Endogenous Macrodynamics in Algorithmic Recourse

Altmeyer, Patrick; Giovan, Angela; Buszydlik, Aleksander ; Dobiczek, Karol ; van Deursen, Arie; Liem, Cynthia C. S.

DOI
10.1109/SaTML54575.2023.00036

Publication date
2023

Document Version
Final published version

Published in
Proceedings of the 2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)

Endogenous Macrodynamics in Algorithmic Recourse

Patrick Altmeyer
Delft University of Technology
Faculty of Electrical Engineering
Mathematics and Computer Science
2628 XE Delft, The Netherlands
P.Altmeyer@tudelft.nl

Giovan Angela
Delft University of Technology
Faculty of Electrical Engineering
Mathematics and Computer Science
2628 XE Delft, The Netherlands
G.J.A.Angela@student.tudelft.nl

Aleksander Buszydlik
Delft University of Technology
Faculty of Electrical Engineering
Mathematics and Computer Science
2628 XE Delft, The Netherlands
A.J.Buszydlik@student.tudelft.nl

Karol Dobiczek
Delft University of Technology
Faculty of Electrical Engineering
Mathematics and Computer Science
2628 XE Delft, The Netherlands
K.T.Dobiczek@student.tudelft.nl

Arie van Deursen
Delft University of Technology
Faculty of Electrical Engineering
Mathematics and Computer Science
2628 XE Delft, The Netherlands
Arie.vanDeursen@tudelft.nl

Cynthia C. S. Liem
Delft University of Technology
Faculty of Electrical Engineering
Mathematics and Computer Science
2628 XE Delft, The Netherlands
C.C.S.Liem@tudelft.nl

2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML) | 978-1-6654-6299-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/SATML54575.2023.00036

***Abstract*—Existing work on Counterfactual Explanations (CE) and Algorithmic Recourse (AR) has largely focused on single individuals in a static environment: given some estimated model, the goal is to find valid counterfactuals for an individual instance that fulfill various desiderata. The ability of such counterfactuals to handle dynamics like data and model drift remains a largely unexplored research challenge. There has also been surprisingly little work on the related question of how the actual implementation of recourse by one individual may affect other individuals. Through this work, we aim to close that gap. We first show that many of the existing methodologies can be collectively described by a generalized framework. We then argue that the existing framework does not account for a hidden external cost of recourse, that only reveals itself when studying the endogenous dynamics of recourse at the group level. Through simulation experiments involving various state-of-the-art counterfactual generators and several benchmark datasets, we generate large numbers of counterfactuals and study the resulting domain and model shifts. We find that the induced shifts are substantial enough to likely impede the applicability of Algorithmic Recourse in some situations. Fortunately, we find various strategies to mitigate these concerns. Our simulation framework for studying recourse dynamics is fast and open-sourced.**

***Index Terms*—Algorithmic Recourse; Counterfactual Explanations; Explainable AI; Dynamic Systems**

I. Introduction

Recent advances in Artificial Intelligence (AI) have propelled its adoption in scientific domains outside of Computer Science including Healthcare, Bioinformatics, Genetics and the Social Sciences. While this has in many cases brought benefits in terms of efficiency, state-of-the-art models like Deep Neural Networks (DNN) have also given rise to a new type of problem in the context of data-driven decision-making. They are essentially **black boxes**: so complex, opaque and underspecified in the data that it is often impossible to understand how they actually arrive at their decisions without auxiliary tools. Despite this shortcoming, black-box models have grown in popularity in recent years and have at times created undesirable societal outcomes [1]. The scientific community has tackled this issue from two different angles: while some have appealed for a strict focus on inherently interpretable models [2], others have investigated different ways to explain the behaviour of black-box models. These two sub-domains can be broadly referred to as **interpretable AI** and **explainable AI** (XAI), respectively.

Among the approaches to XAI that have recently grown in popularity are **Counterfactual Explanations** (CE). They explain how inputs into a model need to change for it to produce different outputs. Counterfactual Explanations that involve realistic and actionable changes can be used for the purpose of **Algorithmic Recourse** (AR) to help individuals who face adverse outcomes. An example relevant to the Social Sciences is consumer credit: in this context, AR can be used to guide individuals in improving their creditworthiness, should they have previously been denied access to credit based on an automated decision-making system. A meaningful recourse recommendation for a denied applicant could be: *"If your net savings rate had been 10% of your monthly income instead of the actual 8%, your application would have been successful. See if you can temporarily cut down on consumption."* In the remainder of this paper, we will use both terminologies—recourse and counterfactual—interchangeably to refer to situations where counterfactuals are generated with the intent to provide individual recourse.

Existing work in this field has largely worked in a static setting: various approaches have been proposed to generate counterfactuals for a given individual that is subject to some pre-trained model. More recent work has compared different

978-1-6654-6299-0/23/$31.00 ©2023 IEEE
DOI 10.1109/SaTML54575.2023.00036

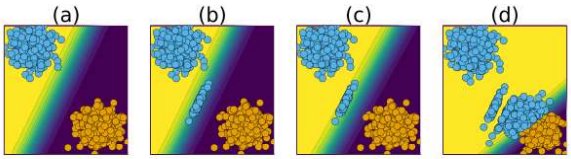

Fig. 1. Dynamics in Algorithmic Recourse: (a) we have a simple linear classifier trained for binary classification where samples from the negative class ($y = 0$) are marked in orange and samples of the positive class ($y = 1$) are marked in blue; (b) the implementation of AR for a random subset of individuals leads to a noticeable domain shift; (c) as the classifier is retrained we observe a corresponding model shift; (d) as this process is repeated, the decision boundary moves away from the target class.

approaches within this static setting [3]. In this work, we go one step further and ask ourselves: what happens if recourse is provided and implemented repeatedly? What types of dynamics are introduced and how do different counterfactual generators compare in this context?

Research on Algorithmic Recourse has also so far typically addressed the issue from the perspective of a single individual. Arguably though, most real-world applications that warrant AR involve potentially large groups of individuals typically competing for scarce resources. Our work demonstrates that in such scenarios, choices made by or for a single individual are likely to affect the broader collective of individuals in ways that many current approaches to AR fail to account for. More specifically, we argue that a strict focus on minimizing the private costs to individuals may be too narrow an objective.

Figure 1 illustrates this idea for a binary problem involving a linear classifier and the counterfactual generator proposed by Wachter et al. [4]: the implementation of AR for a subset of individuals immediately leads to a visible domain shift in the (blue) target class (b), which in turn triggers a model shift (c). As this game of implementing AR and updating the classifier is repeated, the decision boundary moves away from training samples that were originally in the target class (d). We refer to these types of dynamics as **endogenous** because they are induced by the implementation of recourse itself. The term **macrodynamics** is borrowed from the economics literature and used to describe processes involving whole groups or societies.

We think that these types of endogenous dynamics may be problematic and deserve our attention. From a purely technical perspective, we note the following: firstly, model shifts may inadvertently change classification outcomes for individuals who never received recourse. Secondly, we observe in Figure 1 that as the decision boundary moves in the direction of the non-target class, counterfactual paths become shorter. We think that in some practical applications, this can be expected to generate costs for involved stakeholders. To follow our argument, consider the following two examples:

Example I.1 (Consumer Credit)**.** Suppose Figure 1 relates to an automated decision-making system used by a retail bank to evaluate credit applicants with respect to their creditworthiness. Assume that the two features are meaningful in the sense that creditworthiness decreases in the bottom-right direction. Then we can think of the outcome in panel (d) as representing a situation where the bank supplies credit to more borrowers (blue), but these borrowers are on average less creditworthy and more of them can be expected to default on their loan. This represents a cost to the retail bank.

Example I.2 (Student Admission)**.** Suppose Figure 1 relates to an automated decision-making system used by a university in its student admission process. Assume that the two features are meaningful in the sense that the likelihood of students completing their degree decreases in the bottom-right direction. Then we can think of the outcome in panel (b) as representing a situation where more students are admitted to university (blue), but they are more likely to fail their degree than students that were admitted in previous years. The university admission committee catches on to this and suspends its efforts to offer Algorithmic Recourse. This represents an opportunity cost to future student applicants, that may have derived utility from being offered recourse.

Both examples are exaggerated simplifications of potential real-world scenarios, but they serve to illustrate the point that recourse for one single individual may exert negative externalities on other individuals.

