Robust Losses for Decision-Focused Learning

Noah Schutte[1] , **Krzysztof Postek**[2] , **Neil Yorke-Smith**[1]

[1]Delft University of Technology
[2]Independent Researcher

{n.j.schutte, n.yorke-smith}@tudelft.nl, krzysztof.postek@gmail.com

Abstract

Optimization models used to make discrete decisions often contain uncertain parameters that are context-dependent and estimated through prediction. To account for the quality of the decision made based on the prediction, decision-focused learning (end-to-end predict-then-optimize) aims at training the predictive model to minimize regret, i.e., the loss incurred by making a suboptimal decision. Despite the challenge of the gradient of this loss w.r.t. the predictive model parameters being zero almost everywhere for optimization problems with a linear objective, effective gradient-based learning approaches have been proposed to minimize the expected loss, using the empirical loss as a surrogate. However, empirical regret can be an ineffective surrogate because empirical optimal decisions can vary substantially from expected optimal decisions. To understand the impact of this deficiency, we evaluate the effect of aleatoric and epistemic uncertainty on the accuracy of empirical regret as a surrogate. Next, we propose three novel loss functions that approximate expected regret more robustly. Experimental results show that training two state-of-the-art decision-focused learning approaches using robust regret losses improves test–sample empirical regret in general while keeping computational time equivalent relative to the number of training epochs.

1 Introduction

Real-world optimization problems, often formulated and solved as mixed-integer linear problems (MIPs) – such as shortest path problems or machine scheduling problems – involve parameters whose value is not known exactly. It is natural to use data to predict the uncertain parameters' values based on contextual information.

In predictive (regression) problems the goal is to make the most accurate prediction possible in the sense of prediction error. However, since there remains uncertainty around the predictions, and their purpose is to more accurately solve a downstream optimization problem, such a measure of prediction accuracy is of little relevance: the quality of the resulting decisions is what is important. This is the main premise of *decision-focused learning* (DFL) (also named predict+optimize or smart/end-to-end predict-then-optimize), and contrasts with *prediction-focused learning* (PFL) which is focused on prediction accuracy. DFL was first pioneered specifically for portfolio selection by Bengio [1997], and more generally introduced by Elmachtoub and Grigas [2022].

Learning based on decision errors obtained through solving a MIP faces difficulty, since the gradient of the decisions relative to the predicted parameters is zero almost everywhere and otherwise undefined [Vlastelica *et al.*, 2020]. Because of this, approximation methods have been introduced and shown to be more effective than PFL [Elmachtoub and Grigas, 2022; Berthet *et al.*, 2020]. The loss used in these methods is based on *empirical regret*, i.e., the loss incurred from not making the optimal decision for a given, empirical scenario. However, we observe that minimizing empirical regret deviates from minimizing expected regret as it leads to a form of overfitting on the empirical samples.

Recognizing this shortcoming in the most commonly used loss function in DFL, this paper develops three contributions:

1. We examine the usage of empirical regret from an uncertainty perspective, showing how empirical regret can lead to poor generalization and biased learning towards uncertain parameters with high variance.

2. We propose three different robust loss functions that simultaneously (1) improve conditional mean estimation, (2) are robust against errors in the mean estimation, and (3) do not increase computational expense relative to the number of training epochs.

3. We study two state-of-the-art gradient approximation DFL methods using the proposed robust losses compared to them using empirical regret. On three experimental problems, average test–sample regret is improved without additional overhead.

2 Problem Formulation

In this work we adopt the problem setting as introduced in Elmachtoub and Grigas [2022]. Observing some contextual information in the form of feature values $z \in \mathbb{R}^m$, the goal of the decision maker is to solve the following stochastic optimization problem with linear objective:

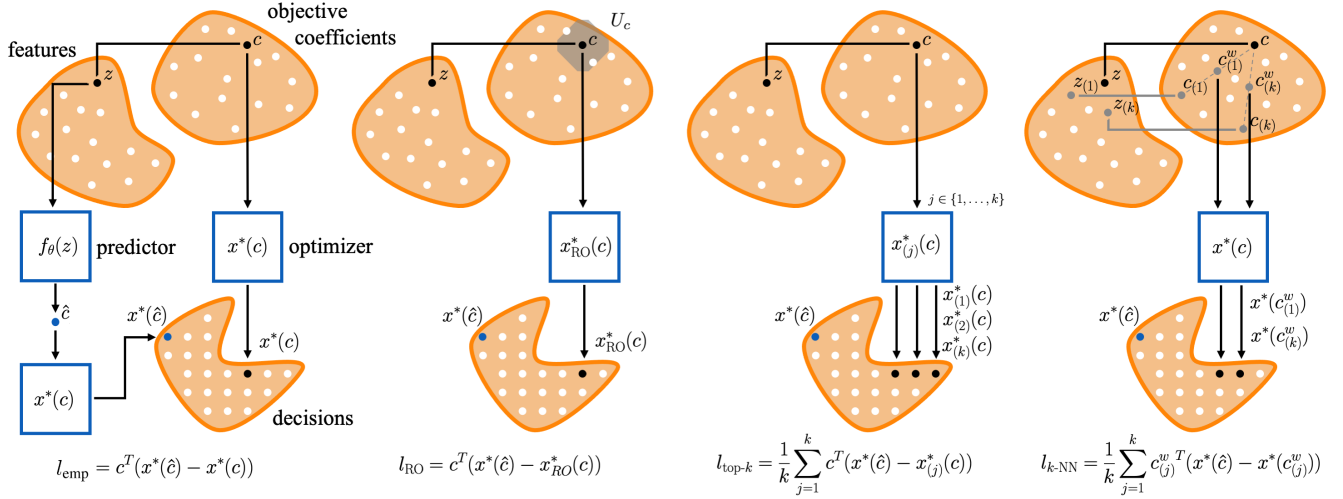$$\min_{x \in X} \mathbb{E}_{c \sim \mathcal{C}_z}[c^T x | z], \qquad (1)$$

