Quantifying the Endogenous Domain and Model Shifts Induced by the DiCE Generator

Aleksander Buszydlik
Supervisors: Cynthia C. S. Liem, Patrick Altmeyer
EEMCS, Delft University of Technology, The Netherlands

June 18, 2022

A Dissertation Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Abstract

Algorithmic recourse aims to provide individuals affected by a negative classification outcome with actions which, if applied, would flip this outcome. Various approaches to the generation of recourse have been proposed in the literature; these are typically assessed on statistical measures such as the validity of generated explanations or their proximity to the training data. However, little to no attention has been paid to the underlying dynamics of recourse. If a group of individuals applies the suggested actions, they may over time induce a shift in the domain or model. We propose a framework for the measurement of such intrinsic shifts, and conduct an analysis of the dynamics of recourse implemented by the generators proposed by Mothilal *et al.* and Wachter *et al.*. Our results suggest that the application of recourse is likely to introduce statistically significant shifts in the system, and that the underlying dataset and model impact the behavior of the generators.

1 Introduction

Many of the commonly employed artificial intelligence techniques have to be treated as black boxes that are impossible to interpret even by experts. At the same time, the ever-increasing impact of these techniques on different areas of human life – from job recruitment to healthcare interventions [1], [2] – emphasizes the importance of transparent decision-making. To that end, the Explainable Artificial Intelligence paradigm has been introduced as an umbrella term for various approaches to increase, among other things, the interpretability and transparency of AI [3]. As suggested in the foundational work of Wachter *et al.*, explainability may be achieved even "without opening the black box" by providing the stakeholders affected by a classification with a set of actions that, if applied, would lead to a different outcome [4]. Such sets of actions may take the form of counterfactual explanations (CEs) informing how the state of the world would have to (realistically) change for the outcome to change (*"if your salary had been €3000 per month, you would have been approved for the loan"*).

Explanations in form of actionable changes are referred to as *algorithmic recourse*. Previous work has presented various approaches for the generation of recourse which typically extend on the optimization problem introduced by Wachter *et al.* [4] that serves as the baseline for our work. While our framework can be used with many generators, we specifically focus on *Diverse Counterfactual Explanations* (`DiCE`) which aims to promote the actionability of suggestions for the interested stakeholders with the addition of diversity constraints [5]. In our research, we use the implementations of the aforementioned techniques available in the *Counterfactual And Recourse LibrAry* (`CARLA`) which is a Python framework "for benchmarking counterfactual explanation methods" [6].

To the best of our knowledge, no research has been conducted on the intrinsic dynamics of the algorithmic recourse. After it is implemented for some individuals, the domain and model may shift even if no new samples are introduced to the model. One can imagine how individuals unhappy with their loan decision request explanations from the bank and implement them. When an individual applies the recommendation some values for their attributes change, which over time leads to a domain shift. As the bank attempts to maximize the performance of its machine learning model, it is periodically retrained on this updated dataset which in turn leads to a model shift.

This is an important limitation as even small shifts may disrupt the performance of models such as deep neural networks [7]. Shifts of the model may inadvertently change the classification outcomes for the individuals who were not part of the recourse group. Shifts in the domain include the formation of new clusters of data points. As the generators generally attempt to find low-cost explanations, such clusters of individuals who have undergone recourse may be created in the vicinity of the (old) decision boundary. Shifts that stem from the application of recourse are *endogenous* in nature [8].

Our work was conducted to answer the research question RQ1: *what are the differences in the characteristics of the domain and model shifts induced by the* `DiCE` *and Wachter et al. recourse generators?*. To that end, we answer six sub-questions:

1. *(SQ1) how can the domain shifts due to the application of algorithmic recourse be measured?*
2. *(SQ2) how can the model shifts due to the application of algorithmic recourse be measured?*
3. *(SQ3) what are the characteristics of shifts induced by the Wachter et al. generator?*

4. *(SQ4) what are the characteristics of shifts induced by the* `DiCE` *generator?*
5. *(SQ5) what factors may influence the potential difference in the dynamics of recourse?*
6. *(SQ6) what appear to be good ways to mitigate the potential endogenous shifts?*

Our contributions to the state of knowledge are two-fold. First, we introduce a framework built on top of `CARLA` which allows for the measurement of algorithmic recourse in terms of the endogenous domain and model shifts. Our framework can be applied to any recourse generator; its implementation is provided along with the paper[1]. Second, we use this framework to provide the first in-depth analysis of the dynamics of recourse induced by Wachter *et al.* and `DiCE` generators.

This paper conforms to the following structure. In Section 2 we analyze the previous research in the relevant domains. Then, Section 3 provides an overview of the proposed experimental procedure. Section 4 describes the experiments which were conducted to analyze the dynamics of recourse. We place the results in a broader context in Section 5. Subsequently, Section 6 is a discussion on the ethics and reproducibility of our research. Finally, Section 7 is the conclusion to this paper.

2 Related work

In this Section we provide a review of the relevant literature. First, Subsection 2.1 discusses the existing research within the domain of counterfactual explanations and algorithmic recourse. Then, Subsection 2.2 describes the previous work on the measurement of dataset and model shifts.

2.1 Algorithmic recourse

Many approaches for the generation of algorithmic recourse have been described in the literature. An October 2020 survey by Karimi *et al.* discovered 60 algorithms that have been proposed since 2014 [9]. Another survey published in the same month by Verma *et al.* described 29 algorithms [10]. There exists a large variety in terms of the objective functions, employed tools (from brute force through gradient-based approaches to graph traversal algorithms), and further constraints placed on the generated counterfactuals such as actionability, plausibility, diversity, or sparsity.

Our paper focuses on the Wachter *et al.* and `DiCE` recourse generators. The former is proposed in [4] as a simple optimization problem where the generator attempts to find the closest possible counterfactual explanation for some original point from the training set. This is implemented as a gradient descent procedure which continues until a CE that satisfies some decision threshold is found. The authors of `DiCE` build on this approach to develop a recourse generator with diversity constraints which aim to provide stakeholders with multiple sets of actions that would flip the classification outcome. To that end, `DiCE` employs determinantal point processes [11] to generate multiple CEs for every factual instance. While the algorithm of Wachter *et al.* finds the most feasible (actionable) counterfactual instances by default, `DiCE` increases the feasibility of its suggestions post hoc.

Although we leave the analysis of the dynamics of other generators as future work, we summarize the approaches of some other generators to emphasize the variety in the field. Dhurandhar *et al.* introduce `CEM` [12] which makes use of a gradient-based procedure to discover what may be present and what must be absent to classify a sample. Antorán *et al.* suggest `CLUE` [13], a method to analyze the factors which influence the certainty of neural networks in a Bayesian setting. `FACE` [14], presented by Poyiadzi *et al.*, employs a graph-based algorithm to avoid explanations which are not feasible for the interested individual. Finally, Joshi *et al.* provide `REVISE` [15] which attempts to produce realistic recourse using a variational auto-encoder to model the underlying data manifold.

Preceding studies on the impact of algorithmic recourse had focused mainly on the quality of generated explanations in terms of statistical measures such as the validity of explanations [6], [9], [16] in static systems. A review of Verma *et al.* called for recourse generators which can work in dynamic systems [10] in their 9th research challenge. Later, Upadhyay *et al.* suggested potential adaptations to the generators to increase their robustness against model shifts due to, for example, temporal or geospatial reasons [17]. Nonetheless, the changes investigated in their work are strictly of external origin (*exogenous*) because they occur regardless of the application of recourse.

[1] https://github.com/abuszydlik/model-shifts-with-dice

2.2 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 [18]. This task is a generalization of the anomaly detection problem for large datasets – could two sets of samples have been generated from the same probability distribution. Numerous approaches to anomaly detection have been summarized by Chandola *et al.* [19]. Another well research topic is that of concept drift – situations where external variables influence the patterns between the input and the output of a model [20]. For instance, Gama *et al.* offer a review of the adaptive learning techniques which can handle concept drift [21]. Less work is available on the related topic of model drift - deterioration of model performance over time. Nelson *et al.* review how resistant different machine learning models are to the model drift. [22]. Ackerman *et al.* offer a method to detect changes in model performance when ground truth is not available [23]. In our research, we are interested in quantifying the characteristics of changes to the model, such as the position of the decision boundary, rather than only detecting these changes. We have not identified previous work on this topic.

3 Methods

As one of our main contributions, we introduce a framework built on top of `CARLA` for the measurement and comparison of endogenous domain and model shifts. Our experimental procedure explained in Subsection 3.1 allows for the parallel implementation of algorithmic recourse using multiple generators. We describe synthetic and real-world datasets used in our experiments in Subsection 3.2. Finally, in Subsection 3.3 we explain the metrics used to compare the dynamics of induced recourse.