To the best of our knowledge, this is the first work investigating endogenous macrodynamics in AR. Our contributions to the state of knowledge are as follows: firstly, we posit a compelling argument that calls for a novel perspective on Algorithmic Recourse extending our focus from single individuals to groups (Sections II and III). Secondly, we introduce an experimental framework extending previous work by Altmeyer [5], which enables us to study macrodynamics of Algorithmic Recourse through simulations that can be fully parallelized (Section IV). Thirdly, we use this framework to provide a first in-depth analysis of endogenous recourse dynamics induced by various popular counterfactual generators proposed in [4], [6], [7], [8] and [9] (Sections V and VI). Fourthly, given that we find a substantial impact of recourse, we propose and assess various mitigation strategies (Section VII). Finally, we discuss our findings in the broader context of the literature in Section VIII, before pointing to some of the limitations of our work as well as avenues for future research in Section IX. Section X concludes.

II. Background

In this section, we provide a review of the relevant literature. First, Subsection II-A discusses the existing research within the domain of Counterfactual Explanations and Algorithmic Recourse. Then, Subsection II-B presents some of the previous work on the measurement of data and model shifts.

A. Algorithmic Recourse

A framework for Counterfactual Explanations was first proposed in 2017 by Wachter et al. [4] and has served as the baseline for many methodologies that have been proposed since then. Let $M : \mathcal{X} \mapsto \mathcal{Y}$ denote some pre-trained model

that maps from inputs $X \in \mathcal{X}$ to outputs $Y \in \mathcal{Y}$. Then we are interested in minimizing the cost[1] $C = \text{cost}(x')$ incurred by individual x when moving to a counterfactual state x' such that the predicted outcome $M(x')$ corresponds to some target outcome y^*:

$$\min_{x' \in \mathcal{X}} \text{cost}(x') \quad \text{s. t.} \quad M(x') = y^* \tag{1}$$

For implementation purposes, (1) is typically approximated through regularization:

$$x' = \arg\min_{x'} \text{yloss}(M(x'), y^*) + \lambda \text{cost}(x') \tag{2}$$

In the baseline work [4], the cost function is proxied by some distance metric based on the simple intuition that perturbations of x are costly to the individual. For models that are differentiable and produce smooth predictions, (2) can be solved through gradient descent. This summarizes the approach followed in [4] which we refer to simply as **Wachter**, the name of the first author, in the remainder of this paper.

Many approaches for the generation of Algorithmic Recourse have been described in the literature since 2017. An October 2020 survey by Karimi et al. laid out 60 algorithms that have been proposed since 2014 [10]. Another survey published around the same time by Verma et al. described 29 algorithms [11]. Different approaches vary primarily in terms of the objective functions they impose, how they optimize said objective (from brute force through gradient-based approaches to graph traversal algorithms), and how they ensure that certain requirements for CE are met. Regarding the latter, the literature has produced an extensive list of desiderata each addressing different needs. To name but a few, we are interested in generating counterfactuals that are close [4], actionable [12], realistic [6], sparse, diverse [8] and if possible causally founded [13].

Efforts so far have largely been directed at improving the quality of Counterfactual Explanations within a static context: given some pre-trained classifier $M : \mathcal{X} \mapsto \mathcal{Y}$, we are interested in generating one or multiple meaningful Counterfactual Explanations for some individual characterized by x. The ability of Counterfactual Explanations to handle dynamics like data and model shifts remains a largely unexplored research challenge at this point [11]. We have been able to identify only one recent work by Upadhyay et al. that considers the implications of **exogenous** domain and model shifts in the context of AR [14]. Exogenous shifts are strictly of external origin. For example, they might stem from data correction, temporal shifts or geospatial changes [14]. Upadhyay et al. [14] propose ROAR: a framework for Algorithmic Recourse that evidently improves robustness to such exogenous shifts.

As mentioned earlier, research has so far also generally focused on generating counterfactuals for single individuals or instances. We have been able to identify only one existing work that investigates black-box model behaviour towards a group of individuals [15]. The authors propose an optimization framework that generates collective counterfactuals. We provide a motivation for doing so from the perspective of endogenous macrodynamics of Algorithmic Recourse.

[1]Equivalently, others have referred to this quantity as *complexity* or simply *distance*.

B. Domain and Model Shifts

Much attention has been paid to the detection of dataset shifts – situations where the distribution of data changes over time. Rabanser et al. suggest a framework to detect data drift from a minimal number of samples through the application of two-sample tests [16]. This task is a generalization of the anomaly detection problem for large datasets, which aims to answer the question if two sets of samples could have been generated from the same probability distribution. Numerous approaches to anomaly detection have been summarized [17]. Another well-established research topic is concept drift: situations where external variables influence the patterns between the input and the output of a model [18]. For instance, Gama et al. offer a review of the adaptive learning techniques which can handle concept drift [19]. Less previous work is available on the related topic of model drift: changes in model performance over time. Nelson et al. review how resistant different machine learning models are to model drift [20]. Ackerman et al. offer a method to detect changes in model performance when ground truth is not available [21].

In the context of Algorithmic Recourse, domain and model shifts were first brought up by the authors behind ROAR [14]. In their work, they refer to model shifts as simply any perturbation Δ to the parameters of the model in question: M. While this also sets the baseline for our analysis here, it is worth noting that in [14] these perturbations are mechanically introduced. In contrast, we are interested in quantifying model shifts that arise endogenously as part of a dynamic recourse process. In addition to quantifying the magnitude of shifts Δ, we aim to also analyse the characteristics of changes to the model, such as the position of the decision boundary and the overall decisiveness of the model. We have not been able to identify previous work on this topic.

C. Benchmarking Counterfactual Generators

Despite the large and growing number of approaches to counterfactual search, there have been surprisingly few benchmark studies that compare different methodologies. This may be partially due to limited software availability in this space. Recent work has started to address this gap: firstly, [22] run a large benchmarking study using different algorithmic approaches and numerous tabular datasets; secondly, [3] introduce a Python framework—CARLA—that can be used to apply and benchmark different methodologies; finally, `CounterfactualExplanations.jl` [5] provides an extensible and fast implementation in Julia. Since the experiments presented here involve extensive simulations, we have relied on and extended the Julia implementation due to the associated performance benefits. In particular, we have built a framework on top of `CounterfactualExplanations.jl` that extends

the functionality from static benchmarks to simulation experiments: `AlgorithmicRecourseDynamics.jl`[2]. The core concepts implemented in that package reflect what is presented in Section IV of this paper.

III. Gradient-Based Recourse Revisited

In this section, we first set out a generalized framework for gradient-based counterfactual search that encapsulates the various Individual Recourse methods we have chosen to use in our experiments (Section III-A). We then introduce the notion of a hidden external cost in Algorithmic Recourse and extend the existing framework to explicitly address this cost in the counterfactual search objective (Section III-B).

A. From Individual Recourse ...

We have chosen to focus on gradient-based counterfactual search for two reasons: firstly, they can be seen as direct descendants of our baseline method (Wachter); secondly, gradient-based search is particularly well-suited for differentiable black-box models like deep neural networks, which we focus on in this work. In particular, we include the following generators in our simulation experiments below: **REVISE** [7], **CLUE** [9], **DiCE** [8] and a greedy approach that relies on probabilistic models [6]. Our motivation for including these different generators in our analysis is that they all offer slightly different approaches to generating meaningful counterfactuals for differentiable black-box models. We hypothesize that generating more **meaningful** counterfactuals should mitigate the endogenous dynamics illustrated in Figure 1 in Section I. This intuition stems from the underlying idea that more meaningful counterfactuals are generated by the same or at least a very similar data-generating process as the observed data. All else equal, counterfactuals that fulfil this basic requirement should be less prone to trigger shifts.

As we will see next, all of them can be described by the following generalized form of Equation (3):

$$\mathbf{s}' = \arg\min_{\mathbf{s}' \in \mathcal{S}} \{\text{yloss}(M(f(\mathbf{s}')), y^*) + \lambda \text{cost}(f(\mathbf{s}'))\} \tag{3}$$

Here $\mathbf{s}' = \{s'_k\}_K$ is a K-dimensional array of counterfactual states and $f : \mathcal{S} \mapsto \mathcal{X}$ maps from the counterfactual state space to the feature space. In Wachter, the state space is the feature space: f is the identity function and the number of counterfactuals K is one. Both REVISE and CLUE search counterfactuals in some latent space $\mathcal{S}$ instead of the feature space directly. The latent embedding is learned by a separate generative model that is tasked with learning the data-generating process (DGP) of X. In this case, f in Equation (3) corresponds to the decoder part of the generative model, that is the function that maps back from the latent space to inputs. Provided the generative model is well-specified, traversing the latent embedding typically yields meaningful counterfactuals since they are implicitly generated by the (learned) DGP [7].

