and Strong subsets of 2, 4, and 6 Pts. The All criteria percentage satisfaction for each model are in parentheses. The Strong column reflects the predictions used in Table 3.2. Highest performing models are grayed.

	Weak	Medium	Strong
2 Pts.	0.63 (42.5)	0.65 (60.9)	0.90 (70.1)
4 Pts.	0.56 (30.1)	0.68 (62.1)	0.73 (83.9)
6 Pts.	0.64 (51.7)	0.63 (57.5)	0.67 (75.9)

treat each prediction model as a different expert and learn a probability distribution over the set of prediction models using the aggregate error as a loss function. Then for each patient in the test set, we use this distribution to randomly sample a prediction model and solve a single-point problem. Implementation details are given in Appendix A.3.6.

We implement the Centroid and the MWA model using all eight predictions per patient (i.e., 8 Pts.), as well as the 4 Pts. predictions (RF-sc., RF, 2-D GANCER-sc., and 3-D GANCER-sc.). Table 3.4 compares our incumbent, the 4 Pts. $\mathbf{RT}$–$\mathbf{IO}_{\mathrm{R}}(\mathcal{D})$, with the best-performing Centroid and MWA models. If all dose predictions were feasible with respect to $\mathbf{RT}$–$\mathbf{FO}(\boldsymbol{\alpha})$, then by Proposition 6, our ensemble model and the Centroid model would be equivalent. Each prediction model outputs feasible doses for approximately 85% of the patients (see Table A.1 in Appendix A.3). Consequently, $\mathbf{RT}$–$\mathbf{IO}_{\mathrm{R}}(\mathcal{D})$ yields different plans from $\mathbf{RT}$–$\mathbf{IO}_{\mathrm{R}}(\bar{\mathbf{d}}_k)$. Our incumbent outperforms the baseline on the 'All' criteria by 6.9%. Nonetheless, the Centroid model is similar to $\mathbf{RT}$–$\mathbf{IO}_{\mathrm{R}}(\mathcal{D})$ for each individual criteria. We intuit that if only a small fraction of points in $\mathcal{D}$ are infeasible, then centroid inverse optimization is an efficient approximation of ensemble inverse optimization.

The MWA baseline randomly selects a single-point inverse optimization model for each patient according to a learned probability distribution. This approach is a tractable alternative to solving eight inverse optimization problems and selecting the best plan for each patient (see Figure 3.1a). As shown in Table 3.3, some single point models are significantly better than others. Thus, most of the test set patients will receive plans from RF, 2-D GANCER-sc. or 3-D GANCER-sc. However, $\mathbf{RT}$–$\mathbf{IO}_{\mathrm{R}}(\mathcal{D})$ already outperforms each of these single-point models on most of the criteria. Consequently, $\mathbf{RT}$–$\mathbf{IO}_{\mathrm{R}}(\mathcal{D})$ outperforms the MWA baseline on all criteria by 14.9%.

3.4.4 Using ρ to validate the best subset of the data

Above, we showed that using a targeted subset of the predictions yielded a better model. The intuition follows Example 3, where points that are individually far from each other

induce poor fit. While our ranking scheme was domain-specific, here we demonstrate a domain-independent validation of the selection of the data sets in the 6 Pts., 4 Pts., and 2 Pts. models.

We consider three variants for each of the 6 Pts., 4 Pts., and 2 Pts. models by selecting subsets of strong, medium, and weak predictions according to our clinical ordering. Strong subsets correspond to the models developed in Section 3.4.2, Weak subsets use the worst predictions and sequentially remove the best from these subsets, and Medium subsets use the predictions from 2-D RGB GAN to 2-D GANCER-sc. and sequentially remove one strong and weak prediction. Table 3.5 compares ρ across models with varying quality of predictions. Note that we are not studying the effect of data set size Q (along columns of Table 3.5), but rather the effect of quality (along rows of Table 3.5). For fixed Q, the Strong model always yields the highest ρ, which suggests that the Top predictions are the best fit for the clinical forward model. Furthermore in parentheses in Table 3.5, we show that the clinical criteria satisfaction rates for each of the ensemble models also reflect similar trends as ρ. Since ρ is a general metric, we can evaluate the model quality for a given number of points without domain specific knowledge, and come to nearly the same conclusion as via the clinical criteria, which are domain-specific and require additional computation due to re-solving the forward model.

However, ρ is not a perfect surrogate for criteria satisfaction. For example, the Weak 6 Pts. model has a slightly higher ρ than the Medium 6 Pts. model. Note that the two data sets share four of six points and the relatively similar ρ reflects a similar criteria satisfaction rate. We also observe that the data set with the best fit from an inverse optimization perspective (Strong 2 Pts.) is not the one resulting in the best clinical criteria evaluation (Strong 4 Pts.). This result is due to the fact ρ is calculated via the average distance of the predictions to the constraints, but the constraints only approximate the criteria (see Appendix A.3.1). Because the predictions are close to the constraints but not criteria, ρ is overly optimistic for this model. Using diverse predictions of high clinical quality allows us to obtain ρ values that are more representative of the clinical problem.

3.5 Conclusion

Inverse optimization is an increasingly popular model-fitting paradigm for estimating the cost vector of an optimization problem given decision data. Motivated by ensemble methods in machine learning, we develop a framework that uses a collection of decisions for a single problem to estimate a cost vector. The data is drawn from different decision-makers attempting to solve a single problem or, as in our application, a family of

machine learning-generated predictions of an optimal solution. We propose a generalized inverse linear optimization framework that unifies several common variants of inverse optimization from the literature and derive assumption-free exact solution methods for each. Comparing with the inverse convex optimization literature shows that by focusing on our specialized context, we can leverage the geometry of linear optimization to produce tighter performance bounds and more efficient solution methods. To complete our framework, we develop a general goodness of fit metric to measure model-data fit in any inverse linear optimization application. By virtue of possessing properties analogous to R^2 in linear regression, this metric is easy to calculate and interpret.

We propose a novel application of ensemble inverse optimization in the automated construction of radiation therapy treatment plans. In contrast to traditional approaches, which generate plans from individual predictions, we use a family of predictions, each with different characteristics and trade-offs, to form treatment plans that better imitate clinically delivered treatments. Finally, while constructing the best inverse optimization model requires careful clinical expertise, we show how our goodness-of-fit metric provides domain-independent validation of our model engineering. Beyond the specific context and application presented in this chapter, we believe there will be new applications of predict-then-inversely optimize frameworks that can build on our foundation.

Chapter 4

Dose generation with generative adversarial networks

A key enabler of automated KBP is the dose generation module. While dose generation typically uses CT images to predict a clinically acceptable dose distribution, previous dose generation models consisted of classical machine learning that relied on low-dimensional hand-tailored geometric features drawn from a CT image (e.g., a regression model predicting the dose to a cubic region may use the distance from that region to the nearest tumor structure). In this chapter, we propose a paradigm for generating KBP predictions that learns to predict a 3-D dose distribution directly from a CT image. Specifically, we recast dose generation as an image colorization problem, which we solve using a generative adversarial network (GAN) (Goodfellow et al., 2014). GANs, which have produced impressive results in other image colorization applications (Isola et al., 2017; Zhu et al., 2017), involve a pair of neural networks: a generator that performs a task and a discriminator that evaluates how well the task is performed. In our application, the generator imitates a dosimetrist that designs a treatment, while the discriminator plays the role of the oncologist who critiques the generated dose distribution by comparing it to the real treatment plan. Both neural networks train simultaneously on historical data, effectively replicating and aggregating the combined knowledge gained during the iterative manual process used to design clinically acceptable treatments.

We develop an automated KBP pipeline for head-and-neck cancer that uses a GAN to predict three-dimensional dose distributions. In contrast to previous machine learning methods, our approach does not require the pre-specification of an extensive set of feature variables for prediction. Instead, our model learns what features are important to produce clinically acceptable treatment plans. We apply our KBP methodology to our clinical dataset of 217 patients with head-and-neck cancer (i.e., 26,279 2-dimensional

CT image slices). We input all predictions into a plan optimization and evaluate on deliverable plans. We compare our approach to feature-based machine learning models and a standard convolutional neural network (CNN) to demonstrate that we outperform these baselines in achieving clinical criteria and similarity metrics.

Remark 2. *The dose generation model that we introduce in this chapter is equivalent to the 2-D RGB GAN used in experiments from Chapter 3. While the results were chronologically completed before the previous chapter, we include it here to also motivate the next chapter, which advances on dose generation techniques.*

4.1 Background

GANs are a well-studied class of deep learning algorithms used in *generative* modeling, i.e., in the creation of new data (Goodfellow et al., 2014). Although initially used to artificially generate 2-D images, and later 3-D models (Wu et al., 2016), their success has garnered increasing interest for healthcare applications. GANs have been used for medical drug discovery (Kadurin et al., 2017), generating artificial patient records (Choi et al., 2017; Esteban et al., 2017), the detection of brain lesions (Alex et al., 2017), and image augmentation for improved liver lesion classification (Frid-Adar et al., 2018).

A GAN consists of two neural networks, a generator and a discriminator, working in tandem. The generator $G(\cdot)$ takes an initial random input $\mathbf{z} \sim p_{\mathbf{z}}$ and attempts to generate an artificial data sample $\mathbf{x} = G(\mathbf{z})$ (i.e., the 3-D dose distribution). The discriminator $D(\cdot)$ is a classifier that takes generated and real data samples, and tries to identify which is which, i.e., $D(\mathbf{x}) \in [0, 1]$ where 1 suggests the generated sample is satisfactory. The interaction between the networks can be formalized mathematically as a minimax game. If $\mathbf{x} \sim p_{\text{data}}$ is the probability distribution over the real data samples, then the game is defined as

$$\min_G \max_D \Big\{ V(G, D) := \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} \Big[\log D(\mathbf{x}) \Big] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}} \Big[\log \big(1 - D(G(\mathbf{z})) \big) \Big] \Big\}.$$

GANs have been proven effective in Style Transfer problems, where the generator input $\mathbf{z}$ is a data sample corresponding to one style (or characteristic) and the output $\mathbf{x}$ is a mapping to a different style (Isola et al., 2017; Zhu et al., 2017). For example, Style Transfer can be used to transform grayscale images to colored photos (Sangkloy et al., 2017), in facial recognition for surveillance-based law enforcement (Wang et al., 2017), and in 3-D reconstruction of damaged artifacts (Hermoza and Sipiran, 2017). In our setting then, $\mathbf{z}$ corresponds to a CT image, while $\mathbf{x}$ represents a dose distribution. The

generator $G(\mathbf{z})$ learns the mapping between styles that generates samples resembling the ground truth. Since key structures in the output may be entangled with noise from the generator, the desired output is often achieved by modifying the original minimax game with a penalty term on large deviations between the real and generated samples:

$$\min_G \max_D \left\{ V(G,D) + \lambda \mathbb{E}_{\mathbf{x}\sim p_{\text{data}},\mathbf{z}\sim p_{\mathbf{z}}} \Big[\left\| \mathbf{x} - G(\mathbf{z}) \right\|_1 \Big] \right\}, \tag{4.1}$$

where λ is a regularizer that balances the trade-off between the two objectives.

4.2 Methods

We use contoured CT images and clinically acceptable dose distributions from the treatment plans of past oropharyngeal cancer patients to train a Style Transfer GAN. We then pass out-of-sample predicted dose distributions through an IO pipeline (Babier et al., 2018a) to generate the final treatment plans. For baseline comparisons, we also implement several methods from the prior literature using the complete pipeline. Details on our clinical data set are given in Chapter 2.

4.2.1 Dose generation

We first divide each 3-D CT image into 2-D slices of 128×128 pixels. The generator uses a single CT image slice to predict the dose distribution along that same plane without considering the vertical relationship between different slices. This process is repeated for every slice until a full 3-D dose distribution can be produced. Our training set consists of all 2-D slices from the 3-D CT images for 130 patients, totaling 15,657 images. The CT images from the remaining 87 patients are used for out-of-sample evaluation.

Our GAN learning model is built on the `pix2pix` Style Transfer architecture of Isola et al. (2017). We use a U-net generator that passes a 2-D contoured CT image slice through consecutive convolution layers, a bottleneck layer, and then through several deconvolution layers. The U-net also employs skip connections, i.e., the output of each convolution layer is concatenated to the input of a corresponding deconvolution layer. This allows the generator to easily pass "high dimensional" information (e.g., structural outlines) between the inputted CT image slice and the outputted dose slice. The discriminator passes a 2-D slice of the dose distribution along several consecutive convolution layers, outputting a single scalar value. In the training phase, the discriminator receives one real and one generated dose distribution before backpropagation. The discriminator

is disconnected after training, at which point the generator receives only a contoured CT slice. We refer the reader to Appendix B.1 for details on the network architectures.

The GAN is trained via the loss function given by (4.1) with $\lambda = 90$ using the Adam optimizer (Kingma and Ba, 2014) with learning rate 0.0002 and $\beta_1 = 0.5$ and $\beta_2 = 0.999$ for 25 epochs. We use the default Adam settings from Isola et al. (2017), as they were proven to be efefctive for a variety of different Style Transfer problems. We find these default settings to be sufficient with minimal subsequent improvement and consequently avoid extensive parameter tuning. The code for all experiments, along with the parameter settings is provided at `http://github.com/rafidrm/gancer`.

4.2.2 Plan generation

We input our predicted dose generation models into a single-point inverse optimization model to generate optimized plans (see Chapter 3). We use `Gurobi 7.5` to solve the inverse and forward optimization problems associated with the IO pipeline. Details on the forward and inverse problems are provided in Appendix A.3 and Appendix B.3.

Remark 3. *Our inverse optimization model follows the same approach as the experiments in Chapter 3 with two differences. First, note that we use a single dose estimate rather than an ensemble. Second, the formulation here relaxes the dose forward safety constraints since these constraints are not necessary for modeling.*

4.2.3 Baseline approaches

We compare our GAN approach to generating predicted dose distributions with several baseline techniques. We briefly describe the baseline approaches here.

- **Bagging query (BQ):** A look-up method identifies patients with similar geometries who have undergone radiation therapy and outputs their doses as predictions. This approach predicts dose volume histograms (DVHs), i.e., 2-D summaries of the 3-D dose delivered to specific targets and OARs (e.g., Babier et al. (2018b); Wu et al. (2009)).

- **Generalized PCA (gPCA):** A method combining PCA with linear regression using patient geometry features. Similar to BQ, this method also predicts DVHs (e.g., Babier et al. (2018b); Yuan et al. (2012)).

- **Random forest (RF):** Predicts dose to each voxel (3-D dose prediction) using ten customized features based on patient geometry (inspired by McIntosh et al. (2017)). Additional details can be found in Appendix B.2.

- **U-net (CNN):** Predicts dose to each voxel in 2-D slices from a CT image using a U-net convolution neural network architecture (e.g., Nguyen et al. (2017)). Additional details can be found in Appendix B.1.

Note that while RF and CNN predict dose distributions, BQ and gPCA do not. We feed all baseline predictions into the same IO pipeline as the GAN to ensure a fair comparison between deliverable plans.

4.3 Results

4.3.1 Sample generated dose distributions

The Style Transfer function mapping the CT image to the predicted dose distribution appears easy to learn. This is because the GAN generated dose distributions had the hallmarks of a deliverable plan, like the sharp dose gradients that are generated by individual beams. However, our experiments demonstrated subtle deliverability characteristics that the GAN cannot always identify. The optimization step enforces these physical deliverability constraints to correct for these idiosyncracies. This result can be observed in Figure 4.1, where sample slices of a clinical, predicted, and optimized plan are shown. Visually, the GAN predictions resemble clinical plans with slight blurring artifacts.

4.3.2 Clinical criteria satisfaction

We measure plan quality by evaluating how frequently each plan satisfies the standard clinical criteria for head-and-neck cancer treatment plans (see Table 2.1). Clinicians commonly use criteria satisfaction as a metric to evaluate plan quality and approve a treatment plan after it satisfies a sufficient number of the criteria. Thus, each criterion (one per OAR and target) is measured on a pass-fail basis depending on whether the mean dose, maximum dose, or the dose to 99% of the volume of that structure (i.e., value-at-risk to the 99-th percentile), is above or below a given threshold.

Remark 4. *The evaluation approach for criteria satisfaction in this chapter differs slightly from Chapter 3. Here, we evaluate generated plans on whether they satisfy each criteria independent from the performance of the ground truth clinical plans. In the previous chapter, we evaluated whether the generated plans that matched the same criteria as the clinical plans; that is, evaluating plans only for the criteria that the corresponding ground truth plans satisfied. Therefore here, we can also compare generated plans with clinical plans and investigate potential inefficiencies with delivered plans. However,*

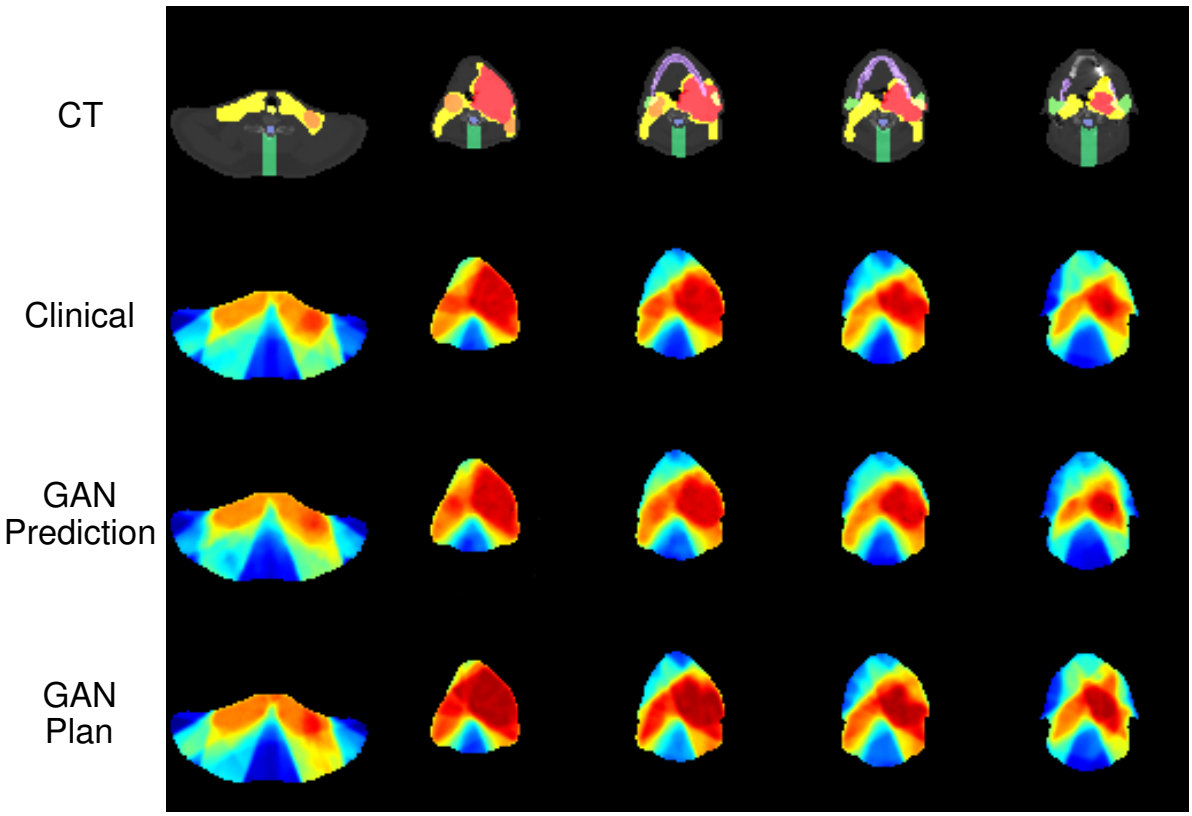


Figure 4.1: Sample of slices from a test patient. From top to bottom: contoured CT image (generator input), clinical plan (ground truth), GAN prediction, and GAN plan (post optimization).

generated plans evaluated on this metric may not make the same clinical trade-offs (i.e., meeting a specific criteria at the cost of another). Furthermore to facilitate comparisons, we re-scale the doses corresponding to all of the generated plans so that the dose delivered to the PTV70 from each generated plan is equal to the dose delivered to the PTV70 of the corresponding clinical plan.

Table 4.1 presents the percentage of the GAN and baseline treatment plans that satisfied the clinical criteria. There are more than triple the number of OAR criteria as PTV criteria since after all plans are normalized to the PTV70, we are left only with the PTV63 and PTV56 to evaluate. We note that clinically acceptable plans typically cannot satisfy all criteria simultaneously because of the proximity of the targets to the OARs and the complexity of the head-and-neck site in general. We observe that the BQ and gPCA plans tend to satisfy PTV criteria more frequently, which suggests that they may recommend delivering a higher dose to the target relative to the clinical plan. However, they fail to achieve mean and maximum dose criteria to the OARs. On the other hand, the RF plans satisfy fewer clinical criteria associated with the target as compared to

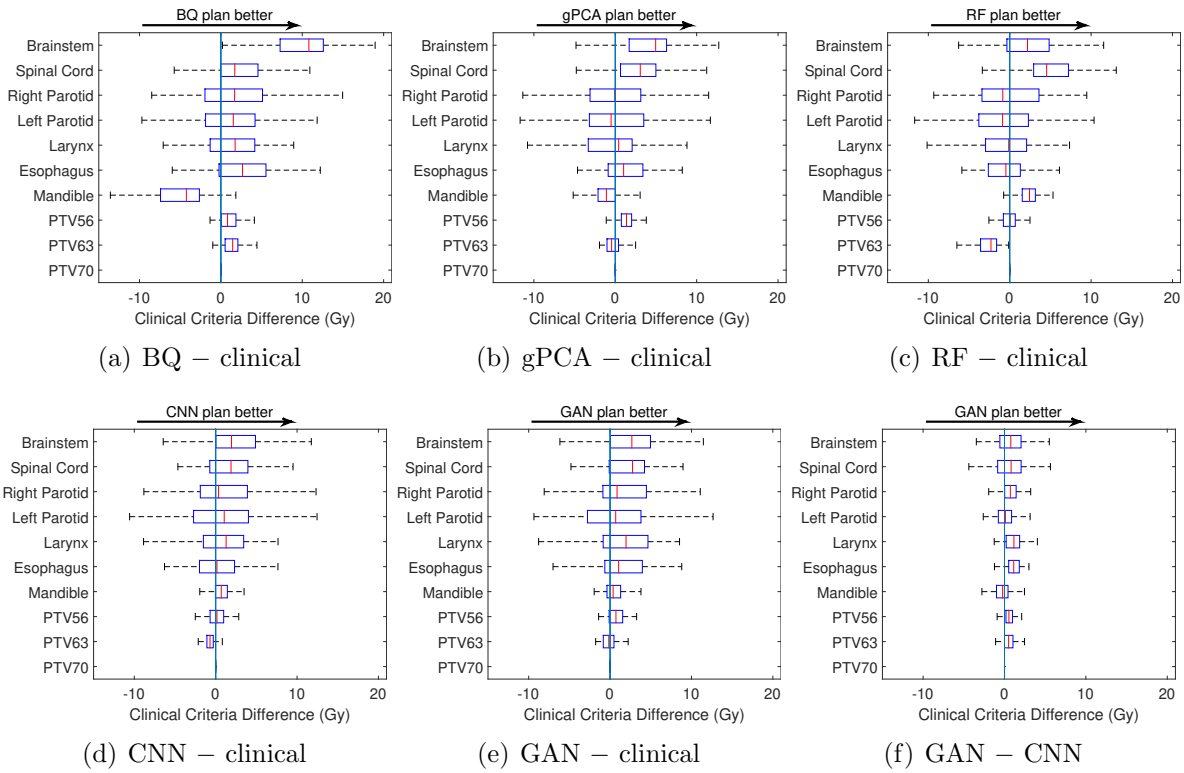
Table 4.1: The percentage of final plans of each KBP population that satisfy all of the clinical criteria of each category.

	BQ	gPCA	RF	CNN	GAN	Clinical
OAR criteria	61.6	65.8	71.5	72.5	72.8	72.0
PTV criteria	83.5	85.7	68.0	76.3	81.3	76.8
All criteria	67.6	71.2	70.7	73.6	75.2	73.3

the clinical plans. While the CNN plans achieve the closest level of performance to the clinical plans, the GAN plans obtain the best overall performance compared to all KBP pipelines. Our GAN-based KBP pipeline offers a balanced trade-off between the OARs and targets, and even outperforms the clinical plans on clinical criteria satisfaction.

The previous results focused on pass-fail performance with respect to the clinical criteria. In addition to pass-fail performance with respect to the clinical criteria, we also examine the magnitude of passing or failing via head-to-head comparisons of the GAN and baseline plans with respect to the clinical plans (see Figure 4.2). The x-axis in each figure is the difference in Gray (Gy) between the KBP and the clinical plans (KBP minus clinical) for the criterion on the corresponding y-axis. For each criterion, the majority of GAN plans outperform their clinical counterparts by several Gy (Figure 4.2(e)). This is a significant result given that the clinical plans are heavily optimized and delivered to actual patients. In contrast, the BQ, gPCA, and RF plans display substantial variability in performance when compared to the clinical plan. Consistent with Table 4.1, performance of the CNN plans are closest to the GAN plans although, as shown in Figure 4.2(f), the GAN plans maintain a small, yet consistent, advantage.

Finally, we compare the KBP plans against the clinical plans using the gamma passing rate (GPR) metric. GPR measures the similarity between two dose distributions on a voxel-by-voxel basis, computing for each voxel, a pass-fail test. We consider the standard choice of GPR, i.e., a 3%/3 mm tolerance (Low et al., 1998), which roughly means that a voxel in the evaluated dose distribution (KBP) "passes" if there is at least one voxel in the reference dose distribution (clinical) within 3 mm that receives a dose that is within ±3% of the reference dose. Table 4.2 summarizes the average GPR achieved over all KBP-generated plans. A score of 1.0 means that every voxel has passed the criteria; in other words, the two dose distributions were considered identical (within the tolerance). Overall, we observe that the GAN plans generate dose distributions that most closely resemble the clinical dose distributions, followed by the CNN, and then the gPCA plans. Notably, the GAN dose distributions best resemble the clinical dose distribution around the target, which is of primary importance. The GAN plans perform poorer on the



(a) BQ − clinical (b) gPCA − clinical (c) RF − clinical

(d) CNN − clinical (e) GAN − clinical (f) GAN − CNN

Figure 4.2: Head-to-head: (a)–(e) the plans from each KBP-generated model versus their clinical counterparts; (f) the plans from the GAN versus the CNN. We mark the 75th and 25th percentiles and the median in the box. Whiskers extend to 1.5 times the interquartile range.

OARs, but this result is expected given the prior results in Table 4.1, which indicate that the GAN plans achieve more OAR clinical criteria than the clinical plan (i.e., the GAN is able to deliver a lower dose to the OARs as compared to the clinical dose distribution).

4.4 The value of GANs for dose generation

Unlike BQ and gPCA, the RF, CNN, and GAN all directly predict the 3-D distribution of dose to each voxel of the patient's geometry. However, there are several differences between these three models that affect their performance. First, the RF is trained via supervised learning to predict the dose to any voxel using ten customized features based on the location of that voxel with respect to the PTVs and OARs. In contrast, the CNN and GAN are deep learning models that learn relevant geoemtric features via the convolution operations in these networks. The size of the networks implies that these models may learn a large number of important features that are more effective than the predetermined features of the RF. Consequently we find that the GAN and CNN

Table 4.2: Average GPR for each population of KBP plans compared to clinical plans.

	BQ	gPCA	RF	CNN	GAN
All OARs	0.548	0.584	0.535	0.566	0.549
All PTVs	0.533	0.728	0.503	0.741	0.761
All Structures	0.536	0.669	0.518	0.670	0.675

outperform the RF on both clinical criteria satisfaction and GPR.

The GAN is further differentiated from the RF and CNN due to the specific Style Transfer loss function. The RF is trained via CART (Breiman, 1996) and thus tries to minimize distance from the ground truth on a per voxel basis. The CNN is trained to minimize l_2 loss, i.e., Mean-Squared Error, which is also a physical distance measure but simply on 2-D axial slices. In contrast, the GAN is trained via an adversarial loss where the discriminator tries to learn the "characteristics" of a clinically desirable plan. This approach is better than conventional supervised learning in our application because our evaluation metric, clinical criteria satisfaction, is not necessarily amenable to physical distance. For example, consider a ground truth clinical plan where every voxel in the PTV56 receives 56 Gy. A plan generated via KBP where the dose to every voxel of the PTV56 is 55.5 Gy will have an average l_2 error of 0.25, but it fails the PTV56 criteria. On the other hand, a generated plan where the dose to every voxel of the PTV56 is 57 Gy will have an average l_2 error of 1, but it would satisfy the PTV56 criteria and therefore be more desirable. We conjecture that a discriminator tries to learn this difference, which then helps train the generator to create clinically acceptable plans.

4.5 Conclusion

In this chapter, we propose the first GAN-based KBP pipeline to generate radiation therapy treatment plans. We train our complete pipeline on 130 patients, test on 87 out-of-sample patients diagnosed with head-and-neck cancer, and compare our technique with several state-of-the-art planning frameworks including a query-based approach, a PCA-based method, a random forest, and a CNN. All methods are evaluated on standard clinical criteria for plan evaluation (i.e., OARs sparing and target coverage), showing that the GAN plans outperform all baseline KBP methods. We also demonstrate that the GAN plans outperformed the clinical plans by satisfying additional criteria on OAR dose sparing and target dose coverage. Finally, we use the gamma passing rate, a standard metric in the radiation therapy literature, to evaluate the similarity of the full 3-D dose distribution between the KBP and clinical plans demonstrating that the GAN plans are

the most similar to clinical plans on average. Note that the performance of automated planning methods should be measured based on their ability to re-create clinical quality plans with minimal manual effort. If the auto-generated plans manage to improve upon clinical plans, that would be a better outcome.

Our approach eschews the classical paradigm of predicting low-dimensional representations, or engineering features, by training a generic neural network to learn desirable dose distributions. Specifically, the GAN recasts KBP prediction as an image colorization problem. Moreover, the GAN is trained by mimicking the iterative process between the dosimetrist and oncologist; the generator network acts as the dosimetrist by designing dose distributions while the discriminator acts as the oncologist by determining whether the plans are good or bad. The implication is that selecting the appropriate neural network architecture may be sufficient when creating an automated KBP pipeline that generates deliverable plans. Further, our approach does not add site-specific feature variables which suggests that the good performance we observe may not be limited to patients with oropharyngeal cancer. Finally, since the GAN plans improve upon the clinical plans, it may be useful to analyze the results to generate useful insights for practitioners.

Chapter 5

Learning to optimize with hidden constraints

Consider a decision-maker who regularly solves instances of a continuous optimization problem. There is a fixed objective and set of constraints, but each instance is also dependent on some auxiliary input that may change the feasible set in a way that cannot be easily characterized. The effect of this input represents the decision-maker's contextual understanding of the problem. For example, a clinician may regularly solve an optimization problem to construct personalized treatments for her patients. However, each patient possesses features which the clinician will take into account. Moreover, this accounting may depend on clinical knowledge, intuition, and anecdotal experiences with prior patients, meaning that it cannot easily be formalized into mathematical representation. In these settings, the relationship between the contextual features and the solution require estimation by analyzing historical data stores. Techniques from operations research and machine learning suggest contrasting approaches to solving these types of problems and each field's advantages reveals the other's deficiencies.

In operations research, the effect of the context is typically first to be estimated. The result is then used as an input to a constrained optimization problem; this is known as the '*predict-then-optimize*' paradigm (Elmachtoub and Grigas, 2017). There are many estimation approaches; each attempts to predict the direct effect of the context on the objective function or the corresponding decisions (e.g., Ban and Rudin, 2018; Bertsimas and Kallus, 2020; Mišić, 2019). For example, a predictive model can estimate the likelihood terms of a conditional stochastic objective with context-dependent features. Such optimization problems can be challenging to solve; the literature primarily explores optimizing over context-dependent objectives involving linear, tree, or neighborhood-based models. However, once a relationship between the contextual features and the optimiza-

tion model is established, the optimization problem can be solved to produce provably optimal solutions.

In machine learning, there has been renewed interest in directly predicting solutions to optimization problems (e.g., Bengio et al., 2018; Hopfield and Tank, 1985). Specifically, deep neural networks can learn a mapping from contexts to decisions, for example, they can estimate an optimal traveling salesman tour given a graph (Kool and Welling, 2018; Vinyals et al., 2015). While inference requires a fraction of the time as compared to solving a mathematical program (Larsen et al., 2018), there are no feasibility or optimality guarantees, especially for out-of-sample instances. Further, these methods rely on a large data set of *optimal* decisions for training, whereas many applications feature smaller data sets of *feasible*, potentially sub-optimal decisions.

Application to radiation therapy

This work is motivated by the dose generation problem in automated RT treatment planning. In the clinical planning procedure (see Figure 1.1), all plans must be approved by an oncologist based on their performance across institutionally mandated criteria. Since it is impossible to simultaneously satisfy all criteria (e.g., tumor dose may be sacrificed to reduce dose to nearby critical structures or vice versa), oncologists make subjective trade-offs by choosing a subset of relevant criteria for a given patient based on prior expertise. These oncologist-driven trade-offs can be viewed as effectively latent constraints that are parameterized by the patient's information and can be learned from examining past decisions (i.e., treatment plans) approved by the oncologist.

Dose generation models estimate dose from CT images (e.g., Chapter 4) use conventional supervised learning techniques rather than incorporating characteristics of the optimization problem within the learning problem. Conventional techniques have several drawbacks. First, they do not take into account the pass-fail nature of clinical criteria, nor any evaluation of delivering "low" dose (i.e., dosage that minimizes the radiation delivered to healthy tissue). Second, the protocols for radiation therapy treatment often vary between institutions (e.g., Geretschläger et al., 2015 versus Babier et al., 2018b). This makes it difficult to deploy the same automated planning pipeline at multiple institutions because off-the-shelf prediction models trained using data from one clinic may not satisfy protocols (e.g., hidden constraints) at other institutions (Wu et al., 2017). A clinic attempting to implement an automated planning pipeline would first need to train a custom prediction model using institution-specific data, which is especially difficult for smaller clinics or those ramping up in developing countries.

Contributions

In this chapter, we combine the best of both machine learning and operations research to learn to generate decisions to contextual optimization problems. From the machine learning perspective, we introduce a predictive approach to generating optimal solutions to constrained optimization problems with context-dependent features; this approach is amenable for use with deep neural networks. From the operations research perspective, we use historical data sets of feasible decisions and ensure that our approach generates decisions that have both in-sample and out-of-sample optimality guarantees and demonstrate how these bounds can be iteratively improved with new data. Specifically, we formulate a linear optimization problem that includes a set of unknown (potentially non-convex), context-dependent constraints. Given a context vector, we estimate the feasible set and generate an optimal solution. To do this, we transform our contextual optimization problem into two prediction problems. The first problem predicts feasibility using a binary classifier that is trained on past context vectors and decisions. The second problem trains a generative model to produce decisions that the classifier would predict as feasible. To augment training, a feasibility oracle, which can be a prior model of the problem or a human decision-maker, labels new decisions as feasible or infeasible in order to achieve better optimality guarantees.

To navigate the region that a classifier deems feasible, we train our generator via an interior point method (IPM). Since IPMs are primarily used for well-defined convex optimization problems, we first derive a new ϵ-optimality guarantee for IPMs when the feasible set of a problem is unknown. As a result, the classifier learns to act as a barrier function within a data-driven IPM while the generator enjoys a related ϵ-optimality guarantee when predicting decisions. Our technical contributions are as follows:

1. We introduce the concept of a δ-*barrier*; a barrier function for a relaxation of the feasible set. We then define a (δ, ϵ)-optimality guarantee for optimization problems where the feasible set can only be partially characterized and generalize key properties of IPM algorithms to this setting. By combining IPMs and adversarial learning, we predict (δ, ϵ)-optimal solutions to problem instances given context-dependent features.

2. We present a new, oracle-guided algorithm—Interior Point Methods with Adversarial Networks (IPMAN)—that progressively predicts tighter (δ, ϵ)-optimal solutions to a constrained optimization problem by iteratively growing the data set.

3. We prove a generalization bound on the out-of-sample (δ, ϵ)-optimality gap for any

model that predicts solutions to an optimization problem. As a result, we characterize the in-sample and out-of-sample optimality gap of the IPMAN algorithm.

We apply IPMAN to predict the dose distribution to be delivered to head-and-neck cancer patients. In this context, we model the clinical criteria that must be met before a treatment is accepted as latent constraints to be learned from historically delivered treatment plans. After the classifier is trained, the generative model produces dose distributions that the classifier predicts will satisfy the relevant clinical criteria for that patient. The oracle labels the generated output as correct if the plan satisfied all of the hidden constraints that the oncologist had determined were relevant for the patient. By incorporating the evaluation of feasibility and optimality in training, our approach extends state-of-the-art generative adversarial network (GAN) frameworks for predicting dose distributions (c.f., Chapter 4). Our final product is a generative model that outputs dose distributions that, with high probability, are guaranteed to be within a neighborhood of optimality (both in-sample and out-of-sample).

In our numerical experiments, we find that the doses predicted by our model better resemble clinical doses compared to current state-of-the-art baselines. We then show that once the latent constraints are learned, they can be altered using IPMAN so that dose distributions can be predicted for institutions with different protocols, without collecting a new institution-specific data set. This result has implications for the transfer of automated treatment planning technology between institutions (Wu et al., 2017), as well as closing the global gap in supply of radiation therapy by enabling all clinics to perform automated planning (Atun et al., 2015).

5.1 Background

Our approach to generating decisions combines methods from several fields. First, we employ two learning models, one to evaluate and one to generate solutions. This is common in reinforcement learning (e.g., actor-critic methods, Konda and Tsitsiklis, 2000) and deep learning (e.g., generative adversarial networks). Our learning function is derived using interior point methods (Nesterov and Nemirovskii, 1994) and our learning guarantees extend Rademacher complexity results for data-driven optimization (Bertsimas and Kallus, 2020). Finally, our learning algorithm bears a loose resemblance to estimation of distribution algorithms (EDAs), commonly used in evolutionary and black-box optimization (Pelikan et al., 2002). We detail the most relavant work below.

5.1.1 Interior point methods

Interior point methods (IPMs) are among the most popular techniques for solving constrained optimization problems (Nesterov and Nemirovskii, 1994). A constrained optimization problem $\min_{\mathbf{x}} \left\{ \mathbf{c}^\mathsf{T}\mathbf{x} \;\middle|\; \mathbf{x} \in \mathcal{P} \right\}$ is transformed into an unconstrained, differentiable problem $\min_{\mathbf{x}}\{\mathbf{c}^\mathsf{T}\mathbf{x} - \lambda \log B(\mathbf{x})\}$, where $\lambda > 0$ is a dual parameter and $B(\mathbf{x})$ is a barrier, i.e., a function that is non-zero when $\mathbf{x}$ is strictly feasible and zero otherwise.

IPMs have been applied to many problems in linear and quadratic optimization where they can quickly converge to optimal solutions (Gondzio, 2012). Recent results on barrier functions for arbitrary convex sets have renewed interest in IPMs (Badenbroek and de Klerk, 2018; Bubeck and Eldan, 2019). IPMs also exist for non-convex optimization problems (Benson et al., 2004; Hinder and Ye, 2018; Vanderbei and Shanno, 1999). These prior papers all assume access to explicit constraints or a barrier that is well-defined for the entire feasible set. Here, we construct a barrier function (i.e., our classifier) that approximates a relaxation of the feasible set and develop an IPM theory for this setting.

5.1.2 Contextual optimization

The most common approach to contextual optimization is the 'predict-then-optimize' paradigm, i.e., to construct a parametric optimization model of the decision-making problem and use machine learning to predict parameters from context-dependent inputs (Angalakudati et al., 2014; Elmachtoub and Grigas, 2017; Ferreira et al., 2015). A non-parametric alternative is to directly estimate the effect of the context in terms of a conditional stochastic optimization model. For example, a stochastic objective that is conditioned on the context vector may be characterized by embedding a machine learning model to estimate probability weights on the objective (Ban and Rudin, 2018; Bertsimas and Kallus, 2020; Bertsimas and McCord, 2018; Hannah et al., 2010; Kao et al., 2009).

Our work is most similar to Ban and Rudin (2018) who use Empirical Risk Minimization (ERM) to construct a predictor for the optimal solution to a Newsvendor problem. The Newsvendor problem is only constrained by the non-negativity of the variables, and thus, we generalize their result to an arbitrary set of restrictions by incorporating constraint satisfaction via a binary classifier. Bertsimas and Kallus (2020) remark on the challenges of constraint satisfaction when using ERM and instead, propose a weighted learning framework that estimates the weights (i.e., conditional probability terms) in a sample-average optimization problem. They prove several generalization results that arise from ERM theory. We extend their generalization bounds to out-of-sample ϵ-optimality guarantees, in particular, for problems where the feasible set is not fully specified.

5.1.3 Deep learning for constrained optimization

Recent advances in deep learning have prompted a growing interest in using neural networks to solve constrained optimization problems (Bengio et al., 2018). Task-specific neural networks are trained by customized learning algorithms to predict feasible and optimal decisions. Different approaches to this objective include supervised learning with a data set of instances and optimal solutions (Larsen et al., 2018; Vinyals et al., 2015), reinforcement learning (Bello et al., 2017; Kool and Welling, 2018), or task-based learning (Donti et al., 2017).

Predicting solutions is faster than using an optimization approach since a prediction model outputs solutions via a single function call. However, a trained model may not guarantee that predicted solutions can satisfy every constraint. There are, however, several approaches that address this issue. For instance, training a supervised learning technique on a large data set may help make predictions more accurate (Larsen et al., 2018). Recently, a class of neural network layers can be used to directly learn the implicit function associated with feasible solutions provided the structure of the constraints are known (Agrawal et al., 2019; Amos and Kolter, 2017). Finally, the loss function may be customized to encourage constraint satisfaction (Donti et al., 2017). Our approach trains a prediction model using a loss function motivated by IPM theory. This encourages the model to produce feasible solutions and allows us to prove optimality guarantees on the generated solutions. In addition, as opposed to previous work in this area, we relax the assumption that the training data must consist of optimal solutions to problem instances with different contexts.

5.2 Problem setup

In this chapter, we use the following standards for notation. Vectors are denoted in bold and sets in calligraphic. The interior, boundary, and closure of a set are $\text{int}(\mathcal{X})$, $\text{bd}(\mathcal{X})$, and $\text{cl}(\mathcal{X})$ respectively. The exclusion of $\mathcal{X}_1$ from $\mathcal{X}_2 \supseteq \mathcal{X}_1$ is denoted $\mathcal{X}_2 \setminus \mathcal{X}_1$. We denote probability distributions with $\mathbb{P}$. The support of a distribution is $\text{supp}(\mathbb{P})$. Finally, we use $\|\cdot\|$ as the l_2-norm.

Let $\mathbf{x} \in \mathbb{R}^n$ denote a decision vector and $\mathbf{u} \in \mathcal{U}$ be a context vector. Consider the problem

$$\mathbf{OP}(\mathbf{u}): \quad \min_{\mathbf{x}} \left\{ \mathbf{c}^\mathsf{T}\mathbf{x} \;\middle|\; \mathbf{x} \in \mathcal{X}(\mathbf{u}),\ \mathbf{x} \in \mathcal{P} \right\},$$

where $\mathbf{c} \in \mathbb{R}^n$ is a cost vector (assumed without loss of generality that $\|\mathbf{c}\| = 1$),

$\mathcal{P} := \{\mathbf{x} \mid \mathbf{a}_m^\mathsf{T}\mathbf{x} \leq b_m, m = 1, \ldots, M\}$ is a known set of linear constraints, and $\mathcal{X}(\mathbf{u})$ is an *unknown* and potentially non-convex set of instance-specific constraints that must be learned. We assume in this paper that the context space $\mathcal{U}$ and feasible sets $\mathcal{X}(\mathbf{u})$ are well-behaved.

Assumption 1 (Compactness and feasibility)**.**

1. *For all $\mathbf{u} \in \mathcal{U}$, the feasible set $\mathcal{X}(\mathbf{u})$ is compact and has a non-empty interior. Further, the joint set $\{(\mathbf{x}, \mathbf{u}) \mid \mathbf{u} \in \mathcal{U},\ \mathbf{x} \in \mathcal{X}(\mathbf{u})\}$ is compact and has a non-empty interior.*

2. *The known constraints $\mathcal{P} = \left\{\mathbf{x} \;\middle|\; \mathbf{a}_m^\mathsf{T}\mathbf{x} \leq b_m,\ m = 1, \ldots, M\right\}$ define a compact polyhedron whose interior bounds the hidden feasible set, i.e., $\mathcal{X}(\mathbf{u}) \subset \text{int}(\mathcal{P})$ for all $\mathbf{u} \in \mathcal{U}$.*

The first statement ensures compactness and feasibility. The second states that the known constraints are relaxations of the hidden set. This is trivially satisfied when $\mathcal{P}$ and $\mathcal{X}(\mathbf{u})$ are compact since one can always re-define a larger $\mathcal{P}$ with no change to $\mathbf{OP}(\mathbf{u})$.

Let $\mathbf{x}^*(\mathbf{u})$ denote an optimal solution to $\mathbf{OP}(\mathbf{u})$. In Section 5.3, we first consider optimizing over a single instance of $\mathbf{OP}(\mathbf{u})$. We introduce a barrier function for a relaxation of the feasible set and show that an optimal solution to the corresponding unconstrained barrier problem satisfies an optimality bound that depends on the size of the relaxation.

In Section 5.4, we address our main problem, which is to construct a generative model $F : \mathcal{U} \to \mathbb{R}^n$, i.e., a function that takes as input a context vector and outputs a decision. Furthermore, decisions produced by this model must satisfy an error bound

$$\left|\mathbf{c}^\mathsf{T}F(\mathbf{u}) - \mathbf{c}^\mathsf{T}\mathbf{x}^*(\mathbf{u})\right| < \epsilon \tag{5.1}$$

for some ϵ. We assume access to a data set of $N_\mathbf{x}$ feasible and $\bar{N}_\mathbf{x}$ infeasible decisions as well as $N_\mathbf{u}$ context-dependent inputs. Due to the complexity of optimally solving $\mathbf{OP}(\mathbf{u})$, we do not assume that the given feasible decisions are optimal. Instead we have:

- A data set of context vectors $\hat{\mathcal{U}} := \{\hat{\mathbf{u}}_i\}_{i=1}^{N_\mathbf{u}}$.

- A training data set of feasible decisions $\mathcal{D} := \{(\hat{\mathbf{x}}_i, \hat{\mathbf{u}}_i)\}_{i=1}^{N_\mathbf{x}}$, where $\hat{\mathbf{u}}_i \in \hat{\mathcal{U}}$ and $\hat{\mathbf{x}}_i \in \mathcal{X}(\hat{\mathbf{u}}_i)$ for all $i \in \{1, \ldots, N_\mathbf{x}\}$. This data may consist of decisions that were implemented in the past. Note that for training, $\mathcal{D}$ may include multiple feasible decisions for each $\hat{\mathbf{u}}_i$.

- A training data set of infeasible decisions $\bar{\mathcal{D}} := \{(\bar{\mathbf{x}}_{\bar{i}}, \hat{\mathbf{u}}_{\bar{i}})\}_{\bar{i}=1}^{\bar{N}_{\mathbf{x}}}$, where $\bar{\mathbf{x}}_{\bar{i}} \in \mathbb{R}^n \setminus \mathcal{X}(\hat{\mathbf{u}}_{\bar{i}})$. This data set may consist of decisions that were not implemented by the decision-maker or generated by random sampling (e.g., **?**).