Figure 1: Visualization of the empirical regret loss in DFL (left) and the robust losses in comparison (predictive pipeline is equal). The robust losses are constructed through using an optimization model (optimizer) that is robust against the mean estimation error and/or using a different mean estimator than empirically observed c. From left to right: empirical regret (l_{emp}), RO loss (l_{RO}), top-k loss ($l_{\text{top-}k}$), k-NN loss ($l_{k\text{-NN}}$).

where X denotes the set of feasible decisions and $\mathcal{C}_z$ is the conditional distribution of uncertain objective coefficients $c \in \mathbb{R}^n$ given feature values z. Since $\mathcal{C}_z$ is unknown, training a parametric predictor $f_\theta(z)$ can assist in picking good decisions. Ideally we find a predictor s.t. for each z we make a decision that has an equal objective to (1), i.e.,

$$\mathbb{E}_{c\sim\mathcal{C}_z}[c|z]^T \left[\operatorname*{argmin}_{x\in X} f_\theta(z)^T x \right] = \min_{x\in X} \mathbb{E}_{c\sim\mathcal{C}_z}[c^T x|z].$$

When the prediction problem is seen as separate from the optimization problem (PFL), regression is performed with coefficients c as responses to features z by minimizing prediction error. However, since the decision maker's goal is not to make accurate predictions but to make optimal decisions, it is preferable to minimize decision error, which is the main premise of DFL. To attain this goal, Elmachtoub and Grigas [2022] introduce a loss function based on the notion of *regret*, i.e., the loss incurred by not making the optimal decision given z:

$$l_{\text{emp}}(\hat{c}, c) = c^T x^*(\hat{c}) - c^T x^*(c), \tag{2}$$

where $\hat{c} = f_\theta(z)$, and $x^*(c) = \operatorname{argmin}_{x\in X} c^T x$ is the optimal decision when c are the objective coefficients. We will refer to this as *empirical regret*, as it is the regret in an empirical realization of (z, c). This loss is a natural choice, for when (z, c) is realized the loss is minimal when the empirical optimal decision is made. Figure 1 visualizes this loss and the robust losses that will be introduced in Section 3.

Due to the effectiveness of (stochastic) gradient descent in training predictive models, much research on DFL concentrates on finding effective gradient approximations. Approximations are necessary when the optimization problem is defined as in (1), as the linearity of the uncertain parameters in the objective function makes the true gradient zero almost everywhere and otherwise undefined. Gradient descent is based on the premise that minimizing empirical losses leads to minimizing the expected loss. The empirical loss is used

as a surrogate, while the main goal is to find predictive parameters such that for every feature realization z we have a minimal expected regret loss defined as:

$$l_{\mathbb{E}}(\hat{c}, \mathcal{C}_z) = \mathbb{E}_{c\sim\mathcal{C}_z}[c|z]^T x^*(\hat{c}) - \min_{x\in X} \mathbb{E}_{c\sim\mathcal{C}_z}[c^T x|z]. \tag{3}$$

Directly minimizing this loss is often impossible since in practice data is only available in the form of realized value pairs $D = \{(z_i, c_i) : i \in \{1, \ldots, t\}\}$. The empirical loss is a valid alternative, as the set of minimizers of (3) is equal to the set of minimizers of the expectation of (2), i.e.:

$$\min_\theta \{l_{\mathbb{E}}(f_\theta(z), \mathcal{C}_z)\} = \min_\theta \{\mathbb{E}_{c\sim\mathcal{C}_z}[l_{\text{emp}}(f_\theta(z), c)]\},$$

and therefore the assumption in the literature until now is that representative data should be sufficient to learn good predictive models through empirical regret. In this paper we challenge this assumption and show that when there is significant uncertainty empirical regret can be an ineffective surrogate. We propose three alternative losses that are more robust.

We begin by showing what issues can arise when training using empirical regret. The main intuition is the following: *When using empirical regret, training is biased towards empirical optimal decisions that are not necessarily expected optimal decisions*. This is due to a form of overfitting, i.e., uncertainty causing a mismatch between training and test data, and the non-smoothness of a discrete optimization problem, which can cause significant changes in optimal decisions by only small perturbations. To analyze the impact of uncertainty more formally, we make the distinction between *epistemic* and *aleatoric* [Kiureghian and Ditlevsen, 2009]. Epistemic uncertainty is uncertainty that occurs due to a lack of knowledge, which in practice often equates to not having enough representative data available. Aleatoric uncertainty is the uncertainty inherent to some underlying process and is therefore irreducible. In our DFL setting, we assume there exists a distribution $\mathcal{C}_z$ for every z that contains purely aleatoric uncertainty, but since we are limited by finite data the distributions $\mathcal{C}_z$ are unknown to us – the epistemic uncertainty.
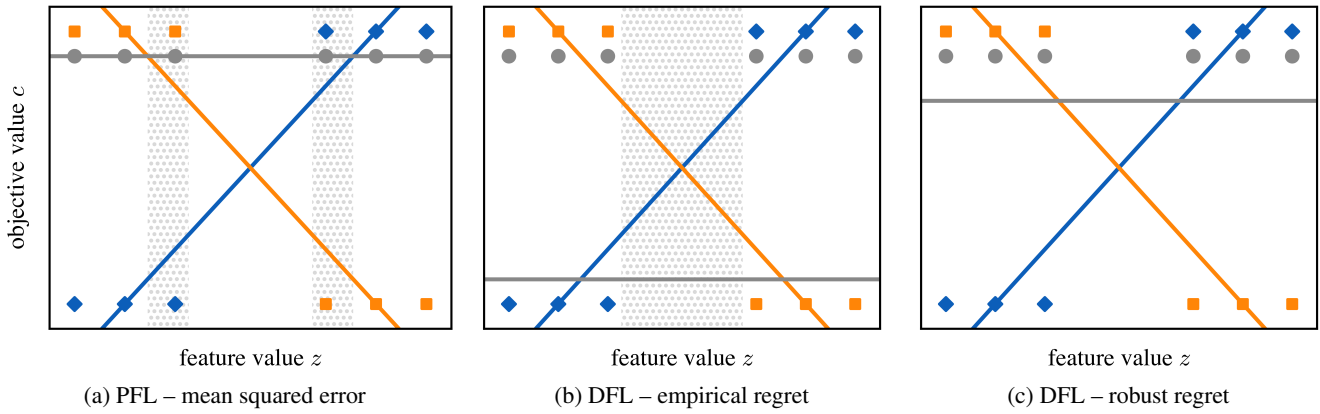
Figure 2: Observed coefficients c (profit) per feature value z (temperature) and optimal linear predictors $\hat{c} = f_\theta^*(z)$ according to different loss functions. The example problem is a coffee stand owner deciding what treat to make from their daily batch of chocolate. There are 3 possible weather related decisions: chocolate ice cream ◆, chocolate cookies ● or hot chocolate ■. In short: $\max_x c^T x$ s.t. $x_i \in \{0, 1\}$, $\sum_i x_i = 1$. The linear predictors are the lines with corresponding colour and shade. The shaded area ░ is considered to be sub-optimal. (It is determined assuming the expected value of c is equal to a linear interpolation between the observed points closest to the middle.)

