arXiv:2405.02183v1 [cs.LG] 3 May 2024

METALEARNERS FOR RANKING TREATMENT EFFECTS

A PREPRINT

Toon Vanderschueren[*]
KU Leuven, University of Antwerp

Wouter Verbeke
KU Leuven

Felipe Moraes
Booking.com

Hugo Manuel Proença
Booking.com

May 6, 2024

ABSTRACT

Efficiently allocating treatments with a budget constraint constitutes an important challenge across various domains. In marketing, for example, the use of promotions to target potential customers and boost conversions is limited by the available budget. While much research focuses on estimating causal effects, there is relatively limited work on learning to allocate treatments while considering the operational context. Existing methods for uplift modeling or causal inference primarily estimate treatment effects, without considering how this relates to a profit maximizing allocation policy that respects budget constraints. The potential downside of using these methods is that the resulting predictive model is not aligned with the operational context. Therefore, prediction errors are propagated to the optimization of the budget allocation problem, subsequently leading to a suboptimal allocation policy. We propose an alternative approach based on learning to rank. Our proposed methodology directly learns an allocation policy by prioritizing instances in terms of their incremental profit. We propose an efficient sampling procedure for the optimization of the ranking model to scale our methodology to large-scale data sets. Theoretically, we show how learning to rank can maximize the area under a policy's incremental profit curve. Empirically, we validate our methodology and show its effectiveness in practice through a series of experiments on both synthetic and real-world data.

Keywords Causal Inference · Treatment Effect Estimation · Learning to Rank

1 Introduction

Decision-makers need to deal with uncertainty regarding the consequences of their decisions. An increasingly popular paradigm to address this challenge is the prediction-optimization framework. In a first *prediction* stage, data is used to estimate the effect of possible actions. In a second *optimization* stage, these predictions are integrated in an optimization problem with the aim of assigning personalized treatment recommendations, i.e., allocating treatments to instances to optimize an objective function, while satisfying operational constraints. These problems are common in various domains: e.g., marketing [1], healthcare [2], maintenance [3], or policy design [4] (see Table 1 for some examples). We focus on a specific class of treatment recommendation problems where instances need to be prioritized for treatment (e.g., recommending whom to treat). Our goal is to learn a treatment policy that prioritizes the optimal instances for treatment. A key part of this problem is estimating each instance's response to a treatment–i.e., its treatment effects–from data using causal inference.

Prediction-Focused Learning: Effect Estimation A common approach to tackle treatment allocation problems is to first *predict* the effect of an action for each instance. To this aim, causal inference can support many decision-making problems: by analyzing the causal effect of past decisions, future decisions can be optimized. A common approach is to first *estimate the causal effect* of possible decisions using methods for treatment effect estimation. For example, in marketing, to estimate how different customers would respond to a marketing incentive. The effect estimates can be integrated in an *optimization* problem to make the final decisions regarding treatment allocation (e.g., to target a specific customer segment). This approach has been adopted to aid decision-making by a variety of technology and e-commerce companies [5, 6, 7].

[*]Correspondence to: toon.vanderschueren@gmail.com. Work conducted while the first author was an intern at Booking.com.

Application	Allocation problem	Treatment outcome	Objective	Constraints
Marketing	Targeted advertising	Conversion	Incremental sales	Marketing budget
Healthcare	Pandemic response	Infection	Mortality reduction	Vaccine supply
Maintenance	Preventive maintenance	Failure rate	Asset uptime	Available technicians
Policy design	Targeted subsidies	Bed net purchase	Malaria prevention	Public/policy budget

Table 1: *Treatment Allocation Examples.* We highlight several applications where treatment allocation is required, characterized by (1) an estimated treatment effect, (2) an optimization objective, and (3) operational constraints. In *marketing*, the goal is to target customer segments and drive conversions, while adhering to budget constraints. In *healthcare and epidemiology*, optimal vaccine allocation during pandemics aims to minimize population mortality, subject to vaccine supplies. In *maintenance*, technicians are assigned to maintain assets, prevent failures, and maximizing operational uptime. Finally, in *economics and policy design*, subsidy usage needs to be optimized by efficiently allocating public funds and maximizing health care impact.

Decision-Focused Learning: Effect Ranking Recent work, referred to as *decision-focused learning*, aims to integrate the learning and optimization steps. This approach recognizes that the predictive task (i.e., estimating treatment effects) is only part of a larger optimization problem (i.e., allocating treatments). By integrating the predictive model in the larger optimization pipeline, decision-focused learning aims to learn a predictive model that results in better performance for the downstream task [8]. The key idea is to align the construction of the predictive model with the optimization task.

We analyze a common type of treatment allocation problem, where treatments are allocated to the instances with the largest treatment effect. In these settings, we argue that directly learning an *effect ranking* might be more useful than *effect estimation*. As operational constraints might prevent decision-makers from treating every instance, we require knowing how to prioritize instances based on their treatment effect. Compared to independently estimating each instance's effect, we argue that directly learning the ranking across instances may yield better results. Because *effect estimation* prioritizes accurate and well-calibrated effect estimates, it overlooks the estimates' ranking and resulting decision quality (i.e., estimation and optimization are not aligned). While successful when predictions are perfect, this misalignment can result in suboptimal decision-making in reality. Conversely, our work demonstrates that *effect ranking* can directly optimize the quality of the treatment assignment (i.e., the ranking objective is perfectly aligned with the decision-making task). We contrast both approaches in Table 2. Additionally, empirical risk minimization only guarantees model generalization for the specific objective that was optimized for [9], further motivating us to find objectives that are aligned to the final optimization problem.

Contributions This work proposes decision-focused learning framework for treatment recommendation problems. We formalize the class of problems that can be tackled using effect ranking and discuss the underlying assumptions. We describe how learning to rank can be used for this task and propose different *causal metalearners for ranking effects*. Our contributions are as follows. (1) Conceptually, we formalize treatment allocation problems that can be solved using ranking and analyze the underlying assumptions (see Section 3). (2) Methodologically, we propose different metalearners for ranking treatment effects, based on pairwise and listwise ranking objectives that scale efficiently to large-scale data sets. We show how our proposed listwise objective directly optimizes the policy's area under the Qini curve (see Section 4). (3) Empirically, we compare our proposed effect ranking with effect estimation using synthetic and real-world data sets (see Section 5).

2 Related Work

In the following, we discuss two areas of related work. First, causal inference and, more specifically, estimating treatment effects. Second, learning policies for treatment recommendation.

2.1 Prediction-Focused Learning: Effect Estimation

Understanding the impact of an action on an instance or individual is crucial in a variety of domains where personalized decision-making is valuable, such as marketing, healthcare, or education. Central to this is causal inference: using data to draw conclusions regarding causal relationships. Especially relevant to our work is treatment (or causal) effect estimation, where the aim is to learn how some treatment, action, or intervention will affect an instance's outcome of interest. For a comprehensive review, we refer to Zhang et al. [10]. In the context of marketing, using machine

	Instances (e.g., customers)						Performance	
							MSE	Ranking
True treatment effect	*0.20*	*0.17*	*0.16*	*0.15*	*0.11*	*0.10*	—	—
Effect estimation: model 1	0.22	0.15	0.16	0.17	0.10	0.11	👍	👎
Effect ranking: model 2	0.25	0.22	0.20	0.15	0.10	0.05	👎	👍

Table 2: *Comparing Pointwise Estimation and Effect Ranking for Treatment Allocation.* This illustrative example shows the true treatment effect for several instances, ordered from largest to smallest. The first model estimates each instance's treatment effect fairly accurately in terms of MSE. However, the ranking of instances based on these estimates differs significantly from their true order, which will result in suboptimal treatment allocation when only some instances can be treated. For the second model, we observe the opposite scenario: estimates are poor in terms of MSE, but their ranking respects the true order.

learning (ML) for treatment effect estimation is generally referred to as uplift modeling, as discussed in several surveys [11, 12, 10].

Specialized methods have been proposed for estimating treatment effects. First, general strategies exist for learning these effects. Causal metalearners are modeling frameworks for effect estimation, compatible with various ML algorithms [13, 14, 15]. Relatedly, response transformation approaches transform an instance's outcome so that it can be modeled using a standard classifier [16, 11]. Second, ML algorithms have been adapted for effect estimation, such as decision trees [17] or random forests [18, 19].

While uplift modeling has traditionally focused on optimizing conversion, practitioners often seek to optimize other metrics related to their business and operational context. Recent work explores cost-sensitive or profit-driven uplift modeling, where the aim is to estimate and maximize profit and cost resulting from targeting policies [20, 21, 22, 23]. For example, the Incremental Profit per Conversion (IPC) has been proposed as a response transformation approach for incremental profit [24].

All work in this category aims to *estimate the effect* of a treatment (e.g., a customer's incremental conversion probability as a result of receiving a marketing incentive). As discussed in the introduction, these estimates can be used to design a treatment allocation policy (e.g., by targeting customers with a large estimatd effect). In practice, operational constraints (such as budget limitations) may call for more complex optimization procedures [25, 26, 27, 1]. The resulting treatment allocation problem requires solving a constrained optimization problem using the effect estimates as input.

2.2 Decision-Focused Learning: Effect Ranking

As argued before, the prediction-focused approach may suffer from a misalignment between prediction and optimization, leading to suboptimal treatment decisions. Additionally, empirical risk minimization only guarantees model generalization for the specific objective that was optimized [9]. Decision-focused learning aims to align the two phases by using an integrated approach and directly optimizing a predictive model for the final optimization problem. Related to this insight, a handful of methods have recently been proposed for treatment recommendation that aim to learn a ranking of instances in terms of their treatment effect. Although these methods were originally proposed in the context of uplift modeling–framing targeted marketing as a ranking problem–they are more generally applicable. Finally, it has been noted that any score–even non-causal estimands–can be used to prioritize instances for treatment, as long as it is a good proxy for the treatment effect's magnitude [28, 29].