3.1 Procedure

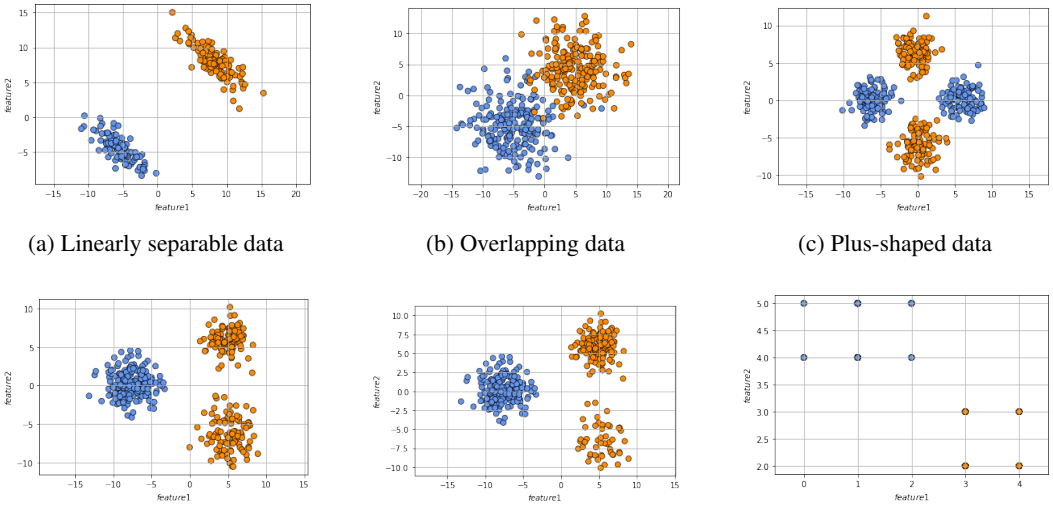
We propose the following experimental procedure to investigate the dynamics of model shifts. It aims to simulate the application of recourse for an increasing number of individuals over multiple rounds. This corresponds to the previously mentioned bank system that generates counterfactual explanations for its customers who make use of them to change their classification outcomes.

1. Sample records from the dataset D to estimate the original probability distribution.
2. Split D into a training set and a test set, use the former to train a classifier M.
3. Quantify the performance of M.
4. Create independent copies D_g and M_g of D and M for each generators g.
5. Calculate in each round the sets of negative instances S predicted by each model.
6. Find the intersection S_r of these sets which will be used to generate recourse.
7. Generate CEs for a set of k samples from S_r, yielding updated datasets $D_g^{'}$.
8. Retrain M_g on $D_g^{'}$ for each generator.
9. Apply the metrics explained in subsection 3.3 to assess the dynamics of shifts.
10. Measure the quality of recourse with `CARLA` benchmarking tools after all N rounds.

A noteworthy practical consideration is the choice of N and k. The higher these values, the more factual instances undergo recourse throughout an experiment. Of course, this is likely to lead to more pronounced domain and model shifts. At the same time, it is generally improbable that a very large part of the population would request an explanation of the algorithm's decisions. In our experiments, we choose the values so that $N \cdot k$ corresponds to the application of recourse on 25-50% of the negative instances from the initial dataset. As we collect data on every round of the experiment, we can also verify the impact of recourse when it is implemented for a smaller number of individuals. Our experiment can be conducted on an arbitrary number of algorithmic recourse generators in parallel. As all generators make use of the same initial model and initial dataset, the differences in domain and model shifts observed throughout the rounds depend solely on the employed generator.

3.2 Datasets

In our experiments we quantify different potential characteristics of recourse using 6 synthetic binary classification datasets consisting of 200-400 samples grouped in normally-distributed clusters[2]. Our datasets are presented in Figure 1 (see also Appendix A for a formal description). Samples from the negative class are marked in blue while samples of the positive class are marked in orange.



(a) Linearly separable data (b) Overlapping data (c) Plus-shaped data

(d) Two balanced clusters (e) Two unbalanced clusters (f) Categorical data

Figure 1: A visualization of the synthetic classification datasets used in our experiments.

Ex ante we expect to see that Wachter *et al.* will create a new cluster of counterfactual instances in the proximity of the initial decision boundary. Thus, the choice of a black-box model may have an impact on the paths of the recourse. In cases 1c, 1d, and 1e with two clusters of points from the positive class, these CEs will be consistently generated closer to one of the positive clusters. In the case 1e, we should see that the CEs are closer to the upper positive cluster which consists of more samples. At the same time, we expect to see the counterfactual explanations of `DiCE` spread around the classification space. In the aforementioned cases, this means that the two initial positive clusters will likely be merged by the generated counterfactuals.

Additionally, we use two real-world datasets from the domain of banking. First, `Give Me Some Credit` dataset with the task to predict whether a borrower is likely to experience financial difficulties in the next two years [24]. Originally consisting of 250000 instances with 11 numerical attributes, the dataset was processed by selecting a sample of 3000 records in a balanced manner resulting in 1500 individuals in the positive class and 1500 individuals in the negative class[3]. Second, `German Credit` dataset with the task to predict the credit risk of bank customers [25]. It consists of 700 positive and 300 negative instances with 7 numerical and 13 categorical attributes. We process the dataset in two ways: (1) the values of the "Personal status and sex" feature are aggregated by the two represented genders; (2) the most common values are calculated for all categorical features such that a feature `F` with the mode `V` is transformed into a new binary feature `is_V`[4].

3.3 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 the 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

[2] https://github.com/abuszydlik/model-shifts-with-dice/blob/main/notebooks/synthetic_datasets.ipynb

[3] https://github.com/drobiu/recourse_analysis/blob/master/notebooks/Dataset_subsampling.ipynb

[4] https://github.com/abuszydlik/model-shifts-with-dice/blob/main/notebooks/GC_processing.ipynb

classification setting, our choice of metrics should remain applicable in the future research on multi-class recourse problems. Second, the set of metrics should allow to capture various aspects of the previously mentioned magnitude, path, and tempo of changes while remaining as small as possible.

Metrics for the domain shifts

Operating point of the k-means algorithm is used to measure whether the counterfactual instances tend to be generated within the positive class or as a separate cluster of data points. It is desirable that CEs resemble the actual positive instances – an accurate representation of the underlying probability distribution will generally reduce domain and model shifts. Thus, the operating point (elbow) of the k-means algorithm should not change over time for a successful recourse generator. While there exist several algorithms for the automated discovery of elbow points, in the framework we employ the Kneedle method introduced in [26] to suggest whether the operating point has changed between rounds. Kneedle selects the optimal value of `k` by rotating the inertia curve onto the horizontal axis and selecting a local minimum. While working with the synthetic datasets we also use domain knowledge and confirm the results of Kneedle through the visual inspection of data.

Unbiased estimate of the squared population Maximum Mean Discrepancy (MMD) given as:

$$MMD_u^2[\mathcal{F}, X, Y] = \frac{1}{m(m-1)} \sum_{i=1}^{m} \sum_{j \neq i}^{m} k(x_i, x_j) + \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} k(y_i, y_j) - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} k(x_i, y_j)$$

where $\mathcal{F}$ is a unit ball in a Reproducing Kernel Hilbert Space $\mathcal{H}$ [27], and X, Y represent independent and identically distributed samples drawn from probability distributions p and q respectively [28]. *MMD* is a measure of the distance between the kernel mean embeddings of p and q in *RKHS* $\mathcal{H}$.
An important consideration is the choice of the kernel function $k(\cdot, \cdot)$. In our implementation we make use of the radial basis function (*RBF*) kernel with a constant length-scale parameter of 0.5. As *RBF* captures all moments of distributions p and q, $MMD_u^2[\mathcal{F}, X, Y] = 0$ if and only if $X = Y$.

At the beginning of the experiment, we sample the initial distribution of both classes and use the samples to calculate changes in the *MMD* after every round of recourse. We additionally follow the ideas of [29] to measure the statistical significance of the observed shifts under the null hypothesis that samples X and Y were drawn from the same probability distribution. To that end, we combine the two samples and generate a large number of permutations of $X + Y$. Then, we split the permuted data into two new samples X' and Y' having the same size as the original. If X and Y have been generated by the same process, $MMD_u^2[\mathcal{F}, X', Y']$ should be approximately equal to $MMD_u^2[\mathcal{F}, X, Y]$. Thus, we can estimate the p-value by calculating how often the latter is greater than or equal to the former.

We calculate the *MMD* for both classes individually based on the ground truth labels. We do not expect the distribution of the negative class to change over time – application of recourse makes this class shrink but it does not mutate the samples. 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 mean the shift in the distribution of the positive class.