Some applications may not possess high-quality data with both feasible and infeasible decisions. Thus in Section 5.5, we demonstrate how to train the models using an iterative framework which includes a data augmentation procedure that creates new labelled points. Here, we assume access to an oracle of feasibility $\Psi(\mathbf{x}, \mathbf{u})$, where $\Psi(\mathbf{x}, \mathbf{u}) = 1$ if $\mathbf{x} \in \mathcal{X}(\mathbf{u})$ and 0 otherwise. Oracles have been used for in-the-loop labelling in machine learning, for example with human annotators correcting predictions in real-time during training (Castrejon et al., 2017). In our numerical experiments, we construct data-driven, rule-based oracles defined only over the training set for which we can make a large number of queries. However, the oracles are not available when generating predictions for out-of-sample instances.

5.3 Optimization with a hidden feasible set

In this section, we extend IPM theory to the case where the barrier function can only partially characterize the feasible region. In particular, consider the problem $\mathbf{OP}(\mathbf{u})$ for a single context vector $\mathbf{u}$. Although we do not know the hidden feasible set $\mathcal{X}(\mathbf{u})$, we do know the constraints associated with the relaxation $\mathcal{P}$. Thus, we first define the notion of a barrier function for partially specified feasible sets and then propose a barrier optimization problem that produces a solution $\mathbf{OP}(\mathbf{u})$ that satisfies an optimality guarantee.

If $\mathbf{OP}(\mathbf{u})$ had no hidden constraints then one could construct a canonical log-barrier, i.e., $\log B(\mathbf{x}) = \sum_{m=1}^{M} \log\left(b_m - \mathbf{a}_m^{\mathsf{T}}\mathbf{x}\right)$, to solve $\mathbf{OP}(\mathbf{u})$ within an IPM (see Nesterov and Nemirovskii, 1994). In our case, $\mathcal{X}(\mathbf{u}) \subset \mathcal{P}$ and thus, a canonical barrier using $\mathcal{P}$ may incorrectly return non-zero values for $\mathbf{x} \in \mathcal{P} \setminus \mathcal{X}(\mathbf{u})$. This canonical barrier can be viewed as a barrier function over a relaxation or superset of $\mathcal{X}(\mathbf{u})$. As a result, we define a new class of functions that are strictly positive for all $\mathbf{x} \in \mathcal{X}(\mathbf{u})$ (as in a barrier), and are zero for all $\mathbf{x}$ that are sufficiently far from $\mathcal{X}(\mathbf{u})$. We refer to such functions as δ-barriers.

Definition 1. *For some $\delta > 0$, let $\mathcal{N}_\delta(\mathcal{X}(\mathbf{u})) = \{\mathbf{x} + \boldsymbol{\epsilon} \mid \mathbf{x} \in \mathcal{X}(\mathbf{u}),\ \|\boldsymbol{\epsilon}\| < \delta\}$ be a neighborhood of $\mathcal{X}(\mathbf{u})$. A* δ-barrier $B_\delta : \mathbb{R}^n \times \mathcal{U} \to [0, 1)$ *is a continuous function that satisfies*

$$\mathcal{X}(\mathbf{u}) \subset \left\{\mathbf{x} \;\middle|\; B_\delta(\mathbf{x}, \mathbf{u}) > 0\right\} \subseteq \mathcal{N}_\delta(\mathcal{X}(\mathbf{u})).$$

Note that any function $B(\mathbf{x})$ supported over a superset of $\mathcal{X}(\mathbf{u})$ is a δ-barrier for δ at least equal to the Hausdorff distance $d_H(\cdot,\cdot)$ between $\mathcal{X}(\mathbf{u})$ and the support of $B(\mathbf{x})$, i.e.,

$$\delta \geq d_H\left(\mathcal{X}(\mathbf{u}), \left\{\mathbf{x} \mid B(\mathbf{x}) > 0\right\}\right) = \min_{\xi \geq 0} \left\{\xi \;\middle|\; \left\{\mathbf{x} \mid B(\mathbf{x}) > 0\right\} \subseteq \mathcal{N}_\xi\left(\mathcal{X}(\mathbf{u})\right)\right\}. \tag{5.2}$$

Given the set of known constraints $\mathcal{P}$ and Assumption 1, a δ-barrier for $\mathcal{X}(\mathbf{u})$ can always be constructed simply by re-scaling the canonical barrier for $\mathcal{P}$. In particular, let $C^{\mathcal{P}} > \max_{m\in\{1,\ldots,M\},\mathbf{x}\in\mathcal{P}} \left\{b_m - \mathbf{a}_m^\mathsf{T}\mathbf{x}\right\}$ be a normalization factor and consider the function

$$B^{\mathcal{P}}(\mathbf{x}) := \prod_{m=1}^{M} \left[\frac{b_m - \mathbf{a}_m^\mathsf{T}\mathbf{x}}{C^{\mathcal{P}}}\right]^+. \tag{5.3}$$

This function is a δ-barrier for $\mathcal{X}(\mathbf{u})$ where $\delta = d_H(\mathcal{X}(\mathbf{u}), \mathcal{P})$.

Suppose we have a δ-barrier $B_\delta(\mathbf{x},\mathbf{u})$ for some δ. Let $\lambda > 0$ be a constant corresponding to a Lagrangian dual variable and consider the unconstrained barrier optimization problem

$$\mathbf{BP}(\mathbf{u}, B_\delta, \lambda): \quad \min_{\mathbf{x}} \left\{\mathbf{c}^\mathsf{T}\mathbf{x} - \lambda \log B_\delta(\mathbf{x},\mathbf{u})\right\}. \tag{5.4}$$

We show that the optimal value of $\mathbf{OP}(\mathbf{u})$ is bounded by the optimal value of $\mathbf{BP}(\mathbf{u}, B_\delta, \lambda)$.

Theorem 6. *For any $\lambda > 0$, $\mathbf{BP}(\mathbf{u}, B_\delta, \lambda)$ has an optimal solution $\mathbf{x}^\lambda(\mathbf{u})$. Furthermore, this solution is (δ,ϵ)-optimal for $\mathbf{OP}(\mathbf{u})$:*

$$\mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\mathbf{u}) - \epsilon < \mathbf{c}^\mathsf{T}\mathbf{x}^*(\mathbf{u}) < \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\mathbf{u}) + \delta, \tag{5.5}$$

where $\epsilon = C\lambda$ with C being a positive constant.

Proof. For notational simplicity, we use $\mathbf{x}^\lambda$ amd $\mathbf{x}^*$ in place of $\mathbf{x}^\lambda(\mathbf{u})$ and $\mathbf{x}^*(\mathbf{u})$, respectively. We first prove that an optimal solution exists for any $\lambda > 0$. From the Weierstrauss Theorem, an optimal solution to $\mathbf{BP}(\mathbf{u}, B_\delta, \lambda)$ exists if there is a sub-level set of the objective that is non-empty, bounded, and closed. Let $\bar{\mathbf{x}}$ be a point in the interior of $\mathcal{X}(\mathbf{u})$; by Assumption 1, $\bar{\mathbf{x}}$ exists. Furthermore, $B_\delta(\mathbf{x},\mathbf{u}) > 0$ for all $\mathbf{x} \in \mathcal{X}(\mathbf{u})$ implies that $\bar{\mathbf{x}}$ admits a finite objective for $\mathbf{BP}(\mathbf{u}, B_\delta, \lambda)$. Moreover, the objective for $\mathbf{BP}(\mathbf{u}, B_\delta, \lambda)$ is only finite within $\{\mathbf{x} \mid B_\delta(\mathbf{x},\mathbf{u}) > 0\} \subseteq \mathcal{N}_\delta\left(\mathcal{X}(\mathbf{u})\right)$. Consequently the sub-level set

$$\left\{\mathbf{x} \;\middle|\; \mathbf{c}^\mathsf{T}\mathbf{x} - \lambda \log B_\delta(\mathbf{x},\mathbf{u}) \leq \mathbf{c}^\mathsf{T}\bar{\mathbf{x}} - \lambda \log B_\delta(\bar{\mathbf{x}},\mathbf{u})\right\} \subseteq \mathcal{N}_\delta\left(\mathcal{X}(\mathbf{u})\right)$$

is non-empty, bounded, and closed (since the objective is continuous) and $\mathbf{x}^\lambda(\mathbf{u})$ exists.

Suppose we choose $\epsilon = -\lambda \log B_\delta(\mathbf{x}^*, \mathbf{u}))$. By definition, $0 < B_\delta(\mathbf{x}^*, \mathbf{u}) < 1$, meaning $C := -\log B_\delta(\mathbf{x}^*, \mathbf{u}) > 0$ is a positive constant. Let $\mathbf{x}^\lambda$ be an optimal solution to $\mathbf{BP}(\mathbf{u}, B_\delta, \lambda)$. We now prove that $\mathbf{c}^\mathsf{T}\mathbf{x}^\lambda - \epsilon < \mathbf{c}^\mathsf{T}\mathbf{x}^*$:

$$\begin{aligned}
\mathbf{c}^\mathsf{T}\mathbf{x}^* + \epsilon &= \mathbf{c}^\mathsf{T}\mathbf{x}^* - \lambda \log B_\delta\left(\mathbf{x}^*, \mathbf{u}\right) \\
&\geq \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda - \lambda \log B_\delta\left(\mathbf{x}^\lambda, \mathbf{u}\right) \\
&> \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda.
\end{aligned}$$

The first inequality follows from the optimality of $\mathbf{x}^\lambda$ for $\mathbf{BP}(\mathbf{u}, B_\delta, \lambda)$ while the second inequality follows from $\log B_\delta(\mathbf{x}^\lambda, \mathbf{u}) < 0$, meaning that $-\lambda \log B_\delta(\mathbf{x}^\lambda, \mathbf{u}) > 0$. Moving ϵ to the right-hand-side gives the lower bound.

The proof of $\mathbf{c}^\mathsf{T}\mathbf{x}^* < \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda + \delta$ has two cases. If $\mathbf{x}^\lambda \in \mathcal{X}(\mathbf{u})$, then by the optimality of $\mathbf{x}^*$, we trivially satisfy $\mathbf{c}^\mathsf{T}\mathbf{x}^* \leq \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda < \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda + \delta$. If $\mathbf{x}^\lambda \in \mathcal{N}_\delta\left(\mathcal{X}(\mathbf{u})\right) \setminus \mathcal{X}(\mathbf{u})$, then let $\tilde{\mathbf{x}} \in \arg\min_{\mathbf{x} \in \mathcal{X}(\mathbf{u})} \left\|\mathbf{x}^\lambda - \mathbf{x}\right\|$ be the projection of $\mathbf{x}^\lambda$ on $\mathcal{X}(\mathbf{u})$. Then,

$$\begin{aligned}
\mathbf{c}^\mathsf{T}\mathbf{x}^* - \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda &\leq \mathbf{c}^\mathsf{T}\tilde{\mathbf{x}} - \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda \\
&\leq \left|\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}} - \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda\right| \\
&\leq \left\|\tilde{\mathbf{x}} - \mathbf{x}^\lambda\right\| \\
&< \delta.
\end{aligned}$$

The first inequality follows from the optimality of $\mathbf{x}^*$ over $\tilde{\mathbf{x}}$ for $\mathbf{OP}(\mathbf{u})$. The third inequality follows from the Cauchy-Schwartz inequality and $\|\mathbf{c}\| = 1$, while the fourth inequality follows from the fact that for $\mathbf{x}^\lambda \in \mathcal{N}_\delta\left(\mathcal{X}(\mathbf{u})\right)$, there exists $\mathbf{x} \in \mathcal{X}(\mathbf{u})$ such that $\left\|\mathbf{x} - \mathbf{x}^\lambda\right\| < \delta$ and that $\tilde{\mathbf{x}}$ minimizes this distance. This proves the upper bound. □

The (δ, ϵ)-optimality inequalities from Theorem 6 generalize the classical ϵ-optimality bound of IPMs (Nesterov and Nemirovskii, 1994) to problems where the feasible set can only be partially characterized. That is, when $\delta = 0$, (5.5) reduces to the classical ϵ-optimality bound $\mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\mathbf{u}) - \epsilon < \mathbf{c}^\mathsf{T}\mathbf{x}^*(\mathbf{u}) < \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\mathbf{u})$ which applies when the feasible set can be fully characterized. Further, similar to classical IPMs, the (δ, ϵ)-optimality of solutions to $\mathbf{BP}(\mathbf{u}, B_\delta, \lambda)$ can be controlled by tuning λ. Because $\epsilon = C\lambda$ for a fixed C, as λ goes to 0, so does ϵ. However, in contrast to classical theory on IPMs, δ is a property of the barrier function and does not change with λ. Intuitively, we may expect that when λ is large (i.e., ϵ is also large), then $\mathbf{x}^\lambda(\mathbf{u})$ may also have large objective function value and be sub-optimal for $\mathbf{OP}(\mathbf{u})$ and that when λ is small (as is ϵ), $\mathbf{x}^\lambda(\mathbf{u})$

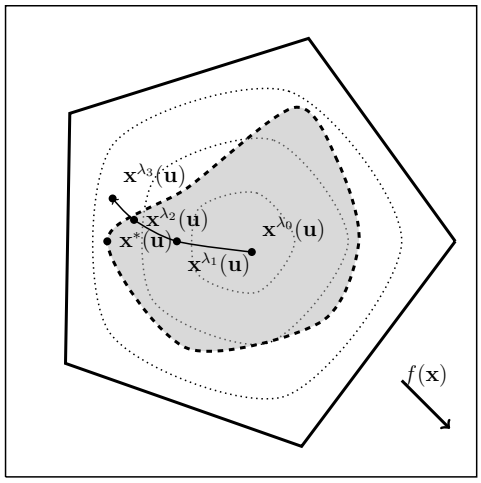


Figure 5.1: The bold shape is $\mathcal{P}$ and the filled region is $\mathcal{X}(\mathbf{u})$. The dotted lines represent level sets for $\log B^{\mathcal{P}}(\mathbf{x})$. An optimal solution to $\mathbf{OP}(\mathbf{u})$ is $\mathbf{x}^*(\mathbf{u})$.

may have small objective function value and be infeasible for $\mathbf{OP}(\mathbf{u})$. We show in the Electronic Companion C.1 that under certain regularity assumptions on the δ-barrier, we can guarantee $\mathbf{x}^\lambda(\mathbf{u}) \in \mathcal{X}(\mathbf{u})$ is feasible for a sufficiently large λ and that as λ decreases, $\mathbf{x}^\lambda(\mathbf{u})$ eventually becomes infeasible. Figure 5.1 shows a sample sequence of decreasing λ values and the corresponding solutions $\mathbf{x}^\lambda(\mathbf{u})$. Thus, we demonstrate that an interior point algorithm using δ-barriers can be developed in much the same way as IPMs that use canonical barrier functions.

5.4 Learning to optimize with hidden constraints

For a constrained optimization problem with a hidden feasible set, Section 5.3 demonstrates that we can solve a barrier optimization problem and obtain a (δ, ϵ)-optimal solution. However, the theory rests on two strong assumptions: (i) that we have access to a δ-barrier; and (ii) that we optimize $\mathbf{OP}(\mathbf{u})$ over a single context vector $\mathbf{u}$. In this section, we relax both assumptions. In particular, we assume that a δ-barrier must be constructed from data and learn to predict (δ, ϵ)-optimal solutions to $\mathbf{OP}(\mathbf{u})$ for any context vector $\mathbf{u}$ as input. To this end, we first construct a function that serves as a δ-barrier for any $\mathbf{u}$ by training a classifier using a data set of feasible ($\mathcal{D}$) and infeasible decisions ($\bar{\mathcal{D}}$). This classifier takes as input a decision $\hat{\mathbf{x}}$ and a context vector $\mathbf{u}$ and outputs 1 when it predicts a decision to be feasible and 0 otherwise. Then, to predict solutions to $\mathbf{OP}(\mathbf{u})$, we adversarially train a generative model, or generator, using the previously developed classifier as a δ-barrier.

Let $\mathcal{B} := \{B : \mathbb{R}^n \times \mathcal{U} \to [0, 1]\}$ denote the model class of classifiers and let $\mathcal{F} := \{F :$

$\mathcal{U} \to \mathbb{R}^n\}$ denote the model class of generators. We assume both models are continuous in their parameters and in the decision vector $\mathbf{x}$. The two learning tasks are defined below:

1. **Learning a δ-barrier from classification:** Train $B \in \mathcal{B}$ to label $(\hat{\mathbf{x}}_i, \hat{\mathbf{u}}_i) \in \mathcal{D}$ as feasible and $(\hat{\mathbf{x}}_{\bar{i}}, \hat{\mathbf{u}}_{\bar{i}}) \in \bar{\mathcal{D}}$ as infeasible via a *Feasibility Classification Problem*:

$$\mathbf{FCP}(\mathcal{D}, \bar{\mathcal{D}}): \quad \max_{B \in \mathcal{B}} \left\{ \frac{1}{N_{\mathbf{x}}} \sum_{i=1}^{N_{\mathbf{x}}} \log B(\hat{\mathbf{x}}_i, \hat{\mathbf{u}}_i) + \frac{1}{\bar{N}_{\mathbf{x}}} \sum_{\bar{i}=1}^{\bar{N}_{\mathbf{x}}} \log\left(1 - B(\bar{\mathbf{x}}_{\bar{i}}, \hat{\mathbf{u}}_{\bar{i}})\right) \right\}. \quad (5.6)$$

   Let $B^*(\mathbf{x}, \mathbf{u})$ be a classifier trained by minimizing $\mathbf{FCP}(\mathcal{D}, \bar{\mathcal{D}})$.

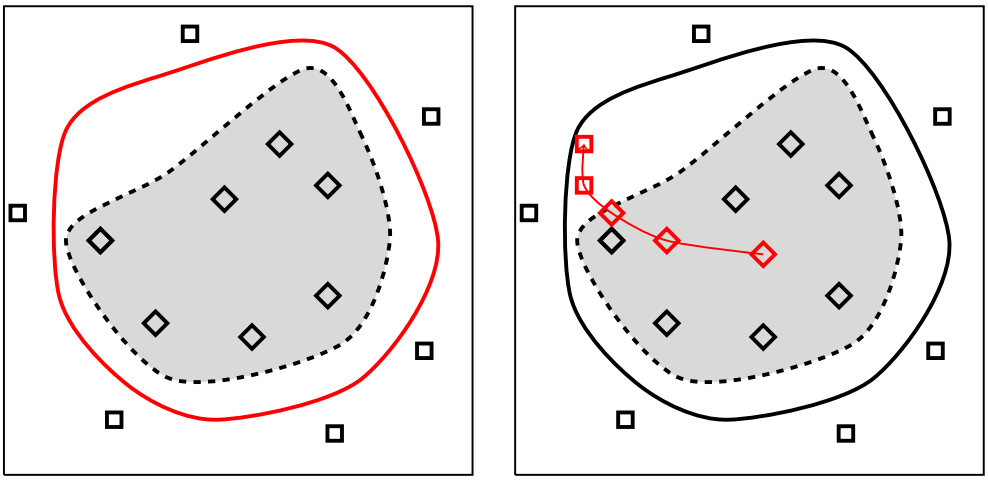
2. **Learning to generate solutions to the barrier problem:** For $j \in \{1, \ldots, J\}$ steps, let λ_j denote a decreasing dual parameter (i.e., $\lambda_{j+1} < \lambda_j$). We train the generator over context vectors $\hat{\mathcal{U}}$ via the *Generative Barrier Problem*:

$$\mathbf{GBP}(\hat{\mathcal{U}}, B^*, \lambda_j): \min_{F \in \mathcal{F}} \left\{ \frac{1}{N_{\mathbf{u}}} \sum_{i=1}^{N_{\mathbf{u}}} \mathbf{c}^\mathsf{T} F(\hat{\mathbf{u}}_i) - \lambda_j \log B^*\big(F(\hat{\mathbf{u}}_i), \hat{\mathbf{u}}_i\big) - \lambda_j \log B^{\mathcal{P}}\big(F(\hat{\mathbf{u}}_i)\big) \right\}, \quad (5.7)$$

   where $B^*(\mathbf{x}, \mathbf{u})$ is a trained classifier and $B^{\mathcal{P}}(\mathbf{x})$ is the canonical barrier from (5.3). For each j, let $F^{(j)}(\mathbf{u})$ be a generator trained by minimizing $\mathbf{GBP}(\hat{\mathcal{U}}, B^*, \lambda_j)$.

In Figure 5.2, we visualize the training procedure for a single context vector $\hat{\mathbf{u}}_i$. In the left-hand figure, a classification model uses points in $\mathcal{D}$ and $\bar{\mathcal{D}}$ to train a barrier function to distinguish between feasible and infeasible decisions. In the right-hand figure, the barrier function is used to train a sequence of generative models with decreasing λ_j. That is, we solve $\mathbf{GBP}(\hat{\mathcal{U}}, B^*, \lambda_j)$ for J steps to obtain a set of trained generators $\big\{F^{(1)}(\mathbf{u}), \ldots, F^{(J)}(\mathbf{u})\big\}$; there is one generator for each value of λ_j. As in classical IPM theory, the sequence of generators produce solutions that are sub-optimal but feasible for large values of λ_j and infeasible for small values of λ_j (see C.1 for conditions the guarantee strict convergence). After training has completed, we select a particular generator $F^{(j^*)}(\mathbf{u})$ by cross-validating the feasibility of decisions outputted by the model with a decision-maker or oracle of feasibility (c.f. Section 5.5) for a held-out set of context vectors. That is, we select the model $F^{(j^*)}(\mathbf{u})$ that most often predicts feasible decisions while still ensuring a suitably low objective function value. Thus, λ_j effectively acts as a tunable regularization parameter.

Given enough training data, the classifier will learn to become a sufficiently strong δ-barrier. Furthermore, the Generative Barrier Problem has a bounded support which



(a) After B^* is trained, the support of the classifier (in red) approximates a δ-barrier.

(b) The points in the line represent the set of predictions $F^{(j)}(\hat{\mathbf{u}}_i)$ for j.

Figure 5.2: The two learning problems for a single $\hat{\mathbf{u}}_i$ and corresponding $\mathbf{OP}(\hat{\mathbf{u}}_i)$. $\Diamond$ and $\square$ represent points in $\mathcal{D}$ and $\bar{\mathcal{D}}$, respectively. The filled region is $\mathcal{X}(\hat{\mathbf{u}}_i)$. The solid line shows the support of $B(\mathbf{x}, \hat{\mathbf{u}}_i)$.

ensures that the final generator, which is trained by minimizing the empirical risk, outputs in-sample predictions that satisfy a (δ, ϵ)-optimality bound with respect to $\mathbf{OP}(\mathbf{u})$. Thus, after solving these two learning tasks, we can use the trained generator to take as input a context vector $\mathbf{u}$ and predict a (δ, ϵ)-optimal decision for $\mathbf{OP}(\mathbf{u})$.

5.4.1 Learning a δ-barrier using classification

A classifier learns from decision data to output larger values when $\mathbf{x} \in \mathcal{X}(\mathbf{u})$ and smaller values when $\mathbf{x} \in \mathbb{R}^n \setminus \mathcal{X}(\mathbf{u})$. If the classifier can output positive values for all $\mathbf{x} \in \mathcal{X}(\mathbf{u})$, and zero for all $\mathbf{x}$ sufficiently far from $\mathcal{X}(\mathbf{u})$, then it is effectively a δ-barrier. By minimizing a Binary Cross Entropy (BCE) loss function, the Feasibility Classification Problem in (5.6) attempts to train a classifier with this behavior. While nearly any learning algorithm and model class can be trained to approximate a δ-barrier, minimizing BCE is easy-to-implement with deep learning models because it is differentiable and can be customized to most tasks by incorporating application-specific terms (Goodfellow et al., 2016).

We now provide a *population level* argument for choosing to minimize BCE. Let $\mathbb{P}_{(\mathbf{x},\mathbf{u})}$ be a distribution of feasible pairs, i.e., $(\mathbf{x}, \mathbf{u})$ where $\mathbf{x} \in \mathcal{X}(\mathbf{u})$, and let $\bar{\mathbb{P}}_{(\mathbf{x},\mathbf{u})}$ be a distribution of infeasible pairs, i.e., $(\mathbf{x}, \mathbf{u})$ where $\mathbf{x} \notin \mathcal{X}(\mathbf{u})$. These pairs can correspond to empirical data distributions or any other arbitrary probabilities. Then, consider the

stochastic optimization variant of the Feasibility Classification Problem:

$$\mathbf{S}\text{–}\mathbf{FCP}(\mathbb{P}_{(\mathbf{x},\mathbf{u})}, \bar{\mathbb{P}}_{(\mathbf{x},\mathbf{u})}) : \max_{B \in \mathcal{B}} \left\{ \mathbb{E}_{\mathbf{x},\mathbf{u} \sim \mathbb{P}_{(\mathbf{x},\mathbf{u})}} \left[\log B(\mathbf{x}, \mathbf{u}) \right] + \mathbb{E}_{\mathbf{x},\mathbf{u} \sim \bar{\mathbb{P}}_{(\mathbf{x},\mathbf{u})}} \left[\log \left(1 - B(\mathbf{x}, \mathbf{u}) \right) \right] \right\}.$$

Lemma 2. *Assume that $\mathbb{P}_{(\mathbf{x},\mathbf{u})}$ and $\bar{\mathbb{P}}_{(\mathbf{x},\mathbf{u})}$ have closed and disjoint supports and that $\mathcal{B}$ contains all continuous functions mapping $\mathbb{R}^n \times \mathcal{U} \to [0, 1]$.*

1. *There exists a continuous function $B^*(\mathbf{x}, \mathbf{u})$ of $\mathbf{S}\text{–}\mathbf{FCP}(\mathbb{P}_{(\mathbf{x},\mathbf{u})}, \bar{\mathbb{P}}_{(\mathbf{x},\mathbf{u})})$ where $B^*(\mathbf{x}, \mathbf{u}) = 1$ for all $(\mathbf{x}, \mathbf{u}) \in \operatorname{supp}(\mathbb{P}_{(\mathbf{x},\mathbf{u})})$ and $B^*(\mathbf{x}, \mathbf{u}) = 0$ for all $(\mathbf{x}, \mathbf{u}) \in \operatorname{supp}(\bar{\mathbb{P}}_{(\mathbf{x},\mathbf{u})})$ that achieves an optimal value of 0 for $\mathbf{S}\text{–}\mathbf{FCP}(\mathbb{P}_{(\mathbf{x},\mathbf{u})}, \bar{\mathbb{P}}_{(\mathbf{x},\mathbf{u})})$.*

2. *If $\operatorname{supp}(\mathbb{P}_{(\mathbf{x},\mathbf{u})}) = \{(\mathbf{x}, \mathbf{u}) \mid \mathbf{x} \in \mathcal{X}(\mathbf{u}), \mathbf{u} \in \mathcal{U}\}$, then for any $\mathbf{u} \in \mathcal{U}$, the product $B^*(\mathbf{x}, \mathbf{u})B^{\mathcal{P}}(\mathbf{x})$ is a δ-barrier with $\delta = d_H(\mathcal{X}(\mathbf{u}), \mathcal{P})$.*

Proof. This result is an application of Urysohn's Smooth Lemma, which states that given two closed and disjoint sets $\mathcal{A}$ and $\mathcal{A}'$, there exists a continuous function $f(\cdot) \in [0, 1]$ for which $f(\mathcal{A}) = 1$ and $f(\mathcal{A}') = 0$ (Engelking, 1977). Letting $\mathcal{A} = \operatorname{supp}(\mathbb{P}_{(\mathbf{x},\mathbf{u})})$, and $\mathcal{A}' = \operatorname{supp}(\bar{\mathbb{P}}_{(\mathbf{x},\mathbf{u})})$ means there is a continuous function $B^*(\mathbf{x}, \mathbf{u})$ that satisfies $B^*(\mathbf{x}, \mathbf{u}) = 1$ for all $(\mathbf{x}, \mathbf{u}) \in \operatorname{supp}(\mathbb{P}_{(\mathbf{x},\mathbf{u})})$ and $B^*(\mathbf{x}, \mathbf{u}) = 0$ for all $(\mathbf{x}, \mathbf{u}) \in \operatorname{supp} \bar{\mathbb{P}}_{(\mathbf{x},\mathbf{u})}$. To prove that $B^*(\mathbf{x}, \mathbf{u})$ is a maximum, note that every $B \in \mathcal{B}$ satisfies $B(\mathbf{x}, \mathbf{u}) \in [0, 1]$, meaning that 0 is an upper bound on the optimal value. Substituting $B^*(\mathbf{x}, \mathbf{u})$ into the objective of $\mathbf{S}\text{–}\mathbf{FCP}(\mathbb{P}_{(\mathbf{x},\mathbf{u})}, \bar{\mathbb{P}}_{(\mathbf{x},\mathbf{u})})$ achieves this value.

To prove Statement 2, consider a fixed $\mathbf{u}$. First note that $B^*(\mathbf{x}, \mathbf{u}) = 1$ for all $\mathbf{x} \in \mathcal{X}(\mathbf{u})$. Because $B^*(\mathbf{x}, \mathbf{u})$ is continuous, we must have $\mathcal{X}(\mathbf{u}) \subset \{\mathbf{x} \mid B^*(\mathbf{x}, \mathbf{u}) > 0\}$. Then, note that $B^{\mathcal{P}}(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathbb{R}^n \setminus \mathcal{P}$. Consequently,

$$\{\mathbf{x} \mid B^*(\mathbf{x}, \mathbf{u})B^{\mathcal{P}}(\mathbf{x}) > 0\} \subseteq \left\{\mathbf{x} \mid B^{\mathcal{P}}(\mathbf{x}) > 0\right\} = \mathcal{P} \subseteq \mathcal{N}_{d_H(\mathcal{X}(\mathbf{u}),\mathcal{P})}\left(\mathcal{X}(\mathbf{u})\right).$$

Finally, although $B^*(\mathbf{x}, \mathbf{u}) \in [0, 1]$ rather than in $[0, 1)$ as per the definition of a δ-barrier, this can be rectified by simply scaling the classifier by a multiplicative factor. □

The first statement Lemma 2 is a variation of a result by Arjovsky and Bottou (2017, Theorem 2.1), which demonstrates that an optimal continuous function binary classifier trained by minimizing BCE will equal to 1 and 0 over the distributions for each respective class. The second statement indicates that a product of classifiers is a δ-barrier when trained with a probability distribution over feasible decisions. Although the lemma is stated in terms of population-level probability distributions, the assumption of closed and disjoint supports is satisfied with empirical data distributions and thus, it holds without loss of generality. Further, Lemma 2 assumes that the model class $\mathcal{B}$ contains

all continuous mappings of $\mathbb{R}^n \times \mathcal{U} \to [0,1]$. This assumption is a sufficient rather than necessary condition. Since large neural networks can approximate any continuous function classifier (see Hornik 1991), we find that in practice, designing a neural network model of appropriate size and architecture produces a classifier of sufficient quality for our prediction tasks.

5.4.2 Learning to generate solutions to the barrier problem

We train a generator $F^{(j)}(\mathbf{u})$ to solve $\mathbf{GBP}(\hat{\mathcal{U}}, B^*, \lambda_j)$ (an empirical risk minimization of the barrier problem presented in Section 5.3) for a decreasing sequence of dual regularizers $\lambda_j > 0$. In this subsection, we show that training a generator via $\mathbf{GBP}(\hat{\mathcal{U}}, B^*, \lambda_j)$ ensures that for all in-sample context vectors $\hat{\mathbf{u}}_i \in \hat{\mathcal{U}}$, the error between the solution generated by $F^{(j)}(\hat{\mathbf{u}}_i)$ and the optimal value of $\mathbf{OP}(\hat{\mathbf{u}}_i)$ is bounded. In particular, the generative model outputs decisions satisfying the optimality guarantee given by (5.1).

Recall that $\mathbf{GBP}(\hat{\mathcal{U}}, B^*, \lambda_j)$ has two barrier terms consisting of the classifier and the known constraints. Let $B^{*,\mathcal{P}}(\mathbf{x}, \mathbf{u}) := B^*(\mathbf{x}, \mathbf{u})B^{\mathcal{P}}(\mathbf{x})$ be the product of the two barriers as described in Lemma 2. We simplify (5.7) to formulate the single-barrier objective function

$$\min_{F \in \mathcal{F}} \left\{ \frac{1}{N_{\mathbf{u}}} \sum_{i=1}^{N_{\mathbf{u}}} \mathbf{c}^\mathsf{T} F(\hat{\mathbf{u}}_i) - \lambda_j \log B^{*,\mathcal{P}}\big(F(\hat{\mathbf{u}}_i), \hat{\mathbf{u}}_i\big) \right\}.$$

From Lemma 2, given sufficient training data, the product barrier will be a δ-barrier. Furthermore, since $\mathcal{P}$ is compact under Assumption 1, $B^{*,\mathcal{P}}(\mathbf{x}, \mathbf{u})$ will always have bounded support. This property ensures that all solutions satisfy an in-sample optimality bound.

Theorem 7. *For $\lambda_j > 0$, let $F^{(j)}(\mathbf{x}, \mathbf{u})$ and $B^{*,\mathcal{P}}(\mathbf{x}, \mathbf{u}) = B^*(\mathbf{x}, \mathbf{u})B^{\mathcal{P}}(\mathbf{x})$ be the trained generator and product barrier, respectively. For any $\hat{\mathbf{u}}_i \in \hat{\mathcal{U}}$, let $\mathbf{x}^{\lambda_j}(\hat{\mathbf{u}}_i)$ be an optimal solution to $\mathbf{BP}(\hat{\mathbf{u}}_i, B^{*,\mathcal{P}}, \lambda_j)$. Then, there exists $\delta, \epsilon > 0$ such that*

$$\left|\mathbf{c}^\mathsf{T} F^{(j)}(\hat{\mathbf{u}}_i) - \mathbf{c}^\mathsf{T} \mathbf{x}^*(\hat{\mathbf{u}}_i)\right| < \left|\mathbf{c}^\mathsf{T} F^{(j)}(\hat{\mathbf{u}}_i) - \mathbf{c}^\mathsf{T} \mathbf{x}^{\lambda_j}(\hat{\mathbf{u}}_i)\right| + \max(\delta, \epsilon). \tag{5.8}$$

Proof. Using the same argument as in Theorem 6, as long as $B^*(\mathbf{x}, \mathbf{u})$ is a continuous function, $\mathbf{x}^{\lambda_j}(\hat{\mathbf{u}}_i)$ will exist. For notational simplicity, we use $\mathbf{x}^*$ and $\mathbf{x}^{\lambda_j}$ in place of $\mathbf{x}^*(\hat{\mathbf{u}}_i)$ and $\mathbf{x}^{\lambda_j}(\hat{\mathbf{u}}_i)$, respectively. By the Triangle inequality,

$$\left|\mathbf{c}^\mathsf{T} F^{(j)}(\hat{\mathbf{u}}_i) - \mathbf{c}^\mathsf{T} \mathbf{x}^*\right| \leq \left|\mathbf{c}^\mathsf{T} F^{(j)}(\hat{\mathbf{u}}_i) - \mathbf{c}^\mathsf{T} \mathbf{x}^{\lambda_j}\right| + \left|\mathbf{c}^\mathsf{T} \mathbf{x}^{\lambda_j} - \mathbf{c}^\mathsf{T} \mathbf{x}^*\right|.$$

Even if $B^{*,\mathcal{P}}(\mathbf{x}, \hat{\mathbf{u}}_i)$ is not a δ-barrier for $\mathbf{OP}(\hat{\mathbf{u}}_i)$, we will prove that $\mathbf{x}^\lambda$ is an optimal

solution to a barrier problem $\mathbf{BP}(\hat{\mathbf{u}}_i, B_\delta, \lambda_j)$ for some δ-barrier, and therefore, satisfies a (δ, ϵ)-optimality bound $|\mathbf{c}^\mathsf{T}\mathbf{x}^{\lambda_j} - \mathbf{c}^\mathsf{T}\mathbf{x}^*| < \max(\delta, \epsilon)$ for some $\delta > 0$ and $\epsilon > 0$. We split the proof into two separate cases: when $B^{*,\mathcal{P}}(\mathbf{x}^*, \hat{\mathbf{u}}_i) > 0$ and when $B^{*,\mathcal{P}}(\mathbf{x}^*, \hat{\mathbf{u}}_i) = 0$.

First, if $B^{*,\mathcal{P}}(\mathbf{x}^*, \hat{\mathbf{u}}_i) > 0$, we construct a new constrained problem for which $\mathbf{x}^*$ is an optimal solution and show that $B^{*,\mathcal{P}}(\mathbf{x}, \hat{\mathbf{u}}_i)$ is a δ-barrier for the new problem. Let $0 < \varepsilon \leq B^{*,\mathcal{P}}(\mathbf{x}^*, \hat{\mathbf{u}}_i)$ be any sufficiently small parameter value and consider the optimization problem with a smaller feasible set than $\mathcal{X}(\hat{\mathbf{u}}_i)$, below:

$$\min \ \left\{ \mathbf{c}^\mathsf{T}\mathbf{x} \;\middle|\; \mathbf{x} \in \mathcal{X}(\hat{\mathbf{u}}_i),\ B^{*,\mathcal{P}}(\mathbf{x}, \hat{\mathbf{u}}_i) \geq \varepsilon \right\} \tag{5.9}$$

and note that because $\mathbf{x}^*$ is feasible for (5.9), it is optimal. Further, $\{\mathbf{x} \mid B^{*,\mathcal{P}}(\mathbf{x}, \hat{\mathbf{u}}_i) > 0\}$ is a superset of the feasible set of (5.9), meaning that $B^{*,\mathcal{P}}(\mathbf{x}, \hat{\mathbf{u}}_i)$ is a δ-barrier for the above problem for some $\delta > 0$. Then, from Theorem 6, $\mathbf{x}^{\lambda_j}$, which is an optimal solution to $\mathbf{BP}(\hat{\mathbf{u}}_i, B^{*,\mathcal{P}}, \lambda_j)$, is (δ, ϵ)-optimal for (5.9) with $\epsilon = -\lambda \log B^{*,\mathcal{P}}(\mathbf{x}^*, \hat{\mathbf{u}}_i) > 0$.

Second, if $B^{*,\mathcal{P}}(\mathbf{x}^*, \hat{\mathbf{u}}_i) = 0$, then $\mathbf{x}^*$ does not lie in the support of the classifier. Instead, we construct a "test" δ-barrier $B^{\text{Test}}(\mathbf{x})$ for $\mathcal{X}(\hat{\mathbf{u}}_i)$ and show that $\mathbf{x}^{\lambda_j}$ is also an optimal solution to $\mathbf{BP}(\hat{\mathbf{u}}_i, B^{\text{Test}}, \lambda_j)$, meaning that it satisfies an alternative (δ, ϵ)-optimality bound. Let $\mathbf{x}^{\mathcal{P}} \in \arg\min_{\mathbf{x}}\{\mathbf{c}^\mathsf{T}\mathbf{x} \mid \mathbf{x} \in \mathcal{P}\}$ and let $\bar{B}$ be a constant defined as follows:

$$\bar{B} = B^{*,\mathcal{P}}(\mathbf{x}^{\lambda_j}, \hat{\mathbf{u}}_i) \min\left\{1\ ,\ \exp\left[-\frac{1}{\lambda_j}\left(\mathbf{c}^\mathsf{T}\mathbf{x}^{\lambda_j} - \mathbf{c}^\mathsf{T}\mathbf{x}^{\mathcal{P}}\right)\right]\right\}. \tag{5.10}$$

Note that $\bar{B} \in (0, 1)$. We now define $B^{\text{Test}}(\mathbf{x})$ as a continuous function that satisfies:

$$B^{\text{Test}}(\mathbf{x}) = \begin{cases} B^{*,\mathcal{P}}(\mathbf{x}, \hat{\mathbf{u}}_i), & \forall \mathbf{x} \in \{\mathbf{x} \mid B^{*,\mathcal{P}}(\mathbf{x}, \hat{\mathbf{u}}_i) \geq \bar{B}\} \\ \bar{B}, & \forall \mathbf{x} \in \{\mathbf{x} \mid B^{*,\mathcal{P}}(\mathbf{x}, \hat{\mathbf{u}}_i) < \bar{B},\ \mathbf{x} \in \mathcal{X}(\hat{\mathbf{u}}_i)\} \\ \leq \bar{B}, & \forall \mathbf{x} \in \{\mathbf{x} \mid B^{*,\mathcal{P}}(\mathbf{x}, \hat{\mathbf{u}}_i) < \bar{B},\ \mathbf{x} \in \mathcal{P} \setminus \mathcal{X}(\hat{\mathbf{u}}_i)\} \\ 0, & \forall \mathbf{x} \in \mathbb{R}^n \setminus \mathcal{P}. \end{cases}$$

The third condition is left as an inequality since we only need a continuous function $B^{\text{Test}}(\mathbf{x})$ to satisfy an inequality in that range. We argue that such a function must exist. Because $B^{*,\mathcal{P}}(\mathbf{x}, \hat{\mathbf{u}}_i)$ is continuous in $\mathbf{x}$, the first region $\{\mathbf{x} \mid B^{\text{Test}}(\mathbf{x}) \geq \bar{B}\}$ is closed. Furthermore because the the support of $B^{*,\mathcal{P}}(\mathbf{x}, \hat{\mathbf{u}}_i)$ is a subset of $\mathcal{P}$, the first region is disjoint from $\mathbb{R}^n \setminus \mathcal{P}$. By Urysohn's Smooth Lemma, a continuous function within $[0, 1]$ that is equal to 1 for $\mathbf{x} \in \{\mathbf{x} \mid B^{\text{Test}}(\mathbf{x}) \geq \bar{B}\}$ and 0 for $\mathbf{x} \in \mathbb{R}^n \setminus \mathcal{P}$ exists; scaling this function gives $B^{\text{Test}}(\mathbf{x})$.

The support of $B^{\text{Test}}(\mathbf{x})$ is a superset of $\mathcal{X}(\hat{\mathbf{u}}_i)$ and $B^{\text{Test}}(\mathbf{x})$ is in the range $[0, 1)$, meaning that it is a δ-barrier for $\mathbf{OP}(\hat{\mathbf{u}}_i)$ for some $\delta > 0$. It only remains to prove that $\mathbf{x}^{\lambda_j}$ is an optimal solution for $\mathbf{BP}(\hat{\mathbf{u}}_i, B^{\text{Test}}, \lambda_j)$, i.e.,

$$\mathbf{c}^\mathsf{T}\mathbf{x}^{\lambda_j} - \lambda_j \log B^{\text{Test}}(\mathbf{x}^{\lambda_j}) \leq \mathbf{c}^\mathsf{T}\mathbf{x} - \lambda_j \log B^{\text{Test}}(\mathbf{x}), \quad \forall \mathbf{x} \in \{\mathbf{x} \mid B^{\text{Test}}(\mathbf{x}) > 0\}.$$

First, consider the region $\{\mathbf{x} \mid B^{\text{Test}}(\mathbf{x}) \geq \bar{B}\}$. From (5.10), $\bar{B} \leq B^{*,\mathcal{P}}(\mathbf{x}^{\lambda_j}, \hat{\mathbf{u}}_i)$ meaning $B^{\text{Test}}(\mathbf{x}^{\lambda_j}) = B^{*,\mathcal{P}}(\mathbf{x}^{\lambda_j}, \hat{\mathbf{u}}_i)$. Because $\mathbf{x}^{\lambda_j}$ is optimal for $\mathbf{BP}(\hat{\mathbf{u}}_i, B^{*,\mathcal{P}}, \lambda_j)$ and this region is a subset of the support of $B^{*,\mathcal{P}}(\mathbf{x}, \hat{\mathbf{u}}_i)$, we have

$$\mathbf{c}^\mathsf{T}\mathbf{x}^{\lambda_j} - \lambda_j \log B^{\text{Test}}(\mathbf{x}^{\lambda_j}) \leq \mathbf{c}^\mathsf{T}\mathbf{x} - \lambda_j \log B^{\text{Test}}(\mathbf{x}), \quad \forall \mathbf{x} \in \{\mathbf{x} \mid B^{\text{Test}}(\mathbf{x}) \geq \bar{B}\}.$$

Now, consider the region $\{\mathbf{x} \mid B^{\text{Test}}(\mathbf{x}) < \bar{B}\}$. Replacing the minimum in (5.10) with an inequality and taking the logarithm on both sides yields

$$-\log \bar{B} \geq -\log B^{*,\mathcal{P}}(\mathbf{x}^{\lambda_j}, \hat{\mathbf{u}}_i) + \frac{1}{\lambda_j}\left(\mathbf{c}^\mathsf{T}\mathbf{x}^{\lambda_j} - \mathbf{c}^\mathsf{T}\mathbf{x}^{\mathcal{P}}\right).$$

We further re-arrange this inequality to

$$\mathbf{c}^\mathsf{T}\mathbf{x}^{\lambda_j} - \lambda_j \log B^{*,\mathcal{P}}(\mathbf{x}^{\lambda_j}, \hat{\mathbf{u}}_i) \leq \mathbf{c}^\mathsf{T}\mathbf{x}^{\mathcal{P}} - \lambda_j \log \bar{B} \tag{5.11}$$

$$\mathbf{c}^\mathsf{T}\mathbf{x}^{\lambda_j} - \lambda_j \log B^{\text{Test}}(\mathbf{x}^{\lambda_j}) \leq \mathbf{c}^\mathsf{T}\mathbf{x} - \lambda_j \log \bar{B} \quad \forall \mathbf{x} \in \{\mathbf{x} \mid B^{\text{Test}}(\mathbf{x}) < \bar{B}\} \tag{5.12}$$

$$\mathbf{c}^\mathsf{T}\mathbf{x}^{\lambda_j} - \lambda_j \log B^{\text{Test}}(\mathbf{x}^{\lambda_j}) \leq \mathbf{c}^\mathsf{T}\mathbf{x} - \lambda_j \log B^{\text{Test}}(\mathbf{x}) \quad \forall \mathbf{x} \in \{\mathbf{x} \mid B^{\text{Test}}(\mathbf{x}) < \bar{B}\} \tag{5.13}$$

We obtain (5.12) by substituting $B^{\text{Test}}(\mathbf{x}^{\lambda_j}) = B^{*,\mathcal{P}}(\mathbf{x}^{\lambda_j}, \hat{\mathbf{u}}_i)$ and noting $\mathbf{c}^\mathsf{T}\mathbf{x}^{\mathcal{P}} \leq \mathbf{c}^\mathsf{T}\mathbf{x}$ for all $\mathbf{x} \in\subset \mathcal{P}$. We obtain (5.13) because $B^{\text{Test}}(\mathbf{x}) < \bar{B}$. Thus, $\mathbf{x}^{\lambda_j}$ is an optimal solution to $\mathbf{BP}(\hat{\mathbf{u}}_j, B^{\text{Test}}, \lambda)$ and is (δ, ϵ)-optimal for $\mathbf{OP}(\hat{\mathbf{u}}_i)$ where $\epsilon = -\lambda_j \log \bar{B}$. □

Theorem 7 ensures that for any predicted decision from an in-sample context vector $\hat{\mathbf{u}}_i \in \hat{\mathcal{U}}$, the optimality gap is bounded from above by the sum of two terms: (i) the empirical error between the optimal solution to $\mathbf{BP}(\hat{\mathbf{u}}_i, B^{*,\mathcal{P}}, \lambda_j)$ and $\mathbf{OP}(\hat{\mathbf{u}}_i)$ as given in Theorem 6; and (ii) a (δ, ϵ)-optimality bound. The value of this bound, and hence the quality of the trained generator $F^{(j)}(\mathbf{u})$, depends on the quality of the classifier. Recall that λ_j is effectively a regularization parameter for training $F^{(j)}(\mathbf{u})$. If the classifier is sufficiently trained so that it is a δ-barrier for $\hat{\mathbf{u}}_i$, then selecting an appropriate λ_j ensures that the trained generator produces a solution that is arbitrarily close to the optimal value for $\mathbf{OP}(\hat{\mathbf{u}}_i)$.