[2]The code has been released as a package: https://github.com/pat-alt/AlgorithmicRecourseDynamics.jl.

CLUE distinguishes itself from REVISE and other counterfactual generators in that it aims to minimize the predictive uncertainty of the model in question, M. To quantify predictive uncertainty, Antoran et al. [9] rely on entropy estimates for probabilistic models. The greedy approach proposed by Schut et al. [6], which we refer to as **Greedy**, also works with the subclass of models $\tilde{\mathcal{M}} \subset \mathcal{M}$ that can produce predictive uncertainty estimates. The authors show that in this setting the cost function $\text{cost}(\cdot)$ in Equation (3) is redundant and meaningful counterfactuals can be generated in a fast and efficient manner through a modified Jacobian-based Saliency Map Attack (JSMA). Schut et al. [6] also show that by maximizing the predicted probability of x' being assigned to target class y^*, we also implicitly minimize predictive entropy (as in CLUE). In that sense, CLUE can be seen as equivalent to REVISE in the Bayesian context and we shall therefore refer to both approaches collectively as **Latent Space** generators[3].

Finally, DiCE [8] distinguishes itself from all other generators considered here in that it aims to generate a diverse set of $K > 1$ counterfactuals. Wachter et al. [4] show that diverse outcomes can in principle be achieved simply by rerunning counterfactual search multiple times using stochastic gradient descent (or by randomly initializing the counterfactual)[4]. In [8] diversity is explicitly proxied via Determinantal Point Processes (DDP): the authors introduce DDP as a component of the cost function $\text{cost}(\mathbf{s}')$ and thereby produce counterfactuals $s_1, ..., s_K$ that look as different from each other as possible. The implementation of DiCE in our library of choice—`CounterfactualExplanations.jl`—uses that exact approach. It is worth noting that for $k = 1$, DiCE reduces to Wachter since the DDP is constant and therefore does not affect the objective function in Equation (3).

B. ... towards Collective Recourse

All of the different approaches introduced above tackle the problem of Algorithmic Recourse from the perspective of one single individual[5]. To explicitly address the issue that Individual Recourse may affect the outcome and prospect of other individuals, we propose to extend Equation (3) as follows:

$$\begin{aligned}\mathbf{s}' = \arg\min_{\mathbf{s}' \in \mathcal{S}} \{&\text{yloss}(M(f(\mathbf{s}')), y^*) \\ &+ \lambda_1 \text{cost}(f(\mathbf{s}')) + \lambda_2 \text{extcost}(f(\mathbf{s}'))\}\end{aligned} \tag{4}$$

Here $\text{cost}(f(\mathbf{s}'))$ denotes the proxy for private costs faced by the individual as before and λ_1 governs to what extent

[3]In fact, there are several other recently proposed approaches to counterfactual search that also broadly fall in this same category. They largely differ with respect to the chosen generative model: for example, the generator proposed by Dombrowski et al. [23] relies on normalizing flows.

[4]Note that (3) naturally lends itself to that idea: setting K to some value greater than one and using the Wachter objective essentially boils down to computing multiple counterfactuals in parallel. Here, $yloss(\cdot)$ is first broadcasted over elements of $\mathbf{s}'$ and then aggregated. This is exactly how counterfactual search is implemented in `CounterfactualExplanations.jl`.

[5]DiCE recognizes that different individuals may have different objective functions, but it does not address the interdependencies between different individuals.

that private cost ought to be penalized. The newly introduced term extcost$(f(\mathbf{s}'))$ is meant to capture and address external costs incurred by the collective of individuals in response to changes in $\mathbf{s}'$. The underlying concept of private and external costs is borrowed from Economics and well-established in that field: when the decisions or actions by some individual market participant generate external costs, then the market is said to suffer from negative externalities and is considered inefficient [24]. We think that this concept describes the endogenous dynamics of algorithmic recourse observed here very well. As with Individual Recourse, the exact choice of extcost$(\cdot)$ is not obvious, nor do we intend to provide a definitive answer in this work, if such even exists. That being said, we do propose a few potential mitigation strategies in Section VII.

IV. Modelling Endogenous Macrodynamics in Algorithmic Recourse

In the following, we describe the framework we propose for modelling and analyzing endogenous macrodynamics in Algorithmic Recourse. We introduce this framework with the ambition to shed light on the following research questions:

Research Question IV.1 (Endogenous Shifts)**.** *Does the repeated implementation of recourse provided by state-of-the-art generators lead to shifts in the domain and model?*

Research Question IV.2 (Costs)**.** *If so, are these dynamics substantial enough to be considered costly to stakeholders involved in real-world automated decision-making processes?*

Research Question IV.3 (Heterogeneity)**.** *Do different counterfactual generators yield significantly different outcomes in this context? Furthermore, is there any heterogeneity concerning the chosen classifier and dataset?*

Research Question IV.4 (Drivers)**.** *What are the drivers of endogenous dynamics in Algorithmic Recourse?*

Below we first describe the basic simulations that were generated to produce the findings in this work and also constitute the core of `AlgorithmicRecourseDynamics.jl`—the Julia package we introduced earlier. The remainder of this section then introduces various evaluation metrics that can be used to benchmark different counterfactual generators with respect to how they perform in the dynamic setting.

A. Simulations

The dynamics illustrated in Figure 1 were generated through a simple experiment that aims to simulate the process of Algorithmic Recourse in practice. We begin in the static setting at time $t = 0$: firstly, we have some binary classifier M that was pre-trained on data $\mathcal{D} = \mathcal{D}_0 \cup \mathcal{D}_1$, where $\mathcal{D}_0$ and $\mathcal{D}_1$ denote samples in the non-target and target class, respectively; secondly, we generate recourse for a random batch of B individuals in the non-target class ($\mathcal{D}_0$). Note that we focus our attention on classification problems since classification poses the most common use-case for recourse[6].

In order to simulate the dynamic process, we suppose that the model M is retrained following the actual implementation of recourse in time $t = 0$. Following the update to the model, we assume that at time $t = 1$ recourse is generated for yet another random subset of individuals in the non-target class. This process is repeated for a number of time periods T. To get a clean read on endogenous dynamics we keep the total population of samples closed: we allow existing samples to move from factual to counterfactual states but do not allow any entirely new samples to enter the population. The experimental setup is summarized in Algorithm 1.

Algorithm 1 Simulation Experiment

1: **procedure** Experiment$(M, \mathcal{D}, G)$
2: $E \leftarrow \emptyset$ ▷ Initialize evaluation E.
3: $t \leftarrow 0$
4: **while** $t < T$ **do**
5: batch $\subset \mathcal{D}_0$ ▷ Sample from $\mathcal{D}_0$ (assignment).
6: batch $\leftarrow G($batch$)$ ▷ Generate counterfactuals.
7: $M \leftarrow M(\mathcal{D})$ ▷ Retrain model.
8: $E \leftarrow$ eval$(M, \mathcal{D}) \cup E$ ▷ Update evaluation.
9: $t \leftarrow t + 1$ ▷ Increment t.
10: **end while**
11: **return** $E, M, \mathcal{D}$
12: **end procedure**

Note that the operation in line 4 is an assignment, rather than a copy operation, so any updates to 'batch' will also affect $\mathcal{D}$. The function eval$(M, \mathcal{D})$ loosely denotes the computation of various evaluation metrics introduced below. In practice, these metrics can also be computed at regular intervals as opposed to every round.

Along with any other fixed parameters affecting the counterfactual search, the parameters T and B are assumed as given in Algorithm 1. Still, it is worth noting that the higher these values, the more factual instances undergo recourse throughout the entire experiment. Of course, this is likely to lead to more pronounced domain and model shifts by time T. In our experiments, we choose the values such that the majority of the negative instances from the initial dataset receive recourse. As we compute evaluation metrics at regular intervals throughout the procedure, we can also verify the impact of recourse when it is implemented for a smaller number of individuals.