Feature mean and feature standard deviation are calculated to verify how the implementation of recourse impacts every attribute in the dataset. Although *MMD* already captures information about the expected value and variance, we may also be interested in the actual values. If little changes in the distribution of a feature are seen, then the generator correctly mimics the probability distribution.

Metrics for the model shifts

Pseudo-distance for the Disagreement Coefficient (Disagreement) introduced in [30] is expressed as $Pr[h(x) \neq h'(x)]$ and describes the probability that two classifiers do not agree on the outcome for a randomly chosen sample. Thus, it is not relevant whether the classification is correct given the ground truth but only whether the sample lies on the same side of the two decision boundaries. In other words, this metric quantifies the overlap between the initial model (trained before the application of recourse) and the updated model. It is desirable that the disagreement is 0 which indicates that the classification has not changed for the interested population. Nonetheless, even two models are in perfect agreement, it is still possible that the decision boundary shifted.

Decisiveness metric is introduced to quantify the likelihood that a model assigns a high probability to its classification of a sample. We define the metric as $\sum_{i=0}^{n}(Pr(x) - 0.5)^2/n$, the average predicted probability that a sample belongs to each of the classes centered around 0.5 which represents the lack of preference for either of the classes. Although the exact values for this metric are unimportant for our study, they can be used to detect the model shifts. If decisiveness changes over time, then it is probable that the decision boundary moves towards either of the classes.

Predicted Probability MMD (PP MMD) is the final measure that we introduce to quantify model shifts. We adapt Maximum Mean Discrepancy as described above and apply it to the probabilities assigned by the model to a set of samples from the dataset. 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. It is worth noting that while we apply the technique to samples drawn from the dataset, it can also be successfully employed on artificial data points selected from the entire classification space. The latter approach is theoretically more robust; however, in practice, it becomes difficult to select enough points to overcome noise, especially in high-dimensional domains.

Metrics for the quality of recourse

We use the evaluation measures as described in `CARLA` [6] to assess the quality of recourse:

- Mean number of features changed, a higher value may suggest lower feasibility of the CEs;
- Mean redundancy measuring how many feature changes were not necessary, a higher value suggests that the CEs impose unnecessary demands on the affected individuals;
- Mean size of the maximum change (Chebyshev distance);
- Mean Taxicab and Euclidean distance between the factual and the counterfactual;

Additionally, we calculate three metrics ourselves:

- Mean predicted probability, measuring the probabilities assigned by the underlying classifier to the newly-generated counterfactual instances. If high, the suggested changes are more reliable and it is less likely that future changes to the model will invalidate them;
- Mean computation time taken to generate a single counterfactual instance. We introduce a time limit (120 seconds) after which the search for a CE is deemed unsuccessful;
- Success rate, measuring the proportion of CEs which are generated within the time limit.

4 Experiments

We measure the dynamics of recourse applied by `DiCE` and Wachter *et al.* in a series of three experiments. First, in Subsection 4.1, we analyze the patterns of recourse given different underlying machine learning models. Then, in Subsection 4.2, we investigate the impact of the data on the dynamics of induced recourse. Finally, in Subsection 4.3, we verify how the hyper-parameters of the `DiCE` generator influence the found counterfactuals. Each experiment was repeated 5 times with different initial conditions, the results presented in this section are averaged over all runs. We consider a CE to be valid if the probability that it is assigned to the positive class is above 0.5. All black-box models are trained using the `RMSProp` optimizer for gradient descent [31] while the internal models of the generators use the `Adam` optimizer [32]. These choices are enforced by `CARLA`.

In our experiments, we use three classifiers with varying levels of complexity. These are always trained over 10 epochs using a stochastic gradient descent procedure with a learning rate of 0.01. In the description of our experiments, we use $C1$, $C2$, and $C3$ to refer to these classifiers.

- ($C1$) a Logistic Regression model;
- ($C2$) a Neural Network, one hidden layer of 5 neurons;
- ($C3$) a Neural Network, two hidden layers of 10 and 5 neurons respectively.

4.1 Impact of the machine learning model on the generated recourse

Recourse is induced over 10 epochs, with 5 randomly selected negative factual instances turned into counterfactuals in each epoch. We simulate the expected usage of `DiCE` by generating 3 diverse

counterfactuals for every factual and randomly turn the factual into one of these instances. We also assess the Wachter *et al.* generator on the default `CARLA` hyper-parameters. This experiment is repeated on two synthetic datasets shown in Figures 1a and 1b. Our results are presented in Table 1.

Model & Generator	*MMD* ↓	*PP MMD* ↓	*Decisiveness* ↑	*Disagreement* ↓	*Clusters*
Dataset: *linearly separable*					
($C1$) `DiCE`	**0.1369** (**)	**0.2392** (**)	**-0.0024**	**0.0580**	2.6
($C1$) Wachter *et al.*	0.3209 (***)	0.3810 (***)	-0.0184	0.2660	**2.0**
($C2$) `DiCE`	**0.1450** (***)	**0.3002** (**)	**0.0000**	**0.0590**	**2.0**
($C2$) Wachter *et al.*	0.3214 (***)	0.3411 (***)	-0.0036	0.2550	**2.0**
($C3$) `DiCE`	**0.1533** (***)	**0.2359** (**)	**0.0000**	**0.0750**	2.6
($C3$) Wachter *et al.*	0.3310 (***)	0.4073 (***)	-0.0182	0.3010	**2.0**
Dataset: *overlapping*					
($C1$) `DiCE`	**0.0275** (ns)	0.2670 (***)	**0.0093**	**0.0260**	**3.0**
($C1$) Wachter *et al.*	0.0854 (**)	**0.2492** (***)	0.0063	0.1535	**3.0**
($C2$) `DiCE`	**0.0401** (ns)	**0.1289** (*)	**0.0009**	**0.0195**	**3.0**
($C2$) Wachter *et al.*	0.0919 (**)	0.1677 (**)	0.0003	0.1190	**3.0**
($C3$) `DiCE`	**0.0270** (ns)	**0.1758** (**)	**-0.0005**	**0.0550**	3.0
($C3$) Wachter *et al.*	0.1047 (**)	0.2909 (***)	-0.0057	0.2212	**2.4**

Table 1: Average shifts of the domain and model with varying complexity of the underlying classifier. Significance levels given in parentheses: (ns) is non-significant, (*) is 5%, (**) is 1%, (***) is 0.1%.

We note that the choice of a model does not have a large impact on the magnitude of domain shifts for either of the generators which can be partly explained by the quasi-linearly-separable nature of the data at hand. We see more variation in terms of the model shifts measured with *PP MMD* although there does not seem to be a clear correlation between the complexity of a model and the value of this metric. Additionally, the inherent characteristics of the algorithm may be a confounding factor. For the Wachter *et al.* generator we observe statistically significant domain and model shifts in all scenarios. Domain shifts induced by `DiCE` are significant on the linearly separable data but in relative terms they are around two times smaller than the baseline. The application of recourse with `DiCE` always results in statistically significant model shifts but again the magnitude of shifts induced by `DiCE` is generally smaller than the baseline. This is also supported by the *Disagreement* metric.

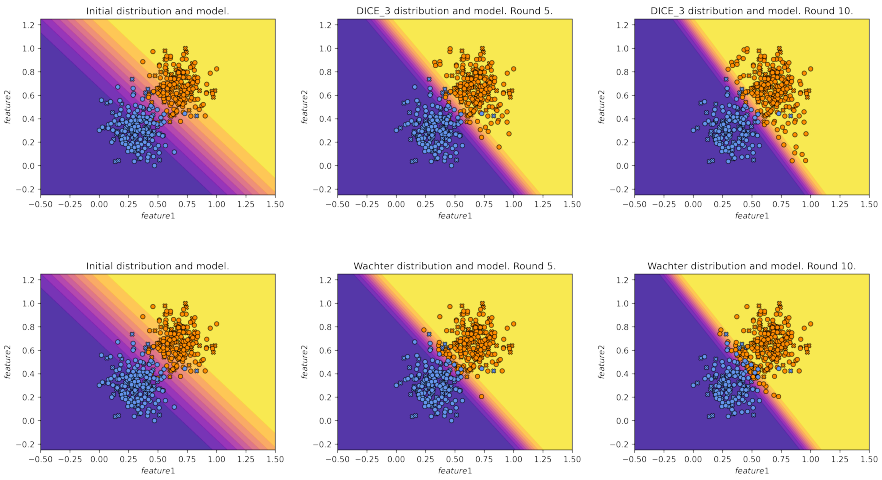


Figure 2: Recourse generated by `DiCE` (top) and Wachter *et al.* (bottom) throughout one experiment.