In the proof of Theorem 7, we demonstrate that ϵ can be calculated using known

quantities. However, δ is a property of the classifier and the hidden feasible set $\mathcal{X}(\hat{\mathbf{u}}_i)$. Since we may not be able to characterize $\mathcal{X}(\hat{\mathbf{u}}_i)$ exactly, we do not know the exact value of δ. The next result demonstrates that we can bound this value as a function of the data.

Corollary 4. *For any $\hat{\mathbf{u}}_i \in \hat{\mathcal{U}}$, δ in (5.8) is bounded by $\delta \leq d_H\left(\{\hat{\mathbf{x}} \mid (\hat{\mathbf{x}}, \hat{\mathbf{u}}_i) \in \mathcal{D}\},\ \mathcal{P}\right)$.*

Proof. The δ term in the bound of (5.8) is derived from the fact that an optimal solution $\mathbf{x}^{\lambda_j}(\hat{\mathbf{u}}_i)$ to $\mathbf{BP}(\hat{\mathbf{u}}_i, B^{*,\mathcal{P}}, \lambda_j)$ is also an optimal solution to some barrier problem with a δ-barrier. The proof of Theorem 7 invokes two separate cases with two different bounds, i.e., δ takes a different value if $B^{*,\mathcal{P}}(\mathbf{x}^*, \hat{\mathbf{u}}_i) > 0$ or if $B^{*,\mathcal{P}}(\mathbf{x}^*, \hat{\mathbf{u}}_i) = 0$. We consider each case separately and show that δ can be bounded by the above in both cases.

If $B^{*,\mathcal{P}}(\mathbf{x}^*, \hat{\mathbf{u}}_i) > 0$, then the proof of Theorem 7 follows by constructing a new optimization problem (5.9) with a smaller feasible set and showing that the product classifier is a δ-barrier for that problem, where δ is equal to the Hausdorff distance between this feasible set and the support of the product classifier

$$\delta = d_H\left(\left\{\mathbf{x} \;\middle|\; \mathbf{x} \in \mathcal{X}(\hat{\mathbf{u}}_i),\ B^{*,\mathcal{P}}(\mathbf{x}, \hat{\mathbf{u}}_i) \geq \varepsilon\right\},\ \left\{\mathbf{x} \;\middle|\; B^{*,\mathcal{P}}(\mathbf{x}, \hat{\mathbf{u}}_i) > 0\right\}\right).$$

Because we can choose any sufficiently small value for the ε parameter, we select a value such that $B^{*,\mathcal{P}}(\hat{\mathbf{x}}, \hat{\mathbf{u}}_i) \geq \varepsilon$ for all $(\hat{\mathbf{x}}, \hat{\mathbf{u}}_i) \in \mathcal{D}$. Furthermore, all $\hat{\mathbf{x}}$ in this data set are feasible decisions to $\mathbf{OP}(\hat{\mathbf{u}}_i)$. Thus, we have $\left\{\mathbf{x} \;\middle|\; \mathbf{x} \in \mathcal{X}(\hat{\mathbf{u}}_i),\ B^{*,\mathcal{P}}(\mathbf{x}, \hat{\mathbf{u}}_i) \geq \varepsilon\right\} \supset \{\hat{\mathbf{x}} \mid (\hat{\mathbf{x}}, \hat{\mathbf{u}}_i) \in \mathcal{D}\}$. Finally, $\left\{\mathbf{x} \;\middle|\; B^{*,\mathcal{P}}(\mathbf{x}, \hat{\mathbf{u}}_i) > 0\right\} \subset \mathcal{P}$. Substituting the subset and superset into the definition of the Hausdorff distance yields the upper bound on δ for this case.

If $B^{*,\mathcal{P}}(\mathbf{x}^*, \hat{\mathbf{u}}_i) = 0$, then the proof of Theorem 7 follows by constructing a new barrier function $B^{\text{Test}}(\mathbf{x})$ for $\mathbf{OP}(\hat{\mathbf{u}}_i)$. Then, $\delta = d_H\left(\mathcal{X}(\hat{\mathbf{u}}_i),\ \left\{\mathbf{x} \mid B^{\text{Test}}(\mathbf{x}) > 0\right\}\right)$. However, again note that $\mathcal{X}(\hat{\mathbf{u}}_i) \supset \{\hat{\mathbf{x}} \mid (\hat{\mathbf{x}}, \hat{\mathbf{u}}_i) \in \mathcal{D}\}$ and that $\left\{\mathbf{x} \mid B^{\text{Test}}(\mathbf{x}) > 0\right\} \subseteq \mathcal{P}$. Substituting these two sets into the definition of the Hausdorff distance yields the same upper bound for this case. □

Corollary 4 reveals the key challenge of learning to optimize over hidden feasible sets. While we may generate decisions that satisfy a (δ, ϵ)-optimality guarantee, the quality of the δ-barrier is highly dependent on the available training data. Indeed, in order to appropriately train a classifier to learn to become a δ-barrier, we require a significant amount of data that includes both feasible $\mathcal{D}$ and infeasible $\bar{\mathcal{D}}$ decisions. This setup is typical of many deep learning applications. For example, the FaceNet architecture (Schroff et al., 2015) used more than 400,000 samples to train a neural network to verify

and recognize faces. In addition, $\mathcal{D}$ only has to contain feasible decisions (as opposed to optimal solutions) which may be easier to obtain. Nevertheless, many operational applications do not feature large data sets (Gupta and Rusmevichientong, 2020). Thus, in the next section, we propose an approach to increase the amount of training data for our prediction models.

5.5 Improving learning to optimize with data augmentation

Data augmentation is the practice of artificially generating data to improve the accuracy of predictive models and has been used with great success, particularly for computer vision applications with deep neural networks (Perez and Wang, 2017). The goal is to artificially create new data points to enhance the training signal so that the model acquires more generalization power. For instance, many image classification models rotate, flip, and blur images in the training set; the transformed images constitute a new sample with the same label as the original images (Yoo et al., 2020). Recently, human oracles have been used to augment data by correcting model predictions during training (Castrejon et al., 2017). The challenge for any data augmentation technique is to ensure that the artificially generated data points do not hinder the model's ability to learn. Thus, in this section, we introduce a computational oracle of feasibility to label predicted decisions in order to augment the classifier data. We then prove that embedding our models within an iterative framework with this data augmentation step will improve learning.

We assume access to a perfect feasibility oracle $\Psi(\mathbf{x}, \mathbf{u})$, which labels decisions by outputting $\Psi(\mathbf{x}, \mathbf{u}) = 1$ if $\mathbf{x} \in \mathcal{X}(\mathbf{u})$ and 0 otherwise. Using this oracle, we propose an iterative learning algorithm where the classifier trains with a progressively larger data set of both feasible and infeasible decisions. We prove that the classifier will learn to better approximate the hidden feasible set after each iteration which, in turn, improves the ability of the generator to learn to predict solutions to the optimization problem $\mathbf{OP}(\mathbf{u})$ for any context $\mathbf{u}$.

5.5.1 Constructing data-driven oracles

In our context, we consider an oracle to be a black-box model used in training to label feasibility with respect to the context vectors in the data set $\hat{\mathcal{U}}$. For example, this may be a human-in-the-loop decision-maker involved in the training process (e.g., Emmanouilidis et al., 2019). It may also be a data-driven oracle which is constructed by assuming some

knowledge of the hidden feasible set. For example, if we know that the hidden set contains a linear constraint but do not know the right-hand-side value, we can analyze $\mathcal{D}$ and $\bar{\mathcal{D}}$ to identify the largest feasible right-hand-side. Below, we present a simple example for creating such a data-driven oracle; we use similar constructions in our numerical experiments.

Example 5 (Data-driven oracle)**.** *Suppose that the hidden feasible set consists of a single linear constraint* $\mathcal{X}(\mathbf{u}) = \{\mathbf{x} \mid \mathbf{a}^\mathsf{T}\mathbf{x} \leq b(\mathbf{u})\}$ *where the vector* $\mathbf{a} \in \mathbb{R}^n$ *is known and is independent of the context. The right-hand-side can take one of two potential values that are dependent on the context vector, i.e.,* $b(\mathbf{u}) \in \{b_1, b_2\}$ *where* $b_1 < b_2$*. Since every* $(\hat{\mathbf{x}}_i, \hat{\mathbf{u}}_i) \in \mathcal{D}$ *is a feasible pair, we can estimate the right-hand-side for any context vector in the training set by evaluating* $\mathbf{a}^\mathsf{T}\hat{\mathbf{x}}_i$ *and designing the corresponding oracle:*

$$\Psi(\mathbf{x}, \hat{\mathbf{u}}_i) = \begin{cases} 1 & \text{if } \mathbf{a}^\mathsf{T}\mathbf{x} \leq b_1 \text{ and } \max_{(\hat{\mathbf{x}}, \hat{\mathbf{u}}_i) \in \mathcal{D}} \mathbf{a}^\mathsf{T}\hat{\mathbf{x}} \leq b_1 \\ 1 & \text{if } \mathbf{a}^\mathsf{T}\mathbf{x} \leq b_2 \text{ and } b_1 \leq \max_{(\hat{\mathbf{x}}, \hat{\mathbf{u}}_i) \in \mathcal{D}} \mathbf{a}^\mathsf{T}\hat{\mathbf{x}} \leq b_2\,, \\ 0 & \text{otherwise} \end{cases} \qquad \forall \hat{\mathbf{u}}_i \in \hat{\mathcal{U}}.$$

Data-driven oracles, such as the one presented in Example 5, offer a significant advantage as compared to a human-in-the-loop. A computational oracle can be called a large number of times during training and does not need to be defined for all possible context vectors. Nevertheless, one disadvantage of a rule-based oracle is that it may represent a tighter subset of $\mathcal{X}$ (e.g., if $\mathcal{D}$ contains feasible decisions that inadvertently satisfy more conservative constraints even though a larger value would have been sufficient). While this edge case may limit the quality of generated decisions, it occurs only if $\mathcal{D}$ is pathologically poor.

5.5.2 Interior Point Methods with Adversarial Networks

We now introduce an iterative oracle-guided algorithm, Interior Point Methods with Adversarial Networks (IPMAN). In each iteration of the algorithm, we train a classifier and a generator in succession as described in Section 5.4. Then, for each context $\hat{\mathbf{u}}_i \in \hat{\mathcal{U}}$, we have the trained generator predict decisions that are not present in the training data. We use the oracle to label each predicted decision as feasible or infeasible and then augment the existing data sets of feasible and infeasible decisions for the next iteration. Let $k \in \{1, \ldots, K\}$ index an iteration of training and superscript (k) index the model and data sets after the k-th iteration. Algorithm 1 summarizes our approach.

In each iteration, the classifier $B^{(k+1)}(\mathbf{x}, \mathbf{u})$ is trained with data sets $\mathcal{D}^{(k+1)}$ and $\bar{\mathcal{D}}^{(k+1)}$,

Algorithm 1 Interior Point Methods with Adversarial Networks (IPMAN)

Input: Number of IPMAN iterations K and IPM steps J; Dual regularizers $\{\lambda_j\}_{j=1}^M$; Initial data sets $\mathcal{D}^{(1)}$, $\bar{\mathcal{D}}^{(1)}$, and $\hat{\mathcal{U}}$.

Output: Final generative models $F^{(j,K)}$ for $j \in \{0, \ldots, J\}$

1: **for** $k = 1$ **to** K **do**
2: Train classifier $B^{(k)}$ via $\mathbf{FCP}(\mathcal{D}^{(k)}, \bar{\mathcal{D}}^{(k)})$.
3: **for** $j = 1$ **to** J **do**
4: Train generator $F^{(j,k)}$ via $\mathbf{GBP}(\hat{\mathcal{U}}, B^{(k)}, \lambda_j)$.
5: Update $\mathcal{D}^{(k+1)} = \mathcal{D}^{(k)} \cup \mathcal{Q}$, $\mathcal{Q} := \left\{ (F^{(j,k)}(\hat{\mathbf{u}}_i), \hat{\mathbf{u}}_i) \;\middle|\; \Psi(F^{(j,k)}(\hat{\mathbf{u}}_i), \hat{\mathbf{u}}_i) = 1, \hat{\mathbf{u}}_i \in \hat{\mathcal{U}} \right\}$.
6: Update $\bar{\mathcal{D}}^{(k+1)} = \bar{\mathcal{D}}^{(k)} \cup \bar{\mathcal{Q}}$, $\bar{\mathcal{Q}} := \left\{ (F^{(j,k)}(\hat{\mathbf{u}}_i), \hat{\mathbf{u}}_i) \;\middle|\; \Psi(F^{(j,k)}(\hat{\mathbf{u}}_i), \hat{\mathbf{u}}_i) = 0, \hat{\mathbf{u}}_i \in \hat{\mathcal{U}} \right\}$.
7: **end for**
8: **end for**
9: **return** $F^{(j,K)}$ for $j \in \{1, \ldots, J\}$

respectively, which are larger than the data sets $\mathcal{D}^{(k)}$ and $\bar{\mathcal{D}}^{(k)}$ due to the labelling procedure of the oracle. Due to this data augmentation step, the optimal solution set of the Feasibility Classification Problem can be shown to contract after each iteration, i.e., $B^{(k+1)}(\mathbf{x}, \mathbf{u})$ represents a tighter approximation to $\mathcal{X}(\mathbf{u})$ than $B^{(k)}(\mathbf{x}, \mathbf{u})$, which we now show formally.

Proposition 10. *Assume that $\mathcal{B}$ contains all continuous functions mapping $\mathcal{U} \times \mathbb{R}^n \to [0, 1]$. For any k, let $\mathcal{B}^{(k)}$ be the optimal solution set of $\mathbf{FCP}(\mathcal{D}^{(k)}, \bar{\mathcal{D}}^{(k)})$. Then, $\mathcal{B}^{(k+1)} \subset \mathcal{B}^{(k)}$.*

Proof. When $\mathcal{B}$ is unrestricted, Lemma 2 states that for any k, the optimal value of $\mathbf{FCP}(\mathcal{D}^{(k)}, \bar{\mathcal{D}}^{(k)})$ is 0 and can be achieved when the optimal solution satisfies $B^{(k)}(\hat{\mathbf{x}}, \hat{\mathbf{u}}) = 1$ for all $(\hat{\mathbf{x}}, \hat{\mathbf{u}}) \in \mathcal{D}^{(k)}$ and $B^{(k)}(\hat{\mathbf{x}}, \hat{\mathbf{u}}) = 0$ for all $(\hat{\mathbf{x}}, \hat{\mathbf{u}}) \in \bar{\mathcal{D}}^{(k)}$.

In the $(k+1)^{\text{st}}$ iteration, $\mathcal{D}^{(k+1)} = \mathcal{D}^{(k)} \cup \mathcal{Q}$ and $\bar{\mathcal{D}}^{(k+1)} = \bar{\mathcal{D}}^{(k)} \cup \bar{\mathcal{Q}}$ where $\mathcal{Q}$ and $\bar{\mathcal{Q}}$ are defined as in Algorithm 1. To show that $\mathcal{B}^{(k+1)} \subset \mathcal{B}^{(k)}$, we first prove $\mathcal{B}^{(k+1)} \subseteq \mathcal{B}^{(k)}$ and then present a counter-example which disproves the equivalence.

The objective function of $\mathbf{FCP}(\mathcal{D}^{(k+1)}, \bar{\mathcal{D}}^{(k+1)})$ is

$$\begin{aligned}
&\frac{1}{|\mathcal{D}^{(k+1)}|} \sum_{(\hat{\mathbf{x}},\hat{\mathbf{u}}) \in \mathcal{D}^{(k+1)}} \log B(\hat{\mathbf{x}}, \hat{\mathbf{u}}) + \frac{1}{|\bar{\mathcal{D}}^{(k+1)}|} \sum_{(\hat{\mathbf{x}},\hat{\mathbf{u}}) \in \bar{\mathcal{D}}^{(k+1)}} \log \big(1 - B(\hat{\mathbf{x}}, \hat{\mathbf{u}})\big) \\
&\qquad = \frac{\alpha}{|\mathcal{D}^{(k)}|} \sum_{(\hat{\mathbf{x}},\hat{\mathbf{u}}) \in \mathcal{D}^{(k)}} \log B(\hat{\mathbf{x}}, \hat{\mathbf{u}}) + \frac{1-\alpha}{|\mathcal{Q}|} \sum_{(\hat{\mathbf{x}},\hat{\mathbf{u}}) \in \mathcal{Q}} \log B(\hat{\mathbf{x}}, \hat{\mathbf{u}}) \\
&\qquad\quad + \frac{\alpha'}{|\bar{\mathcal{D}}^{(k)}|} \sum_{(\hat{\mathbf{x}},\hat{\mathbf{u}}) \in \bar{\mathcal{D}}^{(k)}} \log \big(1 - B(\hat{\mathbf{x}}, \hat{\mathbf{u}})\big) + \frac{1-\alpha'}{|\bar{\mathcal{Q}}|} \sum_{(\hat{\mathbf{x}},\hat{\mathbf{u}}) \in \bar{\mathcal{Q}}} \log \big(1 - B(\hat{\mathbf{x}}, \hat{\mathbf{u}})\big),
\end{aligned}$$

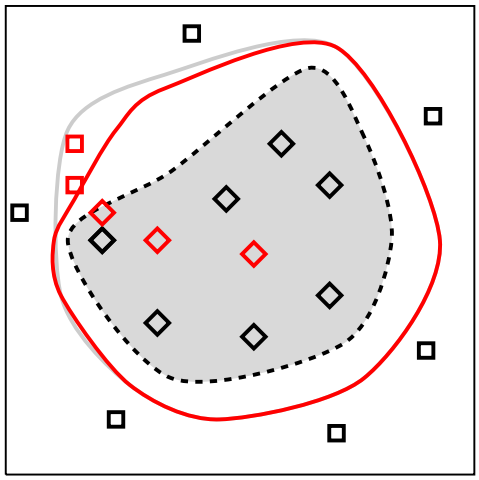


Figure 5.3: The trained classifier after an iteration of data augmentation. $\Diamond$ and $\square$ represent points in $\mathcal{D}^{(k+1)}$ and $\bar{\mathcal{D}}^{(k+1)}$, respectively, with the colored (red) points denoting the augmented points. The filled region is $\mathcal{X}(\mathbf{u})$. The gray line shows the support of $B^{(k)}(\mathbf{x}, \mathbf{u})$ and the red line shows the support of $B^{(k+1)}(\mathbf{x}, \mathbf{u})$.

where $\alpha = |\mathcal{D}^{(k)}|/|\mathcal{D}^{(k+1)}|$ and $\alpha' = |\bar{\mathcal{D}}^{(k)}|/|\bar{\mathcal{D}}^{(k+1)}|$ are the mixture weights defining the ratio of existing to new points in each data set. Because the optimal value of $\mathbf{FCP}(\mathcal{D}^{(k+1)}, \bar{\mathcal{D}}^{(k+1)})$ is 0 and $B(\mathbf{x}, \mathbf{u}) \in [0, 1]$, each of the individual terms must be equal to 0 for an optimal solution. However, the first and third terms define the objective function for $\mathbf{FCP}(\mathcal{D}^{(k)}, \bar{\mathcal{D}}^{(k)})$. Thus, any optimal solution $B^{(k+1)}$ to $\mathbf{FCP}(\mathcal{D}^{(k+1)}, \bar{\mathcal{D}}^{(k+1)})$ must also be optimal for $\mathbf{FCP}(\mathcal{D}^{(k)}, \bar{\mathcal{D}}^{(k)})$ implying $\mathcal{B}^{(k+1)} \subseteq \mathcal{B}^{(k)}$.

To prove the inclusion is strict, consider the sets $\mathcal{D}^{(k)} \cup \bar{\mathcal{Q}}$ and $\bar{\mathcal{D}}^{(k)}$. We can define a function $B^*(\mathbf{x}, \mathbf{u})$ such that $B^*(\mathbf{x}, \mathbf{u}) = 1$ for all $(\mathbf{x}, \mathbf{u}) \in \mathcal{D}^{(k)} \cup \bar{\mathcal{Q}}$ and $B^*(\mathbf{x}, \mathbf{u}) = 0$ for all $(\mathbf{x}, \mathbf{u}) \in \bar{\mathcal{D}}^{(k)}$, i.e., $B^* \in \mathcal{B}^{(k)}$. However, then $B^*(\hat{\mathbf{x}}, \hat{\mathbf{u}}) = 1$ for all $(\hat{\mathbf{x}}, \hat{\mathbf{u}}) \in \bar{\mathcal{Q}}$ and $B^*(\mathbf{x}, \mathbf{u})$ has an infinite objective function value for $\mathbf{FCP}(\mathcal{D}^{(k+1)}, \bar{\mathcal{D}}^{(k+1)})$. Thus, $B^* \notin \mathcal{B}^{(k+1)}$. $\square$

Proposition 10 demonstrates that by iteratively increasing the amount of available data using the oracle, we can sequentially contract the optimal solution set. Because the sets are augmented with correctly labelled data (since the oracle is perfect), data augmentation removes classifiers that are a looser approximation to the feasible set. Intuitively, it allows the classifier to correct regions it had previously mislabelled as feasible and reinforce regions it had correctly labelled so that it does not incorrectly mislabel them in future iterations. Figure 5.3 shows an example of this behavior; the red classifier is a tighter approximation of $\mathcal{X}(\mathbf{u})$ than the gray classifier due to the presence of the additional points.

Recall from Theorem 7 that a trained generator satisfies a (δ, ϵ)-optimality bound where δ is a property of the classifier. Corollary 4 demonstrates that while δ cannot

be directly calculated, we can compute its upper bound which is dependent on the data set of feasible decisions $\mathcal{D}$. In IPMAN, by increasing the size of the data set after each iteration, we ensure that this upper bound is non-increasing and the resulting optimality gap decreases.

5.6 Generalization of optimality guarantees to unseen instances

The IPMAN framework iteratively trains a generative model to predict solutions to a barrier optimization problem when given a context vector. In this section, we analyze the potential for our approach, and any machine learning model used to generate solutions to contextual constrained optimization problems, to predict (δ, ϵ)-optimal solutions when given an out-of-sample context vector $\mathbf{u}$. To this end, assume there exists a probability distribution over context vectors $\mathbb{P}_{\mathbf{u}}$ and that the data set of context vectors $\hat{\mathcal{U}}$ is sampled i.i.d. according to $\mathbb{P}_{\mathbf{u}}$. We then use Rademacher complexity theory to obtain a probabilistic bound on the empirical error from an out-of-sample input (Bartlett and Mendelson, 2002).

Definition 2 (Bertsimas and Kallus (2020))**.** *Let $\mathcal{F} \subset \{F(\mathbf{u}) : \mathcal{U} \to \mathbb{R}^n\}$ be a function class and $\hat{\mathcal{U}} \sim \mathbb{P}_{\mathbf{u}}$ be an i.i.d. data set. The empirical multivariate Rademacher complexity of $\mathcal{F}$ is*

$$\hat{\mathfrak{R}}_{N_{\mathbf{u}}}(\mathcal{F}, \hat{\mathcal{U}}) = \mathbb{E}_{\boldsymbol{\sigma} \sim p_{\boldsymbol{\sigma}}} \left[\frac{2}{N_{\mathbf{u}}} \sup_{F \in \mathcal{F}} \sum_{i=1}^{N_{\mathbf{u}}} \boldsymbol{\sigma}_i^{\mathsf{T}} F(\hat{\mathbf{u}}_i) \;\middle|\; \hat{\mathcal{U}} = \{\hat{\mathbf{u}}_i\}_{i=1}^{N_{\mathbf{u}}} \right],$$

where $\boldsymbol{\sigma}_i \sim p_{\boldsymbol{\sigma}}$ is an n-dimensional vector of i.i.d. Rademacher variables. The multivariate Rademacher complexity of $\mathcal{F}$ is $\mathfrak{R}_{N_{\mathbf{u}}}(\mathcal{F}) = \mathbb{E}_{\hat{\mathcal{U}} \sim \mathbb{P}_{\mathbf{u}}} \left[\hat{\mathfrak{R}}_{N_{\mathbf{u}}}(\mathcal{F}, \hat{\mathcal{U}}) \right]$.

In statistical learning theory, Rademacher complexities are used to generate risk bounds that are dependent on the class of learning model that is used ($\mathcal{F}$) and the training data (Bartlett and Mendelson, 2002). Bertsimas and Kallus (2020) use the theory to develop generalization bounds for predicting decisions to problems with a conditional stochastic optimization objective; we extend their work by providing a probabilistic bound on (δ, ϵ)-optimality when the feasible set is not fully specified. Typically, models that can learn complex relationships will require a more complex model class. The IPMAN algorithm is agnostic to the class of model used. However, if a specific generalization bound is required, a model class $\mathcal{F}$ with a tightly bounded Rademacher

complexity should be selected.

Remark 5. *Although the literature mostly focuses on the single-variate Rademacher complexity, Bertsimas and Kallus (2020) and Maurer (2016) prove bounds for linear multivariate classes (e.g., $\mathcal{F}_R = \{\mathbf{W}\mathbf{u} \mid \|\mathbf{W}\| \leq R\}$). In general, if $F(\mathbf{u}) = (F_1(\mathbf{u}), \ldots, F_n(\mathbf{u}))$, then $\mathcal{F} \subset \times_{\ell=1}^n \mathcal{F}_\ell$, where $\mathcal{F}_\ell = \{F(\mathbf{u})^\mathsf{T}\mathbf{e}_\ell \mid F \in \mathcal{F}\}$ and $\mathbf{e}_\ell$ is the ℓ-th identity vector. Then, $\hat{\mathfrak{R}}_{N_\mathbf{u}}(\mathcal{F}, \hat{\mathcal{U}}) \leq \sum_{\ell=1}^n \hat{\mathfrak{R}}_{N_\mathbf{u}}(\mathcal{F}_\ell, \hat{\mathcal{U}})$ decomposes to a sum of single-variate complexities. We refer to Bartlett and Mendelson (2002) for bounds on linear and tree models and Bartlett and Mendelson (2002); Neyshabur et al. (2015) and Foster et al. (2018) for neural networks.*

Instead of using the product barrier composed of the classifier and the barrier for the known constraints, we assume that the generator is trained using a δ-barrier $B_\delta(\mathbf{x}, \mathbf{u})$ for some $\delta > 0$, i.e., by minimizing $\mathbf{GBP}(\hat{\mathcal{U}}, B_\delta, \lambda)$. Note that since we are using a fixed data set $\hat{\mathcal{U}}$, there is no guarantee that $F(\mathbf{u})$ will be feasible or even satisfy $B_\delta(F(\mathbf{u}), \mathbf{u}) > 0$ for an arbitrary $\mathbf{u}$. Because $\mathcal{P}$ is available, however, we can always project any generated solution to the polyhedron to ensure that the generator predicts decisions that are bounded.

Assumption 2. *Consider a δ-barrier $B_\delta(\mathbf{x}, \mathbf{u})$ and generator $F(\mathbf{u})$ trained via the Generative Barrier Problem $\mathbf{GBP}(\hat{\mathcal{U}}, B_\delta, \lambda)$. We use the projected generator*

$$F^*(\mathbf{u}) = \arg\min_{\mathbf{x}} \{\|\mathbf{x} - F(\mathbf{u})\| \mid \mathbf{x} \in \mathcal{P}\}$$

at test time.

Our generalization bound follows from Bertsimas and Kallus (2020). While they derive an empirical risk bound on an unconstrained stochastic optimization problem, we focus on (δ, ϵ)-optimality for a constrained continuous optimization problem (see Appendix C.2 for proof).

Theorem 8. *Let F^* satisfy Assumption 2, K and L_∞ be sufficiently large positive constants, and fix $\beta \in (0, 1)$. Then, for any $\gamma > 0$, the following inequality holds*

$$\begin{aligned}
&\mathbb{P}_\mathbf{u}\Big\{\mathbf{c}^\mathsf{T}F^*(\mathbf{u}) - \epsilon - \gamma < \mathbf{c}^\mathsf{T}\mathbf{x}^*(\mathbf{u}) < \mathbf{c}^\mathsf{T}F^*(\mathbf{u}) + \delta + \gamma\Big\} \\
&\qquad \geq 1 - \frac{\dfrac{1}{N_\mathbf{u}}\displaystyle\sum_{i=1}^{N_\mathbf{u}} \left|\mathbf{c}^\mathsf{T}F^*(\hat{\mathbf{u}}_i) - \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\hat{\mathbf{u}}_i)\right| + K\sqrt{\dfrac{\log(1/\beta)}{2N_\mathbf{u}}} + \sqrt{2n}L_\infty\mathfrak{R}_{N_\mathbf{u}}(\mathcal{F})}{\gamma},
\end{aligned}$$

with probability at least $1 - \beta$ with respect to the sampling of $\hat{\mathcal{U}} \sim \mathbb{P}_\mathbf{u}$.

Theorem 8 bounds the (δ, ϵ)-optimality of a random context vector. It specifies, given a γ and F^*, the probability that the model will predict a $(\delta + \gamma/L, \epsilon + \gamma)$-optimal solution for an out-of-sample context vector $\mathbf{u}_{N_\mathbf{u}+1} \sim \mathbb{P}_\mathbf{u}$. The first term $(\sum_{i=1}^{N_\mathbf{u}} |\mathbf{c}^\intercal F^*(\hat{\mathbf{u}}_i) - \mathbf{c}^\intercal \mathbf{x}^\lambda(\hat{\mathbf{u}}_i)|)/N_\mathbf{u}$, is the empirical error associated with solving $\mathbf{GBP}(\hat{\mathcal{U}}, B_\delta, \lambda)$ versus $\mathbf{BP}(\hat{\mathbf{u}}_i)$ for all $\hat{\mathbf{u}}_i \in \hat{\mathcal{U}}$. It effectively measures how well the model performs on in-sample data. The second term is dependent on the constants K and $1/\beta$ and scales with $O(1/N_\mathbf{u})$. Thus, as the size of the data set increases, the smaller this term becomes. The third term is dependent the Rademacher complexity of $\mathcal{F}$. The greater the representative capacity of the learning model the larger its Rademacher complexity. The summation of these terms is a prediction bound on the optimality of unseen instances and holds with probability at least $1 - \beta$. Thus, in order to obtain a tight and useful bound, we must balance the trade-off between a model class with high complexity versus obtaining a model with low empirical error.

5.7 Optimal dose generation for radiation therapy treatment planning

We implement IPMAN to predict optimal dose distributions for patients with head-and-neck cancer. Unlike previous machine learning dose generation techniques, we recast the task of predicting a clinically acceptable dose distribution to constructing an optimal dose distribution for a given patient. The selection of clinical criteria to satisfy for a patient is modeled as a latent choice constraint dependent on the CT image, which is both specific to a patient and the institution providing care. Specifically, a given dose is *feasible* if it satisfies the same set of criteria that the oncologist prescribed for the patient.

By treating dose generation as an optimization problem, we set an objective to minimize the average dose to OARs. That is, every dose generated will seek to minimize the radiation to the healthy tissue. Note that this implies for dose generation that we employ a single objective for all patients. While the later stages of automated planning may involve multiple objectives, using a single objective at the dose generation stage allows us to form a standard notion of treatment optimality from the perspective of an oncologist. We show in our comparisons with benchmark prediction models that treatment plans generated from IPMAN both (i) capture the same clinical trade-offs that oncologists would prescribe after evaluation, and (ii) deliver the same or lower dose on average to healthy tissue. We further show that IPMAN has the ability of adapting to learning institution-specific criteria without training on an institution-specific data set of

delivered plans. In particular, we use the oracle to learn a constraint that was not present in the original data to demonstrate how IPMAN can be deployed at cancer centers with different clinical criteria.

5.7.1 Data and model

We use our clinical data set of 217 treatment plans, randomly split into 100, 67, and 50 plans for training, validation, and held-out testing, respectively (see Chapter 2 for details on the data). Each patient contains up to four OARs and three tumor volumes, referred to as planning target volumes (PTVs), that have been contoured and labelled.

Remark 6. *The experiments in this chapter only model the larynx, mandible, left parotid, and right parotid rather than all seven OARs as in the previous chapters. This simpler model facilitates computation and the clinical criteria for these three regions are nearly always satisfied for* 100% *of patients (e.g., see Table 3.3). Further details are available in Appendix C.*

Let $\mathcal{O}$ and $\mathcal{T}$ index the OARs and PTVs, respectively, and let $\mathcal{R} := \mathcal{O} \cup \mathcal{T}$ index all structures of interest. For each structure, let $\mathcal{V}_r$ index the corresponding voxel set (elements of $\mathbf{x}$ and $\mathbf{u}$). Let z_r denote the average dose delivered to structure r and $\mathbf{z} \in \mathbb{R}^7$ denote the vector of z_r. To illustrate the IPMAN methodology, we formulate an RT optimization problem that minimizes the sum of average doses delivered to the OARs subject to satisfying the relevant clinical criteria and known polyhedral constraints (see Babier et al., 2018b). This model closely approximates the traditional weighted optimization models that are used as a surrogate to the treatment planning problem. Although the objective is simplified for the sake of computational efficiency, the constraints are representative of the realistic clinical problem, which is the focus of our methodology (more discussion on model choice is given in Appendix C.3.1). The clinical criteria for each OAR is an upper bound on either the mean z_o or maximum dose delivered $z_o^{\max}$, while the criteria for each PTV is a lower bound on the minimum dose delivered to the 90-th percentile z_t^{90} ($\mathbf{z}^{90} \in \mathbb{R}^3$) of the target structure, a Value-at-Risk (VaR) metric. We formulate each of the clinical criteria as a hidden bound $\hat{z}_r(\mathbf{u})$ dependent on the patient geometry and oncologist choice. The optimization problem is summarized below:

$$\mathbf{RT}(\mathbf{u}): \underset{\mathbf{x},\mathbf{z},\mathbf{z}^{\max},\mathbf{z}^{90}}{\text{minimize}} \frac{1}{|\mathcal{O}|}\sum_{o\in\mathcal{O}} z_o \tag{5.14a}$$

$$\text{subject to } z_r = \frac{1}{|\mathcal{V}_r|}\sum_{v\in\mathcal{V}_r} x_v \qquad \forall r \in \mathcal{R} \tag{5.14b}$$

$$z_o^{\max} \geq x_v \qquad \forall o \in \mathcal{O}, v \in \mathcal{V}_o \tag{5.14c}$$
$$z_t^{90} = \mathrm{VaR}_{90}\big(\{x_v \mid v \in \mathcal{V}_t\}\big) \qquad \forall t \in \mathcal{T} \tag{5.14d}$$
$$\underline{z}_r \leq z_r \leq \overline{z}_r \qquad \forall r \in \mathcal{R} \tag{5.14e}$$
$$z_o \leq \hat{z}_o(\mathbf{u}) \qquad \forall o \in \{\text{Right Parotid, Left Parotid, Larynx}\} \tag{5.14f}$$
$$z_0^{\max} \leq \hat{z}_o(\mathbf{u}) \qquad \forall o \in \{\text{Mandible}\} \tag{5.14g}$$
$$z_t^{90} \geq \hat{z}_t(\mathbf{u}) \qquad \forall t \in \mathcal{T}. \tag{5.14h}$$

Constraints (5.14b)–(5.14d) define the dose summary statistics of the mean, maximum, and VaR. In (5.14e), we mandate a fixed set of polyhedral constraints on $\mathbf{z}$ obtained by calculating the maximum and minimum mean doses over all patients in the ground truth data set; this constitutes the polyhedral relaxation $\mathcal{P}$. Finally, (5.14f)–(5.14h) define the hidden patient-specific constraints, i.e., the clinical criteria that must be learned. Whereas (5.14f) and (5.14g) ensure that the dose delivered to each OAR is below a threshold, (5.14h) ensures that PTVs receive a sufficiently high dose of radiation. Because it is not often possible to simultaneously satisfy all clinical criteria, these hidden constraints are conditional. If the ground truth plan from the data set satisfied a hidden constraint (i.e., an oncologist deemed it necessary), we require that a generated plan must satisfy it as well. In other words for any patient $\hat{\mathbf{u}}_i$ in our clinical data set,

$$\hat{z}_r(\hat{\mathbf{u}}_i) = \begin{cases} \hat{z}_r & \text{if the ground truth dose for } \hat{\mathbf{u}}_i \text{ satisfies the bound in Table 5.1} \\ +\infty & \text{otherwise for } r \in \mathcal{O} \\ 0 & \text{otherwise for } r \in \mathcal{T} \end{cases}.$$

The hidden nature of these constraints arises from the fact that a planner does not know a priori if the constraint is needed. Note that the VaR criteria for each PTV is a non-convex constraint and thus, the model would be difficult to solve even if the hidden constraints were known. The values for the bounds $\hat{z}_r$ are given in Table 5.1 (column 2).

5.7.2 Methods

We use two benchmark models to analyze the quality of the predictions produced by IPMAN: a 3-D U-net convolutional neural network (CNN) (Kearney et al., 2018; Nguyen et al., 2019), and a 3-D generative adversarial network (GAN) (Babier et al., 2020a). Nguyen et al. (2019) implement a CNN that predicts dose distributions from 3-D CT images and show its effectiveness in the prediction stage of automated planning. The CNN is trained

via supervised learning by minimizing an l_2 norm of predicted doses from a ground truth clinical data set. The conditional GAN is a 3-D extension of the one introduced in Chapter 4. This network predicts dose distributions from 3-D CT images in one shot, whereas the GAN in Chapter 4 predicted 2-D slices individually before concatenation.

Remark 7. *The 3-D CNN of Nguyen et al. (2019) is an extension of the original 2-D U-net CNN of Nguyen et al. (2017) which we used as a baseline in Chapter 4. The 3-D GAN is a more powerful variant of the 3-D GANCER-sc. model that was proposed by Babier et al. (2020a) and used as a dose prediction in Chapter 3.*

As the generator and classifier in IPMAN play similar roles to the generator and discriminator in a GAN, we use a slightly modified architecture from Babier et al. (2020a) (c.f. Chapter 4) to create $F(\mathbf{u})$ and $B(\mathbf{x}, \mathbf{u})$ (details are provided in Appendix C.3.2). Specifically for the experiments in Section 5.7.3, we include an l_1 regularization term $\|F(\mathbf{u}) - \mathbf{u}\|_1$ to the loss function of $\mathbf{GBP}(\hat{\mathcal{U}}, B, \lambda_j)$, which is commonly used for model stability in Style Transfer GANs (e.g., Isola et al., 2017). All models are trained using the Adam optimizer with $(\beta_1, \beta_2) = (0.5, 0.999)$. We train the classifier with a learning rate of 1×10^{-3} and the generator with a learning rate of 2×10^{-5}. Initially, the data set of feasible decisions $\mathcal{D} = \{(\hat{\mathbf{u}}_i, \hat{\mathbf{x}}_i)\}_{i=1}^{N_{\mathbf{u}}}$ consists solely of the 100 clinical plans used in training, the data set of parameters $\hat{\mathcal{U}} = \{\hat{\mathbf{u}}_i\}_{i=1}^{N_{\mathbf{u}}}$ contains their corresponding patient CT images, and the data set of infeasible decisions is empty (i.e., $\bar{\mathcal{D}} = \emptyset$).

Using the training set of patients, we train the generator and classifier iteratively with IPMAN. At the end of each iteration, we evaluate the predictions made by the generator on the validation set of patients. After 11 iterations, constraint satisfaction on the validation set stabilizes and we use the held-out test set to assess performance.

We train four generative models corresponding to $\lambda \in \{256, 64, 16, 4\}$. In each iteration, the generator predicts solutions to the corresponding barrier problem, meaning that training over a range of λ ensures that in every iteration, the oracle labels and augments the data sets with predictions lying in a diverse set of areas in and around the feasible set. Similarly, we train each baseline model for 200 epochs (approximately 24 hours). This is roughly the same duration of time required to train 11 iterations of IPMAN, thus maintaining a fair comparison. The data generation, implementation, and training details are provided in Appendix C.3.3. Below, we highlight the initialization steps and refinements to the IPMAN algorithm made for our computational experiments.

Pre-training to initial feasible decisions.

Classical IPMs generally require an initial point $\mathbf{x}^{(0)}$ that is strictly feasible. Analogously in IPMAN, ensuring that $F(\mathbf{u})$ is initialized at a stage where it usually predicts feasible decisions implies that the training loss is not extremely high at early stages and helps to stabilize training. Thus, before starting the IPMAN algorithm, we pre-train $F(\mathbf{u})$ as the generator in a GAN and save the weights.

Generating an initial $\bar{\mathcal{D}}$.

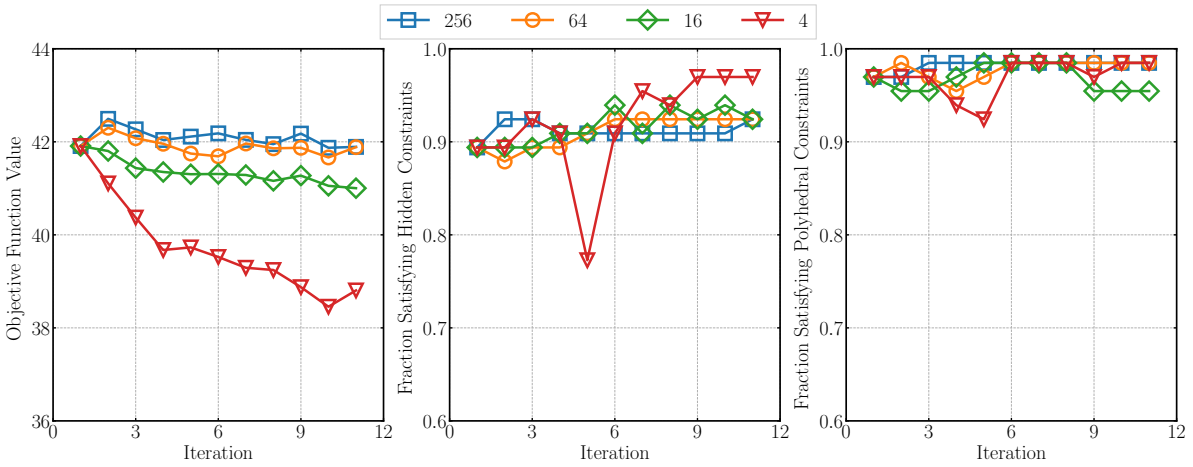
Before training the classifier in the first iteration of IPMAN, we require an initial data set of infeasible decisions $\bar{\mathcal{D}}$. During the pre-training stage, the generator of the GAN creates a set of candidate decisions. We label the generated decisions and assign them to the appropriate sets $\mathcal{D}$ and $\bar{\mathcal{D}}$ in order to initialize IPMAN with an augmented data set.

Learning multi-label feasibility.

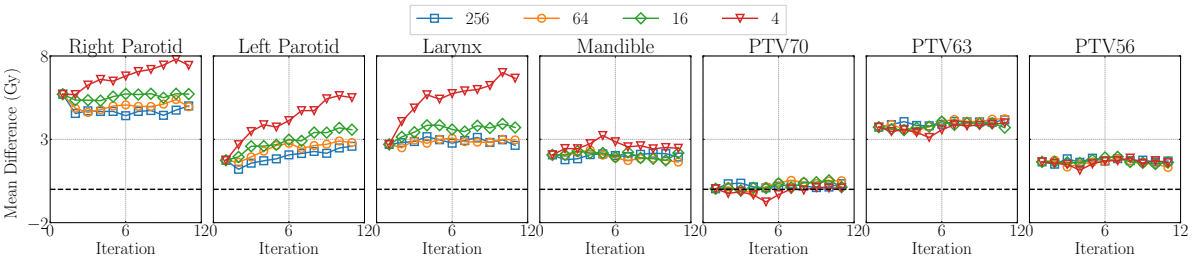
A feasible dose distribution must satisfy multiple polyhedral and hidden constraints corresponding to different PTVs and OARs. Learning to classify a decision as feasible is challenging due to the granularity of constraint satisfaction and the variety of constraints that are present. Consequently, we separate the classification problem into one for each of the four OARs and three PTVs of the patient. That is, for each structure of interest, we train a separate classifier. The δ-barrier optimization problem is then the sum of the different classifier outputs; this is equivalent to the classical barrier problem. That is, let $B_r(\mathbf{x}, \mathbf{u})$ be a classifier for $r \in \mathcal{R}$ that assesses whether the polyhedral and hidden constraints for that structure are satisfied. Then, the barrier problem is $\min_{\mathbf{x}}\{\mathbf{c}^\mathsf{T}\mathbf{x} - \lambda \sum_{r\in\mathcal{R}} B_r(\mathbf{x}, \mathbf{u})\}$.

5.7.3 Learning to predict optimal dose distributions

We use a slightly modified version of the clinical satisfaction criterio introduced in Chapter 2 for our evaluation (see column 2 of Table 5.1). As the relevancy of criteria (i.e., feasibility) for each patient is determined by an oncologist, we evaluate generated doses on whether they satisfy the same set of criteria as the the ground truth, just as in Chapter 3. For numerical stability, satisfaction is defined as meeting the dose bound to within a 1 Gy relaxation. In the clinical literature, dose predictions are commonly evaluated on a voxel-level to within 3% of the maximum prescribed dose, i.e., 2.1 Gy for our problem (Low et al., 1998). In our analysis, we consider constraints rather than direct voxels



(a) Objective function and fraction of feasible plans with respect to the hidden and polyhedral constraints.



(b) Average difference from the hidden constraint bound $\hat{z}_r(\mathbf{u})$. The dashed line is 0 Gy. Above 0 suggests plans satisfy the constraints on average.

Figure 5.4: Statistics on the validation set obtained during training on criteria from our institution.

and tighten the tolerance to 1 Gy. The relaxation can also be interpreted as the δ for a δ-barrier; if all decisions satisfy a given constraint, the the corresponding classifier is a δ-barrier with $\delta = 1$ Gy.

Figure 5.4(a) displays the average objective function value and the fraction of plans that satisfy the hidden and polyhedral constraints over training iterations. We find that in training, the objective function value improves as a function of the number of iterations, while hidden constraint satisfaction also increases. For example, in the first iteration, 89% of predictions in the validation set satisfy all of their hidden constraints, whereas this fraction increases to 97% by iteration 11. Polyhedral constraint satisfaction also increases from 88% to 95%. This suggests that the IPMAN algorithm trains the model to generate fewer infeasible doses. In other words, the classifier is learning to produce a tighter characterization of the feasible set (see Proposition 10).

Figure 5.4(b) shows the average difference from the boundary of the hidden constraint for each structure. If the difference is positive, doses on average satisfy the hidden criteria.

Table 5.1: The percentage of predicted decisions on the held-out test set that satisfy each hidden constraint to 1 Gy relaxation. The best performing models on the summary statistics are highlighted.