Table 3 highlights related methods for ranking treatment effects, describing the metalearners and optimization strategies that were used, where we follow the literature on learning to rank [30, 31]. The first approach, *pointwise* ranking, relies on an estimate of the treatment effect and corresponds to prediction-focused learning. The second approach, *pairwise* ranking, aims to predict the relative ranking between instances over all pairs of instances. The final approach, *listwise* ranking, optimizes the ranking across all instances in the ranking simultaneously. In the literature on learning to rank, pairwise and listwise ranking approaches have surpassed pointwise approaches, with listwise methods typically performing best [31].

Most existing approaches for ranking effects rely on alternative objective functions that aim to integrate prediction and optimization using Lagrangian duality [32, 33, 34] or gradient estimation techniques [35]. Alternatively, the causal profit ranker [23] ranks instances in a post-processing stage using pointwise estimates of the expected conversion More

	Objective			Metalearners					
Ref.	*Point*	*Pair*	*List*	*Z*	*S*	*T*	*X*	*DR*	*R*
[32]	✓	✗	✗	✗	✓	✗	✗	✗	✗
[33]	✓	✗	✗	✗	✓	✗	✗	✗	✓
[36]	✓	✗	✓	✓	✗	✗	✗	✗	✗
[9]	✓	✓	✗	✗	✓	✗	✗	✗	✗
[23]	✓	✗	✗	✗	✓	✓	✗	✗	✗
[34]	✓	✗	✗	✗	✓	✗	✗	✗	✗
Ours	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 3: *Literature Table.* We categorize related work on effect ranking by differentiating between different (1) ranking approaches (point-, pair-, and listwise) and (2) metalearners.

advanced pairwise [9] and listwise [36] learning to rank have also been explored in this context. Conversely, our work explores pointwise, pairwise, and listwise objectives, as well as a wide variety of metalearners.

General methodologies have been proposed for learning a treatment policy, directly mapping an instance's characteristics to a recommended treatment [37, 38, 39, 40, 41, 42]. In the context of our work, these can be seen as pointwise approaches, as they do not consider the ranking structure of the optimization task.

3 Problem Formulation

In this work, we aim to learn a treatment policy that prioritizes instances for treatment to maximize the aggregate effect. As opposed to the existing work, which assumes a problem formulation implicitly, we explicitly formalize our problem setting. In doing so, we reveal the underlying assumptions required for our approach.

3.1 Notation and Optimization Problem

Let an instance (e.g., a customer) be described by a tuple (x_i, t_i, y_i), representing covariates $X \subset \mathbb{R}^d$, an administered treatment $t \in \{0, 1\}$, and the outcome to be optimized $Y \subset \mathbb{R}$. We denote the potential outcome Y associated with a treatment t as $Y(t)$ and an instances i's treatment effect as $\tau_i = Y_i(1) - Y_i(0)$ (e.g., a customer's incremental conversion probability resulting from receiving a discount). We aim to learn a policy π that assigns treatments to instances and maximizes the overall treatment effect, while respecting possible operational constraints. At test time, we assume n instances can be treated, subject to a treatment budget B with $B \leq n$. This yields the following optimization problem:

$$\begin{aligned} \max_{\mathbf{t_i}} \quad & \sum_{i=1}^{n} \tau_i(t_i) \\ \text{s.t.} \quad & \sum_{i=1}^{n} t_i \leq B \\ & t_i \in \{0, 1\} \;\; \forall \, i \in \{0, \ldots, n\} \end{aligned}$$

At test time, treatment effects τ are unknown and need to be estimated. To this end, we assume access to a historical data set $\mathcal{D} = \{x_i, t_i, y_i(t_i)\}_{i=1}^{m}$ describing past treatment decisions and the resulting outcomes. These data can be used to estimate the conditional average treatment effect (CATE): $\hat{\tau} = \mathbb{E}(Y(1) - Y(0) \mid X = x)$.

3.2 Assumptions

We make several assumptions regarding the causal structure of the data and the operational constraint. To estimate the causal effect from historical data, we require the standard assumptions for identifiability in causal inference [43] (see Appendix A for a more extensive discussion). Additionally, we more formally define the operational constraint. Although we assume that not all instances can be treated and, thus, instance prioritization is required, we assume that the exact budget is not known to the decision-maker a priori. More formally, we state there is no information regarding the budget B a priori:

Assumption 1 (Operational constraint)**.** *We assume the exact budget is unknown, but the expectation is uniformly distributed among* $\{1, \ldots, n\}$*:* $B \sim \mathcal{U}(1, n)$.

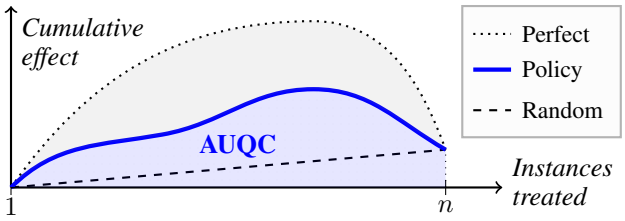


Figure 1: *Evaluating a Treatment Allocation Policy.* We compare targeting policies using a Qini curve, depicting the cumulative total effect of a policy for a number of treated instances, summarized by the area under the Qini curve (AUQC).

We discuss how alternative assumptions regarding the budget would affect our proposed solution below.

3.3 Evaluating a Treatment Policy

We assess the quality of a proposed policy using the Qini curve, illustrated in Figure 1. This curve shows the cumulative total effect of a policy for a number of treated instances [44, 45]. Given that we assume no information regarding the budget, we measure a policy's overall quality using the area under the Qini curve (AUQC), quantifying the total cumulative effect over the entire ranking. Formally, we define the (hypothetical) AUQC as

$$\text{AUQC} = \sum_{k=1}^{n} \sum_{i=1}^{k} \tau_i \tag{1}$$

with τ_i the effect of the instance at position i in the ranking. The normalized AUQC is obtained by comparing it with the expected AUQC of a random ranking and AUQC of a perfect ranking. Typically, the normalized AUQC ranges between zero and one $\in [0, 1]$, though a worse than random policy with AUQC < 0 is also possible. Because effects τ are not observed in reality, Qini curves need to be estimated from data on past treatment allocations [36, 45].

4 Methodology

Given the problem setup described above, we now present our proposed methodology, which essentially learns a ranking (or sorting) of each instance's treatment effect. The optimality of this solution can be seen as follows. If only one instance can be treated (i.e., $B = 1$), the optimal solution is to assign the treatment to the instance with the largest treatment effect τ_i. Given an unknown budget and uniform expectation regarding this budget, the optimal solution is then to rank all instances by their treatment effect τ_i and assign treatments to the top instances until the budget runs out. Therefore, our goal is to predict an optimal ordering or assignment policy $\pi \in \Pi_n$ that permutes the test instances $\{1, \ldots, n\}$ to the optimal ordering based on descending treatment effects τ.

As previously discussed, most existing approaches first estimates the effects $\hat{\tau}$ and then rank these estimates. However, as discussed above and in Table 2, this approach has two drawbacks. First, the estimator's objective is not aligned with the optimization task, possibly resulting in suboptimal decisions [8]. Second, the resulting model is only guaranteed to generalize for the predictive objective that was used [9]. These issues motivate us to directly learn a ranking policy π based on instance characteristics X, which requires addressing two challenges. First, to find an objective that optimizes a ranking of instances instead of a pointwise estimate (see Section 4.1). Second, the ranking needs to be based on the treatment effect τ, which is not observed. Therefore, we extend metalearners for effect estimation to ranking (see Section 4.2).

4.1 Optimizing a Ranking Objective

In this section, we explore approaches for optimizing a ranking. We discuss pointwise, pairwise, and listwise approaches. We propose a listwise objective that optimizes the policy's AUQC directly. Additionally, we propose a sampling strategy to improve the efficiency of our proposed ranking objectives.

4.1.1 Ranking objectives

We describe three objectives for learning a ranking policy π that can be used by ML algorithms.

Pointwise The first approach, used by most existing work, is to learn a pointwise estimate of the effect. As the treatment effect itself is never observed, this corresponds to either learning the observed outcome $y(t)$ or a transformed outcome, depending on the metalearner used (see Section 4.2). In this work, we use the mean squared error as a pointwise objective to learn the estimand:

$$\mathcal{L}_{\text{Point}}(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 . \tag{2}$$

While other objectives are possible [e.g. 34], pointwise approaches by definition ignore the instance ranking resulting from the point estimates. This motivates our exploration of alternative objectives.

Pairwise The first ranking approach is the pairwise approach. The idea is to predict, for each pair of instances, how both instances are ranked respective to each other. If all pairs are ranked correctly, the overall ranking will also be correct. We build upon the approach proposed for RankNet [46]. Before we define the pairwise objective, we define the pairwise outcome $y_{i,j}$ that specifies whether instance i or j should be ranked higher, for each pair of instances i and j:

$$y_{i,j} = \begin{cases} 1 & \text{if } y_i \geq y_j \\ 0 & \text{if } y_i < y_j . \end{cases}$$

Similarly, we define a smooth pairwise prediction $\hat{y}_{i,j}$, combining two instances' predictions $\hat{y}_i$ and $\hat{y}_j$ as follows:

$$\hat{y}_{i,j} = \frac{1}{1 + \exp\big(-\sigma(\hat{y}_i - \hat{y}_j)\big)}, \tag{3}$$

where the sigmoid parameter σ controls the smoothness of the comparison. In the extreme $\sigma = \infty$, this becomes a step function. This way, we define pairwise ranking as a binary classification task with the pairwise cross-entropy loss defined as:

$$\mathcal{L}_{\text{Pair}}(y, \hat{y}) = \sum_{i=1}^{n} \sum_{j=1}^{n} -y_{i,j} \log\left(\hat{y}_{i,j}\right) - (1 - y_{i,j})\big(1 - \log\left(\hat{y}_{i,j}\right)\big). \tag{4}$$

Other pairwise objectives exist [e.g. 9] which can be applied to our approach, though we consider this outside the scope of this work.