The Kneedle algorithm typically discovers that both generators create a new cluster of data points although `DiCE` seems to be more prone to change the operating point of k-means. Nonetheless, the

changes in probability distribution induced by Wachter *et al.* are more perceptible when visually inspected. Figure 2 shows the recourse generated using the $C1$ model. While CEs created by DiCE are spread throughout the classification space, Wachter *et al.* takes negative instances towards the decision boundary where they form a new cluster. This is consistent with our initial expectations.

Finally, we did not observe any potential correlation between the type of the black-box model and the benchmark scores calculated by `CARLA` and our framework for the generators.

4.2 Impact of the initial data distribution on the generated recourse

In this experiment, we apply recourse on the synthetic datasets to assess the patterns of generated CEs. We also verify whether these remain present in use-case scenarios using real-world datasets.

Synthetic datasets

We measure the dynamics of algorithmic recourse induced by `DiCE` and Wachter *et al.* on the 6 synthetic datasets described in subsection 3.2. Algorithmic recourse is implemented over 10 rounds with 5 randomly-selected negative instances turned into counterfactuals in every round. $C2$ is used as the black box model. Our results are presented in Table 2 and visualized in Appendix B.

Generator	*MMD* ↓	*PP MMD* ↓	*Disagreement* ↓	*Pred. proba.* ↑	*Success rate* ↑
Dataset: *linearly separable*					
`DiCE`	**0.1531** (**)	**0.2457** (**)	**0.081**	**0.9859**	0.764
Wachter *et al.*	0.3211 (***)	0.3917 (***)	0.254	0.6449	**1.000**
Dataset: *overlapping*					
`DiCE`	**0.0230** (ns)	**0.1523** (**)	**0.030**	**0.9287**	0.884
Wachter *et al.*	0.0877 (*)	0.1896 (**)	0.154	0.5661	**0.940**
Dataset: *plus-shaped*					
`DiCE`	0.0402 (ns)	**0.1700** (***)	**0.030**	**0.9458**	0.972
Wachter *et al.*	**0.0260** (ns)	0.2059 (***)	0.155	0.5885	**1.000**
Dataset: *two balanced clusters*					
`DiCE`	**0.0393** (ns)	**0.1447** (**)	**0.011**	**0.9876**	0.909
Wachter *et al.*	0.1286 (*)	0.1757 (**)	0.135	0.5608	**0.977**
Dataset: *two unbalanced clusters*					
`DiCE`	**0.0373** (ns)	**0.1515** (**)	**0.012**	**0.9864**	0.904
Wachter *et al.*	0.1383 (**)	0.1689 (**)	0.134	0.5638	**0.988**
Dataset: *categorical*					
`DiCE`	**0.1526** (ns)	**0.3430** (***)	**0.216**	**0.9593**	**1.000**
Wachter *et al.*	-	-	-	-	0.000

Table 2: Dynamics of recourse implemented by `DiCE` and Wachter *et al.* on the synthetic datasets. Significance levels given in parentheses: (ns) is non-significant, (*) is 5%, (**) is 1%, (***) is 0.1%.

Our results show that `DiCE` consistently outperforms the Wachter *et al.* generators, leading to smaller domain shifts, model shifts, and maximizing the predicted probability of the counterfactual instances. Only in one case (the plus-shaped dataset) are the changes in distribution more pronounced for `DiCE`; however, they remain statistically insignificant. On four datasets Wachter *et al.* introduces domain shifts that are up to four times larger than those of `DiCE`. A similar trend holds for the model shifts. Although the differences are typically less pronounced when measured with the *Predicted Probability MMD*, the models used by Wachter *et al.* undergo much larger shifts when measured with the *Disagreement* metric. This suggests that `DiCE` can generate CEs on the positive side of the initial decision boundary (resembling more closely the training data), while for Wachter *et al.* these CEs are close to the negative instances. This is confirmed by the *predicted probability* scores. We also note that Wachter *et al.* as implemented by the `CARLA` team does not work for purely categorical

datasets. We observe that categorical variables are never modified by this generator and in the last case, Wachter *et al.* did not generate any counterfactuals.

`DiCE`'s higher robustness against domain and model shifts is at the cost of the success rate. It is more likely to fail at finding a CE within the time limit (120 seconds). Additionally, samples generated by `DiCE` are less actionable as they require larger changes from the individuals. Figure 3 presents the scores on four criteria of actionability averaged over the five synthetic datasets (1f is omitted).

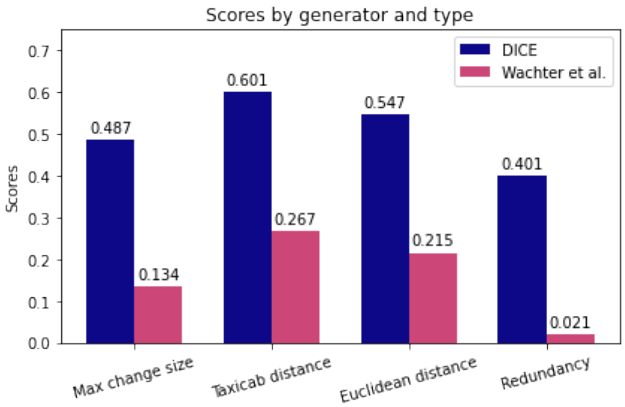


Figure 3: Counterfactual explanations created by `DiCE` are less actionable and more redundant.

We observe that counterfactuals generated by `DiCE` are on average much more demanding to satisfy. Additionally, `DiCE` introduces redundant changes much more often than our baseline.

Real-world datasets

We also assess how the two generators fare on the real-world datasets: `GMSC` (30 rounds with 25 counterfactuals per round) and `German Credit` (15 rounds with 10 counterfactuals per round). We repeat this experiment on $C1$ and $C2$. Our results are summarized in Tables 3 and 4 respectively.

Model & Generator	*MMD* ↓	*PP MMD* ↓	*Disagreement* ↓	*Clusters*	*Euclid. dist.* ↓
		Round 10			
($C1$) `DiCE`	0.0582	0.1971	**0.1578**	3.2	-
($C1$) Wachter *et al.*	**0.0118**	**0.1696**	0.1631	**4.0**	-
($C2$) `DiCE`	0.0485	0.2350	**0.0923**	3.2	-
($C2$) Wachter *et al.*	**0.0322**	**0.2130**	0.1070	**4.0**	-
		Round 20			
($C1$) `DiCE`	0.1098	0.3108	**0.1667**	3.2	-
($C1$) Wachter *et al.*	**0.0386**	**0.2706**	0.1937	**4.0**	-
($C2$) `DiCE`	0.0976	0.2748	**0.0857**	3.6	-
($C2$) Wachter *et al.*	**0.0322**	**0.2642**	0.0987	**4.0**	-
		Round 30			
($C1$) `DiCE`	0.1544 (***)	0.4138 (***)	**0.1737**	4.4	0.9310
($C1$) Wachter *et al.*	**0.0567** (***)	**0.3724** (***)	0.2186	**4.0**	**0.0354**
($C2$) `DiCE`	0.1619 (***)	**0.3422** (***)	**0.0798**	4.8	0.7460
($C2$) Wachter *et al.*	**0.0601** (ns)	0.3444 (***)	0.0955	**4.0**	**0.0257**

Table 3: Dynamics of recourse implemented by the generators on the `GMSC` dataset over 30 rounds. Significance levels given in parentheses: (ns) is non-significant, (*) is 5%, (**) is 1%, (***) is 0.1%.

Model & Generator	*MMD* ↓	*PP MMD* ↓	*Disagreement* ↓	*Clusters*	*Euclid. dist.* ↓
		Round 5			
($C1$) `DiCE`	**0.0514**	**0.1580**	**0.1474**	**4.0**	-
($C1$) Wachter *et al.*	0.0519	0.2213	0.1798	**4.0**	-
($C2$) `DiCE`	0.0516	0.1158	**0.0786**	**4.0**	-
($C2$) Wachter *et al.*	**0.0515**	**0.0934**	0.0848	**4.0**	-
		Round 10			
($C1$) `DiCE`	**0.0396**	**0.1019**	**0.1010**	**3.0**	-
($C1$) Wachter *et al.*	0.0405	0.1450	0.1246	3.2	-
		Last round (15 for $C1$, on average 8 for $C2$)			
($C1$) `DiCE`	**0.0485** (***)	**0.1082** (ns)	0.1128	**3.6**	0.6678
($C1$) Wachter *et al.*	0.0499 (***)	0.1541 (***)	**0.1054**	4.0	**0.3498**
($C2$) `DiCE`	0.0514 (**)	0.1300 (**)	**0.0792**	**4.0**	1.0412
($C2$) Wachter *et al.*	**0.0513** (**)	**0.1044** (**)	0.0856	**4.0**	**0.3485**

Table 4: Dynamics of recourse implemented by the generators on the `German Credit` dataset. Significance levels given in parentheses: (ns) is non-significant, (*) is 5%, (**) is 1%, (***) is 0.1%.