2.1 Impact of Epistemic Uncertainty

Epistemic uncertainty makes generalization, i.e., adapting to unseen data, challenging as there is no complete knowledge, hence making it impossible to learn the exact underlying process. This general challenge for predictive problems is exemplified in DFL due to the focus on optimal decisions. When using PFL, the predictive model is trained to predict all coefficients as accurately as possible. When using DFL with empirical regret, the focus is on predicting the objective coefficients related to optimal decisions, i.e., objective coefficients that directly impact the empirically optimal decisions' objectives. This is because empirical regret (2) is based on the obtained objective given the optimal decision according to prediction $\hat{c}$. This means that if we would like to move towards the empirical optimal loss, i.e., $\hat{c}$ such that $c^T x^*(\hat{c}) = c^T x^*(c)$, only objective coefficients that relate to the empirical optimal objective value based on prediction $\hat{c}$ are relevant. Other objective coefficients are disregarded, which can lead to poor generalization.

Figure 2 gives an illustrative example in which PFL using the mean squared error and DFL using empirical regret both lead to likely sub-optimal predictors. Note that in Figure 2b, the linear predictors are empirically optimal as long as square (■) is predicted with low values of z and diamond (◆) with high values of z, which makes the linear predictor of circle (●) optimal as long as it does not predict the highest value for any of the realized points (c, z). This means the predictors denoted in Figure 2c are also optimal according to empirical regret, but it has such a wide range of parameter values that are considered optimal it is likely the linear predictor of circle (●) predicts low objective values.

2.2 Impact of Aleatoric Uncertainty

In general, a lack of enough representative data leads to epistemic uncertainty. However, even when there is plenty of representative data and only aleatoric uncertainty, we observe that using the empirical loss biases learning towards objective coefficients that have high variance, with variance being a measure of aleatoric uncertainty. This is because the predictive model parameters that affect high variance objective coefficient predictions $\hat{c}$ are more likely to be updated during training than those that affect low variance objective coefficient predictions $\hat{c}$. This can be problematic because it can make predictions of low variance objective coefficients less accurately (also with respect to decision quality), while it is these objective coefficients that relate to optimal decisions with a less uncertain objective value.

To provide intuition behind this, we again note that empirical regret (2) is based on the obtained objective given the optimal decision according to prediction $\hat{c}$. Now we extend previous reasoning to a probabilistic perspective: If certain coefficients are *more often* related to the empirical optimal objective value, they are more often updated during training, i.e., *biased learning* occurs. We provide an example of a simple optimization problem to show that the variance of objective coefficient can affect this. This is specific to DFL compared to regression due to the downstream optimization problem, as $\text{Var}(\min c) \neq \min \text{Var}(c)$ for some random variable c.

Example 1. *Consider optimization problem* (1)*, where decision set X is finite and each decision has its own objective coefficient, i.e., $X = \{x : x \in \{0, 1\}^n, \sum_{i=1}^n x_i = 1\}$, where we recall that $c \in \mathbb{R}^n$. Assume that all c_i are independently distributed with $\mathbb{E}_{c \sim \mathcal{C}_z}[c_i|z] = 0$ but different variances. In expectation, all decisions have equal objective value 0. However, empirically we observe the decision with the lowest realized coefficient as optimal. If the coefficient of decision variable x_i has $Var(c_i) = 0$, it is only optimal if all other coefficients are realized greater or equal to 0. If the coefficient of x_i has high variance, it is more likely for x_i to be optimal as its coefficient is more likely to realize the lowest value. Considering a dataset with realized values, decisions with higher variance coefficients are more often empirically optimal assuming similarly shaped distributions.*

A formal proof of this example is provided in the Appendix.