Listwise A drawback of the pairwise approach is that it overlooks the relative importance of correctly classifying one pair on the listwise ranking quality. To address this issue, the LambdaMART objective [47] adds a weight $\text{NDCG}_{i,j}$ to the pairwise objective, reflecting the increase in normalized discounted cumulative gain (NDCG, see below) achieved by swapping that pair:

$$\mathcal{L}_{\text{List}}(y, \hat{y}) = \sum_{i=1}^{n} \sum_{j=1}^{n} \Big(- y_{i,j} \log\left(\hat{y}_{i,j}\right) - (1 - y_{i,j})\big(1 - \log\left(\hat{y}_{i,j}\right)\big) \Big) \Delta\text{NDCG}_{i,j}.$$

4.1.2 The AUQC as a specific instance of the NDCG

The normalized discounted cumulative gain (NDCG) is a class of metrics measuring the quality of a ranking [48]. Formally, we define a ranking π, with π_i representing the i'th instance in the ranking. An instance's gain g_i represents its value independent of its position in the ranking (e.g., the treatment effect τ). The NDCG decreases the gain for lower ranks, reflecting their decreasing importance, by applying a discount function $d(i)$ to the instance's gain g_i. The discounted cumulative gain (DCG) is the sum of all discounted gains over the ranking $\text{DCG} = \sum_{i=1}^{n} d(i) g_{\pi_i}$. Finally, the normalized discounted cumulative gain (NDCG) is obtained by comparing the DCG with the perfect ranking's DCG to get a value between zero and one.

We propose a specific instantiation of the NDCG that matches the AUQC. More specifically, we define an instance's gain g_i as its treatment effect τ. The discount function is a linearly decreasing function: for rank i, the discount equals $(n - i + 1)$[2]. In this specification, we can show that the listwise objective that optimizes the NDCG allows us to learn an optimal ranking policy π that optimizes the metric of interest, the AUQC (generalizing [36, Section 3.2.2] from the Z-Learner to any model that estimates τ):

[2]Technically, we do not discount lower ranked instances ($d(i) \leq 1$), but rather promote higher ranked instances ($d(i) \geq 1$) [see 36, eq. 25].

Proof. The area under the Qini curve (AUQC) is an instantiation of the normalized discounted cumulative gain (NDCG):

$$\text{AUQC} = \sum_{k=1}^{n}\sum_{i=1}^{k}\tau_{\pi_i} = \sum_{i=1}^{n}\sum_{k=i}^{n}\tau_{\pi_i} = \sum_{i=1}^{n}\tau_{\pi_i}\sum_{k=i}^{n}1 = \sum_{i=1}^{n}\tau_{\pi_i}(n-i+1) = \text{NDCG}$$

for the NDCG with gain $g_{\pi_i} = \tau_{\pi_i}$ and $d(i) = n - i + 1$. □

Given that the listwise objective described above can be shown to optimize the NDCG [49, 47], this result proves that our proposed objective can be used to directly optimize the metric of interest: the AUQC. There is one remaining challenge: we do not known the instance's treatment effect τ and require a valid estimator $\hat{\tau} \to \tau$. We will tackle this part in Section 4.2.

4.1.3 Efficiently scaling to large-scale data sets

Moving from pointwise to pairwise or listwise optimization requires addressing a challenge regarding computational efficiency. Optimizing over pairs of instances results in an increase of the algorithm's time complexity from $O(n)$ to $O(n^2)$. This complexity is not compatible with the large data sets commonly encountered in applications such as marketing or e-commerce [27].

To address this challenge, we propose an efficient sampling procedure that finds a stochastic estimate of the gradient. Intuitively, instead of calculating the gradient based on all possible pairs, we sample k pairs per instance:

$$\mathcal{L}_{\text{Pair}}(y, \hat{y}) = \sum_{i=1}^{n}\sum_{j\in\mathcal{J}} -y_{i,j}\log\left(\hat{y}_{i,j}\right) - (1-y_{i,j})\big(1-\log\left(\hat{y}_{i,j}\right)\big) \text{ with } \mathcal{J} \sim \left(\mathcal{U}_{[1,\dots,n]}\right)^k,$$

and equivalently for $\mathcal{L}_{\text{List}}$. This again makes the procedure scale linearly in the number of instances with complexity $O(kn)$. We observe good results for $k = 1$, effectively obtaining the same computational complexity as the pointwise objective. We present a sensitivity analysis for the number of samples k below. We opt for this sampling procedure for its simplicity, although more advanced sampling schemes are possible [see e.g. 50].

4.2 Ranking Metalearners

One challenge when learning a treatment assignment policy π is that decisions need to be made based on the treatment effect τ. Indeed, the optimization of the AUQC presented above requires a model to predict the treatment effect $\hat{\tau}$. Predicting a treatment effect τ is a challenge. We never observe the treatment effect itself, but only one potential outcome $y(t)$ for each instance, i.e., the outcome when targeted or not targeted–also called the fundamental problem of causal inference [51]. In other words, we never actually observe what we are trying to estimate and optimize over.

To tackle this challenge, we implement the objective functions introduced above for different causal metalearners–general strategies for using any ML method for treatment effect estimation. Whereas metalearners have originally been proposed for effect estimation, we propose adaptations for effect ranking below. In practice, this adaptation consists of integrating ranking (i.e., pairwise or listwise) objectives in each training procedure–instead of the traditional pointwise (regression or classification) objectives. In this section, we describe each metalearner and introduce its ranking equivalent. We focus on several established metalearners, but the extension to other metalearners could be done using a similar approach. While our optimization of the AUQC requires metalearners that directly predict the treatment effect τ, we also discuss adaptations of other metalearners–specifically, the S- and T-Learner.

Z-Learner The first metalearner estimates a transformation z of the outcome y–also called the class transformation approach [52, 53, 24]–adjusting the outcome based on the instance's propensity score[3] $\hat{e}(x) = P(T = 1|X = x)$ and the observed treatment:

$$z_i = \begin{cases} y_i/\hat{e}_i & \text{if } t_i = 1 \\ -y_i/(1-\hat{e}_i) & \text{if } t_i = 0 \end{cases}$$

The Z-Learner estimates the treatment effect using this outcome:

$$f_{\mathtt{Z}}(x) = \mathbb{E}(Z|X) \quad \text{with} \quad \hat{\tau} = f_{\mathtt{Z}}(x)$$

Instead of training this final model $f_{\mathtt{Z}}(x)$ with a pointwise objective, we propose to optimize it using a pairwise or listwise objective.

[3] Given that we assume data comes from a randomized trial, we can estimate the propensity score by the proportion of treated instances without fitting a model.

S-Learner The S-Learner estimates a single model, which takes the treatment as a (regular) feature:

$$f_{\mathtt{S}}(x,t) = \mathbb{E}(Y|X=x,T=t) \quad \text{with} \quad \hat{\tau} = f_{\mathtt{S}}(x,1) - f_{\mathtt{S}}(x,0)$$

Again, the model $f_{\mathtt{S}}(x,t)$ can be trained with either a pointwise, pairwise, or listwise objective. For the ranking objectives, the estimated treatment effect τ corresponds to the increase in the ranking score resulting from receiving the treatment. Importantly, however, because this metalearner does not directly optimize for τ, the theoretical results of the previous section do not apply. Nevertheless, although there is no guarantee that the listwise objective optimizes the AUQC for the S-Learner, the difference in ranking scores could provide a good heuristic for the treatment effect.

T-Learner The T-Learner trains two models: one model for each treatment group— $f_{\mathtt{T}_1}$ for the treatment ($T=1$) and $f_{\mathtt{T}_0}$ for the control group ($T=0$)—trained as follows:

$$f_{\mathtt{T}_1}(x) = \mathbb{E}(Y|X=x,T=1), \quad f_{\mathtt{T}_0}(x) = \mathbb{E}(Y|X=x,T=0).$$

When combined, these can estimate the treatment effect as follows:

$$\hat{\tau} = f_{\mathtt{T}_1}(x) - f_{\mathtt{T}_0}(x).$$

We propose to train both models $f_{\mathtt{T}_1}(x)$ and $f_{\mathtt{T}_0}(x)$ with a pairwise or listwise objective, instead of the traditional pointwise objective. This corresponds to a separate optimization of the AUQCof the treatment ($f_{\mathtt{T}_1}(x)$) and control ($f_{\mathtt{T}_0}(x)$) groups (based on the outcome y instead of the effect τ). An instance's difference in ranking scores between both groups is used as a proxy for its treatment effect. Similarly to the S-Learner, the theoretical results from the previous section do not apply. Nevertheless, this separate optimization could be a good heuristic for the AUQC [see also 11, eq. 26].

X-Learner The X-Learner first estimates an initial treatment effect by imputing the counterfactual potential outcome using a T-Learner model as follows [14]:

$$D_i^1 = y_i - f_{\mathtt{T}_0}(x_i) \text{ if } t_i = 1, \quad D_i^0 = f_{\mathtt{T}_1}(x_i) - y_i \text{ if } t_i = 0.$$

The final two models are then trained on the imputed effects:

$$\hat{\tau}^0 = f_{\mathtt{X}}^0(x) = \mathbb{E}(D_i^0|X=x), \quad \hat{\tau}^1 = f_{\mathtt{X}}^1(x) = \mathbb{E}(D_i^1|X=x).$$

To obtain the final predicted effect $\hat{\tau}_i$, we combine these two models:

$$\text{with} \quad \hat{\tau} = g(x) f_X^0(x) + (1 - g(x)) f_X^1(x).$$

For the weighting function $g(x)$, we use the estimated propensity score $\hat{e}(x) = P(T|X=x)$ following [14]. Compared to the pointwise variant, we propose to train the final models $f_{\mathtt{X}}^0(x)$ and $f_{\mathtt{X}}^1(x)$ with pairwise or listwise ranking objectives, with the initial models still trained using a pointwise objective. The final ranking score $\hat{\tau}$ is a linear combination of two ranking scores [see 47, note 7.1].