Structure	Criteria (Gy)	Baselines		IPMAN (λ)			
		GAN	CNN	256	64	16	4
Right Parotid	$z_o \leq 26$	85.7	85.7	86.2	90.0	93.3	100
Left Parotid	$z_o \leq 26$	70.0	60.0	70.0	90.0	90.0	100
Larynx	$z_o \leq 45$	93.3	83.3	89.7	89.7	93.3	100
Mandible	$z_o^{\max} \leq 73.5$	100	100	100	100	100	100
PTV70	$z_t^{90} \geq 70$	97.6	97.6	97.6	97.6	95.2	92.8
PTV63	$z_t^{90} \geq 63$	96.3	96.3	96.3	96.3	96.3	96.3
PTV56	$z_t^{90} \geq 56$	100	100	100	100	100	100
All hidden constraints		86.0	78.0	82.0	88.0	88.0	94.0
All polyhedral constraints		92.0	90.0	94.0	92.0	90.0	94.0
Objective function value		40.3	41.0	41.0	41.0	40.0	37.8

We observe two important phenomena. First, the four leftmost plots are associated with OAR constraints. By minimizing the objective, the associated OAR constraints see progressively better adherence as expected. Note that the Mandible and PTV70 structures often overlap, meaning their constraints conflict with each other, preventing improvement for this organ. Second, the PTV constraints show small but sustained improvement as the number of training iterations increase. This is because they are solely associated with feasibility and are not part of the objective function. In particular, the PTV70 constraint is typically the hardest to satisfy in practice; IPMAN learns this difficulty and makes predictions that lie close to the boundary of the feasible set.

Table 5.1 shows performance on the held-out test set for IPMAN at iteration 11 against the baseline models. In general, IPMAN models with $\lambda \leq 64$ satisfy the hidden constraints better than the baselines. IPMAN with $\lambda = 4$ dominates all other models, including the baselines, in hidden and polyhedral constraint satisfaction, as well as objective function value. That is, this model predicts dose distributions that deliver lower dose to healthy tissue while better satisfying the clinical criteria. We conclude that training via IPMAN yields prediction models that produce feasible decisions more often and with a lower objective function value than existing state-of-the-art methods. We also observe that with higher values of λ, constraint satisfaction comes with a price; the objective function value is higher.

Recall that optimal solutions to the barrier problem satisfy a (δ, ϵ)-optimality guar-

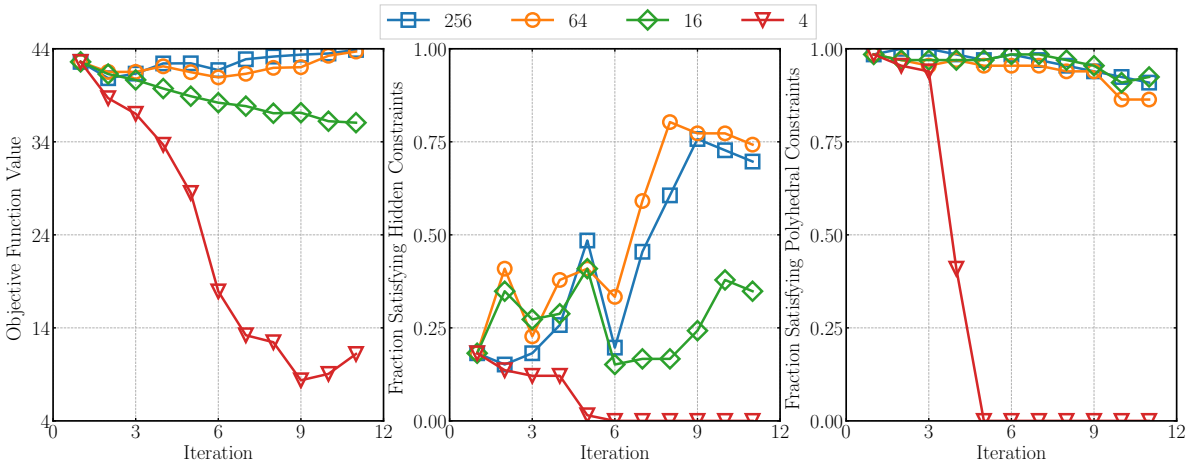
antee. At high λ, this translates to a non-trivial upper bound with respect to the optimal value of the true problem, while at low λ, the guarantee translates to a non-trivial lower bound (see Theorem 6 and also Appendix C.1). Because the generator learns via empirical risk minimization of the barrier problem, the corresponding predictions should also satisfy similar upper and lower bounds (see Theorem 7 and 8). While we may consequently expect setting a low λ to yield predictions that are infeasible (i.e., satisfying a non-trivial lower bound with respect to the optimal value), we find that $\lambda = 4$ yields the best performance on out-of-sample data. In the next experiment, where the data is not perfectly indicative of the constraints, we observe the benefits of using a higher λ value to predict decisions that are more likely to be feasible.

5.7.4 Adapting to the clinical constraints of a new institution

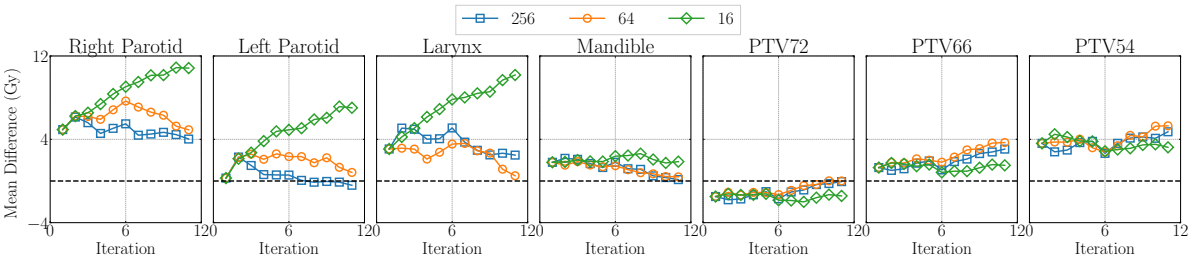
The previous experiments were constructed using the clinical criteria from one institution under which the ground truth plans were developed. However, different clinics often use different criteria (Wu et al., 2017). Further, small clinics may have limited patient volume which may not be sufficient to properly train institution-specific models using existing methods (Boutilier et al., 2016). In this experiment, we show how IPMAN can be trained using the original data set to learn to predict feasible and optimal treatments for new clinical constraints.

We use clinical criteria obtained from Geretschläger et al. (2015) who pursue a more aggressive treatment policy for head-and-neck cancer. They prescribe tumors to receive 72 Gy, 66 Gy, and 54 Gy to their three target sites, respectively, which we re-label in our data sets as PTV72, PTV66, and PTV54. Note that relative to the previous criteria in Section 5.7.3, two of the criteria have become stricter while the third is now easier to satisfy. Although we do not know the exact preferences of oncologists in determining when a criteria is necessary, we assume that any patient in our data set who was prescribed dose that satisfied the PTV hidden lower bound constraint from our institution would be prescribed dose at the corresponding new level at this new institution.

We train the generator and classifier using IPMAN for 11 iterations using the same settings as the previous section with one difference: we omit the l_1 regularization term. While regularization can be useful to ensure that predictions are not vastly different from clinical data (see Appendix C.3.3 for details), in this experiment, the clinical doses tend to be infeasible under the new criteria because they were generated using criteria from the original institution. For example, no plans in our data set received more than 72 Gy of dose to PTV70. Including the l_1 term would, therefore, inappropriately guide the



(a) Objective function and fraction of feasible plans with respect to the hidden and polyhedral constraints.



(b) Average difference from the hidden constraint bound $\hat{z}_r(\mathbf{u})$. The dashed line is 0 Gy. Above 0 suggests plans satisfy the constraints on average. $\lambda = 4$ is omitted to preserve scale.

Figure 5.5: Statistics on the validation set obtained during training on criteria from Geretschläger et al. (2015).

model to generate doses that tried to match the old criteria, rather than learn the new criteria.

Figure 5.5(a) displays the objective function value and the fraction of plans that satisfied the hidden and polyhedral constraints. The models trained for $\lambda \leq 16$ decrease in objective function value and constraint satisfaction as the algorithm progresses. In the early stages, the classifier (which is initially trained mainly using the clinical data and doses sampled from the same distribution) has not yet observed a sufficient and diverse number of feasible plans. Therefore, the classifier is not yet a sufficient δ-barrier, allowing the generator to leave the feasible set.

Generally at high λ, optimal solutions to the barrier problem are less aggressive in terms of minimizing the objective and instead lie well in the interior of the feasible set (see Appendix C.1). As the classifier improves, particularly after iteration 6, the generative models for $\lambda \geq 64$ quickly learn to predict solutions that are more likely to satisfy the hidden constraints. In particular, the distance from the PTV72 boundary in

Table 5.2: The percentage of predicted decisions on the held-out test set that satisfy each hidden constraint of Geretschläger et al. (2015) to 1 Gy relaxation. The best performing models on the summary statistics are highlighted.

Structure	Criteria (Gy)	Baseline	IPMAN (λ)			
		GAN	256	64	16	4
Right Parotid	$z_o \leq 26$	83.3	85.7	85.7	100	100
Left Parotid	$z_o \leq 26$	70.0	50.0	60.0	100	100
Larynx	$z_o \leq 45$	93.3	86.7	76.8	100	100
Mandible	$z_o^{\max} \leq 73.5$	100	81.0	90.5	100	100
PTV72	$z_t^{90} \geq 72$	7.31	95.2	95.2	14.3	0
PTV66	$z_t^{90} \geq 66$	77.8	96.3	96.3	85.2	0
PTV54	$z_t^{90} \geq 54$	100	100	100	96.0	0
All hidden constraints		18.3	64.0	66.0	26.0	0
All polyhedral constraints		93.8	94.0	90.0	88.0	0
Objective function value		40.3	42.3	42.3	34.3	10.0

Figure 5.5(b) begins to increase from the 6th iteration and passes 0 (i.e., satisfy the hidden constraint) by the 10th iteration. This result demonstrates that our model is learning a new constraint, the PTV72 hidden criteria, which cannot be learned by naively using the training data. Recall that no clinical dose in the original data set reached 72 Gy on the PTV70. Thus, learning the PTV72 constraint is entirely attributable to the IPMAN procedure.

Table 5.2 shows the performance on the held-out test set for the generator at iteration 11. The baseline for comparison is the previous GAN which was trained on data from the original institution. Since no new data exists, the GAN cannot be re-trained to recognize the updated clinical criteria. As a result, few plans (18.3%) produced by the GAN satisfy the hidden constraints with only 77.8% and 7.31% of the PTV66 and PTV72 criteria being satisfied, respectively. In contrast, IPMAN is able to learn the new hidden constraints. Overall, hidden constraint satisfaction is 64.0% with 96.3% and 95.2% of plans satisfy the PTV66 and PTV72, constraints, respectively with $\lambda = 256$ or 64. Nevertheless, as a result of learning higher doses to the PTVs, constraint satisfaction on the OARs slightly degrades, which is noted in the slightly higher objective function values.

5.8 Conclusion

Conventional optimization techniques generally require well-structured problem formulations and make limited account of auxiliary data present in problems where different instances must be regularly solved. We propose Interior Point Methods with Adversarial Networks, a learning-based approach for generating solutions to optimization problems whose feasible sets are determined by instance-specific auxiliary information. We develop an unconstrained barrier problem where the barrier is replaced by a classifier trained on historical instances to predict feasibility. Because a classifier is not perfectly accurate, we extend the theory of interior point methods to the setting where only a relaxation of the feasible set is known and develop a corresponding optimality guarantee. Our main algorithm iteratively trains the classifier as well as a generative model via empirical risk minimization of the barrier problem. We demonstrate that the classifier learns to better approximate an effective barrier and the generative model learns to predict solutions with an optimality guarantee for both in-sample and out-of-sample instances. Ultimately, we obtain a deep learning model that can predict optimal solutions to problems in a fraction of the time that it would take a conventional optimization solver. Furthermore, our predictions account for instance-specific variations in the feasible set that conventional optimization would fail to permit.

To illustrate the application of our algorithm, we use it to predict dose distributions for radiation therapy as part of an automated planning pipeline. We find that our method learns to predict doses that better satisfy hidden clinical constraints and minimize objective function values as compared to state-of-the-art baseline learning methods. Furthermore, we show that our approach is adaptable in learning clinical criteria that are different from those that were used to generate the ground truth doses. This result suggests that an institution without a sufficient data set for training a dose prediction model could apply our methodology using data from another clinic; our approach would learn to produce appropriate doses tailored to the unique clinical criteria of the new institution while ensuring all solutions are certifiably optimal. As the global demand for radiation therapy grows and new clinics open in rural and developing areas, such adaptable automated planning methodologies have the potential to close the supply-demand gap in treatment planning capacity.

Chapter 6

Sampling from the complement of a polyhedron

High-dimensional sampling is a fundamental tool in machine learning (Andrieu et al., 2003), optimization (Bertsimas and Vempala, 2004), and stochastic modeling (Ripley, 2009). Sampling from a high-dimensional set is a key component of approximation algorithms for problems that cannot be tractably solved with conventional methods.

The literature on high-dimensional sampling primarily addresses the problem of efficiently sampling points that lie within a convex set, with the family of Markov Chain Monte Carlo (MCMC) sampling methods being the most commonly used approach in this setting (Brooks et al., 2011). Recent applications in ranking have also generated interest in the related problem of sampling from the boundary of convex sets (Dieker and Vempala, 2015). However, to the best of our knowledge, there has not been prior work on sampling from the complement of a convex set.

In this chapter, we consider the task of efficiently sampling from sets defined by the complement of a polyhedron for which there exist many potential applications. For example, the complement operator can be used to represent disjunctions, which when combined with conjunctions, can describe arbitrary sets. Moreover, both disjunctive sets and MCMC sampling are common tools in mixed-integer programming (Balas, 1979; Huang and Mehrotra, 2013). Additional applications exist in data-driven optimization, where historical decision data can inform the construction of optimization models. For example in data-driven robust optimization, an uncertainty set around the parameters of a constraint is created by analyzing prior instances (Bertsimas et al., 2015).

This chapter is motivated by the problem of procuring an initial infeasible data set in Chapter 5. There, our goal was to construct a barrier function for use in an interior point method where we do not know the true feasible set, but are instead given a relaxation

and a data set of feasible decisions. We constructed a binary classifier that predicts whether a decision is feasible or not. In this chapter, we introduce a data augmentation procedure to help the training of such a classifier. By sampling from a known subset of the infeasible region, i.e., the complement of the relaxation, we augment our feasible data set with unimplemented decisions.

While the complement of a polyhedron is a non-convex set and therefore difficult for conventional sampling techniques, our key methodological contribution is to propose an efficient MCMC algorithm based on sampling from the boundary of convex sets. We prove this algorithm covers the entire complement and that it is sufficient to train a binary classifier that learns to distinguish between feasible and infeasible points. We perform numerical experiments over a set of optimization problems with a hidden feasible set and a polyhedral relaxation. For each problem, we use a prior set of feasible decisions and a sampled set of infeasible decisions to train a classifier. We compare our approach with *unsupervised* density estimation baselines that do not use infeasible decision data and show that our approach is essential for creating classifiers that (i) perform well when a tight separating boundary between feasible and infeasible regions is required; and (ii) when the data set of feasible decisions is small. Our experiments over linearized relaxations of MIPLIB (miplib2017) demonstrate that our sampling-based classifier significantly outperforms all baseline models. Code for our experiments are available at `https://github.com/rafidrm/mcmc-complement`.

Although our focus is on polyhedra, our approach can be adapted to non-linear sets similar to how sampling from the boundary of a polyhedron generalizes to the boundary of arbitrary convex sets. In Appendix D, we explore how to sample from the complement of an ellipsoid and prove that our algorithm also covers the complement in this setting. As a result, we demonstrate that our MCMC algorithm has more general applicability, and can be applied to, for instance, problems in robust optimization which commonly involve ellipsoidal uncertainty sets (Bertsimas et al., 2011).

6.1 Background

Consider a polyhedron $\mathcal{P} := \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{a}_m^\mathsf{T}\mathbf{x} \leq b_m, 1 \leq m \leq M\}$. There exist several algorithms for sampling from the interior $\text{int}(\mathcal{P})$, with the most well-known being the Hit-and-Run (HR) algorithm (Smith, 1984). Similarly, the Shake-and-Bake (SB) algorithm is the most well-known approach to sampling from the boundary $\text{bd}(\mathcal{P})$ (Boender et al., 1991). These algorithms fall under the family of MCMC techniques which operate by constructing a sequence of points governed by a proposal function. The sequence

describes a Markov chain whose stationary distribution reflects the desired properties encoded by the proposal function (e.g., HR and SB converge to uniform distributions supported over $\text{int}(\mathcal{P})$ and $\text{bd}(\mathcal{P})$, respectively). Our MCMC approach for sampling from $\mathbb{R}^n \setminus \mathcal{P}$ is based on the SB algorithm.

SB operates on principles of stochastic billiards; intuitively, a ball bounces from each facet of the polyhedron to other facets with the points of contact being the generated points. The algorithm is as follows: assume an initial point $\mathbf{w}_0 \in \text{bd}(\mathcal{P})$ that lies on a single facet. That is, there is a unique m for which $\mathbf{a}_m^\mathsf{T}\mathbf{w}_0 = b_m$ and $\mathbf{a}_{m'}^\mathsf{T}\mathbf{w}_0 < b_{m'}$ for all $m' \neq m$. At every iteration of SB, given a boundary point $\mathbf{x}$ lying on the m-th facet, sample a *feasible direction* vector $\mathbf{r} \in \mathcal{R}_m := \{\mathbf{r} \mid \mathbf{a}_m^\mathsf{T}\mathbf{r} \leq 0, \|\mathbf{r}\| = 1\}$ according to some direction probability distribution $p_\mathbf{r}(\mathbf{r}|\mathbf{w})$. Then, calculate the nearest boundary point $\mathbf{w}' \in \text{bd}(\mathcal{P})$ from $\mathbf{w}$ in the direction of $\mathbf{r}$. This new point is selected as the next point in the Markov chain according to move probability $p_{\mathbf{w}'}(\mathbf{w}'|\mathbf{w})$. If $\mathbf{w}'$ is not selected, then the Markov chain does not update and the iteration repeats. Let N be the total number of iterations. In their seminal work on SB, Boender et al. (1991) proved that (i) the algorithm ensures (almost surely) every point on the Markov chain $\{\mathbf{w}_j\}_{j=1}^N$ lies on a unique facet of $\mathcal{P}$, and (ii) the Markov chain has a stationary uniform distribution over $\text{bd}(\mathcal{P})$.

There exist several variants of the SB algorithm that differ in their choices for the direction and move probabilities (Boender et al., 1991). The two most common variants are the Original SB and the Running SB. In the Original SB, the direction probability is uniform over the half-space defined by the facet $\mathcal{R}_m$. This leads to a move probability proportional to the angles of incidence. On the other hand, for the Running SB, the direction probability is chosen such that the algorithm moves in every iteration, i.e., $p_{\mathbf{w}'}(\mathbf{w}'|\mathbf{w}) = 1$ for all $\mathbf{w}', \mathbf{w} \in \text{bd}(\mathcal{P})$. In this work, we consider the Original SB algorithm due to its simplicity in calculating the direction probabilities.

6.2 Sampling from the complement of a polyhedron

Assume that $\mathcal{P}$ is full-dimensional and non-empty. Given a polyhedron $\mathcal{P}$, we generate a sequence of N points $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$ such that $\mathcal{D} \subset \mathbb{R}^n \setminus \mathcal{P}$. In each iteration of SB, a direction vector is sampled and the next point on the boundary is found by moving in the given direction from the current point. Notice, however, that moving in the negative direction from the current point yields points that lie in the complement of the polyhedron, i.e., $\mathbf{x} \in \mathbb{R}^n \setminus \mathcal{P}$.

In our algorithm, we treat SB as a Hidden Markov chain. In each iteration, the

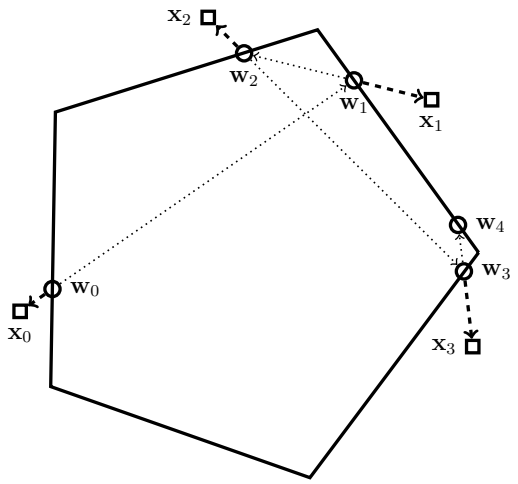


Figure 6.1: A sample sequence of points generated from the Complement SB algorithm. ◯ and □ are points on the boundary and infeasible points respectively.

Algorithm 2 Complement Shake-and-Bake

Require: Polyhedron $\mathcal{P} = \left\{\mathbf{x} \mid \mathbf{a}_m^\mathsf{T}\mathbf{x} \leq b_m, m = 1, \ldots, M\right\}$; Sampling distributions $p_{\mathbf{w}}(\mathbf{w}'|\mathbf{w})$, $p_{\mathbf{r}}(\mathbf{r}|\mathbf{w})$, $p_\xi(\xi|\mathbf{r}, \mathbf{w})$, Number of points N; Initialization $\hat{\mathbf{w}}_i \in \text{bd}(\mathcal{P}), i = 1$, $\mathcal{D} = \emptyset$

for $i = 1$ **to** N **do**

Randomly sample $\mathbf{r}_i \sim p_{\mathbf{r}}(\mathbf{r}|\mathbf{w}_i)$ and $\xi_i \sim p_\xi(\xi|\mathbf{r}, \mathbf{w})$.

Update data set $\mathcal{D} \leftarrow \mathcal{D} \cup \left\{\mathbf{w}_i - \xi_i \mathbf{r}_i\right\}$.

Let $\theta \in \min_m \left\{ \frac{b_m - \mathbf{a}_m^\mathsf{T}\mathbf{w}_i}{\mathbf{a}_m^\mathsf{T}\hat{\mathbf{r}}_i} > 0 \right\}$.

With probability $p_{\mathbf{w}}(\mathbf{w}_i + \theta\mathbf{r}|\mathbf{w}_i)$, update $\mathbf{w}_{i+1} \leftarrow \mathbf{w}_i + \theta\mathbf{r}_i$ and increase $i \leftarrow i + 1$, else $\mathbf{w}_{i+1} = \mathbf{w}_i$.

end for

previous boundary point is the hidden state and the observed state in the complement, $\mathbf{x}$, is generated according to the random direction vector that is sampled. Assume that the direction and move probabilities $p_{\mathbf{r}}(\mathbf{r}|\mathbf{w})$ and $p_{\mathbf{w}}(\mathbf{w}'|\mathbf{w})$ are such that $\{\mathbf{w}_i\}_{i=1}^N$ is a Markov chain of points on $\text{bd}(\mathcal{P})$. If we sample a random scale variable $\xi \sim p_\xi(\xi|\mathbf{r}, \mathbf{w})$ according to some positive distribution function, then $\mathbf{x} = \mathbf{w} - \xi\mathbf{r} \in \mathbb{R}^n \setminus \mathcal{P}$. Figure 6.1 shows a sample path of the chain $\{(\mathbf{w}_i, \mathbf{x}_i)\}_{i=1}^N$ which includes the hidden states. A detailed description of the MCMC algorithm is presented in Algorithm 2.

Our algorithm enjoys all of the computational benefits of the original SB algorithm since we recycle the direction vectors $\mathbf{r}$ and only reverse the signs to ensure the generation of infeasible points. Generating the scale variable ξ is the only additional computation. We refer to Boender et al. (1991) for details on $p_{\mathbf{r}}(\mathbf{r}|\mathbf{w})$ and $p_{\mathbf{w}}(\mathbf{w}'|\mathbf{w})$. Any absolutely continuous distribution $p_\xi(\xi)$ supported over $(0, \infty)$ will suffice. Given an appropriate

choice of $p_\xi(\xi|\mathbf{r},\mathbf{w})$, we show that Algorithm 2 covers all of $\mathbb{R}^n \setminus \mathcal{P}$. That is, any measurable region of $\mathbb{R}^n \setminus \mathcal{P}$ has positive stationary probability.

Theorem 9. *Let μ_n denote the n-dimensional Lebesgue measure on a set. If $p_\xi(\xi|\mathbf{r},\mathbf{w}) > 0$ for all $\xi \in (0,\infty)$ and $\mathbf{r},\mathbf{w} \in \mathbb{R}^n$, then for any initial point $\mathbf{w}_0$ and any μ_n-measurable subset $\mathcal{A} \subset \mathbb{R}^n \setminus \mathcal{P}$,*

$$\lim_{N\to\infty} \mathbb{P}\big\{\mathbf{x}_N \in \mathcal{A} \;\big|\; \mathbf{w}_0\big\} > 0. \tag{6.1}$$

Proof. Without loss of generality, let $\tilde{\mathbf{r}} = \xi\mathbf{r}$ and $p_{\tilde{\mathbf{r}}}(\tilde{\mathbf{r}}|\mathbf{w}) = p_{\mathbf{r}}(\mathbf{r}|\mathbf{w})p_\xi(\xi|\mathbf{r},\mathbf{w})$. Let $p_{SB}(\mathbf{w})$ denote the stationary distribution of the hidden state SB algorithm. Let $\mathcal{A}' \subset \mathbb{R}^n \setminus \mathcal{X}$ denote a μ_n measurable set for which there exists m such that $\mathbf{a}_m^\mathsf{T}\mathbf{x} > b_m$ for all $\mathbf{x} \in \mathcal{A}'$. We first prove (6.1) for all $\mathcal{A}'$ with this specific structure and show that any $\mathcal{A} \subset \mathbb{R}^n \setminus \mathcal{X}$ contains a subset $\mathcal{A}' \subset \mathcal{A}$. Then, the probability for $\mathcal{A}'$ is a lower bound, i.e., $\mathbb{P}\big\{\mathbf{x}_N \in \mathcal{A} \mid \mathbf{w}_0\big\} \geq \mathbb{P}\big\{\mathbf{x}_N \in \mathcal{A}' \mid \mathbf{w}_0\big\}$, completing the proof.

Consider a set $\mathcal{A}'$ with the proposed structure. We will construct two measurable sets $\mathcal{W}$ and $\tilde{\mathcal{R}}(\mathbf{w})$ such that

$$\Big\{\mathbf{w} - \tilde{\mathbf{r}} \;\Big|\; \mathbf{w} \in \mathcal{W}, \tilde{\mathbf{r}} \in \tilde{\mathcal{R}}(\mathbf{w})\Big\} \subseteq \mathcal{A}'.$$

Given their existence, we can bound

$$\begin{aligned}\lim_{N\to\infty} \mathbb{P}\big\{\mathbf{x}_N \in \mathcal{A}' \;\big|\; \mathbf{w}_0\big\} &\geq \lim_{N\to\infty} \int_{\mathcal{W}} \mathbb{P}\big\{\mathbf{w}_N - \tilde{\mathbf{r}}_N \in \mathcal{A}' \big| \mathbf{w}_N\big\} p_{SB}(\mathbf{w}_N) d\mathbf{w}_N \\ &\geq \lim_{N\to\infty} \int_{\mathcal{W}} \int_{\tilde{\mathcal{R}}(\mathbf{w}_N)} p_{\tilde{\mathbf{r}}}(\tilde{\mathbf{r}}_N|\mathbf{w}_N) p_{SB}(\mathbf{w}_N) d\tilde{\mathbf{r}}_N d\mathbf{w}_N.\end{aligned}$$

First, for some fixed $\epsilon > 0$, let

$$\mathcal{W} := \left\{\mathbf{w} \;\middle|\; \mathbf{a}_m^\mathsf{T}\mathbf{w} = b_m, \mathbf{a}_{m'}^\mathsf{T}\mathbf{w} < b_{m'} - \epsilon, \forall m' \neq m\right\}.$$

Because $\mathcal{X}$ is closed and bounded with a non-empty interior, this set must exist for some $\epsilon > 0$. Furthermore, from Boender et al. (1991, Lemma 2), there exists $\varepsilon > 0$ for which $\mathcal{W}$ has positive $(n-1)$-Lebesgue measure, i.e., $\mu_{n-1}(\mathcal{W}) > 0$, and thus, $p_{SB}(\mathcal{W}) > 0$. We avoid degenerate m by assuming that $\mathcal{X}$ has no redundant constraints.

Next for any $\mathbf{w}_N \in \mathcal{W}$, let

$$\tilde{\mathcal{R}}(\mathbf{w}_N) := \left\{\mathbf{w}_N - \mathbf{x} \;\middle|\; \mathbf{x} \in \mathcal{A}'\right\}.$$

Because $\mu_n(\mathcal{A}') > 0$ and $\tilde{\mathcal{R}}(\mathbf{w}_N)$ is a translation, $\mu_n(\tilde{\mathcal{R}}(\mathbf{w})) > 0$ must hold as well. It remains to show that $p_{\tilde{\mathbf{r}}}(\tilde{\mathbf{r}}_N|\mathbf{w}_N) = p_{\mathbf{r}}(\mathbf{r}_N|\mathbf{w}_N)p_{\xi}(\xi_N|\mathbf{r}_N, \mathbf{w}_N) > 0$ for all $\tilde{\mathbf{r}} \in \tilde{\mathcal{R}}(\mathbf{w}_N)$. Since $\mathbf{a}_m^{\mathsf{T}}(\mathbf{w}_N - \mathbf{x}) < 0$ for all $\mathbf{x} \in \mathcal{A}'$, the normalized vector $\mathbf{r}_N = (\mathbf{w}_N - \mathbf{x})/\left\|\mathbf{w}_N - \mathbf{x}\right\|$ is a valid direction for the SB algorithm and $p_{\mathbf{r}}(\mathbf{r}_N|\mathbf{w}_N) > 0$. Furthermore by assumption in the Theorem statement, $p(\xi_N|\mathbf{r}_N, \mathbf{w}_N) > 0$ for $\xi_N = \left\|\mathbf{w}_N - \mathbf{x}\right\| > 0$. Therefore $p_{\tilde{\mathbf{r}}}(\tilde{\mathbf{r}}_N|\mathbf{w}_N) > 0$.

We now extend the proof to any arbitrary set $\mathcal{A} \subset \mathbb{R}^n \setminus \mathcal{X}$ such that $\mu_n(\mathcal{A}) > 0$. Let $\sigma(\{1, \ldots, M\})$ denote the power set, i.e., the set of all subsets of $\{1, \ldots, M\}$. Then, $\mathcal{A}$ can be written as a union of a finite number of disjoint subsets:

$$\mathcal{A} = \bigcup_{\mathcal{M} \subseteq \sigma(\{1,\ldots,M\})} \left\{ \mathbf{x} \in \mathcal{A} \;\middle|\; \mathbf{a}_m^{\mathsf{T}}\mathbf{x} > b_m, \mathbf{a}_{m'}^{\mathsf{T}}\mathbf{x} \leq b_{m'}, \forall m \in \mathcal{M}, \forall m' \notin \mathcal{M} \right\}.$$

Since $\mathcal{A}$ is measurable, at least one of the subsets is also measurable. Furthermore, each of the subsets can be characterized in the form $\mathcal{A}'$, i.e., all points violating a specific constraint. Because the subsets are disjoint, the probability of $\mathcal{A}$ is exactly equal to the sum of the probabilities of the individual subsets, and therefore is positive. □

Theorem 9 extends the main result from Boender et al. (1991) which proves that the SB algorithm generates a stationary distribution which covers the entire boundary uniformly. Here, we show that the Complement SB algorithm generates a stationary distribution which covers the entire complement region. We remark that Algorithm 2 specifically applies to polyhedral sets $\mathcal{X}$ and can be extended to convex sets defined by non-linear constraints which are prominent in many constrained optimization applications. In particular, in Appendix D.1, we demonstrate that a modified version of Algorithm 2 can be used to generate points in the complement of ellipsoidal and spherical feasible sets, thereby extending the main result from McDonald (1989).

Theorem 9 is useful for learning to classify a hidden feasible set. Consider an optimization problem over a feasible set $\hat{\mathcal{X}}$ that is not known a priori. Instead, we have a data set of feasible decisions $\hat{\mathcal{D}} \sim \hat{\mathbb{P}}$ drawn i.i.d. from a distribution supported over $\hat{\mathcal{X}}$. We also have a relaxation of the feasible set $\mathcal{P}$. Using Algorithm 2, we can generate a data set of infeasible decisions $\mathcal{D} \subset \mathbb{R}^n \setminus \mathcal{P}$ with steady state distribution $\mathbb{P}$. With this augmented data set of feasible and infeasible points, we then train a binary classifier $D(\mathbf{x}) : \mathbb{R}^n \to \{0, 1\}$ such that $D(\mathbf{x}) = 1$ for all $\mathbf{x} \in \hat{\mathcal{X}}$ and $D(\mathbf{x}) = 0$ for $\mathbf{x} \in \mathbb{R}^n \setminus \hat{\mathcal{X}}$. Suppose that we minimize the Binary Cross Entropy loss (Goodfellow et al., 2016):

$$\min_{D} \; -\mathbb{E}_{\hat{\mathbf{x}} \sim \hat{\mathbb{P}}}\Big[\log D(\hat{\mathbf{x}})\Big] - \mathbb{E}_{\mathbf{x} \sim \mathbb{P}}\Big[\log\big(1 - D(\mathbf{x})\big)\Big].$$

Recall Lemma 2 (Chapter 5), which stated that for a binary classifier to provably predict points with perfect accuracy, the data distributions of those points must have closed and disjoint supports. We revise the prior result to this setting in order to demonstrate that it is possible to construct a classifier that approximates the feasible set.

Proposition 1. *Consider $\hat{\mathbb{P}}$ supported over a closed set $\hat{\mathcal{X}}$. Let $\mathcal{X}$ be a closed polyhedron such that $\hat{\mathcal{X}} \subset \text{int}(\mathcal{X})$ and let $\mathbb{P}$ be the steady state distribution of the Markov chain generated by Algorithm 2 over $\mathcal{X}$. Then the optimal classifier $D^*(\mathbf{x})$ satisfies $D^*(\mathbf{x}) = 1$ for $\mathbf{x} \in \hat{\mathcal{X}}$ and $D^*(\mathbf{x}) = 0$ for $\mathbf{x} \in \mathbb{R}^n \setminus \mathcal{X}$.*

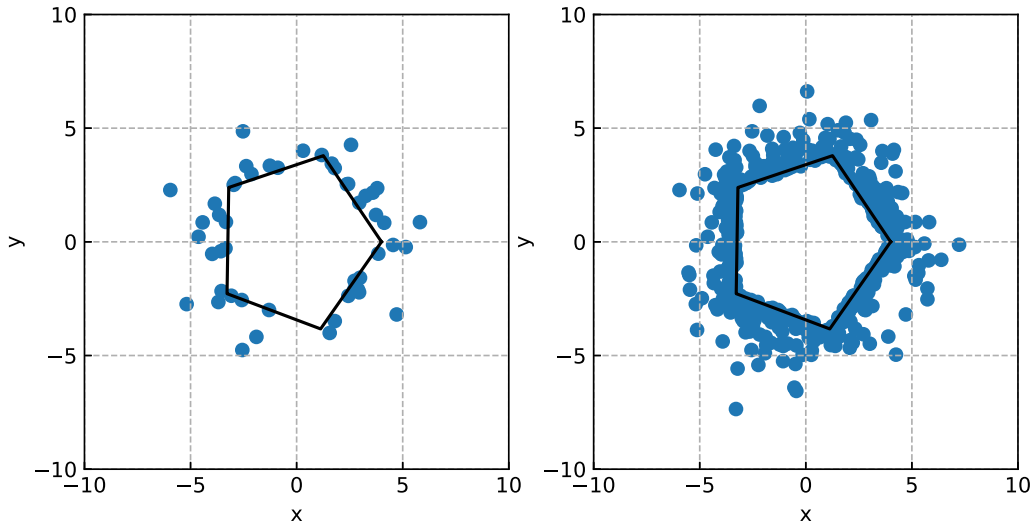
Proof. From Theorem 9, the steady state distribution satisfies $\mathbb{P}\{\mathcal{A}\} > 0$ for any measurable $\mathcal{A} \in \mathbb{R}^n \setminus \mathcal{X}$. Thus, the steady state distribution is supported over the entire $\mathbb{R}^n \setminus \mathcal{X}$. Note that $\hat{\mathcal{X}}$ and $\mathbb{R}^n \setminus \mathcal{X}$ are disjoint. From Lemma 2, the optimal classifier perfectly separates the two supports. □

Proposition 1 states that given training data (i.e., $\hat{\mathbb{P}}$ supported over a hidden set $\hat{\mathcal{X}}$ and $\mathbb{P}$ supported over the complement of a polyhedron $\mathbb{R}^n \setminus \mathcal{X}$ that relaxes the hidden set), a classifier can learn to perfectly distinguish from a hidden set and its polyhedral relaxation. We remark that training via the Binary Cross Entropy loss function is not necessary and nearly any loss function will suffice. However, we seek to distinguish between $\hat{\mathcal{X}}$ and $\mathbb{R}^n \setminus \hat{\mathcal{X}}$ and the proposition is useful insofar as the relaxation $\mathcal{X}$ is relatively tight. In our numerical experiments, we demonstrate that the classifier does indeed accurately learn to identify points in $\hat{\mathcal{X}}$ as feasible and points in the unknown band $\mathcal{X} \setminus \hat{\mathcal{X}}$ as infeasible.

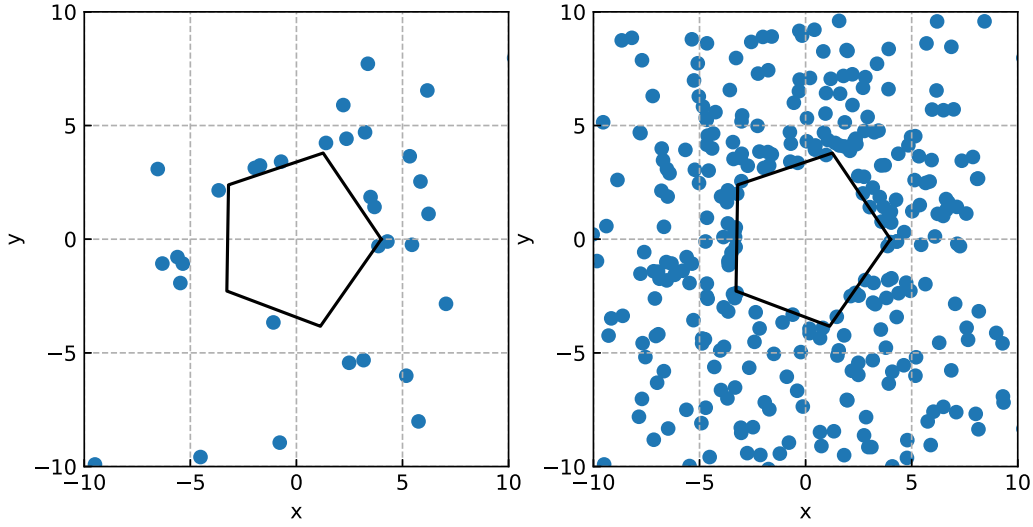
When implementing Algorithm 2, the choice of distribution $p_\xi(\xi)$ will depend on the application. In this work, we assume an Exponential distribution $p_\xi(\xi) = \text{Exp}(\lambda) = \lambda e^{-\lambda \xi}$. In practice when training a classifier $D(\mathbf{x})$ to learn a hidden feasible set, we do not have access to $\mathbb{P}$ and $\hat{\mathbb{P}}$ but rather data sets $\mathcal{D}$ and $\hat{\mathcal{D}}$. Given a limited data set, it is important for the classifier to accurately learn the regions near the boundary $\text{bd}(\mathcal{P})$ because the band near the boundary $\mathcal{P} \setminus \hat{\mathcal{X}}$ is the most challenging region to classify. Note that it is still important to generate some points far from the boundary in order to satisfy Proposition 1. Using an exponential distribution ensures that we generate points with high density near the boundary and low density further away. Figure 6.2 shows several stages of Algorithm 2 for different values of λ.

6.3 Numerical analysis

We implement Algorithm 2 and the corresponding SB-based classifier to learn the hidden feasible set $\hat{\mathcal{X}}$. Given a data set of feasible decisions $\hat{\mathcal{D}} = \{\hat{\mathbf{x}}_i\}_{i=1}^N \subset \hat{\mathcal{X}}$ and a relaxation



(a) $\lambda = 1$



(b) $\lambda = 10$

Figure 6.2: Sample of points generated using an Exponential distribution $p_\xi(\xi|\mathbf{r}, \mathbf{w}) = \text{Exp}(\lambda)$. The left and right plots show $N = 50$ and $N = 500$ samples, respectively.

$\mathcal{P}$, we augment or data set by sampling infeasible decisions before training an SB-based classifier to predict whether a decision is feasible or not with respect to $\hat{\mathcal{X}}$.

Classical approaches towards constructing a classifier $D(\mathbf{x})$ would not have a set of infeasible points $\mathcal{D}$ and would thus be forced to use some form of unsupervised or generative modeling. We implement two baseline models: a Gaussian Mixture Model (GMM) and Kernel Density Estimation (KDE). Both are generative modeling techniques that use $\hat{\mathcal{D}}$ to estimate a probability distribution over $\hat{\mathcal{X}}$.

In our first set of experiments, we consider a simulated fractional knapsack problem. We use this example to investigate the relative tightness of the relaxation and show that when the relaxation is a reasonable approximation of the hidden feasible set, our approach dominates the baseline models. We then investigate the effect that the size of the data set (i.e., $|\hat{\mathcal{D}}|$) has on the ability to learn the feasible set and show that by sampling from the infeasible region, our classifier achieves competitive performance with the unsupervised baseline models while requiring an order-of-magnitude less feasible data. Finally, we show that as the dimension of the problem increases, our approach still learns the hidden feasible set while the baseline models collapse.

Finally, we conduct experiments on linearizations over a set of MIPLIB problems that have less than 80 variables (miplib2017). Our SB-based classifier dominates the baseline models in terms of accuracy and F_1 score on nearly all instances, often by margins of 20%. Furthermore, we show that for challenging instances with a large number of variables, the baseline models once again completely collapse and either indiscriminately predict all test points as infeasible or all points as feasible. In contrast, the SB-based classifier still demonstrates learning even for these challenging problems.

6.3.1 Data and methods

Consider a hidden polyhedron $\hat{\mathcal{X}} = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{b}\}$. We first construct a relaxation $\mathcal{P} = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{b} - \mathbf{d}\}$, where $d_m \sim p_d(d)$ is a random perturbation variable. In order to ensure $\hat{\mathcal{X}} \subset \mathcal{P}$ and that $\mathcal{P}$ is a relatively close approximation to $\hat{\mathcal{X}}$, we use an Exponential distribution $p_d(d) = \text{Exp}(\gamma)$ with a scale parameter γ proportional to the polyhedral constraints, i.e.,

$$\gamma = \gamma_0 \max\{\|\mathbf{b}\|_\infty, \|\mathbf{a}_1\|_\infty, \|\mathbf{a}_2\|_\infty, \ldots, \|\mathbf{a}_M\|_\infty\}, \tag{6.2}$$

for a constant $\gamma_0 > 0$. This ensures that $\mathcal{P}$ is neither too tight nor too loose of a relaxation. We refer to γ as the degree of the relaxation.

For each instance, we use a HR sampler to generate feasible points $\hat{\mathcal{D}} = \{\hat{\mathbf{x}}_i\}_{i=1}^N \subset \hat{\mathcal{X}}$. Thus, $\mathcal{P}$ and $\hat{\mathcal{D}}$ constitute the available information used to learn $\hat{\mathcal{X}}$. Using Algorithm 2, we generate an "infeasible" data set $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N \subset \mathbb{R}^n \setminus \mathcal{P}$ and then train an off-the-shelf Gradient Boosted Tree (GBT) to classify between $\mathcal{D}$ and $\hat{\mathcal{D}}$. We do not tune hyper-parameters for our classifier finding that it outperforms the baseline models in most cases.

We consider two generative baseline models that estimate a probability distribution $\hat{p}(\mathbf{x})$ over $\hat{\mathcal{D}}$. That is, we define a threshold parameter $t = \min_{\hat{\mathbf{x}} \in \hat{\mathcal{D}}} \hat{p}(\hat{\mathbf{x}})$ as the small-

est probability such that the data set consists entirely of feasible decisions. Then, the baseline classifier applies a threshold rule over the generative model, i.e., $D(\mathbf{x}) = \mathbf{1}[\mathbf{x} \in \mathcal{P}]\mathbf{1}[\hat{p}(\mathbf{x}) \geq t]$. The first term in the classifier simply checks if the decision lies within the given relaxation $\mathcal{P}$. Thus, it is an intuitive approach to use the knowledge of the polyhedral relaxation. We implement two baseline models: a KDE and a GMM and cross-validate over their respective hyper-parameters using $\hat{\mathcal{D}}$. As these models typically do not scale efficiently to higher dimensions (Theis et al., 2016), we consider the use of Principal Component Analysis (PCA) to pre-process the training data and reduce the dimensionality of the problem. As a result, we implement all models with and without PCA for ablation.

In order to evaluate our approach, we generate an out-of-sample test set of feasible points $\hat{\mathcal{D}}' \subset \hat{\mathcal{X}}$ and infeasible decisions $\mathcal{D}' \subset \mathcal{P} \setminus \hat{\mathcal{X}}$. Both data sets are generated with an HR sampler. When generating $\mathcal{D}'$, we simply reject points in $\hat{\mathcal{X}}$ for the Markov chain. Note that we do not sample points in $\mathbb{R}^n \setminus \mathcal{P}$ for testing as they would be trivially identified as infeasible given our knowledge of $\mathcal{P}$. Our final out-of-sample test set is $\hat{\mathcal{D}}' \cup \mathcal{D}'$.

6.3.2 A fractional knapsack problem

Consider a fractional knapsack problem with the following feasible set $\hat{\mathcal{X}}$ and polyhedral relaxation $\mathcal{P}$:

$$\hat{\mathcal{X}} = \left\{ \mathbf{x} \;\middle|\; \sum_{i=1}^{n} x_i \leq 5, x_i \geq 0 \right\}$$
$$\mathcal{P} = \left\{ \mathbf{x} \;\middle|\; \sum_{i=1}^{n} x_i \leq 5 + d_0, x_i \geq -d_i \right\}.$$

We compare the SB-based supervised learning approach with the two generative baselines in three different scenarios. We first analyze the degree of relaxation γ to assess the algorithms' ability to learn under different relaxations. We then investigate how the size of the data set (N) affects the performance of the different algorithms. Finally, we assess the ability of the SB-based classifier to learn in n-dimensional spaces. Each experiment varies a single parameter while holding the others constant; we set $\gamma_0 = 0.1$ (i.e., $\gamma = 0.5$), $N = 200$, $n = 2$, as the default settings. We fix $\lambda = 0.5$ when generating $\mathcal{D}$ using Algorithm 2. All results are averaged from 50 trials.