The advantage of `DiCE` over Wachter *et al.* disappears on the `GMSC` dataset where the final domain shifts induced by `DiCE` are up to 2.7 times larger than those of the baseline. `DiCE` also underperforms with regards to the data clustering – the operating point heavily fluctuates from the initial value of 4. At the same time, we note that the magnitude of the model shifts is relatively comparable. We explain the results by looking at the mean Euclidean distance between the factual instance and the corresponding counterfactual – it is approximately 28 times larger for `DiCE`. As it introduces much larger changes than required, it completely fails at modeling the underlying data manifold.

`DiCE` works better on the `German Credit` dataset; however, we again observe the fluctuating number of clusters found by `Kneedle`. The differences in terms of the model shifts are more pronounced on both metrics. We also note a smaller discrepancy in the Euclidean distance metric although again Wachter *et al.* is able to generate more feasible explanations. Importantly, the experiment on $C2$ terminates early due to the lack of common negative instances which suggests that the two models undergo very different changes in terms of the position of the decision boundary.

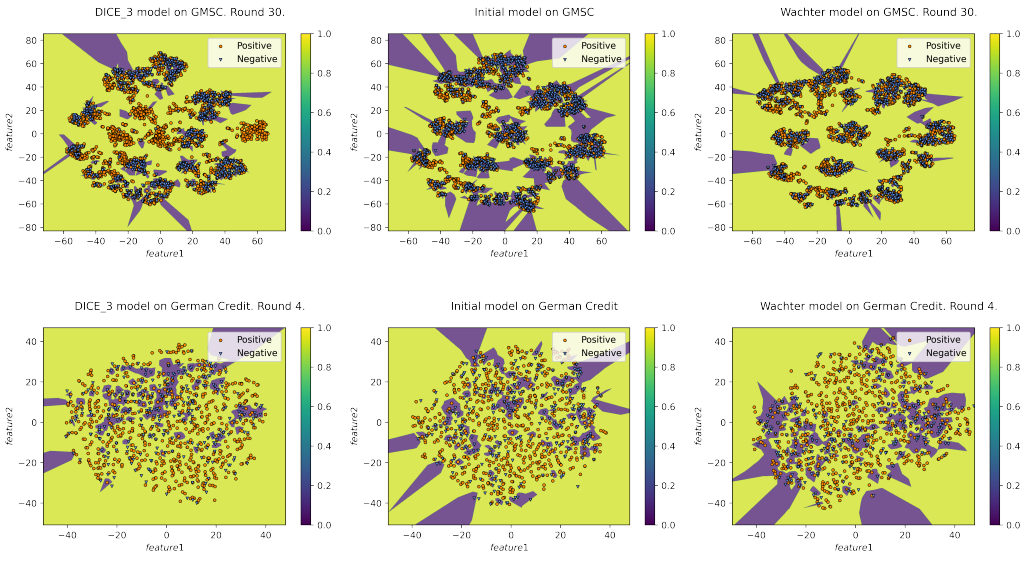


Figure 4: Changes in the decision boundaries of `DiCE` and Wachter *et al.* over one experiment.

To further analyze the dynamics on real-world datasets, we apply the technique described in [33] where the dimensionality of data is first reduced using t-distributed Stochastic Neighbor Embedding [34] and then the decision boundary is approximated with Voronoi tesselation. We approximate the latter step using a 1-NN classifier. Figure 4 above presents the changes to the decision boundary due to the application of recourse. We observe that in both cases DiCE induces erratic model shifts. In particular, on GMSC, DiCE generates four distinct clusters of counterfactual instances at $-20 \leq feature2 \leq 20$ while in the Wachter *et al.* case the structure of clusters resembles the initial data. This is in line with the previously described tendency of DiCE to generate less feasible CEs.

4.3 Impact of the generator hyper-parameters on the generated recourse

In this experiment, we verify whether the hyper-parameters of DiCE impact the dynamics of algorithmic recourse. Again, we induce recourse over 10 epochs, with 5 new factual instances turned into counterfactuals in each epoch. A neural network with one hidden layer of 5 neurons ($C2$) is the underlying black-box model. We obtained the results on the overlapping dataset.

Impact of the number of generated counterfactuals for each factual instance

In this experiment we employ 5 DiCE generators which attempt to find 1, 2, 3, 5, and 8 explanations for each provided factual. Then, we select one of these explanations at random to update the generator's dataset. This simulates a scenario where the client of our bank receives multiple sets of actions and implements the one which they find most feasible. Our results are presented in Table 5.

Generator	*MMD* ↓	*PP MMD* ↓	*Disagreement* ↓	*Success rate* ↑
DiCE (1)	0.0323 (ns)	0.2399 (***)	0.1070	**0.960**
DiCE (2)	**0.0244** (ns)	0.2572 (***)	0.0975	0.936
DiCE (3)	0.0393 (ns)	0.2728 (***)	0.1145	0.876
DiCE (5)	0.0272 (ns)	**0.2253** (***)	**0.0835**	0.836
DiCE (8)	0.0245 (ns)	0.2434 (***)	0.1085	0.728

Table 5: Average shifts for DiCE generators with increasing number of CEs per factual.

We observe that the number of generated counterfactuals for every factual instance does not seem to have an impact on the magnitude of domain shifts which is confirmed by the analysis of changes to the mean and standard deviation. There is also no apparent difference in terms of the model shifts when measured with the *Predicted Probability MMD* and the *Disagreement* metrics.

Impact of the amount of post-processing for each counterfactual instance

In this experiment we employ 6 DiCE generators, each generating 3 counterfactual explanations per factual instance, with an increasing amount of post-processing. The value of this hyper-parameter corresponds to the quantile of the absolute deviation of the feature which will be taken into consideration when attempting to minimize the number of the suggested changes. Table 6 shows the results.

Generator	*MMD* ↓	*PP MMD* ↓	*Mean redundancy* ↓	*Success rate* ↑
DiCE (0.0)	0.0581 (ns)	0.1131 (ns)	0.5880	**1.000**
DiCE (0.2)	0.0412 (ns)	0.0873 (ns)	**0.3934**	0.764
DiCE (0.4)	**0.0249** (ns)	**0.0207** (ns)	0.4549	0.576
DiCE (0.6)	0.0255 (ns)	0.0208 (ns)	0.4579	0.524
DiCE (0.8)	0.0344 (ns)	0.0712 (ns)	0.4331	0.484
DiCE (1.0)	0.0319 (ns)	0.0308 (ns)	0.4414	0.508

Table 6: Average shifts for DiCE generators with increasing fraction of post-processed results.

We would intuitively expect that as the number of post-processing steps increases, the redundancy of suggested changes decreases. Indeed, the counterfactuals generated by the DiCE model with no

post-processing are much more frequently redundant (≈ 1.5 times) than for all other models which negatively impacts their actionability. At the same time, we acknowledge an important limitation of this experiment: increasing the sparsity of the explanations post-hoc becomes very computationally expensive for larger values of the hyper-parameter. This leads to frequent timeouts of the generators – four of them successfully generate a CE in only $\approx 50\%$ of the cases.

5 Discussion

In this section, we review our findings. First, in Subsection 5.1 we summarize the results of our experiments on the synthetic and real-world datasets to answer the research question RQ1. Then, in Subsection 5.2 we analyze the objective functions of both generators to explain our results.

5.1 Characteristics of the observed domain and model shifts

Our results show that both generators are prone to inducing domain and model shifts. We discover that the complexity of the underlying black box model does not have a large impact on the magnitude of the domain shifts measured with *MMD*, but it may have an impact on the operating point of the k-means algorithm. This is especially noticeable on the synthetic datasets although we also observe similar results on the real-world datasets. At the same time, we observe relatively large differences in terms of the model shifts (measured with *PP MMD* and the *Disagreement* metric). Nevertheless, there does not seem to be a clear correlation between the complexity of a model and the magnitude of induced shifts. This holds for synthetic and real-world datasets alike. We note that the underlying data has a much more pronounced impact on the characteristics of the induced shifts. This is expected as domain and model shifts occur when the generator fails at preserving the distribution of the initial samples – some distributions are inherently more difficult to preserve.