DR-Learner The Doubly Robust or DR-Learner [54, 55] also relies on a final model estimating a pseudo-outcome. In this case, the first stage is based on pointwise estimates from a T-Learner and a propensity model $\hat{e}(x) = P(T|X=x)$. These estimates are combined to create a pseudo-outcome ϕ as follows:

$$\phi_i = \frac{t_i - \hat{e}(x_i)}{\hat{e}(x_i)(1-\hat{e}(x_i))}(y_i - f_{\mathtt{T}_{t_i}}(x_i)) + f_{\mathtt{T}_1}(x_i) - f_{\mathtt{T}_0}(x_i).$$

This pseudo-outcome is then used to learn a final model $\hat{\tau} = f_{\mathtt{DR}}(\phi|x)$, which can be learned using a point-, pair-, or listwise objective.

R-Learner For the R-Learner [56], we first fit an outcome model $\hat{m}(x) = \mathbb{E}(Y|X=x)$ and propensity model $\hat{e}(x) = P(T|X=x)$. These can then be used to minimize the R-Loss, based on Robinson's decomposition [57], which can be seen as a weighted MSE:

$$\mathcal{L}^R_{\text{MSE}}(y,\hat{y}) = \frac{1}{n}\sum_{i=1}^{n}\left((y_i - \hat{m}(x_i)) - (t_i - \hat{e}(x_i))\,\tau(x_i)\right)^2 = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{(t_i - \hat{e}(x_i))^2}\left(\left(\frac{y_i - \hat{m}(x_i)}{t_i - \hat{e}(x_i)}\right) - \tau(x_i)\right)^2.$$

Instead of this pointwise objective, we propose to use a weighted pair- or listwise objective based on the same weights and labels.

For simplicity, we do not use out-of-fold estimates for any of the intermediary models for any of the metalearners, but rather train all models on the same train set.

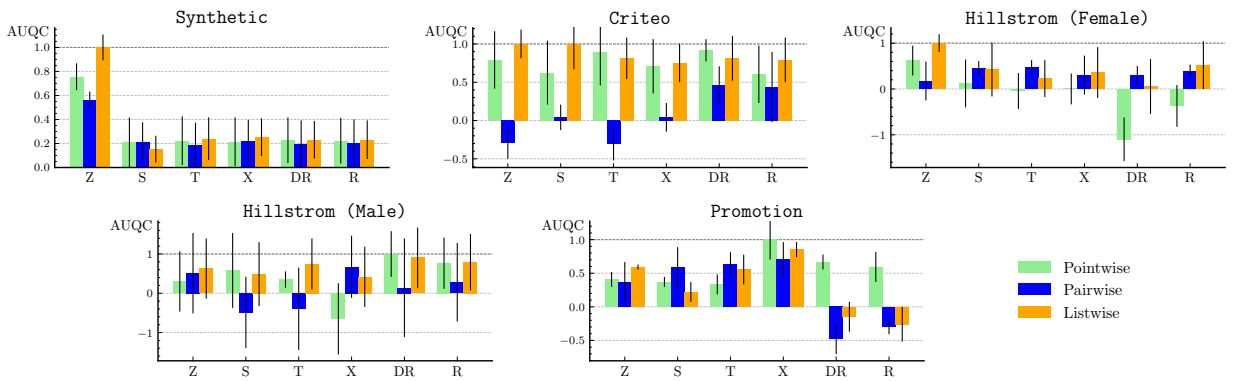


Figure 2: *Ranking Quality for Different Objectives and Metalearners.* For each metalearner, we compare a point-, pair-, and listwise version. We show the AUQC $\pm$ one standard error, scaled to have the best result $= 1$, for five different data sets.

5 Empirical Results

This section presents the empirical results, comparing our proposed (pairwise and listwise) *effect ranking* metalearners with traditional (pointwise) *effect estimation* metalearners. Our experiments aim to answer three research questions. (*RQ1*) What is the treatment recommendation quality resulting from the different methods, as measured in AUQC? (*RQ2*) What are the performance trade-offs of the pointwise, pairwise, and listwise objectives, in terms of MSE (i.e., pointwise accuracy), Kendall τ (i.e., pairwise rank correlation), and AUQC (i.e., listwise ranking quality)? (*RQ3*) How sensitive are our proposed methods to key hyperparameters? This section presents the setup of our experimental evaluation and the empirical results.

5.1 Data and Benchmarks

To evaluate the performance of the proposed approaches, we use a total of four data sets based on randomized trials: 1) a synthetic dataset; 2) Criteo [5]; 3) Hillstrom (Male and Female) [58]; 4) a proprietary data set from a promotion campaign at a global online travel agency.

We first simulate a `Synthetic` data set. Simulated data allows for a more comprehensive evaluation than real data, as we know the treatment effect for test instances. The data generating process is inspired by an e-commerce setting and similar to existing work [24]. First, we generate customer characteristics as follows: $X \sim \mathcal{N}(0,1)^d$. Then, we generate a sale (or conversion) probability S based on these characteristics and random coefficients $U_s \sim \mathcal{U}(-1,1)^d$ as $S = \frac{1}{1+\exp\left(-\sum_d U_s X + \epsilon_s\right)}$, where $\epsilon_s \sim \mathcal{N}(0, 0.1)$. Similarly, we generate a potential revenue R using random coefficients $U_r \sim \mathcal{U}(-1,1)^d$ as $R = 1 + |\sum_d U_r X| + \epsilon_r$, where $\epsilon_r \sim \mathcal{N}(0, 0.1)$. The cost of the treatment C (the marketing incentive or discount) is defined as 10% of the revenue: $C = 0.1R$. The observed outcome Y, the net revenue generated for that customer, is the revenue for that customer minus the treatment cost–simulated as follows:

$$y_i = \begin{cases} r_i - c_i & \text{if } t_i = 1 \text{ and } s_i = 1 \\ -c_i & \text{if } t_i = 1 \text{ and } s_i = 0 \\ r_i & \text{if } t_i = 0 \text{ and } s_i = 1 \\ 0 & \text{if } t_i = 0 \text{ and } s_i = 0. \end{cases}$$

We generate $10,000$ instances with $d = 10$ characteristics.

Next, we also compare with three real-world data sets. The `Criteo` data set [5] is the result of a randomized trial testing whether showing an advertisement increases a customer's visit or conversion probability. For the outcome, we follow [34] and take the net revenue y as conversion minus visit. To reduce training times, we randomly sample $500,000$ instances from this data set.

The `Hillstrom` data set was collected to test whether an e-mail campaign resulted in additional sales. Two treatments were recorded: a `Men`'s and `Women`'s e-mail. Therefore, we split the data in two data sets for both treatments, and use the same control group for both. We calculate each customer's net revenue as revenue (conversion times spend) minus visit.

Finally, data from a `Promotion` campaign at a large online travel agency was used as a randomized dataset to evaluate the offline performance of different approaches in a real-world setting.

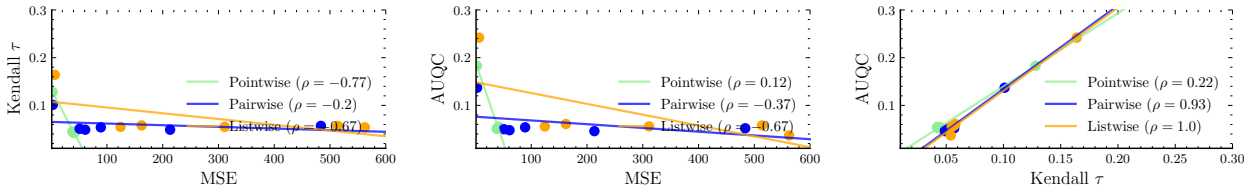
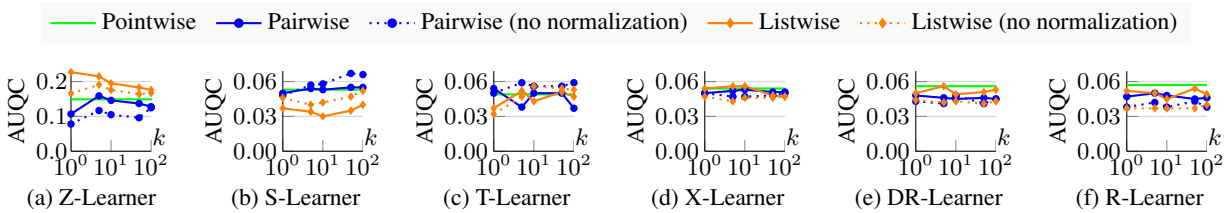


Figure 3: *Analyzing Performance Trade-offs on Synthetic Data.* We compare the three different objectives (point-, pair-, and listwise) across metalearners. Using the `Synthetic` data set, we compare performance in terms of MSE (i.e., pointwise accuracy), Kendall τ (i.e., pairwise rank correlation), and AUQC (i.e., listwise decision quality). For each, we show the correlation ρ.



(a) Z-Learner (b) S-Learner (c) T-Learner (d) X-Learner (e) DR-Learner (f) R-Learner

Figure 4: *What Is the Effect of the Number of Sampling Iterations k?* We show performance in terms of AUQC for the different metalearners on the `Synthetic` data set. We fix the sigmoid parameter $\sigma = 1$ and train with default hyperparameters.