Unsupervised learning techniques that rely only on $\hat{\mathcal{D}}$ to learn an approximation of the feasible set can mis-classify regions where there is no information. For example, regions outside $\hat{\mathcal{X}}$ but close to $\hat{\mathcal{D}}$ may be mis-classified as feasible since a KDE and

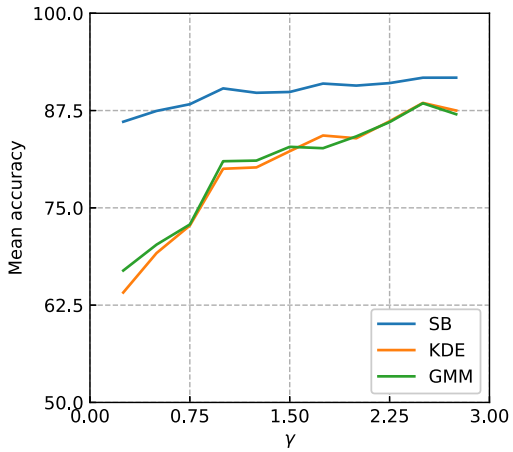


Figure 6.3: Mean accuracy of the models from increasing the degree of the relaxation γ.

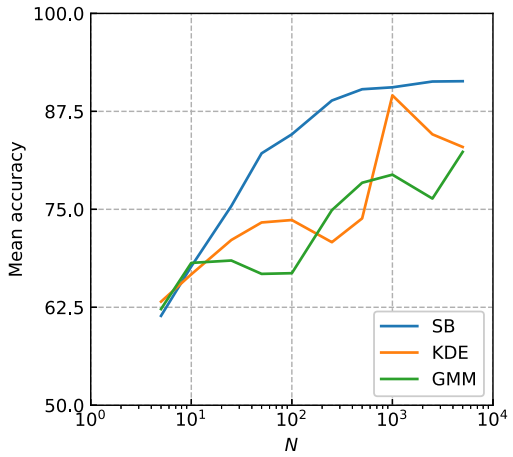


Figure 6.4: Mean accuracy of the models as we increase the training set size N.

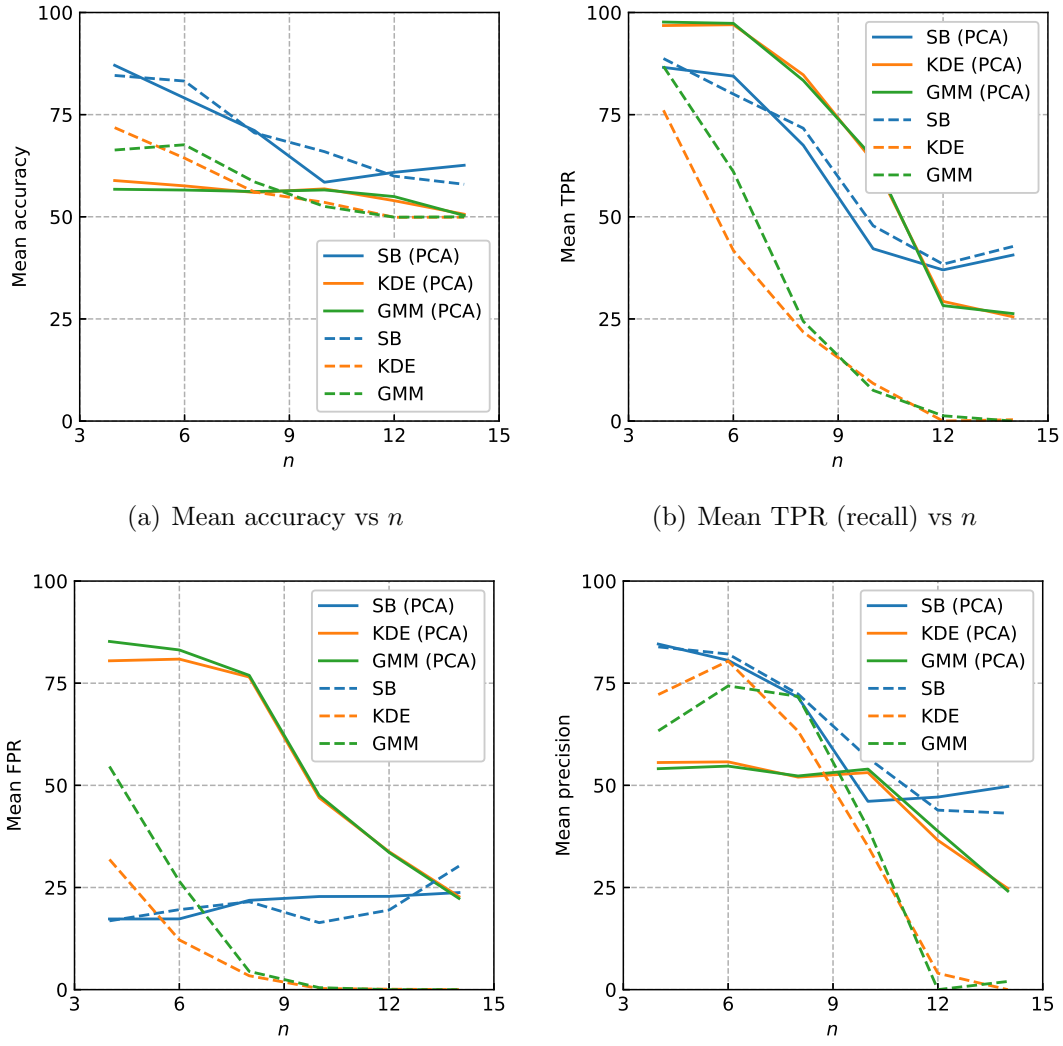
GMM would show a gradual drop in density from the data. Generating points in $\mathbb{R}^n \setminus \mathcal{P}$ offers a counter-balance to unsupervised techniques that incorrectly mis-classify infeasible regions as feasible. This effect is particularly prominent when the relaxation is tighter as demonstrated in Figure 6.3 which plots the out-of-sample accuracy as a function of γ. The SB-based classifier yields out-of-sample accuracy of approximately 91% regardless of the value of γ. The unsupervised baselines, in contrast, show poor out-of-sample accuracy when γ is small and slowly increase in performance as γ increases. Even at the largest value, $\gamma = 2.75$, the SB-based classifier still outperforms the baseline models. Thus, unsupervised learning only becomes competitive once the given relaxed bound is at least 50% larger than the true hidden bound.

The unsupervised learning baselines require significantly more data than our SB-based

classifier. Figure 6.4 plots out-of-sample accuracy as a function of N. When $N = 5$, all of the methods are equally poor and achieve approximately 63% out-of-sample accuracy. However, by using generated infeasible data, the SB-based classifier converges to 93% out-of-sample accuracy with $N = 100$. Note that the baseline models are non-monotone due to the grid-search algorithm used to find the best hyper-parameters of KDE and GMM, respectively. The optimal selection of these hyper-parameters change as we increase the amount of data and thus, the baseline models require extensive tuning. Nonetheless, even if we take the envelope of the KDE and GMM curves, we can still conclude that our sampling approach is on average an order-of-magnitude more data-efficient than the unsupervised baselines at learning the feasible set.

As previously shown, the unsupervised learning baselines require large data sets. As a result, in higher dimensions, these models assign small probabilities to regions where there may not be sufficient data. Thus, the differences in probabilities between regions where there are a small number of points and where there are no points can become negligible and it may appear as if these unsupervised learning models are applying nearly uniform (small) density over large areas. To address this issue, we pre-process the data that is used to train the baselines using PCA in order to reduce the dimension of the problem. While for $n \leq 8$, the baselines have better accuracy without PCA, reducing the dimensionality proves effective for $n \geq 9$. When n is low, using PCA leads to a loss of information for the unsupervised baselines.

More specifically, we consider increasing number of variables in the knapsack n, while holding $N = 200$. Because KDE and GMMs are known to perform poorly in high-dimensions, we use PCA in conjunction with the baseline models and our SB-based classifier to reduce the dimension by 25%. Figure 6.5 plots the accuracy, True Positive Rate (TPR) or recall, False Positive Rate (FPR), and precision over increasing n. Overall, the SB-based classifier (without PCA) strictly dominates all baselines on accuracy and precision. Furthermore, our classifier maintains a relatively flat FPR (around 25%) that scales slowly with the number of variables. All of the baselines converge to exactly 50% accuracy on the out-of-sample data demonstrating that they are not capable of learning the feasible set with the given amount of data. Note that for the baselines, TPR, FPR, and precision all decrease as n increases. When TPR and FPR are both 0, as in the case of the baselines with no PCA for $n \geq 12$, there are zero true and false positives and the models predict all points as infeasible.



(a) Mean accuracy vs n

(b) Mean TPR (recall) vs n

(c) Mean FPR vs n (lower is better)

(d) Mean precision vs n

Figure 6.5: Evaluating accuracy, TPR (recall), FPR, and precision of the different models as we increase the number of variables in the knapsack n. All models are are pre-processed using PCA to reduce the dimension by 25%.

6.3.3 Learning hidden feasible sets on MIPLIB instances

We next consider learning the feasible set of realistic benchmark problems, by drawing all instances of optimization problems with less than 80 variables from the MIPLIB database (miplib2017). We ignore problems marked "infeasible," those with more than 5000 constraints (e.g., `supportcase21i`), and those with large optimal values (e.g., `flugpl`), noting that these instances typically have pathological feasible sets.

For each instance, we use the LP relaxation of the feasible set and convert it to inequality form $\{\mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{b}\}$. However, these problems may yet have pathological

Table 6.1: Out-of-sample accuracy over instances of MIPLIB problems. We implement all models with and without PCA (reducing dimension by 50%). The best performing models per MIPLIB instance are highlighted.

Instance	Without PCA			With PCA		
	SB	KDE	GMM	SB	KDE	GMM
ej	90.6	97.3	94.2	85.5	84.5	82.3
gen-ip002	90.0	58.3	58.9	64.1	61.1	61.7
gen-ip016	53.0	50.0	50.0	47.4	61.0	61.0
gen-ip021	93.6	54.9	59.0	78.2	54.1	52.3
gen-ip036	95.2	56.7	63.6	85.1	61.3	60.3
gen-ip054	89.0	60.6	65.5	67.4	53.6	51.1
gr4x6	79.9	53.0	55.2	67.1	61.3	56.8
markshare_4_0	94.9	59.8	55.0	76.9	61.1	54.5
markshare_5_0	86.7	61.0	64.1	65.1	63.2	61.3
neos5	85.6	50.0	50.0	84.1	51.2	51.7

Table 6.2: Out of sample TPR, precision, and F_1-score over instances of MIPLIB problems. We draw the best-performing version of each model with respect to PCA. The best performing models in terms of F_1-score are highlighted.

Instance	TPR			Precision			F_1-score		
	SB	KDE	GMM	SB	KDE	GMM	SB	KDE	GMM
ej	99.9	99.6	99.9	84.7	95.2	90.0	91.7	97.4	94.7
gen-ip002	93.5	98.0	96.5	90.5	57.0	57.5	92.0	72.1	72.1
gen-ip016	12.2	27.1	27.1	23.7	35.7	35.6	16.1	30.8	30.8
gen-ip021	98.2	9.74	18.0	90.7	70.0	99.9	94.3	17.1	42.6
gen-ip036	99.8	99.6	34.2	91.9	56.6	86.3	95.7	72.2	49.0
gen-ip054	87.4	21.4	32.0	90.3	99.3	97.8	88.8	35.2	48.2
gr4x6	88.6	99.8	94.4	79.6	57.4	55.7	83.9	72.9	70.1
markshare_4_0	97.7	99.3	10.2	92.9	56.7	100	95.2	72.2	18.5
markshare_5_0	92.4	99.8	28.3	85.7	58.2	90.0	88.9	73.5	43.1
neos5	94.9	96.9	100	81.8	51.6	51.0	87.9	67.3	67.5

low-dimensional shapes. Consequently, we relax the right-hand-side terms by γ to obtain $\hat{\mathcal{X}} = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{b} - \gamma\mathbf{1}\}$. We set γ as in (6.2), with $\gamma_0 = 1$. We then construct hidden feasible sets $\mathcal{P} = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{b} - \gamma\mathbf{1} - \mathbf{d}\}$ where $d_m \sim p_d(d) = \text{Exp}(\gamma)$ with the same γ as before. In each experiment, we generate training sets of $N = 4000$ feasible points $\hat{\mathcal{D}} \subset \hat{\mathcal{X}}$ and infeasible points $\mathcal{D} \subset \mathbb{R}^n \setminus \mathcal{P}$. When generating $\mathcal{D}$, we use Algorithm 2 and fix $p_\xi(\xi) = \text{Exp}(1)$ for all instances. All results are averaged over 40 trials.

Table 6.1 shows the accuracy of each model on out-of-sample test sets with and

without the use of PCA (reducing the dimension by 50%). The best performing model for each MIPLIB instance is highlighted. The SB-based classifier outperforms all other models in nearly every instance and is often over 10% better than the best baseline model. Furthermore, the SB-based classifier performs better without the use of PCA. This is because $\hat{\mathcal{X}}$ possesses structure (i.e., linear constraints) that is useful for learning. Reducing the dimensionality of the problem through the application of PCA may result in a loss of important information for training. However, PCA is useful for the baselines and improves the performance of the KDE baseline on 10 instances. The GMM baseline sees a similar improvement for 7 instances.

For many instances, the baseline models achieve an accuracy level that is near 50%. Thus, we explore secondary metrics in order to understand the nature of these errors. Table 6.2 shows the out-of-sample TPR, precision, and F_1-score of the best performing SB-based classifier, KDE, and GMM, respectively. For the majority of the instances, the KDE baseline observes a TPR greater than 95% and precision less than 60% suggesting that the number of false negatives is relatively small but the number of false positives is approximately equal to the number of true positives. That is, the KDE baseline predicts nearly every test point to be feasible. We observe the opposite behavior for the GMM baseline in that TPR is small but precision is greater than 90%. That is, the GMM baseline predicts nearly every test point to be infeasible. Note that the SB-based classifier does not display these biases as can be observed by the F_1-score (which combines TPR and precision). Here, our classifier consistently dominates both of the baseline models.

6.4 Conclusion

We propose an MCMC method for sampling points in the complement of a polyhedron. We prove that our algorithm will eventually sample all points in the complement and demonstrate an application of our approach in a machine learning problem, i.e., augmenting data when learning a hidden feasible set using data from past implemented decisions. In a series of numerical experiments, we show that our method is more data-efficient and effectively scales to high dimensions as compared to the baseline models. We also show that it is more adept at learning to classify feasibility when the separating boundary is tight, as is a requirement in many optimization problems.

A potential extension of this work lies in sampling from the complement of sets that are prominent in other areas of constrained optimization. To this end, in the Appendix, we demonstrate that the Complement SB algorithm can be used to sample points from the complement of ellipsoidal and spherical feasible sets. However, these results require

several technical extensions. In future work, we hope to generalize our results and prove that the Complement SB algorithm generates a stationary distribution which covers the entire complement region for any arbitrary convex set.

Chapter 7

Conclusion

In this thesis, we construct personalized optimization models by by incorporating machine learning from data sets of past decisions. The methodological problem of learning to formulate optimization models can be decomposed along two dimensions: (i) what components of an optimization model to learn, and (ii) how to represent these learned components. This thesis addresses several configurations of these two questions:

- We estimate a parametric linear cost vector for an optimization model using an inverse optimization with an ensemble of decisions. We develop a framework that unifies prior techniques and admits assumption-free exact solution methods. To complement our framework, we develop a goodness-of-fit metric that provides insight into the quality of the imputed model.

- We explore optimization over contextual feasible sets that must be estimated from data. We develop a non-parametric model for such feasible sets and predict optimal decisions over these learned feasible sets. Our algorithm blends interior point methods with adversarial learning and our predicted decisions satisfy optimality guarantees for both in-sample and out-of-sample instances.

- We further explore our non-parametric model for the feasible set and introduce an MCMC sampling algorithm for data augmentation that generates infeasible decisions used to train our model. We demonstrate the effectiveness of this data augmentation for learning the feasible set in comparison to baselines.

We apply these techniques to automate radiation therapy treatment design. KBP is the prevailing framework for automated treatment planning and involves first estimating a dose that a clinician is likely to approve before optimizing a treatment that can deliver the dose. We make the following contributions:

- We propose the first ensemble KBP pipeline by ensembling the outputs of multiple dose generation models to an inverse optimization problem that estimates parameters for the treatment model. Each dose prediction is biased towards certain clinical metrics and the consensus obtained via ensembling better balances these criteria.

- We propose the first generative adversarial network for dose generation. Previous ML approaches used hand-tailored features to predict low-dimensional summaries of the dose. Deep learning treats the problem as a computer vision task and allows for predicting a 3-D dose distribution in one shot. GANs outperform prior KBP and conventional deep learning models in predicting doses that satisfy clinical metrics.

- We re-cast dose generation as an optimization problem of minimizing dose to healthy tissue while satisfying a hidden set of clinical criteria constraints. Our predicted doses outperform other deep learning models, including the GAN, on criteria satisfaction. Furthermore, our algorithm can learn the criteria of one clinic using data from another. As most clinics do not have the data to train custom models, this suggest an easier deployment alternative.

This research offers several directions for future methodological and applied work. We list potential avenues below:

- The inverse techniques developed in Chapter 3 address learning parametric linear objectives. Other inverse optimization frameworks in the multi-point literature have explored non-parametric objectives as well as parametric constraints. The general approach developed in this section can be fitted to learn general convex objectives as well as learning non-parametric models (e.g., Bertsimas et al., 2015). Similarly, we can also explore learning linear constraints in the ensemble setting (e.g., Mahmoudzadeh and Ghobadi, 2020).

- We implement our ensemble KBP pipeline by combining several dose generation models from the literature. However, a key result that we show is that the data set must be carefully selected in order to get the best out of the ensemble pipeline. Given that prior research has shown that certain dose generation models have affinities with certain plan optimization models (e.g., Babier et al., 2020b), there may be dose generation models that can generate a distribution of dose distributions for a patient. Such models would have some affinity for ensemble inverse optimization and therefore may lead to better treatment plans overall.

- Automated planning operates in two steps, by first converting a contoured CT image into a dose estimate and then converting the dose estimate into a treatment plan. Since dose generation can be recast into an optimization problem, the overall KBP pipeline may be improved by integrating dose generation and plan optimization into a single optimization problem over which IPMAN is run. Such an approach would yield an entirely IPMAN-based KBP pipeline that generates treatments in one shot and can significantly reduce the time cost of finalizing treatment plans. However, this extension relies on constructing a neural network model that is suitable for mapping from dose to beamlet vectors.

- A third extension to consider in automated planning is to remove the reliance on contoured CT images. Contouring is often a laborious task and there has been significant research in automating contouring via neural networks. Seeing as this task can be easily learned, it should be possible to implement a KBP pipeline that takes raw, uncontoured CT images to construct a treatment. If such a pipeline yields competitive performance with KBP techniques that use contoured images then the overall time to constructing treatments could be reduced by removing the requirement for manual contouring of CT images.

- The IPMAN algorithm of Chapter 5 is highly suited for training neural networks and other machine learning models that are trained via stochastic gradient descent-based algorithms. However, models that do not rely on gradient-based training such as decision trees and random forests cannot easily train using IPMAN. A key challenge for these models is that the problem, while unconstrained and differentiable, is often non-convex. Since tree-based models are often desirable due to their inherent interpretability, a valuable extension would be to update IPMAN so that it can be amenable for training decision trees. This extension would likely involve a modification of the Feasibility Classification Problem and the Generative Barrier Problem to an integer programming formulation.

- An important but implicit assumption throughout this work is that the data set of clinical treatment plans are independent samples obtained from some fixed distribution. In reality, the data is observational in that it is obtained via oncologist decisions that may have varied in motivation or have been affected by unobserved variables. This assumption does not hurt the results in Chapters 3 and 4 which do not rely on assumptions of the distribution, but may challenge the theory developed in Chapters 5. Consequently, a future direction of work is to update the IPMAN framework in a causal inference setting.

- Finally, we remark that the methods introduced here have been applied primarily to automated planning. However, applications of contextual optimization where decision-maker preferences are taken into account exist in various other domains. Future research will explore new applications towards learning to formulate and solve optimization problems.

Appendix A

Supplement to Chapter 3

A.1 A general solution method for $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$

Although Proposition 5 reformulates $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ into three sub-problems, the norm constraint $\|\cdot\|_N \geq K$ in the sub-problems adds two challenges: first, the constraint itself is non-convex, and second, an appropriate value for K must be chosen in order for Proposition 5 to hold. As the non-convex constraint can be handled by polyhedral decomposition, we first discuss how to choose a valid K. We then consider a relaxed reformulation of $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ that often works well in practice. Finally, we summarize all of these results into a general solution algorithm for inverse optimization minimizing the relative duality gap. These steps are summarized in Algorithm 3.

The proof of Proposition 5 shows that for any $K > 0$, every feasible solution of $\mathbf{GIO}^{+}_{\mathrm{R}}(\mathcal{D}; K)$, $\mathbf{GIO}^{-}_{\mathrm{R}}(\mathcal{D}; K)$, and $\mathbf{GIO}^{0}_{\mathrm{R}}(\mathcal{D}; K)$ can be mapped to a feasible solution of $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$. The normalization constraint $\|\mathbf{c}\|_N \geq K$ implies that the feasible region for each sub-problem grows as K decreases. The proof then shows that for some sufficiently small $K > 0$, an optimal solution to $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ can be mapped to a feasible (and therefore, also optimal) solution of one of (3.11)–(3.13).

To determine a sufficiently small K, note that the mapping of a solution of $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ to solutions of one of (3.11)–(3.13) involves scaling the solution by $\mathbf{b}^{\mathsf{T}}\mathbf{y}$, $-\mathbf{b}^{\mathsf{T}}\mathbf{y}$, or $\mathbf{y}^{\mathsf{T}}\mathbf{1}$, respectively. Bounding these terms allows us to determine a sufficiently small K. Formally, consider the following problem:

$$\begin{array}{ll} \underset{\mathbf{y}}{\text{maximize}} & \max\left\{|\mathbf{b}^{\mathsf{T}}\mathbf{y}|, \mathbf{y}^{\mathsf{T}}\mathbf{1}\right\} \\ \text{subject to} & \left\|\mathbf{A}^{\mathsf{T}}\mathbf{y}\right\|_N = 1,\ \mathbf{y} \geq \mathbf{0}. \end{array} \tag{A.1}$$

We refer to formulation (A.1) as the auxiliary problem for $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$. The auxiliary

problem can be written as three optimization problems, each with the same constraints as (A.1) but a different objective: $\mathbf{b}^\mathsf{T}\mathbf{y}$, $-\mathbf{b}^\mathsf{T}\mathbf{y}$, and $\mathbf{y}^\mathsf{T}\mathbf{1}$. Since the auxiliary problem has a normalization constraint similar to the one in $\mathbf{GIO}_\mathrm{A}(\mathcal{D})$, we can use the same methods to solve it. Let K^* be defined as the reciprocal of the optimal value of the auxiliary problem. Note that K^* is well-defined. That is, the auxiliary problem always has a non-zero solution, because any feasible $\mathbf{y}$ to (A.1) must have $\mathbf{y} \geq \mathbf{0}$ and at least one non-zero $y_i > 0$, meaning $\mathbf{y}^\mathsf{T}\mathbf{1} > 0$ must always hold. We use K^* to reformulate $\mathbf{GIO}_\mathrm{R}(\mathcal{D})$ to $\mathbf{GIO}^+_\mathrm{R}(\mathcal{D}; K^*)$, $\mathbf{GIO}^-_\mathrm{R}(\mathcal{D}; K^*)$, and $\mathbf{GIO}^0_\mathrm{R}(\mathcal{D}; K^*)$.

Theorem 10. *Let z^+ be the optimal value of $\mathbf{GIO}^+_\mathrm{R}(\mathcal{D}; K^*)$ if it is feasible, otherwise $z^+ = \infty$. Let z^- and z^0 be defined similarly for $\mathbf{GIO}^-_\mathrm{R}(\mathcal{D}; K^*)$ and $\mathbf{GIO}^0_\mathrm{R}(\mathcal{D}; K^*)$, respectively. Let $z^* = \min\{z^+, z^-, z^0\}$ and let $\left(\mathbf{c}^*, \mathbf{y}^*, \epsilon_1^*, \ldots, \epsilon_Q^*\right)$ be the corresponding optimal solution. Then, $\left(\mathbf{c}^*/\left\|\mathbf{c}^*\right\|_N, \mathbf{y}^*/\left\|\mathbf{c}^*\right\|_N, \epsilon_1^*, \ldots, \epsilon_Q^*\right)$ is optimal to $\mathbf{GIO}_\mathrm{R}(\mathcal{D})$.*

Proof. Let $(\hat{\mathbf{c}}, \hat{\mathbf{y}})$ be optimal to $\mathbf{GIO}_\mathrm{R}(\mathcal{D})$ and K be defined as in (3.14). Since $\hat{\mathbf{y}}$ is feasible for the auxiliary problem (A.1), $1/K^* \geq \max\left\{|\mathbf{b}^\mathsf{T}\hat{\mathbf{y}}|, \hat{\mathbf{y}}^\mathsf{T}\mathbf{1}\right\}$, implying $K^* \leq K$.

The proof of Proposition 5 showed that scaling $(\hat{\mathbf{c}}, \hat{\mathbf{y}})$ appropriately yielded a corresponding feasible solution to one of $\mathbf{GIO}^+_\mathrm{R}(\mathcal{D}; K)$, $\mathbf{GIO}^-_\mathrm{R}(\mathcal{D}; K)$, or $\mathbf{GIO}^0_\mathrm{R}(\mathcal{D}; K)$. Because $K^* \leq K$, the scaled solution must also be feasible for the respective $\mathbf{GIO}^+_\mathrm{R}(\mathcal{D}; K^*)$, $\mathbf{GIO}^-_\mathrm{R}(\mathcal{D}; K^*)$, or $\mathbf{GIO}^0_\mathrm{R}(\mathcal{D}; K^*)$. Moreover, every solution of the three $\mathbf{GIO}^+_\mathrm{R}(\mathcal{D}; K^*)$, $\mathbf{GIO}^-_\mathrm{R}(\mathcal{D}; K^*)$, or $\mathbf{GIO}^0_\mathrm{R}(\mathcal{D}; K^*)$ can be scaled to a feasible solution of $\mathbf{GIO}_\mathrm{R}(\mathcal{D})$, completing the proof. □

In the most general case, solving $\mathbf{GIO}_\mathrm{R}(\mathcal{D})$ is more computationally intensive than solving $\mathbf{GIO}_\mathrm{A}(\mathcal{D})$. We must first identify K^*, which we can use to reformulate $\mathbf{GIO}_\mathrm{R}(\mathcal{D})$ into three norm-constrained optimization problems. Subsequently, given an appropriate choice of $\|\cdot\|_N$, each problem is decomposed into a series of linear optimization problems. For instance, doing so leads to $2n$ linear programs if $\|\cdot\|_N = \|\cdot\|_\infty$ and 2^n LPs if $\|\cdot\|_N = \|\cdot\|_1$. These steps coupled with the auxiliary problem (A.1) used to determine K^* require the solution of $12n$ linear optimization problems when $\|\cdot\|_N = \|\cdot\|_\infty$, or $6(2^n)$ when $\|\cdot\|_N = \|\cdot\|_1$. In some cases, however, it may be possible to find an optimal solution to $\mathbf{GIO}_\mathrm{R}(\mathcal{D})$ by solving exactly three linear optimization problems.

Corollary 5. *Let $\mathbf{GIO}^+_{\mathrm{R,LP}}(\mathcal{D})$, $\mathbf{GIO}^-_{\mathrm{R,LP}}(\mathcal{D})$, and $\mathbf{GIO}^0_{\mathrm{R,LP}}(\mathcal{D})$ be the LP relaxations of $\mathbf{GIO}^+_\mathrm{R}(\mathcal{D}; K)$, $\mathbf{GIO}^-_\mathrm{R}(\mathcal{D}; K)$, and $\mathbf{GIO}^0_\mathrm{R}(\mathcal{D}; K)$, respectively, obtained by removing the normalization constraint $\|\mathbf{c}\|_N \geq K$. Let z^+_LP be the optimal value of $\mathbf{GIO}^+_{\mathrm{R,LP}}(\mathcal{D})$ if it is feasible, otherwise $z^+_\mathrm{LP} = \infty$. Let z^-_LP and z^0_LP be defined similarly for $\mathbf{GIO}^-_{\mathrm{R,LP}}(\mathcal{D})$ and $\mathbf{GIO}^0_{\mathrm{R,LP}}(\mathcal{D})$, respectively. Let $z^*_\mathrm{LP} = \min\left\{z^+_\mathrm{LP}, z^-_\mathrm{LP}, z^0_\mathrm{LP}\right\}$ and let $\left(\mathbf{c}^*, \mathbf{y}^*, \epsilon_1^*, \ldots, \epsilon_Q^*\right)$*

Algorithm 3 General solution method for $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$

Input: Data set $\mathcal{D}$

Output: Imputed model parameters $\left(\mathbf{c}^*, \mathbf{y}^*, \epsilon_1^*, \ldots, \epsilon_Q^*\right)$

1: Let $z_{\mathrm{LP}}^+ \leftarrow \mathbf{GIO}_{\mathrm{R,LP}}^+(\mathcal{D})$, $z_{\mathrm{LP}}^- \leftarrow \mathbf{GIO}_{\mathrm{R,LP}}^-(\mathcal{D})$, $z_{\mathrm{LP}}^0 \leftarrow \mathbf{GIO}_{\mathrm{R,LP}}^0(\mathcal{D})$ be the optimal values.
2: Let $z_{\mathrm{LP}}^* \leftarrow \min\left\{z_{\mathrm{LP}}^+, z_{\mathrm{LP}}^-, z_{\mathrm{LP}}^0\right\}$ and $\left(\mathbf{c}^*, \mathbf{y}^*, \epsilon_1^*, \ldots, \epsilon_Q^*\right)$ be the corresponding optimal solution.
3: **if** $\mathbf{c}^* \neq \mathbf{0}$ **then**
4: **return** $\left(\mathbf{c}^*, \mathbf{y}^*, \epsilon_1^*, \ldots, \epsilon_Q^*\right)$
5: **else**
6: Solve the auxiliary problem (A.1). Let K^* be the reciprocal of the optimal value.
7: Let $z^+ \leftarrow \mathbf{GIO}_{\mathrm{R}}^+(\mathcal{D}; K^*)$, $z^- \leftarrow \mathbf{GIO}_{\mathrm{R}}^-(\mathcal{D}; K^*)$, $z^0 \leftarrow \mathbf{GIO}_{\mathrm{R}}^0(\mathcal{D}; K^*)$ be the optimal values.
8: Let $z^* \leftarrow \min\left\{z^+, z^-, z^0\right\}$ and $\left(\mathbf{c}^*, \mathbf{y}^*, \epsilon_1^*, \ldots, \epsilon_Q^*\right)$ be the corresponding optimal solution.
9: **return** $\left(\mathbf{c}^*, \mathbf{y}^*, \epsilon_1^*, \ldots, \epsilon_Q^*\right)$
10: **end if**

be an optimal solution of the corresponding problem. If $\mathbf{c}^* \neq \mathbf{0}$*, then* z_{LP}^* *is equal to the optimal value of* $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ *and* $\left(\mathbf{c}^*/\left\|\mathbf{c}^*\right\|_N, \mathbf{y}^*/\left\|\mathbf{c}^*\right\|_N, \epsilon_1^*, \ldots, \epsilon_Q^*\right)$ *is an optimal solution to* $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$*.*

Proof. Let $(\hat{\mathbf{c}}, \hat{\mathbf{y}}, \hat{\epsilon}_1, \ldots, \hat{\epsilon}_Q)$ be an optimal solution to $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$. From Proposition 5, this solution can be rescaled to construct a feasible solution for one of $\mathbf{GIO}_{\mathrm{R,LP}}^+(\mathcal{D})$, $\mathbf{GIO}_{\mathrm{R,LP}}^-(\mathcal{D})$, and $\mathbf{GIO}_{\mathrm{R,LP}}^0(\mathcal{D})$ with the same objective value. Conversely, for each of the relaxed problems, let $(\tilde{\mathbf{c}}, \tilde{\mathbf{y}}, \tilde{\epsilon}_1, \ldots, \tilde{\epsilon}_Q)$ be a feasible solution. Assuming that $\tilde{\mathbf{c}} \neq \mathbf{0}$, this solution can be rescaled to construct $(\hat{\mathbf{c}}, \hat{\mathbf{y}}, \hat{\epsilon}_1, \ldots, \hat{\epsilon}_Q) = (\tilde{\mathbf{c}}/\left\|\tilde{\mathbf{c}}\right\|_N, \tilde{\mathbf{y}}/\left\|\tilde{\mathbf{c}}\right\|_N, \tilde{\epsilon}_1, \ldots, \tilde{\epsilon}_Q)$, which is a feasible solution to $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$ with the same objective value. Thus, if the minimum of $\mathbf{GIO}_{\mathrm{R,LP}}^+(\mathcal{D})$, $\mathbf{GIO}_{\mathrm{R,LP}}^-(\mathcal{D})$, and $\mathbf{GIO}_{\mathrm{R,LP}}^0(\mathcal{D})$ yields an optimal solution with a non-zero imputed cost vector, the two problems share the same optimal solution. □

The key difference between Proposition 5 and Corollary 5 is the non-zero assumption (i.e., $\mathbf{c}^* \neq \mathbf{0}$). By relaxing the normalization constraint, we permit potential solutions for which $\mathbf{c}^* = \mathbf{A}^\mathsf{T}\mathbf{y}^* = \mathbf{0}$ is a linearly dependent combination of the rows of $\mathbf{A}$. However, if $\mathbf{c}^* \neq \mathbf{0}$ is an optimal solution to the relaxed problem, it is also an optimal solution to $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$. Therefore, to solve $\mathbf{GIO}_{\mathrm{R}}(\mathcal{D})$, we suggest first solving the three relaxed problems, which are LPs, from Corollary 5. If $\mathbf{c}^* = \mathbf{0}$, then we use the more general approach. Section 3.4 (with details on the formulations in Appendix A.3) shows a case where the LP relaxations via Corollary 5 are sufficient.

A.2 Related work in inverse convex optimization

Multi-point inverse optimization has recently received significant interest under the setting of convex forward problems, with several notable inverse optimization models having been proposed for arbitrary convex forward problems (i.e., Aswani et al. (2018); Bertsimas et al. (2015); Esfahani et al. (2018)). The methods proposed in this prior work specialize to linear forward problems and overlap in formulation with the absolute duality and the decision space models proposed in this paper. However, the geometric nature of LPs poses new challenges, but also allows for some efficient solutions, that are not present in the strictly convex domain. In this section, we highlight the previous formulations and discuss several differences in the solution methods.

The inverse convex models in prior work assume that the data set consists of points corresponding to different forward problem instances. As we focus on inverse optimization for a fixed forward feasible set, we illustrate the results in the previous work for fixed $\mathcal{P}$.

A.2.1 Inverse variational inequality

Let $f(\mathbf{x};\mathbf{c}) : \mathbb{R}^n \to \mathbb{R}$ be a convex function in $\mathbf{x}$ parametrized by $\mathbf{c}$ and let $\mathcal{K}$ be a convex cone. Bertsimas et al. (2015) considered the forward conic optimization problem $\min_{\mathbf{x}} \{f(\mathbf{x};\mathbf{c}) \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \in \mathcal{K}\}$ and proposed an inverse optimization model that minimized the residuals from failing to satisfy the variational inequality of the first-order optimality condition. The inverse variational inequality problem is

$$\begin{aligned} \underset{\mathbf{c},\mathbf{y}_1,\ldots,\mathbf{y}_Q,\epsilon_1,\ldots,\epsilon_Q}{\text{minimize}} \quad & \sum_{q=1}^{Q} |\epsilon_q| \\ \text{subject to} \quad & \mathbf{A}^\mathsf{T}\mathbf{y}_q \leq_{\mathcal{K}} \nabla f(\hat{\mathbf{x}}_q;\mathbf{c}), \quad \forall q \in \mathcal{Q} \\ & \nabla f(\hat{\mathbf{x}}_q;\mathbf{c})^\mathsf{T}\hat{\mathbf{x}}_q - \mathbf{b}^\mathsf{T}\mathbf{y}_q \leq \epsilon_q, \quad \forall q \in \mathcal{Q} \\ & \mathbf{c} \in \mathcal{C}. \end{aligned} \tag{A.2}$$

Setting $\mathcal{K} = \mathbb{R}^n_+$, $f(\mathbf{x};\mathbf{c}) = \mathbf{c}^\mathsf{T}\mathbf{x}$, and $\mathcal{C} = \{\mathbf{c} \in \mathbb{R}^n \mid \|\mathbf{c}\|_N = 1\}$ makes formulation (A.2) equivalent to $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$, i.e., formulation (3.6).

In the original work, Bertsimas et al. (2015) focused mostly on strictly convex forward problems and on ensuring a convex inverse optimization formulation. While the nonconvex normalization constraint is not always necessary when the forward problem is strictly convex, setting $f(\mathbf{x};\mathbf{c}) = \mathbf{c}^\mathsf{T}\mathbf{x}$ implies that $(\mathbf{c},\mathbf{y},\epsilon_1,\ldots,\epsilon_Q) = (\mathbf{0},\mathbf{0},0,\ldots,0)$ is a trivially optimal solution (Chan et al., 2019; Esfahani et al., 2018). Note furthermore that convex normalization constraints exist in the literature, e.g., Keshavarz et al. (2011)

proposed setting $c_0 = 1$. However, these convex normalization constraints often bias the parameter space. For instance, setting $c_0 = 1$ prevents imputing non-trivial cost vectors where $c_0 = 0$. We enforce the non-convex constraint within all of the inverse optimization models in the current paper and propose polyhedral decomposition-based solution methods in the general setting for $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$. Furthermore, we find it important to explore special cases where the non-convexity can be bypassed, leading to simpler, sometimes analytic results (see Proposition 2 and 3, as well as Corollary 1).

Finally, Bertsimas et al. (2015) discussed a decision space alternative to formulation (A.2) where instead of the variational inequality residual, they minimized $\|\hat{\mathbf{x}}_q - \mathbf{x}_q\|$, where $\mathbf{x}_q$ is a variable that satisfies $f(\mathbf{x}_q; \mathbf{c}) = \mathbf{b}^{\mathsf{T}}\mathbf{y}$. Furthermore, they assumed that the gradient of the objective is strongly monotone, i.e., there exists $\gamma > 0$ such that

$$\big(\nabla f(\mathbf{x}; \mathbf{c}) - \nabla f(\mathbf{y}; \mathbf{c})\big)^{\mathsf{T}}\big(\mathbf{x} - \mathbf{y}\big) \geq \gamma \left\|\mathbf{x} - \mathbf{y}\right\|_2, \quad \forall \mathbf{x}, \mathbf{y} \in \mathcal{P}.$$

By focusing on the variational inequality nature of objective space inverse optimization, Bertsimas et al. (2015, Theorem 1) translated the variational inequality error bound of Pang (1987) to show that if there exists an solution $(\mathbf{c}^*, \mathbf{y}^*, \epsilon_1^*, \ldots, \epsilon_Q^*)$ to formulation (A.2), then there exists $\mathbf{x}_1^*, \ldots, \mathbf{x}_Q^*$ that are optimal solutions to the forward problem and satisfy $\left\|\hat{\mathbf{x}}_q - \mathbf{x}_q^*\right\|_2 \leq \sqrt{\epsilon_q/\gamma}$ for all q. That is, given the feasible solution to an objective space inverse optimization problem, we can obtain a corresponding feasible solution to a decision space problem where the error is bounded. Note, however, that in the linear case, $\nabla f(\mathbf{x}; \mathbf{c}) = \mathbf{c}$ does not satisfy the strong monotone property, i.e., $\gamma = 0$. As a result, the previous bound does not hold for inverse linear optimization.

A.2.2 Inverse risk minimization

Let $f(\mathbf{x}; \mathbf{u}, \mathbf{c}) : \mathbb{R}^n \to \mathbb{R}$ and $g(\mathbf{x}; \mathbf{u}, \mathbf{c}) : \mathbb{R}^n \to \mathbb{R}^m$ be convex functions in $\mathbf{x}$ that are parametrized by $\mathbf{u}$ and $\mathbf{c}$. Given $\min_{\mathbf{x}} \{f(\mathbf{x}; \mathbf{u}, \mathbf{c}) \mid g(\mathbf{x}; \mathbf{u}, \mathbf{c}) \leq \mathbf{0}\}$, Aswani et al. (2018) proposed a bi-level inverse optimization model that minimized the distance between the data set $\mathcal{D} = \{(\hat{\mathbf{x}}_1, \hat{\mathbf{u}}_1), \ldots, (\hat{\mathbf{x}}_Q, \hat{\mathbf{u}}_Q)\}$ of points sampled i.i.d. from a distribution $\mathbb{P}_{\mathbf{x},\mathbf{u}}$ and the optimal solution sets. This inverse risk minimization problem is

$$\begin{array}{ll} \underset{\mathbf{c}, \boldsymbol{\epsilon}_1, \ldots, \boldsymbol{\epsilon}_Q}{\text{minimize}} & \displaystyle\sum_{q=1}^{Q} \left\|\boldsymbol{\epsilon}_q\right\|_p \\ \text{subject to} & \hat{\mathbf{x}}_q - \boldsymbol{\epsilon}_q \in \underset{\mathbf{x}}{\arg\min} \left\{f(\mathbf{x}; \hat{\mathbf{u}}_q, \mathbf{c}) \mid g(\mathbf{x}; \hat{\mathbf{u}}_q, \mathbf{c}) \leq \mathbf{0}\right\}, \quad \forall q \in \mathcal{Q} \\ & \mathbf{c} \in \mathcal{C}. \end{array} \tag{A.3}$$

Setting $f(\mathbf{x};\mathbf{u},\mathbf{c}) = \mathbf{c}^\mathsf{T}\mathbf{x}$, $g(\mathbf{x};\mathbf{u},\mathbf{c}) = \mathbf{b} - \mathbf{A}\mathbf{x}$, and $\mathcal{C} = \{\mathbf{c} \in \mathbb{R}^n \mid \|\mathbf{c}\|_N = 1\}$ specializes formulation (A.3) to an equivalent form as $\mathbf{GIO}_p(\mathcal{D})$.

Formulation (A.2) satisfies statistical consistency (i.e., given sufficient points, the imputed $\mathbf{c}$ converges to a true data-generating $\mathbf{c}$) under several assumptions on the data set and the forward model (Aswani et al., 2018):

1. **Assumption 2:** The parameter space $\mathcal{C}$ is convex.
2. **Regularity 1:** The feasible set $\mathcal{P}$ is closed and bounded.
3. **Identifiability condition:** There exists a unique $\mathbf{c}^*$ such that:
   (a) The data set corresponds to noisy perturbations of optimal solutions, i.e., $\hat{\mathbf{x}}_q = \mathbf{x}_q^* + \mathbf{w}_q$, where $\mathbf{x}_q^* \in \arg\min_{\mathbf{x}} \{f(\mathbf{x};\mathbf{u},\mathbf{c}) \mid g(\mathbf{x};\mathbf{u},\mathbf{c}) \leq \mathbf{0}\}$, and $\mathbf{w}_q$ is a random variable with mean 0 and finite variance.
   (b) For any $\mathbf{c} \neq \mathbf{c}^*$, there exists $\mathcal{U}_\mathbf{c}$ such that the marginal distribution $\mathbb{P}_\mathbf{u}(\mathbf{u} \in \mathcal{U}_\mathbf{c}) > 0$ and the optimal value for
   $$\begin{aligned} \inf_{\mathbf{x},\mathbf{x}^*} \quad & \|\mathbf{x} - \mathbf{x}^*\| \\ \text{s.t.} \quad & \mathbf{x} \in \arg\min_{\mathbf{w}} \{f(\mathbf{w};\mathbf{u},\mathbf{c}) \mid g(\mathbf{w};\mathbf{u},\mathbf{c}) \leq \mathbf{0}\} \\ & \mathbf{x}^* \in \arg\min_{\mathbf{w}} \{f(\mathbf{w};\mathbf{u},\mathbf{c}^*) \mid g(\mathbf{w};\mathbf{u},\mathbf{c}^*) \leq \mathbf{0}\} \end{aligned}$$
   is equal to 0 for all $\mathbf{u} \in \mathcal{U}_\mathbf{c}$.
   (c) For all $\mathbf{c}$,
   $$\mathbb{P}_\mathbf{u}\left(\left\{\mathbf{u} \;\middle|\; \left|\arg\min_{\mathbf{x}} \{f(\mathbf{x};\mathbf{u},\mathbf{c}) \mid g(\mathbf{x};\mathbf{u},\mathbf{c}) \leq \mathbf{0}\}\right| > 1\right\}\right) = 0$$

These assumptions do not hold in this work where we focus on a fixed linear forward problem for all data points. Particularly, setting $f(\mathbf{x};\mathbf{u},\mathbf{c}) = \mathbf{c}^\mathsf{T}\mathbf{x}$ and $g(\mathbf{x};\mathbf{u},\mathbf{c}) = \mathbf{b} - \mathbf{A}\mathbf{x}$ implies that the forward and inverse optimization problem do not depend on $\mathbf{u}$. Consequently, the second Identifiability condition does not hold in many settings. A trivial example is $\mathcal{P} = \{(x_1, x_2) \mid 0 \leq x_1, x_2 \leq 1\}$. Here, for any cost vector $\mathbf{c}^*$, there exists another cost vector $\mathbf{c}_i = \mathbf{a}_i / \|\mathbf{a}_i\|_N$ such that the facet described by $\mathbf{c}_i$ contains an optimal vertex of $\mathbf{FO}(\mathbf{c}^*)$. Furthermore, the third condition is also trivially violated when $\mathbf{c} = \mathbf{a}_i$ for any $i \in \mathcal{I}$. Finally, our application in Section 3.4 is an example where the dataset does not correspond to noisy perturbations, but is obtained via several prediction models; we therefore cannot guarantee a well-behaved $\mathbf{w}_q$. We also remark

that our problem setting permits the feasible set $\mathcal{P}$ to be unbounded. A last consequence of $\mathbf{u}$ not existing in our setting is that the parameter space becomes non-convex due to the norm constraint. Overall, we find our problem setting to be incompatible with the statistical consistency guarantees in Aswani et al. (2018).

Aswani et al. (2018) propose an efficient semi-parametric algorithm to solve formulation (A.3) under the assumption that the forward problem is strictly convex in $\mathbf{x}$. For when $f(\mathbf{x};\mathbf{u},\mathbf{c})$ is linear however, Aswani et al. (2018) introduce an enumerative algorithm for solving formulation (A.3) that relies on quantizing the set $\mathcal{C}$ to a finite set $\hat{\mathcal{C}}$, and solving the corresponding formulation with fixed $\mathbf{c} \in \hat{\mathcal{C}}$. This algorithm is effective primarily because, for fixed $\mathbf{c}$, formulation (A.3) (and incidentally, $\mathbf{GIO}_p(\mathcal{D})$) are convex. However, the authors state that due to the enumerative nature, the algorithm is generally only applicable when the parameter space is modest (e.g., $n \leq 5$ is recommended). We find that the algorithm of Aswani et al. (2018) is complementary to ours. That is, their algorithm is inefficient for large n, while ours is relatively insensitive to the increase in n, but is inefficient for large m.