Algorithm 1 summarizes the proposed simulation experiment for a given dataset $\mathcal{D}$, model M and generator G, but naturally, we are interested in comparing simulation outcomes for different sources of data, models and generators. The framework we have built facilitates this, making use of multi-threading in order to speed up computations. Holding the initial model and dataset constant, the experiments are run for

[6]To keep notation simple, we have also restricted ourselves to binary classification here, but `AlgorithmicRecourseDynamics.jl` can also be used for multi-class problems.

all generators, since our primary concern is to benchmark different recourse methods. To ensure that each generator is faced with the same initial conditions in each round t, the candidate batch of individuals from the non-target class is randomly drawn from the intersection of all non-target class individuals across all experiments $\{\text{EXPERIMENT}(M, \mathcal{D}, G)\}_{j=1}^{J}$ where J is the total number of generators.

B. Evaluation Metrics

We formulate two desiderata for the set of metrics used to measure domain and model shifts induced by recourse. First, the metrics should be applicable regardless of the dataset or classification technique so that they allow for the meaningful comparison of the generators in various scenarios. As knowledge of the underlying probability distribution is rarely available, the metrics should be empirical and non-parametric. This further ensures that we can also measure large datasets by sampling from the available data. Moreover, while our study was conducted in a two-class classification setting, our choice of metrics should remain applicable in future research on multi-class recourse problems. Second, the set of metrics should allow capturing various aspects of the previously mentioned magnitude, path, and pace of changes while remaining as small as possible.

1) Domain Shifts: To quantify the magnitude of domain shifts we rely on an unbiased estimate of the squared population **Maximum Mean Discrepancy (MMD)** given as:

$$\begin{aligned} MMD(X', \tilde{X}') = &\frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} k(x_i, x_j) \\ &+ \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} k(\tilde{x}_i, \tilde{x}_j) \\ &- \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(x_i, \tilde{x}_j) \end{aligned} \tag{5}$$

where $X = \{x_1, ..., x_m\}$, $\tilde{X} = \{\tilde{x}_1, ..., \tilde{x}_n\}$ represent independent and identically distributed samples drawn from probability distributions $\mathcal{X}$ and $\tilde{\mathcal{X}}$ respectively [25]. MMD is a measure of the distance between the kernel mean embeddings of $\mathcal{X}$ and $\tilde{\mathcal{X}}$ in a Reproducing Kernel Hilbert Space, $\mathcal{H}$ [26]. An important consideration is the choice of the kernel function $k(\cdot, \cdot)$. In our implementation, we make use of a Gaussian kernel with a constant length-scale parameter of 0.5. As the Gaussian kernel captures all moments of distributions $\mathcal{X}$ and $\tilde{\mathcal{X}}$, we have that $MMD(X, \tilde{X}) = 0$ if and only if $X = \tilde{X}$. Conversely, larger values $MMD(X, \tilde{X}) > 0$ indicate that it is more likely that $\mathcal{X}$ and $\tilde{\mathcal{X}}$ are different distributions. In our context, large values, therefore, indicate that a domain shift indeed seems to have occurred.

To assess the statistical significance of the observed shifts under the null hypothesis that samples X and $\tilde{X}$ were drawn from the same probability distribution, we follow [27]. To that end, we combine the two samples and generate a large number of permutations of $X + \tilde{X}$. Then, we split the permuted data into two new samples X' and $\tilde{X}'$ having the same size as the original samples. Under the null hypothesis, we should have that $MMD(X', \tilde{X}')$ be approximately equal to $MMD(X, \tilde{X})$. The corresponding p-value can then be calculated by counting how often these two quantities are not equal.

We calculate the MMD for both classes individually based on the ground truth labels, i.e. the labels that samples were assigned in time $t = 0$. Throughout our experiments, we generally do not expect the distribution of the negative class to change over time – application of recourse reduces the size of this class, but since individuals are sampled uniformly the distribution should remain unaffected. Conversely, unless a recourse generator can perfectly replicate the original probability distribution, we expect the MMD of the positive class to increase. Thus, when discussing MMD, we generally mean the shift in the distribution of the positive class.

2) Model Shifts: As our baseline for quantifying model shifts, we measure perturbations to the model parameters at each point in time t following [14]. We define $\Delta = ||\theta_{t+1} - \theta_t||^2$, that is the euclidean distance between the vectors of parameters before and after retraining the model M. We shall refer to this baseline metric simply as **Perturbations**.

We extend the metric in Equation (5) to quantify model shifts. Specifically, we introduce **Predicted Probability MMD (PP MMD)**: instead of applying Equation (5) to features directly, we apply it to the predicted probabilities assigned to a set of samples by the model M. If the model shifts, the probabilities assigned to each sample will change; again, this metric will equal 0 only if the two classifiers are the same. We compute PP MMD in two ways: firstly, we compute it over samples drawn uniformly from the dataset, and, secondly, we compute it over points spanning a mesh grid over a subspace of the entire feature space. For the latter approach, we bound the subspace by the extrema of each feature. While this approach is theoretically more robust, unfortunately, it suffers from the curse of dimensionality, since it becomes increasingly difficult to select enough points to overcome noise as the dimension D grows.

As an alternative to PP MMD, we use a pseudo-distance for the **Disagreement Coefficient** (Disagreement). This metric was introduced in [28] and estimates $p(M(x) \neq M'(x))$, that is the probability that two classifiers disagree on the predicted outcome for a randomly chosen sample. Thus, it is not relevant whether the classification is correct according to the ground truth, but only whether the sample lies on the same side of the two respective decision boundaries. In our context, this metric quantifies the overlap between the initial model (trained before the application of AR) and the updated model. A Disagreement Coefficient unequal to zero is indicative of a model shift. The opposite is not true: even if the Disagreement Coefficient is equal to zero, a model shift may still have occurred. This is one reason why PP MMD is our preferred metric.

We further introduce **Decisiveness** as a metric that quantifies the likelihood that a model assigns a high probability to its classification of any given sample. We define the metric simply

as $\frac{1}{N}\sum_{i=0}^{N}(\sigma(M(x))-0.5)^2$ where $M(x)$ are predicted logits from a binary classifier and σ denotes the sigmoid function. This metric provides an unbiased estimate of the binary classifier's tendency to produce high-confidence predictions in either one of the two classes. Although the exact values for this metric are not important for our study, they can be used to detect model shifts. If decisiveness changes over time, then this is indicative of the decision boundary moving towards either one of the two classes. A potential caveat of this metric in the context of our experiments is that it will to some degree get inflated simply through retraining the model.

Finally, we also take a look at the out-of-sample **Performance** of our models. To this end, we compute their F-score on a test sample that we leave untouched throughout the experiment.

V. Experiment Setup

This section presents the exact ingredients and parameter choices describing the simulation experiments we ran to produce the findings presented in the next section (VI). For convenience, we use Algorithm 1 as a template to guide us through this section. A few high-level details upfront: each experiment is run for a total of $T = 50$ rounds, where in each round we provide recourse to five per cent of all individuals in the non-target class, so $B_t = 0.05 * N_t^{\mathcal{D}_0}$. All classifiers and generative models are retrained for 10 epochs in each round t of the experiment. Rather than retraining models from scratch, we initialize all parameters at their previous levels $(t-1)$ and backpropagate for 10 epochs using the new training data as inputs into the existing model. Evaluation metrics are computed and stored every 10 rounds. To account for noise, each individual experiment is repeated five times.[7]

A. $\mathcal{M}$—Classifiers and Generative Models

For each dataset and generator, we look at three different types of classifiers, all of them built and trained using `Flux.jl` [29]: firstly, a simple linear classifier**—Logistic Regression**—implemented as a single linear layer with sigmoid activation; secondly, a multilayer perceptron (**MLP**); and finally, a **Deep Ensemble** composed of five MLPs following [30] that serves as our only probabilistic classifier. We have chosen to work with deep ensembles both for their simplicity and effectiveness at modelling predictive uncertainty. They are also the model of choice in [6]. The network architectures are kept simple (top half of Table I), since we are only marginally concerned with achieving good initial classifier performance.

The Latent Space generator relies on a separate generative model. Following the authors of both REVISE and CLUE we use Variational Autoencoders (**VAE**) for this purpose. As with the classifiers, we deliberately choose to work with fairly simple architectures (bottom half of Table I). More expressive generative models generally also lead to more meaningful counterfactuals produced by Latent Space generators. But in our view, this should simply be considered as a vulnerability of counterfactual generators that rely on surrogate models to learn realistic representations of the underlying data.