In our experiments, we observe several characteristics of domain and model shifts induced by Wachter *et al.*. As the generator simply brings the negative instances to the other side of the decision boundary, it performs well on datasets that are not linearly separable, provided that the model can generate a non-linear boundary. Conversely, it induces larger shifts when the data is linearly separable as the decision boundary lies between the two classes. Although the change in the number of clusters is sometimes not detected by Kneedle, visual inspection of the data confirms that Wachter *et al.* introduces new clusters of data points consisting of its counterfactual explanations.

We find that the dynamics of recourse implemented by `DiCE` are different. On the synthetic datasets, `DiCE` performed better than the baseline generator – while it also induces significant domain and model shifts, these are typically several times smaller in magnitude on the synthetic datasets. Nonetheless, experiments on the real-world datasets suggest that `DiCE` may not be suitable in all cases as its tendency to generate CEs spread throughout the classification space can become a major problem. In particular, the generator suggests much larger changes than required to flip the outcome of a negative instance. This ultimately leads to much worse performance than the baseline. At the same time, we note that recourse implemented by `DiCE` may be considered more robust. Larger changes in the features result in a higher certainty of the classifier that the counterfactual instance belongs to the positive class. Therefore, we can also expect that there is a lower risk that the CEs of `DiCE` become outdated as the model continues to evolve, which is an advantage for the affected stakeholders.

Given these findings, we provide a set of guidelines to mitigate the unwanted endogenous dynamics of recourse. First, we recommend that until a large-scale survey of the domain and model shifts induced by recourse generators becomes available, the interested stakeholders should assess multiple generators to select the best one for their use case. As previously discussed, the underlying data distribution may have a major impact on the behavior of a generator. Second, although our experiments were conducted in a setting where the size of the dataset does not change over time, it may be useful to consider an approach where the initial negative samples remain in the dataset and the counterfactual samples are treated as new data points. It is important to emphasize that although some individuals may request an explanation of their outcome, this does not mean that the initial model was incorrect in assigning this outcome. Thus, preserving negative instances can likely attenuate the magnitude of induced shifts. Finally, we again remark that we generated recourse for 25-50% of individuals with the negative outcome – in practice, it is not likely that such a large number of individuals would request a CE and successfully apply it. This limits the severity of possible shifts.

5.2 Comparison of the objective function

Here we explain the factors which influence the differences in the dynamics of recourse. To that end, we compare the objective functions of the generators. For Wachter *et al.* it is expressed as:

$$\arg\min_{c} \lambda(f(c), y) + d(c,\ x) \tag{1}$$

with the first term nudging the instance in the direction of the positive outcome, and the second term preventing the counterfactual from being generated too far from the original point.

`DiCE`, on the other hand, optimizes the following function:

$$\arg\min_{c1,\ldots,c_k} \frac{1}{k}\sum_{i=1}^{k} \lambda(f(c_i),\ y) + \frac{r_1}{k}\sum_{i=1}^{k} d(c_i,\ x) - r_2\ det(K) \tag{2}$$

it can be noted that the first two terms of Equation 2 fully correspond to the Equation 1 (although summed over all generated counterfactuals). The third term introduces diversity constraints – it is the determinant of a kernel matrix where $K_{i,\ j} = \frac{1}{1+d(c_i,\ c_j)}$. This also means that when $k = 1$, i.e. when `DiCE` generates a single counterfactual instance, Equation 2 simplifies to:

$$\arg\min_{c} \lambda(f(c), y) + r_1\ d(c,\ x) \tag{3}$$

which is the Wachter *et al.* objective function with an additional regularization hyper-parameter. Nonetheless, our experiments suggest that even when $k = 1$, the patterns of recourse presented by the two generators are different which is presented in Figure 5.

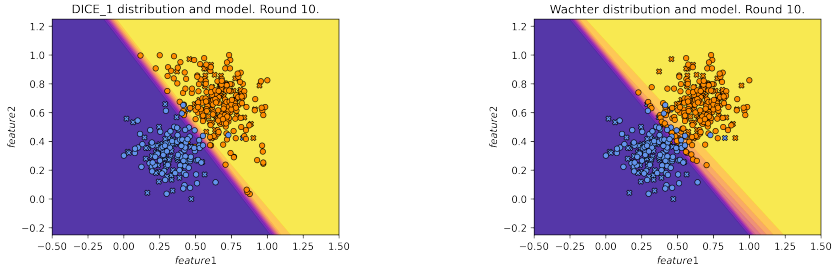


Figure 5: Recourse generated by `DiCE` with $k = 1$ (left) and Wachter *et al.* (right).

It can be observed that when the diversity constraints are removed, `DiCE` still manages to generate counterfactual explanations which are spread around the classification space. This is different from the type of recourse implemented by Wachter *et al.* generator where the samples are brought towards the decision boundary. Additionally, Experiment 4.3 suggestes that, in general, the choice of k does not seem to have a major impact on the dynamics of domain and model shifts induced by `DiCE`. Therefore, the difference in the behavior of the generators cannot be explained the third term of Equation 2. Further, both generators are described in their respective papers using the same distance metric: L_1 norm divided by the sum of the mean absolute deviations of the features. While authors of `DiCE` propose a different metric for categorical features, this also cannot have an impact on the recourse – all but two of our datasets consists of only numerical features. Finally, Wachter *et al.* do not explicitly state the loss function λ, thus, in our experiments, we use binary cross-entropy (BCE). Mothilal *et al.* specifically suggest hinge loss which ensures that valid counterfactuals are not penalized if they satisfy the classification threshold. Although λ may have an impact on the behavior of the generators, we again turn to Figure 5 to explain why this is unlikely to be the only factor. Some of the CEs generated by `DiCE` are "far" from the original decision boundary but this should not be the case if the generator minimizes hinge loss with a threshold of 0.5 as was the case in our experiments.

Thus, we look into the source code of the two algorithms in `CARLA` for another explanation of the differences. Wachter *et al.* is implemented by the framework team[5]. We note that the implementation

[5] https://github.com/carla-recourse/CARLA/blob/main/carla/recourse_methods/catalog/wachter/library/wachter.py

generally follows the principles given in [4] although it uses simple L_1 norm as the distance function rather than the suggested normalized L_1 norm. `DiCE` is available as part of the `InterpretML` package[6] and used in `CARLA` as-is. We discover that – although the authors of `CARLA` refer to a gradient-based implementation of `DiCE` [6] – the framework uses another approach where features are randomly sampled[7] and transformed to increase sparsity. Thus, Equation 2 does not play a role in the counterfactual generation process which explains the differences observed in Section 4.

6 Responsible Research

Our research was conducted without external funding and we do not have any conflicts of interest to report. Throughout the research process we took multiple precautions to maintain the integrity of our results and to uphold high ethical standards of the work. We review our process on two criteria: ethics of the field of research (Subsection 6.1) and future reproducibility of the results (Subsection 6.2).

6.1 Ethical machine learning

Our work was conducted in the domain of trustworthy artificial intelligence with various possible future use cases such as banking, healthcare, job recruitment. The use of black box algorithms for decision-making in cases where the algorithm's verdict has possible impacts on human lives is inherently questionable. While legislation aiming to ensure that affected stakeholders have the right of appeal is being developed around the world [35], [36], the mechanisms in place are still limited. Our work aims to alleviate these issues by introducing a benchmarking framework for the comparison of recourse generators in identical experimental conditions. We see two main scenarios where our framework can increase the explainability, interpretability, and justifiability of black-box decisions. First, we provide a tool for the researchers of algorithmic recourse generators which allows to take into consideration the social welfare of the population by designing generators which are less prone to modifying the domain and model. Second, we hope to empower officials responsible for the introduction of recourse procedures to consider how different generators behave in their use case.

6.2 Reproducibility of the results

We conducted our work on six synthetic and two real-world dataset. We provide the code required to construct the exact versions of our datasets; interested researchers may generate these datasets and use them directly in our framework which is published along with this paper. While we cannot claim that our software is completely bug-free – also because it heavily relies on CARLA which is still under active development – our algorithms were manually tested to minimize the risk that the results are influenced by the faults in our software. To ensure that our findings do not stem from random factors, all experiments were repeated five times and the presented results were averaged over all runs. Additionally, within every single experiment we employed multiple datasets and models to verify whether the observed behavior is inherent to the generator, rather than it being a product of the experimental conditions. Other researchers should be able to repeat our analysis, either using our own datasets and the framework or their own implementation thereof. As we controlled for the randomness of experiments with repeated runs, we opted not to use seeds for the pseudo-random number generators. This also means that while other researchers should arrive to the same conclusions after replicating our steps, the results they obtain will not be exactly equal to our results.