A.2.3 Distributionally robust inverse optimization

Esfahani et al. (2018) study distributionally robust generalized inverse optimization for convex forward problems. Let $\varrho(\cdot)$ denote a risk measure such as the Value-at-Risk (VaR) or Conditional Value-at-Risk (CVaR). The *non-robust* version of their formulation is

$$\begin{array}{ll} \underset{\mathbf{c},\boldsymbol{\epsilon}_1,\ldots,\boldsymbol{\epsilon}_Q}{\text{minimize}} & \varrho(\|\boldsymbol{\epsilon}_1\|,\ldots,\|\boldsymbol{\epsilon}_Q\|) \\ \text{subject to} & \text{Constraints in (A.2) or (A.3)} \end{array} \tag{A.4}$$

Esfahani et al. (2018) consider several different variants of inverse convex optimization to encapsulate previous methods; the variants are referred to as predictability (i.e., inverse risk minimization), sub-optimality, first-order (i.e., inverse variational inequality), and bounded rationality. When the forward problem is a linear program, the sub-optimality loss model is in fact equivalent to the first-order loss model, and therefore also equivalent to $\mathbf{GIO}_{\mathrm{A}}(\mathcal{D})$ proposed here.

A consequence of the general formulation (A.4) is that it leads to a new dominance relationship to bound the optimal values between predictability and sub-optimality losses. Similar to Bertsimas et al. (2015), Esfahani et al. (2018) define the parameter $\gamma \geq 0$ to be the largest parameter satisfying

$$f(\mathbf{x};\mathbf{u},\mathbf{c}) - f(\mathbf{y};\mathbf{u},\mathbf{c}) \geq \nabla f(\mathbf{x};\mathbf{u},\mathbf{c})^{\mathsf{T}}(\mathbf{x}-\mathbf{y}) + \frac{\gamma}{2}\|\mathbf{x}-\mathbf{y}\|_2^2, \quad \forall \mathbf{x},\mathbf{y} \in \mathcal{P}, \mathbf{u} \in \mathcal{U}.$$

Under this definition, Esfahani et al. (2018) show that their sub-optimality (i.e., objective space) loss upper bounds their predictability (i.e., decision space) loss by a multiplicative factor $\gamma/2$. However, similar to the scenario in the previous bound, $\gamma = 0$ when $f(\mathbf{x}; \mathbf{u}, \mathbf{c}) = \mathbf{c}^\mathsf{T}\mathbf{x}$. Consequently, this bound also does not hold for LP forward problems.

Esfahani et al. (2018) focus on solving a distributionally robust version of formulation (A.4), where the robustness is over the worst-case distribution of data. As they primarily address the sub-optimality loss model, which specializes to the absolute duality gap model in this work, the comparison between their solution methods and ours is similar to the comparison between Bertsimas et al. (2015) and ours. That is, we focus on developing efficient algorithms based on linear programming geometry, and as a consequence, yield several new efficiencies in the absolute duality gap setting.

A.3 Automated radiation therapy treatment planning

The design of an IMRT treatment plan is typically done by mathematical optimization where the decision variable $\mathbf{x} = (\mathbf{w}, \mathbf{d})$ is composed of two components that represent the beamlets and the dose delivered (in Gy) as a result of the intensities of the beamlets, respectively. In this section, we detail the forward and inverse models used in automated KBP. Specifically, we highlight the standard formulation used for experiments in Chapter 3. Experiments in Section 4 use a slightly modified version of the forward and inverse problems. We detail those in Appendix B.3.

The forward model for the experiments in Section 3.4 is a modified version of the one used by Babier et al. (2018b). Let $\mathcal{B}$ denote the index set of beamlets and w_b be the radiation intensity of beamlet $b \in \mathcal{B}$. Similarly, let $\mathcal{V}$ denote the index set of voxels within a patient and d_v be the dose of radiation delivered to voxel $v \in \mathcal{V}$. Dose is calculated via a weighted linear combination of all beamlet intensities, i.e., $d_v = \sum_{b \in \mathcal{B}} D_{v,b} w_b$, where $D_{v,b}$ is the dose influence of beamlet b on voxel v.

For each patient, let $\mathcal{T}$ denote the index set of the three planning target volumes (PTVs) with different prescription doses (i.e., PTV56, PTV63, and PTV70 with 56 Gy, 63 Gy, and 70Gy as prescription doses, respectively) and let $\mathcal{O}$ denote the index set of the eight surrounding OARs (i.e., brain stem, spinal cord, right parotid, left parotid, larynx, esophagus, mandible, and limPostNeck). Note that the limPostNeck is an artificially defined region used solely in optimization; it does not possess a clinical criteria. For each $t \in \mathcal{T}$ and $o \in \mathcal{O}$, let $\mathcal{V}_t$ and $\mathcal{V}_o$ denote the set of voxels corresponding to the given target

or OARs, respectively.

A.3.1 Forward objectives

The IMRT forward problem includes 65 different objectives each minimizing some feature of the dose delivered to an OAR or PTV. For each OAR, we minimize the mean dose delivered, the maximum dose delivered, and the average dose above a threshold ϕ_o^θ. Here, ϕ_o^θ is a fraction θ of the average maximum dose to OAR o over the data set of predictions; we consider $\theta \in \Theta := \{0.25, 0.5, 0.75, 0.9, 0.975\}$. Such objectives for each OAR can be computed as follows:

$$z_o^{\text{mean}} = \frac{1}{|\mathcal{V}_o|} \sum_{v \in \mathcal{V}_o} d_v, \quad \forall o \in \mathcal{O} \tag{A.5}$$

$$z_o^{\max} = \max_{v \in \mathcal{V}_o} \{d_v\}, \quad \forall o \in \mathcal{O} \tag{A.6}$$

$$z_o^{\text{thresh},\theta} = \frac{1}{|\mathcal{V}_o|} \sum_{v \in \mathcal{V}_o} \max\left\{0, d_v - \phi_o^\theta\right\}, \quad \forall \theta \in \Theta, \forall o \in \mathcal{O}. \tag{A.7}$$

Each PTV is assigned a prescribed dose ϕ_t, i.e., 56 Gy for PTV56, 63 Gy for PTV63, and 70 Gy for PTV70. For each PTV, we minimize the dose over the prescription, under the prescription, and the maximum dose delivered to the target, which can be computed as follows:

$$z_t^{\text{over}} = \frac{1}{|\mathcal{V}_t|} \sum_{v \in \mathcal{V}_t} \max\{0, d_v - \phi_t\}, \quad \forall t \in \mathcal{T} \tag{A.8}$$

$$z_t^{\text{under}} = \frac{1}{|\mathcal{V}_t|} \sum_{v \in \mathcal{V}_t} \max\{0, \phi_t - d_v\}, \quad \forall t \in \mathcal{T} \tag{A.9}$$

$$z_t^{\max} = \max_{v \in \mathcal{V}_t} \{d_v\}, \quad \forall t \in \mathcal{T}. \tag{A.10}$$

A.3.2 Forward constraints

In order to ensure that no OAR or PTV is prioritized by the objectives at a cost to the other organs, we assign a set of hard constraints for each structure. Every OAR is assigned a constraint to ensure that the mean dose and maximum dose do not exceed baseline safety limits. Similarly, every PTV is assigned a constraint to ensure that it receives a baseline dose on average.

The safety constraints are relaxations of the clinical criteria used to evaluate plans. Recall that clinical plans typically do not satisfy all of the clinical criteria. In fact, satisfying all of the criteria is infeasible for most patients. Consequently, we set these

safety constraints so that all plans can satisfy at least these baseline doses for each of the OARs and PTVs; we then use the objectives to push the doses to achieving the clinical criteria. The baseline values (i.e., right-hand-side), obtained from the average and maximum dose delivered by the 130 clinical plans in our training set, are:

$$\text{Brain stem:} \quad z_o^{\text{mean}} \leq 30, \quad z_o^{\max} \leq 53 \tag{A.11}$$

$$\text{Spinal cord:} \quad z_o^{\text{mean}} \leq 30, \quad z_o^{\max} \leq 46 \tag{A.12}$$

$$\text{Left parotid:} \quad z_o^{\text{mean}} \leq 68, \quad z_o^{\max} \leq 77 \tag{A.13}$$

$$\text{Right parotid:} \quad z_o^{\text{mean}} \leq 68, \quad z_o^{\max} \leq 78 \tag{A.14}$$

$$\text{Larynx:} \quad z_o^{\text{mean}} \leq 68, \quad z_o^{\max} \leq 77 \tag{A.15}$$

$$\text{Esophagus:} \quad z_o^{\text{mean}} \leq 52, \quad z_o^{\max} \leq 75 \tag{A.16}$$

$$\text{Mandible:} \quad z_o^{\text{mean}} \leq 63, \quad z_o^{\max} \leq 76 \tag{A.17}$$

$$\text{limPostNeck:} \quad z_o^{\text{mean}} \leq 21, \quad z_o^{\max} \leq 46 \tag{A.18}$$

$$\text{PTV56:} \quad z_t^{\text{mean}} \geq 58 \tag{A.19}$$

$$\text{PTV63:} \quad z_t^{\text{mean}} \geq 63 \tag{A.20}$$

$$\text{PTV70:} \quad z_t^{\text{mean}} \geq 69 \tag{A.21}$$

Note that we introduce a z_t^{mean} variable for the targets, analogous to z_o^{mean} in (A.5).

Finally, we include a constraint on the "complexity" or physical deliverability of the treatment plan. This constraint, known as the sum-of-positive-gradients (SPG), restricts the variation of radiation doses from neighboring beamlets so that the resulting dose shape is deliverable by the LINAC (Craft et al., 2007). Let $a \in \mathcal{A}$ index each angle of the LINAC, $r \in \mathcal{R}_a$ index each row of the LINAC at that angle, and $\mathcal{B}_r$ be the index set of beamlets along that row. Then, we add the following constraint to restrict the variation of doses to be delivered from different beamlets:

$$\sum_{a \in \mathcal{A}} \max_{r \in \mathcal{R}_a} \left\{ \sum_{b \in \mathcal{B}_r} \max\{0, w_b - w_{b+1}\} \right\} \leq 55, \tag{A.22}$$

where we set $w_{b+1} = 0$ for the last beamlet in each row. The right-hand-side, i.e., the SPG, is set to 55 Gy, following the convention from previous literature (Babier et al., 2020a).

A.3.3 Forward optimization problem

The final forward problem is then to minimize a weighted combination of the objectives:

$$\begin{aligned}\textbf{RT–FO}(\boldsymbol{\alpha}): \ \underset{\mathbf{z},\mathbf{w},\mathbf{d}}{\text{minimize}} \ & \sum_{o\in\mathcal{O}}\left(\alpha_o^{\text{mean}} z_o^{\text{mean}} + \alpha_o^{\max} z_o^{\max} + \sum_{\theta\in\Theta}\alpha_o^{\text{thresh},\theta} z_o^{\text{thresh},\theta}\right) + \\ & \sum_{t\in\mathcal{T}}\left(\alpha_t^{\text{over}} z_t^{\text{over}} + \alpha_t^{\text{under}} z_t^{\text{under}} + \alpha_t^{\max} z_t^{\max}\right) \\ \text{subject to } & (\text{A.5}) - (\text{A.22}) \\ & \sum_{b\in\mathcal{B}} D_{v,b} w_b = d_v, \quad \forall v \in \mathcal{V} \\ & w_b, d_v \geq 0, \quad \forall b \in \mathcal{B}, \forall v \in \mathcal{V}.\end{aligned} \tag{A.23}$$

We compress the notation of the above forward problem to

$$\textbf{RT–FO}(\boldsymbol{\alpha}): \quad \min_{\mathbf{x}} \left\{\boldsymbol{\alpha}^{\mathsf{T}}\mathbf{C}\mathbf{x} \mid \mathbf{A}\mathbf{x} \geq \mathbf{b}, \mathbf{x} \geq \mathbf{0}\right\}.$$

This problem has several useful properties. Firstly under this notation, the matrix of objective functions $\mathbf{C}$ is non-negative. Furthermore, the constraint vector $\mathbf{b}$ is also non-negative. These properties are useful specifically as they allow for constructing almost entirely linear inverse optimization problems. We discuss these in Section A.3.5.

A.3.4 Generating a data set of predicted treatments

We use the training set of 130 patients to implement several machine learning models from the KBP literature. Each model is trained via supervised learning to map from a segmented 3-D CT image (features) to a 3-D dose distribution (target) $\hat{\mathbf{d}}$ using the clinical data set of paired CT images and delivered dose distributions. There are some variations in how each model approaches the task. We use the same features and loss functions for each model as described in their original papers, and summarize the models:

1. **Random Forest:** A random forest that uses 10 hand-crafted geometric features to predict the dose for each voxel $\hat{d}_v$ of the patient sequentially (Mahmood et al., 2018; McIntosh et al., 2017). We run the random forest for all voxels of the patient to complete a dose distribution. The list of features is available in Appendix B.2.

2. **2-D RGB GAN:** A generative adversarial network that uses 2-D axial slices of the patient's CT as an RGB image to predict corresponding 2-D axial slices of the patient's dose also as an RGB image. We convert the images to grayscale and

Table A.1: The percentage of predictions that are feasible with respect to their forward problems.

Predictive model	%-age of feasible predictions
3-D GANCER	95.3
2-D RGB GAN	90.1
2-D GANCER	82.3
2-D RGB GAN-sc.	83.9
RF-sc.	82.3
RF	86.2
2-D GANCER-sc.	87.7
3-D GANCER-sc.	86.9

run 2-D RGB GAN over all 128 axial slices of the patient to produce a 3-D dose distribution. This model is the same as the one introduced in Chapter 4.

3. **2-D GANCER:** A generative adversarial network that uses 2-D axial slices of the patient's CT as an RGB image to predict 2-D axial slices of the patient's dose in grayscale directly (Babier et al., 2020a). We run this model over all 128 axial slices of the patient.

4. **3-D GANCER:** A generative adversarial network that uses the full 3-D patient's CT image as input to predict the full 3-D dose distribution $\hat{\mathbf{d}}$ in one shot (Babier et al., 2020a).

Babier et al. (2020a) noted that plans predicted using the above models often sought to deliver low dose (i.e., significantly spare healthy tissue) at the cost of not satisfying the prescription criteria for the PTVs, and implemented a rescaling method to create a modified prediction to address this issue. In their experiments, they showed that treatment plans constructed using inverse optimization-based KBP and the normalized dose distributions would better satisfy the prescription criteria while performing slightly poorer on sparing healthy tissue. Consequently, we implement the rescaling method on all predictions from the models, and use both the non-scaled and scaled predictions as input for the inverse optimization model. Thus, for each patient there is a data set of 8 dose distributions, i.e., $\mathcal{D} = \{\hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_8\}$. Note that we do not require $\hat{\mathbf{x}}_q = (\hat{\mathbf{w}}_q, \hat{\mathbf{d}}_q)$, but only the objective function values. Inverse optimization then yields a weight vector $\boldsymbol{\alpha}_k$, with which we solve $\mathbf{FO}(\boldsymbol{\alpha}_k)$ to obtain a reconstructed personalized treatment.

Dose predictions may be feasible and sub-optimal or infeasible. Recall from Proposition 6 and 6 that if all decisions in the data set, then solving the ensemble absolute or relative duality gap inverse optimization is equivalent to solving a single-point model

using the centroid. Table A.1 highlights the percentage of the patients for which the predictions are feasible dose distributions with respect to **RT**–**FO**($\boldsymbol{\alpha}$). Typically we observe that about 85% of predictions are feasible, suggesting that there is usually at least one prediction for every patient which is infeasible.

A.3.5 Inverse optimization problems

In order to frame **FO**($\boldsymbol{\alpha}$) for generalized inverse optimization, we restrict imputed cost vectors to be in the image of $\mathbf{C}$, i.e., $\mathcal{C} = \left\{\mathbf{C}^\mathsf{T}\boldsymbol{\alpha} \mid \boldsymbol{\alpha} \geq \mathbf{0}\right\}$. Note that $\boldsymbol{\alpha} \geq \mathbf{0}$ is an application-specific constraint, as there is no clinical interpretation for negative objective function weights.

A specific inverse optimization problem is formulated by appropriately selecting the model hyperparameters $(\|\cdot\|, \mathcal{C}, \mathcal{E}_1, \ldots, \mathcal{E}_Q)$ from **GIO**($\mathcal{D}$). In our experiments, we use the default parameters, except with the custom $\mathcal{C}$ to ensure the objective function is a weighted combination of the different objectives. Moreover, we set $\|\cdot\|_N = \|\cdot\|_1$.

Absolute duality gap

Using Proposition 1 and our specific choice of $\mathcal{C}$, we formulate an absolute duality gap inverse optimization problem:

$$\begin{aligned} \textbf{RT}\text{–}\textbf{IO}_{\mathrm{A}}(\mathcal{D}): \quad & \min_{\boldsymbol{\alpha},\mathbf{y},\epsilon_1,\ldots,\epsilon_Q} \sum_{q=1}^{Q} |\epsilon_q| \\ & \text{s.t.} \quad \mathbf{C}^\mathsf{T}\boldsymbol{\alpha} \geq \mathbf{A}^\mathsf{T}\mathbf{y}, \quad \mathbf{y} \geq \mathbf{0} \\ & \qquad \boldsymbol{\alpha}^\mathsf{T}\hat{\mathbf{z}}_q = \mathbf{b}^\mathsf{T}\mathbf{y} + \epsilon_q, \quad \forall q \in \mathcal{Q} \\ & \qquad (\mathbf{C}^\mathsf{T}\boldsymbol{\alpha})^\mathsf{T}\mathbf{1} = 1 \\ & \qquad \boldsymbol{\alpha} \geq \mathbf{0}. \end{aligned} \tag{A.24}$$

RT–**IO**$_{\mathrm{A}}(\mathcal{D})$ is obtained by substituting $\mathbf{c} = \mathbf{C}^\mathsf{T}\boldsymbol{\alpha}$ into formulation (3.4), and noting that $\left\|\mathbf{C}^\mathsf{T}\boldsymbol{\alpha}\right\|_1 = (\mathbf{C}^\mathsf{T}\boldsymbol{\alpha})^\mathsf{T}\mathbf{1}$ when both $\boldsymbol{\alpha} \geq \mathbf{0}$ and $\mathbf{C} \geq \mathbf{0}$.

Relative duality gap

Using Proposition 4 and our specific choice of $\mathcal{C}$, we formulate a relative duality gap inverse optimization problem. We then use Corollary 5 to obtain the LP relaxation of the relative duality gap problem. The two relevant formulations are given below.

$$\begin{aligned}
&\textbf{RT–IO}_{\mathrm{R}}(\mathcal{D}): \\
\min_{\boldsymbol{\alpha},\mathbf{y},\epsilon_1,\ldots,\epsilon_Q} \quad & \sum_{q=1}^{Q} |\epsilon_q - 1| \\
\text{s.t.} \quad & \mathbf{C}^\mathsf{T}\boldsymbol{\alpha} \geq \mathbf{A}^\mathsf{T}\mathbf{y}, \quad \mathbf{y} \geq \mathbf{0} \\
& \boldsymbol{\alpha}^\mathsf{T}\hat{\mathbf{z}}_q = \epsilon_q \mathbf{b}^\mathsf{T}\mathbf{y}, \quad \forall q \in \mathcal{Q} \\
& (\mathbf{C}^\mathsf{T}\boldsymbol{\alpha})^\mathsf{T}\mathbf{1} = 1 \\
& \boldsymbol{\alpha} \geq \mathbf{0}.
\end{aligned} \tag{A.25}$$

$$\begin{aligned}
&\textbf{RT–IO}_{\mathrm{R,LP}}(\mathcal{D}): \\
\min_{\boldsymbol{\alpha},\mathbf{y},\epsilon_1,\ldots,\epsilon_Q} \quad & \sum_{q=1}^{Q} |\epsilon_q - 1| \\
\text{s.t.} \quad & \mathbf{C}^\mathsf{T}\boldsymbol{\alpha} \geq \mathbf{A}^\mathsf{T}\mathbf{y}, \quad \mathbf{y} \geq \mathbf{0} \\
& \boldsymbol{\alpha}^\mathsf{T}\hat{\mathbf{z}}_q = \epsilon_q, \quad \forall q \in \mathcal{Q} \\
& \mathbf{b}^\mathsf{T}\mathbf{y} = 1 \\
& \boldsymbol{\alpha} \geq \mathbf{0}.
\end{aligned} \tag{A.26}$$

Using Algorithm 3, we first solve the LP relaxation of $\textbf{RT–IO}_{\mathrm{R}}(\mathcal{D})$, stated above as $\textbf{RT–IO}_{\mathrm{R,LP}}(\mathcal{D})$. Note that this relaxation is the application-specific analogue of $\textbf{GIO}^{+}_{\mathrm{R,LP}}(\mathcal{D})$, which is only one of the three reformulations of the relative duality gap problem. We do not construct or solve relaxations of the other two (e.g., $\textbf{GIO}^{-}_{\mathrm{R,LP}}(\mathcal{D})$ and $\textbf{GIO}^{0}_{\mathrm{R,LP}}(\mathcal{D})$) due to the following reasons. First, the analogue to $\textbf{GIO}^{-}_{\mathrm{R,LP}}(\mathcal{D})$ is infeasible; in our application, $\mathbf{b} \geq \mathbf{0}$ implying $\mathbf{b}^\mathsf{T}\mathbf{y} \geq \mathbf{0}$ for all $\mathbf{y} \geq \mathbf{0}$. Second, the application-specific analogue of $\textbf{GIO}^{0}_{\mathrm{R}}(\mathcal{D})$ in practice is often infeasible or generates plans that perform poorly on the clinical criteria satisfaction metrics compared to $\textbf{RT–IO}_{\mathrm{R,LP}}(\mathcal{D})$. Recall that $\textbf{GIO}^{0}_{\mathrm{R}}(\mathcal{D})$ requires $\mathbf{c}^\mathsf{T}\hat{\mathbf{x}}_q = 0$ for all $q \in \mathcal{Q}$. In the application-specific analogue (where the constraint is $\boldsymbol{\alpha}^\mathsf{T}\hat{\mathbf{z}}_q = 0$), both $\boldsymbol{\alpha} \geq 0$ and $\hat{\mathbf{z}}_q \geq 0$, which means that the problem is feasible only when there exists an element of $\hat{\mathbf{z}}_q$ that is equal to 0 for all of the predictions. The only objectives where this situation could occur are the threshold objectives (A.7)–(A.9). Thus, the application-specific analogue of $\textbf{GIO}^{0}_{\mathrm{R}}(\mathcal{D})$ is either infeasible or distributes all of the objective weights to these three objectives. By strictly focusing on the threshold objectives however, the inverse problem then generally fails to meet a large number of the clinical criteria. Consequently, we advocate in this application to strictly use $\textbf{RT–IO}_{\mathrm{R,LP}}(\mathcal{D})$ to solve the relative duality gap inverse optimization problem.

A.3.6 Baseline implementations

In Section 3.4.3, we implement two conventional ensemble learning baselines to compare with ensemble inverse optimization. The first baseline is an ensemble-then-inverse optimization model. Here, we first compute the average of the individual decisions and then solve a single-point inverse optimization problem to obtain a cost vector. The second baseline is a Multiplicative Weights Algorithm (MWA). In our experiments, we implemented both models using all eight predictions as well as for the 4 Pts. predictions (RF-sc., RF, 2-D GANCER-sc., 3-D GANCER-sc.). We also use grid search with the

Algorithm 4 Multiplicative Weights Algorithm Baseline

Input: Data set of CT images for training patients $\mathcal{C}$, Data set of CT images for testing patients $\tilde{\mathcal{C}}$, Dose prediction models $F_1(\cdot), \ldots, F_Q(\cdot)$, Learning rate $\eta \leq 0.5$.

Output: Treatment plans for each patient

Initialize weights $w_q = 1$ for $q \in \mathcal{Q}$.
for Each patient in the training data set $\hat{\mathbf{c}}_k \in \tilde{\mathcal{C}}$ **do**
  for $q \in \mathcal{Q}$ **do**
    Let $\hat{\mathbf{d}}_{q,k} \leftarrow F_q(\hat{\mathbf{c}}_k)$.
    Let $z_{q,k} \leftarrow \mathbf{RT}$–$\mathbf{IO}_{\mathrm{R}}(\{\hat{\mathbf{d}}_{q,k}\})$.
    Let $w_q \leftarrow w_q(1 - \eta z_{q,k})$.
  end for
end for
Normalize weights $w_q \leftarrow w_q / (\sum_{q'=1}^{Q} w_{q'})$.
for Each patient in the testing data set $\hat{\mathbf{c}}_k \in \mathcal{C}$ **do**
  Select prediction model $F_q(\cdot)$ with probability w_q.
  Let $\hat{\mathbf{d}}_{q,k} \leftarrow F_q(\hat{\mathbf{c}}_k)$.
  Let $\boldsymbol{\alpha}_k^*$ be the optimal solution to $\mathbf{RT}$–$\mathbf{IO}_{\mathrm{R}}(\{\hat{\mathbf{d}}_{q,k}\})$.
  Let $\mathbf{x}_k^* \leftarrow \mathbf{RT}$–$\mathbf{FO}(\boldsymbol{\alpha}_k^*)$ and evaluate the corresponding treatment plan.
end for

training set patients to identify the best learning rate for the MWA.

Algorithm 4 summarizes the steps for the MWA. We use an offline learning variant of the Weighted Majority update rule of Arora et al. (2012). Each of the prediction models $F_q(\cdot)$ in the ensemble KBP pipeline is treated as an expert and we initialize a weight $w_q = 1$ for each model. For each of the 130 training set patients k and each prediction model, we predict a dose $\hat{d}_{q,k}$, solve a single-point inverse optimization problem and update the weight w_q by a penalty factor corresponding to the aggregate error of the inverse optimization problem. After repeating this process for all of the training set patients, we normalize the weights to a probability distribution and freeze them. Then for each of the patients in the test set, we randomly select an 'expert' KBP pipeline to generate a treatment plan.

Appendix B

Supplement to Chapter 4

B.1 Network architecture

The general network architecture was adapted from Isola et al. (2017). Contoured CT slices were used as input to the generator as 3-channel, 128×128 images. We used a U-net architecture, where the generator was comprised of an encoder and a decoder stage. We used 4×4 2D convolutions with stride 2 and padding 1. Each convolution layer was followed by a leaky ReLU and batch normalization. Deconvolution layers were followed by 50% dropout, ReLU, and batch normalization.

The encoder consisted of four downsampling layers. The first generated 64 channels, and each subsequent layer downsampled by a factor of 2. This was followed by 2 bottleneck layers, before the data was then passed through 4 upsampling layers. The output of each downsample layer was concatenated to the input of the corresponding upsample layer. The final output was a 3-channel, 128×128 slice. We summarize the generator in Table B.1.

The discriminator consisted of five convolution layers, where the first four each downsample the output by 2. The fifth, and last layer, mapped to a scalar output. We applied batch normalization and leaky ReLU after the first four layers. The final layer was passed through sigmoid activation. We summarize the discriminator in Table B.2

In our experiments, we also compare against a CNN baseline. The architecture for this baseline is identical to the generator for the GAN. The key difference between the CNN and the GAN is that the CNN is trained by minimizing mean-squared error.

Table B.1: Overview of the generator architecture. `BN` refers to batch normalization; `LR`, `R`, and tanh refer to Leaky ReLU (0.2 slope), ReLU, and Tanh activations, respectively; and `D` refers to dropout.

Layer	Concatenate with	Input shape	Block	Activation
1	—	$128 \times 128 \times 128 \times 3$	`conv2d`	`BN-LR`
2	—	$64 \times 64 \times 64 \times 64$	`conv2d`	`BN-LR`
3	—	$32 \times 32 \times 32 \times 128$	`conv2d`	`BN-LR`
4	—	$16 \times 16 \times 16 \times 256$	`conv2d`	`BN-LR`
5	—	$8 \times 8 \times 8 \times 512$	`conv2d`	`BN-LR`
6	—	$4 \times 4 \times 4 \times 512$	`conv2d`	`BN-LR`
7	—	$4 \times 4 \times 4 \times 512$	`conv2d`	`BN-LR`
8	—	$4 \times 4 \times 4 \times 512$	`conv2d`	`BN-LR`
9	—	$2 \times 2 \times 2 \times 512$	`deconv2d`	`LR`
10	layer 5 output	$4 \times 4 \times 4 \times 1024$	`deconv2d`	`BN-R`
11	layer 4 output	$8 \times 8 \times 8 \times 1024$	`deconv2d`	`BN-D-R`
12	layer 3 output	$16 \times 16 \times 16 \times 512$	`deconv2d`	`BN-D-R`
13	layer 2 output	$32 \times 32 \times 32 \times 256$	`deconv2d`	`BN-R`
14	layer 1 output	$64 \times 64 \times 64 \times 128$	`deconv2d`	tanh
Output	—	$128 \times 128 \times 128 \times 3$	—	—

B.2 Random forest model

The random forest used ten custom features outlined in Table B.2 to predict the dose delivered to each voxel in the patient. The RF was trained with ten trees, and default settings with the `randomForestRegressor` from `scikit-learn`.

B.3 Plan optimization model

We follow the same approach as in Chapter 3 to transform dose distributions into treatment plans. First an inverse optimization model takes the 3-D dose distribution as input and estimates the weights for a multi-objective forward optimization problem. We then re-solve the forward optimization problem with the imputed weights to construct a treatment plan.

The forward optimization model in this chapter is a variant of **RT**–**FO**($\boldsymbol{\alpha}$) introduced in Appendix A.3.1. The key difference is that in the previous chapter, we incorporated baseline safety constraints (A.11)–(A.21) in order to ensure that no plans deviate significantly from the desired dose values. The formulation used in this Chapter does not include these safety constraints.

The inverse optimization model used in this chapter is the same as the relative duality

Table B.2: Overview of the discriminator architecture. `BN` refers to batch normalization; `LR`, `R`, and sigmoid refer to Leaky ReLU (0.2 slope), ReLU, and Sigmoid activations.

Layer	Input size	Block	Activation
1	$128 \times 128 \times 128 \times 6$	`conv2d`	`LR`
2	$64 \times 64 \times 64 \times 64$	`conv2d`	`BN-LR`
3	$32 \times 32 \times 64 \times 128$	`conv2d`	`BN-LR`
4	$16 \times 16 \times 16 \times 256$	`conv2d`	`BN-LR`
5	$8 \times 8 \times 8 \times 512$	`conv2d`	sigmoid
Output	1	—	—

Feature	Description
Structure	Structure that the voxel is classified as
y-coordinate	Voxel's positions on the y-axis in a slice
z-coordinate	Plane of voxel's slice
Distance to larynx	Shortest path between voxel and the surface of the larynx
Distance to esophagus	Shortest path between voxel and the surface of the esophagus
Distance to limPostNeck	Shortest path between voxel the surface of the limPostNeck
Distance to PTV56	Shortest path between voxel and the surface of the PTV56
Distance to PTV63	Shortest path between voxel and the the surface of PTV63
Distance to PTV70	Shortest path between voxel and the the surface of PTV70
Influence	Sum of influence matrix elements for the voxel

Table B.3: The ten features used in the RF to predict the dose for any voxel.

gap inverse $\mathbf{RT}$–$\mathbf{IO}_{\mathrm{R}}(\{\hat{\mathbf{x}}\})$ introduced in Appendix A.3.5. Note however that there we used a data set of predictions. In this chapter, we only use a single prediction per inverse optimization model.

Appendix C

Supplement to Chapter 5

C.1 Structural properties of (δ, ϵ)-optimality for the barrier problem

Our learning problem simultaneously trains a classifier and a generative model to learn feasibility and predictive optimal solutions respectively. Alternatively, if we are already given a δ-barrier $B_\delta(\mathbf{x}, \mathbf{u})$, we may consider directly optimizing $\mathbf{BP}(\mathbf{u}, B_\delta, \lambda)$. We show how tuning the λ parameter can yield feasible or infeasible solutions of different qualities.

Under a mild regularity assumption, for a sufficiently large λ, an optimal solution $\mathbf{x}^\lambda(\mathbf{u})$ to $\mathbf{BP}(\mathbf{u}, B_\delta, \lambda)$ is guaranteed to lie inside $\mathcal{X}(\mathbf{u})$. Once λ is sufficiently small, the optimal solutions then enter $\mathcal{N}_\delta(\mathcal{X}(\mathbf{u})) \setminus \mathcal{X}(\mathbf{u})$. We first state this assumption before characterizing the trajectory of the sequence of points obtained via an IPM.

Assumption 3 (Regularity of the δ-barrier)**.**

1. *There exist* $\tilde{\mathbf{x}} \in \text{int}(\mathcal{X}(\mathbf{u}))$ *such that* $B_\delta(\tilde{\mathbf{x}}, \mathbf{u}) > B_\delta(\mathbf{x}, \mathbf{u})$ *for all* $\mathbf{x} \in \text{cl}(\mathcal{N}_\delta(\mathcal{X}(\mathbf{u})) \setminus \mathcal{X}(\mathbf{u}))$.

2. *There exist* $\tilde{\mathbf{x}}' \in \mathcal{N}_\delta(\mathcal{X}(\mathbf{u})) \setminus \mathcal{X}(\mathbf{u})$ *such that* $\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}' < \mathbf{c}^\mathsf{T}\mathbf{x}^*(\mathbf{u})$ *and* $0 < B_\delta(\tilde{\mathbf{x}}', \mathbf{u}) < B_\delta(\mathbf{x}, \mathbf{u})$ *for all* $\mathbf{x} \in \mathcal{X}(\mathbf{u})$.

The first statement implies that there exists a point inside $\mathcal{X}(\mathbf{u})$ for which $B_\delta(\mathbf{x}, \mathbf{u})$ is greater than any point outside of $\mathcal{X}(\mathbf{u})$. Similarly, the second statement implies that there exists a point outside of $\mathcal{X}(\mathbf{u})$ for which $B_\delta(\mathbf{x}, \mathbf{u})$ is lower than any point inside $\mathcal{X}(\mathbf{u})$. Intuitively, the barrier yields higher values for points inside $\mathcal{X}(\mathbf{u})$ rather than outside. Furthermore, the existence of $\tilde{\mathbf{x}}'$ for which $\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}} > \mathbf{c}^\mathsf{T}\mathbf{x}^*(\mathbf{u}) > \mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}'$ is a direct consequence of the linear objective. Figure C.1 shows an example of such points for a

feasible set where the δ-barrier is a canonical barrier for $\mathcal{P}$. Given a barrier function satisfying Assumption 3, λ controls the feasibility of $\mathbf{x}^\lambda(\mathbf{u})$ for $\mathbf{OP}(\mathbf{u})$.

Lemma 3. *If Assumption 3 is satisfied, then there exists $\tilde{\lambda}$ such that for all $\lambda \geq \tilde{\lambda}$, the optimal solution to $\mathbf{BP}(\mathbf{u}, B_\delta, \lambda)$ is feasible for $\mathbf{OP}(\mathbf{u})$, i.e., $\mathbf{x}^\lambda(\mathbf{u}) \in \mathcal{X}(\mathbf{u})$.*

Proof. Let $\mathbf{x}^+ \in \arg\sup_{\mathbf{x}} \{B_\delta(\mathbf{x}, \mathbf{u}) \mid \mathbf{x} \in \mathcal{N}_\delta(\mathcal{X}(\mathbf{u})) \setminus \mathcal{X}(\mathbf{u})\}$ and $\mathbf{x}^- \in \arg\inf_{\mathbf{x}} \left\{\mathbf{c}^\mathsf{T}\mathbf{x} \mid B_\delta(\mathbf{x}, \mathbf{u}) > 0\right\}$. Then, for $\tilde{\mathbf{x}}$ satisfying Assumption 3 Statement 1, we set

$$\tilde{\lambda} = \frac{\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}} - \mathbf{c}^\mathsf{T}\mathbf{x}^-}{\log B_\delta(\tilde{\mathbf{x}}, \mathbf{u}) - \log B_\delta(\mathbf{x}^+, \mathbf{u})}. \tag{C.1}$$

From the optimality of $\mathbf{x}^-$, we have $\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}} > \mathbf{c}^\mathsf{T}\mathbf{x}^-$. Also, Assumption 3 implies that the denominator is positive, and therefore $\tilde{\lambda} > 0$. Rearranging (C.1) yields

$$\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}} - \tilde{\lambda}\log B_\delta(\tilde{\mathbf{x}}, \mathbf{u}) = \mathbf{c}^\mathsf{T}\mathbf{x}^- - \tilde{\lambda}\log B_\delta(\mathbf{x}^+, \mathbf{u}).$$

By optimality of $\mathbf{x}^+$ and $\mathbf{x}^-$, we have $\mathbf{c}^\mathsf{T}\mathbf{x} \geq \mathbf{c}^\mathsf{T}\mathbf{x}^-$ and $\log B_\delta(\mathbf{x}, \mathbf{u}) \leq \log B_\delta(\mathbf{x}^+, \mathbf{u})$ respectively, for all $\mathbf{x} \in \mathcal{N}_\delta(\mathcal{X}(\mathbf{u})) \setminus \mathcal{X}(\mathbf{u})$. Therefore, $\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}} - \tilde{\lambda}\log B_\delta(\tilde{\mathbf{x}}, \mathbf{u}) \leq \mathbf{c}^\mathsf{T}\mathbf{x} - \tilde{\lambda}\log B_\delta(\mathbf{x}, \mathbf{u})$ for all $\mathbf{x} \in \mathcal{N}_\delta(\mathcal{X}(\mathbf{u})) \setminus \mathcal{X}(\mathbf{u})$, concluding that the optimal solution to $\mathbf{BP}(\mathbf{u}, B_\delta, \tilde{\lambda})$ must satisfy $\mathbf{x}^{\tilde{\lambda}}(\mathbf{u}) \in \mathcal{X}(\mathbf{u})$.

Now for any $\varepsilon > 0$, observe that

$$\begin{aligned}\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}} - (\tilde{\lambda}+\varepsilon)\log B_\delta(\tilde{\mathbf{x}}, \mathbf{u}) &\leq \mathbf{c}^\mathsf{T}\mathbf{x} - \tilde{\lambda}\log B_\delta(\mathbf{x}, \mathbf{u}) - \varepsilon\log B_\delta(\tilde{\mathbf{x}}, \mathbf{u}),\ \forall \mathbf{x} \in \mathcal{N}_\delta(\mathcal{X}(\mathbf{u})) \setminus \mathcal{X}(\mathbf{u}) \\ &< \mathbf{c}^\mathsf{T}\mathbf{x} - \tilde{\lambda}\log B_\delta(\mathbf{x}, \mathbf{u}) - \varepsilon\log B_\delta(\mathbf{x}, \mathbf{u}),\ \forall \mathbf{x} \in \mathcal{N}_\delta(\mathcal{X}(\mathbf{u})) \setminus \mathcal{X}(\mathbf{u})\end{aligned}$$

The first line is obtained by adding $\varepsilon\log B_\delta(\tilde{\mathbf{x}}, \mathbf{u})$ to both sides, and the second from $B_\delta(\tilde{\mathbf{x}}, \mathbf{u}) > B_\delta(\mathbf{x}, \mathbf{u})$ for $\mathbf{x} \in \mathcal{N}_\delta(\mathcal{X}(\mathbf{u})) \setminus \mathcal{X}(\mathbf{u})$. Thus, $\mathbf{BP}(\mathbf{u}, B_\delta, \tilde{\lambda}+\varepsilon)$ yields feasible solutions to $\mathbf{OP}(\mathbf{u})$. □

Lemma 4. *If Assumption 3 is satisfied, then there exists $\tilde{\lambda}'$ such that for all $\lambda \leq \tilde{\lambda}'$, the optimal solution to $\mathbf{BP}(\mathbf{u}, B_\delta, \lambda)$ is infeasible for $\mathbf{OP}(\mathbf{u})$, i.e., $\mathbf{x}^\lambda(\mathbf{u}) \in \mathcal{N}_\delta(\mathcal{X}(\mathbf{u})) \setminus \mathcal{X}(\mathbf{u})$.*

Proof. Let $\mathbf{x}^\dagger \in \arg\max_{\mathbf{x}} \{B_\delta(\mathbf{x}, \mathbf{u}) \mid \mathbf{x} \in \mathcal{X}(\mathbf{u})\}$. Then, for $\tilde{\mathbf{x}}'$ satisfying Assumption 3 Statement 2, let

$$\tilde{\lambda}' = \frac{\mathbf{c}^\mathsf{T}\mathbf{x}^*(\mathbf{u}) - \mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}'}{\log B_\delta(\mathbf{x}^\dagger, \mathbf{u}) - \log B_\delta(\tilde{\mathbf{x}}', \mathbf{u})}. \tag{C.2}$$

Assumption 3 Statement 2 ensures $\mathbf{c}^\mathsf{T}\mathbf{x}^*(\mathbf{u}) > \mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}'$ and $\log B_\delta(\mathbf{x}^\dagger, \mathbf{u}) > \log B_\delta(\tilde{\mathbf{x}}', \mathbf{u})$.

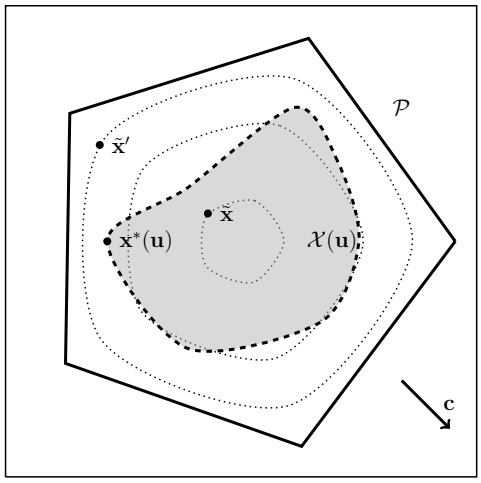


Figure C.1: The canonical barrier $B^{\mathcal{P}}(\mathbf{x})$ where the dotted lines are level sets. $\mathbf{x}^*(\mathbf{u})$ is optimal for $\mathbf{OP}(\mathbf{u})$ while $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{x}}'$ satisfy Lemmas 3 and 4 respectively.

Therefore, $\tilde{\lambda}' > 0$. Rearranging (C.2) gives us

$$\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}' - \tilde{\lambda}' \log B_\delta(\tilde{\mathbf{x}}', \mathbf{u}) = \mathbf{c}^\mathsf{T}\mathbf{x}^*(\mathbf{u}) - \tilde{\lambda}' \log B_\delta(\mathbf{x}^\dagger, \mathbf{u}).$$

By optimality of $\mathbf{x}^*(\mathbf{u})$ and $\mathbf{x}^\dagger$, we have $\mathbf{c}^\mathsf{T}\mathbf{x} \geq \mathbf{c}^\mathsf{T}\mathbf{x}^*(\mathbf{u})$ and $\log B_\delta(\mathbf{x}, \mathbf{u}) \leq \log B_\delta(\mathbf{x}^\dagger, \mathbf{u})$ respectively, for all $\mathbf{x} \in \mathcal{X}(\mathbf{u})$. Therefore $\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}' - \tilde{\lambda}' \log B_\delta(\tilde{\mathbf{x}}', \mathbf{u}) \leq \mathbf{c}^\mathsf{T}\mathbf{x} - \tilde{\lambda}' \log B_\delta(\mathbf{x}, \mathbf{u})$ for all $\mathbf{x} \in \mathcal{X}(\mathbf{u})$, concluding that the optimal solution to $\mathbf{BP}(\mathbf{u}, B_\delta, \tilde{\lambda}')$ must satisfy $\mathbf{x}^{\tilde{\lambda}'}(\mathbf{u}) \in \mathcal{N}_\delta\left(\mathcal{X}(\mathbf{u})\right) \setminus \mathcal{X}(\mathbf{u})$.

Now for any $\varepsilon > 0$, observe that

$$\begin{aligned}\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}' - (\tilde{\lambda}' - \varepsilon) \log B_\delta(\tilde{\mathbf{x}}', \mathbf{u}) &\leq \mathbf{c}^\mathsf{T}\mathbf{x} - \tilde{\lambda}' \log B_\delta(\mathbf{x}, \mathbf{u}) + \varepsilon \log B_\delta(\tilde{\mathbf{x}}', \mathbf{u}), \qquad \forall \mathbf{x} \in \mathcal{X}(\mathbf{u}) \\ &< \mathbf{c}^\mathsf{T}\mathbf{x} - \tilde{\lambda}' \log B_\delta(\mathbf{x}, \mathbf{u}) + \varepsilon \log B_\delta(\mathbf{x}, \mathbf{u}), \qquad \forall \mathbf{x} \in \mathcal{X}(\mathbf{u})\end{aligned}$$

The first line is obtained by subtracting $\varepsilon \log B_\delta(\tilde{\mathbf{x}}', \mathbf{u})$ to both sides, and the second from $B_\delta(\tilde{\mathbf{x}}', \mathbf{u}) < B_\delta(\mathbf{x}, \mathbf{u})$ for all $\mathbf{x} \in \mathcal{X}(\mathbf{u})$. Thus, $\mathbf{BP}(\mathbf{u}, B_\delta, \tilde{\lambda}' - \varepsilon)$ yields infeasible solutions to $\mathbf{OP}(\mathbf{u})$. □

Lemma 4 and Assumption 3 explore the case where the barrier problem produces undesirable results. Otherwise, if $\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}' > \mathbf{c}^\mathsf{T}\mathbf{x}^*(\mathbf{u})$ and $B_\delta(\tilde{\mathbf{x}}', \mathbf{u}) \geq B_\delta(\mathbf{x}, \mathbf{u})$ for all $\tilde{\mathbf{x}}' \in \mathcal{N}_\delta\left(\mathcal{X}(\mathbf{u})\right) \setminus \mathcal{X}(\mathbf{u})$ and $\mathbf{x} \in \mathcal{X}(\mathbf{u})$, $\mathbf{OP}(\mathbf{u})$ could be solved by classical IPMs.

Lemmas 3 and 4 state that when λ is set sufficiently high (or low), the corresponding optimal solution $\mathbf{x}^\lambda(\mathbf{u})$ is a certifiably feasible (or infeasible) solution to $\mathbf{OP}(\mathbf{u})$. Furthermore, there exists a trajectory, i.e., feasibility (or infeasibility) is guaranteed for all λ sufficiently high (or low). Assuming access to an oracle $\Psi(\mathbf{x}, \mathbf{u})$, we con construct a

Algorithm 5 Interior Point Method with a δ-barrier

Input: δ-barrier $B_\delta(\mathbf{x}, \mathbf{u})$; Initial dual variable λ_0 and decay rate $\nu < 1$; Oracle $\Psi(\mathbf{x}, \mathbf{u})$.
Output: Optimal solution $\mathbf{x}^\lambda(\mathbf{u})$ to the barrier problem.

```
for j = 0 to M do
    Solve BP(u, B_δ, λ_j) to obtain optimal solution x^{λ_j}(u).
    if Ψ(x^{λ_j}(u), u) = 0 then
        return Previous optimal solution x^{λ_{j-1}}(u).
    end if
end for
```

simple IPM (see Algorithm 5) to obtain optimal solutions to $\mathbf{OP}(\mathbf{u})$. We initialize with a large λ_0 that satisfies Lemma 3. We define a decay rate $\nu < 1$ and a number of iterations $j \in 0, \ldots, J$. Then, for each j, we simply let $\lambda_j = \lambda_0 \nu^j$ and solve $\mathbf{BP}(\mathbf{u}, B_\delta, \lambda_j)$ to obtain a new (δ, ϵ)-optimal solution in each iteration. At the end of each iteration, the oracle checks if the solution is still feasible, and terminates when the solution exits the feasible set.

Recall that we always have access to the canonical barrier $B^{\mathcal{P}}(\mathbf{x})$ and therefore, we only consider any δ-barrier where $\delta \leq d_H(\mathcal{X}(\mathbf{u}), \mathcal{P})$. We prove the optimality bound for solutions obtained via Algorithm 5.