3 Robust Losses

We introduce novel DFL losses with the goal to mitigate the issues of poor generalization and biased learning that can arise when using empirical regret as a loss. We show that these issues can be mitigated by finding an estimator of $\mathbb{E}_{c \sim \mathcal{C}_z}[c|z]$ with lower variance than c and making the loss robust against the estimation error. Based on this we define three novel loss functions. First, we note that the predictor $f(z) = c$ would lead to an empirical regret loss of 0 that is unattainable due to $\mathcal{C}_z$ being the underlying distribution given z. A perfect predictor would be $f(z) = \mathbb{E}_{c \sim \mathcal{C}_z}[c|z]$, which is because of the assumption of the uncertain parameters c being linear in the objective function, as

$$\min_{x \in X} \mathbb{E}_{c \sim \mathcal{C}_z}[c^T x|z] = \min_{x \in X} \mathbb{E}_{c \sim \mathcal{C}_z}[c|z]^T x$$

leads to an expected regret loss of 0. This is important as it shows that a deterministic predictor can be expressive enough and we do not need to be able to predict the whole distribution $\mathcal{C}_z$ to have a strong predictor. This also means that we can still consider the deterministic problem $x^*(f(z))$ as the problem the decision maker solves when observing feature values z in practice. This consideration is important in designing the robust losses, as this means that during training we can still evaluate the loss by solving a deterministic problem without losing performance. Since DFL using gradient descent is already relatively expensive due to frequently having to solve an optimization problem, it is important computation time is not increased further [Mulamba *et al.*, 2021].

To find a good surrogate for the expected regret loss, we first look at the empirical regret as surrogate. When we optimize using empirical regret, effectively the realized c are used as an estimate of $\mathbb{E}_{c \sim \mathcal{C}_z}[c|z]$ when considering the expected regret as the true regret. Since the realized c is a sample realization from $\mathcal{C}_z$, the expected estimation error is equal to the variance of $\mathcal{C}_z$. This means that the empirical optimal decision $x^*(c)$ can be significantly different from the expected optimal decision $x^*(\mathbb{E}_{c \sim \mathcal{C}_z}[c|z])$, which leads us to two main principles in designing the robust losses:

1. Determine an estimator of $\mathbb{E}_{c \sim \mathcal{C}_z}[c|z]$ that has lower variance than c.

2. Determine close-to optimal decisions that are robust against the estimation error of $\mathbb{E}_{c \sim \mathcal{C}_z}[c|z]$ as an alternative to $x^*(c)$.

We introduce two losses based on the second principle, followed by a loss based on the first principle that naturally extends to a generalization that also includes the second principle. Figure 1 visualizes the three proposed losses compared to empirical regret.

3.1 Robust Optimization (RO) Loss

Assuming c is an erroneous estimation of $\mathbb{E}_{c \sim \mathcal{C}_z}[c|z]$, we consider empirical optimal decision $x^*(c)$ as an erroneous optimal decision. Instead of using $x^*(c)$, we can use the decision that minimizes the worst-case value of this error given some assumptions, i.e., robust optimal decision

$$x^*_{\text{RO}}(c) = \min_{x \in X} \max_{c \in U_c} c^T x, \qquad (4)$$

where U_c is an *uncertainty set*, i.e., a set that specifies the possible estimation error of c. This formulation is commonly used in Robust Optimization (RO), where the goal is to find decisions that are optimal given the worst case in some specified uncertainty set [Bertsimas *et al.*, 2011].

Since in RO the retrieved optimal decision is robust against the uncertainty specified by the uncertainty set, a proper specification of the latter is important. An uncertainty set that is too small makes the optimal decision not robust against the actual uncertainty, while an uncertainty set that is too large leads to conservative decisions that could be far from expected optimal. Further, depending on the uncertainty set the optimization problem (4) has a tractable reformulation. We propose the RO loss defined as follows:

$$l_{\text{RO}}(\hat{c}, c) = c^T(x^*(\hat{c}) - x^*_{RO}(c)).$$

As a choice of uncertainty set we propose the *budget uncertainty set*, which is a special case of a polyhedron uncertainty set [Bertsimas and den Hertog, 2022]. This set is practical as it is not too conservative while it allows for a tractable reformulation of linear optimization problems, i.e., if the optimization problem has a linear objective and only linear constraints specifying the decision set X the reformulation does not increase complexity. Our setting is not limited to this model class, but having a linear tractable reformulation is helpful as it does not change the class of the problem: the budget uncertainty set reformulation of the MIPs we consider is a MIP. Since the robust formulation adds constraints and auxiliary decision variables, the problem can become harder (albeit also easier) to solve. However this is highly dependent on the problem and is not straightforward to quantify.

The budget uncertainty set allows us to model the estimation error of c as percentage deviation $\zeta \in \mathbb{R}^n$. We limit the individual coefficient percentage deviation and the total percentage deviation by ρ and Γ respectively, giving

$$U_c = \{c \circ (1 + \zeta) : ||\zeta||_\infty \le \rho, ||\zeta||_1 \le \Gamma\},$$

where $\circ$ denotes the Hammard product. Depending on knowledge of the optimization problem and/or the uncertainty around c the uncertainty set can be adjusted. Our goal here is to show a proof of concept, leaving for the future a full exploration of specifying uncertainty sets in a DFL setting.

3.2 Best k Decisions (Top-k) Loss

An alternative to finding a decision that is robust against erroneous estimations is to consider multiple decisions that are close-to optimal. For instance, suppose that in the example in Figure 2 the second-best decision is also considered as optimal, then its coefficients would not be disregarded. Looking at multiple close-to optimal decisions can lead to a good area in the decision space, compared to a single optimal decision that might lay in a narrow global optimum. Moreover, high variance of c as an estimate of $\mathbb{E}_{c \sim \mathcal{C}_z}[c|z]$ can lead to significantly different decisions, while considering multiple close-to optimal decisions can lead to certain decisions being found more often and therefore a more stable signal.

The process of finding multiple quality decisions is not complicated, as any solving method can be used subsequently while excluding already found decisions. Some methods even

record suboptimal decisions and therefore a single solve could suffice. This makes the loss we propose viable, as we propose a loss that evaluates regret against the best k decisions (top-k):

$$l_{\text{top-}k}(\hat{c}, c) = \frac{1}{k} \sum_{j=1}^{k} c^T (x^*(\hat{c}) - x^*_{(j)}(c)),$$

where $x^*_{(j)}(c) := \operatorname{argmin}_{x \in X \setminus \{x^*_{(1)}(c), \dots, x^*_{(j-1)}(c)\}} \{c^T x\}$.