7 Conclusions and Future Work

Our research was conducted to discover the differences in dynamics of algorithmic recourse induced by two generators: one proposed by Wachter *et al.*, the other introduced by Mothilal *et al.* to generate *"diverse counterfactual explanations"*. As no previous work has been conducted on the topic of endogenous domain and model shifts, we introduced a framework that allows for a robust comparison of recourse generators. Our contributions there are two-fold: we suggested an experimental framework which allows to quantify different aspects of the dynamics of recourse, and we made use of this framework to analyze the behavior of the aforementioned generators in different scenarios.

[6] https://github.com/interpretml/DiCE

[7] https://github.com/carla-recourse/CARLA/blob/main/carla/recourse_methods/catalog/dice/model.py

We employ multiple metrics to quantify these dynamics. For the domain shifts, we track the changes in the operating point of the k-means algorithm to verify whether recourse introduces new clusters to the data, and make use of the Maximum Mean Discrepancy metric to quantify the distance between the probability distributions. For the model shifts, we adapt the Disagreement metric from the domain of active learning, apply *MMD* to the probabilities assigned by the classifiers to the samples, and introduce our Decisiveness metric which measures the sum of normalized probabilities.

Our results suggest that both `DiCE` and Wachter *et al.* generators are likely to induce significant shifts to the underlying data and model. Although `DiCE` increases the actionability of suggestions by generating multiple counterfactual explanations for every factual instance, this happens at the cost of their feasibility. Suggestions of Wachter *et al.* require fewer changes from the affected stakeholders; however, possible external changes to the model are much more likely to make them outdated. While our experiments on the synthetic datasets suggest that `DiCE` is better equipped to mimic the original training data, it performs worse in real-world scenarios.

Our work is primarily limited by the availability of statistical methods to measure the model shifts. While there are ample techniques to verify whether the performance of a model deteriorates, no techniques quantifying, for example, the position of the decision boundary with respect to available data have been identified in the literature. Nonetheless, we successfully employ a model visualization technique to gain insight into the dynamics of algorithmic recourse on high-dimensional datasets. Additionally, our work relies on `CARLA` which is still under development. Thus, some components of our framework must work around the current limitations of `CARLA`. Notably, due to the restrictions imposed by `CARLA`, our work was conducted only on a single type of the `DiCE` generator. Lastly, the generation of CEs becomes a computationally expensive process on large domains and and with complex models, thus, our experiments were conducted only on small- and medium-sized datasets.

In the future, our work can be extended with further measures of the endogenous shifts. As parametric dimensionality reduction techniques allow for the application of the same transformation of data in all rounds of recourse, they could be a powerful method to measure the changes in the model. In particular, the technique described in [33] could be applied with parametric t-SNE [37]. Further, a more robust method to determine the operating point of the k-means algorithm would be beneficial for the measurement of domain shifts. As one option, we suggest the "gap statistic" [38]. Other researchers may also focus on the dynamics of algorithmic recourse in multi-class classification settings. For instance, when multiple outcomes can be considered "positive" and the affected individuals may freely request a counterfactual explanation informing how to achieve any of those outcomes. Finally, our solution can be used to perform a large-scale survey of recourse generators as the framework supports the benchmarking of generators which follow the interface defined by `CARLA`. If such analysis is conducted on a diverse collection of real-world datasets, it may be possible to relate the characteristics of these datasets to the behavior of the generators. This in turn could facilitate the implementation of recourse procedures in organizations and positively influence the interpretability and explainability in decision-making domains that have a direct impact on human lives.

References

[1] C. Rudin and J. Radin, "Why are we using black box models in AI when we don't need to? a lesson from an explainable AI competition," *Harvard Data Science Review*, vol. 1, no. 2, Nov. 22, 2019, https://hdsr.mitpress.mit.edu/pub/f9kuryi8. DOI: 10.1162/99608f92.5a8a3a3d. [Online]. Available: https://hdsr.mitpress.mit.edu/pub/f9kuryi8.

[2] W. N. Price, "Big data and black-box medical algorithms," *Science Translational Medicine*, vol. 10, no. 471, eaao5333, 2018. DOI: 10.1126/scitranslmed.aao5333. eprint: https://www.science.org/doi/pdf/10.1126/scitranslmed.aao5333. [Online]. Available: https://www.science.org/doi/abs/10.1126/scitranslmed.aao5333.

[3] The Royal Society, *Explainable AI: The Basics. Policy Briefing*. Nov. 2019, ISBN: 9781782524335.

[4] S. Wachter, B. Mittelstadt, and C. Russell, "Counterfactual explanations without opening the black box: Automated decisions and the GDPR," *Harvard Journal of Law & Technology*, vol. 31, no. 2, 2018. [Online]. Available: https://jolt.law.harvard.edu/assets/articlePDFs/v31/Counterfactual-Explanations-without-Opening-the-Black-Box-Sandra-Wachter-et-al.pdf.

[5] 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*, ACM, Jan. 2020. DOI: 10.1145/3351095.3372850.

[6] 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," in *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, J. Vanschoren and S. Yeung, Eds., 2021. [Online]. Available: https://arxiv.org/abs/2108.00783.

[7] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *CoRR*, vol. abs/1607.02533, 2016. arXiv: 1607.02533. [Online]. Available: http://arxiv.org/abs/1607.02533.

[8] P. Altmeyer and C. Liem. "Endogenous model shifts in algorithmic recourse." (2022), [Online]. Available: https://github.com/pat-alt/counterfactual-explanations-student-project.

[9] A. Karimi, G. Barthe, B. Schölkopf, and I. Valera, "A survey of algorithmic recourse: Definitions, formulations, solutions, and prospects," *CoRR*, vol. abs/2010.04050, 2020. arXiv: 2010.04050. [Online]. Available: https://arxiv.org/abs/2010.04050.

[10] S. Verma, J. P. Dickerson, and K. Hines, "Counterfactual explanations for machine learning: A review," *CoRR*, vol. abs/2010.10596, 2020. arXiv: 2010.10596. [Online]. Available: https://arxiv.org/abs/2010.10596.

[11] A. Kulesza and B. Taskar, *Determinantal Point Processes for Machine Learning*. Hanover, MA, USA: Now Publishers Inc., 2012, ISBN: 1601986289.

[12] A. Dhurandhar, P.-Y. Chen, R. Luss, *et al.*, "Explanations based on the missing: Towards contrastive explanations with pertinent negatives," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18, Montréal, Canada: Curran Associates Inc., 2018, pp. 590–601.

[13] J. Antorán, U. Bhatt, T. Adel, A. Weller, and J. M. Hernández-Lobato, "Getting a CLUE: A method for explaining uncertainty estimates," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*, OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=XSLF1XFq5h.

[14] R. Poyiadzi, K. Sokol, R. Santos-Rodriguez, T. De Bie, and P. Flach, "FACE: Feasible and actionable counterfactual explanations," in *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 344–350, ISBN: 9781450371100. [Online]. Available: https://dl.acm.org/doi/10.1145/3375627.3375850.

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

[16] A. Karimi, B. Schölkopf, and I. Valera, "Algorithmic recourse: From counterfactual explanations to interventions," ser. FAccT '21, Virtual Event, Canada: Association for Computing Machinery, 2021, pp. 353–362, ISBN: 9781450383097. DOI: 10.1145/3442188.3445899. [Online]. Available: https://dl.acm.org/doi/10.1145/3442188.3445899.

[17] S. Upadhyay, S. Joshi, and H. Lakkaraju, "Towards robust and reliable algorithmic recourse," *CoRR*, vol. abs/2102.13620, 2021. arXiv: 2102.13620. [Online]. Available: https://arxiv.org/abs/2102.13620.

[18] S. Rabanser, S. Günnemann, and Z. Lipton, "Failing loudly: An empirical study of methods for detecting dataset shift," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019. [Online]. Available: https://proceedings.neurips.cc/paper/2019/file/846c260d715e5b854ffad5f70a516c88-Paper.pdf.

[19] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, Jul. 2009, ISSN: 0360-0300. DOI: 10.1145/1541880.1541882. [Online]. Available: https://dl.acm.org/doi/10.1145/1541880.1541882.

[20] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine Learning*, vol. 23, Nov. 1994. DOI: 10.1007/BF00116900.

[21] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, Mar. 2014, ISSN: 0360-0300. DOI: 10.1145/2523813. [Online]. Available: https://dl.acm.org/doi/10.1145/2523813.

[22] 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. DOI: 10.1109/CISDA.2015.7208643.

[23] S. Ackerman, P. Dube, E. Farchi, O. Raz, and M. Zalmanovici, "Machine learning model drift detection via weak data slices," *CoRR*, vol. abs/2108.05319, 2021. arXiv: 2108.05319. [Online]. Available: https://arxiv.org/abs/2108.05319.