Proposition 11. *Consider a δ-barrier where $\delta \leq d_H(\mathcal{X}(\mathbf{u}), \mathcal{P})$. Suppose that $\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2 \in \mathcal{X}(\mathbf{u})$ and $\tilde{\mathbf{x}}_1', \tilde{\mathbf{x}}_2' \in \mathcal{N}_\delta(\mathcal{X}(\mathbf{u})) \setminus \mathcal{X}(\mathbf{u})$ satisfy Statements 1 and 2 of Assumption 3, respectively. Assume without loss of generality $B_\delta(\tilde{\mathbf{x}}_1, \mathbf{u}) > B_\delta(\tilde{\mathbf{x}}_2, \mathbf{u})$ and $\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}_1' > \mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}_2'$. Let $\mathbf{x}^{\mathcal{P}} \in \arg\min_{\mathbf{x}} \{f(\mathbf{x}) \mid \mathbf{x} \in \mathcal{P}\}$. For $M > 0$ and $j \in \{0, \ldots, J\}$, consider*

$$\lambda_0 = \frac{\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}_1 - \mathbf{c}^\mathsf{T}\mathbf{x}^{\mathcal{P}}}{\log B_\delta(\tilde{\mathbf{x}}_1, \mathbf{u}) - \log B_\delta(\tilde{\mathbf{x}}_2, \mathbf{u})}, \quad \nu = \left(\frac{\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}_1' - \mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}_2'}{-\lambda_0 \log B_\delta(\tilde{\mathbf{x}}_1', \mathbf{u})}\right)^{1/M}, \quad \lambda_j = \lambda_0 \nu^j$$

Then, the following statements are true:

1. *An optimal solution $\mathbf{x}^{\lambda_0}(\mathbf{u})$ to $\mathbf{BP}(\mathbf{u}, B_\delta, \lambda_0)$ is a feasible solution for $\mathbf{OP}(\mathbf{u})$.*
2. *There exists $1 \leq j^* \leq M$ such that for all $j < j^*$, an optimal solution $\mathbf{x}^{\lambda_j}(\mathbf{u})$ to $\mathbf{BP}(\mathbf{u}, B_\delta, \lambda_j)$ is feasible for $\mathbf{OP}(\mathbf{u})$ and for all $j \geq j^*$, $\mathbf{x}^{\lambda_j}(\mathbf{u})$ is infeasible for $\mathbf{OP}(\mathbf{u})$.*
3. *For any $j < j^*$, an optimal solution $\mathbf{x}^{\lambda_j}(\mathbf{u})$ is $(0, \epsilon_j)$-optimal for $\mathbf{OP}(\mathbf{u})$ where*

$$\epsilon_j = \left(\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}_1' - \mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}_2'\right)\nu^{j-M}.$$

Further, for any $j \geq j^*$, $\mathbf{x}^{\lambda_j}(\mathbf{u})$ is (δ, ϵ_j)-optimal for $\mathbf{OP}(\mathbf{u})$, where $\delta \leq d_H(\mathcal{X}(\mathbf{u}), \mathcal{P})$.

Proof. We first make several observations about the parameters. Note that because $\mathcal{X}(\mathbf{u}) \subset \mathcal{P}$ relaxes the feasible set, we have $\mathbf{c}^\mathsf{T}\mathbf{x}^{\mathcal{P}} \leq \mathbf{c}^\mathsf{T}\mathbf{x}^*(\mathbf{u})$. Next for all $j \leq M$, $\lambda_j = \lambda_0\nu^j$ and specifically $\lambda_M = \lambda_0\nu^M = -(\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}_1' - \mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}_2')/\log B_\delta(\tilde{\mathbf{x}}_1', \mathbf{u})$.

To prove the first statement, we show that $\lambda_0 > \tilde{\lambda}$ where $\tilde{\lambda}$ is defined as in (C.1) and constructed using $\tilde{\mathbf{x}}_1$. Note that $\mathbf{c}^\mathsf{T}\mathbf{x}^{\mathcal{P}} \leq \mathbf{c}^\mathsf{T}\mathbf{x}^-$ and by Assumption 3, $\log B_\delta(\tilde{\mathbf{x}}_2, \mathbf{u}) > \log B_\delta(\mathbf{x}^+, \mathbf{u})$. We substitute $\mathbf{c}^\mathsf{T}\mathbf{x}^{\mathcal{P}}$ and $\log B_\delta(\tilde{\mathbf{x}}_2, \mathbf{u})$ in λ_0 and prove $\lambda_0 > \tilde{\lambda}$. By Lemma 3, Statement 1 must hold.

We use a similar argument using $\tilde{\mathbf{x}}_1'$ to show $\lambda_M < \tilde{\lambda}'$ as defined in (C.2). By Lemma 4, an optimal solution $\mathbf{x}^{\lambda_M}$ must be infeasible for $\mathbf{OP}(\mathbf{u})$. Given that λ_j decreases every iteration and using the first statement, there must exist a cutoff point $1 \leq j^* \leq M$ for which $\lambda_{j^*} < \tilde{\lambda}'$ and $\lambda_{j^*-1} \geq \tilde{\lambda}'$. Therefore, Statement 2 must also hold.

In order to prove the third statement, recall that $\delta \leq d_H(\mathcal{X}(\hat{\mathbf{u}}), \mathcal{P})$ for all j. We first prove $(\Delta(\mathbf{u}), \epsilon_j)$-optimality when $j = M$, and then prove for $j < M$. Let $\epsilon_M = \mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}_1' - \mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}_2'$. Note that

$$\lambda_M = \frac{\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}_1' - \mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}_2'}{-\log B_\delta(\tilde{\mathbf{x}}_1', \mathbf{u})} = \frac{\epsilon_M}{-\log B_\delta(\tilde{\mathbf{x}}_1', \mathbf{u})} < \frac{\epsilon_M}{-\log B_\delta(\mathbf{x}^*, \mathbf{u})}.$$

The second equality follows from substituting the value of ϵ_M and the inequality from $B_\delta(\tilde{\mathbf{x}}_1', \mathbf{u}) < B_\delta(\mathbf{x}^*(\mathbf{u}), \mathbf{u})$ (i.e., Assumption 3). We next show that $\mathbf{x}^{\lambda_M}$ satisfies $(\Delta(\mathbf{u}), \epsilon_M)$-optimality,

$$\begin{aligned}
\mathbf{c}^\mathsf{T}\mathbf{x}^*(\mathbf{u}) + \epsilon_M &> \mathbf{c}^\mathsf{T}\mathbf{x}^*(\mathbf{u}) - \lambda_M \log B_\delta\left(\mathbf{x}^*(\mathbf{u}), \mathbf{u}\right) \\
&\geq \mathbf{c}^\mathsf{T}\mathbf{x}^{\lambda_M}(\mathbf{u}) - \lambda_M \log B_\delta\left(\mathbf{x}^{\lambda_M}(\mathbf{u}), \mathbf{u}\right) \\
&> \mathbf{c}^\mathsf{T}\mathbf{x}^{\lambda_M}(\mathbf{u}).
\end{aligned}$$

The first line follows from substituting the value of ϵ_M and the second from the optimality of $\mathbf{x}^{\lambda_M}(\mathbf{u})$ for $\mathbf{BP}(\mathbf{u}, B_\delta, \lambda_M)$. The third line follows from $-\lambda_M \log B_\delta(\mathbf{x}^{\lambda_M}(\mathbf{u}), \mathbf{u}) > 0$.

For each $j < M$, we have $\lambda_j = \lambda_M\nu^{j-M}$. Then, we write $\epsilon_j = (\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}_1' - \mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}_2')\nu^{j-M}$. The same steps used for the $j = M$ case are repeated to obtain $(\Delta(\mathbf{u}), \epsilon_j)$-optimality certificates. Finally, note that from Statement 2, for all $j < j^*$, the optimal solutions $\mathbf{x}^{\lambda_j}(\mathbf{u})$ are feasible for $\mathbf{OP}(\mathbf{u})$. By optimality of $\mathbf{x}^*(\mathbf{u})$ for $\mathbf{OP}(\mathbf{u})$, we have $\delta = 0$ for all $j < j^*$. □

Proposition 11 first provides parameters $\lambda_0 > \tilde{\lambda}$ and $\lambda_M < \tilde{\lambda}'$ for which the optimal solutions to $\mathbf{BP}(\mathbf{u}, B_\delta, \lambda_0)$ and $\mathbf{BP}(\mathbf{u}, B_\delta, \lambda_M)$ lie inside and outside of $\mathcal{X}(\mathbf{u})$, respectively. Next, it shows that the sequence of λ_j produces a sequence of optimal solutions $\{\mathbf{x}^{\lambda_j}(\mathbf{u})\}$ that start within the feasible set $\mathcal{X}(\mathbf{u})$ and proceed to move outside. Finally, it derives a

sequence of corresponding $\{\epsilon_j\}$ such that the sequence of solutions are $(\Delta(\mathbf{u}), \epsilon_j)$-optimal for $\mathbf{OP}(\mathbf{u})$. This implies the final solution is $(0, (\mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}_1' - \mathbf{c}^\mathsf{T}\tilde{\mathbf{x}}_2')\nu^{j^*-1-M}))$-optimal for $\mathbf{OP}(\mathbf{u})$.

The above proposition summarizes an IPM for solving $\mathbf{OP}(\mathbf{u})$ when given a δ-barrier $B_\delta(\mathbf{x}, \mathbf{u})$ at least as good as the canonical barrier and an oracle $\Psi(\mathbf{x}, \mathbf{u})$. The IPM behaves predictably; by initializing with large λ, we ensure that we obtain feasible solutions, but by decreasing λ, we know that the solution will ultimately be infeasible. An oracle could identify the point of termination immediately before the IPM leaves the feasible set. We can from here obtain a tight bound on the (δ, ϵ)-optimality of the final solution.

While direct optimization is desirable for its structural properties, this IPM approach is reliant on access to a δ-barrier. On the other hand, IPMAN learns a classifier that approximates a δ-barrier after several iterations. Therefore, unless we are given an a priori δ-barrier (e.g., a canonical barrier for $\mathcal{P}$), this IPM approach is not necessarily feasible from the onset. A potential fix would be to first train IPMAN until a δ-barrier is obtained and then use the δ-barrier IPM to solve subsequent problems. This ties to the second difference between the two approaches; IPMAN is ultimately a predictive model and is therefore subject to prediction error. On the other hand, prediction from a trained model is much faster than direct optimization. Therefore, in cases where the problem is large and an IPM would be difficult to solve or require numerous queries from an oracle, the predictive power of IPMAN yields more practical benefits.

C.2 Proof of the generalization bound (Theorem 8)

The proof of the generalization bound uses a Generalization Lemma of Bertsimas and Kallus (2020) to bound the (δ, ϵ)-bound in terms of an empirical risk objective function error between $F^*(\mathbf{u})$ versus $\mathbf{x}^\lambda(\mathbf{u})$ and Markov's inequality to translate this bound to a probabilistic (δ, ϵ)-optimality certificate. However, in order to use the lemma in this way, we first require an auxiliary result to relate F^* with $\mathfrak{R}_{N_\mathbf{u}}(\mathcal{F})$.

Assumption 2 states that the generative model F^* is a composition; we project the classifier output to $\mathcal{P}$ whenever $F^{(j,k)}(\mathbf{u}) \notin \mathcal{P}$. Although $F^{(j,k)} \in \mathcal{F}$, the final model $F^*(\mathbf{u}) := \pi(F(\mathbf{u})) = \arg\min_\mathbf{x}\{\|\mathbf{x} - F(\mathbf{u})\| \mid \mathbf{x} \in \mathcal{P}\}$ may not be a member of $\mathcal{F}$. We first bound the Rademacher complexity of models composed from projection below.

Lemma 5. *Let $\mathcal{F} = \{F : \mathcal{U} \to \mathbb{R}^n\}$ be a model class and $\pi(\mathcal{F}) = \{\pi(F) \mid F \in \mathcal{F}\}$ be the class of models composed by a projection to a polyhedron $\mathcal{P}$. Then for any $\hat{\mathcal{U}} \sim \mathbb{P}_\mathbf{u}$, $\hat{\mathfrak{R}}_{N_\mathbf{u}}(\pi(\mathcal{F}), \hat{\mathcal{U}}) \leq \sqrt{2n}\hat{\mathfrak{R}}_{N_\mathbf{u}}(\mathcal{F}, \hat{\mathcal{U}})$.*

Proof. We want to show for any fixed $\hat{\mathcal{U}}$ that

$$\mathbb{E}_{\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}}\left[\frac{2}{N_{\mathbf{u}}}\sup_{F\in\mathcal{F}}\sum_{i=1}^{N_{\mathbf{u}}}\boldsymbol{\sigma}_i^{\mathsf{T}}\pi\big(F(\hat{\mathbf{u}}_i)\big)\right]\leq\sqrt{2n}\mathbb{E}_{\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}}\left[\frac{2}{N_{\mathbf{u}}}\sup_{F\in\mathcal{F}}\sum_{i=1}^{N_{\mathbf{u}}}\boldsymbol{\sigma}_i^{\mathsf{T}}F(\hat{\mathbf{u}}_i)\right]. \tag{C.3}$$

By conditioning and iterating, it suffices to prove the following inequality for any function $\Xi(F):\mathcal{F}\to\mathbb{R}$,

$$\mathbb{E}_{\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}}\left[\sup_{F\in\mathcal{F}}\boldsymbol{\sigma}^{\mathsf{T}}\pi(F)+\Xi(F)\right]\leq\mathbb{E}_{\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}}\left[\sup_{F\in\mathcal{F}}\sqrt{2n}\boldsymbol{\sigma}^{\mathsf{T}}F+\Xi(F)\right]. \tag{C.4}$$

We first prove inequality (C.4), before returning to the main lemma.

As $\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}$ is a random vector of i.i.d. Rademacher variables, it is supported over the (ordered) set $\{(-1,\dots,-1,-1),(-1,\dots,-1,1),\dots,(1,\dots,1,1)\}$ all with equal probability. Let $\hat{\boldsymbol{\sigma}}_\ell$ denote the ℓ-th element of this set. By iterating over all values, we expand the left-hand-side of (C.4) out to:

$$\mathbb{E}_{\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}}\left[\sup_{F\in\mathcal{F}}\boldsymbol{\sigma}^{\mathsf{T}}\pi(F)+\Xi(F)\right]=\frac{1}{2^n}\sum_{\ell=1}^{2^n}\left(\sup_{F\in\mathcal{F}}\hat{\boldsymbol{\sigma}}_\ell^{\mathsf{T}}\pi(F)+\Xi(F)\right) \tag{C.5}$$

$$=\frac{1}{2^n}\sum_{\ell=1}^{2^{n-1}}\left(\sup_{F\in\mathcal{F}}\left\{\hat{\boldsymbol{\sigma}}_\ell^{\mathsf{T}}\pi(F)+\Xi(F)\right\}+\sup_{F\in\mathcal{F}}\left\{-\hat{\boldsymbol{\sigma}}_\ell^{\mathsf{T}}\pi(F)+\Xi(F)\right\}\right) \tag{C.6}$$

$$=\frac{1}{2^n}\sum_{\ell=1}^{2^{n-1}}\left(\sup_{F_1,F_2\in\mathcal{F}}\hat{\boldsymbol{\sigma}}_\ell^{\mathsf{T}}\left(\pi(F_1)-\pi(F_2)\right)+\Xi(F_1)+\Xi(F_2)\right). \tag{C.7}$$

Equation (C.5) follows by letting $\hat{\boldsymbol{\sigma}}_\ell$ iterate over the support of the distribution. Equation (C.6) follows from the symmetry of the Rademacher distribution. That is, for every $\hat{\boldsymbol{\sigma}}_\ell$, there exists $-\hat{\boldsymbol{\sigma}}_\ell$ with equal probability, and we need to only characterize half of the elements in the support. (C.7) merges the suprema.

By the Obtuse Angle Criterion, projection to a convex set is a non-expansive operation (i.e., $\|\pi(F_1)-\pi(F_2)\|\leq\|F_1-F_2\|$). We use the Cauchy-Schwarz inequality and the non-expansiveness property (in (C.8) and (C.9) below, respectively) to remove the dependency on the projection operator:

$$\text{RHS (C.7)}\leq\frac{1}{2^n}\sum_{\ell=1}^{2^{n-1}}\left(\sup_{F_1,F_2\in\mathcal{F}}\|\hat{\boldsymbol{\sigma}}_\ell\|\,\|\pi(F_1)-\pi(F_2)\|+\Xi(F_1)+\Xi(F_2)\right) \tag{C.8}$$

$$\leq\frac{1}{2^n}\sum_{\ell=1}^{2^{n-1}}\left(\sup_{F_1,F_2\in\mathcal{F}}\|\hat{\boldsymbol{\sigma}}_\ell\|\,\|F_1-F_2\|+\Xi(F_1)+\Xi(F_2)\right) \tag{C.9}$$

$$\leq \frac{1}{2^n}\sum_{\ell=1}^{2^{n-1}}\left(\sup_{F_1,F_2\in\mathcal{F}}\sqrt{n}\,\|F_1-F_2\|+\Xi(F_1)+\Xi(F_2)\right) \tag{C.10}$$

$$\leq \frac{1}{2}\left(\sup_{F_1,F_2\in\mathcal{F}}\sqrt{n}\,\|F_1-F_2\|+\Xi(F_1)+\Xi(F_2)\right) \tag{C.11}$$

$$\leq \frac{1}{2}\left(\sqrt{n}\,\|F_1^*-F_2^*\|+\Xi(F_1^*)+\Xi(F_2^*)\right). \tag{C.12}$$

Inequality (C.10) follows by noting $\|\boldsymbol{\sigma}\|\leq\sqrt{n}$ for all $\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}$ and (C.11) from the fact that the dependency on $\hat{\boldsymbol{\sigma}}_\ell$ has been removed. We obtain (C.12) by letting F_1^* and F_2^* be the two values that attain the supremum.

We use the Khintchine inequality to bound $\|F_1^*-F_2^*\|\leq\sqrt{2}\mathbb{E}_{\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}}\left[\left|\boldsymbol{\sigma}^\mathsf{T}(F_1^*-F_2^*)\right|\right]$. We then rearrange the terms as follows:

$$\text{RHS (C.12)} \leq \frac{1}{2}\left(\sqrt{2n}\mathbb{E}_{\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}}\left[\left|\boldsymbol{\sigma}^\mathsf{T}(F_1^*-F_2^*)\right|\right]+\Xi(F_1^*)+\Xi(F_2^*)\right) \tag{C.13}$$

$$= \frac{1}{2}\left(\mathbb{E}_{\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}}\left[\sqrt{2n}\left|\boldsymbol{\sigma}^\mathsf{T}(F_1^*-F_2^*)\right|+\Xi(F_1^*)+\Xi(F_2^*)\right]\right) \tag{C.14}$$

$$\leq \frac{1}{2}\left(\mathbb{E}_{\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}}\left[\sup_{F_1,F_2\in\mathcal{F}}\sqrt{2n}\left|\boldsymbol{\sigma}^\mathsf{T}(F_1-F_2)\right|+\Xi(F_1)+\Xi(F_2)\right]\right) \tag{C.15}$$

$$= \frac{1}{2}\left(\mathbb{E}_{\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}}\left[\sup_{F\in\mathcal{F}}\left\{\sqrt{2n}\boldsymbol{\sigma}^\mathsf{T}F+\Xi(F)\right\}+\sup_{F\in\mathcal{F}}\left\{-\sqrt{2n}\boldsymbol{\sigma}^\mathsf{T}F+\Xi(F)\right\}\right]\right) \tag{C.16}$$

$$= \mathbb{E}_{\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}}\left[\sup_{F\in\mathcal{F}}\sqrt{2n}\boldsymbol{\sigma}^\mathsf{T}F+\Xi(F)\right]. \tag{C.17}$$

Inequality (C.14) brings all of the terms inside the expectation. (C.15) upper bounds by the supremum. Because $\Xi(F_1)+\Xi(F_2)$ is invariant under the exchange of F_1 and F_2, the supremum will be obtained when $\boldsymbol{\sigma}^\mathsf{T}(F_1-F_2)$ is positive, meaning we can remove the absolute value and separate the supremum in (C.16). Finally, the symmetry of the random variable $\boldsymbol{\sigma}$ implies that the two suprema are equal, thereby giving (C.17).

To complete the proof, we use a standard conditioning argument (see Maurer (2016)) to show (C.3) decomposes to (C.4). For any $0\leq m\leq N_{\mathbf{u}}$, we prove the following by induction:

$$\mathbb{E}_{\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}}\left[\sup_{F\in\mathcal{F}}\sum_{i=1}^{N_{\mathbf{u}}}\boldsymbol{\sigma}_i^\mathsf{T}\pi\big(F(\hat{\mathbf{u}}_i)\big)\right]\leq\mathbb{E}_{\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}}\left[\sup_{F\in\mathcal{F}}\sum_{i=1}^{m}\sqrt{2n}\boldsymbol{\sigma}_i^\mathsf{T}F(\hat{\mathbf{u}}_i)+\sum_{i=m+1}^{N_{\mathbf{u}}}\boldsymbol{\sigma}_i^\mathsf{T}\pi\big(F(\hat{\mathbf{u}}_i)\big)\right].$$

The case for $m = 0$ is an identity. Now for fixed values of $\hat{\boldsymbol{\sigma}}_i, \forall i \neq m$, let

$$\Xi(F) = \sum_{i=1}^{m-1} \sqrt{2n}\hat{\boldsymbol{\sigma}}_i^\mathsf{T} F(\hat{\mathbf{u}}_i) + \sum_{i=m+1}^{N_\mathbf{u}} \hat{\boldsymbol{\sigma}}_i^\mathsf{T} \pi\big(F(\hat{\mathbf{u}}_i)\big).$$

Then, assuming the inequality holds for $m-1$, we show

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}}\left[\sup_{F\in\mathcal{F}} \sum_{i=1}^{N_\mathbf{u}} \boldsymbol{\sigma}_i^\mathsf{T} \pi\big(F(\hat{\mathbf{u}}_i)\big)\right] &\leq \mathbb{E}_{\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}}\left[\sup_{F\in\mathcal{F}} \sum_{i=1}^{m-1} \sqrt{2n}\boldsymbol{\sigma}_i^\mathsf{T} F(\hat{\mathbf{u}}_i) + \sum_{i=m}^{N_\mathbf{u}} \boldsymbol{\sigma}_i^\mathsf{T} \pi\big(F(\hat{\mathbf{u}}_i)\big)\right] \\
&= \mathbb{E}_{\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}}\left[\mathbb{E}_{\boldsymbol{\sigma}_m\sim p_{\boldsymbol{\sigma}}}\left[\sup_{F\in\mathcal{F}} \boldsymbol{\sigma}_m^\mathsf{T} \pi\big(F(\hat{\mathbf{u}}_m)\big) + \Xi(F) \;\middle|\; \{\hat{\boldsymbol{\sigma}}_i, \forall i \neq m\}\right]\right] \\
&\leq \mathbb{E}_{\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}}\left[\mathbb{E}_{\boldsymbol{\sigma}_m\sim p_{\boldsymbol{\sigma}}}\left[\sup_{F\in\mathcal{F}} \sqrt{2n}\boldsymbol{\sigma}_m^\mathsf{T} F(\hat{\mathbf{u}}_m) + \Xi(F) \;\middle|\; \{\hat{\boldsymbol{\sigma}}_i, \forall i \neq m\}\right]\right] \\
&= \mathbb{E}_{\boldsymbol{\sigma}\sim p_{\boldsymbol{\sigma}}}\left[\sup_{F\in\mathcal{F}} \sum_{i=1}^{m} \sqrt{2n}\boldsymbol{\sigma}_i^\mathsf{T} F(\hat{\mathbf{u}}_i) + \sum_{i=m+1}^{N_\mathbf{u}} \boldsymbol{\sigma}_i^\mathsf{T} \pi\big(F(\hat{\mathbf{u}}_i)\big)\right].
\end{aligned}$$

The second inequality comes from substituting (C.4). When $m = N_\mathbf{u}$, the proof is complete. □

Lemma 5 can be seen as an extension of the main theorem of Maurer (2016) and is proved using a similar sequence of steps. There, the authors showed that composition of a Lipschitz scalar-valued vector function onto a vector-valued model class bounds the Rademacher complexity of the composed class by $\sqrt{2L}$. In the above, we compose the projection operator, a vector-valued function, to the vector-valued model class and bound the Rademacher complexity by $\sqrt{2n}$. Although we only specifically consider the projection operator, the proof easily extends to any vector-valued function, so long as it is L-Lipschitz, whereupon we would reintroduce L back into the bound.

Before proving Theorem 8, we re-state the Generalization Lemma of Bertsimas and Kallus (2020).

Lemma 6 (Bertsimas and Kallus (2020))**.** *Consider a function $z(\mathbf{x}, \mathbf{u}) : \mathcal{P} \times \mathcal{U} \to \mathbb{R}$ that is bounded and L_∞-Lipschitz continuous in $\mathbf{x}$ using the $\|\cdot\|_\infty$ norm,*

$$\sup_{\mathbf{x}\in\mathcal{P},\mathbf{u}\in\mathcal{U}} z(\mathbf{x}, \mathbf{u}) \leq K, \qquad \sup_{\mathbf{x}_1\neq\mathbf{x}_2\in\mathcal{P},\mathbf{u}\in\mathcal{U}} \frac{z(\mathbf{x}_1, \mathbf{u}) - z(\mathbf{x}_2, \mathbf{u})}{\|\mathbf{x}_1 - \mathbf{x}_2\|_\infty} \leq L_\infty.$$

For any $\beta > 0$, with probability at least $1 - \beta$ with respect to the sampling of $\hat{\mathcal{U}}$,

$$\mathbb{E}_{\mathbf{u}\sim\mathbb{P}_{\mathbf{u}}}\Big[z\big(F(\mathbf{u}),\mathbf{u}\big)\Big] \leq \frac{1}{N_{\mathbf{u}}}\sum_{i=1}^{N_{\mathbf{u}}} z\big(F(\hat{\mathbf{u}}_i),\hat{\mathbf{u}}_i\big) + K\sqrt{\frac{\log(1/\beta)}{2N_{\mathbf{u}}}} + L_\infty \mathfrak{R}_{N_{\mathbf{u}}}\big(\pi(\mathcal{F})\big), \quad \forall F \in \pi(\mathcal{F}).$$

We are now ready to prove Theorem 8.

Proof. The proof follows by first applying Lemma 6, before applying Markov's inequality. We let $z(\mathbf{x},\mathbf{u}) = |\mathbf{c}^\mathsf{T}\mathbf{x}) - \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\mathbf{u})|$, as a function of $\mathbf{x} \in \mathcal{P}$ and $\mathbf{u} \in \mathcal{U}$, and show it is bounded from above

$$\sup_{\mathbf{x}\in\mathcal{P},\mathbf{u}\in\mathcal{U}} z(\mathbf{x},\mathbf{u}) = \sup_{\mathbf{x}\in\mathcal{P},\mathbf{u}\in\mathcal{U}} \big|\mathbf{c}^\mathsf{T}\mathbf{x} - \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\mathbf{u}))\big| \tag{C.18}$$

$$\leq \max_{\mathbf{x}\in\mathcal{P}} \mathbf{c}^\mathsf{T}\mathbf{x} - \min_{\mathbf{x}\in\mathcal{P}} \mathbf{c}^\mathsf{T}\mathbf{x} = K. \tag{C.19}$$

Because $\mathcal{P}$ is compact, (C.19) is bounded. We set K to be equal to RHS (C.19).

We next show L_∞-Lipschitz continuity,

$$\sup_{\mathbf{x}_1\neq\mathbf{x}_2\in\mathcal{P},\mathbf{u}\in\mathcal{U}} \frac{z(\mathbf{x}_1,\mathbf{u}) - z(\mathbf{x}_2,\mathbf{u})}{\|\mathbf{x}_1 - \mathbf{x}_2\|_\infty} = \sup_{\mathbf{x}_1\neq\mathbf{x}_2\in\mathcal{P},\mathbf{u}\in\mathcal{U}} \frac{\big|\mathbf{c}^\mathsf{T}\mathbf{x}_1 - \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\mathbf{u})\big| - \big|\mathbf{c}^\mathsf{T}\mathbf{x}_2 - \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\mathbf{u})\big|}{\|\mathbf{x}_1 - \mathbf{x}_2\|_\infty} \tag{C.20}$$

$$\leq \sup_{\mathbf{x}_1\neq\mathbf{x}_2\in\mathcal{P},\mathbf{u}\in\mathcal{U}} \frac{\big|\mathbf{c}^\mathsf{T}\mathbf{x}_1 - \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\mathbf{u}) - \mathbf{c}^\mathsf{T}\mathbf{x}_2 + \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\mathbf{u})\big|}{\|\mathbf{x}_1 - \mathbf{x}_2\|_\infty} \tag{C.21}$$

$$= \sup_{\mathbf{x}_1\neq\mathbf{x}_2\in\mathcal{P}} \frac{|\mathbf{c}^\mathsf{T}\mathbf{x}_1 - \mathbf{c}^\mathsf{T}\mathbf{x}_2|}{\|\mathbf{x}_1 - \mathbf{x}_2\|_\infty} = L_\infty \tag{C.22}$$

Inequality (C.21) follows from the Reverse Triangle Inequality. (C.22) follows from the fact that $f(\mathbf{x})$ is linear and therefore, Lipschitz continuous using the $\|\cdot\|_\infty$ norm. We let L_∞ be the Lipschitz constant of $f(\mathbf{x})$.

Because $z(\mathbf{x},\mathbf{u})$ satisfies the bounded and Lipschitz continuity assumptions, we apply Lemma 6 to obtain

$$\mathbb{E}_{\mathbf{u}\sim\mathbb{P}_{\mathbf{u}}}\Big[z\big(F(\mathbf{u}),\mathbf{u}\big)\Big] \leq \frac{1}{N_{\mathbf{u}}}\sum_{i=1}^{N_{\mathbf{u}}} z\big(F(\hat{\mathbf{u}}_i),\hat{\mathbf{u}}_i\big) + K\sqrt{\frac{\log(1/\beta)}{2N_{\mathbf{u}}}} + L_\infty \mathfrak{R}_{N_{\mathbf{u}}}\big(\pi(\mathcal{F})\big), \quad \forall F \in \pi(\mathcal{F}).$$

Specifically, this bound holds for $F^* \in \pi(\mathcal{F})$. By Lemma 5, we can bound $\mathfrak{R}_{N_{\mathbf{u}}}(\pi(\mathcal{F})) \leq \sqrt{2n}\mathfrak{R}_{N_{\mathbf{u}}}(\mathcal{F})$.

The remainder of the proof follows from Markov's inequality. For $\gamma > 0$,

$$\mathbb{P}_{\mathbf{u}}\Big\{ z\big(F^*(\mathbf{u}), \mathbf{u}\big) > \gamma \Big\} = \mathbb{P}_{\mathbf{u}}\Big\{ \big|\mathbf{c}^\mathsf{T} F^*(\mathbf{u}) - \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\mathbf{u})\big| > \gamma \Big\} \leq \frac{\mathbb{E}_{\mathbf{u}\sim\mathbb{P}_{\mathbf{u}}}\Big[\big|\mathbf{c}^\mathsf{T} F^*(\hat{\mathbf{u}}_i) - \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\hat{\mathbf{u}}_i)\big| \Big]}{\gamma}.$$

From the Law of Total Probability, we obtain

$$\begin{aligned}
\mathbb{P}_{\mathbf{u}}\Big\{ \big|\mathbf{c}^\mathsf{T} F^*(\mathbf{u}) - \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\mathbf{u})\big| \leq \gamma \Big\} &= 1 - \mathbb{P}_{\mathbf{u}}\Big\{ \big|\mathbf{c}^\mathsf{T} F^*(\mathbf{u}) - \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\mathbf{u})\big| > \gamma \Big\} \\
&\geq 1 - \frac{\mathbb{E}_{\mathbf{u}\sim\mathbb{P}_{\mathbf{u}}}\Big[\big|\mathbf{c}^\mathsf{T} F^*(\hat{\mathbf{u}}_i) - \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\hat{\mathbf{u}}_i)\big| \Big]}{\gamma}, \\
&\geq 1 - \frac{\dfrac{1}{N_{\mathbf{u}}}\displaystyle\sum_{i=1}^{N_{\mathbf{u}}} \big|\mathbf{c}^\mathsf{T} F^*(\hat{\mathbf{u}}_i) - \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\hat{\mathbf{u}}_i)\big| + K\sqrt{\dfrac{\log(1/\beta)}{2N_{\mathbf{u}}}} + \sqrt{2n}L_\infty \mathfrak{R}_{N_{\mathbf{u}}}(\mathcal{F})}{\gamma},
\end{aligned}$$

with probability $1 - \beta$. The second and third line follow from Markov's inequality and substituting the bound from Lemma 6, respectively. Given that we have a probabilistic bound for the error of $F^*(\mathbf{u})$ from $\mathbf{x}^\lambda(\mathbf{u})$, we bound the error to $\mathbf{x}^*(\mathbf{u})$. Recall that $\mathbf{x}^\lambda(\mathbf{u})$ is (δ, ϵ)-optimal. There are two cases to consider. First, if $\mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\mathbf{u}) \leq \mathbf{c}^\mathsf{T} F(\mathbf{u})) \leq \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\mathbf{u}) + \gamma$, then by substitution,

$$\mathbf{c}^\mathsf{T} F^*(\mathbf{u}) - \epsilon - \gamma < \mathbf{c}^\mathsf{T}\mathbf{x}^*(\mathbf{u}) < \mathbf{c}^\mathsf{T} F^*(\mathbf{u}) + \delta.$$

Alternatively, if $\mathbf{c}^\mathsf{T} F^*(\mathbf{u}) \leq \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\mathbf{u}) \leq \mathbf{c}^\mathsf{T} F^*(\mathbf{u}) + \gamma$, then by substitution,

$$\mathbf{c}^\mathsf{T} F^*(\mathbf{u}) - \epsilon < \mathbf{c}^\mathsf{T}\mathbf{x}^*(\mathbf{u}) < \mathbf{c}^\mathsf{T} F^*(\mathbf{u}) + \delta + \gamma.$$

Note that both of these events can be covered by adding and subtracting γ to both the upper and lower bounds respectively. Then,

$$\mathbb{P}_{\mathbf{u}}\Big\{ \mathbf{c}^\mathsf{T} F^*(\mathbf{u}) - \epsilon - \gamma < \mathbf{c}^\mathsf{T}\mathbf{x}^*(\mathbf{u}) < \mathbf{c}^\mathsf{T} F^*(\mathbf{u}) + \delta + \gamma \Big\} \geq \mathbb{P}_{\mathbf{u}}\Big\{ \big|\mathbf{c}^\mathsf{T} F^*(\mathbf{u}) - \mathbf{c}^\mathsf{T}\mathbf{x}^\lambda(\mathbf{u})\big| \leq \gamma \Big\},$$

completing the proof. □

C.3 Implementation details for predicting optimal dose distributions

C.3.1 Problem formulation

In our RT experiments, we assume that each patient contains seven organs-at-risk (OARs) (i.e., brainstem, spinal cord, right parotid, left parotid, larynx, esophagus, and mandible) to which we minimize the average dose. Each patient also contains up to three planning target volumes (PTVs) with different prescription doses (i.e., PTV56, PTV63, and PTV70 with 56 Gy, 63 Gy, and 70Gy as prescription doses, respectively). We remark that constraints to the brainstem, spinal cord, and esophagus are generally easily satisfied by all predictions. Consequently, we focus specifically on the right parotid, left parotid, larynx, mandible, PTV56, PTV63, and PTV70.

Each of the OARs and targets require polyhedral upper and lower bound constraints to the mean dose delivered to that structure. Furthermore, there exists a hidden "clinical criteria" constraint for each OAR and target that must be satisfied at the discretion of an oncologist. That is, if the ground truth treatment plan for a given patient from the data set satisfies a hidden constraint, then any generated plan for that patient must also satisfy that constraint. The hidden constraint for each OAR is an upper bound on either the mean or maximum dose delivered to that structure, while the hidden constraint for each target is a lower bound on the value-at-risk, i.e., minimum dose delivered to 90-th percentile of the target structure. The oracle $\Psi(\mathbf{x}, \mathbf{u})$ is a look-up table that compares the dose generated by our model with the ground truth (i.e., what was actually delivered). In particular, for each structure, $\Psi(\mathbf{x}, \mathbf{u})$ checks whether the input dose satisfies all the constraints (i.e., two polyhedral and one hidden constraints).

C.3.2 Neural network architecture

(Babier et al., 2020a) propose a modified version of the generative adversarial network of Chapter 4, extending their model to a 3-D GAN. We use a modified version of the generative adversarial network (GAN) of (Babier et al., 2020a) where we take as input a one-hot encoded CT image and incorporate average pooling layers. The GAN consists of two networks learn to predict dose distributions. The architectures for $F(\mathbf{u})$ and $B(\mathbf{x}, \mathbf{u})$ are described in Tables C.1 and C.2, respectively.

The generator takes as input a tensor $\mathbf{u} \in \mathbb{R}^{128\times128\times128\times8}$, where the first three dimensions correspond to a voxel in the patient's geometry. The fourth dimension is a concatenation of the CT image greyscale and a one-hot encoded vector in $\{0, 1\}^7$ whose

Table C.1: Overview of the generator architecture. `BN` refers to batch normalization; `LR`, `R`, and tanh refer to Leaky ReLU (0.2 slope), ReLU, and Tanh activations, respectively; `AP` refers to a mean pool; and `D` refers to dropout.

Layer	Concatenate with	Input shape	Block	Activation
1	—	$128 \times 128 \times 128 \times 8$	`conv3d`	`BN-LR`
2	—	$64 \times 64 \times 64 \times 64$	`conv3d`	`BN-LR`
3	—	$32 \times 32 \times 32 \times 128$	`conv3d`	`BN-LR`
4	—	$16 \times 16 \times 16 \times 256$	`conv3d`	`BN-LR`
5	—	$8 \times 8 \times 8 \times 512$	`conv3d`	`BN-LR`
6	—	$4 \times 4 \times 4 \times 512$	`conv3d`	`BN-LR`
7	—	$2 \times 2 \times 2 \times 512$	`deconv3d`	`LR`
8	layer 5 output	$4 \times 4 \times 4 \times 1024$	`deconv3d`	`BN-R`
9	layer 4 output	$8 \times 8 \times 8 \times 1024$	`deconv3d`	`BN-D-R`
10	layer 3 output	$16 \times 16 \times 16 \times 512$	`deconv3d`	`BN-D-R`
11	layer 2 output	$32 \times 32 \times 32 \times 256$	`deconv3d`	`BN-R`
12	layer 1 output	$64 \times 64 \times 64 \times 128$	`deconv3d`	`AP`-tanh
Output	—	$128 \times 128 \times 128 \times 1$	—	—

elements label whether the voxel belongs to one of the seven contoured structures. The generator then outputs a tensor $\mathbf{x} \in \mathbb{R}^{128\times128\times128}$ whose elements specify the dose to be delivered to each voxel of the patient.

The classifier is trained to predict whether a given dose distribution satisfies all of the constraints (both hidden and polyhedral) for each structure of the patient. This network takes as input the concatenated tensor $(\mathbf{x}, \mathbf{u})$ and outputs a vector in $[0, 1]^7$, whose elements each indicate the classifier's belief of whether the given dose distribution has satisfied all of the constraints for each specific structure. Consequently, learning feasibility becomes a multi-label classification problem and the classifier acts as seven separate classifiers each predicting feasibility with respect to an individual structure, but whose model parameters are shared with each other. For any structure, in order to classify a dose distribution as satisfying the relevant constraints, the classifier must: (i) first determine from the dose whether the polyhedral constraints are satisfied, (ii) determine from the CT image whether the patient requires a hidden constraint to be satisfied, and (iii) determine from the dose whether the hidden constraint is satisfied if this constraint is required for the patient. Overall, a dose distribution is feasible only if all constraints are satisfied.

C.3.3 Implementation of the IPMAN algorithm

Table C.2: Overview of the classifier architecture. `BN` refers to batch normalization; `LR`, `R`, and sigmoid refer to Leaky ReLU (0.2 slope), ReLU, and Sigmoid activations.

Layer	Input size	Block	Activation
1	$128 \times 128 \times 128 \times 9$	`conv3d`	`LR`
2	$64 \times 64 \times 64 \times 64$	`conv3d`	`BN-LR`
3	$32 \times 32 \times 64 \times 128$	`conv3d`	`BN-LR`
4	$16 \times 16 \times 16 \times 256$	`conv3d`	`BN-LR`
5	$8 \times 8 \times 8 \times 512$	`conv3d`	sigmoid
Output	7	—	—

Algorithm 6 Generator pre-training and data augmentation

Input: Feasible and input data sets $\mathcal{D} = \{(\hat{\mathbf{x}}_i, \hat{\mathbf{u}}_i)\}_{i=1}^{N_{\mathbf{u}}}, \hat{\mathcal{U}} = \{\hat{\mathbf{u}}_i\}_{i=1}^{N_{\mathbf{u}}}$, infeasible data set $\bar{\mathcal{D}} = \emptyset$, Pre-training number of epochs E_{ST}.
Output: Pre-trained generative model $F^{(0,0)}$, Feasible and infeasible data sets $\mathcal{D}, \bar{\mathcal{D}}$.

```
Initialize generator and discriminator F, D
for e = 1 to E_ST do
    Update generator and discriminator F*, D* ← Adam(∇L_ST).
    for all û_i ∈ Û do
        Append D ← D ∪ (F*(û_i), û_i) if Ψ(F*(û_i), û_i) = 1 else D̄ ← D̄ ∪ (F*(û_i), û_i).
    end for
end for
return F^(0,0) ← F*, D, D̄.
```

In this subsection, we describe the exact implementation of the IPMAN algorithm used in our experiments. We summarize the steps in Algorithm 7. As our generative and classification models are neural networks, we remark on several improvements that can be made to the algorithm.

Pre-training as a GAN

Just as classical IPMs require a good initial point (i.e., lying within the feasible set) in order to construct a trajectory of points leading to an optimal solution, IPMAN can be made more efficient by ensuring that the generative model is initialized to predict points that are likely to be feasible. This initialization can greatly improve the training time and stability of the algorithm. Consequently, we first pre-train the generative model and subsequently apply transfer learning at the beginning of the algorithm (Goodfellow et al., 2016).

Pre-training amounts to training the generative model first as a Style Transfer GAN to learn to predict dose distributions from CT images as in Mahmood et al. (2018). The

Algorithm 7 IPMAN for radiation therapy

Input: Data sets of decisions $\mathcal{D}$, $\bar{\mathcal{D}}$, and inputs $\hat{\mathcal{U}} = \{\hat{\mathbf{u}}_i\}_{i=1}^{N_{\mathbf{u}}}$, Set of dual variables $\{\lambda_j\}_{j=0}^{M}$, Number of iterations K, Number of epochs E_B, E_F, Subset sampling rate s
Output: Final generative models $F^{(j,K)}$ for $j \in \{0, \ldots, M\}$

Pre-train generator using Algorithm 6.
Initialize generator $F^{(j,0)} \leftarrow F^*$ for $j \in \{0, \ldots, M\}$, classifier B.
for $k = 1$ **to** K **do**
  Sample subsets to train $\mathcal{D}^{(k)} = \sigma(\mathcal{D}; s)$, $\bar{\mathcal{D}}^{(k)} = \sigma(\bar{\mathcal{D}}; s|\mathcal{D}|/|\bar{\mathcal{D}}|)$.
  for $e = 0$ **to** E_B **do**
    Update classifier $B^{(k)} \leftarrow \text{Adam}(\nabla L_B)$.
  end for
  for $j = 0$ **to** M **do**
    for $e = 0$ **to** E_F **do**
      Update generator $F^{(j,k)} \leftarrow \text{Adam}(\nabla L_F)$.
    end for
    for all $\hat{\mathbf{u}}_i \in \hat{\mathcal{U}}$ **do**
      Append $\mathcal{D} \leftarrow \mathcal{D} \cup (F^{(j,k)}(\hat{\mathbf{u}}_i), \hat{\mathbf{u}}_i)$ if $\Psi(F^{(j,k)}(\hat{\mathbf{u}}_i), \hat{\mathbf{u}}_i) = 1$ else $\bar{\mathcal{D}} \leftarrow \bar{\mathcal{D}} \cup (F^{(j,k)}(\hat{\mathbf{u}}_i), \hat{\mathbf{u}}_i)$.
    end for
  end for
end for
return $F^{(j,K)}$ for $j \in \{0, \ldots, M\}$

steps are summarized in Algorithm 6. In order to pre-train our generative model, we introduce a discriminator network $D(\mathbf{x}, \mathbf{u}) : \mathbb{R}^{128\times128\times128} \times \mathbb{R}^{128\times128\times128\times8} \to \mathbb{R}$. As in a Style Transfer GAN, we train the discriminator using $\mathcal{D}$ to classify whether a given dose and CT image pair belongs to the data distribution and a generative model to predict dose distributions that fool the discriminator. Specifically,

$$\min_F \max_D \left\{ L_{ST} := \frac{1}{N_{\mathbf{u}}} \sum_{(\hat{\mathbf{u}}_i, \hat{\mathbf{x}}_i) \in \mathcal{D}} \log D(\hat{\mathbf{x}}_i, \hat{\mathbf{u}}_i) + \log\left(1 - D(F(\hat{\mathbf{u}}_i), \hat{\mathbf{u}}_i)\right) + \lambda_{ST} \left\| F(\hat{\mathbf{u}}_i) - \hat{\mathbf{x}}_i \right\|_1 \right\}.$$

The l_1-loss is a regularization term that ensures that the generator predicts dose distributions that resemble the ground truth and λ_{ST} is the regularization parameter. GANs are trained by iterative gradient descent between $F(\mathbf{u})$ and $D(\mathbf{x}, \mathbf{u})$. In our implementation, we set $\lambda_{ST} = 90$ and train the GAN for 50 epochs, following the practice from Mahmood et al. (2018). At the end of pre-training, we discard $D(\mathbf{x}, \mathbf{u})$ and let $F^{(0,0)}(\mathbf{u})$ denote the trained generative model.

Sampling an infeasible data set of decisions $\bar{\mathcal{D}}$

Training IPMAN requires an initial data set of infeasible decisions $\bar{\mathcal{D}}$. In practice, a data set of infeasible decisions would not be available a priori and, instead, is generated by sampling. Note, however, that in every epoch of the pre-training step, the generative model generates candidate solutions $F(\hat{\mathbf{u}}_i)$ to attempt to fool the discriminator. We save the generated decisions during pre-training and label them afterwards as feasible or infeasible using the oracle. By training using 100 patients for 50 epochs, we generate a total of 5000 dose distributions that are labelled as feasible or infeasible and then binned in the appropriate $\mathcal{D}$ or $\bar{\mathcal{D}}$, respectively. The steps are summarized in Algorithm 6.

Learning multi-label feasibility with sub-sampled data sets

Training IPMAN for multiple iterations can produce a large quantity of generated data points. Furthermore, because we consider feasibility for each structure separately, training the classifier quickly becomes prohibitively expensive. In order to reduce training time, we do not use the entire data sets $\mathcal{D}$ and $\bar{\mathcal{D}}$ but rather smaller sampled subsets. Let $\sigma(\cdot; s)$ be a random sampling operator (without replacement) where s is the fraction of points to sample. For example, $\sigma(\mathcal{D}; 0.5)$ denotes a randomly sampled subset of size $0.5|\mathcal{D}|$. In our implementation, we set $s = 0.3$ and trained the classifier using $\mathcal{D}^{(k)} = \sigma(\mathcal{D}; 0.3)$ and $\bar{\mathcal{D}}^{(k)} = \sigma(\bar{\mathcal{D}}; 0.3|\mathcal{D}|/|\bar{\mathcal{D}}|)$; this reduced the training time to 24 hours.