Alternatively one could refrain from specifying the number of decisions that is considered as good enough, but instead specifying a certain quality metric like a certain percentage from the optimal objective. This would depend on the use-case and hence will not be considered here.

3.3 k-Nearest Neighbour (k-NN) Loss

An estimator of $\mathbb{E}_{c \sim \mathcal{C}_z}[c|z]$ that has lower variance than c has to utilize the training data, while preferably being unbiased and consistent. This makes k-nearest neighbour (k-NN) regression a strong candidate as it is a non-parametric regression that has extensively studied strong asymptotic uniform consistency results [Cheng, 1984] and more recently also finite-sample uniform consistency results [Jiang, 2019]. Being non-parametric is beneficial as we will train a parameterized predictive model using signals from the estimator. If the estimator would be in the same model class, this would potentially amplify predictive and decision errors due to model misspecification. The k-NN estimator given feature values z is defined as the arithmetic mean of $c_{(j)}$ where

$$(z_{(j)}, c_{(j)}) = \operatorname*{argmin}_{(z', c') \in D \setminus \bigcup_{i=1}^{j-1} \{(z_{(i)}, c_{(i)})\}} ||z' - z||$$

is the j-th closest data point in the feature space, $|| \cdot ||$ is some norm (we use the Euclidean norm) and D the set of data points. The consistency of the k-NN estimator is intuitive, as with an increasing number of data realizations, the k nearest neighbours converge in distance, i.e.,

$$\lim_{|D| \to \infty} ||z_{(j)} - z|| < \epsilon \quad \forall j \in \{1, \dots, k\}, \forall \epsilon > 0.$$

If the distance between feature values goes to zero, the conditional probability distributions $\mathcal{C}_{z_{(j)}}$ converge to $\mathcal{C}_z$ as well, assuming the underlying cumulative distribution function is continuous in z. This means that when the dataset grows infinitely large, the realized coefficient values $c_{(1)}, \dots, c_{(k)}$ are samples drawn from the same distribution $\mathcal{C}_z$. Taking the mean of this is then a sample mean and therefore a consistent estimator of $\mathbb{E}_{c \sim \mathcal{C}_z}[c|z]$ with variance $\operatorname{Var}_{c \sim \mathcal{C}_z}(c)/k$.

These consistency results do not extend to the decision space, as optimization problem $x^*(\cdot)$ is a discontinuous multi-valued function. Despite this, we have that if the k nearest neighbours are somewhat representative of a set of samples from conditional probability distributions $\mathcal{C}_z$, the obtained decisions $x^*(c_{(1)}), \dots, x^*(c_{(k)})$ could all be empirical optimal decisions given z. Since our k-NN estimator will still have some estimation error, we consider multiple potential optimal decisions in a similar fashion as in designing the top-k loss. This way we naturally include the second principle we proposed for designing robust losses. The loss we propose is the average over the empirical regret of k nearest neighbours:

$$l_{k\text{-NN}}(\hat{c}, c) = \frac{1}{k} \sum_{j=1}^{k} c_{(j)}^w{}^T (x^*(\hat{c}) - x^*(c_{(j)}^w)),$$

where the values of the neighbours are adjusted based on some interpolation weight $w \in [0, 1]$:

$$c_{(j)}^w = w c_{(j)} + (1 - w) c.$$

We introduce this interpolation weight to make sure the k-NN estimator and therefore loss function remains distinct for all observations (z, c). Without this weight, multiple different observations can have the same k nearest neighbours and therefore the same estimator. This is not a general problem in k-NN regression, but it arises because we use the estimator in-sample, i.e., we provide an estimate of observations c that have a known realized value. This problem arises especially when there is little data available as it leads to over-simplification of the existing relationships, i.e., increased assumption bias. This makes w a tool to find a sweet spot in the bias-variance trade-off [Geman *et al.*, 1992]. In the same work, the k-NN estimator is considered as an example where increasing k increases bias, while decreasing k increases variance. Due to our in-sample usage of the k-NN estimator, varying k does not solve the issue of different observations having equal estimators. We note that $w = 0$ makes the k-NN loss equal to the empirical regret loss and therefore this loss is more general.

3.4 Applicability

The second principle we introduce at the beginning of this section to design the proposed losses leads to adjusting target solutions, i.e., the solutions that are considered to be optimal given c. This only adjusts the second term of the empirical loss as shown in (2). It is important to note that gradients with respect to the predictive model parameters are independent of this second term. This means that both the RO loss and top-k loss do not change the true gradient. However, this is also not our aim, as for the class of problems we look at (1) the true gradient is zero almost everywhere and therefore uninformative. In the experimental evaluation we apply our losses to two state-of-the-art gradient-approximation approaches, Smart 'Predict, then Optimize' (SPO+) of Elmachtoub and Grigas [2022] and the Perturbed Fenchel–Young Loss (PFYL) of Berthet *et al.* [2020]. Similar to the approaches by Mulamba *et al.* [2021] and Sahoo *et al.* [2023], these gradient approximation have a term consisting of the empirical optimal solution $x^*(c)$. Only the k-NN loss has the potential to generalize to approaches that are independent of empirically optimal solution $x^*(c)$. Technical details are provided in the Appendix.

3.5 Computational Considerations

Training a predictive model using gradient-based approaches requires frequently evaluating (solving) optimization problem $x^*(\cdot)$. In every epoch, $x^*(f_\theta(z))$ needs to be evaluated for each data point (z, c) to obtain the loss and/or gradient. This is the most computationally expensive part of training and we will therefore quantify the computational expense as a number of problem evaluations. Looking at the empirical regret loss, $x^*(c)$ does not change during training, and can therefore be precomputed. Given t data points and s epochs, problem $x^*(\cdot)$ is solved $t * (s + 1)$ times.