[7]In the current implementation, we use the same train-test split each time to only account for stochasticity associated with randomly selecting individuals for recourse. An interesting alternative may be to also perform data splitting each time, thereby adding an additional layer of randomness.

TABLE I
Neural network architectures and training parameters.

	Data	Hidden Dim.	Latent Dim.	Hidden Layers	Batch	Dropout	Epochs
MLP							
	Synthetic	32	-	1	-	-	100
	Real-World	64	-	2	500	0.1	100
VAE							
	Synthetic	32	2	1	-	-	100
	Real-World	32	8	1	-	-	250

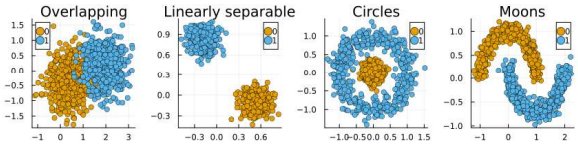

Fig. 2. Synthetic classification datasets used in our experiments. Samples from the negative class ($y = 0$) are marked in blue while samples of the positive class ($y = 1$) are marked in orange.

B. $\mathcal{D}$—Data

We have chosen to work with both synthetic and real-world datasets. Using synthetic data allows us to impose distributional properties that may affect the resulting recourse dynamics. Following [14], we generate synthetic data in $\mathbb{R}^2$ to also allow for a visual interpretation of the results. Real-world data is used in order to assess if endogenous dynamics also occur in higher-dimensional settings.

1) Synthetic data: We use four synthetic binary classification datasets consisting of 1000 samples each: **Overlapping**, **Linearly Separable**, **Circles** and **Moons** (Figure 2).

Ex-ante we expect to see that by construction, Wachter will create a new cluster of counterfactual instances in the proximity of the initial decision boundary as we saw in Figure 1. Thus, the choice of a black-box model may have an impact on the counterfactual paths. For generators that use latent space search (REVISE [7], CLUE [9]) or rely on (and have access to) probabilistic models (CLUE [9], Greedy [6]) we expect that counterfactuals will end up in regions of the target domain that are densely populated by training samples. Of course, this expectation hinges on how effective said probabilistic models are at capturing predictive uncertainty. Finally, we expect to see the counterfactuals generated by DiCE to be diversely spread around the feature space inside the target class[8]. In summary, we expect that the endogenous shifts induced by Wachter outsize those of all other generators since Wachter is not explicitly concerned with generating what we have defined as meaningful counterfactuals.

[8]As we mentioned earlier, the diversity constraint used by DiCE is only effective when at least two counterfactuals are being generated. We have therefore decided to always generate 5 counterfactuals for each generator and randomly pick one of them.

2) Real-world data: We use three different real-world datasets from the Finance and Economics domain, all of which are tabular and can be used for binary classification. Firstly, we use the **Give Me Some Credit** dataset which was open-sourced on Kaggle for the task to predict whether a borrower is likely to experience financial difficulties in the next two years [31], originally consisting of 250,000 instances with 11 numerical attributes. Secondly, we use the **UCI defaultCredit** dataset [32], a benchmark dataset that can be used to train binary classifiers to predict the binary outcome variable of whether credit card clients default on their payment. In its raw form, it consists of 23 explanatory variables: 4 categorical features relating to demographic attributes and 19 continuous features largely relating to individuals' payment histories and amount of credit outstanding. Both datasets have been used in the literature on AR before (see for example [3], [7] and [12]), presumably because they constitute real-world classification tasks involving individuals that compete for access to credit.

As a third dataset, we include the **California Housing** dataset derived from the 1990 U.S. census and sourced through scikit-learn [34]. It consists of 8 continuous features that can be used to predict the median house price for California districts. The continuous outcome variable is binarized as $\tilde{y} = \mathbb{I}_{y>\text{median}(Y)}$ indicating whether or not the median house price of a given district is above the median of all districts. While we have not seen this dataset used in the previous literature on AR, others have used the Boston Housing dataset in a similar fashion [6]. We initially also conducted experiments on that dataset, but eventually discarded it due to surrounding ethical concerns [35].

Since the simulations involve generating counterfactuals for a significant proportion of the entire sample of individuals, we have randomly undersampled each dataset to yield balanced subsamples consisting of 5,000 individuals each. We have also standardized all continuous explanatory features since our chosen classifiers are sensitive to scale.

C. G—Generators

All generators introduced earlier are included in the experiments: Wachter [4], REVISE [7], CLUE [9], DiCE [8] and Greedy [6]. In addition, we introduce two new generators in Section VII that directly address the issue of endogenous domain and model shifts. We also test to what extent it may be beneficial to combine ideas underlying the various generators.

VI. Experiments

Below, we first present our main experimental findings regarding these questions. We conclude this section with a brief recap providing answers to all of these questions.

A. Endogenous Macrodynamics

We start this section off with the key high-level observations. Across all datasets (synthetic and real), classifiers and counterfactual generators we observe either most or all of the following dynamics at varying degrees:

- Statistically significant domain and model shifts as measured by MMD.
- A deterioration in out-of-sample model performance as measured by the F-Score evaluated on a test sample. In many cases this drop in performance is substantial.
- Significant perturbations to the model parameters as well as an increase in the model's decisiveness.
- Disagreement between the original and retrained model, in some cases large.

There is also some clear heterogeneity across the results:

- The observed dynamics are generally of the highest magnitude for the linear classifier. Differences in results for the MLP and Deep Ensemble are mostly negligible.
- The reduction in model performance appears to be most severe when classes are not perfectly separable or the initial model performance was weak, to begin with.
- Except for the Greedy generator, all other generators generally perform somewhat better overall than the baseline (Wachter) as expected.

Focusing first on synthetic data, Figure 3 presents our findings for the dataset with overlapping classes. It shows the resulting values for some of our evaluation metrics at the end of the experiment, after all $T = 50$ rounds, along with error bars indicating the variation across folds.

The top row shows the estimated domain shifts. While it is difficult to interpret the exact magnitude of MMD, we can see that the values are different from zero and there is essentially no variation across our five folds. For the domain shifts, the Greedy generator induces the smallest shifts. In general, we have observed the opposite.

The second row shows the estimated model shifts, where here we have used the grid approach explained earlier. As with the domain shifts, the observed values are clearly different from zero and variation across folds is once again small. In this case, the results for this particular dataset very much reflect the broader patterns we have observed: Latent Space (LS) generators induce the smallest shifts, followed by DiCE, then Wachter and finally Greedy.

The same broad pattern also emerges in the third row: we observe the smallest deterioration in model performance for LS generators, albeit we still find a reduction in the F-Score of around 5-10 percentage points on average. Related to this, the bottom two rows indicate that the retrained classifiers disagree with their initial counterparts on the classification of up to nearly 25 per cent of the individuals. We also note that the final classifiers are more decisive, although as we noted earlier this may to some extent just be a byproduct of retraining the model throughout the experiment.

Figure 3 also indicates that the estimated effects are strongest for the simplest linear classifier, a pattern that we have observed fairly consistently. Conversely, there is virtually no difference in outcomes between the deep ensemble and the MLP. It is possible that the deep ensembles simply fail to capture predictive uncertainty well and hence counterfactual generators like Greedy, which explicitly addresses this quantity, fail to work as expected.

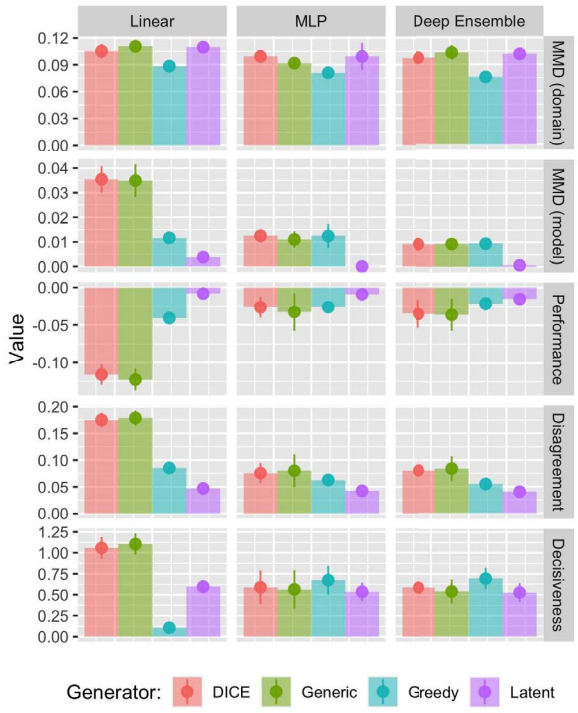

Fig. 3. Results for synthetic data with overlapping classes. The shown model MMD (PP MMD) was computed over a mesh grid of 1,000 points. Error bars indicate the standard deviation across folds.