We next define the multi-label classification problem. For any $(\hat{\mathbf{u}}_i, \hat{\mathbf{x}}_i)$ in $\mathcal{D}$ or $\bar{\mathcal{D}}$, let $\psi_{i,r}$ be a label determining whether the dose distribution had satisfied the polyhedral and (conditional) hidden constraints for structure r. That is, if the ground truth dose for $\hat{\mathbf{u}}_i$ satisfied the hidden constraints, $\psi_{i,r} = 1$ if the polyhedral and hidden constraints were satisfied and zero otherwise. If the clinical dose did not satisfy the hidden constraint, then $\psi_{i,r} = 1$ if only the polyhedral constraints were satisfied; here, the hidden constraint is inactive for this patient. Then, let $[B(\mathbf{x}, \mathbf{u})]_r$ denote the r-th element of the classifier output. The classifier problem is

$$\max_{B \in \mathcal{B}} \left\{ L_B := \frac{1}{N_{\mathbf{x}} + \bar{N}_{\mathbf{x}}} \sum_{\substack{(\hat{\mathbf{x}}_i, \hat{\mathbf{u}}_i) \in \\ \mathcal{D}^{(k)} \cup \bar{\mathcal{D}}^{(k)}}} \sum_{r \in \mathcal{R}} \left(\psi_{i,r} \log \left[B(\hat{\mathbf{x}}_i, \hat{\mathbf{u}}_i) \right]_r + (1 - \psi_{i,r}) \log \left(1 - \left[B(\hat{\mathbf{x}}_i, \hat{\mathbf{u}}_i) \right]_r \right) \right) \right\}.$$

The above problem specializes to $\mathbf{FCP}(\mathcal{D}^{(k)}, \bar{\mathcal{D}}^{(k)})$ in the single-class setting (i.e., $|\mathcal{R}| = 1$). For a dose distribution to be classified feasible, $B(\mathbf{x}, \mathbf{u})_r = 1$ for all $r \in \mathcal{R}$. This approach of separating the constraint satisfaction along all structures individually is

equivalent to modeling the optimization problem via a barrier function for each structure. Furthermore, the barriers are approximated by a neural network classifier with shared weights except in the last layer. As we describe later below, the objective of the barrier optimization problem is obtained by summing all of the separate barriers, i.e., $f(\mathbf{x}) - \lambda \sum_{r \in \mathcal{R}} \log[B(\mathbf{x}, \mathbf{u})]_r$. Finally, we minimize L_B using the Adam optimizer for $E_B = 10$ epochs in every iteration. Note that it is essential to ensure that the classifier accurately predicts feasibility in order to be able to approximate a δ-barrier.

Regularized barrier optimization problem

We include an l_1 regularization term in training. This term is equivalent to the one used in the pre-training stage and is useful to ensure that predicted dose distributions do not deviate too far from the ground truth. Note that we only use this regularization in the first set of experiments (Section 5.7.3) and remove it in the second set of experiments (Section 5.7.4). There, the ground truth plans may not be feasible, meaning that it would be incorrect to replicate ground truth behavior. However, a consequence of removing a regularization term is that certain models may become unstable and deviate significantly if the classifier is not a complete δ-barrier. We observe this behavior in $\lambda = 4$ where the model minimizes dose while ignoring feasibility.

With slight abuse of notation, let $\mathbf{z}(F(\mathbf{u}))$ denote the vector of average doses to each structure as constructed by the generative model. Then, the generative barrier problem in this setting is

$$\min_{F \in \mathcal{F}} \left\{ L_F := \frac{1}{N_{\mathbf{u}}} \sum_{\hat{\mathbf{u}}_i \in \hat{\mathcal{U}}} \left(\frac{1}{\lambda_j} f\Big(\mathbf{z}\big(F(\hat{\mathbf{u}}_i)\big)\Big) + \lambda_{ST} \left\| F(\hat{\mathbf{u}}_i) - \hat{\mathbf{x}}_i \right\|_1 - \sum_{r \in \mathcal{R}} \Big[B^{(k)}\big(F(\hat{\mathbf{u}}_i), \hat{\mathbf{u}}_i\big) \Big]_r \right) \right\}.$$

We minimize L_F using the Adam optimizer for $E_F = 1$ epoch in every iteration. It is important to ensure that the classifier is trained close to optimality to ensure that it approximates a δ-barrier. Furthermore, given the nature of training the classifier, it is often the case that the classifier's support is uneven and may have areas of local optimality that the generator may abuse. A standard practice in the GAN literature is to control the training duration of the two networks; we employ a similar strategy by training the generative model for a shorter duration than the classifier in order to ensure that the generator does not overfit and abuse local optima caused by the classifier. As previously mentioned, we set $\lambda_{ST} = 50$ for the first set of experiments but require that $\lambda_{ST} = 0$ for the second set.

Appendix D

Supplement to Chapter 6

D.1 Generalizing Complement SB to ellipsoids

The Complement SB algorithm can also be extended to sample over the complement of non-polyhedral sets. Here, we focus specifically on ellipsoidal sets, where we develop an analogous result to Theorem 1, i.e., the Complement SB covers the entire complement of the set. However, note that the extension requires a different proof technique.

Assume that our set is a compact, full-dimensional ellipsoid defined as

$$\mathcal{X} = \left\{ \mathbf{x} \mid (\mathbf{x} - \mathbf{x}_r)^\mathsf{T} \mathbf{P} (\mathbf{x} - \mathbf{x}_r) \leq 1 \right\},$$

where $\mathbf{P} \in \mathbb{S}^{n \times n}$ is a positive semi-definite matrix and $\mathbf{x}_r \in \mathbb{R}^n$ is the centroid. We can rewrite this ellipsoid as $\mathcal{X} = \left\{ \mathbf{x} \mid \frac{1}{2} \mathbf{x}^\mathsf{T} \mathbf{A} \mathbf{x} + \mathbf{b}^\mathsf{T} \mathbf{x} + c \leq 0 \right\}$, where $\mathbf{A} \in \mathbb{S}^{n \times n}$, $\mathbf{b} \in \mathbb{R}^n$, and $c \in \mathbb{R}$ are determined by expanding the quadratic term.

Let $f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\mathsf{T} \mathbf{A} \mathbf{x} + \mathbf{b}^\mathsf{T} \mathbf{x} + c$. For every $\mathbf{w} \in \mathrm{bd}(\mathcal{X})$, the sub-gradient $\nabla f(\mathbf{w}) = \mathbf{A}^\mathsf{T} \mathbf{w} + \mathbf{b}$ defines a supporting hyperplane of the ellipsoid, i.e.,

$$(\mathbf{A}^\mathsf{T} \mathbf{w} + \mathbf{b})^\mathsf{T} \mathbf{w} \geq (\mathbf{A}^\mathsf{T} \mathbf{w} + \mathbf{b})^\mathsf{T} \mathbf{x}, \qquad \forall \mathbf{x} \in \mathcal{X}.$$

From Rockafellar (1970), we may also write $\mathcal{X}$ as an intersection of the tangent half-spaces, i.e., $\mathcal{X} = \{ \mathbf{x} \in \mathbb{R}^n \mid \nabla f(\mathbf{w})^\mathsf{T} \mathbf{x} \geq \nabla f(\mathbf{w})^\mathsf{T} \mathbf{x}, \ \forall \mathbf{w} \in \mathrm{bd}(\mathcal{X}) \}$.

The Complement SB algorithm for ellipsoids operates in the same manner as for polyhedra with the only difference being in how we generate a direction vector for the next boundary point. When $\mathcal{X}$ is a polyhedron, we identify the current facet and select a direction on the interior half-space of that facet. When $\mathcal{X}$ is an ellipsoidal set, we select a direction on the interior tangent half-space. This is summarized in Algorithm 8.

Algorithm 8 Complement Shake-and-Bake for Ellipsoids

Require: Ellipsoidal set $\mathcal{X}$; Sampling distributions $p_{\mathbf{w}}(\mathbf{w}'|\mathbf{w})$, $p_{\mathbf{r}}(\mathbf{r}|\mathbf{w})$, $p_{\xi}(\xi|\mathbf{r},\mathbf{w})$, Number of points N; Initialization $\hat{\mathbf{w}}_1 \in \mathrm{bd}(\mathcal{X}), i = 1$, $\mathcal{D} = \emptyset$

for $i = 1$ **to** N **do**

Randomly sample $\mathbf{r}_i \sim p_{\mathbf{r}}(\mathbf{r}|\mathbf{w}_i)$ and $\xi_i \sim p_{\xi}(\xi|\mathbf{r},\mathbf{w})$.

Update data set $\mathcal{D} \leftarrow \mathcal{D} \cup \{\mathbf{w}_i - \xi_i \mathbf{r}_i\}$.

Let $\theta_i = \max\{t \mid \mathbf{w}_i + t\mathbf{r}_i \in \mathcal{X}\}$.

With probability $p_{\mathbf{w}}(\mathbf{w}_i + \theta_i \mathbf{r}_i|\mathbf{w}_i)$, update $\mathbf{w}_{i+1} \leftarrow \mathbf{w}_i + \theta_i \mathbf{r}_i$ and increase $i \leftarrow i+1$, else $\mathbf{w}_{i+1} = \mathbf{w}_i$.

end for

We first state the main theoretical result, that the entire complement is covered. However, before proving the result, we present a technical lemma.

Theorem 11. *Let μ_n denote the n-dimensional Lebesgue measure on a set. If $p_{\xi}(\xi|\mathbf{r},\mathbf{w}) > 0$ for all $\xi \in (0,\infty)$ and $\mathbf{r},\mathbf{w} \in \mathbb{R}^n$, then for any initial point $\mathbf{w}_0$ and any μ_n-measurable subset $\mathcal{A} \subset \mathbb{R}^n \setminus \mathcal{X}$,*

$$\lim_{N\to\infty} \mathbb{P}\big\{\mathbf{x}_N \in \mathcal{A} \;\big|\; \mathbf{w}_0\big\} > 0. \tag{D.1}$$

Recall that SB operates by generating direction vectors on the interior half-space defined by the supporting hyperplanes. Points of the complement of the set are generated by moving in the negative direction, i.e., directions on the exterior half-space of the supporting hyperplanes. Thus to generate points in a specific region $\mathcal{A} \subset \mathbb{R}^n \setminus \mathcal{X}$, the set $\mathcal{X}$ must have supporting hyperplanes that also act as separating hyperplanes between $\mathcal{X}$ and $\mathcal{A}$.

Lemma 1. *For any bounded set $\mathcal{A}'$, let*

$$\mathcal{W} := \left\{ \mathbf{w} \in \mathrm{bd}(\mathcal{X}) \;\middle|\; \exists \delta > 0 : \inf_{\mathbf{x}\in\mathcal{A}'} \nabla f(\mathbf{w})^{\mathsf{T}}\mathbf{x} \geq \nabla f(\mathbf{w})^{\mathsf{T}}\mathbf{w} + \delta \right\} \tag{D.2}$$

denote the points for which there is a corresponding supporting hyperplane of $\mathcal{X}$ that strongly separates $\mathcal{X}$ and $\mathcal{A}'$. If $\mathcal{W}$ is non-empty, then $\mu_{n-1}(\mathcal{W}) > 0$.

Proof. Select a point $\mathbf{w}_0 \in \mathcal{W}$ and let δ_0 be the corresponding slack variable defined in (D.2). Let $x^{\sup} = \sup_{\mathbf{x}\in\mathcal{A}'} \|\mathbf{x}\|_2$ and let $R = \max_{\mathbf{x},\mathbf{x}'\in\mathcal{X}} \|\mathbf{x}-\mathbf{x}'\|_2$ be the maximal diameter of $\mathcal{X}$. Finally, let

$$\mathcal{E} := \left\{ \mathbf{w}_0 - \boldsymbol{\epsilon} \;\middle|\; \|\boldsymbol{\epsilon}\|_2 < \frac{\delta_0}{\|\nabla f(\mathbf{w}_0)\|_2 + \|\mathbf{A}\|_{2,2}\,(R + x^{\sup})} \right\} \cap \mathrm{bd}(\mathcal{X})$$

denote the intersection of the boundary and a ball centered on $\mathbf{w}_0$. Note that $\mu_{n-1}(\mathcal{E}) > 0$ since it is the intersection of a ball and $\text{bd}(\mathcal{X})$. Thus, we only need to prove for each point in $\mathcal{E}$, that the supporting hyperplane of $\mathcal{X}$ is also a separating hyperplane, i.e., $\mathcal{E} \subset \mathcal{W}$. We show for any $\mathbf{w}_0 - \boldsymbol{\epsilon} \in \mathcal{E}$,

$$\nabla f(\mathbf{w}_0 - \boldsymbol{\epsilon})^\mathsf{T}\mathbf{x} > \nabla f(\mathbf{w}_0 - \boldsymbol{\epsilon})^\mathsf{T}(\mathbf{w}_0 - \boldsymbol{\epsilon}), \quad \forall \mathbf{x} \in \mathcal{A}'.$$

We proceed as follows:

$$\begin{aligned}
&\nabla f(\mathbf{w}_0 - \boldsymbol{\epsilon})^\mathsf{T}(\mathbf{w}_0 - \boldsymbol{\epsilon}) && && \text{(D.3)}\\
&\quad = (\mathbf{A}^\mathsf{T}\mathbf{w}_0 - \mathbf{A}^\mathsf{T}\boldsymbol{\epsilon} + \mathbf{b})^\mathsf{T}\mathbf{w}_0 - (\mathbf{A}^\mathsf{T}\mathbf{w}_0 - \mathbf{A}^\mathsf{T}\boldsymbol{\epsilon} + \mathbf{b})^\mathsf{T}\boldsymbol{\epsilon} && && \text{(D.4)}\\
&\quad = (\mathbf{A}^\mathsf{T}\mathbf{w}_0 + \mathbf{b})^\mathsf{T}\mathbf{w}_0 - \boldsymbol{\epsilon}^\mathsf{T}\mathbf{A}(\mathbf{w}_0 - \boldsymbol{\epsilon}) - (\mathbf{A}^\mathsf{T}\mathbf{w}_0 + \mathbf{b})^\mathsf{T}\boldsymbol{\epsilon} && && \text{(D.5)}\\
&\quad = \nabla f(\mathbf{w}_0)^\mathsf{T}\mathbf{w}_0 - \boldsymbol{\epsilon}^\mathsf{T}\mathbf{A}(\mathbf{w}_0 - \boldsymbol{\epsilon}) - \nabla f(\mathbf{w}_0)^\mathsf{T}\boldsymbol{\epsilon} && && \text{(D.6)}\\
&\quad \leq \nabla f(\mathbf{w}_0)^\mathsf{T}\mathbf{x} - \delta_0 - \boldsymbol{\epsilon}^\mathsf{T}\mathbf{A}(\mathbf{w}_0 - \boldsymbol{\epsilon}) - \nabla f(\mathbf{w}_0)^\mathsf{T}\boldsymbol{\epsilon} && \forall \mathbf{x} \in \mathcal{A}' && \text{(D.7)}
\end{aligned}$$

Inequality (D.7) follows from $\mathbf{w}_0 \in \mathcal{W}$ and applying (D.2). We then apply the Cauchy-Schwartz Inequality, decompose $\left\|\boldsymbol{\epsilon}^\mathsf{T}\mathbf{A}\right\|_2$ using the matrix norm, and bound $\|\mathbf{w}_0 - \boldsymbol{\epsilon}\|_2$ by the maximal diameter:

$$\begin{aligned}
\text{RHS(D.7)} &\leq \nabla f(\mathbf{w}_0)^\mathsf{T}\mathbf{x} - \delta_0 + \left\|\boldsymbol{\epsilon}^\mathsf{T}\mathbf{A}\right\|_2 \|\mathbf{w}_0 - \boldsymbol{\epsilon}\|_2 + \|\nabla f(\mathbf{w}_0)\|_2 \|\boldsymbol{\epsilon}\|_2 \quad \forall \mathbf{x} \in \mathcal{A}' && \text{(D.8)}\\
&\leq \nabla f(\mathbf{w}_0)^\mathsf{T}\mathbf{x} - \delta_0 + \|\boldsymbol{\epsilon}\|_2 \|\mathbf{A}\|_{2,2} R + \|\nabla f(\mathbf{w}_0)\|_2 \|\boldsymbol{\epsilon}\|_2 \quad \forall \mathbf{x} \in \mathcal{A}' && \text{(D.9)}
\end{aligned}$$

Note that for any $\mathbf{w}_0 - \boldsymbol{\epsilon} \in \mathcal{W}$, there exists a slack variable $\Delta\delta_0 > 0$ such that $\delta_0 = \Delta\delta_0 + \|\boldsymbol{\epsilon}\| (\|\nabla f(\mathbf{w}_0)\|_2 + \|\mathbf{A}\|_{2,2} R + \|\mathbf{A}\|_{2,2} x^{\text{sup}})$. Furthermore for any $\mathbf{x} \in \mathcal{A}'$, that $\boldsymbol{\epsilon}^\mathsf{T}\mathbf{A}\mathbf{x} \leq \|\boldsymbol{\epsilon}\|_2 \|\mathbf{A}\|_{2,2} x^{\text{sup}}$. Substituting these two terms into (D.9) yields

$$\begin{aligned}
\text{RHS(D.9)} &= \nabla f(\mathbf{w}_0)^\mathsf{T}\mathbf{x} - \Delta\delta_0 - \|\boldsymbol{\epsilon}\|_2 \|\mathbf{A}\|_{2,2} x^{\text{sup}} && \forall \mathbf{x} \in \mathcal{A}'\\
&\leq \nabla f(\mathbf{w}_0)^\mathsf{T}\mathbf{x} - \Delta\delta_0 - \boldsymbol{\epsilon}^\mathsf{T}\mathbf{A}\mathbf{x} && \forall \mathbf{x} \in \mathcal{A}'\\
&= \nabla f(\mathbf{w}_0 - \boldsymbol{\epsilon})^\mathsf{T}\mathbf{x} - \Delta\delta_0 && \forall \mathbf{x} \in \mathcal{A}'
\end{aligned}$$

thus completing the proof. □

Lemma 1 is the ellipsoid analogue of Boender et al. (1991, Lemma 2), which proved a similar result for polyhedra. With this, we now prove Theorem 11.

Proof of Theorem 11. Without loss of generality, let $\tilde{\mathbf{r}} = \xi\mathbf{r}$ and $p_{\tilde{\mathbf{r}}}(\tilde{\mathbf{r}}|\mathbf{w}) = p_{\mathbf{r}}(\mathbf{r}|\mathbf{w})p_{\xi}(\xi|\mathbf{r}, \mathbf{w})$. Let $p_{SB}(\mathbf{w})$ denote the stationary distribution of the hidden state SB algorithm. Let

$\mathcal{A}' \subset \mathbb{R}^n \setminus \mathcal{X}$ denote a μ_n-measurable set that can be strongly separated from $\mathcal{X}$, i.e., $\mathcal{W}$ as defined in (D.2) is non-empty. We first prove (D.1) for all $\mathcal{A}'$ with this specific structure and show that any $\mathcal{A} \subset \mathbb{R}^n \setminus \mathcal{X}$ contains a subset $\mathcal{A}' \subset \mathcal{A}$. Then, the probability for $\mathcal{A}'$ is a lower bound, i.e., $\mathbb{P}\{\mathbf{x}_N \in \mathcal{A} \mid \mathbf{w}_0\} \geq \mathbb{P}\{\mathbf{x}_N \in \mathcal{A}' \mid \mathbf{w}_0\}$, completing the proof.

Consider a set $\mathcal{A}'$ with the proposed structure. We will construct two measurable sets $\mathcal{W}$ and $\tilde{\mathcal{R}}(\mathbf{w})$ such that

$$\left\{\mathbf{w} - \tilde{\mathbf{r}} \;\middle|\; \mathbf{w} \in \mathcal{W}, \tilde{\mathbf{r}} \in \tilde{\mathcal{R}}(\mathbf{w})\right\} \subseteq \mathcal{A}'.$$

Given their existence, we can bound

$$\begin{aligned}
&\lim_{N\to\infty} \mathbb{P}\{\mathbf{x}_N \in \mathcal{A}' \mid \mathbf{w}_0\} \\
&\quad \geq \lim_{N\to\infty} \int_{\mathcal{W}} \mathbb{P}\{\mathbf{w}_N - \tilde{\mathbf{r}}_N \in \mathcal{A}' | \mathbf{w}_N\} p_{SB}(\mathbf{w}_N) d\mathbf{w}_N \\
&\quad \geq \lim_{N\to\infty} \int_{\mathcal{W}} \int_{\tilde{\mathcal{R}}(\mathbf{w}_N)} p_{\tilde{\mathbf{r}}}(\tilde{\mathbf{r}}_N|\mathbf{w}_N) p_{SB}(\mathbf{w}_N) d\tilde{\mathbf{r}}_N d\mathbf{w}_N.
\end{aligned}$$

First, let $\mathcal{W}$ be defined as in (D.2) as the set of points on the boundary of $\mathcal{X}$ for which the supporting hyperplane is a separating hyperplane between $\mathcal{X}$ and $\mathcal{A}'$. From Lemma 1, $\mu_{n-1}(\mathcal{W}) > 0$. Next for any $\mathbf{w}_N \in \mathcal{W}$, let

$$\tilde{\mathcal{R}}(\mathbf{w}_N) := \{\mathbf{w}_N - \mathbf{x} \mid \mathbf{x} \in \mathcal{A}'\}.$$

Because $\mu_n(\mathcal{A}') > 0$ and $\tilde{\mathcal{R}}(\mathbf{w}_N)$ is a translation, we must have $\mu_n(\tilde{\mathcal{R}}(\mathbf{w})) > 0$ as well. It remains to show that $p_{\tilde{\mathbf{r}}}(\tilde{\mathbf{r}}_N|\mathbf{w}_N) = p_{\mathbf{r}}(\mathbf{r}_N|\mathbf{w}_N) p_{\xi}(\xi_N|\mathbf{r}_N, \mathbf{w}_N) > 0$ for all $\tilde{\mathbf{r}} \in \tilde{\mathcal{R}}(\mathbf{w}_N)$. Since $\nabla f(\mathbf{w}_N)^{\mathsf{T}} (\mathbf{w}_N - \mathbf{x}) \leq 0$ for all $\mathbf{x} \in \mathcal{A}'$, the normalized vector $\mathbf{r}_N = (\mathbf{w}_N - \mathbf{x}) / \|\mathbf{w}_N - \mathbf{x}\|$ is a valid direction with $p_r(\mathbf{r}_N|\mathbf{w}_N) > 0$. Furthermore, by assumption in the Theorem statement, $p(\xi_N|\mathbf{r}_N, \mathbf{w}_N) > 0$ for $\xi_N = \|\mathbf{w}_N - \mathbf{x}\| > 0$. Therefore $p_{\tilde{\mathbf{r}}}(\tilde{\mathbf{r}}_N|\mathbf{w}_N) > 0$.

We now extend the proof to any arbitrary measurable set $\mathcal{A} \subset \mathbb{R}^n \setminus \mathcal{X}$ by showing that every $\mathcal{A}$ contains a subset of the structure of $\mathcal{A}'$, i.e., measurable and strongly separated from $\mathcal{X}$. First, for any $\mathbf{w} \in \mathrm{bd}(\mathcal{X})$, let

$$\mathcal{A}(\mathbf{w}) := \{\mathbf{x} \in \mathcal{A} \mid \nabla f(\mathbf{w})^{\mathsf{T}} \mathbf{x} \geq \nabla f(\mathbf{w})^{\mathsf{T}} \mathbf{w}\}$$

denote the intersection of $\mathcal{A}$ with a supporting hyperplane of $\mathcal{X}$. Now observe that the

measure of $\mathcal{A}$ admits a union bound using the (infinite) set of supporting hyperplanes:

$$\mu_n(\mathcal{A}) \leq \sum_{\mathbf{w} \in \mathrm{bd}(\mathcal{X})} \mu_n\left(\mathcal{A}(\mathbf{w})\right)$$

Because $\mu_n(\mathcal{A}) > 0$, at least one of the above subsets has positive measure. Select one such subset and let $\mathbf{w}_0$ be the corresponding boundary point. It remains to construct $\mathcal{A}' \subset \mathcal{A}(\mathbf{w}_0)$ such that $\mathcal{A}'$ is measurable and strongly separated from $\mathcal{X}$. For any $\kappa \in \mathbb{Z}^+$, let

$$\mathcal{H}_\kappa := \left\{ \mathbf{x} \;\middle|\; \nabla f(\mathbf{w}_0)^\mathsf{T} \mathbf{x} \geq \nabla f(\mathbf{w}_0)^\mathsf{T} \mathbf{w}_0 + \frac{1}{\kappa} \right\}.$$

and $\mathcal{B}_\kappa = \mathcal{A}(\mathbf{w}_0) \cap \mathcal{H}_\kappa$. Each $\mathcal{B}_\kappa$ is strongly separated from $\mathcal{X}$, meaning it has the structure required to satisfy Lemma 1 as assumed by $\mathcal{A}'$, and we argue that there must exist $\kappa \in \mathbb{Z}^+$ such that $\mu_n(\mathcal{B}_\kappa) > 0$.

To observe this, note that $\mathcal{A}(\mathbf{w}_0) = \cup_{\kappa=1}^{\infty} \mathcal{B}_\kappa$ is a union of ascending sets. By the continuity of the Lebesgue measure

$$\mu_n\left(\mathcal{A}(\mathbf{w}_0)\right) = \mu_n\left(\bigcup_{\kappa=1}^{\infty} \mathcal{B}_\kappa\right) = \lim_{\kappa \to \infty} \mu_n\left(\mathcal{B}_\kappa\right).$$

For any $\epsilon > 0$, there must exist κ such that $|\mu_n(\mathcal{A}(\mathbf{w}_0)) - \mu_n(\mathcal{B}_\kappa)| < \epsilon$. Setting $\epsilon < \mu_n(\mathcal{A}(\mathbf{w}_0))$ implies that $\mu_n(\mathcal{B}_\kappa) > 0$. Let $\mathcal{A}'$ be equal to any such subset. Therefore, any measurable set $\mathcal{A}$ contains a measurable subset $\mathcal{A}'$ of the required structure. Then, $\mathbb{P}\{\mathbf{x}_N \in \mathcal{A} \mid \mathbf{w}_0\} \geq \mathbb{P}\{\mathbf{x}_N \in \mathcal{A}' \mid \mathbf{w}_0\} > 0$, completing the proof. □

Bibliography

A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter. Differentiable convex optimization layers. In *Advances in Neural Information Processing Systems*, pages 9562–9574, 2019.

V. Alex, M. S. KP, S. S. Chennamsetty, and G. Krishnamurthi. Generative adversarial networks for brain lesion detection. In *Medical Imaging 2017: Image Processing*, volume 10133, page 101330G. International Society for Optics and Photonics, 2017.

B. Amos and J. Z. Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pages 136–145, 2017.

C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan. An introduction to mcmc for machine learning. *Machine Learning*, 50(1-2):5–43, 2003.

M. Angalakudati, Balwani S., J. Calzada, B. Chatterjee, G. Perakis, N. Raad, and J. Uichanco. Business analytics for flexible resource allocation under random emergencies. *Management Science*, 60(6):1552–1573, 2014.

L. M. Appenzoller, J. M. Michalski, W. L. Thorstad, S. Mutic, and K. L. Moore. Predicting dose-volume histograms for organs-at-risk in imrt planning. *Medical Physics*, 39(12):7446–7461, 2012.

M. Arjovsky and L. Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

A. Aswani, Z.-J. Shen, and A. Siddiq. Inverse optimization with noisy data. *Operations Research*, 66(3):870–892, 2018.

A. Aswani, Z.-J. Shen, and A. Siddiq. Data-driven incentive design in the medicare shared savings program. *Operations Research*, 67(4):1002–1026, 2019.

R. Atun, D. A. Jaffray, M. B. Barton, F. Bray, M. Baumann, B. Vikram, T. P. Hanna, F. M. Knaul, Y. Lievens, T. Y. M. Lui, M. Milosevic, B. O'Sullivan, D. L. Rodin, E. Rosenblatt, J. Van Dyk, M. L. Yap, E. Zubizarreta, and M. Gospodarowicz. Expanding global access to radiotherapy. *Lancet Oncology*, 16(10):1153–86, Sep 2015.

A. Babier, J. J. Boutilier, A. L. McNiven, and T. C. Y. Chan. Knowledge-based automated planning for oropharyngeal cancer. *Medical Physics*, 45(7):2875–2883, Jul 2018a.

A. Babier, J. J. Boutilier, M. B. Sharpe, A. L. McNiven, and T. C. Y. Chan. Inverse optimization of objective function weights for treatment planning using clinical dose-volume histograms. *Physics in Medicine & Biology*, 63(10):105004, May 2018b.

A. Babier, R. Mahmood, A. L. McNiven, A. Diamant, and T. C. Y. Chan. Knowledge-based automated planning with three-dimensional generative adversarial networks. *Medical Physics*, 47(2):297–306, 2020a.

A. Babier, R. Mahmood, A. L. McNiven, A. Diamant, and T. C. Y. Chan. The importance of evaluating the complete automated knowledge-based planning pipeline. *Physica Medica*, 72:73–79, 2020b.

R. Badenbroek and E. de Klerk. Complexity analysis of a sampling-based interior point method for convex optimization. *arXiv preprint arXiv:1811.07677*, 2018.

E. Balas. Disjunctive programming. In *Annals of Discrete Mathematics*, volume 5, pages 3–51. Elsevier, 1979.

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

P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio. Neural combinatorial optimization with reinforcement learning. *arXiv preprint arXiv:1611.09940*, 2017.

Y. Bengio, A. Lodi, and A. Prouvost. Machine learning for combinatorial optimization: a methodological tour d'horizon. *arXiv preprint arXiv:1811.06128*, 2018.

H. Y. Benson, D. F. Shanno, and R. J. Vanderbei. Interior-point methods for nonconvex nonlinear programming: jamming and numerical testing. *Mathematical Programming*, 99(1):35–48, 2004.

R. H. Bernhard. Mathematical programming models for capital budgeting—a survey, generalization, and critique. *Journal of Financial and Quantitative Analysis*, 4(2): 111–158, 1969. doi: 10.2307/2329837.

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

D. Bertsimas and C. McCord. Optimization over continuous and multi-dimensional decisions with observational data. In *Advances in Neural Information Processing Systems*, pages 2966–2974,, 2018.

D. Bertsimas and S. Vempala. Solving convex programs by random walks. *Journal of the ACM (JACM)*, 51(4):540–556, 2004.

D. Bertsimas, D. B. Brown, and C. Caramanis. Theory and applications of robust optimization. *SIAM review*, 53(3):464–501, 2011.

D. Bertsimas, V. Gupta, and I. C. Paschalidis. Inverse optimization: a new perspective on the Black-Litterman model. *Operations Research*, 60(6):1389–1403, 2012.

D. Bertsimas, V. Gupta, and I. Ch. Paschalidis. Data-driven estimation in equilibrium using inverse optimization. *Mathematical Programming*, 153(2):595–633, 2015.

C. G. E. Boender, R. J. Caron, J. F. McDonald, A. H. G. Rinnooy Kan, H. E. Romeijn, R. L. Smith, J. Telgen, and A. C. F. Vorst. Shake-and-bake algorithms for generating uniform points on the boundary of bounded polyhedra. *Operations Research*, 39(6): 945–954, 1991.

J. J. Boutilier, T. Craig, M. B. Sharpe, and T. C. Y Chan. Sample size requirements for knowledge-based treatment planning. *Medical Physics*, 43(3):1212–21, 2016.

L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996. ISSN 0885-6125.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

S. Brooks, A. Gelman, G. Jones, and X.-L. Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.

S. Bubeck and R. Eldan. The entropic barrier: Exponential families, log-concave geometry, and self-concordance. *Mathematics of Operations Research*, 44(1):264–276, 2019.

L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler. Annotating object instances with a polygon-rnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5230–5238, 2017.

T. C. Y. Chan and T. Lee. Trade-off preservation in inverse multi-objective convex optimization. *European Journal of Operational Research*, 270(1):25–39, 2018.

T. C. Y. Chan, T. Craig, T. Lee, and M. B. Sharpe. Generalized inverse multiobjective optimization with application to cancer therapy. *Operations Research*, 62(3):680–95, 2014.

T. C. Y. Chan, T. Lee, and D. Terekhov. Inverse optimization: Closed-form solutions, geometry, and goodness of fit. *Management Science*, 65(3):1115–1135, 2019.

E. Choi, S. Biswal, B. Malin, J. Duke, W. F. Stewart, and J. Sun. Generating multi-label discrete electronic health records using generative adversarial networks. *arXiv preprint arXiv:1703.06490*, 2017.

J. Y. J. Chow and W. W. Recker. Inverse optimization with endogenous arrival time constraints to calibrate the household activity pattern problem. *Transportation Research Part B: Methodological*, 46(3):463–479, 2012.

G. Cornuejols and R. Tütüncü. *Optimization methods in finance*, volume 5. Cambridge University Press, 2006.

D. Craft, P. Suss, and T. Bortfeld. The tradeoff between treatment plan quality and required number of monitor units in intensity-modulated radiotherapy. *International Journal of Radiation Oncology, Biology, Physics*, 67(5):1596–605, 2007.

I. J. Das, V. Moskvin, and P. A. Johnstone. Analysis of treatment planning time among systems and planners for intensity-modulated radiation therapy. *Journal of the American College of Radiology*, 6(7):514–7, Jul 2009. doi: 10.1016/j.jacr.2008.12.013.

G. Delaney, S. Jacob, C. Featherstone, and M. Barton. The role of radiotherapy in cancer treatment. *Cancer*, 104(6):1129–1137, 2005.

A. B. Dieker and S. S. Vempala. Stochastic billiards for sampling from the boundary of a convex set. *Mathematics of Operations Research*, 40(4):888–901, 2015.

P. Donti, B. Amos, and J. Z. Kolter. Task-based end-to-end model learning in stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 5484–5494, 2017.

A. N. Elmachtoub and P. Grigas. Smart "predict, then optimize". *arXiv preprint arXiv:1710.08005*, 2017.

C. Emmanouilidis, P. Pistofidis, L. Bertoncelj, V. Katsouros, A. Fournaris, C. Koulamas, and C. Ruiz-Carcel. Enabling the human in the loop: Linked data and knowledge in industrial cyber-physical systems. *Annual Reviews in Control*, 47:249–265, 2019.

R. Engelking. *General Topology*. Polish Scientific Publishers, 1977.

P. M. Esfahani, S. Shafieezadeh-Abadeh, G. A. Hanasusanto, and D. Kuhn. Data-driven inverse optimization with imperfect information. *Mathematical Programming*, 167(1): 191–234, 2018.

C. Esteban, S. L. Hyland, and G. Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*, 2017.

F. J. Fabozzi and J. Valente. Mathematical programming in american companies: A sample survey. *Interfaces*, 7(1):93–98, 1976.

K. J. Ferreira, B. H. A. Lee, and D. Simchi-Levi. Analytics for an online retailer: Demand forecasting and price optimization. *Manufacturing & Service Operations Management*, 18(1):69–88, 2015.

D. J. Foster, A. Sekhari, and K. Sridharan. Uniform convergence of gradients for non-convex learning and optimization. In *Advances in Neural Information Processing Systems*, pages 8759–8770, 2018.

M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan. Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. *arXiv preprint arXiv:1803.01229*, 2018.

A. Geretschläger, B. Bojaxhiu, A. Dal Pra, D. Leiser, M. Schmücking, A. Arnold, P. Ghadjar, and D. M. Aebersold. Definitive intensity modulated radiotherapy in locally advanced hypopharygeal and laryngeal squamous cell carcinoma: mature treatment results and patterns of locoregional failure. *Radiation Oncolology*, 10:20, Jan 2015.

A. Goli. Sensitivity and stability analysis for inverse optimization with applications in intensity-modulated radiation therapy. Master's thesis, University of Toronto, 2015.

A. Goli, J. J. Boutilier, T. Craig, M. B. Sharpe, and T. C. Y. Chan. A small number of objective function weight vectors is sufficient for automated treatment planning in prostate cancer. *Physics in Medicine & Biology*, 63(19):195004, 09 2018.

J. Gondzio. Interior point methods 25 years later. *European Journal of Operational Research*, 218(3):587–601, 2012.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*, volume 1. MIT press Cambridge, 2016.

V. Gupta and P. Rusmevichientong. Small-data, large-scale linear optimization with uncertain objectives. *Management Science*, 2020.

L. Hannah, W. Powell, and D. M. Blei. Nonparametric density estimation for stochastic optimization with an observable state variable. In *Advances in Neural Information Processing Systems*, pages 820–828, 2010.

R. Hermoza and I. Sipiran. 3d reconstruction of incomplete archaeological objects using a generative adversary network. *arXiv preprint arXiv:1711.06363*, 2017.

O. Hinder and Y. Ye. A one-phase interior point method for nonconvex optimization. *arXiv preprint arXiv:1801.03072*, 2018.

J. J. Hopfield and D. W. Tank. "neural" computation of decisions in optimization problems. *Biological Cybernetics*, 52(3):141–152, 1985.

K. Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

K.-L. Huang and S. Mehrotra. An empirical evaluation of walk-and-round heuristics for mixed integer linear programs. *Computational Optimization and Applications*, 55(3): 545–570, 2013.

P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

A. Kadurin, S. Nikolenko, K. Khrabrov, A. Aliper, and A. Zhavoronkov. druGAN: An Advanced Generative Adversarial Autoencoder Model for de Novo Generation of New Molecules with Desired Molecular Properties in Silico. *Molecular Pharmaceutics*, 14 (9):3098–3104, 2017.

Y.-H. Kao, B. V. Roy, and X. Yan. Directed regression. In *Advances in Neural Information Processing Systems*, pages 889–897, 2009.

V. Kearney, J. W. Chan, S. Haaf, M. Descovich, and T. D. Solberg. Dosenet: a volumetric dose prediction algorithm using 3d fully-convolutional neural networks. *Physics in Medicine & Biology*, 63(23):235022, Dec 2018.

A. Keshavarz, Y. Wang, and S. Boyd. Imputing a convex objective function. In *2011 IEEE International Symposium on Intelligent Control (ISIC)*, 2011.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

V. R. Konda and J. N. Tsitsiklis. Actor-critic algorithms. In *Advances in Neural Information Processing Systems*, pages 1008–1014, 2000.

W. W. M. Kool and M. Welling. Attention solves your tsp. *arXiv preprint arXiv:1803.08475*, 2018.

E. Larsen, S. Lachapelle, Y. Bengio, E. Frejinger, S. Lacoste-Julien, and A. Lodi. Predicting solution summaries to integer linear programs under imperfect information with machine learning. *arXiv preprint arxiv:1807.11876*, 07 2018.

D. A. Low, W. B. Harms, S. Mutic, and J. A. Purdy. A technique for the quantitative evaluation of dose distributions. *Medical Physics*, 25(5):656–661, 1998.

Z.-Q. Luo and W. Yu. An introduction to convex optimization for communications and signal processing. *IEEE Journal on Selected Areas in Communications*, 24(8):1426–1438, 2006.

R. Mahmood, A. Babier, A. McNiven, A. Diamant, and T. C. Y. Chan. Automated treatment planning in radiation therapy using generative adversarial networks. In Proceedings of Machine Learning Research, editor, *Machine Learning for Health Care*, volume 85, 2018.

H. Mahmoudzadeh and K. Ghobadi. Inferring linear feasible regions using inverse optimization. *arXiv preprint arXiv:2001.00143*, 2020.

O. L. Mangasarian. Arbitrary-norm separating plane. *Operations Research Letters*, 24 (1):15–23, 1999.

A. Maurer. A vector-contraction inequality for rademacher complexities. In *International Conference on Algorithmic Learning Theory*, pages 3–17. Springer, 2016.

J. F. McDonald. Sb algorithms for generating points which are approximately uniformly distributed over the surface of a bounded convex region. Technical report, Windsor Mathematics and Statistics Report 92-09, 1989.

C. McIntosh and T. G. Purdie. Voxel-based dose prediction with multi-patient atlas selection for automated radiotherapy treatment planning. *Physics in Medicine & Biology*, 62(2):415–431, Jan 2017.

C. McIntosh, M. Welch, A. McNiven, D. A. Jaffray, and T. G. Purdie. Fully automated treatment planning for head and neck radiotherapy using a voxel-based dose prediction and dose mimicking method. *Physics in Medicine & Biology*, 62(15):5926–5944, 2017.

miplib2017. MIPLIB 2017, 2018. http://miplib.zib.de.

V. V. Mišić. Optimization of tree ensembles. *forthcoming in Operations Research*, 2019.

J. Mula, D. Peidro, M. Díaz-Madroñero, and E. Vicens. Mathematical programming models for supply chain production and transport planning. *European Journal of Operational Research*, 204(3):377–390, 2010.

Y. Nesterov and A. Nemirovskii. *Interior-point polynomial algorithms in convex programming*, volume 13. SIAM, 1994.

B. Neyshabur, R. Tomioka, and N. Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401, 2015.

D. Nguyen, T. Long, X. Jia, W. Lu, X. Gu, Z. Iqbal, and S. Jiang. Dose prediction with u-net: A feasibility study for predicting dose distributions from contours using deep learning on prostate imrt patients. *arXiv preprint arXiv:1709.09233*, 2017.

D. Nguyen, X. Jia, D. Sher, M.-H. Lin, Z. Iqbal, H. Liu, and S. Jiang. 3d radiotherapy dose prediction on head and neck cancer patients with a hierarchically densely connected u-net deep learning architecture. *Physics in Medicine & Biology*, 64(6):065020, Mar 2019.

J.-S. Pang. A posteriori error bounds for the linearly-constrained variational inequality problem. *Mathematics of Operations Research*, 12(3):474–484, 1987.

M. Pelikan, D. E. Goldberg, and F. G. Lobo. A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21(1):5–20, 2002.

L. Perez and J. Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

K. Petersson, P. Nilsson, P. Engström, T. Knöös, and C. Ceberg. Evaluation of dual-arc vmat radiotherapy treatment plans automatically generated via dose mimicking. *Acta Oncologica*, 55(4):523–525, 2016.

B. D. Ripley. *Stochastic simulation*, volume 316. John Wiley & Sons, 2009.

R. T. Rockafellar. *Convex analysis*. Number 28. Princeton university press, 1970.

J. Saez-Gallego, J. M. Morales, M. Zugno, and H. Madsen. A data-driven bidding model for a cluster of price-responsive consumers of electricity. *IEEE Transactions on Power Systems*, 31(6):5001–5011, 2016.

P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays. Scribbler: Controlling deep image synthesis with sketch and color. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5400–5409, 2017.

F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

M. B. Sharpe, K. L. Moore, and C. G. Orton. Within the next ten years treatment planning will become fully automated without the need for human intervention. *Medical Physics*, 41(12), 2014.

S. Shiraishi and K. L. Moore. Knowledge-based prediction of three-dimensional dose distributions for external beam radiotherapy. *Medical Physics*, 43(1):378, 2016.

S. Shiraishi, J. Tan, L. A. Olsen, and K. L. Moore. Knowledge-based prediction of plan quality metrics in intracranial stereotactic radiosurgery. *Medical Physics*, 42(2):908, 2015.

R. L. Smith. Efficient monte carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research*, 32(6):1296–1308, 1984.

A. A. Stinnett and A. D. Paltiel. Mathematical programming for the efficient allocation of health care resources. *Journal of Health Economics*, 15(5):641–653, 1996.

L. Theis, A. van den Oord, and M. Bethge. A note on the evaluation of generative models. In *International Conference on Learning Representations (ICLR 2016)*, pages 1–10, 2016.

M. D. Troutt. A maximum decisional efficiency estimation principle. *Management Science*, 41(1):76–82, 1995.

M. D. Troutt, W.-K. Pang, and S.-H. Hou. Behavioral estimation of mathematical programming objective function coefficients. *Management Science*, 53(3):422–434, 2006.

R. J. Vanderbei and D. F. Shanno. An interior-point algorithm for nonconvex nonlinear programming. *Computational Optimization and Applications*, 13(1-3):231–252, 1999.

O. Vinyals, M. Fortunato, and N. Jaitly. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700, 2015.

N. Wang, W. Zha, J. Li, and X. Gao. Back projection: an effective postprocessing method for gan-based face sketch synthesis. *Pattern Recognition Letters*, 2017.

B. Wu, F. Ricchetti, G. Sanguineti, M. Kazhdan, P. Simari, M. Chuang, R. Taylor, R. Jacques, and T. McNutt. Patient geometry-driven information retrieval for IMRT treatment plan quality control. *Medical Physics*, 36(12):5497–505, 2009.

B. Wu, F. Ricchetti, G. Sanguineti, M. Kazhdan, P. Simari, R. Jacques, R. Taylor, and T. McNutt. Data-driven approach to generating achievable dose-volume histogram objectives in intensity-modulated radiotherapy planning. *International Journal of Radiation Oncology, Biology, Physics*, 79(4):1241–7, 2011.

B. Wu, M. Kusters, M. Kunze-Busch, T. Dijkema, T. McNutt, G. Sanguineti, K. Bzdusek, A. Dritschilo, and D. Pang. Cross-institutional knowledge-based planning (KBP) implementation and its performance comparison to auto-planning engine (APE). *Journal of the European Society for Therapeutic Radiology and Oncology*, 123(1):57–62, 2017.

J. Wu, C. Zhang, T. Xue, B. Freeman, and J. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016.

T. Yang, E. C. Ford, B. Wu, M. Pinkawa, B. van Triest, P. Campbell, D. Y. Song, and T. R. McNutt. An overlap-volume-histogram based method for rectal dose prediction and automated treatment planning in the external beam prostate radiotherapy following hydrogel injection. *Medical Physics*, 40(1):011709, 2013.

J. Yoo, N. Ahn, and K.-A. Sohn. Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8375–8384, 2020.

K. C. Younge, R. B. Marsh, D. Owen, H. Geng, Y. Xiao, D. E. Spratt, J. Foy, K. Suresh, Q. J. Wu, F. Yin, S. Ryu, and M. M. Matuszak. Improving quality and consistency in nrg oncology radiation therapy oncology group 0631 for spine radiosurgery via knowledge-based planning. *International Journal of Radiation Oncology, Biology, Physics*, 100(4):1067–1074, Mar 2018. doi: 10.1016/j.ijrobp.2017.12.276.

L. Yuan, Y. Ge, W. R. Lee, F. F. Yin, J. P. Kirkpatrick, and Q. J. Wu. Quantitative analysis of the factors which affect the interpatient organ-at-risk dose sparing variation in IMRT plans. *Medical Physics*, 39(11):6868–78, 2012.

Q. Zhao, A. Stettner, E. Reznik, D. Segrè, and I. C. Paschalidis. Learning cellular objectives from fluxes by inverse optimization. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 1271–1276, Dec 2015.

J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017.

X. Zhu, Y. Ge, T. Li, D. Thongphiew, F. Yin, and Q. J. Wu. A planning quality evaluation tool for prostate adaptive IMRT based on machine learning. *Medical Physics*, 38(2): 719–26, 2011.