[24] Kaggle Competition, *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.* [Online]. Available: https://www.kaggle.com/c/GiveMeSomeCredit.

[25] H. Hofmann, *Statlog (German Credit Data) Data Set*, UC Irvine Machine Learning Repository https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data).

[26] V. Satopaa, J. Albrecht, D. Irwin, and B. Raghavan, "Finding a "Kneedle" in a haystack: Detecting knee points in system behavior," in *2011 31st International Conference on Distributed Computing Systems Workshops*, 2011, pp. 166–171. DOI: 10.1109/ICDCSW.2011.20.

[27] A. Berlinet and C. Thomas-Agnan, "Theory," in *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Boston, MA: Springer US, 2004, pp. 1–54, ISBN: 978-1-4419-9096-9. DOI: 10.1007/978-1-4419-9096-9_1.

[28] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, no. 25, pp. 723–773, 2012. [Online]. Available: http://jmlr.org/papers/v13/gretton12a.html.

[29] M. A. Arcones and E. Gine, "On the Bootstrap of U and V Statistics," *The Annals of Statistics*, vol. 20, no. 2, pp. 655–674, 1992. DOI: 10.1214/aos/1176348650. [Online]. Available: https://doi.org/10.1214/aos/1176348650.

[30] S. Hanneke, "A bound on the label complexity of agnostic active learning," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07, Corvalis, Oregon, USA: Association for Computing Machinery, 2007, pp. 353–360, ISBN: 9781595937933. DOI: 10.1145/1273496.1273541. [Online]. Available: https://dl.acm.org/doi/10.1145/1273496.1273541.

[31] G. Hinton, N. Srivastava, and K. Swersky, *Overview of mini-batch gradient descent*, http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980.

[33] M. A. Migut, M. Worring, and C. J. Veenman, "Visualizing multi-dimensional decision boundaries in 2D," *Data Mining and Knowledge Discovery*, vol. 29, no. 1, pp. 273–295, Jan. 2015, ISSN: 1384-5810. DOI: 10.1007/s10618-013-0342-x. [Online]. Available: https://link.springer.com/article/10.1007/s10618-013-0342-x.

[34] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008. [Online]. Available: http://jmlr.org/papers/v9/vandermaaten08a.html.

[35] R. Wyden, C. Booker, and Y. Clarke, *Algorithmic Accountability Act of 2022*, https://www.congress.gov/bill/117th-congress/house-bill/6580/text?r=2&s=1, 2022.

[36] B. Goodman and S. Flaxman, "European Union regulations on algorithmic decision-making and a "right to explanation"," *AI Magazine*, vol. 38, no. 3, pp. 50–57, Oct. 2017. DOI: 10.1609/aimag.v38i3.2741.

[37] L. van der Maaten, "Learning a parametric embedding by preserving local structure," in *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, D. van Dyk and M. Welling, Eds., ser. Proceedings of Machine Learning Research, vol. 5, Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR, Apr. 2009, pp. 384–391. [Online]. Available: http://proceedings.mlr.press/v5/maaten09a/maaten09a.pdf.

[38] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001. DOI: https://doi.org/10.1111/1467-9868.00293. [Online]. Available: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/1467-9868.00293.

A Formal description of the synthetic datasets

Dataset	*Negative class*			*Positive class*		
	N^{o} samples	Means	Cov. matrices	N^{o} samples	Means	Cov. matrices
1a	100	$(-5,\ -5)$	$\begin{pmatrix} 5 & -4 \\ -4 & 5 \end{pmatrix}$	100	$(8,\ 8)$	$\begin{pmatrix} 5 & -4 \\ -4 & 5 \end{pmatrix}$
1b	200	$(-5,\ -5)$	$\begin{pmatrix} 12 & 0 \\ 0 & 12 \end{pmatrix}$	200	$(5,\ 5)$	$\begin{pmatrix} 12 & 0 \\ 0 & 12 \end{pmatrix}$
1c	100 100	$(-6,\ 0)$ $(6,\ 0)$	$\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$ $\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$	100 100	$(-6,\ 0)$ $(6,\ 0)$	$\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$ $\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$
1d	200	$(-7.5,\ 0)$	$\begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$	100 100	$(5,\ -6)$ $(5,\ 6)$	$\begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}$ $\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$
1e	200	$(-7.5,\ 0)$	$\begin{pmatrix} 3 & 0 \\ 0 & 3 \end{pmatrix}$	50 150	$(5,\ -6)$ $(5,\ 6)$	$\begin{pmatrix} 4 & 0 \\ 0 & 4 \end{pmatrix}$ $\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$
1f	100	$(1.5,\ 5)$	-	100	$(4,\ 3)$	-

Table 7: Parameters of the normally-distributed synthetic datasets used in our experiments.

B Visualization of recourse on the synthetic datasets

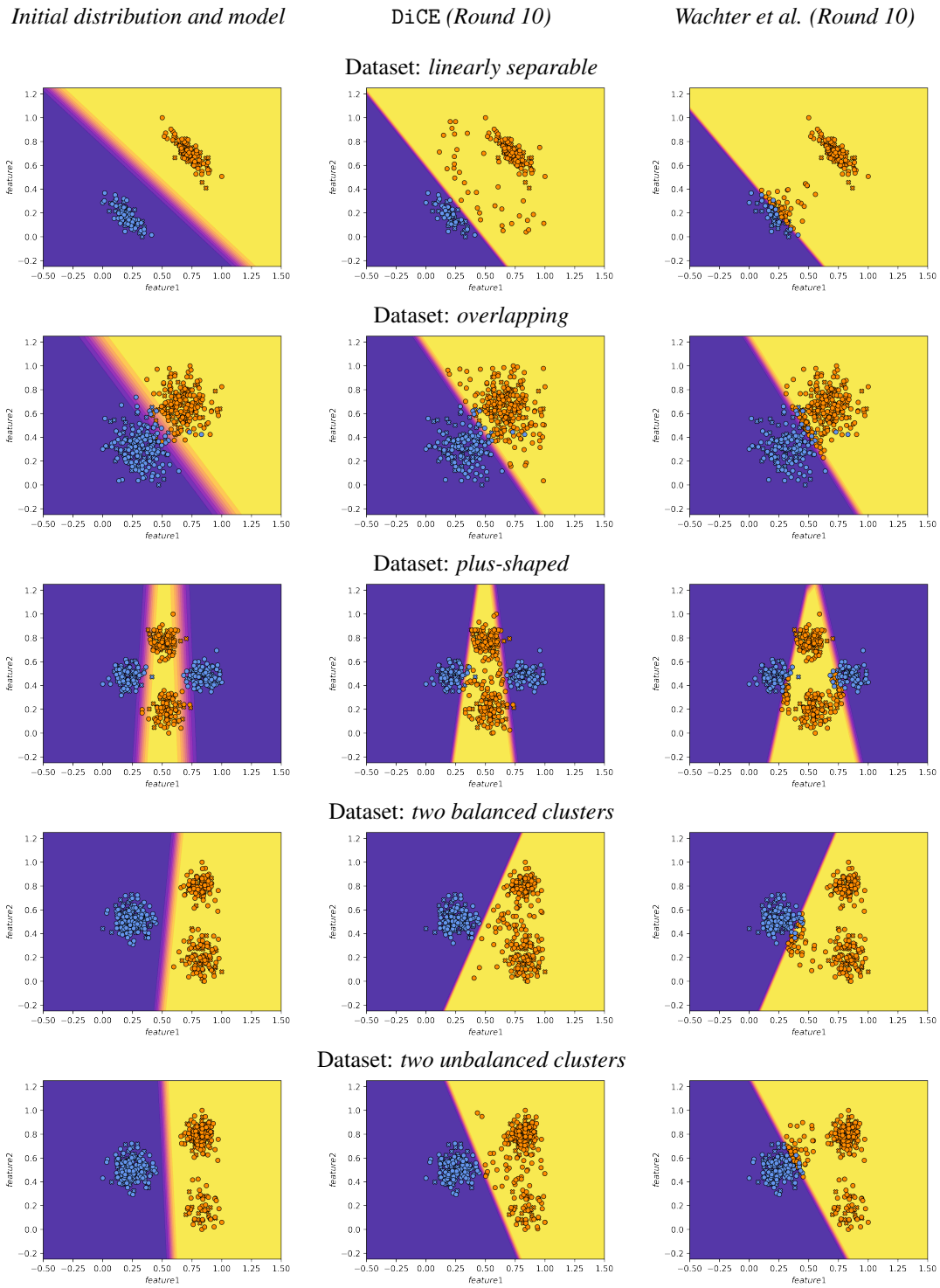


Figure 6: Domain and model shifts induced on the synthetic datasets at the end of round 10.