The findings for the other synthetic datasets are broadly consistent with the observations above. For the Moons data, the same broad patterns emerge, although in this case, the Greedy generator induces comparably strong shifts in some cases. For the Circles data, model shifts and performance deterioration are quantitatively much smaller than what we can observe in Figure 3 and in many cases insignificant. For the Lincarly Separable data we also find substantial domain and model shifts, but almost no reduction in model performance.[9]

Finally, it is also worth noting that the observed dynamics and patterns are consistent throughout the experiment. That is to say that we start observing shifts already after just a few rounds and these tend to increase proportionately for the different generators over the course of the experiment.

Turning to the real-world data we will go through the findings presented in Figure 4, where each column corresponds to one of the three data sets. The results shown here are for the deep ensemble, which once again largely resemble those for the MLP. Starting from the top row, we find significant domain shifts of varying magnitudes. Latent Space search induces shifts that are orders of magnitude higher than for the other generators, which generally induce significant but small shifts.

Model shifts are shown in the middle row of Figure 4: the

[9]You can find a granular overview of all results including bootstraps in our online companion: https://www.paltmeyer.com/endogenous-macrodynamics-in-algorithmic-recourse/.

Fig. 4. Results for deep ensemble using real-world datasets. The shown model MMD (PP MMD) was computed over actual samples, rather than a mesh grid. Error bars indicate the standard deviation across folds.

estimated PP MMD is statistically significant across the board and in some cases much larger than in others. We find no evidence that LS search helps to mitigate model shifts, as we did before for the synthetic data. Since these real-world datasets are arguably more complex than the synthetic data, the generative model can be expected to have a harder time learning the data-generating process and hence this increased difficulty appears to affect the performance of REVISE/CLUE.

The out-of-sample model performance also deteriorates across the board and substantially so: the largest average reduction in F-Scores of more than 10 percentage points is observed for the Credit Default dataset. For this dataset we achieved the lowest initial model performance, indicating once again that weaker classifiers may be more exposed to endogenous dynamics. As with the synthetic data, the estimates for logistic regression are qualitatively in line with the above, but quantitatively even more pronounced.

To recap, we answer our research questions: firstly, endogenous dynamics do emerge in our experiments (RQ IV.1) and we find them substantial enough to be considered costly (RQ IV.2); secondly, the choice of the counterfactual generator matters, with Latent Space search generally having a dampening effect (RQ IV.3). The observed dynamics, therefore, seem to be driven by a discrepancy between counterfactual outcomes that minimize costs to individuals and outcomes that comply with the data-generating process (RQ IV.4).

VII. Mitigation Strategies and Experiments

Having established in the previous section that endogenous macrodynamics in AR are substantial enough to warrant our attention, in this section we ask ourselves:

Research Question VII.1 (Mitigation Strategies)**.** *What are potential mitigation strategies with respect to endogenous macrodynamics in AR?*

We propose and test several simple mitigation strategies. All of them essentially boil down to one simple principle: to avoid domain and model shifts, the generated counterfactuals should comply as much as possible with the true data-generating process. This principle is really at the core of Latent Space (LS) generators, and hence it is not surprising that we have found these types of generators to perform comparably well in the previous section. But as we have mentioned earlier, generators that rely on separate generative models carry an additional computational burden and, perhaps more importantly, their performance hinges on the performance of said generative models. Fortunately, it turns out that we can use a number of other, much simpler strategies.

A. More Conservative Decision Thresholds

The most obvious and trivial mitigation strategy is to simply choose a higher decision threshold γ. This threshold determines when a counterfactual should be considered valid. Under $\gamma = 0.5$, counterfactuals will end up near the decision boundary by construction. Since this is the region of maximal aleatoric uncertainty, the classifier is bound to be thrown off. By setting a more conservative threshold, we can avoid this issue to some extent. A drawback of this approach is that a classifier with high decisiveness may classify samples with high confidence even far away from the training data.

B. Classifier Preserving ROAR (ClaPROAR)

Another strategy draws inspiration from ROAR [14]: to preserve the classifier, we propose to explicitly penalize the loss it incurs when evaluated on the counterfactual x' at given parameter values. Recall that extcost$(\cdot)$ denotes what we had defined as the external cost in Equation (4). Formally, we let

$$\text{extcost}(f(\mathbf{s}')) = l(M(f(\mathbf{s}')), y') \tag{6}$$

for each counterfactual k where l denotes the loss function used to train M. This approach, which we refer to as **ClaPROAR**, is based on the intuition that (endogenous) model shifts will be triggered by counterfactuals that increase classifier loss. It is closely linked to the idea of choosing a higher decision threshold, but is likely better at avoiding the potential pitfalls associated with highly decisive classifiers. It also makes the private vs. external cost trade-off more explicit and hence manageable.

C. Gravitational Counterfactual Explanations

Yet another strategy extends Wachter as follows: instead of only penalizing the distance of the individuals' counterfactual to its factual, we propose penalizing its distance to some sensible point in the target domain, for example, the subsample average $\bar{x}^* = \text{mean}(x),\ x \in \mathcal{D}_1$:

$$\text{extcost}(f(\mathbf{s}')) = \text{dist}(f(\mathbf{s}'), \bar{x}^*) \tag{7}$$

Once again we can put this in the context of Equation (4): the former penalty can be thought of here as the private cost incurred by the individual, while the latter reflects the external cost incurred by other individuals. Higher choices of λ_2 relative to λ_1 will lead counterfactuals to gravitate towards the specified point $\bar{x}^*$ in the target domain. In the remainder of this paper, we will therefore refer to this approach as **Gravitational** generator, when we investigate its usefulness for mitigating endogenous macrodynamics[10].

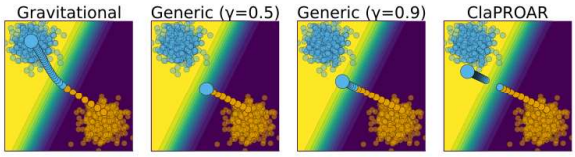

Fig. 5. Illustrative example demonstrating the properties of the various mitigation strategies. Samples from the negative class ($y = 0$) are marked in orange while samples of the positive class ($y = 1$) are marked in blue.

Figure 5 shows an illustrative example that demonstrates the differences in counterfactual outcomes when using the various mitigation strategies compared to the baseline approach, that is, Wachter with $\gamma = 0.5$: choosing a higher decision threshold pushes the counterfactual a little further into the target domain; this effect is even stronger for ClaPROAR; finally, using the Gravitational generator the counterfactual ends up all the way inside the target domain in the neighbourhood of $\bar{x}^*$[11]. Linking these ideas back to Example I.2, the mitigation strategies help ensure that the recommended recourse actions are substantial enough to truly lead to an increase in the probability that the admitted student eventually graduates.

Our findings indicate that all three mitigation strategies are at least at par with LS generators with respect to their effectiveness at mitigating domain and model shifts. Figure 6 presents a subset of the evaluation metrics for our synthetic data with overlapping classes. The top row in Figure 6 indicates that while domain shifts are of roughly the same magnitude for both Wachter and LS generators, our proposed strategies effectively mitigate these shifts. ClaPROAR appears to be particularly effective, which is positively surprising since it is designed to explicitly address model shifts, not domain shifts. As evident from the middle row in Figure 6 model shifts can also be reduced: for the deep ensemble LS search yields results that are at par with the mitigation strategies, while for both the simple MLP and logistic regression our simple strategies are more effective. The same overall pattern can be observed for out-of-sample model performance. Concerning the other synthetic datasets, for the Moons dataset, the emerging patterns are largely the same, but the estimated model shifts are insignificant as noted earlier; the same holds for the Circles dataset, but there is no significant reduction in model performance for our neural networks; in the case of

[10]Note that despite the naming conventions, our goal here is not to provide yet more counterfactual generators. Rather than looking at them as isolated entities, we believe and demonstrate that different approaches can be effectively combined.