We compare the three objectives and metalearners for gradient boosting based on LightGBM [59]. We implement our proposed pairwise and listwise objectives as a custom objectives and metrics in LightGBM. For each data set, we run a five-fold cross-validation procedure. We run hyperparameter tuning using 10 random sampling iterations over the following hyperparameters: "`num_leaves`" $\in [10, 50]$, "`learning_rate`" $\in [0.01, 0.20]$, "`max_depth`" $\in [3, 10]$, and "`min_data_in_leaf`" $\in [10, 30]$. We use 64% of all instances for training, 16% for validation, and 20% for testing.

5.2 Comparing Performance for the Different Objectives and Metalearners (RQ1)

For each metalearner, we compare a model trained with only a pointwise objective to our proposed ranking alternatives, based on either pairwise and listwise objectives. We evaluate the quality of each treatment allocation policy by looking at the cumulative treatment effect over the instance ranking, measured using the AUQC presented above. We evaluate the performance for each metalearner and objective over the different data sets (see Figure 2 and Appendix B.1). Across data sets and metalearners, we observe that listwise metalearners generally result in better treatment prioritization. Over all tested data sets and metalearners, a pointwise objective gives the highest AUQC in only a minority of cases (7/30), while the listwise objective obtains best in class performance in a majority of time (16/30) and the pairwise objective performs similar to the pointwise (7/30). A listwise approach outperforms a pointwise one in a majority of cases (20/30). Only for the `Promotion` data, the pointwise objective performs relatively well.

Interestingly, we observe differences across metalearners in terms of which objective gives the best results. The listwise objective seems favorable for some metalearners (Z-, X-, and R-Learner), while the pairwise objective seems preferable for the S-Learner and the pointwise objective seems best for the DR-Learner. Generally, we also observe that the choice of metalearner is at least equally important as the choice of objective. This finding stresses the importance of testing different metalearners–illustrating the value of our contribution.

When learning a model using ranking objectives, the ranking scores are not properly calibrated. This represents a possible challenge for metalearners that do not directly learn the treatment effect, i.e., the S- and T-Learner. For these metalearners, an instance's predicted ranking score does not necessarily correspond to its potential outcome. Rather, if one instance's outcome is larger than another instance's ($y_i > y_j$), its predicted score will be larger ($\hat{y}_i > \hat{y}_j$). Although the ranking might hold for both potential outcomes, there are no guarantees for the rankings of the estimated treatment effects derived from these ranking scores (τ_i and τ_j). Because ranking scores are not calibrated, arithmetic operations of the scores (as used by these two metalearners) may not be meaningful. Nevertheless, the ranking versions of the S- and T-Learner perform relatively well in practice, illustrating that they may provide a meaningful heuristic. While calibration methods for ranking models exists, we leave this extension for future work (see conclusion).

5.3 Analyzing Alternative Metrics (RQ2) and Design Choices (RQ3)

This section aims to provide a deeper understanding of our proposed approach. To this end, we highlight additional results using the `Synthetic` data, which allows for a more comprehensive analysis as we know the ground truth treatment effects for the test instances.

To answer *(RQ2)* regarding performance trade-offs of the different objectives and metalearners, we can present additional metrics to allow for a more holistic evaluation (see Figure 3 and Appendix B.2). Our main metric of interest remains the AUQC: a listwise metric of ranking quality. Additionally, we present two other metrics: a pointwise error metric (MSE) and a pairwise rank correlation coefficient (Kendall τ). First, we observe that ranking objectives give far worse performance in terms of MSE (Appendix B.2). In other words, the ranking score does not accurately reflect the size of the treatment effect. In relation to this finding, Figure 3 shows that the MSE is not a good predictor of performance in terms of AUQC. Conversely, Kendall τ is a good predictor for AUQC for all models ($\rho > 0$), particularly for the ranking models ($\rho > 0.9$). This finding underscores the importance of ranking metrics for evaluating decision quality and highlights the irrelevance of optimizing pointwise error for treatment prioritization.

Finally, to answer *(RQ3)*, we analyze several design choices of our proposed ranking metalearners in Figure 4 and Appendix B.3. We observe that the default setting used in the experiments above (sampling iterations $k = 1$, sigmoid $\sigma = 1$, normalizing ranking scores) generally performs well across ranking objectives and metalearners. Interestingly, we observe little performance benefits when training with more sampling iterations. This finding shows that our sampled objective can estimate the ranking objective accurately. Only for the S- and T-Learner without normalization, we observe better performance for a higher k. Additionally, the stochasticity of our sampled objectives may even provide a form of regularization [60].

6 Conclusion

This work addressed the problem of optimally prioritizing instances for treatment, an important problem for many applications where not all instances can receive a treatment. Existing approaches typically tackle this problem in a prediction-focused approach by first obtaining a pointwise *effect estimate* for each instance's treatment effect, and then ranking instances based on these estimates. Conversely, we explore an alternative, decision-focused approach: using objectives that learn to *rank the treatment effects*, we aim to optimize the quality of the resulting treatment policy directly. Building on the literature on learning to rank, we propose pairwise and listwise ranking objectives and show that our proposed listwise objective directly optimizes the policy's AUQC. Moreover, we propose different ranking metalearners by integrating these ranking objectives in the construction of each metalearner. Empirical results show that our proposed *effect ranking* approach can outperform a pointwise, *effect estimation* approach. In conclusion, our proposed ranking metalearners offer a valuable tool for applications where instances need to be prioritized for treatment.

Our work opens up several exciting directions for future work. First, by building upon advances in learning to rank, such as more advanced listwise objectives [e.g., 50] or calibration of ranking scores [e.g., 61, 62]. Alternatively, we envision extensions of our approach to more complicated settings with, e.g., more advanced operational constraints [63], multiple or continuous treatments, or more complex objectives (e.g., incremental return on investment). Additionally, it would be insightful to analyze how our proposed ranking metalearners perform when learning from confounded observational data with non-random treatment assignments. Finally, while theoretical results regarding convergence or error bounds are out of the scope of this work, we believe that extending the results obtainedfor the effect estimation models [e.g., 56, 55] to effect ranking models is a fruitful area of future work.

References

[1] Javier Albert and Dmitri Goldenberg. E-commerce promotions personalization via online multiple-choice knapsack with uplift modeling. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2863–2872, 2022.

[2] Alex Savachkin and Andrés Uribe. Dynamic redistribution of mitigation resources during influenza pandemics. *Socio-Economic Planning Sciences*, 46(1):33–45, 2012.

[3] Toon Vanderschueren, Robert Boute, Tim Verdonck, Bart Baesens, and Wouter Verbeke. Optimizing the preventive maintenance frequency with causal machine learning. *International Journal of Production Economics*, 258:108798, 2023.

[4] Debopam Bhattacharya and Pascaline Dupas. Inferring welfare maximizing treatment assignment under budget constraints. *Journal of Econometrics*, 167(1):168–196, 2012.

[5] Eustache Diemert, Artem Betlei, Christophe Renaudin, and Massih-Reza Amini. A large scale benchmark for uplift modeling. In *KDD*, 2018.

[6] Dmitri Goldenberg, Javier Albert, Lucas Bernardi, and Pablo Estevez. Free lunch! retrospective uplift modeling for dynamic promotions recommendation within roi constraints. In *Proceedings of the 14th ACM Conference on Recommender Systems*, pages 486–491, 2020.

[7] Vasilis Syrgkanis, Greg Lewis, Miruna Oprescu, Maggie Hei, Keith Battocchi, Eleanor Dillon, Jing Pan, Yifeng Wu, Paul Lo, Huigang Chen, et al. Causal inference and machine learning in practice with econml and causalml: Industrial use cases at microsoft, tripadvisor, uber. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 4072–4073, 2021.

[8] Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. *arXiv preprint arXiv:2307.13565*, 2023.

[9] Artem Betlei, Eustache Diemert, and Massih-Reza Amini. Uplift modeling with generalization guarantees. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 55–65, 2021.

[10] Weijia Zhang, Jiuyong Li, and Lin Liu. A unified survey of treatment effect heterogeneity modelling and uplift modelling. *ACM Computing Surveys (CSUR)*, 54(8):1–36, 2021.

[11] Floris Devriendt, Darie Moldovan, and Wouter Verbeke. A literature survey and experimental evaluation of the state-of-the-art in uplift modeling: A stepping stone toward the development of prescriptive analytics. *Big data*, 6(1):13–41, 2018.

[12] Diego Olaya, Kristof Coussement, and Wouter Verbeke. A survey and benchmarking study of multitreatment uplift modeling. *Data Mining and Knowledge Discovery*, 34:273–308, 2020.

[13] Artem Betlei, Eustache Diemert, and Massih-Reza Amini. Uplift prediction with dependent feature representation in imbalanced treatment and control conditions. In *Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13–16, 2018, Proceedings, Part V 25*, pages 47–57. Springer, 2018.

[14] Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences*, 116(10):4156–4165, 2019.

[15] Alicia Curth and Mihaela van der Schaar. Nonparametric estimation of heterogeneous treatment effects: From theory to learning algorithms. In *International Conference on Artificial Intelligence and Statistics*, pages 1810–1818. PMLR, 2021.

[16] Kathleen Kane, Victor SY Lo, and Jane Zheng. Mining for the truly responsive customers and prospects using true-lift modeling: Comparison of new and existing methods. *Journal of Marketing Analytics*, 2:218–238, 2014.

[17] Piotr Rzepakowski and Szymon Jaroszewicz. Decision trees for uplift modeling with single and multiple treatments. *Knowledge and Information Systems*, 32:303–327, 2012.

[18] Michał Sołtys, Szymon Jaroszewicz, and Piotr Rzepakowski. Ensemble methods for uplift modeling. *Data mining and knowledge discovery*, 29:1531–1559, 2015.

[19] Leo Guelman, Montserrat Guillén, and Ana M Pérez-Marín. Uplift random forests. *Cybernetics and Systems*, 46(3-4):230–248, 2015.