The proposed robust losses increase precomputation time i.e., computation time before the training, but do not increase training time per epoch. For the k-NN and top-k loss the total number of evaluations is at most $t * (k + s)$. In practice s is significantly larger than k and therefore the increase in total time is small. For the RO loss, the same number of evaluations is done compared to the empirical regret loss. While it could be that solving the robust problem formulation takes longer, this is also during precomputation at most t times.

4 Experimental Evaluation

We compare SPO+ and PFYL (number of samples $M = 1$, perturbation amplitude $\sigma = 1$) trained using empirical regret as loss to the same approaches using the presented robust losses (RO, top-k, k-NN), with PFL using mean squared error as an additional baseline. All predictive models are linear models. We use Python-based open-source package *PyEPO* [Tang and Khalil, 2022] for the data generation of two experimental problems and the training, where the robust losses are implemented on top of the existing code. The k-NN loss is currently available in PyEPO. We use the Adam optimizer with learning rate 0.01 for the gradient descent and *Gurobi* version 10.0.1 [Gurobi Optimization, 2023] as the optimization problem solver. We compare the results on three experimental problems: shortest path, travelling salesperson, and energy-cost aware scheduling. Additional details on the experimental problems are provided in the Appendix.

For simplicity, we do not tune the hyperparameters of the robust losses, but use the same value for all problem configurations. For the top-k loss we use $k = 10$; the same for the k-NN loss where $w = 0.5$. For the RO loss we set $\rho = 0.5$ and $\Gamma = \frac{n}{8}$, where $n = |c|$. The batch size is 32.

Shortest path & travelling salesperson. The shortest path problem considers a decision maker that has as goal to find the shortest path from start (NW) to end (SE) over a pre-specified grid (10×10) with uncertain costs (objective coefficients), allowing only viable paths modelled by a vector of binary decision variables. The travelling salesperson problem also aims at finding a shortest path, but instead of a grid there is a set of fully connected nodes (20) that all need to be visited. For both problems data is generated as in PyEPO, with feature vectors z of size 5 and the polynomial degree parameter fixed at 6. We look at different noise half-width values $\bar{\epsilon} \in \{0, 0.5, 1\}$ that multiply the generated objective coefficients by $\epsilon \sim \mathcal{U}(1 - \bar{\epsilon}, 1 + \bar{\epsilon})$, and different training set sizes $t \in \{100, 1000\}$. We consider these parameter values to mimic different levels of aleatoric and epistemic uncertainty. The noise multiplies the existing data generating distribution by another distribution that increases variance and therefore aleatoric uncertainty, while the training set size is a measure of how much knowledge we have and therefore mimics epistemic uncertainty. In all cases a validation and test set of size 100 and 1000 are used respectively. The validation set is used to pick the best model found during training to prevent overfitting, no model-selection is done. 200 epochs are run for t = 100, and 100 epochs for t = 1000. Since the data generation is a random process, we run 20 generated datasets to be able to measure significance.

Energy-cost aware scheduling. As a third experimental problem we look at energy-cost aware scheduling [Simonis *et al.*, 1999] following precedent in a DFL setting [Mandi *et al.*, 2022]. The dataset consist of 789 days of historical energy price data at 30-minute intervals from 2011–2013 [Ifrim *et al.*, 2012]. The optimization problem consists of scheduling a given number of tasks (3) on a given number of machines (10) with a certain resource capacity. Each task has a given energy consumption per hour and the goal is to minimize the total daily energy consumption. Energy prices are uncertain and are therefore predicted. Instead of using the features in the dataset, we consider the predictive problem as a time series problem consisting of predicting the next day using the day before, both consisting of 48 values. This way the uncertain objective coefficients share the same feature vector as defined in problem formulation (1). We also add noise to the data and we use training sizes of $t \in \{100, 500\}$ with a validation and test set of 100 each. Given we have 789 days, we create 5 non-overlapping data partitions of the data preserving time order when $t = 100$. When $t = 500$ we use a single partition, and in both cases we omit the last 89 days. We run 50 epochs for every training run.

4.1 Results

Figure 3 shows all results. For the shortest path and travelling salesperson, we see that when $t = 100$ the robust loss performs significantly better than the empirical loss in 24 out of the 36 direct comparisons (equal problem, approach, t and $\hat{\epsilon}$) and it never performs significantly worse. When $t = 1000$, we see 13 significantly better and 4 significantly worse results, out of which 3 are on the shortest path problem with $\bar{\epsilon} = 0.0$. These results confirm our hypothesis that both more aleatoric and epistemic uncertainty lead to better performance of the robust losses. Due to the limited size of the dataset, we see only 2 significant results for the energy-cost aware scheduling problem. However, these are both in favour of the robust loss and in all other configurations a robust loss also performs best.

In general, the k-NN loss is the most effective loss, even when there is ample data. The RO loss is mostly effective for the shortest path, while the top-k loss is more effective for the travelling salesperson. We hypothesize this is due to optimization problem characteristics, where the difference between optimal solutions and either robust optimal solutions (RO) or close-to optimal solutions (top-k) is most relevant.