[11]In order for the Gravitational generator and ClaPROAR to work as expected, one needs to ensure that counterfactual search continues, independent of the threshold probability γ.

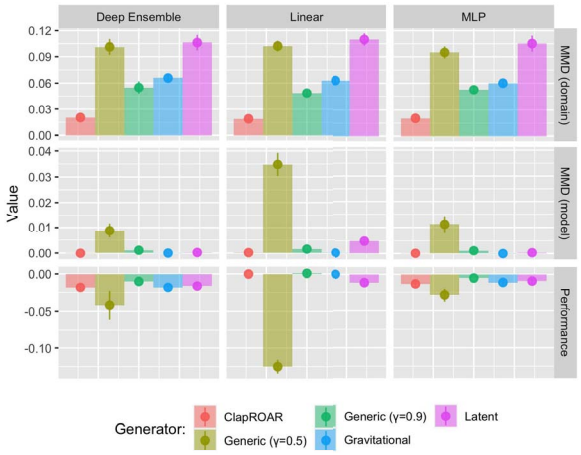

Fig. 6. The differences in counterfactual outcomes when using the various mitigation strategies compared to the baseline approach, that is Wachter with $\gamma = 0.5$. Results for synthetic data with overlapping classes. The shown model MMD (PP MMD) was computed over a mesh grid of points. Error bars indicate the standard deviation across folds.

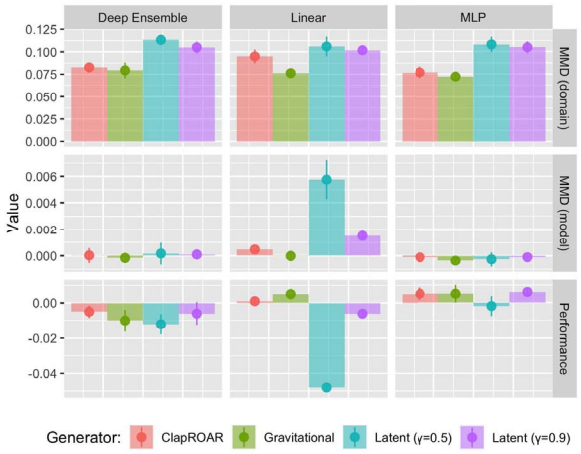

Fig. 7. Combining various mitigation strategies with LS search. Results for synthetic data with overlapping classes. The shown model MMD (PP MMD) was computed over a mesh grid of points. Error bars indicate the standard deviation across folds.

linearly separable data, we find the Gravitational generator to be most effective at mitigating shifts.

An interesting finding is also that the proposed strategies have a complementary effect when used in combination with LS generators. In experiments we conducted on the synthetic data, the benefits of LS generators were exacerbated further when using a more conservative threshold or combining it with the penalties underlying Gravitational and ClaPROAR. In Figure 7 the conventional LS generator with $\gamma = 0.5$ serves as our baseline. Evidently, being more conservative or using one of our proposed penalties decreases the estimated domain and model shifts, in some cases beyond significance.

Finally, Figure 8 shows the results for our real-world data. We note that for both the California Housing and GMSC data,

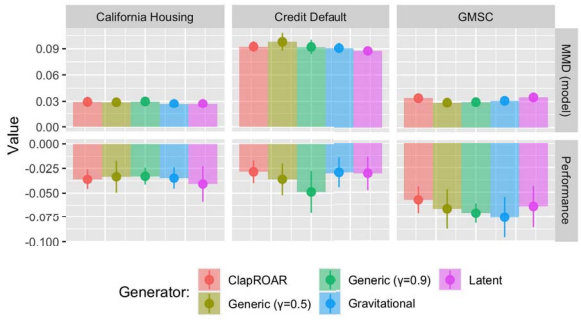

Fig. 8. The differences in counterfactual outcomes when using the various mitigation strategies compared to the baseline approach, that is Wachter with $\gamma = 0.5$. Results for the MLP using real-world datasets. The shown model MMD (PP MMD) was computed over actual samples, rather than a mesh grid. Error bars indicate the standard deviation across folds.

ClaPROAR does have an attenuating effect on model performance deterioration[12]. Overall, the results are less significant, possibly because a somewhat smaller share of individuals from the non-target group received recourse than in the synthetic case[13].

VIII. Discussion

Our results in Section VI indicate that state-of-the-art approaches to Algorithmic Recourse induce substantial domain and model shift if implemented at scale in practice. These induced shifts can and should be considered as an (expected) external cost of individual recourse. While they do not affect the individual directly as long as we look at the individual in isolation, they can be seen to affect the broader group of stakeholders in automated data-driven decision-making. We have seen, for example, that out-of-sample model performance generally deteriorates in our simulation experiments. In practice, this can be seen as a cost to model owners, that is the group of stakeholders using the model as a decision-making tool. As we have set out in Example I.2 of our introduction, these model owners may be unwilling to carry that cost, and hence can be expected to stop offering recourse to individuals altogether. This in turn is costly to those individuals that would otherwise derive utility from being offered recourse.

So, where does this leave us? We would argue that the expected external costs of individual recourse should be shared by all stakeholders. The most straightforward way to achieve this is to introduce a penalty for external costs in the counterfactual search objective function, as we have set out in Equation (4). This will on average lead to more costly counterfactual outcomes, but may help to avoid extreme scenarios, in which minimal-cost recourse is reserved to a tiny minority of individuals. We have shown various types of shift-mitigating strategies that can be used to this end. Since all of these strategies can be seen simply as a specific adaption

[12]Estimated domain shifts (not shown) were largely insubstantial, as in Figure 4 in the previous section.

[13]In earlier experiments we moved a larger share of individuals and the results more clearly favoured our mitigation strategies.

of Equation (4), they can be applied to any of the various counterfactual generators studied here.

IX. Limitations and Future Work

While we believe that this work constitutes a valuable starting point for addressing existing issues in Algorithmic Recourse from a fresh perspective, we are aware of several of its limitations. In the following, we highlight some of these and point to avenues for future research.

A. Private vs. External Costs

Perhaps the most crucial shortcoming of our work is that we merely point out that there exists a trade-off between private costs to the individual and external costs to the collective of stakeholders. We fall short of providing any definitive answers as to how that trade-off may be resolved in practice. The mitigation strategies we have proposed here provide a good starting point, but they are ad-hoc extensions of the existing AR framework. An interesting idea to explore in future work could be the potential for Pareto optimal Algorithmic Recourse, that is, a collective recourse outcome in which no single individual can be made better off, without making at least one other individual worse off. This type of work would be interdisciplinary and could help to formalize some of the concepts presented in this work.

B. Experimental Setup

The experimental setup proposed here is designed to mimic a real-world recourse process in a simple fashion. In practice, models are updated regularly [14]. We also find it plausible to assume that the implementation of recourse happens periodically for different individuals, rather than all at once at time $t = 0$. That being said, our experimental design is a vast oversimplification of potential real-world scenarios. In practice, any endogenous shifts that may occur can be expected to be entangled with exogenous shifts of the nature investigated in Upadhyay et al. [14]. We also make implicit assumptions about the utility functions of the involved agents that may well be too simple: individuals seeking recourse are assumed to always implement the proposed Counterfactual Explanations; conversely, the agent in charge of the model M is assumed to always treat individuals that have implemented valid recourse as if they were truly now in the target class.

C. Causal Modelling

In this work, we have focused on popular counterfactual generators that do not incorporate any causal knowledge. The generated perturbations therefore may involve changes to variables that affect the outcome predicted by the black-box model, but not the true outcome. The implementation of such changes is typically described as **gaming** [36], although they need not be driven by adversarial intentions: in Example I.2, student applicants may dutifully focus on acquiring credentials that help them to be admitted to university, but ultimately not to improve their chances of success at completing their degree [37]. Preventing such actions may help to avoid the dynamics we have pointed to in this work. Future work would likely benefit from including recent approaches to AR that incorporate causal knowledge such as Karimi et al. [13].