[20] Wouter Verbeke, Diego Olaya, Jeroen Berrevoets, Sam Verboven, and Sebastián Maldonado. The foundations of cost-sensitive causal classification. *arXiv preprint arXiv:2007.12582*, 2020.

[21] Floris Devriendt, Jeroen Berrevoets, and Wouter Verbeke. Why you should stop predicting customer churn and start using uplift models. *Information Sciences*, 548:497–515, 2021.

[22] Robin M Gubela and Stefan Lessmann. Uplift modeling with value-driven evaluation metrics. *Decision support systems*, 150:113648, 2021.

[23] Wouter Verbeke, Diego Olaya, Marie-Anne Guerry, and Jente Van Belle. To do or not to do? cost-sensitive causal classification with individual treatment effect estimates. *European Journal of Operational Research*, 305(2):838–852, 2023.

[24] Hugo Manuel Proença and Felipe Moraes. Incremental profit per conversion: a response transformation for uplift modeling in e-commerce promotions. *arXiv preprint arXiv:2306.13759*, 2023.

[25] Kui Zhao, Junhao Hua, Ling Yan, Qi Zhang, Huan Xu, and Cheng Yang. A unified framework for marketing budget allocation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1820–1830, 2019.

[26] Ye Tu, Kinjal Basu, Cyrus DiCiccio, Romil Bansal, Preetam Nandy, Padmini Jaikumar, and Shaunak Chatterjee. Personalized treatment selection using causal heterogeneity. In *Proceedings of the Web Conference 2021*, pages 1574–1585, 2021.

[27] Meng Ai, Biao Li, Heyang Gong, Qingwei Yu, Shengjie Xue, Yuan Zhang, Yunzhou Zhang, and Peng Jiang. Lbcf: A large-scale budget-constrained causal forest algorithm. In *Proceedings of the ACM Web Conference 2022*, pages 2310–2319, 2022.

[28] Carlos Fernández-Loría and Jorge Loría. Learning the ranking of causal effects with confounded data. *arXiv preprint arXiv:2206.12532*, 2022.

[29] Susan Athey, Niall Keleher, and Jann Spiess. Machine learning who to nudge: causal vs predictive targeting in a field experiment on student financial aid renewal. *arXiv preprint arXiv:2310.08672*, 2023.

[30] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136, 2007.

[31] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

[32] Shuyang Du, James Lee, and Farzin Ghaffarizadeh. Improve user retention with causal learning. In *The 2019 ACM SIGKDD Workshop on Causal Discovery*, pages 34–49. PMLR, 2019.

[33] Will Y Zou, Shuyang Du, James Lee, and Jan Pedersen. Heterogeneous causal learning for effectiveness optimization in user marketing. *arXiv preprint arXiv:2004.09702*, 2020.

[34] Hao Zhou, Shaoming Li, Guibin Jiang, Jiaqi Zheng, and Dong Wang. Direct heterogeneous causal learning for resource allocation problems in marketing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 5446–5454, 2023.

[35] Ziang Yan, Shusen Wang, Guorui Zhou, Jingjian Lin, and Peng Jiang. An end-to-end framework for marketing effectiveness optimization under budget constraint. *arXiv preprint arXiv:2302.04477*, 2023.

[36] Floris Devriendt, Jente Van Belle, Tias Guns, and Wouter Verbeke. Learning to rank for uplift modeling. *IEEE Transactions on Knowledge and Data Engineering*, 34(10):4888–4904, 2020.

[37] Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 1097–1104, 2011.

[38] Baqun Zhang, Anastasios A Tsiatis, Marie Davidian, Min Zhang, and Eric Laber. Estimating optimal treatment regimes from a classification perspective. *Stat*, 1(1):103–114, 2012.

[39] Eric B Laber and Ying-Qi Zhao. Tree-based methods for individualized treatment regimes. *Biometrika*, 102(3):501–514, 2015.

[40] Alexander R Luedtke and Mark J van der Laan. Super-learning of an optimal dynamic treatment rule. *The international journal of biostatistics*, 12(1):305–332, 2016.

[41] Toru Kitagawa and Aleksey Tetenov. Who should be treated? empirical welfare maximization methods for treatment choice. *Econometrica*, 86(2):591–616, 2018.

[42] Susan Athey and Stefan Wager. Policy learning with observational data. *Econometrica*, 89(1):133–161, 2021.

[43] Donald B Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

[44] Pierre Gutierrez and Jean-Yves Gérardy. Causal inference and uplift modelling: A review of the literature. In *International conference on predictive applications and APIs*, pages 1–13. PMLR, 2017.

[45] Erik Sverdrup, Han Wu, Susan Athey, and Stefan Wager. Qini curves for multi-armed treatment rules. *arXiv preprint arXiv:2306.11979*, 2023.

[46] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96, 2005.

[47] Christopher JC Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11(23-581):81, 2010.

[48] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, and Tie-Yan Liu. A theoretical analysis of ndcg type ranking measures. In *Conference on learning theory*, pages 25–54. PMLR, 2013.

[49] Pinar Donmez, Krysta M Svore, and Christopher JC Burges. On the local optimality of lambdarank. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 460–467, 2009.

[50] Ivan Lyzhin, Aleksei Ustimenko, Andrey Gulin, and Liudmila Prokhorenkova. Which tricks are important for learning to rank? In *International Conference on Machine Learning*, pages 23264–23278. PMLR, 2023.

[51] Paul W Holland. Statistics and causal inference. *Journal of the American statistical Association*, 81(396):945–960, 1986.

[52] Maciej Jaskowski and Szymon Jaroszewicz. Uplift modeling for clinical trial data. In *ICML workshop on clinical data analysis*, volume 46, pages 79–95, 2012.

[53] Krzysztof Rudaś and Szymon Jaroszewicz. Linear regression for uplift modeling. *Data Mining and Knowledge Discovery*, 32:1275–1305, 2018.

[54] Heejung Bang and James M Robins. Doubly robust estimation in missing data and causal inference models. *Biometrics*, 61(4):962–973, 2005.

[55] Edward H Kennedy. Towards optimal doubly robust estimation of heterogeneous causal effects. *Electronic Journal of Statistics*, 17(2):3008–3049, 2023.

[56] Xinkun Nie and Stefan Wager. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108(2):299–319, 2021.

[57] Peter M Robinson. Root-n-consistent semiparametric regression. *Econometrica: Journal of the Econometric Society*, pages 931–954, 1988.

[58] Kevin Hillstrom. The MineThatData E-Mail Analytics And Data Mining Challenge — blog.minethatdata.com. `https://blog.minethatdata.com/2008/03/minethatdata-e-mail-analytics-and-data.html`, 2008. [Accessed 15-04-2024].

[59] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[60] Jerome H Friedman. Stochastic gradient boosting. *Computational statistics & data analysis*, 38(4):367–378, 2002.

[61] David Sculley. Combined regression and ranking. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 979–988, 2010.

[62] Le Yan, Zhen Qin, Xuanhui Wang, Michael Bendersky, and Marc Najork. Scale calibration of deep ranking models. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4300–4309, 2022.

[63] Toon Vanderschueren, Bart Baesens, Tim Verdonck, and Wouter Verbeke. A new perspective on classification: optimally allocating limited resources to uncertain tasks. *Decision Support Systems*, 179:114151, 2024.

[64] Daniel G Horvitz and Donovan J Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685, 1952.

[65] Miguel A Hernán and James M Robins. Estimating causal effects from epidemiological data. *Journal of Epidemiology & Community Health*, 60(7):578–586, 2006.

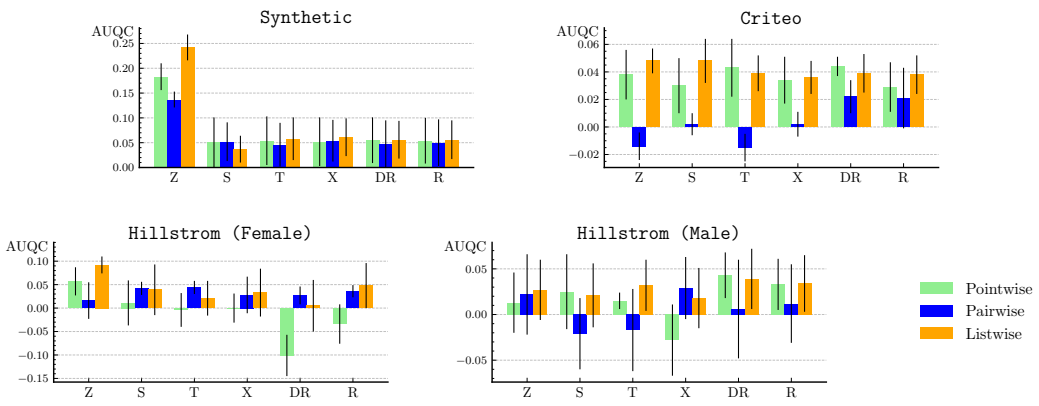


Figure 5: *Ranking Quality for Different Objectives and Metalearners.* For each metalearner, we compare three different objectives: point-, pair-, and listwise. We show performance in terms of AUQC $\pm$ one standard error, for five different data sets. As opposed to the figure in the main body, we do not scale the results here. Due to confidentiality reasons, we cannot share the raw results for the `Promotion` data set.