5 Discussion and Related Work

Our proposed losses are evaluated on DFL gradient-approximation approaches [Elmachtoub and Grigas, 2022; Berthet *et al.*, 2020], which assume a linear objective function. These approximations are needed since in general a quadratic objective is required to have defined non-zero gradients [Amos and Kolter, 2017; Veviurko *et al.*, 2024]. Using the Karush-Kuhn-Tucker conditions, exact methods can be applied to augmented linear programs by adding a quadratic [Wilder *et al.*, 2019] or log-barrier [Mandi and Guns, 2020] objective term. For MIPs, Ferber *et al.* [2020] introduce a cutting-plane approach to convert the discrete problem into an equivalent continuous problem, which is computationally expensive and therefore less applicable compared to gradient approaches.
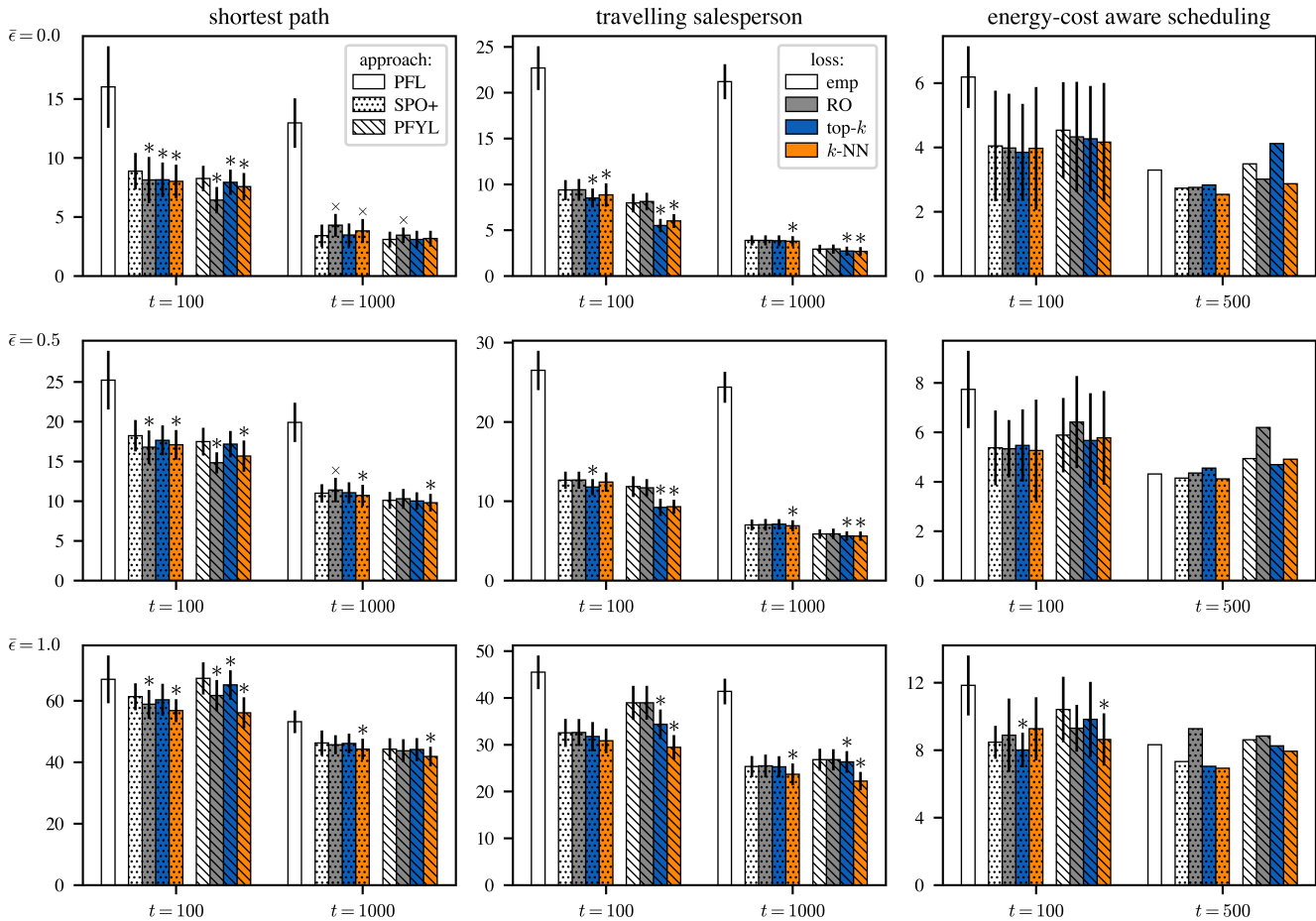
Figure 3: Test set mean normalized empirical regret in % (y-axes) on 3 experimental problems with different noise ($\bar{\epsilon}$) and training size (t) values. Approaches are denoted by patterns; used losses by colour (mean squared error is used for PFL). Error bars denote one in-sample standard deviation on both sides of the mean. Mean values of robust losses that are significantly different (paired t-test, $\alpha = 0.05$) from their empirical regret counterpart are denoted with $*$ (better) or $\times$ (worse). The results are shown as a table in the Appendix.

Gradient-approximation approaches have so far not been able to directly deal with a non-linear objective function or uncertain parameters in the constraints, but Silvestri *et al.* [2023] introduced score function gradient estimation for DFL, by predicting a distribution instead of a point.

Empirical regret is in general used as a loss in DFL and therefore studied here, but some alternative losses have been proposed. Mulamba *et al.* [2021] introduce a contrastive loss that uses non-optimal decisions as negative samples to attain more effective gradients. Mandi *et al.* [2022] extend on this work by defining DFL as a learning-to-rank problem, i.e., accurately ranking multiple decisions leads to being able to pick an optimal decision and we also consider this to be more robust. Experimental results show that losses in these two works are especially effective in reducing training time for comparable decision quality. Shah *et al.* [2022] propose an approach to learn losses that are convex and therefore practical for training, however these losses are learned based on empirical regret.

Since we show that the robust losses improve generalization when training data is limited, a natural alternative would be to apply regularization, as effectively used in deep learning

[Kukačka *et al.*, 2017]. Elmachtoub and Grigas [2022] use a mean squared or absolute error term for regularization. This allows for improving prediction accuracy while preserving decision quality [Tang and Khalil, 2022], but it does not improve decision quality which we do observe using robust losses.