D. Classifiers

For reasons stated earlier, we have limited our analysis to differentiable linear and non-linear classifiers, in particular logistic regression and deep neural networks. While these sorts of classifiers have also typically been analyzed in the existing literature on Counterfactual Explanations and Algorithmic Recourse, they represent only a subset of popular machine learning models employed in practice. Despite the success and popularity of deep learning in the context of high-dimensional data such as image, audio and video, empirical evidence suggests that other models such as boosted decision trees may have an edge when it comes to lower-dimensional tabular datasets, such as the ones considered here ([38], [39]).

E. Data

Largely in line with the existing literature on Algorithmic Recourse, we have limited our analysis of real-world data to three commonly used benchmark datasets that involve binary prediction tasks. Future work may benefit from including novel datasets or extending the analysis to multi-class or regression problems, the latter arguably representing the most common objective in Finance and Economics.

X. Concluding Remarks

This work has revisited and extended some of the most general and defining concepts underlying the literature on Counterfactual Explanations and, in particular, Algorithmic Recourse. We demonstrate that long-held beliefs as to what defines optimality in AR, may not always be suitable. Specifically, we run experiments that simulate the application of recourse in practice using various state-of-the-art counterfactual generators and find that all of them induce substantial domain and model shifts. We argue that these shifts should be considered as an expected external cost of individual recourse and call for a paradigm shift from individual to collective recourse in these types of situations. By proposing an adapted counterfactual search objective that incorporates this cost, we make that paradigm shift explicit. We show that this modified objective lends itself to mitigation strategies that can be used to effectively decrease the magnitude of induced domain and model shifts. Through our work, we hope to inspire future research on this important topic. To this end we have open-sourced all of our code along with a Julia package: `AlgorithmicRecourseDynamics.jl`. Future researchers should find it easy to replicate, modify and extend the simulation experiments presented here and apply them to their own custom counterfactual generators.

Acknowledgements

Some of the members of TU Delft were partially funded by ICAI AI for Fintech Research, an ING — TU Delft collaboration.

References

[1] C. O'Neil, *Weapons of math destruction: How big data increases inequality and threatens democracy*. Crown, 2016.

[2] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[3] M. Pawelczyk, S. Bielawski, J. van den Heuvel, T. Richter, and G. Kasneci, "Carla: A python library to benchmark algorithmic recourse and counterfactual explanation algorithms," 2021. Available: https://arxiv.org/abs/2108.00783

[4] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *Harv. JL & Tech.*, vol. 31, p. 841, 2017.

[5] P. Altmeyer, "CounterfactualExplanations.jl - a Julia package for Counterfactual Explanations and Algorithmic Recourse." 2022. Available: https://github.com/pat-alt/CounterfactualExplanations.jl

[6] L. Schut *et al.*, "Generating Interpretable Counterfactual Explanations By Implicit Minimisation of Epistemic and Aleatoric Uncertainties," in *International Conference on Artificial Intelligence and Statistics*, 2021, pp. 1756–1764.

[7] S. Joshi, O. Koyejo, W. Vijitbenjaronk, B. Kim, and J. Ghosh, "Towards realistic individual recourse and actionable explanations in black-box decision making systems," 2019. Available: https://arxiv.org/abs/1907.09615

[8] R. K. Mothilal, A. Sharma, and C. Tan, "Explaining machine learning classifiers through diverse counterfactual explanations," in *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 2020, pp. 607–617.

[9] J. Antorán, U. Bhatt, T. Adel, A. Weller, and J. M. Hernández-Lobato, "Getting a clue: A method for explaining uncertainty estimates," 2020. Available: https://arxiv.org/abs/2006.06848

[10] A.-H. Karimi, G. Barthe, B. Schölkopf, and I. Valera, "A survey of algorithmic recourse: Definitions, formulations, solutions, and prospects," 2020. Available: https://arxiv.org/abs/2010.04050

[11] S. Verma, J. Dickerson, and K. Hines, "Counterfactual explanations for machine learning: A review," 2020. Available: https://arxiv.org/abs/2010.10596

[12] B. Ustun, A. Spangher, and Y. Liu, "Actionable recourse in linear classification," in *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 2019, pp. 10–19.

[13] A.-H. Karimi, B. Schölkopf, and I. Valera, "Algorithmic recourse: From counterfactual explanations to interventions," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021, pp. 353–362.

[14] S. Upadhyay, S. Joshi, and H. Lakkaraju, "Towards Robust and Reliable Algorithmic Recourse," 2021. Available: https://arxiv.org/abs/2102.13620

[15] E. Carrizosa, J. Ramırez-Ayerbe, and D. Romero, "Generating Collective Counterfactual Explanations in Score-Based Classification via Mathematical Optimization," 2021.

[16] S. Rabanser, S. Günnemann, and Z. Lipton, "Failing loudly: An empirical study of methods for detecting dataset shift," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[17] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[18] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine learning*, vol. 23, no. 1, pp. 69–101, 1996.

[19] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM computing surveys (CSUR)*, vol. 46, no. 4, pp. 1–37, 2014.

[20] K. Nelson, G. Corbin, M. Anania, M. Kovacs, J. Tobias, and M. Blowers, "Evaluating model drift in machine learning algorithms," in *2015 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, 2015, pp. 1–8.

[21] S. Ackerman, P. Dube, E. Farchi, O. Raz, and M. Zalmanovici, "Machine Learning Model Drift Detection Via Weak Data Slices," in *2021 IEEE/ACM Third International Workshop on Deep Learning for Testing and Testing for Deep Learning (DeepTest)*, 2021, pp. 1–8.

[22] R. M. B. de Oliveira and D. Martens, "A framework and benchmarking study for counterfactual generating methods on tabular data," *Applied Sciences*, vol. 11, no. 16, p. 7274, 2021.

[23] A.-K. Dombrowski, J. E. Gerken, and P. Kessel, "Diffeomorphic explanations with normalizing flows," 2021.

[24] R. S. Pindyck and D. L. Rubinfeld, *Microeconomics*. Pearson Education, 2014.

[25] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *The Journal of Machine Learning Research*, vol. 13, no. 1, pp. 723–773, 2012.

[26] A. Berlinet and C. Thomas-Agnan, *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[27] M. A. Arcones and E. Gine, "On the bootstrap of U and V statistics," *The Annals of Statistics*, pp. 655–674, 1992.

[28] S. Hanneke, "A bound on the label complexity of agnostic active learning," in *Proceedings of the 24th international conference on Machine learning*, 2007, pp. 353–360.

[29] M. Innes *et al.*, "Fashionable modelling with flux," 2018. Available: https://arxiv.org/abs/1811.01457

[30] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," 2016. Available: https://arxiv.org/abs/1612.01474

[31] Kaggle, "Give me some credit, Improve on the state of the art in credit scoring by predicting the probability that somebody will experience financial distress in the next two years." Kaggle, 2011. Available: https://www.kaggle.com/c/GiveMeSomeCredit

[32] I.-C. Yeh and C. Lien, "The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients," *Expert systems with applications*, vol. 36, no. 2, pp. 2473–2480, 2009.

[33] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[34] R. K. Pace and R. Barry, "Sparse spatial autoregressions," *Statistics & Probability Letters*, vol. 33, no. 3, pp. 291–297, 1997.

[35] M. Carlisle, "Racist data destruction? - a Boston housing dataset controversy," 2019, Available: https://medium.com/@docintangible/racist-data-destruction-113e3eff54a8

[36] J. Miller, S. Milli, and M. Hardt, "Strategic Classification is Causal Modeling in Disguise," in *Proceedings of the 37th International Conference on Machine Learning*, Nov. 2020, pp. 6917–6926. Accessed: Nov. 03, 2022. [Online]. Available: https://proceedings.mlr.press/v119/miller20b.html

[37] S. Barocas, M. Hardt, and A. Narayanan, "Fairness and machine learning," Dec. 23, 2022. https://fairmlbook.org/index.html (accessed Nov. 08, 2022).

[38] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep neural networks and tabular data: A survey," 2021. Available: https://arxiv.org/abs/2110.01889

[39] L. Grinsztajn, E. Oyallon, and G. Varoquaux, "Why do tree-based models still outperform deep learning on tabular data?" 2022. Available: https://arxiv.org/abs/2207.08815

APPENDIX

Granular results for all of our experiments can be found in this online companion: https://www.paltmeyer.com/endogenous-macrodynamics-in-algorithmic-recourse/. The Github repository containing all the code used to produce the results in this paper can be found here: https://github.com/pat-alt/endogenous-macrodynamics-in-algorithmic-recourse.