A Problem formulation: Identifiability Assumptions

As introduced in the main body, we require the standard assumptions from causal inference to identify the causal effect. In this work, we assumed that historical data comes from a randomized controlled trial:

Assumption 2 (Consistency)**.** *When $Y = y$ and $T = t$, we assume that $Y(T = t) = y$. This implicates that, for each instance, when given treatment t, the outcome we observe is the potential outcome associated to that treatment $Y(t)$.*

Assumption 3 (No interference)**.** *An instance's outcome given a treatment is independent of treatments administered to other instances: $Y_i(t_0, \ldots, t_i, \ldots, t_n) = Y_i(t_i)$.*

Assumption 4 (Unconfoundedness)**.** *We assume $Y(T) \perp\!\!\!\perp T$, i.e., past treatment decisions were made at random, i.e., not based on the instance's characteristics.*

If we do not have data from a randomized trial, we require a stronger assumption called strong ignorability or no hidden confounding: $Y(T) \perp\!\!\!\perp T|x$, i.e., past treatment decisions were exclusively based on the instance's observed characteristics x. In this case, we also require positivity: for each instance, the probability of administering each treatment has to be larger than zero, i.e., $P(T|x) > 0$. We do not consider this scenario in our work. However, our ranking metalearners can easily be extended to these scenarios: while some metalearners already integrate the propensity score in their construction, others may be improved by inverse propensity score weighting [64, 65].

B Empirical Results: Additional Experiments

This section presents additional experimental results. We describe additional results for RQ1 in Appendix B.1, and for RQ2 and RQ3 in Appendix B.2 where we analyze the effect of several design choices and hyperparameters.

B.1 Comparing Performance for the Different Objectives and Metalearners (RQ1): Additional Results

We display the results presented in the main body of the text without normalizing the best value to 1 in Figure 5 and provide them in table format in Table 4.

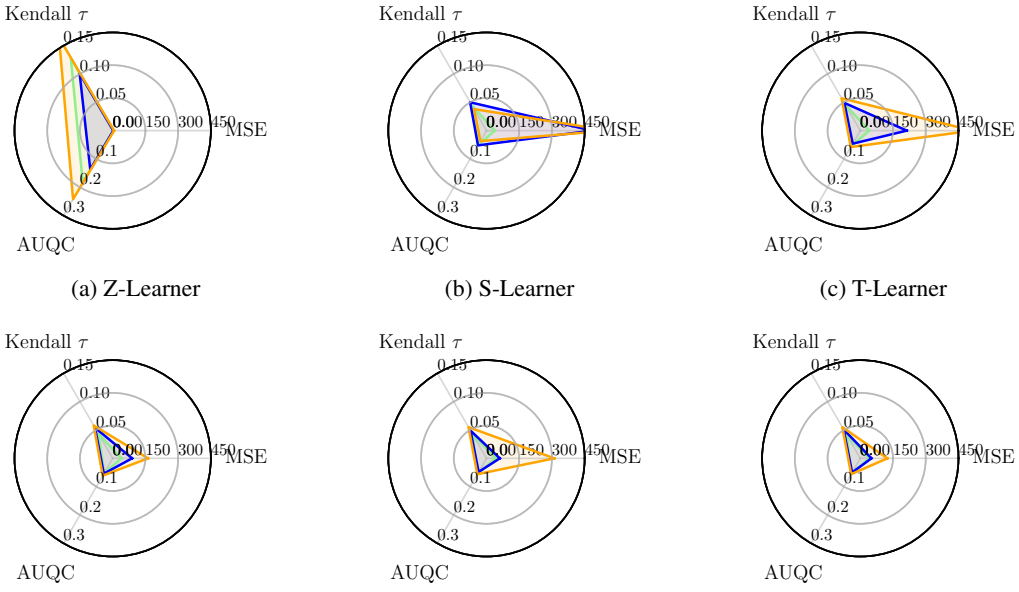
B.2 Analyzing Alternative Metrics (RQ2) and Design Choices (RQ3): Additional Results

First, we present additional results to support our investigation surrounding RQ2. For each metalearner, we visualize the trade-off between different metrics: the MSE (i.e., pointwise accuracy), Kendall τ (i.e., pairwise rank correlation), and AUQC (i.e., listwise ranking quality). This shows that models that perform well in terms of AUQC, also perform well in terms of Kendall τ. Conversely, performance in terms of MSE does not seem related to AUQC or Kendall τ. Finally, we also display these results in table format in Table 5.

Meta	Objective			Data				
	Point	*Pair*	*List*	`Synthetic`	`Criteo`	`Hillstrom|♀`	`Hillstrom|♂`	`Promotion*`
Z	✓	✗	✗	0.183 (0.027)	0.038 (0.018)	0.057 (0.030)	0.013 (0.033)	0.407 (0.111)
	✗	✓	✗	0.137 (0.016)	−0.014 (0.010)	0.016 (0.039)	0.022 (0.044)	0.370 (0.296)
	✗	✗	✓	**0.242** (0.026)	**0.048** (0.009)	**0.092** (0.018)	**0.027** (0.033)	**0.593** (0.037)
S	✓	✗	✗	0.051 (0.050)	0.030 (0.020)	0.011 (0.048)	**0.025** (0.041)	0.370 (0.074)
	✗	✓	✗	**0.052** (0.039)	0.002 (0.008)	**0.042** (0.014)	−0.021 (0.039)	**0.593** (0.296)
	✗	✗	✓	0.037 (0.027)	**0.048** (0.016)	0.039 (0.054)	0.021 (0.035)	0.222 (0.148)
T	✓	✗	✗	0.054 (0.049)	**0.043** (0.021)	−0.004 (0.036)	0.015 (0.009)	0.333 (0.148)
	✗	✓	✗	0.046 (0.044)	−0.015 (0.010)	**0.044** (0.014)	−0.017 (0.045)	**0.630** (0.185)
	✗	✗	✓	**0.058** (0.043)	0.039 (0.013)	0.021 (0.037)	**0.032** (0.028)	0.556 (0.222)
X	✓	✗	✗	0.052 (0.049)	0.034 (0.017)	-0.000 (0.031)	−0.028 (0.039)	**1.000** (0.296)
	✗	✓	✗	0.054 (0.042)	0.002 (0.009)	0.028 (0.039)	**0.029** (0.034)	0.704 (0.259)
	✗	✗	✓	**0.061** (0.038)	**0.036** (0.012)	**0.033** (0.051)	0.018 (0.033)	0.852 (0.111)
DR	✓	✗	✗	0.055 (0.046)	**0.044** (0.007)	−0.101 (0.044)	**0.043** (0.025)	**0.667** (0.111)
	✗	✓	✗	0.048 (0.047)	0.022 (0.012)	**0.027** (0.019)	0.006 (0.054)	−0.481 (0.222)
	✗	✗	✓	**0.056** (0.038)	0.039 (0.014)	0.005 (0.055)	0.039 (0.033)	−0.148 (0.222)
R	✓	✗	✗	0.054 (0.046)	0.029 (0.018)	−0.034 (0.042)	0.033 (0.028)	**0.593** (0.222)
	✗	✓	✗	0.050 (0.047)	0.021 (0.022)	0.036 (0.013)	0.012 (0.043)	−0.296 (0.111)
	✗	✗	✓	**0.056** (0.039)	**0.038** (0.014)	**0.048** (0.048)	**0.034** (0.031)	−0.259 (0.259)

*For the `Promotion` data, we only present scaled results such that the best AUQC = 1 due to reasons of confidentiality.

Table 4: *Ranking Quality for Different Objectives and Metalearners.* For each metalearner, we compare three different objectives: point-, pair-, and listwise. We show performance in terms of AUQC (with standard error in brackets), for the `Synthetic`, `Criteo`, and `Hillstrom` (`Women` (♀) and `Men` (♂) e-mail), and `Promotion` data.



(a) Z-Learner (b) S-Learner (c) T-Learner

(d) X-Learner (e) DR-Learner (f) R-Learner

Figure 6: *Analyzing Performance Trade-offs on Synthetic Data.* For each metalearner, we compare three different objectives: pointwise, pairwise, and listwise. Using the `Synthetic` data set, we compare performance in terms of MSE (measuring pointwise accuracy), Kendall τ (measuring pairwise rank correlation), and AUQC (measuring global, listwise decision quality).

Meta	Objective			Metric		
	Point	*Pair*	*List*	MSE	Kendall τ	AUQC
Z	✓	✗	✗	**2.474** (0.255)	0.128 (0.021)	0.183 (0.027)
	✗	✓	✗	2.867 (0.263)	0.101 (0.013)	0.137 (0.016)
	✗	✗	✓	6.465 (1.260)	**0.164** (0.020)	**0.242** (0.026)
S	✓	✗	✗	**37.916** (3.557)	0.044 (0.033)	0.051 (0.050)
	✗	✓	✗	483.712 (159.054)	**0.050** (0.028)	**0.052** (0.039)
	✗	✗	✓	562.509 (300.946)	0.038 (0.022)	0.037 (0.027)
T	✓	✗	✗	**41.014** (3.852)	0.045 (0.033)	0.054 (0.049)
	✗	✓	✗	213.265 (30.716)	0.049 (0.030)	0.046 (0.044)
	✗	✗	✓	513.908 (94.592)	**0.057** (0.030)	**0.058** (0.043)
X	✓	✗	✗	**40.417** (3.697)	0.043 (0.033)	0.052 (0.049)
	✗	✓	✗	88.987 (16.651)	0.054 (0.030)	0.054 (0.042)
	✗	✗	✓	162.146 (40.750)	**0.058** (0.028)	**0.061** (0.038)
DR	✓	✗	✗	**42.007** (4.017)	0.043 (0.031)	0.055 (0.046)
	✗	✓	✗	61.014 (10.268)	0.049 (0.032)	0.048 (0.047)
	✗	✗	✓	311.236 (105.607)	**0.055** (0.028)	**0.056** (0.038)
R	✓	✗	✗	**41.263** (4.760)	0.042 (0.031)	0.054 (0.046)
	✗	✓	✗	51.008 (12.094)	0.051 (0.032)	0.050 (0.047)
	✗	✗	✓	124.400 (18.588)	**0.055** (0.028)	**0.056** (0.039)

Table 5: *Analyzing Performance Trade-offs on Synthetic Data.* For each metalearner, we compare three different objectives: pointwise, pairwise, and listwise. Using the `Synthetic` data set, we compare performance in terms of MSE (measuring pointwise accuracy), Kendall τ (measuring pairwise rank correlation), and AUQC (measuring global, listwise decision quality). For each, we show the standard error in brackets.