6 Conclusion

DFL is gaining considerable recent interest due to its effective end-to-end approach to data-driven optimization. This paper investigated the shortcomings of using empirical regret in DFL for MIPs, and proposed a trio of robust loss functions to mitigate the issues of poor generalization and biased learning. Experimental results show improved test–sample empirical regret, especially with little or noisy data, without inflating computational time. Future work includes studying the effectiveness of the robust losses relative to each other based on optimization problem characteristics, studying RO uncertainty sets and fine-tuning hyperparameters. A more general direction in DFL is analyzing robustness of DFL predictors in settings where at test time the data and/or the optimization problem are different, as this is currently not well-studied.

Acknowledgements

References

[Amos and Kolter, 2017] Brandon Amos and J. Zico Kolter. OptNet: Differentiable optimization as a layer in neural networks. *International Conference on Machine Learning, PMLR*, 70:136–145, 2017.

[Bengio, 1997] Yoshua Bengio. Using a financial training criterion rather than a prediction criterion. *International Journal of Neural Systems*, 8:433–443, 1997.

[Berthet *et al.*, 2020] Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. Learning with differentiable perturbed optimizers. *Advances in Neural Information Processing Systems*, 33:9508–9519, 2020.

[Bertsimas and den Hertog, 2022] Dimitris Bertsimas and Dick den Hertog. *Robust and adaptive optimization*. Dynamic Ideas LLC, 2022.

[Bertsimas *et al.*, 2011] Dimitris Bertsimas, David B. Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53:464–501, 2011.

[Cheng, 1984] Philip E. Cheng. Strong consistency of nearest neighbor regression function estimators. *Journal of Multivariate Analysis*, 15:63–72, 1984.

[Elmachtoub and Grigas, 2022] Adam N. Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management Science*, 68:9–26, 2022.

[Ferber *et al.*, 2020] Aaron Ferber, Bryan Wilder, Bistra Dilkina, and Milind Tambe. MIPaaL: Mixed integer program as a layer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:1504–1511, 2020.

[Geman *et al.*, 1992] Stuart Geman, Elie Bienenstock, and Renc Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4:1–58, 1992.

[Gurobi Optimization, 2023] Gurobi Optimization. Gurobi Optimizer 10.0.1 reference manual, 2023. Accessed: 2024-04-23.

[Ifrim *et al.*, 2012] Georgiana Ifrim, Barry O'Sullivan, and Helmut Simonis. Properties of energy-price forecasts for scheduling. *International Conference on Principles and Practice of Constraint Programming*, pages 957–972, 2012.

[Jiang, 2019] Heinrich Jiang. Non-asymptotic uniform rates of consistency for k-nn regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:3999–4006, 2019.

[Kiureghian and Ditlevsen, 2009] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? Does it matter? *Structural Safety*, 31:105–112, 2009.

[Kukačka *et al.*, 2017] Jan Kukačka, Vladimir Golkov, and Daniel Cremers. Regularization for deep learning: A taxonomy. *arXiv preprint arXiv:1710.10686*, 2017.

[Mandi and Guns, 2020] Jayanta Mandi and Tias Guns. Interior point solving for LP-based prediction+optimisation. *Advances in Neural Information Processing Systems*, 33:7272–7282, 2020.

[Mandi *et al.*, 2022] Jayanta Mandi, Víctor Bucarey, Maxime Mulamba, and Tias Guns. Decision-focused learning: Through the lens of learning to rank. *International Conference on Machine Learning, PMLR*, 162:14935–14947, 2022.

[Mulamba *et al.*, 2021] Maxime Mulamba, Jayanta Mandi, Michelangelo Diligenti, Michele Lombardi, Victor Bucarey, and Tias Guns. Contrastive losses and solution caching for predict-and-optimize. *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 2833–2840, 2021.

[Sahoo *et al.*, 2023] Subham Sekhar Sahoo, Anselm Paulus, Marin Vlastelica, Vít Musil, Volodymyr Kuleshov, and Georg Martius. Backpropagation through combinatorial algorithms: Identity with projection works. *The Eleventh International Conference on Learning Representations*, 2023.

[Shah *et al.*, 2022] Sanket Shah, Kai Wang, Bryan Wilder, Andrew Perrault, and Milind Tambe. Decision-focused learning without decision-making: Learning locally optimized decision losses. *Advances in Neural Information Processing Systems*, 35:1320–1332, 2022.

[Silvestri *et al.*, 2023] Mattia Silvestri, Senne Berden, Jayanta Mandi, Ali İrfan Mahmutoğulları, Maxime Mulamba, Allegra De Filippo, Tias Guns, and Michele Lombardi. Score function gradient estimation to widen the applicability of decision-focused learning. *arXiv preprint arXiv:2307.05213*, 2023.

[Simonis *et al.*, 1999] Helmut Simonis, Barry O'Sullivan, Deepak Mehta, Barry Hurley, and Milan De Cauwer. CSPLib problem 059: Energy-cost aware scheduling. http://www.csplib.org/Problems/prob059, 1999. Accessed: 2024-04-23.

[Tang and Khalil, 2022] Bo Tang and Elias B. Khalil. PyEPO: A PyTorch-based end-to-end predict-then-optimize library for linear and integer programming. *arXiv preprint arXiv:2206.14234*, 2022.

[Veviurko *et al.*, 2024] Grigorii Veviurko, Wendelin Böhmer, and Mathijs de Weerdt. You shall pass: Dealing with the zero-gradient problem in predict and optimize for convex optimization. *arXiv preprint arXiv:2307.16304*, 2024.

[Vlastelica *et al.*, 2020] Marin Vlastelica, Anselm Paulus, Vít Musil, Georg Martius, and Michal Rolínek. Differentiation of blackbox combinatorial solvers. *International Conference on Learning Representations*, 2020.

[Wilder *et al.*, 2019] Bryan Wilder, Bistra Dilkina, and Milind Tambe. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:1658–1665, 2019.