B.3 Sensitivity Analysis

To single out the effect of the analyzed design choice, we train with default hyperparameters for each training objective and metalearner. We explore three hyperparameters: (1) the number of sampling iterations k, i.e., the number of sampled pairs per instance, (2) the sigmoid parameter σ controlling the steepness of the comparison in the construction of the pairwise score (see Equation (3)), and (3) whether we normalize the ranking score by feeding it to a logistic sigmoid and constraining it to $[0, 1]$. Related work has shown that this normalization might effectively serve as a form of regularization and help with overfitting [32].

In Figure 7, we vary the number of sampled pairs k in the ranking objectives, for each ranking metalearner, and compare it to the pointwise model on the `Synthetic` data. We also show results for the metalearners with normalization of the score (i.e., constraining the model to outputs between zero and one) and without normalization. Somewhat surprisingly, with normalization, we see that increasing k does not yield better results. Conversely, without normalization, increasing k does in fact improve performance for most metalearners. Nevertheless, for the Z-, X-, DR-, and R-Learner, the best performance is achieved with normalization and $k = 1$–the same settings used in the experiments in the main body (i.e., Table 4). For the S- and T-Learner, deviating from these settings might improve performance. Overall, we see that for each metalearner and objective, we can obtain the same performance or better than the pointwise equivalent given that the right hyperparameters are chosen. This insight provides another validation of our proposed approach.

In Figure 8, we vary the sigmoid parametere σ, controlling the steepness of the comparison of the instance scores in the construction of the pairwise score (see Equation (3)). Generally, we obtain good performance for smaller values ($\sigma \leq 1$). When using normalization, our method seems more sensitive to this hyperparameter compared to training without score normalization. Although there may be some benefit of tuning this hyperparameter, we observe that fixing the sigmoid parameter at $\sigma = 1$ seems like a good choice overall–this was the setting used to generate the experimental results in the main body.

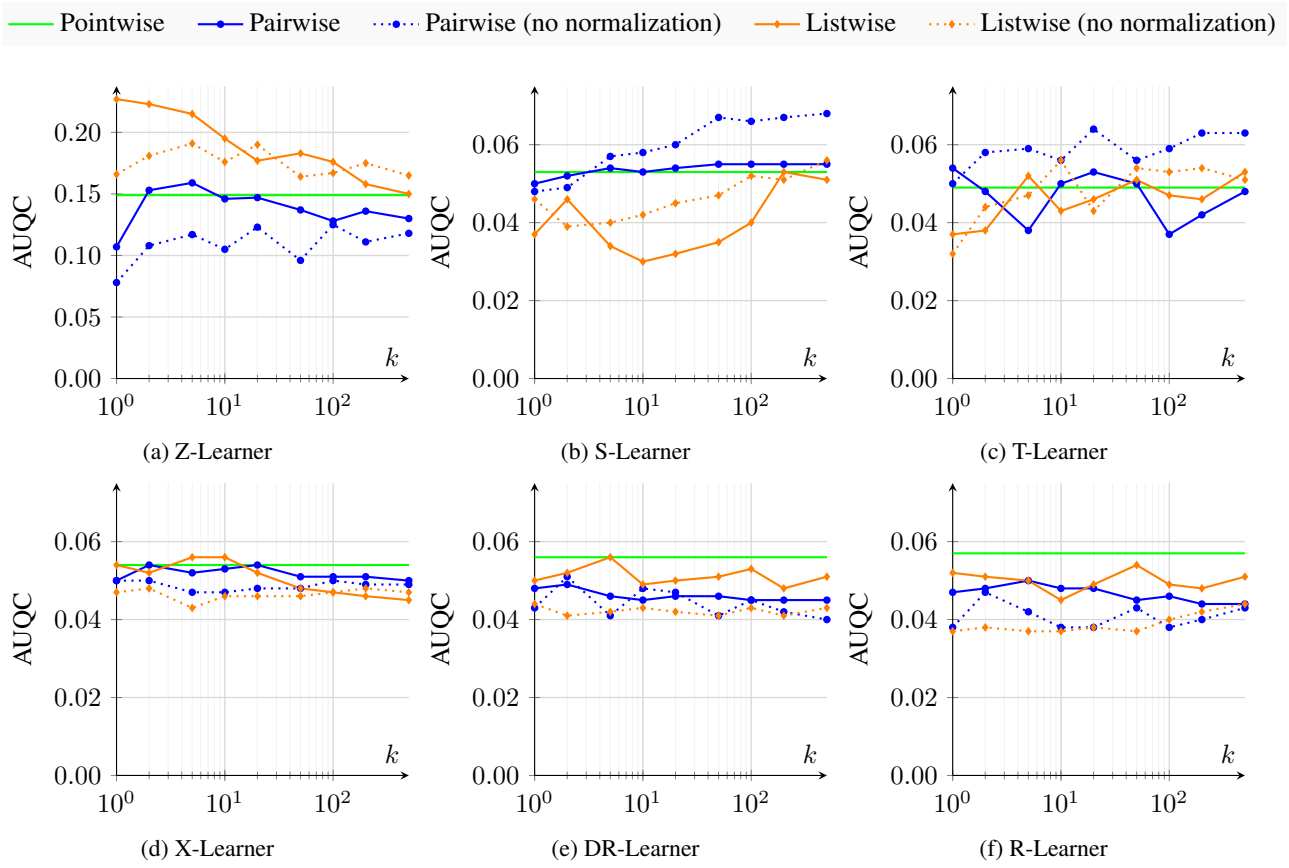


Figure 7: *What Is the Effect of the Number of Sampling Iterations k?* We show performance in terms of AUQC (higher is better) for the different metalearners on the `Synthetic` data set. We fix the sigmoid parameter $\sigma = 1$ and train with default hyperparameters.

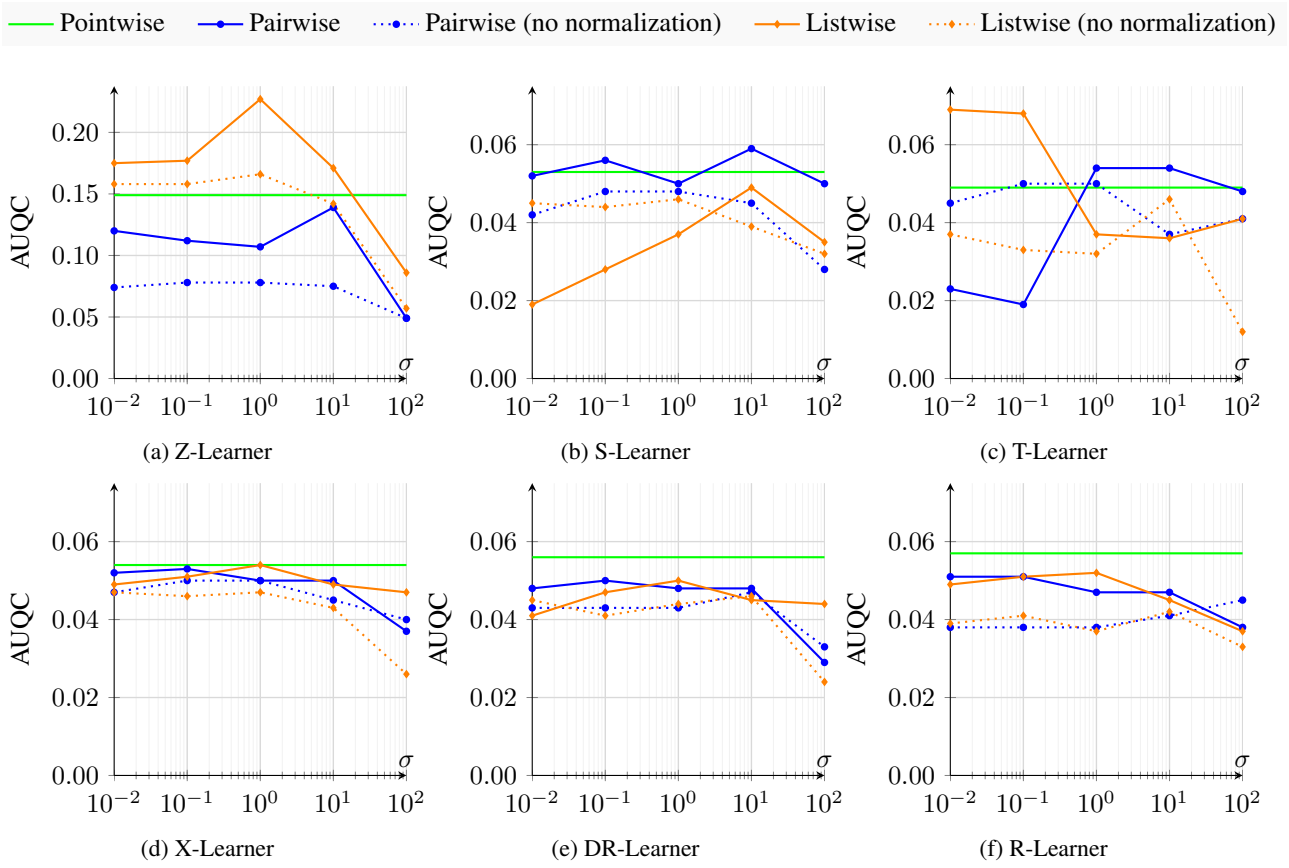


Figure 8: *What Is the Effect of the Sigmoid Parameter σ?* We show performance in terms of AUQC (higher is better) for the different metalearners on the `Synthetic` data set. We fix the number of sampling iterations $k = 1$ and train with default hyperparameters.